Collective Analog Bioelectronic Computation

by

Soumyajit Mandal

B.Tech, Indian Institute of Technology (2002)
M.S., Massachusetts Institute of Technology (2004)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of Electrical Engineering and Computer Science
May 18, 2009

Certified by .....
Rahul Sarpeshkar
Associate Professor
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Collective Analog Bioelectronic Computation

by

Soumyajit Mandal

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Abstract

In this thesis, I present two examples of fast-and-highly-parallel analog computation inspired by architectures in biology. The first example, an *RF cochlea*, maps the partial differential equations that describe fluid-membrane-hair-cell wave propagation in the biological cochlea to an equivalent inductor-capacitor-transistor integrated circuit. It allows ultra-broadband spectrum analysis of RF signals to be performed in a rapid low-power fashion, thus enabling applications for universal or software radio. The second example exploits detailed similarities between the equations that describe chemical-reaction dynamics and the equations that describe subthreshold current flow in transistors to create fast-and-highly-parallel integrated-circuit models of protein-protein and gene-protein networks inside a cell. Due to a natural mapping between the Poisson statistics of molecular flows in a chemical reaction and Poisson statistics of electronic current flow in a transistor, stochastic effects are automatically incorporated into the circuit architecture, allowing highly computationally intensive stochastic simulations of large-scale biochemical reaction networks to be performed rapidly.

I show that the exponentially tapered transmission-line architecture of the mammalian cochlea performs constant-fractional-bandwidth spectrum analysis with $O(N)$ expenditure of both analysis time and hardware, where N is the number of analyzed frequency bins. This is the best known performance of any spectrum-analysis architecture, including the constant-resolution Fast Fourier Transform (FFT), which scales as $O(N \log N)$, or a constant-fractional-bandwidth filterbank, which scales as $O\left(N^2\right)$. The RF cochlea uses this bio-inspired architecture to perform real-time, on-chip spectrum analysis at radio frequencies. I demonstrate two cochlea chips, implemented in standard 0.13μm CMOS technology, that decompose the RF spectrum from 600MHz to 8GHz into 50 log-spaced channels, consume $<$ 300mW of power, and possess 70dB of dynamic range. The real-time spectrum analysis capabilities of my chips make them uniquely suitable for ultra-broadband universal or software radio receivers of the future.

I show that the protein-protein and gene-protein chips that I have built are particularly suitable for simulation, parameter discovery and sensitivity analysis of in-

teraction networks in cell biology, such as signaling, metabolic, and gene regulation pathways. Importantly, the chips carry out massively parallel computations, resulting in simulation times that are independent of model complexity, i.e., $O(1)$. They also automatically model stochastic effects, which are of importance in many biological systems, but are numerically stiff and simulate slowly on digital computers. Currently, non-fundamental data-acquisition limitations show that my proof-of-concept chips simulate small-scale biochemical reaction networks at least 100 times faster than modern desktop machines. It should be possible to get $10^3\times$ to $10^6\times$ simulation speedups of genome-scale and organ-scale intracellular and extracellular biochemical reaction networks with improved versions of my chips. Such chips could be important both as analysis tools in systems biology and design tools in synthetic biology.

Thesis Supervisor: Rahul Sarpeshkar
Title: Associate Professor

বনলতা সেন

জীবনানন্দ দাশ

হাজার বছর ধরে আমি পথ হাঁটিতেছি পৃথিবীর পথে
সিংহল সমুদ্র থেকে নিশীথের অন্ধকারে মালয় সাগরে
অনেক ঘুরেছি আমি; বিম্বিসার অশোকের ধূসর জগতে
সেখানে ছিলাম আমি; আরো দূর অন্ধকারে বিদর্ভ নগরে;
আমি ক্লান্ত প্রাণ এক, চারিদিকে জীবনের সমুদ্র সফেন,
আমারে দুদণ্ড শান্তি দিয়েছিল নাটোরের বনলতা সেন।

চুল তার কবেকার অন্ধকার বিদিশার নিশা,
মুখ তার শ্রাবস্তীর কারুকার্য; অতিদূর সমুদ্রের 'পর
হাল ভেঙে যে-নাবিক হারায়েছে দিশা
সবুজ ঘাসের দেশ যখন সে চোখে দেখে দারুচিনি-দ্বীপের ভিতর
তেমনি দেখেছি তারে অন্ধকারে; বলেছে সে, 'এতদিন কোথায় ছিলেন?'
পাখির নীড়ের মতো চোখ তুলে নাটোরের বনলতা সেন।

সমস্ত দিনের শেষে শিশিরের শব্দের মতন
সন্ধ্যা আসে; ডানার রেদ্রের গন্ধ মুছে ফেলে চিল;
পৃথিবীর সব রঙ নিভে গেলে পাণ্ডুলিপি করে আয়োজন
তখন গল্পের তরে জোনাকির রঙে ঝিলমিল;
সব পাখি ঘরে আসে - সব নদী; ফুরায় এ-জীবনের সব লেনদেন;
থাকে শুধু অন্ধকার, মুখোমুখি বসিবার বনলতা সেন।

This thesis is dedicated to my parents

and

the memory of my grandfather.

Acknowledgments

Graduate school has been an extremely rewarding experience, not just because of the research, but also because of the people I've met along the way, and the institutions these people belonged to. Here is my graphical tribute to some of them.

Figure 0-1: The great ones from on high, aka my wonderful thesis committee. From left to right: Rahul Sarpeshkar, thesis supervisor, motivator extraordinaire, scientific visionary, etc. etc., Joel Dawson, Chris Shera, Bruce Tidor.

Figure 0-2: Some other great people. From left: John Wyatt, Anantha Chandrakasan, Denny Freeman, Soumitro Banerjee, Stephen Wolfram, James and Patricia Poitras. A Poitras pre-doctoral fellowship kept me going towards the end, and I am very grateful.

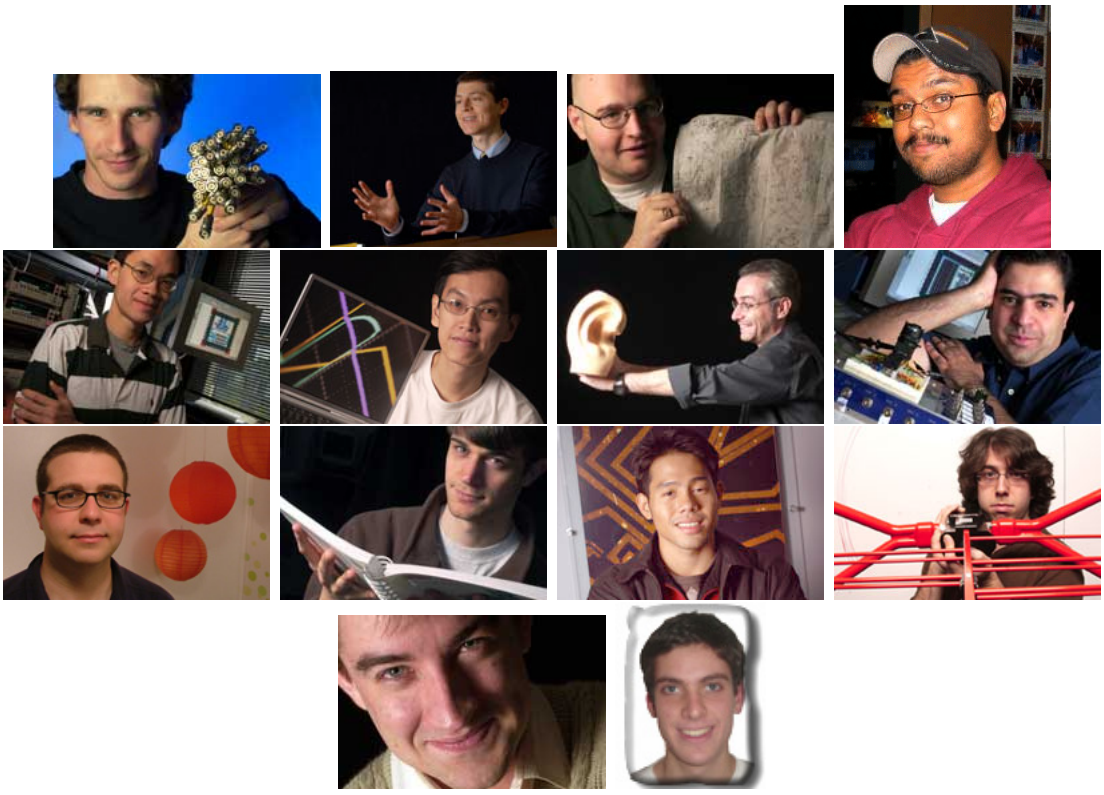


Figure 0-3: Team AVBS over the years, carrying the torch of low-power analog VLSI high. From top left: Alex, Ben, Chris, dpik, the Honorable Ji-Jon, Keng-Hoong, Lorenzo, Maziar, Micah, Baker, Pok, skarfin, Serhii, Bruno. Not pictured: Andrew, Heemin, Tim.

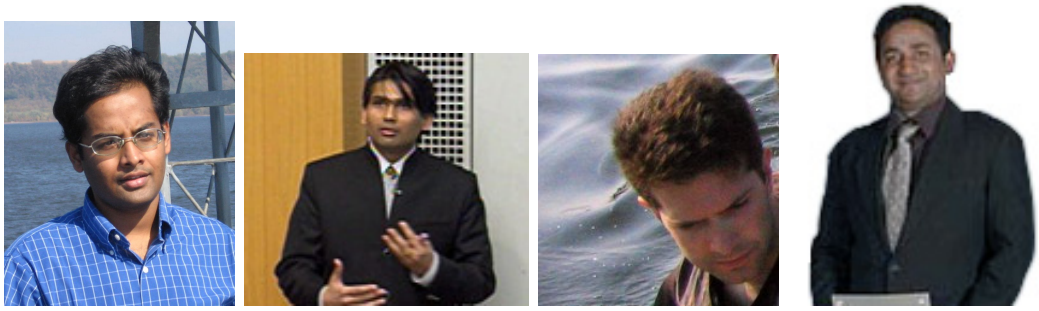


Figure 0-4: Friends, Romans and countrymen. Seriously, you guys are the best. From left: Kalmech, Luke, Phil, Ronny.

Figure 0-5: Some institutions of great worth and value (to me, at least). From top left: IIT Kgp, MIT, NKS Summer School, DBCB.

Figure 0-6: I will never understand why people give away their hard work for free over the web. However, I *am* profoundly grateful. From top left: Google, Ubuntu, Wikipedia, MiKTEX, WinEdt, XCircuit, JabRef.

Figure 0-7: For keeping the body and the spirit going when refreshments were urgently needed, honorable mentions go out to (from left): Goosebeary's, Cinderellas, PhD Comics.

Figure 0-8: And finally... the indefatigable cheering squad, aka my family. Thanks for many things (too many to count, actually). Top: a bunch, middle: some more, bottom: a final few.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The White Rabbit put on his spectacles. "Where shall I begin, please your Majesty?" he asked.
"Begin at the beginning," the King said gravely, "and go on till you come to the end: then stop."
– Lewis Carroll, Alice in Wonderland

1.1 A Very Brief Summary

This thesis shows that dynamical systems implemented as analog electronic circuits can perform useful computational tasks at high speed and low power consumption. Specifically, we discuss two examples. The first is the RF cochlea, which uses passive resistors, inductors and capacitors to implement a partial differential equation describing wave propagation in the mammalian cochlea, or inner ear, but at RF instead of audio frequencies. We show that the RF cochlea can perform fast, parallel, low-power, broadband spectral analysis.

Our second example consists of integrated circuit models of cellular biochemical networks. We show that the equations of chemical kinetics can be elegantly emulated by translinear analog circuits. We also demonstrate a chip that uses such circuits to simulate the stochastic dynamics of large chemical reaction networks, such as metabolic networks, at high speed. In addition, we demonstrate a chip that uses

analog circuits to model the processes of gene activation, transcription and translation.

Finally, we try to put our work in context by discussing the nature and meaning of computation, paying particular attention to the pernicious and unavoidable effects of noise. Throughout the text, clickable hyperlinks are shown in blue (for external links) or red (for internal links).

1.2 "Circuits Model the World"

1.2.1 Models and Modeling

A model is a simplified and idealized representation of a physical system, or part of it. Good models do not attempt to capture every feature of the system being modeled, but are nonetheless complicated enough to capture most of the emphessential features. It is up to the modeler to decude which system properties are essential and must be included in the model. As such, effective modeling is somewhat of an art, and there are no formalized procedures for determining, *a priori*, whether a certain model will be useful and/or influential. However, usually one prefers models that fit available experimental data and also make predictions that can be verified by performing more experiments. An exception to this rule is provided by the "toy model" tradition in physics. Toy models are extremely simplified, to the point where they are not expected to match experimental results. They can nonetheless often provide valuable insights into certain aspects of the physical system.

Dynamical systems can be modeled using many different formalisms, or types of abstraction. The choice of formalism depends to a large extent on the background and tastes of the modeler. However, some choices are obviously preferable to others for a given system. Bond graphs, for example, model systems in terms of power flow between components, and are commonly used for studying large heterogenous systems extending across multiple physical domains (electrical, mechanical, fluid, and so on). Electrical engineers usually prefer the language of electrical circuits,

Figure 1-1: Gabriel Kron (1901-1968).

and tend to convert mechanical and acoustic systems into electrical form. Circuits provide a convenient, high level language for describing and analyzing a wide range of dynamical systems. The utility of such abstract representations, and of abstraction in general, is not in doubt. However, the assumptions made while generating an abstract representation of a physical system should always be clearly specified.

1.2.2 History

The idea of using analog circuits to model physical systems has a long and interesting history. The field, which nowadays has sadly fallen out of fashion, was surely at its peak in the years immediately following World War II. Gabriel Kron (see Figure 1-1), for example, wrote a series of papers in the 1940's discussing circuit models of electromagnetism [146], Schrödinger's equation [148], elastic structures [147], fluid flow [149], vibration spectra of molecules [151], other ordinary and partial differential equations [150] and even nuclear reactors[1].

Figure 1-2 shows Kron's circuit model of Maxwell's equations in three dimensions. The analogy results in the following identifications:

[1]Kron's biography makes for interesting reading. Most of his career was spent as an engineer at General Electric. His many books and papers were all written at night in his spare time.

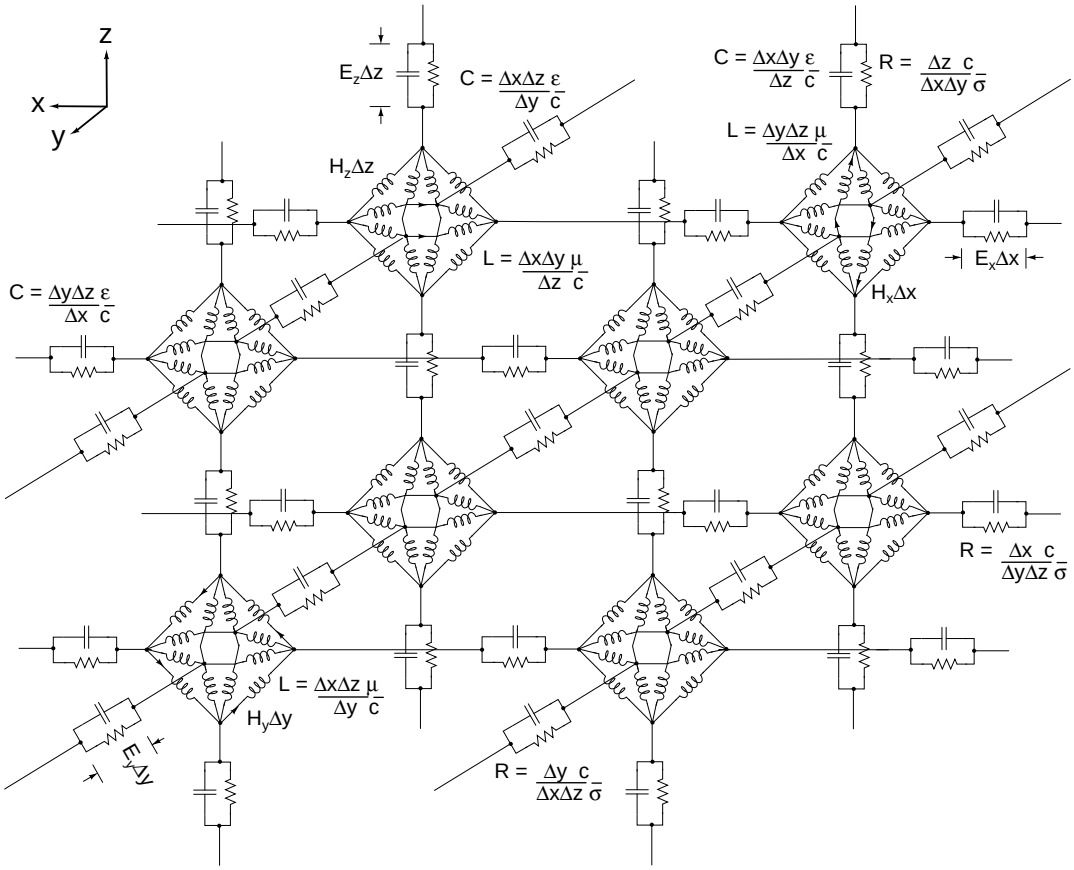


Figure 1-2: Gabriel Kron's circuit model of Maxwell's equations of electromagnetism. Figure redrawn by this author on the basis of Figure 1 in [146].

1. Currents in the inductors represent the magnetic field $\mathbf{H}$.

2. Currents in the capacitors represent displacement currents $\delta\mathbf{D}/\delta t$.

3. Currents in the resistors represent conduction currents $\mathbf{J}$.

4. Voltages across the inductors represent "magnetic displacement currents" $\delta\mathbf{B}/\delta t$.

5. Voltages across the capacitors represent the electric field $\mathbf{E}$.

6. If "magnetic conduction currents" $\mathbf{J}_m$ had existed, they would be represented as resistances placed in series with the inductors.

7. Charges on capacitors represent the charge density ρ.

Note that circuit models of Maxwell's equations that take the form of simple LC (series L, shunt C) or CL (series C, shunt L) transmission lines cannot be correct.

Figure 1-3: An analog electronic computer, the Heathkit H1 from 1956. Image source: http://www.technikum29.de/.

In the first case, the vacuum would be a short at DC, and in the second, at very high frequencies. Neither occurs, and hence a different circuit topology must be needed. Finally, electrical analogs of acoustical and mechanical systems are still widely used in academia and industry, especially when such systems have to interface with electronics (sensors and actuators).

1.2.3 Simulation

Dynamical models, however formalized, must be simulated. Today almost all simulation is performed using an electronic digital computer and various software tools. Prior to the 1950's, however, analog "AC network analyzers" were the dominant simulation technology. These machines were reconfigurable electronic analog computers originally developed for modeling electrical power networks [293], and reached great levels of sophistication. An example is shown in Figure 1-3. Even earlier analog computers had been completely mechanical, for example Vannevar Bush's famous differential analyzer [28].

Analog computers differ from digital ones in two main ways: signal representation and parallelization. Analog computers use physical variables that take on continuous

values, such as voltages, currents or velocities, to store information. This representation is compact but susceptible to corruption by noise. On the other hand, digital computers store information as two-valued, or binary numbers, This representation is verbose, but extremely robust to noise. It has been argued that the most energy-efficient computation is performed by hybrid systems, like the brain, that do not represent information in either of these extreme forms, but in some intermediate fashion [254].

Flynn's taxonomy, proposed in 1966, classifies digital computer architectures into four types. The scheme assumes that data and programs, i.e., instructions for manipulating the data, are stored in "pools", and groups computers into classes based on whether their processors access these pools in a sequential (one memory location at a time) or parallel way. Most desktop machines use the Von Neumann architecture, where data and program memory are physically shared and memory transfers occur on a single bus. Since there is no parallelization, this architecture falls into the SISD (Single Instruction, Single Data) category. The Harvard architecture is similar to the Von Neumann architecture. The only difference is that data and program memories use separate buses to access the processor. Graphics Processing Units (GPUs) are examples of Single Instruction, Multiple Data (SIMD) machines, while most modern supercomputers are Multiple Instruction, Multiple Data (MIMD) machines.

Analog computers are complex dynamical systems that accept parallel inputs and produce parallel outputs. In this sense they are multiple data machines. The richness of their dynamics, obtained by coupling together many state variables, allows them to perform massively parallel computations. As a result, they can also be considered multiple instruction machines. They are particularly suitable for finding approximate solutions to systems of differential equations. More accurate solutions can be found iteratively by a digital computer that uses this approximate solution as the starting point [43]. This hybrid strategy reduces overall solution time by utilizing the respective strengths of analog and digital computers. Performance gains are often significant, particularly for strongly nonlinear problems.

In practice, however, the performance of both analog and digital computers is

often limited by input/output bandwidth, not processing speed. Analog computers have traditionally had other problems. For instance, it is difficult to develop general-purpose error detection and correction schemes for analog dynamical systems. Several error-reduction schemes well-known to analog circuit designers, such as chopper stabilization, were in fact first developed for the operational amplifiers used within analog computers. This issue has limited the precision of analog computers.

In addition, analog computers are programmed by changing the parameters and/or connectivity of the dynamical system, while digital computers are programmed by changing the strings of bits that encode their programs. In general, the former process is a much harder to implement. As a result, analog computers today are only used for historical reasons, or in applications where digital alternatives are unavailable. Examples include special-purpose, high-speed signal processors, such as the front-ends of radio receivers.

Hybrid computers combine the characteristics of both analog and digital computaters. The synthesis can be implemented in many ways. For instance, *iterative differential analyzers* were popular in the 1960's and 70's [141, 142]. These machines consisted of a traditional analog computer whose inputs, parameters, and initial conditions could be controlled at high speed by a digital computer, such as a microprocessor. They were especially popular for Monte Carlo simulation, optimization and parameter estimation of random processes. More sophisticated interactions between the analog and digital parts of a hybrid computer can also be imagined. For example, a digital subsystem can accurately simulate the slow dynamics of a complex system, while an analog subsystem handles the fast dynamics. These system partitioning issues are fundamental to high-performance hybrid computing, which is the focus of this thesis.

1.3 The Notion of Computation

1.3.1 The Physics of Computation

What is computation? The word comes from the Latin *computare*, meaning "to count", or "to cut". A working definition of computation is the processing, i.e., transformation of information. Similarly, communication may be defined as the transfer of information from a source to a receiver. Information is a physical quantity, and the thermodynamic and information-theoretic definitions of entropy are deeply connected. In particular, all computations must obey the laws of thermodynamics [154]. The amount of information (in bits) stored by a physical system is closely related to its entropy. A system with entropy $S(E)$, where E is the mean energy of the system, has an information content I given by

$$I = \frac{S(E)}{k_B \ln(2)} \tag{1.1}$$

In the absence of noise, computation and communication can be performed at finite speed and zero error with zero energy dissipation [75, 153]. Errors will occur in the presence of noise, but we can still compute at finite speed without dissipating energy. Only the erasure or removal of information via irreversible operations requires energy (this is known as Landauer's principle). Each lost bit results in $k_B T \ln(2)$ of dissipation, i.e., generation of heat, where T is the temperature of the system. Landauer's principle provides a simple explanation for the impossibility of creating Maxwell's demon. Such a demon could violate the second law of thermodynamics by sorting gas molecules based, say, on their velocities. However, each measurement of a molecule's velocity adds information to the demon, who must retain it in memory. Eventually the demon runs out of memory capacity and has to delete old measurements. Each deletion increases the entropy of the whole system (gas and demon) by an $k_B \ln(2)$, exactly the amount by which the entropy of the gas decreased when each molecule was sorted. Therefore the second law is not violated.

An important consequence of Landauer's principle is that physically reversible,

isentropic (dissipation-free) computation is possible by using only *logically reversible* operations [15]. In general, reversible operations are those that do not contract phase space, i.e., reduce the number of degrees of freedom of the system. Thus they must have the same number of inputs and output bits. For example, the one-input one-output Boolean NOT function is reversible, but two-input one-output functions like NAND and NOR are not. Certain three-input three-output logic gates, such as the Toffoli and Fredkin gates, have been shown to be both reversible and universal[2].

The input phase space of a reversible computer, unlike an irreversible computer, is not compressed into the single desired output as the computation proceeds. Thus, unwanted information accumulates at every step of the computation. Such unwanted information is usually known as *garbage*. It cannot simply be deleted at the end of the computation, since deletion of information dissipates energy. As a result, handling garbage is a major challenge in implementing any reversible computer. In 1973, Bennett proposed an ingenious solution to the garbage problem [14]. In his model, computation proceeds in two reversible steps. The forward step generates the desired output, plus many unwanted outputs. The desired output is copied (in a lossless way) for later use, and then the computer is run backwards. Since the system is logically reversible, the reverse step "cleans up" all the unwanted outputs generated by the forward step, finally recreating the input. In a seminal paper published in 1982 [75], Fredkin and Toffoli described ways in which the scheme described by Bennett could be implemented using reversible logic gates.

Any practical computer needs additional error correcting mechanisms in order to correct errors and operate reliably in the presence of noise. The most fundamental source of noise is random thermal motion, or heat. Correcting errors involves deleting incorrect bits, and hence requires energy to be dissipated. As a result, the temperature of any computer (reversible or not) must be kept as low as possible to minimize dissipation. In other words, the computer must be weakly coupled to the surrounding environment. However, it is difficult to provide inputs to, and read outputs from,

[2] A universal gate, such as the NAND, is functionally complete. In other words, circuits consisting only of the gate in question can realize any Boolean function

weakly-coupled systems. This contradiction places an important constraint on the design of energy-efficient computers.

Several physical realizations of reversible computers have been proposed, and they fall into three main classes: ballistic, Brownian or adiabatic [15], and quantum. Ballistic computers such as Fredkin and Toffoli's billiard-ball models [75], use the trajectories of an isolated, conservative dynamical system to compute. Unfortunately, such systems are usually non-integrable and exhibit sensitive dependence on initial conditions. As a result, the trajectories rapidly become chaotic. Therefore such computers are difficult to implement without continuous error correction, which dissipates energy. Brownian computers operate at the other extreme: they assume a computer that is close to thermal equilibrium with its surroundings. Molecular machines in biology, such as RNA polymerase, operate in this fashion. A Brownian computer essentially executes a random walk in the computational space, i.e., each computational step is equally likely to proceed forward or backward. In order to drive the overall computation forward, a small constant force is superposed on top of the random walk. This force produces a constant drift velocity, and therefore results in energy dissipation. However, the dissipation can be made asymptotically zero (the so-called adiabatic limit) by reducing the drift velocity and computing slowly.

Quantum computers exploit the physical phenomena of quantum superposition (coherence) and entanglement for performing computation. The resultant computational paradigm exhibits several novel features that are qualitatively different from the classical paradigms that we have described so far. In fact, quantum computers are generalizations of classical computers. Several algorithms that run only on quantum computers, and offer significant speedups over the best-known classical equivalents, have been discovered recently. Shor's integer factorization algorithm is a famous example. All quantum operations are unitary matrices, i.e., they are equivalent to rotations of the state space and preserve the Euclidean, or L_2 norm. Thus, all quantum computers are reversible and free of dissipation. However, in practice they are very susceptible to loss of quantum coherence due to coupling with their surrounding environment. The resultant errors can be corrected, but require the deletion of

incorrect quantum states. This process dissipates energy, just as in the classical case.

Available energy fundamentally limits computational speed, because the energy-time inequality of quantum mechanics states that the time Δt it takes for a system to evolve from one state to another that is orthogonal (distinguishable) to it is inversely proportional to the mean energy E of the system, i.e.

$$E \geq \frac{\pi\hbar}{2\Delta t} \tag{1.2}$$

Since evolution between distinguishable states is the basic definition of a computational operation, the rate at which such operations can be performed is inversely proportional to E [167]. Note that ideally no energy is dissipated during the computation and the average energy of the closed system remains fixed at E. Another fundamental physical constraint on computation is provided by entropy, which limits memory capacity, i.e. the amount of information that can be stored by a computer and the rate at which it can be processed.

1.3.2 The Theory of Computation

The Church-Turing thesis states that any function that is effectively calculable is computable. By "effectively calculable" we mean that it can be calculated using a mechanical procedure, i.e. an algorithm, working on an input of finite length and terminating in a finite number of steps. By "computable" we mean that it can be calculated by any one of a number of computational processes that are known to be equivalent to one another, such as Turing machines, λ-calculus, μ-recursive functions, unrestricted grammars, recursively enumerated languages, common programming languages, counter machines, register machines, cellular automata (CAs) and so on. Intuitively, every function that we would normally regard as being computable can be computed by some Turing machine with an infinitely long tape, i.e., memory. Turing, in his original paper in 1936 [290], went on to show that it was possible to create an universal Turing machine that could emulate any other Turing machine, i.e., compute the same function as that machine. Such a machine is known as an

universal computer.

Note that the Church-Turing thesis is not a theorem, since the notion of "effectively calculable" is not formally defined. However, all attempts to find physically realizable computational processes that compute a larger class of functions than that computable by Turing machines have failed[3]. In fact, an ever-growing list of processes have been shown to compute exactly the same class of functions as Turing machines, and are therefore referred to as being Turing-equivalent. As a result, the thesis is now usually considered to be an axiom, i.e., assumed to be true unless stated otherwise.

It is also interesting to note that when Turing uses the terms "computer", "computable" and "computation", he is not referring to machines but human calculators. According to Turing, an effective procedure can also be defined as one that can be carried out by a unaided human following fixed rules. For example, he writes that [291]:

> A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine.

We should clarify what we mean by a computational process. Such processes manipulate information: they accept inputs and transform them into outputs by applying sets of rules that are called programs. The set of allowable programs defines the process. Equivalently, the process computes a function of its input. Proving a mathematical theorem is also equivalent to performing a computation. The input is the initial statement, the output the final statement and the program is the set of transformations that produces the output from the input. In order for the proof to be valid, each program statement (transformation) must be allowable by the mathematical axiom system.

It can be easily shown, via a version of Cantor's famous diagonal argument, that a necessary but not sufficient condition for any computational process to be universal is that, for certain inputs, the computation does not terminate, i.e., halt. Universal computation therefore requires a set of axioms, i.e., allowable rules, that is rich enough

[3]Such hypothetical processes are known as hypercomputers. For example, so called real computers that operate on infinite-precision real numbers would be hypercomputers.

to allow programs to be written that never terminate. Equivalently, the function being computed must be undefined for certain inputs.

The *halting problem* asks whether there exists any procedure that can predict whether *any* arbitrary program-input pair, when run on an universal computer (assumed to possess unlimited memory and time), results in a computation that halts. No such procedure is known today. Unsolvability of the halting problem is equivalent to a form of Gödel's theorem [38], and is also known as irreducibility. The principle of computational irreducibility [317] extends the notion of irreducibility to assert that the *best possible procedure* to find out whether a program-input pair will halt is to actually carry out the computation by running the program.

A given universal computer can emulate any other universal computer since they both compute exactly the same class of functions. However, in general encoding and decoding functions are required to encode a given problem as the input to an universal computer, and to decode its outputs. Such functions must always halt (and hence cannot be universal themselves), but are otherwise arbitrary. In fact, we say that a function f *emulates* a function g if and only if computable encoding and decoding functions d and e exist such that

$$d[f[e[x]]] = g[x] \tag{1.3}$$

whenever $g[x]$ halts, x being the input[4]. In practice certain types of universal computer are preferable for solving a given problem because of their desirable encoding and decoding properties for that problem. For example, many mathematical axiom systems are computationally universal, just like common programming languages, but in most cases it is difficult to find an encoding or translation function that allows the former to act as the latter [317]. Similarly, certain problems, such as lattice gas simulations, are well suited for CAs because they can easily exploit the parallel nature of CA computation. However, it would be interesting to investigate the intuitively reasonable idea that, *in general*, encoding and decoding functions tend to become

[4]I am indebted to Matthew Szudzik for this definition.

simpler as the internal complexity of the universal computer increases.

Explorations of the so-called computational universe reveal that universal computation is quite common. Universality has been found in many simple formal systems, such as cellular automata, cyclic tag systems, multi-way systems [317] and systems of ordinary differential equations [23]. Several theories and models of physics are known to be computationally universal, such as conformal field theories and quantum electrodynamics [167]. Many axiom systems of mathematics are universal, such as the Robinson and Peano axioms of arithmetic and set theory. Wolfram's principle of computational equivalence [317] conjectures that most systems that are not behaving in an "obviously simple" way are performing computations at an equivalent level of sophistication, i.e. are universal in the Turing machine sense and can emulate each other. On the other hand, it is also important to note that being universal is neither necessary nor sufficient for performing useful computations. Many useful but non-universal computational systems exist, such as finite state machines, and many universal systems are not useful.

1.3.3 Analog and Unconventional Computation

We shall need the notion of Lipschitz continuity to discuss computation in continuous and hybrid dynamical systems. Lipschitz continuity is a smoothness condition for functions that is stronger than the regular definition of continuity. Intuitively, a function that is Lipschitz continuous is limited in how fast it can change. Formally, a function $f(x)$ defined on a subset D of the real numbers is Lipschitz continuous if there exists a constant $K \geq 0$ such that $\forall\, (x_1, x_2) \in D$, we have

$$|f\,(x_1) - f\,(x_2)| \leq K\,|x_1 - x_2| \tag{1.4}$$

It has been shown that smooth, continuous-time dynamical systems can be used to implement logical operations and simulate arbitrary finite state machines [25][5]. Even more interestingly, systems of smooth ordinary differential equations (ODEs)

[5]Finite state machines (FSMs) are also known as finite automata. They accept languages generated by regular grammars and are not computationally universal.

in $2N + 1$ variables can simulate arbitrary Turing machines in N dimensions [23]. Therefore there are systems of ODE's in three variables that are computationally universal. It has also been shown that the equations that describe chemical kinetics in a homogenous material are computationally universal [181].

The computational capabilities of several well-known types of hybrid system have also been studied [23]. Such systems combine continuous-time dynamics (usually ODE's) with discrete-time dynamics (usually FSMs). All the systems studied can simulate arbitrary Turing machines in N dimensions using smooth ODE's in $2N$ variables and a precise clock, generated by the discrete dynamics, as input. The hybrid systems therefore require ODE's in one less variable than purely continuous-time systems in order to simulate the same Turing machines ($2N$ versus $2N + 1$). Intuitively, this difference arises because one of the variables in the purely continuous-time system plays the role of an imprecise clock, while in the hybrid systems the discrete-time dynamics can generate precise clocks. One consequence of the imprecise nature of the clocks generated by Lipschitz-continuous systems is that they cannot solve certain problems that can be solved by hybrid systems, such as the famous asynchronous arbiter problem.

Particle-like behavior has been found in many cellular automata (CA), particularly parity-rule filter automata (PRFA). Particle machines, or general models of computation using collisions of particles, are generalizations of phenomena observed in PRFAs, and have been shown to be universal. The collisions must transfer state information between the colliding particles, i.e., they cannot be elastic (also known as oblivious). Particle machines can also be realized using non-oblivious collisions between soliton solutions generated by certain nonlinear partial differential equations (PDEs). Such systems are known as soliton machines [130, 131][6]. Soliton machines with periodic backgrounds are known to be universal. Soliton machines with quiescent backgrounds have been shown to have at least the computational power of Turing machines with finite tape lengths.

[6]The mathematics of CA-based and PDE-based solitons appears to be completely different, yet, intriguingly, they share many properties.

Some PDE's and ODE's, such as the wave equation in three dimensions (but not the Laplace and heat equations), can transform computable initial conditions into continuous but non-computable solutions [227]. However, certain derivatives must be non-continuous in order for such transformations to occur, i.e., the solutions in question are "weak" or "generalized" solutions that contain shocks, cusps or other non-smooth behavior. Such solutions do not satisfy Lipschitz conditions. Intuitively, this result does not appear too surprising. Partial differential equations perform operations on arbitrary real numbers and, in an ideal world with infinite precision, would appear to be good candidates for implementing real computers, i.e., hypercomputers. The computational performance of PDE's in the presence of finite numerical precision has, as far as I know, yet to be analyzed.

The field of "unconventional" computing appears to be undergoing somewhat of a renaissance. The field even has its own journal, the International Journal of Unconventional Computing. For example, it has been shown that a simple slime mold can find the optimal (lowest cost) path through a maze [211]. Universal computation has been demonstrated using both plasmodia and reaction-diffusion systems such as the Belousov-Zhabotinsky (BZ) reaction [2]. In both cases the occurrence of localized traveling structures is necessary for universal computation to occur in homogenous media. Just like in particle computing models, such structures both communicate information (by traveling) and process it (by colliding with each other). Such particle-like structures only emerge on the so called "edge of chaos", i.e. close to the point where the system undergoes a phase transition between complete order and complete randomness [129][7]. Intuitively, particle-like excitations occur near phase transitions because of the presence of long-range correlations (that decay as power laws) in the medium. In this regime, the medium has long memory in both space and time, which is desirable for universal computation to occur.

Another interesting approach to unconventional computation that has recently been reported is known as *evolution in materio* [108]. In this method an evolutionary

[7]The phrase "edge of chaos" was originally coined by Christopher Langton, perhaps best known for founding the field of Artificial Life, to describe the behavior observed in many CAs at a specific value of his λ parameter.

search algorithm is used to program a bulk material, such as a liquid crystal, array of quantum dots or crystal lattice to perform a specified computation. Eventually the material programs itself by learning how to modify its own local (nearest-neighbor) interactions. The technique is conceptually similar to previous work where the rules of a CA or CA-like system rule were evolved in order to perform a particular computational function [45]. Finally, it has been shown that logical and arithmetic operations can also be performed by chaotic dynamical systems [209].

1.3.4 Computational Efficiency

Computational efficiency is an oft-neglected but very practical factor when considering which universal computer to use for a given problem. For example, the fact that Conway's Game of Life is universal does not mean that it is suitable for solving typical computational problems. The computational complexity and efficiency of analog computation has been studied in [297]. The authors formulate a "strong Church thesis", namely that any analog computer can be simulated *efficiently*, i.e., in polynomial time, by a digital computer. They find that the strong Church thesis is true for a subset of analog computers, namely "well-behaved" ordinary differential equations that obey a uniform Lipschitz condition, i.e., the solution and its derivatives are Lipschitz continuous. The extension of this type of complexity analysis to hybrid systems would be a logical next step, but I have been unable to find published work that specifically analyzes this problem.

A common metric used to quantify the efficiency of a computation, particularly in digital systems, is the product of energy consumed, E and the delay t_d, i.e., the time taken for the computation to complete. This metric, known as $ET1$, can also be written in the form E/R, where $R = 1/t_d$ is the throughput (computation rate). When written in this form it becomes clear that the important quantity is the energy consumed per operation while maintaining a given throughput. In general, slow-and-parallel computational architectures are more efficient than fast-and-serial ones: the throughput of each parallel path decreases linearly with the degree of parallelization, while the energy per operation drops at a faster rate, often quadratically [39].

It has been argued that $E \times t_d^2$, known as $ET2$, is a better efficiency metric for comparing various algorithms [192] because simply scaling the supply voltage can improve the $ET1$ metric. However, the $ET2$ metric was derived under the assumption that transistors behaved as square-law devices. In general, a metric consisting of E times some monotonically increasing function of t_d will prove to be "optimal". Here optimality means that the metric changes only with the computational algorithm and not with external (non-algorithmic) variables like supply voltage. Finally, there are fundamental reasons for believing that asynchronous architectures are more energy-efficient than synchronous ones; as a result, systems-on-chip of the future are expected to be mainly asynchronous [193].

Digital systems restore every signal value to one of two attractor states, i.e., 0 or 1. In other words, they ensure robustness by performing error correction, which requires energy, after every computation. Analog systems, on the other hand, do not implicitly perform error correction. The lack of implicit error correction makes robust analog systems harder to design. Feedback loops for learning errors and removing them must be explicitly added by the designer whenever necessary. In addition, the energy efficiency of the two strategies scale differently with computational *precision*, i.e., the signal to noise ratio or effective number of bits [254].

Digital systems are less energy-efficient at low precision, since many of their ubiquitous local error corrections can be removed without affecting the accuracy of the computation. On the other hand, the overhead associated with explicit analog error correction is a strongly increasing function of precision, which makes analog systems less energy-efficient at high precision. Thus, the optimal computational strategy depends on the required level of precision. The crossover point between the two strategies depends on the *technology*, i.e., physical characteristics of the medium used for performing the computations.

Intuitively, it seems clear that a hybrid strategy that combines aspects of both analog and digital systems might be able to obtain a better trade-off between energy efficiency and precision. For inspiration, one can turn to biology. For example, computations performed by networks of neurons are both precise and energy-efficient.

A variety of hybrid schemes, many of them biologically inspired, have indeed been suggested over the years. However, it has proven difficult to develop schemes that provide a good trade-off between efficiency and precision across a wide range of computational tasks and technologies. In fact, it appears that, in general, efficiency and precision are highly inter-dependent joint functions of strategy, task and technology. The optimum strategy is therefore strongly dependent on both task and technology.

Efficient computations minimize the consumption of resources, such as energy, time and hardware. Such operations must be well-matched to the degrees of freedom of the system performing the computation. In other words, the encoding and decoding functions for mapping computations defined in (1.3) should be as simple as possible. Digital circuits use only one of the many degrees of freedom provided by a transistor, namely the ability to turn it on or off like a switch. Because analog circuits are not restricted in this way, they form a superset of digital circuits.

Relaxing the digital constraint allows some algorithms to be implemented in a significantly more efficient way. Translinear circuits, for example, use the exponential I-V curve of bipolar junction transistors (BJTs) and subthreshold field-effect transistors (FETs) to perform efficient, high-dynamic-range signal processing [91]. Recently, a principled way has been found for representing message-passing algorithms using continuous-time analog circuits [298, 299]. Message-passing algorithms, such as belief propagation (BP), compute marginal probability distributions of functions in factor graphs or other graphical models. They are widely used for solving complex statistical inference problems in a range of fields. This novel framework for performing statistical signal processing promises to be more efficient, i.e., higher speed and lower power, than conventional digital circuit techniques.

Another way to increase computational efficiency is by specializing the computer architecture so that it is optimized for running a restricted set of algorithms. In general, special-purpose computers will always consume less power than general-purpose machines performing similar computational tasks. Only in certain cases, however, will the performance benefits be large enough to justify the inevitable increases in cost and development time. Historically, special-purpose machines have been numerous,

particularly in academic settings, but not very successful. Examples include Margolus and Toffoli's cellular automata machines (CAMs), which were optimized for lattice-gas simulations of fluid mechanics [190], Hillis' connection machines (CMs), optimized for artificial intelligence (AI) applications[8], and several machines optimized for calculating Ising and spin-glass models, lattice quantum chromodynamics (QCD) and gravitational N-body problems [183].

One reason for the relative failure of special-purpose computers is insufficient flexibility for running new and improved algorithms, but a more important reason is Moore's "law". As long as faster general-purpose computers were continuously becoming available there was insufficient motivation for developing the custom hardware needed for a special-purpose computer. General-purpose microprocessors, however, do not utilize the full computational power of modern VLSI. The input/output (I/O) bandwidth available for communication to external memory has not been increasing as fast as the raw computational power of the processor. Fundamentally, this property is an effect of the two-dimensional nature of integrated circuits. I/O bandwidth scales with the die or package perimeter, while computational power scales with the number of transistors, which, for a given area, increases as the square of the feature size. The net result is that the computational speed of modern microprocessors is limited by I/O bandwidth. Performance can be improved by integrating multiple processing cores on the same die and allowing them to share data via an on-chip cache, thus reducing off-chip memory access. In fact, microprocessor designers are now integrating two, four or more cores and large amounts of level 1 and 2 cache into their chips. However, it has proven very difficult to write code for general purpose applications that efficiently utilizes these multiple cores.

Among the most successful special-purpose computers have been the Japanese GRAPE (GRAvity PipEline) machines [183]. These computers were originally designed for solving many-body problems in astrophysics, but variants of the same architecture have been developed for protein-folding and other molecular dynamics

[8]The story of Hillis and his company, Thinking Machines Corporation, serves as a particularly poignant cautionary tale for would-be special-purpose computer enthusiasts. As one author puts it, "history is littered with the corpses of specialized machines" [269].

(MD) problems. Solving gravitational interactions is computationally intensive because gravity is a long-range force[9]. For such computations a special-purpose chip can provide substantial speedup over a general-purpose microprocessor, because, unlike in most other applications, the limiting factor is not memory bandwidth but the number of floating-point operations. The other reason for the relative success of the GRAPE systems has been their philosophy of dividing the machine into two main parts: a general-purpose front-end and a custom back-end. The former, made of off-the-shelf microprocessors, handles everything except the computationally intensive gravitational force calculations. This approach reduces cost and development time over a fully-custom solution.

Recent years have seen renewed interest in special-purpose computers. There are two main reasons. Firstly, the power consumption of today's general-purpose supercomputers (about 4nW per floating-point operation per second) has increased to the point where it has become a limiting factor for cost and reliability. Secondly, the demise of Moore's law appears imminent. In fact, several recent supercomputers have been optimized for performing specific types of scientific simulations at low power and low cost. Examples include the MDGRAPE machines, optimized for many-body simulations in materials science, astrophysics and molecular dynamics [283], and Anton, a dedicated MD machine [269]. In addition, the graphics processing units (GPUs) found inside most personal computers and GPU/CPU hybrids like IBM's Cell processor are arguably special-purpose architectures optimized for data-intensive applications. Field-programmable gate arrays (FPGA's) provide another example of this paradigm. The hardware configuration of an FPGA is defined by a hardware description language, allowing its architecture to be optimized for running specific algorithms.

[9]Cellular automata, because of the purely local nature of their rules, are unsuitable for systems where long-range interactions such as gravity or electromagnetism are dominant.

1.3.5 PDE-Based Computation

Partial differential equations (PDE's) are mathematical objects that arise in many different areas of science and engineering. They may be viewed as continuous generalizations of cellular automata in both amplitude and time. Differential equations consist of purely local rules, defined on smooth manifolds, that determine the rate of change of a continuous variable with respect to the independent variables that form the manifold. In ordinary differential equations (ODE's) there is only one independent variable, while in PDE's there are more than one. In mechanics, ODE's describe spatially-localized *particles*, while PDE's describe spatially-diffuse *fields*.

In its most general form a PDE expresses a constraint on the values of a function u and its derivatives with respect to the independent variables x_1, x_2 and so on:

$$F\left(u, u_{x_1}, u_{x_2}, ...\right) = 0 \tag{1.5}$$

where $F()$ is an arbitrary function. For most PDEs that are important in science and engineering, F can be expanded into a series in u and its derivatives:

$$\sum_{i=1}^{N} c_i u_{x_1 x_2 ...} = 0 \tag{1.6}$$

The PDE is *linear* if the sum on the left can be written as a linear differential operator, operating on u. Linearity implies that the coefficients c_i are either constants or functions only of the independent variables x_j. A subset of nonlinear PDEs consist of equations where the c_i are functions of x_j and u, but not of the derivatives of u. Such equations are referred to as being *quasilinear*. The *order* of a PDE is defined as the order of the highest derivative that occurs within it.

Since superposition does not apply to nonlinear systems, transform-based methods, which are important tools for linear PDEs, cannot be used for solving nonlinear PDEs. Many nonlinear PDEs exhibit *finite-time blowup*, i.e. solutions with unbounded norm (singularities) that develop in a finite amount of time from smooth initial conditions. Such behavior is amplitude-dependent, i.e., the equation is typi-

cally conditionally stable. As blow-up occurs, gradients increase in both space and time. Other PDEs, such as the Navier-Stokes equations that describe fluids, exhibit the phenomenon of turbulence. In turbulence, the velocity field becomes random in space and time and eddies develop that pump kinetic energy into smaller and smaller length scales. The eddies eventually become small enough for viscosity to dissipate away their energy, preserving overall stability.

Symmetries and Conserved Quantities

Noether's theorem is a fundamental result which states that every differentiable symmetry of the action of a physical system corresponds to a specific conservation law. A symmetry of a physical system is any property of the system that is invariant after a transformation, and the action is the integral of a Lagrangian function. Important examples of conservation laws are those of linear momentum, angular momentum and energy, which correspond to Lagrangian functions that are symmetric to rotations and continuous translations in space and time, respectively. Noether's theorem is important for PDE-based computation: systems that conserve energy will not exhibit finite-time blowup, so they are unconditionally stable and can perform useful computations on a large range of boundary conditions.

For example, consider conservation of electrical charge, which forms the basis of Kirchoff's Current Law (KCL). This law arises from a symmetry property, as we would expect based on Noether's theorem. In this case, the underlying symmetry is the fact that *differences* in potential, rather than absolute potentials, determine circuit behavior. In other words, we can designate an arbitrary node in a circuit as "ground", and reference all other potentials with respect to it.

Fundamentally, symmetry to absolute potentials depends on a property of Maxwell's equations known as *gauge invariance*. We are free to pick any "gauge" Λ in defining the vector and scalar potentials (denoted by $\mathbf{A}$ and ϕ, respectively) as long as they leave the physically-observable electric and magnetic fields themselves unchanged. This condition is automatically satisfied if we always choose new values of $\mathbf{A}$ and ϕ such that

$$\begin{aligned} \mathbf{A}' &= \mathbf{A} + \nabla\Lambda \\ \phi' &= \phi - \frac{1}{c}\frac{\partial\Lambda}{\partial t} \end{aligned} \tag{1.7}$$

where c is the speed of light. An important limitation of Noether's theorem is that it can only be applied to systems where a variational principle can be defined. Such systems, also referred to as variational systems, have to be of even order, contain an equal number of dependent variables and equations and have no dissipation *as written*. Appropriate mathematical manipulation can sometimes convert a given system of differential equations from non-variational to variational form, and vice-versa. Since such manipulations can include co-ordinate transformations, Noether's theorem becomes coordinate-dependent. This situation is undesirable since conservation laws are not coordinate-dependent.

Many systems, such as dissipative systems involving diffusion terms, do not satisfy the criteria needed to apply Noether's theorem, i.e. cannot be written in variational form. In such cases direct construction methods [19] can be used to find symmetries (which often turn out to be non-local) and corresponding conservation laws.

PDE Properties

The general existence-uniqueness theorem for ODE's (known as the Picard-Lindelöf or Cauchy-Lipschitz theorem) states that a unique solution $y(x)$ to the initial value problem $dy/dx = f(x, y)$, $y(x_0) = y_0$ exists in a non-zero neighborhood around x_0 provided f is bounded, Lipschitz continuous in y, and continuous in t within the neighbourhood. No theorem of comparable generality exists for PDEs. However, existence and uniqueness theorems have been proven for particular classes of PDEs and boundary conditions. For example, PDE's that are analytic in the unknown solution and its derivatives have unique solutions when subjected to Cauchy boundary conditions, i.e., when both the solution at the boundary and its gradient normal to the boundary are specified.

Linear second-order PDEs can be classified into hyperbolic, parabolic and elliptic equations. There are fundamental differences between hyperbolic and elliptic PDEs. In hyperbolic (or parabolic) equations, boundary and initial condition data propagate into the computational domain at finite velocities along characteristic curves. Such systems are also known as evolution equations since they describe processes that change (evolve) with time. As a general principle, hyperbolic PDEs can be written in simplified form along their characteristic curves, which may be viewed as the natural co-ordinates for the equation [168]. Intuitively, elliptic equations may be viewed as the steady-state problems that remain when all transients in an evolutionary system have died away. Elliptic equations do not support signal propagation at finite speeds, and only boundary (not initial) conditions are relevant for solving them. The lack of a finite signal-propagation velocity means that elliptic equations require global solutions, which makes them more difficult to solve, in general, than hyperbolic or parabolic problems where localized solutions can be found. The wave, heat conduction and Poisson's equations are the prototypical examples of hyperbolic, parabolic and elliptic PDEs, respectively.

Perharps surprisingly, it can be shown that many differential equations possess solutions that are themselves not continuously differentiable. Such solutions are known as weak or generalized solutions. The key is to rewrite the DE in integral form, perhaps using Green's functions, so that no derivatives of the solution appear in the equation. Such integral equations often express conservation laws of a physical quantity such as electrical charge or mass. Solutions of these equations that are not continuously differentiable, i.e., are multi-valued at certain points, are called weak solutions. However, weak solutions have integrals that satisfy the integral equation everywhere. Many PDEs that model real-world phenomena, such as shock waves and tidal bores, can only be shown to possess weak solutions. Discontinuities in weak solutions can be classified into weak discontinuities, where the solution itself is continuous, but one or more of its derivatives are not, and strong discontinuities (shocks), where the solution is discontinuous. Weak discontinuities of quasilinear equations propagate along characteristic curves, but the same is not true for shocks.

PDE Simulations

We have numerically simulated the behavior of many nonlinear hyperbolic and parabolic PDEs (wave and reaction-diffusion type equations). We limited ourselves to PDE's with two independent dimensions (one space and one time) to keep the search space tractable. In particular, wave equations with additive polynomial nonlinearities, i.e., equations of the form

$$u_{tt} = u_{xx} + \sum_{n=0}^{N} c_n u^n \tag{1.8}$$

were studied exhaustively. Cauchy boundary conditions were imposed at $t = 0$ (beginning of the simulation). The boundary function consisted of one or more spatially-localized Gaussian pulses of various amplitudes. Many, if not most, equations exhibit finite-time blowup. We found that the behavior of the other, unconditionally stable, equations typically falls into one of a few well-defined classes. As a result, the vast space of possible nonlinear PDE's can often be effectively explored by studying a few prototypical equations[10].

In certain rarecases, requirements for unconditional stability can be found analytically, For example, it can be shown [317] that the class of wave equations given by

$$u_{tt} = u_{xx} + f(u) \tag{1.9}$$

possess time symmetry. Thus, by Noether's theorem, they conserve the Hamiltonian, or total energy function, given by

$$H = \int_{-\infty}^{\infty} \left(\left[u_x^2 + u_t^2 \right] / 2 + V(u) \right) dx \tag{1.10}$$

where the potential energy $V(u)$ is given by

[10]Most of this work was carried out at the 2008 New Kind of Science (NKS) Summer School, organized by Wolfram Research.

$$V(u) = -\int_0^u f(y)dy \tag{1.11}$$

We see that if $V(u)$ is bounded from below, i.e., always greater than a finite value V_{min}, u_t and u_x must be bounded from above. Thus the PDE will be unconditionally stable. For the subset of equations given by (1.8), it is easy to show that, for unconditional stability, the highest non-zero coefficient c_n must occur for an odd value of n and be negative.

For computational purposes complex spatio-temporal behavior, such as spatially or temporally-localized solutions, shocks, collisions and controllable propagation trajectories, are of interest. We have observed that such complex behavior is often associated with interactions between pulses or wave trains and periodic, spatially uniform "bacckground" solutions. This behavior reminds us of the ability of periodic backgrounds to increase the computational power of soliton machines.

Electrical Models

We would like to design PDEs that carry out specific signal processing functions, such as gain control and producing constant phase shifts. Circuit analogies can be used as starting points to model these systems, thereby reducing the equation structure and parameter value search space. In addition, the equations can be physically implemented as electrical circuits. PDE's in one dimension and one dependent variable can be implemented using transmission lines with series and shunt elements composed of inductors, capacitors and resistors. We allow the values of these two-terminal circuit elements to depend on space and time. In addition, they can also depend on the amplitude of the voltage across them or the current through them, causing nonlinear behavior. Systems with M dependent variables can be implemented using M coupled transmission lines.

1.3.6 Computations in Biology

Biological systems are incredibly diverse. As a result, it is difficult to develop general theories about the types of computations they perform. However, here are some interesting, but isolated facts about biological computation[11]:

- The universe has sometimes been viewed as being a giant computer [317, 327][12]. However, nobody knows what program such an universal computer may be running (but, see [263]). Similarly, biological systems can be viewed as always performing computations, but we often don't know what program is running and what the inputs and outputs are.

- Much of the complexity of biological computing arises from the need to create internal representations of certain aspects of the external world. Such representations are needed for organisms to localize themselves, organize sensory inputs and anticipate environmental changes. Even creatures as simple as bacteria create such representations [282].

- Biological computations frequently use successive approximation algorithms where time is treated as a cost function. In other words, in order to increase the accuracy of a biological computation we must wait. Approximate results are available quickly, and the accuracy of the computation increases continuously with time.

- Biological computing is constrained spatially by transport requirements. For example, cells transport synthesized proteins via diffusion. This fact provides a possible explanation for the fact that the sizes of cells across a multitude of organisms are surprisingly constant [300].

- Biological computing is hybrid. Discrete, stabilized states are used when robustness to noise and memory are required (digital-like), while analog computations

[11] This section is based in part on conversations with Stephen Wolfram, Toshiyuki Nakagaki and Benjamin Rapoport

[12] Such views were first expressed by computer pioneer Konrad Zuse in the late 1960's, and are often referred to as "digital physics". Zuse's papers on the subject may be downloaded from http://www.idsia.ch/~juergen/digitalphysics.html.

are used for sensing weak signals or in other situations where high sensitivity is required [4].

- Biological computation occurs with noisy, heterogenous components but is extremely energy-efficient. The human brain consumes only about 15W of power, and a molecular machine like RNA polymerase consumes about $20k_BT$ of energy to synthesize one nucleotide at a raw error rate of $< 10^{-4}$.

1.3.7 Collective Analog Computation

We may formally define computational efficiency as the number of operations performed per unit of dissipated energy. Fundamentally, efficiency depends on three factors: speed, a suitable measure of precision (such as the number of bits or signal-to-noise ratio) and power consumption. The relationships between these quantities can be summarized by the following equation, which should only be treated as a dimensional analysis:

$$\text{Computational efficiency} = \frac{\text{Speed} \times \text{Precision}}{\text{Power consumption}} \tag{1.12}$$

We should consider the same *task*, implemented in the same *technology*, in order to compare the efficiency of different computational strategies. For a given task, we can improve efficiency by increasing speed or precision while dissipating the same amount of power. In order to increase speed we can exploit the physics of the technology, i.e., the computational medium. For example, transistors are not simply digital switches, but contain many additional degrees of freedom which can be exploited by analog cirucits. In order to improve precision we can use clever circuit design techniques such as feedback, learning, and *collective analog computation*.

The phrase 'collective analog computation' should be explained in greater detail since it is part of the title of this thesis. Common electrical variables used to represent signals include voltages at nodes and currents through wires. Imagine that we want to perform some computation on such signals with a certain amount of precision. The question is one of *representation*. Should we represent input and output signals

with single physical variables, or distribute them over several variables? The former corresponds to a purely analog strategy. The precision of single physical variables in an analog strategy must exceed the required precision of the computation as a whole. A small number of highly precise analog computations process such inputs to generate output signals. On the other hand, a purely digital strategy uses many physical variables, each only one-bit precise, to collectively represent input signals. A large number of single-bit precise, or digital, computations process such input signals to generate output signals.

When the required precision is low the most efficient computational strategies exploit details of device physics, and are thus purely analog in nature. However, such strategies are inefficient at high levels of precision because explicit error correction, which is computationally expensive, must be added to maintain precision in the presence of device noise, non-idealities and mismatch. On the other hand, digital computations perform implicit error correction at each step of the computation. As a result their performance is largely independent of detailed device physics. This property is often referred to as *abstraction*. Abstraction makes the efficiency of digital systems degrade much more slowly with increasing precision than analog systems. Thus digital strategies *scale* much better with increasing precision, and are the most efficient choice when the required precision is high.

Our discussion so far suggests that strategies that lie between the purely analog and digital extremes will be optimal (most efficient) at intermediate levels of precision. Individual signals are represented in such hybrid strategies by several moderate-precision analog variables. These signals are processed by a collection of moderate-precision analog computations. Hence we refer to such strategies as examples of *collective analog computation*. The most efficient computers known, like the brain, do use the collective analog paradigm, which lends support to our theory. Thus collective analog systems obtain the 'best of both worlds' by exploiting the desirable features of both analog and digital systems. Figure 1-4 is a highly simplified graphical representation of the analog, collective analog, and digital computational strategies.

The optimal precision of an individual computational unit in a collective analog

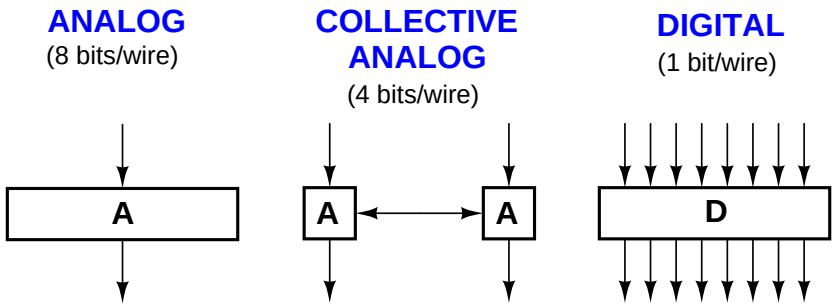


Figure 1-4: Various computational strategies. The strategies are distinguished from each other by the amount of precision assigned to individual physical variables and processing units.

system depends on the relative costs of communication and computation. Communication costs increase as the 'collective' nature of the processing increases, i.e., as we transition from purely analog to digital strategies. In addition, communication costs depend on the physical properties of the computational medium. In a technology where communication costs are low the most efficient strategy is to use a large number of low-precision computational units. As communication costs increase, maximal efficiency is obtained by using a smaller number of higher-precision processors. Thus the optimal strategy becomes more 'analog' as relative communication costs increase [255].

Biology makes extensive use of collective analog computation. Many imprecise analog computational units often interact to solve biological problems precisely or quickly. Good examples include neural networks in the brain and gene-protein networks in the cell. The interactions between computational units in such networks can have variable levels of precision, i.e., be analog or digital in nature. Such interactions serve different purposes. For example, digital symbols can be used to restore analog signals to fixed levels, thereby removing noise from them. Action potentials, which dominate interactions in neural networks, have moderate levels of precision. It has been estimated that they each carry, on average, between 2 and 3 bits of information.

Many practical computers are *heterogenous*, and use a variety of signal representations. The representations are often dictated by the input and output requirements, i.e., the need to interface with other systems. Common examples include sensing sys-

tems that digitize and process analog measurements of physical variables. Broadly speaking, there are two ways in which collective analog computation has been used in such heterogenous systems. The first technique uses analog pre-processing to extract features of interest from the input signal prior to digitization [261]. Such systems increase energy efficiency by digitizing only features in the signal that convey information, not the high-bandwidth input signals themselves. Radio receivers are good examples of such systems. The second technique uses one or more analog and digital subsystems that interact in real time via feedback. Such systems are generalizations of finite state machines that have been referred to as *hybrid state machines* [260].

This thesis focuses on two biological examples that neatly illustrate the two techniques described in the previous paragraph. The mammalian cochlea is an example of delayed digitization. It uses the properties of a physical medium to perform broadband, real-time spectrum analysis of sounds prior to digitization, i.e., conversion to action potentials on the auditory nerve. Intracellular reaction networks of genes and proteins are examples of hybrid state machines. They perform low-power stochastic computations, with genes and proteins acting as digital and analog subsystems, respectively.

1.4 Thesis Overview

We conclude this introductory chapter by briefly summarizing the other chapters and appendices included in this thesis. The summaries provided are intended to assist the reader in negotiating this document.

- **Chapter 2**: This chapter describes the design and implementation of a integrated, active bidirectional RF cochlea that operates between 1GHz and 8GHz and provides 50 exponentially-spaced output channels. We also theoretically analyze the effects of spatially discretizing a wave propagation medium for implementation with lumped on-chip components.

- **Chapter 3**: This chapter describes the design and implementation of a inte-

grated, active unidirectional RF cochlea that operates between 600MHz and 6GHz and provides 51 exponentially-spaced output channels. We also theoretically analyze trade-offs in the design, particularly with respect to frequency resolution, dynamic range, power consumption and chip area.

- **Chapter 4**: This chapter describes theoretical work that explores several analogies between analog to digital converters and frequency estimators. We also describe algorithms that use the broadband filtering provided by the cochlea as a basis for rapid, accurate estimation of a spectrum of input frequencies.

- **Chapter 5**: This chapter takes a fundamental approach towards understanding the phenomenon of noise in physical systems. We describe classical and quantum theories of noise in electronics and chemistry. We also present experimental measurements of flicker noise on extremely long timescales in MOS transistors.

- **Chapter 6**: This chapter describes the design and implementation of programmable, integrated analog computers for rapid stochastic simulation, parameter estimation and optimization of large systems of chemical reactions. Examples include metabolic and cell-signalling pathways, and other interaction networks in systems biology. We also describe experimental results from a prototype chip that can be programmed to simulate arbitrary networks of chemical reactions.

- **Chapter 7**: We describe the design and implementation of integrated hybrid (analog-digital) computers for rapid stochastic simulation, parameter estimation and optimization of gene regulation networks. We also describe experimental results from a prototype chip that models the activation, transcription and translation of multiple genes.

- **Chapter 8**: This chapter concludes the thesis. We summarize our findings, and put forward possible directions for future work.

- **Appendix A**: This appendix describes the design and implementation of highly-power-efficient wireless data transmission links for biomedical implants. We use

the impedance modulation technique to reduce the power consumption of the implanted unit. We also describe theoretical work on finding fundamental physical limits on the performance of these links.

- **Appendix B**: This appendix describes the design and implementation of an ultra-low-power chip specialized for body sensor networks. The chip supplies power to sensors such as microphones, measures their outputs, such as heart sounds, and wirelessly transmits them to a remote location. We also demonstrate that the chip can be combined with a RF power harvester to create a wearable, battery-free tag for pervasive health monitoring.

- **Appendix C**: This appendix describes the classical theory of passive impedance-matching networks, and also discusses the impedance-matching of antennas.

- **Appendix D**: This appendix describes the design of two novel circuit architectures. The first system is a low-supply-voltage, adaptive bandwidth, current-mode phase-locked loop (PLL). The second part of the appendix analyzes the use of switched-gain feedback controllers to improve the performance of analog circuits, such as operational amplifiers.

- **Appendix E**: This appendix describes an interesting number-theoretic problem that originally arose in the design of high-resolution digital-to-analog converters. The problem is to determine the number of unique entries present within a multiplication table.

- **Appendix F**: This appendix presents simplified derivations of the radiation resistance of electrically small dipole and loop antennas.

1.5 Related Publications

- **Chapters 2 and 3**

  1. "A Proposal for an RF Cochlea", S. Zhak, S. Mandal and R. Sarpeshkar,

Asia Pacific Microwave Conference, APMC 2004 (invited paper), New Delhi, December 2004.

2. "Circuits for an RF Cochlea", S. Mandal, S. Zhak and R. Sarpeshkar, *IEEE International Conference on Circuits and Systems (ISCAS)*, Kos, Greece, May 2006.

3. "A Bio-Inspired Active Radio-Frequency Silicon Cochlea", S. Mandal, S. M. Zhak, R. Sarpeshkar, *accepted, IEEE Journal of Solid-State Circuits*, 2009.

- **Chapter 4**

4. "Architectures for Universal or Software Radio", S. Mandal, S. Zhak, and R. Sarpeshkar, U.S. provisional patent application 60/870,719 filed December 19th 2006. Utility application 11/958,990 filed December 18th 2007. Published US2008/0240301 A1, October 2008.

- **Chapter 5**

5. "Sub-μHz MOSFET 1/f Noise Measurements", S. Mandal, S. K. Arfin and R. Sarpeshkar, *Electronics Letters*, Vol. 45, Issue 1, January 2009, pp. 81-82.

- **Chapter 6**

6. "Log-Domain Circuit Models of Chemical Reactions", S. Mandal, R. Sarpeshkar, *accepted, IEEE International Symposium on Circuits and Systems*, Taipei, Taiwan, May 2009.

- **Appendix A**

7. "Low-Power Circuits for Brain-Machine Interfaces", R. Sarpeshkar, W. Wattanapanitch, B. I. Rapoport, S. K. Arfin, M. W. Baker, S. Mandal, M. Fee, S. Musallam and R. A. Andersen, *IEEE International Conference on Circuits and Systems (ISCAS)*, New Orleans, May 2007.

8. "A Bidirectional Wireless Link for Neural Prostheses that Minimizes Implanted Power Consumption", S.Mandal and R. Sarpeshkar, *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Montreal, Canada, November 2007.

9. "Low-Power Circuits for Brain-Machine Interfaces", R. Sarpeshkar, W. Wattanapanitch, B. I. Rapoport, S. K. Arfin, M. W. Baker, S. Mandal, M. Fee, S. Musallam and R. A. Andersen, *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 2, Issue 3, September 2008, pp. 173-183.

10. "Power-Efficient Impedance-Modulation Wireless Data Links for Biomedical Implants", S. Mandal and R. Sarpeshkar, *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 2, Issue 4, December 2008, pp. 301-315.

- **Appendix B**

11. "A Battery-Free Tag for Wireless Monitoring of Heart Sounds", S. Mandal, L. Turicchia, *accepted, Body Sensor Networks Conference*, Berkeley, CA, June 2009.

12. "A Low-Power Battery-Free Tag for Body Sensor Networks", S. Mandal, L. Turicchia, R. Sarpeshkar, *accepted, IEEE Pervasive Computing*, 2009.

Chapter 2

The Bidirectional RF Cochlea

O divine art of subtlety and secrecy! Through you we learn to be invisible, through you inaudible; and hence we can hold the enemy's fate in our hands.

– Sun Tzu, The Art of War (Chapter VI, Weak Points and Strong)

2.1 Introduction

Several silicon-based emulations of the biological cochlea using analog circuits have been reported in the literature. The basic idea has usually been to extract aspects of cochlear behavior using cascaded unidirectional filter sections implemented using analog VLSI at audio frequencies [21, 98, 157, 177, 259, 306, 311]. Less frequently, bidirectional ("traveling wave" or "transmission line") architectures have also been developed [117,305]. In addition, implementations of two dimensional cochlear models have occasionally been reported [219, 272, 296]. A number of digital implementations of the cochlea have also been discussed [137,212]. The related field of cochlear implant processors ("bionic ears") usually employs parallel banks of bandpass filters to obtain frequency selectivity [86, 261].

All the papers mentioned in the previous paragraph discussed cochlea-like structures at audio frequencies. As far as we know, the idea of extending the cochlea concept to higher frequencies (the "RF Cochlea") was first discussed by us in [324].

We have also described some circuits that can be used to build an RF cochlea [188]. Cochlea-like models have also been implemented using micro-fabricated hydromechanical structures [314] and discrete passive components at VHF (20-70MHz) [79,80]. The authors in [80] implement a completely passive bidirectional (transmission-line) model of the biological cochlea. The structure is then used for performing signal separation based on frequency content (i.e., spectral analysis). Elements of cochlear operation have inspired techniques for signal-enhancement [237] and synthesizing RF filters [61]. Micro-strip transmission lines with dispersion have also been used to perform real-time spectrum analysis by converting frequency to time, i.e., group delay [155, 265].

The cochlea performs highly resource-efficient distributed computation by exploiting the properties of a physical medium. Distributed analog physical computation has been used to build several efficient engineering systems, both at RF [3, 102, 241] and at lower frequencies [43].

2.2 Cochlear Models

2.2.1 Introducing Cochlear Mechanics

In this section we provide a brief description of the anatomy and operation of the biological cochlea. For an excellent introduction to cochlear hydrodynamics, see [178]. Further details of cochlear operation may also be found in standard textbooks [85,226] or Lloyd Watts' thesis [305]. The cochlea is a sophisticated signal processing system that converts a single pressure signal into a time-varying pattern of excitation on over 25,000 sensors - fibers of the eighth cranial nerve. Spatiotemporal response patterns on these fibers form the basis of an efficient, robust representation of speech and natural sounds [274, 321]. Active feedback mechanisms are important in normal cochlear operation and have been the subject of much research [27, 47, 48, 74, 176, 244, 270, 271]. In healthy humans, it has 120dB of input-referred dynamic range and consumes only about 14μW of power (see Table 2.1).

Table 2.1: The biological (human) cochlea

Parameter	**Value**
Dynamic range (at input)	120dB
Power consumption	$\approx 14\mu$W
Power supply voltage	150mV
Volume	$\approx$ 35mm$\times$1cm$\times$1cm
Detection threshold at eardrum	0.05Åat 3kHz
Frequency range	20Hz - 20kHz
Output fibers	$\approx 25,000$
Filter bandwidths	$\approx 1/3$ octave
Phase locking threshold	$\approx$ 5kHz

Sound waves travel down the *ear canal* and vibrate the *eardrum* or *tympanic membrane* (see Figure 2-1). Vibrations of the eardrum are coupled into the bones of the middle ear - the *malleus*, *incus* and *stapes*. The flat "footplate" of the stapes transmits acoustic vibrations to the *oval window* of the cochlea. The mammalian cochlea, as shown in Figure 2-2, is a long fluid-filled tube that is partitioned into three compartments, the *scala vestibuli*, *scala media* and *scala tympani*, by two membranes: *Reissner's membrane* and the *basilar membrane*. Reissner's membrane has no known mechanical function. The *round window* is flexible and acts as a pressure reliever for the incompressible cochlear fluid (see Figure 2-3). In humans, the cochlea is about 35mm long; it is coiled up into three-and-a-half turns. The coiling serves no functional purpose but saves space inside the crowded inner ear.

The end of the cochlea closest to the oval window is called the *base*, while the end furthest away is known as the *apex*. Vibrations of the oval window cause waves of fluid pressure and volume velocity[1] to travel down the cochlea. The middle ear couples energy efficiently from the tympanic membrane to the cochlea, but it is not simply an acoustic impedance transformer between the atmosphere and fluid-filled inner ear. This is because wave propagation in the cochlea is not via acoustic compression waves in the cochlear fluid, but via the combined movement of the fluid and the basilar membrane. Thus the middle ear matches the impedances of air and the part of the

[1]Volume velocity is defined as the product of velocity and cross-sectional area perpendicular to the direction of wave propagation.

basilar membrane closest to the oval window. The *organ of Corti* sits on top of the basilar membrane. It contains a single row of *inner hair cells* and three to five rows of *outer hair cells*. To summarize, the basilar membrane, organ of Corti and adjacent cochlear fluid (*endolymph*) all participate in the cochlear traveling wave.

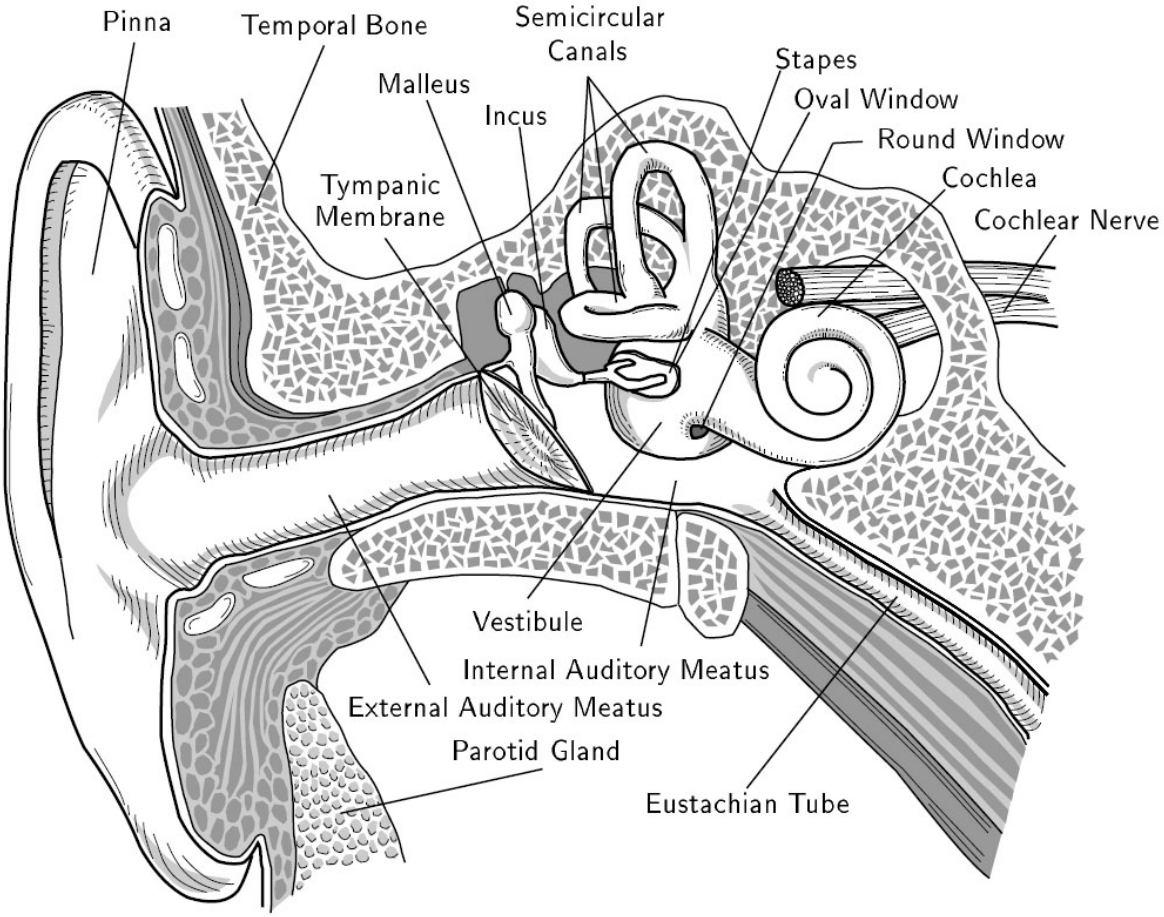


Figure 2-1: Anatomy of the human auditory periphery. Figure adapted from [305].

The basilar membrane becomes wider as we go from the base to the apex; as a result, its stiffness decreases (approximately) exponentially with position. Properties of the hair cells, *tectorial membrane* and *reticular lamina* also scale exponentially with position; thus the mechanical impedance of the organ of Corti also scales similarly. As the traveling wave moves from the base to the apex, its wavelength decreases, and it transitions from a *long wave* to a *short wave*. In the long wave region, the wavelength is much larger than the height h of the cochlear duct; the whole mass of fluid moves horizontally with the wave. When the wavelength becomes much smaller than h, the

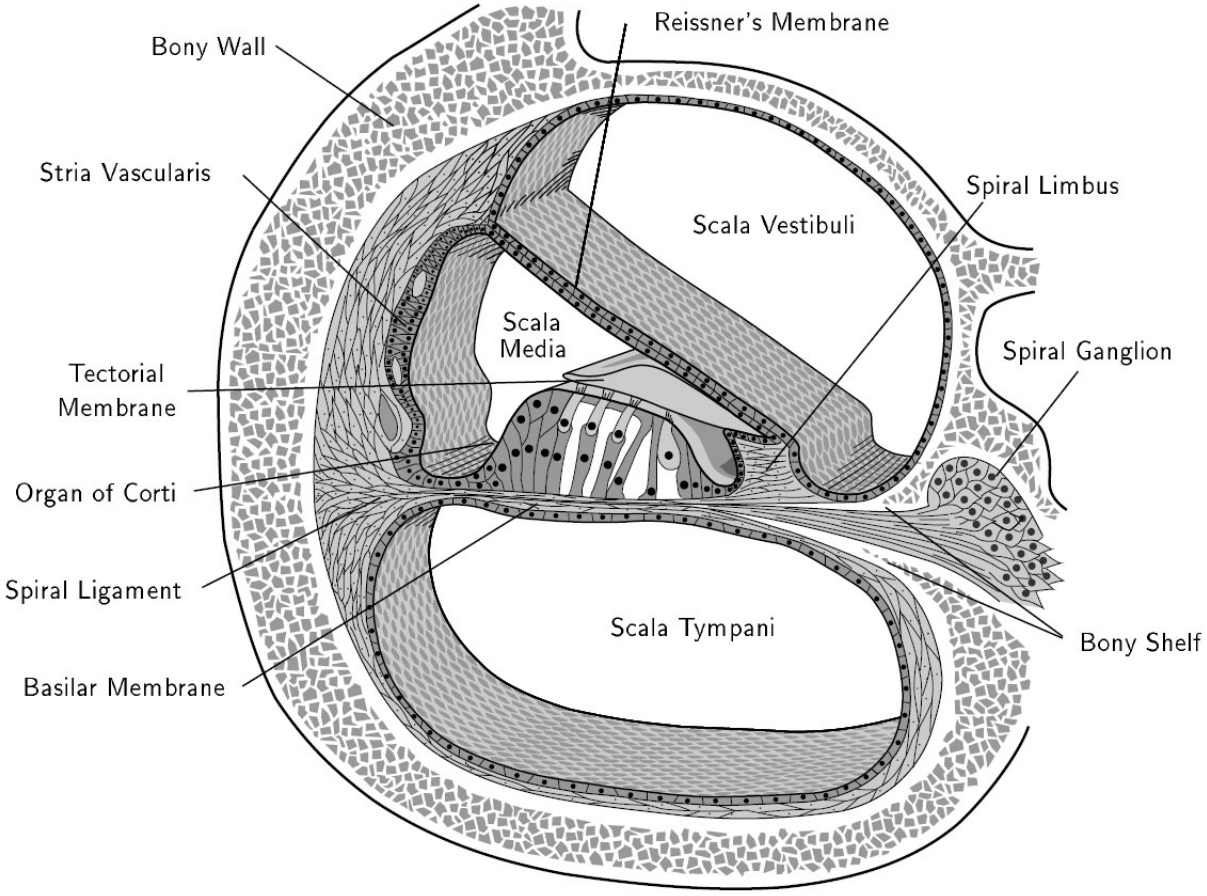


Figure 2-2: Cross-sectional view through the cochlea. Figure adapted from [305].

wave is said to have entered the short wave region. In this region, only the fluid close to the basilar membrane moves[2]. The wave amplitude reaches a maximum at a location x_{max} (the *best place*) that is approximately logarithmic with the frequency ω_{in} of the incoming wave, i.e. $x_{max} \propto \log(\omega(0)/\omega_{in})$, where $\omega(0)$ is the frequency that peaks (the *best frequency*) at the base of the cochlea ($x = 0$). Beyond x_{max}, the wave enters a highly damped *cutoff region* where its amplitude rapidly decreases. Thus high input frequencies excite a response close to the base of the cochlea, while low frequencies peak close to the apex. In effect, the cochlea performs a frequency-to-place conversion.

[2]This is similar to what happens, except in reverse, to an ocean wave as it approaches a gently sloping, sandy beach. For ocean waves, the motion transitions from short wave (deep water) to long wave (shallow water) as it approaches the shore.

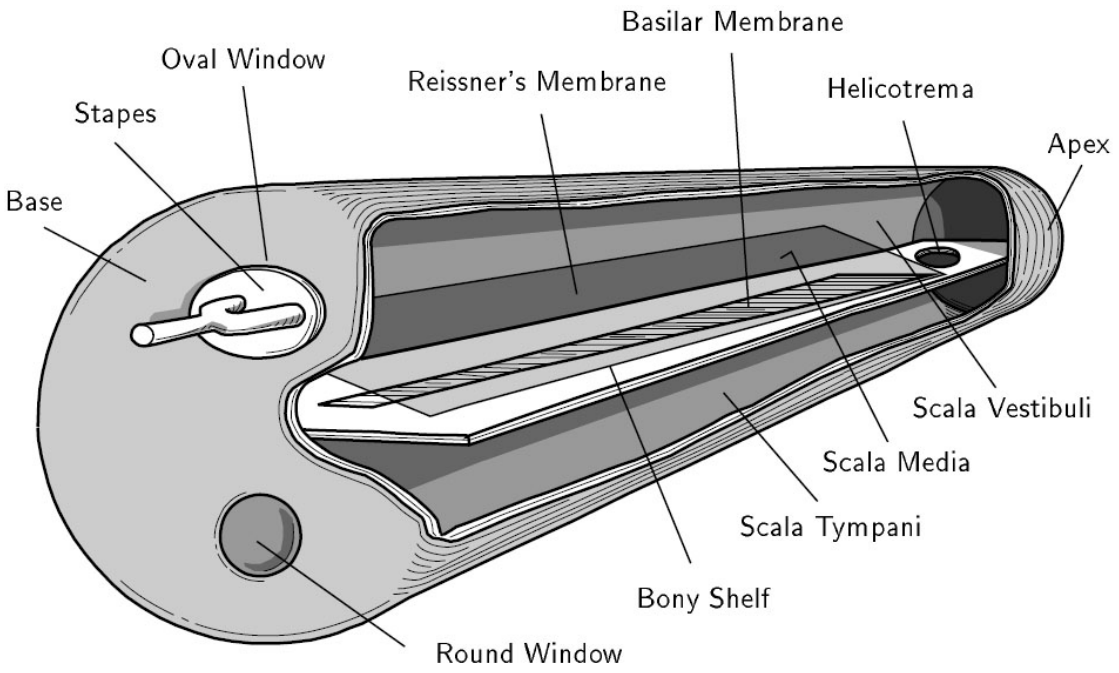


Figure 2-3: Simplified view of the unrolled cochlea. Figure adapted from [305].

There are both active and passive mechanisms involved in the conversion process. First, there is a passive mechanical resonance between the fluid mass and the stiffness of the basilar membrane. Because of the exponential scaling of basilar membrane stiffness, the resonant frequency also scales exponentially from the base to the apex. This gives some frequency selectivity to the cochlea, as shown by measurements on dead cochleas from cadavers. However, *in-vivo* measurements on cochleas that were still alive show much greater frequency resolution: the peaks in the cochlear transfer functions (these functions map input frequency to spatial excitation pattern, or *vice versa* are much higher and sharper. The precise mechanism by which this active amplification occurs is still the subject of active research.

Figure 2-4 shows experimental data from a live mammalian cochlea. The figure contains frequency response magnitudes measured at two different locations x_1 and x_2 along the basilar membrane. The curves are low-pass in character; they increase slowly to peak at the bast place, and then rapidly cut off. The two responses are roughly frequency-shifted versions of each other, showing the frequency-to-place transformation (also known as a tonotopic mapping or specific coding) performed by

the cochlea. The corresponding phase responses show a negative phase accumulation of three to five cycles by the time the best place is reached [240].

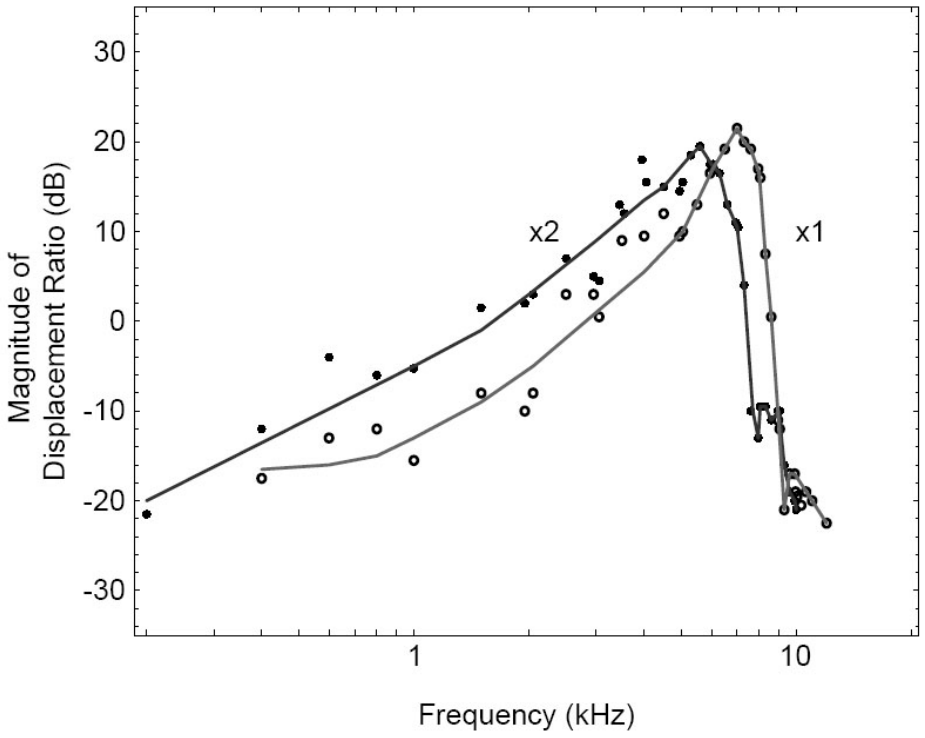


Figure 2-4: Magnitude of the cochlea frequency response, measured from a live squirrel monkey. The two curves correspond to the basilar membrane displacement at two positions x_1 and x_2, where x_1 is 1.5mm closer to the apex than x_2. Figure adapted by [305] from data originally published by Rhode [239, 240].

The organ of Corti almost certainly contains the active amplification mechanism present within the cochlea. The inner hair cells are primarily sensors that are sensitive to the (approximately) half-wave rectified basilar membrane velocity. The inner hair cells stimulate *spiral ganglion* cells synaptically connected to them to fire neural spikes, which are then conveyed to the auditory cortex in the brain. Interestingly, individual nerve fibres only have a dynamic range of 25-30dB in firing rates. It seems that the cochlea increases its effective dynamic range by using 10-20 nerve fibres to encode the output of a single inner hair cell.

According to most current theories, the outer hair cells, which are sensitive to basilar membrane displacement, actively amplify the traveling wave in a region towards the basal side of the best place. This raises and sharpens the peaks of the

cochlear transfer functions. However, the precise mechanism by which this happens is unclear, though many theories have been proposed [173]. Since we are not interested in precisely modeling the biological cochlea, but rather in using ideas from it to build an efficient engineering solution, the controversy need not concern us further. However, the action of outer hair cells makes the cochlear response nonlinear, which is is important for us.

The most important nonlinear response exhibited by the cochlea is compressive gain control. The peaks of the cochlear transfer functions are strongly compressive with incident sound pressure level (SPL) over the range of normal acoustic stimuli. This is shown in Figure 2-5, where the measured amplitude dependence of the basilar membrane response to SPL indicates a strongly compressive nonlinearity. Gain compression, in combination with smart neural encoding, allows the cochlea to compress 120dB of input dynamic range into about 40dB of basilar membrane displacement and subsequently, into nerve fibers with even smaller individual dynamic range.

2.2.2 Zweig's Cochlear Model

The great German physician and physicist Hermann von Helmholtz (1821-1894) was the first to propose a physical model of cochlear action. His work was translated into English in 1885 as a volume entitled *On the Sensations of Tone*. Helmoltz's model resembled a piano, or filter bank. It assumed that sections of the basilar membrane (BM) behave as independent bandpass filters with center frequencies that decrease with position, and ignored the fluids within the cochlear canal.

The Hungarian biophysicist Georg von Békésy (1899-1972) realized the importance of endolymphatic fluid in coupling together sections of the BM. He established the principle of a frequency-place transformation by discovering that different positions along the BM have different characteristic frequencies, and developed the traveling wave model of the cochlea to explain his observations. This model still dominates the field. The most comprehensive description of von Békésy's fundamental work may be found in a book entitled *Experiments in Hearing* that was first published in 1960.

Because the model proposed by Zweig in [328] forms the basis of our work on

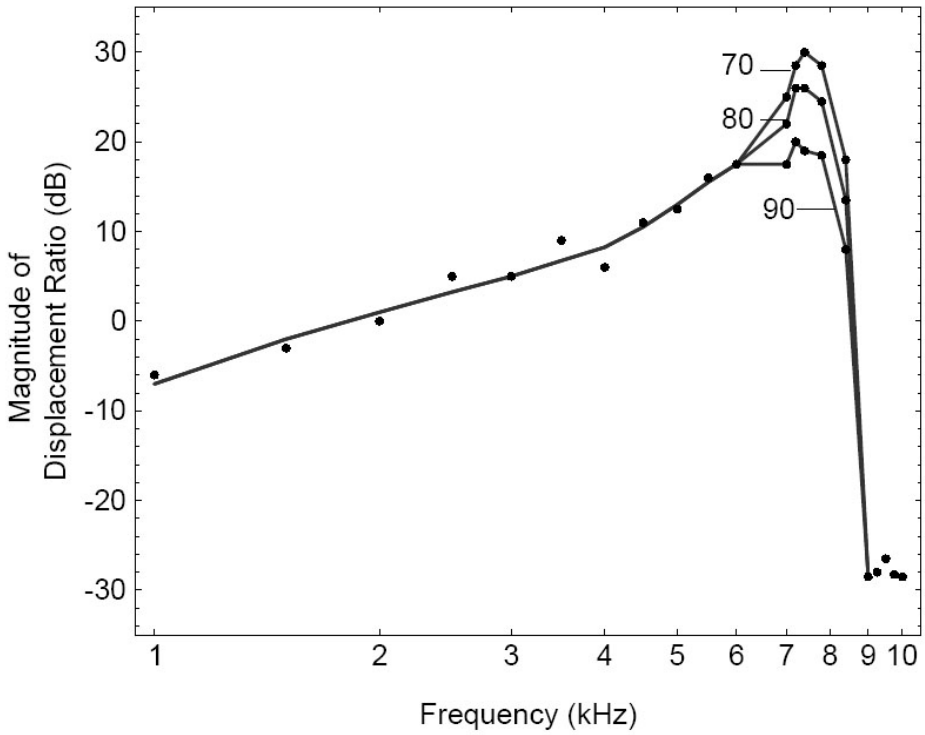


Figure 2-5: Magnitude of the cochlea frequency response, measured from a live squirrel monkey and showing amplitude nonlinearity (gain compression). Sound pressure levels (SPL) in dB are marked on the figure. Figure adapted by [305] from data originally published by Rhode [239, 240].

cochlear modeling, we shall now briefly review it. An electrical circuit equivalent of Zweig's cochlea model is shown in Figure 2-6. The model, which is based on von Békésy's traveling wave paradigm, is a one-dimensional mechanical transmission line. It is a reasonable approximation of the biological cochlea, which is a complex three-dimensional structure, if the following assumptions are valid:

1. The radius of curvature of the cochlear spiral is much larger than the wavelength of traveling waves on the scala media, allowing the spiral to be uncoiled without affecting cochlear operation.

2. The fluids in the scala vestibuli and tympani are incompressible and inviscid.

3. The scala media consists of an array of beams, acting as oscillators, that are only coupled together by the adjacent fluid. The characteristic frequency of these oscillators decreases exponentially with increasing distance from the stapes.

4. The pressure is uniform at any cross-section that is orthogonal to the main axis of the cochlea, i.e., pressures depend only on x, the distance from the stapes. This approximation is valid in the long wave region, i.e., when the wavelength of the traveling wave is much larger than the height of the cochlear duct[3].

The circuit variables in this model are P, the pressure variable across the scala media and U, the volume velocity of the cochlear fluid. These are mapped into voltage V and current I, respectively, in the electrical model. Thus the mechanical-to-electrical domain mapping used is $(P, U) \rightarrow (V, I)$.

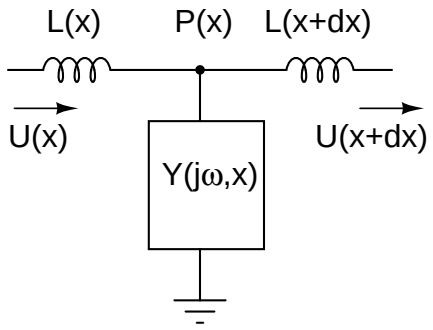


Figure 2-6: Zweig's continuous transmission line cochlea model.

Zweig's cochlea model consists of complex shunt admittances $Y(\jmath\omega, x)$ that model the organ of Corti and the action of outer hair cells. These impedances are coupled together using series impedances $Z(\jmath\omega, x)$ that consists of inductors that model the fluid mass close to the basilar membrane that participates in the traveling wave. The whole system is exponentially tapered, i.e. the shunt admittance and series impedance per unit length both increase exponentially with position x. This model is bidirectional since the coupling inductors are symmetric elements and waves can propagate in both directions along the transmission line. It is a simplified, one dimensional representation of the three dimensional cochlear geometry. One can imagine extending it to two dimensions by replacing the line of series elements (inductors) with a grid of them. Such cochlear models have been built at audio, and exhibit some phenomena,

[3]The approximation breaks down once the short wave region is reached. The transition between long and short waves occurs just basal of the best place.

such as multiple propagating wave modes, which are found in the real cochlea but are absent in our one-dimensional model.

It is important to note that both $Z(\jmath\omega, x)$ and $Y(\jmath\omega, x)$ have values *per unit length*; thus $Z(\jmath\omega, x)$ has units of Ω/m. Similarly, $Y(\jmath\omega, x)$, the admittance per unit length, has units of $\mho$/m.

2.2.3 Analysis of Zweig's Model

Electrically, Zweig's cochlear model consists of a transmission line with properties that vary slowly with position. We now derive the equations for wave propagation in this model. Throughout this analysis, we shall assume that sinusoidal steady state has been reached, i.e., all impedances can be represented using complex exponentials. Firstly, we notice that the cochlea scales exponentially with position variable x. Thus the center frequency (i.e., the frequency at which maximum gain occurs) at any location is given by

$$\omega_c(x) = \omega_c(0) \exp\left(-\frac{x}{l}\right) \tag{2.1}$$

where x increases from the base to the apex of the cochlea, $\omega_c(0)$ is the center frequency at the basal end and l is a characteristic length that defines the scale of the cochlea's exponential taper. Experimentally, it is found that the cochlea's frequency and spatial response functions look similar. In fact, the transfer functions do not depend on ω and x separately, but on the combined variable $\omega/\omega_c(x) = \exp(x/l)\omega/\omega_c(0)$. This relationship, known as *scaling symmetry*, is only approximately true in the biological system. However, we shall always assume its validity, because it allows us to infer one type of response from the other. For example, the effects of doubling ω are identical to increasing x by an amount $l\ln(2)$. Thus, scaling symmetry reduces the number of independent variables in the problem from two to one. We take advantage of it by defining a dimensionless, normalized variable s_n, as follows:

$$s_n = \frac{\jmath\omega}{\omega_c(x)} \tag{2.2}$$

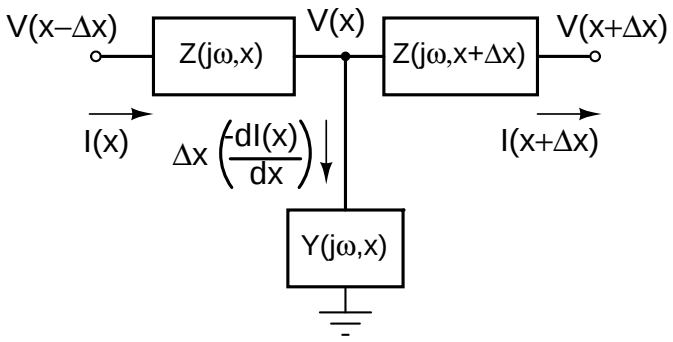


Figure 2-7: A generic spatially-varying one-dimensional transmission line, with series impedances represented by Z and shunt admittances by Y.

By writing circuit impedances in terms of s_n, we can make the equations for wave propagation dependent only on s_n. Using Kirchoff's current and voltage laws, the equations for voltage V (corresponding to fluid pressure P) and current I (corresponding to volume velocity U) on the generic spatially-varying transmission line shown in Figure 2-7 in sinusoidal steady-state are given by

$$\begin{aligned} \frac{dV}{dx} &= -Z(\omega, x)I \\ \frac{dI}{dx} &= -Y(\omega, x)V \end{aligned} \tag{2.3}$$

where $Z(\omega, x)$ and $Y(\omega, x)$ are the impedance and admittance *per unit length* of the line. Because of scaling symmetry in the cochlea, we have $Z(\omega \exp(-x/l), x) = Z(\omega, 0)$ and $Y(\omega \exp(-x/l), x) = Y(\omega, 0)$, i.e., the impedances and admittances at any two positions in the cochlea are identical if we exponentially scale the frequencies at which they are compared.

Since $ds_n/dx = s_n/l$, we can eliminate the separate dependencies on ω and x by using the chain rule of differentiation to rewrite (2.3) as

$$\begin{aligned} \frac{dV}{ds_n} = \frac{dV}{dx} \times \frac{dx}{ds_n} &= -\left(\frac{lZ\,(s_n)}{s_n}\right) I \\ \frac{dI}{ds_n} = \frac{dI}{dx} \times \frac{dx}{ds_n} &= -\left(\frac{lY\,(s_n)}{s_n}\right) V \end{aligned} \tag{2.4}$$

Hereafter, we use the convention that Z and Y refer to impedance and admittance per unit length in the continuous transmission line, Z_d and Y_d refer to impedances and admittances in the spatially-discretized, or lumped transmission line, and Z_n and Y_n are normalized, dimensionless forms of Z_d and Y_d. Each stage contains a series impedance Z_d and a shunt admittance Y_d, given by

$$\begin{aligned} Z_d &= (\Delta x) Z(s_n) \\ Y_d &= (\Delta x) Y(s_n) \end{aligned} \tag{2.5}$$

The series impedance Z_d consists of an inductance that models fluid mass and increases exponentially with position, resulting in

$$\begin{aligned} Z_d(s_n) &= j\omega L_0 \exp(x/l) \\ &= \frac{\omega}{\omega_c(x)} \left(\omega_c(0) \exp(-x/l) \times L_0 \exp(x/l)\right) \\ &= s_n \omega_c(0) L_0 \end{aligned} \tag{2.6}$$

where $L_0 = (\Delta x) L(0)$, and $L(0)$ is the inductance per unit length at $x = 0$. The normalized, dimensionless forms of Z_d and Y_d are given by

$$\begin{aligned} Z_n &= \frac{Z_d}{Z_0} \\ Y_n &= Y_d Z_0 \end{aligned} \tag{2.7}$$

The normalizing impedance Z_0 is a constant that, in a real implementation, scales all dimensionless impedances and provides a degree of freedom in the design. In the RF cochlea, Z_0 is chosen to make on-chip implementation practical, as we discuss later. From (2.6) and (2.7), Z_n is given by

$$Z_n(s_n) = s_n\left(\frac{\omega_c(0)L_0}{Z_0}\right) = s_n Q_{line} \tag{2.8}$$

were $Q_{line} \equiv \omega_c(0)L_0/Z_0$ is a dimensionless constant. The normalized shunt admittance Y_n models the complex behavior of the organ of Corti. In his original paper, Zweig used experimental measurements to propose the following form for Y_n [328]:

$$Y_n(s_n) = \frac{s_n}{s_n^2 + \delta s_n + 1 + m(s_n)} \tag{2.9}$$

One may recognize this admittance as a simple series R-L-C section in series with an unknown impedance $m(s_n)/s_n$. This impedance models the active response of the organ of Corti to the traveling wave, i.e., mostly the effect of outer hair cells. From experimental data, $m(s_n)$ was found to be well approximated by a pure delay with some gain ρ, i.e. $m(s_n) \approx \rho\exp(-\psi s_n)$, where ψ is the delay in radians. Thus $m(s_n)/s_n$ looks like a capacitor with delay, and Y_n is given by

$$Y_n(s_n) = \frac{s_n}{s_n^2 + \delta s_n + 1 + \rho\exp(-\psi s_n)} \tag{2.10}$$

where δ, ρ and ψ are constants. This admittance function can be interpreted as a feedback loop containing an RLC resonator and a pure delay. Its frequency response is shown in Figure 2-8 for the following parameter values (obtained from [328]): $\delta = -0.122$, $\rho = 0.142$, $\psi = 2\pi \times 1.74$.

Since Y_n contains a pure delay term, it is not a rational function of s and has an infinite number of poles. Of these, two nearly coincident pairs of complex poles near the imaginary axis (i.e., $s_n = \jmath\omega_n$) at $\omega_n = 1$) are the most important for determining the collective response of the cochlear model. The *two* pairs of complex poles near the $\jmath\omega_n$ axis ensure that both the admittance Y_n and its first derivative w.r.t. frequency have large values close to $\omega_n = 1$. Symbolically,

$$Y_n(s_n)|_{s_n=j} \approx \infty, \left.\frac{dY_n(s_n)}{ds_n}\right|_{s_n=j} \approx \infty \tag{2.11}$$

Physically, (2.11) means that the shunt admittance $\omega_n Y_n(\omega_n)$ is large for a *range*

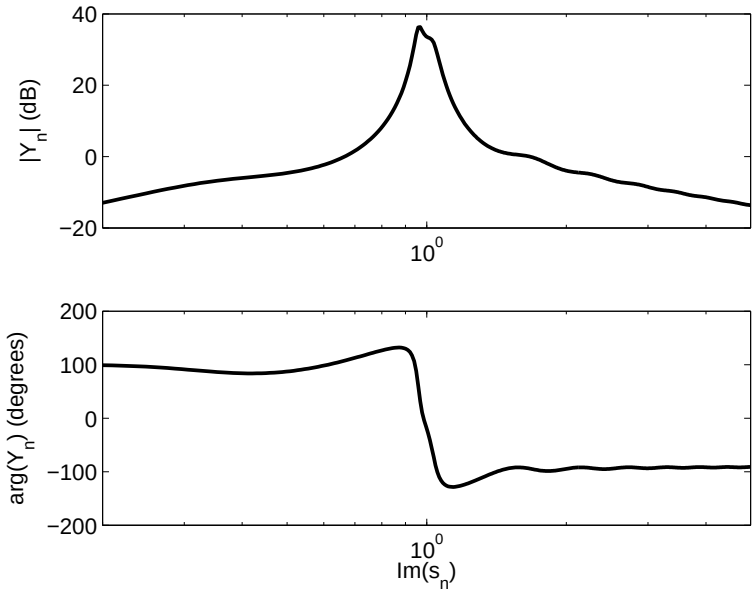


Figure 2-8: Normalized BM admittance Y_n proposed by Zweig [328].

of frequencies around $\omega_n = 1$. This behavior is the essence of collective amplification: a number of cochlea stages will contribute to the overall transfer function of the system. Zweig's model therefore provides important insight into the mechanism by which the cochlea obtains frequency selectivity. Unfortunately, the admittance $Y_n(s_n)$ is difficult to implement. Since Y_n is not a rational function and contains a pure delay, it cannot be synthesized with a finite number of lumped elements. We therefore used a rational function to approximate (2.10) [323]. The function we chose is given by

$$Y_n(s_n) = \frac{s_n\left(s_n^2 + \frac{\mu}{Q}s_n + \mu^2\right)}{\mu^2\left(s_n^2 + 2ds_n + 1\right)^2} \tag{2.12}$$

where μ, Q and d are constants. This admittance is the simplest rational function that contains all essential features of (2.10):

- As described earlier, two pairs of high-Q complex poles near the $j\omega$ axis are needed for collective amplification. Our admittance function therefore includes four poles, divided into two identical complex pairs with some damping $d > 0$.

- We see that the phase of Zweig's admittance function goes from to $-90°$ at

high frequencies. Therefore Y_n must contain one more pole than zero, i.e., look inductive at high frequencies. The three zeros that we need are implemented using a complex pair and a third one at the origin. The zero at the origin is needed to replicate another feature of Zweig's function, which is a phase of $+90°$ at low frequencies (capacitive behavior).

- The phase of Zweig's function shows a positive "bump" between $\omega_n \approx 0.6$ and $\omega_n \approx 0.9$. This behavior must be due to the pair of complex zeros, and can be modeled in equation (2.12) by placing them at a somewhat lower frequency than the poles, i.e., by making $\mu < 1$.

Allowable values of μ, Q and d at all amplitudes are further constrained by the requirement that zero-crossings in the transient response remain approximately invariant with input amplitude, like in the biological cochlea [270]. An important advantage of our rational admittance function is that such invariance can be guaranteed if the single parameter d is varied with signal level as long as $d \ll 1$ [323]. As a result, a purely local gain control strategy can be easily implemented. We used the following default parameter values in our bidirectional cochlea design: $d = 0.1$, $\mu = 0.76$ and $Q = 3.8$. The resultant form of Y_n, which is shown in Figs. 2-9(a) and 2-9(b), is quite similar to Zweig's function, which is shown in Fig. 2-8.

2.2.4 The WKB Solution

At frequencies much smaller than the maximum operating frequency $\omega_c(0)$ the input impedance of the cochlea is given by the following standard expression for a continuous transmission line with series impedances Z_d and shunt admittances Y_d:

$$Z_{in} \approx \sqrt{Z_d/Y_d} = Z_0\sqrt{Z_n/Y_n} \tag{2.13}$$

When $s_n \ll 1$, we see from (2.12) that $Y_n \approx s_n$. Therefore, at frequencies that are much smaller than the local center frequency Y_n looks like a capacitor. Substituting for Y_n and Z_n in (2.13), we find that the input impedance at frequencies much smaller

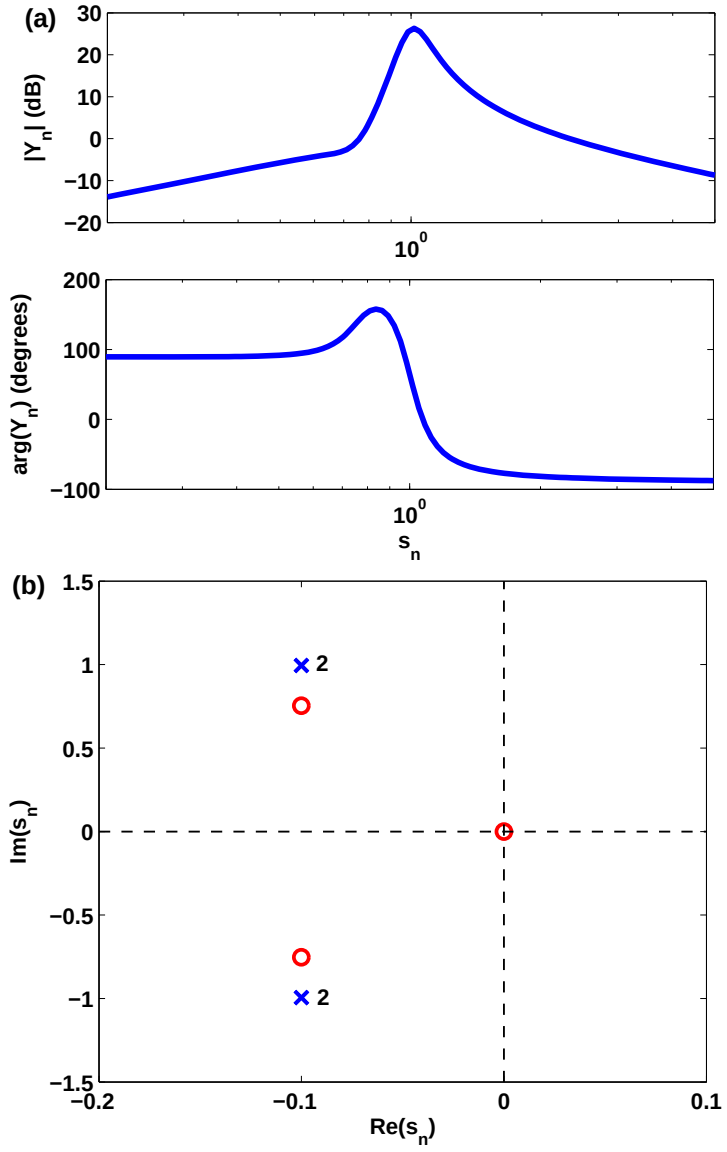


Figure 2-9: (a) Normalized BM admittance Y_n used in the cochlea, (b) pole-zero plot for Y_n.

than $\omega_c(0)$ is given by

$$Z_{in} = \sqrt{\omega_c(0) L_0 Z_0} = Z_0 \sqrt{Q_{line}} \tag{2.14}$$

We usually fix $Z_{in} = 50\Omega$ for compatibility with standard RF test equipment. The sizes of capacitors in the design scale like

$$C_0 = \frac{1}{\omega_c(0) Z_0} \tag{2.15}$$

Equivalently, we have $Z_{in} = \sqrt{L_0/C_0}$. The exponential decrease of center frequency with position is accomplished by increasing inductor and capacitor values in Z_d and Y_d exponentially with stage number i, i.e., making L_i and C_i (and all other inductors and capacitors used to implement Y_d) scale as $\exp(i/N_{nat})$, while resistances remain fixed [230, 305].

The system of first-order ODEs shown in (2.4) can be combined into a single second-order ODE, given by

$$\begin{aligned} \frac{d^2V}{ds_n^2} &= k_n^2 V + \frac{1}{Z}\left(\frac{Z}{s_n} - \frac{dZ}{ds_n}\right)\frac{dV}{ds_n} \\ &= k_n^2 V \end{aligned} \tag{2.16}$$

where $k_n = l\sqrt{ZY}/s_n = N_{nat}\sqrt{Z_n Y_n}/s_n$ is a dimensionless variable. Note that the second term on the first line of (2.16) is zero because $Z/s_n = dZ/ds_n$ when Z is assumed to be a pure inductance, i.e., when $Z \propto s_n$. The simplified equation on the second line is only valid for this specific choice of Z. For example, the addition of series resistance to the inductor (unavoidable in any practical implementation) will makes it invalid. For the moment, we assume that Z is an ideal inductor.

If k_n was constant with s_n, as in an uniform transmission line, the solution to (2.16) would simply be the complex exponential $\exp(k_n s_n)$. Now assume that k_n is not constant, but, as in the cochlea, varies slowly with s_n, i.e., such that $|dk_n/ds_n| \ll k_n$. In this case, we can solve (2.16) for $V(s_n)$ by using the well-known Wentzel-Kramers-

Brillouin (WKB) approximation.

The WKB approximation[4] is a powerful method that can be used to find solutions to various wave propagation problems. It was first used to find semi-classical solutions to the Schrödinger equation in quantum mechanics. Geometric optics is another good example of such a WKB-type solution. The method is generally applicable when the frequency of the wave is much larger than the "spatial frequency" of variations in the propagating medium. Equivalently, WKB solution are valid if the wavelength of the wave is very short compared to the length scale of these variations. Under these conditions, the medium appears to vary slowly relative to the propagating wave. Finally, we should also note that the WKB approximation *assumes a linear system*, and is therefore only valid for small signal amplitudes.

The WKB technique consists of expanding the solution of the differential equation into a product of exponential functions, and then solving for and retaining only the first two terms. The first term in the WKB approximation is a complex exponential where the phase $k_n s_n$ is replaced by the integral of k_n over s_n, i.e., by $\int_0^{s_n} k_n ds'$. It can be shown that the second term in the approximation is proportional to $1/\sqrt{k_n}$. Thus, complete WKB-type solutions to (2.16) are given by

$$V(s_n) = \frac{\alpha}{\sqrt{k_n(s_n)}} \exp\left(\pm \int_0^{s_n} k_n(s')\, ds'\right) \tag{2.17}$$

where α is a constant. Equation (2.17) predicts that the voltage at the input (base) and/or at low frequencies is $V(0) = \alpha/\sqrt{k_n(0)} = \alpha/\left(N_{nat}^{1/2} Q_{line}^{1/4}\right)$. Thus, we may write

$$\frac{V(s_n)}{V(0)} = \frac{N_{nat}^{1/2} Q_{line}^{1/4}}{\sqrt{k_n(s_n)}} \exp\left(\pm \int_0^{s_n} k_n(s')\, ds'\right) \tag{2.18}$$

The *transfer function* (TF) of the cochlea, which is defined as the current flowing through the shunt admittance Y, normalized by the input current $I(0)$ [328], can be written as

[4]Sometimes also known as the Liouville-Green (LG) solution.

$$TF\left(s_n\right) = \frac{\Delta x}{I(0)}\frac{dI}{dx} = \frac{\Delta x}{I(0)}\frac{dI}{ds_n} \times \frac{ds_n}{dx} = \frac{\Delta x}{l}\frac{s_n}{I(0)}\frac{dI}{ds_n} \tag{2.19}$$

Substituting dI/ds_n from (2.4) in (2.19), and remembering that $N_{nat} = l/(\Delta x)$, we find that

$$TF\left(s_n\right) = \frac{s_n}{Z_0 Q_{line}}\left(\frac{k_n}{N_{nat}}\right)^2 \frac{V\left(s_n\right)}{I(0)} \tag{2.20}$$

Also, we know that $V(0)/I(0) = Z_{in}$, the input impedance of the cochlea. Substituting for Z_{in} from (2.14), we get

$$TF\left(s_n\right) = \frac{s_n}{\sqrt{Q_{line}}}\left(\frac{k_n}{N_{nat}}\right)^2 \frac{V\left(s_n\right)}{V(0)} \tag{2.21}$$

We can substitute $V\left(s_n\right)/V(0)$ from (2.18) into the expression for the cochlear TF defined in (2.21), to find that the cochlear TF is proportional to

$$\begin{aligned} TF\left(s_n\right) \quad \propto \quad & s_n k_n^{3/2}\left[c_1 \exp\left(-\int_0^{s_n} k_n ds'\right) + \right. \\ & \left. c_2 \exp\left(+\int_0^{s_n} k_n ds'\right)\right] \end{aligned} \tag{2.22}$$

where c_1 and c_2 are constants determined by $I(0)$, N_{nat}, Z_0 and boundary conditions. The two terms correspond to wave propagation in the $+x$ (forward) and $-x$ (reflected) directions, respectively. The reflected wave is undesirable, and its amplitude c_2 should be minimized.

The exponential part of the WKB solution is an example of collective amplification. Practically, collective amplification increases both the heights and widths of peaks in the cochlear transfer functions. The TF expressed in (2.22) reaches its peak values near maxima of k_n. In our case, $k_n = N_{nat}\sqrt{Z_n Y_n}/s_n$, so $k_n \to \infty$ at poles of Y_n, which occur close to $s_n = \jmath$. Equation (2.22) then predicts $|TF| \to \infty$. In reality, the gain is not infinite. Instead, the WKB approximation breaks down: the assumption that k_n varies slowly is invalid close to its poles. Thus the WKB method

cannot predict the peak value of the cochlear transfer function.

In a real implementation, nonlinear effects also start to appear as the transfer function amplitude rises, limiting its ultimate increase. The physical analog of this behavior occurs in the biological cochlea. Close to the best frequency, the wavelength of the propagating wave decreases as its amplitude builds up: the wave behavior changes from long wave (where the WKB approximation is valid) to short wave (where it is not), and then cuts off [305][5].

Figures 2-10 and 2-11 show the WKB-type solutions predicted by (2.18) and (2.21) for the typical parameter values $d = 0.1$, $\mu = 0.76$, $Q = 3.8$, $Q_{line} = 0.05$, and two different values of N_{nat}. We assumed that the reflected wave was absent, i.e., $c_2 = 0$ in (2.22). Larger values of N_{nat} cause more stages to participate in collective amplification. In other words, the effective number of poles in the transfer function increases. As a result the maximum gain, negative phase accumulation, high-frequency roll-off slope and group delay of the cochlear transfer function all increase with N_{nat}, as shown in Figures 2-10 and 2-11. Group delay, which is defined as $\tau_g = -d\phi/d\omega_n$, has units of time. Here ϕ is the phase of the transfer function. In order to express group delay as cycles of the input frequency, as shown in the figure, we multiply τ_g by $\omega_n/(2\pi)$.

2.2.5 Transmission Line Design

The center frequency of the last, or apical, cochlear stage is given by $\omega_{c,min} = \omega_c(0) \exp(-N/N_{nat})$, where N is the total number of stages. In order to reduce reflections from the apex the transmission line must be terminated with an impedance-matched load. We found that a termination impedance consisting of a resistor R_t in series with an inductor L_t provides adequate matching over the frequency range of interest, namely $0 < \omega < \omega_{c,min}$. We use $R_t = Z_{in}$, which provides a match at frequencies much smaller than $\omega_{c,min}$. We also make the magnitude L_t's impedance at $\omega_{c,min}$ equal to Z_{in}. which provides a match at frequencies comparable to $\omega_{c,min}$.

[5] Another example of the same phenomenon, this time from geometrical optics, is the formation of caustics. Caustics are found, for example, on the focal planes of lenses.

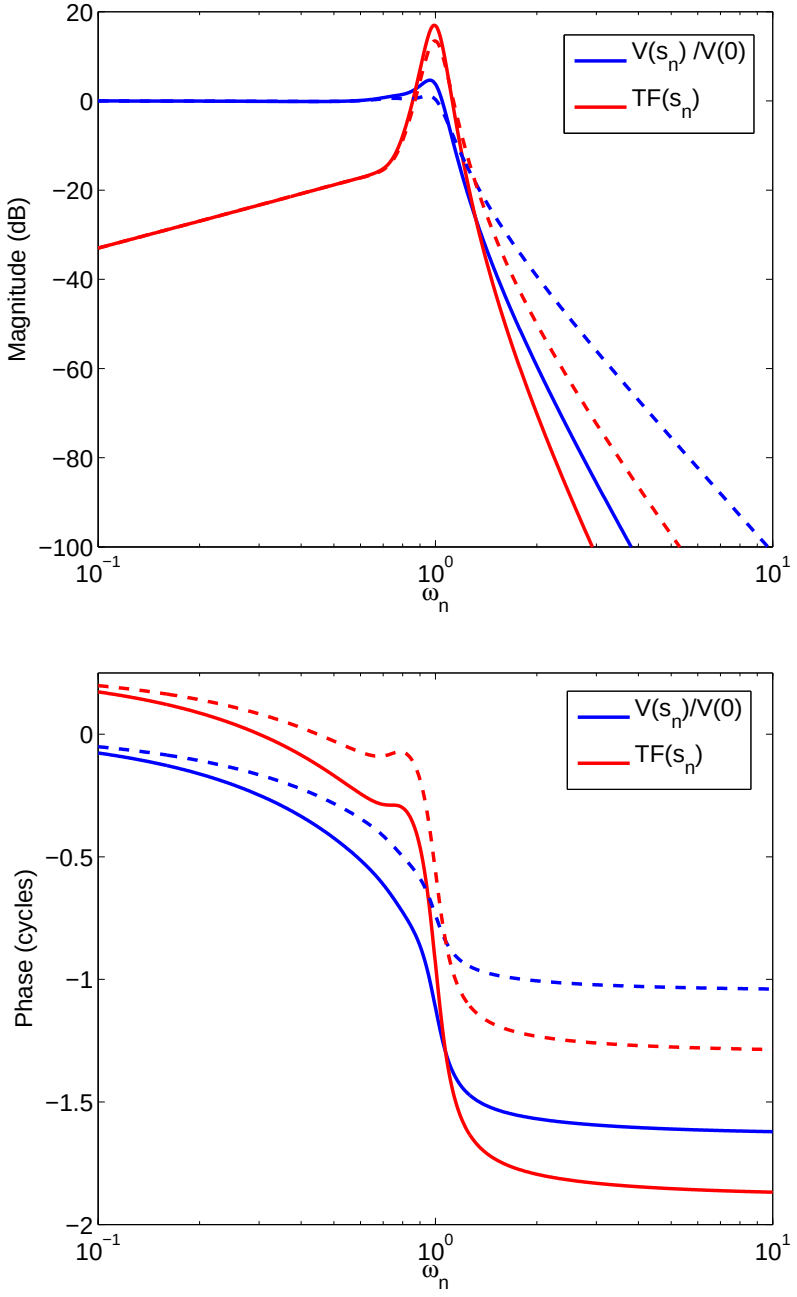


Figure 2-10: Magnitude and phase of WKB-type solutions to the bidirectional cochlea transfer function for $N_{nat} = 16$ (broken lines) and $N_{nat} = 24$ (solid lines).

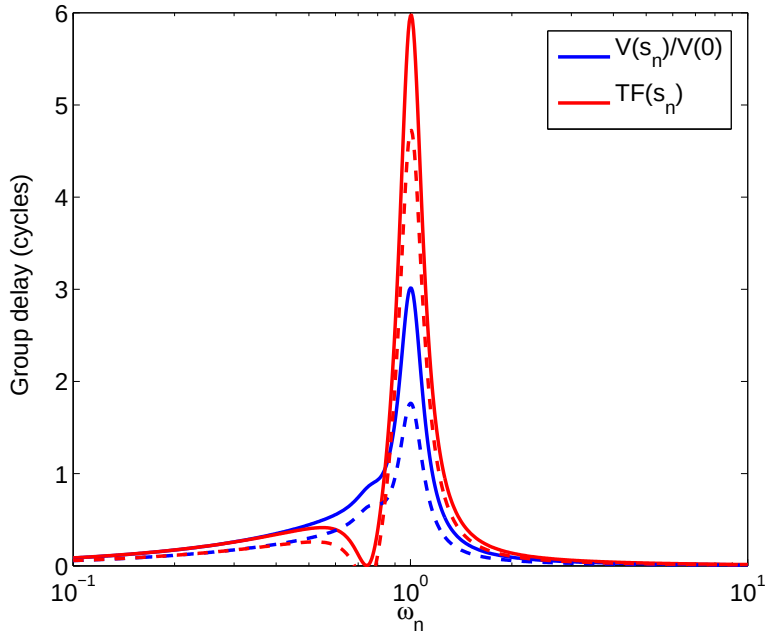


Figure 2-11: Group delay of WKB-type solutions to the bidirectional cochlea transfer function for $N_{nat} = 16$ (broken lines) and $N_{nat} = 24$ (solid lines).

Thus, L_t is given by:

$$L_t = \frac{Z_{in}}{\omega_{c,min}} = \frac{Z_{in} \exp\left(N/N_{nat}\right)}{\omega_c(0)} \tag{2.23}$$

In order for our lumped transmission line to closely approximate the original continuous line each stage should only change the phase of TF, i.e. $\int_0^{s_n} k_n ds'$, by a small amount. If this condition is not met the spatial discretization becomes too coarse, resulting in unwanted inter-stage reflections that show up as secondary peaks in the cochlear TF. In order to avoid such reflections we should have

$$\begin{aligned} |\Delta s_n \times k_n| &\ll 1 \\ \Rightarrow \left| \left(\frac{ds_n}{dx} \right) \Delta x \times k_n \right| &\ll 1 \\ \Rightarrow \frac{|s_n k_n|}{N_{nat}} &\ll 1 \end{aligned} \tag{2.24}$$

where Δs_n is the change in s_n due to a single stage, and we have used the fact that

$ds_n/dx = s_n/l$. Thus, inter-stage reflections increase as $|s_n| = \omega/\omega_c(x)$, the ratio of the input frequency to the best frequency at that location, increases. In other words, a fixed-frequency input tone will suffer increasing reflections as it propagates, since $\omega_c(x)$ decreases exponentially with increasing x.

For a given value of s_n, inter-stage reflections can be reduced by reducing $|k_n|$ and increasing N_{nat}. However, signal gain, i.e., $|TF|$, increases with $|k_n|$, as may be seen from equation (2.22). On the other hand increasing N_{nat} is undesirable because of increased chip area, power consumption, and output noise. The designer must compromise between these conflicting performance requirements.

By substituting $k_n = N_{nat}\sqrt{Z_n Y_n}/s_n$ and using the known values of Z_n and Y_n, the no-reflection condition in (2.24) can be rewritten as

$$\sqrt{Q_{line}}\left|\frac{s_n\sqrt{s_n^2+\mu s_n/Q+\mu^2}}{\mu\left(s_n^2+2ds_n+1\right)}\right| \ll 1 \tag{2.25}$$

Since the values of μ, Q and d are fixed, we must reduce Q_{line} to reduce inter-stage reflections. The quantity Q_{line} has a simple physical interpretation: it is the ratio of the amount of reactive energy stored within each stage, which is given by $L_0 I^2/2$, to the energy transferred per cycle (i.e., in a time $1/\omega_c(0)$) to the other stages. The latter quantity is given by $Z_0 I^2/\left(2\omega_c(0)\right)$, where I is the current along the line. Using (2.15), we can also rewrite Q_{line} in the suggestive form

$$Q_{line} = \frac{2\omega_c(0)}{\omega_{cutoff}(0)} \tag{2.26}$$

Here $\omega_{cutoff}(0) = 2/\sqrt{L_0 C_0}$ is the cutoff frequency of the lumped transmission line. Wave propagation on lumped lines is only possible at frequencies less than the cutoff frequency.

2.2.6 Frequency Scaling

It is well known from classical filter design theory that filter impedances and frequency responses can be easily scaled by scaling circuit parameters by constant ratios. This

property is important for the RF cochlea. Specifically, if a filter needs to be operated with a new load impedance Z_{new}, the following element transformations will leave the transfer function of the filter unchanged:

$$\begin{aligned} R_{new} &= R_{old} Z_r \\ L_{new} &= L_{old} Z_r \\ C_{new} &= \frac{C_{old}}{Z_r} \end{aligned} \tag{2.27}$$

where $Z_r = Z_{new}/Z_{old}$ is the impedance transformation ratio. Similarly, the frequency response of the filter can be scaled if

$$\begin{aligned} R_{new} &= R_{old} \\ L_{new} &= \frac{L_{old}}{f_r} \\ C_{new} &= \frac{C_{old}}{f_r} \end{aligned} \tag{2.28}$$

where $f_r = f_{new}/f_{old}$ is the frequency scaling ratio. We can use these transformations to scale the normalized cochlear admittance $Y_n(s_n)$ into real frequency space. In general, we expect the response of a lumped system to approach that of the continuous one it is trying to approximate as more and more elements are used, i.e., the continuous system is quantized to finer and finer levels of precision[6]. This is indeed true in the bidirectional cochlea. As N_{nat} is increased the transfer functions approach asymptotic values predicted by the continuous transmission line equations. If N_{nat} is too low the cochlear transfer functions are not smooth, and their roll-off slopes decrease, which reduces frequency selectivity. However, power consumption and layout area of the RF cochlea increase as N_{nat} increases. We therefore use the minimum value of N_{nat} that still gives adequate performance.

Once the value of N_{nat} has been decided, the frequency scaling ratio for the n-th stage in the bidirectional cochlea is given by

[6]In the cochlea, fineness of quantization is measured by the number of filter stages used over a fixed ratio of best (peak) frequencies f_1/f_2. An octave, i.e., $f_1/f_2 = 2$, is frequently used for convenience.

$$f_r(n) = \omega_c(0) \exp\left(-\frac{n}{N_{nat}}\right) \tag{2.29}$$

where n increases from the base to the apex. Each stage is also impedance scaled by the same ratio Z_r. We typically design our filters in impedance-normalized form. As a result $Z_{old} = 1\Omega$ and $Z_r = Z_0$, the constant impedance defined earlier.

2.2.7 Transformed Cochlear Models

In this section, we describe transformations of the basic bidirectional cochlear structure shown in Figure 2-6 that are suitable for integrated circuit implementation at RF frequencies. The main problem with Zweig's model from this perspective is that $Y_n \propto 1/s_n$ at high frequencies, i.e., as $s_n \to \infty$, Y_n looks inductive. However, the presence of substrate capacitances means that integrated circuit impedances inevitably look capacitive at high frequencies.

We can avoid this problem if we change the convention used to map mechanical and electrical domain variables to each other. In Zweig's original model, shown in Figure 2-6, the mapping is $(P, U) \to (V, I)$, where P is pressure across the scala media, U is volume velocity of the cochlear fluid and V and I are, respectively, voltage and current in the circuit. However, there is no reason for the alternate mapping $(P, U) \to (I, V)$ not to work. This mapping transforms impedances to admittances and vice versa, so we get the transmission line structure shown in Figure 2-12(a), where Z_1 is a complex series impedance, and the shunt admittance is simply a capacitance of value $C_1(x) = C_1(0) \exp(x/l)$ to ground. As before, the best frequency $\omega_c(x) = \omega_c(0) \exp(-x/l)$ decreases exponentially with position x. In order to keep wave number k_n and input impedance Z_{in} unchanged, we need

$$\begin{aligned} Z_{n1} &= Y_n Q_{line} \\ Y_{n1} &= Z_n / Q_{line} \end{aligned} \tag{2.30}$$

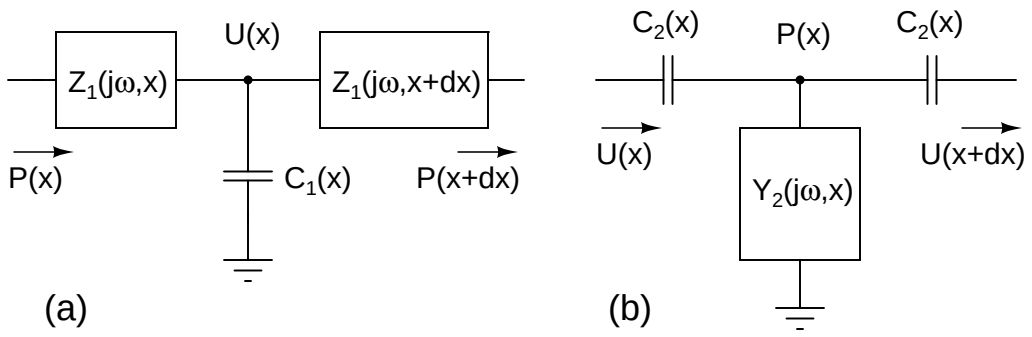


Figure 2-12: Transformations of Zweig's bidirectional cochlea model, obtained by (a) reversing the mechanical-to-electrical mapping convention and (b) through a low-pass to high-pass transformation.

where Z_{n1} and Y_{n1} are the impedance and frequency-normalized forms of Z_1 and C_1. The output variable in this structure is the dual of the shunt current dI/dx, i.e., the voltage across the series impedance, i.e., $V_{out} \equiv (\Delta x) \times (dV/dx)/V(0)$. Thus, the cochlear transfer function (TF) for this modified architecture is given by the normalized voltage drop across each series impedance Z_{n1}. The WKB solution for cochlear TFs is the same as in the original design (given by Eqns. 2.20 and 2.22), except that I and $I(0)$ are replaced by V and $V(0)$, where $V(0)$ is the volume velocity (voltage) at the base of the cochlea.

Since impedances and admittances are interchanged, the series $R-L$ termination network at the apex becomes a parallel $R-C$ network. We implemented this transformed bidirectional cochlear structure on chip, with the following parameter values: $\omega_c(0) = 2\pi \times 10 \times 10^9$rad/s, $C_1(0) = 70$fF, $N_{nat} = 20$, $Z_0 = 225\Omega$, $Q_{line} = 0.05$ and $Z_{in} = 50\Omega$. There are several advantages of this modified architecture for an integrated implementation:

- As the normalized frequency $s_n \rightarrow \infty$, $Z_{n1} \rightarrow 1/s_n$, i.e. it looks capacitive at high frequencies. In addition, shunt capacitances $C_1(x)$ are present by design at every node. This models what actually happens in integrated circuits at RF. This makes implementation easier. For example, parasitic capacitances can be easily absorbed into the design.

- Outputs from this cochlear architecture (the transfer functions) are differential

voltages and not shunt currents. Voltages are easier to sense and transmit off-chip than currents, especially at RF. The differential nature of the TF signal is also advantageous. Another advantage of this design is that the output, being the difference in voltage between two nodes, is resistant to unwanted common-mode signals on the ground node such as substrate noise.

- Gain control circuits that operate on voltage signals are easier to build at RF than those that use current, because RF currents are hard to sense. Building compressive nonlinearities into the transformed architecture will thus be simpler than with the original design.

- The layout area of our transformed architecture will be smaller than the original structure since it contains fewer inductors. Integrated inductors are expensive in terms of chip area. Therefore minimizing the number of inductors is desirable for any RF design.

Another potentially interesting cochlear architecture, which we have not implemented, emerges if we perform a low-pass to high-pass transformation on the circuit shown in Figure 2-6. Such a transformation occurs if we replace s_n by $1/s_n$ everywhere in the design. The resulting structure, shown in Figure 2-12(b), has shunt admittances $Y_2(j\omega, x)$ that are coupled together using series capacitances $C_2(x) = C_2(0)\exp(-x/l)$ that *decrease* exponentially with position x. This is a high-pass cochlea, i.e., the best frequency *increases* exponentially with x, i.e., $\omega_c(x) = \omega_c(0)\exp(x/l)$. We define the normalized frequency s_n in the same way as before. Y_2 can now be normalized into the dimensionless impedance $Y_{n2}(s_n)$, which is given by

$$Y_{n2}(s_n) = Y_n\left(\frac{1}{s_n}\right) = \frac{s_n\left(\mu^2 s_n^2 + \mu s_n/Q + 1\right)}{\left(s_n^2 + 2ds_n + 1\right)^2} \tag{2.31}$$

where $|\mu| < 1$. We can see that $Y_{n2}(s_n)$ looks similar to the original admittance $Y_n(s_n)$, except that the frequency axis s_n has been reversed. This behavior is the essence of the low-pass to high-pass transformation.

The high-pass cochlea reduces the total number of inductors required in the de-

sign, which is desirable. However, there are other implementation issues that get worse with this topology. For example, the highest frequencies have to propagate the longest distance (in contrast to the low-pass cochlea, where they peak close to the base). Since propagation loss, distributed effects and un-modeled parasitic poles become increasingly significant as the frequency increases, it makes sense to have these frequencies travel shorter, not longer, distances before they peak (i.e, reach their best places)[7].

Figure 2-13 shows simulated spatial responses obtained from the modified cochlear structure shown in Figure 2-12(a) for input frequencies varying logarithmically over two orders of magnitude. Curves that peak towards the base of the cochlea correspond to high frequency inputs. As the input frequency decreases, the spatial response patterns shift towards the right (i.e., the apex of the cochlea). For this simulation, we used $N_{nat} = 16$ filter sections per e-fold, and the parameters defining the series impedance Z_{n1} were given by $Q = 3.8$, $\mu = 0.76$ and $d = 0.1$.

2.3 Signal Analysis

In this section we briefly discuss the general theory of signal representation. The information contained in a signal can be represented, or indexed, in various ways. In general, we want to find an indexing that represents information of interest to us in an easily extracted fashion. Spectrum analysis is a common type of representation, indexed by frequency content, that is useful in a large variety of applications.

2.3.1 Introduction

The two most basic characteristics of a dynamical variable, or signal, are time and frequency. The former measures *when* events that convey information occur, while the latter measures the *rate* at which they occur. A complete representation of any signal is obtained by dividing its time-frequency plane into cells of unit area

[7] In other words, all physical systems, including biological ones, have frequency responses that are inherently low-pass in nature.

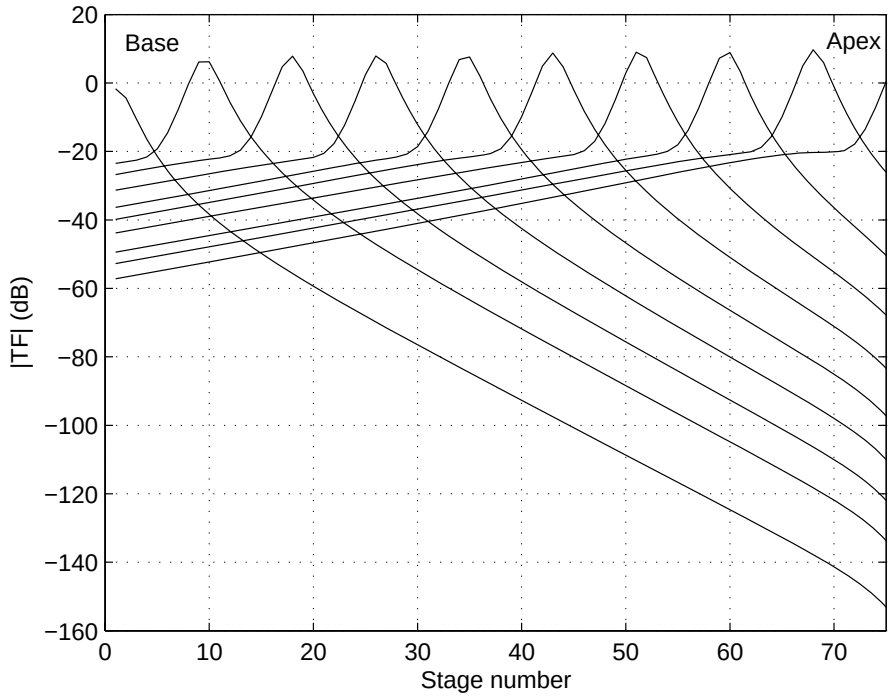


Figure 2-13: Simulated spatial response of the bidirectional RF cochlea for input frequencies varying logarithmically over two orders of magnitude.

$\Delta\omega\Delta t = 1$, where ω and t represent frequency and time, respectively [77, 78]. Each cell contains two degrees of freedom, which we may think of as amplitude and phase. By "complete", we mean that the representation contains enough information about the signal for exact reproduction in the absence of noise. This statement is an example of the well-known Shannon-Nyquist sampling theorem. According to the theorem, a signal of bandwidth B and length T in time is completely characterized by $2BT$ samples. The area of the signal's time frequency plane is BT, and we may imagine that each sample characterizes an area $BT/(2BT) = 1/2$ on the plane. Thus each cell of area 1 corresponds to two samples, or more generally, two degrees of freedom. In practice, both quantities can only be obtained with finite precision due to noise.

We have seen that quantizing the time-frequency plane into cells of unit area results in two degrees of freedom per cell. In other words, the amount of information in the time-frequency domain is limited to $2BT$ degrees of freedom for any noiseless

signal[8]. Each degree of freedom is a scalar measurement, and the $2BT$ measurements define an unique vector within a vector space of dimension $2BT$.

Lossless transformations, such as Fourier transforms, can be used to reversibly map a signal's $2BT$ degrees of freedom to another set of numbers. There may be $2BT$ or more numbers in this set; in the latter case, the transformed representation is *redundant*. When the members of the set represent (in some sense) different frequencies present in the signal the transformation is referred to as *spectrum analysis* or *decomposition*. Consider transformations that map the $2BT$ degrees of freedom into an equal number of coefficents, and assign two to each of the BT cells on the time-frequency plane. Signals with the highest possible localization in this plane have only one non-zero coefficient, thus localizing them to a single cell with unit area. As a result, the minimum possible uncertainty in the location of any signal on the plane, as quantified by the area within which it can be localized, is 1. The classical time-frequency trade-off, or uncertainty principle, that formalizes this result was first derived by Dennis Gabor in 1946 [77]. It states that

$$\sigma_\omega \sigma_t \geq \frac{1}{2} \tag{2.32}$$

where σ_ω and σ_t are defined as the second moments (standard deviations) of the measured probability distributions of the frequency and occurence time of the signal, respectively. This result is related to the energy-time uncertainty principle of quantum mechanics via the identity $E = \hbar\omega$. The minimum value of $1/2$ is reached for Gaussian pulses, which have the same shapes along both time and frequency axes.

The basic problem faced by the familiar Fourier transform while analyzing non-stationary signals, such as speech, is that time-domain information is embedded in the phases of the complex sinusoids that comprise the transformed representation. As a result, it is difficult to interpret and recover such information. In other words, complex sinusoids have compact support along the frequency axis, but not the time axis. Transformations that avoid this problem, and generate well-separated time

[8]In the absence of noise, of course, the signal's amplitude and phase can be used to convey as much information as we please.

and frequency-domain information, can be separated into two broad classes: time-frequency and time-scale representations. We shall consider only real signals in the following discussion, but it can easily be extended to include complex signals as well.

2.3.2 Time-Frequency Representations

The commonest time-frequency representation (TFR) is the windowed, or short-time Fourier transform (STFT), defined as

$$X(t,\omega) = \int_{-\infty}^{\infty} x(\tau)\gamma(\tau - t)e^{-j\omega\tau}d\tau \tag{2.33}$$

where $\gamma(t)$ is known as the window function, or kernel. The window function is chosen to have compact support along the time axis, i.e., to be close to zero except over a finite length of time. This property allows signals to be localized in time. The Gabor transform is a special example of the STFT, and uses a Gaussian window. The STFT generates time-frequency plots known as *spectrograms*, and is an example of a linear transformation. Equation (2.33) can be rewritten as

$$X(t,\omega) = e^{-j\omega t}\int_{-\infty}^{\infty} x(\tau)\gamma(\tau - t)e^{-j\omega(\tau-t)}d\tau \tag{2.34}$$

When rewritten in this form it becomes clear that the STFT is equivalent, up to a phase factor, to convolution of the signal $x(t)$ with the function $\gamma(t)e^{-j\omega t}$. In other words, $X(t,\omega)$ is produced by passing $x(t)$ through a linear, time-invariant (LTI) filter with impulse response $\gamma(t)e^{-j\omega t}$. We shall denote a Fourier transform pair as $x \leftrightarrow X$. It is easy to show that $\gamma(t)e^{-j\omega t} \leftrightarrow \Gamma(\Omega + \omega)$ when $\gamma(t) \leftrightarrow \Gamma(\Omega)$, i.e., $\Gamma(\Omega)$ is the frequency response corresponding to the impulse response $\gamma(t)$. In other words modulation of $\gamma(t)$ by the complex sinusoid $e^{-j\omega t}$ translates its frequency response by an amount ω along the frequency axis[9]. If $\Gamma(\Omega)$ has a bandpass response with center frequency 0 and bandwidth B_0, $\Gamma(\Omega+\omega)$ has a center frequency of $-\omega$ and the same bandwidth. Thus the STFT $X(t,\omega)$ of a signal $x(t)$ is equivalent to applying

[9]This property is the underlying principle behind the operation of all mixers and superheterodye systems.

bandpass filters with varying center frequency ω and constant bandwidth B_0 to the signal, and collecting together their outputs.

The outputs of a filter bank, i.e., a collection of filters with finite bandwidths, constitute a time-frequency representation of their common input signal. The STFT is one way in which such a filter bank may be synthesized. The normal Fourier transform corresponds to an STFT with no windowing, i.e., with $\gamma(t) = 1$. Because the impulse responses are constant with time, this case corresponds to a filter bank with infinitesimally narrow bandwidths. Long windows, i.e., impulse responses that decay slowly with time, result in narrowband analysis filters that have poor timing resolution, but good frequency resolution. The reverse is also true, as predicted by the uncertainty principle. In general, the optimal window function for analyzing a given signal depends on the type of information we want to extract from it. Moreover, finding this optimal function is not always straighforward.

An alternative time-frequency representation is the Wigner-Ville distribution. This is an example of a nonlinear (specificially, quadratic) transformation, and is the simplest example of a class of distributions know as the Cohen class. We begin by finding the *instantaneous* autocorrelation function of the signal. This function is a generalization of the normal autocorrelation function, and is defined as

$$R_{xx}(t,\tau) = x\left(t+\tau/2\right) x\left(t-\tau/2\right) \tag{2.35}$$

where $x(t)$ is the signal. We see that R_{xx} depends both on time t and time lag τ. We then take the Fourier transform of R_{xx}, but only along the τ axis. The result is a time-frequency representation, i.e., a function of both frequency and time. The Wigner-Ville distribution has finite support in both time and frequency (unlike the STFT), which is one of its advantages. However, the multiplicative, or nonlinear nature of the transform results in mixing, i.e., introduces frequency components that did not exist in the orignal signal. It also has poor noise properties. The instantaneous autocorrelation function can be windowed before being Fourier transformed, resulting in other members of the Cohen class that have better mixing and noise properties.

An alternate class of time-frequency representations is based on *reassignment* of the (t, ω) plane to *instantaneous* time and frequency, respectively [84]. The new plane is denoted by (t_{ins}, ω_{ins}), where

$$\begin{aligned} t_{ins}(\omega, t) &= t - \frac{\partial \phi}{\partial \omega} \\ \omega_{ins}(\omega, t) &= \frac{\partial \phi}{\partial t} \end{aligned} \tag{2.36}$$

Here ϕ is the phase of the signal. Reassigning both amplitude and phase information to this new plane allows reconstruction of the original signal. This representation has several advantages. Firstly, it is biologically plausible since it does not require ω and t to be explicitly estimated, but only the derivatives of ϕ. In the case of sound representation, for example, the time derivative can be estimated by measuring the intervals between action potentials on single phase-locked auditory nerve fibers. Similarly, the frequency derivative can be estimated by measuring intervals between adjacent tonotopically-mapped fibers. In addition, the resultant representation is sparse, i.e., most of the (t_{ins}, ω_{ins}) plane is zero, thus allowing signal compression.

Finally, the reassigned representation localizes simple signals with greater precision than that imposed by the uncertainty principle, i.e., (2.32). For example, tones (single frequencies) and clicks (single times) map to horizontal and vertical lines of zero width in the (t_{ins}, ω_{ins}) plane, unlike in the STFT, where they map to Gaussian blurs in the (t, ω) plane. This property reveals features of signals that are hidden in spectrograms but may be perceptually significant. However, it does *not* violate the uncertainty principle, because there is an important distinction between *precision* and *resolution*. Precision refers to the ablity to localize single signals, while resolution refers to the separation of two or more signals. In its strict, or true sense the uncertainty principle only applies to resolution, not precision. In fact, the Wigner-Ville transform also achieves greater precision than predicted by (2.32).

2.3.3 Time-Scale Representation

The canonical example of a time-scale representation is wavelet analysis. In wavelet analysis a time-domain signal is not divided into time segments by a windowing function, but into "scale segments" with a "probing function" known as a wavelet. All wavelets are time-scaled and time-translated versions of a single "mother" wavelet function ψ. The time-scale a and translation b of the mother wavelet function are varied, and the scaled and translated function is correlated with the signal. The result measures how similar the signal is to the wavelet at that scale or translation, and is quantified by wavelet coefficients that are functions of a and b. Mathematically, the mother wavelet is a function of the scaled and translated time variable $(t-b)/a$:

$$W_x(b,a) = |a|^{-1/2} \int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right) dt \tag{2.37}$$

We see that signals in a wavelet representation occupy regions in the (a,b) plane, not the time-frequency plane. Plots of the function $W_x(a,b)$ are known as *scalograms*.

Let the wavelet represent the impulse response of a bandpass filter. The time-scale of this response is proportional to a. However, the time and frequency scales of Fourier transforms are inversely related. It can easily be shown that $\psi(t/a) \leftrightarrow a\Psi(a\omega)$ when $\psi(t) \leftrightarrow \Psi(\omega)$, i.e., $\psi(t)$ and $\Psi(\omega)$ are Fourier transform pairs. In other words the time-scaled impulse response $\psi(t/a)$ corresponds to the frequency-scaled filter transfer function $\Psi(a\omega)$, where the frequency-scale is equal to $1/a$. Thus both the center frequency and bandwidth of the filter are proportional to $1/a$.

Equation (2.37) may be rewritten as a convolution of the signal $x(t)$ with the time-reversed and scaled wavelet $\psi(-t/a)$. Thus the wavelet transform (WT), like the STFT, is equivalent to filtering of the signal $x(t)$ by a bank of linear bandpass filters. Since the center frequency and bandwidth of the filters are both proportional to $1/a$, they have constant *fractional* bandwidth, i.e., constant Q. As a result the WT can also be viewed as spectrum analysis by a constant-Q filter bank. This behavior is in contrast to that of the STFT, where the analysis filter is translated along the frequency axis, but not scaled. Thus, the STFT, as explained earlier, can be viewed

as analysis by a constant-bandwidth filter bank.

The time-frequency uncertainty principle shown in (2.32) is fundamental, and is applicable to both time-frequency and time-scale representations. The actual value of $\sigma_\omega \sigma_t$ depends on the detailed shapes of the analysis filters used to localize the signal. It is invariant on the (t, ω) plane when a single prototype filter is translated or scaled along the frequency axis to perform the analysis, as in the STFT and the WT. In this case we get $\sigma_\omega \sigma_t = \alpha/2$, where $\alpha \geq 1$ is a constant. We can rewrite this equation as follows:

$$\left(\frac{\sigma_\omega}{\omega}\right)\left(\frac{\sigma_t}{T}\right) = \frac{\alpha}{4\pi} \tag{2.38}$$

where ω and T are the mean frequency and time period of the sigal, respectively, and we have used that fact that $\omega T = 2\pi$ by definition. Thus the product of the relative, or fractional uncertainties in frequency and time is also constant. In wavelet representations (or constant-Q filter banks) σ_ω/ω is constant and equal to the inverse of the Q. Thus the relative timing precision T/σ_t is also constant in this case. In particular, it is invariant with the mean frequency $\omega = 2\pi/T$. Thus constant-Q analysis results in constant precision on all time scales. On the other hand, in constant-bandwidth analysis σ_ω is constant, resulting in a relative timing precision that is inversely proportional to the mean frequency ω. In other words, at high center frequencies the constant-bandwidth filters have high Q, resulting in long settling times and poor timing precision.

2.3.4 Redundant Representations

Redundant representations use more than the $2BT$ linearly-independent basis vectors necessary to represent a signal. As a result, they generalize the notion of bases to expansions over sets of vectors that are linearly dependent. Such sets are mathematically known as *frames*. STFT and WT representations, and filter banks in general, are examples of frame representations. Redundancy in the frame representation aids reconstruction of the original signal, especially in the presence of noise. Frames

are particularly useful when higher-order information (features) must be reliably extracted from a noisy signal with unreliable hardware. This situation is common in biological sensory systems [246]. Such systems achieve robustness by using many unreliable variables, such as the firing rates of a large population of neurons, to reliably represent a signal. In general, frame representations distribute precision over multiple redundant variables, and are therefore most naturally processed by computational architectures, such as neural networks, that employ the collective analog paradigm.

The continuous wavelet transform (CWT) shown in (2.37) is highly reduntant because the scalogram $W_x(a, b)$ is defined for all real values of a and b. The discrete wavelet transform (DWT) restricts the variations in scale a and translation b, usually to powers of 2, to obtain a less redundant representation. The fact that reconstruction is possible from the DWT is a consequence of the Shannon-Nyquist sampling theorem, as discussed earlier. The relationship between the CWT and the DWT is analogous to that between the continuous and discrete Fourier transforms (CFT and DFT, respectively)[10].

2.3.5 Cochlear Signal Analysis

The time-frequency representations generated by the cochlea have been estimated using linear traveling-wave models [125]. Such models are useful, but can only approximate the nonlinear behavior of the biological cochlea. For example, they do not account for the presence of phase-locked action potentials in auditory nerve fibers. However, such models do show that the linear cochlea behaves as a nearly ideal time-frequency analyzer, with values of α that vary between 1.2 and 1.6 [77, 125].

As described earlier, Zweig's cochlear model assumes perfect scaling symmetry. As a result, it predicts cochlear transfer functions that are only dependent on the normalized frequency variable $s_n = j\omega/\omega_c(x)$, where $\omega_c(x)$ is the local center frequency. Transfer functions at different locations are produced by changing $\omega_c(x)$, i.e., the frequency scale. Thus the outputs of the model resemble a constant-Q filter bank, where

[10]The well-known fast Fourier transform (FFT) refers to one of a class of related, efficient algorithms for calculating the DFT.

the cochlear transfer functions act as analysis filters and the time-scaling parameter $a = 1/\omega_c(x)$. However, the measured transfer functions of the human cochlea only have constant-Q responses for center frequencies above $\approx$ 300Hz. The Q decreases for center frequencies lower than this value, approaching constant-bandwidth behavior at very low frequencies [206]. Thus scaling symmetry is only approximately satisfied by the biological cochlea.

The representation of sounds in the cochlea is highly redundant. The cochlear transfer functions are relatively broadband, with $\approx 1/3$ octave bandwidth ($Q \approx 4.3$). As a result, only about $10/(1/3) = 30$ transfer functions are mathematically sufficient to represent frequencies within the 10-octave human heaing range. However, the human cochlea has approximately 3,500 sensors (inner hair cells) and 35,000 outputs (nerve fibers). Action potentials on the 10 nerve fibers connected on average to one hair cell cooperatively encode the amplitude and phase of sounds sensed by that cell. As a result, it may be argued that these fibers should not be treated as separate outputs. We can therefore assume that the cochlear representation consists of about 3,500 variables, namely the basilar membrane velocities sensed by each inner hair cell. The result is a *redundancy ratio* R of $3,500/30 \approx 100$.

In general each variable in the cochlear representation is corrupted by noise. In the simplest case each noise source is assumed to be independent of the others. In this case the cochlea provides R-fold redundancy, i.e., R independent estimates of each analysis window in the time-frequency plane. As mentioned earlier, redundancy increases both robustness to hardware failures, and also the accuracy of signal reconstruction. More precisely, it can be shown that the magnitude of the signal reconstruction error is proportional to $1/\sqrt{R}$ [246]. Of course, there is no free lunch, and such improved performance comes at the cost of increases in hardware complexity and power consumption.

At low and moderate sound intensities the cochlea is well described by a redundant constant-Q filter bank, as described earlier. The auditory system performs signal processing on the cochlear outputs to reliably detect higher-level features in sound signals. For example, peaks and edges in the sound spectrum and interaural time de-

lays are detected with very high precision. Such properties remind us of the contrast enhancement and binocular capabilities of the visual system. In fact, there are many analogies between visual and auditory signal processing, suggesting that common neural architectures are used in both systems [268]. However, such analogies have not always been recognized. In fact, several well-known models of auditory signal processing are based on temporal algorithms with no analogs in visual signal processing. Such algorithms involve the computation of correlation functions and absolute frequencies, and require the presence of specialized neural machinery, such as delay lines and oscillators. No unambiguous evidence of such machinery has been found, casting doubt on such models.

An alternative, and conceptually simpler, approach explicitly acknowledges the similarities between vision and audition. Fundamentally, the cochlear traveling wave transforms temporal features in sound signals into spatial features on auditory nerve fibers. This transformation allows such features to be extracted by spatially-distributed neural networks that are similar to well-known ones in the visual system. Such networks consist of two main functional blocks: lateral inhibition for computing spatial derivatives, and coincidence detection for computing correlations. Mathematically, the former function is linear and requires adders and subtractors, while the latter is nonlinear and requires multipliers. Several common auditory tasks can be easily implemented using these functions, as listed below [268]:

- Features of a sound spectrum can be extracted by lateral inibition networks in the cochlear nucleus. Such networks calculate spatial derivatives and enhance edges and peaks in the spectrum.

- The periodicity, pitch, or missing fundamental in a sound can be perceived using a matrix of coincidence detectors. The outputs of the matrix are instantaneous cross-correlation functions, which are large when the inputs are harmonically related.

- Sounds can be localized azimuthally by cross-correlating the spatial output profiles of the two cochleas with a matrix of coincidence detectors. This process

is known as stereausis. Interaural time delays result in instantaneous differences in the spatial profiles which are detected by the matrix.

- Multiscale spatio-temporal profile analysis can be performed by neurons with complex spatio-temporal response functions. A variety of such cells are found in the auditory cortex.

2.4 Spectrum Analysis Algorithms

2.4.1 Swept-Sine Analyzers

Most commercial RF spectrum analyzers are of the swept-sine or superheterodyne type. The basic structure of superheterodyne analyzers is shown in Figure 2-14, and their operation is well summarized in [286]. A tunable local oscillator (LO) source is mixed with the input signal after it has been attenuated and pre-filtered. The LO frequency is swept over a frequency range equal to the frequency span covered on the display. The output of the mixer (the IF) thus sweeps over the same frequency range. This signal is passed through a variable gain IF amplifier and then bandpass filtered by an IF filter. The bandwidth of the IF filter is variable and is known as the resolution bandwidth (RBW) of the analyzer. The output of the RBW filter is passed through a logarithmic compressor to reduce its dynamic range. Its envelope is now detected, low pass filtered by a video filter and sent to the CRT display. The bandwidth of the video filter is known as the video bandwidth (VBW) and is usually (but not always) greater than the RBW.

In principle, the IF frequency could either be lower or higher than the highest frequency expected at the input. In practice, a high-side IF is almost always used, i.e. the IF frequency is always higher than the input frequency (see Figure 2-15). If a low-side IF is used instead, the IF frequency will be equal to some allowable input frequency. This signal will pass through the mixer independent of LO tuning and produce a DC output. The result is a hole in the frequency response where the output amplitude is not a function of the LO frequency. In addition, the allowable

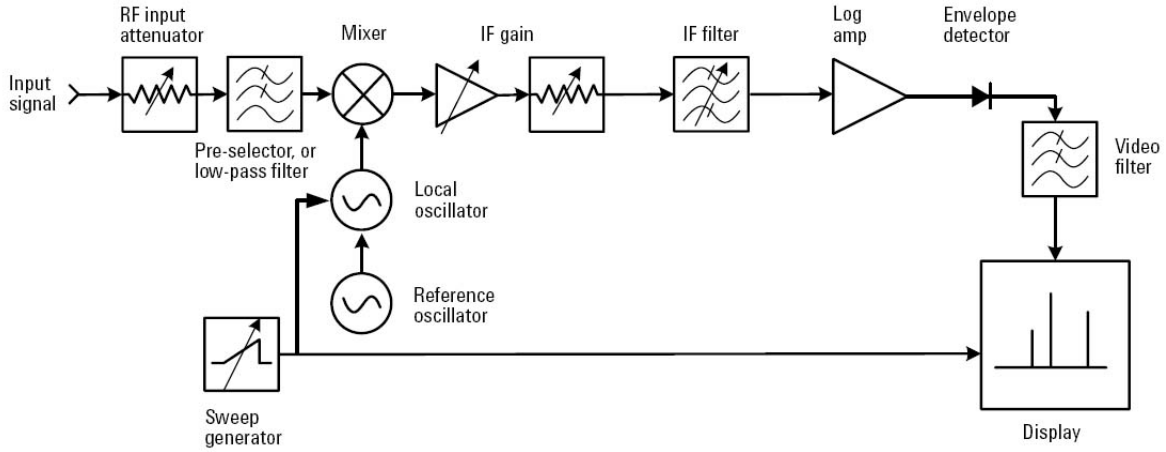


Figure 2-14: Block diagram of a classic superheterodyne spectrum analyzer (figure adapted from [286]).

input frequency range has to be limited by the input pre-selection filter. This is just an image reject filter; it removes unwanted high frequency components that would otherwise be down-converted into IF by the mixer.

We see that this spectrum analyzer architecture is similar in many ways to a superheterodyne radio receiver. The main difference is that a tunable LO is used; radio receivers usually have a fixed LO frequency. We should also note that most high frequency spectrum analyzers, just like radio receivers, use two or more stages of mixing and frequency down-conversion. The final IF frequency is usually a few MHz. We are interested in determining the sweep time of a swept-tuned spectrum analyzer. The way in which such analyzers trade-off frequency resolution and sweep time has long been known [316]. The time spent by a spectral component (output from the mixer) in the passband of the IF filter is given by

$$Time\ spent\ in\ passband = ST\frac{RBW}{B} \tag{2.39}$$

where ST is the sweep time, RBW is the resolution bandwidth of the analyzer and B, also called *span*, is the frequency range of the sweep. Also, the rise, or settling time of a filter is inversely proportional to its bandwidth, so

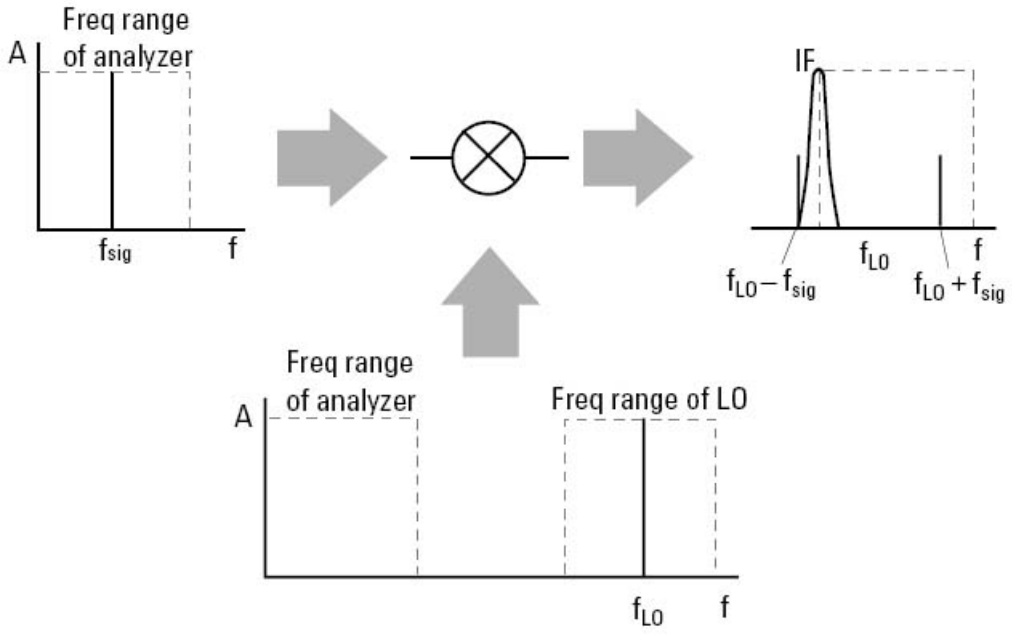


Figure 2-15: The LO must be tuned to $f_{IF} + f_{sig}$ to produce a signal on the display (figure adapted from [286]).

$$Rise\ time = \frac{k}{RBW} \tag{2.40}$$

where k is a constant (in the 2-3 range for typical Gaussian analog IF filters)[11]. In order for a spectral component to be displayed accurately, it must be present in the filter passband for a time greater than the filter rise time. Thus we may equate the two times in (2.39) and (2.40) to get the minimum possible sweep time

$$ST = \frac{kB}{(RBW)^2} \tag{2.41}$$

The swept-since analyzer resolves the spectrum (of bandwidth B) into frequency "bins" of bandwidth RBW. Thus there are $N = B/(RBW)$ such bins. Equation (2.41) can thus be rewritten as

[11]The product of the settling time and the bandwidth of an analog filter is constant. Fundamentally time and bandwidth trade-off with each other. Since quantum mechanics says that energy is proportional to frequency, the trade-off is ultimately a consequence of the time-energy uncertainty principle $\Delta E \Delta t \geq \hbar$.

$$ST = \frac{kN^2}{B} \tag{2.42}$$

We see that the sweep time scales as N^2. However, commercial spectrum analyzers today have greatly improved sweep times over that suggested by (2.42). The main reasons are the use of digital IF filters and the FFT. Digital RBW filters typically allow sweep speed increases by factors of 2-4. More dramatic speed increases occur when an FFT is performed to directly obtain the spectrum of the IF signal, as shown in Figure 2-16. This approach is only feasible over narrow frequency spans (10MHz maximum is typical), but allows very narrow RBW's without long sweep times. For example, an FFT over a frequency span of 1KHz with a RBW of 10Hz effectively analyzes the signal using 100 10Hz filters in parallel. If there was no overhead involved in the digital processing, this would decrease sweep time by a factor of 100 over a purely analog solution.

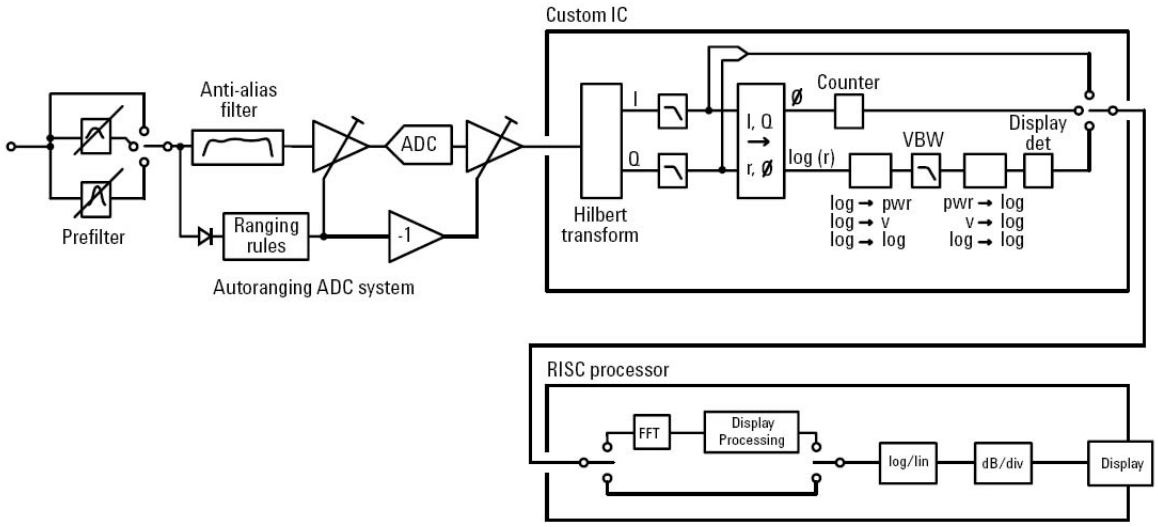


Figure 2-16: Typical all-digital IF architecture used in modern spectrum analyzers (figure adapted from [286]).

The displayed noise floor of a conventional superheterodyne spectrum analyzer depends on the RBW of the IF stage - as the RBW decreases, the noise floor decreases. The displayed average noise level (DANL) is given by

$$Noise\ Floor = NF \times \gamma kT\,(RBW) \tag{2.43}$$

where NF is the noise figure of the superheterodyne receiver in the spectrum analyzer and $\gamma > 1$ is the ratio of the equivalent noise bandwidth of the IF filter to the 3dB resolution bandwidth (RBW). A well designed modern commercial RF spectrum analyzer may have $NF \approx 20dB$. Typical values of γ for analog Gaussian filters range from 1.05 to 1.13. The dynamic range of a spectrum analyzer is limited by three factors: the broadband noise floor (sensitivity) of the system, the phase noise of the local oscillator (LO) and the distortion and gain compression performance of the input mixer.

2.4.2 Other Spectrum Analysis Algorithms

Figure 2-17 graphically illustrates three other types of spectrum analyzers, including the cochlea. To first order, the cochlea can be modeled as a transmission line where shunt admittances Y model sections of the BM, while the series impedances Z are inductors modeling fluid coupling. The values of Y and Z per unit length increase exponentially with position x [324, 328], i.e., $\propto \exp(x/l)$, where l is a constant that characterizes the length scale on which cochlear properties vary from the basal ($x = 0$) to the apical end [328]. The transfer function $TF(x, \omega)$ of the cochlea is defined as the normalized current that flows through $Y(x)$ in response to a input tone with frequency ω [328]. It models the velocity of the BM. At a given position, the magnitude of the TF slowly increases with frequency, reaches its maximum value at a frequency $\omega_c(x) = \omega_c(0)\exp(-x/l)$, known as the center frequency, and then rapidly decreases.

In order to model the continuous cochlear transmission line with a finite number of components we spatially discretize it by lumping sections of line Δx long into individual stages. We assume Δx is constant; as a result the stages have exponentially-spaced center frequencies. The number of stages per e-fold in center frequency is given by

$$N_{nat} = \frac{l}{\Delta x} \tag{2.44}$$

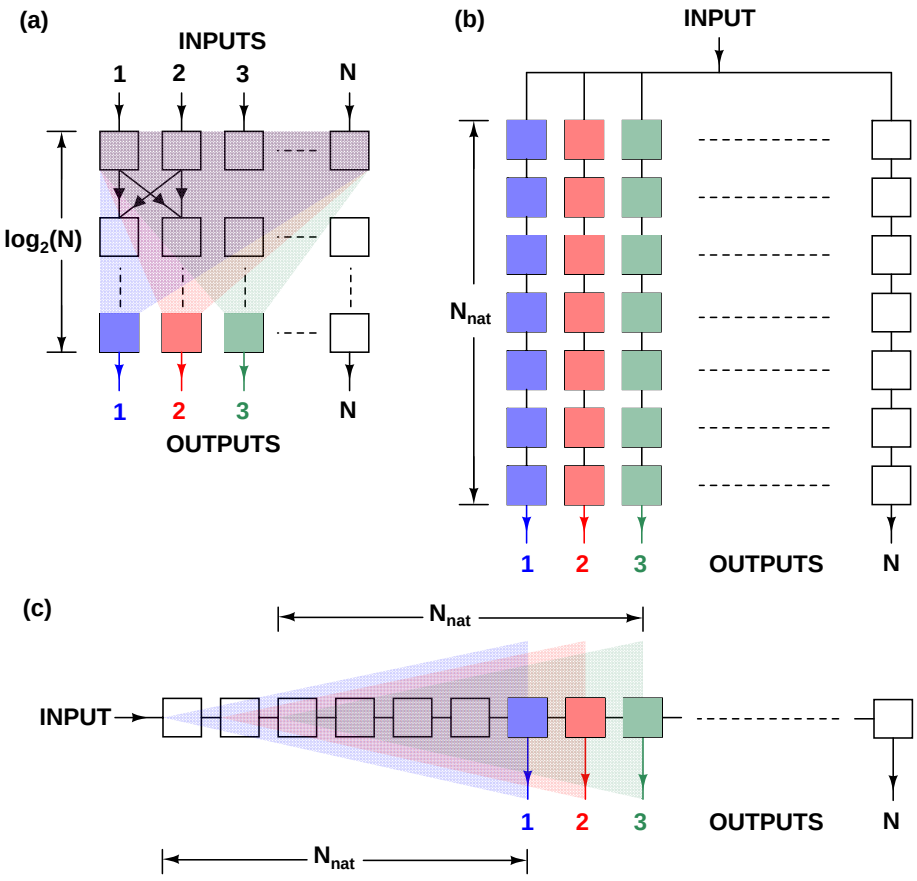


Figure 2-17: Comparison of the spectral analysis algorithms of (a) the FFT, (b) a parallel bank of independent filters and (c) the cochlea. Blocks represent two-input multiply-and-add units in the FFT, elementary filters in the filter bank and cochlear stages in the cochlea. The 'triangular' sliding windows in the cochlea illustrate that cochlear transfer functions are created by contributions from approximately N_{nat} filters basal to that output. Therefore only one new stage needs to be added to create each new output.

The exponentially-tapered structure of the cochlea ensures that the TF at any position is produced by a "sliding window" of the approximately N_{nat} stages basal of that position. Thus any cochlear TF is well approximated as a cascade of N_{nat} identical stages [259]. Each TF is sharply peaked around its center frequency ω_c, and thus selects a frequency "bin" centered about ω_c.

Consider the response of the cochlea to an input tone at $\Delta\omega + \omega_c$ at a location where the center frequency is ω_c. We assume that $|\Delta\omega| \ll \omega_c$, and that each cochlear stage is a linear, second-order, all-pole filter with transfer function H_s and quality factor Q. The magnitude of the cochlear transfer function is then given by

$$\begin{aligned} |TF(\Delta\omega_n)| &\approx |H_s(\Delta\omega_n)|^{N_{nat}} \\ &= \frac{TF(0)}{\left[1 + (2Q\Delta\omega_n)^2\right]^{N_{nat}/2}} \end{aligned} \tag{2.45}$$

where $\Delta\omega_n = \Delta\omega/\omega_c$. Equation (2.45) can be used to define the effective quality factor $Q_{eff} \equiv 1/(2\Delta\omega_{res})$ of the cochlear transfer function, where $\Delta\omega_{res}$ is the 3dB bandwidth, i.e., the value of $\Delta\omega_n$ at which $|TF(\Delta\omega_n)| = TF(0)/\sqrt{2}$. We find that Q_{eff} is given by

$$Q_{eff} = \frac{Q}{\sqrt{2^{1/N_{nat}} - 1}} \tag{2.46}$$

When $N_{nat} \gg 1$ (as is usually the case), the equation above may be simplified to

$$Q_{eff} \approx Q\sqrt{\frac{N_{nat}}{\ln(2)}} \tag{2.47}$$

Thus, in this case we get $Q_{eff} \propto \sqrt{N_{nat}}$. To summarize: According to this highly simplified analysis, the cochlea synthesizes a bank of bandpass filters with exponentially-spaced center frequencies. There are N_{nat} filters per e-fold in center frequency, and each filter has a quality factor of Q_{eff}.

At values of $\Delta\omega_n$ that are significantly larger than 1, we find that $|TF(\Delta\omega_n)| \propto 1/(\Delta\omega_n)^{2N_{nat}}$. In other words, the frequency response rolls off asymptotically with a

slope equal to that of a filter of order $2N_{nat}$. The frequency resolution of the cochlea is ultimately set by the sharpness of these high-frequency roll-off slopes, which, for a given signal-to-noise ratio (SNR), sets the minimum frequency ratio that can be discriminated by adjacent cochlear stages.

The analysis time τ_i of a spectrum analyzer is defined as the time taken to resolve the i-th frequency bin. In the cochlea, the analysis time is equal to the sum of the settling times of stages basal of i, which is approximately equal to $QN_{nat}/\omega_c(i)$. Therefore the analysis time for the whole spectrum is on the order of QN_{nat} cycles of the lowest analyzed frequency. In addition, the total number of stages is given by $N = N_{nat}\,(\ln(\beta) + 1)$, where β is the ratio of maximum and minimum analyzed frequencies, and the '1' accounts for the fact that the very first cochlear output needs an extra N_{nat} stages basal to it. Thus, for a given value of β, we get $N \propto N_{nat}$, which implies that $\tau_i \propto N$. In order to make the shape of TF independent of i we implement cochlear stages as frequency-scaled versions of a common prototype [177]. As a result, the hardware and power requirements of the cochlea also scale as the total number of filters, i.e. N.

A parallel bank of constant-Q, independent filters can also be used to decompose a signal into exponentially-spaced frequency bins. In order to get frequency resolution similar to the cochlea each independent filter must have order $2N_{nat}$. Such filters can be formed by cascading N_{nat} second-order filter stages, as shown in Figure 2-17. There are N such filters, which, unlike in the cochlea, are not shared between outputs. Thus the hardware cost, as measured by the total number of second-order filter stages, scales like $N \times N_{nat} \propto N^2$ if β is fixed. However, the time taken for spectrum analysis in the filter bank is given by the sum of the settling times of the N_{nat} sections in each filter, which scales like N. One advantage of such independent filter banks over the cochlea is *flexibility*. In a filter bank any given filter only participates in creating a single transfer function. Thus any transfer function can be easily modified without affecting the others. The traveling-wave structure of the cochlea results in filters being reused, which dramatically improves hardware efficiency. However, this increase is achieved at the cost of reduced flexibility. A single cochlear filter affects many transfer

Table 2.2: Spectral analysis algorithms: Performance summary

Algorithm	**Analysis time**	**Hardware cost**
Swept-sine	$O\left(N^2\right)$	$O(1)$
FFT	$O\left(N \log N\right)$	$O(N \log N)$
Filter bank	$O(N)$	$O\left(N^2\right)$
Cochlea	$O(N)$	$O(N)$

functions, making individual transfer functions harder to tune.

The output bins of both the cochlea and parallel filter banks are available and updated in parallel, which allows them to continuously monitor the whole spectrum. This behavior is in contrast to most commercial RF spectrum analyzers, which are of the swept-sine or super-heterodyne type. In this type of analyzer a single frequency bin is sampled and updated at a given time, causing aliasing of non-stationary spectra. The sampling rate scales as $1/N^2$, i.e., the time to analyze the whole spectrum scales as N^2 [316]. However, the hardware requirements for this type of analyzer are independent of N, i.e., $O(1)$.

The Fast Fourier Transform (FFT) uses constant-bandwidth frequency bins, unlike the cochlea and constant-Q parallel filter banks. It takes $O(N \ln(N))$ time (measured by the number of multiply-and-add operations) and uses $O(N \ln(N))$ hardware (measured by the number of multipliers and adders) to perform spectrum analysis. Thus, it appears that the cochlear spectral analysis algorithm delivers the most efficient trade-off between analysis time and hardware cost, as summarized in Table 2.2. It exploits the scale-invariant nature of an exponential to achieve $O(N)$ scaling in both quantities.

2.4.3 Why an RF Cochlea?

The "RF cochlea" is an integrated circuit that uses ideas from the biological cochlea to perform fast, parallel, low-power spectrum analysis at RF frequencies. We implemented the RF cochlea on silicon using a standard CMOS process. It has several appealing properties:

Algorithm: The cochlear algorithm uses a traveling-wave to synthesize a set

of exponentially-spaced transfer functions with extremely high roll-off slopes, thus performing real-time spectral analysis over a wide range of frequencies with scale-independent resolution. Alternative algorithms use banks of parallel, independent filters to obtain the same resolution. Transfer functions for the filters within such auditory filter banks are usually derived from physiological data using the technique of reverse correlation [51]. In this technique recordings from auditory nerve fibers are used to find the average input waveforms that trigger action potentials. Such waveforms can be viewed as the impulse responses of cochlear transfer functions, which, as noted previously, are sharply-tuned asymmetric bandpass filters. Analytical approximations of these experimental responses include the well-known Gammatone, Gammachirp and One-Zero Gammatone filters (OZGF) [138]. The OZGF, in particular, seems to strike a good balance between biological realism and ease of implementation using analog hardware.

Traveling waves have two distinct advantages over such filter banks for performing spectral analysis: improved temporal resolution and hardware reuse [256, 259]. The broadband cochlear traveling wave stages settle quickly compared to the high-order, high-Q filters used within filter banks, resulting in improved temporal resolution. In addition, by using each stage multiple times to create closely-spaced transfer functions, the cochlea significantly reduces hardware complexity and power consumption. The analysis time and hardware requirements of the cochlea both scale as N, the number of output frequency bins, which, as noted previously, is efficient compared to both independent filter banks and the FFT.

Parallelism: The RF cochlea is similar to a filter bank in that it monitors the complete spectrum in real time. Unlike the swept-sine spectrum analyzer, it does not use a spectral scanning technique that monitors a given patch of spectrum only at discrete intervals of time. This property makes the cochlea uniquely suitable for capturing brief transient events, which may be important for surveillance applications.

Noise: The RF cochlea is likely to have significantly less noise, and thus higher dynamic range, than silicon cochleas that have been implemented at audio frequencies. This is because integrated passive inductors can be used for the RF cochlea. In

contrast, audio frequency cochleas built in silicon must use active elements to synthesize inductors. Such active inductors consume power, suffer from limited linear range and develop high levels of noise. The performance of active inductors is analyzed in an appendix to this chapter.

Biological Inspiration: The RF cochlea is a "biologically inspired" system. In this context, the phrase refers to systems that *adapt* interesting ideas from biology in order to meet challenging engineering goals [254]. We should emphasize that our goal is not to be bio-mimetic, or model the biological system accurately. As a result, the RF cochlea does not aim to capture all, or even most, of the complex behavior of the biological cochlea. Some of the features exhibited by biology that our RF cochlea does try to replicate are

- Frequency-to-place transformation, i.e. spectral analysis
- Sharp roll-off slope after cutoff
- Extended dynamic range due to active amplification
- Distributed gain control using a compressive nonlinearity to large signal amplitudes
- Broadening of the frequency-response curves as the input amplitude is increased (Q control)
- Masking of adjacent frequencies due to gain and Q control
- Asymmetric attack and release to transient inputs

Finally, the RF cochlea is also an example of a complex analog signal processing system that uses a smart algorithm to reduce power consumption and improve dynamic range. Building it presents us with an opportunity to explore the design and control of large integrated mixed-signal systems with many interacting components.

2.5 On-Chip Implementation

2.5.1 Performance Limits

In this section, we describe a target RF cochlea implementation that seems reasonable based on what we have discussed so far. The ambient RF power spectrum between 30MHz and 3GHz was measured using a broadband antenna and commercial spectrum analyzer, and is shown in Figure 2-18. This plot may be treated as representative of the real-world environment that the RF cochlea will sense[12]. The figure shows that FM and TV broadcast signals, which have historically been exempt from radiated power limits, dwarf the rest of the spectrum.

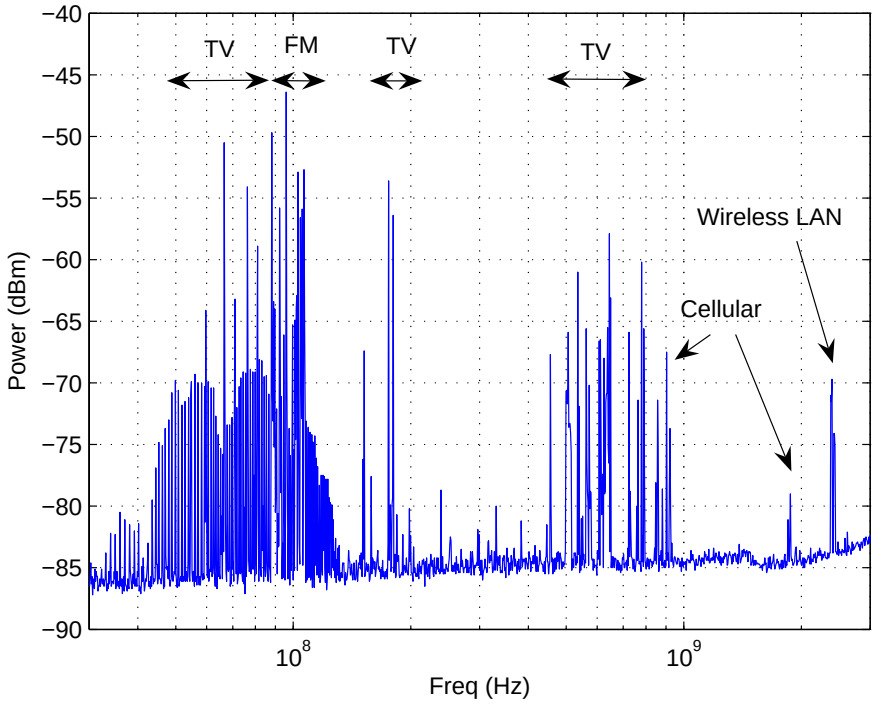


Figure 2-18: Typical radiated RF power profile. The data shown was measured with a broadband antenna and commercial spectrum analyzer, and represents what one may expect to receive indoors in typical urban areas.

In order to demonstrate the concept of the RF cochlea we would like to build one that operates over at least one decade (10:1) in frequency. We also need to implement

[12]Data for this plot was obtained at the author's lab bench, located within building 38 on the M.I.T. campus.

an additional half-octave at the beginning (high-frequency end) to allow the wave to build up. Thus we need to implement about four octaves of filtering. We want the working cochlea to end no lower than about 250MHz. In this way we avoid picking up lower frequency TV signals (VHF channels 2-13, 54-216MHz) and broadcast FM (88-108MHz). Otherwise weak high-frequency signals of greater interest to us would be swamped by these strong low-frequency interferers.

On-chip implementation places further constraints on the design. Integrated inductors and transformers are one limiting factor. In a constant-Q structure like the cochlea the values of these passive components scale as $1/\omega_c$, where ω_c is the center frequency. This type of scaling guarantees that the impedance at the center frequency is invariant with position. In addition, the layout area of an inductor scales rougly as the square of its value. Thus, the chip area, which is usually dominated by the inductors, scales as $1/\omega_c^2$. This sharp increase in chip area with decreasing ω_c limits the lowest operating frequency of the RF cochlea.

Parasitic inductances and capacitances within passive components scale roughly with the value of the component. Therefore the *relative* effect of such parasitics is relatively invariant with ω_c. On the other hand, transistors are used to create negative resistances. The values of these resistances are constant with ω_c, and so transistor sizes (and parasitics) do not scale with ω_c. Thus, the relative effect of such parasitics increases as the passive components become smaller with increasing ω_c. As a result, the highest operating frequency is limited by transistor parasitics, and should improve with process scaling. However, such improvement cannot continue indefinitely. Eventually the bandwidth of the package used to mechanically protect the chip will limit the highest operating frequency.

We selected a 0.13μm CMOS process, because it provided a good compromise between price, performance and availability. A practical upper limit for ω_c in this process appears to be approximately 10GHz. Therefore we propose a four-octave RF cochlea design operating between 10GHz and 625MHz.

2.5.2 Synthesizing the Cochlear Impedance

In this section, we describe how to synthesize circuits that provide the right organ of Corti impedance functions (i.e., Z_n and Z_{n1}) for the bidirectional RF cochlea. Network synthesis is in general a hard problem. Classical synthesis methods [292] are typically iterative: circuit complexity is added in stages. However, much of the difficulties inherent in these analytical techniques (such as Brune and ladder synthesis) can be avoided by utilizing existing knowledge about the likely final form of the synthesized circuit. In our case, for example, it is known that two coupled resonators are a good model for the impedance of the organ of Corti [173].

Our synthesis approach is thus to pick a coupled-resonator topology, such as the one shown in Figure 2-19, and then find the right component values. The circuit in Figure 2-19 has two parallel resonant tanks which are coupled together by the capacitance C_c and mutual inductance M. The input impedance Z_{in} of this particular topology turns out to be particularly useful for modeling the normalized series impedance Z_{n1} in the modified cochlear architecture shown in Figure 2-12(a). For example, it has the right high frequency behavior: as $\omega \to \infty$, Z_{in} looks capacitive. In general, Z_{in} for this circuit can be written in the normalized form

$$Z_{in}(s_n) = \frac{s_n^4 + a_3 s_n^3 + a_2 s_n^2 + a_1 s_n + a_0}{s_n (s_n^2 + b_1 s_n + b_0)} \tag{2.48}$$

where the coefficients a_n and b_n $(n = 0, 1, 2...)$ are functions of the element values R_1, L_1, C_1, C_c, M, R_2, L_2 and C_2. The obvious way to complete the design is to first analytically find the input impedance Z_{in} in terms of the element values[13]. The functions that relate element values to the coefficients of $Z_{in}(s_n)$ are known once this computation has been completed. Given the desired values of these coefficients (obtained from the known impedance Z_{n1}), a series of simultaneous equations can now be written. Unfortunately, they are nonlinear and solving them to get component values is extremely difficult. If the circuit topology being considered cannot produce the desired immittance function, no solutions will exist, and either the topology or

[13]The result is usually a few pages of extremely nasty algebra. However, at least the equations are linear and, *in principle*, can always be solved by hand to produce an unique solution.

the function must be modified).

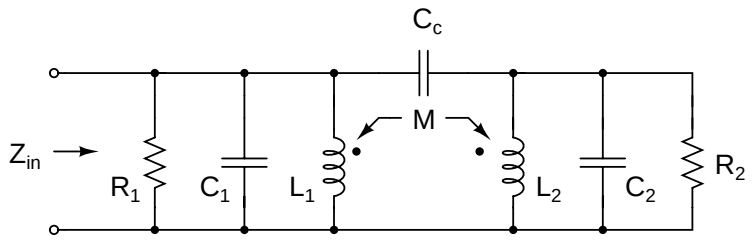


Figure 2-19: Coupled resonator circuit used for synthesizing the impedance Z_{n1}.

Numerical optimization using a computer is a better way to solve this synthesis problem. We have written a program in Mathematica (Wolfram Research, Champaign, IL) that can solve for R_1, L_1, C_1, C_c, M, R_2, L_2 and C_2 given the nodal equations of the circuit, target coefficients for Z_{in} and certain constraints.

The routine accepts a given network topology as input and finds a set of component values (R, L, C, M) that realizes the symbolically-specified, rational driving-point impedance or transfer function. Since the network is assumed to be linear and uses a finite number of lumped components, all driving-point impedances, transfer impedances and transfer functions can be written as ratios of polynomials in s, the frequency variable:

$$F(s) = \frac{a_Z}{b_P}\left(\frac{s^Z + \sum_{i=0}^{Z-1}\frac{a_i}{a_Z}s^i}{s^P + \sum_{i=0}^{P-1}\frac{b_i}{b_P}s^i}\right) \tag{2.49}$$

where P and Z are the number of poles and zeros, respectively. The optimization routine minimizes an objective function E that quantifies differences between the network function $F(s)$ realized by the synthesized network and $F_0(s)$, the ideal or desired function. The function E is given by

$$E = \sum_{i=0}^{Z-1}\left(\frac{a_{i,0} - a_i}{a_{i,0} + a_i}\right)^2 + \sum_{i=0}^{P-1}\left(\frac{b_{i,0} - b_i}{b_{i,0} + b_i}\right)^2 \tag{2.50}$$

Since network synthesis is, in general, a one-to-many problem, the routine finds one of an infinite set of possible solutions that minimize the value of E. However, the set of possible solutions can be restricted by imposing additional constraints on the

component values. For example, we restricted the sizes of the two inductors in the transformer to be within 20% of each other. This condition allows similarly-sized coils to be used to realize them, maximizing coupling $k = M/\sqrt{L_1 L_2}$ for a given layout area. We also typically used the following additional constraints:

- $(L_1, C_1, C_c, L_2, C_2) > 0$
- $|M| < k_{max}\sqrt{L_1 L_2}$

where $0 < k_{max} < 1$ is the maximum allowable coupling coefficient between the inductors L_1 and L_2. It is easy to show that the target impedance function Z_{n1} is not positive real, and cannot be realized as the driving-point immittance of a purely passive network (the properties of impedance functions are described in an appendix to this chapter). However, allowing negative (i.e., active) resistors removes this restriction and allows immittance functions that are not positive real to be synthesized. As a result, we have not restricted the signs of R_1 and R_2. It turns out that only R_1 actually needs to be negative in order to synthesize Z_{n1}. Figure 2-20 shows simulated spatial responses of the bidirectional RF cochlea when the synthesized circuit was used to realize the series impedance Z_{n1}. The normalized circuit parameters that were used are: $L_1 = 0.88$H, $L_2 = 0.61$H, $M = 0.50$H, $C_1 = 0.52$F, $C_2 = 0.60$F, $C_C = 2.62$F, $R_1 = -1.3\Omega$ and $R_2 = 1.79\Omega$. The resultant impedance closely matches Z_{n1} with $d = 0.1$, $\mu = 0.76$ and $Q = 3.8$.

Static compressive nonlinearities in R_1 (the only active element in the design) will cause $|R_1|$ to increase as the signal amplitude increases. Such nonlinearities occur naturally in most implementations of negative resistors. Figure 2-20 shows how the cochlea transfer functions change as R_1 is varied about its nominal value $R_n = -0.375$. We see that the peak gain of the cochlea decreases as $|R_1|$ increases. This means that some of the gain control necessary to extend the dynamic range of the RF cochlea shall occur automatically. Interestingly, the peak gain also decreases as $|R_1|$ decreases below R_n. This is because one of the pairs of complex poles in Z_{in} moves to the right half plane as $|R_1|$ decreases far below its nominal value, thus destroying the collective

amplification of the cochlea[14].

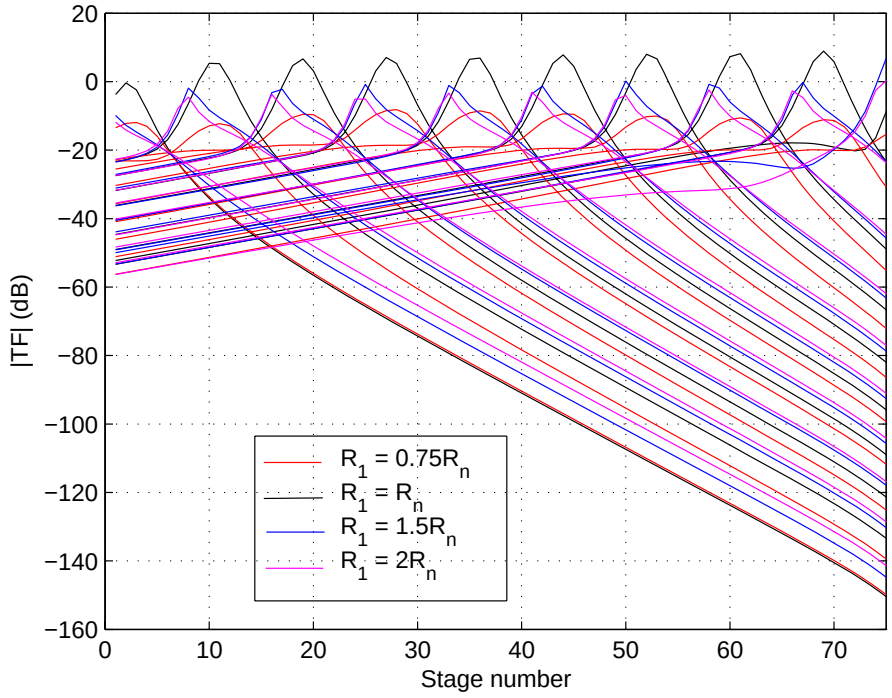


Figure 2-20: Simulated spatial responses of the bidirectional RF cochlea using the synthesized circuit. Different values of R_1 were used to simulate the effects of gain compression.

2.5.3 Negative Resistance Elements

Minimizing the number of active circuit elements in the cochlea is important since such elements are inherently nonlinear and add noise. In the previous section, we have described how the cochlear impedance function $Z_{n1}(s_n)$ can be synthesized using the minimum possible number of actives (a single negative resistor per section). We must now design a circuit to generate this negative resistance. Fortunately, this is relatively easy at RF frequencies. Several well-known circuits exist; these shall now be discussed.

The cross-coupled differential pair (shown in Figure 2-21(a)) is among the simplest.

[14]The two complex pole pairs in Z_{n1} play the same role as the complex zero pairs in Z_n from Zweig's original architecture: they are important for increasing gain through collective amplification.

During differential small-signal operation, the source node V_s may be considered an incremental ground. The input admittance Y_{in} is then given by

$$Y_{in}(s) = -\frac{g_m}{2} + s\left(\frac{C_{gs}}{2} + 2C_{gd}\right) \tag{2.51}$$

where g_m is the small-signal transconductance of each transistor, and C_{gs} and C_{gd} are parasitic gate-source and gate-drain capacitances, respectively. This leads to the equivalent circuit for Y_{in} shown in Figure 2-21(a).

The next circuit that we consider is the common gate transistor with inductive source degeneration shown in Figure 2-21(b) (DC biasing details have been omitted). The input admittance Y_{in} looking into the source is given by

$$Y_{in}(s) = (g_m + sC_{gs})\frac{1 + s^2 L_g C_{gd}}{1 + s^2 L_g (C_{gs} + C_{gd})} \tag{2.52}$$

The frequency response of Y_{in} is therefore given by

$$Y_{in}(\jmath\omega) = (g_m + \jmath\omega C_{gs})\frac{1 - \omega^2 L_g C_{gd}}{1 - \omega^2 L_g (C_{gs} + C_{gd})} \tag{2.53}$$

From Eqn. 2.53, the real part of $Y_{in}(\jmath\omega)$ is negative for $1/\sqrt{L_g (C_{gs} + C_{gd})} < \omega < 1/\sqrt{L_g C_{gd}}$, and positive otherwise. Therefore this circuit can be used to implement a negative resistance over a range of frequencies.

It is well known that capacitively loaded source followers frequently suffer from poor stability. This is because the input impedance of the circuit has a negative real part for certain frequencies. This effect can be utilized to create negative resistances. Consider the source follower shown in Figure 2-21(c). The input admittance Y_{in} looking into the gate is given by

$$Y_{in}(s) = \frac{sg_m C_{gd} + s^2\left[(C_{gd} + C_{gs})C_L + C_{gd}C_{gs}\right]}{g_m + sC_{gs}} \tag{2.54}$$

The frequency response $Y_{in}(\jmath\omega)$ for the circuit is therefore

$$Y_{in}(\jmath\omega) = \frac{-\omega^2 \left(C_{gd} + C_{gs}\right) C_L + \jmath\omega \left(g_m C_{gd} + \omega^2 C_{gs} \frac{\left(C_{gs}+C_{gd}\right)C_L + C_{gd}C_{gs}}{g_m} \right)}{\sqrt{g_m^2 + \omega^2 C_{gs}^2}} \tag{2.55}$$

Eqn. 2.55 shows that this circuit can also provide negative resistance. The final circuit that we want to discuss uses coupled inductors to synthesize a negative resistor [276]. The basic idea is to increase the Q of one inductor by coupling in energy from a secondary, coupled inductor. Consider the circuit shown in Figure 2-21(d). The current in the secondary, i_2 is designed to be an amplified and phase-shifted version of the primary current i_1, i.e., $i_2/i_1 = A\exp(j\theta)$. The input impedance Z_{in} looking into the primary is given by

$$Z_{in} = R_1 + \jmath\omega L_1 + \jmath\omega M \frac{i_2}{i_1} \equiv R_{eff} + \jmath\omega L_{eff} \tag{2.56}$$

where the effective input resistance and inductance R_{eff} and L_{eff} are given by $R_{eff} = R_1 - \omega M A \sin\theta$ and $L_{eff} = L_1 + MA\cos\theta$. By choosing suitable values of A and θ (close to $\pi/2$, for example), we can make R_{eff} negative over a certain range of frequencies. The upper end of this range is determined by parasitic capacitances, which were not included in this analysis. Of the four negative-resistance circuits we have discussed so far, the cross-coupled differential pair appears to be the most promising. It is the only circuit where the negative resistance is at least nominally independent of the operating frequency. This makes the design of a broadband system like the cochlea easier. The linear range of this circuit is the same as a normal differential pair, and is given by $V_L = I_{bias}/g_m$. Thus, the negative resistance seen across the terminals is approximately independent of the differential voltage between them as long as this voltage is less than V_L.

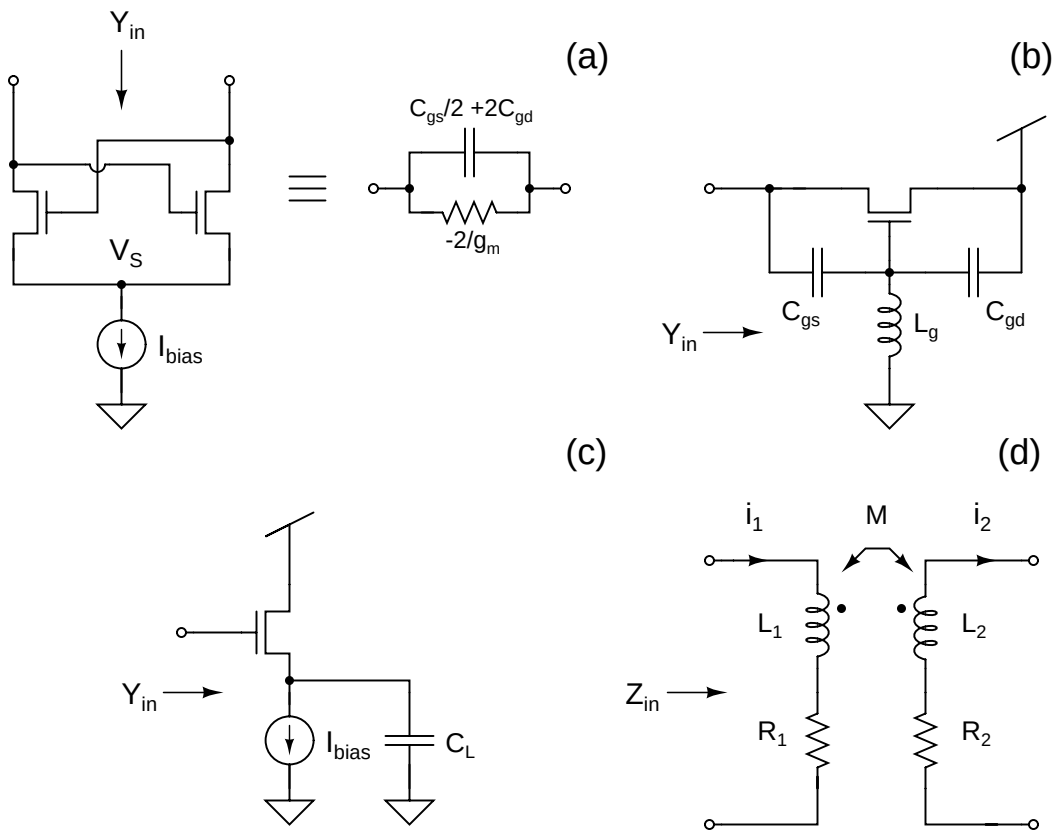


Figure 2-21: Simple circuits that generate negative resistance, i.e., $\Re[Z_{in}] < 0$ for some frequencies: (a) cross-coupled differential pair, (b) inductively gate-degenerated transistor, (c) capacitively-loaded source follower and (d) coupled inductors with feedback.

2.6 Theoretical Analysis

2.6.1 Transfer Functions

In this section we theoretically analyze the behavior of the bidirectional cochlea. We wrote a MATLAB program that can evaluate spatial responses in sinusoidal steady state of discrete (lumped) transmission lines with any number of stages and arbitrary series and shunt impedances. The only assumption we made was linearity. This program can be used to check the accuracy of the WKB solution, and also design the bidirectional cochlea. For example, we can study the effects of choosing different admittance functions, i.e., changing the functional form of Y_n in Zweig's original model. The program uses the generic finite-difference equations for voltage and current on a

lumped transmission line:

$$\begin{aligned} I(n+1) &= I(n) - V(n)Y(n) \\ V(n+1) &= V(n) - I(n+1)Z(n) \\ V_{out}(n) &= V(n+1) - V(n) \end{aligned} \tag{2.57}$$

where $V(n)$, $I(n)$ are the line voltage and current at the n-th stage, $V_{out}(n)$ is the differential voltage across the n-th stage (the discrete analog of dV/dx), and $Z(n)$ and $Y(n)$ are the series impedance and shunt admittance within this stage. This set of equations can be solved iteratively if the initial conditions, i.e., $V(0)$ and $I(0)$, are known. However, since the system is assumed to be linear, only ratios of voltages and currents (transfer functions) are important, not their actual values. Therefore, in practice it is sufficient if the ratio $V(0)/I(0)$ is known, i.e., set by a known impedance.

The impedance at any position n, denoted by $Z_{in}(n)$, can also be found iteratively. First, we decompose the transmission line into two segments, one extending from location n to the apex (low-frequency end), and the other from n to the base (input, or high-frequency end). The input impedances of these segments are given by $Z_{in+}(n)$ and $Z_{in-}(n)$, respectively, and can be found using the following equations, which are easily derived:

$$\begin{aligned} Z_{in+}(n+1) &= Z(n) + \left(Y(n) + \frac{1}{Z_{in+}(n)} \right)^{-1} \\ Z_{in-}(n-1) &= \left(Y(n-1) + \frac{1}{Z(n-1) + Z_{in-}(n)} \right)^{-1} \end{aligned} \tag{2.58}$$

The equations above can be solved when the termination impedances at the base and apex of the cochlea, i.e., $Z_{in-}(1)$ and $Z_{in+}(N)$, are known, where N is the total number of stages. The impedance at n is simply $Z_{in}(n) = Z_{in-}(n)||Z_{in+}(n)$. The input impedance of the entire cochlea, i.e., $Z_{in+}(1)$, which we shall denote simply by Z_{in} for convenience, is usually set to 50Ω for compatibility with RF test equipment.

Also, (2.14) predicts that $Z_{in} = Z_0\sqrt{Q_{line}}$. Thus, Z_0 is fixed once Q_{line} is known.

Figure 2-22 shows the simulated input impedance as a function of frequency for various values of Q_{line}. We see that it is real and equal to the designed value of 50Ω at most frequencies. However, frequencies close to $\omega(0)$, the center frequency of the first stage, see a larger impedance. This behavior is expected: it is how voltage gain is produced in the cochlea. In fact, we see that the maximum value attained by the real part of Z_{in} for such frequencies increases with Q_{line}, which suggests that the peak gain of the cochlear transfer functions should increase with Q_{line}.

In addition, frequencies around $0.03 \times \omega(0)$, which is the center frequency of the apical end of the cochlea, see an input impedance that varies rapidly with frequency. Notice that Z_{in} at such frequencies can have a negative real part, which is always an indicator of potential instability. This behavior is not limited to our cochlear model, and in fact is a problem for the biological cochlea as well. It is caused by the fact that simple apical termination networks cannot provide a good impedance match to $Z_{in-}(N)$ when it changes rapidly with frequency, i.e., at frequencies with best places close to the apex.

Figure 2-23 shows simulated cochlear transfer functions at two different locations ($n = N_{nat}$ and $n = 2N_{nat}$) for different values of N_{nat}. We see that the peak gains of the transfer functions are invariant with N_{nat}, while the center frequencies and bandwidths decrease slightly. This behavior differs from the WKB solution, according to which the peak gain should increase with N_{nat} (see Figure 2-10). This discrepancy is not surprising, since the WKB approximation breaks down around the peak, as mentioned previously. More importantly, the high frequency roll-off slope of the transfer functions is $20N_{nat}$dB/decade, i.e., equal to that of a filter of order N_{nat}. This behavior agrees with that predicted by the WKB solution. We show later that, for a given signal-to-noise ratio (SNR), the frequency resolution of the cochlea improves as the roll-off slope increases, i.e., as N_{nat} increases. In addition, we see that coarse spatial discretization (very low values of N_{nat}) leads to transfer functions that are not smooth functions of frequency. Both effects limit the minimum allowable value of N_{nat}; typical values range from 12 to 24.

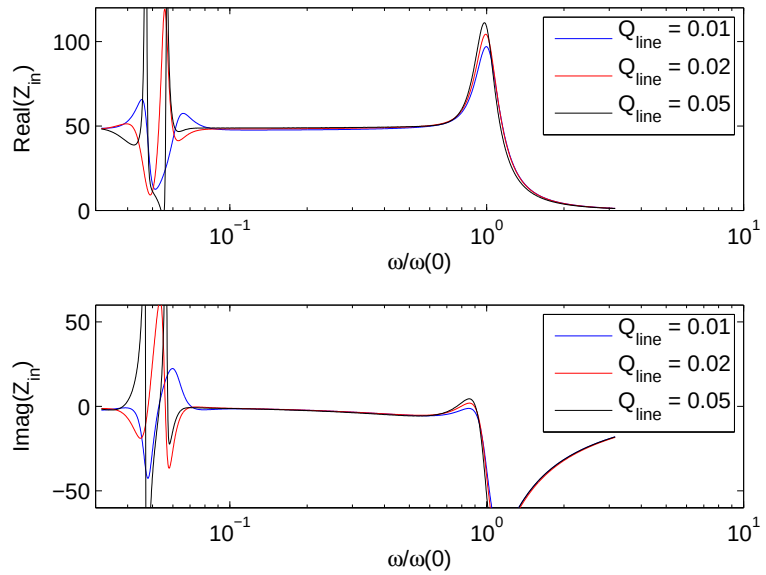


Figure 2-22: Simulated input impedance of the bidirectional cochlea as a function of frequency at various values of Q_{line}. The following parameters were used to draw this plot: $d = 0.15$, $\mu = 0.76$, $Q = 3.8$ and $N_{nat} = 24$.

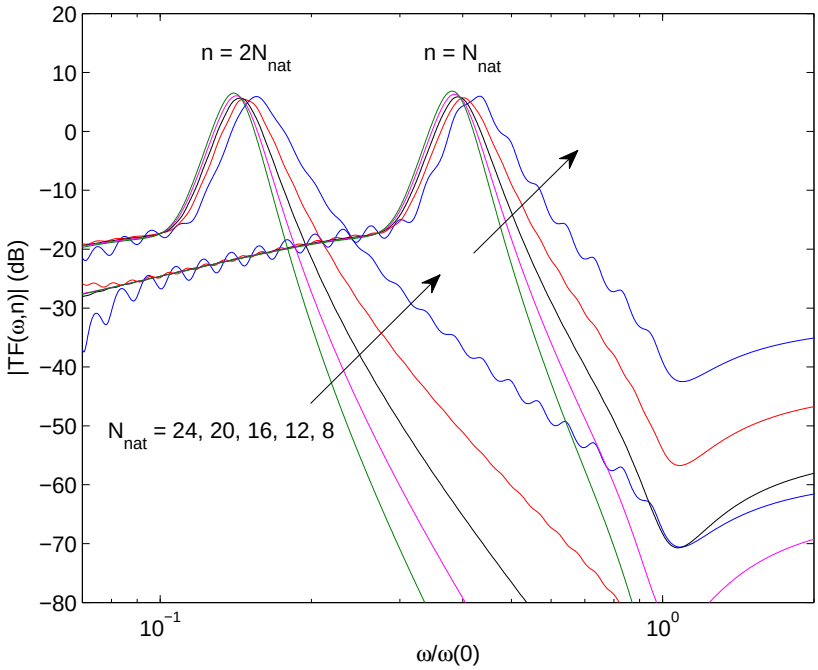


Figure 2-23: Bidirectional cochlea transfer functions as a function of frequency at various values of N_{nat}. The following parameters were used to draw this plot: $d = 0.15$, $\mu = 0.76$, $Q = 3.8$ and $Q_{line} = 0.05$.

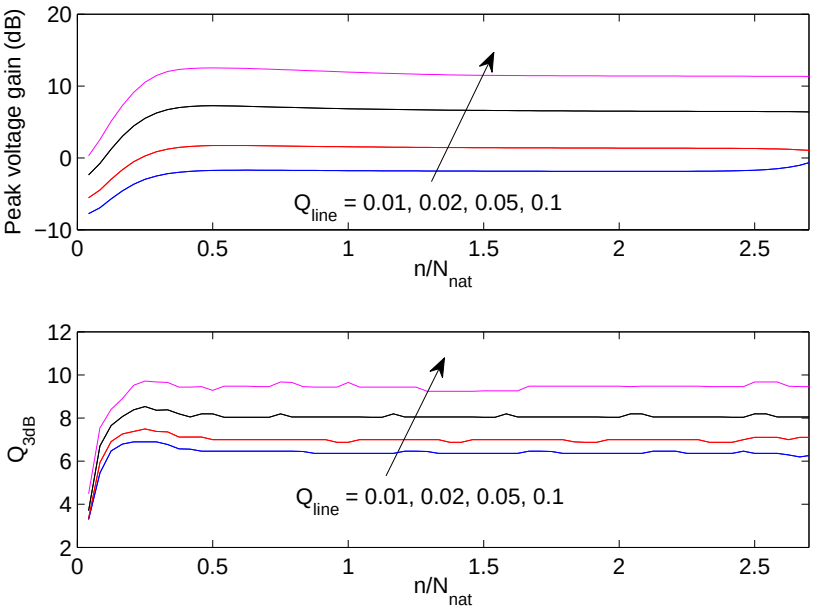


Figure 2-24: Peak voltage gain and quality factor of the bidirectional cochlea transfer functions as a function of position n along the transmission line. The plots are drawn for various values of Q_{line}. The following parameters were used to draw this plot: $d = 0.15$, $\mu = 0.76$, $Q = 3.8$ and $N_{nat} = 24$.

Figure 2-24 shows the peak voltage gain $G(n)$ and 3dB quality factor $Q_{3dB}(n)$ of the cochlear transfer functions as a function of position n along the transmission line for various values of Q_{line}. We see that both peak gain and quality factor increase with n before saturating to constant values for $n > 0.3N_{nat}$. In addition, we see that increasing Q_{line} causes both quantities to increase monotonically for all values of n. These plots were drawn by changing Q_{line} and Z_0 such that the input impedance $Z_{in} = Z_0\sqrt{Q_{line}}$ remained constant at 50Ω. The maximum allowable value of Q_{line} is set by increased inter-stage reflections (see equation (2.25)) and the eventual onset of instability.

Figure 2-25 shows the results of averaging $G(n)$ and $Q_{3dB}(n)$ along the transmission line for various values of Q_{line} and N_{nat}. The functions were averaged for values of n between $N_{nat}/2$ and $N - N_{nat}/2$ to allow the wave amplitude to build up, and also eliminate the effects of an imperfect apical termination. The figures show that average peak gain $\overline{G(n)}$ and quality factor $\overline{Q_{3dB}(n)}$ are both strongly increasing

functions of Q_{line}, and weakly increasing functions of N_{nat}. In fact, the surfaces in Figure 2-25 show that $\overline{G(n)} \propto \sqrt{Q_{line}}$. Roughly speaking, the peak voltage gain $G(n)$ is set by the ratio of the impedances seen by an input tone at its best place and at the input of the cochlea. The former is set by the series impedance Z, whose value scales as $Z_0 Q_{line}$, while the latter is equal to Z_{in}. As a result, we get

$$G(n) = \alpha \left(\frac{Z_0 Q_{line}}{Z_{in}} \right) = \frac{Z_0 Q_{line}}{Z_0 \sqrt{Q_{line}}} = \sqrt{Q_{line}} \tag{2.59}$$

where α depends on d, μ, Q and N_{nat}. Till now we have assumed values for the parameters d, μ and Q that provide a good fit to experimental data from the biological cochlea [270, 323, 328]. Roughly speaking, d controls the peak magnitude of Y_n, while μ controls the group delay. As d decreases the poles of Y_n become less damped and its peak value increases. On the other hand, as μ decreases the pair of zeros within Y_n moves further away from the poles, increasing the positive phase "bump" at $\omega_n < 1$, and hence the group delay. The parameter Q has little effect on either peak value or group delay. In practice its value can be varied over a wide range without significantly affecting performance.

We see that the values of d and μ are largely determined if we need to fit gain and group delay of the cochlear transfer functions to experimental data. However, we are not constrained in this way. Figure 2-25 shows the average peak gain $\overline{G(n)}$ of the cochlear transfer functions as a function of d and μ for a fixed value of Q. We see that the gain is maximized along a well-defined path (a straight line) in the d–μ plane. A similar plot for Q_{3dB} shows that quality factor is also maximized along the same line. This behavior persists as Q_{line} and N_{nat} are varied. We found that Q_{line} changed the slope of this high-gain line, while N_{nat} had little effect on it. Thus, the high-gain line in Figure 2-26 can be expressed by the equation $\mu + md = c$, where the slope m depends on Q_{line}, while the intercept $c \approx 1.05$ does not. Figure 2-27 shows how m varies as a function of Q_{line} for two different values of N_{nat}. We see that m decreases as Q_{line} increases, and is independent of N_{nat} to within measurement error.

The peak magnitude of Y_n, which controls the peak gain of the cochlear transfer

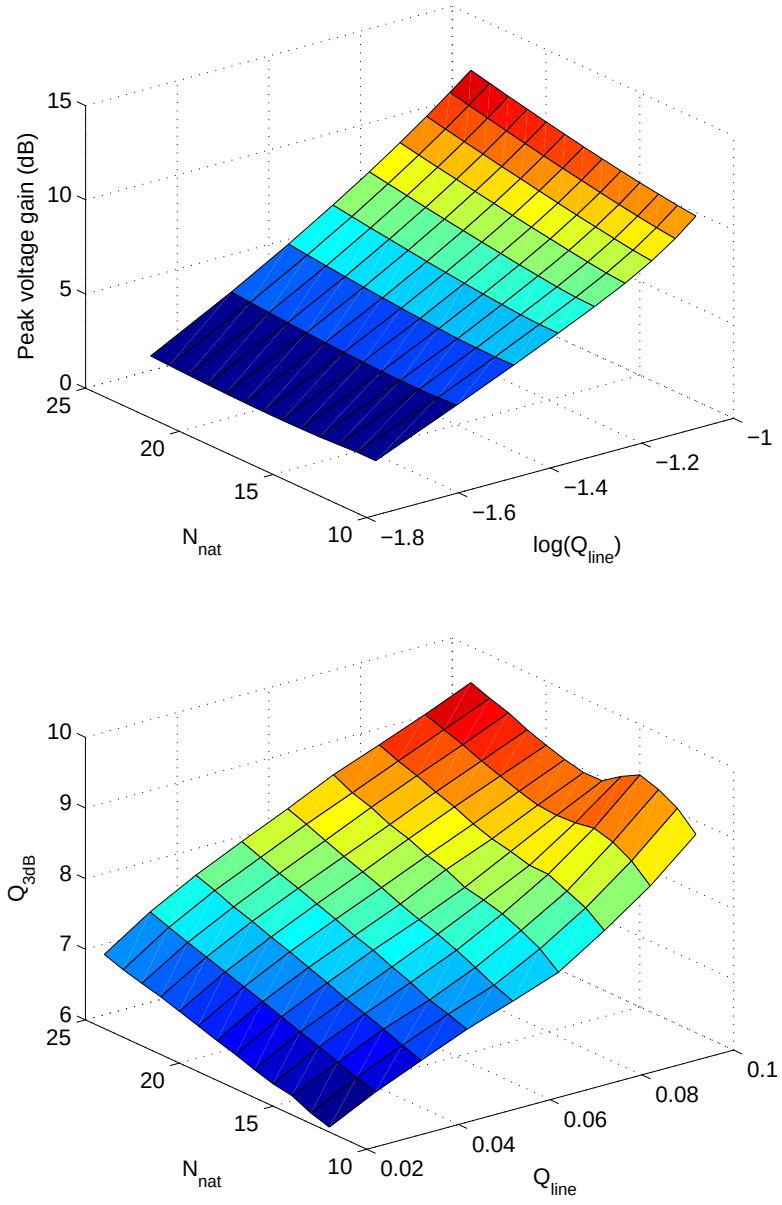


Figure 2-25: Average peak voltage gain (top) and quality factor (bottom) of the bidirectional cochlea transfer functions as a function of Q_{line} and N_{nat}. The following parameters were used to draw this plot: $d = 0.15$, $\mu = 0.76$, $Q = 3.8$ and $N_{nat} = 24$.

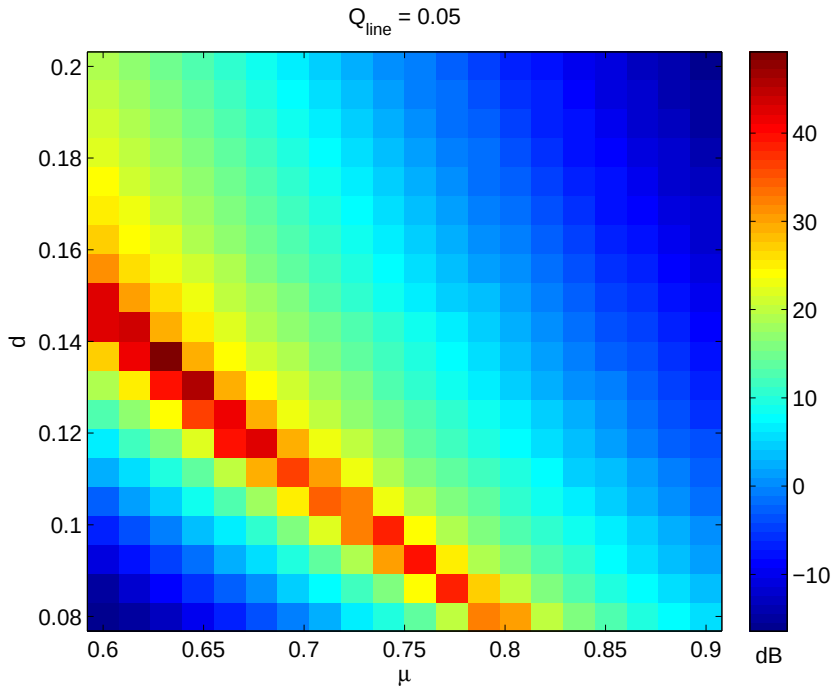


Figure 2-26: Peak voltage gain of the bidirectional cochlea transfer functions as a function of d and μ for $Q_{line} = 0.05$. The following parameters were used to draw this plot: $Q = 3.8$ and $N_{nat} = 24$.

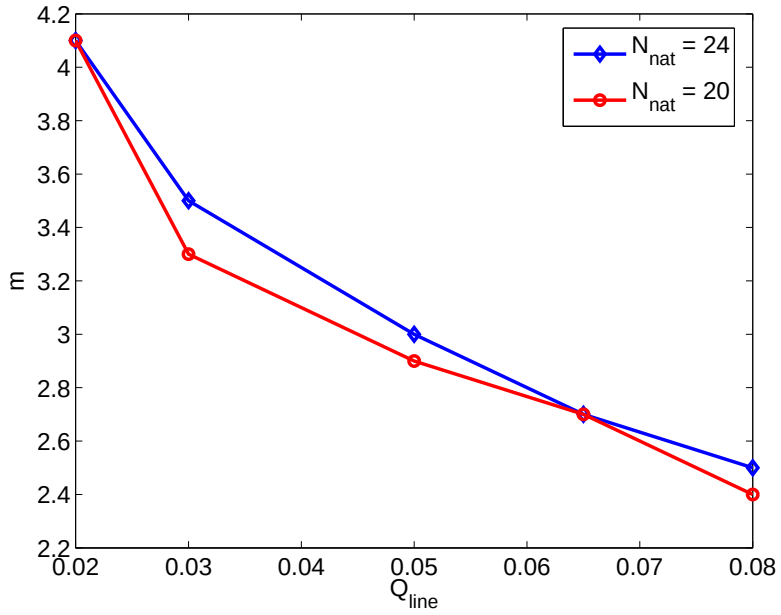


Figure 2-27: Slope of the high-gain line on the d–μ plane as a function of Q_{line} for two values of N_{nat}.

functions, decreases when d or μ increase, and vice-versa. Thus, it seems plausible that keeping a linear combination $\mu + md$ of the two parameters constant might also keep the transfer function gain constant, in general agreement with Figure 2-26. However, the behavior shown in the figure cannot be completely explained by the properties of Y_n alone, since it also depends on Q_{line}, which is purely a property of the transmission line.

An important practical issue with the bidirectional cochlea is the presence of low-frequency losses in the series impedances and shunt admittances, i.e., Z_{n1} and Y_{n1}, respectively. Such losses occur because parasitic resistances dominate the value of Z_{n1} and Y_{n1} for $s_n \ll 1$. For example, $Z_{n1}\left(s_n = 0\right) = R_{s1}$, the parasitic resistance in series with the inductor L_1 (see Figure 2-19). A non-zero value of R_{s1} is unavoidable in any real implementation of L_1. Similarly, the admittance of Y_{n1} at DC is equal to any parasitic conductance $G_{p1} = 1/R_{p1}$ in parallel with it. These parasitic components make the transmission line look like a network of resistors at low frequencies. As a result, such input frequencies are attenuated before they can reach their best place.

Such losses can be modeled by defining the effective quality factors $Q_z = Q_{line}Z_0/R_{s1}$ and $Q_y = R_{p1}/Z_0$ of Z_{n1} and Y_{n1}, respectively, and modifying Z_{n1} and Y_{n1} as follows:

$$\begin{aligned} Z_{n1}\left(s_n\right) &= Q_{line}\frac{\left(s_n + 1/Q_z\right)\left(s_n^2 + \mu s_n/Q + \mu^2\right)}{\mu^2\left(s_n^2 + 2ds_n + 1\right)^2} \\ Y_{n1}\left(s_n\right) &= \left(s_n + 1/Q_y\right) \end{aligned} \tag{2.60}$$

Figure 2-28 shows the spatial transfer function of the bidirectional cochlea to a single input frequency for various values of Q_z. Typical values of Q_z and Q_y extracted from electromagnetic simulations (and used for SPICE simulations) were 5 and 30, respectively. As a result, low-frequency losses in the series impedances, which get worse as Q_z decreases, can significantly reduce the peak gain of the cochlear transfer functions.

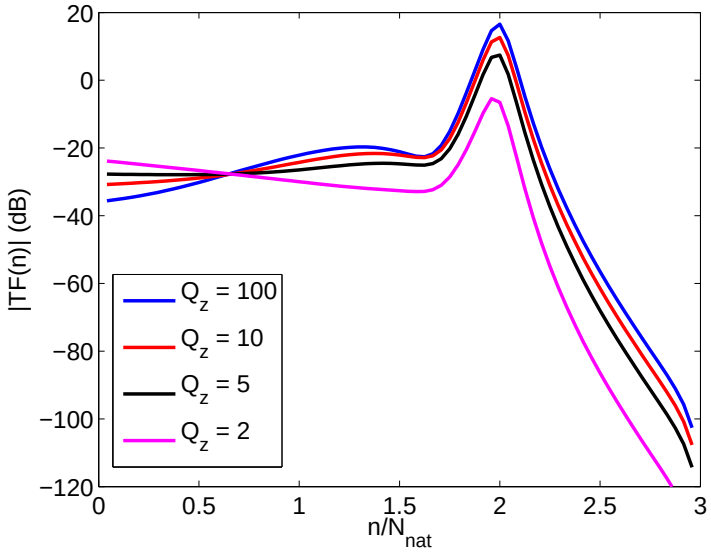


Figure 2-28: Spatial transfer function of the bidirectional cochlea to a single input frequency at various values of the low-frequency line loss parameter Q_z. Parameters used to draw this plot were $d = 0.1$, $\mu = 0.75$, $Q = 3.8$, $Q_y = 100$, $Q_{line} = 0.05$ and $N_{nat} = 24$.

2.6.2 Noise

A simplified circuit of a single cochlear stage, useful for noise calculations, is shown in Figure 2-29. Here C_3 represents the local shunt admittance $Y(n)$ (denoted by $C_1(x)$ in Figure 2-12), while the other components comprise the local series impedance $Z(n)$. The nodes on voltages n, $n-1$ and m are denoted by $v(n)$, $v(n-1)$ and $v(m)$, respectively. Nodes n and $n-1$ are shared with adjacent stages, while node m is internal to $Z(n)$. The input impedances of the sections of transmission line basal and apical to stage n are denoted by $Z_{in-}(n-1)$ and $Z_{in+}(n)$, respectively. We can find the values of $Z_{in-}(n-1)$ and $Z_{in+}(n)$ iteratively, as shown in (2.58).

Only the resistors R_1 and R_2 generate noise, as shown by the noise current sources, with PSD i_{n1} and i_{n2}, connected across them. However, R_1 is an active, negative resistor, and contains additional noise sources, such as transistors that act as current sources or sinks. The effects of such noisy devices have been lumped into the additional noise current source i_{n3}. This source also models the noise produced by any resistance in parallel with C_3. Since $Z_{in-}(n-1)$, $Z_{in+}(n)$ and the values of the various

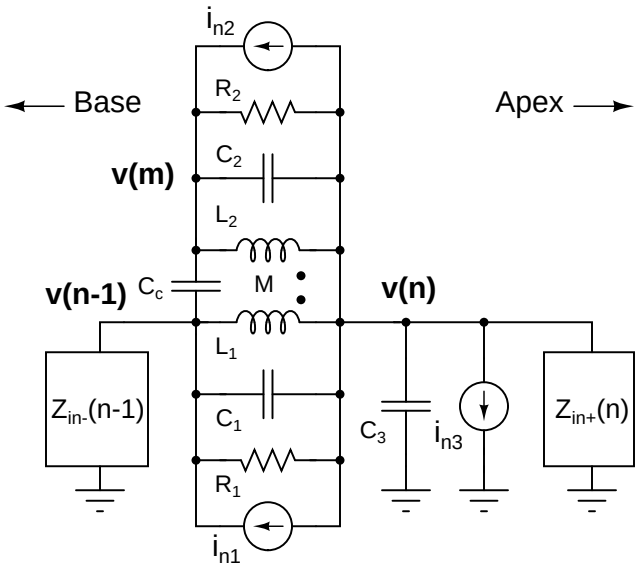


Figure 2-29: Simplified equivalent circuit of a bidirectional cochlea stage used for noise calculations.

resistors, inductors and capacitors in Figure 2-29 are known, we can analytically find *transfer impedances* from the noise sources to the nodes n and $n-1$. The transfer impedance $Z_{j,i}$ is defined the ratio of the voltage produced at node j in a network when a current is fed into node i. The two nodes may be identical, in which case the transfer impedance is known as the driving-point impedance. For example, the driving-point impedance at node n is

$$Z_{n,n} = Z_{in+}(n)||Z_{in-}(n) \tag{2.61}$$

By using superposition, we can find the PSD of the total noise voltage produced at node n. The result is

$$\overline{v_{noise,n}^2} = \overline{i_{n1}^2}\,|Z_{n,n-1} - Z_{n,n}|^2 + \overline{i_{n2}^2}\,|Z_{n,m} - Z_{n,n}|^2 + \overline{i_{n3}^2}\,|Z_{n,n}|^2 \tag{2.62}$$

We can now use the discrete transmission line equations shown in (2.57) to find transfer functions $TF_{j,n}(\omega)$. Each transfer function is defined as the voltage produced at node j in response to an input source placed at node n, divided by the voltage at node n. Normal cochlear transfer functions are given by $TF_{j,1}(\omega)$, i.e., consist of the

subset where the input source is placed at the first node ($n = 1$) of the transmission line. In the current calculation the voltage at node n is the noise voltage $v_{noise,n}$, so we have

$$\overline{v_{out,j,n}^2} = \overline{v_{noise,n}^2} \left[|TF_{j,n}(\omega)|^2 - |TF_{j-1,n}(\omega)|^2 \right] \tag{2.63}$$

where $v_{out,j,n}$ is the output noise of the j-th stage due to noise produced by the n-th stage. Here the output nose of stage j is defined as the differential noise voltage between transmission line nodes j and $j-1$. The PSD of the total noise voltage across the j-th stage is the superposition of noise from all stages, i.e., for $i = 1, 2, ..., N$, where N is the total number of stages. Thus, the PSD of the j-th output of the cochlea is given by

$$\begin{aligned} \overline{v_{out,j}^2} &= \sum_{n=1}^{N} \overline{v_{out,j,n}^2} \\ &= \sum_{n=1}^{N} \overline{v_{noise,n}^2} \left[|TF_{j,n}(\omega)|^2 - |TF_{j-1,n}(\omega)|^2 \right] \end{aligned} \tag{2.64}$$

where $v_{noise,n}$ is given by (2.62). Finally, the total noise voltage of the n-th stage is found by integrating the relevant PSD over all frequencies:

$$\overline{v_{total,n}^2} = \int_0^\infty \overline{v_{out,n}^2(f)} df \tag{2.65}$$

We see that $v_{total,n}$ can be calculated if the noise current PSDs $\overline{i_{n1}^2}$, $\overline{i_{n2}^2}$ and $\overline{i_{n3}^2}$ are known. We will ignore flicker noise, because its effects are usually negligible over our operating frequency range. Therefore the noise current PSDs are white, and each is equal to $4kT\gamma g_m$. Here g_m is the small-signal transconductance of a noisy transistor, or the conductance of a resistor. Also, γ is an excess noise factor that accounts for the presence of multiple noisy devices. Figure 2-30 shows the results of calculating $v_{total,n}$ for a maximum operating frequency of $\omega(0) = 2\pi \times 10\text{GHz}$ and various values of d. The total number of cochlear stages was $N = 3N_{nat}$.

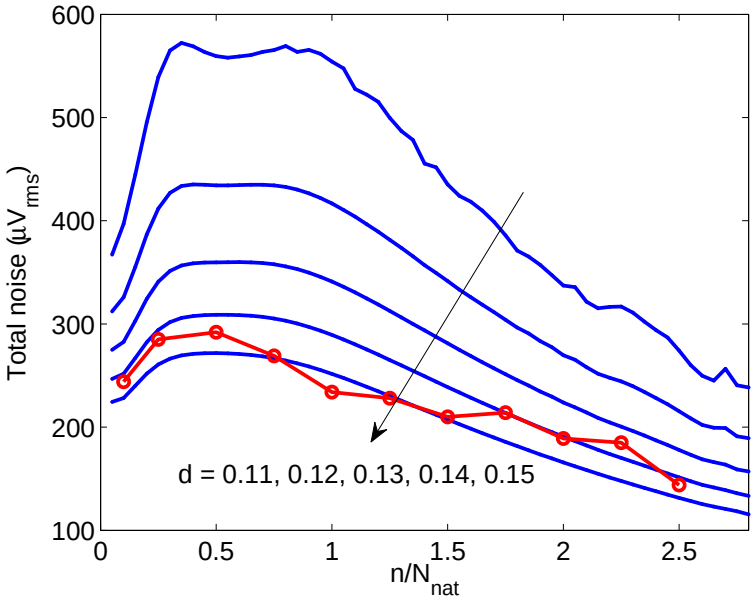


Figure 2-30: Output noise voltages of the bidirectional cochlea at various values of d. Theoretical values are shown in blue, while the result of a SPICE simulation of the integrated circuit implementation is shown in red. Parameters used to draw the theoretical curves were $\mu = 0.76$, $Q = 3.8$, $Q_y = 30$, $Q_z = 5$, $Q_{line} = 0.05$ and $N_{nat} = 20$.

Noise in the cochlea tends to increase with n because more noisy stages contribute. However, the bandwidth of the stages decreases with increasing n, which tends to reduce noise. These opposing effects, which are present in all cochlear models, compete with each other. The results are shown in Figure 2-30: The total integrated noise increases with n, reaches a peak and then decreases for all values of d. The peak gain of the cochlear transfer functions, denoted by $G(n)$, increases as d decreases and approaches the high-gain line shown in Figure 2-26. Since the signal and noise transfer functions of the cochlea are similar, the result is higher levels of output noise with decreasing values of d. Figure 2-30 also shows the results of a SPICE simulation of the bidirectional cochlea implemented on-chip, which had $d \approx 0.15$, showing that it agrees with the theoretical calculations.

Figure 2-31 shows the minimum detectable signal, denoted by v_{mds}, of the bidirectional cochlea as a function of position n for various values of d. The minimum detectable signal is defined as the smallest input amplitude that crosses the output

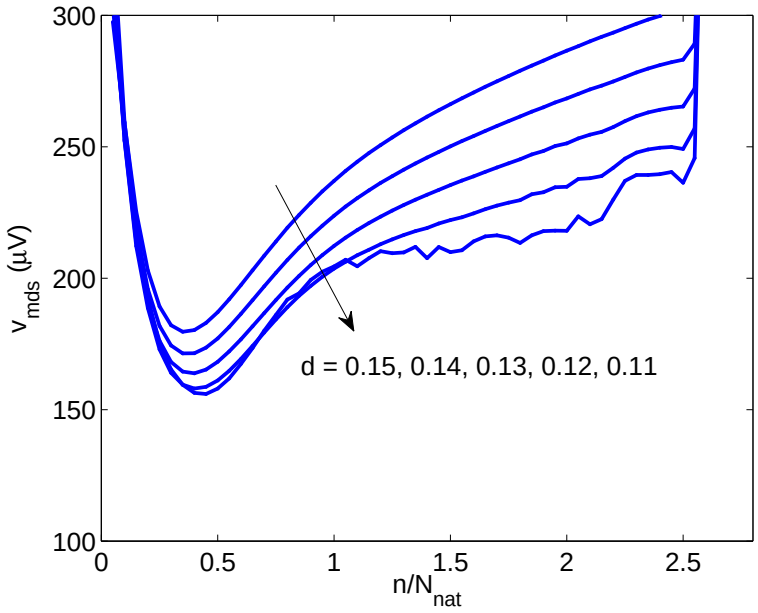


Figure 2-31: Minimum detectable signal of the bidirectional cochlea at various values of d. The same parameter values as in Figure 2-30 were used to draw this plot.

noise floor and can be detected by the cochlea. It is given by $v_{mds}(n) = v_{total,n}/G(n)$. We see that $v_{mds}(n)$ is relatively constant with n for $n > 0.5N_{nat}$ and $n < N - 0.5N_{nat}$, i.e., away from the basal and apical terminations. In addition, we see that v_{mds} is relatively constant with d, indicating that the signal and noise transfer functions roughly track each other. The average value of v_{mds} is approximately 200μV (-64dBm across a 50Ω load) with typical values of $Q_y = 30$ and $Q_z = 5$. This number improves to approximately 150μV (-66dBm) if low-frequency loss is ignored ($Q_y \gg 1$ and $Q_z \gg 1$).

2.7 Circuit Design

All circuits were designed in the 8-metal UMC 0.13μm standard CMOS process. A network that realized Z_{n1} was synthesized by a numerical optimization routine written using Mathematica. The routine, which was described earlier, produces element values, which must be converted into layout geometries for on-chip implementation. This process is known as *physical design*.

Optimized physical design of the magnetic components (inductors and transform-

ers) was important for realizing the whole system. Unlike resistors and capacitors, inductors are not standard integrated circuit components, and no models were available. Normally inductors are designed by hand. Typically, an initial geometry based on an analytical formula is iteratively refined by using electromagnetic simulations. While this approach is sufficient for typical RF designs that use a small number of transformers, it rapidly becomes tedious when this number increases. In addition, the mapping from transformer values to geometry is one-to-many, so the final geometry may not be optimal. We decided to automate this process as much as possible.

We used an analytical formula that predicts the inductance value, L, as a function of geometry [204] and derived the folllowing formula for calculating the DC series resistance $R_{s,DC}$ of a N-turn spiral:

$$R_{s,DC} = \left(\frac{\rho}{wt}\right) NM \tan\left(\pi/M\right) d_{avg} \tag{2.66}$$

where ρ is the resistivity of the metal layer, t its thickness, and w the width of each turn. In addition, each turn is assumed to consist of a regular M-sided polygon, d_{out} and d_{in} are the diameters of the circles that inscribe the outer and inner edges of the spiral, respectively, and $d_{avg} = \left(d_{out} + d_{in}\right)/2$. It is worth noting that, as one might expect, $M \tan\left(\pi/M\right) \to \pi$ as $M \to \infty$ and the polygonal coil becomes circular. The AC resistance can be found by taking the skin effect into account [158]:

$$R_s(\omega) \approx R_{s,DC} \left(\frac{\eta}{1 - e^{-\eta}}\right) \tag{2.67}$$

where $\eta \equiv t/\delta$, and δ is the skin depth. We assumed that ω_c, the center frequency of the stage, was much lower than the self-resonant frequency of the coil. Thus we were able to analytically find the quality factor $Q_c = \omega_c L/R_s$ of the inductor at ω_c.

We then wrote a numerical optimization routine using Mathematica to find the optimal coil geometry. The routine finds the geometry that produces the required value of L while minimizing layout area and also ensuring that Q_c is higher than Q_{min}, a constant. Square coils ($M = 4$) were used because they have the largest inductance for a given layout area.

Planar integrated transformers can be wound in two main ways [170]. In Frlan windings, the two coils are interleaved, i.e., wound within each other on the same metal layer. In Finlay windings, the two coils are wound on different metal layers and stacked vertically. We used the Finlay winding technique because of two advantages. Firstly, it results in higher coupling factors. Secondly, because each coil can be more tightly wound than with Frlan windings, we get greater inductance for the same layout area. Finlay windings do result in higher capacitance between the coils, which lowers the self-resonance frequency, but since this capacitance can be absorbed into our synthesized networks it is not a real problem. The two coils were therefore laid out on different metal layers. In certain cases several metal layers electrically shorted to each other via numerous vias were used to realize each coil. By reducing the series resistance this technique increases the quality factor of the coil. The centers of the two coils were offset from each other, and the amount of offset was varied to control the value of the coupling factor k. This process was repeated for every stage.

A planar electromagnetic simulator (ASITIC, written by Ali Niknejad[15]) was then used to create broadband frequency-domain (two-port S-parameter) models for each transformer. A model-order reduction routine available in Cadence was used to create lumped equivalent circuit models, suitable for time-domain simulations, from the frequency-domain models. Once the simulated performance was judged satisfactory a program written in Matlab was used to automatically generate on-chip layouts (in CIF format) for each transformer.

Capacitors were either of the vertical-field, parallel-plate type or the interleaved horizontal/fringing-field type. In this process the latter has higher capacitance density ($2\text{fF}/\mu\text{m}^2$ versus $1\text{fF}/\mu\text{m}^2$), which is desirable for minimizing chip area, but also somewhat higher parasitic capacitances to the substrate (approximately 2% versus 1%). When linearity was not critical and one terminal was at AC ground capacitors were implemented using MOSFET gate oxide because of the high capacitance density ($12\text{fF}/\mu\text{m}^2$). Bypass capacitors provide one example.

[15]ASITIC may be downloaded for free from http://rfic.eecs.berkeley.edu/ niknejad/asitic.html. However, active program development appears to have stopped.

For the realistic parameter values mentioned in the previous section ($d = 0.1$, $\mu = 0.76$ and $Q = 3.8$), the series impedance Z_{n1} is not physically realizable using only passive elements. In fact, at least two resistances are required to synthesize Z_{n1}, say R_1, must be negative to pump energy into the traveling wave in regions basal of the peak and increase gain. The second resistance R_2 must be positive for overall stability. Our basic stage design is shown in Figure 2-32. Each Z_{n1} consists of two resonators that are coupled both inductively and capacitively, and R_1 is an active negative resistor. Cross-coupled NMOS transistors connected between adjacent nodes were used to the generate R_1.

At positions far before the peak the negative resistor R_1 cannot pump energy into the traveling wave, since Z_{n1} is dominated by the inductor L_2. Any parasitic series resistance of L_2 now absorbs energy from the wave, causing it to attenuate. An additional active negative resistance, R_{LL} in parallel with the shunt admittance C_3, is used to cancel such attenuation. The line was terminated at the low-frequency end by a parallel RC circuit. The element values $R_t = 40\Omega$ and $C_t = 1.8$pF were designed as described in the previous section to provide a impedance-matched, dissipative load, thereby minimizing reflections. The input impedance Z_{in} of the cochlea (approximately real within the operating frequency range) was designed to be 50Ω for interfacing with test equipment.

A more detailed circuit diagram of a single bidirectional cochlea stage is shown in Figure 2-34. A cross-coupled pair of NMOS transistors, M_1 and M_2, creates the negative resistance R_1. The bias current I_B through the pair is set by the control voltage V_B. The value of the negative resistance is controlled by varying V_B. The DC voltage on the line (i.e., the DC value of V_1 and V_2) is set by V_P, which in turn is set by a low-frequency negative feedback loop, shown in Figure 2-33, that senses the line voltage and compares it with a reference voltage V_{REF} that is normally set around $V_{dd}/2$.

The impedance produced by the cross-coupled pair between V_1 and V_2 consists of a resistance $R_1 = -2/g_m$ in parallel with a capacitance $C_{par} = C_{gs}/2 + 2C_{gd}$, where g_m, the small signal transconductance, is an increasing function of I_B, and C_{gs}

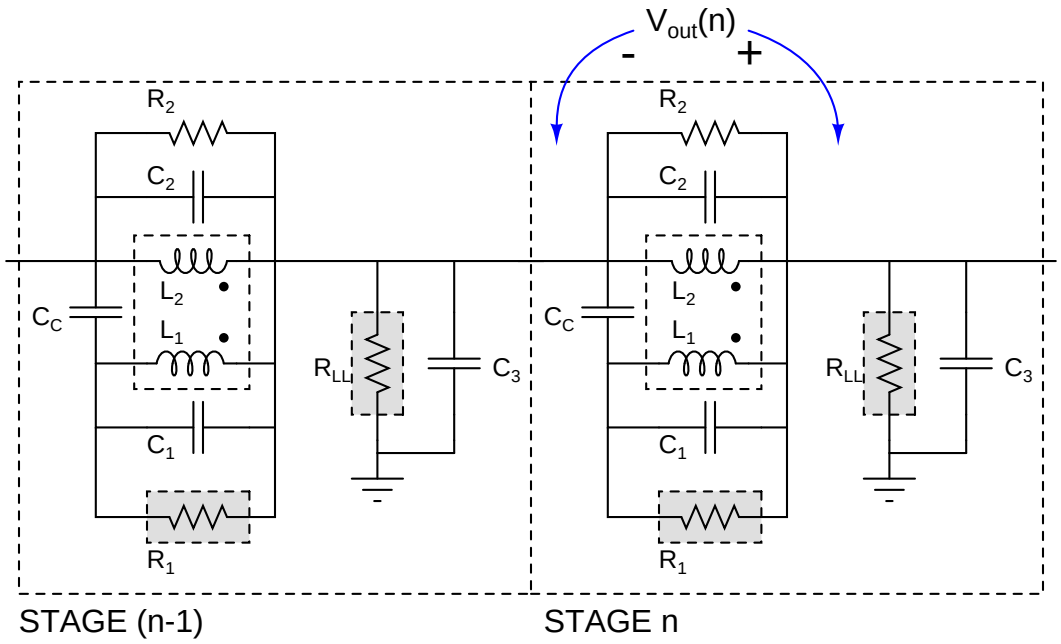


Figure 2-32: The basic bidirectional cochlea cochlea stage.

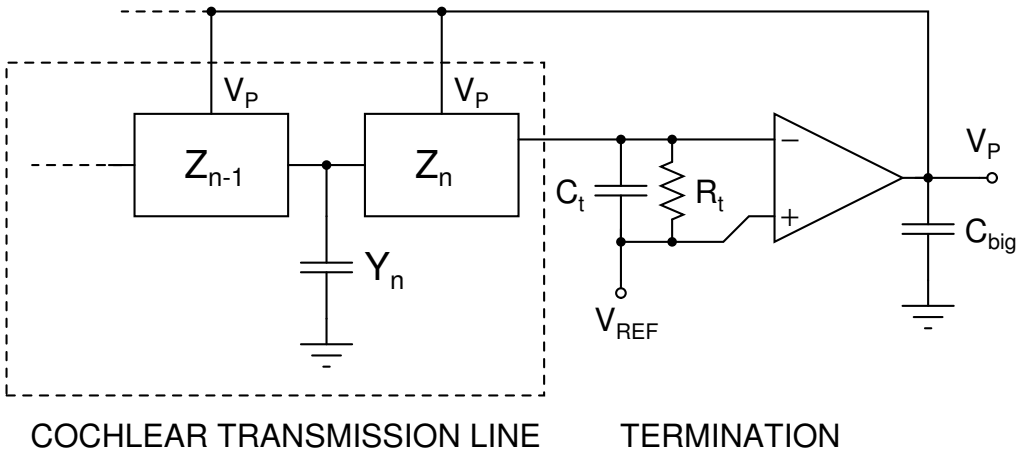


Figure 2-33: The low-frequency feedback loop that sets the DC line voltage.

and C_{gd} are the gate-source and gate-drain capacitances, respectively. An important advantage of this topology is that C_{par} can be absorbed into C_1.

The line loss cancelation network is a single-ended negative resistor R_{LL} that is created in two stages. The voltage at V_2 is first amplified by a common-source amplifier. The output of this amplifier controls a current source, M_4, that can sink or source current from V_2. Because of the sign inversion produced by the amplifier, M_4 pushes current into the node when the voltage on it rises (and vice-versa) thereby creating a negative resistance of value $-1/\left(g_{m3}^2 R_B\right)$, where g_{m3} is the transconductance of M_3 and M_4 (assumed equal). The value of g_{m3}, and thus R_{LL}, is set by the bias voltage V_{B2}. The value of C_B is made large enough for the pole frequency g_{m3}/C_B to be much smaller than the center frequency at the location of interest, allowing the amplifier to reject the DC value of V_1 and only respond to RF (i.e., have a highpass characteristic). Without this loss-cancelation network, low frequencies would be attenuated by a factor $\alpha = 1 - R_{s2}/\left(R_{s2} + Z_{in}\right)$ at every stage, where R_{s1} is the parasitic series resistance of L_2 (not drawn). With $Z_{in} = 50\Omega$ by design and a typical $R_{s2} = 5\Omega$, signals that peak at the end of the cochlea (after 50 stages) would be attenuated by a factor of approximately $\alpha^{50} = 10^{-2}$ (-40dB) before reaching the apex.

A complete cochlear stage, including readout circuits, is shown in Figure 2-35. The output voltages $V_{out}(n)$ are pre-amplified before their envelopes are detected and read out. Each pre-amplifier is a two stage resistively-loaded, common source, differential amplifier, with inductive shunt-peaking in the early stages to increase the bandwidth and a voltage gain of approximately $A_{preamp} = 6.3$(16dB). Their main purpose was to reduce the input-referred dead-zone of the envelope detector (ED), which is shown in Figure 2-36.

The ED can be modelled as a full-wave rectifier followed by a low-pass filter. It rectifies the differential RF voltage $v_{RF} = (RF+) - (RF-)$ and filters the result with a bandwidth BW_{ED}. The resistor R_1 and capacitor C_1 act as a high-pass filter for the RF input. The values of R_1 and C_1 are chosen such that $1/\left(2\pi\tau_1\right) \ll f_{RF}$, where $\tau_1 = R_1 C_1$ and f_{RF} is the RF frequency. In this case the entire differential RF voltage appears between the gate and source of each PMOS transistor, which act as diodes.

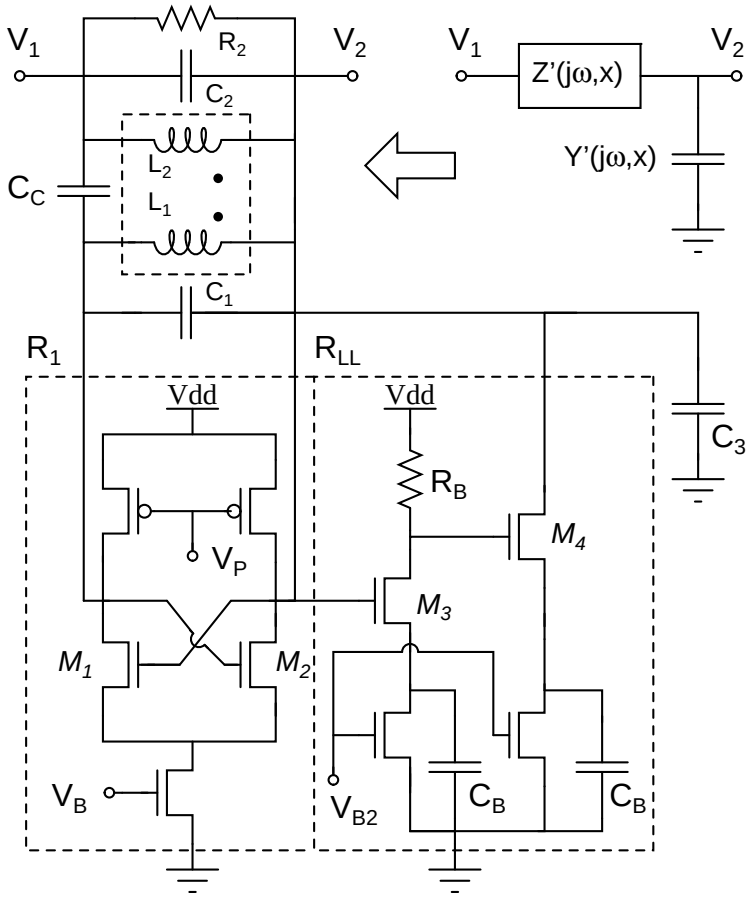


Figure 2-34: A more detailed circuit diagram of a single bidirectional cochlea stage.

We define $g_m = I_L/\left(\phi_T/\kappa\right)$ as the transconductance of these transistors, where ϕ_T is the thermal voltage and κ is the subthreshold constant. The impedance at the output node is given by

$$\begin{aligned} Z_{OUT} &= \frac{1}{g_m}\left[\frac{s\tau_1+1}{\tau_1\tau_2 s^2+\left(C_1/C_L+1\right)\tau_2 s+1}\right] \\ &\approx \frac{1}{g_m}\left[\frac{1}{\tau_{ED}s+1}\right] \end{aligned} \tag{2.68}$$

where $\tau_2 = C_L/g_m$, $\tau_{ED} = \left(C_1/C_L+1\right)\tau_2$, and the approximation is valid when $\tau_1 \ll \tau_{ED}$. Thus the output of the ED behaves as a first-order low-pass filter with a bandwidth that is given by

$$BW_{ED} = \frac{1}{2\pi\tau_{ED}} = \frac{g_m}{2\pi\left(C_1+C_L\right)} = \frac{I_L}{2\pi\left(\phi_T/\kappa\right)\left(C_1+C_L\right)} \tag{2.69}$$

We used the following values in our design: $R_1 = 10\text{k}\Omega$, $C_1 = 250\text{fF}$, resulting in $1/\left(2\pi\tau_1\right) = 64\text{MHz}$, and $C_L = 1.2\text{pF}$. The value of I_L can be set by the user. For a typical value of $I_L = 2\mu\text{A}$, we get $BW_{ED} = 6.3\text{MHz}$.

The output voltage is proportional to V_{RF}, the amplitude of the RF input voltage. To be precise, we get $v_{OUT} \approx \left(V_{DC}+V_{RF}-V_D\right)$, where V_{DC} is the value of v_{OUT} for $V_{RF} < V_D$, and V_D is the dead-zone of the rectifier[16]. Because the rectifier is a passive MOS diode, $V_D \approx \phi_t/\kappa$. Pre-amplification reduces the input-referred dead-zone by a factor of A_{preamp}. Thus, the amplitude of the smallest RF voltage that can be detected is approximately $\left(\phi_T/\kappa\right)/A_{preamp} \approx 5.5\text{mV}$.

Finally, because of a shortage of pins, output voltages from the ED present inside every stage were buffered using source-followers and time-multiplexed onto a single output bus using a token-passing circuit, or scanner. This circuitry is represented by the analog multiplexer block shown in Figure 2-35.

[16]This analysis assumes a "hard" dead-zone, i.e., that $v_{OUT} - V_{DC} = 0$ for $V_{RF} < V_D$. In reality $v_{OUT} - V_{DC}$ approaches zero quadratically as V_{RF} decreases below V_D, allowing signals somewhat smaller than V_D to be detected. See Appendix A for a more detailed analysis.

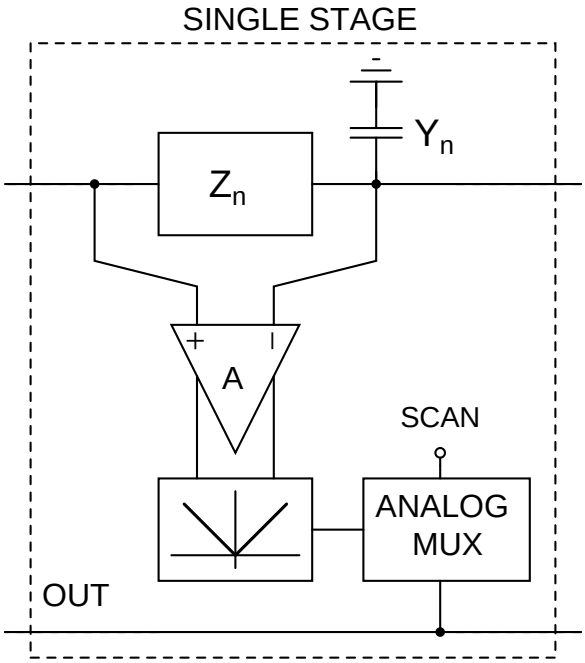


Figure 2-35: A complete bidirectional cochlea stage.

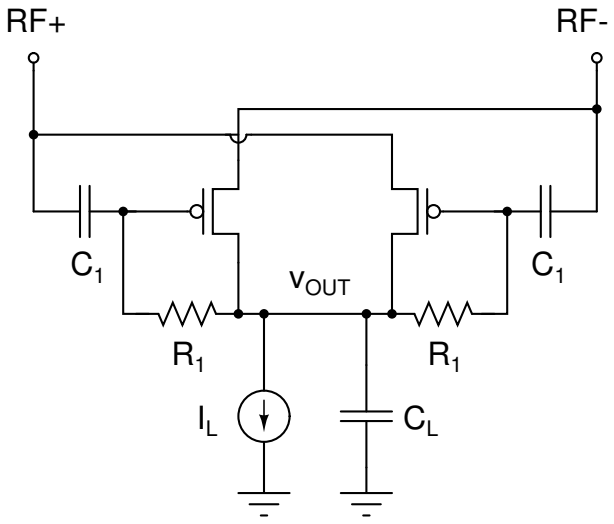


Figure 2-36: The differential envelope detector present within each cochlea stage.

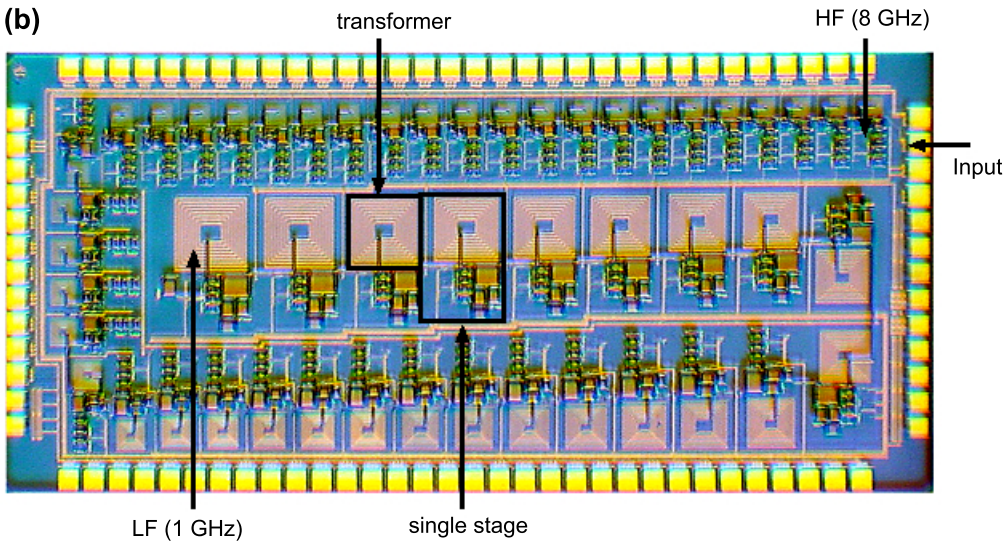


Figure 2-37: Die photograph of the bidirectional RF cochlea chip.

2.8 Measurements

A die photograph of the bidirectional RF cochlea chip is shown in Figure 2-37. The transmission line was arranged such that it spiralled inward from the input terminal to save space, in a manner reminiscent of the biological cochlea. The chip was wirebonded to a printed circuit board for testing. Output voltages were digitized and captured using a digital oscilloscope and custom software written in Labview, a commercially available software package from National Instruments. Further post-processing was performed using Matlab. The measurement floor after processing is limited by quantization noise from the oscilloscope. At a scan rate of 10kHz, we estimate this floor to be approximately 35μV (rms), setting a displayed average noise envelope of 100μV (-80dBV). This value is significantly lower than the measured noise level, which is set by output noise from the circuit.

Fig. 2-38 shows the measured input reflection coefficient, $|S_{11}|$, of the bidirectional cochlea chip at various input power levels. The matching bandwidth, defined as the frequency range over which $|S_{11}| < -8$dB, was DC to 7.2GHz. Matching at high frequencies was limited by chip packaging. Packages attenuate high frequency signals because of bond-wire inductances and bond-pad capacitances, which together form low-pass filters.

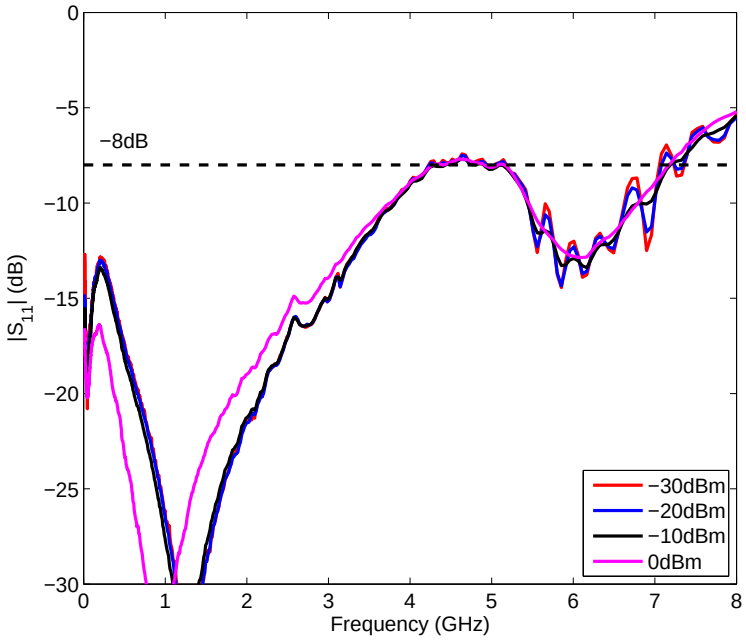


Figure 2-38: Measured input reflection coefficients of the bidirectional cochlea chip at various input power levels.

Figure 2-39 shows measured spatial responses of the bidirectional cochlea to logarithmically spaced input frequencies varying between 1GHz and 8GHz, with the input power level fixed at -10dBm. Figure 2-40 shows how the spatial response at various frequencies changes if the value of the negative resistance, R_1 is changed by varying the bias voltage V_B. As V_B is increased the peak gain increases. For very high values of V_B instabilities develop at several locations and propagate in both directions along the line. The oscillation frequencies were measured using an external spectrum analyzer at the input terminal for various values of V_B. This data is shown in Figure 2-41. Here the color bar encodes the power level of each output frequency relative to the spectrum analyzer's measurement floor, which was approximately -70dBm.

The heights of the spatial response peaks can be increased by decreasing the value of R_1. We can do this by increasing I_B, the bias current through the transistors that create R_1. For very high bias currents multiple instabilities develop, resulting in oscillations that travel in both directions along the line and can be detected at the input terminal. Figure 2-42 is a histogram that shows that the ratio of adjacent

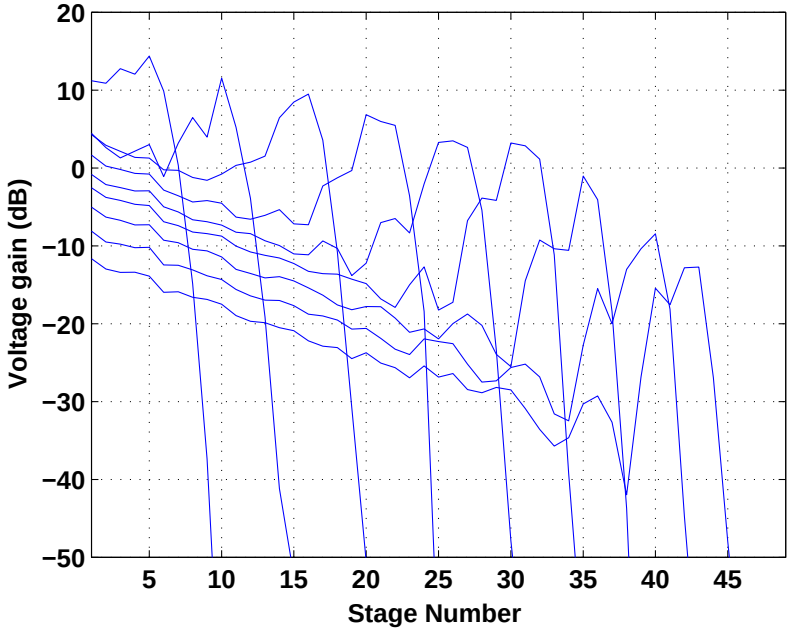


Figure 2-39: Spatial responses of the bidirectional RF cochlea to logarithmically spaced input frequencies varying between 1GHz and 8GHz. The input power level was fixed at -10dBm.

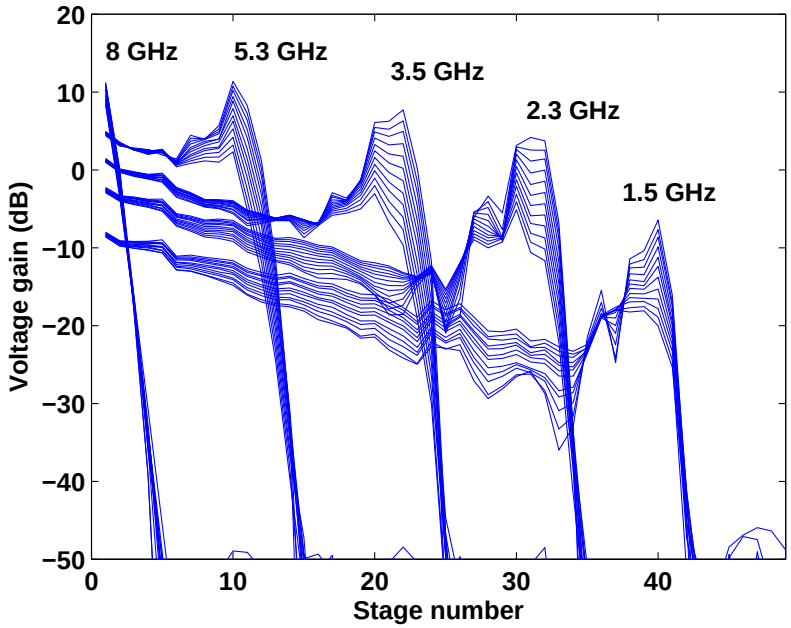


Figure 2-40: Spatial responses of the bidirectional RF cochlea at various frequencies obtained while varying the value of the active element within each stage. The bias voltage V_B that sets this negative resistance R_1 was increased from 0.56V to 0.67V in 10mV steps.

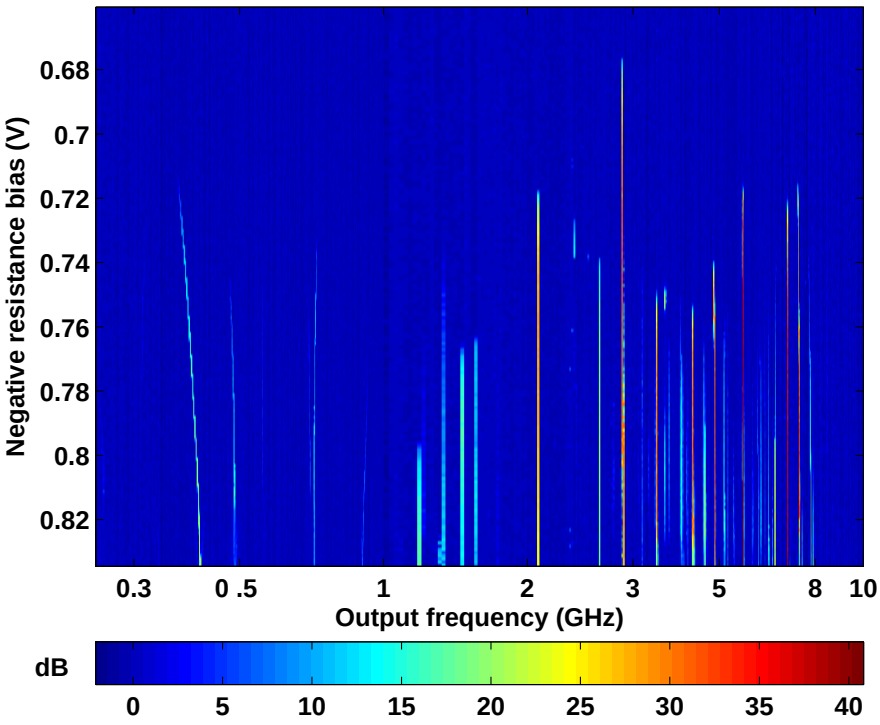


Figure 2-41: Output spectrum of the bidirectional RF cochlea as a function of V_B, the bias voltage that controls the value of the negative resistance R_1 within each stage.

output frequencies f_{i+1} and f_i is constrained to be $\exp(n/N_{nat})$, where n is an integer ($n = 1$ and 2 are shown). Only peaks present in Figure 2-41 at a level 20dB or more above the measurement floor of the spectrum analyzer were used to draw this figure. The simplest explanation for this behavior is that individual stages act as oscillators with oscillation frequencies close to the local resonant frequency $\omega_c(n)$, which scales as $\exp(-n/N_{nat})$. However, the output frequencies vary between 400MHz and 8GHz, larger than the available range of $\omega_c(n)$. In addition, if peaks lower than 20dB are included, several additional frequencies are revealed that do not scale as $\exp(-n/N_{nat})$, as shown in Figure 2-43, where the threshold has been relaxed to 10dB. We therefore suggest an additional mechanism to explain these observations: standing-wave modes that develop in the "cavity" with partially-reflecting walls that is formed between a spatial location n and any impedance discontinuity at the input terminal. This mechanism is similar to a laser, with the transmission line acting as the active gain medium, and may also cause spontaneous otoacoustic emissions in the biological cochlea [271]. The displayed average noise level from the spectrum analyzer itself was only ±1dB (peak-to-peak), so even at the lower threshold setting the frequencies being detected are being generated by the cochlea and are not the result of instrument noise.

When operating as an "RF laser" the cochlea can be periodically turned on and off to produce trains of narrow pulses (modulated wave-packets). These pulses can be used as signal sources in ultra-wideband or impulse radio applications. In addition, the instantaneous frequency within each pulse can be varied with time to produce chirp waveforms that are useful for radar.

Figure 2-44 shows spatial responses of the bidirectional cochlea at -10dB input level for various values of the line loss cancelation resistance R_{LL}. The value of R_{LL} can be varied by changing the bias voltage V_{B2}. We see that a significant amount of line loss can be canceled by increasing V_{B2}, which decreases R_{LL}. However inter-stage reflections also increase, limiting the amount of cancelation that can be applied.

Figure 2-45 shows how the peak gain of the bidirectional cochlea responses decreases with increasing input amplitude. These compression curves were taken by

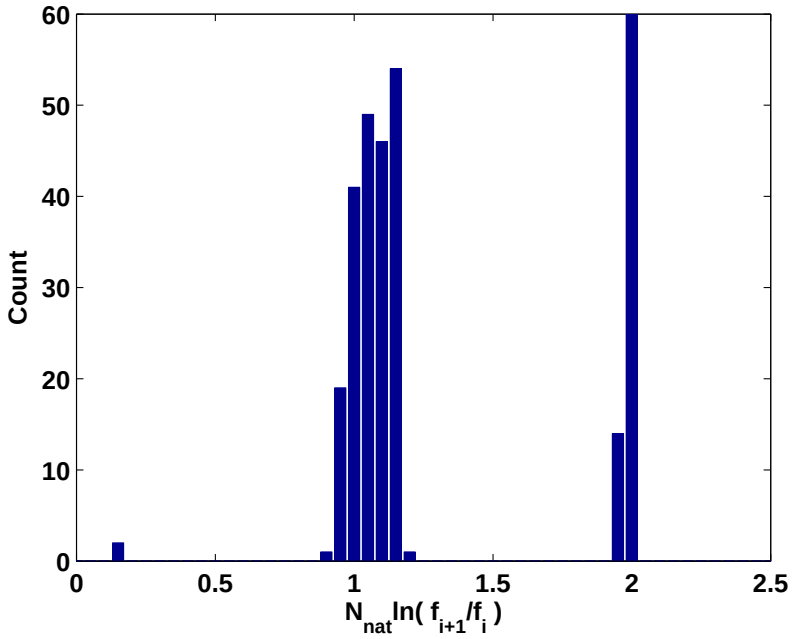


Figure 2-42: Ratio of output frequencies produced by the bidirectional RF cochlea (20dB threshold).

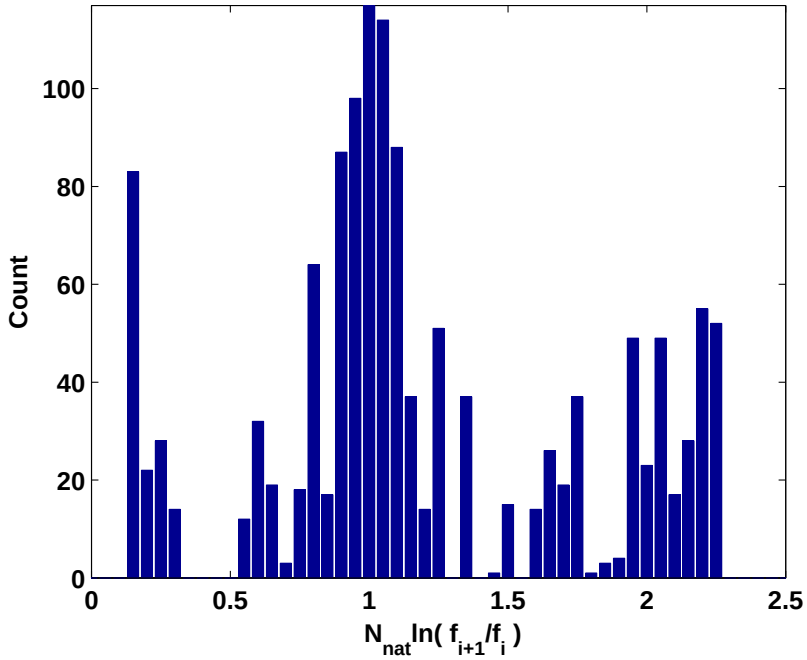


Figure 2-43: Ratio of output frequencies produced by the bidirectional RF cochlea (10dB threshold).

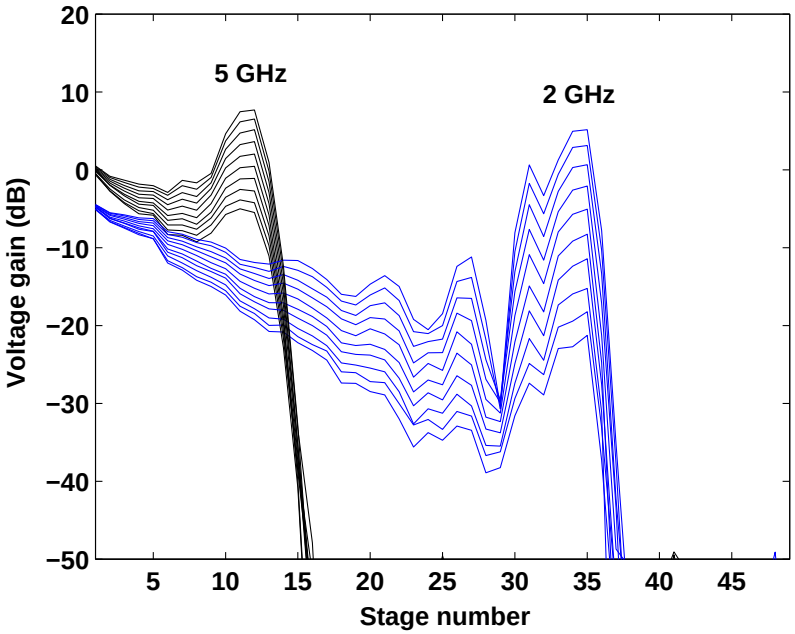


Figure 2-44: Spatial responses of the bidirectional RF cochlea at various frequencies obtained while varying the value of the series loss cancelation within each stage. The bias voltage V_{B2} that sets this negative resistance R_{LL} was increased from 0.40V to 0.58V in 20mV steps.

observing the response at a fixed location, the best position for $f_{max} = 5.3$GHz, to various input frequencies, including f_{max}. We see that the response at f_{max}, being larger, compresses for smaller input power levels than at other frequencies. This behavior is qualitatively similar to that observed in the biological cochlea.

Figure 2-46 shows a two-tone response: here two input frequencies were simultaneously fed into the bidirectional cochlea. One tone was held fixed at 2.4GHz, while the other was swept logarithmically with time. The cochlear outputs were monitored as a function of time and plotted in the figure. As expected, the spatial response due to the second tone moves linearly with time, while that due to the first remains fixed. Both tones had equal input amplitudes (-10dBm). Improvements in frequency resolution, especially in the presence of noise, can be obtained by using the phase information, such as temporal correlations between stages, present within cochlear transfer functions [321].

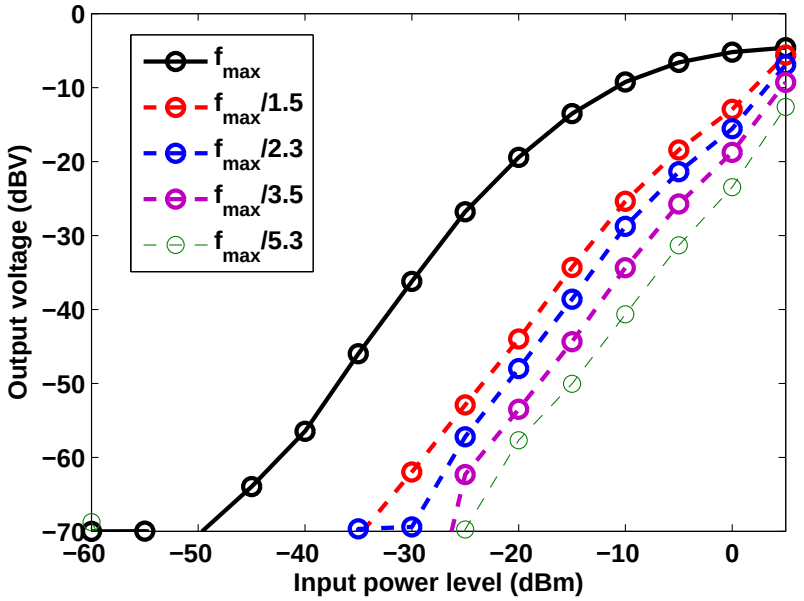


Figure 2-45: Measured compression curves of the bidirectional RF cochlea. The spatial location was fixed at the point where maximum response was obtained for $f_{max} = 5.3$GHz, and the response to frequencies below f_{max} was measured at several power levels.

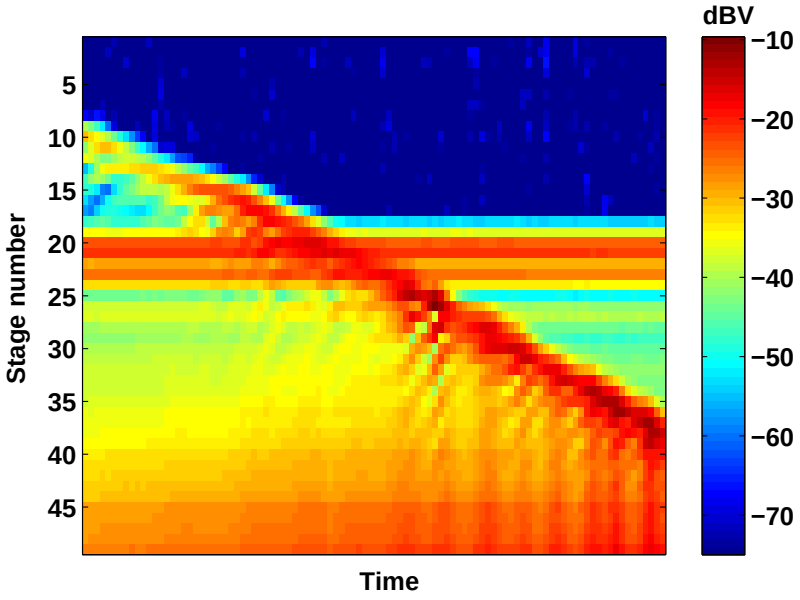


Figure 2-46: Measured response of the bidirectional RF cochlea to two simultaneously applied input frequencies. One input was held fixed at 2.4GHz while the other was increased logarithmically from 1GHz to 8GHz (left to right in the figure). The power level of both inputs was held fixed at -10dBm.

2.9 Appendix A: Impedance Functions

In this section, we discuss the properties of impedance functions. There are several reasons. Firstly, we know that any driving point immittance (impedance or admittance) that is *positive real* can be synthesized using only passive, lumped elements [292]. We therefore want to know whether $Y_n(s_n)$ and $Z_{n1}(s_n)$, which model the effects of the organ of Corti, are positive real. If this were true, the whole cochlea could be implemented passively. A purely passive system would be highly linear and have low noise (i.e., large dynamic range). We therefore discuss the conditions that a driving point immittance function must satisfy in order to be realizable using only passive elements.

Theorem 1 *A real, rational function is the driving point immittance of a linear, passive, lumped, reciprocal and time-invariant network, if and only if it is positive real [62].*

Theorem 1 is a necessary and sufficient condition for the *physical realizability* of an immittance function using a finite number of lumped passive elements is that it be rational and positive real. All positive real functions are not physically realizable. They also have to be rational, i.e. consist of the ratio of two polynomials. Given a rational driving point immittance function $F(s) = N(s)/D(s)$, it is positive real if and only if the following conditions are satisfied:

1. All coefficients of $F(s)$ must be real. This means that all zeros and poles (the roots of the numerator and denominator polynomials $N(s)$ and $D(s)$, respectively) are either real or occur in complex conjugate pairs.

2. (a) As $s \to \infty$, $F(s)$ must approach one of the three following forms: Ks, K or K/s, with $K > 0$. This means that the orders of $N(s)$ and $D(s)$ can differ by at most 1.

   (b) $F(s)$ cannot have any poles in the open right half s-plane (RHP), i.e., the immittance function must be analytic in the open RHP. In addition, if

$F(s)$ is positive real, $1/F(s)$ must also be positive real. Since the zeros of $F(s)$ are the poles of $1/F(s)$, it follows that $F(s)$ cannot have any poles *or* zeros in the open RHP. Thus $F(s)$ must be *minimum phase*.

(c) If $F(s)$ has any poles on the imaginary axis, they have to be simple and have real and positive residues.

3. The real part of the frequency response of $F(s)$ must always be positive, i.e., $\Re\left[F\left(\jmath\omega\right)\right] \geq 0$, $\forall\omega \in \{-\infty, \infty\}$.

Among the other useful properties of positive real functions are the following:

1. The sum of two positive real functions is also positive real.

2. If $F(s)$ is positive real, it must be real if s is real.

3. If $F_1(s)$ and $F_2(s)$ are positive real, so is $F_1\left[F_2(s)\right]$.

4. If $F(s)$ is positive real, so are $1/F(s)$ and $F(1/s)$.

Unfortunately, the cochlear admittance function $Y_n\left(s_n\right)$ and its various obvious transformations, such as $1/Y_n\left(s_n\right)$ and $Y_n\left(1/s_n\right)$, are not positive real. Thus they cannot be realized as driving point immittances of purely passive networks. Hence active circuit elements are needed to implement the cochlea. Specifically, $Y_n\left(\jmath\omega_n\right)$, $1/Y_n\left(\jmath\omega_n\right)$ and $Y_n\left(1/\jmath\omega_n\right)$ have negative real parts for some values of ω_n. Thus, they fail the third test of positive reality (from the list enumerated above). For example, $\Re\left[1/Y_n\left(\jmath\omega_n\right)\right] < 0$ for $\omega_n < 1$ and > 0 for $\omega_n > 1$. Physically, this means that energy is being pumped into the traveling wave by the transmission line when $\omega_n < 1$, causing the wave amplitude to build up. On the other hand, the line takes energy out of the wave when $\omega_n > 1$, causing the wave amplitude to decay.

2.10 Appendix B: Gyrator Noise Analysis

Gyrators are circuits that invert impedances. An example of a completely passive (albeit narrow-band) gyrator is a quarter-wavelength transmission line. If the load

at one end of the line is Z_L and its characteristic impedance is Z_0, the impedance seen looking in from the other end is given by $Z_{in} = Z_0^2/Z_{in}$. Now consider the generic active inductor shown in Figure 2-47. This type of circuit is also a gyrator, since the impedance Z_{in} seen looking into v_{in} is an inverted version of that present at the intermediate node v_1. In this case, the capacitance C is inverted to form an inductance at the input terminal. In the figure g_{m1} and g_{m2} represent small-signal transconductances (voltage-controlled current sources) and r_{o1} and r_{o2} are their output impedances. In the ideal case, r_{o1} and r_{o2} are infinitely large and we get

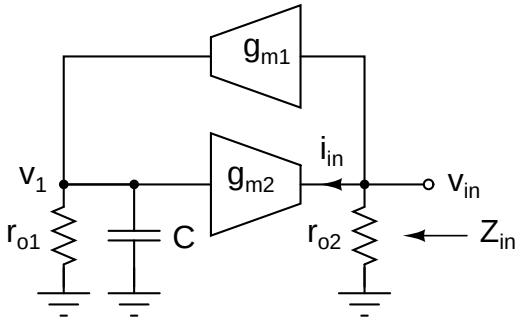


Figure 2-47: A simple small-signal model of an active inductor created with a gyrator.

$$Z_{in} = \frac{v_{in}}{i_{in}} = \frac{sC}{g_{m1}g_{m2}} \equiv sL_{eff} \tag{2.70}$$

Here $L_{eff} = C/(g_{m1}g_{m2})$ is the effective inductance seen at the input terminal. Note that the ideal gyrator (unlike the passive transmission line) gyrates over all frequencies. The synthesized inductor is also ideal, i.e., has an infinite quality factor. When r_{o1} and r_{o2} are finite the situation is more complicated. It is easy to show that the input impedance is given by

$$Z_{in} = \left(\frac{sC}{g_{m1}g_{m2}} + \frac{1}{r_{o1}g_{m1}g_{m2}}\right) \parallel r_{o2} \tag{2.71}$$

For simplicity, consider the case when r_{o2} is so large that it can be ignored. Several techniques can be used to increase the output impedance of g_{m2} so that this condition is met. In this case Z_{in} has the form of an inductance $L_{eff} = C/(g_{m1}g_{m2})$ in series with a resistance $R_{eff} = 1/(r_{o1}g_{m1}g_{m2})$. The quality factor of Z_{in} is given by

$$Q = \frac{sL_{eff}}{R_{eff}} = sCr_{o2} \tag{2.72}$$

Now consider adding noise currents at the output terminals of both transconductors, as shown in Figure 2-48. The noise current power spectral density (PSD, in A^2/Hz) due to each transconductor is given by

$$\overline{i_{nj}^2} = 4kTN_j g_{mj} \tag{2.73}$$

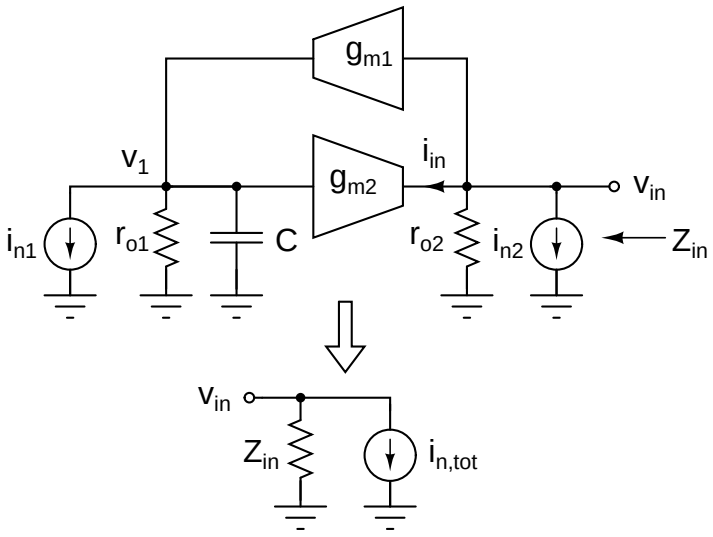


Figure 2-48: The simple gyrator model with noise sources added.

where $j = 1$ or 2 and N_j (known as the effective number of noise sources) is a constant that depends on the topology and biasing details of the transconductor. The total noise current PSD present at the input terminal can be found by superposition. It is given by

$$\overline{i_{n,tot}^2} = \overline{i_{n1}^2} \left| \frac{r_{o1}}{sCr_{o1} + 1} \right|^2 g_{m2}^2 + \overline{i_{n2}^2} \tag{2.74}$$

The expression above may be simplified to read

$$\overline{i_{n,tot}^2} = 4kTg_{m2}\left[\frac{N_1 r_{o1}}{(Q^2+1)\,R_{eff}} + N_2\right] = \frac{4kT}{R_{eff}\,(Q^2+1)}\left[N_1 r_{o1} + N_2 R_{eff}\left(Q^2+1\right)\right] \tag{2.75}$$

Let us now consider the noise current produced by a *passive* series L-R circuit with the same values of inductance and resistance (L_{eff} and R_{eff}, respectively) as the active circuit. The situation is shown in the left-hand circuit in Figure 2-49. The noise current PSD $\overline{i_{n,Reff}^2}$ developed across the resistor is given by the well-known formula

$$\overline{i_{n,Reff}^2} = \frac{4kT}{R_{eff}} \tag{2.76}$$

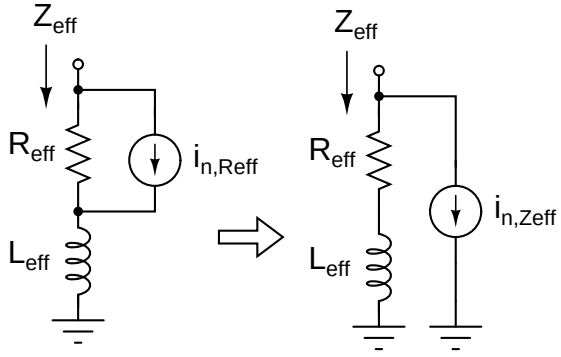


Figure 2-49: The noise model of a passive inductor L_{eff} with some series resistance R_{eff}.

In order to convert this current source into one that is in parallel with the entire impedance $Z_{eff} = R_{eff} + sL_{eff}$ we can Thevenize the current source and then convert the voltage source back into a Norton equivalent. This process results in the situation shown in the right-hand circuit in Figure 2-49. The noise current PSD $\overline{i_{n,Zeff}^2}$ developed across Z_{efff} is given by

$$\overline{i_{n,Zeff}^2} = \frac{4kTR_{eff}}{|R_{eff} + sL_{eff}|^2} = \frac{4kT}{R_{eff}\,(Q^2+1)} \tag{2.77}$$

This result is similar to that of a cascode. As Q, which is the ratio of the impedances sL_{eff} and R_{eff}, increases most of the noise current produced by the

resistor continues to circulate inside the local loop formed by $\overline{i_{n,Reff}^2}$ and R_{eff}. Very little flows through L_{eff}, the higher-impedance path, and eventually makes its way to the output.

Finally, let us compare the noise current PSD produced by the active circuit (2.75) and the passive circuit (2.77). The ratio of the two quantities is given by

$$\beta \equiv \frac{\overline{i_{n,tot}^2}}{\overline{i_{n,Zeff}^2}} = \left[N_1 r_{o1} + N_2 R_{eff}\left(Q^2+1\right)\right] g_{m2} \tag{2.78}$$

This expression for β can be simplified to read

$$\beta = N_2\left[\left(\frac{N_1 g_{m2}}{N_2 g_{m1}}\right) A_{v1} + \frac{Q^2+1}{A_{v1}}\right] \tag{2.79}$$

where $A_{v1} = g_{m1} r_{o1}$ is the DC voltage gain of the first transconductor. In a narrow-band application where Q is approximately constant, the expression above can be viewed as a function of the variable A_{v1}. If we use standard calculus and minimize it, we get

$$A_{v1} = \sqrt{\frac{Q^2+1}{\left(\frac{N_1 g_{m2}}{N_2 g_{m1}}\right)}} \tag{2.80}$$

When A_{v1} is set to this value, β is minimized and is given by

$$\beta_{min} = 2\sqrt{N_1 N_2 \alpha \left(Q^2+1\right)} \tag{2.81}$$

where the parameter $\alpha = g_{m2}/g_{m1}$. We see that we need extreme g_m scaling, i.e. setting $\alpha \ll 1$, to make the noise from the active circuit comparable to that from the passive circuit, especially when Q is large. Finally, it should be noted that the reason r_{o1} and r_{o2} contribute no noise of their own to this analysis is that they are small-signal (AC) impedances, and thus noiseless.

We now study how the noise PSD of the active circuit trades off with its power consumption. The total power consumed by the active inductor is given by

$$P = V_{DD}\left(I_{B1} + I_{B2}\right) = V_{DD}\left(g_{m1}V_{L1} + g_{m2}V_{L2}\right) \tag{2.82}$$

where I_{B1} and I_{B2} are the DC bias currents of the two transconductors, and V_{L1} and V_{L2} are their linear ranges. The expression above can be simplified to

$$P = \frac{V_{DD}}{A_{v1}R_{eff}}\left(\frac{V_{L1}}{\alpha} + V_{L2}\right) \tag{2.83}$$

If A_{v1} has been set the value given by (2.80) to ensure the minimum noise PSD in a narrowband application, the power consumption becomes

$$P = \frac{V_{DD}V_{L2}}{R_{eff}}\sqrt{\frac{N_1/N_2}{Q^2+1}}\left(\frac{V_{L1}}{V_{L2}}\frac{1}{\sqrt{\alpha}} + \sqrt{\alpha}\right) \tag{2.84}$$

Assuming that $\alpha \ll 1$ to reduce noise,

$$P \approx \frac{V_{DD}V_{L1}}{R_{eff}}\sqrt{\frac{N_1/N_2}{\alpha\left(Q^2+1\right)}} \tag{2.85}$$

Comparing (2.85) with (2.81), we see that

$$P = \left(\frac{2V_{DD}V_{L1}N_1}{R_{eff}}\right)\frac{1}{\beta_{min}} \tag{2.86}$$

In other words, there is a direct trade-off between noise and power consumption: the product of the two is a constant. Therefore we need to burn more power to lower the noise PSD of the active inductor L_{eff}. Also note that $P \propto V_{L1}$, the linear range of g_{m1}, i.e. we also need to burn more power to increase the linear range of L_{eff}. High-dynamic-range active inductors are thus quite power-hungry.

Chapter 3

The Unidirectional RF Cochlea

Differential equations are powerful, for their interpretation is legion, and they speak with many tongues.
– George Pólya, Mathematical Methods in Science

3.1 Introduction

We described Zweig's cochlear model in the previous chapter [328]. An important characteristic of this model is that it is a true bidirectional transmission line: waves can propagate in both directions. Among others, Hubbard, in [123] proposed another bidirectional cochlea model that involved two coupled transmission lines. These models have mostly been designed to reproduce experimental data as closely as possible; their circuit analogs are usually quite complicated and have not been implemented in hardware (though they have been simulated).

There is however, an alternative tradition in cochlear modeling. This approach has generally been pursued by electrical engineers who are more interested in capturing certain aspects of cochlear behavior than in precisely modeling the biological system. Most of these workers have implemented unidirectional versions of the cochlea. An important motivation is the fact that the biological cochlea appears to only support forward traveling waves, i.e., is effectively unidirectional, except, possibly, for the production of otoacoustic emissions [271]. This is because the wave amplitude dies

almost completely before it reaches the apex, where the basilar membrane terminates. As a result reflected waves are insignificant. In these systems cochlear behavior is simulated using a cascade of (unidirectional) low pass filters with exponentially tapered cutoff frequencies [177, 259].

For example, the unidirectional cochlea described in [259] used all-pole, second order resonant filter transfer functions of the normalized form

$$H_n(s_n) = \frac{1}{s_n^2 + s_n/Q + 1} \tag{3.1}$$

where $s_n = s/\omega_c(i)$, with $\omega_c(i) = \omega_c(0)\exp^{-i/N_{nat}}$, the center frequency of the i-th filter in the cascade, decreasing exponentially with i (there are N_{nat} filters per e-fold of frequency). In my opinion, this silicon cochlea remains the best on-chip implementation published to date. However, it suffered from limited frequency resolution, low signal-to-noise ratio (SNR), and excessive phase lag and group delay as the signal propagated along the cascade of filters. One way to potentially improve performance is to find a sound physical basis for selecting a particular form for $H_n(s_n)$.

3.2 Filter Cascade Design

The transfer functions $H(s_n)$ for filters in the unidirectional cochlea can be derived from a WKB-type solution of the wave equation by making a series of further approximations [188]. The essence of cochlear operation is collective amplification, as exemplified by the exponential part of the transfer function shown in (2.22). For simplicity, therefore, we ignore $|k_n|^{3/2}$, the pre-exponential term. The reflected wave is also neglected, making the structure unidirectional. The exponential term is modeled by breaking up the integral, which extends from 0 to s_n, into small parts extending from s_{i-1} to s_i, where i is an integer:

$$\exp\left(-\int_0^{s_n} k_n ds'\right) = \prod_i \exp\left(-\int_{s_{i-1}}^{s_i} k_n ds'\right) \equiv \prod_i H_i \tag{3.2}$$

We note that the expression above looks like the transfer function of a cascade of

unidirectional filters with transfer functions H_i. We assume that each filter models the action of a piece of transmission line Δx long. Thus, the i-th filter models the piece of line between $x = (i-1)\Delta x$ and $x = i\Delta x$. By using the definition of s_n, we have

$$s_i = \frac{j\omega}{\omega_c(0)\exp(-i\Delta x/l)} = s_0 \exp(i/N_{nat}) \tag{3.3}$$

where $s_0 = j\omega/\omega_c(0)$. Thus, the values of s_i increase exponentially, i.e., proportional to $\exp(i/N_{nat})$. In other words, we have N_{nat} filters per e-fold in frequency. Now define $\Delta s_i = s_i - s_{i-1}$. If N_{nat} is large enough, we may assume that k_n remains approximately constant between s_i and s_{i-1}. Therefore the integral that defines H_i can be simplified to

$$H_i \approx \exp\left(-\Delta s_i \times k_n\right) \tag{3.4}$$

If $N_{nat} \gg 1$, $|\Delta s_i \times k_n| \ll 1$ and each transfer function can be approximated by using the identity $\exp(-x) \approx 1/(1+x)$ since $|x| << 1$. Also, we may write $\Delta s_i \approx (\Delta x)(ds_n/dx) = s_i/N_{nat}$. Therefore H_i is given by

$$H_i \approx \frac{1}{1 + \frac{s_i k_n(s_i)}{N_{nat}}} \tag{3.5}$$

We see that each transfer function is only a function of s_i, which scales exponentially along the cascade, and N_{nat}, which is constant, but can differ from its value in the bidirectional cochlea. Therefore the transfer functions are simply frequency-scaled versions of each other and we can represent all of them using the single normalized frequency variable s_n. By substituting in $k_n = N_{nat}\sqrt{Z_n Y_n}/s_n$, we get the following normalized transfer function:

$$H_n\left(s_n\right) = \frac{1}{1 + \sqrt{Z_n Y_n}} \tag{3.6}$$

Our new unidirectional cochlea model, shown in Figure 3-1, is thus a cascade of exponentially scaled filters with normalized transfer functions (in either voltage or

current) given by (3.6), where $s_n = s/\omega_c(i)$ and $\omega_c(i) = \omega_c(0)\exp^{-i/N_{nat}}$. In general, however, $\sqrt{Z_n Y_n}$ is not a rational function, which means that $H_n(s_n)$ cannot be implemented using a finite number of lumped components.

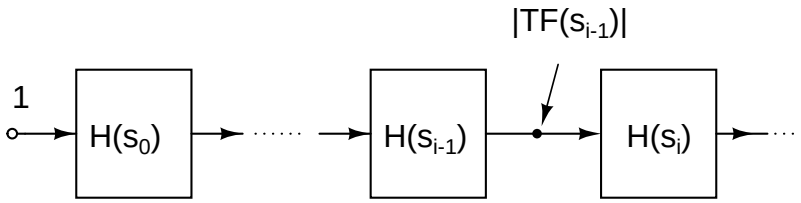


Figure 3-1: Unidirectional cochlea model, consisting of a exponentially scaled cascade of filters.

A way out is shown by the observation that the overall magnitude and phase response shapes of (2.12) are insensitive to the value of Q. In particular, we can use $Q = 0.5$ while keeping the responses qualitatively similar. In this case, the two zeros of $Y_n(s_n)$ coincide on the real axis. We can then complete a square in the numerator, i.e., $(s_n^2 + \mu s_n/Q + \mu^2) = (s + \mu)^2$ when $Q = 0.5$. This choice makes $Z_n Y_n$ a perfect square, and thus the normalized filter transfer function $H_n(s_n)$ becomes rational and is given by

$$H_n(s_n) = \frac{s_n^2 + 2ds_n + 1}{\left(1 + \sqrt{Q_{line}}/\mu\right) s_n^2 + \left(2d + \sqrt{Q_{line}}\right) s_n + 1} \tag{3.7}$$

where $Q_{line} = \omega_c(0)L_0/Z_0$, as in the previous chapter. We used the following parameter values in our unidirectional cochlea design: $d = 0.1$, $\mu = 0.3$ and $Q_{line} = 0.5$ [188]. As μ decreases the peak gain of the unidirectional cochlea increases but the filter poles become more and more under-damped. Our filter TF is shown in Figures 3-2(a) and 3-2(b). Since it contains a pair of poles and a pair of complex zeros, it differs from the all-pole TF's previously used to build audio-frequency silicon cochleas [177, 259]. Gain compression for large signals can be modeled by making d an increasing function of the local signal amplitude, $|A|$. For example, one can use $d = d_{min} + \sigma|A|$, where σ is a constant [323].

In addition to reducing group delay, the figure shows that the zeros also result in an asymmetric frequency response close to the peak of the TF, with a sharper drop-

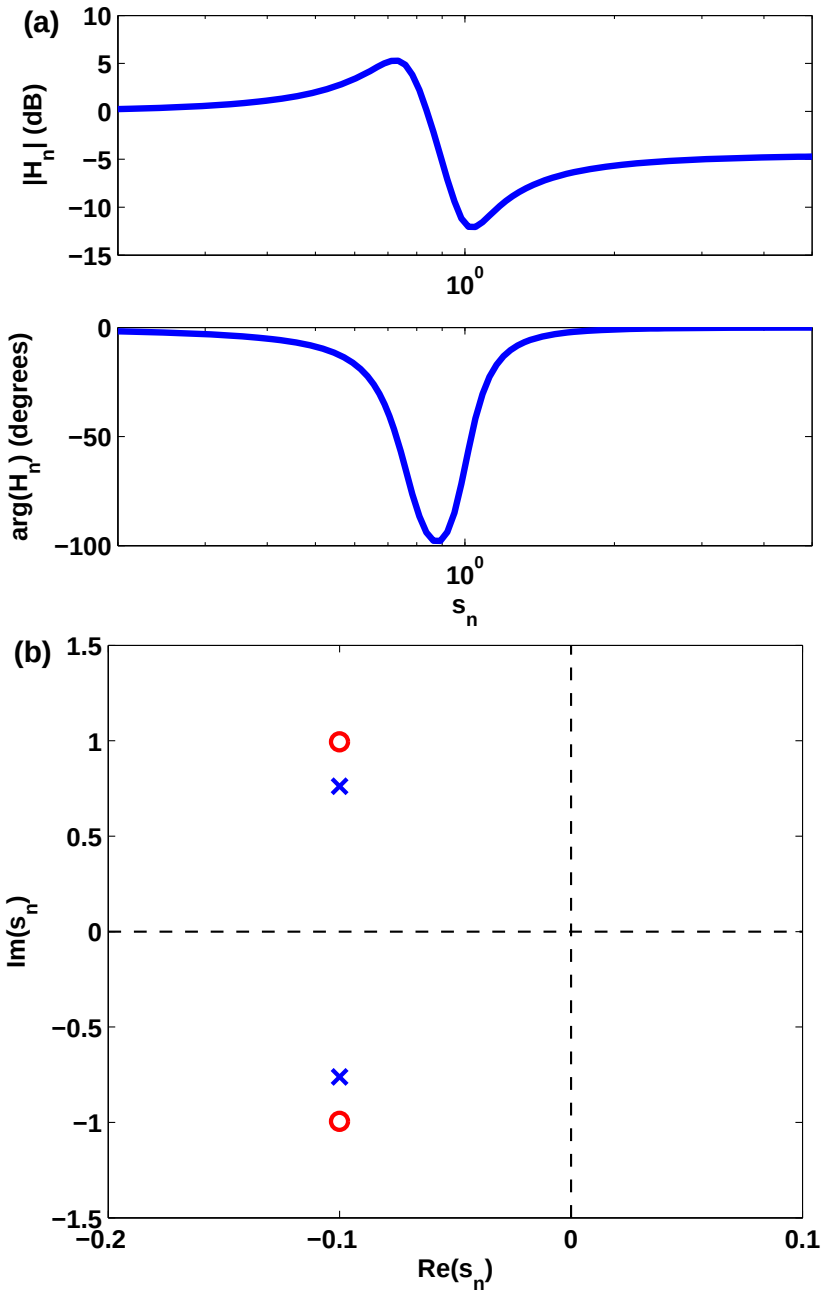


Figure 3-2: (a) Normalized TF H_n used in the unidirectional cochlea, (b) pole-zero plot for H_n.

off on the high-frequency side that increases the frequency resolution of the cochlear TFs. Figure 3-3 compares simulated frequency responses of unidirectional cochleas using all-pole filters [259] (Equation 3.1, with $Q = 1.3$) and our modified filters (Equation 3.7, with the parameters above). These parameters were chosen such that the peak gains of both designs were approximately equal, thereby facilitating comparison. We see that including complex zeros in the filter transfer function increases cochlear frequency resolution by increasing the effective quality factor of the cochlear transfer function.

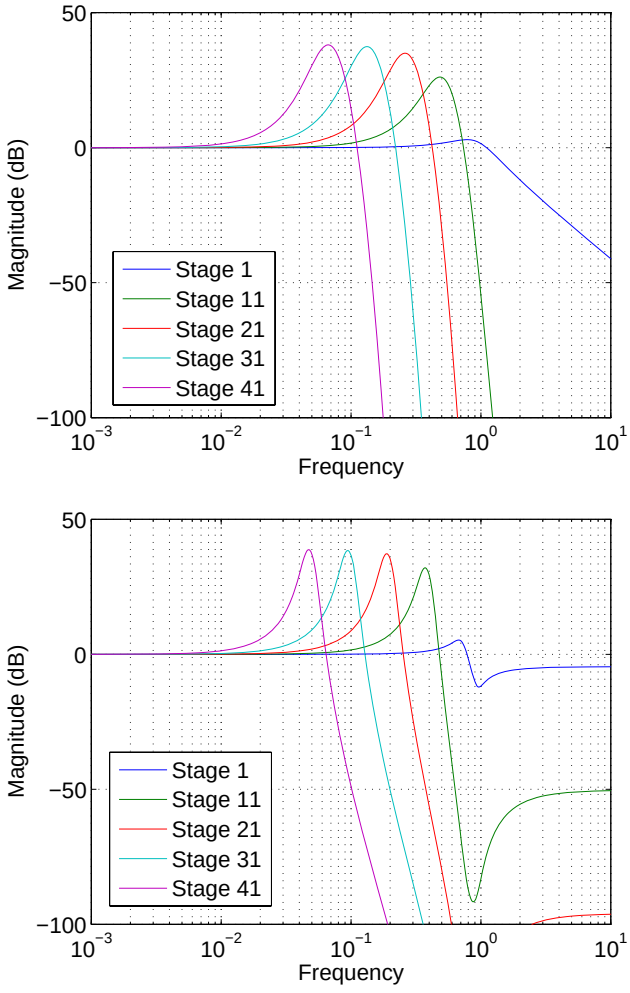


Figure 3-3: Unidirectional cochlea frequency response magnitudes at different spatial locations along the cascade, (a) using the original (all-pole) filter transfer function, and (b) using the modified filter transfer function.

The filter transfer function actually synthesized and implemented on-chip included

an additional high-frequency zero and two additional high-frequency poles. These additional poles and zeros were placed at $s_n \approx 5$. They produce a first-order roll-off at high frequencies which does not have a significant effect on the cochlear transfer function. However, they do make the circuit realizable with integrated passive components, which have finite shunt capacitances to ground. In particular, such components cannot synthesize the ideal filter transfer function shown in Figure 3-2, because it does not roll-off at high frequencies, i.e., has infnite bandwidth.

Finally, one of the key ideas involved in the design of any cochlea-like cascade of filters is distributed gain control. The cascaded nature of the system means that correlated changes in the properties of a large number of stages (such as the parameter Q in (3.1)) can cause large changes in the behavior of the whole system. By the same token, the system is extremely robust to uncorrelated parametric fluctuations between individual stages; the overall transfer function is effectively determined by a "spatial average" over the properties of about N_{nat} stages.

3.3 Theoretical Analysis

3.3.1 Transfer Functions

The filter transfer function $H(s_n)$ shown in (3.7) has a low-frequency magnitude of 1 and a high-frequency magnitude of $1/\left(1+\sqrt{Q_{line}}/\mu\right) < 1$. We shall denote its maximum magnitude by G_{stage}. The value of G_{stage} eventually sets the maximum value (peak) of the cochlear transfer function. The behavior of $H(s_n)$ around $\omega_n = 1$ can be understood by considering the numerator and denominator separately. The numerator contributes a pair of complex zeros with high quality factor $1/(2d)$. These zeros produce a notch in the frequency response, with $|H(s_n)|$ reaching its minimum value at a frequency $\omega_z \approx 1$. The numerator contributes a pair of complex poles with quality factor given by

$$Q_p = \frac{\sqrt{1+\sqrt{Q_{line}}/\mu}}{2d+\sqrt{Q_{line}}} \tag{3.8}$$

Typically $Q_p < 1/(2d)$, i.e., the poles have lower quality factor than the zeros. In the absence of the zeros the poles would cause the magnitude of the transfer function to reach a peak value of $G_{stage} = Q_p$ at a frequency $\omega_p = 1/\sqrt{1 + \sqrt{Q_{line}}/\mu} < 1$. The nearby zeros reduce the actual value of G_{stage}. The transfer function contains three parameters: d, μ and Q_{line}. The parameter d controls the quality factor of both the zero and the pole pair, while μ controls the pole frequency ω_p. Decreasing either d or μ causes G_{stage} to increase. In the case of μ the increase occurs because ω_p decreases, which moves the peak further away from the notch at ω_z. On the other hand Q_{line} has relatively little effect on G_{stage}. This is because Q_{line} affects the transfer function in two ways that tend to cancel themselves out. Increasing Q_{line} decreases ω_p, which tends to increase G_{stage}, but also increases the damping of the poles, which tends to reduce it.

The transfer function of the cochlea after the n-th stage is characterized by its peak gain $G(n)$ and quality factor. We use both traditional definitions of quality factor: center frequency divided by the 3dB or 10dB bandwidth, resulting in $Q_{3dB}(n)$ and $Q_{10dB}(n)$, respectively. In addition to the filter parameters d, μ and Q_{line}, we need N_{nat}, the number of stages per e-fold, to complete the cochlear design. Since each cochlear transfer function may be approximated as the product of N_{nat} identical filter transfer functions, we expect $G(n)$ for $n > N_{nat}$ to be constant and given by

$$G \approx (G_{stage})^{N_{nat}} \Rightarrow G_{dB} = N_{nat} G_{stage,dB} \tag{3.9}$$

where the "dB" subscripts indicate that G and G_{stage} are being expressed in dB. Thus, a plot of G (in dB) versus N_{nat} should be a straight line with a slope of $G_{stage,dB}$. Since $G_{stage} > 1$ the peak gain increases rapidly with N_{nat}. In fact, this behavior is independent of the precise shape of the filter transfer function, since it depends only on exponential scaling of center frequencies. Using (2.47), we also expect Q_{3dB} and Q_{10dB} to vary roughly as $\sqrt{N_{nat}}$.

Figure 3-4 shows simulated values of $G(n)$, $Q_{3dB}(n)$ and $Q_{10dB}(n)$ for $N_{nat} = 14$, and typical filter parameters ($d = 0.1$, $\mu = 0.2$ and $Q_{line} = 0.5$). The peak filter

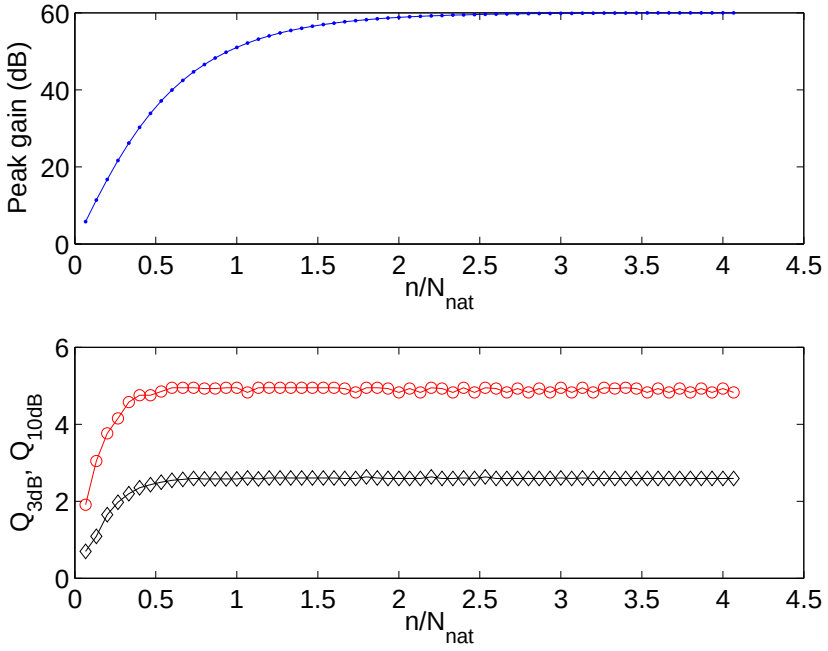


Figure 3-4: Behavior of the unidirectional cochlea transfer functions as a function of position n along the cascade. Top: peak gain, bottom: 3dB (red) and 10dB (black) quality factors Parameters used to draw this plot were: $N_{nat} = 14$, $d = 0.1$, $\mu = 0.2$ and $Q_{line} = 0.5$.

gain $G_{stage} \approx 1.95$ (5.8dB) for these parameter values. We see that $G(n)$ asymptotes to a fixed value for values of n greater than approximately $2N_{nat}$. We would expect this asymptotic gain to be approximately $14 \times G_{stage,dB} = 81$dB. The actual value is somewhat lower, being about 60dB. The figure shows that the quality factors also asymptote with n, but much faster. They essentially reach their final values for $n > N_{nat}/2$. Similar behavior occurs for other filter parameter values.

Figure 3-5 shows how the asymptotic gain and quality factors of the cochlear transfer functions, i.e., their values at large n, vary as a function of N_{nat}. As expected, G_{dB} is very nearly a straight line versus N_{nat}, while the quality factors increase approximately as $\sqrt{N_{nat}}$.

Figure 3-6 shows how the asymptotic gain and quality factors of the cochlear transfer functions, i.e., their values at large n, vary as a function of d, μ and Q_{line} for a fixed value of N_{nat}. As discussed earlier, decreasing d and μ causes G_{stage} to increase,

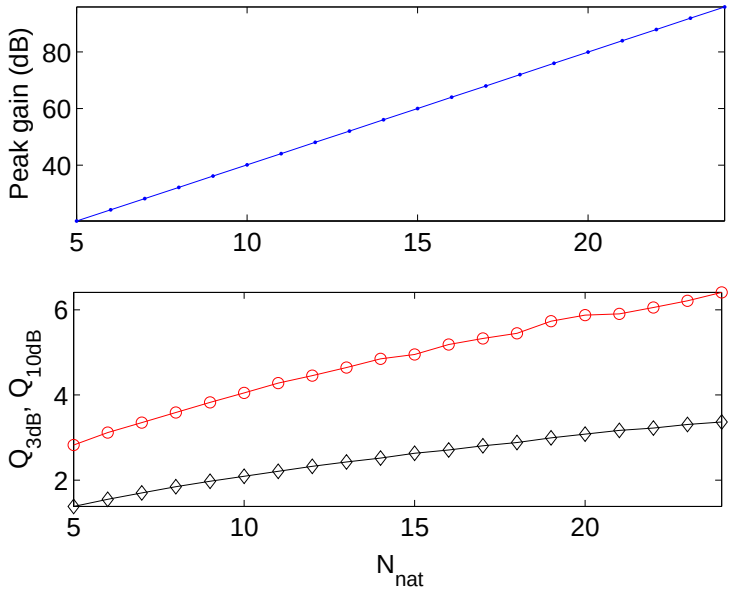


Figure 3-5: Asymptotic behavior of the unidirectional cochlea transfer functions as a function of N_{nat}. Top: peak gain, bottom: 3dB (red) and 10dB (black) quality factors. Parameters used to draw this plot were: $d = 0.1$, $\mu = 0.2$ and $Q_{line} = 0.5$.

while Q_{line} has relatively little effect on G_{stage}. Thus, the cochlear transfer function gain G should increase sharply as d and μ decrease, while remaining approximately independent of Q_{line}. Figure 3-6 confirms this expected behavior.

3.3.2 Noise

In this section we analyze the noise performance of the unidirectional cochlea. For simplicity, each filter is assumed to be strictly unidirectional. The n-th stage filters the output noise voltage $v_{noise}(n-1)$ produced by the previous, $(n-1)$-th stage, and adds some noise of it's own:

$$\overline{v_{noise}^2(n)} = \overline{v_{noise}^2(n-1)}\,|H_n(j\omega)|^2 + \overline{v_n^2} \tag{3.10}$$

where v_n is the noise PSD produced by the n-th stage, and $H_n(j\omega)$ is its transfer function. Note that we are now using real, un-normalized frequencies. Because H_n is an exponentially-frequency scaled version of a common prototype, we have $H_n(\omega) = H_1\left(\omega e^{-(n-1)/N_{nat}}\right)$. We shall assume that all the filtering within each stage is provided

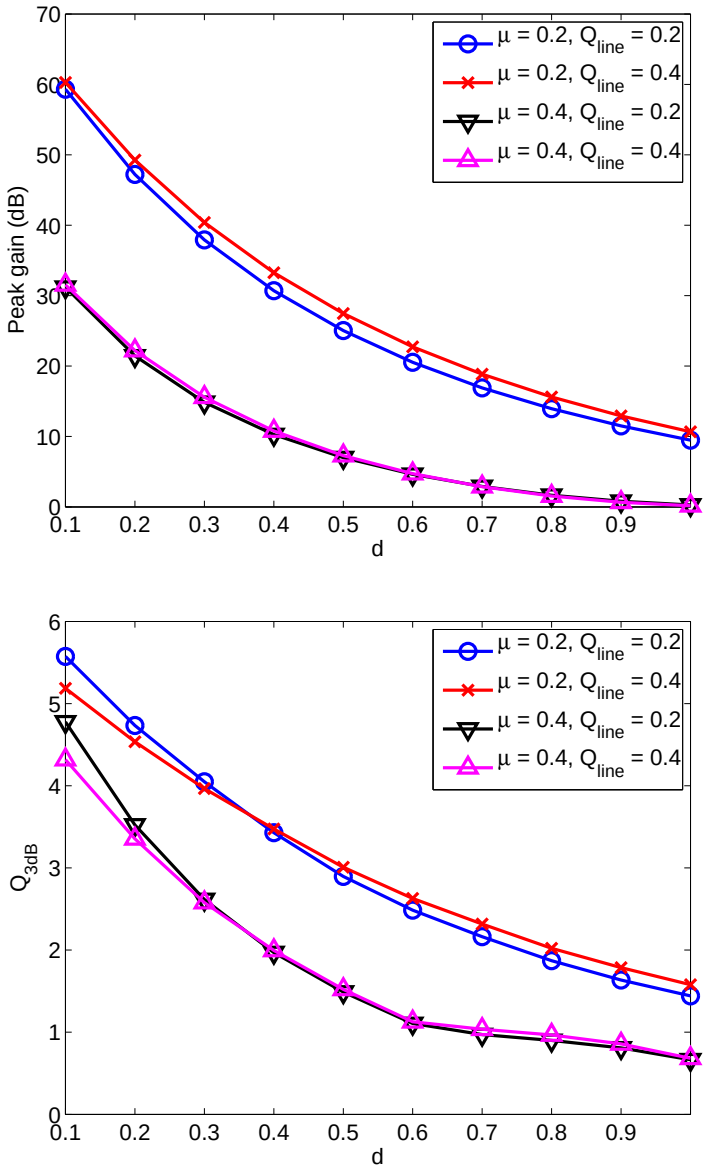


Figure 3-6: Asymptotic behavior of the unidirectional cochlea transfer functions as a function of d, μ and Q_{line}. Top: peak gain, bottom: 3dB quality factor. The value of $N_{nat} = 15$ was fixed for this plot.

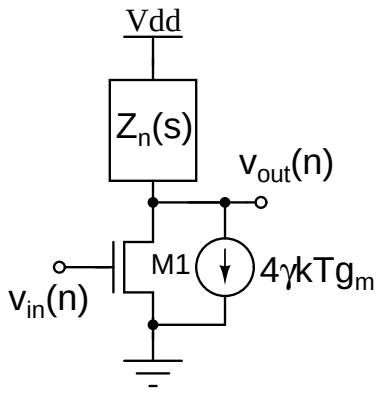


Figure 3-7: Simplified cochlear filter circuit, used for noise calculations.

by a passive impedance $Z_n(s)$ which contributes a negligible amount of noiee to v_n. A single transistor M_1, shown in Figure 3-7 provides buffering and voltage gain. It converts input voltage into current so that it can be filtered by $Z_n(s)$, and is modeled as an ideal transconductance of value g_m. The value of g_m is assumed to be the same for each filter, i.e., all values of n.

We will ignore flicker noise, because its effects are typically negligible over our operating frequency range. Therefore we can assume that each transistor adds a white noise current with $PSD = 4\gamma kTg_m$ to the output node, where γ, known as the excess noise factor, has a value of $2/3$ for long-channel transistors. The filter circuit actually implemented on-chip was somewhat different from the circuit shown in Figure 3-7. However, we shall study this simplified circuit, because its noise properties are very similar to the actual circuit, while requiring considerably less algebra to analyze.

The voltage PSD contributed by the simplified filter circuit to its output node is given by

$$\begin{aligned} \overline{v_n^2} &= 4\gamma kTg_m \left|Z_n(j\omega)\right|^2 \\ &= \frac{4\gamma kT}{g_m} \left|H_n(j\omega)\right|^2 \end{aligned} \tag{3.11}$$

where $H_n(s) = g_m Z_n(s)$ is the filter's transfer function. Combining (3.10) and (3.11), we get the net PSD at the output of the n-th filter:

$$\overline{v_{noise}^2(n)} = \left[\overline{v_{noise}^2(n-1)} + \frac{4\gamma kT}{g_m}\right] |H_n(j\omega)|^2 \tag{3.12}$$

We can use the fact that $v_{noise}(0) = 0$ (i.e., the input is noiseless) and the iteration formula shown in (3.12) to explicitly write down $\overline{v_{noise}^2(n)}$:

$$\overline{v_{noise}^2(n)} = \frac{4\gamma kT}{g_m} \sum_{i=1}^{n} \left(\prod_{j=n+1-i}^{n} |H_j(j\omega)|^2 \right) \equiv \frac{4\gamma kT}{g_m} H_{noise,n}(j\omega) \tag{3.13}$$

where $H_{noise,n}(j\omega)$ is defined as the noise transfer function of the n-th stage. The noise transfer functions $H_{noise,n}$ of the cochlea look very similar to the squared magnitude of the signal transfer functions, which are simply given by

$$H_{signal,n} = \prod_{i=1}^{n} H_i(j\omega) \tag{3.14}$$

The total noise voltage can be found by integrating the PSD over all frequencies:

$$\begin{aligned} \overline{v_{tot}^2(n)} &= \int_0^\infty \overline{v_{noise}^2(n)} df \\ &= \frac{4\gamma kT}{g_m} \int_0^\infty H_{noise,n}(f) df \\ &\equiv \frac{4\gamma kT}{g_m} H_{noise,tot} \end{aligned} \tag{3.15}$$

where $2\pi f = \omega$, as usual. The quantity $H_{noise,tot}$, which is also a function of n, has dimensions of bandwidth, and is plotted for typical filter parameters and various values of N_{nat} in Figure 3-8. Similar curves are obtained for a wide range of filter parameters. We see that the total noise voltage, which is proportional to $\sqrt{H_{noise,tot}}$, increases with increasing n, reaches a maximum around $n = 2N_{nat}$, and then slowly decreases. This behavior may be explained as a compromise between two competing tendencies. The noise PSD tends to increase with increasing n because all stages before it contribute noise. On the other hand the bandwidth decreases as $\exp(-n/N_{nat})$, which tends to reduce the total integrated noise.

The *minimum detectable signal*, $v_{mds}(n)$ is defined as the amplitude of the smallest

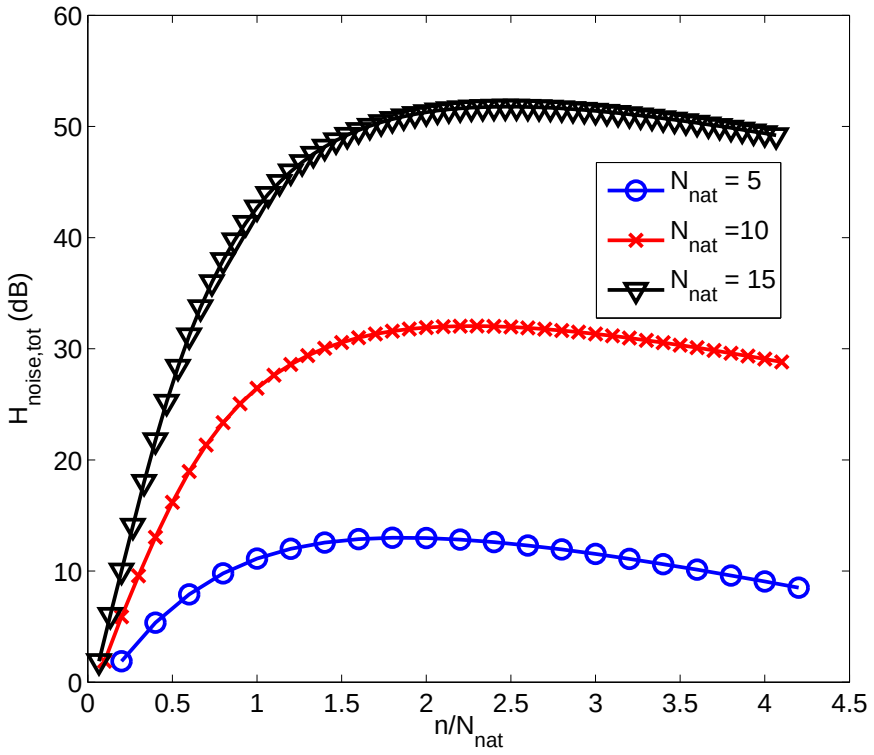


Figure 3-8: Effective noise bandwidth in the unidirectional cochlea as a function of position n along the cascade for various values of N_{nat}. Parameters fixed for this plot were: $d = 0.1$, $\mu = 0.2$ and $Q_{line} = 0.5$.

input signal that just exceeds the output noise of the n-th stage. The inverse of the minimum detectable signal strength is often referred to as the *sensitivity* of the system. The value of $v_{mds}(n)$ reaches a minimum when the frequency of this signal is equal to the center frequency of the n-th stage. In this case, we have $v_{mds}(n)G(n) = v_{tot}(n)$, i.e.,

$$v_{mds}(n) = \frac{v_{tot}(n)}{G(n)} \tag{3.16}$$

Since $v_{tot}(n) \propto 1/\sqrt{g_m}$, we can clearly burn power to increase g_m, lower the minimum detectable signal and increase sensitivity. For the purpose of comparing cochlear designs with different parameter values we wish to eliminate this dependence on g_m. Since g_m is assumed to be constant, a simple way is to define a normalized version of v_{mds}, given by $\sqrt{H_{noise,tot}}/G(n)$, that is proportional to the actual value but independent on g_m. Figure 3-9 plots this normalized version of v_{mds} as a function of n for various values of N_{nat}. The figure shows that changing N_{nat} leaves the minimum detectable signal essentially unchanged. This is because N_{nat} affects the signal and noise transfer functions in similar ways, leaving their ratio, which sets v_{mds}, fixed. We also see that v_{mds} varies in a complex way with n, in general decreasing slowly as n increases. This behavior is explained by the fact that the signal and noise transfer functions vary in similar, but not identical, ways with n. The former is caused by n filters acting sequentially on a single signal that is fed into the *first* filter, while the latter is caused by the same n filters acting on "noise signals" that are distributed, i.e., fed into *each* filter.

3.3.3 SNR and Dynamic Range

The range of signals that can be handled by the unidirectional cochlea is limited at the low end by noise, and at the high end by the linear range of the filters within it. There is only one active component in the filter circuit shown in Figure 3-7, namely the transistor M_1. The linear range of M_1, given by $V_L = I/g_m$, is approximately equal to that of the filter. Here I is the DC bias current flowing through M_1. The

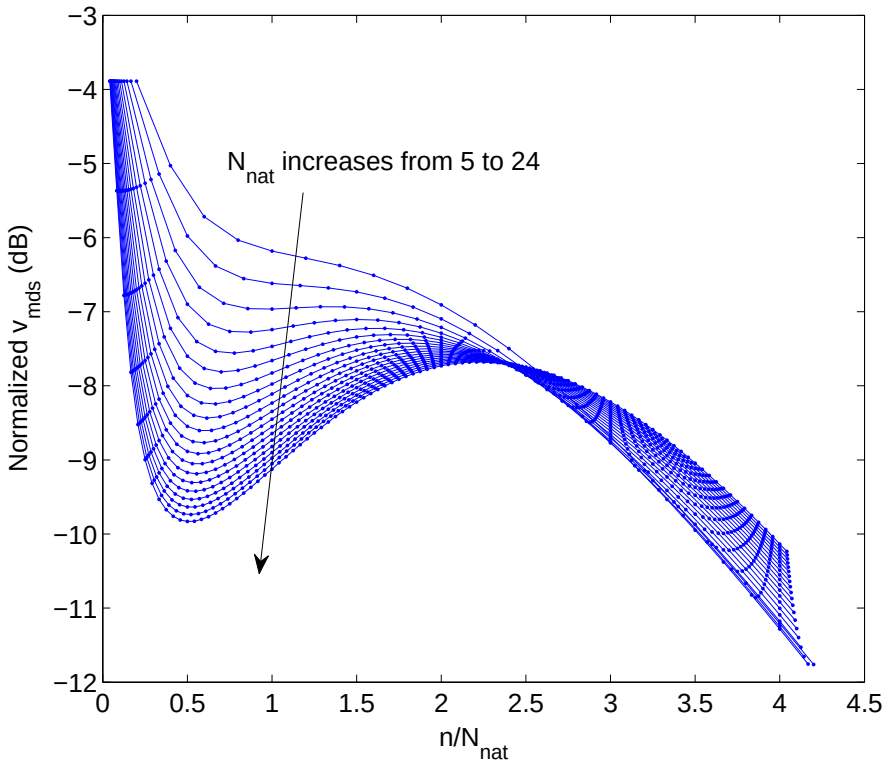


Figure 3-9: Normalized minimum detectable signal in the unidirectional cochlea as a function of position n along the cascade for various values of N_{nat}. Parameters used to draw this plot were: $d = 0.1$, $\mu = 0.2$ and $Q_{line} = 0.5$.

value of V_L is independent of I when M_1 is in sub-threshold, and given by ϕ_T/κ. It increases $\propto \sqrt{I}$ above threshold.

The maximum unidistorted signal amplitude at the output of the n-filter is V_L. As a result, the maximum signal-to-noise ratio (SNR) of the cochlear outputs is given by

$$\mathrm{SNR}_{max}(n) = \left(\frac{V_L}{v_{tot}(n)}\right)^2 \tag{3.17}$$

As long as the cochlea remains completely linear the dynamic range (DR) of input signals it can handle is equal to SNR_{max}, i.e., DR = SNR_{max}. However, the biological cochlea, and other silicon cochleas published in the literature, use gain control to increase input-referred dynamic range. Gain control reduces harmonic distortion at the cost of reduced sensitivity to small signals. In the cochlea, distortion generates harmonics of the input frequency that propagate, and also interact with each other to produce additional intermodulation products. The response of the system to these additional frequencies causes spurious peaks to appear in the cochlear transfer functions.

In the most common gain control technique, the peak gain $G(n)$ of the cochlear transfer functions is reduced for large input signals. This is a form of distributed, or parallel gain control that uses the filtering provided by the cochlea to compress different frequency components by different amounts. An important advantage of the filter transfer function shown in (3.7) is that it allows gain control to be implemented in a very simple way: the single parameter d can be increased with local signal amplitude $|A|$, thus decreasing G_{stage} for large signals. For example, we can increase d linearly with A, i.e., as $d = d_{min} + \sigma|A|$, where σ is a constant. It can be shown, as long as $d \ll 1$, that this simple, purely local technique mimics biology in keeping the timing of zero-crossings in the cochlear impulse responses invariant with $|A|$ [323].

An alternative technique is broadband gain control, where the amplitude of the input to the cochlea is used to equally attenuate all input frequencies. In this strategy, which does not use the cochlea at all, the strongest frequency component will

dominate the input amplitude, and thus set the overall level of compression. Biology seems to use a combination of both strategies. The stapedius reflex, which quickly disconnects the middle ear from the cochlea for loud sounds, is a form of broadband gain control, while distributed gain control is provided by outer hair cells.

In the presence of gain control, the gain seen by a small-amplitude tone (frequency component) will be reduced by the simultaneous presence of a large-amplitude tone, because the latter will dominate the total amplitude (the total amplitude usually determines the amount of gain compression). Thus, the sensitivity of the system to the smaller tone will be diminished. This phenomenon is known as *tone-to-tone supression*, and, in the auditory system, causes a psycho-acoustic phenomenon known as *masking*. The traveling wave structure of the cochlea leads to *asymmetric masking*: Low-frequency tones pass through most of the cochlea before peaking, and therefore suppress the gain of the filters that are tuned to higher frequencies. On the other hand, high-frequency tones peak early and are then filtered out, and therefore cannot affect the gain of later filters tuned to lower frequencies.

We can define the dynamic range of the cochlea by supposing that, at the maximum acceptable level of gain compression the gain of the n-th stage is reduced from $G(n)$, its value for small signals, to a new value $G_{comp}(n)$. The maximum allowable input signal amplitude is then increased from $V_L/G(n)$ to $V_L/G_{comp}(n)$, while the minimum detectable signal is unchanged, and given by $v_{tot}(n)/G(n)$. Thus the input-referred dynamic range becomes

$$\mathrm{DR} = \left(\frac{V_L/G_{comp}(n)}{v_{tot}(n)/G(n)}\right)^2 = \mathrm{SNR}_{max}\left(\frac{G(n)}{G_{comp}(n)}\right)^2 \tag{3.18}$$

The equation above is only valid for the largest tone in the input spectrum. In particular, of course, it is valid if there is only one tone present at the input. As described earlier, the simultaneous presence of other large tones will reduce the dynamic range, because they will turn down $G(n)$ from its small-signal value even when the tone in question is itself small.

What is an acceptable value for $G_{comp}(n)$? In RF systems the limit of linear

operation is often defined as the 1dB compression point, i.e., the input amplitude when $G_{comp,dB} = G_{dB} - 1\text{dB}$. This is an overly pessimistic limit for the cochlea, which is fundamentally a nonlinear system. A convenient upper bound for DR, which we define as DR_{max}, is obtained by setting $G_{comp}(n) = 1$, i.e., assuming that there is no peaking whatsoever. This value corresponds to a cochlea where frequency discriminability is only provided by the high-frequency roll-off slope of the transfer functions. Figure 3-10 shows simulated values of SNR_{max} and DR_{max} as a function of position n for various values of N_{nat} and typical values of other parameters. We see that SNR_{max} decreases as N_{nat} decreases, because the noise transfer functions increase while V_L remains fixed. However, DR_{max} is largely independent of N_{nat}. This behavior is not surprising; DR_{max} may be rewritten as $(V_L/v_{mds}(n))^2$, and, as shown by Figure 3-9, $v_{mds}(n)$ is approximately independent of N_{nat}. For the typical parameter values used to draw Figure 3-10, the average value of v_{mds} is approximately 50μV (-76dBm across a 50Ω load).

Figure 3-10 shows that we have to rely on increasing amounts of gain compression to get the full dynamic range available from the cochlea as N_{nat} increases. We expect similar behavior if the filter parameters d and μ are decreased for a fixed value of N_{nat}, because in either case G_{stage} increases, causing more peak gain in both the signal and noise transfer functions and reducing SNR_{max}. Figure 3-11 confirms our expectations. In this figure we have plotted SNR_{max} and DR_{max} versus d and μ for $N_{nat} = 14$ and typical values of other parameters. The gap between the two surfaces, which is the amount of gain compression required, increases sharply as d and μ decrease.

We designed our unidirectional cochlea to have $N_{nat} = 16$, $\text{SNR}_{max} \approx 35\text{dB}$ and $\text{DR} \approx 70\text{dB}$. The required values of d and μ were obtained from a plot similar to that shown in Figure 3-11.

3.3.4 Frequency Resolution

SNR_{max} is important because it limits the maximum frequency resolution of the cochlea. We can derive frequency resolution as follows. Consider a single input tone with a amplitude of A_1 and a frequency ω_1. We can measure A_1 by using an envelope

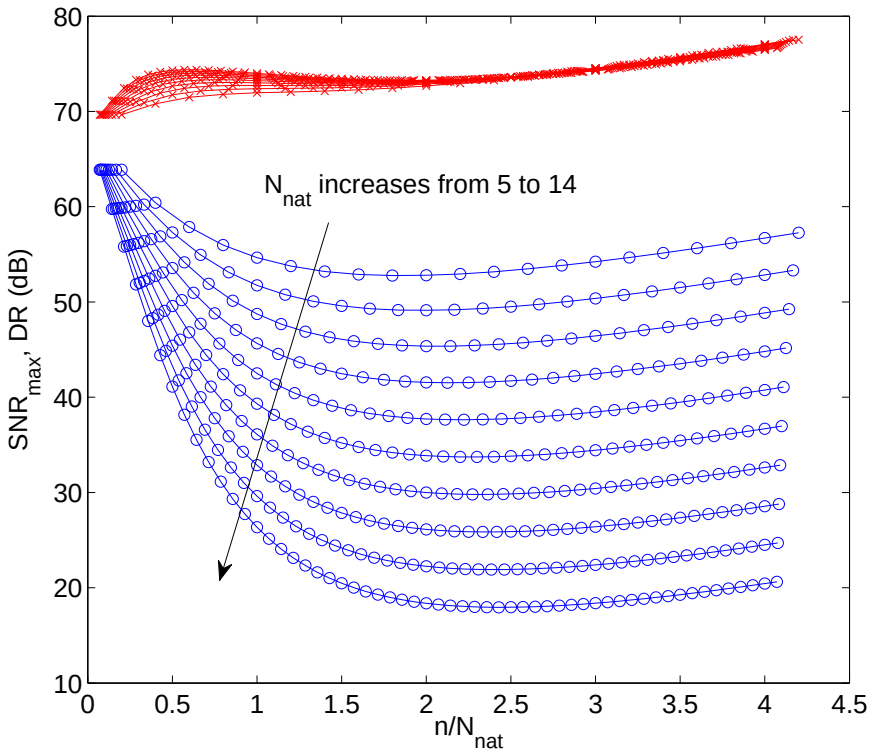


Figure 3-10: Maximum SNR (in blue) and input-referred dynamic range (DR, in red) of the unidirectional cochlea as a function of position n along the cascade for various values of N_{nat}. Parameters fixed for this plot were: $d = 0.1$, $\mu = 0.2$, $Q_{line} = 0.5$, $\gamma = 1$, $g_m = 1/60\mho$, $V_L = 150$mV and a maximum cochlea frequency of 6GHz.

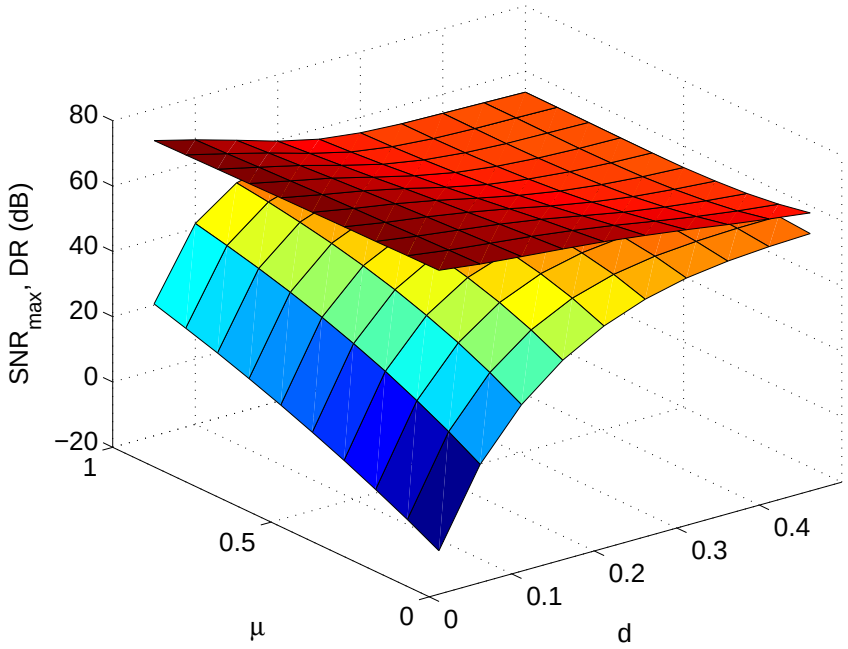


Figure 3-11: Lower limits of maximum SNR and DR of the unidirectional cochlea as a function of d and μ. Parameters fixed for this plot were: $N_{nat} = 14$, $Q_{line} = 0.5$, $\gamma = 1$, $g_m = 1/60\mho$, $V_L = 150\text{mV}$ and a maximum cochlea frequency of 6GHz.

detector (ED). In order to measure ω_1 we use another ED at the output of *any* filter, say the n-th one. The output of this ED is $A_1 \left|TF_n\left(\omega_1\right)\right|$. Since we know what A_1 is, this measurement tells us the value of $\left|TF_n\left(\omega_1\right)\right|$. Since the function $TF_n(\omega)$ is known, in a noiseless world, we can simply invert $\left|TF_n\left(\omega_1\right)\right|$ to find the exact value of ω_1. Simultaneously estimating the frequencies of multiple input tones is a much harder problem (with no general analytical solutions), and we do not discuss it further here.

In the real world the output of the filters has noise, which makes the measurement of $\left|TF_n\left(\omega_1\right)\right|$ imprecise. The measured value fluctuates with time, with a standard deviation that is $1/\sqrt{\text{SNR}}$ of its mean value. As a result, it is impossible to invert the function $TF_n(\omega)$ to determine ω_1 precisely, which limits the frequency resolution of the cochlea. In order to determine this quantity, which we denote by σ_ω, we assume that $|TF(\omega)| \propto 1/\omega^S$, i.e., the transfer function behaves like a filter of order S. We have

$$\frac{|TF(\omega_1)|}{|TF(\omega_1+\sigma_\omega)|} = \left(1+\frac{\sigma_\omega}{\omega}\right)^S \approx 1+\frac{S\sigma_\omega}{\omega} \tag{3.19}$$

where we have assumed that $\sigma_\omega \ll \omega$. Also, based on the definition of σ_ω, we have

$$\frac{|TF(\omega_1)|}{|TF(\omega_1+\sigma_\omega)|} = \frac{1}{1\pm 1/\sqrt{\text{SNR}}} \approx 1\mp\frac{1}{\sqrt{\text{SNR}}} \tag{3.20}$$

where we have assumed that SNR $\gg 1$. Combining (3.19) and (3.20), the normalized frequency resolution is given by

$$\frac{|S|\sigma_\omega}{\omega} = \frac{1}{\sqrt{\text{SNR}}} \Rightarrow \frac{\sigma_\omega}{\omega} = \frac{1}{|S|\sqrt{\text{SNR}}} \tag{3.21}$$

Clearly we should maximize SNR and $|S|$ to improve the frequency resolution. In the cochlea, the maximum value of $|S|$ is proportional to N_{nat}, and is found in the roll-offs on the high-frequency sides of the peaks of the transfer functions. The maximum value of SNR is given by SNR_{max} and occurs at the peaks of the transfer functions. Thus the best frequency resolution is obtained by observing the output of a filter that has a peak frequency just lower than the input frequency ω_1. We see that increasing N_{nat} affects the frequency resolution in two competing ways: $|S|$ increases, but SNR_{max} decreases (see Figure 3-10). The theoretical optimum that maximizes frequency resolution for tones that have $\text{SNR} = \text{SNR}_{max}$ is approximately $N_{nat} = 1/\ln(G_{stage})$, In practice most tones do not reach SNR_{max}, so we use much higher values of N_{nat}.

Finally, we note that the frequency resolution will be constant with SNR if $|S| \propto 1/\sqrt{\text{SNR}}$, i.e., the filter slopes decrease with increasing SNR. This type of adaptation is a form of *automatic Q control*. If SNR is high we can use broad, low-Q filters (low $|S|$), because they provide enough frequency selectivity but settle quickly, thereby increasing the speed of spectrum analysis. On the other hand, at low SNR we need narrow, high-Q filters (high $|S|$) to get the required frequency selectivity. However, these filters take longer to settle, reducing the speed of spectrum analysis. The biological cochlea appears to use a combination of automatic gain and Q control:

At large input amplitudes the cochlear transfer functions have lower gains but are also become significantly broader. Similar behavior is obtained by making d in the cochlear filters an increasing function of the local signal amplitude.

3.3.5 Power and Area Consumption

We have seen that SNR_{max} can be increased by increasing g_m, which costs more power. Since $H_{noise,tot}$ is proportional to bandwidth, another obvious way to reduce noise and improve SNR_{max} is to operate at lower frequencies where there is less bandwidth for the same value of Q. Another design constraint that we must meet is unity DC gain in the cochlear filters. This constraint arises from the fact that the underlying transmission line is a DC short. It ensures that filters with center frequencies much higher than the input frequency simply pass the input, and do not attenuate or amplify it. The transfer function shown in (3.7) meets this constraint, since $H_n(0) = 1$.

The filter circuit shown in (3-7) has a DC gain of $g_m Z(0) = g_m R_{stage}$, where R_{stage}, which must be purely resistive, is the DC value of $Z(s)$. Thus we must ensure that $g_m R_{stage} = 1$. The peak gain of the stage is given by

$$G_{stage} \approx g_m R_{stage} Q_c = \frac{\omega_c L_{stage}}{R_{stage}} = \omega_c L_{stage} g_m \tag{3.22}$$

where Q_c is the quality factor of $Z(s)$ at the center frequency ω_c, L_{stage} is proportional to the total inductance within the filter and we have used the fact that $g_m R_{stage} = 1$. We have seen that G_{stage} determines the peak gain of the cochlear transfer functions for a given value of N_{nat}. In order to maintain performance G_{stage} should be constant as g_m is reduced to save power consumption. Equation (3.22) then predicts that $L_{stage} \propto 1/g_m$, i.e., the sizes of the inductors must increase as g_m is decreased. However, on-chip inductors and transformers account for most of the layout area of the chip. Thus we have a direct trade-off between layout area and power consumption.

Since $Z(s)$ for each filter is a frequency-scaled version of a common prototype, the value of $L_{stage} \propto \omega_c$, i.e., increases as $\exp(n/N_{nat})$. Thus, the total inductance that

we must implement on chip is

$$L_{tot} = \sum_{n=1}^{N} L_{stage}(n) = L_{stage}(1) \sum_{n=1}^{N} \exp\left((n-1)/N_{nat}\right) \approx N_{nat} L_{stage}(N) \qquad (3.23)$$

where there are a total of N filters, and we have assumed that $N \gg N_{nat}$. The total chip area is proportional to L_{tot}. Equation (3.23) predicts that this area is inversely proportional to the lowest analysis frequency, $\omega_c(N)$ (which determines the value of $L_{stage}(N)$), and proportional to N_{nat}.

One can imagine varying the properties of the filters along the cascade, i.e., making d, μ, Q_{line} and g_m functions of n. There may be advantages to this approach. For example, it might be possible to saturate the peak gains of the cochlear transfer functions at lower values of n. However, we have not explored such schemes further. Finally, we have been considering voltage-mode implementations of the cochlea. The analysis is similar for current-mode implementations, with the important caveat that the linear range of the filters is no longer necessarily a constant. In particular, it can be varied with the signal envelope by adjusting a DC bias current. This form of adaptive biasing has been used to build high-dynamic-range silicon cochleas at audio frequencies [323].

3.4 Circuit Design

3.4.1 Cochlear Stages

A simplified view of our implementation of a single filter is shown on the left of Figure 3-12. Each filter, like in the bidirectional version, consists of two resonators that are coupled both inductively and capacitively. The transistor provides active gain and buffering. This topology is efficient because it uses a single transistor, which minimizes the power required to maintain thermal-noise-limited performance. The following impedance and frequency-normalized element values were found by our network synthesis routine (written in Mathematica) and used as nominal parameters

for this design: $L_1 = 0.76\text{H}$, $L_2 = 0.60\text{H}$, $M = 0.61\text{H}$, $C_1 = 0.81\text{F}$, $C_2 = 1.47\text{F}$, $C_C = 0.64\text{F}$, $1/g_m = 0.53\Omega$, $R_1 = 0.63\Omega$, $R_2 = 0.11\Omega$.

A more detailed circuit diagram of a single unidirectional cochlea filter is shown on the right of Figure 3-12. The first important change from the simplified circuit shown on the left of the figure is that the resistor R_1 has been replaced by an active element, the resistance seen looking into the source of the cascode transistor M_1, which is given by $1/g_{m1}$, where g_{m1} is its small-signal transconductance. To this resistance we must add R_1, the parasitic series resistance of L_1, to get the total load R_L at the drain of M_2. The high-pass filter formed by C_3 and R_3 is designed to decouple the DC operating points of individual stages but act as a short at RF. At "low" frequencies (lower than the center frequency of the stage, but higher than the cut-in frequency of C_3 and R_3), the voltage gain of the filter is given by

$$TF\left(s_n \ll 1\right) = \left(R_1 + 1/g_{m1}\right) \frac{g_{m2}}{g_{m2}R_2 + 1} \tag{3.24}$$

Here g_{m2} is the transconductance of M_2 and R_2 is the parasitic series resistance of L_2. In order to prevent low-frequency signals from either attenuating or blowing up as they propagate down the cascade of filters we need $TF\left(s_n \ll 1\right)$ to be as close to 1 as possible.

Assume for now that M_3 carries no bias current. In that case M_1 and M_2 share the same bias current. We designed them to have the same geometry, so $g_{m1} \approx g_{m2}$. Also R_1 and R_2 are parasitic components that are much smaller than $1/g_{m1}$ and $1/g_{m2}$, so we automatically get $TF\left(s_n \ll 1\right) \approx \left(g_{m1}/g_{m2}\right) \times \left(1 + g_{m1}R_1\right)\left(1 - g_{m2}R_2\right) \approx 1$. However, this relationship is only approximate. In particular, since the inductor L_1 is in series with a large resistance, i.e., $1/g_{m1}$, it does not need to have a high quality factor. Therefore we designed L_1 to have higher series resistance than L_2 in order to save layout area, i.e. $R_1 > R_2$. Therefore we should get $TF\left(s_n \ll 1\right) > 1$ (by a small amount).

In order to set the gain exactly equal to 1 we used an on-chip negative feedback loop, shown in Figure 3-13. An on-chip oscillator running at 22MHz was used to

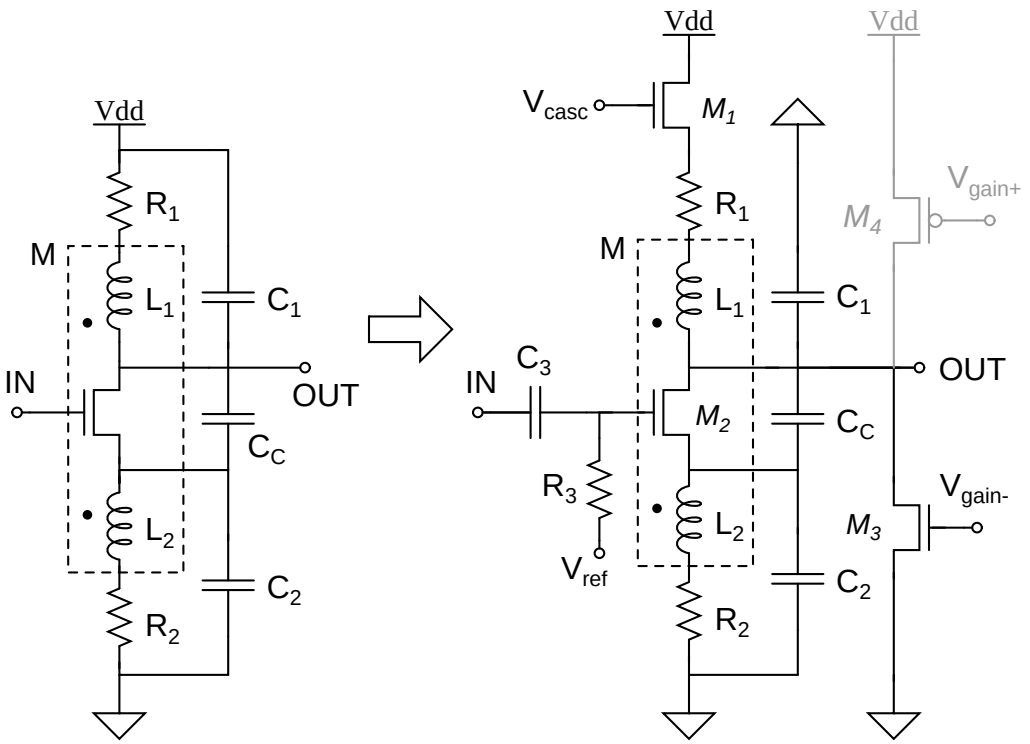


Figure 3-12: Circuit diagram of a single unidirectional cochlea filter: simplified version on the left, a more complete one on the right. The transistor M_4 was not actually implemented on the current chip (see the text for an explanation).

inject a small signal into the cascade. The amplitude of this signal at the end of the cascade is measured, and the feedback loop adjusting the voltage V_{gain-} until these amplitudes are equal to each other. In this way we ensure that the low frequency gain of the cochlea is exactly 1. The voltage V_{gain-} sets the current through the transistor M_3. By increasing V_{gain-}, we can make M_1 carry more current than M_2, thus making $g_{m1} > g_{m2}$ and lowering the gain of each cochlear stage.

The inductor L_1 was drawn on Metal 8, with the connection to the internal terminal on AL_RDL (which is an aluminum layer above Metal 8 meant for interfacing with bonding pads). Inductor L_2 was made up of coils on several metal layers that were shorted together to reduce series resistance: Metal 7 through Metal n (n increased from 7 to 4 as the inductor size increased). The connection to the internal terminal was made on metal $(n-1)$.

Experimentally, however, we found that the gain of the filters was < 1 even

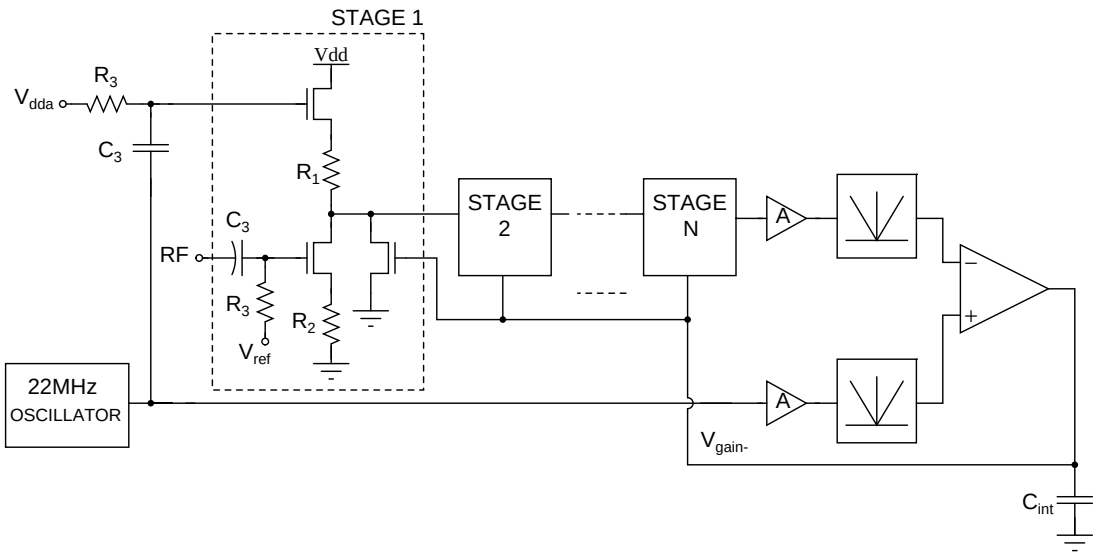


Figure 3-13: The feedback loop that adjusts the low-frequency gain of the cochlea by setting the bias voltage V_{gain-}.

with $V_{gain-} = 0$, and the loop as designed could only decrease the gain further by increasing V_{gain-} and the current through M_3. The most likely reason for the lowered gain is poorly-modeled parasitic resistances and inductances on the ground node, which increase the effective value of R_{s2}. A simple improvement, to be made in future iterations, is to modify the feedback loop so that it can both add and subtract current from M_2. A simple way to do this is by adding the PMOS transistor M_4 (see Figure 3-12). We can now increase the gain by lowering V_{gain+}, which increases the DC current through M_4, thus causing g_{m2} to increase without affecting g_{m1}.

3.4.2 Preamplifiers

Preamplifiers were also used in this design to amplify the output voltage before envelope detection. The amplifiers, shown in Figure 3-14, were two-stage resistively loaded common-source designs. In the first few filters the first amplifier stage was shunt-peaked by an inductor L_L to increase bandwidth, as shown in Figure 3-14. The two preamplifier stages were identical except for the (possible) presence of L_L.

In each amplifier stage the bias current through the transistor M2 (and its transconductance g_{m2}) was set using the current source M1. This current source was by-

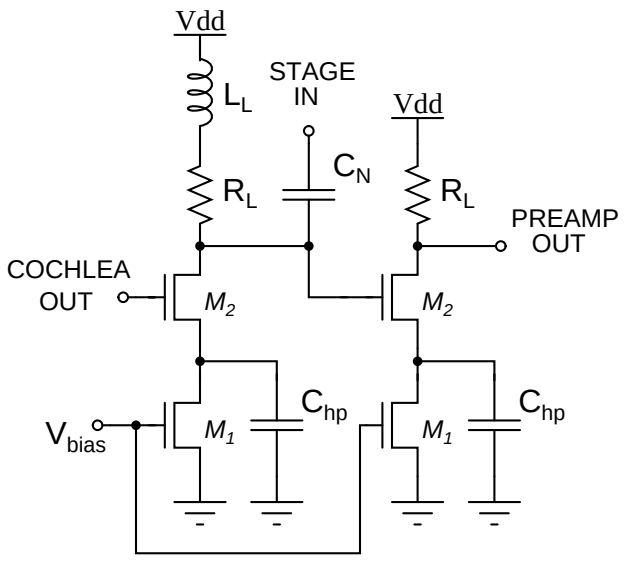


Figure 3-14: The two-stage preamplifier that was used inside every stage.

passed at RF by the capacitor C_{hp}, thus creating a first-order high-pass characteristic with a cut-in frequency of $f_{hp} = g_{m2}/(2\pi C_{hp})$. We used $C_{hp} \propto \exp(n/N_{nat})$, where $1 < n < N$ is the cochlear stage index, like the capacitances within the cochlea itself. Thus $f_{hp}(n)$, the cut-in frequency of the n-th stage, is proportional to $f_c(n)$, the center frequency of that stage. We could set V_{bias} through a DAC. It was usually set to a value that resulted in $f_{hp}(n) \approx f_c(n)/4$. The cochlea and preamplifier outputs have low-pass and asymmetric bandpass frequency responses, respectively. The preamplifier outputs roll off as second-order filters at low frequencies, which decreases masking effects from strong low-frequency inputs, i.e., signals at frequencies $f < f_{hp}(n)$. They roll-off as filters of order N_{nat} at high frequencies $f > f_c(n)$. The total voltage gain of the preamplifiers for frequencies $f > f_{hp}$ was $A_{preamp} = A_1 A_2 = A_1^2$, where $A_1 = A_2 = -g_{m2} R_L$ were the in-band gains of the first and second stages, respectively.

The capacitor C_N shown in Figure 3-14 was used to neutralize the gate-drain capacitance C_{gd2} of transistor M_2 within each cochlear stage. This capacitance is undesirable since it couples together the input and output nodes of each cochlear stage (IN and OUT in Figure 3-12) and distorts the transfer function. This is a particular problem in the high freqency stages, where the magnitude of C_{gd2} may be comparable to C_1, C_2 and C_C. By placing C_N, which has one end tied to IN and the other tied

to an amplified, inverted copy of OUT, C_{gd2} is reduced to a capacitance $C_{gd2} - A_1 C_N$ between IN and OUT. An additional capacitance $(1 + A_1) C_N$ between IN and ground is also produced. Here the voltage gain of the first preamplifier stage is given by $-A_1$, and we have assumed that the stage has no delay. This condition is satisfied if the operating frequency is much lower than the pole frequency $\approx 1/R_L C_L$, where C_L is the total load capacitance at the output of the first preamplifier stage. Thus, the capacitance between input and output nodes is completely cancelled if $A_1 C_N = C_{gd2}$. We can adjust A_1 by changing the bias voltage V_{bias} shown in Figure 3-14.

More generally, we can model the first preamplifier stage as a filter with a voltage transfer function $-A_1(s)$. In this case we can show that C_N modifies the original admittance between IN and OUT, given by sC_{gd2}, into two impedances: Y_1, between IN and ground, and Y_{12}, between IN and OUT, where

$$\begin{aligned} Y_1 &= sC_N \left(1 + A_1(s)\right) \\ Y_{12} &= sC_{gd2} - sC_N A_1(s) \end{aligned} \tag{3.25}$$

3.4.3 Envelope Detectors and Signal Scanner

Figure 3-15 shows the envelope detector that we use to detect the RF voltage amplitude present at the output of each cochlear stage. Each envelope detector, as in the bidirectional cochlea chip, consists of a rectifier and low-pass filter, and forms a simple incoherent receiver. In this implementation, we do not report the actual RF signal because it would take too much power to send these wideband signals off-chip. In future implementations, we plan to include more sophisticated receivers at every cochlear stage.

The rectifier shown in Figure 3-15 is a simple MOS diode-based circuit that produces pseudo-differential outputs. This strategy reduces DC offset by referring each rectifier output voltage to that of a well-matched local replica. The replica rectfier, shown on the left in Figure 3-15 has no input RF signal connected to it. More accu-

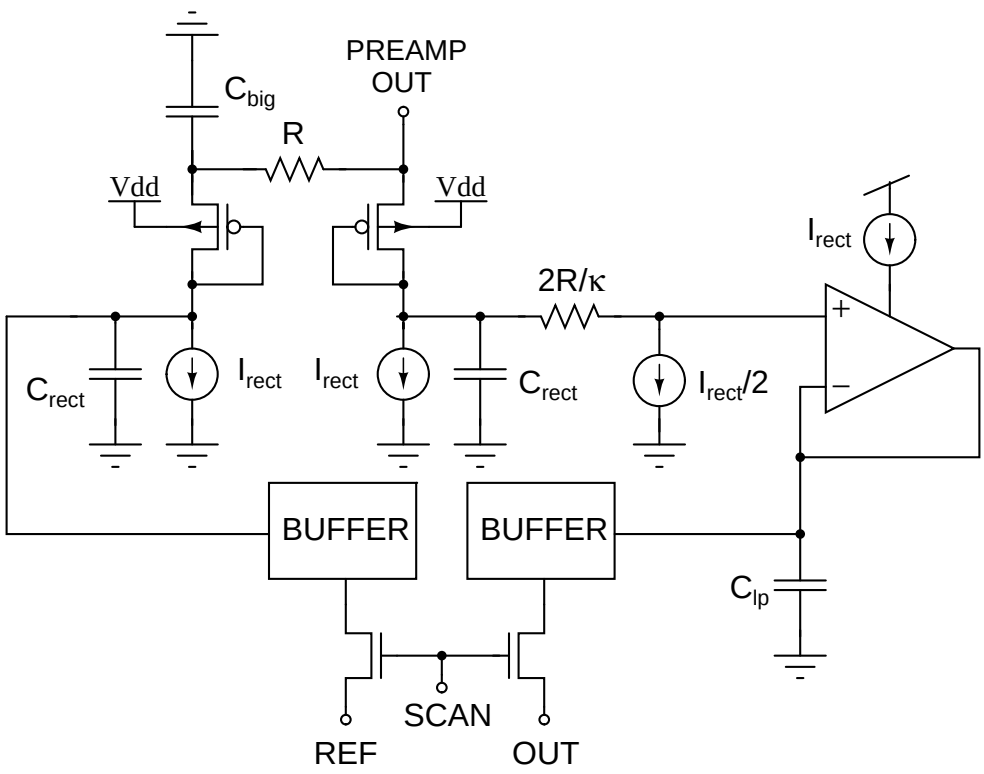


Figure 3-15: The pseudo-differential envelope detector that is used to report the RF signal amplitude present at every stage.

rately, the RF signal is low-pass filtered by R and C_{big} before it reaches the rectifier. The output voltage of the main rectifier, shown on the right, is normally higher than the replica by $I_{rect}R/\kappa$ because of the DC voltage drop on the resistor R. Here κ is the subthreshold exponential constant. In order to eliminate this systematic source of DC offset we make the diode in the main rectifier 1.5 times wider than that in the replica and bias it at 1.5 times the current, so that the two diodes continute to have the same source-gate voltage drop. However, one-third of the current in the main rectifier, i.e., $I_{rect}/2$, is used to bias an additional resistor of value $2R/\kappa$, thereby creating a voltage drop $I_{rect}R/\kappa$ which cancels out the offset.

The dead-zone of the rectifier is $\approx \phi_T/\kappa$, resulting in an input-referred dead-zone of $\approx (\phi_T/\kappa)/A_{preamp}$. The output of the main rectifier is filtered by a first-order $G_m - C$ low-pass filter. Both output and reference voltages are then buffered by source followers. Finally, the outputs of these buffers are time-multiplexed onto a common two-wire bus by an analog multiplexer, or scanner circuit. The $SCAN$

signal, generated by a token-passing circuit (essentially a ring counter) allows the buffers to drive the bus when it is high, and disconnects them from the bus when it is low. This multiplexing strategy was useful for the preliminary implementation described here because of the limited number of I/O pins available.

The current-controlled oscillator (CCO) used to generate the clock for the read-out scanner is shown in Figure 3-16. The control current I_{scan} linearly controls the oscillator frequency f_{scan}, as follows:

$$f_{scan} = \frac{I_{scan}}{2C_{scan}\left(V_{T+} - V_{T-}\right)} \tag{3.26}$$

where V_{T+} and V_{T-} are the positive and negative-going thresholds, respectively, of the Schmitt trigger. The bias current I_{rect} of the rectifier and low-pass filter inside each stage is automatically set to a constant fraction $1 : M^2$ of I_{scan} by a current mirror, as shown in Figure 3-16. Therefore $I_{rect} = I_{scan}/M^2$, where M is an integer. We assume that the overall output bandwidth of each stage is equal to BW_{lp}, the bandwidth of the low-pass filter shown in Figure 3-15. The output bandwidth is then given by

$$BW_{out} = BW_{lp} = \frac{g_{mp}}{2\pi C_{lp}} = \frac{I_{rect}}{2\pi C_{lp} V_L} \tag{3.27}$$

where g_{mp}, V_L and C_{lp} are the small-signal transconductance, linear range and load capacitance of the filter. In order for (3.27) to be true we must have $BW_{lp} < BW_{rect}$, where $BW_{rect} = \kappa I_{rect}/\left(2\pi C_{rect}\phi_T\right)$ is the output bandwidth of the rectifier.

The scanner samples each output voltage at a frequency $f_{frame} = f_{scan}/N$, where N is the total number of stages. In order to satisfy the Nyquist criterion and avoid aliasing, we must have $BW_{out} \leq f_{frame}/2$. In practice BW_{out} should be made lower than $f_{frame}/2$ since we do not have a brick-wall filter at the output (in fact, we only have a first-order one). Combining (3.26) and (3.27), we get

$$M^2 \geq \frac{N}{2\pi}\left[\frac{C_{scan}\left(V_{T+} - V_{T-}\right)}{C_{lp}V_L}\right] \tag{3.28}$$

The inequality in (3.28) is in a convenient form that only depends on constants

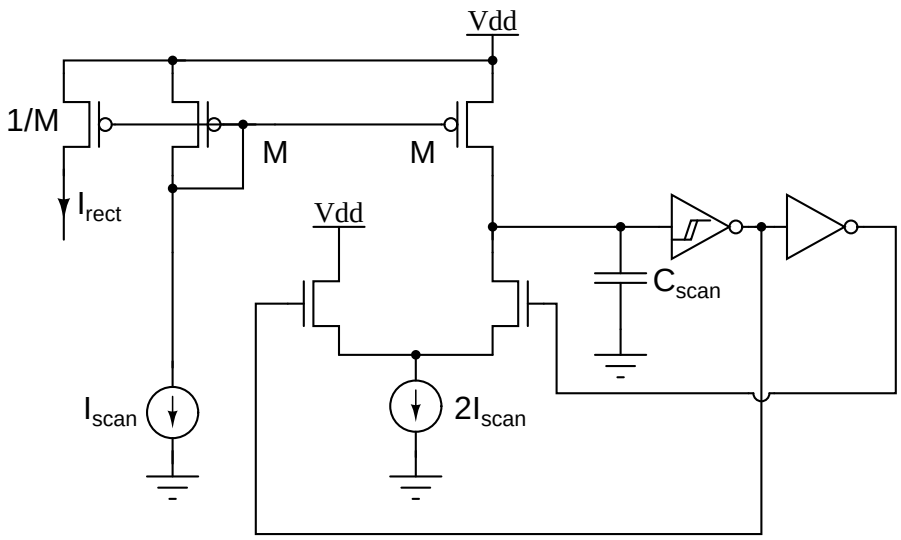


Figure 3-16: The current-controlled oscillator (CCO) that generates the clock for the read-out scanner.

like N and dimensionless ratios like C_{scan}/C_{lp} and $(V_{T+} - V_{T-})/V_L$. In particular, it is independent of f_{scan}. As a result, we can set M to a fixed value. On this chip, we used $M = 12$, which gives us $BW_{out} \approx f_{frame}/3$.

3.4.4 Automatic Gain Control

The local automatic gain control (AGC) circuit reduces the peaking associated with each stage when the signal amplitude increases. It can be disabled if necessary. The AGC works by inserting a resistor in series with the capacitor C_1 and increasing its value with the signal amplitude, thus lowering the quality factor of C_1. The variable resistor is formed by changing the gate voltage of an NMOS that carries no DC current, has $V_{DS} = 0$ and thus operates in the linear region. The circuit is shown in Figure 3-17. The NMOS gate voltage V_{AGC} is given by

$$V_{AGC} = V_{DD} - I_{AGC} R_{AGC} (OUT - REF) \tag{3.29}$$

where the pseudo-differential output voltage produced by the rectifier is given by $v_{out} = (OUT - REF)$. If the NMOS acts as a square-law device, its drain-source resistance is given by

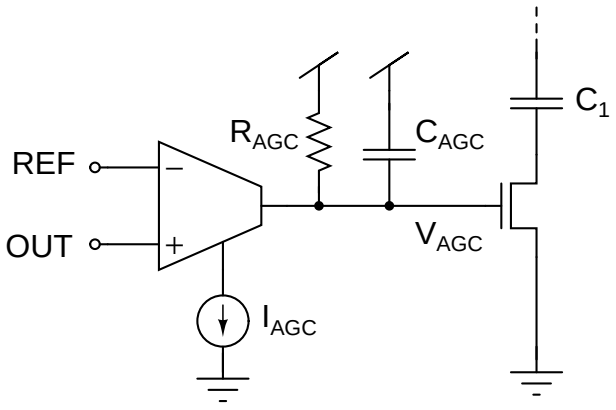


Figure 3-17: Automatic gain control (AGC) circuit used inside every stage.

$$R_{C1} \approx \frac{1}{\mu_n C_{ox} \frac{W}{L} (V_{AGC} - V_T)} = \frac{1}{\mu_n C_{ox} \frac{W}{L} (V_{DD} - V_T - I_{AGC} R_{AGC} (OUT - REF))} \tag{3.30}$$

Therefore the quality factor of C_1 is given by

$$Q_{C1} = \frac{1}{Y_{C1} R_{C1}} = \frac{\mu_n C_{ox} \frac{W}{L}}{Y_{C1}} (V_{DD} - V_T - I_{AGC} R_{AGC} (OUT - REF)) \tag{3.31}$$

where $Y_{C1} = \omega C_1$ is the admittance of C_1, and is approximately constant at the center frequency for every stage because of the scale-invariant nature of the cochlea. We see that Q_{C1} may be written in the form $a(b - v_{out})$, where a and b are constants. In other words Q_{C1} decreases in a linear way with output signal amplitude. It can be shown that the peak gain of the stage is approximately equal to

$$G_{stage} = \frac{G_{stage}(0)}{1 + \sqrt{G_{stage}(0)}/Q_{C1}} \tag{3.32}$$

where $G_{stage}(0)$ is the small-signal value of G_{stage}. Finally, it should be noted that the explicit AGC described in this section operates in parallel with a "natural" AGC that arises from the limited linear range $V_L = I_{bias}/g_m$ of the transistors M1 and M2 within the cochlear stage (see Figure 3-12). Here I_{bias} and g_m are the DC bias current of each transistor and its transconductance, respectively. The effective transconductances g_{m1}

and g_{m2} of M1 and M2 decrease for RF signal amplitudes greater than V_L, reducing G_{stage} to lower values than predicted by (3.32). The value of V_L is equal to ϕ_T/κ in subthreshold, and ideally increases $\propto \sqrt{I_{bias}}$ above threshold.

3.4.5 Calibration Loop

The bias voltage V_{ref} sets the value of g_{m2} and g_{m1} within each stage, as shown in Figure 3-12. The peak gain, or quality factor of the stage is approximately equal to $G_{stage} = g_{m2}g_{m1}\,(L_1/C_1)$. In addition, the DC gain of the stage is approximately $g_{m2}/g_{m1} = 1$, so we get

$$G_{stage} \approx g_{m2}^2 \left(\frac{L_1}{C_1}\right) \tag{3.33}$$

Equation (3.33) shows that we can control the sharpness of the cochlear transfer functions by changing g_{m2}. The values of g_{m1} and g_{m2} are controlled by the bias voltage V_{ref}. This voltage is set by the feedback loop shown in Figure 3-18. Consider the amplifier formed by M_2, which has the same geometry as the transistor M_2 within the filter stages, and R_L. The loop measures input and output amplitudes and adjusts V_{ref} until this amplifier has a gain of 1 at the oscillator frequency (22MHz). This gain is approximately $g_{m2,eff}R_L$, where $g_{m2,eff} = g_{m2}/\,(g_{m2}R_2 + 1)$ and R_2 is designed to be equal to its average value in the cochlear stages. Therefore the loop sets $g_{m2,eff} = 1/R_L$. By adjusting R_L, we can therefore control the cochlear transfer functions (and also the power consumption). The switch and monostable (one-shot) ensures that the loop does not get stuck at the wrong operating point by resetting V_{ref} if it exceeds a reference value V_{max}. This precaution is necessary because the relationship between V_{ref} and gain is not monotonic: for very high values of V_{ref}, the gain drops because the transistor M_2 comes out of saturation.

Figure 3-19 shows the amplifier used inside the gain calibration loop shown in Figure 3-18. The circuit is an AC-coupled cascoded common-source amplifier, with the bias voltage $V_{b,amp}$ setting the bias current and voltage gain. Figure 3-20 shows the differential delay cell used in the 22MHz ring oscillator also shown in Figure 3-18.

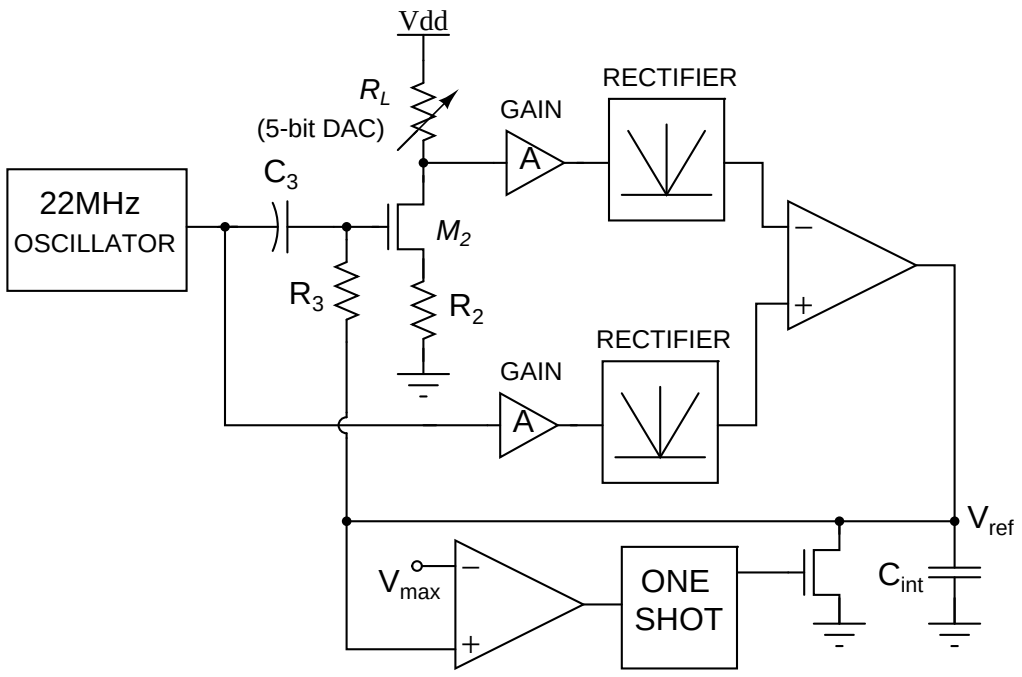


Figure 3-18: Feedback loop used to set the bias voltage V_{ref}.

Source degeneration is used to increase the linear range of the differential pair and reduce harmonic distortion in the output waveform. The stage produces two sets of differntial outputs in response to an input V_{n-1} from the previous stage: V_n, which is fed to the next stage, and $OUT = V_n/10$, which is fed into the calibration circuit. In this way we ensure that the amplitude of OUT is smaller than the linear range of the amplifiers in the calibration circuit; as a result, we ensure that the loop sets the *small-signal* transconductance $g_{m2,eff}$ within each cochlear stage.

3.4.6 Low-Noise Amplifier

Figure 3-21 shows the common-gate low-noise amplifier (LNA) used to interface the unidirectional RF cochlea chip with the outside world. Common gate topologies are efficient because they use the same transistor for both impedance matching and voltage gain. The input impedance of the LNA as seen by the signal source is approximately given by

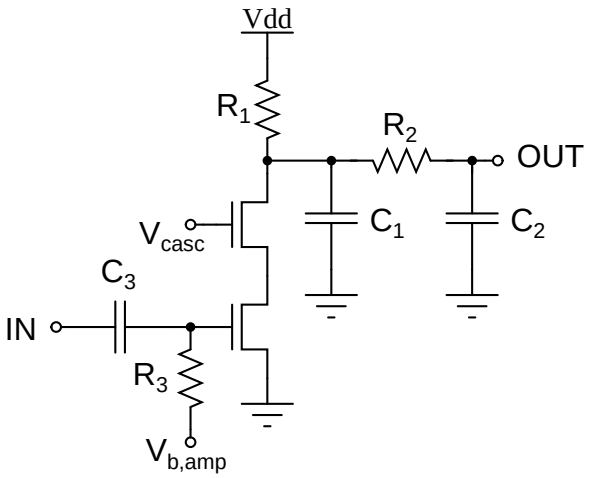


Figure 3-19: Amplifier used inside the gain calibration loop.

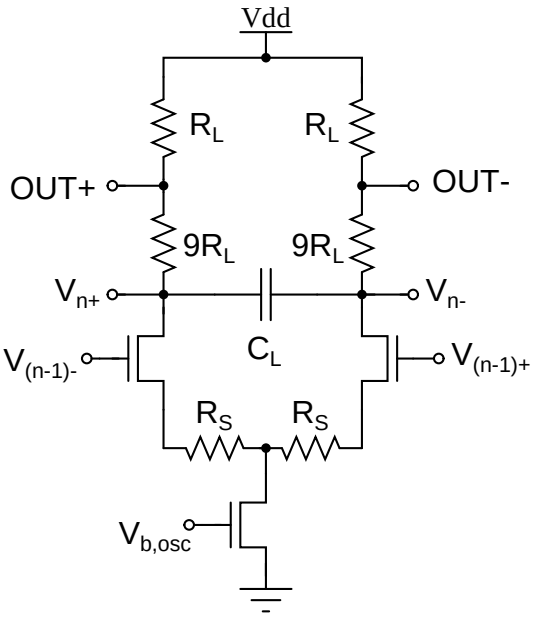


Figure 3-20: The differential delay stage that was used within the 22MHz calibration oscillator.

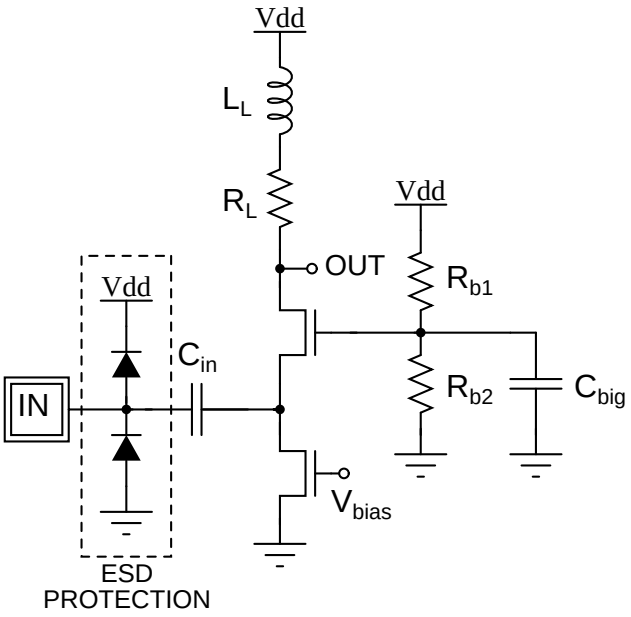


Figure 3-21: Common gate low-noise amplifier (LNA) used in the unidirectional RF cochlea.

$$Z_{in} = sL_{par} + \frac{1}{sC_{par}} \parallel \left(\frac{1}{sC_{in}} + \frac{1}{g_s} \right) \tag{3.34}$$

where L_{par} is the parasitic series inductance (mostly due to the bond wire), C_{par} is the shunt capacitance to ground at the input node IN (due to the package, bonding pad, bottom-plate parasitics of C_{in} and ESD diodes) and $1/g_s$ is the impedance seen looking into the source of the input transistor. At frequencies much larger than $\omega_{min} = g_s/C_{in}$, the impedance of C_{in} may be neglected compared to g_s. Similarly, for frequencies much smaller than $\omega_{max} = 1/\sqrt{L_{par}C_{par}}$, the impedance of L_{par} is much smaller than that of C_{par}. Between ω_{min} and ω_{max}, therefore, the magnitude of the transfer function from the signal source to the source of the input transistor is approximately 1, while the input impedance is given by $Z_{in} \approx 1/\left(g_s + sC_{par}\right)$.

In order to impedance match the real component of Z_{in} to R_A, the output resistance of the signal source, we must have $g_s R_A = 1$. Most RF test equipment uses $R_A = 50\Omega$. In order to minimize reflections we must make the parasitic capacitance C_{par} as small as possible, but its admittance increases with frequency and ultimately limits the impedance-matching bandwidth.

If L_L was absent the bandwidth of the LNA would be given by $1/(2\pi R_L C_L)$, where C_L is the load capacitance present at the output node. The inductor L_L increases bandwidth without affecting gain or power consumption by adding a zero to the low-pass transfer function of the LNA. This technique, known as shunt peaking, has been used since the days of vacuum tubes [158]. In the passband the effects of both L_L and C_L may be neglected and the voltage gain of the LNA is given by

$$G \equiv \frac{v_{out}}{v_{in}} = \frac{g_s R_L}{2} = \frac{R_L}{2R_A} \tag{3.35}$$

where the factor of 2 comes from the resistive voltage division caused by impedance matching at the input node. Ignoring induced gate noise, it can be shown that the noise figure of this LNA in the passband is given by

$$F = 1 + \frac{4}{g_s R_A}\left(\frac{\kappa\gamma}{(g_s R_A + 1)^2} + \frac{1}{g_s R_L}\right) = 1 + \kappa\gamma + \frac{2}{G} \tag{3.36}$$

where γ is the excess noise figure of the input transistor[1], κ is its subthreshold slope factor and we have assumed that the current source at the bottom has been sized such that its transconductance and noise are negligible.

3.4.7 Other Circuits

Other circuits present on the chip include:

- Constant-g_m current reference (1μA nominal output)
- 5-bit current DACs for programming the following chip parameters
  - Calibration oscillator amplitude
  - Calibration amplifier voltage gain
  - Preamplifier and LNA voltage gains
  - Scan frequency f_{scan}, through I_{scan}

[1]The drain current noise PSD of a MOSFET is given by $4kT\gamma g_m$, where $g_m = \kappa g_s$ is its small signal transconductance.

  - AGC gain, through I_{AGC}
  - Stage impedance (resistor DAC)
  - Output buffer bias current
- Global power-on reset
- Output voltage buffers (source followers)
- Local RC lowpass filters on global bias voltage lines to prevent unwanted RF coupling between stages
- About 1nF of on-chip supply bypass capacitance
- Multiple (30+) V_{DD} and ground pins to reduce supply-line inductance

3.5 Measurements

3.5.1 Test Setup

Our unidirectional RF cochlea chip contained $N = 51$ stages with $N_{nat} = 16$. A die photograph of it is shown in Figure 3-22. Both bidirectional and unidirectional cochlea chips was mounted on custom printed circuit boards for testing. An example is shown in Figure 3-23. The chip was tested by hooking it up to two power supplies (V_{DD} and V_{DDA}), setting a few configuration bits using jumpers on the test board and then varying on-chip parameters via the programming interface. The programming interface consists of a 8-bit wide bus divided into two sections: 3-bit address and 5-bit value. The 3 address bits allow 7 chip parameters to be programmed (the eighth setting is not used). A separate "program enable" (PR_EN) pin is raised high once valid data has been placed onto the bus. The on-chip latches that store parameter values are only transparent when PR_EN is high. The interface is controlled from a computer through a National Instruments data acquisition (DAQ) card and a graphical user interface (GUI) written in LabView (National Instruments). A screen-shot of the GUI is shown in Figure 3-24.

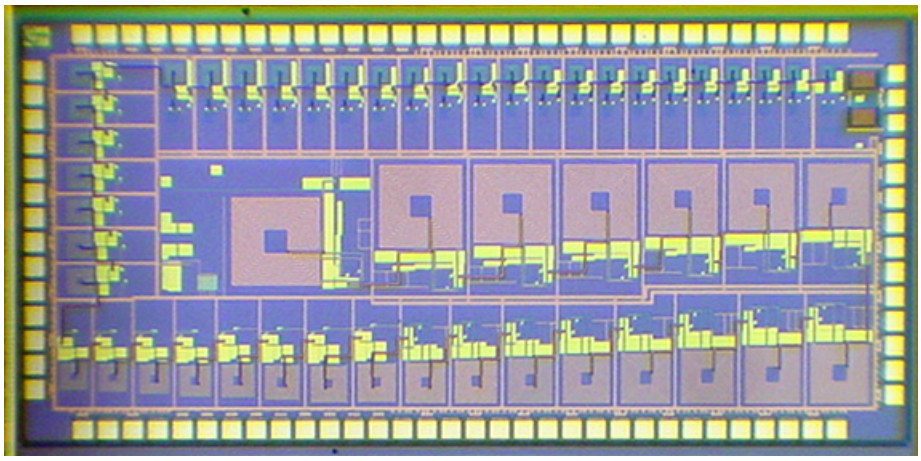


Figure 3-22: Die photograph of the unidirectional RF cochlea chip.

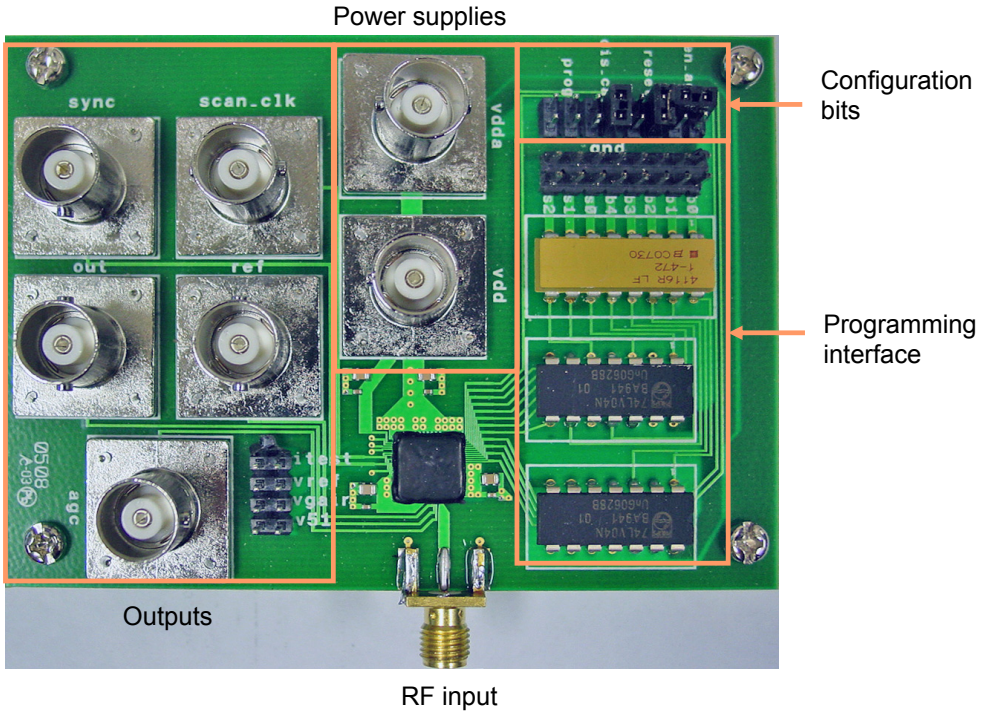


Figure 3-23: Printed circuit board used for testing the unidirectional RF cochlea.

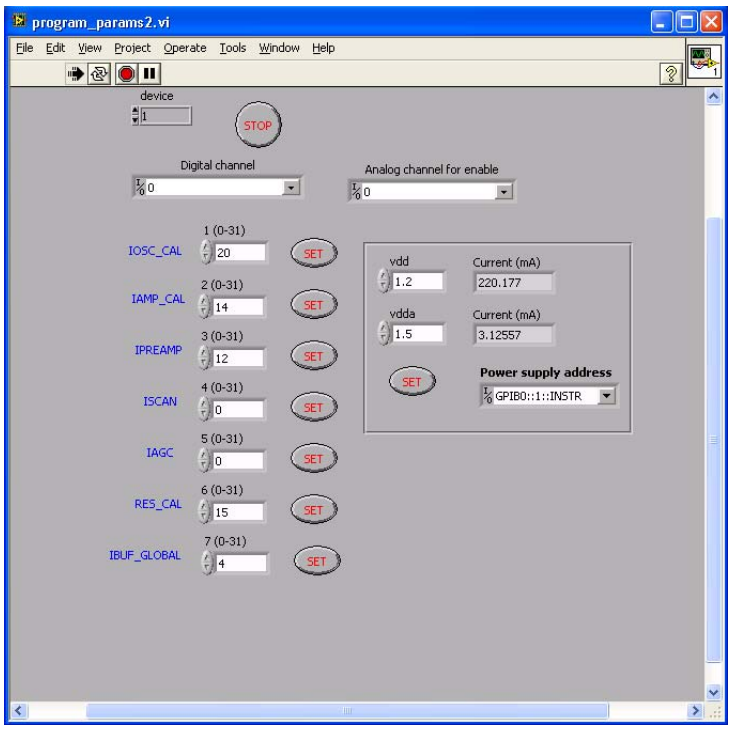


Figure 3-24: The Labview GUI used for programming the unidirectional RF cochlea.

A perennial problem with setting on-chip bits with commercial DAQ cards is that most of them have TTL-compatible (5V) digital input/output ports. Such high voltages should not be fed directly into our chips, which may only be running at 1V (even with ESD-protected inputs). An alternative is to buy DAQ cards with adjustable-voltage digital ports, but these are hard to find and quite expensive. We used an alternative strategy: the TTL-compatible outputs from our DAQ card were fed through current-limiting series resistors (we used 47kΩ, but the exact value is not critical) and then into low-voltage CMOS inverters (74LV04 chips). The power supply on the inverters was tied to V_{DDA}. The 74LV04 comes with built-in ESD protection diodes which turn on when the DAQ card puts out 5V, with the series resistors limiting the current and preventing the diodes from burning out. Thus this hack converts 0 to 5V digital signals into (inverted) 0 to V_{DDA} signals which are suitable for programming the cochlea chip.

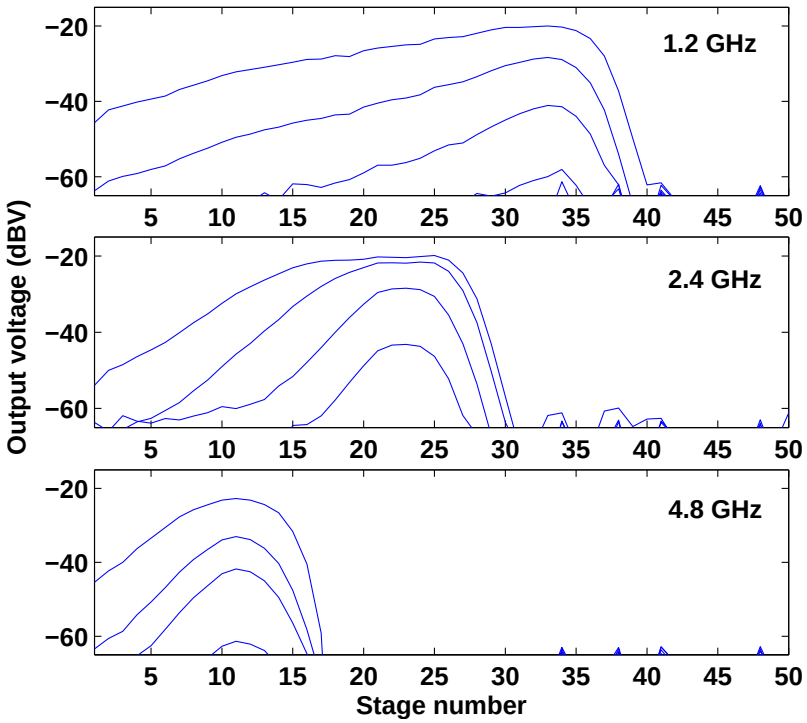


Figure 3-25: Spatial responses of the unidirectional cochlea to different input frequencies at the following power levels: -30, -40, -50, -60dBm.

3.5.2 Experimental Results

Figure 3-25 shows spatial responses of the unidirectional cochlea to three input frequencies and four power levels. The frequency-to-space transform is clearly visible, as is gain compression at high input power levels. Figure 3-26 shows that we can control the peak gain of the cochlear transfer functions by changing the value of R_L, the load resistor shown in Figure 3-18. The figure shows spatial responses at 2 and 4GHz for different values of $R_L = 1\text{k}\Omega/D$, where D is the digital code of the 5-bit DAC that sets R_L. In this case D was increased from 11 to 16, decreasing R_L from 91Ω to 67Ω. The result is increased voltage gain. However, power consumption also goes up, because the transconductance of the transistors M_1 and M_2 inside each filter must increase to keep the gain $g_{m2}R_L$ fixed at 1. In this case power consumption increased from 200mW to 300mW.

Figure 3-27 shows spatial responses of the unidirectional cochlea at 2GHz as the gain of the preamplifiers is varied by changing the DAC code that controls the bias

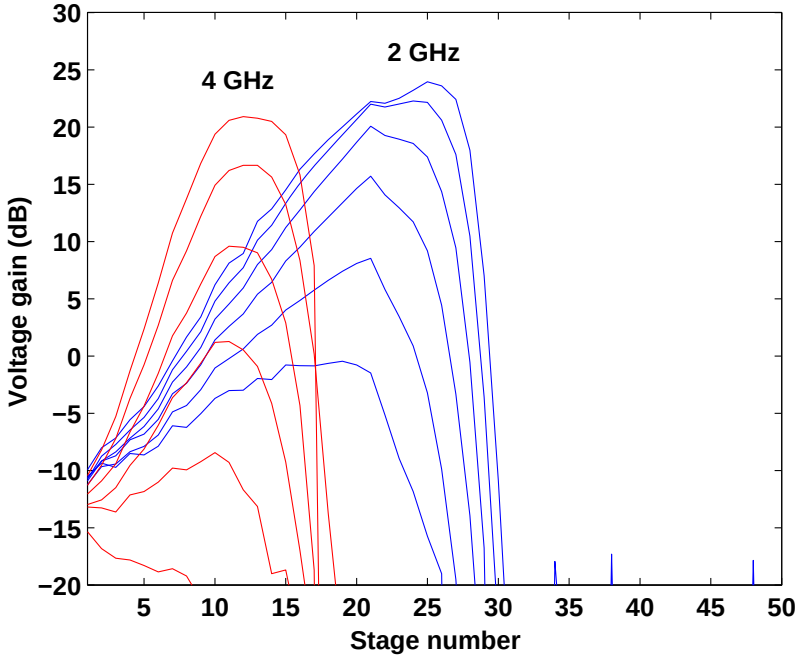


Figure 3-26: Spatial responses of the unidirectional cochlea to two different input frequencies for values of the gain-control resistor R_L varying from 91Ω (DAC code = 11) to 67Ω (DAC code = 16).

current I_{preamp}. The DAC code linearly controls I_{preamp}. The bias current of the LNA, I_{LNA}, is a scaled copy of I_{preamp}. Figure 3-28 shows the measured power consumption of the cochlea for various values of I_{preamp} and the calibration resistor R_L. Note that the DAC code linearly controls I_{preamp} since here we use a current DAC. However R_L is set using a resistive DAC: $R_L = 1k\Omega/D$, where D is the DAC code. Since the calibration loop guarantees that $g_{m1}R_L = 1$, we get $g_{m1} = D/1k\Omega$, i.e., the stage transconductance is a linear function of the DAC code. However, transconductance is a nonlinear function (in fact, a highly compressive one) of the bias current. This explains why the current consumption is a highly *expansive* function of the DAC code that sets R_L.

Figure 3-29 shows measured frame frequency f_{frame} as a function of the DAC code that sets I_{scan}. The clock frequency is $f_{scan} = Nf_{frame}$, where $N = 51$ is the number of stages. We note that f_{frame} is quite linear in the DAC code, indicating that the CCO control characteristic is linear in I_{scan}. The one exception appears

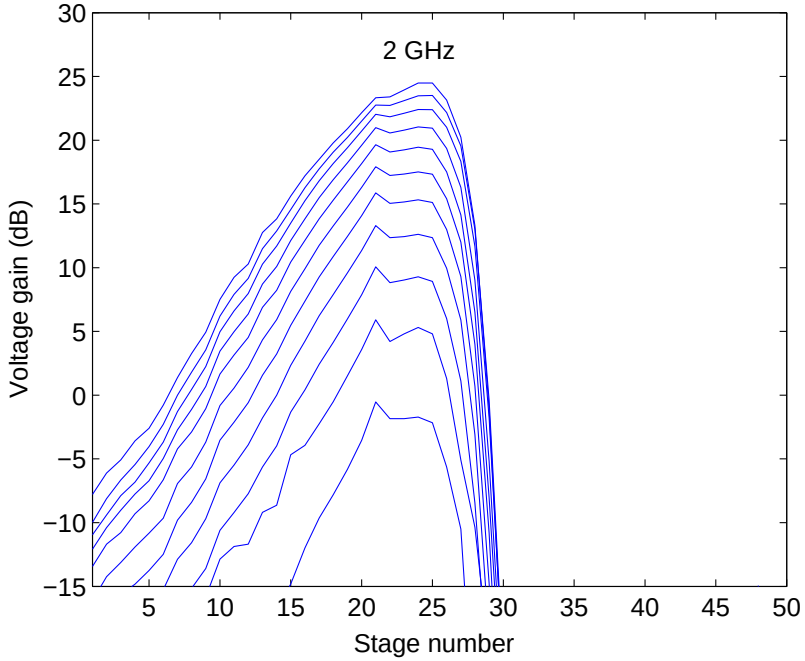


Figure 3-27: Spatial responses of the unidirectional cochlea to a 2GHz input tone at -40dBm for various values of preamplifier gain. The DAC code controlling the bias current in the preamplifiers was increased from 4 to 14 in this figure.

in the small "kink" in the curve between DAC codes 15 and 16. This "kink" was probably introduced by the DAC itself. The DAC consists of binary-weighted current sources that are added up to produce the output current. Between codes 15 ('01111') and 16 ('10000') *all* these current sources switch state from on to off or vice versa. Mismatches between them result in a differential nonlinearity (DNL) in the transfer curve at this point. This behavior is an example of a well-known phenomenon in both ADCs and DACs, namely the strong correlation of large DNL in the transfer curve with major code transitions (those that cause large numbers of bit flips).

Figure 3-30 shows a histogram of the measured jitter in the frame period $T_{frame} = 1/f_{frame}$ when I_{scan} was set to the lowest possible value (DAC code = 0). The standard deviation in T_{frame} was 290.8ns, with the mean value being 86.5μs. If jitter on the underlying clock edges is uncorrelated from edge to edge, it will accumulate as the square root of the number of edges between measurements. Assuming that the measured jitter comes entirely from the clock, the rms clock jitter is then given by

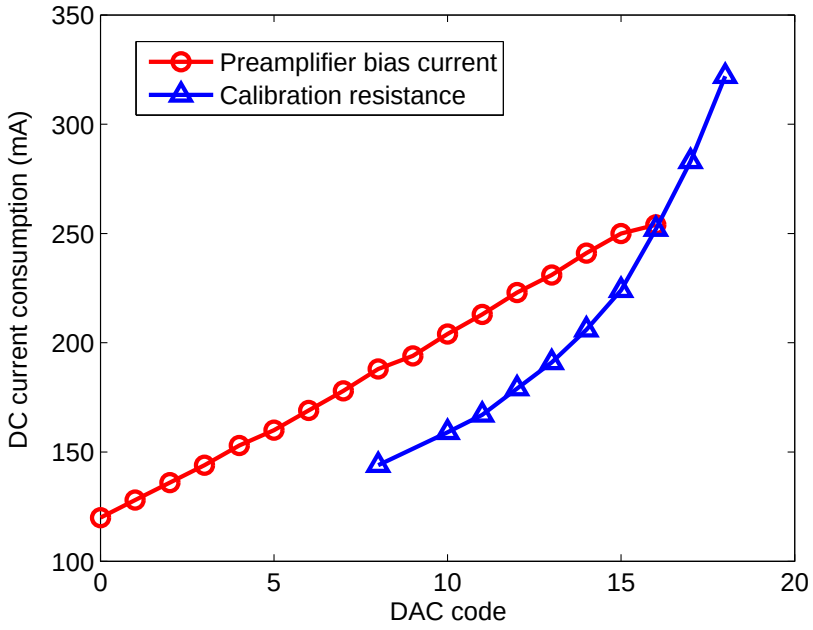


Figure 3-28: Total DC current consumed by the unidirectional cochlea as a function of (a) the preamplifier bias current (with the calibration resistor DAC code fixed at 15), and (b) the calibration resistor (with the preamplifier bias DAC code fixed at 12). The power supply voltage was $V_{DD} = 1.2$V.

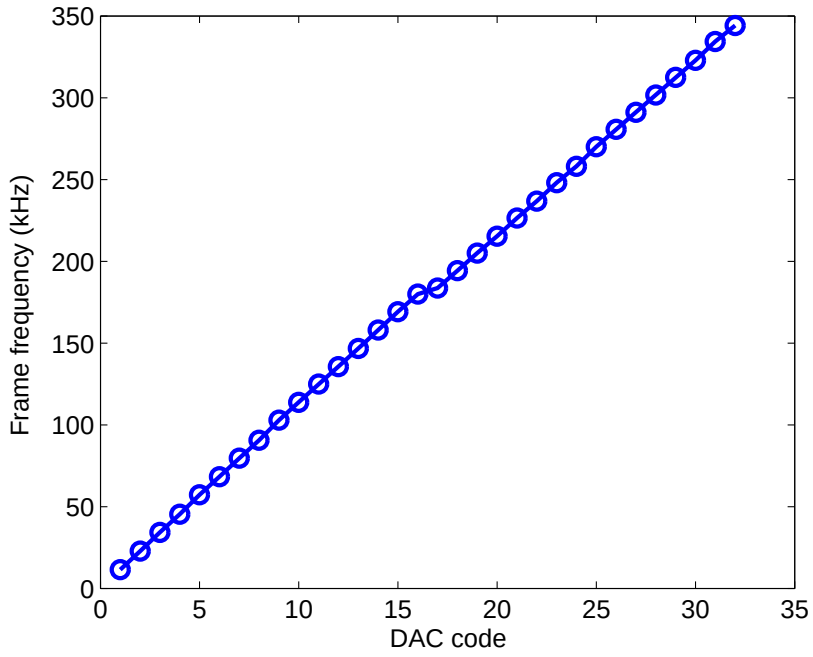


Figure 3-29: The measured frame frequency $f_{frame} = f_{scan}/N$ as a function of the DAC code that sets the reference current I_{scan} used in the clock generator.

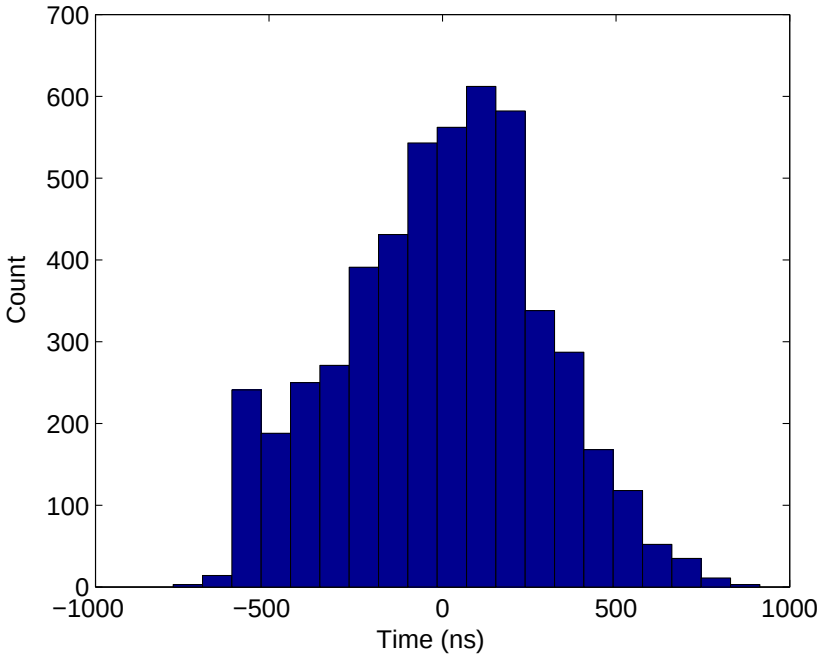


Figure 3-30: Histogram of jitter of the measured frame period $1/f_{frame}$ about its mean value (86.5μs, corresponding to f_{scan} = 11.56kHz). The DAC code controlling I_{scan} was set to 0.

290.8ns/$\sqrt{51}$=40.7ns, the mean clock period being 86.5μs/51=1.70μs. This seems to be an excessively large amount of jitter, perhaps indicating that the measured jitter is not entirely produced by the clock.

Figure 3-31 shows the measured reflection coefficient at the input terminal of the cochlea at a power level of -20dBm. The source impedance was 50Ω. The input impedance can be varied by changing I_{LNA}, the bias current of the LNA. As shown in the figure, the best match was obtained for bias currents obtained by setting the DAC in question to values between 13 and 18. The match degrades at low frequencies because of the high-pass filter formed by C_{in} and $1/g_s$ (see Figure 3-21). It degrades at high frequencies because of the low-pass filter formed by package parasitics: L_{par}, the bond-wire inductance and C_{par}, the shunt capacitance at the input terminal.

Figure 3-32 again shows measured reflection coefficient at the input terminal, but now as a function of the input power level, for a few frequencies. We note that the reflection coefficient begins to increase significantly for power levels greater than -

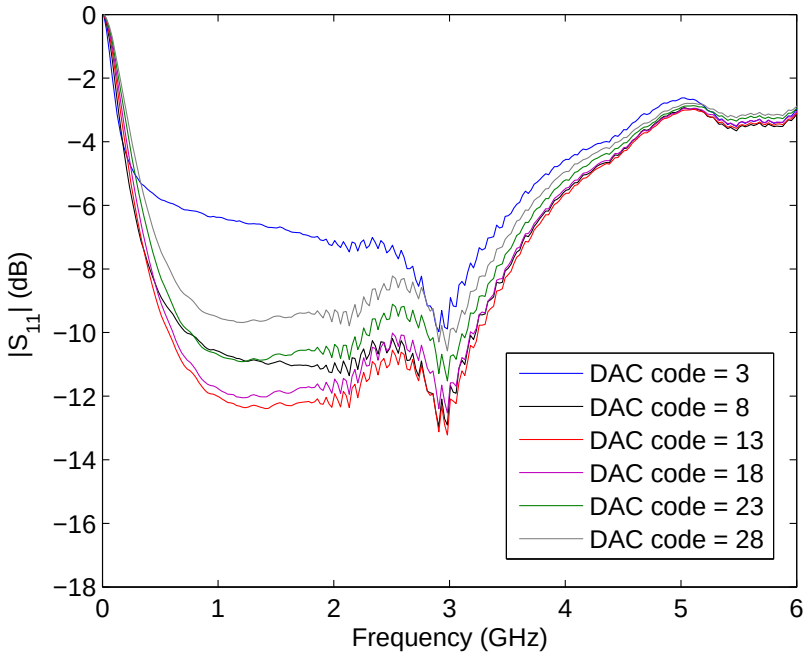


Figure 3-31: Measured input reflection coefficient of the RF cochlea chip for various values of LNA bias current. The input power level was fixed at -20dBm.

10dBm. This behavior occurs because the effective transconductance $g_{m,eff}$ of the input transistor in the LNA decreases for large-signal inputs, increasing the input impedance, which is approximately $1/g_{m,eff}$. From a practical standpoint, this data allows us to set the largest acceptable input signal level at approximately -10dBm.

3.5.3 Performance Summary

Spatial responses in the unidirectional cochlea were broadly similar to those obtained from the bidirectional cochlea, but secondary peaks due to inter-stage reflections were absent because of the unidirectional nature of the cascade. This property allows a lower value of N_{nat} to be used, which reduces noise, power consumption and chip area at the cost of reduced frequency resolution.

The performance of both cochleas is summarized in Table 3.1. In the table, peak voltage gain refers to the gain experienced by small signals (no gain compression). In addition, the quoted dynamic range (DR) is for single input tones, with the maximum signal being set by gain compression, and the minimum signal by the input-referred

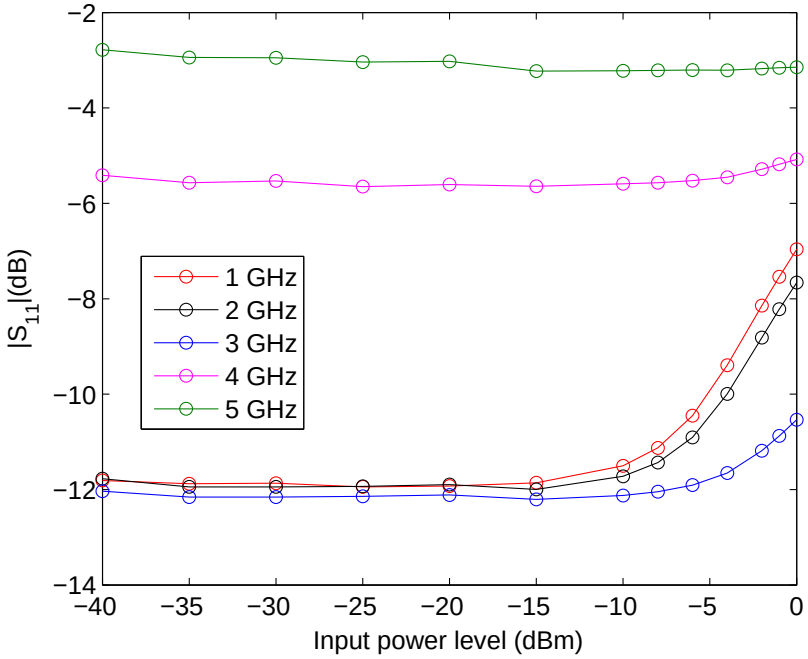


Figure 3-32: Measured input reflection coefficient of the RF cochlea chip as a function of the input power level at various frequencies. The DAC code that sets the LNA bias current was fixed at 12.

noise floor. The presence of other tones will reduce dynamic range, as explained in the previous chapter. The two cochleas have similar dynamic ranges. The unidirectional cochlea has significantly higher voltage gain and output noise, but somewhat lower input-referred noise.

However, the actual DR values of both cochlea chips is lower than the 70dB shown in Table 3.1. This is because the lower end of the DR is not set by the input-referred noise floor, but by the dead zone of the envelope detectors (EDs). The ED dead zone limits the minimum detectable signal at the cochlear outputs to $(\phi_T/\kappa)/A_{preamp}$, where A_{preamp} is the voltage gain of the preamplifiers. The minimum acceptable bandwidth of these preamplifiers is equal to the center frequency of the cochlear stage in question. This bandwidth requirement limits the maximum value of A_{preamp} to approximately 6.3 (16dB) via the usual amplifier gain-bandwidth trade-off[2]. As a

[2]We used two-stage preamplifiers to distribute the gain. The result is a more gentle trade-off between gain and bandwidth. The propagation delay also increases, but this is not an issue since the EDs discard phase information anyway.

Table 3.1: RF cochlea: Performance summary

Parameter	**Bidirectional**	**Unidirectional**
Technology	0.13μm CMOS	0.13μm CMOS
Stages per e-fold, N_{nat}	24	16
Total number of stages, N	50	51
Frequency range, GHz	1.2-8	0.6-6
Maximum output noise, dBV_{rms}	-70	-55
Peak voltage gain, dB	10	35
Maximum output SNR, dB	60	35
Input dynamic range, dB	70	70
Input impedance, Ω	50	50
Typical power consumption	170mA, 1.8V	220mA, 1.3V

result, we get a miniumum detectable output amplitude of $(\phi_T/\kappa)/A_{preamp} \approx 5.5$mV (-45dBV). The minimum detectable *input* amplitudes of the bidirectional and unidirectional cochleas are then equal to -55dBV and -80dBV, resulting in DR values of 45dB and 60dB, respectively.

ED dead-zone impacts the DR of the bidirectional cochlea to a greater extent since its transfer functions have lower peak gain. We can improve DR in several ways. For example, we can build more sophisticated preamplifiers that allow A_{preamp} to be increased without lowering bandwidth. However, in this case we must ensure that the preamplifier outputs have a linear range $> V_L A_{preamp}$, where V_L is the linear range of the cochlea itself. Gain control will have to be added to the preamplifiers if this condition is not met. Alternatively, we can replace the EDs with more sensitive coherent detectors.

Figure 3-33 summarizes the frequency-to-place transform measured for both designs. Deviations from exponential scaling are visible at the low frequency end of the bidirectional design and are caused by the fact that our simple line-termination network cannot perfectly approximate the line impedance at all frequencies. A higher order termination network can be used to reduce this effect. Deviations from exponential scaling in the unidirectional design were mainly caused by gain compression, which makes it difficult to determine the location of the peak response.

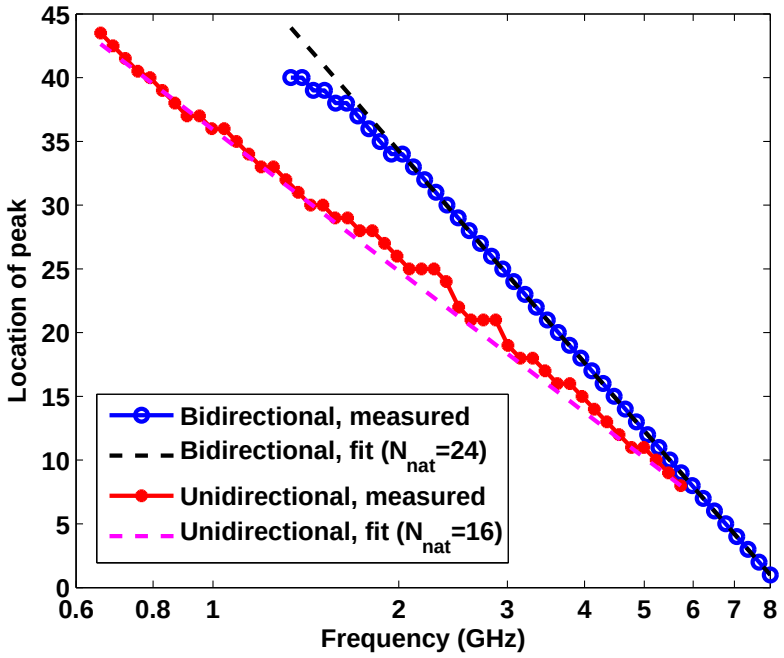


Figure 3-33: Measured frequency-to-space transform for the unidirectional and bidirectional cochleas, showing the location of the peak response as a function of input frequency.

3.6 Comparing Cochlear Designs

In this section, we summarize and compare important characteristics of the bidirectional and unidirectional RF cochlea designs.

Realism: The bidirectional RF cochlea more closely models the biological cochlea, which allows wave propagation in both directions. However, only the forward going wave is likely to be significant during normal cochlear operation[3]. As a result, our unidirectional cochlea design also captures important aspects of cochlear operation. However, it was derived by ignoring the pre-exponential part of the WKB solution. This additional approximation tends to result in lower frequency selectivity than the bidirectional version.

Discretization: The results of spatial discretization are different in the bidirectional and unidirectional cochleas. For example, the performance of the bidirectional cochlea asymptotes as N_{nat}, the number of filters per e-fold increases, but that of the

[3]Wave propagation in the reverse direction (from the apex to the base) is necessary for the generation of oto-acoustic emissions (OAE's).

unidirectional cochlea does not. Fundamentally, cascades of unidirectional filters are not formally valid ways of approximating partial differential equations. The voltage gain and group delay τ_g of the cascade become infinite as $N_{nat} \to \infty$ because a single stage sees an infinite number of poles from all previous filters in the system.

The roll-off slopes of the cochlear transfer functions increase with N_{nat} in both cochleas, resulting in improved frequency resolution. However, the sensitivity, or minimum detectable signal, remains roughly constant, and larger value of N_{nat} also lead to increased power consumption and layout area. Overall performance is therefore maximized at an optimum value of N_{nat}. However, this optimum will be different for the two cochleas.

Stability: The unidirectional cochlea is unconditionally stable as long as each filter stage is stable. This is because, ideally, the stages don't interact with each other. On the other hand, predicting the stability of the bidirectional cochlea is non-trivial. It is difficult to impedance-match the low-frequency end (apex) of the cochlea over all frequencies. Thus, reflections occur if low frequency components that have best places close to the apex are present along the transmission line. There may also be inter-stage reflections, which were analyzed in the previous chapter. Reflected waves with large amplitudes cause secondary peaks in the cochlear transfer functions, and, in extreme cases, instability.

A formal stability analysis of the bidirectional cochlea was presented in [323]. The quality of the impedance match provided by the apical termination was found to be critical for stability, particularly if the cochlear transfer functions have high voltage gain. In order to improve the impedance match we can use higher-order matching networks, or make the last few sections of the bidirectional cochlea heavily damped (dissipative) in order to attenuate the wave both before and after it reflects from the apical termination[4]. Since the cochlea is nonlinear, its stability depends on amplitude level, i.e., it is conditionally stable. However, the nonlinearity is compressive, and peak gain decreases with amplitude, which improves stability. Thus stability for

[4]The latter strategy is analogous to the use of perfectly matched layers (PMLs) for attenuating reflected electromagnetic waves. Such layers are used both for numerical simulations and within anechoic chambers.

small signals (the worst case) guarantees stability at all amplitude levels.

Noise: A serious problem with any transmission line or cascade of filters is accumulation of noise contributed by other stages. The bandwidth of the cochlea decreases exponentially with position. As a result, low-frequency (apical) stages contribute less noise than high-frequency (basal) stages. In the unidirctional cochlea discussed in [259], the result was total output noise that increased with position before saturating to a fixed value. The amount of noise was extremely sensitive to changes in the filter transfer function. For example, the value of Q in (3.1) strongly affected the peak gain and asymptotic noise level. Our unidirectional design shows similar behavior. The total noise increases with distance from the base, reaches a maximum and then slowly decreases (instead of staying constant). The noise and signal transfer functions roughly track each other, so the minimum detectable signal is relatively invariant to changes in filter parameters.

In the bidirectional cochlea, each active element adds noise, as described in the previous chapter. Intuitively, we expect behavior that is similar to the unidirectional implementation. This is because the bandwidth of the bidirectional cochlea decreases exponentially with position, just like in the unidirectional system. Thus later stages (towards the apex) contribute less noise than earlier stages (towards the base). Indeed, the total output noise of our bidirectional cochlea also increases with distance from the base, reaches a maximum and then slowly decreases.

Implementation: The unidirectional cochlea is easier to design, optimize and implement than the bidirectional version, because the unidirectional filter stages do not interact with each other. Designing and simulating single lumped filters and cascading them is easier than designing a distributed structure like the bidirectional cochlea.

3.7 Appendix: Layout Issues

While testing this chip, we became aware of certain layout problems, which are discussed in this appendix. First, some relevant facts: We used the UMC L130 0.13μm

CMOS process (MM/RF). It contains 9 metal layers: Metals 1 through 6 are 0.32μm thick copper, Metal 7 is 0.8μm thick copper, Metal 8 ("top metal") is 2μm thick copper and AL_RDL is a special pad-connect aluminium layer on top of the "top metal".

To get high quality factors from integrated magnetic components, one usually prevents dummy metal fill from being applied in the region around the inductor by using "dummy block" layers, one each for Diffusion, Poly and Metals 1 through 7. Dummy fill is necessary for chemical-mechanical planarization/polishing (CMP), i.e. making the surface of the inter-layer dielectric (ILD) flat enough for the next metal layer to be deposited. In addition, we also want to prevent slots from being cut into the inductor metallization by using "slot block" layers, one each for Diffusion, Poly and Metals 1 through 7. Metal (and oxide) slots are used for relieving oxide stress built-up during the fabrication process over wide metal layers. Cutting slots in the metal allows the oxide above and below the metal layer in question to contact each other, increasing the mechanical stability of the chip and preventing phenomena such as metal lift-off due to differential thermal expansion. Note that in this process slot rules are only activated if the metallization is $> 5\mu$m wide. Finally, since this is a non-epitaxial process, any regions not covered with N-well are usually doped to create P-wells. To reduce eddy conduction losses in the substrate at high frequencies, a "P-well block" layer is placed below each inductor.

We have fabricated three chips (each 3.240mm $\times$ 1.525mm in size) in this process to date:

- Unidirectional cochlea, version 1 (April 2006): There were 46 stages, with one transformer per stage. Below each transformer were 9 dummy block layers (Diffusion, Poly and Metals 1 through 7), 9 slot block layers (Diffusion, Poly and Metals 1 through 7) and 1 P-well block layer. The chip came back with no dummies or slots near any of the inductors, but was largely non-functional for completely unrelated reasons.

- Bidirectional cochlea (August 2006): There were 50 stages, with one transformer

per stage. To simplify layout, the 9 individual slot and dummy block layers used earlier were replaced with just one layer each, named IND_CAD. It was believed, based on a reading of the design rule manual for the process, that IND_CAD would be expanded by the foundry into the 9 individual layers used on the earlier chip. This assumption proved to be incorrect: it turned out that the IND_CAD layer was only present for historical reasons and was ignored by the foundry. The transformers came back with dummy metal fill under all of them, and metal slots cut into transformer stages 34 through 50. The absence of slots in the first 33 transformers can be explained by the fact that they all had metallization that was $< 5\mu$m wide. Unfortunately, this problem was not spotted at the time.

- Unidirectional cochlea, version 2 (August 2007): There were 51 stages, with one transformer per stage. The two IND_CAD layers were again used, with similar results: all transformers had dummy metal fill below them. Also, stages 39 through 51 (which had metallization that was $> 5\mu$m wide) had metal slots cut into them, while the others did not. The problem was finally noticed during testing of this chip. A micro-photograph of the "slot/no-slot" boundary in shown in Figure 3-34.

It appears that dummy metal fill, because it simply consists of small, electrically floating metal squares, has relatively little effect on the quality factor of inductors and the magnetic coupling coefficient of transformers. Similarly, metal slots are much smaller in size than the metallization itself (see Figure 3-34) and do not noticeably affect the series resistance of the inductor. However, some lowering of quality factors and coupling coefficients may be expected. As evidence of this fact, we note that, experimentally, the series line loss of the bidirectional cochlea chip was about 25% higher than simulated values. Similarly, the peak gain of the unidirectional cochlea was about 25% lower than simulated. We believe that both unmodeled layout parasitics and worse-than-expected transformers are likely culprits.

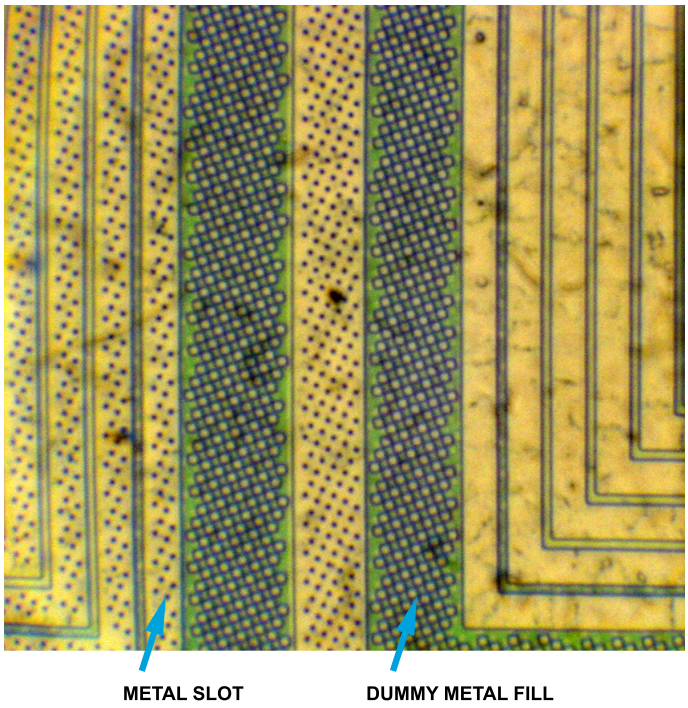


Figure 3-34: Die micro-photograph, taken at 200x magnification, of the second unidirectional RF cochlea chip. The yellow structures are top-layer metal, the green background is the substrate. Parts of two inductors are visible, with a power supply bus separating them. The inductor on the left has oxide-relief slots cut into its metallization, the one on the right does not.

Chapter 4

The RF Cochlea: Enhancements and Applications

You see, wire telegraph is a kind of a very, very long cat. You pull his tail in New York and his head is meowing in Los Angeles. Do you understand this? And radio operates exactly the same way: you send signals here, they receive them there. The only difference is that there is no cat.

– Albert Einstein, when asked to describe radio

In this chapter we describe how to interface the RF cochlea with antennas, how to enhance its performance, and how to use it as a building block for more complex signal processors. In particular, we describe architectures for flexible, programmable frequency estimators and spectrum analyzers that can operate over extremely large bandwidths and yet provide high spectral resolution. The acquisition time and hardware complexity of our algorithms both scale as $O(N)$, where N denotes the number of frequency bins acquired. We also describe several applications of our architectures, including universal and software radios and radar. Parts of this chapter are based on work performed in collaboration with Serhii Zhak.

4.1 Antenna Design

We have designed and tested small broadband antennas for capturing radiated RF from the environment and feeding it into the RF cochlea. Figure 4-1 shows a planar log-periodic dipole antenna (LPDA) that was fabricated on both metal layers of a double-sided printed circuit board. Like the cochlea, the geometry of log-periodic antennas scales exponentially with position. In an LPDA, N half-wavelength dipoles of length l_n $(1 < n < N)$ are separated by distances R_n $(1 < n < N-1)$ and placed on a common feed line. High input frequencies excite the short dipoles and low frequencies, the long ones. The whole structure behaves like a passive cochlea, with exponentially-tapered resonators (the dipoles) fed by a common transmission line. In fact, similar-looking "fishbone" structures have been used to build passive MEMS cochleas [284].

Because of exponential scaling, we have $R_n/R_{n-1} = R_n/R_{n-1} = \tau$, where $\tau = \exp(1/N_{nat})$ is a constant, and N_{nat} is the number of elements per e-fold in length. The minimum and maximum lengths $l_{min} = l_1$ and $l_{max} = l_1 \exp(N/N_{nat})$ set f_{max} and f_{min}, the maximum and minimum frequencies of operation, respectively. Another important design parameter is the taper angle α, where

$$\tan(\alpha) = \frac{l_n}{2R_n} \tag{4.1}$$

The total length of the antenna is given by

$$L_{tot} = \cot(\alpha)\left(\frac{l_{max}}{2} - \frac{l_{min}}{2}\right) = \frac{l_{max}\cot(\alpha)}{2}\left[1 - \exp\left(-N/N_{nat}\right)\right] \tag{4.2}$$

It can be shown that the input impedance of the antenna is resistive between f_{min} and f_{max} and is approximately given by

$$R_{in} \approx \sqrt{\frac{Z_0 Z_a}{N_{nat}\tan(\alpha)}} \tag{4.3}$$

where Z_0 and Z_a are the characteristic impedances of the feed line and the dipole elements, respectively. This expression is similar to that obtained for the input

impedance of the bidirectional RF cochlea. The directivity (maximum, or on-axis gain) of the LPDA increases if the dipoles are spaced further apart and more elements are used, i.e. α decreases and N_{nat} increases. Since $L_{tot} \propto \cot(\alpha)$, we have to trade-off gain for physical size. We used the design curves given in [10] to decide which values of N_{nat} and α to use, given a target directivity of 7.0dBi. Our final values were $N_{nat} = 6.6$ and $\alpha = 22.4°$, resulting in $l_n/R_n = 0.82$.

We had to modify the design procedure outlined in [10] in several ways. The main reason was that we wanted to implement our antenna on a printed circuit board (i.e., a planar geometry), while the original LPDA designs used coaxial feed lines and cylindrical dipole elements. We used a coplanar strips (CPS) line to create a balanced feed line. CPS lines consist of two equal-width conductors separated by a dielectric medium. The characteristic impedance Z_0 decreases as the conductors become wider or the dielectric thinner. In our case, the conductors are copper, etched on either side of the board, and the dielectric is 31-mil thick FR-4, which has a relative dielectric constant of $\epsilon_r \approx 4.2$[1]. Secondly, our radiating dipoles are planar rectangles with a fixed length-to-width ratio. This aspect ratio controls the characteristic impedance Z_a of the dipoles. The two halves of any dipole are attached to the two conductor of the feed line (which are on the top and bottom surfaces of the board) to maintain symmetry and balanced operation. Therefore, unlike normal dipole antennas, the two halves of each dipole are located on different horizontal planes. However, the thickness of the board is much smaller than other antenna dimensions, so any effects on performance are small. We designed Z_0 and Z_a to give us an input impedance $R_{in} = 50\Omega$.

The main electromagnetic effect of fabricating an antenna on the surface of a

[1]FR-4, which stands for "Flame-Retardant 4", is by far the commonest material used for making printed circuit boards. It is hardly an ideal dielectric, however. Its loss tangent at 1GHz is about 0.01 (although it can be as high as 0.02). A more serious problem is that it is an inhomogeneous, layered material, consisting of an epoxy resin that has fiberglass mats embedded in it to increase mechanical strength. The resin has lower dielectric constant than the fiberglass, so the effective dielectric constant of the material as a whole is a weighted average of the two values. If antennas or transmission lines are fabricated on an FR-4 surface, however, they will "see" a dielectric constant that is higher or lower than expected depending on how far away from the surface the nearest fiberglass layer is. This uncertainty introduces unwanted variability in the performance of the electromagnetic component.

dielectric, as opposed to free space, is that it sees an effective dielectric constant ϵ_{eff}, that is different from 1. As a result, in order to maintain the same electrical performance all antenna dimensions must be decreased by a factor $1/\sqrt{\epsilon_{eff}}$. The value of ϵ_{eff} depends on the permittivity and electrical size of the dielectric, as well as the radiation pattern of the antenna. At the interface between two infinitely thick dielectric slabs, for example, it is intuitively clear that ϵ_{eff} for an isotropic radiator will simply be the average of the permittivities of the two media. In our case, the situation is more complicated. Assume that the antenna is mounted in free space (effectively, air), so that each radiating element has a 31-mil thick layer of FR-4 right next to it, and free space everywhere else. The resulting value of ϵ_{eff} will be higher for a short dipole than a long one, since the former operates at high frequencies, where the FR-4 looks electrically thicker than at the low frequencies where the latter operates. Variation in ϵ_{eff} becomes an important issue as the ratio f_{max}/f_{min} increases. We used a linear approximation, based on electromagnetic simulations of single dipoles of various lengths, to quantify the variation in ϵ_{eff} with frequency:

$$\epsilon_{eff}(f) \approx \epsilon_{eff}\left(f_{max}\right) + \left(\epsilon_{eff}\left(f_{max}\right) - \epsilon_{eff}\left(f_{min}\right)\right)\left[\frac{f - f_{min}}{f_{max} - f_{min}}\right] \tag{4.4}$$

Since the center frequency $f_n = f_{max}\exp\left(-n/N_{nat}\right)$ of each dipole is known, we can now compensate for ϵ_{eff} by "pre-distorting", i.e. changing the length of the n-th dipole to $l'_n = l_n/\sqrt{\epsilon_{eff}\left(f_n\right)}$. A Matlab script was written to automatically generate the layout of the LPDA (as a CIF file) based on design parameters specified by the user. The antenna shown in Figure 4-1 (17" $\times$ 12.5" in size) used $f_{min} = 430$MHz, $f_{max} = 5$GHz, $\epsilon_{eff}\left(f_{min}\right) = 1.15$ and $\epsilon_{eff}\left(f_{max}\right) = 1.60$.

The antenna was fed from the high-frequency end, while the low-frequency end of the feed line was terminated using an external load R_L. Figure 4-2 shows measured values of the input reflection coefficient to a 50Ω source for three different values of R_L: 0 (short), ∞ (open) and 50Ω (matched). In all three cases we have good impedance matching ($VSWR < 2$, or $|S_{11}| < -10$dB) between 400MHz and 6GHz. The matched termination also provides a good impedance match at low frequencies

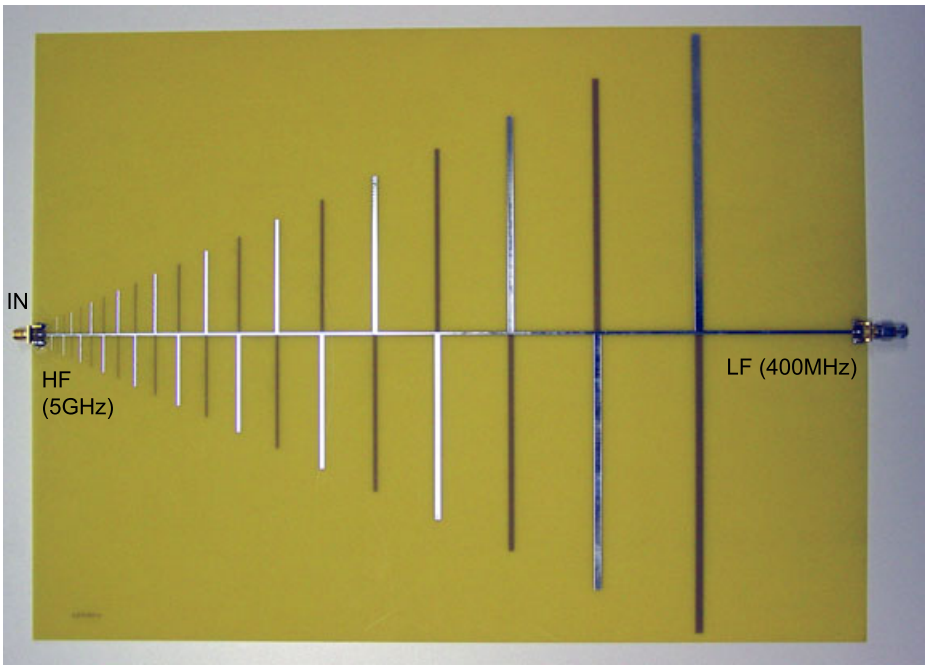


Figure 4-1: A planar, broadband log-periodic dipole antenna (LPDA) that can be interfaced with the RF cochlea for collecting radiated RF energy from the environment.

(< 400MHz), as expected. Figure 4-3 shows measured directivity (on-axis gain) of the antenna with a matched low-frequency termination. We see that the average gain is 5dBi between 400MHz and 3GHz[2]. We designed the LPDA geometry to have an on-axis gain of 7dBi. Some of the reduction in gain is due to dielectric loss in the FR-4. Similar antennas (with different values of f_{min} and f_{max}) have been successfully interfaced with both unidirectional and bidirectional RF cochlea chips.

4.2 Enhancements of the RF Cochlea

We now discuss several enhancements of the basic RF cochlea structure. It is to be understood that what follows applies to both unidirectional or bidirectional cochlear structures (unless mentioned otherwise). Important nonlinear aspects of cochlear operation include spectral masking, enhancement and automatic gain control. The cochlea is energy-efficient because it uses distributed gain to get a large gain-bandwidth

[2]This particular measurement was made with a network analyzer that was limited to 3GHz. Other measurements (not shown here) indicate that the gain continues to exceed 5dBi up to 5GHz.

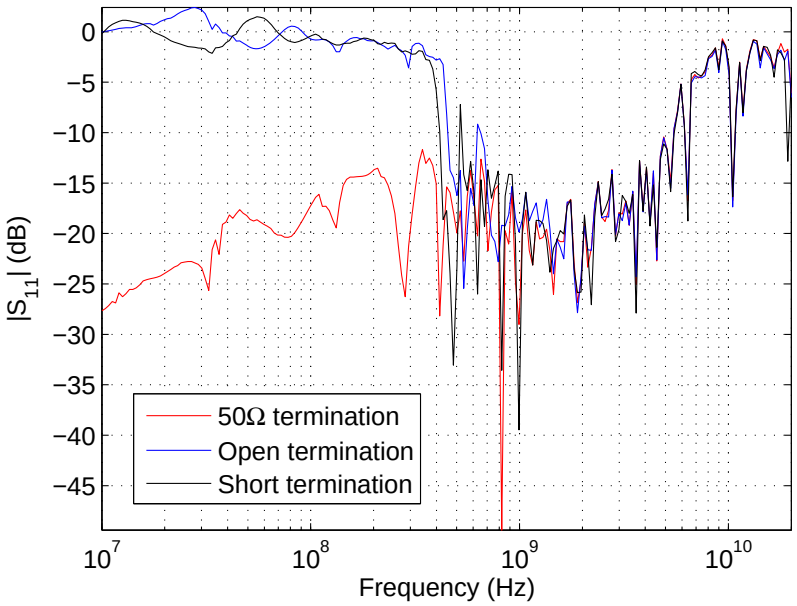


Figure 4-2: Measured input reflection coefficient of the LPDA, measured at the high-frequency end. The low-frequency end was terminated with three different loads: an open, a short and a matched (50Ω) load, resulting in the three curves shown.

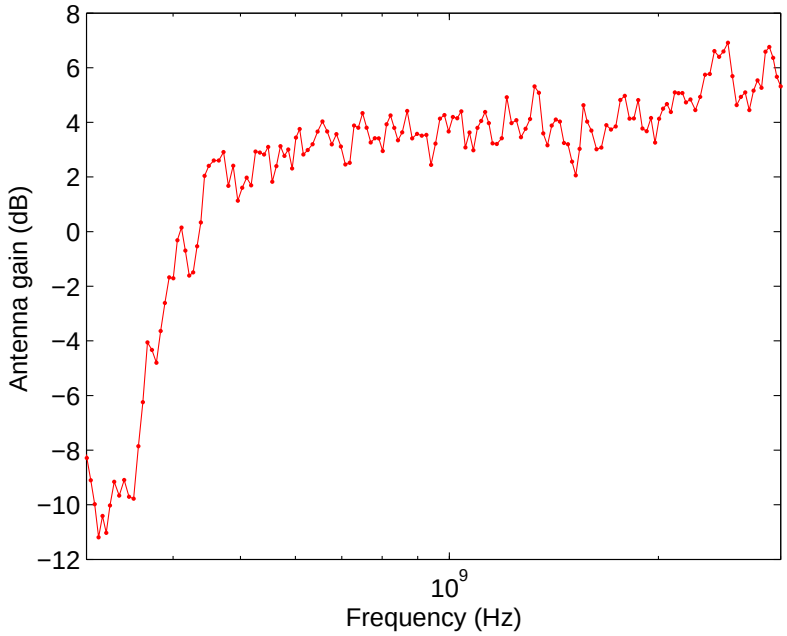


Figure 4-3: Measured far-field on-axis gain of the LPDA, measured at the high-frequency end with the low-frequency end terminated with a 50Ω load.

product. Let us compare the power consumption of the bidirectional RF cochlea with an all-digital software-defined radio which has the same speed (bandwidth) and precision (signal-to-noise ratio or number of bits). To directly digitize 7GHz of bandwidth with 50dB of dynamic range, we need 16GS/s and 9 bits of precision. This step requires 7W even if we assume a state-of-the-art ADC that consumes only 1pJ/quantization level. By contrast, the bidirectional RF cochlea has a bandwidth of 7GHz and an output SNR of greater than 50dB but only consumes 180mW of power.

The RF cochlea has inherently higher dynamic range than audio-frequency silicon cochleas, mainly because integrated passive inductors can be used at RF. As shown earlier, active inductors, which produce more noise as passive inductors with the same quality factor Q, must be used at audio. Nevertheless, cochlea-like structures that take advantage of the ideas described in this chapter can also be used for spectral analysis at lower frequencies.

Different process technologies are optimally suited for hardware implementations of the cochlea at various frequency ranges. For example, MEMS cochleas using masses and springs are attractive at audio (in the kHz range). Power-efficient electronic implementations at these frequencies are also possible using analog subthreshold MOS circuits. Similar circuits using BJT's are suitable for ultrasonic and VHF cochleas. MOS and BJT cochleas can be realized using either voltage-mode or current-mode signal processing. Implementations that are partially or completely digital are also possible. In this thesis we have focused on CMOS RF cochleas that use lumped transmission lines and passive LC filters; these structures are best suited for UHF and low-GHz frequencies. Above 10GHz distributed cochleas using true transmission lines and high-speed compound semiconductor devices become feasible.

An important characteristic of the RF cochlea is masking, where the presence of a strong tone suppresses the response to smaller tones both above and below it in frequency. However, the low-pass response of the cochlea means that masking is asymmetric: large low frequency tones suppress small high frequency tones very effectively and not the other way around. A way to modify this behavior is to use the difference in output signals across each cochlear stage as the output, not the signal

itself, as shown in Figure 4-4(a). This operation is equivalent to spatial differentiation, and converts the low-pass cochlear response to a band-pass one. Simulated results are shown in Figure 4-5 for the unidirectional cochlea. The small, highest-frequency tone is normally almost drowned out; however it is resolved clearly when stage differences are used as outputs.

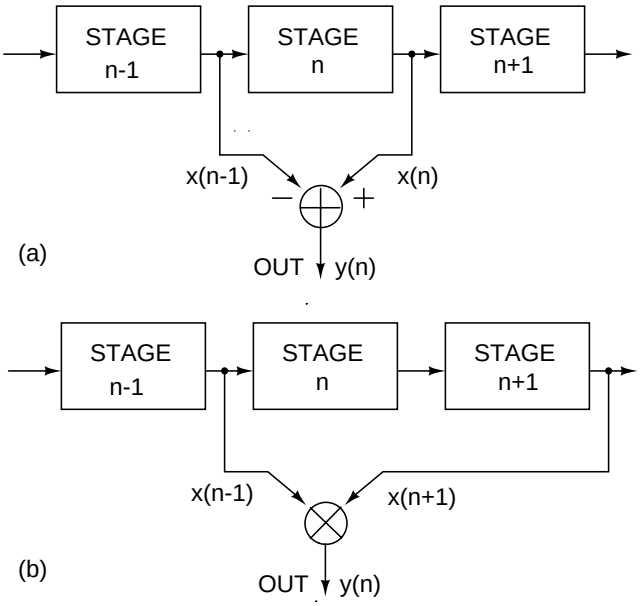


Figure 4-4: Spectral sharpening schemes for the RF cochlea: (a) a linear scheme and (b) a nonlinear scheme inspired by the action of coincidence-detecting cells in the biological cochlear nucleus.

In addition to the linear filtering provided by the RF cochlea, nonlinear spectral sharpening strategies can also be used. These techniques can further improve the frequency selectivity of the system by utilizing spatiotemporal correlations between the outputs of cochlear stages [268]. One simple strategy, inspired by the action of coincidence-detecting "octopus" cells in the biological cochlear nucleus (CN), is shown in Figure 4-4(b) [303]. The CN receives inputs from the cochlea via the auditory nerve, and projects to structures higher up in the auditory pathway, such as the superior olivary complex and the inferior colliculus. Its main task is to extract simple spatio-temporal features, such as the onset of a sound and the direction of frequency sweeps (chirps).

In our strategy, outputs of cochlear stages two positions apart in the cascade are

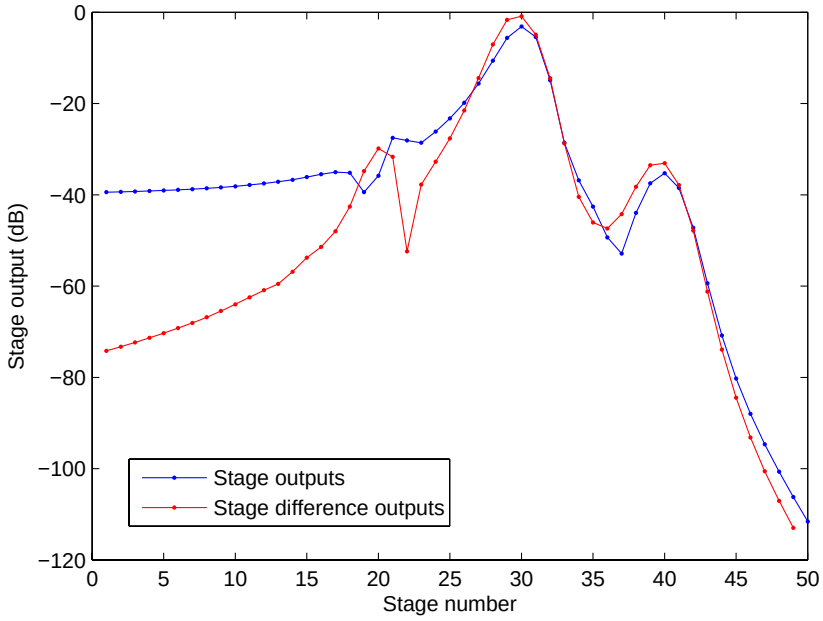


Figure 4-5: Simulated outputs (spatial response) of the unidirectional RF cochlea to three input tones spaced one octave apart. The middle tone is 30dB larger than the other two.

multiplied together to produce the sharpened output. The multipliers act as analog correlators that detect phase-coincidence, thus combining phase and magnitude information present in the cochlear outputs. The average output of a multiplier driven by two sinusoidal inputs with a phase difference θ between them is given by

$$\overline{y} = \overline{x\sin(\omega t) \times x\sin(\omega t + \theta)} = \frac{x^2}{2}\cos(\theta) \tag{4.5}$$

where the averaging operation is carried out over a complete input cycle. We see that the average output magnitude is maximal, and equal to the average power of each input signal, when $|\cos(\theta)| = 1$, i.e., when $\theta = \{0, \pi\}$. The output is zero when the inputs are in quadrature, i.e., $\theta = \{\pi/2, 3\pi/2\}$.

In our case θ corresponds to the sum of the phase differences across two adjacent cochlear stages. At frequencies much lower than the local center frequency θ is small and the output magnitude is maximal. The phase shift θ increases with frequency, causing the output magnitude to decrease. At frequencies just higher and lower than

the center frequency $\theta = \pi/2$, resulting in nulls (zero output). Finally, at the center frequency $\theta \approx \pi$, the inputs to the multiplier are almost in phase, and the output magnitude is again maximized. Thus we expect $y(n)$ to exhibit nulls on either side of the peak, but otherwise track the input amplitude $x(n)$. We refer to this process as spectral sharpening, because it effectively decreases the width of the peak. In fact, it can be shown that our strategy approximately doubles the frequency resolution with no loss in timing precision. Simulation results, shown in Figure 4-6, confirm our predictions.

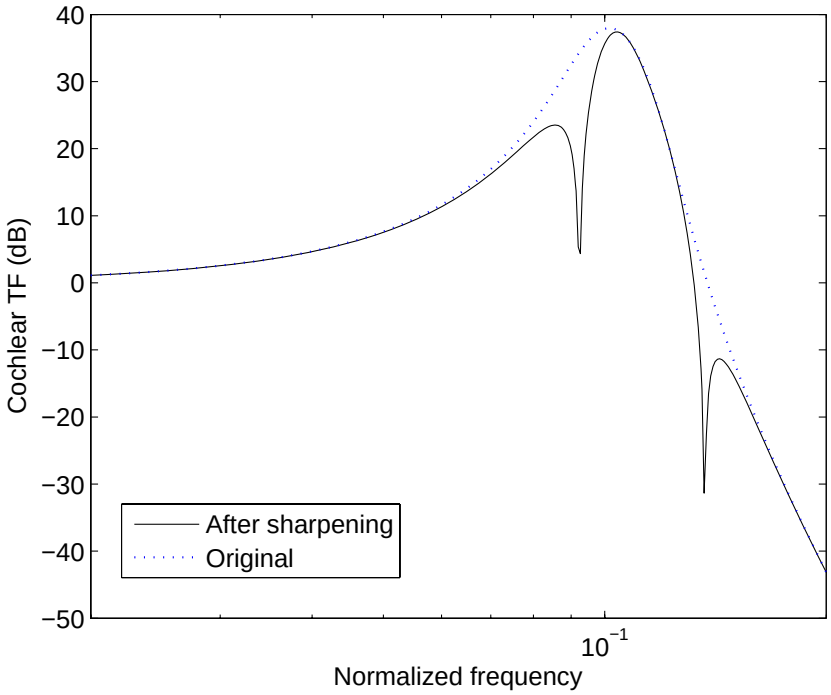


Figure 4-6: Simulated unidirectional cochlea transfer functions at different spatial locations before and after spectral sharpening.

Both strategies shown in Figure 4-4 are used in biological auditory and visual systems [268]. The first, linear strategy is implemented using lateral inhibition networks that compute spatial derivatives. The outputs of lateral inhibition networks are then fed into networks of coincidence detectors. The actions of these neurons can be modeled using analog multipliers, as in our second, nonlinear strategy. We can combine the two strategies in a similar way to efficiently extract features, such as edges and

peaks, from the RF spectrum.

4.3 Architectures for Universal or Software Radio

4.3.1 Background

In this section, we define *radios* simply as devices that communicate with other radios via wireless (radiated or near field electromagnetic energy). The commercially interesting range of frequencies for wireless applications is very large, extending over approximately three orders of magnitude from 100MHz to 100GHz. Nevertheless, this resource is becoming increasingly crowded. Next-generation radios will need to be smart enough to be able to find any available space in the RF spectrum, and flexible enough to be able to operate there [114]. Furthermore, some of these next-generation *software, universal or cognitive* radios may need to serve as *translator* or *gateway* radios in a heterogenous wireless network. Gateway radios receive signals in different formats and frequencies from input radios, translate them to other formats and then transmit them to other radios.

Initial attempts to define software radio architectures emphasized analog-to-digital conversion of the entire RF bandwidth immediately after the antenna in the receive path, and digital-to-analog conversion immediately before the antenna in the transmit path [127, 128]. All other signal processing was to be performed digitally. However, no practical radios of this type have been built: the power consumption involved in digitizing the entire RF bandwidth is prohibitively high, and the situation is not likely to change in the near future. A flexible receiver architecture that has recently received attention is the *software-defined radio receiver* [1,7]. This architecture is intermediate in flexibility and power consumption between "true" software radio receivers and traditional narrowband receivers: it is capable of receiving a *single* channel with any bandwidth situated in any band. Other flexible receiver architectures reduce the bandwidth that the analog-to-digital converter (ADC) in the software radio must handle by performing analog pre-processing [228, 243]. However, the other half of a

software radio, namely the transmitter, has received much less attention.

In this section we describe novel frequency analysis and estimation algorithms that allow only interesting portions of the RF spectrum to be selected and digitized. Our algorithms are based upon a new understanding of the conceptual similarities between voltage estimators, i.e., analog-to-digital converters, and frequency estimators. Their use results in universal radios which have the following advantages over current implementations:

- Wide fractional bandwidth combined with high temporal resolution
- Fast signal acquisition
- Increased agility and programmability
- Low power consumption
- Low hardware complexity

We also show that efficient implementations of our algorithms can be realized by using transmission lines and filter cascades that have characteristic frequencies varying exponentially with position.

4.3.2 Analogies between ADC's and frequency estimators

A signal can have many frequencies at once. We define *frequency estimators* as systems that solve the general problem of estimating the instantaneous frequencies present in a signal. *Spectrum analyzers* are frequency estimators that can, in addition to estimating the frequencies present, also determine their amplitude, phase or power levels. In other words, spectrum analyzers find the power spectral density of a signal. Spectrum analyzers thus form a subset of frequency estimators. They may also be viewed as a parallel, broadband generalization of narrowband radios that estimate amplitude or phase of a narrowband signal centered around a single carrier frequency. Define the minimum and maximum input frequency of interest to a spectrum analyzer

to be f_L and f_H respectively. The bandwidth ratio β characterizes the "frequency-domain dynamic range" required by the corresponding receiving system, where

$$\beta = \frac{f_H}{f_L} \tag{4.6}$$

In general, spectrum analyzers separate an incoming signal spread over a bandwidth $B = f_H - f_L$ into N outputs ("bins") that each span smaller bandwidths which (ideally) don't overlap. The total power present in each bin is an approximation of the local power spectral density.

We have found interesting analogies between frequency estimators and analog-to-digital converters (ADC's) and used it to develop new[3] frequency estimation algorithms based on well-known ADC algorithms. The problems that frequency estimators and ADC's solve are similar, but in different signal domains. ADC's find the value of an unknown analog amplitude (voltage or current); frequency estimators do the same with an analog frequency variable. As a result, similar solution techniques work in the two cases. In particular, ADC algorithms and automatic error-calibration techniques can be easily adapted for frequency estimation. However, a signal can have many instantaneous frequencies of interest, but only one instantaneous amplitude. A better analogy is therefore to view frequency estimators as the frequency-domain analogs of multiple parallel ADC's. Each ADC corresponds to estimating the frequency value (and possibly the power content) of one output bin. These analogies between ADC's and frequency estimators enable us to search for frequency-domain analogs of most well-known ADC topologies. The result is the discovery of several new architectures for frequency estimation. In addition, those architectures that turn out to be already known are seen from a new perspective, improving our understanding of their relative merits and disadvantages. The analogies are summarized in Table 4.1.

An important analogy between ADC and frequency estimator architectures is shown in Figure 4-7. The basic operation of subtraction in the amplitude domain (voltage or current) used by ADC's has a natural frequency-domain analog through

[3] As far as we know.

Table 4.1: Analogous ADC and spectrum analyzer architectures

ADC	**Frequency Estimator/ Spectrum Analyzer**	**Remarks**
Single-slope	Swept-sine (super-heterodyne)	Known
Dual-slope	Dual swept-sine	New
Flash (parallel)	Filter bank	Known
Flash-interpolation	Filter bank with finite filter roll-off slope	Known
Successive sub-ranging	Successive sub-ranging[1]	New
Successive approximation	Cascaded super-heterodyne[2]	New
Algorithmic	[1] or [2]	New
Pipelined	[1] or [2]	New
Sigma-delta (over-sampled)	Digital FM demodulator	New

mixing (super-heterodyning). However, if real signals are used in the frequency domain, only an unsigned subtraction (difference) can be obtained by mixing since positive and negative frequencies cannot be physically distinguished from each other. This case is shown in Figure 4-7(a), where the low-pass filter (LPF) rejects the sum frequency $\omega_1 + \omega_2$ also produced by the mixer, but passes the desired difference frequency component $|\omega_1 - \omega_2|$. A signed subtraction can be performed in the frequency domain if complex signals are used. Examples are shown in Figures 4-7(b) and 4-7(c). These complex multiplier structures use side-band cancelation to reject unwanted frequency components and produce only the desired component, i.e., $\exp(\jmath(\omega_1 - \omega_2)t)$ at the output.

We now describe several common ADC architectures and frequency estimators that are analogous to them. Some of these estimators are well-known, while others are novel. It should be noted that we shall use real signals everywhere in what follows unless mentioned otherwise. However, it is straightforward to extend any of the frequency estimator and spectrum analyzer structures that we describe to the complex signal domain. For this purpose, the complex multipliers shown in Figures 4-7(b) and 4-7(c) can be used to replace the simple multiplier shown in Figures 4-7(a). Frequency estimators that use only real signals can be used to demodulate frequency-modulated waveforms. The use of complex signals allows phase-modulated waveforms to be demodulated as well.

The most obvious way to build a broad-band spectrum analyzer is to use multi-

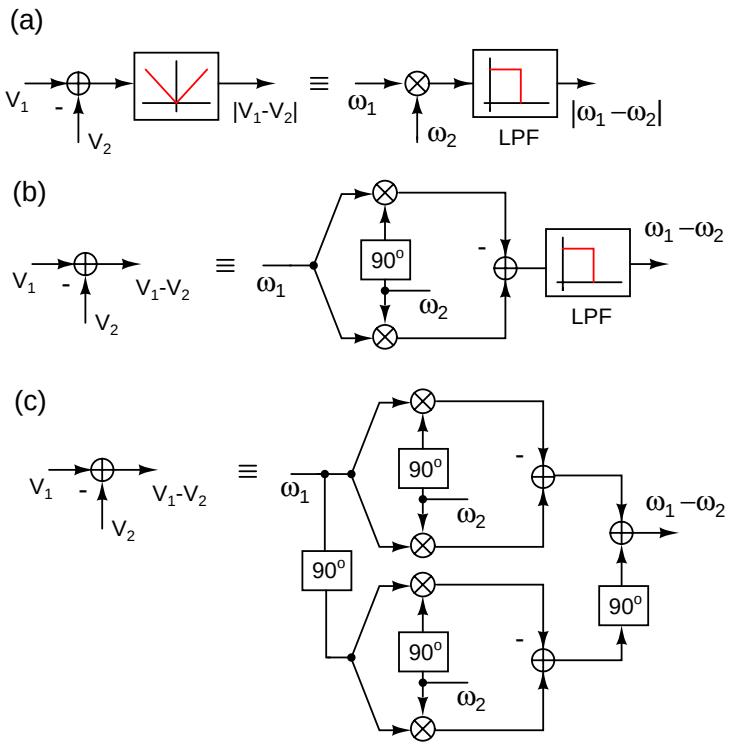


Figure 4-7: The analogy between difference operations in analog-to-digital converters (left) and frequency estimators (right) with (a) real input signals, (b) and (c) complex input signals.

ple narrow-band frequency estimators that operate in parallel (simultaneously). This "filter bank" approach corresponds to the parallel or "flash" ADC. Flash converters are extremely fast (they convert in constant time, i.e, $O(1)$) but are extremely power-hungry. When the filters in the filter bank have finite roll-off slope their frequency responses can overlap. As a result a single input frequency will produce non-zero outputs from multiple filters at the same time. The amplitudes and phases of these outputs provide extra information that can be used to effectively improve the resolution of the frequency estimator. This approach corresponds to the "flash-interpolation" ADC structure.

Swept-sine (tunable super-heterodyne) spectrum analyzers mix the input frequency with a local oscillator (LO) whose frequency is ramped linearly with time. The power level present at the output of the mixer is estimated by a single super-heterodyne receiver. Different frequency components in the input reach the pass band of the output receiver at different times. Effectively, therefore, the architecture converts frequency to time; this corresponds to serial ADC's such as single-slope converters. However, in contrast to the ADC, the swept-sine architecture can produce multiple outputs (frequency estimates) in a single sweep. Nevertheless, both structures use linear sweeps, which make them extremely slow over large dynamic ranges in signal amplitude (ADC's) or frequency (spectrum analyzers). An exponential or geometrically-scaled sweep synthesizes a wavelet-like transformation (a constant-Q structure) that is much faster in such situations. The biological cochlea is an example of a spectrum analyzer that has constant-Q characteristics. The analogous ADC structure is a single-slope converter implemented in the logarithmic signal domain.

Dual-slope ADC's are serial converters that use two sweeps (ramps), usually (but not necessarily) in voltage. An upward ramp with a slope proportional to the input signal is first performed for a fixed time T_{ref}. The voltage at the end of this ramp is $V_0 + \alpha T_{ref} X_{in}$ where X_{in} is the input signal, α is a constant and V_0 is the initial voltage. This is followed by a downward ramp with fixed slope αX_{ref}, where X_{ref} is a constant. The time when the voltage reaches V_0 is given by $T = T_{ref} X_{in}/X_{ref}$. Thus $T \propto X_{in}$ as long as T_{ref} and X_{ref} are constant, thus converting time to voltage.

The advantage of this technique over a single-slope is that T does not depend on the constant α (its effects are canceled out). Thus any uncertainties or nonlinearities in α don't affect the result of the conversion. We can use the same principle to build a dual-slope frequency estimator. In the first phase, the input frequency ω_{in} is ramped up for a fixed time T_{ref} with, for example, a frequency synthesizer. During the second phase a reference frequency ω_{ref} is ramped down in time by using a similar synthesizer. The time at which the two frequencies are equal encodes the value of the input frequency. Note that, unlike the single-slope swept-sine system, only a single frequency can be estimated per sweep with this technique.

Frequency-domain analogs of over-sampled (sigma-delta) ADC's also exist. Figure 4-8 shows analogous sigma-delta architectures for ADC's and frequency estimators. A digitally-controlled oscillator (DCO) replaces the digital-to-analog converter (DAC), but otherwise the two structures are identical. The first component is a subtractor that calculates the instantaneous difference between the output of the DAC (or DCO) and the input amplitude (or frequency). This error signal is fed into an integrator. The output of the integrator drives a comparator, which in turn drives the DAC (or DCO), thereby completing the loop. The output of the comparator is a digital signal which can be low-pass filtered (decimated) in the digital domain to produce the output bits.

Figure 4-9 is an example of how the sigma-delta frequency estimator structure shown in Figure 4-8(b) may be implemented in practice. Note that the sign of the frequency difference that gets fed into the integrator is important, so the simple multiplier and low-pass filter combination shown in Figure 4-7(a) cannot be used. Instead, we use the complex multiplier shown in Figure 4-7(c), which allows us to retain the sign of the frequency difference in the phase of the complex output signal $\exp\left(\jmath\left(\omega_{IN}-\omega_{DC0}\right)t\right)$. In order to avoid problems with phase wrapping, we find the phase of the ratio of this signal and a version of itself delayed by a small constant time $\Delta T < 1/\left|\omega_{IN}-\omega_{DC0}\right|$. This operation is carried out by the 'DIV' and 'ANG' blocks in the figure. The output is $\left(\omega_{IN}-\omega_{DC0}\right)\Delta T$, which is the error signal fed into the integrator. The rest of the system is the same as shown in Figure 4-8(b).

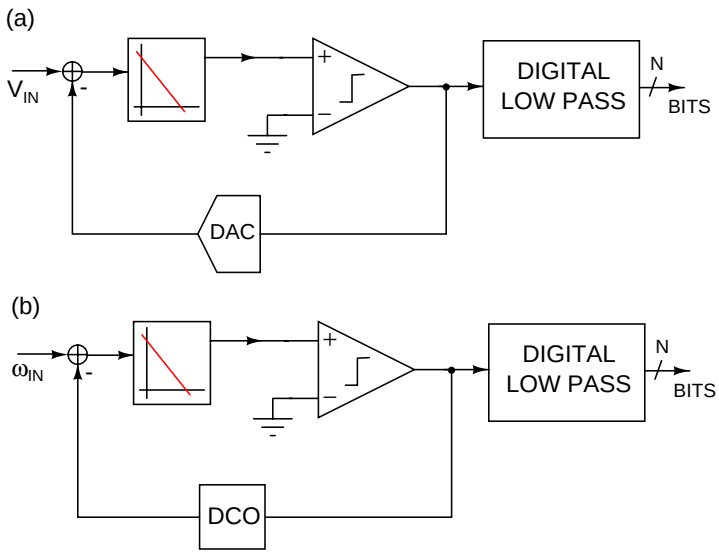


Figure 4-8: Analogous sigma-delta architectures for (a) analog-to-digital conversion and (b) frequency estimation (FM to digital conversion).

Figure 4-10 shows simulation results from the sigma-delta frequency estimator of Figure 4-9. The input frequency was varied sinusoidally and the digital output of the comparator in the loop was low-pass filtered to produce the estimated frequency. We see that the loop faithfully tracks the input frequency (with a slight delay introduced by the output low-pass filter). The whole system acts as a direct frequency-to-digital converter and may prove useful for demodulating frequency modulated (FM) or frequency shift-keyed (FSK) waveforms in software-defined radios or other applications.

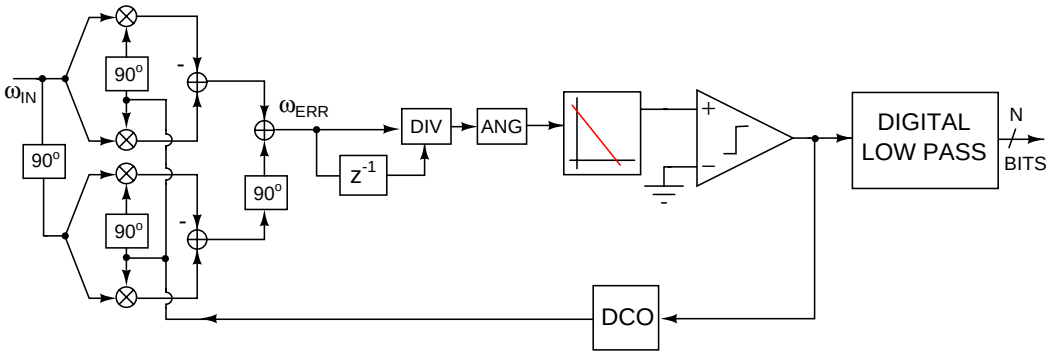


Figure 4-9: Block-level implementation of the sigma-delta frequency estimator architecture shown in Figure 4-8(b).

We now describe the analogy between successive-subranging and successive-approximation

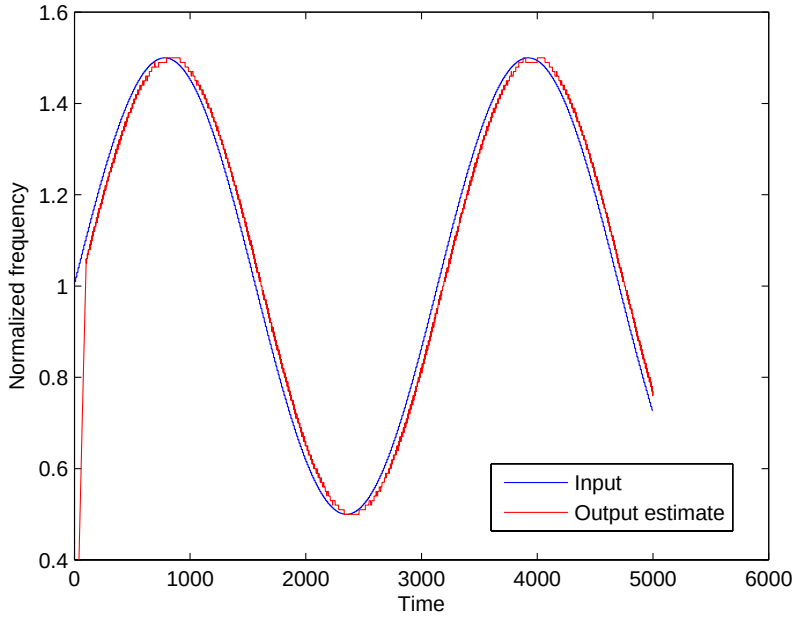


Figure 4-10: Simulation results of the sigma-delta frequency estimator shown in Figure 4-9. The input frequency was varied sinusoidally about a normalized frequency of 1 and the output of the comparator was low-pass filtered to produce the estimated output frequency.

ADC's and frequency estimators in more detail. The n-th conversion stage in successive-subranging ADC algorithms is shown in Figure 4-11(a). This stage produces one output bit and is cascaded successively in order to get more bits. The output of the previous conversion stage, V_{n-1}, is an analog signal, where $V_{REF}-V_{FS} < V_{n-1} < V_{REF}+V_{FS}$ and V_{REF} and V_{FS} are the reference and full-scale voltages, respectively. This signal is digitized by a comparator to produce $b_{n-1} \in [-1, 1]$, the next output bit. This bit is converted into an analog signal by a one-bit digital-analog converter (DAC) whose output voltages are $(V_{REF} + b_{n-1}V_{FS})/2$. This signal is subtracted from the original input signal, V_{n-1} to extract a residue signal that is then amplified by a factor of two to produce V_n, the input to the next conversion stage.

The analogous successive-subranging frequency estimation structure is shown in Figure 4-11(b). The two structures are functionally equivalent but operate in different signal domains: signal amplitude (voltage or current) and frequency, respectively. The reference and full-scale voltages are replaced by the center frequency ω_{REF} and band-

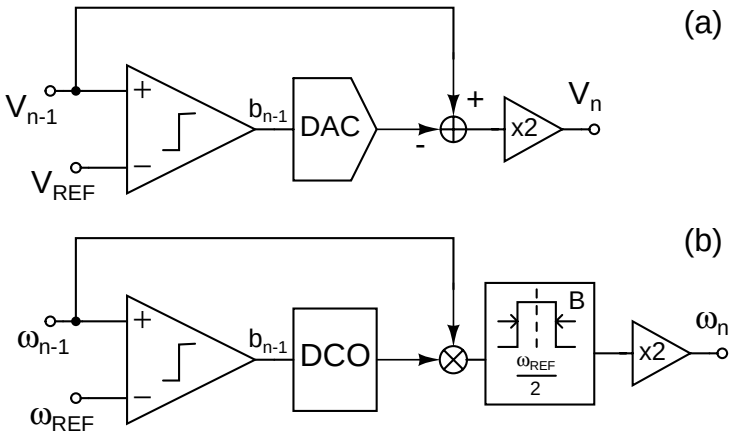


Figure 4-11: A single successive-subranging stage for (a) an analog-to-digital converter and (b) a frequency-to-digital converter (frequency estimator).

width B. Therefore $\omega_{REF} - B < \omega_{n-1} < \omega_{REF} + B$, where $\omega_{REF} > B$. The voltage comparator is replaced by a frequency comparator, the DAC by a digitally-controlled oscillator (DCO) that produces $\omega_{DCO} = (\omega_{REF} + b_{n-1}B)/2$ and the subtractor by a multiplier (mixer) followed by a bandpass filter (BPF). The condition $\omega_{REF} > B$ ensures that $\omega_{n-1} > \omega_{DCO}$, i.e., the frequency difference is always positive and the multiplier and BPF together act as a linear subtractor in the frequency domain. It is interesting to note that, in contrast to the amplitude (ADC) domain, accurate multiplication and division by integer ratios is easy in the frequency domain. Well-known circuits for frequency division and multiplication can be used for this purpose. Harmonics and sub-harmonics of the signal itself may also be used.

Generalizations of the structures shown in Figures 4-11(a) and 4-11(b) to multiple output bits (say $n_1 > 1$) per stage are straightforward. In this case the comparator and DAC must be modified to produce and accept n_1 parallel bits, respectively. Also, the residue must be amplified by 2^{n_1} at the end.

Successive-approximation and successive-subranging ADC structures are similar except for the absence of the residue-amplification (multiply-by-two) step in the former. The n-th stage of a successive-approximation ADC is shown in Figure 4-12(a). We see that it is similar to Figure 4-11(a), except for changed signal ranges and values. The input is $V_{REF} - V_{FS} < 2^{n-1}V_{n-1} < V_{REF} + V_{FS}$, the comparator ref-

erence is $V_{REF}/2^{n-1}$ and the DAC output is $(V_{REF} + b_{n-1}V_{FS})/2^n$. The equivalent frequency estimator structure is shown in Figure 4-12(b), where ω_{REF} and B again replace V_{REF} and V_{FS}. Successive subranging is usually an advantage for ADC's because it reduces hardware performance requirements and noise contributions of later conversion stages and makes pipelining easier. For frequency estimators eliminating residue-multiplication saves hardware and power (approximately a factor of two, assuming power is proportional to bandwidth) but increases analysis time since the later, lower-bandwidth stages take longer to settle. We shall only describe successive-approximation frequency estimation algorithms in detail below since the generalization to successive-subranging architectures is straightforward.

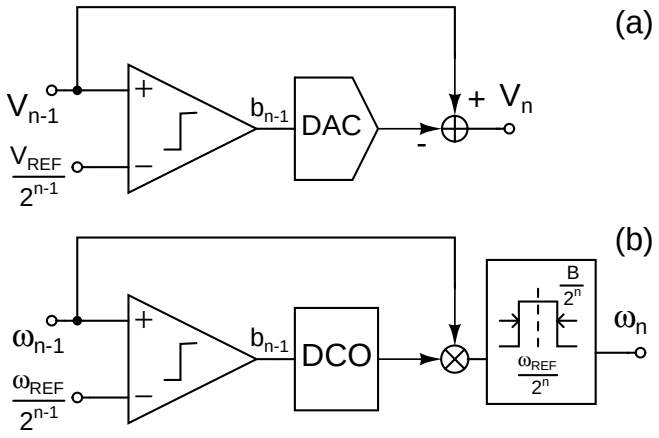


Figure 4-12: A single successive-approximation stage for (a) an analog-to-digital converter and (b) a frequency-to-digital converter (frequency estimator).

An advantage of the successive-subranging and successive-approximation structures shown in Figures 4-11 and 4-12 is that they can be pipelined to increase throughput. Assume several stages are cascaded to perform a multi-bit conversion; pipelining can be enabled simply by adding sample-and-holds between stages, as shown in Figure 4-13(a). Adding a sample-and-hold also allows a single stage to be used for multi-bit conversions by using feedback. This is the basis of the cyclic or algorithmic ADC architecture. In this architecture a single approximation or subranging stage is used inside a feedback loop as shown in Figure 4-13(b). The conversion proceeds in M steps, where M is the total number of output bits. The approxima-

tion or subranging stage produces a single bit and residue in one time step. This residue is then fed back into the same stage during the next time step to produce the next bit. The key feature of algorithmic converters is hardware reuse, which reduces throughput by preventing pipelining but eliminates mismatches between stages. Note that successive-approximation stages (unlike successive-subranging stages) need to be "re-tuned" at every time step during a algorithmic conversion. This is because the comparator reference and BPF bandwidth during the m-th time step are both proportional to 2^{-m}. Successive-subranging stages keep these quantities constant because they multiply the residue by two at the end. Pipelined and algorithmic approaches can be extended to frequency estimation in a straightforward way. The same basic subranging and approximation stages shown in Figures 4-11 and 4-12 can be used. An injection-locked oscillator or frequency-locked loop (FLL) can be used as the frequency-domain analog of a sample-and-hold.

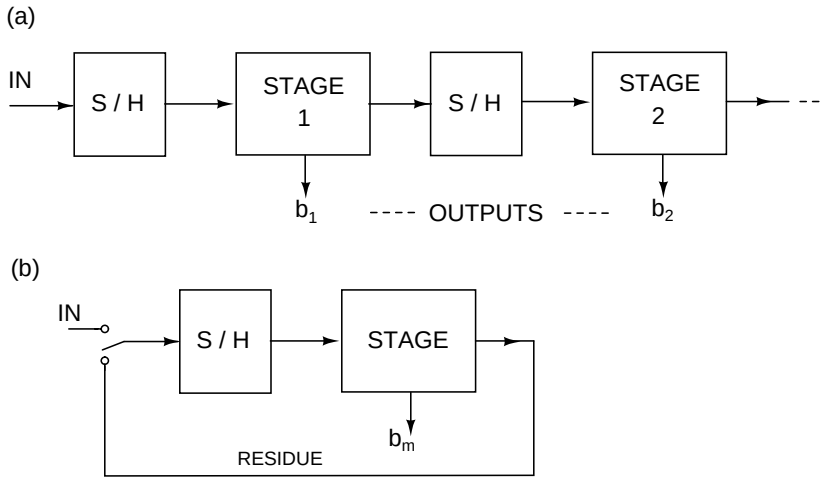


Figure 4-13: Use of successive-subranging or successive- approximation stages for multi-bit analog-digital or frequency-digital conversion by using (a) pipelining and (b) a cyclic or algorithmic architecture.

A broad-band radio architecture that uses successive approximation to perform hierarchical, parallel estimation of multiple frequencies is shown in Figure 4-14. The logic behind this architecture is that, over large bandwidths, the real radio spectrum is extremely sparse. It is dominated by a limited number of relatively narrow-band signals which cumulatively occupy only a small fraction of the total bandwidth. Some

of these signals are of interest to the user, while others are not, and are referred to as interferers. In practical situations these interfering signals can be much larger than the signal of interest. A system that observes several relatively narrow-band signals (placed arbitrarily in frequency space) in parallel would therefore be able to monitor essentially all signals of interest in an extremely large radio bandwidth. Our architecture efficiently solves the problem of estimating where these observed regions of the spectrum should be placed.

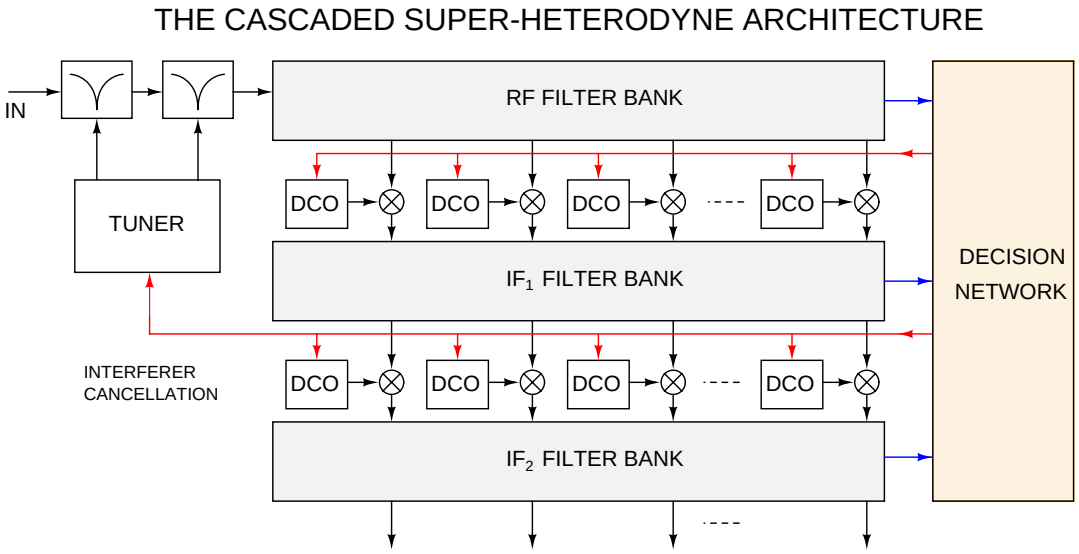


Figure 4-14: A successive-approximation architecture for fast, hierarchical frequency estimation of multiple narrowband signals. Interfering tones can be rapidly estimated to any degree of precision and then canceled out using tunable notch filters.

Consider Figure 4-14. The advantage of this *cascaded super-heterodyne architecture* is that it allows us to monitor N signals of interest in parallel with high temporal resolution and any desired spectral resolution. Here N is the number of filters in each filter bank. The architecture is also efficient in terms of hardware requirements because it only needs M filter banks and $M \times N$ mixers to increase spectral resolution by a factor of α^M. The outputs of each filter bank are bandpass signals with center frequencies that scale arbitrarily with position (linear or exponential scalings are common). The bandwidth of these signals is successively reduced by the mixers and succeeding filter bank. The output of the m-th filter bank has a bandwidth B_m. When bandwidth is reduced by a constant factor α at each step, B_m decreases exponentially

with m, i.e.,

$$B_m = \frac{B_{m-1}}{\alpha} = \frac{B_0}{\alpha^m} \tag{4.7}$$

A common value for α is two, which corresponds to the canonical situation shown in Figure 4-12(b). As in that figure, the decision network in Figure 4-14 consists of frequency comparators and oscillators that determine which part of the input bandwidth B_{m-1} should be included in the output bandwidth B_m. Frequency comparators can be built in several ways. For example, the total energy present on either side of the center frequency ω_{REF} can be integrated and the two results compared. The result of the comparison is a decision on whether the frequency being estimated is higher or lower than ω_{REF}. Thus, the output of each successive filter bank allows the user to examine signals of interest in greater and greater detail. In addition, the outputs of the decision network are digital signals that are the results of N parallel frequency-to-digital conversions (analogous to analog-to-digital conversions). It should also be noted that the structure retains amplitude information about each output signal and thus acts as a spectrum analyzer as well as a frequency estimator.

One application of the cascaded super-heterodyne structure is to allow weak signals to be detected in the presence of strong interferers. A simple interference cancelation strategy is shown in Figure 4-14. The cascaded super-heterodyne structure is used to accurately estimate the frequencies and bandwidths of unwanted interferers. These are then canceled out by using tunable notch filters at the input of the system. Simulated results of this scheme are shown in Figure 4-15. Two tones one octave apart are fed into the structure, with one being 80dB larger in amplitude than the other. The curve labeled "original" shows the results without interferer cancelation: the smaller tone is invisible. The lower curve shows filter bank outputs after the large interfering tone has been estimated and notched out: the small tone is now clearly resolved.

Finally, the cascaded super-heterodyne structure shown in Figure 4-14 can be easily modified to use successive-subranging instead of successive-approximation. For

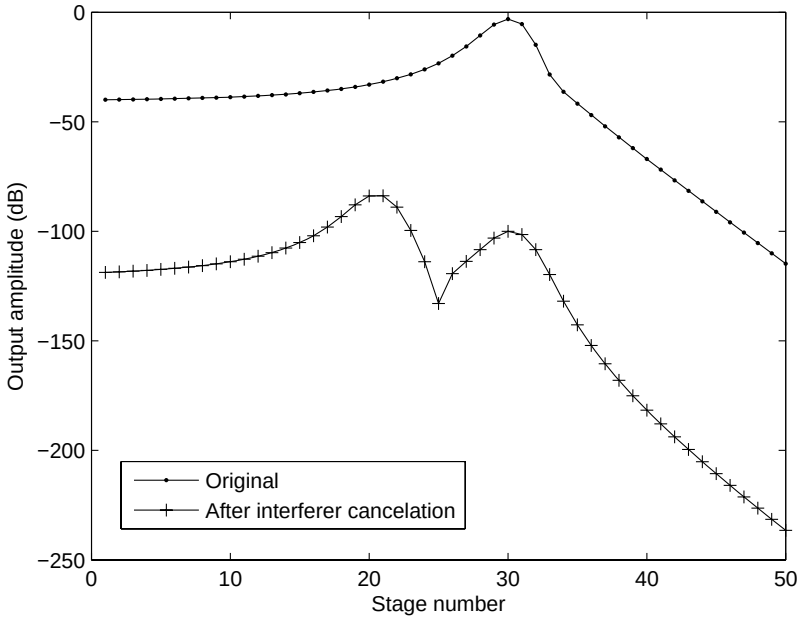


Figure 4-15: Simulated performance of the cascaded super-heterodyne architecture. The plots show outputs of the final filter bank in the presence of a large interfering signal. The interferer, which was 80dB larger than the signal of interest and an octave below it in frequency, was canceled out by using the procedure described in the text.

this purpose, frequency multipliers must be added at each filter bank output. Successive filter banks are then identical to each other (i.e., their frequency range and bandwidth remain constant, instead of decreasing by α every time). In fact, only a single filter bank can now be used, with the output residues being fed back in the manner similar to the cyclic or algorithmic structure shown in Figure 4-13(b).

4.4 Frequency Estimation with the RF Cochlea

4.4.1 Introduction

For the purposes of this section we define the *RF cochlea* as a system that performs real-time, broadband RF spectrum analysis by using either exponentially-tapered bidirectional transmission lines or unidirectional filter cascades. The transmission line(s) can be continuous or discrete/lumped, and by "exponentially-tapered" we mean that the characteristic frequency of the structure scales exponentially with

position.

4.4.2 Complexity of the Cochlear Algorithm

We now show that exponentially tapered traveling-wave structures like the cochlea are faster and more hardware-efficient than other spectral analysis techniques when the fractional bandwidth β (defined in (4.6)) is large. Intuitively, this is because the outputs of the cochlea have bandwidths that scale with center frequency. High-frequency stages have proportionally more bandwidth than low-frequency ones, reducing the total number of stages required to cover large frequency ranges than if the bandwidths had been kept constant. This type of behavior is known as a constant-Q response, and is also displayed by wavelet transforms. Here Q is defined as the bandwidth of each frequency bin normalized by the center frequency, i.e.,

$$Q_n = \frac{f_{n+1} - f_{n-1}}{2f_n} \approx \frac{1}{f_n}\frac{df_n}{dn} = \frac{d\ln(f_n)}{dn} \tag{4.8}$$

where f_n the center frequency of the n-th bin. Clearly, in order to keep Q_n constant, $\ln(f_n)$ must be proportional to n, i.e., $f_n \propto \exp(kn)$, where k is a constant. Thus constant-Q responses occur when the center frequencies of spectral analysis bins scale exponentially with position.

The performance of various commonly used spectrum analysis algorithms is compared in Table 4.2, where N is the number of outputs (frequency bins) obtained within the acquisition time, $\beta = f_H/f_L$ is the bandwidth ratio and $B = f_H - f_L$ is the total bandwidth. The hardware complexity of each algorithm is measured by the number of elementary computational units it consumes. A second-order analog filter is considered the elementary unit for the analog algorithms, while additions and multiplications fulfill a similar role for the FFT. We immediately see that the constant-Q (exponentially-spaced) algorithms have lower acquisition time and hardware complexity than similar constant-bandwidth algorithms (i.e., $(f_n - f_{n-1})$ =constant) when β is large. *Only the cochlea, however, combines low acquisition time with low hardware complexity.*

Table 4.2: Comparing various spectrum analysis algorithms

Name	**Acquisition time**	**Hardware complexity**
Swept-sine (linear sweep)	$O\left(\frac{N^2}{B}\right)$	$O(N)$
Swept-sine (exponential sweep)	$O\left(\frac{1}{f_L}\left(\frac{N}{\ln\beta}\right)^2\right)$	$O\left(\frac{N}{\ln\beta}\right)$
Analog filter bank (linear spacing)	$O\left(\frac{N}{B}\right)$	$O\left(N^2\right)$
Analog filter bank (exponential spacing)	$O\left(\frac{1}{f_L}\left(\frac{N}{\ln\beta}\right)\right)$	$O\left(\frac{N^2}{\ln\beta}\right)$
FFT	$O\left(\frac{N}{B}\right)$	$O(N\log(N))$
Cochlea	$O\left(\frac{1}{f_L}\left(\frac{N}{\ln\beta}\right)\right)$	$O(N)$

The cochlea consists of N stages, each producing one of the spectrally-analyzed, constant Q outputs. It is very efficient in its use of hardware because it uses a traveling-wave architecture. In traveling-wave structures the output signal from each stage is filtered by several other, nearby stages, which are simultaneously also producing their own outputs. In other words, the frequency selectivity of several stages is reused while synthesizing output transfer functions. This co-operative action stands in contrast to all the other algorithms listed in Table 4.2. In these other algorithms, each of the N stages producing outputs are functionally independent of each other. By filtering collaboratively, the cochlea reduces the filter order and complexity required by each of its stages. However, the overall cochlear transfer functions resemble high-order filter responses since they combine the filtering action of several adjacent cochlear stages. They thus exhibit high roll-off slopes and frequency selectivity. In summary, the use of collective computation enables the cochlea to reduce hardware complexity and power consumption without sacrificing performance. This is the reason why the cochlea is *not* well-represented as a bank of constant-Q bandpass filters that analyze the input in parallel. In fact, as shown in Table 4.2, the cochlea has an acquisition time that is similar to exponentially-spaced banks of analog filters, i.e.,

$O(N/(f_L \ln \beta))$. However, its hardware complexity is low ($O(N)$) and similar to that of swept-sine spectrum analyzers.

Figure 4-16 shows simulated outputs of each stage of an RF cochlea in response to an input signal consisting of multiple tones spaced one octave apart. Each peak in this "spatial response" plot corresponds to a single tone, illustrating how the cochlea performs spectral analysis. An added advantage of using constant-Q frequency separation bandwidths is that they are better matched to the real RF environment. The RF spectrum today is divided into licensed and unlicensed bands that roughly follow constant-Q characteristics. Bands at higher frequencies are wider than at lower frequencies. For example, the Q of the unlicensed industrial-scientific-military (ISM) frequency bands allocated by the FCC in the United States only vary by about a factor of ten as the center frequencies vary by over five orders of magnitude.

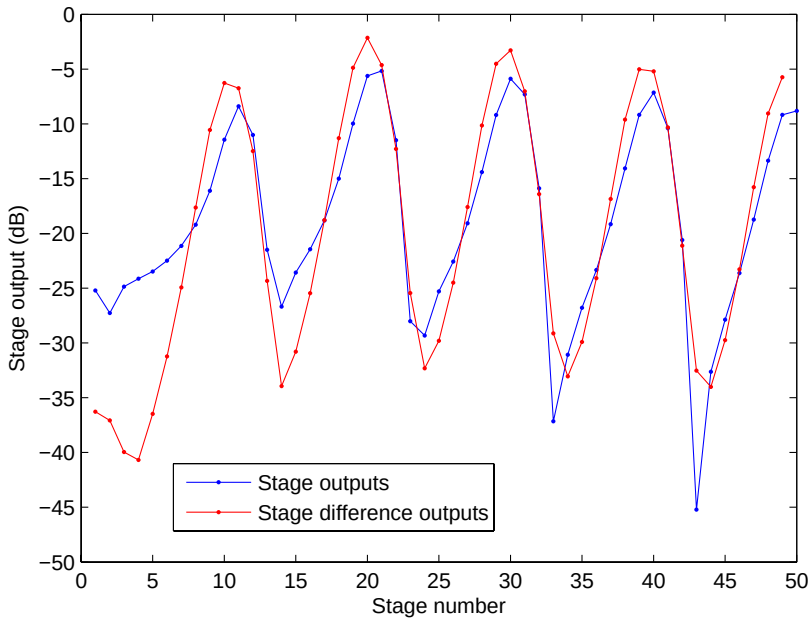


Figure 4-16: Simulated outputs (spatial response) of the unidirectional RF cochlea to multiple equal-amplitude input tones spaced one octave apart.

The RF cochlea is ideally suited for use as a front-end in an ultra-wideband radio receiver because of its extremely rapid frequency scan rate. Possible applications for such a receiver include cognitive radio. For example, the filter banks in Fig. 4-14 can

be replaced by cochleas. We call this example of the cascaded super-heterodyne architecture the *cascaded cochleas architecture*; it is shown in Figure 4-17. In this structure frequency estimation is performed hierarchically by cochlear models operating in different frequency ranges (for example, RF, IF and baseband), thus allowing multiple frequency bands to be monitored in parallel at any desired level of resolution. This structure is analogous to a successive-approximation ADC with many parallel outputs and provides a scalable, efficient way to implement flexible, broadband, "universal" radio receivers.

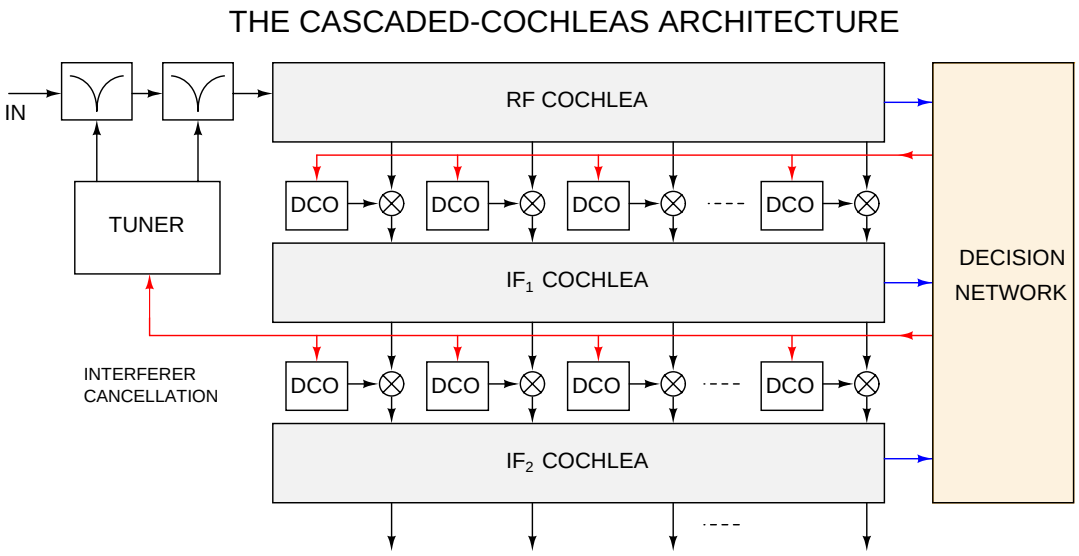


Figure 4-17: A successive-approximation architecture for hierarchically analyzing spectra to any level of precision by cascading cochlea-like structures at each level of the analysis.

The cascaded cochleas structure uses an adaptive successive-approximation approach to efficiently estimate sparse spectra. A typical value of the bandwidth-reduction factor between successive cochleas is $\alpha = 2$. Each cochlea shown in Figure 4-17 then has half the bandwidth of the previous one but is otherwise identical, thus providing double the frequency resolution. The outputs of the previous cochlea are fed into it in parallel after being downconverted using mixers. Each downconversion step selects half of the output bandwidth of the previous cochlear tap. As in Fig. 4-14, a decision network controls which half gets selected by changing the LO frequency f_{ij} of the mixer between $f_{ij,0}$ and $f_{ij,1}$, where $i > 0$ is the cochlea index and $1 < j < N$

is the tap index. A simple but effective decision rule is to use whichever value of f_{ij} gives the higher output amplitude at the same tap location on the next cochlea, i.e., at position $(i+1, j)$. This algorithm results in the structure adaptively "locking in" on features of interest in the spectrum at any level of frequency resolution. Once the algorithm has converged the bits at any tap location j contain information about frequencies present in the input. In fact, the whole process is a massively parallel, successively-approximating frequency-to-digital conversion, with the outputs of cochlea 1 providing the MSB's, those of cochlea 2 the next-most-significant bits and so on. It should also be noted that the structure retains amplitude information about each output signal and thus acts as a spectrum analyzer as well as a frequency estimator.

The total acquisition time of the cascaded cochleas structure is dominated by the final cochlea since it analyzes the lowest input frequencies. It scales as $O\left(N/f_{L,M}\right)$, where $f_{L,M} = f_{L,0}/\alpha^M$ is the lowest frequency analyzed by the final cochlea, there are a total of M cochleas and $f_{L,0}$ is the lowest frequency analyzed by the first (input) cochlea. The hardware complexity of the cascaded cochlea structure is modest: it scales as $O(N\log(M))$. Figure 4-18 shows simulated outputs produced by the cascaded cochleas structure at different levels of spectral resolution. The input contained five sinusoids with different frequencies and amplitudes. Resulting spatial responses are shown for $i = 1, 3$ and 5. As expected, each successive cochlea ("stage") increases spectral resolution. For example, the figure shows how two inputs that were too close to each other in frequency to be resolved after the first stage are clearly separated after the fifth stage.

Finally, the cascaded-cochleas structure shown in Figure 4-17 can be easily modified to use successive-subranging instead of successive-approximation. For this purpose, frequency multipliers must be added at each cochlear output. Successive cochleas are then identical to each other (i.e., their frequency range and bandwidth remain constant, instead of decreasing by α every time). In fact, only a single cochlea can now be used, with the output residues being fed back in the manner similar to the cyclic or algorithmic structure shown in Figure 4-13(b). The total acquisition time

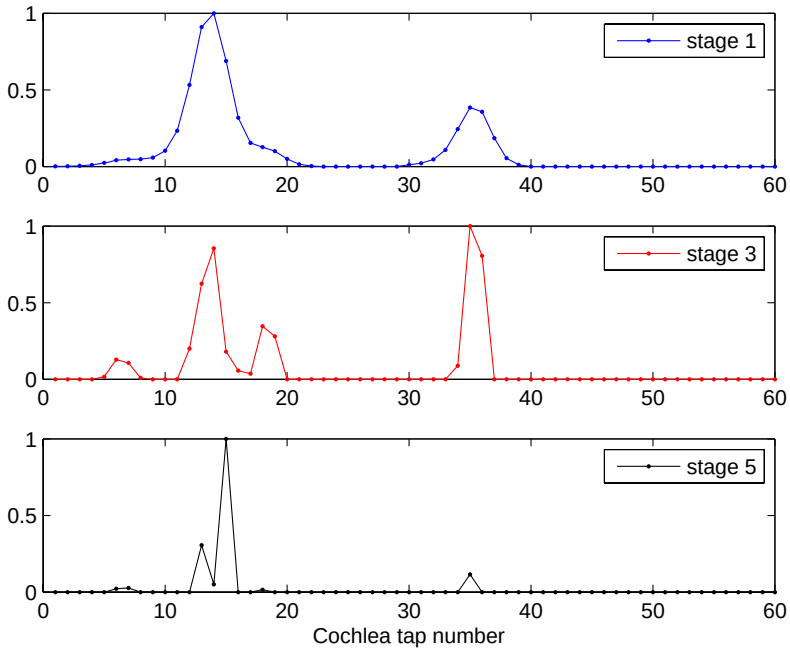


Figure 4-18: Simulated outputs (spatial response) of the cascaded cochleas structure to an input consisting of five sinusoids with different frequencies and amplitudes. Outputs after one, three and five cochleas are shown.

of this successive-subranging cascaded cochleas structure scales as $O\left(NM/f_{L,0}\right)$; it is thus α^M/M times faster than the successive-approximation version and has similar or lower hardware complexity.

Chapter 5

A Detailed Discussion of Noise

What stir is this? what tumult's in the heavens?
Whence cometh this alarum and the noise?
– William Shakespeare, King Henry VI, Part 1

The ultimate precision of any computation is limited by noise. This chapter therefore investigates noise mechanisms in detail. Some parts may strike readers as being rather philosophical; the principle of *caveat emptor* applies. The work described here was initiated by several long discussions the author had with Scott Arfin (who also helped with several of the figures).

5.1 Types of Noise

An important point of departure for our work was Sarpeshkar et al.'s claim that thermal noise is simply shot noise caused by diffusion in physical devices [257]. We agree with this statement, which is contrary to conventional wisdom in electrical engineering. However, we wish to explcitly state *why* shot noise is the more fundamental phenomenon. Essentially, shot noise is more fundamental because it simply involves counting random events and is completely independent of the details of the microscopic process involved, *as long as individual arrival times are uncorrelated with each other*. As a result, shot noise formulas can be applied to systems that are in steady

states far from thermodynamic equilibrium. On the other hand, thermal noise formulas only apply to systems that are actually at thermodynamic equilibrium. However, thermal equlibrium is usually an uninteresting, "dead" state, and most useful systems don't operate there.

In this document, we show that fluctuation-dissipation relationships can be used to derive well-known thermal noise formulas as special cases of shot noise formulas. We emphasize that in general these derivations are only valid at thermal equlibrium. However, shot noise can model counting uncertainty in cases where the system is far from thermodynamic equilibrium and/or the underlying random process is not thermal in origin. For example, the output of a device that measures rates of radioactive decay, such as a Geiger counter, will exhibit shot noise.

Unless stated otherwise, we will limit ourselves to frequencies much smaller than $f_c = kT/h$, where k is Boltzmann's constant, T is the absolute temperature and h is Planck's constant. This is not a major restriction, since, at room temperature (300K) $f_c = 6.25$THz, much higher than the operating limit of solid-state devices. This regime is also known as the Rayleigh-Jeans limit. If individual arrival times at the detector are uncorrelated, a shot noise process has a frequency-independent ("white") PSD given by

$$\overline{i_n^2} = 2qI \tag{5.1}$$

where I is the mean arrival rate of particles carrying charge q. Fundamentally, the factor of two arises because, while we normally plot the PSD for positive frequencies only, in reality the spectrum is double-sided and symmetric. The factor of two is required to account for the equal contribution of positive and negative frequency components to the observed variance of the time domain signal.

In a physical device a fraction of the current is carried by diffusion and the rest by drift. Diffusion currents have random arrival times and show shot noise. Drift current is the result of an electric field imposing a small drift velocity on top of a random thermal velocity distribution. Typical drift velocities are much smaller

than the mean thermal velocity ($\sqrt{8kT/\pi m}$, where m is the mass of the particle) at room temperature. Therefore drift adds a small constant perturbation to the random microscopic thermal motion that causes shot noise. In most cases (when the electric field and drift velocity is not too large), this perturbation can be ignored and we can safely say that drift currents cause no additional noise. This point can be confusing: drift can carry most of the current through a device, yet it causes no noise. This is because the behavior of the system is different on microscopic and macroscopic scales. Microscopically, there is large random thermal motion that is largely unaffected by the small amount of drift imposed by an external electric field. Macroscopically, this small drift current wins out because it always moves the carriers in the same direction (corresponding to the external field), while thermal motion, being random, on average does not move the carriers anywhere.

Some caveats on what follows. We shall simplify our lives throughout by considering ideal lumped devices with no parasitic capacitances present. This limits the validity of our results to relatively low frequencies. MOSFET gate noise shall also be ignored since it is typically negligible at these frequencies. We will always assume NMOS transistors unless mentioned otherwise. Our results can be easily extended to PMOS transistors by reversing the signs of the control voltages, diode polarities, etc.

5.2 The Resistor

In subsequent sections, we shall show that shot noise and thermal noise formulas are equivalent for a range of devices at thermodynamic equilibrium. Only in a resistor, however, does this remain (mostly) true when an external bias voltage is imposed, driving it away from equilibrium. The current noise PSD in a ideal resistor is always $4kT/R$, *independent of the bias voltage applied across it.* This is because an ideal resistor is essentially a featureless lump of resistive material. The only effect of the external bias voltage is to impose a small drift velocity component on top of the random thermal motion of the carriers inside it. Based on previous arguments, the drift current will cause essentially no additional noise, so the noise PSD remains

constant with bias. Another way to see this is to realize that, by definition, an ideal resistor is *linear*. Linearity means that the behavior of the device cannot depend on the bias point. Therefore the noise produced by it cannot change with bias. For nonlinear devices like diodes and transistors there is no reason to expect the thermal noise formulas to apply at arbitrary bias points far from equilibrium.

It is fairly easy to show that thermal and shot noise formulas agree in a resistor. Since the resistor is featureless, no region should contain more charge than the other. Assume that $Q(0)$ is the constant charge per unit length inside the resistor. This rectangular charge profile ($Q(x) = Q(0)$ inside the resistor from $x = 0$ to $x = L$, where L is the length of the resistor) can be split into two equal triangular pieces. These correspond to equal and uncorrelated diffusion currents flowing in each direction. The fact that they are equal is not surprising since the resistor is featureless and uniform and the carriers therefore have no preferred direction of motion. Each diffusion current causes $2qI$ of shot noise when it is detected at the terminals, where I is the diffusion current (*not* the total current, which includes drift) flowing through the resistor. Therefore the total current noise PSD is

$$\overline{i_n^2} = 2qI + 2qI = 4qI \tag{5.2}$$

Since I is a diffusion current, we can use the mathematical definition of diffusion to find out what it is:

$$I = D\frac{dQ}{dx} = \mu\phi_T\frac{dQ}{dx} \tag{5.3}$$

where we have used the Einstein relation $D = \mu\phi_T$, D is the diffusion constant, μ the carrier mobility and $\phi_T = kT/q$ is the thermal voltage[1]. For each triangular piece of charge $dQ/dx = Q(0)/L$, corresponding to a constant diffusion current. Therefore we may write

[1]The Einstein relation is an example of a fluctuation-dissipation theorem. It is only valid under the assumption that carrier drift is a small linear perturbation to the overall system. Not coincidentally, this is exactly the conditions under which drift causes no excess noise.

$$\frac{dQ}{dx} = \frac{Q(0)}{L} = \frac{ANq}{L} \tag{5.4}$$

where $Q(0) = ANq$ is the charge per unit length, N is the number of charge carriers per unit volume and A is the cross sectional area of the resistor. Substituting (5.3) and (5.4) into (5.2) and collecting some terms, we get

$$\overline{i_n^2} = 4kT\left(Nq\mu\right)\frac{A}{L} = 4kT\frac{1}{R} \tag{5.5}$$

where $R = A/\left(\sigma L\right)$ is indeed the resistance of the resistor, with $\sigma = Nq\mu$ being the bulk conductivity of the resistive material. Our derivation was independent of any DC bias voltage applied across the resistor. Therefore, as expected, the thermal and shot noise formulas always agree in this case.

An important caveat before we proceed further. We can only split up the charge in a given two-terminal device into forward and reverse diffusion components injected by the terminals (like we did here) if the charge profile $Q(x)$ is linear or constant. If $Q(x)$ is not linear, simple geometry shows that it is not possible to split up $Q(x)$ into two triangular parts, with one part being the forward diffusion current from terminal A and the other being the reverse diffusion current from terminal B. Therefore, a more complicated formulation is clearly necessary when $dQ(x)/dx$ is not constant.

5.3 The Junction Diode

A diode (see Figure 5-1) is a passive *nonlinear* resistor. In general, therefore, we expect its noise to be DC-bias-dependent. The I-V equation for a diode is given by

$$I = I_s\left(\exp\left(\frac{V}{n\phi_T}\right) - 1\right) \tag{5.6}$$

where I_s and n are constants, I is the current through the device and V is the voltage across the device. The incremental conductance g_d is defined as

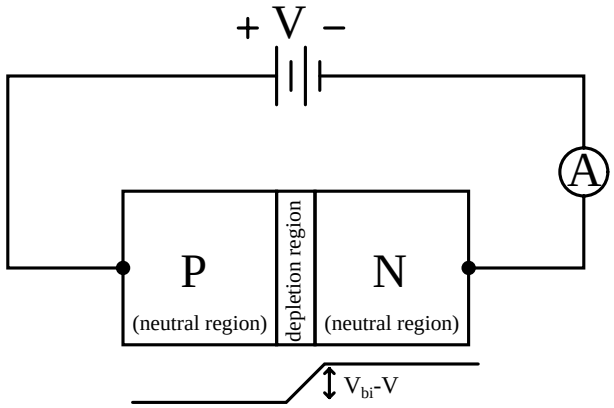


Figure 5-1: A junction diode with a bias voltage V across it. An ammeter A measures the noise current produced by the diode.

$$g_d = \frac{dI}{dV} = \frac{I + I_s}{n\phi_T} \tag{5.7}$$

Therefore, at zero bias the diode has a conductance $g_{d0} = I_s/(n\phi_T)$. Because it is passive, the I-V curve of any diode must pass through the origin. In other words, when there is no DC bias across the device, no DC current can flow through it. As a result, we expect a zero-biased diode to exhibit the same noise as a resistor with the same incremental resistance, i.e., with $R = 1/g_{d0}$ (this is an application of the fluctuation-dissipation theorem). Thus, the noise current power spectral density (A^2/Hz) is given by

$$\overline{i_{n0}^2} = 4kTg_{d0} = 4qI_s/n \tag{5.8}$$

This also looks a lot like the $2qI$ variance we expect from a shot noise process, with $I = 2I_s/n$. This is not a coincidence. Noise current flow through any physical device can be modeled as an electron-counting, i.e., shot noise process. Because the *net* current is zero at zero bias, equal noise currents (assumed to be uncorrelated) must flow in both directions, i.e., we may rewrite (5.8) as

$$\overline{i_{n0}^2} = \overline{i_{n0,+}^2} + \overline{i_{n0,-}^2} = 2qI_s/n + 2qI_s/n \tag{5.9}$$

Physically, at equilibrium the diffusion current flowing across the junction must be exactly balanced by drift current to ensure that the total current is zero. Diodes (and other devices, like transistors) also obey a principle of detailed balance that specifies that drift-diffusion balance be *independently* satisfied by both types of charge carriers, i.e., electrons and holes. In addition, remember that, except under extremely high field conditions, drift currents cause no noise. We may think of $\overline{i^2_{n0,+}}$, the total forward current, as being the sum of electron arrivals at the positive terminal and hole arrivals at the negative terminal. Similar reasoning applies to $\overline{i^2_{n0,+}}$, the total reverse current. These currents are equal at equilibrium because there is no preferred direction of current flow in the device.

The forward current flows from the majority to the minority side of the device *up* the potential gradient at the junction, and is called the diffusion current. The reverse current flows from the minority to the majority side of the device *down* the potential gradient at the junction, and is called the drift current. However, seemingly contrary to what we said earlier, *both cause noise*. It is true that the reverse current drifts with the electric field across the depletion region at the junction. However, it is produced by a random process (thermally induced carrier generation and recombination) and therefore has shot noise to begin with. The fact that it drifts across the junction does not remove this noise. Drift, consistent with what we said before, adds no noise of its own, but does nothing to remove the noise already present[2].

Now consider what happens when we put a bias voltage V across the device. This causes the barrier potential that limits the forward current to be lowered; it becomes $V_{bi} - V$, where V_{bi}, the built-in potential of the junction, is the height of the barrier at zero-bias. We now assume Boltzmann distributions for electron and hole density versus potential. Lowering the potential barrier causes the forward current (flowing from the p-side to the n-side for holes, and in the opposite direction for electrons) to become larger than the reverse current by the Boltzmann factor $\exp(V/\phi_T)$. The reverse current does not change since lowering the potential barrier has very little

[2]Think of drift as adding a large constant component (the drift velocity) on top of a random distribution of velocities and arrival times (the original noisy source).

effect on motion of carriers *down* the potential gradient at the junction (since they're still going downhill in energy, the barrier does not limit their flow rate). Thus the total noise current for non-zero bias voltage is given by

$$\overline{i_n^2} = (2qI_s/n)\left(\exp\left(\frac{V}{n\phi_T}\right) + 1\right) \tag{5.10}$$

By substituting (5.7) in the equation above we get the alternative form

$$\overline{i_n^2} = 2kTg_d\left(1 + \exp\left(\frac{-V}{n\phi_T}\right)\right) \tag{5.11}$$

This expression is also valid for metal-semiconductor (Schottky) diodes. The I-V characteristics of Schottky diodes are identical to p-n junction diodes when appropriate values of I_s and n are used. The noise generation mechanism is also similar.

5.4 The Subthreshold MOSFET

At moderate frequencies, the noise produced by a MOSFET has a flat frequency spectrum. Consider the subthreshold case first. The subthreshold MOSFET is most simply viewed as being two back-to-back p-n junction diodes with the gate-bulk voltage V_{GB} setting the common potential ϕ_s (see Figure 5-2). The diodes are associated with the source and drain junctions, respectively, while ϕ_s is known as the surface potential. The I-V characteristic of the device can be derived as follows. Assume that the source diode injects charge carriers into the channel. These electrons or holes diffuse across the channel and then get swept into the drain by the electric field of the built-in potential of the channel-drain junction diode. Call this current I_f. The drain diode similarly injects a current I_r. Since these currents flow in opposite directions the total device current is $I_D = I_f - I_r$. By using the I-V equation of a junction diode ((5.6), with $n = 1$), we get

$$I_D = I_f - I_r = I_s\left[\exp\left(\frac{\phi_s - V_S}{\phi_T}\right) - 1\right] - I_s\left[\exp\left(\frac{\phi_s - V_D}{\phi_T}\right) - 1\right] \tag{5.12}$$

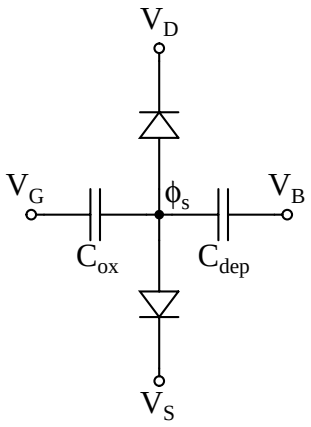


Figure 5-2: A simple model of a subthreshold MOS transistor.

where the same value of I_s has been used for both diodes since the device is symmetric (source and drain are physically indistinguishable) and all terminal voltages are measured with respect to the bulk (body) terminal B. The complicated part is now relating ϕ_s to the gate voltage V_G. The standard technique is to linearize the ϕ_s-V_G characteristic about some convenient operating point and call the slope κ, the subthreshold constant. That is, we assume a relationship of the form

$$\phi_s = \phi_{FB} + \kappa V_G \tag{5.13}$$

where ϕ_{FB} is a constant. By substituting (5.13) in (5.12) and collecting some terms, we get

$$I_D = I_0 \exp\left(\frac{\kappa V_G}{\phi_T}\right)\left[\exp\left(\frac{-V_S}{\phi_T}\right) - \exp\left(\frac{-V_D}{\phi_T}\right)\right] \tag{5.14}$$

where $I_0 = I_s \exp\left(\phi_{FB}/\phi_T\right)$. This is known as the bulk-referenced MOSFET I-V equation since all terminal voltages are referenced to the body (bulk,well) terminal. Most circuit designers, however, prefer to refer terminal voltages to the source. The terms in (5.14) can be rearranged to reflect this. The result is

$$I_D = I_0 \exp\left(\frac{\kappa V_{GS}}{\phi_T}\right) \exp\left(\frac{(1-\kappa)V_{BS}}{\phi_T}\right)\left[1 - \exp\left(\frac{-V_{DS}}{\phi_T}\right)\right] \tag{5.15}$$

This is known as the source-referenced I-V equation of the subthreshold MOSFET.

The incremental conductance g_{ds} of the device is defined as the slope of the I_D-V_{DS} curve when V_{GS} and V_{BS} are fixed. In this case (5.15) can be written in the simplified form

$$I_D = I_{D,sat}\left[1 - \exp\left(\frac{-V_{DS}}{\phi_T}\right)\right] \tag{5.16}$$

where the constant $I_{D,sat}$ absorbs all the other dependencies of I_D. The incremental conductance g_{ds} is given by

$$g_{ds} = \frac{\delta I_D}{\delta V_{DS}} = \left(\frac{I_{D,sat}}{\phi_T}\right)\exp\left(\frac{-V_{DS}}{\phi_T}\right) \tag{5.17}$$

First consider the situation at zero-bias, i.e., $V_{DS} = 0$. The conductance of the device is, from (5.17), $g_{d0} = I_{D,sat}/\phi_T$. The device is in thermodynamic equilibrium and indistinguishable from a resistor with $R = 1/g_{d0}$, so we expect it to produce a thermal noise PSD of

$$\overline{i_{n0}^2} = 4kTg_{d0} = 4qI_{D,sat} \tag{5.18}$$

This looks like the sum of two equal shot noise sources, each producing a noise current PSD equal to $2qI_{D,sat}$. This is exactly what we found for the junction diode at zero bias. Again, this is not a coincidence. It occurs because the source-channel and drain-channel diodes, which each produce shot noise according to (5.10), dominate the total white noise current of the device. Let us examine this in more detail. We recognize that the voltage V across the source diode is $\phi_s - V_S$. Therefore its noise is given by (5.10) with $V = (\phi_s - V_S)$ and $n = 1$ (usually a good assumption):

$$\begin{aligned}
\overline{i_{n,f}^2} &= 2qI_s\left(\exp\left(\frac{(\phi_s - V_S)}{\phi_T}\right) + 1\right) \\
&\approx 2qI_s\left(\exp\left(\frac{(\kappa V_G + \phi_{FB} - V_S)}{\phi_T}\right) + 1\right) \\
&= 2q\left(I_{D,sat} + I_s\right)
\end{aligned} \tag{5.19}$$

Similarly, the noise from the drain diode is given by

$$\begin{aligned}\overline{i_{n,r}^2} &= 2qI_s\left(\exp\left(\frac{(\phi_s - V_D)}{\phi_T}\right) + 1\right) \\ &= 2q\left(I_{D,sat}\exp\left(\frac{-V_{DS}}{\phi_T}\right) + I_s\right)\end{aligned} \tag{5.20}$$

Since the noise produced by the two diodes are uncorrelated, the total noise current produced by the device is

$$\overline{i_n^2} = \overline{i_{n,f}^2} + \overline{i_{n,r}^2} = 2q\left[I_{D,sat}\left(1 + \exp\left(\frac{-V_{DS}}{\phi_T}\right)\right) + 2I_s\right] \tag{5.21}$$

At zero-bias ($V_{DS} = 0$), no DC current flows through the device but noise is present; (5.21) becomes

$$\overline{i_{n0}^2} = 4q\left(I_{D,sat} + I_s\right) \tag{5.22}$$

We can now compare (5.22) with (5.18), our original "intuitive" expression for zero-biased current noise in the subthreshold MOSFET. The two expressions are identical except for the extra I_s term in (5.22). Where did that come from? It's presence indicates that our earlier assumption of thermodynamic equilibrium at zero-bias was incorrect. In fact, (5.19) and (5.20) show that the two diodes in the MOSFET are *not* at thermodynamic equilibrium: the forward and reverse currents inside them are not equal. This is because the gate, which is driven by some external voltage supply, can force ϕ_s not to be equal to V_S and V_D, even when $V_{DS} = 0$. In practice, the I_s term can usually be ignored since it is much smaller than practical values of $I_{D,sat}$.

Another possible source of noise in the MOSFET is the channel. In subthreshold, however, there are so few charge carriers in the channel that it is usually safe to assume that their own electric field is much smaller than that imposed externally by the gate. The gate therefore keeps the whole channel, from source to drain, equipotential at ϕ_s. Therefore the effective channel resistance is zero: it carries a DC current I_D but there

is no voltage drop across it. Therefore, to first order, the channel causes no noise.

5.5 The Above-Threshold MOSFET

We now consider the strongly inverted or above-threshold MOSFET. Long-channel (square-law) behavior shall be assumed. The current-conduction mechanism in the channel is more complicated than below threshold since both drift and diffusion are involved. As the gate voltage V_G increases, the surface potential ϕ_s also increases (approximately following (5.13)). So many carriers are injected by the source and drain diodes into the channel that, eventually, the electric field produced by these carriers themselves becomes comparable to that imposed externally by the gate. At this point the gate ceases to have sole control of the ϕ_s. The charge carriers also begin to influence ϕ_s. As a result the gate can no longer force ϕ_s to remain constant along the length of the channel, and drift begins to contribute to total current flow. By using the mathematical definitions of drift and diffusion, we can write the total current in the channel as

$$I_D = W\left(-\mu Q_I \frac{d\phi_s}{dx} + D\frac{dQ_I}{dx}\right) \tag{5.23}$$

where the first term is the drift current and the second term the diffusion current. The width of the current conduction path is denoted by W. The carrier mobility and diffusion constant in the channel are denoted by μ and D, respectively. Finally, Q_I, known as the inversion charge density, is the total carrier charge density (coulombs/unit area) in the channel. The tricky part is finding ϕ_s (just like in the subthreshold device). The simplest approximation is to say that Q_I is the sum of the charges on the bottom plates of two capacitors. One is formed by the oxide capacitance C_{ox} (to the gate) and the other by the depletion capacitance C_{dep} (to the bulk). The voltage on the plate is ϕ_s. Assuming V_G and V_B are fixed and remembering the fundamental relationship $dQ/dV = C$, we may write

$$\frac{dQ_I}{d\phi_s} = C_{ox} + C_{dep} = \frac{C_{ox}}{\kappa} \tag{5.24}$$

where $\kappa = C_{ox}/\left(C_{ox} + C_{dep}\right)$ and is approximately constant. This definition of κ is also equivalent to that in (5.13), which is comforting. We can now use the chain rule to write

$$\frac{d\phi_s}{dx} = \frac{d\phi_s}{dQ_I}\frac{dQ_I}{dx} = \frac{\kappa}{C_{ox}}\frac{dQ_I}{dx} \tag{5.25}$$

By using the Einstein relation $D/\mu = \phi_T$ and substituting (5.25) in (5.23), we get

$$I_D = \frac{\kappa\mu W}{C_{ox}}\left(-Q_I + \frac{\phi_T C_{ox}}{\kappa}\right)\frac{dQ_I}{dx} \tag{5.26}$$

where x varies from 0 to L along the channel (from source to drain). By recognizing that, because of KCL, I_D must be independent of x, this equation can be easily integrated along the channel to give

$$I_D = \frac{\kappa\mu}{C_{ox}}\frac{W}{L}\left[-\frac{1}{2}\left(Q_I^2(L) - Q_I^2(0)\right) + \frac{\phi_T C_{ox}}{\kappa}\left(Q_I(L) - Q_I(0)\right)\right] \tag{5.27}$$

where $Q_I(0)$ and $Q_I(L)$ are the values of Q_I at the source ($x = 0$) and drain ($x = L$) ends of the channel, respectively. Again, the first term in the equation above is caused by drift and the second by diffusion. Each of them can be split into a forward and reverse current term, so that (5.28) can be rewritten as

$$I_D = I_f - I_r = \mu\frac{W}{L}\left(\frac{\kappa}{2C_{ox}}Q_I^2(0) - \phi_T Q_I(0)\right) - \mu\frac{W}{L}\left(\frac{\kappa}{2C_{ox}}Q_I^2(L) - \phi_T Q_I(L)\right) \tag{5.28}$$

Now what do we use for $Q_I(0)$ and $Q_I(L)$? The simplest approximation is a linear one. Assume that $Q_I = 0$ is zero when $\phi_s = \phi_{sa}$ and follows (5.24) (with constant κ) thereafter. Thus we may integrate (5.24) to write

$$Q_I(x) = \frac{C_{ox}}{\kappa}\left(\phi_s(x) - \phi_{sa}\right) \tag{5.29}$$

However, what is $\phi_s - \phi_{sa}$? This is the *really* confusing part. It seems reasonable to conclude that the point at which Q_I becomes very small (approximately zero) is the transition to subthreshold operation. In subthreshold, we know that ϕ_s is independent of x and is only set by the gate, following (5.13). Therefore, we may write

$$\phi_{sa} = \kappa V_G + \phi_{FB} \tag{5.30}$$

Above threshold, the surface potential $\phi_s(0)$ close to the source terminal is set primarily by the source, *not* the gate. One way to look at it is that the source-channel diode turns on and effectively clamps $\phi_s(0)$ to one diode drop above V_s. Let's assume that, to first order, the clamp is perfect and the diode drops a fixed voltage ϕ_0 across it[3]

$$\phi_s(0) = V_S + \phi_0 \tag{5.31}$$

Similarly, the surface potential $\phi_s(L)$ close to the drain terminal is set primarily by the drain voltage V_D in strong inversion, so we can write

$$\phi_s(L) = V_D + \phi_0 \tag{5.32}$$

As V_S and V_D are increased, therefore, $\phi_s(0)$ and $\phi_s(L)$ follow. However, this only continues up to a point. Eventually the terminal (source or drain) voltage gets high enough that the terminal-channel diode turns off (becomes reverse-biased). In other words, the current through the diode gets smaller and smaller because the electrons would rather stay in the terminal (it is energetically favorable for them to do so, because of the high terminal voltage) rather than go out into the channel. At this point the diode clamp approximation breaks down and the terminal can no longer

[3]We should be careful about our terminology here. The terminal-channel diodes above threshold don't really behave as exponential elements. They are in a high-injection regime: so many electrons are diffusing through them that electrons become the majority carriers in the normally p-type channel (i.e., the channel inverts, making $Q_I > 0$). Since both sides of the diode now contain mostly electrons, no real pn junction remains. This explains why the diodes look almost like batteries (fixed, low-impedance voltage drops) above threshold.

exert control over the surface potential or channel charge close to it. Control over ϕ_s reverts to the gate, which sets it using (5.30). This process happens sooner at the drain than the source, since by definition $V_D > V_S$.

A transistor where $\phi_s(0)$ is controlled by the source but $\phi_s(L)$ by the gate (not the drain) is *saturated*. The drain can no longer effectively control current flow through the device and becomes a high-impedance terminal. This is the normal mode of transistor operation in analog circuits, because the high-impedance drain terminal can be used to create signal gain.

We can now substitute (5.30) and (5.31) or (5.32) in (5.29) to get the inversion charge densities at the source and drain ends of the channel:

$$\begin{aligned} Q_I(0) &= \frac{-C_{ox}}{\kappa}\left(\kappa V_G + \phi_{FB} - \phi_0 - V_S\right) \\ Q_I(L) &= \frac{-C_{ox}}{\kappa}\left(\kappa V_G + \phi_{FB} - \phi_0 - V_D\right) \end{aligned} \tag{5.33}$$

This may be superficially simplified by making the traditional substitution $V_{T0} \equiv \left(\phi_0 - \phi_{FB}\right)/\kappa$, where V_{T0} is known as the threshold voltage of the transistor. The result is

$$\begin{aligned} Q_I(0) &= \frac{-C_{ox}}{\kappa}\left(\kappa\left(V_G - V_{T0}\right) - V_S\right) \\ Q_I(L) &= \frac{-C_{ox}}{\kappa}\left(\kappa\left(V_G - V_{T0}\right) - V_D\right) \end{aligned} \tag{5.34}$$

We can also estimate $V_{D,sat}$, the value of V_D at which the transistor enters saturation. A reasonable approximation is that saturation happens when $\phi_s(L)$ predicted by (5.32) becomes equal to the subthreshold value ϕ_{sa} (thereby making $Q_I(L) \approx 0$), i.e.,

$$\phi_s = \phi_{sa} \Rightarrow V_{D,sat} + \phi_0 = \kappa V_G + \phi_{FB} \Rightarrow V_{D,sat} = \kappa\left(V_G - V_{T0}\right) \tag{5.35}$$

The inversion charge expressions above can be substituted in (5.28) to give us (after a little algebra) the above-threshold MOSFET I-V equations

$$I_D = \mu\kappa\frac{W}{2L}C_{ox}\left[\left(V_G - V_{T0} - \frac{V_S}{\kappa}\right)^2 - \left(V_G - V_{T0} - \frac{V_D}{\kappa}\right)^2 + \frac{2\phi_T}{\kappa^2}\left(V_D - V_S\right)\right] \tag{5.36}$$

when $V_D < \kappa\left(V_G - V_{T0}\right)$ and

$$I_D = \mu\kappa\frac{W}{2L}C_{ox}\left[\left(V_G - V_{T0} - \frac{V_S}{\kappa}\right)^2 + \frac{2\phi_T}{\kappa}\left(V_G - V_{T0} - \frac{V_S}{\kappa}\right)\right] \tag{5.37}$$

when $V_D \geq \kappa\left(V_G - V_{T0}\right)$, i.e., in saturation when $Q_I(L) \approx 0$. These are body-referenced equations since all terminal voltages are referred to the body (bulk) terminal. Traditionally, the diffusion current terms (the ones proportional to ϕ_T) were neglected because it was assumed that the gate overdrive voltage was much larger than the thermal voltage, i.e. $\left(V_G - V_{T0} - V_S/\kappa\right) \gg \phi_T$. It may be verified that the ratio of the drift and diffusion current terms in (5.36) and (5.37) is equal to the ratio of these two voltages. If one makes this approximation, (5.36) and (5.37) simplify to

$$\begin{aligned} I_D &= \mu\kappa\frac{W}{2L}C_{ox}\left[\left(V_G - V_{T0} - \frac{V_S}{\kappa}\right)^2 - \left(V_G - V_{T0} - \frac{V_D}{\kappa}\right)^2\right] \\ I_D &= \mu\kappa\frac{W}{2L}C_{ox}\left(V_G - V_{T0} - \frac{V_S}{\kappa}\right)^2 \end{aligned} \tag{5.38}$$

where the first equation applies if $V_D < \kappa\left(V_G - V_{T0}\right)$, and the second equation otherwise. Just like in the subthreshold case, we can convert these equations to the source-referenced form to keep circuit designers happy. After a little algebra, we find

$$
\begin{aligned}
I_D &= \mu\kappa\frac{W}{2L}C_{ox}\left[2\left(V_{GS}-V_{TS}\right)\left(\frac{V_{DS}}{\kappa}\right)-\left(\frac{V_{DS}}{\kappa}\right)^2\right] \\
I_D &= \mu\kappa\frac{W}{2L}C_{ox}\left(V_{GS}-V_{TS}\right)^2
\end{aligned} \tag{5.39}
$$

where the first equation applies if $V_{DS} < \kappa\left(V_{GS}-V_{TS}\right) \equiv V_{DS,sat}$, and the second equation otherwise. Notice that we have also sneakily redefined the threshold voltage to be

$$
V_{TS} \equiv V_{T0} - \left(\frac{1-\kappa}{\kappa}\right)V_{BS} \tag{5.40}
$$

This change of threshold voltage with V_{BS} is known as the body effect. The conductance of the device is ideally zero in saturation ($V_{DS} > \kappa\left(V_{GS}-V_{TS}\right)$), and is otherwise given by

$$
g_{ds} = \frac{\delta I_D}{\delta V_{DS}} = \mu\frac{W}{L}C_{ox}\left(V_{GS}-V_{TS}-\frac{\left(V_{DS}-2\phi_T\right)}{\kappa}\right) \tag{5.41}
$$

where, for completeness, we have reintroduced the term proportional to ϕ_T that we ignored earlier. We can now (finally!) estimate the noise produced by the above threshold MOSFET. Unlike in the subthreshold MOSFET, the dominant source of noise above threshold is the channel. The channel resistance is no longer zero above threshold: ϕ_s varies with position along the channel when current flows through the device. Since the channel looks like a resistor, it should produce shot noise. We have to be careful while calculating it, though. Only part of the channel current is due to diffusion: most of it, in fact, is due to drift and is therefore noiseless. We start with the zero-bias condition, when $V_{DS} = 0$. At zero-bias $\phi_s(0) = \phi_s(L)$ and therefore there is no drift current. This simplifies matters considerably, because all the device conductance is due to diffusion currents. Therefore we can view the channel simply as a resistor with $R = 1/g_{d0}$, where g_{d0} is the zero-bias conductance, obtained by substituting $V_{DS} = 0$ in (5.41). The noise PSD produced by the channel is then

$$\overline{i^2_{n,channel}} = 4kTg_{d0} = 4kT\left[\mu\frac{W}{L}C_{ox}\left(V_{GS} - V_{TS} + \frac{2\phi_T}{\kappa}\right)\right] \tag{5.42}$$

We can generalize this expression as follows. Since only the diffusive part of the total channel current causes noise, we can write

$$\overline{i^2_{n,channel}} = 2q\left(I_{diffusion,+} + I_{diffusion,-}\right) \tag{5.43}$$

where $I_{diffusion,+}$ and $I_{diffusion,-}$ are the diffusion currents collected at the drain and source, respectively. We also know that, in general $I_{diffusion}$ is given by

$$I_{diffusion} = WD\frac{dQ_I}{dx} = \mu W\phi_T\frac{dQ_I}{dx} \tag{5.44}$$

where we have used the Einstein relation $D/\mu = \phi_T$ for charge carriers. The problem is now to find $Q_I(x)$. We shall (as always), make the simplest possible approximation: let's assume $Q_I(x)$ is linear, varying from $Q_I(0)$ at the source to $Q_I(L)$ at the drain. This approximation is actually not too bad when the electric field $d\phi_s(x)/dx$ in the channel is relatively small (i.e., for long channel devices). We now make the following assumptions:

$$\begin{aligned} I_{diffusion,+} &= WD\left(\frac{dQ_I}{dx}\right)_+ \approx \mu W\phi_T\left(\frac{Q_I(0)}{L}\right) \\ I_{diffusion,-} &= WD\left(\frac{dQ_I}{dx}\right)_- \approx \mu W\phi_T\left(\frac{Q_I(L)}{L}\right) \end{aligned} \tag{5.45}$$

These assumptions may be justified as being the results of dividing the trapezoidal $Q_I(x)$ profile into two triangular pieces, the larger controlled by the source and the smaller by the drain (see Figure 5-3). Each triangular piece has a constant value of $dQ_I(x)/dx$, and therefore produces a constant diffusion current. The source triangle produces $I_{diffusion,+}$, and the drain triangle, $I_{diffusion,-}$. The two currents are independent and therefore we can substitute (5.45) into (5.43) to get the total noise PSD from the channel:

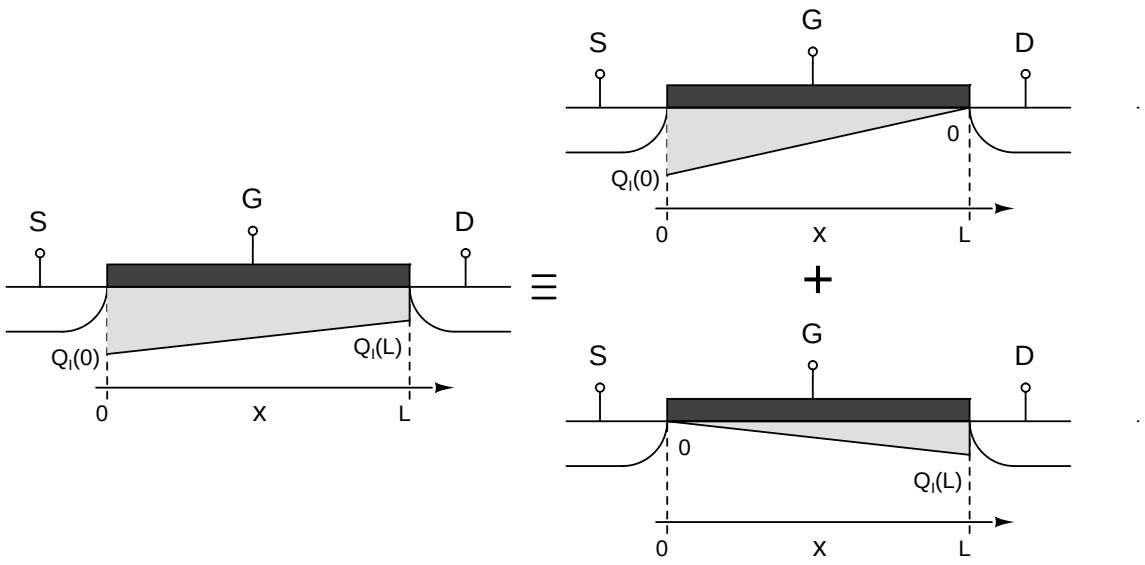


Figure 5-3: Calculating the noise of an above-threshold MOSFET by splitting the inversion charge in the channel into individual diffusion currents controlled by the source and drain.

$$\overline{i^2_{n,channel}} = 2q\mu\phi_T \frac{W}{L}\left[Q_I(0) + Q_I(L)\right] \tag{5.46}$$

After substituting the inversion charge expressions from (5.34) into the equation above, converting them to source-referenced form and doing a little algebra we get

$$\begin{aligned} \overline{i^2_{n,channel}} &= 4kT\left(\mu\frac{W}{L}C_{ox}\right)\left(V_{GS} - V_{TS} - \frac{V_{DS}}{2\kappa}\right) \\ &\approx 4kTg_{d0}\left(\frac{1+\eta}{2}\right) \end{aligned} \tag{5.47}$$

where we have assumed that $(V_{GS} - V_{TS}) \gg 2\phi_T/\kappa$ while substituting for g_{d0}, and $\eta \equiv (1 - V_{DS}/V_{DS,sat})$ parameterizes the degree of saturation of the device. When $V_{DS} = 0$, $\eta = 1$ and the device is completely linear (unsaturated). Equation (5.47) is only valid when $V_{DS} \leq V_{DS,sat}$, i.e., $\eta \geq 0$, since we have assumed that $Q_I(L) > 0$, i.e., that the device is unsaturated. Equation (5.47) predicts that the noise decreases by a factor of two as η goes from 1 to 0, i.e., the device becomes saturated.

As the transistor approaches saturation, $Q_I(L)$ becomes zero but an additional

complication is introduced by the fact that the effective channel length decreases; the channel near the drain is weakly inverted (subthreshold) and $Q_I \approx 0$ there. This increases the slope of $I_{diffusion,+}$ from $Q_I(0)/L$ to $Q_I(O)/L_{eff}$, where $L_{eff} < L$ is the effective channel length of the device. So we get more noise then expected. Once saturation is reached, $Q_I(L) = 0$ and we may write

$$\overline{i^2_{n,channel}} = 2q\mu\phi_T \frac{W}{L_{eff}} Q_I(0) \tag{5.48}$$

We can substitute for $Q_I(0)$ to get

$$\overline{i^2_{n,channel}} = 4kT \left(\mu \frac{W}{L_{eff}} C_{ox} \right) (V_{GS} - V_{TS}) = 4kT\gamma g_{d0} \tag{5.49}$$

where γ is a fudge constant that is trying to hide our ignorance. The problem is that we don't know what L_{eff} to use. In other words, since $Q_I(x)$ is no longer linear, our forward-reverse diffusion current decomposition no longer works. The decomposition isn't self-consistent since $I_{diffusion} \propto dQ_I(x)/dx$ now varies from point to point. Therefore we can't really use a constant diffusion current injected from the source in our formulas and expect that to work.

It turns out that $\gamma = 2/3$ fits experimental data for long-channel MOSFETs (and can be justified theoretically by using more sophisticated models). This corresponds to $L_{eff} = 3L/4$, which is not unreasonable. To summarize, γ decreases (monotonically, as it turns out) from 1 to 2/3 as a long-channel MOSFET moves from $V_{DS} = 0$ to saturation. Equation (5.47) correctly predicts the behavior of γ when V_{DS} is nonzero but small: $\gamma = (1+\eta)/2$. Finally, short-channel MOSFETs have additional noise mechanisms that can substantially increase γ above long-channel values.

So far we have considered only noise produced by the channel. The source-channel and drain-channel diodes also produce noise. However, remember that we assumed that the diodes acted like clamps above threshold, dropping a fixed voltage ϕ_0 across them. As a result, they look like shorts for small signals (such as noise). In other words, the effective impedance of the diodes is much smaller then the channel. Therefore their noise contribution can be ignored.

There is one added complication, however. The argument above is always valid for the source diode, but is only valid for the drain diode when the device is not saturated. In saturation, the drain diode is reverse-biased; the drop across it is set by the gate-drain voltage and is not constant. However, because it is reverse biased, current flows through it via drift, which, ideally, adds no noise. Therefore, the noise contribution of the drain diode can also, to first order, be ignored[4]. In summary, therefore, the total white noise produced by a long-channel transistor can be assumed to be approximately equal to $\overline{i^2_{n,channel}}$.

5.6 Noise Calculations based on Transit Time

Since the random arrival times of charge carriers leads to shot noise, it makes sense to focus on diffusion, because it is inherently a random process. Consider an experiment where the current produced by a physical device is being measured by an ammeter. An electron starts at one end of the device at some known time, is transported through it by drift, diffusion or both and then arrives at the ammeter, where it is detected. The electrical circuit must be complete (KCL), so the electron eventually returns to its starting position to repeat the process. The total time τ taken for this loop to be completed is known as the *transit time* and is a random variable. We shall focus on the case when the main contributor to τ is diffusion through the device. This is an excellent assumption as long as drift is much faster than diffusion. We shall also assume that the diffusion constants of the wires (comprising the rest of the circuit) are much larger than that of the device. This is usually the case. The mean value of the transit time is

$$\overline{\tau} = \frac{L^2}{2D} \tag{5.50}$$

where L is the length of the device in the direction of current flow, and D is

[4]This turns out to be a bad assumption for short channel devices, where the strong electric field near the channel-drain junction causes the "drift is noiseless" approximation to break down. This is one reason why γ typically increases as the channel is made shorter.

a frequency and temperature-dependent quantity misleadingly called the diffusion *constant*. Our assumption that drift is much faster than diffusion means that

$$\frac{L}{v_{drift}} \ll \frac{L^2}{2D} \Rightarrow L \gg \frac{2D}{v_{drift}} \tag{5.51}$$

where v_{drift} is the drift velocity in the device. If L is small enough ($\lessapprox$ 25 nm for Silicon devices), (5.51) is not true and the following derivation is not applicable. Equation (5.50) is easy to derive. Consider a typical particle inside the device. At $t = 0$ it is moving in a given direction with an average velocity v_{th}. It travels an average distance l_{mfp} (the mean free path) before it suffers a collision with another particle. After the collision, it moves in a different direction with a different speed, neither of which can be predicted. In effect, collisions randomize velocities and directions of travel. We want to find out how long it takes an average particle to move a distance L along a *known* direction, say x. However, v_x, the component of the particle's velocity along x is a random number that varies with time, because the direction the particle is traveling in is constantly changing. The average distance traveled after N collisions, $\overline{x(N)}$, is the sum of these random numbers and is zero. This makes sense since, in the absence of an external force, the particle will not on average, will not go anywhere. The standard deviation $\sqrt{\overline{x^2(N)}}$ of the particle's position after N collisions, is however non-zero and increases with time. If N is large $\sqrt{\overline{x^2(N)}}$ becomes Gaussian (by the central limit theorem). The variance in position due to a single collision is about l_{mfp}^2. Each collision causes more variance. Since we are summing up random numbers, the variances add, therefore

$$\overline{x^2(N)} = l_{mfp}^2 N \Rightarrow \sqrt{\overline{x^2(N)}} = l_{mfp}\sqrt{N} \tag{5.52}$$

In order to get $\sqrt{\overline{x^2(N)}} = L$, we therefore need $N = L^2/l_{mfp}^2$. The average time between collisions is $\tau_{coll} = l_{mfp}/v_{th}$. Therefore the average time taken to "move" a distance L (i.e., to make the standard deviation in the particle's position be equal to L) is

$$\overline{\tau} = N\tau_{coll} = \frac{L^2}{l_{mfp}v_{th}} \equiv \frac{L^2}{2D} \tag{5.53}$$

where $D = l_{mfp}v_{th}/2$ is the diffusion constant in one dimension and has units of cm^2/sec. Diffusion is an example of a transport process. In a transport process some conserved quantity is moved around by a potential gradient. In diffusion, the number of particles is conserved. Other examples include heat conduction in solids and viscosity in fluid flow (the conserved quantities are energy and momentum, respectively). The speed of transport processes are characterized by transport constants (D is an example). The characteristic form of a transport constant is λv, where λ is the characteristic length scale (l_{mfp} in this case) and v is the characteristic velocity (v_{th} in this case).

Since each electron takes an average time to complete the circuit loop, it's arrival rate at the detector is $1/\overline{\tau}$, corresponding to an average current of $q/\overline{\tau}$. The total average current at the detector is therefore $I_{av} = qN_{tot}/\overline{\tau}$, where N_{tot} is the total number of electrons flowing in the circuit. We now make the approximation that at any given time almost all the N_{tot} electrons are to be found in the device (and not in the wires or the detector). This follows from our earlier assumption that the total transit time is dominated by the device. On average, therefore any electron spends most of its time inside the device. Therefore we may write $Q_{tot} = qN_{tot}$, where Q_{tot} is the (time-averaged) total charge contained in the device. We also assume that each electron diffuses independently. Since their arrival times at the detector are then uncorrelated, we can use simple shot noise formula (5.1) and write the current noise PSD of the device as

$$\overline{i_n^2} = 2qI_{av} = 2q\left(\frac{|Q_{tot}|}{\overline{\tau}}\right) = 4qD\left(\frac{|Q_{tot}|}{L^2}\right) \tag{5.54}$$

where the modulus around Q_{tot} represents the fact that we don't care about the direction of the noise current, but only its magnitude. Note that $I_{av} \neq I_{DC}$, the average current through the device, because in general some of this current will be carried by drift, which is assumed to be noiseless. Let's now see if this formula works

in the simplest possible case, i.e., for a resistor. For a resistor $Q_{tot} = qN_{tot} = qNAL$, where N is the density of charge carriers (electrons) inside it, A is its cross-sectional area and L is its length. Therefore, from (5.54) we have

$$\overline{i_n^2} = 4qD\left(\frac{NAL}{L^2}\right) = 4kT\left(\frac{(Nq\mu)\,A}{L}\right) = 4kT\left(\frac{\sigma A}{L}\right) \equiv 4kTG \tag{5.55}$$

where $\sigma = Nq\mu$ is the conductivity of the resistive material, $G \equiv \sigma A/L$ is the total conductance of the resistor and we have used Einstein's relation $D = \mu\phi_T$. Therefore (5.54) agrees with the thermal noise formula for a resistor. The reader may question whether anything has been gained, since we've already proven this result. The great advantage of our current technique, however, is that it does not force us to split Q_{tot} into two linear charge profiles to get constant, oppositely directed drift currents inside the device. As we have mentioned before, this split cannot be made if the charge profile is not linear, such as in an above-threshold MOSFET. Let us therefore now use our new technique to find the noise of the above-threshold MOSFET in a more rigorous way.

First we need to find the channel charge density profile $Q_I(x)$. Remember that the total device current I_D must be independent of x (KCL). Now consider splitting the transistor into two shorter transistors at some location x along the channel. Each shorter transistor, with lengths x and $L - x$ respectively, should still carry I_D. It is easiest to see this if one considers integrating (5.26) from 0 to x instead of from 0 to L. The left-hand side must remain constant at I_D in either case. Therefore the right-hand sides must be equal as well, guaranteeing that

$$\frac{-\left(Q_I^2(L) - Q_I^2(0)\right) + \frac{2\phi_T C_{ox}}{\kappa}\left(Q_I(L) - Q_I(0)\right)}{2L} = \frac{-\left(Q_I^2(x) - Q_I^2(0)\right) + \frac{2\phi_T C_{ox}}{\kappa}\left(Q_I(x) - Q_I(0)\right)}{2x} \tag{5.56}$$

We can immediately solve the quadratic equation above to determine $Q_I(x)$. However, the final result is too complicated for hand calculations or physical insight. We shall instead make the approximation that all the current is carried by drift. This is a reasonable approximation when $(V_{GS} - V_{TS}) \gg \phi_T$. In this situation

$Q_I^2(x) \gg (2C_{ox}\phi_T/\kappa) \times Q_I(x)$ for any $x \in [0, L]$ and (5.56) simplifies to[5]

$$\frac{Q_I^2(L) - Q_I^2(0)}{2L} \approx \frac{Q_I^2(x) - Q_I^2(0)}{2x} \tag{5.57}$$

This can be easily solved to give

$$Q_I(x) = \sqrt{Q_I^2(0)\left(1 - \frac{x}{L}\right) + Q_I^2(L)\left(\frac{x}{L}\right)} \tag{5.58}$$

Remembering that $Q_I(x)$ is the charge per unit area of the channel, the total channel charge is given by

$$Q_{tot} = W\int_0^L Q_I(x)dx = \frac{2}{3}WLQ_I(0)\left[\frac{Q_I^3(L) - Q_I^3(0)}{Q_I^2(L) - Q_I^2(0)}\right] \tag{5.59}$$

We now substitute for $Q_I(0) = C_{ox}\left(V_{GS} - V_{TS}\right)$ using the source-referenced model, and also define

$$\eta \equiv \frac{Q_I(L)}{Q_I(0)} = \frac{C_{ox}\left(V_{GS} - V_{TS} - \frac{V_{DS}}{\kappa}\right)}{C_{ox}\left(V_{GS} - V_{TS}\right)} = 1 - \frac{V_{DS}}{V_{DS,sat}} \tag{5.60}$$

where $V_{DS,sat} = \kappa\left(V_{GS} - V_{TS}\right)$ is the saturation voltage. The result is

$$Q_{tot} = -\frac{2}{3}WLC_{ox}\left(V_{GS} - V_{TS}\right)\left[\frac{1 + \eta + \eta^2}{1 + \eta}\right] \tag{5.61}$$

where we have used the algebraic identities $1 - x^3 = (1 - x)\left(1 + x + x^2\right)$ and $1 - x^2 = (1 - x)(1 + x)$. Plugging $|Q_{tot}|$ from (5.61) into our transit-time based noise formula, i.e., (5.54), gives

$$\overline{i_n^2} = 4kT\mu\left(\frac{2}{3}\frac{W}{L}C_{ox}\left(V_{GS} - V_{TS}\right)\left[\frac{1 + \eta + \eta^2}{1 + \eta}\right]\right) = 4kTg_{d0}\left(\frac{2}{3}\left[\frac{1 + \eta + \eta^2}{1 + \eta}\right]\right) \tag{5.62}$$

where we have used Einstein's relation to write $qD = kT\mu$, and g_{d0}, as defined previously, is the zero-bias conductance of the device. This is the classic equation for

[5]The discerning reader will immediately notice a problem with this argument at the drain end ($x = L$) when the device is saturated. Since the drain of the saturated MOSFET is weakly inverted, $Q_I(L)$ is likely to be small. More on this later.

the noise of a long-channel, above-threshold MOSFET. It may be rewritten as

$$\overline{i_n^2} = 4kT\gamma g_{d0} \tag{5.63}$$

where we now have an explicit, bias-dependent formula for γ

$$\gamma \equiv \frac{2}{3}\left[\frac{1+\eta+\eta^2}{1+\eta}\right] \tag{5.64}$$

It is easily seen that γ decreases from 1 ($\eta = 1$, zero-biased condition) to 2/3 ($\eta = 0$, saturation) as V_{DS} increases. Finally, let's try to clear up what happens to $Q_I(L)$, the channel charge density at the drain end, in saturation. The above-threshold equation (5.34) predicts that it should be zero, but that equation is inaccurate since the drain end of the channel is weakly inverted. By differentiating (5.58), we get

$$\frac{dQ_I}{dx} = \frac{1}{L}\left(\frac{Q_I^2(L) - Q_I^2(0)}{Q_I(x)}\right) \tag{5.65}$$

At the drain end, $x = L$, this becomes

$$\left(\frac{dQ_I}{dx}\right)_{x=L} = \frac{1}{L}\left(\frac{Q_I^2(L) - Q_I^2(0)}{Q_I(L)}\right) \approx \frac{1}{L}\left(\frac{Q_I^2(0)}{Q_I(L)}\right) \tag{5.66}$$

since $Q_I(0) \gg Q_I(L)$ in saturation. As $Q_I(L)$ decreases, dQ/dx at the drain increases. Note that the diffusion component of the channel current I_D is $WD(dQ/dx)$, where D is the diffusion constant. Therefore more and more of I_D is carried by diffusion at the drain end as the device approaches saturation. Saturation occurs when the concentration profile at the drain becomes steep enough to carry all of I_D via diffusion. This sets $Q_I(L)_{sat}$, the lowest possible (limiting) value of $Q_I(L)$. We can easily estimate this value by setting

$$DW\left(\frac{dQ_I}{dx}\right)_{x=L} = I_D \tag{5.67}$$

After substituting for $Q_I(0)$ and I_D (in saturation), using the Einstein relation and doing some algebra we get

$$Q_I(L)_{sat} = \frac{2\phi_T}{\kappa} C_{ox} \tag{5.68}$$

Going back to our derivation of the long-channel above-threshold MOSFET noise formula (5.62), it is now evident why we should take the predicted value of γ in saturation, 2/3, with a pinch of salt. That derivation assumed that $Q_I(x) \gg Q_I(L)_{sat}$ for all $x \in [0, L]$. This is obviously not true close to the drain when the device is saturated. Intuitively, we expect the real value of γ to be somewhat greater than 2/3 because the finite transit time of carriers through the channel region close to the drain (where most of the device current is carried by diffusion) should cause excess shot noise.

5.7 The Resistor at High Frequencies

In this section we consider what happens to our noise expressions outside the Rayleigh-Jeans limit. Let's first try to find the noise of a resistor as a function of frequency. We shall assume that the resistor is lumped and has no internal structure or dynamics of its own. In order for this to be true, the physical dimensions of the resistor must be much smaller than $\lambda_{min} = c/f_{max}$, where f_{max} is the highest frequency of interest.

A resistor generates noise because it exchanges energy with its surroundings. It captures photons from the electromagnetic radiation field surrounding it, and occasionally releases photons ("radiates") back into this field. Each photon has an energy $E = hf$. Therefore, the process of picking up and losing photons causes the total energy inside the resistor to fluctuate with time. This is thermal noise.

How does the resistor absorb and radiate photons? Consider the capture process first. A photon can be absorbed only if its quantum-mechanical wave function happens to intersect the volume occupied by the resistor. Let's start with the simplest case: a beam of monochromatic photons traveling in some direction z. Let the size of the photon's wave function (i.e., the rms value of the uncertainty in its position) in some perpendicular direction x be Δx. It seems reasonable to associate Δx with the length over which the photon detection probability falls to half. In other words, a

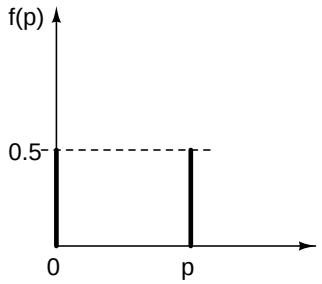


Figure 5-4: The probability distribution function $f(p)$ of momentum detected by a resistor illuminated by a monochromatic source of photons.

photon at a distance Δx from the resistor has 50% chance of being absorbed. Remember that a photon is said to be absorbed if its energy is transferred to charge carriers or lattice vibrations in the resistor, thereby changing the energy of the resistor and causing noise. The Heisenberg uncertainty principle tells us that

$$\Delta x \Delta p \geq \frac{h}{4\pi} \tag{5.69}$$

where Δp is the rms uncertainty (standard deviation) in the momentum of the photon. We now need to calculate Δp. Remember that 50% of the photons are absorbed and transfer a momentum $p = E/c = hf/c$ to the resistor, while the remainder are not absorbed and contribute nothing. Therefore the distribution of p, the detected momentum, is two-valued, with equal values of 0.5 at $p = 0$ and $p = hf/c$ and nothing in-between (see Figure 5-4)[6]. The standard deviation of this distribution is $\Delta p = p/2 = hf/(2c)$. By substituting this value into (5.69) and remembering that $c = f\lambda$, where λ is the wavelength of the photon, we get

$$\Delta x \geq \left(\frac{\lambda}{2\pi}\right) \tag{5.70}$$

Since the orientation of x in the plane perpendicular to z, was arbitrary, in general photon capture occurs within a circle of radius Δx centered around the resistor. This defines the *capture area* A_c of the resistor:

[6]Remember that we assumed that the incident photon beam was monochromatic.

$$A_c = \pi \left(\Delta x\right)^2 \geq \frac{\lambda^2}{4\pi} \tag{5.71}$$

The fact that this expression looks exactly like the effective area of an omnidirectional antenna is no coincidence [59, 162]. A resistor is, after all, nothing but an antenna: it captures photons from the electromagnetic radiation surrounding it. The only difference between resistors and antennas lies in how they are used. Antennas are typically used for picking up desired photons ("signal"), while resistors pick up unwanted photons ("noise"). We may therefore write

$$A_c = \frac{G(f)\lambda^2}{4\pi} \tag{5.72}$$

where $G(f) \geq 1$, known as the antenna gain, is in general a function of frequency. Also, in steady state (thermodynamic equilibrium) the resistor cannot be gaining or losing any energy on average. Therefore it must be radiating photons at the same rate it is capturing them. The radiation field surrounding our resistor can usually be assumed to be produced by, and in equilibrium with, a blackbody at a temperature T. This defines what we mean by the temperature of our resistor. Therefore, the power density (W/m^2) present in the radiation field per unit solid angle over a small frequency range df is the same as that radiated by the blackbody, which is known to be given by

$$B(f,T)df = \frac{2hf^3}{c^2}\left(\frac{df}{\exp\left(\frac{hf}{kT}\right)-1}\right) = \frac{1}{\lambda^2}\left(\frac{2hf}{\exp\left(\frac{hf}{kT}\right)-1}\right)df \tag{5.73}$$

(this is known as the Planck radiation function). Let's assume that the radiation field is isotropic, i.e., there is no preferred direction for photon motion. At thermodynamic equilibrium, the power present in the resistor must be the same as that present in the surrounding radiation field over an area $4\pi A_c$, where A_c is the capture area of the resistor (defined earlier)[7]. The 4π arises because we considered a beam of photons traveling in some direction while deriving A_c. In an isotropic radiation field

[7] If this wasn't true, the resistor couldn't be at thermodynamic equilibrium (steady state), since the average power entering it would not be zero.

equal numbers of photons travel in all possible directions, and so we must sum over all possible directions to get the total number of photons captured by the resistor. This summation gives us the total solid angle contained by a sphere, i.e., 4π. The power (in W) contained within the resistor between f and $f + df$ is therefore

$$S(f,T)df = 4\pi A_c B(f,T)df = \left(\frac{2hfG(f)}{\exp\left(\frac{hf}{kT}\right) - 1}\right) df \tag{5.74}$$

Remembering that the power present in a resistor is V^2/R, the power spectral density (in V^2/Hz) of the noise voltage developed by the resistor is given by

$$\overline{v_n^2(f,T)} = R \times S(f,T) = \left(\frac{2hfG(f)R}{\exp\left(\frac{hf}{kT}\right) - 1}\right) \tag{5.75}$$

However, since the time domain signal we measure is real, it has a power spectrum that is symmetric about $f = 0$. By Parseval's theorem, the total power in the time domain signal (extending from $-\infty$ to $+\infty$) is equal to that in the power spectrum from $-\infty$ to $+\infty$. Therefore only half the observed time domain noise power is contained in the positive frequency components of the power spectrum (extending from 0 to $+\infty$). However, in deriving (5.75), we implicitly assumed that the frequency f was positive. Therefore the measured voltage PSD will be twice that predicted by (5.75), i.e.,

$$\overline{v_n^2(f,T)} = \left(\frac{4hfG(f)R}{\exp\left(\frac{hf}{kT}\right) - 1}\right) \tag{5.76}$$

The low-frequency behavior of the function in (5.76) constitutes the Rayleigh-Jeans limit. At low frequencies $hf \ll kT$, $\exp(hf/kT) - 1 \approx hf/kT$, so we get

$$\overline{v_n^2(f,T)} \geq \left(\frac{4hfG(f)R}{hf/kT}\right) = 4kTG(f)R \tag{5.77}$$

The lower limit of this expression, i.e. $G(f) = 1$ at all frequencies, is the familiar, frequency-independent thermal noise expression for a resistor. Therefore an ideal resistor may be viewed as the most inefficient possible antenna. Its capture area A_c is determined solely by the uncertainty principle and is independent of geometry. A

practical antenna uses various geometrical tricks to increase A_c and thus the gain $G(f)$. From now on, we shall assume $G(f) = 1$ unless otherwise specified. An intuitive way to visualize the frequency-independent result in (5.77) is to realize that the thermal power contained in the resistor (in W) is the product of two terms: the blackbody spectrum $B(f,T)$, which is a radiation intensity (W/m^2) and the capture area A_c in m^2. At low frequencies $hf \ll kT$, $B(f,T)$ becomes (from (5.91)):

$$B(f,T) = \frac{2h}{c^2}\left(\frac{f^3}{exp\left(\frac{hf}{kT}\right)-1}\right) \approx \frac{2h}{c^2}\left(\frac{f^3}{hf/kT}\right) = 2kT\left(\frac{f}{c}\right)^2 \tag{5.78}$$

Therefore $B(f,T) \propto f^2$ at low frequencies. By contrast, the capture area is given by

$$A_c = \frac{\lambda^2}{4\pi} = \frac{1}{4\pi}\left(\frac{c}{f}\right)^2 \tag{5.79}$$

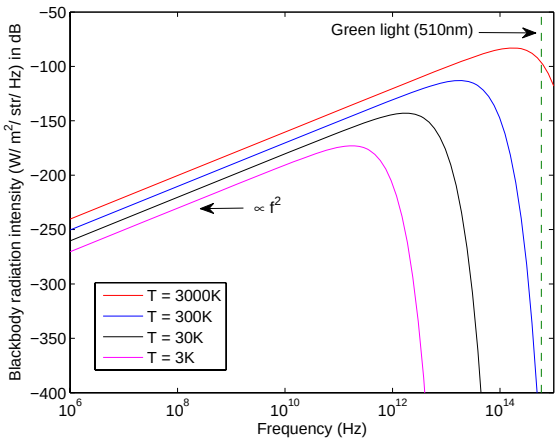
We see that $A_c \propto f^{-2}$. As before, the noise PSD of the resistor (W/Hz) for $hf \ll kT$ is given by

$$S(f,T) = 4\pi A_c(f) \times B(f,T) = 2kT \tag{5.80}$$

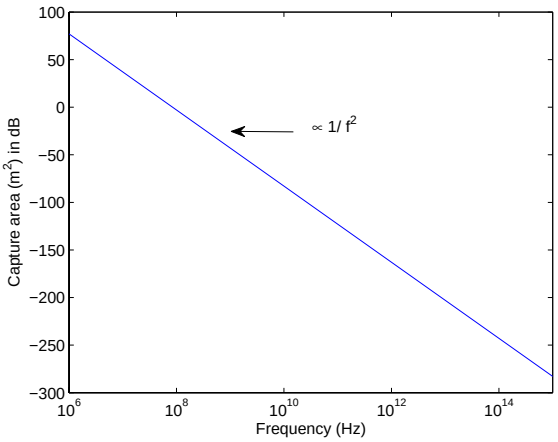
We see that $S(f,T)$ becomes independent of f because it is the product of two terms with opposing frequency dependencies. One term, $B(f,T) \propto f^2$, means that the energy density of photons in the space surrounding the resistor increases with frequency[8]. The other, $A_c \propto f^{-2}$, means that the volume of space from which the resistor can capture photons decreases with frequency. These relationships are shown graphically for several values of T in Figures 5-5(a), 5-5(b) and 5-5(c).

The sharp roll-off of the thermal noise PSD at high frequencies $hf \gg kT$ means that a detector with infinite bandwidth would still measure a finite variance in the voltage of the resistor. The total voltage variance is obtained by integrating $S(f)$ in (5.76) over all positive frequencies

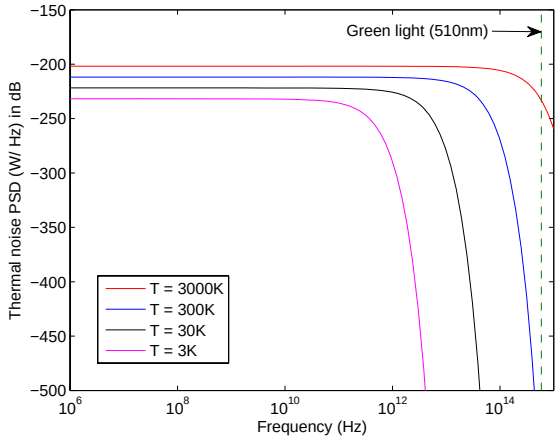
[8]Only if $hf \ll kT$, of course. If this relationship held for all f, $B(f,T)$ would increase without bound as $f \to \infty$. This is the so-called *ultra-violet catastrophe* that originally led Planck to postulate the existence of discrete quanta of energy (now known as photons).



(a) Planck blackbody radiation spectra at various temperatures.



(b) Capture area as a function of frequency.



(c) Thermal noise spectra at various temperatures.

Figure 5-5: Intuitive explanation for the presence of frequency-independent thermal noise in a resistor at low frequencies ($hf \ll kT$).

$$\begin{aligned}\overline{v_n^2} &= 4R\int_0^\infty \frac{hfdf}{\exp\left(\frac{hf}{kT}\right)-1} \\ &= 4R\frac{(kT)^2}{h}\int_0^\infty \frac{xdx}{\exp(x)-1}\end{aligned} \tag{5.81}$$

where $x = hf/kT$. The integral above can be found from the more general tabulated integral

$$\int_0^\infty \frac{x^n dx}{\exp(x)-1} = \zeta(n+1)\Gamma(n+1) \tag{5.82}$$

where $\zeta(n)$ is the Riemann zeta function, and $\Gamma(n)$ is the gamma function. In our case, since $n = 1$, the integral is equal to $\zeta(2)\Gamma(2)$. We also know that $\zeta(2) = \pi^2/6$ and $\Gamma(n) = (n-1)!$ for integer n. Therefore

$$\overline{v_n^2} = 4R\frac{(kT)^2}{h}\left(\frac{\pi^2}{6}\right) \equiv 4kTRB \tag{5.83}$$

where $B = (\pi^2/6) \times (kT/h)$ is the equivalent noise bandwidth of the system. If we plug in values for the constants, this formula predicts that a 1Ω resistor produces thermal noise with an rms magnitude of 413μV at $T = 300$K if the detector used to measure the noise has infinite bandwidth. Interestingly, (5.83) can also be written as

$$\overline{v_n^2} = \phi_T^2\left(\frac{2\pi^2}{3}\right)\frac{R}{R_K} \tag{5.84}$$

where $R_K = h/q^2 = 25812.8\Omega$ is known as the von Klitzing constant. The von Klitzing constant is a fundamental resistance unit that is used for accurately calibrating resistance values. It can be directly measured using the quantum Hall effect.

Since the shot noise produced by devices at thermodynamic equilibrium agrees with the thermal noise formulas for low frequencies, we also expect their high frequency behaviors to be related. Consider a spherical detector surrounding a isotropic source of diffusion current. The shot noise spectrum is flat at low frequencies because successive electron arrivals at the detector are uncorrelated. Therefore the autocor-

relation function $R(\tau)$ of $i(t)$, the detected current waveform, is approximately $\delta(\tau)$, the Dirac delta function. This is only *approximately* true because each electron arrival takes some characteristic time Δt to occur. Correlations do exist in $i(t)$ for timescales shorter than Δt, making $R(\tau) \neq 0$ for $\tau < \Delta t$. Also, the power spectrum is the Fourier transform of $R(\tau)$. The finite width of $R(\tau)$ makes the shot noise power spectrum roll off at high frequencies.

Electron arrivals are not instantaneous because of Heisenberg's uncertainty principle; electron wave packets have a characteristic size that is non-zero. As a result, we have to wait some time before the whole packet can enter the detector and be counted. This situation can be quantified by using another quantum-mechanical uncertainty relationship

$$\Delta E \Delta t \approx \frac{h}{2\pi} \tag{5.85}$$

where Δt is the rms uncertainty of the arrival time and ΔE is the rms uncertainty in the detected energy. Once the electron has been detected, ΔE simply becomes the rms thermal energy of the electrons in three dimensions, i.e.,

$$\Delta E = \frac{3}{2} kT \tag{5.86}$$

where each dimension (degree of freedom) contributes $kT/2$ of thermal energy. Substituting (5.86) in (5.85), we get

$$\Delta t \approx \frac{1}{3\pi} \left(\frac{h}{kT} \right) \tag{5.87}$$

Since the characteristic width of each electron arrival pulse is Δt, the bandwidth of the shot noise spectrum is given by

$$\Delta f = \frac{\text{TBP}}{\Delta t} = \text{TBP} \times 3\pi \left(\frac{kT}{h} \right) \tag{5.88}$$

where $\text{TBP} = \Delta f \Delta t$ is the time-bandwidth product of the electron arrival pulses. The actual value of TBP depends on the exact shape of the pulses. For example,

TBP=0.441 for Gaussian pulses, 0.142 for Lorentzian pulses and 0.443 for rectangular pulses. Comparing the shot noise bandwidth predicted by (5.88) with the equivalent thermal noise bandwidth B in (5.83) shows that they are both of the form $\alpha kT/h$, where α is a constant of order unity. In the absence of other mechanisms that cause correlation between electron arrival times[9], the thermal and shot noise behavior of devices at thermodynamic equilibrium therefore remain similar to each other at high frequencies.

5.8 The Real Resistor at High Frequencies

The derivation of thermal noise in the previous section is, while appealing, flawed. Our problems begin with the Planck blackbody formula, equation (5.91). Planck derived this formula in 1900 by calculating the average energy of a single energy radiator/vibrating atom/oscillator at a temperature T. His result was

$$E_1 = \frac{hf}{\exp\left(\frac{hf}{kT}\right) - 1} \tag{5.89}$$

However, in 1913 Einstein & Stern modified this formula to take into account real experimental data. Their result was

$$E_2 = \frac{hf}{\exp\left(\frac{hf}{kT}\right) - 1} + \frac{hf}{2} = \frac{hf}{2} \coth\left(\frac{hf}{2kT}\right) \tag{5.90}$$

where the additional $hf/2$ term is known as zero-point energy, since it is present in the oscillator even at zero temperature. Zero-point energy is the lowest possible energy that a quantum-mechanical system may have, and is also known as the ground-state energy. Intuitively, non-zero ground-point energy ensures that the system's energy can fluctuate randomly when interacting with a measurement apparatus while in

[9]This is a big assumption. In practice significant correlations may exist between carrier arrival times due to other mechanisms, such as Coulomb interactions. For example, shot noise is reduced by the electric field generated by the carriers themselves. The field tends to cause the carriers to bunch together, making their arrival times correlated. This process is known as space-charge smoothing and occurs in an above-threshold MOSFET. It is easy to check that the noise of a saturated above-threshold MOSFET is significantly lower than $2qI_D$, where I_D is the total drain current. In fact, the reduction factor (in power units) is approximately $(V_{GS} - V_T)/\phi_T$, which makes intuitive sense.

the ground state, thereby satisfying the uncertainty principle. It is now known that (5.90) is the correct expression for the mean energy of a quantum harmonic oscillator at temperature T. Therefore the radiation field surrounding the resistor should really be given by

$$B_2(f)df = \frac{2E_2}{\lambda^2} = \frac{2}{\lambda^2}\left[\frac{hf}{\exp\left(\frac{hf}{kT}\right)-1} + \frac{hf}{2}\right] \tag{5.91}$$

where the factor of 2 arises because of the two possible spin orientations of a photon. Therefore the power contained in the resistor between f and $f+df$ is given by

$$S(f)df = 4\pi A_c B_2(f)df = 2\left(\frac{hf}{\exp\left(\frac{hf}{kT}\right)-1} + \frac{hf}{2}\right) = (hf)\coth\left(\frac{hf}{2kT}\right) \tag{5.92}$$

The measured voltage PSD in the resistor is therefore

$$\overline{v_n^2(f)} = 2R \times S(f) = (2hfR)\coth\left(\frac{hf}{2kT}\right) \tag{5.93}$$

It is easy to show that (5.93) predicts the same PSD as our earlier result, (5.76), at low frequencies $hf \ll kT$. The result is $4kTR$, as expected. On the other hand, (5.93) blows up (becomes infinite) as $\omega \to \infty$, while (5.76) goes to zero. This also means that the total integrated noise power (V^2) of a constant resistor R is, in contrast with our earlier derivation, infinite. Why does our resistor then not blow up? Clearly, R itself must decrease at high frequencies for a physical resistor[10].

Optical systems typically operate in the limit where $hf \gg kT$. The power spectrum of noise at these high frequencies is simply

[10]The presence of zero-point energy imparts infinite total energy to the vacuum. This is a real problem because it makes quantum field theory calculations of apparently simple quantities blow up. This problem can be avoided by postulating that only *differences* in energy are meaningful, not absolute energy values. The resulting mathematical trick involves subtracting two infinite quantities to get a finite final answer and is known as renormalization. Fortunately, QED (quantum electrodynamics, the theory of electron-photon interactions) is amenable to this process. Not all quantum field theories are this lucky.

$$S(f) = hf \tag{5.94}$$

This is frequently referred to as the quantum noise limit of an optical detector.

5.9 The Fluctuation-Dissipation Theorem

It is interesting to note that the Nyquist noise formula for a resistor is really a fluctuation-dissipation theorem, with equilibrium voltage fluctuations at equilibrium resulting in dissipation at macroscopic scales. The resistance R is the macroscopic variable characterizing the amount of dissipation observed. The original quantum-mechanical proof of the fluctuation-dissipation theorem was due to Callen & Welton in 1951 [29]. An excellent review of the fluctuation-dissipation theorem in both classical and quantum contexts may be found in [152].

A system is said to be dissipative if it can absorb energy when a time-periodic perturbation (force) is applied to it. It is said to be linear if the power dissipation varies quadratically with the amplitude of the perturbation. Callen & Welton showed that for small perturbations, any system with energy levels (quantum mechanical stationary states) that are densely distributed in energy is dissipative and linear. Let the Hamiltonian of the system in the absence of perturbation be defined as H_0. A small perturbation $V(t)$ will change it to

$$H = H_0 + V(t)Q \tag{5.95}$$

where H_0, H and Q are functions of the coordinates and momenta of the particles in the system. The quantity $\delta Q/\delta t \equiv I$ is known as the response function (or susceptibility) of the system. Assume $V(t)$ varies sinusoidally with time, i.e., $V(t) = V_0 \sin(\omega t)$. For a linear system we can now define a complex impedance $Z(\omega)$ such that

$$V(\omega, t) = Z(\omega)I \tag{5.96}$$

In the familiar electrical case, an example of this relationship is $V = (R + \jmath\omega L)I$, where $I \equiv \delta Q/\delta t$. The instantaneous power in the system is $VI\cos(\theta)$, where θ is the angle between V and I. It is easy to see that $\theta = R(\omega)/\left|Z(\omega)\right|$, where $R(\omega)$ is the real (in-phase) component of $Z(\omega)$. The average power dissipated is

$$P = \frac{1}{2}V_0^2 \frac{R(\omega)}{\left|Z(\omega)\right|^2} \tag{5.97}$$

which is, as expected, quadratic in V_0. Now consider the system in thermodynamic equilibrium with no applied force. Even at equilibrium the system may generate spontaneously fluctuating forces and responses (i.e., noise). Let the mean square value of this spontaneously fluctuating force be denoted by $\langle v^2 \rangle$. Callen & Welton's fundamental theorem is the statement that

$$\langle v^2 \rangle = \frac{2}{\pi}\int_0^\infty R(\omega)E(\omega, T)d\omega \tag{5.98}$$

where T is the temperature and

$$E(\omega, T) = \frac{\hbar\omega}{2} + \frac{\hbar\omega}{\exp\left(\frac{\hbar\omega}{kT}\right) - 1} = \frac{\hbar\omega}{2}\coth\left(\frac{\hbar\omega}{2kT}\right) \tag{5.99}$$

Equation (5.98) is known as the generalized Nyquist relation. It can be applied to many types of systems, which do not have to electrical in nature[11]. In an electrical system the interpretation of forces and responses is obvious: V and v are voltages, I is current and R is the resistance. The term $E(\omega, T)$ is equal to the mean energy of a quantum harmonic oscillator at temperature T and frequency $\omega = 2\pi f$. It is easy to show that (5.98) reduces to the original Nyquist formula at high temperatures; when $kT \gg \hbar\omega$, we have $E(\omega, T) \approx kT$ and therefore

$$\langle v^2 \rangle \approx \frac{2kT}{\pi}\int R(\omega)d\omega \tag{5.100}$$

If $R(\omega)$ is constant, (5.98) gives us the noise produced by an ideal resistor R:

[11] Motion of particles in a viscous fluid is an example.

$$\langle v^2 \rangle = \frac{2R}{\pi} \int_0^\infty \left[\frac{\hbar\omega}{2} + \frac{\hbar\omega}{\exp\left(\frac{\hbar\omega}{kT}\right) - 1} \right] d\omega \tag{5.101}$$

which diverges to infinity. Equation (5.98) and Parseval's theorem also allows us to find $S(\omega)$, the power spectral density of $v(t)$ (the noise at thermodynamic equilibrium):

$$S(\omega) = \frac{2}{\pi} R(\omega) E(\omega, T) = R(\omega) \left[\frac{2\hbar\omega}{2\pi} \coth\left(\frac{\hbar\omega}{2kT}\right) \right] \tag{5.102}$$

Equation (5.102) can be rewritten in terms of $f = \omega/(2\pi)$ to get

$$S(f) = R(f) \left[(2hf) \coth\left(\frac{hf}{2kT}\right) \right] \tag{5.103}$$

Equation (5.103) agrees with the noise power spectral density of a resistor R derived earlier by considering the blackbody spectrum, but only when the ground-state energy $\hbar\omega/2$ of a quantum harmonic oscillator has been added to it. Ground state energy is the minimum energy that can be possessed by any quantum system.

The ground-state energy term is important: it reflects the presence of zero-point energy (vacuum fluctuations in quantum field theory) in the system. It also causes the total integrated noise to diverge as $\omega \to \infty$ (contrary to what would happen without it). One measurable effect of zero-point energy is the Casimir effect, where closely-spaced conductors attract each other with a measurable force even in vacuum. Another effect of zero-point energy is van der Waals forces. Zero-point energy, which is the energy in the ground/lowest possible energy state of a system, provides enough fluctuations to make sure that the uncertainty principle is not violated even at $T = 0$.

5.10 Physically-Observable Fluctuations

Equation (5.98) predicts that $\langle v^2 \rangle$ diverges when $R(\omega)$ is constant with frequency. The culprit is the zero-point energy, i.e., $E(\omega, 0) = \hbar\omega/2$. There are two possible ways for the measured value of $\langle v^2 \rangle$ to not diverge. The first possibility is that $R(\omega)$ is a strongly decreasing function of frequency. For example, any practical resistor R

will contain some parasitic shunt capacitance C that will tend to short it out at high frequencies. As a result $R(\omega)$ will be given by

$$R(\omega) = \mathrm{Re}[Z(\omega)] = \mathrm{Re}\left(\frac{1}{1/R + \jmath\omega C}\right) = \frac{R}{1+\omega^2\tau^2} \tag{5.104}$$

where $\tau = RC$. In this particular case the integral in (5.98) still diverges, although logarithmically rather than quadratically. The second possibility was first discussed by Weber in the 1950s [308, 309]. In the real world, we need some sort of apparatus to measure $\langle v^2 \rangle$. This measurement apparatus should have much lower input-referred noise than $\langle v^2 \rangle$. In fact, it should ideally be noiseless, i.e., at zero temperature with a well-defined energy E_a. For example, the apparatus could consist of electrons at a fixed potential V, in which case $E_a = qV$. In order to make accurate measurements we do not allow the apparatus to thermally equilibrate with the noisy system (the resistor), which is at temperature T[12]. Instead, the two systems are only connected for short periods of time. Inbetween these measurement periods the system is allowed to re-equlibrate with a heat bath at temperature T. When the systems are connected, they exchange energy, causing fluctuations in E_a which can be detected.

When the system and the measurement apparatus exchange energy, the system can transfer quanta of any frequency to the apparatus, because its energy can have any value (weighted by the usual Boltzmann probability distribution). However, the apparatus can only transfer quanta of frequency $\leq E_a/\hbar$ to the system, because its energy has a well defined value E_a prior to the transfer, and cannot go negative[13]. In other words, we must have $E_a - \hbar\omega > 0$ for such transitions. Weber showed that this restriction causes the physically-observable value of $\langle v^2 \rangle$, i.e., that detected by our measurement apparatus, to be given by

[12] Reducing the temperature of the measurement apparatus is sometimes the only way to increase measurement accuracy.

[13] All energies are measured with respect to the vacuum, which is, by definition, the lowest energy state. Negative energies would contradict this definition.

$$\begin{aligned}\langle v^2\rangle &= \frac{2}{\pi}\left[\int_0^{\omega_a} R(\omega)\frac{\hbar\omega}{2}\coth\left(\frac{\hbar\omega}{2kT}\right)d\omega + \int_{\omega_a}^{\infty} R(\omega)\frac{\hbar\omega}{\exp\left(\frac{\hbar\omega}{kT}\right)-1}d\omega\right] \\ &= \frac{2}{\pi}\left[\int_0^{\omega_a} R(\omega)\frac{\hbar\omega}{2}d\omega + \int_0^{\infty} R(\omega)\frac{\hbar\omega}{\exp\left(\frac{\hbar\omega}{kT}\right)-1}d\omega\right] \qquad (5.105)\end{aligned}$$

where the cutoff frequency $\omega_a = E_a/\hbar$. Comparing (5.105) with (5.101), we see that the temperature-dependent part of the noise extends over all frequencies, as before. However, the temperature-independent part, contributed by the zero-point energy, only results in fluctuations at frequencies less than ω_a. As a result, $\langle v^2\rangle$ does not diverge even when $R(\omega)$ is constant with frequency. To summarize, the fluctuations at $T = 0$ are limited by the fact that the energy of the system (also at $T = 0$) that is used to measure them cannot go negative. From (5.105), these measured zero-point fluctuations are given by

$$\langle v^2\rangle_{zp} = \frac{\hbar}{\pi}\int_0^{\omega_a} \omega R(\omega)d\omega \qquad (5.106)$$

5.11 Noise In Quantum Systems

The power spectral density $S(\omega)$ of a time-domain signal $x(t)$ is the Fourier transform of its autocorrelation function. The autocorrelation function in classical systems is defined as $R(\tau) = \langle x(t)x(t-\tau)\rangle$, where $\langle\rangle$ denotes averaging over time. For classical systems, $R(\tau)$ is a real signal and so $S(\omega) = S(-\omega)$, i.e., the power spectrum is symmetric. This applies to noise spectra as well. However, this is no longer true when quantum mechanical effects are significant.

Define $\langle\hat{x}(t)\rangle$ to be the expected value of the operator $\hat{x}(t)$ when the quantum system is in a stationary state[14]. The quantum autocorrelation operator is not com-

[14]In quantum mechanics, a stationary state is an eigenstate of a Hamiltonian, or in other words, a state of definite energy. It is called stationary because the corresponding probability density has no time dependence. As an eigenstate of the Hamiltonian, a stationary state is not subject to change or decay (to a lower energy state) over time. In practice, small perturbations eventually cause all stationary states to decay to the lowest energy state of the system, which is known as the ground state.

mutative. In other words, $\langle \hat{x}(t)\hat{x}(t-\tau)\rangle$ is not equal to $\langle \hat{x}(t-\tau)\hat{x}(t)\rangle$. Concisely, $[\hat{x}(t)\hat{x}(t-\tau)] \neq 0$ for $\tau \neq 0$, where $[AB] \equiv AB - BA$ is known as the commutator. As a result, a common way to define $R(\tau)$ for quantum systems is the so-called symmetric form

$$R(\tau) = \frac{1}{2}\langle \hat{x}(t)\hat{x}(t-\tau) + \hat{x}(t-\tau)\hat{x}(t)\rangle \tag{5.107}$$

This definition makes the autocorrelation a complex function. In fact $R(\tau) = R(-\tau)^* \neq R(-\tau)$, where * denotes the complex conjugate. As a result $S(\omega)$ is not symmetric in general, i.e., $S(\omega)$ is not equal to $S(-\omega)$. For example, the autocorrelation function of flux fluctuations in a quantum LC oscillator is given by [57]

$$R(\tau) = \langle \phi(t)\phi(t-\tau)\rangle = \frac{\hbar Z_0}{2}\left[\coth\left(\frac{\hbar\omega_0}{2kT}\right)\cos(\omega_0\tau) - \jmath\sin(\omega_0\tau)\right] \tag{5.108}$$

where $Z_0 = \sqrt{L/C}$ is the characteristic impedance, $\omega_0 = 1/\sqrt{LC}$ is the oscillation frequency and $\phi(t)$ is the flux in the inductor L. Note again that $R(\tau)$ is not real.

The power spectrum $S(\omega)$ measures the random fluctuations of a system and is proportional to the power exchanged with surrounding photons (i.e., an electromagnetic field). Positive frequencies can be thought of as the system adding energy by absorbing photons, while negative frequencies correspond to the system losing energy by radiating photons. Classically the two processes are identical and $S(\omega)$ is symmetric. A non-symmetric power spectrum means that, in general, radiated power is not equal to absorbed power. This is a violation of the antenna reciprocity theorem.

In particular, at zero temperature the system is in its lowest energy state and cannot radiate, though it can absorb energy. Therefore $S(-\omega)$ is zero at zero temperature, but $S(\omega)$ is not. An explicit formula for $S(\omega)$ was derived by van Hove in 1954 [120] based on a Born scattering approximation. It is given by

$$S(\omega) = \hbar\sum_{if} P_i \left|\langle f\left|\hat{x}\right|i\rangle\right|^2 \delta(E_i - E_f - \hbar\omega) \tag{5.109}$$

where $|i\rangle$ are the stationary states (Eigenstates) with energies E_i and populations (occupancies) P_i, and $\delta()$ is the Dirac delta function. The delta function may be considered a statement of energy conservation: the energy change $E_f - E_i$ in the system is the result of the absorption or radiation of a photon with energy $\hbar\omega$. The term $|\langle f\,|\hat{x}|\,i\rangle|$ may be interpreted the transition probability between states $|i\rangle$ and $|f\rangle$. At a finite temperature T, the populations P_i follow a Boltzmann distribution, as follows

$$P_i = \frac{\exp\left(\frac{-E_i}{kT}\right)}{\sum_i \exp\left(\frac{-E_i}{kT}\right)} \equiv \frac{\exp\left(\frac{-E_i}{kT}\right)}{Z} \tag{5.110}$$

where Z is known as the partition function of the system. The summation in (5.109) is carried over each possible state transition in the system. From detailed balance considerations it can be shown that, at equilibrium, (5.109) implies that

$$S(-\omega) = S(\omega)\exp\left(\frac{-\hbar\omega}{kT}\right) \tag{5.111}$$

This asymmetry in the power spectrum of noise disappears only when $\hbar\omega \ll kT$ and has been measured experimentally. The quantum version of the fluctuation-dissipation theorem [57, 152] predicts that the power spectral density of spontaneous voltage fluctuations exhibited by a system with resistance $R(\omega)$ at thermodynamic equilibrium is given by

$$S(\omega) = R(\omega)\left[\frac{\hbar\omega}{2\pi}\left(\coth\left(\frac{\hbar\omega}{2kT}\right) + 1\right)\right] \tag{5.112}$$

The quantum fluctuation-dissipation theorem is essentially a generalization of Planck's blackbody radiation law. The impedance $Z(\omega) = R(\omega) + \jmath X(\omega)$ acts as the black body radiator. It couples with a surrounding electromagnetic field by exchanging photons. It can be easily verified that $S(\omega)$ in (5.112) satisfies the relation in (5.111), and $S(\omega) \neq S(-\omega)$. In the various limits $S(\omega)$ is given by

$$\begin{aligned} S(\omega) &= \frac{2kT}{2\pi} R(\omega), |\hbar\omega| \ll kT \\ &= \frac{\hbar\omega}{2\pi} R(\omega), \hbar\omega \gg kT \\ &= 0, \hbar\omega \ll -kT \end{aligned} \tag{5.113}$$

Note that in experiments one usually measures (for example, with a spectrum analyzer) the total noise between two frequencies ω_1 and ω_2 in *both positive and negative frequency bands*. Therefore, the measured noise spectrum is the "symmetric" spectrum $S_{sym}(\omega) \equiv S(\omega) + S(-\omega)$, where in general $S(\omega)$ is not equal to $S(-\omega)$. From (5.112), we have

$$S_{sym}(\omega) \equiv S(\omega) + S(-\omega) = R(\omega) \left[\frac{2\hbar\omega}{2\pi} \coth\left(\frac{\hbar\omega}{2kT} \right) \right] \tag{5.114}$$

Comparing (5.114) with (5.102), we see that S_{sym} is exactly the same power spectrum predicted by Callen & Welton's original derivation of the fluctuation-dissipation theorem! The various formulas for $S(\omega)$ for a constant resistance $R(\omega) = 1\Omega$ are graphed in Figure 5-6. In a real experiment, the zero-temperature part of the PSDs will only contribute noise up to a finite frequency, as shown in (5.105). As a result, the total measured noise will always be finite.

The net absorption of energy by a device in an electromagnetic field with N_ω photons is equal to $N_\omega S(-\omega) - (N_\omega + 1) S(\omega)$. In a classical field with $N_\omega \gg 1$ this results in the device having a net DC conductance G given by

$$G = \alpha G_K \left(\frac{S(-\omega) - S(\omega)}{\hbar\omega} \right) = 2\alpha G_K \tag{5.115}$$

where $G_K = e^2/h$ is the basic unit of conductance (the van Klitzing constant) and α is a device-dependent constant. The relationship above can be proved by substituting $S(\omega)$ from (5.112) and is known as the Kubo conductance formula. Many mesoscopic systems have been shown to exhibit the Kubo conductance.

An interesting aside now follows. It can be shown that a active linear amplifier *has*

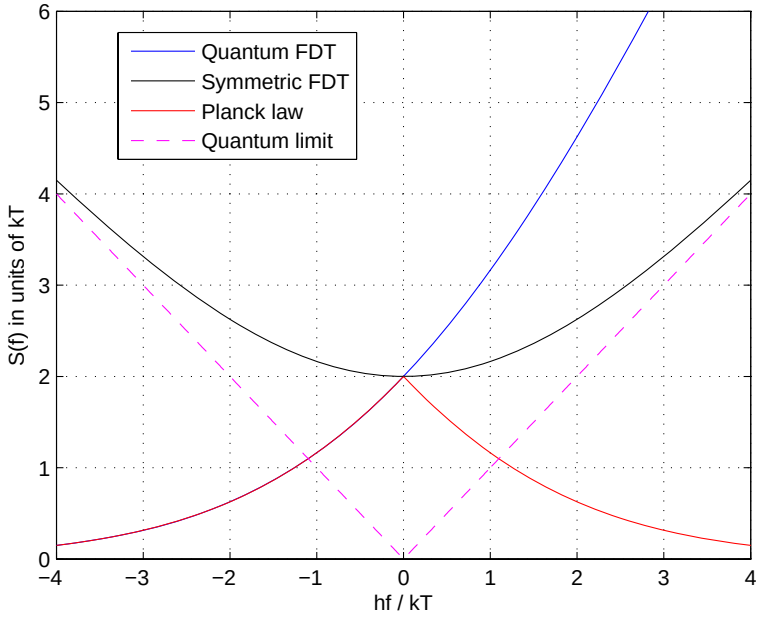


Figure 5-6: Various expressions for the voltage noise PSD of a 1Ω resistor. The curves show: (a) the quantum fluctuation-dissipation theorem (FDT) prediction (5.112), (b) the symmetric quantum FDT spectrum (5.114), (c) the symmetrized spectrum predicted by the Planck radiation law with no zero-point energy (5.75) and (d) the high-frequency quantum noise limit (5.94). Note that formulas (a)-(c) agree with each other at low frequencies $hf \ll kT$ - they all predict $2kT$.

to add noise to the input signal [113]. Otherwise it could violate the Heisenberg uncertainty principle (i.e., $\Delta x \Delta p \geq \hbar/2$). Fundamentally, this is because all active devices, such as amplifiers, must be coupled to power sources ("reservoirs"). As a result their governing equations cannot be derived from a Hamiltonian (which is time-reversible). In other words, they exhibit dissipation, which causes them to spontaneously generate fluctuations (noise). It *is* however, possible to get signal amplification without adding noise. Parametric amplification, where the value of a lossless reactance (such as a capacitance) is varied with time, is an example. The governing equations of a parametric amplifier can be derived from a Hamiltonian. In other words the system is (ideally) lossless and therefore adds no noise to the signal.

Standard Hamiltonian formulations of quantum mechanics have trouble dealing with dissipation and dissipative elements like resistors because the underlying equations are time-reversible. One way to avoid this problem is known as the Caldeira-Leggett model. It replaces each dissipative element by an infinite array of lossless independent oscillators. This provides the large number of degrees of freedom needed to model dissipation. In spirit, this is similar to why an infinite transmission line can provide a resistive input impedance even though it is lossless.

Calculations of noise in mesoscopic devices open up a whole can of worms. The fluctuation-dissipation theorem still applies in its quantum version, so we can find the noise at zero-bias pretty easily. However, at non-zero bias the shot noise expressions can get arbitrarily complex depending upon how much correlation is exhibited by the charge carriers (electrons, Cooper pairs, etc.). The high-temperature limit when electron arrivals are uncorrelated usually reduces to the form $2\gamma qI$, where γ (the "Fano factor" or "shot noise suppression/ enhancement factor") is a constant that depends on the physical structure and charge transport properties of the system. A nice discussion is available in [277].

Mesoscopic devices are usually analyzed somewhat similarly to RF systems, by using a N-port scattering matrix. The reflected and transmitted terms are complex, reflecting the fact that we have wave phenomena (i.e., phases are important). The waves of course are quantum mechanical wave functions.

At very low temperatures the noise in physical devices is dominated by zero-point fluctuations. This transitions smoothly into thermal noise as temperature increases, i.e. $kT > \hbar\omega$. Similarly, the equilibrium thermal noise fluctuations transition smoothly into non-equilibrium (bias-dependent) shot noise as the bias voltage V increases, i.e., $qV > kT$. In general therefore, the current noise PSD in a mesoscopic device depends on $\omega = 2\pi f$, V and T, and can be written as $S_I(\omega, V, T)$. This reflects the presence of three different energy scales in the system: $\hbar\omega$, kT and qV. See [277] for an example.

5.12 Noise in Antennas

Antennas are interesting from a thermodynamic point of view since their effective noise temperature is not necessarily that of their physical surroundings. This is because they can be in thermodynamic equilibrium with distant blackbody sources at any temperature. The thermal noise voltage PSD at the terminals of an impedance-matched antenna is given by $4kT_AR_A$, where R_A is the real part of the antenna terminal impedance and T_A is the noise temperature. In general T_A is a function of the antenna's radiation pattern and physical orientation with respect to visible sources of radiation. For example, consider a high-gain antenna in free space. If the main lobe is pointing away from bright nearby sources, like the sun, the measured noise temperature will be close to 2.73K, which is the temperature of the cosmic microwave background. On the other hand, if the antenna is pointing directly at the sun, the measured noise temperature will be close to that of the solar photosphere, i.e., about 5700K.

Now imagine that we have a receiver connected to the antenna through a transmission line of length L. If the line is lossless it is thermodynamically isolated from its surroundings and adds no noise of its own to the received signal. Therefore the noise temperature T_{rec} seen by the receiver will be equal to T_A, that of the antenna. However, if the line has loss this statement is no longer true. Consider a slightly lossy line where the signal attenuates by a factor of e in a distance $1/\alpha$, i.e.

$I(x) = I(0)\exp(-\alpha x)$, where $I(x)$ is the power at position x along the line and the line extends from $x = 0$ to $x = L$. Here α is known as the attenuation constant of the line. Say the temperature of the line is T_{line}, i.e., it absorbs and reradiates photons in thermodynamic equilibrium with a blackbody radiation field at temperature T_{line}.

It is easy to show that the noise temperature of the receiver is given by

$$T_{rec} = T_A \exp(-\tau) + T_{line}\,(1 - \exp(-\tau)) \tag{5.116}$$

where $\tau = \int_0^L \alpha dx$ is known as the optical depth. We see that as the optical depth increases, the line becomes opaque. In other words, the receiver reaches thermodynamic equilibrium with the line and no longer "sees" the antenna.

We now add a final note on antenna effective areas. We have shown that the minimum possible effective area $A_{eff,min}$ of an antenna is set by the uncertainty principle and is equal to $\lambda^2/(4\pi)$. In general, the effective area A_{eff} is bigger than $A_{eff,min}$ by a ratio $G > 1$, known as the antenna gain. In other words, $A_{eff} = GA_{eff,min}$. We now want to relate A_{eff} to the aperture area A_{phys} of large aperture antennas such as horns and dishes. For simplicity, consider a circular, uniformly illuminated aperture of diameter $D \gg \lambda$. The effective area of the aperture is then equal to the physical area, and is given by $A_{phys} = \pi D^2/4$. In order to calculate the gain of this antenna, we use the well known fact that the first nulls in the diffraction pattern from circular apertures occur at angles $\Delta\theta = \pm 1.22\,(\lambda/D)$ when $\lambda \ll D$ and θ is the solid angle measured with respect to the axis perpendicular to the aperture and passing through its center.

The main lobe of the antenna radiation pattern fits within the nulls in the diffraction pattern, i.e., $\pm\Delta\theta$, instead of spreading uniformly over a solid angle of 2π (the hemisphere on the outward, radiating face of the aperture). If we approximate the intensity within $\pm\Delta\theta$ as uniform, simple geometry shows that the illuminated spot at a distance r from the antenna has an area $\pi\,(r\Delta\theta)^2$. Therefore the antenna gain G is given by

$$G \approx \frac{2\pi r^2}{\pi \left(\Delta\theta r\right)^2} = 1.34 \left(\frac{D}{\lambda}\right)^2 \tag{5.117}$$

By definition, therefore, the effective area of the antenna is given by

$$A_{eff} = \frac{\lambda^2}{4\pi} G = \frac{1.34}{\pi^2} \left(\frac{\pi D^2}{4}\right) = 0.136 A_{phys} \tag{5.118}$$

Thus the effective area of the antenna is a constant fraction of its physical area, a result that makes intuitive sense. The result may also be generalized to other geometries, i.e., dishes which are non-circular and/or not uniformly illuminated. The proportionality between A_{eff} and A_{phys} always remains, but the numerical constant relating them is geometry-dependent.

5.13 Flicker Noise

Flicker or $1/f$ noise is an example of a dynamic process with a power-law power spectrum, i.e. $P(f) = f^{-\alpha}$, where α is a constant that in the case of $1/f$ noise is equal to 1[15]. Typically a device exhibits a $1/f$ power spectrum only over a certain range of frequencies. However, power spectra with power-law behavior are found in a surprisingly wide range of physical systems. To give just a few examples:

- The distribution of stellar luminosities follows a broken power law (as do many other astrophysical distributions).
- The spatial frequency distribution of natural images is approximately $1/f^2$, i.e., is analogous to $1/f$ noise, but in two dimensions [247].
- The power spectral density of much orchestral music is approximately $1/f$ [304].
- The degree distribution of many natural and man-made "scale-free" networks, such as the internet and ecological webs, follow power laws.

[15] Flicker noise was first discovered in vacuum tubes. The term "flicker" appears to have been proposed by Schottky in a paper published in 1926 [264], where he says: "If we had to do with emission of light instead of electrons, we would speak of a chaotic variation in light intensity taking place over the surface of the cathode, a phenomenon which we should describe by the word "flicker.""

- Zipf's law says that a power-law plot with slope -1 results if the words in a language are ordered (ranked) by frequency of usage, and then these frequencies plotted versus the rank. In other words, the frequency with which a word is used is inversely proportional to its rank.

Examples may be multiplied. The basic feature these diverse systems share is a tendency for phenomena to occur over large ranges of parameter values. Adding up these "scale-free" or "scale-invariant" phenomena results in power spectra that are power-law in nature. Several attempts have been made to derive more quantitative mechanisms to explain the somewhat vague statement above, but without too much success. The truth appears to be that the detailed mechanisms that cause power-law behavior are different across systems, so "universal" theories are likely to fail.

5.13.1 MOSFET Flicker Noise

We shall briefly consider $1/f$ noise in MOSFETs. Further details are available in many places, for example [101, 312]. Long-term measurements of flicker noise in bulk semiconductors are also available in the literature [30]. There are two main mechanisms that have been postulated for $1/f$ noise generation in MOSFETs:

- Mobility fluctuations due to carriers being scattered by vibrations of immobile atoms in the channel. Such vibrational modes are quantized by the regularity of the Silicon crystal lattice, and are known as phonons.

- Threshold voltage fluctuations due to the presence of electron traps in the gate oxide. Traps are metastable energy levels within the oxide band-gap. Electrons can reach such energy levels by quantum-mechanical tunneling through the oxide. Traps are produced by impurities or defects in the oxide crystal lattice, and their average number per unit area of gate oxide is a measure of the "cleanliness" of the fabrication process.

Each mechanism can produce $1/f$ spectra by itself; it seems likely that both are involved in real devices. However, we shall use the oxide charge trap concept, as

it has been shown to adequately describe the experimentally-measured $1/f$ noise of MOSFETs [124]. Experimental data from MOSFETs that are small enough to contain a single trap within their gate oxide has been analyzed in [101]. The drain current in such devices fluctuates between just two levels, corresponding to the trap being either empty or full. In the simplest model, each trap receives carriers as Poisson arrivals with a mean rate λ_i that depends on the average current through the device and the location of the trap. Once filled, the probability of the trap being empty increases exponentially with a time constant $1/\mu_i$, which varies across traps. We define the *trap occupancy function* F_i as having a value 1 if the trap is occupied, and 0 if it is empty.

Each trap can be compactly described by a two-state Markov process. By requiring detailed balance in steady-state it is easy to show that the probability of the trap being filled is given by

$$D_i = \frac{\lambda_i}{\lambda_i + \mu_i} \tag{5.119}$$

The effect of a filled trap is to change the threshold voltage of the transistor by an amount ΔV_T, which to a good approximation can be taken as constant [101] and given by $\Delta V_T = q/\left(C_{ox} W L\right)$. The resultant change in drain current is given by

$$\Delta I = g_m \times \Delta V_T = \overline{I}\left(\frac{q}{C_{ox} W L V_L}\right) \tag{5.120}$$

where $g_m = \overline{I}/V_L$ is the small-signal transconductance of the transistor, and V_L is its linear range. The amplitude distribution of ΔI has two states: 0, with probability $(1 - Di)$ and ΔI, with probability D_i. The variance of this distribution is

$$\overline{\Delta I^2} = D_i \left(1 - Di\right) \left(\Delta I\right)^2 = \sigma_i^2 \left(\Delta I\right)^2 \tag{5.121}$$

where $\sigma_i^2 = D_i \left(1 - D_i\right)$ is the variance of F_i. Note that σ_i^2 is maximized when the "duty cycle" $D_i = 1/2$, i.e., $\lambda_i = \mu_i$. Electrons arrivals and departures at the trap are assumed to be uncorrelated, and occur with rates λ_i and μ_i, respectively.

Counting either flux leads to a Poisson distribution. Therefore the average rate at which Poisson events occur at the trap is $\lambda_i + \mu_i$, allowing us to define the *trap time constant* $\tau_i = 1/(\lambda_i + \mu_i)$. It is easy to show that the autocorrelation function of the F_i is simply $\exp(-|t|/\tau_i)$. As a result, the power spectrum $P_i(f)$ of F_i is simply that of a first-order low-pass filter, and is given by

$$P_i(f) = 4\sigma_i^2 \left[\frac{\tau_i}{1 + (2\pi f \tau_i)^2}\right] \tag{5.122}$$

We have seen that fluctuations in drain current are proportional to the trap occupancy function F_i. This function varies between 0 and 1, and its variance σ_i^2 varies between 0 and 1/4. Now assume we have a total of N traps and that their dynamics are uncorrelated. The total drain current fluctuation will simply be the sum of fluctuations due to individual traps. The central limit theorem guarantees that, in the limit of large N, the amplitude distribution distribution of drain current fluctuations will converge to a Gaussian with variance $\propto N\overline{\Delta I}^2$.

The arrival rate λ_i is proportional to the probability of electron tunneling between the channel and the trap, and decreases exponentially as the distance d between the two increases. It is usually assumed that traps are uniformly spread throughout the gate oxide. In this case d has an uniform distribution, and both λ_i and the trap time constant $\tau_i \approx 1/\lambda_i$ should have exponential distributions. Let us assume that trap time constants extend from τ_{min} to τ_{max}, with the i-th trap having a time constant given by $\tau_i = \tau_{min} \exp(i/N_{nat})$. Here N_{nat} is the number of traps per e-fold. The power spectrum produced by the combined affect of all N traps is simply

$$P(f) = (\Delta I)^2 \sum_{i=1}^{N} P_i(f) \tag{5.123}$$

We can obtain an upper bound on the power spectrum by assuming that $\lambda_i = \mu_i$ for all traps. This choice makes the variance $\sigma_i^2 = 1/4$, its maximum value $\forall i$. The sum can be approximated as an integral when $N_{nat} \gg 1$. The result is

$$P(f) = (\Delta I)^2 N_{nat} \left[\frac{\tan^{-1}(2\pi f/f_{min}) - \tan^{-1}(2\pi f/f_{max})}{2\pi f}\right] \tag{5.124}$$

where $f_{min} = 1/\tau_{min}$ and $f_{max} = 1/\tau_{max}$. For frequencies much smaller than f_{max}, but much larger than f_{min}, i.e. for $f_{min} \ll f \ll f_{max}$, the expression above can be simplified to

$$P(f) \approx \frac{N_{nat}}{4}\left(\frac{(\Delta I)^2}{f}\right) \tag{5.125}$$

which is indeed proportional to $1/f$, and also to N_{nat}, the spectral density of traps, as one might expect. Thus, $1/f$ power spectra result for frequencies between f_{min} and f_{max}. The results of adding together a finite number of individual trap spectra are shown in Figure 5-7 for $\Delta I = 1$, $f_{max}/f_{min} = 10^3$ and $N_{nat} = 145$. We see that a very good approximation to the $1/f$ behavior predicted by (5.125) results between f_{min} and f_{max}. The figure also shows that this behavior is robust: replacing the constituent first-order low-pass functions with resonant second-order low-pass or band-pass functions does not significantly change $P(f)$[16].

In general, we do not expect $\lambda_i = \mu_i$ for each trap. In this case we cannot analytically show that a $1/f$ spectrum results. However, to first order we might expect (5.125) to still hold, but with $\sigma_{av}^2 = \left(\sum_{i=1}^{N} \sigma_i^2\right)/N$, the average variance of the traps, replacing the factor of 1/4. Note that we must have $\sigma_{av}^2 \leq 1/4$. In this case, we get

$$P(f) \approx \sigma_{av}^2\left(\frac{(\Delta I)^2 N_{nat}}{f}\right), \tag{5.126}$$

Substituting (5.120) in (5.126), we get

$$\frac{P(f)}{\bar{I}^2} = \left(\frac{\sigma_{av} q}{C_{ox} W L V_L}\right)^2 \frac{N_{nat}}{f} \tag{5.127}$$

Writing $N_{nat} = N_{ot} W L$, where N_{ot} is a process-dependent parameter that scales with the number of traps per unit gate area, we get

[16]This statement is true as long as the quality factor Q of the resonant low-pass or band-pass functions is lower than approximately $1/\sqrt{2}$. For larger values of Q peaks appear in $P(f)$ around f_{min} and f_{max}.

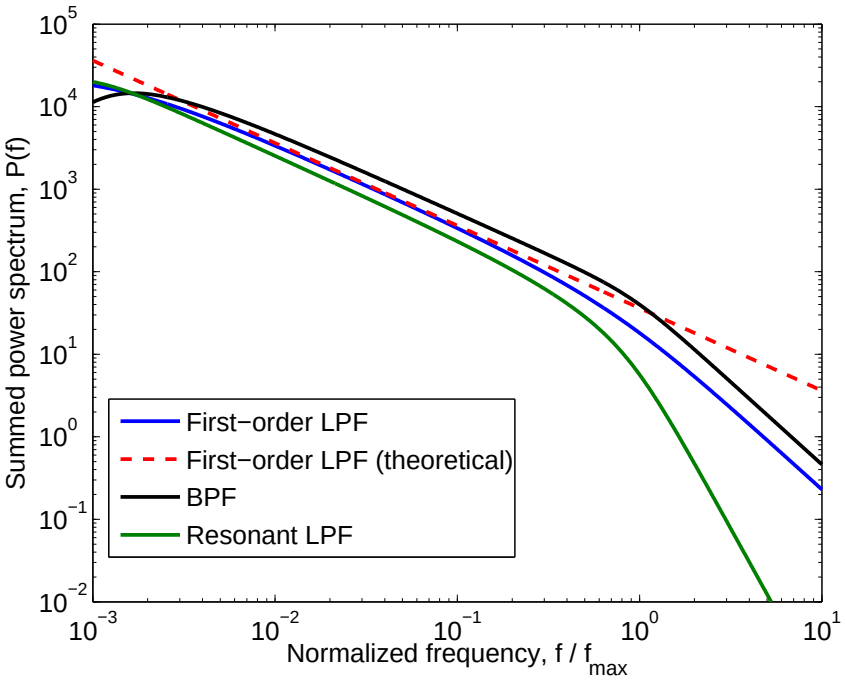


Figure 5-7: Flicker noise power spectra produced by adding together low-pass (LPF) and band-pass (BPF) spectra with exponentially distributed time constants. The theoretical result, i.e. (5.125), is also included for comparison.

$$\frac{P(f)}{\overline{I}^2} = \frac{N_{ot}}{WL}\left(\frac{\sigma_{av}q}{C_{ox}V_L}\right)^2\frac{1}{f} \tag{5.128}$$

Equation (5.128) has all the right scaling relationships associated with $1/f$ noise in MOSFETs. In particular, the power spectral density of drain current fluctuations scales inversely with the square root of the gate area WL, indicating that we can trade-off between layout area and $1/f$ noise. The same scaling behavior is predicted by the classical Pelgrom model for MOS transistor mismatch [223], again illustrating that DC offsets and flicker noise are manifestations of the same physical phenomenon.

Let us denote the lowest frequency resolvable in a given $1/f$ noise measurement to be $f_{exp} = \omega_{exp}/(2\pi)$. The time-frequency tradeoff guarantees that $\omega_{exp} \approx 1/T_{exp}$, where T_{exp} is the length of the measurement. Traps with rate constants $(\lambda_i + \mu_i) < \omega_{min}$ will, on average, show no dynamic behavior during the duration of the measurement. However, we cannot predict whether they are empty or filled. The result is static variability, or "DC" offset between transistors. Thus, flicker noise and DC

offsets (mismatches) between transistors are deeply related. The variance of DC offset in drain current may be defined as

$$I_{off}^2 = \int_0^{f_{exp}} P(f)df \tag{5.129}$$

The alert reader may have spotted a problem with our definition of I_{off}. If trap time constants extend to ∞, i.e., $f_{min} \rightarrow 0$, it is evident from (5.125) that $P(f)$ diverges as $f \rightarrow 0$. As a result, I_{off} must also diverge as $f_{min} \rightarrow 0$. However, we do not measure transistor mismatch that increases without bound as measurement time increases (in fact, it seems to saturate to a fixed value). Of course, the integral only diverges logarithmically, which may simply be too slow to be observed. The other obvious possibility is that $f_{min} \neq 0$, i.e., there is a "slowest" trap. In fact, modern CMOS transistors often contain a small number of traps, i.e., N is frequently a small number. In this case $P(f)$ should flatten out and reach a constant value at frequencies smaller than $1/(2\pi\tau_{max})$, where τ_{max} is the time constant of the slowest trap. Any measurement that is longer than approximately τ_{max} will then measure a fixed value of DC offset, i.e. the integral $\int_0^{f_{exp}} P(f)df$ will converge.

5.13.2 Experimental Measurements

We decided to experimentally check whether $P(f)$ flattens out at very low frequencies. Our experimental setup is shown in Figure 5-8. The test device was an NMOS transistor of size 3.6μm $\times$1.8μm, fabricated in the AMI 0.5μm process and biased at a current of 10μA. Its drain current was sampled by measuring the drain voltage at a rate of 0.7Hz with a Keithley 2400 source-meter, and dividing by the known load resistance ($R_L = 100\text{k}\Omega$). The power supply was an Agilent E3610A, with V_{DD} set to 2V. The whole setup was placed inside a grounded metal box to reduce noise pickup from environmental electromagnetic fields. The whole system was kept in a room whose temperature was maintained at 298$\pm$0.5K. In order to prevent aliasing due to sampling the bandwidth at the output node was reduced by placing a capacitor of value 220μF there. In conjunction with the 100KΩ load this capacitor forms a first-

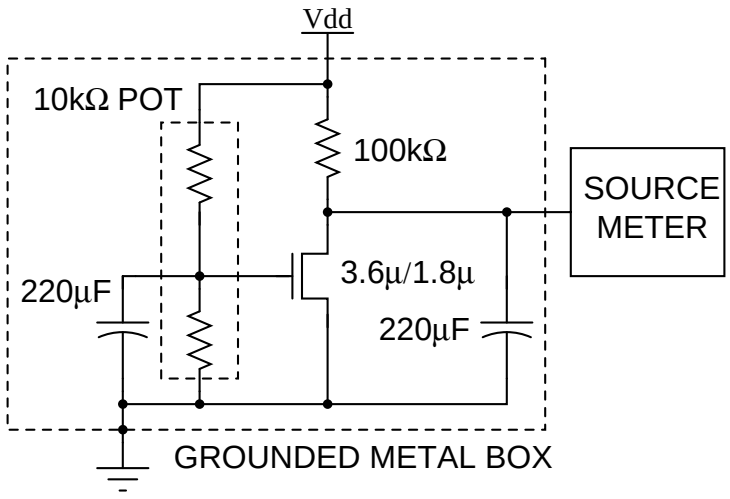


Figure 5-8: Experimental setup for measuring flicker noise in a MOSFET.

order low-pass filter with corner frequency $\tau_{LPF} = 7.2$mHz. The effects of this filter on the power spectrum $P(f)$ were later removed in software by multiplying the measured $P(f)$ with the inverse of the its (squared) transfer function, i.e. $1 + (2\pi f \tau_{LPF})^2$.

The power spectrum $P(f)$ was obtained from the time-domain data (after the average value had been removed) using Welch's method. In this method the time-domain waveform is split into M overlapping windows (we used 8 Hamming-weighted windows). The discrete fourier transform (FFT) $X(f)$ of each window is taken. The squared magnitude of $X(f)$ is then averaged over all windows to get $P(f)$. The method, because it effectively creates an ensemble average of M runs, decreases the displayed variance of the $P(f)$ curve by a factor of M. Its disadvantage is that the minimum frequency that can be resolved also increases by a factor of M. Figure 5-9 shows the measured power spectrum obtained from 340 days of time-domain data (20.7 million points). The sharp peaks correspond to a daily periodicity (1/86400 seconds or 11μHz) and harmonics of it. Diurnal temperature fluctuations are probably responsible for this periodicity, but other diurnal phenomena, such as fluctuations in the AC line voltage due to varying power requirements on the grid, may also be involved. By taking data for close to an entire year we avoid possible long term trends due to temperature fluctuations that occur on an annual scale.

A potential source of error arises from the well-known fact that discrete resistors

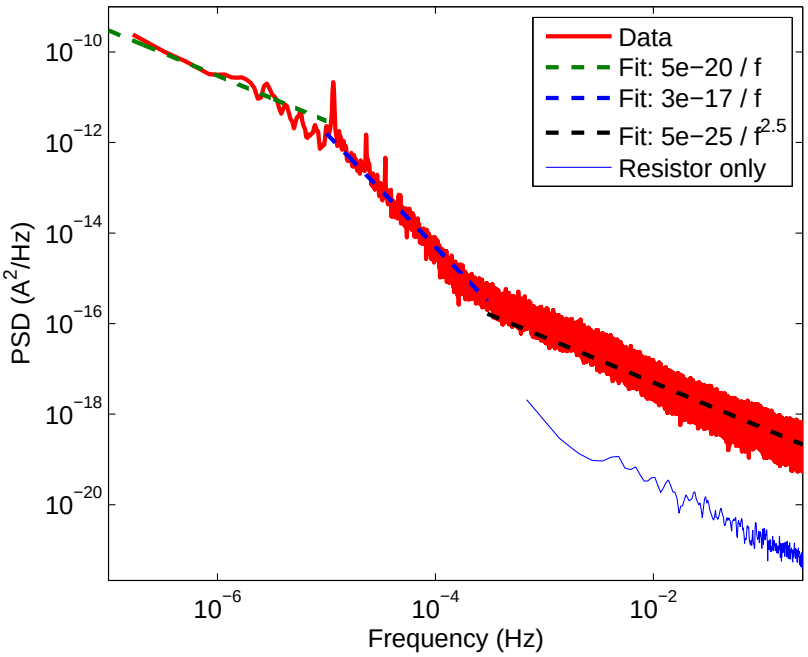


Figure 5-9: Measured drain current power spectrum. The power spectrum of the resistor by itself and fits to $1/f$ and $1/f^{2.5}$ spectra are also shown.

themselves exhibit $1/f$ noise (for example, see [158]). We therefore ran a calibration test where we replaced our experimental setup with a simple resistive divider connected between V_{DD} and ground. Resistors identical to R_L (100kΩ, 0.5W, carbon) were used. Since the $1/f$ noise in a resistor often depends on the DC voltage across it, we also ensured that the DC voltage drop across each resistor was the same (1V) as that across R_L in our experimental setup. The measured power spectrum for a single resistor is shown in Figure 5-9. It is also $1/f$ in nature, but at least two orders of magnitude lower than the spectrum measured with the transistor. Therefore the transistor produces essentially all the measured $1/f$ noise. This makes intuitive sense: the discrete resistor is physically much larger than the integrated transistor and contains many more traps or scattering centers, making their *relative* effects on current smaller. In other words, these randomly-distributed objects are spatially averaged out to a larger extent in the big discrete resistor.

Figure 5-9 shows that there are three main regions in the power spectrum: $1/f$ at high frequencies, $1/f^{2.5}$ at intermediate frequencies and $1/f$ again at low frequencies.

The high-frequency region is due to the dynamics of multiple traps (or scattering centers) within the device itself. The intermediate region ends almost exactly at 1/(day) (11μHz), leading us to believe that it is the result of diurnal temperature fluctuations. The power spectrum of atmospheric temperature fluctuations has been measured to be approximately $1/f^{1.5}$ for frequencies higher than 1/month (0.3μHz), and $1/f^{0.5}$ for lower frequencies [224].

From SPICE simulations we expect this transistor to have a small-signal transconductance at $\overline{I} = 10\mu$A of $g_m = 5.4\times10^{-5}\mho$[17]. The expected thermal noise power spectral density at the drain node due to the transistor and load resistor is $4kT\left(1/R_L + \gamma g_m\right)$, where $\gamma \approx 2/3$ is the transistor's excess noise factor. Substituting known values, we get $7.7\times10^{-25}\text{A}^2/\text{Hz}$, which is orders of magnitude smaller than $P(f)$ at all frequencies of interest. Therefore thermal noise can be ignored.

The standard deviation in drain current that we expect to measure with a measurement of length $\tau_{meas} = 1/f_{stop}$ can be obtained by integrating the power spectrum from high frequencies till f_{stop}. Thus it is given by

$$\sigma_I\left(f_{stop}\right) = \int_{f_{stop}}^{\infty} P(f)df \tag{5.130}$$

In this equation '∞' should be understood to be the highest available frequency, which, by Nyquist, is $f_{samp}/2$, where f_{samp} is the sampling frequency of 0.7Hz. The measured values of $\sigma_I\left(f_{stop}\right)$ are plotted in Figure 5-10. We see that most of the variance accumulates during the $1/f^2$ region, i.e., between 10^{-4}Hz and 10^{-5}Hz (a coupled of hours to 1 day). The maximum measured value of σ_I is $\sigma_{I,max} = 13.7$nA, corresponding to $f_{stop} \approx 10^{-7}$Hz. The standard deviation of the entire measured time-domain waveform is 15.4nA, which matches the measured value of $\sigma_{I,max}$ from the power spectrum to within 10%. Our methods therefore pass a sanity check: Parseval's theorem is indeed valid.

A note on the noise floor of our instrumentation. The accuracy of our measurement is set by that of the Keithley source meter. This instrument was set to its "high-

[17]Simulations also show that the expected value of output resistance is $r_o = 3.4\text{M}\Omega$, much larger than $R_L = 100\text{k}\Omega$, allowing its effects to be ignored.

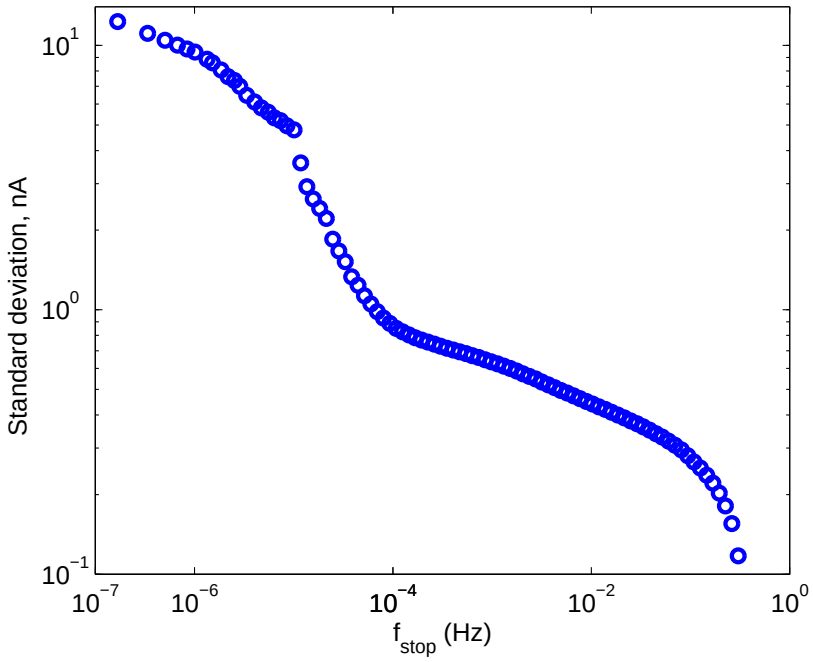


Figure 5-10: Measured standard deviation in the drain current as a function of f_{stop}, the lowest frequency of interest.

accuracy" mode, where it averages over 10 power line cycles for each measurement and displays 6.5 significant digits. This mode corresponds to a noise floor of approximately ±10pA, orders of magnitude lower than the standard deviation of the measured flicker noise. However, according to the instrument manual, the reading is only guaranteed to be approximately ±4nA *on an annual timescale*. Slow drifts in the values of electrical components within the instrument may be responsible for this decrease in accuracy over long time scales (hence explaining the need for periodic re-calibration of the instrument). However, no overall linear trend is visible in the time-domain data, indicating that component ageing and instrument drift are not significant sources of error.

The power spectrum does not completely characterize a random process. For example, there are infinitely many types of white noise with different amplitude probability distributions [144]. Thermal noise is that particular variety of white noise that has a Gaussian probability distribution[18]. Numerous measurements of the power

[18]Objections to this statement usually involve appeals to the Central Limit Theorem (CLT).

spectral density of MOSFET flicker, or $1/f$ noise are available in the literature, but its amplitude distribution has received much less attention. The few measurements that are available were made in bulk semiconductors or discrete carbon resistors, and showed Gaussian distributions [13,26,119], as predicted theoretically [107]. Deviations from Gaussian behavior have, however, been found in small (μm scale) resistors [238] and thin resistive films [220]. The matter is worth investigating because the power spectrum alone does not completely characterize a random process unless it is stationary, ergodic and Gaussian [144, 304]. In particular, the amplitude distribution is important for evaluating reliability.

Figure 5-11 shows the measured probability distribution of the amplitude of the drain current noise. The distribution is asymmetric, with a long negative tail. It also contains several distinct peaks (according to a rough visual estimate, 15-20). The data is well fit by a bi-exponential distribution, i.e.

$$P(\Delta I) = P(0) \exp\left(-|\Delta I|/I_{nat}\right) \tag{5.131}$$

where $P(\Delta I)$ is the probability that the drain current has a value $\overline{I} + \Delta I$, and I_{nat} is a constant. The best fit values of I_{nat} are 8.9nA for $I > 0$ and 13.3nA for $I < 0$.

Initially, temperature fluctuations in the environment were suspected of producing the deviations from $1/f$ behavior visible in the power spectra. However, the two identical resistors of value R_L used in the calibration experiment should vary in the same way with temperature, making the output voltage temperature-insensitive. As a result $P_R(f)$ should not, to first order, exhibit any temperature dependence. We therefore decided to check whether slow fluctuations in the power supply voltage V_{DD} were responsible for part of the measured noise. At frequencies that are slow enough for the capacitors to be ignored, a change ΔV_{DD} in V_{DD} causes the drain voltage given by

It must be remembered that the CLT is only applicable to the sum of independent, identically-distributed random variables with finite variance. The latter requirement is relaxed in generalized versions of the CLT due to Gnedenko and Kolmogorov. However, the attractor distribution in such cases is not necessarily Gaussian, but one of the family of Lévy α-stable distributions. The parameter α lies between 0 and 2, with the Gaussian corresponding to $\alpha = 2$.

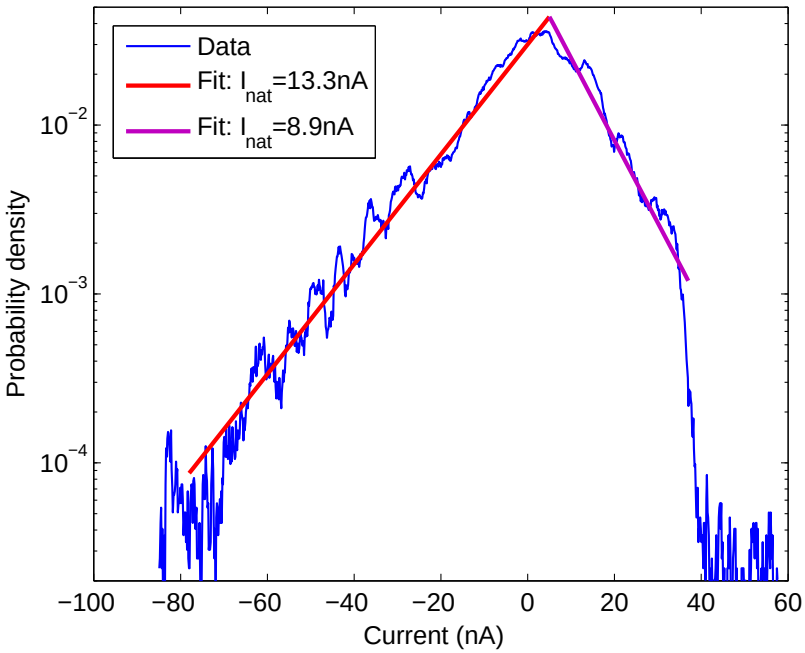


Figure 5-11: Probability distribution of the amplitude of the drain current noise as measured from time-domain data. A fit to an asymmetric bi-exponential distribution is also shown.

$$\Delta V = \Delta V_{DD}\left[1 - g_m R_L\left(\frac{V_G}{V_{DD}}\right)\right] \tag{5.132}$$

where V_G is the average gate voltage. With $V_G/V_{DD} = 0.6$, the equation above predicts that $\Delta V \approx -2.2\Delta V_{DD}$. Figure 5-12 shows the measured time-domain voltage generated by the so-called "DC" power supply in a 8-day period. The waveform has a standard deviation of 0.37mV and displays a pronounced daily periodicity. Assuming that the magnitude of the gain between ΔV and ΔV_{DD} is 2.2, the standard deviation in drain current caused by power supply fluctuations is approximately 8.1nA. From Figure 5-10, the total standard deviation in the drain current for $f_{stop} = 1\mu$Hz is about 9nA. Therefore the power supply is responsible for about 90% of the measured noise amplitude, and also the bi-exponential (Laplacian) shape of the amplitude distribution! It is interesting that the power-supply voltage is bi-exponentially and not normally distributed. The reason for this behavior is unclear, but we note that such distributions arise naturally in birth-death processes (continuous time Markov chain

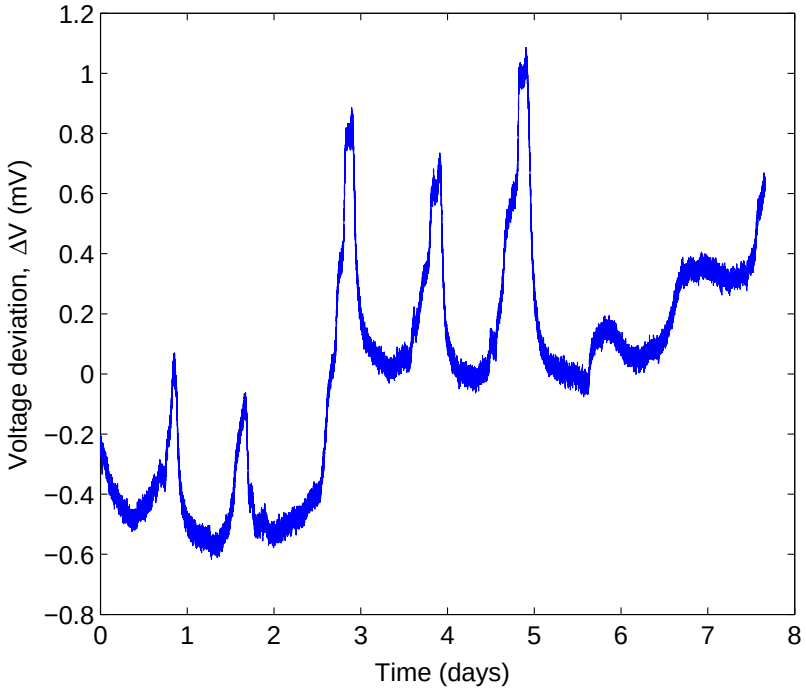


Figure 5-12: Measured time-domain voltage generated by the DC power supply.

models) and other queueing problems [100].

In order to remove the effects of power supply fluctuations, we high-pass filtered the time domain data obtained from the test setup and the load resistor alone. The resultant amplitude distributions, both of which are almost perfectly Gaussian, are shown in Figure 5-13. Here $N(\mu, \sigma)$ denotes the Gaussian (normal) distribution with mean μ and standard deviation σ (both in nA). The power spectral densities after high-pass filtering both display nearly ideal $1/f$ behavior; they are shown in Figure 5-14.

In order to verify that the time-domain samples came from a stationary random process we divided them into $M = 20$ time intervals. The mean square value $\overline{x_i^2}$ of the samples in each interval was calculated, where $0 < i \leq M$. The sequence of $\overline{x_i^2}$ values was subjected to the runs test, a standard non-parametric test of randomness. The sequence passed the test with a probability value $p \approx 1$, indicating that the random process is stationary with high probability. We also found that the distributions shown in Figure 5-13 are robust to the exact cutoff frequency and shape of the high-pass

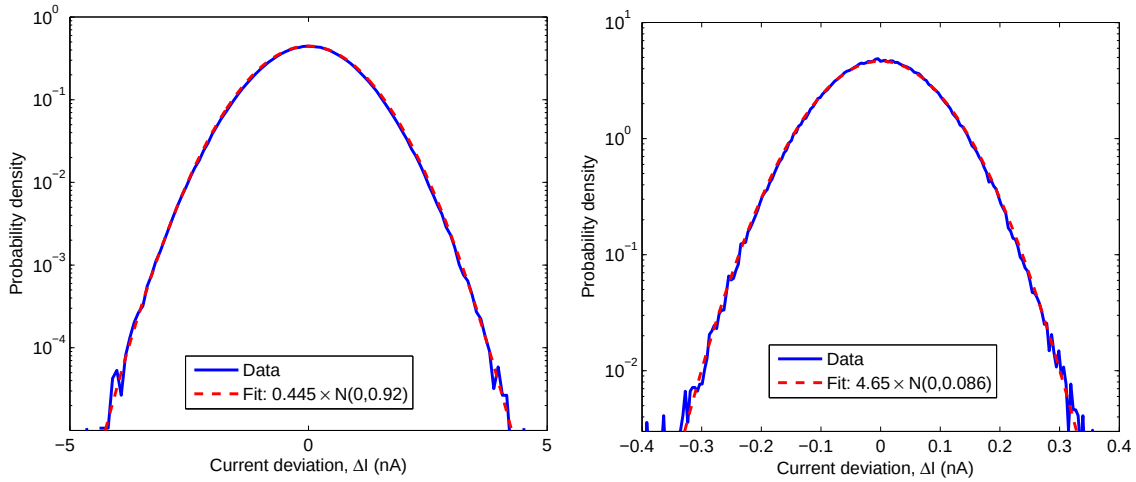


Figure 5-13: Measured amplitude distributions of the noise produced by the entire test setup (left) and load resistor alone (right) after high-pass filtering. We used fourth-order Butterworth filters with cutoff frequencies equal to 0.5mHz and 1mHz, respectively. Fits to Gaussian (normal) distributions are also shown.

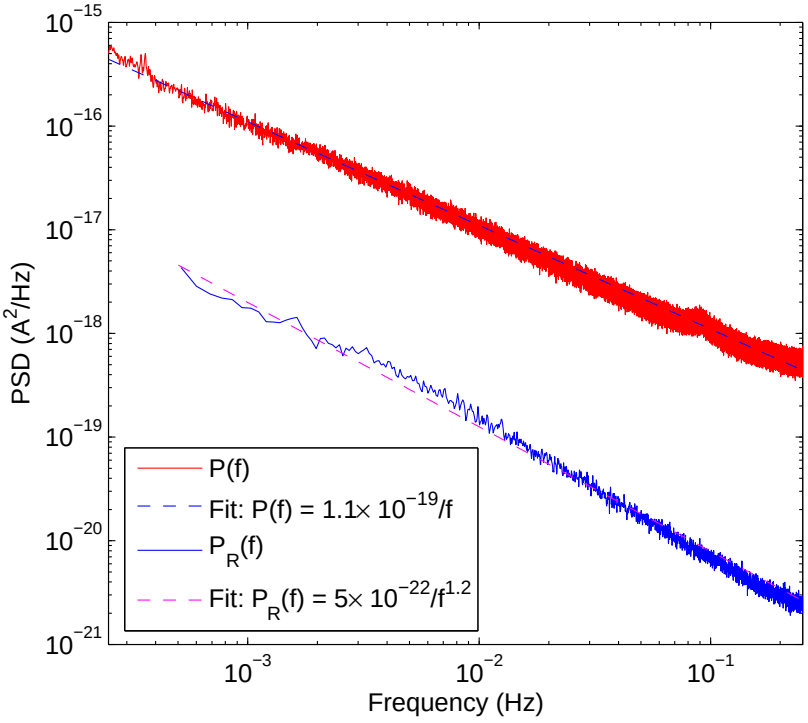


Figure 5-14: Measured power spectral densities of the entire test setup and load resistor alone after high-pass filtering. Fits to power-law spectra are also shown.

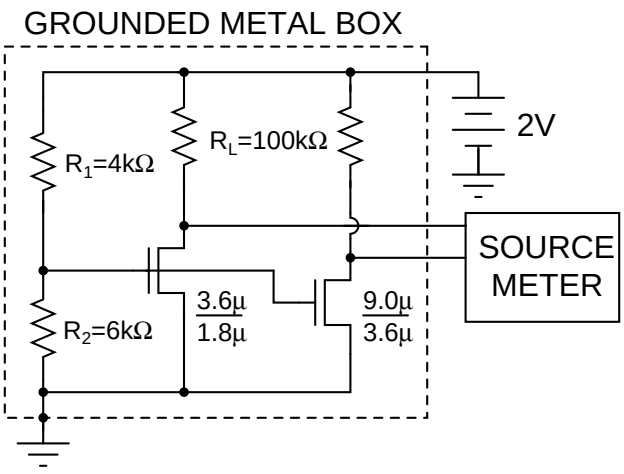


Figure 5-15: Modified test setup for measuring MOSFET flicker noise. The DC power supply has been replaced by a lead-acid battery (nominal output voltage = 2V).

filters used for pre-whitening and removal of power-supply fluctuations. In fact they persist even if the pre-whitening filter is omitted altogether. It therefore appears to be a fundamental feature of the random process that generates the noise. We conclude that the flicker noise processes in both the transistor and resistor have stationary amplitude distributions that are Gaussian in nature.

5.13.3 More Experimental Measurements

As described in the previous section, noise from the DC power supply limited our measurement of $1/f$ noise to frequencies greater than approximately 1mHz. We therefore decided to replace the power supply with a battery. In general, large batteries are extremely "quiet": they produce much less noise current than power supplies [139]. In order to facilitate comparison with our previous results, obtained with $V_{DD} = 2V$, we used a single lead-acid cell (Hawker Cyclon, size X, 5Ah or size BC, 25Ah) with a fully-charged open circuit voltage of 2.14V. We also decided to simultaneously measure the drain current of two NMOS transistors, with different W/L ratios, laid out next to each other on the same die and sharing the same gate voltage. The modified experimental setup is shown in Figure 5-15.

Figure 5-16 shows 40 days of measured time-domain data. The DC component of each voltage has been removed. The 9.0μm/3.6μm transistor carries an average

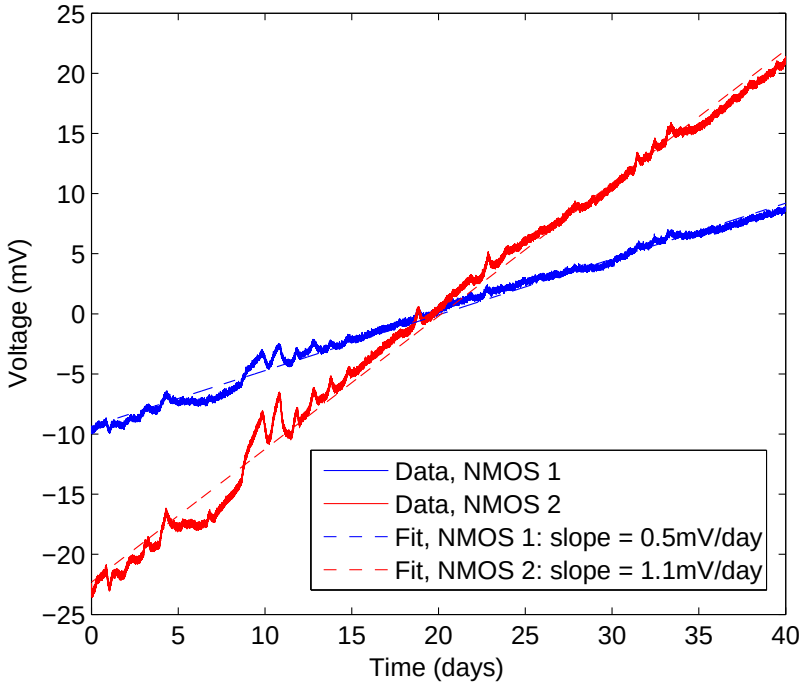


Figure 5-16: Measured time-domain voltages from the modified test setup, showing the effects of battery droop.

current of $\overline{I_1} = 5\mu$A, while the 3.6μm/1.8μm transistor carries an average current of $\overline{I_2} = 10\mu$A. The linear decrease of both output voltages with time is because of droop in the battery voltage, estimated at approximately 0.5mV/day. This droop is removed before further analysis.

The total current being drawn from the battery is approximately 220μA, leading to an estimated battery life of 950 days. In practice the experiment must be stopped much earlier since the gate voltage V_G also droops: $\Delta V_G \approx 0.6\Delta V_{DD}$. As a result the average drain current of both transistors decreases slowly with time. We decided to limit this decrease to 10% of the initial value to avoid non-stationarity in the noise process. It is easy to show that the allowable battery droop is limited by this requirement to

$$\Delta V_{DD} = 0.6 V_L \frac{\Delta I}{\overline{I}} \tag{5.133}$$

Using $\Delta I/\overline{I} = 0.1$ (i.e., 10%) and $V_L = 180$mV gives us $\Delta V_{DD} = 30$mV. At the

measured droop rate of 0.5mV/day this corresponds to an experimental time-span of 60 days.

The main goal of measuring two transistor currents simultaneously was to eliminate the effects of temperature fluctuations. The total drain current fluctuation ΔI can be divided into a temperature-independent part ΔI_n and a temperature-dependent part ΔI_T, i.e., $\Delta I = \Delta I_n + \Delta I_T$. However, the transistors are located close to each other on the same die and share the same temperature. Therefore the ΔI_T term is correlated across the two transistors, while the ΔI_n term, which is caused by thermal and flicker noise, is not.

MOSFET currents depend polynomially on temperature via two parameters: threshold voltage and carrier mobility in the channel. Since temperature fluctuations ΔT are much smaller in magnitude than the average room temperature of 298K, both polynomials can be linearized. As a result, ΔI_T is linearly related to ΔT, i.e.,

$$\Delta I_{Ti} = \beta_i \Delta T \tag{5.134}$$

where i is 1 or 2 depending on which transistor is being referred to, and β_i is bias-dependent. We now define

$$\alpha = \frac{\Delta I_{T2}}{\Delta I_{T1}} \tag{5.135}$$

where $\alpha \equiv \beta_2/\beta_1$. We can therefore eliminate the effects of temperature fluctuations by forming the linear combination $\Delta I' = \Delta I_2 - \alpha \Delta I_1$. Figure 5-17 shows the standard deviation σ of this combination as a function of α. In subsequent measurements we use $\alpha = 2.2$ since, as shown in the figure, this choice minimizes the standard deviation. The existence of an unique minimum in σ indicates that our assumption of a linear dependence of drain current on temperature was correct. The minimum corresponds to the true value of α, i.e. the value that cancels as much of the correlated fluctuations as possible. For $\alpha = 2.2$, the standard deviation of $\Delta I'$ is only 1.46nA, compared to 4.86nA and 10.18nA for ΔI_1 and ΔI_2, respectively. Most of this excess variance accumulates at frequencies below 0.1mHz.

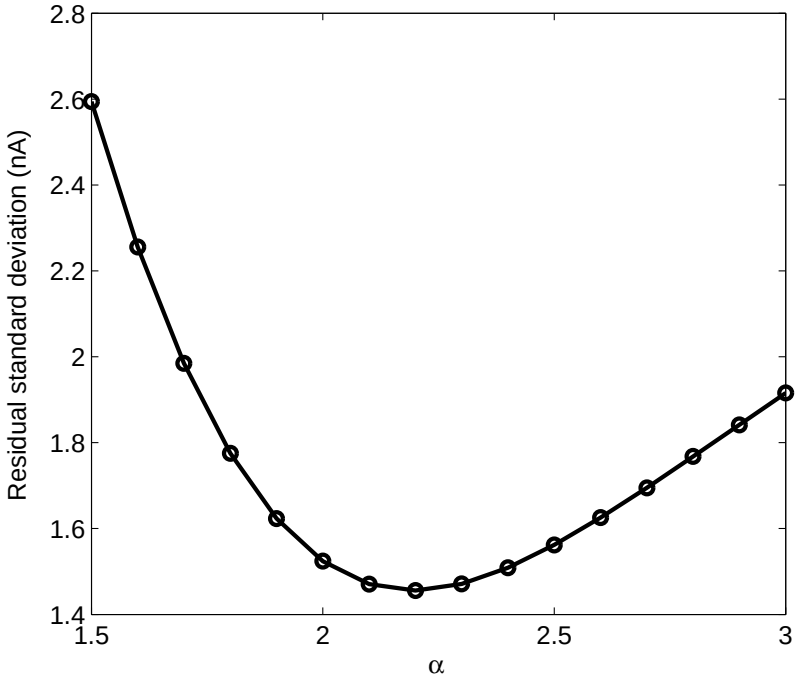


Figure 5-17: Residual standard deviation of the measured flicker noise as a function of the parameter α.

Figure 5-18 shows the amplitude distribution of $\Delta I'$ for $\alpha = 2.2$. The amplitude distribution is well described by a Gaussian with zero mean. Figure 5-19 shows the power spectral density (PSD) of $\Delta I'$ for $\alpha = 2.2$. We see that the spectrum is very nearly an uniform power law with a slope of 1.2 down to approximately 1μHz, with no traces of diurnal periodicity. Such periodicity, caused by temperature fluctuations, reappears if our cancelation scheme is not used, i.e. ΔI_1 and ΔI_2 are analyzed separately.

Figure 5-20 shows the measured PSD (after removal of temperature fluctuations) obtained by repeating the experiment five times with the same setup. The five experiments lasted 59, 78, 40, 30 and 73 days, respectively. We see that the plots line up very well with each other, indicating that the PSD remained essentially stationary over the 280 days for which data was taken. The lowest frequency shown on these plots is approximately 0.5μHz. The best-fitting power law spectrum has a slope of -1.2, and is given by $1.6 \times 10^{-20}/f^{1.2}\mathrm{A}^2/\mathrm{Hz}$.

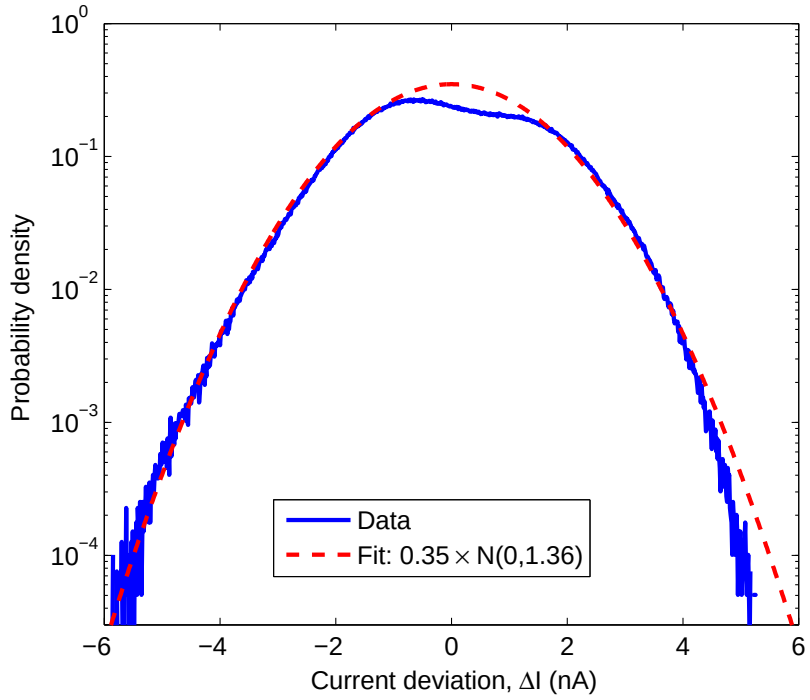


Figure 5-18: Measured amplitude distribution of the flicker noise for $\alpha = 2.2$. The best-fitting normal distribution is also shown.

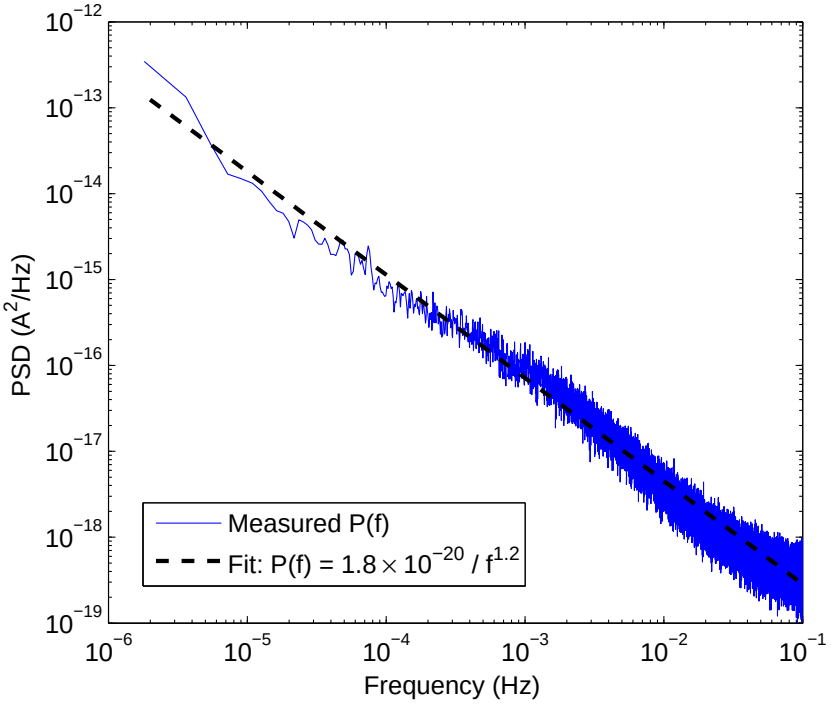


Figure 5-19: Measured PSD of the flicker noise for $\alpha = 2.2$. The best-fitting power-law spectrum is also shown.

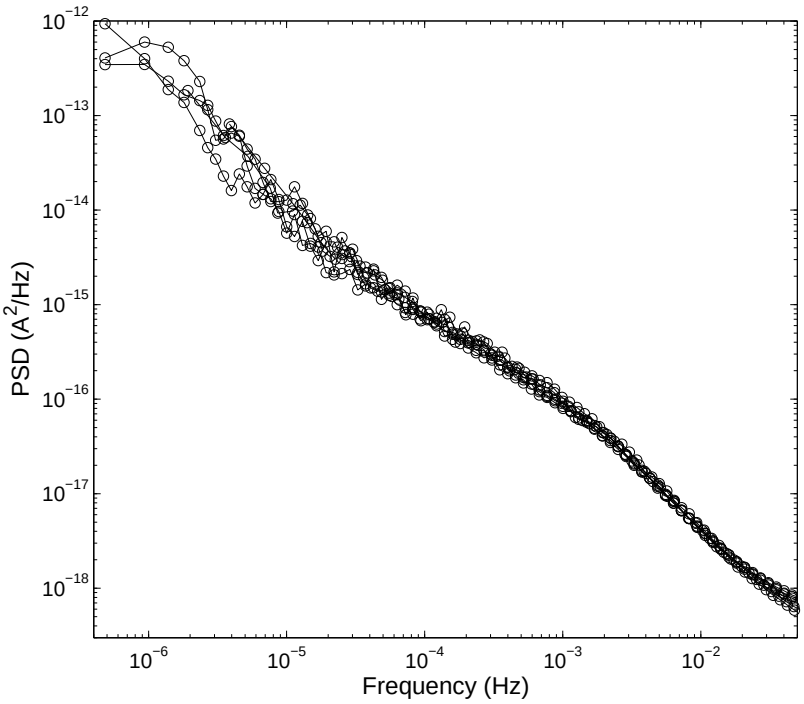


Figure 5-20: Measured PSD of the flicker noise from five independent experimental runs, each lasting an average of 56 days.

5.14 Noise in Oscillators

The basic components needed for an autonomous oscillator are passive frequency-selective element(s) and active energy restoration element(s). Real oscillators follow closed, stable paths in state space, known as *limit cycles*. A continuous-time dynamical system needs at least two state variables (energy storage elements) to exhibit regular limit-cyle oscillations, and three or more state variables to exhibit srange attractors and chaos. However, discrete-time dynamical systems in only one variable (1D maps) can exhibit both regular oscillations and chaos. The frequency (rate of completion of limit cycles) of a real oscillator is not perfectly stable with time, a phenomenon known as *phase noise*.

5.14.1 Why is Phase Noise Bad?

As a result of phase noise, the spectrum of an oscillator is not a delta function, but has finite width. Phase noise is measured in dBc/Hz units: power present in a 1Hz bandwidth centered about an offset of $\Delta\omega$ from the carrier ω_0, normalized to the carrier power. Phase noise is an important problem in wireless systems [236]. Phase noise in the Local Oscillator (LO) creates adjacent channel interference in wireless systems via "reciprocal mixing", as shown in Figure 5-21. In this process, the LO mixes with a strong interfering symbol close to the desired frequency band. As a result, some of the power in the interfering signal appears in-band after down-conversion.

The maximum allowable amount of LO phase noise depends on the modulation scheme, data rate, wireless environment and other system parameters. Some typical numbers are shown in Table 5.1. Here NADC stands for "North American Digital Cellular", while the 802.15.4 specification assumes the co-existence of other standards in the 2.4GHz ISM band.

Standard	Center Frequency	Offset	Phase Noise
NADC	869-894MHz	60kHz	< -115dBc/Hz
802.15.4	2.4GHz	1MHz	< -110dBc/Hz
802.11b/g	2.4GHz	1MHz	< -115 to < -125dBc/Hz
802.11a	5.3GHz	1MHz	< -110dBc/Hz

Table 5.1: Typical LO phase noise specifications in wireless systems

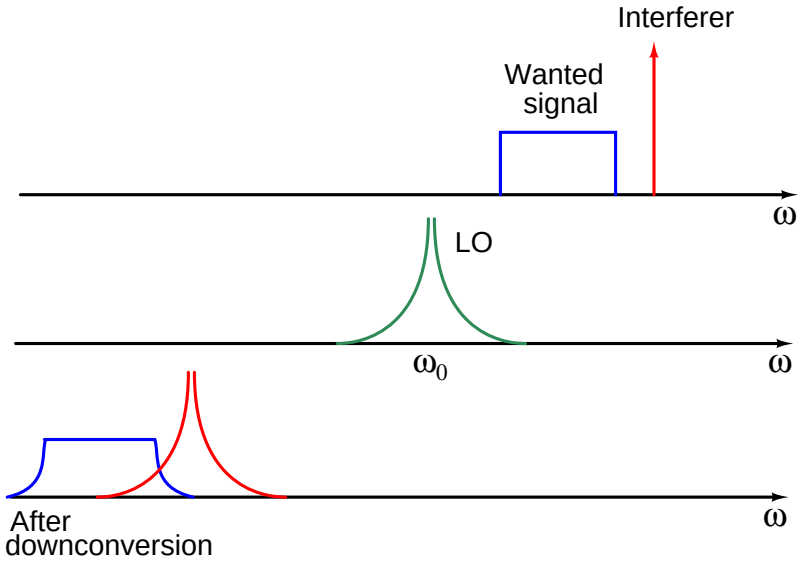


Figure 5-21: Reciprocal mixing in wireless systems.

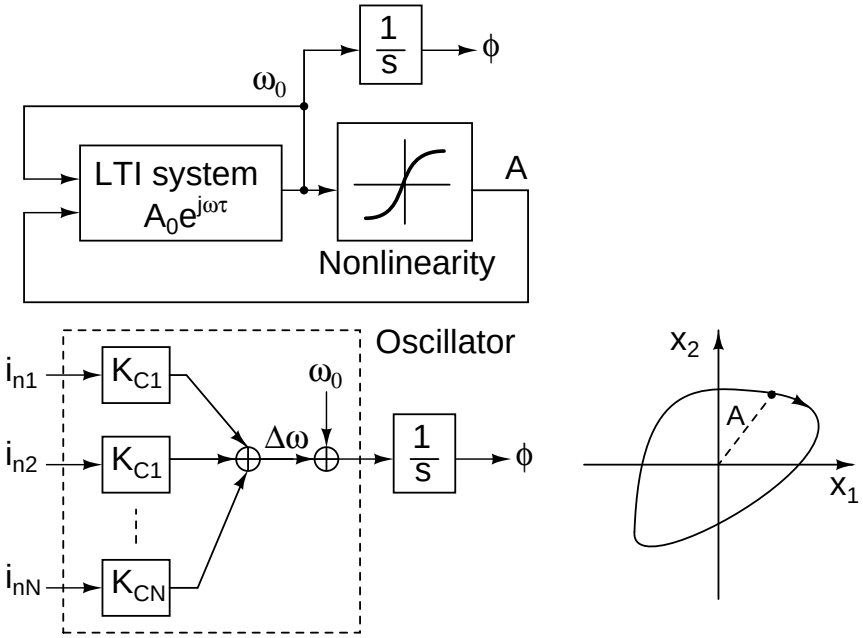


Figure 5-22: LTI oscillator model used in this section.

5.14.2 LTV Theory

A real oscillator can be modeled as a controlled oscillator with many control inputs and two outputs: amplitude and phase, as shown in Figure 5-22. We will focus on effects on phase, since there is no feedback loop to regulate it. Oscillator frequency variability can be modeled by assuming that each noise source (i_n or v_n) can change the oscillator frequency via a control gain K_C:

$$\begin{aligned}\omega(t) &= \omega_0 + \sqrt{\sum_{j=1}^{N} K_{Cj}^2 i_n(t)^2} \\ &= \omega_0 + \Delta\omega \end{aligned} \tag{5.136}$$

Frequency errors are integrated into phase errors, so if $\Delta\omega$ is white $\Delta\phi$ will have a $1/f^2$ power spectrum.

Noise sources that affect oscillators can be divided into two types. The first consists of fundamental sources, such as resistors and active elements, like transistors, within the oscillator. These sources generate white noise (thermal/shot) and colored

noise (flicker). In addition, many oscillators are subject to external, non-fundamental sources of interference, typically consisting of a mixture of broadband and narrowband sources. Such sources are particularly important in mixed-signal systems, where the clock is a prominent source of narrowband noise, and are usually grouped into substrate and power-supply noise terms.

The well-known Linear-Time-Varying (LTV) theory of oscillator phase noise was first developed by Hajimiri & Lee in 1998 [103, 159]. They assumed that the transfer function from noise current to phase is linear (superposition still applies) but periodically-time-varying (frequency conversion occurs). Conceptually, we can think of the oscillator as having a periodically-time-varying control gain $K_C(\omega_0\tau)$, where $0 < \omega_0\tau < 2\pi$. Noisy devices either act as current sources i_n in parallel with capacitors that have charge swing amplitude $q_{max} = CV_0$, or voltage sources v_n in series with inductors that have flux swing amplitude $\phi_{max} = LI_0$.

As before, noise impulses are integrated into step changes in phase. However, the sizes of the steps are now allowed to depend on τ, the time when they occur, giving us a time-varying impulse response $h_\phi(t,\tau)$. The *Impulse Sensitivity Function* (ISF), Γ defines the normalized height of the step response, as follows

$$\begin{aligned} h_\phi(t,\tau) &= \frac{\Gamma(\omega_0\tau)}{q_{max}} u(t-\tau) \\ \phi(t) &= \int_{-\infty}^{\infty} h_\phi(t,\tau) i_n(\tau) d\tau \end{aligned} \tag{5.137}$$

Figure 5-23 shows why a time-varying theory is necessary: changes in capacitor voltage due to noise result in different amounts of phase change, depending on where in the limit cycle the noise is applied. As a result, Γ is not constant but is periodically time-varying. As power consumption increases, so does the charge swing q_{max}, which causes the phase noise to decrease. Thus phase noise, like other forms of noise, can be reduced by burning more power. Phase noise can also be reduced by increasing the quality factor of the drequency-selective system within the oscillator, since, for a given amount of power consumption, increasing Q causes q_{max} to increase.

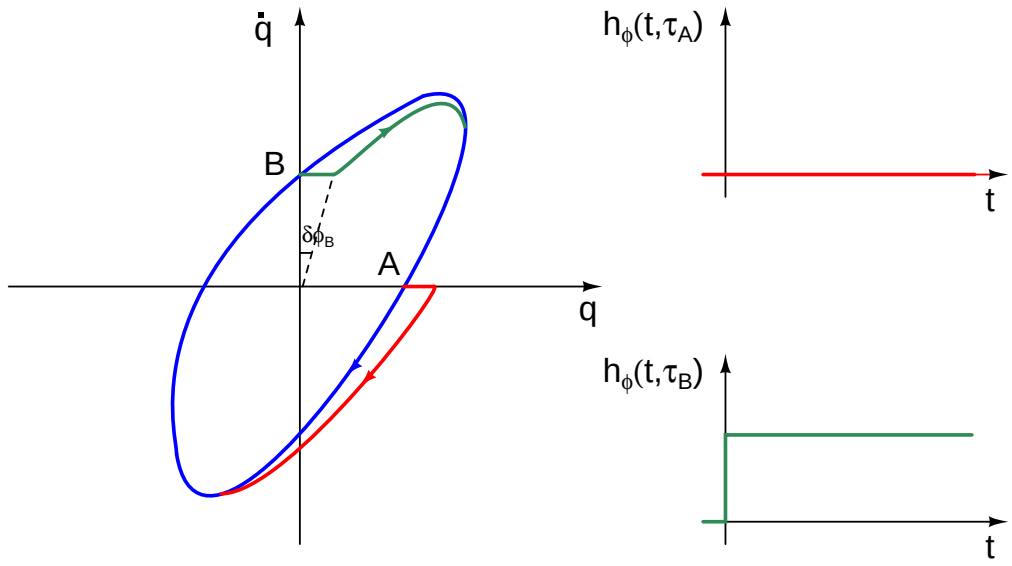


Figure 5-23: Noise impulses injected into an oscillator result in a periodically time-varying amount of phase change.

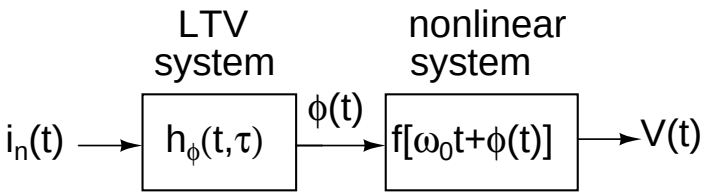


Figure 5-24: Conversion of phase noise into voltage is a nonlinear process.

The actual output of an oscillator is not phase, but voltage. The transfer function from phase to voltage is nonlinear, unlike that between noise amplitude and phase. In fact, as shown in Figure 5-24, noise in the phase, $\phi(t)$, causes phase-modulation (PM) of the oscillator's voltage output $V(t)$.

$$V(t) = A(t)f\left[\omega_0 t + \phi(t)\right] \tag{5.138}$$

Since the ISF is periodic, we can write it as a Fourier series:

$$\Gamma\left(\omega_0\tau\right) = \frac{c_0}{2} + \sum_{n=1}^{\infty} c_n \cos(n\omega_0\tau + \theta_n) \tag{5.139}$$

In addition, since oscillators are large-signal circuits, active noise sources are usually *cyclostationary*, i.e., produce periodically-time-varying amounts of noise during

one oscillation cycle. As a result, we can define a Noise Modulation Function (NMF), $\alpha(\omega_0\tau)$, such that

$$i_n(t) = i_{n0}(t)\alpha(\omega_0\tau) \tag{5.140}$$

where $i_{n0}(t)$ is assumed to be a white, stationary noise process. Cyclostationary noise can be modeled by replacing the ISF with an "effective ISF function" given by $\Gamma_{eff}(x) = \Gamma(x)\alpha(x)$, $x = \omega_0\tau$, and the cyclostationary noise source $i_n(t)$ with the stationary source $i_{n0}(t)$. From now on, we shall assume that Γ refers to the effective ISF, unless mentioned otherwise.

In general, the ISF is proportional to the time derivative of the oscillator voltage waveform. For sinusoidal oscillators, therefore, the ISF is 90° out of phase with the voltage, becoming exactly zero at voltage extrema. Figure 5-25 shows a typical sinusoidal oscillator, the Colpitts oscillator, and Figure 5-26 shows approximate plots of its waveforms, ISF and NMF. We see that the drain current of the transistor, and the noise produced by it, is zero except close to the negative peak of the drain voltage. At this location, however, the ISF is zero. Therefore the ISF and NMF are out of phase with each other, reducing the rms value of Γ_{eff} and the amount of phase noise for a given power consumption. We also note that there is an optimum value of the capacitive divider ratio n in this circuit for minimizing phase noise. As n decreases, the conduction angle decreases, decreasing Γ_{eff}, but the oscillation amplitude also decreases, decreasing q_{max}, which increases phase noise. Therefore there is an optimum value of n; it is often found to be in the vicinity of 4.

Figure 5-27 shows waveforms, ISF and NMF for a typical non-sinusoidal oscillator, a ring oscillator. In contrast to the Colpitts oscillator, Γ and α reach maxima at the same points in the cycle, which increases Γ_{eff} and is bad for phase noise.

Consider applying identical, correlated noise sources, such as supply voltage fluctuations, to all nodes of a N-stage ring oscillator. The resultant ISFs differ only in phase by multipes of $2\pi/N$. By using superposition, we see that

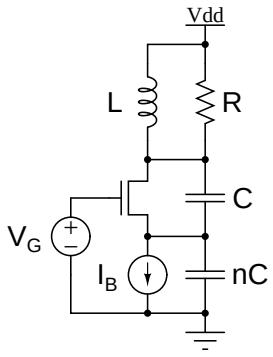


Figure 5-25: Circuit diagram of the Colpitts oscillator.

$$\phi(t) = \frac{1}{q_{max}} \int_{-\infty}^{t} i(\tau) \left[\sum_{n=0}^{N-1} \Gamma \left(\omega_0 \tau + \frac{2\pi n}{N} \right) \right] d\tau \tag{5.141}$$

The term in brackets in the equation above is the sum of N symmetric phasors and disappears except at DC and integer multiples of $N\omega_0$. Thus we can reduce the effects of substrate and supply noise by making their effects identical at every stage [104,116]. Practically, this goal can be reached by laying out stages symmetrically with respect to power supply buses and placing identical, possibly dummy, loads on them.

Because of the time-varying nature of the ISF, frequencies around integral multiples of ω_0 are downconverted ("folded") to symmetric sidebands on either side of ω_0, becoming phase noise, as shown in Figure 5-28. This type of frequency conversion is not predicted by LTI theories of phase noise, such as Leeson's original work [160]. Frequency conversion causes a sinusoid $I_n \left[(n\omega_0 \pm \Delta\omega)\, t \right]$ to be weighted by c_n and appear as phase noise at an offset of $\pm\Delta\omega$ from ω_0. As a result of this conversion, the sideband power relative to the carrier is given by

$$P_{SBC}(\Delta\omega) = \left(\frac{c_n}{q_{max}} \frac{I_n}{4\Delta\omega} \right)^2 \tag{5.142}$$

We can use Parseval's theorem to define Γ_{rms}, the RMS value of the ISF, as follows

$$\frac{1}{2} \sum_{n=0}^{\infty} c_n^2 = \frac{1}{2\pi} \int_0^{2\pi} |\Gamma(x)|^2 dx = \Gamma_{rms}^2 \tag{5.143}$$

Since a white noise source i_n can be decomposed into a sum of sinusoids with

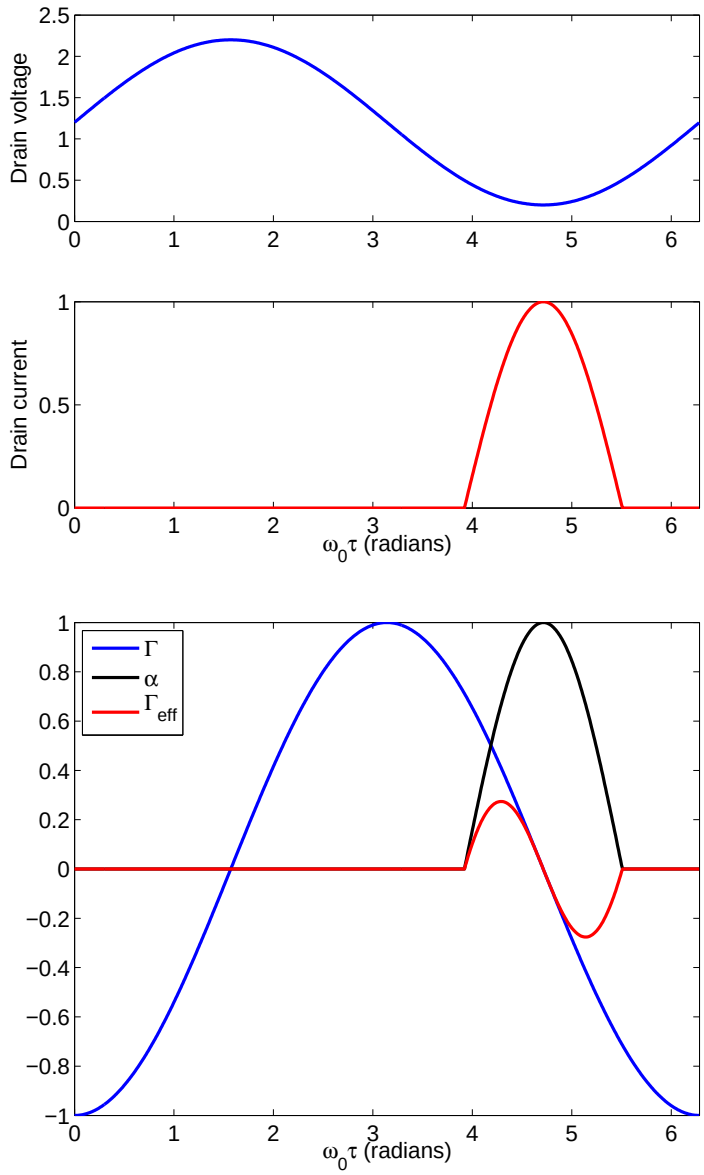


Figure 5-26: Colpitts oscillator phase noise properties, (top) typical drain voltage and current waveforms and (bottom) typical impulse sensitivity and noise modulation functions.

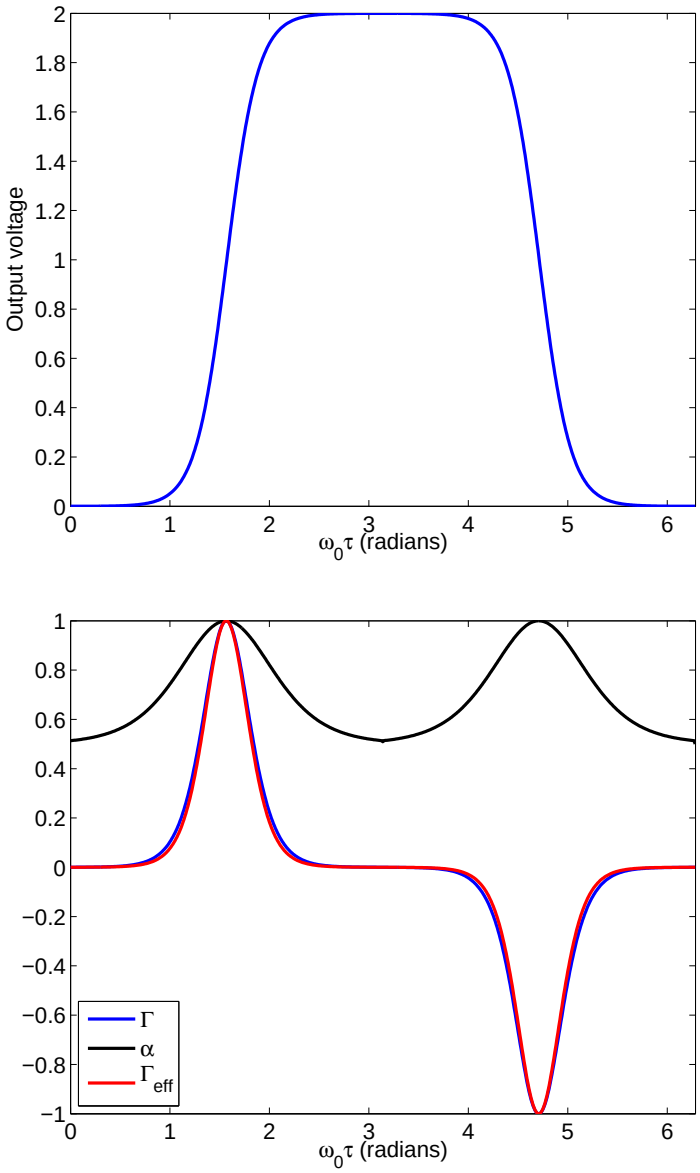


Figure 5-27: Ring oscillator phase noise properties, (top) typical output voltage waveform and (bottom) typical impulse sensitivity and noise modulation functions.

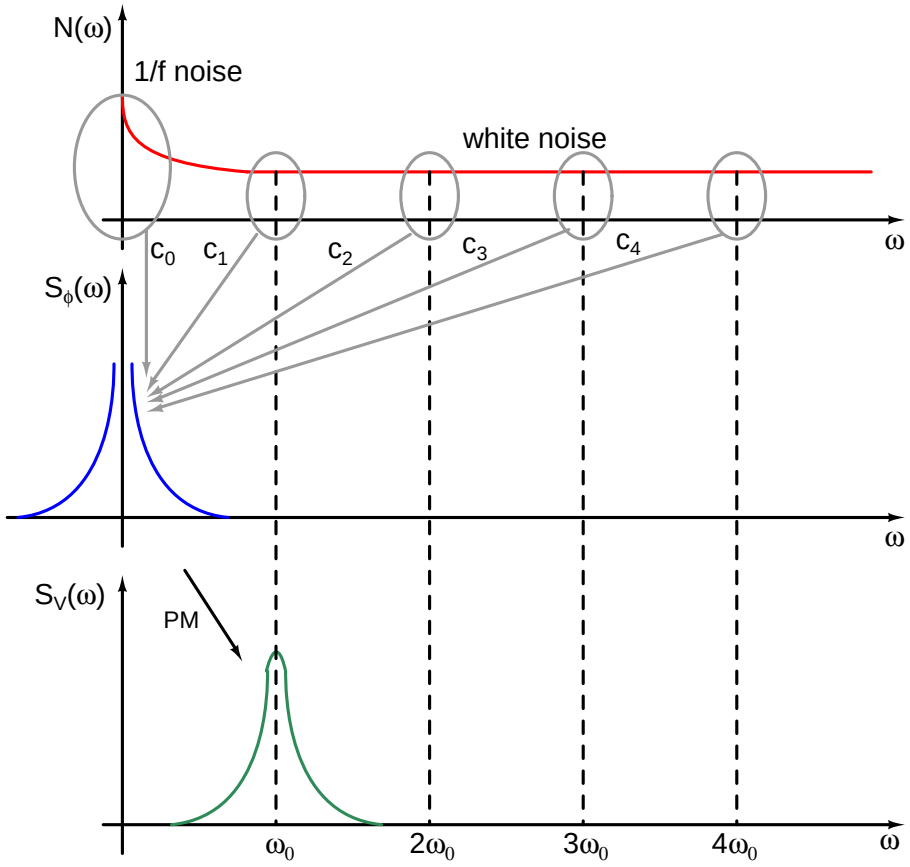


Figure 5-28: Noise present around integer multiples of the oscillation frequency ω_0 appear as phase noise sidebands around ω_0.

random phases, we can use superposition to find the phase noise caused by i_n. Thus the sideband power relative to the carrier is given by

$$\begin{aligned} L(\Delta\omega) &= \frac{\sum_{n=0}^{\infty} c_n^2}{q_{max}^2} \frac{\overline{i_n^2}/\Delta f}{4(\Delta\omega)^2} \\ &= \frac{\Gamma_{rms}^2}{q_{max}^2} \frac{\overline{i_n^2}/\Delta f}{2(\Delta\omega)^2} \end{aligned} \tag{5.144}$$

Flicker noise near DC is upconverted by the c_0 term of the ISF, creating $1/f^3$ frequency dependence in phase noise close to the carrier frequency. We define $\Gamma_{dc} = c_0/2$ to be the average (DC) value of the ISF. It can now be shown that the $1/f^3$ corner frequency is given by

$$\omega_{1/f^3} = \omega_{1/f} \left(\frac{\Gamma_{dc}}{\Gamma_{rms}} \right)^2 \tag{5.145}$$

In order to reduce ω_{1/f^3} we can make the waveform shape symmetric to reduce Γ_{dc}. In ring oscillators, for example, relative NMOS and PMOS sizes can be adjusted to vary rise and fall times. Differential *half circuits* must be symmetric to reduce ω_{1/f^3}. Linear loads (like resistors) can also be used to increase waveform symmetry.

5.14.3 Amplitude Noise

Because of the amplitude-feedback loop present inside practical oscillators, amplitude (unlike phase) perturbations die away with some time constant, typically $\approx Q/\omega_0$. However, the amplitude impulse response $h_A(t,\tau)$ is still time-varying, so we can define another periodic sensitivity function Λ, as follows:

$$\begin{aligned} h_A(t,\tau) &= \frac{\Lambda(\omega_0\tau)}{q_{max}} d(t-\tau) \\ A(t) &= \int_{-\infty}^{\infty} h_A(t,\tau) i_n(\tau) d\tau \end{aligned} \tag{5.146}$$

If we assume a first-order amplitude-feedback loop with $\tau = Q/\omega_0$, we get an expo-

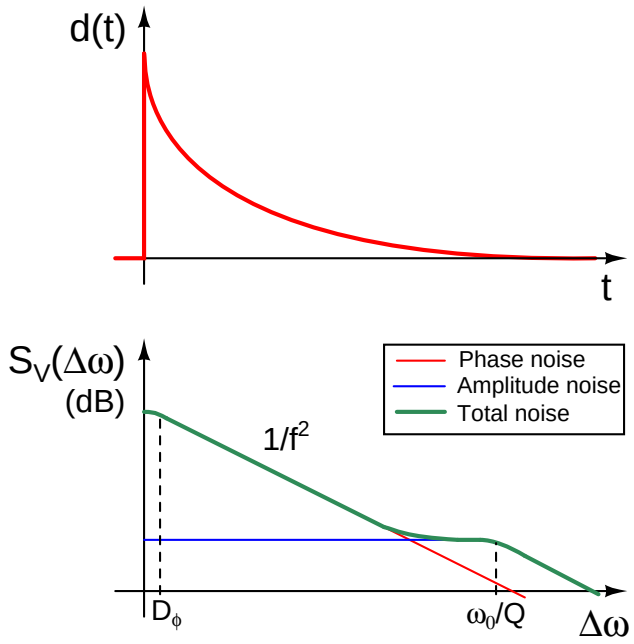


Figure 5-29: The total voltage PSD of an oscillator consists of contributions from both phase and amplitude noise.

nential impulse response, i.e., $d(t) = e^{-t/\tau}u(t)$. In this case the PSD due to amplitude noise is the squared magnitude of a first-order lowpass filter, i.e., a Lorentzian:

$$S_{V,amp}(\Delta\omega) = \frac{\Lambda_{rms}^2}{q_{max}^2}\left(\frac{\overline{i_n^2}/\Delta f}{\frac{\omega_0^2}{Q^2} + (\Delta\omega)^2}\right) \tag{5.147}$$

We see that the amplitude-regulating negative feedback loop converts the pure integrator present in phase noise into a low-pass filter with finite low-frequency gain. The total voltage PSD is given by the sum of phase and amplitude noise terms, i.e., $S_{V,tot} = S_{V,phase} + S_{V,amp}$. As shown in Figure 5-29, it consists of a part near the carrier dominated by phase noise. Further away from the carrier, the power spectrum follows the Lorentzian shape of amplitude noise, consisting of a "pedastal", or flat portion, and an eventual $1/f^2$ roll-off.

5.14.4 Phase Noise as a Diffusive Process

If we consider only white noise sources, phase noise sidebands are $\propto 1/f^2$. Diffusive processes like Brownian motion also produce $1/f^2$ spectra. In fact, we can view oscilla-

tor phase as diffusing along the limit cycle with periodically-varying diffusion constant $D_\phi(\omega_0\tau)$. We now define the cycle-averaged diffusion constant $D_\phi \equiv \overline{D_\phi(\omega_0\tau)}$. If all noise sources are white, frequency deviations $\Delta\omega$ are also white, so the autocorrelation of such deviations is a delta function:

$$R_{\Delta\omega}(\tau) = 2D_\phi\delta(\tau) \Rightarrow \sigma^2_{\Delta\omega} = 2D_\phi \tag{5.148}$$

Since phase $\phi(t)$ is a pure integral of frequency, we see that its variance increases without bound, just like the variance in position of a random walker increases linearly with time:

$$\sigma^2_\phi = \sigma^2_{\Delta\omega} \times t = 2D_\phi t \Rightarrow S_\phi(\Delta\omega) = \frac{2D_\phi}{(\Delta\omega)^2} \tag{5.149}$$

However, what is the phase diffusion constant? To answer this question, we compare (5.144) with (5.149), and find that

$$D_\phi = \frac{\Gamma^2_{rms}}{4q^2_{max}}\frac{\overline{i^2_n}}{\Delta f} \tag{5.150}$$

We have $q_{max} = CV_0$ and $\overline{i^2_n}/(\Delta f) = 4kT/R$, so

$$\begin{aligned} D_\phi &= \frac{kT}{CV_0^2}\frac{\Gamma^2_{rms}}{RC} \\ &= \frac{1}{SNR}\frac{\omega_0}{Q_{eff}} \end{aligned} \tag{5.151}$$

Thus, we see that $D_\phi = BW/SNR$, where $BW = \omega_0/Q_{eff}$. Here BW and $Q_{eff} \equiv \omega_0 RC/\Gamma^2_{rms}$ are the open-loop bandwidth and effective quality factor of the frequency-selective element in the oscillator, respectively, and $SNR = CV_0^2/kT$ is the carrier to noise power ratio of the closed-loop oscillator.

We see that phase diffusion is a non-stationary "random walk". Perturbations in phase persist forever because phase is a pure integral of frequency, just like position is a pure integral of velocity. However, the voltage waveform $V_0\cos(\omega_0 t + \phi(t))$ is

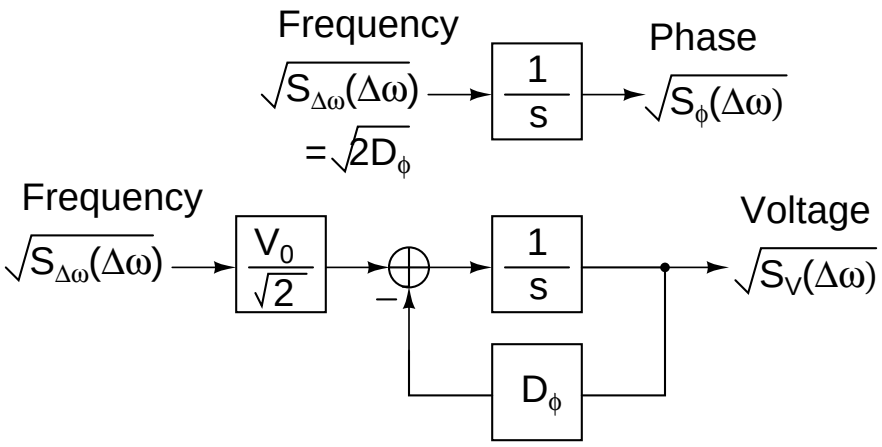


Figure 5-30: Frequency deviations in an oscillator are continually integrated into phase (top), but are only integrated into voltage for a time $\approx 1/D_\phi$ (bottom).

periodic: unlike phase, voltage diffuses in a closed domain. As a result, the probability distribution around the limit cycle feeds back on itself and relaxes to a stationary value with time constant $1/D_\phi$, converting the pure integrator into a leaky one, as shown in Figure 5-30. It can be shown [156] that the autocorrelation function of the oscillator voltage is given by

$$R_V(\tau) = \frac{V_0^2}{2} \exp\left(-D_\phi|\tau|\right) cos(\omega_0\tau) \tag{5.152}$$

The process shown in Figure 5-30 is similar to what happens within electrical circuits. Electrons diffusing through a series of components, like resistors and transistors, eventually have to return to the starting node to complete the loop and satisfy KCL. The relaxation time in this case is given by $\tau_r \approx L^2/2D$, where L is the total length diffused by electrons during one transit of the loop, and D is the diffusion constant.

When the limit cycle is circular (or, more generally, the surface of a hypersphere in phase space), the stationary probability distribution is uniform [267]. We now use the fact that the autocorrelation function and power spectral density of stationary signals are Fourier transform pairs. Again, the exponential decay in autocorrelation with time results in a Lorentzian power spectrum, given by

$$S_V(\omega) = \left(\frac{V_0^2}{2}\right) \frac{2D_\phi}{(\omega_0 - \omega)^2 + D_\phi^2} \tag{5.153}$$

The power spectrum in (5.153) is flat for frequency offsets less than D_ϕ, and then shows a second-order rolloff, as expected. A similar result holds for shot noise in circuits, which, as described earlier, is also caused by diffusion within closed domains (current loops, in this case). For circuits, D_ϕ is replaced by its equivalent $2D/L^2 \approx 1/\tau_r$, and $\omega_0 = 0$. Also, transistors have a unity-gain frequency $\omega_T \approx 1/\tau_r$, showing that the PSD of the noise is flat (white) for $\omega \ll \omega_T$, and rolls off for $\omega > \omega_T$. Intuitively, charge storage within transistors causes them to low-pass filter the noise at such high frequencies. In practical circuits, however, this behavior is almost never observed: the slowest node within each current loop, which sets the overall noise bandwidth for that loop, is usually much lower than ω_T. The total amount of noise can be found by combining noise from all loops in the circuit.

By integrating $S_V(\omega)$, we find that $\int_{-\infty}^{\infty} S_V(f)df = V_0^2/2$, the total output power. Thus the total power in frequency and time domains is identical, as required by Parseval's theorem. The effective bandwidth of the oscillator, i.e., D_ϕ, is $1/SNR$ of that of the frequency-selective element within it. Thus closing the positive feedback loop reduces bandwidth by a factor of SNR, a phenomenon known as *linewidth compression* [105]. For practical oscillators $SNR \gg 1$, i.e., D_ϕ is very small ($10^{-6}\omega_0$ to $10^{-9}\omega_0$), so $S_V(\omega_0 - \omega) \approx (V_0^2/2) \times S_\phi(\Delta\omega)$.

5.14.5 Jitter

Oscillator frequency instability shows up as phase noise in the frequency domain and as timing uncertainty, i.e., jitter, in the time domain. There are two main types of jitter: random, due to fundamental noise mechanisms present within the oscillator, and deterministic, due to external noise or interference. The former usually has a Gaussian probability distribution, while the latter is often non-Gaussian. Figure 5-31 compares the amplitude distribution function of thermal noise, which is Gaussian, with that of a typical source of external interference, namely a sinusoid. Sinusoids

have bimodal probability distributions, as shown in the figure.

Phase diffusion is the underlying cause of jitter. Assume that we know the precise phase of an oscillator at time $t = 0$. The variance in phase of the oscillator with white-noise inputs after a time ΔT is given by

$$\sigma_\phi^2 = 2D_\phi(\Delta T) \tag{5.154}$$

We can find timing variance can be found from phase variance easily by using the relationship $\sigma_{\Delta T} = \sigma_\phi/\omega_0$. Thus, we get

$$\sigma_{\Delta T} = \left(\frac{\sqrt{2D_\phi}}{\omega_0}\right)\sqrt{\Delta T} \equiv \kappa\sqrt{\Delta T} \tag{5.155}$$

where $\kappa = \left(\sqrt{2D_\phi}/\omega_0\right)$ is a constant. We see that $\sigma_{\Delta T} \propto \sqrt{\Delta T}$. However, in the presence of non-white (correlated) noise sources $\sigma_{\Delta T}$ can grow faster than $\sqrt{\Delta T}$. In certain cases, however, such as in a phase-locked loop, $\sigma_{\Delta T}$ ceases to grow with time and reaches a finite steady-state value [197]. In such cases a stationary phase power spectrum S_ϕ exists, and we can relate it to the jitter:

$$\begin{aligned}\sigma_{\Delta T}^2 &= \frac{1}{\omega_0^2}\overline{(\phi(t+\Delta T) - \phi(t))^2} \\ &= \frac{2}{\omega_0^2}\left[R_\phi(0) - R_\phi(\Delta T)\right]\end{aligned} \tag{5.156}$$

where R is the aurocorrelation function. Since autocorrelation and power spectral density are Fourier transform pairs, we can write

$$\sigma_{\Delta T}^2 = \frac{8}{\omega_0^2}\int_0^\infty S_\phi(f) sin^2(\pi f \Delta T) df \tag{5.157}$$

5.14.6 Jitter in Phase-Locked Loops

Phase noise of the VCO inside a phase-locked loop (PLL) is attenuated at offset frequencies within the loop bandwidth f_l. Thus, jitter produced by the VCO only

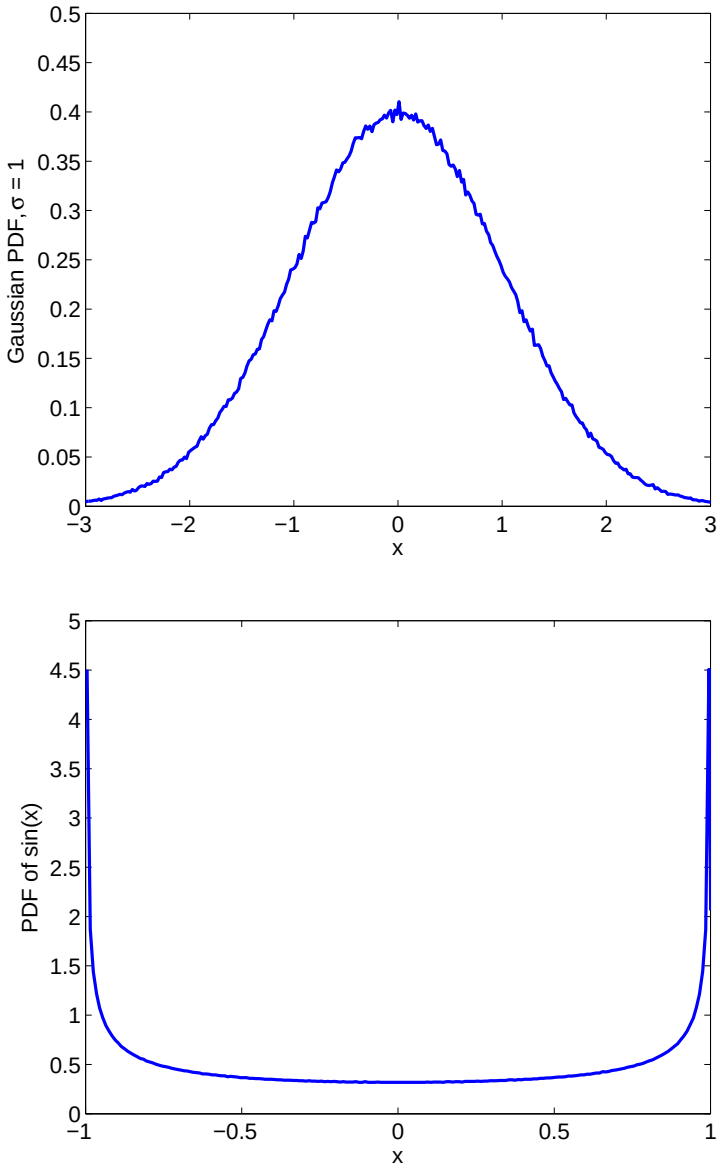


Figure 5-31: Amplitude distribution functions for (top) a Gaussian source, such as white thermal noise, and (b) a non-Gaussian source, in this case a sinusoid.

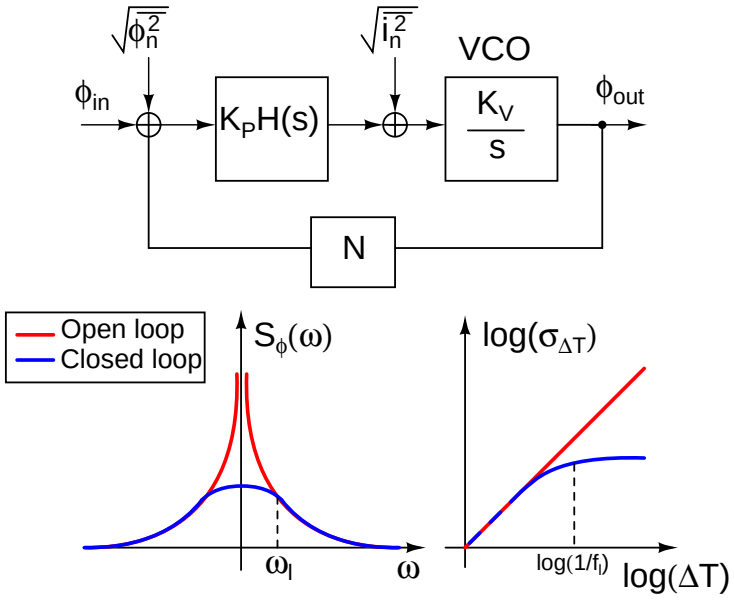


Figure 5-32: Jitter and phase noise of phase-locked loops, (top) linearized model with main noise sources, (bottom left) output power spectra and (bottom right) output jitter.

accumulates for time scales faster than the loop bandwidth. As a result, the total jitter is finite. In other words, "the loop locks". In the frequency domain, this behavior manifests itself as attenuation of the VCO's phase noise by the loop gain, i.e., within an offset approximately equal to the loop bandwidth. Typical power spectra and jitter plots are shown in Figure 5-32 in two cases: with the loop locked and with the loop unlocked (free-running VCO). In either case, however, jitter present at the PLL input is transferred to the VCO output by the closed-loop transfer function, i.e., it is not attenuated.

When the PLL is close to lock, we can use the linearized model shown in Figure 5-32. The loop gain is given by $L(s) = NK_PK_VH(s)/s$, and the transfer functions from ϕ_n and i_n to the output phase of the VCO are given by

$$\begin{aligned} \frac{\overline{\phi_{out}^2}}{\overline{\phi_n^2}} &= \left| \frac{L(s)}{1+L(s)} \right| \\ \frac{\overline{\phi_{out}^2}}{\overline{i_n^2}} &= \frac{1}{K_P \left| H(s) \right|} \frac{\overline{\phi_{out}^2}}{\overline{\phi_n^2}} \end{aligned} \tag{5.158}$$

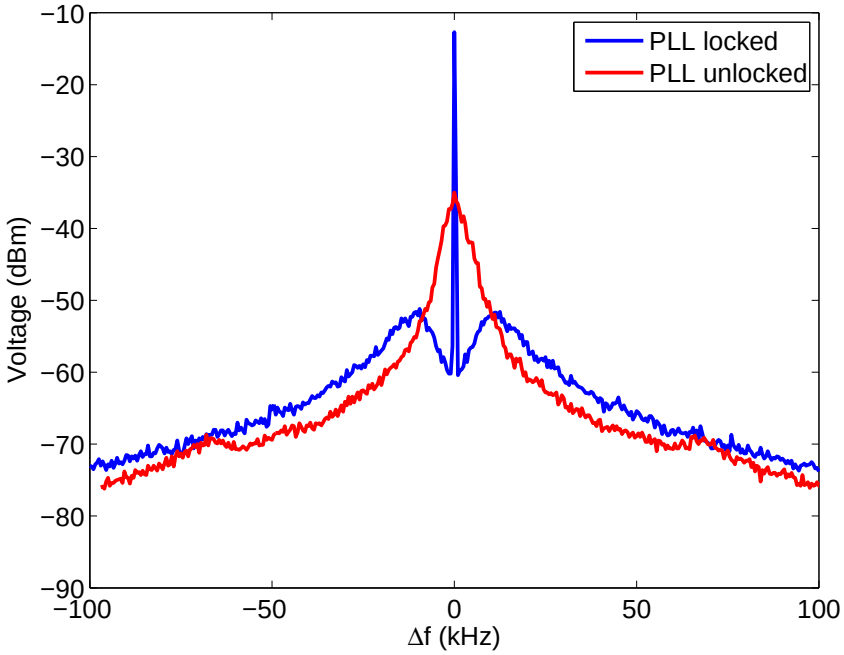


Figure 5-33: Measured power spectrum of the output of a PLL with the loop either locked or unlocked.

Figure 5-33 shows measured open and closed-loop phase noise plots of an integrated PLL built by the author in a 0.5μm CMOS process for clock and data recovery applications. The PLL had a third-order loop filter, operated at a center frequency of $f_0 = 2.5$MHz, and had a loop bandwidth of $f_l \approx 20$kHz. We see that, as expected, phase noise is attenuated within the loop bandwidth when the loop is locked. Figure 5-34 shows measured output jitter of the same circuit with the loop locked. The probability distribution is approximately Gaussian, indicating that random jitter is dominant.

5.15 Stochasticity in Chemical Reactions

5.15.1 Mathematical Formulation

Consider a spatially homogenous (well-stirred) reaction medium at constant volume and temperature, and containing $N \geq 1$ species and M reactions. The basic assump-

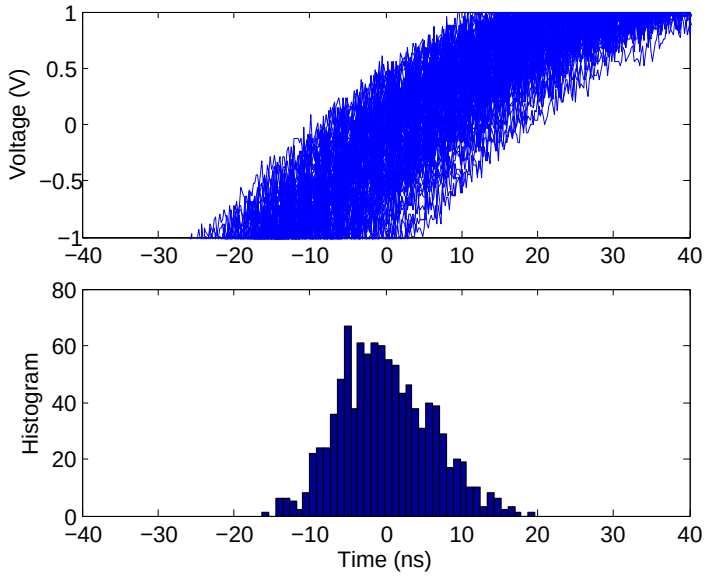


Figure 5-34: Measured jitter of a PLL with the loop locked, (top) eye diagram and (bottom) histogram.

tion made in most stochastic chemical simulation algorithms is that the underlying random process is Markovian, i.e., the probability that a certain reaction j occurs within a time period dt is $a_j(\mathbf{x})dt$, where a_j, known as the propensity, is only a function of the current state of the system, denoted by $\mathbf{x}$. Thus the random process has no memory and different reaction events are not correlated with each other. Chemical reactions change the numbers of various molecular species based on a $M \times N$ stoichiometry matrix $\mathbf{S}$. Element s_{ji} of the matrix is an integer that specifies the change in number of molecules of the i-th species due to one occurrence of the j-th reaction. We shall denote the j-th row of $\mathbf{S}$ by $\mathbf{s}_j$.

The propensity function $a_j(\mathbf{x})$, when multiplied by a small time interval dt, measures the probability that the j-th reaction will occur during that time interval. It can be written as the product of two terms: the *specific probability* c_j, and the number of distinct combinations of reactant molecules, denotd by $h_j(\mathbf{x})$. Thus, we may write

$$a_j(\mathbf{x}) = c_j h_j(\mathbf{x}) \tag{5.159}$$

The quantity $c_j dt$ is equal to the probability that a randomly chosen combination of reactant molecules will react in the time interval dt. For the first-order (unimolec-

ular) reaction $A \rightarrow$ products, we have $c_j = k_j$, the rate constant in the deterministic reaction rate equations, and $h_j(\mathbf{x}) = x_A$, the number of reactant molecules. For the second-order (bimolecular) reaction $A + B \rightarrow$ products, we have $c_j = k_j/V$ or $2k_j/V$ depending on whether A and B are distinct or identical, and V is the volume of the reaction medium. The values of $h_j(\mathbf{x})$ in the two cases are $x_A x_B$ and $x_A (x_A - 1)/2$, respectively.

In most cases of practical interest the Markovian assumption is a good one for each elementary chemical reaction. However, in certain cases we may wish to formally combine several elementary reactions into one for modeling or simulation purposes. In that case the composite reactions are no longer guaranteed to be Markovian, i.e., the random process can have memory. For example, kinetic models that include time delays are non-Markovian.

The state vector $\mathbf{X}(t)$ consists of the numbers of molecules of each species at time t, and is a random variable. A particular value of this random variable will be denoted by $\mathbf{x}$. Since species numbers must be non-negative integers, the Markov process occurs on an N-dimensional, non-negative integer space. Such processes are known as discontinuous or jump Markov processes in contrast to continuous Markov processes, which operate with real numbers. The time-evolution of probability distributions in jump Markov processes is given, without any approximations, by the *master equation* [198], which may be written as

$$\frac{\delta P(\mathbf{x}, t|\mathbf{x}_0, t_0)}{dt} = \sum_{j=1}^{M} \left[a_j(\mathbf{x} - \mathbf{s}_j) P(\mathbf{x} - \mathbf{s}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x}) P(\mathbf{x}, t|\mathbf{x}_0, t_0) \right] \quad (5.160)$$

where $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ is the probability that $\mathbf{X} = \mathbf{x}$ at time t, given that $\mathbf{X} = \mathbf{x}_0$ at time t_0. We see that the master equation is the discrete analog of a conservation law: the probability that the system is in a certain state changes at a rate equal to the difference between inward and outward fluxes to that state. While conceptually simple, in all but the simplest cases the master equation is impossible to solve, even numerically. Daniel Gillespie, one of the pioneers of the field, has shown [94] that, under certain conditions,

the original Markov condition can be used to derive a stochastic differential equation of the Langevin type that approximates the master equation, and is much easier to solve. In other words, we can approximate the underlying discrete Markov process with a continuous Markov process. The necessary conditions can be summarized as follows: the system must be such that there exists a domain of "macroscopically infinitesimal" time intervals. During any time interval dt in this domain,

1. No propensity function changes noticeably in value, i.e., $a_j(\mathbf{x}(t+dt)) \approx a_j(\mathbf{x}(t))$, $\forall j \in [1, M]$, and

2. each reaction is expected to occur many times, i.e., $a_j(\mathbf{x}(t))dt \gg 1$. $\forall j \in [1, M]$.

Similar time intervals were used by Einstein in his original analysis of Brownian motion. If these conditions are satisfied, the following chemical Langevin equation holds for $\mathbf{X}(t)$:

$$\frac{d\mathbf{X}(t)}{dt} = \sum_{j=1}^{M} \mathbf{s}_j a_j\left(\mathbf{X}(t)\right) + \sum_{j=1}^{M} \mathbf{s}_j \sqrt{a_j\left(\mathbf{X}(t)\right)}\Gamma_j(t) \tag{5.161}$$

where the $\Gamma_j(t)$ are uncorrelated Gaussian white-noise sources, defined as follows:

$$\Gamma(t) \equiv \lim_{dt \to 0} N(0, 1/dt) \tag{5.162}$$

where $N(0, \sigma)$ is a Gaussian random variable with zero mean and standard deviation σ. We should note that, since the time interval dt in (5.161) is a macroscopic infinitesimal, we cannot rigorously take the limit as $dt \to 0$ in the definition above. In other words, the noise sources cannot actually be white, though they usually fluctuate on timescales much faster than the deterministic dynamics given by the first term on the right-hand side of (5.161). We also note that the size of the second term in (5.161), relative to the first, is a measure of SNR and scales as $1/\sqrt{a_j\left(\mathbf{X}(t)\right)}$.

Every ordinay stochastic differential equation of the Langevin type, with white noise sources, can be uniquely associated with a deterministic partial differential equation (PDE). This equation, known as the Fokker-Planck equation, describes the

temporal and spatial evolution of the probability distribution of the random variable in the Langevin equation. Fokker-Planck equations may be considered as the continuous analogs of master equations for discrete-space systems, and have the form

$$\frac{\delta P(x,t)}{\delta t} = -\frac{\delta \left[D_1(x,t)P(x,t)\right]}{\delta x} + \frac{\delta^2 \left[D_2(x,t)P(x,t)\right]}{\delta x^2} \tag{5.163}$$

Here the first term describes drift, with D_1 being the drift vector, and the second term describes diffusion, with D_2 being the diffusion tensor. In our case, the Fokker-Planck equation has the form

$$\begin{aligned}\frac{\delta P\left(\mathbf{x},t|\mathbf{x}_0,t_0\right)}{\delta t} = &-\sum_{j=1}^{M} \frac{\delta \left[\mathbf{s}_j a_j(\mathbf{x}) P\left(\mathbf{x},t|\mathbf{x}_0,t_0\right)\right]}{\delta \mathbf{x}} \\ &+ \frac{1}{2}\sum_{j=1}^{M}\sum_{i,i'=1}^{M} \frac{\delta^2 \left[s_{ji}s_{ji'} a_j(\mathbf{x}) P\left(\mathbf{x},t|\mathbf{x}_0,t_0\right)\right]}{\delta x_i \delta x_{i'}}\end{aligned} \tag{5.164}$$

There are a couple of caveats, however. Converting the master equation into continuous phase space actually creates a PDE of infinitely large order, known as a Kramers-Moyal equation. Discarding all terms except those of first and second order is what gives us a Fokker-Planck equation. We are justified in dropping these terms only if a domain of macroscopically infinitesimal time intervals exists.

In the *thermodynamic limit*, the volume and number of molecules in the system are both assumed to be infinite. In this limit, all stochastic equations usually, but not always, reduce to the deterministic mass action rate equations of chemical kinetics. Exceptions include systems with multiple steady states: their long-term behavior is not well described by deterministic rate equations. Analytically solving master equations, Fokker-Planck equations or stochastic differential equations like Langevin equations is hard because the variable being solved for is a random variable. However, such equations can be reduced into an infinite hierarchy of equations for the moments (expectation values) of the random variable. In general, the hierarchy is not closed, i.e., the equation for a given moment (such as the mean, or first moment) always includes terms that depend on higher-order moments (such as the variance, or second

moment).

The hierarchy of moment equations can be closed if we assume that higher-order moments are either related to lower-order moments in a known way, or are zero. For example, Gomez-Uribe and Verghese [97] assumed that all third-order moments were zero, and obtained closed-form equations for the means, variances and covariances of $\mathbf{X}(t)$ directly from the master equation. They referred to these equations as mass-fluctuation kinetics, and showed that they provide, at a computational cost comparable to deterministic mass-action kinetics, fairly accurate estimates of the means and variances of $\mathbf{X}(t)$. Mass-fluctuation kinetics takes into account the fact that means are coupled to variances, and is therefore more accurate (for the same number of molecules) than mass-action kinetics, where variances are ignored.

5.15.2 Simulation Algorithms

Solving the master equation, even numerically, is intractable except in the simplest of cases, because it requires us to solve simultaneously for the probability of all possible reaction trajectories. However, finding a *particular* trajectory that satisfies the master equation is a much easier problem, and forms the basis of *stochastic simulation algorithms* (SSAs). A given SSA run only generates one value, say $\mathbf{x}$ of the random variable $\mathbf{X}(t)$. It must be run many times to generate the probability distribution $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ via random sampling. Thus SSAs are examples of sampling, or Monte Carlo methods [201] that are widely used in all areas of science. Formally, SSAs generate Markov chains that transform the probability distributions of random variables from known, standard distributions (typically uniform or Gaussian) into the ones that satisfy the master equation.

A given SSA is considered "exact" if it produces a given reaction trajectory $\mathbf{x}(t)$ with exactly the same probability as would be obtained by solving the chemical master equation. For a system in a given state, any exact SSA answers two questions: which reaction occurs next, and when does it occur? Gillespie's original SSA, proposed in a landmark 1976 paper [92] as the *first-reaction method*, is an exact algorithm in this sense. The first-reaction method is simple, easy to implement, and *does not even*

require that the master equation be explicitly written down. These appealing features have contributed to its enduring popularity. The algorithm is summarized below:

1. Initialize (i.e., set initial numbers of molecules, set $t = 0$).
2. Calculate the propensity function, a_j, for all j.
3. For each j generate a putative time, τ_j, according to an exponential distribution with parameter a_j.
4. Let μ be the reaction whose putative time, τ_μ, is least.
5. Change the number of molecules to reflect execution of reaction μ. Set $t = t+\tau_\mu$.
6. Go to Step 2.

The first-reaction algorithm needs M random numbers per iteration, where M is the number of reactions. In 1977, Gillespie proposed a mathematically equivalent "direct" SSA that uses only 2 random numbers per iteration [93]. This is an advantage since random numbers are computationally intensive to generate. For example, Gibson & Bruck found that a single call to their random number generator took 10 times as long as a division [90]. Further optimizations of Gillespie's algorithm have been proposed over the years; the prominent ones are listed in Table 5.2. All the "exact" algorithms on that list are mathematically equivalent. They suffer from the common problem of being computationally intensive for large numbers of molecules. Under these conditions propensities are large and reactions (which have to be individually accounted for) happen frequently, increasing the computational workload for a given simulation timespan.

SSA algorithms can be easily extended to model time-dependent Markov processes, i.e., systems where the specific probabilities, or rate "constants" c_j are actually functions of time. For example, modeling cell growth and division requires time-varying specific probabilities. As a cell grows, a given pool of molecules collides less frequently, causing reaction rates to decrease; thus, the values of c_j decrease with time.

Name of algorithm	Precision	Reference
First-reaction	Exact	Gillespie, 1976 [92]
Direct	Exact	Gillespie, 1977 [93]
Next-reaction	Exact	Gibson & Bruck, 2000 [90]
Modified direct	Exact	Cao et al., 2004 [34]
Just-in-time (moleculizer)	Exact	Lok & Brent, 2005 [169]
Event-leaping	Exact	Riedel & Bruck, 2005 [242]
Sorting direct	Exact	McCollum et al., 2006 [195]
Explicit tau-leaping	Approximate	Gillespie, 2001 [95]
Langevin leaping	Approximate	Gillespie, 2001 [95]
Implicit tau-leaping	Approximate	Rathinam et al., 2003 [235]
Trapezoidal implicit tau-leaping	Approximate	Cao & Petzold, 2005 [35]
Binomial tau-leaping	Approximate	Chatterjee et al., 2005 [40]
Optimized tau-leaping	Approximate	Cao, Gillespie & Petzold, 2005 [31]
Slow-scale SSA	Approximate	Cao, Gillespie, & Petzold, 2005 [32]
Adaptive tau-leaping	Approximate	Cao, Gillespie, & Petzold, 2007 [33]

Table 5.2: Stochastic simulation algorithms (SSAs) for chemical systems. This table was inspired by a diagram on Mario Pineda-Krch's blog, which may be found at http://pineda-krch.com/.

In 2001, Gillespie proposed a technique known as *tau-leaping* for speeding up SSAs while sacrificing some accuracy [95]. Tau-leaping techniques generate reaction trajectories that only approximately satisfy the master equation. The basic idea behind tau-leaping is to not keep track of the time at which *every* reaction occurs. Instead, a number of reaction events are treated as a single event whenever possible, thereby advancing, or "leaping" the simulation by a larger time step. In general, we can write the state vector at time $t+\tau$, i.e., $\mathbf{X}(t+\tau)$, in terms of the vector at time t as follows

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{M} K_j\left(\tau; \mathbf{x}, t\right) \mathbf{s}_j \tag{5.165}$$

where $\mathbf{X}(t) = \mathbf{x}$ and $K_j\left(\tau; \mathbf{x}, t\right)$ is the number of times reaction j occurs in the interval $[t, t+\tau)$. In general the M random variables $K_1, K_2 ... K_M$ are statistically dependent and it is not possible to calculate their joint probability distribution, or generate samples corresponding to that distribution. However, suppose τ is small enough that none of the propensity functions a_j changes by a significant amount during the

interval $[t, t+\tau)$. This is known as the *leap condition*. In other words, during the interval $[t, t+\tau)$ the number of molecules of any species changes by an amount that is much smaller than its value at the beginning of the interval. In electrical engineering terms, this condition allows us to use a small-signal approximation. In particular, if the leap condition is satisfied each component of $\mathbf{X}(t)$ evolves as a Poisson process. Therefore $K_1, K_2 ... K_M$ can be approximated by a set of M statistically independent Poisson random variables P_j, i.e.,

$$K_j(\tau; \mathbf{x}, t) \approx P_j(a_j(\mathbf{x}), \tau) \tag{5.166}$$

We see that tau-leaping will increase simulation speed if many reactions occur during the interval $[t, t+\tau)$, since the algorithm will treat them all as one event. If this condition is true for all reactions, i.e., $a_j(\mathbf{x})\tau \gg 1$, $\forall j \in [1, M]$, τ is a macroscopically infinitesimal time interval by our previous definition. In this case, we can approximate P_j by Gaussian random variables N_j with the same mean and variance, namely, $a_j(\mathbf{x})\tau$:

$$\begin{aligned} \mathbf{X}(t+\tau) &\approx \mathbf{x} + \sum_{j=1}^{M} N_j\left(a(\mathbf{x}), a(\mathbf{x})\right)\mathbf{s}_j \\ &= \mathbf{x} + \sum_{j=1}^{M} a(\mathbf{x})\tau \mathbf{s}_j + \sum_{j=1}^{M} \sqrt{a(\mathbf{x})\tau} N_j(0,1)\mathbf{s}_j \end{aligned} \tag{5.167}$$

where we have used the fact that $N(\mu, \sigma^2) = \mu + \sigma N(0,1)$, where μ and σ are the mean and standard deviation of the Gaussian random variable N. Equation (5.167) is simply a restatement of the Langevin equation, i.e., (5.161). Thus tau-leaping algorithms simulate the Langevin equation if the average time between successive reaction events is much less than τ, the leap time. Picking τ so that substantial speedup occurs while maintaining acceptable simulation accuracy is the main challenge in designing tau-leaping algorithms, and many schemes have been proposed. Some of these are listed in Table 5.2.

Exact SSA's are particularly slow when dynamics occur on widely-separated time

scales, i.e., the system is stiff, because the algorithms keep track of each event that belongs to the fast dynamics. Several approximate SSA algorithms optimized for stiff systems have been proposed. For example, an implicit state update rule allows larger tau-leaps to be made without sacrificing accuracy [235], and quasi-steady-state approximations can be made for certain species [32]. An excellent recent review of stochastic chemical simulation methods may be found in [96].

An important alternative to Gillespie-type SSAs for chemical systems is StochSim [207]. StochSim discretizes the master equation in time, converting it into a difference equation that is then solved numerically. The algorithm also treats each molecule as a software object with modifiable properties, which makes it extremely suitable for modeling reactions between multi-state systems such as protein complexes. However, it is significantly slower than Gillespie-type SSAs for most biochemical reaction networks, especially when there are many molecules present. A detailed comparison between the Gillespie and StochSim algorithms shows that StochSim is a first-order approximation of the Gillespie algorithm when its time step, Δt, is small [166]. Thus, the two algorithms become mathematically equivalent as $\Delta t \to 0$.

5.15.3 Spatial Inhomogenities

Broadly speaking, our initial assumption of a well-mixed, spatially homogenous reaction medium may fail for two separate reasons. Firstly, there may be non-negligible gradients in species concentrations due to mass transport (drift and diffusion), and secondly, there may be inhomogeneties in the reaction medium such as cell organelles, membranes and other three-dimensional structures. Roughly speaking, the former affects species concentrations, while the latter affects reaction rate constants. In both cases these quantities become functions of both position and time. Deterministic simulations of such spatially-inhomogenous systems are usually performed using finite-element methods. SSAs for modelling such systems have also been proposed. The basic idea is to divide space into cells. The number of molecules in each cell can change in two ways: via chemical reactions and transport processes, typically diffusion.

In any chemical system, only a small subset of the collisions between molecules results in chemical reactions; the rest cause diffusion. The cells used in reaction-diffusion simulations must be small enough to remain spatially homogenous, yet large enough to be in local thermal equilibrium, i.e, larger than the mean free path between collisions. In practice, a good choice for cell size is the *reactive mean free path*, defined as the average distance traveled by molecules between *reactive* collisions. By assuming Markovian behavior, a master equation, known as the reaction-diffusion master equation (RDMS), can be derived for the system [12,64]. In addition to drift and diffusion in the state space of species numbers, the resultant Fokker-Planck equation now includes terms corresponding to drift and diffusion in physical space. In other words, both chemical reactions and molecular diffusion cause fluctuations in the numbers of molecules. Extensions of Gillespie-type SSAs can be used to generate trajectories that satisfy the RDMS. A popular technique is known as the next-subvolume method [112]. It is a combination of the direct method [93] and the next-reaction method [90]. The former is used to determine which reaction happens next (and when), while the latter is used to determine the cell in which it occurs.

In addition to general simulation programs such as Matlab, there are literally hundreds of specialized software packages for stochastic modeling of both spatially homogenous and inhomogenous systems. Some freely-downloadable examples are StochSim, StochKit, MCell, GridCell and MesoRD. StochKit and StochSim are essentially general-purpose spatially-homogenous SSA solvers, though they can be extended to simple inhomogenous systems. MCell and GridCell use ray-tracing algorithms to track the three-dimensional trajectories of individual particles, and are suitable for modeling complicated spatial geometries. MesoRD uses an extension of the Gillespie SSA to model reaction-diffusion systems.

5.16 Stochasticity in Biology

Biological systems function in noisy, dynamic environments. Thus, they need to adapt, or evolve, based on changing environmental conditions, and also be robust, or

resistant, to perturbations from internal and external noise sources. Biology is robust at several levels of organization, ranging from the genetic code to embryonic development and ecological structure [302]. Herein lies a fundamental dilemma that must be addressed by any theory of complex adaptive systems: what is signal (to which we should adapt), and what is noise (which we should reject)? In other words, how can we be robust and adaptive at the same time? There are no completely general answers, but studies of evolutionary dynamics [216], learning theory, and systems biology [4,233] have provided several clues. Heterogenity in the population is important for rapid responses to environmental stress. Regulatory networks in dynamic environments should be *modular*, so that certain functions can be modified without affecting other functions. In fact, modular networks, which assemble complexity hierarchically, are evolutionarily favored in dynamic environments, but not in static ones, because in static environments the additional overhead required to maintain modularity can be removed without affecting fitness. Not surprisingly, human engineers, such as chip designers, tend to build modular networks because they are easier to maintain and upgrade.

Some natural and engineered systems utilize noise to improve performance, a phenomenon known as stochastic resonance (SR) [83, 144, 315]. For example, it is known that neural signal processing, particularly within sensory systems, often involves SR [58]. Behavioral evidence for enhancement of feeding performance, and search strategies in general, using SR has also been found [248]. However, the functional significance of noise in cellular processes remains controversial. There are only a few cases where it has been unambiguously shown that noise is necessary for normal operation of the cell. For example, consider cellular differentiation into specialized subtypes. This process is usually deterministic, and controlled by cell lineage and extracellular signalling molecules. In certain cases, however, it is known to involve a stochastic (probabilistic) choice between two or more alternatives [172, 179, 279].

Theoretical studies have shown that the same biological models can behave in *qualitatively* different ways depending on whether they are noisy (stochastic) or noise-free (deterministic). For example, stochasticity can generate bistability in phosphoryla-

tion cycles and cascades operating out of equilibrium [203]. Noise can also induce limit-cycle oscillations around a deterministically-unstable steady state. This phenomenon has been postulated as a possible mechanism for the creation of genetic timers, such as those within circadian oscillators [289]. Finally, noise can induce instability and oscillations in deterministically-stable systems containing time delays, such as models of gene expression [24].

Many biological systems operate far from equilibrium. As a result, the fluctuation-dissipation theorem cannot be applied directly. In particular, temperature gradients cannot be used to predict the direction of heat flow. However, an extended version of the theorem can be used to define an "effective temperature" if the non-equilibrium system changes slowly with time [46, 174]. Dynamics on different time scales exhibit different ratios of response (dissipation) and fluctuation (noise) functions, and thus have different effective temperatures[19].

It has been argued that biochemical signalling networks within cells operate close to fundamental physical limits set by molecular shot noise [16, 17].

5.16.1 Stochasticity in Gene Expression

Noise in gene expression has usually been divided into two parts: intrinsic, or uncorrelated across genes, and extrinsic, or correlated across genes. There are two major sources of extrinsic variability: upstream transcriptional regulatory networks shared across multiple genes, and heterogenity in the cellular population [232, 301]. The former tends to dominate when the number of expressed proteins is small, while the latter tends to dominate at high expression levels. The coupling of population dynamics to gene expression levels means that extrinsic variability can often be significantly reduced by measuring a subset of cells that is more homogenous than the population as a whole. This process, known as gating, is often carried out on the basis of cell size [215].

[19] We note that the fluctuation-dissipation theorem defines the normalized temperature variable kT to be the ratio of fluctuation to dissipation functions, as in the Einstein relationship $D/\mu = kT$. Effective temperature is defined in the same way.

Cross-correlation functions can be used to separate extrinsic noise into components due to shared genetic regulatory networks, and truly global sources such as variations in cell size [60]. The former reaches maximal magnitude for a characterstic, non-zero value of delay set by the dynamics of the regulatory network. By contrast, the latter affects different genes almost simultaneously, leading to cross-correlation functions that peak at zero delay. Both intrinsic and extrinsic noise sources have low-pass power spectra. In the simplest case, such spectra can be modeled as single-pole filters with single time constants[20]. For extrinsic noise, the dominant pole frequency is $\approx 1/T_{div}$, where T_{div} is the average time between cell divisions. For intrinsic noise, the pole frequencies are set by the dynamics of transcription, translation and gene regulation, and are usually (but not always) much faster than $1/T_{div}$.

Population variability depends both on the initial conditions and also on the environmental, i.e., growth conditions [175]. Small initial populations result in high initial variability, following Poisson statistics, both in the cell numbers distributed amongst different phenotypes in the same population (known as intra-colony variability), and different populations (known as cross-colony variability). Intra-colony and cross-colony variability are analogous to instrinsic and extrinsic gene expression noise, respectively. In unbounded growth media the dynamical system describing cellular growth has no feedback terms[21]. As a result, variability due to initial conditions persists forever. As we might expect, when growth is subject to environmental constraints, variability due to initial conditions eventually dies away, and steady-state variability is determined only by the internal dynamics of the system.

5.16.2 Stochasticity in Neural Systems

Broadly speaking, the nervous system performs three tasks: signal transduction, information processing and actuation. Each of these steps is subjected to random fluctuations, or noise. Noise puts upper limits on the density of neural information

[20] Known to mathematicians as Ornstein-Uhlenbeck or mean-reverting processes, or AR(1) processes in discrete time.

[21] For example, in the linear dynamical system $d\mathbf{x}/dt = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ absence of feedback means that $\mathbf{A} = 0$.

processing, particularly in the brain [66, 67]. This limit is imposed through various mechanisms: the number of ion channels in small axons, the amount of neurotransmitter in synapses, and crosstalk, or ephaptic coupling, between adjacent neurons. In each case the amount of noise increases as the structures become smaller, because their behavior is then averaged over fewer noisy events, such as openings and closings of ion channels or release of neurotransmitter molecules in synapses.

Ion channels in nerve membranes open and close randomly because of thermal motion. The flow of charge through these channels causes spontaneous sub-threshold fluctuations in the membrane voltage of nerve cells. It has been suspected for over 50 years that such noise can have important physiological consequences [71]. Let us assume we have N_0 ion channels and a capacitance C_0 per unit area of the membrane. If nearby ion channels are uncorrelated the noise voltage amplitude scales as $N_0\sqrt{A}/\left(C_0 A\right) \propto 1/\sqrt{A}$, where A is the total area of the membrane. Thus large cells are less noisy than small cells. In fact, the noise amplitude scales as $1/\sqrt{N}$, where $N = N_0 A$ is the total number of ion channels. This behavior is an example of the law of large numbers, or reduction in variance by averaging.

There are several families of ion channels that change states (open or close) in response to different stimulii, such as membrane voltage, the presence of ligand molecules, light, and mechanical pressure or displacement. Voltage-sensitive channels are required for the generation of action potentials, and are therefore common in nerve cells. The opening and closing of a single channel causes charge to flow, which affects the membrane voltage. However, the changed membrane voltage now affects the opening and closing of *every* channel in the membrane. This interesting feedback loop between single-channel and collective dynamics can affect the voltage noise across the membrane. In particular, the noise amplitude was observed to decrease as $1/N$, not $1/\sqrt{N}$ when N, the total number of ion channels, was large [249, 250]. Thus, it is possible that the nervous system exploits correlated fluctuations in order to improve computational precision.

The effects of noise are felt at all levels of the nervious system, ranging from molecular to behavioral, but are not always detrimental. Stochastic resonance is a

common example. Other benefits include linearization of neural activation functions, and improved learning and adaptation to dynamic environments [67].

5.17 Log-Normal Distributions

The central limit theorem guarantees that most random processes that can be decomposed into the sum of many independent parts have Gaussian, or normal probability distributions. However, many probability distributions in biological systems are log-normally distributed, i.e., they are normally distributed on a logarithmic scale [164]. In particular, cellular properties with relatively high coefficients of variation (more noise) are usually well described by log-normal distributions. Such distributions occur when the random process can be decomposed into the *product* of many independent parts.

Consider a practical example. The low-frequency gain of the unidirectional RF cochlea is the product of the gains of the individual stages. Ideally each stage should have unity gain. Let us assume that the actual gain of the n-th stage is $1+\chi_n$, where χ_n is a zero-mean random variable. Then the gain after n stages is given by

$$G(n) = \prod_{i=1}^{n} (1 + \chi_n) \tag{5.168}$$

As described earlier, we use feed-forward and feedback techniques to ensure that stage gains are close to 1, so we can assume that $|\chi_n| \ll 1 \ \forall n$. In this case we can use the identity $\log(1+x) \approx x$ for $|x| \ll 1$ to rewrite (5.168) as

$$\log\left(G(n)\right) \approx \sum_{i=1}^{n} \chi_n \tag{5.169}$$

We can now apply the central limit theorem: As long as each χ_n is independent and has finite variance, $\log\left(G(n)\right)$ approaches a normal distribution as $n \to \infty$. Thus, $G(n)$ approaches a log-normal distribution. For simplicity, assume that the variance of χ_n is equal to σ^2, a constant $\forall n$. Then the log-normal variable $G(n)$ has the following statistical properties:

$$\begin{aligned} \text{mean} &= e^{n\sigma^2/2} \\ \text{median} &= 1 \\ \text{mode} &= e^{-n\sigma^2} \\ \text{variance} &= \left(e^{n\sigma^2} - 1\right) e^{n\sigma^2} \end{aligned} \tag{5.170}$$

Note that the mean value of $G(n)$ is greater than unity even though individual stages have a mean gain of unity. The asymmetry, or skewness of the log-normal distribution is responsible for this surprising phenomenon.

5.18 Further Reading

The literature on noise, even within the restricted sub-field of semiconductor device physics, is vast, and, one is tempted to say, beyond mortal reckoning. However, listing a few highlights may prove useful. A. van der Ziel's original papers from the 1950's and 1960's are still excellent references for the basic theory of noise in diodes, junction transistors (like BJTs), field-effect transistors (like MOSFETs) and other electronic devices. An excellent review of his work is presented in [295]. Sarpeshkar, Delbruck and Mead have, in [257], argued that shot and thermal noise are actually the same phenomenon. An excellent summary of various types of MOSFET noise can be found in [158]. Another feature of [158] is the large collection of excellent references on the subject, several of them historical in nature. An excellent book that, amongst other things, discusses evidence for $1/f$ noise in cognition is [304].

Chapter 6

Circuit Models of Chemical Reactions

The rest is chemistry!

– Carl D. Anderson, after discovering the positron in 1932

In this chapter we describe fast, scalable hardware models of chemical reaction networks. The detailed analogy between transistors and chemical reactions is used to build integrated circuits that can efficiently simulate the dynamics of complex chemical reactions. We also present experimental results from a prototype chip designed for simulating chemical kinetics.

6.1 Introduction

Cells are truly impressive molecular machines. Table 6.1 summarizes the performance of a typical mammalian cell. We see that cells are very *efficient*: they perform large numbers of biochemical operations within a small volume while consuming very little power. We define biochemical operations as active or endothermic chemical reactions that only occur when energy, in the form of ATP molecules, is supplied to them. Cells are also very *flexible*, and use such active reactions to perform a wide variety of functions. Examples include nanoscale sensing, actuation, and pattern recognition; communication, transport, and feedback regulation; maintenance, growth, and

Table 6.1: Performance numbers of a typical mammalian cell

Parameter	**Value**
Power consumption	1pW
Molecular interaction network	≈20,000 nodes (genes & proteins)
Biochemical operations	10^7/sec
Volume	$10 \times 10 \times 10 \mu m^3$

Table 6.2: Some typical cellular parameter values

Parameter	**Bacterial** (*E. coli*)	**Yeast** (*S. cerevisae*)	**Mammalian** (human fibroblast)
Cell volume	$\approx 1\mu m^3$	$\approx 10^3\mu m^3$	$\approx 10^4\mu m^3$
Concentration of single proteins/cell	≈1nM	≈1pM	≈0.1pM
Transcription delay	≈ 1min	≈ 1min	≈ 30min
Translation delay	≈ 2min	≈ 2min	≈ 30min
Typical mRNA lifetime	2 – 5min	10min – 1hr	10 min – 10hrs
Cell generation time	30min – hrs	2 – hrs	20hrs – ∞
Protein active/inactive transition timescale	1 – 100μs	1 – 100μs	1 – 100μs
Small molecule/protein binding timescale	≈ 1ms	≈ 1s	≈ 1s
Transcription factor binding timescale	≈ 1s		

reproduction; detoxification, and defense. Thus cells should serve as fertile sources of biological inspiration for building efficient computers.

Cellular processes can be modeled at many different levels of abstraction [281]. We shall focus on mechanistic models based on systems of chemical reactions. Table 6.2 shows typical values of some parameters that are useful for modeling cellular dynamics. Numbers are quoted for several representative species, and were taken from a more complete list in [4]. Many other numbers of biological interest can also be obtained from the BioNumbers website (http://bionumbers.hms.harvard.edu/).

The idea of using analog computers for simulating chemical kinetics is at least 50 years old [180]. Simple circuit analogies can also be used for modeling genetic networks. Basic electronic models of such networks have recently been discussed in the literature [50, 110]. However, no one thus far appears to have exploited detailed similarities behind the equations of chemistry and the equations of electronics. This

chapter outlines how we may do so, thus allowing us to implement models of complex molecular networks in chemistry and biology as analog integrated circuits. Our chips, which take full advantage of parallelism, may be useful in performing fast simulations, parameter discovery, optimizations, and other operations. Such computations run slowly on microprocessors, especially when stochastic effects have to be included. The performance of both analog and digital computers, as measured by chip size, speed and power consumption, improves with process scaling [44]. Thus, our performance advantage should not diminish over time.

Field-Programmable Gate Arrays (FPGAs) constitute another approach towards reducing the computational cost of stochastic simulations [251]. The data flow and processing architecture of FPGA's can be dynamically reconfigured to suit a particular algorithm. By implementing multiple parallel processing paths, FPGA's can run stochastic simulation algorithms much faster than conventional microprocessors. As an added bonus, such slow-and-parallel processing decreases the energy needed per operation, i.e., improves energy efficiency. Similar improvements in speed and energy efficiency can be obtained by running stochastic simulation algorithms on graphics processing units (GPUs) instead of conventional microprocessors [163].

The most computationally expensive part of Gillespie's well-known stochastic simulation algorithm, consuming approximately 98% of total processor time, is the generation of exponentially-distributed random numbers. Custom analog integrated circuits for generating such numbers have recently been reported [191]. The authors of the latter paper claim a potential speed-up of about two orders of magnitude over a purely software implementation of the algorithm. Thus, it appears that special-purpose hardware provides significant performance advantages over general-purpose computers for studying large biological networks.

6.2 Models of Chemical Reactions

The standard formulation of chemical kinetics in terms of the law of mass action is not universally valid in intracellular environments, because of several reasons [99,325].

The high concentrations and large sizes of biomolecules leads to *crowding*, in which a significant fraction of the reaction volume is excluded, i.e., occupied by the molecules themselves. Some biomolecules are spatially *confined*, which can lower the effective dimensionality of the reaction envrionment. Time-dependent rate "constants" are often observed [140]. In addition, cells are spatially *heterogenous* because of the presence of organelles, large molecules and electric fields; as a result, the "well-mixed" approximation may not be valid. Finally, *non-specific interactions*, i.e., interactions between molecules who not react chemically, are common in cells and can change reaction rate constants.

In practice, however, mass action models are found to be approximately valid for a wide range of cellular processes. Excluded volume effects can be modeled by retaining mass-action rate laws, but replacing molecular concentrations with quantities called *activities*. The activity of a molecule is defined as the product of its concentration and an empirically-measured or calculated *activity coefficient*. Spatial heterogeneity can also often be handled within the mass-action framework: We can define multiple reaction compartments, each of which is assumed to be internally homogenous. For example, the extra-cellular environment, the cell membrane, the cytoplasm and the nucleus can be modeled as individual compartments. In this section, we shall therefore focus on mass-action models of chemical reactions.

6.2.1 Simple Circuit Analogs

Analog circuit designers have to deal with many of the same problems as biological systems while processing information. Both fields use low levels of abstraction. As a result, they have to rely on the detailed physics of noisy and unreliable devices to perform computation. However, this same property also enables them to minimize consumption of hardware and energy resources, i.e., to be efficient. These deep underlying similarities suggest that representing biological systems using analog electronics might be useful both for modeling and design purposes. The field of neuromorphic VLSI grew from this basic premise [199].

In this chapter, we show that chemistry can also be efficiently modeled using ana-

log electronic circuits. For example, Figure 6-1 shows two examples of how simple resistor-capacitor circuits can model mass-action kinetics of reversible chemical reactions [5]. In such models, the concentration of each species is represented by the charge stored on a capacitor. The figure on the left shows a resistive-divider network that models Michaelis-Menten kinetics in enzyme-substrate binding reactions. The reaction in question may be represented as

$$E + S \underset{k_r}{\overset{k_f}{\rightleftharpoons}} ES \tag{6.1}$$

where E is the enzyme, S is the substrate, ES the enzyme-substrate complex and k_f and k_r are the kinetic rate constants of the forward and reverse reactions, respectively. The voltage source represents the total enzyme concentration $[E_t] = [E] + [ES]$ (whether bound to substrate or not), while the capacitor voltage represents the bound enzyme concentration $[ES]$; the rates of the forward and backward reactions are modeled by the resistors. The figure on the right shows a similar model for an acid dissociation reaction (hydrolysis), which may be represented as

$$A \underset{k_r}{\overset{k_f}{\rightleftharpoons}} A^- + H^+ \tag{6.2}$$

where A is the undissociated acid and A^- is its anion. In the figure $[A_t] = [A] + [A^-]$ represents the total concentration of the acid anion (whether bound to hydrogen or not), and $[A^-]$ is the concentration of the dissociated acid anion.

Saturation effects that arise in such chemical reactions transparently manifest themselves as the laws of resistive-divider action in electronic systems, the time constant of the reaction manifests as that of a parallel RC circuit, and the forward and backward reaction fluxes manifest themselves as current flows through the k_f and k_r resistors. The resistor values can depend on other concentrations in the circuit, as shown, resulting in nonlinear dynamics. Complex reactions can be decomposed into elementary molecular steps, each of which can be modeled by a simple RC circuit similar to the ones shown in Figure 6-1.

Figure 6-2 shows a more complicated example: a network that models the pro-

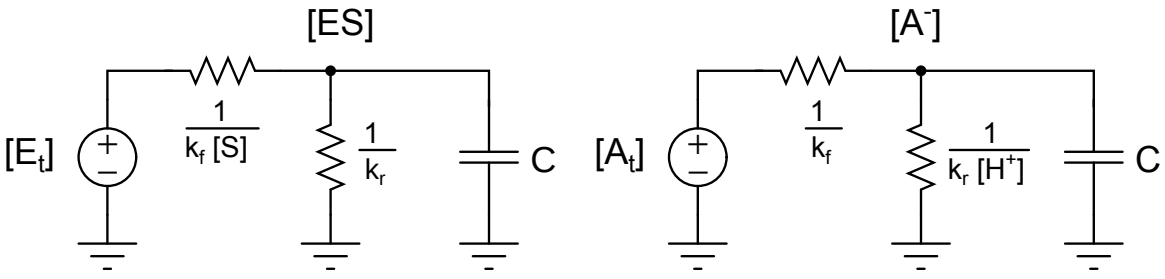


Figure 6-1: Electronic circuit models of two simple, but important chemical reactions: Michaelis-Menten kinetics (left) and an acid dissociation reaction (right).

duction of a protein from a gene. The kinetic parameters g_{mtri} $(1 < i < n)$, g_{rep}, g_{mtl}, K_R, τ_{mRNA} and τ_P characterize the behavior of the network. Several parallel promoters, with concentration $[T_{Pi}]$, where $1 < i < n$, and a common repressor with concentration $[R]$ and rate constant K_R control the rate of transcription of a gene into messenger RNA (mRNA). The concentration of mRNA, $[mRNA]$, is encoded as the voltage across a capacitor, while a resistor in parallel models its degradation. The voltage across another RC network represents $[P]$, the concentration of the protein P that is produced when this mRNA is translated. Networks of this type can be extended in an obvious way to incorporate the effects of multiple genes and proteins and interactions between them.

6.2.2 General Analog Circuit Model

We will now generalize the examples considered in the previous section. Consider a reaction network composed of N distinct molecular species. We will always assume that individual chemical reaction events are uncorrelated. To start with, the reaction medium is assumed to be a well-stirred single-phase system, such as a dilute aqueous solution. This is a "single compartment" model, i.e., we assume that spatial concentration gradients are negligible, so that the concentration of any species can be uniquely represented by a single number[1]. The set of reactant concentrations at any time t can be represented as a vector $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]$ of length N, where $x_i(t), 1 \leq i \leq N$ is the concentration of the i-th species.

[1]We later describe how to model multi-compartment systems.

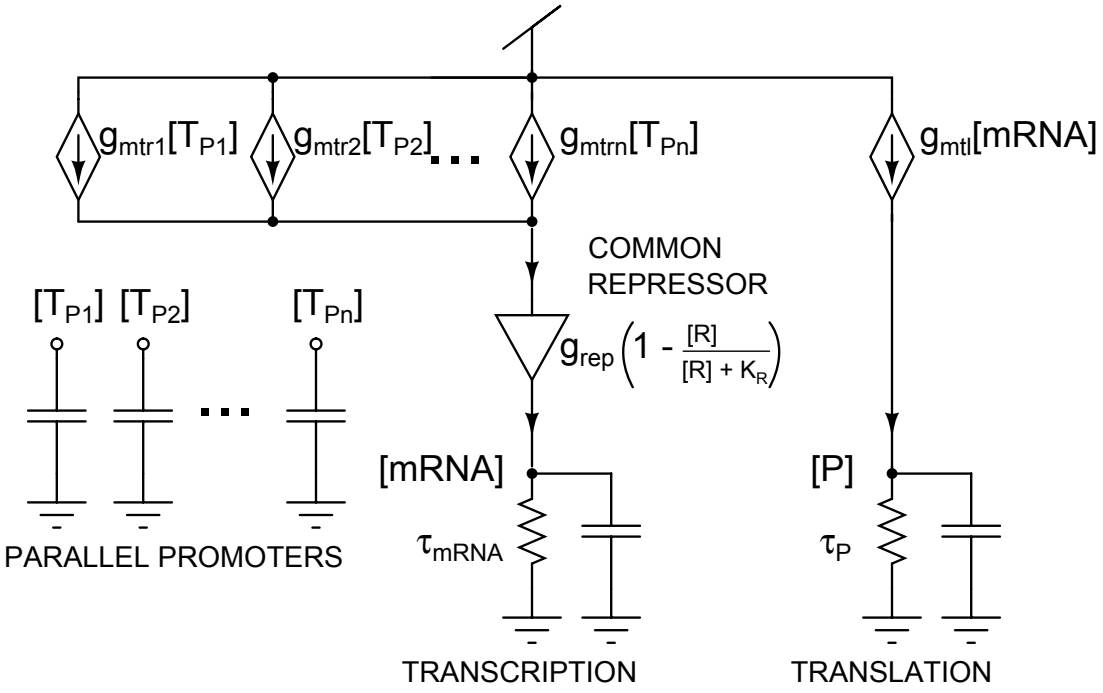


Figure 6-2: A simple electronic circuit that models the processes of transcription and translation of a single gene.

A given set of chemical reactions can be decomposed into elementary molecular steps in many ways. The physically correct set of such steps is known as the *mechanism* of the reaction. each elementary step follows mass-action kinetics [222], the rate of change of x_i with time is given by

$$\frac{dx_i}{dt} \equiv \dot{x_i} = c_i + \sum_{j=1}^{N} d_{ij}x_j + \sum_{j=1}^{N}\sum_{k=1}^{N} e_{ijk}x_j x_k + \ldots \tag{6.3}$$

where the first, second, third... terms on the right-hand side correspond to zeroth, first, second... order kinetics, respectively. Also, c_i, d_{ij}, e_{ijk},... are constants known as *kinetic rate constants*. Each rate constant can be positive (if species i is being produced in that reaction) or negative (if it is being consumed). In this formulation each reaction is unidirectional, i.e., the forward and backward parts of a reversible reaction are considered separately.

Using the usual Einstein summation-over-indices convention, we can rewrite equation (6.3) in simplified form as

$$\dot{x_i} = c_i \cdot 1 + d_{ij} x_j + e_{ijk} x_j x_k + \ldots \tag{6.4}$$

The mathematical structure of the expression on the right hand side now becomes clearer. Remember that i is fixed (we are considering the i-th species). The first term is the product of the vector c_i with the constant 1 (i.e., a rank 1 tensor with a rank 0 tensor). The second term is the inner product of the matrix d_{ij} and the vector x_j (i.e., a rank 2 tensor with a rank 1 tensor). The third term is the inner product of e_{ijk} and $x_j x_k$ (i.e., a rank 3 tensor with a rank 2 tensor). Here $x_j x_k$, which is a $N \times N$ matrix, is the outer (tensor or Kronecker) product of the vectors x_j and x_k. The series in (6.4) can be continued to include higher order tensor products in an obvious way. In general, the n-th term in this series represents $(n-1)$-th order rate kinetics and consists of the inner product of two tensors, with ranks n and $(n-1)$, respectively.

In principle, M-body molecular collisions result in M-th order mass-action kinetics. However, the probability of three or more molecules colliding simultaneously is usually negligible at practical temperatures, concentrations, and pressures. Therefore elementary steps are limited to zeroth, first or second order kinetics and the series in (6.3) can be safely terminated after the first three terms. In order to generalize our formulation we also note the following:

- Species concentrations can also depend on external inputs to the system. Let the vector of such external inputs be denoted by $\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_M(t)]$, where in general $N \neq M$. Inputs can affect species concentrations directly (resulting in first-order kinetics) or in combination with other species (resulting in second-order kinetics).

- The outputs of interest may consist of linear combinations of all the N species in the reaction network. Let the vector of such outputs be denoted by $\mathbf{y}(t) = [y_1(t), y_2(t), \ldots, y_P(t)]$, where in general $P \neq N$ or M.

Thus, our complete reaction model is given by

$$\begin{aligned} \frac{dx_i}{dt} &= c_i \cdot 1 + d_{ij}x_j + e_{ijk}x_j x_k + f_{ij}u_j + g_{ijk}x_j u_k \\ y_i &= h_{ij}x_j + k_{ij}u_j \end{aligned} \tag{6.5}$$

where c_i, d_{ij}, e_{ijk}, f_{ij}, g_{ijk}, h_{ij} and k_{ij} are constant coefficients. In matrix notation (6.5) becomes

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{C} + \mathbf{D}\mathbf{x} + \mathbf{E}\left(\mathbf{x} \otimes \mathbf{x}\right) + \mathbf{F}\mathbf{u} + \mathbf{G}\left(\mathbf{x} \otimes \mathbf{u}\right) \\ \mathbf{y} &= \mathbf{H}\mathbf{x} + \mathbf{K}\mathbf{u} \end{aligned} \tag{6.6}$$

where $\otimes$ denotes the tensor, outer or Kronecker product. The similarity of (6.6) to the standard **ABCD** matrix model of a linear dynamical system is evident:

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \tag{6.7}$$

where, as before, $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y}$ denote the vectors of inputs, state variables and outputs, respectively. We see that the chemical system is a generalization of the linear system represented by (6.7). The tensor product term (i.e., second-order mass-action kinetics) adds a second-order nonlinearity to it.

6.2.3 Electrical Circuit Equivalent

Our goal is to emulate the dynamics of the reaction system described in the previous section with an electrical circuit. Chemistry and electronics are analogous in several ways. Chemical potentials map naturally to voltages, i.e, electronic potentials, while molecular fluxes map to electron flows, i.e., currents. The chemical potential of an enzyme or catalyst A, denoted by μ_A, controls the energy barrier of a chemical

reaction, exponentially changing its speed. In an analogous fashion, gate voltage V_G controls the electron energy barrier between source/drain terminals and the channel of a transistor, exponentially changing electron flow rate. The detailed analogy between chemical reactions and transistors with exponential I-V characteristics, in this case, subthreshold MOSFETs, is illustrated in Figure 6-3. We can use this analogy to efficiently model large-scale chemical information-processing systems found in biology.

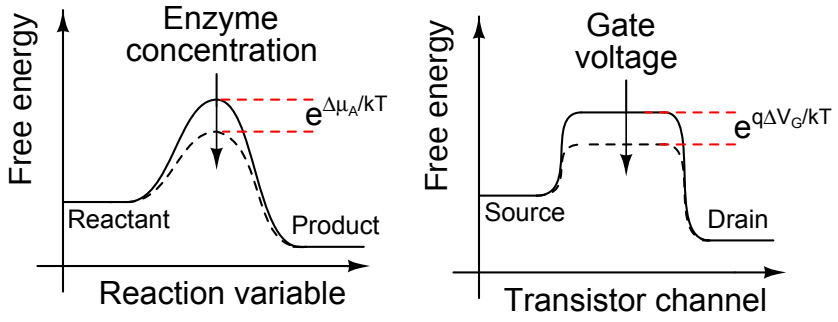


Figure 6-3: Reactant/product chemical potentials, molecular flux and enzyme chemical potential in a chemical reaction (left) are analogous to source/drain voltages, electronic current and gate voltage in a subthreshold MOS transistor, respectively (right).

We encode the chemical potential of each species as the voltage V on a capacitor of value C. In dilute solutions[2] the chemical potential of the i-th species is given by

$$\mu_i = \mu_0 + k_B T \ln\left(\frac{x_i}{X_0}\right) \tag{6.8}$$

where μ_0 and X_0 are constants referred to as the reference chemical potential and reference concentration, respectively, and x_i is the concentration of the species. Note that μ_0 and X_0 are assumed to be the same for all species. It can be shown that (6.8), where this assumption is implicit, can be used to derive a number of well-known colligative laws, such as the ideal gas law, the law of mass action, Henry's law, Raoult's law, Nernst's distribution law and the osmotic pressure law [132]. Conceptually, therefore, (6.8) should be treated as a basic law, like the laws of thermodynamics.

For non-dilute solutions and/or charged species, i.e., ions, (6.8) can often still be

[2] A solution is considered dilute when interactions between solute particles are negligible compared to solute-solvent interactions. In this situation solute molecules essentially behave like an ideal gas.

applied, but with x_i replaced by an effective concentration $x_i a_i$, where a_i is known as the activity coefficient of the species in question. To convert from μ to V we divide by κq, where κ is a constant and q is the electronic charge. Equation (6.8) can then be written as

$$\ln\left(\frac{x_i}{X_0}\right) = \frac{\kappa\left(v_i - V_0\right)}{\phi_T} \Rightarrow x_i = X_0 \exp\left(\frac{\kappa\left(v_i - V_0\right)}{\phi_T}\right) \tag{6.9}$$

where $\phi_T = k_B T/q$ is the thermal voltage and $V_0 = \mu_0/(\kappa q)$ is a constant reference voltage. The concentrations of the input and output species are encoded similarly. Differentiating (6.9) on both sides, we get

$$\frac{d\ln\left(x_i\right)}{dt} = \frac{1}{x_i}\frac{dx_i}{dt} = \frac{\kappa}{\phi_T}\left(\frac{dv_i}{dt}\right) \tag{6.10}$$

For convenience we now convert concentrations to currents by defining $i_i/I_0 = x_i/X_0$, i.e., $i_i = I_0 \exp\left(\kappa\left(v_i - V_0\right)/\phi_T\right)$, where I_0 is a constant reference current. Similarly, we also define $i_{ui}/I_0 = u_i/X_0$ and $i_{yi}/I_0 = y_i/X_0$. Substituting (6.10) in (6.5), we get

$$\begin{aligned} C\frac{dv_i}{dt} = \frac{C\phi_T}{\kappa I_0}\Bigg[&\frac{c_i}{X_0}\frac{I_0^2}{i_i} + \sum_{j=1}^{N} d_{ij}\frac{I_0 i_j}{i_i} + X_0\sum_{j=1}^{N}\sum_{k=1}^{N} e_{ijk}\frac{i_j i_k}{i_i} \\ &+ \sum_{j=1}^{M} f_{ij}\frac{I_0 i_{uj}}{i_i} + X_0\sum_{j=1}^{N}\sum_{k=1}^{M} g_{ijk}\frac{i_j i_{uk}}{i_i}\Bigg] \end{aligned} \tag{6.11}$$

$$i_{yi} = \sum_{j=1}^{N} h_{ij} i_j + \sum_{j=1}^{M} k_{ij} i_{uj} \tag{6.12}$$

Equations (6.11) and (6.12) are statements of KCL. The index i runs from 1 to N in the first equation (N state variables) and 1 to P in the second (P outputs). The reference concentration and current (X_0 and I_0) are normally chosen to be the geometric means of the minimum and maximum concentrations and currents of interest. In subthreshold CMOS implementations the minimum allowable current is set by leakage and parasitic capacitances, while the maximum is set by the onset of strong

inversion. The circuit equations shown in (6.11) and (6.12) are dynamically equivalent to the original chemical equations. Dynamical equivalence refers to the fact that the dynamics of *normalized* chemical and electrical state variables, i.e., x_i/X_0 and i_i/I_0, respectively, are identical.

Equation (6.11) can be easily implemented in hardware using log-domain circuits [76]. The currents i_i are proportional to $\exp(\kappa v_i/\phi_T)$, where $i_i \geq 0, \forall i$. Thus each current can be created by a single BJT or subthreshold MOSFET operated in its forward active (BJT) or saturated (MOSFET) region. In addition, real biochemical networks are sparse: most species participate in fewer than four reactions. Because of this sparseness, most of the coefficients c_i, d_{ij}, e_{ijk}, f_{ij} and g_{ijk} are zero (the reactions in question do not occur). Therefore only a small subset of the $1+N+N^2+M+MN$ terms on the right hand side of (6.11) are non-zero. Each of these contributes a current $\pm\beta i_1 i_2/i_i$ to Cdv_i/dt, where β is a dimensionless, non-negative constant and i_1 and i_2 are non-negative currents. As a result, (6.11) can be easily implemented with single-quadrant log-domain integrators, which can be implemented with very few transistors.

Equation (6.12) is also easy to implement: the state variable currents i_j and input currents i_{uj} (we have N of the former and M of the latter) are summed together at a single node with appropriate weighting factors h_{ij} and k_{ij}. The result is the output current i_{yi}. We carry out P such summations to produce the P output currents.

6.2.4 Thermodynamics

All reaction networks must satisfy the thermodynamic constraint that the net change in thermodynamic potential around any reaction loop is zero. In other words, thermodynamic potentials are state variables. It is also a statement of the first law of thermodynamics, i.e. that total energy is conserved. Biological systems usually operate at constant temperature and pressure. As a result the appropriate potential is Gibbs free energy G. Mathematically, any reaction network, irrespective of whether or not it is at equilibrium, must satisfy

$$\sum_{j \in loop} \Delta G_j = 0 \tag{6.13}$$

where ΔG_j is the change in Gibbs free energy of the j-th reaction. This quantity is defined as

$$\Delta G_j = \left(\sum_{k \in products_j} G_k \right) - \left(\sum_{k \in reactants_j} G_k \right) \tag{6.14}$$

In dilute solutions, the free energy change of any reaction is given by

$$\Delta G_j = \Delta G_{j0} + N_A \Delta \mu_j \tag{6.15}$$

where N_A is Avogadro's number and $\Delta \mu_j$ is the change in chemical potential of the reaction (analogous to (6.14)). Thus, the free energy change ΔG_j in any reaction is the sum of two parts. The first part, ΔG_{j0}, is independent of species concentrations. However, it depends on the rate constant, i.e., the properties of reactant and product species. The second part, $\Delta \mu_j$, depends only on species concentrations via (6.8). In addition, the second law of thermodynamics predicts that $\Delta G_j = 0 \; \forall \; j$ at thermodynamic equilibrium. Thus, we get

$$\Delta G_{j0} = -N_A \Delta \mu_{j,eq} \tag{6.16}$$

where $\mu_{j,eq}$ is the value of $\Delta \mu_j$ at equilibrium. As a result, (6.13) may be rewritten as

$$\sum_{j \in loop} (\mu_j - \mu_{j,eq}) = 0 \Rightarrow \sum_{j \in loop} (\Delta v_j - \Delta v_{j,eq}) = 0 \tag{6.17}$$

where the second equation follows from the first by using (6.8) and (6.9). This equation, which is applicable to our circuit model, is a version of Kirchoff's Voltage Law (KVL). However, we cannot directly connect the voltage drops Δv_j into a KVL loop because of the presence of the $\Delta v_{j,eq}$ terms, which act as batteries. Active circuits, such as current mirrors, must be used to generate these terms. Fundamentally, the

circuits compensate for the fact that electrons, unlike different chemical species, are indistinguishable.

Chemical rate "constants" are actually strong functions of temperature. A simple, widely used, and remarkably accurate formula for predicting this dependence is the Arrhenius equation. It is given by

$$k = Ae^{-E_a/k_BT} \tag{6.18}$$

where A and E_a are relatively weak (often polynomial) functions of temperature. This empirical formula can be derived by assuming the presence of an high-energy intermediate, or transition state, between the reactants and products, as shown in Figure 6-3. The quantity E_a, also known as the activation energy, can be viewed as the energy difference between the transition state and the reactants. Transistors have a similar energy landscape, with the channel corresponding to the transition state. At room temperatures both electronics and chemistry are well described by Boltzmann statistics. Thus, flux or current flow in both cases is proportional to $\exp(-E_a/k_BT)$, where E_a is the height of the energy barrier that controls the flow. The Arrhenius equation follows as long as E_a is a weak function of temperature. In transistors E_a is set by the gate-source voltage. As a result, current flow in a transistor with fixed gate-source voltage obeys the Arrhenius equation.

We can exploit the similar thermodynamic properties of chemical reactions and transistors. All rate constants in our model will satisfy the Arrhenius equation if the reference current I_0 increases exponentially with temperature. This behavior occurs if I_0 is produced by a transistor with temperature-independent gate-source voltage V_{GS0}. In this case, we get

$$I_0 = I_S e^{\kappa(V_{GS0}-V_T)/\phi_T} \tag{6.19}$$

where I_S and V_T are assumed to be weak functions of temperature. In general, however, our model will not reproduce *deviations* from Arrhenius-like behavior caused by temperature-dependent prefactors A and activation energy E_a. This is because the

analogous electronic quantities, namely I_S and V_T, need not have the same temperature dependence as A and E_a. In fact, activation energies can depend on temperature in complicated ways that have no direct electronic analog. Intuitively, this is because molecules may have several internal degrees of freedom that affect how they react with each other. For example, diatomic molecules can rotate about the bond linking the two atoms, a process which has its own characteristic dependence on temperature. Similarly, the threshold voltage V_T of a transistor is, in general, also a complicated function of temperature with no direct chemical analog.

The total potential energy of charged molecules (ions) is determined both by chemical and electrostatic potentials. Their combined effects are expressed via the *electrochemical potential*, which is defined as

$$\mu_{ec} = \mu_c + zqN_AV \tag{6.20}$$

where μ_c is the chemical potential, zq is the charge on the ion, N_A is Avogadro's number, and V is the voltage, or electrostatic potential. Since μ_{ec} is proportional to V, we can use voltage to exponentially speed up or slow down the rates of reactions that involve loss or gain of electrons (redox reactions). Such dependence of reaction rates on voltage is basic to the operation of batteries and fuel cells. We can model such effects by individually programming the rate constants of forward and reverse reactions. In addition, voltage gradients, i.e., electric fields, cause gradients in μ_{ec} that cause ions to flow via drift. However, such gradients are absent by definition in homogenous media. As a result, fluxes in well-mixed, homogenous media are purely diffusive in nature.

6.2.5 Polynomially Nonlinear Dynamical Systems

The circuit formulation described by the KCL equation in (6.11) can be extended to dynamically simulate any *polynomially nonlinear dynamical system*. Such a system can be used to model mass-action chemical kinetics of any order; it consists of a set of N differential equations of the form

$$\frac{dx_i}{dt} = \sum_{j=1}^{R} c_{ij} \left(x_1^{p_1} x_2^{p_2} ... x_N^{p_N} u_1^{q_1} u_2^{q_2} ... u_M^{q_M} \right) \tag{6.21}$$

where R is a positive integer, $[p_1 ... p_N]$ and $[q_1 ... q_M]$ are integers that in general are different for each value of i and j, the c_{ij}'s are real constants and, as before, we have N state variables x_i and M inputs u_i. Following the same procedure described in the previous section, equation (6.21) can be rewritten in terms of the rate of change of $\ln(x_i)$. The result, which is easier to implement in log-domain circuit form, is

$$\frac{d \ln(x_i)}{dt} = \sum_{j=1}^{R} c_{ij} \left(\frac{x_1^{p_1} x_2^{p_2} ... x_N^{p_N} u_1^{q_1} u_2^{q_2} ... u_M^{q_M}}{x_i} \right) \tag{6.22}$$

Equation (6.22) can be interpreted as KCL, i.e., the rate of change of $\ln(x_i)$, the voltage on a capacitor, is equal to the sum of R currents that add and subtract charge from it. Each term on the right hand side of (6.22) represents a current that is a multinomial function of the state variables and inputs. Log-domain circuits can easily implement such functions. Therefore any polynomially nonlinear dynamical system can be modeled using a dynamically equivalent log-domain circuit. The *order* S of the each term of the summation in equation (6.22) is defined as the sum of all the power-law coefficients in the numerator, i.e.,

$$S = \sum_{k=1}^{N} p_k + \sum_{k=1}^{M} q_k \tag{6.23}$$

The system of chemical reactions modeled by (6.11) is a special case of (6.22) when $S \in [0, 1, 2]$, i.e. only zeroth, first and second-order kinetics are allowed.

6.2.6 Multi-Compartment Models

So far, we have considered models of chemical reactions where all species reside within a single, well-mixed reaction compartment. A single compartment implies that its volume is constant for all species. This fact allows us to write *mass*-action rate laws in terms of concentrations, i.e., mass/volume, because the volume is simply a scalar. However, our circuits can also simulate more general, multi-compartment models.

Such models, where each compartment is assumed to be well-mixed, are common in biology. Cell-signalling pathways are an important example, where the extracellular environment, cell membrane, cytoplasm and nucleus are often considered to be separate compartments. Drug delivery is another example.

Systems that are not well-mixed exhibit spatial gradients in concentration, which cause mass transport due to diffusion. Such systems can be viewed as the continuous limit of multiple-component models. Our circuits can simulate these systems after they have been spatially discretized into chambers small enough to be approximated as well-mixed on the time scales of interest.

In multi-compartment models we cannot replace mass by concentration, because the volume is not a scalar. However, we can usually assume that all reactants that participate in a reaction are present in the same compartment. Consider the second-order reaction $A + B \rightarrow C$. Assuming mass-action kinetics and well-mixed compartments, the rate of production of C is given by

$$\begin{aligned} \frac{dM_C}{dt} &= k\frac{M_A M_B}{V_{AB}} \\ \Rightarrow \frac{d\left(V_C X_C\right)}{dt} &= kV_{AB}X_A X_B \end{aligned} \tag{6.24}$$

where M, V and X denote mass, volume and concentration, respectively, k is the usual rate constant and V_{AB} is the volume of the compartment containing A and B. For now, we assume that all reaction volumes are constant with time. In this case we can rewrite (6.24) as

$$\frac{dx_C}{dt} = \left(kX_0\frac{V_{AB}}{V_C}\right)x_A x_B \tag{6.25}$$

where we have normalized all concentrations by the constant reference concentration X_0, as before. We would like to simulate such equations using the translinear circuits described earlier. Firstly, we drop our earlier assumption that the value of the capacitor C is constant for all nodes. Here the voltage across C encodes the chemi-

cal potential of a chemical species. In this case we can modify (6.11) to encode the reaction $A + B \rightarrow C$ as

$$\frac{di_C}{dt} = \beta \left(\frac{\kappa I_0}{C_C \phi_T} \right) i_A i_B \tag{6.26}$$

where we have normalized all currents by the constant reference current I_0, as before, and β is a dimensionless number that can be programmed on-chip. In order to get dynamical equivalence with a speedup factor of α, we need

$$\beta \left(\frac{\kappa I_0}{C_C \phi_T} \right) = \alpha \left(k X_0 \frac{V_{AB}}{V_C} \right) \Rightarrow \beta = \alpha \tau_0 X_0 \left(\frac{V_{AB}}{V_C} \right) \left(\frac{C_C}{C_0} \right) k \tag{6.27}$$

where the time constant $\tau_0 \equiv C_0 \phi_T \, (\kappa I_0)$, and C_0 is a constant reference capacitance. We want to ensure that β is only a function of the reactants, since a single reaction can generate products in compartments with different volumes. This condition is satisfied by scaling capacitors with the volumes of the compartments where their respective molecular species are located, i.e., by using

$$\frac{C_i}{C_0} = \frac{V_i}{V_0} \tag{6.28}$$

where i is any species, and V_0 is a constant reference volume. In our chip the ratio C_i/C_0 can be set in discrete steps using a capacitor DAC. Using this scaling, (6.27) becomes

$$\beta = \alpha \tau_0 X_0 \left(\frac{V_{AB}}{V_0} \right) k \tag{6.29}$$

Equation (6.29) can be generalized to reactions of any order, giving us the following rule for setting β:

$$\beta = \alpha \tau_0 X_0^{S-1} \left(\frac{V_R}{V_0} \right) k \tag{6.30}$$

where S is the order of the reaction and V_R is the volume of the compartment containing the reactants. In certain situations we might also be interested in time-varying volumes. Modeling cell growth is a common and important example. In this case

(6.25) must be modified to

$$v_C \frac{dx_C}{dt} + x_C \frac{dv_C}{dt} = (kX_0 v_{AB})\, x_A x_B \tag{6.31}$$

where we have normalized all volumes by V_0. We see that the equation has an extra term that depends on the time-derivative of the normalized volume v_C. One might expect that the scaling rule in (6.28) would also apply if v_i was a function of time, i.e., that we could set $c_i(t) = v_i(t)$, where $c_i = C_i/C_0$ is the normalized capacitance. Firstly, we rewrite (6.11) by replacing Cdv/dt with the version required for applying KCL with time-varying capacitances, namely $dq/dt = Cdv/dt + vdC/dt$. After some algebra, we find that (6.26) must be modified to

$$c_C \frac{di_C}{dt} + i_C \ln(i_C) \frac{dc_C}{dt} = \left(\frac{\beta}{\tau_0}\right) i_A i_B \tag{6.32}$$

We see that (6.31) and (6.32) are dynamically equivalent once β is chosen in the usual way (using (6.29)), *except* for the additional $\ln(i_C)$ term in the latter equation. This unwanted term prevents us from accurately simulating systems where compartment volumes vary with time. The problem arises because in our analogy the voltage across a capacitor does not represent concentration, but free energy. As a result its charge does not represent mass, even when the capacitance is made proportional to compartment volume. This issue disappears in a voltage-mode implementation where capacitor voltages directly represent concentrations.

Equation (6.31) can be rewritten as

$$\frac{dx_C}{dt} + x_C \frac{d\ln(v_C)}{dt} = \left(kX_0 \frac{v_{AB}}{v_C}\right) x_A x_B \tag{6.33}$$

Consider the special case when both reactant and product volumes change exponentially with time, i.e., $\propto e^{\lambda t}$, where λ is a constant. In this case (6.33) can be simplified to

$$\frac{dx_C}{dt} + \lambda x_C = \left(kX_0 \frac{v_{AB}}{v_C}\right) x_A x_B \tag{6.34}$$

Since the ratio v_{AB}/v_C is constant with time, we see that the only effect of exponential growth or contraction is to add the extra term λx_C to the equation. This term causes first-order decay (if $\lambda > 0$) or growth (if $\lambda < 0$) of the product x_C with time. Its effects can be modeled by adding another reaction to the system, giving us $A + B \xrightarrow{k} C \xrightarrow{\lambda} \phi$, where ϕ is a "dummy" species [4]. Thus, we can model the special, but common case of exponential growth using our circuits.

6.2.7 Conservation Laws in Chemical Reactions

Consider a generic chemical reaction with N_R reactants and N_P products. We may represent the stoichiometry of such a reaction as

$$\sum_{i=1}^{N_R} r_i R_i \leftrightarrow \sum_{i=1}^{N_P} p_i P_i \tag{6.35}$$

The definition of chemical stoichiometry guarantees that, *irrespective of the form of the rate law*,

$$\frac{1}{r_1}\frac{dR_1}{dt} = \frac{2}{r_2}\frac{dR_1}{dt} = ... = \frac{N_R}{r_{N_R}}\frac{dR_{N_R}}{dt} = -\frac{1}{p_1}\frac{dP_1}{dt} = -\frac{1}{p_2}\frac{dP_2}{dt} = ... = \frac{2}{r_{N_P}}\frac{dP_{N_P}}{dt} \tag{6.36}$$

Note that we have dropped the square brackets around species concentrations. These relationships, which are more fundamental than the rate laws themselves, imply the presence of conservation laws of the form $d\left(R_i - R_j\right)/dt = 0$, $d\left(P_i - P_j\right)/dt = 0$ and $d\left(R_i + P_j\right)/dt = 0$, where i and j vary between 1 and N_R or N_P. Thus, the rates of change of concentration of every pair of chemical species participating in the reaction obey a conservation law. Either the difference between them, or their sum, is constant with time. The former occurs if they are both reactants or products, while the latter occurs if one is a reactant and the other a product. These laws hold irrespective of how many molecules are present, i.e., are valid in both deterministic and stochastic regimes. There are a total of N_{CL} such conservation laws, where

$$N_{CL} = \frac{1}{2}\left(N_R + N_P\right)\left(N_R + N_P - 1\right) \tag{6.37}$$

All chemical reactions can be decomposed into elementary molecular steps, which are limited to

1. **Transformation**: Reactions of the form $A \leftrightarrow B$, where $N_R = N_P = 1$ and $N_{CL} = 1$. The single conservation law is $(dA/dt) + (dB/dt) = \text{constant}$.

2. **Combination/Decomposition**: Reactions of the form $A + B \leftrightarrow C$, where $N_R = 2$, $N_P = 1$ and $N_{CL} = 3$. The three conservation laws are $d\left(A/dt\right) - (dB/dt) = \text{constant}$, $dA/dt + dC/dt = \text{constant}$, and $(dB/dt) + (dC/dt) = \text{constant}$.

3. **Combination/Decomposition with Indistinguishable Species**: Reactions of the form $A \leftrightarrow B + B$, i.e., $A \leftrightarrow 2B$, where $N_R = N_P = 1$ and $N_{CL} = 1$. The single conservation law is $2\left(dA/dt\right) + (dB/dt) = \text{constant}$.

In order to model chemical reactions with electronic circuits we have to deal with a smaller "basis set" of interactions than these elementary molecular steps. All electronic computations must be performed with electrons, which, unlike molecular species, are indistinguishable. Electrons are also indestructible. As a result, their total number, i.e., the electronic charge, is conserved. This is the only conservation law obeyed by electronic circuits. Because electronic circuits use only one type of charge, we must encode the diversity of chemical species using some other property of the circuit. The most obvious way is to use the current location of an electron to encode its simulated chemical identity, which is why we have used node voltages or branch currents to represent concentrations. However, it is now evident that we must also ensure that our electronic circuit satisfies all relevant chemical conservation laws.

Assume that we are representing concentrations using node voltages. The question is if a purely passive circuit, consisting of a collection of capacitors connected together by two-terminal elements, will suffice. The elements may be linear or non-linear. However, because of charge conservation, the currents entering and leaving them must

be equal and opposite. In other words, a two-terminal element connected between two nodes A and B always ensures that $i_A = -i_B$, where positive currents leave nodes by convention. Within this simulation framework, the three elementary molecular steps described earlier imply the following relationships:

1. **Transformation**: Reactions of the form $A \leftrightarrow B$ require $i_A + i_B = 0$.

2. **Combination/Decomposition**: Reactions of the form $A + B \leftrightarrow C$ require $i_A - i_B = 0$, $i_A + i_C = 0$, and $i_B + i_C = 0$.

3. **Combination/Decomposition with Indistinguishable Species**: Reactions of the form $A \leftrightarrow B + B$, i.e., $A \leftrightarrow 2B$, require $2i_A + i_B = 0$.

We see that two-terminal circuit elements only satisfy conservation laws of the form $i_X + i_Y = 0$, which only occur when X is a reactant in a chemical reaction and Y a product, or vice versa. This is enough for simulating chemical reaction networks consisting only of transformations. They cannot satisfy laws of the type $i_X - i_Y = 0$, which occur when both X and Y are reactants or products, and so cannot model chemical combination or decomposition. In order to model such reactions we need a circuit element that satisfies $i_X = i_Y$. Such an element is known as a *current copier*, and is commonly realized using two transistors configured as a current mirror. Current copiers are active circuits, because they require a power supply to operate. Power supply nodes, usually denoted by V_{DD} and ground, are sources of free energy, and can sink and source currents without changing their own potentials. This fact allows us to generate conservation laws, such as $i_X = i_Y$, which do not conserve the total amount of electronic charge.

Elementary molecular steps of the third type require a conservation law of the form $2i_X + i_Y$ to be satisfied. This law also cannot be constructed out of two-terminal elements, and needs *current amplifiers* with gains of 2 and 1/2. Such amplifiers are active circuits that require a power supply to operate. Thus, *irrespective of the rate laws and kinetics*, simulating chemical reactions electronically requires, at a minimum, two-terminal elements, current copiers and current amplifiers. The problem is that

while passive elements like diodes and resistors behave as two-terminal elements with very high precision, the active circuits required to build current copiers and amplifiers are fundamentally imprecise. These circuits rely on matched, or identical, circuit components for their operation, but real circuit components always mismatch by finite amounts.

The situation is even worse for translinear circuits, because they use active circuits to implement all conservation laws, including those of the form $i_X + i_Y = 0$. As a result, such laws will only be satisfied on-chip to a finite level of precision, i.e., will be subjected to mismatch. The situation for the three types of conservation laws can be represented as

$$\begin{aligned} i_X + (1+\epsilon) i_Y &= 0 \\ i_X - (1+\epsilon) i_Y &= 0 \\ 2i_X + (1+\epsilon) i_Y &= 0 \end{aligned} \tag{6.38}$$

where the relative, or fractional error, denoted by ϵ, is a random variable that is typically assumed to have a Gaussian probability distribution. Finally, we note that these conservation laws should be individually satisfied by the fluxes of every reaction in the system.

The only equilibrium state for a passive electronic circuit occurs when all nodes reach a common potential, which we usually designate as ground. One might similarly expect a chemical system to reach equilibrium when all species have equal chemical potentials. Equation (6.8) predicts that, in dilute solutions where the law of mass action is valid, the reference potential μ_0 and reference concentration X_0 are the same for all molecular species. Thus equal chemical potentials imply equal concentrations. However, we know that species concentrations are *not* equal at chemical equilibrium, implying that they have different chemical potentials. For example, at equilibrium forward and backward reactions have non-zero, but equal fluxes, so the reversible reaction $A \underset{k_-}{\overset{k_+}{\rightleftharpoons}} B$ satisfies

$$K_d \equiv \frac{k_-}{k_+} = \frac{A_{eq}}{B_{eq}} \Rightarrow \Delta\mu_{eq} = (\mu_{B,eq} - \mu_{A,eq}) = -k_B T \ln(K_d) \tag{6.39}$$

where K_d is the dissociation constant of the reaction, and eq subscripts denote quantities at equilibrium. We see that the equilibrium chemical potentials of A and B are not equal unless $K_d = 1$, i.e., $k_+ = k_-$. Thus the equilibrium state of a passive electronic circuit cannot emulate chemical equilibrium, unless all forward and backward reaction rates are equal. For example, two terminal devices can only emulate transformation reactions where $k_+ = k_-$. Thus, in general active electronic circuits with power supplies are required to simulate chemical reactions. Conceptually, we can endow each simulated reaction with its own battery, or power supply voltage $\Delta v_{eq} = -k_B T \ln(K_d)/q$, thus creating a distributed power supply. For practical reasons, however, we we use a common power supply voltage for the entire circuit. Transistor-based circuits generate effective, or "virtual" batteries that emulate the distributed power supply.

6.2.8 Effects of Mismatch

Consider the unidirectional first-order reaction $A \rightarrow B$, and assume that the rate constant is equal to k. The reaction conserves the total amount of A and B, since $d([A]+[B])/dt = -k[A] + k[A] = 0$. The conserved quantity arises from the physical fact that the creation of a molecule of B implies the destruction of a molecule of B. However, when this reaction is simulated electronically the flux (current) leaving the node that represents $[A]$ (or its associated free energy) will in general not be precisely equal to the flux entering the node that represents $[B]$. This issue arises because of mismatch between the transistors used to carry the fluxes, and is unavoidable in any real electronic circuit. Circuit design techniques can reduce the amount of mismatch, but not eliminate it entirely. The only way to completely avoid mismatch is to use the *same* transistor, or other two-terminal device) to shuttle charge between the two nodes. However, this is not always possible, as discussed earlier.

In general, therefore, we will be left with some fractional mismatch between the

fluxes, which we denote by ϵ. The equations describing the reaction can be written in matrix form as

$$\begin{bmatrix} \dot{A} \\ \dot{B} \end{bmatrix} = \begin{bmatrix} -k \\ (1+\epsilon)k \end{bmatrix} A \tag{6.40}$$

where, for convenience, we have dropped the square brackets denoting concentrations. We see that the mathematical effect of mismatch is to make the column vector on the right hand side not sum up to zero. Solving these equations gives us

$$\begin{aligned} A(t) &= A(0)e^{-kt} \\ B(t) &= B(0) + (1+\epsilon)A(0)\left(1 - e^{-kt}\right) \end{aligned} \tag{6.41}$$

where $A(0)$ and $B(0)$ are the initial concentrations of A and B. We see that the sum of A and B is not conserved with time, but is instead given by

$$A(t) + B(t) = A(0) + B(0) + \epsilon A(0)\left(1 - e^{-kt}\right) \tag{6.42}$$

In this case the consequences of mismatch do not appear to be too serious. The sum of A and B in steady state ($t \to \infty$) is increased by an amount $\epsilon A(0)$ over its initial value. However, now consider adding the reverse reaction $B \to A$. The new system of reactions is $A \leftrightarrow B$. The rate constants and mismatches are k_+ and k_-, and ϵ_+ and ϵ_-, respectively. The dynamics are described by the following set of equations

$$\begin{bmatrix} \dot{A} \\ \dot{B} \end{bmatrix} = \begin{bmatrix} -k_+ & (1+\epsilon_-)\,k_- \\ (1+\epsilon_+)\,k_+ & -k_- \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \tag{6.43}$$

Again, flux mismatch causes the columns of the matrix on the right hand side, which we denote by K, to not sum up to zero. The original reaction (with no mismatch) has a single stable steady state where $A/B = K_d$, and $K_d = k_-/k_+$ is the dissociation constant. Equation (6.43), however, has *no steady-state solutions*, unless,

by some miracle, we have $(1+\epsilon_-)(1+\epsilon_+) = 1$[3].

Since the system is linear we can gain further insight by finding the eigenvalues of K. After much algebra, we find that

$$\lambda_{1,2} = \frac{(k_+ + k_-)}{2}\left[1 \pm \sqrt{1+\chi}\right] \tag{6.44}$$

where χ is defined as

$$\chi = \frac{4k_+k_-}{(k_+ + k_-)^2}(\epsilon_+ + \epsilon_- + \epsilon_+\epsilon_-) \tag{6.45}$$

Assuming that $|\chi| \ll 1$, (6.44) can be simplified to

$$\begin{aligned} \lambda_{1,2} &\approx (k_+ + k_-), \; -(k_+ + k_-)\frac{\chi}{4} \\ &= (k_+ + k_-), \; -\frac{k_+k_-}{(k_+ + k_-)}(\epsilon_+ + \epsilon_- + \epsilon_+\epsilon_-) \end{aligned} \tag{6.46}$$

We see that there are two non-zero eigenvalues, indicating that in general $A(t)$ and $B(t)$, which are given by $c_1e^{-\lambda_1 t} + c_2e^{-\lambda_2 t}$ where c_1 and c_2 are constants, have no steady-state solutions except 0 or ∞. The two eigenvalues correspond to two modes: λ_1 is the normal, or desired mode, while λ_2 is an undesirable "slow" mode. In fact, $\lambda_2/\lambda_1 = \chi/4$, so that this mode becomes slower as the mismatches decrease in magnitude and χ becomes smaller. When $\epsilon_+ = \epsilon_- = 0$, $\chi = 0$ and we get $\lambda_2 = 0$, which gives us the usual steady state solution.

Our simulation will be valid if the fast mode is much faster than the slow mode, i.e., if $|\chi| \ll 1$. In this case the system will reach a quasi-steady state, where the simulation can be halted, before the slow mode has had time to act. Therefore $|\chi|$ should be as small as possible. Examining (6.45), we find that χ is the product of two parts: χ_1, which depends only on the rate constants, and $\chi_2 = (\epsilon_+ + \epsilon_- + \epsilon_+\epsilon_-)$, which depends only on the mismatches. We can rewrite χ_1 as

[3] Assuming that $|\epsilon| \ll 1$, this condition reduces to $\epsilon_- = -\epsilon_+$.

$$\chi_1 = \frac{4k_+k_-}{(k_+ + k_-)^2} = \frac{4}{\left(\sqrt{k_+/k_-} + \sqrt{k_-/k_+}\right)^2} \tag{6.47}$$

Equation (6.47) shows that if $k_+ \gg k_-$, we get $\chi_1 \approx 4(k_-/k_+) \ll 1$. Similarly, if $k_- \gg k_+$, we get $\chi_1 \approx 4(k_+/k_-) \ll 1$. The maximum value of $\chi_1 = 1$ occurs when $k_+ = k_-$. Thus, for a given amount of mismatch, $|\chi|$ and the effects of the slow mode are minimized when one rate constant is much larger than the other. Each reversible reaction in a system of chemical reactions that is being electronically simulated will have its own slow mode. We can guarantee the accuracy of the simulation if the fastest of these undesirable modes is much slower than the slowest mode of interest. Fundamentally, this requirement limits the range of rate constants that can be successfully simulated. In general ϵ_+ and ϵ_- are set by transistor geometry and do not scale with k_+ and k_-. Therefore the allowable dynamic range of rate constants is approximately $1/\chi_2 \propto 1/\epsilon$.

Further insight into the situation can be obtained by rewriting (6.43) as follows

$$\begin{bmatrix} \dot{A} \\ \dot{B} \end{bmatrix} = \begin{bmatrix} -(1+\epsilon_+)\,k_+ & (1+\epsilon_-)\,k_- \\ (1+\epsilon_+)\,k_+ & -(1+\epsilon_-)\,k_- \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} + \begin{bmatrix} \epsilon_+ k_+ \\ -\epsilon_- k_- \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \tag{6.48}$$

We see that the rate constant matrix can be broken up into two parts. The first part consists of an "ideal" reversible reaction with no mismatch[4]. The second part consists of either first-order creation or degradation reactions, depending on the signs of ϵ_- and ϵ_+. These additional reactions cause exponential growth or decay of A and B, and are responsible for the so-called "slow mode". In the electrical circuit equivalent, mismatch causes an unwanted positive or negative-valued resistor to appear between node A and ground. A similar phenomenon occurs at node B.

We shall now describe the effects of mismatch on a slightly more complicated set of reactions, namely Michaelis-Menten kinetics. This set of reactions adequately describes the kinetics of many common enzymes, and is given by

[4]Notice that both columns in the first matrix sum to zero.

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_{cat}} E + P \tag{6.49}$$

where E, S, ES, and P represent the concentrations of enzyme, substrate, enzyme-substrate complex, and product, respectively. The rates of the reversible reaction are denoted by k_1 and k_{-1}, and that of the irreversible reaction by k_{cat}. We assume that mismatch is present in each reaction, and denote the fractional mismatches by ϵ_1, ϵ_{-1}, ϵ_{cat} and ϵ'_{cat}, respectively. Finally, we also assume that the substrate concentration S is constant. This assumption is valid at the beginning of the reaction when the substrate has not been appreciably depleted. The resultant differential equations are given by

$$\begin{bmatrix} \dot{E} \\ \dot{ES} \\ \dot{P} \end{bmatrix} = \begin{bmatrix} -k_1[S] & (1+\epsilon_{-1})\,k_{-1} + (1+\epsilon_{cat})\,k_{cat} & 0 \\ (1+\epsilon_1)\,k_1[S] & -(k_{-1}+k_{cat}) & 0 \\ 0 & (1+\epsilon'_{cat})\,k_{cat} & 0 \end{bmatrix} \begin{bmatrix} E \\ ES \\ P \end{bmatrix} \tag{6.50}$$

To solve this set of equations we make the usual quasi-steady-state assumption (QSSA) that $d[ES]/dt = 0$. This assumption is valid if the following statements are true:

1. The time taken for ES to reach its QSSA value is much shorter than the overall time for the substrate to be depleted

2. Negligible substrate is consumed during the approach of ES to its QSSA value

It can be shown that both conditions are satisfied if

$$\frac{E_0}{K_m + S_0} \ll 1 \tag{6.51}$$

where E_0 and S_0 are the initial concentrations of the enzyme and substrate, respectively, and $K_m \equiv (k_{-1} + k_{cat})/k_1$ is known as the Michaelis-Menten constant of the enzyme. In our case, we further assume that the magnitudes of the mismatch terms

are much smaller than one. After much algebra, the rate of product formation is found to be

$$\begin{aligned} v \equiv \frac{dP}{dt} &= k_{cat}\left(1+\epsilon'_{cat}\right)ES \\ &= v_{max}\left(\frac{S}{K_m+S}\right)e^{k_\epsilon t} \end{aligned} \tag{6.52}$$

where

$$\begin{aligned} k_\epsilon &= k_1 S\left(\epsilon_1 + \frac{k_{-1}\epsilon_{-1}+k_{cat}\epsilon_{cat}}{k_{-1}+k_{cat}}\right) \\ v_{max} &= k_{cat}\left(1+\epsilon'_{cat}\right)E_0 \\ K_m &= \frac{k_{-1}+k_{cat}}{k_1\left(1+\epsilon_1\right)} \end{aligned} \tag{6.53}$$

We see that the rate law has the usual Michaelis-Menten form, except for the exponential term. Thus, the effects of the mismatch are to cause small errors in the values of v_{max} and K_m, and, much more importantly, to introduce an unwanted "slow" mode with rate constant k_ϵ. Its properties are very similar to that of the slow mode in the simple reversible reaction analyzed earlier. Firstly, k_ϵ consists of the product of two terms: the first, equal to $k_1 S$, is independent of the mismatch terms, while the second is a weighted sum of them. Secondly, the mode causes species concentrations to blow up or die with time, depending on the signs of the mismatch terms. Finally, dynamic simulations will be valid on timescales where the mode has not had time to act, i.e., when $t \ll 1/k_\epsilon$.

To summarize, mismatch limits the stiffness of the differential equations that can be simulated electronically. Thus, the effects of mismatch are analogous to the effects of finite-precision arithmetic on simulations performed by digital computers. Transistor mismatch can be reduced in two ways: passive and active. Passive techniques do not require changes to the circuits themselves. For example, the transistors can be made physically larger, and also laid out closer to each other on the die [111].

To first order, the standard deviation between two similar MOS transisors scales as $1/\sqrt{WL}$, where WL is their layout area [223]. Active techniques include the use of fully-differential circuit topologies, and also dynamic matching schemes where the circuit uses one of an array of transistors at any given time, and rapidly switches between them.

6.2.9 Noise in Chemistry and Electronics

Individual chemical reaction events are usually uncorrelated. As a result, molecular fluxes exhibit shot noise. This behavior is exactly analogous to electronic shot noise, which is caused by diffusion currents within physical devices. In both electrical circuits and chemistry, thermodynamic equilibrium, which results in zero net flux, does not result in zero noise. Instead, equal and opposite reaction fluxes diffuse independently and cause fluctuations in the concentrations of electrons and molecules. In both cases, individual fluxes usually consist of uncorrelated events, and thus exhibit Poisson statistics[5].

We want chemical and electronic state variables that behave identically in the high SNR or deterministic limit to also have identical noise properties, thus allowing our circuits to peform fast, accurate stochastic simulations. This ability is important because while noise has important effects in many biological systems, noisy systems are numerically stiff and simulate slowly on digital computers.

In log-domain circuits noisy fluxes (currents) do not directly act on a state variable, i.e., species concentration. Instead, they add or subtract charge from a capacitor, the voltage on which is log-compressed, i.e., must be exponentiated to get a current that is the state variable. Because this operation is nonlinear, positive and negative fluxes that affect state variables will not display Poisson statistics, *unless* the log-compressed fluxes are somehow 'predistorted'. We shall describe how translinear circuits naturally generate such predistortion.

Chemical conservation laws arise at the microscopic level because of interactions between molecules with distinct identities. For example, in the reaction $A \leftrightarrow B$,

[5]Exceptions include flows caused by drift, such as in above-threshold MOSFETs.

reaction fluxes obey the conservation law $(dN_A/dt) + (dN_B/dt) = 0$ because, microscopically, the production of a molecule of B implies the destruction of a molecule of A, and vice versa. Thus, in the absence of other reactions, $N_A + N_B =$ constant and fluctuations in A and B, which we denote by ΔN_A and ΔN_B, respectively, are completely anti-correlated *on all time scales*. Mathematically, $(N_A + \Delta N_A + N_B + \Delta N_B) = (N_A + N_B) \Rightarrow \Delta N_A = -\Delta N_B$.

We should emphasize that conservation laws for reaction fluxes only cause correlated fluctuations in concentrations (or numbers of molecules) in the absence of other reactions. Consider the chain of reactions $A_1 \leftrightarrow A_2 \leftrightarrow ... \leftrightarrow A_n ... \leftrightarrow A_{N+2}$, where we have interposed N intermediate species between A_1 and A_{N+2}. Clearly A_1 and A_{N+2} will display anti-correlated fluctuations if $N = 0$. However, these correlations should decrease in magnitude as N increases. The reactions that produce A_1 and consume A_{N+2}, for example, will occur at uncorrelated times, leading to reaction fluxes that do not fluctuate in a correlated way.

Figure 6-4 shows the simulated cross-correlation between A_1 and A_{N+2} in steady state as a function of N. All forward and backward reactions were assumed to have the same rate constant k. The plot confirms our expectations. It shows that A_1 and A_{N+2} are ideally anti-correlated (correlation coefficent of -1) for $N = 0$, and become increasingly decorrelated as N increases. Plots are shows for various values of N_f, the average number of molecules of each species A_n in steady state. We see that the correlation coefficient is essentially independent of N_f, and depends only on N. These simulations were performed using an optimized version of Gillespie's stochastic simulation algorithm [90].

Electronic analogs of chemistry use the location of electrons (circuit node or branch) to encode molecular identity. We assume that concentrations are represented by the number of electrons on capacitors. There are two basic ways for an electronic analog of a chemical reaction to change the amount of charge at various nodes in a circuit. Firstly, two-terminal elements can physically transport electrons between different nodes. Such transport occurs via drift and diffusion in resistors and transistors, and takes a finite amount of time. However, we have seen previously that

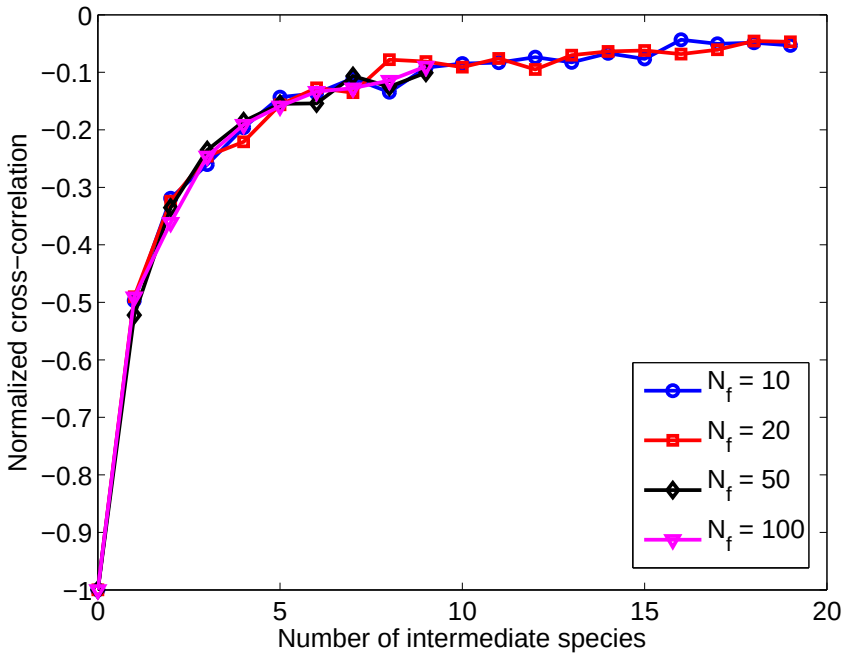


Figure 6-4: Correlation between two chemical species as a function of the number of intermediate species between them, as predicted by the Gillespie stochastic simulation algorithm.

this scheme can only implement transformation reactions.

For example, consider an electron that has to move between nodes A and B to emulate the single chemical reaction $A \leftrightarrow B$. It takes a time τ to do so, where τ is approximately equal to the diffusion time through the charge-transfer device. Because of charge conservation we have $Q_A + Q_B + Q_T = \text{constant}$, where Q_T is the charge stored in the transport device, rather than the desired $Q_A + Q_B = \text{constant}$. The effects of Q_T become significant on time scales smaller than τ. As a result, correlated fluctuations ony occur on time scales longer than τ, the transport delay. In transistors, $\tau \approx 1/\omega_T$, where ω_T is the unity-gain frequency.

The second scheme uses active circuits and needs a power supply. The basic idea can be illustrated by trying to construct the conservation law $(dQ_A/dt) = (dQ_B/dt)$, i.e., $i_A = i_B$, for the reaction $A + B \leftrightarrow C$. One way to enforce this constraint is to first draw a current i_A from node A and convert it to a voltage, which is then used to

control the amount of current i_B drawn from node B[6]. Thus, fluctuations in i_A affect i_B, but not vice-versa. As a results, fluctuations in i_A and i_B are not fully correlated.

The noise sources associated with different transistors are uncorrelated as long as they act as 'small-signal' perturbations, i.e., as long as the SNR is not too low. Active circuits must use several transistors to implement the rate laws in chemical reactions. As a result, it is very difficult to build circuits that accurately reproduce correlated fluctuations in reaction fluxes, as shown in the previous example. In practice, however, this may not be an important limitation, for two reasons. Firstly, we shall show that active circuits can match the power spectral density (PSD) of fluctuations produced by chemical reactions. PSD ignores correlations, or phase information, but this level of realism may be enough for many applications. Secondly, transport delays due to diffusion are also ubiquitous in chemical reactions. These random delays make correlated fluctuations difficult to measure experimentally.

6.3 Design of the Chemical Network Chip

6.3.1 Translinear Circuits

Let us re-examine (6.8), which defines the chemical potential. In our model chemical potential energy maps to electrical potential energy, i.e., μ_i maps to qv_i. Similarly, concentrations map to currents, so x_i maps to current i_i. Chemical potential differences generate "concentration currents", while electrical potential differences generate electronic currents. Since the current increases exponentially with potential difference, (6.8) is equivalent to a diode.

Chemical reactions create molecular fluxes that depend on species concentrations and change them. In our model, reactions correspond to current-dependent changes in current, i.e., $di_i/dt = f(i_1, i_2, ...)$, where $f()$ is any function. The most obvious way to implement a reaction is to use a a current-controlled voltage source, i.e., transresistance:

[6]What we have described here is simply a current mirror!

$$v_L = Lf\left(i_1, i_2, ...\right) \tag{6.54}$$

where L is a constant that gives the equation the right dimensions. The output voltage of the transresistance, i.e., v_L is applied across an inductance of value L that carries the current i_i:

$$\frac{di_i}{dt} = \frac{v_L}{L} = f\left(i_1, i_2, ...\right) \tag{6.55}$$

However, such a scheme is inconvenient for integrated circuit implementations for two reasons. Firstly, passive inductors are hard to build below RF frequencies, while active inductors are noisy and have limited dynamic range. Secondly, transistors are inherently voltage-controlled current sources, i.e., transconductors, and not current-controlled voltage sources, i.e., transresistors. Therefore, we need another way to generate di_i/dt terms. Fortunately, the *dynamic translinear principle* comes to the rescue. Subthreshold FETs and BJTs are exponential transconductances, i.e.,

$$i_i = I_0 \exp\left[\frac{\kappa\left(v_i - V_0\right)}{\phi_T}\right] \tag{6.56}$$

where v_i is the gate-to-source voltage, and I_0, V_0 and κ are constants. As a result, di_i/dt can be replaced by the product of two currents: i_i itself, and the current $i_C = Cdv_i/dt$ that flows through a capacitor C connected between the gate and source terminals:

$$\frac{di_i}{dt} = \frac{i_C i_i}{C\phi_T} \tag{6.57}$$

Therefore the problem of generating di_i/dt as a function of other currents has been replaced by the easier problem of generating i_C as a function of these currents and i_i itself. By inverting (6.56) we get

$$v_i = v_0 + \frac{kT}{\kappa q}\ln\left(\frac{i_i}{I_0}\right) \tag{6.58}$$

Since the mathematical forms of (6.58) and (6.8) are identical, v_i simply represents

$\mu_i/(\kappa q)$. Thus another advantage of the translinear circuit model is that it gives us access to both species concentrations and their associated chemical potentials. Equations (6.58) and (6.8) are identical because they are both caused by particle concentrations that follow exponential, i.e., Boltzmann probability distributions on energy landscapes. Thus, at thermodynamic equilibrium, concentration $\propto \exp(-E_i/kT)$, where E_i is the energy of species i. For chemical reactions $E_i = \mu_i = \delta G/\delta x_i$, where G is the Gibbs free energy of the system. Similarly, for electronics $E_i = \kappa q v_i$.

6.3.2 Reaction Circuits

We reference all state variables to V_{DD} since PMOS transistors are our exponential elements. We use the translinear, or log-domain integrator proposed in [231] as our primary building block. This circuit is guaranteed to be stable at all current levels and can be implemented on low power-supply voltages. Each integrator only needs to be unidirectional, since it models how flux from an unidirectional reaction changes the concentration of one species. In other words, the capacitor storing the chemical potential of the species is either charged or discharged by a current $i_C = \beta i_1 i_2/i_i$, depending on whether the species is a product or a source, respectively. In some cases one of the inputs (i_1 or i_2) to the integrator is equal to the output i_i. In these cases i_C simplifies to either i_1 or i_2 and the integrator can be replaced by a current mirror. A complete reaction is modeled by using an integrator or current mirror for every participating species.

Both transient and steady-state behaviors of chemical networks can be simulated using our circuits, since the circuit equations shown in (6.11) and (6.12) are dynamically equivalent to the original chemical equations. Dynamical equivalence refers to the fact that the dynamics of *normalized* chemical and electrical state variables, i.e., x_i/X_0 and i_i/I_0, respectively, are identical. However, in order to simulate typical biochemical time constants of seconds to hours rapidly, our electronic circuits should be dynamically equivalent, not to the chemical dynamics themselves, but a time-scaled (sped-up) version of them. In order to get a speedup factor of α, normalized electronic state variables i_i/I_0 must have time derivatives that are α times larger than

their chemical equivalents. As a result, the dimensionless number β that scales the capacitor current i_C is given by

$$\beta = \alpha\tau_0 X_0^{S-1} k \tag{6.59}$$

where the characteristic electronic time constant $\tau_0 = C\phi_T/(\kappa I_0)$, $S \in [0, 1, 2]$ is the order of the chemical reaction, and k is its kinetic rate constant. Consider a reversible reaction with forward and backward rate constants k_1 and k_2, respectively. The ratio β_2/β_1 is proportional to the equilibrium dissociation constant $K_d = k_2/k_1$ of the reaction:

$$\frac{\beta_2}{\beta_1} = \left(\frac{k_2}{k_1}\right) X_0^{S_2-S_1} = K_d X_0^{S_2-S_1} \tag{6.60}$$

where S_1 and S_2 are the orders of the forward and backward reactions, respectively. From (6.11), a single unidirectional second-order reaction of the form $A + B \rightarrow C$ is described by the following equations:

$$\begin{aligned} C dv_A/dt &= +\beta i_A i_B/i_A = +\beta i_B \\ C dv_B/dt &= +\beta i_A i_B/i_B = +\beta i_A \\ C dv_C/dt &= -\beta i_A i_B/i_C \end{aligned} \tag{6.61}$$

Note that the signs of the currents have been reversed since the state variable is now referenced to the constant voltage V_{REF}, i.e., given by $V_{REF} - v_i$. As a result increasing v_i decreases the state variable, and vice-versa. The first two equations in (6.61) require current mirrors, while the third requires a log-domain integrator. A simplified circuit implementation is shown in Figure 6-5. In order to ensure that all transistors remain saturated we usually set V_{REF} to a value that is a few hundred mV below V_{DD}. The W/L ratio of some transistors, indicated in the figure, are made β_1 and β_2 times larger than the other transistors using binary-weighted N-bit transistor arrays ($N = 5$ in this implementation). Therefore β_1 and β_2 can vary between 1 and

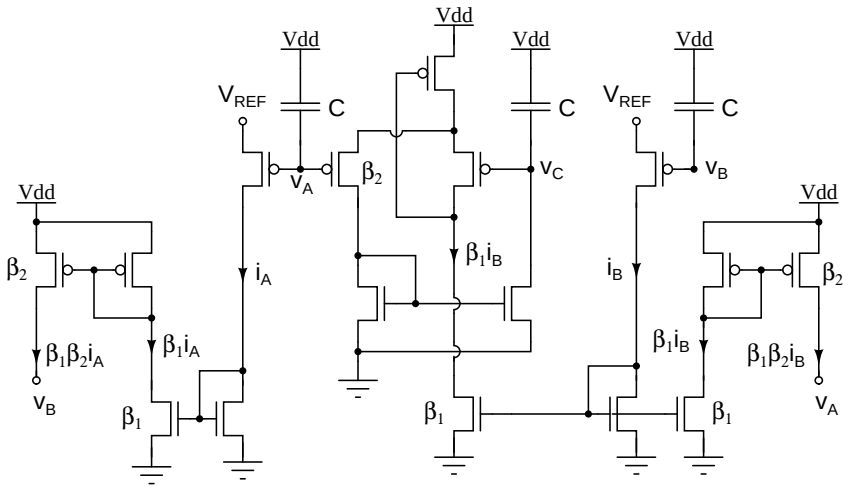


Figure 6-5: Simplified schematic of a circuit that models a second-order chemical reaction.

2^N, and $\beta = \beta_1\beta_2$ between 1 and 2^{2N}. Not all numbers between 1 and 2^{2N} can be obtained in this way. A detailed analysis is provided in Appendix E.

Once the chemical rate constants k are given, α, τ_0 and X_0 must be chosen such that β for all reactions falls within this range. We also add a fixed current I_{min} to $\beta_1 i_B$ in the actual implementation to ensure that i_B does not become small enough for parasitic capacitances inside the integrator to noticeably affect the dynamics of the state variables, i.e., A, B and C. An additional integrator and current mirror (not shown) is used to remove the effect of I_{min}, as follows:

$$\begin{aligned} C dv_A/dt &= +\beta_2 \left(\beta_1 i_B + I_{min}\right) - \beta_1\beta_2 I_{min} \\ &= +\beta_1\beta_2 i_B \\ C dv_C/dt &= -\beta_2 \left(\beta_1 i_B + I_{min}\right) i_A/i_C + \beta_2 I_{min} i_A/i_C \\ &= -\beta_1\beta_2 i_A i_B/i_C \end{aligned} \tag{6.62}$$

A first-order reaction $A \rightarrow B$ is defined by the following equations:

$$Cdv_A/dt = +\beta I_0 i_A/i_A = +\beta I_0$$
$$Cdv_B/dt = -\beta I_0 i_A/i_B \tag{6.63}$$

These equations can be implemented with a current mirror and an integrator. However, since the value of the constant current I_0 is known *a priori*, I_{min} is not needed. This fact simplifies the circuit implementation. Finally, a zeroth-order reaction $[\,] \rightarrow A$, where the species A is produced by an external flux (current source) is defined by the equation

$$Cdv_A/dt = -\beta I_0^2/i_A \tag{6.64}$$

This equation can be implemented with a single integrator, and again, I_{min} is not needed. The value of β for both first and zeroth-order reaction circuits is set in a similar way to the second-order case, i.e., by factorizing β into β_1 and β_2, which are set by binary-weighted transistor arrays. We can now combine reaction circuits of various types to implement arbitrarily complicated systems of chemical reactions.

6.3.3 Scaling

We want to fit as much of the dynamic range in reaction rates as possible into the limited dynamic range of the on-chip dimensionless scaling variable β, which is programmed by transistor geometry. The value of β is given by

$$\beta = \beta_{min}\,(D_1 + 1)\,(D_2 + 1) \tag{6.65}$$

where D_1 and D_2, which represent the DAC codes used to program β, are integers between 0 and $2^N - 1$. In the current chip we used $N = 5$. Thus, a limited set of β values is available. Given a desired value of β we can always pick D_1 and D_2 such that the fractional error η is minimized, where η is given by

$$\eta = \left| \frac{\beta/\beta_{min}}{(D_1+1)(D_2+1)} - 1 \right| \tag{6.66}$$

The relationship between β and the rate constant k is given by (6.59). We shall denote β/k for a reaction of order $S \in [0,1,2]$ by $c_S = \alpha\tau_0 X_0^{S-1}$, where α is the speedup factor, X_0 is the reference concentration, and $\tau_0 = C\phi_T/(\kappa I_0)$. The definition of c_S guarantees that the following relationship is always true:

$$c_1^2 = c_0 c_2 \tag{6.67}$$

Mathematically, we have to pick the values of the three coefficients c_0, c_1 and c_2 so that we know what values of β to use for given values of k. The coefficients depend on two variables, $\alpha\tau_0$ and X_0. Once the coefficients are known we can determine the values of $\alpha\tau_0$ and X_0 using the relationships

$$\begin{aligned} \alpha\tau_0 &= c_0 X_0 = c_1 = c_2/X_0 \\ X_0 &= c_1/c_0 = c_2/c_1 = \sqrt{c_2/c_0} \end{aligned} \tag{6.68}$$

On this chip the minimum and maximum values of β were $\beta_{min} = 1/2^4$ and $\beta_{max} = 2^6$, respectively, resulting in a dynamic range of $2^{10} : 1 = 1024 : 1$. We also denote the geometric mean of β_{min} and β_{max} by $\overline{\beta} = \sqrt{\beta_{min}\beta_{max}} = 2$. In the simplest scaling strategy, c_S is defined to be

$$c_S = \frac{\overline{\beta}}{\overline{k_S}} \tag{6.69}$$

where $\overline{k_S}$ is the geometric mean of all the rate constants of reactions with order S[7]. Values of k that result in $\beta < \beta_{min}$ or $\beta > \beta_{max}$ will be clipped. This strategy simultaneously aligns the center of the dynamic ranges of β and k for all values of S. As a result the maximum possible range of k values can be accommodated on

[7]Other measures of centrality of a distribution can also be used. One which we have found particularly useful is the geometric mean of the maximum and minimum rate constants that have to be implemented.

chip. However, the strategy does not work if zeroth, first and second order reactions are simultaneously present on the chip. This restriction arises because the three coefficient values cannot be picked independently; they are related by (6.67). In this case we modify (6.69) as follows

$$c_S = \frac{\overline{\beta}}{\gamma_S \overline{k_S}} \tag{6.70}$$

where γ_S is a constant which determines the extent to which the centers of the dynamic ranges of β and k are misaligned. Ideally $\gamma_S = 1 \; \forall S$, so there is no misalignment, as in (6.69). Combining (6.67) and (6.70), we get

$$\frac{\gamma_1}{\sqrt{\gamma_0 \gamma_2}} = \frac{\sqrt{\overline{k_0 k_2}}}{\overline{k_1}} \tag{6.71}$$

Based on symmetry arguments, we devised the following heuristic rules for picking γ_S, which work well in practice: $\gamma_1 = 1/\sqrt{\gamma_0 \gamma_2}$, and $\gamma_0 = \gamma_2$. These choices allow us to write

$$\gamma_0 = \frac{1}{\gamma_1} = \gamma_2 = \frac{\overline{k_1}^{1/2}}{\left(\overline{k_0 k_2}\right)^{1/4}} \tag{6.72}$$

In order to complete our on-chip implementation of the chemical reaction model, we also need to set initial conditions. We want normalized electrical and chemical state variables, $i_j = I_j/I_0$ and $x_j = X_j/X_0$, to be identical. Here j is the index of the state variable. This condition applies to initial conditions as well, and so we have

$$I_{j,init} = I_0 \left(\frac{X_{j,init}}{X_0} \right) \tag{6.73}$$

where the subscript *init* denotes initial conditions. We pick values for $I_{j,init}$ from a set generated by a binary current divider. The relevant circuit is described later in more detail. It produces a set of currents described by

$$I_{init}(N) = I_{min} 2^N \tag{6.74}$$

where, on this chip N is an integer between 0 and 15 and $I_{min} = 30\text{pA}$. The maximum current available is $I_{max} = I_{min}2^{15} \approx 1\mu\text{A}$. We use the value of N that provides the closest match between the required initial current $I_{j,init}$ and the available set $I_{init}(N)$. The result is

$$N = \left[N_0 + \log_2 \left(\frac{X_{j,init}}{X_0} \right) \right] \tag{6.75}$$

where $N_0 = \log_2 (I_0/I_{min})$, so that $I_0 = I_{min}2^{N_0}$, and [ ] is the nearest integer function.

6.3.4 Noise Analysis

Noise Statistics

We know that counting the arrivals of uncorrelated events leads to a Poisson probability distribution in the value of the counter N, i.e., we get

$$p(N) = \frac{\overline{N}^N \exp\left(-\overline{N}\right)}{N!} \tag{6.76}$$

where $\overline{N}$ is the mean (expected) number of events. The variance of a Poisson distribution is also equal to $\overline{N}$. White shot noise is caused by the arrival of uncorrelated events at a detector, and counting is the discrete analog of integration. Therefore, integrating one or more uncorrelated white shot noise currents I_i (with PSD's equal to $2q\overline{I_i}$) onto a capacitor causes its charge to have a Poisson distribution, with mean $\overline{Q}$ and variance $2q\overline{Q}$. Similarly, a Poisson distribution also results when chemical reactions produce molecular fluxes with white PSDs that are then integrated into concentrations. However, there are many cases in electronics and chemistry when the distribution of capacitor charge or species concentration is *not* Poisson. The main reasons are

- There is feedback in the system, so that the mean value of the charging current depends on the voltage that is being charged. This is true of a resistor, for example. Feedback changes the bandwidth over which the noise PSD is integrated

from $1/T$, where T is the observation time, to a value approximately equal to the bandwidth of the feedback loop.

- The current that charges the voltage is not the same as the current that sets its variance. For example, in a resistor some of the current is carried by drift, which is noiseless as long as the drift velocity v_d is much smaller than the mean thermal velocity $\sqrt{kT/m}$. The total current through a resistor R is V_R/R, where V_R is the voltage drop across it. However, the noise PSD is $2qI_D$, where the diffusion current $I_D = 2\phi_T/R$. Similarly, in an operational transconductance amplifier with bias current I_B, the output noise current PSD is $2qNI_B$, where N is the effective number of noisy devices inside the amplifier. In particular, the noise currents of all transistors connected to the input node directly add up and contribute to the total output noise current. The current that charges the capacitor, on the other hand, is equal to the *difference* between the currents carried by PMOS and NMOS transistors connected to the output node.

- There are correlations between charging currents.

- There are other, external sources of noise. For example, the mean values of the charging currents could be functions of other potentials in the system, and could therefore themselves be noisy. In genetic circuits, for example, the noise in protein concentrations can be divided into intrinsic and extrinsic parts [215, 234]. Intrinsic noise is uncorrelated across genes, unlike extrinsic noise, which is caused by environmental fluctuations and global gene regulatory networks.

The signal-to-noise ratio (SNR) if a variable X is defined as the average power of X divided by its total variance, i.e.,

$$\mathrm{SNR} = \frac{\left(\overline{X}\right)^2}{\overline{\left(X - \overline{X}\right)^2}} = \frac{\left(\overline{X}\right)^2}{\sigma_X^2} \tag{6.77}$$

In the biological literature $\mathrm{CV} = 1/\sqrt{\mathrm{SNR}}$ is often used instead of SNR. Here CV stands for 'coefficient of variation', and is equal to the standard deviation divided

by the mean. Another measure of variance that is commonly used is *noise strength*, defined as the variance divided by the mean. Noise strength is a type of population-normalized variance: for Poission processes variance is equal to the mean, so noise strength is equal to 1 for all values of the mean. The functional dependence of SNR on mean level can have a variety of forms because of the mechanisms described previously. Some commonly-observed types of dependence are:

1. Noise is constant with signal level. For example, the variance of capacitor voltages in resistor-capacitor circuits scales as $\overline{v_n^2} \propto kT/C$, independent of signal level. In this case σ_X^2 is independent of $\overline{X}$, so SNR $\propto \left(\overline{X}\right)^2$.

2. Noise power scales with signal level. This case corresponds to Poission statistics. Since $\sigma_X^2 \propto \overline{X}$, SNR $\propto \overline{X}$.

3. Noise power scales with signal power. As we shall show, this case occurs in current-mode circuits at large signal levels. Since $\sigma_X^2 \propto \left(\overline{\overline{X}}\right)^2$, SNR is constant.

Elementary Chemical Reactions

Let us consider the elementary chemical reaction $A + B \leftrightarrow C$ in steady state. The forward and backward rate constants are denoted by k_1 and k_2, respectively. We want to analyze the concentration fluctuations, i.e., noise, produced by this reaction. In general all the species concentrations have noise. However, we will consider the simple case when two of the species are assumed to be effectively noiseless. For example, they could be present at much higher concentrations than the third species. Since A and B can be reversed without affecting the reaction, their noise properties will be identical. Thus, there are only two unique cases: either A or C is noisy.

We assume that every reaction flux consists of uncorrelated arrivals or departures of molecules, and thus displays Poisson statistics. A signal-flow block diagram of the chemical reaction is shown in Figure 6-6. Consider part (a) of the figure, which models fluctuations in A. The figure shows that the difference between the fluxes that create and destroy A is integrated into the concentration of A. Each of these fluxes has a power spectral density equal (PSD) to twice the mean flux, just like any

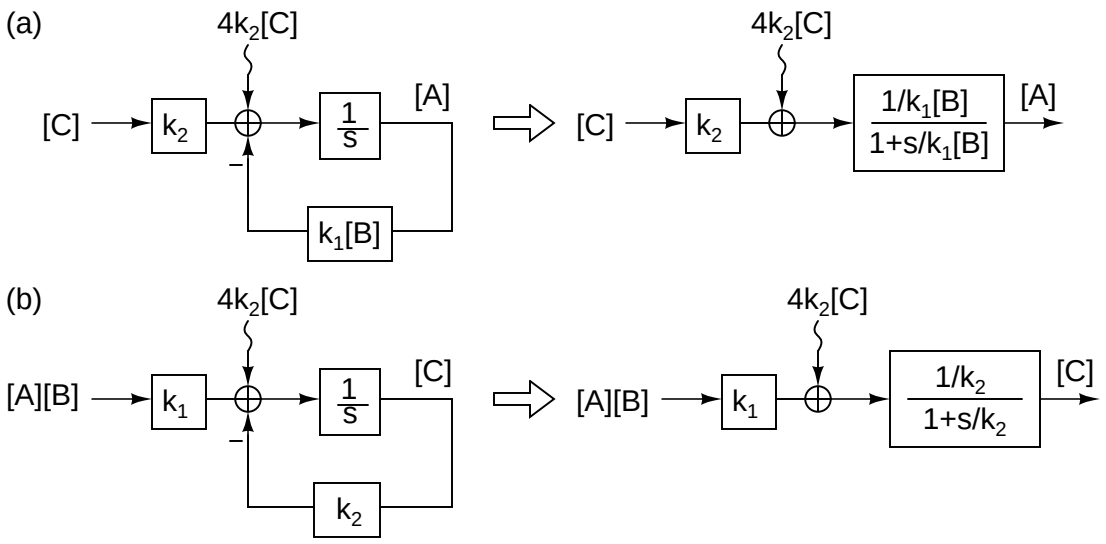


Figure 6-6: Signal-flow block diagrams for calculating the noise of (a) the species A and (b) the species C due to the elementary chemical reaction $A + B \leftrightarrow C$. Wavy lines represent noise power spectral densities in steady state.

Poisson variable. The two mean fluxes are $k_2[C]$ and $k_1[A][B]$, leading to PSDs of $2k_2[C]$ and $2k_1[A][B]$, respectively. These PSDs are independent, and therefore the total PSD that is filtered by the low-pass filter is the sum of the two. However, at steady state the two fluxes must have identical mean values, and so the total PSD is $2 \times 2k_2[C] = 4k_2[C]$. Similarly, if we consider part (b) of the figure we see that the total PSD that is filtered by the low-pass filter in this case is also $2k_2[C]+2k_1[A][B] = 4k_2[C]$ in steady state.

We note that the noise bandwidth of a first-order low-pass filter is $\pi/2$ times larger than its 3dB bandwidth. Thus the noise bandwidth of the filter in Figure 6-6(a) is $(\pi/2) \times k_1[B]/(2\pi) = k_1[B]/4$. Similarly, the noise bandwidth of the filter is filter in Figure 6-6(b) is $k_2/4$. We can now immediately write down, *by inspection*, that

$$\begin{aligned} \sigma_A^2 &= 4k_2[C] \times \frac{1}{(k_1[B])^2} \times \frac{k_1[B]}{4} \\ &= \frac{k_2[C]}{k_1[B]} \\ &= \overline{[A]} \end{aligned} \tag{6.78}$$

for A, and

$$\begin{aligned} \sigma_C^2 &= 4k_2[C] \times \frac{1}{k_2^2} \times \frac{k_2}{4} \\ &= \overline{[C]} \end{aligned} \tag{6.79}$$

for C, where $\overline{[A]}$ and $\overline{[C]}$ are the mean concentrations of A and C, respectively. We see that the variances of A and C are both equal to their mean values, i.e., they both behave as Poisson random variables. Electronically, the same reaction $A + B \leftrightarrow C$ is modeled by the following equations:

$$\begin{aligned} C\frac{dV_A}{dt} &= -\beta_1 I_B + \beta_2 \frac{I_C I_0}{I_A} \\ C\frac{dV_B}{dt} &= -\beta_1 I_A + \beta_2 \frac{I_C I_0}{I_B} \\ C\frac{dV_C}{dt} &= \beta_1 \frac{I_A I_B}{I_C} - \beta_2 I_0 \end{aligned} \tag{6.80}$$

Here β_1 and β_2 are defined as

$$\begin{aligned} \beta_1 &= \alpha\tau_0 X_0 k_1 \\ \beta_2 &= \alpha\tau_0 k_2 \end{aligned} \tag{6.81}$$

where α is the speedup factor, $\tau_0 = C\phi_T/(\kappa I_0)$, and X_0 and I_0 are the reference concentration and current, respectively. The electronic analog of the dissociation constant of the reaction is given by

$$K_d = \left(\frac{\beta_2}{\beta_1}\right) I_0 = K_{d,chem} \left(\frac{I_0}{X_0}\right) \tag{6.82}$$

where $K_{d,chem} = k_2/k_1$ is the dissociation constant of the original chemical reaction. A simplified version of the circuit that determines the concentration of C, i.e., imple-

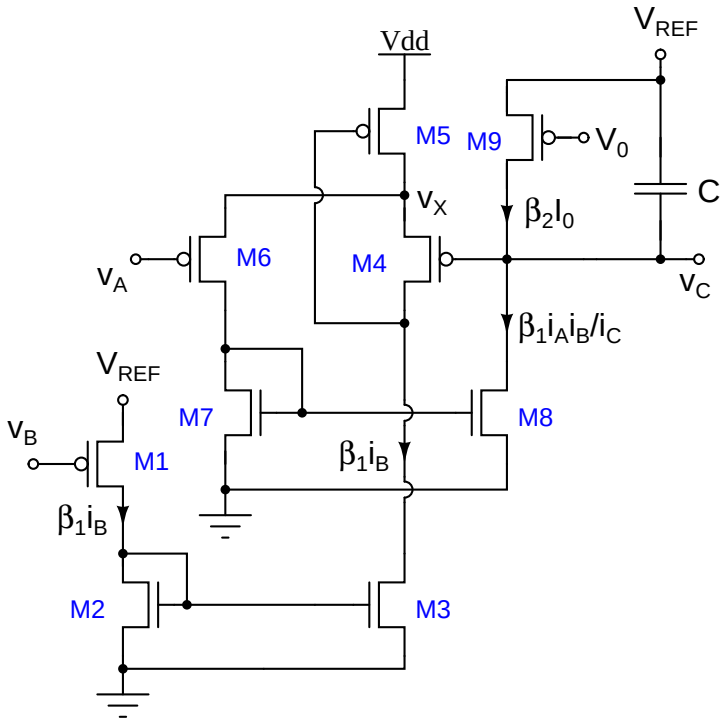


Figure 6-7: A simplified circuit that controls the concentration of C in the reaction $A + B \leftrightarrow C$.

ments the third equation in (6.80), is shown in Figure 6-7.

In general the concentrations of A, B and C can depend on other species. Thus, fluctuations in the concentrations of other species can cause fluctuations in $[A]$, $[B]$ and $[C]$. We shall ignore such extraneous sources of noise, because we want to calculate the fluctuations in concentration produced by fluxes within a single reaction. Thus, as in our earlier analysis, we will only allow one species in the reaction to fluctuate at any time. The other two will be held at fixed values. In addition, the circuit is assumed to be in steady-state, so that the average current through the capacitor C is zero. In this analysis we shall ignore non-fundamental sources of noise, such as flicker noise. Each transistor's noise model then consists of a white current source with PSD $2qI_i$ connected between the source and drain, where I_i is the average (DC) current through the transistor M_i. These sources are assumed to be uncorrelated with each other. The total short-circuit output noise current at v_C is found by adding up contributions from each transistor. The results are shown in Table 6.3.

It is possible in principle to run the NMOS mirror transistors M2, M3, M7 and

Transistor	**PSD at v_{OUT}**
M1	$2q\beta_1 I_B \left(g_{m6}/g_{m4}\right)^2 = 2q\left(\beta_2 I_0\right)^2 / \left(\beta_1 I_B\right)$
M2	$2q\beta_1 I_B \left(g_{m6}/g_{m4}\right)^2 = 2q\left(\beta_2 I_0\right)^2 / \left(\beta_1 I_B\right)$
M3	$2q\beta_1 I_B \left(g_{m6}/g_{m4}\right)^2 = 2q\left(\beta_2 I_0\right)^2 / \left(\beta_1 I_B\right)$
M4	$2q\beta_1 I_B \left(g_{m6}/g_{m4}\right)^2 = 2q\left(\beta_2 I_0\right)^2 / \left(\beta_1 I_B\right)$
M5	0
M6	$2q\beta_2 I_0$
M7	$2q\beta_2 I_0$
M8	$2q\beta_2 I_0$
M9	$2q\beta_2 I_0$
Total	$2q\beta_2 I_0 \left[4 + 4\left(\beta_2 I_0\right) / \left(\beta_1 I_B\right)\right]$

Table 6.3: Current-mode LPF output noise sources

M8 far enough above threshold for most of their current to be carried by drift. As a result their PSD's will be neglible compared to those of M1, M4, M6 and M9 (which must remain below threshold since they are part of the translinear loop). In this case the total short-circuit current noise PSD at v_C is reduced by a factor of two from the value shown in Table 6.3, and is given by

$$\frac{\overline{i_C^2}}{\Delta f} = 2q\beta_2 I_0 \left[2 + 2\left(\frac{\beta_2 I_0}{\beta_1 I_B}\right)\right] \tag{6.83}$$

It is easy to see that the small-signal impedance at v_C is equal to $1/g_{m6}$. Since the steady-state current through M6 is equal to $\beta_2 I_0$, the impedance is given by

$$r_C = \frac{1}{g_{m6}} = \frac{\left(\phi_T/\kappa\right)}{\beta_2 I_0} \tag{6.84}$$

The noise bandwidth of v_C is that of a first-order low-pass filter with time constant $\tau = r_C C$. It is given by

$$BW = \frac{1}{2\pi} \times \frac{1}{\tau} \times \frac{\pi}{2} = \frac{\beta_2 I_0}{4C\left(\phi_T/\kappa\right)} \tag{6.85}$$

where we have substituted for r_C using (6.84). Therefore the total variance of v_C is given by

$$\overline{v_n^2} = \frac{\overline{i_C^2}}{\Delta f} \times BW \times r_C^2 = \frac{kT}{2\kappa C}\left[1 + \left(\frac{\beta_2 I_0}{\beta_1 I_B}\right)\right] \tag{6.86}$$

We see that $\overline{v_n^2}$ depends on I_B, but not on the other input A. In a real implementation we use two translinear circuits to make the circuit symmetric with respect to A and B, as described earlier. In this case it is easy to show that the variance at C is modified to

$$\overline{v_n^2} = \frac{kT}{2\kappa C}\left[1 + \left(\frac{\beta_2 I_0}{\beta_1 I_B}\right) + \left(\frac{\beta_2 I_0}{\beta_1 I_A}\right)\right] \tag{6.87}$$

Also, in steady-state, we have $\beta_2 I_0 = \beta_1 I_A I_B / I_C$. Substituting this relationship into (6.88) we get

$$\overline{v_n^2} = \frac{kT}{2\kappa C}\left[1 + \left(\frac{I_A + I_B}{I_C}\right)\right] \tag{6.88}$$

We see that the variance in v_C is proportional to kT/C, like in simple resistor-capacitor circuits, but also depends on the ratio $(I_A + I_B)/I_C$. Assuming that $\overline{v_n^2} \ll \phi_T/\kappa$, the variance in i_C is given by

$$\begin{aligned}\overline{i_n^2} &= \overline{v_n^2}\left(\kappa I_C/\phi_T\right)^2 \\ &= \frac{qI_C^2}{2C\left(\phi_T/\kappa\right)}\left[1 + \left(\frac{I_A + I_B}{I_C}\right)\right]\end{aligned} \tag{6.89}$$

where we have assumed that the exponentiation of v_C to i_C generates no additional noise. In our case this is a valid assumption since exponentiation is done off-chip in software. Thus the SNR of C is given by

$$\mathrm{SNR}_C \equiv \frac{I_C^2}{\overline{i_n^2}} = \left(\frac{2C\phi_T}{\kappa q}\right)\frac{1}{\left[1 + \left(I_A + I_B\right)/I_C\right]} \tag{6.90}$$

We see that the SNR increases $\propto I_C$ when I_C, i.e., the mean output current, is much smaller than $(I_A + I_B)$. This behavior is similar to a Poisson process, and occurs because the noise voltage at C is "predistorted" by the factor $1/I_C$ before being exponentiated into a current. Thus, in this regime the noise properties of our circuit are similar to the chemical reaction. However, unlike in chemistry our SNR saturates to a value $\mathrm{SNR}_{\max}$ for large values of I_C. The maximum SNR is given by

$$\mathrm{SNR}_{\max} = \frac{2C\phi_T}{\kappa q} \tag{6.91}$$

In practice several noisy transistors will be needed to implement current mirrors within the circuit, reducing $\mathrm{SNR}_{\max}$ to

$$\mathrm{SNR}_{\max} = \frac{4C\phi_T}{\kappa q N_{eff}} \tag{6.92}$$

where N_{eff} is the number of noisy transistors, i.e., the excess noise factor. For example, it is easy to show that $\mathrm{SNR}_{\max}$ is lowered by a factor of 2, i.e., 3dB, if the NMOS transistors shown in Figure 6-7 are assumed to operate in sub-threshold. In other words we get $N_{eff} = 4$ in this case. We can carry out a similar analysis to find the SNR of A and B. The result is

$$\begin{aligned} \mathrm{SNR}_A &= \frac{\mathrm{SNR}_{\max}}{[1 + I_C/I_A]} \\ \mathrm{SNR}_B &= \frac{\mathrm{SNR}_{\max}}{[1 + I_C/I_B]} \end{aligned} \tag{6.93}$$

where SNR_A and SNR_B refer to the SNR of A and B, respectively, and $\mathrm{SNR}_{\max}$ is the same as before. We see that the SNR of A also goes up linearly with its mean level before saturating to $\mathrm{SNR}_{\max}$, and that B behaves similarly. Thus, the noise properties of our circuit are similar to the original chemical reaction for values of $\mathrm{SNR} < \mathrm{SNR}_{\max}$. The effects of noise are important at low values of SNR. Thus, such effects will be accurately modeled by our circuit if $\mathrm{SNR}_{\max} \gg 1$. For $C = 1\mathrm{pF}$ at room temperature, and assuming $\kappa = 0.7$, (6.91) predicts $\mathrm{SNR}_{\max} = 56.7\mathrm{dB}$.

In order to verify our theoretical predictions we used SPICE to simulate the reaction $A + B \leftrightarrow C$. The system was implemented using the reaction circuits that were described earlier. The same circuits were later implemented on-chip. Figure 6-8 shows the simulated SNR of I_C as a function of the capacitance C for various mean levels of I_C. The values of K_d, I_A and I_B were fixed at 28nA, and flicker noise was absent. We see that SNR increases linearly with C but saturates with increasing I_C,

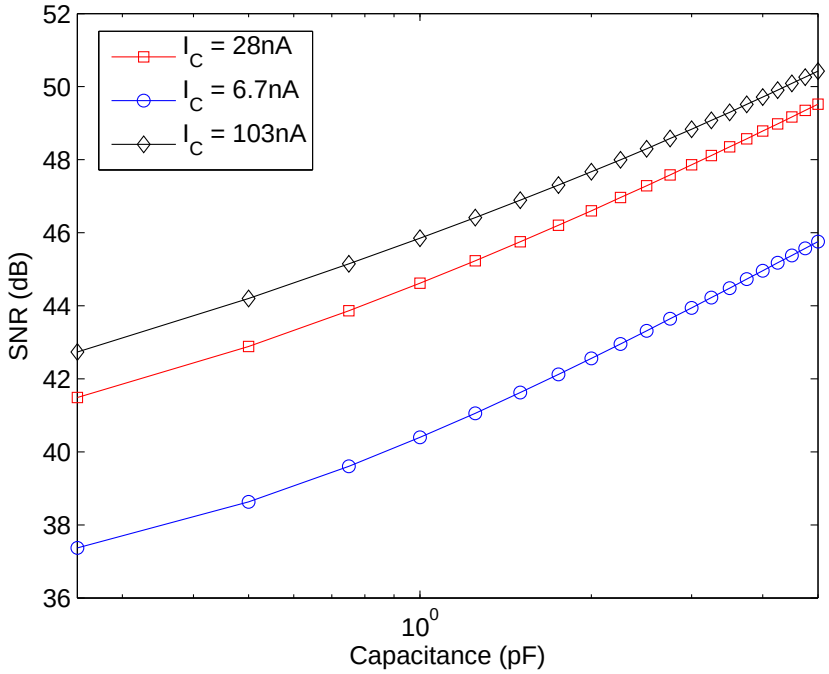


Figure 6-8: Simulated SNR of I_C as a function of the capacitance C. Plots are shown for various mean levels of I_C in the absence of flicker noise and with $K_d = 28$nA.

as expected. However the slopes as a function of C are somewhat lower than predicted, particularly for small values of C. It is possible that parasitic capacitances, which produce larger effects as C decreases, are responsible.

Figures 6-9 and 6-10 show the simulated SNR of I_A and I_C, respectively, as a function of their mean values. Plots are shown for various values of K_d in the presence and absence of flicker noise. We see that SNR is almost independent of K_d when flicker noise is absent, as expected. In addition, we obtain a good fit to theory by assuming $N_{eff} = 16$. However, when flicker noise is taken into account higher values of K_d result in somewhat higher SNR. Also, the SNR curves with flicker noise saturate more quickly, and at lower values of SNR_{max}, than those without. The PSD of flicker noise $\propto \overline{I}^2$, as predicted by (5.128), while that of thermal noise is $\propto \overline{I}$. Here $\overline{I}$ is the mean current level. The faster increase in flicker noise PSD with $\overline{I}$ explains why its relative contribution to the SNR increases with $\overline{I_A}$ and $\overline{I_C}$.

Both DC mismatch and flicker noise are primarily caused by traps within the gate oxide, and can be reduced by using larger transistors. However, larger transistors

also have larger parasitic capacitances, which decrease the stability of our translinear loops for a given value of C. Thus the oxide trap density, which is a process-dependent parameter, imposes lower bounds on C. The speedup factor α is inversely proportional to C when β is fixed, implying that the amount of simulation speedup is also ultimately limited by trap density.

Equation (6.92) can be rewritten as $\mathrm{SNR}_{\max} = 4CV_L/(qN_{eff})$, where $V_L = \phi_T/\kappa$ is the linear range of a subthreshold transistor. In reality V_L is not constant, but increases gradually with mean current level, particularly as the transistors approach the moderate inversion region. As a result $\mathrm{SNR}_{\max}$ increases slowly with $\overline{I_A}$ and $\overline{I_C}$, explaining why the SNR curves shown in Figures 6-9 and 6-10 do not saturate as rapidly as predicted theoretically, at least in the absence of flicker noise.

Finally, we see that very small capacitors and current levels are required to get low values of SNR. Small currents are undesirable because they are difficult to generate reliably, and eventually become comparable to transistor leakage. Parasitic leakage currents within transistors are mainly caused by quantum-mechanical tunneling of electrons across the gate oxide and source/drain junctions. They are poorly modeled, do not match particularly well, and increase in magnitude as process scaling shrinks device sizes [213]. On the other hand, at low values of C the dynamics of our circuits are distorted by parasitic poles within the translinear loops. Eventually the loops go unstable, limiting the minimum allowable value of C. In a standard 0.18μm CMOS technology it appears that the minimum SNR that can be reliably achieved is limited to approximately 25dB.

6.3.5 System Design

In this section we describe the circuits used to build the chemical simulator chip in greater detail. A high-level block diagram of the chip is shown in Figure 6-11. The current chip occupies 1.5mm x 1.5mm in a 0.18μm CMOS process. It contains 81 second-order reaction circuits, 40 first-order reaction circuits, 40 zeroth order reaction circuits, $N = 32$ state variables, $N_I = 8$ inputs and $P = 16$ outputs.

The main connection matrix represents an array of programmable switches that

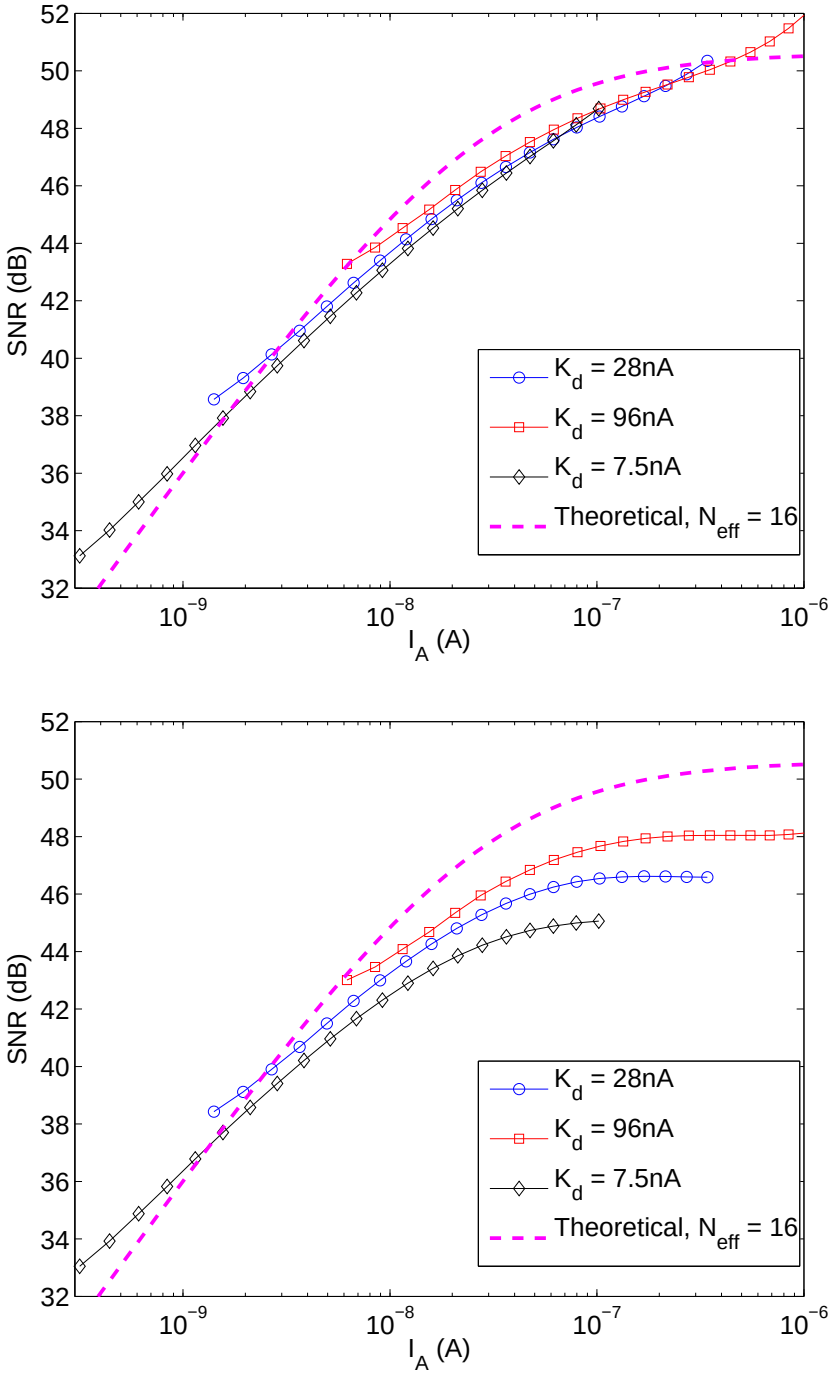


Figure 6-9: Simulated SNR of I_A as a function of its mean value. Plots are shown for various values of K_d in the absence (top) and presence (bottom) of flicker noise, and compared to the theoretical prediction for $N_{eff} = 16$.

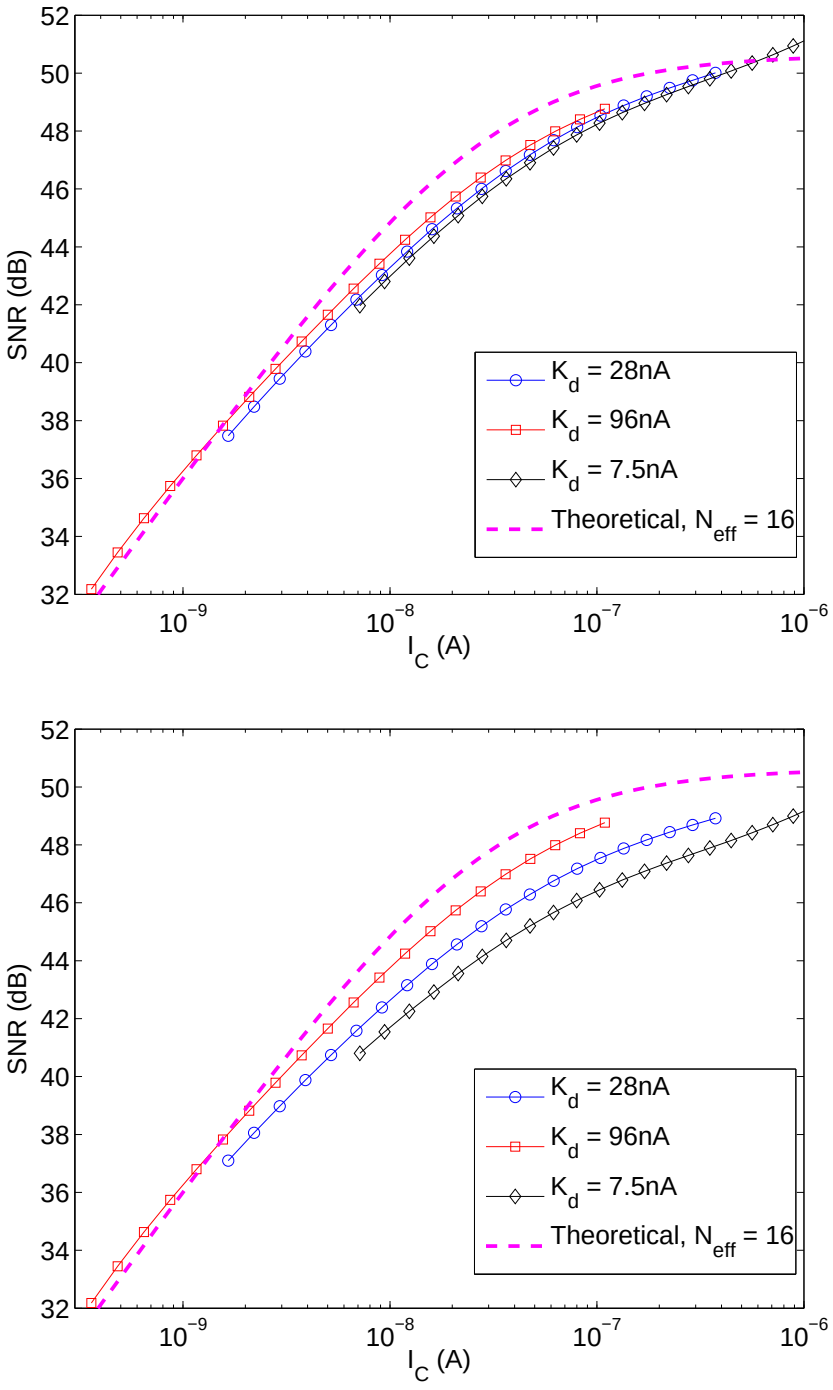


Figure 6-10: Simulated SNR of I_C as a function of its mean value. Plots are shown for various values of K_d in the absence (top) and presence (bottom) of flicker noise, and compared to the theoretical prediction for $N_{eff} = 16$.

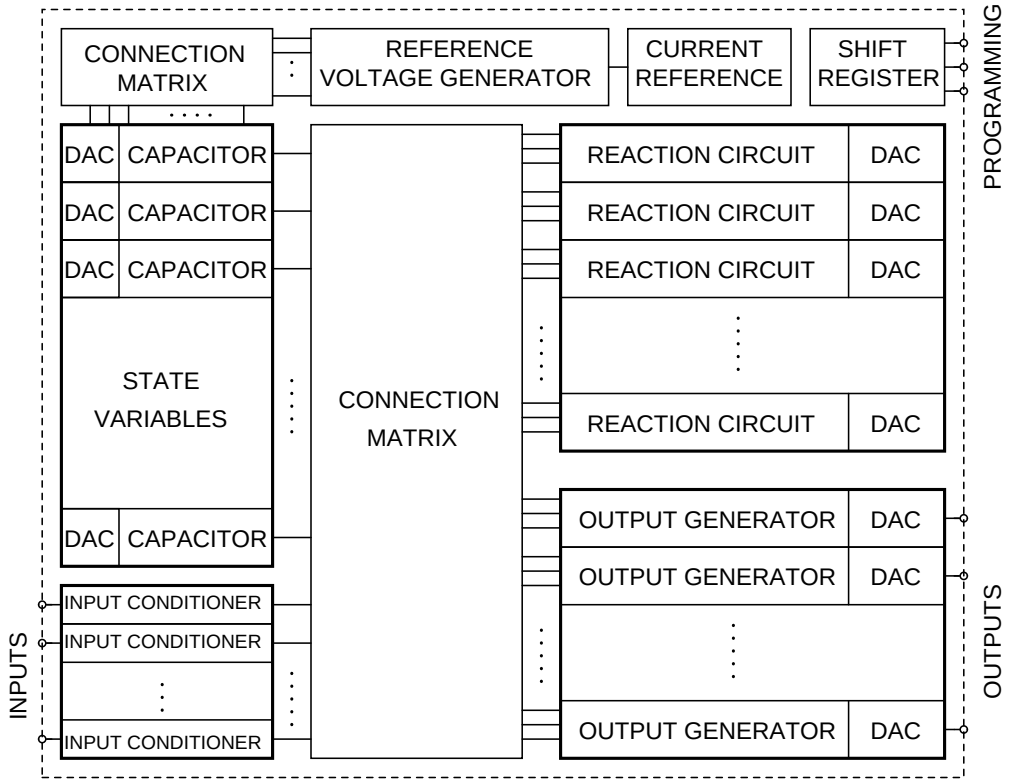


Figure 6-11: A high-level block diagram of the entire chemical network simulator chip.

makes the topology of the reaction network completely arbitrary. Thus, it allows the user to connect any terminal in any of the equation circuits and output generators to any of the state variables or external inputs. The parameters of a given network topology, i.e., reaction rates and initial conditions, are also individually programmable by transistor DACs within the reaction circuits.

State variables are stored on capacitors in log-compressed form, i.e., as chemical potentials. The value of these capacitors can be individually set by capacitor DACs, allowing us to simulate systems where reactants and products are present in compartments with different volumes. The reference voltage generator determines the allowable set of initial conditions that can be set on these capacitors. A second connection matrix allows the user to select an arbitrary member of this set for each state variable.

Reaction Circuits

We used the reaction circuits described in Section 6.3.2. All state variables were stored on capacitors and distributed to the various reaction circuits as log-compressed voltages, not currents. There are two advantages to distributing voltages: less long-distance routing, as disussed in more detail below, and lower noise because of the elimination of one or more current mirrors.

We use a flipped voltage follower circuit [36], shown in Figure 6-12, to regenerate state variable currents. In this circuit M2 and the current source I_{SET} act as a common-source amplifier wrapped around the transistor M1. The small-signal impedance at the node V_{REF} is reduced by the the gain of this amplifier, i.e., goes down from $1/g_{m1}$ to approximately $2/\left(g_{m1}g_{m2}r_{o1}\right)$. In addition, the circuit can supply V_{REF} with large amounts of current, but can only sink an amount I_{SET}. Thus, the flipped voltage follower provides a simple way of creating a variable, unidirectional power supply. One of these circuits is located within each reaction circuit. The user sets the values of the global variables I_{SET} and V_{SET}, which fixes the reference voltage V_{REF}. The user can also set the values of the global variables I_{min} and I_0 required by the reaction circuits.

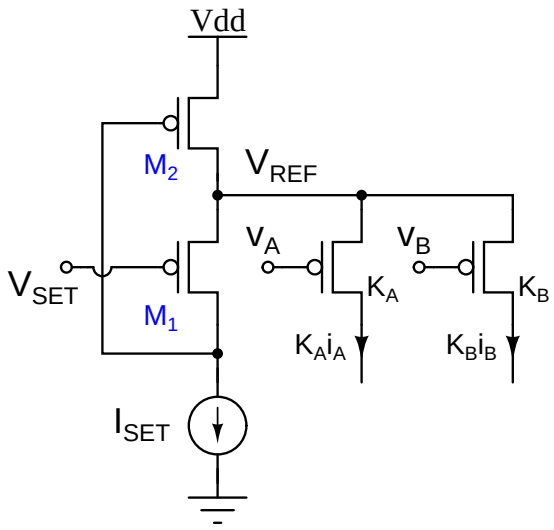


Figure 6-12: The scheme used on the chip for locally regenerating state variable currents from log-compressed voltages.

Each state variable is regenerated by a PMOS transistor with its source tied to V_{REF} and its gate tied to the log-compressed, globally distributed voltage in question. The width of the transistor can be set by a DAC, allowing its current to be scaled in a programmable way. For example, Figure 6-12 shows that the two state variables i_A and i_B have been regenerated with scaling factors K_1 and K_2, respectively.

Transistor mismatches will cause static errors in the local value of a given state variable. They will also cause errors in the rate constants and initial conditions assigned to a reaction. Thus, mismatch and device noise both act as sources of random variability in the behavior of our reaction circuits. The dominant source of transistor mismatch is the threshold voltage V_T. A given threshold voltage offset ΔV_T is exponentiated into a fractional error $\Delta V_T/V_L$ in the value of the state variable, where $V_L = \phi_T/\kappa \approx 37\text{mV}$ is the linear range of the transistor. The most widely-accepted model of mismatch between integrated MOS transistors was proposed by Pelgrom et al. in 1989 based on experimental measurements [223]. The model states that the variance in V_T between two similar MOS transistors on the same die is approximately given by

$$\sigma_{VT}^2 = \frac{A_{VT}^2}{WL} + S_{VT}^2 D^2 \tag{6.94}$$

where WL is the gate area of each transistor, D is the distance between them and A_{VT} and S_{VT} are process-dependent constants. Both A_{VT} and S_{VT} tend to decrease as processes scale and fabrication methods improve. For example, values of A_{VT} decreased from 30mV-μm in an old 2.5μm process [223] to ≈5mV-μm in our current 0.18μm process. Our default transistor size, corresponding to K_A or $K_B = 1$ in Figure 6-12, was $W = 10\mu$m and $L = 1\mu$m, resulting in $A_{VT}/\sqrt{WL} \approx 1.6$mV.

Typical values of S_{VT} range from 1-10μV/μm, showing that the first term is usually dominant. Thus, the fractional error between two nominally-identical state variables is expected to have a standard deviation of 1.6mV/37mV≈ 4.3%. Nevertheless, the second term can contribute significant error when voltages are distributed across chips and locally exponentiated into currents, as in our case, because transistors that should match can be far apart (large D).

The generic problem of distributing a state variable to multiple locations on-chip is known as *fanout*. Our current fanout strategy is based on distributing log-compressed voltages. An alternate fanout strategy is to generate many copies of each current at one location and then distribute them across the chip. This process keeps D low, thus improving accuracy. The main disadvantage of this strategy is increased wiring: Currents, unlike voltages, cannot be reused, and so we have to distribute many currents per state variable, instead of a single voltage. Hybrid fanout strategies located somewhere between these two extremes often provide the best trade-off between accuracy and layout area.

The first term in (6.94) is caused by random fluctuations on length scales shorter than $\sqrt{WL}$, such as the presence of oxide charge traps. The second term is caused by fluctuations, such as gradients in doping density, that occur on longer length scales. Deviation in other transistor properties follow a similar equation, but with different constants replacing A_{VT} and S_{VT}. Thus, we must use large transistors located close to each other in order to get low mismatch. In principle mismatches can also be measured during an initial calibration step and then removed during actual simulations. This process can be carried out in hardware, in which case it is known as auto-zeroing, or in software.

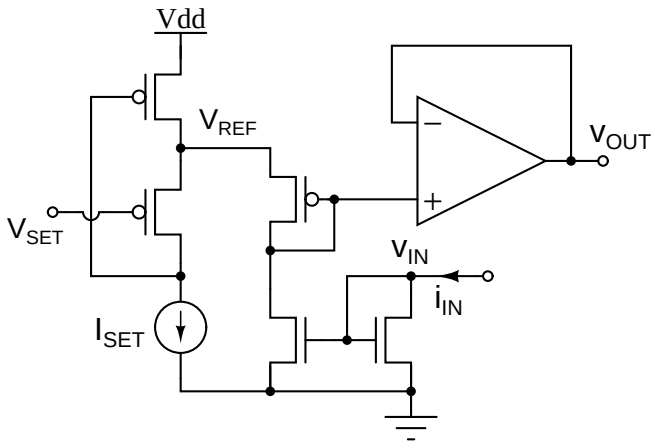


Figure 6-13: A circuit that accepts an external current or voltage input and converts it into a form suitable for on-chip use.

Input/Output Circuits

Figure 6-13 shows how external inputs are fed into the chip. The chip accepts both currents i_{IN} and log-compressed, ground-referenced voltages v_{IN} as inputs. A current mirror and a flipped voltage follower circuit are used to generate a V_{REF}-referenced voltage v_{OUT} from v_{IN}. In other words, the circuit sets v_{OUT} such that the current through a standard-size PMOS transistor with its source at V_{REF} and gate at v_{OUT} is equal to i_{IN}. We also see that the voltage v_{OUT} is buffered by an operational amplifier before being fed to the rest of the chip. The buffer guarantees stability and rapid transient response when a single input is connected to many reaction circuits and therefore has a large capacitive load on it. The current chip has $N_I = 8$ such inputs.

Figure 6-14 shows the circuit that generates outputs from the chip. The circuit implements one row of the output matrices **H** and **K** of the original chemical reaction system shown in (6.6). The electrical equivalent of (6.6) is given by (6.12), and shows that a single output is the weighted sum of all on-chip state variables and external inputs. We expect most of these weights to be zero, and therefore only allow S terms to be present in the summation, where S is much smaller than the total number of state variables and inputs, i.e., $N + N_I$. The current chip used $S = 2$, and provided $P = 16$ independent outputs.

Each input voltage $v_{IN,s}$ to the circuit shown in Figure 6-14 is selected from the $N+$

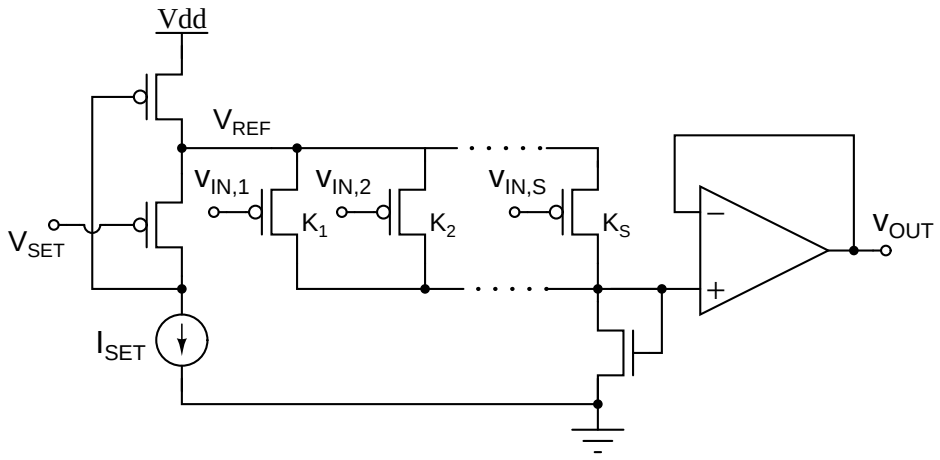


Figure 6-14: A circuit that generates the weighted sum of multiple on-chip currents and converts it into a buffered, log-compressed voltage that can be sent off-chip.

N_I possible state variables and inputs by a connection matrix, i.e., analog multiplexer (not shown). The connection matrix is similar to the one used for connecting reaction circuits to state variables and inputs. The S transistors that convert these V_{REF}-referenced voltages into currents actually represent 5-bit transistor DACs. Each DAC value K_s can be individually set by the user. Thus the s-th transistor carries a current $K_s i_{IN,s}$. Using KCL, the output current is given by the weighted sum

$$i_{OUT} = \sum_{s=1}^{S} K_s i_{IN,s} \tag{6.95}$$

The current i_{OUT} is log-compressed into a ground-referenced voltage v_{OUT} by an NMOS transistor before being buffered and taken off-chip. The buffer ensures that large capacitive loads, such as oscilloscope channels, can be driven with adequate bandwidth.

Initial Conditions

We used a Bult-Geelen current-splitter circuit [56] to generate binary-weighted reference currents. The divider accepts an input current I_0, which is generated on-chip by a constant-g_m reference circuit similar to the one described in [184]. It divides I_0 into M output currents, where the m-th current is given by $I_{IC}(n) = I_0 2^{-m}$, where m is

an integer between 1 and M. On this chip, we used $M = 16$ and $I_0 = 2\mu$A, giving us reference currents between $I_0/2 = 1\mu$A and $I_0/2^{16} = 30$pA.

The current divider circuit is shown in Figure 6-15. It contains M stages, each consisting of a lateral transistor and two series-connected transverse transistors. At every stage, equal currents flow through the lateral and transverse transistors, resulting in binary current splitting. The current splitting ratio at every stage is only determined by the geometries of the lateral and transverse transistors, which share the same source and gate voltages. Thus the circuit operates accurately both above and below threshold. The circuit also contains a termination stage that has a single transverse transistor. The termination prevents an impedance discontinuity at the last stage, and makes the splitter appear infinitely long[8]. Finally, each current is log-compressed and referenced to the reference voltage V_{REF} to generate a set of voltages V_{IC} that can be used to set initial conditions on the chip.

An operational amplifier, shown in Figure 6-16, was used in its unity-gain configuration to buffer both the voltages used to set initial conditions, and also the off-chip inputs. This circuit may need to drive large capacitive loads. For example, several state variables may share the same initial condition, causing their capacitors to appear in parallel at the corresponding line within the initial condition bus. We used a class-AB push-pull output stage, which has low output impedance and high slew rate in both directions, to guarantee low overshoot and rapid settling under these conditions. The quiescent bias current of the output stage is set by the current through the diode-connected MOS transistor. This current is ratioed to be K times smaller than I_{BIAS}, the tail current of the input stage.

At the balance point the input transistor connected to v_+ carries a current $I_{BIAS}(1+1/K)/2$, while the one connected to v_- carries a current $I_{BIAS}(1-1/K)/2$. In order to prevent systematic offset these transistors should have the same V_{GS} drop. We enforce this constraint by making the former transistor R times wider than the latter, where the ratio R is given by

[8]The current-splitter circuit can be viewed as a lumped transmission line. The matched termination eliminates reflections, which would disturb the steady-state current profile on the line.

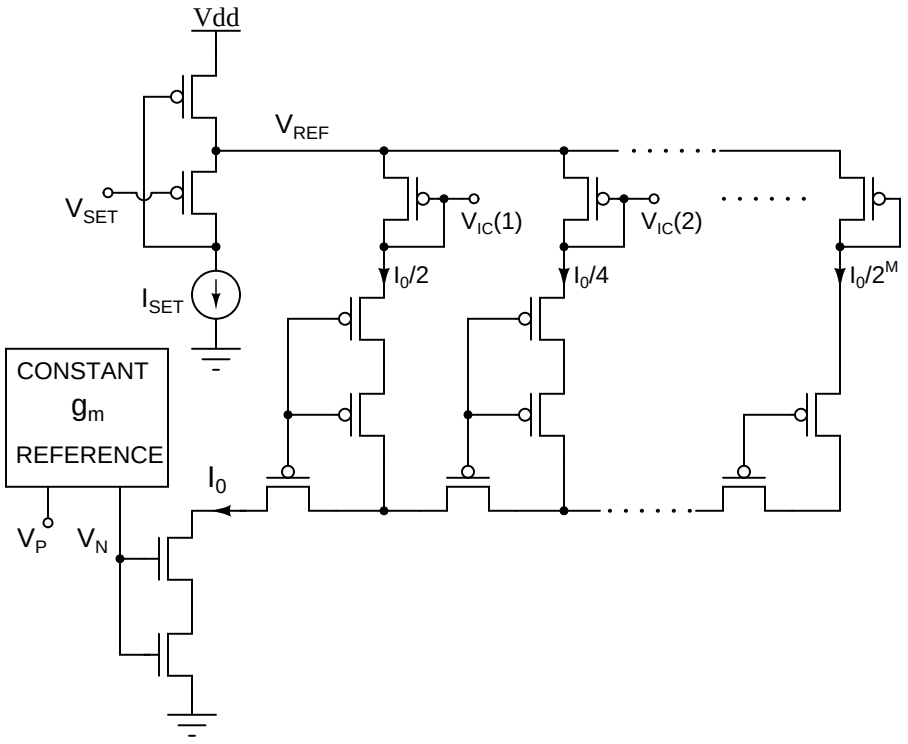


Figure 6-15: An R-$2R$ current divider circuit. The circuit generates DC voltages V_{IC} used for setting initial conditions.

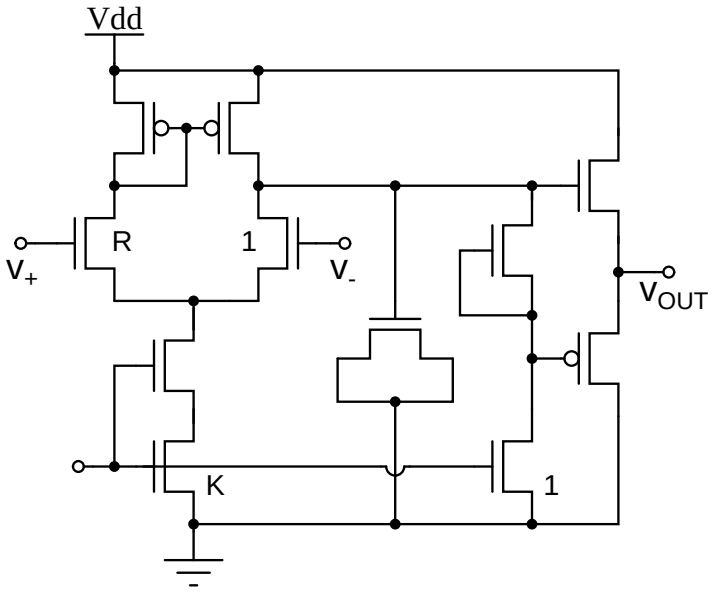


Figure 6-16: A two-stage operational amplifier with a class-AB output stage.

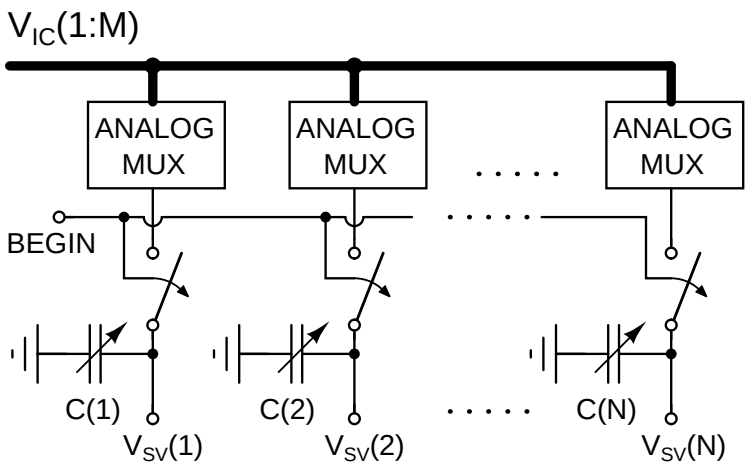


Figure 6-17: The scheme used on-chip for setting initial conditions. BEGIN is a global control signal that signals the beginning of a simulation.

$$R = \left(\frac{K+1}{K-1} \right) \tag{6.96}$$

We used $K = 4$ in our design, which results in $R = 5/3$. In order to save layout area we used an MOS capacitor for internal compensation. We also self-cascoded the tail current source to increase its output impedance. The lower transistor in the cascode was a 3.3V I/O transistor, which has higher threshold voltage than the upper transistor (a 1.8V device). The difference in threshold voltages provides enough V_{DS} to keep the lower transistor saturated.

Figure 6-17 shows how initial conditions are set on the chip. The chip contains N capacitors, each of which can store a log-compressed state variable. Each capacitor is actually implemented as the parallel combination of binary-weighted NMOS capacitors. Its value can be programmed by the user over a five-bit range (0.25pF to 8pF), thus allowing the simulation of multi-compartment models. The capacitors can be hooked up to any of the $M = 16$ available initial condition voltages V_{IC} through analog multiplexer circuits. On exponentiation, these voltages correspond to binary-weighted currents, as described earlier.

The user raises the global control signal BEGIN to start a simulation. Series switches disconnect the initial condition bus from the capacitors when BEGIN is high. The simulated model's own dynamics can now set the voltages V_{SV} on these

Table 6.4: Structure of the on-chip shift register

Bit Range	Function
0–159	Connection matrix/analog multiplexers
160–169	Transistor and capacitor DACs
170–179	Address

capacitors. The BEGIN signal is also used to trigger off-chip data acquisition equipment that records these dynamics for posterity. For details, please see the Appendix attached to this chapter. Finally, the reaction circuits can be hooked up to external inputs instead of on-chip state variables, as described earlier. The BEGIN signal does *not* disconnect these external inputs during a simulation.

Programming Interface

Chip parameters are programmed by the user through a three-wire bus. The signals on the bus are denoted by CLK. DATA and ENABLE. Bits on the DATA pin are loaded into an internal shift register at rising edges of the CLK signal. On the current chip, this global shift register is 180 bits long, and consists of three sections, as shown in Table 6.4. Each reaction circuit, output generator and state variable capacitor contains local memory and an unique 10-bit address. Local addresses are always being compared with the address bits currently stored within the global shift register. All shift register bits are loaded into local memory if the two addresses match and the ENABLE signal is high. Thus, in order to program the chip the user first sends all 180 bits, including the address, on the DATA pin, and then raises the ENABLE pin.

Our three-wire programming interface is simple and extremely robust. The chip does not have to recover a clock from the data stream, because it is generated externally and supplied on a separate pin. However, the scheme does have the disadvantage of not conforming to any industry standard, making it more difficult to interface our chip to microcontrollers and other peripherals. This issue can be addressed in later iterations of the design.

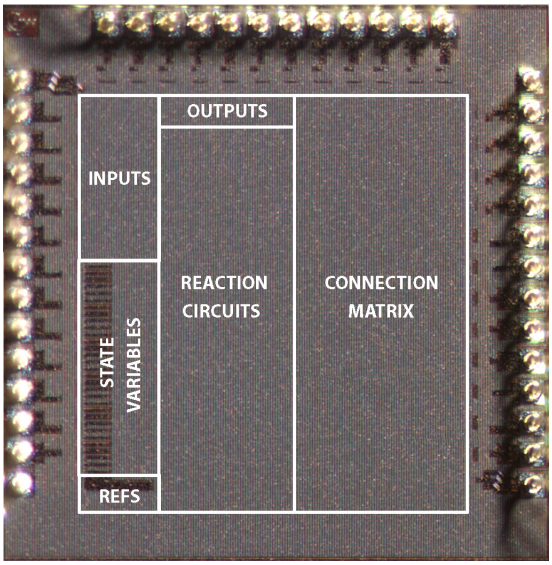


Figure 6-18: Die photograph of the programmable chemical simulator chip.

6.4 Measurements

A die photograph of our chemical simulator chip, which is 1.5×1.5mm in size, is shown in Figure 6-18. The chip was mounted on a printed circuit board and interfaced with a desktop computer running MATLAB. Please see the Appendix at the end of this chapter for more details of the chip/computer interface. The nominal power supply voltage was $V_{DD} = 1.6\text{V}$.

As described earlier, networks of switches allow one of a set of 16 voltages V_{IC} to be connected to each capacitor that stores a state variable. The set V_{IC} was designed to yield binary-weighted currents I_{IC} after exponentiation. Figure 6-19 shows the measured initial concentration I_{IC} of one state variable as a function of the digital code D_{IC} used to program its initial condition selection network. The voltage set by the network on the capacitor was connected to three different output buffers and exponentiated off-chip in software to obtain the three I_{IC} curves shown in Figure 6-19. We see that I_{IC} decreases exponentially as D_{IC} is lowered, which is what we expected. However, at very low values of D_{IC} the current stops decreasing and saturates to a value of $\approx$ 4nA. This behavior is probably caused by poorly-modelled junction leakage currents within transistors.

Multiple output buffers were connected to the same input V_{IC} to estimate the offset

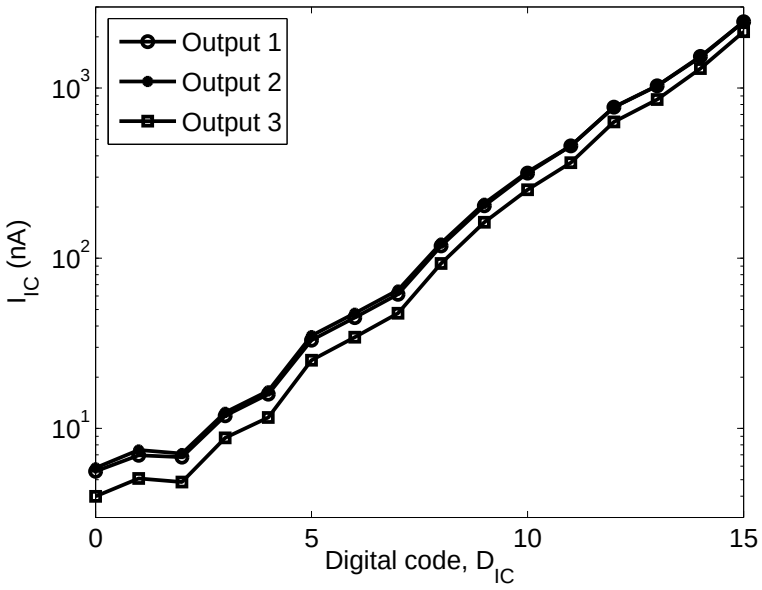


Figure 6-19: Measured current I_{IC} set at a node by its initial condition selection network. The log-compressed voltage corresponding to the current was measured by connecting the node to three different output buffers.

between them. A static voltage offset of ΔV_{off} betweem two buffers causes a constant fractional error of $\Delta I/I = V_{off}/\left(\phi_T/\kappa\right)$ in their output currents after exponentiation. The largest offset measured between any pair of output buffers on a single chip was approximately 15mV, resulting in $\Delta I/I = 40\%$. Outputs 1 and 3 in Figure 6-19 have an offset that is close to this worst case. Fortunately, the offset is constant with current level, as shown in the figure, and can be easily removed by creating a static lookup table in software. Alternatively, we can remove it in hardware by periodically auto-zeroing the output buffers. Auto-zeroing is a technique for cancelling the offset voltage of a closed-loop amplifier. The offset is first sampled on a capacitor C_{AZ} during an auto-zero phase. It is then removed during the operating phase by connecting C_{AZ} to the negative input terminal of the amplifier. We can easily auto-zero our buffers by using the periods between simulation runs as auto-zero phases.

Figure 6-20 shows measured concentrations of C as a function of time in the second-order reaction $A + B \rightarrow C$. We used equal initial concentrations of A and B, and a low initial concentration ($D_{IC} = 0$) of C. The plots in the figure correspond to

various values of the normalized rate constant β. We used the following set of values: $\beta = \{1, 4, 9, 16, 25, 49, 81\}/2^4$. We see that β has no effect on the final concentration of C, as expected.

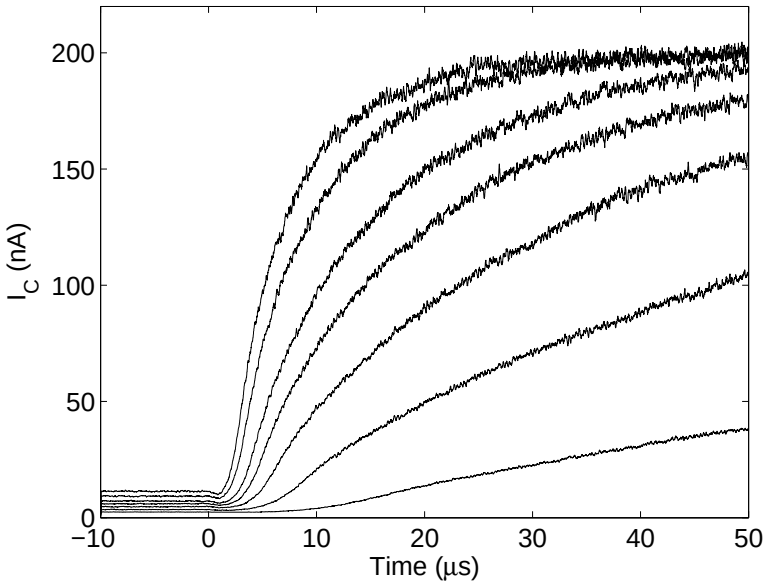


Figure 6-20: Measured concentration of C as a function of time in the second-order reaction $A + B \rightarrow C$ for various values of the normalized rate constant β.

Figure 6-21 shows the results of implementing the reversible reaction $A \rightleftharpoons B$ on-chip. The normalized forward and reverse rate constants were $\beta_1 = \beta_2 = 1$. The initial concentration of A was varied by changing the digital code D_{IC} between 7 and 13, i.e., $I_A(0)$ was varied over a range of $2^{13-6} : 1 = 64 : 1$. The initial concentration of B was fixed to a low value by setting $D_{IC} = 0$, resulting in $I_B(0) \approx 0$. This reaction should obey the conservation law $I_A + I_B = I_A(0) + I_B(0)$. As a result the steady-state values of I_A and I_B should be

$$\begin{aligned} I_{A,ss} &= \left(\frac{\beta_1}{\beta_1 + \beta_2}\right)(I_A(0) + I_B(0)) \\ I_{B,ss} &= \left(\frac{\beta_2}{\beta_1 + \beta_2}\right)(I_A(0) + I_B(0)) \end{aligned} \tag{6.97}$$

In this case we used $\beta_1 = \beta_2$, resulting in $I_{A,ss} = I_{B,ss} = (I_A(0) + I_B(0))/2 \approx$

$I_A(0)/2$. Figure 6-21 shows that the reaction indeed reaches a steady state where both species concentrations are equal. In addition, this steady state value is $\approx I_A(0)/2$ and scales exponentially with D_{IC}, as expected. However, we also notice a slow, but noticeable decrease in the "steady-state" value with time. In other words, I_A and I_B decrease slowly while tracking each other, instead of remaining fixed at $I_{A,ss}$ and $I_{B,ss}$. This behavior is because of random mismatch, and corresponds to the unwanted "slow" mode that we had earlier predicted theoretically. We implemented the same reaction using other reaction circuits on the same chip, and noted that the magnitude and sign of this slow node was highly variable, as expected. In particular, the sign of the mode, i.e., whether it caused exponential decay or growth, was often concentration-dependent, and could be changed by varying initial conditions.

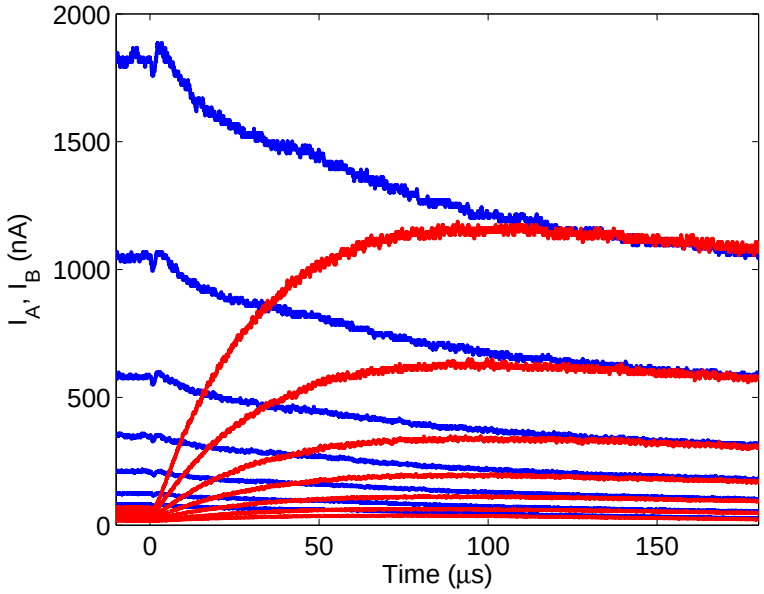


Figure 6-21: Measured concentrations of A (blue) and B (red) in the first-order reversible reaction $A \rightleftharpoons B$. The plots correspond to different initial concentrations of A. The initial concentration of B was fixed at a low value.

We simulated the zeroth-order reaction $[\,] \rightarrow A$ for various values of the normalized rate constant β. The initial concentration of A was set to a low value. The equation that describes the concentration of A as a function of time is $dI_A/dt = \alpha k I_0/X_0$, where k is the rate constant, X_0 is the reference concentration, I_0 is the reference current, and α is the speedup factor. Also, $\alpha k/X_0 = \beta/\tau_0$. Thus, we expect I_A to

increase linearly with time with a slope given by $\beta I_0/\tau_0$.

Figure 6-22 shows measured values of dI_A/dt as a function of β. We notice that dI_A/dt increases with β, as expected. However, very low values of β result in a slope that is significantly lower than $\beta I_0/\tau_0$. This behavior probably occurs because the current charging V_A, which is proportional to β, becomes comparable to transistor junction leakage when β is very small. In addition, dI_A/dt tends to saturate to a fixed value for large values of β because the output buffers used to measure it have limited slew rates. Finally, the three plots shown in Figure 6-22 were measured by connecting V_A to three different output buffers on the same chip. There is static offset between the buffers, as in Figure 6-19.

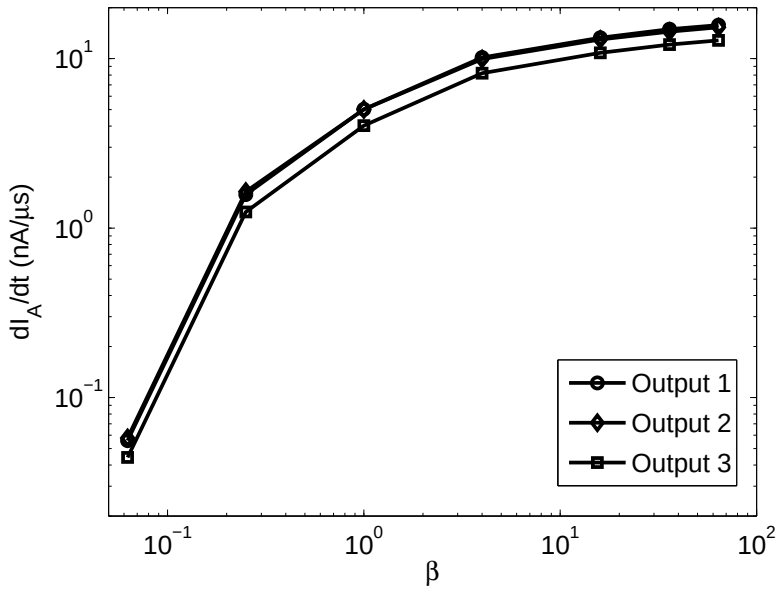


Figure 6-22: Measured rate of change of the concentration of A in the zeroth-order reaction $[\,] \rightarrow A$, as a function of the normalized rate constant β. Multiple independent measurements of I_A were made by connecting V_A to different output buffers on the same chip.

We verified that our chip responded to external inputs by studying the reaction $IN \rightarrow A$, where IN is the input species. In this particular case we kept IN constant with time. The initial concentration of A was set to a low value. The equation that describes the concentration of A as a function of time is $dI_A/dt = \alpha k I_{IN}$, where k is the rate constant, and α is the speedup factor. Also, $\alpha k = \beta/\tau_0$. Thus, we expect

I_A to increase linearly with time with a slope given by $\beta I_{IN}/\tau_0$. Figure 6-23 shows measured values of dI_A/dt as a function of both I_{IN} and β. We notice a good match to theory when the slope is less than 20nA/μs. Measured slopes saturate beyond this value because of two reasons. First, the output buffers used to measure V_A have limited slew rates. Secondly, at high values of I_A transistors enter moderate and strong inversion, which increases the value of τ_0 and decreases dI_A/dt.

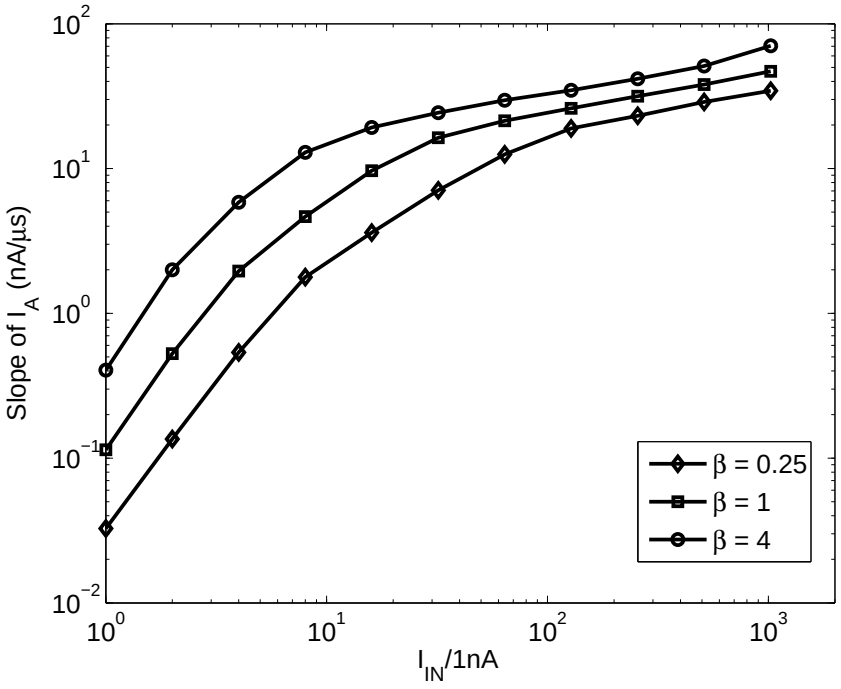


Figure 6-23: Measured charging slope of the first-order reaction $IN \rightarrow A$, where IN is an external input that is constant with time, for various values of I_{IN}. Plots are also shown for several values of β.

We can analyze output slew rate limitations as follows. Let us suppose that the maximum current that can be produced by the buffers is I_{max}. As a result the maximum rate of change of the output voltage is $(dV_{OUT}/dt)|_{max} = I_{max}/C_L$, where C_L is the load capacitance. The output current I_{OUT} is exponentially related to V_{OUT}, giving us

$$\left.\frac{dI_{OUT}}{dt}\right|_{max} = \frac{I_{OUT}}{(\phi_T/\kappa)} \left.\frac{dV_{OUT}}{dt}\right|_{max} = I_{OUT}\left(\frac{I_{max}}{C_L\phi_T/\kappa}\right) \tag{6.98}$$

We see that $(dI_{OUT}/dt)|_{max} \propto I_{OUT}$. Thus it is hard to detect rapid changes

in small currents. We typically used $I_{max} = 5\mu$A and $C_L = 25$pF, resulting in $(dV_{OUT}/dt)|_{max} = 200$mV/$\mu$s and $(dI_{OUT}/dt)|_{max} \approx 5.4 I_{OUT}/\mu$s. For example, consider the curves in Figure 6-23, which were drawn by fitting straight lines to the measured values of I_A. They saturate because I_A cannot move rapidly when it is small, i.e., at the beginning of the charging process. Our analysis predicts that the maximum initial charging slope is limited to $\approx 5.4 I_A(0)/\mu$s, where $I_A(0)$ is the initial value of I_A. Measured results are in fair agreement with this prediction.

We showed earlier that the volume of the compartment containing a product species is proportional to the value of the capacitor that stores its chemical potential. Consider the simple unidirectional transformation reaction $A \rightarrow B$. In this case, we have $W_B \propto C_B$, where W_B and C_B are the reaction volume and capacitor size of species B, respectively. Figure 6-24 shows transient responses of this reaction for various values of C_B. We used the following set: $C_B = \{0.5, 1, 2, 4, 8\}$pF. The species A and B were assumed to have high and low initial conditions, respectively, and the normalized rate constant was fixed at $\beta = 4$. The simulation speedup factor is given by $\alpha = \beta/(\tau k)$, where $\tau = C\phi_T/(\kappa I_0)$ and k is the chemical rate constant. Thus, the value of α should scale as $1/C_B$, i.e., as $1/W_B$, when β and k are fixed. In other words, the speedup factor is inversely proportional to reaction volume.

Experimentally, Figure 6-24 shows that k indeed decreases as C_B increases. However, the measured rate constants only vary by a factor of 6 as the capacitance varies by a factor of 16 between 0.5pF and 8pF. We believe that the rise and fall times for low values of C_B are being limited by the bandwidth of the buffers that drive the data capture equipment. In addition, we notice overshoot in the I_B curves at very low values of C_B. Parasitic poles within the translinear filters are probably responsible for this effect.

As a final example, we implemented the simple reaction system $A + A \rightarrow B$, $B \rightarrow C$, which consists of one second-order and one first-order reaction, both in software (using MATLAB) and on our chip. The system was initialized at time $t = 0$ with a high initial concentration of A and low initial concentrations of B and C. The MATLAB simulation used an optimized version of the Gillespie stochastic simulation

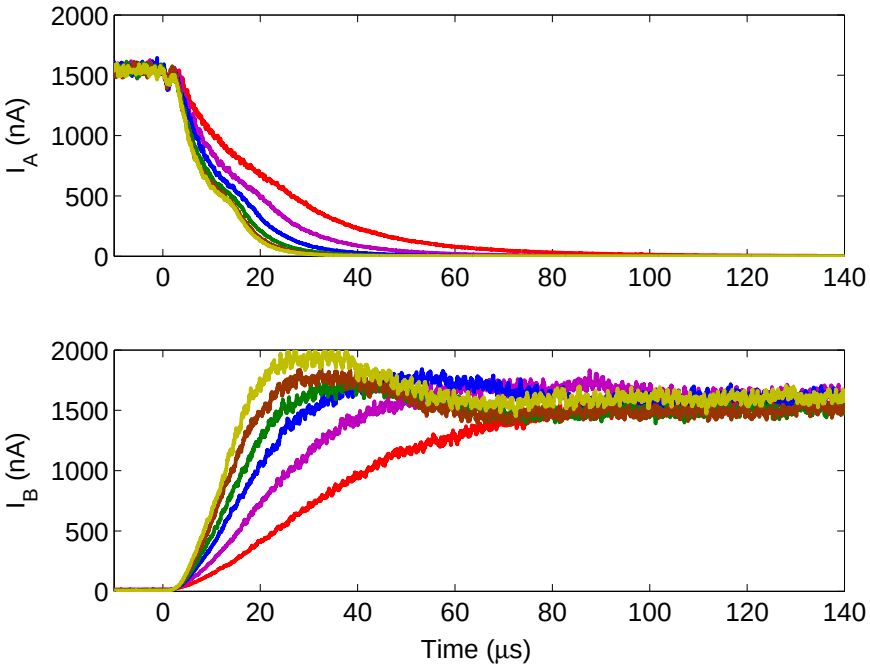


Figure 6-24: Measured species concentrations of the reaction $A \rightarrow B$ for a fixed value of $\beta = 4$ and varying reaction volume W_B.

algorithm (SSA) [92], with the initial number of molecules of A set to a value that results in the same SNR as obtained experimentally from our chip (approximately 32dB).

Figure 6-25 compares the results of simulation and experiment. The two sets of trajectories are very similar, being always within 10% of each other. Since biological systems are both noisy and heterogenous, this level of accuracy may be sufficient for simulating many interesting phenomena. We see that the chip runs approximately 30 times faster than the simulation, which was performed on a 2.4GHz quad-core desktop computer. The speed advantage increases with the complexity of the reaction network: The simulation time of this optimized SSA scales as $\log(r)$, where r is the number of reaction channels [90], whereas it is independent of r on the chip.

When the SNR of *every* species is high enough, we can in principle use deterministic differential equations instead of the SSA. Since they ignore noise, the former run much faster then the latter, particularly when SNR levels are high. However, the SNR of each species varies with time, making it difficult to determine *a priori* if

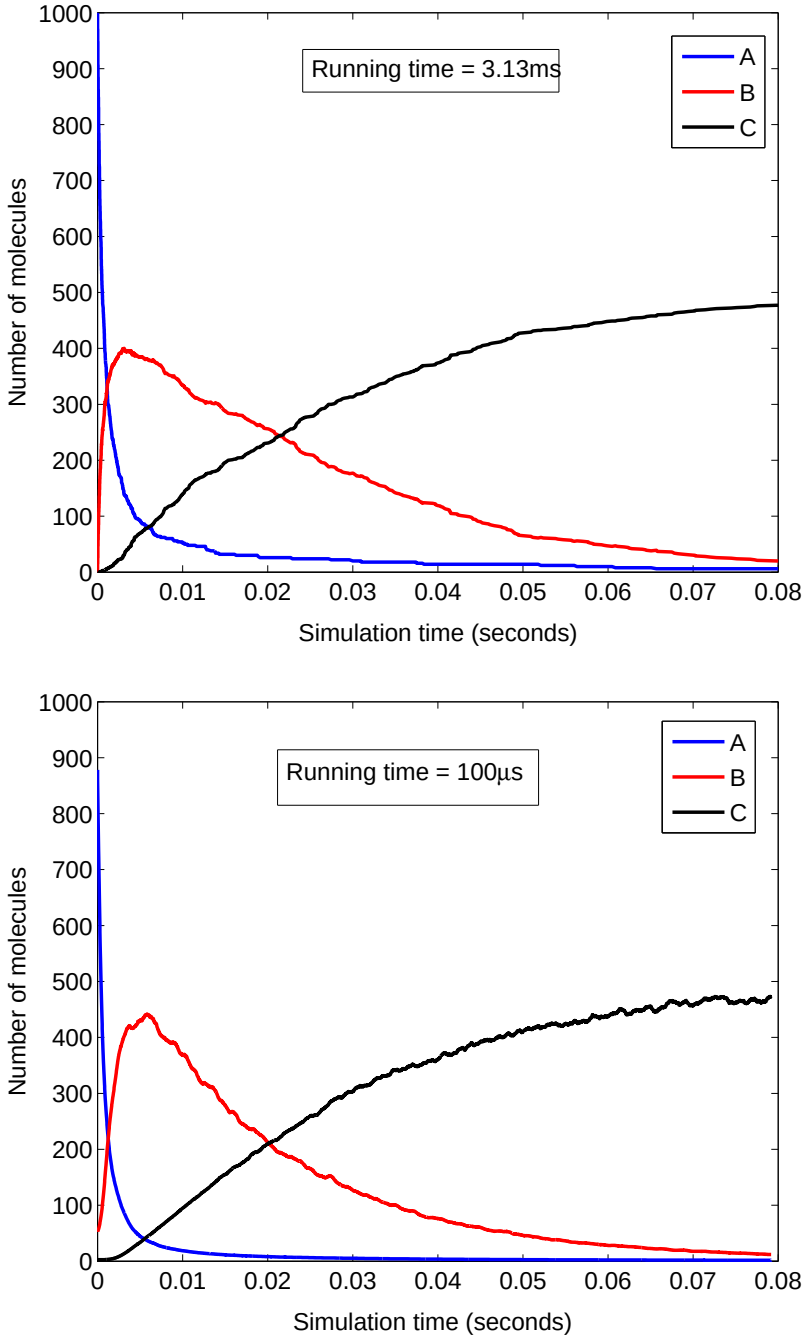


Figure 6-25: Software simulation (top) and measurements from our chip (bottom) of the dynamics of the system of chemical reactions described in the text.

the resultant loss in accuracy will be acceptable. We can potentially avoid this issue entirely by using our chips, since they run stochastic simulations with no performance penalties.

Figure 6-26 shows the measured dynamics of the same set of reactions as in Figure 6-25. However, in this case we varied the rate constant of the second reaction, i.e., $B \rightarrow C$, while keeping rate constant of the first reaction, i.e., $A+A \rightarrow B$, fixed. Plots are shown for the following values of β_2, the normalized rate constant of the second reaction: $\beta_2 = \{0.0625, 0.25, 1, 2.25, 4, 9\}$. We see that the intermediate species B reaches a maximum concentration and then decays away to zero. Larger values of β_2 result in lower maximum concentrations and higher rates of decay. This behavior matches what we would expect theoretically.

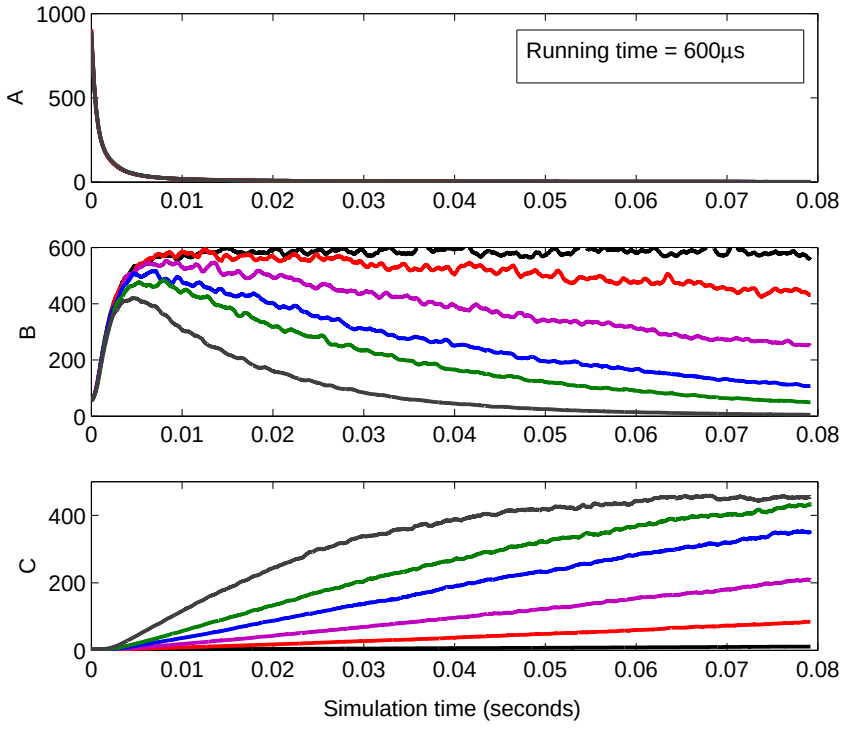


Figure 6-26: Measurements from our chip of the dynamics of the system of reactions $A + A \rightarrow B$, $B \rightarrow C$. Plots are shown for various values of the rate constant of the second reaction. The y-axis indicates the simulated number of molecules of each species.

6.5 Appendix A: Software

There are several ways for the user to interact with our chip. The most obvious is to keep the network topology fixed, but change the values of the rate constants. The goal of such programming may be to match model outputs with experimental data or theoretical predictions. Some form of supervised learning can be used to determine the optimum parameter values. Such learning may be implemented using analog hardware, digital hardware or software. It is easiest to store parameter values digitally by using current DACs, as we have described. However, since the values only need to be held for as long as a single simulation is running, i.e. a relatively short period of time, analog storage can also be used [218]. Our chips also support a second level of programmability, namely changing network topology, i.e., connectivity. In theory, this process can also be automated using learning, although in general it is much harder to learn connectivity than it is to learn parameter values.

Programming can be carried out both off-line (i.e. before a simulation is started) or on-line (i.e. while a simulation is running). In either case, programming is usually performed with a digital computer, since it allows us to implement arbitrarily complex programming algorithms. The resultant system is usually considered a hybrid computer, i.e., a computer that contains both analog and digital subsystems, and interfaces between them [141, 142, 260].

Our chemical network simulator chip is best viewed as the analog subsystem or coprocessor within a hybrid computer. The chip is difficult to use without specialized software, which we wrote in MATLAB. The program, named *protein_ui*, is essentially a hardware driver. It provides a set of high-level, chip-specific functions that abstract out all low-level communication with the chip. This set of functions may be viewed as a rudimentary Application Programming Interface (API). The functions within the API can be grouped into broad categories:

1. Utility functions for displaying help files and interacting with MATLAB.

2. "Write" functions for adding and modifying reactions, outputs, initial conditions

and genes, and also loading mass-action models in KroneckerBio format[9].

3. "List" functions for viewing the current contents of the chip.

4. "Read" functions for performing simulations and capturing data from the chip.

The *protein_ui* program was written in a modular, object-oriented way, which allows it to be easily modified to work with new hardware. The help file of this program, which includes names and brief descriptions of the functions within the API, is reproduced below for reference.

- *protein_ui* help file (version 0.22, 03/01/09)

```
--------------------------------------
1. Utility functions:
help: display this help file
bye/done/exit: finished, exit to MATLAB
--------------------------------------
2. Write functions:
chip_reset: reset the chip (all reactions will be lost)
add_output: add an output from the chip
add_reaction: add a reaction to the chip
change_reaction: change a reaction rate constant
load_model: load a KroneckerBio model into the chip
delete_output: delete an output from the chip
delete_reaction: delete a reaction from the chip
set_initial: set one initial condition
set_initials: set all initial conditions
--------------------------------------
3. List functions:
```

[9]KroneckerBio is a MATLAB-based toobox for simulating and analyzing mass-action models of biological systems. The software is currently under development. For more details, please contact Prof. Bruce Tidor (tidor@mit.edu).

```
list_initial: list all initial conditions
list_outputs: list all assigned outputs
list_reactions: list all assigned reactions
list_sv: list all assigned state variables
---------------------------------------
4. Read functions:
set_scope: change oscilloscope settings
simulate: perform a single simulation and grab data
simulate_repeat: perform several simulations and grab data
---------------------------------------
```

A description of our test setup, including programming and data acquisition hardware, is provided in an appendix attached to the next chapter of this thesis.

6.6 Appendix B: Some Notes on Thermodynamics

Thermodynamic potentials reach minimum values at thermodynamic equilibrium. Systems at constant temperature and volume minimize Helmholtz free energy, while those at constant temperature and pressure minimize Gibbs free energy. Concentrations of all chemical species are constant in time at equilibrium. As a result, the net production rate of all species is zero: production (positive) fluxes are exactly balanced by consumption (negative) fluxes. This is known as flux balance, and corresponds to KCL in electrical networks. In addition, the flux of every chemical reaction must be exactly balanced by that of its reverse reaction at thermodynamic equilibrium. This is the principle of detailed balance, and ensures that the net flux around any closed loop of reactions (cycle) is always zero. Such cycles can be present in a non-equilibrium steady state (NESS), though flux balance must still be satisfied.

Most chemical and electronic systems are not, strictly speaking, in thermodynamic equilibrium, i.e., there are non-zero gradients in intensive system variables like temperature, chemical potential and electrical potential[10]. For example, if the potentials

[10]These intensive variables are similar in the sense that they all measure average energies per

at the two terminals of a diode are different, the electrical potential graidient causes net current flow. Therefore we cannot simply assume the presence of Boltzmann distributions in such systems. However, consider pieces of the system that are small enough for the change ΔX in an intrinsic variable X to be much smaller than the average value X, but that are much larger than the mean free path λ for collisions between particles. Each particle in such subsystems will collide many times with other particles with almost the same thermodynamic properties before it can reach a region with different properties. Such subsystems are therefore said to be at *local thermodynamic equilibrium* (LTE), and Boltzmann distributions will still apply inside them. Thus, for LTE to apply, $\lambda \ll X/(dX/dy)$, where y is the spatial variable.

For example, semiconductor devices are usually modeled as LTE systems that exhibit Boltzmann statistics at every point, i.e., by assuming that carrier concentrations are $\propto \exp\left(-\left[E_i - E_F(y)\right]/kT\right)$, where $E_F(y)$ is known as the quasi-Fermi energy. Applying non-zero potential across the device causes a gradient in $E_F(y)$; based on our earlier reasoning, the LTE approximation is valid as long as $E_F/(dE_F/dy)$ is much larger than λ. At 300K, $\lambda \approx 20$nm for heavily-doped ($\approx 10^{19}\text{cm}^{-3}$) and ≈ 40nm for lightly-doped ($\approx 10^{16}\text{cm}^{-3}$) silicon [307]. Experimentally, exponential flux-potential curves usually imply LTE.

particle. For example, temperature is the average internal, i.e., kinetic energy per particle, while chemical and electrical potentials are the average chemical and electrical potential energies per particle, respectively.

Chapter 7

Circuit Models of Genetic Networks

> *They are in you and me; they created us, body and mind; and their preservation is the ultimate rationale for our existence. They have come a long way, those replicators. Now they go by the name of genes, and we are their survival machines.*
>
> – Richard Dawkins, The Selfish Gene

In this chapter we describe fast, scalable hardware models of genetic networks. We also present experimental results from a prototype chip that was designed for simulating gene activation, transcription and translation.

7.1 Models of Genetic Networks

7.1.1 Binding and Activation

Genes are transcribed into messenger RNA (mRNA) by an enzyme known as RNA polymerase (RNAp). The rate of transcription is regulated by other proteins, known as transcription factors, that promote (activate) or hinder (repress) the action of RNA polymerase. Transcription factors bind to specific DNA regions, known as binding sites. Binding sites are usually clustered into so-called enhancer and promoter regions

located near the beginning of the gene. Activator proteins bind at the enhancer site, while repressor proteins usually bind somewhere on the promoter, as shown in Figure 7-1. We will denote such transcription factors as X. The inducer or substrate S is typically a small molecule that binds to a transcription factor (whether repressor or activator). Inducer binding converts inactive transcription factors into active forms that can bind to the DNA:

$$S + X \leftrightarrow X^* \tag{7.1}$$

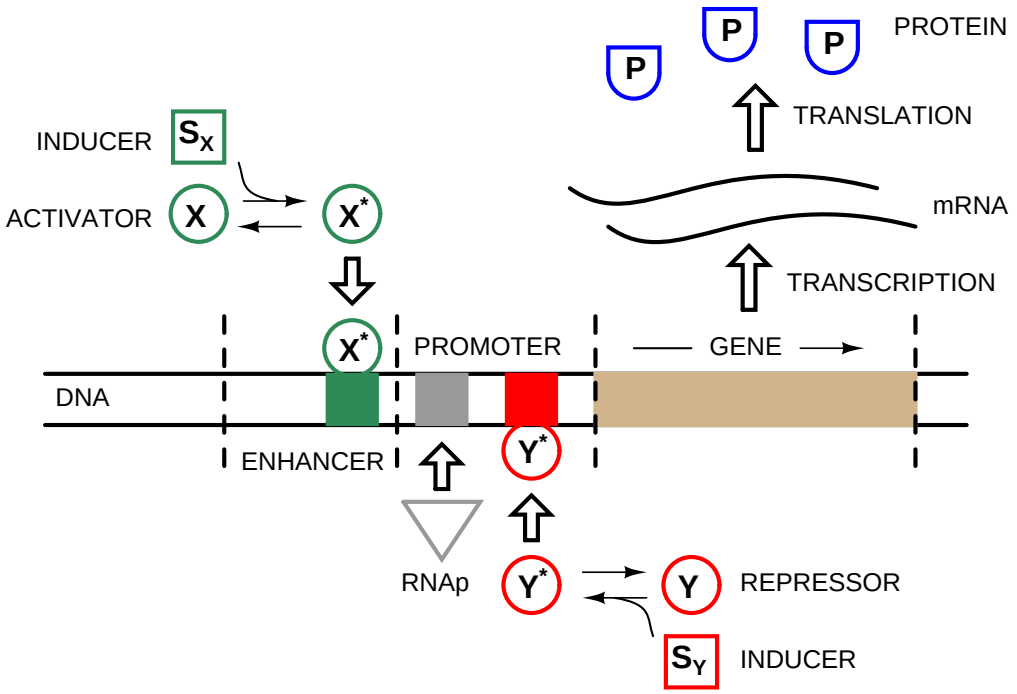


Figure 7-1: A highly simplified view of gene regulation. The gene is activated and repressed by transcription factors X and Y, respectively. Each transcription factor is converted into its active form by binding to an inducer.

where X and X^* are the inactive and active forms of the transcription factor, respectively. We will assume that this reaction is always at equilibrium. The typical parameter values quoted in Table 6.2 indicate that this asumption is valid. As a result, we have

$$[X^*] = \frac{[S]\,[X]}{K_{SX}} \tag{7.2}$$

where K_{SX} is the dissociation constant of the reaction. We will also assume that $[X^*] \ll [S]$ and $[X]$, i.e., that S and X are present in excess. As a result $[S]$ and $[X]$ are not depleted by the reaction, and can be treated as external inputs to the system. On the chip, we can program K_{SX} between $I_0/2^6$ to $2^4 I_0$ using a DAC, where I_0 is a constant reference current.

The relationship between the concentrations of active transcription factors X^* and the rate of transcription of a gene is known as its *input function*. Most genes are regulated by multiple transcription factors. As a result, input functions are usually multi-dimensional, i.e., depend on multiple variables. The measured input functions of bacterial genes are diverse and often non-monotonic [135], and those of eukaryotic genes are frequently even more complex. In many cases, however, these functions show *separation of variables*, i.e., can be factorized into products of simple, single-variable functions. Such functions are often assumed to be proportional to the fraction of DNA binding sites B occupied by an activator, or left vacant by a repressor. The co-operative binding of n transcription factor molecules to B may be formally expressed as $B + nX^* \leftrightarrow B^*$, where B and B^* represent vacant and occupied binding sites, respectively. We assume that this binding reaction is always at equilibrium, an approximation which is again justified by the parameter values shown in Table 6.2. As a result the single-variable input functions are described by Hill functions:

$$\begin{aligned} f_A(X^*) \equiv \beta_{max}\left(\frac{B^*}{B+B^*}\right) &= \beta_{max}\left(\frac{(X^*/K_I)^n}{1+(X^*/K_I)^n}\right) \\ f_R(X^*) \equiv \beta_{max}\left(\frac{B}{B+B^*}\right) &= \beta_{max}\left(\frac{1}{1+(X^*/K_I)^n}\right) \end{aligned} \tag{7.3}$$

where f_A and f_R are the input functions for an activator or repressor, respectively, K_I is the concentration of X^* at which half the binding sites are occupied, β_{max} is the maximal rate of transcription, and n is known as the Hill coefficient. Values of $n > 1$ occur due to co-operative binding. As n increases the functions f_A and f_R become switch-like, i.e., transition abruptly between 0 and β_{max} as the value of $[X^*]$ crosses K_I. In the limit as $n \to \infty$ they may be represented as two-state, or logic functions:

$$
\begin{aligned}
f_A(X^*) &= \beta_{max}\theta(X^* > K_I) \\
f_R(X^*) &= \beta_{max}\theta(X^* < K_I)
\end{aligned} \tag{7.4}
$$

where the "threshold" function $\theta(\ )$ is equal to 1 if its argument is true, and 0 otherwise. In this approximation, transcription abruptly turns on or off when the concentrations of activators or receptors cross the threshold level K_I. Simple differential-pair-based circuits can be used to implement Hill functions of the form shown in (7.3) when n is an integer, i.e., for $n = 1, 2, ...$ On this chip, however, we used the logic approximation shown in (7.4) for simplicity.

Promoter and enhancer regions in our model are assumed to bind N different transcription factors, i.e., the genes have N-dimensional input functions. We know that in general the input function cannot be factorized into the product of single-variable functions such as those shown in (7.4). However, we know that arbitrary Boolean functions, i.e., functions of binary variables, can always be written in one of two equivalent *canonical forms*. These forms are known as "sum of products" (SoP) and "product of sums" (PoS). The SoP form is the sum (logical OR) of *minterms*. Minterms of N-variable Boolean functions are formed by ANDing together each variable or its negation (inverse). For example, XYZ, $X\overline{Y}Z$ and $\overline{X}Y\overline{Z}$ are minterms of the boolean function $f(X, Y, Z)$. An N-variable function has 2^N minterms.

Our approach to realizing arbitrary input functions was inspired by the SoP canonical form. We made two changes to generalize its applicability. Firstly, the i-product term is not restricted to a minterm, which must include all N variables, but can be the product of the normal or negated forms of n_i variables, where $0 \leq n_i \leq N$. Here $n_i = 0$ corresponds to product terms that are always 0 or 1, independent of the inputs. It is easy to see that the canonical SoP form corresponds to the special case where $n_i = N\ \forall i$. By allowing n_i to vary, we can express many functions with far fewer product terms than the 2^N minterms required for the SoP form.

Our second change to the SoP form was more profound. We did not restrict our

output to be the logical OR of the product terms (which are Boolean variables), but allowed it to be any arbitary *weighted sum* of them. Thus, our output is no longer a Boolean variable. In fact, it is a "soft", or analog OR of terms that are formed by "hard", or binary AND operations. Intuitively, each product term is allowed to set its own maximal transcription rate β_{max}, and the total transcription rate is found by adding the rates of all active terms. This approach has a couple of advantages. Firstly, it is more general than a purely Boolean function. Secondly, it is easy to implement in hardware, since the product terms only require logic, and KCL can be used for weighted summations of currents. The output, which we refer to as the activation strength I_{ACTV}, controls the rate of transcription. Physically, activation strength is proportional to the number of active copies of a gene, and the rate at which transcripts are produced from each active copy. Formally, it is given by

$$I_{ACTV} = I_0 \sum_{i=1}^{s} \beta_i \left[\prod_{j=1}^{n_i} \left\{ f_A\left(X_k^*\right), f_R\left(X_k^*\right) \right\}_j \right] \tag{7.5}$$

where s is the number of product terms, $k \in \{1, 2, ..., N\}$, the β_i are dimensionless numbers, and I_0 is a constant reference current. The term within curly braces in (7.5) denotes the selection of either the activator function f_A or the repressor function f_R for any transcription factor X_k. We also define the Boolean variable $ACTV$ as the logical OR of the same product terms as in (7.5):

$$ACTV = OR \left[\prod_{j=1}^{n_1} \left\{ f_A\left(X_k^*\right), f_R\left(X_k^*\right) \right\}_j, ..., \prod_{j=1}^{n_s} \left\{ f_A\left(X_k^*\right), f_R\left(X_k^*\right) \right\}_j \right] \tag{7.6}$$

In other words, the gene is being actively transcribed whenever $ACTV = 1$, with a non-zero activation strength I_{ACTV}. It is inactive whenever $ACTV = 0$, which also implies $I_{ACTV} = 0$. It is also interesting to speculate on other ways of implementing genetic input functions. The basic goal is to approximate an arbitrary multi-dimensional function using a finite set of easily-implemented basis functions. We have described one approach to the problem, but there are many others. Even

restricting ourselves to SoP and PoS-type expansions leads to several distinct possibilities. For example, one can first perform soft-OR operations, threshold the outputs and then form product terms using logical-AND operations. Alternatively, both OR and AND operations could be "soft" - in which case, the order in which they are applied matters! At this stage, it is not clear to us, given the large diversity of genetic input functions, what the best way to approximate them might be.

7.1.2 Transcription and Translation

We shall assume that a single mRNA transcript is directly translated into a single protein. This assumption often leads to an acceptable model of the dynamics of real genes and gene networks [4]. However, one should be aware of the fact that such models are gross simplifications of real cellular processes. For example, prokaryotic genes are often organized into units, known as *operons*, that share a common regulatory region and are transcribed into a single mRNA molecule. Such 'polycistronic' mRNA molecules code for multiple proteins. On the other hand, eukaryotic genes are organized into regions that code for proteins, known as exons, and non-coding regions, known as introns. Primary mRNA transcripts contain both exonic and intronic regions. However, the introns are subsequently removed, in a process known as *gene splicing* to form the mature mRNA that is eventually translated. It is not difficult, in principle, to develop more complicated models that include such effects, and implement them in hardware.

We model transcription and translation as pseudo-chemical reactions. They are not true chemical reactions because the production, of, say mRNA from a gene does not result in the gene being consumed. In general, reaction fluxes in such unidirectional systems only change the concentrations of products, not reactants. The simplest reasonable model that describes transcription and translation is given by

$$X \xrightarrow{k} Y \xrightarrow{\gamma} \phi \tag{7.7}$$

where X and Y are the reactant and product, respectively, of either process, and

ϕ is a degraded, inactive form of Y. The symbols k and γ represent the rates of first-order pseudo-reactions, and are known as the synthesis and degradation rates of Y, respectively. The dynamics of Y are described by the equation

$$\frac{dY}{dt} = kX - \gamma Y \tag{7.8}$$

The frequency response of this system is that of a first-order low-pass filter with the following transfer function

$$\frac{Y}{X} = \frac{k/\gamma}{1 + s/\gamma} \tag{7.9}$$

Thus the DC gain and time constant of this filter are given by k/γ and $1/\gamma$, respectively. The time constant $1/\gamma$ is often referred to as the *lifetime* of Y. We can exactly emulate the dynamics described above on-chip by using a current-mode low-pass filter. The dynamics of this filter are based on the following constraint, which is established by a translinear loop:

$$I_B I_X = (I_A + I_C)\, I_Y \tag{7.10}$$

where I_A and I_B are constant currents, I_C is the current flowing through the capacitor C connected to the output of the filter and I_X and I_Y are the concentrations of the input X and output Y, respectively. In the case of transcription, $I_X = I_{ACTV}$, the activation strength of the gene, while I_Y refers to the concentration of mRNA. Similarly, in the case of translation I_X and I_Y refer to the concentrations of mRNA and translated protein, respectively. By using the dynamic translinear principle, we can write

$$I_C I_Y = \left(\frac{C\phi_T}{\kappa}\right) \frac{dI_Y}{dt} \tag{7.11}$$

Substituting (7.11) in (7.10), we get

$$\frac{dI_Y}{dt} = \left(\frac{\kappa I_B}{C\phi_T}\right) I_X - \left(\frac{\kappa I_A}{C\phi_T}\right) I_Y \tag{7.12}$$

Comparing equations (7.8) and (7.12), we see that the systems are dynamically equivalent, with an electronic speedup factor of α, if

$$\begin{aligned} \frac{I_B}{I_0} &= \alpha\tau_0 k \\ \frac{I_A}{I_0} &= \alpha\tau_0\gamma \end{aligned} \tag{7.13}$$

where, as before $\tau_0 = C\phi_T/(\kappa I_0)$, and I_0 is a constant reference current. The currents I_A and I_B are programmable using a binary-weighted DAC, such that

$$\beta_j \equiv \frac{I_j}{I_0} = \beta_{min}(1 + D_j) \tag{7.14}$$

where $j \in A, B$ and β_{min} is the minimum possible value of the dimensionless number β_j used to set γ and k. Also, D_j, the DAC code, is an integer between 0 and $2^N - 1$. On this chip we used $\beta_{min} = 1/4$ and $N = 5$, allowing β_j to be varied between 1/4 and 8.

7.1.3 Transcription and Translation Delays

We know that RNA polymerase moves at a finite velocity along the gene during transcription. As a result it takes a finite amount of time to "read" a gene and produce each mRNA transcript. This time-span is known as the *transcription delay*. Eukaryotic cells, particularly from multi-cellular organisms, often have large genes with many introns, resulting in long transcription delays. For example, an everage human gene is 27kb (base pairs) long, of which only 5% consists of exons [280]. As a result, transcription delays are often of dynamical significance in eukaryotic cells, but not in prokaryotic cells, which have short genes with no introns [4].

RNA polymerase velocities in mammalian cells range from 18-72 nucleotides/second. Similar numbers have been measured for other species, including prokaryotes (see the BioNumbers website for references). However, these transcript elongation rates are best viewed as upper bounds, since the enzyme frequently pauses during tran-

scription in response to rearrangements of the activation complex, i.e., transcription factors bound to the enhancer and promoter regions. Such pauses can significantly reduce the *average* transcript elongation rate. For example, a peak elongation rate of 70 nucleotides/sec was measured to result in an average elongation rate of 6.3 nucleotides/sec in mammalian cells [49]. At this rate a typical 27kb gene is read in 70 minutes. The time needed for gene splicing must be added to this number to get total transcription delay. Moreover, individual molecules within RNA polymerase populations exhibit rather broad distributions in peak and average elongation rates.

There is also significant delay involved in translation, because ribosomes synthesize proteins relatively slowly. Typical rates range from 15 amino acids/sec in *E. coli* to 0.75 amino acids/sec in rabbits[1]. Thus, the backbone of a typical protein, which contains about 300 amino acids, takes between 20 seconds and 7 minutes to form. The protein then folds into its active form, usually within a few milliseconds.

In general transcription and translation delays will be significant if their values are comparable to or larger than the lifetimes of their associated products, namely mRNA and protein. In general, as shown in Table 6.2, mRNA lifetimes are much shorter than protein lifetimes. In fact, proteins are often stable, and their effective lifetimes are often equal to the cell generation time. In other words, protein lifetimes are often set by dilution due to cell growth, as in (6.34), and not degradation. Thus we expect transcription delays to have much more dynamical significance than translation delays. We therefore ignored translation delays in our model, but retained transcription delays.

We will ignore the complicated internal dynamics of transcript formation. Instead, we simply assume that gene activation turns on mRNA formation after some transcription delay time T_D. In other words, the concentration of mRNA begins increasing at $t = T_D$ when $ACTV$ goes high at $t = 0$. However, the delay T_D is sign-sensitive, and occurs only when the gene is activated. Thus, when the gene is deactivated mRNA formation is assumed to stop immediately, because incomplete

[1]DNA synthesis is much more rapid than protein synthesis. In *E. coli*, for example, DNA polymerase synthesizes at approximately 10^3 nucleotides/sec.

transcripts do not lead to functional proteins and can be ignored. Figure 7-2 shows a simple circuit that generates such sign-sensitive delays. The output of the D-type flip-flop is $ACTD$, the delayed version of $ACTV$. It is immediately reset when $ACTV$ goes low. However, it can only go high on rising clock edges, which are delayed by a time T_D.

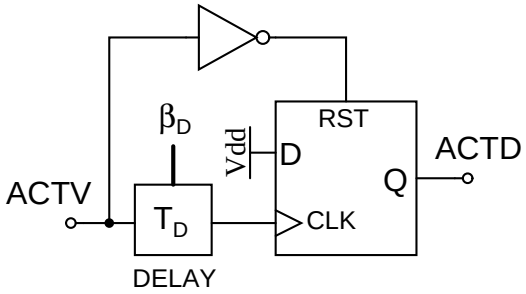


Figure 7-2: A simple circuit that generates sign-sensitive transcription delay.

The value of T_D is the product of average elongation rate and gene length, and is assumed to be independent of activation strength I_{ACTV}. However, we allow the user to independently set the transcription delay for every gene by making T_D dependent on an external control parameter β_D, as shown in Figure 7-2.

7.1.4 Noise

The study of stochasticity, or random fluctuations in gene expression, has recently received much attention. Several recent review articles provide useful summaries of the field [134, 182, 232]. Stochastic behavior is usually studied by observing the concentration of the protein encoded by the gene as a function of time. Fluorescence microscopy is often used for this purpose. In general, as explained in Chapter 5, fluctuations in protein concentration can be attributed to two sources: intrinsic and extrinsic. Intrinsic noise is uncorrelated across genes, and arises from the inherent stochasticity within the processes of gene activation, transcription and translation.

Extrinsic noise is correlated across multiple genes, and arises from fluctuations external to the gene itself, such as changes in transcription factor and inducer concentrations with time. We can model extrinsic noise by varying the inputs to our gene

circuit, i.e., inducers and transcription factors (S and X, respectively) as functions of time.

Models of intrinsic noise have been proposed for both prokaryotic and eukaryotic cells. The basic model for prokaryotes was first proposed by Oudenaarden et al. in 2002 [221]. The authors of this paper studied the bacterium *B. subtilis*. An important model for eukaryotes was proposed by Raser and O'Shea in 2004 based on observations of the yeast *S. cerevisiae* [234]. Both models can be summarized by the system of reactions shown in Figure 7-3(a) for a single active transcription factor X^*.

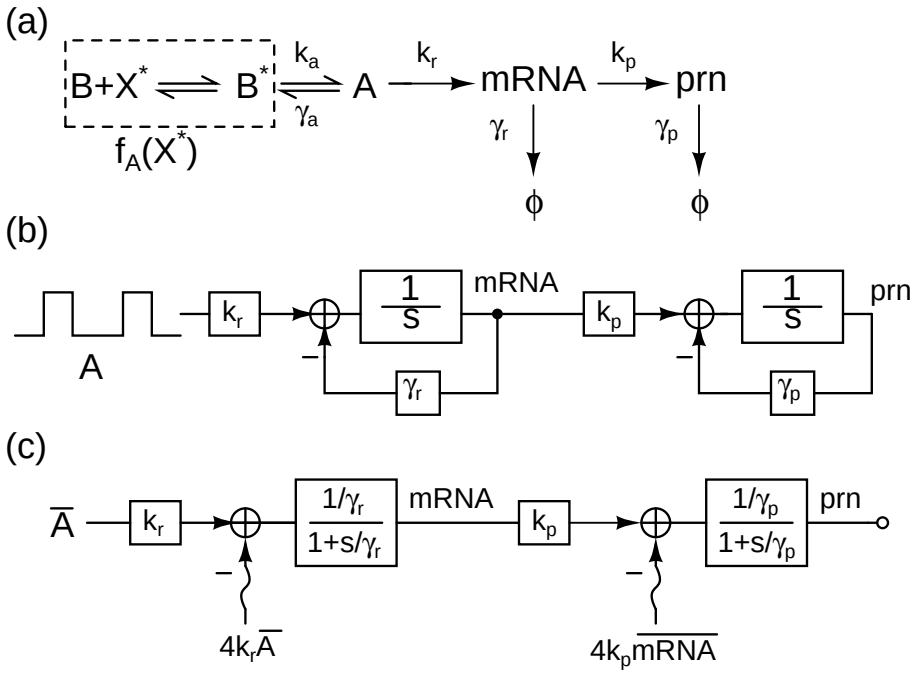


Figure 7-3: A model of intrinsic noise in gene expression. The model is shown as (a) a set of pseudo-chemical reactions, (b) a feedback block diagram and (c) simplified block diagram with noise sources. In (c) wavy lines represent the power spectral density of random fluctuations (noise) in steady state.

In prokaryotes, transcription factor binding occurs on timescales that are much faster than transcription and translation, as shown in Table 6.2. As a result, fluctuations caused by binding and unbinding of X^* are low-pass filtered and do not contribute significantly to variability in gene expression. In eukaryotes, however, transcription may not be initiated even when all required transcription factors are bound, because it depends on the completion of other regulatory processes, such as

chromatin remodelling [232]. Such processes are often relatively slow, causing gene activation to fluctuate on slow timescales and generating an additional source of variability.

Consider the model shown in Figure 7-3. The transcription factor binds cooperatively to vacant binding sites B in the promoter region, as described earlier. As a result, the fraction of sites that are bound, which is defined as the activation function $f_A(X^*)$, is given by the Hill function shown in (7.3). In prokaryotes we can assume that B^* corresponds to the activation function $ACTV$, abbreviated in this section to A. In eukaryotes, however, activation is also subjected to other regulatory control, as described in the previous paragraph. The dynamics of these regulatory processes is modelled by defining B^* and A as two separate states, and allowing transitions between them with first-order rate constants k_a and γ_a. In the former state transcription factors are bound but the gene is not active, while in the latter the gene is being actively transcribed.

We are interested in calculating the noise in mRNA and protein concentrations when X^* is high enough for most activator binding sites to be occupied, i.e. $B^* \gg B$. In prokaryotes, the result is a gene that is always on, i.e. $A = 1$. However, the model predicts that eukaryotic genes stochastically transition between on and off states ($A = 1$ and $A = 0$, respectively). Thus, for eukryotic genes A can be modeled as a two-state *random telegraph signal*.

Assuming random, uncorrelated transitions results in the fluxes $k_a B^*$ and $\gamma_a A$ having Poisson distributions. The average residence time in each state is $1/k_a$ and $1/\gamma_a$, respectively. Also, residence times follow the exponential waiting-time distribution corresponding to a Poisson variable. Mathematically, this sytem is identical to one that describes threshold-voltage fluctuations in a MOSFET due to single oxide traps. We analyzed the latter in Chapter 5 in the context of flicker noise. Using our previous results, we see that the gene is in its actively transcribing state for an 'on' fraction D given by

$$D = \frac{k_a}{(k_a + \gamma_a)} \tag{7.15}$$

Thus, the mean value of A is given by $\overline{A} = DA$. Also, the power spectrum of A is that of a first-order low-pass filter with time constant $\tau_A = 1/(k_a + \gamma_a)$. A signal-flow block diagram view of the entire model is shown in Figure 7-3(b). Dynamically, we see that both transcription and translation are modeled as first-order low-pass filters with DC gain. This is the same result as derived mathematically in Section 7.1.2. The cutoff frequencies of the filters are given by $\tau_r = 1/\gamma_r$ and $\tau_p = 1/\gamma_p$, respectively, while their DC gains are given by $A_r = k_r/\gamma_r$ and $A_p = k_p/\gamma_p$, respectively.

Let us consider the case when $\tau_A \ll \{\tau_r, \tau_p\}$, i.e., promoter transitions are rapid compared to both transcription and translation. In this case most of the variance due to A lies at frequencies larger than $1/\tau_r$ and $1/\tau_p$ and will be low-pass filtered out. Thus A can be replaced with its mean value $\overline{A} = AD$, where $D = 1$ for prokaryotic genes. We can easily extend the noise analysis that follows to the more general case where τ_A is arbitrary.

We assume that every reaction flux consists of uncorrelated arrivals or departures of molecules, and thus displays Poisson statistics. The difference between the mRNA synthesis and degradation fluxes is integrated into the mRNA concentration $[mRNA]$. Each of these fluxes has a power spectral density equal (PSD) to twice the mean flux, just like any Poisson variable. The two mean fluxes are $k_r\overline{A}$ and $\gamma_r[mRNA]$, leading to PSDs of $2k_r\overline{A}$ and $2\gamma_r[mRNA]$, respectively. These PSDs are independent, and therefore the total PSD that is filtered by the transcription filter is the sum of the two. However, at steady state the two fluxes must have identical mean values, and so the total PSD is $2 \times 2k_r\overline{A} = 4k_r\overline{A}$. Similarly, the total PSD that is filtered by the translation filter is $2k_p[mRNA] + 2\gamma_p[prn] = 4k_p[mRNA]$ in steady state, where $[prn]$ is the protein concentration. The resultant steady-state noise model is shown in Figure 7-3(c).

Our noise model is very similar to a situation one might encounter in electronics while analyzing a cascade of low-pass filters, and yields similar insights. It also allows us to analytically calculate the variance in mRNA and protein concentrations. We note that the noise bandwidth of a first-order low-pass filter is $\pi/2$ times larger than its 3dB bandwidth. Thus the noise bandwidth of the transcription filter is

$(\pi/2) \times 1/(2\pi\tau_r) = \gamma_r/4$. Similarly, the noise bandwidth of the translation filter is $\gamma_p/4$. We can now immediately write down, *by inspection*, that

$$\begin{aligned} \sigma^2_{mRNA} &= 4k_r\overline{A} \times \frac{1}{\gamma_r^2} \times \frac{\gamma_r}{4} \\ &= \frac{k_r\overline{A}}{\gamma_r} \\ &= \overline{mRNA} \end{aligned} \tag{7.16}$$

for mRNA, and

$$\begin{aligned} \sigma^2_{prn} &= \left(4k_r\overline{A}\left(\frac{k_p}{\gamma_r}\right)^2 + 4k_p\overline{mRNA}\right) \times \frac{1}{\gamma_p^2} \times \frac{\gamma_p}{4} \\ &= \overline{prn}\left(1 + \frac{k_p}{\gamma_r}\right) \\ &\equiv \overline{prn}(1+b) \end{aligned} \tag{7.17}$$

for protein, where

$$\begin{aligned} \overline{mRNA} &= \frac{k_r\overline{A}}{\gamma_r} \\ \overline{prn} &= \left(\frac{k_p}{\gamma_p}\right)\overline{mRNA} \end{aligned} \tag{7.18}$$

are the mean concentrations of mRNA and protein, respectively. We see that the concentration of mRNA has variance equal to its mean, i.e., behaves like a Poisson random variable. However, the upstream noise from transcription increases the variance of the protein concentration by a factor $(1 + b)$ over a Poisson variable. The quantity $b = k_p/\gamma_r$ is known as the *burst parameter*. Physically, it is equal to the average number of protein molecules produced from a single mRNA transcript, and often has a value > 1. We see that the protein has $(1 + b)$ times more noise than we might have naively expected. Interestingly, however, the quantity $(1 + b)$ is inde-

pendent of the signal level, i.e., mean protein concentration. In electronic circuits a similar situation often occurs, with $(1+b)$ being referred to as the *excess noise factor*, or the *effective number of noisy devices*.

While deriving (7.17) we implicitly assumed that $\gamma_p \ll \gamma_r$, i.e., the translation bandwidth is much smaller than the translation bandwidth, and sets the overall noise bandwidth $\gamma_p/4$ that determines σ^2_{prn}. This case is typical of prokaryotic genes, and many eukaryotic genes. However, some eukaryotic mRNA lifetimes are very long, which might make our assumption invalid. We can easily analyze this case by making the opposite assumption, i.e. $\gamma_p \gg \gamma_r$. In this case the transcription bandwidth is much smaller than the translation bandwidth, and the slow mRNA fluctuations are simply amplified by the DC gain k_p/γ_p of the translation filter. Thus, (7.17) is modified to

$$\begin{aligned} \sigma^2_{prn} &= 4k_r\overline{A} \times \frac{1}{\gamma_r^2} \times \frac{\gamma_r}{4} \times \left(\frac{k_p}{\gamma_p}\right)^2 + 4k_p\overline{mRNA} \times \frac{1}{\gamma_p^2} \times \frac{\gamma_p}{4} \\ &= \overline{prn}\left(1 + \frac{k_p}{\gamma_p}\right) \\ &\equiv \overline{prn}\,(1 + A_p) \end{aligned} \tag{7.19}$$

where $A_p = k_p/\gamma_p$. We see that (7.17) and (7.19) have similar forms, except for the fact that mRNA fluctuations affect the protein in proportion to the low-frequency translation gain $A_p = k_p/\gamma_p$ instead of the burst parameter $b = k_p/\gamma_r$. Our noise model predicts the same result as the full stochastic analysis described in [221]. However, it has the important advantage (for electrical engineers, anyway) of being simple to analyze using standard block-diagram techniques, thus yielding important insights quickly. Our current gene circuit implements the same model as shown in Figure 7-3(c), with $I_{ACTV} = \overline{A}$, and other concentrations also represented by currents.

7.2 Design of the Genetic Network Chip

7.2.1 System Design

Our first genetic network chip contained 6 gene circuits and 16 external (off-chip) inputs, and was fabricated in standard 0.18μm CMOS technology. A high-level block diagram of the chip is shown in Figure 7-4. A power-on reset circuit initializes all on-chip memory to known states when the chip first turns on. The input conditioning circuits and programming interface were very similar to those used in the chemical network simulator chip. A connection matrix, consisting of an array of programmable switches, allows any external input to act as an inducer or transcription factor for any gene. Programmable parameter values were set on-chip by individually-addressable transistor or capacitor DACs. Both the connection matrix and DACs were programmed by loading bits into a 30-bit long global shift register. The contents of this shift register can be divided into two sections:

- A 10-bit address section, which can be further split into the 4 most significant digits (used to locate a gene) and the 6 least significant digits (used to locate sub-circuits within that gene).
- A 20-bit data section.

Each gene circuit contains programmable circuits for modeling activation, transcription and translation. The outputs generated by the chip consist of the activation signal $ACTV$, the mRNA concentration, and the protein concentration associated with each gene.

7.2.2 Input Selection and Activation

Figure 7-5 is a high-level block diagram of the circuits that simulate the induction and binding of transcriptional activators and deactivators to DNA. Each gene can be activated or repressed by a maximum of N active transcription factors, denoted by X^*. Thus, the input function of the gene can be at most N-dimensional. Active

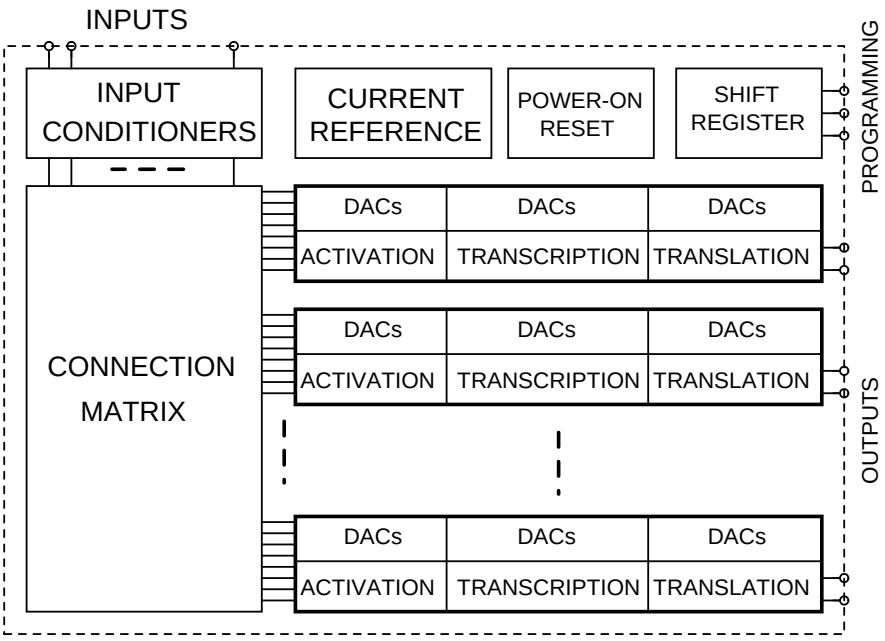


Figure 7-4: A high-level block diagram of the entire genetic network simulator chip.

transcription factors can be induced from their inactive forms, denoted by X, by molecules known as inducers and denoted by S, We allow each transcription factor to have its own inducer, resulting in a total of $N + N = 2N$ input signals for the gene (N inducers and N trancsription factors). The connection matrix allows the concentration of each of these signals, denoted by I_S and I_X respectively, to be set equal to any of the $2N$ external inputs, denoted by I_{IN}.

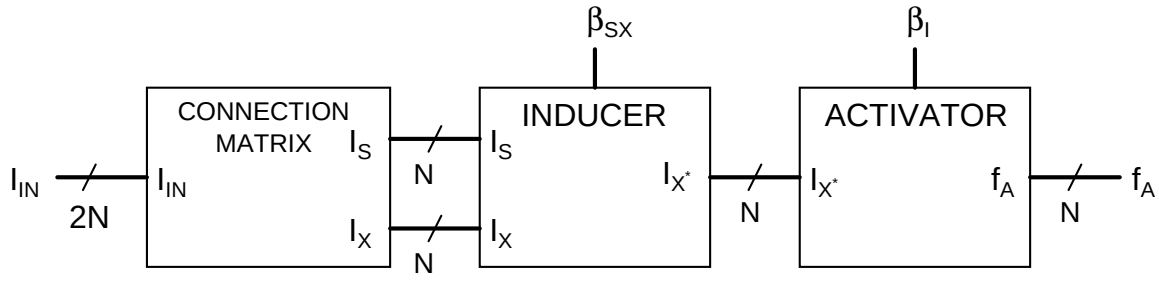


Figure 7-5: High-level block diagram of the circuits that simulate the induction and binding of transcriptional activators and deactivators to DNA.

The concentration of any active transcription factor is given by the electronic analog of (7.2), i.e., with chemical concentrations replaced by currents:

$$I_{X^*} = \frac{I_S I_X}{\beta_{SX} I_0} \tag{7.20}$$

where β_{SX} is a programmable, dimensionless number and $\beta_{SX} I_0$ is the electronic analog of the dissociation constant K_{SX}, i.e.,

$$\beta_{SX} I_0 = K_{SX} \left(\frac{I_0}{X_0} \right) \Rightarrow \beta_{SX} = \frac{K_{SX}}{X_0} \tag{7.21}$$

where X_0 is the reference chemical concentration. On this chip we can program the value of β_{SX} between 2^6 and $1/2^4$, i.e., over a 10-bit range. Figure 7-6 shows the circuit that implements (7.20) for a single inducer-transcription factor pair. We use N of these circuits per gene. Each circuit uses a translinear loop that guarantees the constraint $I_{X^*} (\beta_{SX} I_0) = I_S I_X$. We have used this circuit extensively in the past. For example, it was used within the reaction circuits of the chemical simulator chip, as shown in Figure 6-5.

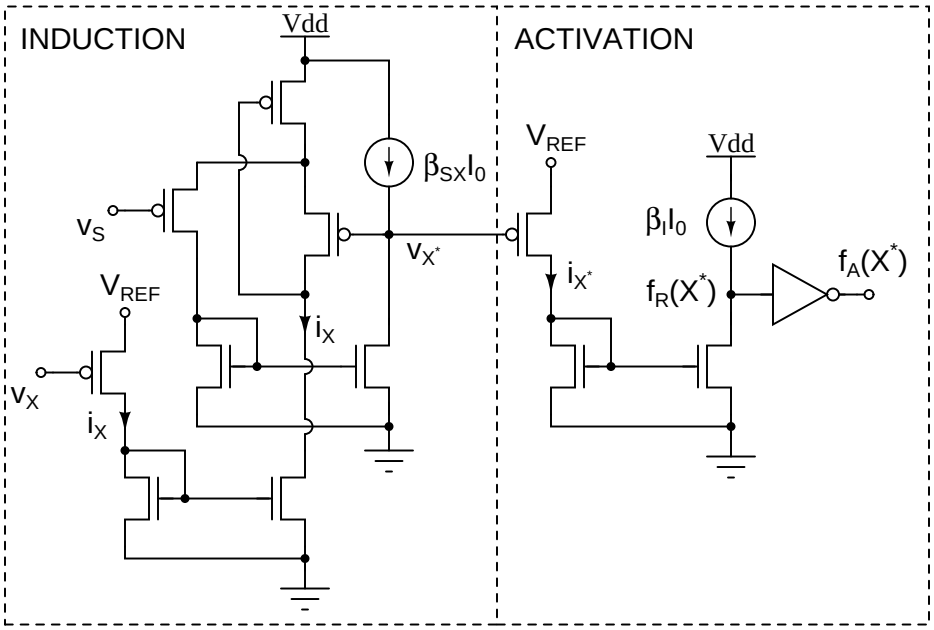


Figure 7-6: Schematic of the circuit that generates the input function of a single transcription factor X. The function can either be activating or repressing (f_A or f_R, respectively).

The active transcription factor concentration, denoted by X^*, is compared with

the threshold concentration K_I to generate the "logic" activation function f_A, and its inverted version f_R, as shown in (7.4). In our electronic analog, these equations are written as

$$\begin{aligned} f_A\left(I_{X^*}\right) &= \theta\left(I_{X^*} > \beta_I I_0\right) \\ f_R\left(I_{X^*}\right) &= \theta\left(I_{X^*} < \beta_I I_0\right) \end{aligned} \tag{7.22}$$

where β_I is a dimensionless number whose value on this chip can be programmed between 2^6 and $1/2^4$, i.e., over a 10-bit range. The value of β_I is set such that the threshold of the electronic analog matches that of the original biological system, i.e.,

$$\beta_I I_0 = K_I \left(\frac{I_0}{X_0}\right) \Rightarrow \beta_I = \frac{K_I}{X_0} \tag{7.23}$$

The circuit that implements (7.22) is a simple current comparator that requires only two transistors, and is also shown in Figure 6-5. The circuit acts as an inverter with a current-source load, and therefore produces the repression function f_R. Another inverter is used to convert this signal into the activation function f_A.

The next step is to combine N single-variable activation functions into the activation function of the entire gene, as shown in (7.5) and (7.6). It is easier to implement these equations if the number of terms in the products and summations are fixed, i.e., n_i and s are constants. We can indeed make n_i and s constant without loss of generality by allowing each term to be chosen, not from the set $\{f_A, f_R\}$, but from the larger set $\{f_A, f_R, 0, 1\}$. In this case, all the products can have N terms, because inputs that should be ignored for a given product are assigned the value 1. Similarly, S products can always be summed to form the output, because unwanted products can be removed by assigning 0 to one or more of the terms within them. Here N and S are fixed for a given chip. For example, on the current chip we used $N = S = 8$. Thus, (7.5) and (7.6) may be rewritten as

Table 7.1: Truth table of selector blocks in the PLA

$SEL(1)$	$SEL(0)$	$OUT(IN, f_A)$
0	0	0
0	1	$IN.f_A$
1	0	$IN.\overline{f_A}$
1	1	IN

$$
\begin{aligned}
I_{ACTV} &= I_0 \sum_{i=1}^{S} \beta_i \left[\prod_{j=1}^{N} \left\{ f_A\left(X_j^*\right), f_R\left(X_j^*\right), 0, 1 \right\} \right] \\
ACTV &= OR \left[\prod_{j=1}^{N} \left\{ f_A\left(X_j^*\right), f_R\left(X_j^*\right), 0, 1 \right\} \right]
\end{aligned} \tag{7.24}
$$

where it is understood that the OR operation has S inputs. We can implement (7.24) using a generalization of the well-known Programmable Logic Array (PLA) approach for synthesizing arbitrary Boolean functions. PLAs consist of an AND-plane and an OR-plane. The AND plane consists of an array of AND gates that compute the minterms of the function, while the OR-plane uses OR gates to combine them into the output(s). In our modification of the PLA, the AND gates are replaced by more complex combinational logic blocks. Each block accepts the previous block's output, and ANDs it with a member of the set $\{f_A, f_R, 0, 1\}$, thus building up a product term. Two control bits, $SEL(0)$ and $SEL(1)$, are fed into each block and determine which member of the set is selected for product formation. The truth table for each block is shown in Table 7.1, where IN and f_A are the inputs to the gate and OUT is its output. In addition, since f_A and f_R are binary-valued functions, $f_R = \overline{f_A}$, the logical negation of f_A.

Once the product terms have been formed they can be combined by an S-input OR gate to create $ACTV$, or added together with weighting factors β_s to create I_{ACTV}. We simply use KCL to perform the latter operation. The values of β_s can be individually programmed by transistor DACs over a five-bit range (1/4 to 8, in this case). A simplified block diagram of our generalized PLA is shown in Figure 7-7. Thick lines in the figure denote multi-wire buses. For example, the f_A bus is N bits

wide.

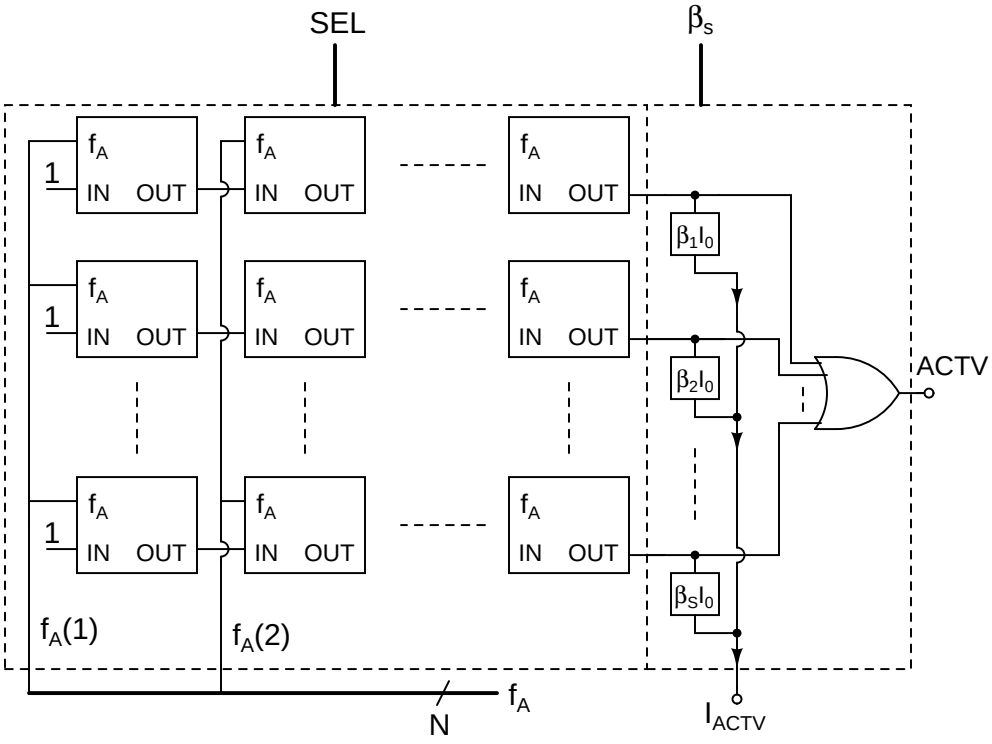


Figure 7-7: Block diagram of a generalized programmable logic array (PLA) circuit that simulates the activation of a single gene by combining the effects of several transcription factors.

7.2.3 Transcription and Translation

Figure 7-8 shows a high-level block diagram of the circuits that simulate the processes of transcription and translation of a single gene. The circuits can be divided into three blocks. The first block contains out model of transcription delay, the second simulates the dynamics of transcription, and the third simulates translation.

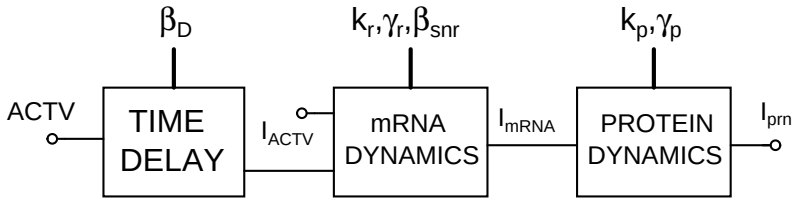


Figure 7-8: Simplified block diagram of the circuits that simulate the processes of transcription and translation of a single gene.

Transcription Delay

On this chip transcription delays were produced by the circuit shown in Figure 7-2. Each delay block was implemented with an M-bit long shift register that was driven by an on-chip, current-controlled ring oscillator running at a frequency f_D. The frequency of the oscillator can be set by the user over a 10-bit range. The DAC value β_D is linearly proportional to the oscillator frequency, i.e., $f_D = \beta_D f_0$, where f_0 is a constant. The value of f_0 is in turn proportional to the global reference current I_0, i.e., $f_0 = I_0/q_{osc}$, where q_{osc} depends on the oscillator design and has units of charge. Thus, we get

$$f_D = \frac{\beta_D I_0}{q_{osc}} \tag{7.25}$$

The quantity q_{osc} can be viewed as the total charge supplied to the capacitors within the oscillator during one oscillation cycle. This charge ultimately comes from the power supply. For ring oscillators $q_{osc} \approx NC_L V_{osc}$, where N is the number of stages, C_L is the capacitance at every stage, and V_{osc} is the oscillation amplitude[2].

The activation signal $ACTV$ is not synchronized with respect to the oscillator. As a result, it goes high at a random time within the oscillator cycle. However, the shift register only loads new values at rising edges of the oscillator. Thus the total delay consists of two parts. The first part consists of the time between $ACTV$ going high and the next rising oscillator edge, and is a uniformly-distributed random number between 0 and $1/f_D$. Assuming that oscillator jitter is negligible, the second part of the delay is fixed and equal to M cycles of the oscillation frequency. Thus, the total delay is given by

$$T_D = \left(\frac{M+1/2}{f_D}\right)\left(1+\frac{\chi}{M+1/2}\right) = \frac{q_{osc}}{I_0}\left(\frac{M+1/2}{\beta_D}\right)\left(1+\frac{\chi}{M+1/2}\right) \tag{7.26}$$

[2]The power dissipated by the oscillator per cycle, which we may think of a figure of merit, is approximately $q_{osc}V_{osc}$, and scales as $C_L V_{osc}^2/Q$, where Q is the open-loop quality factor, i.e., the quality factor of the frequency-selective network within the oscillator. Ring and relaxation oscillators have $Q = 1$, making them less power-efficient than LC oscillators, which can have $Q \gg 1$.

where χ is a random variable that is unifomly distributed between $\pm 1/2$. The mean value and standard deviation of the delay are given by

$$\begin{aligned} \overline{T_D} &= \frac{q_{osc}}{I_0}\left(\frac{M+1/2}{\beta_D}\right) \\ \sigma_{TD} &= \frac{q_{osc}}{I_0}\left(\frac{1}{\sqrt{12}\beta_D}\right) \end{aligned} \tag{7.27}$$

From (7.27), the signal-to-noise ratio (SNR) of T_D is given by $\overline{T_D}^2/\sigma_{TD}^2 = 12\,(M+1/2)^2$. Thus, we can set the mean value of the transcription delay using β_D, and its SNR using M. On this chip we used the fixed value $M = 16$, which results in an SNR of 35dB. In addition, we found that $q_{osc} = 1.5 \times 10^{-12}\text{C} = 9.2 \times 10^6 q$ for $V_{osc} = V_{DD}/2 = 0.9\text{V}$, where q is the electronic charge. Finally, β_D could be varied between 2^6 and $1/2^4$. Thus, for a typical value of $I_0 = 10\text{nA}$, we get $f_0 = 6.8\text{kHz}$, allowing $\overline{T_D}$ to be varied between 38μs and 39ms. Finally, we also allow the user to bypass the delay circuit entirely, thereby setting $T_D = 0$.

Scaling

Equations (7.13) and (7.14) are applicable to both mRNA and protein concentrations. They show that relative synthesis and degradation rates, denoted by β_B and β_A, respectively, are programmable over a 5-bit range. Our goal is to scale the values of these parameters such that this limited dynamic range in β values (2^5:1) is maximally utilized. We came across the same problem while scaling chemical reaction models, and use a similar approach in this case. We assume that

$$\alpha\tau_0\sqrt{k\gamma} = \overline{\beta} \Rightarrow \alpha\tau_0 = \frac{\overline{\beta}}{\sqrt{k\gamma}} \tag{7.28}$$

where $\overline{\beta} = \sqrt{\beta_{min}\beta_{max}} = \sqrt{2}$ is the center of the available dynamic range in β values. Thus, our assumption maps the geometric mean of k and γ to $\overline{\beta}$. Equation (7.13) provides the additional constraint that $\beta_B/\beta_A = k/\gamma$, leading to the following equations:

$$\begin{aligned} \beta_B &= \overline{\beta}\sqrt{\frac{k}{\gamma}} \\ \beta_A &= \overline{\beta}\sqrt{\frac{\gamma}{k}} \end{aligned} \tag{7.29}$$

Equation (7.29) was used to determine the values of β for on-chip implementation. Values of β outside the allowable range were clipped to β_{min} or β_{max}. Finally, the capacitor C was implemented as a 5-bit DAC, and can be varied between 0.25pF and 8pF. By varying C we can adjust $\tau_0 = C\phi_T/(\kappa I_0)$ over a 5-bit range. However, choosing β_A and β_B fixes the product $\alpha\tau_0$. Thus, we get $\alpha \propto 1/\tau_0$, allowing us to control the speedup factor α independent of the values of k and γ.

SNR Adjustment

We have seen that obtaining very low values of SNR with intrinsic device noise would require impractically small capacitors and bias currents. We have designed a feedback loop that solves this problem by dynamically adjusting the level of noise of any reaction flux in a system of chemical reactions. The circuit generates fluctuations that are consistent with Poisson statistics, i.e., ensures that the SNR of the flux is always proportional to its mean value.

Our SNR adjustment loop can be used to set the level of noise in a system of simulated chemical reactions to any desired value. Thus, the circuit allows us to perform stochastic simulations of systems that have small numbers of molecules, and thus low SNR. We have implemented the loop on-chip for the special case of transcription, since the concentration of mRNA within a cell is usually low enough for it to be the noisiest molecular species.

Our circuit is shown in Figure 7-9. A current $2I_A$ that is proportional to the mRNA degradation rate γ is shut off by a pseudo-random waveform generated by the linear-feedback shift register (LFSR). The waveform has 50% duty cycle, resulting in an average degradation current of I_A, as assumed in previous sections. However, in

this case the current is not constant, but generated by a square wave that switches pseudo-randomly between $2I_A$ and 0. The clock frequency of the LFSR is controlled by a current-controlled oscillator (CCO) whose input is a scaled version of the mRNA concentration, i.e. $\beta_{snr} I_{mRNA}$, where β_{snr} is a dimensionless number that can be programmed by the user. The output frequency of the CCO is given by

$$f_{cco} = \frac{I_{mRNA}}{(q_{osc}/\beta_{snr})} \tag{7.30}$$

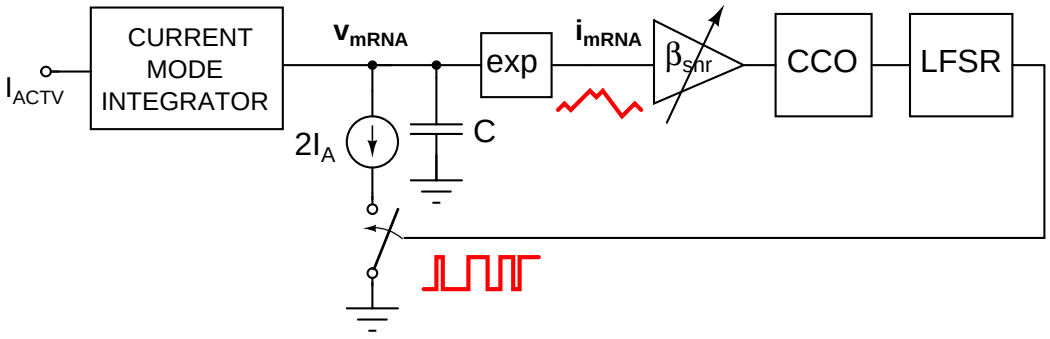


Figure 7-9: Block diagram of a circuit that controls the signal-to-noise ratio of mRNA in a manner consistent with Poisson statistics.

where q_{osc}, which has units of charge, is characteristic of the CCO, as described earlier. LFSRs produce waveforms with properties close to random telegraph (RT) signals, as described in the appendix at the end of this chapter. RT signals consist of random, uncorrelated transitions between two states[3], which in this case are $\pm I_A$. The power spectral density of the switched current source in steady state is given by

$$\begin{aligned} \overline{i_n(f)^2} &= 2I_A^2 T_{cco} \mathrm{sinc}^2\left(\pi f T_{cco}\right) \\ &= \left(\frac{q_{osc}}{\beta_{snr}}\right) \frac{2I_A^2}{I_{mRNA}} \mathrm{sinc}^2\left(\pi f T_{cco}\right) \end{aligned} \tag{7.31}$$

where the CCO time period is $T_{cco} = 1/f_{cco}$, and $\mathrm{sinc}(x) \equiv \sin(x)/x$. We have substituted from (7.30) to get the second line of the equation. We have also assumed that f_{cco} is an instantaneous, or memoryless function of I_{mRNA}. In order for this

[3]More precisely, state transitions of an RT signal constitute a Poisson process.

assumption to be valid we need the CCO frequency to be much larger than the bandwidth of the low-pass filter. In other words, we need $f_{cco} \gg 1/\tau_r$, where $\tau_r = \alpha\tau_0 I_0/I_A$, as defined earlier. The pseudo-noise voltage PSD at the node v_{mRNA} is given by

$$\overline{v_n(f)^2} = \overline{i_n(f)^2}\,|Z(f)|^2 \tag{7.32}$$

where $|Z(f)|$ is the impedance at this node. The PSD of the pseudo-noise component of the mRNA concentration I_{mRNA} is given by

$$\overline{i_{mRNA}(f)^2} = g_m^2\overline{v_n(f)^2} \tag{7.33}$$

where $g_m = I_{mRNA}/(\phi_T/\kappa)$ is the small-signal transconductance of the transistor that converts v_{mRNA} to i_{mRNA}. We are assuming here that the pseudo-noise voltage amplitude at v_{mRNA} is smaller than the linear range ϕ_T/κ of this transistor. Putting together (7.31), (7.32) and (7.33), we find that

$$\overline{i_{mRNA}(f)^2} = 2\left(\frac{q_{osc}}{\beta_{snr}}\right) I_A^2 I_{mRNA}\left[\frac{\mathrm{sinc}\,(\pi f T_{cco})\,|Z(f)|}{(\phi_T/\kappa)}\right]^2 \tag{7.34}$$

The total pseudo-noise power in I_{mRNA} is found by integrating $\overline{i_{mRNA}(f)^2}$ over all frequencies:

$$\overline{i_{tot}^2} = \int_0^\infty \overline{i_{mRNA}(f)^2}df = 2\left(\frac{q_{osc}}{\beta_{snr}}\right)\frac{I_A^2 I_{mRNA}}{(\phi_T/\kappa)^2}\int_0^\infty |Z(f)|^2\mathrm{sinc}^2\,(\pi f T_{cco})\,df \tag{7.35}$$

It is easy to show that $Z(s)$ is given by

$$Z(s) = \frac{1}{C}\left(\frac{\tau_r}{1+s\tau_r}\right) \tag{7.36}$$

We can substitute (7.36) in (7.35) and solve the integral by again assuming that $f_{cco} \gg 1/\tau_r$, which ensures that $\mathrm{sinc}(\pi f/f_{cco}) \approx 1$. In this case we get the same first-order noise integral we have seen many times. The solution is

$$\int_0^\infty |Z(f)|^2 \text{sinc}^2\left(\pi f T_{cco}\right) df \approx \frac{\tau_r}{4C^2} \tag{7.37}$$

Physically, in this case the LFSR looks like a flat, or white noise source over the bandwidth of interest, i.e., $1/\left(2\pi\tau_r\right)$. As a practical matter, the spectrum is flat to within 0.1dB for frequencies $f < f_{cco}/\left(4\pi\right)$, so we can set our lowest allowable CCO frequency to

$$\frac{f_{cco,min}}{4\pi} \approx \frac{1}{2\pi\tau_r} \Rightarrow f_{cco,min} = \frac{2}{\tau_r} \tag{7.38}$$

Equation (7.38) is a version of the Nyquist sampling theorem. The CCO must update the state of the node v_{mRNA} at a rate that exceeds twice its bandwidth, which is approximately $1/\left(2\pi\tau_r\right)$. We can generalize (7.38) to

$$f_{cco,min} = \frac{\alpha_s}{\tau_r} \tag{7.39}$$

where the value of α_s depends on how much deviation from a white noise spectrum we are willing to tolerate. Finally, substituting (7.37) in (7.35), we get

$$\overline{i_{tot}^2} = \left[\frac{I_A\left(q_{osc}/\beta_{snr}\right)}{2C\left(\phi_T/\kappa\right)}\right] I_{mRNA} \tag{7.40}$$

We see that the total noise power of I_{mRNA} is proportional to I_{mRNA} itself. When $\overline{i_{tot}^2} \ll I_{mRNA}^2$, i.e. the SNR is high, the mean mRNA concentration $\overline{I_{mRNA}} \approx I_{mRNA}$, and noise power scales with mean just like a Poisson random variable. On this chip we found that $q_{osc} = 1.5 \times 10^{-12}\text{C} = 9.2 \times 10^6 q$ for $V_{osc} = V_{DD}/2 = 0.9\text{V}$, where q is the electronic charge. We can compare (7.40) with our previous results for the noise of the translinear filters used for modeling chemical reactions. For example, by comparing with (6.89), we see that we have essentially replaced the electronic charge q by the much larger charge q_{osc}/β_{snr}, resulting in lower SNR for the same mean level of mRNA. The SNR at a given mean level can be set by the user through the parameter β_{snr}. Clearly we get SNR $\propto \beta_{snr}$ when all other parameters are fixed.

The minimum achievable SNR is set by the samppling criterion shown in (7.39).

We substituting (7.30) in (7.39), we get

$$\beta_{min} = \frac{\alpha_s q_{osc}}{\tau_r I_{mRNA}} \tag{7.41}$$

We can calculate the noise when $\beta = \beta_{min}$ by substituting (7.40) in (7.40). The result is

$$\overline{i_{tot}^2}\Big|_{\beta=\beta_{min}} = \frac{I_{mRNA}^2}{2\alpha_s} \tag{7.42}$$

As a result, the minimum achievable SNR is

$$\text{SNR}_{\text{min}} = \frac{I_{mRNA}^2}{\overline{i_{tot}^2}} = 2\alpha_s \tag{7.43}$$

For $\alpha_s = 2$, $\text{SNR}_{\text{min}} = 6\text{dB}$. On this chip, β_{snr} could be varied between $1/2^4$ and 2^6. For the typical values $I_{mRNA} = 100\text{nA}$, $I_A = 10\text{nA}$ and $C = 2\text{pF}$, we get $\beta_{min} = 4.3$ for $\alpha_s = 2$, allowing us to vary the SNR between 6dB and 17.8dB. Lower values of SNR can be achieved by lowering α_s, i.e., relaxing our spectral constraint.

We can rewrite (7.40) as

$$\overline{i_{tot}^2} = \left[\frac{(q_{osc}/\beta_{snr})}{2\tau_r}\right] I_{mRNA} \tag{7.44}$$

Equation (7.44) shows a potential problem with our technique. We see that $\overline{i_{tot}^2} \propto 1/\tau_r$, while in the original chemical system the variance in mRNA concentration was simply $\overline{mRNA}$, and independent of τ_r. This is not a problem if τ_r is a constant. However, if τ_r is varied (for example, by programming the value of I_A) we must also vary $\beta_{snr} \propto 1/\tau_r$, i.e., $\beta_{snr} \propto I_A$, to keep the noise invariant. However, this may not always be convenient.

We can calculate the variance of the switched current source to find the root of the problem. It switches between $\pm I_A$ with 50% duty cycle, and thus has a variance $\sigma^2 = I_A^2$. Since $I_A \propto \gamma_r$, the mRNA degradation rate, we get $\sigma^2 \propto \gamma_r^2$. However, the variance of a Poisson variable like the reaction flux should be proportional to the mean arrival rate γ_r, and not γ_r^2. Thus, our circuit does not scale properly with

respect to $\gamma_r = 1/\tau_r$. Figure 7-10 shows a modified circuit that solves this problem, but has not yet been implemented on-chip.

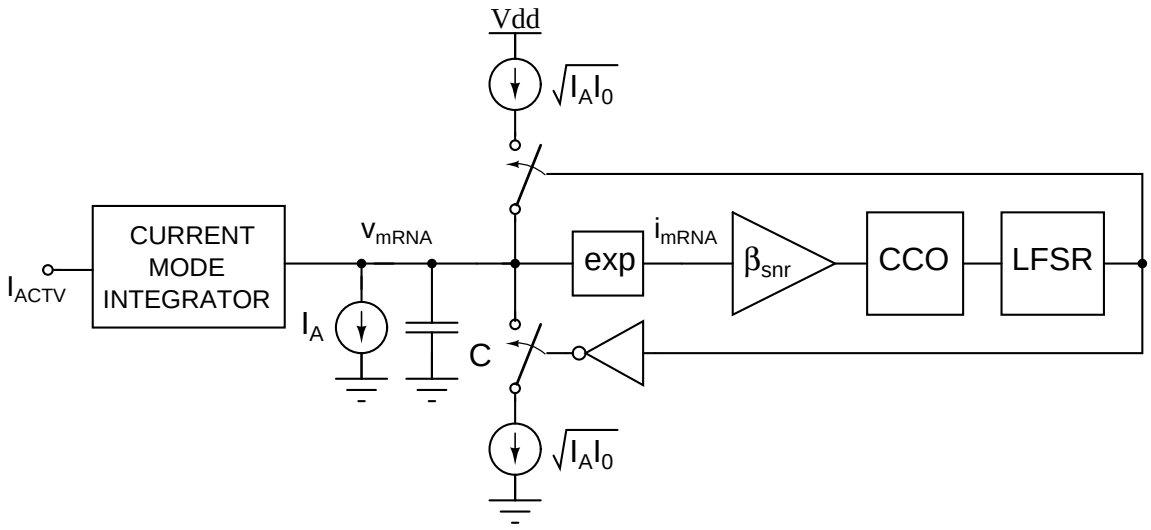


Figure 7-10: Block diagram of an improved circuit for adjusting the signal-to-noise ratio of mRNA in a manner consistent with Poisson statistics.

The improved circuit is similar to the original version, except for the fact that I_A is now a continuous current source, and we inject two pseudo-random switched currents of value $\sqrt{I_0 I_A}$ with opposite polarities at v_{mRNA}. Here I_0 is a globally constant reference current. The current $\sqrt{I_0 I_A}$ can easily be generated from I_A and I_0 by a single-quadrant translinear circuit similar to the one used within the current-mode filter. It is easy to show that the PSD of the switched current sources is modified from (7.31) to

$$\overline{i_n(f)^2} = 2I_0 I_A T_{cco} \mathrm{sinc}^2\left(\pi f T_{cco}\right) \tag{7.45}$$

We see that the variance of these sources is now proportional to I_A, not I_A^2, as desired. The analysis of the improved circuit is very similar to the original version. We find that the total variance in I_{mRNA} is

$$\overline{i_{tot}^2} = \left[\frac{I_0\left(q_{osc}/\beta_{snr}\right)}{2C\left(\phi_T/\kappa\right)}\right] I_{mRNA} = \left[\frac{\left(q_{osc}/\beta_{snr}\right)}{2\tau_0}\right] I_{mRNA} \tag{7.46}$$

where $\tau_0 = C\phi_T/\left(\kappa I_0\right)$. Thus, the total noise $\overline{i_{tot}^2}$ produced by the modified circuit is

proportional to $1/\tau_0$, instead of $1/\tau_A$, which makes it independent of I_A. In addition, the minimum achievable SNR changes to

$$\text{SNR}_{\text{min}} = \frac{I_{mRNA}^2}{\overline{i_{tot}^2}} = 2\alpha_s \left(\frac{I_A}{I_0} \right) \tag{7.47}$$

We see that SNR_{min} is no longer automatically fixed by α_s, but can be lowered by increasing I_0. This is another advantage of the improved circuit shown in Figure 7-10.

We can easily extend our analysis to calculate the noise in I_{prn}, the protein concentration in the presence of the SNR adjustment loop. Consider the same cases as before: $\gamma_p \ll \gamma_r$, which is usually valid, and $\gamma_p \gg \gamma_r$, which occurs in certain eukaryotic genes. Afer some fairly simple algebra, we get

$$\begin{aligned} \overline{i_{prn}^2} &= b \left[\frac{(q_{osc}/\beta_{snr})}{2\tau_0} \right] I_{prn}, \ \gamma_p \ll \gamma_r \\ \overline{i_{prn}^2} &= A_p \left[\frac{(q_{osc}/\beta_{snr})}{2\tau_0} \right] I_{prn}, \ \gamma_p \gg \gamma_r \end{aligned} \tag{7.48}$$

where $b = k_p/\gamma_r$ and $A_p = k_p/\gamma_p$, as before. We can compare (7.48) with the equivalent expressions for the original chemical system, i.e., (7.17) and (7.19). We see that they are completely analogous: Chemical concentrations display Poisson statistics, and variances of currents in the electronic analog also have the usual Poisson form, i.e., $\propto 2q_{eff}\overline{I}\Delta f$, where $q_{eff} = q_{osc}/\beta_{snr}$ is the effective charge and $\Delta f = 1/\tau_0$ is the bandwidth.

We also see that the excess noise factors $(1 + b)$ and $(1 + A_p)$ in the chemical system are replaced by factors of b and A_p, respectively, in the electronic analog. The two sets of excess noise factors are slightly different because our SNR adjustment circuit only models noise from transcription, not translation. We can remove this discrepancy, if necessary, by simply adding another SNR adjustment circuit to model translation noise. This circuit can be identical in form to that shown in Figure 7-10, but with I_{mRNA} replaced by I_{prn}.

Finally, simple modifications to the circuits shown in Figure 7-9 or Figure 7-

10 would allow the output of a single LFSR to introduce the same psuedo-random waveform into more than one current. For example, suppose we have a current that discharges the node corresponding to a reactant species, and another, with the same mean value, that charges a product node. The two currents together simulate a single reaction flux, and fluctuate in sync since they are perturbed by the same LFSR. Thus, our circuit allows us to simulate correlated fluctuations in reaction fluxes. We have shown earlier that intrinsic device noise cannot, in general, generate such correlations, which may have important effects at low SNR.

Current-Mode Filtering

We used similar translinear low-pass filter circuits to model the dynamics of both transcription and translation. However, the filter for modeling transcription, which is shown in Figure 7-11, also included circuitry for SNR adjustment. The theory behind SNR adjustment was described in the previous section. We can enable SNR adjustment by making the control bit EN = 1, or disable it by making EN = 0. The current discharging the output capacitor C has an average value of I_A in both cases, resulting in similar *average* waveforms. However, this average is produced by a constant current of value I_A when EN = 0, and a current that switches between $2I_A$ and 0 with a 50% duty-cycle when EN = 1. The switching waveform in the latter case is set by the LFSR. All switches were implemented as differential, or current-steering structures. Such structures significantly reduce "switching glitches" that result from charge injected to nearby nodes via capacitive coupling.

The translinear filter in Figure 7-11 is similar to that used within the reaction circuits of the chemical simulator chip. The log-compressed version of the input current I_{IN} is denoted by V_{IN}. It is easy to show that the transfer function of this filter in the noiseless case, i.e., when $EN = 0$, is given by

$$\frac{I_{mRNA}}{I_{IN}} = \frac{I_B/I_A}{1 + s\left(\tau_0 I_0/I_A\right)} \tag{7.49}$$

where $\tau_0 = C\phi_T/\left(\kappa I_0\right)$. The values of I_A, I_B, and C can be varied over a 5-bit range,

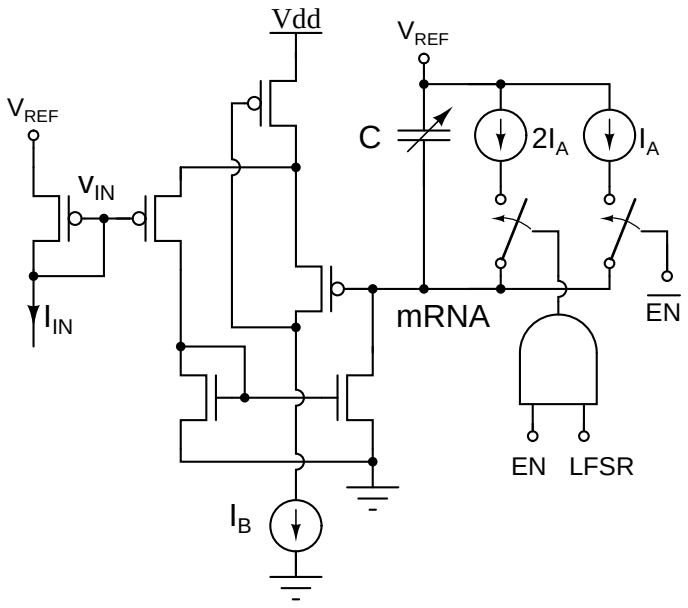


Figure 7-11: Simplified version of the circuit that simulates the process of transcription.

as described previously. In order to model mRNA dynamics we set I_B and I_A based on mRNA synthesis and degradation rates k and γ, respectively, as shown in (7.13). Once I_A and I_B have been set the transfer function in (7.49) is dynamically equivalent to the original chemical transfer function, i.e., (7.9). We used the following scheme for setting the input current I_{IN}:

- When $ACTV = 0$, i.e., the gene is "off": $I_{IN} = I_{OFF}$, where I_{OFF} is a global constant that can be set by the user.
- When $ACTV = 1$, i.e., the gene is "on": $I_{IN} = I_{ACTV}$.

We used a network of switches (not shown in Figure 7-11) to change the input to the filter as a function of the activation signal $ACTV$. The steady-state mRNA concentrations, as predicted by (7.49) are then given by

$$\begin{aligned} I_{mRNA,on} &= \frac{I_{ACTV} I_B}{I_A} \\ I_{mRNA,off} &= \frac{I_{OFF} I_B}{I_A} \end{aligned} \tag{7.50}$$

We define the on/off ratio of the gene as the ratio of steady-state mRNA concentrations in the two cases. This quantity, which may have biological significance, is given by

$$R_{mRNA} = \frac{I_{mRNA,on}}{I_{mRNA,off}} = \frac{I_{ACTV}}{I_{OFF}} \tag{7.51}$$

There are two advantages of this scheme. Firstly, it gives us independent control of $I_{mRNA,on}$ and R_{mRNA}. For example, we can change I_B to vary the former quantity, and I_{ACTV} to vary the latter. Secondly, by making $I_{OFF} > 0$ we ensure that transistors within the filter are always saturated, and that the transfer function in (7.49) is always valid.

The circuit that models translation is identical to that shown in Figure 7-11, except for the fact that the switches are absent and I_A is simply a constant current source. This circuit takes as input $I_{IN} = I_{mRNA}$, and generates a current I_{prn} that is the electronic analog of the protein concentration. We can guarantee that $I_{mRNA} > 0$ in both active and inactive states because of the scheme discussed previously. As a result transistors within the translation filter will also be saturated in both these states. The parameters of this filter, namely I_B, I_A, and C, can be set by the user using 5-bit DACs.

7.3 Measurements

A die photograph of the gene network simulator chip is shown in Figure 7-12. It was mounted on a printed circuit board and interfaced with a desktop computer running MATLAB. Please see the Appendix at the end of this chapter for more details of the chip/computer interface. The nominal power supply voltage was $V_{DD} = 1.6$V.

In our model, a gene that is activated by a single transcription factor X abruptly turns on when the concentration of active transcription factor X^* exceeds the threshold $K_I = \beta_I I_0$. The concentration of X^* depends both on X and the inducer S, as shown in (7.20). Thus, the gene will be 'on' (actively transcribing) if

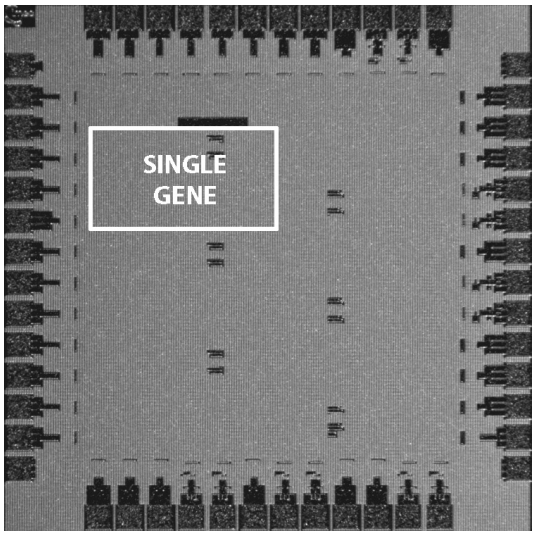


Figure 7-12: Die photograph of the gene network simulator chip.

$$I_S I_X > \beta_{SX} \beta_I I_0^2 \tag{7.52}$$

We programmed the input function of an on-chip gene circuit to have this form, and used a triangular pulse of inducer S to activate it. The pulse is shown in Figure 7-13. It had a peak concentration of 200nA and a total width of 400μs. The concentration of transcription factor X was kept fixed during a particular simulation, but varied between different simulation runs. We also used fixed values of $\beta_{SX} = \beta_I = 1$.

Measured mRNA concentrations behave as expected, and are also shown in Figure 7-13. When I_X is very low the gene never turns on because the inequality in (7.52) is never satisfied. Conversely, when I_X is large the inequality is always satisfied and the gene never turns off. At intermediate values of I_X the gene turns on and off abruptly when $I_S = \beta_I \beta_{SX} / (I_0 I_X)$. As a result the I_{mRNA} waveforms resemble square pulses with widths that increase with I_X. Brief 'glitches' in the I_{mRNA} waveforms are due to unwanted capacitive coupling between V_{mRNA} and nodes on the chip that rapidly switch between V_{DD} and ground, such as the activation signal $ACTV$.

Figure 7-14 shows mRNA concentrations that were produced by the same gene circuit when the triangular inducer waveform shown in Figure 7-13 was replaced by a

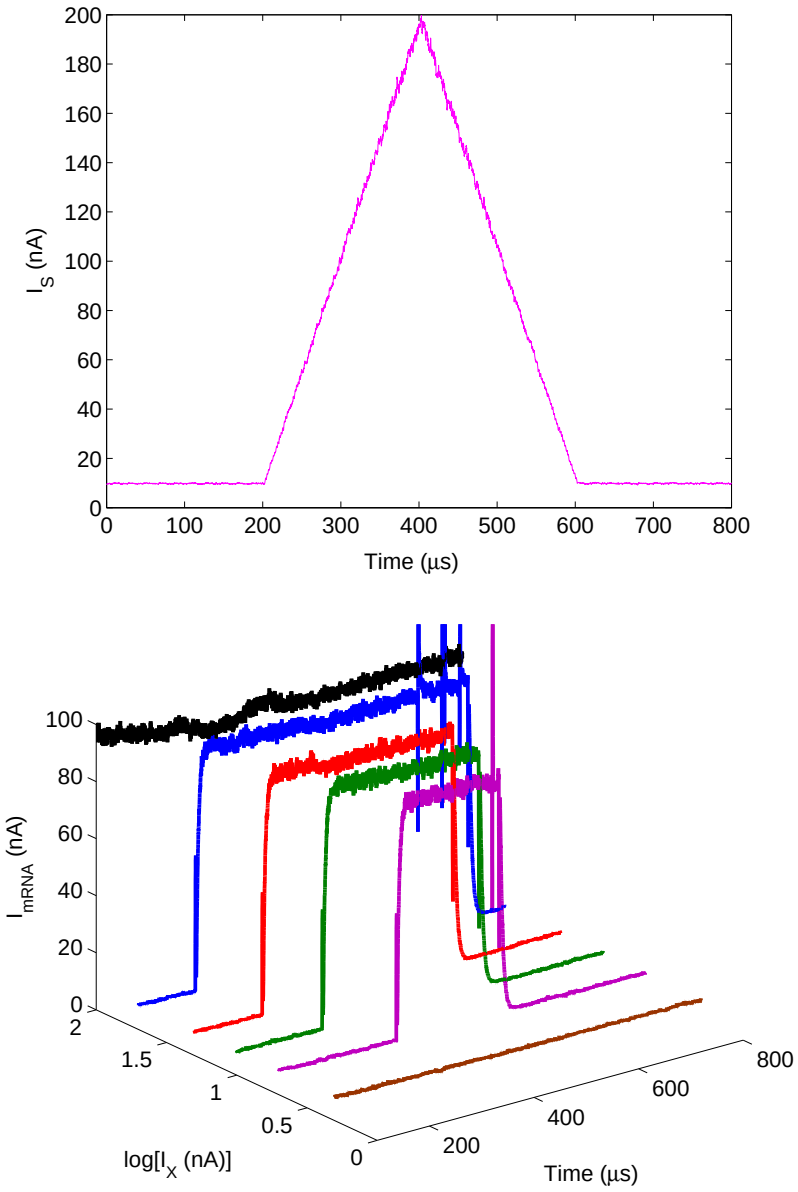


Figure 7-13: Measured mRNA concentration (bottom) in response to a triangular pulse of inducer S (top). Plots are shown for various values of the transcription factor concentration I_X.

square pulse of I_S. The pulse had a width of 80μs, as indicated by the grey rectangle in the figure. Its height was fixed to a value that was high enough to activate the gene, i.e., to satisfy inequality (7.52). We varied C, the size of the capacitor that stores the log-compressed mRNA concentration, to draw the plots in Figure 7-14.

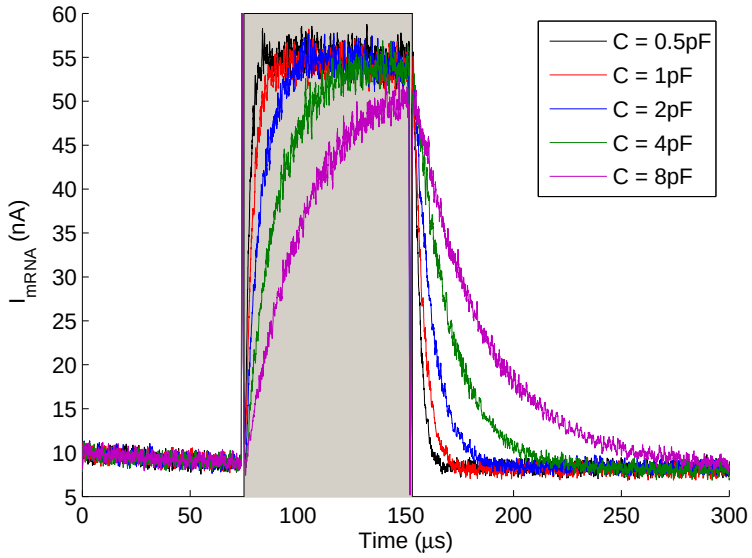


Figure 7-14: Measured mRNA concentration in response to a square pulse of inducer S. The duration of the pulse is indicated by the grey rectangle. Plots are shown for various values of the mRNA capacitor C, which is proportional to the volume of the cell.

The volume W of the simulated reaction volume, i.e., the cell, is proportional to C, as shown earlier for chemical reactions. We used the following set of values: $C = \{0.5, 1, 2, 4, 8\}$pF. The simulation speedup factor is given by $\alpha = \beta_A / (\tau\gamma)$, where $\tau = C\phi_T / (\kappa I_0)$, and $I_A = \beta_A I_0$ sets the mRNA degradation rate γ. Thus, the value of α scales as $1/C$, i.e., as $1/W$, when β_A and k are fixed. In other words, the speedup factor is inversely proportional to reaction volume. Alternatively, the time constant of the low-pass filter due to transcription, which is given by $\tau I_0 / I_A = C\phi_T / (\kappa I_A)$, increases linearly with C. The responses shown in Figure 7-14 behave as predicted. They are low-pass filtered versions of the input pulse, and the filter time constant is proportional to C.

The user can program the mean amount of transcription delay $\overline{T_D}$ using the parameter β_D, as described earlier. In fact, equation (7.27) predicts that $\overline{T_D} \propto 1/\beta_D$.

Figure 7-15 shows the measured transcription delay as a function of β_D. The delay only occurred at rising edges of the $ACTV$ signal, as expected. We see that the delay increases linearly with $1/\beta_D$, also as expected. On this chip, the delay can be programmed over a 10-bit range, i.e., $(D_{DEL,max}/\beta_D)$ can vary between 1 and 2^{10}. Experimentally, we were indeed able to vary the delay over this full range. However, biologically relevant values of delay usually fall within the smaller range shown in Figure 7-15.

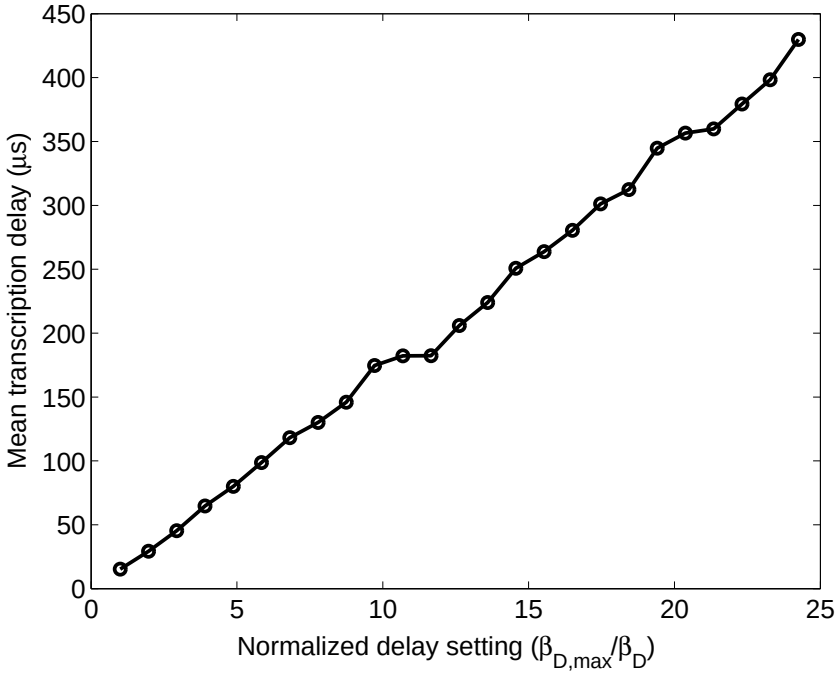


Figure 7-15: Measured transcription delay as a function of the programmable parameter β_D. Here $D_{DEL,max} = 2^4$ is the maximum possible value of β_D.

We also tested the SNR adjustment circuit shown in Figure 7-9. Equation (7.44) can be rewritten in the form

$$\mathrm{SNR_{mRNA}} = \beta_{snr} \left(\frac{2\tau_r I_{mRNA}}{q_{osc}} \right) \tag{7.53}$$

Thus, we expect the mRNA SNR to increase linearly with β_{snr}. The slope of this straight line should be given by $(2\tau_r/q_{osc})\, I_{mRNA}$. From (7.48) we see that the protein SNR should also increase linearly with β_{snr}. However, the slope is given

by $(\tau_r/q_{osc})\, I_{prn}/\gamma_n$, where γ_n is a dimensionless excess noise factor. Based on our previous analysis, we expect $\gamma_n = b$ when $\gamma_p \ll \gamma_r$. Also, we should get $\gamma_n = A_p$ in the opposite limit, i.e., when $\gamma_p \gg \gamma_r$.

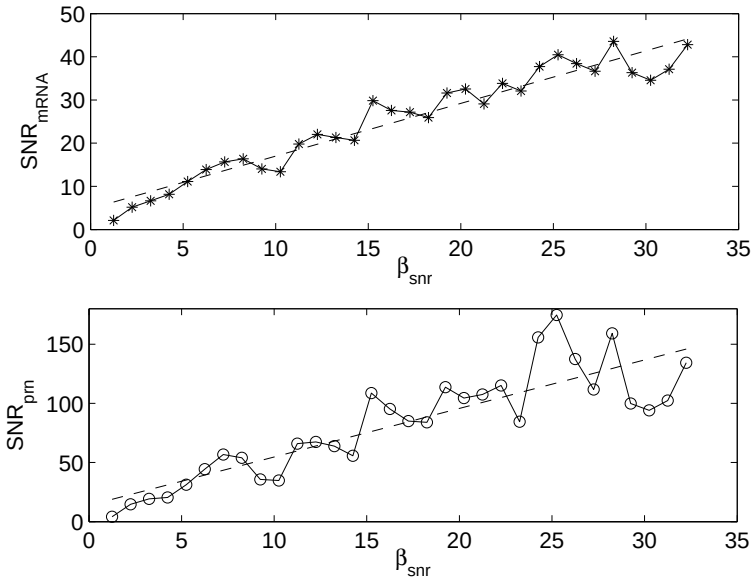


Figure 7-16: Measured SNR of mRNA and protein as a function of β_{snr}. The mean mRNA and protein concentrations were fixed at 40nA and 100nA, respectively.

Figure 7-16 shows the measured SNR of mRNA and protein as a function of β_{snr}. We used the following parameter values: $\gamma_p = \gamma_r$, $A_p = 1$ and $b = 1$. We see that both mRNA and protein SNRs increase linearly with β_{snr}, as expected. The best-fitting straight lines, also shown in the figure, were SNR $\approx 1.2\beta_{snr}$ for mRNA, and SNR $\approx 4.1\beta_{snr}$ for protein. Also, the mean values of mRNA and protein concentration were $\overline{I_{mRNA}} \approx 40$nA and $\overline{I_{prn}} \approx 100$nA, respectively. As a result, we get $2\tau_r I_{mRNA}/q_{osc} = 1.2 \Rightarrow q_{osc}/\,(2\tau_r) = 33$nA.

The plots in Figure 7-16 were obtained for $\gamma_p = \gamma_r$. In this case it can be showm that the excess noise in protein concentration is $\gamma_n = A_p/2$. Intuitively, the noise transfer function behaves as a *second*-order low-pass filter when the transcription and translation pole frequencies γ_p and γ_n coincide. The sharper high-frequency rolloff slope of this transfer function reduces the effective noise bandwidth by a factor of 2. As a result, γ_n is lowered by the same factor. Experimentally, the measured

slope of the SNR versus β_{snr} curve yields $\gamma_n = 0.74$, in reasonable agreement with the predicted value of $A_p/2 = 0.5$.

We know that the SNR of a standard Poisson variable N is equal to its mean value $\overline{N}$, i.e., the mean number of random events that occur during a counting (integration) period. Equation (7.53) shows that I_{mRNA} can be normalized into N_{mRNA}, a standard Poisson variable, where

$$\begin{aligned} N_{mRNA} &= \beta_{snr}\left(\frac{2\tau_r I_{mRNA}}{q_{osc}}\right) \\ SNR = \overline{N_{mRNA}} &= \beta_{snr}\left(\frac{2\tau_r \overline{I_{mRNA}}}{q_{osc}}\right) \end{aligned} \tag{7.54}$$

Here $N_{mRNA} = SNR\left(I_{mRNA}/\overline{I_{mRNA}}\right)$ is a normalized version of the mRNA concentration I_{mRNA}. Thus N_{mRNA} should obey the Poisson probability distribution, which is given by (6.76) and also rewritten below in terms of SNR:

$$p\left(N_{mRNA}\right) = \frac{SNR^{N_{mRNA}} \exp\left(-SNR\right)}{N_{mRNA}!} \tag{7.55}$$

Equation (7.55) predicts that the SNR of N_{mRNA} is the only parameter that affects its own probability distribution. Both β_{snr} and $\overline{I_{mRNA}}$ can control the SNR. We estimated probability distributions at various values of SNR by varying β_{snr} and measuring I_{mRNA} during a time interval of $T = 200\mu$s when transcription was active. Figure 7-17 shows the measured probability distributions. The x-axis has been normalized to $N_{mRNA}/\text{SNR} = N_{mRNA}/\overline{N_{mRNA}} = I_{mRNA}/\overline{I_{mRNA}}$ to facilitate comparison. The figure also shows the theoretically-predicted Poisson distributions, i.e., (7.55)[4]. The excellent match between measured and predicted probability distributions shows that our theoretical analysis is valid.

In Figure 7-16 we varied SNR by programming the value of β_{snr}, while keeping the mean concentration $\overline{I_{mRNA}}$ fixed. However, (7.53) shows that the SNR of mRNA

[4]We actually used [SNR] instead of SNR to draw these theoretical plots, since the Poisson distribution is only defined when both the number of counts and the mean count rate are integers. Here [ ] is the nearest-integer function.

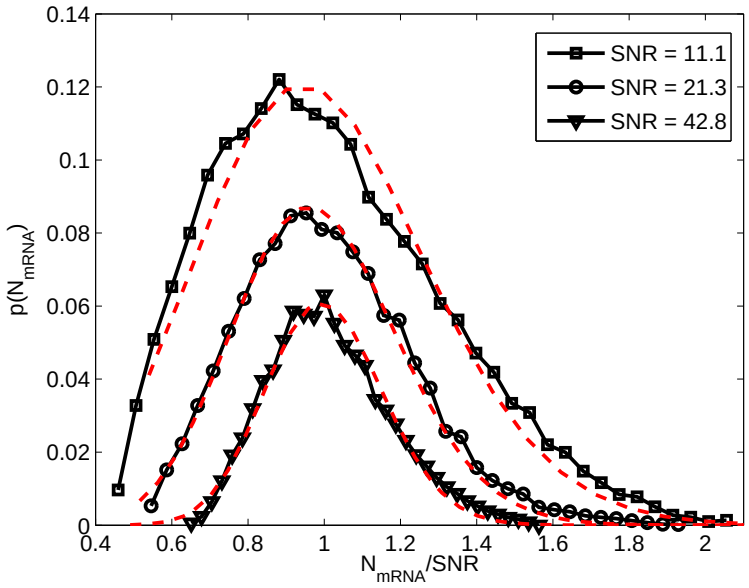


Figure 7-17: Measured probability distribution of the mRNA concentration I_{mRNA} at several values of SNR. The broken red lines shows the theoretically-predicted Poisson distributions.

is proportional to the product $\beta_{snr} I_{mRNA}$. Therefore it should increase linearly with $\overline{I_{mRNA}}$ when β_{snr} is fixed, just like any Poisson variable. The protein concentration should behave similarly, but with an excess noise factor γ_n. We verified this prediction experimentally by changing the input to the transcription filter, i.e., the activation strength I_{ACTV}.

The mean mRNA and protein concentrations are given by $\overline{I_{mRNA}} = A_r I_{ACTV}$ and $\overline{I_{prn}} = A_r A_p I_{ACTV}$ respectively, where $A_r = k_r/\gamma_r$ and $A_p = k_p/\gamma_p$. Thus, the SNR of both mRNA and protein should be proportional to I_{ACTV}. We varied I_{ACTV} over a 32:1 range by programming $\beta_s = I_{ACTV}/I_0$ between 1/4 and 8, while maintaining a fixed value of $\beta_{snr} = 10$. We also kept the following parameter values fixed: $\gamma_p = \gamma_r$, $A_p = 2$ and $b = 2$. The results are shown in Figure 7-18. We see that the SNR increases linearly with mean concentration in both cases, as expected.

The best-fitting straight lines, also shown in Figure 7-18, were SNR $\approx 0.53\overline{I_{mRNA}}/\text{nA}$ for mRNA, and SNR $\approx 0.65\overline{I_{mRNA}}/\text{nA}$ for protein. As a result, we get $q_{osc}/(2\tau_r) =$ 19nA. Our earlier estimate of this quantity was based on Figure 7-16, and was equal to 33nA. The estimates differ because $\tau_r = \tau_0/\beta_A$ was different in the two cases. Here

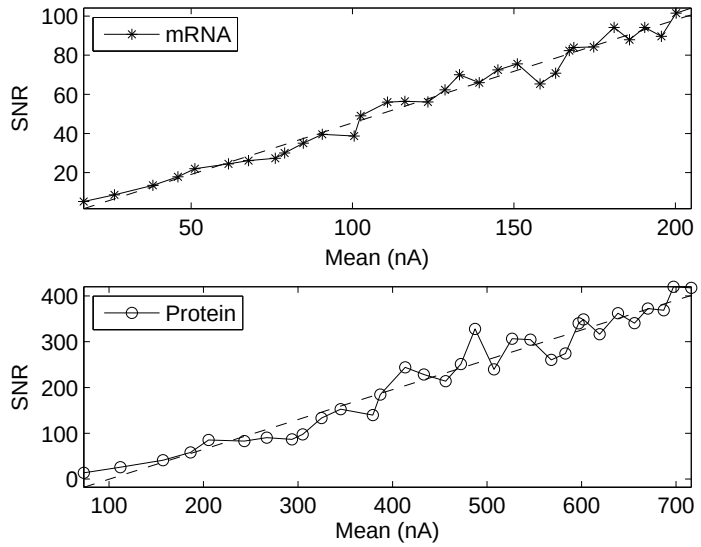


Figure 7-18: Measured SNR as a function of mean concentration for both mRNA and protein. Mean concentrations were set by varying I_{ACTV} while keeping all transcription and translation parameters fixed.

$\beta_A = I_A/I_0$ is set by the user, and $\tau_0 = C\left(\phi_T/\kappa\right)/I_0$. Finally, the measured value of protein excess noise factor was $\gamma_n = 0.81$, which is close to the predicted value of $A_p/2 = 1$.

A possible reason for differences between predicted and measured values of γ_n is static offset between the output buffers used to measure V_{mRNA} and V_{prn}, the log-compressed versions of I_{mRNA} and I_{prn}, respectively. Offset causes errors in the measured values of $\overline{I_{mRNA}}$ and $\overline{I_{prn}}$, but not in SNR. The result is errors in the slopes of the SNR curves. It is easy to show that an offset of ΔV_{off} causes a fractional error of $\Delta V_{off}/\left(\phi_T/\kappa\right)$ in the ratio of the two slopes, i.e., the measured value of γ_n.

Gene regulation networks from a wide variety of organisms have been found to contain specific sub-networks in much greater abundance than expected by chance. Such sub-networks, which are repeatedly used by cells, are known as *network motifs*. The widespread occurrence of a small set of common motifs implies that transcription networks are highly modular. Motifs are of great scientific interest because they act as reusable modules, i.e., carry out functions that are generic enough to be useful to cells in a variety of environments. The simplest motifs, namely those that contain

two or three genes, have been extensively studied [4]. We tested our chip by building some well-known network motifs with it.

We first considered the coherent feed-forward loop (FFL) motif described in [4], and shown in Figure 7-19(a). FFLs act as sign-sensitive delay elements and are common in cell signaling networks. In the coherent FFL shown in the figure the protein transcribed from gene X activates gene Y, while gene Z is activated only when both the protein products of genes X and Y are present. In other words Z has a two-input, AND-like input function. Induction of X by a pulse of the inducer S_X turns gene Z on after some delay. The delay is a result of the fact that *both* X and Y have to be higher than their threshold values for the output of the AND gate to go high. However, Z turns off immediately when the pulse of S_X is removed, because the output goes low when *either* X or Y go below their threshold values.

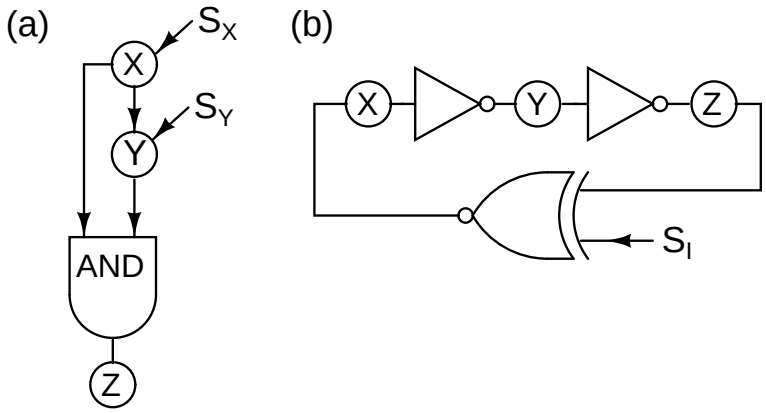


Figure 7-19: Two simple motifs found in transcription networks, (a) the coherent feed-forward loop, and (b) a ring oscillator that can be inhibited by the signal S_I.

Another interesting property of the FFL is that it rejects short input pulses, i.e., pulses of S_X that are shorter than the turn-on delay do not activate X. It has been suggested that cells use this property to distinguish between long chemical pulses, which might be signals from other cells, and shorter bursts, which might be due to noise. Figure 7-20 shows measured transient responses of our coherent FFL circuit to a aquare pulse of inducer S_X. The figure shows concentrations of the proteins transcribed from genes X, Y and Z as functions of time for inducer pulses of two different durations. We see that the circuit behaves as expected. It distinguishes

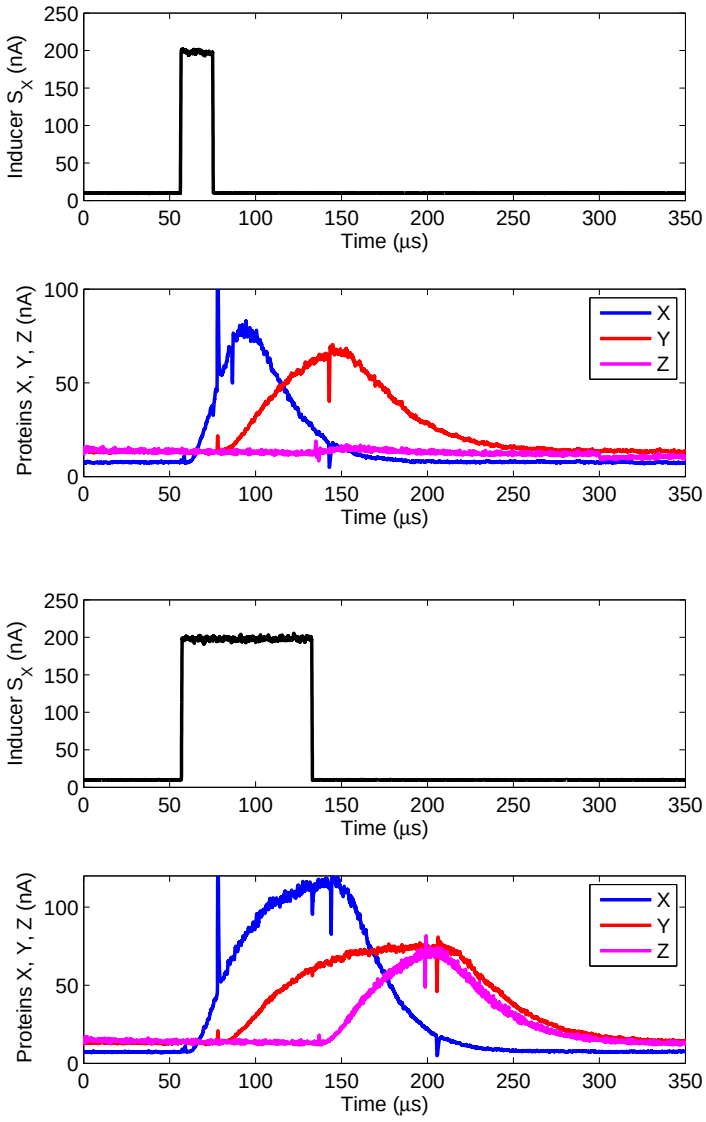


Figure 7-20: Measured transient response of the coherent FFL circuit to a pulse of inducer concentration. Plots are shown for a brief pulse (top) and a longer pulse (bottom).

between short pulses (which do not activate Z) and long pulses (which do activate Z). We also note the presence of sign-sensitive delay: the activation of Z is delayed at rising, but not falling, edges of inducer pulses.

The measurements in Figure 7-20 were made in the noiseless case, i.e., with the SNR adjustment circuits of all genes disabled. These circuits are disabled by setting $EN = 0$ within the transcription filters, as shown in Figure 7-11. We also simulated the FFL in the presence of noise in mRNA concentrations, i.e., with the SNR adjustment circuits turned on. We noticed that random fluctuations in X can turn on Y and Z in the absence of S_X when the SNR is set to low values. Similar behavior has been observed in living cells.

The second network motif that we built using our chip was a ring oscillator. This system was first synthesized in living *E. coli* cells by Elowitz and Leibler in 2000, and named the *reprissilator* [63]. It contained three genes, which we denote by X, Y and Z. The protein produced by gene X represses gene Y. Similarly, Y represses Z, and Z in turn represses X. Thus each gene can be modeled as a repressor, or signal inverter. The three inverters within the motif are arranged in a feedback loop.

We replaced one of the repressors, X in this case, by a gene with a two-input XNOR input function. The other input of the XNOR was connected to an external signal S_I, as shown in Figure 7-19(b). In other words, X is activated when S_I and Z are both active, or when they are both inactive. Thus the signal S_I acts as an inhibitor. When S_I is inactive (low) the XNOR acts as an inverter, and the system behaves like the original reprissilator. When S_I is active (high) the XNOR acts as a non-inverting buffer, changing the sign of the feedback loop and making the system act as a bistable element, or latch.

Figure 7-21 shows measured waveforms (protein concentrations) generated by the ring oscillator circuit. We see that the circuit exhibits stable, undamped oscillations with a period of approximately 700μs when S_I has a low value. In addition, the oscillations are inhibited when the concentration of S_I is increased, as expected. The oscillation period, as well as the shapes of individual waveforms within the oscillator, can be tuned by adjusting the parameters of the genes X, Y and Z. In particular,

oscillations only occur when the overall feedback loop gain is greater than unity. Loop gain increases when the three genes have similar 'on' and 'off' protein concentrations and activation thresholds, and decreases when the genes are very different from each other. In the latter case one or more genes will be saturated (permanently stuck in 'on' or 'off' positions) and provide no signal gain.

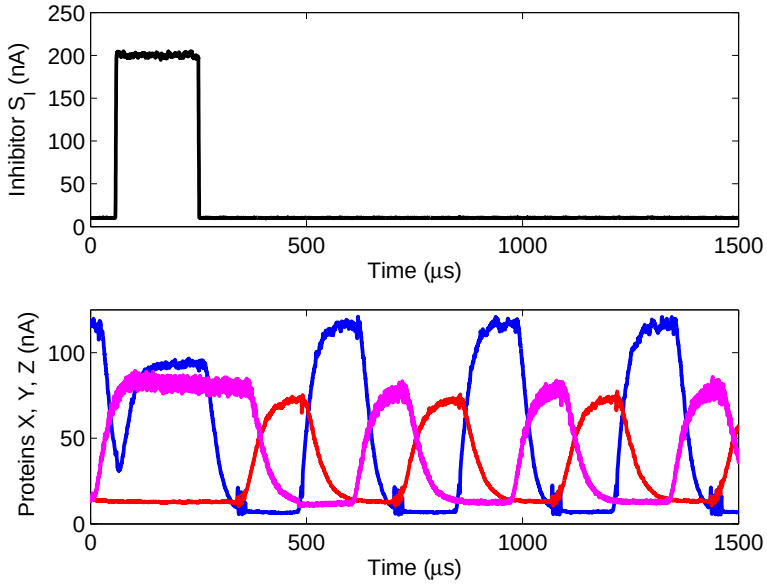


Figure 7-21: Measured waveforms (protein concentrations) generated by the ring oscillator circuit. Waveforms are shown for two different concentrations of the inhibitor S_I.

7.4 Appendix A: Software and Data Acquisition

The software required to test the gene network chip was similar to that required for the chemical network chip. The program, named *gene_ui*, contains a hardware driver and a simple user interface. The driver provides an API with similar functionality as the *protein_ui* API described in the previous chapter. The help file of the program, which includes names and brief descriptions of the functions within the API, is reproduced below for reference.

- *gene_ui* help file (version 0.3, 03/13/09)

```
--------------------------------------
1. Utility functions:
help: display this help file
bye/done/exit: finished, exit to MATLAB
--------------------------------------
2. Write functions:
chip_reset: delete all genes
add_gene: add a new gene
modify_gene: modify properties of a gene
delete_gene: delete an existing gene
set_input: input constant current
--------------------------------------
3. List functions:
view_gene: view properties of a gene
list_genes: list status of all genes
--------------------------------------
4. Read functions:
set_scope: change oscilloscope settings
simulate: input arbitrary waveform, grab data
simulate_repeat: simulate several times
param_sweep: run simulations over a range of parameter values
plot_data: plot data from past simulations
--------------------------------------
```

Figure 7-22 shows the experimental setup used for testing both the gene and protein network chips. The chips can be interfaced simultaneously to the same computer. They both accept inputs that are currents or log-compressed voltages, and generate buffered, log-compressed voltage outputs. Input and output voltages are always referenced to a common ground terminal. As a result, the two chips can easily communicate with each other: one chip's outputs, in the form of protein or mRNA

concentrations, can be directly hooked up to the other's inputs.

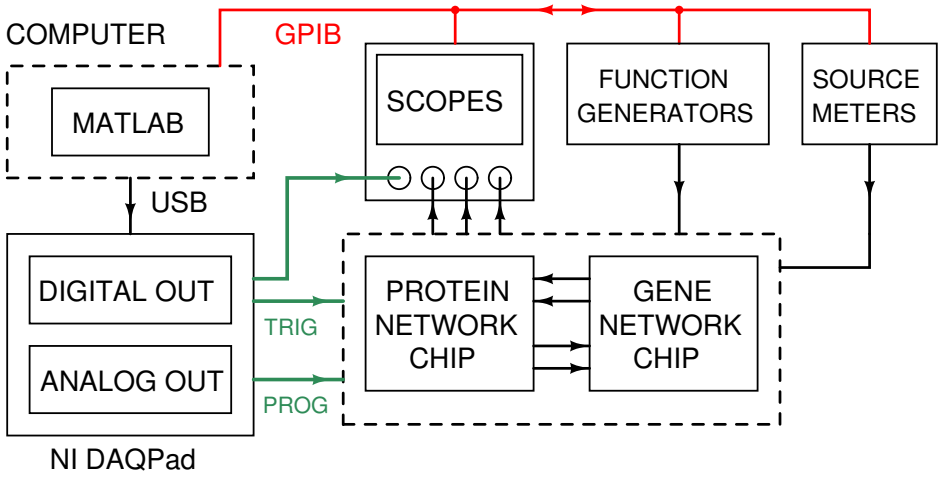


Figure 7-22: Experimental setup used for testing the gene and protein network chips.

The analog outputs of a National Instrument DAQPad, hooked up to the computer's USB port, were used to generate the CLK and DATA signals required to program each chip. The digital outputs of the DAQpad were used to generate ENABLE and BEGIN signals. The ENABLE signals allow bits presented to the chip on the DATA pin to be loaded into internal memory. BEGIN signals initiate computations on the chip, and also trigger function generators and oscilloscopes. Function generators produce arbitrary time-varying chip inputs, while the oscilloscopes record chip outputs. Keithley source meters are used to generate chip inputs that don't need to vary during a given simulation. All these instruments are interfaced to the computer via a common GPIB bus. The MATLAB programs *protein_ui* and *gene_ui* handle data transfer on the USB and GPIB buses, provide the user interface and allow further processing of data acquired from the chips.

Oscilloscopes were used for data acquisition because no other equipment could sample the chip outputs fast enough. However, oscilloscopes have limited sampling resolution, which results in large amounts of quantization noise. For example, our Tektronix TDS3014 oscilloscopes had 9 bits of resolution, i.e., quantized the signal to one of $2^9 = 512$ levels. The full-scale voltage of the oscilloscope's ADC is 10 times the vertical scale V_Y. For example, it is ± 0.25V when $V_Y = 50$mV/division. Thus

the quantization step size is given by

$$\Delta V_Q = \frac{10 V_Y}{2^9} \approx \frac{V_Y}{50} \tag{7.56}$$

We make the usual approximation that the quantization noise is uniformly distributed between $\pm \Delta V_Q/2$. It is easy to show that in this case the variance is given by $\sigma_Q^2 = (\Delta V_Q)^2/12$. The total variance of the detected voltage is given by $\sigma_{OUT}^2 = \sigma_S^2 + \sigma_Q^2$, where σ_S is the standard deviation of the original voltage being sensed. In order to make accurate measurements the quantization noise should be neglible compared to σ_S, i.e., we need $\sigma_Q \ll \sigma_S$. Other sources of noise, such as electromagnetic pickup, may also be important.

Equation (7.56) shows that we can reduce σ_Q by using a smaller vertical scale V_Y. Unfortunately this is not an option when the signal has a large DC component, because we would exceed the linear range of the ADC and saturate. The outputs from our chips are of this form. We cannot use reject the DC component by AC coupling, because the value of this component is also of interest. Instead, we used the oscilloscope's 'offset' feature to subtract a known, fixed voltage from each channel. We adjust this offset voltage until most of the DC component has been canceled, which allows us to decrease V_Y and σ_Q. The subtracted voltage is eventually added back using software. Finally, the detected voltage V_{OUT} is exponentiated in software to generate the actual output of the chip, namely the current I_{OUT}. Thus we perform the operation $\exp(V_{OUT}/V_L)$, where $V_L = \phi_T/\kappa$ is the linear range. The SNR of I_{OUT} is then given by

$$\mathrm{SNR}_{OUT} = \left(\frac{V_L}{\sigma_{OUT}}\right)^2 \tag{7.57}$$

The value of σ_Q can be reduced by acquiring the same output several times and averaging the results in the time-domain, i.e., sample by sample. This technique is known as *synchronous averaging*. Averaging N traces decreases the standard deviation of a stationary signal by a factor of $1/\sqrt{N}$, as one might expect. However, this technique cannot distinguish between noise due to quantization and due to the chip.

One can also run a moving-average or other low-pass filter on a single oscilloscope trace to reduce its noise level. This technique can preferentially renove quantization noise if it has wider bandwidth than the signal itself, which is usually the case. Figure 7-23 shows measured values of σ_{OUT} of one of the output voltages from the protein network chip as a function of V_Y. We see that σ_{OUT} increases with V_Y, indicating the presence of quantization noise. The figure also shows the results of low-pass filtering the waveform with moving-average filters of various lengths.

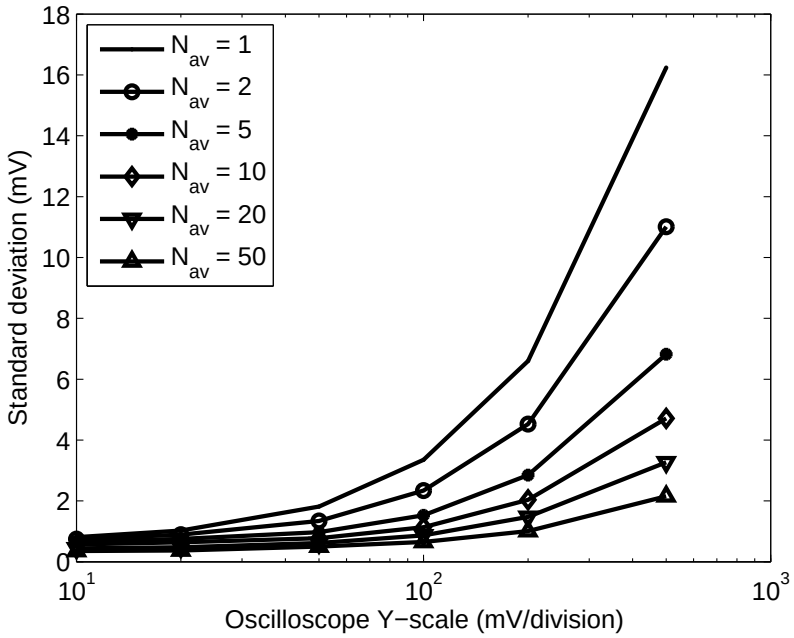


Figure 7-23: Measured standard deviation σ_{OUT} of one of the output voltages of the protein network chip after sampling and acquisition by one of our oscilloscopes at various vertical gain settings. The data was later averaged by a moving-average filter of length N_{av} to reduce quantization noise.

We also tried various other techniques for reducing the amount of noise. These techniques included capacitively bypassing all bias nodes, separating the core and I/O power supplies used by the chip, running the chip off batteries, and reducing the input bandwidth of the oscilloscope channels to 20MHz. All our techniques helped to some extent. Lowering the oscilloscope bandwidth was particularly effective, reducing the displayed noise level by approximately a factor of 6. All our efforts, however, only managed to reduce σ_{OUT} to approximately 0.8mV, which limits SNR_{OUT} to 33dB

without averaging. As a result, we were unable to accurately measure the actual SNR of the chemical simulator chip. However, the SNR of the gene simulator chip, with the SNR adjustment circuits enabled, was low enough to be easily measurable.

7.5 Appendix B: Linear-Feedback Shift Registers

Linear-feedback shift registers (LFSRs) have been extensively used in communication systems. They have also been used to generate pseudo-random noise for hybrid and digital simulations of random processes [143]. LFSRs are synchronous circuits, i.e., they only change states at discrete intervals determined by a clock. There are two basic classes of LFSRs, Fibonacci and Galois, which are mathematically equivalent and produce the same sequences. In either case the next state of a shift register of length r is formed by feeding back a linear function of its current state. In modulo-2 arithmetic[5] a function is considered 'linear' if it consists only of modulo-2 additions, i.e., XOR or XNOR operations. Thus the LFSR feedback function is formed by XOR-ing the states of two or more stages. In Fibonacci, or external LFSRs the feedback function has a single output that feeds the input, or first stage of the shift register. In Galois, or internal LFSRs the feedback function has multiple outputs which feed both the first and one or more intermediate stages. We will analyze Fibonacci-type LFSRs, since we implemented one on-chip.

Mathematically, an LFSR is described by its *generator polynomial*, defined as

$$P(x) = 1 + \sum_{i=1}^{r} c_i x^r \tag{7.58}$$

where the input is defined as state 0, and x^i is the i-th stage after the input. Also, $c_i = 1$ if the i-th stage is part of the feedback function, and 0 if it is not. The output of any LFSR stage as a function of time is known as the *sequence* generated by it. Sequences from different stages are simply time-shifted versions of each other. Each stage in the shift register has 2 possible states (0 or 1), and so the LFSR as a whole

[5] Also known as GF(2), the finite, or Galois field of the two elements 0 and 1.

has 2^r states. Since this is a finite number the system is necessarily periodic, with a maximum sequence length N of 2^r. It is remarkable that feedback functions can be found that result in sequences nearly as long, i.e., $N = 2^r - 1$. Such sequences are known as Maximum-length (ML) sequences. ML sequences are useful for generating psedo-random waveforms. In particular, they possess three key "pseudo-randomness" properties:

1. **Balance**: In each period the shift-register assumes every one of its 2^r states, except for the all-zero state[6]. The sequence is nearly balanced: it contains 2^{r-1} 1's and $(2^{r-1} - 1)$ 0's.

2. **Run length**: The probability of a run, or continuous series, of 1's or 0's decreases exponentially with the length of the run. The probability a run of length n, consisting either of 0's or 1's, is $P_n = 1/2^{n+2}$ [187]. Run length is analogous to the time between successive events. konwn as the inter-arrival time, of a continuous time random process. Thus, the inter-arrival time distribution of an ML sequence is similar to that of a Poisson process.

3. **Auto-correlation**: The periodic, or discrete-time auto-correlation function of an ML-sequence is noise-like and only takes two values:

$$R_n(k) = \begin{cases} 1 & \text{if } k = iN \\ -1/N & \text{otherwise} \end{cases} \tag{7.59}$$

   where i is an integer and $N = 2^r - 1$ is the length of the sequence.

We would like to know how to pick the set of coefficients $\{c_i\}$ in order to generate ML sequences. Clearly $c_r = 1$, i.e. the coefficient of the highest term must be non-zero, because otherwise succeeding stages will not be part of the feedback loop, shortening the effective length of the shift register. There are only two other conditions that $P(x)$ must satisfy to produce ML sequences:

- $P(x)$ cannot be factored into lower-order polynomials.

[6]This state must be avoided during startup, since it is stable.

- $P(x)$ is a factor of $x^N + 1$, where $N = 2^r - 1$ is the length of the sequence.

Polynomials which satisfy these properties are known as *primitive polynomials*. There are many primitive polynomials for a given value of r, but they all produce time-shifted versions of the same ML sequence.

LFSR sequences are typically mapped to real signals $x(t)$ by assigning voltages $\pm A$ to states 1 and 0, or vice-versa. State transitions occur at integer multiples of the clock period T_{clk}, and the sequence is periodic with a period NT_{clk}. Periodicity results in LFSR sequences having densely-packed line spectra, where individual spectral lines are separated by $\Delta f = 1/(NT_{clk})$. Individual states look like rectangular pulses in the time domain when transitions between them are rapid compared to T_{clk}. In this case the lines in the power spectrum have a sinc function envelope given by [143]:

$$S_{xx}(\omega) = A^2 T_{clk} \text{sinc}^2 (\omega T_{clk}/2) \tag{7.60}$$

The power spectral envelope shown in (7.60) is approximately flat (to within 0.1dB) for $\omega < 1/(2T_{clk})$. On timescales shorter than the period NT_{clk} the individual spectral lines cannot be distinguished, and the spectrum appears continuous. Thus, over the frequency range $1/(NT_{clk}) < \omega < 1/(2T_{clk})$ the spectrum resembles that of white noise. The continuous-time autocorrelation function $R_{xx}(\tau)$ of an ML sequence is also periodic. It consists of two parts. Firstly, correlations within the same state result in triangular pulses of peak height A^2 and width $\pm T_{clk}$ centered around $\tau = iNT_{clk}$, where i is an integer. For values of τ outside these regions $R_{xx}(\tau) = A^2/N \approx 0$ for large values of r. Thus, as r increases $R_{xx}(\tau)$ looks more and more noise-like.

We used $r = 23$-bit long shift registers with the generator polynomials $x^{23} + x^{18} + 1$ and $x^{23} + x^{14} + 1$, both of which produce maximum-length sequences, i.e., sequences of length $N = 2^{23} - 1 = 6.39 \times 10^6$. In addition, at a typical value of $I_{mRNA} = 10nA$ and with $\beta_{snr} = 1$ we get $f_{CCO} = \beta_{snr} f_0 = 6.8$kHz, resulting in a repetition period of 196.3 seconds. Thus, the LFSR sequence will not repeat on the time scale of our simulations (typically no longer than 1ms), making its spectrum look continuous.

Finally, it is interesting to compare ML sequences to the random telegraph (RT)

signals they seek to emulate. RT signals have two states, again denoted by $\pm A$. State transitions are uncorrelated and can be described by a Poisson process with mean rate λ. The power spectra and autocorrelation functions are given by

$$\begin{aligned} S_{xx}(\omega) &= 2A^2 \left(\frac{T}{1 + (\omega T)^2} \right) \\ R_{xx}(\tau) &= A^2 e^{-|\tau|/T} \end{aligned} \tag{7.61}$$

where $T \equiv 1/\lambda$. Thus, the power spectrum of the RT signal is a Lorentzian, and the autocorrelation function decays exponentially with τ. The corresponding functions for ML sequences look similar, being sinc^2 and triangular, respectively. In both cases $S_{xx}(\omega)$ looks flat at low frequencies, with values that are equal to each other if

$$T_{clk} = 2T \Rightarrow f_{clk} = \lambda/2 \tag{7.62}$$

Thus, an LFSR sequence with clock frequency f_{clk} has statistical properties that are similar to an RT signal with mean event rate $\lambda = 2f_{clk}$. Intuitively, every state transition in the RT signal constitutes an event, while only active clock edges (rising or falling), can change the state of the LFSR. As a result, the RT signal has double the event rate.

Chapter 8

Conclusion

Behold, I am insignificant; what can I reply to You?
I lay my hand on my mouth.
Once I have spoken, and I will not answer;
Even twice, and I will add nothing more.
– Book of Job, Chapter 40, The Bible (New American Standard Version)

In this chapter, we summarize the work described in this thesis, and suggest directions for future research.

8.1 Summary

The shortest possible summary of our work is provided by Figure 8-1, which shows all the printed circuit boards that were used to test the integrated circuits described in this thesis. A slightly more detailed summary is provided below.

8.1.1 Chapter 1: Introduction

We began this thesis by placing the modeling and simulation of dynamical systems within a historical context. We went on to review the physics and theory of computation. We paid particular attention to computations performed by analog dynamical

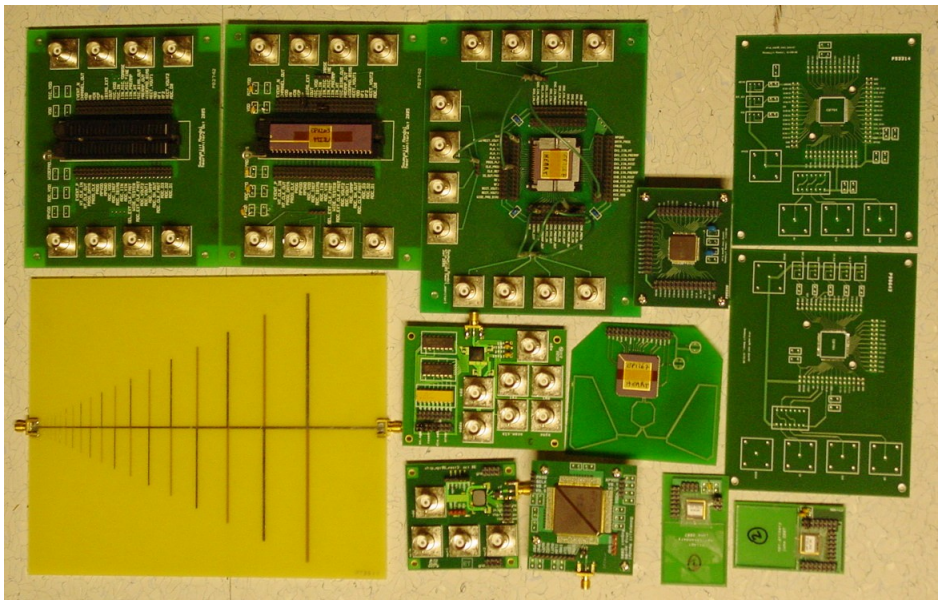


Figure 8-1: The printed circuit boards used to test the integrated circuits described in this thesis.

systems, such as partial differential equations. Finally, we provided a brief overview of other chapters and appendices in the thesis.

8.1.2 Chapter 2: The Bidirectional RF Cochlea

In this chapter we described the design and on-chip implementation of a bidirectional transmission-line model of the biological cochlea at radio frequencies (RF). We began by describing the biological cochlea, and summarizing its basic characteristics. Functionally the cochlea performs real-time, broadband spectrum analysis, which is of importance in many scientific and commercial applications. In fact, we concluded, after studying several common spectrum analyzer architectures, that the cochlear algorithm provides the best trade-off between analysis time and hardware cost.

We began the design of an integrated spectrum analyzer based on the cochlear algorithm by developing a simple rational model for the impedance of the basilar membrane. We then described a network synthesis procedure that allowed us to implement this impedance using two coupled RLC resonators. Finally, we performed a theoretical analysis of the frequency resolution, sensitivity and dynamic range of our spectrum analyzer.

We concluded this chapter by describing the circuits that were used to implement the bidirectional RF cochlea in a 0.13μm CMOS technology, and presenting experimental results. Developing an automated design flow for optimized on-chip inductors and transformers was important for the overall success of the design. We demonstrated cochlear operation between 1GHz and 8GHz with 50 output channels (frequency bins) and a dynamic range of 70dB for single input tones.

8.1.3 Chapter 3: The Unidirectional RF Cochlea

We began this chapter by describing a formal prodcedure for approximating the behavior of the bidirectional cochlear model described in the previous chapter by a cascade of exponentially-scaled, unidirectional filters. We performed a theoretical analysis of the frequency resolution, sensitivity and dynamic range of a spectrum analyzer implemented using these novel filters. We found that our procedure yields filter transfer functions that generate cochlear outputs with improved frequency selectivity and lower group delay.

In the next part of this chapter we described a network synthesis procedure that allowed us to implement our improved cochlear filters using two coupled RLC resonators and a single transistor. We also described other circuits that were needed to implement our improved unidirectional RF cochlea in a 0.13μm CMOS technology. For example, we designed feedback loops for automatically calibrating the peak and low-frequency gains of the cochlear transfer functions. Finally, we described experimental results, which demonstrate cochlear operation between 600MHz and 6GHz with 51 output channels (frequency bins) and a dynamic range of 70dB for single input tones.

8.1.4 Chapter 4: The RF Cochlea: Enhancements and Applications

In the first part of this chapter we described the design and implementation of a broadband log-periodic dipole antenna that can be interfaced to our RF cochlea chips. In

the next part we briefly considered strategies that enhance the performance of the RF cochlea by exploiting timing information present in its outputs. In particular, we showed that correlations between outputs can be used for nonlinear spectral sharpening, which increases the frequency resolution without degrading timing precision.

In the final part of this chapter we described interesting analogies between frequency estimators and analog-to-digital converters. These analogies yield several novel classes of frequency estimators. For example, we developed a successive approximation architecture that used cascaded RF cochleas and a decision network, or state machine, to estimate multiple input frequencies in parallel. We also showed simulation results, and discussed the use of our architectures within ultra-wideband, cognitive, or universal radio receivers.

8.1.5 Chapter 5: A Detailed Discussion of Noise

In this chapter we took a detailed look at the physical basis of noise, which limits the precision of all practical computations. We began by considering a lumped resistor, and went on to consider diodes and MOSFETs in detail. Next, we reviewed the fundamental fluctuation-dissipation theorem that links fluctuations in thermal equilibrium to disspation (loss) outside equilibrium. We concluded that thermal noise formulas can always be derived as special cases of shot noise formulas by applying a version of the fluctuation-dissipation theorem.

In the next part of this chapter we reviewed the basics of noise in quantum systems, and found that they possess complex autocorrelation functions and consequently, nonsymmetric power spectra. We went on to describe $1/f$, or flicker noise in MOSFETs in some detail. Experimental measurements showed that the power spectrum of an integrated NMOS transistor retained a $1/f$ dependence down to at least 0.5μHz. We concluded the chapter by discussing noise in several other systems: antennas, oscillators, chemical reactions, and biology.

8.1.6 Chapter 6: Circuit Models of Chemical Reactions

In this chapter we described scalable integrated circuit models of chemical reaction networks. We began by discovering detailed similarities between fluxes in chemical reactions and the flow of electrons in transistors. These similarities allowed us to develop simple translinear circuits that were dynamically equivalent to mass-action chemical kinetics. We then analyzed the effects of device mismatches and noise on these circuits. We concluded that mismatches created unwanted extra reactions, and must be minimized. However, we found that our circuits had noise properties similar to chemistry, allowing them to be used for stochastic simulations.

In the next part of the chapter we described a 0.18μm CMOS chip designed for parallel simulations of chemical reaction networks. The topology and parameter values of the simulated network are completely programmable by the user through custom driver software. The chip contained 32 state variables, 160 reaction circuits, accepted 8 inputs and generated 16 outputs. We concluded by presenting experimental results for simple reaction networks, and demonstrated a 30x speedup in simulation time over a quad-core desktop computer.

8.1.7 Chapter 7: Circuit Models of Genetic Networks

In this chapter we described scalable integrated circuit models of gene regulation networks. We bagan by developing a simplified model for induction, transcription factor binding, activation, transcription and translation. We then described circuits that implement the model, including programmable logic for multi-dimensional activation functions and translinear filters for transcription and translation. We also described a circuit that can adjust the SNR of mRNA or protein concentrations to any value, allowing us to perform stochastic simulations.

In the next part of the chapter we described a 0.18μm CMOS chip designed for parallel simulation of gene regulation networks. The topology and parameter values of the simulated network are completely programmable by the user through custom driver software. The chip accepted 16 inputs, and contained 6 independent genes with

8-dimensional input functions. We concluded by presenting experimental results for single genes and simple gene circuits, such as a coherent FFL and a ring oscillator.

8.2 Future Work

8.2.1 The RF Cochlea

Circuit Enhancements

The bidirectional RF cochlea chip had a substantial amount of low-frequency line loss. Reducing this loss constitutes an important topic for future research. We would also like to predict the effects of nonlinearities, particularly in the active negative resistors, on wave propagation at large amplitudes. The large body of existing research on nonlinear transmission lines [241, 245] may be of assistance in this regard. Other useful improvements could include independent gain calibration circuits for each stage, a larger chip area (which would allow to increase the number of stages), more sensitive detectors (such as coherent detectors), and on-chip digitzation of the outputs. We would also like explore the use of adaptive gain control for improving dynamic range, including schemes for automatically notching out large interferers.

Most of the same improvements that were described in the previous paragraph also apply to the unidirectional cochlea chip. In addition, we would like to modify the low-frequency gain calibration loop so that it can both increase and decrease gain, and implement a more sophisticated peak gain control circuit for each stage.

Larger Systems

We would like to build novel 'universal' receivers that detect multiple RF bands in parallel by using the RF cochlea as a building block. In particular, we would like to implement the cascaded cochleas architecture, which allows us to adaptively 'zoom' into spectral features of interest. We would also like to explore the use of temporal, or phase, information present in the cochlear outputs for rejecting noise, detecting transients, and improving the frequency resolution of such receivers. The biological

cochlear nucleus contains several types of cells that perform analogous functions, and may prove to be a fruitful source of inspiration.

8.2.2 Circuit Models of Chemical and Genetic Networks

Circuit Enhancements

There are many ways in which the prototype chemical network simulation chip could be improved. In order to reduce device mismatches, which generate unwanted extra reactions, we can lay out reversible reactions as single blocks that are laid out to match. Currently such reactions are produced by connecting two different reaction circuits, which can be physically far apart, to the same set of state variables. We can also develop a scheme for auto-zeroing out the mismatches before each simulation run. The output buffers can also be auto-zeroed to remove their offsets. The use of fully-differential circuits will reduce mismatches and, potentially, the effects of leakage currents, and should also be explored.

The ratio r, defined as the the total number of reactions to the total number of state variables, appears to be between 2 and 3 for many reaction systems within cells. The value of r in the current chemical simulator chip was approximately 5, which often led to state variables being in short supply while implementing complex models. We plan to decrease r to approximately 2.5 in future iterations of our chips in order to solve this problem.

Similar design improvements can also be made to the genetic network simulation chip. In addition. we would like to explore the use of non-abrupt activation functions (Hill functions of low order), because they are more biologically realistic. We would like to add an SNR adjustment circuit for the protein concentration, as well as modifying the circuit itself, as described earlier. Stochastic switching of the activation function may be important for modeling eukaryotic genes, and can be modeled with a variant of the same circuit. Such circuits can be used to adjust the noise of any chemical species, and may be included in future versions of the chemical network chip as well. Finally, both chips had some simple design errors, such as an initial condition

setting bug in the chemical simulator and a faulty truth table for the PLA in the gene simulator, which will be fixed in future design iterations.

Larger Systems

We would like to use improved versions of our chemical and genetic network simulation chips as the basis for a high-performance co-processing system that can be used by biologists. The two chips will be combined into a single hybrid system, as shown in Figure 8-2. The combined system will allow the two parts to interact closely with each other. Thus, the chemical simulator will accept translated proteins as inputs, while its outputs will be fed into the genetic simulator, where they can act as transcription factors.

We would like to interface our co-processing system to standard desktop computers by implementing a universal serial bus (USB) interface. In addition to our custom chips and the USB transceiver, the system will also contain analog-to-digital converters, flash memory and a micro-controller. The micro-controller will act as the interface between our chips and the computer. It will handle all low-level communications with our chips, including programming and data acquisition. For example, it will allow the user to perform multiple simulation runs, store the results in the on-board flash memory, perform simple analyses, and extract meaningful information, *without transferring large amounts of raw data to the computer*. Thus, the microcontroller will alleviate the input/output bandwidth bottleneck that plagues high-performance computing.

Random Number Generation

Finally, we would like to explore techniques for generating random numbers in hardware. Such numbers are important for a variety of applications where the lower power and/or higher speed of specialized hardware would prove beneficial. The statistical properties of the numbers should be reliable and easily controlled. We have already developed some techniques for this purpose. For example, an on-chip transcription delay with controllable signal-to-noise ratio (SNR) was produced by using an oscilla-

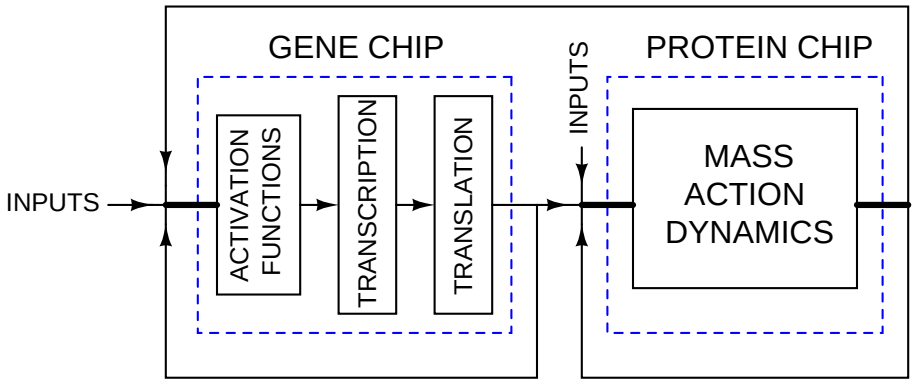


Figure 8-2: A conceptual view of how the chemical reaction network and gene network chips can be combined to create a single hybrid dynamical system.

tor to sample the rising edge of an activation signal, and a shift register to delay it[1]. We would like to extend these techniques into the development of application-specific random number generation circuits. Such circuits will specialize in generating random numbers for specific purposes. For instance, the randomness requirements for cryptography are much more stringent than for stochastic simulations, and demand more sophisticated circuits.

[1]Conceptually, this technique is related to oscillator sampling, which is a well-known random number generation method [225]. In this method a freely-running oscillator is used to sample the output of another freely-running oscillator.

Appendix A

Power-Efficient Impedance-Modulation Wireless Data Links for Biomedical Implants

In this chapter we analyze the performance of wireless data telemetry links for implanted biomedical systems. An experimental realization of a bidirectional half-duplex link that uses near-field inductive coupling between the implanted system and an external transceiver is described. Our system minimizes power consumption in the implanted system by using impedance modulation to transmit high-bandwidth information in the uplink direction, i.e. from the implanted to the external system. We measured a data rate of 2.8Mbps at a bit error rate (BER) of $< 10^{-6}$ (we could not measure error rates below 10^{-6}) and a data rate of 4.0Mbps at a BER of 10^{-3}. Experimental results also demonstrate data transfer rates up to 300kbps in the opposite, i.e. downlink direction. We also perform a theoretical analysis of the bit error rate performance. An important effect regarding the asymmetry of rising and falling edges that is inherent to impedance modulation is predicted by theory and confirmed by experiment. The link dissipates 2.5mW in the external system and only 100μW in

the implanted system, making it among the most power-efficient inductive data links reported. Our link is compatible with FCC regulations on radiated emissions.

A.1 Introduction

Inductively-coupled near-field wireless links have been extensively used in various implanted medical devices [81, 89, 121, 165, 205, 278, 288]. Some far-field links have also been reported [109]. Many of these links have been used to transmit power to the implanted device in addition to carrying unidirectional or bidirectional data signals. The system described in this paper was designed for use in an implanted neural prosthesis with multiple recording electrodes [262]. The prosthesis requires a wireless transcutaneous link to transmit data between the implanted sub-system and an external unit (see Figure A-1).

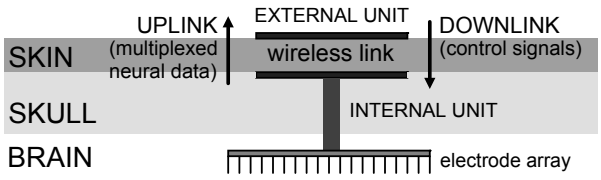


Figure A-1: Simplified view of the neural prosthesis system described in [262].

It is generally advantageous to separate the power and data transfer functions of a wireless link [87, 287]. Power signals carry no information, and power transfer efficiency is maximized for narrowband (high-Q) links that operate at low frequencies to minimize losses in body tissue. On the other hand, data signals carry information and therefore require larger link bandwidths, which are more easily obtained at higher operating frequencies. Separating the two functions therefore allows them to be independently optimized, improving overall performance. In this paper, we therefore assume an independent power link is already present, and focus only on the data link. A detailed discussion of RF power link design may be found in [9].

Wireless power transfer may or may not be continuous. Continuous transfer uses an implanted rectifier, and may be avoided by incorporating a sealed rechargeable battery into the implant. In either case, the power dissipated by the implanted

system should be minimized. Excess heat dissipated within the body results in tissue damage. In addition, the number of useful battery recharges is limited (typically to about 10^3), so reducing implant power also prolongs battery life.

In implanted systems with many neural recording electrodes [118], the data rate from the implanted system to an external unit can be quite high since each electrode typically requires at least 5kHz of bandwidth. We term this the *uplink*. Sending high-bandwidth data on the uplink is expensive in power. In this paper, we describe a wireless data link that solves this problem by pushing as much of the power and complexity to the external unit as possible. The power consumption of the external unit is less critical since its batteries may be easily changed. We also need a low-bandwidth data link from the external unit to the implanted unit for control, programming and feedback information. We term this data link the *downlink*. Our system minimizes implanted power in the downlink as well.

A brief version of this paper with preliminary results and no analysis was presented at a conference [185]. The current paper is organized as follows. A theoretical analysis of impedance modulation is carried out in Section A.2. The external and internal transceiver chips are described in Sections A.3 and A.4, respectively. The performance of the communication system is analyzed in Section A.5, while experimental results are presented in Section A.6. Finally, Section A.7 concludes our discussion.

A.2 Theoretical Analysis of Impedance Modulation

Our data link is designed to be half-duplex, i.e., either uplink or downlink data (but not both) can be transmitted at any given time. We use impedance modulation to transmit uplink data, while downlink data is transmitted via inductive coupling (transformer action). Impedance modulation, also known as load modulation, absorption modulation or backscattering, is a well-known technique used in RFID tags, low-power wireless sensors and biomedical implants [37, 52, 82, 88, 106, 122, 136, 165,

186, 214, 273, 285, 326]. One side (the source) of a coupled pair of resonators is driven with a sinusoidal source. The impedance of the other, load resonator is switched between two or more discrete states to transmit data. The resultant amplitude and/or phase modulation of the source waveform is detected to receive the data. The coupling between the resonators can be near-field (inductive) or far-field (radiative) in nature. In this section we theoretically analyze the performance of such communication systems. Our analysis simplifies and generalizes previous work, for example that in [37, 88, 285].

A.2.1 General Considerations

Assume a voltage source v_G with an output impedance Z_G is coupled to a load Z_L via a linear, reciprocal two-port network. The situation is shown in Figure A-2, where Z_{11} and Z_{22} are the input impedances at the source and load terminals, respectively, with the other side open-circuited. In addition Z_{12} is known as the *transfer impedance* between the source and load. It can be shown that the input impedances seen at the source and load terminals, denoted by Z_T and Z_R, are given by

$$\begin{aligned} Z_T &= Z_{11} - \frac{Z_{12}^2}{Z_{22} + Z_L} \\ Z_R &= Z_{22} - \frac{Z_{12}^2}{Z_{11} + Z_G} \end{aligned} \tag{A.1}$$

where the second term on the right-hand-side of each equation is known as the *reflected impedance* and is the net effect of the coupling network [9]. Equation (A.1) reveals that the transfer impedance of the coupling network acts as an impedance inverter (gyrator). The circuit shown in Figure A-2 is generic and can represent both near-field (reactive) and far-field (radiative) coupling between the source and load. For near-field coupling between two inductors L_1 and L_2, we have

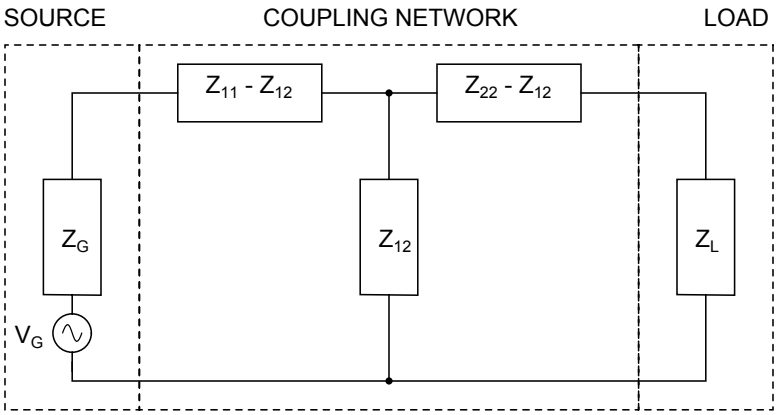


Figure A-2: Generic circuit representation of coupling between a source (driven network) and a load (passive network) via a linear, reciprocal two-port.

$$Z_{11} = \jmath\omega L_1 + R_1$$

$$Z_{12} = \jmath\omega M \tag{A.2}$$

$$Z_{22} = \jmath\omega L_2 + R_2 \tag{A.3}$$

Here M is the mutual inductance between L_1 and L_2, while R_1 and R_2 are their series resistances. Notice that Z_{12} is purely imaginary since the coupling was assumed to be purely reactive, with no energy lost due to radiation. In general the medium separating the coils has finite impedance $Z_m = R_m + \jmath X_m$, which will modify Z_{11}, Z_{12} and Z_{22} to new values Z'_{11}, Z'_{12} and Z'_{22}. If the medium, such as biological tissue, has non-zero conductivity, i.e., $R_m \neq 0$, Z'_{12} will contain a real component even in the absence of radiation. In addition, the reactance of the medium, X_m, will introduce capacitive coupling between the resonators. We shall ignore the effects of Z_m to simplify our analysis. These assumptions are a good approximation for transcutaneous links unless the coils are designed to have very high values of Q or the operating frequency is higher than 30-40MHz.

We can also model the far-field case, in which case the coupling occurs via antennas. Assuming that both antennas are impedance-matched, i.e., $Z_G = Z_T^*$ and $Z_L = Z_R^*$, and that the coupling medium is free space, it can be shown that

$$
\begin{aligned}
Z_{11} &= R_1 + \jmath X_1 \\
Z_{11} &= R_2 + \jmath X_2 \qquad \text{(A.4)} \\
Z_{12} &= \sqrt{G_1 G_2 R_1 R_2}\left(\frac{\lambda}{2\pi r}\right) \qquad \text{(A.5)}
\end{aligned}
$$

Here the antennas have input (terminal) impedances $Z_{11} = R_1 + \jmath X_1$ and $Z_{22} = R_1 + \jmath X_1$, gains G_1 and G_2 and are separated by a distance r. The operating wavelength is λ. Notice that in this case Z_{12} is entirely real because power transfer is assumed to occur only via radiation.

A.2.2 Impedance Modulation

Consider a near-field inductive link. The canonical example of this system is shown in Figure A-3(a), along with an equivalent representation in Figure A-3(b). For simplicity, we have ignored the impedance of the medium separating the coils. Consider this system in sinusoidal steady state. The source network is driven by a sinusoidal current source $i_{in} = I_{in}\sin(\omega_0 t)$. However, Figure A-3(b) is identical to Figure A-2 if we use the relationships in (A.2) and also identify $Z_G = 1/(sC_1)$ and $Z_L = 1/(sC_2)$. We can therefore immediately use (A.1) to write

$$
\begin{aligned}
Z_{in} &= \frac{v_1}{i_{in}} = Z_G + Z_T = Z_G + Z_{11} - \frac{Z_{12}^2}{Z_{22} + Z_L} \qquad \text{(A.6)} \\
&= \frac{1}{\jmath\omega C_1} + (\jmath\omega L_1 + R_1) + \left(\frac{\omega^2 M^2}{\jmath\omega L_2 + R_2 + \frac{1}{\jmath\omega C_2}}\right)
\end{aligned}
$$

The system is usually operated around a frequency ω_0 where both the source and load networks resonate, i.e., $\omega_0 = 1/\sqrt{L_1 C_1} = 1/\sqrt{L_2 C_2}$. Let us define $Q_1 = \sqrt{L_1/C_1}/R_1$ and $Q_2 = \sqrt{L_2/C_2}/R_2$ to be the quality factors of the source and load networks, respectively. At the resonance frequency ω_0, we can simplify (A.6) to

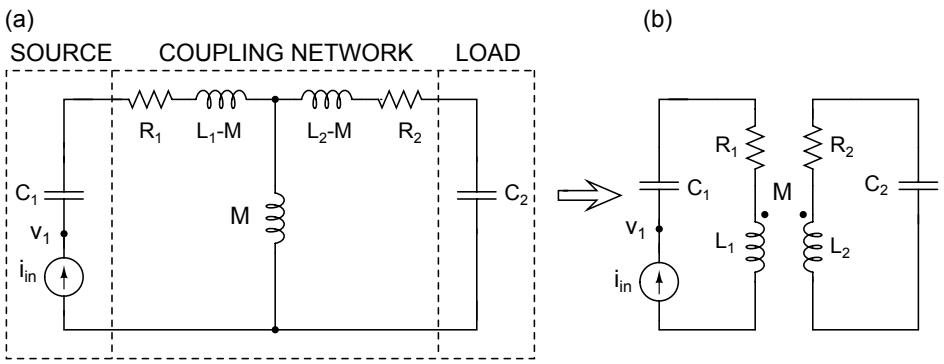


Figure A-3: The basic near-field inductive link, represented in two equivalent ways.

$$Z_{in} = R_1\left(1 + k^2 Q_1 Q_2\right) \equiv R_1\left(1 + m\right) \tag{A.7}$$

where the *coupling coefficient* between the inductors is denoted by k, defined as $k = M/\sqrt{L_1 L_2}$ where $0 < k < 1$, and is a function of coil geometry and separation. In addition, $m = k^2 Q_1 Q_2$ is known as the *modulation index*. In impedance modulation, the data to be transmitted changes the quality factor Q_2 of the load network, thereby modulating the voltage across the source network via the change in reflected impedance. The amount of modulation is maximized by changing Q_2 to a low value Q_2' in one state, say when a '1' is to be transmitted, and keeping $Q_2 \gg 1$ in the other state (when a '0' is transmitted). The easiest way to implement this scheme is by closing a switch to short-circuit the load network whenever a '1' bit is to be transmitted, making $Q_2' = 1$, and opening the switch whenever a '0' bit is to be transmitted. Intuitively, $Q_2' = 1$ because all the energy stored inside the inductor L_2 during a single RF cycle must be dissipated inside its parasitic series resistance R_2. None of that energy can be stored in the capacitor C_2, because it is shorted out. Therefore the ratio of energy stored to energy dissipated per cycle is unity, i.e., $Q_2' = 1$[1].

The resultant values of v_1 are then given by

[1]For similar reasons, the effective quality factor of inductorless (ring and relaxation) oscillators for phase noise calculations is also unity.

$$
\begin{aligned}
v_1|_{BIT=0} &= i_{in} R_1 \left(1 + m/Q_2\right) \\
v_1|_{BIT=1} &= i_{in} R_1 \left(1 + m\right)
\end{aligned}
\tag{A.8}
$$

We see that impedance modulation causes amplitude shift-keying (ASK) of the source voltage v_1. Since Z_{in} remains purely real in either state, phase modulation is absent. For simplicity, we shall assume that data bits are encoded as rectangular pulses.

The coupling coefficient k typically decreases as d^{-3}, where d is the separation between the coils. The strong dependence of m on k therefore makes impedance modulation unsuitable for long-range links. For short-range links, however, it possesses the great advantage of dissipating almost no power on the transmitting (load) side of the link. This advantage allows an implanted system to transmit data while keeping its power consumption to a minimum; most of the power is dissipated by the external unit.

A.2.3 An Alternative Topology

A practical problem with low-power implementations of the system shown in Figure A-3 is that Z_{in} is on the order of R_1, which is usually just the series loss in the inductor L_1, and is quite small (on the order of a few Ω). With any reasonable level of input current i_{in}, the result is extremely small voltage amplitudes v_1 across the source network. This low impedance level may not be a problem when the inductive link is used to transfer power, but it is undesirable for data links. A simple way to increase the impedance across which v_1 is expressed is to convert the series resonant circuit shown in Figure A-3 into a parallel one, as shown in Figure A-4. The input current source i_{in} is created by driving the gate of a transistor (which has transconductance g_m) with a sinusoidal voltage $v_{in} \sin\left(\omega_0 t\right)$. Again, assuming that $\omega_0 = 1/\sqrt{L_1 C_1} = 1/\sqrt{L_2 C_2}$, we find that

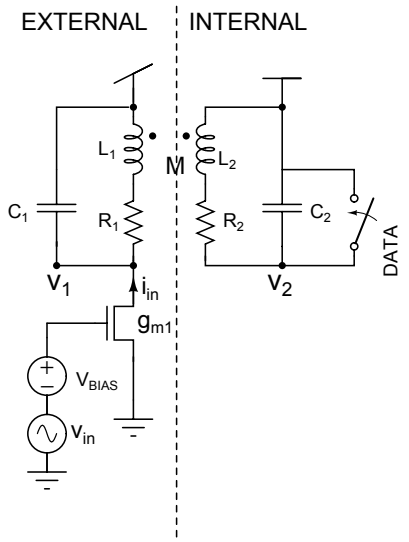


Figure A-4: Simplified schematic of the impedance-modulated wireless data link described in this paper.

$$Z_{in} = \frac{v_1}{i_{in}} \approx R_1 \left(1+m\right) \left(\frac{Q_1}{1+m}\right)^2 = \frac{R_1 Q_1^2}{1+m} \tag{A.9}$$

with the approximation being valid if $1 + k^2 Q_1 Q_2 \ll Q_1$, and we have used a series-to-parallel impedance transformation to derive the result. Since $Q_1 \gg 1$, this condition reduces to $k \ll 1/\sqrt{Q_2}$. It is instructive to rewrite (A.9) as follows:

$$Z_{in} = \frac{\left(R_1 Q_1\right)^2}{R_1 \left(1+m\right)} = \frac{Z_0^2}{R_1 \left(1+m\right)} \tag{A.10}$$

Here $Z_0 = R_1 Q_1 = \sqrt{L_1/C_1}$ and $m = k^2 Q_1 Q_2$, as before. We see that going from a series to a parallel resonant circuit has resulted in an impedance inversion (gyration) of the form $Z_{in} \rightarrow Z_0^2/Z_{in}$. The effective modulation depth m_{eff} is now given by

$$m_{eff} = \frac{1/\left(1+m/Q_2\right) - 1/\left(1+m\right)}{1/\left(1+m/Q_2\right)} = \frac{m\left(1-1/Q_2\right)}{1+m} \tag{A.11}$$

and is approximately equal to m if $m \ll 1$ and $Q_2 \gg 1$. The external resonator voltage amplitude switches between $v_1/\left(1+m/Q_2\right)$ and $v_1/(1+m)$ based on whether a '1' or '0', respectively, is being transmitted from the implanted unit. Equivalently,

coupling to the internal resonator lowers the external resonator's quality factor from Q_1 to $Q_1/\sqrt{1+m/Q_2}$ or $Q_1/\sqrt{1+m}$, depending on whether the internal resonator is shorted or not.

A.2.4 Downlink

During the downlink phase, the switch across the internal resonator is kept open. The amplitude of the RF voltage across the external resonator is modulated by the data; some of this amplitude variation is coupled to the internal resonator via the mutual inductance between L_1 and L_2. The voltage transfer function between the external resonator voltage v_1 and the internal resonator voltage v_2 is given by

$$m_d \equiv \frac{v_2}{v_1} = kQ_2\sqrt{\frac{L_2}{L_1}} \tag{A.12}$$

If we assume that the two coils are matched, so that $L_1 = L_2$ and $Q_1 = Q_2$, we see that $m_d = \sqrt{m}$. The amplitude of the received data signal is given by $m_d v_1$ for the downlink and $m_{eff} v_1$ for the uplink. The weaker dependence of m_d on k means that it falls off less rapidly with distance than m.

A.3 External Transceiver Implementation

We now describe the circuits used in our wireless data link. Separate internal and external transceiver chips have been designed and built. In this section we describe the transceiver designed for the external unit.

A.3.1 Front-end Circuits

A simplified schematic of the external transceiver is shown in Figure A-5. All analog circuits on the chip are biased using an on-chip 2μA supply-independent CMOS current reference. The current reference is cascoded to improve its output impedance; it also uses a startup circuit that operates off the power supply turn-on transient and consumes no static power [184]. A power-on-reset circuit is used to ensure that

all digital registers and latches initialize to a known state when the power supply is first turned on. The power-on-reset consists of a $C-R$ circuit that differentiates the power supply turn-on ramp to produce a spike. This spike is converted to a logic-level global reset pulse by a Schmitt trigger.

The source resonator is incorporated into an RF oscillator, described later and shown in Figure A-8, that runs continuously when the transceiver is receiving data (TX is low). This arrangement saves power by combining the functions of the local oscillator and power amplifier into a single circuit. The internal transceiver uses a switch to short out or open-circuit its resonator (the load), thus modulating the envelope of the RF oscillator. Two envelope detectors, each using an MOS diode as the nonlinear element, individually track the positive and negative sides of the RF envelope (which move differentially). Each envelope detector (ED) leak current I_{ED} and load capacitor C_{ED} is set to 2μA and 2.6pF, respectively. These values are chosen so that the ED time constant τ_{ED} is slow enough to filter out most of the RF carrier but fast enough not to significantly attenuate the data signal. The ED time constant is given by

$$\tau_{ED} = \frac{C_{ED}}{g_{m,diode}} = \frac{C_{ED}\phi_T}{I_{ED}} \tag{A.13}$$

The difference between the two ED outputs is amplified by an AC-coupled differential amplifier and fed into a comparator. The op-amps in this preamplifier are standard two-stage CMOS designs that consume 45μA each. The values of the resistors R_1 and R_2 are 20kΩ and 100kΩ, respectively; this choice makes the nominal gain of the preamplifier equal to $G_A = 1 + R_2/R_1 = 6$. The AC-coupling network uses $C_3 = 10$pF and $R_3 = 1$MΩ. We want to make the R_3C_3 time constant as large as possible to avoid signal attenuation during long runs of '0's or '1's, especially when the data rate is low. We use an asynchronous comparator that consists of a wide-output-swing operational transconductance amplifier (OTA) biased at a total current of 37.5μA. The output current when the comparator is slewing is 30μA.

Figure A-6 shows experimentally measured comparator inputs and output at a

data rate of 2Mbps. Feeding the two amplified envelopes differentially into the comparator increases our received signal-to-noise power ratio by a factor of 2 (3dB) compared to a single-ended scheme that tracks only one side of the modulated RF signal.

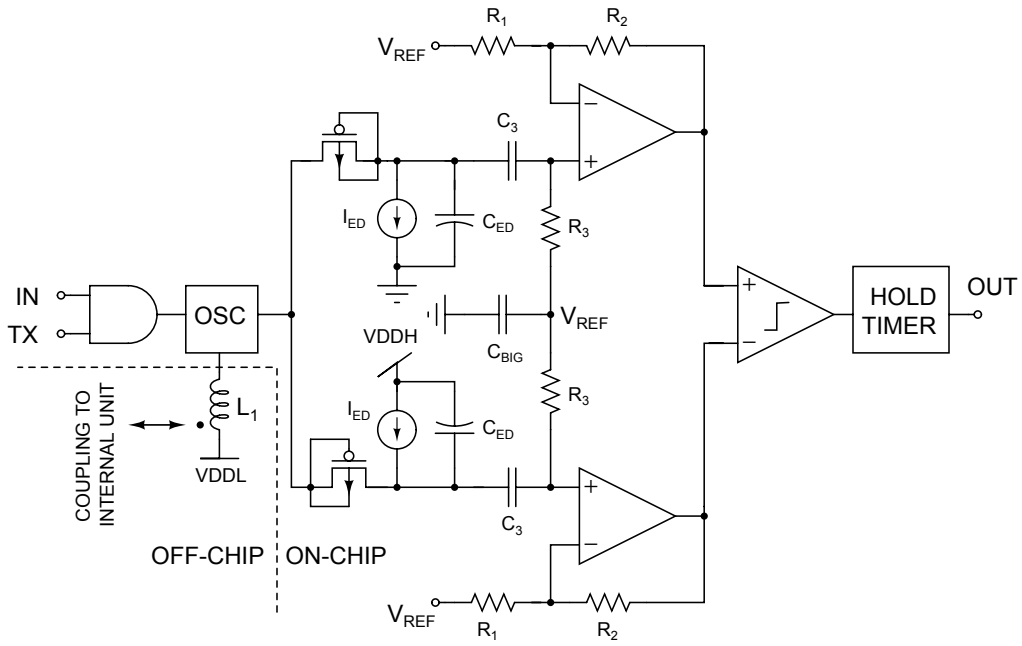


Figure A-5: Simplified schematic of the external transceiver. The output of the hold timer is fed into a PLL (not shown in this figure).

The output of the comparator is passed into a hold timer circuit, shown in Figure A-7. The hold timer eliminates pulses that are shorter than a certain fixed duration. This strategy eliminates multiple transitions along data edges because of noise. The dead zones on the low-high and high-low input transitions are denoted by t_{LH} and t_{HL} respectively, and are given by

$$\begin{aligned} t_{LH} &= \frac{C_L\left(V_{DDH} - V_{TL}\right)}{I_N} \\ t_{HL} &= \frac{C_L V_{TH}}{I_P} \end{aligned} \tag{A.14}$$

where V_{TL} and V_{TH} are, respectively, the low and high trip-points of the Schmitt trigger. The overall effect of the hold timer is similar to using a hysteretic comparator. The hold time for both low and high transitions was set to 100ns. Its output is fed

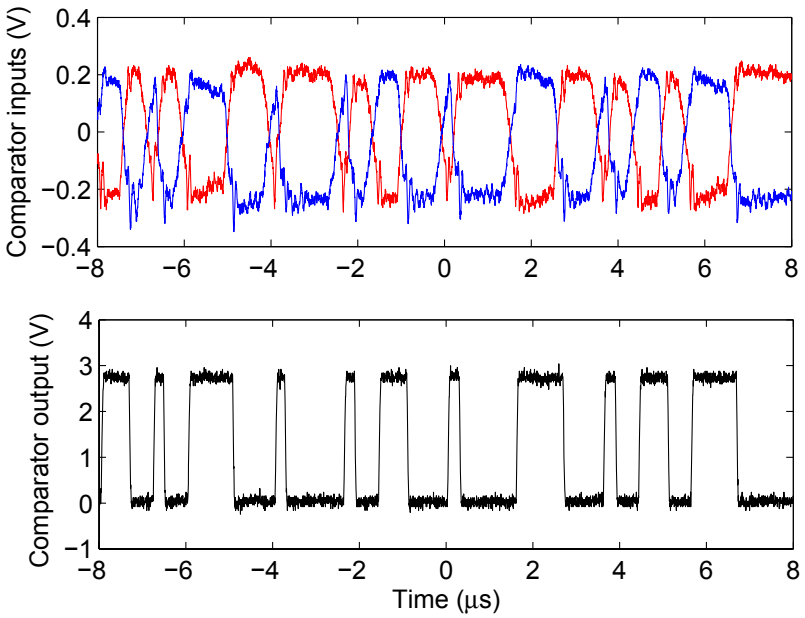


Figure A-6: Experimentally measured comparator inputs (top), and output (bottom). The data rate was 2.5Mbps and the coils were placed 3.5cm apart.

into a phase-locked loop (PLL) for clock and data recovery (CDR).

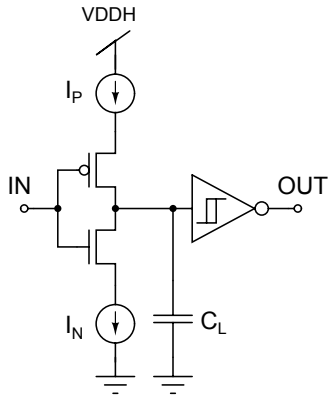


Figure A-7: Schematic of the hold timer circuit.

A.3.2 The RF Oscillator

Figure A-8 shows the RF oscillator. All components except L_1 are on-chip. The oscillator uses the resonant tank formed by L_1 and C_1 as the main frequency-selective element. A CMOS inverter in its high-gain region is used to provide enough loop gain

to induce oscillations. Operation in this region is ensured by setting $V_{DDL} \approx V_{DDH}/2$, where V_{DDH} is the power supply for the inverter and the rest of the chip. The voltages V_{DDL} and V_{DDH} can be created by connecting two identical batteries in series. The amplitude of the oscillation, V_1 (and power consumption) increases with the aspect ratio of the transistor in parallel with L_1 and C_1. The high-pass filter formed by R_2 and C_2 acts as a negative delay element (predictor) that cancels out the inverter delay t_{inv} [202].

If the inverter delay has been completely canceled, the oscillation frequency is simply given by $\omega_0 = 1/\sqrt{L_1 C_1}$. By using an identical L and C as the internal resonator, we can now guarantee operation at the resonant frequency of both resonators. Since inductor and capacitor values match well, the need for any additional frequency-tuning is eliminated. It can be shown that, in order for the inverter delay to be completely canceled, we must have

$$R_2 C_2 = \frac{1}{\omega_0 \tan(\omega_0 t_{inv})} \tag{A.15}$$

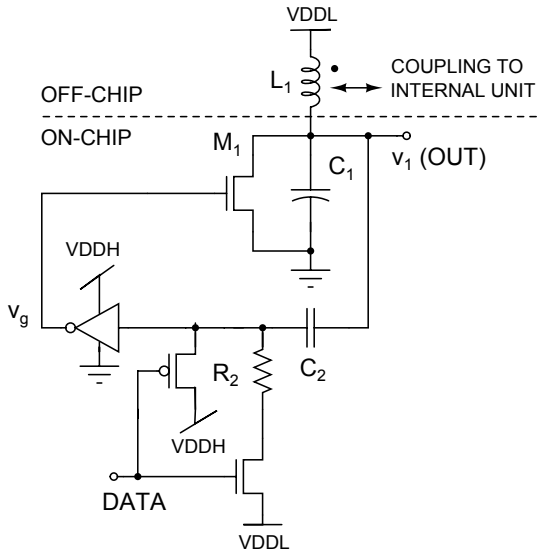


Figure A-8: The source-side (external) RF oscillator.

Increasing the oscillation frequency f_{osc} increases the quality factors of the inductors, which increases m_{eff}, allowing smaller oscillation amplitudes to be used and thereby lowering power consumption. However, at very high frequencies the coils

eventually self-resonate and losses due to body tissues also increase, effectively lowering Q. In our case a good compromise between these competing factors was obtained around 25MHz, which is where we operate.

A.3.3 The Phase-Locked Loop

The PLL, shown in Figure A-9, allows the uplink to use non-return (NR) data encoding, which maximizes data rate for a given link bandwidth. It uses a Hogge-type phase detector (PD) [158], shown in Figure A-10, a cascoded charge pump and a passive, third-order loop filter. The charge pump uses differential switching to reduce charge injection errors (see Figure A-11). The loop filter contains two additional high-frequency poles (indicated in the figure) that are placed beyond the crossover frequency of the loop transmission. These poles filter out high-frequency ripple on V_{OUT}, thereby reducing jitter in the output clock while only minimally degrading the phase margin of the loop. The reference voltage V_{REF} is normally set to V_{DDL}, which is a convenient value midway between the rails.

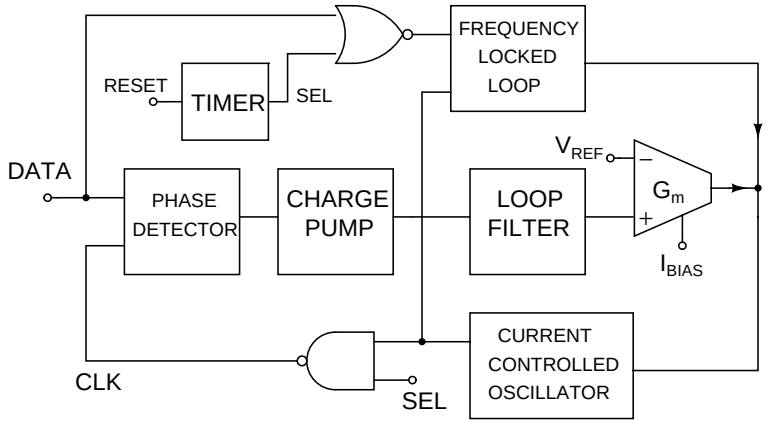


Figure A-9: Block diagram of the phase-locked loop (PLL). The timer enables an auxiliary frequency-locked loop after a system reset occurs.

The output of the loop filter, V_{OUT}, is converted to a current by a wide-linear-range transconductor (WLR) [258] that combines a well-input differential pair and other linearization techniques to achieve over 1.5V of input linear range. The output current of the WLR, I_{CCO}, is fed into a current-starved ring oscillator (CCO). The

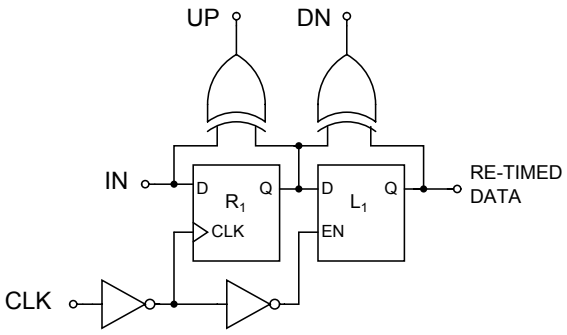


Figure A-10: The Hogge phase detector used in the CDR PLL.

loop bandwidth is set to 20KHz (about 1% of the nominal data rate, which is 2Mbps). This value is low enough to ensure that the loop remains locked even when long 'runs' of consecutive '1's or '0's occur in the input data stream. The loop locks when the CCO frequency is twice the data rate (a result of using the Hogge PD, which also outputs re-timed data with edges synchronized to the recovered clock).

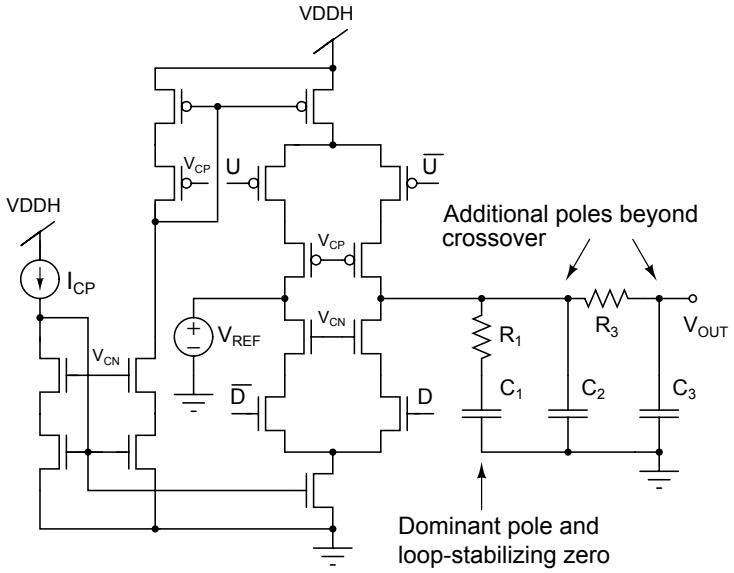


Figure A-11: Schematic of the charge pump and loop filter used within the PLL.

True phase detectors, such as the Hogge, do not provide much *frequency* error information. This property enables a CDR PLL to remain locked when missing edges appear in the input data stream, but reduces its capture range. As a result, CDR PLL's usually need additional circuitry to aid acquisition of lock. We have designed a frequency-locked loop (FLL) for this purpose. The FLL is shown in Figure A-12. A

timer activates it for a fixed number of clock cycles after a system reset (see Figure A-9). During this period, the main PLL is disabled and the internal unit transmits a synchronization sequence consisting of alternating '0' and '1' bits. The FLL is a first-order loop that counts clock and data edges and sets the bias current of the CCO so that their rates are equal. It does this by using a digital accumulator (which acts as the loop filter) and current DAC. Each data edge (rising or falling) increments the accumulator, while the divide-by-2 circuit ensures that only rising clock edges decrement it. Therefore the FLL, like the PLL itself, is locked when the clock runs at double the data rate. When the timer disables the FLL the CCO is already running close to the right frequency, so the PLL locks more easily. In our implementation, the accumulator and DAC have five bits each, which limits the initial frequency error that must be handled by the PLL to less than $1/2^5 \approx 3\%$ of the data rate.

Figure A-13 shows experimentally measured oscillator output (i.e., clock) frequencies as a function of the loop filter output voltage for two different power supply voltages. The loop filter output voltage is fed into the WLR transconductor to generate the CCO control current I_{CCO}. The measured curves are fairly linear, but the slope changes noticeably when V_{DDH} is changed from 2.8V to 3.3V, indicating that the tuning gain of the CCO changes with V_{DDH}. This change occurs because t_{del}, the delay of each current-starved inverter forming the CCO, increases with V_{DDH}. In fact, if the PMOS and NMOS sides are ratioed such that the inverters are balanced, the threshold voltage must be approximately $V_{DD}/2$. Therefore, to first order we have $t_{del} \approx C_L V_{DD} / (2I_{cco})$.

A.3.4 Downlink

During the transmit phase (TX is high in Figure A-5) data is sent to the implanted system via on-off keying (OOK). In this phase the data signal turns the oscillator on and off (see Figure A-8), thus changing the amplitude of the voltage across the source resonator from v_1 to 0. This change v_1 is coupled to the load resonator (in the implanted unit) by transformer action between L_1 and L_2. The amplitude of the voltage across it varies between v_2 to 0, where $v_2/v_1 = m_d$ and m_d is given by (A.12).

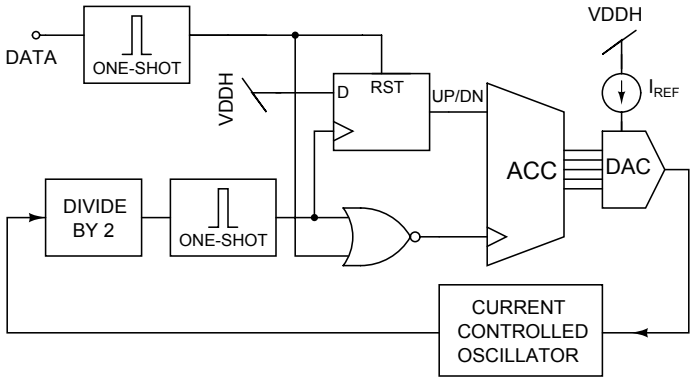


Figure A-12: Block diagram of the frequency-locked loop.

To save power, all receiver circuits in the external unit are turned off during this phase.

It is interesting to compare the relative amounts of amplitude modulation obtained for the uplink (which uses impedance modulation) and the downlink (which uses inductive coupling) for the same RF carrier voltage v_1. The ratio of signal amplitudes obtained in the two cases is given by

$$\frac{m_{uplink}}{m_{downlink}} \equiv \frac{m_{eff}}{m_d} \approx \frac{k^2 Q_1 Q_2}{kQ_2\sqrt{L_2/L_1}} = kQ_1\sqrt{\frac{L_1}{L_2}}, \tag{A.16}$$

where we have assumed that $m_{uplink} \ll 1$. We see that $m_{uplink} > m_{downlink}$ for large values of k. However, the situation is reversed as k decreases.

Downlink data is encoded using 25 / 75% pulse-width modulation (PWM) before transmission using a counter-based on-chip modulator. Pulse-width modulation, pulse-position modulation and other return-to-zero (RZ) encoding schemes are spectrally inefficient. However, it makes sense to use an RZ scheme for the downlink since the data rate is low in this direction. Every bit period in RZ data contains at least one level transition that can be used as a clock edge. Therefore, a PLL is not needed for CDR in the implanted system, minimizing its complexity and power consumption.

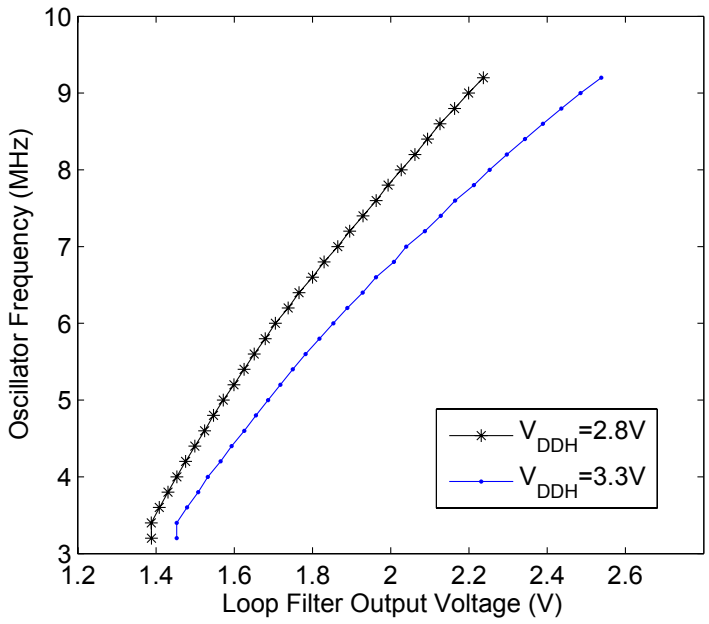


Figure A-13: Experimentally measured VCO tuning curves for two different values of V_{DDH}.

A.4 Internal Transceiver Implementation

A simplified block diagram of the internal transceiver is shown in Figure A-14. All analog circuits on the chip are biased using an on-chip 0.2μA CMOS current reference. During the transmit phase (TX is high in Figure A-14), the data stream to be transmitted turns the impedance-modulation switch M_2 in parallel with the load resonator on and off. To save power, all receiver circuits in the internal unit are turned off during this phase.

During the receive phase (TX is low) the impedance-modulation switch is turned off and an envelope detector is used to track the voltage on the load resonator. This voltage follows the PWM-encoded OOK bitstream transmitted from the external transceiver. The output of the envelope detector is compared with a reference voltage V_{ref} that is implicitly generated by making the MOS diode connected to the negative input terminal of the comparator 16 times wider than that connected to the positive terminal. Since the two diodes carry the same current, in the absence of any coupled signal the voltage on the negative terminal exceeds that on the positive terminal by

$V_{ref} = (\phi_T/\kappa)\ln(16) \approx 96\text{mV}$, and the comparator output is low (here $\kappa \approx 0.75$ is the subthreshold constant). It goes high only when enough signal is coupled into L_2 for the voltage at the positive terminal to increase by an amount greater than V_{ref}. The value of V_{ref} is chosen to be large enough to comfortably exceed the unknown input-referred offset voltage of the comparator, which is usually in the range of 10-20mV.

The envelope detector time constant is $\tau_{ED} = R_{ED}C_{ED}$, where $R_{ED} = 500\text{k}\Omega$ and $C_{ED} = 5.2\text{pF}$. The comparator is a wide-output-swing OTA biased at a total current of $0.5\mu\text{A}$. A hold timer, similar to the one in the external transceiver, is used to remove spurious transitions in the comparator output waveform due to noise. The hold time for both low and high transitions was set to $1\mu\text{s}$. The diode D_2 is a parasitic p-n junction diode present between the source/drain and well terminals of M_2. It clamps the voltage across the internal resonator if its amplitude exceeds one diode drop (about 0.6V).

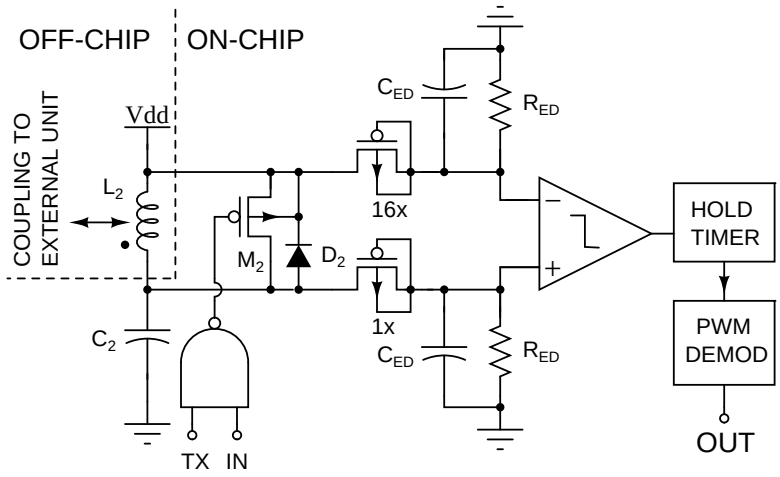


Figure A-14: Simplified block diagram of the internal transceiver.

The output of the hold timer is fed into a pulse-width demodulator circuit that regenerates the downlink data stream from the PWM waveform. The demodulation circuit, shown in Figure A-15, uses two capacitors, of value C_A and C_B, and charges them using the currents I_A and I_B when the PWM waveform is high and low, respectively. At the end of a bit period, the voltages across the capacitors, V_A and V_B, are

compared to determine if the bit transmitted was a '0' or a '1'. The capacitors are then reset and the process starts again for the next bit. The reset pulses R_1 and R_2 are obtained by digitally differentiating V_{PWM}.

A reasonable way to optimize the circuit is to require that $\Delta V_0 = -\Delta V_1$, where V_0 (V_1) is the difference between V_A and V_B at the end of a '0' ('1') bit. Assuming that '0' and '1' bits are equally likely, this condition minimizes the bit error ratio f or a given signal-to-noise ratio. It can be shown that, in this optimal situation, we must have

$$\frac{(I_A/C_A)}{(I_B/C_B)} = \left(\frac{2}{\alpha_0 + \alpha_1} - 1\right) \tag{A.17}$$

where $0 < \alpha_0 < 1$ and $0 < \alpha_1 < 1$ are the pulse widths (normalized by the bit period) used to signal '0' and '1', respectively. In our case $\alpha_0 = 0.25$ (high pulse = 25% of bit period) and $\alpha_1 = 0.75$ (high pulse = 75% of bit period). From (A.17) we should use $(I_A/C_A) \,/\, (I_B/C_B) = 1$. To simplify layout, we make $I_A = I_B$ and $C_A = C_B$.

A.5 Performance Analysis

In this section we theoretically analyze the performance of our communication system. This analysis will be important later for showing that the experimental performance of our system matches that predicted by theory.

A.5.1 Pulse Width Distortion Mechanism

The voltage across the implanted resonator has asymmetric rising and falling edges because its quality factor Q_2 is different in the two states (switch OFF and ON, respectively). When the switch is turned off, the voltage gradually increases (with an exponential envelope) since Q_2 is high. On the other hand, when the switch is turned on, the voltage quickly dies away since Q_2 is now approximately zero (see Figure A-16).

The envelope of the external resonator voltage is a low-pass filtered version of

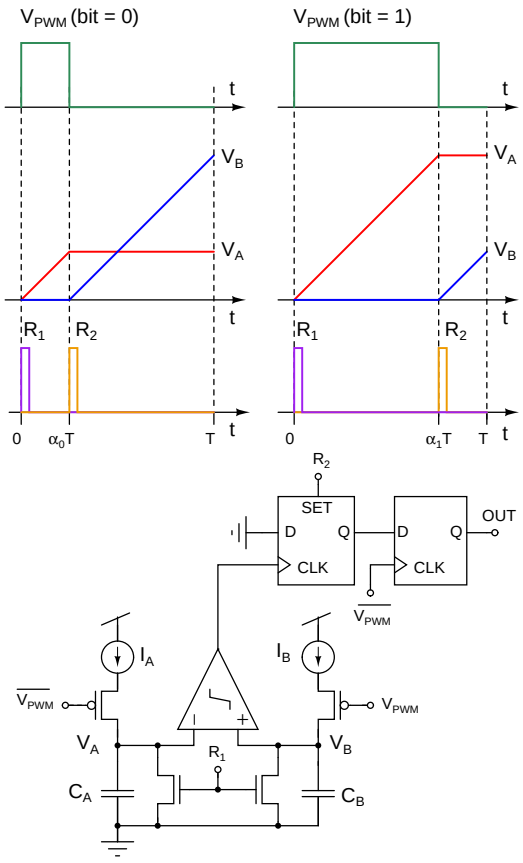


Figure A-15: The pulse-width demodulator circuit, and relevant waveforms.

that on the implanted resonator, since the external resonator behaves like a low-pass filter centered around the carrier frequency. The corner frequency of the low-pass filter formed by the external resonator is constant since its quality factor Q_1 remains almost constant. To first order, filtering provided by the external resonator adds an equal amount of delay to rising and falling edges. As a result, the detected external envelope continues to have rising data edges that are delayed *relative to the falling edges*. Hence, $0 \rightarrow 1$ transitions are delayed compared with $1 \rightarrow 0$ transitions, as shown in Figure A-16. Therefore, the length of the first '1' bit in a continuous sequence of '1' bits is shortened. Figure A-6 illustrates that isolated '1' bits have a shorter pulse-width compared to isolated '0' bits, another manifestation of this effect.

An additional complication, also illustrated in Figure A-16, is that diode clamping occurs when the peak internal voltage V_2 exceeds about 0.6V. Under these conditions, the parasitic source/drain junction diode D_2 associated with the switch M_2 (see Fig-

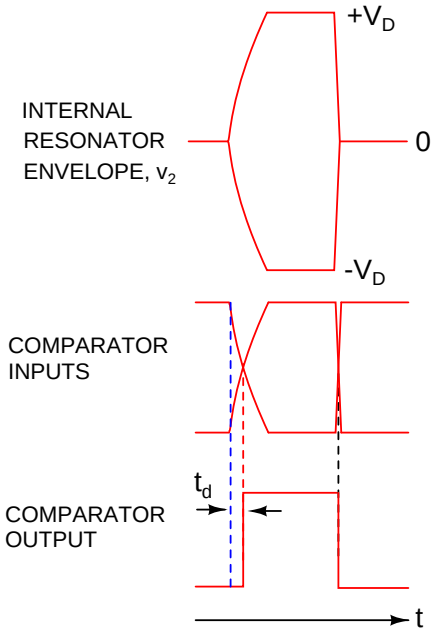


Figure A-16: Waveforms that show how asymmetric rise and fall times at the comparator inputs delay rising edges, thus shortening the width of high pulses ('1' bits). The shortening is reduced when the internal resonator voltage amplitude is diode clamped to lie between $\pm V_D$.

ure A-14) turns on, clamping v_2 and preventing it from increasing further. This effect only occurs when $V_2 = m_d V_1$ exceeds V_D, where m_d is the voltage transfer function between the internal and external resonator voltages (given by (A.12)), V_1 is the amplitude of the external resonator voltage (set by the RF oscillator), and $V_D \approx 0.6$V is the turn-on voltage of diode D_2, which we assume to be approximately constant. Although diode clamping in Figure A-14 occurs only when v_2 is larger than V_{DD} by V_D, the bandpass nature of the L_2C_2 circuit transforms asymmetric clamping to symmetric clamping in v_2 : The inductor L_2 and capacitor C_2 in Figure A-14 together form a high-Q filter centered around the carrier frequency when the switch M_2 is open, thus ensuring that the positive and negative envelopes of the voltage across the internal resonator are both be equal to V_D (as shown in Figure A-16).

The comparator in the external transceiver compares the positive and negative envelopes of the RF oscillator. These envelopes are differential signals (see Figure A-5). Therefore the comparator output switches state when the envelope voltages reach their average (DC) value. It is conceptually easy (but algebraically tedious; for details,

please see the Appendix) to show that t_{dr}, the mean delay of rising data edges, is

$$\begin{aligned} t_{dr} = & \quad \tau_2 \ln(2) - \tau_2 \ln(2-\alpha), & m_d V_1 < V_D \\ = & \quad \tau_2 \ln(2) - \tau_2 \ln\left(2 - \frac{\alpha V_D}{m_d V_1}\right), & m_d V_1 \geq V_D \end{aligned} \tag{A.18}$$

where $\tau_1 = 2Q_1/\omega_0$ and $\tau_2 = 2Q_2/\omega_0$ are the time constants of the external and internal resonator voltage envelopes, respectively. The parameter α is the mean (DC) voltage of the envelope waveform normalized to its ideal value, which is half the peak voltage. Since we get high pulses that have less area than low pulses, $\alpha < 1$. Equation (A.18) captures two physical effects. Firstly, diode clamping serves to "speed up" time constants because RC waveforms start to look more like square waves. Secondly, a non-symmetric duty cycle in the envelope waveforms ($\alpha < 1$) causes zero crossings to occur before the halfway point between the 'low' and 'high' levels is reached, again reducing delay.

The exact value of α depends on statistical properties of the data stream that establish its mean (DC) value. We see from equation (A.18) that t_{dr} remains constant with m_d when m_d is small, and then gradually decreases. Since $m_d \propto k$, we expect t_{dr} to remain constant with coil separation when the coils are far apart, and then gradually decrease, following (A.18), as they are brought together.

It is important to note that only asymmetries between rising and falling edges distort pulse widths at the PLL input. Comparator delay, for example, does not distort pulse widths as long as it is the same for both edges. However, any comparator offset V_{off}, since it delays the comparator trip point in one direction but advances it in the other, does affect pulse widths. However, we found that in our case comparator voltage offset was not a significant source of pulse-width distortion.

A.5.2 Bit Error Mechanism

Bit errors occur when the PLL samples the wrong value for the input bit. As long as both '0' and '1' data bits have the same length (i.e., there is no duty cycle distortion

in the input) and the PLL is locked, rising clock edges are aligned with input data transitions. As a result, the Hogge phase detector samples the input data stream in the middle of each input bit. This fact may be seen from the circuit diagram shown in Figure A-10. When the recovered clock from the VCO goes low, the current value of the input data stream is sampled by the D-register R_1. This value appears at the output of the D-latch L_1 (as the re-timed data bit) half a bit period later, when the clock goes high. It is easy to see that an error will occur if the rising data edge is delayed by more than half a bit period with respect to the negative edge of the clock, or the falling edge advanced by the same amount. In either case, the clock samples the wrong data value, i.e. '0', rather than its correct value, i.e., '1'.

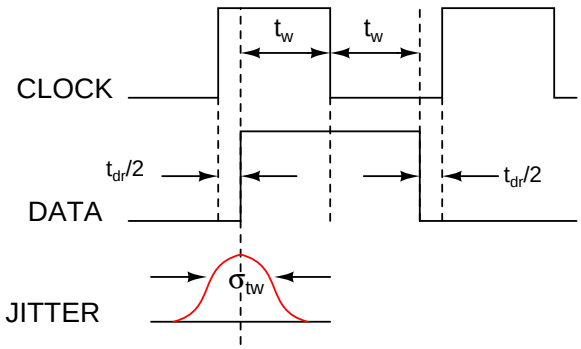


Figure A-17: Typical PLL input and output waveforms. Delay and jitter reduce the time window t_w, thereby increasing the bit error rate.

We call the mean value of the time between data transitions and negative clock edges the *time window* t_w. It is easy to see that, when no duty-cycle distortion is present, $t_w = T/2$, where T is the bit period.

For simplicity, consider a PLL with no timing offset and 50% duty cycle in the recovered clock. In the presence of duty cycle distortion in the input data stream, rising clock edges can no longer remain aligned with both rising and falling data transitions since this would make the clock non-periodic. The PLL resolves this situation as shown in Figure A-17: it shifts the phase of the clock such that there is equal but opposite timing error at both types of data transitions. In other words, the PLL ensures that the delay between rising clock and rising data edges is equal to that between falling data and rising clock edges. Since the two timing errors are

equal, equal-width 'UP' and 'DN' signals are generated by the Hogge phase detector whenever a data transition occurs. As a result, the average charge being fed into the loop filter remains zero, i.e., the PLL remains locked.

Asymmetric rising and falling waveforms in the internal resonator cause duty-cycle distortion: rising data edges are delayed relative to falling edges by an amount equal to t_{dr}, thereby making '1' bits shorter than '0' bits. At lock, the PLL will divide any timing error equally between rising and falling data edges. Therefore the timing error at each data transition becomes $t_{dr}/2$. Thus errors at rising and falling transitions occur with equal probability: either the first or the last '1' in a run of several '1' bits can be sampled incorrectly, resulting in bit errors. The time window t_w we have for sampling the data correctly is now given by

$$t_w = \frac{T}{2} - \frac{t_{dr}}{2} \tag{A.19}$$

Both the input to the PLL and the clock contain random timing jitter. Jitter in the input data stream is caused by voltage noise that is present at the output of the comparator while it is changing state. Such noise is generated by the envelope detectors, the differential amplifiers and the comparator itself. Some of the jitter in the synthesized clock is caused by input jitter that has been low-pass filtered by the PLL, while the rest is introduced by the VCO. The presence of timing jitter makes t_w a random variable (see Figure A-17); its variance is given by

$$\sigma_{tw}^2 = \sigma_{data}^2 + \sigma_{clk}^2 \tag{A.20}$$

where we have assumed that σ_{data} and σ_{clk}, the timing jitters of the input data and synthesized clock, respectively, are uncorrelated with each other and equal for positive and negative transitions in both waveforms. Assuming Gaussian (normal) probability distributions for both these quantities, the probability that the PLL samples an incorrect value is given by

$$P_e = \frac{1}{2}\text{erfc}\left(\frac{t_w}{\sqrt{2}\sigma_{tw}}\right) \tag{A.21}$$

A.5.3 BER Calculation

We used a maximal-length pseudo-noise (PN) sequence of length $2^N - 1$, generated on-chip by the internal transceiver, to test the uplink. Since most of the bit errors happen because of timing error at data transitions, the bit error ratio (BER) of the system is dominated by errors where the *first or last* bit in a run (consecutive sequence) of '1' bits is misread as a '0'. The BER is thus given by

$$BER = P_e \times \left(P_{1,1} + \sum_{n=2}^{N} 2P_{1,n} \right) \tag{A.22}$$

where P_e is given by (A.21), $P_{1,n}$ is the probability that a run of n '1's occurs, and the longest runs are N bits long. This equation expresses the fact that we can get two, and only two, bit errors whenever we have runs of two or more '1's. For maximal-length PN sequences, it can be shown that $P_{1,n}$ decreases exponentially with n [210], i.e.,

$$P_{1,n+1} = \frac{P_{1,n}}{2}, \qquad (N-1) \geq n \geq 1 \tag{A.23}$$

Maximal-length PN sequences are almost balanced: the number of '1's is only one more than the number of zeros. Therefore the probability of a given bit being a '1' approaches 1/2 for large values of N, i.e., $\sum_{n=1}^{\infty} nP_{1,n} = 1/2$. We can solve this equation by using (A.23). The result is

$$P_{1,n} = \frac{1}{2^{n+2}} \tag{A.24}$$

By substituting (A.24) into (A.22) and approximating the sum as an infinite series (i.e., assuming that $N \to \infty$), we find that $BER \approx 3P_e/8$. Therefore, from (A.21), our overall BER is given by

$$BER \approx \frac{3}{16} \mathrm{erfc} \left(\frac{t_w}{\sqrt{2\left(\sigma_{data}^2 + \sigma_{clk}^2\right)}} \right) \tag{A.25}$$

Since the run-length statistics of PN sequences are known (see (A.24)), we can also calculate the value of α, the normalized mean voltage of the external resonator envelope waveform. Using (A.18), we can then analytically find the time delay t_{dr}. We find that, to a very good approximation,

$$\begin{aligned} \alpha = & \quad 1 - \frac{\tau_2}{T}\frac{\left(1 - e^{-T/\tau_2}\right)}{\left(2 - e^{-T/\tau_2}\right)}, & \beta < 1 \\ = & \quad 1 - \frac{\tau_2}{2T}\left[1 + (\beta - 1)\ln\left(1 - \frac{1}{\beta}\right)\right], & \beta \geq 1, \end{aligned} \tag{A.26}$$

where the parameter $\beta = m_d V_1 / V_D$. For a detailed derivation of this equation, please see the Appendix.

A.5.4 Downlink

During the downlink, the voltage across the internal resonator is modulated by turning the external oscillator ON and OFF. The internal resonator is always in a high-Q state. Therefore, to first order, rise and fall times are equal and no pulse-width distortion occurs. In addition, the data rate is much lower for the downlink, reducing sensitivity to any residual duty cycle distortion mechanism. As a result, uplink limitations dominate the performance of our system. Therefore we do not further analyze the downlink.

A.6 Experimental Results

Separate external and internal transceiver chips, each 1.5mm × 1.5mm in size, were fabricated in the AMI 0.5μm CMOS process. Figure A-18 shows die photographs of both chips. About 40% of the core area of the external transceiver chip (shown on the right) is occupied by the resistors and capacitors in the PLL loop filter. The internal transceiver chip is shown on the left.

Figure A-19 shows the two printed circuit boards that were used to test the wireless link. Identical transmit and receive coils were printed on the boards. Each coil was

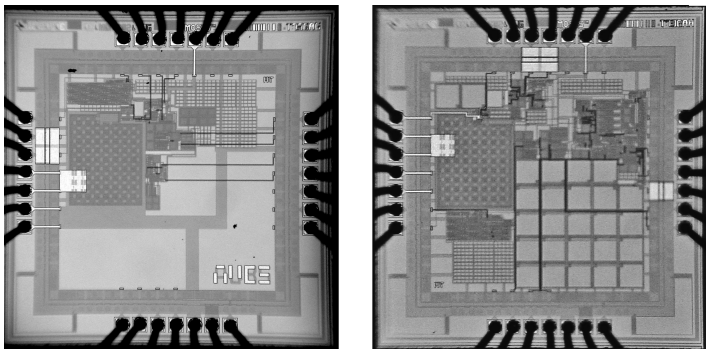


Figure A-18: Internal (left) and external (right) transceiver chip photographs. Each die measures 1.5mm × 1.5mm in size.

square, 3.5cm on a side and had two turns. The designed inductance was 500nH with a simulated quality factor of 30 at 25MHz. Packaged chips were surface mounted on the boards and they were aligned parallel to each other at various separations for testing. No external components were needed apart from the coils, decoupling capacitors and power supplies.

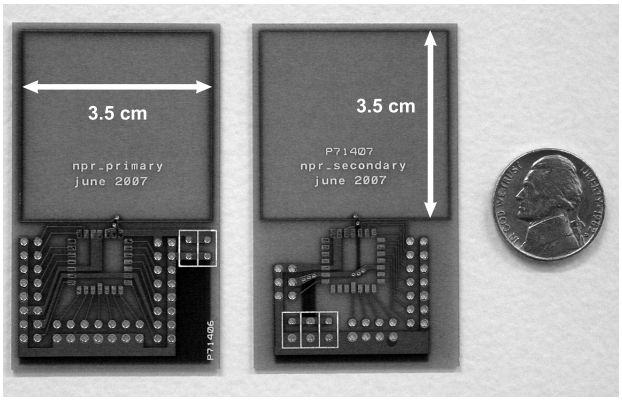


Figure A-19: Test boards used for making experimental measurements.

Implanted coils for neural recording are typically < 2cm on a side and operate at link distances between 0.3cm and 1cm. However, our coils are somewhat larger than average (3.5cm on a side). To reduce the coupling constant k to more typical values, and allow for possible coil misalignment, we tested our link over larger distances (between 1.5cm and 5cm). Figure A-20 shows the measured value of the effective modulation depth for the uplink as a function of r, the separation between the coils. We expect the coupling between the coils to vary approximately as $k \propto (a^2 + r^2)^{-3/2}$,

where $a = 2$cm is the radius of a circular coil with the same area as our square coils. We get a good fit to the measured data using this model for k when we assume that the internal voltage is clamped to a maximum of $V_D = 0.6$V, $Q_1 = 10$ and $Q_2 = 25$. The fit is relatively insensitive to the exact value of a and V_D. The value of Q_1 is significantly lower than the predicted quality factor of the unloaded coil (about 30) because of the finite output impedance of the transistor M_1 in Figure A-8. Figure A-20 also shows that ignoring the diode clamp predicts much higher values of modulation index than are actually observed.

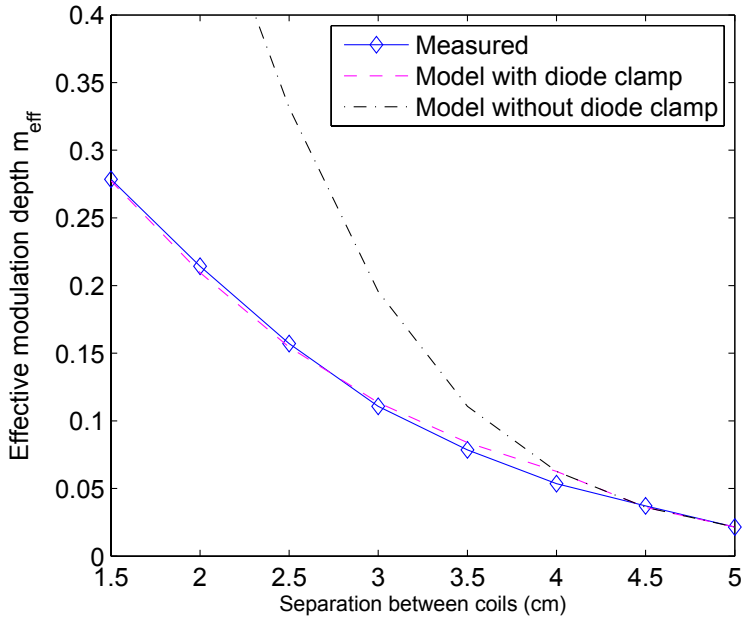


Figure A-20: Measured uplink modulation depth m_{eff} as a function of link distance, compared to predictions made by analytical models with or without a diode clamp on the internal resonator.

Figure A-21 shows transmitted and received data and recovered clock waveforms measured for the uplink at 5.8Mbps with the coils 2cm apart. The PLL synchronizes rising edges of the recovered clock to data transitions. Falling edges of the clock therefore appear in the middle of each bit and are used to sample the data stream. At this separation, the PLL was observed to lock over data rates varying between 1Mbps and 5.8Mbps. The upper end of the lock range is set by the loop filter, which runs out of linear range and hits V_{DDH}. At low data rates the loop filter

voltage becomes low enough to turn on the well-source diode present at the input of the wide-linear-range transconductor [258], thus setting the lower end of the lock range. Notice that isolated '1' bits at the output of the comparator are significantly narrower than isolated '0' bits . This effect, which is also visible in Figure A-6, is a manifestation of the pulse-width distortion mechanism discussed in the previous section.

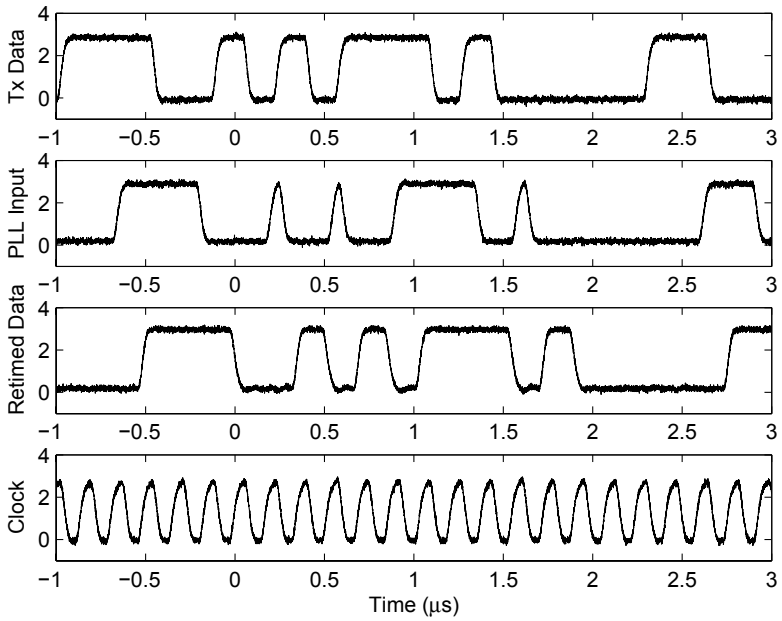


Figure A-21: Uplink data transmission at 5.8Mbps with the coils 2cm apart.

The top plot in Figure A-22 shows typical experimentally measured transitions in the comparator output. Many transitions have been overlaid on top of each other to evaluate the statistics of the threshold-crossing instant. The probability distribution function (pdf) of this random variable is shown in the lower plot. We see that the pdf shows two distinct peaks. The first peak is produced by falling edges, while rising edges produce the second peak. The pdf is fit to a model consisting of a weighted sum of two Gaussian distributions. The results of the fit are also shown on the lower plot. The difference between the means of the two Gaussian distributions constitutes the amount of pulse-width narrowing t_{dr} measured under these conditions. The standard deviations of the two Gaussian curves are approximately equal, confirming one of our

earlier assumptions. This measured quantity is σ_{data}, the rms jitter of rising data edges.

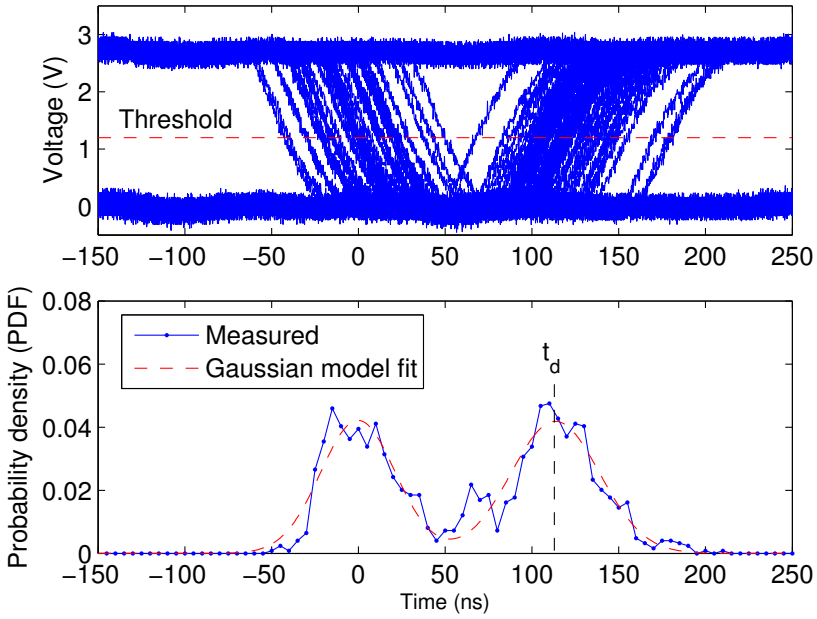


Figure A-22: Measured comparator output at a data rate of 2.5Mbps and 3cm link distance: (top) level transitions, showing the delay between falling and rising edges, and (bottom) probability distribution of threshold-crossing times and fit to a Gaussian mixture model.

The measurements shown in Figure A-22 were repeated for various data rates and link distances. Similar time-domain measurement techniques were used to also find σ_{clk}, the rms jitter of the clock produced by the PLL. The results were found to depend weakly on data rate but strongly on link distance. To reduce measurement errors, t_d, σ_{data} and σ_{clk} were therefore averaged over the data rate. The resultant values are plotted in Figures A-23(a) and A-23(b) as a function of the link distance. The rms clock jitter σ_{clk} remains almost constant with link distance. Its average value is 7.2ns.

The measured delay was also fitted to the value predicted by theory, i.e. equation (A.18), with α given by (A.26). An excellent fit was obtained, as shown in Figure A-23(a).

Figure A-24 shows experimentally measured BER for the wireless uplink when

the coils were placed 2, 3 and 4cm apart. Error rates less than 10^{-6} could not be measured with our experimental setup in a reasonable amount of time. Experimentally measured values of t_w, σ_{data} and σ_{clk} were also plugged into equation (A.25) to draw predicted BER curves. The experimentally measured BER values match well with those predicted theoretically for the 2cm case. The predicted values are significantly lower than the experimental values for the two other link distances, implying the existence of an additional error mechanism. Experimental evidence indicates that transient loss of lock (so-called cycle-slipping) becomes increasingly common as the link distance increases and the received signal-to-noise ratio decreases. This process may account for the increased number of bit errors at large link distances. Our hypothesis is supported by the fact that the lock range of the PLL (which limits the range of data rates for which we could measure BER) decreases as the link distance increases.

Figure A-25 shows transmitted and received data waveforms (before and after pulse-width demodulation) measured for the downlink at 200kbps with the coils 2cm apart. Recovered data transitions are aligned with rising edges of the pulse-width modulated signal. Falling edges of this signal can therefore be used to sample the data stream. The downlink was observed to operate over data rates varying between 15kbps and 300kbps. When the data rate is less than 15kbps the capacitor voltages V_A and V_B in the PWM demodulator (see Figure A-15) hit V_{DD} before the bit period T is complete, resulting in demodulation errors. When the data rate is greater than 300kbps the bit period becomes smaller than the minimum allowable pulse width in the data waveform (set to 1μs by the hold timer). The hold timer starts ignoring data transitions separated by a single bit period, again resulting in errors.

We have also measured the BER for the downlink. No bit errors were observed when the coils were placed 2cm apart and approximately 2×10^5 bits were transmitted at data rates varying between 15kbps and 300kbps. Assuming bit errors follow Poisson arrival statistics, we can therefore say, with approximately 85% confidence, that the downlink BER is less than 10^{-5} over the tested range of data rates.

We tested two sets of chips and boards. No major differences in performance

were noted between them. We also inserted a 2cm thick layer of 0.9% saline solution between the coils and repeated some of our tests. As in prior reports, for example [205], the goal was to verify that the wireless link would operate normally in the presence of body tissue. No significant differences in performance were noted. We have also estimated the amount of power radiated by our system (see Appendix). The calculation shows that our current design meets regulatory requirements in the United States for unlicensed, low-power wireless devices.

A.7 Conclusions

We have presented a low-power bidirectional wireless link for neural prostheses and analyzed its performance. Table A.1 summarizes the measured performance of our bidirectional wireless link. A useful figure of merit that characterizes the power-efficiency of any communication system is the *energy consumed per bit* E_{bit}, which is defined as

$$E_{bit} = \frac{P_{diss}}{R} \tag{A.27}$$

where P_{diss} is the total power dissipation and $R = 1/T$ is the data rate. Significant power optimization can be carried out on our current design. For example, the RF oscillator's power consumption can be considerably reduced by adapting its power supply voltage V_{DDL} based on the value of k. Nevertheless, we get $E_{bit} = 0.96$nJ/bit for a bit error rate of $< 10^{-6}$ and 2cm coil separation. We were unable to measure bit error rates below 10^{-6} in reasonable amounts of time so this bit-error rate should only be viewed as an upper bound.

Table A.2 compares our uplink to wireless data links for biomedical implants that have recently been reported in the literature. The link described in [109] is a far-field link, while the others are inductive near-field links. Ideally it should be possible to compare the efficiencies of these designs by using energy consumed per bit as the metric. However, in each case what is available is not the power consumed by the the entire communication system, but by the receiver or transmitter alone. The absence

of total power consumption data makes quantitative system efficiency comparisons difficult. However, attractive features of our design include high data rate, low bit error rate and extremely low power consumption in the internal unit.

Table A.1: Data link: Performance summary

Parameter	**Value**
Link distance	tested up to 5cm
Center frequency	25MHz
Uplink data rate	1Mbps - 5.8Mbps (at 2cm)
Uplink encoding	Non-return (NR)
Downlink data rate	15kbps - 300kbps (at 2cm)
Downlink encoding	Pulse-width modulation (PWM)
Power supply voltages	$V_{DDH} = 2.8$V / $V_{DDL} = 1.4$V
External power consumption	2.5mW (uplink) / 1.5mW (downlink)
Internal power consumption	100μW (uplink) / 140μW (downlink)
Fabrication process	AMI 0.5μm CMOS
Chip size	1.5mm×1.5mm (each transceiver)

Table A.2: Data link: Performance comparison

Reference	**Coil diameter (Tx/Rx)**	**Link distance**	**Carrier frequency**
[89]	2cm/1.2cm	0.5cm	5MHz/10MHz
[121]	3.5cm/2.7cm	1.5cm	10MHz
[109]	0.47mm/35cm (antenna)	13cm	433MHz
This work	3.5cm/3.5cm	2cm	25MHz
Reference	**Data rate**	**BER**	**Power consumption**
[89]	2.5Mbps	10^{-5}	0.38mW (Rx only)
[121]	1.12Mbps	10^{-5}	0.61mW (Rx only)
[109]	330kbps	3×10^{-3}	1.81mW (Tx only)
This work	2.8Mbps	$< 10^{-6}$	0.1mW (Tx)/2.5mW (Rx)

We conclude by discussing two techniques for further improving the performance of our wireless link. We have simulated these techniques and plan to implement them on a future design iteration.

A.7.1 Resonator Amplitude Control

The RF amplitude across the external resonator can be reduced to save power when the modulation depth m_{eff} is high enough. There are three main parameters of the

current design that can be varied to control this amplitude: the power supply voltage V_{DDL}, the aspect ratio of the transistor M_1 that drives the external resonator and, finally, the conduction angle θ, i.e. the fraction of the oscillation period during which M_1 conducts current. Alternatively, a more easily controllable oscillator topology can be used. For example, in a Colpitts oscillator amplitude can be conveniently controlled using bias current.

A.7.2 Soft Switch Turn-On

As discussed in this paper, the main cause of uplink bit errors is pulse-width distortion caused by asymmetric rising and falling waveforms in the internal resonator. A simple technique that substantially reduces this effect is to turn on the switch M_2 (shown in Figure A-14) in a "soft", i.e. gradual way. By slowing down the falling edges of the resonator voltage waveform, this technique decreases t_{dr}, the amount of pulse-width narrowing caused by the resonators. A basic circuit implementation of this idea uses an inverter that is current starved on the pull-down, i.e. NMOS side to drive M_2. The value of the maximum pull-down current is set using feed-forward or feedback to ensure equal delay on rising and falling edges.

Acknowledgements

The authors wish to thank Scott K. Arfin for help with figures and photography, and the anonymous reviewers. We also gratefully acknowledge the support of the McGovern Institute Neurotechnology (MINT) program.

Appendix

Derivation of Equation (A.18)

Consider the envelope of the voltage on the secondary resonator. In the absence of diode clamping, i.e., $m_d V_1 < V_D$ or $\beta < 1$, the voltage increases like an RC circuit

with time constant τ_2 from zero towards its peak value $m_d V_1$. However, as shown in Figure A-26, it falls abruptly to zero. The delay t_{dr} is equal to the length of time taken by the voltage to rise to the average value of the waveform, which by definition is $\alpha \times m_d V_1/2$. Therefore we have

$$\begin{aligned} m_d V_1 \left(1 - e^{-t_{dr}/\tau_2}\right) &= \frac{\alpha m_d V_1}{2} \\ \Rightarrow t_{dr} &= \tau_2 \ln(2) - \tau_2 \ln(2 - \alpha) \end{aligned} \tag{A.28}$$

When diode clamping occurs, i.e., $m_d V_1 \geq V_D$ or $\beta \geq 1$, the peak value of the envelope voltage is limited to V_D. Therefore its average value is given by $\alpha \times V_D/2$. The delay is still given by the time it takes for the voltage to reach this value, so we have

$$\begin{aligned} m_d V_1 \left(1 - e^{-t_{dr}/\tau_2}\right) &= \frac{\alpha V_D}{2} \\ \Rightarrow t_{dr} &= \tau_2 \ln(2) - \tau_2 \ln\left(2 - \frac{\alpha V_D}{m_d V_1}\right) \\ &= \tau_2 \ln(2) - \tau_2 \ln\left(2 - \frac{\alpha}{\beta}\right) \end{aligned} \tag{A.29}$$

Derivation of Equation (A.26)

We want to calculate the normalized mean, i.e., α, of the input to each envelope detector in the external unit. For mathematical convenience, we instead calculate the normalized mean voltage envelope across the internal resonator. The two quantities are equal since they are related to each other by the transfer function of the external resonator, which is that of a linear low-pass filter with respect to the envelope.

The internal resonator waveform, as shown in Figure A-26, has slow rising edges but fast falling edges. Therefore it is zero everywhere within '0' bit periods. In other words, '0' bits do not contribute to the mean value. To find the mean we therefore start by decomposing the '1' bits into runs of 1,2,3...n consecutive '1's and

finding $\overline{v_2(1)}$, $\overline{v_2(2)}$...$\overline{v_2(n)}$, the mean value in each case. The mean value of the entire waveform, $\overline{v_2}$, can now be obtained by summing up, with appropriate probability weighting factors, $\overline{v_2(n)}$ for different values of n. The n-th term in the summation must be weighted by the product of two terms. The first is the probability that a run of n '1' bits occurs, i.e., from (A.24), $P_{1,n} = 1/2^{n+2}$. The second term is the number of '1' bits, which is simply n.

For mathematical convenience, we consider very long sequences, i.e. $N \rightarrow \infty$. The exponential decrease in $P_{1,n}$ with n makes the resultant infinite sum a very good approximation to the actual mean voltage even for relatively small values of N (for example, our on-chip implementation used $N = 17$). Therefore, $\overline{v_2}$ is well approximated by

$$\overline{v_2} = \sum_{n=1}^{\infty} nP_{1,n}\overline{v_2(n)} = \sum_{n=1}^{\infty} \frac{n}{2^{n+2}}\overline{v_2(n)} \tag{A.30}$$

The parameter α is defined as the normalized mean voltage and is therefore given by

$$\begin{aligned} \alpha = & \quad \frac{\overline{v_2}}{m_d V_1/2}, & \beta < 1 \\ = & \quad \frac{\overline{v_2}}{V_D/2}, & \beta \geq 1 \end{aligned} \tag{A.31}$$

Now we just need to find $v_2(n)$ for two different cases: with and without diode clamping. As shown in Figure A-26(a), when diode clamping is absent, i.e., $\beta < 1$, $v_2(n)$ is given by

$$\begin{aligned} \overline{v_2(n)} &= \frac{1}{nT}\int_0^{nT} m_d V_1 \left(1 - e^{-t/\tau_2}\right) dt \\ &= m_d V_1 \left[1 - \frac{\tau_2}{nT}\left(1 - e^{-nT/\tau_2}\right)\right], \end{aligned} \tag{A.32}$$

where T is the bit period. When diode clamping is present (as shown in Figure A-

26(b)), i.e., $\beta \geq 1$, the mean is given by

$$\begin{aligned} \overline{v_2(n)} &= \frac{1}{nT}\int_0^{t_d} m_d V_1 \left(1 - e^{-t/\tau_2}\right) dt + V_D \frac{(nT - t_d)}{nT} \\ &= V_D \left[1 - \frac{t_d}{nT}(1-\beta) - \frac{\beta\tau_2}{nT}\left(1 - e^{-t_d/\tau_2}\right)\right], \end{aligned} \tag{A.33}$$

where t_d is the time taken by the envelope to reach the clamp voltage V_D. Its value can be calculated using

$$m_d V_1 \left(1 - e^{-t_d/\tau_2}\right) = V_D \tag{A.34}$$

Substituting equation (A.34) in equation (A.33) and simplifying, we get

$$\overline{v_2(n)} = V_D \left[1 - \frac{\tau_2}{nT}\left(1 + (\beta - 1)\ln\left(1 - 1/\beta\right)\right)\right] \tag{A.35}$$

Evaluating the sum in equation (A.30) for both cases ($\beta < 1$ and $\beta \geq 1$, respectively) and using the definition of α as given by equation (A.31) gives us equation (A.26).

Radiated Emissions

Regulatory issues are a concern for any implanted medical devices. Unfortunately, none of the unlicensed ISM frequency bands below 433MHz have enough bandwidth to support uplink data rates larger than a few hundred kbits/sec (the widest band, centered around 27.12MHz, is only 670KHz wide). Therefore high-data-rate systems like our uplink cannot hope to fit within an ISM band. Permissible radiation limits in the United States for unlicensed, low-power wireless devices operating *outside* ISM bands are set forth in Part 15 of the F.C.C. rules (Title 47 of the Code of Federal Regulations). According to the code, radiated emissions between 1.705 and 30.0MHz (with the exception of certain restricted frequency bands) must not exceed 30μV/m at a distance of 30m from the device.

We now perform a simple analysis to see whether F.C.C. regulations on radiated emissions are likely to be a concern for our system. A formula that predicts the radiation resistance R_{rad} of a small (much smaller than the wavelength λ) circular loop of N turns is (see, for example, [11]):

$$R_{rad} \approx 3.1171 \times 10^4 N^2 \left(\frac{S^2}{\lambda^4}\right) \Omega \tag{A.36}$$

where S is the area of the coil. Approximating our square external coil, which is 3.5cm on a side and has $N = 2$ turns, as a circular coil with the same area and number of turns, we get $R_{rad} = 9.0\mu\Omega$ for $\lambda = 12$m (25MHz). The radiated power is given by $P_{rad} = I^2 R_{rad}$, where I is the amplitude of the AC current in the coil. With $I = 1$mA, a typical value for our design, we get $P_{rad} = 9.0$pW. The maximum radiated power density (W/m^2) at a distance R from the coil is

$$P_{dens} = \frac{D_0 P_{rad}}{4\pi R^2} \tag{A.37}$$

where $D_0 = 1.5$ is the maximum gain produced by a small loop antenna. We also have $P_{dens} = E_{rad}^2/Z_0$, where E_{rad} is the radiated electric field and $Z_0 = 120\pi$ Ω is the impedance of free space. Plugging numbers into (A.37) for $R = 30$m gives us $E_{rad} = 0.67\mu$V/m, which is well below the F.C.C. specification. However, going to higher center frequencies with the current coil design is inadvisable since λ would decrease, increasing R_{rad}. Similarly, using a bigger coil, or one with more turns, would also cause R_{rad} to increase. In addition, keeping a safety margin is advisable. Increased radiation from the corners makes the actual radiation resistance of square coils larger than predicted by A.36.

A.8 Performance Limits of Impedance-Modulation Communication Systems

In this section we theoretically analyze the performance of near-field communication systems that use impedance-modulation.

A.8.1 Coupled Resonators

In this section we analyze inductively-coupled resonators in more detail. Consider the circuit shown in Figure A-27, where, as is usually the case in practice, losses are dominated by the series resistances of the inductors. In addition, G_3 represents the series resistance of the switch that is used for impedance modulation[2]. Assuming that $L_1C_1 = L_2C_2$ and defining $m = k^2Q_1Q_2$ as usual, it can be shown that the exact expression for the driving-point impedance of the primary at the resonance freqency, i.e., $\omega_0 = 1/\sqrt{L_1C_1} = 1/\sqrt{L_2C_2}$, is given by

$$Z_{1o}(\omega_0) = R_1Q_1^2\left[\frac{1}{1+m} - \frac{j}{Q_1}\right] \tag{A.38}$$

when $G_3 = 0$ and

$$Z_{1s}(\omega_0) = R_1Q_1^2\left[\frac{1}{1+m/Q_2} - \frac{j}{Q_1}\right] \tag{A.39}$$

when $G_3 \gg \omega_0C_2$, i.e., the secondary resonator is shorted out. All the quality factors are defined at the resonant frequency ω_0. The amplitude modulation in primary voltage v_1 produced by shorting and open-circuiting the secondary resonator is given by

$$\Delta v_1 = i_{in}\left(|Z_{1s}| - |Z_{1o}|\right) \tag{A.40}$$

When $1 + m \ll Q_1$, i.e., $k \ll 1/\sqrt{Q_2}$, we can ignore the imaginary parts of Z_{1o} and Z_{1s}, because they will be much smaller than the real parts. In that case the effective modulation depth, defined as $m_{eff} = \left(|Z_{1s}| - |Z_{1o}|\right)/|Z_{1s}|$, simplifies to our earlier approximate result, i.e.,

$$m_{eff} = \frac{m\left(1 - 1/Q_2\right)}{1+m} \tag{A.41}$$

At ω_0. the transfer function between v_2 and v_1 (assuming $G_3 = 0$) is given by

[2]For power links, G_3 will also include any effective load impedance being driven by the secondary.

$$\left.\frac{v_2}{v_1}\right|_{\omega_0} = kQ_1Q_2\sqrt{\frac{L_2}{L_1}}\left[\frac{1}{Q_1^2/(1+m)+jQ_1}-\frac{j}{Q_1}\right]$$
$$\approx -jkQ_2\sqrt{\frac{L_2}{L_1}} \tag{A.42}$$

where the approximation is again valid when $1+m \ll Q_1$, i.e., $k \ll 1/\sqrt{Q_2}$. Thus, for weak coupling between the resonators, v_2 and v_1 are 90° out of phase at resonance. In fact, the transfer function v_2/v_1 for weak coupling between the resonators is that of a resonant low-pass filter, with DC gain $k\sqrt{L_2/L_1}$, peak gain $kQ_2\sqrt{L_2/L_1}$ and quality factor Q_2. Since inductance L is ideally proportional to the square of the number of turns N, we get $v_2/v_1 \propto k\,(N_2/N_1)$, the usual transformer relationship. Using a resonant transformer increases the voltage gain by a factor of Q_2, but only over a limited bandwidth $\approx \omega_0/Q_2$.

Analysis of the system shown in Figure A-27 at frequencies other than ω_0 is too complicated to perform by hand. Instead, we wrote KCL equations and used Mathematica to analytically solve for the various transfer functions and driving point impedances. To be completely general, we allowed for the possibility that $L_1C_1 \neq L_2C_2$. Values of all components were normalized by the mean resonant frequency, defined as $\omega_0 = 1/\left(L_1C_1L_2C_2\right)^{1/4}$, and mean characteristic impedance, defined as $Z_0 = (L_1L_2/C_1C_2)^{1/4}$.

Figure A-28 shows calculated driving-point impedances Z_{1o} for identical resonators and various values of k, the coupling constant. We see that the resonant peaks split into two as k increases. The presence of two well-defined peaks defines the *strongly coupled* regime. Similarly, Figure A-29 shows calculated values of Z_{1s}. In this case we see that there is always a single resonant peak, but that it shifts to higher frequencies as k increases. In fact, (A.38) and (A.39) predict that both Z_{1o} and Z_{1s} at ω_0 saturate to $-jR_1Q_1$, which is independent of k, as $k \to 1$. This behavior can be seen in Figures A-28 and A-29, but they also illustrate the limitations of single-frequency analysis for understanding the strongly-coupled regime.

Figure A-30 shows how impedance modulation changes Z_1 between Z_{1o} and Z_{1s} in

the weakly and strongly-coupled regimes. Note that the effective modulation depth can be greatly increased in the strongly-coupled case by changing the operating frequency from ω_0 to the new, higher frequency where Z_{1o} is minimized and Z_{1s} maximized.

Figure A-31 shows the transfer impedance v_2/i_{in} for identical resonators and various values of k in the open-circuit case. We see that the transfer impedance also develops two resonant peaks as k increases, though the "notch" between them is less pronounced than for the driving-point impedance. Finally, Figure A-32 shows the voltage transfer function v_2/v_1 in the open-circuit case. In contrast to the transfer impedance, the voltage transfer function, which is a resonant low-pass function as expected, has a single peak for all values of k. However, this peak shifts to higher frequencies as k increases.

A.8.2 Synchronous Demodulation

Power Amplifier Analysis

In the sections that follow we shall analyze the receiver architecture shown in Figure A-33. We shall assume throughout that we are in the weakly-coupled regime, i.e., $k \ll 1/\sqrt{Q_2}$. There are two reasons for this assumption: firstly, it makes the problem analytically tractable, since the driving-point impedance still looks similar to that of a single resonator with some effective quality factor. Secondly, practical transcutaneous data links are usually not strongly coupled. Extending our analysis to the strongly-coupled case is a work in progress.

Figure A-33 shows a simple zero-IF receiver that uses a Gilbert mixer to synchronously demodulate the amplitude-modulated waveform on the external resonator. By using a series-to-parallel transformation it is easy to show that the total AC power dissipated in the source and load resonators (assuming that Q_1^2 is much greater than $(1+m)$) is given by

$$\begin{aligned} P_{tot,1} &= \frac{\left(v_1/(1+m)\right)^2}{2R_1\left(Q_1^2/(1+m)+1\right)} \approx \frac{v_1^2}{2R_1Q_1^2\left(1+m\right)} \\ P_{tot,0} &= \frac{v_1^2}{2R_1Q_1^2} \end{aligned} \tag{A.43}$$

where $P_{tot,1}(P_{tot,0})$ is the power dissipated when a '1'('0') is transmitted. The voltage transfer function between the source and load (with the source being the driven side) is given by

$$\frac{v_2}{v_1} = kQ_2\sqrt{\frac{L_2}{L_1}} \tag{A.44}$$

Assuming that Q_2^2 is much greater than $(1+m)$ when a '1' is transmitted and $Q_2 = 1$ when a '0' is transmitted, the power dissipated in the secondary resonator alone is

$$\begin{aligned} P_{sec,1} &= \frac{v_{2,1}^2\left(1+m\right)}{2R_2Q_2^2} = m_{eff}P_{tot} \\ P_{sec,0} &\approx 0 \end{aligned} \tag{A.45}$$

where $P_{sec,1}(P_{sec,0})$ is the power dissipated when a '1'('0') is transmitted. We see that the power consumed by the load resonator when it is not shorted out is m times that consumed by the source resonator. The average total power consumption is

$$\begin{aligned} \overline{P_{tot}} &= P_1P_{tot,1} + \left(1-P_1\right)P_{tot,0} \\ &= \frac{v_1^2\left(\left(1-P_1\right)+\frac{P_1}{1+m}\right)}{2R_1Q_1^2} \end{aligned} \tag{A.46}$$

where $0 < P_1 < 1$ is the probability that the bit being transmitted is '1'.

We shall assume that the current source i_{in} is produced by a transistor $M1$ with transconductance $g_{mp} = I_{Bp}/V_{Lp}$, where I_{Bp} is the DC bias current and V_{Lp} is the linear range of the transistor. The gate of $M1$ is driven a sinusoidal voltage of

amplitude v_1 and frequency ω_0. The voltage across the source resonator is given by

$$v_1 = g_{mp} v_{in} \times R_1 Q_1^2 \Rightarrow I_{Bp} = \left(\frac{v_1 V_{Lp}}{v_{in}} \right) \frac{1}{R_1 Q_1^2} \tag{A.47}$$

The power consumption of this power amplifier (PA) is

$$P_{PA} = I_{Bp} V_{DDp} \equiv \frac{\overline{P_{tot}}}{\eta} \tag{A.48}$$

where P_{tot} is the total AC power delivered to the two resonators, V_{DDp} is the PA's power supply voltage and η is its efficiency. Some algebra allows us to write

$$\eta = \left((1 - P_1) + \frac{P_1}{1+m} \right) \left(\frac{v_{in} v_1}{2 V_{DDp} V_{Lp}} \right) \tag{A.49}$$

To minimize power consumption we want to set $v_{in} = V_{Lp}$ so that the AC signal covers the entire linear range of the amplifier. If $v_{in} > V_{Lp}$ the gain will drop over some portion of the signal swing, causing clipping and distortion. Similarly if $v_{in} < V_{Lp}$, V_{Lp} can be lowered to decrease I_{Bp} (and thus the power consumption) while keeping g_m (and thus the gain) constant. By using (A.49), we get

$$\eta = \left((1 - P_1) + \frac{P_1}{1+m} \right) \left(\frac{v_1}{2 V_{DDp}} \right) \tag{A.50}$$

Since v_1 cannot swing below ground, it can only be as large as V_{DDp}. Therefore $\eta \leq 50\%$, which is exactly what we expect with a Class-A biasing strategy like the one described. Higher efficiencies can be achieved by using Class B or C biasing, at the cost of decreased power-handling capability and increased harmonic distortion. In any case the efficiency will always be less than 100%, so we may write

$$\eta = \left((1 - P_1) + \frac{P_1}{1+m} \right) \left(\frac{\eta_r v_1}{V_{DDp}} \right) \tag{A.51}$$

where $0 < \eta_r < 1$ is a dimensionless efficiency parameter. In terms of this parameter, therefore, the PA power consumption is

$$P_{PA} = \frac{v_1 V_{DDp}}{2\eta_r R_1 Q_1^2} \tag{A.52}$$

Mixer Analysis

The modulation of the source resonator amplitude is detected by removing the large carrier component at ω_0 using a synchronous scheme (such as a mixer) or an asynchronous scheme (such as a diode or other non-linear device used to build an envelope detector (ED)).

Since the source itself generates the RF signal, the carrier phase at the receiver is known. Thus a synchronous demodulation scheme can be used. Let's assume that we use a double-balanced Gilbert mixer to multiply v_1, the RF signal, with an amplified version of v_{in}, the input to the PA. This signal acts as the local oscillator (LO). For simplicity, let's also assume that the single-ended to differential converter (balun) required to drive the RF inputs of the mixer with v_{in} is noiseless and requires no power. A center-tapped transformer can be used for this purpose.

As shown in Figure A-33), the total bias current of the mixer is $2I_{Bm}$. The lower differential pair has transconductance $g_{m2} = I_{Bm}/V_{Lm}$ and is assumed to remain linear, while the LO is assumed to be a large-amplitude square wave that switches the upper differential pairs completely and instantaneously. The baseband output is produced across a resistive load R_L, while the capacitor C_L ensures that higher harmonics are filtered out. The output voltages are fed into a matched-filter receiver (not shown in Figure A-33) which recovers the transmitted bits.

Because the fundamental component of a square wave is $4/\pi$ times its amplitude and only half the energy goes to the difference frequency component, the conversion gain of the mixer is $G_c = 2g_{m2}R_L/\pi = 2I_{Bm}R_L/(\pi V_{Lm})$. Typically, G_c is made large enough to make the noise of any succeeding stages in the receiver irrelevant for its overall noise figure. The power consumption of the mixer is

$$
\begin{aligned}
P_{mixer} &= 2I_{Bm}V_{DDm} = \left(\frac{\pi V_L G_c}{R_L}\right) V_{DDm} \\
\Rightarrow G_c &= \frac{R_L P_{mixer}}{\pi V_{Lm} V_{DDm}}
\end{aligned} \tag{A.53}
$$

Using similar reasoning as for the PA, the optimal class-A biasing strategy is make the AC signal cover the entire linear range of the transconductor (in this case, the RF differential pair). Since each transistor in the differential pair sees an AC signal of amplitude $v_{in}/2$, we should make $v_{in}/2 = V_{Lm}$, i.e., $v_{in} = 2V_{Lm}$.

Noise Analysis

In this section we analyze the noise present within our model. We shall assume that we are dominated by thermal noise, i.e., that flicker ($1/f$) noise contributions are negligible. We first calculate the noise voltage present across the source resonator. There are three noise sources: the source-side resistance R_1, the transistor $M1$ and the load-side resistance R_2. The power spectral density (PSD) of the noise produced by R_1 and $M1$ is given by

$$
\begin{aligned}
\overline{v_{n1}^2} &= 4k_B T R_{1,eff}^2 \left(\frac{1}{R_{1,eff}} + \gamma_p g_{d0}\right) \\
&\approx 4k_B T \frac{R_1 Q_1^2}{(1+m)} \left(1 + \frac{1}{1+m}\left(\frac{\gamma_p v_1}{V_{Lp}}\right)\right)
\end{aligned} \tag{A.54}
$$

Here $g_{d0} \approx g_m$ is the drain-source conductance g_{ds} when $V_{DS} = 0$. Also, γ_p is the excess noise factor of the transistor and is always greater than $2/3$, its value for a long-channel MOSFET. The PSD at the load resonator due to R_2 is simply $4k_B T R_{2,eff} \approx 4k_B T R_2 Q_2^2/(1+m)$. The voltage transfer function between the source and load resonators, with the *load* being the driven side, is given by

$$\begin{aligned}\frac{v_1}{v_2} &= \sqrt{\frac{L_1}{L_2}}\left(\frac{kQ_1}{1+k^2\left(Q_1-1\right)}\right) \\ &\approx kQ_1\sqrt{\frac{L_1}{L_2}} \end{aligned} \tag{A.55}$$

Therefore the source-side noise PSD due to R_2 is given by

$$\overline{v_{n2}^2} = 4k_BT\frac{R_2Q_2^2}{(1+m)}\left(kQ_1\right)^2\left(\frac{L_1}{L_2}\right) \tag{A.56}$$

Thus the total noise PSD across the source resonator is given by

$$\begin{aligned}\overline{v_n^2} &= \overline{v_{n1}^2}+\overline{v_{n2}^2} \\ &= 4k_BT\frac{R_1Q_1^2}{(1+m)}\left(1+\frac{1}{1+m}\left(\frac{\gamma_p v_1}{V_{Lp}}\right)+k^2Q_1Q_2\right) \\ &= 4k_BTR_1Q_1^2\left(1+\frac{1}{\left(1+m\right)^2}\left(\frac{\gamma_p v_1}{V_{Lp}}\right)\right) \end{aligned} \tag{A.57}$$

Assuming that the LO switches the upper differential pair transistors instantaneously, the output noise PSD of the mixer due to the mixer itself is given by

$$\begin{aligned}\overline{v_m^2} &= 8k_BTR_L\left(1+\gamma_m g_{d0,1}R_L\left(\frac{g_{ds2}}{g_{ds2}+g_{m1}+g_{ds1}}\right)^2\right. \\ &\quad \left.+\gamma_m g_{d0,2}R_L\left(1+\frac{g_{ds2}}{g_{m1}+g_{ds1}}\right)^2\right) \\ &\approx 8k_BTR_L\left(1+\gamma_m g_{m,2}R_L\right) \\ &= 4k_BTR_L\left(2+\gamma_m\pi G_c\right) \end{aligned} \tag{A.58}$$

where g_{m1} and g_{ds1} refer to the LO transistors and g_{m2} and g_{ds2} refer to the RF transistors. We have assumed the LO transistors act as switches while the RF ones remain saturated, so $g_{m1} \ll g_{m2}$ and $g_{ds1} \gg g_{ds2}$. The total output noise of the mixer is given by

$$\overline{v_{m,out}^2} = B_{in} G_c^2 \overline{v_n^2} + B_{out} \overline{v_m^2} \tag{A.59}$$

where $\overline{v_n^2}$ is the noise PSD at the mixer RF input (the LO, being a square wave, is assumed to be noiseless), $\overline{v_m^2}$ is given by (A.58) and B_{in} and B_{out} are the input and output bandwidths, respectively. Substituting for $\overline{v_m^2}$, assuming $G_c \gg 1$ and simplifying using (A.53), we get

$$\begin{aligned} \frac{\overline{v_{m,out}^2}}{G_c^2} &= B_{in}\overline{v_n^2} + \frac{4k_B T B_{out} R_L \left(2 + \gamma_m \pi G_c\right)}{G_c^2} \\ &\approx B_{in}\overline{v_n^2} + 4k_B T B_{out} \left(\frac{\gamma_m \pi^2 V_{Lm} V_{DDm}}{P_{mixer}}\right) \end{aligned} \tag{A.60}$$

Calculating the SNR

Since we measure the difference between two amplitude levels to get our signal, the noise power present at each level will add (assuming they are uncorrelated) in our decision variable. Therefore the noise power in our decision variable is

$$\overline{v_{m,out}^2} = \overline{v_{m,out,0}^2} + \overline{v_{m,out,1}^2} \tag{A.61}$$

where $\overline{v_{m,out,0}^2}$ and $\overline{v_{m,out,1}^2}$ are the noise power present during transmission of a '0' and '1', respectively, and are given by (A.60). The noise produced by the mixer remains unchanged in both cases, but the noise at v_1 varies since both the PSD $\overline{v_n^2}$ and the bandwidth B_{in} are functions of m. The PSD is given by (A.57), and B_{in} by

$$B_{in} \approx \frac{\pi}{2} \times \frac{f_0}{\left(Q_1/\sqrt{1+m}\right)} = \frac{\omega_0 \sqrt{1+m}}{4Q_1} \tag{A.62}$$

where the factor of $\pi/2$ converts between the actual channel bandwidth $\sqrt{1+m}B$, where $B = f_0/Q_1$, and the noise bandwidth B_{in}. It arises because of the first-order (20dB/dec) roll-off of the impedance at v_1 on either side of ω_0. Assuming the conversion gain G_c of the mixer is large enough that succeeding stages to not degrade the SNR significantly, the output SNR that is used to make bit decisions is given by

$$
\begin{aligned}
SNR_{out} &\approx \frac{\left(G_c v_1 - G_c v_1/(1+m)\right)^2}{\overline{v_{m,out,0}^2} + \overline{v_{m,out,1}^2}} \\
&= \frac{\left(m_{eff} v_1\right)^2}{\left(\overline{v_{m,out,0}^2} + \overline{v_{m,out,1}^2}\right)/G_c^2}
\end{aligned}
\tag{A.63}
$$

The ratio of the energy per bit E_b to the noise PSD N_0 is a dimensionless quantity that is used to calculate the bit error rate (BER) of a digital communication system. It may be viewed as a ratio of power spectral densities (PSDs): that of the signal divided by that of the noise. When the noise PSD is uniform (white), this ratio is given by

$$
\frac{E_b}{N_0} = \left(\frac{E_b \times R}{N_0 \times B}\right)\frac{B}{R} = \frac{SNR}{(R/B)} \tag{A.64}
$$

where B is the bandwidth of the channel in Hz and R is the data rate in bits/sec. The dimensionless ratio R/B is known as the *link spectral efficiency*. To convert SNR to E_b/N_0, we replace the bandwidth B in an SNR formula with the data rate R. In our case, the relevant bandwidth is the output bandwidth B_{out} of the mixer. Therefore we set $B_{out} = R$ in (A.63) to get E_b/N_0.

By substituting for v_1 from (A.52) and using (A.57) and (A.62), we can now rewrite (A.63) as

$$
\frac{P_{PA}^2}{\left(P_{PA} + a + \frac{b}{P_{mixer}}\right)} = c \tag{A.65}
$$

where the quantities a, b and c are given by

$$
\begin{aligned}
a &= \left(\frac{V_{DDp} V_{Lp}}{R_1 Q_1^2}\right)\left(\frac{\sqrt{1+m}+1}{2\eta_r \gamma_p}\right)\frac{(1+m)^2}{\sqrt{1+m}+(1+m)^2} \\
b &= a\left(\frac{V_{DDm} V_{Lm}}{R_1 Q_1^2}\right)\left(\frac{4\pi\gamma_m}{\sqrt{1+m}+1}\right)\frac{R}{B} \\
c &= \frac{E_b}{N_0}\left(k_B T B\right)\frac{\left(\sqrt{1+m}+(1+m)^2\right)}{m^2}\left(\frac{\pi\gamma_p V_{DDp}}{\eta_r V_{Lp}}\right)
\end{aligned}
\tag{A.66}
$$

Optimum Power Consumption

We shall now calculate the minimum possible power consumption of the system given some BER requirement. We know that the BER, i.e. the probability of bit errors P_e, is a unique function of E_b/N_0, so we can invert the relationship and write $E_b/N_0 = f(P_e)$. This allows us to write our optimization problem as follows. Minimize the total power consumption

$$\begin{aligned} P_{tot} &= P_{PA} + P_{mixer} \\ &= P_{PA} + \frac{bc}{(P_{PA}^2 - cP_{PA} - ac)} \end{aligned} \tag{A.67}$$

where a, b and c are given by (A.66). In the general case minimizing P_{tot} by differentiating (A.67) w.r.t. P_{PA} and looking for stationary points leads to a fourth-order (quartic) equation. Quartic equations can be solved analytically but they are algebraically messy and do not provide much insight.

However, it is easy to solve the optimization problem analytically when $b/(P_{PA}P_{mixer})$ is much larger than the other two terms in the denominator of Equation (A.65). In this case (A.65) simplifies to

$$P_{PA}^2 P_{mixer} = bc \tag{A.68}$$

and the total power consumption is minimized when

$$\begin{aligned} \frac{P_{PA}}{P_{mixer}} &= 2 \\ P_{min} &= (bc)^{1/3}\left(2^{1/3} + 2^{-2/3}\right) = 1.89 \times (bc)^{1/3} \end{aligned} \tag{A.69}$$

We can rewrite bc as

$$bc = f(P_e)\frac{k_B T R}{m_{eff}^2}\left(\frac{2\pi^2 \gamma_m V_{Lm} V_{DDp}^2 V_{DDm}}{\eta_r^2 k^4 R_1^2 Q_1^4}\right) \tag{A.70}$$

Since $m_{eff} = k^2 Q_1 Q_2 / (1 + k^2 Q_1 Q_2)$, Equation (A.70) allows us to draw several conclusions. Since the total power consumption is $P_{min} = 1.89 \times (bc)^{1/3}$, when $m \ll 1$ and $V_{DDp} = V_{DDm} = V_{DD}$ we get $P_{min} \propto V_{DD} k^{-4/3} R_1^{-2/3} Q_1^{-2} Q_2^{-2/3}$. However, k is usually fixed by the link geometry. We conclude that in order to lower power consumption, we should lower the power supply voltage, increase the quality factors of the coils and lower the series resistance of the external (source) coil. However, it is much more important to increase the Q of the external coil since P_{min} is a much stronger function of Q_1 than Q_2. Finally, because k is based on near-field coupling between magnetic dipole fields, it decreases as d^{-3}, where d is the separation between the coils. Therefore $P_{min} \propto d^4$. This strong dependence on d explains why impedance modulation is only a viable technique for short-range links. In addition, notice that γ_p and V_{Lp} are not present in the expression for bc. When P_{mix} and P_{PA} are both small, k must be relatively large; in this case the power amplifier's noise is insignificant and system performance is dominated by the noise of the mixer.

The other case that is analytically tractable occurs when both a/P_{PA} and $b/(P_{PA} P_{mixer})$ are both much smaller than 1. In this case (A.65) simplifies to

$$P_{PA} \approx c \tag{A.71}$$

In this case, which occurs when k is small, $P_{PA} \propto m_{eff}^{-2}$, i.e. $P_{PA} \propto k^{-4}$ if $m_{eff} \ll 1$. In this regime the PA has to make v_1 large in order to get any signal out. This increases its output noise, making the noise and power consumption of the mixer largely irrelevant.

Inter-Symbol Interference

We shall assume that all the noise in the system is additive and Gaussian and that a sampling clock waveform can be recovered from the data. If there is no inter-symbol interference (ISI), the probability of bit errors P_e (i.e., the BER) is that expected from a synchronous matched-filter receiver using binary orthogonal signaling and is equal to [229]:

$$P_e = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) \Rightarrow \frac{E_b}{N_0} = 2 \times \left(\text{erfc}^{-1}\left(2P_e\right)\right)^2 \tag{A.72}$$

where $\text{erfc}(x)$ is the complementary error function. This expression is valid when the data rate is much smaller than the channel bandwidth, i.e., $R \ll B$. We refer to this as *power-limited* communication. When the data rate becomes comparable to or exceeds the channel bandwidth, we are *bandwidth-limited* and ISI increases the probability of bit errors. Physically, the finite channel bandwidth results in finite rise and fall times in the bit stream, thereby reducing the effective energy per bit. To increase the link spectral efficiency, we use non-return-to-zero (NRZ) data encoding. In NRZ data streams, transitions only occur when the value of the bit being transmitted changes. For simplicity, let us assume that the rise and fall times are smaller than one bit period $T = 1/R$. This assumption means that ISI only occurs between adjacent bits. The BER is given by

$$P_e = \sum_{i=0}^{1}\sum_{j=0}^{1} \alpha_{ij} P_{ij} \tag{A.73}$$

where α_{ij} is the probability of a $i \rightarrow j$ bit transition, and P_{ij} is the error probability associated with that transition. With NRZ encoding, the two P_{ii} terms are much smaller than the others since no transition occurs in the data stream if the value of the bit does not change from one bit period to the next. In the simplest case all bit transitions are equally likely, so $\alpha_{ij} = 1/4$, and error probabilities are symmetric, so $P_{01} = P_{10}$. Therefore we have

$$P_e \approx \frac{P_{01}}{2} \tag{A.74}$$

We can calculate P_{01} by recalling that the step-response of an under-damped second-order LTI system decays with an envelope proportional to $\exp\left(-\omega_0 t/\left(2Q\right)\right) = \exp\left(-\pi f_0/Q\right)$, where $\omega_0 = 2\pi f_0$ is its natural frequency and Q its quality factor. When transitions occur in the data waveform $s(t)$, the energy per bit is reduced by this finite decay time (see Figure A-34) and is given by

$$
\begin{aligned}
E_b &= \int_0^T s^2(t)dt \\
&= \frac{E_{b0}}{T}\int_0^T \left[1-\exp\left(-\pi Bt\right)\right]^2 dt \\
&= E_{b0}\left[1-\frac{\left(2-\exp\left(-\pi BT\right)\right)^2-1}{2\pi BT}\right]
\end{aligned} \tag{A.75}
$$

where B is the channel bandwidth, given by $f_0/\left(Q_1/\sqrt{1+m}\right)$ and $E_b \rightarrow E_{b0}$ as $BT \rightarrow \infty$, i.e., $R/B \rightarrow 0$ so that we are not bandwidth-limited. Therefore the effect of finite channel bandwidth is to reduce the signal energy per bit from E_{b0} to E_b, where E_b is given by (A.75).

Combining (A.72), (A.74) and (A.75), we can write E_{b0}/N_0 as a function of P_e. The result is

$$
\begin{aligned}
P_e &= \frac{1}{4}\mathrm{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) \\
\Rightarrow \frac{E_{b0}}{N_0} &\equiv f\left(P_e\right) = \frac{2\left(\mathrm{erfc}^{-1}\left(4P_e\right)\right)^2}{\left(1-\frac{\left(2-\exp\left(-\pi B/R\right)\right)^2-1}{2\pi B/R}\right)}
\end{aligned} \tag{A.76}
$$

This relationship can now be substituted into (A.70)[3]. Therefore we have solved our optimization problem. Given the center frequency, data rate and BER required, we can write down the minimum possible power consumption as a function of the coupling factor k, the power supply voltage and other constants. Even in the presence of ISI, power consumption is minimized when the quality factors Q_1 and Q_2 are maximized. This is because the increase in SNR at high Q more than compensates for the loss in SNR due to reduction in channel bandwidth and consequent ISI. Finally, it should be noted that it is possible to reduce the amount of E_b reduction due to ISI for a given R/B ratio by using non-rectangular pulses, such as raised cosines, for

[3]Since ISI was implicitly ignored while deriving (A.70), the E_b used in that formula is actually E_{b0}.

signaling.

Results and Discussion

The formulas derived in the previous section can now be examined further and plotted. Several features of the formulas are of interest. For example, we want to eliminate R_1, the series resistance of inductor L_1, since it is typically a parasitic component whose value is hard to predict. The simplest way to do this is to assume that the effect of inter-turn capacitances in L_1 is negligible and to use the well-known formula

$$Q_1 = \frac{\omega_0 L_1}{R_1} \tag{A.77}$$

We see that bigger coupling inductances are desirable since they result in larger values of Q at a given operating frequency. Practically, the largest value of L_1 that can be used decreases with f_0: parasitic capacitances cause larger inductors to self-resonate at lower frequencies. An useful metric that we shall use to characterize the power-efficiency of our communication system is the *energy consumed per bit* E_{bit}, which is defined as

$$E_{bit} = \frac{P_{min}}{R} \tag{A.78}$$

There are two additional constraints on our design. Both are nonlinear saturation mechanisms. The primary resonator amplitude v_1 cannot exceed either V_{DDp} or the linear range $2V_{Lm}$ of the mixer. In order to make these two saturation events occur simultaneously, we set $V_{DDp} = 2V_{Lm}$. As a result, we have

$$v_{1,max} = 2V_{Lm} \tag{A.79}$$

$$v_1 \leq 2V_{Lm} \Rightarrow P_{PA} \leq \frac{2V_{Lm}^2}{\eta_r R_1 Q_1^2} \tag{A.80}$$

The minimum possible mixer and PA power consumption and energy per bit predicted by (A.63) and (A.78) are shown in Figures A-35(a) and A-35(b), respectively as

a function of k for various data rates R. The maximum possible PA power consumption, $P_{PA,max}$ is also shown. The receiver can only be used when $k > k_{min}$, where k_{min} is the value of k at which the P_{PA} curve intersects the $P_{PA,max}$ line. The following parameter values were assumed: $V_{DDm} = 2.0\text{V}$, $V_{DDp} = 2V_{Lm} = 0.4\text{V}$, $V_{Lp} = 35\text{mV}$, $\gamma_p = \gamma_m = 1$, $\eta_r = 0.35$, $P_e = 10^{-4}$, $L_1 = 0.5\mu\text{H}$, $f_0 = 25\text{MHz}$ and $Q_1 = Q_2 = 25$. The curves saturate at high values of k since m_{eff}, unlike m, cannot exceed 1.

Interestingly, E_{bit} increases with R for low values of k. However, when k is large it slowly decreases with R, eventually saturating at a fixed value.

A.8.3 Asynchronous Demodulation

The receiver architecture described so far (shown in Figure A-33) places severe demands on the linearity of the mixer, since its input is the large RF voltage v_1 present on the source resonator. This limits the usability of this receiver to relatively large values of k. An alternative receiver architecture is shown in Figure A-36. In this receiver the envelope of the RF voltage v_1 is extracted before any further processing. The extracted envelope is then fed into a low-noise amplifier (LNA). The LNA is assumed to have enough gain for the noise of succeeding stages to have minimal effects on the received SNR. Since the LNA only sees the small modulated signal $m_{eff}v_1$, its linear range requirements are considerably reduced.

We immediately notice that the dead zone of the envelope detectors (EDs) will limit the performance of this architecture when v_1 is small, i.e. k is large. In a sense the two architectures are complementary: asynchronous (synchronous) demodulation is preferable when k is small (large). The analysis of this receiver architecture is similar to that described in Section A.8.2.

Envelope Detector Analysis

We consider the simple transistor-based envelope detector (ED) shown in Figure A-36. The circuit is an emitter follower. In the absence of a RF input signal, the current through the transistor is given by

$$I_{ED} = I_s \exp\left(\frac{V_{IN} - V_{REF}}{\phi_T/\kappa}\right) \tag{A.81}$$

where V_{REF} is the DC voltage at the output of the ED when no RF signal is present. The average current through the transistor must still be I_{ED} with an RF signal v_{RF} present at the input, since in periodic-steady-state operation the capacitor C_L cannot charge or discharge *on average*. Therefore we may write

$$\begin{aligned} I_{ED} &= \overline{I_s \exp\left(\frac{(V_{IN} + v_{RF}) - (v_{OUT} + V_{REF})}{\phi_T/\kappa}\right)} \\ &= I_s \exp\left(\frac{V_{IN} - V_{REF}}{\phi_T/\kappa}\right) \overline{\exp\left(\frac{v_{RF} - v_{OUT}}{\phi_T/\kappa}\right)} \end{aligned} \tag{A.82}$$

where the averaging is carried out over one RF cycle. By combining (A.81) and (A.82), we get

$$1 = \overline{\exp\left(\frac{v_{RF} - v_{OUT}}{\phi_T/\kappa}\right)} \tag{A.83}$$

Let us assume that the AC signal at the output ("the ripple", i.e. v_{out}) is much smaller than ϕ_T/κ. In that case we can bring the v_{OUT} term outside the averaging operation, since $\overline{\exp\left(\kappa v_{OUT/\phi_T}\right)} \approx \exp\left(\overline{\kappa v_{OUT}/\phi_T}\right) = \exp\left(\kappa V_{OUT}/\,\phi_T\right)$, where $V_{OUT} + V_{REF}$ is the steady-state (DC) value of the output voltage. Therefore we may write the ED's steady-state input-output relation as

$$V_{OUT} = \frac{\phi_T}{\kappa} \ln\left(\overline{\exp\left(\frac{v_{RF}}{\phi_T/\kappa}\right)}\right) \tag{A.84}$$

The average in (A.84) cannot be found analytically when the RF signal is sinusoidal, i.e. $v_{RF} = V_{RF} \sin(\omega_0 t)$. However, it is easily solvable if the input is a square wave. Suppose v_{RF} switches between $+V_{RF}$ and $-V_{RF}$ with 50% duty cycle. In this case

$$\begin{aligned} V_{OUT} &= \frac{\phi_T}{\kappa} \ln\left(\frac{1}{2}\exp\left(\frac{V_{RF}}{\phi_T/\kappa}\right) + \frac{1}{2}\exp\left(\frac{-V_{RF}}{\phi_T/\kappa}\right)\right) \\ &\approx V_{RF} - \frac{\phi_T \ln(2)}{\kappa} \end{aligned} \tag{A.85}$$

where we have assumed that $V_{RF} \gg \phi_T/\kappa$. When $V_{RF} < \phi_T \ln(2)/\kappa$, (A.85) predicts that $V_{OUT} \approx 0$. Therefore the dead zone of the ED is approximately $\phi_T \ln(2)/\kappa$. For sinusoidal inputs the dead zone can be found numerically; it is approximately $1.5\phi_T/\kappa$. In reality, however, the output of the ED is not quite zero for signals smaller than its dead zone, i.e., when $V_{RF} < \phi_T/\kappa$. In this case we can expand the exponential in (A.84) as a convergent series:

$$\begin{aligned} V_{OUT} &= \frac{\phi_T}{\kappa} \ln\left(\overline{\exp\left(\frac{v_{RF}}{\phi_T/\kappa}\right)}\right) \\ &= \frac{\phi_T}{\kappa} \ln\left(\overline{1 + \left(\frac{v_{RF}}{\phi_T/\kappa}\right) + \frac{1}{2}\left(\frac{v_{RF}}{\phi_T/\kappa}\right)^2 + \ldots}\right) \\ &\approx \frac{V_{RF}^2}{(4\phi_T/\kappa)} \end{aligned} \tag{A.86}$$

where several mathematical steps have been compressed into the last line. Firstly, we ignore terms higher than the second order. Secondly, we note that $\overline{v_{RF}} = 0$, which gets rid of the first-order (linear) term. Thirdly, we expand the logarithm by using the identity $\ln(1+x) \approx x$, which is valid when $|x| \ll 1$. Finally, we note that $\overline{v_{RF}^2} = V_{RF}/2$ when v_{RF} is a sinusoid. Thus, for input signals smaller than ϕ_T/κ, the ED produces an output that is proportional to the *square* of the input signal. In this region the ED is known as a square-law or energy detector.

We have shown that the ED output is approximately $V_{RF} - 1.5\phi_T/\kappa$ when $V_{RF} \gg \phi_T/\kappa$. Thus the ED output will follow small changes in V_{RF} one-for-one, i.e., when $m \ll 1$ the ED has a gain of 1 to both the amplitude-modulated data signal and the input noise. However, it also adds noise of its own to the output. The two sources

of noise are the transistor and the DC current source, both of which are assumed to generate shot noise with a PSD of $2qI_{ED}$. The output noise voltage PSD due to the ED

$$\begin{aligned} \overline{v_{n,ED}^2} &= (2qI_{ED} + 2qI_{ED}) \frac{1}{\overline{g_s^2}} \\ &= 4kT \left(\frac{\phi_T}{I_{ED}} \right) \end{aligned} \tag{A.87}$$

where $\overline{g_s} = I_{ED}/\phi_T$ is the average source conductance of the transistor. The total output noise voltage is

$$\overline{v_{n,tot}^2} = \overline{v_n^2} B_{in} + \overline{v_{n,ED}^2} B_{out} \tag{A.88}$$

where $\overline{v_n^2}$ and B_{in} are the noise PSD and bandwidth of the primary resonator (unchanged from the values calculated in the previous section), and B_{out} is the final noise bandwidth. Also, the power consumption of the ED is $P_{ED} = V_{DDe} I_{ED}$, where V_{DDe} is its power supply voltage.

Low Noise Amplifier Analysis

The output of the ED is fed into a low-noise amplifier (LNA). The gain of the LNA, G_L, is assumed to be high enough that the noise of succeeding stages has negligible impact on the noise figure of the receiver. We consider the LNA to be a single-transistor common-source amplifier with resistive load R_L. Its gain is given by $G_{LNA} = g_m R_L$, where $g_m = I_{LNA}/V_{Ll}$ is the transconductance of the transistor, I_{LNA} the DC bias current and V_{Ll} the linear range. Remembering that the power consumption of the LNA is $P_{ED} = V_{DDl} I_{LNA}$, where V_{DDl} is its power supply voltage, we can write

$$\frac{G_{LNA}}{R_L} = \frac{P_{LNA}}{V_{DDl} V_{Ll}} \tag{A.89}$$

The output noise voltage PSD produced by the LNA is given by

$$\begin{aligned}\overline{v_{n,LNA}^2} &= \left(\frac{4kT}{R_L} + 4kT\gamma_l g_m\right) \times R_L^2 \\ &= 4kTR_L\left(1 + \gamma_l G_{LNA}\right) \end{aligned} \tag{A.90}$$

The total noise voltage present at the output of the LNA is the sum of the noise contributions of the input, the ED and the LNA and is given by

$$\overline{v_{l,out}^2} = B_{in}G_{LNA}^2\overline{v_n^2} + B_{out}\left(G_{LNA}^2\overline{v_{n,ED}^2} + \overline{v_{n,LNA}^2}\right) \tag{A.91}$$

Calculating the SNR

The output SNR used to make bit decisions is given by

$$\begin{aligned} SNR_{out} &\approx \frac{\left(G_{LNA}v_1 - G_{LNA}v_1/(1+m)\right)^2}{\overline{v_{l,out,0}^2} + \overline{v_{l,out,1}^2}} \\ &\approx \frac{\left(m_{eff}v_1\right)^2}{2\overline{v_{l,out}^2}/G_{LNA}^2} \end{aligned} \tag{A.92}$$

where we have assumed (for simplicity) that $m \ll 1$, so that $\overline{v_{l,out,0}^2}$ and $\overline{v_{l,out,1}^2}$, the output noise during the transmission of '0' and '1', respectively, are approximately equal. Notice the similarity of this equation to (A.63). As before, we can set $B_{out} = R$ to convert the left-hand side of this equation into E_b/N_0. By substituting for v_1 from (A.52), assuming $G_{LNA} \gg 1$ and using (A.91) and (A.62), we can now rewrite (A.92) as

$$\frac{P_{PA}^2}{\left(P_{PA} + a + \frac{b}{P_{LNA}} + \frac{d}{P_{ED}}\right)} = c \tag{A.93}$$

where the quantities a, b, c and d are given by

$$
\begin{aligned}
a &= \left(\frac{V_{DDp}V_{Lp}}{R_1 Q_1^2}\right)\left(\frac{1}{2\eta_r \gamma_p}\right) \\
b &= a\left(\frac{\gamma_l V_{DDl} V_{Ll}}{R_1 Q_1^2}\right)\frac{2R}{\pi B} \\
d &= a\left(\frac{V_{DDe}\phi_T}{R_1 Q_1^2}\right)\frac{2R}{\pi B} \\
c &= \frac{E_b}{N_0}\left(k_B T B\right)\frac{2}{m_{eff}^2}\left(\frac{\pi\gamma_p V_{DDp}}{\eta_r V_{Lp}}\right)
\end{aligned}
\tag{A.94}
$$

Optimum Power Consumption

We want to minimize the total power consumption, i.e. $P_{tot} = P_{PA} + P_{ED} + P_{LNA}$, subject to the constraint imposed by (A.94). Unlike in the previous section, this optimization problem is under-constrained since we have three unknowns (P_{PA}, P_{ED} and P_{LNA}) and only one constraint relating them. It can be solved numerically if the various parameter values are specified.

A heuristic that simplifies the optimization considerably can be obtained by recognizing that P_{LNA} and P_{ED} occur in a symmetric fashion in (A.94). Let us keep P_{PA} fixed for now, and minimize the sum $P_{LNA} + P_{ED}$ while keeping $(b/P_{LNA} + d/P_{ED})$ constant. The result is

$$
\frac{P_{LNA}}{P_{ED}} = \sqrt{\frac{b}{d}} = \sqrt{\frac{\gamma_l V_{DDl} V_{Ll}}{V_{DDe}\phi_T}}
\tag{A.95}
$$

In practice this ratio is likely to be close to 1, since typically we have $V_{DDl} = V_{DDe}$ and $V_{Ll} \approx \phi_T/\kappa$. Our optimization problem then simplifies to the following: minimize

$$
P_{tot} = P_{PA} + \left(1 + \sqrt{d/b}\right) P_{LNA}
\tag{A.96}
$$

subject to the constraint that

$$
\frac{P_{PA}^2}{\left(P_{PA} + a + \frac{b+\sqrt{bd}}{P_{LNA}}\right)} = c
\tag{A.97}
$$

This problem is very similar to the one we solved for the synchronous demodulation receiver, and can be solved in a similar way. We get the same limiting behavior as before: if P_{LNA} is small, so that the $\left(b+\sqrt{bd}\right)/P_{LNA}$ term is much larger than both a and P_{PA}, we have

$$P_{PA}^2 P_{LNA} = \left(b+\sqrt{bd}\right) c \tag{A.98}$$

Similarly, when P_{PA} is much larger than both a and $\left(b+\sqrt{bd}\right)/P_{LNA}$, we have

$$P_{PA} \approx c \tag{A.99}$$

Results and Discussion

The effects of ISI on this system can be modeled exactly as before, leading to the same expression, i.e. (A.76). The additional constraints on the receiver are that v_1 cannot exceed V_{DDp}, or be smaller than V_{dead}, the dead zone of the ED ($V_{dead} \approx 1.5\phi_T/\kappa$ in our case). Therefore we have

$$V_{dead} \leq v_1 \leq V_{DDp} \tag{A.100}$$

As a result

$$\frac{V_{DDp} V_{dead}}{\eta_r R_1 Q_1^2} \leq P_{PA} \leq \frac{V_{DDp}^2}{\eta_r R_1 Q_1^2} \tag{A.101}$$

In other words the ratio of the maximum and minimum values of P_{PA} is V_{DDp}/V_{dead}.

The minimum possible mixer and PA power consumption and energy per bit predicted by (A.97) are shown in Figures A-37(a) and A-37(b), respectively as a function of k for various data rates R. The maximum and minimum possible PA power consumption, $P_{PA,max}$ and $P_{PA,min}$, are also shown. The receiver can only be used when $k_{min} < k < k_{max}$, where $k_{min}(k_{max})$ is the value of k at which the P_{PA} curve intersects the $P_{PA,min}(P_{PA,max})$ line. The following parameter values were assumed: $V_{DDl} = V_{DDe} = V_{DDp} = 2.0\text{V}$, $V_{Lp} = V_{Ll} = 35\text{mV}$, $V_{dead} = 50\text{mV}$, $\gamma_p = \gamma_l = 1$,

$\eta_r = 0.35$, $P_e = 10^{-4}$, $L_1 = 0.5\mu$H, $f_0 = 25$MHz and $Q_1 = Q_2 = 25$. The curves saturate at high values of k since m_{eff}, unlike m, cannot exceed 1.

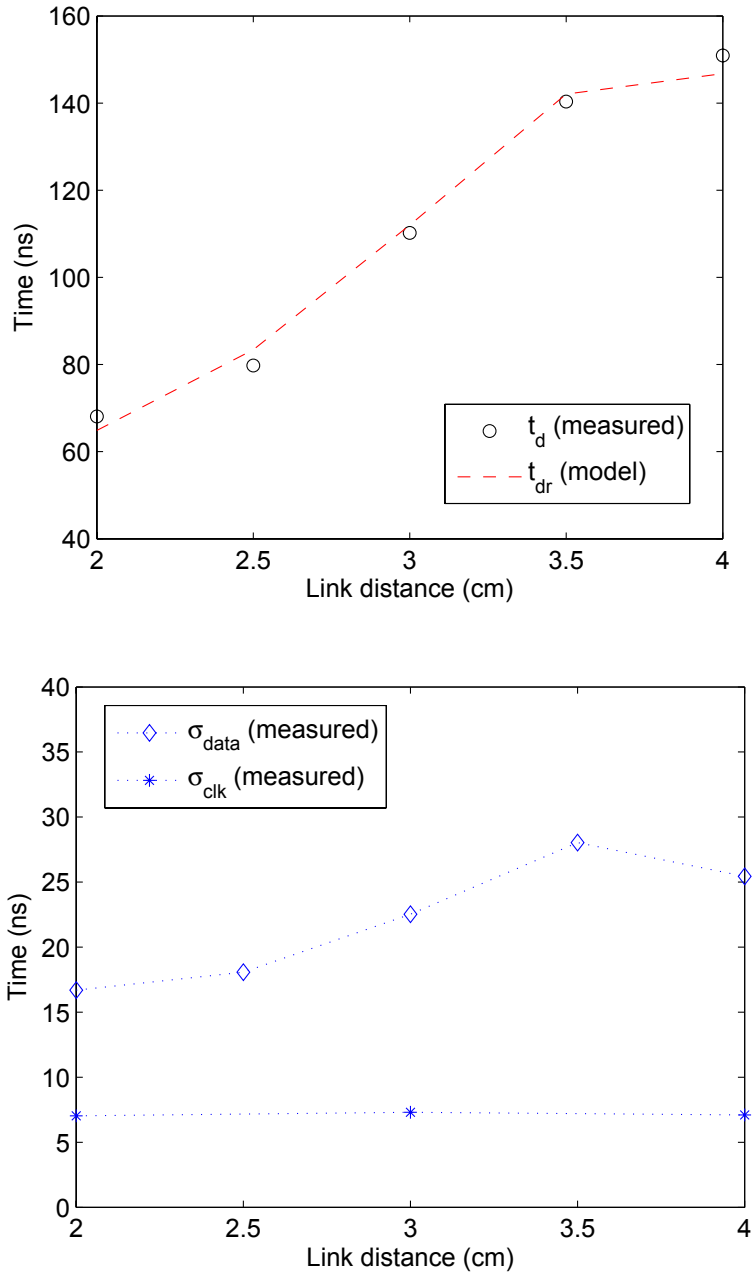


Figure A-23: Measured delay time of rising data edges (t_{dr}), rms data jitter (σ_{data}) and rms clock jitter (σ_{clk}) as a function of the link distance. Fits to a theoretical model of t_{dr} are also shown.

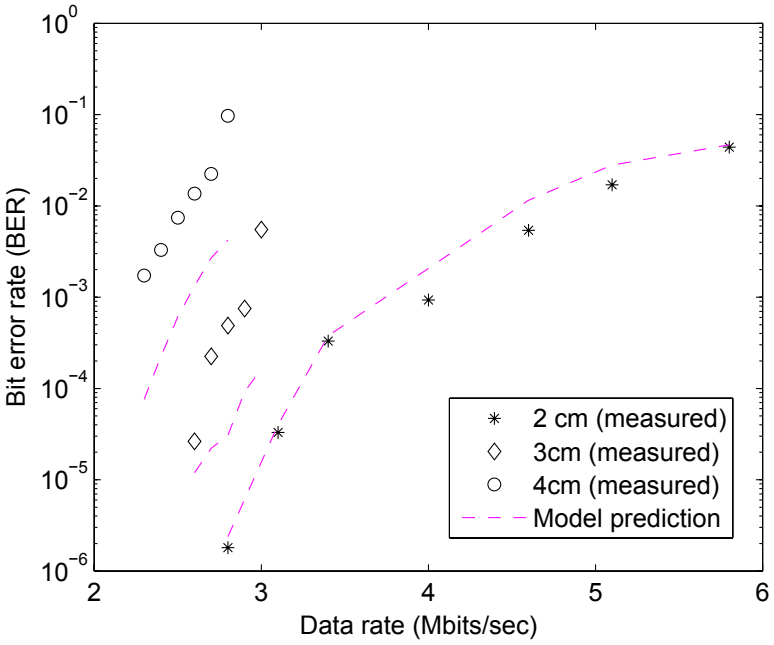


Figure A-24: Measured bit error rate (BER) for the uplink with the coils 2, 3 and 4cm apart, compared with predictions from our model.

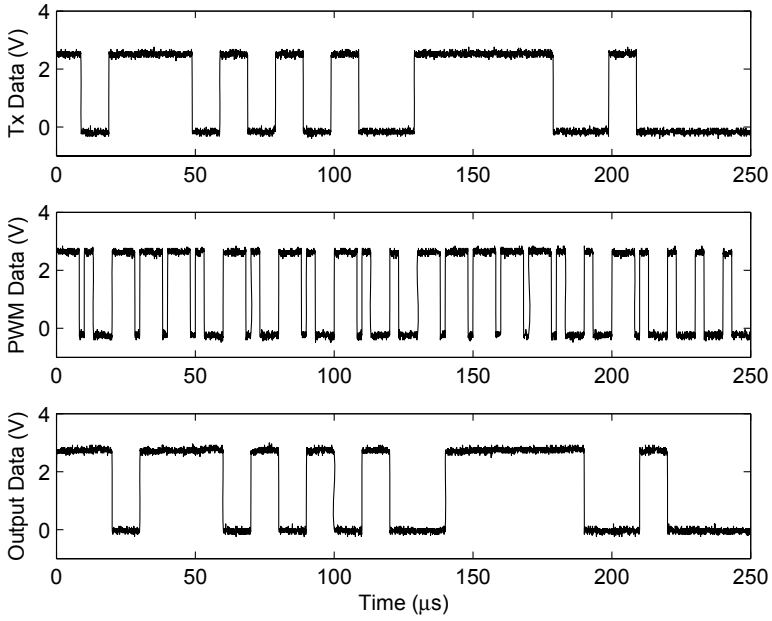


Figure A-25: Downlink data transmission at 200kbps with the coils 2cm apart.

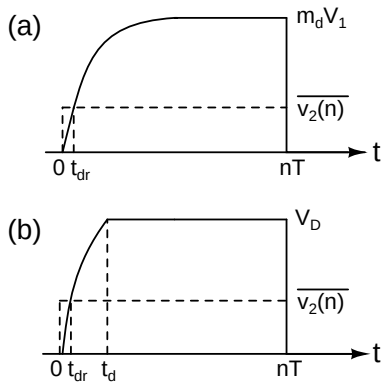


Figure A-26: Envelopes of the voltage across the internal resonator, (a) without the diode clamp and (b) with the diode clamp.

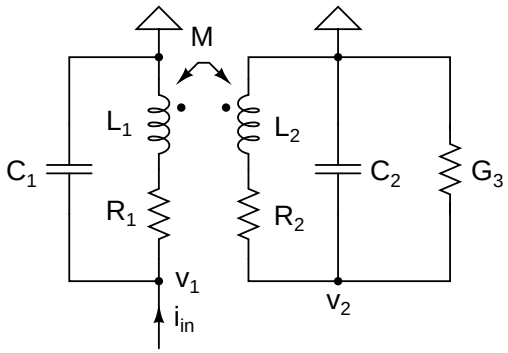


Figure A-27: Model of inductively-coupled resonators used for analysis.

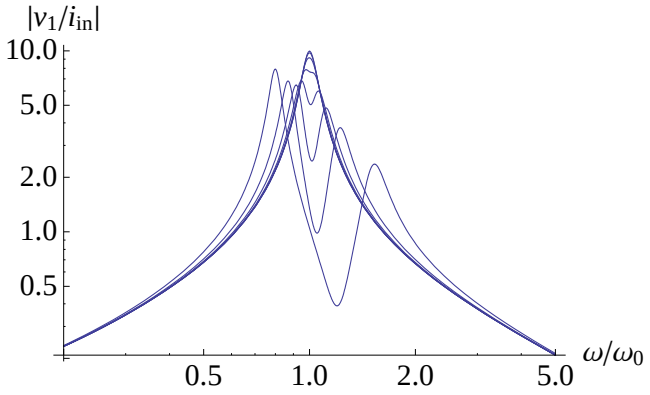


Figure A-28: Driving-point impedance of the coupled-resonator network with identical primary and secondary resonators, $Q_1 = Q_2 = 10$ and $G_3 = 0$ for logarithmically-spaced values of k between 0.01 and 0.5.

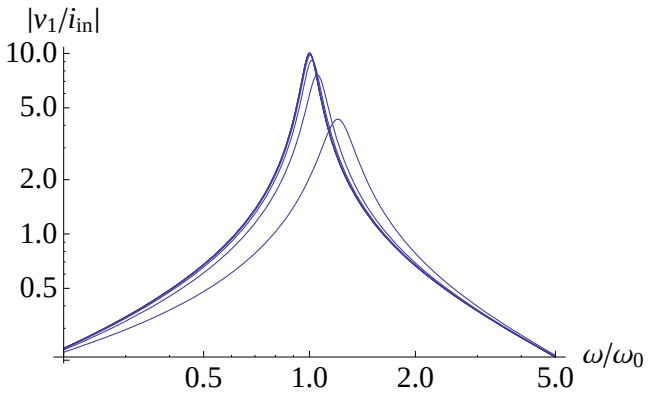


Figure A-29: Driving-point impedance of the coupled-resonator network with identical primary and secondary resonators, $Q_1 = Q_2 = 10$ and $G_3 = 10$ (effectively a short) for logarithmically-spaced values of k between 0.01 and 1.

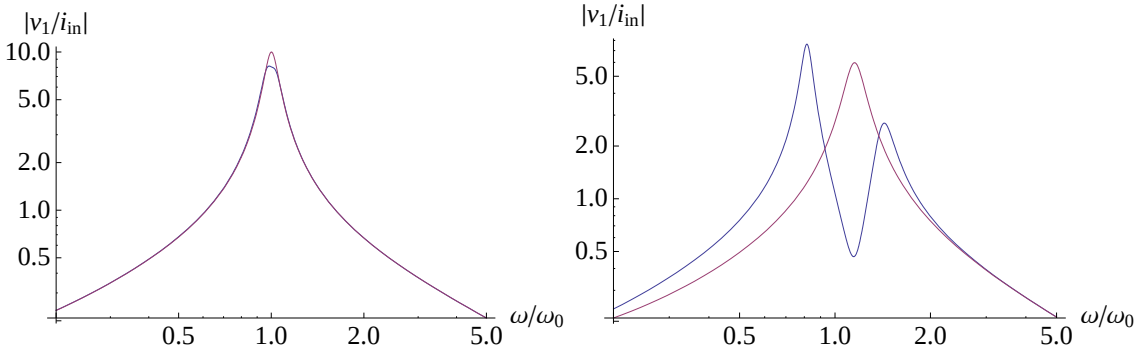


Figure A-30: Driving-point impedance of the coupled-resonator network with identical primary and secondary resonators, $Q_1 = Q_2 = 10$ as G_3 is switched between 0 (open, blue) and 10 (effectively a short, red) for $k = 0.05$ (left) and 0.5 (right).

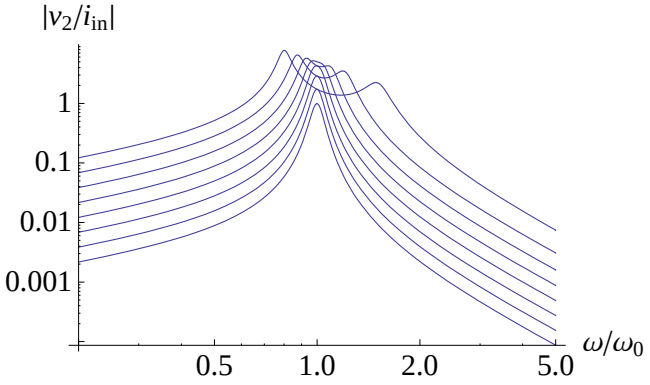


Figure A-31: Transfer impedance of the coupled-resonator network with identical primary and secondary resonators, $Q_1 = Q_2 = 10$ and $G_3 = 0$ for logarithmically-spaced values of k between 0.01 and 1.

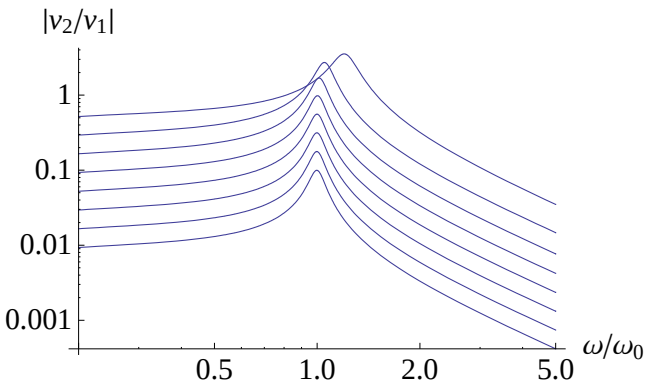


Figure A-32: Voltage gain of the coupled-resonator network with identical primary and secondary resonators, $Q_1 = Q_2 = 10$ and $G_3 = 0$ for logarithmically-spaced values of k between 0.01 and 1.

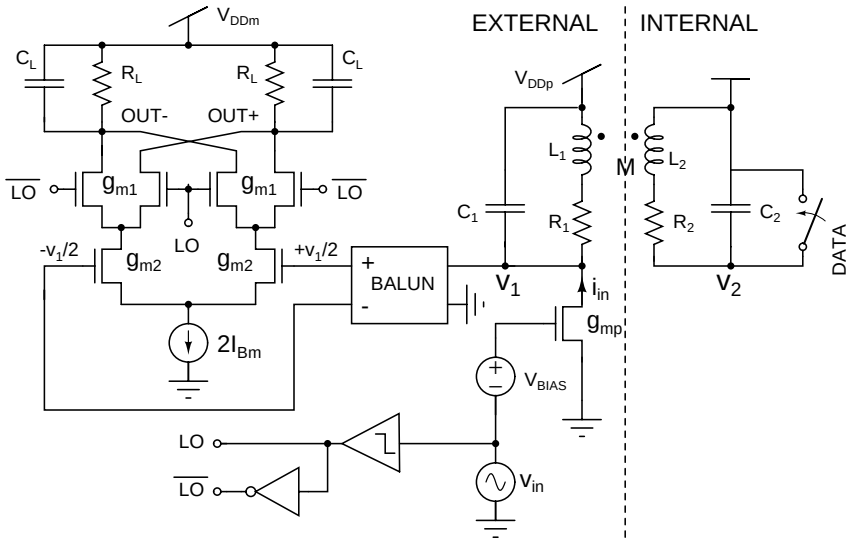


Figure A-33: Model of the impedance-modulated wireless data link analyzed in this section.

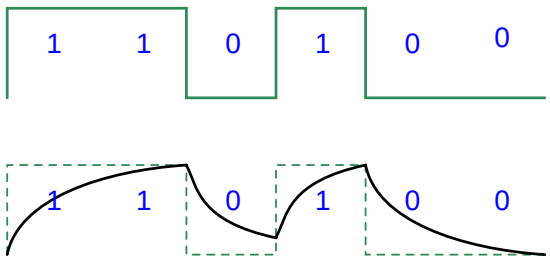


Figure A-34: The effects of inter-symbol interference (ISI) on recovered baseband data. Waveforms without ISI (top) and with ISI (bottom) are shown.

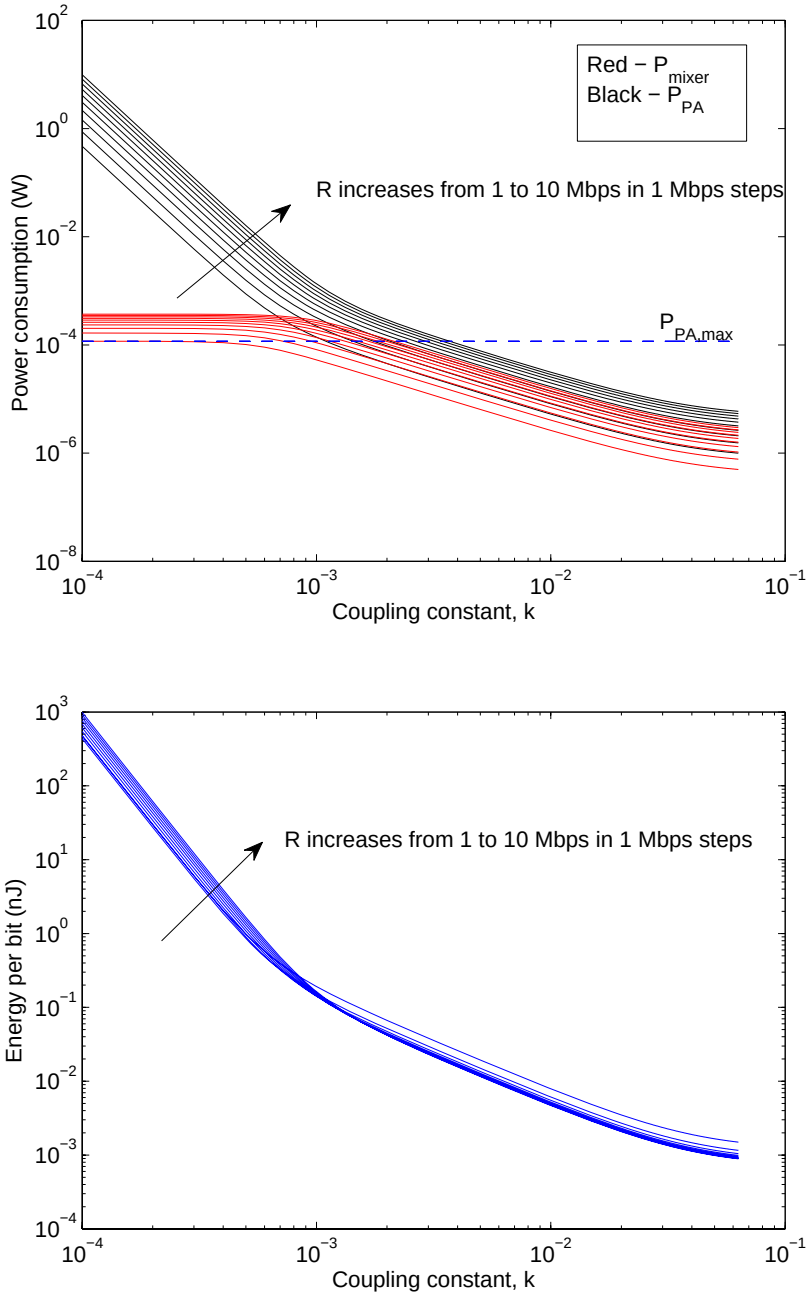


Figure A-35: Theoretically predicted minimum (a) power consumption and (b) energy dissipated per bit for the synchronous receiver, plotted as a function of the coupling constant k for various data rates R.

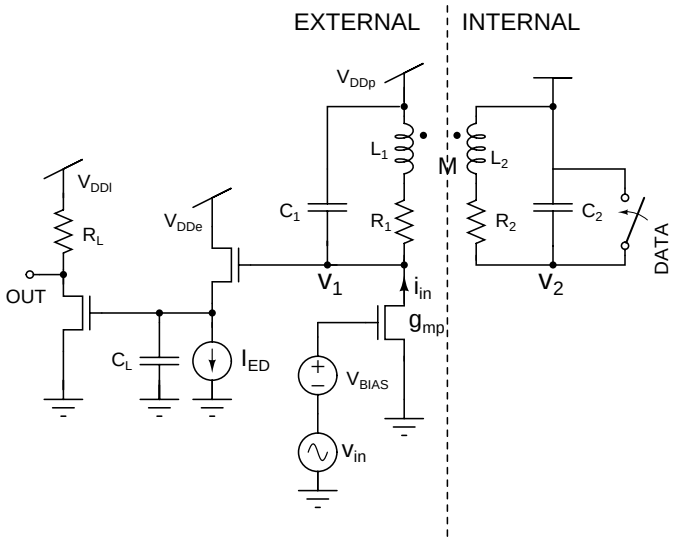


Figure A-36: Model of the impedance-modulated wireless data link analyzed in this section.

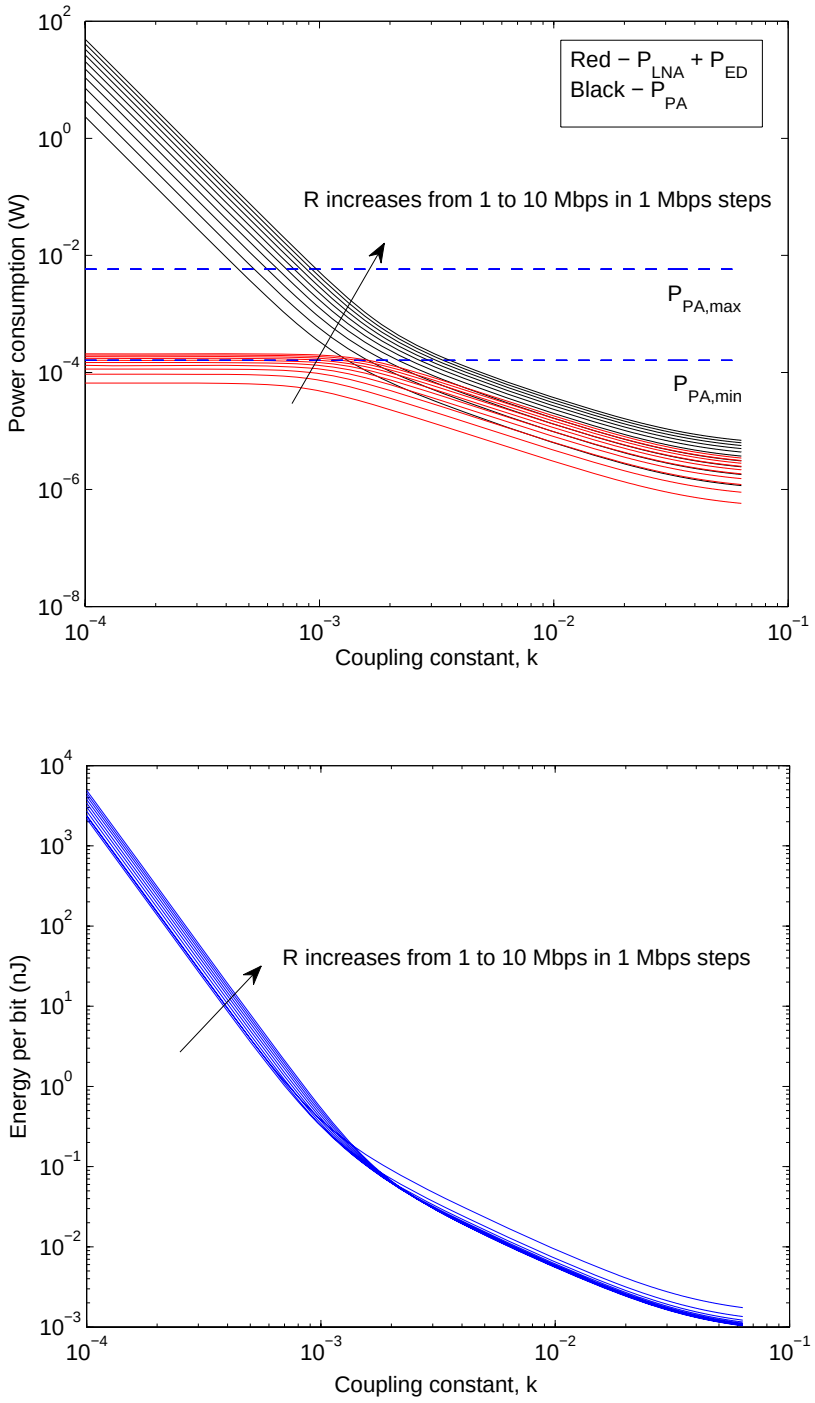


Figure A-37: Theoretically predicted minimum (a) power consumption and (b) energy dissipated per bit for the asynchronous receiver, plotted as a function of the coupling constant k for various data rates R.

Appendix B

A Programmable Event Processor for Body Sensor Networks

In this chapter we describe a low-power integrated circuit for use in wearable patient-monitoring systems. The chip contains four independent programmable channels that can be used to create events from biomedical signals and/or actuate outputs. Channel gain, event threshold, event duration and event hold time are programmable and channel outputs can be combined using a programmable logic array (PLA). Experimental results with phono-cardiogram (PCG) and photo-plethysmogram (PPG) signals are presented along with a scheme for patient localization using audio in the case of an alarm. This chapter was based on work performed in collaboration with Lorenzo Turicchia.

B.1 Introduction

With the world population ageing rapidly, providing care for the elderly is becoming an increasingly important problem. For instance, more than 5,000 people experience Sudden Cardiac Arrest (SCA) every week in the United States alone. The only definitive treatment for SCA is early defibrillation: no more than 6 minutes from arrest to first shock. The chance for survival drops 10% per minute without defibrillation, and today, over 95% of SCA victims die. Since automatic defibrillators are increasingly

available, pervasive monitoring of those at risk can save many lives [53]. Infants constitute another segment of the population where pervasive monitoring could enable rapid responses to life-threatening situations. In the United States alone, approximately 2,000 infants die each year from Sudden Infant Death Syndrome (SIDS). Since slow heart-rate (bradycardia) is an important indicator of SIDS [200], early detection of bradycardia in infants may save many lives each year.

Wireless networks of context-aware body-mounted sensors have come into prominence recently for pervasive patient monitoring [126, 320]. However, to be effective, monitoring systems should be unobtrusive, robust, and low-cost. One way to lower the size, weight and cost of the system is to eliminate batteries and rely entirely on harvested ambient energy, in this case RF. This is the approach adopted by passive radio-frequency identification (RFID) tags [72], which face similar size and cost constraints. We describe progress on a low-cost wearable tag that can be attached to the skin and used for such applications. The tag monitors multiple biomedical signals and allows the patient to be localized if necessary. The signals that we want to monitor include heart sounds (PCG), electrical heart signals (EKG), blood oxygen saturation (PPG), respiratory sounds [208], blood pressure, and body temperature. However, maintaining reliable and robust operation is challenging for any biomedical device, but particularly problematic when a reliable local energy source, such as a battery, is absent. We describe algorithms that detect malfunctioning tags, thereby improving the reliability of our results.

Mechanical coupling causes the skin to vibrate in synchrony with motions of the heart. Such vibrations generate acoustic pressure waves, referred to as heart sounds, that can be picked by microphones or by trained human ears using stethoscopes. Such sounds provide important information about cardiac health. For example, consider the characteristic *lub/dub* rhythm present within each heart beat, referred to as the first heart sound (S_1) and second heart sound (S_2), respectively. These sounds are produced by the turbulent flow of blood against the closed atrioventricular and semilunar valves, respectively, and are thus sensitive to the condition of these valves. The resting value and short-term variability of the heart rate, which are both of clin-

ical importance [42], can also be measured from such acoustical recordings, known as phono-cardiograms (PCGs).

Continuous monitoring and analysis of heart sounds is likely to provide important health benefits because these sounds contain large amounts of useful information. In this instantiation, we only use sounds, not electrical signals, for heart monitoring (i.e., the PCG instead of the EKG). The PCG has three advantages over the EKG: Firstly, it does not require any electrical contact with the body - something which is often difficult to obtain with dry skin. Secondly, it can be implemented in a very low-power fashion with a low-cost microphone since the heart is easy to sense acoustically as it is the loudest organ in the body. Reducing power consumption becomes extremely important for systems such as ours that rely on energy-harvesting, since the available ambient energy density is usually orders of magnitude lower than that present within a battery. Thirdly, it requires little maintenance, unlike the EKG.

Our tag uses multiple sensors to generate three types of alarm: disconnection from the body, device malfunction, and patient emergency. For example, the simplified version shown in Figure B-1 has two microphones, one facing up (away from the body) and the other facing down. The downward-facing microphone monitors heart sounds, while the upward-facing one is usually switched off to save power. It is turned on only when the downward-facing microphone does not detect any heart sounds and a disconnection or patient emergency is suspected. If both microphones now pick up similar environmental sounds, a 'disconnection alarm' is generated since it is probable that the tag is no longer in proximity to the skin. A 'patient emergency alarm' is generated if the downward-facing microphone does not pick up environmental sounds, but the upward-facing one does, since in this case it is likely that the tag is still attached and the heart has stopped. If neither microphone picks up any sounds, the tag is probably malfunctioning; therefore a 'device malfunction alarm' is generated.

Each tag contains a unique identification code and powers up using harvested RF power [275]. A fixed base station communicates with multiple tags and decides, based on transmitted patient data, if an alarm should be triggered. Such a system will be useful for hospitals, facilities that care for infants and the elderly, and also ordinary

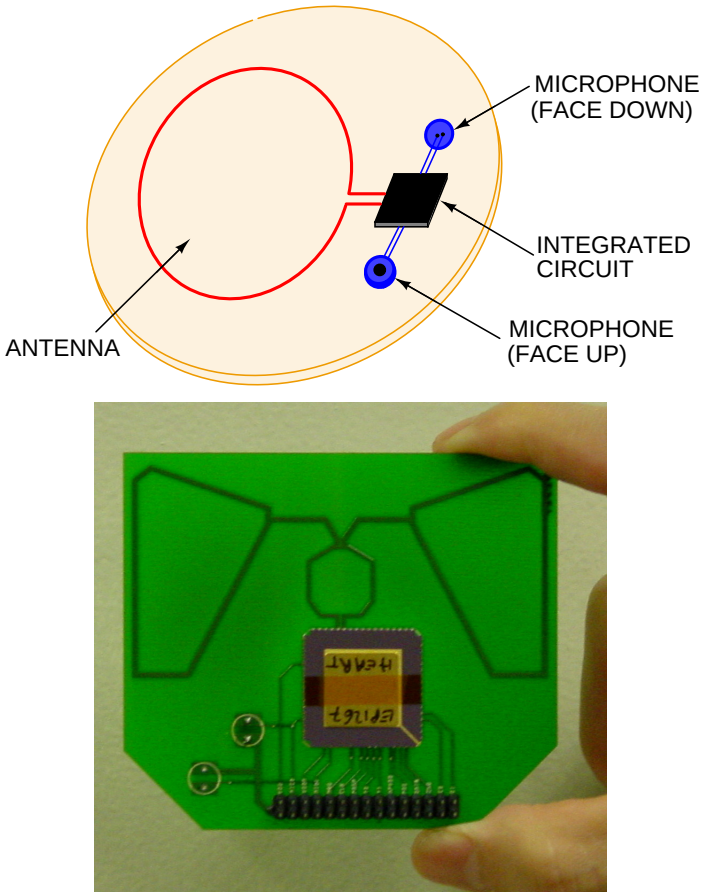


Figure B-1: A simplified version of our system with two microphones and an antenna attached to a flexible, adhesive surface. Conceptual view, attached to a flexible, adhesive surface (top), and photograph of the actual experimental 800MHz prototype that was tested (bottom). We estimate that the area of the prototype can easily be halved by using a smaller package for the chip and a more optimized antenna. In addition, the pins at the bottom of the prototype were for testing purposes and are not necessary for a final commercial system.

homes. In order to quickly cover a large fraction of the population at risk we need to keep the tag low-cost (ideally, less than $2 each when manufactured in volume), disposable, small and easy to use. In this article we describe the low-power custom integrated circuit that forms the central component of the patient-monitoring tag and demonstrate its power harvesting, sensing and actuation capabilities. We do not describe system-level issues like polling of multiple tags, software and deployment.

In this paper we describe a low-power chip designed for use in pervasive medical monitoring applications. Section B.2 describes the design of the system in detail, Sec-

tion B.3 presents experimental results, and Section B.4 summarizes our conclusions.

B.2 System Design

A block diagram of the overall chip is shown in Fig. B-2. It was designed to be extremely low power by incorporating only minimally acceptable amounts of computation and signal processing; most complexity is transferred to the fixed base station. The chip can harvest radiated RF power, making a low-cost battery-free tag possible. An efficient two-stage CMOS rectifier [186] is connected to an external loop antenna. The input capacitance of the chip, C_L resonates with the inductive input reactance of the antenna at the operating frequency. The resultant L-type impedance match provides passive voltage gain that reduces the amount of RF power needed to overcome the dead-zone of the rectifier, thereby increasing operating range. The first rectifier stage is designed to have low output impedance since it powers up external sensors, which typically consume much more power than the chip itself. The second stage, which provides a higher-impedance output, is used to power up the chip. Over-voltage protection circuits at the power supply and RF input nodes prevent damage due to large RF amplitudes.

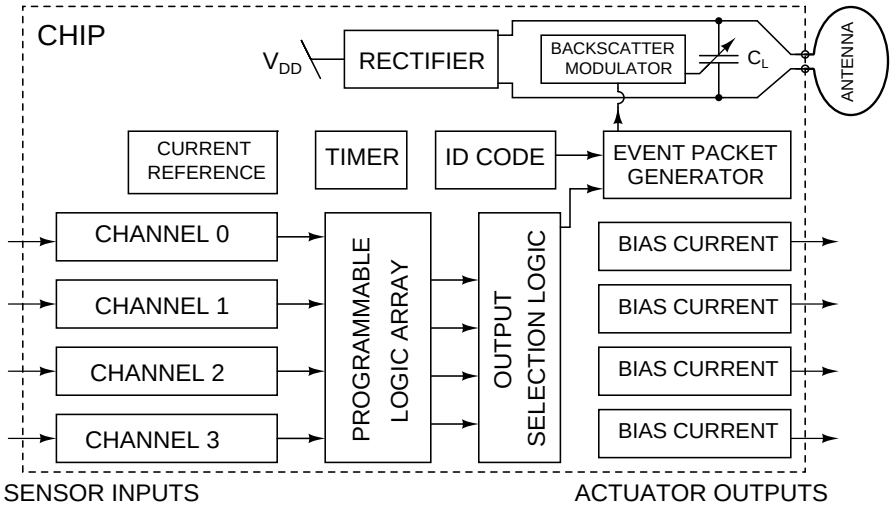


Figure B-2: Block diagram of the low-power patient-monitoring chip.

Path loss models predict the fall-off of radiated power density P_r (in W/m^2) with

distance D from the transmitter. A simple version commonly used for modeling indoor environments recognizes two zones: $P_r \propto D^{-n_1}$ for $D \leq D_0$, and $P_r \propto D^{-n_2}$ for $D > D_0$, where D_0, n_1 and n_2 are constants [161]. Typically $n_1 \approx 2$, the free-space value, and n_2 varies between 2.5 and 4. The value of n_2 exceeds 2 because of absorption and reflection of the RF by environmental obstacles, such as furniture and people. We used the following conservative values: $D_0 = 5\text{m}$, $n_1 = 2$, $n_2 = 3.5$. Combining the predicted path loss with the rectifier model proposed in [186] gives us Figure B-3, where P_A is the RF power that can be harvested at different distances from the transmitter.

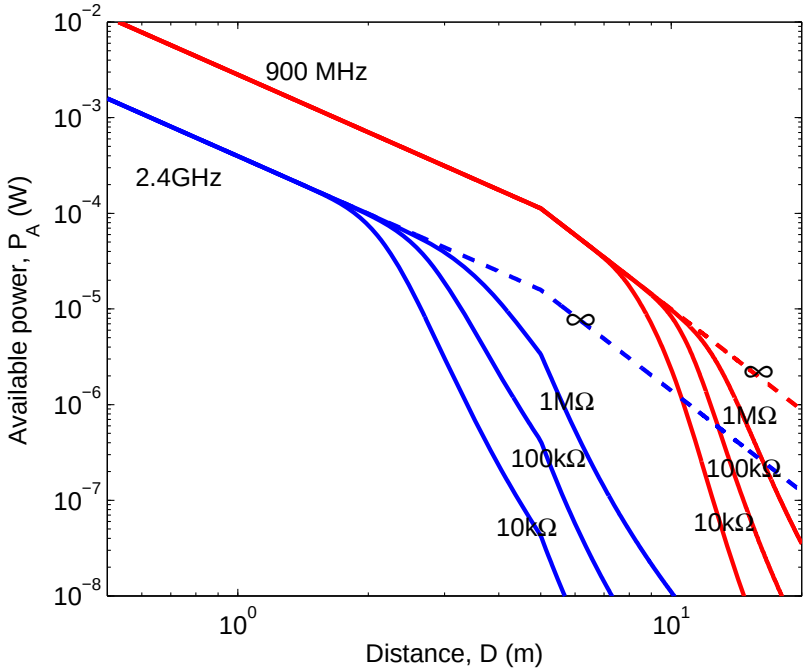


Figure B-3: Harvested RF power available as a function of distance from the transmitter for different load resistances at 900MHz and 2.4GHz.

Figure B-3 assumes that the equivalent isotropic radiated power (EIRP) is 4W, which is the maximum allowed in the United States for radio-frequency identification (RFID) applications. It shows P_A as a function of D at two popular RFID frequencies: 900MHz and 2.4GHz. The main reason for going to higher operating frequencies is to reduce the physical size of the antenna. Loop antennas are normally operated at their first resonant frequency. At this frequency the circumference of the loop is half the

wavelength. Therefore a single-turn circular loop has a diameter of 5.3cm at 900MHz and 2.0cm at 2.4GHz. Multiple-turn loops can be used to reduce antenna size at the cost of increased fabrication complexity.

The various curves in Figure B-3 correspond to different load resistances R_L driven by the rectifier. They decrease rapidly at large distances because the received RF amplitude becomes smaller than the rectifier's dead zone. The load resistance is usually dominated by the power consumed by off-chip sensors and not the chip itself. For example, a microphone biased at 30μA and 0.5V (typical values used in our experiments) dissipates 15μW, corresponding to an effective load driven by the rectifier of $R_L = 16.7\text{k}\Omega$ [186]. Figure B-3 then predicts an operating range of approximately 12m at 900MHz and 3m at 2.4GHz. In practice the reliable operating range will be somewhat smaller because some tags will be mistuned by their proximity to conductive and dielectric surfaces. In addition, we have to allow for transient drops in received RF power level (fades), which are ubiquitous in indoor environments because the received signal is the superposition of multiple waves with time-varying amplitude and phase. Nevertheless, a single base-station operating at 900MHz is sufficient for a moderately-sized room.

Our chip contains four independent channels, that can be used to interface to various types of sensors. A block diagram of a single channel is shown in Fig. B-4. The outputs of each channel are digital spikes, i.e., 'event' signals. These signals can be combined in a flexible way using a Programmable Logic Array (PLA) that can implement a variety of Boolean logic functions. Our PLA is a four-input four-output design with a 8×8 AND plane and a 4×8 OR plane implemented using static logic gates. The PLA allows us to implement any of the 2^{2^4} possible logic functions of four inputs for any of its four outputs in a programmable fashion. These outputs can be monitored individually, allowing us to implement rudimentary sensor-fusion algorithms that combine the outputs of multiple channels. Programmable output selection logic multiplexes the four PLA outputs into a single signal that is transmitted to the base station as "event packets" containing the chip identification code and time stamps. Data is transmitted using backscatter modulation [72]: a

100fF capacitor is added and subtracted from C_L to change the amount of RF power scattered by the tag. Backscatter modulation is popular in passive RFID systems because all the complexity and power consumption is pushed to the base station; the tag remains simple and low-power. An incoherent on-off keying (OOK) receiver will be added to later versions of the chip to allow wireless reprogramming of chip parameters.

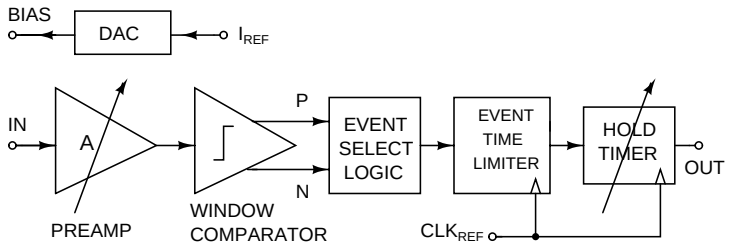


Figure B-4: A single signal processing channel.

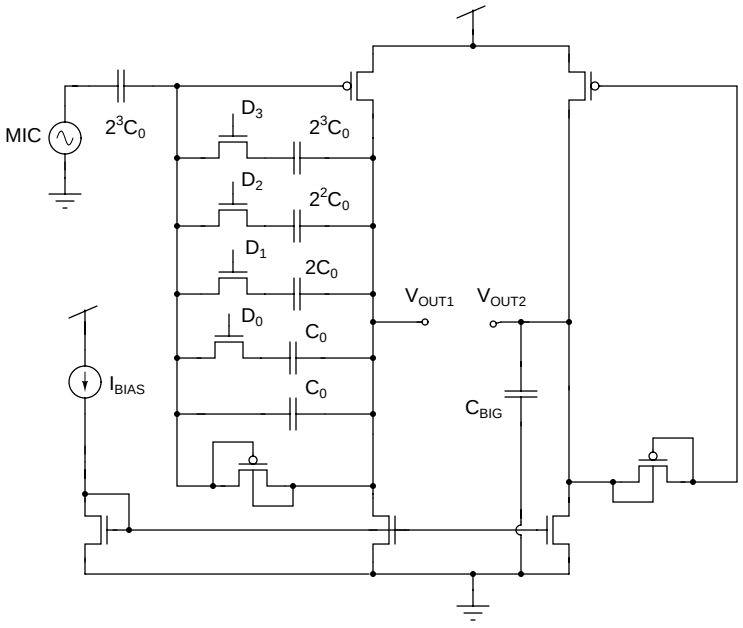


Figure B-5: Pseudo-differential programmable gain preamplifier with DC offset rejection.

The preamplifier, shown in Fig. B-5, consists of a common source stage with capacitive feedback that can be varied to set the gain. The feedback capacitor can be varied from C_0 to $16C_0$ using a binary-weighted array. PMOS input transistors are used because they have lower $1/f$ noise. A PMOS adaptive element [55] that acts as

an extremely large resistor R_f for small input signals is also connected in feedback to set the DC voltage at the input equal to that at the output, resulting in DC rejection, i.e., a band-pass transfer function with a very low cut-in frequency (typically <1Hz), given by

$$A(D) = \frac{v_{out}}{v_{in}} = -\frac{2^N}{D+1}\left(\frac{s\tau_1}{s\tau_1+1}\right)\left(\frac{1}{1+s\tau_2}\right) \tag{B.1}$$

where in this case, $N = 4$, D, the digital code used to program the gain, varies from 0 to 15, $\tau_1 = DC_0R_f$ and $\tau_2 = \frac{(D+2^N)C_0}{g_m}$, with g_m being the transconductance of the input transistor. Our design uses a nominal bias current of 10nA and $C_0 = 0.5$pF, resulting in a bandwidth that decreases from 12KHz to 6KHz as the gain increases from 1 to 16. Preamplifiers from different channels can be cascaded together if more gain is needed. A matched copy of the amplifier (minus the capacitors) is used to determine the quiescent operating point V_{OUT2}. The resultant pseudo-differential output voltage $V_{OUT1} - V_{OUT2}$ is fed into the comparator.

The comparator generates events whenever the output voltage of the amplifier differs from its quiescent value by more than a fixed threshold voltage V_{th}, i.e. $|V_{OUT1} - V_{OUT2}| > V_{th}$. There are two types of events: positive-going, when $V_{OUT1} > V_{OUT2}$, and negative-going, when $V_{OUT1} < V_{OUT2}$. Our comparator operates in continuous time and consists of two high-gain differential amplifiers. The two input terminals of each amplifier are reversed relative to each other so that one produces positive-going, and the other, negative-going, events. Each amplifier consists of a differential first stage with built-in offset to create V_{th}, and a common source second stage. The offset is created by making the W/L ratio of one of the input differential pair transistors S times larger than the other. Assuming subthreshold operation [199], this results in a threshold voltage of $V_{th} = (\phi_T/\kappa)\ln(S)$, where ϕ_T is the thermal voltage and κ is the subthreshold exponential constant. In our case, $S = 10$ and $\kappa \approx 0.75$, giving $V_{th} \approx 80$mV. The input-referred threshold for detecting an event thus decreases from 80mV to 5mV as the preamplifier gain increases from 1 to 16. The well terminals of the input PMOS differential pair transistors were forward biased by 0.3V using a

level-shifter to increase the input common-mode range (ICMR) of the comparator at low supply voltages. This strategy lowers the PMOS threshold voltage by about 100mV, increasing the ICMR by a similar amount.

Spike selection logic is connected to the comparator output in each channel and allows only positive or negative-going spikes, both, or neither to be detected. This combinational block is followed by a pulse-stretcher circuit that adds hysteresis in the time domain to prevent multiple comparator transitions due to noise when an event is detected. It also ensures that output spikes always last long enough for at least one complete data packet to be broadcast during every spike. The pulse-stretcher circuit is a digitally-timed one-shot: It allows an incoming event edge to set its output high, and a delayed version of this edge to reset it low.

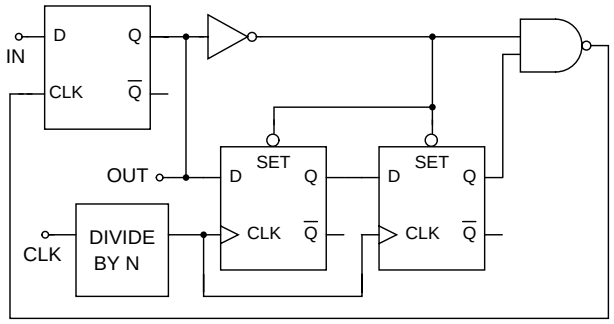


Figure B-6: Programmable hold timer circuit.

The pulse-stretcher is followed by a programmable hold timer circuit. This circuit, shown in Fig. B-6, prevents a single channel from firing too rapidly by imposing a variable dead time T_{hold} after each event is detected. No new events can occur during this dead time period. Adding the hold timer places an upper bound of $1/T_{hold}$ on the event rate that a single tag, reducing greatly the chances of events from different tags colliding in time when a single base station is being used to read them. The hold time is varied by changing the reference clock frequency f_{ref} with a programmable divide-by-N counter. Since events occur asynchronously with respect to this clock, T_{hold} varies between

$$\frac{N}{f_{ref}} \leq T_{hold} \leq \frac{2N}{f_{ref}} \tag{B.2}$$

where $f_{ref} = 16$Hz, obtained by dividing down a 32KHz on-board clock reference, and $1 < N < 15$. The average hold time can thus be varied between 94ms and 1.4s. Both the pulse-stretcher and the hold timer in each channel can be bypassed if required.

Each channel contains its own clock reference, generated by a three-stage ring oscillator. One of these oscillators also serves as the source for the event time stamp that is transmitted inside each event packet. This locally-clocked strategy alleviates the problem of clock distribution at the cost of some mismatch between the local clock frequencies. Relatively large capacitive loads were placed at the outputs of each of the ring oscillator stages to reduce this mismatch and also the phase noise. In addition, each of the ring oscillator delay cells are simple CMOS inverters which are supplied a fixed amount of current by a common high-impedance PMOS current source, which is the only node in the circuit connected to V_{DD}. This technique greatly improves the power supply insensitivity of the oscillator.

A programmable bias current source was designed for every channel. This can be used to power up external sensors, such as microphones, and consists of a 8-bit binary-weighted current DAC that can supply between 0.5μA and 128μA. The current source transistors are low-threshold voltage PMOS transistors to reduce the saturation voltage $V_{DS,sat}$ at a given current level. Their well terminals are forward biased by about 300mV to further reduce V_T. These techniques reduce the worst-case voltage compliance to 250mV, which reduces power consumption, while keeping layout area relatively small.

A supply-independent PTAT current reference [184] was used to bias all the analog circuits on the chip. The circuit is cascoded, contains no resistors, operates normally down to supply voltages of 0.5V and contains circuitry that ensures fast startup. It produces a nominal bias current of 10nA. The use of a PTAT reference and subthreshold bias currents cancels out the temperature dependence of transistor transconductances, thereby making our design robust to temperature fluctuations. To reduce power consumption, the chip was designed to operate on power supply voltages as low as 0.8V (core) and 0.5V (programmable current sources). Care was also taken to maintain good power supply rejection in all analog circuits on the chip

by bypassing the DC bias voltage and current lines.

The programming interface consists of a shift register that can be serially loaded with the desired function by a single external input during a programming phase, and latches to hold the programmed values during normal operation (see Fig. B-7). A similar serial interface was also implemented for programming the analog channel parameters (output bias current, preamplifier gain and hold time) and channel selection logic. The serial interface also allows the chip's identification code (16 bits long) to be programmed if necessary. In addition, the channel control bits can be directly programmed in parallel using analog current inputs that are quantized by low-power on-chip analog-to-digital converters (ADC's). Each control bit is multiplexed from 2 sources: the serial programming interface described earlier, and the ADC's, which allow channel parameters to be programmed more quickly.

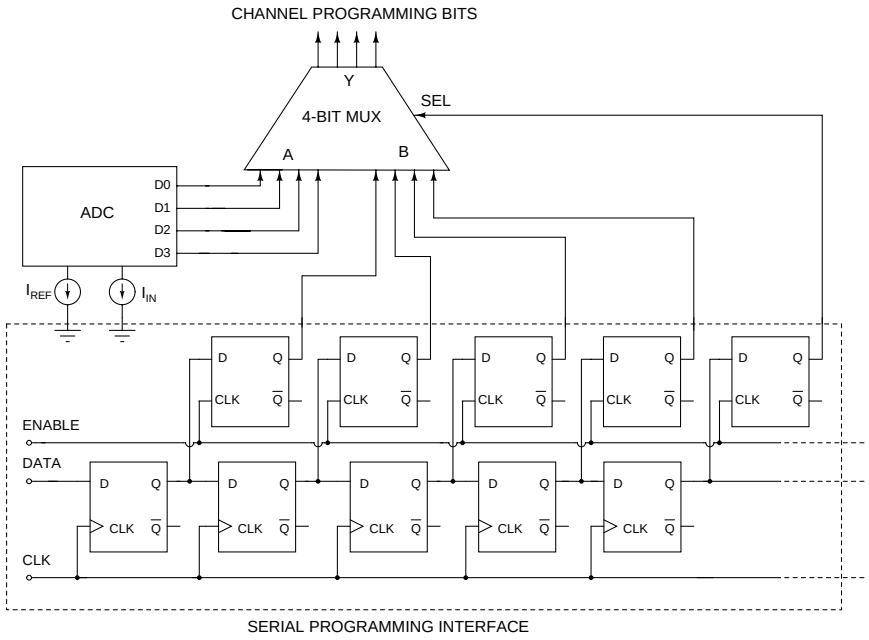


Figure B-7: Programming strategy for the event processor.

A die micro-photograph of the chip is shown in Fig. B-8, and its performance is summarized in Table B.1. The static power consumption with no external sensors is only 1.0μW. The power consumption with sensors present depends on their bias currents, which are application-dependent.

Table B.1: Chip specifications

Parameter	**Value**
Process	UMC 0.18μm CMOS
Chip area	1.5mm $\times$ 1.5mm
Core power consumption	2.7μA @ $V_{DD} = 0.8$V
Maximum sensor bias	128μA per channel
On-chip frequency reference	32KHz
On-chip current reference	10nA
Number of programmable bits	112

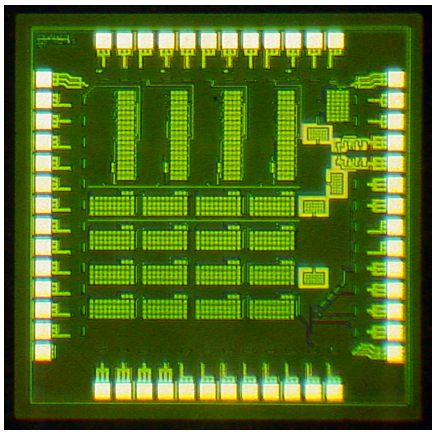


Figure B-8: Die photograph.

B.3 System Testing

The chip was tested and found to be fully functional. In this section we describe some of our experimental results. We tested our system in a cluttered laboratory with numerous barriers to RF propagation, as shown in Figure B-9. Experimentally, we were able to obtain 15μW of output power at a distance of 3.1m from an 800MHz RF source broadcasting 800mW EIRP, which was the maximum allowed by our equipment. This is enough power to run the chip and one microphone at a typical bias current of 30μA and power supply voltage of 0.5V. Increasing the transmit power to the allowed maximum of 4W should give us an operating range of 5-7m, depending upon the multipath fading characteristics of the environment. A circularly-polarized transmitter antenna can be used to make the power received by the tag less sensitive to propagation barriers and its own spatial orientation.

We also note that the curves in Figure B-3 begin to drop off sharply with increasing distance when the received RF voltage becomes smaller than the rectifier's dead zone, which causes its power conversion efficiency to decrease rapidly. If the load resistance is such that the tag operates in this region, range can also be increased by using a smaller, but more expensive chip package with lower parasitic input capacitance C_{in}. Decreasing C_{in} increases the quality factor of the input matching network, increasing the received RF voltage. Specifically, a given amount of available power is obtained at a distance that scales as $1/\sqrt{C_{in}}$. For example, the rectifier in [186] used a package with approximately half the input capacitance, resulting in $\sqrt{2} \approx 1.4$ times more range.

We noticed that the operation of the tag was largely unaffected by the presence of the human body up to a distance of approximately 1cm from the skin. The received power decreased sharply for smaller separations. The microphone must also be separated from the skin to operate normally. Both problems can be solved by backing the tag with dielectric foam approximately 1cm thick. At 3m from the source, the presence of another person directly between the source and the tag had almost no effect on received power, presumably because of multipath effects. Finally, the tag

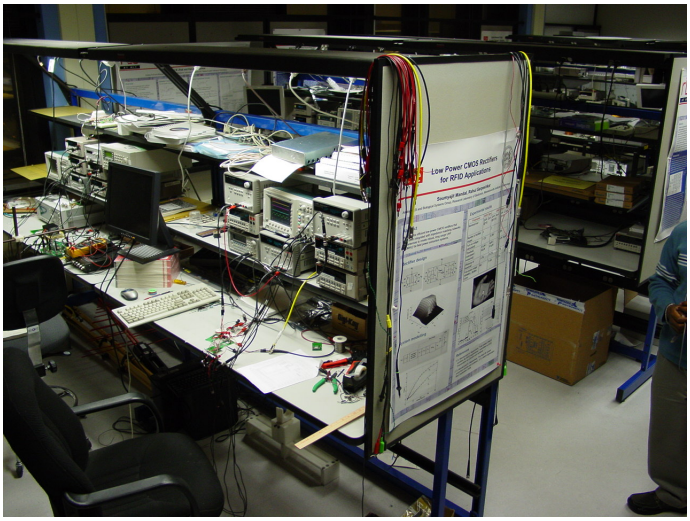


Figure B-9: Photograph of the environment within which the tag was tested.

continued to operate at 2m when the user's back was facing the transmitter, or he was lying, tag downwards, on the floor.

Figure B-10 shows measured waveforms produced by the chip in response to a single event. The backscatter signal MOD consists of an 48-bit long event packet that is repeated for the duration of the event. Each packet begins with a 16-bit '010101...' sequence to aid synchronization (visible in the figure), followed by the tag's unique 15-bit identification code, a parity bit and a 16-bit time-stamp. In this case we see that three complete packets were transmitted before the event ended.

We shall now focus on monitoring heart activity using different types of sensors since the heart is among the most important organs in the body and is also relatively easy to sense. For instance, a recording of the sounds produced by the heart constitutes a phono-cardiogram (PCG). The PCG is important for diagnosis; in particular, it contains information about the heart valves and rhythm (e.g., tachycardia and bradycardia).

B.3.1 Microphone Interface

We used a microphone to detect heart sounds. Microphone responses are bandpass, with typical lower (cut-in) frequencies between 20Hz and 100Hz and upper (cut-off) frequencies between 16kHz and 20kHz. Commercial electret microphones contain

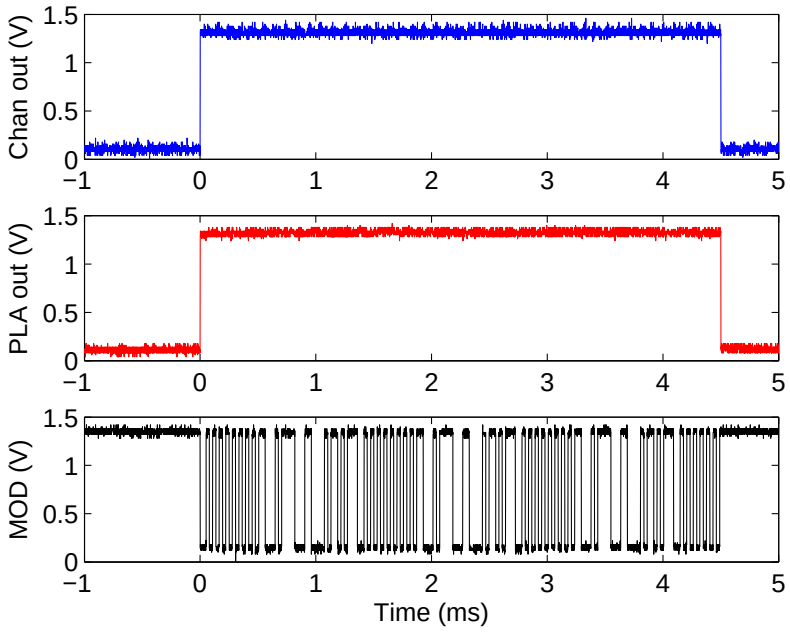


Figure B-10: Experimentally-measured waveforms generated by the chip for a single event. The figure shows (a) channel output, (b) PLA output, (c) backscatter modulation signal MOD.

built-in low-noise JFETs for buffering [8]. There are two common configurations, as shown in Figure B-11. In two-terminal microphones, shown on the left, the drain of the n-type JFET, which is normally a depletion-mode device, acts as the output terminal. It is usually connected to an an external resistive load, creating a common-source amplifier. The gate voltage is internally tied to ground at DC with the large resistor R_{big}, while C_{par} is a small, unwanted parasitic capacitance. The incoming sound pressure wave creates the voltage source v_{elec} in series with the elecret capacitance C_{elec}. Three-terminal microphones, shown on the right, configure the JFET as a source follower, and are typically more expensive.

We used a two-terminal microphone, but replaced the resistive load with a programmable current source I_{bias} running off a very low supply voltage, $V_{DD,MIC}$ to save power. In this regime the JFET is unsaturated and acts as a voltage-controlled resistor. Using the standard long-channel FET equation,

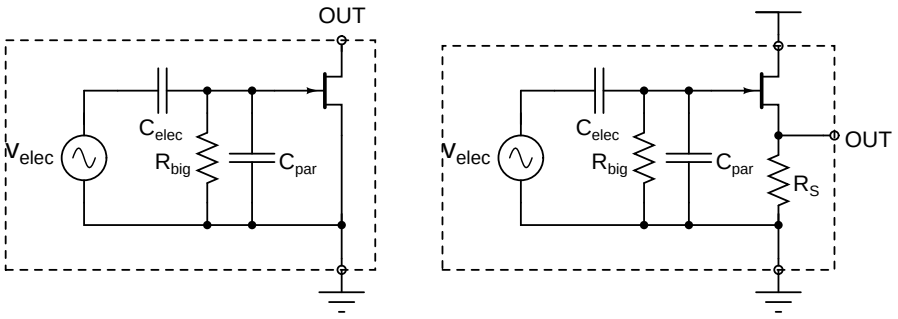


Figure B-11: Circuit diagram of common electret microphones, two-terminal (left) and three-terminal (right).

$$\begin{aligned} I_{bias} &= \beta \left[(V_{GS} - V_T) V_{DS} - V_{DS}^2/2 \right] \\ &= \beta \left[V_{DSAT} V_{OUT} - V_{OUT}^2/2 \right] \end{aligned} \tag{B.3}$$

where $V_{DSAT} = (V_{GS} - V_T)$, the saturation voltage, and β are fixed for a given microphone. Within the microphone passband the sound-generated source v_{elec} changes V_{GS} by an amount $v_{elec} C_{elec} / (C_{elec} + C_{par})$. Since the current through the JFET is fixed at I_{bias}, v_{out} decreases as v_{gs} increases, and vice-versa. The transfer function between v_{elec} and v_{out} is given by

$$\begin{aligned} \frac{v_{out}}{v_{elec}} &= -\frac{C_{elec}}{C_{elec} + C_{par}} \left(\frac{V_{DS}}{V_{GS} - V_T - V_{DS}} \right) \\ &= -\frac{C_{elec}}{C_{elec} + C_{par}} \left(\frac{\eta}{1 - \eta} \right) \end{aligned} \tag{B.4}$$

where η is defined as $V_{DS}/V_{DSAT} = V_{OUT}/V_{DSAT}$, and varies between 0 and 1. We see that the voltage gain increases as η increases, appearing to diverge as $\eta \to 1$. In reality the gain saturates at a high value because of the finite output impedances of the saturated transistor and the current source.

Since V_{DSAT} depends on I_{bias}, we can vary signal gain by using the on-chip DAC to change the bias current I_{bias}. We find, after some algebra, that

$$\frac{v_{out}}{v_{elec}} = -\frac{C_{elec}}{C_{elec} + C_{par}} \left[\sqrt{\frac{1}{1 - I_{bias}/I_{SAT}}} - 1 \right] \tag{B.5}$$

where $I_{SAT} = \beta \left(V_{GS} - V_{TS} \right)^2 / 2$, the JFET current in saturation, is fixed for a given microphone. We see that gain increases with I_{bias}, i.e., we can trade-off sensitivity with power consumption. In practice we save considerable amounts of power because heart sounds are relatively loud and do not require large amounts of gain. They occur in the range 20-250Hz. We used a Panasonic omnidirectional electret condenser microphone (WM-63PR) in a plastic enclosure. The WM-63PR, which has $I_{SAT} = 500\mu$A, was selected since it is a small, thin device (diameter = 6mm, thickness = 1.3mm) that has a low cut-in frequency (20Hz) and is also low-cost. Similar microphones that are even less expensive can also be used since sound quality is not important for this application.

A picture of the microphone sensor is shown in Fig. B-12. It is normally attached to the chest while monitoring heart activity. However, the membrane of the microphone cannot vibrate freely if it is directly attached to the skin. Therefore we built our microphone sensor with a very small cylindrical air chamber between the two. The chamber has no vents, reducing the amount of ambient noise, but its diameter and shape have little effect on sound pickup [145].

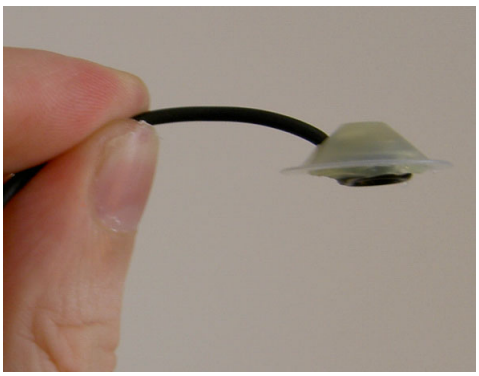


Figure B-12: Picture of microphone sensor used for measuring the PCG.

B.3.2 Experimental Results

Microphones were biased using on-chip current sources operating on $V_{DD,MIC} = 0.5$V in all our experiments. In the first experiment, two microphones were connected to channels on the chip and attached to the neck and wrist of an healthy adult subject. Proximity to the carotid and radial arteries causes a strong pulse to be commonly observed at these positions. Each microphone was biased at 30μA and the preamplifier gain was set to 8. In other successful experiments (not shown) the sensor was placed at its default position, the chest. In this position heart sounds are louder, enabling the microphone bias current to be further reduced.

Measured waveforms are shown in Figure B-13. There are two reasons why high-frequency components, such as the S_1 and S_2 sounds found in a conventional PCG waveform, are almost completely absent in these recordings. Firstly, the coupling between the skin and the microphone is a low-pass filter. Secondly, microphone sensitivities were deliberately kept low by reducing their bias currents. This was because we were mainly interested in heart rate information, which resides in the loud, low-frequency components of the PCG (from 10 - 80Hz). Each large negative event is caused by the pulse, i.e., systolic upsurge in blood pressure and consequent dilation of the arteries. The pulse travels along arterial walls as a pressure wave, with a velocity that ranges between 5 and 15m/s but is always significantly higher than that of the blood itself. The waveform at the wrist is delayed relative to that at the neck by about 95ms because of the finite velocity of this wave, i.e., because of the time taken by the systolic pulse to propagate down the length of the arm. The delay decreases as the artery walls get stiffer or blood pressure increases, and is therefore diagnostic of atherosclerosis and hypertension [22, 217].

In another experiment, we combined the wrist microphone (still biased at 30μA) with an external pulse oximeter [310] connected to another channel. The oximeter, which is used to measure oxygen saturation level in the blood, was attached to the index finger of the subject. Pulse oximeters can also be used to measure the variation in blood volume in the arteries as a function of time. Such a recording is known as a

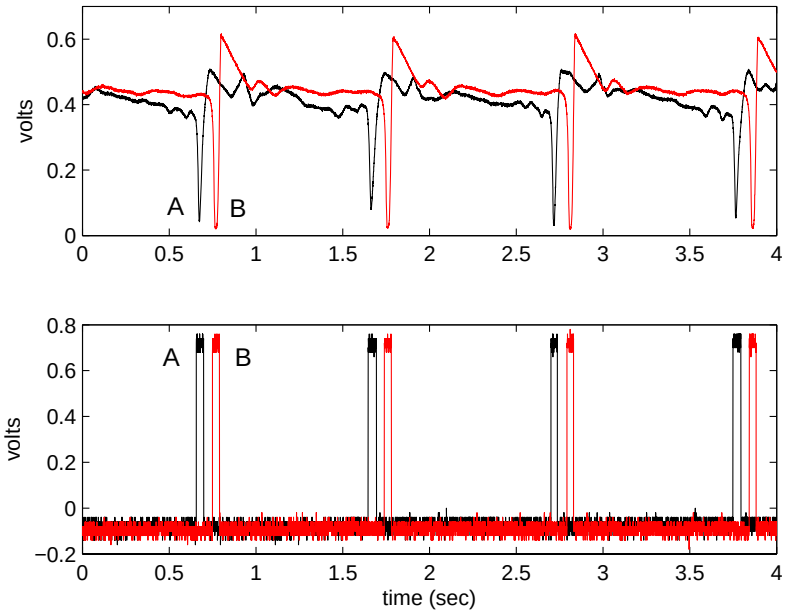


Figure B-13: Measured PCG waveforms at the neck (A) and wrist (B) of a single subject. Preamplifier (top) and channel (bottom) outputs are shown.

photo-plethysmogram, or PPG. For simplicity, we used an off-the-shelf infra-red LED light source and a Texas Instruments OPT101 photo-sensor. The OPT101 consists of a photodiode and transimpedance amplifier integrated into a single package. Its output is fed into our chip. Figure B-14 shows measured PCG and PPG waveforms. The peaks in the PPG waveform line up with the negative spikes in the PCG because we are now recording from adjacent locations, i.e., the pulse propagation delay from the wrist to the finger is small. This PPG sensor will eventually be replaced by an ultra-low-power version to create a wearable, battery-free solution.

Our system generates audio alarms during suspected patient emergencies. However, location-awareness can improve the response time to such alarms by guiding care-givers directly to the patient [115]. In addition, patterns of patient movement can be inferred by monitoring location versus time. Such additional information can be used to improve the robustness and diagnostic capabilities of our system. We now show that our tag can be localized within a room using acoustic time-of-flight measurements [18, 266]. In this experiment we used a single microphone attached to

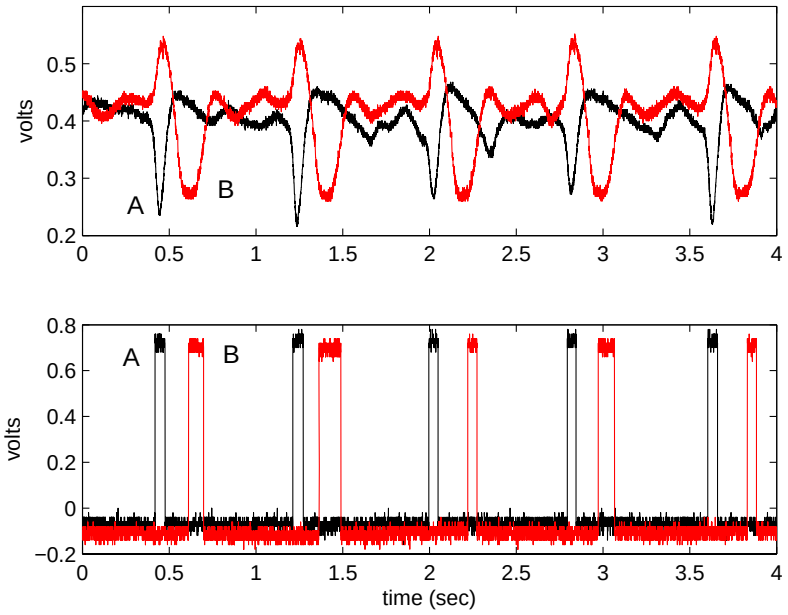


Figure B-14: PCG (A) and PPG (B) waveforms measured at the wrist and fingertip, respectively. Preamplifier (top) and channel (bottom) outputs are shown.

the chip and two speakers (L and R) placed a distance d apart. By measuring the time delays t_1 and t_2 between when each speaker beeps and the chip starts generating spikes, we can locate the position of the microphone in two dimensions.

The microphone was biased at 128μA and preamplifiers from two channels were cascaded to give a total gain of $8 \times 12 = 96$. Small, cheap speakers were placed $d = 12$ft apart and programmed to transmit 100ms tone bursts at 230Hz. The burst frequency was kept as low as possible to minimize audibility and attenuation with distance, but was limited by the poor low-frequency response of the small speakers being used. The bursts were spaced 400ms apart to ensure that all echoes from the first burst would die down before the second one arrived. The measured sound level at the center of the room with either speaker on was 87dB SPL, which is loud enough to serve as an alarm signal. The entire experiment was performed in a highly noisy indoor environment (49dB SPL, a typical quiet room being 40dB SPL) that included sound propagation barriers in the shape of furniture and people (see Figure B-9).

We estimated the propagation times t_1 and t_2 from the speaker to the microphone

by using a simple threshold-based algorithm to measure the time between the onset of each burst and the first spike detected by the chip. By using the first spike, we measure the time delay corresponding to the shortest, i.e., line of sight path. The distances of the microphone from each speaker are given by $d_1 = ct_1$ and $d_2 = ct_2$, where $c = 1130$ft/s is the speed of sound in air. Assuming that the speakers are located at $(0, -d/2)$ and $(0, d/2)$, a simple derivation shows that the estimated microphone location $(\hat{x}, \hat{y})$ is given by

$$(\hat{x}, \hat{y}) = \left(\frac{d_1^2 - d_2^2}{2d}, \pm\sqrt{d_1^2 - \left(\hat{x} + \frac{d}{2}\right)^2} \right) \tag{B.6}$$

One of the two values of $\hat{y}$ can always be rejected since it lies outside the room, thereby giving us an unique solution. The measured microphone positions are shown in Fig. B-15 for 9 different positions and 20 trials. The average standard deviation in the measured positions was 1.4ft (0.43m), and the average error between the measured and actual positions was 1.97ft (0.6m). This number includes error due to electronic (signal detection) delay on the tag, which increases with distance from the speaker, becoming approximately 1ft (0.3m) at the furthest postions.

The total measured delay includes two components: acoustic and electronic. We denote $T = 1/230$Hz to be the time period of the sound signal, and c to be the speed of sound in air. In the case that was tested (a single tag present in the room), electronic delay is primarily caused by the time taken by the output of the preamplifier to increase to the threshold value required to trigger an event (see Figure B-16). This delay ranges from zero (if the peak value of the preamplifier output is much larger than the threshold) to $T/4$ (if the peak value is just equal to the threshold). Since the peak value of the preamplifier output depends on the distance of the microphone from the speaker, so will the electronic delay. This effect causes an error of approximately $cT/4 \approx 0.3$m in measured position when the microphone is far away from the speaker and an event is barely triggered. This error can be reduced by using a higher audio frequency, i.e., lower value of T.

The current accuracy of our system already provides important information about

the location of the patient. For example, we can distinguish between the bed, a chair and the bathroom. Localization accuracy can be further increased if necessary by using louder sounds or a higher transmission frequency to improve timing precision, and extended to three dimensions by adding a third speaker. Finally, our tags can also be localized to particular rooms by determining which RF transmitter powers them up. This information allows localization accuracy to degrade gracefully in case the audio scheme fails.

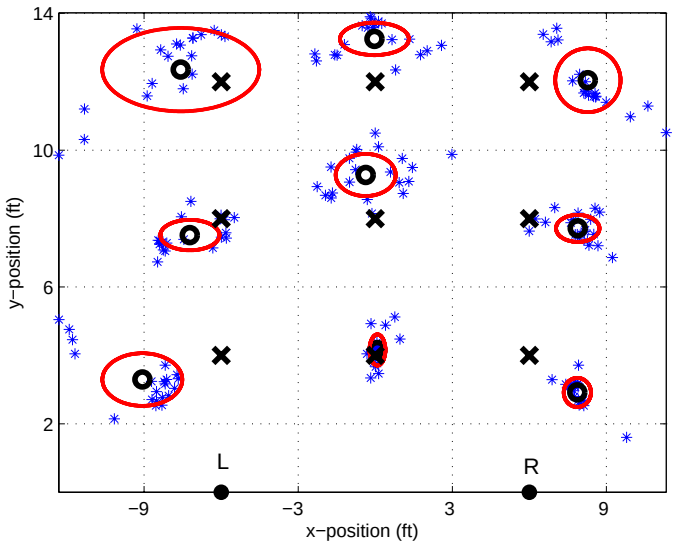


Figure B-15: Localization in two dimensions using acoustic time delays. Measured data points ($*$), mean positions ($\circ$) and standard deviation ellipses are shown for nine cases. Actual positions ($\times$) and the two speakers ($\bullet$) are also marked.

The audio alarm and localization technique that we have described is quite general and can be extended to other wireless sensor applications. For example, it can form the basis for sensor-fusion algorithms where sensors such as video cameras that provide high-bandwidth information can be activated by the audio alarm only when abnormal events are detected. The amount of information that needs to be continuously monitored by a human operator is thereby reduced. This mode of operation also allows power-hungry sensors to be turned off most of the time.

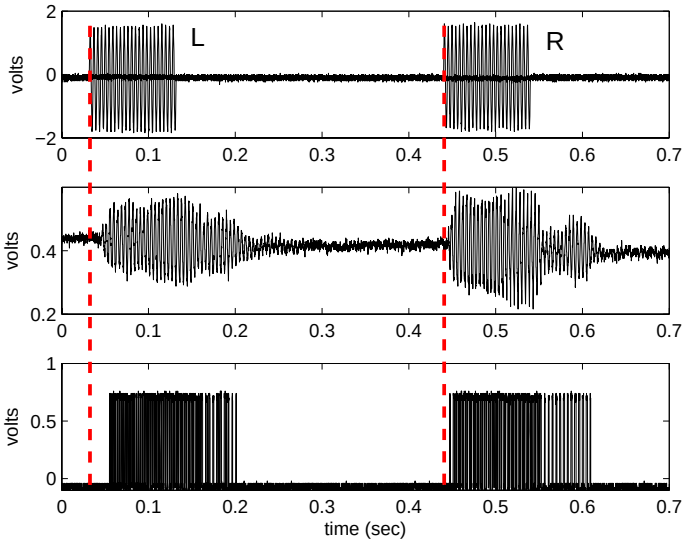


Figure B-16: Measured time-domain waveforms of the outputs of the left (L) and right (R) speakers (top), the output of the preamplifier (middle) and the channel (bottom). Echoes and significant differential time delay between the two propagation paths are visible.

B.4 Conclusions

We have demonstrated a programmable, general-purpose low-power chip that can be powered-up by harvesting radiated RF energy. The chip forms the basis for a low-cost, battery-free sensor platform for medical monitoring and alarm generation. We have also shown examples of how our chip can be used with PCG and PPG input signals, and in the context of an audio localization scheme.

Appendix C

Impedance Matching

C.1 Single-Element Matching

Consider matching an load impedance $Z_L = R_L + \jmath X_L$ to a real source impedance $Z_0 = R_0$ by using a single reactive element $\jmath X_M = -\jmath X_L$ and an ideal transformer with a turns ratio equal to $\sqrt{R_L/R_0}$. This problem can always be exactly solved at one frequency, say ω_0. In other words, by picking an inductor or capacitor that has a reactance equal to $-X_L$ at ω_0, the reflection coefficient can be made zero, i.e., $\Gamma(\omega_0) = 0$. The ideal transformer can then transform R_L, which is the remaining part of the load, into R_0. This transformation can be exact at all frequencies since both R_L and R_0, being purely resistive, are independent of frequency[1]. However, $dX_L/d\omega$ and $dX_M/d\omega$ must both be positive[2]. Since $X_M = -X_L$ at ω_0, $X_M + X_L$ can only be zero at that one frequency. Thus reactance cancelation is inexact everywhere apart from ω_0, making the reflection coefficient non-zero. Fortunately, because reactances change only gradually with frequency the cancelation is approximately correct for frequencies close to ω_0. This allows us to define a fractional impedance matching bandwidth B, as follows

[1]The alert reader may spot a problem with this argument. While R_0, presumably a known quantity, can be reasonably expected to be constant, can we really expect the same from R_L? After all, R_L models the amount of loss present in the unknown load Z_L. In fact this is a real problem when Z_L represents an antenna's input (terminal) impedance. More on this later.

[2]This is known as Foster's Reactance Theorem. It states that the slope of reactive (lossless, purely imaginary) driving-point impedance functions versus frequency must be positive.

$$B \equiv \frac{\omega_H - \omega_L}{\sqrt{\omega_H \omega_L}} \approx \frac{\omega_H - \omega_L}{\left(\frac{\omega_H + \omega_L}{2}\right)} \tag{C.1}$$

where $\omega_H = \omega_0 + \Delta\omega_1$ and $\omega_H = \omega_0 - \Delta\omega_2$ are the frequencies at which $|\Gamma(\omega)| = \Gamma_{max}$, the maximum allowable value of the reflection coefficient, and $(\Delta\omega_1, \Delta\omega_2) > 0$. We shall now assume that $|\Gamma(\omega)|$ increases monotonically from zero as we move away from ω_0 in either direction. We shall also assume that the increase is symmetric about ω_0. Both assumptions are usually correct if $\Delta\omega_1$ and $\Delta\omega_2$ are much smaller than ω_0, i.e., close to the perfectly matched frequency. As a result, we can define $\Delta\omega \equiv \Delta\omega_1 = \Delta\omega_2$. The fractional bandwidth B of our *first-order matching network* is then

$$B \approx \frac{2\Delta\omega}{\omega_0} \tag{C.2}$$

Let us expand $X_L(\omega)$ and $X_M(\omega)$ as a Taylor series about ω_0. The total (residual) reactance $X_L + X_M$ at a frequency $\omega_0 + \Delta\omega$ is

$$X_{total}(\Delta\omega) = X_L(\omega_0) + X_M(\omega_0) + \left.\frac{dX_L}{d\omega}\right|_{\omega_0} \Delta\omega + \left.\frac{dX_M}{d\omega}\right|_{\omega_0} \Delta\omega \tag{C.3}$$

$$= \left(\left.\frac{dX_L}{d\omega}\right|_{\omega_0} + \left.\frac{dX_M}{d\omega}\right|_{\omega_0} \right) \Delta\omega \tag{C.4}$$

The magnitude of the reflection coefficient at $\omega_0 \pm \Delta\omega$ is Γ_{max} by definition. Assuming the ideal transformer converts the source resistance R_0 into R_L over the frequency range of interest, we get

$$\Gamma_{max}^2 = \left| \frac{Z_L(\Delta\omega) + \jmath X_M(\Delta\omega) - R_L}{Z_L(\Delta\omega) + \jmath X_M(\Delta\omega) + R_L} \right|^2 = \left| \frac{\jmath X_{total}(\Delta\omega)}{\jmath X_{total}(\Delta\omega) + 2R_L} \right|^2 \tag{C.5}$$

Combining (C.2), (C.3) and (C.5) and performing some algebra, we get

$$B\omega_0 \left(\frac{\left.\frac{dX_L}{d\omega}\right|_{\omega_0} + \left.\frac{dX_M}{d\omega}\right|_{\omega_0}}{4R_L} \right) = \frac{1}{\sqrt{\frac{1}{\Gamma_{max}^2} - 1}} \tag{C.6}$$

The *quality factor* of the load is defined as

$$Q_L(\omega_0) = \frac{\omega_0 \left(\frac{dX_L}{d\omega}\Big|_{\omega_0} + \frac{dX_M}{d\omega}\Big|_{\omega_0} \right)}{2R_L} \tag{C.7}$$

For example, if $X_L = \omega L$ and $X_M = -1/(\omega C)$ with $\omega_0 = 1/\sqrt{LC}$, we get $dX_L/d\omega = dX_M/d\omega = L$. Therefore

$$Q_L(\omega_0) = \frac{\omega_0 L}{R_L} = \frac{\mathrm{Im}(Z_L)}{\mathrm{Re}(Z_L)} \tag{C.8}$$

which is the more familiar expression for the quality factor of a passive impedance. Intuitively we can see why we need the factor of 2 in the denominator of (C.7). At resonance ($\omega = \omega_0$) the magnitudes of the load and matching reactances are equal and they carry the same current (since they are in series); therefore they should the same amount of energy. As a result the quality factor of the load is obtained by dividing the total energy stored in both reactances by *twice* the energy lost per cycle. Substituting (C.7) in (C.6), we get

$$BQ_L(\omega_0) = \frac{2}{\sqrt{\frac{1}{\Gamma_{max}^2} - 1}} \tag{C.9}$$

C.2 Higher-Order Matching Networks

The problem of matching an arbitrary load impedance to a source resistance by using a passive, lossless network of arbitrary complexity was first considered by Fano [68], building on earlier work by Bode [20]. Bode had considered the special case of a parallel R-C load, and had proven that

$$\int_0^\infty \ln\left(\frac{1}{|\Gamma(\omega)|}\right) d\omega \leq \frac{\pi}{RC} \tag{C.10}$$

where $\Gamma(\omega)$ is the reflection coefficient. The goal of the matching network is to keep $|\Gamma(\omega)| < \Gamma_{max}$, the maximum acceptable reflection coefficient, over the widest possible bandwidth. An n-th order matching network uses n tuned circuits (i.e., $2n$ reactive

elements) to achieve this goal. Since the minimum possible value of the integral in (C.10) is fixed, it is most efficient to make $|\Gamma(\omega)|$ fixed at Γ_{max} over the entire matching bandwidth, and completely mismatched ($|\Gamma(\omega)| = 1$) everywhere else. In this way only frequencies within the matching bandwidth contribute to the integral; the matching bandwidth is therefore as large as possible. The function $\Gamma(\omega)$ is now a rectangular boxcar versus frequency. A matching network that produces it must contain an infinite number of elements. This is reminiscent of the fact that a boxcar filter with infinitely sharp roll-off also cannot be made without an infinite number of elements. In fact, the synthesis of impedance matching networks and passive filters are closely related subjects [133, 194]. Within the passband of a filter, most of the power from the source is absorbed in the output load, so the input reflection coefficient presented by the filter to the source is small. Conversely, in the stopband almost no power makes it to the output; most of it is reflected back towards the source.

Bode's criterion in (C.10) can be easily extended to both series and parallel R-C and R-L loads if (C.10) is written in terms of Q_L, the quality factor of the load. In addition, let us assume the existence of an infinite-order matching network that synthesizes a boxcar reflection coefficient function that is equal to Γ_{max} in magnitude within a fractional bandwidth B_∞ about ω_0, and is unity everywhere else. Equation (C.10) can then be rewritten as

$$B_\infty Q_L(\omega_0) = \frac{\pi}{\ln\left(\frac{1}{\Gamma_{max}}\right)} \tag{C.11}$$

Fano investigated the maximum fractional bandwidth B_n that was achievable using a matching network containing a finite number of circuit elements [69, 70]. As an important example, consider matching a parallel R-C or series R-L load. In either case the reflection coefficient function $|\Gamma(\omega)|$ is that of a low-pass filter, ideally a rectangular boxcar. Common basis functions used to approximate rectangular boxcars are Chebyshev and elliptic polynomials. Using the former, the reflection coefficient of a n-th order matching network is defined as

$$\left|\Gamma\left(\omega/\omega_0\right)\right|^2 = 1 - \frac{1}{(1+K^2) + \epsilon^2 T_n^2\left(\omega/\omega_0\right)} \tag{C.12}$$

where ω_0 is the center frequency, K and ϵ are arbitrary constants and $T_n(x) = \cos\left(n\cos^{-1}(x)\right)$ is the n-th order Chebyshev polynomial. We now define a and b such that

$$\begin{aligned} \sinh^2(na) &= \frac{1+K^2}{\epsilon^2} \\ \sinh^2(nb) &= \frac{K^2}{\epsilon^2} \end{aligned} \tag{C.13}$$

The quantities a and b control the location of the poles and zeros, respectively, of the function $\left|\Gamma\left(\omega/\omega_0\right)\right|$. Fano considered the problem of minimizing the maximum in-band reflection coefficient Γ_{max} for a given value of n. His solution was expressed in the forms of the following simultaneous equations:

$$\begin{aligned} B_n Q_L\left(\omega_0\right) &= \frac{2\sin\left(\frac{\pi}{2n}\right)}{\sinh(a) - \sinh(b)} \\ \Gamma_{max} &= \frac{\cosh(nb)}{\cosh(na)} \\ \frac{\tanh(na)}{\cosh(a)} &= \frac{\tanh(nb)}{\cosh(b)} \end{aligned}$$

where the third equation is derived by minimizing Γ_{max} (as given by the second equation) subject to the bandwidth constraint imposed by the first equation. These equations can be solved analytically for $n = 1$ to give

$$B_1 Q_L\left(\omega_0\right) = \frac{2}{\left(\frac{1}{\Gamma_{max}} - \Gamma_{max}\right)} \tag{C.14}$$

Similarly, for $n = 2$, we get

$$B_2 Q_L(\omega_0) = \frac{2}{\left(\frac{1}{\sqrt{\Gamma_{max}}} - \sqrt{\Gamma_{max}}\right)} \tag{C.15}$$

The analysis gets significantly harder for higher values of n, and is preferably performed with the help of a computer. Standard network synthesis methods can be used to find the element values of the matching network. An important case uses a ladder structure with series and shunt tuned circuits ("tanks") that alternate as shown in Figure C-1. All the tanks resonate at the same frequency, and n of them are required to realize a Chebyshev or elliptic function of the n-th order, i.e., an n-th order matching network.

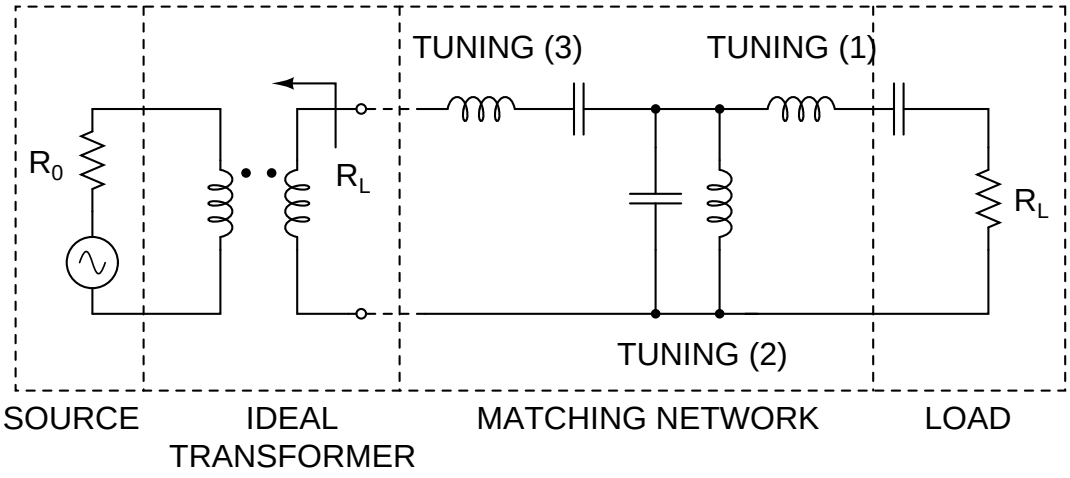


Figure C-1: Impedance-matching to a resistive source impedance. A series R-C load is shown, but other series and parallel R-C and R-L loads may be treated similarly.

We now want to quantify the loss in bandwidth that we suffer by using a finite number of tuned circuits in the matching network. In order to do this we keep the load fixed, which makes Q_L constant, and plot B_∞/B_n. We refer to this quantity, which must be greater than unity, as the *bandwidth loss ratio*. Note that our original result for a first-order matching network, (C.10), was approximate because it was based on a Taylor series expansion about ω_0. However, we expect it to be close to the value predicted by (C.14) (which is exact) for small values of Γ_{max}. This is because low reflection coefficient values occur close to ω_0, which is where the expansion should be valid. The results are shown in Figure C-2. We see that the approximate formula

for $n = 1$ behaves as expected. Also, the minimum values of the other curves occur at $\Gamma_{max} = 1$ and are equal to π and $\pi/2$ (for $n = 1$ and 2, respectively). This behavior is the basis of the following common claims [171]:

1. Increasing the order of the matching network from 1 to 2 increases bandwidth by a factor of 2.

2. Further increases in matching network complexity produce diminishing returns. The bandwidth improvement factor between $n = 2$ to $n = \infty$ is only $\pi/2$.

We see that strictly speaking, these statements are only true if $\Gamma_{max} = 1$. However, it is clear that $n = 2$ is already much better than $n = 1$, especially for small values of Γ_{max}, i.e., in applications where only small amounts of reflected power can be tolerated.

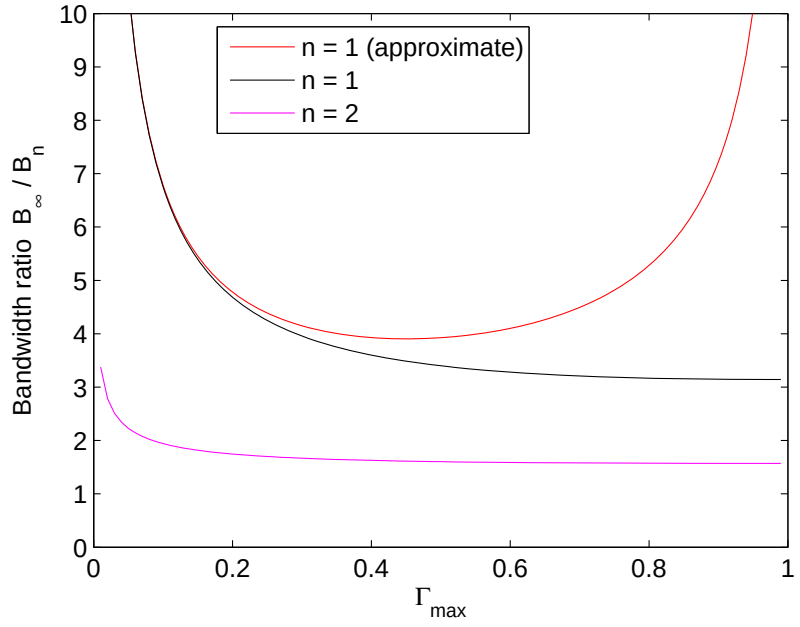


Figure C-2: Bandwidth loss factor of first and second order impedance matching networks.

C.3 Impedance Matching of Antennas

We now consider the problem of matching the input, or terminal impedance of an antenna, given by $Z_A = R_A + jX_A$. Most antennas (and other distributed structures) support standing waves, and thus show behavior that is roughly periodic with frequency[3]. For example, a simple dipole is *resonant* at frequencies where its length l is an odd multiple of $\lambda/2$, and *anti-resonant* when l is an even multiple of $\lambda/2$ (here λ is the wavelength). At resonances, Z_A looks like a series resonator (low impedance, or voltage null), while at anti-resonances, it looks like a parallel resonator (high impedance, or current null). This behavior is completely analogous to that of other standing-wave systems, such as acoustic pipes.

Resistances and anti-resonances must alternate in any antenna. This behavior is predicted by Foster's reactance theorem, which states that the poles (anti-resonances) and zeros (resonances) of any reactive impedance function must alternate[4] However, Foster's theorem also predicts that the slope of the reactance X_A versus frequency must be positive. Antennas violate this prediction, fundamentally because radiation looks like loss, making the antenna terminal impedance not purely reactive ($R_A \neq 0$). In other words, $X'_A = dX_A/d\omega$ can be negative in certain frequency ranges. These frequency ranges become smaller as the quality factor of the antenna increases, i.e. Z_A begins to look like a pure reactance.

The most general definition of the quality factor Q of a tuned system is 2π times the ratio of the time-averaged stored energy in the system to the energy dissipated by the system per cycle. In the case of an antenna, dissipation includes radiation. If the antenna contains no active or nonlinear materials and is impedance-matched to a source (or load) at a frequency ω_0 by using a reactance $-jX_A$, the quality factor can be found by adding up the power in the reactive and radiated fields. A comprehensive recent review article [319] finds that the quality factor is given by

[3]Exceptions include frequency-independent antennas, which have geometries that can be specified using angles alone, and log-periodic antennas, which behave periodically on a *logarithmic* frequency scale.

[4]Reactive impedance functions are the driving-point impedances of networks of pure reactances. Therefore they cannot contain a real component.

$$Q_A(\omega_0) = \frac{\omega_0 \left| Z'_A(\omega_0) \right|}{R_A(\omega_0)} \tag{C.16}$$

where $Z'_A = dZ_A/d\omega$ and ω_0 is the tuned frequency. Note that the expression above uses Z'_A instead of the more common X'_A to define the quality factor. This definition makes Q_A valid in both resonant and anti-resonant frequency ranges. At anti-resonances, $R'_A = dR_A/d\omega$ can be large and make a non-negligible contribution to the frequency dependence of Z_A, thus affecting bandwidth. At resonances, R'_A is small and Z'_A can be replaced by X'_A.

Note that (C.16) was defined for an antenna that was impedance-matched with a single matching element $-jX_A$. Also, for frequencies near ω_0, Z_A looks either like a series or parallel resonator (RLC) circuit. Therefore, if the quality factor of the antenna is high enough for the behavior close to a *single* resonance or anti-resonance to be relevant, we expect the impedance-matching bandwidth to be simply that of a first-order network, and given by (C.9) or (C.14). Indeed, [319] derive the expression

$$BQ_A(\omega_0) = \frac{2}{\sqrt{\frac{1}{\Gamma_{max}^2} - 1}} \tag{C.17}$$

which matches (C.9) and is only valid when $\Gamma_{max} \leq 0.5$. The analysis can now be extended to higher-order matching networks in a similar way. The impedance-matching properties of narrowband antennas is therefore identical to that of simple RLC resonators. Broadband antennas that need to be impedance-matched over several resonances and anti-resonances cannot be modeled as single resonators, making them harder to analyze.

Since the bandwidth of an antenna is inversely proportional to its quality factor, possible restrictions on achievable values of quality factor are of interest. Fundamental limitations exist on the quality factor of electrically small antennas. Classical work by Chu [41], later refined by McLean [196], predicts that the minimum possible quality factor, $Q_{A,min}$ for a linearly polarized antenna is given by

$$Q_{A,min} = \frac{1}{k^3 a^3} + \frac{1}{ka} \tag{C.18}$$

where k is the wavenumber and a is the radius of the smallest sphere required to completely enclose the antenna. In order to derive this expression it was assumed that the antenna excited only a single wave mode (TE_{01} or TM_{01}) and stored no energy inside the enclosing sphere. All practical antennas have a quality factor greater than $Q_{A,min}$ because the second condition cannot really be met. A similar result can be derived for elliptical and circularly polarized antennas. Also, in the electrically small limit as $a \to 0$, we note that $Q_{A,min}$ becomes inversely proportional to the volume of the enclosing sphere, i.e. $Q_{A,min} \propto 1/a^3$. This approximate result was originally derived by Wheeler [313].

Appendix D

Some Novel Circuit Architectures

D.1 An Adaptive Current-Mode PLL

This section describes some thoughts on building current-mode phase-locked loops (PLLs) with adaptive loop bandwidth and the ability to operate on very low power supply voltages. The ideas were developed over several years but never properly followed up owing to lack of time.

D.1.1 Introduction

In this section we describe the design of a low-voltage self-biased current-mode phase-locked loop (PLL) that can be used for clock and data recovery (CDR) on RFID tags and other low-power systems. The basic idea behind self-biased PLLs [189] can be understood from the linearized model of the system shown in Fig. D-1. The diagram is drawn in the phase domain, with ϕ_{ref} and ϕ_{out} representing the input and output phases, respectively. K_p represents the linearized gain of the phase detector, and K_{cco}, with units of Hz/A, is the gain of the current-controlled oscillator (CCO). I_{cco} is the control current for the CCO (i.e., $f_{out} = K_{cco}I_{cco}$). The output of the CCO is divided in frequency by a factor of N before being fed back to the phase detector, making this structure an integer-N frequency synthesizer. Finally, $I_{cp} = \alpha I_{cco}$ converts the output voltage of the phase detector into a current and feeds it into the (current-

mode) loop filter $H(s)$. In most implementations, I_{cp} represents the bias current of a charge pump. Fig. D-1 shows a self-biased topology; since I_{cp} is set by I_{cco}, no external bias currents are required by the system.

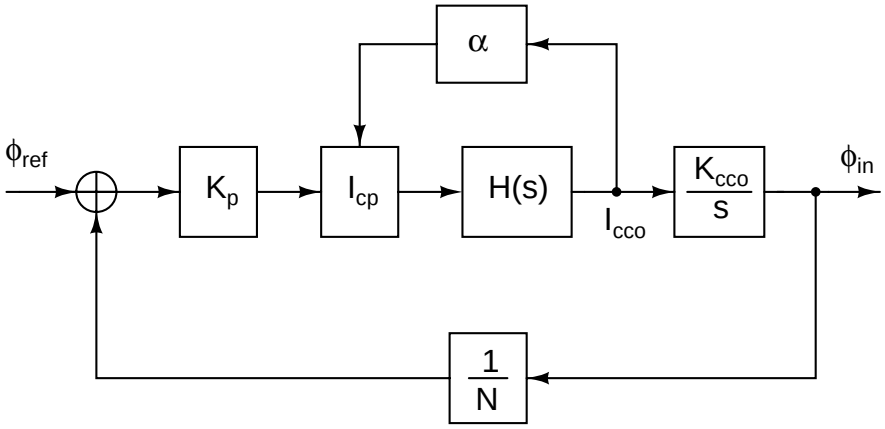


Figure D-1: Linear model of a self-biased phase-locked loop. $H(s)$ represents the s-domain transfer function of the loop filter.

The self-biased topology shown in Fig. D-1 possesses several features that make it attractive for IC applications. These advantages are specially apparent at lower operating frequencies. Firstly, the loop transmission is given by

$$L(s) = \frac{K_p K_{cco}}{N} \frac{H(s)}{s} I_{cp} \tag{D.1}$$

If we now make the substitutions $I_{cp} = \alpha I_{cco}$, $f_{out} = K_{cco} I_{cco}$, and $f_{ref} = f_{out}/N$ (this is true when the PLL is locked; f_{ref} refers to the input frequency), (D.1) becomes

$$L(s) = \alpha K_p f_{ref} \frac{H(s)}{s} \tag{D.2}$$

Eqn. D.2 shows that $L(s)$ is not a function of K_{cco} or N, and only depends on α, K_p,f_{ref} and $H(s)$. Of these parameters, α can be easily set with a current multiplication ratio (i.e., a current mirror) and K_p is characteristic of the phase detector topology and is implementation-independent. f_{ref} is the input frequency and is set by the user. This means that $L(s)$ essentially depends only on the loop filter transfer function $H(s)$. The fact that K_{cco} does not appear in $L(s)$ means that the loop transmission does not depend on the CCO and its parasitic capacitances. The in-

dependence of $L(s)$ with N means that the system can synthesize a wide range of output frequencies $f_{out} = Nf_{ref}$ by varying N without changing the phase margin or other measures of loop stability.

The self-biased topology has other advantages. The first is that current-starved ring oscillators, which are the simplest and commonest oscillators found inside IC PLL's, are naturally current controlled. The oscillation frequency of such circuits can typically be controlled in an extremely linear way over 6 or more orders of magnitude by varying a bias current. Using I_{cco} to directly control the CCO thus results in a highly linear loop. In addition, because $H(s)$ now has current input and output, it can be implemented using current-mode filter design techniques. This results in a topology that is naturally amenable to low supply-voltage operation and is extremely efficient: no extra $I \leftrightarrow V$ conversions are required in the loop.

D.1.2 System level design

We started our PLL design procedure by defining the system requirements. The input frequency range over which the loop remains well-behaved should be at least 100KHz-500KHz to accommodate typical RFID data rates. Also, for most CDR applications, Hogge's phase detector is an excellent (and common) choice. We therefore decided to use this type of phase detector. This choice results in $K_p = 1/2\pi$. Next, we assume a second order loop, so that the loop filter is simply an integrator. Since no integrator is ideal, let us assume that it has a first-order low pass filter characteristic, with the pole at ω_p and a finite DC gain β. The resultant unity gain frequency of the integrator is approximately $\omega_u \approx \beta\omega_p$. As a result, the crossover frequency ω_c of the loop is given by

$$\omega_c \approx \sqrt{\frac{\alpha f_{ref}\omega_u}{2\pi}} \tag{D.3}$$

We now want to place a zero to stabilize the loop and a third high-frequency pole above crossover to reduce high frequency ripple on the CCO control line. As a first guess, we place the zero at $\omega_z = \omega_c/2$ and the third pole at $\omega_3 = 20\omega_c$. However,

as soon as they are placed, the crossover frequency itself changes. Thus the design process becomes iterative. In practice, we have found that the placement locations given above yield good performance in most situations. The problem is that f_{ref} may vary based on the application. If the loop filter pole and zero locations are fixed, loop stability becomes dependent on f_{ref}. This is evident from Fig. D-2, which plots $L(s)$ for values of f_{ref} varying from 10kHz to 1MHz and $\alpha\beta = 500$. We see that the phase of $L(s)$ is invariant with f_{ref}, even though ω_c varies by about an order of magnitude. This causes the phase margin of the loop to change significantly as f_{ref} varies. Fig. D-2 shows that the phase margin increases as f_{ref} increases, reaches a maximum at some value of f_{ref} and then decreases.

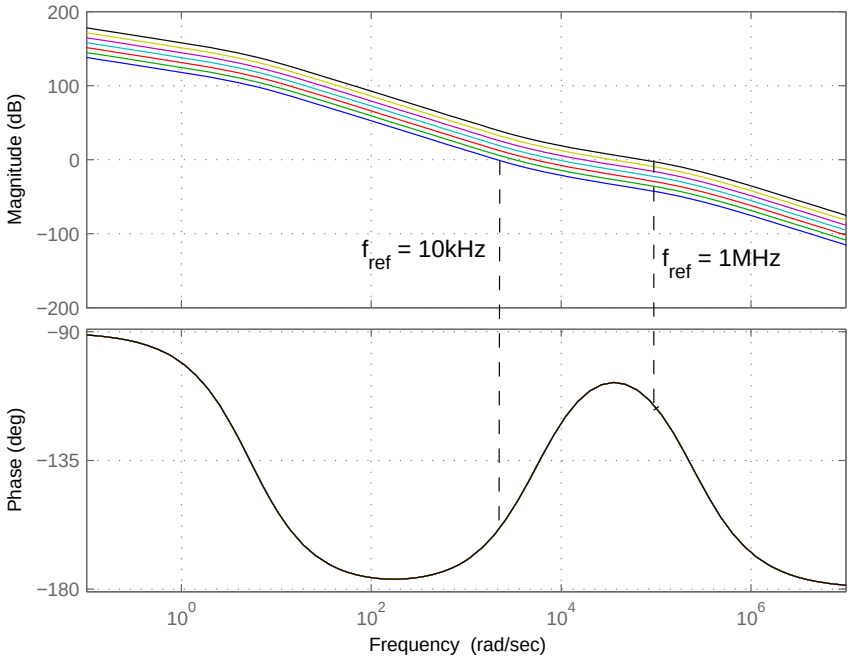


Figure D-2: PLL loop transmission $L(s)$ as a function of the input frequency f_{ref}.

There is an interesting way to solve this problem. A current-mode integrator with a DC gain of β typically has a pole frequency ω_p that is given by

$$\omega_p = \frac{\kappa I_0}{\beta \Phi_T C_L} \tag{D.4}$$

where Φ_T and κ have their usual meanings, C_L is the load capacitance at the output

of the integrator, and I_0 is a DC bias current. Substituting (D.4) in (D.3), we get

$$\omega_c \approx \sqrt{\frac{\alpha\kappa}{2\pi\Phi_T}\frac{f_{ref}I_0}{C_L}} \tag{D.5}$$

Eqn. D.5 suggests that we should have $I_0 \propto 1/f_{ref}$ in order to have a loop crossover frequency that is invariant with f_{ref}. Varying the bias current is much easier than varying C_L to get the same result. Since ω_c is now approximately constant, the loop-stabilizing zero and high frequency pole locations are also fixed. This means that they can be implemented with fixed passive components (R's and C's). By allowing a single pole location in $H(s)$ to be variable and controlling it via a bias current, loop stability has been made independent of the input frequency. The resultant PLL architecture is shown in Fig. D-3. The input frequency f_{ref} is converted into a current by using a $f \rightarrow I$ converter. The converter is represented as sK_f in Fig. D-3, where s converts from phase to frequency and K_f is a constant. The output is then inverted and fed forward to create the bias current for the loop filter. There are several ways to build the $f \rightarrow I$ converter. For example, a "leaky" integrator provides the right characteristics. The $1/x$ block can be easily implemented using a translinear current divider.

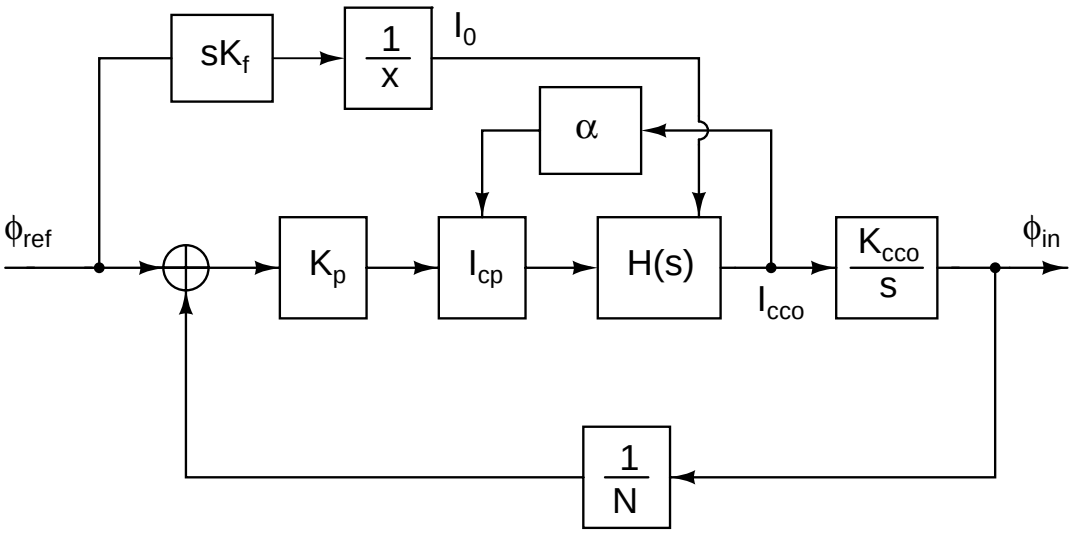


Figure D-3: Linear model of a self-biased phase-locked loop with feed-forward adjustment of the loop filter integration time constant.

Finally, it should be noted that the self-biased topology can easily be made temperature-insensitive by deriving I_0 from a proportional-to-absolute-temperature (PTAT) current reference. Constant-g_m references, for example, produce PTAT outputs. Since the temperature dependence of ϕ_T in the denominator has been canceled, both ω_p and ω_c now become independent of temperature.

The results of implementing the architecture of Fig. D-3 are shown in Fig. D-4. Fig. D-4 shows calculated loop transmissions for the PLL with f_{ref} varying from 10KHz to 1MHz. The value of I_0 for the loop filter is varied $\propto 1/f_{ref}$ using the equation $I_0 = (I_{0,nom} f_{ref,nom}) / f_{ref}$, where $I_{0,nom}$ is selected to give good phase margin at $f_{ref,nom} = 300\text{KHz}$.[1] We see that, in contrast to Fig. D-2, $L(s)$ is almost invariant w.r.t. f_{ref} near crossover. As a result, the phase margin and stability of the loop are invariant with f_{ref}. This property allows us to use this PLL for several interesting CDR applications. One of these, which has particular relevance to RFID, is variable data rate communications. Dynamically changing the data rate for reader-tag communications requires a frequency-agile CDR circuit (such as the one we have just described) but has several advantages over a constant data rate system. For example, by varying the transmitted data rate based on received power at the tag, a constant bit error rate (BER) can be maintained for tags scattered over a range of distances from the reader.

D.1.3 Other Loop Compensation Techniques

There are at least two other obvious techniques to adaptively change the structure of the loop so good performance is retained as f_{ref} varies over wide ranges. We describe these techniques below.

The first technique consists of making $I_0 \propto \sqrt{f_{ref}}$, so that the loop crossover frequency ω_c becomes roughly proportional to f_{ref} (instead of remaining constant as before). In order to keep the phase margin constant as f_{ref} varies, we also have to change the frequency of the loop-stabilizing zero $\omega_z \propto f_{ref}$. Because the loop bandwidth scales with the reference frequency f_{ref}, the loop will settle in a constant

[1] *nom* refers to nominal values for various parameters.

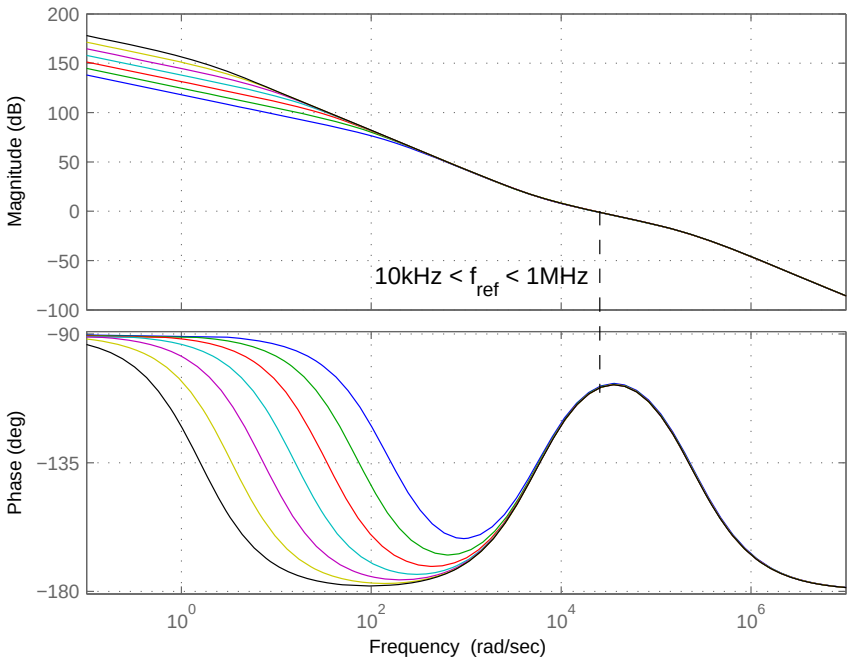


Figure D-4: PLL loop transmission $L(s)$ as a function of the input frequency f_{ref} with feed-forward adjustment of $H(s)$.

number of cycles of f_{ref}, which may be desirable in certain applications. Loop transmissions $L(s)$ for various values of f_{ref} using these scheme are shown in Figure D-5. As expected, the phase margin remains almost constant.

The second technique keeps I_0 constant, so that the loop crossover frequency ω_c becomes roughly proportional to $\sqrt{f_{ref}}$. In order to keep the phase margin constant as f_{ref} varies, we also have to change the frequency of the loop-stabilizing zero $\omega_z \propto \sqrt{f_{ref}}$. Loop transmissions $L(s)$ for various values of f_{ref} using this scheme are shown in Figure D-6. As expected, this scheme also keeps the phase margin almost constant.

Either of these alternative techniques may be used, but the latter has a particularly simple circuit implementation. The only loop parameter that needs to be varied in this case is the frequency of the loop-stabilizing zero $\omega_z \propto \sqrt{f_{ref}}$. The value of ω_z is usually set by adding a resistor R_z in series with the load capacitor C_L at the output of the integrator:

$$\omega_z = \frac{1}{R_z C_L} \tag{D.6}$$

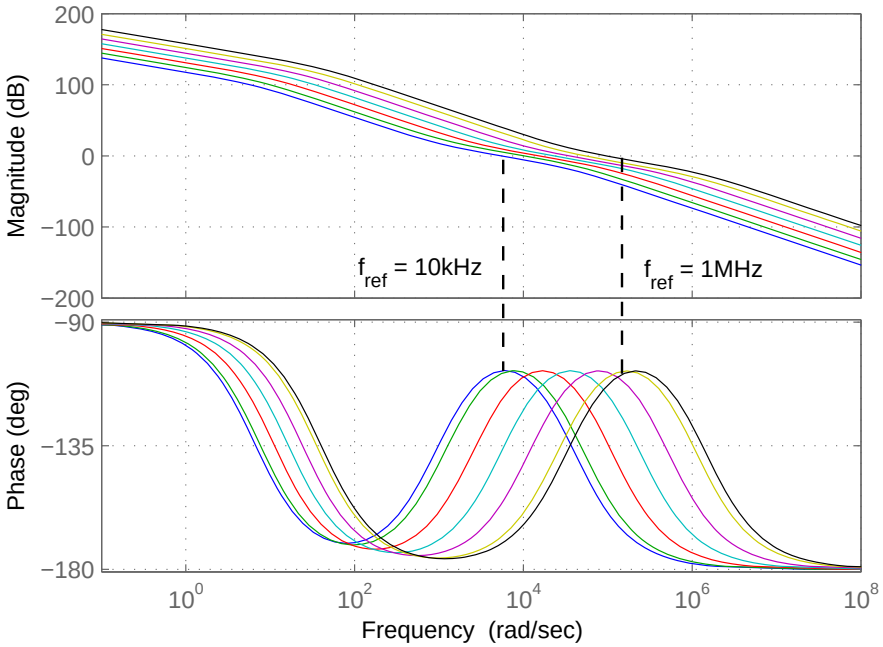


Figure D-5: PLL loop transmission $L(s)$ as a function of the input frequency f_{ref} when I_0 is adjusted such that the loop bandwidth becomes proportional to f_{ref}.

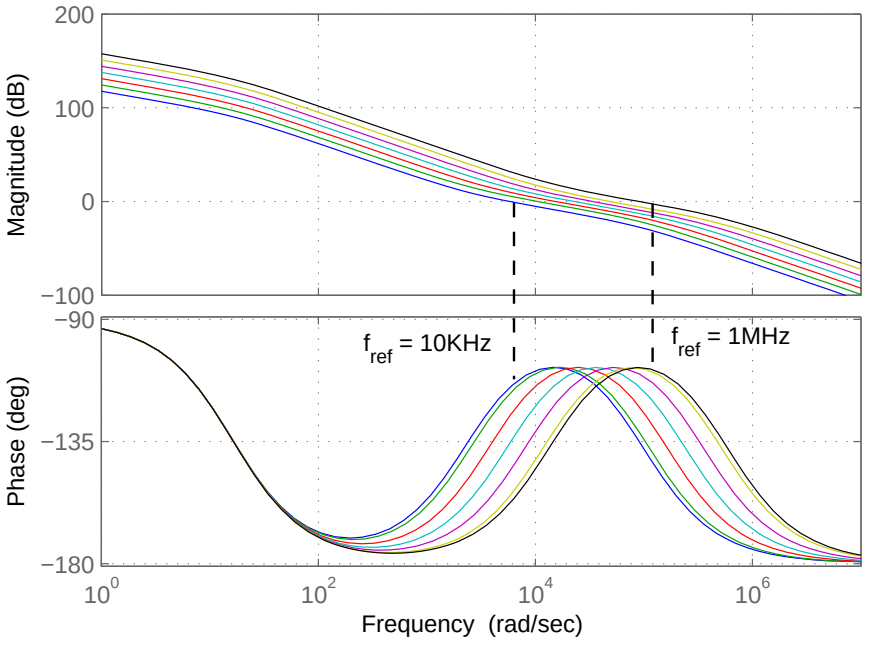


Figure D-6: PLL loop transmission $L(s)$ as a function of the input frequency f_{ref} when I_0 is kept constant, which makes the loop bandwidth proportional to $\sqrt{f_{ref}}$.

The easiest way to get the desired behavior is to keep C_L fixed and vary $R_z \propto 1/\sqrt{f_{ref}}$ using the simple circuit shown in Figure D-7. The transistor in series with C_L acts as a linear resistor R_z since it carries no DC current, has zero V_{DS} and is therefore in the linear (triode) regime of operation. Assuming that both transistors are identical and sized such that they are in strong inversion, we have

$$\begin{aligned} I_z &= \frac{\beta}{2}\left(V_{GS} - V_T\right)^2 \\ R_z &= \frac{1}{\beta\left(V_{GS} - V_T\right)} \end{aligned} \tag{D.7}$$

where $\beta = \mu C_{ox} W/L$. Solving the two equations above, we get

$$R_z = \frac{1}{\sqrt{2\beta I_z}} \tag{D.8}$$

In order to make $R_z \propto 1/\sqrt{f_{ref}}$ as desired, we set $I_z \propto f_{ref}$ by using an $f \rightarrow I$ converter.

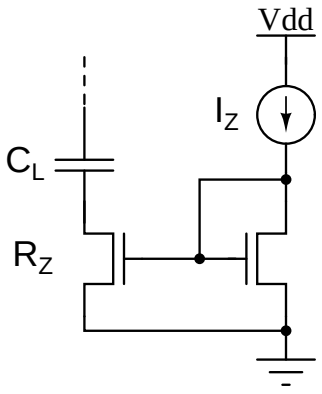


Figure D-7: A simple circuit that adjusts the frequency of the loop-stabilizing zero to maintain approximately constant phase margin as f_{ref} varies.

Finally, we should note that any adaptive scheme that feeds forward information about the input frequency to adjust loop parameters must perform such adjustments on a timescale much faster than the settling time of the loop to be effective. However, all analog frequency estimation schemes (i.e., $f \rightarrow I$ converters) ultimately look like low-pass filters. As a result, there is an obvious trade-off between estimation speed

and the amount of steady-state ripple present in the output. Such ripple is undesirable because some of it will appear in the CCO control voltage and get converted to clock jitter.

Digital frequency estimation strategies are preferable in this case because it is easy to get rid of steady-state ripple[2]. Imagine that we have access to a clock f_{slow} that is significantly slower than f_{ref}, but much faster than the loop bandwidth, which is approximately $\omega_c/(2\pi)$. To detect the value of f_{ref}, we count the number of f_{ref} edges within one period of f_{slow}. This number, N, is proportional to f_{ref} and can be used to control a current DAC that sets loop parameters like I_0 or I_z. However, because of the fact that f_{slow} will not in general be synchronous with f_{ref}, it is easy to see that the count will fluctuate between N and $N+1$ in steady state. This phenomenon is an exact analog of the steady state ripple in an analog frequency estimator. We can, however, easily eliminate the ripple in this case by using a little digital logic. A copy of the count obtained during the previous f_{slow} cycle is held in memory, and is only updated if the current count differs from it by more than 1 in either direction. We have implemented this strategy in circuit form using 5-bit counters and DACs.

D.1.4 Circuit design

In this section, we describe the circuit-level implementation of our PLL design. We must first mention that we have not yet implemented the feed-forward adjustment of loop-filter time constant that was described in the previous section and shown in Fig. D-4. We plan to extend our work in this direction soon. For now, however, I_0 represents a fixed external current source. The loop was found to operate normally at a power supply voltage of only 0.6V when the 'typical' process corner was used.

Digital circuits

The digital circuits in our design (the phase detector and clock divider) did not prove to be limiting factors for low-voltage operation, and were thus fairly standard designs.

[2]One can of course imagine analog circuits that also eliminate ripple, but they usually end up being quite complicated.

Static logic was nevertheless used in both combinational and sequential circuits in order to ensure robust operation at low V_{DD}. Proper operation was also verified at all process corners.

We have used Hogge's phase detector and cascaded divide-by-2 cells, built using D-registers, to form the clock divider. However, in our current design N, the divide value, is fixed at 2. We want to eventually implement a programmable frequency divider in the feedback path. This will allow our design to generate a variety of clock frequencies from a single input signal.

Loop filter

Instead of the usual charge pump and passive loop filter that is commonly used in IC PLL's, we use an interesting exponential state space (log-domain or current-mode) integrator circuit as the loop filter. The circuit is shown in Fig. D-8 and is based on the log-domain integrator described in [231]. The circuit is surprisingly similar to a normal differential amplifier. It accepts current inputs I_{i1} and I_{i2} that are log-compressed into voltages V_{i1} and V_{i2} and fed into the differential pair. However, the common source node V_s of the differential pair is low-impedance and an adjustable level shifter has been added between V_{out} and V_s. The voltage V_{out} is exponentiated by a transistor to produce the output current I_{out}. It should be noted that in order for this circuit to operate properly, all transistors have to be below threshold (so that they exhibit exponential I-V characteristics) and must be saturated. Assuming these conditions are satisfied, the circuit equations are given by

$$\begin{aligned} C_L \tfrac{dV_{out}}{dt} &= I_{D0} \exp\left(\tfrac{-V_s}{\Phi_T}\right) \left[\exp\left(\tfrac{\kappa V_{i1}}{\Phi_T}\right) - \exp\left(\tfrac{\kappa V_{i2}}{\Phi_T}\right)\right] \\ I_0 &= I_{D0} \exp\left(\tfrac{\kappa V_{out} - V_s}{\Phi_T}\right) \end{aligned} \tag{D.9}$$

where I_{D0} is a constant (the 'pre-exponential current') and is the same for all NMOS transistors in the circuit (i.e., we have assumed matched devices). Eliminating V_s from (D.9) and substituting $I_{i1,2} = I_{D0} \exp\left(\tfrac{\kappa V_{i1,2}}{\Phi_T}\right)$ and $I_{out} = I_{D0} \exp\left(\tfrac{\kappa V_{out}}{\Phi_T}\right)$, we get

$$\frac{dI_{out}}{dt} = \frac{\kappa I_0}{C_L \Phi_T} (I_{i1} - I_{i2}) \tag{D.10}$$

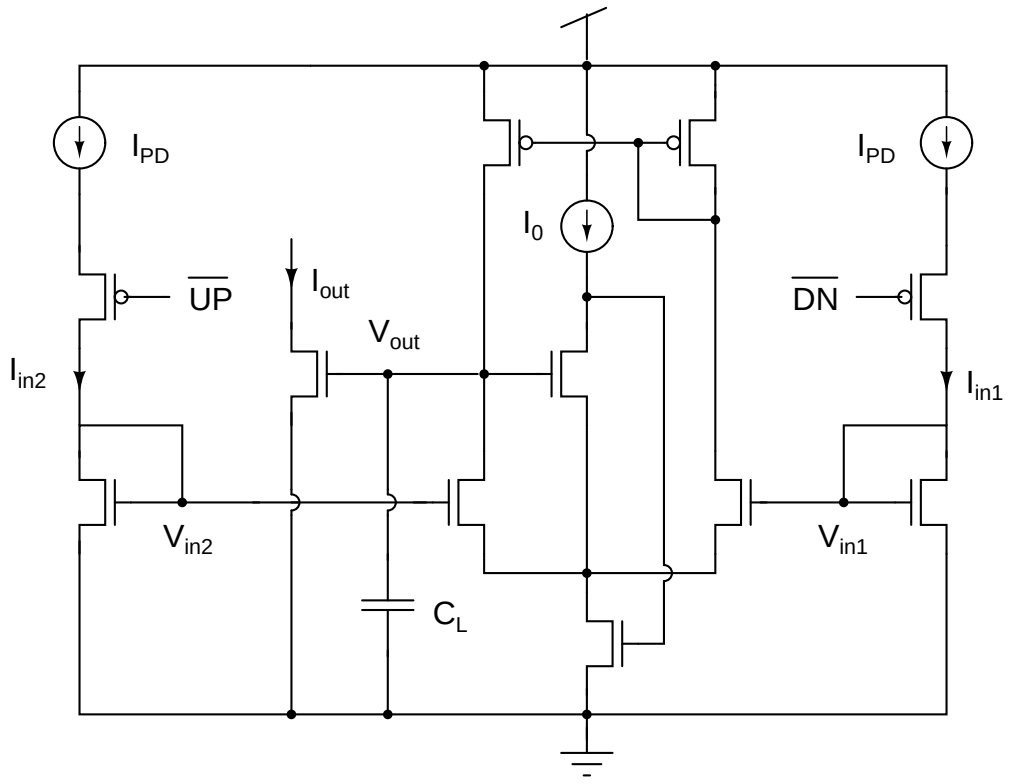


Figure D-8: Current-mode (exponential state space) integrator used in the PLL loop filter.

The s-domain transfer function of the circuit is thus given by[3]

$$\frac{I_{out}}{I_{i1} - I_{i2}} = \frac{\kappa I_0}{C_L \Phi_T} \frac{1}{s} \tag{D.11}$$

Eqn. D.11 behaves as an ideal differential current mode integrator with infinite DC gain. In practice the DC gain of the circuit is finite (it is set by the finite output resistance of the transistors). This changes the transfer function of the circuit to

$$\frac{I_{out}}{I_{i1} - I_{i2}} = \frac{A}{1 + s\tau A} \tag{D.12}$$

where A is the DC gain and $\tau = \frac{C_L \Phi_T}{\kappa I_0}$ is a time constant. As $A \to \infty$, (D.12) reduces to (D.11), as expected. In other words, like all real integrators, the circuit acts like a low-pass filter with high DC gain, and not an ideal integrator. In our application, $I_{in1,2} \in (0, I_{PD})$, where I_{PD} is a constant current. Therefore, the differential input current (controlled by the phase detector UP and DN outputs) is $I_{in1} - I_{in2} \in (-I_{PD}, I_{PD})$. Finally, we note that the circuit's primary time constant $A\tau$ can be varied electronically by controlling I_0. This property opens up interesting possibilities for adaptive PLL architectures where the properties of the loop filter are dynamically controllable.

A potential problem with the integrator as implemented is that its core works at much lower currents than I_{PD}. This can be seen from (D.9). The currents through the NMOS differential pair are given by $I_{PD} \exp(-V_s/\Phi_T) \ll I_{PD}$ if the corresponding input current source is switched ON. Ideally, this does not affect the overall transfer function, since the output transistor exponentiates V_{out} to the right output current. However, in practice parasitic poles, particularly those contributed by the PMOS current mirror, slow down and start affecting the circuit response if V_s increases. In order to fix this problem, either I_{PD} must be increased, or a source voltage offset built into the diode-connected transistors that compress $I_{i1,2}$ to $V_{i1,2}$ to compensate for the fact that $V_s > 0$.

[3]It is interesting to note that this is a *large signal* transfer function since it was not derived by linearizing the exponential I-V characteristics of transistors about a DC operating point.

In this section, we describe our current mode loop filter. The loop filter circuit is shown in Fig. D-9. It uses the same basic exponential state-space integrator core shown in Fig. D-8, but significant changes have been made to improve performance. These changes are described below:

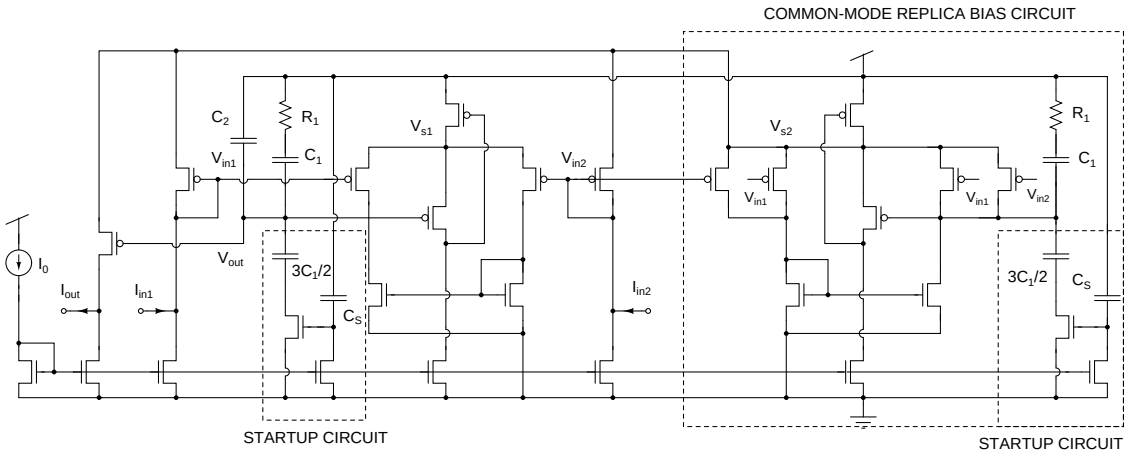


Figure D-9: Current-mode loop filter used in the PLL. Startup and common-mode replica biasing circuits are shown, but well biasing details for the PMOS transistors have been omitted.

- The NMOS and PMOS transistors in the core have been flipped. The circuit now uses a PMOS differential pair and level shifter and an NMOS current mirror. This means that all internal voltages are referenced to V_{DD} and not to ground. This increases rejection of power supply noise.

- The current mirrors carrying the input currents $I_{in1,2}$ have DC bias currents added to them. This makes it possible for $I_{in1,2}$ to be negative. We don't actually need this feature in the PLL since $I_{in1,2} \in (\alpha I_{cco}, 0)$ and are thus always positive. It does, however, make the circuit more general-purpose.

- Similarly, I_{out} also has a DC bias current added to it. This allows the circuit to both source and sink current if required. Again, this is not important for us because we always have $I_{cco} = \beta I_{out} > 0$ (β is a constant).

- A common-mode replica bias circuit (shown inside the box in Fig. D-8) has been added. This circuit adjusts V_{s2}, the source terminal voltage of the input

mirrors and the output transistor (that exponentiates V_{out} into I_{out}) so that it is approximately equal to V_{s1}, the source voltage of the integrator core *in the quiescent state*, i.e., when $V_{i1} = V_{i2}$. This makes the core operate at similar current levels to $I_{in1,2}$ and fixes one of the problems with our original integrator design.

- The transfer function of the circuit has been modified by changing the passive load at V_{out} from a single capacitor C_L. The new load has a large capacitor C_1 in series with a resistor R_1, and a smaller capacitor C_2 in parallel. The R_1-C_1 branch provides the loop-stabilizing zero, and C_2 is responsible for the high-frequency pole that we want to place beyond loop crossover. It can be shown that the transfer function of the loop filter is given by

$$H(s) = \frac{I_{out}}{I_{in,1} - I_{in,2}} \approx \frac{A\left(1 + sC_1R_1\right)}{1 + sC_1\left(R_{eff} + R_1\right) + sC_2R_{eff}\left(1 + sC_1R_1\right)} \tag{D.13}$$

  where $R_{eff} = A\Phi_T/\left(\kappa I_0\right)$ is the effective resistance of the current-mode integrator, and A is the DC gain of the circuit. R_{eff} can be controlled by changing the bias current I_0.

- We have also addressed the start-up issues inherent in this circuit. Essentially, V_{out} needs to be initialized to a reasonable value somewhere in the middle of the rails when the power supply is switched on. This ensures that all transistors are saturated and no internal nodes are stuck at V_{DD} or ground. We ensure that this happens by using a capacitively-coupled start-up circuit.

  C_S is a small capacitor that is initially uncharged. This makes the gate voltage of the transistors in series with the capacitor labeled $1.5C_1$ high for a short time after the power supply is switched on. While these transistors are on, V_{out} is set by the capacitive divider formed by C_1 and $1.5C_1$ connected between V_{DD} and ground. V_{out} is thus rapidly set to $V_{DD} \times C_1/\left(C_1 + 1.5C_1\right) = 0.4V_{DD}$ while the power supply ramps up. After a short time C_S charges up, the switches

in series with $1.5C_1$ switches off and the circuit functions normally. A similar circuit is connected at the output of the common-mode replica bias circuit that sets V_{s2}. These innovative start-up circuits work very well in practice. The only disadvantage is that they add a large amount of extra capacitance to the loop filter, thereby increasing its layout area.

- Finally, the range of currents the loop filter can handle is limited by two factors: the power supply voltage and the onset of strong inversion. In practice, the first issue dominates at low V_{DD}: the maximum source-gate voltage V_{SG} available on the PMOS transistors that sink or source current, and thus the current itself, is limited by the value of V_{DD}. Larger values of V_{SG} cause other transistors, such as the NMOS mirrors, to come out of saturation (this causes the transfer function of the circuit to change in undesirable ways). Since I_{cco} has to be supplied by the loop filter, this limits the maximum frequency that can be synthesized by the loop at a given value of V_{DD}.

  We can improve the situation by lowering the threshold voltage V_{th} of the PMOS transistors. This enables them to carry more current at the same value of V_{SG}. In order to do this, NMOS source followers carrying a fixed bias current ($= I_0$ in this case) are used to forward bias the wells of each PMOS transistor in the circuit by about 300mV. This reduces V_{th} by about 100mV and gives us approximately an extra decade of available current range. Well-biasing details have not been shown in Fig. D-9 for the sake of simplicity.

Fig. D-10 shows simulated small-signal (AC) transfer functions of the current-mode loop filter for different values of I_0. I_0 was allowed to vary between 1.6nA and 32nA. We used the following component values: $C_1 = 100$pF, $R_1 = 300$kΩ, $C_2 = 2.5$pF and $C_S = 2.5$pF. As expected, changing I_0 does not significantly affect the DC gain $A \approx 45$dB, but does change the primary time constant $A\tau$, since $\tau = C_1\Phi_T/(\kappa I_0) \propto 1/I_0$. As a result, the unity-gain frequency ω_u of the filter is $\propto I_0$. Fortunately, this is exactly the behavior we need for implementing the feed-forward loop transmission adjustment scheme shown in Fig. D-4. We therefore plan to use

the same loop filter circuit in our next design iteration (where this scheme shall be implemented). Finally, the presence of both the loop-stabilizing zero and the additional high-frequency pole can easily be seen from the transfer functions.

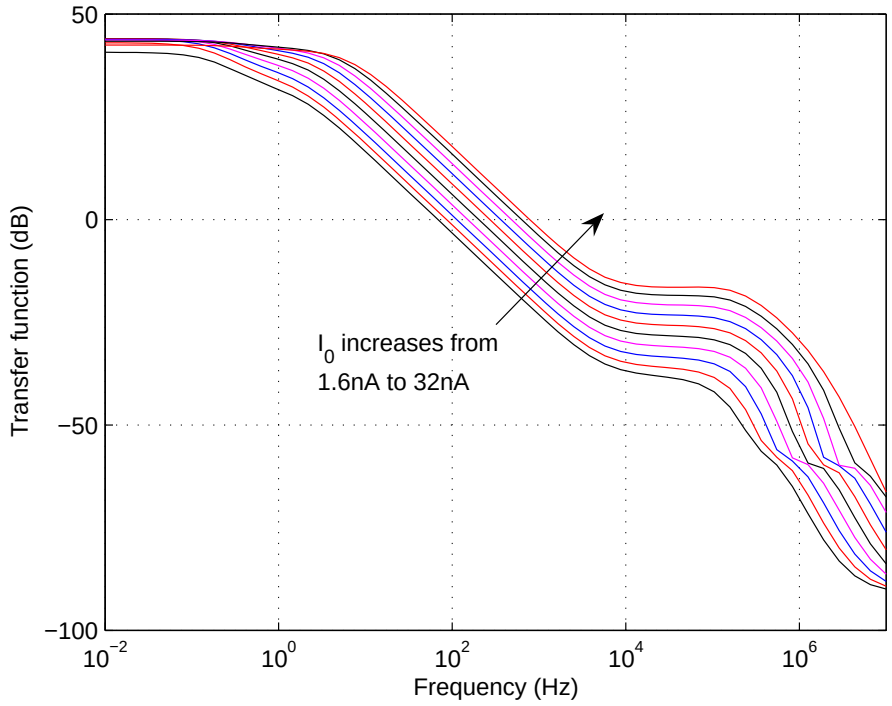


Figure D-10: Simulated AC transfer functions of the current-mode loop filter for I_0 varying between 1.6nA and 32nA.

Current-controlled oscillator

We want to design a ring oscillator that possesses the following qualities, which make it desirable for PLL applications: power supply independence and rail-to-rail voltage swings. An oscillator whose frequency of oscillation does not depend on V_{DD} results in the PLL having good power supply rejection, which is important for many applications. Rail-to-rail (0 to V_{DD}) swings of the PLL maximize noise margins, reduce phase noise and improve compatibility with digital circuits. Unfortunately the two requirements are contradictory. An oscillator that has an output amplitude equal to V_{DD} is charging and discharging capacitors through V_{DD}, so its oscillation frequency will depend on the value of V_{DD}. Conversely, a supply-independent oscillator must have

an amplitude of operation that is independent of V_{DD}, and thus cannot be V_{DD} itself. Having failed to reconcile these requirements, we therefore decided to implement two separate CCO designs. Each circuit was designed to satisfy one of the properties above; we plan to fabricate and test both of them so that their performances can be compared.

The two current-controlled oscillator circuits are shown in Fig. D-11. The circuit on the left, which we refer to as the CCO-I structure, uses a single current source I_{cco} that fixes the total amount of current drawn from the supply. The current source, constructed in Fig. D-11(a) using a simple current mirror, can be cascoded for higher output resistance. This source supplies current to a ring oscillator that uses standard CMOS inverters as delay cells (a three-stage version is shown). The maximum possible oscillation amplitude is limited to $V_S < V_{DD}$; in practice $V_{osc} < V_S$ and increases with the oscillation frequency f_{cco}. For a given value of f_{cco}, V_{osc} increases with C_{cco} (i.e. increased bias current I_{cco}, since f_{cco} is fixed). A bypass capacitor (not shown) is usually added at V_S to reduce coupling of power-supply noise. The power consumption of this circuit is less than the CCO-II structure shown in Fig. D-11(b) for the same oscillation frequency since V_{osc} is lower. However, this results in larger phase noise. Also, an additional amplifier is required to amplify V_{osc} to logic levels (rail-to-rail swings) before it can be fed into any digital circuits. This amplifier consumes extra power and may negate any power savings in the CCO itself. In our PLL design we use a capacitively-coupled common-source amplifier with resistive feedback at the output of the CCO.

The second CCO circuit, shown in Fig. D-11(b), shall be referred to as the CCO-II structure. A current I_{cco} sets bias voltages V_P and V_N which limit the amount of current that each delay cell can draw from the supply. Thus I_{cco} controls the propagation delay of each stage, and consequently the oscillation frequency. The delay cells are four-transistor inverters similar to those used in C^2MOS (clocked-CMOS) logic. This oscillator usually swings rail-to-rail, i.e., $V_{osc} \approx V_{DD}$. Therefore, no additional amplifiers are required. We have designed and simulated both circuits. CCO-I usually takes longer to start up. If I_{cco} or C_{cco} is too small, it may not start up

at all. On the other hand, its frequency of oscillation is almost independent of V_{DD} for $V_{DD} > 0.6V$, while CCO-II produces $f_{osc} \propto 1/V_{DD}$ when I_{cco} is fixed. However, it has never demonstrated start up problems. Finally, both oscillators have extremely linear control characteristics, i.e., $f_{cco} \propto I_{cco}$ over several orders of magnitude.

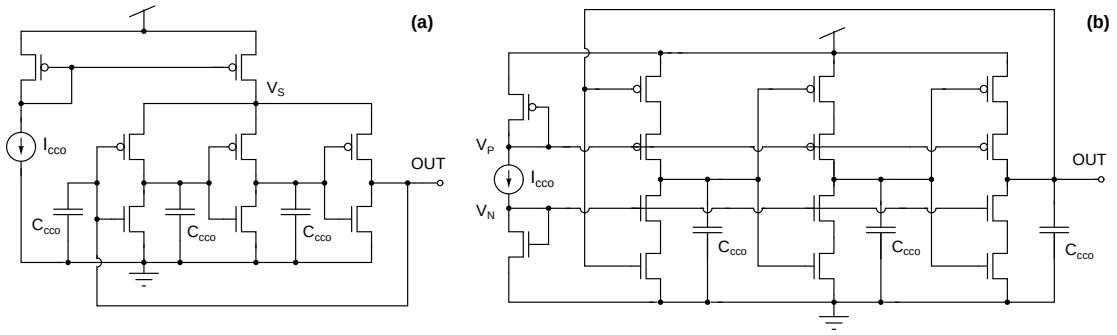


Figure D-11: Current-controlled oscillator (CCO) circuits, (a) CCO-I structure and (b) CCO-II structure.

D.1.5 Simulated performance

Our PLL design was simulated at the transistor level using SPICE parameters available from UMC[4] for their 0.18μm mixed-mode CMOS process. Performance was verified at all the process corners and using both types of CCO. Fig. D-12 shows the simulated transient response of V_{out}, the loop filter output voltage that is exponentiated into I_{cco}. The CCO-I circuit was being used. The input was a square wave with a time period of 3μs, i.e., $f_{ref} = 333$kHz. The supply voltage was $V_{DD} = 0.6V$ and the average power consumption was 420nW. The supply was ramped up from zero so that the start-up behavior of the loop could be observed. The fast initial transient in the response is due to the start-up circuits inside the loop filter. These circuits rapidly pull up V_{out} to $\approx 0.4V_{DD} = 250$mV (as expected) and then shut off. The control voltage then shows a long nonlinear transient, including several cycle slips, before the loop eventually locks. The eventual linear response just before lock (around 1.5ms into the simulation) indicates a phase margin of about 50°. This is somewhat

[4]United Microelectronics Corp., Taiwan.

less than predicted by the linear loop model, but is hardly surprising considering the sampled nature of the loop and the presence of unmodeled parasitic poles.

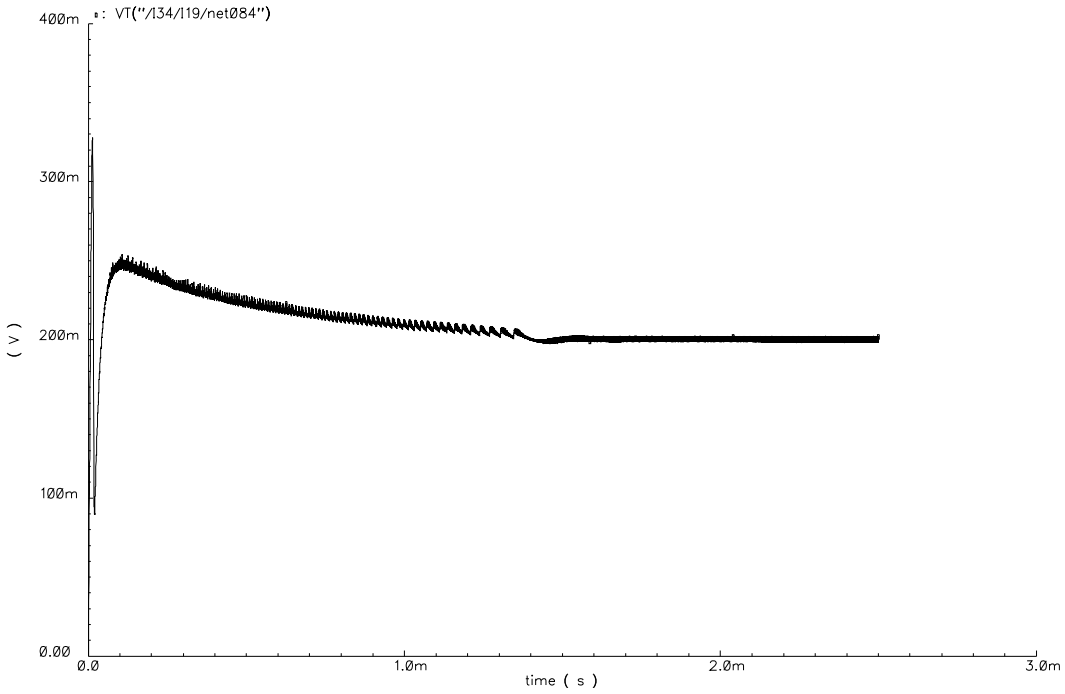


Figure D-12: Simulated transient response of the loop filter output voltage, showing the PLL (using the first CCO design) coming into lock.

Fig. D-13 shows a similar transient response as Fig. D-12, but this time the loop was using the CCO-II circuit. The input frequency was the same as before, and the power consumption of the loop was 670nW ($V_{DD} = 0.6V$). The two waveforms look very similar close to lock, indicating that the choice of CCO does not significantly affect the linear dynamics of the loop. However, the nonlinear approach to lock takes longer in this case. This cannot be predicted by the linear system model, and is instead controlled by the slewing rate of V_{out}, the eventual operating point of the loop in lock and other factors. The final operating point of V_{out} was further away from $0.4V_{DD}$ than in the previous case, making approach to lock slower. There are also indications of a long-tailed transient (usually associated with a pole-zero pair in the loop transmission) in V_{out} after locking occurs around 3ms into the simulation. The cause of this behavior is currently unknown; it is being investigated.

Fig. D-14 shows input f_{ref}, synthesized clock f_{out} and recovered data RD wave-

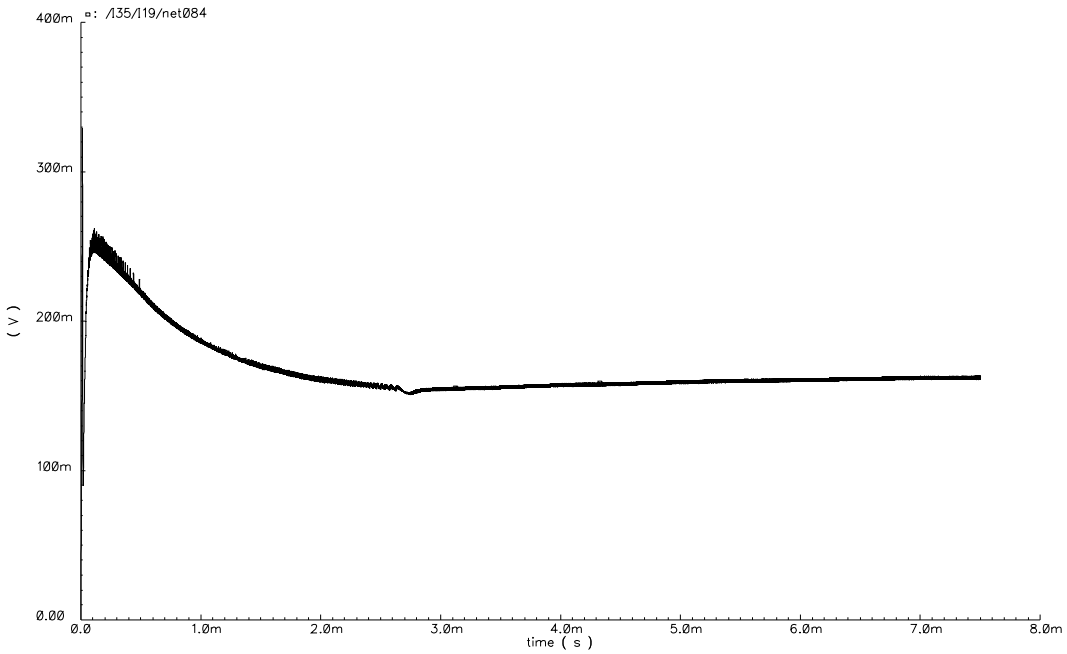


Figure D-13: Simulated transient response of the loop filter output voltage, showing the PLL (using the second CCO design) coming into lock.

forms from the same transient simulation as Fig. D-13. The figure also shows the loop filter output V_{out}. The loop was locked by this time, so V_{out} is constant except for a small triangular ripple at the same frequency as f_{out}. As expected, edges of the recovered data waveform RD line up with the synthesized clock f_{out}. Also, RD and f_{ref} match (except for a constant phase shift), indicating that the transmitted data has been successfully recovered. The loop, as designed, can thus be used for CDR applications.

D.2 Improving Circuit Performance with Switching Control

Switching control is an example of variable-structure control. In such systems the controller can switch at any instant between members of a set of continuous functions of the state of the plant [294]. Variable structure systems can possess properties not present in any of their constituents. For example, stable phase-plane trajectories, so

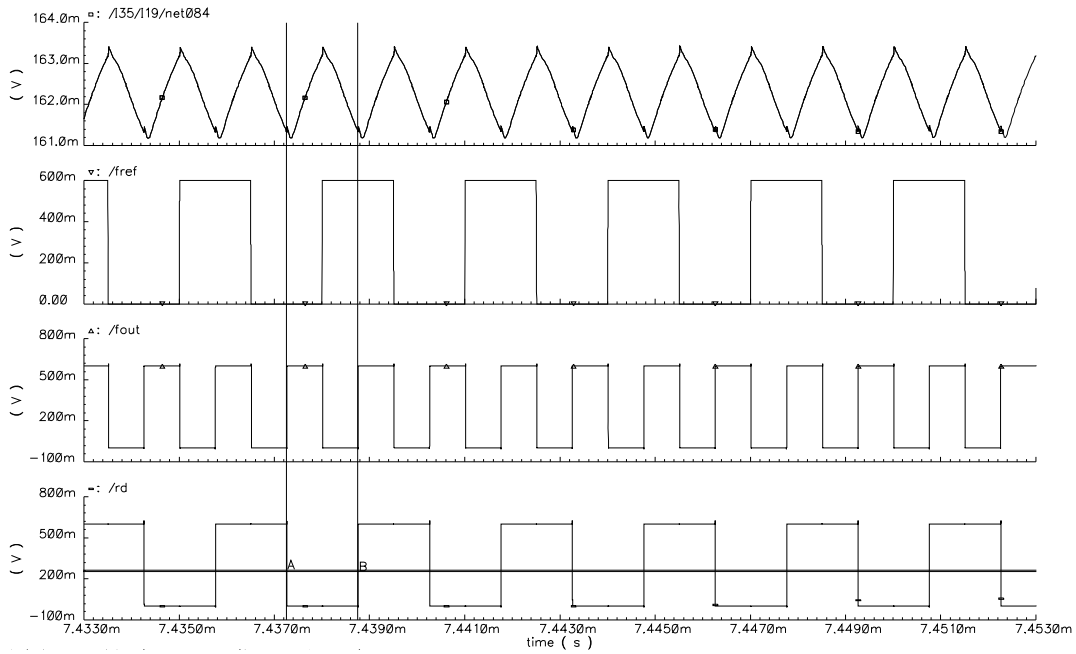


Figure D-14: Simulated waveforms produced by the PLL while locked. From the top: the loop filter output voltage V_{out}, input data f_{ref}, clock f_{out} and recovered data RD.

called "sliding modes", can emerge [54,322]. An important advantage of sliding mode control is that the behavior of the system along sliding modes is robust, i.e., largely independent of plant parameters and disturbance inputs. Instead of synthesizing a sliding mode, we shall design a switching controller that stabilizes a system by placing it on a trajectory that is *already present* in one of the constituent structures.

D.2.1 Switching Controller

Switching controllers were first studied in the context of control systems containing relays and other nonlinear components. Santarelli considered a switching control problem that was simple enough to treat analytically [252,253]: the plant was assumed to be second-order, linear and time invariant (LTI), and a "supervisor" could choose between two feedback laws based on the state of the plant. To further simplify the problem, the two feedback laws were restricted to be constant gains K_0 and K_1. To design the controller we need to specify the structure of the supervisor and the values of K_0 and K_1.

For the special case when the plant has no zeros in its transfer function (i.e., has relative degree two), Santarelli devised an algorithm for synthesizing optimal switching controllers and proved that they were L_2-gain stable[5]. The controllers are optimal in the sense that they maximize the value of a parameter that measures the rate at which state trajectories converge to the origin, i.e., they minimize settling time.

We shall now design an optimal switching controller for plants that are well-modeled as double integrators. Analog circuits that satisfy this criterion are common and important, including two-stage operational amplifiers and second-order phase-locked loops. A block diagram of the system is shown in Figure D-15. In order for the plant to be considered "close" to a double integrator both stages must have high DC gain, i.e., $A_1 \gg 1$ and $A_2 \gg 1$. Choosing feedback gains $\in [-1, 1]$ simplifies circuit implementation since we will not need variable amplifiers or attenuators inside the feedback loop: a couple of switches will suffice. We therefore restrict ourselves to the following values: $K_0 = 1$, $K_1 = -1$, i.e., $s(t) \in [-1, 1]$.[6].

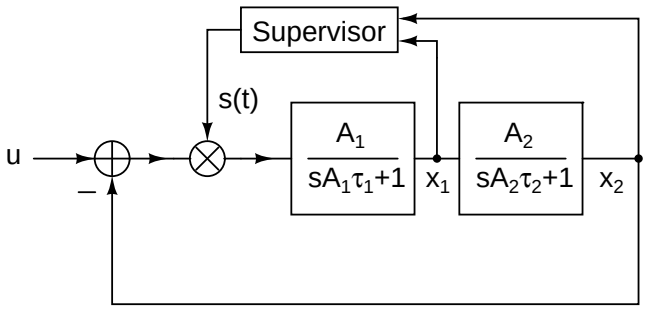


Figure D-15: Block diagram of the two-pole LTI plant and two-state switching controller considered in this section.

Graphical tools such as root locus and phase-plane plots allow us to visualize the behavior of the switching controller. The root-locus plot of a two-pole system with two real open-loop poles, such as the plant shown in Figure D-15, is shown in Figure D-16 for both possible values of $s(t)$. When $s(t) = -1$ (positive feedback)

[5]Intuitively, L_2-gain stability implies that a bounded-power input to the system will result in bounded-power output.

[6] The analysis that follows can be generalized to the case of asymmetric gains. Santarelli's work suggests that the symmetric gain case provides a lower bound to the performance of the switching architecture, so the generalization is worth exploring.

we end up with two real poles, one stable and the other unstable. When $s(t) = +1$ (negative feedback), two stable, complex poles are produced. As the DC gains A_1 and A_2 increase, the system approaches an ideal double integrator and the complex poles become purely imaginary, i.e., approach the $j\omega$ axis.

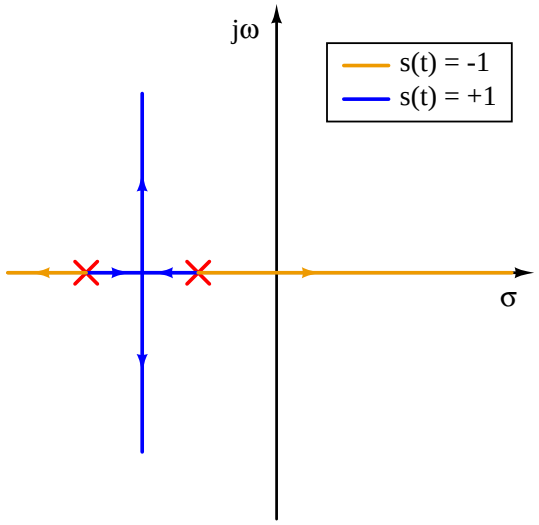


Figure D-16: Root locus of a two-pole system under positive and negative feedback ($s(t) = -1$ and $s(t) = +1$, respectively).

We denote the two state variables as x_1 and x_2. Note that we have implicitly assumed that the supervisor has access to the full state of the plant. Thus an observer is not needed to estimate the value of the internal state variable x_2. This is usually a safe assumption when the plant is an integrated circuit. Since the double integrator has presumably been put into feedback to track an input $u(t)$ we now define the new variables $z_2 = x_2 - u = -e$, where $e(t)$ is the error signal fed into the multiplier, and $z_1 = x_1$. The plane with z_1 and z_2 as axes is known as the phase plane. Under negative feedback the two nearly imaginary poles result in a phase-plane trajectory that spirals inward or outward from the origin. Under positive feedback the origin becomes a saddle point, i.e., the intersection of the eigenvectors w_s and w_u corresponding to the stable and unstable pole, respectively. The basic idea behind the control scheme can now be explained. Initially we use negative feedback. The complex eigenvectors cause us to spiral around the phase plane. We wait until the stable positive-feedback eigenvector w_s is reached, and then switch to positive feedback. Since the system

is already on its stable eigenvector, it smoothly converges to the origin (equilibrium state). A typical phase-plane trajectory in shown in blue in Figure D-17.

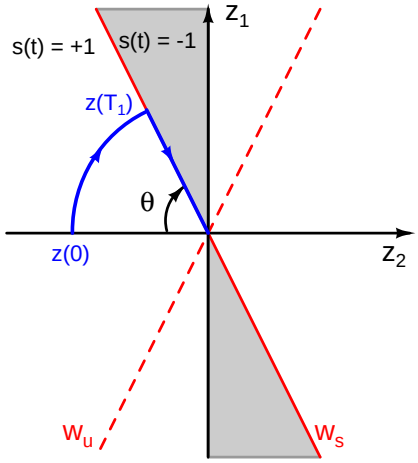


Figure D-17: A phase plane portrait of the switching controller, showing the control regions (gray and white), eigenvectors in positive feedback (red) and a typical trajectory during an input-tracking task (blue).

Figure D-17 shows that positive feedback is applied not only when the system is actually on w_s, but within the cone enclosed by w_s and the line bifurcating the angle between w_s and w_u, i.e., the z_1 axis. Choosing one side of the cone as w_s maximizes rate of convergence to the origin. We can in fact stabilize the system if this side of the cone was a line inclined at a smaller angle to the z_2 axis than w_s. It is easy to show that phase trajectories under both signs of feedback will converge to such a line, creating a sliding mode. However, Santarelli showed that the resulting rate of convergence will be slower than in our case, where the switching line is chosen to be precisely w_s. The choice of the other side of the cone is not critical but provides some robustness to switching delays.

"Chattering" is a well-known phenomenon in discontinuous control schemes like switching and sliding mode control. Chattering is defined as high control activity in steady-state. Fundamentally, it arises because of the finite speed at which any real controller can switch, i.e., change state. As a result of finite switching speed asymptotic convergence of the system trajectory cannot be guaranteed. However, we can guarantee that the trajectory remains within an ϵ-bound of the asymptotic

(ideal) solution [54]. In most cases, the system will reach a steady-state marked by small periodic deviations from equilibrium, known as ripple. The amplitude of the ripple decreases as the controller becomes faster, leading to a trade-off between speed/power and precision. Various chattering-reduction or elimination schemes have been proposed in the literature, including fuzzy control laws, fuzzy switching boundaries, higher-order sliding modes and systems with strictly continuous (non-switching) control laws [318]. We have not implemented any of these schemes yet.

D.2.2 Comparison with Linear Controllers

In this section we compare the performance of the switching controller with the traditional linear technique used to stabilize two-pole amplifiers, namely Miller, split-pole or minor-loop compensation. A small-signal model of a generic two-pole amplifier with Miller compensation is shown in Figure D-18. Current fed back through the compensation capacitor C_f connected across the second stage splits the poles. Because C_f is a bidirectional element, it also introduces a feed-forward right half-plane zero. The resistor R_f in series with C_f cancels this zero, thereby increasing the phase margin. Precise cancelation occurs when $R_f = 1/g_{m2}$. We shall assume this value for the remainder of this section.

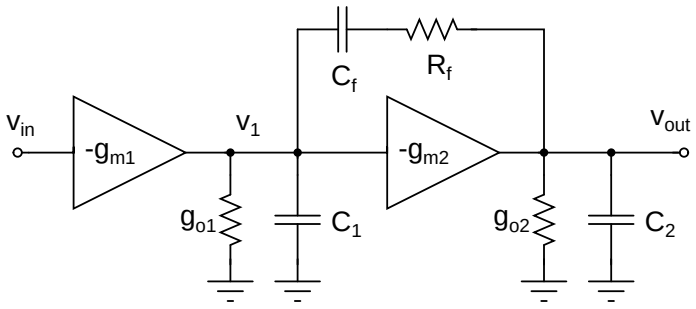


Figure D-18: A small-signal model of a generic two-pole amplifier with Miller compensation provided by C_f and R_f.

Circuit analysis aided by the symbolic capabilities of Mathematica tells us that the transfer function of the circuit shown in Figure D-18 is given by

$$\frac{v_{out}}{v_{in}} = \frac{g_{m1}g_{m2}^2}{s^3C_1C_2C_f + s^2a_2 + sa_1 + g_{m2}g_{o1}g_{o2}} \tag{D.14}$$

where

$$\begin{aligned} a_1 &= \left[\left(g_{m2}+g_{o2}\right)\left(g_{m2}+g_{o1}\right)C_f + g_{o2}g_{m2}C_1 + g_{o1}g_{m2}C_2\right] \\ a_2 &= \left[\left(g_{m2}+g_{o2}\right)C_fC_1 + g_{m2}C_1C_2 + \left(g_{m2}+g_{o1}\right)C_2C_f\right] \end{aligned} \tag{D.15}$$

Equation (D.14) can be simplified if the following assumptions are made:

- The frequency is low enough for the s^3 term in the denominator to be neglected.
- The DC gain of the second stage is high, implying that $g_{m2} \gg g_{o1}$, $g_{m2} \gg g_{o2}$, $C_fg_{m2} \gg C_1g_{o2}$ and $C_fg_{m2} \gg C_2g_{o1}$.

The simplified transfer function can be expressed in the standard form for a second-order system:

$$\frac{v_{out}}{v_{in}} = \frac{A}{s^2\tau^2 + s\tau/Q + 1} \tag{D.16}$$

where

$$\begin{aligned} A &= \frac{g_{m1}g_{m2}}{g_{o1}g_{o2}} \\ \tau &= \sqrt{\frac{C_fC_1 + C_1C_2 + C_2C_f}{g_{o1}g_{o2}}} \\ Q &= \frac{\sqrt{g_{o1}g_{o2}}}{g_{m2}}\sqrt{\frac{C_fC_1 + C_1C_2 + C_2C_f}{C_f^2}} \end{aligned} \tag{D.17}$$

We see that increasing C_f will decrease Q, increasing the phase margin and damping of the closed-loop system. We have written a Mathematica program that uses the original, un-simplified transfer function (D.14) to evaluate the step-response of the compensated amplifier configured as a follower in unity feedback. Our goal is to

find the settling time and peak overshoot as a function of amplifier parameters, particularly g_{m1}, g_{m2} and C_f. Since we can always decrease the settling time by burning more power, we impose a power-consumption constraint:

$$g_{m1} + g_{m2} = g_{m,tot} \tag{D.18}$$

where $g_{m,tot}$ is a constant. We also define the following parameters:

$$\begin{aligned} \alpha &= \frac{g_{m2}}{g_{m1}} \\ \beta &= \frac{\tau_2}{\tau_1} = \frac{C_2/g_{m2}}{C_1/g_{m1}} = \frac{1}{\alpha}\frac{C_2}{C_1} \end{aligned} \tag{D.19}$$

For convenience we shall assume from now on that all capacitances are in pF, conductances in μS and times in μs. Simulated unit step responses as a function of α are shown in Figure D-19 for $C_1 = 0.1$, $C_2 = 1$, $C_f = 1.1$ and $g_{m,tot} = 5$. The phase margin worsens as α decreases because the output pole becomes slower and contributes negative phase before crossover. Interestingly, settling time appears to be maximized for intermediate values of α. If α is low the system rises fast initially but then rings for a long time before settling down. On the other hand, if α is high the system is very stable and does not ring, but takes a long time to rise to its final value. The same type of variation in settling time is observed for other values of C_f and $g_{m,tot}$.

We now calculate the settling time of the same two-pole amplifier when the Miller compensation elements C_f and R_f are removed and switching control is used instead. The system may be modeled by Figure D-15, where $A_1 = g_{m1}/g_{o1}$, $A_2 = g_{m1}/g_{o1}$, $\tau_1 = C_1/g_{m1}$ and $\tau_2 = C_2/g_{m2}$. For simplicity, assume that $A_1 \gg 1$ and $A_2 \gg 1$, so that the stages are well modeled as pure integrators with transfer functions $1/(s\tau_1)$ and $1/(s\tau_2)$, respectively. It is easy to show that the closed-loop poles are $s = \pm j/\sqrt{\tau_1\tau_2}$ when $s(t) = +1$ (negative feedback) and $s = \pm 1/\sqrt{\tau_1\tau_2}$ when $s(t) = -1$ (positive feedback). The dynamics of the system are given by:

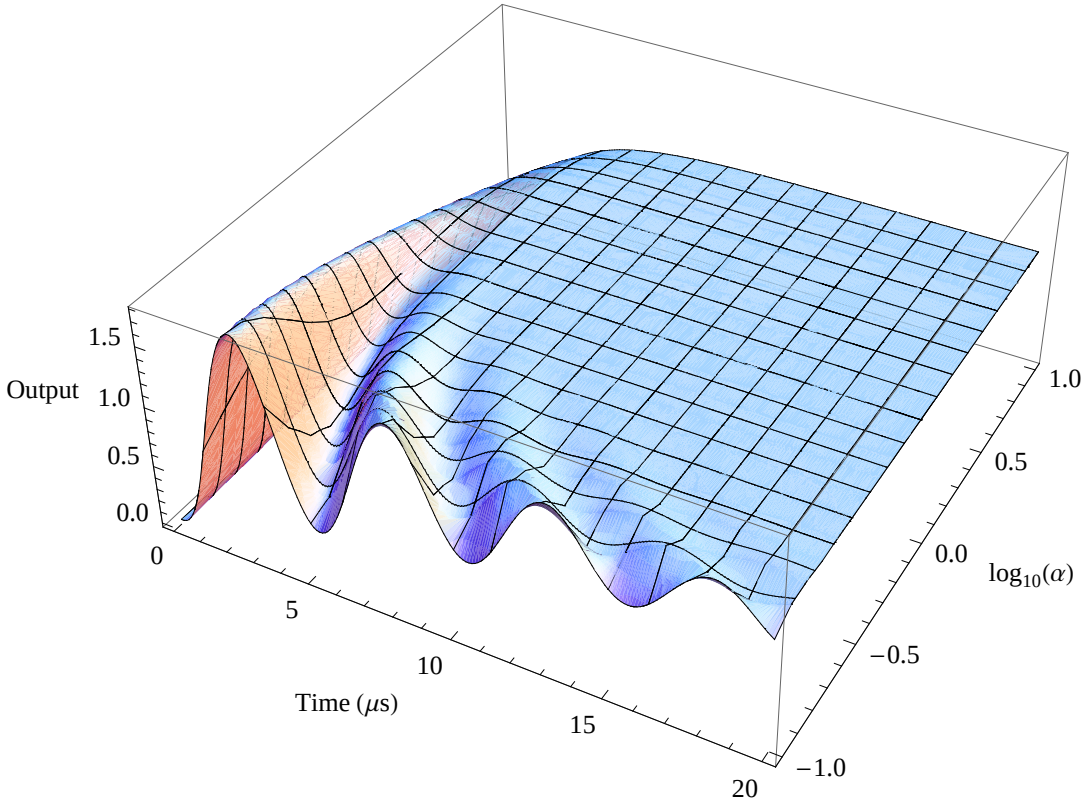


Figure D-19: Step responses of a Miller-compensated two-pole amplifier as a function of α, the ratio of second and first-stage trans-conductance, when their sum is held fixed.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & -s(t)/\tau_1 \\ 1/\tau_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} s(t)/\tau_1 \\ 0 \end{bmatrix} u \tag{D.20}$$

By defining $z_1 = x_1$ and $z_2 = x_2 - u$, as before, the equations above can be simplified to

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \end{bmatrix} = \begin{bmatrix} 0 & -s(t)/\tau_1 \\ 1/\tau_2 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{D.21}$$

The eigenvalues of the system are the solutions to the equation $\lambda^2 + s(t)/(\tau_1\tau_2) = 0$. When $s(t) = -1$ (positive feedback), they are given by $\lambda = \pm 1/\sqrt{\tau_1\tau_2}$. The corresponding stable and unstable eigenvectors w_s and w_u are given by

$$w_s = \begin{bmatrix} -\sqrt{\beta} \\ 1 \end{bmatrix}, w_u = \begin{bmatrix} +\sqrt{\beta} \\ 1 \end{bmatrix} \tag{D.22}$$

where $\beta = \sqrt{\tau_2/\tau_1}$, as before. The switching control law shown graphically in Figure D-17 can now be found explicitly. It is given by

$$s(t) = \begin{cases} -1 & \text{if} \quad z_2\left(z_2 + z_1/\sqrt{\beta}\right) \leq 0 \\ +1 & \text{if} \quad z_2\left(z_2 + z_1/\sqrt{\beta}\right) > 0 \end{cases} \tag{D.23}$$

In order to calculate the settling time T_s for unit step inputs we note that the system starts from the initial condition $(z_1(0), z_2(0)) = (0, -1)$. The trajectory for $t > 0$ can be divided into two parts. For $t < T_1$ the system is in negative feedback and follows a circular trajectory with oscillation period $\omega_{osc} = \sqrt{\tau_1\tau_2}$. This trajectory is followed until it hits w_s, i.e., an angle θ is covered, where $\tan(\theta) = \sqrt{\beta}$. Therefore T_1 is given by

$$T_1 = \frac{\theta}{\omega_{osc}} = \sqrt{\tau_1\tau_2}\tan^{-1}\left(\sqrt{\beta}\right) \tag{D.24}$$

For $t > T_1$ the system is in positive feedback and converges to the origin along the stable eigenvector w_s, which corresponds to an eigenvalue of $\lambda_s = -1/\sqrt{\tau_1\tau_2}$. Therefore both z_1 and z_2 decrease exponentially with time as $\exp(\lambda_s t) = \exp\left(-t/\sqrt{\tau_1\tau_2}\right)$.

In addition, since the trajectory for $t < T_1$ is circular, we have

$$z_1^2(T_1) + z_2^2(T_1) = z_1^2(0) + z_2^2(0) = 1$$

Therefore $z_2(t)$ for $t > T_1$ is given by

$$z_2(t) = \frac{1}{\sqrt{1+\beta}} \exp\left(-(t - T_1)/\sqrt{\tau_1 \tau_2}\right) \tag{D.25}$$

Note that, because λ_s is purely real, the system settles monotonically, i.e., with no overshoot. This is an important advantage of the switching controller over the linear one. The system is defined to have settled when $|z_2(t)| \leq \epsilon$. Defining $|z_2(T_s)| = \epsilon$, we get

$$T_s - T_1 = \sqrt{\tau_1 \tau_2} \ln\left(\frac{1}{\epsilon\sqrt{1+\beta}}\right) \tag{D.26}$$

Combining (D.24) and (D.26) we get

$$T_s = \sqrt{\tau_1 \tau_2}\left[\tan^{-1}\left(\sqrt{\beta}\right) + \ln\left(\frac{1}{\epsilon\sqrt{1+\beta}}\right)\right] \tag{D.27}$$

Figure D-20 compares the settling time predicted by (D.27) with that obtained using Miller compensation. The figure plots T_s as a function of α for the switching controller and various values of the Miller capacitor C_f. The other parameters were $\epsilon = 0.01$ (i.e., 1% settling), $C_1 = 0.1$, $C_2 = 1$ and $g_{m,tot} = 5$. We see that the minimum settling time with the switching controller is about a factor of four lower than with any linear compensator. In addition, the settling time is almost constant with α, thus increasing robustness.

Another advantage of the switching controller is its lack of overshoot. Overshoot can be found from the peak (maximum) value of the output z_2 during a unit step response by subtracting the final value, i.e., unity. Figure D-21 shows calculated peak output values for the amplifier using Miller compensation. We see that, as predicted by (D.16), overshoot decreases as C_f is increased.

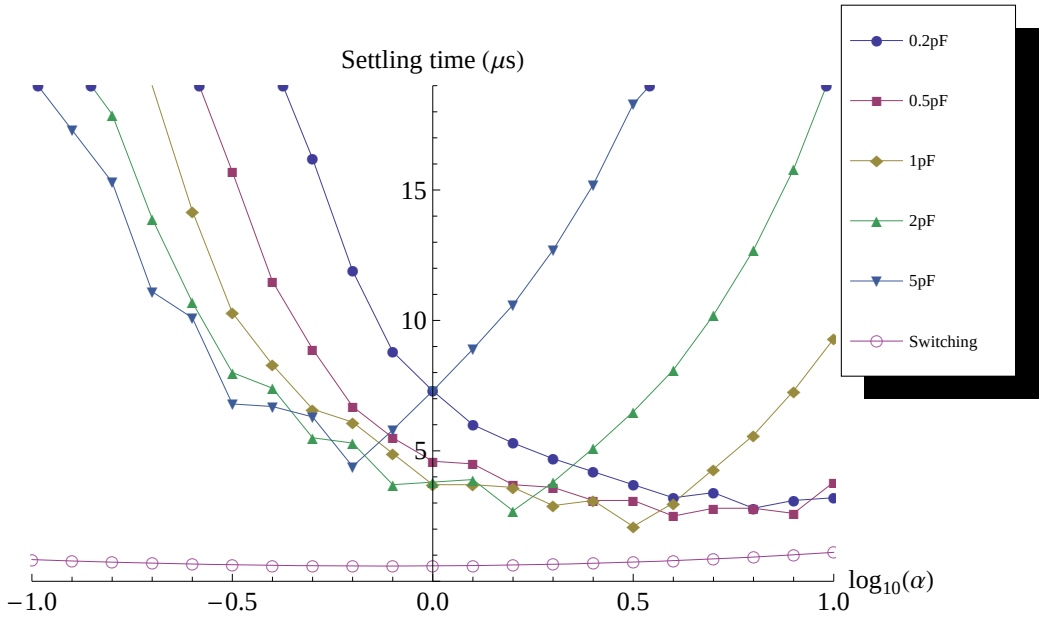


Figure D-20: Calculated settling times of the two-pole amplifier considered in this section using a) linear Miller compensation and b) the two-state switching controller.

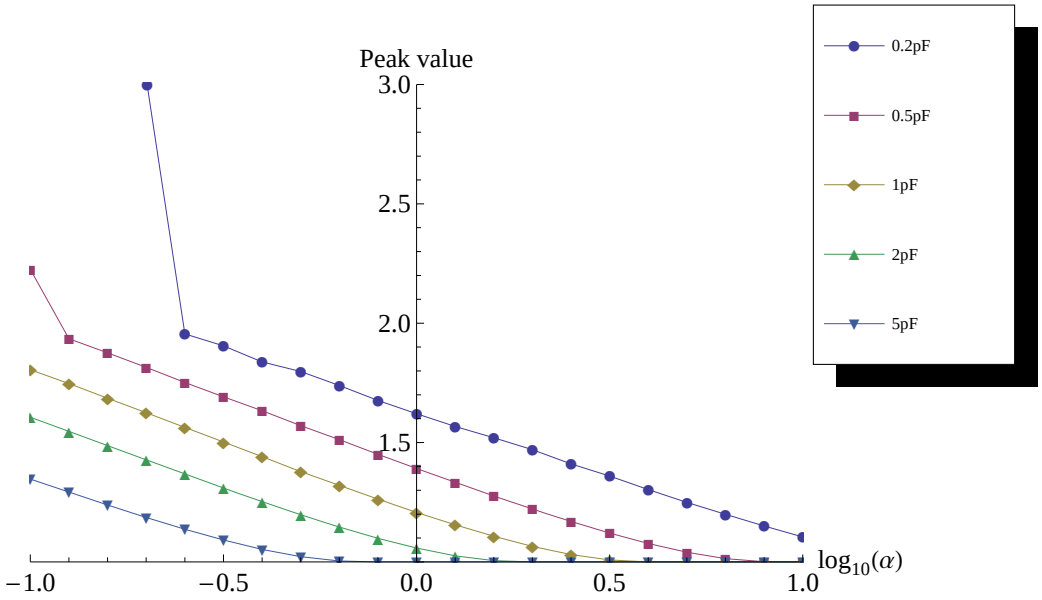


Figure D-21: Calculated peak values in the unit step response of the two-pole amplifier considered in this section using linear Miller compensation. The two-state switching controller has no overshoot, so its peak value is always unity.

D.2.3 Circuit Implementation

In this section we describe a simple circuit implementation of a two-state switching controller for a double integrator connected as a voltage follower, i.e., in unity feedback. We used the UMC 0.18μm CMOS process for our simulations. The circuit is shown in Figure D-22. The integrators in the feedback loop are single-stage folded-cascode amplifiers. The network of switches connected to the input of the first integrator switches the loop between negative and positive feedback based on $s(t)$, the output of the supervisor. The voltage V_{CM} is a constant common-mode voltage normally set to $V_{DD}/2$ to maximize signal swing.

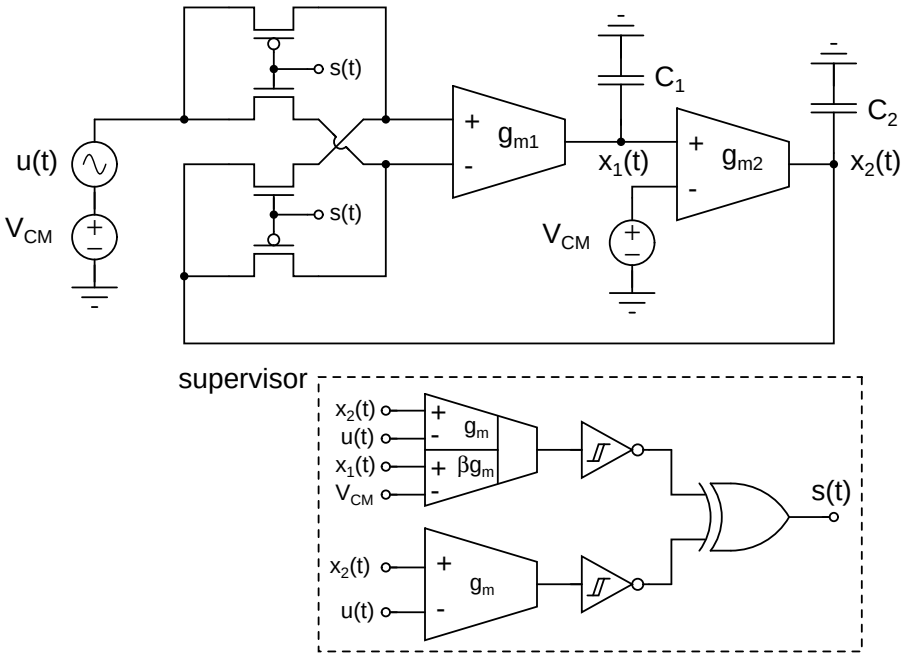


Figure D-22: Circuit implementation of a two-state switching controller for a double integrator connected as a voltage follower.

The supervisor must compute $s(t)$ based on the control law given by (D.23). There are two ways to calculate the sign of the product $z_2\left(z_2 + z_1/\sqrt{\beta}\right)$. The first way is to calculate the product using an analog multiplier and then determine its sign using a comparator. The second is to find the sign of each term in the product using a comparator and then use a one-bit multiplier, i.e., an XOR gate, to find the sign of the product. We used the latter technique since it was easier to implement. We

initially used operational amplifiers with resistive feedback to calculate the quantity $z_2 + z_1/\sqrt{\beta} = (x_2 - u) + (x_1 - V_{CM})/\sqrt{\beta}$, but then realized that a simpler way to carry out the addition was in the current domain.

Figure D-22 shows how the final version of the supervisor was implemented. Two differential amplifiers, implemented as wide-swing OTAs, calculate the signs of z_2 and $z_2 + z_1/\sqrt{\beta}$. The latter has four input transistors that share one tail current source, and acts as a differential difference amplifier. The outputs of the amplifiers are fed into the XOR gate that generates the switching control signal $s(t)$.

Figure D-23 shows a transient simulation of the circuit for $\beta = 2$ and a square wave input. Both integrators were biased at 0.5μA, resulting in $g_{m1} = g_{m2} = 7.1\mu$A, and the load capacitors were given by $C_1 = 10$pF and $C_2 = 40$pF. The output step response in Figure D-23 shows no overshoot. The controller $s(t)$ chatters in steady state, resulting in ripple. A small amount of ripple is visible on x_1 but, because of the additional low-pass filtering provided by the second integrator, not on x_2. The 1% settling time is about 25μs, which is about a factor of two higher than predicted by (D.27). However, simulations for $\beta = 1$ result in a settling time of about 11μs, in excellent agreement with the formula.

Other circuit parameters of interest were $V_{DD} = 1.8$V, $V_{CM} = 1.1$V and $g_m = 23\mu$S. The latter quantity is the transconductance of the differential amplifiers in the supervisor. As g_m is reduced to save power or τ_1 and τ_2 in the amplifier decrease, the supervisor becomes slower relative to the plant, increasing steady-state ripple. Overshoot also appears in the step response if the ripple is big enough for the system to overshoot the origin while on the stable eigenvector w_s.

The linear range of the follower is limited in this implementation to the linear range of the differential amplifiers in the supervisor. However, well-known techniques exist for extending the linear range of differential amplifiers [258], so this is not a serious limitation. A more important disadvantage of the switching controller is chattering. Chattering slows down convergence and increases settling time because the control signal is only active during part of the settling transient. The chattering frequency should be much faster than the crossover frequency of the loop to keep

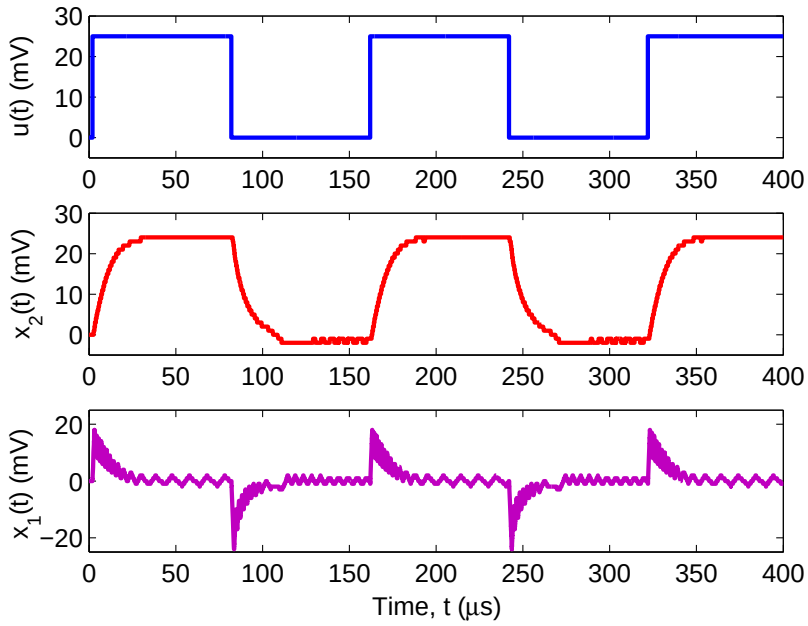


Figure D-23: A transient simulation of the double integrator circuit using switching control. Component values were given by $g_{m1} = g_{m2} = 7.1\mu$s, $C_1 = 10$pF and C_2=40pF.

steady-state ripple small. The loop therefore "sees" a low-pass-filtered version of the control signal, with an effective value equal to its duty cycle. The rate of convergence is decreased because this effective value is always less than one. Another disadvantage of high-frequency chattering is constant switching activity in the supervisor, which burns power. The subject of chattering reduction therefore warrants further study.

Appendix E

The Multiplication Table Problem

A mathematician is a machine for turning coffee into theorems.

– Alfréd Rényi

The author gratefully acknowledges extensive and helpful discussions with Benjamin Rapoport on this topic.

E.1 Introduction

This appendix deals with a mathematical problem that appears at first to be trivial, but proves to be quite hard to solve. In fact, it turns out that generalized versions of this problem remain an object of investigation for professional number theorists. I am emphatically not a number theorist, not even an amateur one. However, an elucidation of my attempts to investigate the problem may be of interest to some readers.

I begin with a simplified statement of the problem, namely the one that first came to my attention:

Problem Statement. Given positive integers i, j and n, how many unique ways are there to write $ij = n$, where $1 \leq (i, j) \leq k$ and $1 \leq n \leq k^2$? In other words, how many unique numbers exist in the multiplication table for k?

The motivation behind considering this problem is as follows. A simple way to construct a high-precision digital-to-analog converter (DAC) is by multiplying together

the outputs of two lower-precision DACs. For simplicity, consider two identical N-bit DACs whose outputs are multiplied together in an attempt to create a single DAC with $2N$-bit precision. The two lower-precision DACs are assumed to generate numbers between 1 and 2^N. We notice that the largest and smallest required output numbers are generated by this scheme, since $1 \times 1 = 1$, and $2^N \times 2^N = 2^{2N}$. However, it is easy to see that, if N is sufficiently large, *at least* half the output numbers will be skipped, since there are only 2^{2N} distinct input combinations and at least half of them are equivalent because multiplication is commutative. In other words, $ij = ji$, so the two input combinations (i, j) and (j, i) map to the same output[1]. The problem is to determine exactly *how many* of the 2^{2N} output numbers are reachable via this scheme as a function of N. The formal equivalence of this problem to the one stated in the previous paragraph follows once one makes the identification $N = \log_2(k)$.

E.2 Towards a Solution

Let us define the function $M(k)$ to be the fraction of integers between 1 and k^2 that can be reached by the multiplication table. In other words,

$$M(k) = \frac{(\text{number of unique products})}{k^2} \tag{E.1}$$

Based on our previous arguments, we must have $0 \leq M(k) \leq 1/2$ as $k \to \infty$. It might be instructive to look at the case of real numbers first. If i, j and k are real numbers instead of integers, we can use integration since $M(k)$ becomes a continuous function. It is easy to see that $M(k)$ is given by

$$M(k) = \frac{\int_0^k \int_0^k ij di dj}{\int_0^{k^2} ij d(ij)} = \frac{k^4/4}{k^4/2} = \frac{1}{2} \tag{E.2}$$

In other words, deviations of $M(k)$ from $1/2$ are fundamentally related to the

[1]When N is small more than half the input combinations can be reached because of the diagonal terms $i = j$ in the table. If i is prime such terms map one-for-one to outputs, i.e. the mapping between i and i^2 is unique when i is prime. The effect of these terms becomes negligible as N increases since their number scales at least as slowly as 2^N, compared to 2^{2N} for the whole space of input combinations.

properties of integers. We used two algorithms to calculate the exact value of $M(k)$ for various values of k, as described below.

E.2.1 Brute force algorithm

This algorithm is extremely simple (and fast). A vector V of length k^2 is initialized with zeros. If a number i is present in the output the value of the i-th element in V is changed to 1. A counter keeps track of $M(k)$, i.e. $\sum_{i=1}^{k^2} V(i)/k^2$. A MATLAB code listing is shown below:

```
% Calculate M(k) for k between 1 and kmax using a brute force method
% Begin by initializing values
M=zeros(1,kmax); numcodes=0;
V=zeros(1,kmax^2);
% Begin loop
for k=1:kmax
    for i=1:k
        if V(k*i)== 0
            % Increment output counter
            numcodes=numcodes+1;
            V(k*i)=1;
        end
    end
    % Calculate M(k)
    M(k)=numcodes/(k*k);
end
```

E.2.2 Another exact algorithm

This algorithm is slower than the brute force method but is included to show another way of approaching the problem. The main steps involved are:

1. Begin by pre-computing the unique prime factorization of every integer between 1 and k^2. This is typically the slowest step.

2. Exclude all numbers where the largest prime factor, m_f exceeds k.

3. Temporarily exclude all numbers i where $m_f \leq k$, but a decomposition into two parts, with m_f being one part, is not possible since $(i/m_f) > k$.

4. Search for a possible decomposition by testing if m_f/j and $(i/m_f) \times j$ are both integers $< k$ for some integer j. If such a decomposition exists, reinstate i into the list of valid outputs.

A MATLAB code listing of this algorithm is shown below:

```
% Calculate M(k) for k between 1 and kmax using another algorithm
% Begin by initializing values
M=zeros(1,kmax); mf=zeros(1,kmax^2); rem=mf;
% Pre-compute prime factorizations to increase speed
for i=3:kmax^2
    f=factor(i);
    % Find maximum prime factor mf and remainder rem=i/mf
    mf(i)=max(f); rem(i)=i/mf(i);
end
% Begin loop
for k=2:kmax
    numcodes=k^2;
    for i=k+1:k^2
        if mf(i)>k
            % Exclude numbers with mf > k
            numcodes=numcodes-1;
        else
            if rem(i)>k
                % Temporarily exclude this number
```

```
                numcodes=numcodes-1; j=2;
                % Try to create a feasible decomposition
                while mf(i)*j<k+1
                    if rem(i)/j==floor(rem(i)/j) && rem(i)/j<k+1
                        % A decomposition exists, re-include the number
                        numcodes=numcodes+1;
                        break;
                    end
                    j=j+1;
                end
            end
        end
    end
    % Calculate M(k)
    M(k)=numcodes/(k*k);
end
```

E.2.3 Analytical Upper Bound

In addition to the algorithms described in the previous section for calculating $M(k)$, we can also derive an approximate upper bound for $M(k)$ analytically. We shall need the prime-counting function $\pi(k)$. The prime-counting function is defined as the number of primes less than or equal to k. The simplest form of the celebrated *prime number theorem* states that $\pi(k) \sim k/\log(k)$, where the symbol $\sim$ denotes asymptotic equality as $k \to \infty$. In other words

$$\lim_{k\to\infty} \frac{\pi(k)}{k/\log(k)} = 1 \tag{E.3}$$

We can calculate an upper bound by using an exclusion method, i.e., proving that certain numbers between 1 and k^2 *cannot* be formed by multiplying together numbers between 1 and k. Prime numbers are a good place to start. Clearly primes between

k^2 and $k^2/2$ cannot be formed. Similarly, primes between $k^2/2$ and $k^2/3$, and their products with 2, cannot be formed. In general, primes between k^2/i and $k^2/(i+1)$, and all their products with $1, 2, ...i$, cannot be formed, as long as $k^2/(i+1) > k$, i.e. $i < k-1$. We therefore have

$$M(k) < 1 - \frac{1}{k^2}\sum_{i=1}^{k-1} i\left[\pi\left(\frac{k^2}{i}\right) - \pi\left(\frac{k^2}{i+1}\right)\right] \tag{E.4}$$

The right-hand side of the equation above is only an upper bound since certain unreachable numbers are not excluded by this technique. The most balanced decomposition of such numbers still results in one factor that is larger than k. For example, consider the number 952 for $k = 32$. Its most balanced decomposition is 34×28, but it will fall through our exclusion method since its prime factorization is $17 \times 7 \times 2^3$.

Interestingly enough, one can use the prime number theorem to express the right-hand side of (E.4) in closed form when $k \to \infty$. Replacing $\pi(k)$ with its asymptotic form $k/\log(k)$ gives us

$$\begin{aligned}
\lim_{k\to\infty} M(k) &< 1 - \frac{1}{k^2}\sum_{i=1}^{k-1} i\left[\frac{k^2/i}{\log\left(k^2/i\right)} - \frac{k^2/(i+1)}{\log\left(k^2/(i+1)\right)}\right] \\
&< 1 - \sum_{i=1}^{k-1} \frac{1}{(i+1)\log\left(k^2/(i+1)\right)} \\
&< 1 - \int_2^{\infty} \frac{dx}{x\log\left(k^2/x\right)} \\
&< 1 - \log\left[1 + \frac{\log(k/2)}{\log(k)}\right]
\end{aligned} \tag{E.5}$$

Finally, I have been unable to find a reasonably tight lower bound on $M(k)$. A *very* loose bound may be derived by recognizing that all numbers formed by multiplying together pairs of primes between 1 and k are unique numbers between 1 and k^2, and therefore present in the output. There are $\pi(k)[\pi(k)+1]/2$ such pairs, since each prime is also allowed to multiply itself. We should also add the k numbers between 1 and k to the output list, giving us

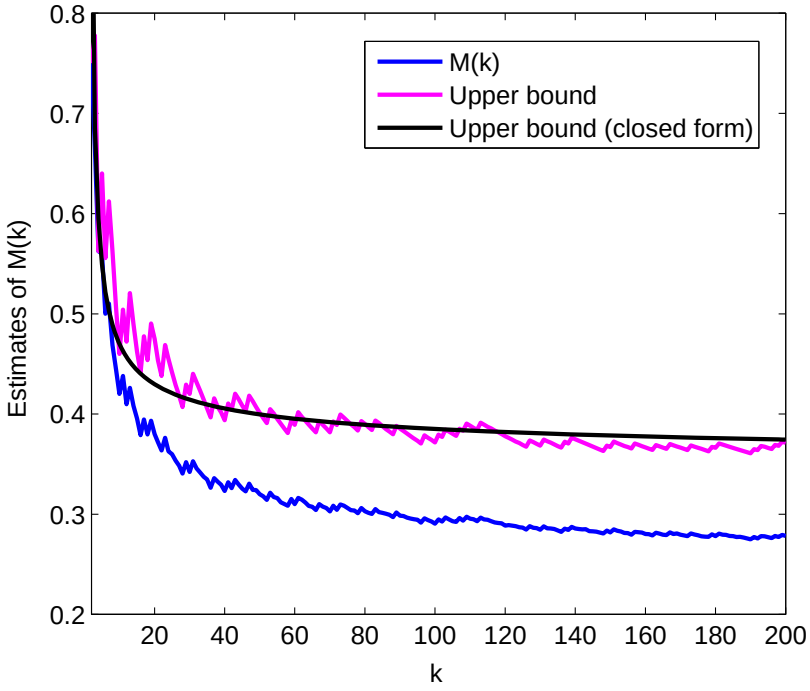


Figure E-1: Upper bounds of the multiplication table function $M(k)$, compared with its true value calculated using either algorithm described in the text.

$$M(k) > \frac{k + \pi(k)[1 + \pi(k)]/2}{k^2} \tag{E.6}$$

Combining the prime number theorem with the inequality above reveals that $M(k)$ must decay slower than $1/(\log(k))^2$ as $k \to \infty$. Figure E-1 compares the upper bounds for $M(k)$ (equations (E.4) and (E.5)) with the function itself. Figure E-2 shows the bounding error of these upper bounds, i.e. the gaps between the upper bounds and the function.

In addition to computing $M(k)$, which measures the fraction of numbers reached by the multiplication, we may also be interested in the distribution of these numbers in the space of possible values, i.e. 1 to k^2. In order to visualize this distribution, we bin the outputs produced by the multiplication table for k into k bins, each k numbers wide. Thus the (k, m)-th bin contains numbers between $1 + (m - 1)k$ and mk, where $1 \leq m \leq k$. The *output density* $d(k, m)$ is defined as

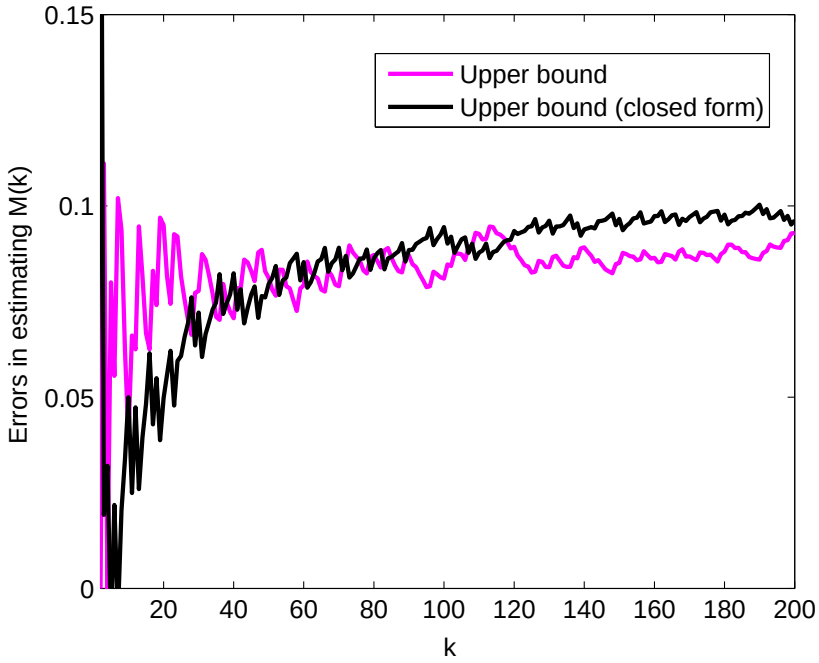


Figure E-2: Bounding errors of the two upper bounds for $M(k)$ described in the text.

$$d(k,m) = \frac{\text{number of numbers in the } (k,m)\text{-th bin}}{k} \tag{E.7}$$

It is easy to see that $0 \leq d(k,m) \leq 1$. Figure E-3 shows the computed output density function for $2 \leq k \leq 100$. We see that the density consistently decreases for large bin numbers, i.e., outputs close to k^2.

Our target application was $N = 5$, i.e., using two 5-bit DACs to create a 10-bit DAC. We used this strategy to set reaction rate constants in the chemical kinetics chip discussed earlier. Figure E-1 shows that for $k = 2^5 = 32$, we can reach 354 of the $2^{10} = 1024$ possible output levels, i.e. $M(32) = 0.346$, corresponding to about 8.5 bits of (non-uniformly distributed) precision.

E.3 Status as a Mathematical Problem

An asymptotic formula for $M(k)$ as $k \to \infty$ is unknown and remains an unsolved mathematical problem. The On-Line Encyclopedia of Integer Sequences lists $k^2 M(k)$

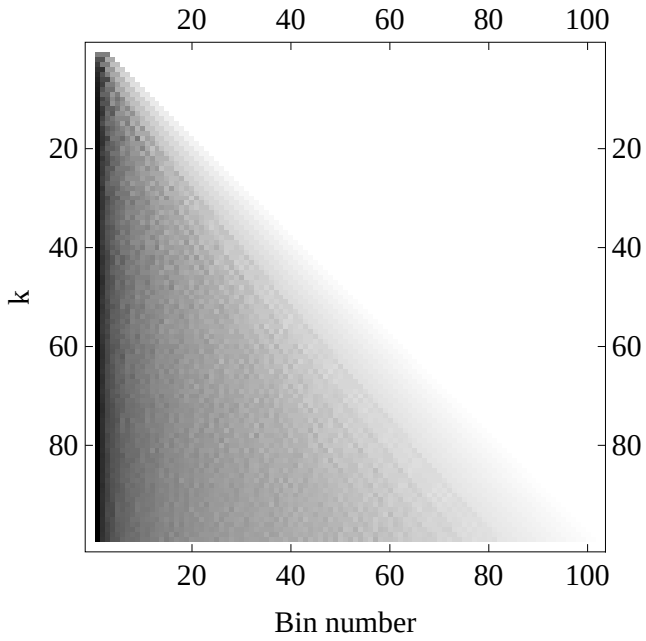


Figure E-3: Density of the numbers produced by multiplication tables of various sizes. The gray-scale value of a pixel is equal to the value of the density function, and ranges from 0 (white) to 1 (black).

as sequence A027424. Based on Figure E-1, one may suspect that $M(k)$ converges to some value around 0.25 as k increases. However, further investigation shows that it goes below 0.25 somewhere around $k = 850$. In fact, in 1960 the great Paul Erdős proved the surprising result that [6]:

$$\lim_{k\to\infty} M(k) = 0 \tag{E.8}$$

This result is known as the "Erdős multiplication table theorem". To be more precise, Erdős showed that

$$\lim_{k\to\infty} M(k) = \frac{1}{(\log(k))^{\delta+o(1)}} \tag{E.9}$$

where the constant δ is defined as

$$\delta = 1 - \frac{1 + \log(\log(2))}{\log(2)} = 0.086071... \tag{E.10}$$

An intuitive explanation for Erdős' result now follows. It is well-known that a pos-

itive integer k has about $\omega(k) = \log(\log(k))$ distinct prime factors (this theorem was initially proven in 1917 by Hardy and Ramanujan). It is also known that the probability that two randomly picked integers are relatively prime is $[\zeta(2)]^{-1} = 6/\pi^2 = 0.60792...$, where $\zeta(z)$ is the Reimann zeta function[2]. Therefore most numbers in the table will be the product of relatively prime numbers, and have about $2\log(\log(k))$ distinct prime factors. This is an abnormally large number of prime factors for a number k^2, which should normally have about $\log(\log(k^2)) = \log(\log(k))+\log(2) \sim \log(\log(k))$ prime factors as $k \to \infty$. There are very few such abnormal numbers: Erdős and Kac showed in 1939 that $\omega(k)$, the number of prime factors in a number k, follows a Gaussian distribution, with the mean being $\log(\log(k))$ [65]. Therefore there are very few unique products in the multiplication table.

A recent improvement to Erdős' formula (E.9) has been made by Ford [73], who considered a generalized version of the problem and showed as a corollary that

$$M(k) \asymp \frac{1}{(\log(k))^{\delta} (\log(\log(k)))^{3/2}} \tag{E.11}$$

where δ is defined as before and the symbol $\asymp$ means "asymptotically bound in both directions". In other words, $M(k)$ as $k \to \infty$ must lie between two positive constant multiples of the quantity on the right-hand side of (E.11)[3].

[2]For more information about relatively prime numbers, see, for example, A059956 in the On-Line Encyclopedia of Integer Sequences.

[3]The $\asymp$ symbol in this context is equivalent to the Θ function used in theoretical computer science to simultaneously place upper and lower bounds on computational complexity.

Appendix F

Radiation Resistance of Small Antennas

In this appendix, I present simplified derivations of the radiation resistance of electrically small dipole and loop antennas. The results are not new, and can be found in standard antenna design textbooks, such as [11], but the presentation may interest some readers.

F.1 Maxwell's Equations

Consider a homogenous, lossless medium in sinusoidal steady-state. In this case the permeability and permittivity are scalars, represented by μ and ϵ, respectively, and Maxwell's first two equations can be written in differential form as:

$$\begin{aligned} \nabla \times \mathbf{E} &= -j\omega\mu\mathbf{H} \\ \nabla \times \mathbf{H} &= \mathbf{J} + j\omega\epsilon\mathbf{E} \end{aligned} \tag{F.1}$$

where ω is the frequency, $\mathbf{E}$ is the electric field, $\mathbf{H}$ is the magnetic field and $\mathbf{J}$ is the current density vector. Now define the magnetic vector potential $\mathbf{A}$ to be the vector whose curl produces the magnetic flux density $\mu\mathbf{H}$, i.e.

$$\nabla \times \mathbf{A} = \mu \mathbf{H} \tag{F.2}$$

Substituting (F.2) into the first equation in (F.1) gives us

$$\nabla \times [\mathbf{E} + j\omega \mathbf{A}] = 0 \tag{F.3}$$

Since the curl of the vector on the left hand side of the equation above is zero, it is the gradient of some scalar potential. Let us denote this potential by ϕ (it is commonly known as the electrostatic potential):

$$\mathbf{E} + j\omega \mathbf{A} = \nabla \phi \tag{F.4}$$

Now take the curl of both sides of (F.2) and use the common vector identity

$$\nabla \times \nabla \times \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} \tag{F.5}$$

to give

$$\nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} = \mu \left(\nabla \times \mathbf{H}\right) \tag{F.6}$$

By substituting the right-hand side of the equation above using the second Maxwell equation in (F.1) and re-arranging some terms, we get

$$\nabla^2 \mathbf{A} = \nabla \left(\nabla \cdot \mathbf{A}\right) - \mu \mathbf{J} - j\omega \mu \epsilon \mathbf{E} \tag{F.7}$$

Substituting for $\mathbf{E}$ using (F.4) and using the facts that $\mu\epsilon = 1/c^2$ and $\omega/c = k$, the wave vector, gives us

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = -\mu \mathbf{J} + \nabla \cdot \left[\nabla \cdot \mathbf{A} + \frac{j\omega}{c^2} \phi\right] \tag{F.8}$$

Since only the curl of $\mathbf{A}$ is used to define $\mathbf{H}$, we are free to choose its divergence to be whatever we want. This choice is known as a gauge. The usual choice is a Lorenz

gauge, which maintains Lorentz invariance[1], and is defined as

$$\nabla \cdot \mathbf{A} + \frac{j\omega}{c^2}\phi = 0 \tag{F.9}$$

By substituting the Lorenz gauge condition defined above into (F.8) we get a wave equation for $\mathbf{A}$:

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = -\mu \mathbf{J} \tag{F.10}$$

Solutions to (F.10) in three dimensions at points that are distant from the source $\mathbf{J}$ are radial waves proportional to $\exp(-jkr)/r$ and $\exp(+jkr)/r$, where r is the distance from the source. Since time increases as $\exp(j\omega t)$, the former solution corresponds to waves that propagate outward. In other words, the phase of the wave, which is $j(\omega t - kr)$, remains constant if r increases with time as $r = ct$, where $c = \omega/k$. Similarly, the latter solution corresponds to inward-propagating waves where constant phase corresponds to $r = -ct$. We choose forward-propagating solutions and assume that every point within the source volume launches vector potential waves. Integrating over the source volume V' gives us the net vector potential:

$$\mathbf{A} = \frac{\mu}{4\pi} \iiint_{V'} \mathbf{J} \frac{e^{-jkr}}{r} dv' \tag{F.11}$$

If the source $\mathbf{J}$ is located at primed co-ordinates $\mathbf{r}' = (x', y', z')$ and the observer at unprimed co-ordinates $\mathbf{r} = (x, y, z)$ the equation above can be rewritten as follows:

$$\mathbf{A} = \frac{\mu}{4\pi} \iiint_{V'} \mathbf{J}(\mathbf{r}') \frac{e^{-j\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')}}{|\mathbf{r}-\mathbf{r}'|} dv' \tag{F.12}$$

If the source J is a produced by an infinitely thin wire, the volume integral above is replaced by a line integral along C', the path of the wire:

$$\mathbf{A} = \frac{\mu}{4\pi} \int_{C'} \mathbf{I}(\mathbf{r}') \frac{e^{-j\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')}}{|\mathbf{r}-\mathbf{r}'|} dl' \tag{F.13}$$

[1]The two spellings of Lorenz are not a typo. They refer to two different scientists: Ludvig Lorenz (1829-1891) and Hendrik Lorentz (1853-1928), respectively

where $\mathbf{I}$ is the current in the wire. Equation (F.13) will form the basis for our subsequent calculations of radiation resistance. In the far-field case, we can ignore any terms in $\mathbf{A}$ that fall off faster than $1/r$ with distance, so $\mathbf{A}$ may be written in spherical co-ordinates as

$$\mathbf{A} = \left[\hat{\mathbf{a}}_r A_r(\theta,\phi) + \hat{\mathbf{a}}_\theta A_\theta(\theta,\phi) + \hat{\mathbf{a}}_\phi A_\phi(\theta,\phi)\right] \frac{e^{-jkr}}{r} \tag{F.14}$$

where $\hat{\mathbf{a}}_r$, $\hat{\mathbf{a}}_\theta$ and $\hat{\mathbf{a}}_\phi$ are the unit vectors along the r, θ and ϕ directions, respectively. In order to carry out our radiation resistance calculation we will need to find $\mathbf{E}$ and $\mathbf{H}$ in terms of $\mathbf{A}$. We have the following relations based on our definition of $\mathbf{A}$ and the Lorenz gauge:

$$\begin{aligned} \mathbf{E} &= -j\omega\mathbf{A} - \frac{jc^2}{\omega}\nabla(\nabla\cdot\mathbf{A}) \\ \mathbf{H} &= \frac{1}{\mu}(\nabla\times\mathbf{A}) \end{aligned} \tag{F.15}$$

Using these relations and again throwing away any terms that decay faster than $1/r$ gives us

$$\begin{aligned} E_r &= 0 \\ (E_\theta, E_\phi) &= -j\omega(A_\theta, A_\phi) \end{aligned} \tag{F.16}$$

and

$$\begin{aligned} H_r &= 0 \\ (H_\theta, H_\phi) &= -\frac{j\omega}{\eta}(-A_\phi, A_\theta) \end{aligned} \tag{F.17}$$

where $\eta = \sqrt{\mu/\epsilon} = 120\pi\Omega$ is the characteristic impedance of free space. Intuitively, $\mathbf{H} \propto 1/\eta$ because the term that retains a $1/r$ dependence on taking the cross product

involves differentiating the exponential $\exp(-jkr)$ w.r.t. r. This gives us a coefficient of $-jk/\mu = -j\omega/(c\mu) = -j\omega/\eta$ in front. For θ and ϕ (not r, since both E_r and H_r are zero) equations (F.16) and (F.17) can be compactly written as

$$\begin{aligned} \mathbf{E} &= -j\omega\mathbf{A} \\ \mathbf{H} &= -\frac{j\omega}{\eta}\left(\hat{\mathbf{a}}_r \times \mathbf{A}\right) \end{aligned} \tag{F.18}$$

The time-averaged Poynting vector now gives us the average power density present in the medium:

$$\mathbf{W}_{av} = \frac{1}{2}\mathbf{Re}\left[\mathbf{E} \times \mathbf{H}^*\right] \tag{F.19}$$

where $\mathbf{H}^*$ is the complex conjugate of $\mathbf{H}$. The factor of $1/2$ is the usual ratio of rms to peak power that arises when time-averaging a sinusoid. The total average power P_{rad} radiated by an antenna can be found by integrating $\mathbf{W}_{av}$ over any sphere S which has the antenna at its center:

$$P_{rad} = \oint_S \mathbf{W}_{av} \cdot d\mathbf{s} \tag{F.20}$$

where $d\mathbf{s}$ is a differential area element vector that points in the radial direction, i.e., along $\hat{\mathbf{a}}_r$, since S is a sphere.

We will also need matrices to convert between co-ordinate systems. For example, between cartesian and cylindrical:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_\rho \\ a_\phi \\ a_z \end{bmatrix} \tag{F.21}$$

and also between cartesian and spherical:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \sin(\theta)\cos(\phi) & \cos(\theta)\cos(\phi) & -\sin(\phi) \\ \sin(\theta)\sin(\phi) & \cos(\theta)\sin(\phi) & \cos(\phi) \\ \cos(\theta) & -\sin(\theta) & 0 \end{bmatrix} \begin{bmatrix} a_r \\ a_\theta \\ a_\phi \end{bmatrix} \tag{F.22}$$

F.2 Small Dipole

A dipole antenna may be considered "small" if it has a current distribution that is nearly uniform. Let the dipole be much thinner than its length, oriented along the z axis from $-l/2$ to $+l/2$, and fed from the origin. Therefore we can use (F.13) to calculate the vector potential $\mathbf{A}$, once we make the following identifications:

$$\begin{aligned} \mathbf{I}(\mathbf{r}') &= \hat{\mathbf{a}}_z I_0 \\ |\mathbf{r} - \mathbf{r}'| &\approx r \\ dl' &= dz' \end{aligned} \tag{F.23}$$

where I_0 is the amplitude of the current in the antenna, and we have assumed far-field operation, i.e. $|\mathbf{r}| \gg l$. Substituting (F.23) in (F.13), we get

$$\mathbf{A} = \hat{\mathbf{a}}_z \frac{\mu I_0}{4\pi} \left(\frac{e^{-jkr}}{r} \right) \int_{-l/2}^{l/2} dz' = \hat{\mathbf{a}}_z \frac{\mu I_0 l}{4\pi} \left(\frac{e^{-jkr}}{r} \right) \tag{F.24}$$

Using (F.22), we can convert $\mathbf{A}$ to spherical co-ordinates, the result being

$$\mathbf{A}(\mathbf{r}) = (\cos(\theta)\hat{\mathbf{a}}_r - \sin(\theta)\hat{\mathbf{a}}_\theta) \frac{\mu I_0 l}{4\pi} \left(\frac{e^{-jkr}}{r} \right) \tag{F.25}$$

Using (F.18) and (F.19), the time-averaged Poynting vector is given by

$$\begin{aligned}
\mathbf{W}_{av} &= \frac{1}{2}\mathbf{Re}\left[\mathbf{E}\times\mathbf{H}^*\right] \\
&= \frac{\omega^2}{2\eta}\mathbf{Re}\left[\mathbf{A}\times(\hat{\mathbf{a}}_r\times\mathbf{A})\right] \\
&= \left(\sin^2(\theta)\hat{\mathbf{a}}_r+\cos(\theta)\sin(\theta)\hat{\mathbf{a}}_\theta\right)\left(\frac{\omega\mu I_0 l}{4\pi r}\right)^2\frac{1}{2\eta}
\end{aligned} \tag{F.26}$$

We can now find the total power radiated by the antenna by integrating $\mathbf{W}_{av}$ over a sphere of radius r, i.e., using (F.20), once we remember that the differential area vector in spherical co-ordinates is given by $d\mathbf{s}=\hat{\mathbf{a}}_r r^2\sin(\theta)d\theta d\phi$:

$$\begin{aligned}
P_{rad} &= \frac{1}{2\eta}\left(\frac{\omega\mu I_0 l}{4\pi r}\right)^2\int_0^{2\pi}\int_0^{\pi}\left(\sin^2(\theta)\hat{\mathbf{a}}_r+\cos(\theta)\sin(\theta)\hat{\mathbf{a}}_\theta\right)\cdot\hat{\mathbf{a}}_r r^2\sin(\theta)d\theta d\phi \\
&= \frac{1}{2\eta}\left(\frac{\omega\mu I_0 l}{4\pi}\right)^2\int_0^{2\pi}\int_0^{\pi}\sin^3(\theta)d\theta d\phi \\
&= \frac{4\pi}{3\eta}\left(\frac{c\mu I_0 l}{2\lambda}\right)^2 \\
&= \frac{\pi}{3\eta}(c\mu)^2\left(\frac{l}{\lambda}\right)^2 I_0^2 \\
&= \frac{\pi\eta}{3}\left(\frac{l}{\lambda}\right)^2 I_0^2
\end{aligned} \tag{F.27}$$

where we have used the facts that $\omega=ck=2\pi c/\lambda$ and $c\mu=\eta$. Finally, the radiation resistance R_{rad} is defined as

$$P_{rad}=\frac{1}{2}R_{rad}I_0^2 \tag{F.28}$$

From (F.27) and (F.28), we get

$$R_{rad}=\frac{2\pi\eta}{3}\left(\frac{l}{\lambda}\right)^2=80\pi^2\left(\frac{l}{\lambda}\right)^2 \tag{F.29}$$

We see that R_{rad} increases as the square of the electrical length l/λ of the dipole. Intuitively, this is because of the uniform current distribution, which makes the mag-

nitude of $\mathbf{A}$ proportional to l, and thus the magnitude of the Poynting vector $\mathbf{W}_{av}$ proportional to l^2.

F.3 Small Circular Loop

In this section we consider a small circular loop of very thin wire and radius a that is located in the $x-y$ plane with its center at the origin of the co-ordinate system. Like the dipole, a loop is considered "small" if it has a nearly uniform current distribution. We will use primed co-ordinates for the source, i.e. the loop, and non-primed co-ordinates for the observer, assumed to be in the far field, i.e., $r \gg a$.

First we have to express the current vector in spherical co-ordinates. The most natural representation is in cylindrical co-ordinates, where we may write $\mathbf{I} = \hat{\mathbf{a}}_\phi I_0$, where I_0 is the amplitude of the current. We can convert this representation to cartesian co-ordinates by using (F.21), the result being

$$\mathbf{I} = \left[-\sin\left(\phi'\right)\hat{\mathbf{a}}_x + \cos\left(\phi'\right)\hat{\mathbf{a}}_y\right] I_0 \tag{F.30}$$

where ϕ' goes from 0 to 2π around the loop. In order to convert this representation into spherical co-ordinates, we can use (F.22). Because of the symmetry of the problem, however, the fields and potentials should not depend on ϕ, the observer's azimuthal angle. We can therefore simplify our algebra by fixing ϕ at any value; let us choose $\phi = 0$. Using (F.22) we then get

$$\mathbf{I} = \left[-\sin(\theta)\sin\left(\phi'\right)\hat{\mathbf{a}}_r + \cos(\theta)\sin\left(\phi'\right)\hat{\mathbf{a}}_\theta + \cos\left(\phi'\right)\hat{\mathbf{a}}_\phi\right] I_0 \tag{F.31}$$

Next we need to calculate the distance between the source and the observer, i.e $|\mathbf{r}-\mathbf{r}'| = \sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2}$. By using the following relationships

$$\begin{array}{lll} x' = a\cos\left(\phi'\right) & x = r\sin(\theta)\cos(\phi) & x^2+y^2+z^2 = r^2 \\ y' = a\sin\left(\phi'\right) & y = r\sin(\theta)\sin(\phi) & (x')^2+(y')^2+(z')^2 = a^2 \\ z = 0 & z = r\cos(\theta) & \end{array} \tag{F.32}$$

and setting $\phi = 0$, we get

$$|\mathbf{r} - \mathbf{r}'| = \sqrt{r^2 + a^2 - 2ar\sin(\theta)\cos(\phi')} \tag{F.33}$$

By substituting (F.31) and (F.33) into (F.13), and remembering that $dl' = ad\phi'$, we get

$$\mathbf{A} = \frac{\mu a}{4\pi}\int_0^{2\pi} \mathbf{I}\frac{e^{-jk\sqrt{r^2+a^2-2ar\sin(\theta)\cos(\phi')}}}{\sqrt{r^2 + a^2 - 2ar\sin(\theta)\cos(\phi')}}d\phi' \tag{F.34}$$

We now perform a Taylor expansion of the function that multiplies $\mathbf{I}$ in this integral. If we assume $r \gg a$, all terms that are second-order or higher in a/r can be thrown away to yield

$$\frac{e^{-jk\sqrt{r^2+a^2-2ar\sin(\theta)\cos(\phi')}}}{\sqrt{r^2 + a^2 - 2ar\sin(\theta)\cos(\phi')}} \approx e^{-jkr}\left[\frac{1}{r} + a\sin(\theta)\cos(\phi')\left(\frac{jk}{r} + \frac{1}{r^2}\right)\right] \tag{F.35}$$

Substituting (F.31) and (F.35) into (F.34), we get

$$\begin{aligned}\mathbf{A} &= \frac{\mu a I_0}{4\pi}\int_0^{2\pi} e^{-jkr}\left[\frac{1}{r} + a\sin(\theta)\cos(\phi')\left(\frac{jk}{r} + \frac{1}{r^2}\right)\right] \times \\ &\quad [-\sin(\theta)\sin(\phi')\,\hat{\mathbf{a}}_r + \cos(\theta)\sin(\phi')\,\hat{\mathbf{a}}_\theta + \cos(\phi')\,\hat{\mathbf{a}}_\phi]\,d\phi'\end{aligned} \tag{F.36}$$

The r and θ components in this integral involve integrals of $\sin(\phi')$ and $\sin(2\phi')$ from 0 to 2π, and thus integrate to zero. Only the ϕ component remains; the result is

$$\begin{aligned}\mathbf{A} &= \frac{\mu a I_0}{4\pi} \times \pi\sin(\theta)e^{-jkr}a\left(\frac{jk}{r} + \frac{1}{r^2}\right)\hat{\mathbf{a}}_\phi \\ &\approx \hat{\mathbf{a}}_\phi\left(\frac{j\mu k a^2 I_0}{4}\right)\sin(\theta)\frac{e^{-jkr}}{r}\end{aligned} \tag{F.37}$$

We can now use (F.18) and (F.19) to find the time-averaged Poynting vector $\mathbf{W}_{av}$:

$$\begin{aligned}
\mathbf{W}_{av} &= \frac{1}{2}\mathbf{Re}\left[\mathbf{E}\times\mathbf{H}^{*}\right] \\
&= \frac{\omega^2}{2\eta}\mathbf{Re}\left[\mathbf{A}\times(\hat{\mathbf{a}}_r\times\mathbf{A})\right] \\
&= \hat{\mathbf{a}}_r\frac{\omega^2}{2\eta}\left(\frac{\mu k a^2 I_0}{4r}\right)^2\sin^2(\theta) \\
&= \hat{\mathbf{a}}_r\frac{\eta}{2}\left(\frac{k^2 a^2 I_0}{4r}\right)^2\sin^2(\theta)
\end{aligned} \tag{F.38}$$

where we have used the facts that $\omega = ck$ and $c\mu = \eta$. We can now find the total power radiated by the antenna by integrating $\mathbf{W}_{av}$ over a sphere of radius r, i.e., using (F.20), once we remember that the differential area vector in spherical co-ordinates is given by $d\mathbf{s} = \hat{\mathbf{a}}_r r^2\sin(\theta)d\theta d\phi$:

$$\begin{aligned}
P_{rad} &= \frac{\eta}{2}\left(\frac{k^2 a^2 I_0}{4r}\right)^2\int_0^{2\pi}\int_0^{\pi}\sin^2(\theta)\times r^2\sin(\theta)d\theta d\phi \\
&= \frac{\eta}{2}\left(\frac{k^2 a^2 I_0}{4}\right)^2\int_0^{2\pi}\int_0^{\pi}\sin^3(\theta)d\theta d\phi \\
&= \frac{8\pi}{3}\frac{\eta(ka)^4}{32}I_0^2 \\
&= \frac{\pi\eta}{12}(ka)^4 I_0^2
\end{aligned} \tag{F.39}$$

From (F.28) and (F.39), we get

$$R_{rad} = \frac{\pi\eta}{6}(ka)^4 = 20\pi^2\left(\frac{2\pi a}{\lambda}\right)^4 \tag{F.40}$$

We see that, unlike the small dipole, in which R_{rad} is proportional to the square of the electrical length, the radiation resistance of the small loop is proportional to the *fourth power* of the electrical size of the loop, i.e. $(ka)^4$. This statement explains why the radiation resistance of small loops is much smaller than dipoles of comparable electrical size. Intuitively, the magnitude of $\mathbf{A}$ in a loop is proportional to a^2, not l as in the case of the dipole. One factor of a comes from the conversion from angle $d\phi'$

to differential length dl', while the other arises from a first-order Taylor expansion of the complex function on the left hand side of (F.35).

Loop antennas have an additional degree of freedom not possessed by dipoles: we can use many turns of wire, each carrying the same current I_0, to create the loop. Suppose we have N such turns, and that the turns are very close to each other, such that the bundle of wires still has a diameter that is much smaller than the loop diameter a. The effective source current then increases from I_0 to NI_0, so we get $\mathbf{A} \to N\mathbf{A}$ and $P_{rad} \to N^2 P_{rad}$. Since the current flowing in the external terminals is still I_0, we still have $R_{rad} = 2P_{rad}/I_0^2$. Therefore R_{rad} increases by a factor of N^2, i.e.

$$\begin{aligned} R_{rad} &= 20\pi^2 \left(\frac{2\pi a}{\lambda}\right)^4 N^2 \\ &= 320\pi^4 \left(\frac{NS}{\lambda^2}\right)^2 \\ &= 3.1170 \times 10^4 \left(\frac{NS}{\lambda^2}\right)^2 \end{aligned} \tag{F.41}$$

where $S = \pi a^2$ is the surface area of the loop. The formula above is identical to (A.36), which is reassuring. We see that the radiation resistance of very small loops can be increased to respectable values by using many turns. A small loop is essentially an inductor with some loss due to radiation. Its inductance and radiation resistance both increase as N^2, and for the same reason: transformer action between the N turns increases the impedance by a factor of N^2.

It can also be shown that the electric and magnetic fields produced by small circular and square loops are identical in the far-field region, even though their near fields are different[2]. Therefore small circular and square coils have the same radiation resistance, given by (F.41).

[2] As a rule of thumb, loop antennas can usually be considered "electrically small" if their circumferences are less than $\lambda/10$.

Bibliography

[1] A. A. Abidi. The path to the software-defined radio receiver. *IEEE Journal of Solid-State Circuits*, 42(5):954–966, May 2007.

[2] A. Adamatzky, B. de L. Costello, and T. Shirakawa. Universal computation with limited Resources: Belousov-Zhabotinsky and Physarum computers. *International Journal of Bifurcation and Chaos*, 18, 2008. In press.

[3] E. Afshari, H. S. Bhat, A. Hajimiri, and J. E. Marsden. Extremely wideband signal shaping using one-and two-dimensional nonuniform nonlinear transmission lines. *J. Appl. Phys.*, 99(5):054901, March 2006.

[4] U. Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC, Boca Raton, FL, First edition, 2006.

[5] Adam P. Arkin. *Signal Processing by Biochemical Reaction Networks*, volume 1, Chapter 5, pages 112–144. Cambridge University Press, Cambridge, U.K., First edition, 2000.

[6] L. Babai, C. Pomerance, and P. Vértesi. The mathematics of Paul Erdős. *Notices of the American Mathematical Society*, 45(1):19–32, January 1998.

[7] R. Bagheri, A. Mirzaei, S. Chehrazi, M. E. Heidari, M. Lee, M. Mikhemar, W. Tang, and A. A. Abidi. An 800-MHz-6-GHz software-defined wireless receiver in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 41(12):2860–2276, December 2006.

[8] M. W. Baker and R. Sarpeshkar. A low-power high-PSRR current mode microphone preamplifier. *IEEE Journal of Solid-State Circuits*, 38(10):1671–1678, October 2003.

[9] M. W. Baker and R. Sarpeshkar. Feedback analysis and design of rf power links for low-power bionic systems. *IEEE Transactions on Biomedical Circuits and Systems*, 1(1):26–38, March 2007.

[10] C. A. Balanis. *Antenna Theory: Analysis and Design*, pages 551–566. John Wiley and Sons, New York, NY, Second edition, 1997.

[11] C. A. Balanis. *Antenna Theory: Analysis and Design*, pages 210–214. John Wiley and Sons, New York, NY, Second edition, 1997.

[12] F. Baras and M. Malek Mansour. Reaction-diffusion master equation: A comparison with microscopic simulations. *Phys. Rev. E*, 54(6):6139–6148, December 1996.

[13] D. A. Bell. Distribution function of semiconductor noise. *Proceedings of the Physical Society, Section B*, 68(9):690–691, September 1955.

[14] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, November 1973.

[15] C. H. Bennett. The thermodynamics of computation - A review. *International Journal of Theoretical Physics*, 21(12):905–940, December 1982.

[16] H C Berg and E M Purcell. Physics of chemoreception. *Biophys. J.*, 20(2):193–219, November 1977.

[17] William Bialek and Sima Setayeshgar. Physical limits to biochemical signaling. *Proceedings of the National Academy of Sciences of the United States of America*, 102(29):10040–10045, July 2005.

[18] X. Bian, G. D. Abowd, and J. M. Rehg. Using sound source localization in a home environment. In *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 19–36. Springer, may 2005.

[19] G. Bluman. Connections between symmetries and conservation laws. *Symmetry, Integrability and Geometry: Methods and Applications*, 1:011, 2005.

[20] H. W. Bode. *Network Analysis and Feedback Amplifier Design*. D. Van Nostrand Company, Princeton, NJ, First edition, 1945.

[21] J.-C. Bor and C.-Y. Wu. Analog electronic cochlea design using multiplexing switched-capacitor circuits. *IEEE Transactions on Neural Networks*, 7(1):155–166, January 1996.

[22] J. C. Bramwell and A. V. Hill. The velocity of the pulse wave in man. *Proc. Royal Society of London. Series B*, 93(652):298–306, April 1922.

[23] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, February 1995.

[24] Dmitri Bratsun, Dmitri Volfson, Lev S. Tsimring, and Jeff Hasty. Delay-induced stochastic oscillations in gene regulation. *Proceedings of the National Academy of Sciences of the United States of America*, 102(41):14593–14598, October 2005.

[25] R. W. Brockett. *Smooth Dynamical Systems which Realize Arithmetical and Logical Operations*, volume 135 of *Lecture Notes in Control and Information Sciences*, Chapter 2, pages 19–30. Springer-Verlag, Berlin, Germany, First edition, 1989.

[26] J. J. Brophy. Statistics of $1/f$ noise. *Physical Review*, 166(3):331–339, February 1968.

[27] W. E. Brownell, C. R. Bader, D. Bertrand, and Y. de Ribaupierre. Evoked mechanical responses of isolated cochlear outer hair cells. *Science*, 259:194–196, January 1985.

[28] V. Bush. The differential analyzer. A new machine for solving differential equations. *Journal of the Franklin Institute*, 212(4):447–488, October 1931.

[29] H. B. Callen and T. A. Welton. Irreversibility and generalized noise. *Physical Review*, 83(1):34–40, July 1951.

[30] M. A. Caloyannides. Microcycle spectral estimates of $1/f$ nnoise in semiconductors. *Journal of Applied Physics*, 45(1):307–316, January 1974.

[31] Yang Cao, Dan Gillespie, and Linda Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *Journal of Computational Physics*, 206(2):395–411, July 2005.

[32] Yang Cao, Daniel T. Gillespie, and Linda R. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, 122(1):014116–18, January 2005.

[33] Yang Cao, Daniel T. Gillespie, and Linda R. Petzold. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *J. Chem. Phys.*, 126(22):224101–9, June 2007.

[34] Yang Cao, Hong Li, and Linda Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121(9):4059–4067, September 2004.

[35] Yang Cao and Linda Petzold. Trapezoidal tau-lepaing formula for the stochastic simulation of biochemical systems. In *Proceedings of Foundations of Systems Biology in Engineering (FOSBE)*, pages 149–152, 2005.

[36] R.G. Carvajal, J. Ramirez-Angulo, A.J. Lopez-Martin, A. Torralba, J.A.G. Galan, A. Carlosena, and F.M. Chavero. The flipped voltage follower: A useful

cell for low-voltage low-power circuit design. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(7):1276–1291, 2005.

[37] M. Catrysse, B. Hermans, and R. Puers. An inductive power system with integrated bi-directional data-transmission. *Sensors and Actuators A*, 115(2-3):221–229, September 2004.

[38] G. J. Chaitin. Gödel's theorem and information. *International Journal of Theoretical Physics*, 21(12):941–954, December 1982.

[39] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen. Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4):473–484, April 1992.

[40] Abhijit Chatterjee, Dionisios G. Vlachos, and Markos A. Katsoulakis. Binomial distribution based tau-leap accelerated stochastic simulation. *J. Chem. Phys.*, 122(2):024112–7, January 2005.

[41] L. J. Chu. Physical limitations of omni-directional antennas. *Journal of Applied Physics*, 19:1163–1175, December 1948.

[42] M. A. Cohen and J. A. Taylor. Short-term cardiovascular oscillations in man: Measuring and modelling the physiologies. *Journal of Physiology*, 542(3):669–683, 2002.

[43] G. E. R. Cowan, R. C. Melville, and Y. P. Tsividis. A VLSI analog computer/digital computer accelerator. *IEEE J. Solid-State Circ.*, 41(1):42–53, January 2006.

[44] G. E. R. Cowan, R. C. Melville, and Y. P. Tsividis. A VLSI analog computer/digital computer accelerator. *IEEE Journal of Solid-State Circuits*, 41(1):42–53, January 2006.

[45] J. P. Crutchfield, M. Mitchell, and R. Das. *The Evolutionary Design of Collective Computation in Cellular Automata*, Chapter 14, pages 361–407. Santa Fe

Institute Studies on the Sciences of Complexity. Oxford University Press, New York, NY, First edition, 2003.

[46] Leticia F. Cugliandolo, Jorge Kurchan, and Luca Peliti. Energy flow, partial equilibration, and effective temperatures in systems with slow dynamics. *Phys. Rev. E*, 55(4):3898–3914, April 1997.

[47] P. Dallos and B. N. Evans. High-frequency motility of outer hair cells and the cochlear amplifier. *Science*, 267(5206):2006–2009, March 1995.

[48] P. Dallos, B. N. Evans, and R. Hallworth. Nature of the motor element in electrokinetic shape changes of Cochlear Outer Hair Cells. *Nature*, 350:155–157, March 1991.

[49] Xavier Darzacq, Yaron Shav-Tal, Valeria de Turris, Yehuda Brody, Shailesh M Shenoy, Robert D Phair, and Robert H Singer. In vivo dynamics of rna polymerase ii transcription. *Nat Struct Mol Biol*, 14(9):796–806, September 2007.

[50] S. W. Davies. Visualizing genetic circuits using concepts borrowed from electronics. *Proceedings of the Biomedical Circuits and Systems Conference (BioCAS)*, 1:123–125, November 2007.

[51] E. de Boer and H. R. de Jongh. On cochlear encoding: Potentialities and limitations of the reverse-correlation technique. *The Journal of the Acoustical Society of America*, 63(1):115–135, 1978.

[52] N. de N. Donaldson. Passive signalling via inductive coupling. *Medical and Biological Engineering and Computing*, 24(2):223–224, March 1986.

[53] J. J. M. de Vreede-Swagemakers, A. P. M. Gorgels, W. I. Dubois-Arbouw, J. W. van Ree, M. J. A. P. Daemen, L. G. E. Houben, and H. J. J. Wellens. Out-of-hospital cardiac arrest in the 1990s: A population-based study in the Maastricht area on incidence, characteristics and survival. *Journal of the American College of Cardiology*, 30(6):1500–1505, November 1997.

[54] R. A. DeCarlo, S. H. Zak, and G. P. Matthews. Variable structure control of nonlinear multivariable systems: A tutorial. *Proceedings of the IEEE*, 76(3):212–242, March 1988.

[55] T. Delbrück. *Investigations of Analog VLSI Visual Transduction and Motion Processing*. PhD thesis, California Institute of Technology, Computation and Neural Systems Program, January 1993.

[56] Tobi Delbrck and Andr Van Schaik. Bias current generators with wide dynamic range. *Analog Integrated Circuits and Signal Processing*, 43(3):247–268, June 2005.

[57] M. H. Devoret. Quantum fluctuations in electrical circuits. In *Quantum Fluctuations, Les Houches, Session LXIII*, pages 351–384, 1995.

[58] John K. Douglass, Lon Wilkens, Eleni Pantazelou, and Frank Moss. Noise enhancement of information transfer in crayfish mechanoreceptors by stochastic resonance. *Nature*, 365(6444):337–340, September 1993.

[59] G. Dubost and J. Dupuy. Effective area of an antenna. *Electronics Letters*, 12(4):98–99, February 1976.

[60] Mary J Dunlop, Robert Sidney Cox, Joseph H Levine, Richard M Murray, and Michael B Elowitz. Regulatory activity revealed by dynamic correlations in gene expression noise. *Nat Genet*, 40(12):1493–1498, December 2008.

[61] E. H. Callaway, Jr., R. L. Barrett, Jr., G. J. Hernandez, and D. H. Weisman. U. S. Patent #6,768,398: Method and apparatus for creating a radio frequency filter. Patent, 2004.

[62] W.-K. Chen (Ed.). *The Circuits and Filters Handbook*. CRC Press, Boca Raton, FL, First edition, 1995.

[63] Michael B. Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, January 2000.

[64] Radek Erban, Jonathan Chapman, and Philip Maini. A practical guide to stochastic simulations of reaction-diffusion processes. *arXiv:0704.1908v2, available at http://arxiv.org/abs/0704.1908v2*, 2007.

[65] P. Erdős and M. Kac. The gaussian law of errors in the theory of additive number theoretic functions. *American Journal of Mathematics*, 62(1):738–742, January 1940.

[66] A. Aldo Faisal and Simon B Laughlin. Stochastic simulations on the reliability of action potential propagation in thin axons. *PLoS Comput Biol*, 3(5):e79–, May 2007.

[67] A. Aldo Faisal, Luc P. J. Selen, and Daniel M. Wolpert. Noise in the nervous system. *Nat Rev Neurosci*, 9(4):292–303, April 2008.

[68] R. M. Fano. *Theoretical Limitations on the Broadband Matching of Arbitrary Impedances*. DSc dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering, May 1947.

[69] R. M. Fano. Theoretical limitations on the broadband matching of arbitrary impedances. *Journal of the Franklin Institute*, 249(1):57–83, January 1950.

[70] R. M. Fano. Theoretical limitations on the broadband matching of arbitrary impedances. *Journal of the Franklin Institute*, 249(2):139–154, February 1950.

[71] P. Fatt and B. Katz. Some observations on biological noise. *Nature*, 166(4223):597–598, October 1950.

[72] K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley, Chicester, Sussex, UK, Second edition, 2003.

[73] K. Ford. The distribution of integers with a divisor in a given interval. *Annals of Mathematics*, 168(2):367–433, September 2008.

[74] G. Frank, W. Hemmert, and A. W. Gummer. Limiting dynamics of high-frequency electromechanical transduction of outer hair cells. *Proc. Natl. Acad. Sci. USA*, 96:4420–4425, April 1999.

[75] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3-4):219–253, April 1982.

[76] D. R. Frey. Log-domain filtering: An approach to current-mode filtering. *IEE Proceedings-G: Circuits, Devices and Systems*, 140(6):406–416, December 1993.

[77] D. Gabor. Theory of communication. *Journak of the Institute of Electrical Engineers (London)*, 93(26):429–457, November 1946.

[78] D. Gabor. Communication theory and physics. *Transactions of the IRE Professional Group on Information Theory*, 1(1):48–59, Feb. 1953.

[79] C. Galbraith, R. White, K. Grosh, and G. M. Rebeiz. A mammalian cochlea-based RF channelizing filter. *IEEE MTT-S International Microwave Symposium Digest*, pages 1935–1938, June 2005.

[80] C. Galbraith, R. D. White, L. Cheng, K. Grosh, and G. M. Rebeiz. Cochlea-based RF channelizing filters. *IEEE Transactions on Circuits and Systems-I*, 55(4):969–979, 2008.

[81] D. C. Galbraith, M. Soma, and R. L. White. A wide-band efficient inductive transdermal power and data link with coupling insensitive gain. *IEEE Transactions on Biomedical Engineering*, 34(4):265–275, April 1987.

[82] D. C. Galbraith, M. Soma, and R. L. White. Electronic design of a cochlear implant for multichannel high-rate pulsatile stimulation strategies. *IEEE Transactions on Rehabilitation Engineering*, 3(1):112–116, March 1995.

[83] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni. Stochastic resonance. *Reviews of Modern Physics*, 70(1):223–287, January 1998.

[84] Timothy J. Gardner and Marcelo O. Magnasco. Sparse time-frequency representations. *Proceedings of the National Academy of Sciences*, 103(16):6094–6099, 2006.

[85] C. D. Geisler. *From Sound to Synapse: Physiology of the Mammalian Ear*. Oxford University Press, Oxford, U.K., First edition, 1998.

[86] J. Georgiou and C. Toumazou. A 126-μw cochlear chip for a totally implantable system. *IEEE Journal of Solid-State Circuits*, 40(2):430–443, February 2005.

[87] M. Ghovanloo and S. Atluri. A wide-band power-efficient inductive wireless link for implantable microelectronic devices using multiple carriers. *IEEE Transactions on Circuits and Systems-I*, 54(10):2211–2221, October 2007.

[88] M. Ghovanloo and S. S. Atluri. An integrated full-wave CMOS rectifier with built-in back telemetry for RFID and implantable biomedical applications. *Accepted for publication, IEEE Transactions on Circuits and Systems-I*, 2008.

[89] M. Ghovanloo and K. Najafi. A wideband frequency-shift keying wireless link for inductively powered biomedical implants. *IEEE Transactions on Circuits and Systems-I*, 51(12):2374–2383, December 2004.

[90] Michael A. Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.

[91] B. Gilbert. Translinear circuits: A proposed classification. *Electronics Letters*, 11(1):14–16, January 1975.

[92] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, December 1976.

[93] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[94] Daniel T. Gillespie. The chemical langevin equation. *J. Chem. Phys.*, 113(1):297–306, July 2000.

[95] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115(4):1716–1733, July 2001.

[96] Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, 2007.

[97] Carlos A. Gomez-Uribe and George C. Verghese. Mass fluctuation kinetics: Capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. *J. Chem. Phys.*, 126(2):024109–12, January 2007.

[98] I. Grech, J. Micallef, and T. Vladimirova. Silicon cochlea and its adaptation to spatial localisation. *IEE Proceedings on Circuits, Devices and Systems*, 146(2):70–76, April 1999.

[99] R. Grima and S. Schnell. A systematic investigation of the rate laws valid in intracellular environments. *Biophysical Chemistry*, 124(1):1–10, October 2006.

[100] G. P. Grimmett and D. R. Stirzaker. *Probability and Random Processes*, volume 1. Oxford University Press, Oxford, UK, Third edition, 2001.

[101] B. J. Gross. $1/f$ *Noise in MOSFETs with Ultrathin Gate Dielectrics*. PhD thesis, Massachusetts Institute of Technology, Electrical Engineering and Computer Science, June 1992.

[102] A. Hajimiri. Distributed integrated circuits: An alternative approach to high-frequency design. *IEEE Comm. Mag.*, 40(2):168–173, February 2002.

[103] A. Hajimiri and T.H. Lee. A general theory of phase noise in electrical oscillators. *IEEE Journal of Solid-State Circuits*, 33(2):179–194, 1998.

[104] A. Hajimiri, S. Limotyrakis, and T.H. Lee. Jitter and phase noise in ring oscillators. *IEEE Journal of Solid-State Circuits*, 34(6):790–804, 1999.

[105] D. Ham and A. Hajimiri. Virtual damping and Einstein relation in oscillators. *IEEE Journal of Solid-State Circuits*, 38(3):407–418, 2003.

[106] Z. Hamici, R. Itti, and J. Champier. A high-efficiency biotelemetry system for implanted electronic device. *Proceedings of the IEEE Annual Conference, Engineering in Medicine and Biology Society (EMBS)*, 2:1649–1650, September 1995.

[107] P.H. Handel and D. Wolf. Amplitude distribution of $1/f$ noise. In *Proc. of the 5th Intl. Conf. on 'Noise in Physical Systems'*, pages 125–130, 1969.

[108] S. L. Harding, J. F. Miller, and E. A. Reitman. Evolution in materio: Exploiting the physics of materials for computation. *International Journal of Unconventional Computing*, 4(2):155–194, April 2008.

[109] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher. A low-power integrated circuit for a wireless 100-electrode neural recording system. *IEEE Journal of Solid-State Circuits*, 42(1):123–133, January 2007.

[110] S. M. R. Hasan. A novel mixed-signal integrated circuit model for DNA-protein regulatory genetic circuits and genetic state machines. *IEEE Transactions on Circuits and Systems-I*, 55(5):1185–1196, June 2008.

[111] Alan Hastings. *The Art of Analog Layout*. Prentice Hall, Upper Saddle River, NJ, Second edition, 2005.

[112] Johan Hattne, David Fange, and Johan Elf. Stochastic reaction-diffusion simulation with mesord. *Bioinformatics*, 21(12):2923–2924, 2005.

[113] Hermann A. Haus. *Electromagnetic Noise and Quantum Optical Measurements*. Springer, Berlin, Germany, First edition, 2000.

[114] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, February 2005.

[115] M. Hazas, J. Scott, and J. Krumm. Location-aware computing comes of age. *IEEE Computer*, 37(2):95–97, February 2004.

[116] F. Herzel and B. Razavi. A study of oscillator jitter due to supply and substrate noise. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(1):56–62, 1999.

[117] T. Hinck, Z. Yang, Q. Zhang, and A. E. Hubbard. A current-mode implementation of a traveling wave amplifier model similar to the cochlea. *Proc. IEEE Intl. Symposium on Circuits and Systems*, 2:228–231, June 1999.

[118] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, July 2006.

[119] F. N. Hooge and A. M. H. Hoppenbrouwers. Amplitude distribution of $1/f$ noise. *Physica (Amsterdam)*, 42:331–339, 1969.

[120] L. Van Hove. Correlations in space and time and born approximation scattering in systems of interacting particles. *Physical Review*, 95(1):249–262, July 1954.

[121] Y. Hu and M. Sawan. A fully integrated low-power BPSK demodulator for implantable medical devices. *IEEE Transactions on Circuits and Systems-I*, 52(12):2552–2562, December 2005.

[122] Q. Huang and M. Oberle. A 0.5-mw passive telemetry IC for biomedical applications. *IEEE Journal of Solid-State Circuits*, 33(7):937–946, July 1998.

[123] A. Hubbard. A traveling-wave amplifier model of the cochlea. *Science*, 259:68–71, January 1993.

[124] K. K. Hung, P. K. Ko, C. Hu, and Y. C. Cheng. A unified model for the flicker noise in metal-oxide-semiconductor field-effect transistors. *IEEE Transactions on Electron Devices*, 37(3):654–665, March 1990.

[125] Rolf Hut, Marinus M. Boone, and Andries Gisolf. Cochlear modeling as time-frequency analysis tool. *Acta Acustica united with Acustica*, 92:629–636, July/August 2006.

[126] IEEE. *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, Cambridge, MA, April 2006. IEEE.

[127] J. Mitola III. The software radio architecture. *IEEE Communications Magazine*, 33(5):26–38, May 1995.

[128] J. Mitola III. Software radio architecture: A mathematical perspective. *IEEE Journal on Selected Areas in Communications*, 17(4):514–538, April 1999.

[129] A. Ilachinski. *Cellular Automata: A Discrete Universe*. World Scientific Publishing Company, Singapore, First edition, 2001.

[130] M. H. Jakubowski, K. Steiglitz, and R. K. Squier. Computing with solitons: A review and prospectus. *Multiple-Valued Logic*, 6(5-6):439–462, 2001.

[131] M. H. Jakubowski, K. Steiglitz, and R. K. Squier. When can solitons compute? *Complex Systems*, 10(1):1–21, January 2001.

[132] G. Job and F. Herrmann. Chemical potential–a quantity in search of recognition. *European Journal of Physics*, 27(2):353–371, 2006.

[133] David F. Tuttle Jr. *Network Synthesis*, volume 1, Chapter 12, pages 710–722. John Wiley, New York, First edition, 1958.

[134] Mads Kaern, Timothy C. Elston, William J. Blake, and James J. Collins. Stochasticity in gene expression: From theories to phenotypes. *Nat Rev Genet*, 6(6):451–464, June 2005.

[135] Shai Kaplan, Anat Bren, Alon Zaslaver, Erez Dekel, and Uri Alon. Diverse two-dimensional input functions control bacterial sugar genes. *Molecular Cell*, 29(6):786–792, March 2008.

[136] U. Karthaus and M. Fischer. Fully integrated passive UHF RFID transponder IC with 16.7μw minimum RF input power. *IEEE Journal of Solid-State Circuits*, 38(10):1602–1608, October 2003.

[137] J. M. Kates. A time-domain digital cochlear model. *IEEE Transactions on Signal Processing*, 39(12):2573–2592, December 1991.

[138] A. G. Katsiamis, E. M. Drakakis, and R. F. Lyon. A biomimetic, 4.5μw, 120+ db, log-domain cochlea channel with AGC. *IEEE Journal of Solid-State Circuits*, 44(3):1006–1022, March 2009.

[139] K. F. Knott. Measurement of battery noise and resistor-current noise at sub-audio frequencies. *Electronics Letters*, 1(5):132–133, July 1965.

[140] Raoul Kopelman. Fractal reaction kinetics. *Science*, 241(4873):1620–1626, Sep. 23, 1988.

[141] G.A. Korn. Progress of analog/hybrid computation. *Proceedings of the IEEE*, 54(12):1835–1849, 1966.

[142] G.A. Korn and R. Vichnevetsky. Analog/hybrid computation and digital simulation. *IEEE Transactions on Computers*, C-25(12):1312–1320, 1976.

[143] Granino A. Korn. *Random-Process Simulation and Measurements*. McGraw Hill Book Company, First edition, 1966.

[144] B. Kosko. *Noise*. Viking Penguin, New York, NY, First edition, 2006.

[145] S. S. Kraman, G. R. Wodicka, Y. Oh, and H. Pasterkamp. Measurements of respiratory acoustic signals. Effect of microphone air cavity width, shape and venting. *Chest*, 108(4):1004–1008, October 1995.

[146] G. Kron. Equivalent circuit of the field equations of Maxwell-I. *Proceedings of the I.R.E.*, 32(5):289–299, May 1944.

[147] G. Kron. Tensorial analysis and equivalent circuits of elastic structures. *Journal of the Franklin Institute*, 238(6):399–442, December 1944.

[148] G. Kron. Electric circuit models of the Schrödinger equation. *Physical Review*, 67(1):39–43, January 1945.

[149] G. Kron. Equivalent circuits of compressible and incompressible fluid flow fields. *Journal of the Aeronautical Sciences*, 12:221–231, April 1945.

[150] G. Kron. Numerical solution of ordinary and partial differential equations by means of equivalent circuits. *Journal of Applied Physics*, 16:174–188, March 1945.

[151] G. Kron. Electric circuit models for the vibration spectra of polyatomic molecules. *The Journal of Chemical Physics*, 14(1):19–31, January 1946.

[152] R. Kubo. The fluctuation-dissipation theorem. *Reports on Progress in Physics*, 29(1):255–284, 1966.

[153] R. Landauer. Dissipation and noise immunity in computation and communication. *Nature*, 335(27):779–784, October 1988.

[154] R. Landauer. Information is physical. *Physics Today*, 44(5):23–29, May 1991.

[155] M. A. G. Laso, T. Lopetegi, M. J. Erro, D. Benito, M. J. Garde, M. A. Muriel, M. Sorolla, and M. Guglielmi. Real-time spectrum analysis in microstrip technology. *IEEE Transactions on Microwave Theory and Techniques*, 51(3):705–717, March 2003.

[156] Melvin Lax. Classical noise. v. noise in self-sustained oscillators. *Phys. Rev.*, 160(2):290–307, August 1967.

[157] J. Lazzaro. A silicon model of an auditory neural representation of spectral shape. *IEEE Journal of Solid-State Circuits*, 26(5):772–777, May 1991.

[158] T. H. Lee. *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge Uinversity Press, Cambridge, UK, Second edition, 2004.

[159] T.H. Lee and A. Hajimiri. Oscillator phase noise: A tutorial. *IEEE Journal of Solid-State Circuits*, 35(3):326–336, 2000.

[160] D.B. Leeson. A simple model of feedback oscillator noise spectrum. *Proceedings of the IEEE*, 54(2):329–330, 1966.

[161] K. S. Leong, M. L. Ng, and P. H. Cole. Operational considerations in simulation and deployment of RFID systems. *Proc. 17th Intl. Zurich Symp. on Electromagnetic Compatibility*, 1:521–524, February 2006.

[162] B. L. Lewis. Antenna effective area and directivity viewed from photon theory. *IEEE Transactions on Antennas and Propagation*, 21(2):229–230, March 1973.

[163] Hong Li and Linda Petzold. Efficient parallelization of stochastic simulation algorithm for chemically reacting systems on the graphics processing unit. *submitted, preprint available at http://www.cs.ucsb.edu/ cse/*, 2008.

[164] Eckhard Limpert, Werner A. Stahel, and Markus Abbt. Log-normal distributions across the sciences: Keys and clues. *BioScience*, 51(5):341–352, 2001.

[165] W. Liu, K. Vichienchom, M. Clements, S. C. DeMarco, C. Hughes, E. McGucken, M. S. Humayun, E. de Juan, J. D. Weiland, and R. Greenberg. A neuro-stimulus chip with telemetry unit for retinal prosthetic device. *IEEE Journal of Solid-State Circuits*, 35(10):1487–1497, October 2000.

[166] Z. Liu and Y. Cao. Detailed comparison between StochSim and SSA. *IET Syst. Biol.*, 2(5):334–341, September 2008.

[167] S. Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, August 2000.

[168] J. D. Logan. *An Introduction to Nonlinear Partial Differential Equations*. Wiley-Interscience, Malden, MA, Second edition, 2008.

[169] Larry Lok and Roger Brent. Automatic generation of cellular reaction networks with moleculizer 1.0. *Nat Biotech*, 23(1):131–136, January 2005.

[170] J. R. Long. Monolithic transformers for silicon RF IC design. *IEEE Journal of Solid-State Circuits*, 35(9):1368–1382, September 2000.

[171] A. R. Lopez. Review of narrowband impedance-matching limitations. *IEEE Antennas and Propagation Magazine*, 46(4):88–90, 2004.

[172] Richard Losick and Claude Desplan. Stochasticity and cell fate. *Science*, 320(5872):65–68, April 2008.

[173] T. K.-T. Lu. A feedback analysis of outer hair cell dynamics. M. Eng thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, May 2003.

[174] Ting Lu, Jeff Hasty, and Peter G. Wolynes. Effective temperature in stochastic kinetics and gene networks. *Biophys. J.*, 91(1):84–94, July 2006.

[175] Ting Lu, Tongye Shen, Matthew R. Bennett, Peter G. Wolynes, and Jeff Hasty. Phenotypic variability of growing cellular populations. *Proceedings of the National Academy of Sciences*, 104(48):18982–18987, November 2007.

[176] J. Ludwig, D. Oliver, G. Frank, N. Klöcker, A. W. Gummer, and B. Fakler. Reciprocal electromechanical properties of rat prestin: The motor molecule from rat outer hair cells. *Proc. Natl. Acad. Sci. USA*, 98(7):4178–4183, March 2001.

[177] R. F. Lyon and C. A. Mead. An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1119–1134, July 1988.

[178] Richard F. Lyon and Carver A. Mead. Cochlear hydrodynamics demystified. Technical report, California Institute of Technology, Pasadena, CA, USA, 1988.

[179] Hedia Maamar, Arjun Raj, and David Dubnau. Noise in gene expression determines cell fate in Bacillus subtilis. *Science*, 317(5837):526–529, July 2007.

[180] E. F. Macnichol. An analog computer to simulate systems of coupled bimolecular reactions. *Proceedings of the IRE*, 47(11):1816–1820, 1959.

[181] M. O. Magnasco. Chemical kinetics is Turing universal. *Physical Review Letters*, 78(6):1190–1193, February 1997.

[182] Narendra Maheshri and Erin K. OShea. Living with noisy genes: How cells function reliably with inherent variability in gene expression. *Annu. Rev. Biophys. Biomol. Struct.*, 36(1):413–, June 2007.

[183] J. Makino and M. Taiji. *Scientific Simulations with Special-Purpose Computers - the GRAPE Systems*. John Wiley, New York, NY, First edition, 1998.

[184] S. Mandal, S. K. Arfin, and R. Sarpeshkar. Fast startup CMOS current references. In *Proc. IEEE Symposium on Circuits and Systems (ISCAS)*, May 2006.

[185] S. Mandal and R. Sarpeshkar. A bidirectional wireless link for neural prostheses that minimizes implanted power consumption. *IEEE Biomedical Circuits and Systems Conference (BioCAS), Montreal, Canada*, November 2007.

[186] S. Mandal and R. Sarpeshkar. Low power CMOS rectifier design for RFID applications. *IEEE Transactions on Circuits and Systems-I*, 54(6):1177–1188, June 2007.

[187] S. Mandal and R. Sarpeshkar. Power-efficient impedance-modulation wireless data links for biomedical implants. *IEEE Transactions on Biomedical Circuits and Systems*, 2(4):301–315, 2008.

[188] S. Mandal, S. Zhak, and R. Sarpeshkar. Circuits for an RF cochlea. *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 3610–3613, May 2006.

[189] J. G. Maneatis. Low-jitter process-independent DLL and PLL based on self-biased techniques. *IEEE Journal of Solid-State Circuits*, 31(11):1723–1732, November 1996.

[190] N. Margolus, T. Toffoli, and G. Vichniac. Cellular-automata supercomputers for fluid-dynamics modeling. *Physical Review Letters*, 56(16):1694–1696, April 1986.

[191] Bo Marr, Stephen Brink, Paul Hasler, and David V. Anderson. A reconfigurable, analog system for efficient stochastic biological computation. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 293–296, 2008.

[192] A. J. Martin. Towards an energy complexity of computation. *Information Processing Letters*, 77(2):181–187, February 2001.

[193] A. J. Martin and M. Nyström. Asynchronous techniques for system-on-chip design. *Proceedings of the IEEE*, 94(6):1089–1120, June 2006.

[194] G. Matthaei, E. M. T. Jones, and L. Young. *Microwave Filters, Impedance-Matching Networks and Coupling Structures*. Artech House, Boston, First edition, 1980.

[195] James McCollum, Gregory Peterson, Chris Cox, Michael Simpson, and Nagiza Samatova. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, 30(1):39–49, February 2006.

[196] J. S. McLean. A re-examination of the fundamental limits on the radiation q of electrically small antennas. *IEEE Transactions on Antennas and Propagation*, 44(5):672–676, May 1996.

[197] J.A. McNeill. A simple method for relating time- and frequency-domain measures of oscillator performance. In *Mixed-Signal Design, 2001. SSMSD. 2001 Southwest Symposium on*, pages 7–12, 2001.

[198] Donald A. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4(3):413–478, December 1967.

[199] C. A. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA, First edition, 1989.

[200] R. G. Meny, J. L. Carroll, M. T. Carbone, and D. H. Kelly. Cardiorespiratory recordings from infants dying suddenly and unexpectedly at home. *Pediatrics*, 93(1):44–49, January 1994.

[201] Nicholas Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[202] A. C. H. MeVay and R. Sarpeshkar. Predictive comparators with adaptive control. *IEEE Transactions on Circuits and Systems-II*, 50(9):579–588, September 2003.

[203] Clark A. Miller and Daniel A. Beard. The effects of reversibility and noise on stochastic phosphorylation cycles and cascades. *Biophys. J.*, 95(5):2183–2192, September 2008.

[204] S. S. Mohan, M. del Mar Hershenson, S. P. Boyd, and T. H. Lee. Simple accurate expressions for planar spiral inductances. *IEEE Journal of Solid-State Circuits*, 34(10):1419–1424, October 1999.

[205] P. Mohseni, K. Najafi, S. J. Eliades, and X. Wang. Wireless multichannel biopotential recording using an integrated FM telemetry circuit. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3):263–271, September 2005.

[206] Brian C. J. Moore and Brian R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *The Journal of the Acoustical Society of America*, 74(3):750–753, 1983.

[207] CJ Morton-Firth and D Bray. Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology*, pages 117–128, May 1998.

[208] Z. M. K. Moussavi, editor. *IEEE Engineering in Medicine and Biology Magazine*, volume 26, chapter Special Issue on Respiratory Sound Analysis, pages 15–70. IEEE, January 2007.

[209] T. Munakata, S. Sinha, and W. L. Ditto. Chaos computing: Implementation of fundamental logical gates by chaotic elements. *IEEE Transactions on Circuits and Systems - I*, 49(11):1629–1633, November 2002.

[210] R. N. Mutagi. Pseudo noise sequences for engineers. *Electronics and Communication Engineering Journal*, 8(2):79–87, April 1996.

[211] T. Nakagaki and H. Yamada. Maze-solving by an amoeboid organism. *Nature*, 407:470–470, September 2000.

[212] M. Namiki, T. Hamamoto, and S. Hangai. Spoken word recognition with digital cochlea using 32 DSP-boards. *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2:969–972, May 2001.

[213] Siva G. Narendra and Anantha P. Chandrakasan. *Leakage in Nanometer CMOS Technologies*. Birkhäuser, First edition, 2006.

[214] P. A. Neukomm and H. Kündig. Passive wireless actuator control and sensor signal transmission. *Sensors and Actuators*, A21(1-3):258–262, February 1990.

[215] J. R. S. Newman, S. Ghaemmaghami, J. Imhels, D. K. Breslow, M. Noble, J. L. DeRisi, and J. S. Weissman. Single-cell proteomic analysis of *S. cerevisiae* reveals the architecture of biological noise. *Nature*, 441:840–846, June 2006.

[216] Martin A. Nowak. *Evolutionary Dynamics: Exploring the Equations of Life*. Belknap Press, September 2006.

[217] E. R. Nye. The effect of blood pressure alteration on the pulse wave velocity. *British Heart Journal*, 26(2):261–265, March 1964.

[218] M. O'Halloran and R. Sarpeshkar. A 10-nW 12-bit accurate analog storage cell with 10-aA leakage. *IEEE Journal of Solid-State Circuits*, 39(11):1985–1996, 2004.

[219] H. A. Om'mani and P. M. Furth. A subthreshold CMOS implementation of a two-dimensional mechanical cochlear model. *IEEE Midwest Symposium on Circuits and Systems*, 1:560–563, August 1998.

[220] V. Orlyanchik, V. I. Kozub, and Z. Ovadyahu. Non-Gaussian conductance noise in disordered electronic systems due to a nonlinear mechanism. *Physical Review B*, 74(23):235206, December 2006.

[221] Ertugrul M. Ozbudak, Mukund Thattai, Iren Kurtser, Alan D. Grossman, and Alexander van Oudenaarden. Regulation of noise in the expression of a single gene. *Nat Genet*, 31(1):69–73, May 2002.

[222] L. Pauling. *General Chemistry*. Dover Publications, Mineola, NY, Third edition, 1988.

[223] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers. Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits*, 24(5):1433–1439, 1989.

[224] J. D. Pelletier. The power spectral density of atmospheric temperature from time scales of 10^{-2} to 10^{6} yr. *Earth and Planetary Science Letters*, 158(3-4):157–164, May 1998.

[225] C.S. Petrie and J.A. Connelly. A noise-based IC random number generator for applications in cryptography. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(5):615–621, 2000.

[226] J. O. Pickles. *An Introduction to the Physiology of Hearing*. Academic Press, London, U.K., Second edition, 1988.

[227] M. B. Pour-El and I. Richards. The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics*, 39(3):215–239, March 1981.

[228] P. K. Prakasam, M. Kulkarni, X. Chen, Z. Yu, S. Hoyos, J. Silva-Martinez, and E. Sánchez-Sinencio. Applications of multipath transform-domain charge-sampling wide-band receivers. *IEEE Transactions on Circuits and Systems-II*, 55(4):309–313, April 2008.

[229] J. G. Proakis. *Digital Communications*. McGraw Hill Higher Education, New York, NY, Fourth edition, 2000.

[230] S. Puria and J. Allen. A parametric study of cochlear input impedance. *J. Acoust. Soc. Am.*, 89(1):287–309, January 1991.

[231] D. Python, M. Punzenberger, and C. Enz. A 1-V CMOS log-domain integrator. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2:685–688, June 1999.

[232] Arjun Raj and Alexander van Oudenaarden. Nature, nurture, or chance: Stochastic gene expression and its consequences. *Cell*, 135(2):216–226, October 2008.

[233] Christopher V. Rao, Denise M. Wolf, and Adam P. Arkin. Control, exploitation and tolerance of intracellular noise. *Nature*, 420(6912):231–237, November 2002.

[234] J. M. Raser and E. K. O'Shea. Control of stochasticity in eukaryotic gene expression. *Science*, 304:1811–1814, June 2004.

[235] Muruhan Rathinam, Linda R. Petzold, Yang Cao, and Daniel T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. Chem. Phys.*, 119(24):12784–12794, December 2003.

[236] B. Razavi. A study of phase noise in CMOS oscillators. *IEEE Journal of Solid-State Circuits*, 31(3):331–343, 1996.

[237] S. S. Reese and P. F. Neidhardt. U. S. patent #5,179,542: Signal processor. Patent, 1993.

[238] P. J. Restle, R. J. Hamilton, M.. Weissman, and M. S. Love. Non-Gaussian effects in $1/f$ noise in small silicon-on-sapphire resistors. *Physical Review B*, 31(4):2254–2262, February 1985.

[239] W. S. Rhode. *Measurement of the Amplitude and Phase of Vibration of the Basilar Memmbrane using the Mössbauer Effect*. PhD thesis, University of Wisconsin - Madison, 1970.

[240] W. S. Rhode. Observations of the vibration of the basilar membrane in squirrel monkeys using the Mössbauser technique. *Journal of the Acoustical Society of America*, 49(4):1218–1231, April 1971.

[241] D. S. Ricketts, X. Li, N. Sun, K. Woo, and D. Ham. On the self-generation of electrical soliton pulses. *IEEE J. Solid-State Circ.*, 42(8):1657–1668, August 2007.

[242] Marc Riedel and Jehoshua Bruck. Exact stochastic simulation with event leaping. In *Proceedings of the International Conference on Systems Biology*, 2005.

[243] F. Rivet, Y. Deval, J.-B. Begueret, D. Dallet, P. Cathelin, and D. Belot. A disruptive receiver architecture dedicated to software-defined radio. *IEEE Transactions on Circuits and Systems-II*, 55(4):344–348, April 2008.

[244] L. Robles and M. A. Ruggero. Mechanics of the mammalian cochlea. *Physiology Review*, 81(3):1305–1352, July 2001.

[245] M.J.W. Rodwell, S.T. Allen, R.Y. Yu, M.G. Case, U. Bhattacharya, M. Reddy, E. Carman, M. Kamegawa, Y. Konishi, J. Pusl, and R. Pullela. Active and nonlinear wave propagation devices in ultrafast electronics and optoelectronics. *Proceedings of the IEEE*, 82(7):1037–1059, 1994.

[246] Christopher J. Rozell. *Distributed Redundant Representations in Man-Made and Biological Sensing Systems*. PhD thesis, Rice University, Houston, TX, April 2007.

[247] Daniel L. Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Phys. Rev. Lett.*, 73(6):814–, August 1994.

[248] David F. Russell, Lon A. Wilkens, and Frank Moss. Use of behavioural stochastic resonance by paddle fish for feeding. *Nature*, 402(6759):291–294, November 1999.

[249] Hanna Salman and Erez Braun. Voltage dynamics of single-type voltage-gated ion-channel protein ensembles. *Phys. Rev. E*, 56(1):852–, July 1997.

[250] Hanna Salman, Yoav Soen, and Erez Braun. Voltage fluctuations and collective effects in ion-channel protein ensembles. *Phys. Rev. Lett.*, 77(21):4458–, November 1996.

[251] Lukasz Salwinski and David Eisenberg. In silico simulation of biological network dynamics. *Nat Biotech*, 22(8):1017–1019, August 2004.

[252] K. R. Santarelli. *On the Synthesis of Switched Output Feedback Controllers for Linear, Time-invariant Systems*. PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, February 2007.

[253] K. R. Santarelli, M. A. Dahleh, and A. Megretski. Optimal controller synthesis for second order LTI plants with switched output feedback. *Proceedings of the 45th IEEE Conference on Decision and Control*, 1:4127–4132, December 2006.

[254] R. Sarpeshkar. *Efficient Precise Computation with Noisy Components: Extrapolating from an Electronic Cochlea to the Brain*. PhD thesis, California Institute of Technology, Computation and Neural Systems, April 1997.

[255] R. Sarpeshkar. Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation*, 10(7):1601–1638, October 1998.

[256] R. Sarpeshkar. Traveling waves versus bandpass filters: The silicon and biological cochlea. In H. Wada, editor, *Proceedings of the International Symposium on*

Recent Developments in Auditory Mechanics, pages 216–222. World Scientific, 2000.

[257] R. Sarpeshkar, T. Delbruck, and C. Mead. White noise in MOS transistors and resistors. *IEEE Circuits and Devices*, 9(6):23–29, June 1993.

[258] R. Sarpeshkar, R. F. Lyon, and C. A. Mead. A low-power wide-linear-range transconductance amplifier. *Analog Integrated Circuits and Signal Processing*, 13(1):123–151, May 1997.

[259] R. Sarpeshkar, R. F. Lyon, and C. A. Mead. A low-power wide-dynamic-range analog VLSI cochlea. *Analog Integrated Circuits and Signal Processing*, 16(3):245–274, August 1998.

[260] R. Sarpeshkar and M. G. O'Halloran. Scalable hybrid computation with spikes. *Neural Computation*, 14(9):2003–2024, September 2002.

[261] R. Sarpeshkar, C. Salthouse, J.-J. Sit, M. W. Baker, S. M. Zhak, T. K.-T. Lu, L. Turicchia, and S. Balster. An ultra-low-power programmable analog bionic ear processor. *IEEE Transactions on Biomedical Engineering*, 52(4):711–727, April 2005.

[262] R. Sarpeshkar, W. Wattanapanitch, B. I. Rapoport, S. K. Arfin, M. W. Baker, S. Mandal, M. S. Fee, S. Musallam, and R. A. Andersen. Low-power circuits for brain-machine interfaces. *Proc. IEEE Intl. Symposium on Circuits and Systems (ISCAS)*, May 2007.

[263] J. Schmidhuber. *A Computer Scientist's View of Life, the Universe, and Everything*, volume 1337 of *Lecture Notes in Computer Science*, Chapter 18, pages 201–208. Springer-Verlag, Berlin, Germany, First edition, 1997.

[264] W. Schottky. Small-shot effect and flicker effect. *Physical Review*, 28(1):74–103, January 1926.

[265] J. D. Schwartz, J. Azaña, and D. V. Plant. Experimental demonstration of real-time spectrum analysis using dispersive microstrip. *IEEE Microwave and Wireless Components Letters*, 16(4):215–217, April 2006.

[266] J. Scott and B. Dragovic. Audio location: Accurate low-cost location sensing. In *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 1–18. Springer, may 2005.

[267] A. D. Selbst. Clock division as a power saving strategy in a system constrained by high transmission frequency and low data rate. M. eng thesis, Massachusetts Institute of Technology, 2005.

[268] S. Shamma. On the role of space and time in auditory processing. *Trends In Cognitive Science*, 5(8), August 2001.

[269] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, M. P. Eastwood, J. Gagliardo, J.P. Grossman, C. R. Ho, D. J. Ierardi, I. Kolossváry, J. L. Klepeis, T. Layman, C. McLeavey, M. A. Moraes, R. Mueller, E. C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, and S. C. Wang. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51(7):91–97, July 2008.

[270] C. A. Shera. Intensity-invariance of fine time structure in basilar-membrane click responses: Implications for cochlear mechanics. *Joural of the Acoustical Society of America*, 110(1):332–348, July 2001.

[271] C. A. Shera. Mammalian spontaneous otoacoustic emissions are amplitude-stabilized cochlear standing waves. *Joural of the Acoustical Society of America*, 114(1):244–262, July 2003.

[272] H. Shiraishi and A. van Schaik. A 2-dimensional active cochlear model for analog VLSI implementation. *IEEE Intl. Symposium on Circuits and Systems*, 4:924–927, May 2004.

[273] B. Smith, Z. Tang, M. W. Johnson, S. Pourmehdi, M. H. Gazdik, J. R. Buckett, and P. H. Peckham. An externally-powered, multichannel, implantable stimulator-telemeter for control of paralyzed muscle. *IEEE Transactions on Biomedical Engineering*, 45(4):463–475, April 1998.

[274] E. Smith and M. S. Lewicki. Efficient auditory coding. *Nature*, 439:978–982, February 2006.

[275] J. R. Smith, A. P. Sample, P. S. Powledge, S. Roy, and A. Mamishev. A wirelessly-powered platform for sensing and computation. *Proc. Ubicomp 2006: Eighth International Conference on Ubiquitous Computing*, 1:495–506, September 2006.

[276] T. Soorapanth and S. S. Wong. A 0-dB IL 2140±30 MHz bandpass filter using Q-enhanced spiral inductors in standard CMOS. *IEEE Journal of Solid-State Circuits*, 37(5):579–586, May 2002.

[277] L. F. Spietz. *The Shot Noise Thermometer*. PhD dissertation, Yale University, Department of Applied Physics, May 2006.

[278] G. J. Suaning and N. H. Lovell. CMOS neurostimulation ASIC with 100 channels, scaleable output, and bidirectional radio-frequency telemetry. *IEEE Transactions on Biomedical Engineering*, 48(2):248–260, February 2001.

[279] Gurol M. Suel, Rajan P. Kulkarni, Jonathan Dworkin, Jordi Garcia-Ojalvo, and Michael B. Elowitz. Tunability and noise dependence in differentiation dynamics. *Science*, 315(5819):1716–1719, March 2007.

[280] Ian A. Swinburne and Pamela A. Silver. Intron delays and transcriptional timing during development. *Developmental Cell*, 14(3):324–330, March 2008.

[281] Zoltan Szallasi, Jrg Stelling, and Vipul Periwal, editors. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. MIT Press, 2006.

[282] I. Tagkopoulos, Y.-C. Liu, and S. Tavazoie. Predictive behavior within microbial genetic networks. *Science*, 320:1313–1317, June 2008.

[283] M. Taiji, T. Narumi, Y. Ohno, N. Futatsugi, A. Suenaga, N. Takada, and A. Konagaya. Protein explorer: A petaflops special-purpose computer system for molecular dynamics simulations. *ACM/IEEE Conference on Supercomputing*, 1:15–24, November 2003.

[284] K. Tanaka, M. Abe, and S. Ando. A novel mechanical cochlea "fishbone" with dual sensor/actuator characteristics. *IEEE/ASME Trans. Mechatronics*, 3(2):98–105, June 1998.

[285] Z. Tang, B. Smith, J. H. Schild, and P. H. Peckham. Data transmission from an implantable biotelemeter by load-shift keying using circuit configuration modulator. *IEEE Transactions on Biomedical Engineering*, 42(5):524–528, May 1995.

[286] Agilent Technologies. Spectrum analysis basics. Application Note 150, Agilent Technologies, 2004.

[287] L. Theogarajan, J. Wyatt, J. Rizzo, B. Drohan, M. Markova, S. Kelly, G. Swider, M. Raj, D. Shire, M. Gingerich, J. Lowenstein, and B. Yomtov. Minimally invasive retinal prosthesis. *Proc. IEEE Intl. Solid-State Circuits Conference (ISSCC)*, pages 99–108, February 2006.

[288] P. R. Troyk, I. E. Brown, W. H. Moore, and G. E. Loeb. Development of BION^{TM} technology for functional electrical stimulation: Bidirectional telemetry. *Proceedings of the IEEE Annual Conference, Engineering in Medicine and Biology Society (EMBS)*, 2:1317–1320, October 2001.

[289] Marc Turcotte, Jordi Garcia-Ojalvo, and Grol M. Sel. A genetic timer through noise-induced stabilization of an unstable state. *Proceedings of the National Academy of Sciences*, 105(41):15732–15737, October 2008.

[290] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, January 1937.

[291] A. M. Turing. Intelligent machinery. Technical report, National Physical Laboratory, available online at http://www.alanturing.net/intelligent_machinery, July 1948.

[292] D. F. Tuttle. *Network Synthesis*, volume 1. John Wiley & Sons, Hoboken, NJ, First edition, 1953.

[293] A. Tympas and D. Dalouka. Metaphorical uses of an electric power network: Early computations of atomic particles and nuclear reactors. *metaphorik.de, Available online at http://www.metaphorik.de*, 12:65–84, June 2007.

[294] V. I. Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2):212–222, April 1977.

[295] A. van der Ziel. Noise in solid-state devices and lasers. *Proceedings of the IEEE*, 58(8):1178–1206, August 1970.

[296] A. van Schaik and E. Fragnière. Pseudo-voltage domain implementation of a 2-dimensional silicon cochlea. *IEEE Intl. Symposium on Circuits and Systems*, 3:185–188, May 2001.

[297] A. Vergis, K. Steiglitz, and B. Dickinson. The complexity of analog computation. *Mathematics and Computers in Simulation*, 28(2):91–113, April 1986.

[298] B. Vigoda. *Analog Logic: Continuous-Time Analog Circuits for Statistical Signal Processing*. PhD thesis, Massachusetts Institute of Technology, Program in Media Arts and Sciences, September 2003.

[299] B. Vigoda, J. Dauwels, M. Frey, N. Gershenfeld, T. Koch, H.-A. Loeliger, and P. Merkli. Synchronization of pseudorandom signals by forward-only message

passing with application to electronic circuits. *IEEE Transactions on Information Theory*, 52(8):3842–3851, August 2006.

[300] S. Vogel. *Life's Devices: The Physical World of Animals and Plants*. Princeton University Press, Princeton, NJ, First edition, 1988.

[301] Dmitri Volfson, Jennifer Marciniak, William J. Blake, Natalie Ostroff, Lev S. Tsimring, and Jeff Hasty. Origins of extrinsic variability in eukaryotic gene expression. *Nature*, 439(7078):861–864, February 2006.

[302] Andreas Wagner. *Robustness and Evolvability in Living Systems*. Princeton University Press, First edition, July 2007.

[303] J. W. Wang, R. Sarpeshkar, M. Jabri, and C. Mead. A low power analog front end module for cochlear implants. *Proceedings of the XVI World Congress on Otorhinolaryngology, Head and Neck Surgery, Sydney, Australia*, March 1997.

[304] L. M. Ward. *Dynamical Cognitive Science*. MIT Press, Cambridge, MA, First edition, 2001.

[305] L. Watts. *Cochlear Mechanics: Analysis and Analog VLSI*. PhD thesis, California Institute of Technology, Computation and Neural Systems, April 1993.

[306] L. Watts, D. A. Kerns, R. F. Lyon, and C. A. Mead. Improved implementation of the silicon cochlea. *IEEE Journal of Solid-State Circuits*, 27(5):692–700, May 1992.

[307] E. Weber and E. Gmelin. Transport properties of silicon. *Applied Physics A: Materials Science and Processing*, 53(2):136–140, August 1991.

[308] J. Weber. Vacuum fluctuation noise and dissipation. *Phys. Rev.*, 96(3):556–559, November 1954.

[309] J. Weber. Fluctuation dissipation theorem. *Phys. Rev.*, 101(6):1620–1626, March 1956.

[310] J. G. Webster. *Design of Pulse Oximeters*. Taylor & Francis, Abingdon, Oxfordshire, UK, First edition, 1997.

[311] B. Wen and K. Boahen. A 360-channel speech preprocessor that emulates the cochlear amplifier. *Procedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 556–557, February 2006.

[312] E. Wheeler. $1/f$ noise: A pedagogical review. *Invited talk to E-GLEA-2, Buenos Aires, available online at* http://arxiv.org/abs/physics/0204033, September 2001.

[313] H. A. Wheeler. Fundamental limitations of small antennas. *Proceedings of the I.R.E.*, 35(12):1479–1484, December 1947.

[314] R. D. White and K. Grosh. Microengineered hydromechanical cochlear model. *Proceedings of the National Academy of Sciences*, 102(2):1296–1301, February 2005.

[315] Kurt Wiesenfeld and Frank Moss. Stochastic resonance and the benefits of noise: From ice ages to crayfish and SQUIDs. *Nature*, 373(6509):33–36, January 1995.

[316] E. M. Williams. Radio-frequency spectrum analyzers. *Proceedings of the IRE*, 34(1):18–22, January 1946.

[317] S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, First edition, 2002.

[318] Y. Xu. Chattering-free robust control for nonlinear systems. *IEEE Transactions on Control Systems Technology*, 16, 2008.

[319] A. D. Yaghjian and S. R. Best. Impedance, bandwidth and Q of antennas. *IEEE Transactions on Antennas and Propagation*, 53(4):1298–1324, April 2005.

[320] G.-Z. Yang, editor. *Body Sensor Networks*. Springer-Verlag, Berlin, Germany, First edition, 2006.

[321] X. Yang, K. Wang, and S. A. Shamma. Auditory representations of speech signals. *IEEE Trans. Info. Theory*, 38(2):824–839, March 1992.

[322] K. D. Young, V. I. Utkin, and Ü. Özgüner. A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 7(3):328–342, May 1999.

[323] S. M. Zhak. *Modeling and Design of an Active Silicon Cochlea*. Ph.d. thesis, Massachusetts Institute of Technology, September 2008.

[324] S. M. Zhak, S. Mandal, and R. Sarpeshkar. A proposal for an RF cochlea. *Proceedings of the Asia-Pacific Microwave Conference (APMC)*, December 2004.

[325] Huan-Xiang Zhou, Germn Rivas, and Allen P. Minton. Macromolecular crowding and confinement: Biochemical, biophysical, and potential physiological consequences. *Annu. Rev. Biophys.*, 37(1):375–397, June 2008.

[326] L. Zhou and N. Donaldson. A fast passive data transmission method for ENG telemetry. *Neuromodulation*, 6(2):116–121, April 2003.

[327] Konrad Zuse. Calculating space (translation of "Rechnender raum"). Technical Translation AZT-70-164-GEMIT, Massachusetts Institute of Technology, 1970.

[328] G. Zweig. Finding the impedance of the organ of Corti. *Journal of the Acoustical Society of America*, 89(3):1229–1254, March 1991.