Radio Frequency Rectifiers for DC-DC Power Conversion

by

Riad Samir Wahby

S.B., Massachusetts Institute of Technology (2002)

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

Master of Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2004

Author .....
Department of Electrical Engineering and Computer Science
May 20, 2004

Certified by .....
David J. Perreault
Carl Richard Soderberg Assistant Professor of Power Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Radio Frequency Rectifiers for DC-DC Power Conversion

by

Riad Samir Wahby

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2004, in partial fulfillment of the
requirements for the degree of
Master of Engineering

Abstract

A significant factor driving the development of power conversion technology is the need to increase performance while reducing size and improving efficiency. In addition, there is a desire to increase the level of integration of DC-DC converters in order to take advantage of the cost and other benefits of batch fabrication techniques. While advances in the power density and integration of DC-DC converters have been realized through development of better active device technologies, much room for improvement remains in the size and fabrication of passive components.

To achieve these improvements, a substantial increase in operating frequency is needed, since intermediate energy storage requirements are inversely proportional to frequency. Unfortunately, traditional power conversion techniques are ill-suited to handle this dramatic escalation of switching frequency. New architectures have been proposed which promise to deliver radical performance improvements while potentially reaching microwave frequencies. These new architectures promise to enable substantial miniaturization of DC-DC converters and to permit much a higher degree of integration.

The principal effort of this thesis is the development of design and characterization methods for rectifier topologies amenable to use in the new architectures. A computational design approach allowing fast and accurate circuit analysis and synthesis is developed and applied, along with traditional analysis, to two demonstrative rectifier topologies. In addition, the application of coupled magnetic structures for parasitic mitigation is considered. Experimental implementations are investigated to verify analytic and computational results.

Thesis Supervisor: David J. Perreault
Title: Carl Richard Soderberg Assistant Professor of Power Engineering

Acknowledgments

Thanks first and foremost to my parents. Their dedication to my education, unconditional love, and unwavering support have been, simply put, astounding. To my sisters, too, I say thanks for everything.

Not long ago, I was unsure of my career path. It was Professor Dave Perreault who brought me firmly back into the electrical engineering fold, so to speak, and for that I am in his debt. His incredible enthusiasm and unparalleled knowledge of seemingly *everything* have made working for him the past year an awesome learning experience and a source of constant intellectual stimulation.

I owe thanks to my friends as well. In no real order, thanks to Jack, Jim, Mar, Andrew, Rob, Chris, Josh, Juan, Claudio, and everyone else in LEES; to Hippo, Ariel, Gautham, Sherv, Ian, Rodin, Amrys, George, Max, Sam, Kurt, Petar, May, and the rest of the Fort Awesome crew; and to Kevin, Dave, Lael, Perry, and the rest of Clam's old guard. If for some reason I didn't list you here—thank you, too.

A long time ago, I used to consider myself a violinist (and what's more, other people did as well!). My musical training has proven an immeasurable help in many unexpected ways; thus, special thanks go to Doris Preucil and to Claudia Johnson.

Finally, I would like to thank the Massachusetts Institute of Technology and the member companies of the MIT/Industry Consortium on Advanced Automotive Electrical/Electronic Components and Systems, whose sponsorship enabled this research.

To my parents, whose dedication has been without parallel.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

CONTEMPORARY research in DC-DC power conversion is strongly motivated by the need to increase performance while reducing size and maintaining or improving efficiency. In addition, there is a desire to achieve a higher degree of integration of DC-DC converters, allowing major portions of systems to be manufactured using batch fabrication techniques. While some improvements in the power density and integration of DC-DC converters can be realized through extensive miniaturization and integration of active components, equal or greater benefit can be achieved by reducing the size of the passive components. Doing so requires in almost all cases an increase in the frequency at which the converter operates, since intermediate energy storage requirements are inversely proportional to operating frequency.

Unfortunately, this increase in operating frequency is not without costs. At higher frequencies, non-idealities in circuit components become much more important. Moreover, loss mechanisms that can generally be ignored at low frequencies become crucial design considerations. Finally, control strategies commonly used at low frequencies become unwieldy when used at high frequencies.

To combat these difficulties, new circuit topologies and system architectures can be used.

Replacing hard-switched square-wave topologies with resonant, soft-switched converters allows high frequency converter designs that take advantage of techniques employed in tuned radio frequency power amplifiers. To solve the problem of controlling these high-frequency DC-DC converters, a new architectural approach will be employed which partially decouples the problems of efficient power conversion and controlled power delivery.

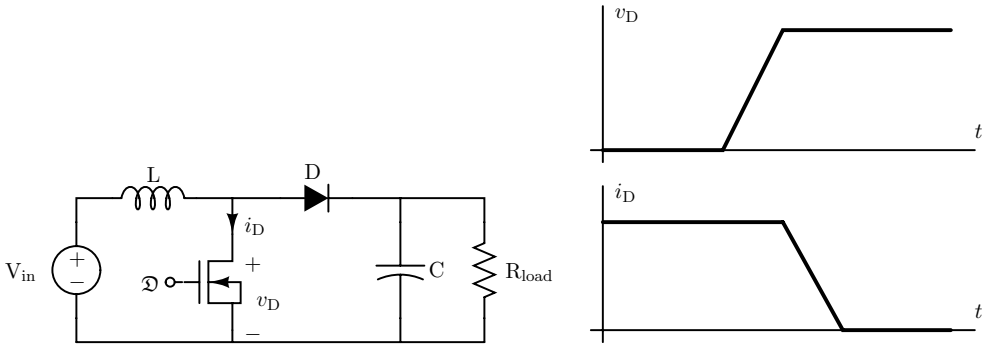
This thesis will explore rectification of power at high frequencies (100 MHz–1 GHz) and its application to new architectures for DC-DC conversion. In particular, rectifier topologies suitable for parallel combination in a cellular converter architecture will be designed and characterized. In addition, a new computational approach to circuit analysis and design synthesis will be presented.

1.1.1 Problems With High-Frequency Power Conversion

Whereas at low frequencies conduction losses are generally dominant, at high frequencies two other loss mechanisms, switching loss and gating loss, must also be considered. In addition, traditional control strategies for low frequency converters are impractical at high frequencies. Finally, implementing passive components compatible with efficient power processing at high frequency is often difficult; inductors are of particular concern, as there exist few permeable materials whose performance is acceptable for application in high frequency power conversion.

Switching Loss

Switching loss arises from the fact that no practical active element can turn on or off instantaneously: there is some interval during which the device must traverse the region between the on and off states. During this time, the device both conducts current and drops voltage, dissipating power. Figure 1-1 illustrates a boost converter (1-1(a)) and the on-to-off transient of the MOSFET (1-1(b)). Assuming a large input inductor (i.e., continuous conduction operation), the transistor must conduct current until its drain voltage rises to one diode drop above the output voltage; only then is the diode forward biased and able



(a) A boost converter circuit.

(b) Switching waveforms.

Figure 1-1: An illustration of switching loss in the boost converter.

to conduct current to the output. For a typical device such a transition takes about 15 nanoseconds—insignificant at 100 kHz, but untenable at 100 MHz [1].

To ameliorate switching losses, zero-voltage switching topologies are often employed [2]. In these topologies, a continuous resonating action is used to ensure that switches only change state when supporting little or no voltage. While ZVS can be advantageous when applied to DC-DC conversion at full load, it becomes a problem at light load: since the losses accompanying resonant operation are present at all load conditions, efficiency when delivering only a fraction of full power is severely reduced.

Gating Loss

Gating loss is a result of the fact that turning any active device on or off involves a transfer of energy. In a MOSFET, for example, the gate capacitance must be charged to turn the device on and discharged to turn it off. In a switching scheme where the gate terminal is charged from the supply and discharged into ground, power loss proportional to frequency results. By employing resonant structures in driving the gate, energy can be recovered and reused in subsequent cycles. In the simplest of such circuits, the energy transferred onto the gate capacitor is transferred off and stored on an external inductor until the next switching

cycle; in this way, energy is only lost in conduction [3–6]. In effect, a resonant gate driver is itself an RF amplifier; thus, the benefits of resonant gate drive can often be most fully developed by using a cascade of resonant converters, one driving the next.

Control Strategies

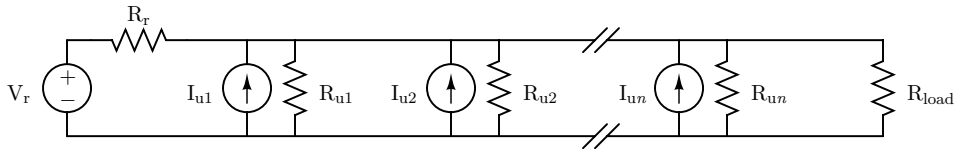
Control strategies employed at low frequencies are not easily adapted to efficient high frequency topologies. Since such strategies often require direct manipulation of the harmonic content of operating waveforms, they are generally incompatible with ZVS and resonant gate drive topologies. Regulation can be achieved by other techniques, such as frequency control [7, 8]; even so, realizing regulation over a wide load range becomes increasingly difficult as frequencies increase, as do considerations of converter dynamics and the complexity of implementing control circuitry. Moreover, cascaded resonant gate drive is often at odds with such a strategy, forcing the designer to trade off ease of control for efficiency.

Magnetic Components

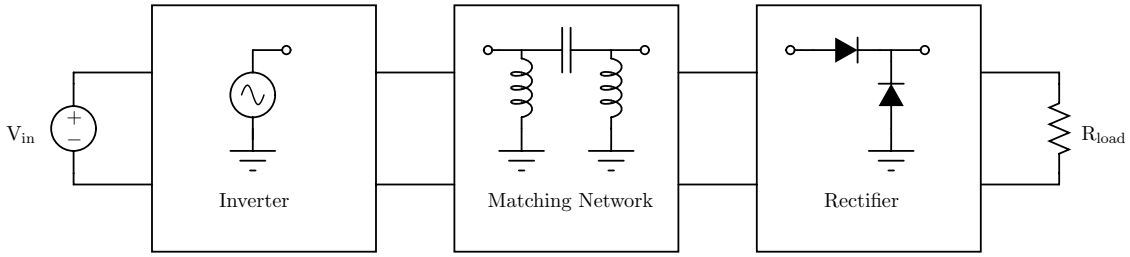
At low frequencies, passive component sizing is dominated by volumetric energy storage limits. At higher frequencies, losses become the most important consideration in the sizing of magnetic components. For most ferrite materials, core losses increase dramatically with frequency; to mitigate these loss characteristics, flux derating becomes necessary. As a result, increasing frequency can cause inductor sizes to worsen instead of improving [9]. Air-core magnetic passives, lacking a lossy permeable core material, do not suffer this limitation; however, the consequently reduced inductance per turn squared requires very high frequency operation to achieve reasonable component size and Q.

1.1.2 Proposed Architectural Improvements

To overcome the limitations of very high frequency operation while taking advantage of its benefits, new cellular architectures have been proposed [10]. These architectures utilize



(a) The proposed architectures employ one regulating cell (V_r, R_r) and several unregulated cells ($I_{u1 \ldots un}$, $R_{u1 \ldots un}$).



(b) The basic structure of an unregulated cell comprises an inverter and a rectifier connected by an impedance matching network.

Figure 1-2: The basic structure of the proposed architectures.

high efficiency tuned RF power amplifier circuits while partitioning the energy conversion and control functions so as to avoid the limitations discussed in section 1.1.1.

The new architectures employ two types of cells. RF power converters comprising an inverter and rectifier operating at very high frequency are used as unregulated, on/off-controlled parallel cells. Each unregulated cell operates only when required, and only over a narrow range of conditions; this allows nearly maximum performance to be achieved at all times. Control is provided by a regulating cell, which only provides a fraction of the output power; this reduces the impact of regulating cell loss on total efficiency. Figure 1-2(a) illustrates the proposed architecture; the basic structure of one unregulated cell is depicted in Fig. 1-2(b).

The principal effort of this thesis is the development of design and characterization methods for rectifier topologies amenable to use in the aforementioned architectures[1]. In

[1]The design of a suitable inverter is the topic of a separate investigation.

this application, efficiency is of paramount importance. In addition, a well-characterized input impedance is crucial for optimizing the matching network that connects the inverter and rectifier. Finally, a rectifier having high output impedance is desirable, since this facilitates load sharing among a parallel combination of several unregulated cells; failure to share the load equally could result in inefficient operation or cell failure [11–13].

In radio frequency rectifiers employing diodes, both conduction losses and switching losses are important considerations[2]. To achieve acceptable efficiency, a resonant zero-voltage switching topology which minimizes peak diode currents is necessary [14–20]. Moreover, parasitics associated with the active device are very important at high frequencies; thus, a topology that makes use of these parasitics will likely outperform one that does not. Alternatively, undesirable parasitic inductances might be cancelled using coupled magnetic structures [21, 22].

1.2 Thesis Objectives

This thesis explores the selection, characterization, design, and implementation of radio frequency rectifiers with the aim of improving both the process and product of future rectifier designs. To this end, I have several objectives.

First, I will demonstrate general[3] methods for cocktail napkin-style analysis of linear resonant circuits. As much as possible, I will attempt to employ intuition and reasonable estimates, and will show the methods used for developing both. When necessary, arduous mathematical derivations will be accompanied by ample verbiage, allowing the casual reader to proceed quickly to the final result.

Second, I will develop an approach for computational modeling of nonlinear resonant circuits. In particular, I will show algorithms for fast and accurate computational circuit synthesis, allowing the user to proceed from specification to design very quickly. Though

[2] Synchronous rectification, not considered in this thesis, incurs additional penalty in gating, but may result in overall efficiency improvement, since diode drop and the resulting conduction loss are largely avoided.

[3] . . . though not necessarily original!

some minimal programming skill will likely be useful, the reader will not be unduly taxed with abstruse code listings[4].

Third, I will apply these analytic and computational techniques to example rectifier topologies, and establish the groundwork for similar analysis of others. In addition to standard topologies, I will explore the use of coupled magnetic structures (which can be, e.g., implemented on printed circuit boards), allowing device parasitic mitigation and reduced manufacturing cost. These will be accompanied by both simulation and experimental evaluation.

1.3 Thesis Organization

This thesis is divided into six chapters, including this introduction.

Chapter 2 reviews analytic techniques useful in linear resonant circuit analysis, and demonstrates the application of these techniques to the analysis and design of resonant impedance transformers.

Chapter 3 introduces the use of computational modeling for nonlinear circuit analysis. In this chapter, algorithms are developed that allow for extremely fast design of rectifier circuits and which pave the way for the analyses performed in the proceeding chapters.

Chapter 4 discusses the first of two resonant rectifier topologies, exploring the design and implementation process of the series resonant rectifier. Chapter 4 comprises a discussion of analytic models including Fourier and describing function analysis of input impedance, a comparison of the analytic models with simulation, and a discussion of the strengths and weaknesses of the series resonant topology. Experimental results are also discussed.

Chapter 5 covers another topology, the shunt resonant rectifier. In addition to a discussion of analytic models and simulated results, the exploration of the shunt topology is extended by application of coupled magnetic structures which allow the mitigation of undesired parasitic effects. Implementations of rectifiers with and without parasitic mitigation are presented.

[4]For those of true grit, full code is provided in the appendices.

Finally, Chapter 6 concludes the thesis with a summary of the results and suggestions for continued work.

Chapter 2

Useful Techniques for Resonant Circuit Analysis

THIS CHAPTER is intended to familiarize the reader with useful analytic techniques for linear resonant circuits. Much of the material is available from other sources [23–25]; this chapter has been included nevertheless, since analytic approaches to the same problem differ substantially from one book to another, and because the particular techniques reviewed here are a key component of the insight leading to the algorithms presented in chapter 3. In addition, the last section discusses an active impedance matching network which might be employed in a feedforward or feedback system to effect a dynamic match.

2.1 Resonant Impedance Analysis

A major challenge in designing RF rectifiers, and indeed almost any RF power circuit, is ensuring that maximum power is transferred from input to output. Because many RF circuits operate over only a very limited frequency band, the use of narrowband impedance transformation circuits is exceedingly common. Before delving into the analysis of such circuits, however, it's useful to remember the reason for their use.

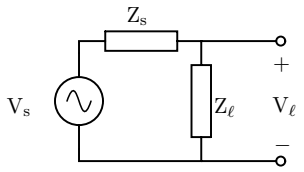


Figure 2-1: A simple circuit for deriving Jacobi's theorem.

2.1.1 Jacobi's Theorem

In low frequency designs, there is no difficulty in obtaining power gain. At radio frequencies, however, the scarcity of power gain necessitates the resurrection of an oft-forgotten theorem taught in every basic circuits course: for a fixed source impedance, maximum power is always transferred into a conjugate matched load[1]. Figure 2-1 shows an example circuit from which derivation of the maximum power transfer theorem is straightforward.

$$P_\ell = \frac{{V_\ell}^2}{R_\ell} = \frac{{V_s}^2 \cdot R_\ell}{(R_\ell + R_s)^2} \tag{2.1}$$

$$\frac{\partial P_\ell}{\partial R_\ell} = V_s \left[\frac{1}{(R_\ell + R_s)^2} - \frac{2R_\ell}{(R_\ell + R_s)^3} \right] \tag{2.2}$$

$$\therefore R_\ell = R_s \tag{2.3}$$

$$P_\ell = V_\ell I_\ell \cos(\phi) \tag{2.4}$$

$$\phi = \tan^{-1}\left(\frac{X_s + X_l}{R_\ell} \right) \tag{2.5}$$

$$\therefore X_\ell = -X_s \tag{2.6}$$

[1]This theorem is often referred to as the Maximum Power Transfer theorem to avoid confusion with a mathematical theorem of the same name. Moritz Hermann Jacobi is responsible for the former; his younger brother, Carl Richard, stated the latter. In the mid-19$^{\text{th}}$ century, engineers (including James Prescott Joule!) misinterpreted this theorem to imply that an electrical machine could never be made more than 50% efficient while delivering substantial power. This is, of course, false—for a *fixed* source impedance, maximum power is transferred into a conjugate matched load. On the other hand, differentiating equation 2.1 with respect to R_s quickly shows that if instead the load impedance is fixed, maximum power is transferred from zero source impedance. This fact was realized by Thomas Edison or his assistant, Francis Robbins Upton, whereupon they designed their dynamos for minimal armature resistance and realized efficiencies near 90%.

2.1.2 Series-Shunt Transformation

If we are interested in the impedance of a two-element passive network only over a narrow frequency range, it is possible to replace a series-connected network with a shunt-connected one; to derive the new component values, simply equate the shunt and series impedances and solve. For example, to replace a series RL network with a shunt one,

$$\mathrm{Z} = \mathrm{R_{ser}} + j\omega\mathrm{L_{ser}} \tag{2.7}$$

$$= \frac{j\omega\mathrm{L_{shn}}\mathrm{R_{shn}}}{\mathrm{R_{shn}} + j\omega\mathrm{L_{shn}}} = \frac{\mathrm{R_{shn}}\left(\omega\mathrm{L_{shn}}\right)^2 + j\omega\mathrm{L_{shn}}\mathrm{R_{shn}^2}}{\mathrm{R_{shn}^2} + \left(\omega\mathrm{L_{shn}}\right)^2} \tag{2.8}$$

Recalling that

$$\mathrm{Q} = \frac{\omega\mathrm{L_{ser}}}{\mathrm{R_{ser}}} = \frac{\mathrm{R_{shn}}}{\omega\mathrm{L_{shn}}} = \frac{1}{\omega\mathrm{R_{ser}}\mathrm{C_{ser}}} = \omega\mathrm{R_{shn}}\mathrm{C_{shn}} \tag{2.9}$$

we can quickly simplify 2.7 and 2.8 to

$$\mathrm{R_{shn}} = \mathrm{R_{ser}}\left(\mathrm{Q}^2 + 1\right) \tag{2.10}$$

$$\mathrm{L_{shn}} = \mathrm{L_{ser}}\left(\frac{\mathrm{Q}^2 + 1}{\mathrm{Q}^2}\right) \tag{2.11}$$

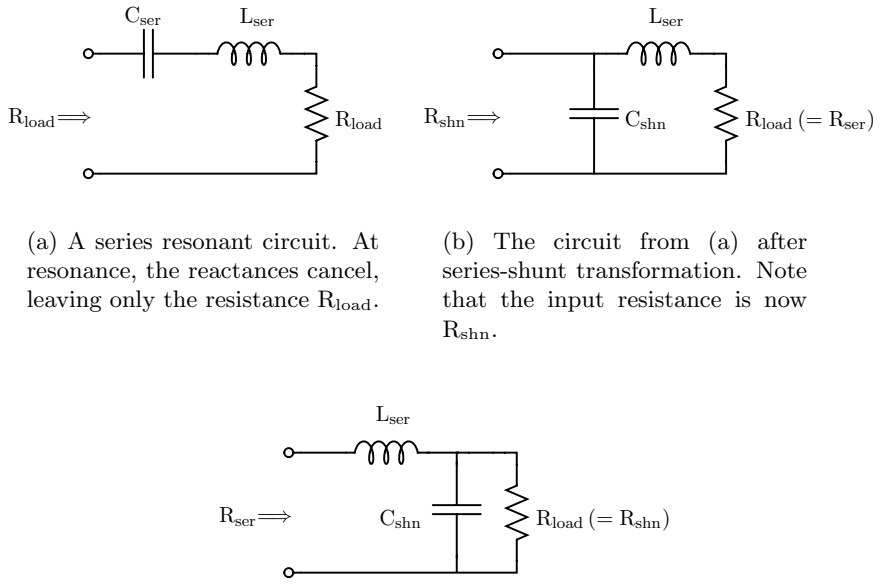
Equation 2.11 can be expressed more generally in terms of reactance:

$$\mathrm{X_{shn}} = \mathrm{X_{ser}}\left(\frac{\mathrm{Q}^2 + 1}{\mathrm{Q}^2}\right) \tag{2.12}$$

From this we surmise that

$$\mathrm{C_{shn}} = \mathrm{C_{ser}}\left(\frac{\mathrm{Q}^2}{\mathrm{Q}^2 + 1}\right) \tag{2.13}$$

These series-shunt equivalence relations form the basis for all resonant impedance matching circuits.



(a) A series resonant circuit. At resonance, the reactances cancel, leaving only the resistance R_{load}.

(b) The circuit from (a) after series-shunt transformation. Note that the input resistance is now R_{shn}.

(c) As in (b), this circuit is produced by a series-shunt transformation. In this case, the resulting input resistance is *smaller* than before transformation.

Figure 2-2: The L-match is derived by series-shunt transformation of one element in a series resonant circuit.

2.1.3 The L-Match

In the circuit of Fig. 2-2(a), assuming that the inductor and capacitor are at resonance, the resistance seen across the terminals is R_{load}. Transforming the capacitance via equation 2.13 to the circuit of Fig. 2-2(b) also transforms the apparent input resistance; we now see R_{shn} at the input. Since we know from equation 2.10 that R_{shn} must always be larger than R_{ser}, the input impedance of Fig. 2-2(b) must always be larger than that of Fig. 2-2(a). To transform a large impedance downward into a smaller one, it is necessary to instead look into the series port of the network, as shown in Fig. 2-2(c).

When designing an L-match, the two parameters that are generally specified are the transformation ratio $\frac{R_{shn}}{R_{ser}}$ and the operating frequency ω. Since there are only two en-

ergy storage modes in this circuit, selecting both the transformation ratio and operating frequency completely specifies the circuit parameters[2].

From equation 2.10, we know that

$$Q = \sqrt{\frac{R_{shn}}{R_{ser}} - 1} \tag{2.14}$$

Some minimal algebra and equation 2.9 then produce the rest of the network:

$$C_{shn} = \frac{Q}{\omega R_{shn}} \tag{2.15}$$

$$L_{ser} = \frac{QR_{ser}}{\omega} \tag{2.16}$$

It is also possible to interchange the inductor and capacitor, making a high-pass L-match instead of the low-pass matches depicted in Fig. 2-2(b) and 2-2(c). Once again, using equation 2.9:

$$C_{ser} = \frac{1}{\omega R_{ser} Q} \tag{2.17}$$

$$L_{shn} = \frac{R_{shn}}{\omega Q} \tag{2.18}$$

2.2 Higher Order Matching Networks

While theoretically the L-match allows arbitrary impedance transformations, in many practical circumstances it produces a sub-optimal solution. Since the transformation ratio fixes Q, it is not possible to select the bandwidth of an L-match arbitrarily. Moreover, extreme transformation ratios require impractically large network Q.

To get around this limitation, it is possible to combine several L-matches, stepping from the source impedance to one or more intermediate values and then to the load. If a very large transformation ratio is required, the intermediate impedance(s) serve to reduce the

[2]In principle, one could choose any two circuit parameters—Q and L_{ser}, for example—and design the rest of the match. Of course, this is not terribly practical, since it is not usually the case that the operating frequency is a free parameter to be specified when designing the matching network!

loaded Q necessary at each stage. On the other hand, if a very narrow bandwidth is desired, but with only a very small change in impedance, an image resistance can be selected to produce the appropriate overall Q. In the former case, an L-match ladder results. In the latter, an upward step is followed by a downward one (or vice-versa); such networks are known as T- or Π-matches.

Alternatively, the designer may choose to use a tapped resonator for a matching network. In this configuration, a parallel LC resonator is tapped by splitting the capacitor or the inductor (or both) into two series elements, then connecting the load and source across the appropriate ports. Like the T- and Π-matches, the tapped capacitor and tapped inductor resonator provide three degrees of freedom, while tapping both yields four.

2.2.1 The L-Match Ladder

When a very large transformation is necessary, it is often not possible to use a single L-match stage. If, for example, the required Q is comparable to the Q_L of the inductor to be used, circuit performance will be substantially impaired, since in this case the series resistance of the inductor becomes significant. In cases such as this, a series of L-match stages, each doing a portion of the total transformation, can be used.

In designing large L-match ladders, the issue of loss may be of particular concern. On the one hand, continually adding more stages will inevitably result in decreasing efficiency; at the same time, using fewer stages increases the required loaded Q per stage, increasing the resonant currents and consequently the conduction losses. While the optimal solution may change substantially depending on other constraints (e.g., required size, available component values, and so on), it is interesting to note that in the limiting case of an extremely large L-match ladder, the optimal transformation ratio per stage approaches a constant value[3].

Imagine that we want to match two impedances requiring a very large transformation ratio G. To do this, we will use N stages, each with a transformation ratio K, loaded

[3] A similar argument is often made for bandwidth optimization in a cascade of amplifiers, each with a constant gain-bandwidth product. In that case, most arguments conclude that the optimum gain is either e (based on open-circuit time constant analysis, as in [26]) or $\sqrt{e}$ (based on bandwidth shrinkage arguments, as in [24]). In reality, $\sqrt{e}$ is somewhat more accurate.

$Q = Q_I$, and unloaded Q (that is, due to loss in the passives) of Q_u. So

$$G = K^N \tag{2.19}$$

$$N = \log_K(G) = \frac{\ln(G)}{\ln(K)} \tag{2.20}$$

$$Q_I = \sqrt{K-1} \tag{2.21}$$

Q_I is the loaded Q of one stage, so we know

$$Q_I = \frac{\text{Energy stored}}{(\text{Energy passed to the next stage}) + (\text{Energy lost})} \tag{2.22}$$

If we assume that we are transferring much more power than we are losing (i.e., the unloaded Q is much greater than the loaded Q), then each stage delivers approximately the total output energy to the next stage, stores E_I, and loses $E_{I,diss}$, where

$$E_I = E_{tot} Q_I \tag{2.23}$$

$$E_{I,diss} = \frac{E_I}{Q_u} \tag{2.24}$$

The total dissipation is therefore

$$E_{diss} = \frac{NE_I}{Q_u} = NE_{tot}\left(\frac{Q_I}{Q_u}\right) \tag{2.25}$$

$$= E_{tot}\left(\frac{\sqrt{K-1}}{Q_u} \cdot \frac{\ln(G)}{\ln(K)}\right) \tag{2.26}$$

To minimize the loss, we differentiate with respect to K

$$\frac{\partial}{\partial K}\left(\frac{E_{diss}}{E_{tot}}\right) = -\frac{1}{2}\frac{\ln(G)\left[K\ln(K) - 2K + 2\right]}{Q_u K \ln(2K)\sqrt{K-1}} \tag{2.27}$$

The mathematically inclined reader will no doubt note that the solutions to this equation are related to the Lambert W function [27]; choosing the principally valued branch,

$$K \approx 4.92 \tag{2.28}$$

$$Q_I \approx 1.98 \tag{2.29}$$

While in many practical cases a transformation ratio of 5 per stage is inconvenient, even in such cases this derivation can serve as a handy rule of thumb: transformation ratios much greater than 25 in a single stage are likely less efficient than an appropriately chosen ladder configuration.

2.2.2 The T- and Π-Matches

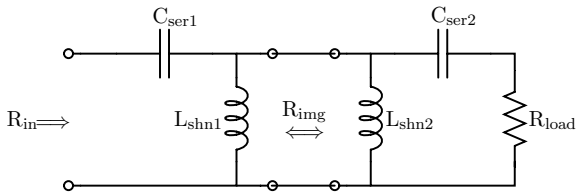
If the L-match is unacceptable not because of an impractical transformation ratio but because of an over-specified design, the T- and Π-matches provide a solution. Both allow the designer an additional degree of freedom by transforming up or down to an intermediate or *image* resistance and then back the other way to the desired resistance.

The T- and Π-matches are synthesized by putting two L-matches either front to front or back to back, as depicted in Fig. 2-3(a) and 2-3(b). Design proceeds in both cases by first choosing (or calculating based on the imposed constraints) the image resistance, then calculating the two resulting L-matches independently.

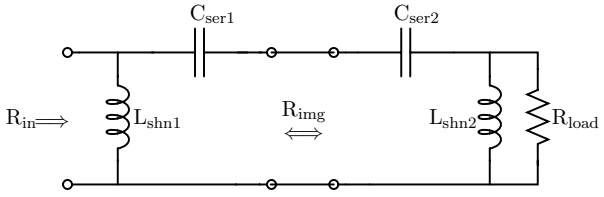
The equations for determining the values for the two L-matches are identical to those in section 2.1.3. A few additional equations relate overall network Q to R_{img} and to the component values adjacent to R_{img}. If we call the Q of the L-matches to the left and right of the image resistance Q_{left} and Q_{right}, respectively,

$$Q = Q_{left} + Q_{right} \qquad \text{(the overall network Q)} \tag{2.30}$$

$$= \frac{R_{img}}{\omega \left(\frac{L_{shn1} L_{shn2}}{L_{shn1} + L_{shn2}} \right)} \qquad \text{(for the T-match)} \tag{2.31}$$



(a) The T-match is formed by two L-matches with the image resistance in the shunt position. Thus, R_{load} is transformed up to R_{img} and then back down to R_{in}.



(b) In the Π-match, the image resistance is on the series port. R_{load} is transformed down to R_{img}, which is transformed up to R_{in}.

Figure 2-3: The T- and Π-matches, composed of two L-matches, give the designer another degree of freedom.

$$= \frac{1}{\omega R_{img}\left(\frac{C_{ser1}C_{ser2}}{C_{ser1}+C_{ser2}}\right)} \qquad \text{(for the Π-match)} \qquad (2.32)$$

If we were instead to make low-pass networks, i.e., swap the inductors and capacitors,

$$Q = \omega R_{img}\left(C_{shn1} + C_{shn2}\right) \qquad \text{(for the T-match)} \qquad (2.33)$$

$$= \frac{\omega\left(L_{ser1} + L_{ser2}\right)}{R_{img}} \qquad \text{(for the Π-match)} \qquad (2.34)$$

Since there is an additional degree of freedom, the designer can choose, for example, the total shunt inductance at the center of the T-match, the overall transformation ratio, and the operating frequency. While it's possible to then derive an analytic expression for R_{img},

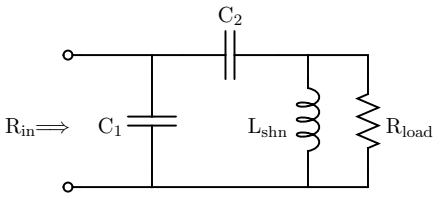


Figure 2-4: The tapped capacitor match gives more freedom than the L-match and, in some cases, more reasonable component values than the T- or Π-match.

the result is truly hideous; it is almost certainly better to employ an aid such as MATLAB or to guess at the overall Q and iterate to the final solution.

2.2.3 Tapped Capacitor Resonator

A departure from transformers based on the L-match, tapped resonators give three degrees of freedom while allowing in some cases more reasonable component values than the T- or Π-match. Note that for all of the tapped resonator matches, I have arbitrarily chosen the input and output ports; as with any linear network, we can freely interchange these as appropriate to a particular design.

The analysis of the tapped capacitor match is straightforward: convert R_{load} into its series equivalent, then convert that series equivalent to the shunt impedance across C_1. Of course, one must take care to remember which Q value is used for each of the transformations.

Clearly,

$$\frac{R_{in}}{R_{load}} = \frac{{Q_{left}}^2 + 1}{Q^2 + 1} \tag{2.35}$$

$$Q = \frac{R_{load}}{\omega L_{shn}} \tag{2.36}$$

$$Q_{left} = \omega R_{in} C_1 \tag{2.37}$$

We define

$$R_{ser} = \frac{R_{load}}{Q^2 + 1} = \frac{R_{in}}{{Q_{left}}^2 + 1} \tag{2.38}$$

$$C_{1,ser} = C_1 \left(\frac{{Q_{left}}^2 + 1}{{Q_{left}}^2} \right) \tag{2.39}$$

$$C_{equiv} = \frac{C_{1,ser} C_2}{C_{1,ser} + C_2} \tag{2.40}$$

This gives us another expression for the overall network Q

$$Q = \frac{1}{\omega R_{ser} C_{equiv}} \tag{2.41}$$

To design a tapped capacitor match, first decide which parameters will be set. For example, we can pick the transformation ratio, frequency, and overall network Q. This lets us use equation 2.36 to determine L_{shn}. Equation 2.35 then gives a value for Q_{left}, from which equation 2.37 gives us the value for C_1. To get the value for C_2, we use equation 2.41. A similar process allows us to instead choose C_{equiv} or L_{shn} in addition to transformation ratio and center frequency.

2.2.4 Tapped Inductor Resonator

The tapped inductor resonator is very similar to the tapped capacitor resonator in that it allows an additional degree of freedom over the L-match. Once again, component values for this circuit may in some cases be more reasonable than those for other third-order matching networks.

Analysis and design of the tapped inductor match proceeds along the same lines as the process for the tapped capacitor network.

$$\frac{R_{in}}{R_{load}} = \frac{{Q_{left}}^2 + 1}{Q^2 + 1} \tag{2.42}$$

$$Q = \omega R_{load} C_{shn} \tag{2.43}$$

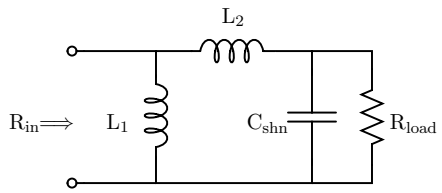


Figure 2-5: The tapped inductor match is a close relative of the tapped capacitor match of section 2.2.3.

$$Q_{left} = \frac{R_{in}}{\omega L_1} \tag{2.44}$$

Again we make some useful definitions:

$$R_{ser} = \frac{R_{load}}{Q^2 + 1} = \frac{R_{in}}{{Q_{left}}^2 + 1} \tag{2.45}$$

$$L_{1,ser} = L_1 \left(\frac{{Q_{left}}^2}{{Q_{left}}^2 + 1} \right) \tag{2.46}$$

$$L_{equiv} = L_{1,ser} + L_2 \tag{2.47}$$

Which leads us to an expression for network Q in terms of the inductors:

$$Q = \frac{\omega L_{equiv}}{R_{ser}} \tag{2.48}$$

2.2.5 The Double-Tapped Resonator

If we can tap either the capacitor or the inductor in the resonator, why not tap both? Doing so gives us *two* degrees of freedom beyond transformation ratio and center frequency, allowing us to choose, for example, total inductance and network Q.

As in the other resonator matches, R_{load} goes through two series-shunt transformations to become R_{in}; in this case, however, the overall network Q is present in neither transfor-

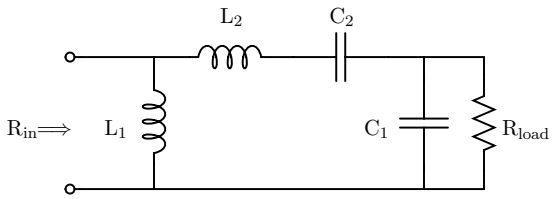


Figure 2-6: The double-tapped resonator gives an additional degree of freedom beyond the third-order matching networks.

mation, a result of the addition of another energy storage mode.

$$\frac{R_{in}}{R_{load}} = \frac{Q_{left}^2 + 1}{{Q_{right}}^2 + 1} \tag{2.49}$$

$$Q_{left} = \frac{R_{in}}{\omega L_1} \tag{2.50}$$

$$Q_{right} = \omega R_{load} C_1 \tag{2.51}$$

Define

$$R_{ser} = \frac{R_{load}}{{Q_{right}}^2 + 1} = \frac{R_{in}}{{Q_{left}}^2 + 1} \tag{2.52}$$

$$L_{1,ser} = L_1 \left(\frac{{Q_{left}}^2}{{Q_{left}}^2 + 1} \right) \tag{2.53}$$

$$C_{1,ser} = C_1 \left(\frac{{Q_{right}}^2 + 1}{{Q_{right}}^2} \right) \tag{2.54}$$

$$Q = \frac{\omega \left(L_{1,ser} + L_2 \right)}{R_{ser}} = \frac{1}{\omega R_{ser}} \left(\frac{1}{C_{1,ser}} + \frac{1}{C_2} \right) \tag{2.55}$$

Of course, we can also say

$$Q_{left} = \frac{\omega L_{1,ser}}{R_{ser}} \tag{2.56}$$

$$Q_{right} = \frac{1}{\omega R_{ser} C_{1,ser}} \tag{2.57}$$

which lets us simplify equation 2.55 slightly:

$$Q = Q_{left} + \frac{\omega L_2}{R_{ser}} = Q_{right} + \frac{1}{\omega R_{ser} C_2} \tag{2.58}$$

It is useful to note that

$$Q_{left} = \sqrt{\frac{R_{in}}{R_{load}} \left({Q_{right}}^2 + 1 \right) - 1} \tag{2.59}$$

$$Q_{right} = \sqrt{\frac{R_{load}}{R_{in}} \left({Q_{left}}^2 + 1 \right) - 1} \tag{2.60}$$

2.3 Active Impedance Matching

A system involving a dynamically changing load or source impedance may not be amenable to the static matching networks heretofore discussed. In these cases, a dynamically adjustable matching network with appropriate control might allow more optimal operation.

While the fabrication of an adjustable inductor might be possible by using core saturation characteristics, as in [28], such a device would almost certainly suffer at very high frequencies from the effects discussed in section 1.1.1. On the other hand, adjustable capacitors are readily available: the junction capacitance of a reverse-biased diode varies as a (very nonlinear) function of voltage. The use of diodes as nonlinear capacitors is a well-known technique; so-called *varactors* (shortened from "variable reactor") are often used in the tuning circuits of, e.g., FM receivers [24, 29].

Junction capacitance as a function of reverse voltage V_r can be modeled as

$$C_j = \frac{C_{j0}}{\left(1 + \frac{V_r}{V_\phi}\right)^m} \tag{2.61}$$

where C_{j0} is the zero-bias junction capacitance, V_ϕ is the built-in potential (generally around .5 Volts), and m is a constant determimined by the doping gradient at the junction (for an abrupt junction, $m = .5$) [29].

Figure 2-7 shows one possible active matching implementation. In this circuit, two

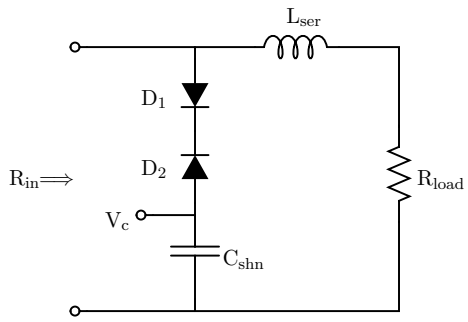


Figure 2-7: Diode junction capacitances can be used to obtain a controlled L-match. A DC voltage V_c applied as shown controls the shunt capacitance.

diodes are used to ensure that one always remains reverse biased, preventing the input from shorting to V_c. By changing the value of V_c, the junction capacitance of the diodes can be changed. Since V_c is a DC value, it might easily be supplied by a simple feedforward network (perhaps relating the matching network's characteristics to supply voltage) or by the error signal from a phase-locked loop [30, 31] or delay-locked loop [32].

Chapter 3

Computational Techniques for Nonlinear Circuits

3.1 Motivations for Computational Analysis

TECHNIQUES were developed in chapter 2 for the matching of linear networks with known input and output impedances. While this is extremely useful, e.g., when matching a power amplifier to an antenna, nonlinear resonant networks present an additional challenge: their input impedance is dependent on the conditions under which they operate. In order to match these networks, it is necessary to establish an operating point and design a matching network at that operating point. In many cases, however, the operating point changes with the addition of the matching network!

In chapter 4, Fourier analysis and describing function methods are used to derive an expression for the input impedance of a rectifier circuit under specified conditions. It is certainly possible to design a matching network using such an expression; unfortunately, this approach is of limited use in practical designs. First, to make the Fourier analysis tractable, idealization of the nonlinear circuit elements is required (for example, a diode might be treated as a switch in parallel with a linear capacitor). Unfortunately, doing so reduces the accuracy of the expression (by a substantive amount in many cases) and simultaneously

makes error analysis difficult. Moreover, the time spent in deriving an analytic expression may be considerably more than is warranted, especially given the inherent inaccuracies in the analysis.

For these reasons, the development of a more accurate and much faster computational approach is a significant step forward in the design of nonlinear resonant networks such as rectifiers. Some hesitation to abandon pencil-and-paper engineering in favor of a computational crutch is understandable; thus, in cases where back-of-the-envelope designs are desirable, the computational approaches developed in this chapter may still be of some use. For example, given a particular topology, curves of constant impedance versus power might be generated and used for fast initial estimates[1].

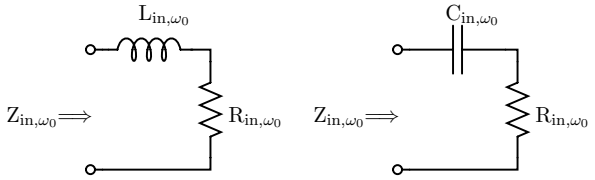
3.2 Nonlinear Network Impedance Considerations

When driving a nonlinear circuit, the definition of input impedance is not altogether clear. One might choose, for example, to define the input impedance as $Z_{in} = \frac{V_{in}}{I_{in}}$ instantaneously. It is quite likely, however, that the value of this expression would fluctuate with time, since the particular nonlinear network under consideration could produce substantial harmonic distortion.

Recalling that under sinusoidal excitation power can only be transferred at the fundamental frequency, a convenient solution to this problem presents itself: for a sinusoidal drive, the input impedance should be considered in a describing function [24, 33, 34] sense—that is, only components of the voltage and current at the fundamental frequency contribute to the input impedance. This solution is of particular value because almost any circuit simulation package can measure node voltages and branch currents. Once these are obtained, the fundamental magnitudes and phases can be computationally determined via a simple fast Fourier transform (FFT) calculation [25, 35].

Given the FFT results for input voltage and current, a linearized model of the input

[1]For those who feel this verges too closely on the Red Rubber Ball school of engineering, a quick comparison between the results produced with Fourier/describing function analysis and those produced by computational methods should be sufficiently convincing.



(a) Input impedance model for positive reactance.

(b) Input impedance model for negative reactance.

Figure 3-1: Linearized input impedance models.

impedance is easily constructed. If we call the fundamental voltage and current V_{in,ω_0} and I_{in,ω_0}, respectively, then

$$Z_{in,\omega_0} = \frac{V_{in,\omega_0}}{I_{in,\omega_0}} \tag{3.1}$$

$$R_{in,\omega_0} = \Re\{Z_{in,\omega_0}\} \tag{3.2}$$

$$X_{in,\omega_0} = \Im\{Z_{in,\omega_0}\} \tag{3.3}$$

If the reactance X_{in,ω_0} is positive, then the linearized model corresponds to Fig. 3-1(a) where

$$L_{in,\omega_0} = \frac{X_{in,\omega_0}}{\omega_0} \tag{3.4}$$

Otherwise, the model is represented by Fig. 3-1(b) with

$$C_{in,\omega_0} = \frac{1}{\omega_0 X_{in,\omega_0}} \tag{3.5}$$

Note that we can instead represent these input impedances[2] as parallel RL or RC networks via the transformations discussed in section 2.1.2.

[2]For brevity's sake, input impedances will be assumed to be in the describing function sense when appropriate throughout the rest of this thesis. Thus, the R_{in,ω_0} notation will be dropped in favor of simply R_{in}; input impedances are assumed to be at the fundamental frequency unless otherwise specified.

Even after we have computed the input impedance via the aforementioned method, care is necessary to ensure that our calculations remain valid: from the nonlinear network's point of view, nothing may change. Clearly the impedance viewed from the input of the nonlinear circuit looking back towards the source must remain the same. In addition, ensuring that the total power throughput remains constant guarantees that the input voltage and current to the nonlinear network do not change.

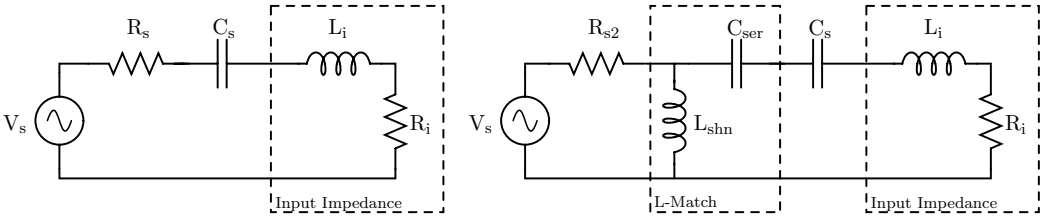
This requirement presents a particular difficulty when attempting to design a matching network: if we wish to make a conjugate match between source impedance and linearized input impedance, we must insert an appropriate matching network and at the same time keep constant the apparent source impedance. That is, we require that looking back into the matching network from the nonlinear network's input we see the same impedance that was used to calculate the linearized input impedance. Due to the linear nature of the resonant impedance transformers we are designing, the transformation ratio looking into the network from one side is the reciprocal of the ratio looking into the other side; thus, in order to achieve a conjugate match at the established operating point, the calculation of the source impedance must have been from a conjugate match in the first place—requiring that we knew (or were extremely lucky in guessing) the input impedance before calculating it!

3.3 The Iterated Approach

The problem of calculating input impedance from a conjugate match can be circumvented by iterating the calculation procedure.

3.3.1 Basic Iterated Matching

Taking as an example Fig. 3-2(a), our goal is to choose $R_s = R_i$ and $X_s = -X_i$ (for this example, we assume that the input impedance looks inductive; of course, this procedure applies equally for a capacitive input impedance). To begin, we choose a source impedance somewhat close to the conjugate of the expected input impedance value. After simulating



(a) A conjugate match is achieved by iteratively selecting R_s and C_s.

(b) After R_s and C_s are conjugate matched with R_i and L_i, a matching network can be inserted without changing the apparent source impedance

Figure 3-2: A demonstration of the iterated impedance matching approach.

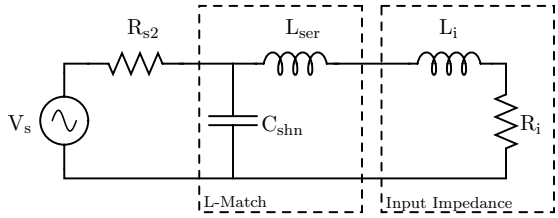
the circuit and calculating the input impedance value, we set $R_s = R_i$ and $\frac{1}{\omega C_s} = \omega L_i$, then repeat the simulation. Neglecting issues of convergence for the moment (these will be discussed in section 3.3.3), we will arrive after several iterations at a conjugate match between Z_s and Z_i.

Once the appropriate R_s and C_s are chosen and R_i and L_i calculated, a matching network can be inserted between R_s and C_s. From the point of view of the source, C_s cancels L_i and R_i is transformed to R_{s2}, the desired source impedance. Looking towards the source, the nonlinear network sees C_s in series with R_{s2} transformed by the matching network—the same values used to calculated L_i and R_i.

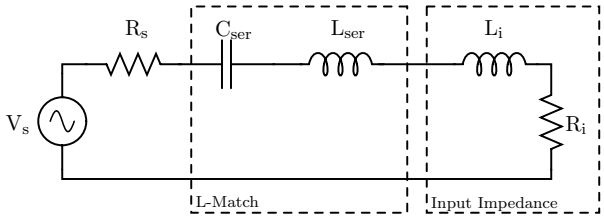
3.3.2 Improved Iterated Matching

While matching has now been achieved, this circuit is not optimal: instead of utilizing the input reactance of the nonlinear circuit, we are cancelling it and matching only the input resistance. Not only is this a waste of usable reactance, it is also likely an efficiency hazard: the use of series resonant cancellation causes large resonant currents as energy passes back and forth between inductance and capacitance, likely resulting in additional conduction loss.

This situation is easily remedied with judicious application of series-shunt transfor-



(a) Utilizing the input reactance of the nonlinear circuit in the matching network results in improved efficiency.



(b) Analysis of the new matching network is aided by series-shunt transformation of C_{shn} and R_{s2} into C_{ser} and R_s

Figure 3-3: An improved matching network for the circuit of Fig. 3-2(a).

mations. Figure 3-3(a) shows another possible matching configuration for the circuit of Fig. 3-2(a). This time, C_s is not explicitly present; nevertheless, the choice of appropriate values for L_{ser} and C_{shn} gives an equivalent C_s looking back from the input port of the nonlinear network while creating a matching network that utilizes the reactance of L_i. To calculate these values for a chosen R_{s2}, first transform Fig. 3-3(a) into a series network as pictured in Fig. 3-3(b).

$$R_s = R_i \tag{3.6}$$

$$Q_m = \sqrt{\frac{R_{s2}}{R_s} - 1} = \frac{1}{\omega R_s C_{ser}} = \omega R_{s2} C_{shn} \tag{3.7}$$

$$C_{shn} = C_{ser}\left(\frac{{Q_m}^2}{{Q_m}^2 + 1}\right) \tag{3.8}$$

$$L_{ser} = \frac{1}{\omega^2 C_{ser}} - L_i \tag{3.9}$$

Looking back from the input of the nonlinear network, we see a reactance

$$X_s = \omega L_{ser} - \frac{1}{\omega C_{ser}} \tag{3.10}$$

$$= -\omega L_i \tag{3.11}$$

$$= \frac{1}{\omega C_s} \tag{3.12}$$

so the nonlinear network sees the same conditions as in Fig. 3-2(a). On the other hand, viewed from the source, all the reactances cancel, leaving just R_i series-shunt transformed by the matching network. Thus, a conjugate match is achieved without unnecessary impedance cancellation.

The same method can be used to design a third-order network by first choosing the constraints and calculating the image resistance (R_{img} for the T- and Π-matches, or R_{ser} in the case of the tapped resonators). Design then proceeds by determining the half of the matching network that transforms R_i to the image resistance and then computing the remaining values to transform the image resistance to the desired source resistance.

3.3.3 Convergence of the Iterated Matching Procedure

A sufficient (though not necessary) condition for convergence is monotonicity of the variation of input impedance magnitude and phase with source impedance. That is, the slopes of the source and input impedances are directly or inversely related such that a change in source impedance always produces a change in the same direction (though not necessarily of the same magnitude) in the input impedance. The rectifiers explored in chapters 4 and 5 were empirically determined to have this property.

A more general requirement is that input impedance be a fixed, deterministic function of source impedance and power throughput. Circuits where there are several switches with independently-determined operation such that there are multiple switching configurations

that can lead to the same behavior (such as the polyphase rectifiers analyzed in [36]) are likely to be problematic. Of course, many multi-switch circuits do not have this property: a series diode rectifier with a shunt flyback diode enforces a well-known relationship between the switching behavior of the two active devices.

On the whole, it is overwhelming likely that any network consisting of only one nonlinear element will converge when using this method. In circuits with multiple switches but where state transitions depend on a common variable this method is also likely to succeed. For more exotic cases, convergence may depend on the initial impedance guess; in such cases, Fourier analysis as in chapter 4 could be used in conjunction with this technique to produce an initial guess from which iteration would likely converge on an accurate value.

3.4 SAMwIICh: An Automated Matching Implementation

For the purposes of design in the proceeding chapters and to verify the usefulness of iterated impedance matching, a fully automated matching system named *Software for Automated Matching with Iterated Impedance Calculations*[3] (SAMwIICh) was developed. Since SAMwIICh was developed expressly for use on the circuits discussed later, it is not completely general: it assumes that the input impedance is inductive, since in all cases of interest this happened to be true; this is, of course, easily generalized. Full code is provided in appendix A.

SAMwIICh uses the HSPICE circuit analysis package to simulate circuits and do FFT analysis [37]. First, it provides an initial guess at source impedance; it then performs the initial simulation run. From FFT data, the input impedance and its conjugate are calculated, and the appropriate circuit elements are modified. Iteration continues until the input impedance is matched to within the specified tolerance.

For single-diode rectifiers and 0.1% tolerance, fewer than ten iterations were necessary in almost all cases. In the circuits tested, no stable limit cycles were encountered; convergence

[3]Thanks to Jim Paris for his help in deciding on a name and for suggestions on how *not* to reinvent the wheel.

proceeded smoothly from the initial guess to the final value.

One possible modification to SAMwIICh to speed convergence would be slope extrapolation: given the values of one or more previous iterations, guess at the final value. This was not implemented because it makes the system more susceptible to the effects of non-monotonicity; correcting this would require code to detect oscillations and reduce step size or even turn off prediction entirely. Moreover, the observed convergence behavior did not seem to warrant this modification.

Chapter 4

The Series Resonant Rectifier

THE FIRST topology to be considered is the *series resonant* rectifier [19, 20, 38], pictured in Fig. 4-1. Discussion of an analytic model of rectifier operation, including Fourier analysis of input impedance, is followed by verification of the model by simulation of a demonstrative design. The strengths and weaknesses of this topology as a candidate for employment in the DC-DC converter architecture described in section 1.1.2 are then considered. Finally, an experimental implementation is examined.

Both of the rectifiers considered, in this chapter and in the next, employ only one diode. While multi-diode rectifiers for RF operation have been proposed [14, 15, 19], the performance of the rectifiers examined was high enough that the inclusion of additional diodes seemed unnecessary and, indeed, would likely decrease efficiency. Moreover, analysis of these circuits is simpler than those involving multiple diodes, and problems of convergence when using SAMwIICh (section 3.4) for computational analysis are seemingly nonexistent with single-diode rectifiers.

4.1 Analysis of the Series Resonant Rectifier

A detailed derivation of the analytic model of the rectifier would be voluminous and redundant; those interested in such a derivation should consult [38]. However, it will be useful for later discussion to note Kazimierczuk and Czarkowski's results. For the purposes of

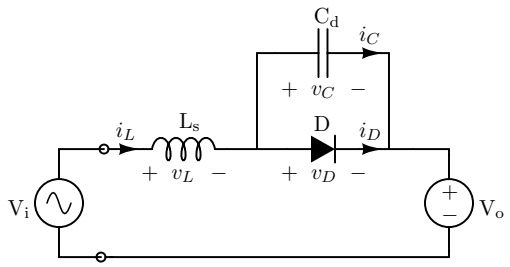


Figure 4-1: The series resonant rectifier.

the derivation, an ideal diode with a constant parallel capacitance is assumed. In addition, output ripple is neglected, and the circuit is assumed to be driven at the resonant frequency $\frac{1}{\sqrt{\mathrm{L_s C_d}}}$. For ease of derivation, the diode is assumed to be off $0 \leq \omega_0 t \leq 2\pi\,(1 - \mathfrak{D})$; ϕ defines the phase relationship between the input voltage and the state of the diode.

$$v_I = \mathrm{V_i} \sin(\omega_0 t + \phi) \tag{4.1}$$

It is useful to define

$$\mathrm{V_{rms}} = \frac{\mathrm{V_i}}{\sqrt{2}} \tag{4.2}$$

$$\mathrm{M_{VR}} = \frac{\mathrm{V_o}}{\mathrm{V_{RMS}}} \tag{4.3}$$

$$i_O = i_D + i_C \tag{4.4}$$

$$\mathrm{R_L} = \frac{\mathrm{V_o}}{\langle i_O \rangle} \tag{4.5}$$

Thus,

$$\mathrm{Q} = \frac{\mathrm{R_L}}{\omega_0 \mathrm{L_s}} = \omega_0 \mathrm{C_d R_L} \tag{4.6}$$

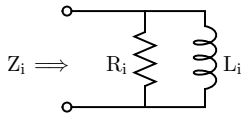


Figure 4-2: A parallel RL model for series rectifier input impedance.

For $0 \leq \omega_0 t \leq 2\pi\,(1-\mathfrak{D})$,

$$i_L = i_C = \frac{\langle i_O \rangle \mathrm{Q}}{\sqrt{2}\mathrm{M_{VR}}}\left[\omega_0 t \sin(\omega_0 t+\phi) + \left(\sin(\phi) - \mathrm{M_{VR}}\sqrt{2}\right)\sin(\omega_0 t)\right] \tag{4.7}$$

$$v_C = v_D = \mathrm{V_o}\left[\frac{\cos(\phi)\sin(\omega_0 t) - \omega_0 t\cos(\omega_0 t+\phi)}{\mathrm{M_{VR}}\sqrt{2}} + \cos(\omega_0 t) - 1\right] \tag{4.8}$$

$$v_L = \frac{\mathrm{V_o}\sqrt{2}}{\mathrm{M_{VR}}}\left[\frac{\omega_0 t\cos(\omega_0 t+\phi)}{2} + \frac{cos(\phi)\sin(\omega_0 t)}{2} + \left(sin(\phi) - \frac{\mathrm{M_{VR}}}{\sqrt{2}}\right)\cos(\omega_0 t)\right] \tag{4.9}$$

The diode conducts during the interval $2\pi\,(1-\mathfrak{D}) < \omega_0 t \leq 2\pi$. During this time,

$$v_L = \mathrm{V_o}\left[\frac{\sqrt{2}\sin(\omega_0 t+\phi)}{\mathrm{M_{VR}}} - 1\right] \tag{4.10}$$

$$i_D = \frac{\langle i_O \rangle \mathrm{Q}\sqrt{2}}{\mathrm{M_{VR}}}\left[\cos(\phi) - \cos(\omega_0 t+\phi) - \frac{\mathrm{M_{VR}}\,(\omega_0 t - 2\pi)}{\sqrt{2}}\right] \tag{4.11}$$

4.2 Fourier Analysis of Input Impedance

As argued in section 3.2, the condition of sinusoidal drive means that power is only transferred at the fundamental drive frequency; this means that we can once again construct a linearized input impedance model. To model the series resonant rectifier's input impedance, we must find the fundamental component of input current. We can do this by applying the Fourier integral [25] to equations 4.7 and 4.11; once again, this derivation closely follows that of [38].

Expressing the input impedance in terms of a parllel RL circuit as in Fig. 4-2 allows us to quickly identify the resistive component of input current. If we assume the network

is lossless, then the resistance of the parallel RL model must be the output impedance reflected through the ratio of input to output voltages. That is,

$$\mathrm{R_i} = \frac{\mathrm{R_o}}{\mathrm{M_{VR}}^2} \tag{4.12}$$

If the input voltage $v_{IN} = \mathrm{V_i}\sin(\omega_0 t + \phi)$, then

$$i_{IN} = \frac{\mathrm{V_i}}{\mathrm{R_i}}\sin(\omega_0 t + \phi) - \mathrm{I_{Lin}}\cos(\omega_0 t + \phi) \tag{4.13}$$

We can then find the current through the inductor by computing

$$\mathrm{I_{Lin}} = -\frac{1}{\pi}\int_0^{2\pi} i_L \cos(\omega_0 t + \phi) d(\omega_0 t) \tag{4.14}$$

$$= \frac{\mathrm{V_i}}{\omega_0 \mathrm{L_i}} = \frac{1}{\omega_0 \mathrm{L_i}}\frac{\mathrm{V_o}\sqrt{2}}{\mathrm{M_{VR}}} = \frac{\mathrm{R_o}}{\omega_0 \mathrm{L_i}}\frac{\langle i_O\rangle\sqrt{2}}{\mathrm{M_{VR}}} = \mathrm{Q}\frac{\mathrm{L_s}}{\mathrm{L_i}}\frac{\langle i_O\rangle\sqrt{2}}{\mathrm{M_{VR}}} \tag{4.15}$$

After a substantial amount of math, we arrive at the ratio

$$\frac{\mathrm{L_s}}{\mathrm{L_i}} = \frac{1}{\pi}\left\{\frac{\pi(5\mathfrak{D}-1)}{4} - a\cdot\sin^2(\phi) - b\cdot\cos^2(\phi) + \right.$$
$$\left. + c\cdot\sin(\phi)\cos(\phi) + \frac{\mathrm{M_{VR}}}{\sqrt{2}}\left[d\cdot\sin(\phi) + e\cdot\cos(\phi)\right]\right\} \tag{4.16}$$

where

$$a = \left[\frac{3}{8}\cos(2\pi\mathfrak{D}) - \frac{\pi(1-\mathfrak{D})}{2}\sin(2\pi\mathfrak{D})\right]\sin(2\pi\mathfrak{D}) - \frac{\pi(1-\mathfrak{D})}{2} \tag{4.17}$$

$$b = \sin(2\pi\mathfrak{D}) - \frac{5}{16}\sin(4\pi\mathfrak{D}) - \frac{\pi(1-\mathfrak{D})}{2}\cos^2(2\pi\mathfrak{D}) \tag{4.18}$$

$$c = \left[1-\cos(2\pi\mathfrak{D})\right]\cos(2\pi\mathfrak{D}) + \frac{\pi(1-\mathfrak{D})}{2}\sin(4\pi\mathfrak{D}) \tag{4.19}$$

$$d = 2\pi\mathfrak{D}\cos(2\pi\mathfrak{D}) - \pi(1-\mathfrak{D}) - \left[1 + \frac{\cos(2\pi\mathfrak{D})}{2}\right]\sin(2\pi\mathfrak{D}) \tag{4.20}$$

$$e = 1 - \cos(2\pi\mathfrak{D}) + \left[\frac{\sin(2\pi\mathfrak{D})}{2} - 2\pi\mathfrak{D}\right]\sin(2\pi\mathfrak{D}) \tag{4.21}$$

As mentioned previously, this derivation makes two significant assumptions: first, that the diode is ideal, and second, that the capacitance across the diode is constant over a cycle. At radio frequencies, it is unlikely that we would put additional capacitance in parallel with the diode, electing instead to use the junction capacitance of the diode to provide C_d. Recalling from section 2.3 that diode junction capacitance is a strong function of voltage, it is clear that some care is needed in choosing the appropriate capacitance value.

In order to make use of our analysis with a nonlinear capacitance, we will assume that we must replace the linear capacitor with a nonlinear capacitance that stores the same average energy each cycle. Preserving the average stored energy keeps the resonant operation closest to the behavior predicted by the model, since it preserves average inductor current, thus preserving average output current. Since we have fixed the output voltage of the circuit for the purpose of analysis, preserving average output current keeps output power constant. Moreover, as power is only transferred at the fundamental frequency, keeping output power constant means that the input characteristics at the driving frequency are unchanged. Thus, macroscopic operation of the circuit is preserved[1].

To calculate a first-order estimate of the equivalent linear capacitance for a given diode, the voltage waveform from equation 4.8 is applied to the capacitance function in equation 2.61. This produces capacitance as a function of time, from which energy is quickly calculated:

$$E(t) = \frac{C(t)\,[V(t)]^2}{2} \tag{4.22}$$

The equivalent capacitance is then easily found:

$$C_{equiv} = \frac{\langle E(t)\rangle}{\langle [V(t)]^2\rangle} \tag{4.23}$$

Note that this derivation assumes that the addition of a nonlinear capacitance does not change the voltage waveform across the diode. While this assumption is not completely

[1]Further discussion of the use of nonlinear capacitance in resonant radio frequency circuits can be found in [39, 40].

accurate, it is nevertheless reasonable for a first-order estimate. Precise derivation of voltage waveforms can be found in [39–41].

4.3 Comparison of Analytic Model with Simulation

In order to determine the usefulness of the analytic model, a rectifier circuit will be designed by the process described in [38] and then simulated. Predicted input impedance and output power will be compared to simulated results.

Since the circuit parameters are easiest to determine at $\mathfrak{D} = .5$, we will choose this operating point for our test case. The rectifier will run at 100 MHz and provide 5 W into 5 V, meaning that $R_o = 5\ \Omega$. From [38] page 105, $\mathfrak{D} = .5$ gives

$$Q \approx .3884 \tag{4.24}$$

$$M_{VR} \approx .3684 \tag{4.25}$$

Thus,

$$V_i = \frac{V_o\sqrt{2}}{M_{VR}} = 19.19\ \text{V} \tag{4.26}$$

The necessary inductance and capacitance are

$$L_s = \frac{R_o}{\omega_0 Q} = 20.5\ \text{nH} \tag{4.27}$$

$$C_d = \frac{Q}{\omega_0 R_o} = 123.6\ \text{pF} \tag{4.28}$$

The analytic model predicts that

$$L_i = 16.42\ \text{nH} \tag{4.29}$$

$$R_i = 36.83\ \Omega \tag{4.30}$$

$$Q = 3.6 \tag{4.31}$$

Table 4.1: Input impedance, output power, and efficiency, calculated versus simulated.

Parameter	Calculated	Simulated, Ideal Diode	Simulated, Real Diode	Simulated, Conjugate Match
P_o	5 W	4.99 W	4.95 W	.66 W
R_i	36.83 Ω	38.86 Ω	33.5 Ω	3.78 Ω
L_i	16.42 nH	15.72 nH	16.02 nH	6.6 nH
η	100%	100%	89.2%	91.3%

or, for the series RL model,

$$L_{i,ser} = 15.12 \text{ nH} \tag{4.32}$$

$$R_{i,ser} = 2.64\ \Omega \tag{4.33}$$

Table 4.1 compares the computed results with those obtained in three HSPICE simulation runs (netlists for these simulations are available in appendix B.1). In the first simulation, an ideal diode in parallel with a linear capacitor was employed. In the second, the ideal diode was replaced by a real diode model, though still with a linear capacitance. Finally, in the third simulation, the rectifier was driven from a conjugate match. As expected, the input impedance of the rectifier changed substantially with the addition of source impedance, resulting in severely reduced power throughput. The slight increase in efficiency is a result of reduced conduction loss because of lower average output current; this is an illusory improvement, since we have sacrificed almost a factor of ten in output power in order to gain two percent "improvement" in efficiency.

The results of the third simulation highlight the need for a model including an arbitrary source impedance, which would allow the input impedance to be calculated for a conjugate match as described in section 3.2. This idea is unattractive for two reasons: first, the precision of the calculations for a range of real diodes is likely to be poor, and second, the effort involved in this calculation would only be useful for a single topology—parasitics would require explicit consideration and would complicate the model considerably. A more

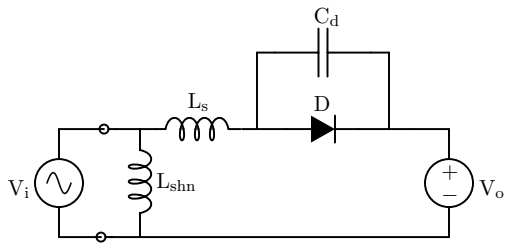


Figure 4-3: To provide a DC current path, all or part of L_s can be transformed into L_{shn}.

attractive alternative has already been suggested in section 3.3, to wit, the iterated computational approach. Design can proceed much more quickly without the need for special provisions for nonlinear capacitance, diode drop, and passive component loss, none of which the analytic model considers. Moreover, a computational approach can take into account additional parasitics (such as those present on a printed circuit board) with only the effort required to modify the circuit description file.

4.4 Application of the Series Resonant Rectifier

As demonstrated in section 4.3, the performance of the series resonant rectifier is good when applied under appropriate conditions: efficiencies approaching 90% were obtained using a realistic model of diode loss. As argued in section 4.2, the use of a nonlinear capacitor should not change circuit operation substantially; thus, the series rectifier is a design with significant practical potential.

One limitation of the resonant rectifier not yet discussed is that it must be driven from a low DC impedance. This may cause problems in some applications, e.g., when driven by a class-E inverter topology employing a series resonant output tank; in such cases, there is no path for DC current to flow to the output. This limitation can easily be overcome in practice by transforming all or part of L_s to its shunt equivalent as illustrated in Fig. 4-3; since there is now a path for DC output current through the shunt inductor, a source with high DC impedance can be used.

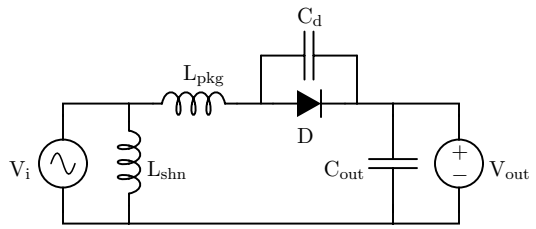


Figure 4-4: Experimental implementation of the series rectifier.

Since this topology involves resonant operation, the reverse voltage stress on the active device is quite high, requiring the use of diodes with high reverse voltage ratings to withstand the voltage developed on C_d each cycle. Going from a low voltage diode to one with a higher reverse voltage rating has some practical effects worth considering: in most cases, high reverse voltage rating is achieved with the addition of guard rings or with a metal-overlapped laterally-diffused process (see [41] and references therein). Both of these processes have the effect of reducing junction capacitance, since the diode is effectively placed in series with its guard structure. Unfortunately, these structures also tend to increase forward drop, reducing efficiency[2].

As an alternative to using high-voltage devices, topologies might be derived from the series resonant rectifier which exploit the multi-resonant structures designed and implemented in [42]. Using such techniques, resonant operation can be utilized with reduced voltage stress, since higher harmonic content is used to achieve square switching waveforms. Topologies of this nature would allow greater device utilization and enjoy enhanced efficiency, since lower voltage diodes (with lower forward drops) could be employed.

4.5 Experimental Implementation

The series resonant rectifier was implemented in the DC-DC converter in [10]. Fig. 4-4 shows the rectifier as implemented, with component values and part numbers listed in

[2]In some designs, the selection of a high voltage diode to obtain the junction capacitance prescribed by the analytic model could cause a reduction in efficiency. In this case, the computational model is of great assistance, since it may be applied to designs which utilize the increased capacitance of a lower voltage diode.

Table 4.2: Component values used in the series rectifier experimental implementation.

Circuit Element	Nominal Value	Part Number
D		MBRS1540T3
L_{pkg}	1 nH	Diode package parasitic
C_d	$C_{j0} = 318.5$ pF	Diode junction capacitance
L_{shn}	12.5 nH	Coilcraft A04TJ
C_{out}	9×47 nF	X7R Ceramic, 50 V

Table 4.2. Overall efficiency of the radio frequency power converter cell was greater than 75% from 16 V DC input to 5 V DC output. Rectification efficiency was around 85%; the discrepancy between experimental efficiency and that calculated in section 4.3 is likely a result of finite inductor Q.

As discussed in section 1.1.2, in addition to high efficiency and well-characterized input impedance, a high DC output impedance is extremely important to ensure proper current sharing between parallel converters. Simulation results (netlist provided in appendix B.2) suggests that output impedance ranges from 20 Ω at light load to 5 Ω at maximum load. In the experimental implementation, the output current of eight parallel cells under maximum load was matched to within ±6.5% of average.

Chapter 5

The Shunt Resonant Rectifier

THE SECOND topology considered is the *shunt resonant* rectifier [7, 18, 19]. This rectifier, pictured in Fig. 5-1, employs a diode in the shunt position; it is in essence the dual of the traditional class-E architecture [16, 43].

This chapter proceeds in much the same fashion as the previous one: an analytic model is developed and compared to simulation, practical concerns are discussed, and an experimental implementation is presented. In addition, the use of the coupled magnetic structures similar to those presented in [21, 22] is examined, accompanied by simulation and experimental results.

5.1 Analysis of the Shunt Resonant Rectifier

The shunt resonant rectifier is essentially the dual of the traditional class-E inverter. Stated more precisely, the shunt rectifier as analyzed here is the *bilateral inverse* of the class-

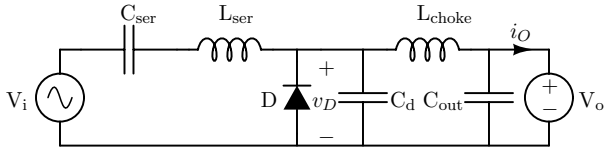


Figure 5-1: The shunt resonant rectifier.

E circuit [44]. Since this means that the rectifier operating waveforms are exactly time-reversed copies of the class-E waveforms, analysis of this circuit is easily obtained by slight modifications to the standard class-E inverter analysis. A particularly good analysis of the class-E inverter, and the one from which the following is derived, appears in [45].

As required by bilateral inversion, the rectifier duty cycle $\mathfrak{D} = (1 - \mathfrak{D}_{\text{inverter}})$. Hence, following [45], we define

$$v_I = \mathrm{V_i} \sin(\omega_0 t + \phi) \tag{5.1}$$

$$\tan(\phi) = \frac{\frac{\sin(y)}{y} - \cos(y)}{\frac{\xi y}{\pi}\cos(y) - \left(1 + \frac{\xi}{\pi}\right)\sin(y)} \tag{5.2}$$

$$y = \pi\mathfrak{D} \tag{5.3}$$

$$\xi = \frac{1}{\mathrm{V_o}} \frac{dv_D}{d\theta} \tag{5.4}$$

where the conduction angle y is centered at $\omega_0 t = \frac{\pi}{2}$, and ϕ defines the phase between the input voltage and the state of the diode. Assuming $\mathrm{L_{choke}}$ is large, the output current is approximately constant, so

$$\mathrm{I_o} = \langle i_O \rangle \approx i_O \tag{5.5}$$

$$\mathrm{P_o} = \mathrm{V_o I_o} \tag{5.6}$$

$$\mathrm{R_o} = \frac{\mathrm{V_o}}{\mathrm{I_o}} \tag{5.7}$$

$$\mathrm{r} = \frac{2\mathrm{R_o}}{g^2} \tag{5.8}$$

$$\mathrm{g} = \frac{y}{\cos(\phi)\sin(y)} \tag{5.9}$$

$$\mathrm{V_i} \approx \mathrm{V_o}\left(\frac{2}{g}\right) \tag{5.10}$$

The susceptance of $\mathrm{C_d}$ is given by

$$\mathrm{B} = \omega_0 \mathrm{C_d} = \frac{2\left(y^2 + yg\sin(\phi - y) - g\sin(\phi)\sin(y)\right)}{\pi g^2 r} \tag{5.11}$$

The input tank components, C_{ser} and L_{ser}, are given by

$$C_{ser} = \frac{1}{\omega_0 Q r} \tag{5.12}$$

$$L_{ser} = \frac{(Q + \tan(\psi))\, r}{\omega_0} \tag{5.13}$$

ψ represents the additional reactance angle which effectively cancels C_d, and is given by

$$\tan(\psi) = \frac{\omega_1 \sin(\phi) + \omega_2 \cos(\phi) + \omega_3 \cos(2\phi) + gy}{\omega_2 \sin(\phi) + \omega_3 \sin(2\phi) - \omega_1 \cos(\phi)} \tag{5.14}$$

where

$$\omega_1 = -2g \sin(\phi - y) \sin(y) - 2y \sin(y) \tag{5.15}$$

$$\omega_2 = 2y \cos(y) - 2 \sin(y) \tag{5.16}$$

$$\omega_3 = -g \sin(y) \cos(y) \tag{5.17}$$

These equations allow the design of a shunt rectifier for a given Q, ξ, $\mathfrak{D}$, V_o, and P_o. Whereas in the design of RF amplifiers, Q might be chosen for, e.g., output spectral purity, in this case Q is in practice generally chosen such that the input tank serves as the impedance matching network after a suitable series-shunt transformation of one of the tank elements.

The derivation of an analytic expression for input impedance is not presented, since it was shown in the previous chapter that such an expression is of little practical value. Were such an expression desirable, a procedure largely similar to the one in section 4.2 could be used. It is unlikely, however, that such a model would fare well in the face of nonidealities and unmodeled parasitics.

Table 5.1: Output power and efficiency, calculated versus simulated.

Parameter	Calculated	Simulated, Ideal Diode	Simulated, Real Diode
P_o	5 W	5 W	1.12 W
η	100%	100%	89.7%

5.2 Comparison of Analytic Model with Simulation

To verify that the analytic model derived in section 5.1 is valid, we will design a rectifier to the same specifications as in section 4.3: $V_o = 5$ V, $P_o = 5$ W, $\mathfrak{D} = .5$, $f_0 = 100$ MHz. We choose $\xi = 0$, since in practice optimal efficiency is obtained at this point [45], and arbitrarily select $Q = 5$. The design equations yield

$$\phi = -0.5669 \tag{5.18}$$

$$g = 1.862 \tag{5.19}$$

$$r = 2.884 \tag{5.20}$$

$$B = .06366 \tag{5.21}$$

$$\psi = .8561 \tag{5.22}$$

We then calculate

$$C_d = 101.32 \text{ pF} \tag{5.23}$$

$$C_{ser} = 110.4 \text{ pF} \tag{5.24}$$

$$L_{ser} = 28.24 \text{ nH} \tag{5.25}$$

$$V_i \approx 5.4 \text{ V} \tag{5.26}$$

Table 5.1 compares calculated values for output power and efficiency to simulation results with an ideal diode and with a diode model that accounts for forward drop but still has

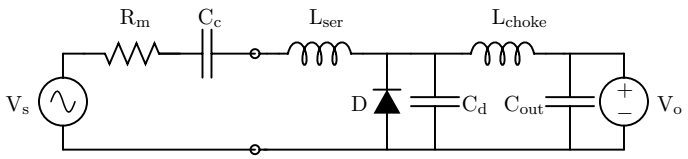
a constant junction capacitance (HSPICE netlists are provided in appendix B.3). While the ideal diode gives reasonable results, the use of the more realistic diode model is problematic. This likely results from the fact that the diode forward drop is considerable compared to the difference between V_i and V_o. Once again, the value of a computational approach is highlighted by the shortcomings of the analytic model.

5.3 Application of the Shunt Resonant Rectifier

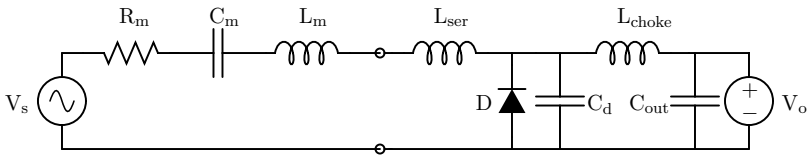
Unlike the series resonant rectifier, the shunt resonant rectifier provides its own DC current path. Hence, it is tolerant of a range of input impedances without modification. As mentioned earlier, however, shunt transformation of one of the elements of the input tank is likely still desirable, since the input tank can then be used to interface the source and input impedances. When using the input tank elements in this way, the designer no longer has a choice of Q, since it is fixed by the ratio of source to load impedance.

The analysis of section 5.1 makes the assumption that a "large" choke inductor is used and ignores the choke inductor thereafter. In practice, the size of the choke inductor is dictated by ripple requirements, and it is generally desirable to make it as small as possible for a given application. While the analytic model is tolerant of some amount of ripple, it is unlikely that model accuracy will be good with a current ripple that is greater than 10% of DC output current; in such cases, the computational approach is certain to be preferable.

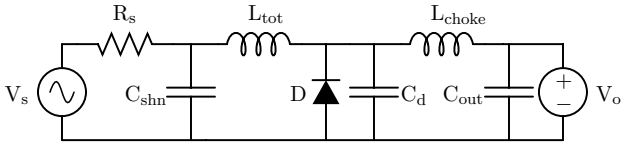
Also neglected in the foregoing analysis is any package inductance appearing in series with the diode. It is convenient when considering the package inductance to break the circuit into two loops: the AC loop, including the input tank and C_d, and the DC loop, containing the diode, L_{choke}, and the load. Practically, the addition of an inductance in series with the diode has two effects: first, it appears in series with C_d when the diode is off, adding with L_{ser} and resulting in a larger effective tank inductance; second, since it appears in both the AC and DC loops, it has the effect of coupling additional ripple into the output. Section 5.5 explores the latter effect in detail, including its mitigation through the use of coupled magnetic structures of the kind described in [21, 22].



(a) First, the input tank is replaced by $L_{ser} \approx \frac{2}{\omega_0{}^2 C_d}$, and R_m and C_c are determined by iterated computation.



(b) Next, C_c is replaced with a series transformed L-match comprising C_m and L_m.



(c) Finally, R_m and C_m are shunt transformed and L_m and L_{ser} are combined into L_{tot}.

Figure 5-2: The computationally-assisted design process for the shunt resonant rectifier.

When using iterated computation to design the shunt resonant rectifier, it is convenient to replace the input tank with only an inductance having a reactance approximately twice that of C_d; after computing the matching parameters, the matching network serves as the input tank. Figure 5-2 illustrates the procedure for computing the appropriate matching network values. First, determine R_m and C_c by iteration such that they form a conjugate match with the rectifier circuit (Fig. 5-2(a)). Next, following the procedure outlined in section 3.3.2, an L-match which has been series transformed can be inserted in the place of C_c (Fig. 5-2(b)). The final design is reached by shunt transforming C_m into C_{shn} and R_m into R_s, and combining L_m and L_{ser} into L_{tot} (Fig. 5-2(c)).

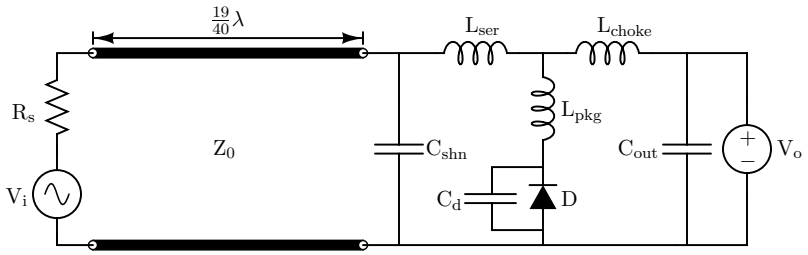


Figure 5-3: Experimental implementation of the shunt rectifier.

Table 5.2: Component values used in the shunt rectifier experimental implementation.

Circuit Element	Nominal Value	Part Number
D		20CJQ060
L_{pkg}	2 nH	Diode package parasitic
C_d	C_{j0} = 145 pF	Diode junction capacitance
C_{shn}	56 pF	CDE MC12FA560J
L_{ser}	39 nH	Coilcraft 1812SMS-39NJ
L_{choke}	120 nH	Coilcraft 1812SMS-R12J
C_{out}	9×47 nF	X7R Ceramic, 50 V
R_s	50 Ω	Amplifier output impedance
Z_0	51.5 Ω	Belkin 8240 RG-58A/U coax, 95 cm

5.4 Experimental Implementation

To verify the results of section 5.1 and the application of iterative matching as described in section 5.3, a test rectifier was implemented. The rectifier provides 3.3 W into 5 V from a 100 MHz sinusoidal excitation. Radio frequency input was provided by a General Radio Company Model 1363 VHF Oscillator with a Model 1264-B Modulating Power Supply[1] driving an Amplifier Research Model 10W1000 power amplifier [47], and the rectifier was loaded with fifteen 1 W, 5.1 V Zener diodes in parallel with a bulk capacitance comprising two 15 μF tantalum capacitors. Figure 5-3 illustrates the rectifier implementation, with component values listed in Table 5.2. The transmission line at the input of the rectifier

[1]Due to the age of the General Radio equipment and the circumstances under which this unit was obtained (I found it abandoned in a hallway!), no manual or other reference is available. It is worth noting that, in addition to producing all manner of RF signal and noise generators, the General Radio Company invented the Variac variable autotransformer. More information on this very interesting company can be found in [46].

models the coaxial cable used to connect it to the amplifier.

An HSPICE netlist of the rectifier circuit is provided in appendix B.4. Simulated and experimental results are well matched: the rectifier provided 3.32 W at 4.97 V with nearly 80% efficiency.

5.5 Parasitic Mitigation in the Shunt Rectifier

In the series resonant rectifier, both of the principal parasitics of the diode (C_j and L_{pkg}) are employed by the topology. While the shunt rectifier still uses C_d, L_{pkg} is not present in the ideal circuit representation, and it is ignored by the analytic development in section 5.1.

As briefly mentioned in section 5.3, L_{pkg} has two main effects: first, it adds to L_{ser} and acts as part of the AC loop while the diode is off. When the diode is conducting, L_{pkg} appears in the DC loop, which has the additional effect of coupling additional AC current into the output, increasing output voltage ripple. Techniques have been described in [21, 22] that can be used to move inductance from one branch to two adjacent branches. This is achieved by the use of coupled magnetic structures which produce an equivalent T network having negative inductance in one branch.

For a design of the kind presented in section 5.4, a center tapped magnetic structure is a good choice, since it produces more inductance cancellation for a given size than an end tapped structure [21, 22]. Thus, we will consider exclusively the use of center tapped magnetics. In a center tapped structure, the two physical inductances L_1 and L_2 appear in the positive branches of the equivalent T network. The negative branch of the network has a value equal to $-L_M$, the negative mutual inductance between L_1 and L_2.

Application of coupled structures to the shunt rectifier results in the circuit depicted in Fig. 5-4. The magnetic structure is designed such that the mutual inductance cancels all or part of L_{pkg}, while L_1 and L_2 add to L_{ser} and L_{choke}, respectively. In some designs, one or both of the latter can be entirely contained in the transformer structure. The use of this technique allows the designer to reverse the effect of L_{pkg} in coupling the AC and DC loops and eliminate consequent increases in output ripple.

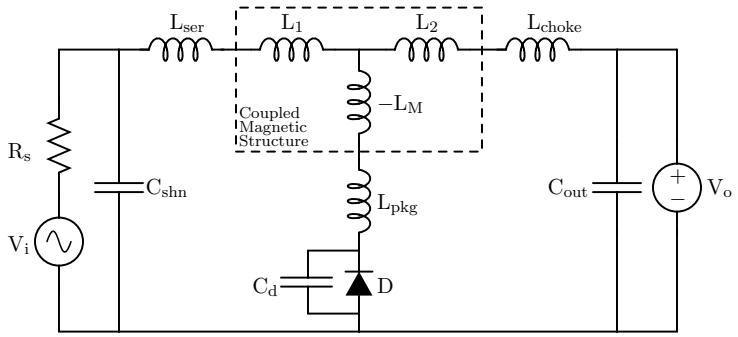


Figure 5-4: The shunt resonant rectifier with coupled magnetics for parasitic mitigation.

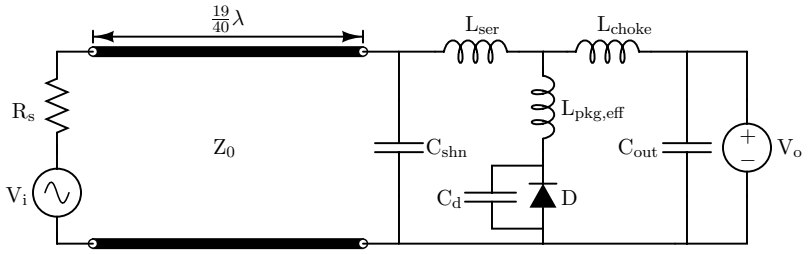


Figure 5-5: Experimental implementation of the shunt rectifier with parasitic mitigation.

While it seems that complete cancellation of L_{pkg} should result in minimum ripple at the output, simulation and experimentation both show that there is a particular (nonzero) value for which output ripple is minimized. This minimization is a result of higher harmonics generated by the interaction between the series combination $L_{pkg} - L_M = L_{pkg,eff}$ and C_d. For the optimal value of $L_{pkg,eff}$, these harmonics are phased such that peak-to-peak AC ripple is minimized.

To test the effect of inductance cancellation, several rectifiers incorporating varying amounts of cancellation were simulated, designed, and built. For comparison, a matching rectifier design without inductance cancellation was also built. All rectifiers were designed to provide 8 W into 5 V from a 100 MHz RF source. The rectifiers were driven and loaded in precisely the same manner as in section 5.4. Figure 5-5 shows the experimental rectifier circuit; component values are listed in Table 5.3. In all cases, coupled magnetic structures

Table 5.3: Component values used in the experimental implementation of the shunt rectifier with parasitic mitigation.

Element	$L_M = 0$	$L_M = 3.5$ nH	$L_M = 4.0$ nH	$L_M = 4.6$ nH	$L_M = 6.3$ nH
D	1N5822				
$L_{pkg,eff}$	6.5 nH	3 nH	2.5 nH	1.9 nH	0.2 nH
C_d	$C_{j0} = 548$ pF				
C_{shn}	138 pF (82 pF and 56 pF) CDE MC12FA560J and CDE MC12FA820J				
L_{ser}	12.5 nH Coilcraft A04TJ	13.5 nH			16.1 nH
		L_1 of coupled structure			
L_{choke}	120 nH Coilcraft 1812SMS-R12J	135 nH			140 nH
		Coilcraft 1812SMS-R12J and L_2 of coupled structure			
C_{out}	9×47 nF X7R Ceramic, 50V				
R_s	50 Ω amplifier output impedance				
Z_0	51.5 Ω Belkin 8240 RG-58A/U coax, 95 cm				

Table 5.4: Experimental results for shunt rectifiers with parasitic mitigation.

Parameter	No Cancellation	$L_M = 3.5$ nH	$L_M = 4.0$ nH	$L_M = 4.6$ nH	$L_M = 6.3$ nH
P_o	8 W				
$L_{pkg,eff}$	6.5 nH	3 nH	2.5 nH	1.9 nH	0.2 nH
$V_{rip,pp}$, measured	38 mV	18 mV	16 mV	20 mV	27 mV
$V_{rip,pp}$, adjusted	38 mV	20 mV	18 mV	22 mV	32 mV

were implemented on printed circuit board traces as described in [22]. The scripts used to generate the circuit board are available in appendix C.

Simulation results suggested that $L_{pkg,eff} = 2.5$ nH should give the greatest reduction in output ripple (an HSPICE netlist for these simulations can be found in appendix B.5). As evidenced by Table 5.4, experimental findings agree with the results of simulation. Note that L_{choke} was larger for the cases with inductance cancellation, since L_2 from the coupled magnetic structure appeared in series with the Coilcraft 1812SMS-R12J. Thus, adjusted ripple figures which account for the discrepancy in choke inductance are listed along with measured values in Table 5.4.

In the best case, $L_{pkg,eff} = 2.5$ nH, L_{choke} was reduced to a total of 54 nH (including L_2) while maintaining ripple performance comparable to the uncancelled case. This strongly

suggests that the application of coupled magnetic structures may in many cases be of great benefit in terms of total required filter size.

An effect of using coupled magnetic structures not considered before implementation was an increase in EMI susceptibility. In particular, all three cases in which inductance cancellation were employed displayed a marked increase in ripple when pointed broadside to a strong source of EMI at ω_0 (in this case, the power amplifier used to drive the test circuits). Ripple figures in Table 5.4 were measured with the circuit raking the EMI source (that is, the principal component of flux linked by the PCB inductors was perpendicular to the flux emitted by the EMI source).

It is likely that this enhanced EMI susceptibility arises as a result of the much greater enclosed area of the PCB inductors compared to the Coilcraft inductors employed in the rectifier without inductance cancellation. While the diameter of the Coilcraft A04TJ is approximately 3.1 mm, the PCB inductors had diameters of 7.1 mm, 7.6 mm, and 8.1 mm (L_M = 3.5 nH, 4.0 nH, and 4.6 nH, respectively). Another likely factor is that the Coilcraft inductor was located substantially closer to a ground plane than the inductors implemented in PCB traces (appendix D gives detailed PCB layouts for both the cancelled and non-cancelled cases). This effect is by no means a fundamental problem with inductors fabricated on a PCB. In fact, the implementation of multilayer toroidal PCB magnetics or self-shielded components as in [42] would likely prove a substantial boon to those wishing to manufacture integrated passive components using PCB fabrication techniques.

Chapter 6

Conclusion

6.1 Thesis Summary

THE PRINCIPAL effort of this thesis is the development of design and characterization methods for rectifier topologies amenable to use in the new architectures proposed in [10]. A computational design approach allowing fast and accurate circuit analysis and synthesis is developed and applied, along with traditional analysis, to two demonstrative rectifier topologies. In addition, the application of coupled magnetic structures for parasitic mitigation is considered. Experimental implementations are investigated to verify analytic and computational results.

This thesis is divided into six chapters, including this conclusion.

Chapter 1 introduces background material and presents motivations for the development of new architectures for DC-DC power conversion.

Chapter 2 reviews analytic techniques useful in linear resonant circuit analysis, and demonstrates the application of these techniques to the analysis and design of resonant impedance transformers.

Chapter 3 introduces the use of computational modeling for nonlinear circuit analysis. In this chapter, algorithms are developed that allow for extremely fast design of rectifier circuits and which pave the way for the analyses performed in the proceeding chapters.

Chapter 4 discusses the first of two resonant rectifier topologies, exploring the design and implementation process of the series resonant rectifier. Chapter 4 comprises a discussion of analytic models including Fourier and describing function analysis of input impedance, a comparison of the analytic models with simulation, and a discussion of the strengths and weaknesses of the series resonant topology. Experimental results are also discussed.

Chapter 5 covers another topology, the shunt resonant rectifier. In addition to a discussion of analytic models and simulated results, the exploration of the shunt topology is extended by application of coupled magnetic structures in the rectifier, allowing the mitigation of undesired parasitic effects. Implementations of rectifiers with and without parasitic mitigation are presented.

6.2 Thesis Conclusions

There are two main conclusions that can be drawn from this thesis. First, while analytic models of nonlinear resonant circuits are useful for developing an understanding of circuit operation, the crude models generally discussed are wholly inadequate for practical designs. Moreover, the effort involved in extending these models to account for nonidealities is so great as to motivate the search for an alternate approach. The computational design approach presented herein fulfills all practical design requirements while allowing the designer to proceed from specification to synthesis in a fast and accurate manner. It is anticipated that the development of more general computational approaches will allow even greater speed and accuracy for large classes of nonlinear resonant circuits.

The second conclusion is that the application of integrated, batch fabricated passive components, both to replace existing passives and to enhance circuit performance through, e.g., parasitic mitigation, has great potential. Substantial cost reduction can be realized through the elimination of discrete passives in favor of components integrated on printed circuit boards. In addition, circuit performance can be enhanced and passive size requirement reduced through the judicious application of integrated magnetic and resonant structures.

6.3 Recommendations for Future Work

Two principal improvements to the work here should be investigated. First, generalization of the computational techniques presented in chapter 3 is needed. Armed with such techniques, designers could realize a range of circuit topologies with great speed and accuracy. Moreover, the development of these techniques in parallel with similar methods for the design of integrated passive components will allow power converter technologies to take advantage of the benefits of batch fabrication.

In all simulated and experimental circuits presented, diode conduction loss was the dominant mechanism contributing to degraded efficiency. To circumvent this loss, investigation of synchronous rectification techniques is necessary. Such techniques would likely resemble the shunt rectifier implementation of chapter 5, since this allows the use of a ground-referenced switch. With the use of self-resonant gate drives of the type presented in [10], the gating losses inherent in synchronous rectifier designs could be made insignificant compared to the conduction loss savings realized by the elimination of diode forward drop.

The new architectures presented in [10], combined with accurate computational modeling and highly integrated passive components, promise dramatic increases in performance and, simultaneously, radical reductions in the cost of DC-DC power conversion systems.

Bibliography

[1] International Rectifier, *IRFD110 HEXFET Power MOSFET*.

[2] K. Watanabe, S. Takeishi, I. Norigoe, and R. Hiramatsu, "Self running converter utilizing partial resonance," in *10^{th} International Telecommunications Energy Conference Proceedings*, pp. 186–193, 1988.

[3] D. Maksimović, "A MOS gate drive with resonant transitions," in *22^{nd} Annual IEEE Power Electronics Specialists Conference Proceedings*, pp. 527–532, 1991.

[4] K. Yao and F. C. Lee, "A novel resonant gate driver for high frequency synchronous buck converters," *IEEE Transactions on Power Electronics*, vol. 17, no. 2, pp. 180–186, March 2002.

[5] D. M. Van de Sype, A. P. M. Van den Bossche, J. Maes, and J. A. Melkebeek, "Gate-drive circuit for zero-voltage-switching half- and full-bridge converters," *IEEE Transactions on Industry Applications*, vol. 38, no. 5, pp. 1380–1388, Sept./Oct. 2002.

[6] I. D. de Vries, "A resonant power MOSFET/IGBT gate driver," in *17^{th} Annual IEEE Applied Power Electronics Conference Proceedings*, vol. 1, pp. 179–185, 2002.

[7] R. J. Gutmann, "Application of RF circuit design principles to distributed power converters," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. IEC127, no. 3, pp. 156–164, 1980.

[8] M. M. Jovanovic, W. A. Tabisz, and F. C. Lee, "Zero-voltage-switching technique in high-frequency off-line converters," in *Third Annual IEEE Applied Power Electronics Conference and Exposition Proceedings*, pp. 23–32, 1988.

[9] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics,*, ch. 14,15. Norwell, MA: Kluwer Academic Publishers, 2000.

[10] J. M. Rivas, R. S. Wahby, J. S. Shafran, and D. J. Perreault, "New architectures for radio-frequency dc/dc power conversion," in *35^{th} Annual Power Electronics Specialists Sonference Proceedings*, 2004.

[11] D. J. Perreault, *Design and Evaluation of Cellular Power Converter Architectures*. PhD thesis, Massachusetts Institute of Technology, June 1997.

[12] D. J. Perreault, R. L. Selders, and J. G. Kassakian, "Frequency-based current sharing techniques for paralleled power converters," *IEEE Transactions on Power Electronics*, vol. 13, no. 4, pp. 626–634, July 1998.

[13] S. Luo, Z. Ye, R. L. Lin, and F. C. Lee, "A classification and evaluation of paralleling methods for power supply modules," in *30th Annual IEEE Power Electronics Specialists Conference Proceedings*, pp. 901–908, June 1999.

[14] D. C. Hamill, "Half bridge class DE rectifier," *IEEE Electronics Letters*, vol. 31, pp. 1885–1886, October 1995.

[15] D. C. Hamill, "Class DE inverters and rectifiers for DC-DC conversion," in *27th Annual IEEE Power Electronics Specialists Conference Proceedings*, pp. 854–860, June 1996.

[16] N. O. Sokal and A. D. Sokal, "Class E—a new class of high-efficiency tuned single-ended switching power amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 10, pp. 168–176, June 1975.

[17] H. L. Kraus, C. W. Bostian, and F. H. Raab, *Solid State Radio Engineering*. New York: John Wiley and Sons, 1980.

[18] R. J. Gutmann and J. M. Borrego, "Power combining in an array of microwave power rectifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-27, no. 12, pp. 958–968, Dec. 1979.

[19] W. A. Nitz, W. C. Bowman, F. T. Dickens, F. M. Magalhaes, W. Strauss, W. B. Suiter, and N. G. Zeisse, "A new family of resonant rectifier circuits for high frequency DC-DC converter applications," in *Third Annual Applied Power Electronics Conference Proceedings*, pp. 12–22, 1988.

[20] A. Ivascu, M. K. Kazimierczuk, and S. Birca-Galateanu, "Class E resonant low dv/dt rectifier," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, pp. 604–613, August 1992.

[21] T. C. Neugebauer, J. W. Phinney, and D. J. Perreault, "Filters and components with inductance cancellation," *IEEE Transaction on Industry Applications*, vol. 40, pp. 483–491, March–April 2004.

[22] T. C. Neugebauer and D. J. Perreault, "Filters with inductance cancellation using printed circuit board transformers," *IEEE Transactions on Power Electronics*, vol. 19, pp. 591–602, May 2004.

[23] F. E. Terman, *Electronic and Radio Engineering*. New York: M^{c}Graw-Hill, 4th ed., 1955.

[24] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge, UK: Cambridge University Press, 2nd ed., 2004.

[25] E. A. Guillemin, *The Mathematics of Circuit Analysis*. Cambridge, MA: The MIT Press, 1949.

[26] K. Lundberg, *Becoming One with The Transistor*. Cambridge, MA: MIT Department of Electrical Engineering and Computer Science, 2004.

[27] E. W. Weisstein, "Lambert W-function." From *Mathworld*—A Wolfram Web Resource. http://mathworld.wolfram.com/LambertW-Function.html.

[28] T. C. Neugebauer, *Advanced Filters and Components for Power Applications*. PhD thesis, Massachusetts Institute of Technology, May 2004.

[29] P. E. Gray and C. L. Searle, *Electronic Principles: Physics, Models, and Circuits*. New York: John Wiley and Sons, 1969.

[30] F. M. Gardner, *Phaselock Techniques*. New York: John Wiley and Sons, 2nd ed., 1979.

[31] J. W. Phinney, "Filters with active tuning for power applications," Master's thesis, Massachusetts Institute of Technology, May 2001.

[32] T. H. Lee, K. S. Donnelly, J. T. C. Ho, J. Zerbe, M. G. Johnson, and T. Ishikawa, "A 2.5 V CMOS delay-locked loop for 18 Mbit, 500 megabyte/s DRAM," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1491–1496, 1994.

[33] G. J. Thaler and M. P. Pastel, *Analysis and Design of Nonlinear Feedback Control Systems*. New York: McGraw-Hill, 1962.

[34] J. K. Roberge, *Operational Amplifiers: Theory and Practice*. New York: John Wiley and Sons, 1975.

[35] J. W. Cooley and J. W. Tukey, "An algorithm for machine calculation of complex Fourier series.," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, April 1965.

[36] J. W. Holloway, "Harmonic control of multiple-stator induction machines for voltage regulation," Master's thesis, Massachusetts Institute of Technology, May 2004.

[37] Avant! Corporation, *Avant! Star-Hspice Manual*.

[38] M. K. Kazimierczuk and D. Czarkowski, *Resonant Power Converters*. New York: John Wiley and Sons, 1995.

[39] P. Alinikula, K. Choi, and S. I. Long, "Design of class E power amplifier with nonlinear parasitic output capacitance," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 114–119, 1999.

[40] M. J. Chudobiak, "The use of parasitic nonlinear capacitors in class-E amplifiers," *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, vol. 41, no. 12, pp. 941–944, 1994.

[41] A. F. Goldberg and J. G. Kassakian, "The application of power MOSFETs at 100MHz," in *16th Annual IEEE Power Electronics Specialists Conference Proceedings*, pp. 91–100, 1985.

[42] J. W. Phinney, J. H. Lang, and D. J. Perreault, "Multi-resonant microfabricated inductors and transformers," in *35th Annual Power Electronics Specialists Conference Proceedings*, June 2004.

[43] M. Kazimierczuk and K. Puczko, "Exact analysis of class E tuned power amplifier at any Q and switch duty cycle," *IEEE transactions on Circuits and Systems*, vol. 34, no. 2, pp. 149–159, Feb. 1987.

[44] R. P. Severns and G. Bloom, *Modern DC-to-DC Switchmode Power Converter Circuits*. San Rafael, CA: e/j BLOOM Associates, Inc., 1985.

[45] M. Albulet, *RF Power Amplifiers*. Atlanta, GA: Noble Publishing, 2001.

[46] A. E. Thiessen, *A History of the General Radio Company*. West Concord, MA: General Radio Company, 1965.
PDF available from http://www-2.cs.cmu.edu/ ram/electro/gr/history.pdf.

[47] Amplifier Research, Inc., *Amplifier Research Model 10W1000C*.

[48] L. Wall, T. Christiansen, and R. L. Schwartz, *Programming Perl*. Sebastapol, CA: O'Reilly Associates, 2nd ed., 1996.

[49] M. Cooper, "Advanced Bash-Scripting Guide."
http://www.tldp.org/LDP/abs/html/.

[50] "GNU make." http://www.gnu.org/software/make/manual/.

[51] "XFig Version 3.2.4 user manual." http://www.xfig.org/userman/.

[52] "Adobe PostScript 3." http://www.adobe.com/products/postscript/.

[53] M. Kamon, L. M. Silveira, C. Smithhisler, and J. White, *FastHenry User's Guide*. Cambridge, MA: MIT Research Laboratory of Electronics, 1996.

[54] "EAGLE: The easily applicable graphical layout editor."
http://www.cadsoft.de/info.htm.

Appendix A

SAMwIICh Code Listings

This appendix consists of code listings for the SAMwIICh system. SAMwIICh uses a combination of perl scripts [48], bash scripts [49], and Makefiles [50].

A.1 `Makefile`

Using a Makefile `make`s everything easier (sorry, pun intended).

```
WORKDIR := $(shell basename $$PWD)
CALCOPTS = -V -C

all: lis imp pwr

lis: $(WORKDIR).lis

imp: $(WORKDIR).imp

pwr: $(WORKDIR).pwr

clean:
	rm -f *.lis *.imp *.pwr *.st0 *.tr0 *.mt0 *.ipp

%.imp: %.lis ../bin/impcalc.pl ../bin/impextract.sh
	../bin/impextract.sh $< $@ | perl ../bin/impcalc.pl $(CALCOPTS) | tee $@

%.pwr: %.lis ../bin/effcalc.pl ../bin/effextract.sh
	../bin/effextract.sh $< $@ | perl ../bin/effcalc.pl | tee $@

%.lis: %.sp values.dat
	hspice $< | tee $@
```

A.2 `automatch.pl`

This is the heart of SAMwIICh; it dispatches all the simulation jobs and sets up component values for the next iteration.

```
#!/usr/bin/perl

use Getopt::Std;
my %opts;
getopts('r:c:v:l:d:m:', \%opts);

# get the name of the directory. Everything else
# should have this name, too
$PWNAM = `basename $ENV{'PWD'}`;
chomp $PWNAM;

# set up utility variables
$done = 0;
$maxiter = 15;
$TOLER = .002;

# the component values for the first iteration
$nextr = $opts{'r'} || 6;
$nextc = $opts{'c'} || 2e-9;
$vin   = $opts{'v'} || 9;
$lser  = $opts{'l'} || 10e-9;
$ldio  = $opts{'d'} || 6e-9;
$lmut = $opts{'m'} || 0;

# set the first round of values
open(VALS,">values.dat") or die "Could not set R and C values: $!";
print VALS join(' ', ($nextr, $nextc, $vin, $lser, $ldio, $lmut)), "\n";
close(VALS);

# until $done is true or if we do too many iterations
while ((!$done) && ($maxiter-- > 0))
{
        # clean up
        `make clean`;
        # do the body of calculations
        # hspice is called from the makefile
        `make CALCOPTS="-V -C" imp`;
        # read in the impedance values
        open(IMP, "<".$PWNAM.".imp");
        ($mag, $rval, $xval, $inx, $inxc) = split(' ', <IMP>);
        close(IMP);

        # we match assuming that we're cancelling inductance with capacitance
        # and then adding a high-pass L-match.
        if ($inxc =~ /H/)
        {
                # oops! the input looks capacitive for some reason...
```

```
                $done = 1;
                undef $nextc;
                undef $vin;
                undef $ℓser;
                undef $ℓdio;
                undef $ℓmut;
                $nextr = "*ERROR: Input looks capacitive!*";
        }
        # OK, things are looking good
        else
        {
                # how much have things changed from the last run?
                $rtoℓer = abs((abs($nextr)−abs($rvaℓ))/$rvaℓ);
                $ctoℓer = abs((abs($nextc)−abs($inxc))/$inxc);

                # show the user what's going on
                print "$nextr => $rval ($rtoler); $nextc => $inxc ($ctoler)\n";

                # if we're within $TOLER, we're done
                ($done = 1) if (($rtoℓer < $TOLER) && ($ctoℓer < $TOLER));

                # set up the new values
                $nextr = $rvaℓ + 0;
                $nextc = $inxc + 0;
        }

        # write out the new component values for spice
        open(VALS,">values.dat") or die "Could not set R and C values: $!";
        print VALS join(' ', ($nextr, $nextc, $vin, $ℓser, $ℓdio, $ℓmut)), "\n";
        close(VALS);
}
```

A.3 matching.sp

This is the HSPICE input file used for the analysis. Most of this file is setup commands to make HSPICE produce appropriate output. Comments note the location of the circuit under test.

```
SAMwIICh Circuit

.options post=1 nomod nopage INGOLD=1 accurate unwrap

* SPICE model for the diode
* courtesy of International Rectifier
.MODEL d20cjq030 d
+IS=7.45927e−09 RS=0.0632601 N=0.89875 EG=1.3
+XTI=4 BV=30 IBV=0.0001 CJO=2.9346e−10
+VJ=0.4 M=0.427299 FC=0.5 TT=0
+KF=0 AF=1
```

```
* simulation parameters
.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM LMUT = 0
.PARAM LSER = 10e-9
.PARAM QLSER = 100
.PARAM LSHN = 47e-9
.PARAM QLSHN = 100
.PARAM LDs = 6n
.PARAM RSOURCE = 6.03
.PARAM CSOURCE = 2015e-12
.PARAM INV = 9

* the circuit under test
Vin rin 0 sin(0 INV FOSC 0 0)

Rin   rin   inc   RSOURCE
Cin   inc   inm   C=CSOURCE
Vinm inm in    0

Lser1 in    1     L=LSER R='WOSC * LSER / QLSER'

Lshn1 1     nrfoo L=LSHN
Rfoo nrfoo 0     R='WOSC * LSHN / QLSHN'

Ksershn Lser1 Lshn1 K='LMUT / SQRT(LSER * LSHN)'

Ldio 1     3     LDs
Dtest 3    4     d20cjq030
Cout 4     0     30n
Vout 4     0     5

* some useful analysis options
.save all
.probe tran i(Vinm) V(in) V(1) v(3) v(4)
.probe tran p(Vin) p(Dtest) p(Vout) POWER

* these .measure statements allow us to examine input
* and output power, efficiency, and diode dissipation
.measure tran avgpin avg p(Vin) from=.5u to=1u
.measure tran avgprin avg p(Rin) from=.5u to=1u
.measure tran avgpd avg p(Dtest) from=.5u to=1u
.measure tran avgpout avg p(Vout) from=.5u to=1u
.measure tran avgptot avg POWER from=.5u to=1u

* the next line causes HSPICE to do the Fourier analysis
.four 100x v(in) i(vinm)

* this is data from automatch.pl
.DATA valdat MER
+FILE='values.dat' RSOURCE=1 CSOURCE=2 INV=3 LSER=4 LDs=5 LMUT=6
.ENDDATA
```

```
* perform a transient sweep, using the input data
.tran .01n 1u UIC SWEEP DATA=valdat

.end
```

A.4 Efficiency Calculations

The two files in here parse data out of the HSPICE listing and then do the crunching. I've kept the two functions separate so that I can easily swap in a different parser while keeping the number crunching facilities the same, or vice-versa.

A.4.1 `effextract.sh`

This code extracts the data necessary for efficiency calculations from the HSPICE listing in a format useful to `effcalc.pl`.

```
#!/bin/bash

# grab the power calculations out of the spice listing

grep avgpin= $1 | awk '{print($2);}' > $2.pin
grep avgprin= $1 | awk '{print($2);}' > $2.prin
grep avgpout= $1 | awk '{print($2);}' > $2.pout
paste $2.pin $2.prin $2.pout | tr '\t' , > $2.tmp
cat $2.tmp
rm $2.pin $2.prin $2.pout $2.tmp
```

A.4.2 `effcalc.pl`

This script calculates input and output power and conversion efficiency.

```
#!/usr/bin/perl

# reads in data from effextract.sh
# subtracts power burned in the source resistance
# from total power in, then divides output/input

while(<>)
{
    chomp;
    ($pin, $prin, $pout) = split(/,/);
    printf("%s,%s,%s\n",
           abs($pin)-abs($prin),
           abs($pout),
           abs($pout)/(abs($pin)-abs($prin)));
}
```

A.5 Impedance Calculations

Like efficiency, impedance calculations are separated from parsing functionality.

A.5.1 `impextract.sh`

```
#!/bin/bash

# extract impedance parameters from input spice file
# this script grabs the fundamentals of the fourier series
# resulting from '.four FREQ v(in) i(vinm)'

grep -A7 'v(in)' $1 | grep '^[[:space:]]*1[[:space:]]' | \
       awk 'BEGIN {OFS=",";} {print($3,$5);}' > $2.v
grep -A7 'i(vinm)' $1 | grep '^[[:space:]]*1[[:space:]]' | \
       awk 'BEGIN {OFS=",";} {print($3,$5,$2);}' > $2.i

paste $2.v $2.i | tr '\t' ,
rm $2.i $2.v
```

A.5.2 `impcalc.pl`

This is a slightly more complex script; its output is configurable via commandline parameters.

```
#!/usr/bin/perl

# one of the workhorses of the SAMwIICh suite

use Getopt::Std;
my %opts;

# hooray for excessive precision
$pi = 3.14159265358979323846;

getopts('CVhpm:', \%opts);

if ($opts{'h'})
{
    print "Usage: $0 [-C] [-V] [-p] [-m <res>] [-h]\n";
    print " -V             Print impedance magnitude, resistance, reactance, " .
                           "and equivalent component value\n";
    print " -C             Print complement of equivalent component value " .
                           "(i.e. component corresponding to -1*X)\n";
    print " -p             Print equivalent parallel input network\n";
    print " -m <res>       Print high pass L-match resulting in <res> input impedance\n";
    exit(0);
}

sub engnot
```

```
{
# change a number to engineering notation

    # argument
    my $num = shift @_;
    return undef unless $num;

    # if it's not in scientific notation,
    # make it so
    ($num = sprintf('%4.4g',$num)) unless ($num =~ /e/);

    my ($basen, $expnt) = split('e', sprintf('%e', $num));
    my $adj = $expnt % 3;
    $expnt -= $adj;
    $basen *= 10**$adj;
    return $basen . "e" . $expnt;
}

while (<>)
{
# main loop
# read in data from each line of stdin or filenames provided as arguments

    chomp;
    # read in the data
    ($vmag, $vph, $imag, $iph, $freq) = split(/,/);
    $phase = $pi * ($vph - $iph) / 180;
    $omega = 2 * $pi * $freq;
    $zmag = abs($vmag) / abs($imag);
    $R = $zmag * cos($phase);
    $X = $zmag * sin($phase);

    # print out impedance values
    if ($opts{'V'})
    {
        printf("%s %s %sj %s%s ",
                engnot($zmag),
               engnot($R),
               engnot($X),
               engnot((($X < 0) ? (-1 / ($X * $omega))
                      : ($X / $omega))),
               (($X < 0) ? 'F' : 'H')
               );
    }

    # print out conjugate element corresponding to X*
    if ($opts{'C'})
    {
        printf("%s%s ",
                engnot((($X>0) ? (1 / ($X * $omega))
                       : (-1 * $X / $omega))),
               (($X<0) ? 'H' : 'F')
               );
```

```
    }

    # convert series impedance to parallel impedance
    if ($opts{'p'})
    {
        $q = abs($R/$X);
        $Xp = $X * ($q**2+1)/($q**2);
        $Rp = $R * ($q**2+1);

        if ($X < 0)
        {
            $nmstring = 'F';
            $Ep = 1 / ($Xp * $omega);
        }
        else
        {
            $nmstring = 'H';
            $Ep = $Xp / $omega;
        }

        printf("%s %s %s%s ", $q, engnot($Rp), engnot($Ep), $nmstring);
    }

    # matching with a high-pass L-match, cancelling
    # input reactance as necessary
    if ($opts{'m'})
    {
        # first cancel the reactance
        $ecanc = (($X > 0) ? (1 / ($X * $omega))
                           : (1 * ($X / $omega)));

        $q = sqrt(abs(($opts{'m'}/$R) - 1));

        $cm = 1 / ($omega * $q * $R);
        $ℓm = $opts{'m'} / ($q * $omega);

        if ($ecanc > 0)
        {
            $ctot = ($ecanc*$cm)/($ecanc+$cm);
            printf("%s %sF %sH",
                   $q,
                   engnot($ctot),
                   engnot($ℓm));
        }
        else
        {
            printf("%sH %s %sF %sH",
                   engnot(-1*$ecanc),
                   $q,
                   engnot($cm),
                   engnot($ℓm));
        }
    }
```

```
    print "\n";
}
```

Appendix B

SPICE Netlists

This appendix contains all SPICE netlists used.

B.1 Comparison of Analytic Model with Simulation, Series Rectifier

The three netlists in this section are for the simulations performed in section 4.3.

B.1.1 Ideal Diode, Linear Capacitor

The diode is represented by a voltage-controlled resistor.

```
Series−resonant rectifier, run #1

.options post=1 nomod nopage INGOLD=1 accurate unwrap

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM CDp = 123.6e−12
.PARAM LDs = 20.5e−9
.PARAM INV = 19.19

Vin rin 0 sin(0 INV FOSC 0 0)

Vinm rin  in    0

Ldio in   3     LDs
Cdio 3    4     CDp
Gtest 3   4     VCR PWL(1) 3 4 0,1e24 1e−24,1e−24
Vout 4    0     5

.save all
.probe tran i(Vinm) V(in) v(3) v(4)
.probe tran p(Vin) p(Vout) POWER
.measure tran avgpin avg p(Vin) from=.5u to=1u
```

```
.measure tran avgpout avg p(Vout) from=.5u to=1u
.measure tran avgptot avg POWER from=.5u to=1u

.four 100x v(in) i(vinm)

.tran .01n 1u UIC

.end
```

B.1.2 Real Diode with Constant Junction Capacitance

This simulation uses a real diode model, but with the junction capacitance replaced by a constant capacitor.

```
Series-resonant rectifier, run #2

.options post=1 nomod nopage INGOLD=1 accurate unwrap

.MODEL d20cjqfoo d
+IS=7.45927e-09 RS=0.0632601 N=0.89875 EG=1.3
+XTI=4 BV=30 IBV=0.0001 CJO=1e-24
+VJ=0.4 M=0.427299 FC=0.5 TT=0
+KF=0 AF=1

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM CDp = 123.6e-12
.PARAM LDs = 20.5e-9
.PARAM INV = 19.19

Vin rin 0 sin(0 INV FOSC 0 0)

Vinm rin  in    0

Ldio in   3     LDs
Cdio 3    4     CDp
Dtest 3   4     d20cjqfoo
Vout 4    0     5

.save all
.probe tran i(Vinm) V(in) v(3) v(4)
.probe tran p(Vin) p(Dtest) p(Vout) POWER
.measure tran avgpin avg p(Vin) from=.5u to=1u
.measure tran avgpd avg p(Dtest) from=.5u to=1u
.measure tran avgpout avg p(Vout) from=.5u to=1u
.measure tran avgptot avg POWER from=.5u to=1u

.four 100x v(in) i(vinm)

.tran .01n 1u UIC
```

```
.end
```

B.1.3 Real Diode, Constant C_j, Conjugate Source Impedance

As above, a "real" diode with constant junction capacitance was used. This time, the source impedance is the conjugate of the calculated input impedance.

```
Series−resonant rectifier, run #3

.options post=1 nomod nopage INGOLD=1 accurate unwrap

.MODEL d20cjqfoo d
+IS=7.45927e−09 RS=0.0632601 N=0.89875 EG=1.3
+XTI=4 BV=30 IBV=0.0001 CJO=1e−24
+VJ=0.4 M=0.427299 FC=0.5 TT=0
+KF=0 AF=1

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM CDp = 123.6e−12
.PARAM LDs = 20.5e−9
.PARAM INV = '19.19 * 2'
.PARAM RSOURCE = 36.83
.PARAM CSOURCE = 154.26e−12
* since we are driving from a conjugate match,
* we have to double the voltage at the source
* to keep the input voltage the same

Vin rin 0 sin(0 INV FOSC 0 0)

* we must use a shunt capacitor
* because otherwise there is no
* path for DC current
Rin  rin  inm  RSOURCE
Cin  inm  0    CSOURCE
Vinm inm in    0

Ldio in   3    LDs
Cdio 3    4    CDp
Dtest 3   4    d20cjqfoo
Vout 4    0    5

.save all
.probe tran i(Vinm) V(in) v(3) v(4)
.probe tran p(Vin) p(Rin) p(Dtest) p(Vout) POWER
.measure tran avgpin avg p(Vin) from=.5u to=1u
.measure tran avgprin avg p(Rin) from=.5u to=1u
.measure tran avgpd avg p(Dtest) from=.5u to=1u
.measure tran avgpout avg p(Vout) from=.5u to=1u
```

```
.measure tran avgptot avg POWER from=.5u to=1u

.four 100x v(in) i(vinm)

.tran .01n 1u UIC

.end
```

B.2 Series Resonant Rectifier Output Impedance

This rectifier was implemented in [10]. The netlist uses a real diode model, including nonlinear output capacitance.

```
Series-resonant rectifier, output impedance testing

.options post=1 nomod nopage INGOLD=1 accurate unwrap

.MODEL Dmbrs1540t3 d
+IS=3.54179e-05 RS=0.0306875 N=1.4038 EG=0.6
+XTI=1.97409 BV=40 IBV=0.0008 CJO=3.18451e-10
+VJ=0.4 M=0.428536 FC=0.5 TT=0
+KF=0 AF=1

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM LDs = 1e-9
.PARAM LSHN = 12.5e-9
.PARAM QLSHN= 150
.PARAM INV = 7.5
.PARAM VOUT = 5

Vin rin 0 sin(0 INV FOSC 0 0)

Vinm rin  in   0

Lshn in   nfoo LSHN
Rfoo nfoo 0    R='WOSC * LSHN / QLSHN'
* we account for the finite Q of the
* inductor by calculating the appropriate
* series resistance

Ldio in   3    LDs
Dtest 3   4    Dmbrs1540t3
Vout 4    0    VOUT

.save all
.probe tran i(Vinm) V(in) v(3) v(4)
.probe tran p(Vin) p(Dtest) p(Vout) POWER
.measure tran avgpin avg p(Vin) from=.5u to=1u
.measure tran avgpd avg p(Dtest) from=.5u to=1u
```

```
.measure tran avgpout avg p(Vout) from=.5u to=1u
.measure tran avgptot avg POWER from=.5u to=1u

.four 100x v(in) i(vinm)

.data vodat VOUT
+4.98 4.99 5.00 5.01 5.02
.enddata

.tran .01n 1u UIC SWEEP DATA=vodat

.end
```

B.3 Comparison of Analytic Model with Simulation, Shunt Rectifier

The netlists in this section are for the simulations performed in section 5.2.

B.3.1 Ideal Diode, Linear Capacitor

```
Shunt−resonant rectifier, run #1

.options post=1 nomod nopage INGOLD=1 accurate unwrap itl4=100

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM CSER = 110.4e−12
.PARAM LSER = 28.24e−9
.PARAM CD = 101.32e−12
.PARAM LCHOKE = 2.5e−6
.PARAM INV = 5.6
.PARAM OUTV = 5

Vin in 0 sin(0 INV FOSC 0 0)

Cser1 in  cin   CSER
Lser1 cin  lin   LSER
Vinm lin  1     0

Cd   1     0     CD
Gtest 0    1     VCR PWL(1) 0 1 0,1e24 1e−24,1e−24

Lchk 1     4     LCHOKE
Voutm 4    out   0
Vout 4     0     OUTV

.save all
.probe tran i(Vinm) V(1) v(4) v(out) i(Voutm)
.probe tran p(Vin) p(Vout) p(Gtest) POWER
```

```
.measure tran avgpin avg p(Vin) from=19u to=20u
.measure tran avgpd avg p(Gtest) from=19u to=20u
.measure tran avgpout avg p(Vout) from=19u to=20u
.measure tran avgptot avg POWER from=19u to=20u

.tran .1n 20u

.end
```

B.3.2 Real Diode with Constant Junction Capacitance

```
Shunt-resonant rectifier, run #2

.options post=1 nomod nopage INGOLD=1 accurate unwrap itl4=100

.MODEL d20cjqfoo d
+IS=7.45927e-09 RS=0.0632601 N=0.89875 EG=1.3
+XTI=4 BV=30 IBV=0.0001 CJO=1e-24
+VJ=0.4 M=0.427299 FC=0.5 TT=0
+KF=0 AF=1

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM CSER = 110.4e-12
.PARAM LSER = 28.24e-9
.PARAM CD = 101.32e-12
.PARAM LCHOKE = 2.5e-6
.PARAM INV = 5.6
.PARAM OUTV = 5

Vin in 0 sin(0 INV FOSC 0 0)

Cser1 in  cin   CSER
Lser1 cin  lin   LSER
Vinm lin  1     0

Cd   1     0     CD
Dtest 0    1     d20cjqfoo

Lchk 1     4     LCHOKE
Voutm 4    out   0
Vout 4     0     OUTV

.save all
.probe tran i(Vinm) V(1) v(4) v(out) i(Voutm)
.probe tran p(Vin) p(Vout) p(Gtest) POWER
.measure tran avgpin avg p(Vin) from=19u to=20u
.measure tran avgpd avg p(Gtest) from=19u to=20u
.measure tran avgpout avg p(Vout) from=19u to=20u
.measure tran avgptot avg POWER from=19u to=20u

```

```
.tran .1n 20u

.end
```

B.4 Shunt Resonant Rectifier, Experimental Implementation

This netlist corresponds to the experimental implementation of the rectifier described in section 5.4.

```
Shunt−resonant rectifier, no inductance cancllation

.options post=1 nomod nopage INGOLD=1 accurate unwrap itl4=100

.MODEL d20cjq060 d
+IS=7.45927e−09 RS=0.0632601 N=0.89875 EG=1.3
+XTI=4 BV=30 IBV=0.0001 CJO=144.56e−12
+VJ=0.43217 M=0.47728 FC=0.5 TT=0
+KF=0 AF=1

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
.PARAM LMUT = 0
.PARAM LSER = 1e−9
.PARAM QLSER = 150
.PARAM LSHN = 120e−9
.PARAM QLSHN = 150
.PARAM LDs = 2e−9
.PARAM RSOURCE = 11.95
.PARAM CMAT = 74.6378e−12
.PARAM LMAT = 38e−9
.PARAM QLMAT = 150
.PARAM INV = 21
.PARAM FOOB = 0

Vin rin 0 sin(0 INV FOSC 0 0)

* note that the input capacitance is
* in the series path, not the shunt
* path. This is the reason for the
* discrepancy between the value used
* for experimentation and the value
* used in this simulation. Note that
* series-shunt conversion yields the
* correct value

Rin  rin  ino  RSOURCE
Cm   ino  inm  CMAT
Lm   inm  int  L=LMAT R='WOSC * LMAT / QLMAT'
Tin  int 0 inmo 0 Z0=51.5 TD=5e−9 L=.95
Vinm inmo in  0
```

```
Lser1 in   1    L=LSER R='WOSC * LSER / QLSER'

Ldio 1     3    Lds
Dtest 0    3    d20cjq060

C3par 3    0    2e-12

Lshn1 1    4    L=LSHN R='WOSC * LSHN / QLSHN'

Ksershn Lser1 Lshn1 K='LMUT / SQRT(LSER * LSHN)'

Voutm 4    out  0

Cout out   0    100n
Vout out   0    5

.save all
.probe tran i(Vinm) V(inm) V(in) V(1) v(3) v(4) v(out) i(Voutm)
.probe tran p(Vin) p(Rin) p(Dtest) p(Vout) POWER
.measure tran avgpin avg p(Vin) from=.5u to=1u
.measure tran avgprin avg p(Rin) from=.5u to=1u
.measure tran avgpd avg p(Dtest) from=.5u to=1u
.measure tran avgpout avg p(Vout) from=.5u to=1u
.measure tran avgptot avg POWER from=.5u to=1u
.measure tran ippout PP i(Voutm) from=.7u to=.72u

.four 100x v(in) i(vinm) i(voutm)

.tran .01n 1u UIC

.end
```

B.5 Shunt Resonant Rectifier with Parasitic Mitigation

This netlist corresponds to the experimental implementation of the shunt rectifier with parasitic mitigation described in section 5.5.

```
Shunt-resonant rectifier with inductance cancellation

.options post=1 nomod nopage INGOLD=1 accurate unwrap itl4=100

.model legd d is = 2.37487E-007 n = 1.98477 rs = 0.0171579
+ eg = 1.79999 xti = 3.99991
+ cjo = 5.47556E-010 vj = 1.64135 m = 0.603662 fc = 0.5
+ tt = 1.4427E-009 bv = 48.4 ibv = 3.5 af = 1 kf = 0
.model grd d is = 1.27781E-005 n = 1.2149 rs = 0.0250254
+ eg = 0.55507 xti = 0.794212

.subckt d1n5822 1 2
```

```
ddio 1 2 legd
dgr 1 2 grd
.ends

.PARAM FOSC = 100e6
.PARAM WOSC = '6.283185 * FOSC'
* set this parameter to the desired mutual inductance
.PARAM LMUT = 0
* dummy value; most of the inductance is in LMAT
.PARAM LSER = 1e-9
.PARAM QLSER = 150
.PARAM LSHN = 120e-9
.PARAM QLSHN = 150
.PARAM LDs = 6.5e-9
.PARAM RSOURCE = 2.56
* note that this is in series, not in shunt as in the actual implementation
.PARAM CMAT = 144e-12
.PARAM LMAT = 13.8e-9
.PARAM QLMAT = 150
.PARAM INV = 29.7

Vin rin 0 sin(0 INV FOSC 0 0)

Rin   rin   ino   RSOURCE
Cm    ino   inm   CMAT
Lm    inm   int   L=LMAT R='WOSC * LMAT / QLMAT'
Cm2 inm   int   C=CMAT2
Tin   int 0 inmo 0 Z0=51.5 TD=5e-9 L=.95
Vinm inmo in  0

Lser1 in    1     L=LSER R='WOSC * LSER / QLSER'

Ldio 1      3     Lds
Xtest 0     3     d1n5822

C3par 1     0     2e-12

Lshn1 1     4     L=LSHN R='WOSC * LSHN / QLSHN'

Ksershn Lser1 Lshn1 K='LMUT / SQRT(LSER * LSHN)'

Voutm 4   out   0

Cout out  0     100n
Vout out  0     5

.save all
.probe tran i(Vinm) V(inm) V(in) V(1) v(3) v(4) v(out) i(Voutm)
.probe tran p(Vin) p(Rin) p(Dtest) p(Vout) POWER
.measure tran avgpin avg p(Vin) from=.5u to=1u
.measure tran avgprin avg p(Rin) from=.5u to=1u
.measure tran avgpd avg p(Dtest) from=.5u to=1u
.measure tran avgpout avg p(Vout) from=.5u to=1u
```

```
.measure tran avgptot avg POWER from=.5u to=1u
.measure tran ippout PP i(Voutm) from=.7u to=.72u

.four 100x v(in) i(vinm) i(voutm)
70
.tran .01n 1u UIC

.end
```

Appendix C

Scripts for PCB Inductor Generation

This appendix provides listings of the perl scripts [48] used to generate coupled magnetic structures on PCBs.

C.1 mkcan.pl

`mkcan.pl` generates 20 coupled magnetic structures of increasing size and produces both FastHenry simulations and scripts which produce a single PCB with the magnetic structures laid out in a tiled pattern.

```
#!/usr/bin/perl -w

# the endpoints are always the same,
# allowing consistent via placement
$endpttop = "270,-90";
$endptbot = "440,250";
$viapt    = "270,0";

# these define the base radius and increment
$baserad = 79;
$radmul = 10;

for (my $i=0; $i<20; $i++)
{
    # make a place to store the data
    mkdir("p1r" . ($baserad+($radmul*$i)));
    chdir("p1r" . ($baserad+($radmul*$i)));

    # computer the new radius
    $raditer = $baserad + ($radmul * $i);

    # generate the top spiral
```

```
    $topcmd = ". ./bin/spiral.pl -t 1 -w 50 -c 35 -r "
              . $raditer . " -C -e " . $endpttop
              . " -b " . $viapt . ",0 > top.dat";
    print `$topcmd`;

    # generate the bottom spiral
    $botcmd = ". ./bin/spiral.pl -f -t 1 -w 50 -c 35 -r "
              . $raditer . " -C -z -62 -e " . $endptbot
              . " -b " . $viapt . "> bot.dat";
    print `$botcmd`;

    # since we are tiling these on a PCB, give each
    # one an appropriate offset to make tiling simple
    $xoff = 1000 + ($i % 5)*2000;
    $yoff = 1000 + int($i/5)*2000;

    # generate the Eagle script for the top
    $topscr = ". ./bin/dat2scr.pl -f top.dat -p . ./gpdat -x "
              . $xoff . " -X " . $xoff . " -y "
              . $yoff . " -Y " . $yoff . " > top.scr";
    print `$topscr`;

    # generate the Eagle script for the bottom
    $botscr = ". ./bin/dat2scr.pl -f bot.dat -p . ./gpdat -x "
              . $xoff . " -X " . $xoff . " -y "
              . $yoff . " -Y " . $yoff . " > bot.scr";
    print `$botscr`;

    # generate the part placement script
    $plcmd = ". ./bin/placescr.pl -f . ./placedat -x $xoff "
              . "-y $yoff > place.scr";
    `$plcmd`;

    # generate the data for FastHenry simulation
    $indcmd = ". ./bin/dat2ind.pl -1 top.dat -2 bot.dat -F "
              . "1e8,1e8,1 -b -H 3 -W 4 -p . ./gpdat > full.ind";
    print `$indcmd`;

    # make the zbuf file
    $fhzcmd = "fasthenry -f refined full.ind";
    `$fhzcmd &>/dev/null `;
    `zbuf zbuffile2 &>/dev/null `;
    # fork off a new process to display the inductors
    unless (fork())
    {
        # oooooh, pretty
        print `gv -geometry +0-0 zbuffile2.ps`;
        exit;
    }

    # meanwhile, Rick James simulates them
    `fasthenry full.ind &>/dev/null`;
```

```
    # extract the data from the FastHenry simulation
    print `cat Zc.mat | ../bin/qcalc.pl | tee Ldat`;

    # get ready to run another cycle
    chdir('..');
}
```

C.2 spiral.pl

spiral.pl generates spirals and circles in a generic format processed by dat2fig.pl, dat2ind.pl, and dat2scr.pl.

```
#!/usr/bin/perl -w

use strict;
use Getopt::Std;

# ha ha
my $pi = 3.14159265358979323846264338327950;
my $maxtheta;
my $radrange;
my %opts;
getopts('t:w:s:c:x:y:z:b:e:r:Cfph', \%opts);

# show help if they don't give -t foo
$opts{'h'}++ unless ($opts{'t'});
$opts{'h'}++ if ($opts{'C'} && !($opts{'r'}));

if ($opts{'h'})
{
    print "Usage: $0 -t <turns> [opt [opt [opt]]]\n";
    print " -t <turns>        Number of turns (required)\n";
    print " -w <width>        Width of conductor in each turn (default 50)\n";
    print " -s <space>        Turn spacing multiplier (default 1.5)\n";
    print " -c <chunks>       Number of chunks (line segments) in output file\n";
    print " -x <xoff>         Offset in X-dimension (ignored in cylindrical mode)\n";
    print " -y <yoff>         Offset in Y-dimension (ignored in cylindrical mode)\n";
    print " -z <zoff>         Offset in Z-dimension\n";
    print " -e <x1,y1[:x2,y2...]> Connects the outer end of the spiral to the given points\n";
    print " -b <x1,y1[:x2,y2...]> Connects the inner end of the spiral to the given points\n";
    print "                   (These points are referred to the center of the spiral,\n";
    print "                   i.e. they are offset by <xoff> and <yoff>)\n";
    print " -r <rad>          Start with initial radius <rad> instead of zero\n";
    print " -C                Make a circle instead of a spiral. Must also specify -r.\n";
    print " -f                Flip spiral direction (default counterclockwise)\n";
    print " -p                Output points in cylindrical form (default Cartesian)\n";
    print " -h                Show this help screen\n";
    exit(0);
}
```

```
$opts{'w'} ||= 50;
$opts{'s'} ||= 1.5;
$opts{'c'} ||= 100;
$opts{'x'} ||= 0;
$opts{'y'} ||= 0;
$opts{'z'} ||= 0;
$opts{'r'} ||= 0;
$opts{'C'} ||= 0;

if ($opts{'b'})
{
    my @points = split(/:/, $opts{'b'});

    foreach my $point (@points)
    {
        my ($xval, $yval) = split(/,/, $point);
        $xval += $opts{'x'};
        $yval += $opts{'y'};

        if ($opts{'p'})
        {
            my $radius = sqrt($xval**2 + $yval**2);
            my $theta = atan2($yval,$xval);

            print join(",", ($radius, $theta, $opts{'z'}, $opts{'w'})) . "\n";
        }
        else
        {
            print join(",", ($xval, $yval, $opts{'z'}, $opts{'w'})) . "\n";
        }
    }
}

if ($opts{'C'})
{
    unless ($opts{'t'} < 1)
    {
        # if it's bigger than 1 turn, reduce it to 1 turn less
        # 1.75 wire widths
        $maxtheta = ((2 * $pi) - (1.75 * $opts{'w'} / $opts{'r'})) * (($opts{'f'}) ? -1 : 1);
    }
    else
    {
        # just calculate based on the number of turns
        $maxtheta = $opts{'t'} * 2 * $pi * (($opts{'f'}) ? -1 : 1);
    }
}
else
{
    # spiral starts at 0, goes to 2*pi*#turns
    $maxtheta = $opts{'t'} * 2 * $pi * (($opts{'f'}) ? -1 : 1);
    # radrange is the distance we'll be travelling
```

```
    $radrange = $opts{'w'} * $opts{'t'} * $opts{'s'};
}

for (my $i=0; $i<($opts{'c'}+1); $i++)
{
    my $radius = ($opts{'C'}) ? $opts{'r'} : (($radrange * $i / $opts{'c'}) + $opts{'r'});
    my $theta = $maxtheta * $i / $opts{'c'};

    if ($opts{'p'})
    {
        print join(",", ($radius, $theta, $opts{'z'}, $opts{'w'})) . "\n";
    }
    else
    {
        my $xval = $radius * cos($theta);
        my $yval = $radius * sin($theta);
        print join(",", ($xval+$opts{'x'}, $yval+$opts{'y'}, $opts{'z'}, $opts{'w'})) . "\n";
    }
}

if ($opts{'e'})
{
    my @points = split(/:/, $opts{'e'});

    foreach my $point (@points)
    {
        my ($xval, $yval) = split(/,/, $point);
        $xval += $opts{'x'};
        $yval += $opts{'y'};

        if ($opts{'p'})
        {
            my $radius = sqrt($xval**2 + $yval**2);
            my $theta = atan2($yval,$xval);

            print join(",", ($radius, $theta, $opts{'z'}, $opts{'w'})) . "\n";
        }
        else
        {
            print join(",", ($xval, $yval, $opts{'z'}, $opts{'w'})) . "\n";
        }
    }
}
```

C.3 dat2fig.pl

`dat2fig.pl` generates a graphical representation of the output of `spiral.pl` suitable for display with XFig [51] or translation to PostScript [52] using the `transfig` program [51].

```
#!/usr/bin/perl -w
```

```
use strict;
use Getopt::Std;

my %opts;
my @data;

getopts('f:p:X:Y:h', \%opts);

if ($opts{'h'})
{
    print "Usage: $0 [opts]\n";
    print " -f <filename>   Read data from filename (stdin)\n";
    print " -p <planefile>  Ground plane data file\n";
    print " -X <xpoff>      Ground plane offset xpoff in x direction\n";
    print " -Y <ypoff>      Ground plane offset ypoff in y direction\n";
    print " -h              Show this help screen\n";
    exit(0);
}

$opts{'f'} ||= '-';

open FILE, '<' . $opts{'f'} or
    die "Couldn't open file: $!";

while (<FILE>)
{
    chomp;
    my @tmp = split(/,/);
    push @data, \@tmp;
}

close FILE;

# we assume that width is the same for all of the points
# maybe next time it would be good to use a different width
# for each one, but this is good enough for my purposes
my $width = int((8/100 * $data[0][3]));
my $numpoints = $#data + 1;

print <<END;
\#FIG 3.2
Landscape
Center
Inches
Letter
100.00
Single
-2
1000 2
2 1 0 $width 0 7 50 -1 -1 0.000 1 0 -1 0 0 $numpoints
END
```

```
# make the pretty XFig picture
for (my $i=0; $i<=$#data; $i++)
{
    print int($data[$i][0]) . " ";
    print int($data[$i][1]) . "\n";
}

@data = ();

# put in the ground planes if desired
if ($opts{'p'})
{
    open FILE, '<' . $opts{'p'} or
        die "Couldn't open plane data file: $!";

    while (<FILE>)
    {
        chomp;
        my @tmp = split(/,/);
        push @data, \@tmp;
    }

    for (my $i=0; $i<=$#data; $i++)
    {
        print "2 2 0 1 0 0 50 -1 20 0.000 0 0 -1 0 0 5\n";
        printf("%d %d %d %d %d %d %d %d %d %d\n",
               int(${$data[$i]}[0] + $opts{'X'}),
               int(${$data[$i]}[1] + $opts{'Y'}),
               int(${$data[$i]}[3] + $opts{'X'}),
               int(${$data[$i]}[4] + $opts{'Y'}),
               int(${$data[$i]}[6] + $opts{'X'}),
               int(${$data[$i]}[7] + $opts{'Y'}),
               int(${$data[$i]}[0] + $opts{'X'}),
               int(${$data[$i]}[7] + $opts{'Y'}),
               int(${$data[$i]}[0] + $opts{'X'}),
               int(${$data[$i]}[1] + $opts{'Y'}));
    }
}
```

C.4 dat2ind.pl

`dat2ind.pl` generates a representation of data from `spiral.pl` suitable for simulation with the FastHenry field solver [53].

```
#!/usr/bin/perl -w

use strict;
use Getopt::Std;

my @planedata;
```

```
my $onefile = 0;
my @data;
my %opts;
getopts('f:1:2:F:p:bes:r:H:W:t:hX:Y:Z:', \%opts);

$opts{'h'}++ unless ($opts{'f'} || $opts{'1'} || $opts{'2'});

if ($opts{'f'})
{
    if ($opts{'1'} || $opts{'2'})
    {
        print "Cannot specify both -f and -1/-2.\n";
        $opts{'h'}++;
    }

    $onefile = 1;
}
elsif (!($opts{'1'} && $opts{'2'}))
{
    print "Must specify either -f or -1 and -2.\n";
    $opts{'h'}++;
}

if ($opts{'s'} && $opts{'r'})
{
    print "Cannot specify both sigma and rho.\n";
    $opts{'h'}++;
}

unless ($opts{'r'})
{
    $opts{'s'} ||= 1513.84;
}

if ($opts{'h'})
{
    print "Usage: $0 (-f <file> | -1 <file1> -2 <file2>) [opt [opt [opt]]]\n";
    print " **NOTE** Input units must be mils\n";
    print " -f <file>         Process a single file\n";
    print " -1 <file1> -2 <file2> Process two files connected together\n";
    print " -F <min,max,ndec> Simulate from min to max with ndec pts/dec (default DC)\n";
    print " -p <planefile>    Ground plane data file\n";
    print " -b                Connect two files at initial points\n";
    print " -e                Connect last point of <file1> to first point of <file2>\n";
    print " -s <sigma>        Set conductivity to sigma (1/(mil*Ohm)) (default Cu)\n";
    print " -r <rho>          Set resistivity to rho\n";
    print " -H <nhinc>        Number of height increments (default 5)\n";
    print " -W <nwinc>        Number of width increments (default 10)\n";
    print " -t <thickness>    Copper thickness (default 1.4 mils = 1 oz Cu)\n";
    print " -X <pxoff>        Ground planes offset by pxoff in x axis\n";
    print " -Y <pyoff>        Ground planes offset by pxoff in y axis\n";
    print " -Z <pzoff>        Ground planes offset by pxoff in z axis\n";
    print " -h                Show this help screen\n";
```

```
    exit(0);
}

$opts{'H'} ||= 5;
$opts{'W'} ||= 10;
$opts{'t'} ||= 1.4;
$opts{'F'} ||= "0,1,1";
$opts{'X'} ||= 0;
$opts{'Y'} ||= 0;
$opts{'Z'} ||= 0;

if ($onefile)
{
    open (FILE, "<" . $opts{'f'})
        or die "Couldn't open input file: $!";
    @data = <FILE>;
    close FILE;
    chomp @data;

    print "* Conversion of " . $opts{'f'} . "\n";
}
else
{
    open (FILE, "<" . $opts{'1'})
        or die "Couldn't open first input file: $!";
    @{$data[0]} = <FILE>;
    close FILE;
    open (FILE, "<" . $opts{'2'})
        or die "Couldn't open second input file: $!";
    @{$data[1]} = <FILE>;
    close FILE;
    chomp @{$data[0]};
    chomp @{$data[1]};

    print "* Conversion of " . $opts{'1'} . " and " . $opts{'2'} . "\n";
}

print ".units mil\n";
print ".default " . ($opts{'r'} ? ("rho=" . $opts{'r'}) : ("sigma=" . $opts{'s'})) . "\n";
print ".default nhinc=" . $opts{'H'} . " nwinc=" . $opts{'W'} . "\n\n*NODES:\n";

if ($opts{'p'})
{
    open FILE, '<' . $opts{'p'} or
        die "Couldn't open ground plane file: $!";

    while (<FILE>)
    {
        chomp;
        my @tmp = split(/,/);
        push @planedata, \@tmp;
    }
```

```
    close FILE;

    for (my $i=0; $i<=$#planedata; $i++)
    {
        printf("G%d x1=%g y1=%g z1=%g
                + x2=%g y2=%g z2=%g
                + x3=%g y3=%g z3=%g
                + thick=%g seg1=%g seg2=%g\n",
               $i, ${$pℓanedata[$i]}[0] + $opts{'X'},
               ${$pℓanedata[$i]}[1] + $opts{'Y'}, ${$pℓanedata[$i]}[2]
               + $opts{'Z'}, ${$pℓanedata[$i]}[3] + $opts{'X'},
               ${$pℓanedata[$i]}[4] + $opts{'Y'}, ${$pℓanedata[$i]}[5]
               + $opts{'Z'}, ${$pℓanedata[$i]}[6] + $opts{'X'},
               ${$pℓanedata[$i]}[7] + $opts{'Y'}, ${$pℓanedata[$i]}[8]
               + $opts{'Z'}, $opts{'t'}, ${$pℓanedata[$i]}[9] || 10,
               ${$pℓanedata[$i]}[10] || 10);
    }
}

if ($onefiℓe)
{
    for (my $i=0; $i<=$#data; $i++)
    {
        my @tmp = split(/,/, $data[$i]);
        printf("N%d x=%g y=%g z=%g\n", $i, $tmp[0], $tmp[1], $tmp[2]);
    }

    print "\n*EDGES\n";

    for (my $i=0; $i<$#data; $i++)
    {
        my @tmp = split(/,/, $data[$i]);
        printf("E%d N%d N%d w=%g h=%g\n", $i, $i, $i+1, $tmp[3], $opts{'t'});
    }

    printf("\n.external N1 N%d", $#data);
}
else
{
    for (my $i=0; $i<2; $i++)
    {
        print "*file " . $opts{$i+1} . "\n";
        for (my $j=0; $j<=$#{$data[$i]}; $j++)
        {
            my @tmp = split(/,/, ${$data[$i]}[$j]);
            printf("Nf%dn%d x=%g y=%g z=%g\n", $i, $j, $tmp[0], $tmp[1], $tmp[2]);
        }
    }

    print "\n*EDGES\n";

    for (my $i=0; $i<=$#data; $i++)
    {
```

```
        print "*file " . $opts{$i+1} . "\n";
        for (my $j=0; $j<$#{$data[$i]}; $j++)
        {
            my @tmp = split(/,/, ${$data[$i]}[$j]);
            printf("Ef%dn%d Nf%dn%d Nf%dn%d w=%g h=%g\n", $i, $j, $i, $j, $i, $j+1, $tmp[3], $opts{'t'});
        }
    }

    if ($opts{'e'})
    {
        printf("\n.equiv Nf0n%d Nf1n1\n", $#{$data[0]});
    }
    else
    {
        print "\n.equiv Nf0n1 Nf1n1\n";
    }

    printf(".external Nf0n1 Nf0n%d\n", $#{$data[0]});
    printf(".external Nf1n1 Nf1n%d\n", $#{$data[1]});
}

my ($fmin, $fmax, $ndec) = split(/,/, $opts{'F'});

printf("\n.freq fmin=%g fmax=%g ndec=%g\n", $fmin, $fmax, $ndec);

print "\n.end\n";
```

C.5 dat2scr.pl

`dat2scr.pl` generates placement scripts for the EAGLE CAD program [54] from data generated by `spiral.pl`.

```
#!/usr/bin/perl -w

use strict;
use Getopt::Std;
my %opts;
getopts('f:p:x:y:X:Y:h', \%opts);

if ($opts{'h'})
{
    print "Usage: $0 -f <file> [opts]\n";
    print " -f <file>      Data file to read (stdin)\n";
    print " -p <planefile> Ground plane datafile\n";
    print " -x <xoff>      Offset by xoff mils in x\n";
    print " -y <yoff>      Offset by yoff mils in y\n";
    print " -X <xpoff>     Offset ground plane by xpoff mils in x\n";
    print " -Y <ypoff>     Offset ground plane by ypoff mils in y\n";
    print " -h             Show this help screen\n";
    exit(0);
```

```
}

my @data;
$opts{'f'} ||= '-';
$opts{'x'} ||= 0;
$opts{'y'} ||= 0;
$opts{'X'} ||= 0;
$opts{'Y'} ||= 0;

open FILE, '<' . $opts{'f'} or
    die "Couldn't open input file: $!";

while (<FILE>)
{
    chomp;
    my @tmp = split(/,/);
    push @data, \@tmp;
}

close FILE;

# width should never change, so we just go with the first one
my $width = $data[0][3] / 1000;
my $layer = ($data[0][2] < 0) ? "bottom" : "top";

print "set wire_bend 2;\n";
print "change layer $layer;\n";

print "wire " . $width . " ";

for (my $i=0; $i<=$#data; $i++)
{
    print("(" . (($data[$i][0] + $opts{'x'})/1000) . " ");
    print((($data[$i][1]+$opts{'y'})/1000) . ")\n");
}

print ";\n";

@data = ();

if ($opts{'p'})
{
    open FILE, '<' . $opts{'p'} or
        die "Couldn't open ground plane file: $!";

    while (<FILE>)
    {
        chomp;
        my @tmp = split(/,/);
        push @data, \@tmp;
    }

    for (my $i=0; $i<=$#data; $i++)
```

```
    {
        printf("rect (%g %g) (%g %g);\n",
              ($data[$i][0] + $opts{'X'})/1000,
              ($data[$i][1] + $opts{'Y'})/1000,
              ($data[$i][6] + $opts{'X'})/1000,
              ($data[$i][7] + $opts{'Y'})/1000);
    }
}
```

C.6 placescr.pl

`placescr.pl` takes input from `placedat` and generates a script to place components on an EAGLE layout.

```
#!/usr/bin/perl -w

use strict;
use Getopt::Std;
my %opts;
getopts('f:x:y:h', \%opts);

if ($opts{'h'})
{
    print "Usage: $0 -f <file> [opts]\n";
    print " -f <file>      Data file to read (stdin)\n";
    print " -x <xoff>      Offset by xoff mils in x\n";
    print " -y <yoff>      Offset by yoff mils in y\n";
    print " -h             Show this help screen\n";
    exit(0);
}

my @data;
$opts{'f'} ||= '-';
$opts{'x'} ||= 0;
$opts{'y'} ||= 0;

open FILE, '<' . $opts{'f'} or
    die "Couldn't open input file: $!";

while (<FILE>)
{
    chomp;
    my @tmp = split(/:/);
    push @data, \@tmp;
}

for (my $i=0; $i<=$#data; $i++)
{
    if (${$data[$i]}[0] eq "WIRE")
    {
```

```
        my ($part, $xstart, $ystart, $xend, $yend, $ℓayer, $width) = @{$data[$i]};

        $width /= 1000;
        $xstart = ($xstart+$opts{'x'})/1000;
        $ystart = ($ystart+$opts{'y'})/1000;
        $xend = ($xend+$opts{'x'})/1000;
        $yend = ($yend+$opts{'y'})/1000;

        print "change layer $layer;\n";
        print "wire $width ($xstart $ystart) ($xend $yend);\n";
    }
    elsif (${$data[$i]}[0] eq "VIA")
    {
        my ($part, $xpos, $ypos, $diameter, $driℓℓ, $shape) = @{$data[$i]};

        $diameter /= 1000;
        $driℓℓ    /= 1000;
        $xpos = ($xpos+$opts{'x'})/1000;
        $ypos = ($ypos+$opts{'y'})/1000;

        print "change drill $drill;\nvia $diameter $shape ($xpos $ypos);\n";
    }
    else
    {
        my ($part, $xpos, $ypos, $rotation) = @{$data[$i]};

        $xpos = ($xpos+$opts{'x'})/1000;
        $ypos = ($ypos+$opts{'y'})/1000;

        print "add $part $rotation ($xpos $ypos);\n";
    }
}
```

C.7 qcalc.pl

qcalc.pl parses the Zc.mat file generated by FastHenry to provide self and mutual inductance and Qdata.

```
#!/usr/bin/perl -w

use strict;
my ($L1, $L2);
# heh
my $pi = 3.14159265358979323846264338327 95;

sub engnot
{
    # argument
    my $num = shift @_;
    return undef unless $num;
```

```
    my ($basen, $expnt) = split('e', sprintf('%e', $num));
    my $adj = $expnt % 3;
    $expnt -= $adj;
    $basen *= 10**$adj;
    return $basen . "e" . $expnt;
}

while (<>)
{
    if (/Impedance matrix for frequency = (\S*)/)
    {
        my $freq = $1;
        my $line1 = <>;
        my $line2 = <>;

        my ($r11, $l11, $r12, $l12) = split(' ', $line1);
        my ($r21, $l21, $r22, $l22) = split(' ', $line2);

        # get rid of the 'j' in the L terms
        chop ($l11, $l12, $l21, $l22);

        print "Frequency: ", &engnot($freq), " Hz\n";
        print "      L1: ", &engnot(($L1 = $l11 / (2 * $pi * $freq))), "\n";
        print "      L2: ", &engnot(($L2 = $l22 / (2 * $pi * $freq))), "\n";
        print "      Q1: ", 2 * $pi * $freq * $L1 / $r11, "\n";
        print "      Q2: ", 2 * $pi * $freq * $L2 / $r22, "\n";
        print "      Lm: ", &engnot(($l12 + $l21) / (4 * $pi * $freq)), "\n";
    }
}
```

C.8 Miscellaneous Data Files

C.8.1 placedat

`placedat` contains input to `placescr.pl` which places components on the magnetic structure layouts.

```
b35n61:430:-480:R0
WIRE:430:-480:430:-100:top:50
C1812K:350:-130:R0
C1812K:510:-240:R0
C1812K:400:350:MR90
VIA:270:0:50:35:round
WIRE:270:0:400:0:top:50
403a-03:340:-5:R0
WIRE:400:430:0:430:bot:50
VIA:0:430:75:50:square
VIA:0:540:75:50:square
WIRE:300:-350:300:-600:top:100
```

```
WIRE:300:−350:300:−600:bot:100
```

C.8.2 `gpdat`

`gpdat` contains information on where to place the ground planes around the magnetic structures. It is processed by `dat2scr.pl` and `dat2ind.pl`.

```
−1000,1000,0,1000,1000,0,1000,500,0,10,5
−1000,500,0,−500,500,0,−500,−550,0,5,5
500,500,0,1000,500,0,1000,−550,0,5,5
−1000,−550,0,1000,−550,0,1000,−1000,0,10,5
−1000,1000,−62,1000,1000,−62,1000,500,−62,10,5
−1000,500,−62,−500,500,−62,−500,−550,−62,5,5
500,500,−62,1000,500,−62,1000,−550,−62,5,5
−1000,−550,−62,1000,−550,−62,1000,−1000,−62,10,5
```

Appendix D

Shunt Rectifier PCB Layouts and Magnetic Structures

Layouts for the shunt resonant rectifiers implemented in section 5.5 are provided in this appendix. In the layouts, diagonal top left to bottom right hatching indicates copper on the top layer of the board, whereas top right to bottom left indicates the bottom copper. Crosshatching is used where both the top and bottom copper are present.

In all cases, the input signal comes from the BNC at the bottom right edge of the picture. C_{shn} connects from the input trace to ground directly adjacent to the connector. L_{ser} connects from the input node to the trace above and to the left of the BNC. Since L_{ser} is provided entirely by the coupled magnetic structures when cancellation is employed, the input node is simply shorted to the input of the magnetic structure in these cases.

In the layout without cancellation, the cathode of the diode D connects from ground to the node adjoining L_{ser}; in the cases where cancellation is used, D is located in the same place, but is connected to the other end of L_1. L_2 is on the bottom side of the board, and connects to pads for L_{choke}, also located on the bottom of the board. When no cancellation is employed, the diode cathode is simply shorted to one of the L_{choke} pads.

The output node is located at the top of the layout, and provides room for several bypass capacitors and through-hole mounting for an output connector block.

The magnetic structures that were implemented are also represented in 3-dimensional renderings which give a more detailed structural view. The renderings were made using the `zbuf` utility provided with the FastHenry software package [53]. In all cases, the top winding is L_1 and the bottom winding is L_2. While it is not explicitly represented in the renderings, the connection between L_1 and L_2 is located on the short overlapping segments. On the PCB, this connection is implemented as a via.

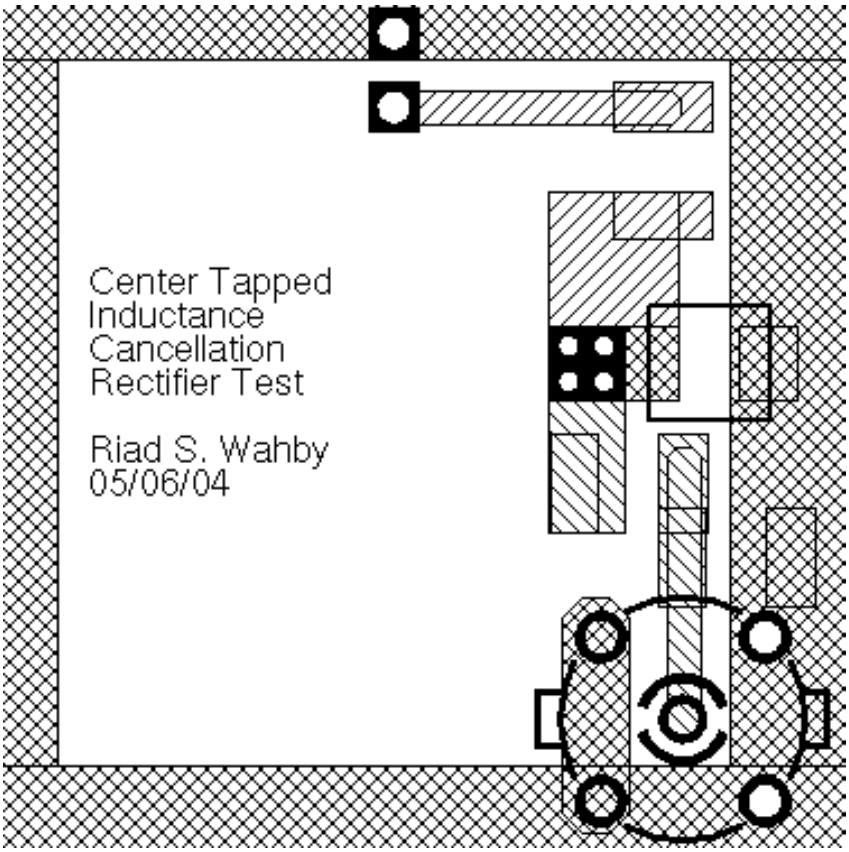
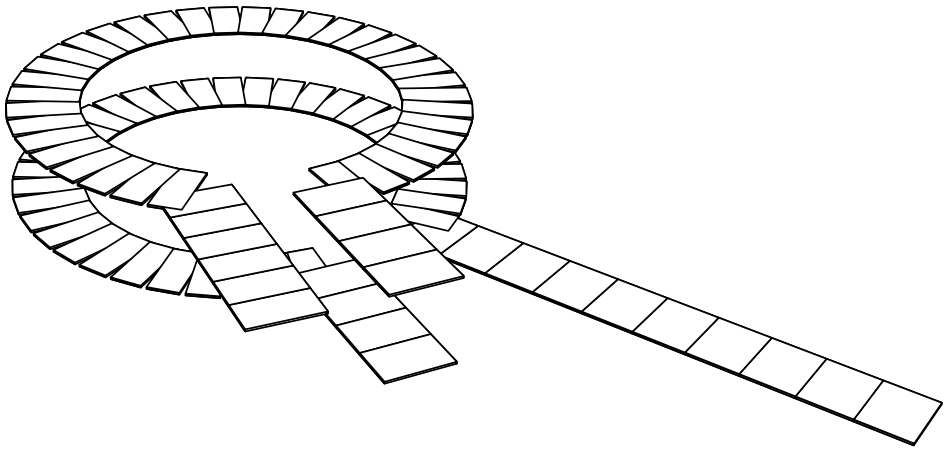
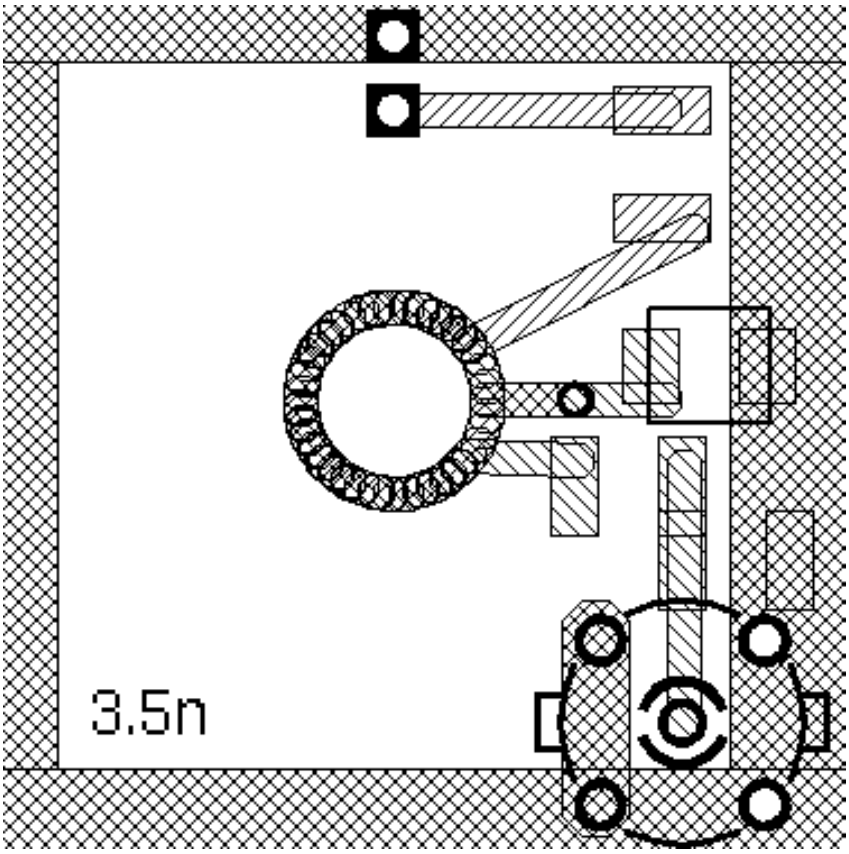


Figure D-1: Layout for shunt rectifier without cancellation.

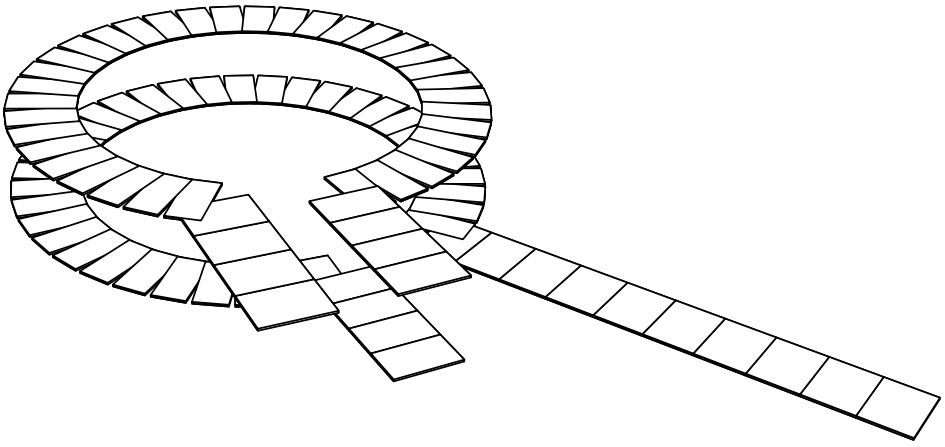


(a) 3D rendering of the magnetic structure.

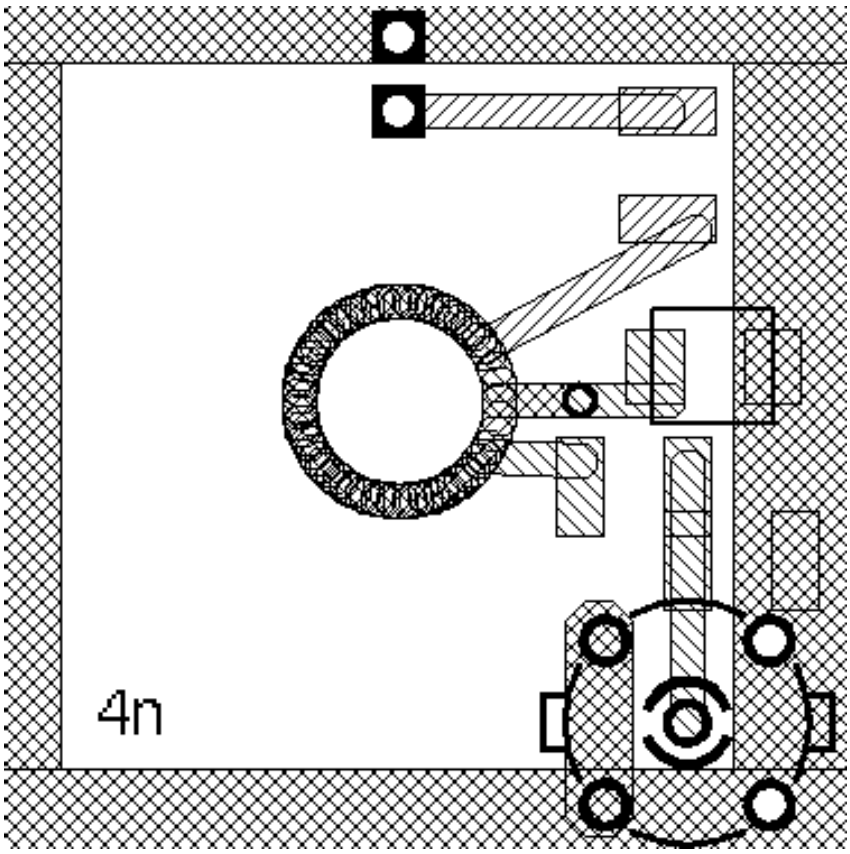


(b) PCB layout.

Figure D-2: $L_M = 3.5$ nH

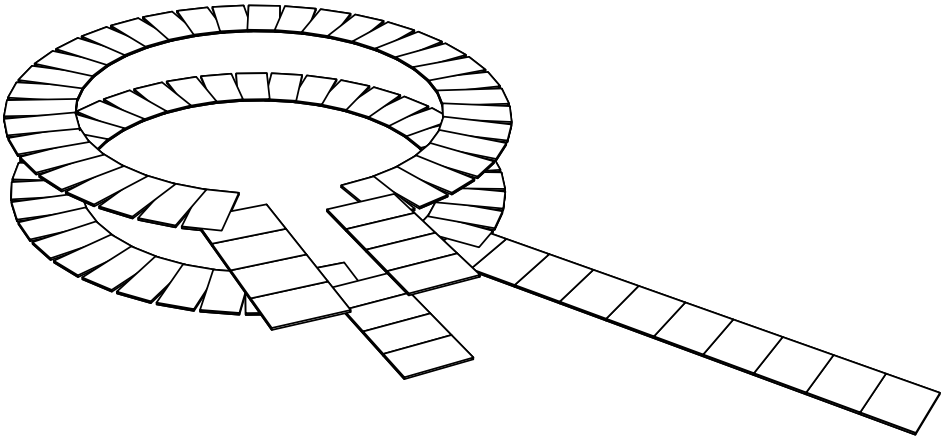


(a) 3D rendering of the magnetic structure.

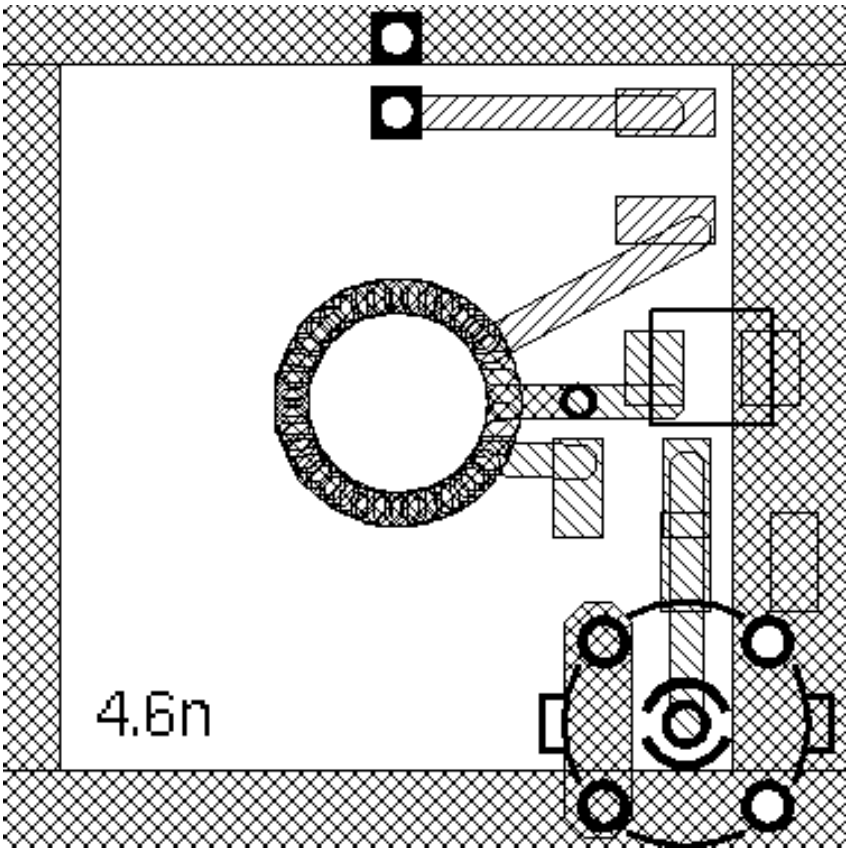


(b) PCB layout.

Figure D-3: $L_M = 4.0$ nH

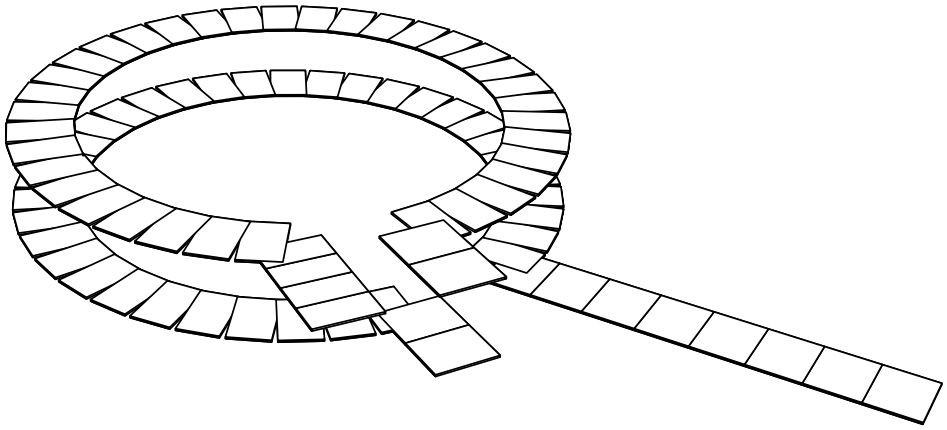


(a) 3D rendering of the magnetic structure.

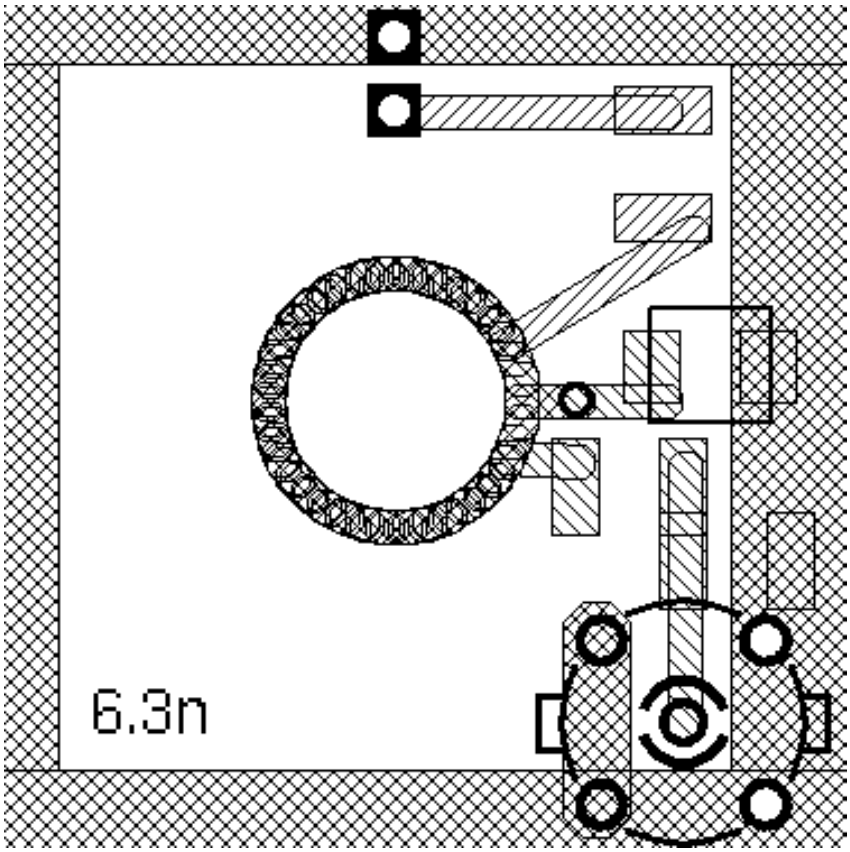


(b) PCB layout.

Figure D-4: $L_M = 4.6$ nH



(a) 3D rendering of the magnetic structure.



(b) PCB layout.

Figure D-5: $L_M = 6.3$ nH