Trust In Analog: Analog Circuit Techniques for Reducing the Risk of Malicious Circuits and Software

by

Eugene Kuznetsov

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

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2011

Author ....
Department of Electrical Engineering and Computer Science
January 31, 2011

Certified by ....
Professor James K. Roberge
Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Dr. Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

Trust In Analog: Analog Circuit Techniques for Reducing the Risk of Malicious Circuits and Software

by

Eugene Kuznetsov

Submitted to the Department of Electrical Engineering and Computer Science
on January 31, 2011, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Malicious circuits and software present a significant security risk, especially in control applications. This work is concerned with increasing the trustworthiness of control circuitry by reducing its complexity. The security benefits of substituting analog control techniques in place of digital control are analyzed, and both discrete and integrated circuit designs are demonstrated.

Thesis Supervisor: Professor James K. Roberge
Title: Professor of Electrical Engineering

Acknowledgments

I would like to thank my advisor Professor J.K. Roberge for the extensive advice and kind encouragement that made this project and many other things possible. I am grateful to Professor J. Dawson for access to essential technical resources. I also thank my family and friends.

Contents

List of Figures

List of Tables

Chapter 1

Trust in Analog

1.1 Introduction

This project is concerned with the design and demonstration of analog signal processing circuitry for functions that are usually implemented using digital circuitry today. These circuits will perform as well as their popular digital counterparts while offering one or more additional benefits. Before the rise of the software-dominated digital computer, signal processing, feedback control, and computation were performed using analog means. Recently, there has been some renewed interest in these techniques in specific situations, such as efficient analog computation [CMT06] and analog logic [Vig03], or low-power consumption medical implants [ASC$^+$08]. This research project will build on both of these areas, but has a different primary motivation.

1.2 Motivation

In 2007, DARPA launched a research program called "TRUST in Integrated Circuits", which was in turn motivated by a 2005 Defense Science Board task force report on "High Performance Microchip Supply". It seeks to address a national security concern associated with use of complex chips in defense and infrastructure systems while semiconductor manufacturing and design are increasingly performed outside the US or have other potential vulnerabilities [Col07]. The program objectives include hardware validation of an IC to determine whether it matches the intended design, comparing ICs to determine whether they are identical, preventing or detecting addition of malicious circuitry, trusted design of ASICs and FPGAs, and system integration of the individual capabilities into an overall framework. The initial goals included developing fast analysis techniques suitable for evaluation of 100K, 1M and 100M transistors in hundreds of hours [Wil07], [Sha07].

This project, Trust In Analog, is motivated primarily by the idea that the trustworthiness of electronic systems can be improved if some or all of their functions are implemented using analog techniques. Such an approach operates by seeking to reduce the number of circuit elements to be verified and to eliminate digital signal paths into the chip that could be used to introduce malicious commands (such as a "kill chip" code [Ade08]). The research is expected to be complementary to the ongoing work under the DARPA program.

1.3 Goals

One of the goals of this work was to learn whether one or more of the following can be achieved by replacing select digital circuitry with analog circuitry: reduce complexity as measured by number of active and passive elements, reduce effort required to validate an IC, improve one or more aspects of application-specific performance, reduce power consumption. The project also establishes criteria for what types of digital sub-systems are suitable for replacement, to outline a reusable toolbox of analog building blocks, and to find other means of making analog solutions more readily accessible and appealing for industrial use in trusted applications.

What we've sought to demonstrate is not only that it is possible to implement certain functions in the analog domain and such implementations can offer an advantage in trust or performance, but also that they can be made convenient and even attractive to the end user.

1.4 Types of Functions

Examples of the types of functions that are expected to be of interest are: i. signal analysis ii. feedback control / servo loops iii. simple computation. The functions are tied to applications and may require taking a whole-system view, such as replacing an entire microcontroller-based servo loop with an analog servo loop rather than attempting to replace the digital microcontroller with an analog one. Obviously, replication of really complex or general-purpose digital functions, such as a modern CPU, in purely analog design has not been a goal of this effort. However, such breadth of potential functions at the outset has not lead to a lack of focus as the project progressed, with several circuit topologies (such as the multiplier) receiving considerable attention.

A specific candidate function is a general-purpose feedback controller suitable for easy configuration and control of a broad range of systems. It would include signal conditioning, compensation and output drive sections in order to eliminate or minimize the number of external components required.

1.5 Initial Implementation Objectives

The goal has been to design and prototype a series of general-purpose building blocks for control and signal processing in trusted systems. Design was based on hand-calculations, with some modeling or simulation where appropriate. The prototyping consists of two separate prototypes: building a working prototype using discrete components and simulating a layout of an IC. (If there is a suitable opportunity to manufacture and test an IC, this may be attempted in the future, but it is a non-goal).

1.6 History of Analog Control and Computation

The current dominance of digital computer control may make it difficult to imagine that there was a time when the most sophisticated systems were controlled entirely by analog means. Indeed, it is the need for improving this control circuitry that motivated many well-known developments in electronics. The history of operational amplifiers, specifically, is covered extensively in [Sta66] and [Jun06]. An excellent introduction to the many-faceted history of analog computation is the special issue of Control Systems Magazine on the topic [CSM05].

This work will not devote more time to this history other than to draw attention to the fact that very complex systems were previously built using analog signal processing. One such example is the Nike defensive missile system which provided radar-guided intercept of fast-moving targets using analog, rather than digital, means. This system serves as historical proof that many requirements can be met with an analog approach when such an approach offers other benefits. In the case of Trust in Analog, the analog approach offers increased trust and decreased verification effort for circuits of a sensitive nature.

1.7 Rise of Digital Complexity

However, the recent trend in system design has been in the opposite direction: away from analog control and towards adding general-purpose digital computer function and software into the smallest and simplest subsystems. Several examples from major semiconductor firms' application notes illustrate the resulting increase in complexity in practice.

1.7.1 Maxim 16 bit RISC volume fader

Maxim Semiconductor application note AN4242, "Using the MAXQ3210 in an Audio Attenuator Circuit"[Max08], describes how to implement a volume sleep fader – an attenuator that slowly reduces audio volume to zero upon user request. The proposed circuit uses the company's 16 bit RISC microcontroller and digital potentiometer ICs to implement this basic capability. To quote, "low power 16-bit RISC microcontroller is the system controller that generates various timing intervals and drives the gradual attenuation of an audio signal ending with a muted condition." The abstract finishes with a cheerful but ominous note: "Source code for the firmware is available for download from the Maxim website." The RISC machine contains 2 kilobytes of program memory, 128 words of EEPROM, 64 bytes of RAM, 33 instructions, and numerous registers. While it is not known how many transistors are used to implement the MAXQ3210, a reasonable estimate would be 100,000 or more. With all of those features and a clock speed of up to 3.58 MHz, it is more powerful than the microprocessors in the Apple II or IBM PC, which launched the microcomputer revolution and powered the world's first video games, word processors and spreadsheets.

The digital potentiometer and opamp amp add several hundred additional transistors to the system. If volume control were a sensitive system from a security standpoint, this would mean many targets for both malicious software (since the general-purpose CPU can run any software) and malicious circuits (where a harmful block could be concealed among thousands of transistors). All of this complexity is many orders of magnitude higher than the simple fader function which it is meant to implement. While this design approach offers the benefits of (i) promoting the use of new products (ii) shielding the user from analog circuit details and (iii) easing future changes or extensions, these benefits come at the expense of a tremendous increase of complexity compared to a simple analog or even barebones digital implementation. In consumer electronics, this drawback may be regrettable on grounds of engineering taste, but it brings real risks in mission critical systems.

1.7.2 Analog Devices hard disk protection system

Analog Dialogue article 39-11, "Using Dual-Axis Accelerometers to Protect Hard Disk Drives" [LZ05] describes improved ways of detecting that a hard drive is falling so that the drive heads can be moved to a safe position before impact. The authors proposed an improved algorithm for detecting a fall, by comparing the sum of the squares of the time derivatives of acceleration to a threshold.

The circuit implementation (figure 1-1) uses an ADuC832 microconverter (comprising a mux, 12-bit ADC, a full 8052 microcontroller core and a UART) to implement a single-bit decision for whether the heads should be parked. Gate count for another company's 8051 core is between 4,000 and 14,500 NAND2 equivalent gates ([IPE10]), depending on configuration. Each NAND2 gate in CMOS is 4 transistors, so 16,000 transistors is the minimum for the microcontroller. Again assuming several hundred transistors for the amplifiers and the mux, plus 300 for the ADC, yields a total estimate of digital complexity of 16,500 transistors. All of these transistors are used to take the time-derivative of two values, square each of them, add them and compare them to a threshold! An alternative analog implementation with discrete components requires approximately 200 transistors.

The flexibility afforded by using standard parts and in-the-field programmability is valuable and perhaps essential in some applications. But the use of a general-purpose processor and associated

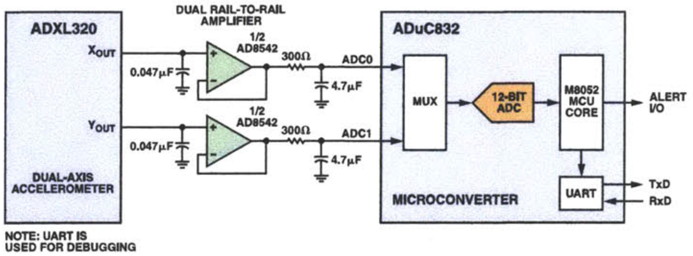


Figure 7. Simplified schematic of HDD-protection hardware system.

Figure 1-1: Analogue Dialogue 39-11

80-fold increase in complexity makes the circuit more susceptible to malicious changes in hardware or firmware, and more difficult to verify.

The preceding examples are from two leading semiconductor companies and, even allowing for the marketing nature of application notes, mirror the trend in production designs. General-purpose programmable digital computers are cheap, power-efficient and powerful. They are now used at every level of a complex system, from its top-level controller down to the lowest: controllers in volume faders, drive head parkers, motor drivers and power supplies. This rise of digital complexity brings, along with its benefits, new vulnerabilities.

Chapter 2

Malicious Circuits

2.1 Types of Attacks

There are several broad categories of potential problems with integrated circuits from untrusted sources, in order of increasing severity. First, a chip may be defective at the time of delivery, failing to meet one or more easily measurable specifications at the point of incoming inspection. This situation is identical to using components from a low-quality but non-malicious supplier, and is closely related to "counterfeiting" or "relabeling", where semiconductor components are fraudulently sold with claims exceeding their actual specifications, often by means of changing their physical labels.

Second, a chip may be marginal and either fail permanently at some later point or have intermittent failures, with the probability of failures potentially dependent on broad operating conditions such as utilization, temperature or power levels. This situation is more difficult to detect, causes greater impact in the field, and can offer the attacker some additional advantage in select cases. Still, it is similar to normal failures, including those resulting from "counterfeit" ICs. For example, a microprocessor that has been intentionally mislabeled for a higher operating frequency than the one it achieved at fab test will, when operated at this higher frequency, often dissipate more power and/or experience intermittent lockups or other failures.

Third, a compromised chip may contain circuitry specifically introduced to create a vulnerability (such as a "back door") which can be activated remotely. A typical means for doing so would involve transmitting a code to the target system that causes the malicious circuitry to alter the IC's behavior or to disable it entirely. Such a chip would perform within specifications until such time as it received the special signal, and thus is both most difficult to detect and offers the greatest advantage to a potential adversary – who can control the time and potentially the manner of its failure.

Examining this "type 3" class of attack more deeply, it becomes clear that unlike the first two, it cannot be accomplished without introducing malicious control circuitry and that it also requires a path for injecting malicious signals into the chip or subsystem. Reducing the risk of such an attack therefore involves detecting extraneous (or altered) circuitry and preventing malicious commands from reaching the chip. It can be shown that both techniques contribute equally to enhancing trust: a system with a backdoor that cannot be accessed is equivalent to a system without a backdoor to which the attacker has ready access.

Malicious circuitry may be used both to alter the chip's function on command and to deliver such a command to the chip's internals, but only one is necessary. In such a case, the intended chip function provides the other capability required for a successful attack, either the control circuitry or the command path.

The more flexible an electronic or computer system, the greater an adversary's ability to control its behavior. For example, it is easier to change the gain of a variable gain amplifier than that of one with a fixed gain. Easier in this context means fewer malicious circuit elements required, less time required for attack and less probability of failure to attack successfully. In the example amplifier, having benign circuitry which allows its gain to be controlled simplifies the attacker's task by obviating the need to add such circuitry. In general, a circuit's flexibility and complexity may be desirable for other reasons, but they increase security risks and make achieving trust more difficult.

2.2 Attacks Detail

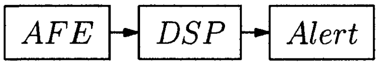


Figure 2-1: Malicious circuit in front end

We will now discuss several examples of malicious circuitry and resulting vulnerabilities. Figure 2-1 presents a data acquisition system which samples a broadband input signal, digitizes it, performs digital signal processing and produces an output. Imagine that the analog front end circuitry includes malicious circuitry, which, in response to a specific bit sequence at its input, turns itself off. If this system was part of a radar, an adversary would need only to send a specially crafted bit string to disable the radar. This is a type 3 attack and offers a readily obvious tactical advantage.

It is difficult to detect this kind of malicious circuitry by means of non-destructive, external ("black box") testing for several reasons. First, the number of required tests increases exponentially with the length of the bit string. For anything beyond short bit strings, the problem becomes computationally similar to a brute-force effort to recover a cryptographic key: 2^N permutations, with no a priori information about N, the length of the kill chip code. The problem is further complicated by the attacker's ability to break up the key or hide it using steganography in an input signal. The second reason is that there may be different paths for the kill-code into the compromised chip. In this system, instead of the direct RF path, the attacker may have access to other inputs that eventually transfer information to the targeted IC.

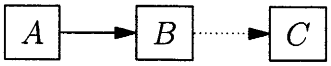


Figure 2-2: Indirect path for kill chip code

Figure 2-2 presents an example of such a system, where the kill code is received by a properly functioning block A. The signal then flows to another properly functioning, secure block B. In the course of its operation, block B produces input-dependent current consumption and thereby introduces power supply ripple into the system power bus. Both the two blocks' operation and resulting power supply ripple are within specification and not dangerous to the system's proper operation on their own. However, let us imagine now that subsystem C is compromised. While it may not have any direct connection to the outside, the malicious circuitry added to it can sense power supply ripple and after signal conditioning, extract the command signal.

Therefore, even ICs or system components which are on the inside of the system, without untrusted input flowing to them, must be secured, since paths other than the intended signal path may be used to deliver malicious orders to them. In addition to power supply rails, every connection between blocks is potentially a covert communication channel for the attacker monitoring buses such as I2C (Inter IC bus originally developed by Philips), one-way buses converted to bi-directional use and digital signals covertly carrying analog signals. Eliminating these connections should always be the first option, but often this is not possible.

In that case, it is helpful to view the problem of covert channels as a communication problem where the goal is to minimize, rather than maximize, the data rate available, $r = B * log_2(1+SNR)$. Reducing the effective error-free data rate may lengthen the amount of time required to transmit the kill code ($t = N/r$), may force the attackers to use shorter kill codes (easing detection of modified circuits via black-box tests) or more complex encoding/decoding means (easing detection of malicious circuitry due to its size and power consumption). For the case of power supply ripple, reducing the bandwidth is accomplished with additional filtering, while improving PSRR, increasing uncontrolled ripple sources and reducing signal-dependent injected ripple will all serve to reduce the effectiveness of the channel for transmitting the kill code. In the context of malicious circuits, PSRR becomes an important metric of system security. Notably, this analysis suggests that no aspect of the system should have more bandwidth or resolution than absolutely necessary to meet its specifications, since this extra inter-link capacity can be used by attackers.

In addition to wired connections between blocks, it is worth noting that electromagnetic waves transmitted over short distances may also be use be used to obtain secure system information or transmit commands to malicious circuitry. The advances in fully-integrated RF blocks make it possible to embed entire transceivers in chips manufactured in standard CMOS, opening new avenues for malicious circuitry. Imagine a version of the system in figure 2-2 that has no wired connections between the blocks A and B. A is connected to outside sources but does not perform sensitive functions. B performs sensitive functions but has a separate power supply and control buses. Low power RF signals transmitted by malicious circuits in A and B provide a covert channel. Also, more powerful RF signals transmitted by the attacker externally can be used to send a kill code to B even though it has no intended connections to the outside world. For RF attacks, broad-spectrum shielding even between blocks in the same system, as well as the outside world, is essential. Testing of chips for malicious circuits should also include monitoring of any unexpected electromagnetic radiation to ensure that no covert transmitter has been added.

The preceding discussion motives breaking up malicious circuits into two broad categories: the first is receiving the remote command such as a kill code and the second is acting on the kill code. Both of these functions are required for an effective attack against a system, but only one of the two functions has to be provided by surreptitiously inserted circuitry – either reception or activation may be provided by the intended functionality of the chip or subsystem. Whether the system is hardware or software, greater complexity makes it more difficult to detect back doors or intended capabilities that could be misused.

Minimizing complexity can be considered in the following way. If a system requires a number of states M, a number of combinatorial elements X and a data communication rate R to accomplish a task, any excess above M+X+R means that complexity is not minimal and risk of malicious exploitation is higher than the minimum achievable. Of course, the definition of tasks and computations is far more nuanced and open to interpretation than such a minimalist formulation would admit, but it provides a path to establishing some useful FOM (Figures-of-Merit) to compare different architectural approaches.

Much of the preceding analysis for maliciously introduced defects also applies to accidental defects. The accidental circuit defects may then be exploited by attackers who learn of them, without the need to insert malicious circuitry, or may be accidentally triggered during normal operation. Without engaging the broad topic of quality assurance and defect reduction, it is obvious that greater complexity unaccompanied by increased redundancy or error-correction leads to greater probability of defects.

2.3 Multiplication Example

Multiplication is one of the basic building blocks of any signal processing chain, whether used as a variable-gain amplifier, modulator or some other purpose. Its nature is not affected by whether it is implemented in a digital or analog circuitry. We will proceed to examine two possible implementations of a low-precision multiplier from the perspective of resisting malicious circuitry: one built from digital logic gates, and one built with operational amplifiers and transistors.

2.3.1 Quarter Square Multiplier

Although Gilbert cell multipliers have come to dominate multipliers in both bipolar and FET technology, there are other topologies for analog multiplication [KK72]. One of these is the quarter-square (or difference-of-squares) multiplier (see [KK72], p. 214), which utilizes the square-law regime and operates as follows. First, the sum and the difference of the two signals to be multiplied (X and Y) are computed. Second, the two results are squared using devices operating in a square-law regime (in this case, MOSFETs). Third, the resulting squared values ($X^2+2XY+Y^2$ and $X^2-2XY+Y^2$) are subtracted, which eliminates the X^2 and Y^2 terms, leaving the desired cross-multiplied term $4XY$.

In this project, the quarter square multiplier was initially pursued for its advantages when using low supply voltages with discrete transistors, where their high and ill-specified threshold voltages make the tall stacks of transistors of other topologies run out of head room or require folding. As the discrete phase of the design progressed, the quarter square multiplier implementation became a larger portion of the project and an area of deeper investigation in its own right, as described in a subsequent chapter. Among its disadvantages is the requirement for good matching between the added and subtracted paths including the amplifiers, the transistors and the resistors since any offsets will result in incomplete cancellation of the undesirable terms or propagation of extraneous errors to the output. And because of the use of standard opamps, it adds more complexity than a custom implementation or a Gilbert cell multiplier.

The multiplier was implemented using discrete components and designed to operate from a modern supply voltage of 3.3 V, unlike classic 15 or 30 V designs. The circuit and its performance is described fully in chapter 3, specifically the final circuit diagram is on figure 3-1, and the PCB layout can be seen on figure 3-2. It turns out that additional steps beyond the above theoretical principle are required. But for the purposes of this example, it is only necessary to note that the analog multiplier uses one and a half quad opamp IC's and five transistors, along with a handful of passive components.

2.3.2 Comparing Digital and Analog Complexity

In order to compare the complexity of the digital and analog implementations, an estimate is made of the precision of the analog multiplier. Slightly better than 1 percent accuracy translates into 7 bits of digital precision. A straightforward digital implementation requires $(N-2)(N-1)$ full adder or half-adder cells, with each adder comprising six gates: two XOR gates, three NAND gates and

multiplier type	**transistors**	**components**
digital (7b)	1500	1500
digital (8b)	2100	2100
digital (9b)	2800	2800
analog (discrete)	305	350
analog (IC)	11	11

Table 2.1: Digital and analog complexity comparison

one 3-way NAND gate[OVL96]. We take each XOR gate as equivalent to four NAND gates, and the 3-way NAND as six, instead of the usual four, transistors. The resulting transistor count is then: $(7-2)*(7-1)*((2*4*4)+(3*4)+6) = 1500$. This estimate reflects only the multiplication function, without memory and without A/D or D/A conversion, which, while also complex, would probably be amortized across many different blocks in a digital system and thus should not be allocated entirely to the multiplier. On the other hand, the analog multiplier described requires 5 discrete transistors and a pessimistic estimate of 50 transistors for each of the opamps used, for a total of 305 transistors. Therefore, for similar performance specifications, the digital multiplier requires about five times more transistors (1500 vs 300) than the analog multiplier. It is also possible to estimate and compare the number of states in the two multipliers.

The complexity of the digital multiplier increases rapidly with word length. If the same analog multiplier can deliver accuracy equivalent to 8 or 9 bits, the digital circuit will now require 2100 or 2800 transistors to keep up. Furthermore, an integrated circuit implementation of a multiplier can require far fewer transistors than a discrete implementation – on the order of dozens (chapter4 describes a possible implementation of 11 transistors).

Verifying an integrated circuit implementation of an analog multiplier block is commensurately easier than doing the same for the digital block. First, destructive verification via examination of the chip requires examining five times fewer components. Second, external, non-destructive testing is eased by reducing the number of signal paths into the device and the number of accessible internal states. The multiplier is used primarily as a variable gain control block in the user-configurable transfer function, where it controls both loop gain and the location of system poles. The state space filter approach allows the location of poles and zeros to be controlled by setting gains in the feed-back and feed-forward paths, as shown in the classic figure 2-3 [KK72]. This topology is due to analog computers of yesteryear and provides excellent flexibility. An important goal of the project is to make the analog controller as easy to use as the digital one, providing easy gain-setting and compensator configuration.

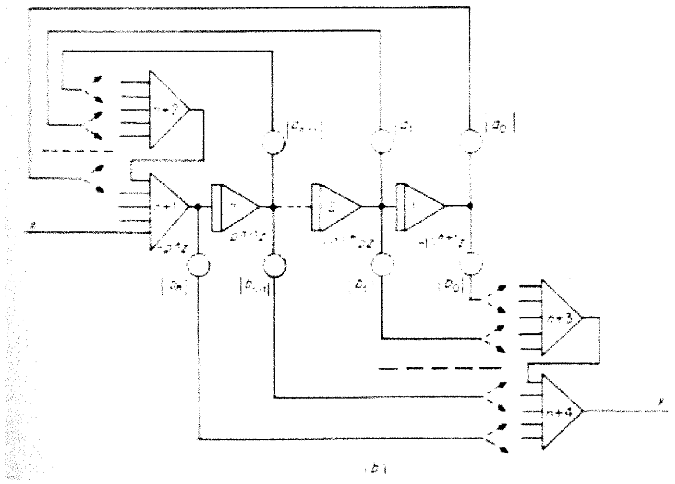


Figure 2-3: Transfer operator setups (figure from Korn & Korn)

Chapter 3

Discrete Design

3.1 Overview

A significant part of this project is the design and construction of a simple analog controller using discrete components. Modern parts and supply voltages were used to demonstrate that analog control can be successfully employed in a modern system environment. (For the prototype, the supply voltage is 3.3V total with +/-1.65VDC dual-sided supplies). The basic blocks include a VGA/multiplier, operational amplifiers and analog switches. These can be interconnected to form larger blocks, including filters with configurable transfer functions. Several versions were developed, a representative circuit diagram can be seen in figure 3-9.

Quarter Square Multiplier

The principle of operation of the multiplier is summarized in chapter 2. It uses AD8544 operational amplifiers and BSS138 MOSFET transistors and operates from a modern supply voltage of 3.3 V, distinguishing it from 30 V designs found in the classical literature. The initial circuit diagram is on figure 3-1, and the PCB layout can be see on figure 3-2. The final circuit diagram is figure 3-5. Its performance is summarized in table 3.1.

It is essential that the transistors operate in strong inversion rather than subthreshold, so that they follow the square-law and not the exponential-law model. The bias current must be sufficiently high so that maximum excursion in the negative direction does not approach weak inversion. Since threshold voltages of discrete MOSFETs vary widely, the proper biasing scheme is to control current density rather than merely setting a voltage.

It turns out that transistor behavior is also the major source of error in a simple implementation like the one above. The transfer characteristic of a MOSFET is not $I_{ds} = V_{gs}^2$ but $I_{ds} = \alpha(V_{gs} - V_{th})^2$, where α is a constant and effect of V_{DS} is ignored. At first glance, it would seem that merely

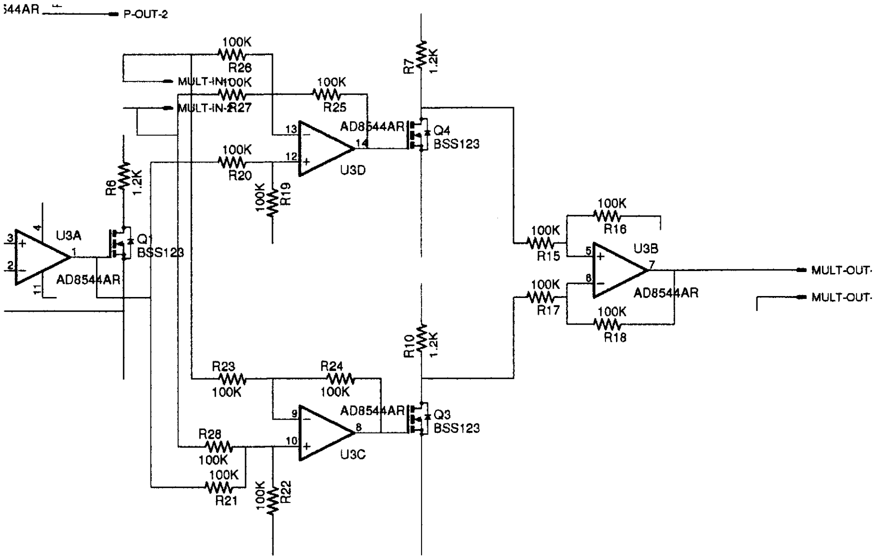


Figure 3-1: Basic quarter-square multiplier schematic

offsetting the input by exactly V_{TN} would be sufficient, but this is not practical for two reasons. First, it violates the earlier point, as the square-law model is increasingly less accurate closer to the threshold voltage. Second, the threshold voltage is not readily available.

Let us analyze the error of the quarter-square multiplier resulting from MOSFET transfer characteristics. We calculate the current output, assuming that the input is the signal voltage plus threshold voltage and some offset.

$$M = (B - V_{TN})^2 = B^2 - 2BV_{TN} + V_{TN}^2$$

$$whereB = V_{in} + V_{TN} + V_x$$

$$M = (V_{in} + V_{TN} + V_x - V_{TN})^2 - 2(V_{in} + V_{TN} + V_x - V_{TN})V_{TN} + V_{TN}^2$$

$$= V_{in}^2 + 2V_xV_{in} + V_x^2$$

If the transistor was ideal and the input voltage was offset by exactly the threshold voltage, V_x would be equal zero and $M = V_{in}^2$, the ideal case. Instead, this error has the following impact on the multiplier, if $B = (X + Y)$ and $B = (X - Y)$ are substituted into the above equation and the results subtracted from each other:

$$(X + Y)^2 + 2V_x(X + Y) + V_x^2$$

$$-$$

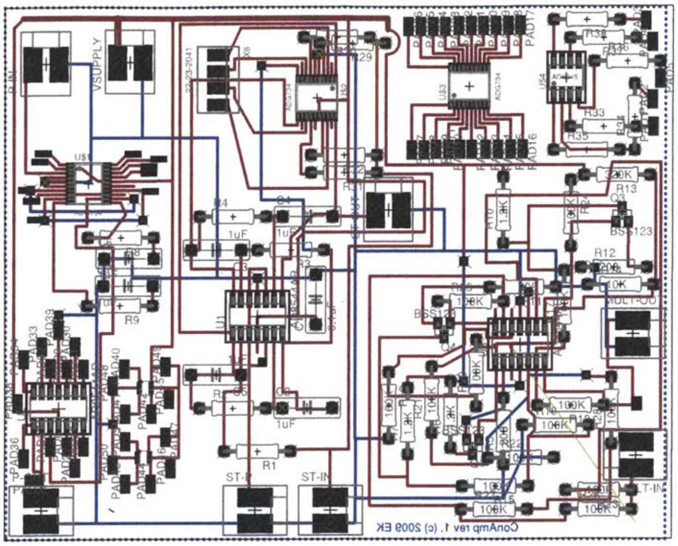


Figure 3-2: Quarter-square multiplier (version 1) PCB

$$
\begin{aligned}
(X-Y)^2 + 2V_x(X-Y) + V_x^2 \\
= \\
4XY + 4V_xY
\end{aligned}
$$

Some of the error terms are successfully cancelled, but because the sign of Y differs, the term linear in Y does not disappear. This is most easily (and distressingly) observed by setting X input to zero and watching the output vary with Y. A tell-tale sign is a transfer characteristic (as observed in laboratory testing but illustrated best by the simulation result in figure 3-3) which fails to pass through the origin. Notably, multiplication by zero proceeds properly if Y is set to zero instead, so not only is it incorrect, but also non-commutative. The error is proportional to V_x and a first-order

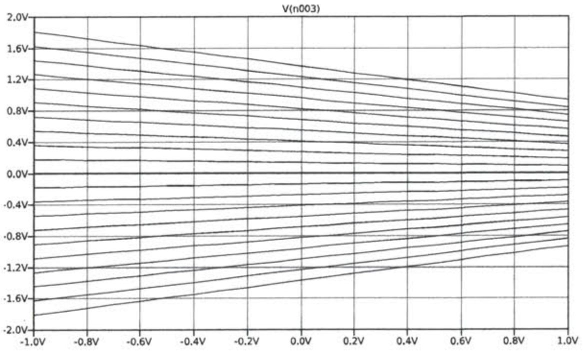


Figure 3-3: Spice simulation of multiplier DC transfer characteristic. (X axis: X, sweep lines: Y from -1V to 1V)

correction would be to operate close to the threshold voltage and with large inputs. This approach is more difficult with low power supply voltages, so a better solution is to cancel the error term. By squaring Y and $-Y$ and subtracting the results, one obtains just the $4V_xY$ term, which can then be subtracted from the multiplier output. As long as the V_x is sufficiently large to keep V_{gs} above V_{TN} for all signal inputs, its precise value is no longer important. After some effort at combining stages, it is possible to add this correction to the multiplier without too many additional parts or complexity, as can be seen in figure 3-5. The transfer characteristic (figure 3-4) now looks proper. There are some additional sources of error in such a multiplier: i. mismatches in the two critical transistors, especially mismatches of threshold voltage ii. voltage offsets in the operational amplifiers, especially 'D' and 'C', the two driving the main transistors iii. resistor mismatch in opamp gain-setting, summing or drain resistors iv. noise from any of the components. For comparison, the classic literature (Korn) reports 0.2 or 0.5 percent accuracy as being achievable for this kind of multiplier implemented using older technologies.

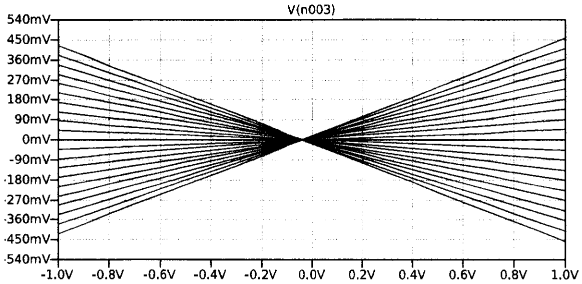


Figure 3-4: Multiplier DC transfer characteristic with correction. (X axis: X, sweep lines: Y from -1V to 1V)

Amplifiers

The discrete design uses general purpose AD8544 quad operational amplifiers manufactured by Analog Devices, specified for rail-to-rail operation from 2.7 to 5.5 volt supplies. These CMOS opamps are far from high performance devices: while they offer low supply current (45uA), input offset ranges from 1 to 6 mV, gain bandwidth product is 980 kHz and large signal gain ranges from 100,000 to 500,000 [Dev08]. The selection of these parts is not entirely accidental, since it helps to demonstrate that even low-voltage CMOS opamps are suitable for analog control uses. While these introduce limitations in achievable system bandwidths and pole/zero locations, the topologies discussed here are equally applicable to better components.

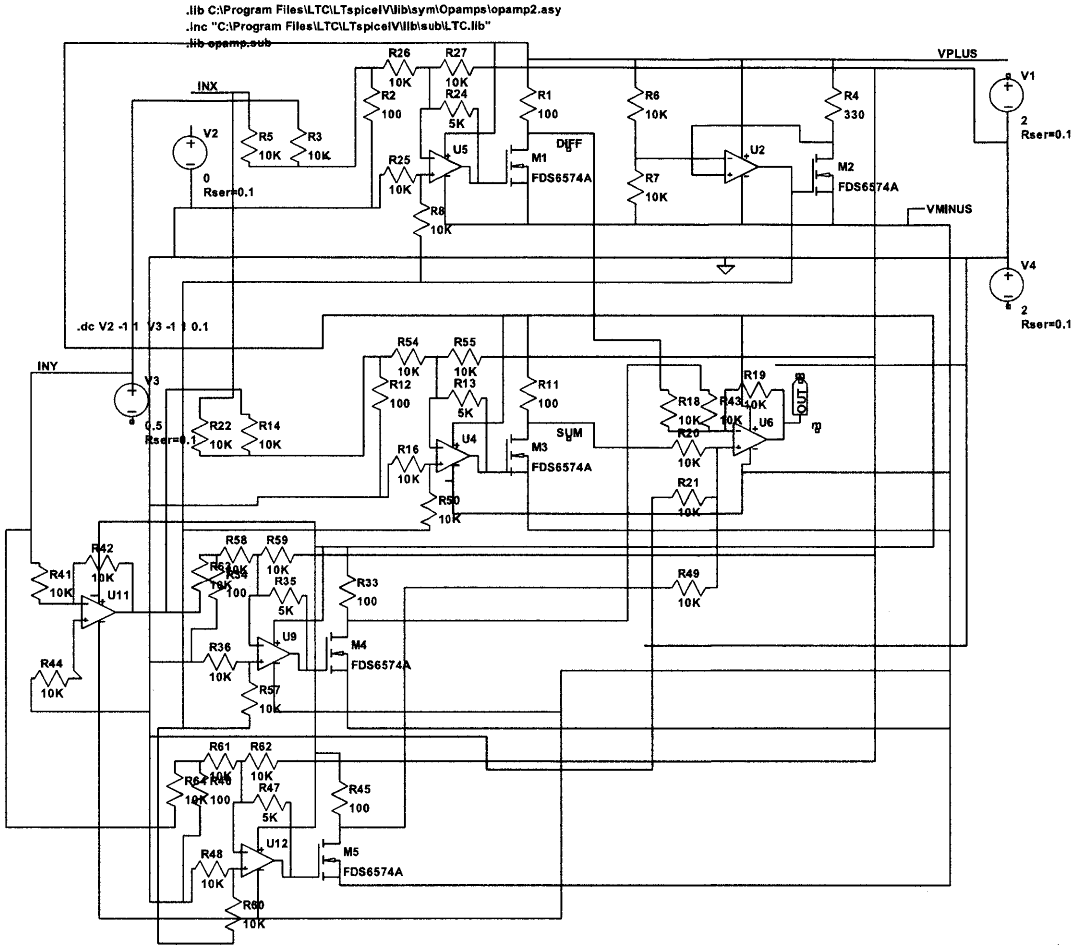


Figure 3-5: Squares multiplier version 2

3.1.1 Construction

The discrete component design was prototyped on a custom printed circuit board. The first revision two-layer 3-by-4 inch PCB was designed using Eagle CAD software and manufactured by Advanced Circuits. The second revision 4-by-6 PCB was designed using Eagle, as well. Surface mount components were placed manually on pads coated with solder paste, and entire board heated in an oven until the solder melted. After rework of any solder bridges, thru-hole components were added. A photo of one of the completed boards can be seen in figure 3-7. The second revision of the PCB includes the additional multiplier correction circuitry, omits some of the other experimental components, and features an improved PCB layout. By standardizing the layout of components around the opamps, even if some are unused, it is possible to have a much more regular and easy to follow layout, with ample room for troubleshooting-induced modification or experimentation.

During assembly and test, minor errors were found in the printed circuit boards. For the revision

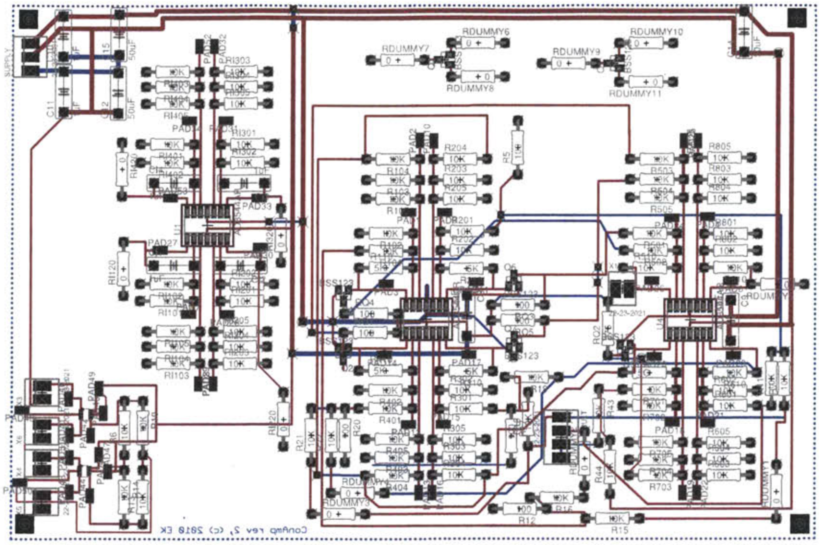


Figure 3-6: Squares multiplier PCB (rev 2)

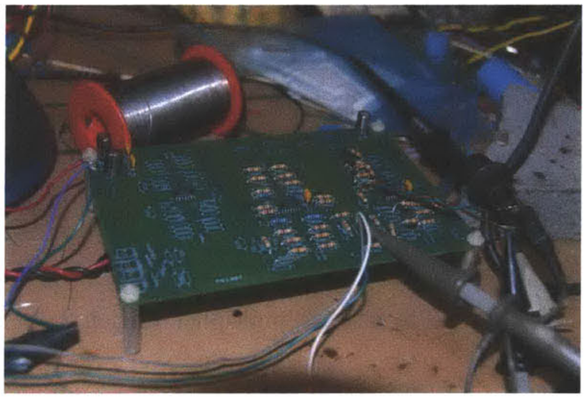


Figure 3-7: Photograph of one of the prototypes

2 board, the errata are as follows: in PCB layout, missing connection between Q4 and RQ4 (easily bridged with a wire), in both schematic and PCB, R43 should be connected to VMINUS, not ground, to allow for full range of adjustment of bias point (omitting that resistor is sufficient), connection to R502 and R503 should be swapped (accomplished by lifting the two resistors and running small rework wires from the respective pads). The following jumpers should be installed: Rdummy2, Rdummy4, R701. In practice, R703 is 100K and R705 10K, reducing the gain around the bias-setting loop. Depending on characteristics of the transistors used and desired signal voltage ranges, the bottom legs of the input attenuators (resistors R5, R12, R17 and R20) can be increased or omitted.

Y =	**-0.5**	**-0.4**	**-0.3**	**-0.2**	**-0.1**	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
X:											
-1.57	-1.7	-1.50	-1.15	-0.85	-0.50	-0.20	0.10	0.45	0.75	1.10	1.75
-1.25	-1.4	-1.20	-1.0	-0.75	-0.50	-0.20	0.10	0.40	0.70	0.95	1.15
-1	-1.0	-0.90	-0.75	-0.55	-0.35	-0.15	0.10	0.30	0.50	0.65	0.80
-0.75	-0.72	-0.60	-0.40	-0.26	-0.14	-0.04	0.04	0.10	0.22	0.36	0.50
-0.50	-0.50	-0.34	-0.20	-0.10	-0.06	-0.02	0	0.04	0.08	0.18	0.28
-0.25	-0.36	-0.28	-0.20	-0.14	-0.10	-0.04	0	0.06	0.12	0.18	0.26
-0.12	-0.38	-0.32	-0.25	-0.20	-0.12	-0.06	0.02	0.08	0.16	0.24	0.30
0	-0.45	-0.40	-0.32	-0.24	-0.10	-0.07	0.02	0.10	0.20	0.30	0.38
0.125	-0.28	-0.25	-0.20	-0.15	-0.10	-0.05	0	0.05	0.11	0.16	0.22
0.25	-0.64	-0.56	-0.44	-0.36	-0.22	-0.12	0	0.12	0.26	0.40	0.56
0.5	-0.80	-0.72	-0.60	-0.44	-0.30	-0.12	0	0.16	0.32	0.50	0.64
0.75	-1.10	-0.90	-0.75	-0.60	-0.40	-0.20	0	0.20	0.40	0.60	0.75
1.0	-1.25	-1.10	-0.85	-0.70	-0.45	-0.20	0	0.20	0.50	0.70	0.90
1.25	-1.40	-1.25	-1.00	-0.75	-0.50	-0.25	0	0.25	0.50	0.75	1.00
1.5	-1.50	-1.30	-1.10	-0.80	-0.50	-0.25	0	0.30	0.60	0.90	1.15

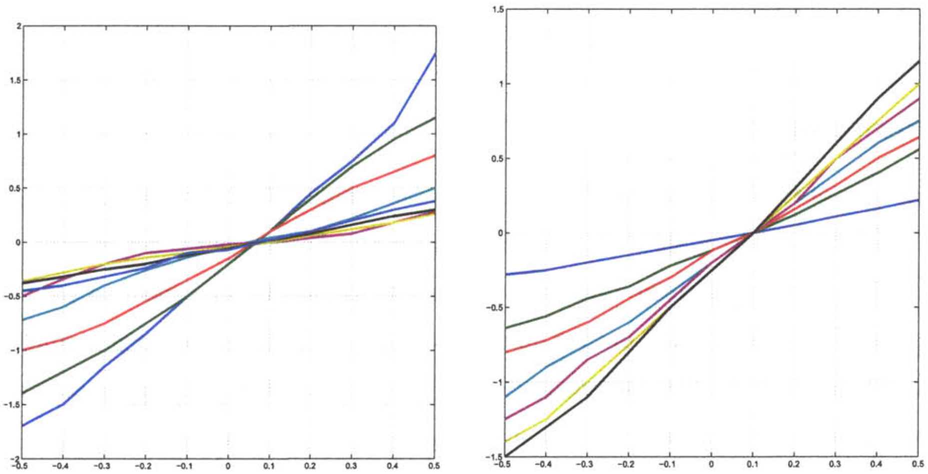
Table 3.1: Multiplier output

3.1.2 Performance

The performance of the final version of the quarter square multiplier is presented below. It achieves the goal of operating at low supply voltages over a fairly broad range, but is not entirely satisfactory. The cancellation scheme described above does make a significant improvement to the performance; without it, the offset errors at the output quickly overwhelm the dynamic range available.

In table 3.1, a selection of measurement results is presented. The signal generator was setup to provide 1 volt peak-to-peak near-saw-tooth output to the Y input of the multiplier. The X input was varied using a potentiometer, the voltage value measured and data points taken from the oscilloscope trace of the output. At least some of the distortions are due to inputs at the extremes of the power supply rails, with both opamps and transistors operating outside the expected regime. There are clearly some DC offsets at both inputs and the output, but the correction scheme clearly provides the first-order correction intended despite the wide input ranges. Behavior is better with smaller inputs. The same data is graphed in figure 3-8, a plot similar to the simulation results seen earlier.

This performance does not match the 0.2 to 0.5 percent accuracy suggested by classical literature, but it is not entirely clear whether this is due to components selected (for example, 5 percent tolerance resistors were used in the prototype, and any mismatch in gain between the different signal paths would create commensurate error at the output; the particular opamp has significant offsets whose effects would be greatly increased by the MOSFET square-law action) or fundamental limitations of this topology operating at lower supply voltages.



(a) X=-1.57V to 0.8mV

(b) x=0.125 to 1.5V

Figure 3-8: Graphs of multiplier output

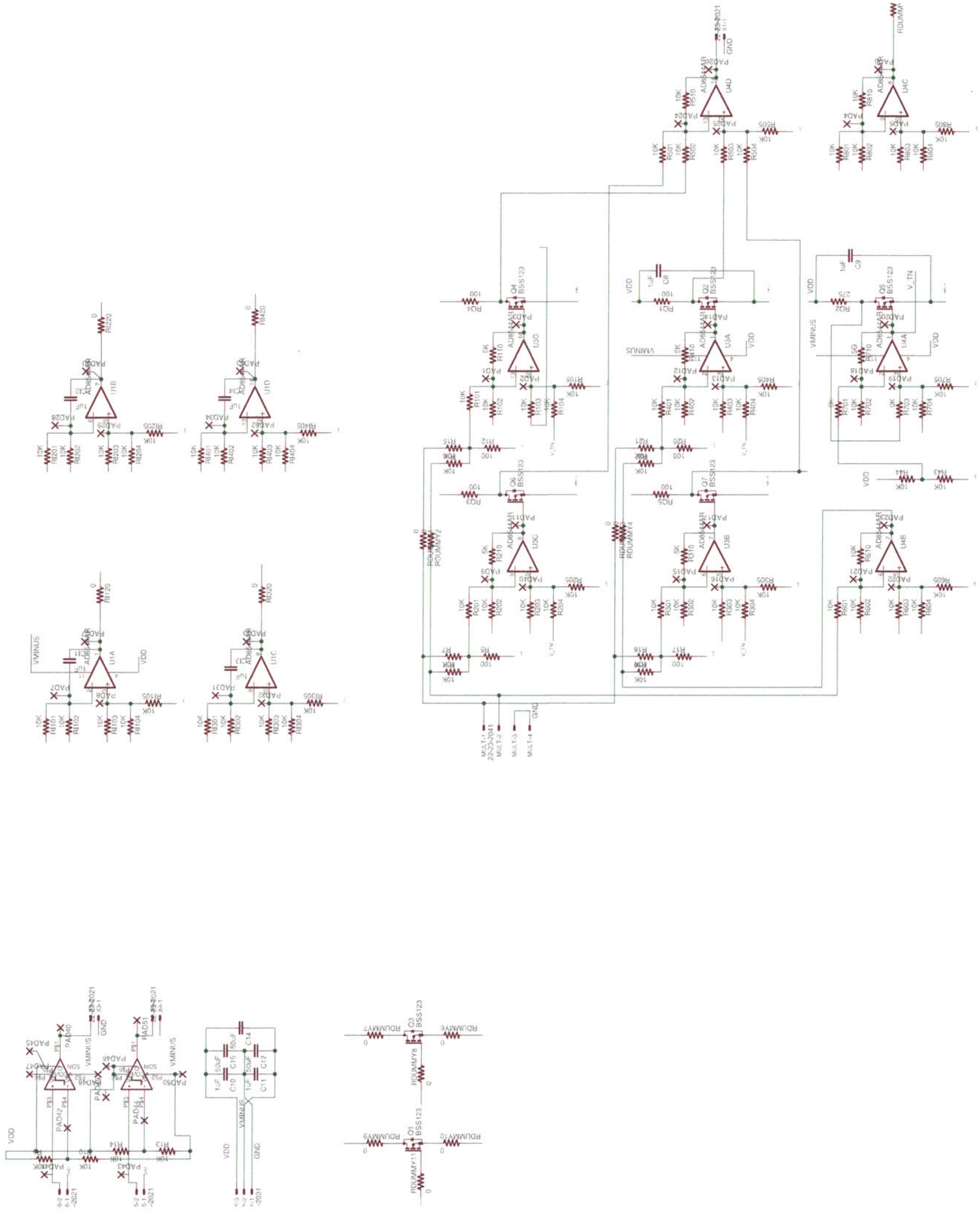


Figure 3-9: Discrete (rev 2) schematic (Eagle)

Chapter 4

IC Design

4.1 Overview

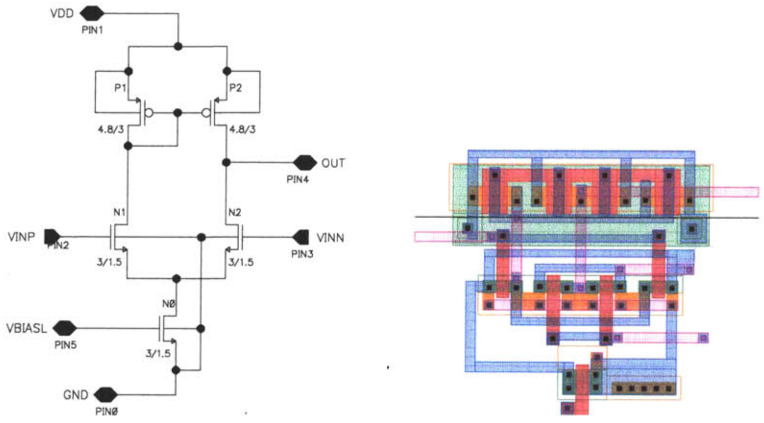
A significant part of this project is the design and construction of a simple analog controller as an integrated circuit. Industry-standard CAD software (Cadence), common supply voltages and well-known fab process were used to demonstrate that analog control can be successfully employed in a modern IC environment. (For the prototype, the supply voltage is 3.3 V and process is CMOS AMI 0.5 um). It should be noted that minimizing component count is a key objective of this design, so simpler topologies were employed whenever possible. The different circuit building blocks and their simulation results will be described below. Simulation was performed using Cadence software, primarily with hspice analog simulator. The smallest building blocks were validated individually, followed by the larger system.

4.2 Blocks and Simulation Results

4.2.1 OTA

Operational transconductance amplifier (OTA) is a simple but important building block of the chip. Its schematic can be seen in figure 4-1a and its layout in figure 4-1b below.

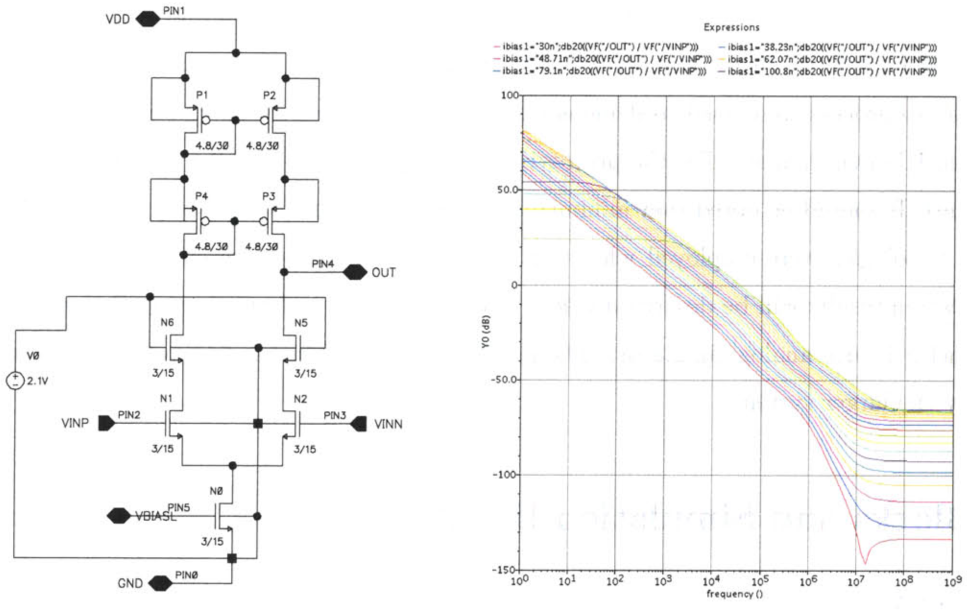
While simple, the transconductor design may be limiting for some applications due to the low output impedance of the MOSFETs. For example, if used as a Gm-C integrator, instead of the ideal transfer function is $H(s) = G_m/sC$, the result will instead be $H(s) = G_m/(sC + 1/R)$. In this design, C of 10 picofarads is used, and output impedance makes a difference. At the expense of additional complexity (4 additional transistors and longer-channel devices), the OTA4 design (figure 4-2) improves the output impedance considerably. The resulting AC transfer characteristic is quite regular for a broad range of bias currents, and shows only minor aberrations. As with the other



(a) Schematic (b) Layout

Figure 4-1: OTA1

circuit blocks, the OTA is affected by feed-forward at higher frequencies, a topic discussed in more detail in connection with the current amplifier.

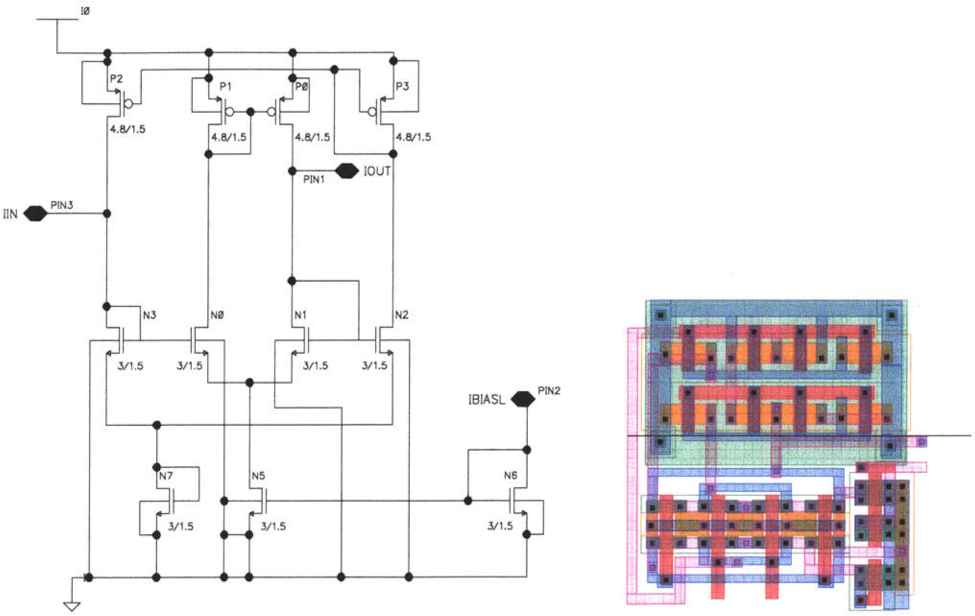


(a) Schematic (b) AC response

Figure 4-2: OTA4, transconductor with cascoding and longer-channel devices

4.2.2 Current amplifier - Multiplier

The current amplifier (or multiplier) is a key component of the system, since it provides the feedback and feedfoward controllable gains needed for hsblock operation. The multiplier is a simple translinear multiplier shown in figure 4-3. This is one of several designs, and is distinguished by its single-ended (rather than differential) operation and relative simplicity. Current output at IIOUT is dependent on the product of current input IIN and IBIASL. The implementation uses only 11 transistors.

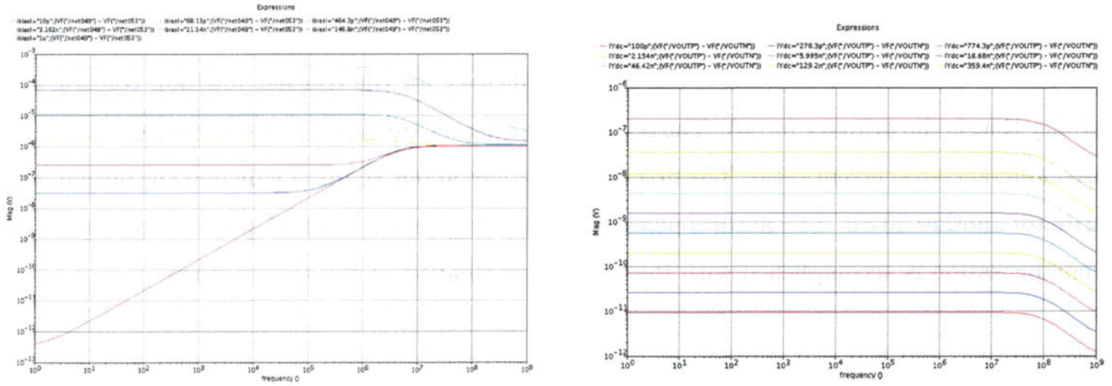


(a) Schematic

(b) Layout

Figure 4-3: Current amplifier (version 5).

A careful observer will notice the rise in the frequency response at lower bias currents for the single-ended design (figure 4-4a). This rise is caused by a zero, which moves to progressively lower frequencies with lower currents. It was eventually confirmed that the zero is indeed caused by the feedforward of C_{gd} into r_{ds}, the transistor output resistance which is, of course, increased proportionately with decreasing bias current. In short, the transistor and the multiplier is difficult to turn off at these higher frequencies, even with very little current flowing through it. The simplest solution is to restrict the range of bias currents and operating frequencies in such a way as to keep the zero well above the maximum frequency. This was the approach used in the initial revision of the design, and it is practical because the problem is exaggerated for illustration purposes in this figure by sweeping down to very low current values (such as 464 pA or even 10 pA).



(a) Single ended

(b) Improved differential

Figure 4-4: Current amplifier frequency sweep

A more complicated solution is to switch to a true differential multiplier design. This was done and simulated, as well, and can be seen in figure 4-5. While it adds some complexity (of a couple more transistors in this and other stages, as well as differential wiring throughout), the frequency response (figure 4-4) is now free of the undesirable zero even at lower currents. The added benefit is that the pole is also higher, above 20 MHz instead of 2 MHz. This increased bandwidth largely removes the dynamics of the multiplier from the hsblock, widening the usable dynamic range of the system. In both cases, the old adage of keeping plenty of current flowing through MOSFETs in general and Gilbert cells specifically is clearly confirmed.

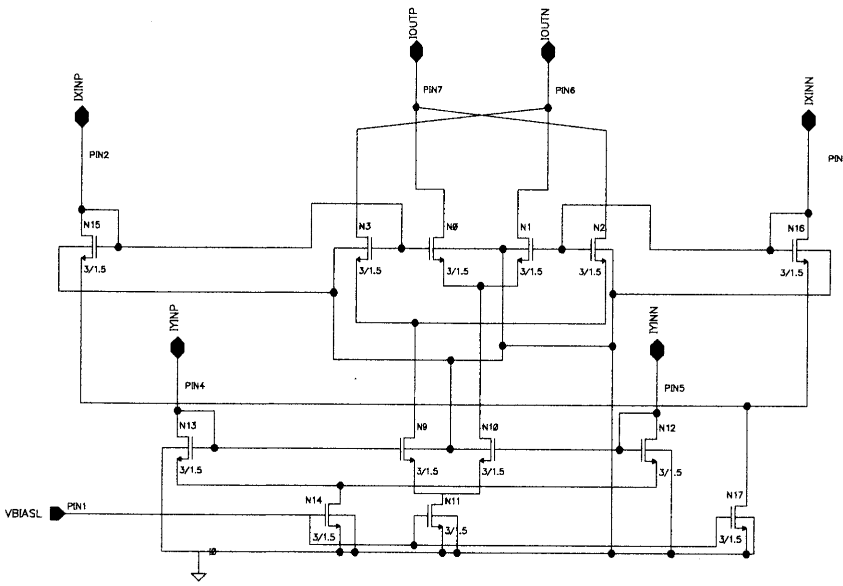


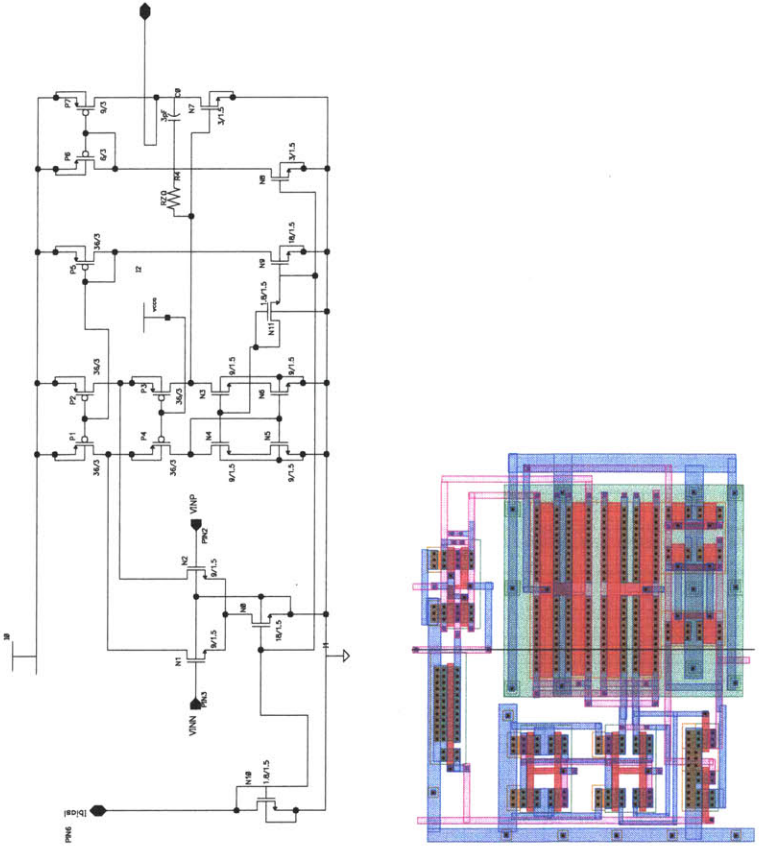
Figure 4-5: Differential current amplifier

Operational amplifier

The operational amplifier is unremarkable. It employs a traditional folded-cascode topology and finds uses in several portions of the chip. (However, whenever possible, special amplifiers are used instead of this general purpose amplifier, so it has little significance for the transfer function configuration capability that is the primary focus). The opamp's transistors are biased in weak inversion for better transconductance-to-current ratio. The first stage provides about 80 dB of gain and the output stage contributes another 40 dB. The natural poles of this topology in the AMI 0.5 process are unpleasant, and compensation is accomplished with the Cp-Rz combination at the second stage.

4.2.3 Pulse width modulator

Many practical applications of control systems are not based on voltages as the control output. Instead, controlling a plant such as a switching power converter or a motor, involves producing a



(a) Schematic (b) Layout (possibly out of date)

Figure 4-6: Opamp

switching waveform where its duty cycle, relative phase or frequency control the behavior. In keeping with the original project plans, a simple pulse-width-modulator was designed. This topology still requires some improvements (for example, its 100pF capacitor would consume too much chip area and some conversion and biasing details are omitted). However, it answers the question of interest: requiring only about 30 transistors for a final implementation, it offers much lower complexity than a general-purpose digital controller.

The circuit (seen in figure 4-7) operates as follows: (i) an input current (modeled by I4) is switched with a dual-sided mirror to the node net11 (ii) the sum of I5, a constant bias, and the mirrored input current (which may be negative or positive, depending on the value of V2 at the time) charges or discharges capacitor C0 (iii) N2-N3, N7-N6 and NOR gates acts as comparators; they compare the capacitor voltage to a threshold, and switch V2, changing whether the input current is added or removed from the bias. A simulation run showing the change in duty cycle can be seen in figure 4-8. (Note that the change in frequency is not an artifact, but matches the theory; if found undesirable, a traditional constant ramp and comparator topology could be utilized instead).

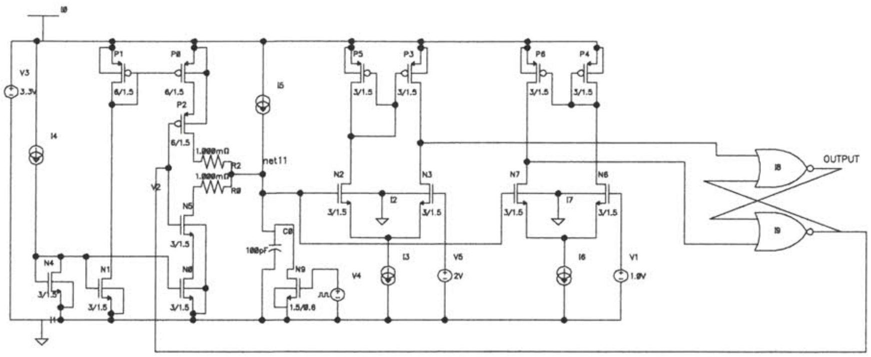


Figure 4-7: Example PWM schematic

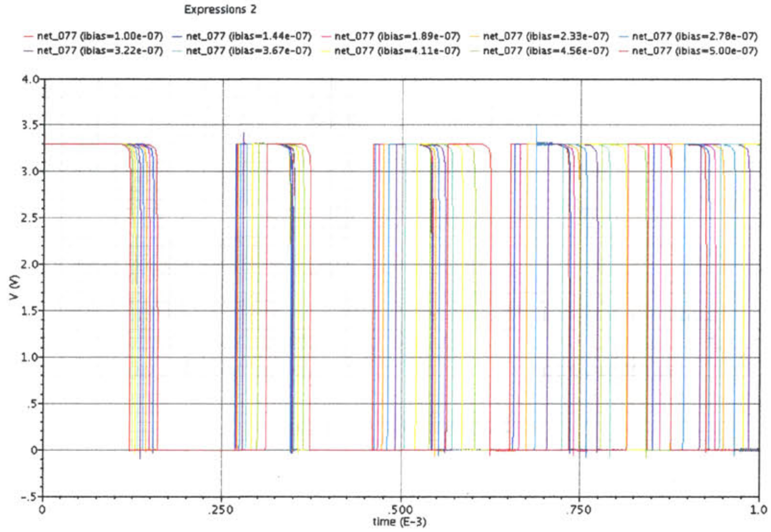


Figure 4-8: PWM block simulation

4.2.4 State space filter

The hs2 block is the core of the chip and provides the user-configurable transfer function. Built out of multipliers, transconductance elements and a few passives, its circuit diagram can be seen in figure 4-9 and layout – in figure 4-10. Each of its four stages is an integrator consisting of an OTA and a capacitor, with another OTA feeding current back to the input and the output via two separate multipliers. By using currents instead of voltages, summing is easily accomplished by simply tying the nodes together. The configuration of the block is accomplished via bias currents fed into the current amplifiers and the OTA blocks (in simulations, these are named ifaN for left side (feedback terms), ifbN for right side (feedforward terms) and ibiasN for the remaining bias currents.

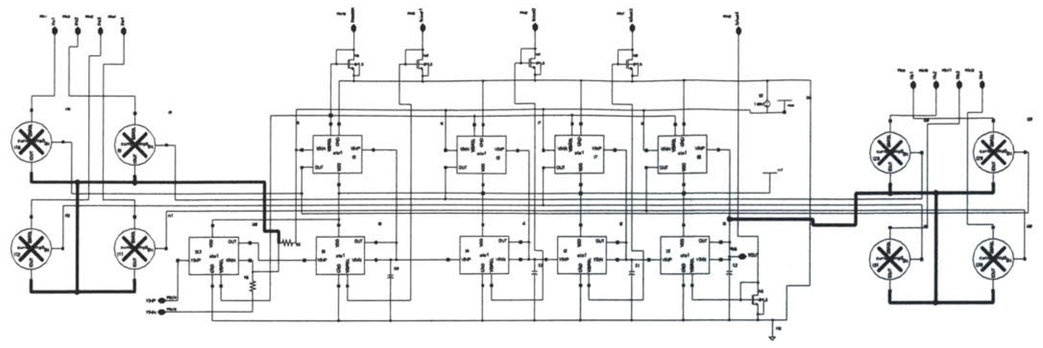


Figure 4-9: Arbitrary transfer function block (hs2) schematic.

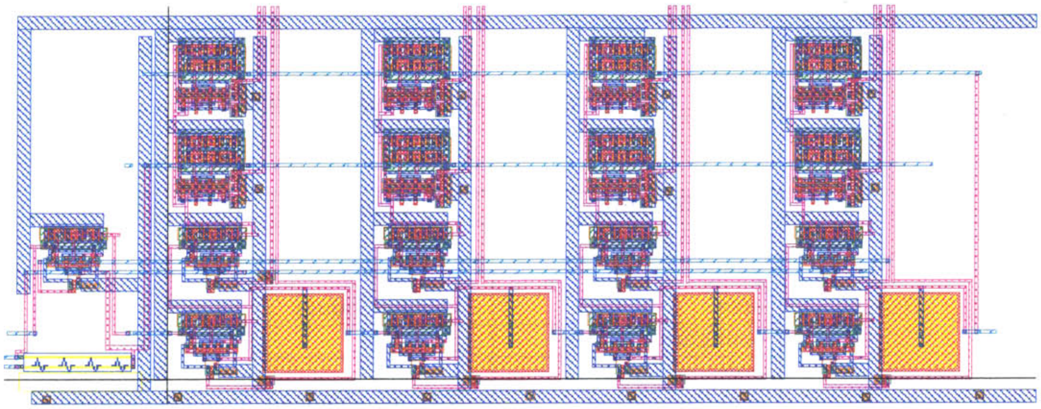


Figure 4-10: Arbitrary transfer function block (hs2) layout.

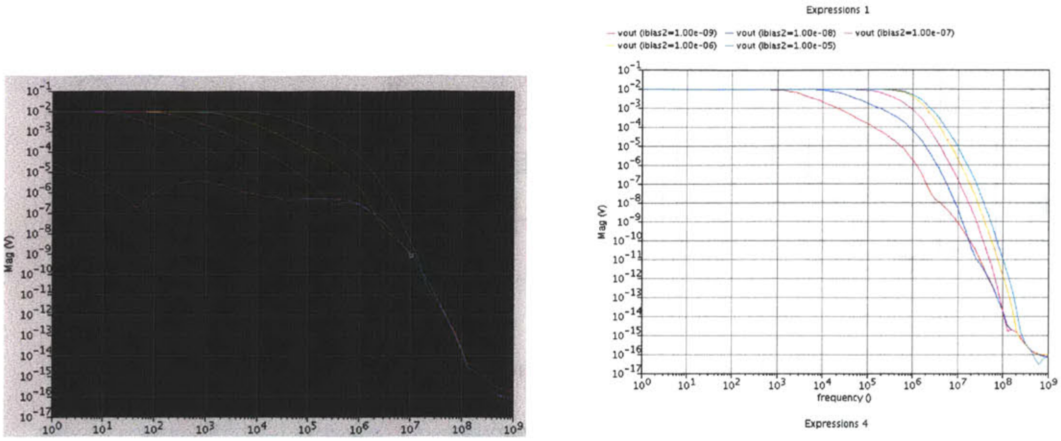
4.2.5 Whole chip

The whole analog controller chip is built around the transfer function block, input and output amplifiers, and auxiliary circuitry. It is arranged similar to the conceptual block diagram, and its schematic can be seen in figure 4-12. Table 4.1 summarizes the key specifications of the design.

4.3 Discussion

4.3.1 Tools used

This project used the following software tools. For IC design: Cadence version 5.10.41USR6.127.29, with Virtuouso for schematic capture, Virtuoso for layout, Analog Design Environment and hspice for simulation. For discrete design: Eagle PCB versions 5.5 to 5.10 for schematic capture and printed circuit board design, LTSpice for schematic capture and simulation. Some of the hardware tools included: Tektronix 2465 and 7854 analog oscilloscopes, Hewlett Packard 3435A and 34401A multimeters, Wavetek 288 signal generator.



(a) 1Hz to 10kHz

(b) 1kHz to 100kHz

Figure 4-11: Changing pole location using hs2 block

Specification	Value
technology	AMI 0.6um
power	106uW
area	185umx480um
configurable filter	yes
poles or zeros	4
pole/zero range	10 Hz to 100kHz
no external capacitors	yes

Table 4.1: Chip summary

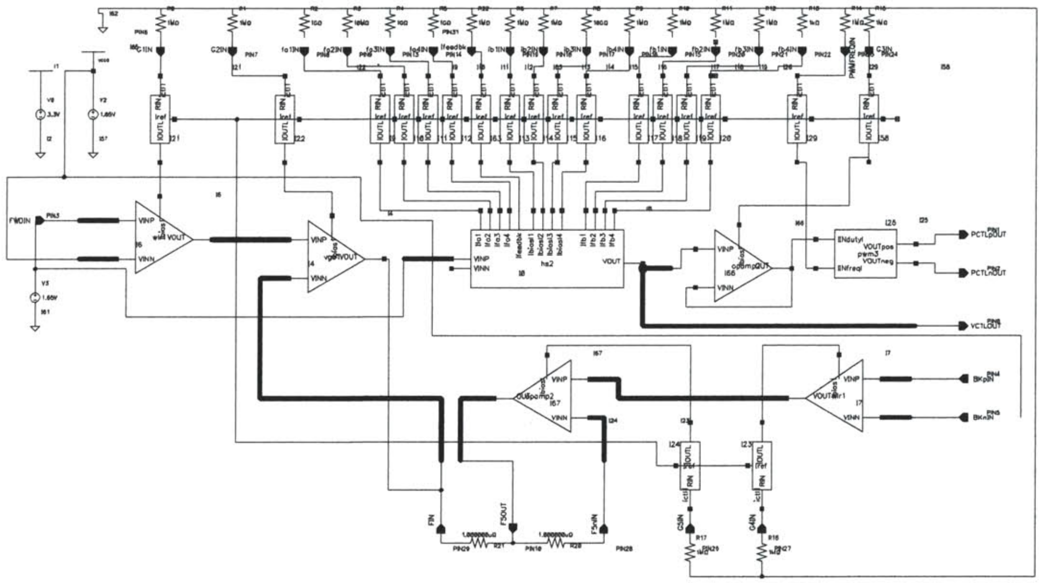


Figure 4-12: Schematic of entire chip, sans pads

Chapter 5

Conclusion

5.0.2 Completed work

The work completed as part of this project can be grouped into three broad categories. The first is initial analysis of possible malicious circuit techniques and ways in which limiting complexity by the use of analog computation can reduce the verification requirements and improve the trustworthiness of mission-critical systems. The second is the design of a broad architecture for a general-purpose analog controller, inspired by classical control and analog computation literature. The third is the practical implementation of certain components of such a controller in both discrete component and integrated circuit versions. For the discrete design this led to a lot of detailed work on the quarter-square multiplier. The IC design involved the investigation and solution of the feed-through problem and other departures from idealized transistor behavior.

Based on the completed work, it can be said that (i) analog circuit techniques can reduce the complexity, as measured by transistor count, of selected control circuits by 100x or more compared to general-purpose digital controllers (ii) classical analog techniques such as state-space filters and quarter-square multipliers can still be used with lower (sub-5V) supply voltages common in today's systems (iii) analog control should be considered for its possible trust and security benefits even if digital or software controllers offer other advantages.

5.0.3 Future work

Some of the work that would be interesting and useful to pursue in the future would include manufacturing the chip. Also, the topic of making the design really easy to use, to rival the prevalent digital microcontroller as the tool of choice of the regular engineer in industry, did not get as much time as originally hoped. For theory, creation and detection of malicious circuits, there is currently a certain lack of open literature on the topic, which makes it difficult to go as far in this area was one would like – one area of future work would necessarily require additional access to those materials.

Bibliography

[Ade08] S. Adee. The Hunt for the Kill Switch. *IEEE Spectrum*, May 2008.

[ASC$^+$08] A.T. Averstruz, W. Santa, D. Carlson, R. Jensen, S. Sanslaski, A. Helfenstine, and T. Denison. A 5uW/Channel Spectral Analysis IC for Chronic Bidirectional Brain-Machine Interfaces. *IEEE Journal of Solid State Circuits*, 43(12), December 2008.

[CMT06] 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), January 2006.

[Col07] D. Collins. DARPA 'Trust in IC's' Effort. `http://www.mtosymposium.org/2007/presentation/21_Collins_Trust_version2.pdf`, March 2007.

[CSM05] CSM. A Look Back in Time - the History of Analog Computing. *IEEE Control Systems Magazine*, 25(3), June 2005. A Look Back in Time - the History of Analog Computing.

[Dev08] Analog Devices. General-Purpose CMOS Rail-to-Rail Amplifers. *Datasheet*, 2008.

[IPE10] Mentor Graphics M8051EW+ 8-bit Microcontroller. `http://www.ip-extreme.com/IP/m8051ew+.shtml`, 2010.

[Jun06] Walter Jung, editor. *Op Amp Applications Handbook*. Newnes, 2006.

[KK72] Granino A. Korn and Theresa M. Korn. *Electronic analog and hybrid computers*. McGraw-Hill, second edition, 1972.

[LZ05] W. Liao and Y. Zhao. Using Dual-Axis Accelerometers to Protect Hard Disk Drives. *Analogue Dialogue*, 39-11, November 2005.

[Max08] Maxim. Using the MAXQ3210 in an Audio Attenuator Circuit. *Application Note*, June 2008.

[OVL96] Vojin G. Oklobdzija, David Villeger, and Simon S. Liu. A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach. *IEEE Transaction on Computers*, 45(3):294–306, March 1996.

[Sha07] Brian Sharkey. Trust in Integrated Circuits Program, Briefing to Industry. `http://www.darpa.mil/mto/solicitations/baa07-24/Industry_Day_Brief_Final.pdf`, March 2007.

[Sta66] Philbrick Researches Inc. Engineering Staff. *Applications Manual for Computing Amplifiers for Modeling, Measuring, Manipulating & Much Else*. Philbrick Researches, Inc., 1966.

[Vig03] Benjamin Vigoda. *Analog Logic: continuous-time analog circuits for statistical signal processing*. PhD dissertation, Massachusetts Institute of Technology, 2003.

[Wil07] D. Wilt. Trusted ICs Proposers Day Metrics Discussion. `http://www.darpa.mil/MTO/solicitations/baa07-24/Proposers_Day_Final.pdf`, 2007.