Design of Miniaturized Radio-Frequency DC-DC Power Converters

by

Anthony D. Sagneri

B.S., Rensselaer Polytechnic Institute (1999)
S.M., Massachusetts Institue of Technology (2007)

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2012

Author_____
Department of Electrical Engineering and Computer Science
January 13, 2012

Certified by_____
David J. Perreault
Professor, Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by_____
Leslie A. Kolodziejski
Chairman, Departmental Committee on Graduate Students

Design of Miniaturized Radio-Frequency DC-DC Power Converters

by
Anthony D. Sagneri

Submitted to the Department of Electrical Engineering and Computer Science
on January 13, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

POWER electronics appear in nearly every piece of modern electronic hardware, forming an essential conduit from electrical source to load. Portable electronics, an area where a premium is placed on size, weight, and cost, are driving the development of power systems with greater density and better manufacturability. This motivates a push to higher switching frequencies enabling smaller passive components and better integration. To realize these goals this thesis explores devices, circuits, and passives capable of operating efficiently into the VHF regime (30-300 MHz) and their integration into power electronic systems of high power density.

A good integrated power MOSFET presages high-density converters. Previous VHF systems were demonstrated with bulky and expensive RF Lateral, Double-Diffused MOSFETs (LDMOSFET). We show that through a combination of layout optimization and safe operating area (SOA) extension integrated devices can achieve near-parity performance to their purpose-built RF discrete cousins over the desired operating regime. A layout optimization method demonstrating a 2x reduction in device loss is presented alongside experimental demonstration of SOA extension. Together the methods yield a 3x reduction in loss that bolsters the utility of the typical (and relatively inexpensive) LDMOS IC power process for VHF converters.

Passive component synthesis is addressed in the context of an isolated VHF converter topology. We present a VHF topology where most of the magnetic energy storage is accomplished in a transformer that forms an essential part of the resonant network. The reduced component count aids in manufacturability and size, but places difficult requirements on the transformer design. An algorithm for synthesizing small and efficient air-core transformers with a fully-constrained inductance matrix is presented. Planar PCB transformers are fabricated and match the the design specifications to within 15%. They are 94% efficient and have a power density greater than 2kW per cubic inch.

To take full advantage of good devices and printed passives, we develop an IC for the isolated converter having optimized power devices, and integrated gate driver, controller, and hotel functions. The chip is assembled into a complete converter system using the transformers and circuits described above. Flip-chip mounting is used to overcome bondwire parasitics, and reduce packaging volume. The final system achieves 75% efficiency at 75 MHz at 6W.

Thesis Supervisor: David J. Perreault
Title: Professor, Department of Electrical Engineering and Computer Science

Acknowledgements

First and foremost I thank Prof. Dave Perreault, my thesis advisor, for his guidance, support, and patience over the course of my time at MIT. He's an incredible source of knowledge and a good example to follow. To (Now) Prof. Juan Rivas who's talent and skill are second to none in this most esoteric of disciplines, and who was a friend through all the interesting times: Thank You.

In addition I make a special thanks to my thesis committee: Prof. John Kassakian, Prof. Jeffrey Lang, and Prof. Tomas Palacios for their support, advice, and help along the way.

National Semiconductor deserves a big boost: They sponsored the thesis work contained herein. A particular thanks to Dave Anderson, another incredible source of knowledge and a particular aid when non-technical problems arose. Also, he has a great car.

To Jackie Hu, whom I originally mistook for a big-glasses, big-sweatshirt nerd: You are a great friend who stuck with me through all the muck. It's impossible to express how much that has meant to me. Thank You.

To Grace Cheung, who deserves this separate paragraph more than any other: I'm grateful we crossed paths. The MIT experience would not have been complete without you. Thank You.

To George Hwang: You are a unique animal, the fusion of emotion and a determination that you have thus far chosen to point toward engineering to great effect. You have a lot left to do. Thank you for being a great friend and buttress in the stormy times (and the good ones!).

To Jut (Justin Burkhart): Thanks for reminding me that starting a company was why I came to MIT. To Vanessa Green: Thanks for teaching me (however didactically) that trust is the most essential element of entrepreneurship, whether that's trust of self, others, or just the universe.

Other lab folks: (Now) Prof. Yehui Han, (Now) Prof. Robert Pilawa, Olivia Leitermann, Bran(s)don Pierquet, Alex Hayman, Alex Trubisyn (collectively, "The Alexes"), Wei Li, Fergus Hurley Sam Chang, of course Jiankang Wang, Shahriar Khushushahi, Uzoma Orji, Warit Wichakool, you have been a great network of friendship, support, and technical knowledge at various points over the years.

MIT Cycling: Thanks! Couldn't have had more fun on a bike. In particular I want to call out A.J Schrauth, Alex Chaliff, Yuri Matsumoto, Zuzana Trnovcova, Chewie Chew, David Singerman, Zack Labry, and Nick Loomis. You guys made it fun.

I owe the most to my parents: Richard F., and Mary, without whom I wouldn't exist or be where I am. And to my siblings: Kris, Rich, David, and Catherine who have been many things over the years—there's still a lot to discover.

"Nothing in the world can take the place of Persistence. Talent will not; nothing is more common than unsuccessful men with talent. Genius will not; unrewarded genius is almost a proverb. Education will not; the world is full of educated derelicts. Persistence and determination alone are omnipotent. The slogan 'Press On' has solved and always will solve the problems of the human race."

Calvin Coolidge: Amhearst College, 1894.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

MINIATURIZATION is the pursuit of modern technology. Today's smart phone packs a computing wallop that would have been considered magical[1] in the days of the Apollo program. Yet the consumer expects still more and so goes a silicon arms race, rushing ever closer to that elusive limit of semiconductor physics. Expectations aside, delivering on the promise our potent portables portend—the society-shaping fusion of information and person—is no small technical feat. As minimum feature size drifts south of 20 nm more functionality is squeezed onto less silicon, igniting an appetite for more power at steadily increasing density. While clever circuit, device, and systems solutions strive to quench this thirst, the desire for increased functionality and the availability of more energy-dense batteries ensure that power consumption in portables continues to rise. The result is continual increase in the demands made upon the power subsystem.

Taken in the context of increasing integration, the power management subsystem now occupies a conspicuous area that could be otherwise spent on more functionality (several hundreds of millions of CMOS transistors, for instance), or smaller overall size. This is owing to the need for more power and the not-insignificant challenges that exist around shrinking the converter's energy storage. The latter is a thorny problem, and currently defines the limits on reduction of size, weight, and cost across the spectrum of power conversion.

One means of reducing the size of the energy storage is is to increase switching frequency [2]. This allows for passive components to shrink as the amount of energy storage required to service the converter function falls. A small increase in frequency is generally not effective. For instance, in order to manage loss in magnetics as frequency is increased it's often necessary to decrease flux density yielding a counter-intuitive increase in passive component volume. Instead, operating in the VHF regime (30 MHz-300 MHz) offers the potential to utilize air-core inductors or low-permiability RF magnetic materials to achieve an overall size reduction while maintaining or improving performance in other areas such as efficiency, bandwidth, and power density. VHF power conversion has been demonstrated for voltage and power levels ranging from about a watt and a volt to hundreds of watts and hundreds of volts. [3–8].

[1] The magic that is iPad is left aside in this assertion.

The approach to VHF power outlined in [3] has shown promise in discrete implimentations. However, while this has resulted in higher power density in many cases, significant reduction is still possible by going from discrete components to more integrated designs. This requires efforts on the fronts of power device development, passive components, circuits, and systems to be considered in concert. Exploration in these areas are the core of the thesis work presented here. Since a silicon power process that also provides CMOS is a prerequisite to integration, optimization of LDMOS (Lateral Double-Diffused MOSFETS) power devices was undertaken first. Once viable integrated power switches were demonstrated, work on passive components and circuit design was accomplished, ultimately leading to an attempt to increase converter density by using an IC designed for VHF converter operation in concert with a PCB substrate carrying a mix of embedded and surface-mount passives. The results are shown to be effective, if one overlooks an error in the IC design phase that led to an inability to demonstrate closed-loop operation without some external hardware.

In order to better frame the work presented here, the following two sections introduce a discussion of the loss mechanisms in hard switched power converters alongside the application of resonant techniques used to mitigate them. These considerations lie at the core of the VHF approach, but delivering on the promise of VHF to reach Lilliputian dimensions requires addressing the topics at the core of this work.

1.1 Losses in hard switched converters

A switched mode power converter constructed of ideal elements has no intrinsic loss mechanism. Rather, they arise inevitably from the use of real components. These losses, distributed among the active and passive components constrain not only the efficiency of the SMPS (switched mode power supply), but the size, cost, form-factor, and even converter responsiveness. Finding ways to beat these losses is, in a sense, tantamount to miniaturization.

On considering a typical dc-dc converter, one fact that becomes obvious is that the bulk of the system, that is its weight and volume, comprises the passive energy storage elements. Semiconductor devices, having benefited from tremendous improvement since their inception, occupy only a small fraction of a typical converter footprint. This is made clear in figure 1.1 showing a common implementation of a synchronous buck converter where switches, gate drives, controller, startup and protection circuits, and the housekeeping power systems are integrated onto a die and placed in a QFN (quad flat-pack, no-lead) package. The remainder of the components are the energy storage devices which require roughly an additional four times the board area (not accounting for interconnect) and nearly *six* times the

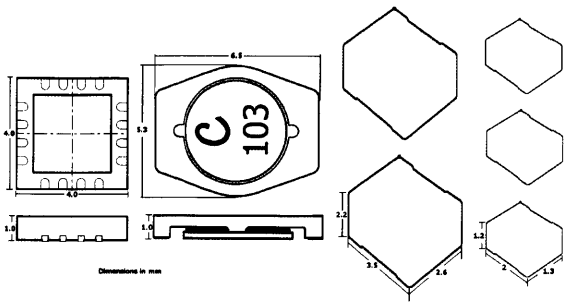
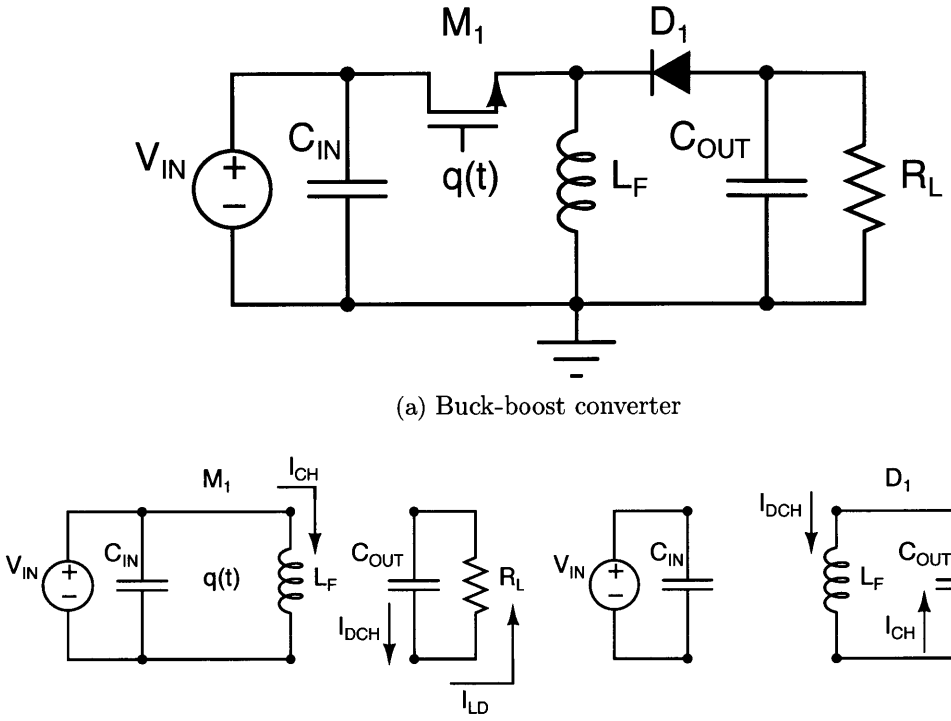


Figure 1.1: Synchronous buck converter active vs. passive volume. The synchronous buck converter components pictured will supply 7.5 W into 5 V. The QFN on the left encompasses the active switches, gate drive, control, and housekeeping functions. The remaining passive elements require 4 times the board area and 6 times the volume, not accounting for board interconnect.

volume. It is not surprising, then, that techniques to reduce converter footprint might be aimed at minimizing or eliminating passive energy storage.

Where the goal is to reduce the size of the energy storage components, there are two primary ways to proceed. Either energy density may be increased or total converter energy storage reduced. Increasing the energy density implies shrinking a device for a constant amount of storage. Even if this can be accomplished, given the physical constraints imposed by power dissipation, the increased losses that result often cannot be reconciled with good converter performance. Considering a solenoidal inductor, it is demonstrated in [9] that fundamental scaling between linear dimensions and flux- or current-carrying area causes inductor Q to decrease as α^2 where $\alpha < 1$ is a constant scaling each linear dimension. Similar relationships are enumerated in [10] for other geometries. In the case of capacitors, analogous problems arise. Where a given dielectric material is available, a lower bound exists on the capacitor plate separation for a set working voltage. Further, plate resistance also increases as plate thickness is decreased or plate area is increased, both are necessary to improve energy density. These conditions imply that the capacitor Q will become unacceptably low with continued scaling at a constant capacitance. Thus a host of factors — Q, dielectric breakdown, and dissipation — impose a maximum energy density on passive components. Unfortunately, practical densities leave something to be desired for converter size.

With very limited leeway to increase energy density, we turn our attention to reducing the required energy storage. The classic solution is to raise the switching frequency [2], thereby reducing the amount of energy processed per cycle, a condition that leads directly to smaller numerical values of inductance and capacitance. The flyback converter in figure 1.2 is a convenient means to an explanation.



(a) Buck-boost converter

(b) First part of cycle

(c) Second part of cycle

Figure 1.2: Buck-Boost Converter Illustrating Energy Storage Requirements. In the buck-boost converter, L_F acts as temporary storage. In the first part of the cycle L_F is charged by current I_{CH} while C_{OUT} holds up the output. In the second portion of the cycle L_F discharges into the load while replenishing C_{OUT}.

The buck-boost converter is an indirect converter. This type of converter transfers energy from the source to intermediate storage in the first portion of a cycle and then from intermediate storage to the load in the second portion of the cycle. The intermediate storage in the flyback converter is the inductor, L_F. As the switching frequency is increased and the amount of energy processed each cycle gets smaller, the numerical value L_F can be reduced and the inductor made physically smaller for constant energy density. The same applies to the capacitors C_{IN} and C_{OUT}. For instance, C_{OUT} must hold up the output voltage during the half of the cycle when L_F is charging. The holdup time is inversely proportional to frequency as is the associated RC time constant for a constant droop in output voltage. Another way to see that C_{OUT} can be reduced is to consider that R_L and C_{OUT} form a low-pass filter which attenuates the switching ripple. As the switching frequency increases the low-pass corner frequency moves up for a given attenuation, relaxing the capacitance requirement.

Though increased switching frequency attends less energy storage, it is not a technique that may be used haphazardly: A cohort of loss mechanisms arise rapidly to place limits on

the operating frequency. While not necessarily the largest of these, important frequency dependent losses in passive elements are limited almost exclusively to inductors and their magnetic materials. Most magnetic materials, used to increase inductance per unit volume, operate well at low frequency but have losses that rise rapidly otherwise. The basic trend is captured by the Steinmetz equation:

$$\overline{P_v(t)} = kf^{\alpha}B^{\beta} \tag{1.1}$$

where $\overline{P_v(t)}$ is the time-average loss per unit volume $[kW/m^2]$, B is the peak ac flux amplitude [Gauss], f is the frequency of sinusoidal excitation [Hz], and the constants k, α, and β are found by curve fitting. Examining 1.1 it is clear that for α greater than one (it's often in the range of 1-3) that the loss will rise briskly with frequency. Another important implication is that the core volume may be increased to reduce the flux density, trading increased size for higher frequency—the opposite of the desired effect[2]. In truth, the Steinmetz equation is only valid in a narrow range of situations, primarily where the excitation is sinusoidal and relatively low frequency. At high frequencies and under the non-sinusoidal excitation typical of power converters, the losses tend to be greater than predicted in the Steinmetz model and many different modeling approaches have been undertaken to get a more accurate prediction (for instance, [11]). The upshot, however, is that most bulk magnetic materials are not suitable for operation at frequencies much higher than a few megahertz.

One way of avoiding magnetic core losses is to do away with the magnetic core. The lower energy density demands even higher operating frequencies, but to the extent that the frequency can be increased, the magnetic loss picture looks much better. For a simple air core inductor, the inductance and resistance are determined primarily by geometry and the choice of conductor. Inductor quality factor Q is:

$$Q = \frac{\omega L}{R} \tag{1.2}$$

In this simple relationship, expressing the ratio of energy stored to energy lost per cycle, Q increases with reactance and decreases with resistance. The frequency dependence of R and L are very difficult to calculate for any geometry other than isolated straight wires. In general, skin effect, proximity effect, and interwinding capacitance affect both L and R [10]. If the proximity effect and the interwinding capacitance are ignored, the skin effect results

[2]Often loss becomes the limiting factor at high frequency and flux derating is necessary to avoid excessive heat build up. Thus at high frequency cored inductors can actually get *larger*.

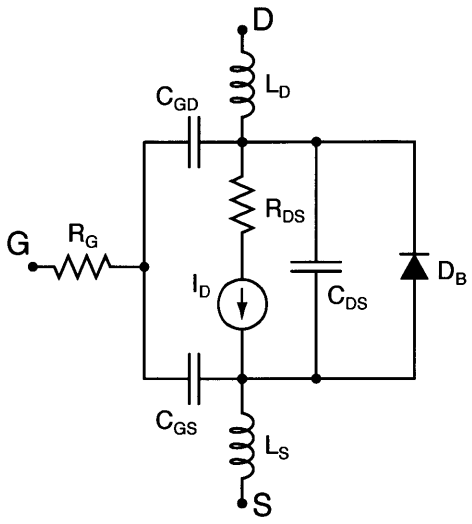


Figure 1.3: A MOSFET model including parasitic elements usually important in hard switched dc-dc converter design

in approximately a square-root increase in resistance with frequency. Since reactance rises linearly under these assumptions, then Q will increase $\propto \sqrt{f}$. Measurements of inductor Q and information available from manufacturers of air-core RF inductors indeed show that Q increases with frequency as a general trend.

The seemingly synergistic effect of increasing Q with frequency for air-core inductors is only advantageous provided that the other frequency dependent loss factors are dealt with. These losses are associated with active semiconductor devices. Semiconductor losses can be divided into three main mechanisms for MOSFETs: conduction loss, switching loss, and gating loss. A MOSFET model including the parasitic elements usually considered in dc-dc converter design is shown in fig. 1.3.

Conduction loss, due to the effective resistance of the channel, the lightly doped drain region (LDD), and metal/bondwire resistance, is only slightly frequency dependent[3]. Switching loss, however, depends significantly on frequency. It is helpful to further divide switching loss into overlap loss and losses resulting from discharge of the drain-source capacitance, C_{DS}. Overlap loss refers to the condition where the MOSFET supports simultaneous voltage and current at its drain-source port and thereby dissipates power. This condition arises from the need to charge or discharge the device channel through finite source impedance (whether this impedance arises externally or as a result of device parasitic resistance and inductance) which imposes a minimum on switch transition times. Simplified models of overlap loss

[3]At typical operating frequencies the quasistatic assumptions for MOSFETs are valid, so the channel and LDD components of R_{DS-ON} are constant. Bondwire resistance is usually a small enough component that skin effect only accounts for a small change in the total R_{DS-ON}.

parameterized in converter nominal voltage (V_O) and current (I_O), and MOSFET rise (τ_r) and fall (τ_f) times are readily available [2, 12]:

$$E_r + E_f = kV_O I_O(\tau_r + \tau_f) \tag{1.3}$$

The constant k reflects the circuit in which the device is used and varies between 1/6 and 1/2 depending on whether the load is purely resistive or clamped inductive. Since this result is basically fixed once the device and circuit are chosen, the energy per transition ($E_r + E_f$) is also fixed. Therefore, as switching frequency rises, so does overlap loss.

The loss due to C_{DS} occurs at device turn on, when the energy stored on the output capacitance is dumped into the switch yielding a loss than can be roughly approximated as: $\frac{1}{2}C_{DS}V^2_{GATE-PK}f$. This effect can be significant even at frequencies well below a megahertz—C_{DS} is usually fully charged just before turn-off.

Gating loss results from charging and discharging the input capacitance, $C_{ISS} = C_{GS} + C_{GD}$. Calculating the gating loss is somewhat complicated by the presence of C_{GD} which is multiplied according to the Miller effect during transitions. In lateral MOSFETs where C_{GD} tends to be very small and its effects can be ignored, the gating loss is approximately expressed as:

$$P_{GATE} = C_{GS}V^2_{GATE-PK}f \tag{1.4}$$

This reflects that the loss is associated with the loss of charging a capacitor from a dc voltage through a resistor, $\frac{1}{2}CV^2$, and the subsequent dumping of the stored energy once per cycle. In other types of MOSFETs, such as vertical DMOS and even some lateral devices, C_{GD} is a significant portion of C_{ISS} and the effects can't be ignored. Then the gate power is usually expressed in terms of the total charge required per cycle to enhance the device:

$$P_{GATE} = Q_G V_{GATE_{PK}}f \tag{1.5}$$

In both cases the frequency dependence is clearly linear. This mechanism becomes important at switching frequencies of a few megahertz and beyond where gating loss for typical devices can range from hundreds of milliwatts to several watts.

Diodes also account for a fraction of the converter loss budget. All diodes have an associated forward voltage drop, V_F, that combines with the forward current, I_F, and resistive losses in the bulk regions to result in diode conduction loss. This mechanism is not explicitly frequency dependent. PN-junction diodes and PIN diodes, however, do have a frequency dependent loss mechanism - reverse recovery. Reverse recovery names the process in which stored minority carriers are removed during commutation. During the reverse recovery time, τ_{RR}, the carriers are extracted across a constant voltage. Since this time is related to the amount of stored charge and the impedance of the external circuit, τ_{RR} is fixed for a given configuration. Therefore, the energy wasted per cycle to reverse recovery is constant implying frequency dependence. Schottky diodes, which are formed as metal-semiconductor junctions are majority carrier devices. They do not suffer heavily from reverse recovery losses, but are only available with breakdown voltages below about 120 volts.

1.2 Resonant Power Conversion

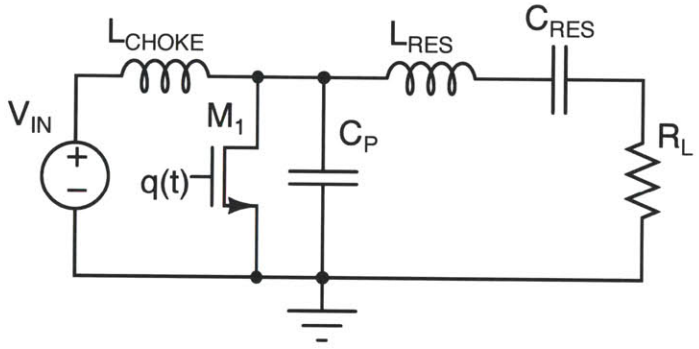
Resonance, usually ascribed to systems with complex poles displaying oscillatory behavior, is of some significance in power conversion. In filtering, for example, it plays a role to develop large immitance in comparatively little volume[4]. Here we look at resonance as a means to push back converter loss mechanisms and realize operation in the very high frequency regime (VHF, 30 MHz - 300 MHz).

A number of converter topologies exist that draw from RF amplifier techniques to achieve efficient energy conversion [5, 14–21] at high frequencies. These designs rely on reactive networks to shape the switch voltage and current and reduce switching loss. The class E converter, fig. 1.4a, is a widely practiced topology whose network enforces a zero-voltage switching (ZVS) opportunity at turn-on. Its basic operation can be classified as indirect. The inductor L_{CHOKE} is an open at the switching frequency, ensuring that only dc current flows from the source. With no dc path to the load energy from the source must first be stored on the switch shunt capacitor C_P. The energy stored in C_P then rings towards the load in a cycle that is determined by the switching function, q(t), and the resonant tank formed by the load, L_{RES}, and C_{RES}. It functions by ringing the energy on C_{DS} to the load once per cycle. When these components are tuned according to [14, 15], the drain voltage will naturally return to zero as the energy in C_P rings toward the load. At this point, the switch may be turned on with minimal loss. This mode of operation avoids the losses

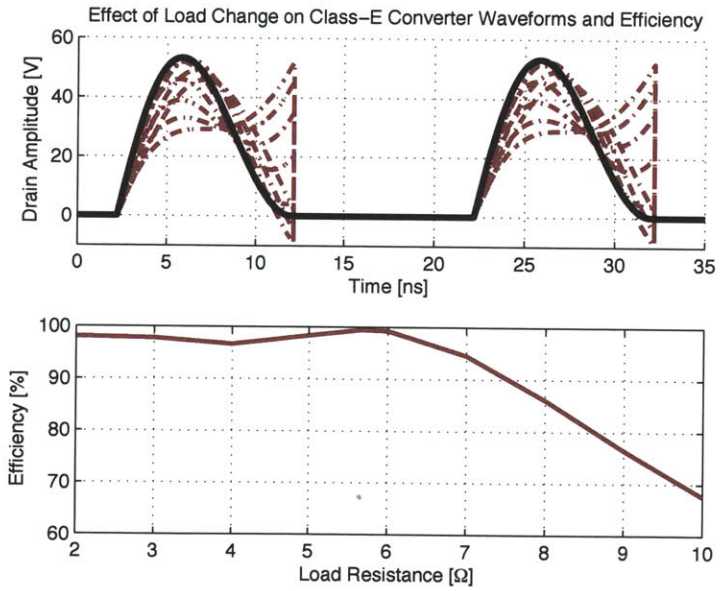
[4]Series and parallel resonant filters can be used to shunt or block ripple in power converters. It was demonstrated in [13] that by using resonance, filter element volume could be reduced by better than a factor of three.

usually ascribed to the switch drain-source capacitance and largely avoids overlap loss, as well.

In practice, the drawbacks of such resonant topologies have prevented them from seeing widespread use. To begin with, the load range is severely restricted when compared with the 100:1 or better range achievable with conventional converters. Resonating losses from circulating reactive currents become significant as the load is reduced, hurting efficiency.



(a) Class E inverter



(b) 50 MHz Class E drain voltage waveforms

Figure 1.4: Resonant topologies often suffer from limited load range in power conversion applications. Here, the Class E inverter waveforms are pictured as the load is varied from $\frac{1}{2}R$ to $2R$. The properly tuned waveform is displayed in heavy black. The loss of ZVS and negative impact on efficiency are evident.

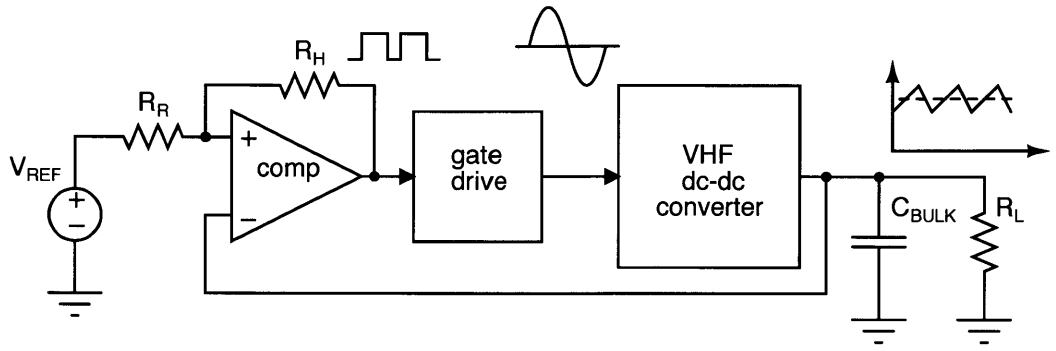


Figure 1.5: Schematic depiction of VHF converter under on-off modulation. The closed-loop system keeps output voltage constant allowing the converter to deliver a constant power (effectively it sees a constant load) at it's most efficient point. Actual power delivered to the load depends on the duty ratio of the control signal.

The situation is made worse in the many cases where the load is an integral part of the resonance. Then, any change in load disrupts the ZVS condition and switching loss inevitably arises. This situation is depicted in fig. 1.4b. Further difficulty arises because duty ratio control is often not possible. Instead, control is achieved by varying the switching frequency. The resulting poor dynamics worsen with frequency and place an artificial upper bound on practically achievable switching speed. Many resonant converters also suffer from high peak switch stresses. The class E converter, in particular, has peak drain voltages rising as high as 4.4 times the dc input voltage [22]. This is particularly troublesome where integration is concerned because integrated devices tend to have lower breakdown voltages.

Several of these issues can be resolved by partitioning the energy storage and control functions [3, 4, 23]. Instead of controlling the output by varying the switching frequency, on-off modulation of the converter determines the fraction of output power delivered (see fig. 1.5). When the converter is on, it delivers a fixed power maintaining ZVS and maximum efficiency. When off, no power is delivered and there are no associated resonating losses. Under these conditions, the load range is a function of the minimum achievable modulation index. Such operation allows the network to be tuned to enforce ZVS at one particular operating point. The result is maximum efficiency, better dynamics, and higher achievable operating frequency. The fact that this mode of operation allows much higher switching frequency is self-reinforcing—high frequency means less energy storage so the converter can be started and stopped more rapidly and achieve a wider load range.

1.3 Contributions and Organization of the Thesis

Raising the switching frequency is a well known means of reducing required storage in power converters. It is precisely this reduction that can put inductor and capacitor values into the range that they might be considered for integration. For inductors, in particular, VHF operation allows magnetic materials to be set aside, avoiding the difficulty and expense of incorporating them on chip or on board. Capacitors benefit from similar consideration primarily owing to the much smaller required values. However, simply moving to higher frequency is not enough. Existing power devices that satisfy VHF operation tend to be RF LDMOSFETs which are usually over-packaged, optimized for linearity versus more desireable characteristics for power transfer, usually only come in a few discrete ranges, and are difficult to drive owing to gate-loop parasitic inductance as well as drain feedback via C_{RSS}. In addition, control, driver, and hotel circuits implemented with discrete components become a significant portion of converter size after power-stage passives are shrunken via increased switching frequency. When these facts are considered jointly, it is clear that integrated LDMOS would be of great benefit, since the control and power devices can be absorbed onto the same die. The prerequisite here is that the power devices be "good enough." In conjunction with an on die methodology for the semiconductors, furthering the goal of being small and cheap includes addressing passive design issues that permit batch fabrication and lower component counts. Finally, overall system design needs consideration to achieve the best tradeoffs for size and performance. The ensuing chapters address the topics in turn.

Chapter 2 examines methods to improve integrated LDMOS performance for use in VHF applications. Most LDMOS available in integrated form have been optimized primarily for the hard-switched application. This has implications both in terms of the differing loss mechanisms between the hard-switched and VHF soft-switched operating modes and the safe operating area (SOA). The chapter first examines the loss mechanisms peculiar to soft-switched VHF operation, then presents a methodology for optimizing device layout in their consideration. The improvement in performance is substantial (over 50% loss reduction in the semiconductor) post-optimization. Further work is accomplished as regards the SOA. Normally, SOA in an LDMOS is set by a combination of $B_{V_{DSS}}$ and hot-carrier effects. With a switching trajectory that strays to simultaneous high voltage and current, the hard switched case tends to find its most severe constraints in the hot-carrier regime. In this chapter we look at relaxing SOA requirements based on a soft-switching trajectory and provide reasonable evidence that working voltages much closer to $B_{V_{DSS}}$ may be used with converters like the Φ_2. As a result, devices with lower per-unit capacitance yield higher performance. Simultaneous optimization of layout and relaxation of the SOA yield

a greater than 70% reduction in device loss showing integrated LDMOS to be feasible for VHF operation.

The networks that are utilized to achieve VHF operation come with the caveats of tuning and component count. The latter not only makes the converter more difficult to shrink (since more power-stage interconnect is necessary), but also harder to tune. Chapter 3 looks at a means to reduce component count and provide repeatability for the passives in the resonant power stage. Specifically, a technique for synthesizing a transformer that provides both galvanic isolation and the bulk of the magnetic energy storage is presented. By absorbing some of the inductors into the transformer parasitics, the component count is reduced. Printing the transformer on a PCB helps to mitigate variability and achieve more consistent tuning. However, these requirements necessitate that the transformer have a fully-specified inductance matrix. Since a given inductance matrix can be realized by any number of physical transformer implementations, we examine how to synthesize devices that both meet the electrical requirements and have a good loss-volume tradeoff in this inverse-problem space.

Chapter 4 develops an isolated Φ_2 converter. This converter is both a platform to utilize the transformer designs of Chapter 3 and a demonstration of furthered circuit design techniques. A number of design tradeoffs in the circuit regime arise during the development of the isolated Φ_2 converter. These are addressed and their implications exposed. Challenges around diode performance and modeling at VHF also cropped up as part of the isolated Φ_2 converter design. This investigation and the results of diode performance studies are presented. A thermal resistance model of the converter system is created and used to model the loss distribution in the converter. This allowed informed iteration to improve the overall performance of the system as well as a degree of validation of the VHF loss modeling efforts undertaken to date. The combination of these efforts yields an isolated VHF power converter with good power density.

The goal of this work, miniaturization of power converters, is examined most directly in Chapter 5. An IC power process is chosen as the basis for a converter design. Optimization of the available power devices is followed by the design of a VHF gate driver that takes advantage of the integrated regime and demonstrates superior performance to thus-far implemented discrete designs. A number of challenges are overcome by taking advantage of on-die opportunities such as more precise timing and the relative absence of loop inductance among critical circuit components. This allows for a more-integrated version of the isolated Φ_2 converter of Chapter 4. Control and hotel components are pulled on die while magnetics are printed in the board and capacitors remain small surface-mount components.

The concluding chapter summarizes the results of the work and looks forward to future potential.

Chapter 2

Power Device Optimization

2.1 VHF Device Loss Model

2.1.1 Overview

SEMICONDUCTOR device losses place critical limits on the design and performance of power converters. As a result, significant effort has been devoted to the optimization of power devices. Most converters operate under hard-switching conditions, or at frequencies below a few megahertz, and optimization has focused on reducing loss under these conditions. This has led to devices that are very good for these applications, but do not realize the potential of power silicon in the VHF regime. In this work, optimization is accomplished for the set of device losses that result when soft switching is employed to attain very high switching frequencies. This requires a model that captures the loss mechanisms and their scaling behaviors such as the one proposed in Section 2.1.2.

2.1.2 VHF Device Losses

To construct a model to optimize devices for VHF operation it is first necessary to consider the loss mechanisms of interest and their scaling behaviors. In general, to achieve extreme high frequency operation requires a means of rescaling or otherwise mitigating frequency-dependent loss. One example of a circuit topology that achieves this result is the Class-Φ_2 converter illustrated in Figure 2.1 [6]. It employs fully-resonant soft-switching and soft-gating to enable efficient operation in the VHF regime. The application of these techniques render overlap loss and capacitive discharge losses at both the gate and drain ports insignificant. This permits scaling the switching frequency until a new set of frequency dependent losses obtains. These VHF losses derive from the recirculation of energy stored in the device parasitics through lossy media, rather than its complete dissipation. As a result their scaling behaviors and values (normalized to converter output power) differ. These are the parameters that must be considered when optimizing devices for the VHF regime.

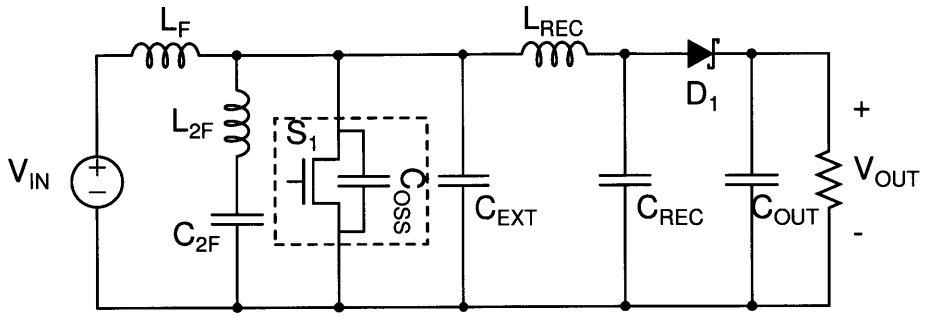


Figure 2.1: Φ_2 resonant boost converter with C_{OSS} and the external capacitance C_{EXT} explicitly drawn.

The Φ_2 voltage and current waveforms (Figure 2.2) elucidate the device loss and scaling behaviors alluded to above. The drain- and gate-voltage waveforms, v_{DS} and v_{GS} respectively, are plotted together with the drain current. The latter is subdivided into conduction current, i_{COND}, that flows through the active channel when the device is turned on and displacement current, i_{COND}, that flows through the device output capacitance, C_{OSS}, when the device is turned off. Owing to a soft-switching trajectory enforced by the Φ_2 network [24], v_{DS} and i_{COND} have almost no overlap and therefore no overlap loss over a switching cycle. Similarly, before the device is commutated v_{DS} approaches zero and capacitive discharge loss is also eliminated. This mitigation of switching loss is accomplished through the resonant action of the converter network and any number of similar fully-resonant schemes may be employed to reduce gating loss.

Of the device losses that dominate VHF operation, conduction loss maintains the same scaling behavior as the hard-switching case. It behaves as an i^2R loss. The RMS conduction current, $i_{COND-RMS}$, is independent of frequency as is the on-state resistance, R_{DS-on}, to first order. Therefore, even as frequency is scaled into the VHF regime, conduction loss remains significant and sets the minimum device area necessary to process a given amount of power. Indeed, the conduction loss normalized to output power is higher than direct hard-switching topologies such as the buck and boost converters because the RMS current in VHF converters is augmented by the resonating currents necessary to stem the frequency dependent losses. However, the tradeoff remains favorable over a few orders of magnitude in switching frequency and allows for efficient converter designs into the 100 MHz regime with typical silicon technology [24].

In contrast to conduction loss, the frequency-dependent losses behave differently from the hard-switching case. With overlap and capacitive discharge mitigated, what remains for loss when as the drain-source voltage transitions is the circulating current i_{DISP}. This current circulates through the device output capacitance. A resistance, R_{OSS}, comprising the drain access resistance, the bulk resistance, and drain-source metal resistance, appears in series

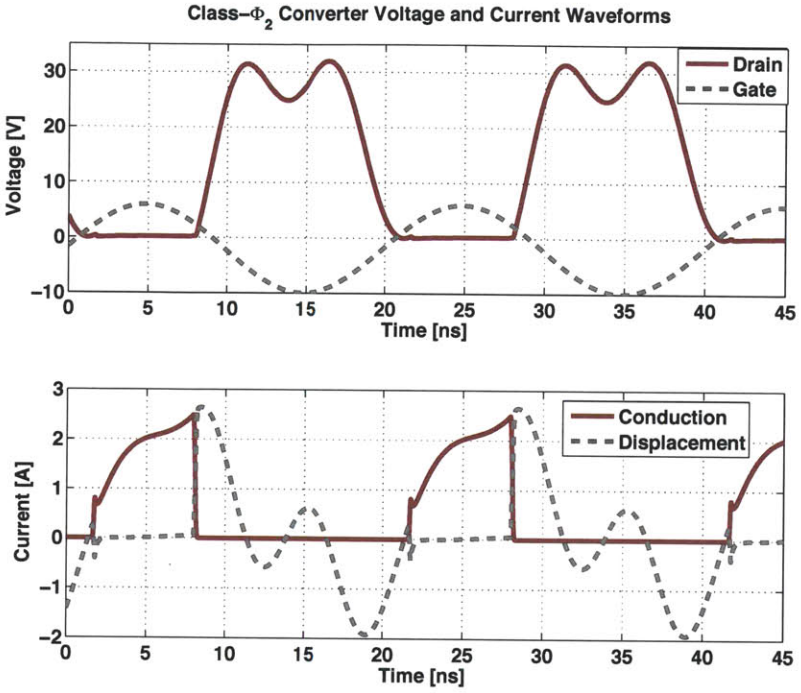


Figure 2.2: Simulated Φ_2 resonant boost converter switch voltage and current waveforms.

with C_{OSS} giving rise to loss. As a result the displacement loss, as it is referred to here, takes the form of an i^2R loss. Gating loss under the assumption of resonant gating takes the same i^2R dependence, since a current i_{GATE} circulates through the gate capacitance and its equivalent series resistance, R_{GATE}. R_{GATE} is composed of the source access resistance, poly resistance, and gate metal resistance.

An important consequence of the i^2R scaling of the frequency-dependent losses under soft-switching is their behavior as a function of frequency. This can be determined by establishing how i_{DISP} and i_{GATE} scale. In each case, the currents flow in a circuit branch that comprises a device capacitance in series with an equivalent resistance where the impedance is dominated by the capacitance. As frequency scales, the capacitor impedance falls linearly resulting in a linear increase in the branch current. This implies that both displacement and gating losses scale with the square of frequency, since loss is dependent on i^2_{RMS} in each case [1]. In contrast, both gating and switching losses under hard gating scale proportionally to frequency. With respect to device parameters, an increase in capacitance corresponds

[1] In the case where the switching frequency is large enough that the transition times are on the order of an RC time constant (eg., $R_{OSS}C_{OSS}$ and $R_{GATE}C_{GATE}$), resonant and hard-switching losses become similar in value. Fortunately, the frequency where this happens for switches suitable for VHF operation is well beyond what is considered here.

Table 2.1: VHF vs. Hard-Switched Loss Mechanisms

Loss Mechanism	Hard-Switched	Soft-Switched VHF
Conduction	$\propto I^2_{cond,RMS} R_{DS}$	$\propto I^2_{cond,RMS} R_{DS}$
Gating	$\propto C_{ISS} f_{SW}$	$\propto C^2_{ISS} R_{GATE} f^2_{SW}$
Off-State Conduction	N/A	$\propto C^2_{OSS} R_{OSS} f^2_{SW}$
Overlap	$\propto f_{SW}$	N/A
Cap. Discharge	$\propto C_{OSS} f_{SW}$	N/A

to a proportional increase in current and a square-law increase in loss while scaling with respect to resistance is linear. When this is coupled to device geometry, device width and frequency-dependent losses are directly proportional to first order. Table 2.1 outlines the device loss mechanisms and their scaling behaviors for both hard- and soft-switching cases.

The loss mechanisms discussed above are captured in Figure 2.3. It is a simplified where the resistances R_{DS-on}, R_{OSS}, and R_{GATE} correspond to the three important VHF device loss mechanisms: conduction loss, displacement loss, and gating loss. C_{ISS} and C_{OSS} are the lumped input and output capacitances. The coupling from the drain to the gate via C_{GD} is ignored in favor of lumping it with the input and output capacitances. This simplification is possible because a prerequisite of VHF operation is a small C_{GD} relative C_{GS}, therefore the current error introduced by lumping is small.

The primary motivation for using the simplified model in Figure 2.3 when many substantially more complex models exist in the literature is its ease of application to the optimization problem. In this case, the drain- and gate-voltage waveforms are enforced by resonant networks at both the drain and gate ports. Since these are determined at the design-time of the converter, the total device loss for a switch implementation is easily found once the values of the branch elements in the model are known. This significantly reduces the computational requirements necessary for a device optimization over the use of more complex compact models.

Equation 2.1 makes explicit the relationships among device, circuit, and loss. It parameterizes loss in two separate sets of variables, the intrinsic resistances and capacitances of the semiconductor device and circuit constants (K_1, K_2, K_3) derived from the circuit in which the device is employed. This facilitates optimization of the device because once a circuit design is established device performance is only a function of the intrinsic characteristics of the device, which are in turn related to the semiconductor process and layout geometry. Regarding the circuit constants, K_3 is shown for sinusoidal resonant gating. Other schemes such as trapezoidal resonant gating [3] result in different relationships. The

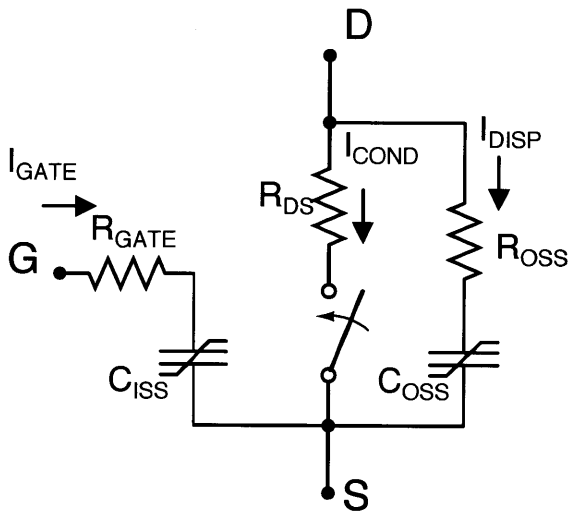


Figure 2.3: MOSFET model with loss elements relevant under soft-switched VHF operation

currents, $i_{COND,RMS}$ and $i_{DISP,RMS}$ are circuit dependent and may be found by SPICE simulation, or directly calculated depending on the circuit topology.

The addition of an external capacitance in parallel with C_{OSS}, C_{EXT}, is a technique often used for VHF converters. It establishes a particular drain-source impedance for proper circuit operation [6, 24]. For a given converter design, the total drain-source capacitance is held constant. In the case of device optimization, where minimizing loss dictates a certain C_{OSS}, C_{EXT} is adjusted to compensate the total drain-source capacitance. This allows the optimization of the device without requiring that the circuit parameters be recalculated. It also permits trading the conduction loss against the displacement and gating losses because total device area is scalable independent of the circuit design. For instance, in the case of displacement loss the total circulating current during the off-state is shared between C_{OSS}, a relatively lossy capacitance, and C_{EXT}, a capacitor with much higher Q. Therefore, reducing die area corresponds to a decrease in displacement loss as C_{EXT} carries a larger fraction of the off-state circulating currents. These relationships typically lead to an optimal device size as discussed in Section 2.1.3.

Table 2.2: Measured Device Parameters

Parameter	MRF6S9060	Integrated LDMOS (F)
R_{DS-ON}, V_{GS} = 8 V, 25°C	175 mΩ	200 mΩ
C_{OSS}, V_{DS} = 14.4 V	50 pF	132 pF
R_{OSS}	170 mΩ	500 mΩ
C_{ISS}	110 pF	275 pF
R_{GATE}	135 mΩ	1300 mΩ
P_{TOT}	213 mW	915 mW

$$
\begin{aligned}
P_{TOT} &= P_{cond} + P_{cond-off} + P_{gate} \\
P_{cond} &= K_1 \cdot R_{DS-ON} \\
P_{cond-off} &= K_2 \cdot R_{OSS,eq} \cdot C_{OSS,eq}^2 \\
P_{gate} &= K_3 \cdot R_{GATE,eq} \cdot C_{ISS,eq}^2 \\
\\
K_1 &= I_{cond,RMS}^2 \\
K_2 &= \left(\frac{I_{disp,RMS}}{C_{TOT}} \right)^2 \\
K_3 &= 2(\pi \cdot V_{gate,AC-pk} \cdot f_{SW})^2 \\
\\
C_{TOT} &= C_{OSS,eq} + C_{EXT}
\end{aligned}
\tag{2.1}
$$

The model outlined above can be used to make comparisons between devices given a target power converter design. Here a Class-Φ_2 resonant boost converter switching at 50 MHz is used as a case study. This design has V_{IN} = 12 V, V_{OUT} = 33 V, P_{OUT} = 12 W, C_{TOT} = 143 pF, $i_{disp,RMS}$ = 954 mA, $i_{COND,RMS}$ = 1040 mA, and $V_{GATE,AC-pk}$ = 8 V. To establish a performance baseline a discrete commercial RF LDMOSFET, the Freescale MRF6S9060, is compared to a custom LDMOSFET fabricated on an integrated BCD power process. Table 2.2 shows that the commercial part dissipates only 213 mW, while the integrated device dissipates 915 mW [2]. This difference motivates a desire to improve performance via techniques such as the optimization described below.

[2]Unlike the BCD MOSFET, which is implemented in a process that provides a range of device types and functions at low cost, the RF LDMOSFET enjoys a process designed exclusively for high-performance radio-frequency transistors employed in applications such as cell-phone base station amplifiers. This creates a substantial fraction of the performance difference

2.1.3 Device Scaling Considerations

By applying scaling relationships to intrinsic device parameters, operating frequency, and circuit components it is possible to identify an optimal ratio between device area and converter power as well as an optimal operating frequency. The applied scalings assume first-order relationships, for example doubling device area doubles each capacitance and halves each resistance. The true scaling relationship between geometry and parasitic elements is complicated by non-linear effects, such as junction sidewall capacitances that scale with perimeter rather than area and coupling of electric field lines in multiple dimensions. The additional complexity can be accounted for by performing full device optimization, but the first order rules are quite useful because once an optimal device is found in the middle of the scaling range, the device parameters will scale linearly over a wide range until the effects mentioned above come into play.

The parameters of interest are the device losses: conduction, displacement, and switching losses, and their behavior as device area, A, switching frequency, f_{SW}, and converter output power, P_{OUT} are scaled. For the first-order model it is assumed that: i) capacitance scales in direct proportion to effective width, ii) resistance scales inversely with effective width, iii) i_{DISP} is proportional to P_{OUT} and f_{SW}, iv) i_{cond} is proportional to P_{OUT}, v) C_{TOT} is proportional to output power, and vi) i_{GATE} is proportional to f_{SW}. Under these assumptions, the following relationships are established:

$$\begin{aligned} P_{cond} &= k_1 \cdot \frac{P_{OUT}^2}{A} \\ P_{disp} &= k_2 \cdot f_{SW}^2 A \\ P_{gate} &= k_3 \cdot f_{SW}^2 A \end{aligned} \tag{2.2}$$

Normalizing by output power yields:

$$P_{TOT} = k1 \cdot \frac{P_{OUT}}{A} + k2 \cdot \frac{f_{SW}^2 A}{P_{OUT}} + k3 \cdot \frac{f_{SW}^2 A}{P_{OUT}} \tag{2.3}$$

where k_1, k_2, and k_3 are constants to relate the scaling parameters to the as-measured loss. Equation 2.3 implies that an optimum ratio between device area and output power exists given the choice of circuit and semiconductor process because the conduction loss term has a power-area dependence opposite that of the frequency-dependent terms. This is

illustrated in the top plot in Figure 2.4, which shows the conduction loss (curved lines) and the frequency-dependent losses (displacement + gating–straight lines) versus normalized area and parameterized in P_{OUT}. At each output power level the conduction and frequency dependent losses cross and an optimum area exists. However, as power is scaled the optimum area scales in direct proportion exposing the optimum ratio between device area and power. The latter is intuitive upon imagining the paralleling of two identical converters operating at the same power and efficiency. The device area doubles along with the output power and branch currents maintaining a constant loss density and equivalently, efficiency.

The middle plot in Figure 2.4 compares the conduction and frequency-dependent losses versus normalized device area with frequency as a parameter. Conduction loss only appears as a single line because it is independent of frequency. The thin, solid lines are the total device loss for each frequency. As frequency is increased, the switching and gating losses increase quadratically as expected. This results in a continually decreasing optimal area. Comparing the total loss at the optimal area for each frequency point reveals a linear dependence despite a quadratic increase in the frequency-dependent losses. This obtains because area scaling of the device with converter design frequency allows the exchange of frequency-dependent loss for conduction loss.

Bottommost in Figure 2.4 is a plot of the total device loss, air-core inductor loss, and total converter loss versus normalized frequency. The device loss is plotted at the optimal device area for each frequency point. The quadratic tail is an artifact. It arises at very low frequencies because the maximum normalized area was limited to 10. Air core inductor loss is approximated as inversely proportional to the square root of frequency, because a linear increase in reactance (and therefore inductor Q) is partially offset by a square-root rise in AC resistance owing to skin effect. The device loss and inductor loss have opposite behaviors and an optimum frequency exists given the circuit topology, process, and intended operating conditions. For the power semiconductor process and circuits considered here, this ranges between about 50 MHz and 100 MHz.

Consideration of the scaling behaviors outlined above provides what is essentially the outer loop on the device optimization discussed below. For a given converter power and frequency, one can perform a full optimization on the device geometry and then iterate over width. This will provide the optimal device at that desired power level including the non-linear geometric effects associated with scaling.

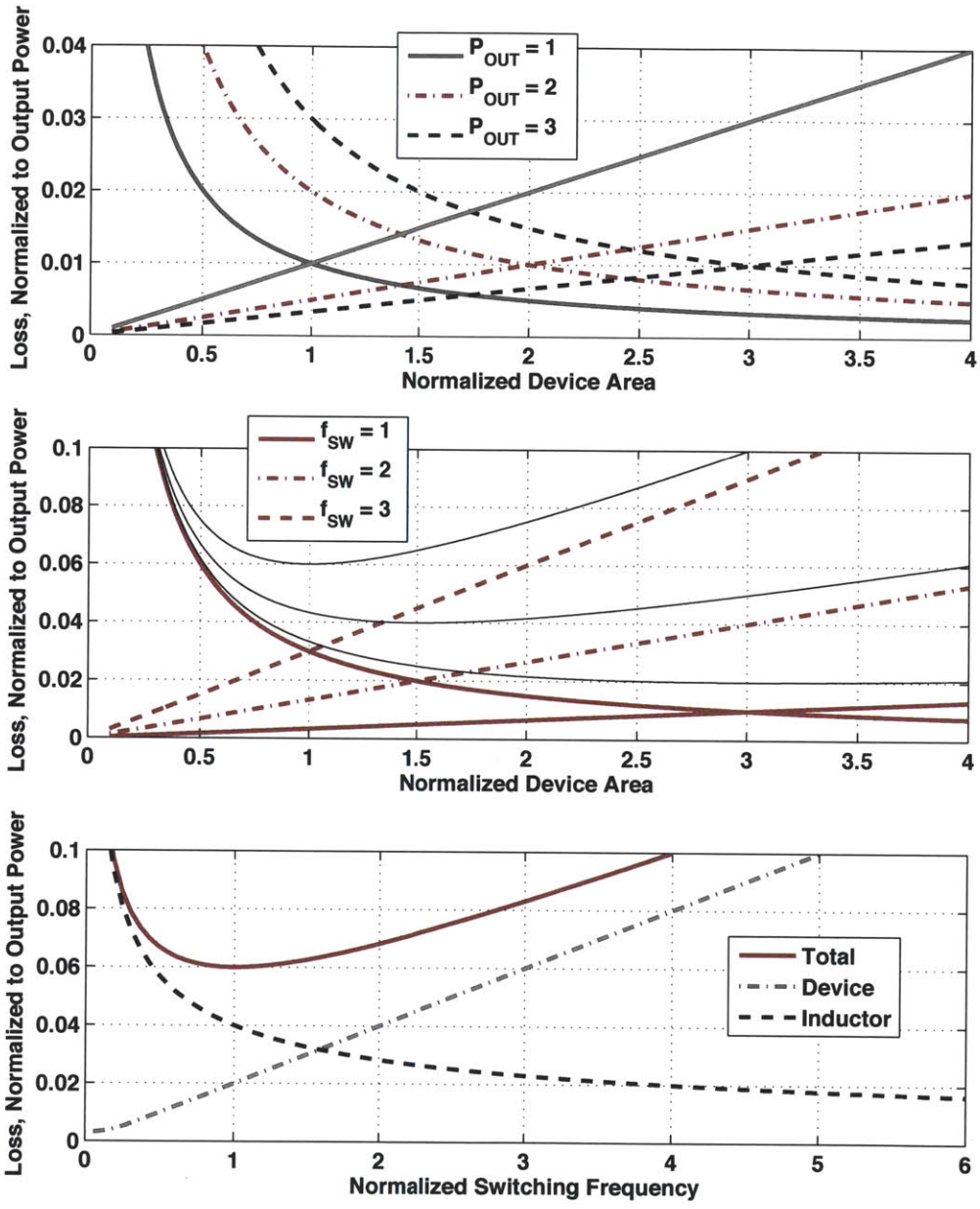


Figure 2.4: Top: Plotting conduction loss (curved lines) and frequency-dependent losses (displacement and gating loss, straight lines) vs. normalized device area reveals an optimum ratio of normalized area to output power. Middle: Frequency dependent losses scale quadratically with frequency, but the total device loss is linear when area is simultaneously adjusted for minimum loss. Bottom: When inductor loss is considered, an optimum operating frequency given semiconductor process, circuit, and operating point.

2.2 Layout Optimization

2.2.1 Overview

Power device optimization can be addressed on several levels. These include making changes to the process recipe, design rules, and layout. Among these options, layout changes typically represent the least investment in time or capital, but still offer substantial gains. In order to realize the full benefit of layout modification, edge and interconnect effects must be considered in addition to scaling as discussed in Section 2.1.3. For instance, as a device grows in size, metal resistance becomes a significant concern. Similarly for a small device, or devices comprising a very large number of small cells, capacitances along the diffusion edges, which do not scale with cell conductance, become significant. The relative importance of these parameters to the device parasitics identified in Equation 2.1 also depends on frequency, aspect ratio, and back-end process parameters such as the size and spacing requirements of inter-metal vias. These must be evaluated simultaneously to find an optimum layout for a given circuit design.

The optimization algorithm used here looks at all layout changes in concert. It is depicted in Figure 2.5. The outer loop finds the optimal device effective gate width, and the inner loop finds the best geometry given the chosen width. As a result, at each width the best geometry is determined and the width that provides the lowest total loss is the best overall geometry.

2.2.2 Layout Description

Without some means of bounding the geometry search space, layout optimization would be impractical. The first bound derives from the process design rules. Applying the constraint of using only layouts that fall within the design rules guarantees that any output produces a viable device from an electrical perspective although it foregoes the possibility of realizing the best possible designs given the process recipe. The latter would require physics-level device simulations be integrated into the optimization routine, which is prohibitive in terms of computation expense.

Even within the framework of process design rules, the number of possible layouts that could be realized is enormous. Many of these are obvious losers and easily excluded, such as widely spaced transistor cells that yield long metal runs and high capacitance. Regardless, it is still necessary to settle on a framework with a manageable number of parameters that can be easily related back to parasitic element values. Any parameter set is also constrained

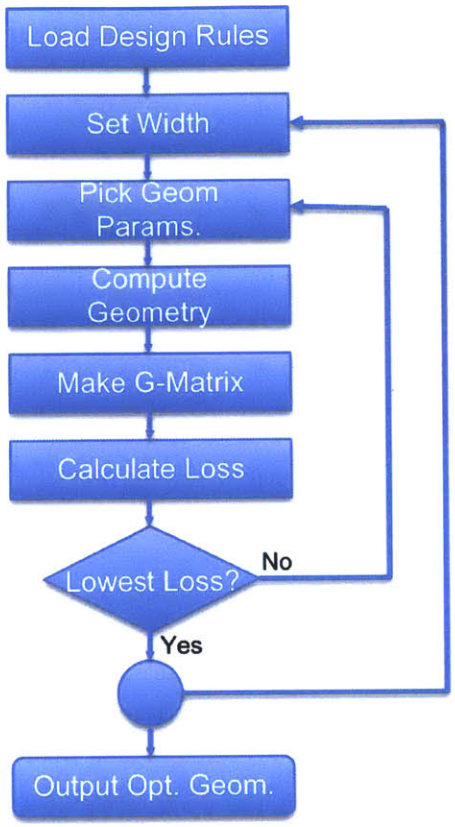


Figure 2.5: Optimization flowchart

by the requirement that it specify a unique device geometry when coupled with the design rules.

With these criteria in mind the framework settled on here is a 2-dimensional array of transistor cells with their drains, gates, and sources interconnected in parallel. The chosen geometry is depicted in Figures 2.6 and 2.7. The single cell in Figure 2.6 is an LDMOS gate finger [3]. It is a rectangular ring of polysilicon defining two channels with shared source and bulk diffusions and a drain diffusion along each vertical length of polysilicon. Since this is a lateral device, the channel exists directly under the gate and drain current flows just beneath the silicon-silicon dioxide interface, inward from the drains and out the source. The bulk and source diffusions are shorted together in the center of the polysilicon ring and contacted by three columns of tungsten plugs, while the drain diffusions are intended to be shared between adjacent cells, such that an array of cells comprising N columns will have N+1 drain diffusions. Most of the geometric parameters are fixed by the process design rules

[3]LDMOS stands for Lateral, Double-Diffused MOSFET. See [25] for a more complete description of LDMOS devices

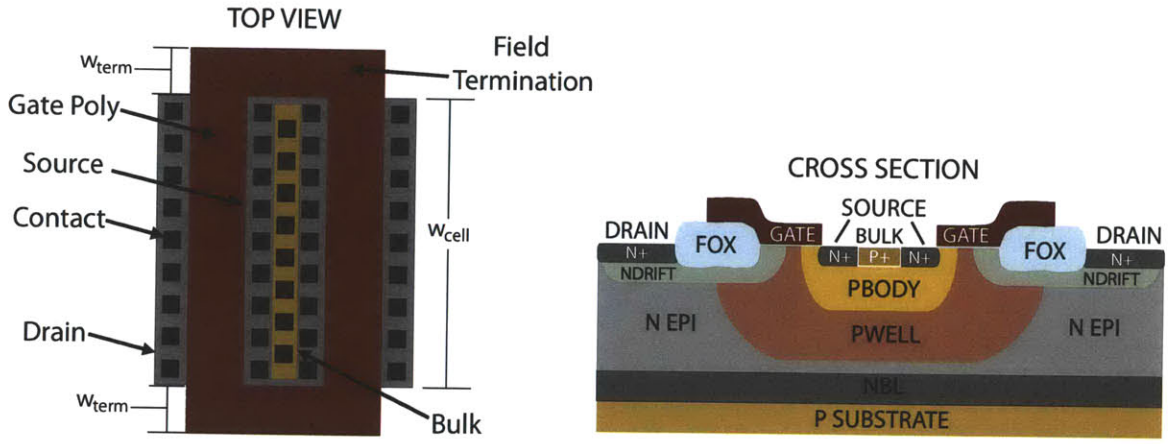


Figure 2.6: The top view of a single LDMOS cell. All the dimensions are fixed excepting w_{cell} which is scalable.

including the channel length, diffusions, and contact-contact spacing. The free variable is cell width, which sets the dimensions of the polysilicon fingers as well as the number of contacts. The polysilicon end caps act as field terminations and have a fixed width that contributes capacitance without the benefit of additional channel conductance. As a result, cells with very small channel widths will tend to have higher specific capacitances than those that are much wider.

A complete power transistor is typically assembled from a large array of cells interconnected with back-end metallization. The bottom picture in Figure 2.7 shows the cells arranged adjacent to one another along with the gate interconnect scheme and metal-1 drain-source straps running vertically in the direction of each cell. To minimize device area the field terminations of adjacent rows are abutted as are the drains of adjacent columns. This decreases specific on resistance and device capacitance, which is beneficial for both the hard- and soft-switching cases.

Figure 2.7 shows the device layout and metal interconnect. The bottom-most picture shows the cells arranged in a regular array with shared drain diffusions and abutted field terminations. Since the polysilicon forms the gate, contacts are added on top of the field termination regions to provide electrical connection for the gate terminals of each cell. Contacts are provided at both ends of each cell, so the polysilicon portion of the resistance of a single cell is reduced by a factor of four over a single-ended contact scheme. The translucent blue layer labeled, "GATE," depicts the gate interconnect scheme. Each row of gate contacts (black squares) has a horizontal stringer of metal-1 terminating on a vertical gate buss at either end of the power device (only the left-hand termination is visible). In this process, for devices with a desired effective width on the order of 50 cm or greater, the gate metal dominates R_{GATE}. Therefore, the number of stringers and width of each are optimization

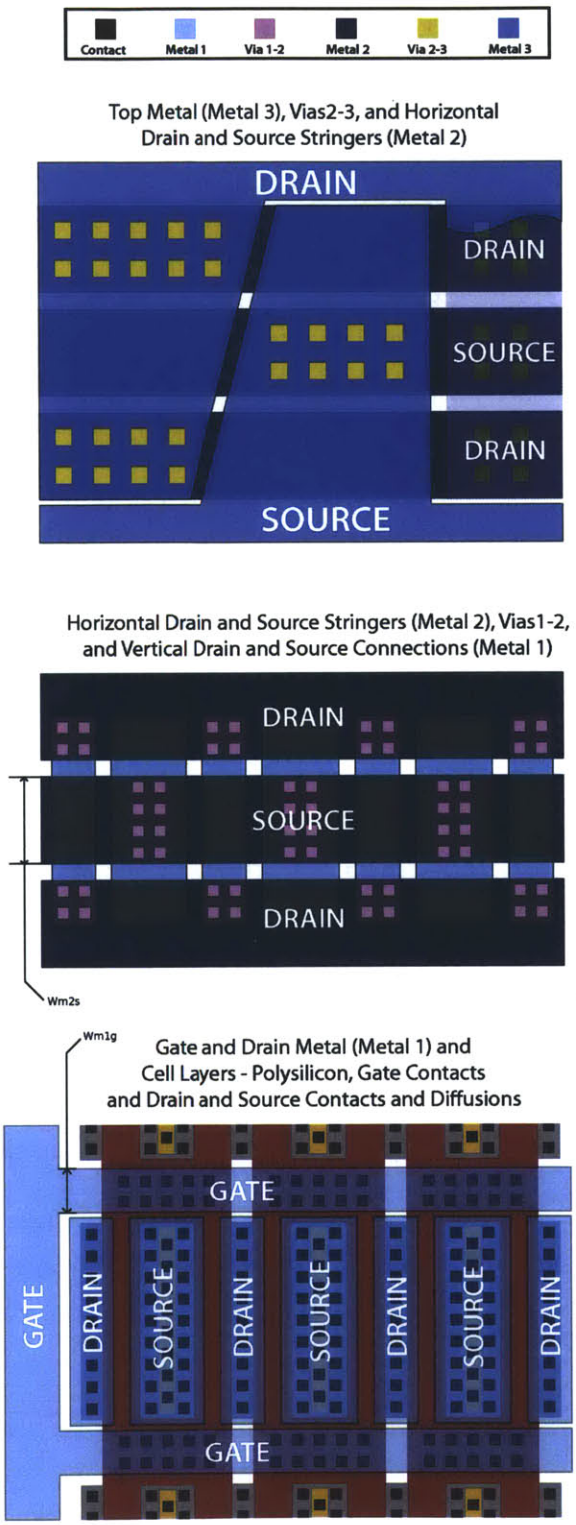


Figure 2.7: Layout overview showing the interconnect stack-up.

parameters. While the width of the stringers, W_{m1g} may be set to any reasonable value, the number of stringers is fixed by the number of rows of cells. The row count is itself determined by the total effective width, cell width, and overall device aspect ratio.

Vertical metal-1 straps labeled "DRAIN" and "SOURCE" short the drain and source contacts of each cell in the bottom figure and provide the first layer of drain and source interconnect. Moving up the interconnect stack, the middle image of Figure 2.7 shows the vertical drain and source straps located under horizontal metal-2 stringers. Each stringer forms either a drain or source buss and ties an entire row of cells (or adjacent rows in the case of the drain stringers) in parallel. The pink squares are the vias, which represent a connection from metal-1 to metal-2 where all three features are coincident. The total width of a metal-2 drain and source stringer stack is fixed by the cell width, but the ratio of drain to source metal is set by the parameter, W_{m2s} which trades the proportion of metal-2 devoted to drain versus source connectivity. At the uppermost level of interconnect, the top drawing in Figure 2.7 illustrates how the metal-2 straps interface with the final layer of metal. The metal-3 interconnect is cut similarly to a zipper down the length of the device. The resulting teeth are connected using vias to the metal-2 drain and source stringers that lie beneath. The base of each tooth is connected to a buss that forms the bondpads used to connect the device to the drain and source terminals at the package level.

2.2.3 Geometric Parameters for Optimization

The number and choice of optimization parameters are determined by balancing the degree to which each parameter can affect device performance against the cost of including it in the optimization. In this case, seven distinct parameters were chosen. Each has some significant effect on overall device performance, and while they can be varied independently of one another, they are coupled through the design rules and physical constraints necessary to build a realizable device. Table 2.3 lists them and briefly describes their significance.

Device width has the greatest impact in an optimization for VHF operation. The device must be sized at least large enough to handle the design current, but not so large that frequency-dependent losses become untenable. The scaling discussion in Section 2.1.3 informs this most basic of tradeoffs. Almost as significant to VHF performance is cell width (labeled as w_{cell} in Figure 2.6). Figure 2.8 illustrates the primary motivation behind making cell width an optimization parameter. Each cell comprises two polysilicon legs that contribute conductance and capacitance to the device and two legs that contribute only capacitance. As a result, shorter cells will have a lower ratio of capacitance to conductance.

Table 2.3: Optimization Parameters

Parameter	Importance to Device
Device Width	Sets intrinsic R_{DS-ON}, overall device size
Cell Width	Affects $R_{GATE}C_{ISS}$ and $R_{DS-ON}C_{OSS}$
Aspect Ratio	Trades drain/source and gate metal losses
w_{m1g}	Trades R_{GATE} and C_{OSS} and C_{ISS}
# metal-3 cuts	Drain-source metal resistance
angle metal-3 cuts	Drain-source metal resistance
w_{m2s}	Drain-source metal resistance
# gate bondpad arrays	Trades R_{GATE} and total device area

As illustrated in Figure 2.8, the effective width of a power device is the sum of all the widths of the conducting legs of polysilicon. Over the range of permissible cell widths, the minimum capacitance obtains when the total effective width comprises a single cell. If the device is chopped into two or four cells whose total effective width is identical, then the fraction of device area made up of non-conducting lengths of polysilicon rises and so

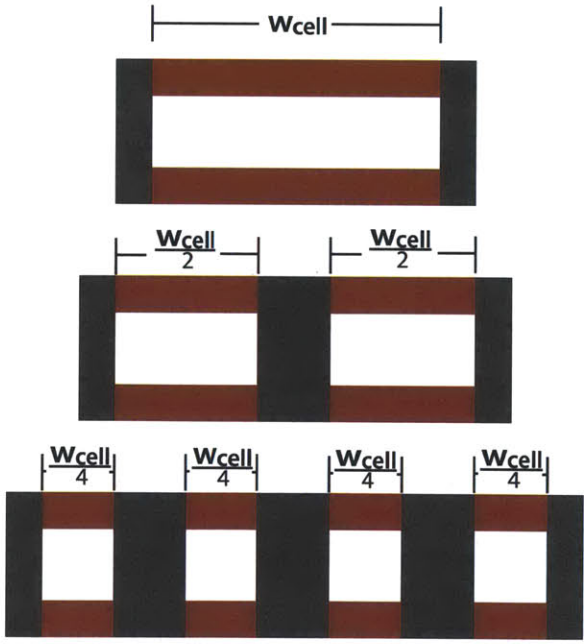


Figure 2.8: Each LDMOS cell comprises a ring of polysilicon, where two field terminations contribute capacitance but no additional conductance (depicted by the vertical gray bars in this illustration). As a result, increasing the number of cells to achieve a desired conductance will increase capacitance because the terminations take up more of the total device area. On the other hand, the gate resistance falls as the number of cells increases because of shorter poly runs and more gate contact points. This leads to an optimum w_{cell}.

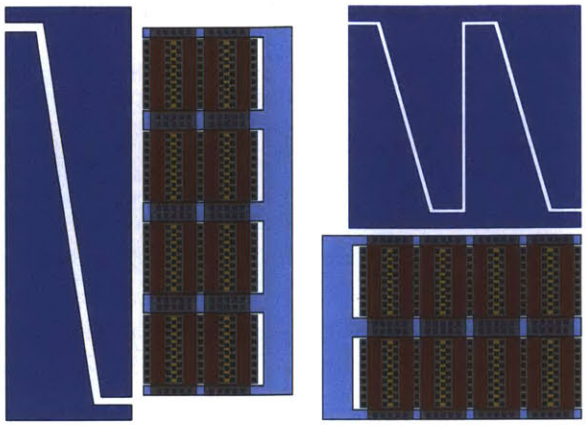


Figure 2.9: At a given device width, the aspect ratio has a strong effect on the overall performance. As the cells are stacked from a tall, narrow structure to a short, broad one, the gate resistance increases and the drain-source resistance decreases. In addition, C_{OSS} will rise because of the greater number of exposed junction ends.

does the capacitance. This affects both C_{ISS} directly through gate area and C_{OSS} via increasing the proportion that derives from end-effects where the drain diffusions spread into the surrounding silicon.

There are a few reasons that the optimal cell is not a very long one. First, as the cell is lengthened, the distance from the gate contacts to the center of the cell rises. This increases gate resistance, which has a negative effect on the gating loss. Further, as the cell length increases, the cell count decreases. This implies that the maximum possible number of cell rows and their associated gate stringers falls. The direct result is more gate metal resistance, which also deteriorates device performance at long cell lengths. Finally, as cells become very long, it is no longer possible to commutate the entire cell length simultaneously on the timescale of the converter waveforms. This introduces more loss as some portions of the device are conducting outside of strong enhancement. In practice the latter effect doesn't come into play at VHF when resonant gating is employed since the desire for low gate resistance keeps the cells relatively short.

Aspect ratio changes are illustrated in Figure 2.9. This primary determination is whether the device is tall and skinny or short and wide. The shape of the device has the most pronounced effect on the resistance components that arise from the metal interconnect. In a low aspect ratio device (the 4 x 2 array in the figure) there will tend to be many, short gate stringers that contribute to lowering the gate resistance. The opposite is true of the drain-source metal, where there are few, long metal-3 teeth. This raises the resistance of the drain-source interconnect directly impacting R_{DS-on}. Another more subtle effect of favoring a large aspect ratio over a small one is an increase in total device capacitance owing

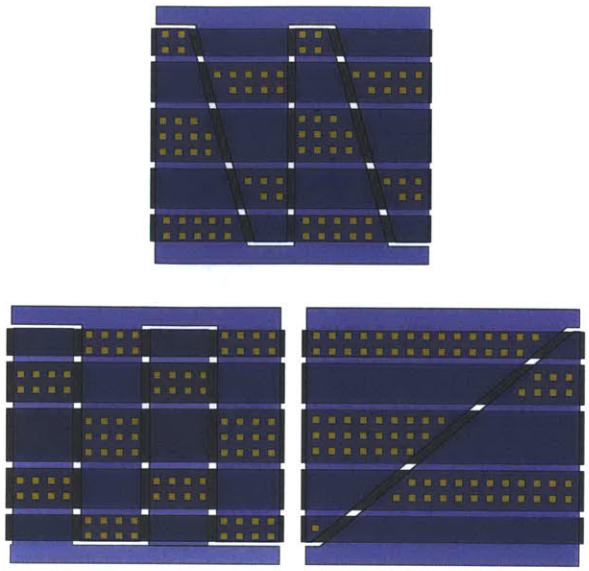


Figure 2.10: The top layer of interconnect, metal-3 in this case, is chopped into a zipper-like structure. Each tooth experiences rising current from tip to base as successive rows are aggregated. In order to minimize the loss, the teeth are tapered to keep the current density more constant. The final taper angle is influenced by metal-2, via density, and relationship between the taper angle and final tooth count.

to a larger number of exposed drain diffusion ends. For this process and the devices under consideration, this effect was strongly outweighed by the metal resistance considerations.

The other metal-1 and metal-2 parameters, w_{m1g} and w_{m2s} are used to minimize the loss in their respective layers and are the least coupled parameters. w_{m2s} has essentially no effect other than reducing the metal-2 loss, while w_{m1g} does influence the specific capacitance because spacing rules require that the field terminations are widened as it increases. Optimization of the metal-3 layout involves two parameters simultaneously the tooth taper and pitch. Figure 2.10 depicts a few possible metal-3 layouts. Tapering the metal-3 teeth reduces the peak current density along each tooth, which would otherwise rise toward the tooth-base as successively more metal-2 stingers are intersected. This keeps the total loss in metal-3 to a minimum. The final taper angle is chosen to minimize the total loss which involves current density in both metals-2 and -3, via density, and maximum tooth-tooth distance. The tooth pitch is important because it affects the total available metal-3 area to conduct current–a decrease in tooth pitch increases the percentage of the total metal-3 area that is used by the cuts between adjacent features. This in turn balances with the reason for reducing tooth pitch in the first place, the corresponding reduction in peak current density in metal-2 that leads to lower loss.

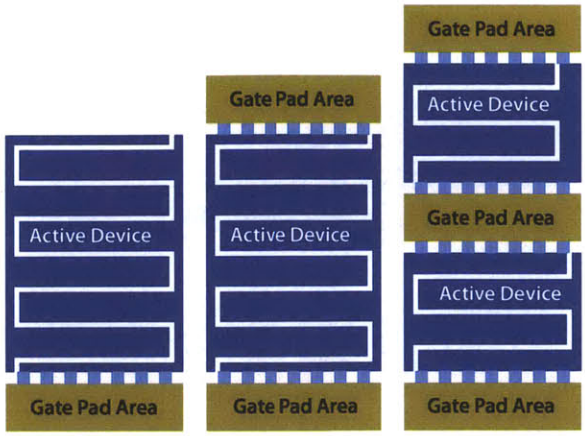


Figure 2.11: By breaking the device into a number of smaller devices separated by gate pad arrays, the total gate resistance can be dropped dramatically without strongly impacting R_{DS-on} or device capacitance.

The final parameter that has significant bearing on device performance is the number of gate bondpad arrays. Figure 2.11 shows device configurations including various numbers of pad arrays. The single array case has the largest gate resistance and the smallest area, all things equal. Placing gate-pad arrays at either end of the device reduces the metal component of the gate resistance dramatically. A further reduction in gate resistance can be achieved by effectively splitting the power device into sub-segments located between gate pad arrays. This proves a particularly effective tradeoff because VHF optimization tends to push devices to fairly high aspect ratios. By inserting an additional array of gate pads mid-device, the resistance can be dropped significantly without a major penalty to total device area. Successively chopping the device into smaller segments suffers a steep decline in effectiveness because the reduction in gate stringer length decreases.

Intercalating gate pad arrays between device segments is a winning trade over building two completely separate devices both from the perspective of total silicon area and avoiding the problems of parasitic inductance between two packages. However, a gate pad array mid-device does pose some challenge to a successful bondout. Typically, bondwires and bondpads need to be arranged symmetrically around a common center to maximize the wire-wire clearance and minimize shorts during the encapsulation process. The package used for this work is a 28-pin TSSOP with a central heat spreader. To minimize inductance and resistance associated with the bond pads, the devices were bonded with 14 wires on each of the drain and source connections. These were connected to individual bondpads on the die. In the case of the drain, the bondwires were connected to the top half of the lead frame. For the source, inductance and resistance were minimized by down-bonding directly to the heat spreader. This leaves space for the gate wires to run from the ends and center of the die to the bottom half of the lead frame.

2.2.4 Layout to Device Parasitic Parameters

The interrelationships between geometric parameters and device performance drive the need for a simultaneous optimization of each parameter with minimizing device loss as the goal function. This requires the ability to to extract the parameters of the loss model given a geometry. Once the model values are known, calculating the device loss for a given converter design is a straightforward task.

Device capacitances and intrinsic resistance components can be extracted from layout geometry in a fairly straight-forward fashion. The capacitance data is derived from basic process characterization information which is typically available from the process owner. Within the framework of the design rules, capacitance can be determined by accounting. That is, each cell has capacitance components that scale with the cell width and perimeter as well as a fixed component that is due to the field termination regions. The perimeter and end effects are dependent on where the cell is located in the device matrix. In general, the total capacitance contributed by a cell at a device edge is larger than that contributed by a cell that is surrounded by neighboring cells on each side. For a given geometry this information is known, so determining the total capacitance is just a matter of summing up the contributions of each type of cell. The addition of an areal component that depends on the total size of the N-buried layer under the device, and the interconnect capacitance completes the sum. This is a typical means of accounting for device capacitance in a well-characterized process (see, for instance, [26]) and is easily extended to the intrinsic resistances such as the polysilicon, access, and drift resistances.

The interconnect-related portions of the resistances come from the metal layout, bondwires, and leadframe. Among these, determining the resistance due to the metal layout requires the most effort. The various portions of the interconnect resistance are calculated by approximating the metal layout as a network of resistors where each is determined by the sheet resistance of its respective layer. This avoids the computational overhead of a field solution while providing results accurate enough for optimization work. The resulting resistor networks are used to populate a conductance matrix which is then solved for the equivalent resistance of the network driven by the current waveforms from the target converter design. Many layout details can be distilled into a few resistance parameters and iteratively populated in a matrix using this method.

Several networks were used to extract the effective resistances of the interconnect that affect the loss model parameters. By way of example, the gate interconnect network is shown in Figure 2.12. Each cell is modeled as a set of resistances due to the polysilicon, the gate contacts and an access resistance along with the gate capacitance. The metal stringers are chopped into small segments according to the cell pitch allowing R_{m1g} to be calculated as

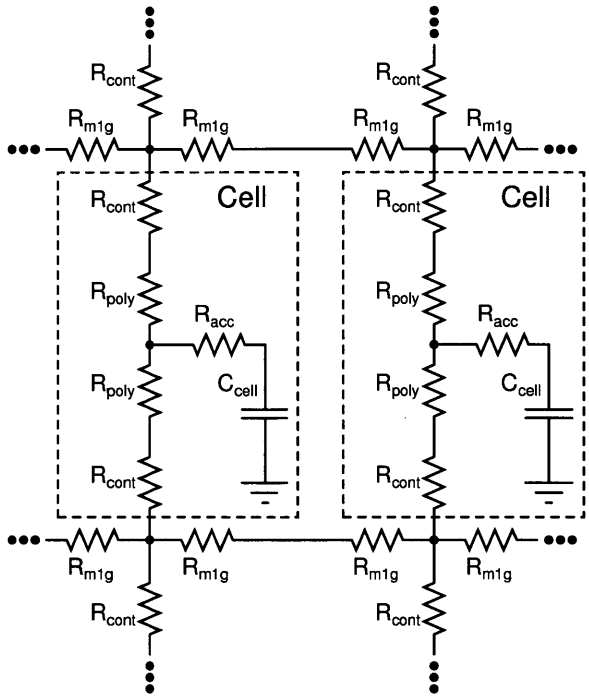


Figure 2.12: This network is used to calculate an equivalent gate resistance. The values are populated by relating the geometry and material constants in the device. Several similar networks are used to calculate interconnect resistance for the drain and source networks, as well.

$\rho_{sheet} * cellpitch/w_{m1g}$. R_{cont} is determined by dividing the contact resistance by the total number of contacts per cell, and R_{poly} is determined based on the poly sheet resistance and the cell width. Once all the values are populated, a conductance matrix is created allowing the calculation of an equivalent resistance for the network. This is similar to the method employed in [27]. The result is summed with the relevant intrinsic device resistances to extract a value for R_{gate}. The same procedure is used for the drain-source metal, with the addition that the metal-3 tooth taper and width is accommodated by using a 2-dimensional array of resistors to account for the change in current flow. The grid density was chosen as a tradeoff between computation time and accuracy. It proves sufficient to use a grid with a density of one node per cell for this work.

2.2.5 Optimization

The optimization procedure was implemented using MATLAB (scripts are available in Appendix A.1. The optimization algorithm ultimately chosen is a brute-force search of the parameter space, effectively an exhaustive search using combinatorial optimization. Gradi-

ent descent and stochastic gradient descent algorithms were also explored, but ultimately proved no faster at achieving convergence. Instead, the choice of a small set of optimization parameters, exploitation of symmetry, and the suitability of the problem for sparse matrix operations reduced the computation time such that an exhaustive search could be performed in a few hours using a typical 3 GHz dual-core workstation.

The goal of the optimization is to minimize total device loss, the magnitude of Eq. 2.1. This is accomplished by permuting the seven optimization parameters in Table 2.3. For each set of parameter values a unique device geometry is created by using the geometric framework outlined in Section 2.2.2 and the process design rules. Once a given geometry is created, the various parasitic elements are extracted according to Section 2.2.4. It is then straight forward to compute the loss of the specified geometry by applying the converter parameters derived from SPICE simulation to the device model.

This procedure is performed over the entire set of geometry parameters for a given device width. Device width is initially seeded by choosing a value that sets conduction loss due to the intrinsic R_{DS-on} to approximately 5% of the desired output power. Successive device widths are chosen by comparing the total loss of the new trial to the previous one and then implementing a hill-climbing algorithm to find the best device size. The lowest loss device at the optimum device size is the optimal device for VHF performance in the exhaustive search set.

To make an exhaustive search practical symmetry was exploited as much as possible. A straight permutation of the geometry parameters would result in many complementary geometries, which are identical from a device performance perspective. These cases were pre-filtered and cut the number of permutations necessary by a factor of two. Further, the devices consist of a number of identical segments that can be cut along the metal-3 tooth edges. Since no appreciable currents flow across the boundaries, the device can be treated as many paralleled devices of reduced effective width. This results in much smaller conductance matrices for the drain and source. Since the time to back solve such a matrix depends strongly on its size, the total optimization time was reduced by an order of magnitude by adopting this approach.

Figure 2.13 plots the results of one optimization run at an effective width of 73.5 cm versus aspect ratio and cell width. At the desired operating frequency of 50 MHz, the minimum loss is about 380 mW. This occurs when the cell length is 70 um (35 um per finger), and the device aspect ratio is near 2. Figures 2.14 and 2.15 are the drain-source resistance and gate resistance respectively plotted versus the same variables. R_{DS-on} shows a general trend of decreasing both with reduced cell width and higher aspect ratio. The dependence on cell-width derives primarily from the increased number of drain and source metal-2

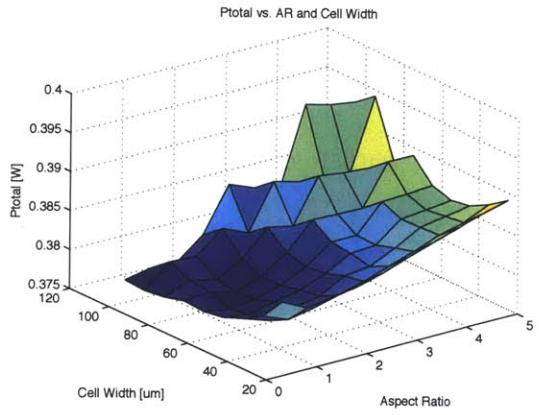


Figure 2.13: Device power dissipation vs. cell width and aspect ratio for a 73.5 um device in the case-study converter example.

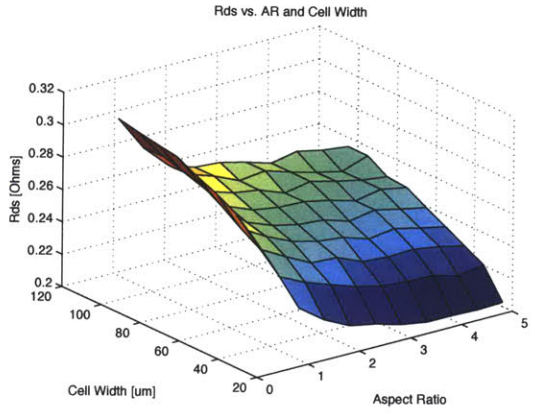


Figure 2.14: R_{DS-on} vs. cell width and aspect ratio

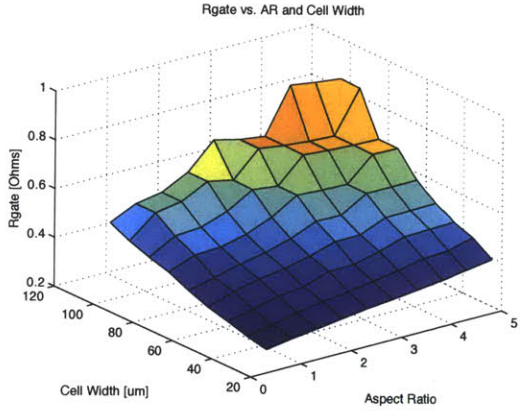


Figure 2.15: R_{gate} vs. cell width and aspect ratio

stringers as w_{cell} is lowered. With more stringers, the increased frequency of vias to the three-times thicker metal-3 results in a lower resistance path to the bondpads. This trend continues over the range plotted, but is eventually overcome by the exchange of metal-2 for the inter-trace spaces necessary with each successive stringer. Aspect ratio influences R_{DS-on} largely by reducing the length of metal-3 runs as it increases. The dependence of R_{gate} on cell width and aspect ratio comes directly from the number and length of the metal-1 gate stringers. Small aspect ratios result in many short gate stringers, and smaller cell widths require more gate stringers at a given aspect ratio. This yields the trend in Figure 2.15 of lowest resistance at the extremes of square tall narrow devices and small cells. The plots of R_{DS-on} and R_{gate} show complementary behavior over aspect ratio and cell width, and this is reflected in the minimum that shows up in P_{total}.

Table 2.4: Measured Device Parameters

Device	R_{DS-ON}	R_{OSS}	R_{GATE}	C_{ISS}	C_{OSS}
MRF6S	175 mΩ	170 mΩ	135 mΩ	50 pF	110 pF
F	200 mΩ	400 mΩ	1300 mΩ	274 pF	132 pF
HV1	181 mΩ	145 mΩ	370 mΩ	266 pF	126 pF
MV1	113 mΩ	174 mΩ	300 mΩ	136 pF	97 pF
HV2	172 mΩ	165 mΩ	201 mΩ	268 pF	127 pF
MV2	112 mΩ	154 mΩ	133 mΩ	151 pF	108 pF

2.2.6 Layout Optimization Results

To test the optimization procedure and glean better devices for converter development, six devices were fabricated in the same integrated power process. The power process is a Bipolar-CMOS-DMOS (BCD) process with a 700 um minimum feature size. In addition to providing an array of standard 5V CMOS transistors and bipolar transistors, there are two LDMOS devices, one supporting 50-V operation and the other 20-V operation. The first device was a sample transistor provided by the fabrication facility. It was originally used for process characterization purposes. The VHF characteristics, in particular the gate resistance which was over 7 Ω, of the characterization device made it impossible to test at the intended operating frequency. The subsequent device, designated the "F" device, was hand-optimized for RF operation. The gate resistance was substantially reduced and the finger lengths and drain source metal were similarly modified to improve performance.

Although the F device proved capable of operating at VHF frequencies [28], performance was poor when compared to a commercially available LDMOS, the MRF6S. This motivated the optimization effort and the four additional devices designated HV1, HV2, MV1 and MV2. The HVx devices have a 50-V rating identical the F device, whereas the MVx devices share a 20-V rating and were made specifically to test the hypothesis that hot carrier effects should be minimal under soft-switching (see Section 2.3).

The performance improvements provided by layout optimization alone were compared against two baselines. The first is the ability of the integrated power devices to approach the performance of a discrete LDMOS process such as that the MRF6S, which was picked as the best-available commercial device at the time of the investigation. The second is to compare the merit of simultaneous optimization of all the device parameters by computer versus a hand optimization where it was only possible to consider a subset of the parameters individually.

Table 2.5: Calculated Loss Comparison

Device	Conduction	Displacement	Gating	Total
MRF6S	189 mW	18.9 mW	5.2 mW	213 mW
F	216 mW	310 mW	308 mW	835 mW
HV1	196 mW	102 mW	82.7 mW	381 mW
MV1	122 mW	72.9 mW	17.5 mW	213 mW
HV2	186 mW	118 mW	45.6 mW	350 mW
MV2	121 mW	79.9 mW	9.6 mW	211 mW

Tables 2.4 and 2.5 summarize the results concisely. Table 2.4 shows the device parameters as measured on an Agilent 4395a impedance analyzer. Each device was subjected to a series of one-port measurements where the complex impedance versus drain-source bias was extracted. This provides the requisite capacitances and resistances for the loss model, when the drain-source bias voltage is taken at the DC-average operating value (in this case 14V). For the R_{DS-on} measurements, V_{GS} was set to 8V, which is the intended gate drive voltage during converter operation. The individual loss values were tabulated using Equation 2.1 and the case-study converter parameters from Section 2.1. The results show that from layout optimization alone on the 50-V devices, a 54% reduction in device loss was achieved over the F device, and a 58% reduction with the addition of copper top metal. This is still about 65% more loss over the commercial MRF6S, but a substantial improvement nonetheless, where no change to the process or design rules was necessary.

Figures 2.16- 2.18 are drawings of the F, HV1, and HV2 devices showing the overall layout schemes. Each of these devices has an effective gate width close to 7.2 cm, which was the original design width of the F device. Identical widths allow for direct comparison among the devices. Device optimization was performed on HV1 and HV2 as described above. Table 2.4 shows that the optimization had the greatest effect on R_{GATE}, which drops from 1.3 Ω in the F-device to approximately 200 mΩ in the HV2 device. This is a direct consequence of changes to gate layout driven by the optimizer. The F-device has 13, 1800 μm x 2.7 μm gate metal stringers connected to a gate pad array at one end of the device. In contrast, the HV1 device has 3 gate pad arrays. One pad array is located at each end of the device and the third splits it into two halves. The nine gate strips in HV1 are nearly twice as wide and less than half as long at 800 μm x 5.7 μm. HV2 has a similar gate metal layout, but the top drain-source metal is copper allowing for a lower aspect ratio device (F and HV1 are about 500 μm x 2 mm, where as HV2 is about 1.1 mm x 1.3 mm). This doubles the number of gate stringers dropping the total gate resistance to 201 mΩ.

The HV1 and HV2 devices also have 35 μm cells in contrast with the F device's 25 μm cells. This reduces the intrinsic input and output capacitance associated with the cells. The

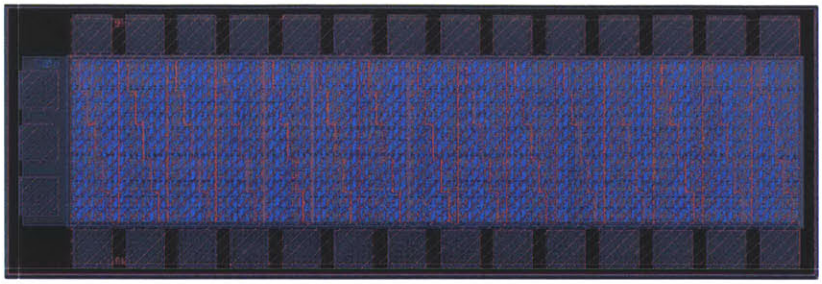


Figure 2.16: F device layout

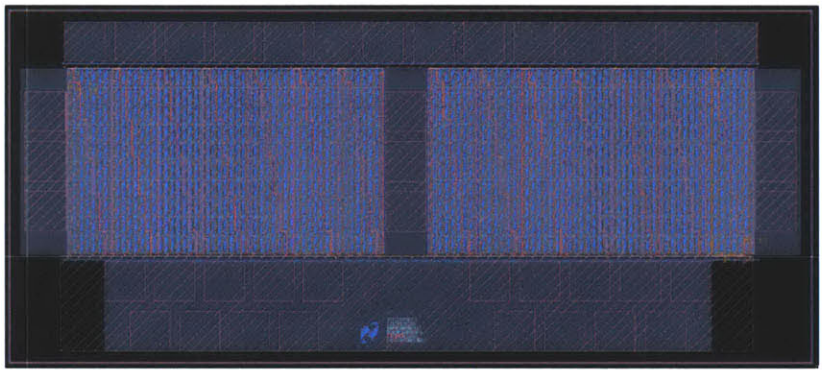


Figure 2.17: HV1 device layout with multiple-gate pad arrays and longer fingers

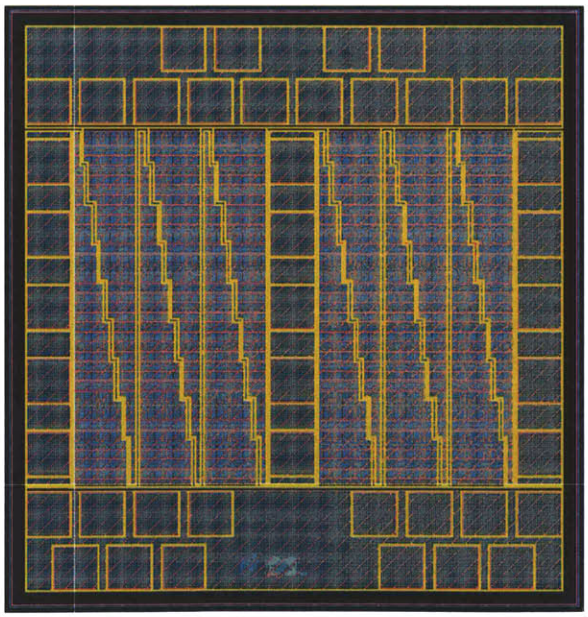


Figure 2.18: HV2 device layout with a square aspect possible for the introduction of copper top metal

total reduction in C_{OSS} and C_{ISS} is only on the order of 5%, being partially compensated for by larger interconnect capacitances owing to the larger devices and layout differences. However, the exchange of capacitance from intrinsic device capacitance to back-end metal capacitance helps to reduce R_{OSS} because the latter is much less lossy. Increasing the cell width also allows for wider metal-2 conductors (the largest source of resistance in the drain-source metal for these high aspect ratio devices) in the drain source path. In conjunction with a somewhat shallower metal-3 angle in the HV1 device, a modest reduction in overall interconnect resistance was achieved, contributing to a lower R_{DS-ON}. In the HV2 device, metal-2 and metal-3 are paralleled to further reduce the contribution from the drain and source stringers, and copper is used in place of metal-3 for the topmost layer.

2.3 Safe Operating Area Considerations

Soft-switched converters are able to achieve high efficiency at VHF by avoiding voltage and current overlap in the switching device. The resulting switching trajectory closely follows the voltage and current axes for both turn-on and turn-off transitions. Figure 2.19 shows the simulated switching trajectories for a Class-Φ_2 boost converter and an ideal hard-switched boost converter. In the Class-Φ_2 converter the switch never has simultaneously high voltage and current, while under hard-switching the device experiences both high voltage and current simultaneously. The very different switch stress patterns that result have significant implications for the portions of the switch safe operating area that can be reached during operation.

Hot carrier effects result from the accumulation of damage in a device caused by high energy carriers [25, 29–31]. For LDMOS devices, hot carrier effects manifest as shifts in threshold voltage, V_{TH}, or R_{DS-ON}. Threshold shifts are generally the result of hot carriers becoming embedded in the gate oxide. R_{DS-ON} shifts arise as hot carriers create interface traps in any of the lightly-doped drain region, the accumulation region under the gate, and/or the bird's beak region, located at the tip of the FOX-gate interface area.

Under normal operation, a small number of carriers will attain the energy necessary to cause damage. Over time, the damage accumulates and eventually the shift in V_{TH} or R_{DS-ON} becomes severe enough that the device is no longer useful. As the local electric fields increase, a larger fraction of the carrier population has sufficient energy and damage accumulates more rapidly. The simultaneous condition of high current and high fields is particularly bad, and ultimately requires a restriction on the safe operating area (SOA) to prevent operation in regions that will dramatically shorten the service life of the device. For LDMOS power devices hot carrier reliability, SOA, and R_{DS-ON} are tradeoffs [25, 29]

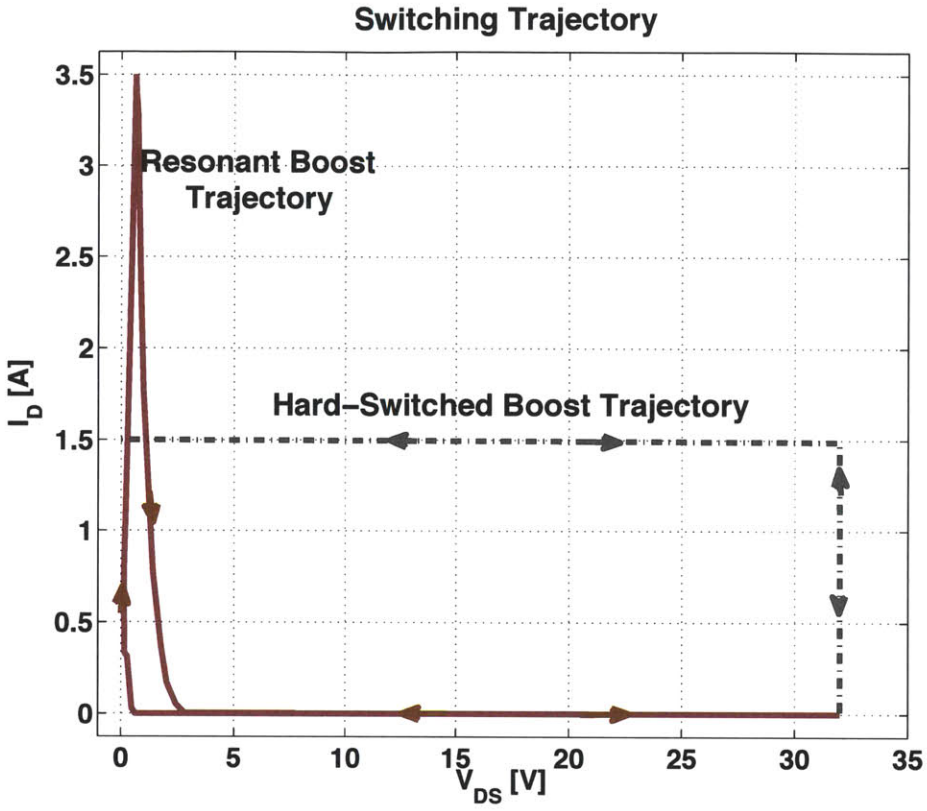


Figure 2.19: Switching trajectory for Class-Φ_2 Boost converter and an ideal hard-switched boost for the same voltage and power level.

controlled primarily via the drain drift region. To reach a desired safe operating voltage, while ensuring reliability, the device must have certain minimum dimensions and a carefully controlled doping profile that minimizes hot-carrier effects, but generally drives the device to have a breakdown voltage higher than the working voltage. The consideration of hot-carrier reliability thus imposes a tax on device design in the form of higher parasitic capacitance for a given R_{DS-ON}

In soft-switched VHF converters, device voltage and current are never simultaneously high. Without the conditions to create large numbers of hot carriers, device degradation does not occur, and we are free to extend the peak drain-source voltage towards the much higher avalanche limit. This extension of the SOA was validated through a set of experiments discussed discussed below. The result is significant in terms of VHF device performance. Without the need to constrain operating voltage in light of hot carriers, devices with a shorter drift region can be used. These devices will have substantially lower capacitance at a given R_{DS-ON}. Since frequency dependent loss in VHF resonant converters is square-law

dependent on capacitance, the efficiency improvements are significant, as can be seen in Table 2.5 by comparing the MV1 and MV2 devices against the others.

2.3.1 SOA results

As mentioned in Section 2.2.6 layout changes alone reduced device loss in this process by 54%. Further reduction is realized by substituting the 20-V MV1 and MV2 devices in place of the 50-V devices that were being used in the experimental converters. This requires that the 20-V devices operate to a peak working voltage of at least 35 V, substantially beyond their rating for hard switching. This required SOA extension turns out to be possible given the general hot-carrier discussion outlined above.

The MV1 and MV2 devices take advantage of 20-V design rules that allow for a shorter drift region and lower specific on resistance to provide higher performance. When these devices are compared to the discrete MRF6S9060, in the example 50-MHz Φ_2 converter, they achieve very similar total loss. This means that in the intended application at 50-MHz the integrated process can achieve parity with a discrete device picked from among the best available.

While improved performance is expected from a device with a lower voltage rating, the point of interest is that it can be used in this application at all. In the experimental converters constructed to test these devices, the peak drain voltage attained during operation is 35 V, a 75% increase over the rated voltage of the MV1 and MV2 devices. As discussed in the introduction to Section 2.3, the mechanism that underscores this ability is a switching trajectory that never has simultaneous high voltage and current. This minimizes hot carrier effects, allowing the MV1 and MV2 devices to be used at peak voltages closer to their avalanche voltage which is around 40 V.

To assess hot carrier reliability in this process under soft-switching we used standard hot carrier reliability criteria used in industry. These require the device to run for 1 year at 10% duty ratio, or 876 hours per year. To meet standards R_{DS-ON} must shift by 10%, or less, and V_{TH} by 100 mV, or less for this operating period. In order to evaluate our devices, we ran the MV1 device in a Class-Φ_2 resonant boost converter (see Figure 2.22 and Table 2.6) at successively higher voltages for 100 hour periods. The test started with a peak v_{DS} of 15 V. Once 35 V was reached, the converter was allowed to run for an additional 1000 hours. In terms of the hot-carrier reliability test this exceeds the required run time by over 500 hours. Hot carrier damage occurs primarily at switching transitions. Since the test converter ran at 50 MHz, the total number of transitions substantially exceeds what

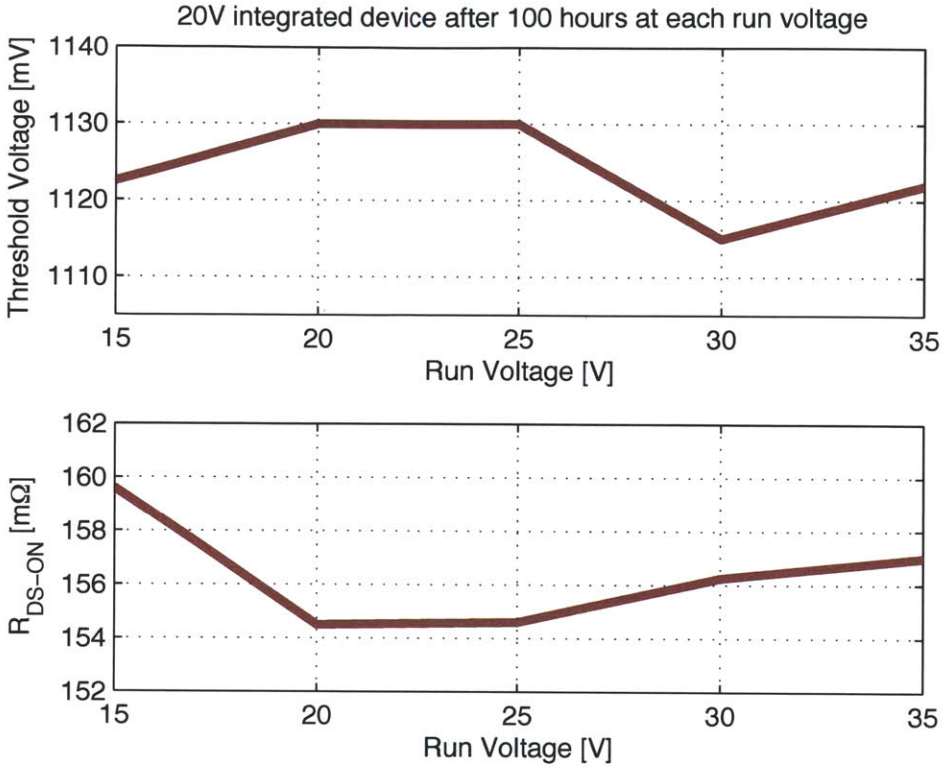


Figure 2.20: The shifts in V_{TH} and R_{DS-ON} are well within the established testing criteria.

would be expected of a hard-switched converter and hot carrier damage would show up at nearly 50 times the rate for hard-switching measurements if the conditions were excited.

Testing began by measuring V_{TH} and R_{DS} of a new device, in this case an MV1 device. Threshold voltage was determined by holding V_{DS} at 100 mV and measuring the V_{GS} that results in a current density of 0.1 μA/μm. R_{DS-ON} was measured with V_{GS} = 5 V and V_{DS} = 100 mV. Over the course of testing, the converter was periodically stopped and the device measured. The plots of Figures 2.20 and 2.21 show the accumulated results. Both the threshold voltage and on-state resistance lie well within the requirements. The total threshold shift was approximately 20 mV after 1000 hours of running with a peak v_{DS} of 35 V, and the shift in R_{DS-ON} was on the order of 4%. At 712 hours the converter input voltage was doubled, stressing the devices and producing the steep rise in R_{DS-ON} demarcated by the black line in figure 2.21. Even with this additional stress, the total shift is well within the evaluation criteria.

As a control, a hard-switched boost converter was designed around an MV1 device to operate at the same voltage and power level. The converter was then connected to an electronic load so that RMS current through the switch could be maintained near 1.75 A, identical to the Φ_2

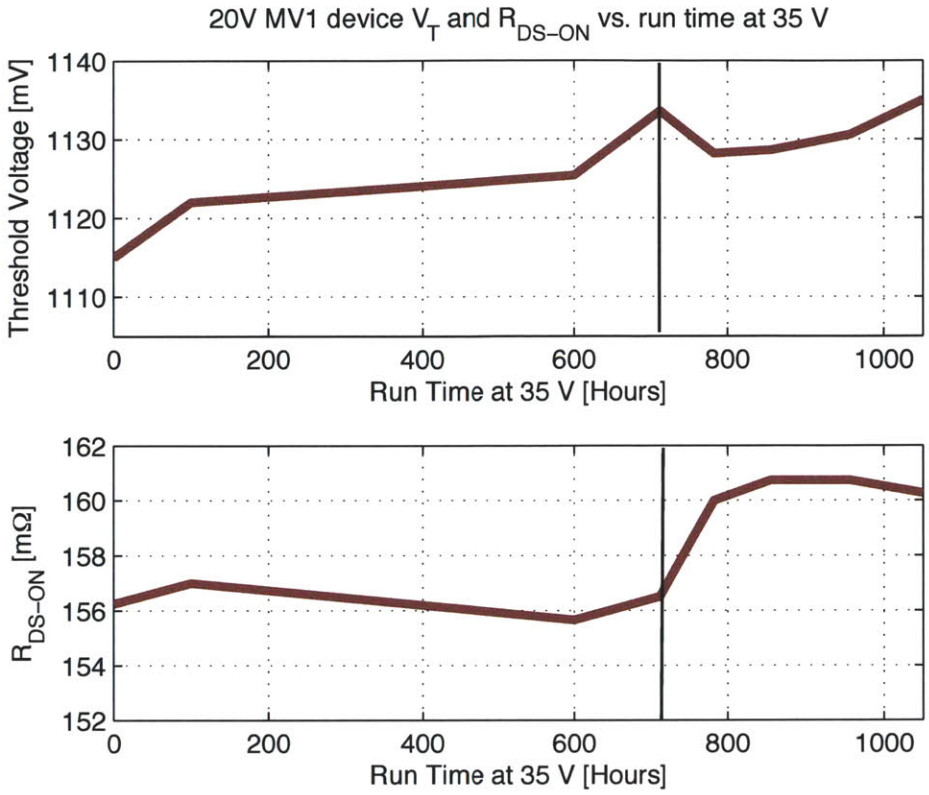


Figure 2.21: After 1000 hours of operation at 35V, the 20V MV1 device has a total V_{TH} shift of around 20mV, and about a 4% change in R_{DS-ON}. The allowable maximums are 100mV and 10%, respectively.

resonant boost converter when the peak v_{DS} is 35 V. After an initial run of 100 hours with a peak drain source voltage of 20 V, little change in V_{TH} or R_{DS-ON} was observed. After the initial run, the converter was then operated for 5 minute intervals at successively higher peak v_{DS}. This short interval was picked because shifts were expected to appear rapidly as the device voltage increased outside of the SOA. At a peak v_{DS} of 30 V, no changes were evident. Upon increasing the peak drain-source voltage to 35 V, the same voltage at which another MV1 device operated for over 1000 hours under soft-switching, the hard-switched device failed in 18 seconds. During the course of these measurements, the device temperature was monitored to ensure that the failure was not caused by simple overheating of the junctions. The temperature stayed well below that of the devices operating at VHF (less than 60 C vs the 75 C observed at VHF).

The soft-switching trajectory that permits SOA extension may not exist in the Φ_2 converter (or other VHF resonant converters) if the converter is not operating in steady state. For example, a typical method of controlling VHF soft-switching converters is full on-off modulation. [3]. During the start-up and shut-down transients, the switching trajectory will not

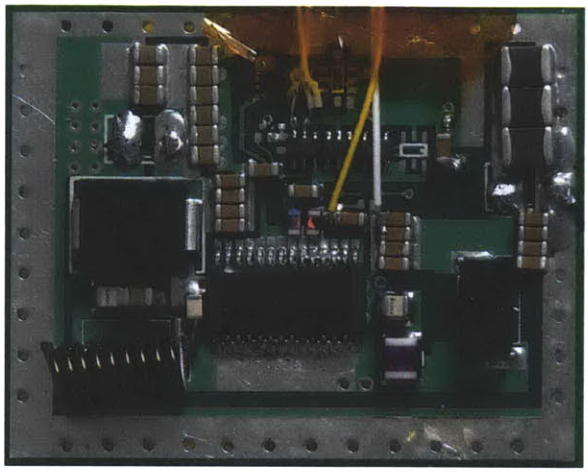


Figure 2.22: A Class-Φ_2 boost converter built using the MV1 device and operated to 35V. It achieves 88% conversion efficiency at 17W, V_{IN}=12V, V_{OUT}=33V.

always closely follow the voltage and current axes. During these periods, it is necessary that the trajectory does not leave the SOA defined for hard-switched converters, or significant hot carrier damage could occur. To assess the feasibility of operating an SOA-extended 20-V switch under these conditions, a Φ_2 converter was configured for modulation. Under modulation, the entire power stage is turned on and off at a frequency far below the switching frequency. In this case, a 50-kHz signal was used to modulate a 50-MHz converter. After running the converter with a peak v_{DS} of 35 V for 120 hours, there was no measurable shift in either V_{TH} or R_{DS-ON}.

The benefits of extending SOA are clearly delineated in Tables 2.4 and 2.5. The MV devices enjoy a 76% reduction in loss over the hand-optimized F device. The primary benefit beyond the layout optimization comes from the lower specific R_{DS-ON}. This results in substantially lower capacitance and devices with an active area roughly 20% smaller than the HV versions. The smaller dimensions also reduce the total interconnect length and the MV2 device, which has copper top metal and a small aspect ratio posts the lowest R_{GATE}, 133 mΩ. While the larger capacitances of the integrated devices over the discrete example (MRF6S9060) means that they won't scale as well in frequency, 50 MHz is sufficiently high to make converters with dramatically reduced energy storage.

2.3.2 Converters

To illustrate the gains from device optimization and SOA extension, two 50-MHz Class-Φ_2 resonant boost converters were constructed. The details are found in Table 2.6. One converter uses an un-optimized F device. The other uses the MV1 device, which is lay-

Table 2.6: Experimental DC-DC Converter Specifications

Parameter	**w/F LDMOSFET**	**w/ MV1 LDMOSFET**
Device	50V rules	20V rules, 35V validation
Efficiency, V_{IN} = 14V	75%	88%
V_{IN} Range	8-18V	8-16V
V_{OUT}	33 V	33 V
P_{OUT}	17 W	17 W
D_1	Fairchild S310	Fairchild S310
L_F	22 nH	43 nH
L_{REC}	56 nH	90 nH
L_{2F}	22 nH	22 nH
C_{REC}	47 pF	24 pF
C_{EXT}	56 pF	47 pF
C_{2F}	115 pF	115 pF

out optimized and operated with an extended SOA to a peak drain voltage of 35 V. The converter using the F device achieves 75% conversion efficiency, and the converter with the MV1 device a substantially higher 88%. A photograph of the converter with the MV1 device appears in Figure 2.22.

Chapter 3

Transformer Synthesis

3.1 Motivation

IN Chapter 2 we demonstrated the feasibility of using a standard BCD power process in the VHF regime. That integrated power transistors are a viable option means control, hotel, and auxiliary functions can be put on the same die, allowing VHF converters to benefit from the powerful shrink-down that modern silicon processing has to offer. What remains are the passive components. Passive components dominate the size, weight, and cost of power converters, so any attempt at realizing the full benefits of going to VHF switching frequencies includes addressing this fact. Fortunately, very high switching frequencies already help in this regard by greatly reducing the need to store energy in the first place. Switching cycles repeat frequently enough that only tiny chunks of energy need buffering, and for the converters demonstrated her, that translates into inductance measured in nanohenries and capacitance in picofarads. Even so, there is much to be gained from a closer look at passive component synthesis.

One of the crucial elements harnessed during silicon processing is planar lithography. This permits the simultaneous synthesis of billions of unique, interconnected elements on a single slab of silicon. The tremendous scale-up in complexity and associated cost reductions are profound. Sadly, even at VHF frequencies power-stage passive components don't yet lend themselves to easy inclusion in an integrated power process. Where the materials and processing challenges don't provide a significant barrier (for example, planar spiral inductors are commonly used on RFICs) the fundamental problems of reduced flux- and current-carrying area [9] as dimensions shrink conspire to reduce chip-integrated passive efficiency to the point where the cost-performance tradeoff is difficult to justify. Nevertheless, greatly reduced energy storage at VHF unlocks analogous benefits albeit at somewhat larger dimensions.

At VHF, magnetic materials are eschewed in favor of air-core structures. For frequencies exceeding about 50 MHz, it air-core inductors can offer higher performance at a given size than their magnetic-containing counterparts, with rare exception [8,32,33]. This is important for two reasons. First, it means that the inductance of a structure is set almost exclusively by

geometry. Second, without a need to process magnetic materials, existing printed-circuit techniques become attractive as a means of mass manufacturing. With numerous PCB processes achieving fine dimensional tolerances, the synergy is apparent–photolithography allows the rapid and inexpensive construction of the tuned networks common in VHF power converters with enough precision and repeatability to meet the requirements of mass production. A further consequence is that the incremental cost of additional components, now printed as traces in a PCB, is lower than in the discrete case. It avoids the need to produce, handle, and mount a separate component in favor of changing a lithographic mask. Combined with integrated transistors this helps reset the complexity-cost tradeoffs that often drive power converter design (particularly in the commodity market) such that new architectures and higher performance are possible.

3.2 Transformer Synthesis Background

There are several advantages to implementing transformers for VHF converters as planar elements in a PCB. First off, the typical wound bobbin and magnetic core structure is costly and cumbersome to manufacture. It's the single most expensive component in the flyback converter depicted in Figure 1.2 of Chapter 1. It's also generally difficult to attain precise control over the leakage inductance in a wire-wound transformer. This poses a challenge to the resonant systems necessary for VHF operation, which rely on a precisely tuned network to achieve high efficiency. Power density can also benefit since planar structures have a high ratio of surface area to volume.

The isolated Φ_2 converter described in Chapter ?? and pictured in Figure 3.1 requires a two-winding transformer. In the schematic, the transformer is modeled with primary-side magnetizing inductance and leakage components on both windings. Equations 3.1-3.5 are the transformer inductance matrix, giving the relationships among the terminal voltages and currents, and the parameters linking the terms in the matrix to the model. Where L_{l1} and L_{l2} are the leakage inductances, $L_{\mu 1}$ is the magnetizing inductance, k is the coupling coefficient, and N_P and N_S are the primary and secondary turns, respectively.

Once a particular tuning point for the isolated Φ_2 converter is established, the terms of the inductance matrix are fixed as is the coupling coefficient. For most designs of interest, the coupling coefficient is somewhere in the range of 0.5-0.6 versus the much higher values typical of a magnetic core transformer used in most converter topologies. With weak coupling and no core, the magnetizing and leakage terms are on the same order of magnitude. As a result, all of the terms must be tightly controlled to ensure that the proper tuning point is achieved. In contrast, cored transformers at lower frequencies are typically treated as ideal

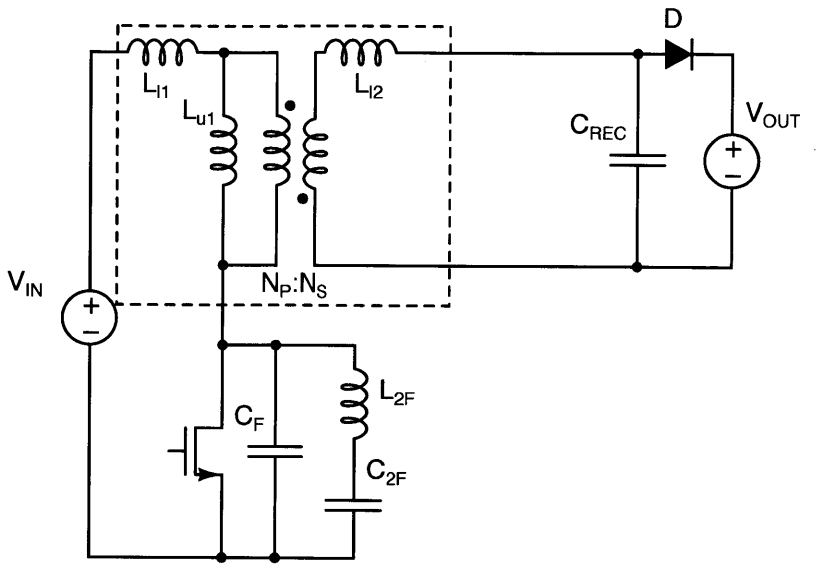


Figure 3.1: An isolated Φ_2 converter schematic including the transformer magnetizing and leakage inductances.

from a coupling standpoint while the leakage reactance is mitigated with external circuit techniques (eg., snubbing [2]).

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} L_{11} & L_M \\ L_M & L_{22} \end{bmatrix} \begin{bmatrix} i_1\prime \\ i_2\prime \end{bmatrix} \tag{3.1}$$

$$L_{11} = L_{l1} + L_{\mu 1} \tag{3.2}$$

$$L_{22} = L_{l2} + \left(\frac{N_S}{N_P}\right)^2 L_{\mu 1} \tag{3.3}$$

$$L_M = \left(\frac{N_S}{N_P}\right) L_{\mu 1} \tag{3.4}$$

$$k = \frac{L_M}{\sqrt{L_{11} \cdot L_{22}}} \tag{3.5}$$

From Equations 3.1-3.5 it is evident that given a particular converter design we are free to vary any of the transformer parameters (L_{l1}, L_{l2}, L_{u1}, N_P, and N_S) provided that the values of the inductance matrix are held constant. This reflects the fact that any number

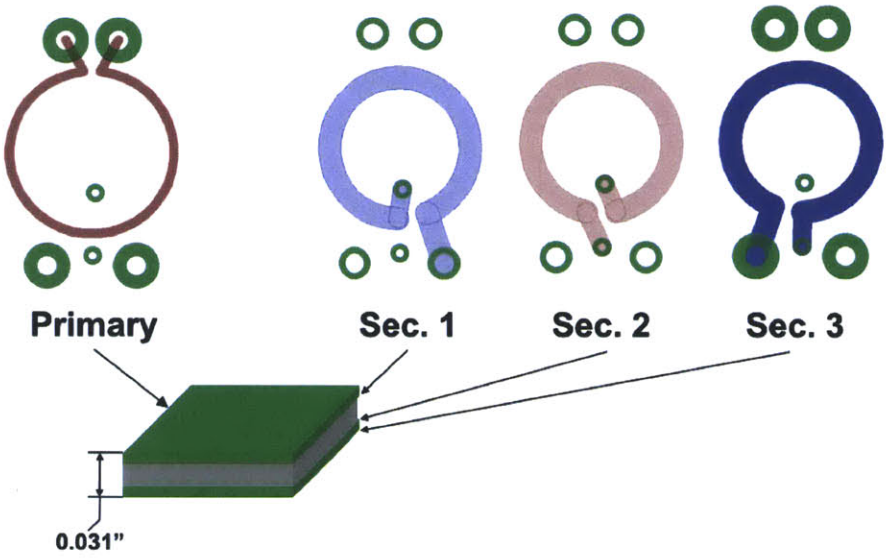


Figure 3.2: A 4-layer transformer with a 1-turn primary and 3-turn secondary. Each turn is on a separate layer which avoids increased loss due to flux shielding as compared to spiral designs.

of geometries can satisfy a particular inductance matrix [1]. Any particular geometry may have dramatically different size or loss parameters, however, and picking the best among them is the goal.

The desire to control all the parameters in the inductance matrix introduces the general problem addressed: How to synthesize a physical structure that realizes a given inductance matrix while finding a desirable tradeoff between volume and efficiency. The question arises often in power electronics, particularly in the context of components such as transformers and integrated magnetics [34, 35]. It also arises in the creation of coreless magnetics for parasitic compensation in filters [36]. In some cases the problem has been extended to include both inductive and capacitive reactances as part of the synthesis [37, 38]. This includes L-C-T structures which are designed to provide isolation and specific impedance characteristics, such as a series resonance to replace the tank and transformer in a resonant converter [39,40]. Such integrated designs usually employ magnetic materials. The resulting constrained flux path provides for simplifying assumptions that lead to analytical design equations which make finding an optimal structure relatively straightforward.

In the case of VHF switching frequencies, air-core magnetics are the norm because this avoids prohibitive core losses. Without a well-defined flux path, finding a sufficiently accu-

[1] Indeed, many different physical realizations are possible even for the same turns ratio and constant magnetizing and leakage inductance.

rate analytical solution to the inductance and resistance for most geometries is extremely difficult. A high degree of accuracy in the magnetic parameters is necessary for proper operation of the VHF resonant converters of interest. For the planar transformer structures considered here (eg., Figure 3.2), previous work provides approximate analytical solutions for the mutual and self inductances [1], but not the self or mutual ac resistance [41], nor a means to compensate for the change in inductance that arises when two coils are brought into close proximity. In the work by Tang [42] expressions are proposed to estimate efficiency, but these are accomplished by curve-fitting to experimental measurements and are only valid for a very specific set of structures and parameter variations.

In this work, as with some other resonant designs, [16], the transformer magnetizing and leakage inductances serve as an integral part of the converter energy storage. This is desirable at VHF because it circumvents the need to design around transformer parasitics and it reduces the component count of the power stage, aiding the ultimate goal of achieving a tightly integrated system. As was pointed out above, the leakage and magnetizing inductances play a critical role in tuning the converter to operate efficiently at VHF. Therefore, the transformer's inductance matrix is fully specified by the converter tuning point. Designing a transformer with the right inductance parameters while offering a good trade-off between volume and efficiency thus requires the ability to accurately compute the inductances and ac resistances at the operating frequency of the structure while including skin and proximity effects. This is possible for a given structure using any of a number of finite element field-solver packages.

While numerical solution can provide the accuracy required, it comes at the penalty of a heavy computational burden. For the relatively simple geometry of a two-winding planar transformer, simulation of a single design at sufficient accuracy for our purposes can be accomplished in a matter of minutes. However, answering the inverse problem with numerical simulation—finding which geometry provides the desired inductances while satisfying size and efficiency constraints—requires many successive simulations. Any algorithm that solves this problem essentially takes the form of evaluating a large pool of candidate geometries, picking those that match the inductance matrix, and analyzing the efficiency of the matching subset to find the loss-size tradeoff. If the pool of potential candidates is too large, the computational overhead is massive; too small and a good design may be missed. Thus the effort in solving this synthesis problem is establishing a means of finding acceptable designs without requiring more computation than may be performed in a reasonable amount of time.

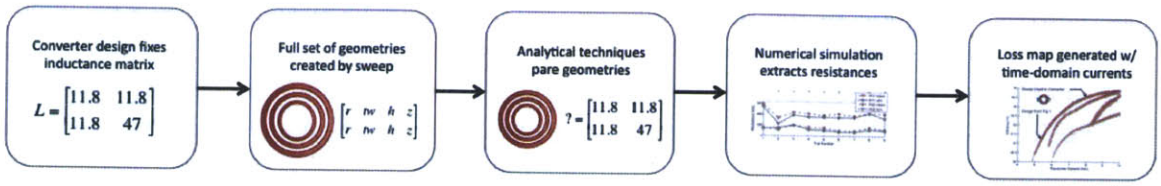


Figure 3.3: Flowchart depicting general approach used to for synthesis

3.3 Algorithm Overview

The transformer designs considered in this work are constrained to planar PCB fabrication techniques because of their widespread availability and the ease with which complex systems can be fabricated. This de facto constrains the synthesis space dramatically. Even so, finding an optimal transformer design involves searching several parameters and an exhaustive search can easily require tens or hundreds of thousands of simulations. This would translate into days and weeks of compute time where each simulation requires a few minutes.

To avoid very long solution times, the general synthesis algorithm employed here generates a set of geometries, pares them to a much smaller set by analytical means, and submits the results to detailed numerical simulation to account for skin and proximity effects. Figure 3.3 shows a simple flow chart of the approach. The first step is the design of a power converter, which fixes the values in the inductance matrix. Once the matrix is known, a set of geometries is analyzed providing the values of L_{11}, L_{22}, and L_M. Only a subset of the transformers will match the inductance matrix to within a predefined tolerance. These are simulated using numerical packages such as FastHenry or Comsol to calculate the resistance parameters for the transformers as well as inductance variations due to skin and proximity effects not accounted for analytically. The power loss in each geometry is then calculated using the time-domain current waveforms derived from the same converter simulation that provided the inductance matrix. At this point, a map of transformer dimensions vs. efficiency is generated and the final selection is made based on the targets of the external system design.

The use of analytical pre-processing reduces the solution time by as much as three orders of magnitude. Even so, numerical simulation still eats the bulk of the compute time. As a result the geometric sweep parameters and their granularity must be chosen carefully. The post-analytical locus of viable transformers can be hundreds, thousands, or more depending on the number of swept parameters and choice of step size. Without sufficient trials, however, optimal designs could easily be missed. A useful means of determining whether the parameter steps are fine enough is to simply look at the final size-efficiency map for discontinuities. For instance, a set of transformers with the same basic configuration (number of turns per winding, turn locations relative to one another, etc.) will tend to have

a consistent trend between diameter and loss–usually loss drops as diameter increases. If the parameters are swept without enough granularity for the desired inductance tolerance in the analytical stage, then discontinuities will appear that may result in a missed global optimum. This is usually corrected by decreasing the step size.

Another way to miss a global optimum is to choose a set of parameters that precludes the relevant cases from ever being tested. Since a brute-force sweep through all geometry cases is out of the question, one way to deal with this is to choose a parameter set that couples the individual turn parameters in a geometrically significant pattern. For instance, one configuration that is considered in this work is helical windings implemented by placing the turns of a single winding on successive layers of the PCB so that they progress along the z-axis (in effect forming a helix). Each turn has its radius, trace width, thickness, and z-location as a parameter. Varying all of these independently is the most robust means of ensuring that the best design is found, but it makes for a very time consuming process.

A different approach to find successful helical designs is to set the radii and trace widths of all the rings in a given coil identical. This foregoes generality for expedience and allows the description of the coil with two parameters rather than 4x the number of turns. By adding in a taper factor, such that each turn's radius is a function of z-location, only one more parameter is necessary but the additional freedom nets about a 1% improvement in transformer efficiency.

3.3.1 Analytical Solution

The analytical models used to pare the search space are taken from work that specifically addresses the challenges of calculating the self and mutual impedances in planar magnetic structures [1]. In planar structures the thickness of a turn is typically much smaller than it's width. Thus the calculation of the mutual inductance between two concentric filaments as originally presented by Maxwell [43] and extended by Grover [10] starts to break down. This is primarily because the current in a given turn changes significantly along its cross section. This problem is specifically addressed in [1] for the case of two concentric rings.

The basic geometry of the two concentric rings under consideration is illustrated in Figure 3.4. The formula for the mutual inductance between them is:

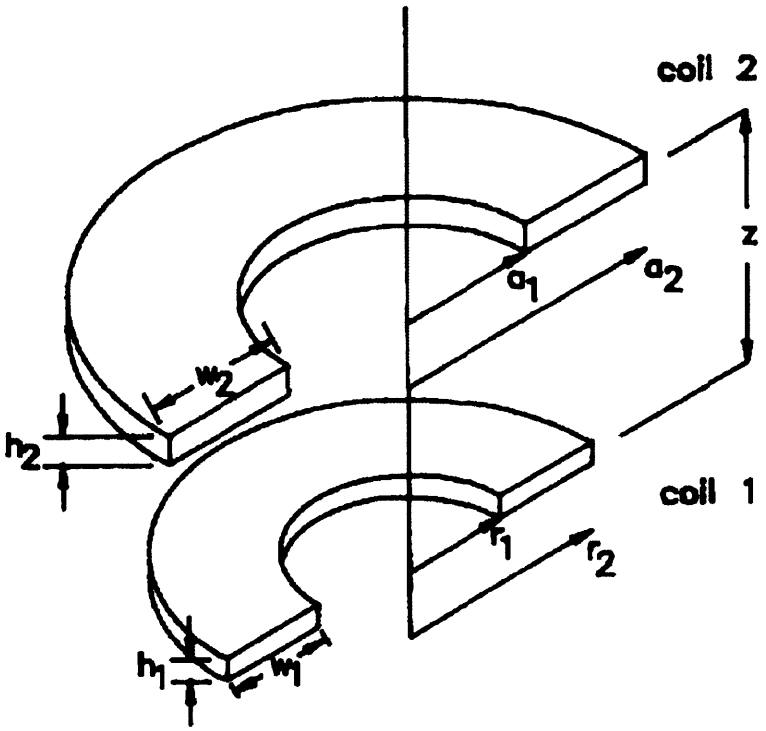


Figure 3.4: Concentric planar ring structure that [1] solves for mutual inductance.

$$M_{12} = \frac{\mu_0}{h_1 h_2 ln\frac{r_2}{r_1} ln\frac{a_2}{a_1}} \int_0^\infty S(kr_2, kr_1) S(ka_2, ka_1) \cdot Q(kh_1, kh_2) \exp^{-k|z|}\, dk \tag{3.6}$$

$$Q(kx, ky) = \frac{2}{k^2}\left[\cosh k\frac{x+y}{2} - \cosh k\frac{x-y}{2}\right], z > \frac{h_1 + h_2}{2} \tag{3.7}$$

$$= \frac{2}{k}\left(h + \frac{\exp^{-kh} - 1}{k}\right), z = 0, x = y = z \tag{3.8}$$

$$S(kx, ky) = \frac{J_0(kx) - J_0(ky)}{k} \tag{3.9}$$

Where J_0 is a Bessel function of order 0, r_1 and r_2 are the inner and outer radii of coil 1, a_1 and a_2 are the inner and outer radii of coil 2, h_1 and h_2 are the thicknesses of coils 1 and 2 respectively, and z is the center-center vertical distance separating the coils.

The self inductance is calculated using the Maxwell formula for the mutual inductance between two filaments:

$$M = \mu_0 \sqrt{ar} \frac{2}{f} \left[(1 - \frac{f^2}{2}) K(f) - E(f) \right] \tag{3.10}$$

$$f = \sqrt{\frac{4ar}{z^2 + (a+r)^2}} \tag{3.11}$$

Where K(f) and E(f) are complete elliptic integrals of the first and second kind, respectively, z is replaced by the geometric mean distance of the coil from itself: $GMD \approx 0.2235(r_2 - r_1 + h)$, and the filament center is placed at the geometric mean of the coil radii: $r_0 = \sqrt{ar}$. The above equations are not closed form solutions and must be solved by numerical techniques, but this is easily accomplished on the order of seconds.

These formulas provide the self and mutual terms for a single turn or turn-pair. The transformers considered here are generally more complex than two single-turn windings, however. Designs where each winding is a multi-turn spiral with all turns in the same plane were considered as well as those where the spiral windings are essentially helical by virtue that each turn is formed on successive PCB layers (see Figure 3.2). The self inductance of a single winding comprising multiple turns is the sum of the mutual inductances between each turn and the self inductances of each turn, ignoring skin and proximity effects. For a two winding structure, the self inductances of each coil are computed in this way and the mutual inductance between the two windings is just the sum of all the mutual terms that couple the turns on opposite windings. This is easily handled algorithmically by treating the entire transformer structure as a set of rings numbering N. Then an N x N inductance matrix is composed using the self and mutual inductance formulas given above (Equations 3.6 and 3.11).

$$L = \left[\begin{array}{cccc:cccc} L_{1,1} & L_{1,2} & \cdots & L_{1,P} & L_{1,P+1} & L_{1,P+2} & \cdots & L_{1,N} \\ L_{2,1} & L_{2,2} & \cdots & L_{2,P} & L_{2,P+1} & L_{2,P+2} & \cdots & L_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{P,1} & L_{P,2} & \cdots & L_{P,P} & L_{P,P+1} & L_{P,P+2} & \cdots & L_{P,N} \\ \hdashline L_{P+1,1} & L_{P+1,2} & \cdots & L_{P+1,P} & L_{P+1,P+1} & L_{P+1,P+2} & \cdots & L_{P+1,N} \\ L_{P+2,1} & L_{P+2,2} & \cdots & L_{P+2,P} & L_{P+2,P+1} & L_{P+2,P+2} & \cdots & L_{P+2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{N,1} & L_{N,2} & \cdots & L_{N,P} & L_{N,P+1} & L_{N,P+2} & \cdots & L_{N,N} \end{array}\right]$$

If care is taken to treat the primary turns first, by numbering them 1 to P, where P is the total number of primary turns, then reducing the N x N matrix to the 2 x 2 inductance matrix of the two winding transformer is readily accomplished. The block sums of the sections delineated above will give the desired inductance terms. Specifically, collecting the upper-left terms gives L_{11}, the lower right terms L_{22}, and either the lower-left or upper-right terms can be summed to get L_M for the entire structure. These block sums are given in Equations 3.12-3.14.

$$L_{11} = \sum_{i=1}^{P}\sum_{j=1}^{P} L_{i,j} \tag{3.12}$$

$$L_{22} = \sum_{i=P+1}^{N}\sum_{j=P+1}^{N} L_{i,j} \tag{3.13}$$

$$L_M = \sum_{i=P+1}^{N}\sum_{j=1}^{P} L_{i,j} \tag{3.14}$$

With this approach the inductance matrix for several thousand trials can be computed in a matter of minutes. The subset that is within a desired tolerance of the exact inductance parameters computed from the circuit simulation are culled into a set of viable geometries. While this allows rapid convergence toward transformer designs that have the right terminal characteristics, neither skin nor proximity effects are taken into account. Both the inductance and AC resistance are influenced significantly over the design frequency range by these factors.

3.3.2 Numerical Simulator Packages

The basic requirements on the numerical simulation packages for this work include being fast and having software hooks for control by an external program (in this case MATLAB. Two such packages that meet these requirements are FastHenry [44], a numerical simulation program designed to calculate inductance and resistance for 3-D interconnect problems, and Comsol, a general purpose FEM package that has tight integration with MATLAB owing to its FEMLAB origins [2].

[2]FEMLAB was originally a MATLAB package for FEM simulations that eventually evolved into Comsol.

FastHenry is particularly attractive because of it's ease of use and speed. Geometries are described by input files similar to a SPICE deck that can be easily created automatically in MATLAB. It solves magneto-quasistatic problems using a multipole-accelerated GMRES solver where solution time grows order-n with the number of elements [44]. This is useful because it enables a 3D representation of a transformer to be solved in a few minutes, including proximity and skin effects. It does have limitations, however, as elements that are located in close proximity to one another require the use of a different algorithm. Correspondingly, the solution time to grows more rapidly than order-n when the problem is discretized into many closely spaced elements to handle skin and proximity effects accurately. This caused a problem in the first set of transformer designs which drove the simulator to find solutions where the center of the transformer was almost completely covered by copper, effectively creating a flux shield that hurt efficiency and inductance. This is detailed in Section 3.3.5. Since FastHenry enables 3D simulation, a more realistic model of the transformer may be simulated, as opposed to a set of concentric rings. For instance, ring-ring interconnect and vias were included in the simulation in an effort to improve the agreement between simulation and experiment.

Comsol has the ability to be a full 3D solver, but for this application it is impractically slow. The mesh size directly relates to accuracy and solution time. Solutions with the required degree of accuracy take hours to simulate. In 2D axisymmetric mode, the solution times are comparable to what FastHenry achieves, about two minutes for a typical transformer geometry. This constrains the calculation to be 2D axisymmetric, but it also allows finer discretization of the cross-sections of each transformer turn, which enhances the ability to deal with skin and proximity effects. The overall accuracy of the 2D solution for transformer cases where there is not substantial flux shielding proved to be significantly better than FastHenry computing inductance values to about 15% vs. the peak errors of 35% with the latter package.

3.3.3 MATLAB Scripts

To manage the simulations and find a reasonable set of transformers, a series of MATLAB scripts was developed. They are included in Appendix A.2 and implement the following functions:

1. Build set of transformer geometries based on a pre-defined parameter set
2. Solve for L_{11}, L_{22} and L_M analytically
3. Cull the transformer candidates to those meeting the inductance parameters

4. Create the FastHenry run decks, or control Comsol to perform numerical simulations
5. Manage simulations across multiple computer systems to arrive at solutions more rapidly
6. Extract results of numerical simulations and compute transformer losses
7. Produce efficiency-size mapping of successful transformer designs

Transformer geometries were constructed by simply deciding on the set of relevant geometric parameters and then implementing code that permutes the full set while trapping for non-physical conditions and repeat geometries. The most important aspect of building a set of test geometries is to pick parameters that correspond to meaningful changes in transformer performance. For instance, stepping the diameter by too small a value leads to many transformer candidates that will have almost identical loss, wasting simulator time. Some versions of the MATLAB code forego pre-building geometries in favor of creating working geometries on the fly, for instance, by implementing a binomial search algorithm around coil diameter or other parameters. These cases will be discussed in the sections below.

Analytical solution was handled using MATLAB's built in numerical solvers (the quad function) on Equations 3.6-3.11. Care was taken to minimize the total number of calculations necessary per transformer by only computing the lower diagonal of the inductance matrix and dynamically adjusting the solver tolerance depending on the particular geometry. Computing the block sums of the full N x N to generate the 2 x 2 inductance matrix is a straightforward process that gives the values of L_{11}, L_{22}, and L_M.

Once the inductance values from the full set of geometries is available, a search for matches is performed. Transformers with an inductance matrix having all terms within 2 nH of their desired value are considered matches and stored for numerical simulation. Ultimately, the error in the follow-on numerical simulation determines whether or not a design is workable for an actual converter. As a result, the choice of 2 nH for the parameters is more of a means of filtering the total number of cases. In some instances it was necessary to relax this constraint, and in others to tighten it. The appropriate choice depends on the geometry space being explored and how strongly the geometric parameters influence the inductance matrix over the sweep range.

For the subset of transformers to be simulated numerically, MATLAB generates the ASCII run decks for FastHenry. Creating the run decks requires MATLAB to create discretized geometries from the chosen parameter set that are compatible with the FastHenry flat-plate construct. Everything in the simulator must be represented by a series of rectangular elements which have width, length, and height parameters as well as a location on the simulation grid. Each element's normal can take any direction in 3-space. To account

for skin and proximity effects, FastHenry will automatically cut each element into a set of filaments perpendicular to the cross section. The COMSOL simulations were handled more directly, as that program has a means to accept geometry inputs from MATLAB as a set of functions.

When the run decks are ready, MATLAB calls a small program running as a shell script that spawns multiple FastHenry instances simultaneously. This allows better utilization of system resources–a single thread of FastHenry is only able to use a relatively small portion of a workstation [3]. A typical run requires several hundred FastHenry simulations, but some versions exploring a large set of geometries require tens of thousands of simulations. For these cases, a pool of computers is necessary and MATLAB scripts perform dynamic load balancing to keep all machines occupied. Load balancing is necessary because the time to solve a geometry varies by a factor of 3-5 depending on the total number of elements necessary to represent it, and most machines would sit idle if the simulations were simply farmed out evenly, reducing the effectiveness of a cluster.

Simulation results are then collected and used to compute the losses for each design relative to a set of current waveforms extracted from SPICE simulation of the target converter. Once the loss is tabulated, the scripts produce a scatter plot of the loss for each transformer vs. maximum diameter providing a convenient means of picking a design with the right trade-off.

3.3.4 Transformer Geometries

The specific transformer geometries designed and tested are discussed below. As mentioned above, even with the constraint of a planar substrate, there is still a very large number of possible designs. The first type of geometry attempted was to use planar spirals placed across a substrate to create 2-winding transformers. Follow-on designs used helical windings where each turn in a given coil was constrained to be identical. This helped to improve efficiency and agreement with the simulation results. Additional broader sweeps were performed to determine how much efficiency gain was possible by opening up the search space.

3.3.5 Spiral Winding Transformers

A natural starting place is to implement the transformer as two spiral windings where each winding is on an opposite face of a PCB core. This approach has been widely practiced [42,

[3] For reference a "typical" workstation in this case is a 2.93 GHz quad Core i7 with 12 MB of RAM

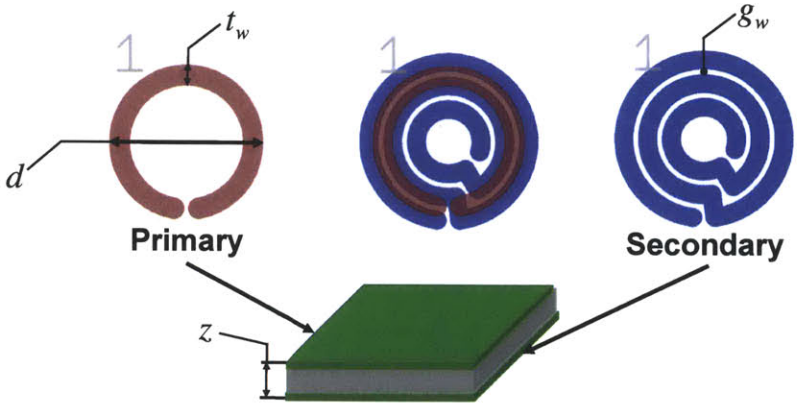


Figure 3.5: PCB geometry of a 2-winding spiral transformer. The parameters optimized for are the diameter, d, the trace width, t_w, the gap width, g_w, and the z-spacing, z and the number of turns in each winding.

Table 3.1: Spiral Coil Parameters

Parameter	Range	Step Size	Description
Outer Diameter	4 mm - 9 mm	200 um	Max OD of winding
Trace Width	160 um - max	100 um	Width of PCB trace
Turn Count	1-10	1	Number of turns in a winding
Gap	Min. gap 160 um	100 um	Turn-turn spacing
Min. Inner Diameter	250 um	NA	Minimum ID of winding

45–49] and the primary extension here is that we desire to completely specify the inductance matrix and achieve a desirable tradeoff between size and loss for the isolated Φ_2 converter. Figure 3.5 shows a PCB layout of one transformer geometry from this series. The parameters that were swept to generate the set of geometries are labeled on the drawings. The desired L-matrix values are L_{11}=11.8 nH, L_{22}=47 nH, and L_M=11.8 nH.

For this transformer design run, test geometries were generated in a batch by first generating the set of possible spiral windings and then permuting the two-winding sets to create transformers. To generate the test geometries, d_w was swept from 4 mm to 9 mm in 100 um steps. The minimum permitted inner diameter of the spiral was set to 500 um. For each O.D. an inter-turn gap is picked, starting with the minimum value of 160 um, set by the PCB process. The number of turns was allowed to vary from 1 to 10 and for each turn count, trace width was swept until the minimum inner radius could not be satisfied. The process was then iterated for successively larger gaps until the gap size precluded the minimum trace width from meeting the minimum radius requirement. Table 3.1 lists the parameters and ranges over which each vary.

The above process produces about 7700 discrete coil geometries. Each of these coils is analyzed to find the set of coils that has the required self inductance - either L_{11} or L_{22}. Setting the tolerance to 1% variation produced 25 coils having a self inductance within 1% of L_{11} and 52 coils having a self inductance within 1% of L_{22}. These results are plotted in Figures 3.6 and 3.7. Clearly, there was no need to construct coils with as many as 10 turns for the inductance ranges required as the primary never exceeds 2 turns and the secondary 4 turns. In both cases, the diameter increments resulting in corresponding changes in g_w and t_w to meet the required inductance value, until a diameter is reached where the self inductance exceeds the upper bound regardless of the other parameter values.

One of the difficulties with setting up the geometry sweeps using spiral inductors is the tendency of the coils to interfere with one another at regular intervals. That is, some sets of coils will naturally have windings where the turns are overlaid leaving the interwinding gaps available for flux traversal, while others will have winding structures where the turns of each winding alternately overlap the gaps of the other. As a result, when sweeping over a set, it's possible to miss the inductance requirements for an entire diameter value. Excessive flux shielding compromises coupling and it becomes impossible to satisfy the inductance matrix. Avoiding this problem forces some parameters such as trace width or gap width to be selected more granularly, adding to simulation time. This was dealt with by simply tweaking parameter ranges and step sizes to minimize the number of missing transformer diameters, but a more sophisticated algorithm could be developed to adjust the spacing or trace width in consideration of the other coil.

The successful coils are next permuted to calculate the mutual inductance of each coil pair. In this case, the separation between the coils was selected to fit on standard PCB stack ups ranging from 8 mil to 62 mil in thickness. Figure 3.8 plots the 152 designs that have the appropriate L-matrix values. Changing the number of turns, trace width, or gap strongly influences the diameter of the secondary winding even while its self inductance and mutual coupling remain nearly constant. This implies a wide performance variation in terms of size and loss.

FastHenry simulation of the 152 transformer cases provides the resistance terms. Follow-on time domain simulation using the converter current waveforms yields the expected loss for each transformer. The design power of the converter is 10 W, and the power vs. diameter scatter plot (Figure 3.9), shows the best transformer designs simulate to better than 94% efficiency at that power level. The size-loss map shows a general trend of higher efficiency for larger diameter, as expected. The vertical columns of trial geometries derive from the discrete steps in diameter that were used to create the test cases. Moving up any given column is a less optimal design, but still satisfies the inductance matrix. Figure 3.10 shows several designs from the 6.6 mm column. The lowest loss designs, according to the

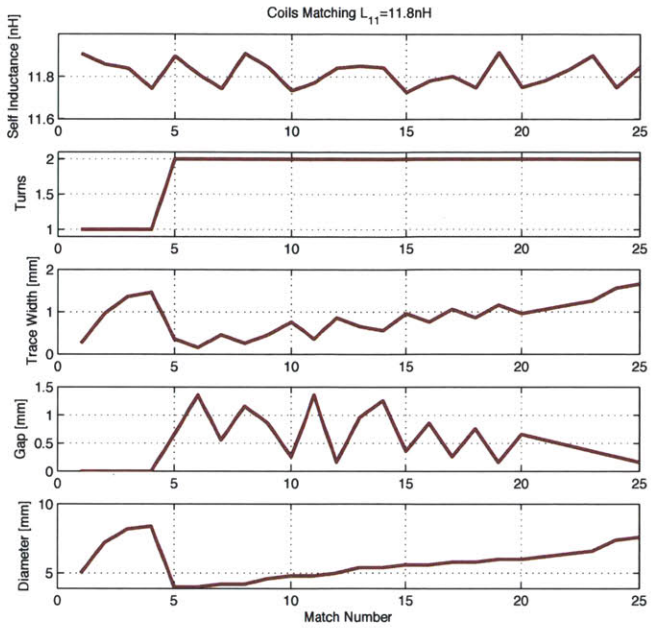


Figure 3.6: 25 coils have the desired self inductance, L_{11}. Their geometric parameters are plotted together with the calculated inductance.

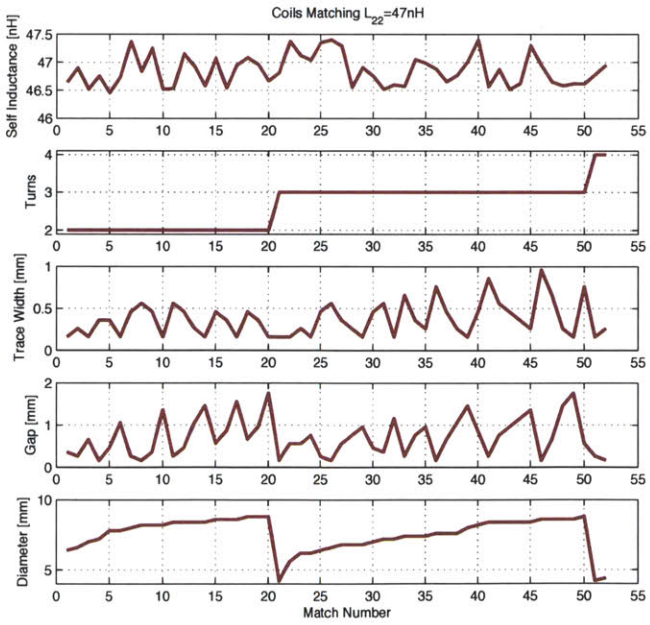


Figure 3.7: 52 coils have the desired self inductance, L_{22}. Their geometric parameters are plotted together with the calculated inductance.

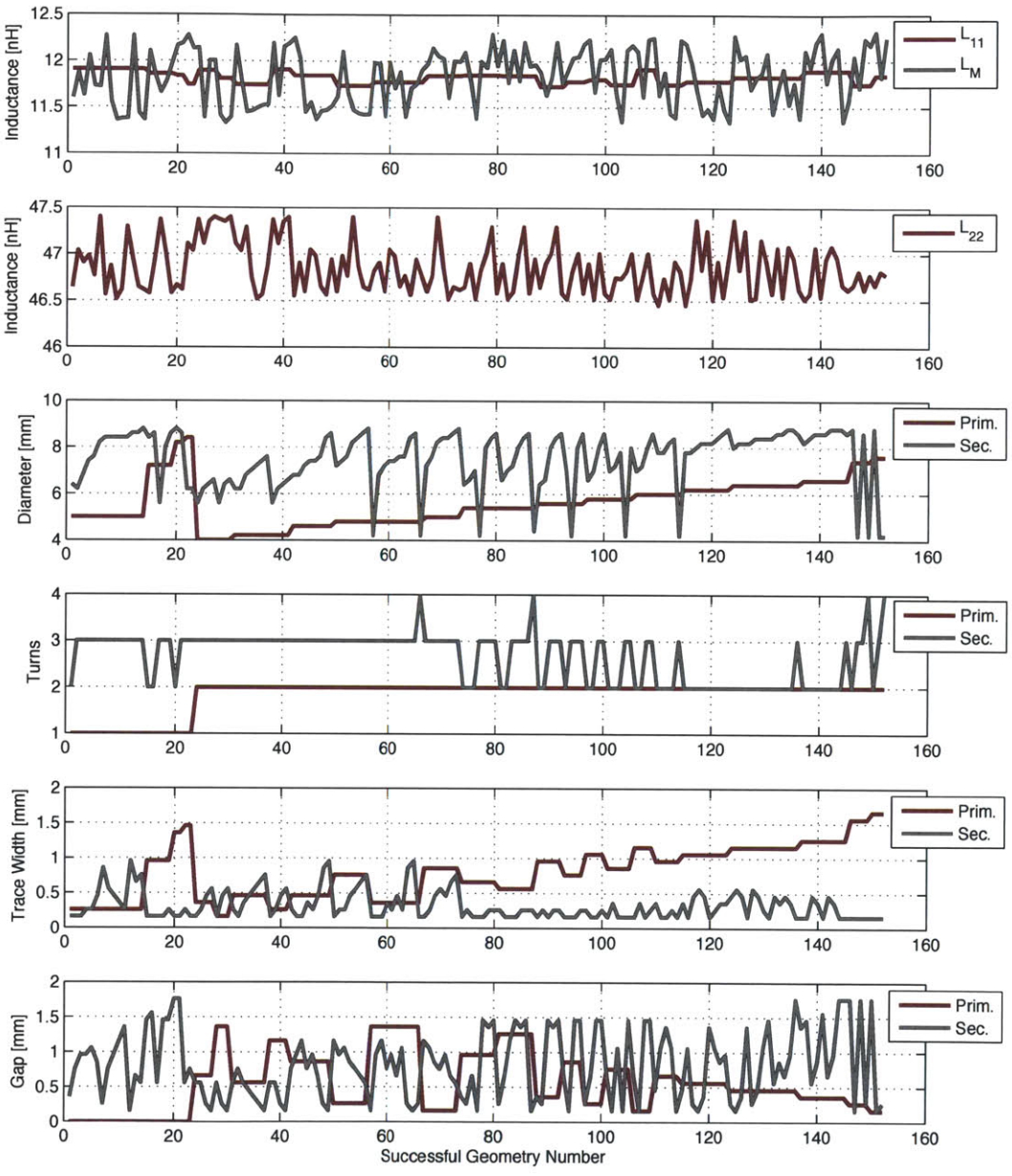
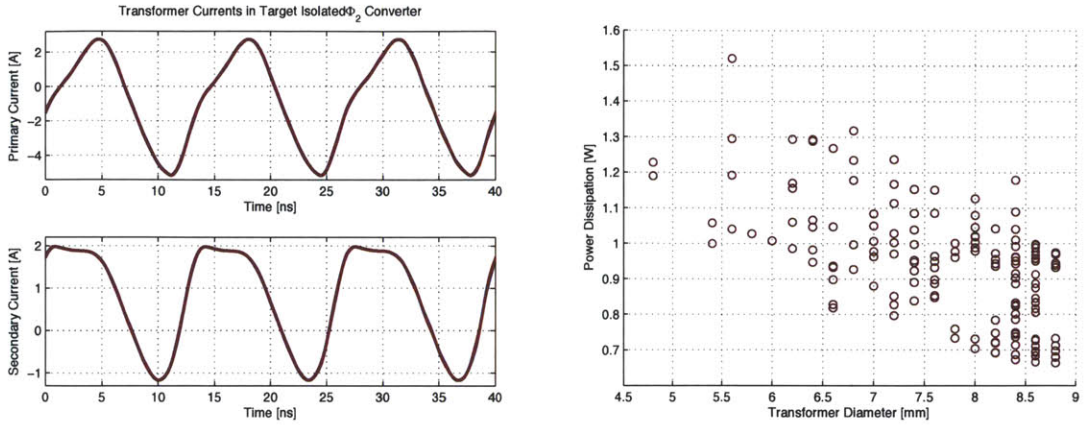


Figure 3.8: Plot of the successful transformer designs as captured analytically.



(a) Converter currents

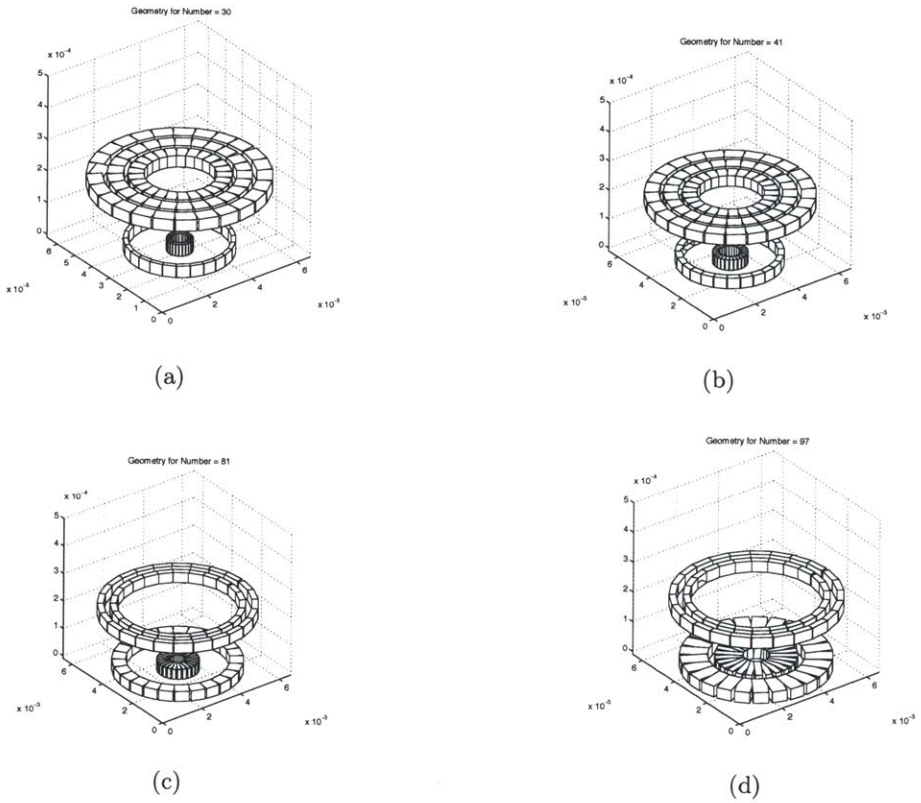
(b) Size-loss map

Figure 3.9: Converter currents used to baseline transformer performance and size-loss map of successful transformer designs for isolated Φ_2 converter.

simulator tend to have increasing amounts of copper filling the primary winding window, whereas those exhibiting the highest losses have more copper in the secondary winding. This helps to offset the significantly higher current in the primary. It should be pointed out that images have been deliberately scaled in the z-dimension to make the structures easier to see, as the actual designs have very large x- and y-dimensions when compared to the z-dimension.

The inductance parameters from the analytical solution and FastHenry are compared in Figure 3.11. The mutual inductance calculation and simulation seem to be in good agreement. Winding self inductances are about 10% lower in the FastHenry simulation than predicted analytically. This is not entirely unexpected as the analytical calculations for self-inductance do not account for the presence of a nearby coil. Where skin and proximity effects are significant, as they will be in these structures because of the 75 Mhz fundamental and the desire for physically small coils, co-locating two coils will tend to reduce their respective self inductance. One mechanism by which this occurs is the rejection of flux from the internal volume of the winding at high frequency. While this is exactly the skin effect, it will tend to lower the inductance of the adjacent coil because that volume is now unavailable to store energy in the magnetic field. There is also the redistribution of current on the winding owing to the presence of the other coil. This will inevitably change the total inductance as well, since it depends on the distribution of current in space rather than the shape of the winding, per se.

In order to evaluate the simulation results, the 152 geometry test cases were further divided into a set of transformer designs where the inter-spiral spacing fit on a standard PCB stack



(a) (b)

(c) (d)

Figure 3.10: FastHenry transformer geometries from the 6.6 mm column arranged from lowest efficiency to highest efficiency. More copper is favored in the primary winding owing to the higher currents.

up. These designs were then fabricated on a 1 oz., FR-4 PCB process. Figure 3.12a shows the board layout of the transformers with the primary and secondary traces separated for visibility. Impedance measurement on an Agilent 4395a characterized the degree to which the devices met the target inductance and resistance values.

A series of three one-port measurements were used to extract the inductance and resistance parameters of the transformer. The first two measurements were the impedance looking into the primary with the secondary open and vice versa. The third measurement relied on shorting the secondary and measuring the impedance at the primary again. The values at 75 MHz, properly extracted, give the transformer parameters of interest. The inherently low coupling coefficient required by inductance matrix made for relatively straightforward measurement of the inductance parameters as there were no numerical scaling issues inherent to the measurement. Loop inductance in the measurement fixture was subtracted term-by-term by first measuring a shorted-turn mock-up transformer. The primary challenge in the impedance measurement pertained to the resistance measurements which were on the order

of the contact resistance of the measurement fixture. In order to minimize contact resistance of the jig, a fixture was created that attached to the analyzer head by mechanical means to ensure low contact resistance. Individual transformers were then soldered in place.

The measurement results are plotted in Figure 3.12b. The most striking feature of the plots is the departure of the measured values from the simulated values for devices numbered nine and higher. For these devices the inductance is substantially smaller than the target

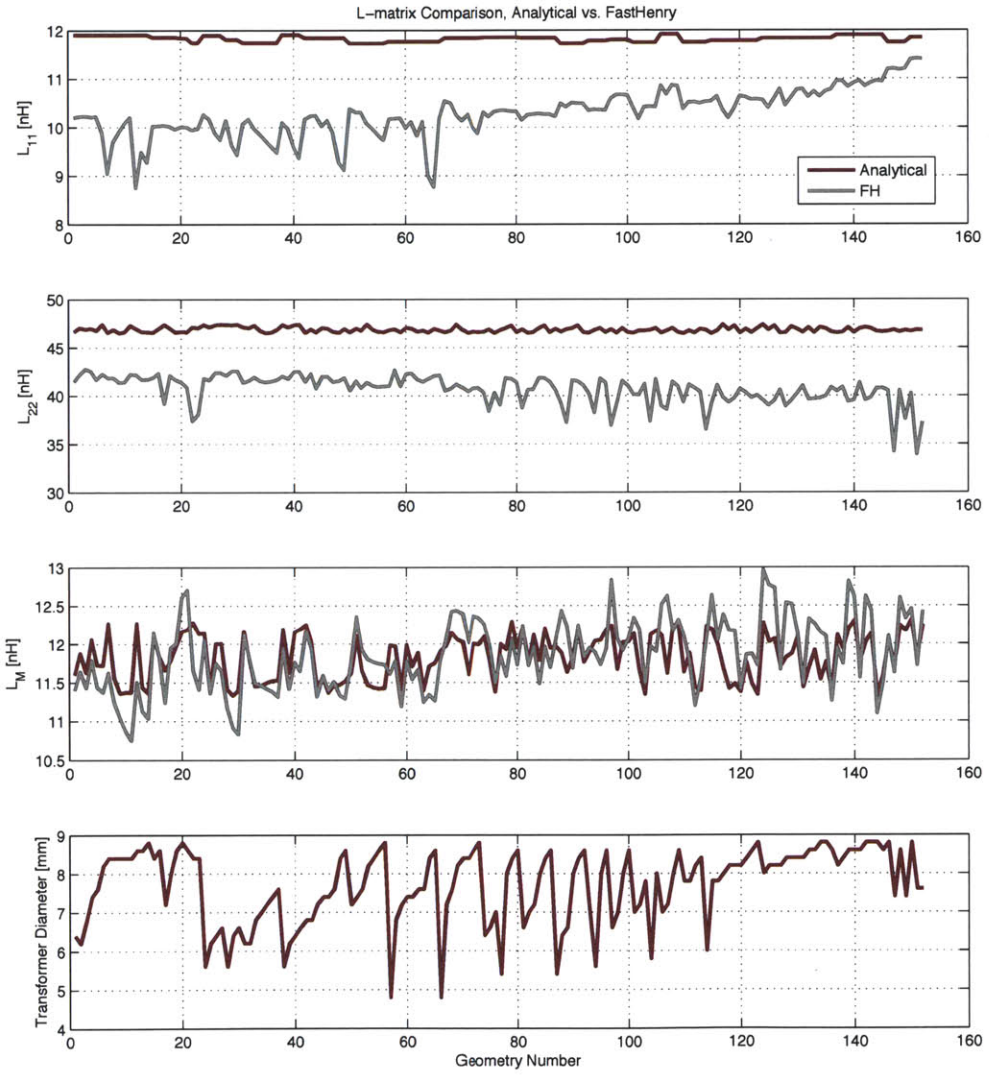
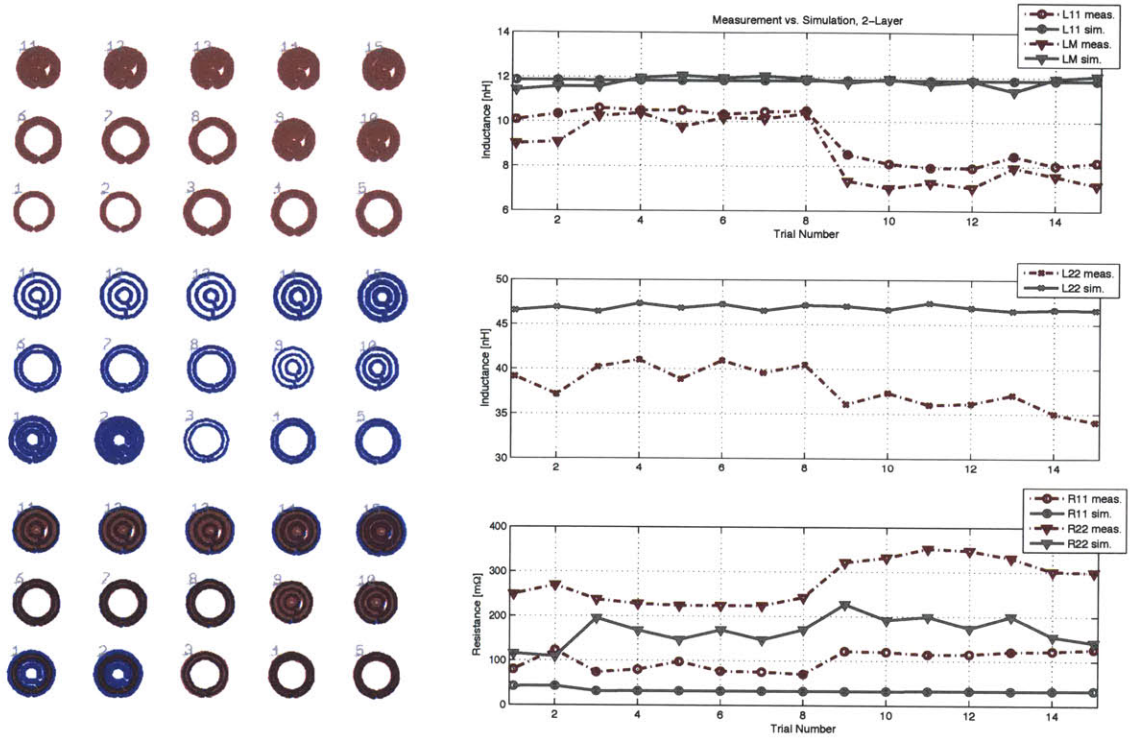


Figure 3.11: Plot of the successful transformer designs as captured analytically.



(a) PCB tracks for measured spirals

(b) Measured results

Figure 3.12: Fabricated PCB and measured results showing the discrepancy between measurement and simulation as the transformer winding area becomes increasingly filled with copper.

value and the resistance is comparatively higher. The measured and simulated inductance values differ by 41% in the divergent region and by about 25% for the devices 1-8. Similarly, the measured resistance is over 1.5x higher than simulation results in the divergent region. Examination of the transformers in the adjacent figure exposes the underlying reason. The devices that differ greatly are all designs where copper from the windings fills most of the transformer area, effectively shielding the flux path. Flux tubes that would otherwise traverse the center of the transformer instead impinge on the copper surfaces there and induce eddy currents leading to loss and reduced inductance. This potential for this problem was noted during the design phase, but it was not possible to configure the simulator to produce the same behavior (in this case FastHenry). As a result, the "best" transformer geometries were predicted to be those with their centers filled with copper (as for instance in Figure 3.10d).

3.3.6 Helical Winding Transformers

Where copper impeding the flux path is an issue, one practical solution is to use windings that avoid placing too much copper at the center of the transformer. However, the intersection between a wide-open center and spiral winding structures does not lead to the desired level of performance. Those designs do not exceed about mid-80% efficiency, per the results of the previous section. Further, geometries with copper in the center which simulated in FastHenry as over 90% efficient actually have much worse performance, typically less than 80% when measured experimentally.

Another means to get the same coupling and self inductances without placing much copper in the center of the winding is to use multiple layers of a printed circuit board and place each successive turn of a winding on the next layer of the board (see Figure 3.2). This effectively forms a helix, although one with a very fine pitch owing to the close inter-layer spacing typical of most PCBs. The primary constraint on this approach is that each additional turn requires another PCB layer if minimal shielding is desired. Under this constraint, realizing inductances larger than a few tens of nanohenries could become impractical. There is always the option to add multiple turns to the same layer, but only at the expense of more flux shielding. Fortunately, for the targeted converter design space, a few 10s of nanohenries is all that is required and it was possible to create transformers in a standard 4-layer PCB process where no winding exceeded three turns.

For windings comprising multiple turns with identical radii and trace widths, many turns can be overlaid without exposing additional copper. As mentioned, this keeps the amount of copper in the center of the transformer to a minimum. Besides reducing flux shielding, it also helps to make the transformer smaller while realizing the same coupling. The effect results because helical windings allow larger turn-to-turn flux linkage while occupying a given surface area on the board. It further gives greater freedom for coils with substantially different inductance values to achieve a the desired level of coupling. For instance, here we try to achieve a device with $L_{11} = 11.8$ nH and $L_{22} = 47$ nH. Most of the spiral designs have significantly different primary and secondary coil diameters, which makes coupling less effective. Those that have similar diameters inevitably expose significant copper resulting in lossy eddy currents (see designs in Figure 3.10). With a helical design, one possibility is to have turns of nearly identical diameters but more windings on the secondary to get the required inductance.

3.3.6.1 Coils with Identical Turn Parameters

In order to limit the number of test geometries, we can specify that the trace width and radius of all turns in a given coil are identical. This also has the effect of maximizing the open area in the center of the transformer for a particular coil, minimizing shielding. In addition, if the trace width and radius of each turn in a given winding be equal, it becomes easy to perform a binomial search for the coil radius given the desired self inductance.

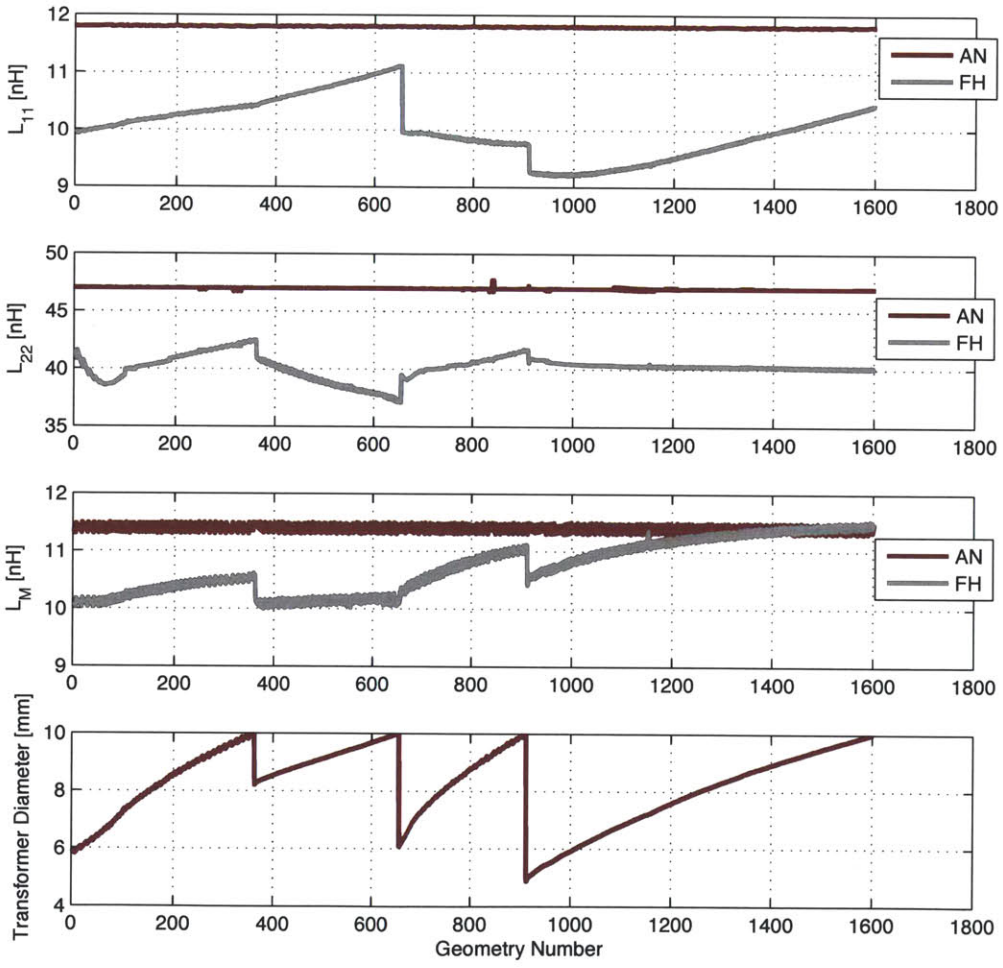


Figure 3.13: Comparison of analytical and simulated results for helical winding transformers.

The algorithm to find coils with the desired self inductance begins by picking a trace width (starting with the minimum value). Then the self inductance is calculated for the minimum and maximum allowed radii. If the desired inductance lies between these endpoints, the midpoint value is calculated and whichever half interval the value lies between is bisected again. This continues until the inductance converges to a predefined tolerance (0.5 nH in this case). Since the inductance is monotonic with radius, convergence is assured, although some care must be taken to adjust the radius step size to avoid stepping over the tolerance regime. The total number of calculations necessary to generate a set of coils this way is substantially smaller than building a full set of trials. By repeating this procedure for each desired trace width and turn count, primary and secondary coils sets are created. Permuting the coil sets and filtering for only those that have the required mutual inductance gives the complete set of transformers to be simulated.

Figure 3.13 shows the comparison between the analytical and simulation results for a set of trials. Examining the analytical curves reveals the fingerprints of the synthesis method - the self inductances are very close to the desired values while L_M oscillates over the tolerance range as the various coil pairs are tested. The maximum deviations in L_{11} and L_{22} between the analytical and simulation results is about the same as for the spiral case. L_M shows a slightly larger deviation. The efficiency-diameter tradeoff for this set is plotted in Figure 3.14. In contrast to Figure 3.9 the transformers are not lined up in neat columns by diameter because the coil radius was found by binomial search. Instead, the results trace out a a set of relatively smooth curves that bound the performance characteristics of each device set.

The multiple curves in Figure 3.14 derive from the various coil configurations that were simulated. The designs were for a 4-layer PCB substrate. With the constraint that the radii and trace widths of all the turns in a given coil set are identical and all windings comprise adjacent turns, there are 4 unique transformer configurations. These are detailed in Table 3.2 where P stands for a turn in the primary winding, and S the secondary winding. The difference between 1T-2T and 1T-2T (sub) has to do with the spacing of the primary relative to the secondary, as both are 1-turn primary, 2-turn secondary designs. In the first case, the primary winding is located on the top layer of the board, and the secondary turns sit directly underneath, straddling the PCB core, which is about twice as thick as the prepreg layers [4]. In the second case, the secondary turns straddle prepreg and the primary winding is across the PCB core.

[4]Multi-layer PCBs are constructed using combinations of stiff core material and flexible laminates. For instance, the 4-layer board used here has a 31 mil finished thickness, with a 14 mil FR-4 core at the center and 2 6-mil pre-impregnated epoxy laminate layers separating the top and route1 layers and route2 and bottom layers respectively.

Table 3.2: 4-Layer Transformer Winding Configurations

Layer (z-location)	1T-3T	1T-2T	1T-2T (sub)	2T-2T
1 (0)	P	P	N/A	P
2 (7.2 mil)	S	S	P	P
3 (21.9 mil)	S	S	S	S
4 (31 mil)	S	N/A	S	S

Figure 3.14 tells an interesting story regarding the performance of the various configurations. First, the 1T-3T designs have the best overall performance, defining the upper bound in efficiency for the 4-layer PCB transformers that meet the inductance matrix. It is also the most densely populated curve, having the largest number of solutions. For an inductance matrix where L_{22} was smaller, perhaps equal to L_{11} or only 2x as large, this would likely not be the case. Secondly, no 2T-2T designs could satisfy the inductance matrix. With 2-turns,

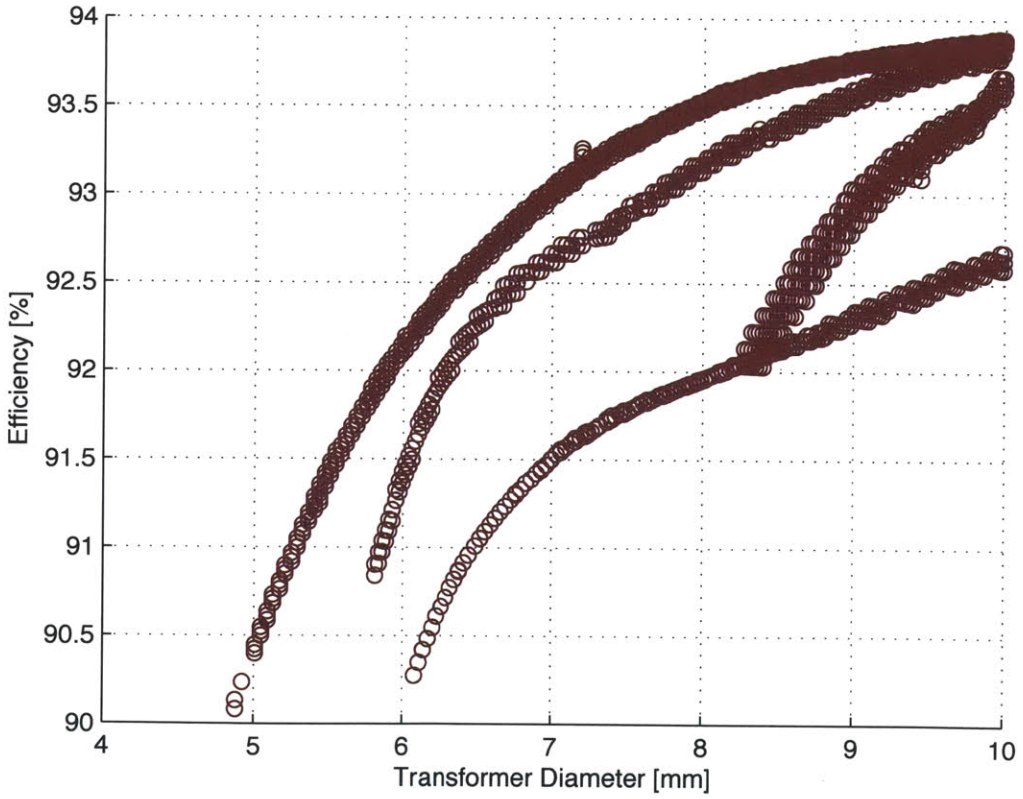


Figure 3.14: The efficiency-diameter map for the helical winding series showing the performance of the various winding configurations. The best overall results come from the 1T-3T curve, which bounds the upper efficiency limit.

the primary winding was too small and sufficient coupling with the secondary winding was not possible.

The middle set of curves comprises the 1T-2T case. Overall, it's capable of slightly lower performance than the 1T-3T case, most likely due to the better match a 3-turn secondary makes to L_{22}. The most surprising feature however, is the second tail of 1T-2T designs which has greatly reduced performance at a given diameter. This solution set was not expected, and originally thought to be the 2T-2T case. It arises when the radii of the primary and secondary invert. That is, for the "high-efficiency" 1T-2T designs, the radius of the primary is smaller than the secondary. The designs on the second tail all have primary windings with trace widths 3-4x larger than the other primary windings and radii larger than their respective secondaries. This larger trace width is key. It effectively acts as a flux shield, reducing the mutual coupling between the windings to the point where the design meets the required inductance parameters. This comes at the cost of efficiency owing to the resulting eddy currents.

The bottom-most curve comprises the 1T-2T (sub) designs, with the secondary coil wound across the thin prepreg layer and the primary winding coupling to it across the thicker FR-4 core. This case is interesting, because the secondary windings are somewhat smaller than the 1T-2T case, owing to the tighter coupling of their turns (which are more closely spaced). This actually enhances the prospects for coupling with the primary, since the radial change outstrips the z-axis change of moving the primary about 8 mils further from the secondary. The higher coupling was thought to offer a potential for improved performance. However, this turns out not to be the case. A similar situation arises to the second-tail 1T-2T designs. Working solutions either have a thin primary and a fat secondary (eg. design #9 in Figure 3.16a) or vice versa (design #16). In both cases, the thicker winding acts to squelch coupling so that the inductance matrix is satisfied at the cost of efficiency.

In typical transformer designs, more coupling is a good thing. The closer the coupling coefficient is to unity, the more ideal the transformer. Since we are looking to explicitly take advantage of the non-ideal characteristics of the transformer by using the magnetizing and leakage inductances to attain a desired converter tuning point, more is not always better. With a fully constrained inductance matrix, an increase in coupling due to winding diameter or spacing needs to compensated by a change in trace width or another parameter. In this case it turns out to be expensive from the perspective of efficiency. This suggests that there is an optimal substrate stack-up that will give the best overall transformer. Removing the constraint of a fixed z-axis spacing between turns is tantamount to adding another variable to the set of optimization parameters, greatly increasing the number of cases to be tested. Rather than allowing the z-locations to be completely free, instead transformer series were computed for a number of available PCB stack-ups. Six separate 4-layer stack-ups were

Table 3.3: 4-Layer PCB Stack-Ups

Finished Thickness	Layer 1	Layer 2	Layer 3	Layer 4
20 mil	0	6.15	12.50	17.20
31 mil	0	6.45	21.80	28.20
40 mil	0	8.15	30.45	38.65
62 mil	0	10.05	50.40	60.45
93 mil	0	15.65	76.00	91.65
125 mil	0	14.65	109.00	123.65

available from the board house ranging from 20 mil to 125 mil finished thicknesses. The details of each stack are identified in Table 3.3.

A transformer synthesis series was run for each stack-up, or roughly 10^4 numerical simulations. For each substrate, the best case designs were compared on a loss-diameter basis. In all cases that produced successful designs, the 1T-3T transformer provided the best overall performance. The 31 mil substrate achieves the highest efficiencies, and this was used to fabricate actual transformers. The other substrates have universally lower performance, and spacing on the 125 mil substrate precludes designs meeting the inductance targets.

3.3.6.2 Coils with Different Turn Parameters

Since shielding plays an important role in both meeting the final inductance parameters as well as determining transformer efficiency, a full search of the design space seemed prudent. This involves allowing the turns of each coil to take on any value of trace width and radius independent of the other turns in the same coil. This has the effect of enlarging the range for achievable self inductance and coupling with the possible result of improving the shielding-loss tradeoff.

In addition to allowing the turn parameters to vary independently, a set of 10 turn configurations were checked, where the turns of individual coils were allowed to be interleaved. For instance, previous 1T-3T transformers required the turns from each coil to be adjacent so that the primary was always on layer 1 and the remaining turns layers 2-4. Allowing interleaved coils adds the case where the primary is on layer 2 and the secondary comprises layers 1,3, and 4.

The total number of analytical cases evaluated was over 200k before being pared to about 27k transformer simulations. Automatic load balancing was employed across multiple workstations to evaluate many cases in tandem and reduce the total time to solution. The total

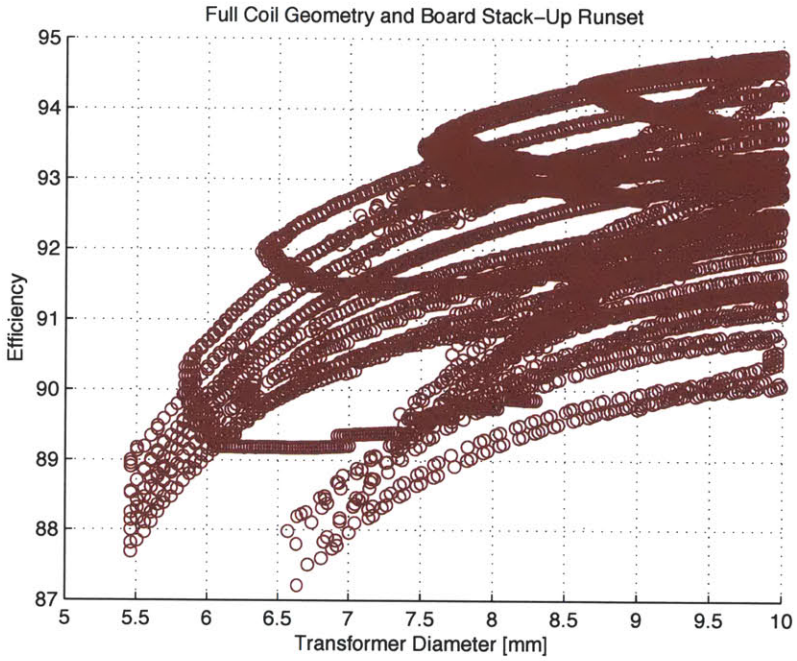


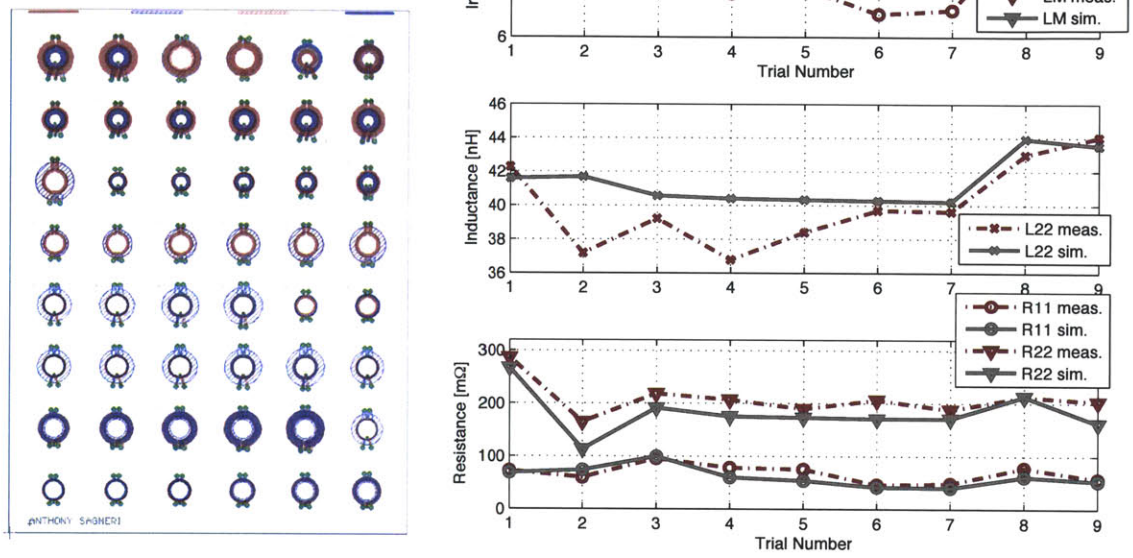
Figure 3.15: The efficiency-diameter map for the complete helical winding series allowing independent scaling of turn dimensions as well as coil interleaving.

compute time was approximately 56 days, single threaded. Load balancing across machines in combination with running multiple threads per machine reduced the actual compute time to about 2.5 days.

Figure 3.15 shows the full results of the calculation. The full sweep results in about a 1% improvement in efficiency as compared to the geometries evaluated in Section 3.3.6.1. This improvement is modest for an order of magnitude increase in compute time.

3.3.6.3 Helical Transformer Measurements

A round of transformers was selected and fabricated on the 31 mil 4-layer PCB process. Transformers were selected from both the constrained helix runs and those that allowed full variation. Figure 3.16a shows a CAD drawing of the fabricated transformer board. The measured results are compared against simulation in Figure 3.16b. Overall the match between the experimental and simulated values was improved significantly over the spiral winding case. In particular, the AC resistances track well over most of the range staying within about 15% of the simulated values, although the worst case error is 35% for trial number 2. The inductance match is not quite as good, with peak deviations of up to



(a) PCB tracks for helical transformers

(b) Measured results

Figure 3.16: Fabricated PCB and measured results for the helical winding transformers.

37% and a worse match over the range, but this is still better than the spiral case which demonstrated 41% deviation.

3.3.7 Improved Synthesis Algorithm

The above results are encouraging regarding the ability to synthesize transformers with controlled parasitics and having a known efficiency-volume tradeoff. However, the peak inductance errors for the helical winding transformers is 37%. This complicates converter design, requiring either iteration of the converter tuning point, the transformer geometry, or both. Part of the error in the design phase derives from the calculation of the winding self inductance independent of the other winding. This may be addressed by iterating the winding design after two coils are brought in proximity that are "almost right." Regardless, iteration with the numerical simulator does nothing to correct the error in the simulator

Table 3.4: Simulator Accuracy Comparison for 1T-3T transformer

Parameter	Measured Value	COMSOL	FastHenry
L_{11}	8.39 nH	8.84 nH	9.26 nH
L_{22}	39.22 nH	40.00 nH	40.56 nH
L_M	8.60 nH	8.82 nH	11.33 nH
R_{11}	95.8 mΩ	95 mΩ	90 mΩ
R_{22}	218.4 mΩ	215.65 mΩ	192 mΩ

itself. Improvements in that regard either require a more accurate simulator or more effective use of the existing one.

For more accurate simulation results, we explored a package known as COMSOL. It is capable of full-blown FEM simulations and brings with it the long solution timelines and resource intensiveness. However, when used in 2D axisymmetric mode, the solution time for the devices under consideration is roughly two minutes–a manageable amount in the context of the previous simulator work. In order to determine if more accurate simulations were possible using this package, a few transformers which were previously measured were evaluated. Figure 3.17 is a streamline plot of a 1T-3T transformer simulation where the secondary is supplied with a drive current. Table 3.4 lists the results of the inductance calculations for the same simulation. The FEM simulator, even in axisymmetric 2D mode provides results very close to the measurement values. In particular, COMSOL appears to do significantly better at estimating the coil self inductance.

A MATLAB script was written to control COMSOL and implement a transformer synthesis routine. The algorithm is presented in Figure 3.18. It utilizes the same analytical front-end to generate a locus of coil geometries that has approximately the correct inductance terms. The follow-on numerical simulation includes an additional step to compensate for skin and proximity effects that otherwise influence the final results. Each transformer geometry submitted to the numerical simulator first undergoes a simulation to find the inductance terms. If they are all within range, then the script moves on to the next geometry. Otherwise, a bisection algorithm is used to adjust the primary to meet L_{11} with the secondary in place. This usually only takes a few tries to converge. After L_{11} is established, the secondary radius is changed, via bisection, until L_{22} is achieved to within the preset tolerance of the desired value. At this point, the value of L_{11} is rechecked to ensure that the changes to the secondary have not pushed it out of range. If it is out of range, bisection begins again, this time with the last radius of L_{22} as determined by the numerical simulator. Next the secondary is adjusted and the values of L_{11} and L_{22} rechecked. The process is repeated until both are within range.

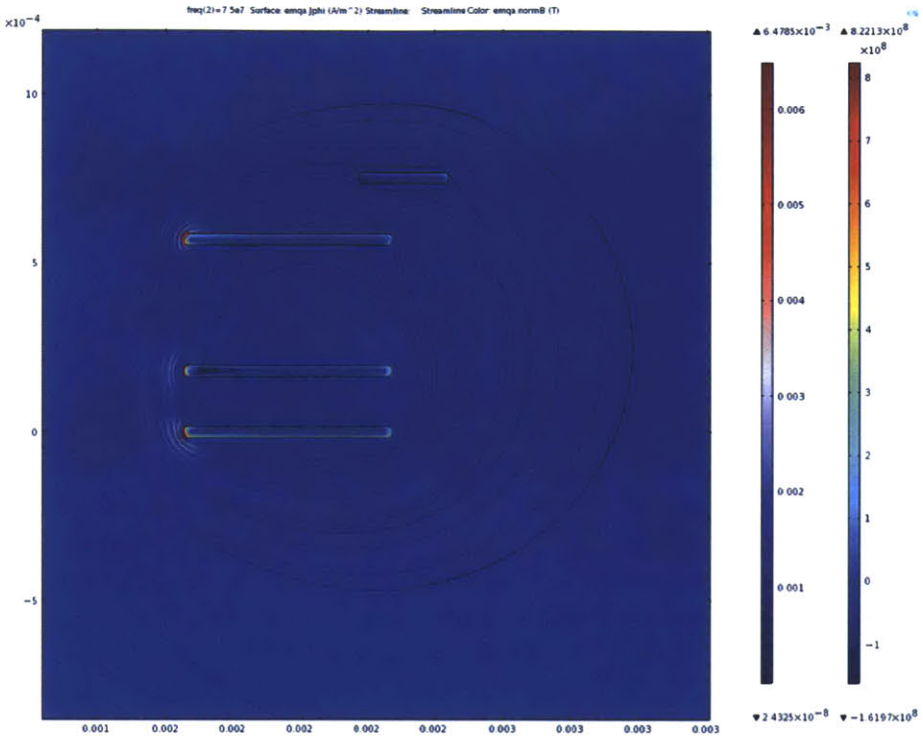


Figure 3.17: COMSOL simulation of a measured transformer geometry showing the magnetic field lines and current density in the coil cross sections when the secondary is driven in sinusoidal steady state at 75 MHz.

For a small number of cases, a limit cycle is reached whereby adjustment of the primary throws the secondary out of range and adjustment of the secondary does the same to the primary. This is dealt with by first reducing the radial step size. If it still fails to converter after an iteration limit is reached, the coil pair is rejected.

Once the primary and secondary coils have the correct self inductances while in proximity, the mutual inductance is evaluated. Coils pairs that have the their mutual inductance within the desired range of L_M are saved. This final set of cases represents the transformers that will provide the full inductance matrix. These are evaluated for efficiency by using the same time-domain simulation in conjunction with the target converter current waveforms. The total process for a run set that results in several hundred viable transformer designs takes about 18 hours when run in single threaded mode.

A set of transformers was fabricated and measured to compare simulated results to measurement. The results are plotted in Figure 3.19. The overall results are fairly good, with less than 15% error over the set of measured samples. A transformer was chosen from the mea-

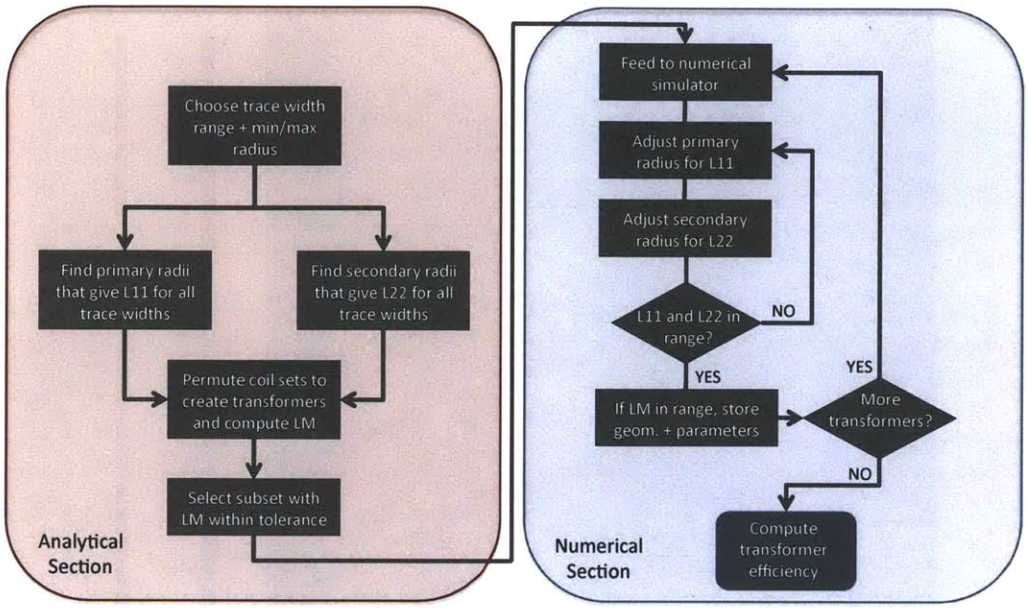


Figure 3.18: Flowchart showing algorithm including radius compensation

Table 3.5: Transformer #48 geometric parameters

Layer	Radius	Trace Width	Z-position
Primary	3.47 mm	0.93 mm	0 mm
Secondary 1	2.13 mm	0.60 mm	0.18 mm
Secondary 2	2.34 mm	0.93 mm	0. 56 mm
Secondary 3	2.96 mm	1.75 mm	0.79 mm

sured devices and used in the target isolated Φ_2 converter. The final electrical parameters and layout details are shown in Figure 3.20 and Tables 3.5-3.6.

During operation, the efficiency was extracted using a thermal model. Figure 3.22 shows the transformer in the running converter. Further details of the thermal model and extraction

Table 3.6: Transformer #48 electrical parameters

Parameter	Value
L_{11}	10.1 nH
L_{22}	44.0 nH
L_M	10.7 nH
R_{11}	56.7 mΩ
R_{22}	203.6 mΩ

process are described in Chapter **??**. The model indicates that the transformer is operating at about 94% efficiency, in agreement with the simulator results. Besides having relatively good efficiency, this transformer has very high power density, calculated as approximately 3.1 kW/in^3 where the volume includes air surrounding the transformer out one half of a radius on both sides of the device. The half-radius distance was established based on FEM simulations including copper sheets on either side of the transformer spaced 1/2 radius away for shielding purposes. At this distance, change in inductance and loss are small. The simulation results are plotted for the secondary winding in Figure 3.21

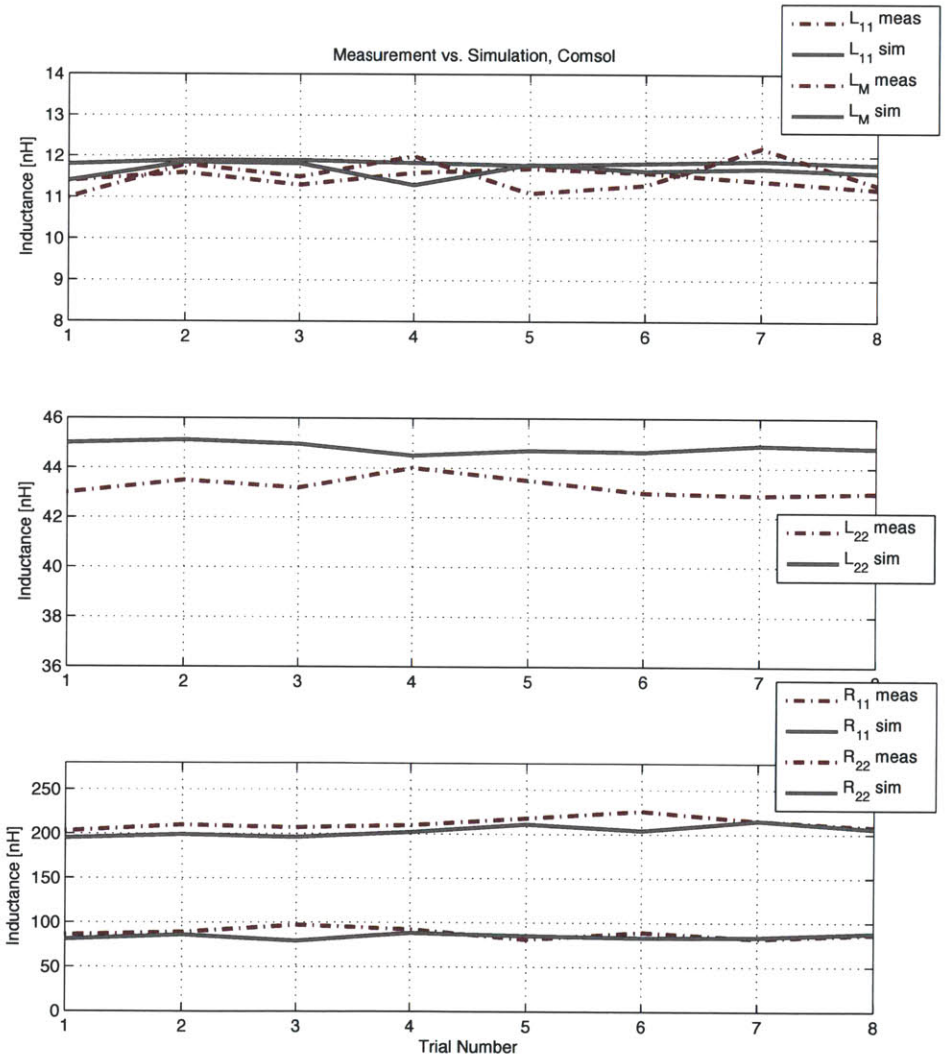


Figure 3.19: PCB measurements vs. COMSOL simulation for helical transformers

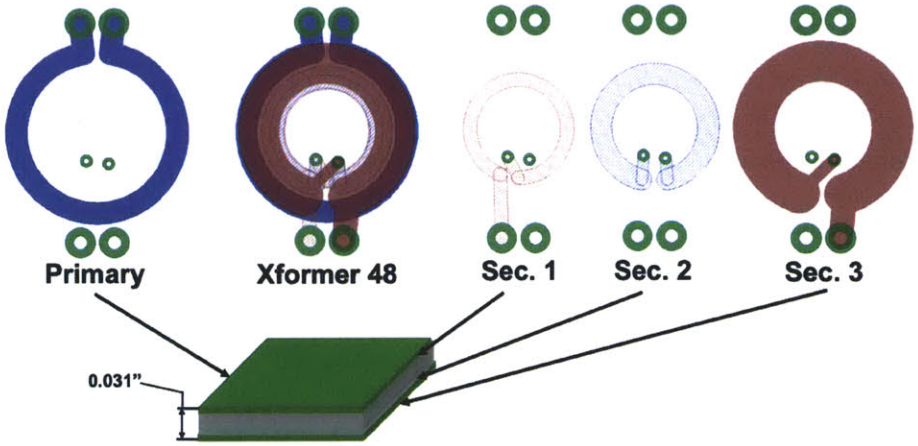
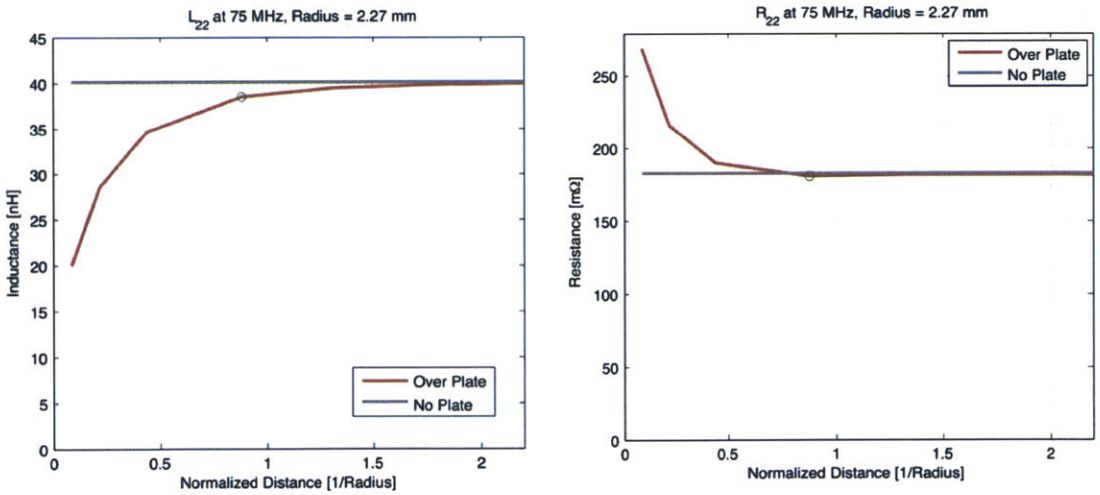


Figure 3.20: Transformer #48 layout details



(a) L_{22} vs. 1/plate distance

(b) R_{22} vs. 1/plate distance

Figure 3.21: COMSOL simulation of the transformer secondary winding between two plates shows that for a 1/2 radius spacing loss is barely affected and the change in inductance is small enough to be compensated by redesign.

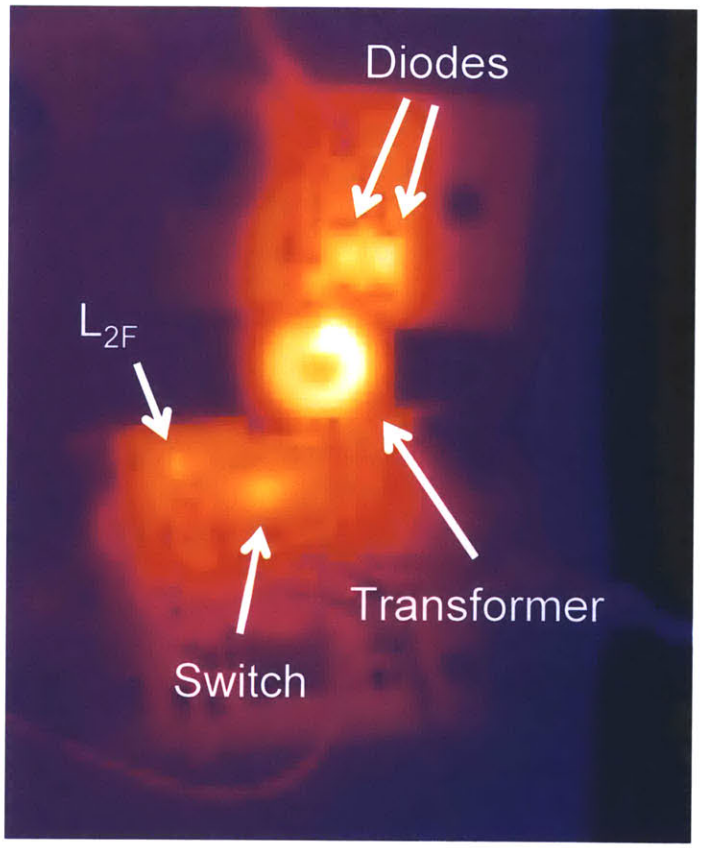


Figure 3.22: Thermal measurement of the transformer in operation at 94% efficiency in an Isolated Φ_2 Converter

Chapter 4

Isolated Φ_2 Converter

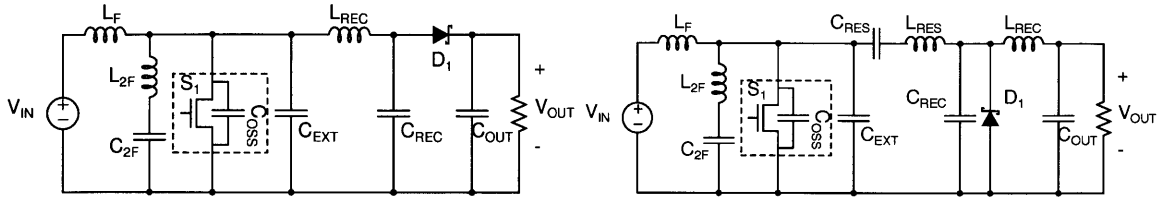
4.1 Background

The Φ_2 converter is one of a number of resonant converter topologies that have been successful in reaching VHF operation at high efficiency. In particular, these converters have achieved operation from a few volts and a few watts into the hundreds of volts and hundreds of watts ranges at frequencies up to 110 MHz and efficiencies into the low 90% range. With peak V_{DS} from 10s to 100s of volts, this combination of frequency, power, and efficiency highlights the benefits of fully-resonant converter topologies for achieving VHF operation [3, 6, 24, 50, 51].

The converters highlighted above are all various forms of non-isolated VHF designs. Many applications require isolation between the converter input and output terminals that would also benefit from small size, low cost, high bandwidth, or their various combinations. These include applications where galvanic isolation is desired for safety, as is often the case in medical power supplies. Other uses such as floating power supplies for gate drivers in large converters, instrumentation power supplies, and cell balancing in battery packs could all benefit from the reduced size, weight, and cost possible with a VHF converter. Another interesting extension that an isolated converter enables is stacking of the input and output terminals into series and/or parallel to meet voltage, current, and power requirements through a combination of many smaller cells. Such an architecture can enable VHF converters to use relatively low voltage switches and still maintain high performance.

4.2 Isolation Options

There are at least two means to achieve galvanic isolation between the input and output of a VHF converter cell. These include capacitive isolation and transformer isolation. Each has relative merits and tradeoffs we discuss below. Before their consideration, however, it is useful to examine the converter topologies of Figure 4.1. Depicted are a Φ_2 boost converter and an indirect Φ_2 converter. Both designs follow the general scheme of inverter, followed



(a) Φ_2 boost converter

(b) Φ_2 indirect Φ_2 converter

Figure 4.1: Φ_2 converters that exploit resonant networks and separation of energy storage and control to achieve high efficiency at 10s of volts and 10s of MHz

by transformation stage, followed by rectifier. The primary distinction between the two derives from the existence of a dc path between the input and the output of the boost converter. This means that some fraction of the output power is delivered directly from the input without suffering resonating losses. While this aids the converter's efficiency, the configuration of the rectifier precludes buck operation. On the other hand, the indirect Φ_2 converter has no dc path. It transfers all the power at ac and requires an additional blocking capacitor to sustain the dc offset between input and output, and enough impedance in the reactive link to permit the inverter and rectifier to operate properly. Typically, the required impedance comes in the form of an inductor which contributes to the loss budget and adds to the component count. As we will see, the isolated topology introduced below has the buck and boost abilities of the indirect Φ_2 converter, but lower component count than the boost-only.

The addition of a single capacitor to the converter in Figure 4.1b is a ready way to achieve isolation between the converter input and output. The approach is illustrated in Figure 4.2. It is essentially the indirect Φ_2 converter with a capacitor inserted in the rectifier return leg. Under the right circumstances, this is a very effective topology for isolation. While lower frequencies would impose the use of very large capacitors, in the VHF space the required capacitance can be as small as a surface-mount ceramic package. In addition, for VHF applications capacitive energy storage tends to be less lossy and more compact than the equivalent magnetic storage. Converters where the isolation voltage remains below a few hundred volts can benefit greatly from capacitive isolation.

One place where capacitive isolation becomes a questionable approach is for higher isolation voltages. For instance, off-line applications require kilovolt-scale isolation. Capacitors capable of standing off kilovolts are not generally designed for high frequency operation. As such Q [1] tends to be quite low. In addition, even for 10s or 100s of nanofarads, such X- and Y-rated isolation capacitors become bulky. Besides typically being larger than their

[1] Q in a capacitor is defined as $1/\omega RC$

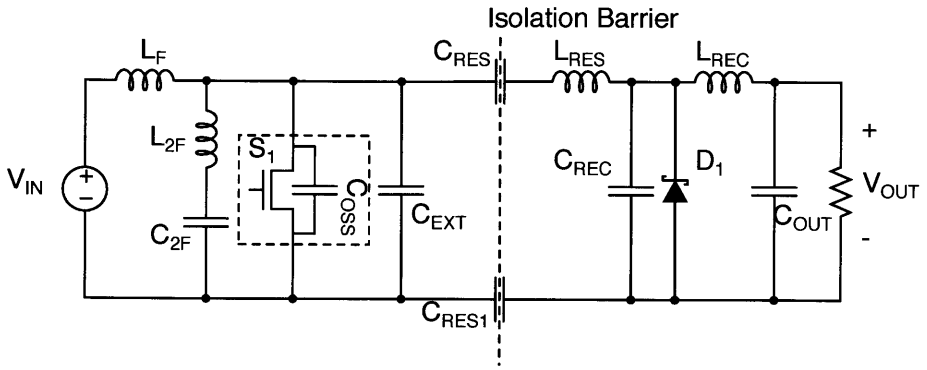


Figure 4.2: A capacitively isolated Φ_2 topology

magnetic counterparts, the additional uncontrolled inductance they introduce can stymie the implementation of a design. Another drawback of capacitive isolation in the indirect Φ_2 converter is that it requires an additional component. For applications where converters with stacked dc ports may be desired the component count becomes king, and issues like cost and routing traces for low inductance are dominant concerns that drive a strong desire for minimal component count. Finally, in applications where the floating output slews with a substantial dv/dt relative to the input, the resulting common mode currents can be unacceptable as compared to transformer isolation where the inter-winding capacitance can be much smaller with careful design, since the energy transfer occurs in the magnetic field rather than the electric field.

Magnetic isolation can solve many of the highlighted problems, albeit while introducing some of its own. The first salient characteristic of a transformer that matters for operation at VHF frequencies is parasitic inductance. By nature, transformers store some amount of energy in their magnetic field. This implies inductance. For reference, a standard two-winding transformer model is illustrated in Figure 4.3. The distribution of the energy storage relative to the ports, and the magnitude of the inductances is of interest. In a typical transformer operating below about 1 MHz, the approach is to make the transformer appear as ideal as possible by minimizing leakage inductance and maximizing magnetizing inductance (with some exceptions when energy storage in the core is desired). A high-permeability magnetic core allows for a large magnetizing inductance, keeping the magnetizing current small over a switching cycle. When combined with various winding and core structures intended to minimize leakage, high coupling coefficients are possible and the transformer simply acts as to transfer energy from the primary winding to the secondary winding while storing very little [2].

[2]Some converter topologies like the flyback deliberately exploit the magnetizing inductance as an energy storage medium, and use gapped core structures to bring L_μ to a more desirable value.

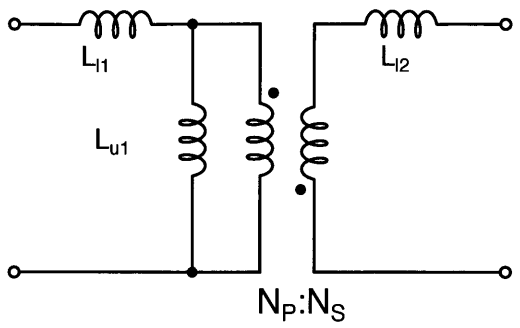


Figure 4.3: A two winding transformer model with magnetizing and leakage inductances.

Air core transformers that are feasible to use at VHF typically have lower coupling coefficients and relatively small magnetizing inductance. This puts the transformer parasitic inductances on par with the primary converter energy storage, typically 10s or 100s of nanohenries for VHF designs, squarely in a range that must be addressed. One way to deal with the extra inductance is to account for it in the balance of the converter design. However, a potentially better solution is to use the transformer parasitic inductance to replace the other magnetic energy storage in the converter. Either way, for a VHF design to be feasible the transformer parasitics must be well controlled, imposing a requirement to synthesize a desired set of inductance parameters in a repeatable fashion. This topic is addressed in Chapter 3.

4.3 Transformer-Isolated Φ_2

There are many possible ways to insert transformer isolation into the Φ_2 converter topology. The configuration adopted here is similar to the Class-E implementation in [?, 16]. In addition to creating an isolation barrier, we desire to to reduce the component count by absorbing as much of the existing magnetic energy storage as possible. Starting with the Φ_2 boost topology of Figure 4.1a and adding a transformer allows easy realization of a transformer-isolated Φ_2 converter. One obvious place to insert an isolation barrier is between the inverter and the rectifier. Figure 4.4a shows that by doing so, L_{REC} ends up in series with L_{l2}, which can replace its energy storage function. This leaves the total number of magnetic storage components constant, but requires the addition of a dc-blocking capacitor on the primary side to avoid a dc short.

The next step is to realize that in the vicinity of the switching frequency, the input is a short because nearly all of the ac current in L_F shunts through the input capacitance (or the ideal source as depicted in the schematic) which is deliberately made large enough to have very small ripple. Therefore all the ac voltage that appears across the drain-source port of

the switch also appears across L_F with opposite phase per KVL. If the primary winding terminals are connected from the top of V_{IN} to the drain terminal in lieu of L_F, then L_{11} can stand as its replacement (see Figure 4.4b). The winding sense must be reversed to maintain the correct polarity on the secondary side, as indicated by the dots in the schematic. It should be noted that this placement of the transformer avoids the need for a dc blocking capacitor as the input voltage will simply appear across the switch in periodic steady state. Finally, letting L_{l1} go to zero in the two-winding model and establishing a unity turns ratio makes the isolated Φ_2 converter of Figure 4.4c a one-for-one analog to the Φ_2 boost converter from a tuning point perspective. There is one significant change: there is no dc power transfer and this penalizes the efficiency by increasing the circulating current for a given output power level [3]. Figure 4.4 shows the various stages of the transformation.

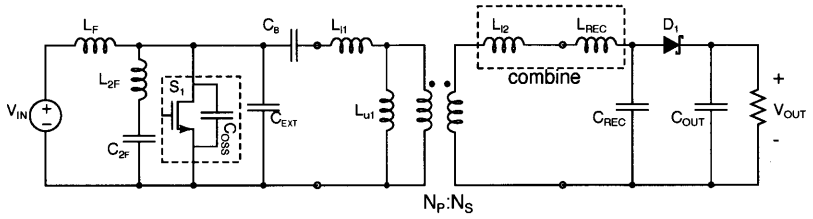
When comparing component count, relative to the Φ_2 boost converter, inserting the transformer absorbs L_F and L_{REC} for a net savings of one component. However, it exchanges two discrete components that must be sourced, handled, and mounted for traces in a PCB, which are generally cheaper and give more reproducible impedance characteristics. When compared to the indirect Φ_2 converter, it fares better still. The transformer-based Φ_2 allows output voltages above and below the input voltage giving it the functionality of the indirect Φ_2 topology. Since the transformer provides isolation, C_{RES} is not necessary. The further absorption of L_{RES}, L_F and L_{REC} means that three components are eliminated from the system, in total.

Besides the benefit of a galvanically isolated output, having a transformer isolation barrier that absorbs the converter energy storage has significant benefit to the overall converter size as compared to implementing a transformer and separate tuning inductors. The integrated transformer structure utilizes the same area on the board (or volume around the board) maximizing the opportunity for flux linkage. With the transformer and tuning inductors implemented as separate elements, flux linkage is precluded, or at least reduced, and inductance for a given total volume will diminish. Figure 4.5 shows a transformer design that meets the requirements of $L_{11} = 11.8$ nH, $L_{22} = 47$ nH, and $L_M = 11.8$ nH, next to two transformer-inductor pairs that meet the same inductance parameters with nearly the same efficiency. The transformer that is paired with the inductors has been modified to provide the required L_{11} and coupling while reducing L_{22} so that the sum of the secondary leakage and the external inductor provides the desired value. In the first pair, labeled B, the transformer is paired with a discrete air core solenoidal "spring" inductor from Coilcraft [4]. The

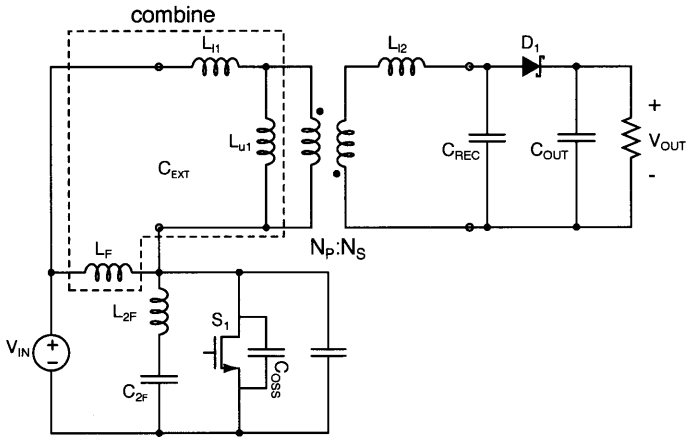
[3]Note, there is no reason that we couldn't have started with a two-winding model that has all the leakage reflected on the secondary side of the transformer. However, it is simply more convenient to have the somewhat more physical model when thinking about the implications that coil geometry has on leakage and coupling.

[4]Coilcraft part number B09T-LC

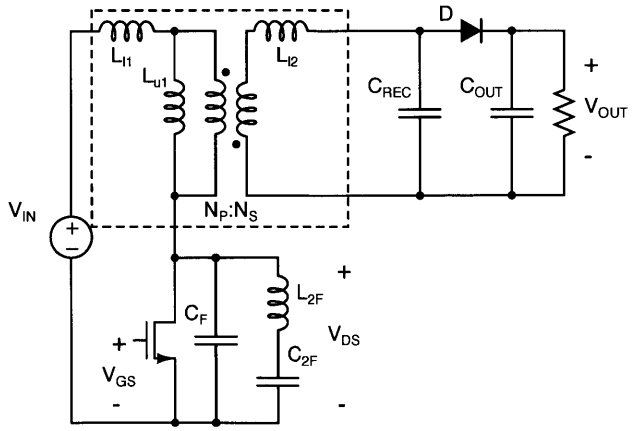
spring inductor is conservatively 4x the volume of it's PCB footprint, so the total volume occupied by the set is about 3x the single transformer case. For case C, the inductor is implemented as a planar toroid in a board twice as thick as the transformer board of A. The overall system has about 1.9x the size of case A. For reference, the PCB transformers in A and B share an identical 31-mil 4-layer stackup: 1 oz. copper, a 14-mil core, and 6-mil



(a) Φ_2 boost converter with transformer absorbing L_{REC}



(b) Transformer repositioned and absorbing L_F



(c) Isolated Φ_2 topology

Figure 4.4: Transformation from Φ_2 boost converter to isolated Φ_2 converter

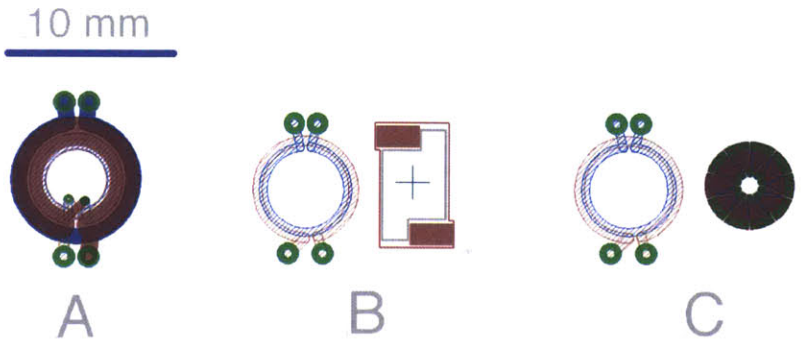


Figure 4.5: Volume comparison for a single transformer that meets the inductance parameters and two transformer-inductor pairs that do the same at comparable efficiency. Case B uses a discrete solenoidal inductor that has roughly 4x the volume for the same board area as the planar components. The toroidal inductor in Case C, provided by George Hwang, is planar in the PCB. Case A is smaller volume-wise by about a factor of 3 over Case B, and 1.5 over Case C.

prepreg on each side of the core defining the 1-2 and 3-4 layers. The toroidal inductor of C has a 62-mil 2-layer stackup: 1 oz copper, and a 59-mil core. Therefore it occupies roughly twice the volume of a transformer of the same area.

4.3.1 Tuning

Tuning an isolated Φ_2 converter can be accomplished in similar fashion to that of other Φ_2 converter systems. The general approach is to split the system into the inverter and the rectifier. Then the rectifier is tuned to appear resistive at the fundamental. The value of that resistance is used as the load resistor in tuning the inverter. Since almost all the power is delivered from inverter to rectifier at the fundamental, when the two pieces are combined, the operating power of the converter will be close to that observed in the individual case. A full description of the tuning procedure is provided in [24, 28]. To facilitate a follow-on discussion an overview of the tuning procedure is provided below.

Starting with the rectifier, the basic operating parameters are chosen (V_{IN}, V_{OUT}, P_{OUT}). For the isolated Φ_2 case the rectifier topology in question is a series-loaded resonant rectifier as depicted in Figure 4.6. For tuning purposes, the rectifier output is loaded with an ideal voltage source in SPICE whose value is set equal to V_{OUT}. The input is driven by a sinusoidal source at the fundamental with an amplitude equal to $4V_{IN}/\pi$. In this rectifier, the primary handles for tuning are the values of L_{REC} and C_{REC}. To make tuning more intuitive, these are parameterized in terms of a center frequency, $f_C = \frac{1}{2\pi\sqrt{L_{REC}C_{REC}}}$ and the characteristic impedance, $Z_0 = \sqrt{L_{REC}/C_{REC}}$. By sweeping Z_0, output power can

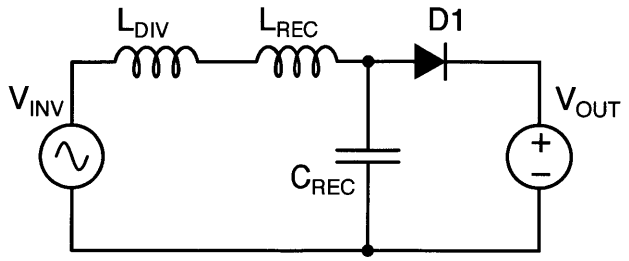


Figure 4.6: The resonant rectifier pictured is used for tuning purposes. The actual rectifier inductances are part of the transformer structure.

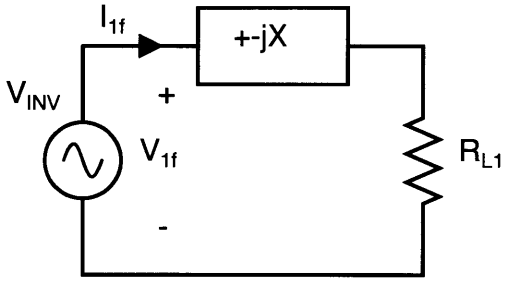


Figure 4.7: Power is controlled by adding a reactance in series with the resonant rectifier. In order to calculate the desired value, the inverter is approximated as a sine wave with voltage equal to $4V_{IN}/\pi$. The resistor R_{L1} is the describing function resistance of the rectifier derived from simulation.

be controlled and by sweeping f_C, the rectifier input impedance can be made to appear resistive at the fundamental or to include a reactive component. The input impedance in this case is the describing impedance, which is the impedance found by looking only at the fundamental voltage and fundamental current and ignoring harmonics. Once the procedure is carried out, the describing resistance of the rectifier is found, and this may be used in the inverter design. In some cases, it may desirable to first add some series reactance.

Series reactance in the rectifier leg is useful to set the converter output power. The reactance forms an impedance divider with the describing resistance of the rectifier. By assuming that the inverter voltage approximates a square wave, the power at the fundamental delivered through the impedance divider to the describing resistance can be explicitly calculated. Figure 4.7 captures the scenario. Since both the inverter voltage and rectifier resistance are known, choosing the desired output power allows direct calculation of the series reactance. The series reactance is typically an inductor because an inductively tuned rectifier aids the creation of a ZVS event and has less circulating current than the capacitive case, which aides efficiency. The value of the inductor can be calculated with the following impedance divider relationship:

$$L_{DIV} = \frac{\sqrt{\left(\frac{4V_{IN}}{2\pi\sqrt{P_{OUT}R_{L1}}}^2\right) - 1}}{2\pi f_S} \tag{4.1}$$

Once the load network is defined it is used in designing the Φ_2 inverter where the load is the L_{DIV}-R_{L1} pair. The inverter is tuned to operate as near as possible to ZVS mode over the desired input voltage range. The inverter tuning procedure starts by choosing the value of a fictitious shunt capacitance, C_{PAR} such that the characteristic impedance at the drain-source port of the inverter switch is not swamped by the load impedance. The implied constraint is that C_{PAR} must be at least equal to the nominal device capacitance [5]. The resulting value of C_{PAR} is used to calculate the values of L_F, L_{2F}, and C_{2F} according to the equations below:

$$L_F = \frac{1}{9C_{PAR}\pi^2 f_S^2} \tag{4.2}$$

$$L_{2F} = \frac{1}{15C_{PAR}\pi^2 f_S^2} \tag{4.3}$$

$$C_{2F} = \frac{15C_{PAR}}{16} \tag{4.4}$$

At this point, additional shunt capacitance is added in parallel to the switch and L_F is adjusted in concert to achieve a phase angle at the fundamental of at least 30° and an amplitude ratio of approximately -3 dB between the 3rd harmonic and the fundamental. The result is an inverter, that operates in ZVS mode, as deired. When it is married to the rectifier it will deliver the desired output power subject to some additional tweaking that typically takes the form of small adjustments to the series reactance, L_{DIV} used to set the output power.

Many tuning points achieve both ZVS operation and the desired output power at a single operating point. They are necessarily different, however, when efficiency and operating range are considered. While the Φ_2 boost converter has achieved efficiencies into the 88% range, the isolated Φ_2, with an AC-only path is a more challenging design from that perspective. This motivates careful choice of the tuning point.

[5]The nominal device capacitance is typically C_{OSS} measured at $V_{DS} = V_{IN}$

Table 4.1: Component Values for Example Isolated Converter Designs

Component	Value		
	11.4-W Design	15.3-W Design	8-W Design
$L_{\mu 1}$	38.2 nH	6.5 nH	13.5 nH
L_{2F}	34.3 nH	9.8 nH	20 nH
C_{2F}	32.8 pF	115 pF	56.3 pF
C_{EXT}	70 pF	207 pF	35 pF
L_{l2}	22.3 nH	17 nH	32.7 nH
C_{REC}	45 pF	0 pF	0 pF
Diode	2x On-Semi SS16, 60V, 1A	2x SS16	1 x SS16
Switch	MITMV1 (97 pF @ 12 V)	MITMV1 97 pF @ 12 V	MITMV1 97 pF @ 12 V

The impedance divider in Figure 4.7 can be designed to deliver the same output power for many values of L_{DIV} and R_{L1}. Equally as many phase and amplitude relationships between the fundamental voltage and current at the input port of the network obtain. If output power is held constant while R_{L1} scales, the amplitude and phase of the current must change relative to input voltage. As circulating current scales while output power remains constant efficiency will suffer. This suggests that one means of improving efficiency is to tune the rectifier to be purely resistive, and then design the inverter to drive the resulting resistive load.

This is illustrated by three converter designs in the isolated Φ_2 topology. The input and output voltage for these designs were fixed at 12 V and the frequency at 75 MHz. The component values for each design are detailed in Table 4.1. The first design is an initial attempt to establish converter performance using the MITMV1 switch discussed in Chapter 2. The rectifier has a relatively low equivalent resistance of 3Ω and substantial series reactance. The second design uses a retuned rectifier with a resistive input impedance. The third is a rectifier using only one diode, which allows for a larger input resistance than in the second case. The second and third designs were deliberate attempts to minimize circulating current for a given output power by tuning the rectifier to look resistive with it's maximum resistance value. The maximum value of the describing resistance occurs when all the rectifier capacitance is provided by the diode parasitic capacitance. In the 15.3-W case, two parallel diodes were used allowing a maximum R_{L1} of 6Ω. The 8-W case uses only a single diode, and as a result the equivalent resistance can be higher, approximately 12Ω.

Table 4.2 provides the simulated performance of these designs (SPICE) simulations can be found in Appendix A.3) which were obtained when transformer Q was assumed to be 100 and the Q of the second-harmonic inductor, L_{2F}, 80. First, it's important to note that the voltage and frequency were the fixed operating point parameters. Designed output power necessarily varies when C_{REC} comprises entirely diode parasitic capacitance. This is because there is only one tuning point for the rectifier that simultaneously meets this requirement and is purely resistive at the fundamental. Nevertheless, looking at both the efficiency,

Table 4.2: Simulated Loss Breakdown and Converter Efficiency for Isolated Converters

Loss Element	11.4W Design		15.3W Design		8W Design	
	Power (W)	Normalized	Power (W)	Normalized	Power (W)	Normalized
Switch	1.6	10.5 %	2.2	11.6 %	0.9	9.4 %
Diode	0.5	3.4 %	0.7	3.6 %	0.4	3.7 %
Transformer	1.4	9.0 %	0.8	4.0 %	0.4	3.9 %
L_{2F}	0.4	2.9 %	0.2	1.0 %	0.1	1.0 %
P_{IN}	15.3		19.2		9.8	
P_{OUT}	11.4		15.3		8	
Drain Efficiency	74.2 %		79.8 %		81.9 %	

and loss normalized to output power tells the story. The 11.4-W design, which includes substantial series reactance, posts an efficiency of 74%. Both of the designs with resistively tuned rectifiers have efficiency numbers near the low 80% range. Closer examination of the normalized component loss reveals the difference. The bulk of the improvement comes from reduced transformer losses owing to lower circulating current. The normalized loss in the 11.4-W case is 9%, while in the resistively tuned designs it's closer to 4%. A similar reduction occurs to the loss in L_{2F}, which also benefits from lower circulating currents. The 6% efficiency improvement is significant on the scale of converter performance, but it poses a challenge over another design constraint–input voltage range.

The target input voltage range of the converter design is 8-16 V. Beyond efficiency considerations, the input voltage range is set on the high end by the peak permissible drain voltage and on the low side by the ability of the rectifier to function properly (below some cutoff voltage, the diode never conducts and the rectifier delivers no power). Unfortunately, a purely resistive rectifier design, as in the 15.3-W and 8-W designs of Table 4.2, has a narrow input voltage range. The plots of Figure 4.8 serve to illustrate the case. The top plot shows converter output power versus input voltage. The 11.4-W benchmark design operates over the entire input voltage range of 8-16 V. At the low end of the range it is capable of operating all the way down to V_{IN}=4 V. Efficiency stays relatively constant over this range, as well. The plot of the resistively tuned designs show much narrower operating range. Both have rectifiers that cut out below 11 V and the steeper rise in drain voltage versus V_{IN} also restricts the upper limit to about 15.7 V. This input voltage range is substantially smaller than desired. On the other hand, the efficiency is higher over almost the entire operating range.

A reasonable tradeoff is possible by adding some series reactance to the rectifier. This will expand the input voltage operating range while trading only a small amount of efficiency. To demonstrate this two additional converters were designed in SPICE for the same power levels as the 15.3-W and 8-W designs. The new component values from the designs are listed in Table 4.3. In the 15.3-W case, the R_{L1} is dropped from 6Ω to 5Ω by adding shunt capacitance and adjusting L_{REC} to re-establish resistive tuning. Power was adjusted back

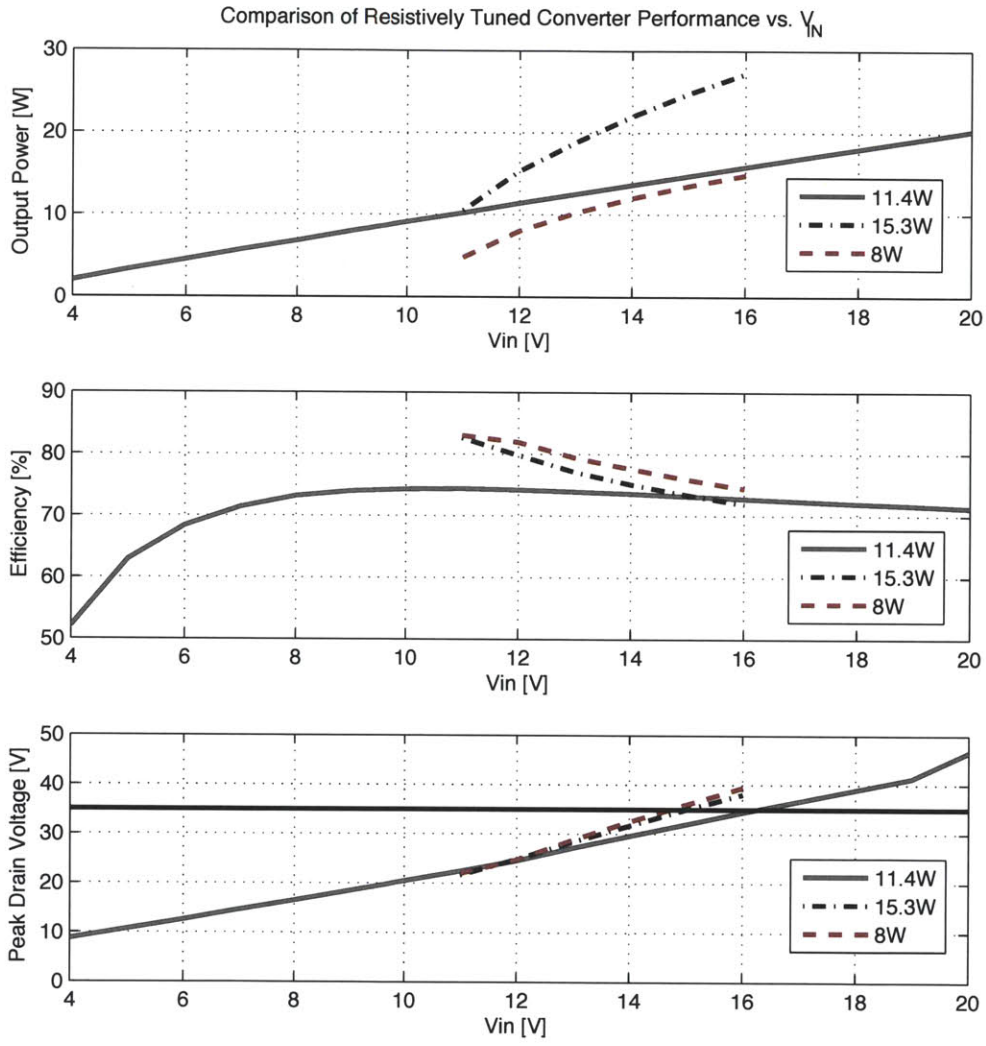


Figure 4.8: Tuning the rectifier to have a purely resistive input impedance at the fundamental (15.3-W and 8-W designs) increases efficiency but dramatically curtails the input voltage range. The horizontal black line in the bottom plot is the peak allowed V_{DS} of the inverter power device.

Table 4.3: Component Values for Designs with Series Reactance

Component	Value	
	15.3-W Design	8-W Design
$L_{\mu 1}$	38.2 nH	11.8 nH
L_{2F}	18.5 nH	15.0 nH
C_{2F}	61 pF	75 pF
C_{EXT}	189 pF	122 pF
L_{l2}	22 nH + 8 nH	25.2 nH + 10 nH
C_{REC}	14 pF	15 pF
Diode	2x On-Semi SS16, 60V, 1A	1x SS16
Switch	MITMV1	MITMV1

to 15.3-W by placing an additional 7.7 nH in series with L_{REC}. For the 8-W case, R_{L1} was lowered from 12Ω to 9Ω and 10 nH of series inductance was added to adjust power. The plots in Figure 4.9 compare the original 11.4-W design with the two new designs having rectifiers including some series reactance. The plots show that the input voltage range is greatly increased while efficiency improves, as well. The full 8-16 V range is possible and a 5% increase in drain efficiency is realized at the nominal input voltage of 12 V. The converter simulations can be found in Appendix A.3.

4.3.2 Initial Converter Design and Experimental Results

An experimental converter design was completed using the procedures outlined above. The basic characteristics and component values are included in Table 4.4. The converter was designed for 8.5 W output power, nominal input and output voltages of 12 V, and a nominal efficiency of 77%. The transformer used in the converter design derives from the techniques presented in Chapter 3. It is a two-winding structure with a 1-turn primary and 3-turn secondary. For simulation purposes, it was treated as a 1:1 transformer with L_{μ}=3.4 nH and L_{l2}=30.6 nH.

The basic converter waveforms as simulated are presented in Figure 4.10 (simulation files are available in appendix A.3). The device parasitic lead inductances are modeled in the simulation, which introduces the extra ringing visible in the drain and gate waveforms. The bottom plot compares the fundamental of the rectifier voltage and current. The current has been scaled up by a factor of 5 to make it easy to see the phase relationship between the two waveforms. Per the tuning discussion, the converter has been tuned to have a slightly inductive rectifier, with the current lagging the voltage by about 21.6°. This provides for a good input voltage range without greatly sacrificing efficiency.

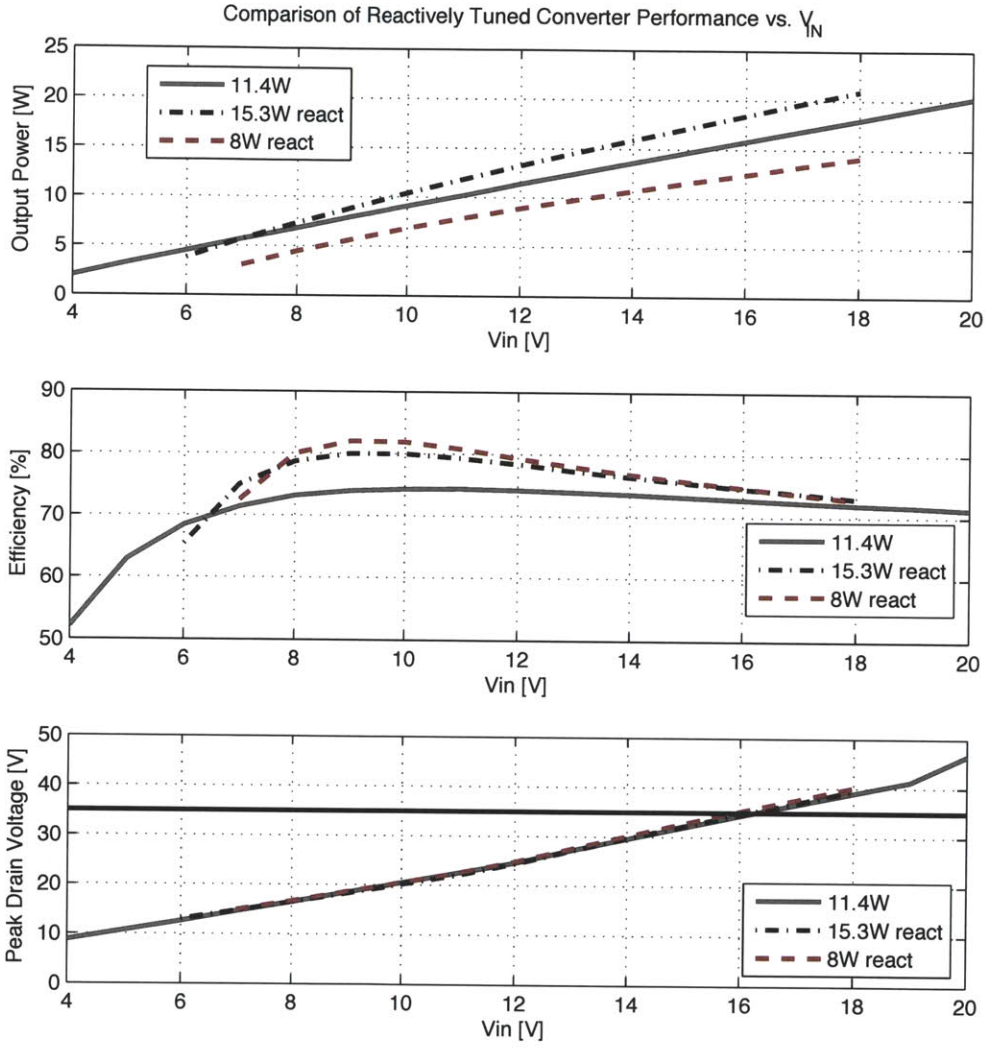


Figure 4.9: Retuning the converters to have a small inductance in series with the rectifier offers wide load range and substantial improvement in efficiency over the original converter (11.4-W case) design. The horizontal black line in the bottom plot is the peak allowable V_{DS} of the inverter power device.

Table 4.4: Component Values and Specifications for First Prototype Isolated Φ_2 Converter

Component/Spec	Value
L_{2F}	10.6 nH
C_{2F}	106 pF
C_{EXT}	220 pF
C_{REC}	50 pF
L_{μ}	8.4 nH
L_{l2}	30.6 nH
N	1:1
Diode	Vishay 10MQ100NP6F 100V, 2.1A Schottky
Switch	MITMV1(97 pF @ 12 V)
$V_{IN-nom.}$	12
$V_{OUT-nom.}$	12
$P_{OUT-nom.}$	8.5 W
η (designed)	77 %
f_{SW}	75 MHz

The converter was constructed using standard PCB techniques and then run through a series of measurements to assess performance. Figure 4.11 shows a picture of the as-tested converter along with a plot of the measured V_{DS}. The converter efficiency and output power at the nominal operating point was measured. Both parameters posted substantially lower, with converter output power at roughly 4 W, and efficiency hovering around 37%. Subsequent close examination of the drain-voltage waveform provided an initial clue as to the discrepancy. As compared to the simulated waveform plots, the drain peaks are offset. The particular shape difference can result, among other factors, from excess inductance in the input loop. Careful measurement of the PCB using an Agilent 4395A in impedance analyzer mode revealed an excess parasitic inductance of approximately 3 nH around the input loop. That is, the loop formed by the transformer primary, the power device, and the capacitors that form the input ac short. Figure 4.12 highlights the loop where this inductance appears. Subsequent simulation in SPICE shows that the addition of this inductance accounts for most of the output power drop, as the simulation power drops from 8.5 W to about 4.5 W.

The reason that this parasitic is so significant in this topology is clear from Figures 4.13a and 4.13b. In the first figure, an equivalent Φ_2 inverter network is created by replacing the rectifier with it's describing resistance, R_{L1}, adding a dc-blocking capacitor, and referring the impedances to the primary side of the transformer. This Φ_2 inverter network will transfer the same power at ac as the dc-dc converter at the nominal operating point. The salient characteristic of this equivalent drawing is the location of the load network (formed by L_{L2}, R_{L1}, and C_B in this case), versus the typical Φ_2 inverter. The network is part of the input mesh, rather than forming another mesh with the power device, as it is usually drawn.

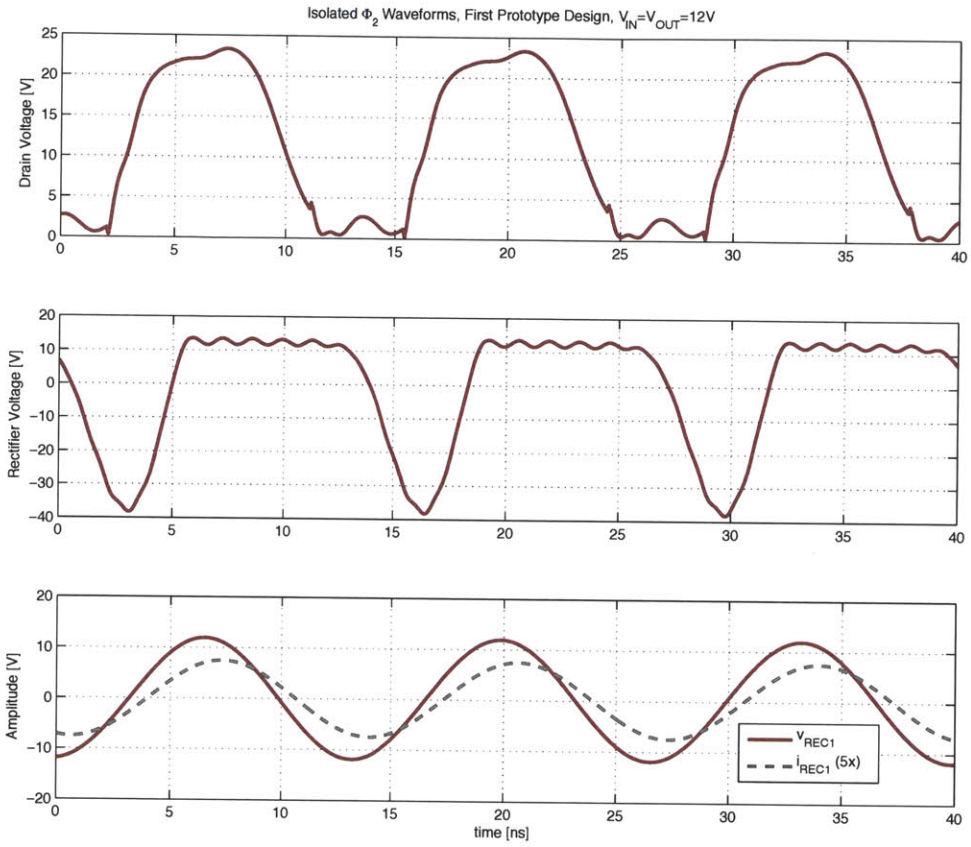
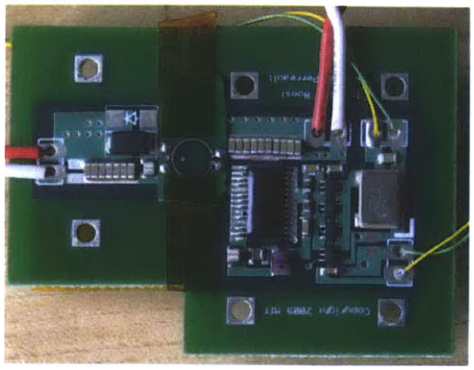


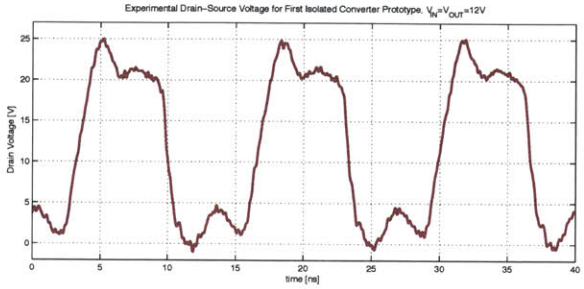
Figure 4.10: Simulated waveforms from the initial isolated Φ_2 converter prototype. The bottommost plot shows that the rectifier is tuned slightly inductive per the tuning discussion in the previous section.

When the input-loop parasitic inductance is added, as in Figure 4.13b, the significance is manifest. L_{PAR} forms an impedance divider with the load network. As a result, the output power is lowered because only a fraction of the drain-source voltage actually drives the load. This hurts efficiency as well as power because the circulating currents in the device parasitic elements don't necessarily decrease. For instance, gate loss remains constant and to the extent that the drain-source voltage waveform is nearly the same, so does displacement loss. In contrast, Φ_2 networks with the power take-off across the drain-source terminals are not nearly as sensitive.

Since L_{PAR} is set fundamentally by the sizes of the various components and their ability to be co-located, it is not easy to eliminate. As a result several methods of compensating for this shortcoming were considered (see Figure 4.14). The first method considered was



(a) First isolated Φ_2 prototype



(b) Experimental drain waveforms

Figure 4.11: Experimental isolated Φ_2 converter and the measured drain waveforms. This converter suffers from input-loop parasitic inductance and diode losses that diminish output power and efficiency to 4 W and 37%, respectively.

inductive peaking. The capacitor C_{PEAK} is added to the load network to resonate with the net inductance and offset the effect of L_{PAR}. Simulation showed that to raise the output power to the design level required a large peak factor, around 5 in this converter design. The large peak interferes with the tuning point impedance required for proper Φ_2 operation and while the load power can be raised, proper ZVS is not achieved. Another option is to use the network of Figure 4.14b. This circuit rearranges the Φ_2 network to put an inductor in series with the input loop, while still providing isolation. However, the result is that the 2^{nd} harmonic zero becomes a function of all the component values. This makes tuning very difficult since a critical part of the tuning characteristic is the 2^{nd} harmonic short, which is now sensitive to any component variation, as opposed to just L_{2F} and C_{2F}.

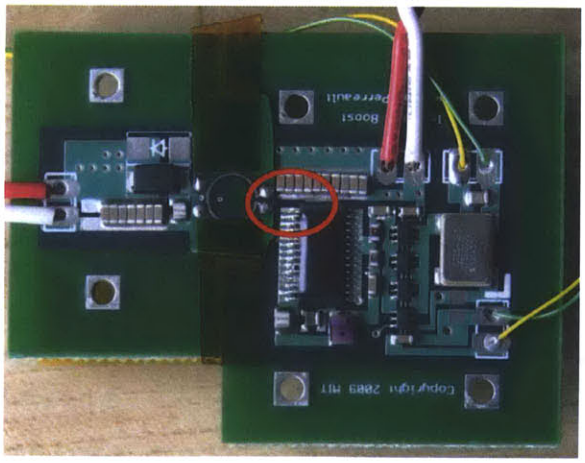
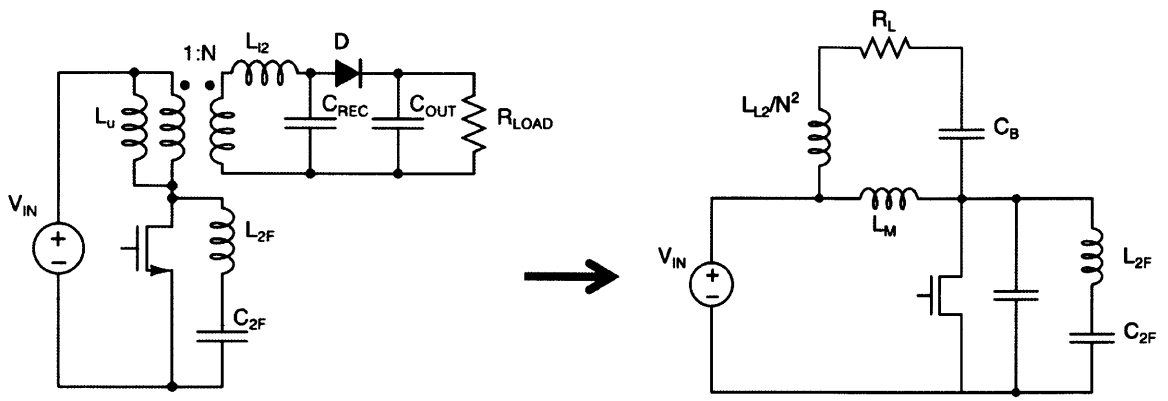
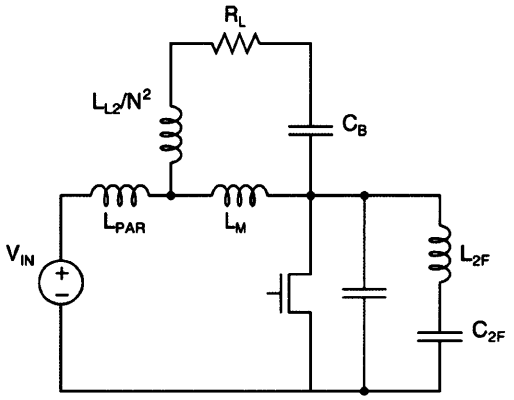


Figure 4.12: Experimental isolated Φ_2 converter. The red ring indicates the approximate current path that leads to input loop parasitic inductance.



(a) Translation to equivalent Φ_2 inverter network

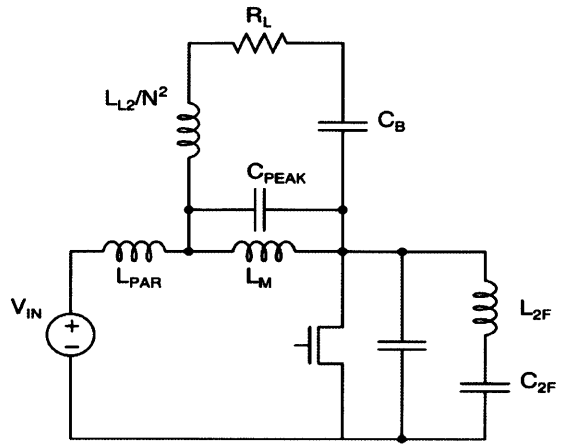


(b) L_{PAR} appears in the input mesh

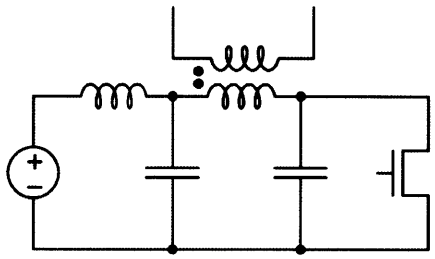
Figure 4.13: Translating the isolated dc-dc converter into an equivalent inverter network helps to illustrate the effect of input parasitic inductance on this topology. L_{PAR} effectively forms an impedance divider with the load network, negatively impacting output power and efficiency.

The solution in Figure 4.14c makes the input ac short resonant with L_{PAR} to eliminate its voltage divider effect in the circuit. This technique was used to successfully raise the output power in an experimental converter. However, it requires the addition of a filter inductance because C_{RES} necessarily has significant ac voltage during operation. That said, the required inductance L_{FILT} was only 50 nH, when the input voltage ripple characteristics were held to the same specification as the original design. For the experimental test, C_{PAR} was set equal to 1.5 nH, which offsets the 3 nH parasitic at 75 MHz and 50 nH of filter inductance was added, as well.

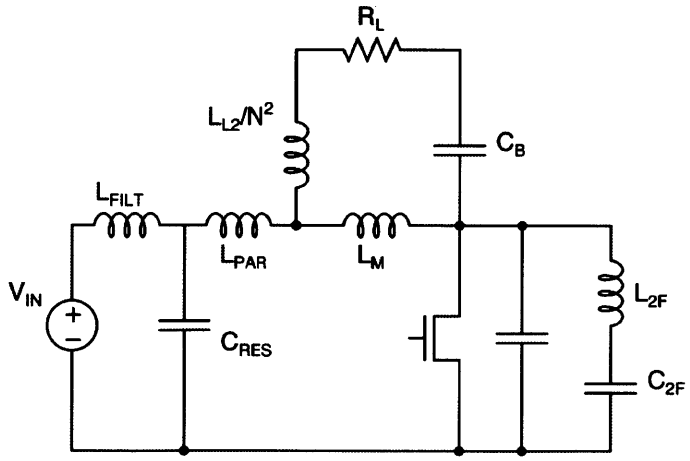
By resonating out the parasitic inductance, the converter output power was increased from 4 W to 8.3 W. Efficiency doubled, going from 31% to 62%. The close match in output



(a) Use C_{PEAK} to resonate load net inductance



(b) Network has input-mesh inductance



(c) Resonate C_{RES} with L_{PAR}

Figure 4.14: Various potential solutions to mitigate the effect of the input-loop parasitic inductance.

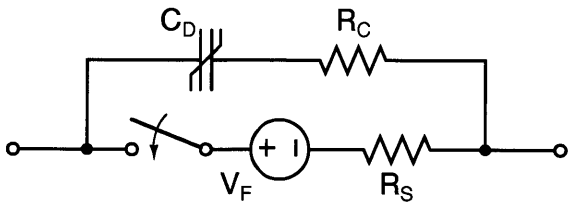


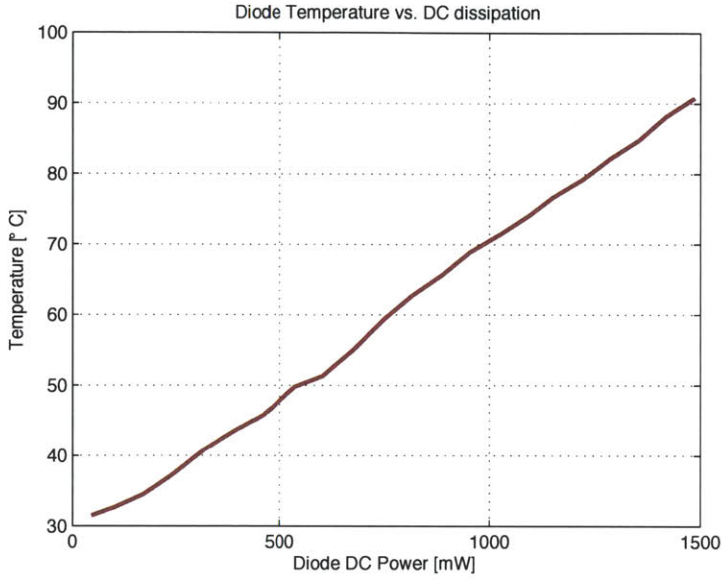
Figure 4.15: Large-signal diode model used for VHF simulations

power between simulation and experiment confirms that L_{PAR} is the primary mechanism by which output power was curtailed in the original design. While some inductance around the input loop is unavoidable, it is possible to redesign the transformer to accommodate. Since the primary-side leakage inductance is effectively in series with L_{PAR}, by increasing the coupling coefficient, the same power transfer can be achieved despite the presence of loop inductance at the input because L_{PAR} is offset by a reduction in the transformer's primary-side leakage. The efficiency discrepancy is another matter, however. While there is some penalty for resonating the input parasitic and C_{RES}, the experimental efficiency was nearly 15% lower than simulation predicted, suggesting the negative influence of another mechanism. This is addressed in the next section.

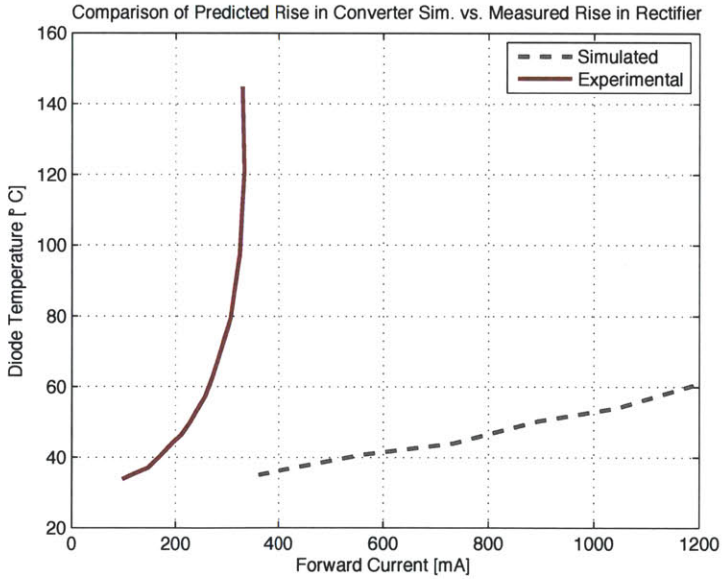
4.3.3 Power Schottky Diode Performance at VHF

The large discrepancy in converter efficiency between simulation and experiment in Section 4.3.2 led to an investigation into the loss mechanism. Basic thermal measurements were accomplished by running the converter with thermocouples attached to the semiconductors and the transformer. A much larger than expected temperature rise was observed in the rectifier diode.

Modeling of schottky diodes for simulation in VHF converters is similar to the techniques used for LDMOS transistors. That is, the diode capacitance is measured using an impedance analyzer (in this case the Agilent 4395a) as reverse bias voltage is swept from zero to the desired peak operating voltage. Since the measurements include the complex impedance, rather than just the reactance, the equivalent series resistance is extracted simultaneously. Forward conduction characteristics are modeled in the usual way, by plotting the diode I-V curve and extracting the series resistance and the other parameters of the diode equation. The latter are ignored in favor of a simpler model. The complete model used for VHF is depicted in Figure 4.15. This model was used successfully for many converter designs. However, in the case of the first isolated Φ_2 prototype, it failed to predict the diode losses accurately.



(a) In situ diode temperature



(b) Experimental vs. measured temperature

Figure 4.16: Diode case temperature vs. input power is characterized and used to generate the expected diode temperature rise during converter operation. Actual measurement shows a very steep increase over model predictions for the pulsating current waveforms in the resonant rectifier.

Directly measuring voltage and current at VHF to determine loss is difficult. Current probes either lack the requisite bandwidth, or introduce too much parasitic inductance, disturbing the circuit operation. As a result, thermal characterization techniques were adopted. The first step characterized the diode case temperature as a function of an applied DC bias. A diode was mounted on a converter board exactly as it would be during operation. A DC bias was provided to dump a known quantity of energy into the diode while the temperature was simultaneously measured via thermocouple. Since DC signals are comparatively easy to measure, this provided an accurate measurement of the diode dissipation as a function of temperature. This result can then be used to assess the diode loss during converter operation by measuring the case temperature in situ. Figure 4.16a shows the case temperature as a function of DC input power. It rises from room temperature to about 90° C as the power ramps to 1500 mW. This temperature rise is largely consistent with the information available in the data sheet.

This DC temperature-power transfer curve was used to plot the predicted diode case temperature in the running converter as a function of it's average forward current. When the prediction is compared with the measured case temperature the results are striking. The actual diode temperature rises extremely rapidly, exceeding the model predictions by greater than 4x. This result is plotted in Figure 4.16b. To determine whether this was the result of unexpected converter behavior or the effect of operating under AC drive, a stand-alone resonant rectifier was designed and driven by a power amplifier. The rectifier was precisely tuned to match a SPICE simulation in the impedance domain (see Figure 4.17), a technique that has provided an excellent match in power and efficiency as well as transient performance for other VHF converter systems. Overlaid simulated and experimental waveforms in the same plot show that the transient match is very good. However, when the diode temperature is plotted versus average forward current under DC and AC drive conditions (75 Mhz) the same steep temperature rise was observed (Figure 4.18), yielding a clear discrepancy in efficiency.

In an attempt to capture the increased loss in simulation, the VHF diode model parameters were adjusted. It was not possible to simultaneously satisfy both loss and transient behavior in this way, however. The diode manufacturer's models were also checked in SPICE and neither loss nor transient behavior was accurately predicted. The fundamental mechanism causing the increased dissipation under the AC drive condition was not identified. However, V_{REC} was recorded as the AC drive was increased successively. Overlaid plots reveal that the peak forward voltage of the diode during operation rises substantially as the power transferred through the rectifier increases. Figure 4.19 is the plot of V_{REC} overlaid for increasing forward power. The heavy black horizontal line indicates the output voltage of the rectifier. The sloping heavy black line approximates the local average voltage of the diode. Under high AC drive conditions, the forward voltage reaches at least 3 V. According to

simulation, forward current doesn't exceed 1.8 A, at which point the forward drop should be less than 900 mV worst case, and 700 mV at temperature based on the manufacturer's data sheets. This discrepancy reveals non-quasistatic behavior that contributes to increased loss. When the diode forward drop variation is accounted for, the simulated and experimental converter loss are in much closer agreement.

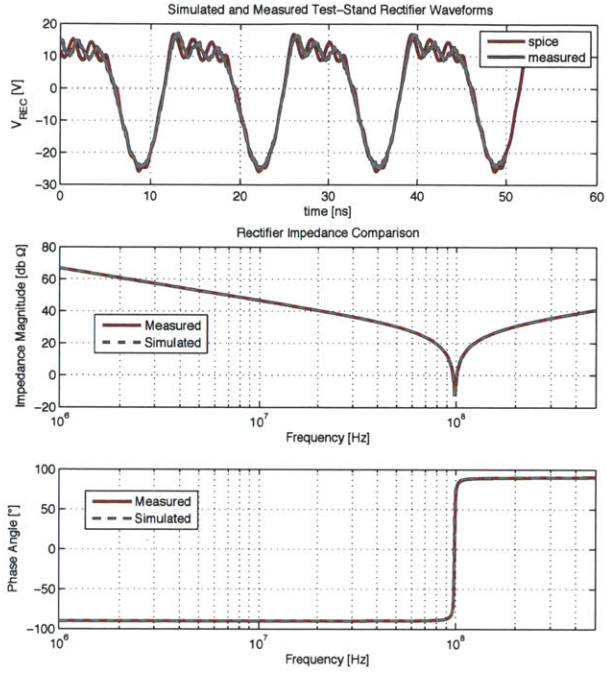
To improve rectifier performance in the isolated Φ_2 converter, a number of diodes were evaluated in various combinations. An automated test stand was set up to perform the evaluation. Figure 4.20 shows the test-stand schematic. A signal generator and power amplifier provide AC drive to the input of a resonant rectifier built with the devices under test. The output of the rectifier is held at a constant DC value (12 V, the isolated converter designed output voltage) by an electronic load. Peak reverse voltage, DC output power, and diode temperature are measured and recorded by the system at regular intervals. Each test cycle starts at a minimum AC drive condition and ramps slowly until either the peak drive voltage is exceeded or the maximum permitted temperature is exceeded. After each increase in AC drive, the system is allowed to achieve thermal equilibrium before the next jump in drive power.

The first set of characterization curves use the Vishay 10MQ100 schottky diode in three different rectifiers - a single diode version, and 2- and 3-paralleled diode versions. Figure 4.21a shows the results. The three curves show that paralleling the diodes reduces the temperature rise at a given operating power, but even with three diodes the rectifier is efficiency is not high enough to meet the desired converter performance [6]. Table 4.5 shows the various combinations of diodes that were tested. In Figure 4.21b more diodes are compared. The several 100-V diodes show similarly poor characteristics, with the 100V ON-Semiconductor version performing best. A 60-V ON-Semi device was tested as well, and faired significantly better in both single and parallel configurations. Figure 4.22 shows the 2-diode ON-Semi rectifier versus the performance of the model used for the converter design. When the rectifier loss is extracted for 8 W operation, the 2-Diode rectifier dissipates about 200 mW more power than the model, which corresponds to a 2.5% efficiency drop. This was the best set of diodes tested and was used for a subsequent isolated converter design.

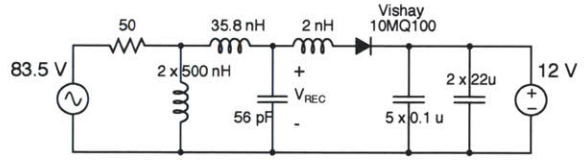
4.3.4 Second Converter Design and Experimental Results

After identifying the issues hampering the efficiency of the first prototype, namely inductance around the input-side mesh and excessive losses in the schottky diode, a second

[6] The non-monotonicity of the curves arises from the temperature-power coupling of the diode and rectifier. When the AC drive amplitude is first stepped, there is an immediate increase in output power. As the diode temperature rises, power falls until thermal equilibrium is reached.



(a) Impedance and transient data



(b) Rectifier simulated in SPICE

Figure 4.17: Tuned test rectifier transient and impedance simulation and measurements.

Table 4.5: Diodes and Configurations Evaluated for Thermal Performance

Part Number	Voltage, Current	Configuration
Vishay 10MQ100	100V, 2.1A	1 diode
Same	Same	2 diodes in parallel
Same	Same	3 diodes in parallel
ON MBRA1H100	100V, 1A	1 diode
Fairchild S100	100V, 1A	1 diode
ST STPS1H100A	100V, 1A	1 diode
On SS16	60V, 1A	1 diode
Same	Same	2 diodes in parallel

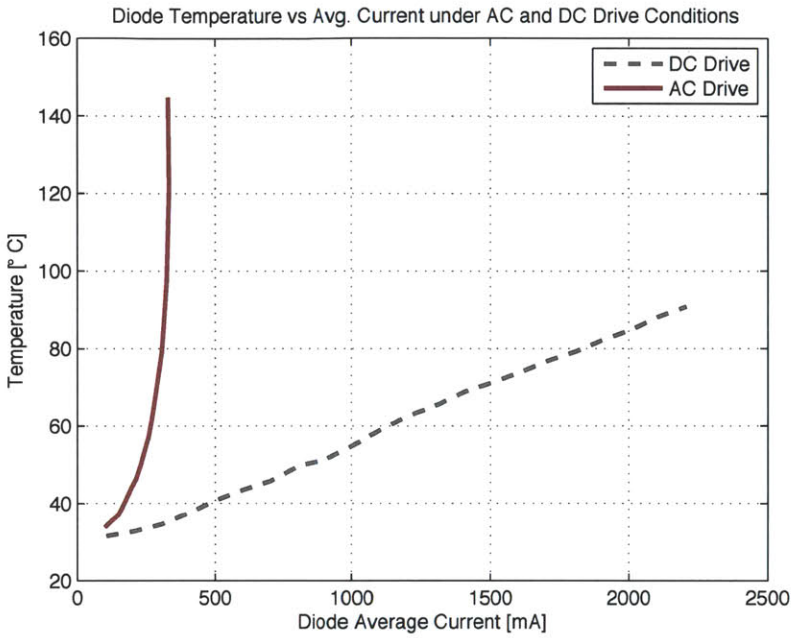


Figure 4.18: Diode temperature rise under AC and DC drive conditions.

prototype design was undertaken. The second converter design accounts for the limitations discussed above by choice of operating point and redesign of the transformer. Operating power was constrained to below 8 W because above this point the rectifier performance suffers owing to the diode characteristics. Similarly, the transformer was redesigned to the new operating point while accounting for approximately 3 nH of parasitic inductance in the input-side mesh. The converter design parameters are detailed in Table 4.6.

The final tuning point was chosen by iterating design values and measured values in the impedance domain. This allows characterization of the PCB and individual components to ensure the best possible match between simulation and experiment. The tuning point was measured in two separate parts. The rectifier and inverter were split, with the transformer attached to the inverter side. One impedance measurement looked directly into the rectifier port. The other looks into the transformer secondary back towards the inverter. After a few iterations, very good agreement between the simulated and experimental impedance measurements was achieved, as is illustrated in Figure 4.23.

In order to gain a better understanding of the power loss distribution in the converter and a better match between simulation and experiment, a thermal model of the converter was created. The model starts by assuming a linear relationship between the power dissipated in a given element, and its temperature rise and the temperature changes of the surrounding components. In this case, an R-matrix reflecting the coupling from component

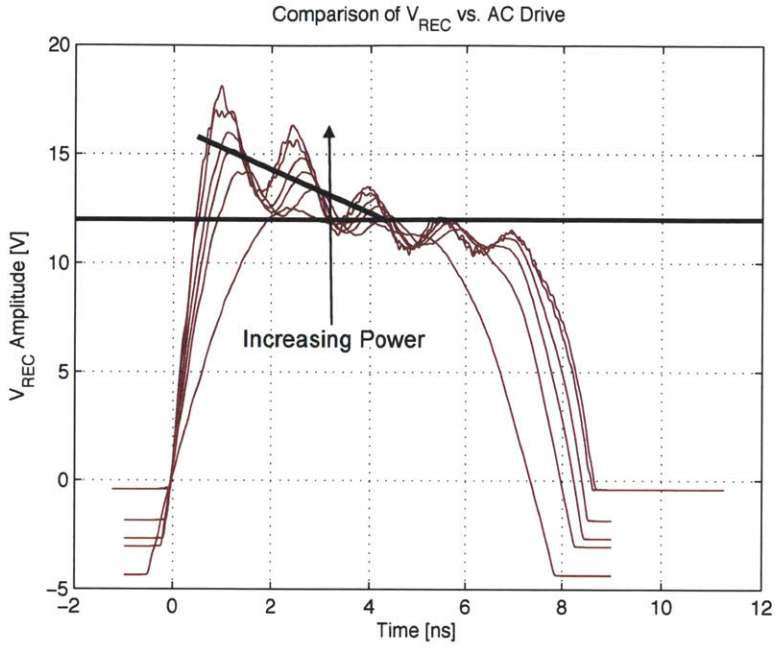


Figure 4.19: As AC drive is increased, the peak forward voltage drop rises as well, dramatically increasing diode loss. V_{REC} is the voltage labeled in Figure 2.17b, effectively the diode forward drop plus 12 V.

to component is easily constructed. Once the R-matrix is known, taking the inverse and measuring the component temperatures during converter operation yields the power loss in each component. The system of equations is simply represented by:

$$\mathbf{T = RP}$$

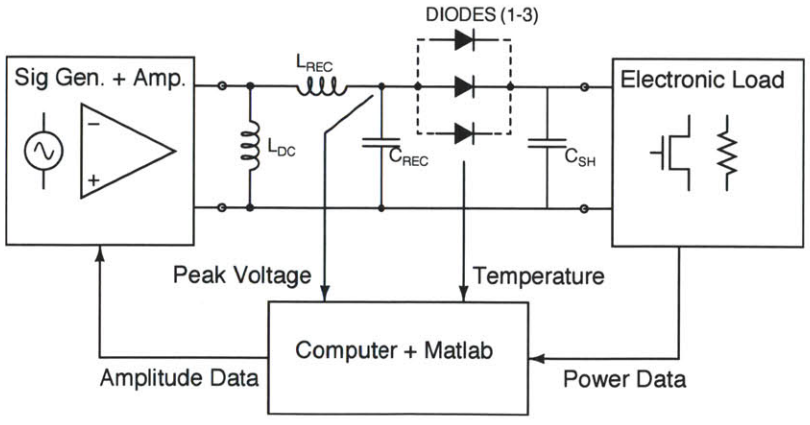
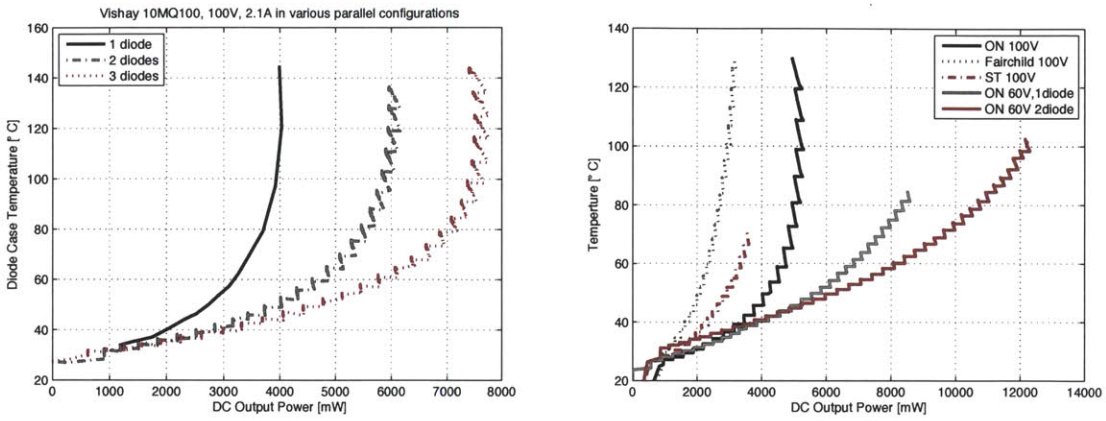


Figure 4.20: Schematic of diode test setup



(a) Vishay 100V diodes

(b) Various diodes compared

Figure 4.21: Performance of various diodes as power is modulated and rectifier configuration is changed

Where T is the vector of component temperatures, R is the thermal resistance matrix, and P is the power dissipation in each component.

The primary sources of power loss in the converter are the MOSFET, the diode, the transformer, and L_{2F}. A thermal camera was used to characterize the temperature rise of each of these components as a DC bias was applied to the component to simulate dissipation.

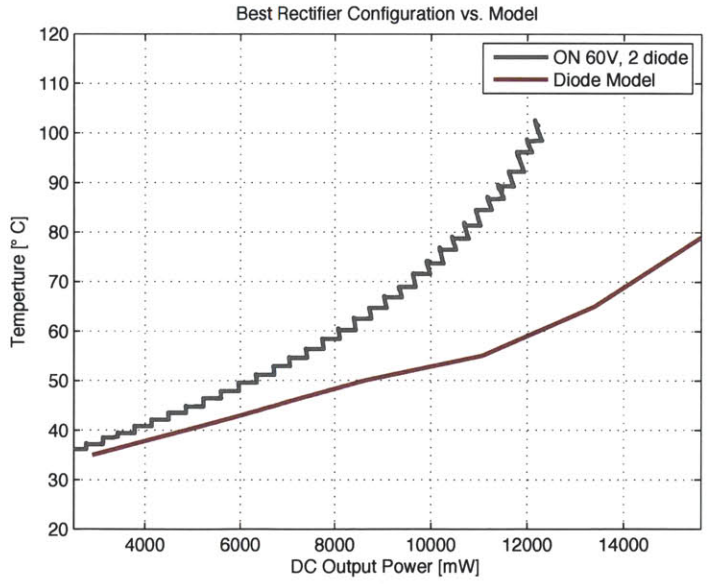
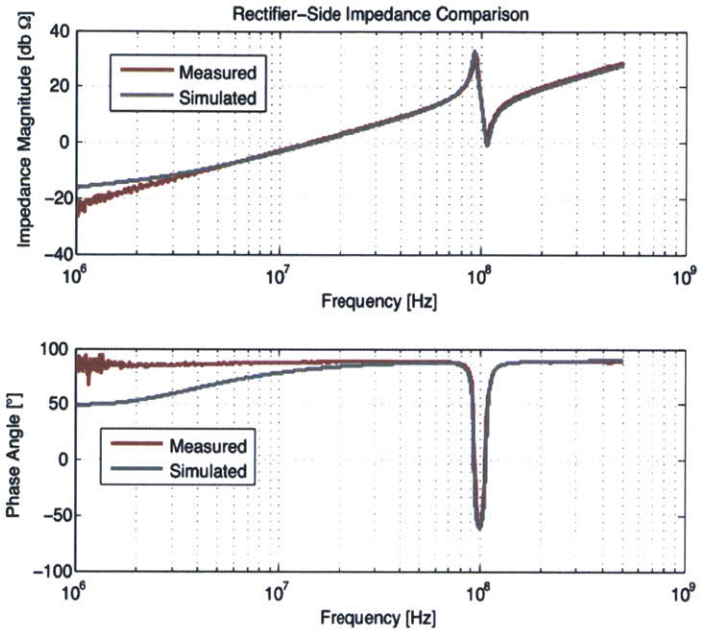
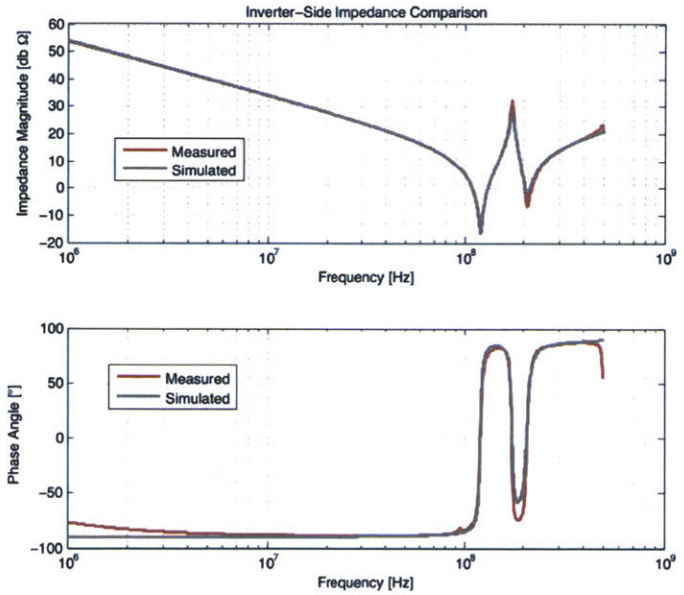


Figure 4.22: Final rectifier design versus original simulation.



(a) Impedance looking into the rectifier



(b) Impedance looking into the transformer secondary

Figure 4.23: The converter impedance was simulated in SPICE and then measured experimentally to ensure proper tuning. Iteration allowed for very close agreement.

Table 4.6: Component Values and Specifications for Second Prototype Isolated Φ_2 Converter

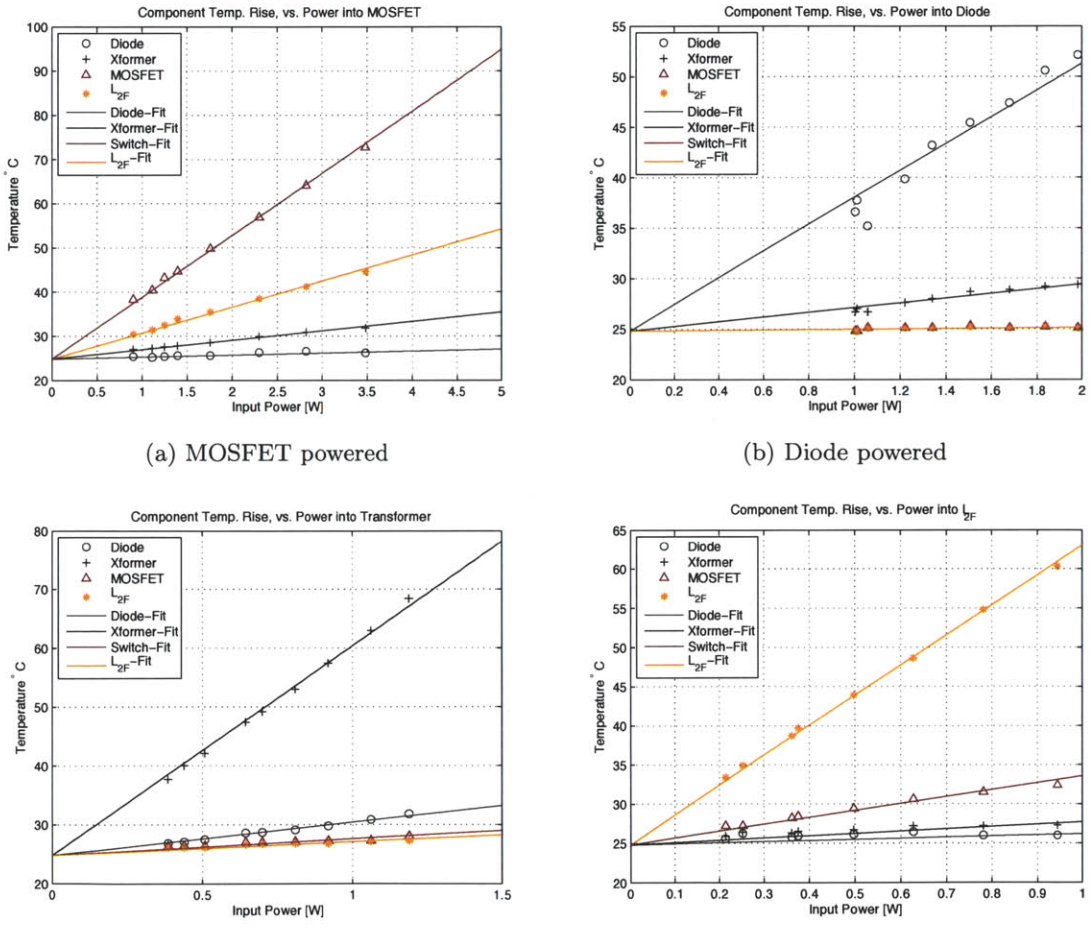
Component/Spec	Value
L_{2F}	12 nH
C_{2F}	74.8 pF
C_{EXT}	97 pF
C_{REC}	8.2 pF
L_{μ}	3.61 nH
L_{l1}	6.52 nH
L_{l2}	11.53 nH
N	1:3
Diode	2x ON SS16, 60V, 1A
Switch	MITMV2 (108 pF @ 12V)
$V_{IN-nom.}$	12
$V_{OUT-nom.}$	12
$P_{OUT-nom.}$	7.5 W
η (designed)	75 %
f_{SW}	75 MHz

For instance, in the case of the MOSFET small gauge wires were attached to the drain and source terminals and a current applied. The dc input power was measured and the temperature rise of the MOSFET, Diode, transformer, and L_{2F} were also measured. To check for linearity, the process was repeated for several values of input power. This provides the on-diagonal term in the resistance matrix for the MOSFET as well as coupling resistances to the other components. By repeating the procedure for the diode, transformer, and L_{2F}, the entire resistance matrix was populated. The R-matrix values are listed below for the second isolated Φ_2 prototype.

$$\mathbf{R} = \begin{bmatrix} 13.8 & 5.9 & 0.4 & 1.3 \\ 2.3 & 36.7 & 2.0 & 2.6 \\ 0.2 & 2.7 & 13.8 & 8.0 \\ 0.2 & 2.1 & 0.7 & 37.6 \end{bmatrix}$$

Figure 4.24 shows the plots of the temperature data and their curve fits as each device is successively swept over a range of drive powers. From the plots, it's clear that the behavior is quite linear over the range of interest. As a result, fitting to linear curves works well and the simple thermal resistance model is valid. Once the R-matrix was constructed, the inverse was calculated. The condition number of the R-matrix was low (the 2-norm condition is about 3.6) meaning that the system is not too numerically sensitive to invert.

With R^{-1} available, the converter was operated over the input voltage range and temperature data taken via thermal camera. The temperature of each device was taken during operation once the system thermally stabilized. After the data was collected, power dissipation in each device was calculated according to the thermal model. Figure 4.25a shows the comparison of the loss distribution in the converter as measured thermally, versus the SPICE simulations. Agreement between the two is reasonably good over the operating range. The MOSFET and diode dominate converter loss. The transformer, which was custom designed as a two-winding planer transformer is approximately 94% efficient. A thermal picture (Figure 4.25b) of one operating point shows the temperature measurement points used to determine the loss distribution.



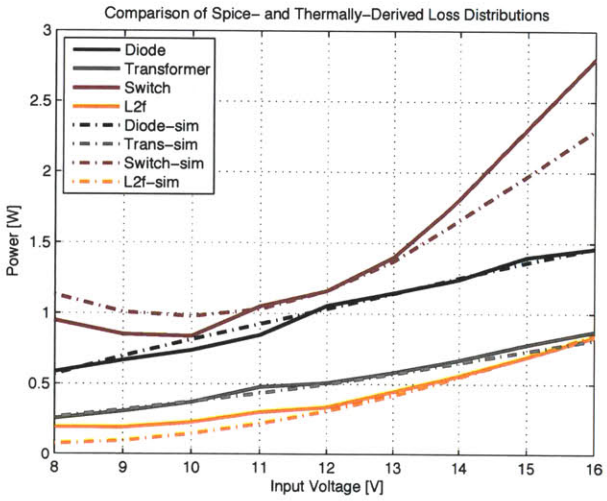
(a) MOSFET powered

(b) Diode powered

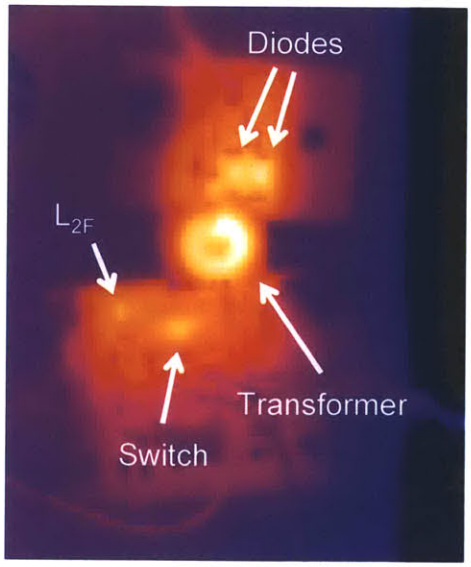
(c) Transformer powered

(d) L_{2F} powered

Figure 4.24: Power was injected to each major loss component successively and used to build a thermal model of the system. The linear behavior is evident from the plot and the successful curve fits to a linear model.



(a) Loss distribution SPICE vs. thermal

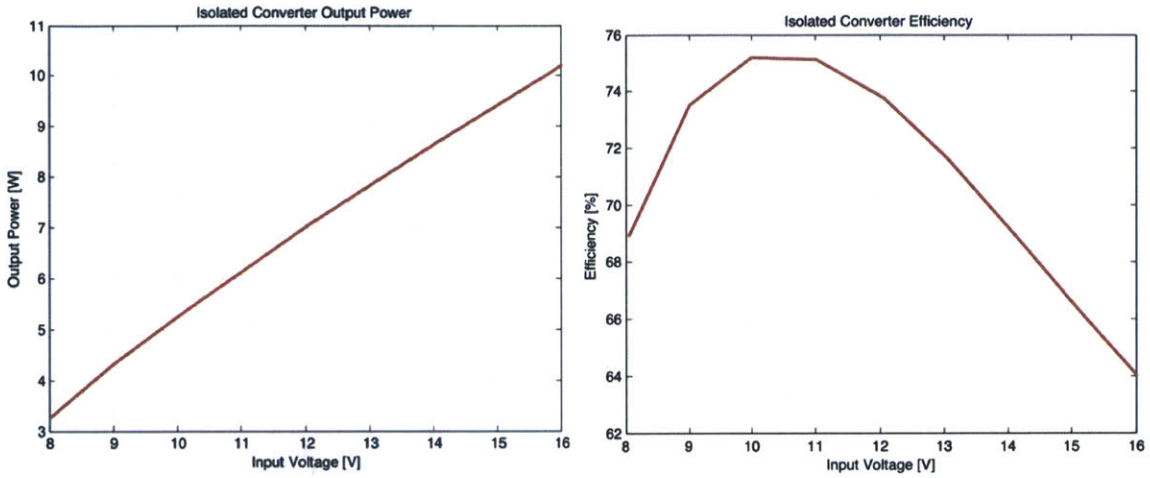


(b) Thermal camera image of converter

Figure 4.25: Results of thermal modeling show good agreement with SPICE and reveal the loss distribution in the converter. Data was obtained by repeated thermal imaging

Final converter measurements were performed to determine the efficiency and power versus input voltage. The results are plotted in Figure 4.26. The waveforms at the nominal operating point are also included in Figure 4.27. The converter posts nominal operating power of 7 W at an efficiency of 74%, with peak efficiency hitting 75%. The biggest limiting factor in converter efficiency is the device loss, which rises rapidly as power is increased. This is partly due to the choice of tuning point, which results in an undershoot around V_{IN}=13 V. The undershoot likely turns on the body diode, which helps to explain the increased loss at higher voltages and this could be partially addressed by a change in tuning point. Regardless, simulation and experiment match fairly well.

One aspect of the prototypes presented that was not discussed is the ancillary circuits needed to make the power stage operate. In Figure 4.12 two thirds of the active board area occupied by the inverter comprises a resonant gate driver and oscillator. Additional components including hotel supplies and control would require another significant chunk of area. These circuits are suitable to be integrated onto the same die as the power device. The second isolated prototype uses the MITMV2 integrated power device discussed in Chapter **??**. The area savings possible from this change alone are significant. Further reduction can be realized by addressing the packaging constraints. The die area occupied by the power device is 1.5 mm x 2 mm. Yet the package occupies several times more area (about a factor of 10 in area). Not only does this make the converter larger, but it introduces packaging parasitic inductance and resistance that constrains the design space and reduces converter



(a) Power versus V_{IN}

(b) Efficiency versus V_{IN}

Figure 4.26: Efficiency and power measurements of the final isolated prototype implemented with discrete MOSFETs

performance. To achieve the full benefit of integrating passives and devices requires the further step of designing custom integrated circuits to absorb the ancillary functions of the power converter. This is the subject of a follow-on chapter.

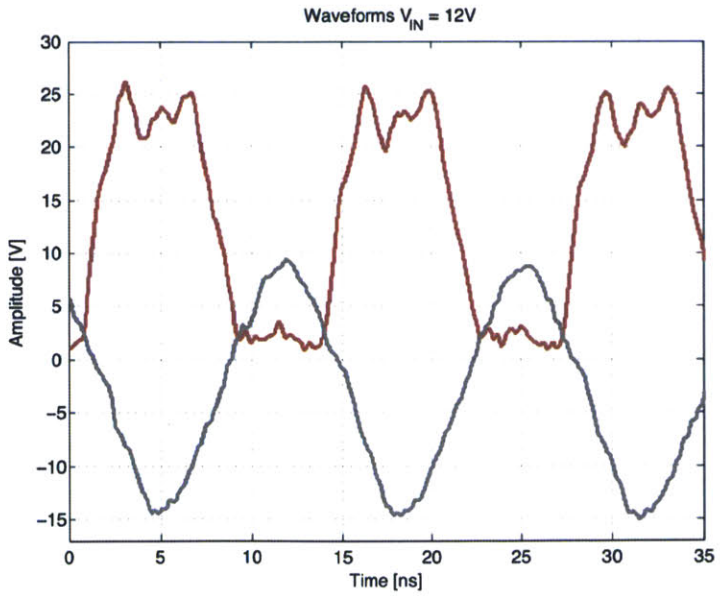


Figure 4.27: Converter waveforms for V_{IN}=12V. Red curve is V_{DS}, gray curve is V_{GS}.

Chapter 5

More-Integrated Φ_2 Converters

5.1 Background

The preceding chapters addressed some of the primary issues facing the miniaturization of power electronics. The areas explored – circuits, devices, and passive components – are of perennial concern. Each forms a critical component of the overall power converter system that demands attention if reductions in size, weight, and cost are to be achieved. In Chapter 4 an isolated power converter was demonstrated using the Φ_2 topology. The system included examples of device optimization, passive component design and integration, as well as circuit design with the goal of achieving a VHF converter with high performance and high power density.

While the component count is reduced and magnetic materials eliminated, a significant portion of the converter comprises discrete components necessary for functions such as gate drive, oscillator, hotel supply, and control. The size of these subsystems makes up more than half of the converter footprint, limiting the achievable size reduction for the overall system. These functions are well-suited to incorporation into an IC.

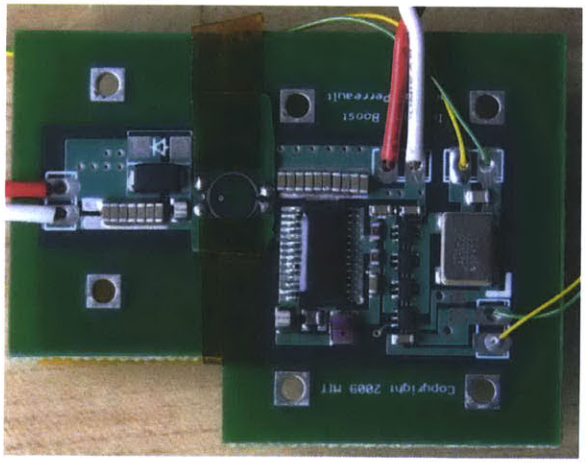


Figure 5.1: Isolated Φ_2 converter with 28-lead ETSSOP. The package footprint is more than 10x the die area

In this chapter we discuss the design and implementation of an IC to realize further reduction in converter size. The IC integrates most of the ancillary circuits necessary for a complete power converter including a clock, reference, gate drivers, control circuitry, hotel power, and configuration logic. In addition to integrating most of the converter subsystems onto the die, we also address packaging, which is a significant constraint on the overall size of the system.

In Figure 5.1 the power device is packaged in a 28-lead ETSSOP. While this package permits good heat transfer and relatively low parasitic resistance it is actually quite large in comparison to the die, which is about 2.1 mm x 1.2 mm as compared to the 6 mm x 7 mm package footprint. Besides contributing to increased size, the additional loop inductance from the package complicates circuit design because it introduces undesired resonance with the external shunt capacitance, C_{EXT}. By utilizing a flip-chip-on-board (FCOB) approach both the wasted package volume and excess parasitics can be addressed simultaneously. This leads to significant improvement in power density.

5.2 Converter Overview

Figure 5.2 shows a high-level schematic of the converter architecture. It implements the transformer-isolated Φ_2 converter using two separate ICs that provide the main power device plus all the peripheral converter system components. Both ICs in the schematic are identical, but different parts of the chip are used depending on whether it's on the input or output side of the converter. On the input side, the IC provides the main power device, clock, gate driver, hotel power, reference, and receiver. On the output side the IC acts as a comparator to sense the output voltage and a transmitter to send control signals across the isolation barrier.

Rectification on the output side is handled by a diode as in earlier implementations. Gate driver power in this case is provided via a small external buck regulator. Since the gate driver DC rail is about 2 V, a liner regulator is not acceptable from an efficiency standpoint, and the implementation of a complete buck regulator or switched-capacitor converter would have been required. The passive components in the system are similar to the isolated Φ_2 converter presented earlier. The transformer is implemented as traces in the PCB while the capacitors are surface mount ceramic devices. The capacitors connected between the ICs forming the isolated communications bus are also PCB traces placed on opposite sides of the dielectric for high breakdown strength. The basic converter specifications are listed in Table 5.1.

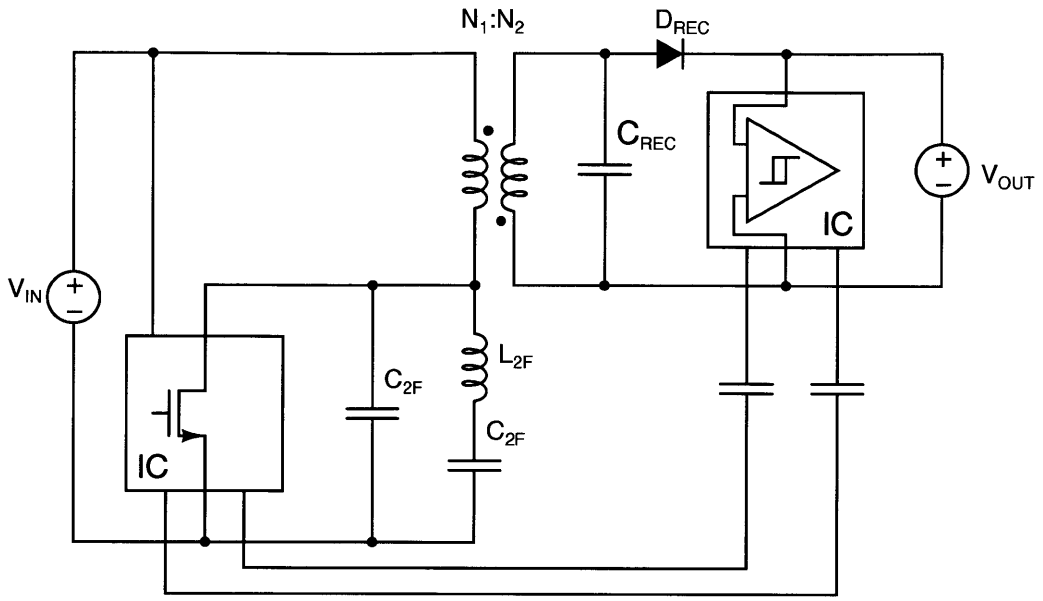


Figure 5.2: Isolated Φ_2 converter architecture showing the relationship between the converter IC and power stage components.

Table 5.1: Targeted Converter Specifications

Parameter	Value
Input Voltage	8 V - 12 V
Output Voltage	12 V
Isolation Voltage	2500 V
Output Power	7 W
Switching Frequency	75 MHz

5.3 Integrated Circuit

5.3.1 Overview

The integrated circuit is intended to implement a Φ_2 converter operating under hysteretic, voltage-mode control. It incorporates a programmable register that allows configuration of some of the operating parameters and a choice of whether the system will operate as an isolated or non-isolated converter. When operating in non-isolated mode only a single IC is required to make the complete system. For isolated operation two ICs are required. The input-side IC performs most of the converter functions while the output-side IC measures the output voltage and provides a signal across the isolation barrier in order to achieve regulation. In both cases, converter start-up is handled by setting up a default run state

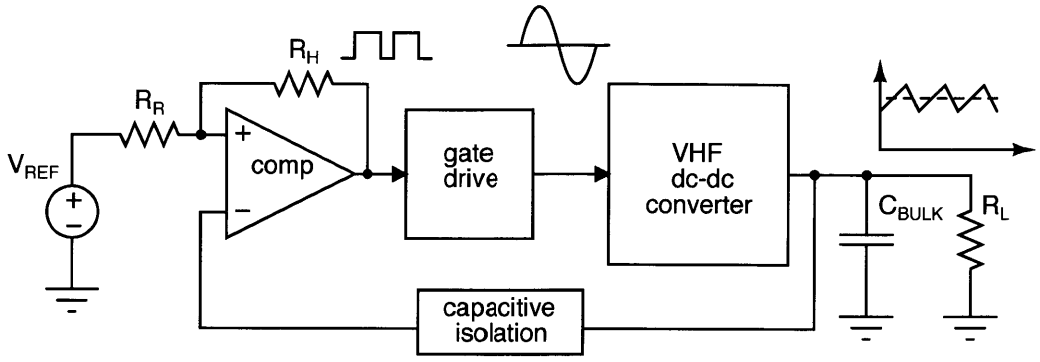


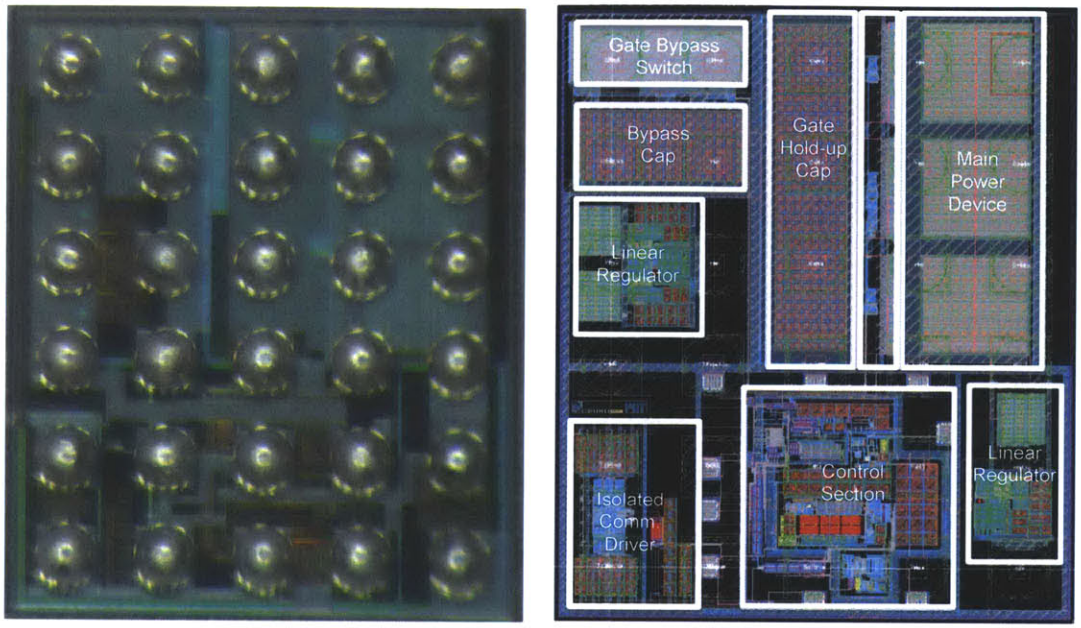
Figure 5.3: Overview of the voltage-mode hysteretic control implemented with the IC.

that is active unless the chip is put into sleep mode. The main gate driver operates as soon as power is available and until it receives a signal that the output voltage has exceeded the upper threshold. Once the upper voltage threshold is exceeded, the converter stops running and the output capacitor is discharged by the load until the lower voltage threshold is reached and the converter begins to run again. In this way, the output voltage is regulated. Both the regulated voltage and the hysteresis (which translates into voltage ripple at the output) are programmable with the configuration register. This basic operation is depicted in Figure 5.3.

A die photograph (Figure 5.4a) shows the integrated circuit designed and fabricated for this work. The chip is implemented in a 0.7 μm, 3-metal, two-oxide, BCD process that includes a number of LDMOS devices, standard 5V CMOS for logic and analog blocks, and some higher-voltage, drain-extended devices. The latter aid in the design of components that deal with the unique challenges of power IC design – namely the high voltage rails. As mentioned in the introduction and visible in Figure 5.4a, the chip is designed for FCOB. It has a 30 bump, 6 x 5 array at a spacing of 500 μm. The solder ball diameter is 300 μm. When the corresponding PCB pads are held to 270 μm routing with a 3 mil minimum trace width and 3 mil minimum trace-trace spacing PCB process is possible. This is an important consideration for manufacturing cost since a reduction to 2-2 rules (for instance if the bump pitch is tightened for the same ball diameter) increases cost by 20%-40% while simultaneously reducing the yield. The CAD layout appears in Figure 5.4b. It shows the relative sizing of the power devices, their drivers, and the ancillary control circuitry.

5.3.2 Block Diagram

The IC block diagram is depicted in Figure 5.5. It shows the various subsystems that are included on the IC at a high level and how they are interconnected from a functional point of



(a) Die photograph

(b) CAD Layout

Figure 5.4: Converter IC and associated layout files. The IC is a 30-ball 6x5 array at 500 um pitch intended for flip-chip-on-board assembly.

view. The upper right corner contains the main power transistor and associated gate driver transistors. They form the gate driver and power device system depicted in Figure 5.6. It is a half-sine resonant driver that is described in detail in Section 5.3.5. The Synchronizer and Delay module (also described in more detail in Section 5.3.6) controls the gate driver operation when the system is under modulation by the main control loop. It ensures that the gate driver circuit does not overdrive the gate at the start of each modulation cycle. The Temperature Stable Current/Voltage Reference provides a voltage and current reference that is approximately constant over the expected operating temperature range. The Oscillator is a programmable frequency system with a ring oscillator core. It uses a feedback control system and the IV reference to maintain a constant frequency as the system temperature changes. This is important as on-die oscillator options that are open loop exhibit drifts of +/- 10% over the desired temperature range–too large to maintain proper operation of the resonant circuit. Variable duty-ratio is achieved via an adjustable one-shot that is triggered on the rising edge of each clock pulse. Control is effected by a comparator which is referenced to the on-die IV reference. It includes adjustable hysteresis to set the output ripple. The hotel supplies are identical linear regulators with the exception of different references. The control register is a scan chain implemented with a series of D flip-flops. It supports the loading of configuration bits to program the reference, oscillator, one shot, and comparator. An additional TX clock designed to operate around 10 MHz supplies the signals for the isolated communications system.

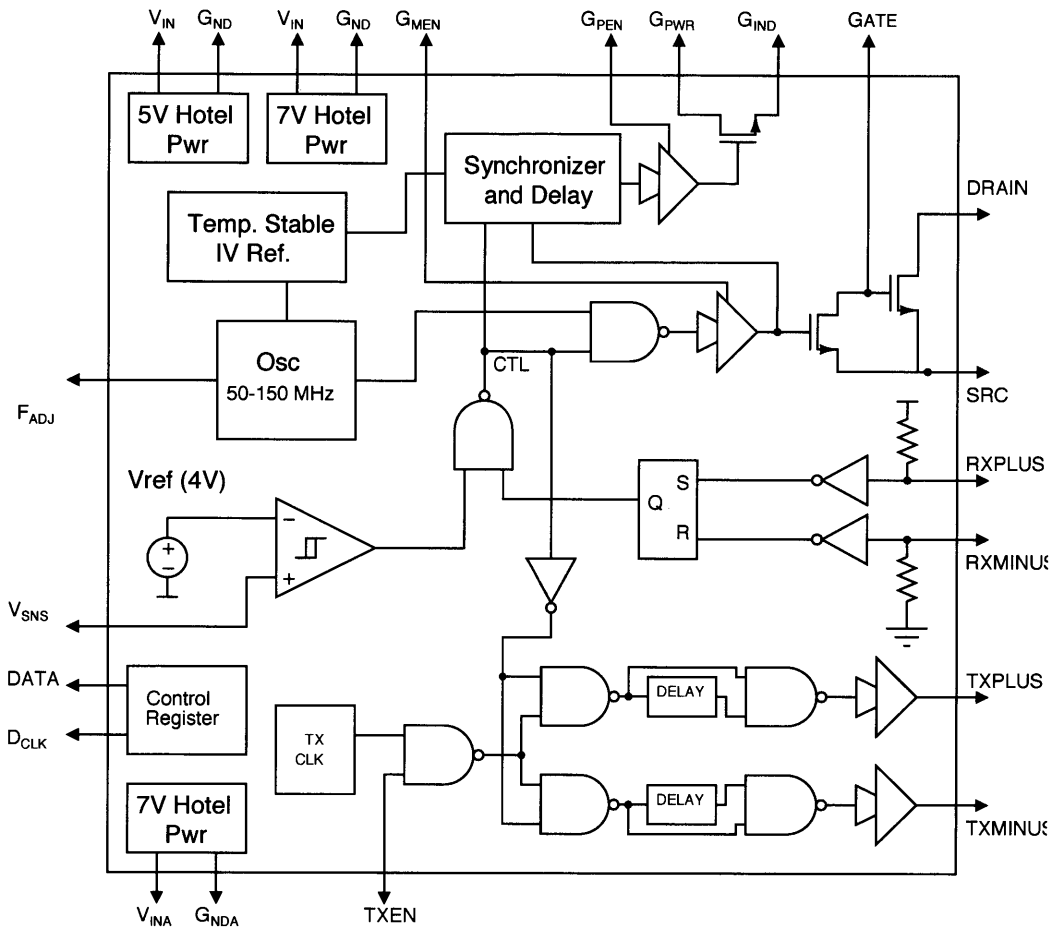


Figure 5.5: Block diagram of the isolated Φ_2 converter IC

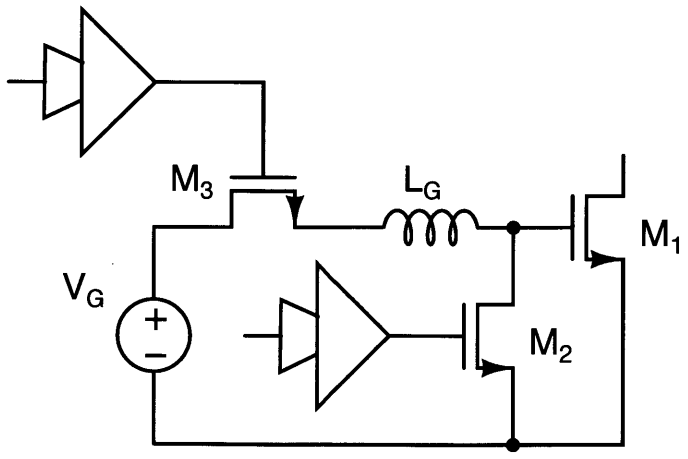


Figure 5.6: Schematic of the half-sine resonant gate driver implemented on the IC

Table 5.2 lists the IC pins and a description of their functions

5.3.3 Functional Description

The isolated Φ_2 converter implementation uses two identical ICs. In the low-side IC, VSNS is held low and TXEN is asserted to deactivate the transmitter. RXPLUS and RXMINUS

Table 5.2: IC Pin Descriptions

Pin	Function
VIN	VDD for digital voltage rails
VINA	VDD for analog voltage rail
GND	Ground for digital voltage rails
GNDA	Ground for analog voltage rails
FADJ	Analog input: Input ranging from 1.2 - 4 V, controls frequency range from 50 MHz - 150 MHz
VSNS	Analog input: Senses output voltage and compares it to the on-die reference to change converter run state, when VSNS exceeds reference voltage + hysteresis, the main power device is held off, it starts again when VSNS drops below reference voltage - hysteresis
GPEN	Digital input: When asserted disables pass transistor to gate driver circuit, shutting down driver. Can be used to override on-die control of gate drive
GMEN	Digital input: When asserted disables gate pull down transistor. Can be used to override on-die control of gate drive
TXEN	Digital input: When asserted disables TX clock
V7V	7-V rail for digital logic on die
V7VA	7-V rail for analog components on die
V5V	b 5-V rail for tapered driver that controls the state of the gate pull-down transistor
TXPLUS	Digital output: Positive rail of isolated communications system. TXPLUS set high when VSNS exceeds reference voltage + hysteresis
TXMINUS	Digital output: Negative rail of isolated communications system. TXMINUS set high when VSNS falls below reference - hysteresis
RXPLUS	Digital input: Postive rail of isolated communications system. When set high by TXPLUS signal (through isolation capacitance), converter stops
RXMINUS	Digital input: Negative rail of isolated communications system. When set high by TXMINUS signal, converter runs
DATA	Digital input: Scan chain input. Serial input to load 23-bit control register
DCLK	Digital input: Scan chain clock. Clocks the scan chain for serial load
DRAIN	Main power device drain. Drain connection for power stage
GATE	Main power device gate. Gate connection for external gate drive inductor, L_{gate}
SRC	Main power device source. Source connection for power stage
GPWR	V_{DD} rail for gate driver power (2.7-V)
GIND	Connection for external gate drive inductor, L_{gate}. Supplies power to gate driver in proper sequence to avoid driver over-voltage during modulation

have pull-down and pull-up resistors respectively such that the converter will run on power-up even before the sensors come online. An external microcontroller is needed to program the scan chain and manage chip startup. Startup begins with GPEN set to high to disable the converter. When the input exceeds 7 V, the microcontroller programs the scan chain. Programming configures the reference to be temperature stable, and determines the duty ratio. Once programming is complete and the input voltage is at least 8 V GPEN is pulled low and the main power device is clocked at the switching frequency initiating power transfer through the power stage.

As energy is delivered to the output capacitance the output-side IC begins to operate. A second microcontroller programs the scan chain to set the reference voltage and then asserts GPEN and GMEN to prevent the main gate driver from running under any circumstance. The reference and comparator begin operating effectively when the input voltage exceeds 5 V, well below the 12 V desired regulation voltage. The transmitter also begins to operate at this time and will deliver continuous pulses at approximately 10 MHz from TXMINUS while VSNS is below $V_{REF} + V_{hyst}$. This ensures that the input-side IC continues to run the power stage. VSNS is tied to the converter output by a 3:1 resistor divider. When the converter output reaches 12 V plus the hysteresis voltage, TXMINUS is set to ground and TXPLUS is pulsed at a frequency of 10 MHz. This causes the input-side IC to stop gating the main power device. As the load begins to draw energy from the output capacitor, the output voltage drops. When the output falls below the hysteresis band, the 10 MHz pulse appears at TXMINUS and TXPLUS is set low. The converter restarts and the process begins again. During the start of each modulation cycle, the Synchronizer and Delay block ensures that the gate is not overdriven.

The chip also supports a non-isolated operating mode. The details are very similar excepting that only one chip is used. RXPLUS is asserted and RXMINUS is held low bypassing the receiver function. This allows the on-die comparator to change the operating state of the power stage. Similarly, TXEN is asserted disabling the transmit clock so that the pad drivers do not waste energy. The rest of the system functions as described above. When the converter is first powered up, a microcontroller holds GPEN high, disabling gate drive, and then programs the scan chain when the input voltage is at least 7 V. Once this is accomplished the converter will begin running. The output is regulated in the same way described above except that the transmit-receive system is not involved because there is no isolation barrier. Instead the comparator input VSNS is tied to a voltage divider across the converter output which results in direct regulation of the output voltage.

Detailed descriptions of the various circuit blocks that make up the IC follow below.

5.3.4 Power Device

The main switch comprises an NLDMOS device with 2.5-μm fingers arranged on an 11.7-μm pitch with a 54-V breakdown. The device is placed in an isolation tub. In this case the isolation tub helps to ensure that few stray carriers reach the substrate directly. Coupled with source pins that are separate from chip ground, this minimizes ground bounce effects on the die. In this process, the 54-V LDMOS device requires that the operating peak drain voltage does not exceed the well voltage by more than 30-V, thus the well is tied to VDD. This ensures that even at the high end of the input voltage range, the peak drain excursion is less than the allowed maximum. The switch was optimized using the procedures outlined in Chapter 2. In order to minimize metal losses, the drain, source, and gate terminals are each allocated 3 solder bumps. The solder bumps in each set are tied in parallel by copper traces on the PCB. The device consists of 1287 LDMOS cells tied in parallel. Each has an effective gate width of 70 μm for a total device width of about 90 mm. This yields an on-state resistance around 120 $m\Omega$ and a nominal output capacitance of about 83 pF.

5.3.5 Gate Driver

A half-sine, fully-resonant, gate drive scheme was picked for the Φ_2 converter IC design. As compared to hard gating it is capable of delivering substantially higher efficiency at the switching frequency (75 MHz). The ideal loss of intended design is approximately 27 mW instead of the 350 mW hard gating case. In practice, when the losses from the gate driver device parasitics are included the savings are less. The gate loss is reduced by close to a factor of six. This is an improvement over the fully sinusoidal scheme where practical implementations have yielded about about a 50% reduction in loss. It also offers a more compact solution requiring only one external inductor and a single hold-up capacitor.

The schematic of the gate driver is shown in Figure 5.6. M_1 is the main power device and M_2 is the gate drive pull-down device. M_3 is a pass transistor that prevents a DC short when M_2 is held on to turn off the power stage during modulation. All the transistors, including the tapered drivers are on die. In addition, buffer capacitance is included on die so that the gate drive power rail does not experience excessive bounce during transitions. The inductor is a discrete surface-mount inductor manufactured by Coilcraft.

The gate drive circuit operates as follows: When the gate drive is in periodic steady state, M_3 is held on continuously by the tapered driver T_1. M_2 is driven at the switching frequency (via tapered driver T_2) with a duty ratio equal to $1 - D$, where D is the duty ratio of the main power device. The gate voltage v_{GS} and the inductor current i_G appear in Figure 5.7 for steady-state operation. During the period M_2 is on, i_G ramps to its maximum value.

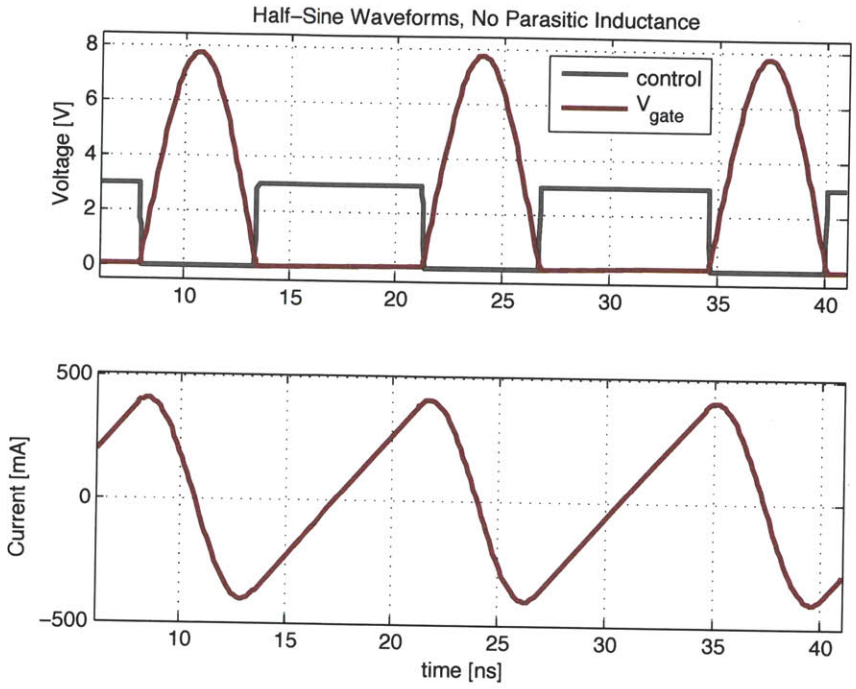


Figure 5.7: Half-Sine resonant gate driver waveforms when no parasitic inductance is included in the drain-source loop of the pull-down device.

Once M_2 commutates L_G reacts with C_{GS} plus the output capacitance of M_2 generating a half-sinusoid that drives the main device through a switching cycle as v_{GS} rings to $v_{GS_{peak}}$ and then back to zero. Once v_{GS} returns to zero, M_2 turns back on and i_G ramps from its minimum value of $-i_{G_{peak}}$ back to $i_{G_{peak}}$ and the cycle repeats.

Several points regarding the operation of the circuit are worth clarifying. First, the resonant half-sine driver recovers all the energy from the gate capacitance. This makes it ideally as efficient as the fully sinusoidal driver. Since the duty ratio with this circuit is set by the resonant frequency of L_G and the net gate capacitance, the ideal efficiency depends on the threshold voltage of the device, the desired resonant frequency (duty ratio), and $v_{GS_{peak}}$. For instance, there may be some desire to increase $v_{GS_{peak}}$ and the resonant frequency to achieve sharper switching transitions. While this increases loss in the gate driver somewhat, it may reduce loss in the power stage. Therefore assessing whether the a half-sine resonant driver is more or less efficient than a sinusoidal resonant driver requires co-design with the power stage and device models.

In practice, the efficiency of the half-sine driver is dictated largely by the devices M_2 and M_3 and their tapered drivers. All must be optimally sized to minimize loss. Once the desired gate voltage shape is determined by choosing the resonant frequency and peak voltage,

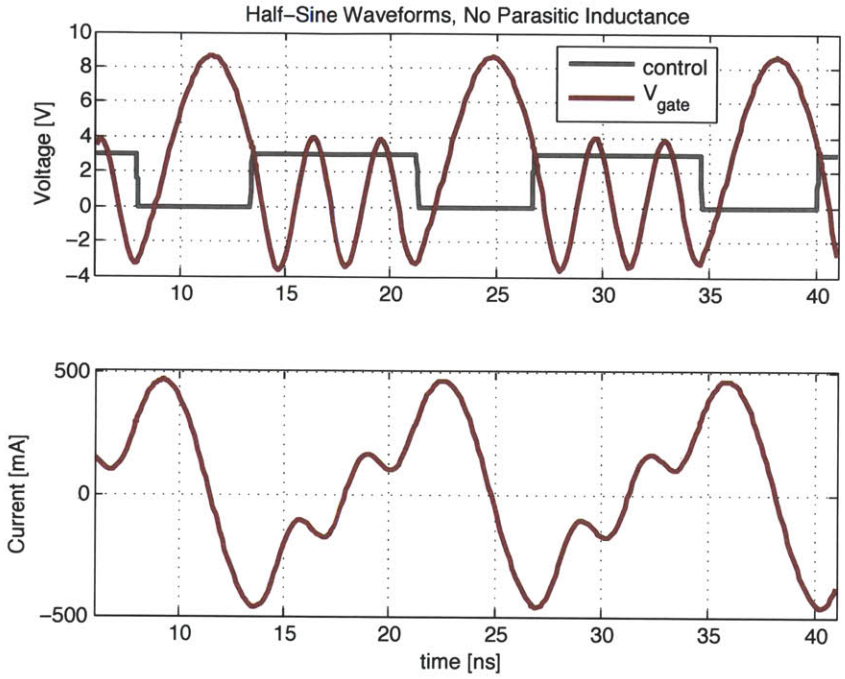


Figure 5.8: Half-Sine resonant gate driver waveforms including parasitic inductance in the drain-source loop of the pull-down device.

design of the gate driver can proceed. The losses in M_2 can be found by calculating the conduction and displacement losses from the RMS values of their respective currents and then summing the gating loss. A good approximation of the latter is to multiply the ideal hard-gating loss by about 1.3x, which is the practical limit for the process used in this work. By scaling the device area towards its optimum (thus exchanging conduction and frequency-dependent losses), the total loss from M_2 is minimized. M_3 is sized by similar considerations. However, with the modulation frequency only a few percent of the switching frequency, the minimum achievable loss is limited by the amount of area available for the device. M_3 tends to be much larger than M_2 because it carries the same RMS current as L_G (rather than the fraction that M_2 carries) during steady-state operation. While it's possible to place a capacitor to ground at the node common to M_3 and L_G to shunt the resonating current, the reduced conduction losses are offset by capacitive discharge losses that result from modulation. As implied in the previous paragraph, iteration around this procedure and the choice of $v_{GS_{peak}}$ and the resonant frequency will lead to a global optimum efficiency.

It is difficult to achieve a practical half-sine gate driver using discrete transistors (or transistors that are not on the same die) at switching frequencies near 10s of MHz or above. This is due to the effect of inductance in the loop formed by M_1's gate-source terminals and M_2's drain-source terminals. As little as a few hundred pH of inductance appearing

in this loop results in significant ringing that destroys the effectiveness of the gate driver (see Figure 5.8). [1]. Inductances of this scale occur for any practical loop area that can be achieved where M_2 and M_1 are not on the same substrate. On the other hand, with both transistors on the same die and sharing the same metal structures, the loop inductance can be maintained at a very small value. Thus, an integrated implementation makes the half-sine practical.

Another matter that must be addressed is startup of the half-sine driver. In periodic steady state the current in L_G swings negative during each cycle. While M_2 is on, i_G goes from its minimum value, $-i_{G_{peak}}$, to its maximum value, $i_{G_{peak}}$. This happens in the period (1-D)T. When the gate driver is under modulation as part of a converter control scheme, PSS is not guaranteed. The first half of the intended modulation occurs when the gate driver is shut down by turning M_1 off and holding M_2 on. Restarting the gate driver to complete the modulation cycle amounts to turning M_1 on (while keeping M_2 on) for a period to ramp the current in L_G and then operating M_2 as in PSS. If the initial startup period is still (1-D)T, then i_G will reach twice its intended value and the peak voltage on the main gate of the power device will overshoot. To avoid this problem, the first cycle at the beginning of each modulation period should be exactly half of the PSS period, or (1-D)T/2. This amounts to a single-cycle soft-start function and requires fairly precise timing. Fortunately, these are characteristics compatible with on-chip implementations. Figure 5.9 shows the waveforms of the half-sine driver when the startup effects are not taken into account. Figure 5.10 is the case when the initial ramp period is held to exactly half of the PSS value. Clearly, the second case is desirable. In this design the generation of a runt pulse at the start of each modulation cycle is accomplished via the Delay and Synchronizer block which will be described in the following section.

Physical implementation of the gate driver for this design used a substantial area (about 25% of the total die), largely due to a combination of hold-up capacitor requirements and the pass transistor. The gate driver pull-down device, M_2 is relatively small, less than 10% of the power device area. The function is handled with a low voltage CMOS logic device intended to operate at logic-level voltages of 5 V. However, this device has a 12-V breakdown making it useful for the 8 V peak chosen for this gate driver design. The device is constructed by paralleling 312, 54-μm fingers for a total gate width of 16.8 mm. Since this device operates at high frequency and relatively high current levels, bulk contacts were included between every pair of source terminals. The bulk and source terminals of this device share the same metallization as the source terminals of the main power device.

[1] The point at which parasitic loop inductance starts to become unacceptable is largely a question of the characteristic impedance of the circuit. This is determined by the operating frequency and C_{ISS} for half-sine driver topology. Unfortunately, the magnitude of Z_0 with respect to typical damping in the circuit makes ringing very likely even for small parasitic inductances

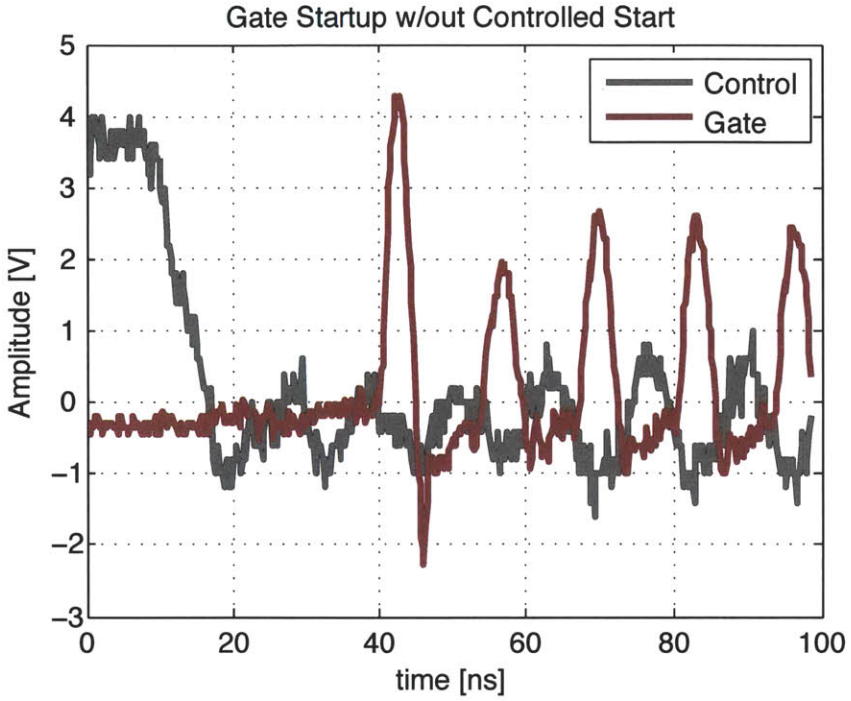


Figure 5.9: When the gate driver is started from a zero-state condition without compensating for the lack of negative inductor current at t_{0-}, the first gate pulse is oversized, risking damage to the power device gate oxide.

This ensures a tight return loop for gate current avoiding the excessive ringing that would negatively impact the gate signal. It also permits the gate currents to traverse the source bumps out to the PCB further easing ground bounce. The gate of the main power device and the drain terminals of the pull-down devices also share metal to keep the loop small. The pass transistor is quite large because the RMS current it carries requires a low resistance to avoid excessive loss. The device uses the same low-voltage CMOS device, but is 48 mm in equivalent gate width. In terms of die area, it occupies approximately 30% of the area of the main power device. The tapered drivers for both cases occupy a small area relative to the power devices, approximately 10% of the area of each of M_2 and M_3.

In addition to the power devices and tapered drivers associated with the half-sine driver, substantial area is devoted to on-die capacitors. These capacitors ensure that the power rails don't collapse each cycle as the gate driver operates. The approximate 1 nH of loop inductance from the rail to the external hold-up capacitor would otherwise result in substantial bounce. In the case of this design, the total die area was pin-limited. Approximately 30 bumps were required to bring all the signals off die, and the resulting silicon area was greater than that required for the power devices plus all the converter sub-systems. As a result the capacitors were allowed to fill the remaining die area.

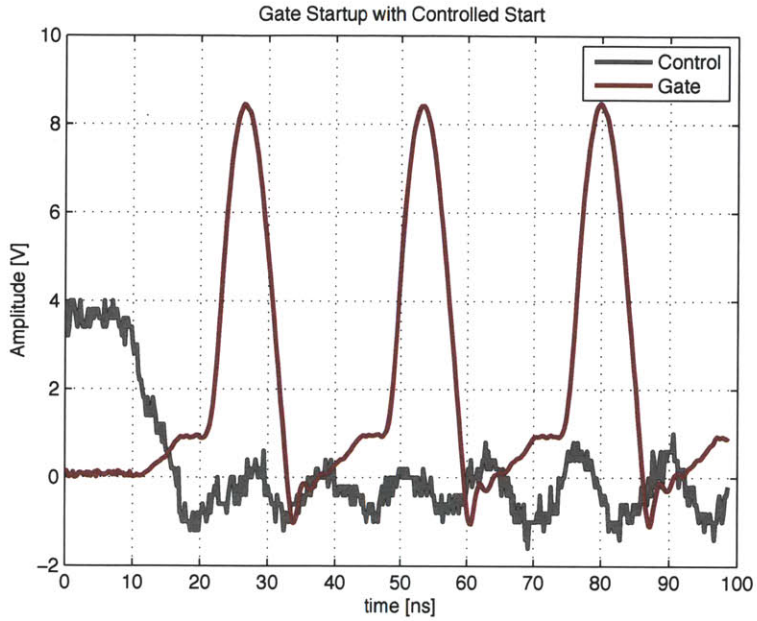


Figure 5.10: By only charging the inductor during the first cycle after startup for half the steady-state charging period, an over-voltage condition is avoided.

5.3.6 Synchronizer and Delay

The Synchronizer and Delay block serves the important function of managing the gate driver as the power stage is modulated to control the output. As mentioned above, each time the gate driver is starts from the zero-energy condition (i.e. i=0) in the gate drive inductor, L_G, i_G should only be allowed to ramp for half of the steady-state period. This is because the current would otherwise overshoot resulting in gate oxide breakdown. The synchronizer and delay block achieves this by first synchronizing the start of the gate driver to the main clock by waiting for the next rising edge of the clock and then providing an appropriate delay before powering the pass transistor M_3. The delay results in an approximately half-length ramp to ensure the proper value of i_G is reached at the end of the clock period.

The circuit to achieve this operation is depicted in Figure 5.11. It takes the form of a single D flip-flop with a reset circuit. The inputs to the circuit are CTRL, CLK, and GateReturn while the outputs are GateMain and GatePwr. CTRL derives from the controller and when it is asserted the gate driver runs. CLK is the output of the one-shot that generates the desired duty ratio pulse after being triggered by the system clock that drives the power stage. GateReturn is a signal taken from the output of the tapered driver which drives the gate pull-down switch M_2. GateMain drives the tapered driver for M_2 and GatePwr drives the tapered driver for M_3. When the controller sets CTRL to high, the data input of the

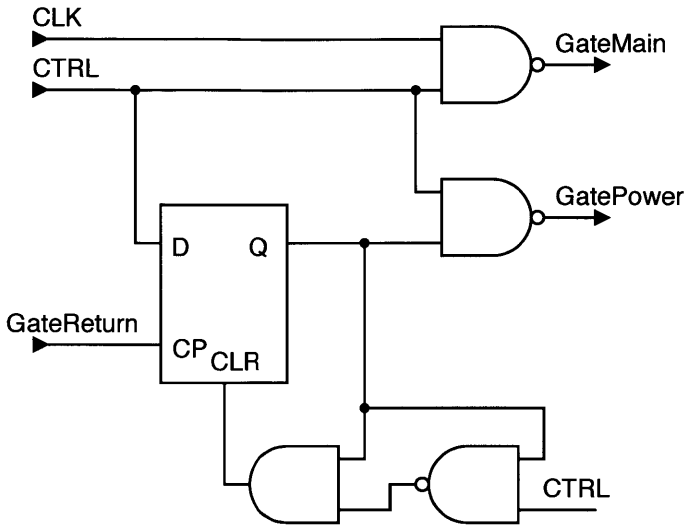


Figure 5.11: The synchronizer and delay block

D flip-flop is set high and the NAND gate J_2 begins to pass CLK. As CLK propagates, GateMain eventually changes state and propagates through the tapered driver T_2 to drive M_2's gate high, turning it on. During this period, M_3 remains off because the Q-output of J_1 will remain at zero until it is clocked.

Once GateMain propagates to the gate of M_2, it is fed to GateReturn and clocks J_1 at which point the high value of CTRL is passed to Q. This signal propagates through J_3 on its way to the tapered driver T_1 where it ultimately drives the gate of M_3 high, beginning the gate-current ramp. The falling edge of CLK turns off M_2 and the gate of the main power device rings at this point. As long as CTRL remains asserted, M_3 will remain on and the gate driver will run in periodic steady state as described earlier. The length of the half-ramp is set mainly by the propagation delay of J_1 which was adjusted during the design phase for the desired clock frequency. When CTRL is set low by the controller, the reset circuit generates a pulse on CLR of J_1, which causes GatePwr to go low and shuts down the gate driver. The waveforms for a couple of modulation cycles are depicted in Figure 5.12.

5.3.7 Voltage and Current Reference

The on-chip reference is used to set various currents and voltages in the timing blocks as well as to provide a voltage reference for the controller. Since the converter IC is subjected to a wide operating envelope, the reference circuit must be designed to accommodate large temperature swings while maintaining fairly constant output voltage and current. A classic bandgap reference will provide a stable voltage and various methods exist to create con-

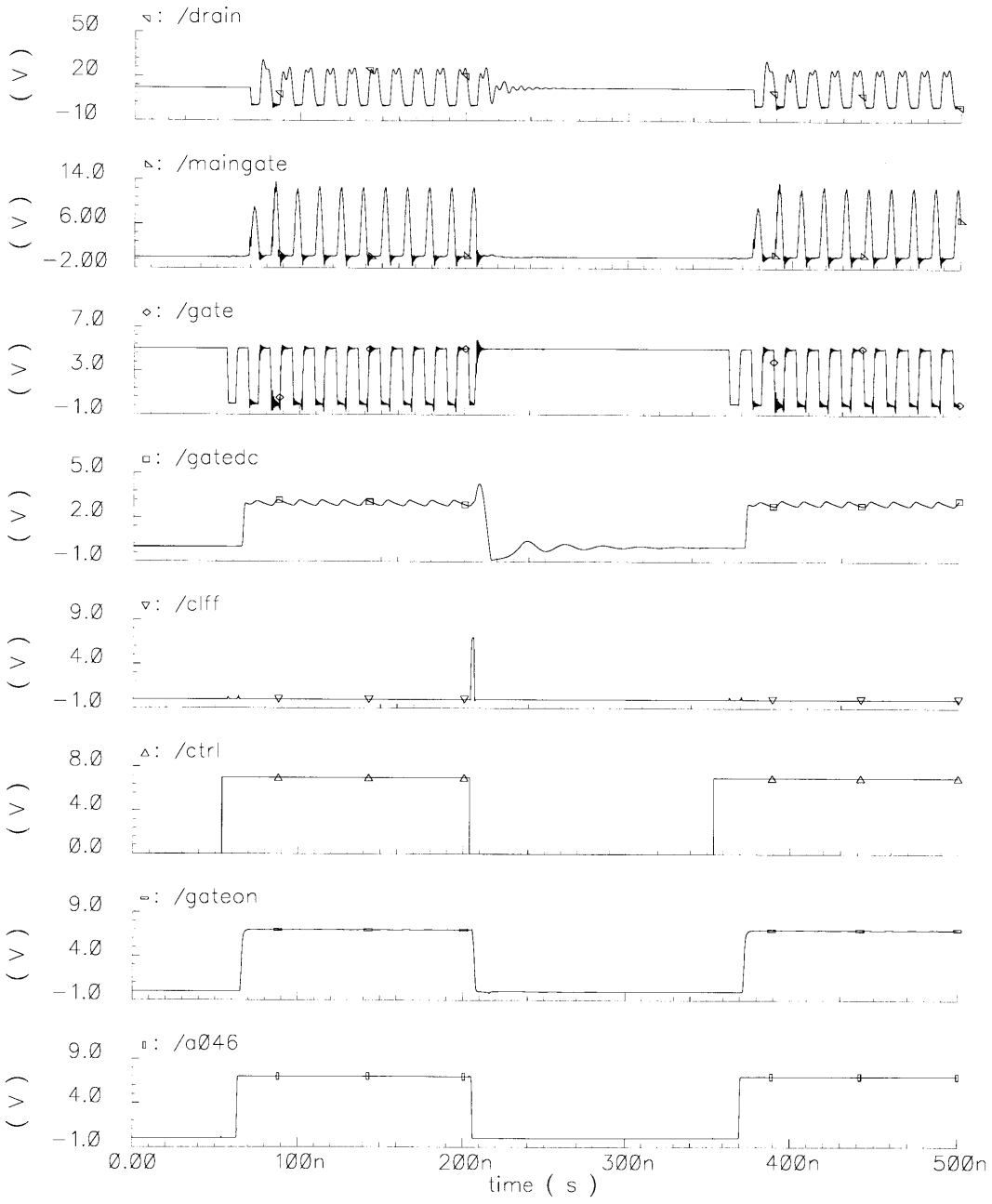


Figure 5.12: These are the simulated waveforms of the power stage, gate driver and gate-driver control signals during a few modulation cycles.

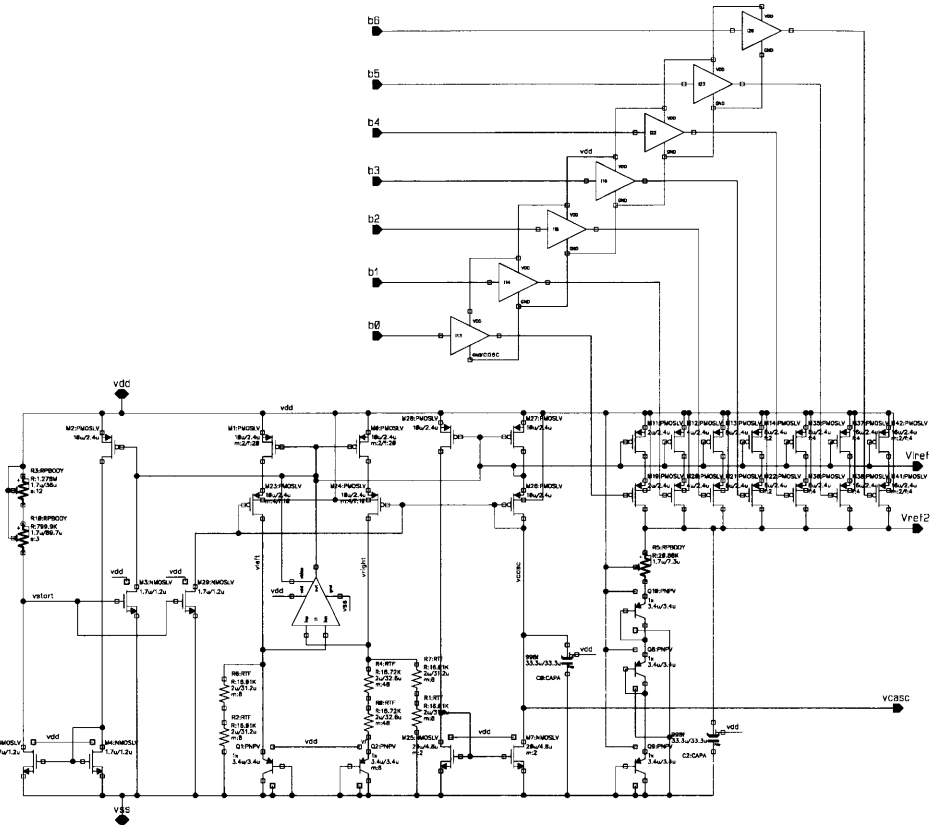


Figure 5.13: Schematic of the IV reference implemented on the chip

stant currents. A choice was made to use a reference circuit that would yield correlated variation in the output voltage and current. This behavior enhances the stability of the clock generator circuit over temperature, and helped direct the selection of the reference topology, an adaptation of the work found in [52].

A schematic of the reference appears in Figure 5.13. The core of the reference is an op-amp-servoed bandgap circuit. Q_1 and Q_2 are operated at an 8:1 current density ratio in order to generate a PTAT (proportional to absolute temperature)voltage. This voltage in turn generates a PTAT current in R_1 owing to the fact that nodes V_{right} and V_{left} are servoed equal by the op-amp. The current generates a CTAT (complementary to absolute temperature) voltage in Q_2 which sums with the PTAT voltage across R_0 and R_4 to yield a temperature-independent voltage at node V_{right}. Since a temperature insensitive current is desired, additional resistors R_1, R_2, R_6 and R_7 are connected to nodes V_{right} and V_{left}. These resistors have a positive tempco that causes their currents to decrease as the temperature rises. The resulting current is summed at node V_{right} with the PTAT current from Q_2

yielding a temperature-stabilized I_{REF} when the resistor values are chosen appropriately. The resistor values in the bandgap circuit are chosen according Equation 5.1 in order to get the smallest temperature sensitivity.

$$r = \frac{qV_{EB,T0}}{kT_0 nln(N)} \cdot \frac{\beta + \gamma}{\alpha - \gamma} \tag{5.1}$$

Where $V_{EB,T0} = nV_T ln(N)$, T_0 is room temperature, k is Boltzmann's constant, β is the thermal coefficient of V_{EB} as represented in equation: $V_{EB} = V_{EB,T0}(1 - \beta \cdot \Delta T)$, γ is the thermal coefficient of R_0 as represented in the equation: $R_0 = R_{0,T0}(1 + \gamma \cdot \Delta T)$, and α is the thermal coefficient of the thermal voltage as represented in the equation: $V_T = kT_0(1 + \alpha \cdot \Delta T)$.

The voltage reference is derived from the current reference by using a string of diode-connected BJTs in series with a resistor. The current in the series string is set by a bias network that acts as an adjustable-ratio current mirror. The latter is used to trim process-induced offsets. By selecting the appropriate current value the variation in the bias current offsets the temperature coefficient of R_E yielding only a small variation. The bias network comprises a set of binary-weighted transistors that allows for a 7-bit change in current.

The leftmost portion of the circuit is devoted to startup. The startup circuit guarantees that the identical-zero condition on power-up (where no current is flowing in any device) does not persist indefinitely. At startup if no current is flowing in the bias transistors, resistors R_3 and R_10 pull the gates of Q_3 and Q_29 high. These devices bias the gates of the balancing mirror which starts the operation of the bandgap circuit. As it approaches steady state a small bias current is drawn through R_3 and R_10 which turns off Q_3 and Q_29 and allows the reference to operate under control of the operational amplifier. The op-amp is a simple two-stage design (depicted in Figure 5.14) with a PMOS diff-pair to deal with the low common-mode voltages. Instead of having an internal start and bias networks, it's bias is bootstrapped from VI_{REF}, which provides for a smooth startup and avoids duplication of bias current paths. With both a stable reference voltage and current generated from the same circuit the variations between them are correlated, the desired behavior from the perspective of the clock circuit.

5.3.8 Oscillator

Since the Φ_2 power stage is a tuned, resonant system the oscillator needs to maintain a constant frequency over the entire converter operating envelope. Variations in clock

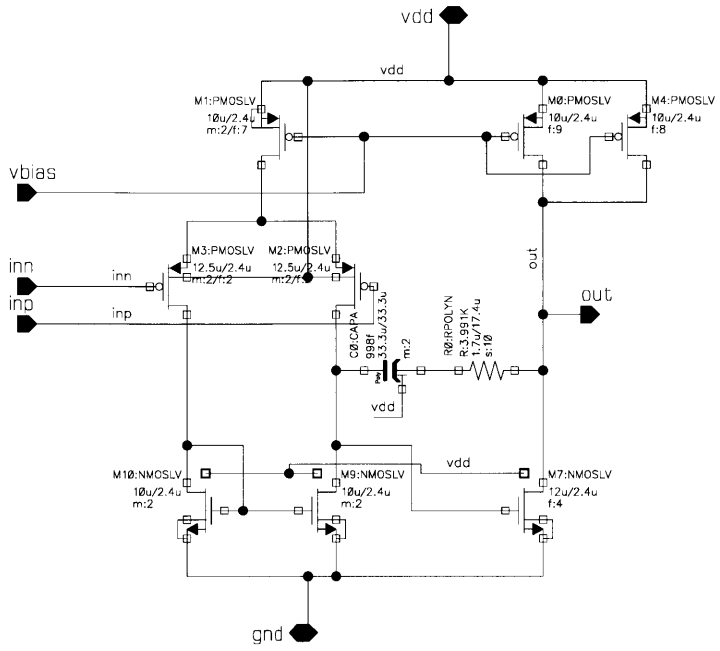


Figure 5.14: Opamp used in the IV reference with a boot-strapped biasing scheme that simplifies the design and makes startup of the IV reference easier.

frequency result in reduced output power and efficiency. While the exact numbers depend on the tuning point of the converter as well as the load network, the converter design in this work has an efficiency drop of 12% for a 10% increase in switching frequency. Similarly the output power drops by 25%. Since most on-chip oscillator solutions vary with temperature and voltage by at least 10%, another solution is necessary.

One possible solution is to achieve a stable operating frequency is to use either a crystal controlled oscillator, or an LC tank. Both solutions are off-die and require additional

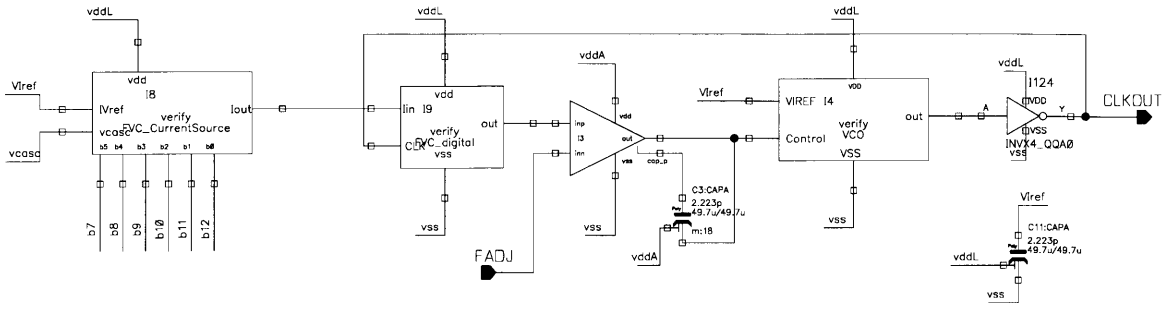


Figure 5.15: Top-level schematic of the VLRO, which is used to generate a temperature-stable clock waveform.

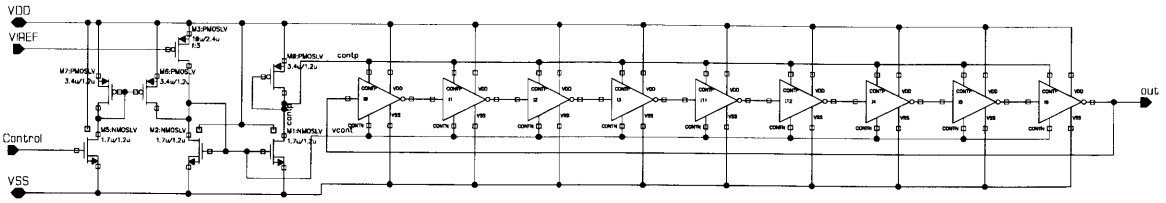


Figure 5.16: The 9-stage ring oscillator comprising current starved inverters. It forms the VCO used in the VLRO and has a minimum idle frequency of approximately 35 MHz to avoid startup problems.

components. In the case of the crystal, in particular, only a limited selection of frequencies is available so hitting a specific operating frequency would likely require the addition of a fractional-N synthesizer. The power and area implications in a 0.7 μm process such as this are significant. The LC option was avoided in favor of an on-chip option that relies on feedback around a stable reference to produce a constant frequency.

The particular oscillator system chosen is known as a voltage-locked ring oscillator (VLRO) [52–54]. At its core is a standard current-starved ring oscillator which functions as a voltage-controlled oscillator (VCO). The VCO is wrapped in a feedback loop that keeps the frequency constant in the face of voltage, temperature, and process variations. [2] The basic architecture of the system is depicted in Figure 5.15. The output of the VCO drives a frequency-to-voltage converter (FVC) that produces a DC voltage proportional to frequency. That is compared to a reference voltage by an op-amp that in turn servoes the VCO control voltage to maintain constant frequency. A temperature and voltage stable reference is supplied by the on chip IV reference.

The ring oscillator, depicted in Figure 5.16, is a 9-stage ring comprising minimum-size, current-starved inverters. Each inverter has current limiting devices at both the positive and negative rails to make the response to control inputs more linear as well as to maintain the duty ratio close to 50% over the operating range (a requirement of the FVC, which is sensitive to duty ratio). The control input is channeled through a series of mirrors that provide for roughly the same charging and discharging currents during current-starved operation. It also provides an idle current setting so that the oscillator is guaranteed to start. This is accomplished through transistor Q_3 that takes a gate voltage driven by the on-chip VI reference to set a constant current. With the idle current and a control voltage of 0 V, the ring oscillator starts and will run at a minimum frequency of 35 MHz over process corners at room temperature. This ensures that the loop will eventually close, rather than sit in a zero state as would be possible if the oscillator were allowed to simply stop running.

[2] A typical ring oscillator will see a variation in oscillating frequency greater than +/- 25% over a 0-100C temperature range.

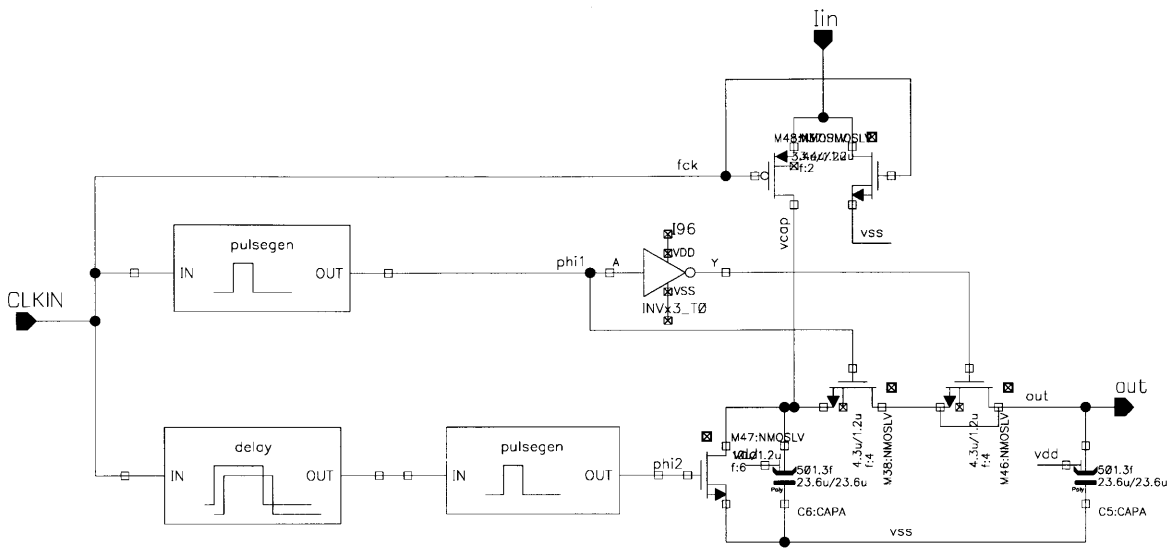


Figure 5.17: Frequency-to-Voltage Converter used in the VLRO. It charges C_6 with a constant current for a time equal to the low swing of CLK. This result is an output voltage proportional to frequency which is then fed back for regulation.

The control input is valid from 0 V to V_{DD} and the ring will oscillate over this range from about 35 MHz to 180 MHz. This exceeds the design range of the VLRO which is targeted at being able to operate from 50 MHz to 150 MHz as the off-chip control voltage, F_{ADJ} is changed.

The FVC [55] (Figure 5.17) works by charging capacitor C_6 with a constant current while CLK is low. When CLK goes high the charging current sourced via Q_{30} is cut off and Q_{48} clamps node $V_{I_{in}}$ to avoid excessive charge injection on C_6 and to allow the FVC to come to steady state well within the period of a clock cycle when CLK next goes low. Immediately after charging is stopped, a pulse is generated by pulse generator, P_1. This shorts C_6 and C_5 via Q_{38} and causes charge redistribution. Q_{46} has a shorted drain-source and is driven with opposite phase to Q_{38} to stem charge-inection effects. Once the output of P_1 goes low again, Q_{38} is opened and the capacitors are isolated. P_2 subsequently pulses Q_{47} on, shorting C_6 to ground and resetting it's voltage to zero. When CLK goes low, the cycle repeats and C_6 is charged up to a peak voltage that depends on the charging current (fixed by the on-chip reference) and the absolute time that CLK is low (which depends on both the duty ratio and frequency of the clock). Over successive cycles at constant frequency and duty ratio C_5 will charge to an equilibrium voltage that is proportional to the frequency of CLK. This allows closure of the feedback loop in the VLRO.

When the input signal, CLK, has a 50% duty ratio, the output voltage V_{OUT} is expressed as [52]:

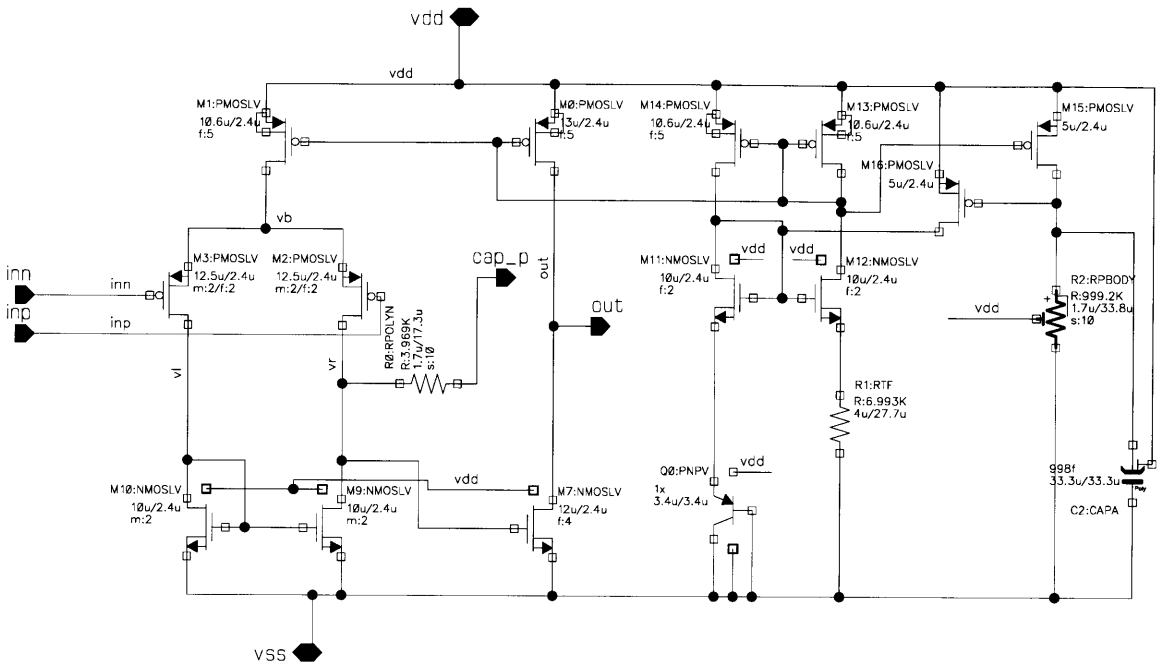


Figure 5.18: Opamp for the VLRO.

$$V_{OUT} = \frac{I_C(\frac{T}{2})}{C} = \frac{I_C}{2 \cdot Cf} \tag{5.2}$$

Where T is the period, C_6 and C_5 are equal and represented as C, and I_C is the charging current. After the first cycle, V_{OUT} is half of V_{C1} and will approach V_{C1} over many cycles:

$$V_{OUT} = V_C \left(1 - \frac{1}{2^N}\right) \tag{5.3}$$

In order to accommodate process variation, the current I_C that charges C_6 is generated by a variable-gain mirror from the IV reference. Six bits control the current magnitude and can be programmed using the scan chain.

The output of the FVC feeds directly into the op-amp that closes the feedback loop. The op-amp is a simple 2-stage design depicted below in Figure 5.18. It has a PMOS input stage for common-mode range that reaches toward V_{SS} and uses a standard compensation scheme that is used to set the bandwidth of the VLRO loop. An internal bias generator and start-up circuit manage the housekeeping essentials for the op-amp.

Figure 5.19 shows the simulated operating waveforms of the complete VLRO. The signal V_{REF} is the external adjustment voltage derived from the F_{ADJ} pin. At start-up, V_{CONT} is sitting at 0 V and the ring-oscillator is at a free-running frequency set by the idle current as discussed earlier. The output of the FVC, V_{FB} is sitting at zero. As the ring oscillator continues to drive the input of the FVC, V_{FB} rises up to a peak voltage of around 4 V where it saturates. During this time, V_{CONT} begins rising and the frequency of the ring oscillator ramps. This results in a drop in V_{FB} which stops when V_{FB} and V_{REF} are nearly equal (offset by the gain of the opamp) and the VLRO is in lock. At this point, the output frequency will remain approximately constant over temperature and voltage variations (approximately 0.5% variation over temperature and voltage corners was observed in simulation).

5.3.9 One-Shot

The one-shot generates a pulse on the rising edge of CLK that creates the desired duty ratio for the gate drive. It's implemented as a variable delay that is created by a series of current-starved inverters. The inverters are biased by a network of devices that allows the delay to be varied with 7-bit precision. The output of the delay drives one input of a NAND gate while CLK drives the other input. At the rising edge of CLK, the output of the NAND goes low until the signal propagates through the delay and causes the NAND to change state again. This results in a pulse that is variable in duty ratio from about 15% to 50%. When the output of the NAND is inverted, a square wave with a duty ratio from 50% to 85% is generated, and this provides the approximately 66% signal that is necessary to drive the high-frequency gate drive. The schematic is displayed below (Figure 5.20).

5.3.10 Comparator

The comparator effectively provides the bulk of the control loop for the Φ_2 converter as implemented here. The control scheme is a voltage-mode hysteresis implementation as outlined in the schematic of Figure 5.3. Under voltage mode hysteresis, the output voltage is sensed by resistor divider (in this case formed by R_1 and R_2). The divided-down output voltage is fed to the comparator input, where it is compared against a reference voltage (4 V) generated internally to the IC. When the sensor voltage, V_{SNS}, falls below the hysteresis band (i.e. $V_{SNS} < V_{REF} - \frac{V_{HYS}}{2}$) the comparator outputs a high and the converter begins to run. At this point, the converter delivers its rated power which splits between power delivered to the load and energy stored in the output capacitor, C_{B1}. The output voltage rises until V_{SNS} reaches $V_{REF} + \frac{V_{HYS}}{2}$. The comparator then outputs a low signal and the

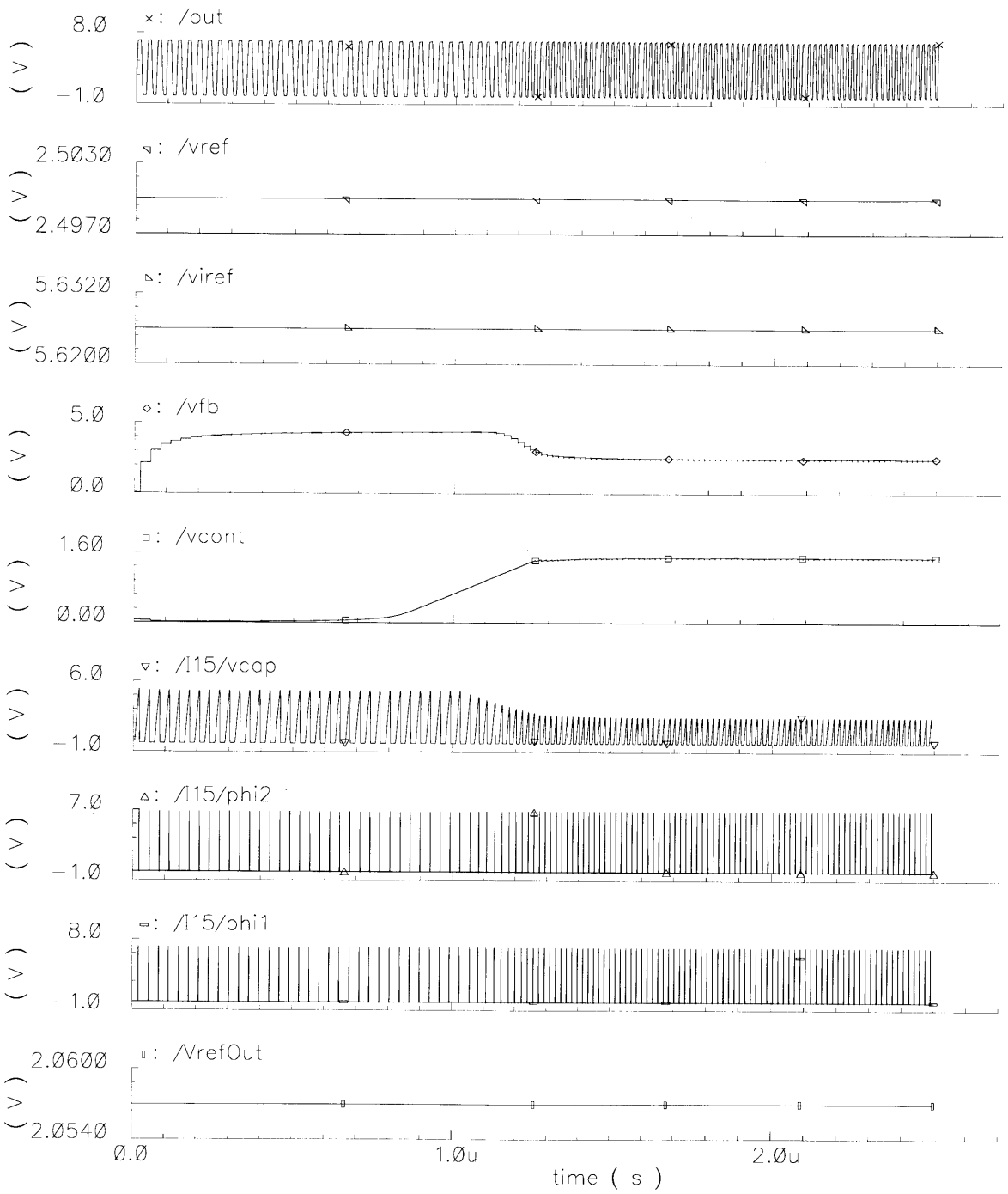


Figure 5.19: Wavforms for the VLRO from a cold start through lock. The system is in lock when vfb equals vref.

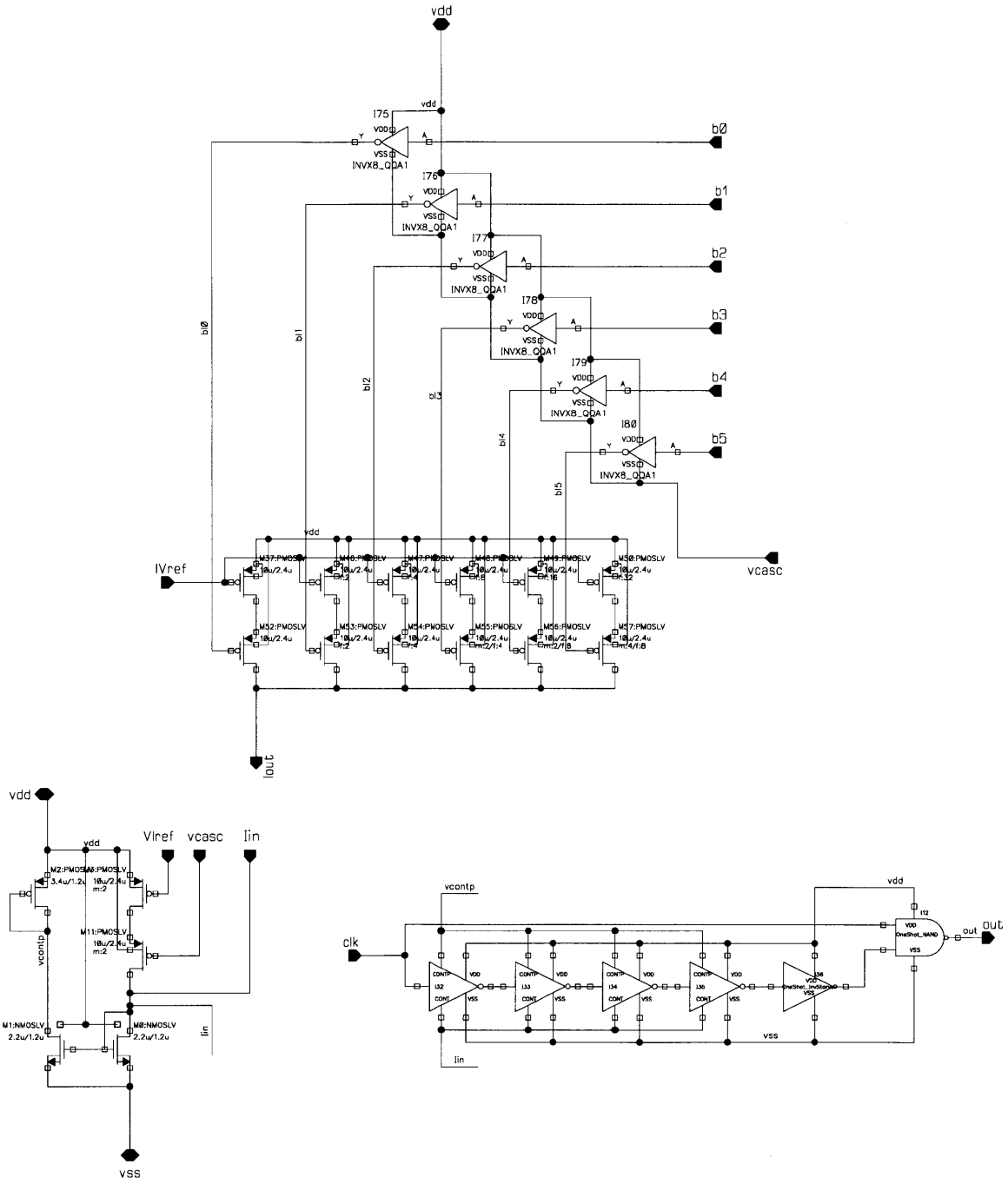


Figure 5.20: The one-shot is a simple pulse generator comprising current-starved inverters that are controlled to vary the pulse width. It is triggered on the positive state change of CLK

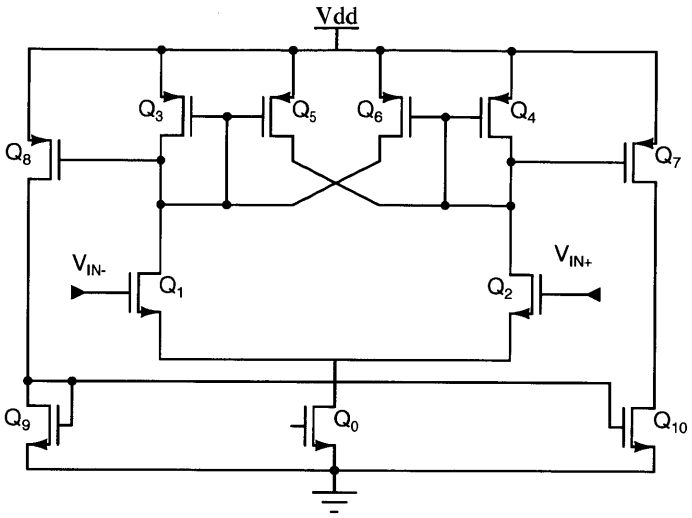


Figure 5.21: Regenerative comparator scheme used in the converter

converter stops running. The load is now supplied by C_{B1} and the output begins to fall. It continues to fall until it drops out of the hysteresis band and the cycle repeats.

The result of this control scheme is that the output is regulated to an average DC voltage equal to $V_{REF}\left(\frac{R2}{R1}\right)$ and the AC ripple on the output is equal to $V_{HYS}\left(\frac{R2}{R1}\right)$. The frequency of the ripple is set by a combination of the desired output power, converter source power, and size of C_{B1}. It is highest when the output power is equal to the converter power. The details of sizing the capacitor and resulting frequency variation can be found in [?].

The comparator has several requirements to effectively implement this scheme. First, some form of hysteresis must be provided for. Without it, the converter will chatter causing severe efficiency penalties. It also needs to sustain a common-mode range somewhat larger than the range of the reference voltage, in this case V_{REF} is intended to sit at 4 V. The propagation delay through the comparator should be short enough that it does not constrain the ripple to be substantially larger than the hysteresis band (it will be if the propagation delay ends up being a significant fraction of a modulation cycle). In this case, the total delay needs to be less than about 200 ns to ensure minimal overshoot. The input offset voltage is not critical for this comparator as it shows up directly as an offset in the converter output voltage and can be corrected with a change in either R_1 or R_2.

In order to satisfy this set of constraints a regenerative comparator with adjustable hysteresis was chosen. It allows the comparator to be implemented entirely on-die which reduced the pin count (a constraint in this system) and eliminated the need for a fast output stage capable of driving off-chip loads. The schematic of a basic regenerative comparator stage is shown in Figure 5.21. The stage comprises an actively loaded differential pair with two extra

devices. These function to provide positive feedback [56]. This can be seen by considering the case when Q_1 is grounded and V_{IN-} is well below ground so that Q_2 is off and Q_1 carries the entire bias current. Under these conditions, Q_5, Q_6, and Q_4 are also off while Q_3 carries the full bias current, as well. If V_{IN-} is then raised slowly, Q_2 begins to conduct stealing some of the bias current. At this point, as V_{IN-} is further increased, Q_5 provides current to Q_2 so that V_{O1} remains high. Once the current in Q_2 equals the current in Q_5 (set by the current in Q_3), any additional increase of V_{IN-} causes V_{O1} to drop which initiates positive feedback that tends to drive the comparator to flip state.

Since Q_5 and Q_6 mirror the currents in Q_3 and Q_4 an opportunity arises to introduce hysteresis by scaling the gain of the mirrors. As noted in the operational description above, the value of V_{IN-} that causes the current in Q_2 to equal that of Q_5 represents the trip point. By increasing the size of Q_5 relative to Q_3, the current in Q_5 increases and correspondingly V_{IN-} must be raised further to reach the trip point. If the W/L ratio of Q_5 to Q_3 is called α, then when $\alpha < 1$ transistors Q_5 and Q_6 simply serve to add gain, when $\alpha = 1$ the system functions as a latch, and when $\alpha > 1$ hysteresis is introduced. As α grows, so does the apparent hysteresis.

Under the conditions listed above, the positive trigger voltage is:

$$V_{TRIG+} = \sqrt{\frac{I_0}{k(W/L)_1}} \cdot \frac{\sqrt{\alpha} - 1}{\sqrt{1+\alpha}} \tag{5.4}$$

And the negative trigger voltage is:

$$V_{TRIG-} = \sqrt{\frac{I_0}{k(W/L)_1}} \cdot \frac{1 - \sqrt{\alpha}}{\sqrt{1+\alpha}} \tag{5.5}$$

The total hysteresis is then found as:

$$V_{HYS} = 2\sqrt{\frac{I_0}{k(W/L)_1}} \cdot \frac{\sqrt{\alpha} - 1}{\sqrt{1+\alpha}} \tag{5.6}$$

Where $\alpha = [(W/L)_5/(W/L)_3] = [(W/L)_6/(W/L)_4]$.

The complete comparator schematic is drawn in Figure 5.22. Three control bits permit 8 steps of hysteresis which varies from about 10 mV to 200 mV over the range of selection

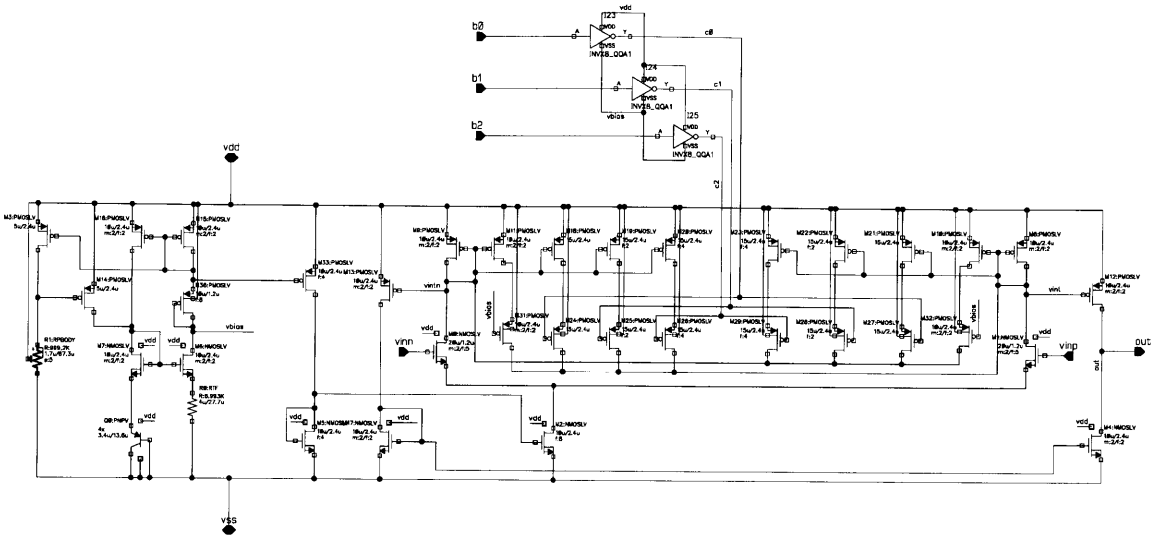


Figure 5.22: Regenerative comparator scheme with adjustable hysteresis used to implement control of the converter.

by changing the effective widths of Q_5 and Q_6. The comparator outputs are mirrored to a single-ended output by transistors Q_7 to Q_{10}. Bias is set by the remaining devices, which form a replica biasing network.

5.3.11 Linear Regulators

Linear regulators are necessary to supply constant rails for the on-die circuits as the input voltage to the converter varies. The circuit used here is a basic design that implements a reference, an error amplifier, and a pass transistor using a combination of drain-extended and medium voltage LDMOS devices. The circuit is shown in Figure 5.23. The pass transistors, Q_0 for the 5-V output and Q_25 for the 7-V output, are drain-extended devices with large enough area that the drop-out voltage remains below a volt at full load. Their gates are driven by actively loaded differential pairs that serve as the error amplifiers. Feedback is taken from a resistor divider at the regulator outputs and fed back to the non-inverting inputs (the negative feedback comes in because driving the PMOS gate of the pass transistor higher tends to shut off the regulator) of their respective error amplifiers. The bias current for the differential pairs and the reference generator formed by Q_2-Q_4 and R_4 comes from a self-biasing cascode which generates a PTAT current.

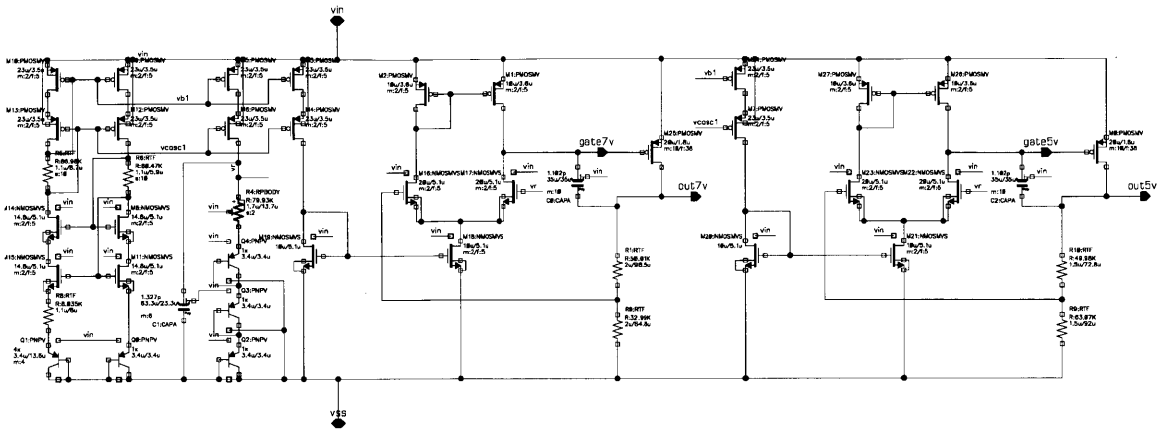


Figure 5.23: Combined 5-V and 7-V linear regulators provide logic and auxiliary analog supply rails.

5.3.12 Isolated Communication

Several options were considered for an isolated communications system, including designs with and without decoder schemes and using physical layer implementations employing magnetic and capacitive isolation. The schematic in Figure 5.24 depicts the chosen implementation. To use the least amount of space on the board, capacitive isolation was chosen. The transmitter works by simply sending a continuous pulse on TXMINUS when the converter should run, and TXPLUS when the converter should stop. This is received by a set of inverters on the receiver side each of which has an RC-filter attached. The RC provides the average voltage of the pulse. When a pulse is present the inverter switches to a set state. When no pulse is present it returns to a reset state. The inverters drive the SET and RESET pins of an asynchronous SR latch such that when TXMINUS is active, the output RxSig is high and the converter runs. When TXPLUS is active, the converter stops. The default state of the latch is determined by level-setting resistors at the inputs of the inverters, which ensure the converter runs at initial power-on.

5.3.13 Ancilliary Systems

In order to make the chip configurable for testing and various operating scenarios a scan chain was implemented. The scan chain, a serial-input, parallel-output set of registers, controls the states of various subsystems on the chip. It comprises a string of D flip-flops that pass data from one to the next as they are synchronously clocked. The resulting register is 23 bits and the function of each bit is detailed in Table 5.3.

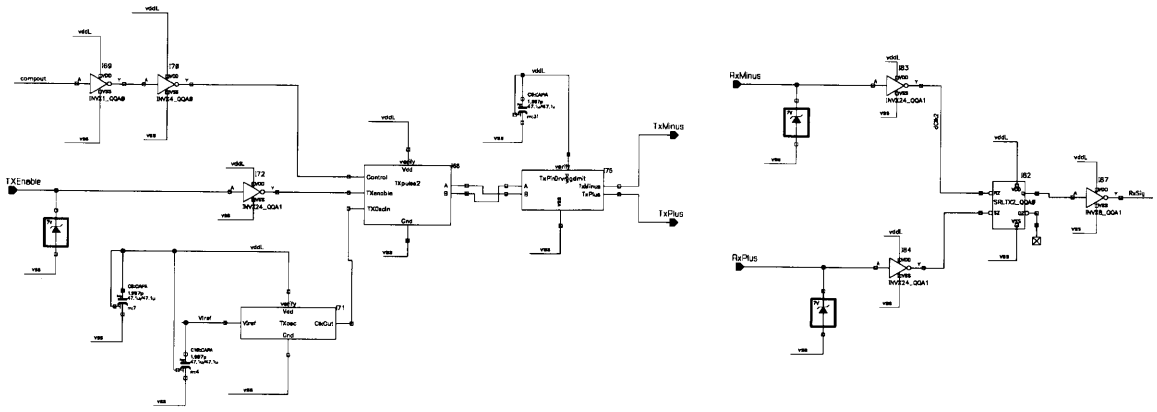


Figure 5.24: The isolated communications system was kept as simple as possible and relies on capacitive isolation. The capacitors are realized as copper features in the PCB substrate.

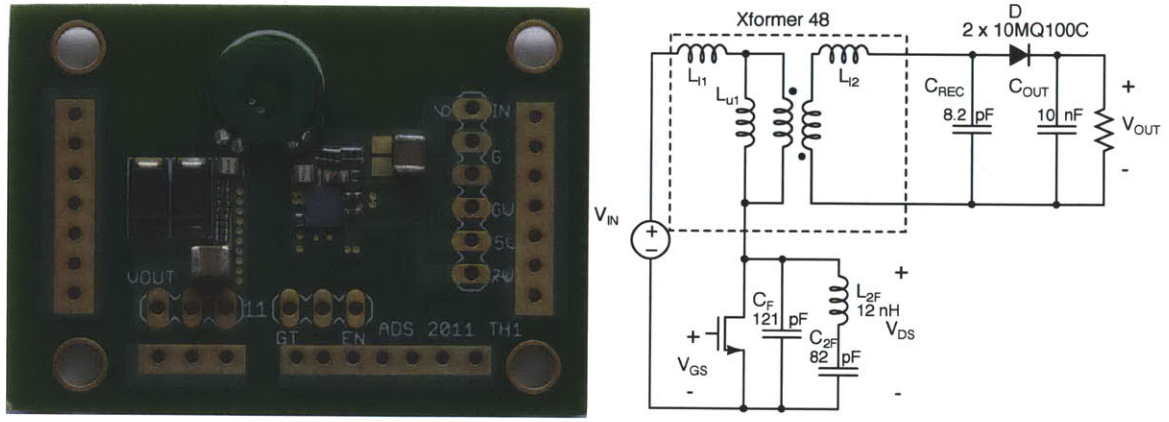
Table 5.3: Register Bits and Description

Bit	Function
b[0:5]	OneShot b[1:6], sets pulse width
b[13]	OneShot b[0], sets pulse width
b[6:11]	FVC b[5:0], sets FVC charging current
b[14:16]	Comparator b[0:2], sets hysteresis
b[17:b23]	IVRef trim current b[6:0], minimizes temp. drift

Unfortunately during the design phase, a serious oversight was made regarding the design of the scan chain. The flip-flops were connected in series Q-D-Q and clocked simultaneously. An unanticipated race condition precluded reliable loading of the register. This had the effect of crippling many aspects of the IC because it was impossible, for instance to adjust the bias point of the reference, or the charging current of the FVC. While most of the subsystems could be demonstrated, operating the complete chip under its own control was not possible. Instead the various pieces of the chip such as the gate driver system, the linear regulators, and the comparator were used together with external circuitry to create a functioning converter. Nevertheless, useful results were produced and follow in Section 5.4.

5.4 Experimental Results

Two different converters were constructed using the IC. The first is an isolated Φ_2 converter similar to the system described in Section 5.2. It operates from 8 V to 16 V input and provides a 12-V output at 6 W nominal power and 73% efficiency. The second is a non-isolated Φ_2 boost converter delivering 14W nominal over a 10 V to 20 V input range with a 33-V output and achieving 85% nominal efficiency. While both converters use the same



(a) Isolated converter photo

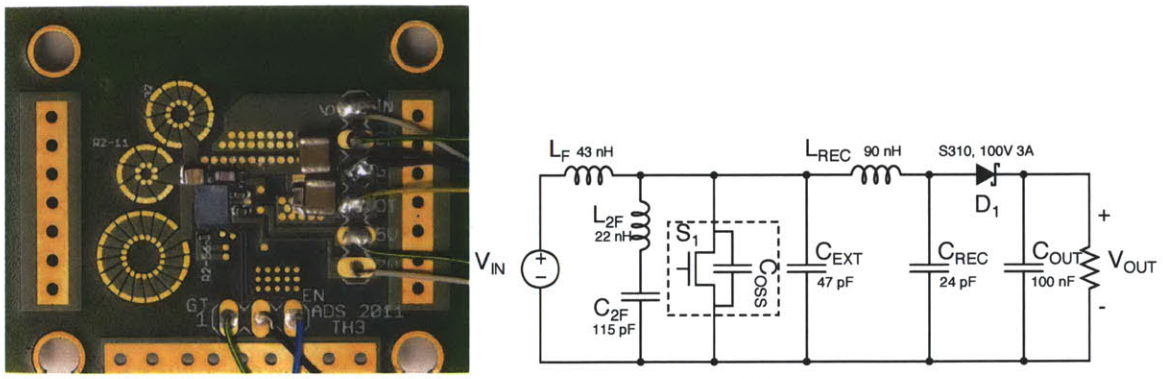
(b) Schematic

Figure 5.25: A photograph of the Isolated Φ_2 converter shows that the inverter is quite small without IC packaging to deal with. The rectifier is still relatively large because of diode packages.

integrated circuit, the gate driver and power device were optimized for use in the isolated design using the techniques outlined in the device chapter.

Figure 5.25 is a picture of the isolated version along with the schematic of the power stage as implemented. The transformer is 7.8 mm in diameter and is constructed on a a separate PCB from the main converter shell. It provides the magnetic energy storage for the resonant circuit with the exception of the second harmonic branch which includes L_{2F} as a discrete inductor mounted on the PCB. The balance of the inverter circuit forms a very compact system, with a total area around 35 mm^2. This includes the resonant energy storage, gate driver components, bypass capacitors, and AC-short capacitors. As compared to the isolated Φ_2 converter from Chapter 4, this is is a dramatic reduction in the inverter area. In that converter which only used a custom discrete transistor rather than an IC, the inverter circuits, IC package, and auxiliary support circuits take up more than 6-times the area. The rectifier on the secondary of the transformer similar in size to the non-integrated isolated Φ_2 converter owing to the fact that it is dominated by the diode packaging, which was not addressed in the integrated version. It requires about 50 mm^2 despite a lower component count. The majority of the PCB is area provided for test and debug. The full details of component values, layout, and dimensions are included in Appendix A.4.

Figure 5.26 is a picture of the non-isolated Φ_2 boost converter and schematic of the power stage. In this design, the resonant inductors were implemented as copper traces in the



(a) Boost converter photo

(b) Schematic

Figure 5.26: The non-isolated Φ_2 boost converter with planar toroidal inductors printed in the PCB and associated schematic.

Table 5.4: Measured Converter Parameters

Parameter	**Isolated Φ_2**	**Φ_2 boost**
V_{IN}	8-16 V	10 - 20 V
V_{OUT}	12.7 V	32 V
P_{OUT}	6 W, nom	14 W, nom
Efficiency	73%, nom	85%, nom
F_{SW}	75 MHz	50 MHz
Ripple	50 mV (0.4%)	100 mV (0.3%)

PCB. [3] The planar toroidal inductors are arranged next to the IC in a manner that minimizes loop area where high frequency currents need to flow. A rectifier diode is mounted on the back of the board along with the capacitors that form the AC-short and C_{2F}. This design achieves substantially higher power using the same IC in part because of a DC path that exists in the boost converter. Table A.1 lists the specifications of both converter designs, the full details including component values, part numbers, layout, and dimensions are included in Appendix A.4.

5.4.1 IC testing

IC bringup was performed on a PCB created to test the various sub systems on the chip while maintaining as much control as possible. The board is pictured in Figure 5.27. The first test provided DC power to the V_{DD} inputs of the chip. This established that no direct

[3]The inductors were designed by a colleague George Hwang using a set of optimization and layout tools he developed.

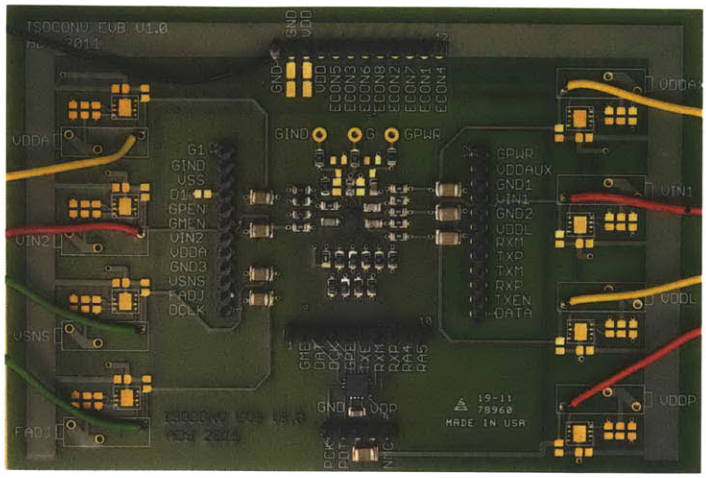


Figure 5.27: The IC mounted on the evaluation PCB used for bringup

shorts existed, as the idle current for the chip (6 mA) was within about 10% of the design value.

The linear regulators were tested subsequently. Both 5-V regulators show output voltages within 5% of the desired values, or 5.1 V and 5.3 V in the first IC tested. The 7-V linear regulator has a quiescent output of 7.1 V. The variation of the output voltage as a function of input voltage is less than 10 mV over the rated input voltage range of 6 V to 20 V. Load regulation is small with the regulator outputs drooping approximately 100 mV between no load and full load for both the 5-V and 7-V regulators. Dynamic testing shows regulation bandwidth on the order of 10 kHz which is the design value.

The next phase of the testing cycle was to test the scan chain. An FPGA was programmed to serially clock a set of bits onto the scan chain. Unfortunately, the this test revealed that the scan chain does not work. It is not possible to update the scan chain with new values over the defaults. This is owing to a design flaw in the scan chain. It was assembled from a series of D flip-flops. During testing it became clear that a race condition exists precluding the ability to load the scan registers properly. Unfortunately, because the scan chain touches so many of the subsystems on the chip, it prevents a full converter system from being implemented using the IC.

The comparator was tested by applying an input voltage ramp and watching the output of the RXPLUS pin. For proper operation RXPLUS should have an approximately 10 MHz pulse when VSNS (the comparator input) is above VREF. When a ramp is applied, the run signal appears on RXPLUS after VSNS exceeds approximately 1.16 V. It shuts off again when VSNS is brought below 1.04 V. This reveals that the hysteresis of the comparator is approximately 120 mV, close to the designed default value (as set by the

default scan-chain values). The reference voltage is about 1.1 V. This is in contrast to the 4 V expected from the design. Dynamic testing shows that the comparator is close to its design speed, with a propagation delay of about 200 ns for a 100-mV overdrive. The adjustable hysteresis function cannot be reliably tested because of the scan chain malfunction, but several attempts at reprogramming caused the hysteresis to vary between about 30 mV and 140 mV on a fairly random basis.

The outputs of the IV reference are not directly accessible because of pin-count limitations owing to the flip-chip design (there are only 30 bumps available in the 2.5 mm x 3 mm die, many of which are used for power devices or hotel power). However, the voltage output of the reference was determined to be low based on the results of the comparator testing. This suggests that either process variation, incorrect bias setting due to the scan chain, or a combination of both is at fault. Repeated attempts to program the scan chain result in the reference voltage shifting between 0 V and about 2.5 V. It is not possible to achieve the 4 V design point, although the reason remains unclear because of the ambiguity in the scan chain values (there's no way to know if all possible bit combinations were achieved). It was possible to verify that the IV reference does have decent temperature performance by heating the chip and re-measuring the comparator trip points. Over a temperature range of approximately room temperature to about 100C (measured with a thermocouple, achieved with a heat gun) the reference voltage drifts about 50 mV when the nominal room-temperature value starts at 1.1 V. This is worse than the intended design by a factor of four, probably due to the fact that current-trimming wasn't possible with a broken scan chain.

Despite sub-optimal reference performance vs. temperature, the internal clock and gate driver seem to function well. The clock runs stably at 83 MHz and successfully drives the on-chip gate driver such that gate drive waveforms appear at the gate of the main power device. Cycling the die from room temperature to 100C yielded frequency drift of about 4%, which is more than desired, but far less than would be the case with a ring oscillator running open loop. The clock frequency cannot be varied over a wide range by changing the voltage on FADJ as was intended by design. Instead, the frequency tops out at around 90 MHz and could not be made to run below about 80 MHz with any kind of reliability. The likely cause is the inability to adjust the FVC charging current to bring it's output voltage characteristics in line with what is required by the op-amp. This is also a function that is supposed to be provided by the scan chain. On the other hand, the fact that the VLRO maintains 83 MHz within 4% over temperature suggests that the loop locks, demonstrating basic functionality. One additional point worth mentioning is that the duty ratio can't be set properly (again due to the scan chain) so late turn-on of the gate driver pull-down device negatively impacts efficiency. When the gate driver is run from the on-chip clock,

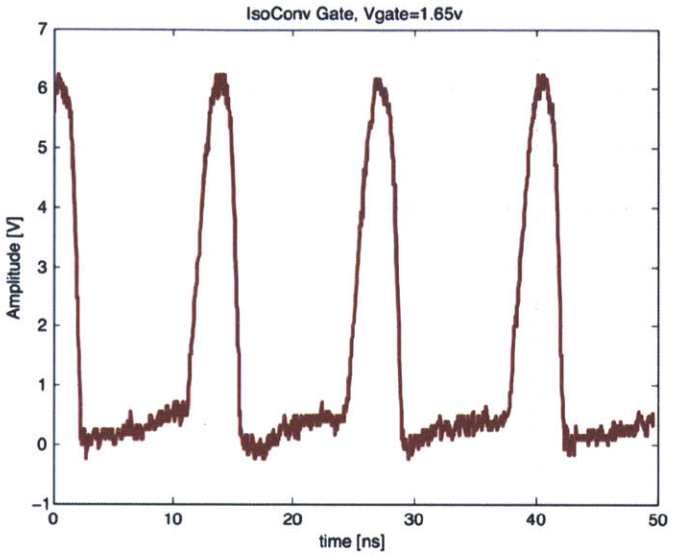


Figure 5.28: Gate drive waveform for properly tuned operation.

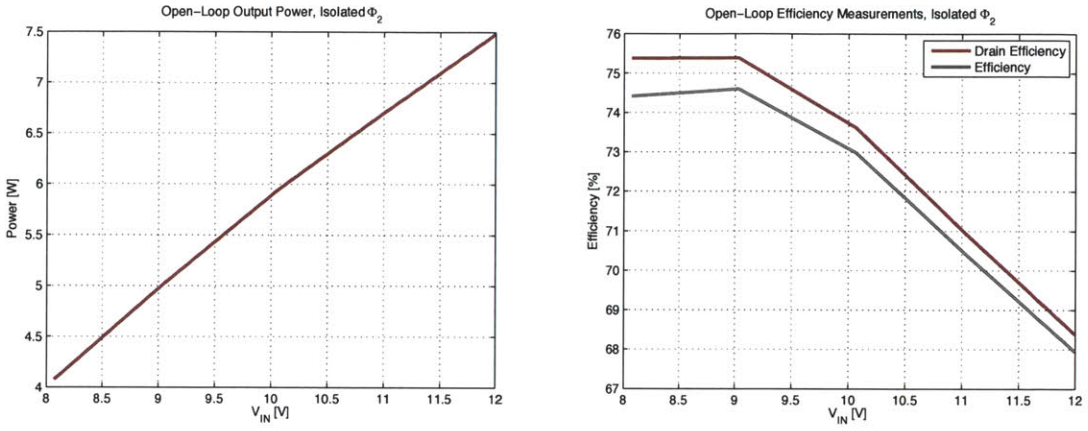
it draws about 140 mW, or more than double what is achieved with off-chip control of the gate driver.

The isolated communications subsystem was tested by connecting the respective TX pins of one chip and RX pins of another with 10 pF capacitors (the design value, intended to be implemented as features in a PCB). The VSNS input on the TX-side chip was than cycled so that the comparator would trip driving the communications bus. On the RX side, watching the clock signal showed that the control passes from one chip to another as expected.

The gate driver was tested with by using an FPGA (a Xilinx Spartan-3E) to generate the required clock signals externally. Both steady-state and startup conditions are achieved with the FPGA handling the runt-pulse generation required of the gate driver. Figure 5.28 shows the gate drive waveform when the system is properly tuned. The gate driver draws 110 mW which is close to the design power and substantially less than the power required for hard gating (320 mW).

5.4.2 Converter Testing

Open loop testing of the converters was accomplished by connecting the devices to zener loads (12 V for the isolated Φ_2 converter and 33 V for the Φ_2 boost converter). The input voltage was swept over the ranges specified Table A.1. Both converters function



(a) Converter output power

(b) Open-loop efficiency

Figure 5.29: Output power and open-loop efficiency for the isolated Φ_2 converter prototype built using the custom IC

as expected. The output power, and efficiency vs. voltage plots are for the isolated Φ_2 converter are shown below in Figures 5.29a and 5.29b. Figure 5.30 shows the drain and gate voltages for the isolated Φ_2 converter displaying ZVS switching characteristics.

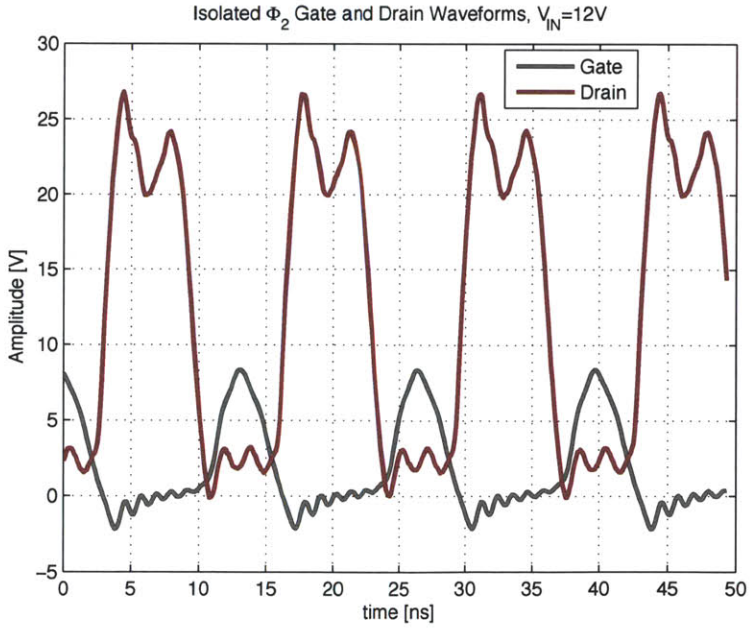
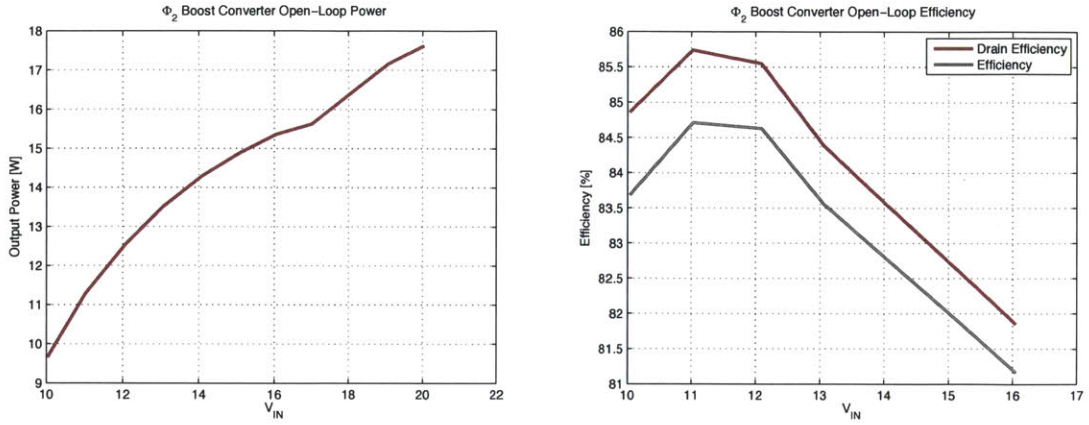


Figure 5.30: Experimental gate and drain voltage waveforms for the isolated Φ_2 converter while operating under full load (6 W).



(a) Converter output power

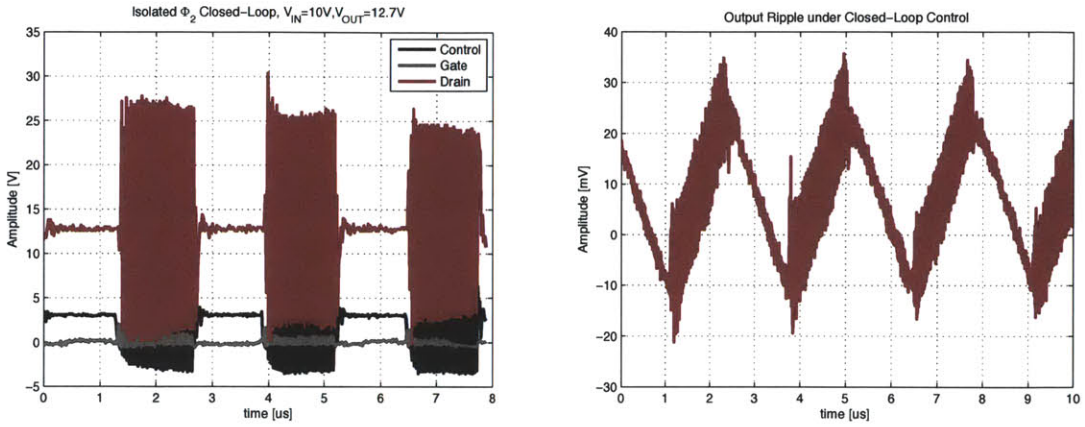
(b) Open-loop efficiency

Figure 5.31: Φ_2 boost converter power and efficiency measurements show the performance to be higher in both aspects owing to the presence of a DC path that reduces resonating currents for a given output power.

The Φ_2 boost converter has significantly higher efficiency and output power owing to the presence of a DC path. This information is captured in Figures 5.31a and 5.31b.

In order to perform closed-loop testing of the isolated Φ_2 converter, external hardware was necessary. While some on-chip functions were close to their design values, it was not possible to configure the scan chain for a set of conditions that simultaneously met the desired operating frequency and allowed effective control. As a result, the FPGA used during gate driver testing was configured to accept a logic-level input to control the modulation of the converter. A hysteretic comparator was created using discrete components and used to run the converter in voltage-mode hysteretic control. An optocoupler was also added to permit control across the isolation barrier. Figure 5.32a shows the converter modulating at approximately 500 kHz to regulate the output voltage to 12.7 V. The converter startup and shutdown during each modulation cycle behave as expected. Figure 5.32b shows the output voltage ripple, which has a peak-to-peak amplitude of 50 mV as set by the comparator. Measuring the effects of modulation showed that over the control range of minimum to maximum load (1W-6W) the modulation reduces efficiency by between 1% and 2%. The gate driver reduces efficiency by another 1% for the tuning point used during the measurements. Figure 5.33 shows the closed-loop efficiency of the isolated Φ_2 converter.

As a final point, it is useful to compare the power density of the isolated Φ_2 converter with similar isolated converters available on the market. The converters chosen for comparison have similar voltage and power levels and are intended to provide auxiliary isolated rails. For the devices compared, the power density of the isolated Φ_2 converter is at least a factor



(a) Closed-loop operation

(b) Output ripple voltage

Figure 5.32: The isolated Φ_2 converter under closed loop modulation above 500 kHz. The resulting output ripple is shown at the right, and is about 0.4% of the 12.7 V output.

of two better. The power density calculation includes a bounding box volume around the transformer of 1 radius (about 4 mm) to account for the fields in the local vicinity of the

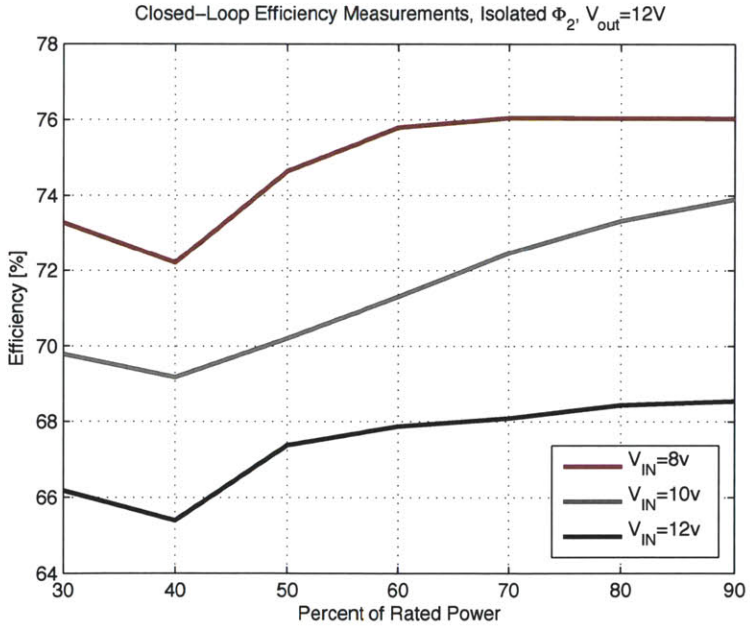


Figure 5.33: Efficiency map of the isolated Φ_2 converter vs. load under closed loop control for various input voltages

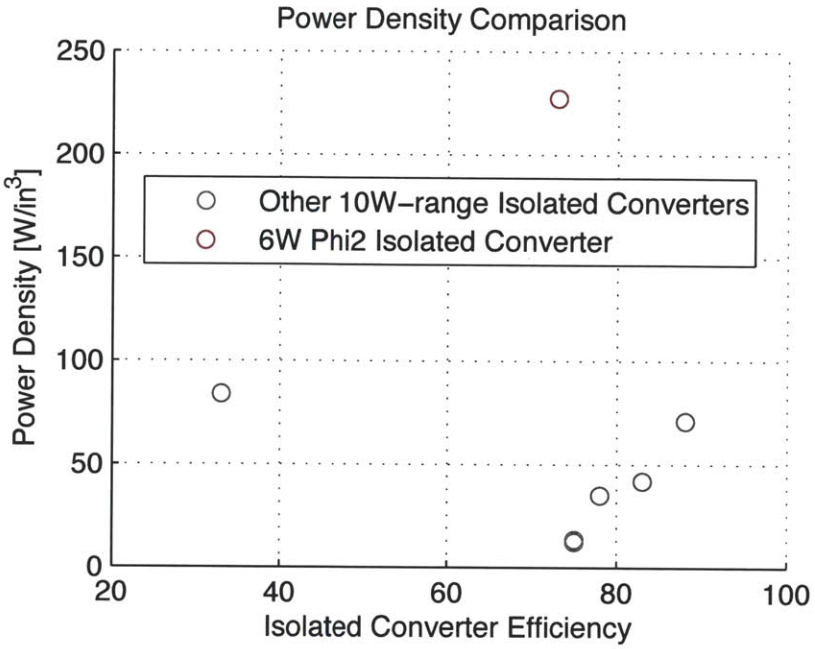


Figure 5.34: A comparison of the power density of various comparable isolated converters with the isolated Φ_2 converter presented in this work.

device. [4]. A plot showing a sampling of converters and their respective power density and efficiencies is included in Figure 5.34.

[4]It was found by experiment and simulation that beyond 1 radius there is very little effect on converter performance even if the converter is placed between two solid copper blocks

Chapter 6

Conclusion

THIS THESIS explores resonant power conversion techniques suitable for VHF dc-dc operation. Through a combination of device, circuit, passive, and systems development, a miniaturized prototype of an isolated Φ_2 converter is demonstrated. By operating at VHF frequencies, the energy storage required to affect conversion is reduced to the point that air-core passives implemented as traces in a PCB may be used. Combined with device optimization performed on an integrated power process, this work sets out a pathway to greater levels of integration that takes the form of a PCB substrate with a VHF PMIC mounted in flip-chip-on-board fashion.

6.1 Thesis summary and key take-aways

Chapter 1 introduces the concept of energy storage as a limiting factor in reducing converter volume. Using conventional techniques to increase frequency can reduce energy storage requirements at the cost of efficiency. By employing resonant switching the frequency-dependent loss mechanisms can be largely skirted. Even with the reduced passive volume enabled by VHF operation, more work is necessary to achieve smaller converters. The power devices must be optimized for VHF operation, new means of synthesizing passives developed, and various circuit and system challenges (such as resonant gate drivers that take advantage of integrated switched) need to be tackled. This work addresses these challenges to achieve a more-integrated converter, but stops short of putting passives onto the die. The latter is a goal that remains, for the moment, out of reach at the voltage and power levels considered here.

Chapter 2 discusses techniques for optimizing integrated LDMOS devices for use at VHF. Typical integrated power process LDMOS devices are intended for hard switching. The losses in this scenario are well understood. Overlap losses due to non-zero commutation time in a given circuit are proportional to frequency over a wide range (until the frequency exceeds the threshold of non-quasistatic operation when additional losses grow in super-linear fashion). The balance of the semiconductor losses comprise capacitive discharge loss and gating loss. These mechanisms are $\propto CV^2f$, again growing linearly in frequency. In

a soft-switched converter with resonant gating, circulating currents replace RC-charging currents. The upshot is that frequency-dependent loss is proportional to $C^2 f_{SW}^2 R$. This difference tends to drive divergence between the ideal device for hard switching and the ideal device for soft switching. By defining the losses with Equation 2.1, a goal function can be derived in the context of a power converter design that allows for layout optimization.

Since the optimization on geometry is essentially an inverse problem, the time consumed for calculating the correct geometry in an unconstrained system can be substantial. As a result, it's necessary to first pick an overarching geometry framework that defines the basics of the device, such as the LDMOS layout chosen in this work, then determine the appropriate subset of geometric parameters to vary in order to satisfy the goal function. Doing this well amounts to picking a parameter set where each element has a significant effect on one or more of the VHF device-loss parameters (i.e. conduction, gating, and/or displacement losses). Performing the computer optimization on a previously hand-optimized layout yielded a 53% loss reduction, highlighting the significance of simple layout changes alone.

A further avenue for improving device performance at VHF derives from the use of SOA extension. SOA extension is possible because the switching trajectory in the resonant converters presented here closely tracks the voltage and current axes. This results in a relative absence of hot carrier effects that normally constrain peak operating voltage. As a result, devices may be used at voltages closer to their actual breakdown voltage rather than that set by the hot carrier limit. The reduced specific capacitance results in a significant reduction in device loss and higher converter efficiency. Testing to evaluate hot-carrier effects showed very little movement on both R_{DS} and V_T which are traditionally used to test LDMOS devices. In this work, a device with a 42-V breakdown is substituted for one with an 85-V breakdown and demonstrated to operate successfully for over 1000-hours. When the 42-V device is optimized using the same layout optimization techniques, the total energy savings over the hand-optimized case is about 74%.

Chapter 3 examines the challenge of synthesizing passive magnetic components for VHF resonant power converters. Creating passives that match size and form-factor is key in taking full advantage of the effort taken to operate in the VHF regime. By printing structures in the PCB and absorbing energy storage into the parasitics, the total size of the energy storage may be reduced. This chapter presents a two-winding planar transformer as an example. The transformer is implemented in a planar form factor where each winding comprises one or more turns in the PCB substrate. By using the transformer parasitic inductances to stand in as the bulk of the energy storage, both a reduction in component count and an increase in density are possible. In order to accomplish this, a transformer with a fully-constrained inductance matrix is necessary. As a second inverse synthesis prob-

lem, this presents a significant challenge. Typical transformers use magnetic materials to constrain flux to well-defined paths. This permits simplification of the models and optimization of closed-form results. In the air-core case, which is commonly employed at VHF, no such simplification is possible. In order to calculate the losses the detailed flux and current patterns need to be accounted for to sort out skin and proximity effects in the windings. As a result, it's necessary to rely on a fields solution.

Since any number of physical transformers can represent a single inductance matrix, finding the optimum transformer is tantamount to establishing the tradeoff between volume and efficiency and picking the point that best satisfies the external system constraints. Determining the volume-efficiency tradeoff requires looking at many designs, and is prohibitively time consuming to do as a fields solution alone. However, in the case of the planar ring transformer, closed-form expressions are available to compute the inductance matrix of each design. This allows a rapid search through the design space to establish the locus of transformer solutions that meet the inductance matrix requirements. Once this locus is identified, the edge of the area (bottom left edge on an efficiency-size plot) is the volume-efficiency tradeoff. At this point picking an optimal design can be accomplished based on the requirements of the external system.

In terms of possible planar designs, it was discovered that better performance (as compared to spiral windings) was possible when each turn of each winding was kept to a separate layer on the PCB. This results because less flux is forced to impinge on the copper windings to cause eddy current losses. In a similar vein, the fully-constrained nature of the inductance matrix causes some interesting design results. First, while it's generally held in transformer design that higher coupling can yield better power transfer and efficiency, these designs diverge from those results. Higher coupling is achieved by using boards with thinner substrates, and it was found that intermediate FR-4 substrate thicknesses yielded the best results. This is because when the windings are sized to support the required self inductance and then placed close together on a thin-substrated board, often the coupling is too high and the balance of leakage and mutual inductance is upset.

In order to reduce the coupling, the copper traces of some turns can be widened. This places them in the flux path and impinging flux sets of eddy currents that result in the desired inductance matrix. However, the eddy currents correspond to extra loss, an undesired effect. Since another way to reduce the coupling that does not involve additional eddy currents is to space the windings at greater distances, thicker substrates can help improve the efficiency. Of course, if the windings are spaced at too great a distance the desired coupling is either not achieved or comes with greater loss. Thus for the range of substrate thicknesses tested (20-mil to 125-mil) the best performance was found on a 31-mil substrate. For 75-MHz designs, transformer efficiencies of 94% were achieved at a power density of 94 kW/in^3.

In order to reduce the search space, the turns of each coil were constrained to have identical trace width and diameter for many of the synthesis runs. This greatly reduces time to solution at the possible expense of slightly lower efficiency. For comparison purposes a design where the trace widths and diameters were allowed to vary independently was performed. A 1% increase in efficiency was achieved, but at much greater computational effort (more than two order of magnitude more compute time).

Chapter 4 discusses a new isolated Φ_2 topology that uses the transformer from Chapter 3. It is a topology that incorporates a transformer both to provide galvanic isolation as well as energy storage. This results in a reduced component count that can facilitate the realization of more-integrated converter form factors. The converter benefits from several tuning considerations. While the best efficiency can be achieved by reducing the amount of reactive current flowing in the system, the input voltage range is severely curtailed. Therefore a better tradeoff can be established by sacrificing a few percentage points to increase the range. This is accomplished by adding some series reactance to the rectifier loop, which is amounts to increasing the transformer leakage inductance.

One of the difficulties in physically implementing this converter design arises in terms of the primary-side loop that forms the inverter circuit. Parasitic loop inductance forms a voltage divider that reduces power transfer substantially without a concomitant reduction in circulating currents. This drops power and efficiency in the converter simultaneously. To achieve better results, the parasitic inductance can be offset by reducing the total leakage for the particular transformer design.

Another area of difficulty arose in the selection of the Schottky diodes needed on the rectifier side. At the operating point, much higher loss was observed than expected from the diode models. Experimental testing revealed that the problem could be mitigated by choosing the lowest voltage diodes possible for the particular design and paralleling diodes to reduce current density. Part of the problem seems to be associated with non-quasistatic conduction that yields higher loss above some current density. Several diode combinations are tested against power levels to evaluate the performance and the best combination was chosen to implement a converter design that ultimately achieved about 73% efficiency.

Chapter 5 draws together pieces from each previous chapter–device optimization, circuit design, and passive components–and synthesizes them into a complete converter system that utilizes an IC. The IC absorbs subsystems which would otherwise consume substantial space relative to the power stage. A flip-chip-on-board construction is used. The elimination of bondwires aids in reducing parasitic resistance and inductance. The latter is important both to the gate driver and overall tuning of the system. For the gate driver, discrete implementations of the half-sine system proved impractical because loop inductance between

the main power gate and hold-down device compromises functionality. With both devices on the same die, the inductance can be reduced greatly, and a practical driver was achieved. Similarly, in the case of tuning point, with very small inductance around the C_{EXT} loop owing to the substitution of bumps for bondwires, more freedom is available to choose tuning points with high efficiency.

The combination of integrated control, gate driver, and power device; printed passives, and relatively low component count offers attractive power density. One of the key challenges associated with this type of implementation however, is avoiding PCBs with difficult requirements, such as very small vias, blind and buried vias, or extremely narrow trace and space rules. Not only does this drive cost up, but it also reduces manufacturing yield. On the up side, once a design has been validated, component tolerances are tight and the tuning point is well maintained.

6.2 Thesis Conclusions

The use of fully resonant power circuits in the context of architectures that separate energy storage and control provides a viable means to miniaturization. The fact that integrated LDMOS devices can be made to run 1 to 2 orders of magnitude faster than intended in their original design targets aids greatly in this regard. Not only does integration allow for shrinking the hotel and control circuits, but it also allows customization of the power device for the particular circuit design. This is useful where both performance and cost are concerned. Generally speaking, there is a bathtub curve that describes the power device loss as a function of gate width within a given VHF converter. At the low end of width conduction losses dominate, at the high end of the scale frequency-dependent losses dominate. However, for practical VHF switching frequencies, the center of the curve is relatively flat. This means that converters at various tuning points can be made to work well on a single switch. It also allows trimming of the device toward the small end of the width scale in the event that cost reduction is desired (eg., for margin-sensitive applications). Further, the suite of auxiliary devices available on the same substrate as the power device permits construction of exceptionally good gate drivers. Typically a discrete resonant gate drive design suffers from parasitic inductance around the gate loop. With on-die pull down switches this problem is eliminated paving the way to higher performance gating. Addressing device considerations (or for that matter passive constraints) alone does not yield the smallest result. In order to reach a level of integration where significant improvement over state of the art is necessary, passives, devices, and circuits must be simultaneously addressed. In this work passive design was undertaken that provides both galvanic isolation and energy storage. The planar-winding PCB transformers that resulted

provide good performance and high power density. When all the aspects of miniaturization are considered together, a significant size reduction is achieved.

6.3 Future Work

There is a great deal of room for future exploration of the VHF power converter space. The bare essentials of a power conversion technology have been laid down in this work as well as those cited. More work is needed to extend the capabilities of VHF circuits into other regimes of power conversion. For instance, to date no VHF synchronous rectification has been accomplished. Achieving this is a means to extend the low-end of the output voltage spectrum. From the perspective of solving the problem, the granular control afforded by merging all control functions onto a single silicon substrate is ideal. One of the chief problems with synchronous rectification is the control of the main and rectifier switches relative to one another. In addition, a synchronous rectifier is also the gateway to topologies that permit bidirectional power flow. This has numerous uses especially where the output voltage is desired to be slewed with very high bandwidth. This would be the case in an envelope tracking system for power amplifier applications. It remains to be seen whether the penalty of increased circulating currents over a high peak to average ratio signal will permit such an application to be feasible. There is also the area of line-connected applications. For the moment, there are no switches that permit convenient connection to the AC line and VHF operation. When a 700-V switch is needed, the parasitics are too large to permit extreme high frequency operation. New material systems offer a good deal of promise in this regard. For the converters presented here, frequency-dependent semiconductor loss is dependent on $C^2 \cdot R$ terms. A GaN-on-Si device has specific capacitance that's better than an order of magnitude lower than its Si-only counterpart. This translates into a 100-fold reduction in loss in the VHF regime, offering tremendous opportunity. On the other hand, the flexibility of small, cell-based architectures can be explored to enable enhanced converter performance in the silicon-only space, for instance permitting off-line operation with lower voltage silicon devices extending the reach of the ever-virtuous silicon materials system.

In the device space in particular, a great deal of improvement is possible. The typical integrated LDMOS power device is optimized explicitly for the hard-switching case, and for relatively low frequency. As a result, capacitance is typically not important. Instead, the focus on specific on resistance reveals preoccupation with minimizing silicon area for cost reasons. In the VHF space, this makes little sense as the result is very often higher capacitance per unit resistance. Particularly troubling is the tendency to introduce additional reverse-transfer capacitance that interferes with VHF gating. As we demonstrated here, small changes can go a long way when working so far from the optimum. By simply

adjusting layout, device performance was greatly improved. By going one step further and changing diffusion dosage and energy, devices tailored for VHF operation should be capable of yet lower loss densities. One particular area deserving focus in this regard is the ESR of C_{OSS} which is irrelevant in the hard-switched case. In some LDMOS devices, this parameter can be inordinately high, which makes VHF operation at high efficiency more difficult. However, by accommodating this as a design parameter in the device development phase, significant opportunity arises for better performance. Ultimately, a full device design from the ground up using VHF loss criteria as a goal function to minimize device loss will provide the best silicon performance.

There is also a significant amount of work that could be done with passives. The passives explored here were effectively 2-dimensional planar structures. While the transformers were made to perform well, a 3D structure should perform better. Designs with low-permeability magnetic materials are also worth exploring further. With a relative permeability of around 4, and decent loss behavior through about 50 MHz, there should be room for some good designs. Moving to on-die passives that are cost-effective will be a challenge. Not only do they tend to take a lot of silicon area, but they are also lossy. This ultimate goal is certainly worth exploring. Perhaps another iteration on converter architectures will be necessary before on-die passives make sense - microstrips anyone?

Ultimately the relevance of VHF power converter systems will be determined by a combination of materials, device, passives and systems development. Their consideration in synergy must provide both reduced cost and enhanced performance over a wide segment of the power converter space to justify the design effort and investment necessary to establish a new technology base. Another challenge is that VHF converter systems have thus far not achieved the same wide operating point range as their more classical brethren. Either this must be overcome by circuit and systems design or a by harnessing batch manufacturing (the promise of small) to the point that holding many versions in inventory makes economic sense. Along those lines, a fair amount of work in modeling and design of VHF-type systems to reduce the level of design effort for any given converter is a significant need. No rigorous, closed-form mathematical model of the Φ_2 converter exists at the time this writing. Having one would not only speed the design of a given converter, but allow for full exploration of the potential of silicon and other material systems to work in the VHF regime.

Appendix A

A.1 Device Optimization Code

```
execute_sim = 0;

if execute_sim == 1
    clear all;
    clc;
end

% Wcell = 25
% Wm1g = 2.9;
% aspect_ratio = [0.5 1 1.5 2 2.5 3 3.5 4 4.5 5]
%
% cell_pitch = 15.3;
% cell_height = Wcell + 9.5 + (2.9 - Wm1g);
% a_k = cell_pitch/cell_height;
% AR = a_k*(columns/rows);
%
% total_cells = Weff/(2*Wcell);
% rows = round(sqrt((k/aspect_ratio)*total_cells));
% columns = round(sqrt((aspect_ratio/k)*total_cells));



% parameters
test = 0;
cell_pitch = 15.3;
Weff = 2*25*12*120;
aspect_ratio = [0.5 1 1.5 2 2.5 3 3.5 4 4.5 5];
limit_wcell = 105;
%limit_wcell = 20;
limit_wm1g = 5.7;
limit_aspect = 10;
%limit_wcell = 1*25;
%limit_wm1g = 2.9;
%limit_aspect = 1;
```

```
tot_count = 1;
execute_gate = 1;

for Wm1g = 2.9:2.8:limit_wm1g
    %[rstub] = stub_resistance(Wm1g);

    for Wcell = 25:10:limit_wcell
        total_cells = Weff/(2*Wcell);
        cell_height = Wcell + 9.5 + (Wm1g - 2.9);
        a_k = cell_pitch/cell_height;
        for n = 1:1:limit_aspect
            rows = round(sqrt((a_k/aspect_ratio(n))*total_cells));
            columns = round(sqrt((aspect_ratio(n)/a_k)*total_cells));
            actual_Weff = rows*columns*2*Wcell;
            actual_AR = a_k*(columns/rows);
            [Lstub,Wm2d,Wm2s,m3_geom_s,m3_geom_d,segments]=
            ...dimensions(rows,columns,Wcell,Wm1g);
            [rcount,ccount] = size(m3_geom_s);

            for m = 1:1:rcount
                trials{tot_count,1} = tot_count;
                trials{tot_count,2} = execute_gate;
                trials{tot_count,3} = rows;
                trials{tot_count,4} = columns;
                trials{tot_count,5} = Wcell;
                trials{tot_count,6} = Wm1g;
                trials{tot_count,7} = Lstub;
                trials{tot_count,8} = Wm2d;
                trials{tot_count,9} = Wm2s;
                trials{tot_count,10} = m3_geom_s(m,:);
                trials{tot_count,11} = m3_geom_d(m,:);
                trials{tot_count,12} = segments(m);

                tot_count = tot_count + 1;
            end
            execute_gate = execute_gate + 1;
        end
    end
end
total_runs = tot_count - 1

if execute_sim == 1
% %%%%%%% Execute code to compute results based on runs set up above

%%%%%% initialize the run

start_run = 1;
%stop_run = 2798;
stop_run = total_runs;
```

```
execute_flag = 0;
run_number = 1
gate_run_check = trials{start_run,2}(1) - 1;
max_bonds = 3;
if start_run == 1
    index_g = 1;
else
    index_g = max_bonds*(start_run-1) + 1;
end
results_count = 1;
gate_runs = 0;
%%%%%%%%%%


%%%%%%% enter main loop
for k = start_run:1:stop_run
    index = trials{k,1}(1);
    execute_gate = trials{k,2}(1);
    rows = trials{k,3}(1);
    columns_tot = trials{k,4}(1);
    Wcell = trials{k,5}(1);
    Wm1g = trials{k,6}(1);
    Lstub = trials{k,7}(1);
    Wm2d = trials{k,8}(1);
    Wm2s = trials{k,9}(1);
    m3_geom_s = trials{k,10}(:);
    m3_geom_d = trials{k,11}(:);
    segments = trials{k,12}(1);

   %Compute the drain-source resistance and Coss

   [source_contacts, source_vias, rmetal_s] =
   ...metal1_source(Wcell, Wm2s, execute_flag);
   [total_contacts_drain, total_vias1_drain, rmetal_d] =
   ...metal1_drain(Wcell, Wm2d, Wm1g, execute_flag);
   top_cells = m3_geom_s(1);
   bottom_cells = m3_geom_s(rows);
   columns = round(columns_tot/segments);
   [Rds_device, Rbond, number_bondpads, Ross, Coss] =
   ...drain_source_resistance(rows, columns, columns_tot, m3_geom_s,
   ...m3_geom_d, top_cells, bottom_cells, segments, Wcell, Wm1g,
   ...Wm2s, Wm2d, total_vias1_drain,source_vias,total_contacts_drain,source_contacts,test);
   Rds_total = Rds_device/segments + Rbond;
   %%%%%%%%%%%%% PLACEHOLDERS!!!!!!!!!! %%%%%%%%%%%%
   %Coss = 10;
   %Ross = 10;
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   %Compute the gate resistance and Rgate
```

```
if execute_gate > gate_run_check
    [rcell, ccell] = cell_impedance(Wcell);
    [rstub] = stub_resistance(Wm1g);
    number_bonds = 1;
    max_bonds = 3;

    for number_bonds = 1:1:max_bonds
       [rgate(number_bonds),cgate(number_bonds)] =
       ...gate_resistance(rows, columns_tot, Wcell, number_bonds, rcell,
       ...ccell, rstub, Wm1g);
       %rgate_mat(number_bonds) = rgate;
       %cgate_mat(number_bonds) = cgate;
       gate_runs = gate_runs + 1;
    end
    gate_run_check = execute_gate;
end

%%%% populate a new matrix called results, and include the gate results

for count = 1:1:max_bonds
    results{results_count,1} = index_g;
    results{results_count,2} = index;
    results{results_count,3} = Rds_total;
    results{results_count,4} = Rbond;
    results{results_count,5} = Coss;
    results{results_count,6} = rgate(count);
    results{results_count,7} = cgate(count);
    results{results_count,8} = Ross;
    results{results_count,9} = rows;
    results{results_count,10} = columns_tot;
    results{results_count,11} = count;  %%%% max_bond_outs
    results{results_count,12} = Wcell;
    results{results_count,13} = Wm1g;
    results{results_count,14} = m3_geom_s;

    index_g = index_g + 1;
    results_count = results_count+1;
end



if k < stop_run
    %if rem(run_number,100) == 0
        run_number = run_number + 1
    %else
        %run_number = run_number + 1;
   % end
end
```

```
   if rem(results_count,25) == 0
      save calculated_results.mat results
   end

end

save calculated_results.mat results

end


function [Rcell, Ccell] = cell_impedance(Wcell)

Rpoly =32;
Eox = 3.1*8.85E-12;
Tox = 318E-10;
Tfox = 4500E-10;
Cox = Eox/Tox;
Cfox = Eox/Tfox;


%Cell Params
f = 50e6;
Wpoly = 4.1e-6;
Wgate_act=2.5e-6;
Lpoly = Wcell*1e-6;  %25e-6;
Wpoly_stub = 12.3e-6;
Lpoly_stub = 4.75e-6;
Wcomp = 9.1e-6;
Lcomp = 2.5e-6;
w0 = f*2*3.14159;

% get segment resistance and capacitance

sample_num = 10000;  %number of segments to break poly section into
Rseg = Rpoly*((Lpoly/(sample_num-1))/(Wpoly*2)); %segment resistance
c_gate_act = 2*Lpoly*Wgate_act*Cox; %capacitance under the active region of the poly strips
c_gate_fox = 2*Lpoly*(Wpoly-Wgate_act)*Cfox; %capacitance under the field oxide region of the poly strips
c_gate_stub_act = 2*Lcomp*Wcomp*Cox; %capacitance under the active region of both ends of the cell
c_gate_stub_fox = 2*Wpoly_stub*(Lpoly_stub-Lcomp)*Cfox; %cap under fox region of both ends of cell
Cseg = (c_gate_act+c_gate_fox)/sample_num; %segment capacitance under long sections of poly
Zseg = complex(0,-1/(Cseg*w0)); % vertical segment impedance (just the segment capacitance)
Zstart = complex(Rseg,-1/(Cseg*w0)); %impedance of end RC of ladder (the first impedance we start with
Ztot = Zstart;

for m = 1:1:sample_num-1 %chop one of sample number to account for starting with first RC
    if m == sample_num-1 %don't add Rseg after the last element of the ladder...the poly is directly abuttin
        Ztot = (Ztot*Zseg)/(Ztot+Zseg);
    else
```

```
        Ztot = (Ztot*Zseg)/(Ztot+Zseg)+Rseg;
    end
end

Zfinal = Ztot;
Rcell = real(Zfinal);   % cell resistance is real part of impedance

k_ccell = 1.22485;  % fudge factor to get right gate capacitance
Ccell = (-1/(w0*imag(Zfinal))+c_gate_stub_act+c_gate_stub_fox)*k_ccell; %capacitance is real part plus the s
%%% note: actual capacitance is off by a factor because fringing efffects
%%% weren't accounted for

end


%%% Calculate all the basic dimensions and geometry



function [Lstub,Wm2d,Wm2s,m3_geom_s,m3_geom_d,segments]=dimensions(rows,columns,Wcell,Wm1g)

%%%% Parameters
cell_pitch = 15.3;
cell_height = Wcell + 9.5 + (Wm1g-2.9);
cont_poly = 0.5;
cont_poly_center = 1.2;
cont_cont_center = 1.4;
leg_width = 4.1;
Gpoly = 1/32;
Gshort = 10000;
contact_node = 1;
leg_node = 2;
leg_width = 4.1;
cont_size = 0.7;
cont_poly = 0.5;
cont_met1 = 0.4;
cont_metal1_nsc = 0.7;
cont_cont = 0.7;
cont_cont_center = 1.4;
via_size = 0.7;
via1_met1 = 0.4;
via1_met2 = 0.4;
via1_via1 = 0.7;
via1_via1_center = 1.4;

met1_met1_20 = 1.5;
met1_met1 = 0.8;
met2_met2_20 = 1.5;
cell_cell_min_v = 9.5;
gate_metal_min = 2.9;
```

```
%%%%% Calculate length of poly stub on gate.  This is fixed, until the metal1
%%%%% gate runners are increased beyond thier current minimum width, 2.9u

Lstub = 4.75 + (2.9 - Wm1g);  % simple,hehe.  4.75 is the current stub width, 2.9 is current metal1 width


%%%%% Calculate metal2 widths given metal1, and cell width

Wcell_int = cell_cell_min_v + (Wm1g - gate_metal_min);  % the cell-cell internal distance, vertically
Wm2_tot = Wcell+ Wcell_int - 2*met2_met2_20; %total width available for metal2 drain and source lines
min_met1_met2_ovp = via_size + via1_met1 + via1_met2; %minimum overlap of metal1 and metal2 to cover a via (

if Wm1g < 20  % design rules change on 20um metal1
    Wm2d_min = min_met1_met2_ovp + 0.5*gate_metal_min + met1_met1; %minimum met2 drain strip width
else
    Wm2d_min = min_met1_met2_ovp + 0.5*gate_metal_min + met1_met1_20; %minimum met2 drain strip width, wide
end

k_m2ds = 0.04761907; %% metal2 drain-source scale factor. k==0, equal, k==1, all drain metal, starting value
Wm2d = (1+k_m2ds)*0.5*Wm2_tot; %width of metal-2 drain strip
Wm2s = (1-k_m2ds)*0.5*Wm2_tot; %width of metal2 source strip




%%%%%%%% Create the metal3 geometry

%find all two-number sums that divide evenly into columns
match_count = 1;
finish_flag = 0;
remainder = 0;
stop_flag = 0;

while stop_flag == 0

    for top = 1:1:columns
        for bottom = 1:1:columns
            if rem(columns,(top+bottom)) == remainder
                cell_cases(match_count,1) = top;
                cell_cases(match_count,2) = bottom;
                match_count = match_count +1;
            end
        end
    end


    % remove duplicate cases and limit the total sum to some number
```

```
    test_count = 1;
    sum_limit = round(columns/2);
    [row_cell,col_cell] = size(cell_cases);
    for m = 1:1:row_cell
        if (cell_cases(m,1) + cell_cases(m,2)) <= sum_limit
            if cell_cases(m,1) <= cell_cases(m,2)
                test_cases(test_count,:) = cell_cases(m,:);
                segments(test_count) = columns/(test_cases(test_count,1)+test_cases(test_count,2));
                test_count = test_count + 1;
            end
        end
    end


      if test_count == 1
          clear test_cases
          clear segments
          remainder = remainder + 2;
      else
          stop_flag = 1;
      end
end

[row_test,col_test] = size(test_cases);

for row_count = 1:1:row_test
    top_cells = test_cases(row_count,1);
    bottom_cells = test_cases(row_count,2);
    slope = (cell_pitch/cell_height)*((bottom_cells-top_cells)/(rows-1)); %slope defined by top-bottom cell

    for k = 1:1:rows
        if k == 1  %set the top cell count
            m3_geom_s(row_count,k) = top_cells;
        elseif k == rows %set the bottom cell count
            m3_geom_s(row_count,k) = bottom_cells;
            m3_geom_s(row_count,k+1) = bottom_cells;
        else
            test = slope*(k-1)*cell_height + (top_cells-1)*cell_pitch; %location of ideal line at row of int
            cell_loc = 0;
            old_distance = cell_pitch*columns;  %initialize old distance with maximum distance of part
            for count_cell = 1:1:bottom_cells   %find cell closest to line and put it's number in the matrix
                cell_loc = (count_cell-1)*cell_pitch;
                distance = abs(test - cell_loc);
                if count_cell <= top_cells  % fill all cells up to at least 'top_cells' in case there is no
                    m3_geom_s(row_count,k) = m3_geom_s(row_count,k-1);
                    old_distance = distance;
                elseif (distance <= old_distance)   %put the closest cell into the slot
                    m3_geom_s(row_count,k) = count_cell;
                    old_distance = distance;
```

```
                end
            end
        end
    end
end

%m3_geom_d = m3_geom_s;
%m3_geom_d(:,14) = m3_geom_d(:,13); %build the drain matrix...it requires one more column

[rck,cck] = size(m3_geom_s);
m3_geom_d = zeros(rck,cck+1);
for m = 1:1:rck
    for n = 1:1:cck
        if n == 1
            m3_geom_d(m,n) = m3_geom_s(m,(cck-(n-1)));
            m3_geom_d(m,n+1) = m3_geom_s(m,(cck-(n-1)));
        else
            m3_geom_d(m,n+1) = m3_geom_s(m,(cck-(n-1)));
        end
    end
end

end


%%%% Integrated drain-main-source script because I'm stoopid...

%clear all;
%clc;
%format long

function [Rds_device, Rbond, number_bondpads, Ross, Coss] = ...
drain_source_resistance(rows, columns, columns_tot, m3_geom_s, m3_geom_d,
...top_cells, bottom_cells, segments, Wcell, Wm1g, Wm2s, Wm2d,
...total_vias1_drain,total_vias1_source,total_contacts_drain,
...total_contacts_source,test)

%%% General Parameters
disp_geom_map = 0;
rm1square = 0.051;
rm2square = 0.053;
rm3square = 0.016;
rcont_poly = 5;
rcont_pplus = 16;
rcont_nplus = 20;
rvia1 = 0.7;
rvia2 = 0.18;
r_mm2 = 0.137819;
%r_mm2 = 0.15093;
%r_mm2 = 0.190;
```

```
cell_pitch = 15.3;
met3_met3 = 3;
via_size = 0.7;
via2_met2 = 0.4;
drain_connect = 1;
source_connect = 2;
cells_per_m3 = top_cells + bottom_cells;
bondpad_bondpad = 134.7;
bondpad_size = 100;
rbondwire = 0.100; %1 mil gold bondwire is about 42 mOhm/mm..
%say about 2mm average length


%%%%%%%%%%%%%%%% TEST VALUES (MATCH F-TYPE PART) %%%%%%%%%%%%%%%%%%%%%%%%%%
% to use comment out 'Establish Conductance Parameters' block

% Gm3v_d = 1/0.036;   %vertical metal3 resistance
% Gm3h_d = 1/.0071;    %horizontal metal3 resistance
% Gm2_d = 1/.04915;     %metal2 resistance
% Gm2_dtb = 1/.10;   %metal2 resistance at top and bottom (~Gm2_d)
% Gv2_d = 1/.18;    %via2 resistance
% Gv1_d = 16/.7;
%
% Gm3v = 1/0.036;
% Gm3h = 1/.0071;
% Gm2 = 1/.05406;
% Gv2 = 1/.180;
% Gshort = 10000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% % % %Gcell = 1/(246+1.11+0.95+0.0175); %(1/288) rcell + rcont_s + rcont_d
%+ rvia1_s
% % % Gcell = 1/(288+1.11+0.95+0.0175)
% % % cells_per_m3 = 8;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%% ESTABLISH CONDUCTANCE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%

if test == 0
    %%% Calculate the  source-side conductances
    w_m3v = cell_pitch - ((segments-1)*met3_met3)/columns_tot;
    l_m3v = Wcell+9.5+(Wm1g-2.9);
    Gm3v = w_m3v/(l_m3v*rm3square);
    Gm3h = l_m3v/(w_m3v*rm3square);
    Gm2 = Wm2s/(cell_pitch*rm2square);
```

```
    num_via2_source = 21*floor(Wm2s - 2*0.5*via_size - 2*via2_met2);
    Gv2 = num_via2_source/rvia2;
    Gshort = 100000;
    %rcont_s = 1/(total_pplus/rcont_pplus + total_nplus/rcont_nplus);


    %%% Calculate the drain-side conductances
    w_m3v = cell_pitch - ((segments-1)*met3_met3)/columns_tot;
    l_m3v = Wcell+9.5+(Wm1g-2.9);
    Gm3v_d = w_m3v/(l_m3v*rm3square);  %vertical metal3 conductance
    Gm3h_d = l_m3v/(w_m3v*rm3square);  %horizontal metal3 conductance
    Gm2_d = Wm2d/(cell_pitch*rm2square); % metal2 conductance
    Gm2_dtb = Gm2_d;  %metal2 conductance for top and bottom runs (~Gm2_d 'cause I'm lazy)
    num_via2_drain = 21*floor(Wm2d - 2*0.5*via_size - 2*via2_met2);
    Gv2_d = 1/rvia2;  %via2 conductance
    Gv1_d = total_vias1_drain/rvia1; %equivalent conductance of drain-side vias1

        % %%% Calculate the cell conductance
    % %%% cell resistance is the drain contact resistance + source contact
    % %%% resistance + source via1's + intrinsic cell resistance
    % %rcell_int = (288*(2*Wcell*rows*columns))/(2*25*12*120);
    % %Rcell = rcont_d + rcell_int + rcont_s + rvias1_s;
    % %Gcell = 1/Rcell;
    rvias1_s = rvia1/total_vias1_source;   % resistance of the via1's on the _source_side
    rcont_s = rcont_nplus/total_contacts_source; % resistance of the drain-side contacts
    rcont_d = rcont_nplus/total_contacts_drain; % resistance of the source-side contacts
    area_um2 = rows*cell_pitch*columns*(Wcell+9.5+(Wm1g-2.9)); % total active area of device um^2
    area_mm2 = area_um2*1e-6; % rescale to square mm
    rdevice = r_mm2/area_mm2; % intrinsic device resistance
    rcell_int = rdevice*rows*columns; % resistance of a single cell
    Rcell = rcont_d + rcell_int + rcont_s + rvias1_s; % total resistance to go into model
    Gcell = 1/Rcell; % conductance

else
    % dummy parameters to compare intrinsic resistance vs intrinsic + metal
    Gm3v = 100000;
    Gm3h = 100000;
    Gm2 = 100000;
    Gv2 = 100000;
    Gshort = 100000;
    Gm3v_d = 100000;
    Gm3h_d = 100000;
    Gm2_d = 100000;
    Gm2_dtb = 100000;
    Gv2_d = 100000;
    Gv1_d = 100000;

    % %%% Calculate the cell conductance
    % %%% cell resistance is the drain contact resistance + source contact
```

```
    % %%% resistance + source via1's + intrinsic cell resistance
    % %rcell_int = (288*(2*Wcell*rows*columns))/(2*25*12*120);
    % %Rcell = rcont_d + rcell_int + rcont_s + rvias1_s;
    % %Gcell = 1/Rcell;
    %rvias1_s = rvia1/total_vias1_source;   % resistance of the via1's on the _source_side
    %rcont_s = rcont_nplus/total_contacts_source; % resistance of the drain-side contacts
    %rcont_d = rcont_nplus/total_contacts_drain; % resistance of the source-side contacts
    area_um2 = rows*cell_pitch*columns*(Wcell+9.5+(Wm1g-2.9)); % total active area of device um^2
    area_mm2 = area_um2*1e-6; % rescale to square mm
    rdevice = r_mm2/area_mm2; % intrinsic device resistance
    rcell_int = rdevice*rows*columns; % resistance of a single cell
    Rcell = rcell_int; % total resistance to go into model
    Gcell = 1/Rcell; % conductance




end


% %%% Calculate the cell conductance
% %%% cell resistance is the drain contact resistance + source contact
% %%% resistance + source via1's + intrinsic cell resistance
% %rcell_int = (288*(2*Wcell*rows*columns))/(2*25*12*120);
% %Rcell = rcont_d + rcell_int + rcont_s + rvias1_s;
% %Gcell = 1/Rcell;
rvias1_s = rvia1/total_vias1_source;   % resistance of the via1's on the _source_side
rcont_s = rcont_nplus/total_contacts_source; % resistance of the drain-side contacts
rcont_d = rcont_nplus/total_contacts_drain; % resistance of the source-side contacts
area_um2 = rows*cell_pitch*columns*(Wcell+9.5+(Wm1g-2.9)); % total active area of device um^2
area_mm2 = area_um2*1e-6; % rescale to square mm
rdevice = r_mm2/area_mm2; % intrinsic device resistance
rcell_int = rdevice*rows*columns; % resistance of a single cell
Rcell = rcont_d + rcell_int + rcont_s + rvias1_s; % total resistance to go into model
Gcell = 1/Rcell; % conductance


%%%%%%%%%%%%%%%%% END ESTABLISH CONDUCTANCE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%




%%%%%%%%%%%%%%%%% CREATE NODE MATRICES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%create the drain-side metal-3 nodes
drain_m3 = zeros(rows+2,columns);
nodecount = 3;
set_node = 1;
```

```
for m = 1:1:(rows+2)
    for n = 1:1:columns
        if set_node > cells_per_m3 - m3_geom_d(m)
            if m == 1
                drain_m3(m,n) = drain_connect;
            else
                drain_m3(m,n) = nodecount;
                nodecount = nodecount + 1;
            end
        end
        set_node = set_node + 1;
        if set_node > cells_per_m3 | n == columns
            set_node = 1;
        end
    end
end



% create the source-side cell nodes
source_nodes = zeros(rows,columns);
for m = 1:1:(rows)
    for n = 1:1:columns
        source_nodes(m,n) = nodecount;
        nodecount = nodecount + 1;
    end
end



% create the source-side metal-3 nodes
source_m3 = zeros(rows+1,columns);
set_node = 1;
for m = 1:1:rows+1
    for n = 1:1:columns
        if set_node <= m3_geom_s(m)
            if m == (rows+1)
                source_m3(m,n) = source_connect;
            else
                source_m3(m,n) = nodecount;
                nodecount = nodecount + 1;
            end
        end
        set_node = set_node + 1;
        if set_node > cells_per_m3 | n == columns
            set_node = 1;
        end
    end
end
```

```
% create the drain-side cell nodes
drain_nodes = zeros(rows,columns);
for m = 1:1:rows
    for n = 1:1:columns
        drain_nodes(m,n) = nodecount;
        nodecount = nodecount + 1;
    end
end



% create the drain-side metal-2 nodes
drain_m2 = zeros(rows+1,columns);
for m = 1:1:(rows+1)
    for n = 1:1:columns
        drain_m2(m,n) = nodecount;
        nodecount = nodecount + 1;
    end
end

%%%%%%%%%%%%%%%%%%% END CREATE NODE MATRICIES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% Create the conductance matrix
Gmatrix = zeros(nodecount-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%% Insert the cell conductances between the drain-side cell nodes and
%%%% source-side cell nodes

for m = 1:1:rows
    for n = 1:1:columns
        dnode_num = drain_nodes(m,n);
        snode_num = source_nodes(m,n);
        Gmatrix(dnode_num,dnode_num) = Gmatrix(dnode_num,dnode_num) + Gcell;  % add self-term at drain node
        Gmatrix(snode_num,snode_num) = Gmatrix(snode_num,snode_num) + Gcell;  % add self-term at source node
        Gmatrix(dnode_num,snode_num) = Gmatrix(dnode_num,snode_num) - Gcell;  % add the mutual term from dra
        Gmatrix(snode_num,dnode_num) = Gmatrix(snode_num,dnode_num) - Gcell;  % add the mutual term from sou
     end
end
```

```
%%%%%%%%%%%%%%%%%%%%%% Populate all the source side metal%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%% Insert the source-side metal-2 row conductances
for m = 1:1:rows
    for n = 1:1:columns
        if n == 1 & source_nodes(m,n+1)~=0  %if at first column and next node isn't empty
            Gmatrix(source_nodes(m,n),source_nodes(m,n)) =
            ...Gmatrix(source_nodes(m,n),source_nodes(m,n)) + Gm2; % add self-term
            Gmatrix(source_nodes(m,n),source_nodes(m,n+1)) =
            ...Gmatrix(source_nodes(m,n),source_nodes(m,n+1)) - Gm2; % add mutual term  (ie off-diagonal ter
            Gmatrix(source_nodes(m,n+1),source_nodes(m,n)) =
            ...Gmatrix(source_nodes(m,n+1),source_nodes(m,n)) - Gm2; % add opposite mutual term (ie, corresp
        elseif n == columns & source_nodes(m,n-1)~=0 %if at last column and the previous node wasn't empty
            Gmatrix(source_nodes(m,n),source_nodes(m,n)) =
            ...Gmatrix(source_nodes(m,n),source_nodes(m,n)) + Gm2; % add self-term only
        elseif source_nodes(m,n+1)~=0 & source_nodes(m,n-1)~=0 %if in the middle somewhere
            Gmatrix(source_nodes(m,n),source_nodes(m,n)) =
            ...Gmatrix(source_nodes(m,n),source_nodes(m,n)) + 2*Gm2; % add twice the metal-2 conductance to
            Gmatrix(source_nodes(m,n),source_nodes(m,n+1)) =
            ...Gmatrix(source_nodes(m,n),source_nodes(m,n+1)) - Gm2; % add mutual term  (ie off-diagonal ter
            Gmatrix(source_nodes(m,n+1),source_nodes(m,n)) =
            ...Gmatrix(source_nodes(m,n+1),source_nodes(m,n)) - Gm2; % add opposite mutual term (ie, corresp
        end
    end
end

%%%% Insert the source-side metal-2 row to metal-3 via2 conductances (note
%%%% that metal-2 row nodes and source nodes are the same on the source
%%%% side this is _not_ the case for the drain, where the structure is
%%%% different
for m = 1:1:rows
    for n = 1:1:columns
        if source_nodes(m,n)~=0 & source_m3(m,n)~=0
            Gmatrix(source_nodes(m,n), source_m3(m,n)) =
            ...Gmatrix(source_nodes(m,n), source_m3(m,n)) - Gv2; % add the mutual term
            Gmatrix(source_m3(m,n), source_nodes(m,n)) =
            ...Gmatrix(source_m3(m,n), source_nodes(m,n)) - Gv2; % add the opposite mutual term
            Gmatrix(source_nodes(m,n), source_nodes(m,n)) =
            ...Gmatrix(source_nodes(m,n), source_nodes(m,n)) + Gv2; % add the Gv2 component to the self-term
            Gmatrix(source_m3(m,n), source_m3(m,n)) =
            ...Gmatrix(source_m3(m,n), source_m3(m,n)) + Gv2; % add the Gv2 component to the self-term on th
        end
    end
end

%%%% Insert the source-side metal-3 sheet conductances

for m = 1:1:rows+1
```

```
for n = 1:1:columns

    %%% add the horizontal conductances

    if m == rows+1    % since the bottom row is where each metal-3 triangle runs into the bond-strip, mo
    elseif n == 1   % trap for special case - prevent index under-run when checking for presence of prev
        if source_m3(m,n)~=0 & source_m3(m,n+1)~=0 % if there's a node in the next column then...
            Gmatrix(source_m3(m,n), source_m3(m,n)) =
            ...Gmatrix(source_m3(m,n), source_m3(m,n)) + Gm3h; % add horizontal self-term
            Gmatrix(source_m3(m,n), source_m3(m,n+1)) =
            ...Gmatrix(source_m3(m,n), source_m3(m,n+1)) - Gm3h; % add next-node mutual term
            Gmatrix(source_m3(m,n+1),source_m3(m,n)) =
            ...Gmatrix(source_m3(m,n+1),source_m3(m,n)) - Gm3h; % add opposite next-node mutual term
        end
    elseif n == columns %trap for special case - prevent index over-run when checking for presence of ne
        if source_m3(m,n)~=0 & source_m3(m,n-1)~=0 % we're at the end of the device, if there was a prev
            Gmatrix(source_m3(m,n), source_m3(m,n)) =
            ...Gmatrix(source_m3(m,n), source_m3(m,n)) + Gm3h; % add self-term looking back, but don't a
        end
    elseif  source_m3(m,n)~=0 & source_m3(m,n+1)~=
    ...0 & source_m3(m,n-1)~=0  %if there's a node behind and in front, then...
        Gmatrix(source_m3(m,n), source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n), source_m3(m,n)) + 2*Gm3h; % add 2x horizontal self-term
        Gmatrix(source_m3(m,n), source_m3(m,n+1)) =
        ...Gmatrix(source_m3(m,n), source_m3(m,n+1)) - Gm3h; % add next-node mutual term
        Gmatrix(source_m3(m,n+1),source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n+1),source_m3(m,n)) - Gm3h;  % add opposite next-node mutual term
    elseif  source_m3(m,n)~=0 & source_m3(m,n+1)==0
    ...& source_m3(m,n-1)~=0 % if the next node doesn't exist, we're at the end of an m3 strip (horizont
        Gmatrix(source_m3(m,n), source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n), source_m3(m,n)) + Gm3h; % add horizontal self-term
    elseif source_m3(m,n)~=0 & source_m3(m,n+1)~=0 & source_m3(m,n-1)==0 % if the previous node didn't e
        Gmatrix(source_m3(m,n), source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n), source_m3(m,n)) + Gm3h; % add self-term
        Gmatrix(source_m3(m,n), source_m3(m,n+1)) =
        ...Gmatrix(source_m3(m,n), source_m3(m,n+1)) - Gm3h; % add next-node mutual term
        Gmatrix(source_m3(m,n+1), source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n+1), source_m3(m,n)) - Gm3h; % add opposite next-node mutual term
    end

        %%% add the vertical conductances

    if m == 1  % trap for a special case - prevent index under-run when checking for presence of previou
        if source_m3(m,n)~=0 & source_m3(m+1,n)~=0  % if node exists and next node down exists then...
            Gmatrix(source_m3(m,n),source_m3(m,n)) =
            ...Gmatrix(source_m3(m,n),source_m3(m,n)) + Gm3v; %add self-term
            Gmatrix(source_m3(m,n),source_m3(m+1,n)) =
            ...Gmatrix(source_m3(m,n),source_m3(m+1,n)) - Gm3v; % add next-node mutual term
            Gmatrix(source_m3(m+1,n),source_m3(m,n)) =
```

```
            ...Gmatrix(source_m3(m+1,n),source_m3(m,n)) - Gm3v; % add opposite next-node mutual term
        end
    elseif m == rows+1  % trap for special case - prevent index over-run when checking for presence of n
        if source_m3(m,n)~=0 & source_m3(m-1,n)~=0  % if node (and previous node above) exists we've rea
            Gmatrix(source_m3(m,n),source_m3(m,n)) =
            ...Gmatrix(source_m3(m,n),source_m3(m,n)) + Gm3v; % add self-term (mutual already done)
        end
    elseif source_m3(m,n)~=0 & source_m3(m+1,n)~=0 & source_m3(m-1,n)~=0 %if node exists and nodes above
        Gmatrix(source_m3(m,n),source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n),source_m3(m,n)) + 2*Gm3v; % add 2x self-term
        Gmatrix(source_m3(m,n),source_m3(m+1,n)) =
        ...Gmatrix(source_m3(m,n),source_m3(m+1,n)) - Gm3v; % add next-node mutual term
        Gmatrix(source_m3(m+1,n),source_m3(m,n)) =
        ...Gmatrix(source_m3(m+1,n),source_m3(m,n)) - Gm3v; % add opposite next-node mutual term
    elseif source_m3(m,n)~=0 & source_m3(m+1,n)~=0
    ...& source_m3(m-1,n)==0  %if node exists and node below exists but node above doesn't we're on a sh
        Gmatrix(source_m3(m,n),source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n),source_m3(m,n)) + Gm3v; %add self term
        Gmatrix(source_m3(m,n),source_m3(m+1,n)) =
        ...Gmatrix(source_m3(m,n),source_m3(m+1,n)) - Gm3v; % add next-node (lower) mutual term
        Gmatrix(source_m3(m+1,n),source_m3(m,n)) =
        ...Gmatrix(source_m3(m+1,n),source_m3(m,n)) - Gm3v; % add opposite next-node mutual term
    elseif source_m3(m,n)~=0 & source_m3(m+1,n)==0 &
    ...source_m3(m-1,n)~=0 % if node exists and node above exists but node below doesn't, we've reached
        Gmatrix(source_m3(m,n),source_m3(m,n)) =
        ...Gmatrix(source_m3(m,n),source_m3(m,n)) + Gm3v; % add self-term only
    end

  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Source Side Metal Population complete %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Drain-side Metal Population begins
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%% Add the drain-side metal-2 row conductances
for m = 1:1:rows+1
    for n = 1:1:columns

        % This adds the metal-2 row conductances and accounts for the top
        % and bottom rows (more narrow metal strips) separately
        if m == 1 | m == rows  % if we're at either the top or bottom row...
            if n == 1 & drain_m2(m,n+1)~=0  % if we're at the start of a row and the next node isn't zero
                Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
                ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + Gm2_dtb; % add self-term
                Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) =
                ...Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) - Gm2_dtb; % add mutual term  (ie off-diagonal ter
                Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) =
```

```
            ...Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) - Gm2_dtb; % add opposite mutual term (ie, corresp
        elseif n == columns & drain_m2(m,n-1)~=0 % if we're at the end of a row and the previous node wa
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + Gm2_dtb; % add self-term only
        elseif drain_m2(m,n+1)~=0 & drain_m2(m,n-1)~=0  % if we're in the middle of a row...
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + 2*Gm2_dtb; % add twice the metal-2 conductance to
            Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) - Gm2_dtb; % add mutual term  (ie off-diagonal ter
            Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) - Gm2_dtb; % add opposite mutual term (ie, corresp
        end
    else
        if n == 1 & drain_m2(m,n+1)~=0  % if we're at the start of a row and the next node isn't zero...
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + Gm2_d; % add self-term
            Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) - Gm2_d; % add mutual term  (ie off-diagonal term
            Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) - Gm2_d; % add opposite mutual term (ie, correspon
        elseif n == columns & drain_m2(m,n-1)~=0  % if we're at the end of a row and the previous node w
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + Gm2_d; % add self-term only
        elseif drain_m2(m,n+1)~=0 & drain_m2(m,n-1)~=0 %if we're in the middle of a row...
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + 2*Gm2_d; % add twice the metal-2 conductance to th
            Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n+1)) - Gm2_d; % add mutual term  (ie off-diagonal term
            Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n+1),drain_m2(m,n)) - Gm2_d; % add opposite mutual term (ie, correspon
        end
    end
end


%%%% Insert the source-side metal-2 row to metal-3 via2 conductances
%%%% The drain_m2 matrix and drain_m3 matrix must be offset by one row
%%%% An extra row is added to m3 to provide an attachment point to the
%%%% drain bondpads.  Since this occurs in row-1 we have to account for it,
%%%% unlike in the case of the source metal-3
for m = 1:1:rows+1
    for n = 1:1:columns
        if drain_m2(m,n)~=0 & drain_m3(m+1,n)~=0
            Gmatrix(drain_m2(m,n), drain_m3(m+1,n)) =
            ...Gmatrix(drain_m2(m,n), drain_m3(m+1,n)) - Gv2_d; % add the mutual term
            Gmatrix(drain_m3(m+1,n), drain_m2(m,n)) =
            ...Gmatrix(drain_m3(m+1,n), drain_m2(m,n)) - Gv2_d; % add the opposite mutual term
```

```
            Gmatrix(drain_m2(m,n), drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n), drain_m2(m,n)) + Gv2_d; % add the Gv2 component to the self-term on th
            Gmatrix(drain_m3(m+1,n), drain_m3(m+1,n)) =
            ...Gmatrix(drain_m3(m+1,n), drain_m3(m+1,n)) + Gv2_d; % add the Gv2 component to the self-term o
        end
    end
end


%%%% Add the drain-side metal-3 sheet conductances

for m = 1:1:rows+2
    for n = 1:1:columns

        %%%% add the horizontal conductances
        if m == 1    % since the top row is where each metal-3 triangle runs into the bond-strip, modeled as
        elseif n == 1   % trap for special case - prevent index under-run when checking for presence of prev
            if drain_m3(m,n)~=0 & drain_m3(m,n+1)~=0 % if this node exists and there's a node in the next co
                Gmatrix(drain_m3(m,n), drain_m3(m,n)) =
                ...Gmatrix(drain_m3(m,n), drain_m3(m,n)) + Gm3h_d; % add horizontal self-term
                Gmatrix(drain_m3(m,n), drain_m3(m,n+1)) =
                ...Gmatrix(drain_m3(m,n), drain_m3(m,n+1)) - Gm3h_d; % add next-node mutual term
                Gmatrix(drain_m3(m,n+1),drain_m3(m,n)) =
                ...Gmatrix(drain_m3(m,n+1),drain_m3(m,n)) - Gm3h_d; % add opposite next-node mutual term
            end
        elseif n == columns %trap for special case - prevent index over-run when checking for presence of ne
            if drain_m3(m,n)~=0 & drain_m3(m,n-1)~=0 % we're at the end of the device, if there was a previo
                Gmatrix(drain_m3(m,n), drain_m3(m,n)) =
                ...Gmatrix(drain_m3(m,n), drain_m3(m,n)) + Gm3h_d; % add self-term looking back, but don't a
            end
        elseif  drain_m3(m,n)~=0 & drain_m3(m,n+1)~=0 &
        ...drain_m3(m,n-1)~=0  %if there's a node behind and in front, then...
            Gmatrix(drain_m3(m,n), drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n), drain_m3(m,n)) + 2*Gm3h_d; % add 2x horizontal self-term
            Gmatrix(drain_m3(m,n), drain_m3(m,n+1)) =
            ...Gmatrix(drain_m3(m,n), drain_m3(m,n+1)) - Gm3h_d; % add next-node mutual term
            Gmatrix(drain_m3(m,n+1),drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n+1),drain_m3(m,n)) - Gm3h_d;  % add opposite next-node mutual term
        elseif  drain_m3(m,n)~=0 & drain_m3(m,n+1)==0 &
        ...drain_m3(m,n-1)~=0 % if the next node doesn't exist, we're at the end of an m3 strip (horizontall
            Gmatrix(drain_m3(m,n), drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n), drain_m3(m,n)) + Gm3h_d; % add horizontal self-term
        elseif drain_m3(m,n)~=0 & drain_m3(m,n+1)~=0 &
        ...drain_m3(m,n-1)==0 % if the previous node didn't exist, but there's a next node, we're at the sta
            Gmatrix(drain_m3(m,n), drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n), drain_m3(m,n)) + Gm3h_d; % add self-term
            Gmatrix(drain_m3(m,n), drain_m3(m,n+1)) =
            ...Gmatrix(drain_m3(m,n), drain_m3(m,n+1)) - Gm3h_d; % add next-node mutual term
```

```
            Gmatrix(drain_m3(m,n+1), drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n+1), drain_m3(m,n)) - Gm3h_d; % add opposite next-node mutual term
        end

            %%% add the vertical conductances

        if m == 1  % trap for a special case - prevent index under-run when checking for presence of previou
            if drain_m3(m,n)~=0 & drain_m3(m+1,n)~=0  % if node exists and next node down exists then...
                Gmatrix(drain_m3(m,n),drain_m3(m,n)) =
                ...Gmatrix(drain_m3(m,n),drain_m3(m,n)) + Gm3v_d; %add self-term
                Gmatrix(drain_m3(m,n),drain_m3(m+1,n)) =
                ...Gmatrix(drain_m3(m,n),drain_m3(m+1,n)) - Gm3v_d; % add next-node mutual term
                Gmatrix(drain_m3(m+1,n),drain_m3(m,n)) =
                ...Gmatrix(drain_m3(m+1,n),drain_m3(m,n)) - Gm3v_d; % add opposite next-node mutual term
            end
        elseif m == rows+2  % trap for special case - prevent index over-run when checking for presence of n
            if drain_m3(m,n)~=0 & drain_m3(m-1,n)~=0  % if node (and previous node) exists we've reached the
                Gmatrix(drain_m3(m,n),drain_m3(m,n)) =
                ...Gmatrix(drain_m3(m,n),drain_m3(m,n)) + Gm3v_d; % add self-term
            end
        elseif drain_m3(m,n)~=0 & drain_m3(m+1,n)~=0 &
        ...drain_m3(m-1,n)~=0 %if node exists and nodes above and below exist...
            Gmatrix(drain_m3(m,n),drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n),drain_m3(m,n)) + 2*Gm3v_d; % add 2x self-term
            Gmatrix(drain_m3(m,n),drain_m3(m+1,n)) =
            ...Gmatrix(drain_m3(m,n),drain_m3(m+1,n)) - Gm3v_d; % add next-node mutual term
            Gmatrix(drain_m3(m+1,n),drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m+1,n),drain_m3(m,n)) - Gm3v_d; % add opposite next-node mutual term
        elseif drain_m3(m,n)~=0 & drain_m3(m+1,n)~=0 &
        ...drain_m3(m-1,n)==0  %if node exists and node below exists but node above doesn't..
            Gmatrix(drain_m3(m,n),drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n),drain_m3(m,n)) + Gm3v_d; %add self term
            Gmatrix(drain_m3(m,n),drain_m3(m+1,n)) =
            ...Gmatrix(drain_m3(m,n),drain_m3(m+1,n)) - Gm3v_d; % add next-node (lower) mutual term
            Gmatrix(drain_m3(m+1,n),drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m+1,n),drain_m3(m,n)) - Gm3v_d; % add opposite next-node mutual term
        elseif drain_m3(m,n)~=0 & drain_m3(m+1,n)==0 &
        ...drain_m3(m-1,n)~=0 % if node exists and node above exists but node below doesn't, we've reached a
            Gmatrix(drain_m3(m,n),drain_m3(m,n)) =
            ...Gmatrix(drain_m3(m,n),drain_m3(m,n)) + Gm3v_d; % add self-term only
        end

    end
end


%%%% Add the drain-side metal-2 - via1 - metal-1 connections
for m = 1:1:rows
```

```
    for n = 1:1:columns
        if m == 1
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + Gv1_d; % add self-term at m2 node
            Gmatrix(drain_nodes(m,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_nodes(m,n)) + 2*Gv1_d; % add self_term at drain cell node
            Gmatrix(drain_m2(m,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_nodes(m,n)) - Gv1_d;  % mutual metal-2 - drain-cell term
            Gmatrix(drain_nodes(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_m2(m,n)) - Gv1_d;  % mutual drain-cell - metal-2 term
        elseif m == rows
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + 2*Gv1_d; % add self-term at second-to-last m2 term
            Gmatrix(drain_m2(m+1,n),drain_m2(m+1,n)) =
            ...Gmatrix(drain_m2(m+1,n),drain_m2(m+1,n)) + Gv1_d; % add self-term to last m2 term
            Gmatrix(drain_nodes(m,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_nodes(m,n)) + 2*Gv1_d; % add self-term to last drain cell node
            Gmatrix(drain_m2(m,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_nodes(m,n)) - Gv1_d; % second-to-last m2-node to last drain cell
            Gmatrix(drain_nodes(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_m2(m,n)) - Gv1_d; % last drain cell node to second-to-last m2-
            Gmatrix(drain_m2(m,n),drain_nodes(m-1,n)) =
            ...Gmatrix(drain_m2(m,n),drain_nodes(m-1,n)) - Gv1_d; %second-to-last m2-node to second-to-last
            Gmatrix(drain_nodes(m-1,n),drain_m2(m,n)) =
            ...Gmatrix(drain_nodes(m-1,n),drain_m2(m,n)) - Gv1_d; %second-to-last drain cell node to second-
            Gmatrix(drain_m2(m+1,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_m2(m+1,n),drain_nodes(m,n)) - Gv1_d; %last m2-node to last drain-cell node mutu
            Gmatrix(drain_nodes(m,n),drain_m2(m+1,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_m2(m+1,n)) - Gv1_d; %last drain-cell node to last m2-node mutu
        else
            Gmatrix(drain_m2(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_m2(m,n)) + 2*Gv1_d; % add self-term at m2 node
            Gmatrix(drain_nodes(m,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_nodes(m,n)) + 2*Gv1_d; % add self-term at drain-cell node
            Gmatrix(drain_m2(m,n),drain_nodes(m,n)) =
            ...Gmatrix(drain_m2(m,n),drain_nodes(m,n)) - Gv1_d; % add m2-node to next drain-cell below node
            Gmatrix(drain_nodes(m,n),drain_m2(m,n)) =
            ...Gmatrix(drain_nodes(m,n),drain_m2(m,n)) - Gv1_d; % add next drain-cell below node to m2-node
            Gmatrix(drain_m2(m,n),drain_nodes(m-1,n)) =
            ...Gmatrix(drain_m2(m,n),drain_nodes(m-1,n)) - Gv1_d; % add m2-node to drain-cell-above node mut
            Gmatrix(drain_nodes(m-1,n),drain_m2(m,n)) =
            ...Gmatrix(drain_nodes(m-1,n),drain_m2(m,n)) - Gv1_d; % add drain-cell-above node to m2-node mut
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%End of DRAIN POPULATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%% Add conductances from the source terminal to ground
for n = 1:1:columns
    if source_m3(rows+1,n) == source_connect
        Gmatrix(source_m3(rows+1,n),source_m3(rows+1,n)) =
        ...Gmatrix(source_m3(rows+1,n),source_m3(rows+1,n)) + Gshort;  %add a short to ground at this node
    end
end


%currents = zeros(nodecount,1);
%currents(nodecount,1) = 1;
%volts = Gmatrix\currents;
%rdrainsource = volts(nodecount)

currents = zeros(nodecount-1,1);
currents(1,1) = 1;
volts = Gmatrix\currents;
Rds_device = volts(1);


%number_bondpads = floor(((columns*cell_pitch - 2*0.5*bondpad_size)/bondpad_bondpad)+1);
number_bondpads = floor((columns_tot*cell_pitch)/bondpad_bondpad)+1;
Rbond = (rbondwire/number_bondpads)*2;
%Rds_total = Rds_device + Rbond;


if disp_geom_map == 1
figure(1)
spy(drain_m3)
title('drain')
figure(2)
spy(source_m3)
title('source')
end

%%%% Calculate ROSS (very approximate!)

normal_Weff = 2*25*12*120;
weff_calc = 2*Wcell*rows*columns_tot;
Ross = 0.65*(normal_Weff/weff_calc);

%%%%% Calculate COSS

Coss = 135e-12*(weff_calc/normal_Weff);

end


clc;
```

```
start = 0;
stop = 0;
rows_test = 11;
columns_test = 127;
Wcell_test = 25;
rcomp = 0
ccomp = 0
[trial_counter,trial_col_counter] = size(trials);
for i = 1:1:trial_counter

    %%% find the
    if rcomp ~= trials{i,3}(1) || ccomp ~= trials{i,4}(1)
            rcomp = trials{i,3};
            ccomp = trials(i,4);
            if toplot(k,3) == cell_width && toplot(k,4) == 2.9
                counter = counter + 1;
                check = 1000;
            end


    seg_check(i) = trials{i,12}(1);


    %%%% find the start and stop rows where the conditions are met (rows,
    %%%% columns, and cell width)
    if trials{i,3}(1) == rows_test && trials{i,4}(1) == columns_test && trials{i,5}(1) == Wcell_test
        if start == 0
            start = i
        end
        stop = i;
    end
end
stop

clear geom
clear segs
geom_count = 1;
for i = start:1:stop
    geom(geom_count,:) = trials{i,10}(:);
    segs(geom_count) = trials{i,12}(1);
    geom_count = geom_count + 1;
end

smallest_segment_num = min(seg_check)


clear all;
clc;
```

```
%function [rgate]=gate_resistance(rows, columns, Gm1, Ggc, Gcap, number_bonds,f)

function [rgate,cgate] = gate_resistance(rows, columns, Wcell, number_bonds,
...rcell, ccell, rstub, Wm1g)


%%%%% Compute the gate resistance

%%% Parameters
f = 50e6;
cell_pitch = 15.3;
cont_cont_center = 1.4;
rm1square = 0.051;
number_contacts = 8 + 4*floor((Wm1g-2.9)/cont_cont_center); %assumes constant cell pitch, scaling of metal1 p
rcontacts = 5/(number_contacts);
normalized_width = (2*Wcell*rows*columns)/(2*25*12*120);  %this is in comparison to the F-type sample part
num_cells = rows*columns;
Rcap = (num_cells*0.2)/normalized_width; % this is also related to the F-type sample, with 0.2 being Rds-on a
rbondwire = 0.100; % average resistance of a 3-bondwire set, 1mil, gold


% Calculate conductances
Gm1 = Wm1g/(cell_pitch*rm1square);  %metal1 resistance
Ggc = 1/(rcell+rstub+rcontacts); %cell poly leg resistance, poly stub resistance, cont-poly resistance
Gcap = 1/Rcap;  % actually it's a complex admittance, but the capacitance really doesn't seem to affect thin


w = 2*3.14159*f; %50e6


%%% Complex cell admittance Ycap
% Cpoly = 191.6e-15;
% Gvert = 1/360;
% Ycap = complex(Gvert,Cpoly*w); % complex impedance of gate cell if we wish to account for it (time consumi


% determine the number of cells in a row between each gate bond-out area
bond_ref = 1;
if number_bonds > 1
    cells_per_segment = floor(columns/(number_bonds-1));
else
    cells_per_segment = columns;
end

even = rem(columns,cells_per_segment);


%% Create the node matrix
num_segments = 0;
```

```
nodecountg = 2;
cells_in_row = 0;
if number_bonds == 1
    for m = 1:1:(2*rows+1)
        for n = 1:1:columns+1
            if n == 1
                gate_nodes(m,n) = 1;
            else
                gate_nodes(m,n) = nodecountg;
                nodecountg = nodecountg + 1;
            end
        end
    end
elseif even == 0;
    for m = 1:1:(2*rows+1)
        for n = 1:1:columns+number_bonds
            if cells_in_row == cells_per_segment || n == 1
                gate_nodes(m,n) = 1;
                cells_in_row = 0;
            else
                gate_nodes(m,n) = nodecountg;
                nodecountg = nodecountg + 1;
                cells_in_row = cells_in_row + 1;
            end
        end
    end
else
    for m = 1:1:(2*rows+1)
        for n = 1:1:columns+number_bonds
            if even > num_segments
                if cells_in_row == cells_per_segment + 1 || n == 1
                    gate_nodes(m,n) = 1;
                    cells_in_row = 0;
                    if n~=1
                        num_segments = num_segments + 1;
                    end
                else
                    gate_nodes(m,n) = nodecountg;
                    nodecountg = nodecountg + 1;
                    cells_in_row = cells_in_row + 1;
                end
            else
                if cells_in_row == cells_per_segment || n == 1
                    gate_nodes(m,n) = 1;
                    cells_in_row = 0;
                    num_segments = num_segments + 1;
                else
                    gate_nodes(m,n) = nodecountg;
                    nodecountg = nodecountg + 1;
```

```
                cells_in_row = cells_in_row + 1;
            end
        end
    end
    num_segments = 0;
end
%gate_nodes;

%%% populate the conductance matrix
Gmatrixg = zeros(nodecountg - 1);

for m = 1:1:(2*rows+1)
    for n = 1:1:columns+number_bonds

        %%% add the row conductances due to the metal-1 lines
        if rem((m+1),2) == 0 % do this only on the odd rows of the matrix, since those are the lines
            if n == 1
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + Gm1;
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n+1)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n+1))-Gm1;
                Gmatrixg(gate_nodes(m,n+1),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n+1),gate_nodes(m,n))-Gm1;
            elseif n == columns+number_bonds
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + Gm1;
            else
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + 2*Gm1;
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n+1)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n+1))-Gm1;
                Gmatrixg(gate_nodes(m,n+1),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n+1),gate_nodes(m,n))-Gm1;
            end
        end

        %%% add the vertical conductances due to the cell polysilicon and
        %%% contacts

        if gate_nodes(m,n) ~= 1
            if m == 1
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + Ggc;
                Gmatrixg(gate_nodes(m,n),gate_nodes(m+1,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m+1,n))-Ggc;
                Gmatrixg(gate_nodes(m+1,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m+1,n),gate_nodes(m,n))-Ggc;
            elseif m == (2*rows+1)
```

```
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + Ggc;
            else
                Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + 2*Ggc;
                Gmatrixg(gate_nodes(m,n),gate_nodes(m+1,n)) =
                ...Gmatrixg(gate_nodes(m,n),gate_nodes(m+1,n))-Ggc;
                Gmatrixg(gate_nodes(m+1,n),gate_nodes(m,n)) =
                ...Gmatrixg(gate_nodes(m+1,n),gate_nodes(m,n))-Ggc;
            end
        end

        %%%% add the admittance at each cell due to the capacitance

         if rem(m,2) == 0 && gate_nodes(m,n)~=1
             Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) =
             ...Gmatrixg(gate_nodes(m,n),gate_nodes(m,n)) + Gcap;
         end
    end
end

test_current = zeros(nodecountg-1,1);
test_current(1,1) = 1;
volts = Gmatrixg\test_current;


rgate_int = volts(1);
Rbond_gate = rbondwire/number_bonds;
k_rfudge = 1.608; % factor to correct the results based on measurements
rgate = rgate_int*k_rfudge + Rbond_gate;

cgate = ccell*rows*columns;

end


%clear all
%clc

function [total_contacts_drain, total_vias1_drain, rmetal] =
...metal1_drain(Wcell, Wm2d, Wm1g, execute_flag)

%%%% This code is to determine the resistance from metal-2 through vias-1,
%%%% metal-1 and the contacts to the _drain_.  It takes as an input the
%%%% cell width, the metal-1 width, and the metal-2 width.  Other geometrical aspects, such as
%%%% cell pitch are assumed fixed.  Under these assumptions, the width of
%%%% the metal-1 strip is constant, as are the numbers of contacts and vias
%%%% across its width.  This result arises from the design rules of the
%%%% process.
```

```
%%%%              -------
%%%%             | v v v |  <--- Metal-1, c=contact, v=via
%%%%             | v v v |  <--- contacts continue under vias
%%%%             | v v v |       (limitations of ascii!)
%%%%     Length  | v v v |
%%%%             | v v v |
%%%%             | v v v |
%%%%             | v v v |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |   c   |
%%%%             |_______|
%%%%               Width


%%% Parameters to be passed into the function
% execute_flag = 1;
% Wcell = 25;
% Wm2d = 16.5;
% Wm1g = 2.9;


%%% Definitions
via_node = 1;
contact_node = 2;
Gshort = 1000;
Gmet1 = 1/.051;

%Parameters

cont_size = 0.7;
cont_poly = 0.5;
cont_metal1 = 0.4;
cont_metal1_nsc = 0.7;
cont_cont = 0.7;
cont_cont_center = 1.4;
via_size = 0.7;
via1_metal1 = 0.4;
via1_metal2 = 0.4;
via1_via1 = 0.7;
via1_via1_center = 1.4;
met1_met1 = 0.8;
metal1_w = 6.9;
```

```
metal1_l = Wcell+2*2.4;  % cell width plus metal-1 extension on either end
metal1_metal2_ovp = 0.5*Wm2d - met1_met1 - 0.5*Wm1g;




% determine contact and via rows and columns and total number
cont_rows = floor(((metal1_l-2*cont_metal1-2*.5*cont_size)/1.4)+1);
cont_columns = 1;
via_rows = floor(((metal1_metal2_ovp - via1_metal2 -
...via1_metal1-2*0.5*via_size)/1.4)+1);
via_columns = floor((metal1_w - 2*via1_metal1 - 2*0.5*via_size)/1.4+1);
total_contacts_drain = round(cont_rows*cont_columns);
total_vias1_drain = round(via_rows*via_columns);


%%%% Physical location of first via (upper left corner via)
via_x = (metal1_w - (via_columns-1)*via1_via1_center)/2;
via_y = (metal1_metal2_ovp - (via_rows-1)*via1_via1_center)/2;

%%%% Physical location of first contact (on metal-1), always start with
   %%%% 2-contact row
cont_x = metal1_w/2;
cont_y = (metal1_l - (cont_rows-1)*cont_cont_center)/2;


   %%%%% Convert floats to integers
cont_rows = int16(cont_rows);
cont_columns = int16(cont_columns);
via_rows = int16(via_rows);
via_columns = int16(via_columns);

%%%%% Find rows and columns of metal1
%grid = 0.2
grid = 0.36;
met1_columns = int16(round(metal1_w/grid));
met1_rows = int16(round(metal1_l/grid));

%%%%% Locate vias and contacts on the grid
column_via = int16(floor(via_x/grid));  %first column via appears in
row_via = int16(floor(via_y/grid));  % first row via appears in
grid_via = int16(via1_via1_center/grid);  % number of grid spaces center-to-center per via

column_cont = int16(floor(cont_x/grid));  % contact column
row_cont = int16(floor(cont_y/grid));       % first row contact appears in
grid_cont = int16(cont_cont_center/grid); % grid spacing center-to-center
```

```
if execute_flag == 1

    %%%%% build node matrix
    met1d_nodes = zeros(met1_rows/2,met1_columns);


            %%% add the contacts first
    %last_cont_row = row_cont+((cont_rows/2-1)*grid_cont)
    last_cont_row = met1_rows/2;
    cont_flag = 1;
    n = column_cont;
    for m = row_cont:1:last_cont_row
            if m >= row_cont & rem((m-row_cont),grid_cont)==0
                met1d_nodes(m,n) = contact_node;
            end
    end



    %%% add the vias

    last_via_row = row_via + ((via_rows-1)*grid_via);
    last_via_column = column_via + ((via_columns-1)*grid_via);

    for m = 1:1:last_via_row
        for n = column_via:1:last_via_column
            if m >= row_via & rem((m-row_via),grid_via)==0
                if n == column_via | rem((n-column_via),grid_via)==0
                    if met1d_nodes(m,n) == 0
                        met1d_nodes(m,n)= via_node;
                    elseif met1d_nodes(m,n+1) == 0
                        met1d_nodes(m,n+1) = via_node;
                    end
                end
            end
        end
    end


    spy(met1d_nodes)


    %%% add the rest of the nodes
    nodecount = 3;
    for m = 1:1:met1_rows/2
        for n = 1:1:met1_columns
            if met1d_nodes(m,n)==0
                met1d_nodes(m,n) = nodecount;
                nodecount = nodecount + 1;
```

```
        end
    end
end

%figure(2)
%spy(met1d_nodes)



%%% create the conductance matrix
Gmatrix_met1d = zeros(nodecount-1);

for m = 1:1:met1_rows/2
    for n = 1:1:met1_columns

        %%% add the row conductances
        if n == 1
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + Gmet1;
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n+1)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n+1))-Gmet1;
            Gmatrix_met1d(met1d_nodes(m,n+1),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n+1),met1d_nodes(m,n))-Gmet1;
        elseif n == met1_columns
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + Gmet1;
        else
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + 2*Gmet1;
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n+1)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n+1))-Gmet1;
            Gmatrix_met1d(met1d_nodes(m,n+1),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n+1),met1d_nodes(m,n))-Gmet1;
        end

        %%% add the vertical conductances


        if m == 1
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + Gmet1;
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m+1,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m+1,n))-Gmet1;
            Gmatrix_met1d(met1d_nodes(m+1,n),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m+1,n),met1d_nodes(m,n))-Gmet1;
        elseif m == met1_rows/2
            Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
            ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + Gmet1;
        else
```

```
                Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
                ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + 2*Gmet1;
                Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m+1,n)) =
                ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m+1,n))-Gmet1;
                Gmatrix_met1d(met1d_nodes(m+1,n),met1d_nodes(m,n)) =
                ...Gmatrix_met1d(met1d_nodes(m+1,n),met1d_nodes(m,n))-Gmet1;
            end

            %%% add a connection to ground at the ground node
            if met1d_nodes(m,n) == contact_node
                Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) =
                ...Gmatrix_met1d(met1d_nodes(m,n),met1d_nodes(m,n)) + Gshort;
            end

        end
    end


    met1d_current = zeros(nodecount-1,1);
    met1d_current(1,1) = 1;
    volts = Gmatrix_met1d\met1d_current;
    rmetal = volts(1);
else
    rmetal = 0;
end
end


%clear all;
%clc;

%function [total_contacts, total_vias1_source, rmetal] = metal1_source(Wcell, Wm2s, execute_flag)
function [total_contacts_source, total_vias1_source, rmetal] =
...metal1_source(Wcell, Wm2s, execute_flag)

%%%% This code is to determine the resistance from metal-2 through vias-1,
%%%% metal-1 and the contacts to the _source_.  It takes as an input the
%%%% cell width and the metal-2 width, other geometrical aspects, such as
%%%% cell pitch are assumed fixed.  Under these assumptions, the width of
%%%% the metal-1 strip is constant, as are the numbers of contacts and vias
%%%% across its width.  This result arises from the design rules of the
%%%% process.

%%%%             -------
%%%%            |  c c  |  <--- Metal-1, c=contact, v=via
%%%%            |   c   |
%%%%            |  c c  |
%%%%            |   c   |
%%%%            |  c c  |
%%%%            | v v v |  <--- contacts continue under vias
```

```
%%%%            | vcvcv |       (limitations of ascii!)
%%%%     Length | v v v |
%%%%            | vcvcv |
%%%%            | v v v |
%%%%            | vcvcv |
%%%%            |   c   |
%%%%            |  c c  |
%%%%            |   c   |
%%%%            |  c c  |
%%%%            |   c   |
%%%%            |_______|
%%%%              Width


%%%% Parameters to be passed into the function
%Wcell = 25; %cell physical width (effective width = 2* Wcell)
%Wm2s = 15;   %metal-2 width
%execute_flag = 1;

%Parameters
via_node = 1;
contact_node = 2;
Gshort = 1000;
Gmet1 = 1/.051;
cont_size = 0.7;
via_size = 0.7;
cont_poly = 0.5;
cont_metal1 = 0.4;
cont_metal1_nsc = 0.7;
cont_cont = 0.7;
cont_cont_center_v = 1.4;
cont_cont_center_h = 2.0;
via1_metal1 = 0.4;
via1_metal2 = 0.4;
via1_via1 = 0.7;
via1_via1_center = 1.4;
w_poly_int = 4.1;
w_metal1 = 6.9;

%%% find the numbers of rows and columns of vias and contacts and thier
%%% total
cont_rows = floor((Wcell-2*cont_poly-2*0.5*cont_size)/1.4+1);
cont_columns = floor((w_poly_int - 2*cont_metal1 - 2*0.5*cont_size)/1.4+1);
via_rows = floor((Wm2s - 2*via1_metal2 - 2*0.5*via_size)/1.4+1);
via_columns = floor((w_metal1 - 2*via1_metal1 - 2*0.5*via_size)/1.4+1);
l_metal1 = (cont_rows-1)*1.4 + 2*cont_size/2 + 2*cont_metal1_nsc;
total_vias1_source = round(via_rows*via_columns);

if rem(cont_rows,2) == 0
```

```
    %total_contacts = int16(0.5*cont_rows + 0.5*cont_rows*cont_columns);
    %pplus_contacts = int16(0.5*contact_rows);
    %nplus_contacts = int16(0.5*cont_rows*cont_columns);
    total_contacts_source = (cont_rows/2)*cont_columns;

else
    %total_contacts = int16(floor(cont_rows/2) + (floor(cont_rows/2)+1)*cont_columns);
    %pplus_contacts = int16(floor(cont_rows/2)
    %nplus_contacts = int16((floor(cont_rows/2)+1)*cont_columns);
    total_contacts_source = floor((cont_rows/2+1))*cont_columns;
end

%%%% Physical location of first via (upper left corner via)
via_x = (w_metal1 - (via_columns-1)*via1_via1_center)/2;
via_y = (l_metal1 - (via_rows-1)*via1_via1_center)/2;

%%%% Physical location of first contact (on metal-1), always start with
   %%%% 2-contact row
cont_x2 = (w_metal1 - (cont_columns-1)*cont_cont_center_h)/2;
cont_y = (l_metal1 - (cont_rows-1)*cont_cont_center_v)/2;
   %%%% 1-contact row
   cont_x1 = w_metal1/2;


%%%%% Convert floats to integers
cont_rows = int16(cont_rows);
cont_columns = int16(cont_columns);
via_rows = int16(via_rows);
via_columns = int16(via_columns);


if execute_flag == 1

    %%%%% Find rows and columns of metal1
    %grid = 0.2
    grid = 0.36;
    met1_columns = int16(round(w_metal1/grid));
    met1_rows = int16(round(l_metal1/grid));

    %%%%% Locate vias and contacts on the grid
    column_via = int16(floor(via_x/grid));  %first column via appears in
    row_via = int16(floor(via_y/grid));  % first row via appears in
    grid_via = int16(via1_via1_center/grid);  % number of grid spaces center-to-center per via (vertical and

    column_cont2 = int16(floor(cont_x2/grid));  % first column contact appears in
    column_cont1 = int16(floor(cont_x1/grid));  % column number of center contact
    row_cont = int16(floor(cont_y/grid));       % first row contact appears in
    grid_cont_v = int16(cont_cont_center_v/grid); % grid spacing center-to-center vertical
    grid_cont_h = int16(cont_cont_center_h/grid); % grid spacing center-to-center horizontal
```

```
%%%%% build node matrix
met1s_nodes = zeros(met1_rows,met1_columns);


        %%% add the contacts first
last_cont_row = row_cont+((cont_rows-1)*grid_cont_v);
last_cont_column = column_cont2+((cont_columns-1)*grid_cont_h);
cont_flag = 1;
for m = row_cont:1:last_cont_row
    for n = column_cont2:1:last_cont_column
        if m >= row_cont & rem((m-row_cont),grid_cont_v)==0
            if cont_flag == 1
                if n == column_cont2 | rem((n-column_cont2),grid_cont_h)==0
                    met1s_nodes(m,n) = contact_node;
                end
                if n == last_cont_column
                    cont_flag = 0;
                end
            elseif cont_flag == 0
                if n == column_cont1
                    met1s_nodes(m,n) = contact_node;
                end
                if n == last_cont_column
                    cont_flag = 1;
                end
            else
            end
        end
    end
end

%%% add the vias

last_via_row = row_via + ((via_rows-1)*grid_via);
last_via_column = column_via + ((via_columns-1)*grid_via);

for m = row_via:1:last_via_row
    for n = column_via:1:last_via_column
        if m >= row_via & rem((m-row_via),grid_via)==0
            if n == column_via | rem((n-column_via),grid_via)==0
                if met1s_nodes(m,n) == 0
                    met1s_nodes(m,n)= via_node;
                elseif met1s_nodes(m,n+1) == 0
                    met1s_nodes(m,n+1) = via_node;
                end
            end
        end
```

```
    end
end



spy(met1s_nodes)

%%% add the rest of the nodes
nodecount = 3;
for m = 1:1:met1_rows
    for n = 1:1:met1_columns
        if met1s_nodes(m,n)==0
            met1s_nodes(m,n) = nodecount;
            nodecount = nodecount + 1;
        end
    end
end

%%% create the conductance matrix
Gmatrix_met1s = zeros(nodecount-1);

for m = 1:1:met1_rows
    for n = 1:1:met1_columns

        %%% add the row conductances
        if n == 1
            Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
            ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + Gmet1;
            Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n+1)) =
            ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n+1))-Gmet1;
            Gmatrix_met1s(met1s_nodes(m,n+1),met1s_nodes(m,n)) =
            ...Gmatrix_met1s(met1s_nodes(m,n+1),met1s_nodes(m,n))-Gmet1;
        elseif n == met1_columns
            Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
            ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + Gmet1;
        else
            Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
            ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + 2*Gmet1;
            Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n+1)) =
            ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n+1))-Gmet1;
            Gmatrix_met1s(met1s_nodes(m,n+1),met1s_nodes(m,n)) =
            ...Gmatrix_met1s(met1s_nodes(m,n+1),met1s_nodes(m,n))-Gmet1;
        end

        %%% add the vertical conductances


        if m == 1
```

```
                Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
                ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + Gmet1;
                Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m+1,n)) =
                ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m+1,n))-Gmet1;
                Gmatrix_met1s(met1s_nodes(m+1,n),met1s_nodes(m,n)) =
                ...Gmatrix_met1s(met1s_nodes(m+1,n),met1s_nodes(m,n))-Gmet1;
            elseif m == met1_rows
                Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
                ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + Gmet1;
            else
                Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
                ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + 2*Gmet1;
                Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m+1,n)) =
                ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m+1,n))-Gmet1;
                Gmatrix_met1s(met1s_nodes(m+1,n),met1s_nodes(m,n)) =
                ...Gmatrix_met1s(met1s_nodes(m+1,n),met1s_nodes(m,n))-Gmet1;
            end

            %%% add a connection to ground at the ground node
            if met1s_nodes(m,n) == contact_node
                Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) =
                ...Gmatrix_met1s(met1s_nodes(m,n),met1s_nodes(m,n)) + Gshort;
            end

        end
    end

    met1s_current = zeros(nodecount-1,1);
    met1s_current(1,1) = 1;
    volts = Gmatrix_met1s\met1s_current;
    rmetal = volts(1);
else
    rmetal = 0;
end
end


%clear all;
%clc;

%function [rstub] = stub_resistance(Wstub,Lstub,Wm1g)



function [rstub] = stub_resistance(Wm1g)



%%%%%%%%%%%%%%%%%%%%%%%%% NOTE NOTE NOTE %%%%%%%%%%%%%%%%%%%%%%
%  Matlab has a problem dividing 4.1/.1 ... it comes out to
```

```
%  40.99999999...check! (really a problem that will occur in most languages
%  but the wya they deal with it in Matlab seems particularly odious.
%  Thus, when taking the floor of 4.1/.1, one gets 40, which
%  isn't what one is looking for... skeptical? try this loop:
%   for i = 1:.1:100
%       i/0.1
%   end
%
%  It sucks!  Beware of the code below if you change Wstub, etc.  roundoff
%  will bite you.  I just hacked it because of limited time.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%% This will find the resistance contributed by the stub portion of each
%%%% cell to the total cell resistance


%%%%%           Wstub
%%%%%          --------
%%%%%         | xxxxxx |  <--the resistance of this part
%%%%%   Lstub |________|  <--(ie from contacts to each strip 'P')
%%%%%         |  |  |  |
%             |P |  |P |
%             |  |  |  |
%             |  |  |  |
%             |  |  |  |

%%%% All dimensions in um unless noted

%%%% Parameters to be passed into the function
%Wstub = 12.3;
%Lstub = 4.75;
%Wm1g = 2.9;

%%%% Parameters
cont_poly = 0.5;
cont_poly_center = 1.2;
cont_cont_center = 1.4;
leg_width = 4.1;
Gpoly = 1/32;
Gshort = 10000;
contact_node = 1;
leg_node = 2;
leg_width = 4.1;
cont_size = 0.7;
cont_poly = 0.5;
cont_metal1 = 0.4;
cont_metal1_nsc = 0.7;
cont_cont = 0.7;
```

```
cont_cont_center = 1.4;
via_size = 0.7;
via1_metal1 = 0.4;
via1_metal2 = 0.4;
via1_via1 = 0.7;
via1_via1_center = 1.4;
Wstub = 12.3;

%%%%% Calculate length of poly stub on gate.  This is fixed, until the metal1
%%%%% gate runners are increased beyond thier current minimum width, 2.9u

Lstub = 4.75 + (2.9 - Wm1g);  % simple,hehe.  4.75 is the current stub width, 2.9 is current metal1 width

% determine number of contacts per cell
cont_columns = floor(((Wstub-2*cont_poly-2*.5*cont_size)/1.4)+1);
cont_rows = floor(((0.5*(Wm1g-2.9))/cont_cont_center)+1);

% find locations of end of first leg and start of second leg
first_leg_end = leg_width;
second_leg_start = Wstub-leg_width;

% Physical location of first via (upper left corner via)
cont_x = (Wstub - (cont_columns-1)*cont_cont_center)/2;
cont_y = cont_poly + 0.5*cont_size;

cont_columns = int16(cont_columns);
cont_rows = int16(cont_rows);


%%%%% Find rows and columns of poly
grid = 0.6;
%grid = 0.36;
poly_columns = int16(round(Wstub/grid));
poly_rows = int16(round(Lstub/grid));

%%%%% Locate vias and contacts on the grid
grid_cont = int16(cont_cont_center/grid);  % number of grid spaces center-to-center per via (vertical and ho
extra_columns = poly_columns - (cont_columns-1)*grid_cont - 1;
if rem(extra_columns,2)==0;
    first_cont_column = extra_columns/2 + 1;
else
    first_cont_column = floor(extra_columns/2);
end

%first_cont_column = int16((cont_x/grid));  %first column contact appears in
last_cont_column = int16(first_cont_column+((cont_columns-1)*grid_cont));
first_cont_row = int16((cont_y/grid));  % first row contact appears in
last_cont_row = int16(first_cont_row+((cont_rows-1)*grid_cont));
```

```
%%%% Locate legs on the grid
leg_end_chk = int16((first_leg_end/grid));
leg_start_chk = poly_columns - int16((second_leg_start/grid));

if leg_end_chk < leg_start_chk
    leg_width = leg_end_chk;
else
    leg_width = leg_start_chk;
end

leg_end_column = leg_end_chk;
leg_start_column = poly_columns-leg_end_chk;




%%%%% build node matrix
stub_nodes = zeros(poly_rows,poly_columns);


% Add the contacts first

for m = first_cont_row:1:last_cont_row
    for n = first_cont_column:1:last_cont_column
        if m >= first_cont_row & rem((m-first_cont_row),grid_cont)==0
            if n>= first_cont_column & rem((n-first_cont_column),grid_cont)==0
                stub_nodes(m,n) = contact_node;
            end
        end
    end
end

% Add the leg terminals
for n = 1:1:poly_columns
    if n <= leg_end_column | n > leg_start_column
        stub_nodes(poly_rows,n) = leg_node;
    end
end

%spy(stub_nodes)

%Add the rest of the nodes

nodecount = 3;
for m = 1:1:poly_rows
    for n = 1:1:poly_columns
        if stub_nodes(m,n) == 0
            stub_nodes(m,n) = nodecount;
```

```
            nodecount = nodecount + 1;
        end
    end
end

%%% create the conductance matrix

 Gmatrix_stub = zeros(nodecount-1);

 for m = 1:1:poly_rows
    for n = 1:1:poly_columns

        %%% add the horizontal conductances of the poly sheet
        if stub_nodes(m,n) ~= leg_node
            if n == 1
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + Gpoly;
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1))-Gpoly;
                Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n))-Gpoly;
            elseif n == poly_columns
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + Gpoly;
            elseif stub_nodes(m,n-1) == leg_node
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + 2*Gpoly;
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1))-Gpoly;
                Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n))-Gpoly;
                Gmatrix_stub(stub_nodes(m,n-1),stub_nodes(m,n-1)) =
                ...Gmatrix_stub(stub_nodes(m,n-1),stub_nodes(m,n-1)) + Gpoly;
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n-1)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n-1))-Gpoly;
                Gmatrix_stub(stub_nodes(m,n-1),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n-1),stub_nodes(m,n))-Gpoly;
            elseif stub_nodes(m,n+1) == leg_node
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + 2*Gpoly;
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1))-Gpoly;
                Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n))-Gpoly;
                Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n+1)) =
                ...Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n+1))+Gpoly;
            else
                Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
                ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + 2*Gpoly;
```

```
            Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1)) =
            ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n+1))-Gpoly;
            Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n)) =
            ...Gmatrix_stub(stub_nodes(m,n+1),stub_nodes(m,n))-Gpoly;
        end
    end

    %%% add the vertical conductances of the poly sheet

    if m == 1
        Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
        ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + Gpoly;
        Gmatrix_stub(stub_nodes(m,n),stub_nodes(m+1,n)) =
        ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m+1,n))-Gpoly;
        Gmatrix_stub(stub_nodes(m+1,n),stub_nodes(m,n)) =
        ...Gmatrix_stub(stub_nodes(m+1,n),stub_nodes(m,n))-Gpoly;
    elseif m == poly_rows
        Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
        ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + Gpoly;
    else
        Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
        ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + 2*Gpoly;
        Gmatrix_stub(stub_nodes(m,n),stub_nodes(m+1,n)) =
        ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m+1,n))-Gpoly;
        Gmatrix_stub(stub_nodes(m+1,n),stub_nodes(m,n)) =
        ...Gmatrix_stub(stub_nodes(m+1,n),stub_nodes(m,n))-Gpoly;
    end


    %%% add a connection to ground at the leg nodes
    if stub_nodes(m,n) == leg_node
        Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) =
        ...Gmatrix_stub(stub_nodes(m,n),stub_nodes(m,n)) + Gshort;
    end

  end
end

stub_current = zeros(nodecount-1,1);
stub_current(1,1) = 1;
volts = Gmatrix_stub\stub_current;
rstub = volts(1);
%stub_nodes;

end
```

A.2 Transformer Synthesis Code

```
%clear all;
%clc;


function [A] = trials_buildervf_self_only();

clear Coils;
clear A;

%%%%% 28 Oct 2008:  This code adds in the opportunity for coils of
%%%%% different outer radii.  Oh, and jackie's a biatch.


%%%%% 15 Oct 2008:  This code builds the trials matrix A for the trans_calc
%%%%% matlab file.
od_min = 2;
od_max = 4.5;  % maximum radius defining the max_od of the coil
r_step = .1;
r_max_cases = floor((od_max-od_min)/r_step);

r_min = .25;  % minumum radius, defines the min_id of the coil
gap_min = 0.16; %minimum separation between turns (about 6 mils)
w_min = 0.16; %minimum width of a turn (about 6 mils)
max_turns = 10;%maximum number of turns allowed on a coil
min_turns = 1;
gap_step = 0.1;
w_step = 0.1;

%%%% build each set of n-turn coils.  The code below creates all the n-turn
%%%% coils that satistfy the minimum spacing and maximum size... The outer
%%%% loop starts by fixing the number of turns.  Since 1 turn is a special
%%%% case (there are no inter-winding gaps because there is only one
%%%% winding), the code shunts to a while loop that deals with the 1-turn
%%%% case explicitly.  The while loop starts with the minimum trace width
%%%% coil.  The outer radius is set to r_max.  The inner radius is
%%%% calculated by subtracting the current width.  Each width results in a
%%%% two column entry in the array located in the cell array Coils{1}().
%%%% Once the width of the single turn is large enough that it encroaches
%%%% on the minimum inner radius, the flag is set and the program falls
%%%% through the while loop.  It then returns to the top loop, increments
%%%% turns to 2 and this time falls into code which first sets the gap size
%%%% (starting with the minimum gap).  Once the gap is set, a while loop
%%%% similar to the one described above is entered.  This loop sets the
%%%% turn width, then writes out the coil description to an array which has
%%%% 2*turns columns and an indeterminite number of rows located in the
%%%% cell array Coils{turns}().  The width of each turn is successively
```

```
%%%% increased until the minimum inner radius condition can't be met.  The
%%%% flag is then set and the program returns to the while loop that checks
%%%% the gap condition.  The gap count is incremented, increasing the gap
%%%% width by gap_step and then the code re-enters the while loop to make
%%%% the next set of coils corresponding to all the trace widths that will
%%%% fit.  This process continues until the gap is large enough that
%%%% minimum-width turns cannot fit inside the coil.  Then the gap flag is
%%%% set and the code returns to the main for loop. At this point, the
%%%% number of turns is incremented and the process repeats. The result is
%%%% a cell array that contains an array of coils for each turn-number
%%%% (i.e. if the min. number of turns is 1, and the max number is 10,
%%%% Coils{}() wil contain 10 arrays.  Each array contains all the possible
%%%% coils for that number of turns.

for turns = min_turns:1:max_turns
    n_coil = 1;
    if turns == 1
        for r_count = 1:1:r_max_cases
            r_max = od_min + r_step*(r_count-1);
            w_fits = 1;
            w_count = 0;
            while w_fits == 1
                width = w_min + w_step*w_count;
                if r_max - width >= r_min
                    r_outer = r_max;
                    r_inner = r_max - width;
                    Coils{turns}(n_coil,1) = r_inner;
                    Coils{turns}(n_coil,2) = r_outer;
                    w_count = w_count + 1;
                    n_coil = n_coil + 1;
                else
                    w_fits = 0;
                end
            end
        end
    else
        for r_count = 1:1:r_max_cases
            r_max = od_min + r_step*(r_count-1);
            gap_fits = 1;
            gap_count = 0;
            while gap_fits == 1
                gap = gap_min + gap_step*gap_count;
                if r_max - w_min*turns - gap*(turns - 1) >= r_min
                    w_fits = 1;
                    w_count = 0;
                    while w_fits == 1
                        width = w_min + w_step*w_count;
                        if r_max - width*turns - gap*(turns-1) >= r_min
                            for coil_turn = 1:1:turns
```

```
                            r_outer = r_max - (width + gap)*(coil_turn-1);
                            r_inner = r_max - width*coil_turn -
                            ...gap*(coil_turn-1);
                            inner_r_column = 1 + 2*(coil_turn-1);
                            outer_r_column = 2 + 2*(coil_turn-1);
                            Coils{turns}(n_coil,inner_r_column) = r_inner;
                            Coils{turns}(n_coil,outer_r_column) = r_outer;
                        end
                        w_count = w_count + 1;
                        n_coil = n_coil + 1;
                    else
                        w_fits = 0;
                    end
                end
                gap_count = gap_count + 1;
            else
                gap_fits = 0;
            end
        end
    end
end

%%%%% Build the complete trials matrix, A.
z_prim = 0; % z-location of primary
z_sec = 1.5748; % z-location of secondary
h_prim = 0.036;
h_sec = 0.036;
windings = 1; % now we're just dealing with one coil

%pre-allocate the A-matrix
columns = windings*max_turns*4; % the total number of columns
%%%now we need the total number of rows...
% rows = 0;
% for prim_turns = min_turns:1:max_turns
%     for sec_turns = prim_turns:1:max_turns
%         if prim_turns == sec_turns
%             tot = 0;
%             [rows_prim cols_prim] = size(Coils{prim_turns});
%             for i = 1:1:rows_prim
%                 for j = i:1:rows_prim
%                     tot = tot + 1;
%                 end
%             end
%             rows = rows + tot;
%         else
%             int_sum = 0;
%             [r_prim c_prim] = size(Coils{prim_turns});
%             [r_sec c_sec] = size(Coils{sec_turns});
```

```
%             int_sum = r_prim*r_sec;
%             rows = rows + int_sum;
%         end
%     end
% end
% rowmem = rows;

%%% now finding the total rows is easy
rows = 0;
for prim_turns = min_turns:1:max_turns
    [rows_prim cols_prim] = size(Coils{prim_turns});
    rows = rows + rows_prim;
end
rowmem = rows;


A = zeros(rows,columns);
%A = zeros(29161,16);

% Building the A-matrix isn't so hard either....
n_coil = 1;
for prim_turns = min_turns:1:max_turns
    [rows_prim cols_prim] = size(Coils{prim_turns});
    for prim_coil = 1:1: rows_prim
        for coil_turn = 1:1:prim_turns
            A(n_coil,1+4*(coil_turn-1)) = z_prim;
            A(n_coil,2+4*(coil_turn-1)) = h_prim;
            A(n_coil,3+4*(coil_turn-1)) =
            ...Coils{prim_turns}(prim_coil,1+2*(coil_turn-1));
            A(n_coil,4+4*(coil_turn-1)) =
            ...Coils{prim_turns}(prim_coil,2+2*(coil_turn-1));
        end
    n_coil = n_coil + 1;
    end
end

total_coils = n_coil - 1


filename = strcat('fast_henry_runs_',date,'.txt');
fid = fopen(filename, 'wt'); %open a file to write the FastHenry deck for
%%%each candidate geometry

start_file = 1;
stop_file = 152;

%count(1,:) = start_file:1:stop_file;
%count(2,:) = start_file:1:stop_file;
```

```
for count = start_file:1:stop_file
    input_file = strcat('geom',num2str(count),'_',date,'.inp');
    suffix = strcat('_geom',num2str(count));
    command = ['fasthenry ' input_file ' -S ' suffix];
    fprintf(fid,command);
    fprintf(fid,'\n');
end

fclose(fid);


function [X Y nodes] = circle_nodes(r1,r2,xcenter,ycenter)

theta = 0:0.25:2*pi;

X = (r1+(r2-r1)/2)*cos(theta) + xcenter;
Y = (r1+(r2-r1)/2)*sin(theta) + ycenter;

[m nodes] = size(X);



%%% find good cases
clear matches11;
clear Amatch11;
clear Lex11 turns11 large_d11;
clear mathces22;
clear Amatch22;
clear Lex22 turns22 large_d22;
clc;

[row column] = size(L11);

%LM_scale = 1e9*LM;
L11_scale = 1e9*L11;
%L22_scale = 1e9*L22;


%LM_scale = [5 11.059 7 11.059];
%L11_scale = [16.873 16.873 7 16.873];
%L22_scale = [36.389 36.389 36.389 7];
%[row column]= size(LM_scale);

%LM_d = 11.8;
L11_d = 11.8176;
L22_d = 46.9506;


counter11 = 1;
counter22 = 1;
```

```
tolm = .1;
tol11 = .1;
tol22 = .5;

for i = 1:1:column
    if (L11_scale(i) > L11_d-tol11 && L11_scale(i) < L11_d+tol11)
        matches11(counter11) = i;
        counter11 = counter11 + 1;
    end
    if (L11_scale(i) > L22_d-tol22 && L11_scale(i) < L22_d+tol22)
        matches22(counter22) = i;
        counter22 = counter22 + 1;
    end
end

% populate the inductance matches and corresponding turns and diameters for
% L11
[r11 c11] = size(matches11);
[r22 c22] = size(matches22);
loc = 1;
large_d11 = 0;
large_d22 = 0;
for i = 1:1:c11
    loc = matches11(i);
    Lex11(i) = L11_scale(loc);
    turns11(i) = prim_turns(loc);
    biggest_r = 0;
    for j = 1:1:prim_turns(loc);
        if A_mm(loc,j*4) >= biggest_r;
            biggest_r = A_mm(loc,j*4);
        end
        large_d11(i) = 2*biggest_r;
    end
    tw11(i) = A_mm(loc,4)-A_mm(loc,3);
    if turns11(i) > 1
        gap11(i) = A_mm(loc,3)-A_mm(loc,8);
    else
        gap11(i) = 0;
    end
   % Lex11(i,2) = L22_scale(matches11(i));
    %Lex11(i,3) = LM_scale(matches11(i));
end

% populate the inductance matches and corresponding turns and diameters
% for L22.
loc = 1;
for i = 1:1:c22
    loc = matches22(i);
    Lex22(i) = L11_scale(loc);
```

```
    turns22(i) = prim_turns(loc);
    biggest_r = 0;
    for j = 1:1:prim_turns(loc)
        if A_mm(loc,j*4) >= biggest_r
            biggest_r = A_mm(loc,j*4);
        end
        large_d22(i) = 2*biggest_r;
    end
    tw22(i) = A_mm(loc,4)-A_mm(loc,3);
    if turns22(i) > 1
        gap22(i) = A_mm(loc,3)-A_mm(loc,8);
    else
        gap22(i) = 0;
    end
end

for i = 1:1:c11
    Amatch11(i,:) = A_mm(matches11(i),:);
end
for i = 1:1:c22
    Amatch22(i,:) = A_mm(matches22(i),:);
end

% figure(100)
% subplot(3,1,1)
% a=plot(Lex11);
% set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
% title('Coils Matching L_{11}=11.8nH');
% grid on;
% xlabel('Match Number');
% ylabel('Self Inductance [nH]');
% axis([0 25 11.6 12])
% %legend('L11');
% subplot(3,1,2)
% b=plot(turns11);
% set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
% xlabel('Match Number');
% ylabel('Turns');
% grid on;
% axis([0 25 0.9 2.1]);
% subplot(3,1,3);
% c=plot(large_d11);
% set(c,'linewidth',2,'color',[0.597 0.199 0.199]);
% grid on;
% xlabel('Match Number');
% ylabel('Diameter [mm]');
% axis([0 25 3.9 10]);

figure(100)
```

```
subplot(5,1,1)
a=plot(Lex11);
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
title('Coils Matching L_{11}=11.8nH');
grid on;
%xlabel('Match Number');
ylabel('Self Inductance [nH]');
axis([0 25 11.6 12])
%legend('L11');
subplot(5,1,2)
b=plot(turns11);
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
%xlabel('Match Number');
ylabel('Turns');
grid on;
axis([0 25 0.9 2.1]);
subplot(5,1,3)
b=plot(tw11);
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
%xlabel('Match Number');
ylabel('Trace Width [mm]');
grid on;
subplot(5,1,4)
b=plot(gap11);
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
%xlabel('Match Number');
ylabel('Gap [mm]');
grid on;
subplot(5,1,5);
c=plot(large_d11);
set(c,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on;
xlabel('Match Number');
ylabel('Diameter [mm]');
axis([0 25 3.9 10]);


figure(101)
subplot(5,1,1)
a=plot(Lex22);
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
title('Coils Matching L_{22}=47nH');
%legend('L22');
grid on;
%xlabel('Match Number');
ylabel('Self Inductance [nH]');
axis([0 55 46 47.5])
%legend('L22');
subplot(5,1,2)
```

```
b=plot(turns22);
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
%xlabel('Match Number');
ylabel('Turns');
grid on;
axis([0 55 1.9 4.1]);
subplot(5,1,3)
b=plot(tw22);
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
%xlabel('Match Number');
ylabel('Trace Width [mm]');
grid on;
axis([0 55 0 1])
subplot(5,1,4)
b=plot(gap22);
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
%xlabel('Match Number');
ylabel('Gap [mm]');
grid on;
axis([0 55 0 2]);
subplot(5,1,5);
c=plot(large_d22);
set(c,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on;
xlabel('Match Number');
ylabel('Diameter [mm]');
axis([0 55 4 10]);



[junk1,junk2]=system(['mkdir ' date]); %create a directory with named by the date
save_dir = date; %define this as the save directory

version = '_v1p0_';
file_prefix = 'Lself_extract';
high_flag = 0;
file_num = 1;

%check to see if a file already exists, if not create it, if so, create the
%next highest file number
while high_flag == 0
    file_exists = strcat('./',save_dir,'/',file_prefix ,
    ...version, num2str(file_num),'.mat');
    [a b] = system(['ls ' file_exists]);
    if a == 0
         file_num = file_num + 1;
    else
       filename = strcat(file_prefix, version, num2str(file_num));
       fileloc = strcat('./',date,'/',filename);
```

```
        save(fileloc,'Amatch11','Amatch22','Lex11','turns11','large_d11','Lex22','turns22','large_d22','gap11
         high_flag = 1;
         file_num = file_num + 1;
    end
end



%%% find good cases
clear matches11;
clear Amatch11;
clear Lex11 turns11 large_d11;
clear mathces22;
clear Amatch22;
clear Lex22 turns22 large_d22;
clc;

[row column] = size(L11);

%LM_scale = 1e9*LM;
L11_scale = 1e9*L11;
%L22_scale = 1e9*L22;


%LM_scale = [5 11.059 7 11.059];
%L11_scale = [16.873 16.873 7 16.873];
%L22_scale = [36.389 36.389 36.389 7];
%[row column]= size(LM_scale);

%LM_d = 11.8;
L11_d = 11.8176;
L22_d = 46.9506;


counter11 = 1;
counter22 = 1;
tolm = .1;
tol11 = .1;
tol22 = .5;

for i = 1:1:column
    if (L11_scale(i) > L11_d-tol11 && L11_scale(i) < L11_d+tol11)
        matches11(counter11) = i;
        counter11 = counter11 + 1;
    end
    if (L11_scale(i) > L22_d-tol22 && L11_scale(i) < L22_d+tol22)
        matches22(counter22) = i;
        counter22 = counter22 + 1;
    end
end
```

```
% populate the inductance matches and corresponding turns and diameters for
% L11
[r11 c11] = size(matches11);
[r22 c22] = size(matches22);
loc = 1;
large_d11 = 0;
large_d22 = 0;
for i = 1:1:c11
    loc = matches11(i);
    Lex11(i) = L11_scale(loc);
    turns11(i) = prim_turns(loc);
    biggest_r = 0;
    for j = 1:1:prim_turns(loc);
        if A_mm(loc,j*4) >= biggest_r;
            biggest_r = A_mm(loc,j*4);
        end
        large_d11(i) = 2*biggest_r;
    end
   % Lex11(i,2) = L22_scale(matches11(i));
    %Lex11(i,3) = LM_scale(matches11(i));
end

% populate the inductance matches and corresponding turns and diameters
% for L22.
loc = 1;
for i = 1:1:c22
    loc = matches22(i);
    Lex22(i) = L11_scale(loc);
    turns22(i) = prim_turns(loc);
    biggest_r = 0;
    for j = 1:1:prim_turns(loc)
        if A_mm(loc,j*4) >= biggest_r
            biggest_r = A_mm(loc,j*4);
        end
        large_d22(i) = 2*biggest_r;
    end
end

for i = 1:1:c11
    Amatch11(i,:) = A_mm(matches11(i),:);
end
for i = 1:1:c22
    Amatch22(i,:) = A_mm(matches22(i),:);
end

figure(100)
subplot(3,1,1)
plot(Lex11);
```

```
title('Data for L11');
grid on;
xlabel('Match Number');
ylabel('Self Inductance [nH]');
legend('L11');
subplot(3,1,2)
plot(turns11);
xlabel('Match Number');
ylabel('Turns');
grid on;
subplot(3,1,3);
plot(large_d11);
grid on;
xlabel('Match Number');
ylabel('Diameter [mm]');

figure(101)
subplot(3,1,1)
plot(Lex22);
title('Data for L22');
legend('L22');
grid on;
xlabel('Match Number');
ylabel('Self Inductance [nH]');
legend('L22');
subplot(3,1,2)
plot(turns22);
xlabel('Match Number');
ylabel('Turns');
grid on;
subplot(3,1,3);
plot(large_d22);
grid on;
xlabel('Match Number');
ylabel('Diameter [mm]');



[junk1,junk2]=system(['mkdir ' date]); %create a directory with named by the date
save_dir = date; %define this as the save directory

version = '_v1p0_';
file_prefix = 'Lself_extract';
high_flag = 0;
file_num = 1;

%check to see if a file already exists, if not create it, if so, create the
%next highest file number
while high_flag == 0
```

```
    file_exists = strcat('./',save_dir,'/',file_prefix ,
    ...version, num2str(file_num),'.mat');
    [a b] = system(['ls ' file_exists]);
    if a == 0
         file_num = file_num + 1;
    else
       filename = strcat(file_prefix, version, num2str(file_num));
       fileloc = strcat('./',date,'/',filename);
       save(fileloc,'Amatch11','Amatch22','Lex11','turns11','large_d11','Lex22','turns22','large_d22');
        high_flag = 1;
        file_num = file_num + 1;
    end
end



%%% This script will extract the useful 2-winding transformer designs after the
%%% script "self_to_mut.m" is run.  It provides two results matrices.  The
%%% first is one called "transformers" which is the input for the script
%%% "fasthenry2" that creates the fast-henry input files.  The other is
%%% "trans-parameters" which provides the inducatance matrix parameters,
%%% the primary and secondary turns, and the diameter for each design.
%%% Since the "self_to_mut" script finds designs that both conform to the
%%% reference coil separation distances (ie, standard pc-board thicknesses)
%%% as well as those that don't, a switch can be thrown to select for
%%% either case. When the switch "match_override" is set to zero, only
%%% designs which correspond to standard PCB thicknesses are considered
%%% valid.  When match_override=1, all designs with the correct inductance
%%% matrix parameters are taken as valid.

clear transformers
clear success_index

clc

units = 1e-3; %convert mm to meters
z_ref = [0.2032 0.9906 1.5748]*units;
[junk z_ref_size] = size(z_ref);
match_override = 0;

[junk tot_designs] = size(fail);
[rx, cx] = size(xformers);

max_prim_turns = 0;
max_sec_turns = 0;
counter = 1;
for geom = 1:1:tot_designs
    %see if the thickness of the current design matches any of the PCB
    %reference values
    zh_match = 0;
```

```
    for test = 1:1:z_ref_size
        if z_choice(geom) == z_ref(test)
            zh_match = 1;
        end
    end
    %find successfull designs, and put them into the vector "success_index"
    %which will contain the indicies of each successful design that appears
    %in the array "xformers."  Successful designs are determined by
    %considering the vector "fail" which was produced by self-to-mut and,
    %if the match_override switch is off, whether the z-height matches one
    %of the reference values in z_ref, which correspond to standard PCB
    %thicknesses
    if (fail(geom) == 0 && (zh_match == 1 || match_override == 1))
        success_index(counter) = geom;
        counter = counter + 1;
        prim_turns = turns11(xformers(geom,81));
        sec_turns = turns22(xformers(geom,82));
        if prim_turns > max_prim_turns
            max_prim_turns = prim_turns;
        end
        if sec_turns > max_sec_turns
            max_sec_turns = sec_turns;
        end
    end
end

[junk suc_designs] = size(success_index);
transformers = zeros(suc_designs,4*(max_prim_turns+max_sec_turns));

%build the array of transformers which will be turned into FastHenry decks
%for simulaton.  Transformers is formatted as follows:  each turn of the
%primary or secondary has four parameters: z, the z-axis position; h, the
%thickness of the turn (in the z-direction); r1, the inner diameter of the
%turn; r2, the outer diameter of the turn.  By default, the z-location of
%all primary turns is 0, the z-location of the secondary turns is
%determined by the value in the matrix, z_choice.  All primary turns must
%be listed before any secondary turns, though the order of the primary
%turns doesn't matter

for geom = 1:1:suc_designs
    prim_turns =turns11(xformers(success_index(geom),81)); %get primary turns for this successful geometry
    sec_turns = turns22(xformers(success_index(geom),82)); %get secondary turns for this successful geometry
    p2 = 4*prim_turns; %index of last primary turn data
    s1 = 4*prim_turns + 1; %index of first secondary turn data
    s2 = 4*sec_turns + p2; %index of last secondary turn data
    sx1 = (cx - 2)/2 + 1; %index of first secondary turn in xformers
    sx2 = sx1 - 1 + 4*sec_turns; %index of last secondary turn in xformers
    transformers(geom,1:p2) = xformers(success_index(geom),1:p2);
    transformers(geom,s1:s2) = xformers(success_index(geom),sx1:sx2);
```

```
    %put in z-locations for primary coils...they're defined as zero by
    %default
    z_prim = 0;
    for turn = 1:1:prim_turns
        transformers(geom,1+4*(turn - 1)) = z_prim;
    end

    %put in z-locations for secondary coils...they come from z_choice.
    z_sec = z_choice(success_index(geom));
    for turn = 1:1:sec_turns
        transformers(geom,s1+4*(turn-1)) = z_sec;
    end
end

%build an array of transformer parameters that gives the basic information
%of interest.  That is, it provides the inductance matrix parameters, L11,
%L22, and LM, the primary turns, secondary turns, and maximum diameter of
%the transformer (defined as the od of the largest turn).
trans_parms = zeros(suc_designs,5);
for geom = 1:1:suc_designs
    prim_turns =turns11(xformers(success_index(geom),81)); %get primary turns for this successful geometry
    sec_turns = turns22(xformers(success_index(geom),82)); %get secondary turns for this successful geometry
    L11 = Lex11(xformers(success_index(geom),81));
    L22 = Lex22(xformers(success_index(geom),82));
    Lmutual = LM(success_index(geom))*1e9;
    prim_diam = large_d11(xformers(success_index(geom),81));
    sec_diam = large_d22(xformers(success_index(geom),82));

    if prim_diam >= sec_diam
        diameter = prim_diam;
    else
        diameter = sec_diam;
    end

    trans_parms(geom,1) = L11;
    trans_parms(geom,2) = L22;
    trans_parms(geom,3) = Lmutual;
    trans_parms(geom,4) = prim_turns;
    trans_parms(geom,5) = sec_turns;
    trans_parms(geom,6) = diameter;
    trans_parms(geom,7) = prim_diam;
    trans_parms(geom,8) = sec_diam;
    trans_parms(geom,9) = xformers(success_index(geom),81);
    trans_parms(geom,10) = xformers(success_index(geom),82);
    trans_parms(geom,11) = z_choice(success_index(geom));
    trans_parms(geom,12) = tw11(xformers(success_index(geom),81));
    trans_parms(geom,13) = tw22(xformers(success_index(geom),82));
    trans_parms(geom,14) = gap11(xformers(success_index(geom),81));
```

```
    trans_parms(geom,15) = gap22(xformers(success_index(geom),82));
end
saveflag = 0;

if saveflag == 1
    [junk1,junk2]=system(['mkdir ' date]); %create a directory
    %%with named by the date
    save_dir = date; %define this as the save directory

    version = '_v1p0_';
    file_prefix = 'Extract_self_mut';
    high_flag = 0;
    file_num = 1;

    %check to see if a file already exists, if not create it, if so, create the
    %next highest file number
    while high_flag == 0
        file_exists = strcat('./',save_dir,'/',file_prefix ,
        ...version, num2str(file_num),'.mat');
        [a b] = system(['ls ' file_exists]);
        if a == 0
             file_num = file_num + 1;
        else
           filename = strcat(file_prefix, version, num2str(file_num));
           fileloc = strcat('./',date,'/',filename);
           save(fileloc,'trans_parms','transformers','success_index');
           high_flag = 1;
           file_num = file_num + 1;
        end
    end
end

% figure(101)
% subplot(5,1,1)
% a=plot(Lex22);
% set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
% title('Coils Matching L_{22}=47nH');
% %legend('L22');
% grid on;
% %xlabel('Match Number');
% ylabel('Self Inductance [nH]');
% axis([0 55 46 47.5])
% %legend('L22');
% subplot(5,1,2)
% b=plot(turns22);
% set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
% %xlabel('Match Number');
% ylabel('Turns');
% grid on;
```

```
% axis([0 55 1.9 4.1]);
% subplot(5,1,3)
% b=plot(tw22);
% set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
% %xlabel('Match Number');
% ylabel('Trace Width [mm]');
% grid on;
% axis([0 55 0 1])
% subplot(5,1,4)
% b=plot(gap22);
% set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
% %xlabel('Match Number');
% ylabel('Gap [mm]');
% grid on;
% axis([0 55 0 2]);
% subplot(5,1,5);
% c=plot(large_d22);
% set(c,'linewidth',2,'color',[0.597 0.199 0.199]);
% grid on;
% xlabel('Match Number');
% ylabel('Diameter [mm]');
% axis([0 55 4 10]);

[ref junk] = size(trans_parms);
plotvar = 1:ref;
figure(102)
subplot(6,1,1)
title('Transformer Designs with Analytical L-matrix Match');
a=plot(trans_parms(:,1));
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
c=plot(trans_parms(:,3));
set(c,'linewidth',2,'color',[0.4 0.4 0.4]);
legend('L_{11}','L_M');
ylabel('Inductance [nH]');
grid on;
subplot(6,1,2)
b=plot(trans_parms(:,2));
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
ylabel('Inductance [nH]');
legend('L_{22}');
grid on;
subplot(6,1,3)
a=plot(plotvar,trans_parms(:,7),plotvar,trans_parms(:,8));
legend('Prim.','Sec.');
set(a(1),'linewidth',2,'color',[0.597 0.199 0.199]);
set(a(2),'linewidth',2,'color',[0.4 0.4 0.4]);
grid on;
ylabel('Diameter [mm]');
```

```
subplot(6,1,4)

a=plot(plotvar,trans_parms(:,4),plotvar,trans_parms(:,5));
ylabel('Turns');
set(a(1),'linewidth',2,'color',[0.597 0.199 0.199]);
set(a(2),'linewidth',2,'color',[0.4 0.4 0.4]);
legend('Prim.','Sec.');
grid on;

subplot(6,1,5)
a=plot(plotvar,trans_parms(:,12),plotvar,trans_parms(:,13));
ylabel('Trace Width [mm]');
set(a(1),'linewidth',2,'color',[0.597 0.199 0.199]);
set(a(2),'linewidth',2,'color',[0.4 0.4 0.4]);
legend('Prim.','Sec.');
grid on;

subplot(6,1,6)
a=plot(plotvar,trans_parms(:,14),plotvar,trans_parms(:,15));
ylabel('Gap [mm]');
set(a(1),'linewidth',2,'color',[0.597 0.199 0.199]);
set(a(2),'linewidth',2,'color',[0.4 0.4 0.4]);
legend('Prim.','Sec.');
grid on;
xlabel('Successful Geometry Number');


%%%% function to read in jacked-up fasthenry matlab files...baaaah!!!!
%clear all;
clc;
clear A;

file_count = 1;
dirloc = './03-Nov-2008/fast_henry_runs/'; %directory where files are located
filename = 'Zc_geom145_junk.mat'; %name of file to process
fileloc = strcat(dirloc,filename);
fid = fopen(fileloc);


%%%% read file into a cell array.  Each string will be read into one row in
%%%% the first array (C{1}). Strings are separated by whitespace, so there
%%%% will be one row for each group of contiguous characters.
C = textscan(fid,'%s');
file_count = 1; %note, this can't be changed, it is determined by the textscan
%function which creates a cell for each conversion specifier...here there
%is one, %s.

%find the indicies (actually, row numbers) where the word "frequency"
%appears.  These mark the start of a FastHenry output matrix for the given
%frequency
```

```
freqs = strmatch('frequency',C{file_count});

%now for each frequency, build the impedance matrix, nominally an L-R
%matrix which is nxn rows where n is the total number of turns (p+s)
%simulated.
for runs = 1:1:size(freqs)
    fsw(runs) = str2double(C{file_count}(freqs(runs)+2)); %get simulation frequency
    tot_rows(runs) = str2double(C{file_count}(freqs(runs)+3)); %find the total #of rows
    tot_columns(runs) = str2double(C{file_count}(freqs(runs)+5)); %find the total #of columns

    if tot_rows(runs) == tot_columns(runs)
        % put the impedance values into the cell array, A
        for row = 1:1:tot_rows(runs)
            for column = 1:1:tot_columns(runs)
                real_temp = str2double(C{file_count}(freqs(runs)+6+
                ...2*(column-1)+2*tot_columns(runs)*(row-1)));
                imag_temp = str2double(C{file_count}(freqs(runs)+7+
                ...2*(column-1)+2*tot_columns(runs)*(row-1)))/(fsw(runs)*2*pi);
                A{runs}(row,column) = real_temp + imag_temp;
            end
        end
        %Lself(runs) = imag(sum(sum(A{runs})));
        %Rself(runs) = real(sum(sum(A{runs})));
    else
        disp('The L-R matrix is not symmetric, this crap is broken!')
        break;
    end
end

fclose(fid);
```

```
%%%% function to read in jacked-up fasthenry matlab files...baaaah!!!!
%clear all;
clc;
clear A;

file_count = 1;
filename = 'Zc_geom1.mat';
fid = fopen(filename);

C = textscan(fid,'%s');

freqs = strmatch('frequency',C{file_count});

for runs = 1:1:size(freqs)
```

```
    fsw = str2double(C{file_count}(freqs(runs)+2));
    tot_rows = str2double(C{file_count}(freqs(runs)+3));
    tot_columns = str2double(C{file_count}(freqs(runs)+5));

    for row = 1:1:tot_rows
        for column = 1:1:tot_columns
            real_temp = str2double(C{file_count}(freqs(runs)+6+
            ...2*(column-1)+2*tot_columns*(row-1)));
            imag_temp = str2double(C{file_count}(freqs(runs)+7+
            ...2*(column-1)+2*tot_columns*(row-1)))/(fsw*2*pi);
            A{runs}(row,column) = real_temp + imag_temp;

        end
    end
    Lself(runs) = imag(sum(sum(A{runs})));
    Rself(runs) = real(sum(sum(A{runs})));
end
```

```
%%% FastHenry Script

clc;
%clear all;


%   A = [0 0.0001 0.03 0.05 0 0.0001 0.06 0.09 0.001 0.0001 0.01 0.012 0.001 0.0001 0.017 0.025;
%        0 0.00157 0.003 0.005 0 0.00157 0.008 0.009 0.001 0.00157 0.003 0.005 0.001 0.00157 0.008 0.009;
%        0 0.00157 0.003 0.005 0.001 0.00157 0.003 0.005 0 0 0 0 0 0 0 0;
%        0 0.00157 0.003 0.005 0 0.00157 0.008 0.009 0.001 0.00157 0.003 0.005 0 0 0 0
%        0 0.0001 0.004 0.006 0 0.0001 0.009 0.010 0.001 0.0001 0.004 0.006 0.001 0.0001 0.009 0.010];
%Amatch = [0 0.0001 0.002 0.0035 0 0.0001 0.004 0.006 0 0.0001 0.007 0.009]*1e3;

%Amatch = [0 3.6e-5 0.0018 0.0020]*1e3;
%Amatch = [0 3.6e-5 0.0009 0.0020]*1e3;
     %A = [0 0.0001 0.002 0.0035 0 0.0001 0.004 0.006 0 0.0001 0.007 0.009]

make_plots = 0;
system(['mkdir ' './',date,'/fast_henry_runs']);
dirloc = strcat('./',date,'/fast_henry_runs/');
Amatch = transformers*1e3;

[m,n] = size(Amatch);

% find the location of the last ring that located in the same plane as the
```

```
% first winding.  Note: to define two windings, all the rings in the first
% winding must be defined before any of the rings in the second winding are
% defined.  Otherwise this crap will blow up....


p = zeros(1,m);
for i = 1:1:m
    for j = 1:1:n
        if Amatch(i,j)~=0
            terms(i) = j; % find the total terms to identify the # of rings
        end
    end
end
a=terms/4; %this is the number of independent rings in each row

units = 1; %scaling for mm units


%%%loop here to cycle through each valid geometry%%%%
% start_node = 1;
% start_element = 1;
% geom = 1;

for geom = 1:1:m
    start_node = 1;
    start_element = 1;

    filename = strcat(dirloc,'geom',num2str(geom),'_',date,'.inp');
    fid = fopen(filename, 'wt'); %open a file to write the FastHenry deck for
    %%%each candidate geometry
    fprintf(fid, '**** FastHenry deck for geometry #%1.0f ****\n',geom);
    fprintf(fid, '**** Created by fasthenry2.m v0.2 on %s ****\n',date);
    fprintf(fid, '**** Anthony Sagneri **********\n\n');
    fprintf(fid, '.Units MM\n');
    fprintf(fid, '.Default nhinc = 6 nwinc=12\n\n');


    %%% find the ring with the largest outer diameter
    large_radius = 0;
    for i=1:1:a(geom)
        if Amatch(geom,4*(i-1)+4) > large_radius
            large_radius = Amatch(geom,4*(i-1)+4);
        end
    end

    %%% choose the center point of the rings so that the largest diameter is
    %%% fully in the simulation space
    c1 = large_radius %+ 0.005;
    c2 = large_radius %+ 0.005;
```

```
for i=1:1:a(geom)
    z1 = Amatch(geom,4*(i-1)+1); % z-location of coil 1
    h1 = Amatch(geom,4*(i-1)+2); % thickness of coil 1
    r1 = Amatch(geom,4*(i-1)+3); % inner radius of coil 1
    r2 = Amatch(geom,4*(i-1)+4); % outer radius of coil 1

    [X Y nodes] = circle_nodes(r1*units,r2*units,c1*units,c2*units);
    Nodes = zeros(nodes,4);
    Nodes(:,1) = start_node:1:(start_node-1 + nodes);
    Nodes(:,2) = X;
    Nodes(:,3) = Y;
    Nodes(1:nodes,4) = z1*units;

    Elements = zeros(nodes-1,5);
    Elements(:,1) = start_element:1:(start_element-1 + nodes-1);
    Elements(:,2) = Nodes(1:(nodes-1),1);
    Elements(:,3) = Nodes(2:nodes,1);
    Elements(1:nodes-1,4) = (r2-r1)*units;
    Elements(1:nodes-1,5) = h1*units;


    nodes_t = Nodes';
    start_node = start_node + nodes;
    elements_t = Elements';
    start_element = start_element + nodes - 1;
    node_record(i) = nodes;

    if make_plots == 1
        figure(i);
        plot(Nodes(:,2),Nodes(:,3));
    end

    fprintf(fid, '\n************************************************\n');
    fprintf(fid, '** Node and Element Definition for Ring Number %0.0f**\n', i);
    fprintf(fid, '************************************************\n\n\n');
    fprintf(fid, '**********NODE LIST**********\n');
    fprintf(fid, 'N%0.0f x=%0.4f y=%0.4f z=%0.4f\n',nodes_t);
    fprintf(fid, '\n\n**********ELEMENT LIST**********\n');
    fprintf(fid, 'E%0.0f N%0.0f N%0.0f w=%0.4f h=%0.4f\n',elements_t);
end

for i = 1:1:a(geom)
    if i == 1
        Ports(i,1) = 1;
        Ports(i,2) = Ports(i,1)+ node_record(i)-1;
    else
```

```
            Ports(i,1) = node_record(i-1)+Ports(i-1,1);
            Ports(i,2) = Ports(i,1) + node_record(i)-1;
        end
    end
    ports_t = Ports';

    fprintf(fid, '\n\n*************************************************\n');
    fprintf(fid, '***********NETWORK PORT DEFINITIONS****************\n');
    fprintf(fid, '*************************************************\n\n');
    fprintf(fid, '.external N%0.0f N%0.0f\n',ports_t);

    fprintf(fid, '\n\n*************************************************\n');
    fprintf(fid, '***********FREQUENCY RANGE*************************\n');
    fprintf(fid, '*************************************************\n\n');
    fprintf(fid, '.freq fmin=75e6 fmax=75e6 ndec=1\n');
    fprintf(fid, '.end');

    fclose(fid);
end

% atest = [122 2.04456740 3.007686 4.04567476;
%                123 5.06  3.000 4.3443]';
%       fid = fopen('test.txt', 'wt');
%       fprintf(fid, 'N%0.0f x=%0.2f y=%0.2f z=%0.2f\n',atest);
%       fprintf(fid, '\nHow do I write an intervening space\n\n');
%       fprintf(fid, 'Oh, like that');




% r2 = 5;
% r1 = 3;
% xcenter = (r2-r1) + 0.5;
% ycenter = (r2-r1) + 0.5
%
% theta = 0:0.5:2*pi;
%
% X = (r2-r1)*cos(theta) + xcenter;
% Y = (r2-r1)*sin(theta) + ycenter;
%
% plot(X,Y)
% axis([0 5 0 5]);
%
% %[m n] = size(theta)



%%% FastHenry Script
```

```
clc;
%clear all;
clear Ports
clear ports_t

%   A = [0 0.0001 0.03 0.05 0 0.0001 0.06 0.09 0.001 0.0001 0.01 0.012 0.001 0.0001 0.017 0.025;
%        0 0.00157 0.003 0.005 0 0.00157 0.008 0.009 0.001 0.00157 0.003 0.005 0.001 0.00157 0.008 0.009;
%        0 0.00157 0.003 0.005 0.001 0.00157 0.003 0.005 0 0 0 0 0 0 0 0;
%        0 0.00157 0.003 0.005 0 0.00157 0.008 0.009 0.001 0.00157 0.003 0.005 0 0 0 0
%        0 0.0001 0.004 0.006 0 0.0001 0.009 0.010 0.001 0.0001 0.004 0.006 0.001 0.0001 0.009 0.010];
%Amatch = [0 0.0001 0.002 0.0035 0 0.0001 0.004 0.006 0 0.0001 0.007 0.009]*1e3;

%Amatch = [0 3.6e-5 0.0018 0.0020]*1e3;
%Amatch = [0 3.6e-5 0.0009 0.0020]*1e3;
     %A = [0 0.0001 0.002 0.0035 0 0.0001 0.004 0.006 0 0.0001 0.007 0.009]

make_plots = 0;
system(['mkdir ' './',date,'/fast_henry_runs']);
dirloc = strcat('./',date,'/fast_henry_runs/');
Amatch = transformers*1e3;

[m,n] = size(Amatch);

% find the location of the last ring that located in the same plane as the
% first winding.  Note: to define two windings, all the rings in the first
% winding must be defined before any of the rings in the second winding are
% defined.  Otherwise this crap will blow up....


p = zeros(1,m);
for i = 1:1:m
    for j = 1:1:n
        if Amatch(i,j)~=0
            terms(i) = j; % find the total terms to identify the # of rings
        end
    end
end
%a=terms/4; %this is the number of independent rings in each row

units = 1; %scaling for mm units


%%%loop here to cycle through each valid geometry%%%%
% start_node = 1;
% start_element = 1;
% geom = 1;

for geom = 1:1:m
```

```
%%%%%%%%%%%%%% reset variables %%%%%%%%%%%%%%%
clear node_record;
clear Ports;
clear Nodes;
clear Elements;
clear elements_t;
clear nodes_t
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% initialize variables %%%%%%%%%%
start_node = 1;
start_element = 1;
a = terms(geom)/4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

filename = strcat(dirloc,'geom',num2str(geom),'_',date,'.inp');
fid = fopen(filename, 'wt'); %open a file to write the FastHenry deck for
%%%each candidate geometry
fprintf(fid, '**** FastHenry deck for geometry #%1.0f ****\n',geom);
fprintf(fid, '**** Created by fasthenry3.m v0.3 on %s ****\n',date);
fprintf(fid, '**** Anthony Sagneri ********************\n\n');
fprintf(fid, '.Units MM\n');
fprintf(fid, '.Default nhinc = 6 nwinc=12\n\n');


%%% find the ring with the largest outer diameter
large_radius = 0;
for i=1:1:a
    if Amatch(geom,4*(i-1)+4) > large_radius
        large_radius = Amatch(geom,4*(i-1)+4);
    end
end

%%% choose the center point of the rings so that the largest diameter is
%%% fully in the simulation space
c1 = large_radius %+ 0.005;
c2 = large_radius %+ 0.005;



for i=1:1:a
    z1 = Amatch(geom,4*(i-1)+1); % z-location of coil i
    h1 = Amatch(geom,4*(i-1)+2); % thickness of coil i
    r1 = Amatch(geom,4*(i-1)+3); % inner radius of coil i
    r2 = Amatch(geom,4*(i-1)+4); % outer radius of coil i

    [X Y nodes] = circle_nodes(r1*units,r2*units,c1*units,c2*units);
    Nodes = zeros(nodes,4);
    Nodes(:,1) = start_node:1:(start_node-1 + nodes);
```

```
    Nodes(:,2) = X;
    Nodes(:,3) = Y;
    Nodes(1:nodes,4) = z1*units;

    Elements = zeros(nodes-1,5);
    Elements(:,1) = start_element:1:(start_element-1 + nodes-1);
    Elements(:,2) = Nodes(1:(nodes-1),1);
    Elements(:,3) = Nodes(2:nodes,1);
    Elements(1:nodes-1,4) = (r2-r1)*units;
    Elements(1:nodes-1,5) = h1*units;


    nodes_t = Nodes';
    start_node = start_node + nodes;
    elements_t = Elements';
    start_element = start_element + nodes - 1;
    node_record(i) = nodes;

    if make_plots == 1
        figure(i);
        plot(Nodes(:,2),Nodes(:,3));
    end

    fprintf(fid, '\n*************************************************\n');
    fprintf(fid, '** Node and Element Definition for Ring Number %0.0f**\n', i);
    fprintf(fid, '*************************************************\n\n\n');
    fprintf(fid, '**********NODE LIST**********\n');
    fprintf(fid, 'N%0.0f x=%0.4f y=%0.4f z=%0.4f\n',nodes_t);
    fprintf(fid, '\n\n**********ELEMENT LIST**********\n');
    fprintf(fid, 'E%0.0f N%0.0f N%0.0f w=%0.4f h=%0.4f\n',elements_t);
end

for i = 1:1:a
    if i == 1
        Ports(i,1) = 1;
        Ports(i,2) = Ports(i,1)+ node_record(i)-1;
    else
        Ports(i,1) = node_record(i-1)+Ports(i-1,1);
        Ports(i,2) = Ports(i,1) + node_record(i)-1;
    end
end
ports_t = Ports';

fprintf(fid, '\n\n*************************************************\n');
fprintf(fid, '**********NETWORK PORT DEFINITIONS**********\n');
fprintf(fid, '*************************************************\n\n');
fprintf(fid, '.external N%0.0f N%0.0f\n',ports_t);

fprintf(fid, '\n\n*************************************************\n');
```

```
    fprintf(fid, '***********FREQUENCY RANGE**********************\n');
    fprintf(fid, '************************************************\n\n');
    fprintf(fid, '.freq fmin=75e6 fmax=75e6 ndec=1\n');
    fprintf(fid, '.end');

    fclose(fid);
end

% atest = [122 2.04456740 3.007686 4.04567476;
%                123 5.06  3.000 4.3443]';
%       fid = fopen('test.txt', 'wt');
%       fprintf(fid, 'N%0.0f x=%0.2f y=%0.2f z=%0.2f\n',atest);
%       fprintf(fid, '\nHow do I write an intervening space\n\n');
%       fprintf(fid, 'Oh, like that');




% r2 = 5;
% r1 = 3;
% xcenter = (r2-r1) + 0.5;
% ycenter = (r2-r1) + 0.5
%
% theta = 0:0.5:2*pi;
%
% X = (r2-r1)*cos(theta) + xcenter;
% Y = (r2-r1)*sin(theta) + ycenter;
%
% plot(X,Y)
% axis([0 5 0 5]);
%
% %[m n] = size(theta)




%%%% Function to read in jacked-up fasthenry matlab files...baaaah!!!! The
%%%% function accepts the directory location and filename that it is to
%%%% process.  It returns the cell matrix A, which contains an impedance
%%%% array of size nxn for each simulation frequency in the FastHenry
%%%% Zc.mat file.  The vector fsw contains the record of frequencies.  Thus
%%%% A{1} will have been computed at frequency fsw(1).


function [A fsw] = fh_parse(dirloc, filename);

clc;
clear A C fsw tot_rows tot_columns freqs;
```

```
%dirloc = './03-Nov-2008/fast_henry_runs/'; %directory where files are located
%filename = 'Zc_geom145_junk.mat'; %name of file to process
fileloc = strcat(dirloc,filename);
fid = fopen(fileloc);


%%%% read file into a cell array.  Each string will be read into one row in
%%%% the first array (C{1}). Strings are separated by whitespace, so there
%%%% will be one row for each group of contiguous characters.
C = textscan(fid,'%s');
file_count = 1; %note, this can't be changed, it is determined by the textscan
%function which creates a cell for each conversion specifier...here there
%is one, %s.

%find the indicies (actually, row numbers) where the word "frequency"
%appears.  These mark the start of a FastHenry output matrix for the given
%frequency
freqs = strmatch('frequency',C{file_count});

%now for each frequency, build the impedance matrix, nominally an L-R
%matrix which is nxn rows where n is the total number of turns (p+s)
%simulated.
for runs = 1:1:size(freqs)
    fsw(runs) = str2double(C{file_count}(freqs(runs)+2)); %get simulation frequency
    tot_rows(runs) = str2double(C{file_count}(freqs(runs)+3)); %find the total #of rows
    tot_columns(runs) = str2double(C{file_count}(freqs(runs)+5)); %find the total #of columns

    if tot_rows(runs) == tot_columns(runs)
        % put the impedance values into the cell array, A
        for row = 1:1:tot_rows(runs)
            for column = 1:1:tot_columns(runs)
                real_temp = str2double(C{file_count}(freqs(runs)+6+
                ...2*(column-1)+2*tot_columns(runs)*(row-1)));
                imag_temp = str2double(C{file_count}(freqs(runs)+7+
                ...2*(column-1)+2*tot_columns(runs)*(row-1)))/(fsw(runs)*2*pi);
                A{runs}(row,column) = real_temp + imag_temp;
            end
        end
        %Lself(runs) = imag(sum(sum(A{runs})));
        %Rself(runs) = real(sum(sum(A{runs})));
    else
        disp('The L-R matrix is not symmetric, this crap is broken!')
        break;
    end
end

fclose(fid);
```

```
%%% This script just displays any geometry file you want to see
clc

geom = 27;

rundate = '03-Nov-2008';
dirloc = strcat('./',rundate,'/fast_henry_runs/');
geom_num_best = num2str(geom);
input_file_best = strcat(dirloc, 'geom',geom_num_best,'_',rundate,'.inp');

[a b] = system(['/opt/local/bin/fasthenry -f simple '
...input_file_best ' -S _geom' geom_num_best]);
[a b] = system(['/opt/local/bin/zbuf zbuffile_geom' geom_num_best]);
[a b] =system(['/opt/local/bin/zbuf zbuffile_geom' geom_num_best ' -m']);


figure (geom+1000);
plotfastH1(['zbuffile_geom' geom_num_best '.mat']);
title(['Geometry for Number = ' geom_num_best]);


system(['rm zbuffile_geom' geom_num_best]);
system(['rm zbuffile_geom' geom_num_best '_shadings']);
system(['rm zbuffile_geom' geom_num_best '.mat']);
system(['rm zbuffile_geom' geom_num_best '.ps']);


%%% This script will compare the FastHenry results with the Duffy-Hurley
%%% calculations.  It will also compute loss using the resistance matrix
%%% from FastHenry and data from a spice file that provides the primary and
%%% secondary currents.  It will then produce a plot showing the size vs.
%%% loss of the candidate transformer designs.
clear L_fh R_fh A_fh F_fh L11_fh L22_fh LM_f R11_fh R22_fh RM_fh;
clc;

scale = 1e9; %convert from henrys to nanohenrys
dirloc = '/Users/sagnea/Documents/Research/PhD/transformer
.../3Nov2008/03-Nov-2008/fast_henry_runs/'
filename = 'Zc_geom';
file_suf = '.mat';


%trans parms: L11, L22, LM, prim_turns, sec_turns, diameter
[tot_geom c] = size(trans_parms);

for geom = 1:1:tot_geom
```

```
file_proc = strcat(filename,num2str(geom),file_suf);

[A_fh F_fh] = fh_parse(dirloc,file_proc);

f_index = 0;
F_desired = 75e6;
for count = 1:1:size(F_fh)
    if F_fh(count) == F_desired
        f_index = count;
    end
end

L_fh{geom} = imag(A_fh{f_index});%inductance matrix
R_fh{geom} = real(A_fh{f_index});%resistance matrix

%%% Condense the full inductance matrix into a 2x2 to represent a 2-winding
%%% transformer.  This is done by dividing the axa matrix into four
%%% sub-blocks.  The sum of all the terms in the upper left block is L11.
%%% The sum of the terms in the lower right block is L22.  The upper left
%%% block terms sum to LM, which is identical for the remaining block.
%%% The upper left block has size pxp where p is the number of turns
%%% comprising the first spiral.  Thus, p,p defines the lower right corner
%%% of the block used to compute L11.  The upper left corner of the L22
%%% block, then, starts at p+1,p+1.  Either of the remaining two blocks may
%%% be used to calculate the mutual inductance terms, this code uses the
%%% upper right block.  It's bottom left corner is p,p+1.

a = trans_parms(geom,4) + trans_parms(geom,5);
p = trans_parms(geom,4);

% extract the inductance parameters by completing the block sums
if a <= 2
    L11_fh(geom) = L_fh{geom}(1,1)*scale;
    L22_fh(geom) = L_fh{geom}(2,2)*scale;
    LM_fh(geom) = L_fh{geom}(1,2)*scale;
else
    L11_fh(geom) = sum(sum(L_fh{geom}(1:p,1:p)))*scale;  %sum the terms for L11
    L22_fh(geom) = sum(sum(L_fh{geom}(p+1:a,p+1:a)))*scale; %sum the terms for L22
    LM_fh(geom) = sum(sum(L_fh{geom}(1:p,p+1:a)))*scale; %sum the terms for LM
end

% extract the resistance parameters by completing the block sums
% I should really create a function, "block_sum"
if a <= 2
    R11_fh(geom) = R_fh{geom}(1,1);
    R22_fh(geom) = R_fh{geom}(2,2);
    RM_fh(geom) = R_fh{geom}(1,2);
else
    R11_fh(geom) = sum(sum(R_fh{geom}(1:p,1:p)));  %sum the terms for L11
```

```
        R22_fh(geom) = sum(sum(R_fh{geom}(p+1:a,p+1:a))); %sum the terms for L22
        RM_fh(geom) = sum(sum(R_fh{geom}(1:p,p+1:a))); %sum the terms for LM
    end
end

%%% Calculate the Loss

%%%% spice current array: time, i_primary, -i_secondary
current_file = 'xformer_currents_8Wreact.txt';

[Pav] = power_calcf(R11_fh,R22_fh,RM_fh,current_file);


%%%% Make some plots

figure(1)
subplot(2,1,1)
title('L11 comparsions, DH vs. FH');
a = plot(trans_parms(:,1));
title('L11 comparsions, DH vs. FH');
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
b = plot(L11_fh)
set(b,'linewidth',2,'color',[0.5 0.5 0.5]);
grid on
xlabel('Geometry Number');
ylabel('Inductance [nH]');
legend('DH','FH');

subplot(2,1,2)
a = plot(trans_parms(:,6));
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on
xlabel('Geometry Number');
ylabel('Transformer Diameter [mm]');


figure(2)
subplot(2,1,1)
title('L22 comparison, DH vs. FH');
a = plot(trans_parms(:,2));
title('L22 comparison, DH vs. FH');
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
b = plot(L22_fh)
set(b,'linewidth',2,'color',[0.5 0.5 0.5]);
grid on
xlabel('Geometry Number');
ylabel('Inductance [nH]');
```

```
legend('DH','FH');

subplot(2,1,2)
a = plot(trans_parms(:,6));
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on
xlabel('Geometry Number');
ylabel('Transformer Diameter [mm]');


figure(3)
subplot(2,1,1)
title('LM comparison, DH vs. FH');
a = plot(trans_parms(:,3));
title('LM comparison, DH vs. FH');
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
b = plot(LM_fh)
set(b,'linewidth',2,'color',[0.5 0.5 0.5]);
grid on
xlabel('Geometry Number');
ylabel('Inductance [nH]');
legend('DH','FH');

subplot(2,1,2)
a = plot(trans_parms(:,6));
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on
xlabel('Geometry Number');
ylabel('Transformer Diameter [mm]');

figure(4)
a = scatter(trans_parms(:,6),Pav);
set(a,'markeredgecolor',[0.597 0.199 0.199]);
grid on
xlabel('Transformer Diameter [mm]');
ylabel('Power Dissipation [W]');


figure(5)
subplot(4,1,1)
title('L-matrix Comparison, Analytical vs. FastHenry');
a = plot(trans_parms(:,1));
title('L-matrix Comparison, Analytical vs. FastHenry');
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
b = plot(L11_fh)
set(b,'linewidth',2,'color',[0.5 0.5 0.5]);
grid on
```

```
%xlabel('Geometry Number');
ylabel('L_{11} [nH]');
legend('Analytical','FH');

subplot(4,1,2)
%title('L_{22} comparison, DH vs. FH');
a = plot(trans_parms(:,2));
%title('L22 comparison, DH vs. FH');
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
b = plot(L22_fh)
set(b,'linewidth',2,'color',[0.5 0.5 0.5]);
grid on
%xlabel('Geometry Number');
ylabel('L_{22} [nH]');
legend('Analytical','FH');

subplot(4,1,3)
%title('L_M comparison, DH vs. FH');
a = plot(trans_parms(:,3));
%title('LM comparison, DH vs. FH');
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
hold on
b = plot(LM_fh);
set(b,'linewidth',2,'color',[0.5 0.5 0.5]);
grid on
%xlabel('Geometry Number');
ylabel('L_M [nH]');
legend('Analytical','FH');

subplot(4,1,4)
a = plot(trans_parms(:,6));
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on
xlabel('Geometry Number');
ylabel('Transformer Diameter [mm]');


function f = mutual(z1,z2,h1,h2,r1,r2,a1,a2,start,stop,tol)

order=0;
mu_0 = 4*pi*1e-7;

M_const = (mu_0*pi)/(h1*h2*log(r2/r1)*log(a2/a1));
f = quad(@mut,start,stop,tol)*M_const;

    function f = mut(k);
        Q = (2./k.^2).*(cosh(k*(h1 + h2)/2) - cosh(k*(h1 - h2)/2));
        S1 = (1./k).*(besselj(order,k*r2)-besselj(order,k*r1));
        S2 = (1./k).*(besselj(order,k*a2)-besselj(order,k*a1));
```

```
        f = Q.*S1.*S2.*exp(-k.*(z2-z1));
    end
end


currents = dlmread('xformer_currents_8Wreact.txt','\t',1,1);
start = 1;
stop = 1070;


time = (currents(start:stop,1)-currents(start,1))*1e9;
iprim = currents(start:stop,2);
isec = currents(start:stop,3);

figure
subplot(2,1,1)
a = plot(time,iprim)
title('Transformer Currents in Target Isolated \Phi_2 Converter')
set(a,'linewidth',2,'color',[0.597 0.199 0.199]);
grid on
ylabel('Primary Current [A]');
xlabel('Time [ns]');
axis([0 40 -5.4 3.2]);
subplot(2,1,2)
b = plot(time,isec)
set(b,'linewidth',2,'color',[0.597 0.199 0.199]);
ylabel('Secondary Current [A]');
xlabel('Time [ns]');
grid on
axis([0 40 -1.3 2.2]);


%function plotfastH(name);
% Load and plot a 3D fasthenry structure produced with "zbuf -m name"
% where name is without the ".mat"

%name = 'zbuffile_geom510.mat';

function [] = plotfastH1(name);

xt = []; yt = []; zt =  [];
xq = []; yq = []; zq = [];

eval(['load ' name]);
fprintf(1, 'loaded %d panels\n', length(xt) + length(xq));
%hold off;
if length(xt) > 0,
  ht = fill3(xt, yt, zt, 'r');
  hold on;
```

```
end
X = max([max(xt) max(yt) max(zt)]);
Y = min([min(xt) min(yt) min(zt)]);

if length(xq) > 0,
  hq = fill3(xq, yq, zq, 'y');
end;

X = max([X max(xq) max(yq) max(zq)]);
Y = min([Y min(xq) min(yq) min(zq)]);

%figure(2)

axis([Y X Y X Y X]);
fprintf(1, 'finished filling polygons\n');

%return;


axis('square');
%return;

if length(xt) > 0
    set(ht, 'FaceColor', 'w')
    set(ht, 'EdgeColor', 'k')
end

set(hq, 'FaceColor', 'w')
set(hq, 'EdgeColor', 'k')
axis('square');
grid on
%axis('off');
return;

%g = get(hq(1), 'LineWidth'); set(hq, 'LineWidth', 2*g);

f = gcf;
set(f, 'Color', [1 1 1]);
set(f, 'InvertHardcopy', 'off');

% %set(f, 'PaperOrientation', 'landscape');
% print -deps panels.ps
%
% return;
%
% fprintf(1, 'printing...\n');
% print -dps -Plouvre
% !lpq -Plouvre
```

```
%%%% test power calculation....

function [Pavg] = power_calcf(R11_fh, R22_fh, RM_fh, current_file)

clear current currents R11 R22 RM i_prim i_sec v_prim v_sec E_tot delta_t E
clc

%current_file = 'xformer_currents_8Wreact.txt';
current = dlmread(current_file,'\t',1,1);
%current = dlmread('xformer_currents_8Wreact.txt','\t',1,1);
currents(:,1:2) = current(:,1:2);
currents(:,3) = -current(:,3);
i_prim = currents(:,2);
i_sec = currents(:,3);
time = currents(:,1);

[r c] = size(currents);

%R11 = 0.09;
%R22 = 0.3;
%RM = 0.009;



[junk geom_tot] = size(R11_fh);

for geom = 1:1:geom_tot

    clear v_prim v_sec E E_tot delta_t
    v_prim = zeros(1,r);
    v_sec = zeros(1,r);

    for count = 1:1:r
        v_prim(count) = R11_fh(geom)*i_prim(count) + RM_fh(geom)*i_sec(count);
        v_sec(count) = RM_fh(geom)*i_prim(count) + R22_fh(geom)*i_sec(count);
    end

    E_tot = 0;
    for count = 1:1:(r-1)
        delta_t = time(count+1)-time(count);
        E(count) = (v_prim(count)*i_prim(count) +
        ...v_sec(count)*i_sec(count))*delta_t;
        E_tot = E_tot + E(count);
    end


    Pavg(geom) = E_tot/(time(r)-time(1));
end
```

```
%%%% test power calculation....

clear current currents R11 R22 RM i_prim i_sec v_prim v_sec E_tot delta_t E
clc

current = dlmread('xformer_currents_8Wreact.txt','\t',1,1);
currents(:,1:2) = current(:,1:2);
currents(:,3) = -current(:,3);

[r c] = size(currents);

R11 = 0.09;
R22 = 0.3;
RM = 0.009;

i_prim = zeros(1,r);
i_sec = zeros(1,r);
v_prim = zeros(1,r);
v_sec = zeros(1,r);

for count = 1:1:r
    i_prim(count) = currents(count,2);
    i_sec(count) = currents(count,3);

    v_prim(count) = R11*i_prim(count) + RM*i_sec(count);
    v_sec(count) = RM*i_prim(count) + R22*i_sec(count);

    time(count) = currents(count,1);
end

E_tot = 0;
for count = 1:1:(r-1)
    delta_t = time(count+1)-time(count);
    E(count) = (v_prim(count)*i_prim(count) +
    ...v_sec(count)*i_sec(count))*delta_t;
    E_tot = E_tot + E(count);
end


Pavg = E_tot/(time(r)-time(1))



%%% This script computes the mutual inductance for the set of individual
```

```
%%% primary and secondary windings selected from the self-inductance only
%%% scripts.  It takes a primary coil and a secondary coil, picks some
%%% z-distance between them (trying the standard distances for each layer)
%%% and computes the mutual inductance.  It then adjusts z until the mutual
%%% coupling is within some predefined tolerance.  The entire cycle is
%%% repeated for the full set of primary and secondary windings

%%%%%%% constants %%%%%%%%%
mu_0 = 4*pi*1E-7;
units = 1e-3; %convert mm to meters
%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%% integration parameters %%%%%%%%%%
start = 1e-1;%1e-24;
stop = 10000;
tol = 1e-18;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%temporary test stuff %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% clear Amatch11;
% clear Amatch22;
% clc;
%
%
% Amatch11 = [0 0.036 2.24 2.5 0 0 0 0;
%             0 0.036 1.64 2.4 0 0.036 0.62 1.38;
%             0 0.036 1.84 2.8 0 0.036 0.52 1.48];
% Amatch22 = [0 0.036 3.04 3.2 0 0.036 2.52 2.68 0 0 0 0;
%             0 0.036 3.84 4.3 0 0.036 2.82 3.28 0 0 0 0;
%             0 0.036 3.64 4.1 0 0.036 2.32 2.78 0 0.036 1.0 1.46];
% turns11 = [1 2 2];
% turns22 = [2 2 3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% desired magnetizing inductance and tolerance %%%%%%%%%
LM_d = 11.8176e-9;
tol_LM = 0.5e-9;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[r11 c11] = size(Amatch11);
[r22 c22] = size(Amatch22);

xformers = zeros(r11*r22,c11+c22);
primary_turns = zeros(1,r11*r22);
secondary_turns = zeros(1,r11*r22);
coilID = zeros(r11*r22,2);
fail = zeros(1,r11*r22);
z_choice = zeros(1,r11*r22);
counter = 1;
```

```
for primary_coil = 1:1:r11
    for secondary_coil = 1:1:r22
        xformers(counter,1:c11) = Amatch11(primary_coil,:)*units;
        xformers(counter,(c11+1):(c11+c22)) = Amatch22(secondary_coil,:)*units;
        xformers(counter,(c11+c22+1))= primary_coil;
        xformers(counter,(c11+c22+2)) = secondary_coil;
        primary_turns(counter) = turns11(primary_coil);
        secondary_turns(counter) = turns22(secondary_coil);
        coilID(counter,1) = primary_coil;
        coilID(counter,2) = secondary_coil;
        counter = counter + 1;
    end
end

z_ref = [0.2032 0.9906 1.5748]*units;%note, it is critical that the reference
% are entered from smallest to largest, otherwise this code will BREAK!!!!

[junk ref_max] = size(z_ref);
[rx cx] = size(xformers);
sec_col = c11 + 1; %column in xformers that begins the secondary coil descriptions
LM = zeros(rx,ref_max+1);
for geom = 1:1:rx
    geometry = geom
    p = primary_turns(geom); % primary turns
    s = secondary_turns(geom); % secondary turns
    a = primary_turns(geom) + secondary_turns(geom); % total turns
    M{geom} = zeros(p,a-(p+1)); %initialize cell matrix
    z_match = 0; %z_flag
    ref_calc = 0;
    closest = 0;
    in_loop = 0;

    while z_match == 0;
        %in_loop = in_loop + 1
        if ref_calc == 0
            %calculate LM for each of the possible standard PCB separations
            %(in this case layer1-2, 2-3, and 1-4, other sets of z_ref,
            %will do this for other board configurations).
            for reference = 1:1:ref_max
                LMtemp = 0;
                z_trial = z_ref(reference);
                for prim_coil = 1:1:p
                    for sec_coil = 1:1:s

                        %get geometric parameters for the primary coil
                        z1 = 0; % z-location of primary always 0
                        h1 = xformers(geom,4*(prim_coil-1)+2);
                        % thickness of coil 1
                        r1 = xformers(geom,4*(prim_coil-1)+3);
```

```
                % inner radius of coil 1
                r2 = xformers(geom,4*(prim_coil-1)+4);
                % outer radius of coil 1

                %get geometric parameters for the secondary coil
                z2 = z_trial;% z-location of coil 2
                h2 = xformers(geom,4*(sec_coil-1)+1+sec_col);
                % thickness of coil 2
                a1 = xformers(geom,4*(sec_coil-1)+2+sec_col);
                % inner radius of coil 2
                a2 = xformers(geom,4*(sec_coil-1)+3+sec_col);
                % outer radius of coil 2

                MUTtemp = mutual3(z1,z2,h1,h2,r1,r2,a1,a2,start,stop,tol);
                M{geom,reference}(prim_coil,sec_coil) = MUTtemp;
                LMtemp = MUTtemp + LMtemp;
            end
        end
        LM(geom,reference) = LMtemp;
    end

    %check to see if any of the reference values of z give an LM
    %within tolerance.  If so, great, pop out of the loop and start
    %on the next geometry
    for reference = 1:1:ref_max
        if LM(geom,reference) > LM_d - tol_LM && LM(geom,reference)
        ...< LM_d + tol_LM
            LM(geom,ref_max+1) = LM(geom,reference);
            z_choice(geom) = z_ref(reference);
            z_match = 1;
        end
    end
    ref_calc = 1;
else
    %find the reference height it's closest to...
    if closest == 0;
        Ldiff = 10000;
        for reference = 1:1:ref_max
            if abs(LM(geom,reference) - LM_d) < Ldiff
                Ldiff = abs(LM(geom,reference) - LM_d);
                closest = reference;
            end
        end
        %determine which way to adjust the height
        if closest == 1 && LM(geom,1) - LM_d < 0
            fail(geom) = 1;
            % z-sep would have to be closer than allowed
            LM(geom,ref_max+1) = -10000;
            z_choice(geom) = -10000;
```

```
        z_match = 1;

    elseif closest == ref_max && LM(geom,ref_max) - LM_d > 0
        fail(geom) = 2;
        % z-sep would need to be further away than allowed
        LM(geom,ref_max+1) = 10000;
        z_choice(geom) = 10000;
        z_match = 1;

    elseif LM(geom,closest) - LM_d > 0
        z_trial = (z_ref(closest) + z_ref(closest + 1))/2;
        %move a little further away
        z_old = z_ref(closest);
    else
        z_trial = (z_ref(closest) + z_ref(closest - 1))/2;
        %move a little closer together
        z_old = z_ref(closest);
    end
end

if fail(geom) == 0
    LMtemp = 0;
    for prim_coil = 1:1:p
        for sec_coil = 1:1:s

            %get geometric parameters for the primary coil
            z1 = 0; % z-location of primary always 0
            h1 = xformers(geom,4*(prim_coil-1)+2);
            % thickness of coil 1
            r1 = xformers(geom,4*(prim_coil-1)+3);
            % inner radius of coil 1
            r2 = xformers(geom,4*(prim_coil-1)+4);
            % outer radius of coil 1

            %get geometric parameters for the secondary coil
            z2 = z_trial;% z-location of coil 2
            h2 = xformers(geom,4*(sec_coil-1)+1+sec_col);
            % thickness of coil 2
            a1 = xformers(geom,4*(sec_coil-1)+2+sec_col);
            % inner radius of coil 2
            a2 = xformers(geom,4*(sec_coil-1)+3+sec_col);
            % outer radius of coil 2

            M{geom,ref_max+1}(prim_coil,sec_coil) =
            ...mutual3(z1,z2,h1,h2,r1,r2,a1,a2,start,stop,tol);
            LMtemp = M{geom,ref_max+1}(prim_coil,sec_coil) + LMtemp;
        end
    end
```

```
                %now figure out what to do next
                if LMtemp > LM_d - tol_LM && LMtemp < LM_d + tol_LM
                %met requirements, start on next geometry
                    LM(geom,ref_max+1) = LMtemp;
                    z_choice(geom) = z_trial;
                    z_match = 1;

                elseif LMtemp - LM_d > 0 %move a little further away
                    z_diff = abs(z_trial - z_old)/2;
                    z_old = z_trial;
                    z_trial = z_trial + z_diff;

                else % move a little closer together
                    z_diff = abs(z_trial - z_old)/2;
                    z_old = z_trial;
                    z_trial = z_trial - z_diff;
                end
            end
        end
    end
end

%filename = strcat('trans_mut_v1p0_',date);
%save(filename,'LM','z_choice','fail','M','xformers');

[junk1,junk2]=system(['mkdir ' date]); %create a directory with named by the date
save_dir = date; %define this as the save directory

version = '_v1p0_';
file_prefix = 'Trans-mutual';
high_flag = 0;
file_num = 1;

%check to see if a file already exists, if not create it, if so, create the
%next highest file number
while high_flag == 0
    file_exists = strcat('./',save_dir,'/',file_prefix ,
    ...version, num2str(file_num),'.mat');
    [a b] = system(['ls ' file_exists]);
    if a == 0
         file_num = file_num + 1;
    else
       filename = strcat(file_prefix, version, num2str(file_num));
       fileloc = strcat('./',date,'/',filename);
       save(fileloc,'LM','z_choice','fail','M','xformers','coilID');
        high_flag = 1;
        file_num = file_num + 1;
    end
end
```

```
%%% This script computes the mutual inductance for the set of individual
%%% primary and secondary windings selected from the self-inductance only
%%% scripts.  It takes a primary coil and a secondary coil, picks some
%%% z-distance between them (trying the standard distances for each layer)
%%% and computes the mutual inductance.  It then adjusts z until the mutual
%%% coupling is within some predefined tolerance.  The entire cycle is
%%% repeated for the full set of primary and secondary windings

%%%%%%% constants %%%%%%%%%
mu_0 = 4*pi*1E-7;
units = 1e-3; %convert mm to meters
%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% integration parameters %%%%%%%%%%
start = 1e-1;%1e-24;
stop = 10000;
tol = 1e-18;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%temporary test stuff %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% clear Amatch11;
% clear Amatch22;
% clc;
%
%
% Amatch11 = [0 0.036 2.24 2.5 0 0 0 0;
%             0 0.036 1.64 2.4 0 0.036 0.62 1.38;
%             0 0.036 1.84 2.8 0 0.036 0.52 1.48];
% Amatch22 = [0 0.036 3.04 3.2 0 0.036 2.52 2.68 0 0 0 0;
%             0 0.036 3.84 4.3 0 0.036 2.82 3.28 0 0 0 0;
%             0 0.036 3.64 4.1 0 0.036 2.32 2.78 0 0.036 1.0 1.46];
% turns11 = [1 2 2];
% turns22 = [2 2 3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% desired magnetizing inductance and tolerance %%%%%%%%%%
LM_d = 11.8176e-9;
tol_LM = 0.5e-9;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[r11 c11] = size(Amatch11);
[r22 c22] = size(Amatch22);

xformers = zeros(r11*r22,c11+c22);
primary_turns = zeros(1,r11*r22);
secondary_turns = zeros(1,r11*r22);
fail = zeros(1,r11*r22);
```

```
z_choice = zeros(1,r11*r22);
counter = 1;
for primary_coil = 1:1:r11
    for secondary_coil = 1:1:r22
        xformers(counter,1:c11) = Amatch11(primary_coil,:)*units;
        xformers(counter,(c11+1):(c11+c22)) = Amatch22(secondary_coil,:)*units;
        xformers(counter,(c11+c22+1))= primary_coil;
        xformers(counter,(c11+c22+2)) = secondary_coil;
        primary_turns(counter) = turns11(primary_coil);
        secondary_turns(counter) = turns22(secondary_coil);
        counter = counter + 1;
    end
end

z_ref = [0.2032 0.9906 1.5748]*units;%note, it is critical that the reference
% are entered from smallest to largest, otherwise this code will BREAK!!!!

[junk ref_max] = size(z_ref);
[rx cx] = size(xformers);
sec_col = c11 + 1; %column in xformers that begins the secondary coil descriptions
LM = zeros(rx,ref_max+1);
for geom = 1:1:rx
    geometry = geom
    p = primary_turns(geom); % primary turns
    s = secondary_turns(geom); % secondary turns
    a = primary_turns(geom) + secondary_turns(geom); % total turns
    M{geom} = zeros(p,a-(p+1)); %initialize cell matrix
    z_match = 0; %z_flag
    ref_calc = 0;
    closest = 0;
    in_loop = 0;

    while z_match == 0;
        %in_loop = in_loop + 1
        if ref_calc == 0
            %calculate LM for each of the possible standard PCB separations
            %(in this case layer1-2, 2-3, and 1-4, other sets of z_ref,
            %will do this for other board configurations).
            for reference = 1:1:ref_max
                LMtemp = 0;
                z_trial = z_ref(reference);
                for prim_coil = 1:1:p
                    for sec_coil = 1:1:s

                        %get geometric parameters for the primary coil
                        z1 = 0; % z-location of primary always 0
                        h1 = xformers(geom,4*(prim_coil-1)+2);
                        % thickness of coil 1
                        r1 = xformers(geom,4*(prim_coil-1)+3);
```

```
                % inner radius of coil 1
                r2 = xformers(geom,4*(prim_coil-1)+4);
                % outer radius of coil 1

                %get geometric parameters for the secondary coil
                z2 = z_trial;% z-location of coil 2
                h2 = xformers(geom,4*(sec_coil-1)+1+sec_col);
                % thickness of coil 2
                a1 = xformers(geom,4*(sec_coil-1)+2+sec_col);
                % inner radius of coil 2
                a2 = xformers(geom,4*(sec_coil-1)+3+sec_col);
                % outer radius of coil 2

                MUTtemp = mutual3(z1,z2,h1,h2,r1,r2,a1,a2,start,stop,tol);
                M{geom,reference}(prim_coil,sec_coil) = MUTtemp;
                LMtemp = MUTtemp + LMtemp;
            end
        end
        LM(geom,reference) = LMtemp;
    end

    %check to see if any of the reference values of z give an LM
    %within tolerance.  If so, great, pop out of the loop and start
    %on the next geometry
    for reference = 1:1:ref_max
        if LM(geom,reference) > LM_d - tol_LM && LM(geom,reference)
        ...< LM_d + tol_LM
            LM(geom,ref_max+1) = LM(geom,reference);
            z_choice(geom) = z_ref(reference);
            z_match = 1;
        end
    end
    ref_calc = 1;
else
    %find the reference height it's closest to...
    if closest == 0;
        Ldiff = 10000;
        for reference = 1:1:ref_max
            if abs(LM(geom,reference) - LM_d) < Ldiff
                Ldiff = abs(LM(geom,reference) - LM_d);
                closest = reference;
            end
        end
        %determine which way to adjust the height
        if closest == 1 && LM(geom,1) - LM_d < 0
            fail(geom) = 1; % z-sep would have to be closer than allowed
            LM(geom,ref_max+1) = -10000;
            z_choice(geom) = -10000;
            z_match = 1;
```

```
    elseif closest == ref_max && LM(geom,ref_max) - LM_d > 0
        fail(geom) = 2;
        % z-sep would need to be further away than allowed
        LM(geom,ref_max+1) = 10000;
        z_choice(geom) = 10000;
        z_match = 1;

    elseif LM(geom,closest) - LM_d > 0
        z_trial = (z_ref(closest) + z_ref(closest + 1))/2;
        %move a little further away
        z_old = z_ref(closest);
    else
        z_trial = (z_ref(closest) + z_ref(closest - 1))/2;
        %move a little closer together
        z_old = z_ref(closest);
    end
end

if fail(geom) == 0
    LMtemp = 0;
    for prim_coil = 1:1:p
        for sec_coil = 1:1:s

            %get geometric parameters for the primary coil
            z1 = 0; % z-location of primary always 0
            h1 = xformers(geom,4*(prim_coil-1)+2);
            % thickness of coil 1
            r1 = xformers(geom,4*(prim_coil-1)+3);
            % inner radius of coil 1
            r2 = xformers(geom,4*(prim_coil-1)+4);
            % outer radius of coil 1

            %get geometric parameters for the secondary coil
            z2 = z_trial;% z-location of coil 2
            h2 = xformers(geom,4*(sec_coil-1)+1+sec_col);
            % thickness of coil 2
            a1 = xformers(geom,4*(sec_coil-1)+2+sec_col);
            % inner radius of coil 2
            a2 = xformers(geom,4*(sec_coil-1)+3+sec_col);
            % outer radius of coil 2

            M{geom,ref_max+1}(prim_coil,sec_coil) =
            ...mutual3(z1,z2,h1,h2,r1,r2,a1,a2,start,stop,tol);
            LMtemp = M{geom,ref_max+1}(prim_coil,sec_coil) + LMtemp;
        end
    end

    %now figure out what to do next
```

```
                if LMtemp > LM_d - tol_LM && LMtemp < LM_d + tol_LM
                %met requirements, start on next geometry
                    LM(geom,ref_max+1) = LMtemp;
                    z_choice(geom) = z_trial;
                    z_match = 1;

                elseif LMtemp - LM_d > 0 %move a little further away
                    z_diff = abs(z_trial - z_old)/2;
                    z_old = z_trial;
                    z_trial = z_trial + z_diff;

                else % move a little closer together
                    z_diff = abs(z_trial - z_old)/2;
                    z_old = z_trial;
                    z_trial = z_trial - z_diff;
                end
            end
        end
    end
end

%filename = strcat('trans_mut_v1p0_',date);
%save(filename,'LM','z_choice','fail','M','xformers');

[junk1,junk2]=system(['mkdir ' date]); %create a directory with named by the date
save_dir = date; %define this as the save directory

version = '_v1p0_';
file_prefix = 'Trans-mutual';
high_flag = 0;
file_num = 1;

%check to see if a file already exists, if not create it, if so, create the
%next highest file number
while high_flag == 0
    file_exists = strcat('./',save_dir,'/',file_prefix ,
    ...version, num2str(file_num),'.mat');
    [a b] = system(['ls ' file_exists]);
    if a == 0
         file_num = file_num + 1;
    else
       filename = strcat(file_prefix, version, num2str(file_num));
       fileloc = strcat('./',date,'/',filename);
       save(fileloc,'LM','z_choice','fail','M','xformers');
        high_flag = 1;
        file_num = file_num + 1;
    end
end
```

```
function f = self(z,h,r1,r2)

mu_0 = 4*3.14159E-7;
%center = ((r2+r1)/2);
center = sqrt(r1*r2);
gmd = 0.2235*((r2-r1)+h);
f1 = sqrt((4*center^2)/(gmd^2+(2*center)^2));
[K,E]=ellipke(f1);
f = mu_0*center*(2/f1)*((1-f1^2/2)*K-E);


clear all;
clc;


%%%%% 28 Oct 08:  This code takes the trials and computes the inductance
%%%%% parameters (L11, L22, and LM)



%[A_mm] = trials_buildervf4f(); % gives the A-matrix in mm
[A_mm] = trials_buildervf_self_only();
A = A_mm*1e-3; %scales A-matrix in mm to meters

%A = [0 0.0360 4.24 4.4 0 0 0 0; 0 0.0360 3.9 4.2 0 0.0360 4.24 4.4]*1e-3;

A_matrix_created = 1



%%%%%%% constants %%%%%%%%%%
mu_0 = 4*pi*1E-7;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% integration parameters %%%%%%%%%%
start = 1e-1;%1e-24;
stop = 10000;
tol = 1e-18;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



% A = [0 0.0001 0.03 0.05 0 0.0001 0.06 0.09 0.001 0.0001 0.01 0.012 0.001 0.0001 0.017 0.025;
%       0 0.00157 0.003 0.005 0 0.00157 0.008 0.009 0.001 0.00157 0.003 0.005 0.001 0.00157 0.008 0.009;
%       0 0.00157 0.003 0.005 0.001 0.00157 0.003 0.005 0 0 0 0 0 0 0 0;
%       0 0.00157 0.003 0.005 0 0.000157 0.008 0.009 0.001 0.00157 0.003 0.005 0 0 0 0;
%       0 0.0001 0.004 0.006 0 0.0001 0.009 0.010 0.001 0.0001 0.004 0.006 0.001 0.0001 0.009 0.010];

%A = [0 0.0001 0.004 0.006 0.001 0.0001 0.004 0.006 ];
```

```
%A = [0 0.0001 0.004 0.006 0.001 0.0001 0.004 0.006;
%      0 0.0001 0.004 0.006 0 0 0 0 ];

%A = [0 0.0001 0.002 0.0035 0 0.0001 0.004 0.006 0 0.0001 0.007 0.009];
%A = [0 0.0360 2.0400 2.4000 0 0.0360 0.7200 1.0800]*1e-3; %test case for primary only...


[m,n] = size(A);

% find the location of the last ring that located in the same plane as the
% first winding.  Note: to define two windings, all the rings in the first
% winding must be defined before any of the rings in the second winding are
% defined.  Otherwise this crap will blow up....

%this code will provide the column of the last non-zero term of each row
%(ie, each geometry)
tot_terms = zeros(1,m);
for i = 1:1:m
    for j = 1:1:n
        if A(i,j)~=0
            tot_terms(i) = j; % find the total terms to identify the # of rings
        end
    end
end

%this code finds all the turns in the plane of the first turn provided in
%the A-matrix.  These turns all form the primary winding, so it effectively
%determines the number of turns in the primary
prim_turns = zeros(1,m);
for i = 1:1:m
    for j = 1:1:n
        if (A(i,j)==A(i,1) && (mod(j-1,4)==0 || j==1) && j < tot_terms(i))
            prim_turns(i)= 1+(j-1)/4; % number of turns (rings) in primary winding
        end
    end
end


for geom = 1:1:m

    a = tot_terms(geom)/4;  %a is the total number of turns (rings)
    p = prim_turns(geom); %p is the number of turns in the primary

    if a > 1 %If this is actually a transformer, or a coil that has more
        %%%% than a single turn, then do the following code....

        %%%% Build the inductance matrix
        %%%% This will build an inductance matrix for two spiral windings of an
        %%%% arbitrary number of turns.  The winding inductance and coupling is
```

```
%%%% calculated from the formulas in the Duffy-Hurley paper for the mutual
%%%% inductance between two planar rings.  To represent multiple turns, on
%%%% a single planar spiral winding, multiple rings are assumed.  This
%%%% code sizes the inductance matrix as appropriate and then populates
%%%% each term by calculating the mutual inductance between all the ring
%%%% pairs and the self inductance of each ring.

L = zeros(a);

for i = 1:1:a
    for j = i:1:a

        % get the geometric parameters
        %if mod(i-1,4)==0 || i == 1
            z1 = A(geom,4*(i-1)+1); % z-location of coil 1
            h1 = A(geom,4*(i-1)+2); % thickness of coil 1
            r1 = A(geom,4*(i-1)+3); % inner radius of coil 1
            r2 = A(geom,4*(i-1)+4); % outer radius of coil 1
        %end
        %if mod(j-1,4)==0 || j == 1
            z2 = A(geom,4*(j-1)+1); % z-location of coil 2
            h2 = A(geom,4*(j-1)+2); % thickness of coil 2
            a1 = A(geom,4*(j-1)+3); % inner radius of coil 2
            a2 = A(geom,4*(j-1)+4); % outer radius of coil 2
        %end

        %add the self inductance terms on the main diagonal
        if i == j
            L(i,j) = self(z1,h1,r1,r2);

        %add the mutual inductance terms
        else
            L(i,j) = mutual3(z1,z2,h1,h2,r1,r2,a1,a2,start,stop,tol);
            L(j,i) = L(i,j); %complementary mutual terms are equal
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Condense the full inductance matrix into a 2x2 to represent a 2-winding
%%% transformer.  This is done by dividing the axa matrix into four
%%% sub-blocks.  The sum of all the terms in the upper left block is L11.
%%% The sum of the terms in the lower right block is L22.  The upper left
%%% block terms sum to LM, which is identical for the remaining block.
%%% The upper left block has size pxp where p is the number of turns
%%% comprising the first spiral.  Thus, p,p defines the lower right corner
%%% of the block used to compute L11.  The upper left corner of the L22
%%% block, then, starts at p+1,p+1.  Either of the remaining two blocks may
```

```
        %%% be used to calculate the mutual inductance terms, this code uses the
        %%% upper right block.  It's bottom left corner is p,p+1.

        if p == a
            L11(geom) = sum(sum(L(1:p,1:p)));
            L22(geom) = 0;
            LM(geom) = 0;
        elseif a <= 2
            L11(geom) = L(1,1);
            L22(geom) = L(2,2);
            LM(geom) = L(1,2);
        else
            L11(geom) = sum(sum(L(1:p,1:p)));  %sum the terms for L11
            L22(geom) = sum(sum(L(p+1:a,p+1:a))); %sum the terms for L22
            LM(geom) = sum(sum(L(1:p,p+1:a))); %sum the terms for LM
        end
        Lmem{geom} = L;

    else % if it's just a single ring, then do the following:
        z1 = A(geom,1); % z-location of coil 1
        h1 = A(geom,2); % thickness of coil 1
        r1 = A(geom,3); % inner radius of coil 1
        r2 = A(geom,4); % outer radius of coil 1
        L11(geom) = self(z1,h1,r1,r2);
        L22(geom) = 0;
        LM(geom) = 0;
        %Lmem{geom}(:,:) = L(:,:);

    end
geom_number = geom
end

%filename = strcat('coil_self_inductance_v1p0_',date);
%save(filename,'A_mm','L11','prim_turns','tot_terms','Lmem');
%save(filename,'A_mm','L11','prim_turns','tot_terms');




[junk1,junk2]=system(['mkdir ' date]); %create a directory with named by the date
save_dir = date; %define this as the save directory

version = '_v1p0_';
file_prefix = 'Coil_self_inductance';
high_flag = 0;
file_num = 1;

%check to see if a file already exists, if not create it, if so, create the
%next highest file number
```

```
while high_flag == 0
    file_exists = strcat('./',save_dir,'/',file_prefix ,
    ...version, num2str(file_num),'.mat');
    [a b] = system(['ls ' file_exists]);
    if a == 0
         file_num = file_num + 1;
    else
       filename = strcat(file_prefix, version, num2str(file_num));
       fileloc = strcat('./',date,'/',filename);
       save(fileloc,'A_mm','L11','prim_turns','tot_terms');
        high_flag = 1;
        file_num = file_num + 1;
    end
end
```

A.3 Spice Code

```
***************************
*** LIST OF LIBRARIES   ***
***************************

.LIB CLASSE.LIB"
.LIB RECT21.LIB"
.LIB MITMV1_LOSS.LIB
.LIB TRANSFORMER_LOSS_FASTHENRY.LIB

*********************************************************
**** OPTIONS FOR BETTER CONVERGENCE       ***
*********************************************************
.OPTIONS ABSTOL=1nA
+          GMIN=1p
+          ITL1=6000
+          ITL2=4000
+          ITL4=5000
+          RELTOL=0.001
+          VNTOL=0.001mV
.OPTION STEPGMIN


*****************************************
*** OPTIONS TO KEEP A SMALL OUTPUT FILE ***
*****************************************
.OPTIONS
+ NOPAGE
+ NOBIAS
```

```
+ NOECHO
+ NOMOD
+ NUMDGT=8
.WIDTH OUT=132 ;TO PRINT MORE COLUMNS

*****************************************
*** SPECIAL PARAMETERS AND CONSTANTS    ***
*****************************************
.PARAM
+ PI=3.14159    ;GUESS WHAT


*************************
* CIRCUIT PARAMETERS     *
*************************

*---DESIGN PARAMETERS

.PARAM:

+FS=75MEG         ;SWITCHING FREQUENCY
+VIN=12  ;INPUT VOLTAGE
+VOUT=12          ;OUTPUT VOLTAGE
+DUTY=.33;.4;.33   ;DUTY CYCLE
+QC=5k          ;Q CAPACITORS
+QI=80        ;Q INDUCTORS
+TRMULT = 200   ;GATE DRIVE RISE-TIME MULTIPLIER (1/(TRMULT*FS))
+QLREC = 109
+QLF = 80
+QL2F = 80
+QLDIV = 109

*---L2F AND C2F PARAMETRIZED BY ZO AND F2S
.PARAM:


*+ F2S = 100MEG
*+ ZO = 10;13.83128
*+ L2F = {ZO/(2*PI*F2S)}
*+ C2F = {1/(2*PI*F2S*ZO)}

+ L2F = 15.0106n
+ C2F = 75p

*---COUT, CEXTRA, LF, LREC, CREC

+ KAREA = 1;2
+ CEXTRA = 122.2p;165p
+ COUTNOM = 97p
```

```
+ LF = 10.58n;5.575n
*+ LF = 11.8176n
*+ L11 = 6.2426n
+ L11 = 1n
+ LREC = 21.97n;8n
+ LDIV = 0
*+ LREC = 25.188n
*+ LDIV = 9.945n
+ CREC = 15.1406p;45.2024p
+ RGATE = 300M
+ LRECP = 1p
+ LCRECP = 1p
*+ N2 = 2
+ N2 = 1;1
+ CBLOCK = 100n
+ LDC = 1u
**************************
***CIRCUIT DESCRIPTION ***
**************************

*---DC source
VIN IN 0 {VIN}
Rbig IN 0 10g

*---LUMPED MR NETWORK

*XLF IN DRAIN LCHOKE       ;MULTIRESONANT ELEMENT
*+ PARAMS:
*+ L={LF}
*+ QL = {QLF}
*+ FQ={FS}
*+ IC=0
*+ RDC=10m


XL2F  DRAIN V2FX LQS       ;MULTIRESONANT ELEMENT
+ PARAMS:
+ L={L2F}
+ QL={QL2F}
+ FQ={2*FS}
+ IC=0


VD2F  V2FX V2F 0         ;DUMMY TO MEASURE CURRENT L2F

XC2F   V2F 0 CQS          ;MULTIRESONANT ELEMENT
+ PARAMS:
+ C={C2F}
+ QC={QC}
```

```
+ FQ={FS}
+ IC=0


*---NON LINEAR MOSFET MODEL
VDMOS DRAIN DRAINS 0

XSWITCH GATEX DRAINS SOURCE ICOND IDISP MGND MOSFETNLC
+ PARAMS:
*+ LDRAIN = 400p;400p
*+ LSOURCE = 100p;200p
*+ LGATE = 400p
+ KRES = 1.5;1.5
+ RCOUT = {0.627/Karea}
+ CJO = {215.45p*Karea}
+ RDSON = {0.113/Karea}



RCOND ICOND MGND 1MEG
RDISP IDISP MGND 1MEG
RMGND MGND 0 1MEG



VDMOS1 SOURCE 0 0
VDGATE GATE GATEX 0

XCEXTRA DRAIN SOURCEX CQS
+ PARAMS:
+ C = {CEXTRA - (KAREA-1)*COUTNOM}
*+ C = {CEXTRA*(2-Karea)}
*+ C={CEXTRA}
+ QC={QC}
+ FQ={FS}
+ IC=0

XLCEXTRAP SOURCEX SOURCE LQS
+ PARAMS:
+ L=1f;1n;400p;240p
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0

ibode 0 drain ac 1


********** RECTIFIER WITH TWO PARALLEL DIODES ********************
VDPRIM DRAIN DRAINP 0
```

```
XL11 DRAINP DRAIN11 LQS
+ PARAMS:
+ L={L11};1n;400p;240p
+ QL=100;{QI}
+ FQ={FS}
+ IC=0

XTRANSFORMER DRAIN11 GNDISO DRAINZ 0 IACP IDCP IACS IDCS MGND XFORMER
+ PARAMS:
+ N2={N2}
+ RDCP = 1u
+ RDCS = 1u
+ LM = {LF}
+ LMCALC = {LF}
*+ QP = 100
+ LL2 = {LREC+LDIV}
*+ QS = 100
+ FQ = {FS}
+ IC =0
+ R11 = 42.9M;59.2M;30.2M
+ R22 = 260M;127.7M;304.1M
+ RM = -0.1022M;3M;10.3M

RexACP IACP MGND 1MEG
RexDCP IDCP MGND 1MEG
RexACS IACS MGND 1MEG
RexDCS IDCS MGND 1MEG



VSHORT GNDISO VSTAB 0
RSTAB2 VSTAB IN 1u


VDREC DRAINZ DRAINZ1 0
XLDIV DRAINZ1 DRAINX LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=14.373n;16.633n;{LDIV}
+ QL = 100;{QLDIV} ;109
+ FQ={FS}
+ IC=0


XLREC DRAINX VREC LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LREC}
+ QL = 10000;{QLREC} ;109
+ FQ={FS}
```

```
+ IC=0

XCREC VRECC 0 CQS ; RECTIFIER CAPACITOR
+ PARAMS:
+ C={CREC}
+ QC={QC}
+ FQ={FS}
+ IC=0

VVREC VREC VRECD 0 ;current dummy
XLCRECP VRECD VRECC LQS    ; PARASITIC INDUCTOR FOR SHUNT CAPACITOR
+ PARAMS:
+ L={LCRECP}
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0


VDIODE VREC VREC1 0

DREC1 VREC1 OUT DSS16
*DREC2 VREC1 OUT DSS16

*************** MODIFIED BREAKDOWN VOLTAGE*********************************
.MODEL Dss16 d
+IS=0.000119367 RS=0.0658135 N=1.73244 EG=0.64331
+XTI=0.5 BV=600 IBV=2e-08 CJO=2.05074e-10
+VJ=0.4 M=0.421584 FC=0.5 TT=0
+KF=0 AF=1

VLOAD OUT 0 {VOUT}

************************ GATE DRIVER *****************************
.PARAM
+ TR = 1/(100*FS)
+ PWIDTH = {DUTY/FS - 2*TR}

VDRIVE GATE 0 PULSE (0 5 0 {TR} {TR} {PWIDTH} {1/FS})



*---PARAMETER EXTRACTION
VPLF PLF 0 {LF}
VPL2F    PL2F   0  {L2F}
VPC2F    PC2F   0  {C2F}
VPLREC    PLREC  0   {LREC}
VPCREC    PCREC 0    {CREC}
```

```
*************************
*** SIMULATION CONTROL ***
*************************

.TRAN 200p 5U .1U 200P UIC
.PROBE
.END



*************************
*** LIST OF LIBRARIES   ***
*************************

.LIB CLASSE.LIB
.LIB RECT21.LIB
.LIB MITMV1_LOSS.LIB
.LIB TRANSFORMER_LOSS3.LIB

*********************************************************
**** OPTIONS FOR BETTER CONVERGENCE       ***
*********************************************************
.OPTIONS ABSTOL=1nA
+          GMIN=1p
+          ITL1=6000
+          ITL2=4000
+          ITL4=5000
+          RELTOL=0.001
+          VNTOL=0.001mV
.OPTION STEPGMIN


*****************************************
*** OPTIONS TO KEEP A SMALL OUTPUT FILE ***
*****************************************
.OPTIONS
+ NOPAGE
+ NOBIAS
+ NOECHO
+ NOMOD
+ NUMDGT=8
.WIDTH OUT=132 ;TO PRINT MORE COLUMNS

*****************************************
*** SPECIAL PARAMETERS AND CONSTANTS    ***
*****************************************
.PARAM
+ PI=3.14159    ;GUESS WHAT
```

```
*************************
* CIRCUIT PARAMETERS    *
*************************

*---DESIGN PARAMETERS

.PARAM:

+FS=75MEG         ;SWITCHING FREQUENCY
+VIN=12  ;INPUT VOLTAGE
+VOUT=12          ;OUTPUT VOLTAGE
+DUTY=.33;.4;.33   ;DUTY CYCLE
+QC=5k          ;Q CAPACITORS
+QI=80        ;Q INDUCTORS
+TRMULT = 200   ;GATE DRIVE RISE-TIME MULTIPLIER (1/(TRMULT*FS))
+QLREC = 70
+QLF = 80
+QL2F = 80
+QLDIV = 109

*---L2F AND C2F PARAMETRIZED BY Z0 AND F2S
.PARAM:


*+ F2S = 100MEG
*+ Z0 = 10;13.83128
*+ L2F = {Z0/(2*PI*F2S)}
*+ C2F = {1/(2*PI*F2S*Z0)}

+ L2F = 15.0106n
+ C2F = 75p

*---COUT, CEXTRA, LF, LREC, CREC

+ KAREA = 1
+ CEXTRA = 122.2p
+ COUTNOM = 97p
+ LF =11.8176n;6.5n;7.8n ;11.8176n
+ LREC = 25.188n
+ LDIV = 9.945n
+ CREC = 15.1406p
+ RGATE = 300M
+ LRECP = 1p
+ LCRECP = 1p
+ N2 = 1
+ CBLOCK = 100n
+ LDC = 1u
```

```
+ LL1 = 1p;4n;3n;1p
*************************
***CIRCUIT DESCRIPTION ***
*************************

*---DC source
VIN IN 0 {VIN}
Rbig IN 0 10g

*---LUMPED MR NETWORK

XL1 GNDISO TRANIN LQS        ;PRIMARY SIDE LEAKAGE INDUCTANCE...
+ PARAMS:
+ L= {LL1}
+ QL = {QLF}
+ FQ={FS}
+ IC=0
*+ RDC=10m


XL2F  DRAIN V2FX LQS        ;MULTIRESONANT ELEMENT
+ PARAMS:
+ L={L2F}
+ QL={QL2F}
+ FQ={2*FS}
+ IC=0


VD2F  V2FX V2F 0         ;DUMMY TO MEASURE CURRENT L2F

XC2F   V2F 0 CQS           ;MULTIRESONANT ELEMENT
+ PARAMS:
+ C={C2F}
+ QC={QC}
+ FQ={FS}
+ IC=0


*---NON LINEAR MOSFET MODEL
VDMOS DRAIN DRAINS 0

XSWITCH GATEX DRAINS SOURCE ICOND IDISP MGND MOSFETNLC
+ PARAMS:
*+ LDRAIN = 400p;400p
*+ LSOURCE = 100p;200p
*+ LGATE = 400p
+ KRES = 1.5;1.5
+ RCOUT = {0.627/Karea}
+ CJO = {215.45p*Karea}
```

```
+ RDSON = {0.113/Karea}



RCOND ICOND MGND 1MEG
RDISP IDISP MGND 1MEG
RMGND MGND 0 1MEG



VDMOS1 SOURCE 0 0
VDGATE GATE GATEX 0

XCEXTRA DRAIN SOURCEX CQS
+ PARAMS:
+ C = {CEXTRA - (KAREA-1)*COUTNOM}
*+ C = {CEXTRA*(2-Karea)}
*+ C={CEXTRA}
+ QC={QC}
+ FQ={FS}
+ IC=0

XLCEXTRAP SOURCEX SOURCE LQS
+ PARAMS:
+ L=1f;1n;400p;240p
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0

ibode 0 drain ac 1


************* RECTIFIER WITH TWO PARALLEL DIODES ********************
VDPRIM DRAIN DRAINP 0

XTRANSFORMER DRAINP TRANIN DRAINZ 0 IACP IDCP IACS IDCS MGND XFORMER ; same dots
*XTRANSFORMER DRAINP GNDISO DRAINZ 0 IACP IDCP IACS IDCS MGND XFORMER ; same dots
*XTRANSFORMER DRAINP GNDISO 0 DRAINZ IACP IDCP IACS IDCS MGND XFORMER ; opposite dots
+ PARAMS:
+ N2={N2}
+ RDCP = 1u
+ RDCS = 1u
+ LM = {LF}
+ LMCALC = {LF}
+ QP = 100
+ LL2 = {LREC+LDIV}
+ QS = 100
+ FQ = {FS}
+ IC =0
```

```
RexACP IACP MGND 1MEG
RexDCP IDCP MGND 1MEG
RexACS IACS MGND 1MEG
RexDCS IDCS MGND 1MEG



VSHORT GNDISO VSTAB 0
RSTAB2 VSTAB IN 1u


VDREC DRAINZ DRAINZ1 0
XLDIV DRAINZ1 DRAINX LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LDIV}
+ QL = 100000;{QLDIV} ;109
+ FQ={FS}
+ IC=0


XLREC DRAINX VREC LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LREC}
+ QL = 10000;{QLREC} ;109
+ FQ={FS}
+ IC=0

XCREC VRECC 0 CQS ; RECTIFIER CAPACITOR
+ PARAMS:
+ C={CREC}
+ QC={QC}
+ FQ={FS}
+ IC=0

VVREC VREC VRECD 0 ;current dummy
XLCRECP VRECD VRECC LQS    ; PARASITIC INDUCTOR FOR SHUNT CAPACITOR
+ PARAMS:
+ L={LCRECP}
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0


VDIODE VREC VREC1 0

DREC1 VREC1 OUT DSS16
*DREC2 VREC1 OUT DSS16
```

```
********** MODIFIED BREAKDOWN VOLTAGE**********************************
.MODEL Dss16 d
+IS=0.000119367 RS=0.0658135 N=1.73244 EG=0.64331
+XTI=0.5 BV=600 IBV=2e-08 CJO=2.05074e-10
+VJ=0.4 M=0.421584 FC=0.5 TT=0
+KF=0 AF=1


VLOAD OUT 0 {VOUT}



********************** GATE DRIVER ******************************
.PARAM
+ TR = 1/(100*FS)
+ PWIDTH = {DUTY/FS - 2*TR}

VDRIVE GATE 0 PULSE (0 5 0 {TR} {TR} {PWIDTH} {1/FS})



*---PARAMETER EXTRACTION
VPLF PLF 0 {LF}
VPL2F    PL2F   0  {L2F}
VPC2F    PC2F   0  {C2F}
VPLREC    PLREC  0   {LREC}
VPCREC    PCREC 0    {CREC}


**************************
*** SIMULATION CONTROL ***
**************************

.TRAN 200p 5U 4.9U 200P UIC
.PROBE

.END



**************************
*** LIST OF LIBRARIES   ***
**************************
.LIB CLASSE.LIB
.LIB RECT21.LIB
```

```
.LIB TRANSFORMER_LOSS3.LIB

*********************************************************
**** OPTIONS FOR BETTER CONVERGENCE       ***
*********************************************************
.OPTIONS ABSTOL=1nA
+          GMIN=1p
+          ITL1=6000
+          ITL2=4000
+          ITL4=5000
+          RELTOL=0.001
+          VNTOL=0.001mV
.OPTION STEPGMIN


*****************************************
*** OPTIONS TO KEEP A SMALL OUTPUT FILE ***
*****************************************
.OPTIONS
+ NOPAGE
+ NOBIAS
+ NOECHO
+ NOMOD
+ NUMDGT=8
.WIDTH OUT=132 ;TO PRINT MORE COLUMNS

*****************************************
*** SPECIAL PARAMETERS AND CONSTANTS    ***
*****************************************
.PARAM
+ PI=3.14159    ;GUESS WHAT


*************************
* CIRCUIT PARAMETERS     *
*************************

*---DESIGN PARAMETERS

.PARAM:

+FS=75MEG         ;SWITCHING FREQUENCY
+VIN=12  ;INPUT VOLTAGE
+VOUT=12          ;OUTPUT VOLTAGE
+DUTY=.33;.4;.33   ;DUTY CYCLE
+QC=5k          ;Q CAPACITORS
+QI=80        ;Q INDUCTORS
+TRMULT = 200   ;GATE DRIVE RISE-TIME MULTIPLIER (1/(TRMULT*FS))
+QLREC = 109
```

```
+QLF = 80
+QL2F = 80
+QLDIV = 109

*---L2F AND C2F PARAMETRIZED BY Z0 AND F2S
.PARAM:


*+ F2S = 100MEG
*+ Z0 = 10;13.83128
*+ L2F = {Z0/(2*PI*F2S)}
*+ C2F = {1/(2*PI*F2S*Z0)}

+ L2F = 15.0106n
+ C2F = 75p

*---COUT, CEXTRA, LF, LREC, CREC

+ KAREA = 1;2
+ CEXTRA = 122.2p;165p
+ COUTNOM = 97p
+ LF = 11.8176n;13.5n;12.5968n
+ LREC = 25.188n
+ LDIV = 9.945n;11.73n
+ CREC = 15.1406p;45.2024p
+ RGATE = 300M
+ LRECP = 1p
+ LCRECP = 1p
+ N2 = 1;1
+ CBLOCK = 100n
+ LDC = 1u
*************************
***CIRCUIT DESCRIPTION ***
*************************

*---DC source
VIN IN 0 {VIN}
Rbig IN 0 10g

*---LUMPED MR NETWORK

*XLF IN DRAIN LCHOKE       ;MULTIRESONANT ELEMENT
*+ PARAMS:
*+ L={LF}
*+ QL = {QLF}
*+ FQ={FS}
*+ IC=0
*+ RDC=10m
```

```
XL2F  DRAIN V2FX LQS        ;MULTIRESONANT ELEMENT
+ PARAMS:
+ L={L2F}
+ QL={QL2F}
+ FQ={2*FS}
+ IC=0


VD2F  V2FX V2F 0          ;DUMMY TO MEASURE CURRENT L2F

XC2F   V2F 0 CQS           ;MULTIRESONANT ELEMENT
+ PARAMS:
+ C={C2F}
+ QC={QC}
+ FQ={FS}
+ IC=0


*---NON LINEAR MOSFET MODEL
VDMOS DRAIN DRAINS 0

XSWITCH GATEX DRAINS SOURCE ICOND IDISP MGND MOSFETNLC
+ PARAMS:
*+ LDRAIN = 400p;400p
*+ LSOURCE = 100p;200p
*+ LGATE = 400p
+ KRES = 1.5;1.5
+ RCOUT = {0.627/Karea}
+ CJO = {215.45p*Karea}
+ RDSON = {0.113/Karea}



RCOND ICOND MGND 1MEG
RDISP IDISP MGND 1MEG
RMGND MGND 0 1MEG



VDMOS1 SOURCE 0 0
VDGATE GATE GATEX 0

XCEXTRA DRAIN SOURCEX CQS
+ PARAMS:
+ C = {CEXTRA - (KAREA-1)*COUTNOM}
*+ C = {CEXTRA*(2-Karea)}
*+ C={CEXTRA}
+ QC={QC}
```

```
+ FQ={FS}
+ IC=0

XLCEXTRAP SOURCEX SOURCE LQS
+ PARAMS:
+ L=1f;1n;400p;240p
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0

ibode 0 drain ac 1


********** RECTIFIER WITH TWO PARALLEL DIODES ********************
VDPRIM DRAIN DRAINP 0


XTRANSFORMER DRAINP GNDISO DRAINZ 0 IACP IDCP IACS IDCS MGND XFORMER ; same dots
*XTRANSFORMER DRAINP GNDISO 0 DRAINZ IACP IDCP IACS IDCS MGND XFORMER ; opposite dots
+ PARAMS:
+ N2={N2}
+ RDCP = 1u
+ RDCS = 1u
+ LM = {LF}
+ LMCALC = {LF}
+ QP = 42.3;100
+ LL2 = {LREC+LDIV}
+ QS = 84.6;100
+ FQ = {FS}
+ IC =0

RexACP IACP MGND 1MEG
RexDCP IDCP MGND 1MEG
RexACS IACS MGND 1MEG
RexDCS IDCS MGND 1MEG



VSHORT GNDISO VSTAB 0
RSTAB2 VSTAB IN 1u


VDREC DRAINZ DRAINZ1 0
XLDIV DRAINZ1 DRAINX LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LDIV}
+ QL = 100000;{QLDIV} ;109
+ FQ={FS}
+ IC=0
```

```
XLREC DRAINX VREC LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LREC}
+ QL = 10000;{QLREC} ;109
+ FQ={FS}
+ IC=0

XCREC VRECC 0 CQS ; RECTIFIER CAPACITOR
+ PARAMS:
+ C={CREC}
+ QC={QC}
+ FQ={FS}
+ IC=0

VVREC VREC VRECD 0 ;current dummy
XLCRECP VRECD VRECC LQS    ; PARASITIC INDUCTOR FOR SHUNT CAPACITOR
+ PARAMS:
+ L={LCRECP}
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0


VDIODE VREC VREC1 0

DREC1 VREC1 OUT DSS16
*DREC2 VREC1 OUT DSS16

*************** MODIFIED BREAKDOWN VOLTAGE*********************************
.MODEL Dss16 d
+IS=0.000119367 RS=0.0658135 N=1.73244 EG=0.64331
+XTI=0.5 BV=600 IBV=2e-08 CJO=2.05074e-10
+VJ=0.4 M=0.421584 FC=0.5 TT=0
+KF=0 AF=1

VLOAD OUT 0 {VOUT}

************************ GATE DRIVER ******************************
.PARAM
+ TR = 1/(100*FS)
+ PWIDTH = {DUTY/FS - 2*TR}

VDRIVE GATE 0 PULSE (0 5 0 {TR} {TR} {PWIDTH} {1/FS})



*---PARAMETER EXTRACTION
```

```
VPLF PLF 0 {LF}
VPL2F    PL2F   0  {L2F}
VPC2F    PC2F   0  {C2F}
VPLREC    PLREC  0   {LREC}
VPCREC    PCREC 0    {CREC}


**************************
*** SIMULATION CONTROL ***
**************************

.TRAN 200p 5U 4U 200P UIC
.PROBE
.END




**************************
*** LIST OF LIBRARIES   ***
**************************

.LIB CLASSE.LIB
.LIB RECT21.LIB
.LIB MITMV1_LOSS.LIB
.LIB TRANSFORMER_LOSS4.LIB

*********************************************************
**** OPTIONS FOR BETTER CONVERGENCE       ***
*********************************************************
.OPTIONS ABSTOL=1nA
+          GMIN=1p
+          ITL1=6000
+          ITL2=4000
+          ITL4=5000
+          RELTOL=0.001
+          VNTOL=0.001mV
.OPTION STEPGMIN

*****************************************
*** OPTIONS TO KEEP A SMALL OUTPUT FILE ***
*****************************************
.OPTIONS
+ NOPAGE
+ NOBIAS
+ NOECHO
+ NOMOD
+ NUMDGT=8
.WIDTH OUT=132 ;TO PRINT MORE COLUMNS
```

```
*****************************************
*** SPECIAL PARAMETERS AND CONSTANTS    ***
*****************************************
.PARAM
+ PI=3.14159    ;GUESS WHAT

*************************
* CIRCUIT PARAMETERS     *
*************************

*---DESIGN PARAMETERS

.PARAM:

+FS=75MEG         ;SWITCHING FREQUENCY
+VIN=12  ;INPUT VOLTAGE
+VOUT=12          ;OUTPUT VOLTAGE
+DUTY=.33;.4;.33   ;DUTY CYCLE
+QC=5k          ;Q CAPACITORS
+QI=80        ;Q INDUCTORS
+TRMULT = 200   ;GATE DRIVE RISE-TIME MULTIPLIER (1/(TRMULT*FS))
+QLREC = 109
+QLF = 80
+QL2F = 80
+QLDIV = 109

*---L2F AND C2F PARAMETRIZED BY Z0 AND F2S
.PARAM:
+ L2F = 15.0106n
+ C2F = 75p

*---COUT, CEXTRA, LF, LREC, CREC

+ KAREA = 1;2
+ CEXTRA = 65p;122.2p;165p
+ COUTNOM = 97p
+ LF = 11.8176n;13.5n;12.5968n
+ LREC = 25.188n
+ LDIV = 9.945n;11.73n
+ CREC = 10p;15.1406p;45.2024p
+ RGATE = 300M
+ LRECP = 1p
+ LCRECP = 1p
+ N2 = 1;1
+ CBLOCK = 100n
+ LDC = 1u
+ LPAR = 3n
+ QLPAR = 50
```

```
*************************
***CIRCUIT DESCRIPTION ***
*************************

*---DC source
VIN IN1 0 {VIN}
Rbig IN1 0 10g

*---LUMPED MR NETWORK

XLF IN1 IN LQS        ;MULTIRESONANT ELEMENT
+ PARAMS:
+ L={LPAR}
+ QL = {QLPAR}
+ FQ={FS}
+ IC=0
*+ RDC=10m


XL2F  DRAIN V2FX LQS        ;MULTIRESONANT ELEMENT
+ PARAMS:
+ L={L2F}
+ QL={QL2F}
+ FQ={2*FS}
+ IC=0


VD2F  V2FX V2F 0         ;DUMMY TO MEASURE CURRENT L2F

XC2F   V2F 0 CQS           ;MULTIRESONANT ELEMENT
+ PARAMS:
+ C={C2F}
+ QC={QC}
+ FQ={FS}
+ IC=0


*---NON LINEAR MOSFET MODEL
VDMOS DRAIN DRAINS 0

XSWITCH GATEX DRAINS SOURCE ICOND IDISP MGND MOSFETNLC
+ PARAMS:
*+ LDRAIN = 400p;400p
*+ LSOURCE = 100p;200p
*+ LGATE = 400p
+ KRES = 1.5;1.5
+ RCOUT = {0.627/Karea}
+ CJO = {215.45p*Karea}
+ RDSON = {0.113/Karea}
```

```
RCOND ICOND MGND 1MEG
RDISP IDISP MGND 1MEG
RMGND MGND 0 1MEG



VDMOS1 SOURCE 0 0
VDGATE GATE GATEX 0

XCEXTRA DRAIN SOURCEX CQS
+ PARAMS:
+ C = {CEXTRA - (KAREA-1)*COUTNOM}
*+ C = {CEXTRA*(2-Karea)}
*+ C={CEXTRA}
+ QC={QC}
+ FQ={FS}
+ IC=0

XLCEXTRAP SOURCEX SOURCE LQS
+ PARAMS:
+ L=1f;1n;400p;240p
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0

ibode 0 drain ac 1


************* RECTIFIER WITH TWO PARALLEL DIODES ********************
VDPRIM DRAIN DRAINP 0

XTRANSFORMER DRAINP IN DRAINZ 0 IACP IDCP IACS IDCS VRECTRANS MGND XFORMER ;
*XTRANSFORMER DRAINP GNDISO DRAINZ 0 IACP IDCP IACS IDCS VRECTRANS MGND XFORMER ; same dots
**XTRANSFORMER DRAINP GNDISO 0 DRAINZ IACP IDCP IACS IDCS MGND XFORMER ; opposite dots
+ PARAMS:
+ N2={N2}
+ RDCP = 1u
+ RDCS = 1u
+ LM = {LF}
+ LMCALC = {LF}
+ QP = 100
+ LL2 = {LREC+LDIV}
+ QS = 100
+ FQ = {FS}
+ IC =0
```

```
RexACP IACP MGND 1MEG
RexDCP IDCP MGND 1MEG
RexACS IACS MGND 1MEG
RexDCS IDCS MGND 1MEG
RexVREC VRECTRANS MGND 1MEG



*VSHORT GNDISO VSTAB 0
*RSTAB2 VSTAB IN 1u


VDREC DRAINZ DRAINZ1 0
XLDIV DRAINZ1 DRAINX LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LDIV}
+ QL = 100000;{QLDIV} ;109
+ FQ={FS}
+ IC=0


XLREC DRAINX VREC LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LREC}
+ QL = 10000;{QLREC} ;109
+ FQ={FS}
+ IC=0

XCREC VRECC 0 CQS ; RECTIFIER CAPACITOR
+ PARAMS:
+ C={CREC}
+ QC={QC}
+ FQ={FS}
+ IC=0

VVREC VREC VRECD 0 ;current dummy
XLCRECP VRECD VRECC LQS    ; PARASITIC INDUCTOR FOR SHUNT CAPACITOR
+ PARAMS:
+ L={LCRECP}
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0


VDIODE VREC VREC1 0

DREC1 VREC1 OUT DSS16
*DREC2 VREC1 OUT DSS16
```

```
********** MODIFIED BREAKDOWN VOLTAGE**********************************
.MODEL Dss16 d
+IS=0.000119367 RS=0.0658135 N=1.73244 EG=0.64331
+XTI=0.5 BV=600 IBV=2e-08 CJO=2.05074e-10
+VJ=0.4 M=0.421584 FC=0.5 TT=0
+KF=0 AF=1

VLOAD OUT 0 {VOUT}

********************** GATE DRIVER ****************************
.PARAM
+ TR = 1/(100*FS)
+ PWIDTH = {DUTY/FS - 2*TR}

VDRIVE GATE 0 PULSE (0 5 0 {TR} {TR} {PWIDTH} {1/FS})




*---PARAMETER EXTRACTION
VPLF PLF 0 {LF}
VPL2F    PL2F   0  {L2F}
VPC2F    PC2F   0  {C2F}
VPLREC    PLREC  0   {LREC}
VPCREC    PCREC 0    {CREC}


*************************
*** SIMULATION CONTROL ***
*************************

.TRAN 200p 5U 4U 200P UIC
.PROBE
.END



*************************
*** LIST OF LIBRARIES   ***
*************************

.LIB CLASSE.LIB
.LIB RECT21.LIB
.LIB MITMV1_LOSS.LIB
.LIB TRANSFORMER_LOSS3.LIB

*******************************************************
```

```
**** OPTIONS FOR BETTER CONVERGENCE       ***
*********************************************************
.OPTIONS ABSTOL=1nA
+          GMIN=1p
+          ITL1=6000
+          ITL2=4000
+          ITL4=5000
+          RELTOL=0.001
+          VNTOL=0.001mV
.OPTION STEPGMIN


*****************************************
*** OPTIONS TO KEEP A SMALL OUTPUT FILE ***
*****************************************
.OPTIONS
+ NOPAGE
+ NOBIAS
+ NOECHO
+ NOMOD
+ NUMDGT=8
.WIDTH OUT=132 ;TO PRINT MORE COLUMNS

*****************************************
*** SPECIAL PARAMETERS AND CONSTANTS    ***
*****************************************
.PARAM
+ PI=3.14159


*************************
* CIRCUIT PARAMETERS     *
*************************

*---DESIGN PARAMETERS

.PARAM:

+FS=75MEG         ;SWITCHING FREQUENCY
+VIN=12  ;INPUT VOLTAGE
+VOUT=12          ;OUTPUT VOLTAGE
+DUTY=.33;.4;.33   ;DUTY CYCLE
+QC=5k          ;Q CAPACITORS
+QI=80        ;Q INDUCTORS
+TRMULT = 200   ;GATE DRIVE RISE-TIME MULTIPLIER (1/(TRMULT*FS))
+QLREC = 109
+QLF = 80
+QL2F = 80
+QLDIV = 109
```

```
*---L2F AND C2F PARAMETRIZED BY ZO AND F2S
.PARAM:


*+ F2S = 100MEG
*+ ZO = 10;13.83128
*+ L2F = {ZO/(2*PI*F2S)}
*+ C2F = {1/(2*PI*F2S*ZO)}

+ L2F = 18.4745n
+ C2F = 60.9375p;32.8125p

*---COUT, CEXTRA, LF, LREC, CREC

+ KAREA = 1;2
+ CEXTRA = 188.8p;165p
+ COUTNOM = 97p
+ LF = 11.5109n;6.14162n;7.11162n;38.183n
+ LREC = 14.7561n;20.0195n;10.6103n
+ LDIV = 7.69n;11.73n
+ CREC = 14.3852p;45.2024p
+ RGATE = 300M
+ LRECP = 1p
+ LCRECP = 1p
+ N2 = 1;1
+ CBLOCK = 100n
+ LDC = 1u
*************************
***CIRCUIT DESCRIPTION ***
*************************

*---DC source
VIN IN 0 {VIN}
Rbig IN 0 10g

*---LUMPED MR NETWORK

*XLF IN DRAIN LCHOKE       ;MULTIRESONANT ELEMENT
*+ PARAMS:
*+ L={LF}
*+ QL = {QLF}
*+ FQ={FS}
*+ IC=0
*+ RDC=10m


XL2F  DRAIN V2FX LQS       ;MULTIRESONANT ELEMENT
+ PARAMS:
```

```
+ L={L2F}
+ QL={QL2F}
+ FQ={2*FS}
+ IC=0


VD2F  V2FX V2F 0        ;DUMMY TO MEASURE CURRENT L2F

XC2F   V2F 0 CQS          ;MULTIRESONANT ELEMENT
+ PARAMS:
+ C={C2F}
+ QC={QC}
+ FQ={FS}
+ IC=0


*---NON LINEAR MOSFET MODEL
VDMOS DRAIN DRAINS 0

XSWITCH GATEX DRAINS SOURCE ICOND IDISP MGND MOSFETNLC
+ PARAMS:
*+ LDRAIN = 400p;400p
*+ LSOURCE = 100p;200p
*+ LGATE = 400p
+ KRES = 1.5;1.5
+ RCOUT = {0.627/Karea}
+ CJO = {215.45p*Karea}
+ RDSON = {0.113/Karea}



RCOND ICOND MGND 1MEG
RDISP IDISP MGND 1MEG
RMGND MGND 0 1MEG



VDMOS1 SOURCE 0 0
VDGATE GATE GATEX 0

XCEXTRA DRAIN SOURCEX CQS
+ PARAMS:
+ C = {CEXTRA - (KAREA-1)*COUTNOM}
*+ C = {CEXTRA*(2-Karea)}
*+ C={CEXTRA}
+ QC={QC}
+ FQ={FS}
+ IC=0
```

```
XLCEXTRAP SOURCEX SOURCE LQS
+ PARAMS:
+ L=1f;1n;400p;240p
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0

ibode 0 drain ac 1


********** RECTIFIER WITH TWO PARALLEL DIODES ********************
VDPRIM DRAIN DRAINP 0


XTRANSFORMER DRAINP GNDISO DRAINZ 0 IACP IDCP IACS IDCS MGND XFORMER ; same dots
*XTRANSFORMER DRAINP GNDISO 0 DRAINZ IACP IDCP IACS IDCS MGND XFORMER ; opposite dots
+ PARAMS:
+ N2={N2}
+ RDCP = 1u
+ RDCS = 1u
+ LM = {LF}
+ LMCALC = {LF}
+ QP = 100
+ LL2 = {LREC+LDIV}
+ QS = 100
+ FQ = {FS}
+ IC =0

RexACP IACP MGND 1MEG
RexDCP IDCP MGND 1MEG
RexACS IACS MGND 1MEG
RexDCS IDCS MGND 1MEG



VSHORT GNDISO VSTAB 0
RSTAB2 VSTAB IN 1u


VDREC DRAINZ DRAINZ1 0
XLDIV DRAINZ1 DRAINX LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LDIV}
+ QL = 100000;{QLDIV} ;109
+ FQ={FS}
+ IC=0


XLREC DRAINX VREC LQS    ; RECTIFIER INDUCTOR
```

```
+ PARAMS:
+ L=1p;{LREC}
+ QL = 10000;{QLREC} ;109
+ FQ={FS}
+ IC=0

XCREC VRECC 0 CQS ; RECTIFIER CAPACITOR
+ PARAMS:
+ C={CREC}
+ QC={QC}
+ FQ={FS}
+ IC=0

VVREC VREC VRECD 0 ;current dummy
XLCRECP VRECD VRECC LQS   ; PARASITIC INDUCTOR FOR SHUNT CAPACITOR
+ PARAMS:
+ L={LCRECP}
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0


VDIODE VREC VREC1 0

DREC1 VREC1 OUT DSS16
DREC2 VREC1 OUT DSS16

*************** MODIFIED BREAKDOWN VOLTAGE*********************************
.MODEL Dss16 d
+IS=0.000119367 RS=0.0658135 N=1.73244 EG=0.64331
+XTI=0.5 BV=600 IBV=2e-08 CJO=2.05074e-10
+VJ=0.4 M=0.421584 FC=0.5 TT=0
+KF=0 AF=1



VLOAD OUT 0 {VOUT}




************************ GATE DRIVER ******************************
.PARAM
+ TR = 1/(100*FS)
+ PWIDTH = {DUTY/FS - 2*TR}

VDRIVE GATE 0 PULSE (0 5 0 {TR} {TR} {PWIDTH} {1/FS})
```

```
*---PARAMETER EXTRACTION
VPLF PLF 0 {LF}
VPL2F    PL2F   0  {L2F}
VPC2F    PC2F   0  {C2F}
VPLREC    PLREC  0   {LREC}
VPCREC    PCREC 0    {CREC}


*************************
*** SIMULATION CONTROL ***
*************************

.TRAN 200p 5U 4.9U 200P UIC
.PROBE
.END



*************************
*** LIST OF LIBRARIES   ***
*************************
.LIB CLASSE.LIB
.LIB MITMV1_LOSS.LIB
.LIB TRANSFORMER_LOSS3.LIB
.LIB RECT21.LIB

*********************************************************
**** OPTIONS FOR BETTER CONVERGENCE       ***
*********************************************************
.OPTIONS ABSTOL=1nA
+          GMIN=1p
+          ITL1=6000
+          ITL2=4000
+          ITL4=5000
+          RELTOL=0.001
+          VNTOL=0.001mV
.OPTION STEPGMIN


*****************************************
*** OPTIONS TO KEEP A SMALL OUTPUT FILE ***
*****************************************
.OPTIONS
+ NOPAGE
```

```
+ NOBIAS
+ NOECHO
+ NOMOD
+ NUMDGT=8
.WIDTH OUT=132 ;TO PRINT MORE COLUMNS

*****************************************
*** SPECIAL PARAMETERS AND CONSTANTS    ***
*****************************************
.PARAM
+ PI=3.14159     ;GUESS WHAT


*************************
* CIRCUIT PARAMETERS     *
*************************

*---DESIGN PARAMETERS

.PARAM:

+FS=75MEG         ;SWITCHING FREQUENCY
+VIN=12 ;INPUT VOLTAGE
+VOUT=12          ;OUTPUT VOLTAGE
+DUTY=.33;.4;.33    ;DUTY CYCLE
+QC=5k           ;Q CAPACITORS
+QI=80        ;Q INDUCTORS
+TRMULT = 200    ;GATE DRIVE RISE-TIME MULTIPLIER (1/(TRMULT*FS))
+QLREC = 109
+QLF = 80
+QL2F = 80
+QLDIV = 109

*---L2F AND C2F PARAMETRIZED BY ZO AND F2S
.PARAM:


+ F2S = 150MEG
+ ZO = 10;14.4147
+ L2F = {ZO/(2*PI*F2S)}
+ C2F = {1/(2*PI*F2S*ZO)}

*+ L2F = 15.0106n
*+ C2F = 75p

*---COUT, CEXTRA, LF, LREC, CREC

+ KAREA = 1
+ CEXTRA = 190p;220p
```

```
+ COUTNOM = 97p
+ LF = 8.389n;11.8176n
+ LREC = 30.616n;25.188n
+ LDIV = 0;9.945n
+ CREC = 30p;50p;15.1406p
+ RGATE = 300M
+ LRECP = 1n
+ LCRECP = 1n
+ N2 = 1
+ CBLOCK = 100n
+ LDC = 1u
*************************
***CIRCUIT DESCRIPTION ***
*************************

*---DC source
VIN IN 0 {VIN}
Rbig IN 0 10g

*---LUMPED MR NETWORK

*XLF IN DRAIN LCHOKE       ;MULTIRESONANT ELEMENT
*+ PARAMS:
*+ L={LF}
*+ QL = {QLF}
*+ FQ={FS}
*+ IC=0
*+ RDC=10m


XL2F  DRAIN V2FX LQS       ;MULTIRESONANT ELEMENT
+ PARAMS:
+ L={L2F}
+ QL={QL2F}
+ FQ={2*FS}
+ IC=0


VD2F  V2FX V2F 0         ;DUMMY TO MEASURE CURRENT L2F

XC2F   V2F 0 CQS          ;MULTIRESONANT ELEMENT
+ PARAMS:
+ C={C2F}
+ QC={QC}
+ FQ={FS}
+ IC=0


*---NON LINEAR MOSFET MODEL
```

```
VDMOS DRAIN DRAINS 0

XSWITCH GATEX DRAINS SOURCE ICOND IDISP MGND MOSFETNLC
+ PARAMS:
+ LDRAIN = 400p;400p
+ LSOURCE = 100p;200p
*+ LGATE = 400p
+ KRES = 1.5;1.5
+ RCOUT = {0.460/Karea}
+ CJO = {348.54p*Karea}
+ RDSON = {0.181/Karea}



RCOND ICOND MGND 1MEG
RDISP IDISP MGND 1MEG
RMGND MGND 0 1MEG

****For plotting only!!!!
Var Varea 0   {Karea}

VDMOS1 SOURCE 0 0
VDGATE GATE GATEX 0

XCEXTRA DRAIN SOURCEX CQS
+ PARAMS:
+ C = {CEXTRA - (KAREA-1)*COUTNOM}
*+ C = {CEXTRA*(2-Karea)}
*+ C={CEXTRA}
+ QC={QC}
+ FQ={FS}
+ IC=0

XLCEXTRAP SOURCEX SOURCE LQS
+ PARAMS:
+ L=400p;240p
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0

ibode 0 drain ac 1


************ RECTIFIER WITH TWO PARALLEL DIODES ********************
VDPRIM DRAIN DRAINP 0


XTRANSFORMER DRAINP GNDISO DRAINZ 0 IACP IDCP IACS IDCS MGND XFORMER ; same dots
*XTRANSFORMER DRAINP GNDISO 0 DRAINZ IACP IDCP IACS IDCS MGND XFORMER ; opposite dots
```

```
+ PARAMS:
+ N2={N2}
+ RDCP = 10m
+ RDCS = 29m
+ LM = {LF}
+ LMCALC = {LF}
+ QP = 42.31;100
+ LL2 = {LREC+LDIV}
+ QS = 84.62;100
+ FQ = {FS}
+ IC =0

RexACP IACP MGND 1MEG
RexDCP IDCP MGND 1MEG
RexACS IACS MGND 1MEG
RexDCS IDCS MGND 1MEG



VSHORT GNDISO VSTAB 0
RSTAB2 VSTAB IN 1u


VDREC DRAINZ DRAINZ1 0
XLDIV DRAINZ1 DRAINX LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LDIV}
+ QL = 100000;{QLDIV} ;109
+ FQ={FS}
+ IC=0


XLREC DRAINX VREC LQS    ; RECTIFIER INDUCTOR
+ PARAMS:
+ L=1p;{LREC}
+ QL = 10000;{QLREC} ;109
+ FQ={FS}
+ IC=0

XCREC VRECC 0 CQS ; RECTIFIER CAPACITOR
+ PARAMS:
+ C={CREC}
+ QC={QC}
+ FQ={FS}
+ IC=0

VVREC VREC VRECD 0 ;current dummy
XLCRECP VRECD VRECC LQS    ; PARASITIC INDUCTOR FOR SHUNT CAPACITOR
+ PARAMS:
```

```
+ L={LCRECP}
+ QL=80000;{QI}
+ FQ={FS}
+ IC=0


VDIODE VREC VREC1 0

*DREC1 VREC1 OUT DSS16
**DREC2 VREC1 OUT DSS16

XDREC VREC1 OUT DIODENL
+ PARAMS:
+ LDS = {LRECP}
+ VDON = 0.55
+ RDS = 55e-3
+ CJO = 98.185P
+ VJ = 0.454055475001809
+ M = 0.469138223594758
+ RC = 425.38M
+ FS = {FS}


XDREC2 VREC1 OUT DIODENL
+ PARAMS:
+ LDS = {LRECP}
+ VDON = 0.55
+ RDS = 55e-3
+ CJO = 98.185P
+ VJ = 0.454055475001809
+ M = 0.469138223594758
+ RC = 425.38M
+ FS = {FS}


********** MODIFIED BREAKDOWN VOLTAGE**********************************
.MODEL Dss16 d
+IS=0.000119367 RS=0.0658135 N=1.73244 EG=0.64331
+XTI=0.5 BV=600 IBV=2e-08 CJO=2.05074e-10
+VJ=0.4 M=0.421584 FC=0.5 TT=0
+KF=0 AF=1

*10MQ100NPbF
*****************************************
* SPICE Model Diode *
*****************************************

*Define diode model
.MODEL 10MQ100NPbF d
```

```
+IS=1.0022E-6 N=1.9566 RS=52.583E-3 IKF=.62128 XTI=2 EG=1.1400
+CJO=93.409E-12 M=.44859 VJ=.3905 ISR=10.010E-21 NR=4.9950 BV=103.90
+IBV=248.70E-6

*****************************************



VLOAD OUT 0 {VOUT}



********************** GATE DRIVER ******************************
.PARAM
+ TR = 1/(100*FS)
+ PWIDTH = {DUTY/FS - 2*TR}

VDRIVE GATE 0 PULSE (0 5 0 {TR} {TR} {PWIDTH} {1/FS})


 1


*---PARAMETER EXTRACTION
VPLF PLF 0 {LF}
VPL2F    PL2F   0  {L2F}
VPC2F    PC2F   0  {C2F}
VPLREC    PLREC  0   {LREC}
VPCREC    PCREC 0    {CREC}


*************************
*** SIMULATION CONTROL ***
*************************
.TRAN 200p 5U 4.0U 200P UIC
.PROBE
.END
```

A.4 Boards and Schematics

Table A.1: Measured Converter Parameters and Components

Parameter	**Isolated Φ_2**	**Φ_2 boost**
V_{IN}	8-16 V	10 - 20 V
V_{OUT}	12.7 V	32 V
P_{OUT}	6 W, nom	14 W, nom
Efficiency	73%, nom	85%, nom
F_{SW}	75 MHz	50 MHz
Ripple	50 mV (0.4%)	100 mV (0.3%)
C_F	121 pF	NA
C_{2F}	82 pF	115 pF
C_{REC}	8.2 pF	24 pF
C_{OUT}	10 x 0.1	10 x 0.1 uFμu
C_{EXT}	NA	47 pF
Diode	10MQ100C	S310-100V, 3A
L_{u1}	3.6 nH	NA
L_{l1}	6.5 nH	NA
L_{l2}	11.5 nH	NA
L_F	NA	43 nH
L_{REC}	NA	90 nH

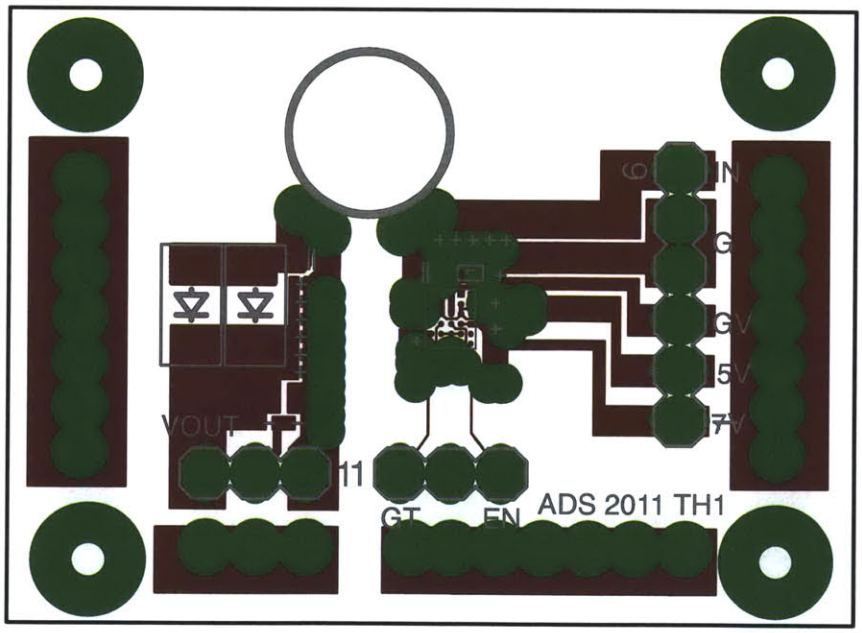


Figure A.1: Isolated Φ_2 converter board - top copper layer

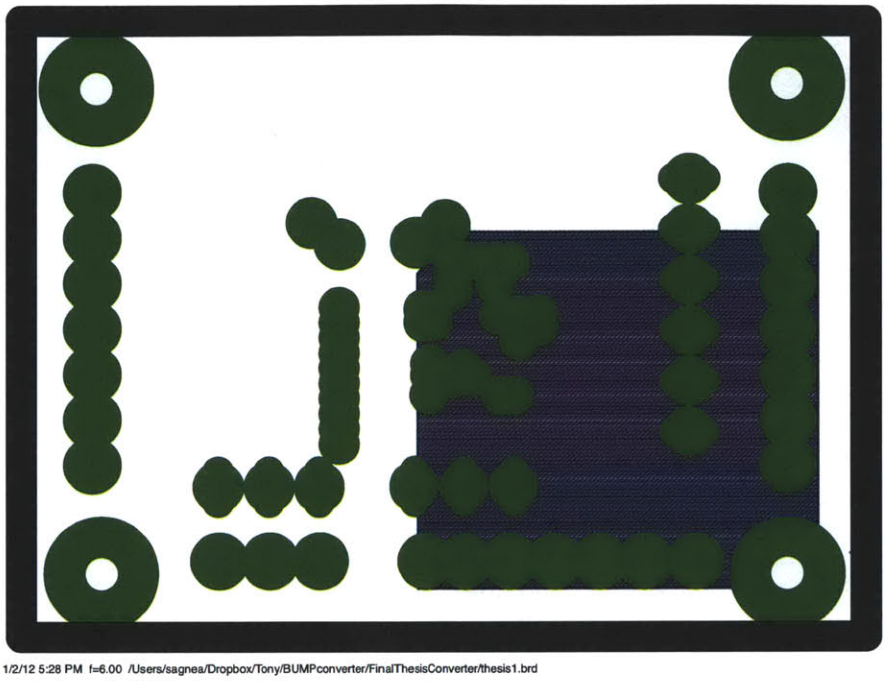


Figure A.2: Isolated Φ_2 converter board - layer-2 copper

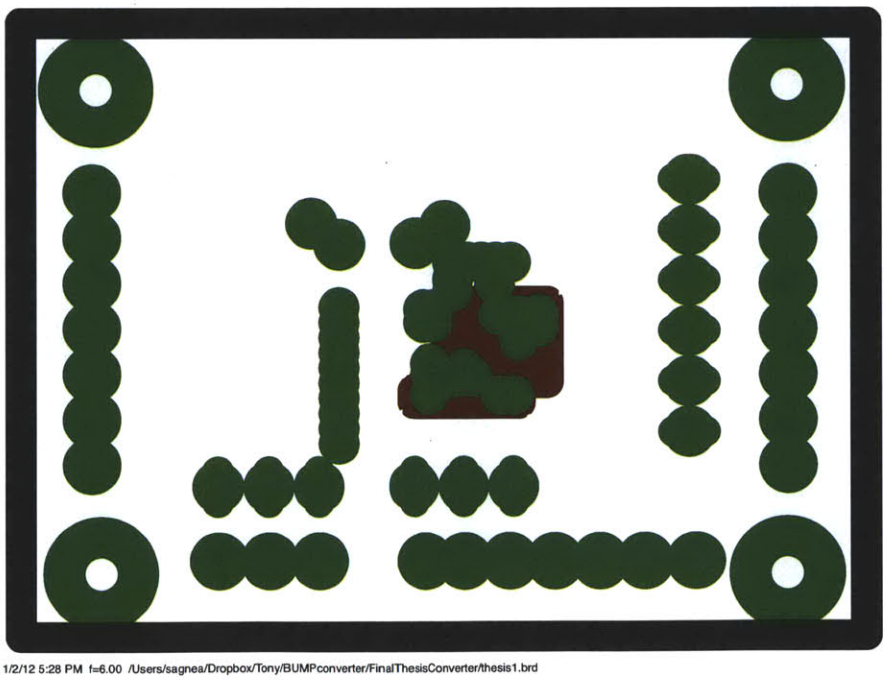


Figure A.3: Isolated Φ_2 converter board - layer-3 copper

Figure A.4: Isolated Φ_2 converter board - bottom copper layer

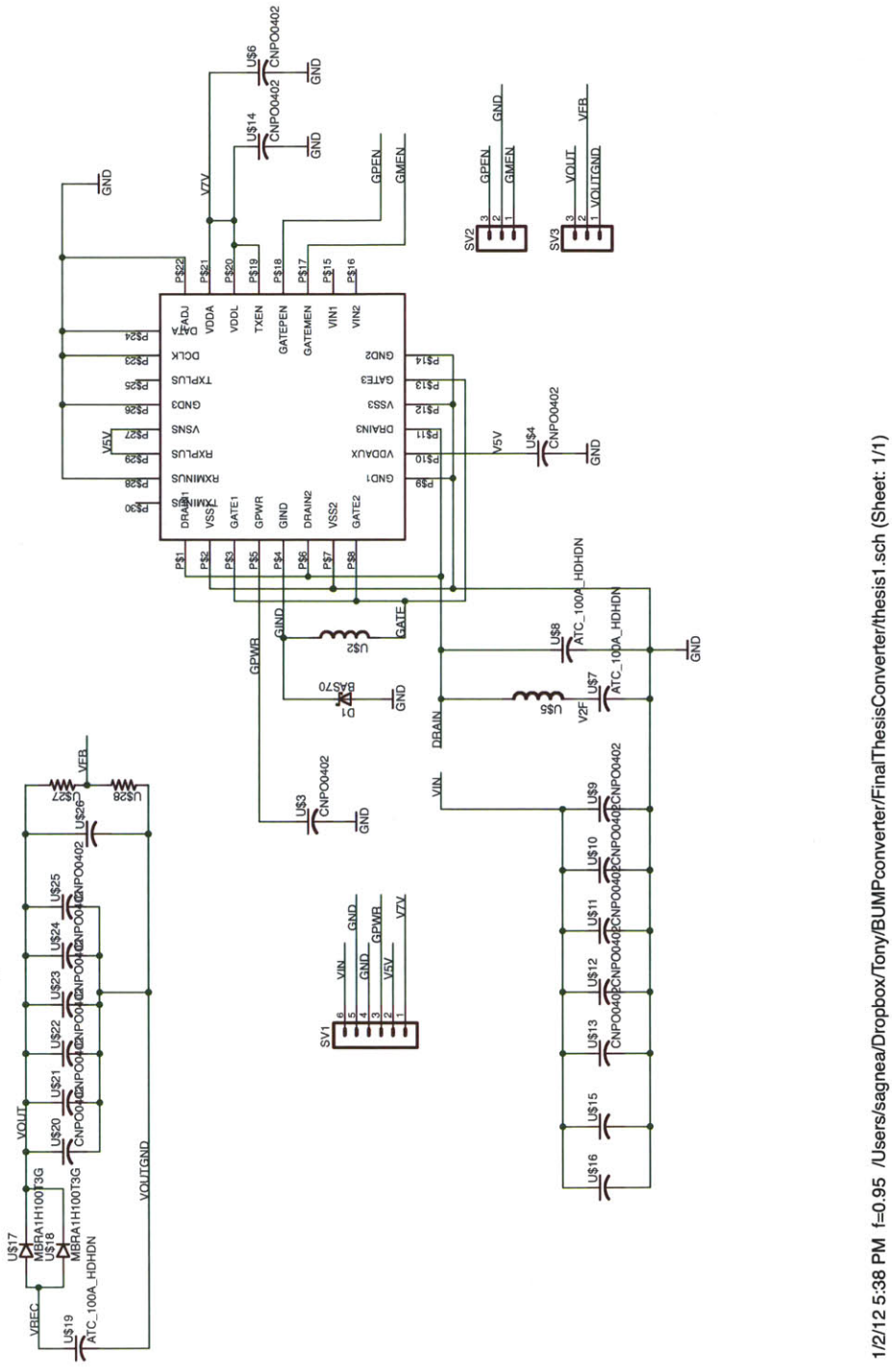


Figure A.5: Isolated Φ_2 converter board - schematic

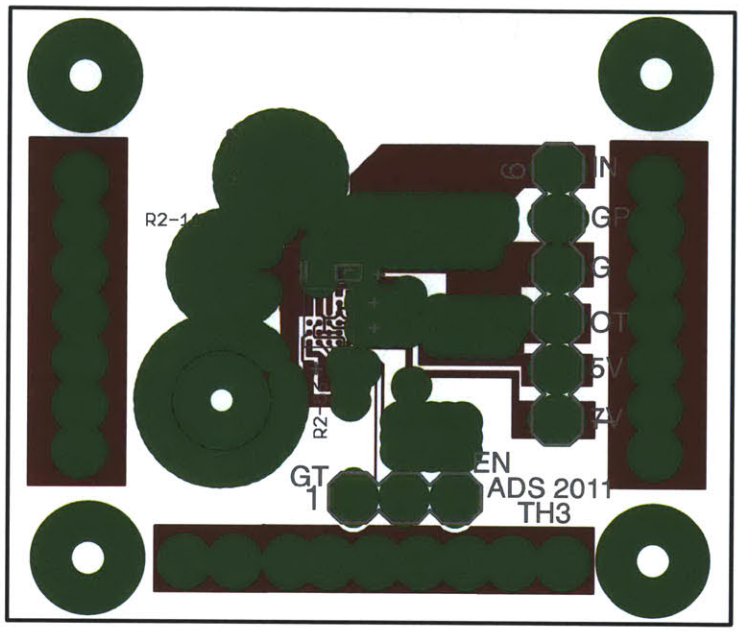


Figure A.6: Φ_2 boost converter board - top copper layer

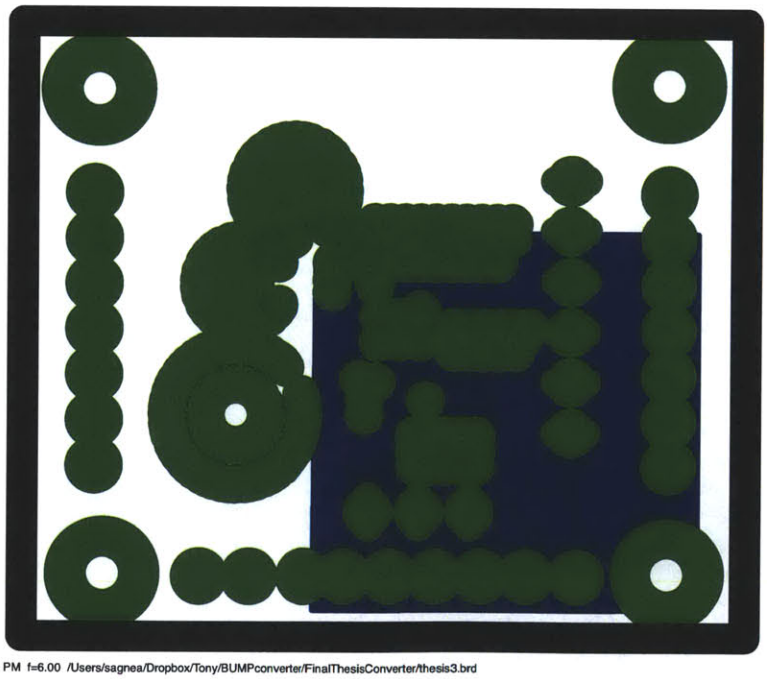


Figure A.7: Φ_2 boost converter board - layer-2 copper

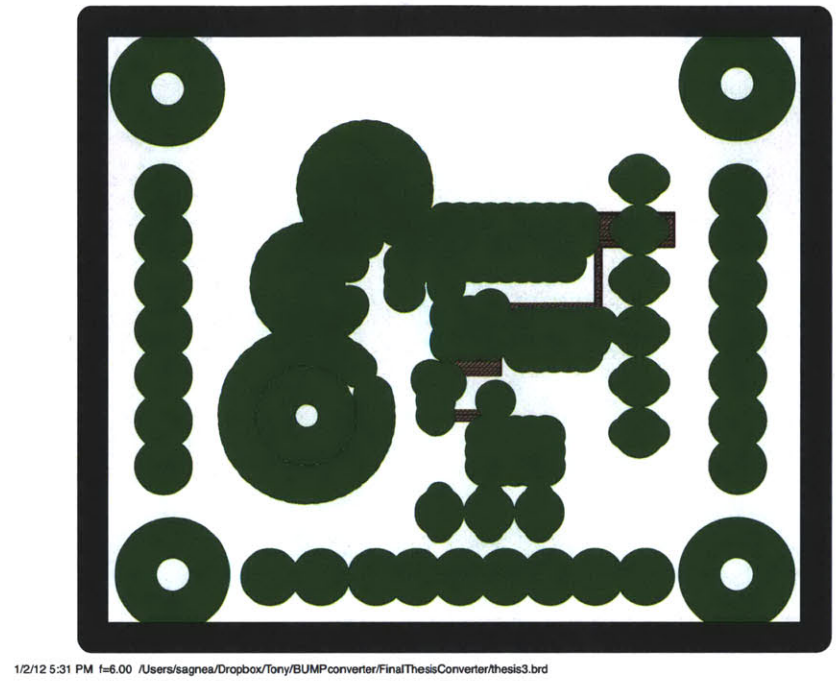


Figure A.8: Φ_2 boost converter board - layer-3 copper

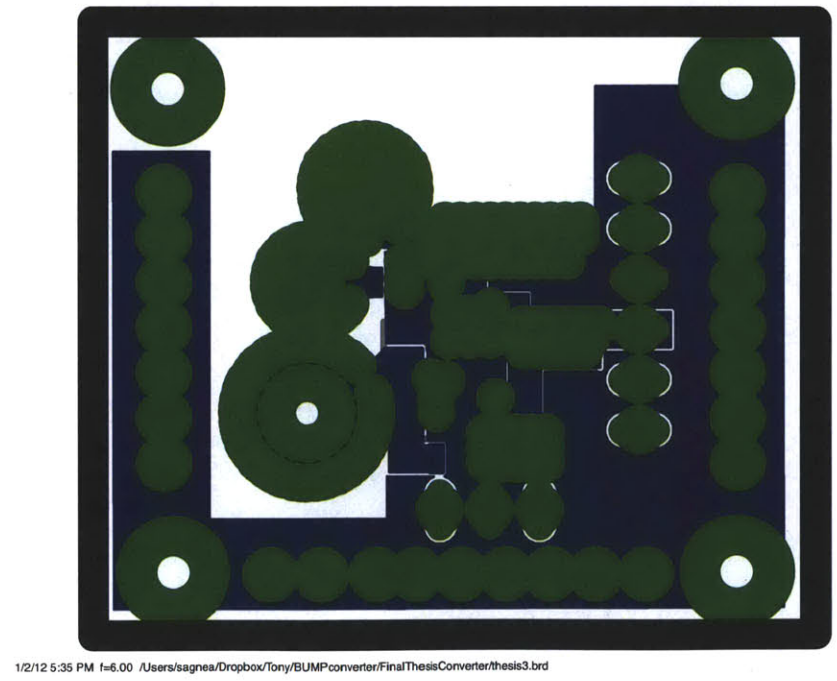


Figure A.9: Φ_2 boost converter board - bottom copper layer

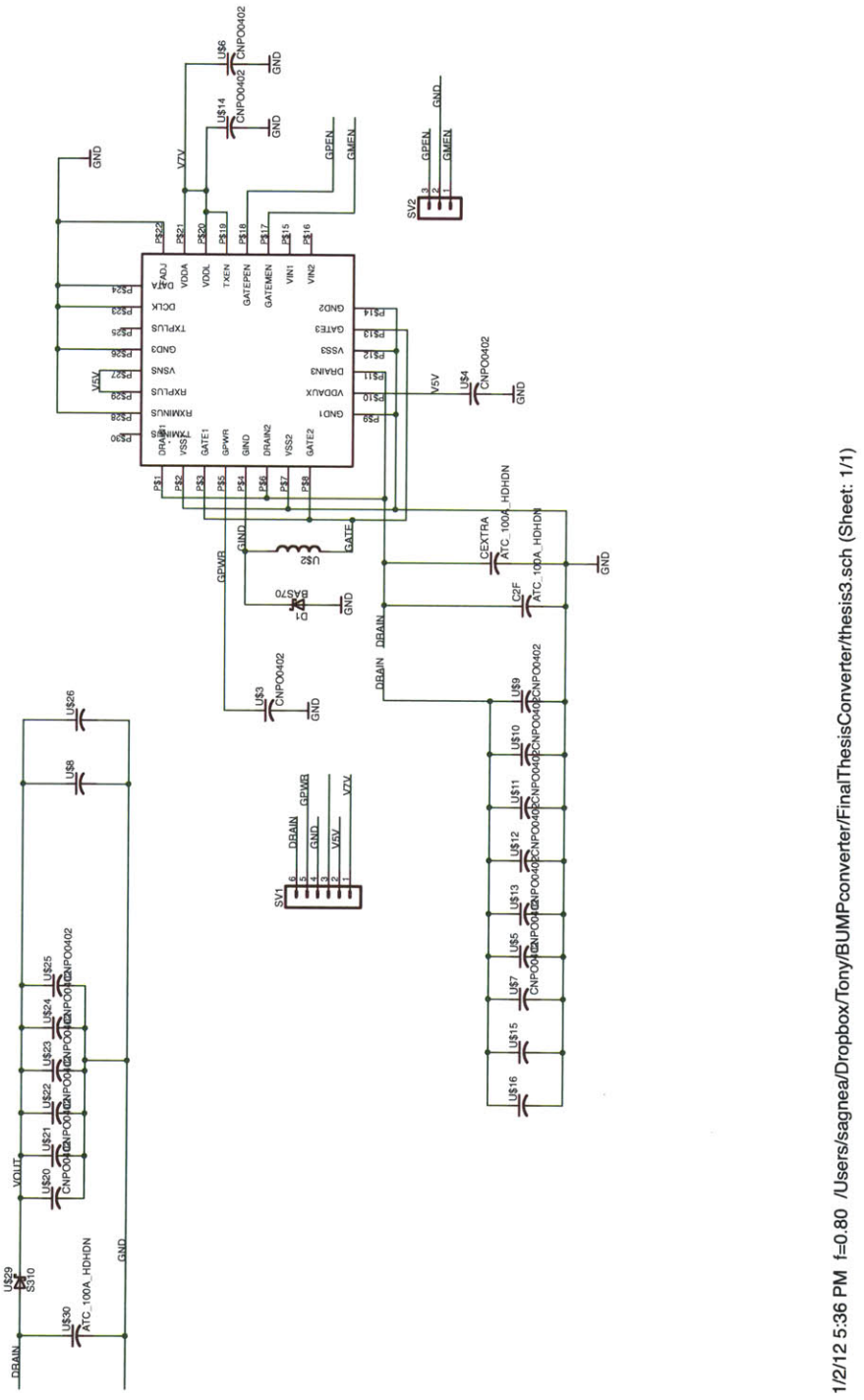


Figure A.10: Φ_2 boost converter board - schematic

Bibliography

[1] W. G. Hurley and M. C. Duffy. Calculation of self and mutual impedances in planar magnetic structures. *IEEE Transactions on Magnetics*, Vol. 31. No. 4:2416–2422, 1995.

[2] J.G. Kassakian, M.F. Schlecht, and G.C. Verghese. *Principles of Power Electronics.* (Addison-Wesley), NY, 1991.

[3] J.M. Rivas, J. Shafran, R.S. Wahby, and D.J. Perreault. New architectures for radio-frequency dc/dc power conversion. *IEEE Transactions on Power Electronics*, 21(2):380–393, March 2006.

[4] Juan M. Rivas. *Radio Frequency dc-dc Power Conversion.* PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 2006.

[5] R.J. Gutmann. Application of RF circuit design principles to distributed power converters. *IEEE Transactions on Industrial Electronics and Control Instrumentation*, IEC127(3):156–164, 1980.

[6] Anthony D. Sagneri, Robert C. N. Pilawa-Podgurski, Juan M. Rivas, David I. Anderson, and David J. Perreault. Very-high-frequency resonant boost converters. *IEEE Transactions on Power Electronics*, 24:1654–1665, 2009.

[7] Jinying Hu, Anthony D. Sagneri, Juan M. Rivas, Seth M. Davis, and David J. Perreault. High frequency resonant sepic converter with wide input and output voltage ranges. In *Power Electronics Specialist Conference Proceedings 2008*, June 2008.

[8] David J. Perreault, Jingying Hu, Juan M. Rivas, Yehui Han, Olivia Leitermann, Robert C. N. Pilawa-Podgurski, ANthony D. Sagneri, and Charles R. Sullivan. Opportunities and challenges in very high frequency power conversion. In *The proceedings of the Applied Power Electronics Conference and Exposition*, 2009.

[9] Joshua W. Phinney. *Multi-resonant Passive Components for Power Conversion.* Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA., June 2005.

[10] Francis Grover. *Inductance Calculations.* New York: Dover, 1946.

[11] *Improved Calculation of Core Loss with Nonsinusoidal Waveforms*, 4 Oct 2001.

[12] Duncan A. Grant and John Gowar. *Power Mosfets: Theory and Applications*. John Wiley & Sons, 1989.

[13] Joshua W. Phinney. Filters with active tuning for power applications. Master's thesis, Massachusetts Institute of Technology, 2001.

[14] 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*, SC-10(3):168–176, June 1975.

[15] Nathan O. Sokal. Class-E RF Power Amplifiers. *QEX*, pages 9–20, Jan/Feb 2001.

[16] R. Redl and N.O. Sokal. A new class-e dc/dc converter family with reduced parts count: Derivation, topologies, and design considerations. In *Technical Papers of the Fourth International High Frequency Power Conversion Conference*, pages 395–415, 1989.

[17] R. Redl and N.O. Sokal. A 14 MHz 100 Watt class E resonant converter: Principles, design considerations, and measured performance. In *17^{th} Annual IEEE Power Electronics Specialists Conference Proceedings*, pages 68–77, June 1986.

[18] M. Iwadare, S. Mori, and K. Ikeda. Even harmonic resonant class e tuned power amplifier without rf choke. In *Electronics and Communications in Japan*, volume 79, pages 23–30, Jan. 1996.

[19] J.J. Jozwik and M.K. Kazimierczuk. Analysis and design of class-e dc/dc converter. *IEEE Transactions on Industrial Electronics*, 37(2):173–183, April 1990.

[20] H. Koizumi, Iwadare M., and Mori S. Class e^2 dc/dc converter with second harmonic resonant class e inverter and class e rectifier. In *Third Annual Applied Power Electronics Conference Proceedings*, pages 1012–1018, 1994.

[21] H. Koizumi, H. Sekiya, M. Matsuo, S. Mori, and I. Sasae. Resonant dc/dc converter with class de inverter and class e rectifier using thinned-out method (deleting some of the pulses to the rectifier). *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on]*, 48(1):123–126, Jan. 2001.

[22] Michael J. Chudobiak. The use or parasitic nonlinear capacitors in class e amplifiers. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(12):941–944, December 1994.

[23] J.M. Rivas, R.S. Wahby, J.S. Shafran, and D.J. Perreault. New architectures for radio-frequency dc/dc power conversion. In *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, volume 5, pages 4074–4084Vol.5, 20-25 June 2004.

[24] Juan M. Rivas, Yehui Han, Olivia Leitermann, Anthony D. Sagneri, and David J. Perreualt. A high-frequency resonant inverter topology with low-voltage stress. *IEEE Transactions on Power Electronics*, 23(No. 4):1759–1771, July 2008.

[25] Alan Hastings. *The Art of Analog Layout.* Pearson, Prentice Hall, 2006.

[26] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital Integrated Circuits, A Design Perspective.* Prentice Hall, 2003.

[27] Tonio Biondi, Giuseppe Greco, Maria Concetta Allia, Salvatore Fabio Liotta, Gaetano Bazzano, and Salvatore Rinaudo. Distributed modeling of layout parasitics in large-area high-speed silicon power devices. *IEEE Transactions on Power Electronics*, Vol. 22(No. 5):1847–1856, September 2007.

[28] Anthony D. Sagneri. The design of a very high frequency dc-dc boost converter. Master's thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, February 2007.

[29] Douglas Brisbin, Philipp Lindorfer, and Prasadd Chaparala. Anomalous safe operating area and hot carrier degredation of nldmos devices. *Device and Materials Reliability, IEEE Transactions on*, Vol. 6, No.3:364–370, 2006.

[30] P. Moens, J. Mertens, F. Bauwens, P. Joris, W. De Ceuninck, and M. Tack. A compreshensive model for hot carrier degredation in ldmos transistors. In *IEEE 45th Annual International Reliability Physics Symposium*, 2007.

[31] Peter Moens, Geert Van den Bosch, Catherine De Keukeleire, Robin Degraeve, Marniz Tack, and Guido Groeseneken. Hot hole degredation effects in lateral ndmos transistors. *Electron Devices, IEEE Transactions on*, Vol. 51, No. 10:1704–1710, 2004.

[32] Yehui Han and David Perreualt. Inductor design methods with low-permeability rf core materials. In *Energy Conversion Congress and Exposition, 2010 ECCE*, 2010.

[33] Yehui Han, Grace Cheung, A. Li, Charles Sullivan, and David Perreault. Evaluation of magnetic materials for very high frequency power applications. *Power Electronics, IEEE Transactions On*, PP, Issue 99, 2011.

[34] Gerald W. Ludwig and Sayed-Amr El-Hamamsy. Coupled inductance and reluctance models of magnetic components. *IEEE Transactions on Power Electronics*, Vol. 6:pp. 240–250, April 1991.

[35] Rudolph P. Severns and Gordon E. Bloom. *Modern DC-DC Switchmode Power Converter Circuits.* Bloom Associates, Inc., 1985.

[36] T. C. Neugebauer and D. J. Perreault. Filters with inductance cancellation using printed circuit board transformers. *IEEE Transactions on Power Electronics*, vo. 19, no. 3:pp 591–602, May 2004.

[37] J. Phinney, J.H. Lang, and D.J. Perreault. Multi-resonant microfabricated inductors and transformers. In *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, volume 6, pages 4527–4536Vol.6, 20-25 June 2004.

[38] M. Ehsani, Oskar Heino Stielau, Jacobus Daniel van Wyk, and Ira J. Pitel. Integrated reactive components in power electronic circuits. *IEEE Transactions on Power Electronics*, vol. 8:pp. 208–215, 1993.

[39] I. W. Hofsajer, J.A. Ferreira, and J.D. van Wyk. Optimised planar integrated l-c-t components. In *Power Electronics Specialists Conference, 1997.*, volume vol. 2, pages pp. 1157–1163, June 1997.

[40] Kien Lai-Dac, Yvs Lembeye, Abdelhadi Besri, and Jean-Pierre Keradec. Analytical modeling of losses for high frequency planar lct components. In *The 2009 IEEE Energy Conversion Congress and Exposition (ECCE), San Jose*, September 2009.

[41] J. H. Spreen. Electrical terminal representation of conductor loss in transformers. *IEEE Transactions on Power Electronics*, vol. 15, no. 4(no. 4):pp. 424–429, Oct. 1990.

[42] S. C. Tang, S. Y. Hui, and Henry Shu-Hung Chung. Coreless planar printed-circuit-board (pcb) transformers- a fundamental concept for signal and energy transfer. *IEEE Transactions on Power ELectronics*, Vol. 15:pp. 931–941, Sept. 2000.

[43] J.C. Maxwell. *A Treastise on Electricity and Magnetism.* Oxford, Clarendon Press, 1873.

[44] M. Kamon, M. J. Tsuk, and J. White. Fasthenry: A multipole-accelerated 3-d inductance extraction program. *Microwave Theory and Techniques, IEEE Transactions on*, 42:1750–1758, 1994.

[45] S. C. Tang, S. Y. (Ron) Hui, and Henry Shu-Hung Chung. Coreless printed circuit board (pcb) transformers with multiple secondary windings for complementary gate drive circuits. *Power Electronics, IEEE Transactions On*, 14:431–437, 1999.

[46] S.C. Tang, S.Y.R. Hui, and H. Chung. Characterization of coreless printed circuit board (pcb) transformers. In *Power Electronics Specialists Conference, 1999. PESC 99. 30th Annual IEEE*, 1999.

[47] C. M. Arturi, A. Gandelli, and F. Grassi. High frequency model of printed-circuit-board based three-winding transformers. In *Circuits and Systems 2002. WMSCAS-2002. The 2002 45th Midwest Symposium*, 2002.

[48] R. P. Ribas, J. Lescot, J.-L. Leclercq, J. M. Karam, and F. Ndagijimana. Micromachined microwave planar spiral inductors and transformers. *Microwave Theory and Techniques, IEEE Transactions on*, 48:1326–1365, 2000.

[49] W. C. Ho and M. H. Pong. Analysis and design of printed windings of power transformers using partial inductance method. In *Industrial Electronics, Contol, and Instrumentation, Proceedings of the 1995 IEEE IECON 21st International Conference on*, Nov. 1995.

[50] J.M. Rivas, D. Jackson, O. Leitermann, A.D. Sagneri, Yehui Han, and D.J. Perreault. Design considerations for very high frequency dc-dc converters. In *37th Annual IEEE Power Electronics Specialists Conference Proceedings*, pages 1–11, 18-22 June 2006.

[51] James R. Warren III, Kathryn A. Rosowski, and David J. Perreault. Transistor selection and design of a VHF dc-dc power converter. In *IEEE Transactions on Power Electronics*, volume 23, pages 27–37, January 2008.

[52] Ilkaa Nissinen and Juha Kostamovaara. A low voltage constant current-voltage reference circuit. In *Circuits and Systems 2004, ISCAS '04, Proceedings of the 2004 International Symposium On*, volume Volume 1, pages I–381 to I–384, 2004.

[53] Ilkka Nissinen and Juha Kostamovaara. A temperature stabilized cmos ring oscillator for a time-to-digital converter of a laser radar. In *Instrumentation and Measurement Technology Conference, 2004. IMTC 04, Proceedings of the 21st IEEE*, volume Vol. 3, pages 2342–2345, 2004.

[54] Illka Nissinen and Juha Kostomavaara. Time-to-digital converter based on an on-chip voltage reference locked ring oscillator. In *Instrumentaion and Measurement Technology Conference, 2006 IMTC 2006. Proceedings of the IEEE*, pages 250–254, 2006.

[55] A Djemouai, M Sawan, and M Slamani. New circuit techniques based on a high performance frequency-to-voltage converter. In *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on*, volume vol. 1, pages 13–16, 1999.

[56] Roubik Gregorian. *Introduction to CMOS opamps and comparators.* John Wiley & Sons, 1999.