A Baseband, Impulse Ultra-Wideband Transceiver Front-end for Low Power Applications

Ian David O'Donnell

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-47
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-47.html

May 8, 2006

Acknowledgement

This research was supported by the Office of Naval Research (Award No. N00014-00-1-0223), an Army Research Office MURI grant (#065861), and the industrial members of the Berkeley Wireless Research Center. The authors would like to thank Mike S. W. Chen for valuable discussions and assistance with the digital backend design, and Stanley B. T. Wang (U.C. Berkeley) for his work on the transmitter design and layout.

A Baseband, Impulse Ultra-Wideband Transceiver Front-end for Low Power Applications

by

Ian David O'Donnell

B.S. (University of California, Berkeley) 1993
M.S. (University of California, Berkeley) 1996

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Robert W. Brodersen, Chair
Professor Kristofer S. J. Pister
Professor Paul K. Wright

Spring 2006

The dissertation of Ian David O'Donnell is approved:

Chair Date

Date

Date

University of California, Berkeley

Spring 2006

A Baseband, Impulse Ultra-Wideband Transceiver Front-end for Low Power Applications

Ian David O'Donnell

Abstract

A Baseband, Impulse Ultra-Wideband Transceiver Front-end for Low Power Applications

by

Ian David O'Donnell

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Robert W. Brodersen, Chair

Interest in indoor wireless communications has been increasing. In addition to high throughput WLAN systems such as 802.11a/b/g/n, attention is also being focused on lower rate, short distance systems such as Bluetooth and Zigbee. These low rate radios are being proposed for a variety of applications including automation/security, smart toys, remote sensing/control, asset tracking, and as a replacement for computer peripheral wires. While not demanding aggressive throughput, these radios do require low cost, power efficient operation and optionally the ability to perform ranging. Unfortunately, currently reported radios are up to an order of magnitude away from these power and cost targets or do not support ranging. However, a recent ruling from the FCC has opened up nearly $8GHz$ of unlicensed spectrum (from *dc* to $960MHz$ and from $3.1GHz$ to $10.6GHz$) for ultra-wideband (UWB) deployment. One attractive method of UWB signaling that seems suited to a low power, highly integrated implementation communicates with short pulses, on the order of a nanosecond, that spread energy over at least $500MHz$ of bandwidth. Termed "impulse-UWB," the baseband nature of this signaling promises low cost and low power consumption through design simplicity, pulsed (or "duty-cycled") operation, and a "mostly-digital" implementation. The benefits of this approach are balanced by the risk of jamming from in-band interference, of stricter sampling and gain constraints, and of increased digital complexity. This dissertation presents the system exploration, specification, design, and demonstration of a low power, highly integrated, flexible, baseband, impulse ultra-wideband transceiver front-end. Comprising a 1-bit, $1.92Gsample/s$ ADC, 50Ω input matched gain stage with

$0dB$ to $42dB$ of variable gain, programmable control logic, a sub-$1PPM$ trimmable $60MHz$ third-harmonic oscillator, and pulse transmitter, this front-end was implemented in a standard digital $0.13\mu m$ CMOS process in $2.52mm^2$ of active area. Aggressively designed at the circuit level for low power, the front-end gain and sampling are also duty-cycled between pulses to further reduce power consumption, yielding $4mW$ (RX) and $2mW$ (TX) at $30Mpulse/s$, and $0.6mW$ (RX) and $0.4mW$ (TX) at $1Mpulse/s$. Communication rates on the order of $1Mbps$ are supported over short distances and ranging is possible through time-of-flight measurements.

Professor Robert W. Brodersen
Dissertation Committee Chair

Dedicated to my family and friends who supported me during this undertaking.

Contents

List of Figures

List of Tables

Acknowledgments

As with every research project, and especially for one this large, the work could not have been completed without the support and assistance of many others. I apologize in advance if I fail to acknowledge every one who would deserve it, unfortunately time is short and the pressure is on. So, without further ado:

Firstly I would like to thank Professor Robert Brodersen for his years of support and guidance, and for his uncanny knack for steering research directions. I would also like to thank Professor Jan Rabaey and Professor Paul Wright who, with the help of Professor Brodersen, created the Berkeley Wireless Research Center (BWRC), and the InfoPad project before it, and provided a stimulating, energizing, and entertaining research environment. None of this would be possible without all of their effort. This research was also supported by the Office of Naval Research (Award No. N00014-00-1-0223), an Army Research Office MURI grant (#065861), and the industrial members of the Berkeley Wireless Research Center. And I would like to express gratitude to Professor Robert Meyer for agreeing to chair my qualifying exam and Professor Kris Pister for always being excited and interested to hear about my project progress and results. Additionally I would also like to thank Professor Ali Niknejad and Professor Bora Nikolić for their advice and insight, as well as their company at the retreats.

Bob Fleming and Cherie Kushner from Aetherwire & Location also deserve thanks for helping instigate the investigation into impulse ultra-wideband. Their technical expertise, openness and willingness to share is greatly appreciated. I would also like to acknowledge Robert Frye for helping to explore UWB's potential and assistance with writing the first grant proposal in the early days of the project.

No department could operate without a support staff, and we were lucky to have such superb people. Tom Boot deserves thousands of awards for keeping the center running in all aspects above and beyond the call of duty. I would also like to thank Elise Mills for always being helpful and efficient. Ruth Gjerde and Carol Sitea are also wonderful, helpful, and patient. I will miss conversations with Brenda Vanoni and hope time finds her well. We are fortunate to have all of them as part of the team. Finally I wish to thank Brian Richards and Kevin Zimmerman. How they kept the computers and software working, I'll never know! (But I appreciate it!)

I would like to explicitly thank two colleagues: Mike Shuo-Wei Chen and Stanley

Bo-Ting Wang. Mike Chen provided valuable discussions and assistance with the digital backend design[1] for this project and Stanley Wang did as well, and his work on the transmitter design and layout[2] is greatly appreciated.

Speaking of people I am indebted to at the center, I am especially grateful to have known and interacted with such a high caliber of cohort. In particular I would like to thank Johan Vanderhaegen, Ada Poon, Mike Sheets, Josie Ammer, David Sobel, Dennis Yee, Brian Limketkai, Chinh Doan, Sohrab Emami, Patrick McElwee, Tufan Karalar, Sayf Alalusi, Delynn Bettencourt, Mike Chen, and Stanley Wang for all of their friendship, assistance, and feedback over these many years. I will miss their company: discussing research, lunch and Friday beer'o'clock conversations, and their help and advice with complex problems and software environment setup. May the road rise up to meet you and may the wind always be at your back. Additionally, Tufan Karalar, Sayf Alalusi, Dennis Yee, Brian Limketkai, Nate Pletcher, and Delynn Bettencourt deserve a special mention for helping keep me in physical (as well as mental) shape. Whether it be weight lifting, biking, or running or all three. I also value and will miss the contributions from fellow colleagues and visiting scholars at the BWRC: Brian Otis, Chris Rudell, Jeff Weldon, Henry Jen, Dejan Marković, Danijela Čarbić, Chris Taylor, Paul Husted, Jason Musicer, Chris Savarese, Jeff Gilbert, and Andre Vladimirescu. I respect our time and interaction and hope that we meet again down the road. I also feel honored to have met and worked with Marian Verhelst during her visit to the states. Long live Belgium! Finally, I would like to thank Tom Burd, Jim Young, and Roy Sutton who had a similar long, strange road to travel. Here's to the end and what lies beyond. I also would like to thank ST. Microelectronics for chip fabrication and Bhusan Gupta and Engling Yeo for their support in getting this chip taped out.

Working part-time through the Ph.D. gave me a sense of having a double-life, without the excitement, sexiness, and cool toys of international spydom (à la Alias and James Bond). I would like to thank my co-workers at Silicon Graphics (oops, I mean 'sgi') and Nvidia for their wisdom, experience, and camaraderie. In particular, Adriano Jeday, Derek Bosch, Danny Lee, Ferdi Mack, Eric Linstadt, Yeffi Van Atta, Becky Harwell, Lynn Frake, James Tringali, and J. D. Alegrucci were valuable friends and savvy engineers. Let's get lunch to celebrate, on me! I would also like to specially thank my managers at Nvidia for their support and understanding through this process: Joe Greco and Peter Lim. I respect and admire their diligence and experience and hope we have the opportunity to work together again in the future.

I must especially thank my family and friends for their comfort, encouragement, and backing over the duration of this project. I am grateful to my older brother Bob and his wife Jennifer, to my younger brother Pat and his fiancée Sandy, my mother Blair and her husband Joe. Also my grandmother Dotti, her husband Glen, and Don and Lisa as well as my cousins Michelle, Ben, Mara, Josie and John deserve mention. I would also like to include my very good friends Paul Pinarretta, Derek Bosch and Leah Fera. I dedicate this dissertation to them, and to the memory of Stephen and Michael O'Donnell. I would also like thank the long-standing friendship of Caitlin Burke, Morrisa Sherman and Narciso Jaramillo, and specifically Jamie Zawinski, who warned me not to go back into the machine. Additionally I thank those who kept my artistic side inspired: Marco Vangelisti, Brenda Vanoni, Larry Robinson, Kahren Kim, Vace Shakoori, and Morrisa Sherman. Keep painting and creating. May you always find your muse.

Finally, I thank music and musicians. I don't know if I could have done all of this without the 60+ Gigabytes of mp3's (yes, all legally owned and burned by myself) that became the soundtrack to my life. Oh, and a special mention to red wine, without which there could not be civilization. Truly, in vino veritas. A last comment: It is good to see that being briefly exposed to the subject of UWB has not resulted in any discernible harm[3]. I can only hope that longer exposure (without the sedatives) also has no lasting ill effects. (In fact, some may argue that the results of longer exposure have been beneficial. Hopefully this dissertation will inspire future researchers to more experiments).

Chapter 1

Introduction

Inspired by indoor, "picocellular" wireless applications, this dissertation asserts that baseband, impulse ultra-wideband signaling is a viable solution for short distance, low to moderate data rate, low cost, highly integrated, low power radio communication that is capable of ranging. Existing narrowband transceivers are either order(s) of magnitude away from the power and cost targets, or unable to perform ranging (and hence locationing) as accurately as an impulse ultra-wideband approach. Utilizing a "mostly digital" architecture to simplify the analog front-end and ease integration, pulse-based transmission reduces the receiver ADC and transmitter DAC requirements and allows for power savings through duty-cycled operation. The recent FCC approval of nearly $8GHz$ of unlicensed spectrum for ultra-wideband deployment creates an enormous opportunity to research new modes and methods of radio design. The goal of this dissertation is to identify and specify a candidate architecture for these low power, indoor, picocellular wireless applications that may be ultimately demonstrated with a highly integrated, low power implementation in a low cost technology like a generic digital CMOS process. Furthermore, it is the aim of the author to achieve this in a flexible and extensible manner that allows for future experimentation and exploration.

1.1 Motivation

Recently there has been an increased interest in wireless communications especially for indoor, wireless networking. In addition to high throughput local area networks such as 802.11a/b/g, attention is now also being focused on lower rate, indoor communications.

Bluetooth, originally conceived as a replacement for computer peripheral wires, is also being proposed for other applications requiring short distance, low rate communications, such as a wireless intercom or an ubiquitous, wireless bridge to the internet. Many other applications, like home automation/security, smart toys, and remote sensing/control while not demanding aggressive throughput, do require a low cost, fully-integrated radio that can operate for years without battery replacement. Additionally, another class of low cost, low rate wireless applications have emerged. Although they have only moderate communication requirements, they are predicated upon the ability to do ranging or locationing. These applications involve personnel or asset tracking, security, and industrial control, e.g. robotics, anti-collision, and remote sensing/servicing. Currently available commercial radios are in general nowhere near the power or cost levels needed by these applications. While very recent low power results for narrowband radios have been published in research, they do not support ranging.

Such a low rate, low power radio would only need to operate over short distances. This implies less transmit power as well as a lower dynamic range requirement for the receiver. The low bit-rate requirement allows for duty-cycling the radio (i.e. turning off to save power), and implies that spectral efficiency is not terribly important. Ideally, such a radio could operate indefinitely using a single battery or by scavenging power from the environment. As an example, at $1mW$ consumption a standard $1.5V$ AA battery would last for approximately 3,100 hours, or 130 days (almost 4 Months). Duty cycling at 1%, this implies 30 years of operation with a total communication ability of 1 Terabit from a single battery. In addition, the radio needs to be easy to deploy; self-contained and flexible, requiring no special infrastructure or topology for communications. The target transmit distance is approximately 10 meters with an estimated maximum of 32 active users at one time per cell. The anticipated bit-rate is approximately $100kbps$ to $1Mbps$ (uncoded BER $\sim 10^{-3}$) with a total (TX+RX) power budget of $1mW$ for the transceiver when in operation.

1.2 Existing low power radios: Narrowband

To evaluate potential radio candidates for these applications, an appropriate metric is the energy cost per "useful" bit (i.e. a bit transmitted and received without error)[4]. This metric includes header and packet length, and takes into account the communication from a systems-level perspective. Unfortunately this requires knowledge beyond the char-

acteristics of the physical layer. It also depends upon the communication protocol, medium access control (MAC), network statistics, etc. Hence, a simpler approach is taken to gauge the suitability for low rate, low power, indoor applications. An assumption is made that communication will be infrequent (and thus collisions are unlikely to occur) so we may ignore the contributions from the higher layers (i.e. MAC and network). Additionally, we assume packets are long enough to depreciate the header cost. (Note that care must be taken as this assumption is not always accurate, e.g. for sensor networks short packets are common and the header length is an important design parameter[4].) What we examine is the energy cost per transceived bit with the postulate that a radio that scores very poorly is unlikely, when taking into account all of the system parameters, to yield the most efficient solution.

Current state-of-the-art narrowband radios are shown plotted in figure 1.1 along with two UWB radio operation targets (for high and low data rates). The plot shows receiver power plotted against data throughput. (In this case transmitter power is neglected to provide a fairer comparison since the required transmit distances are short, and most of these radios target much longer range (i.e. higher power) transmissions.) Dashed lines indicate the power/throughput trajectory if perfect duty-cycling were available; that is, one could simply get $1/N$ the power for $1/N$ the data rate from a given radio. The duty-cycling trajectory is shown to illustrate how well a higher data rate radio (i.e. 802.11a) would fare if it were scaled into this application space. Data for Bluetooth, 802.11a and 802.11b, Zigbee, ISM-band commercial, and research radios are included in the plot, and listed in tables 1.1 and 1.2. Note that this is a naïve plot for this class of radios; it does not correct for different operational frequencies, different standards, or equalize to equivalent input sensitivities. (This would be a difficult, but interesting task worthy of examination and speaks towards the question of efficient implementation of narrowband radios. While most results for narrowband radios perform much poorer than the application space requires, recent research has reported power efficiency at the desired target of operation [5]. This indicates that higher power consumption is not innate to narrowband reception. However, the issue of ranging with narrowband radios would also have to be analyzed.)

An interesting observation from this plot is that almost all of the available radios function at up to order of magnitude less than the target efficiency of $1 Mbps/mW$.

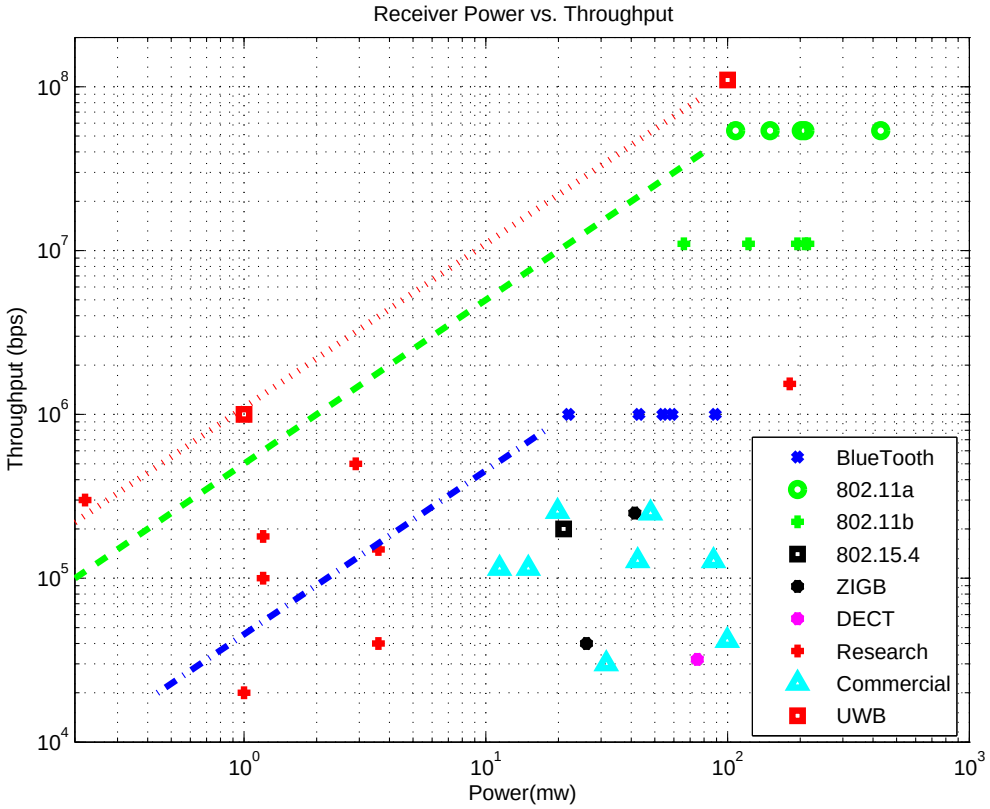

Figure 1.1: Receiver Power vs. Throughput

Table 1.1: Receiver Power vs. Throughput: Part 1

Power Consumption	**Throughput**	**Description**	**Reference**
108mW	54000 kbps	802.11a	[6]
108mW	54000 kbps	802.11a	[7]
202mW	54000 kbps	802.11a	[8]
209mW	54000 kbps	802.11a	[9]
150mW	54000 kbps	802.11a	[10]
430mW	54000 kbps	802.11a	[11]
66mW	11000 kbps	802.11b	[12]
122mW	11000 kbps	802.11b	[6]
209mW	11000 kbps	802.11b	[9]
194mW	11000 kbps	802.11b	[8]
215mW	11000 kbps	802.11b	[13]
59mW	1000 kbps	Bluetooth	[14]
54mW	1000 kbps	Bluetooth	[15]
43mW	1000 kbps	Bluetooth	[16]
89mW	1000 kbps	Bluetooth	[17]
22mW	1000 kbps	Bluetooth	[18]
21mW	200 kbps	802.15.4	[27]
26mW	40 kbps	802.15.4 / ZigBee	[28]
39.4mW	250 kbps	802.15.4 / ZigBee	[29]
48mW	250 kbps	802.15.4 / ZigBee	[30]

Table 1.2: Receiver Power vs. Throughput: Part 2

Power Consumption	**Throughput**	**Description**	**Reference**
35mW	32 kbps	DECT	[19]
11mW	115 kbps	Commercial ISM	[20]
15mW	115 kbps	Commercial ISM	[21]
42mW	128 kbps	Commercial ISM	[22]
87mW	128 kbps	Commercial ISM	[23]
100mW	42 kbps	Commercial ISM	[24]
31mW	30 kbps	Commercial ISM	[25]
20mW	256 kbps	Commercial ISM	[26]
0.22mW	300 kbps	Research	[5]
2.9mW	500 kbps	Research	[31]
0.4mW	5 kbps	Research	[32]
3.6mW	40 kbps	Research	[33]
181mW	1536 kbps	Research	[34]
1.0 mW	20 kbps	Research	[35] [36]
1.2mW	180 kbps	Research	[37]
3.6mW	150 kbps	Research	[38]
1.2mW	100 kbps	Research	[39]
100mW	110000 kbps	Ultra-Wideband	802.15.3a Target
1mW	1000 kbps	Ultra-Wideband	Our Target

An interesting exception is 802.11a (and to some extent 802.11b) which display admirable throughput/power efficiency. Unfortunately their data rates are much larger then needed, and they are considerably larger (die area) and more costly than the low power, moderate data rate applications being examined. Duty-cycling to save power consumption is not possible even though 802.11a acquires on each packet header, as the standby power consumption would dominate and greatly reduce the efficiency per bit. However, if the cost and standby power problems could be solved, the duty-cycled trajectory indicates that 802.11a could be an appealing approach.

While Bluetooth ($2.4GHz$ FHSS), and RF Monolithic transceivers ($433MHz$, $916MHz$, $2.4kbps$ OOK, $115kbps$ ASK) are more specifically targeted to the low power, moderate data rate application space, they consume too much power relative to the throughput. (And in Bluetooth's case, they may wind up costing too much as well.) Although no viable commercial products exist yet, promising research results have been published recently. The lowest power, most integrated, conventional narrowband radio to date is a frequency shift keyed (FSK) transceiver [35] [36], which approaches a $1mW$ receive power at $20kbps$ with minimal external parts, but unfortunately utilizes a $21mW$ transmitter. This is presumably for longer range transmissions, but also shows a characteristic of narrowband transceivers – namely that they tend to be transmit-power dominated. The reason for this is that narrowband transmitters often require linear amplifiers, which are inherently less power efficient, to maintain power spectral density shape. Also, narrowband link budgets often add margin to the transmitter to overcome the probability of deep fading. The combination of added margin and poorer amplification efficiency can increase the worst-case

power consumption to orders or magnitude above the average necessary power consumption. For short transmit distances this effect is tempered, but it indicates a potential weakness for that architecture.

The application characteristics of low rate, short distance, indoor communication ease the burden of design (such as lower transmit power and relaxed sensitivity). Until very recently, it was unclear how far traditional, sinusoidal-based architectures will scale or whether they present the best approach for achieving a low power, highly integrated solution. However, [5] reports admirable performance of $300kbps$ in $220\mu W$. In fact, these results are exactly at the desired level for sensor network applications. This indicates that narrowband radios are able to achieve the communication power and performance targets. Other promising results have also been published for non-traditional narrowband architectures. For example [39] reports a $100kbps$, $2.4mW$ a narrowband super-regenerative receiver which, while suffering poor channel selectivity, comes closer to the power and cost goal. [31] also employs a super-regenerative architecture to achieve $500kbps$ in $2.9mW$. The super-regenerative approach uses nonlinear modulation and nearly simplifies the radio down to just the oscillator. However, in these cases, a narrowband transmitter would still have the same tendency towards larger power consumption.

It is also worth noting that digital processing may be used to assist in removing or relaxing constraints on analog blocks[40], thus allowing a reduction in an overall system power consumption. One suggested alternative approach to the traditional sinusoidal radio architecture is the "mostly-digital" approach of moving the digital logic as close to the antenna as possible. The anticipated power savings, over a traditional sinusoidal-based

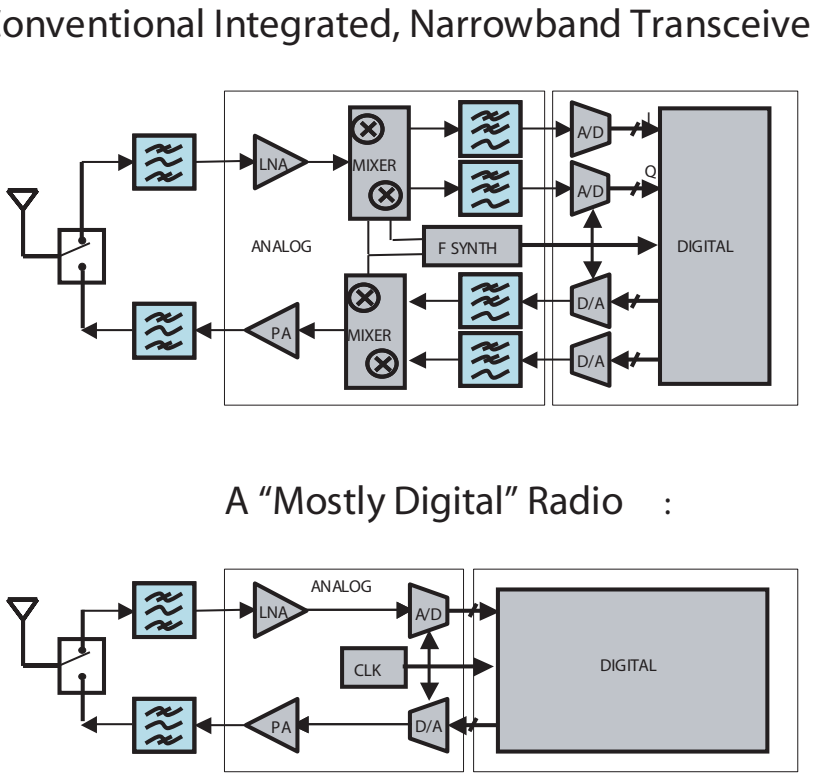

Figure 1.2: Comparison of Narrowband and "Mostly-Digital" Architectures

transceiver, come from the elimination of frequency translation and synthesis, removal of filtering and reduction of external components (as shown in figure 1.2). However, the primary problem with this architecture for narrowband systems is a drastic increase in the difficulty of analog to digital (A/D) and digital to analog (D/A) conversion. The receiver A/D must sample an RF input, requiring high precision to handle the incoming dynamic range and difficult sampling switch design and clocking requirements. Likewise, the transmit D/A operates at the RF frequency and must have precision high enough to keep within the transmit spectral mask.

The problems with a "mostly digital" architecture are a largely a function of

narrowband transmission. If we consider using something other than modulated sinusoids to communicate, such as a pulse, we may overcome the limitations of both the traditional sinusoidal systems and take fuller advantage of the digital approach. This architecture is essentially carrier-less, communicating at baseband by sending out very short (on the order of nano-second) pulses. The nature of this radio lends itself to a digital implementation and promises power reduction through lower supply voltages and scaled geometries, as well as a more efficient and straight-forward transmitter design. However, the use of impulse signaling with a "mostly digital" approach, while easing some problems, moves the design challenge to different dimensions. Pulses occupy a large amount of bandwidth and will overlap with pre-existing narrowband users. In particular, the ADC speed and resolution become of utmost importance. Baseband Nyquist sampling of the larger bandwidth requires ADC clocking on the order of a GHz which has the potential to consume enormous amounts of power relative to our $1mW$ target. Also, the impact of including large in-band, sinusoidal interferers must be determined. Additionally, the sub-nanosecond timing for ADC sampling may place severe limitations on oscillator matching or jitter requirements. The power consumed in the wideband front-end gain stages and the necessary sensitivity and gain requirements for those blocks are also important. Finally, the area and power burden required for digital signal processing and demodulation must be considered.

1.3 Alternate Modulation: Ultra-Wideband (UWB)

The recent approval of approximately $8GHz$ of unlicensed spectrum in the U.S.A. (from dc to $960MHz$ and from $3.1GHz$ to $10.6GHz$) for ultra-wideband deployment presents

an interesting research opportunity. The lack of specified physical layer signaling and low transmit power levels for UWB similar to Part 15[41] have opened a wide design space with a large possibility for innovation. While the majority of attention is focussed on high-speed communication applications in the $3.1GHz$ to $10.6GHz$ band over very short distances ($\sim 1m$ due to the transmit power density constraints), there is also interest in power efficient ranging, imaging, and distance measurements with communication at relatively low data rates. One attractive method of ultra-wideband signaling suitable for low power operation uses short pulses, on the order of nanoseconds, to spread energy over at least $500MHz$ of bandwidth. The baseband-like nature of this signaling promises a low cost, low power architecture because of the simplified, low-Q analog front-end design. Using a "mostly-digital" architecture, this radio attempts to bring the digital logic as close to the antenna as possible, thereby reducing analog complexity and power, and increasing integration. Further power savings are possible through circuit operation duty-cycling between pulse reception windows.

Due to historical development, the spectrum is more heavily used in the lower frequencies; primarily frequencies less than $1GHz$. In spite of this crowding, this spectrum is desirable as it exhibits good material penetration and longer transmit distances. Additionally lower frequencies are easier for design and imply lower power operation. Unfortunately, the benefit of lower frequencies (longer wavelengths) also come at the cost of larger passive devices (which are harder to integrate on-chip), larger antennas, and interference from the aforementioned pre-existing spectrum users.

Ultra-wideband signaling is illustrated in figure 1.3 which compares the time and

frequency domain signatures of two UWB approaches with narrowband, sinusoidal modulation. Ultra-wideband operation may be obtained by either scaling narrowband approaches (as is done for OFDM UWB), using pulses, or other techniques. Impulse signaling seems like an advantageous solution – amenable to a "mostly-digital" implementation, and without the narrowband transmit problems of low efficiency linear amplification and deep fades (deep fading is fundamentally a narrowband phenomenon). Pulse generation is also suited to a fast digital logic implementation. As will be shown, the prior concerns of A/D resolution and clocking, narrowband interference, wideband gain, and the digital processing burden will find favorable exposition, and impulse-UWB will be demonstrated as a viable low power radio candidate.

1.4 Existing UWB radios

UWB signaling has traditionally been employed in the domain of radar systems and extremely high rate data networks [42]. For radar, the wide and shallow frequency spectrum of the signal allows for good penetration, multipath immunity and low probability of intercept. Most importantly, the ultra-fine timing resolution allows for locationing/positioning. For high rate data networks, the value lies in the sheer speed of communication. Current UWB commercial systems tend to be rather bulky and expensive as well as power hungry due to accurate clock requirements, high voltage pulse generation, long distance operation, and relatively low levels of integration. Recently, interest has been growing in adopting and scaling UWB technology towards low rate locationing problems (e.g. RF tags) and high rate, short distance communication (i.e. "the last mile" broadband connectivity to

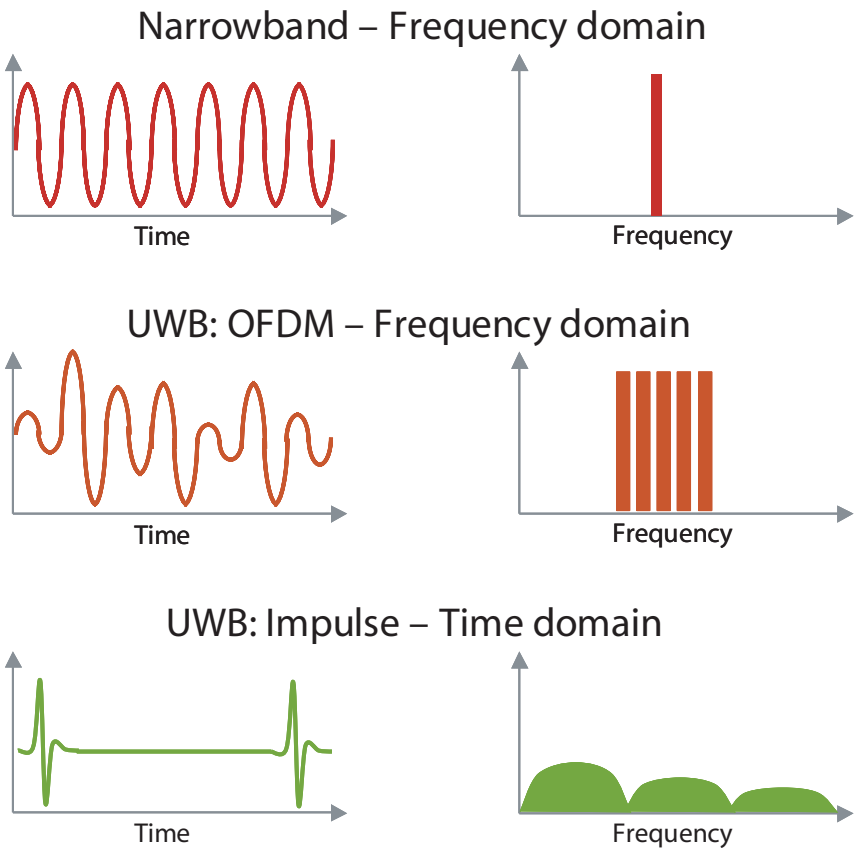

Figure 1.3: Narrowband vs. Ultra-Wideband in Time and Frequency

the home). In addition, the FCC has revised the part 15 rule on maximum, unintentional radiation limits, allowing for low power UWB emissions. However, present designs are relatively power hungry (on the order of Watts), moderately integrated, and not specifically focused on low rate, wireless network communications.

Earlier approaches to adapting UWB were based on different analog architectures where the pulse correlation is performed in the analog domain before A/D conversion [43] [44] [45] [46] [47]. However, none of these architectures specifically targeted milliwatt power levels, and other than [45], they tend to offer scant details on actual circuit implementation. Very recently, two low power, analog-based architectures have appeared in the literature [48] [49]. [48] realizes $1Mbps$ and $\pm 2.5cm$ ranging over a $1m$ distance with $4.0mW$ receiver and $0.7mW$ transmitter power consumption using a simple, single clocked-correlator circuit. Another analog-based architecture[49] which simply thresholds the incoming pulse stream to get a digital output boasts very low power consumption ($299\mu W$) and a trivial implementation. This radio is confined to operate only in very short distance, high-SNR environments ($< 35cm$ operation claimed at $25kbps$) and does not scale well. The main disagreement between analog vs. digital architectures condenses to an argument of where to put the A/D conversion relative to the signal processing (i.e. template filtering). The main arguments for a digital approach to the signal processing are: 1. digital scales more easily for parallel searching (reducing acquisition time and hence packet overhead); 2. digital promises improved performance in the presence of multipath and heavy interference, allowing for channel estimation, interference cancellation, etc. that would be more costly to perform in analog; and 3. the inherent benefits (robustness, CAD support, flexibility,

scalability, ease of design, complexity, etc.) of using digital signal processing.

Recent UWB digital architecture publications are split between a channelized, or frequency-based (eigen value), approach [50] [51] [52] [53] and direct time-based sampling of the UWB signal [54] and [55]. The digital frequency-based approaches, while predicting good results, seem overly complex for the relatively low data rates (and power consumption) and high levels of integration (low cost) needed by a sensor network application. These architectures seem better suited to high-speed communication links. The time-based architecture found in [55], intended as a baseband system for a $3.1GHz$ to $10.6GHz$ communication link, is similar to what is proposed here[54] (and [56]), but the reported power consumption is far too high. The target power consumption of $1mW$ (TX+RX) in this dissertation is over $50\times$ lower than the $200kbps$ front-end in [55] (antenna matching and gain) and [57] (4-bit A/D, clock generation and digital backend).

1.5 UWB Regulations and Standards

In 2002 the FCC issued a report detailing the regulations for ultra-wideband emissions. Focusing on bandwidth and power constraints, the issue of physical signaling was left open to encourage innovation and discussion. Since 2002, there have been two commercial attempts to standardize a UWB physical layer. One targeting high speed operation, 802.15.3a, has been abandoned due to disagreement about the choice of physical signaling. The other, 802.15.4a, targeting ultra-low power operation with a moderate communication rate and ranging ability, is still on schedule for completion.

1.5.1 FCC Regulations

With the release of the First Report and Order[41], the FCC set the world's first regulatory guidelines for the commercial use of UWB technology beyond RADAR or the military. UWB operating restrictions are grouped into three categories based on the potential for interference generated by these applications: imaging systems (such as ground-penetrating RADAR, through-wall imaging, medical and surveillance systems); vehicular RADAR systems; and communication and measurement systems. The regulation specifies the minimum signal bandwidth (measured at $-10dB$ points) of $500MHz$ or $0.2 * f_{carrier}$. The peak to average ratio is constrained to a $20dB$ maximum and the spectrum is divided into roughly two bands: dc to $960MHz$ and $3.1GHz$ to $10.6GHz$. The power levels are limited to a 3 meter equivalent isotropic radiated power (EIRP) that is similar to part 15[58]. The aggregate power available is small – approximately $-20dBm$ over dc to $960MHz$ and approximately $-2dBm$ over $3.1GHz$ to $10.6GHz$. Table 1.3 shows the allowable emission limits for low frequency imaging systems from section 15.509. Power levels at or below $960MHz$ for this class of systems are regulated as per section 15.209.

Conceived as an overlay technology, UWB allows for short-range, possibly high rate communication to coexist with previous spectrum users without causing undue damage to those users' services. The low power levels compel short distance use, although the abundance of available bandwidth implies that very high throughput (larger than $1Gbps$) is possible. This profusion of bandwidth can be traded for longer distance, lower data rate operation.

Interestingly, while the power spectral density is regulated, the specific physical

Table 1.3: FCC Part 15.509 EIRP Levels

Frequency	EIRP in dBm/MHz
$< 30MHz$	-56.2
$30MHz$ **to** $88MHz$	-55.8
$88MHz$ **to** $216MHz$	-52.2
$216MHz$ **to** $960MHz$	-49.7
$960MHz$ **to** $1.61GHz$	-65.3
$1.61GHz$ **to** $1.99GHz$	-53.3
$1.99GHz$ **to** $10GHz$	-51.3

signaling is not. This implies that any waveform shape that meets the bandwidth and power specifications may be considered ultra-wideband. For example, a chirp or pulse may be used as well as broadband noise (i.e. from a chaotic system) or even OFDM (simply applied over a larger bandwidth, i.e. $> 500MHz$). This freedom dramatically expands the design space for UWB systems, leaving open unexplored avenues of innovation and research.

1.5.2 802.15.3a and 802.15.4a

Two attempts at standardizing UWB have occurred since the initial First Report and Order from the FCC. 802.15.3a, now officially disbanded, targeted high-data rate, short distance consumer applications. Aiming to replace wires (like USB) that interconnect computer peripherals, 802.15.3a proposed three data rates: $110Mbps$ at $10m$ in $100mW$, $200Mbps$ at $4m$ in $250mW$, and $480Mbps$ up to a meter or so, at less than $500mW$. Unfortunately, the standardization process became mired by in-fighting between two different coalitions of companies, each advancing a particular style of UWB physical layer signaling: multi-band OFDM (MB-OFDM) and direct sequence impulse-UWB (DS-UWB). These signaling approaches were incompatible, and the standardization attempt was eventually abandoned when it became apparent there would be no resolution to this disagreement. Each group has vowed to commercialize their solution and let the customers decide. As of this writing, chips are available for both solutions, but products are not yet on the market.

802.15.4a was proposed as an alternate physical layer for the 802.15.4 wireless personal area network (WPAN) standard. The principal interest is in ultra-low power communication and precise ranging with scalable data rate/range functionality. The physical layer (PHY) specification is not complete as of this writing, but includes a pulse-based

ultra-wideband signaling approach. The technical requirements for this PHY are: data rates from $1kbps$ to up $1Mbps$ (for "aggregators"), range from 0 to 30 meters, power consumption low enough to last for months or years, precise ranging (location aware from 10's cm to 1m), dynamic networking, non-directional antenna, a form-factor appropriate for sensor network/RF tags, potential for motion tracking, and robust operation. The standard is expected to be finalized in 2006.

1.6 Opportunity

There are a number of futuristic applications that make use of the unique capabilities of UWB. Examples include smart RFID chips that know their location, imaging through walls, sensor networks, and the ability to track expensive assets or even people. Many novel architectures and systems have been proposed, and the investigation into the subject of ultra-wideband continues to expand. One technique in particular, impulse-based ultra-wideband, offers an exciting research opportunity for very low power and highly integrated CMOS radio implementations. By using a moderate pulse transmission rate and taking advantage of the nature of impulse-UWB, significant power savings may be realized in a transceiver that is capable of ranging as well communicating.

At first glance, ultra-wideband may not seem a likely candidate for a low rate, low power, highly integrated radio; however, the overall architecture scales nicely into this domain. Short distances no longer require high-power, high-voltage pulses, and the relaxed sensitivity requirements allow for a more noise-tolerant, more easily integratable receiver. A simpler transceiver architecture with a minimum number of analog blocks leverages the

capability of digital design and removes several traditionally power hungry circuits. By taking advantage of the nature of impulse-based signaling, namely that the received energy tends to be concentrated in time, we can lower power consumption on the receiver by sampling only during that window in time where the energy is concentrated, while turning off the radio between pulses. This has the additional benefit of separating the analog sampling and digital processing in time, thereby reducing the amount of digital noise coupling into sensitive circuits. Power may be further reduced by running the system only at the slower baseband pulse repetition rate (typically less than $30MHz$) and by dividing the sampling operation into parallel paths. To save additional power, aid integration, and allow for flexibility, an almost completely "digital" architecture is employed; sampling the signal close to the antenna and processing the data in the digital domain.

An UWB approach is not without problems, however. Concerns exist about high rate A/D sampling and resolution, the consequence of narrowband interference, wideband circuit power consumption, and the backend digital processing. As will be shown in this dissertation, these issues, though important, are not fatal. For example, the energy and amplitude of an impulse are concentrated in time and hence more easily discerned within heavy interference than a simple comparison of signal energies might suggest. Adding a spreading code and digital processing further helps increase the received SNR. System simulation results indicate that the A/D bit-width is reasonably realized in low power CMOS circuits. Integration into a deep submicron CMOS process also eases wideband circuit power consumption due to the inherent speed and lower parasitics of the process technology. In addition, analysis reveals that synchronization may be maintained with

moderately stable and accurate crystals, and clock generation may be implemented simply with variable, on-chip delay lines.

1.7 Outline of Dissertation

The goal of this research is the exploration, design, implementation, and demonstration of a highly integrated, low power, impulse-UWB transceiver for a low rate, indoor network. This research focuses on the complete transceiver and will attempt to quantify the trade-offs between system performance and implementation. The maximum power target for communication (the sum of the transmit and receive powers) is set at $1mW$ and $1Mbps$. $0.13\mu m$ CMOS was chosen for implementation as it allows for the potential of full integration of the digital logic with the analog processing blocks. The transceiver front-end does not require any external components beyond a bias resistor and external crystal circuit for clock generation. In order to achieve a single-chip solution with low power consumption, the following approach is proposed:

- Investigation of ultra-wideband communications and determination of feasibility.
- Identification and modeling of the proposed architecture and establishment of selection criteria for power efficient implementations.
- Identification of low power design techniques for integrated circuits implemented in a low cost, generic CMOS process.

The remainder of this dissertation is divided into three chapters. The first explores the system-level requirements and specifications for designing a new "mostly-digital"

impulse-UWB radio. The second details the circuit-level design and power/performance optimization along with measured results for each circuit block. The final chapter demonstrates the radio functionality and presents measured results from radio operation and duty-cycled power consumption. The appendices contain a complete description of the programmable on-chip registers, a description of the testboard design and setup, and a pin list for the chip.

Chapter 2

System

The proposed operation for a low power, impulse-UWB transceiver is shown in figure 2.1. Conceptually, a pulse of short duration is sampled at a high rate, and then the analog circuitry is shut down and the result is processed digitally afterward. Power savings is expected from the duty-cycled nature of both pulse generation and the receiver operation. Also, self-interference is avoided by separating the pulse reception and processing in time. To realize the promise of low power, impulse-UWB a journey must be taken from this conceptual operation to a transceiver architecture specification. In particular, the system requirements and performance trade-offs must be mapped to specifications.

As impulse-UWB is relatively unexplored terrain, the potential design space is enormous. Additionally, not all of the more commonly taught narrowband techniques can be simply applied. For example, impulse-UWB is not a simple frequency-swept extension from narrowband. Transient phenomena such as reflection at an impedance boundary may occur which would be missed with a narrowband, steady-state analysis. Additionally,

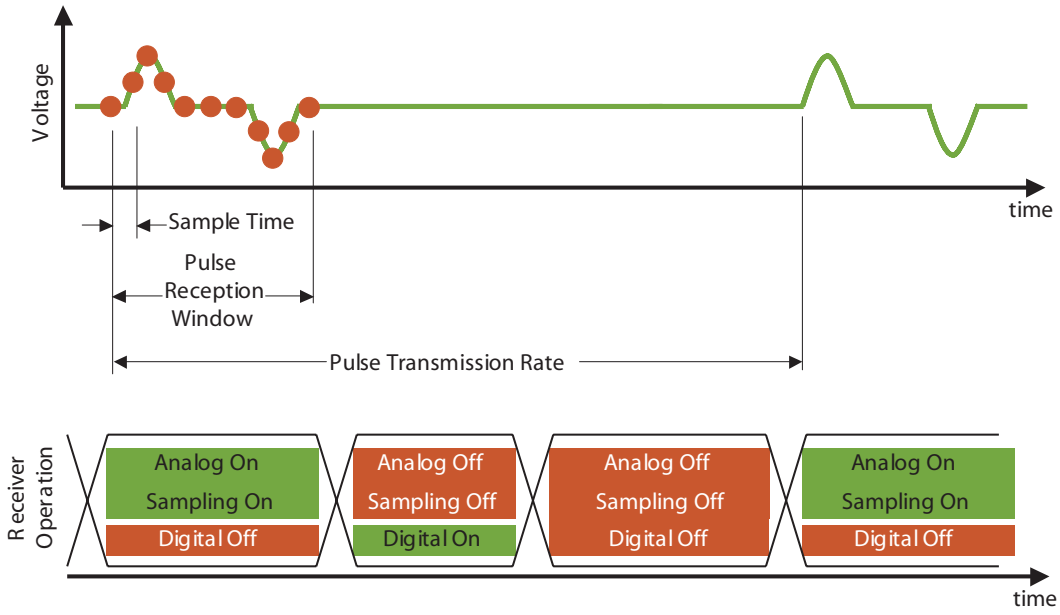

Figure 2.1: Impulse Sampling Operation

certain narrowband terminology like "sensitivity," "phase," or "carrier frequency" become ambiguous or misleading. (E.g. what is the "phase" of a pulse?) Instead, it will be necessary to investigate each aspect of the transceiver chain and to develop an appropriate ultra-wideband transmission model. Figure 2.2 shows a macro block level of such a model. Pulse generation occurs and then travels through an ultra-wideband channel where noise and interference are added. The receiver amplifies the sum of signal, noise and interference, provides filtering and then quantizes the signal before performing signal processing to extract the signal.

In the following sections we will examine the constituent parts of an ultra-wideband transceiver: pulse generation, antenna choice, interference, the UWB channel, and receiver signal processing options. A system framework will be developed including both analytical

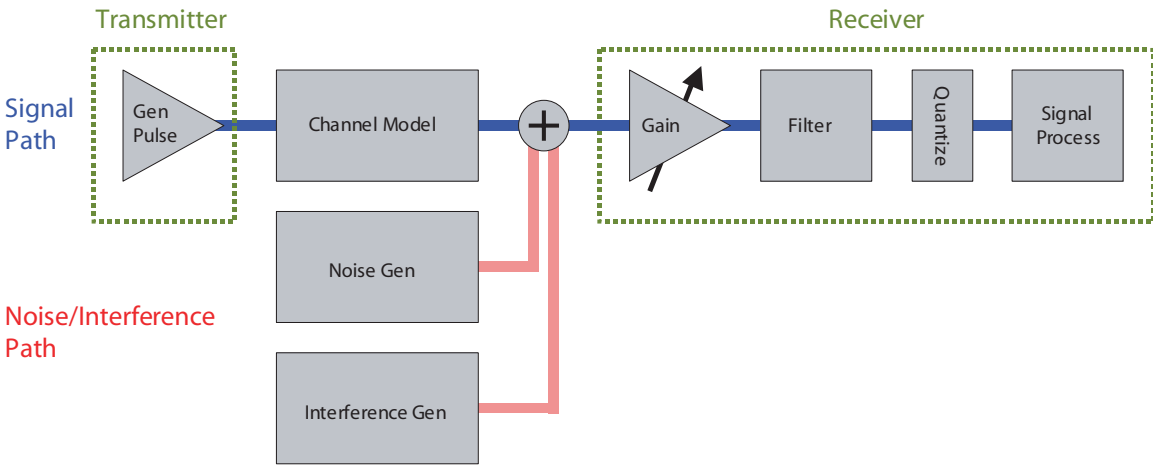

Figure 2.2: Transceiver System Model

and simulation modeling, allowing for performance trade-off evaluation under different scenarios [59]. From this, ADC resolution, template filter length and resolution, noise figure, gain, offset, modulation, and filtering requirements will be determined. Additionally, sampling clock matching and jitter bounds will be derived. The end result will be a complete system specification for a low, power, baseband, impulse-UWB transceiver.

2.1 Pulse Generation

Existing UWB pulse transmitters may be found in [60] [61] [42] and [62]. However, these pulse generators were designed for older RADAR technology and are often not easily integratable into submicron CMOS: e.g. high-voltage capacitive discharge, step-recovery diodes, spark-gaps, gas tubes, and transmission lines. One very promising approach uses fast switches, an attribute of submicron CMOS, for pulse generation. Submicron CMOS cannot support high voltages, but FCC mandated power levels are low, this is not a prob-

lem. The specific pulse shape generated is not necessarily important itself, rather, satisfying the FCC spectral mask and limiting the pulse duration (to maximize the SNR over that interval) are important. This may be achieved through pulse generation itself, the transmitter/antenna combination, or by co-design including a filtering circuit between transmitter and antenna[2]. Generally, ringing should be avoided because it reduces pulse bandwidth and increases the peak power spectral density, resulting in poorer FCC mask filling.

The system simulation may be setup to generate any variety of input pulse. Figure 2.3 shows three pulse shapes that commonly occur in the literature. In addition, ideal (dirac) impulses, ideal pulses, finite-edge-rate pulses, Manchester pulses, step responses, triangular pulses, etc. may be generated. Many pulses perform similarly in simulation if they have roughly equivalent energy over the same duration.

2.2 Antennas

Characterization of antennas is an important part of system modeling. While narrowband antennas are generally understood and simpler to model, broadband antennas may vary drastically in their impulse responses and utility for ultra-wideband applications. Understanding antenna properties and their consequences, at least from an abstract level, is necessary to obtain reliable knowledge about system performance. It is beyond the scope of this thesis to delve into the electromagnetics of antenna design; instead, desirable antenna behavior and possible UWB candidates are identified.

Functionally, the antenna exists to efficiently couple energy to the atmosphere. There can be a complex interaction between the driving source (or receiver) impedance and

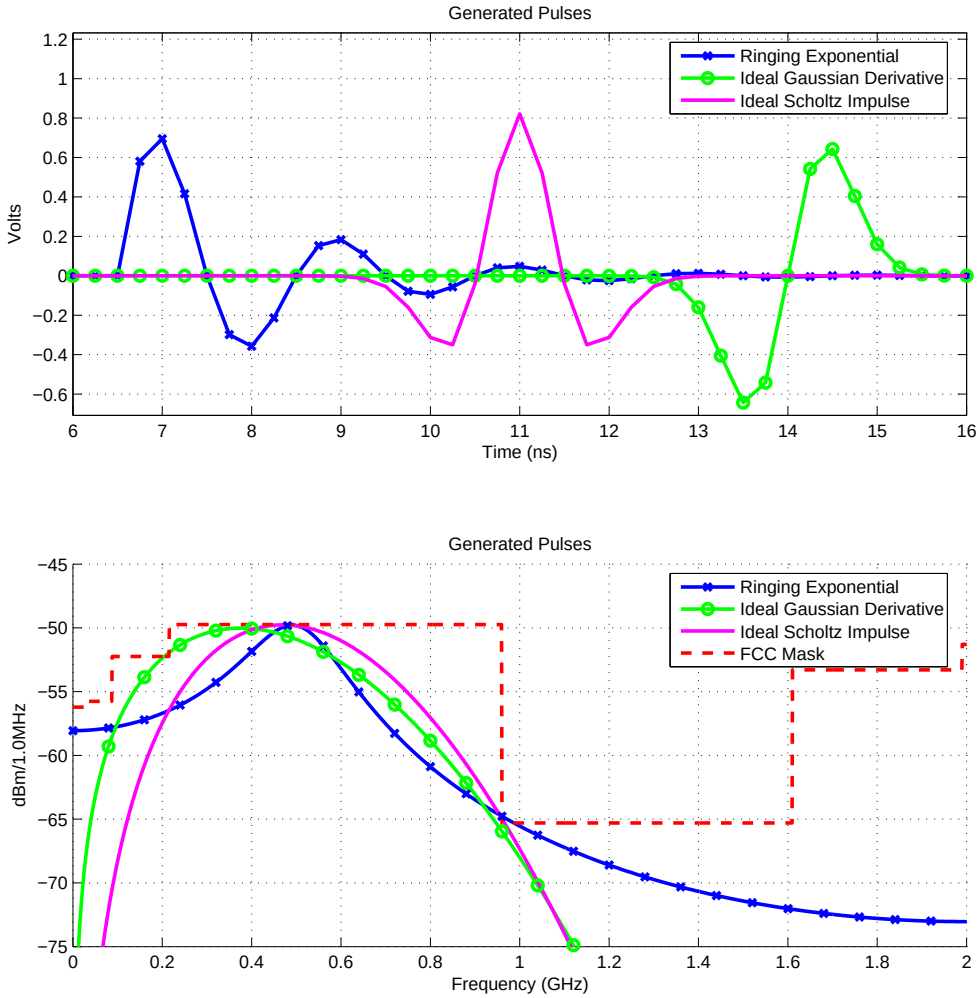

Figure 2.3: Example Pulses in Time and Frequency

the optimal coupling impedance for the antenna. For narrowband systems, this interaction can usually be simplified into a single phasor (complex number) for analysis. Wideband antennas may sometimes be analyzed by varying the phasor as a function of frequency, but this presumes a steady-state sinusoidal environment. For impulsive (or short time duration) inputs, impedance mismatches between the driver and antenna may result in transient reflections before settling to a steady state. Using a phasor analysis will overlook these transients and may give misleading results. From that perspective, it is better to operate in the time domain, as opposed to the frequency domain, to capture wideband behavior in antenna excitation. Many narrowband antennas are essentially resonant in the time domain which allows for a steady-state analysis. Wideband antennas are treated more similarly to traveling wave guides.

Many common antenna parameters such as radiation resistance and VSWR (Voltage Standing Wave Ratio) are inherently narrowband and therefore not useful for wideband characterization.[42] Antenna gain (directivity) also may vary as a function frequency and must be approached carefully, since it is an important application-dependent parameter. For our sensor-network application, an omnidirectional pattern would be ideal, because it allows arbitrary placement of the radio without link degradation. Although a highly scattering environment may allow more directive antennas to obtain fuller coverage as well. This issue may also be approached from the network layer in a densely populated sensor network as the probability that every radio will at least see some number of neighbors increases and thus connectivity may be achieved by routing packets through neighbors.

Another important parameter for a wideband antenna is that it has a constant

phase center as a function of frequency. This prevents dispersion (which can smear a pulse over time). From a design perspective, it would also simplify matters if the wideband antenna had a relative constant input impedance over frequency. (Although, co-design between the circuit drivers and receivers and the antenna may be performed to compensate for this[2].) An example of a wideband, dispersive antenna is the common log periodic dipole array (LPDA) used for television reception. By arranging a sequence of scaled dipoles spaced out along an armature, a frequency range from $100MHz$ (VHF) to $800MHz$ (UHF) may be spanned. A pulse input would be spread in time, though, as the phase center varies in frequency along the armature.

References [42] and [63] are good introductions to antenna requirements for wideband systems. From that source, several viable wideband antenna candidates were identified: loaded dipole or loop, biconical or horn (although very directive), and large current radiator (Hertzian dipole). For a sensor network application, a small form factor antenna would be ideal. Unfortunately, antenna size is directly related to wavelength for efficient radiation. At $1GHz$, the free space wavelength is approximately $1foot$ ($0.3m$). This can be reduced somewhat with a higher dielectric, but the size is roughly on this order of magnitude. This represents an unchangeable consequence of operating in the $100MHz$ to $1GHz$ band. Physically smaller (and electrically smaller) antennas may be employed at the expense of radiation efficiency, as is analyzed in [2]. Because the allowed radiated power limits from the FCC are so low ($-10dBm$ total power over dc to $960MHz$), it is possible to compensate for these less efficient antennas through an increase in transmit power without adversely increasing the overall system power and performance. (This allows for roughly

another factor of 10 before power transmit consumption surpasses the milliwatt target.) It seems likely, though, that a good compromise between antenna size and efficiency can be found.

Antenna output power can be increased by either increasing the applied voltage or current as appropriate. Because integration into a $0.13\mu m$ CMOS process is desired, it is easier to increase current than voltage. This pushes the antenna choice towards a loop or large current radiator structure which are called "current-mode" antennas (e.g. as opposed to a dipole which is called "voltage-mode" antenna). An example UWB system using the large current radiator can be seen in [45].

Once a suitable antenna candidate has been selected, it is possible to be agnostic about the actual pulse shape from the system simulation perspective. What is important is that it is bounded in time (and therefore the energy is concentrated in that interval). The channel environment, antenna orientation and response, and generated input pulse are not known a priori. However, all of these items can be lumped together into a generic channel response. (In fact, we can also include the front-end circuit nonlinearities.) We assume that the aggregate pulse shape is knowable, in that it may be measured or simulated (or estimated), and proceed from there.

2.3 Interference

Interference (caused to an UWB system by pre-existing users of the spectrum or by other nearby UWB transmitters) will be significant given the anticipated weak power levels at the receiver. In this dissertation, we will only consider interference to the receiver.

Interference caused by UWB transmission to other pre-existing radio systems is a large topic itself but is already regulated through the FCC Report and Order. Numerous documents regarding the interference caused by UWB (measured and theoretical) are available through the FCC website. An interested reader is encouraged to explore this resource for more information.

The designer needs to have a model of incoming interference for an UWB system. As UWB transmit power levels are low, the dominant source of interference is expected to be in-band, pre-existing narrowband transmitters. Out-of-band interference may be attenuated through filtering, and CDMA or TDMA schemes may be employed if the density of active UWB transmitters were high enough to cause self-interference. (The target application, wireless sensor networks, assumes relatively low rates of communication so this scenario is considered unlikely.) The frequency band from dc to $1GHz$ is the most crowded section of spectrum and compromises many potentially powerful or nearby transmitters. Some of the more common interferers are listed in table 2.1. If a single transmitter dominates the interference, it can usually be modeled as a simple sinusoid of equivalent power because these sources are all narrowband. If the interference consists of the summation of a large number of randomly phased and similarly powered narrowband transmitters, the result, due to the central limit theorem [64], is Gaussian, and may be modeled as an equivalent-power increase in the noise floor. However, if the interference is a mix of these two conditions, modeling may consist of a combination of gaussian noise on one (or several) larger sinusoids.

To gain a handle on the interference environment, time-domain measurements were taken with a TEM horn capable of resolving from $1GHz$ to below $100MHz$ in the

Table 2.1: Some Common Interferers

Frequency	**Interferer**
$88MHz - 108MHz$	FM Radio
$54MHz - 88MHz$	TV VHF (chan. 2 - 6)
$174MHz - 216MHz$	TV VHF (chan. 7 - 13)
$470MHz - 806MHz$	TV UHF (chan. 14 - 69)
$157MHz$, $452MHz$, $457MHz$	Taxi
$154 - 6MHz$, $158 - 9MHz$, $460MHz$, $465MHz$	Police
$462MHz$, $467MHz$	GMRS
$902 - 28MHz$	ISM Band
$824 - 49MHz$, $870 - 93MHz$	Cell Phone
$929 - 30MHz$	Pager

laboratory using a $20Gsample/s$ oscilloscope. Figure 2.4 shows the power spectral density from several of these measurements. In this plot, interference from cell phones, television, and taxi/police radios may be clearly seen. In general, the power of a particular interferer was found to be less than $-50dBm$; however, power from cell phone communication may be $-30dBm$ or larger. Figure 2.5 shows the time domain plots (over 50Ω) of the worst and best cases of measured interference. A histogram of the minimum interference case is a good match to a gaussian with a standard deviation of $2mV$. The maximum interference case, dominated by a few large interferers, appears more bi-modal with peaks at $\pm 7mV$. The standard deviation is $9mV$ and does not seem a good fit to a gaussian model.

For the purposes of system simulation, interference will need to be modeled in all of these ways: sinusoidal, gaussian, and in combinations. System simulation may directly use these measurements for interference. Additionally, combinations of interference may be generated using the power spectral mask from measurements and an inverse Fourier transform. One may also directly inject sinusoids at specific frequencies and amplitudes to test the effect of a dominant jammer. Gaussian noise may be added at any desired power level as well. The simulation framework supports arbitrary combinations of measured and hand-generated interference (and noise).

It is worth noting that the received interference is a function of the antenna. The antenna will naturally provide some filtering (although a desirable impulse-UWB antenna normally would not be frequency selective as it may disperse the pulse, reducing the peak amplitude in time). To accommodate different antenna choices in the simulation, one may either re-measure the interference profile for that antenna or filter the broad-band TEM

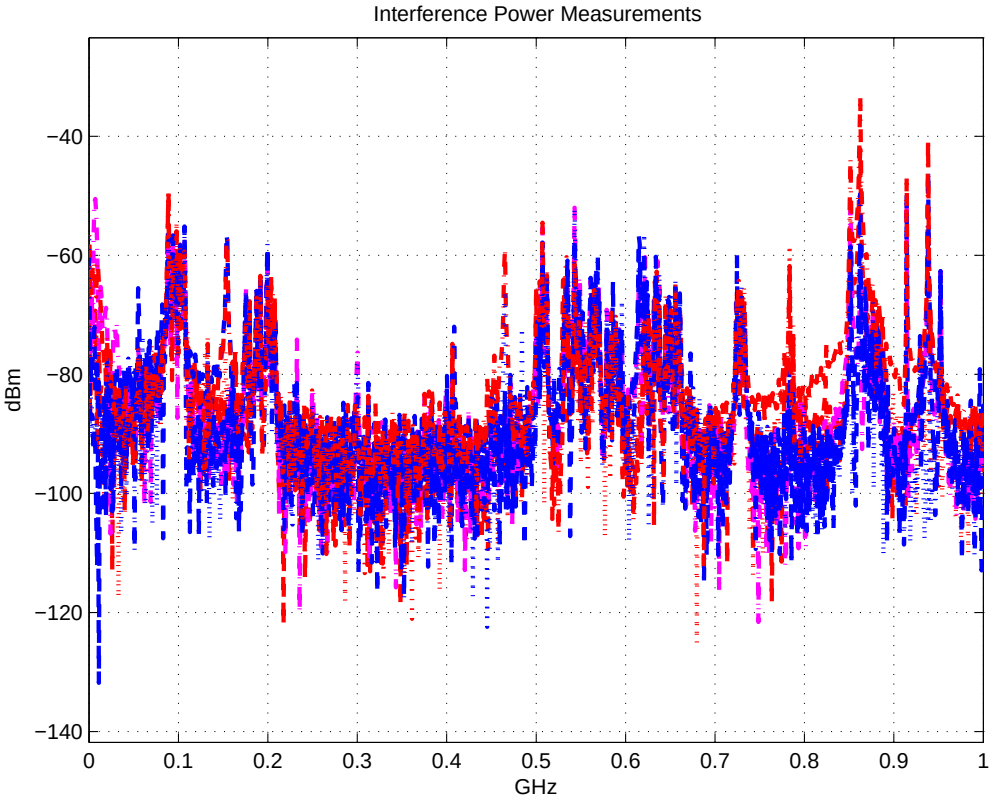

Figure 2.4: Measured Interference PSD (TEM Horn)

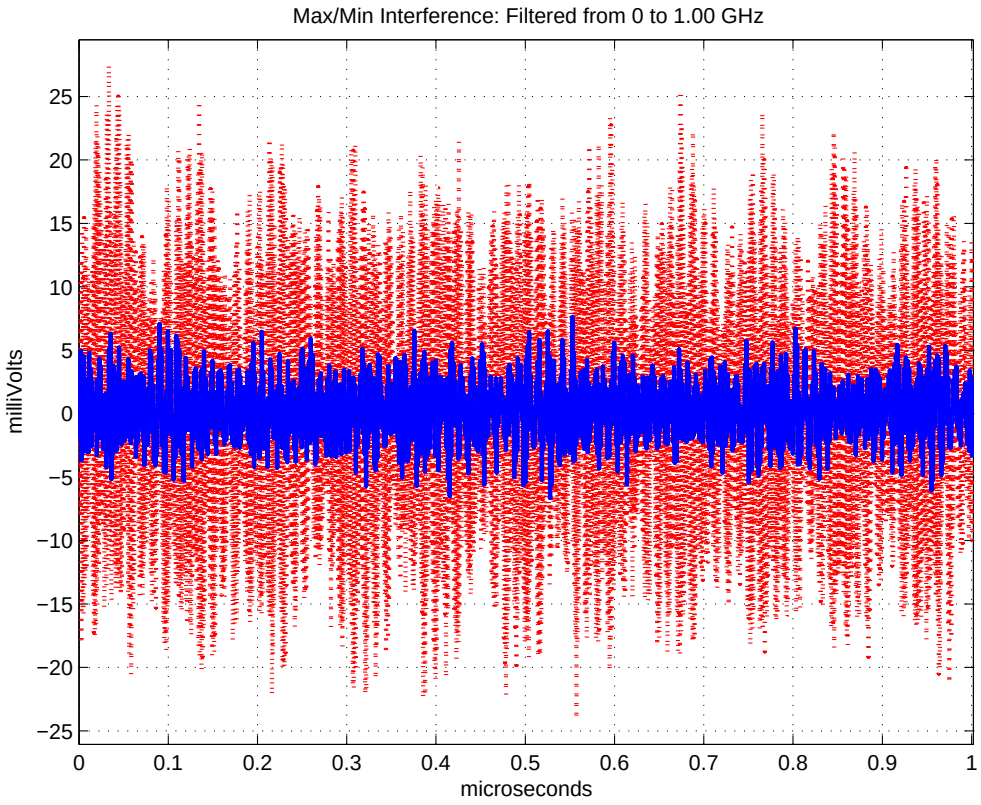

Figure 2.5: Measured Interference in Time: Max. and Min. (TEM Horn)

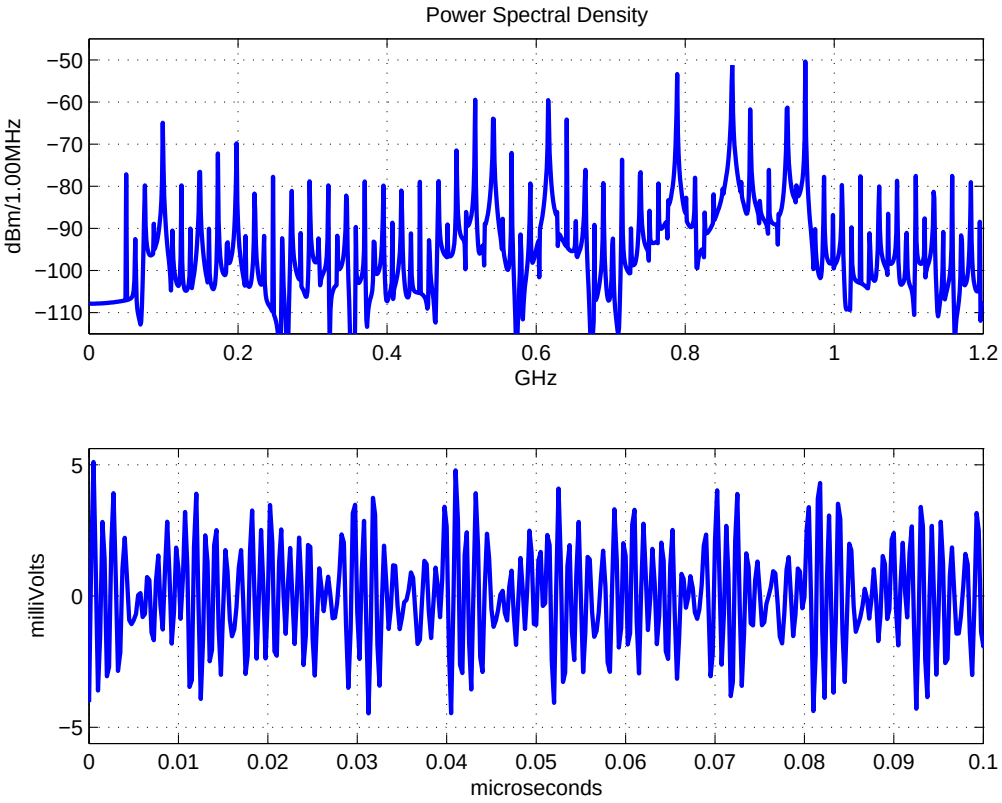

Figure 2.6: Measured Interference PSD ($440MHz/880MHz$ Stub)

measurements and generate equivalent-power time domain sequences. Figure 2.6 shows the use of a power spectral density measurement from a dual-band $440MHz$, $880MHz$ stub antenna to generate interference. The equivalent time domain sequence is shown below the measured PSD; generated by giving a static random phase to the measured sinusoidal amplitudes.

2.4 The UWB Channel

To design the system, knowledge of the characteristics of the *dc* to $1GHz$, indoor, ultra-wideband channel is necessary. In particular, the energy capture over time (often expressed in the delay spread: τ_{rms} and in the RAKE tap diversity) strongly impacts the hardware complexity and hence the power consumption. This parameter dictates the necessary length of time to receive in order to recover the incoming pulse energy and is one of the most important parameters for the signal processing requirements. Additionally, the path loss exponent, a measure of how that energy attenuates with distance, is important to establishing range and gain criteria.

Measurements taken in [65] show a median rms delay spread of $45ns$ to $50ns$ and and average rms delay spread of $59ns$ to $65ns$. Earlier measurements at the University of Southern California also reported rms delay spreads on the order of $50ns$ to $100ns$ [66] and a RAKE diversity of 50 for low SNR environments to capture at least 50% of the energy [67] . The results from [68] agree and provide a model suitable for channel creation for simulations. The path loss exponent for large scale attenuation was previously reported as $\propto (d/1meter)^{-2.4}$ by the same authors in [69].

Channel measurements were also taken in-house by [70], using an antenna array setup to characterize the spatial aspects of the channel. Pulses were generated with a $100MHz$ to $6GHz$ TEM horn, but received on a 2×2 array of smaller $1GHz$ to $6GHz$ horns. While some information exists in the band of our interest ($100MHz$ to $1GHz$), it is rather heavily attenuated and hence was not used for this design.

Channel modeling software was also generated in-house using a ray-tracing algo-

rithm and material reflectivity and absorptive properties in [71] for a model of the upper floor of Cory Hall. This software was capable of producing similar looking multi-path indoor channels (to measured and reported channels in the literature) and has the advantage of not being bandwidth limited by antennas. The disadvantage is that extensive modeling must be done to create a realistic channel response.

As the pulse energy in an indoor, UWB channel has been shown to be quite spread, a simple, single-tap approach to reception may lose too much energy over a multi-tap approach. That, coupled with the lack of knowledge about the receive pulse's specific shape, motivates a more general approach to pulse reception. Ideally we do not wish to reduce system performance too much in order to reduce power consumption. Owing to the large potential variation of the multipath response, some form of channel estimation (and hence added receiver complexity) is needed to recover a significant amount of the available received energy.

A final note about the channel characterization: the Doppler effect can be safely ignored. The fractional change in wavelength due to motion can be found in most introductory physics textbooks [72] as:

$$\frac{\Delta\lambda}{\lambda} = -\frac{\nu_s}{\nu} \tag{2.1}$$

Even when traveling at $100mph$ ($161kph$, or $44.7m/s$), the fractional change in wavelength relative to radio wave propagation speed (approximately the speed of light: $3e8m/s$) is less than 0.000015%.

2.5 Receive Signal Processing

The subject of the backend signal processing required for pulse reception is a broad one, worthy of a dissertation itself. Fundamentally, most of the receive signal processing algorithms attempt to project the received signal plus noise and interference onto a basis that optimizes the detectability of the incoming data. The techniques that can be employed are numerous and beyond the scope of this dissertation. In the face of such a large design space, the decision was made to use a more canonical, matched filter approach to the backend signal processing.

In the presence of AWGN, a matched-filter (consisting of a template of the expected pulse waveform) is the optimal approach [73]. It is worth noting that in the presence of AWGN and narrowband (approximately sinusoidal) interference, though, that the AWGN matched filter is no longer optimal. (The optimal filter attempts to cancel the interference while matching to the pulse shape.) The original AWGN matched filter, called a "template filter" here, while not optimal is still a useful measure of the degradation due to interference. In the interest of simplicity, the approach was taken to use a template filter for the backend signal processing, even though interference was expected to dominate over the background noise. As has been mentioned, in some cases, a summation of interfering sinusoids will appear Gaussian, and the template filter will be optimal. However, in the case of a single (or few) jammer sinusoids, the template filter will perform sub-optimally. Even though that may be the case, it is still a useful metric against which to compare the system performance. Also, it presents a pessimistic (sub-optimal) result, which may be considered conservative from the design perspective. Simulations and analysis will show that even with a simple,

non-optimal signal processing approach, performance sufficient for a sensor network application may be obtained. Note that a relatively low to moderate pulse rate longer than the delay spread of the channel is used to avoid the need for equalizatoin. Future work will include pulse estimation (channel estimation), equalization for faster pulse rates, and perhaps more importantly, interference cancellation. For now, the template filter will be kept programmable, and we will assume that at least the expected pulse shape is known to the receiver.

As the expected pulse is finite in time, the template filter may be easily implemented with a finite-impulse-response (FIR) filter. The length of the filter (and tap bit-width) are a function of the pulse and channel and will be considered later in section 2.8.2 with regard to overall system performance.

Because we consider the channel to be interference dominated, it is likely that at the boundary of communication, the SNR per pulse may not be adequate to satisfy the BER decision criterion. To compensate for this, and to increase the possible transmit distance at the expense of less throughput, we propose overlaying a spreading sequence on the pulse train. This spreading code may be correlated to get an adequate statistic for decision thresholding. The spreading code is labeled as a "PN" ("pseudo-noise") code, although it is fully programmable and may be any spreading code. "PN" refers to a possible implementation of the spreading code.

Low rate, impulse-UWB has an innate problem with fast acquisition because one must search over the entire cycle to find the pulse. This is made more difficult if a spreading sequence is overlaid, since the spreading phase must also be determined. To accommodate

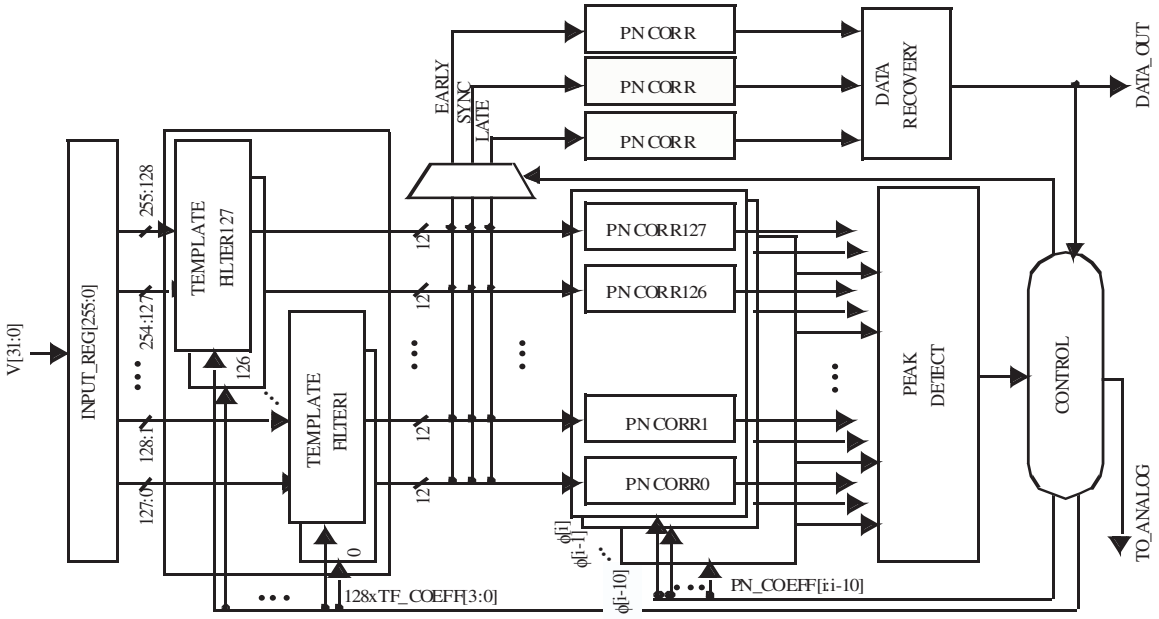

Figure 2.7: Proposed Digital Backend Architecture

this, a hybrid parallel/serial architecture is proposed consisting of a bank of pulse template filters with each filter followed by an independent bank of despreading correlators.

In order to explore the implementation cost of the digital backend, we assume binary antipodal signaling with a spreading code overlaid on top of the pulses to improve reception range. Data from the front-end enters the digital backend, as shown in figure 2.7, which aggregates several consecutive windows of data, $16ns$ long, each sampled at $2GSample/s$ into a block of up to 256 samples ($128ns$). To speed acquisition, 128 samples are searched in parallel by 128 matched filters. To guarantee that a pulse doesn't straddle the boundary between steps, 256 samples are needed. The pulse-matched filters are of length 128 samples ($64ns$), sized to the expected delay spread for an UWB indoor channel, as mentioned in section 2.4. This is done to allow for future experimentation with channel estimation

and/or interference cancellation, even though a smaller pulse template is being used. The matched filter outputs are then sent to either an acquisition or synchronization block. For synchronization, because only three values, 'early,' 'on-time' (or 'sync,') and 'late,' are needed, all of the other matched filter inputs are disabled to save power. For acquisition, we search over all 128 samples and 11 spreading code phases at a time as a compromise between area and search time. Once a correlation peak above the programmable threshold is found by the peak detector logic, the backend switches from acquisition to tracking mode. Because binary antipodal signaling is used, the data recovery block is a simple slicer based on a programmable threshold. In the interest of flexibility, two different spreading sequences are used: one for acquisition and one while synchronized. Both sequences may be of length 1 to 1024. The digital backend is described in detail in [1], which includes simulations of expected BER vs. spreading sequence length, optimum detection threshold selection for the correlators, and implementation results, such as expected power consumption and area estimates.

2.5.1 Acquisition vs. Area

To allow longer distance operation (or to overcome a large jammer), we trade data rate for range by overlaying a spreading code, i.e. in a direct-sequence spread spectrum (DS-SS) manner. This additional coding increases the acquisition burden on the digital backend, and there is a geometric increase in complexity for correlating in parallel over the spreading sequence. For a hybrid parallel/serial architecture, a balance must be achieved between acquisition time and area.

Figure 2.8 shows the trade-off between the total digital area (template filter bank

plus correlator banks) and acquisition time versus the number of phases of the spreading code correlated in parallel for various bit-width inputs. For our design example, an area of around 10.2mm^2 is predicted using a window size of 256 samples for a 128 sample pulse size with 4-bit coefficients in the matched filter, pulses sent at a $5MHz$ rate and a maximal spreading length of 1024 chips. Note that to search a bigger (or smaller) window in parallel, these curves will move up (or down) by the same factor, i.e. $2\times$ window is $2\times$ area. Likewise, if either the pulse rate is sped up or the spreading length decreased, the acquisition time will improve by the square of the factor. For example, a shorter spreading sequence means fewer phases over which to search in addition to a shorter wait for each sequence. Depending upon the desired conditions, these curves may be scaled or shifted to predict the area consumption.

2.6 Link Budget

Using the information from the previous sections, it is now possible to do a rough link budget calculation for impulse-UWB radio. It is worth noting that a typical link budget found in a textbook [74] is often intended only for narrowband systems and has a frequency-based derivation implicitly including narrowband assumptions (e.g. that the signal bandwidth is small enough that it experiences similar fading to the center frequency). Care must be taken if this narrowband link budget model is used, as certain terms like "center frequency" or "sensitivity" may appear misleading. An UWB signal may be defined to have a center frequency, but what happens at that frequency may not accurately represent what happens over the entire bandwidth. Additionally, impulse-UWB systems have a sensitivity, but it is in part a function of the pulse shape and is better represented as

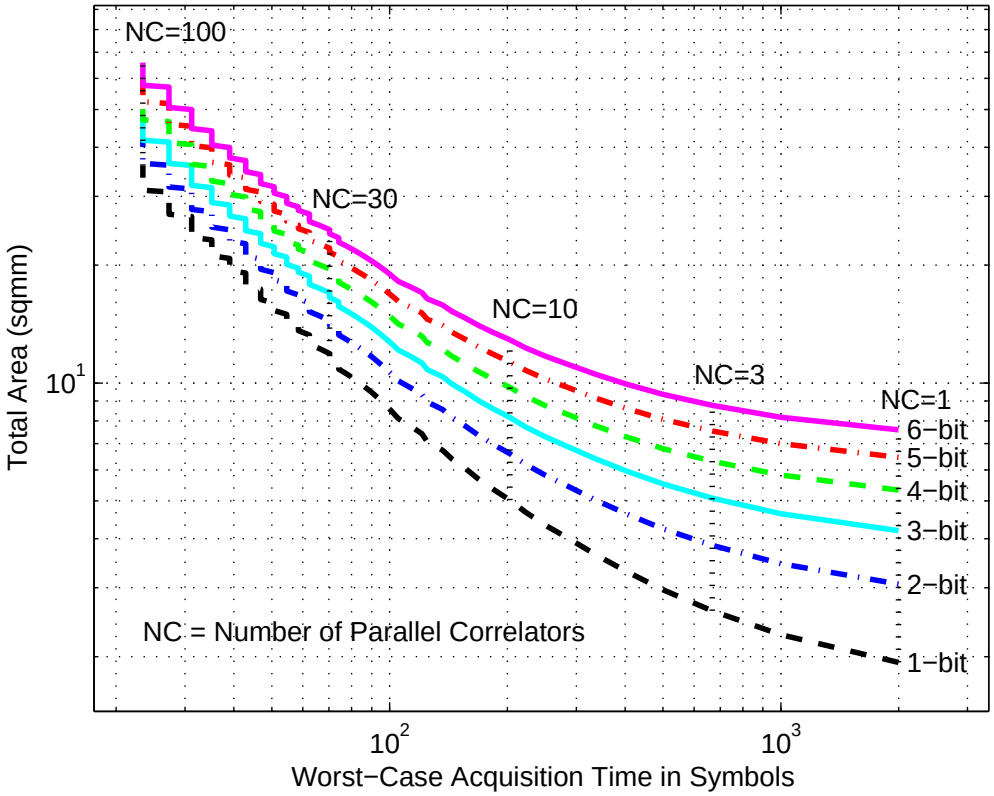

Figure 2.8: Worst-Case Acquisition Time vs. Area

a received power over some period of time than as a simple, single number. A broadband (or, indeed, "ultra-wideband") link budget may be created by either: 1. treating the signals from only the perspective of power (either in the frequency or time domain), as in [75] or by 2. expanding the narrowband Friis power transmission formula (e.g. in [76]) to be a function of frequency and integrating over the desired bandwidth (which also will ultimately result in power) as in [77]. The only difference is that a power-based approach needs no specific knowledge of the antenna characteristics over frequency or the channel fading over frequency, etc. and is therefore easier to calculate.

As an example, using the ideal "Scholtz impulse" waveform from figure 2.3, and a noise power based on interference PSD measurements for the dual-band stub antenna in figure 2.6, we can calculate the received SNR vs. distance using the following equation (assuming free space propagation):

$$SNR = \frac{P_{sig\ at\ 3m} * (3/dist)^2}{P_{noise+interference}} \tag{2.2}$$

If we also note that the FCC only regulates the power spectral levels (see section 1.5.1), then we have an added degree of freedom: the pulse rate. We may send pulses faster (but at a lower amplitude) to meet the FCC mask, or slower (but at a larger amplitude). Taking a simple model for the power spectrum (the Fourier Transform of a repeated BPSK modulated pulse train) then we find that the signal power may be scaled with the pulse rate directly, i.e.

$$P_{sig\ at\ 3m\ at\ f_{pulse}} = P_{sig\ at\ 3m\ at\ 1Mpulse/s} * \frac{f_{pulse}}{1Mpulse/s} \tag{2.3}$$

In the case where the SNR per pulse is not adequate to meet the 10^{-3} BER target of $7dB$

(for BPSK), a spreading sequence is assumed of length:

$$Length_{PN} = ceil\left(10^{(7dB - SNR_{dB})/10}\right) \tag{2.4}$$

The result is plotted in figure 2.9 for pulse rates from $1Mpulse/s$ to $100Mpulse/s$ and from 3 to 100 meters. This simple link budget predicts $1Mbps$ up to perhaps 60 meters, with almost $100Mbps$ possible for 10 meters, albeit for a good (free-space propagation) line-of-sight channel with a relatively bad (non-UWB) antenna choice. It also serves to illustrate the large potential link margin available for trading off data rate vs. distance vs. power consumption.

2.7 System Simulation

To explore the performance trade-offs for this system, a linear simulation model was created that incorporates signal, noise, and interference, including circuit non-idealities such as limited gain, filtering effects, quantization noise, and noise figure. For this model, the metric chosen is the "Signal to Noise + Interference Ratio" (SNIR) at the output of a pulse template filter (as discussed in section 2.5). For the purposes of a simple low power digital implementation, no channel estimation beyond the knowledge of the pulse template is assumed and no interference cancellation is utilized. This provides an estimate of the degradation due to interference and allows us to examine the impact of ADC resolution and the subsequent digital correlation precision on system performance.

We define the sampled, received signal after the ADC as:

$$\vec{V} = \vec{S} + \vec{N} + \vec{I} + \vec{X} \tag{2.5}$$

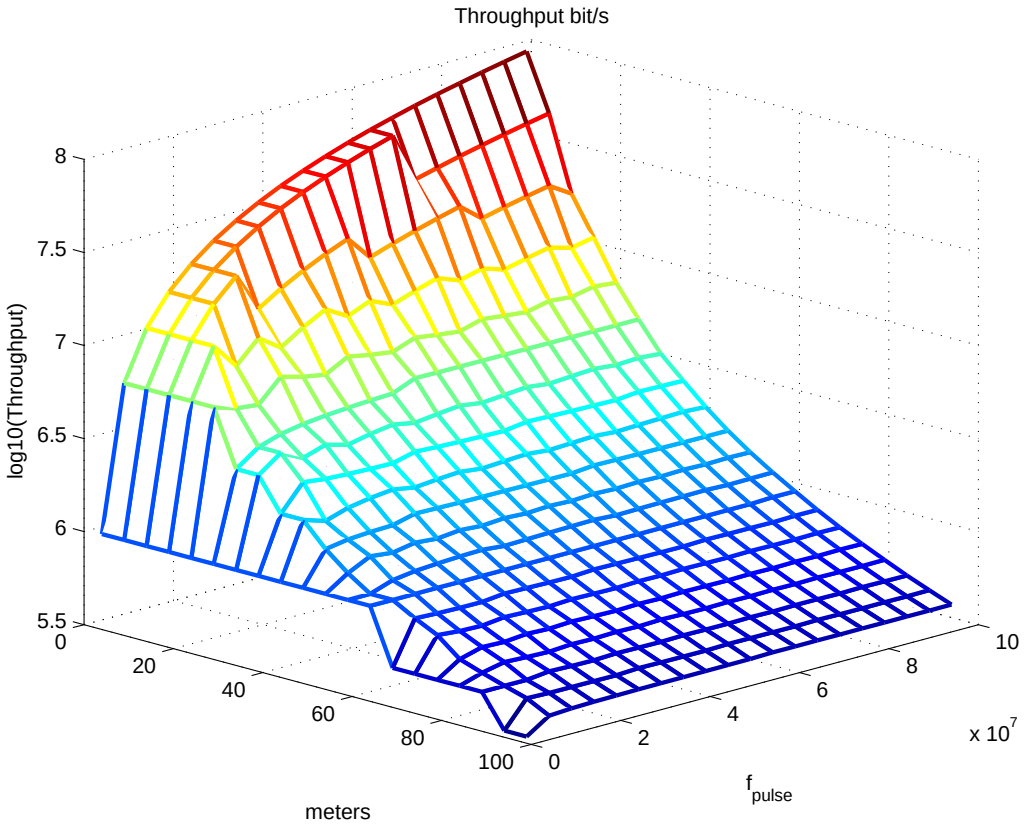

Figure 2.9: Throughput vs. Pulse Rate for Simple Link Budget

where $\vec{S}$ is K samples in time of the desired pulse, which is equal to the received pulse after gain and filtering:

$$\vec{S} = [s[0], s[1], \cdots s[K-1]] \tag{2.6}$$

and $\vec{N}$ is K samples in time of Gaussian noise, where the variance set by the background noise floor times the system power gain and noise factor of the front-end:

$$n[k] = \mathcal{N}\left(0, A_v^2 \cdot NF \cdot kTBR\right) \tag{2.7}$$

and $\vec{I}$ is K samples in time of the total narrowband interference seen at the ADC input,

$$\vec{I} = [i[0], i[1], \cdots i[K-1]] \tag{2.8}$$

where a narrowband interferer is modeled as a sinusoid with the equivalent power and uniform random phase:

$$i[k] = \sum_{n=0}^{N-1} A_n cos\left(\omega_n T_{sample} k + \Theta_n\right) \tag{2.9}$$

and, $\vec{X}$ represents the quantization error (assumed to be zero mean and uniform over $\pm 0.5 lsb$):

$$x[k] = \mathcal{U}\left(0, \frac{\Delta_{ADC}^2}{12}\right) \tag{2.10}$$

Defining the matched filter coefficients as:

$$\vec{W} = \vec{S} + \vec{Y} \tag{2.11}$$

where $\vec{S}$ is again K samples of the desired pulse, and $\vec{Y}$ represents the quantization error for the matched filter coefficients (also assumed to be zero mean and uniform over $\pm 0.5 lsb$):

$$y[k] = \mathcal{U}\left(0, \frac{\Delta_{MF}^2}{12}\right) \tag{2.12}$$

Then the output of the matched filter, Z is equal to:

$$Z = \vec{V}\vec{W}^t \tag{2.13}$$

and we may define the SNIR as:

$$SNIR = \frac{E[Z]^2}{Var[Z]} \tag{2.14}$$

Recall that the noise is zero mean, hence:

$$E[Z] = \left(\vec{S}\vec{S}^t\right) = \sum_K s[k]^2 = P_s \tag{2.15}$$

Then, SNIR is:

$$\frac{P_s^2}{P_s\left(\sigma_{NX}^2 + \sigma_Y^2\right) + \left(\vec{S}\vec{R_{II}}\vec{S}^t\right) + K\sigma_Y^2\left(\sigma_{NX}^2 + \sum_{n=0}^{N-1}\frac{A_n^2}{2}\right)} \tag{2.16}$$

where:

$$\sigma_{NX}^2 = \sigma_N^2 + \sigma_X^2 \tag{2.17}$$

and $\vec{R_{II}}$ is a $K \times K$ matrix whose elements, for $i, j = [0, 1, \cdots K-1]$, are given by:

$$r_{II}[i,j] = \left(\sum_{n=0}^{N-1} \frac{A_n^2}{2} \cos\left(\omega_n T_{sample}\left(i-j\right)\right) \right) \tag{2.18}$$

Note that this simulation model is based on a time-domain approach, assuming that a pulse is sent episodically (on average every T_{pulse}) and sampled at discrete time steps. One period is treated at a time, over-sampling by at least 10× the Nyquist signal bandwidth to ensure accuracy. A time domain simulation is more natural for impulse-UWB, but use of frequency domain information is desired as well (i.e. interference PSD measurements or filtering from the antenna or front-end circuits). To accommodate this, the Fourier and inverse-Fourier transforms are used as appropriate. (Also note that Matlab scales the Fourier transform by the length of the sequence, N, and in some cases dividing or multiplying by N may be required to maintain constant power.)

This simulation model also provides a flexible, extensible testbed to allow us to vary or change any aspect of the communication link. For example, the gain, front-end noise figure, number and position of filtering poles, pulse shape, interference, finite A/D and template precision, etc. may be arbitrarily varied to determine the effect on the system $SNIR$. Additionally, by simply using the time-domain results directly (i.e. skipping the $SNIR$ calculation and actually performing the backend correlation on the received pulse + interference + noise), one may also run monte-carlo simulations to double-check the $SNIR$ results.

A final comment should be made on the use of

$$\frac{\Delta^2}{12} \tag{2.19}$$

for the ADC quantization error, σ_X, at low ADC resolutions. This approximation is in-

valid for low ADC resolutions. For high-resolution ADC's, the error from one output to another is considered approximately flat, and hence white over the region. Given this probability distribution, it is relatively trivial to calculate the above equation. However, for low resolutions, the probability distribution is not flat over the interval and the resulting quantization error will be larger. Assuming the input is AWGN (i.e. in a low SNR case where quantization error will be more significant), then the ratio of the standard deviation of the AWGN to the quantization step determines where the $\Delta^2/12$ approximation is valid. As the ratio tends towards zero (the quantization step size is much larger than the input standard deviation), the quantization noise power tends towards $\Delta^2/4$ [78]. As the ratio increases, the probability of clipping increases and again the quantization noise power will increase. The calculation of σ_X should be corrected as appropriate given the simulation conditions.

2.8 Simulation Conclusions

Using the simulation framework and $SNIR$ calculation from the previous section, it is possible to specify many of the system requirements. In particular, we may use it to analyze and specify the ADC and template filter precision, template filter length, noise figure requirements, and to explore the power efficiency of modulation choices.

2.8.1 ADC Precision

To examine the level of ADC precision needed, the SNIR per pulse is calculated at a given level of interference, noise, gain, etc. The results are shown in figure 2.10, plotted

against the total received interference power. Calculations used a gaussian monocycle pulse [43] sent at a $5MHz$ rate. Interference was generated based on measurements taken in our lab with a spectrum analyzer to represent 'typical' levels and then scaled over the range shown. The UWB channel model was for a 3 meter path, derived from an in-house ray tracing tool which estimates the impulse response using a 3-D indoor building model [71]. A input-referred noise figure of $10dB$ was assumed for the gain stages, and the gain was set with automatic gain control (AGC) to allow only an infrequent amount of limiting. To model finite bandwidth of the input gain stages, the pulse is filtered with a 5-pole roll-off at $1GHz$.

In figure 2.10 we see that only at low levels of interference, where thermal noise dominates, does extra resolution in the ADC improve SNR. As interference increases the impact of higher resolution in the ADC decreases. This realization, that ADC resolution in an interference dominated environment is not critical, allows us to simplify the ADC design to 1-bit to save power without incurring a tremendous penalty in performance. Typical values for the aggregate interference over dc to $1GHz$ measured in our labs were around $-40dBm$, predicting about $\sim 7dB$ of loss relative to a higher resolution ADC. This result agrees with previous work on time-based mono-bit digital receiver modeling using a matched filter in the presence of AWGN[79]. Note that this performance is not optimal, as it makes no effort to cancel the interference or gather more pulse energy from reflections. For the goal of low power operation and low cost/complexity, though, the predicted performance is adequate for the application requirements of sensor network radios, and the power savings is predicted to be significant.

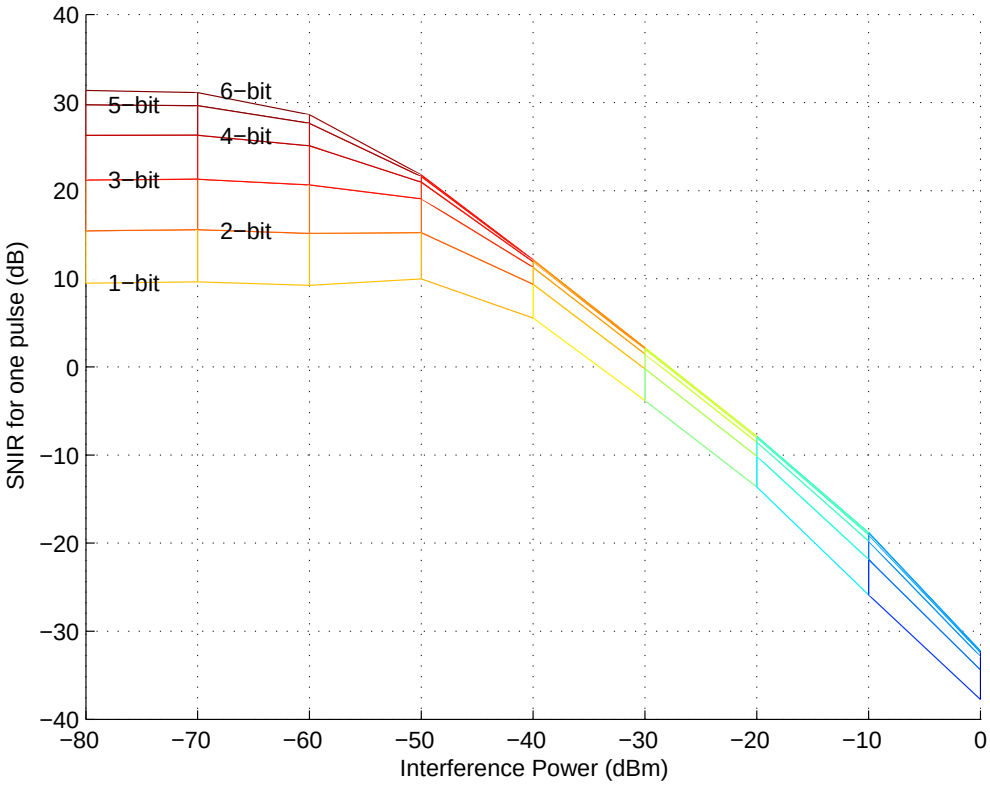

Figure 2.10: SNIR vs. ADC Bitwidth

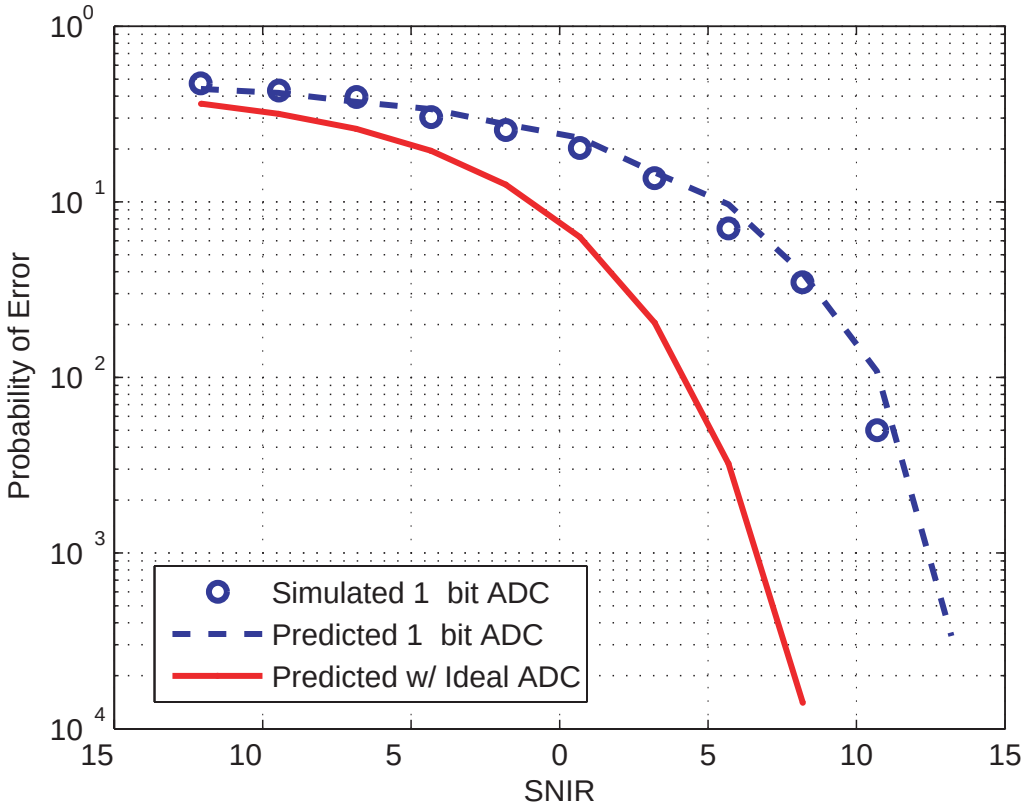

Figure 2.11: Probability of Error: Predicted vs. Time Domain Simulation

The linear model calculation over ADC resolution shows a trend that indicates that 1-bit is sufficient; however, linear modeling of quantization noise breaks down at very low resolutions (1 to 2 bits). To verify these results, a time domain BER simulation was also run with and without the ADC resolution limit for the same conditions. The results are shown in figure 2.11. The time-domain simulation matches the linear model well, deviating by only a dB or so over the range. This confirms that the linear model accurately evaluated the effect of ADC resolution and that a 1-bit ADC may be used to save system power without an excessive degradation in performance. Note that a 1-bit system also helps simplify the front-end design by obviating the need for automatic gain control (AGC).

To compensate for the loss due to ADC quantization in the system link budget it would be more efficient to increase the transmit power by 4 given the low transmit power regulation: approximately $-10dBm$ total of average power over dc to $960MHz$. Unfortunately, the FCC limits the power spectral density for UWB emissions, so transmit power is fixed for a fixed pulse rate. This results in a loss of data throughput (by approximately 1/4) to compensate for the choice of a 1-bit A/D converter. However, high throughput is not critical for sensor network applications which require data rates on the order of $10kbps$ to $100kbps$ [80]. If higher performance or data rates were desired, it is likely that 2-bits to 4-bits would be required (as is also shown in [78][81]).

ADC resolution is one of the most important parameters of the design requirements, as the high sampling rate (on the order of $2GHz$) may preclude low power operation entirely. Low resolution, high sample rate ADC designs published in recent years may be surveyed to get a predictive estimate of power consumption for a given specification. Using

the results from [82], ADC's are compared using the following figure of merit (FOM):

$$FOM = \frac{2^{N_{bits}} F_{sample}}{P_{diss}} \tag{2.20}$$

Table 2.2 shows ADC FOM performance for recent publications. Using the best figure of merit (prior to 2006) of approximately $4*10^{11}$, we estimated the power consumption of a 4-bit $2Gsample/s$ ADC at $80mW$. As we move to lower resolutions, the ADC simplifies and the power decreases roughly as $1/2^N$ as most high-speed ADC's are a flash architecture and power scales roughly as the number of comparators. While this implies that even a 1-bit ADC (degenerate case) is on the order of $5mW$, in reality, the value will be less than $1mW$. In fact, two very recently published ADC's boast a factor of $20\times$ improvement in performance vs. power consumption. This reduces the 4-bit, $2Gsample/s$ ADC above to $4mW$, and the 1-bit ADC to $250\mu W$ (which is on the order of what is achieved in this dissertation.) Even a $1mW$ power consumption can represent a substantial fraction of the total power budget for a sensor network radio, though. For a given process there is a sense of the "natural" dynamic range based on the matching performance of the devices, usually on the order of 2-bits or 3-bits. Below this resolution, only simple comparators are needed (i.e. without offset cancellation or averaging) which reduces ADC power consumption. However, given the power concern there is still a strong drive to lower the ADC resolution, if possible, to save on power consumption.

Table 2.2: Low Resolution ADC Figures of Merit

f_{sample} $(Gsample/s)$	P_{diss} (mW)	**# bits**	**FOM**	**Ref**	**Year**
1.2	2.5	4	$7680 * 10^9$	[83]	2006
0.6	5.3	6	$7250 * 10^9$	[84]	2006
1.0	70	4	$229 * 10^9$	[85]	2004
2.0	310	6	$413 * 10^9$	[86]	2003
1.6	328	6	$312 * 10^9$	[87]	2002
0.5	200	6	$160 * 10^9$	[88]	2002
1.3	545	6	$153 * 10^9$	[88]	2001
1.1	363	6	$194 * 10^9$	[88]	2001
0.7	187	6	$240 * 10^9$	[88]	2000
0.8	400	6	$128 * 10^9$	[88]	2000
0.5	400	6	$80 * 10^9$	[88]	1999
0.5	375	6	$85 * 10^9$	[88]	1999
0.4	188	6	$136 * 10^9$	[88]	1998

2.8.2 Template Filter

Because the processing has been moved into the digital domain, one concern is that the computational load will either balloon in area or dominate power consumption itself. To evaluate this, the pulse template filter resolution and length are examined.

Tap Width

Using the same conditions as section 2.8.1, the SNIR per pulse is recalculated against the template filter coefficient bit-width as shown in figure 2.12. In this case we see that a 1-bit template coefficient is not necessarily adequate, as performance is predicted to be worse over all levels of interference. Note that template coefficient resolution is a separate issue from ADC resolution. For a 1-bit ADC input, the filter is correlating the zero crossings of the input against the expected pulse shape. Intuitively, the more accurate our knowledge of the expected pulse shape, the larger we can weight those zero crossings to estimate the presence or absence of a pulse. This is distinct from the case where the ADC becomes overwhelmed in noise. Increased ADC resolution only captures the noise more accurately, which doesn't aid in estimating the presence of a pulse unless we are able to subtract the noise.

Figure 2.13 depicts the result of a time domain simulation to double-check the linear system model. Results for this case agree that more than 4-bits of coefficient resolution in the template filter produce no improvement in the system performance.

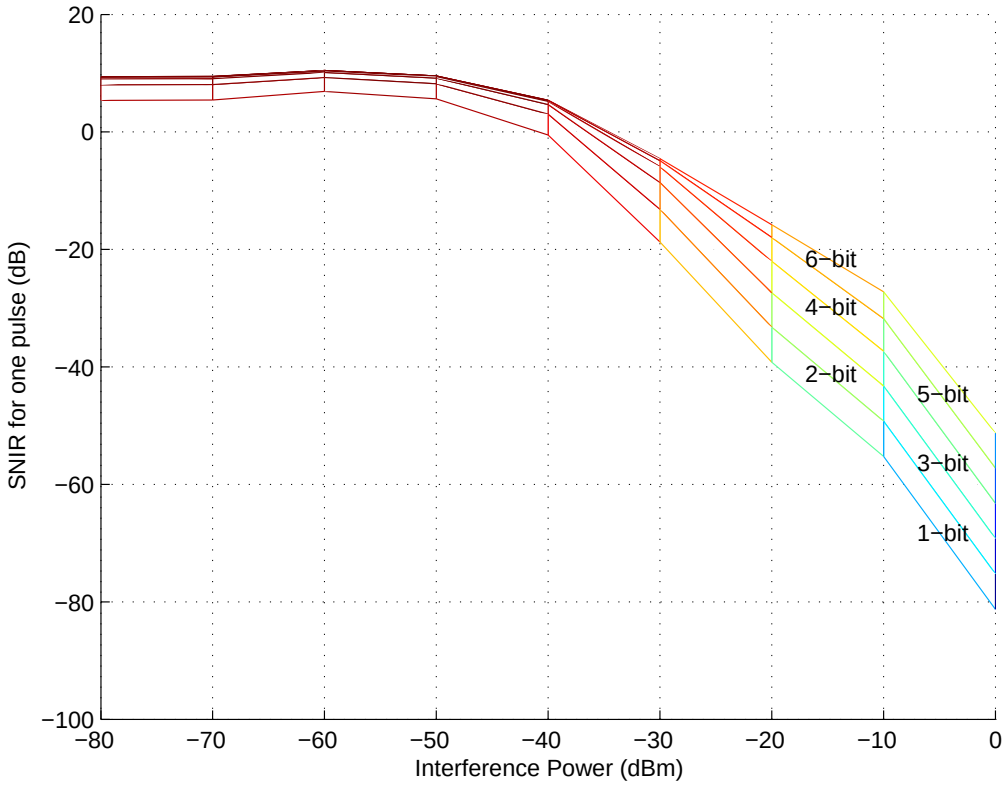

Figure 2.12: SNIR vs. Template Filter Bit Width (ADC Bitwidth=1)

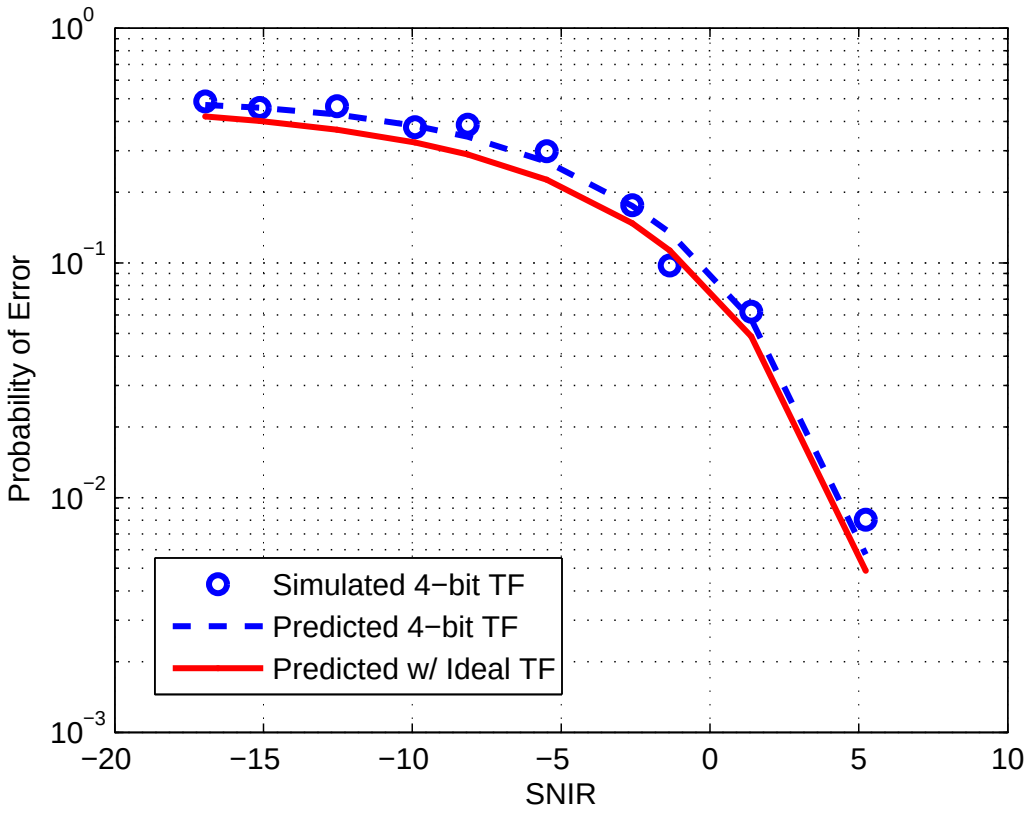

Figure 2.13: Probability of Error: Predicted vs. Time Domain Simulation

Template Filter Length

The filter length is a function of energy capture. This is an issue of the channel characteristics and has repercussions in chip area and power. It is possible to change the simulation to capture a wider or smaller window than the received pulse to observe the impact on SNIR. As expected, capturing a smaller window misses energy which is reflected in the SNIR.

Figure 2.14 depicts the impact of template filter coefficient resolution in area per filter block for different filter lengths. Increasing filter length causes a geometric increase in area. Increasing the matched filter resolution (bit-width) linearly increases the area with a tap-size dependent slope. While the delay spread of the channel may be larger than $64ns$ (128 samples), often a majority of the energy is concentrated in $32ns$ or less of time, which saves area at the expense of worst-case performance. Note that the filter coefficients are kept fully programmable, in the interest of maintaining flexibility for experimentation.

2.8.3 Noise Figure

The simulation was re-run with a 1-bit ADC, 4-bit template filter and various values of noise figure (NF) up to $20dB$. No differences were noted until the noise figure approached $20dB$. This confirms that the interference levels are well above the noise floor and that we can trade-off the NF for power consumption in the front-end gain circuits without degrading performance. The primary front-end design constraint becomes one of impedance matching to the antenna and providing reasonable gain with a moderate to poor noise figure. This helps reduce power, because often the only way to achieve a low noise

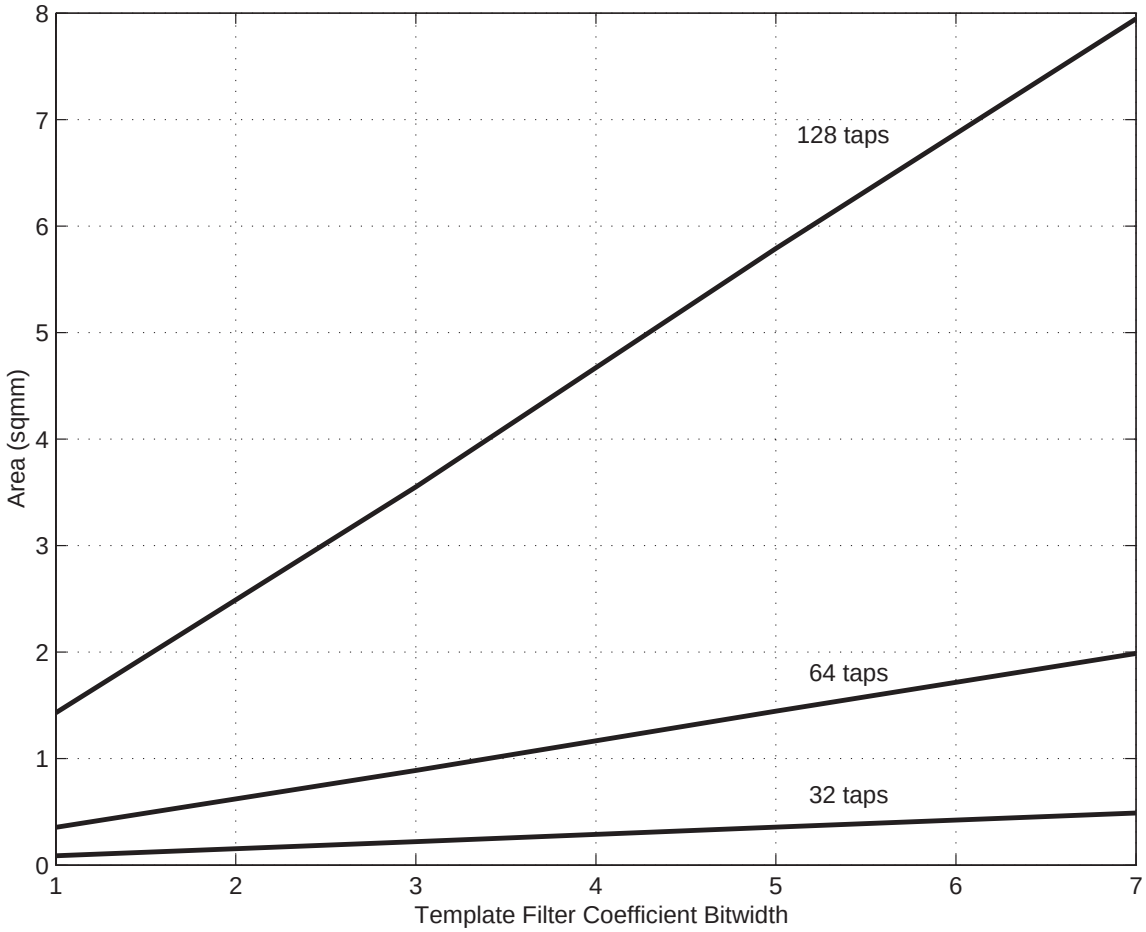

Figure 2.14: Template Filter Length and Bitwidth vs. Area

figure is to consume a large amount of current.

2.8.4 Modulation

Equation 2.14 predicts the SNIR at the output of a pulse template filter responding to an input pulse in the presence of noise and interference. This may be used to quantify the performance of pulse amplitude modulation (PAM), but to investigate on-off keying (OOK), pulse position modulation (PPM) or Bi-Orthogonal modulation, the output of the template filter must be determined in the absence of signal as well as the presence. Expecting that the channel will be interference dominated, and hence low SNIR, only the simple variants of these modulation schemes will be discussed. (No higher order modulation schemes, which generally require higher SNR, will be discussed.)

In the absence of a received signal, we define $\vec{U}$ as:

$$\vec{U} = \vec{N} + \vec{I} + \vec{X} \tag{2.21}$$

where the output of the matched filter is now Z_o:

$$Z_o = \vec{U}\vec{W}^t \tag{2.22}$$

which is zero-mean, and has variance $Var\left[Z_o\right]$ equal to

$$P_s\left(\sigma_{NX}^2\right) + \left(\vec{S}\vec{R_{II}}\vec{S^t}\right) + K\sigma_Y^2\left(\sigma_{NX}^2 + \sum_{n=0}^{N-1}\frac{A_n^2}{2}\right) \tag{2.23}$$

which is:

$$Var\left[Z_o\right] = Var\left[Z\right] - P_s\left(\sigma_Y^2\right) \approx Var\left[Z\right] \tag{2.24}$$

assuming $\sigma_Y^2 \ll \sigma_X^2$ (i.e. that the matched filter's resolution is at least 3-bits larger than the ADC).

As the variance of Z_o is essentially equal to the denominator of the $SNIR$ per pulse, we expect OOK and 2-PPM performance to be approximately $3dB$ worse than binary antipodal (2-PAM). Furthermore, receiver power consumption for 2-PPM will increase as the analog front-end is on for two reception windows during a pulse repetition period, assuming the PPM separation is larger than the delay spread of the channel to ensure orthogonality. This implies that OOK is also undesirable compared to binary antipodal since it achieves worse performance for the same receiver power consumption. 2-PPM is even worse compared to binary antipodal and OOK for modulation because it has both worse BER performance and doubled power consumption. Note that transmit power is severely limited by FCC specification, hence the receiver power dominates the total power consumption even for low transmit efficiency.

A variant on PAM and PPM modulation is Bi-Orthogonal signaling which combines binary antipodal and 2-PPM. While this achieves a higher data rate at 2-bits/symbol, this is cancelled out by doubling the power consumption, and the BER performance is predicted to be similar to 2-PPM due to the shorter minimum distance between the orthogonal signal components. This implies that Bi-Orthogonal signaling is not as power efficient as Binary Antipodal. For this reason, binary antipodal was chosen as the preferred modulation scheme.

2.9 Sampling Clock Requirements

The use of impulse-UWB signaling may imply tight timing tolerances, but we will show that the requirements are reasonable. The main issues associated with sampling clock generation are the jitter performance of the system clock and matching between the TX and RX clocks. In the two sections below, performance bounds are derived and compared to existing work.

2.9.1 Jitter

The allowable jitter variance may be approximately mapped to a phase noise requirement for the oscillator[89]. The lowest frequency we need to consider for jitter is the symbol rate since the digital backend will track any frequency variations slower than that. Assuming that the mean square phase deviation over a symbol is much less than 1 radian and taking the phase noise spectral density to be of the form:

$$\mathcal{L}(\Delta f) = \frac{K}{(\Delta f)^2} \tag{2.25}$$

then the corresponding phase noise, given the total accumulated jitter σ_T over T_{symbol}, is:

$$\mathcal{L}(\Delta f) \approx 2\pi^2 \sigma_T^2 \left(\frac{f_c}{\Delta_f}\right)^2 \frac{1}{T_{symbol}} \tag{2.26}$$

For an accumulated jitter of $75ps$ over a $100\mu s$ symbol, which allows for a $-0.14dB$ degradation in the SNR for an ideal matched filter with a gaussian monocycle, we would require $-103dBc/Hz$ at a $100kHz$ offset for $f_c = 100MHz$. This level of performance seems achievable, as [90] reports a low power oscillator, digitally trimmable to $0.3PPM$ with phase

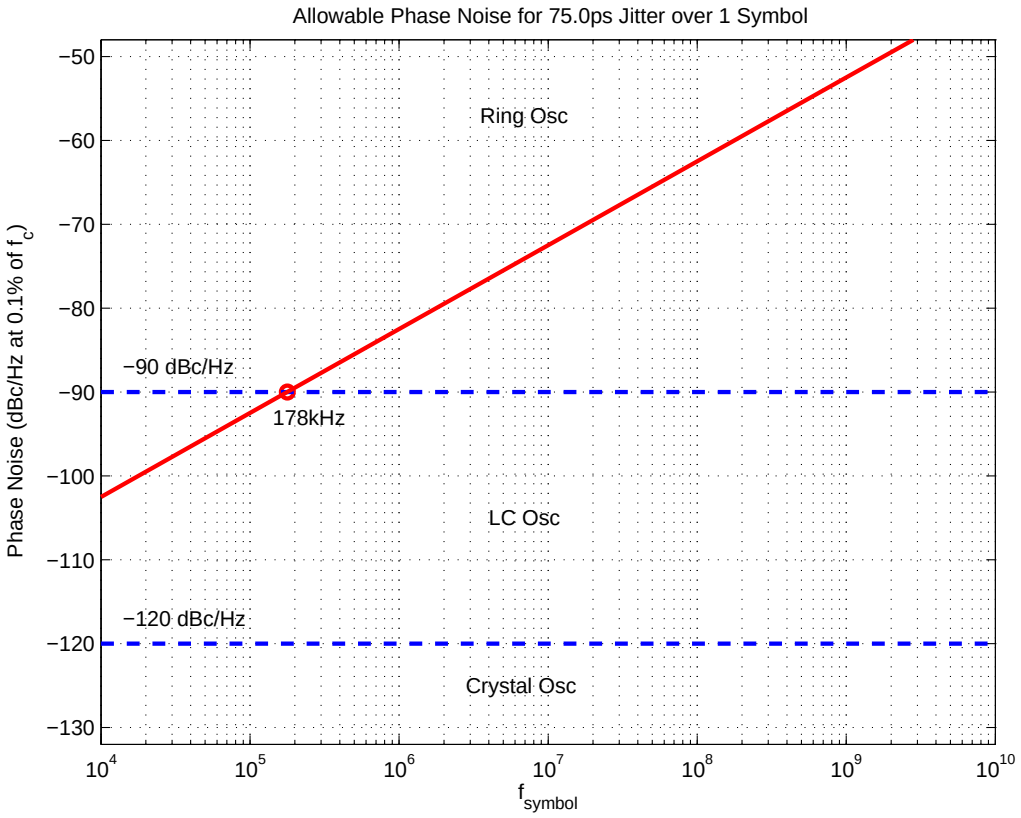

Figure 2.15: Allowable Phase Noise for 75ps Std. Dev. Jitter Over One Symbol

noise $-100dBc$ at a $100Hz$ offset. Figure 2.15 shows these phase noise requirements versus symbol rate along with boundaries for performance from common implementations based on reported results in the literature. One can see that the jitter specification is relaxed enough to allow for a ring oscillator implementation, suggesting that complete integration is possible. However, as we will see, the matching requirement between the TX and RX oscillators will be more stringent and will likely preclude a ring oscillator without an external precision component or crystal.

2.9.2 Frequency Mismatch

The matching between the transmit and receive clocks must be accurate enough to allow the digital backend to track the drift. In our design, the correlation results are compared at the symbol rate, thus requiring the drift over a symbol's reception to be a fraction of a sampling bin to keep the energy within that correlator. Defining $fc = 0.5\,(f_{RX} + f_{TX})$ and $\Delta f = |f_{RX} - f_{TX}|$, we may express this constraint as:

$$\frac{\Delta f}{f_c} \approx \frac{1}{2} f_{symbol} \Delta T_{bin} \tag{2.27}$$

Given a minimum symbol rate of $10kHz$, then for a $-0.15dB$ degradation from an ideal matched filter, $100ps$ of drift is the worst-case allowable. From figure 2.16 we see that this requires very stringent matching with $\Delta f / f_c$ equal to $0.5PPM$. This value is only necessary to support the slowest symbol rate, i.e. a $1MHz$ pulse rate with a length 1000 spreading-gain sequence. This implies that a crystal oscillator will be necessary if longer transmit ranges (and hence longer spreading codes) or slower pulse rates (for heavily duty-

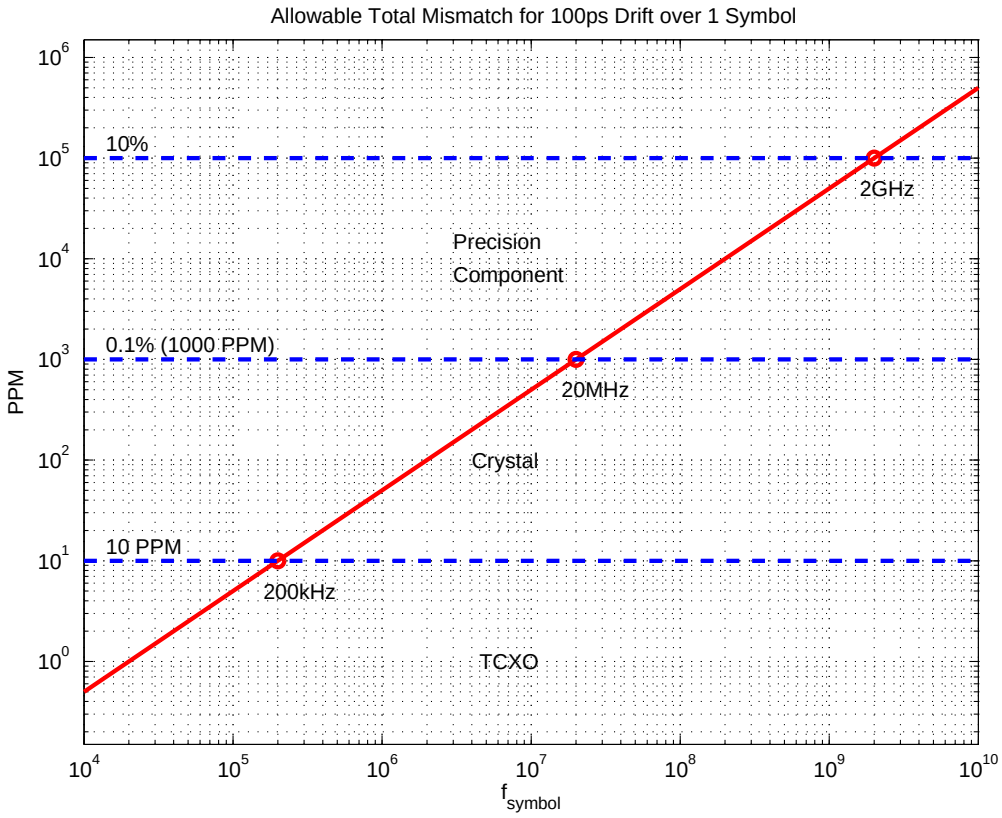

Figure 2.16: Allowable Mismatch for 100ps Drift over One Symbol

cycled power savings) are used. For our system, a lower cost and lower precision crystal was selected in conjunction with tuning through oscillator pulling to meet the matching specification.

2.10 Gain Range and Offset

Ideally, a comparator switches exactly when one input is infinitesimally larger than the other. With this level of accuracy, no gain stages would be necessary as one could simply sample the antenna voltage directly. In practice, the offset voltage seen at the input of the comparator will determine the minimum amount of gain necessary to ensure accurate sampling. Modeling the offset, the probability of a comparator making a mistake is calculated as:

$$P\left(Error\right) = \int P\left(Error|V_{os}\right) p\left(V_{os}\right) \mathrm{d}V_{os} \tag{2.28}$$

Assuming the offset voltage (V_{os}) is Gaussian with a mean systematic offset $\mu_{V_{os}}$ and variance $\sigma^2_{V_{os}}$, and taking the input as Gaussian with a zero mean and variance σ^2_N, we can calculate the probability of a comparator making an error. The exact impact of a comparator error depends on a particular set of $\vec{Y}$, the matched filter coefficients. Hence the probability of a sampling error is analyzed for different $\sigma_N/\sigma_{V_{os}}$ ratios (assuming the comparators are designed without systematic error).

For an error rate of 1%, the minimum gain necessary may be determined relative to the expected offset voltage variance. Mismatch simulations indicate that simple differential sampling with near minimum sized devices yields offsets on the order of $10mV$ for a $1GHz$

Table 2.3: Error Probability Given ADC Offset

$\sigma_N/\sigma_{V_{os}}$	10	20	33	50	100
$p\,(Error)$	3.2%	1.6%	1.0%	0.6%	0.3%

tracking bandwidth. Incorporating offset cancellation into the comparator can bring this number down to several millivolts. The data in table 2.3 implies that the input signal to a $1-bit$ comparator must be on the order of $33mV$. Maximum gain would be needed for a minimum input signal, i.e. thermal noise at room temperature over $1GHz$ of bandwidth at the 50Ω input times the input noise figure, corresponding to $75dB$ of gain. However, the expected levels of interference are much higher than this minimum, e.g. for a $-40dBm$ input, we need only about $25dB$. The minimum gain required depends upon the maximum interference level we wish to accommodate without clipping. At very high interference levels, no gain would be required at all, but performance would be very poor due to the large amount of interference. A reasonable range was chosen from about $50dB$ to $10dB$, with a gain stage architecture that allows the ability to directly trade current consumption for gain.

Especially for high-gain, but even for low gain, voltage offset will need to be controlled. Note that offset arises not only from the comparators, but also from the preceding gain stages. Without the use of offset cancellation techniques and/or capacitive coupling between stages, the systematic offset would saturate the comparators.

2.11 Filtering

The gain stages will naturally filter the incoming signal. The questions arise as to what level of filtering is tolerable and whether or not more sophisticated filtering could be incorporated to improve performance (through, for example, filtering out interferers).

To examine the possibility of filtering interference to improve performance, a simple single LC notch filter response centered at $1GHz$ was plotted in figure 2.17. It is easily noted that a single LCR notch is inadequate: heavily attenuating the desired band from $100MHz$ to $1GHz$ in addition to the target $1GHz$ interferer. A much higher-order notch would be required to filter only the interferer. Comparing a typical interference bandwidth for a fixed notch, for example the cell-phone band from $824MHz$ to $849MHz$, a Q of greater than 50 would be required. This would be very difficult to create on-chip for a typical CMOS process without incurring a large power penalty. Likewise, if that notch were made tunable, perhaps to cover cellphone, pager, or the ISM band, the difficulty only increases. The desire to use a lower-order filter stems from an argument that integrating complex, high-order analog filtering often implies higher power and/or area through the need for large inductance values and intermediate buffer/biquad gain stages. As our simulations have indicated that satisfactory communication for sensor network applications may still be maintained even in the presence of large interferers, the decision was made to abandon integrating interference-canceling filtering into the front-end gain stages

The question still remains as to what amount of filtering is tolerable. Power may be saved through limiting the bandwidth of the gain stages to only the minimum necessary. Likewise, several of the largest expected interferers are present at the upper edge of the

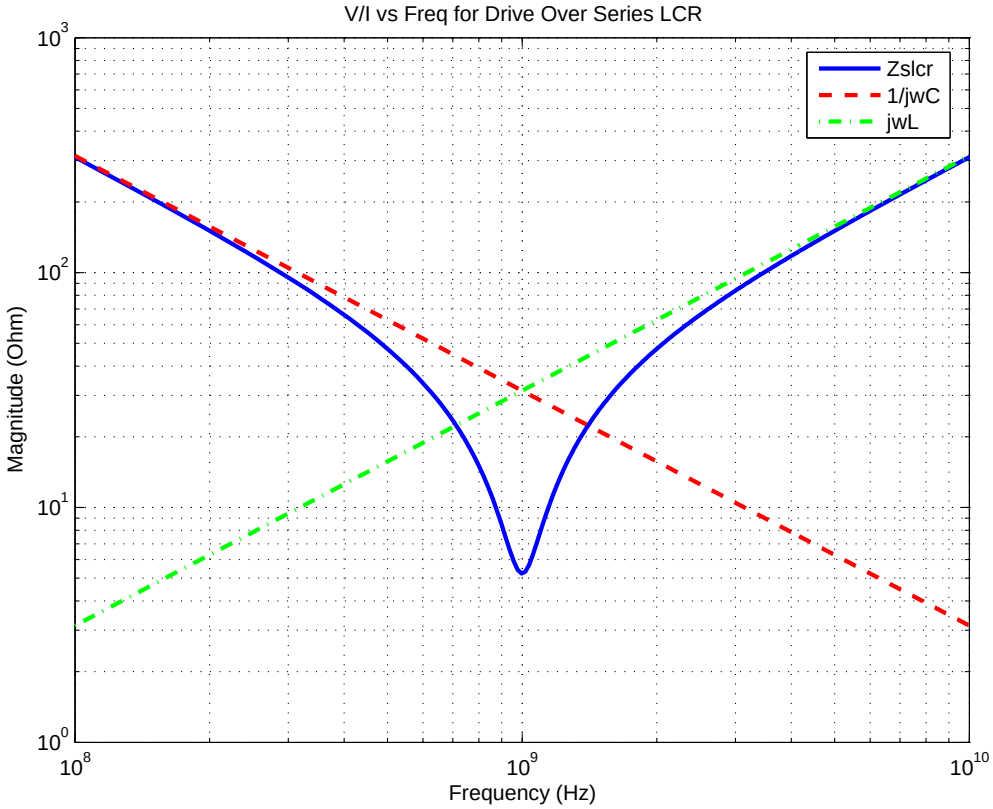

Figure 2.17: Single Pole Notch Filter Impedance

band (around $900MHz$). Further attenuation at these frequencies may help temper the interference impact even as it causes signal loss.

Firstly we investigate the affect of the simplest scheme: all gain stages have the same bandwidth as compared to a more advanced approach of a Butterworth filter for various orders. A plot of the frequency domain response is shown in figure 2.18. Note that as order increases, the $-3dB$ bandwidth of the simple concatenation of $900MHz$ poles begins to seriously degrade. The Butterworth filter bandwidth was chosen such that for mid-order ($N = 3$), the $-3dB$ bandwidth is $600MHz$ and the attenuation at $900MHz$ is $> -10dB$. However, to keep the comparison fair, each simple filter is compared to an equivalent order Butterworth filter response. As mentioned previously, this complex Butterworth filter will not be implemented, but exists as a comparison point for evaluation. The impulse responses are also shown in figure 2.18. From a time-domain perspective, the signal energy loss also may be observed in the spreading and peak reduction of the impulse response. The incoming signal effectively will be convolved with this impulse response.

A slightly more ambitious approach to simple filtering involves increasing the $-3dB$ bandwidth of each filter stage as the order increases in order to maintain constant $-3dB$ bandwidth for the entire filter. Figure 2.19 shows the frequency domain response for simple concatenated poles with a constant $-3dB$ frequency against the Butterworth filter for various orders. The impulse responses are shown below the frequency domain plots. This approach may seem promising to regain signal power while providing some attenuation at $900MHz$.

Table 2.4 compares the energy difference between the filter orders. Ideally, a pulse

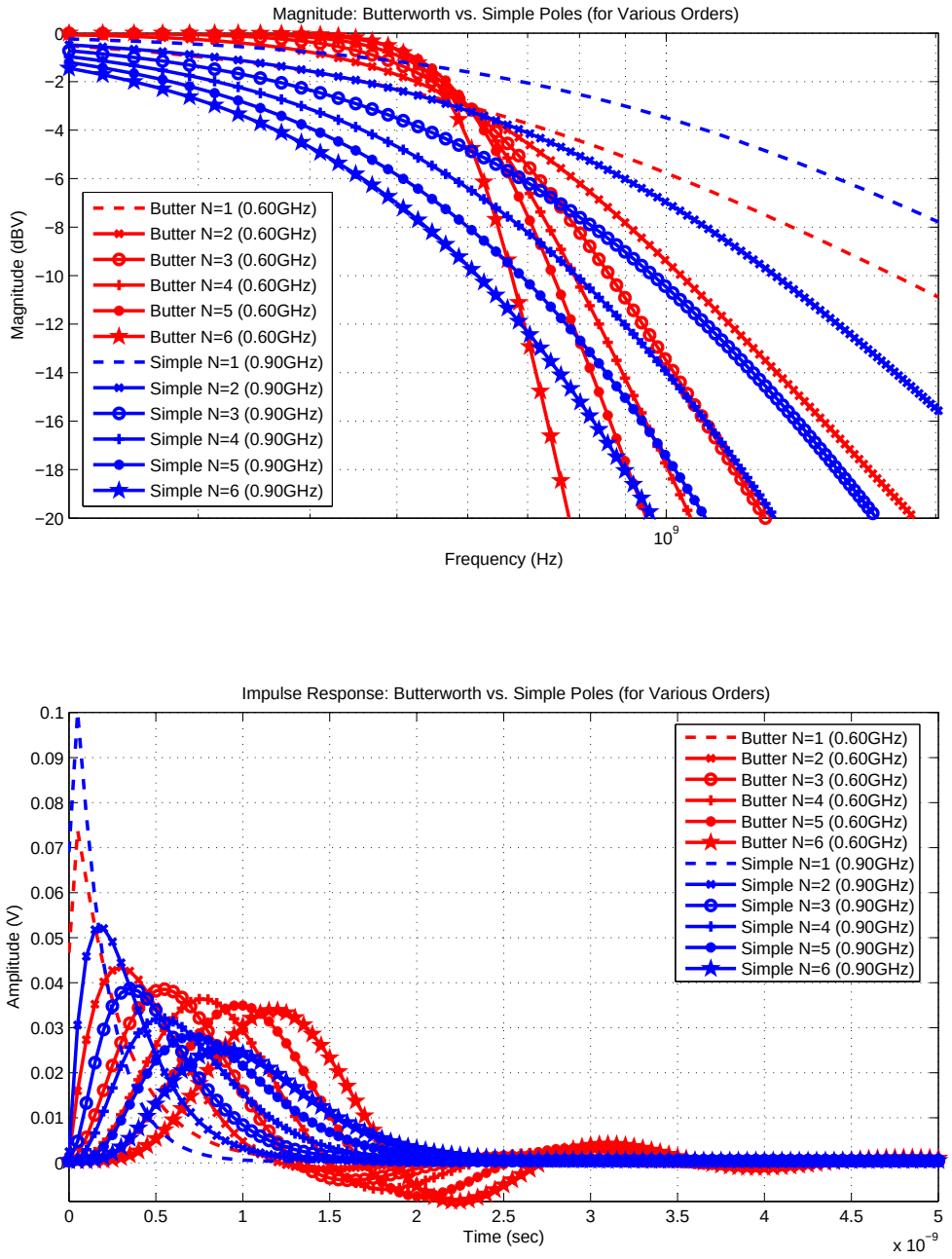

Figure 2.18: Simple Pole Filtering vs. Butterworth for Various Orders

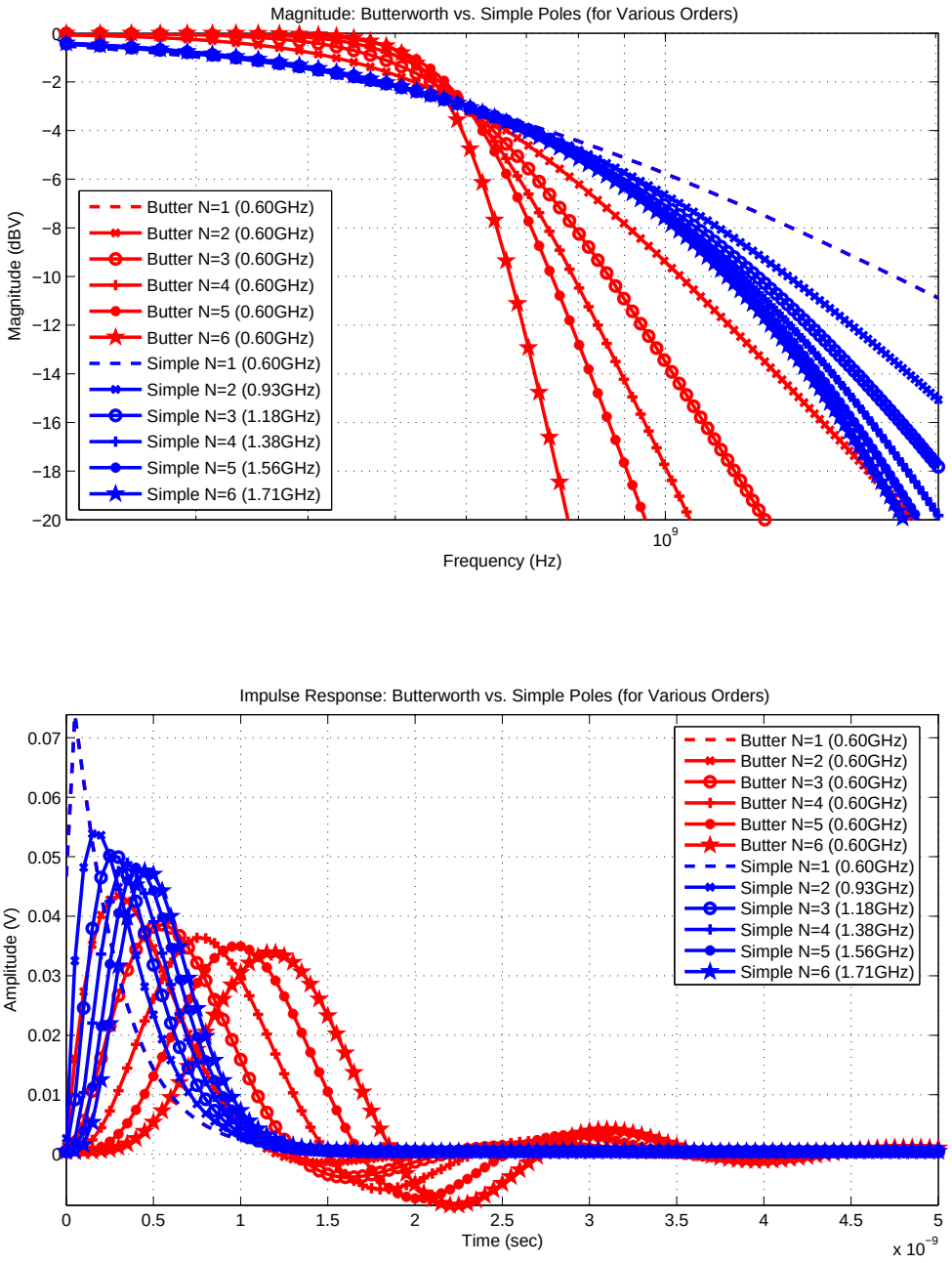

Figure 2.19: Simple Pole With Const. 3dB BW vs. Butterworth for Various Orders

spreads energy evenly over frequency, hence the ratio of the energy through the filters measures the energy attenuation, i.e. insertion loss, of the filter. The results indicate that the loss of concatenated, constant-pole-location filters relative to scaling poles for overall constant $-3dB$ BW is at most $-3dB$ for $N = 6$ order (as compared to an equivalent order Butterworth filter). Moreover, as scaled (constant $-3dB$ BW) concatenated poles attenuate $900MHz$ by $10dB$ less, the resulting interference power could be $7dB$ higher. Additionally, the expected power consumption of the overall constant bandwidth filter increases with order. If we assume that the capacitive load (C_{load}) is fixed for a stage, then as bandwidth increases, the effective R must decrease (to increase $1/RC$) and g_m (and hence the input current) must increase by a proportional amount. If we further assume that current scales directly with g_m (an optimistic assumption) then the overall power consumption for higher orders increases directly as the ratio of poles (+90% for $N = 6$). Note that if g_m scales less than linearly with current, the power consumption will be even larger. Therefore, we conclude that simple constant pole filter stages represent a better trade-off between insertion loss, interference attenuation and power consumption.

Note that if this front-end were to serve as a baseband for a higher frequency UWB system, it may not be desirable to have so much roll-off at the higher edge of the band. The bandwidth could be enhanced through the use of inductive peaking [91] to regain flatness without increasing the current consumption. (This will cost die area for inductors.)

Table 2.4: Comparison of Simple Filter Energy to Butterworth vs. Order

Energy	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
$\frac{P_{Simple\ \ fixed\ 900MHz}}{P_{Butterworth}}$	+1.46dB	+0.25dB	-0.70dB	-1.36dB	-1.87dB	-2.27dB
$\frac{P_{Simple\ \ fixed\ 3dB\ \ BW}}{P_{Butterworth}}$	0dB	+0.39dB	+0.41dB	+0.40dB	+0.38dB	+0.37dB

2.12 System Architecture

Now that the impulse ultra-wideband communication link has been examined and constrained from a system perspective, the question of architecture choice is still open. Conceptually we will be correlating the received waveform with a pulse template and the main decision is whether this operation should be performed in the analog or digital domain.

In an analog-based transceiver A/D conversion is typically placed after wideband gain, filtering, and high-speed analog correlation operations. This results in a slower (on the order of the symbol rate) A/D converter with higher resolution requirements. While the ADC power consumption is manageable, this requires more analog circuitry to operate at the full signal bandwidth. In particular, the correlation operation and template generation must be very high speed, which implies a trade-off between power consumption and template generation accuracy. For low power operation, a simple template is desired, such as a rectangular pulse. However, this template is inflexible, reducing system performance out

of proportion to the power savings gained. In addition, scaling up such an architecture for RAKE reception or faster acquisition places a large load at the critical, high-speed input to the correlators, thus increasing power consumption beyond linear scaling of the number of correlators.

Due to these issues and to take advantage of digital circuitry's flexibility, scalability, and ability to trade area for power consumption, the partitioning between analog and digital sections was chosen as close to the antenna as is feasible. A direct, time-based sampling approach was taken to avoid the need to design an integrated well-matched bank of filters in the analog front-end and to keep the digital backend simple. This has the effect of increasing the burden upon the A/D converter, possibly to the point where the ADC block consumes the most power in the system. However, as shown in section 2.8.1 in an interference-dominated environment very low resolution ADC's may be utilized thereby mitigating the power consumption penalty. A by-product of the flexibility of this architecture is that it also provides a platform for further experimentation. By moving the signal processing into the digital domain, it is easier to prototype different receiver approaches.

Figure 2.20 is a block diagram for the proposed "mostly digital" architecture. As the received energy is localized in time around the channel delay spread, the receiver only needs to operate during that relatively narrow time window. To meet the Nyquist criterion, this window must be sampled at a high rate, on the order of $2Gsample/s$ for one GHz of bandwidth. Reception consists of gain, matched to the antenna impedance, followed immediately by sampling and digitization. The digital samples are then fed to the digital backend for processing, e.g. acquisition, synchronization, and detection (see section 2.5).

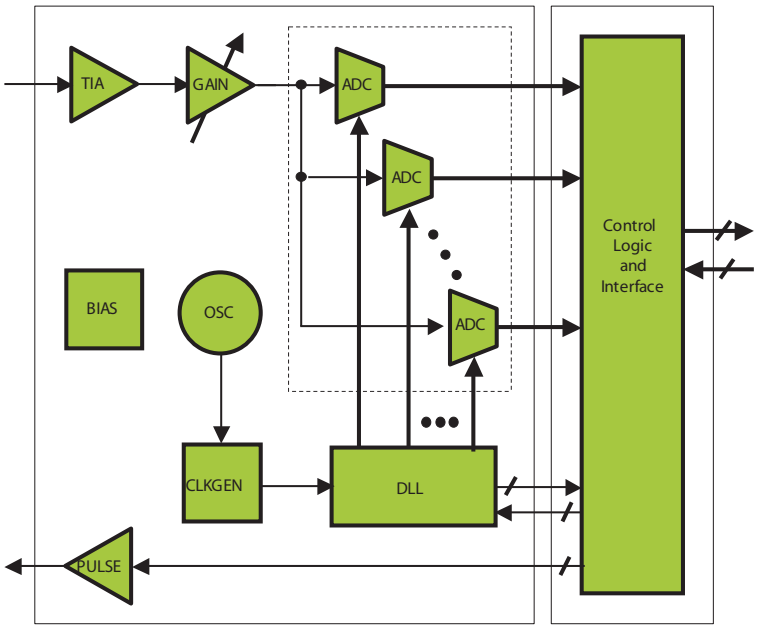

Figure 2.20: Proposed Analog Front-end Architecture

With the architecture selected and specified, we now move on to the circuit level design and power consumption optimization for the impulse-UWB transceiver.

Chapter 3

Circuit

The next step after architecture selection and specification is the mapping of circuit constraints into a CMOS implementation. Figure 3.1 depicts a more precise model of the system operation of pulse reception. In this case, a global clock is assumed and control of the sampling clock generation (from a delay locked loop), gain, and the A/D sampling intervals are derived from a counter which loops over a programmed range of values. Start and stop commands are generated by counter comparisons to independent, programmable registers to allow for flexible and arbitrary control of the blocks. Pulse transmission occurs at the end of the count length. Thus, by dynamically changing the counter length pulse-position modulation or time-hopping may be achieved.

In the following sections each major block from figure 2.20 is examined, designed, laid out, and measured results are reported (beyond those reported in [92] and [93]). On the receiver side the blocks are: the gain stages, global system oscillator and clock, delay locked loop, A/D converter, and digital control logic. On the transmitter side, the blocks

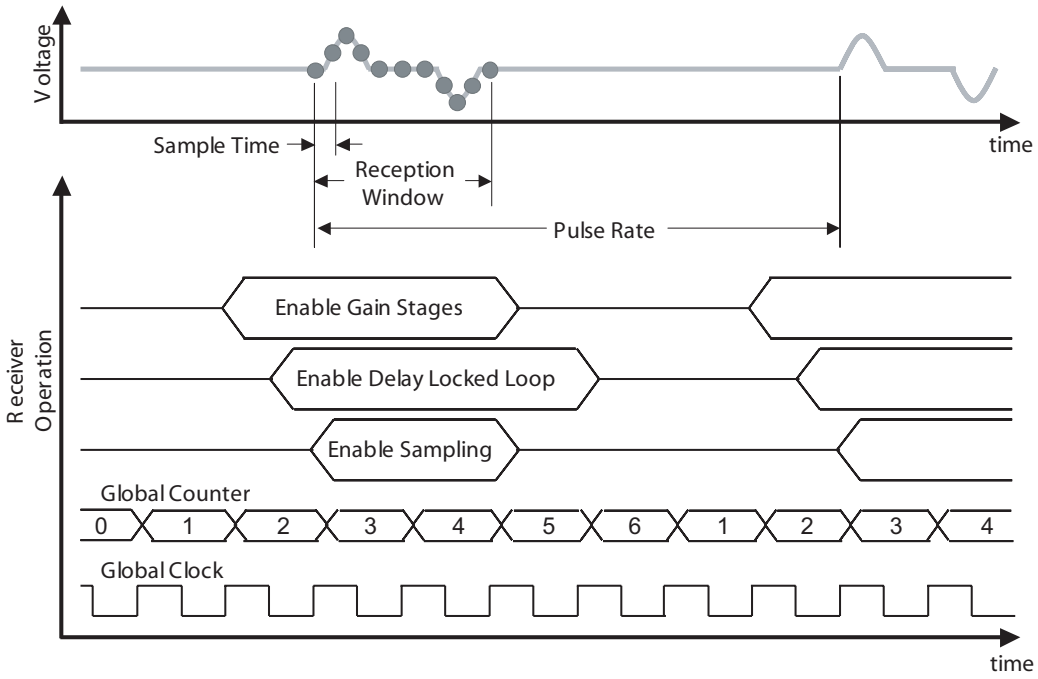

Figure 3.1: Block-Level Sampling Operation

utilize the same global clock and control, and include a differential pulse generator. Finally the overall chip floorplan and die photo are shown.

3.1 Receiver Gain Stages

Derived from the system specification, the necessary gain functionality includes $10dB$ to $50dB$ of variable gain, filtering of $-3dB$ per-stage bandwidth around $900MHz$ (for 5 to 6 concatenated stages total), and a noise figure less than $20dB$. We also require the aggregate offset seen at the A/D input to have less than $10mV$ standard deviation. This implies offset cancellation on the order of a millivolt. Because duty-cycling the gain stages between receive pulses is a priority, this offset cancellation must be robust to duty-cycled

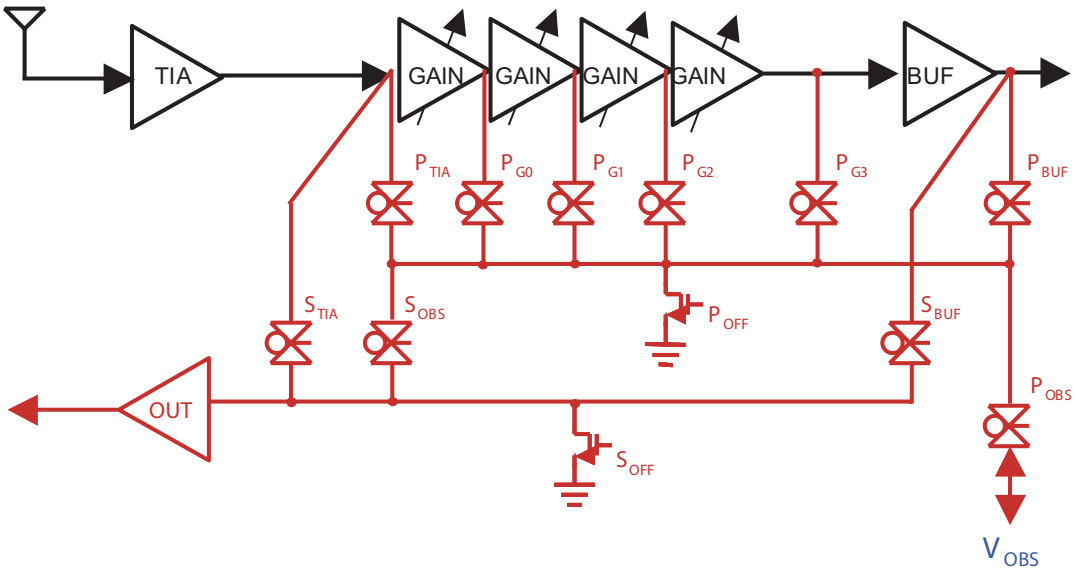

Figure 3.2: Single-Ended Gain Stage Diagram

operation. Soft, memoryless limiting of the gain stages is also of interest since large input interference signals are expected. Additionally, to accommodate different antenna choices a programmable input impedance is desired; from "low" to "high", encompassing a 50Ω single-ended value for easier test and measurement. Finally, circuit operation was designed at V_{DD} of $1.0V$ worst-case

Figure 3.2 shows the single-ended receiver gain stages along with test/measurement circuitry. The signal from the antenna is connected to the differential input of the transimpedance amplifier (TIA) whose primary task is to match impedance while providing reasonable gain. As the noise figure requirements are very relaxed, this block is not called a "low noise amplifier" (LNA). The noise figure is usually dominated by the first stage, assuming there is a reasonable amount of gain provided. Following the TIA are a series of variable-gain stages. The remaining gain was partitioned over the blocks resulting in four

stages. Finally a wideband buffer drives the ADC input capacitance. Also shown are test and debug circuitry: a 50Ω output driver, pass transistors, and an input V_{OBS} pin which will be discussed in section 3.1.5.

The gain stages are all implemented differentially as we were worried about substrate and supply noise coupling. Even though this increases the overall power consumption, it vastly reduces the possibility of self-interference (or worse, self-reception) due to coupling from the control logic or transmitter (if it is operating concomitantly). As the digital logic has gates that switch in relation to the spreading code, the possibility of coupling may cause the receiver to falsely lock onto its own control logic. With differential circuits, this noise appears as common-mode and is vastly attenuated relative to the differential gain. To reduce injected noise care was also taken with layout, substrate taps, power supply decoupling, and floorplanning.

Because the ADC is only 1-bit, no automatic gain control (AGC) loop is required. The input signal may be railed (or allowed to saturate) as long as recovery can occur fast enough to track the input signal bandwidth. This also simplifies the comparator, allowing for a relatively higher offset to be allowed (as discussed in section 2.10).

The test circuitry in the gain stages allows measurements to be take over the whole chain, as well as for the TIA and output buffer individually. This allows us to be able to separate the TIA and variable-gain stage gain from each other and the buffer stage. Measurements for the whole gain chain are shown in the following figures. Figure 3.3 shows the total measured S21 gain for the gain chain (neglecting the output buffer) over various bias conditions from minimum to maximum gain. Roughly $0dB$ to $42dB$ of gain is available

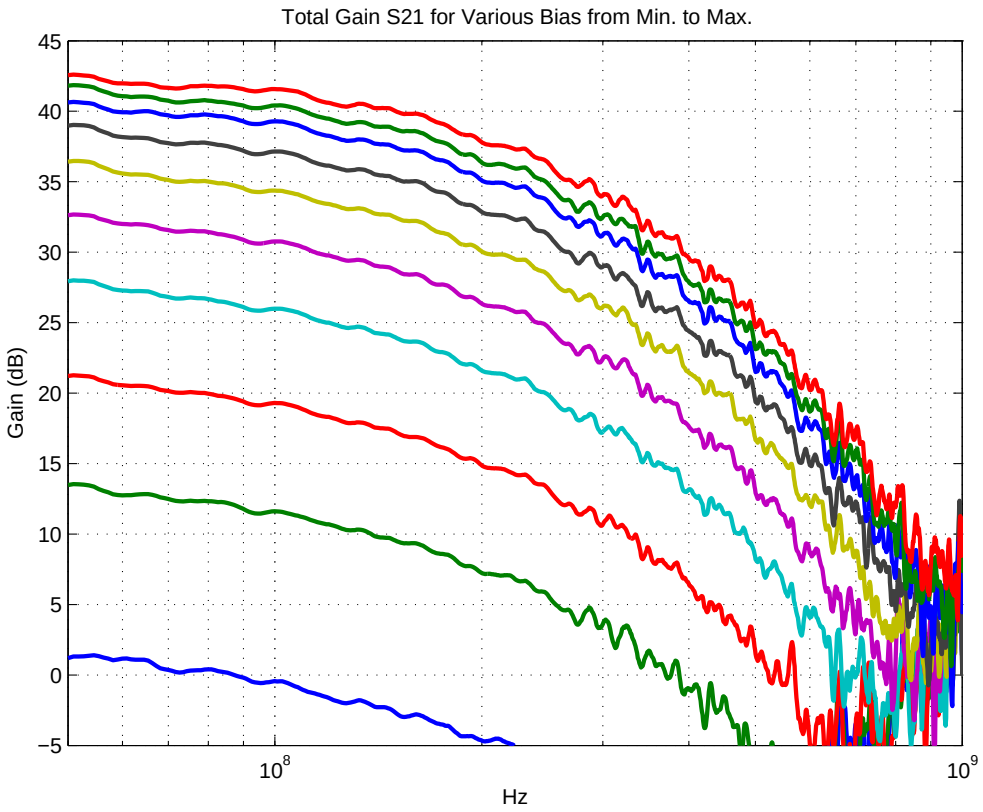

Figure 3.3: Gain Stage S-Parameters

and the bandwidth shape matches the expectation from section 2.11 due to concatenated simple pole stages. Figure 3.4 shows the noise figure, measured at clean frequencies. The values is approximately $12dB$, well within the $20dB$ design target. Due to a lack of shielding in our lab, measurements had to be taken at frequencies where large interferers were not present. (Otherwise the interference would be misinterpreted as additional noise and corrupt the measurement.) Finally figure 3.5 compares the maximum gain measured during noise figure measurements with the S21 from S-parameter measurements (using the network analyzer). There is good agreement.

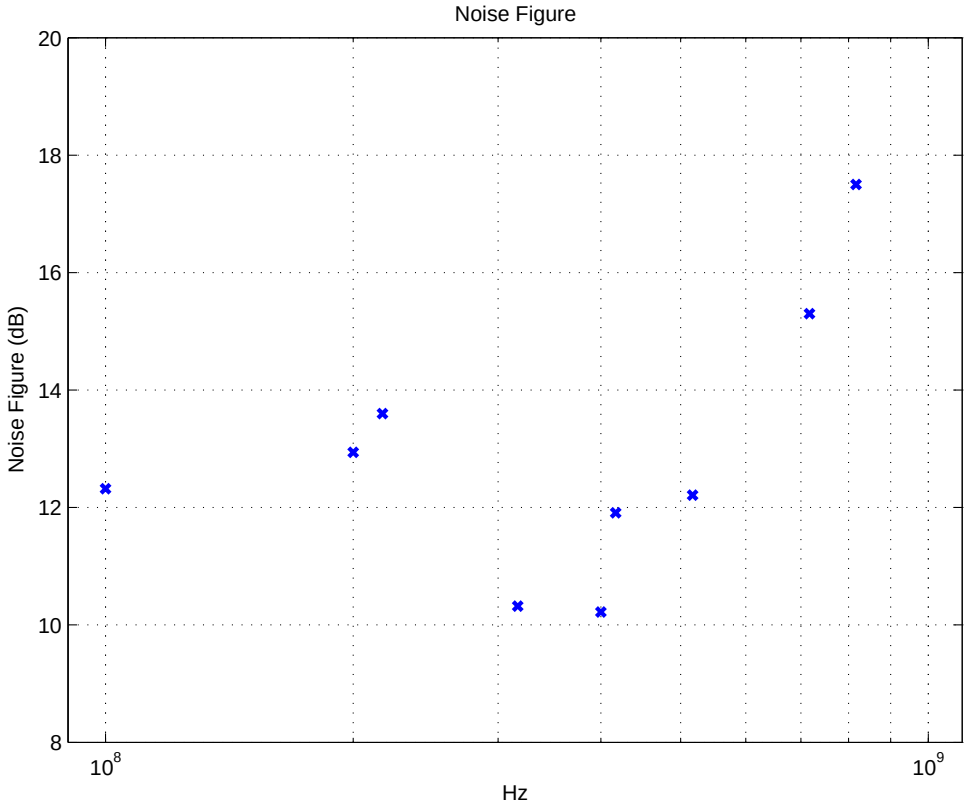

Figure 3.4: Noise Figure for Gain Stages

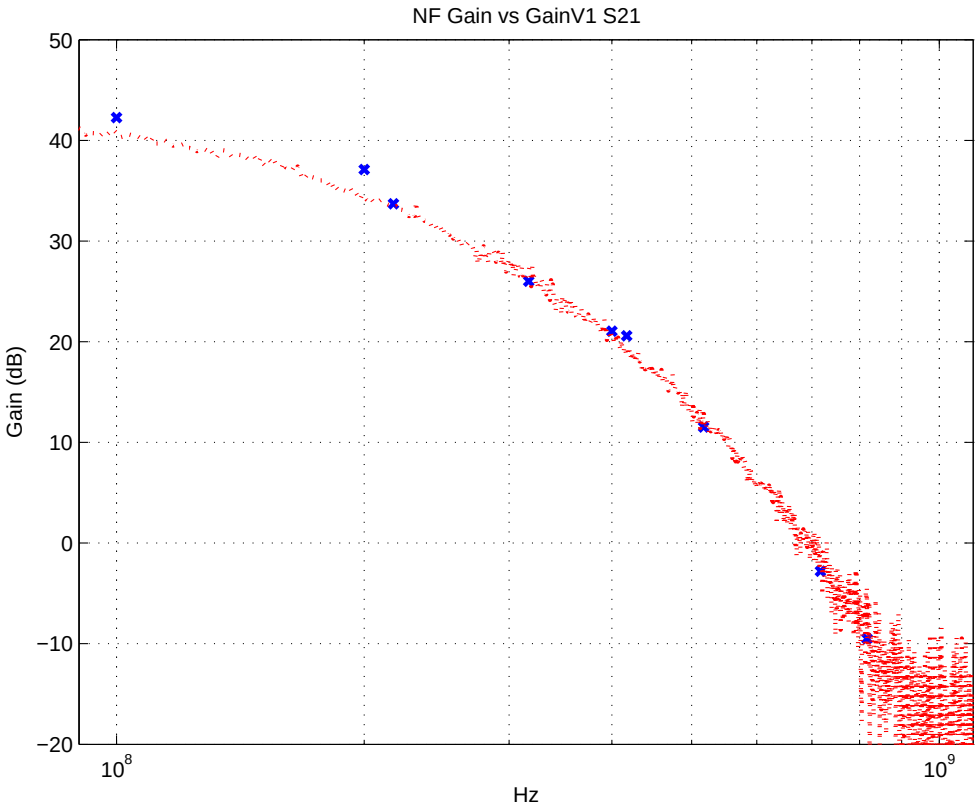

Figure 3.5: Gain Stage S21 vs. Gain from Noise Figure Meter

3.1.1 Transimpedance Amplifier

The main task of the input amplifier is to match impedance to the antenna and provide some reasonable amount of gain. The noise figure requirements are so lax that they do not constrain the design much. The difficult aspect of design becomes the production of a relatively low impedance (e.g. 50Ω) without expending a large amount of power. To illustrate this difficulty, one can calculate the current necessary for a source-follower to produce a 50Ω input. Assuming saturation operation for a submicron CMOS device (i.e. g_m/I_{bias} of 10), then for $1/g_m$ of 50Ω, an I_{bias} of $2mA$ is needed. For a differential circuit, this is $4mA$ total. The current from the input stage alone surpasses the total allowed power budget for the entire transceiver!

Figure 3.6 illustrates four commonly used circuit topologies to achieve a low input impedance that may be found in popular circuit textbooks, such as [94] [95] [96] [97]. To evaluate these options, the current consumption, input impedance, gain and noise figure were calculated for each topology.

Although topology A is straight-forward, it is usually avoided due to the penalty in noise figure. For our case this is not the dominant problem. Rather, topology A was avoided due to the difficulty in producing a variable input impedance. (Although this is, admittedly, not an unsolvable problem.) Likewise, topology C was avoided due to difficulty in elegantly scaling the feedback impedance and to concerns about duty-cycling behavior (the feedback resistor sets the input voltage level which could have a large capacitance). Topology B was avoided due to excess current consumption (as illustrated in our example). Topology D was chosen because it allows the input impedance to be simply controlled through the

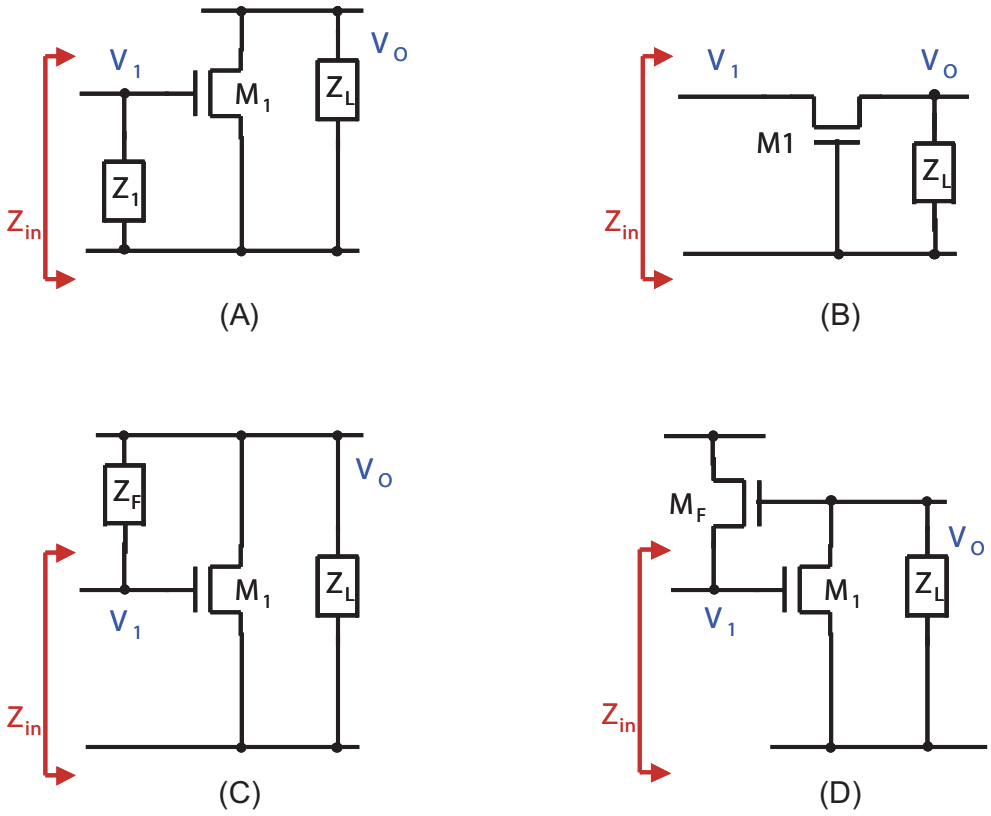

Figure 3.6: Transimpedance Amplifier Topologies

bias of M_F and it seemed more agreeable to duty-cycled operation. The input impedance is approximately $1/\left(A_V * g_m\right)$ which allows for a savings of A_V in current necessary to produce the input impedance (at an expense of increasing the output noise by an equivalent amount).

The transimpedance circuit diagram is shown in figure 3.7. To accommodate duty-cycling, switches (driven by "ON") were added to turn on and off the bias source, and to isolate M1 from the input capacitance to avoid long recovery transients. The TIA bias, shown in figure 3.8 is derived from global V_{BIASP} and V_{BIASN} voltages using a current digital to analog converter (DAC). I_1 directly selects a bias DAC current value set by the global bias current unit value (referred to as the least significant bit or "lsb" value; section

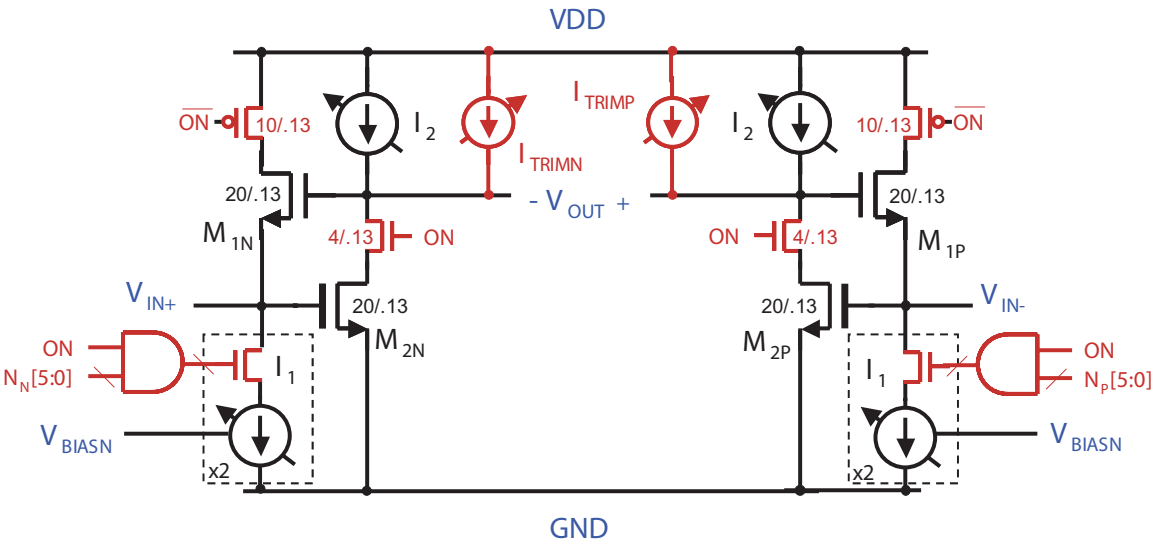

Figure 3.7: Transimpedance Amplifier Circuit

3.1.6 details the global bias circuitry). I_2 uses two control bits to select a couple of units of bias current from V_{BIASP}. These are multiplied and mirrored to produce [1 to 31]*10*[1 to 3]/8 units of bias for I_2. Trim is accomplished in a similar manner, shown in figure 3.9. Again a couple units of global bias are divided by 8, and then mirrored separately to each differential output ($\times 2$) to correct for offset. Note that offset trim is monotonic, and allows for a wide range of operation (up to 20% trim).

Figure 3.10 shows the measured TIA impedance magnitude range (derived from S11) over maximum and minimum bias for I_1 and I_2. We expect the overall minimum Z_{IN} to occur for maximum I_1 and I_2 which was measured to be 35Ω. The maximum impedance (minimum I_1 and I_2) was measured to be approximately 175Ω, although a peak of 250Ω was observed around $100MHz$. Due to shunting from the input capacitance, the maximum impedance rolls off early within the band. The input bias used for system operation is

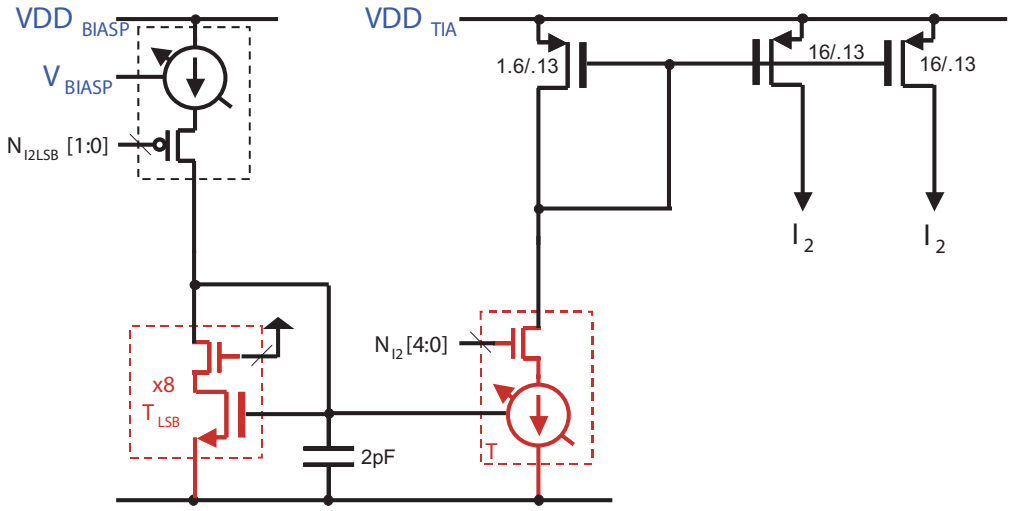

Figure 3.8: Transimpedance Amplifier Bias Circuit

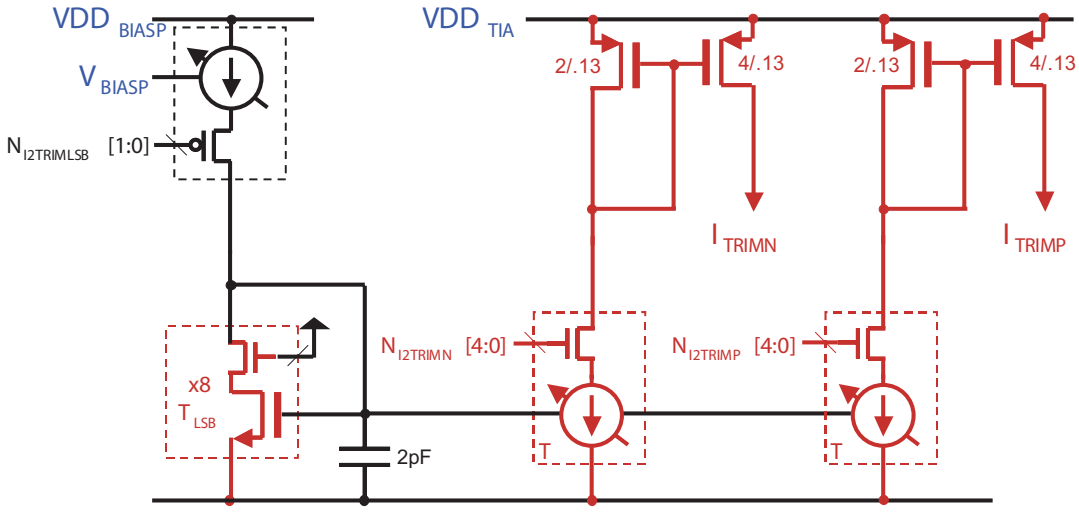

Figure 3.9: Transimpedance Amplifier Trim Circuit

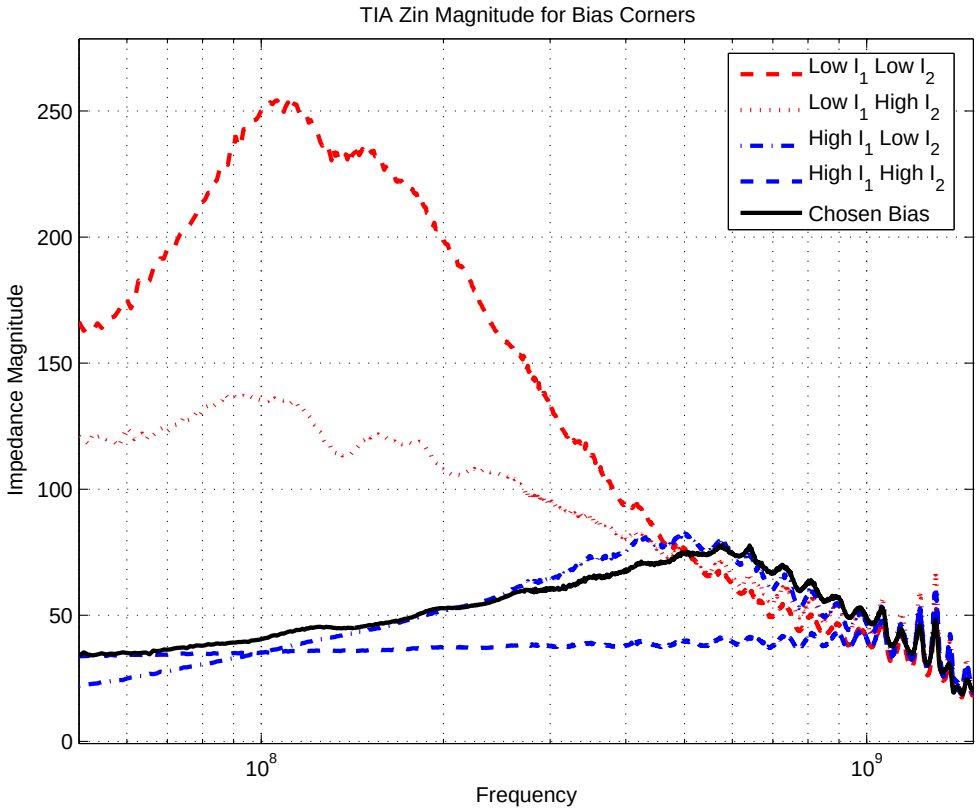

Figure 3.10: Transimpedance Amplifier Input Impedance Magnitude

also indicated ("chosen bias"), providing $< -10dB$ S11 over $600MHz$ of bandwidth seen in figure 3.12. Figure 3.12 also indicates the TIA S21 gain, measured at approximately $10dB$ over the bandwidth. S12 was measured at $-40dB$ or less over the entire band. The input impedance is also shown in figure 3.11 as a complex value. Note that the feedback topology can appear inductive for low frequencies (likely resulting in the peak seen for the maximum impedance).

The TIA trim (I_{TRIMP} and I_{TRIMN}) were measured over the trim code range. Over $\pm 20mV$ of monotonic trim was observed with a step-size less than $1mV$.

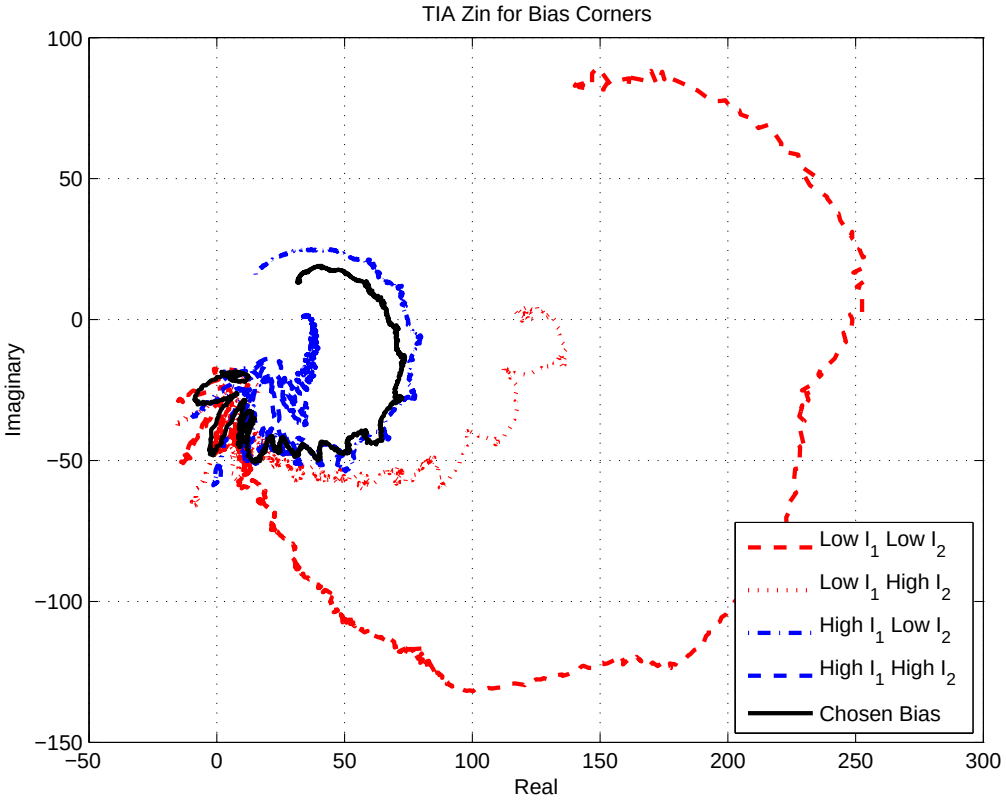

Figure 3.11: Transimpedance Amplifier Input Impedance Real vs. Imaginary

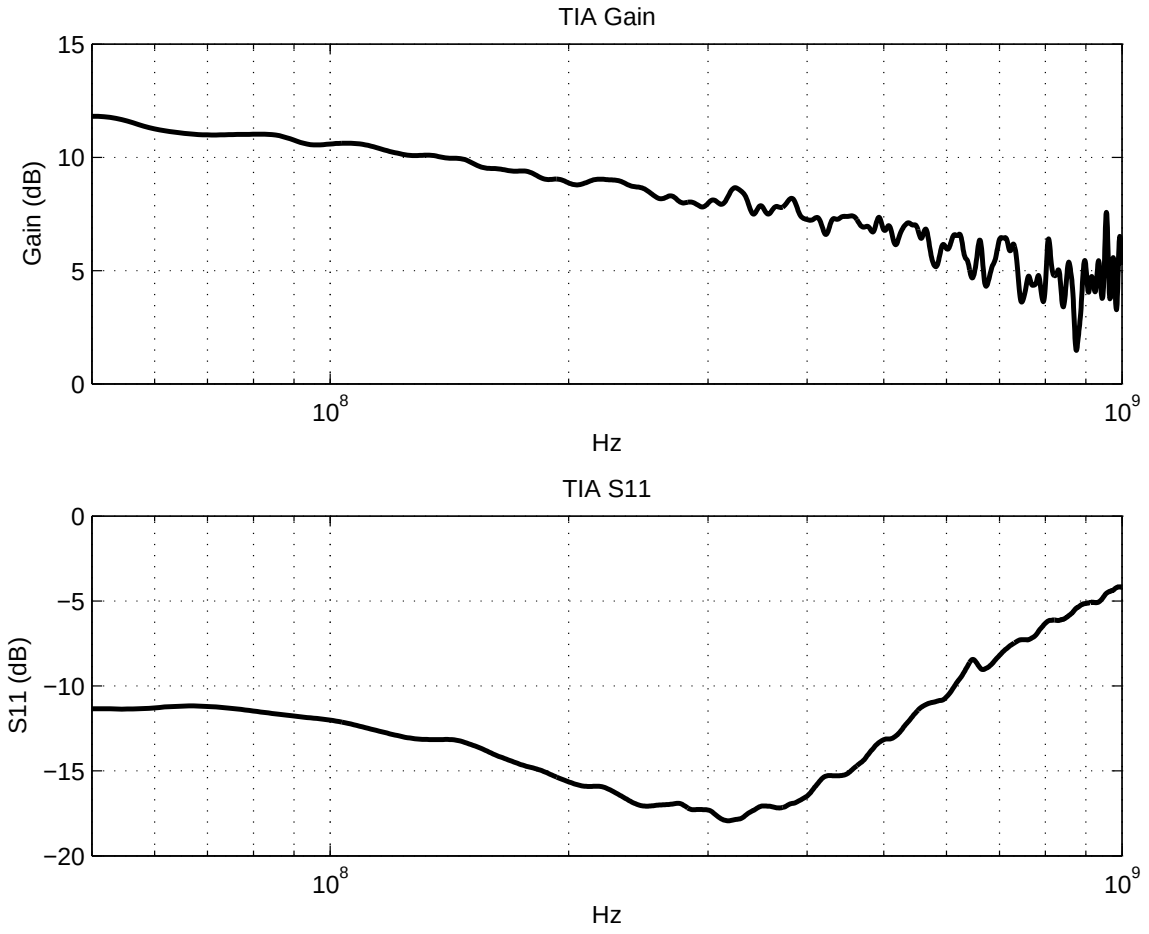

Figure 3.12: Transimpedance Amplifier S-Parameters

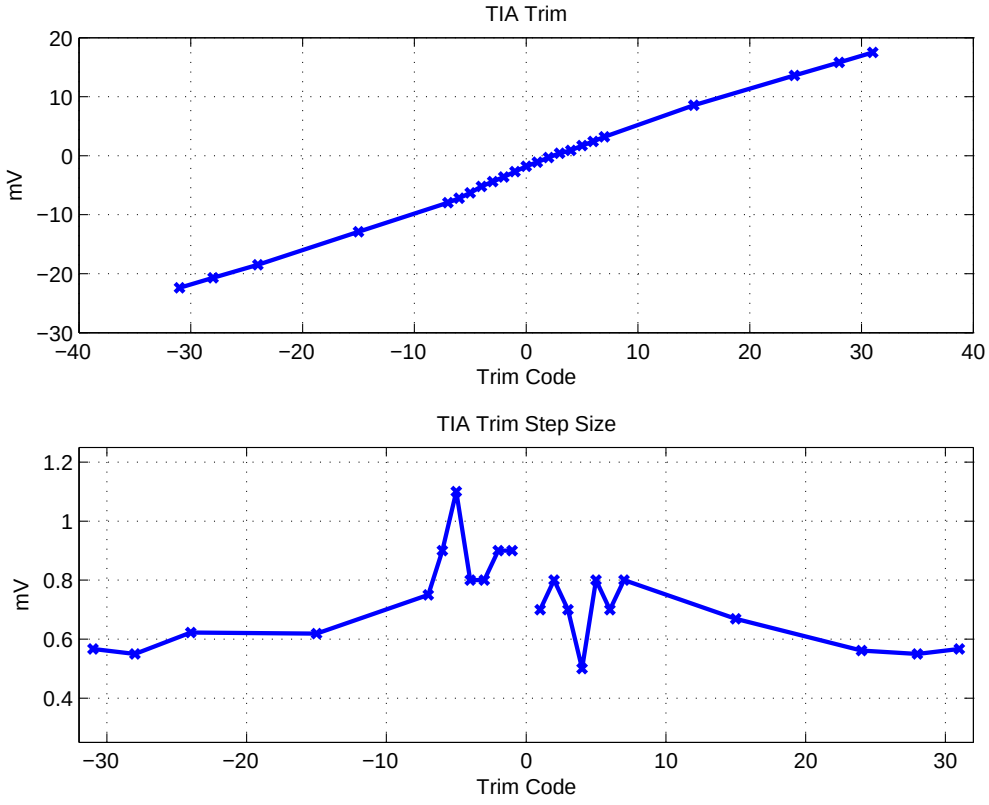

Figure 3.13: Transimpedance Amplifier VOS Trim

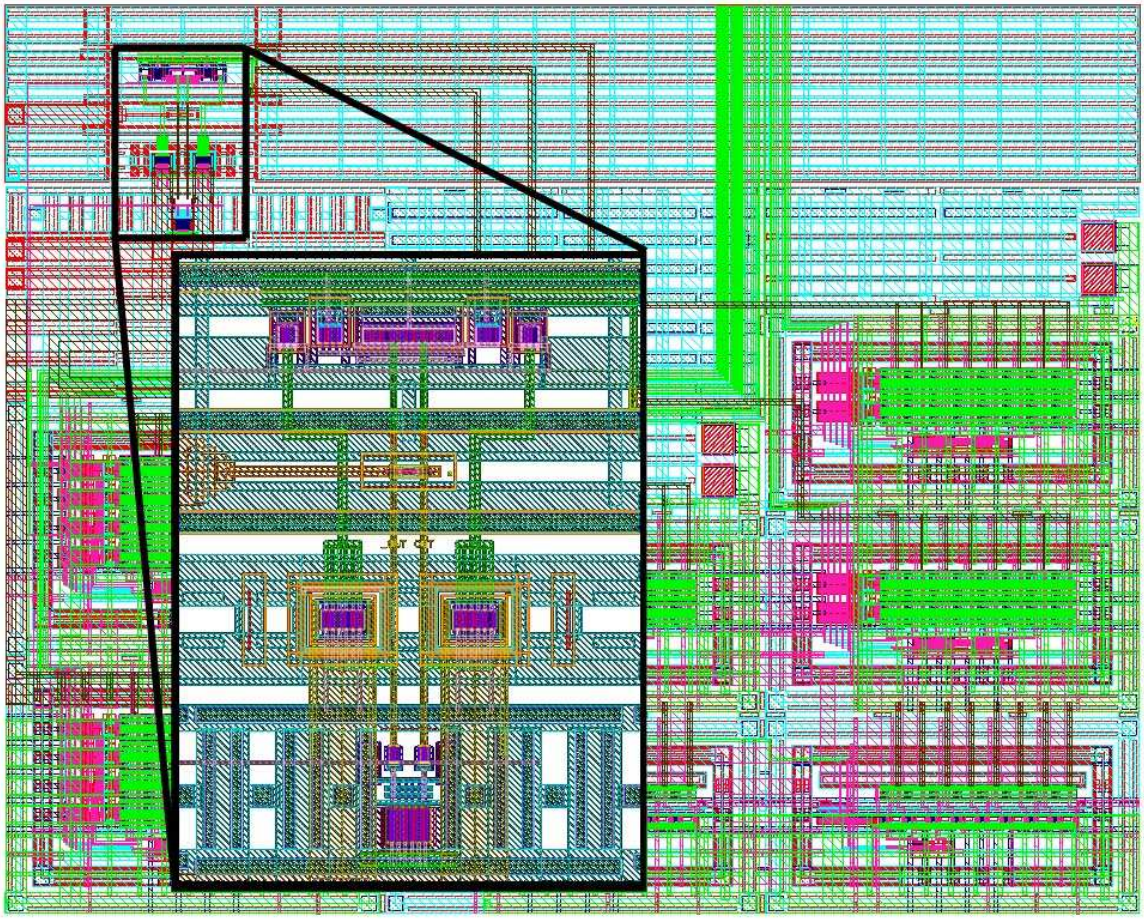

Figure 3.14: Transimpedance Amplifier Layout

A final plot of the TIA layout is shown in figure 3.14. The large blocks at the bottom of the layout are the bias (and trim) current DAC's. Inset is a close-up of the gain stage layout. For matching, layout was kept translation and mirror symmetric for at least 10 to 20 microns around sensitive devices through the use of dummy structures.

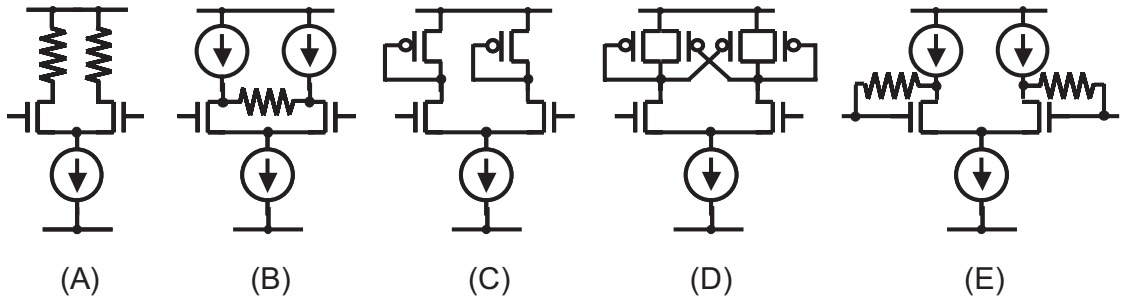

Figure 3.15: Variable Gain Amplifier Topologies

3.1.2 Intermediate Gain Stages

The task of the intermediate gain stages is to provide variable gain (on the order of $0dB$ to $40dB$) with a per-stage $-3dB$ bandwidth of $900MHz$ in the minimum current possible. Additionally we need the variable gain stages to be duty-cycle-able (including any offset cancellation or common-mode feedback (CMFB)), have graceful limiting (fixed maximum V_{swing}, memoryless recovery), relatively linear gain (preserve zero-crossings), and have low noise generation (constant current and have a high power supply rejection ratio (PSRR)).

From these requirements, a list of candidate topologies, shown in figure 3.15, was generated. In general, common-mode feedback implied more complexity and hence was avoided. Architecture B was rejected because it required CMFB, the dc bias headroom limited the total gain and swing. Architecture C seemed better, but maximum gain turns out to be very limited. Taking a g_m/I_{bias} perspective, without going into subthreshold[98] (roughly $g_m/I_{bias} > 15$ which would imply large devices and hence parasitics) or triode (roughly $g_m/I_{bias} < 8$ which implies a large voltage headroom), the gain of g_{mn}/g_{mp} is

practically limited to 2 or less. Gain could be increased by current stealing from the load, but that would increase the parasitic capacitance. Trimming this load could be achieved with current stealing again. Finally, the R_{ds} of the load is not very linear over the swing and hard-limiting might send spikes into the supply. Architecture D attempts to solve the gain problem by adding positive feedback (kept under 1/2 the total resistance to maintain stability). However, parasitics are increased again, it is still possible to spike the supply, and linearity suffers. Architecture E uses feedback to bias the common-mode problem with architecture B, but winds up loading the previous stage output, as the input impedance is no longer large (i.e. the gate input of a MOSFET). This limits the gain for this architecture. After examining many variants, architecture A seemed the best suited. There is no CMFB necessary, linearity is good, gain may be controlled through the bias current, offset trimming may occur by trimming the resistor load, and parasitics are not large. Gain is still limited, in fact it is limited to:

$$A_V = \frac{g_m}{I_{ds}} * \frac{V_{swing}}{2} \tag{3.1}$$

Where V_{swing} is dictated by the allowable bias headroom from V_{DD} through V_{GS} to the load stage's current source V_{sat}. A nice feature of this topology is that, as I_{bias} is scaled, the gain is also scaled, as is the slew rate. At a particular I_{bias} we would have a maximum swing of:

$$V_{swing} = I_{bias} * R_L = 2 * I_{ds} * R_L \tag{3.2}$$

The maximum slew rate is given by I/C which would be:

$$SlewRate = \frac{I_{bias}}{C_L} \tag{3.3}$$

An input sinusoid of maximum amplitude at the $-3dB$ bandwidth edge has maximum edge

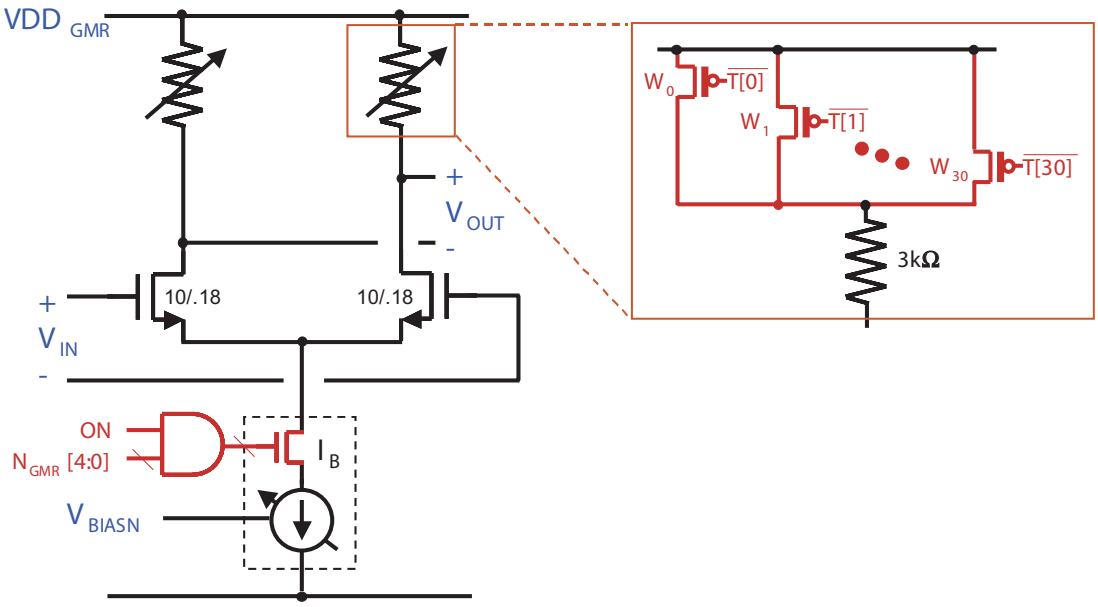

Figure 3.16: Variable Gain Amplifier Circuit Diagram

rate equal to:

$$\frac{dV}{dt} = \frac{1}{R_L * C_L} * V_{swing} \tag{3.4}$$

As $V_{swing} = I_{bias} * R_L$, we can rewrite the above equation as:

$$\frac{dV}{dt} = \frac{I_{bias}}{C_L} \tag{3.5}$$

which matches the available slew rate. (Thus frequencies above the $-3dB$ will begin slewing in addition to being attenuated throughout the gain stages.) Hence the $g_m * R$ architecture allows for variable gain by setting the bias current. This is nice as it allows the transceiver to conserve current if full gain is not needed. Also, it avoids the need to use more complicated techniques to achieve variable gain, i.e. switches to bypass gain stages (runs into bandwidth problems), or $\frac{1+x}{1-x}$ circuits (which generally require more current).

The variable gain circuit used is shown in figure 3.16. Bias is set by a digital to

analog converter (DAC) current source driven from a global bias (see section 3.1.6) which operates down to approximately $0.4V$. This limits the per-stage swing to approximately $0.85V \pm 0.25V$ for $V_{DD} = 1.1V$. The number of variable gain stages was set to four based on an optimization between the number of stages, total current consumption for fixed gain, and offset cancellation complexity. The optimization pushed towards more stages (with less gain per stage), but the curve was rather flat, implying little additional improvement in exchange for more stages to trim, so the knee of the curve, four, was chosen. Figure 3.17 shows the measured S21 for the variable gain stages under different bias conditions from minimum gain to maximum gain. Figure 3.18 shows the measured trim for a variable gain stage. Offset is trimmed by varying the load resistors on one leg. The load resistor on each leg is in series with 31 parallel PMOS devices, sized to give monotonic, roughly linear trim when activated (triode). These PMOS devices are capable of trimming up to 10% of the load resistor value. Section 3.1.4 covers the trimming operation in more detail. Finally, a screen capture of the layout of a variable gain stage is shown in figure 3.19.

3.1.3 ADC Buffer

The input to the ADC has a rather large capacitive value. Even though only 5 sampling capacitors may be seen at one time (see section 3.4.3 for more information on the ADC operation), there is a large amount of parasitic capacitance from routing and the 32 input switches. Because of this, we can't use exactly the same circuit as we used for the variable gain stages. We can use the same topology by reducing the load resistor, but this places a burden on the offset trimming. This will be discussed more thoroughly in section 3.1.4, but the problem with using resistor trim is that the parasitic capacitance from 31

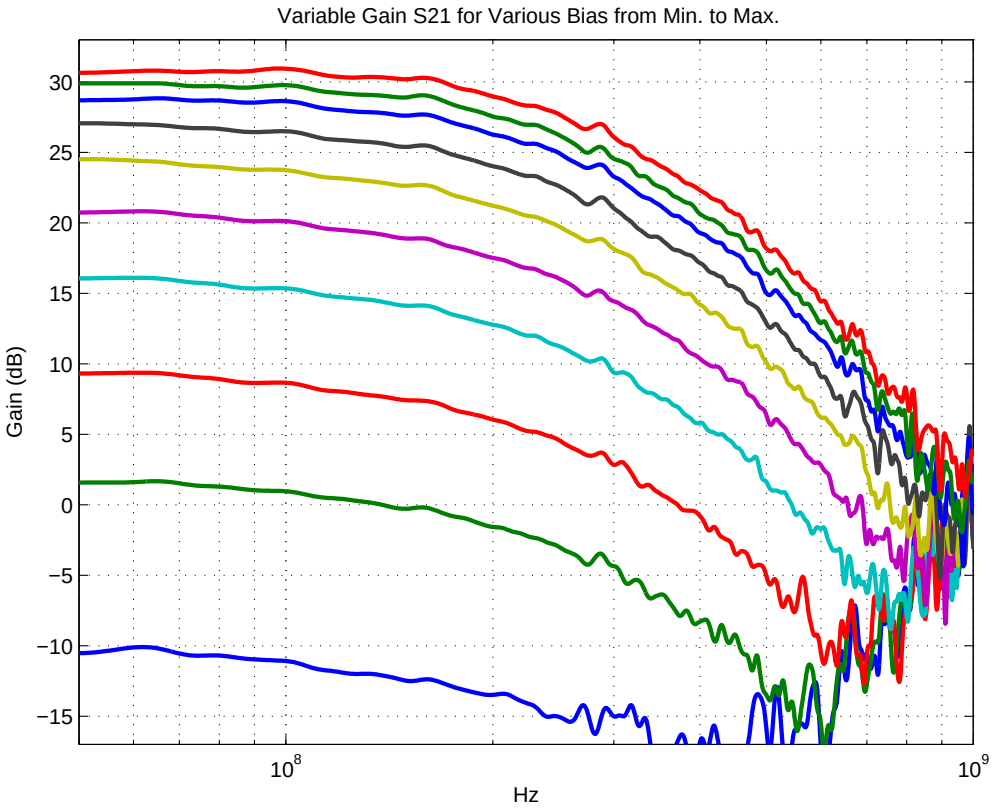

Figure 3.17: Variable Gain Stage Measurements: S-Parameters

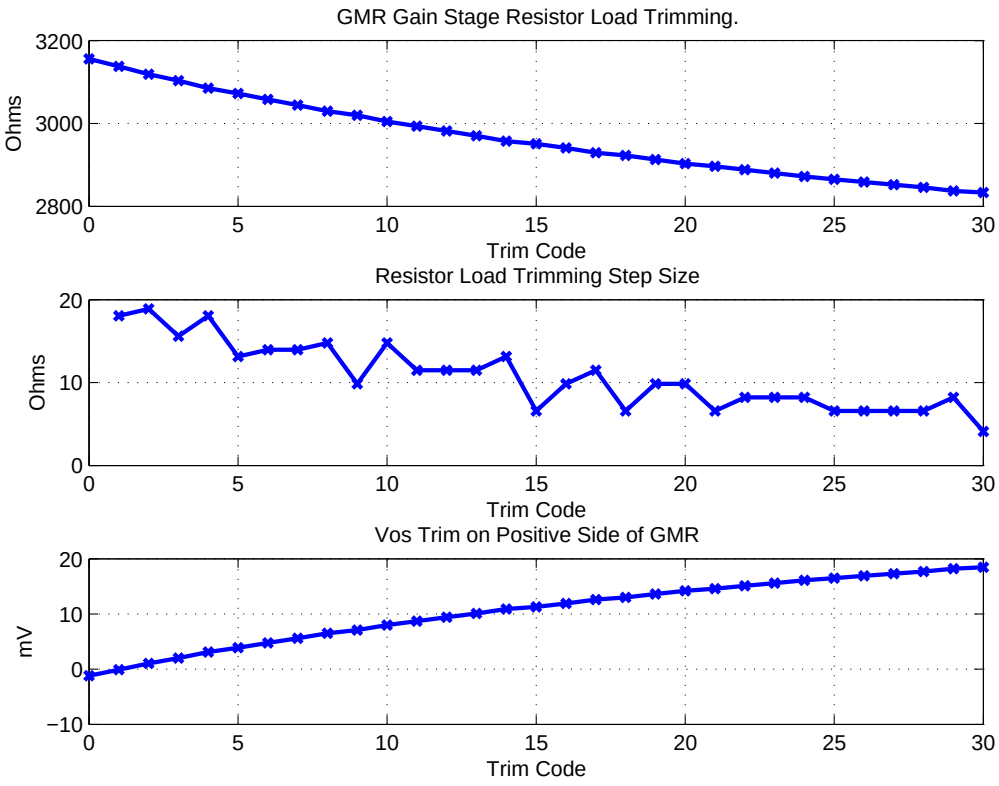

Figure 3.18: Variable Gain Stage Offset Cancellation Measurements

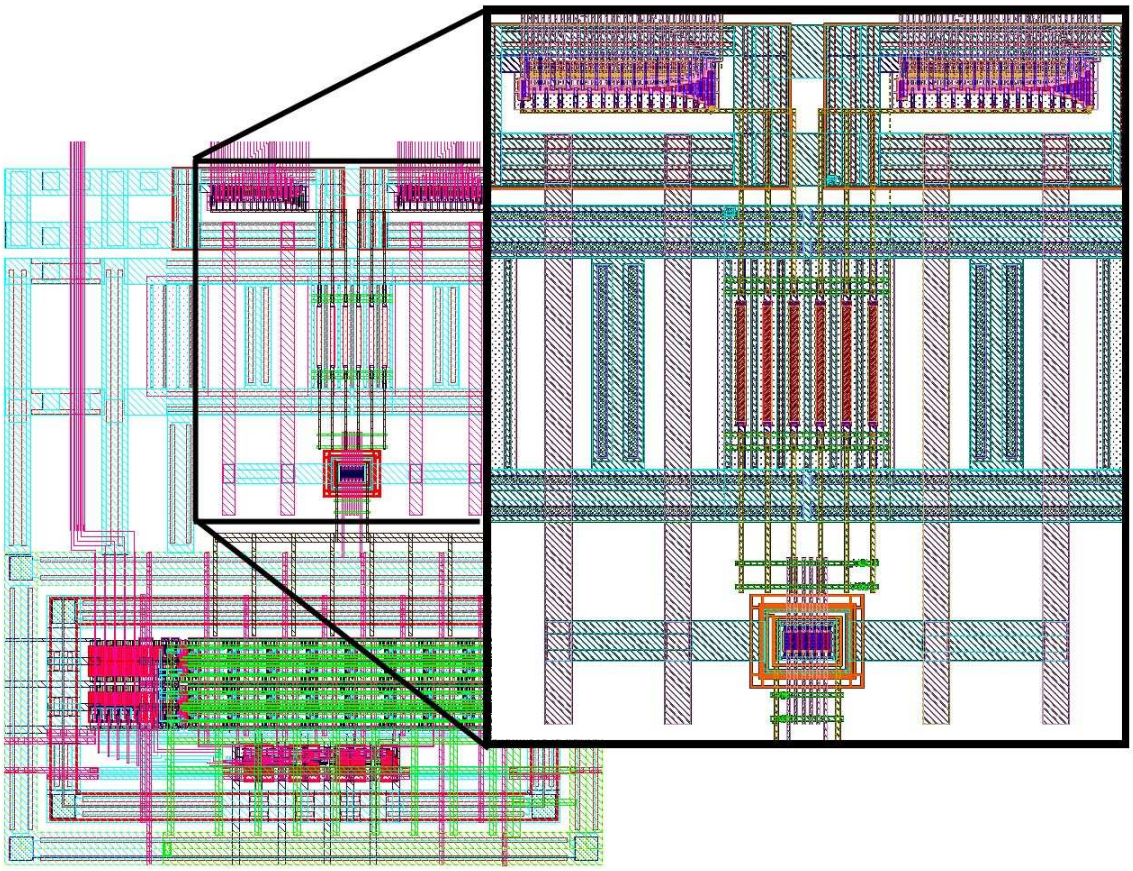

Figure 3.19: Variable Gain Stage Layout

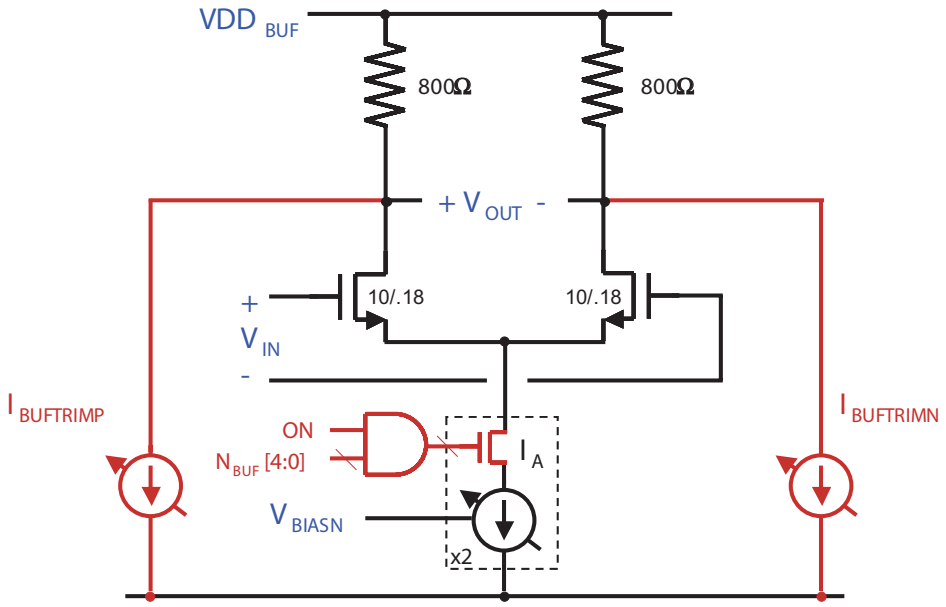

Figure 3.20: ADC Buffer Circuit

PMOS devices corrupts the frequency response. Thus, to trim this stage, current is stolen (or leaked) off of one leg to equalize the output voltage. This does increase the parasitic capacitance at the output, but capacitance is already large on this node, so the increase is not significant. A diagram of the circuit is shown in figure 3.20 and the offset trim circuit is shown in figure 3.21.

The ADC buffer was configured to have a gain of approximately unity. A small output impedance was desired to maintain bandwidth and to reduce the common-mode perturbation from charge kickback from the ADC sampling switches. To trim the output stage, a small amount of current is leaked from one leg (or the other). The trim range is shown in figure 3.22 and spans $\pm 35mV$ with a $1.25mV$ stepsize. A screen capture of the layout is shown in figure 3.23. The four large boxes at the bottom are the bias DAC current source, two trim DAC sources, and trim mirror DAC. On top, dummy resistors and devices

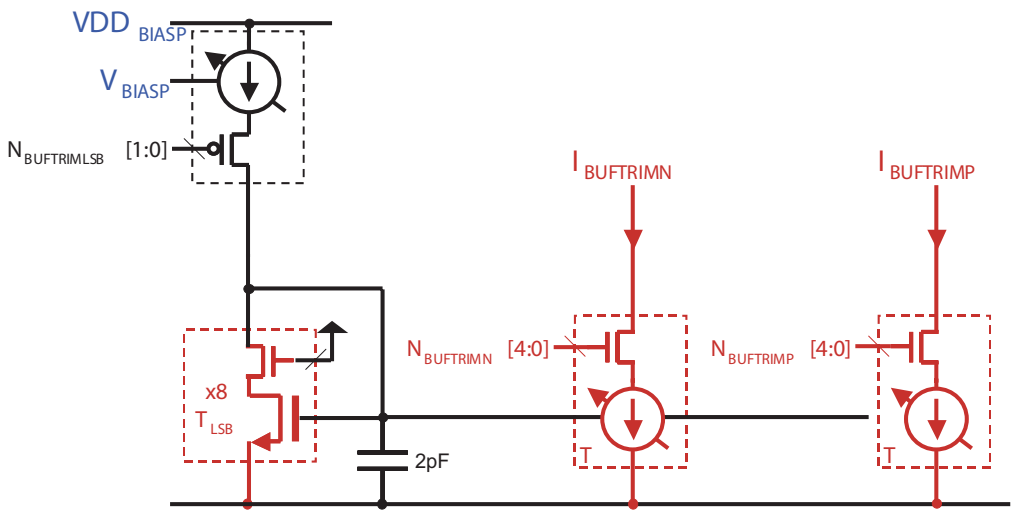

Figure 3.21: ADC Buffer VOS Trim Circuit

are used to maintain matching.

3.1.4 Offset trim

The main challenge of duty-cycled operation is offset trimming. The gain stages themselves are wideband, and hence have small time constants, which easily settle within a clock cycle when duty-cycled. The problem is that, due to mismatch, they may settle to non-zero values which cause the last stage amplifier to rail out. One typical method of offset cancellation employs low frequency (i.e. near *dc*) feedback loops which cancel the offset (as shown in figure 3.24). This method could be amendable to duty-cycled operation if the feedback poles were low enough to filter out the switching transients to guarantee convergence. Given the wide range of possible pulse rates (from the order of magnitude of $100MHz$ down to below $1MHz$), the filter would have to be very low frequency. This limits the loop settling behavior to long time constants, which run counter to the desire to

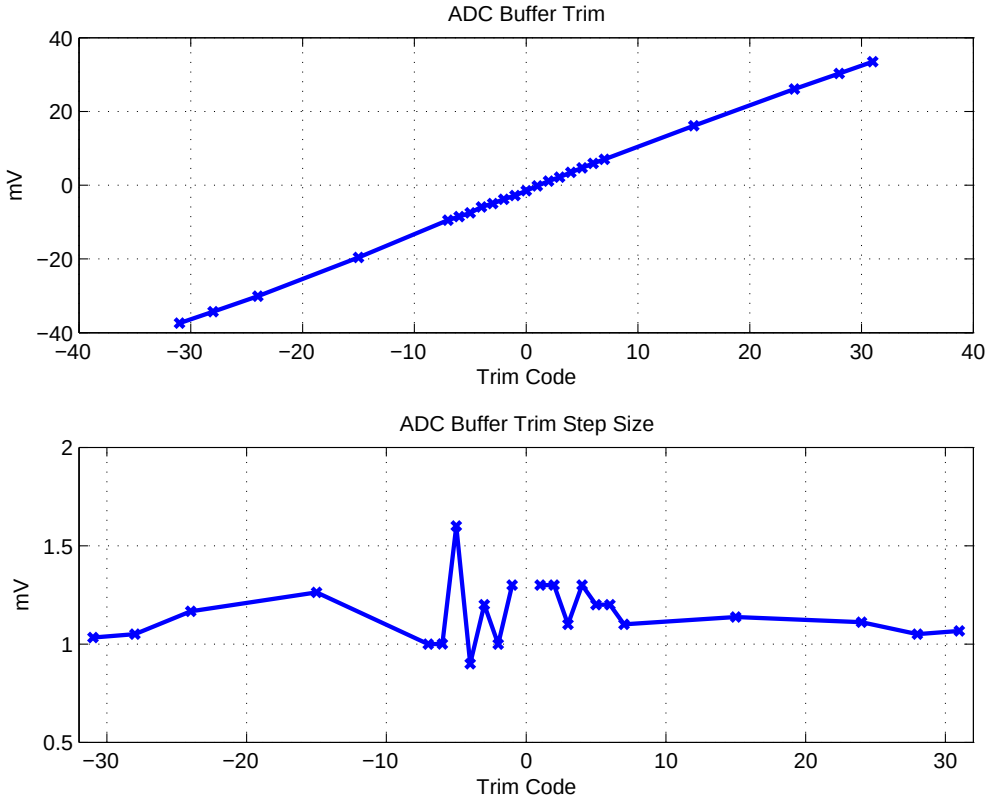

Figure 3.22: ADC Buffer VOS Measurements

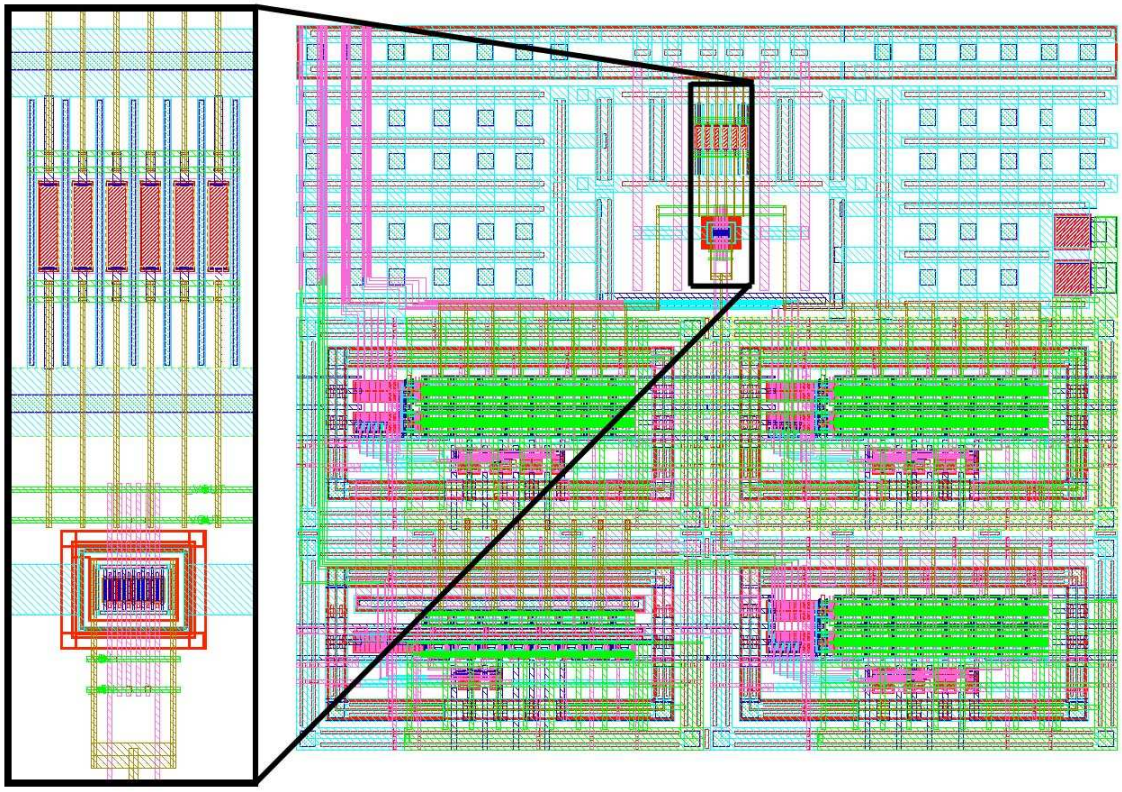

Figure 3.23: ADC Buffer Circuit Layout

have fast duty-cycled operation. Another typical method has each gain stage sample and store its own offset. This was avoided due to the extra power consumption and complexity. (The feedback loop needs to be closed with enough gain to properly cancel the offset and each gain stage has at most a gain of only 3 to 4.) One way to avoid offset cancellation is to utilize capacitive coupling between the gain stages. An example of this is shown in figure 3.25. Note that the coupling capacitor forms a capacitive divider with the next stage input and hence attenuates the signal from the previous output stage to the input of the next stage. The concatenation of 5 or more capacitive dividers can result in attenuation on the order of 1/2 unless the coupling cap is very large and the parasitic is very small. The loss of 1/2 implies another gain stage would need to be added to compensate for the use capacitive coupling, resulting in a power supply increase on the order of 20% for the gain stages. If larger coupling capacitors are used, there is an area penalty, and a limit to the capacitive divider loss set by the top and bottom plate parasitic capacitances. The conclusion was that it would be tough to make this approach work without using a low parasitic (on the order of a couple percent) MIMCAP. (The cost is area: each MIMCAP was estimated to be roughly as large as a pad, and we would need $2 * N_{stages}$ of them.)

The approach taken to cancel offset is a feedforward combination of load trimming and current stealing. Each stage has a monotonic, digital circuit structure to either trim the resistive load or pull current from the output node by at least $\pm 20mV$ in roughly $1mV$ steps. Because the method is feedforward, it doesn't require long settling each time the gain stages are duty-cycled. Rather, a specific calibration operation is required prior to operation (although this may be performed infrequently as a function of temperature

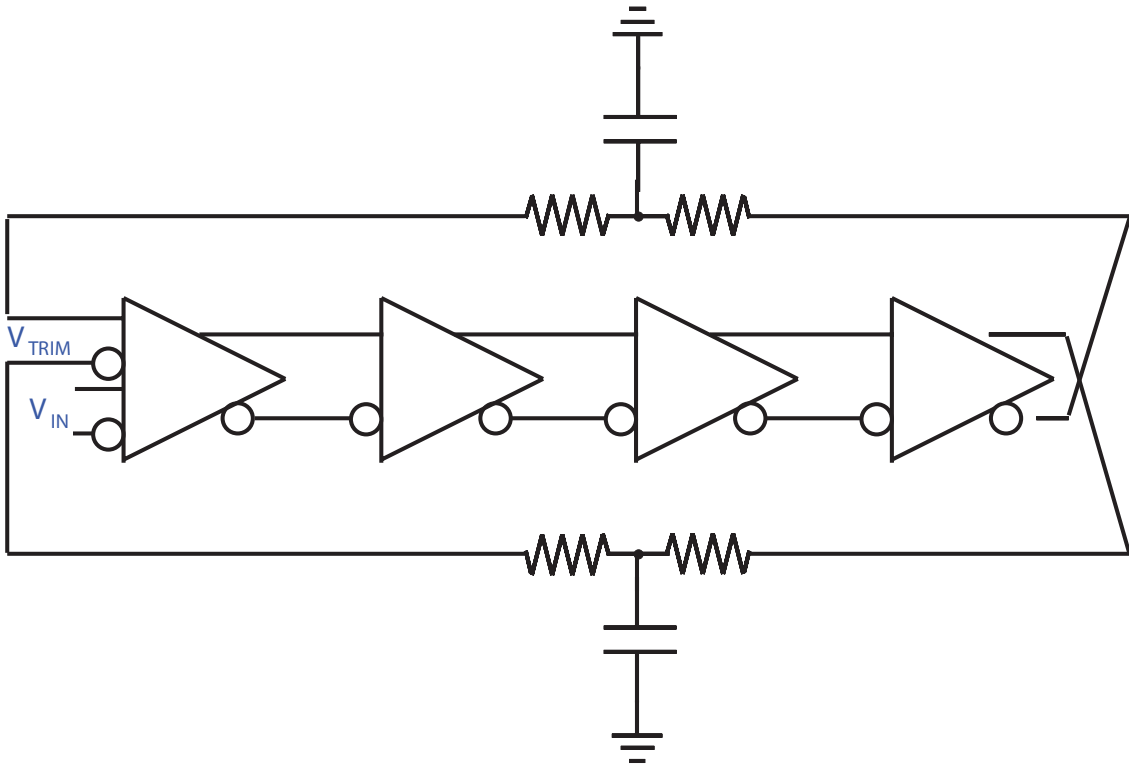

Figure 3.24: Typical Offset Cancellation Approach

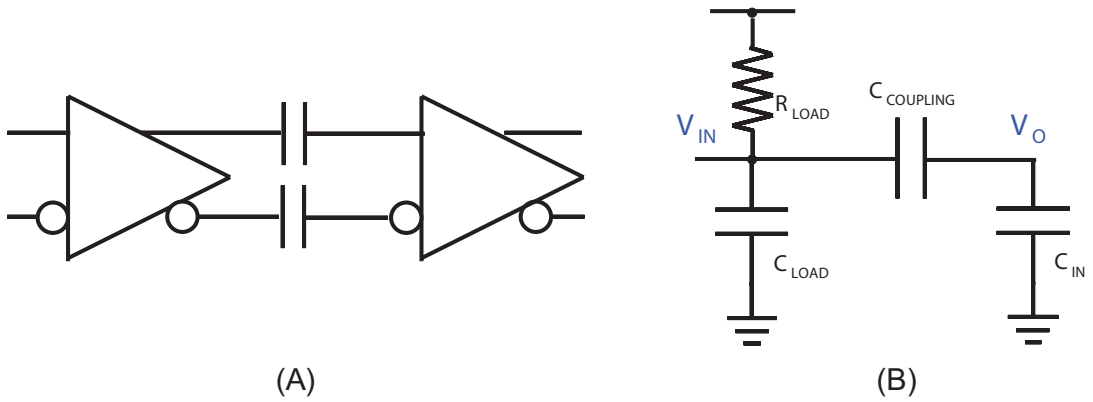

Figure 3.25: Offset Cancellation Through Capacitive Coupling

of supply voltage changes). Using a single, precision (input offset voltage around $1mV$) comparator, the analog multiplexors intended for debugging (discussed in section 3.1.5) may be used to observe the offset. An automatic algorithm may be employed to turn on all the gain stages and sequentially trim the stages from beginning to end, skipping to the next stage when the precision comparator switches (indicating offset within $1mV$ of zero). The speed of calibration would be a function of the precision comparator speed, and may be faster than the low frequency feedback approach, at the expense of only a small amount of added power consumption. Note that per-stage trim monotonicity is not strictly required, but simplifies the automatic trimming algorithm. For the purposes of testing, this precision comparator was implemented off-chip and per-stage offset trimming was explicitly programmed through the digital interface. The measured trim for each stage is shown in figure 3.26.

The use of feedforward load trimming and current stealing does cause a minor performance degradation. Current stealing causes an increase in parasitic capacitance at each node from the drain output of the trim current DAC. Also, a frequency-dependent mismatch is generated if more DAC sources are active on one leg versus the other (as the capacitive values will differ between the legs). This difference was designed to be insignificant and simulations support that conclusion. Load trimming can also create a frequency dependent offset. As shown in figure 3.27, adding load trimming creates a two pole system instead of the expected single pole. Figure 3.28 shows the resulting magnitude and phase difference (relative to a single pole) as a function of frequency for the variable gain stages with $C_{parasitic}$ varied from $50fF$ to $150fF$ in $10fF$ steps. This plot is pessimistic

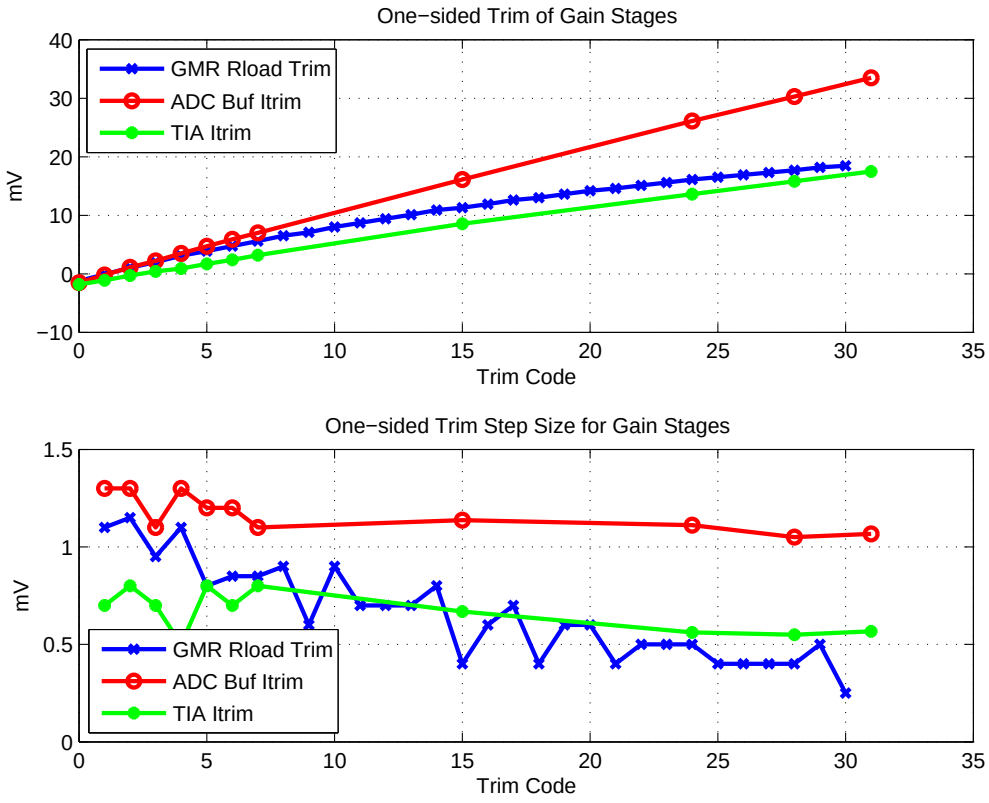

Figure 3.26: Gain Stage VOS Trim Measurements

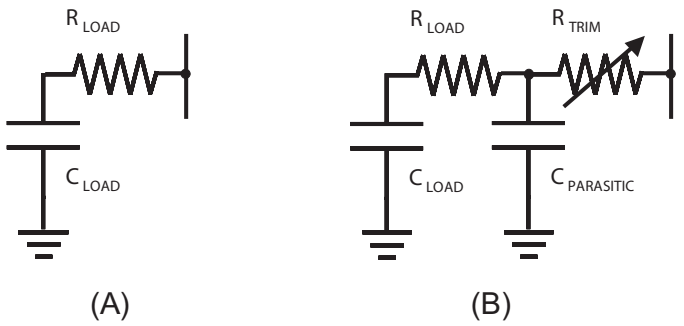

Figure 3.27: Resistor Trim Non-ideality

in the parasitic estimation to illustrate the impact that load trimming can have. (The actual R_{ds} trim value would not normally be set to the maximum of the trim range = 450Ω.) Figure 3.29 shows a similar scenario for the ADC buffer stage. As the ADC buffer stage has a smaller load resistor, the problem is exacerbated. (It was for this reason that current stealing was used to trim this stage.) Again the plot is pessimistic to illustrate that care must be taken when using this technique (or any trim technique that imbalances a differential pair).

3.1.5 Debug/Observability

The overall gain stage diagram in figure 3.2 shows the debug circuitry included with the design. It was desired to observe the output of each stage (to calibrate offset throughout the gain stages), although this could be at low frequency and hence did not require full bandwidth. For this purpose, two pass-transistor switches were used (with roughly $100MHz$ bandwidth – a compromise between parasitic loading and series resistance). One pass-transistor is placed near the gain stage, the other is at the "OBS" pins on the die. In

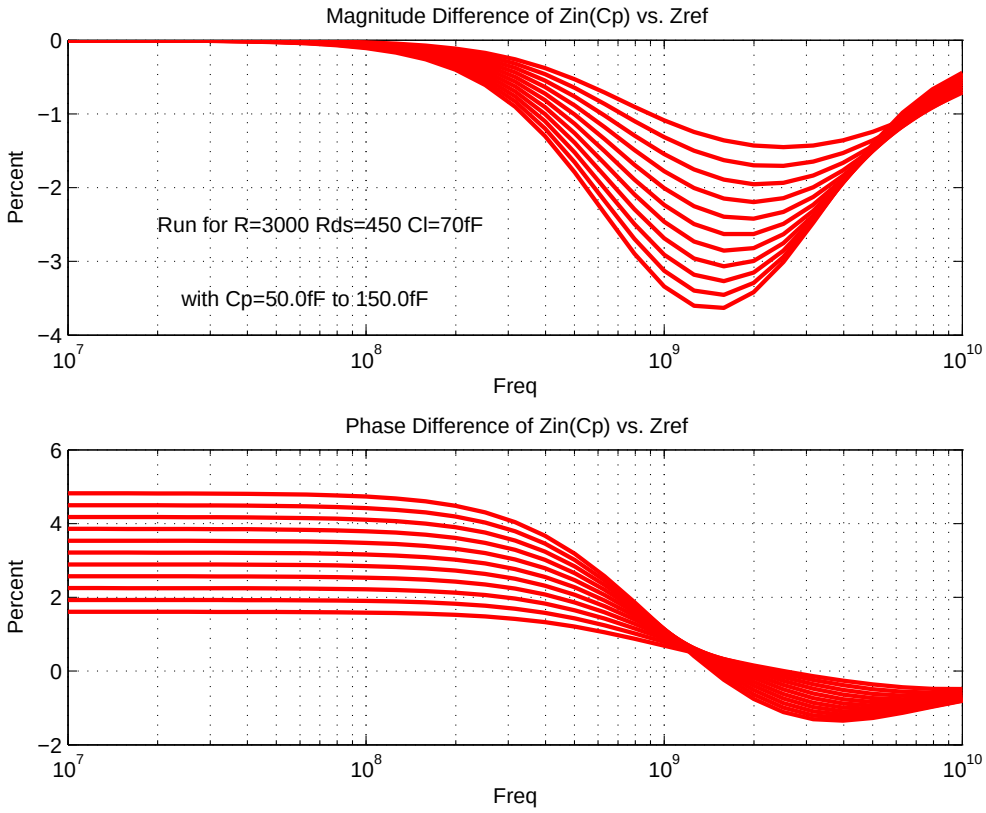

Figure 3.28: Worst-Case Magnitude/Phase Error due to Rload Trim in Variable Gain Amplifier Stages

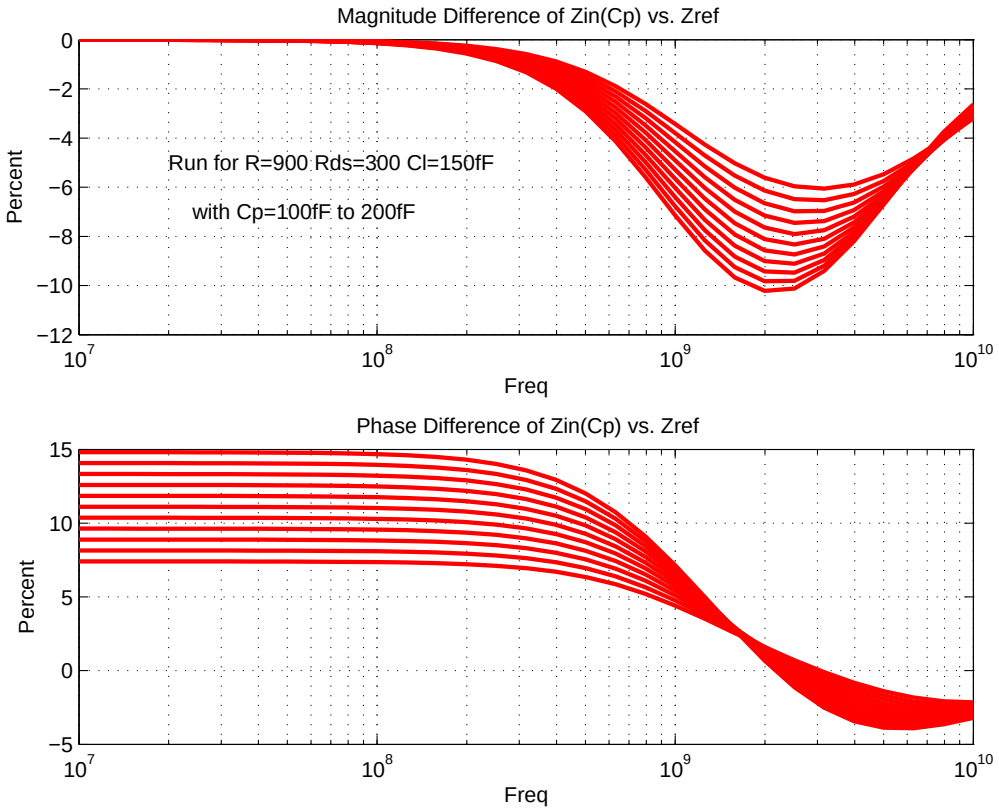

Figure 3.29: Worst-Case Magnitude/Phase Error due to Rload Trim in ADC Buffer Stage

between, a metal route is used as an analog bus, which is clamped to ground when not in use to discourage noise coupling between stages through the debug circuitry. For the places where we do want to maintain signal bandwidth through the observation path, larger pass-transistor switches were used to connect to a separate analog bus that drives a 50Ω output buffer. Additionally, for calibration, the "OBS" pins may connected to the output buffer. A circuit diagram of the output buffer is shown in figure 3.30. Note that the 50Ω buffer is essentially a pair of differential source followers (biased at several milliamps). Because these source followers represent a large capacitive load, the ADC buffer (mentioned previous in section 3.1.3) was replicated to drive the output buffer. This reduces the capacitive load presented on the analog bus, and hence reduces the necessary switch size. The output buffer S-parameters were measured over $50MHz$ to $1GHz$ and are shown in figure 3.31. Note that $-10dB$ S22 is maintained over $600MHz$ of bandwidth, although the $-3dB$ S21 corner is lower than desired. It is believed that the wiring capacitance was larger than expected, loading the analog bus line.

The layout for the overall gain block is shown in figure 3.32. Note that the locations where large, possibly clipping, signals will be present (i.e. the output and ADC buffer) are kept as far as possible from the input (TIA). The analog observation multiplexor lines, although not visible, are in the middle space between the blocks to equalize run lengths. The layout for the 50Ω output buffer is shown in the lower right corner. It is similar to the ADC buffer, only with two additional NMOS transistors and two current DAC bias sources adjacent.

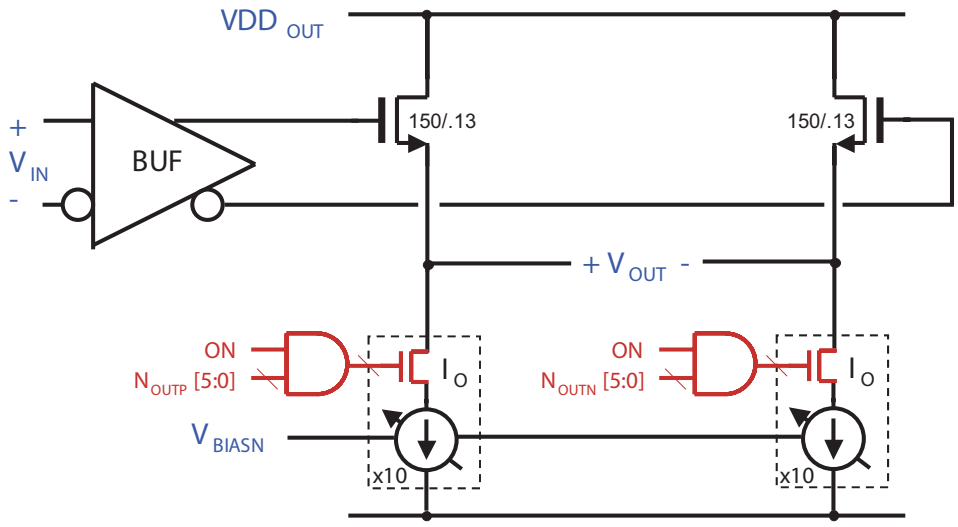

Figure 3.30: Output Buffer Circuit

3.1.6 Bias

The biasing was kept simple and digitally programmable to allow for separate control for each circuit block. Generally, the absolute value of the bias current was not important and hence strict control and matching was not required. (I.e. if the bias were 10% high or low to a particular block, it did not drastically effect performance.) Because the bias is not duty-cycled, it was desired to try to keep the bias current consumption to less than 5% of the total power consumption (although for low duty-cycled rates it rises to 30% or more of the total). A digital to analog converter (DAC) approach was taken, referenced from a single external resistor as shown in figure 3.33. A global control bias voltage is generated for NMOS (V_{BIASN}) and PMOS (V_{BIASP}) devices. Note that this means that the bias control voltage and V_{DDBIAS} supply need to be routed to the local bias mirror in each block to maintain matching fidelity. This was done to save current versus exporting a

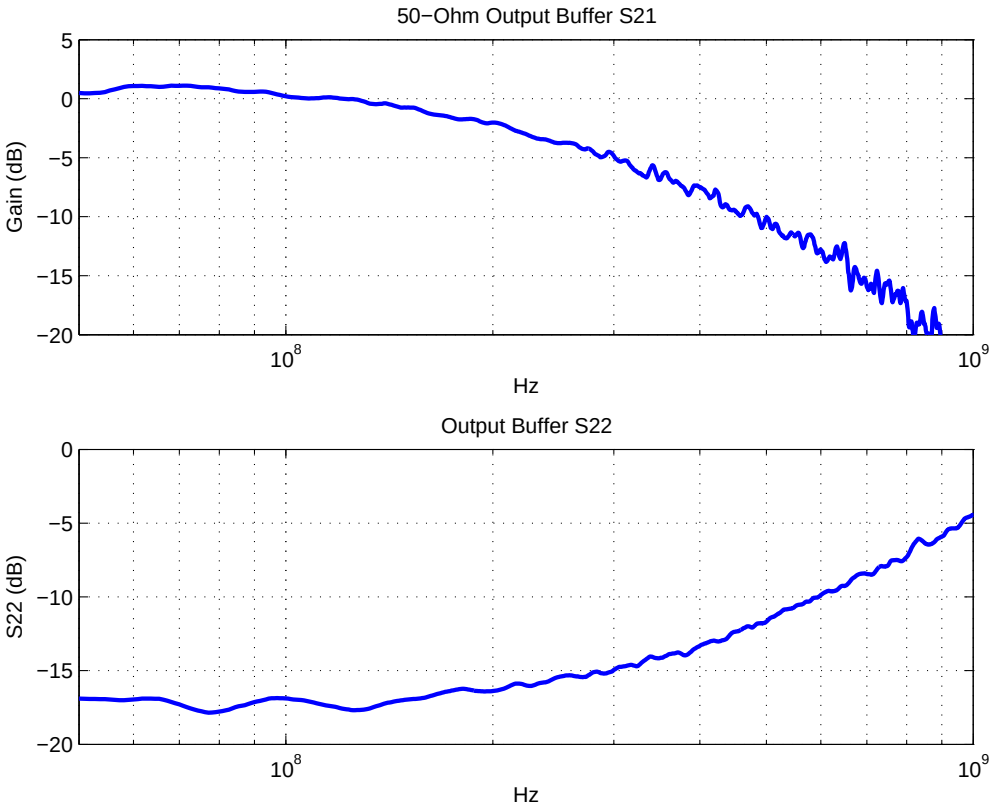

Figure 3.31: Output Buffer S-Parameter Measurements

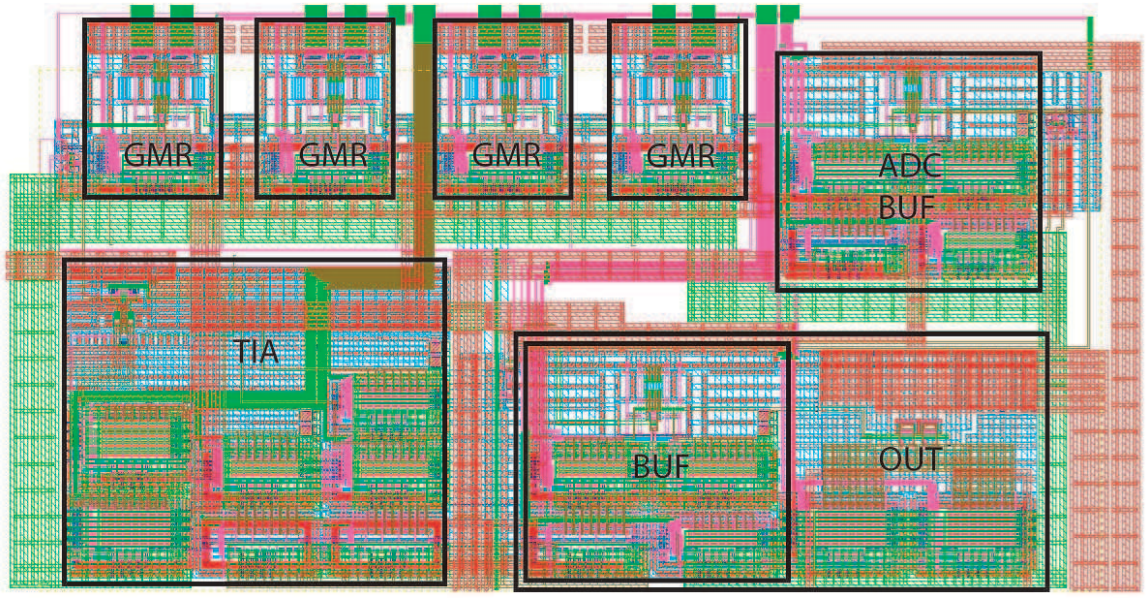

Figure 3.32: Gain Stage Layout

separate bias current to each block. Admittedly this is a small savings, and it is probably more prudent to send only the current line (and locally generate a bias) than to send the bias supply and power lines over the whole chip. To reduce any digital coupling into the bias lines the VDD and BIAS routes were overlaid and any switching digital lines were kept away or at minimal overlap. As the bias sources for the gain stages are duty-cycled, they kick charge back into the bias line. Hence, the bias pole was kept high (smaller amounts of decoupling) to allow the bias to clear quickly and recover. Larger values of decoupling could be used to smooth the kickback, but a decision was made to save area and go with a lighter, faster approach.

For debug purposes, the NMOS and PMOS bias circuits may separately generate their own voltages from external resistors. For the measurements in this document, the

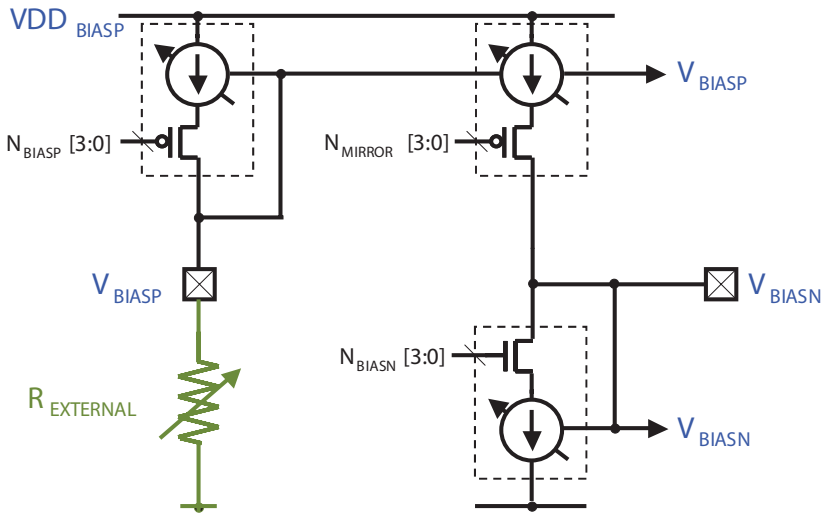

Figure 3.33: Bias Circuit

PMOS was mirrored to the NMOS to create the global bias voltages and only one external resistor was used. The global unit current used, referred to as the "lsb value", was set to approximately $8\mu A$. Circuit diagrams for the "lsb" unit current for the bias DAC's are shown in figure 3.34. Row and column decode signals are multiplexed to each DAC element using a switch to pass current if needed. (Row and column signals are decoded with simple logic gates from an unsigned 2's complement representation using a thermometer and one-hot code.) This results in a monotonic DAC curve which is advantageous for simple digital control loops (e.g. to automatically trim offset).

Figure 3.35 shows the global bias layout. It consists of three bias current DACs. The same bias cell footprint (and decoding logic) were used for DACs throughout the chip to save design time (at the expense of inefficient area usage). To get trim, bias, or capacitive DACs, the unit cell layout may be swapped out. All bias and trim DACs were kept in the same orientation to improve matching across the die.

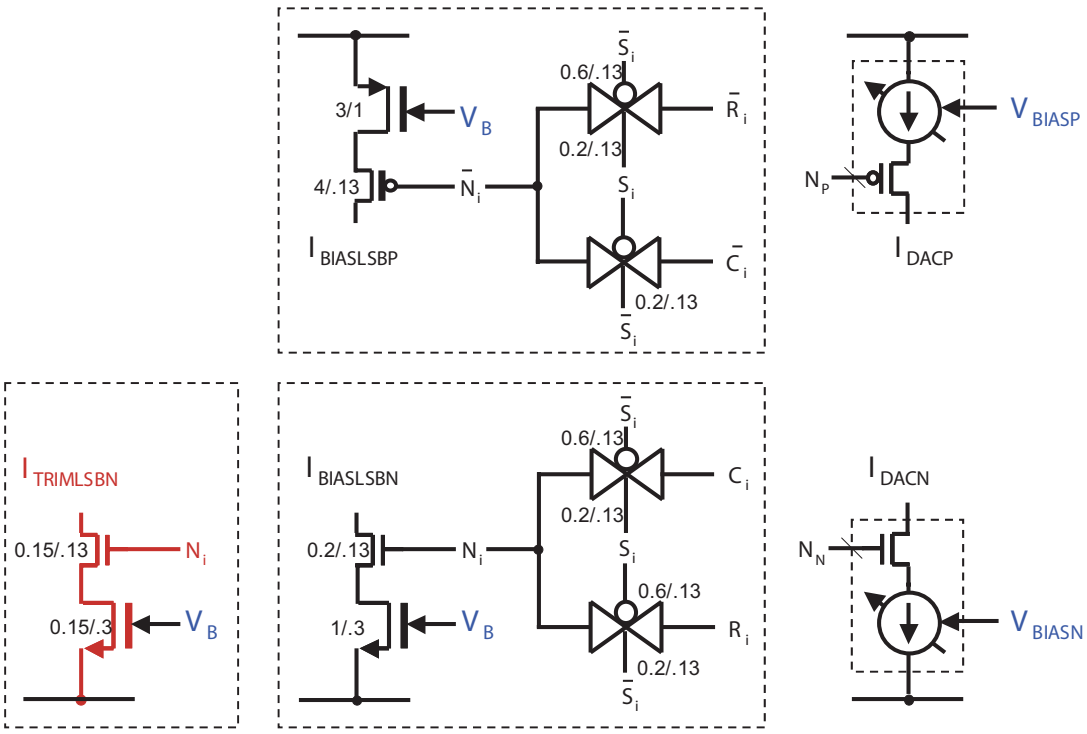

Figure 3.34: Bias DAC lsb Circuit

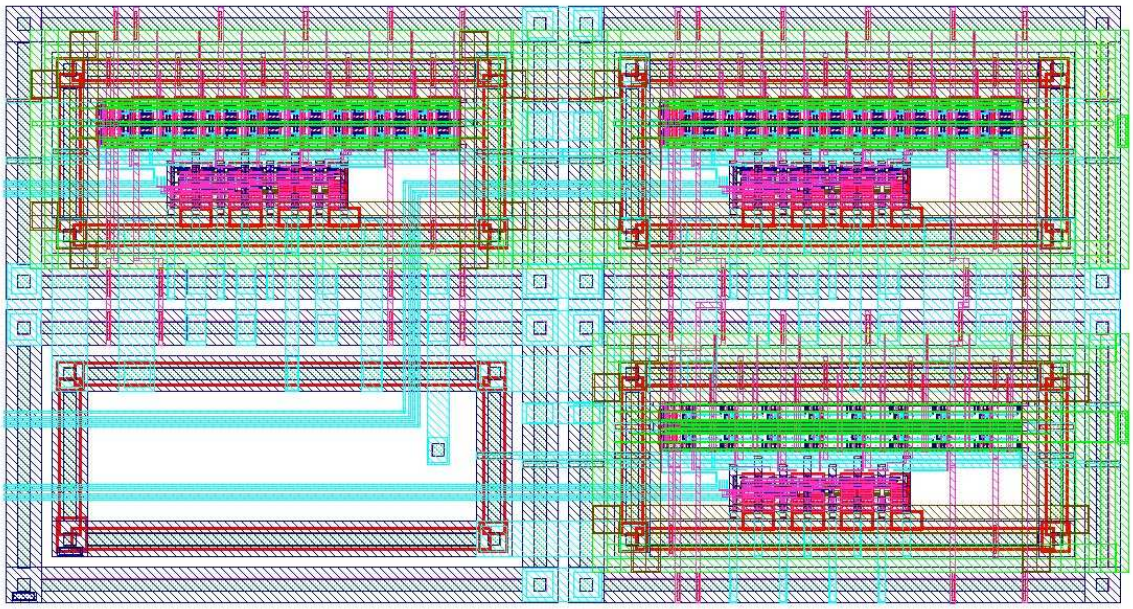

Figure 3.35: Bias Circuit Layout

3.2 Oscillator

Because we are deriving the sampling clock from the oscillator, its performance must satisfy the criteria in section 2.9. (Ideally it must have better performance to allow for margin in the delay locked loop which generates the sampling edges.) In particular it is desirable to have sub-1PPM matching and a jitter standard deviation much less than 75ps. To facilitate low cost, a crystal with a larger manufacturing tolerance may be used which requires a wider pulling range to ensure the matching criterion is met. Additionally the oscillator is always active so low power consumption is a main concern. Oscillator power as related to the Q of the resonator may be calculated as:

$$I_{bias} > \left(\frac{1}{Q}\right)^2 (\omega_o C_p)^2 \left(\frac{C_1 + C_2}{C_1}\right)^2 \frac{1}{2k'W/L} \tag{3.6}$$

Fortunately crystals have enormously large Q's, on the order of 10^4 to 10^5, and thus power consumption can be very low. Several examples of very low power crystal oscillators have been reported in the literature. Current consumption below $200\mu A$ was reported in [90] for a $78MHz$ oscillator digitally trimmable to $0.3PPM$. Also, for a $2.1MHz$ oscillator in [99] only $0.5\mu A$ was consumed. As can be observed, the power consumption is a function of frequency. In fact, I_{bias} increases approximately proportional to f_{osc}^2 for submicron devices [90]. This implies a global power consumption trade-off for the system clock frequency choice that is also discussed in section 3.3. Lower frequency operation lowers power consumption for the oscillator, but requires more power consumption from a phase locked loop (PLL) or delay locked loop (DLL) to generate the sampling clock. Likewise, too high of a frequency constrains crystal cost and availability and burdens the oscillator with much larger power consumption.

The Pierce topology was chosen as it is commonly used for integrated oscillators. Both capacitors are referenced to ground (as are any top or bottom plate parasitics) and the gate and drain capacitances of the MOSFET as well as the bonding pad capacitances may be incorporated into the total capacitance value. Additionally, oscillator frequency may be varied (pulled) by changing the value of these capacitances. Two pins are needed (one for the gate and one for the drain connection), as well as an external crystal and parallel LC (inductor - capacitor) tank to obtain third harmonic oscillation. It is possible to get crystals cut to operate in the fundamental mode for a $60MHz$ frequency, which would avoid the need for an external parallel LC tank. Unfortunately the author was unable to obtain a small number of these crystals for a reasonable price. Hence the decision was made to use

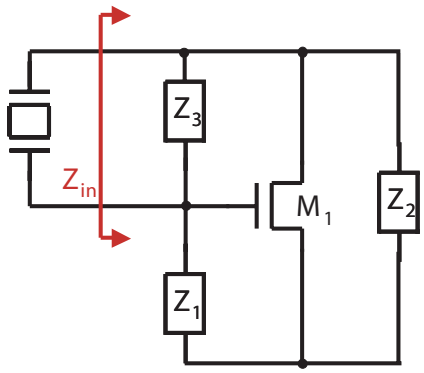

Figure 3.36: Three-Point Oscillator Circuit

third harmonic mode oscillation.

Using theory developed for "three-point oscillators" [100], shown in figure 3.36, the impedance seen by the crystal may be written as:

$$Z_c = \frac{Z_1 Z_3 + Z_2 Z_3 + g_m Z_1 Z_2 Z_3}{z_1 + Z_2 + Z_3 + gm Z_1 Z_2} \tag{3.7}$$

The real part of Z_c must be negative and larger than the crystal resistance to guarantee oscillation. Additionally, the effect of capacitive pulling may be found as:

$$p = \frac{C_x}{2 * \left(C_3 + \frac{C_1 C_2}{C_1 + C_2}\right)} \tag{3.8}$$

Using these equations, it is possible to design the oscillator for a particular operation frequency and determine the pulling range and estimate the current consumption. To cover the possible design corners, a Matlab model that incorporates the expected variation from the NMOS devices, temperature, pin and pad capacitance, on-chip metal-metal capacitance, and bias current was generated. Figure 3.37 shows the worst-case predicted

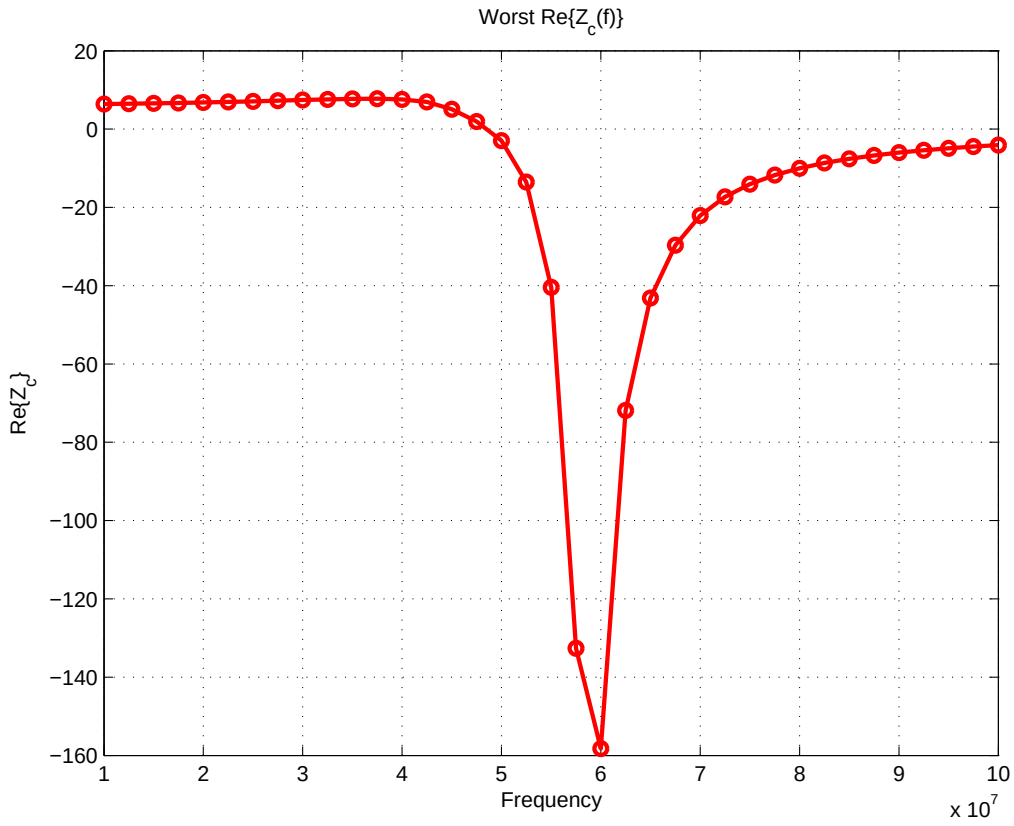

Figure 3.37: Oscillator Impedance vs. Frequency with Third Harmonic Tank

real component to the impedance seen by the crystal over frequency. The target pulling range was designed for approximately $30PPM$ with a $0.5PPM$ to $1PPM$ stepsize for a $100\mu A$ bias, using two $6.4pF$ DAC capacitor arrays with a $25fF$ stepsize. The phase noise (and hence jitter) were expected to be well within desired limits due to the use of a crystal (high-Q) oscillator (see figure 2.15).

Figure 3.38 shows the oscillator circuit diagram. The DAC capacitor array was designed similar to [101], but taking advantage of the same decoding and unit cell floorplan as for the bias DAC sources in section 3.1.6. The capacitors were implemented using metal-

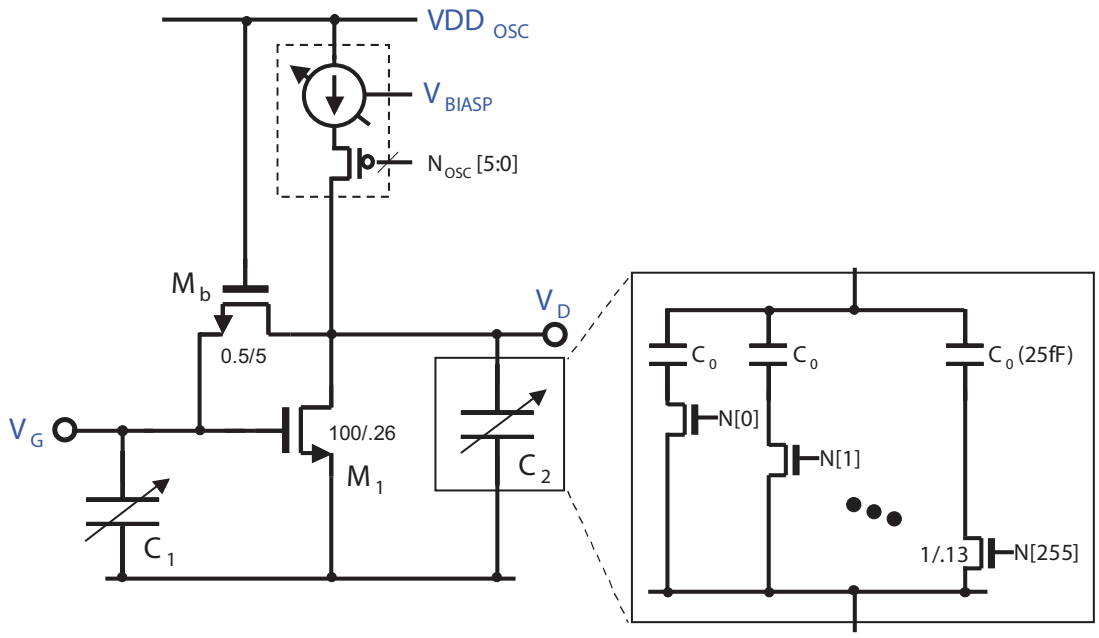

Figure 3.38: Oscillator Circuit

metal finger capacitors that may be connected or isolated from the gate/drain through a switch. A screen capture of the oscillator layout is shown in figure 3.39 with a close-up of the DAC unit capacitance cell (i.e. a "lsb") inset on the left. A PMOS current source is located at the bottom middle with the NMOS device in the middle. Interdigitated V_{DD} and GND rails surround the block for isolation. Measurements for the oscillator yielded a jitter standard deviation on the output clock pin of $23ps$ (this is after oscillator to clock conversion, discussed below) with $72\mu W$ of power consumption from a $1.1V$ supply. Pulling was only observed over $10PPM$; limited by a larger C_1 and C_2 used to guarantee oscillation. The interplay between the value of C_1, C_2, bias current, and LC trap tuning and resulting pulling is complex. In the laboratory, the LC trap was implemented with a fixed inductor ($0.39nH$) and trim capacitor ($5pF$ to $20pF$) to allow for proper tuning. In future work, the

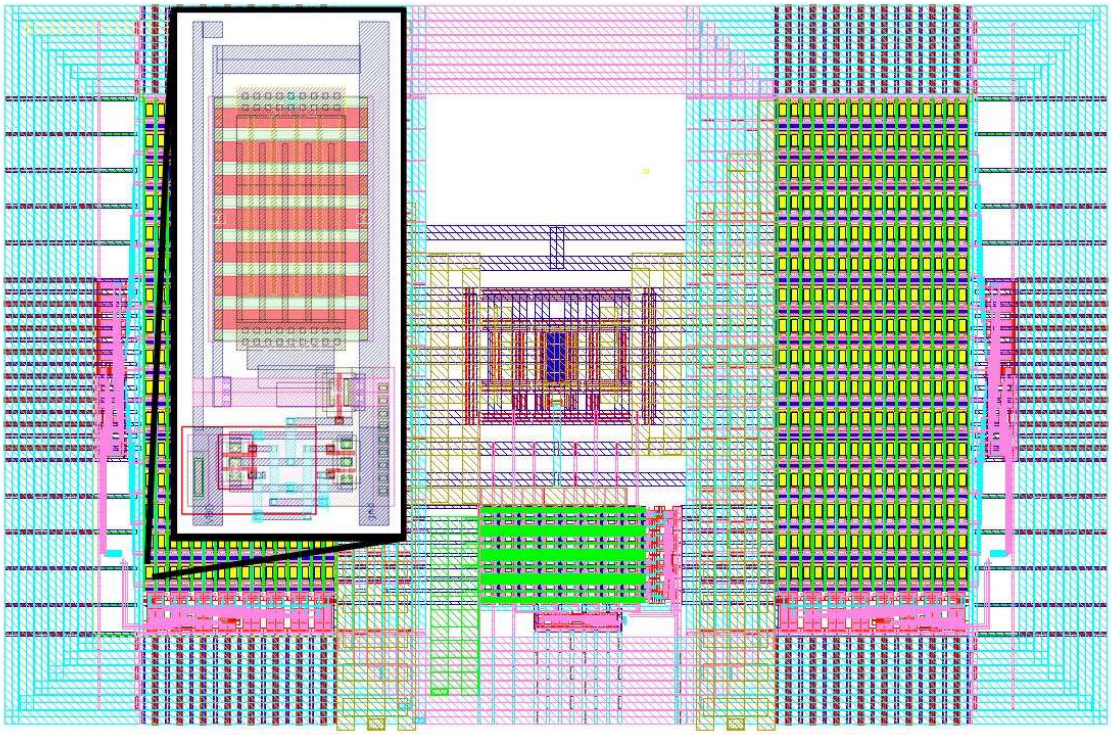

Figure 3.39: Oscillator Layout with Capacitive DAC "lsb"

trade-offs between pulling range for low jitter, low power operation may be further analyzed. Additionally the LC trap may be implemented on-chip (using a capacitive DAC for C_{trap}).

3.2.1 Oscillator to Clock Conversion

To minimize oscillator bias current, maintain good jitter performance (by keeping a clean sinusoidal oscillation), and reduce the possibility of squegging (heavily nonlinear operation, sometimes resulting from overdriving the crystal), the oscillator amplitude is kept below the power supply rail limits. Because the oscillation voltage is not large, it is not well suited to driving digital circuits as a clock. This compels the use of an additional circuit to convert the oscillation voltage into a stable, well-controlled digital clock signal.

Ideally this conversion occurs without consuming much current and without adding any additional jitter to the signal. In practice, meeting both of these goals is difficult. Jitter may be controlled through the use of a linear amplifier which unfortunately draws constant current. This may be an opportunity for future work investigating the trade-off between the oscillator and clock generation designed because no topology was definitively identified as superior. After investigating several approaches, the circuit from [100] was chosen and a circuit diagram is shown in figure 3.40. Essentially a class-B amplifier, the oscillator input is capacitively coupled into an inverter input. The bias is set to the appropriate tripping point (threshold) by using an identical inverter with shorted input and output. Current through the inverters is choked through proper device sizing to prevent a large amount of short-circuit current dissipation. As the signal is amplified through the inverter stages it begins to clip, becoming more digital, which allows us to scale the device sizes towards minimum length (to improve speed and edge rate and decrease the capacitance switched). To allow for debugging, two multiplexors were added: one to choose between clock polarity, and the other to override the local clock and use and off-chip source. Also, the output clock is buffered and sent to a pin for observation. A pin was also connected to node V_B (through a pass-transistor) to allow for inspection. Power gating transistors (driven by "On") were added to allow us to shutdown the amplifier and provide a valid logic output, in the event that the clock is overridden and the amplifier is not needed. Layout for the clock generation circuit is shown in figure 3.41. The input coupling capacitor is located in the upper left and the inverters and logic are implemented in a standard cell array.

A couple non-idealities were observed with this design. Firstly, the capacitively

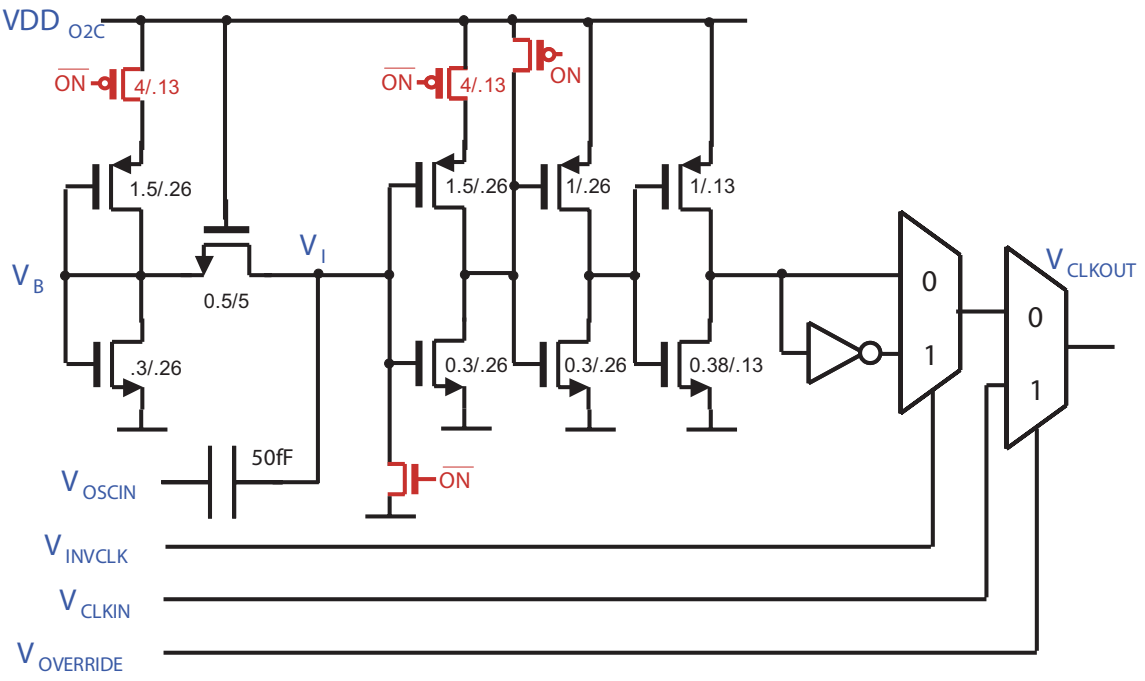

Figure 3.40: Oscillator to Clock Conversion Circuit

coupled input creates a capacitive divider which attenuates the input. Care must be taken to isolate the parasitic capacitance at node V_I to prevent undue signal loss (which will require more gain, and hence current, to re-amplify the signal as well as possibly increase jitter due to lower input SNR). A very long channel NMOS is placed between bias node V_B and V_I to provide a large resistance to isolate these nodes. Secondly, due to the nonlinearity of the gate capacitance as the inverter devices transition from weak inversion to strong inversion, a shift in the bias point was observed. This shift mildly perturbs the bias away from the ideal threshold value, changing the resulting clock duty-cycle (although the overall impact was small).

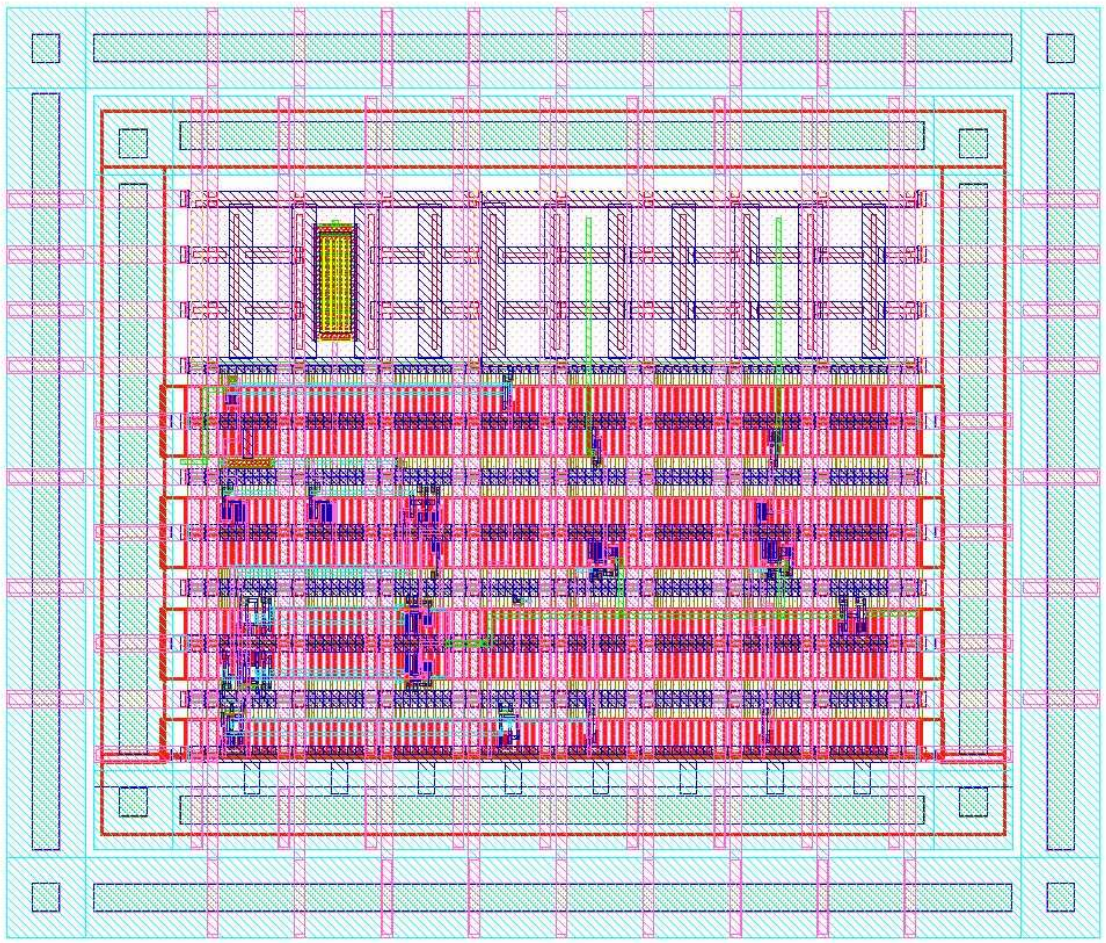

Figure 3.41: Oscillator to Clock Circuit Layout

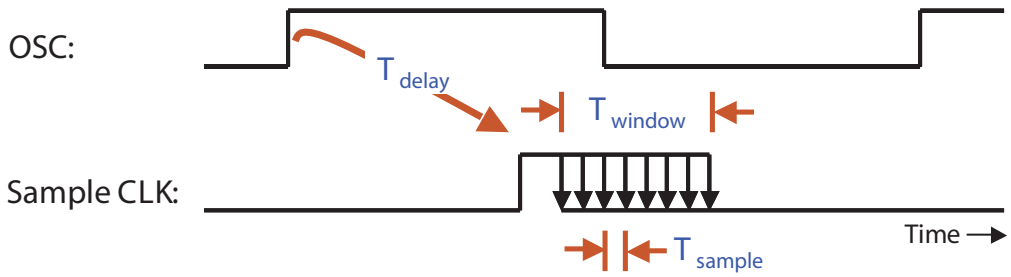

Figure 3.42: Desired Sample Clock Generation

3.3 Delay Locked Loop

The main function of the delay locked loop is to turn the global system clock into an effective $2GHz$ sampling clock as illustrated in figure 3.42. This may be achieved by sliding the oscillator frequency along a continuum from the sampling clock rate down through the window rate to the pulse rate, as shown in figure 3.43. Using dividers (essentially counters), lower frequency events may be easily derived from the system clock. To get higher frequency events, a more complicated approach must be taken using either a phase locked loop (to multiply the input clock frequency) or a delay locked loop (to divide the input clock frequency into smaller intervals).

To choose an architecture for clock generation, first we observe that we want the oscillator frequency to be as low as possible to save power in the oscillator, as g_m, and hence I_{bias} in sub-micron CMOS scales roughly as f_{osc}^2 [90]. To accommodate this, either a phase locked loop (PLL) or delay locked loop (DLL) may be employed. A DLL was chosen as it is possible to drive the entire system from a moderate speed clock if the pulse rate for the system is limited to be near the delay spread of the channel, i.e. around $50ns$, to avoid the added complexity of channel equalization in the receiver. By selecting the delay line length

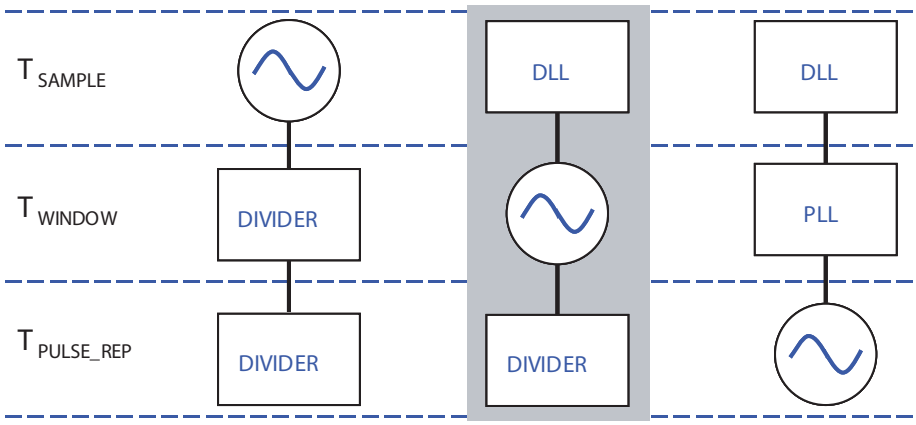

Figure 3.43: Sample Clock Generation Strategies

and f_{osc} appropriately, we can get well-controlled $0.5ns$ steps between consecutive delay cells and generate a virtual $2GHz$ effective sampling rate by combining the delayed clock phases. The window and pulse rate may also be easily created by dividing (or counting) f_{osc} edges. An additional benefit of a DLL is that it does not accumulate jitter [102] and hence should have better jitter performance than a PLL. The per-stage divider jitter is proportional to the output slope[103], and with careful design, the total jitter (oscillator plus DLL plus control logic generation for the ADC sampling) can be kept below the $75ps$ target. From the oscillator we also may derive the pulse repetition clock for transmission using a programmable divider for flexibility.

A block diagram of the DLL is shown in figure 3.44. With a system clock frequency of $60MHz$, 32 delay cells are needed to approximately create the desired $2GHz$ sampling rate. (An actual rate of $1.92Gsample/s$ is achieved.) Because $60MHz$ is relatively high for a crystal oscillator, future work should include investigation the use of a PLL/DLL structure to multiply the input frequency by a small factor (i.e. '2' or '4') thus allowing

for a cheaper, more readily available crystal choice as well as some savings in oscillator power. This change is not very complicated, requiring a phase frequency detector (PFD) instead of a phase detector (PD), a loop filter change, and some feedback on the voltage-controlled delay line to turn it into a voltage-controlled oscillator. (In fact, the choice to use a PFD instead of a PD was originally made with this in mind.) However, in the interest of simplicity and better jitter performance, a DLL-only was chosen in the end.

The DLL layout is shown in figure 3.45. The delay cells and DLL output phases are generated at the top. The charge pump and PFD are shown in the inset. The reference and feedback clocks enter on opposite sides at the bottom in the inset, have their phases compared, and feed into the charge pump circuit at the top. The loop filter is split into two capacitor banks on either side of the DLL and are implemented with metal-metal finger capacitors. In the middle bottom of the DLL is the global bias current DAC which is mirrored (as for the gain stages) to set the charge pump current value.

3.3.1 Delay Cell

A survey of DLL and PLL papers was performed with a focus on the delay cell design to attempt to determine a good candidate for a low power, low jitter implementation. No clear circuit emerged as the most suitable, although the RC delay approach exemplified by [104] seemed to be more popular. Of the many topologies observed, most appeared to reduce to one of three basic approaches: $C\Delta V/I$, RC, or t_d; shown in figure 3.46. The first topology, A, uses a (possibly) variable current into a (possibly) variable load capacitor to achieve a delay that is proportional to $C\Delta V/I$. The second topology uses an RC delay, with a (possibly) variable resistor and (possibly) variable load capacitance. The third

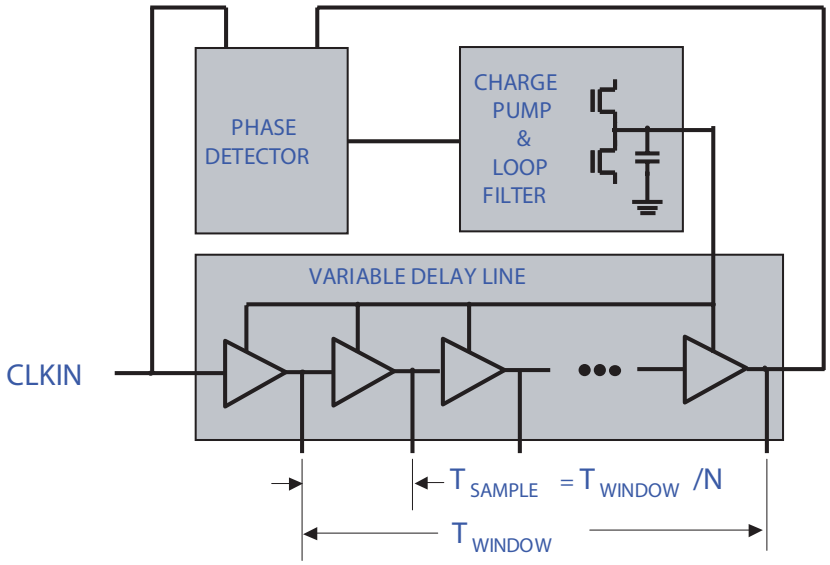

Figure 3.44: DLL Diagram

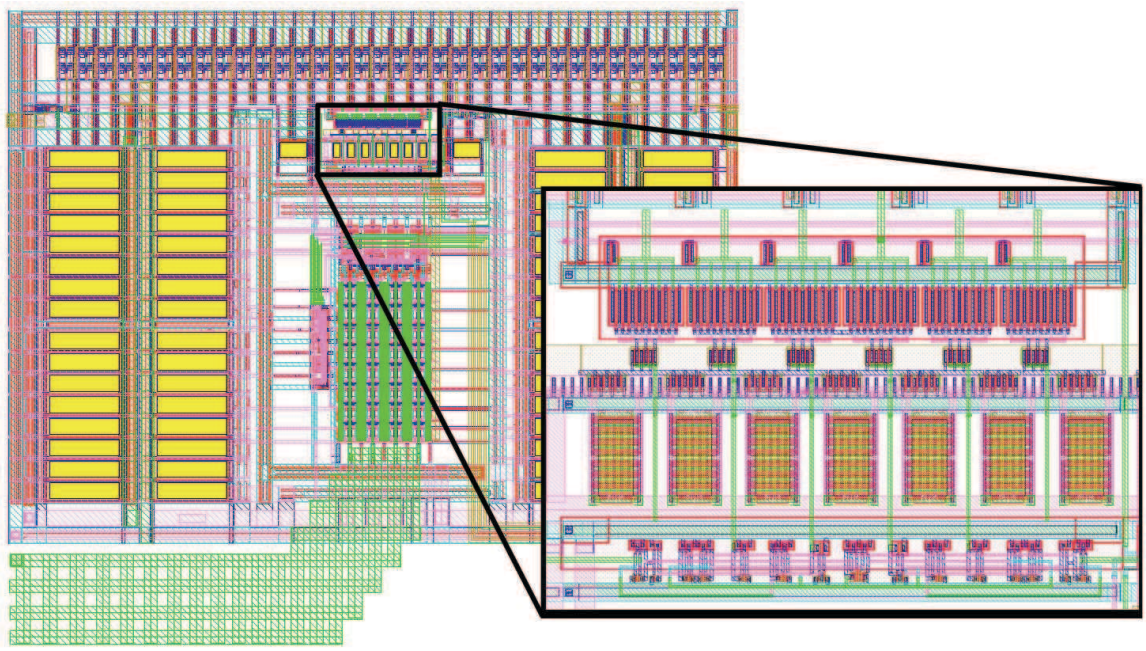

Figure 3.45: DLL Layout with CP and PFD Inset

topology, typified by but not limited to "all-digital" approaches, selects a variable delay from the inherent propagation delay through a gate, device, wire, etc. The third topology was rejected because a wide delay range with a fine spacing generally implies larger area and power consumption than the first two approaches. Of the $C\Delta V/I$ and RC options, RC circuits generally used statically biased delay cells and had voltage swings less than the power supply range requiring re-conversion to digital logic levels. Hand analysis predicted similar power consumption for $C\Delta V/I$ as well as RC delay cells was possible given the delay target. Also RC is predicted to achieve better supply noise immunity (and hence have lower jitter). However the extra power consumption to convert an RC signal back to digital values (which would be needed for every tap, not just at the end of the delay line as for some DLL applications) pushed the decision to use an $C\Delta V/I$ topology. Single-ended operation was chosen for lower power consumption, in spite of the increased risk of supply noise coupling. This necessitated careful design of the power supplies and decoupling on the supply and bias lines. For the $C\Delta V/I$ approach, only variable current was used to change the delay; capacitance was fixed. Varying C_{load} seemed inefficient from a current-use perspective, as the current was fixed, independent of delay, not allowing a savings in power if lower speed operation were desired. Additionally it seemed conceptually easier to vary current over a larger range than a variable capacitance. Figure 3.47 shows the circuit diagram of the delay cell.

Extensive simulations over process and temperature were performed to characterize the delay cell range and sensitivity to supply and bias variation. A combination of a Perl script and ELDO (a SPICE variant) was used to automate this characterization; extracting

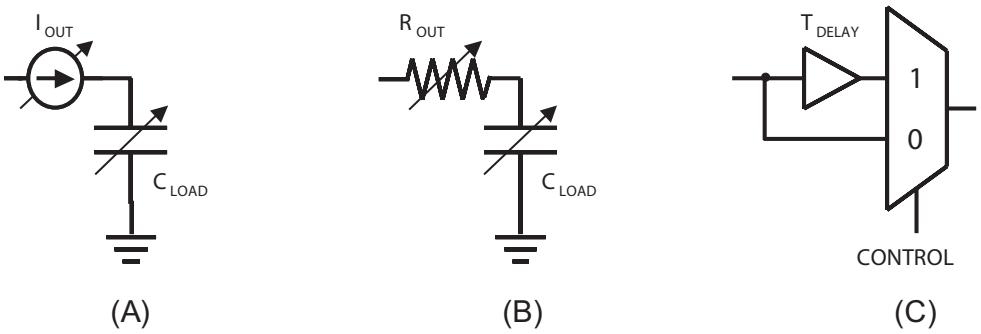

Figure 3.46: Delay Cell Topologies

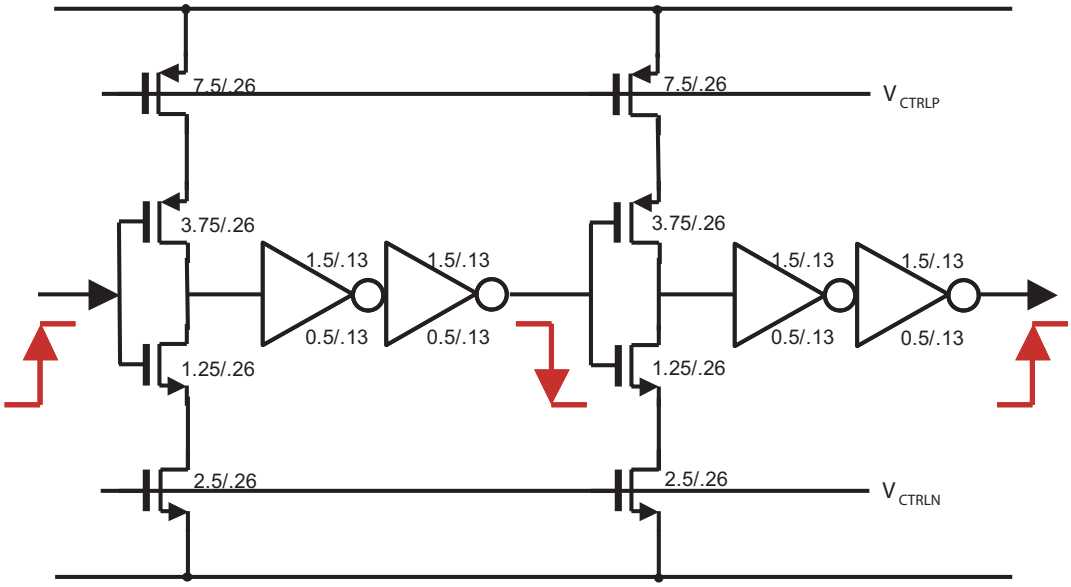

Figure 3.47: Delay Cell Circuit

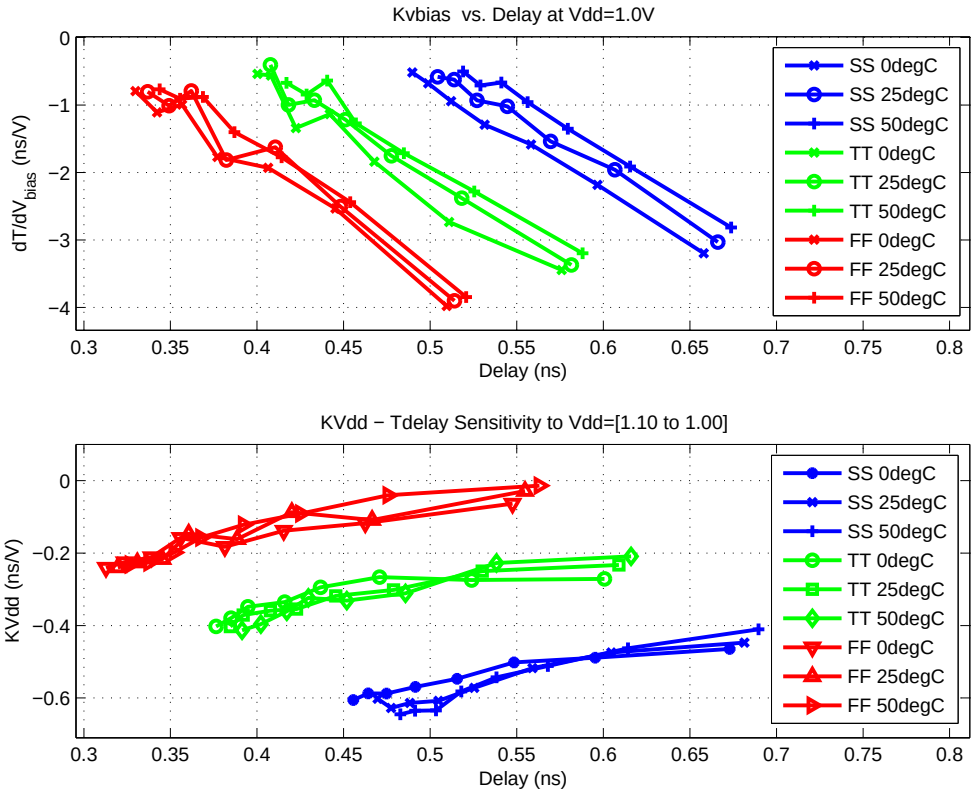

Figure 3.48: Delay Cell Sensitivity Simulation Results

delay and edge rate information over the design corners. Figure 3.48 shows the bias and supply delay sensitivity (dT_{delay}/dV) at the low supply corner with $V_{DD} = 1.0V$ over process and temperature corners. As expected, the bias voltage exerts more influence on the delay (which is desirable), by almost an order of magnitude. The delay as a function of bias current is also plotted in figure 3.49 over process and temperature corners for a $1.0V$ supply voltage (the minimum target supply voltage). Note that a delay of $0.5ns$ is the target.

Jitter performance was analyzed using the first-crossing approximation[105] for the conversion of voltage noise into jitter. This theory posits that jitter arises from voltage

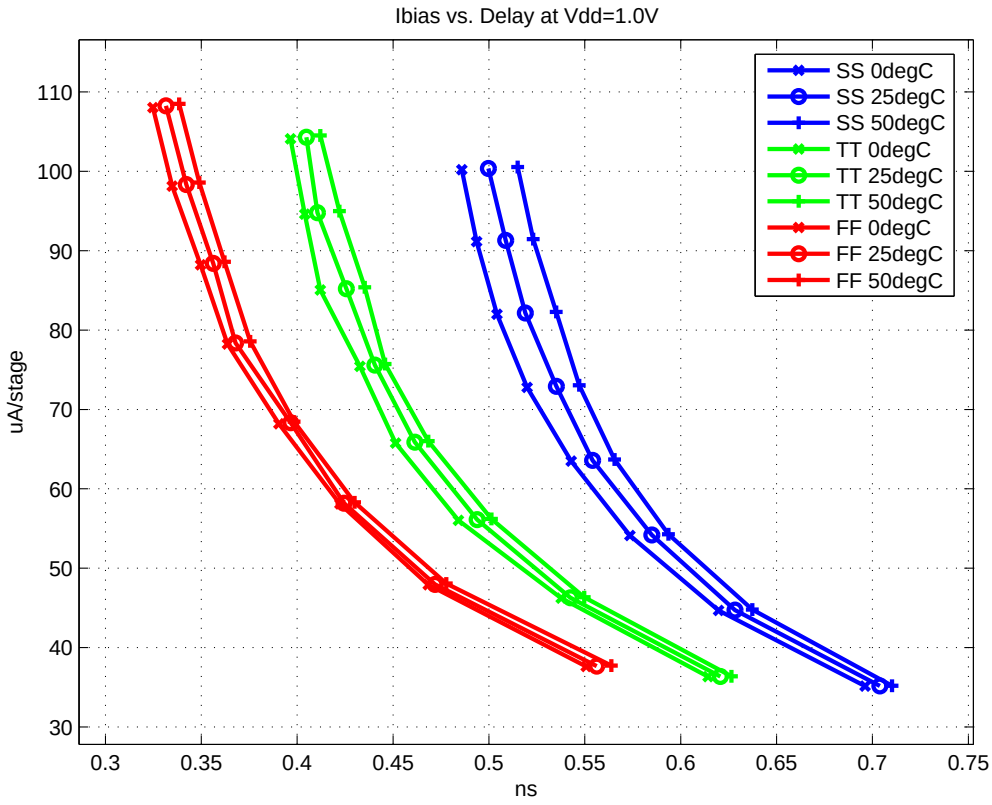

Figure 3.49: Delay Cell Delay vs. Bias Current Simulation Results

noise being converted into time uncertainty around the threshold/tripping point of a circuit. Jitter, therefore, is an inverse function of the edge rate. As we achieve delay by slowing the edge rate, this places a practical limitation on the lowest edge rate that may be used before our sampling jitter constraint is violated. A minimum edge rate of $1GV/s$ (or $1V/ns$) was selected to ensure that jitter was kept to acceptable levels. Using [106], the jitter standard deviation was calculated for the last delay tap (the worst-case is last delay tap) to be from $\sim 17ps$ for a $2.6GV/s$ edge rate to $\sim 44ps$ for a $1GV/s$ edge rate. The chip was fabricated in a slow/typ process and measured at $1.1V$. For this case we expect a $\sim 2GV/s$ edge rate, and hence a $\sim 22ps$ standard deviation.

Another problem that may arise in delay chains, especially long chains such as ours, is that of "pulse-swallowing." Illustrated in figure 3.50 pulse swallowing occurs if the rise and fall times through a delay chain are not well matched. As the pulse propagates through the chain it winds up expanding or shrinking depending upon the mismatch. Ultimately, if the mismatch is large enough and the delay chain long enough, the pulse might disappear entirely. Even if the pulse does not disappear, the duty cycle of the pulse may be drastically changed, causing problems to downstream circuits that expect a more symmetric duty cycle. The delay cell designed here avoids this problem by concatenating two current-starved delay cells separated by a logical inversion into a single delay stage. (Refer to figure 3.47.) This ensures that every input edge produces both a rising and falling edge inside the delay cell, equalizing the mismatch to first order at the expense of doubling the transition activity (and hence increasing power consumption). While preserving the input pulse duty cycle is not strictly required (downstream circuits were designed to handle duty cycle variation

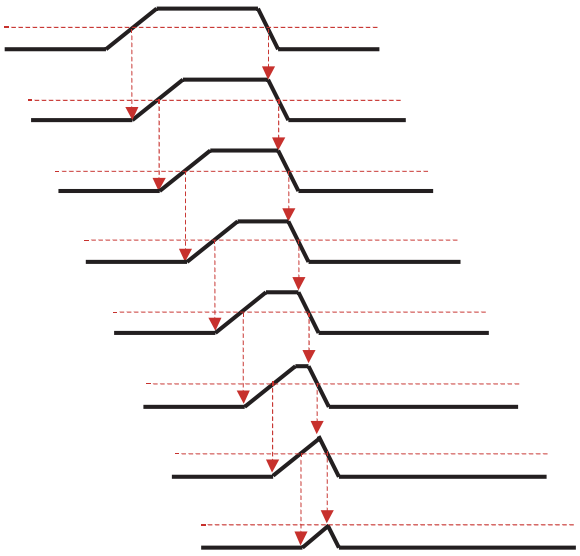

Figure 3.50: Illustration of Pulse Swallowing in a Long Delay Line

from 25% to 75% worst-case), the power supply cost was acceptable and the guarantee of robust operation was reassuring.

Finally, a screen capture of the layout of the delay cell is shown in figure 3.51. The current starved inverters are to the right with non-minimum sized lengths, each followed by two inverters.

3.3.2 Phase Detector

The phase detector used is actually a phase-frequency detector (comparator), not strictly a phase detector. Originally this was to support the possible use of a PLL instead of a DLL, however, the extra power (and area) penalty due to the more complicated circuitry is minor, so no redesign was done. The circuit is based on the PFD mentioned in [107] and [104], but modified to allow duty-cycling of operation and to produce nearly balanced

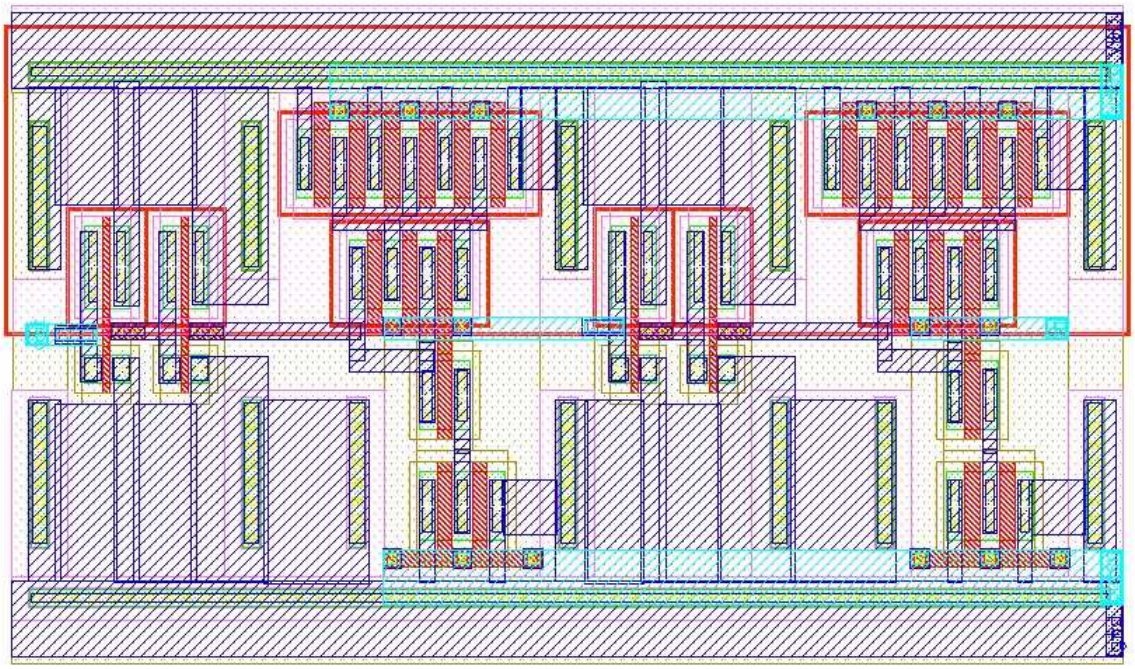

Figure 3.51: Delay Cell Layout

differential outputs. Additional NAND gates at the input are used by the controller to mask which pulses are allowed to update the DLL. Thus a pulse may be sent down the delay line and then compared to the following clock edge without causing the DLL to false lock. The PFD was implemented with hand-sized complementary static CMOS logic gates to equalize delays for the "UP" and "DN" (down) pulses.

3.3.3 Charge Pump and Loop Filter

The charge pump design is similar to the circuit in [104]. A replica bias circuit, based on the delay cell design, is used to duplicate the charge pump output voltage V_{CTRLP} for the NMOS delay control voltage: V_{CTRLN}. Additional switches were inserted to support duty-cycled operation. The switch "M1" is used to tie the gates of the charge pump

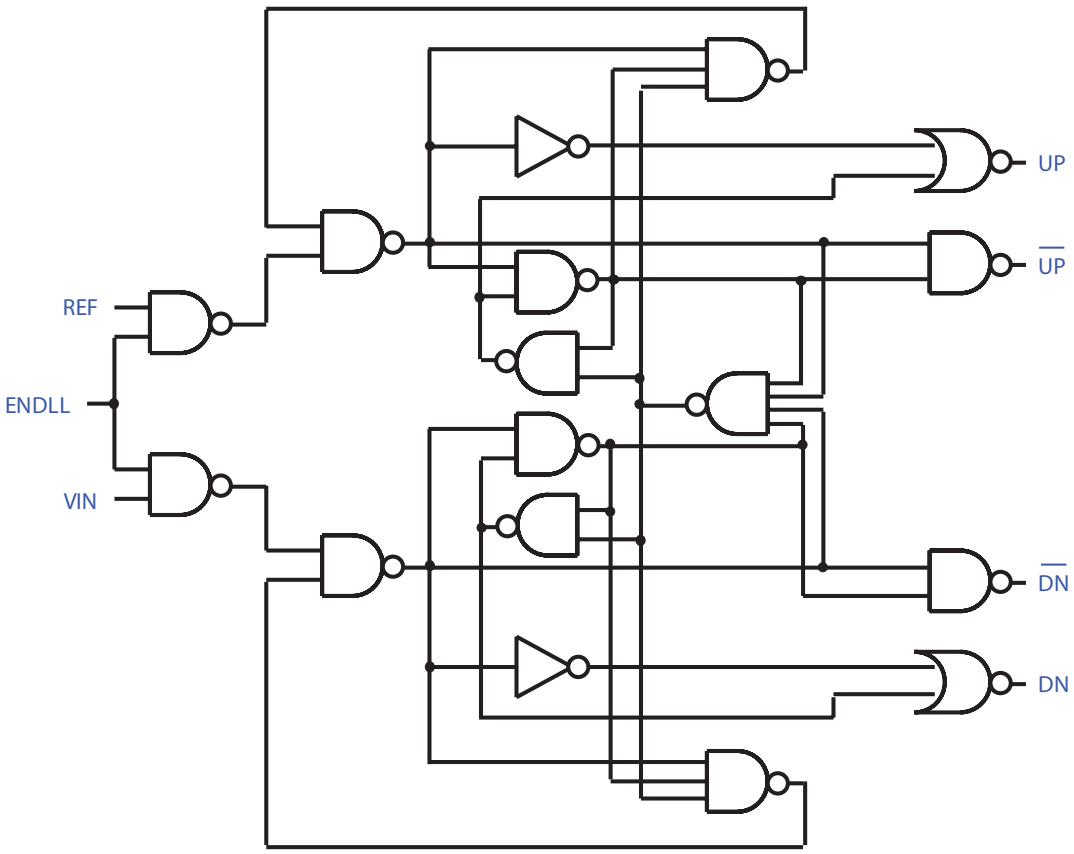

Figure 3.52: Phase-Frequency Detector Diagram

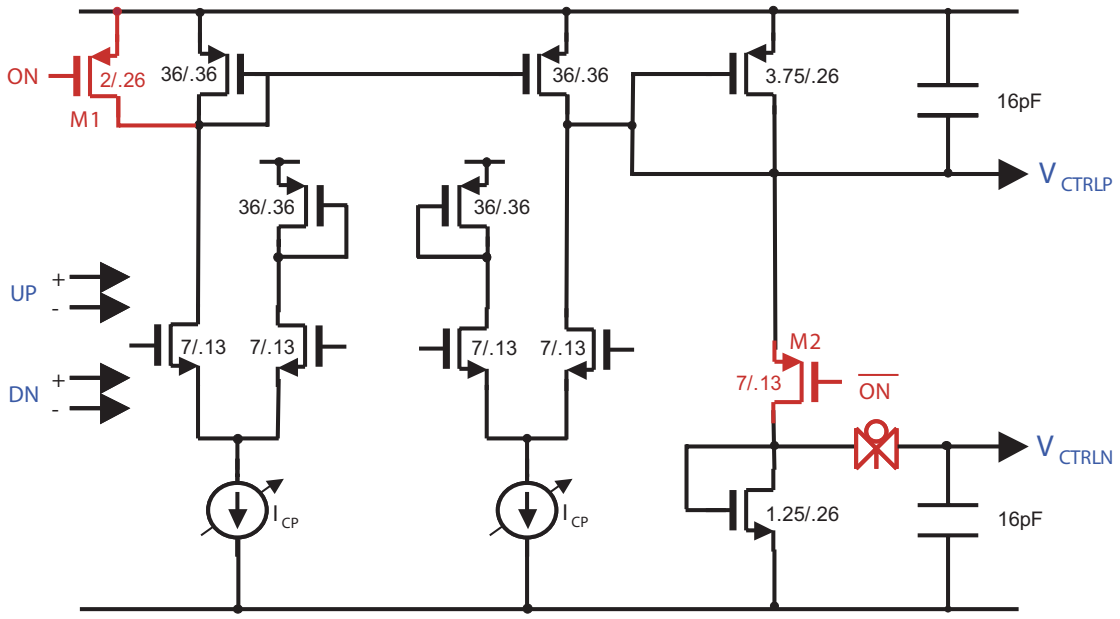

Figure 3.53: Charge Pump and Loop Filter Circuit Diagram

PMOS mirror to V_{DD} to guarantee that both devices connected to V_{CTRLP} are off when the DLL is disabled. ("UP" and "DN" are also all driven low when the DLL is disabled). Switch "M2" allows the user to turn off the replica mirroring that generates V_{CTRLN} from V_{CTRLP}. Note that the replica circuit is half-size relative to the delay cells to limit the static current consumption when active. The additional pass gate isolates V_{CTRLN} from the diode connected NMOS when "M2" is off (to prevent charge loss).

One problem observed with this circuit is that the time-domain profile of the up and down pulses are not well-matched, even if the total charge is. The current pulse from the NMOS differential pair tends to be short in time, resulting in a spike. Whereas the capacitance on the PMOS mirror tends to smooth out the pulse, resulting in a more leisurely application of charge. This difference in time creates a brief, momentary perturbation in

the control voltage. No good, obvious solution to this problem was discovered other than increasing the size of the loop filter capacitors to minimize the disturbance. Because we are using finely-spaced delay elements, this perturbation could slightly skew the first tap or two in the delay line, resulting in non-uniform sampling for the ADC.

Matching between the "UP" and "DN" pulses (or mismatch caused by leakage) is very important for this DLL because it is dividing the input clock by 32. Any error or mismatch over the whole cycle is also applied over each of the phases, accumulating to the largest value at the end of the delay line. This causes a larger proportional error for later taps, as opposed to earlier ones. The issue can become more critical as the DLL is duty-cycled and the update rate (the rate at which pulses are compared and charge is applied to the control voltage capacitor) is reduced. A small leakage current integrated over a slow update rate, can grow to a quantity of charge that cannot be compensated for with only a couple pulse input adjustments. (This is, in fact, seen in the measurements of the last tap of the DLL for duty-cycled operation in figure 3.57. Note how the offset grows are the update rate is decreased.) Care was taken with the design to ensure that the DLL would operate acceptably over an update rate from $60MHz$ to $600kHz$.

3.3.4 Measurements

To isolate and test the DLL separately, the oscillator clock was overridden and an Agilent 81134A $3.35GHz$ pulse/pattern generator was used as the input clock source. The Agilent pulse/pattern generator input was measured to have $1.7ps$ of cycle-to-cycle jitter. Data was measured with the Agilent Infiniium 54855A $20GHz$ Digital Sampling Oscilloscope. Measured results were taken at $V_{DD} = 1.1V$ and room temperature ($25°C$).

The DLL delay per tap were measured and are shown in figure 3.54. The delay per tap relative to an ideal delay chain is shown, followed by the error relative to that ideal delay (both the calibrated and uncalibrated error), and finally the standard deviation of the jitter observed on each tap relative to the input clock. A linear curve fit predicts approximately $1ps$ of added jitter (standard deviation) per delay stage. The total jitter seen at the last tap is $\sim 15ps$, better than the hand analysis predicted ($\sim 22ps$). Figure 3.55 shows a close-up of the calibrated error and cycle-to-cycle jitter standard deviation per tap. Figure 3.56 shows the measured delay values used for calibration. Because routing through the chip and testboard could not be kept identical in length (due to pad location, etc.), the ability to send the same output clock to all taps was created to allow for the calibration of this discrepancy. The calibration delay profile matched expectation based on die and board routing lengths.

Duty-cycled operation of the DLL was also examined. The top of figure 3.57 shows the measured delay error in the last tap of the delay line (expected to be the worst case if a mismatch exists) relative to the ideal expected delay. The bars at each point indicate the 1-sigma cycle-to-cycle jitter standard deviation measured at those update ratios. Below, in figure 3.57 the 1-sigma cycle-to-cycle standard deviation of the last tap is shown for the first 5 cycles of operation during duty-cycled operation. If there were a transient phenomenon upon reactivation of the DLL after the wait between pulses we would expect to see a change in the values. However, the measurements were flat, indicating no such perturbation. The current consumption of the DLL for these update ratios is shown in figure 3.58. Note that power consumption falls below $50\mu A$ for a 5% update ratio (while the DLL still maintains

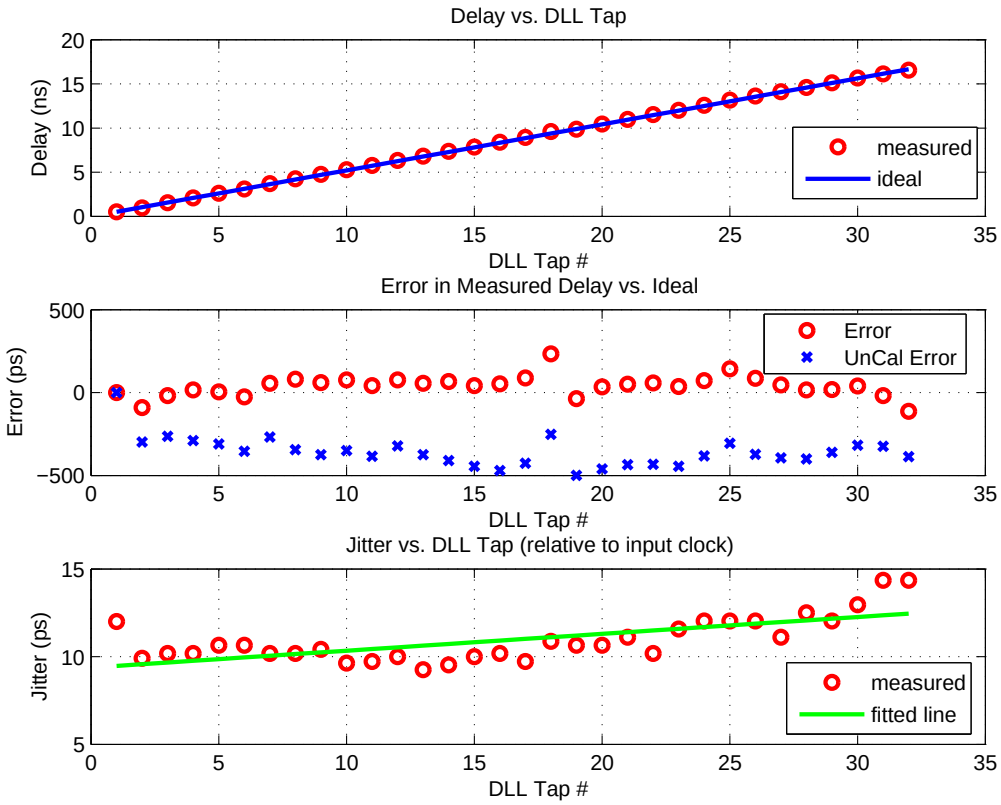

Figure 3.54: DLL Delay Measurements

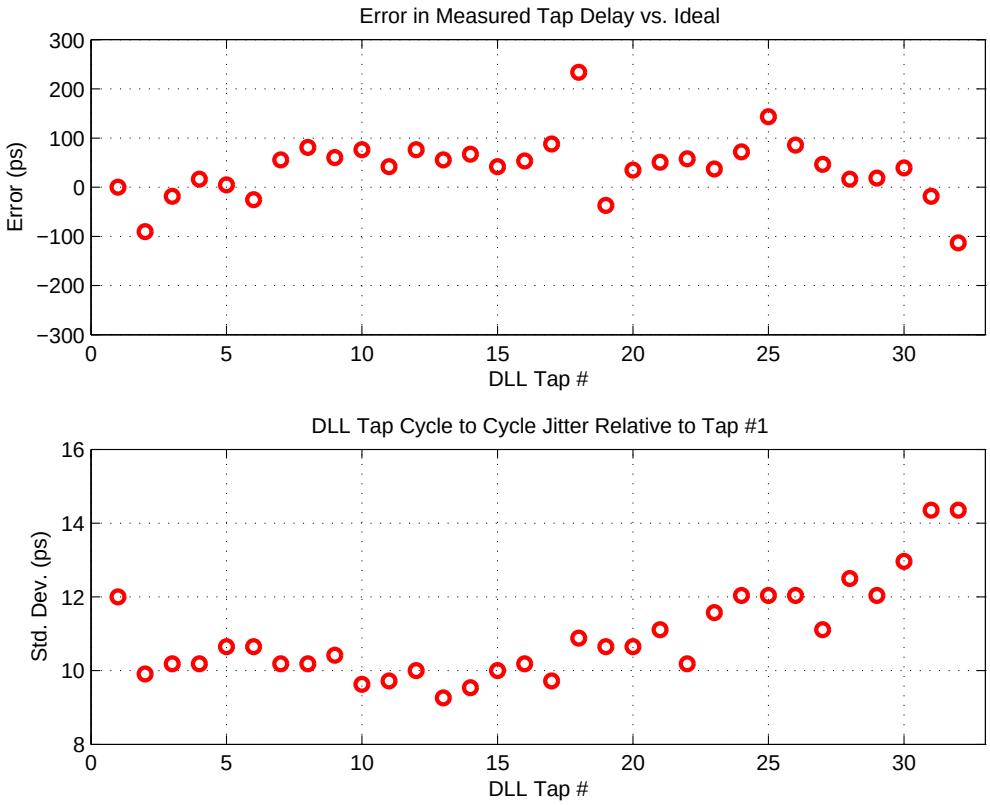

Figure 3.55: DLL Error StdDev Measurements

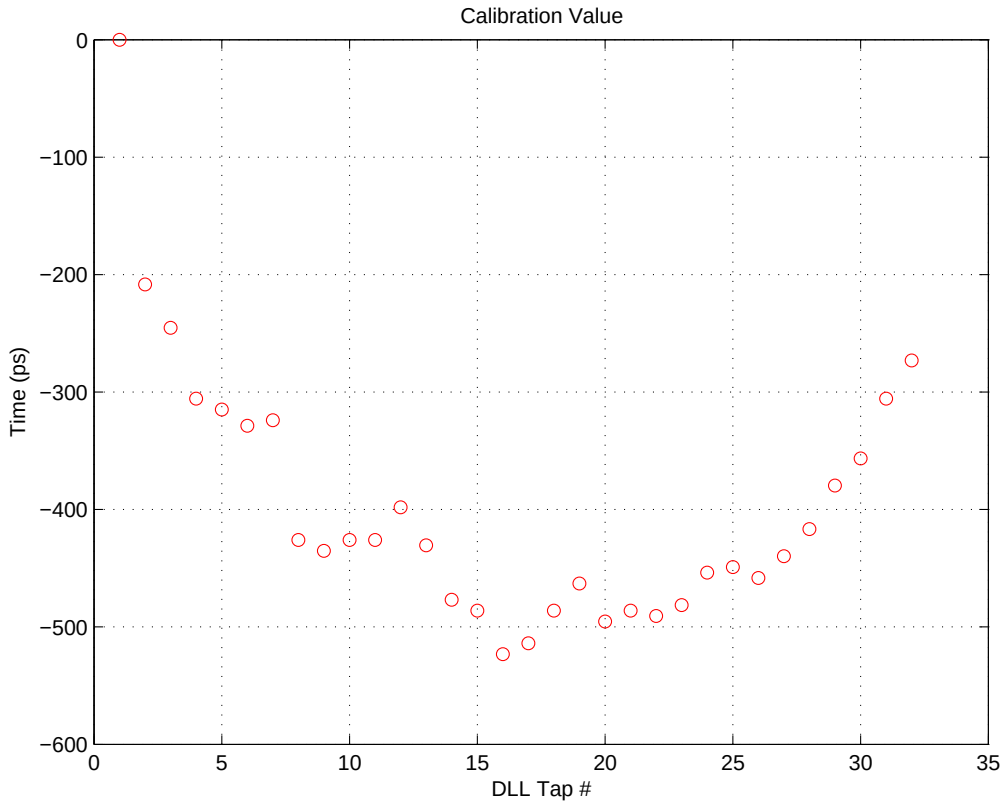

Figure 3.56: DLL Calibration Measurements

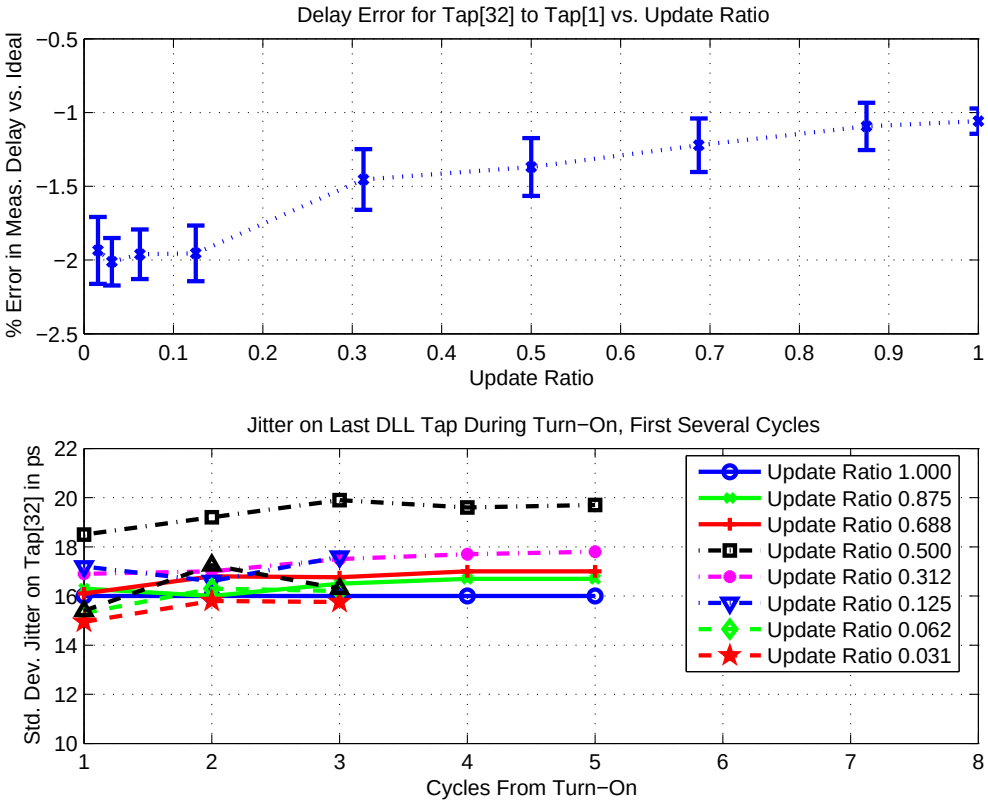

Figure 3.57: DLL Duty-Cycled Measurements

accurate $1.92GHz$ sampling!)

The impact of the charge pump current on DLL operation was also measured. Figure 3.59 shows the delay, delay error, and cycle-to-cycle jitter over four charge pump bias current values for the last 4 taps in the delay line. Here, if a significant leakage current existed at the charge pump summing node, V_{CTRLP}, we would expect to see performance degrade for low charge pump current. No change was observed, indicating that no significant leakage current exists.

Finally the capture range of the DLL was measured. Figure 3.60 shows the delay

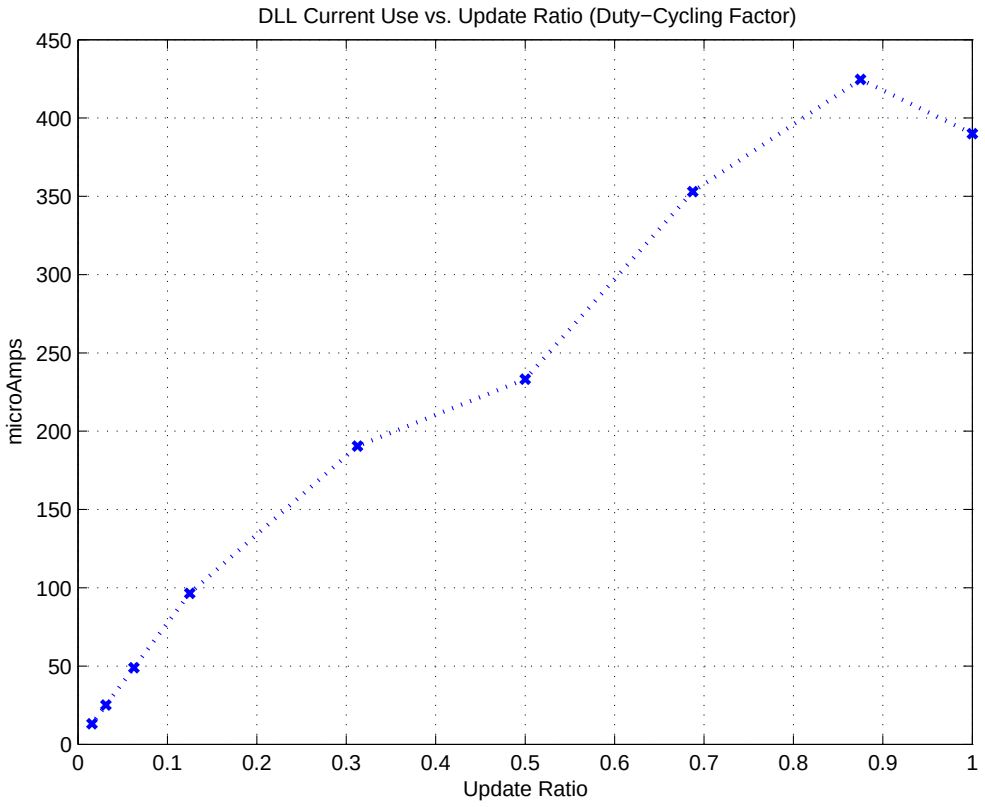

Figure 3.58: DLL Duty-Cycled Current Consumption

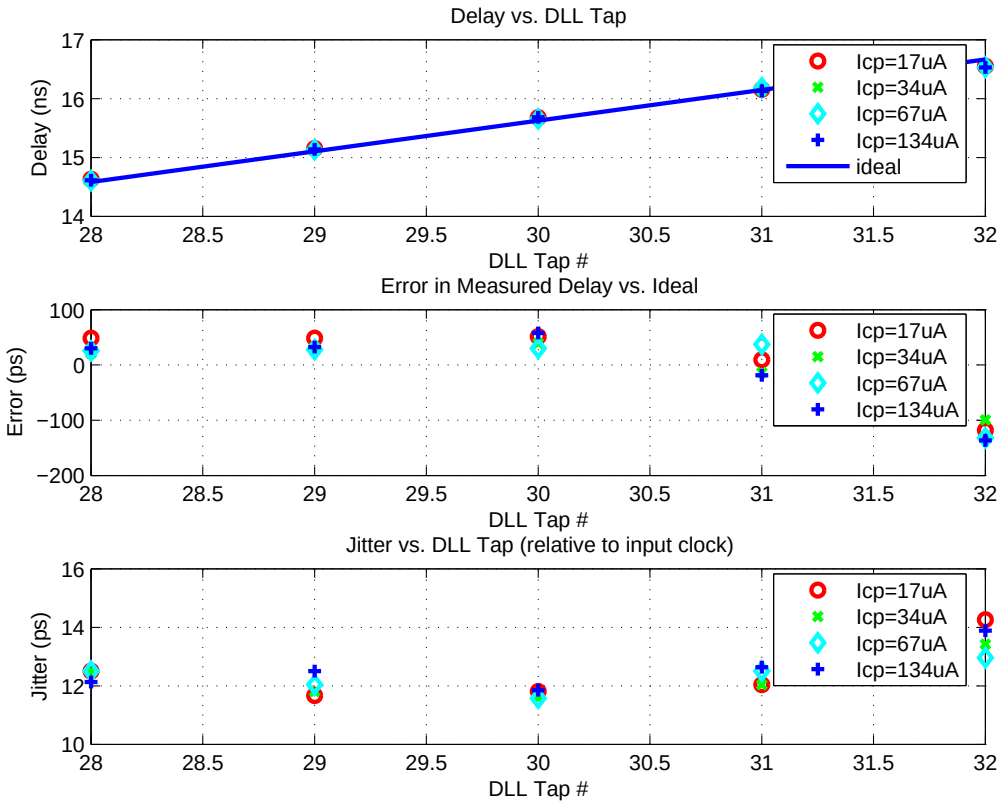

Figure 3.59: DLL Charge Pump Current Measurement

tap measurements, error, and jitter (normalized to a percent of the input clock period, called the "unit interval", or "UI"). Again only the last four taps were measured as any expected offset from charge pump imbalance or leakage would be largest towards the end of the delay line. Measurements indicate that the DLL locks over an octave of frequency, as per design, from $32MHz$ (or $1.024Gsample/s$ ADC operation) up to $62.5MHz$ (or $2.00Gsample/s$ ADC operation). (The choice of $60MHz$ operation was dictated by crystal availability.) The capture range is on the low side of the target (an octave centered around $62.5MHz$: $90MHz$ to $45MHz$) because the layout capacitance was larger than expected (although the chip run wound up closer to the "slow" corner than "typical" as well).

3.4 Analog to Digital Converter

System simulations indicate that a 1-bit ADC is adequate. A 1-bit ADC reduces to a sampler and a comparator (or 'slicer') which decides the sign of the sampled input. There are two elements to this operation: 1) the sampling switch, which takes a sample in time of the input voltage and holds it, while 2) a comparator resolves to the sign of the sampled value. The main performance criterion, as mentioned in section 2.10 is the ADC offset voltage. (As we will see, the ADC speed relative to our cycle time ($1/60MHz = 16.667ns$) is not a concern.) A smaller offset voltage allows for less gain which results in direct power savings. However, a smaller offset voltage also implies larger devices (for matching) and/or more complicated comparator structures (e.g. comparators that sample their offset and attempt to cancel it) which will cause the ADC power to rise. It should be noted that an increase in ADC power is felt 32 times over, as there are 32 ADC's operating in parallel.

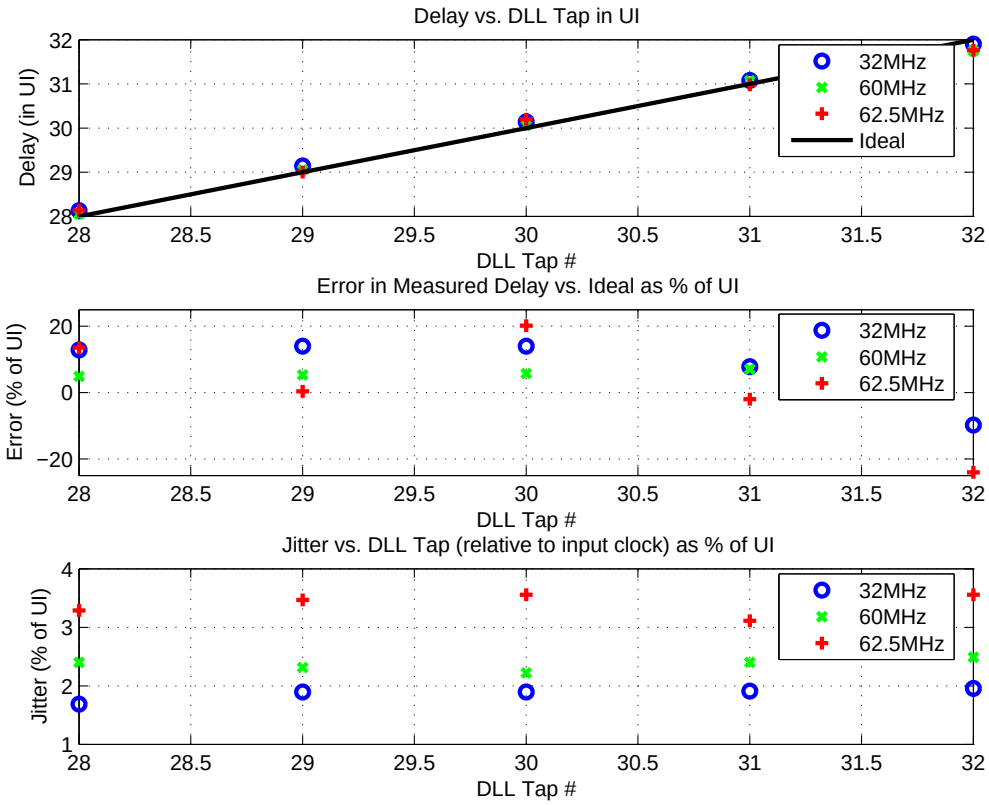

Figure 3.60: DLL Capture Range Measurement

This weights power heavily in the ADC direction. For example, removing a variable gain stage (with $7dB$, or $2.2\times$, of gain) saves approximately $150\mu A$. To save a factor of 2 in gain implies that our standard deviation has decreased by 2 (to keep the same probability of error in our analysis). This factor of 2 decrease in V_{OS} must happen for no more than a $150\mu A/32 = 4.7\mu A$ increase in current consumption per comparator. Taking the power consumption as proportional to gate capacitance switching ($\propto WL$, ignoring the short-circuit current for now), the offset voltage is proportional to the square root of the gate dimension ($\propto \sqrt{WL}$) [108]. A power optimization would push power back onto the gain stages and opt for a smaller comparator (with a relatively higher offset) driven by a larger amplitude input.

3.4.1 Sampler

With that power/offset trade-off in mind, we now turn towards the sampler design. It is important to get an accurate model of the offset contribution from the sampler as part of the overall ADC offset. While it is possible to sample current, we have chosen to sample voltage in order to avoid static current consumption. The simplest sampling element is just a switch, as shown in figure 3.61. The main sources of offset for a sampling switch come from: channel charge injection[109], overlap capacitive coupling, and kT/C noise from the channel (integrated onto the sampling capacitor)[97]. Unfortunately offset tends to increase for minimum sized sub-micron CMOS as the dimensions shrink and offsets on the order of 10's of millivolts to 100's of millivolts have been reported [110]. Luckily, we are sampling differentially which tends to cancel the offset contribution. If the signals are very close (a case where offset would be of critical importance) the contributions from capacitive coupling

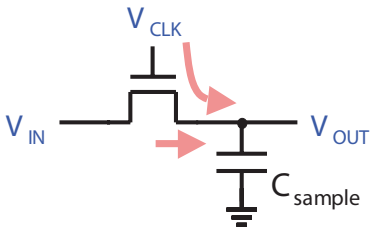

Figure 3.61: Sampling Switch Diagram

and change injection cancel to first order.

To determine the offset of the sampling switch, the offset contributions from each source is considered:

- 1. Sampling capacitance mismatch
- 2. Gate WL mismatch
- 3. Threshold voltage mismatch
- 4. Overlap Cap mismatch

1. Calling C_1 and C_2 our sampling caps, and taking $C_1 = C + \Delta C/2$ and $C_2 = C - \Delta C/2$, then we can write the contributions from charge injection (Q_{inj}), overlap capacitance (C_{ov}) and noise (kT/C) as:

$$V_{OS1}^{kT/C} = \sqrt{2kT/C} \tag{3.9}$$

$$V_{OS1}^{C_{ov}} = -\frac{C_{ov} * C}{(C + C_{ov})^2} * V_{CLK} * \frac{\Delta C}{C} \tag{3.10}$$

$$V_{OS1}^{Q_{inj}} = \frac{WLC_{OX}}{C} * (V_{CLK} - V_{IN} - V_T(V_{IN})) * \frac{\Delta C}{C} \tag{3.11}$$

2. Looking at the gate WL mismatch, we see that this mainly effects the charge injection. (Overlap and noise contributions are second order, so may be ignored.) The

resulting offset effect is:

$$V_{OS2}^{Q_{inj}} = \frac{WLC_{OX}}{C} * (V_{CLK} - V_{IN} - V_T(V_{IN})) * \frac{\Delta WLC_{OX}}{WLC_{OX}} \tag{3.12}$$

3. The threshold mismatch also mainly effects the charge injection contribution:

$$V_{OS3}^{Q_{inj}} = -\frac{WLC_{OX}}{C} * \frac{\Delta V_T}{V_T} \tag{3.13}$$

4. Finally, overlap mismatch effects mainly the overlap contribution:

$$V_{OS4}^{C_{ov}} = \frac{C_{ov} * C}{(C + C_{ov})^2} * V_{CLK} * \frac{\Delta C_{ov}}{C_{ov}} \tag{3.14}$$

Summing up these contributions, we can estimate the overall offset contribution for differential sampling as:

$$V_{OS} = \sqrt{\sum_i \left(V_{OSi}^*\right)^2} \tag{3.15}$$

Note that offset also may arise from differences between the tracking bandwidth of the switches. We assume the designer will size the switch appropriately so that most of the signal energy is below the τ of the tracking bandwidth. This will drastically reduce the effect of a small mismatch in tracking bandwidth.

Figure 3.62 shows the expected offset contributions by source vs. C_{sample} which is swept from $1fF$ to $1pF$ for three process corners: slow, cold; typical; and fast, hot. In all cases the channel charge injection dominates the offset contribution. The overlap capacitance is sometimes a close second, and the kT/C noise barely has a discernible effect. Figure 3.63 shows the expected aggregate offset standard deviation for differential sampling vs. sampling capacitance size over process corners. The switch is scaled with the sampling capacitance to keep a constant $1GHz$ tracking bandwidth. For a minimum sized switch,

increasing the sampling capacitance value decreases the overlap, charge injection, and kT/C noise contributions. As the sampling capacitance increases, though, a point is reached where the switch must be sized up proportionally as well (to maintain bandwidth) and hence the offset contributions are equivalently sized up and no more improvement is seen. This is expected to occur around $40fF$ in the typical case.

3.4.2 Comparator

A survey of low power ADC publications was made and, to the author's surprise, there was no obvious agreement about which comparator architecture presented the best power consumption vs. offset trade-off. [111] presents a good overview of comparator design but does not specifically address this trade-off. Speed and power consumption trade-offs have been covered for latches and flipflops [112] but not comparators. (Comparators are often simply latches preceded by gain and sampling.) As the comparator design space is quite large, this seems like a good candidate for future research. From a high-level perspective most reported comparator topologies were simplified into one of the three forms shown in figure 3.64. Topology A uses positive feedback to resolve from an initial voltage state that is representative of the sampled voltage difference. Topology B pushes or pulls a current related to the sampled voltage difference into (or out of) a positive feedback cell. Topology C uses a transresistive technique to choke the current in the positive feedback cell (and feedback a voltage to the adjacent current leg). Gain stages were commonly placed between the sampling switch and feedback cell to reduce charge kickback (provide isolation) and to reduce the input referred offset voltage (provide gain). As previously mentioned at the beginning of this section, placing gain before offset can be a more power efficient way

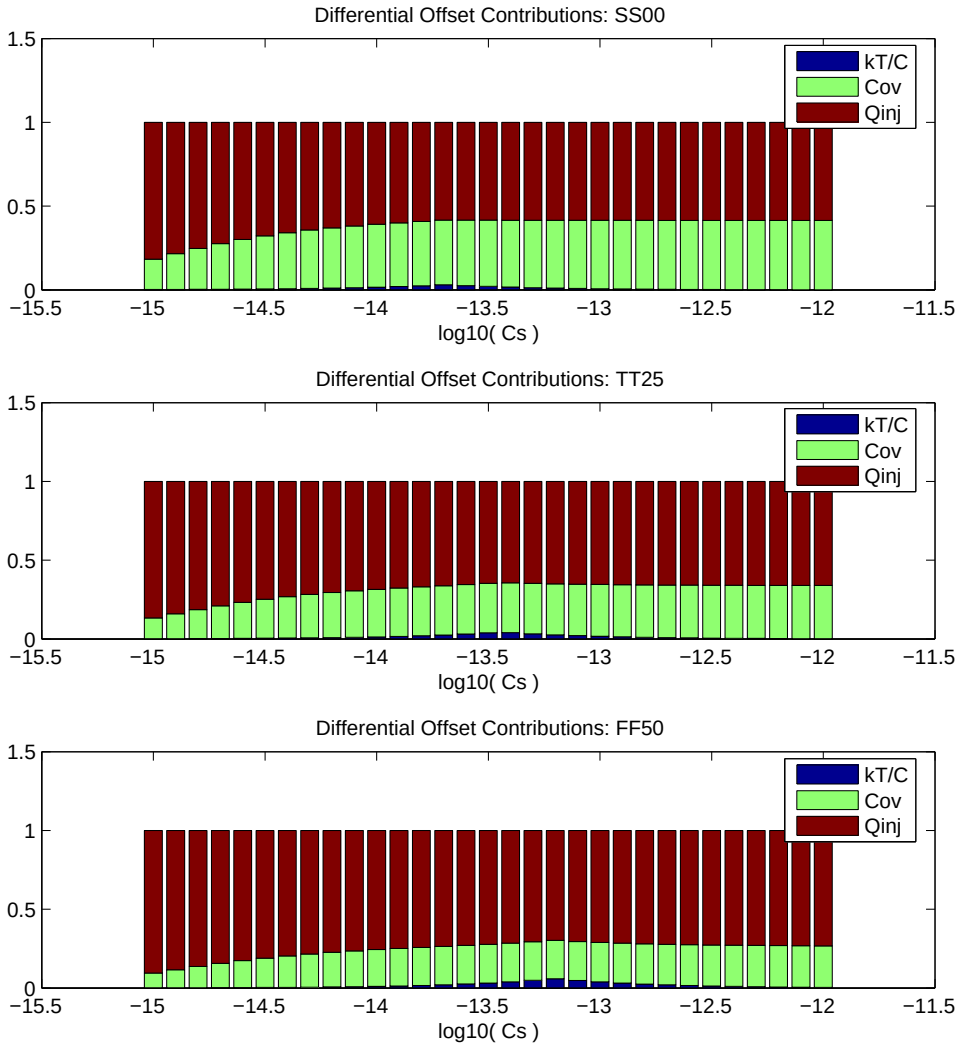

Figure 3.62: Differential Sampling Offset Contributions

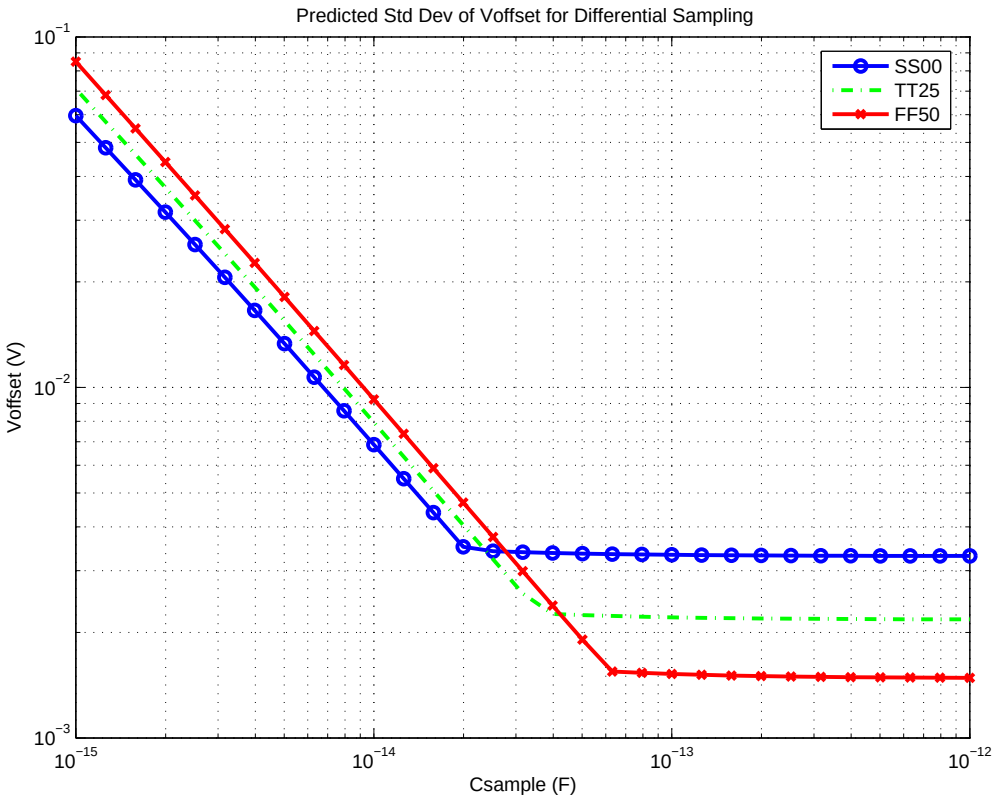

Figure 3.63: Predicted Differential Sampling Offset Standard Deviation

to achieve a desired input-referred offset as opposed to increasing device size to lower the offset to the same target.

Selecting a low power, low offset comparator topology was not obvious so a few "rules of thumb" were employed: no static current was allowed, and complexity was eschewed. Static current is avoided because a small increase in comparator power is multiplied by all 32 parallel comparators into a large overall increase in power. Complexity was avoided as it may also lead to larger power consumption through an increase in switched capacitance or extra control signal generation. It was not possible to provide static gain for the full signal bandwidth before every latch at the allowable power levels. Thus the preceding gain stages were considered to be part of the comparator design and optimization: providing low-offset gain to overcome the sampler and latch offset. Note that this means that charge will be kicked back into the gain stage output when a sampling switch opens. An explicit reset cycle is included to drive both outputs of the comparator to a roughly mid-band voltage before opening for sampling to render the kickback common-mode. To gain insight into latch performance, several simple examples of each topology were simulated for the same size devices. Power consumption, offset, and resolution speed were examined and a voltage-sample based topology, was selected. Consisting of a cross-coupled inverter core with power supply gating devices, this topology[113] [114] is shown in figure 3.65.

The cross-coupled latch offset voltage was analyzed by contribution resulting in the following equations:

$$V_{OS1}^{V_{tp}} = \frac{2 * \Delta V_{tp}}{\left(1 + \sqrt{k_n'/k_p'}\right)} \tag{3.16}$$

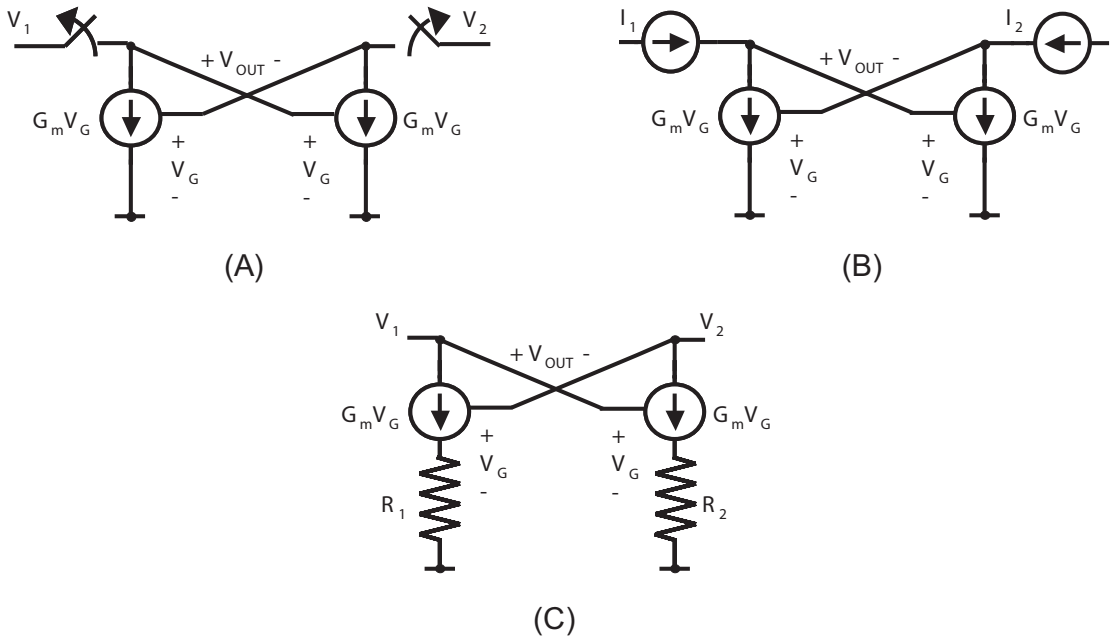

Figure 3.64: Comparator Topologies

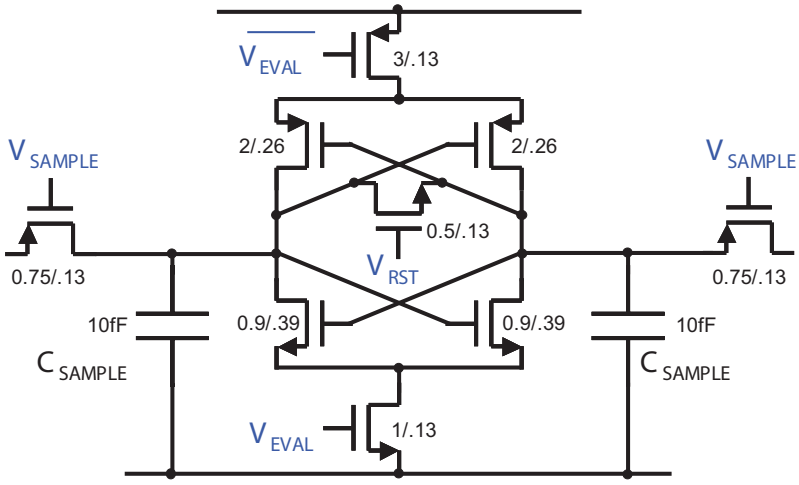

Figure 3.65: Comparator Circuit

$$V_{OS2}^{V_{tn}} = \frac{2 * \Delta V_{tn} * \sqrt{k_n'/k_p'}}{\left(1 + \sqrt{k_n'/k_p'}\right)} \tag{3.17}$$

$$V_{OS3}^{k_p'} = \frac{\sqrt{k_n'/k_p'}}{\left(1 + \sqrt{k_n'/k_p'}\right)^2} * \left(V_{DD} - V_{tp} - \sqrt{k_n'/k_p'} * V_{tn}\right) * \frac{\Delta k_p'}{k_p'} \tag{3.18}$$

$$V_{OS4}^{k_n'} = \frac{\sqrt{k_n'/k_p'}}{\left(1 + \sqrt{k_n'/k_p'}\right)^2} * \left(V_{DD} - V_{tp} - \sqrt{k_n'/k_p'} * V_{tn}\right) * \frac{\Delta k_n'}{k_n'} \tag{3.19}$$

A monte carlo simulation was performed in ELDO (a SPICE variant) and the standard deviation of the offset was found to be $6.5mV$ for the initial device size choices. Some fine sizing was performed to reduce the offset to $5mV$ and power consumption to $12\mu A$ (simulated at $60Mconversion/s$). The monte carlo simulation results of static latch offset are shown in figure 3.66. Regarding sizing, as expected the improvement in offset due to increased sizing was counterbalanced by a larger increase in power. However, it is worth noting that nearly minimum size devices, which have lower power consumption, exhibited offset greater than $16mV$.

3.4.3 Control Logic

To clock the comparator we need to generate "reset", "sample", and "evaluate" control signals. We also need a "clk" to latch and hold (or flop) the comparator output. These signals are derived from a delay line with approximately $0.5ns$ spacing between each tap. Because we support duty-cycling, this delay line is not necessarily free-running. Thus we require state-based control logic to determine if the incoming pulse is the first (or last) to handle boundary conditions properly. Additionally as the last valid sample occurs at the end of the delay chain, approximately $0.5ns$ before the next rising edge, an entire extra

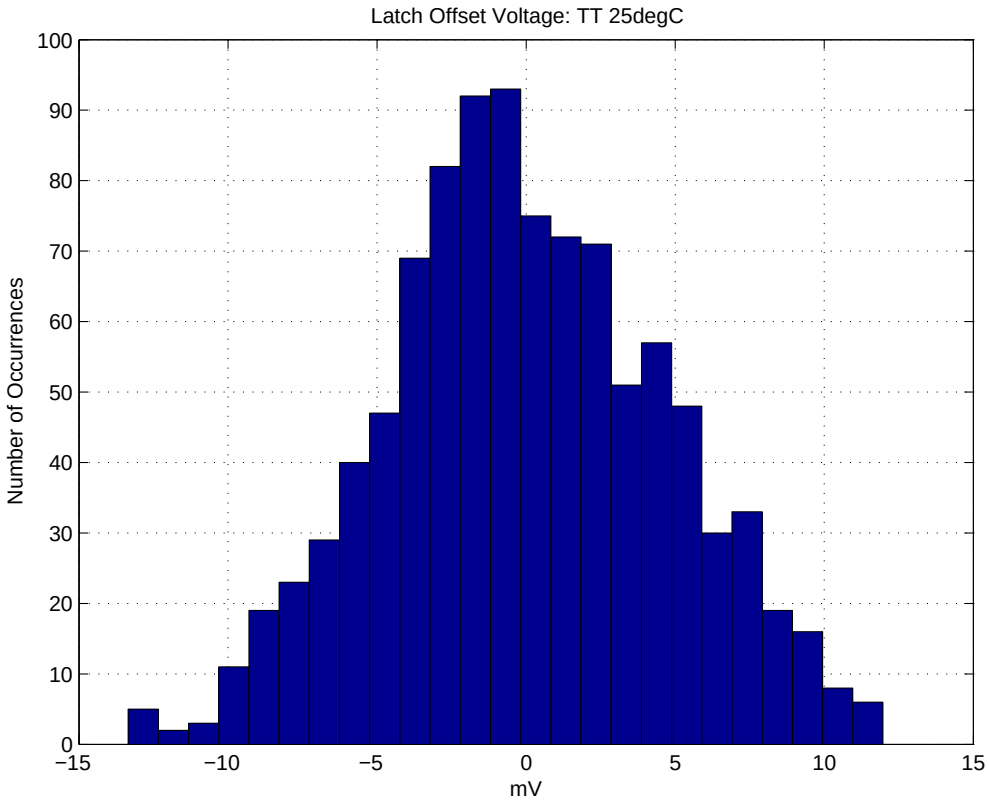

Figure 3.66: Latch Static Offset

cycle is needed to push that sample through. This results in some power consumption inefficiency, but is unavoidable.

The desired comparator control signals are shown in figure 3.67. The input signals are taken at "plusN" (modulo 32) from the DLL phases to generate the output control. For example, if phase N is "plus00" and starts sampling (ending reset), then the next phase $(N+5)\,mod32$ is used to end sampling (and start evaluating). In this manner, the DLL phases may be used to manage comparator sampling at an effective $1.92Gsample/s$ rate with 32 parallel $60MHz$ slices. The logic to implement this control is shown in figure 3.68. The input gates include a global reset (to reset state) and are implemented as a single, complex, static, complementary CMOS circuit. The output driving circuits are sized to ensure the non-overlap of control starting and stopping. The $60MHz$ clock period gives ample time to complete a comparator cycle. The expected input, generated from the DLL phases is shown in figure 3.69, illustrated for 5 input cycles (and hence, 4 output samples as we lose the last cycle). Figure 3.70 shows the comparator control signal state for that input, using colors to demarcate between the different states. The comparators stay in a reset state until triggered to sample. When sampling, the sampling switch is open for $2.5ns$ to allow for adequate settling. Then after sampling, they evaluate for nearly $12ns$, clocking the final result into a flipflop on the falling edge of the "clk" signal. (Note that the rising edge on "clk" is delayed for $7ns$ into evaluate to give the comparator time to resolve to a healthy magnitude such that any noise coupling in from the "clk" signal does not perturb the output.)

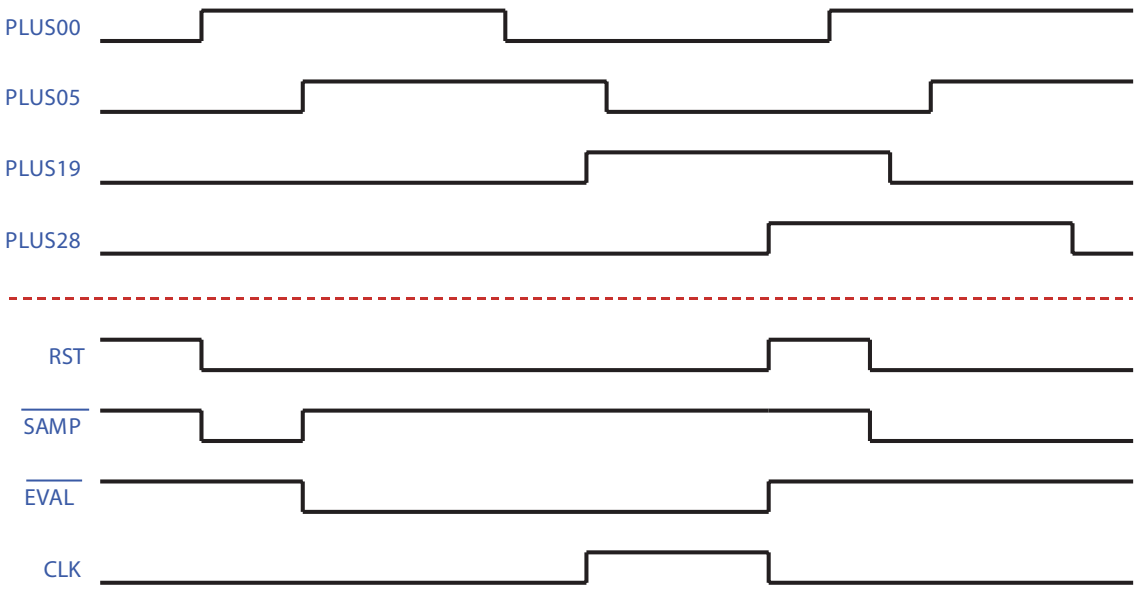

Figure 3.67: Desired Comparator Control Signals

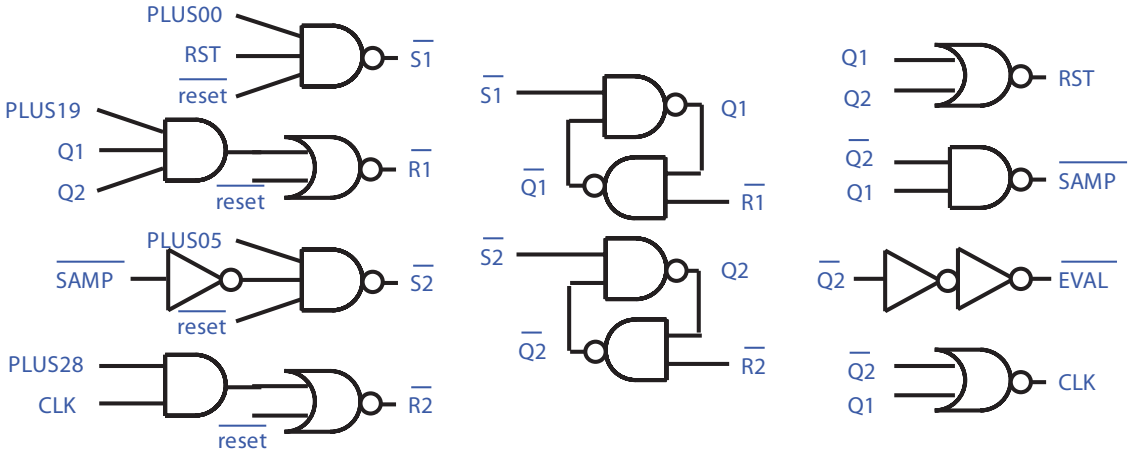

Figure 3.68: Comparator Control Logic Diagram

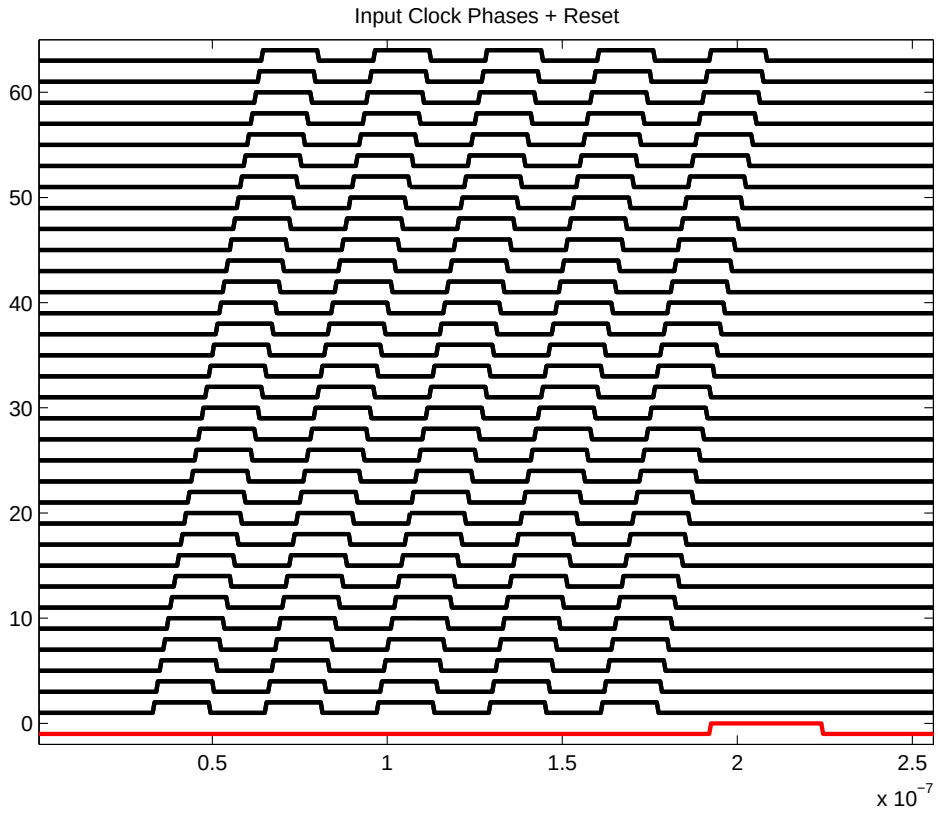

Figure 3.69: Input DLL Phases for ADC Control

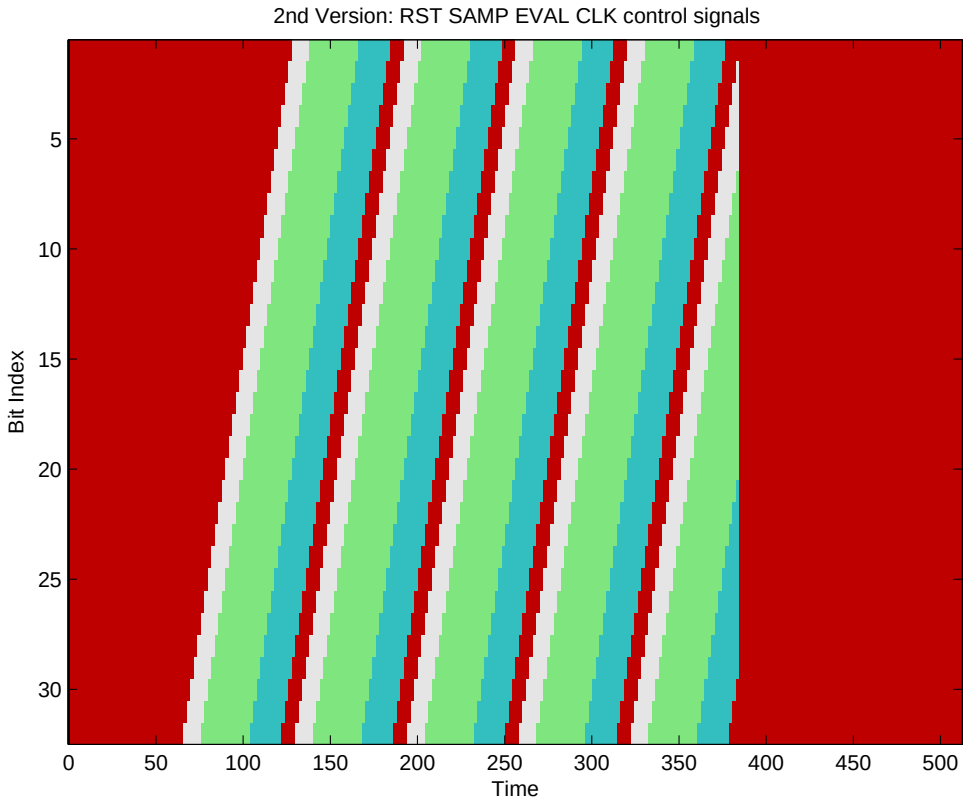

Figure 3.70: Generated Comparator Control Signals

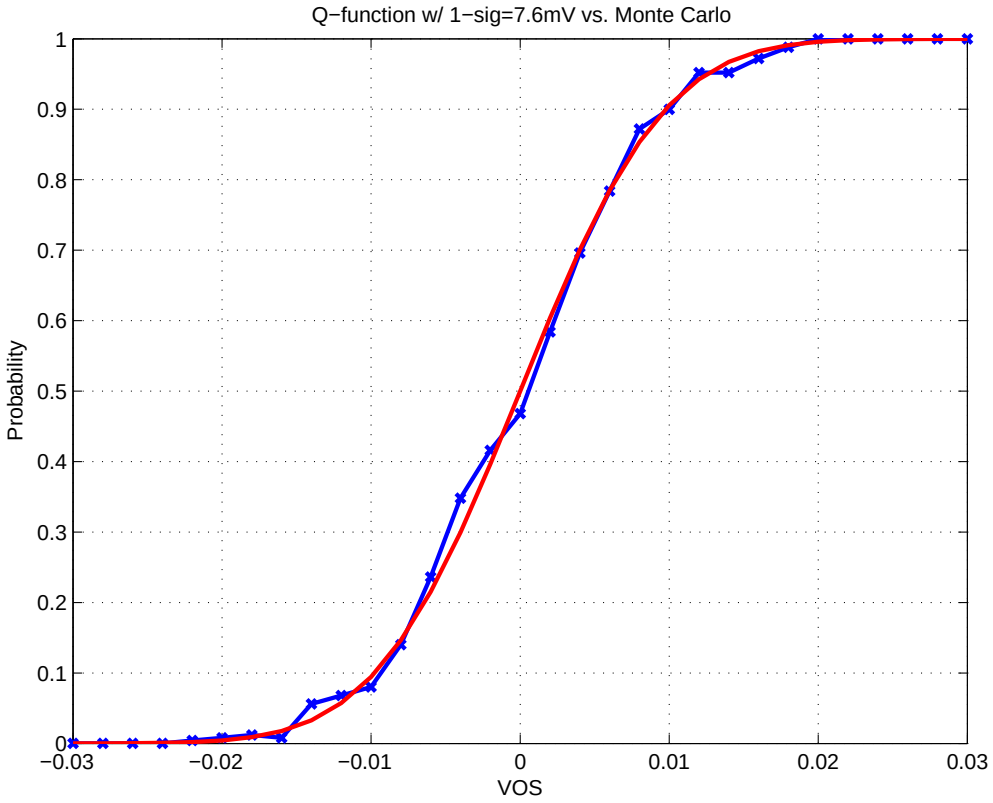

Figure 3.71: Monte Carlo Transient Simulation Results vs. Hand Prediction

3.4.4 Performance

Monte carlo simulations were performed in ELDO (a SPICE variant) for the ADC slice in the time domain as the input voltage was held constant for values spaced at 1mV intervals from $-30mV$ to $+30mV$. The number of times the simulation resolved high were counted to determine the probability that the comparator would resolve high. This result is plotted in figure 3.71 against the Q-function estimate based on our hand analysis which predicted an offset standard deviation of $7.6mV$ ($\sqrt{\sigma^2_{sample} + \sigma^2_{latch}}$). There is very good agreement between these curves.

In terms of measurements for the ADC slices, the specific offset of each comparator was not measured in the laboratory. Instead, the offset range for the ADC array was observed through driving a voltage onto the "OBS" debug pins. The entire comparator array switched over $30mV$ of input range. (Assuming 3-sigma coverage, this implies roughly a $10mV$ standard deviation for ADC offset which roughly agrees with the value expected from simulation and analysis.) Also, large waveforms from a pulse generator were fed into the ADC and the proper shape was observed at the output to verify operation. The measured current consumption was $13\mu A$ per conversion at $60MConversion/s$, very close to the predicted $12\mu A$ value. Due to time considerations the ADC array was not further characterized beyond this.

3.4.5 Layout

The layout for an ADC slice is shown in figure 3.72. At the top of the slice is the comparator, laid out to be both translational and mirror symmetric. Note that dummy devices appear to the right to ensure good matching when slices are tiled together. In the middle of the slice is a pass transistor register (see section 3.5.2) to hold the comparator decision. At the bottom is the control logic from figure 3.68. The control logic is kept at a distance from the comparator to reduce the possibility for noise coupling. (I.e. at the sensitive time when the switch closes and evaluation begins). An screenshot of the overall layout of the ADC (32 parallel slices) is shown in figure 3.73. (Dummy slices were also used on each end to maintain good matching.) The differential analog input is at the top. At the bottom is the 32-bit output bit bus and the control inputs are tapped off a 32-bit horizontal bus driven by a controlled window of DLL phase outputs.

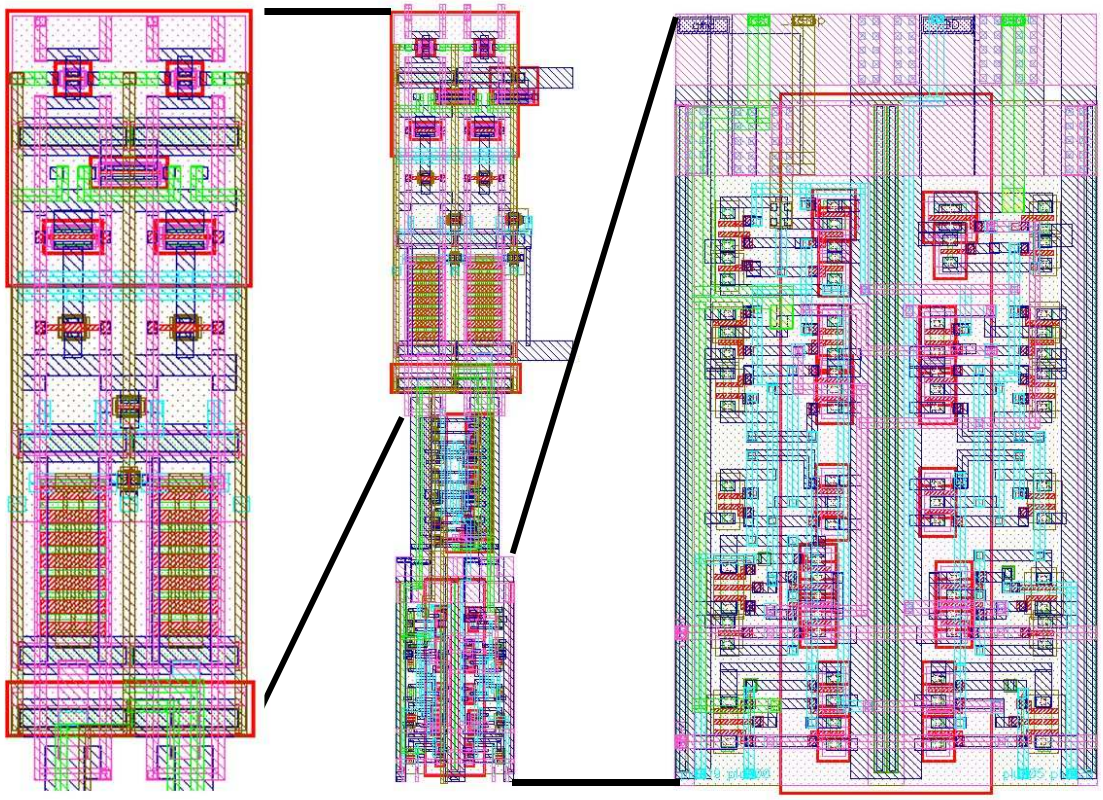

Figure 3.72: ADC Comparator Slice Layout

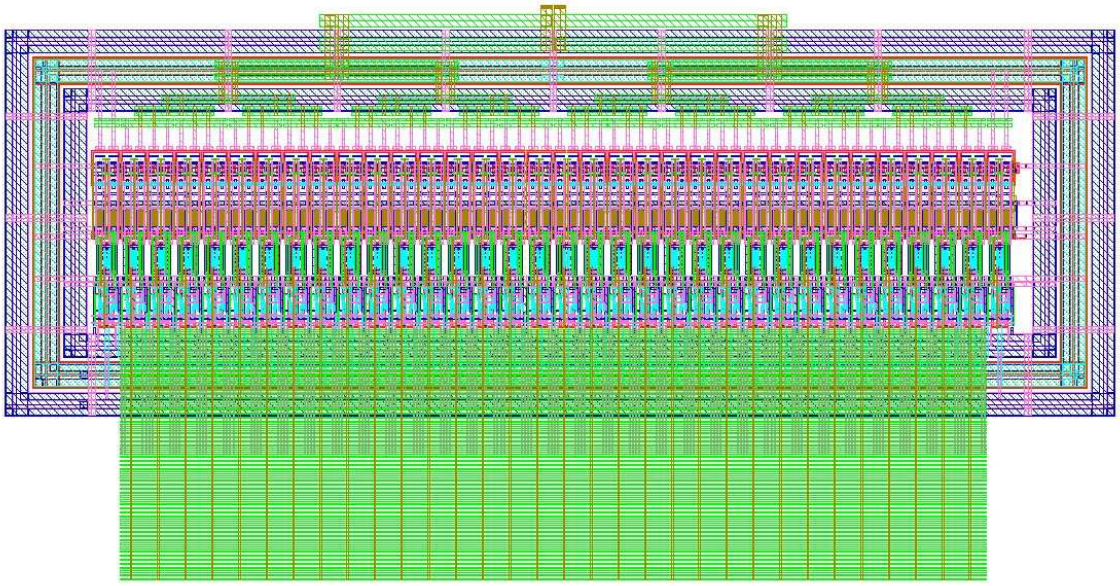

Figure 3.73: ADC Bank Layout

3.5 Digital

The digital control logic used for this project were hand designed and laid out. The circuits are generally implemented as normal, static complementary CMOS gates or pass transistor logic[115] [116]. No special attempt was made to decrease power beyond using a small device size and attempting to minimize capacitance. (I.e. voltage scaling, charge recovery, adiabatic, etc. techniques were not employed.) A number of well-known techniques exist [117] to implement low power digital logic and their discussion is beyond the purview of this dissertation. Additionally the control logic power consumption is low: budgeted at 10% of the total system power when fully active. Measured results indicate 14% at full activity, which scales to 19% at a $1Mpulse/s$ duty-cycled rate. Most of the control logic, with the exception of the global counter which is always active, exhibits reduced activity with

duty-cycled operation. If the digital backend computation were included, it would be more worthwhile to explore power reduction options. The digital backend discussed in section 2.5 from [1] was synthesized, placed, and routed using the same $0.13\mu m$ CMOS process as the front-end transceiver and the vendor's standard cell library. It occupies 245,000 cells in $10mm^2$ and simulations predict power consumption at $1.2V$ of $12mW$ during acquisition and $1.5mW$ during synchronization for a $10Mpulse/s$ pulse transmission rate. No special attempts to decrease the power consumption (i.e. voltage scaling, etc.) beyond careful system design were made. The acquisition power is large because of the large number of parallel search correlators, but this decreases the acquisition time commensurately, and hence the length of time that these correlators will be active.

3.5.1 Control Blocks

A simple block diagram of the control logic is shown in figure 3.74. From reception of a "Go" (or start) pulse, a delay counter waits until the output matches a programmable trigger level. The "Go" pulse may come an external pin or be internally generated through a register write. This allows the designer the flexibility to start operation at any particular phase or offset relative to the incoming pulse period (granular to the system $60MHz$ clock). After triggering, the cycle counter loops over a programmable range from start to reload. The count is used by the "CtrlGensig" module to generate block control signals by comparing it to separate start and stop registers for each block. The block control signals determine when the gain stages are biased (i.e. "On"), when a clock input is fed to the DLL, when the DLL charge pump is biased, when the DLL can make comparisons, and when to allow the ADC to sample. The "CtrlADC" block uses the ADCSamp control signal to mask the

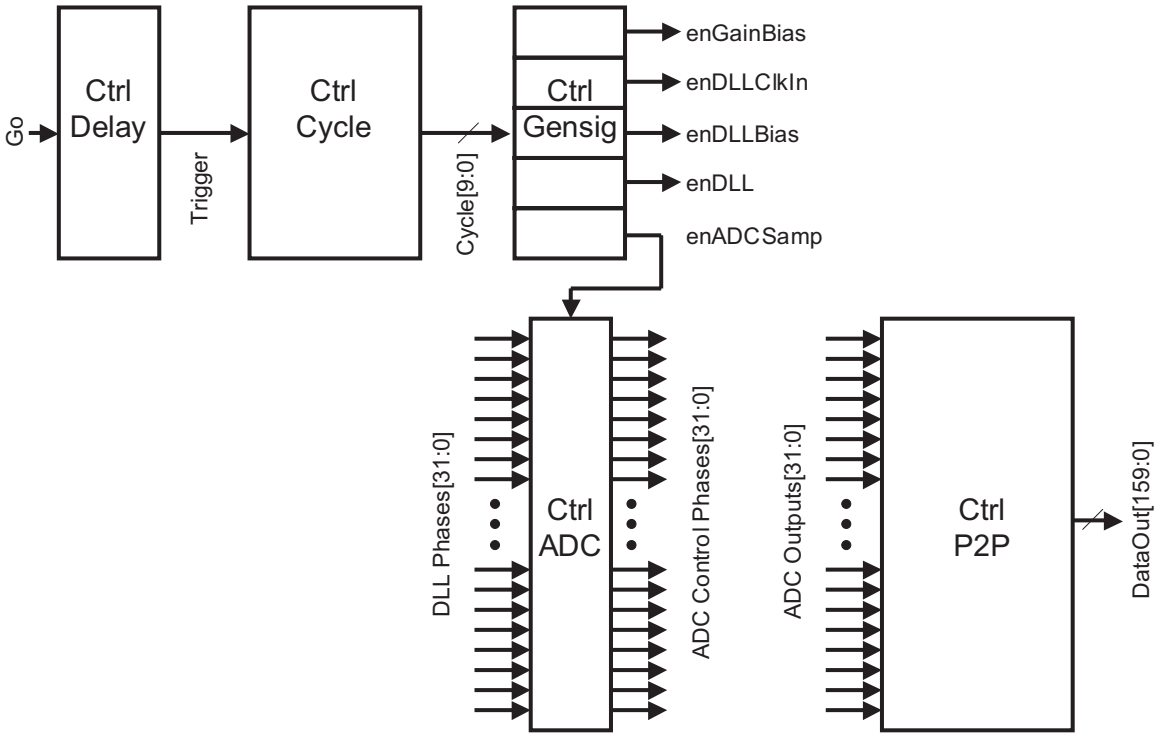

Figure 3.74: Control Logic Block Diagram

DLL output phases in the manner depicted in figure 3.69. (We cannot simply pass the DLL phases directly to the ADC at that point in time as some phases will be clipped or cut off. Instead a shift register is used to propagate the enable signal only during valid intervals to ensure the integrity of the phases generated for the ADC.) Similarly the outputs from the ADC's are all staggered by approximately $0.5ns$ in time and cannot be simply registered. Instead the "CtrlP2P" block re-times the ADC outputs and samples them relative to the falling edge of the system clock, aggregating up to 5 chunks of 32-bit outputs before passing them to the digital backend. (As the digital backend was not implemented on-chip, a 32-bit debug/observation bus was used to view the re-timed ADC outputs.) (See Appendix A for a complete description of the on-chip register space.)

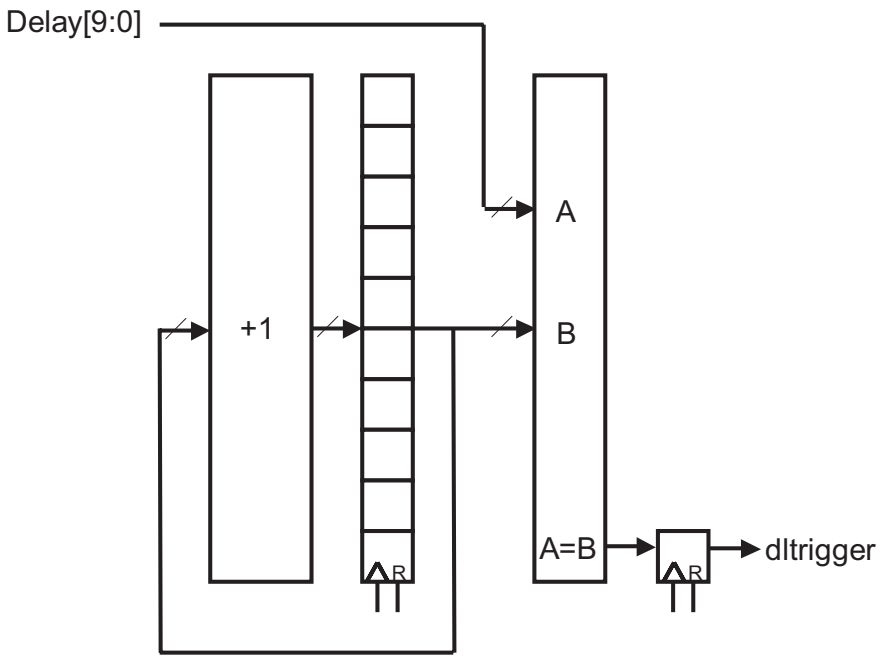

Figure 3.75: Control Delay Logic Block Diagram

CtrlDelay

Figure 3.75 depicts the "CtrlDelay" logic block diagram. Correct operation was measured using a logic analyzer to record the system debug/observation bus, as shown in figure 3.76. Note that the delay cell counts up until 'dltrigger' activates and the cycle clock begins looping. Because the 'go' pulse was left high during this run, the delay reactivates and also loops (this is harmless). A short pulse for 'go' would give one-shot operation for the delay block. The delay counter is the same length as the cycle counter (10-bits), allowing for arbitrary delay relative to the cycle period.

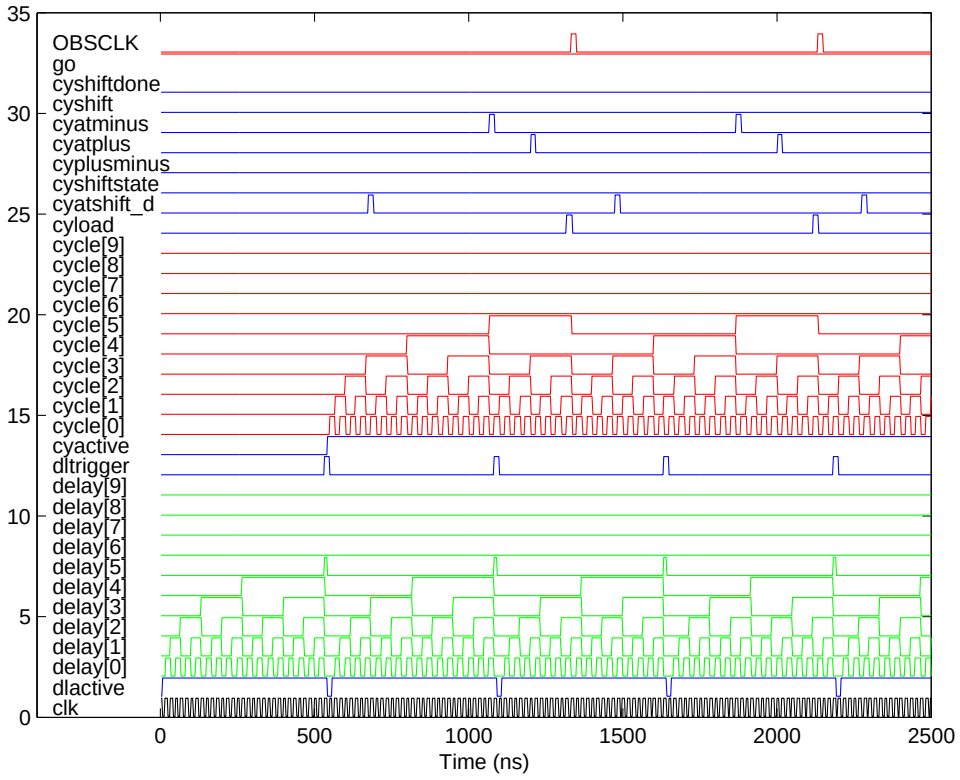

Figure 3.76: Control Delay Logic Measured Operation

CtrlCycle

Figure 3.77 depicts the "CtrlCycle" logic block diagram – the global counter. This counter may increment by +1, reload to start, or shift by a separate programmable amounts forward or backward, as controlled by comparators and state. The shift functionality allows the digital backend to feedback synchronization control signals to track drift or mismatch between the transmitter's pulse repetition frequency and the receive system clock. By skipping a count up or down, the receive controller may re-align the reception window relative to the pulse. Note that while this may also be achieved to some extent by pulling the oscillator (as mentioned in section 3.2), this digital shift functionality is necessary to maintain synchronization lock. Oscillator pulling was needed to allow less precise crystals to be employed while maintaining the accuracy necessary for long symbols as indicated in figure 2.16. However, with discrete pulling steps, it will not always be possible to exactly match the transmit and receive clock frequencies. This resulting mismatch would eventually cause the transmitter and receiver to drift from one another and lose synchronization. Thus, we require shift logic to track the receive pulse frequency mismatch. Note that the shift logic may also be used by the transmitter to shorten or extend the cycle count, creating a pulse-position modulated output. (The transmit pulse generation is trigger from the "load" signal in the control logic.)

To illustrate CtrlCycle operation the next three figures show the cycle value, load, and shift control signals along with the "GenSig" block control signals while in duty-cycled operation. Figure 3.78 illustrates an example duty-cycled operation for ADC sampling over three clock cycles. Figure 3.79 shows that same operation, but with shift active in

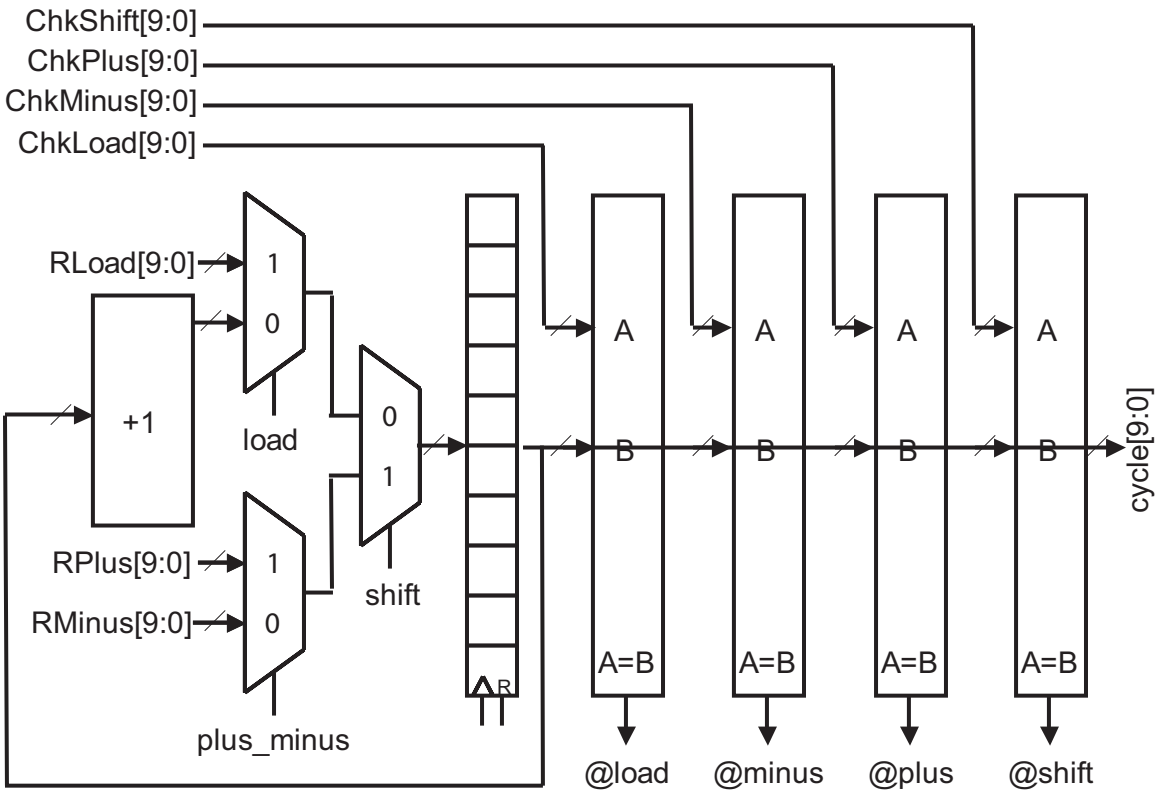

Figure 3.77: Control Cycle Logic Diagram

the 'minus' direction. (Note that as the comparison and load register values are fully programmable, 'minus' or 'plus' may shift in an arbitrary direction. The nomenclature used is consistent with the direction expected by the digital backend synchronization design.) Figure 3.80 shows the same operation, but with shift active in the 'plus' direction. Finally, as a contrast non-duty-cycled, or "always on", operation is shown in figure 3.81. In this case once the block control signals activate, they stay on until explicitly reset. This situation corresponds to operation at the fastest pulse rate.

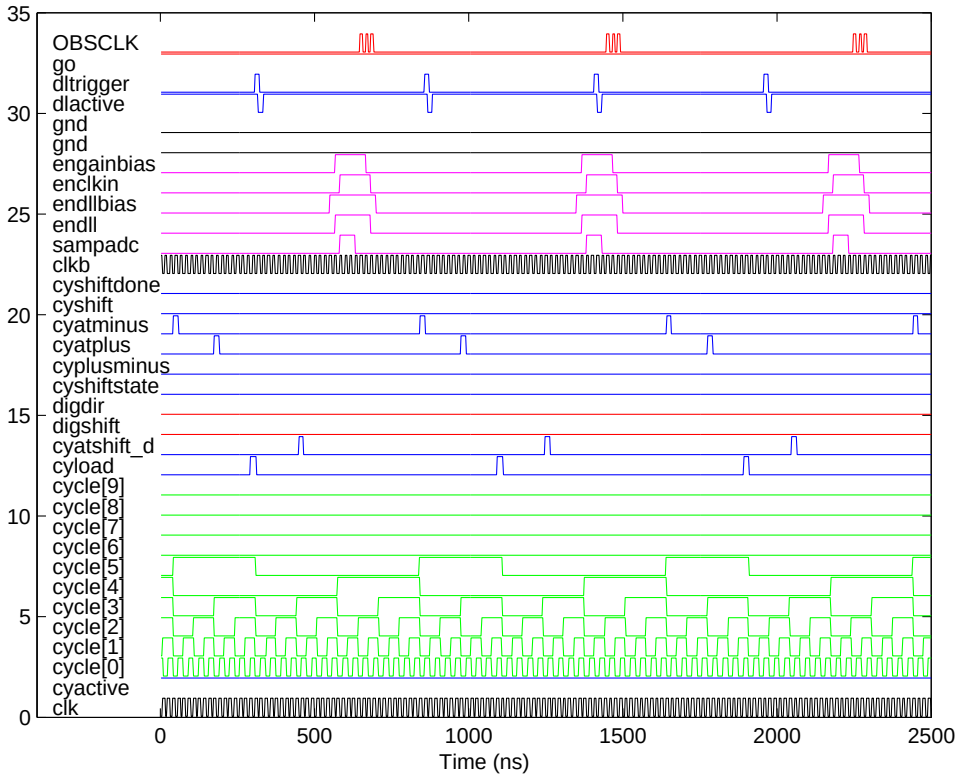

Figure 3.78: Control Cycle Operation, No Shifting

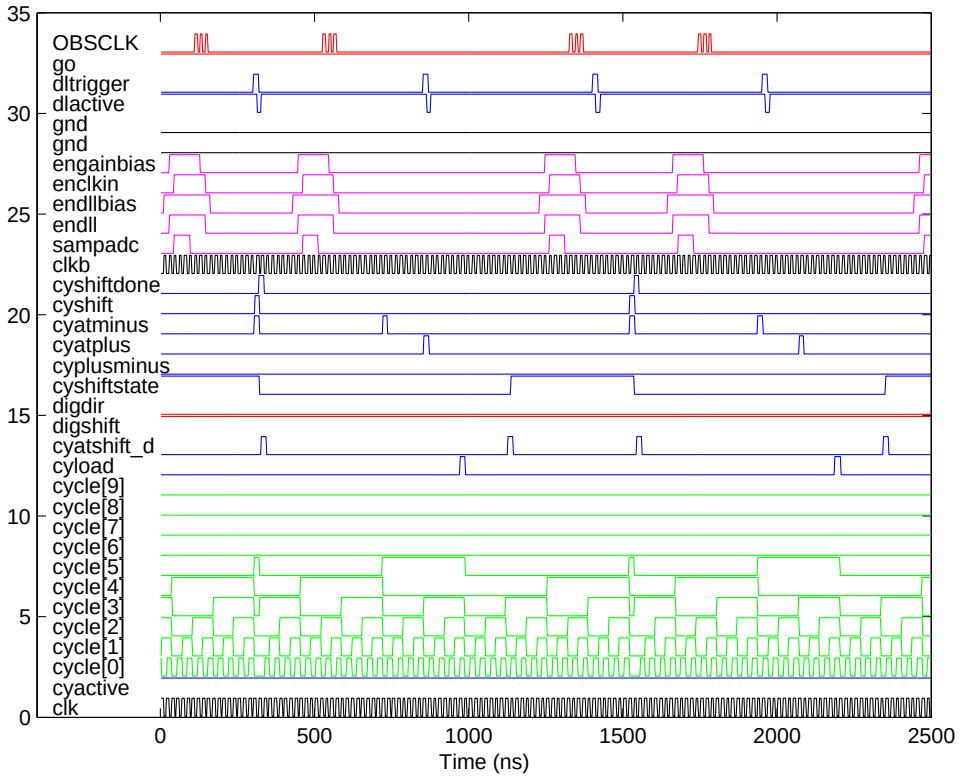

Figure 3.79: Control Cycle Operation, 'Minus' Shifting

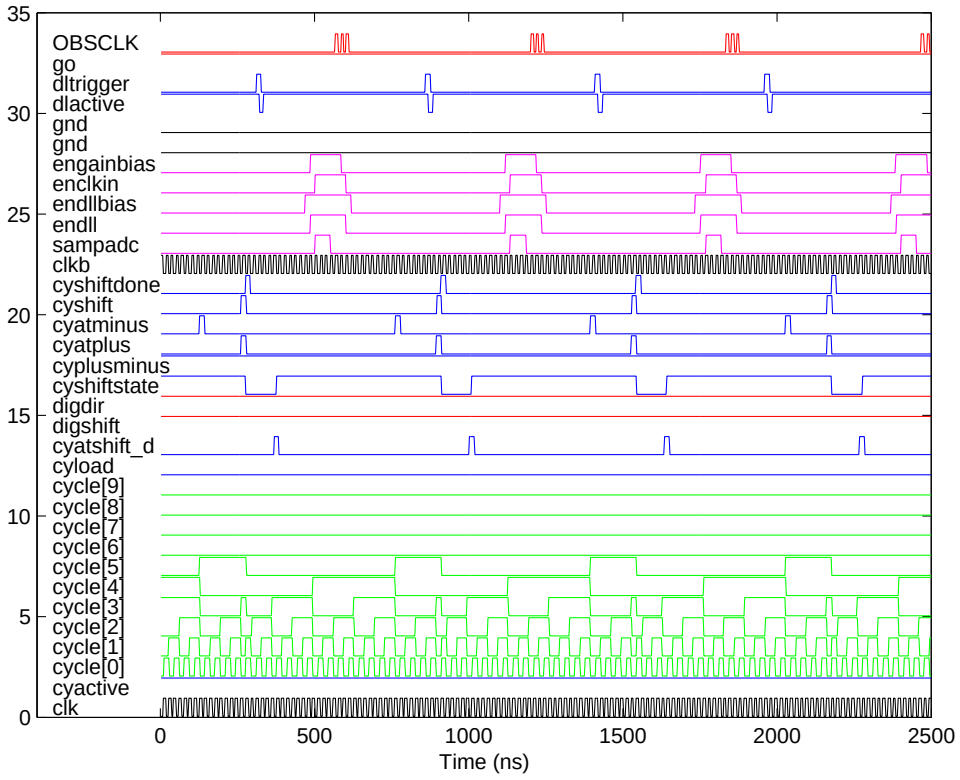

Figure 3.80: Control Cycle Operation, 'Plus' Shifting

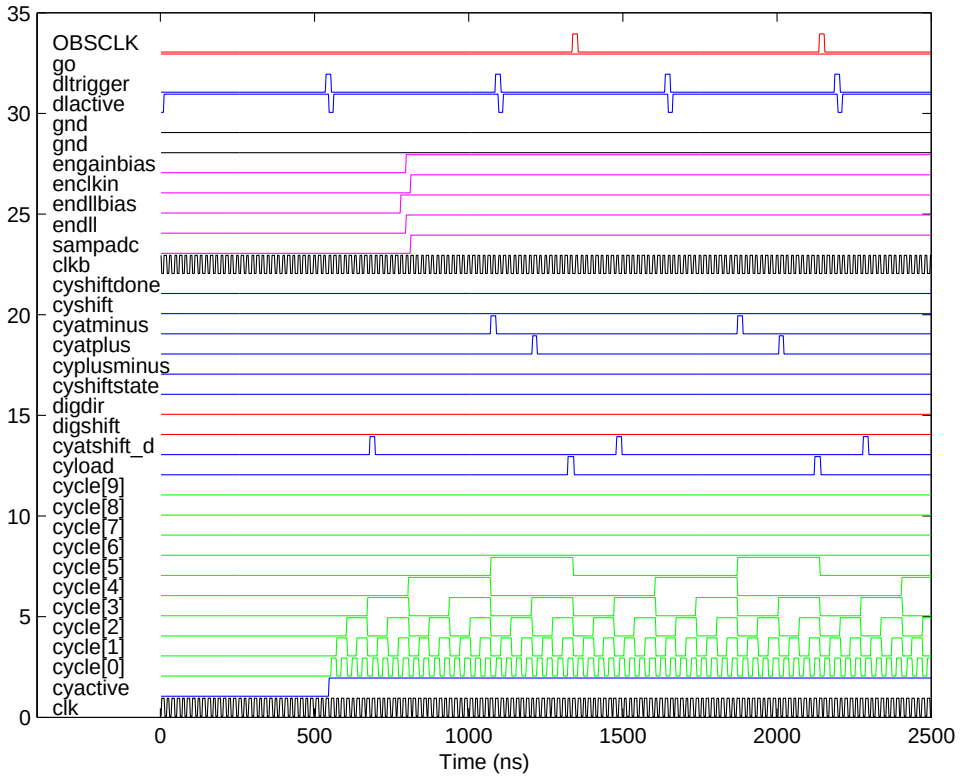

Figure 3.81: Control Cycle Operation, Always ON

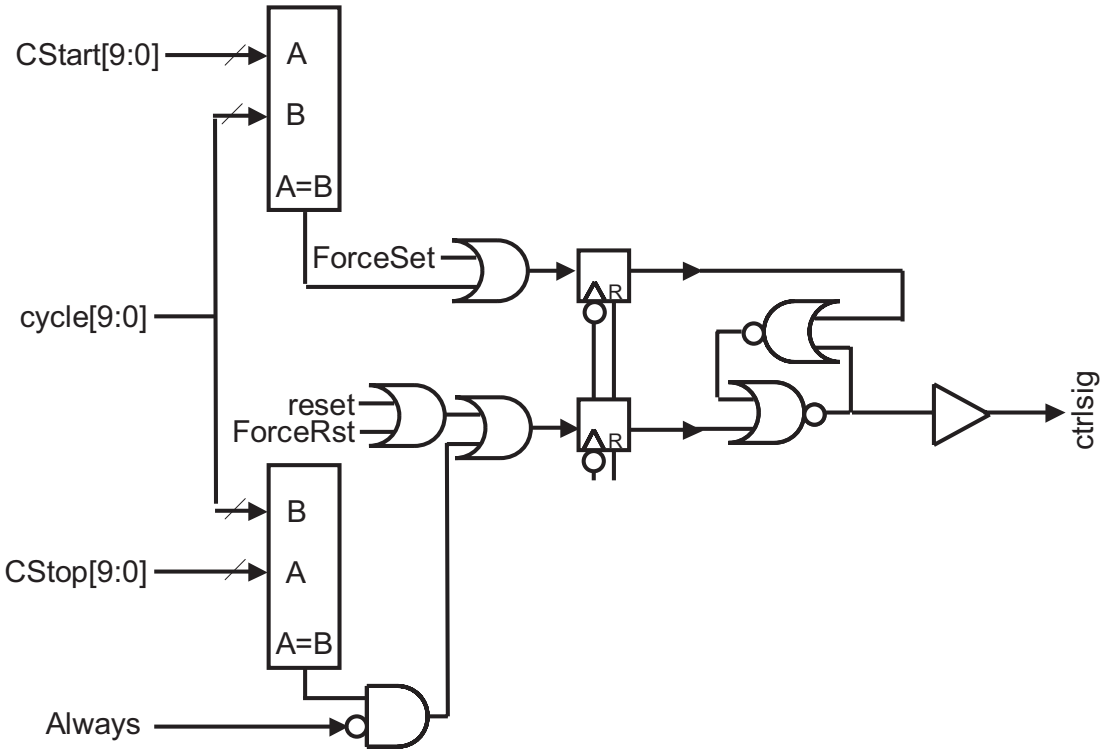

Figure 3.82: Control Signal Generation Logic Diagram

CtrlGenSig

The block diagram for the "CtrlGenSig" module is shown in figure 3.82. The cycle count is compared to two programmable values (to "start" or set the bit high and to "stop" or reset the bit low). The result is clocked on the falling edge of the global clock to make the result available by the next rising edge. A NOR set-reset latch is used to hold state. "Always on" operation is achieved by masking the "stop" signal during normal operation. To keep bit toggling to a minimum, the compare occurs from msb to lsb in a serial fashion.

CtrlADC

The task of the "CtrlADC" unit is to mask incoming DLL phases to generate a nice phased input to the ADC like that shown in figure 3.69. Essentially quarter clocks are used with a shift register to propagate the enable during quiet moments within the clock cycle. As long as the clock duty cycle is within 50% ±25% it will function properly. This gives a wide range of acceptable clock duty-cycles which was easily met. A note should be made about balanced loads. To keep the DLL phase accuracy (maintain delay separation and edge rate), the load seen by the phases used as quarter clocks must be the same as the load seen by all the other phases. To accomplish this, dummy register inputs were used as loads and the layout was carefully designed for matching. Also, this block fits between the DLL delay chain and the ADC's and was designed to have at least 100 micron spacing between them to discourage noise coupling.

CtrlP2P

The "CtrlP2P" block re-times the incoming 32 ADC output bits and aggregates that parallel word into a bank of up to five 32-bit words (for a maximum of 160-bits, or an $80ns$ pulse window) which is applied to the digital backend. Quarter clocks are used to retime the data; with the quarter clocks generated from the CtrlADC block. The "CtrlP2P" control logic uses a 1-hot encoding to select which register to write next to save clock power. This block dominates the power consumption of the control logic, hence care was taken to reduce the amount of logic clocked on each cycle. A block diagram is shown in figure 3.84. Operation is illustrated in figure 3.85 which shows the "CtrlADC" and "CtrlP2P" internal

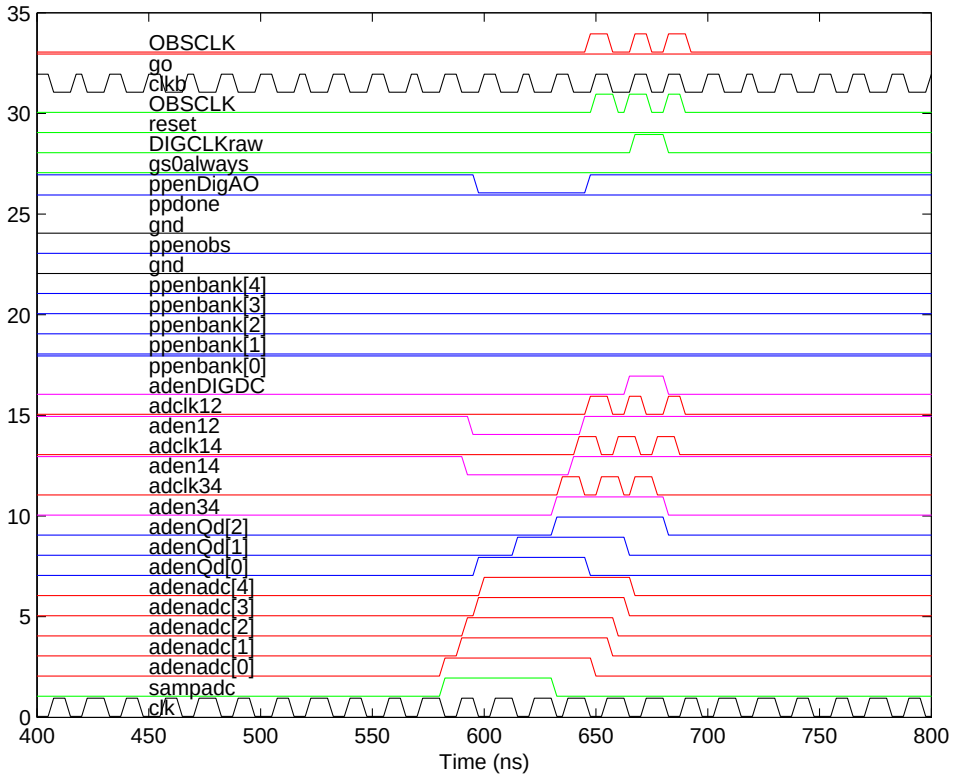

Figure 3.83: Control ADC Phase Generation

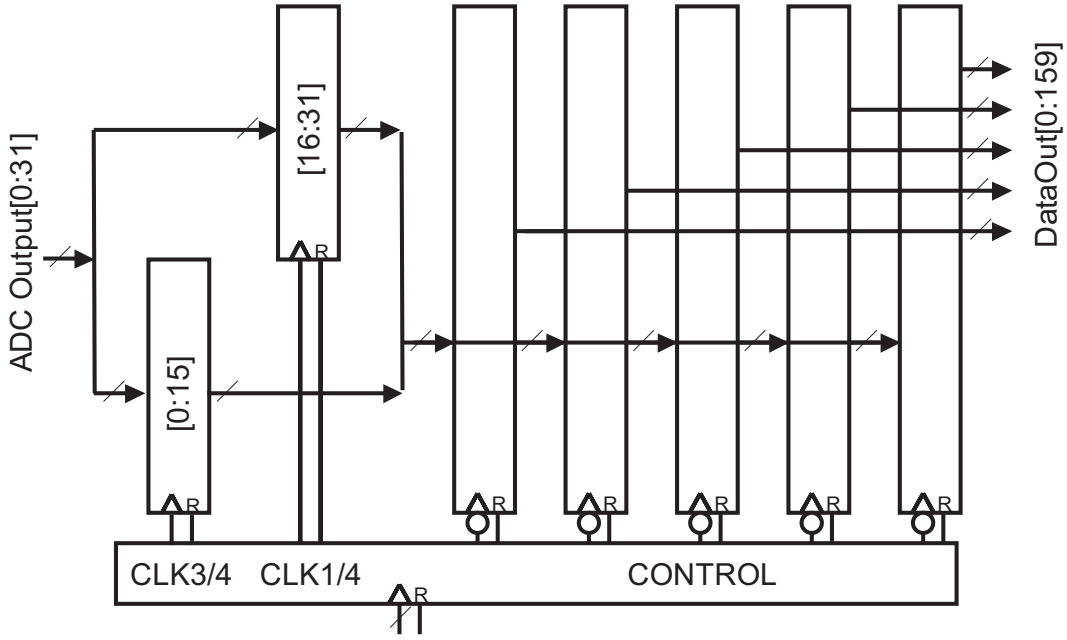

Figure 3.84: Control ADC Output Parallel Aggregation Logic Diagram

logic signals for a 3 word aggregation (3 by 32-bits) under duty-cycled operation. During "always on" operation, shown in figure 3.86, parallel aggregation is not performed and the data is always stored in the first 32-bit register bank and directly passed to the digital backend. For debug, an extra "obs" 32-bit register is included which is always clocked (only in debug mode) and may be examined by reading the appropriate register in the control logic. (See Appendix A address 16 for more information).

Floorplanning

Figure 3.87 shows the overall layout and floorplanning of the control logic blocks. "CtrlADC" extends out, directly interfacing to the DLL phase delay chain outputs and the ADC control phase inputs. A 32-bit bus on either side allows the data to be observed

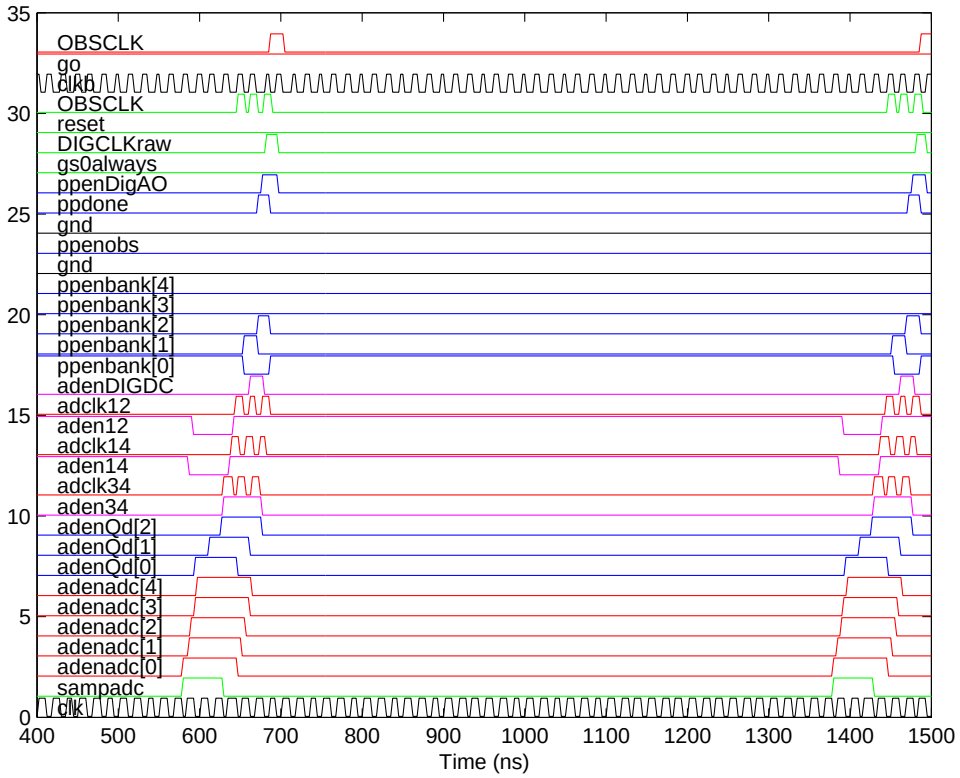

Figure 3.85: Control ADC Output Parallel Aggregation Logic Operation Measurement

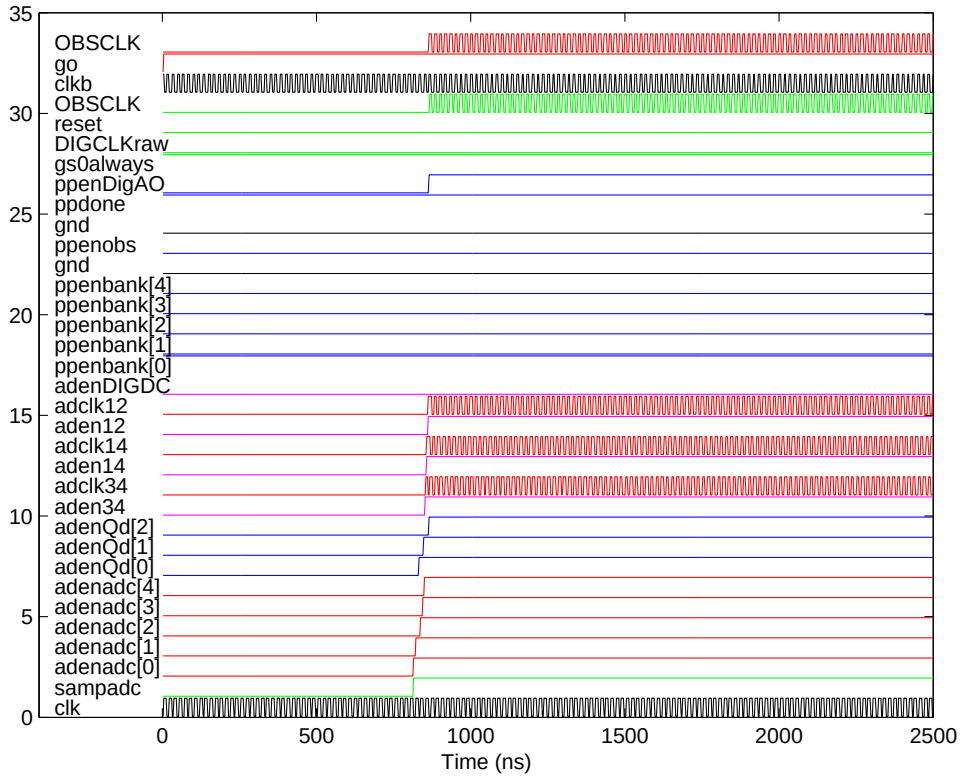

Figure 3.86: Control ADC Output Parallel Aggregation Operation, Always ON

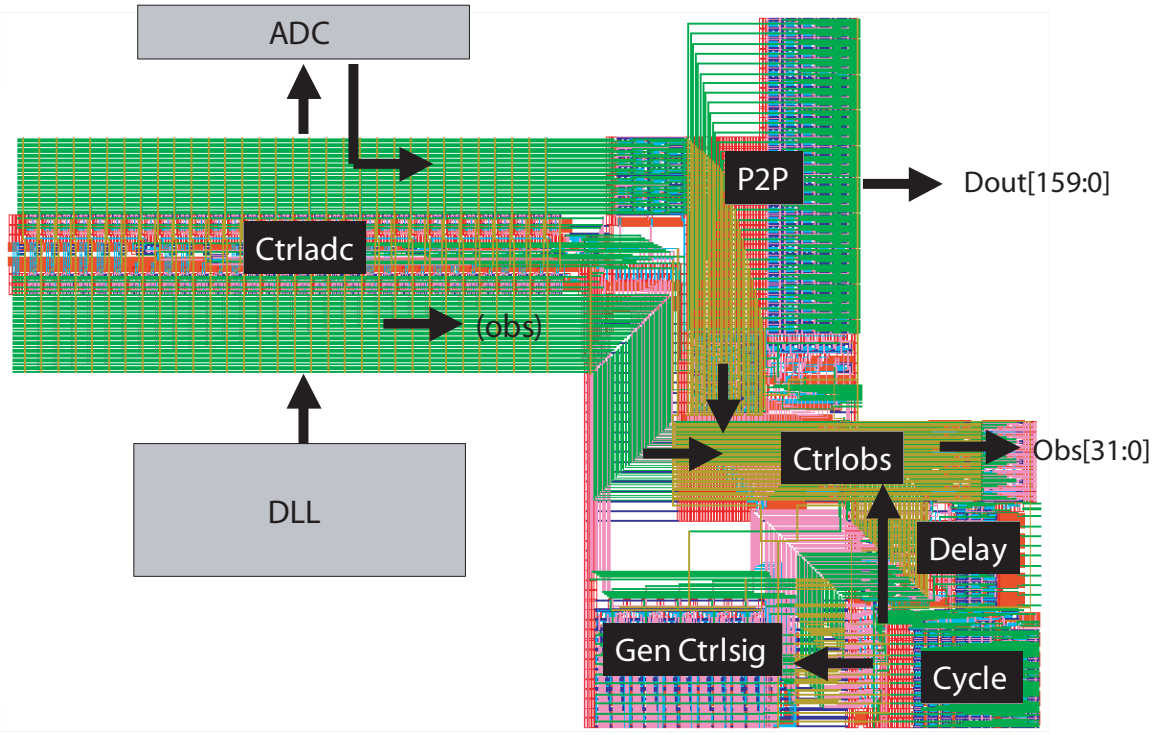

Figure 3.87: Control Logic Layout and Floorplanning

("obs") for debug/testing purposes. (These buses do not switch unless selected for viewing.) Pasted around the logic blocks are AND and MUX gates which allow 32-bits of control state to be observed at a time. Scan chains were not implemented to save design time, as these cells were hand-designed and laid out. For such a relatively small and straightforward design, skipping scan did not incur a large overhead or risk. Care was taken to ensure that the paths of "CtrlObs" were balanced as some observed data (i.e. the DLL phases) have more important timing information than state information. Additionally an extra mode will output the global clock to all 32-bits to allow for delay calibration from the DLL phases to the output pins and onto the testboard.

3.5.2 Custom Standard Cell Logic

The foundry standard cell library was not used with ASIC synthesis tools, even though that is generally faster to design, because of two problems: 1) balanced paths were desired for certain blocks (i.e. "CtrlADC" and "CtrlObs"), and 2) the minimum gate width for a standard cell block was often 2× or 3× larger than the actual minimum feature size available. In the first case, it can often be easier and faster to hand-layout critical, balanced paths rather than fight the place-and-route tool, if the size and complexity is not very large. In the second case, the power consumption of the entire control block may be halved or more by simply creating a small subset of standard library cells and tiling them up. The typical usage scenario for a foundry standard cell library involves fast driving capabilities with possibly large fanout load and tight timing constraints. Our design had limited logic depth (perhaps 10 or 20 gates), small loads, and an abundance of time. For example, even a half clock cycle is $8ns$ long. Taking a pessimistic $100ps$ propagation delay this yields an allowable logic depth of 80 gates.

The standard cells created were typical 2-input AND, NAND, OR, NOR, XOR, XNOR gates, 3-input half adder (ADDH) cells and positive/negative edge resettable flipflops as well as inverters (INV) and buffers (BUF) of different drive strengths. All of the gates are typical static complementary CMOS logic, except for the low power pass-transistor master-slave flip flop, shown in figure 3.88, taken from [118]. The control blocks were implemented in a serial datapath style, as in [119]. A bitslice is taken and replicated to implement counters, comparators, register banks, etc. Digital design and characterization were performed similar to [120] where Perl scripts and ELDO (a SPICE variant) were used to

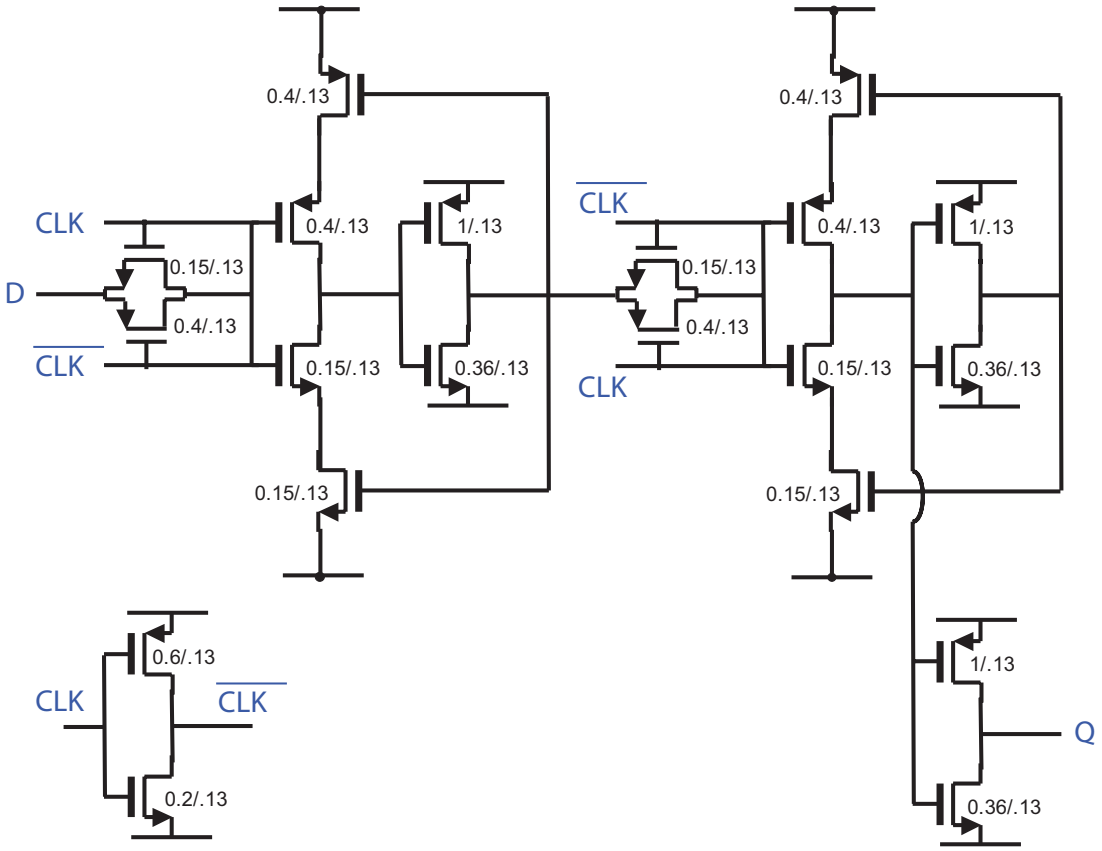

Figure 3.88: Low Power Pass Transistor Flipflop Circuit

automatically characterize the standard cell edge rate and delays over process, temperature and load size. A screen capture of the hand-designed standard logic cells is shown in figure 3.89. The figure is for illustrative purposes only. Note that under normal circumstances the logic cells would alternate orientation to share n-wells, and not all be positioned in the same orientation as in the figure. As can be seen, only a small number of cells were needed.

Figure 3.89: Standard Cell Layout

3.5.3 Chip Interface

A separate digital interface was designed to handle the register programming on the chip because the judgment was made to exclude the digital backend from this tapeout. (This I/O functionality was present in the digital backend.) The decision to wait on the digital backend tapeout was made to minimize risk, aid debugging, and speed up the project schedule. As mentioned in section 2.5, the digital backend signal processing was not optimal, but rather a quality benchmark. In the course of the project design, better ideas were suggested, but could not be pursued due to a lack of personnel. To allow for future experimentation and further reduce design risk, the digital backend was moved into a Xilinx FPGA where its functionality could be easily emulated. The remaining logic needed supports chip I/O involves address decoding and simple register reading/writing. The specific circuits are not of much interest themselves. Figure 3.90 shows the logic floorplan on a screen capture of the chip interface I/O circuits.

3.6 Pulse Generation

Pulse generation is achieved through the use of a simple H-bridge circuit as in [45]. The design for this pulse transmitter is detailed in [2]. The transmitter consists of an NMOS pull-up and PMOS pull-down device per side as shown in figure 3.91. The pulse shape and edge rate are controlled in a limited manner by varying the timing and edge rate of the gate control voltages using a current-starved inverter delay cell similar to that in section 3.3. Current is steered through the output in either a positive (i.e. “1”) or negative (i.e. “0”) direction to create binary amplitude modulation. As mentioned previously, the

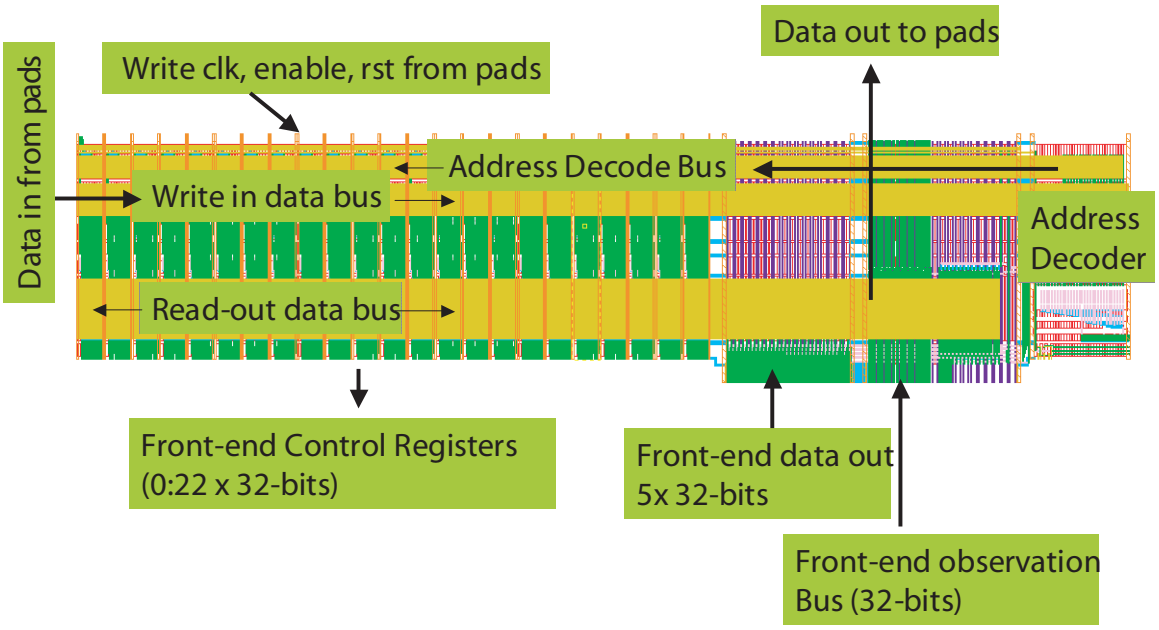

Figure 3.90: Interface (I/O) Logic Layout

transmit pulse is generated from a global counter signal and pulse position modulation may also be generated by dynamically varying the total count at the granularity of the clock pulse ($16.67ns$). Measured pulses over a variety of bias conditions are shown in figure 3.92. The transmitter may generate either a simple pulse with energy at dc, as is shown for "mode 0", or a Manchester or split phase pulse [74] with little energy at dc as shown for "mode 1." Transmit pulse width may be varied from $1ns$ to $2ns$ and the edge rate may be varied from $150ps$ to $750ps$ for both modes.

3.7 Berkeley Impulse Transceiver (BIT)

The front-end transceiver chip was named the 'Berkeley Impulse Transceiver' or BIT for short. The global floorplan is shown superimposed on a die photo in figure 3.93.

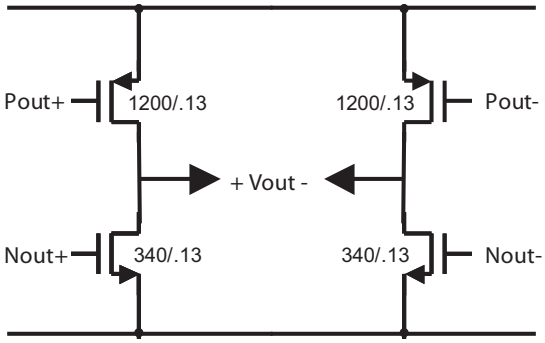

Figure 3.91: Transmit H-Bridge Circuit

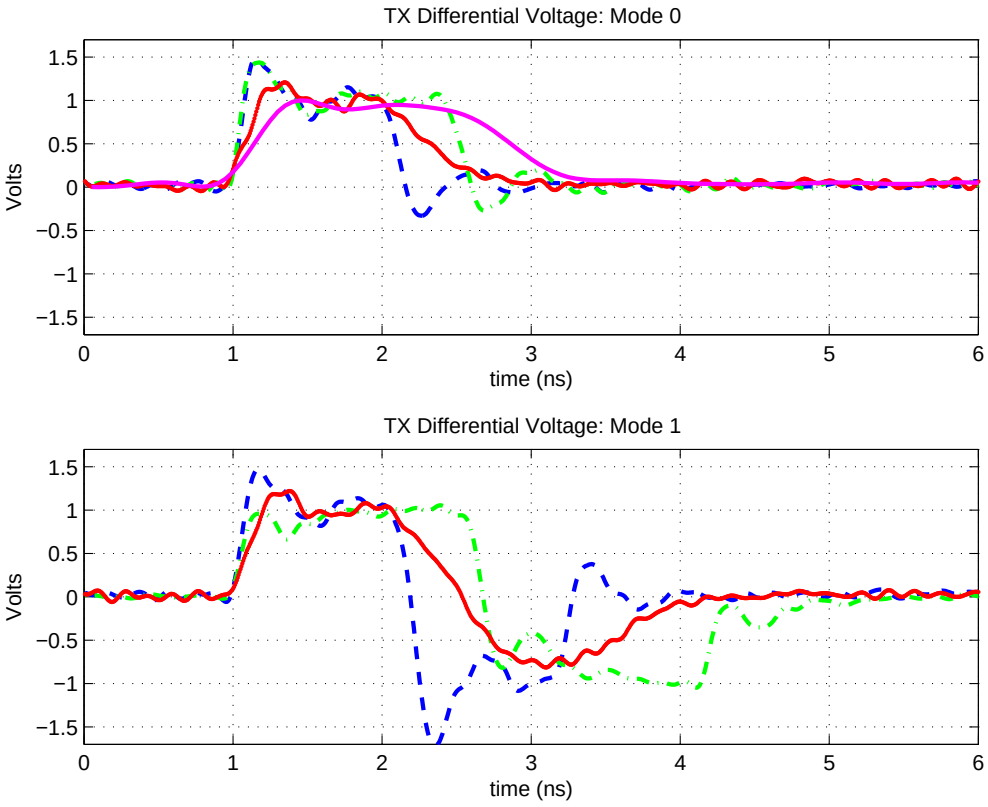

Figure 3.92: Measured Transmit Pulses

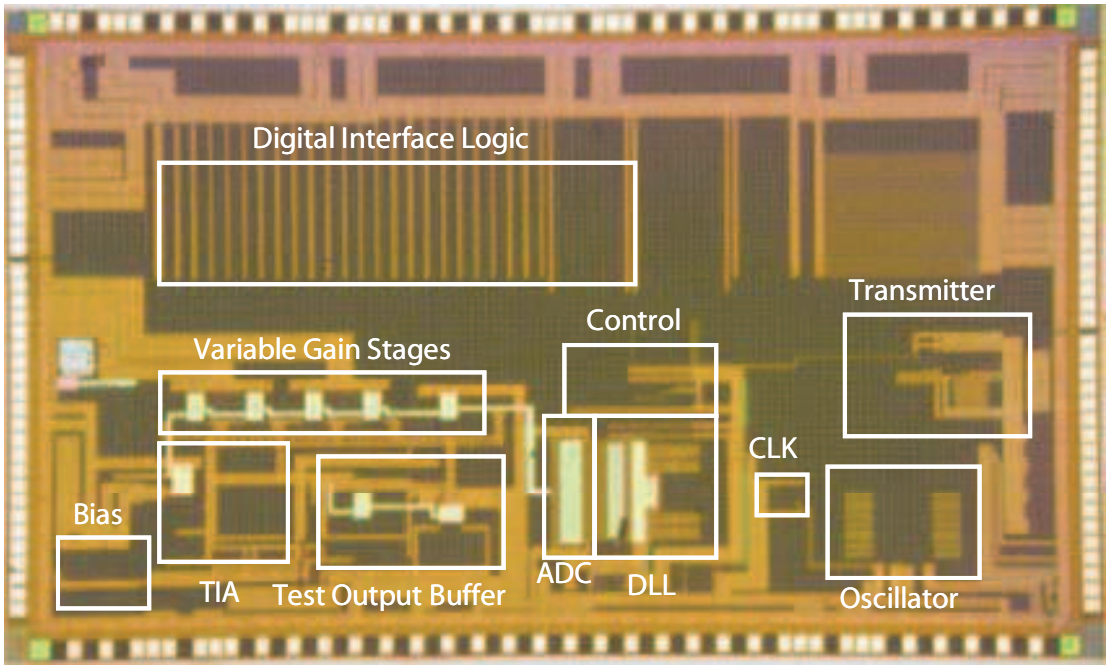

Figure 3.93: Die Photo

Note that the transmitter, oscillator and control logic are kept as far from the bias and TIA input as possible to protect the sensitive input circuits from local large switching transients. The die is padring dominated, with 128 I/O's, mainly for debugging purposes. The digital interface is 57 pins wide: 32 bidirectional I/O, 6 address bits and 6 digital inputs, with 14 VDD and GND pins. The transmitter interface occupies 18 pins: 5 bias pins, 4 TX differential output pins (doubled up to reduce bondwire inductance), and 9 VDD and GND pins. The analog front-end has 53 pins: 3 digital inputs, 2 digital outputs, 4 differential inputs for the TIA (doubled up), 4 differential outputs for the 50Ω buffer (again doubled up), 2 differential OBS analog I/O pins, 5 bias pins, 2 oscillator pins, and 31 VDD and GND pins. Appendix C further enumerates the pinout. The size of each block is listed in table 3.1.

Table 3.1: Chip Characteristics

Process Technology	$0.13\mu m$ standard CMOS 6-metal, 1-poly
Total Die Size	2.8mm x 4.7mm ($13.2mm^2$)
Clock Frequency	$60MHz$ ($1.92Gsample/s$)
Nominal Supply Voltage	$1.1V$
RX Area: Oscillator	0.33 mm^2
RX Area: Clock Buffer	0.02 mm^2
RX Area: Delay Locked Loop	0.16 mm^2
RX Area: A/D Slicer	0.10 mm^2
RX Area: Control Logic	0.24 mm^2
RX Area: Transimped. Amp.	0.33 mm^2
RX Area: Var. Gain Stages	0.20 mm^2
RX Area: ADC Buffer	0.12 mm^2
RX Area: Bias	0.06 mm^2
RX Area Total	2.04 mm^2
TX Area Total	0.48 mm^2

As mentioned in section 3.5.3, the digital backend was not taped out with this die. The first version of the chip layout, shown in figure 3.94, was completed and waiting for the digital design when the decision was made to minimize risk and speed up the project schedule. Future work may include a more thorough analysis of backend signal processing designs and the opportunity to marry it with the analog front-end on the same die.

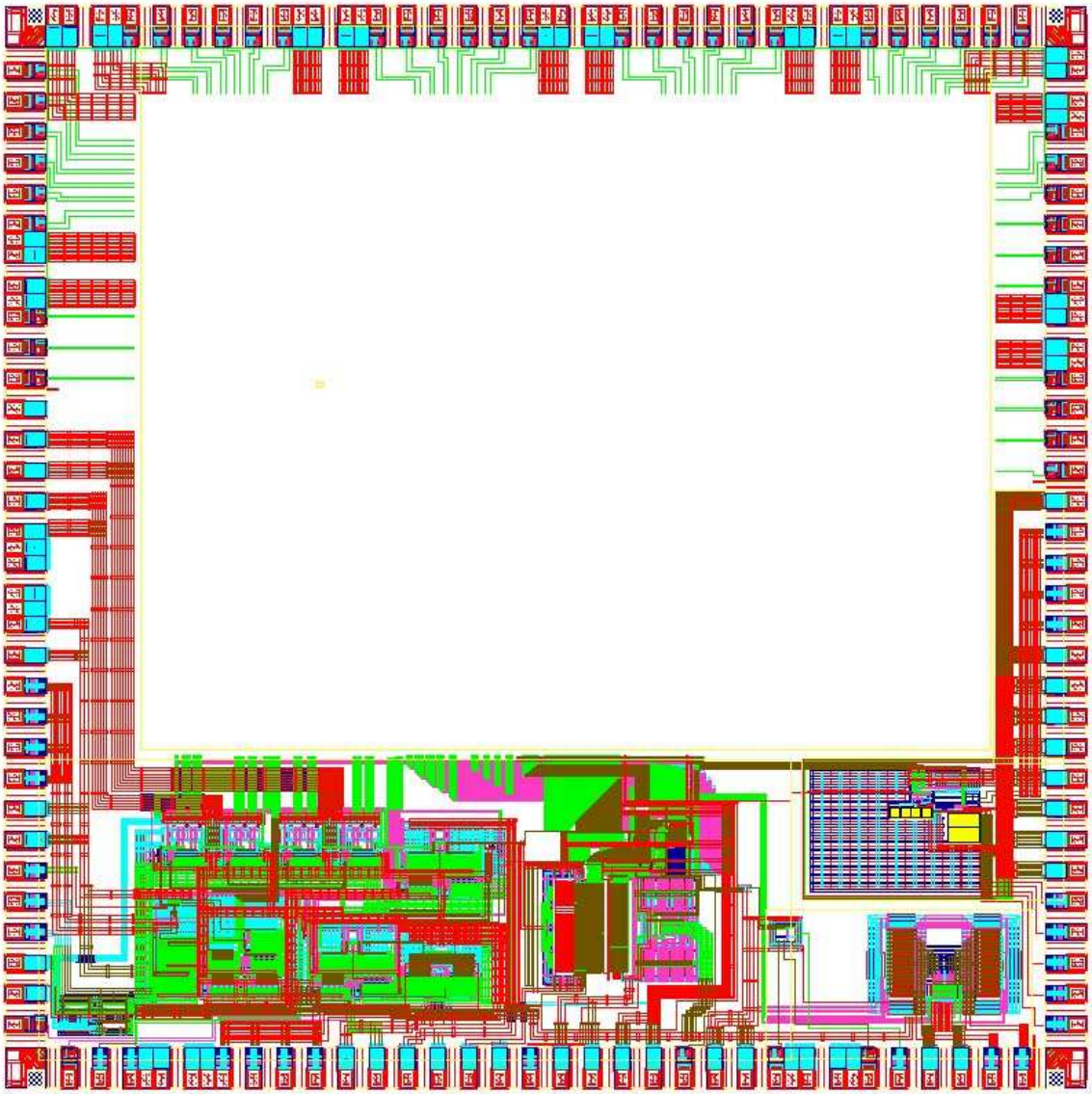

Figure 3.94: First Version of Chip Layout with Space for Digital Backend

Chapter 4

Conclusion

Having discussed the system specification, design and individual block measurements, we move on to the final step: demonstration of a low power, highly integrated impulse-UWB communication transceiver. In the following sections, transceiver operation is presented along with a detailed power consumption breakdown and discussion of duty-cycled operation. Lastly, the research contributions are summarized and avenues for future work are presented.

4.1 Transceiver Operation

To demonstrate the transceiver operation, a digital backend (see appendix B for the test setup) was programmed to transmit a length-11 Barker code [62] XORed with a programmable input bit at a $10Mpulse/s$. The Barker code used is shown in table 4.1. When a sequence longer than 11 was needed (e.g. to present a clearer correlation profile) the same repeating length-11 Barker code is overlaid as the input stream to create a

Table 4.1: Transmitter Barker Code

Input Bit	Transmitted Bit Sequence										
0	+1	+1	+1	-1	-1	-1	+1	-1	-1	+1	-1
1	-1	-1	-1	+1	+1	+1	-1	+1	+1	-1	+1

repeating 11×11 concatenated Barker code. The Barker code was chosen arbitrarily for its autocorrelation profile with uniform, minimum sidelobe energy (to ease demonstration of correlation). There is nothing inherent to this code; any code may be chosen. The receiver was programmed to capture a window of time at the same $10MHz$ rate (or in "always on" mode – capturing data continuously) and data was measured using a high speed oscilloscope, logic analyzer or backend Xilinx, and then post-processed as noted. TEM horns were used for the transmit and receive antennas as they have good performance from $100MHz$ to $1GHz$.

4.1.1 Loopback Test

The first test was a loopback test, directly connecting the transmitter outputs with SMA cables to the receiver input. Figure 4.1 shows the loopback results: measured differentially at the transmitter and measured at the 50Ω output buffer (programmed to observe the ADC input). The correlation profile was post-processed in Matlab. Note that the transmit pulse rings for nearly half of the pulse repetition cycle. The transmitter H-bridge circuit creates very low impedance when the pulse is generated, then turns off, creating high impedance. The receiver input impedance is approximately 100Ω differential,

and this large impedance mismatch causes substantial ringing. The effect of the ringing may be seen in the ADC output and correlation profile as it effectively widens the received pulse.

4.1.2 Transmission and Reception

The second test was to send pulses over the air to the receive antenna. Figure 4.2 shows the transmit pulses measured on the testboard and receive pulses measured at the TEM horn antenna. Measurements were taken at a 2 foot distance so that the pulses would be easily discernible relative to the background noise. By 3 feet (or approximately $1m$ separation, the receive pulse amplitude decreased enough to be similar to the noise amplitude. The data was captured with a $20Gsample/s$ digital sampling oscilloscope and post-processed (filtered in Matlab) for presentation to remove the presence of large, out-of-band interference from a WLAN 802.11a/b transmitter that was less than $3m$ away in the lab. As the receiver gain stage bandwidth is less than $1GHz$, this interference was not expected to have any impact on its operation.

For this setup, the receive ADC output bits were captured and post-processed to generate a correlation profile for various antenna separation distances as is shown in figure 4.3, figure 4.4, and figure 4.5. As expected, a healthy correlation peak is clearly visible (121 is the expected peak for a 11×11 Barker sequence). In addition, we can see some of the channel impulse response (ringing) after the main peak. Note that for these profiles, we are only correlating per bin. A template filter would assume a pulse shape across time (i.e. the sample bins) and thereby capture more energy than a single bin. This illustrates the benefit of a digital approach to pulse correlation as opposed to a simpler analog pulse correlation

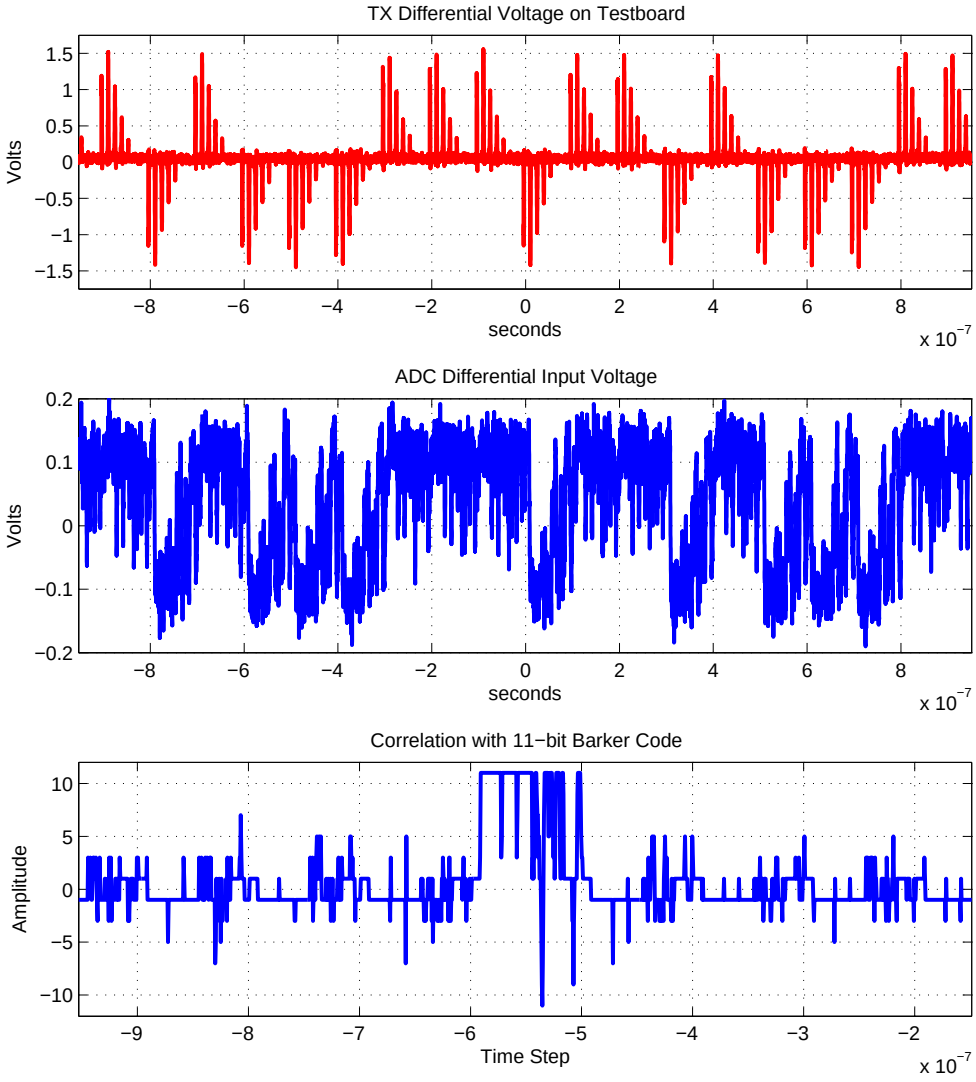

Figure 4.1: Loopback Test

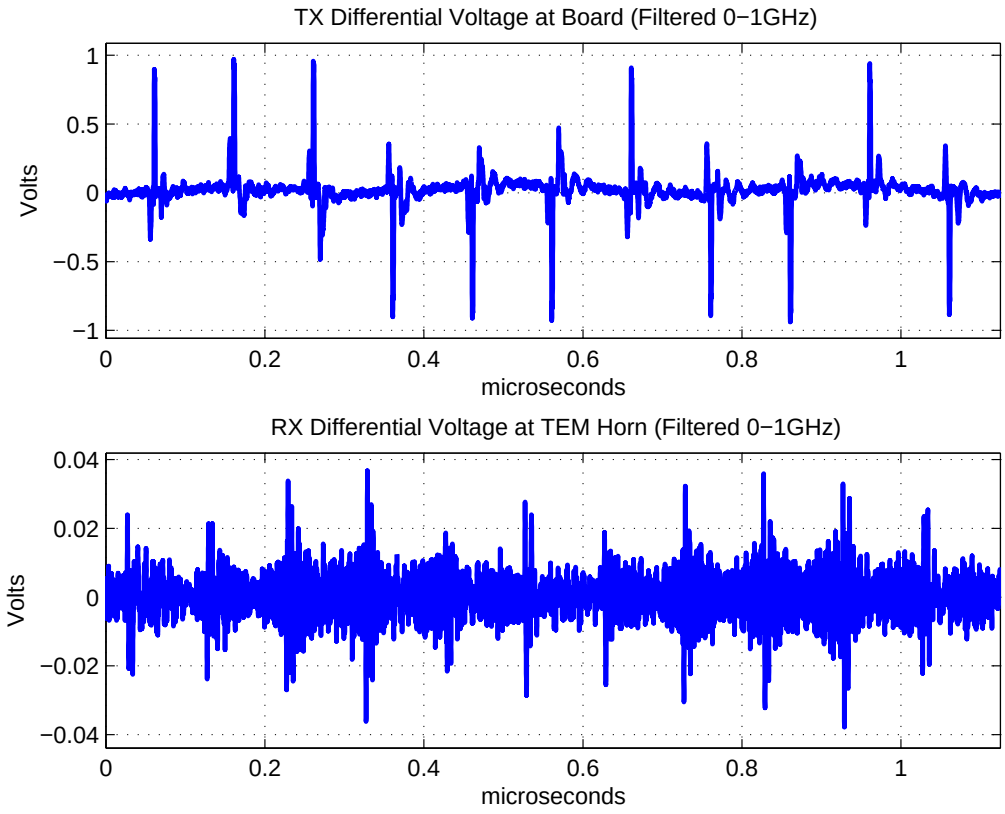

Figure 4.2: Pulse Transmission and Reception

circuit that may only capture one peak (around a couple nanoseconds or so of time) of the received impulse response. Also note that a shift in the peak is observed as the antenna is moved. This shift may be used as a simple method of ranging because it corresponds to a time of flight difference in transmission. (More sophisticated ranging and locationing algorithms may be employed, but that is a dissertation on its own.)

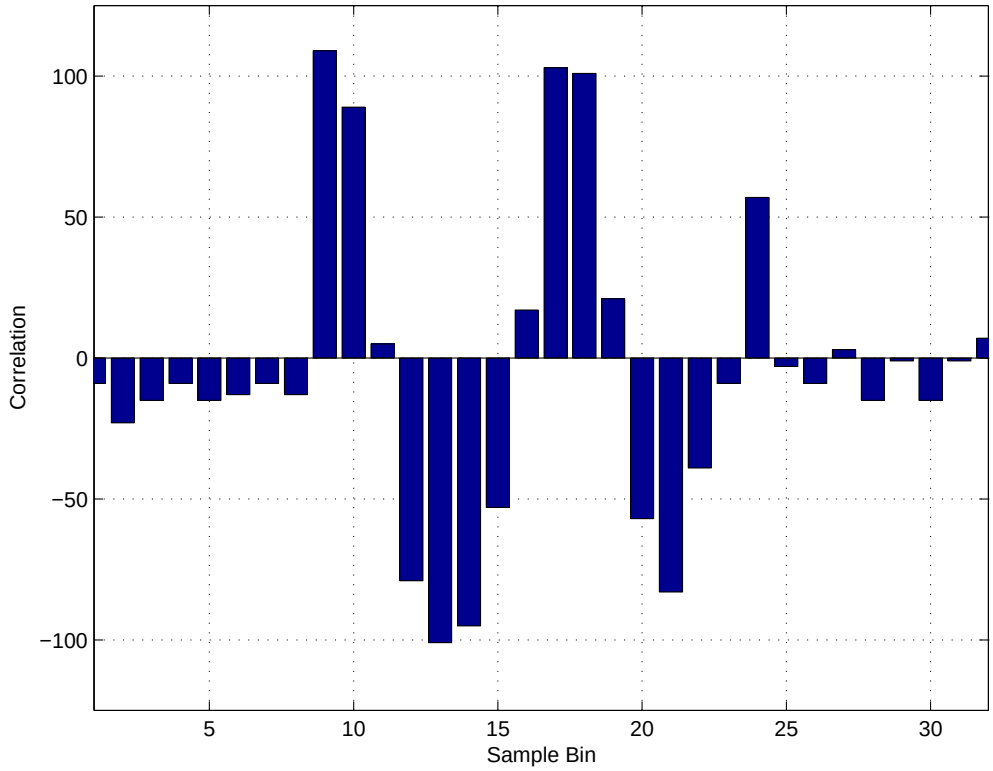

Figure 4.3: Correlation at 1 ft.

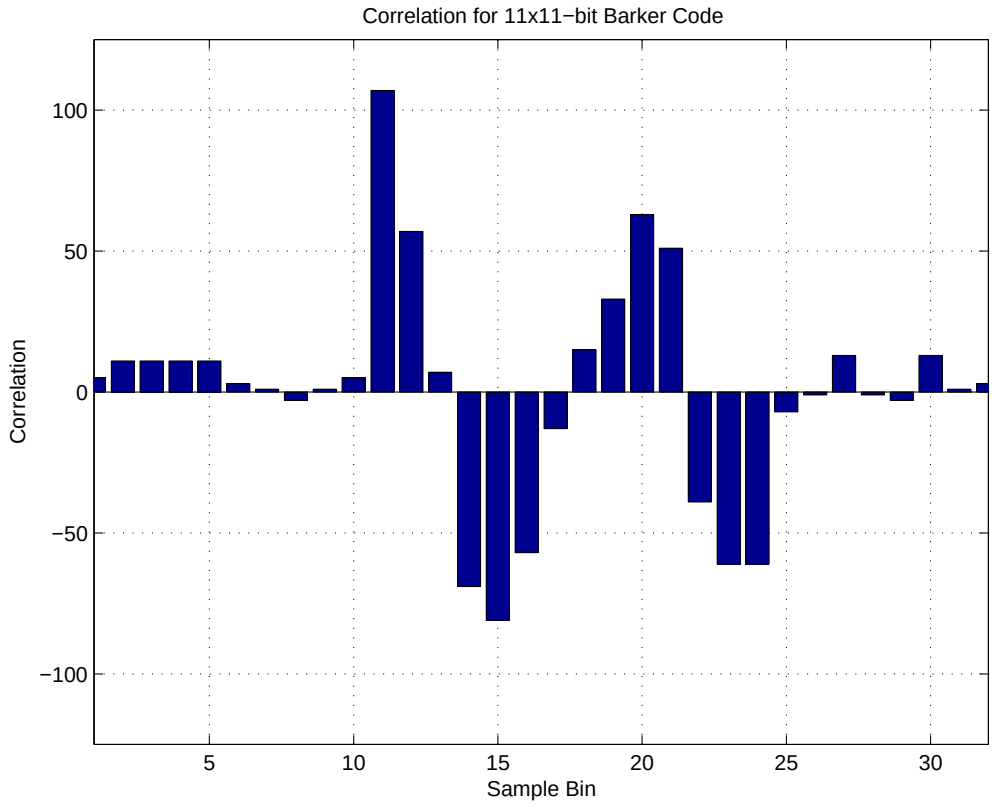

Figure 4.4: Correlation at 2 ft.

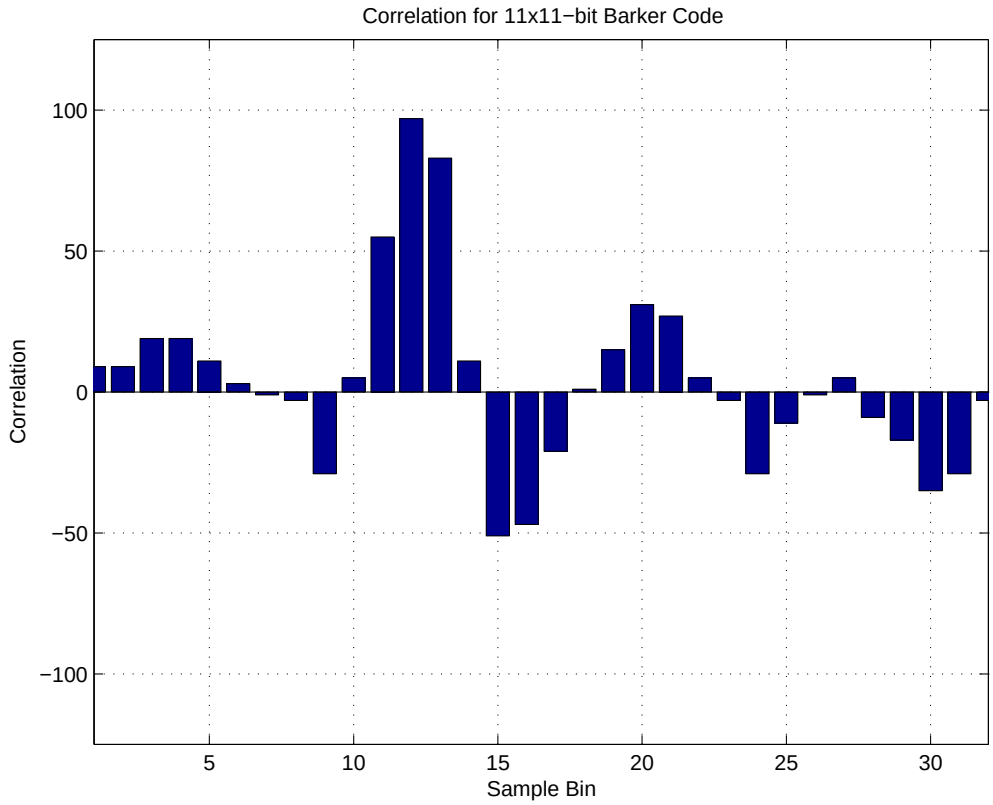

Figure 4.5: Correlation at 3 ft.

4.1.3 Power Consumption

The receiver power consumption was measured for various transmit pulse rates (allowing for various amounts of duty-cycling) and is shown in figure 4.6. The dominant power consumer in the receiver is the gain block, consuming half ($1.7mW$) the total power ($3.4mW$) in continuous operation. In particular, the TIA input stage, which must provide a differential 100Ω input impedance, dominates the power consumption in the gain block at nearly $1mW$. The 1-bit, $1.92Gsample/s$ sampling operation consumes nearly a milliwatt itself in constant operation. (The sampling operation consists of the ADC, DLL, and ADC buffer gain stage.) The rest of the power is distributed mainly to the digital control logic (global counter) which consumes approximately a half milliwatt (but could probably be designed to consume less power). Bias consumes $170\mu W$ but also could be designed to take less. The oscillator is negligible ($70\mu W$) in comparison (unless the chip operation is heavily duty-cycled).

The effect of duty-cycling on power consumption is shown in figure 4.7 relative to non-duty-cycled operation. Using $1Mpulse/s$, the duty-cycled power consumption is less than 1/4 of the non-duty-cycled power at that same pulse rate. What limits this improvement is the granularity of duty-cycling used. For this prototype, the global $60MHz$ clock dictates the duty-cycling granularity. Since we need at last $6ns$ to allow the gain stages to settle upon re-activation, to sample a full clock width ($16.7ns$) of time requires two clock cycles total ($33.3ns$), even though only $22.7ns$ is needed. If we emulate many simple impulse-UWB systems which do not capture much energy from the channel delay spread, perhaps only a couple nanoseconds of sampling time, we can reduce this duty-cycling

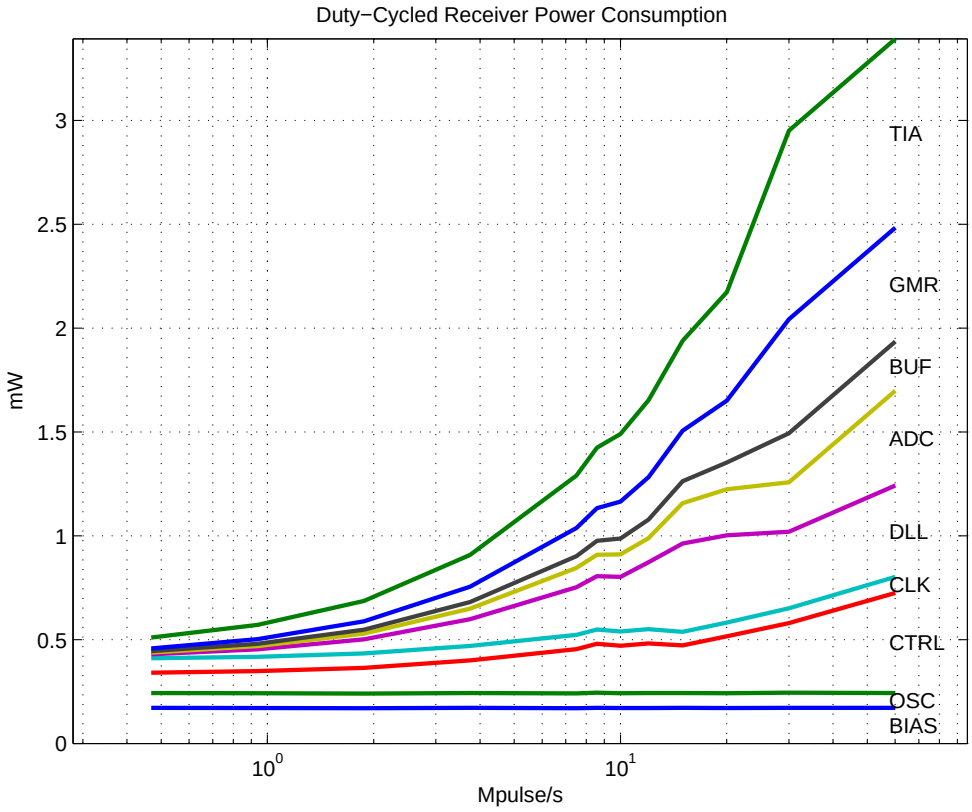

Figure 4.6: Receiver Power Consumption Breakdown vs. Duty-Cycling

by another factor of 2 or 3 (resulting in a factor of 1/8 to 1/12 less power when duty-cycled at $1Mpulse/s$). To properly compare the simpler UWB systems with this duty-cycled approach, a more aggressive (finer) granularity of duty-cycling should be used.

In spite of the non-optimal duty-cycling granularity, note that at $4Mpulse/s$ the total receiver power is less than $1mW$ and at $1Mpulse/s$ the total receiver power is $580\mu W$. In fact, at $1Mpulse/s$ the total transmitter + receiver power is less than $1mW$. These numbers are very encouraging as they meet the power target for our application and imply that even lower power consumption is possible by optimizing the duty-cycling.

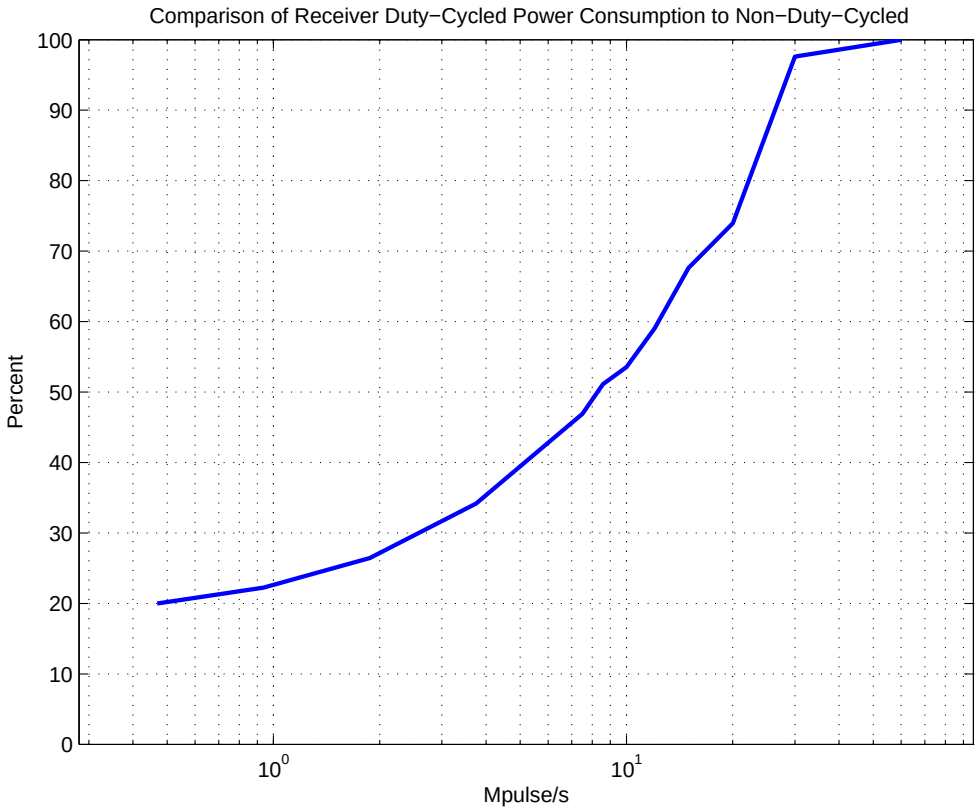

Figure 4.7: Power Consumption Duty-Cycled vs. Non-Duty-Cycled Operation

One interesting observation that might be made from figure 4.6 is that a simple division of pulse rate by power indicates that the most "efficient" area of operation is at the maximum pulse rate. This might lead one to conclude that a "burst" UWB transmission will be more efficient that a slower, continuous transmission for the same total number of bits (illustrated in figure 4.8). This is not necessarily accurate though. As the pulse rate scales down, the ability exists to scale up the amplitude, maintaining constant power spectral density (PSD), as per the FCC UWB regulations. If we do not scale the pulse amplitude, then it may be more efficient to operate at the faster pulse rate (assuming the acquisition cost from the header length may be amortized over the packet length for burst operation). However, if we scale the transmit amplitude up as we reduce the pulse rate and examine the efficiency of impulse-UWB communication in terms of throughput divided by power (as in figure 1.1 in the introduction), a different picture emerges.

To present a more realistic scenario, a spreading code is assumed to be overlaid on the pulses. This spreading code length is changed relative to the pulse rate (and hence scaled amplitude) to maintain a minimum receive SNR for a fair comparison. The result is shown in figure 4.9 and it predicts a roughly constant efficiency of $1.5 Mbps/mW$ down to $1 Mpulse/s$ (After $1 Mpulse/s$, the spreading code has length = 1 and we lose throughput efficiency.) Note that this scaling is conservative: assuming random data applied over a long period of time, the peak PSD magnitude scales as $\sqrt{30 Mpulse/s/f_{pulse}}$. However the FCC actually measures the peak power in a limited bandwidth of $50 MHz$ around the frequency under test, which may allow the power to be scaled directly with the pulse rate, i.e. as $(30 Mpulse/s/f_{pulse})$. In this case the efficiency increases with larger pulse amplitude to

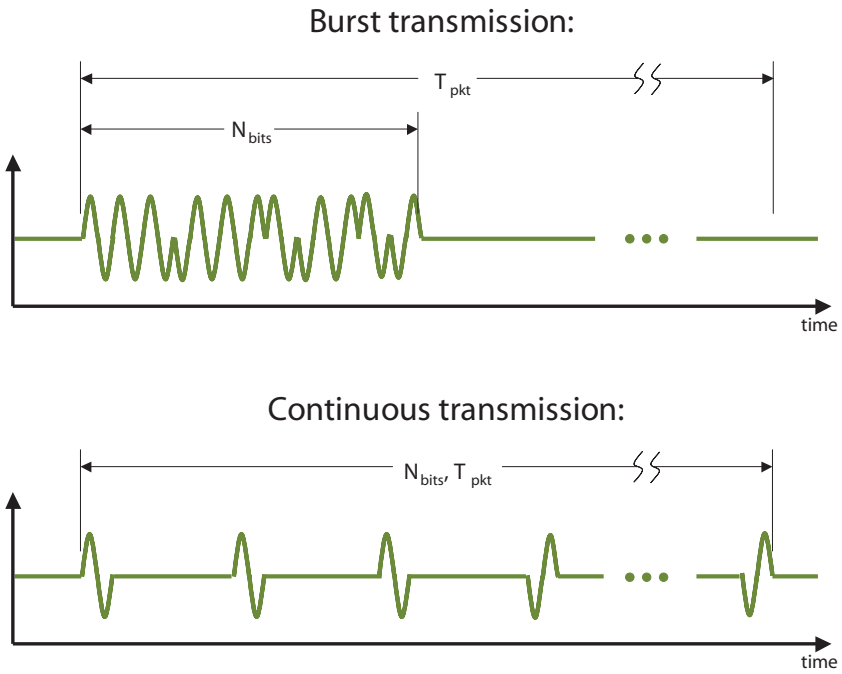

Figure 4.8: Burst vs. Continuous Signaling

the point where the maximum FCC peak to average ratio of $20dB$ is attained and no more scaling is allowed.

The transmitter power consumption is also shown in figure 4.10. There is no breakdown here; the dominant power consumer is the H-bridge pre-drivers and the current that is consumed by the H-bridge stage itself. A relatively small amount of static bias current on the order of $300\mu W$ is present, but this bias could be designed to consume less power.

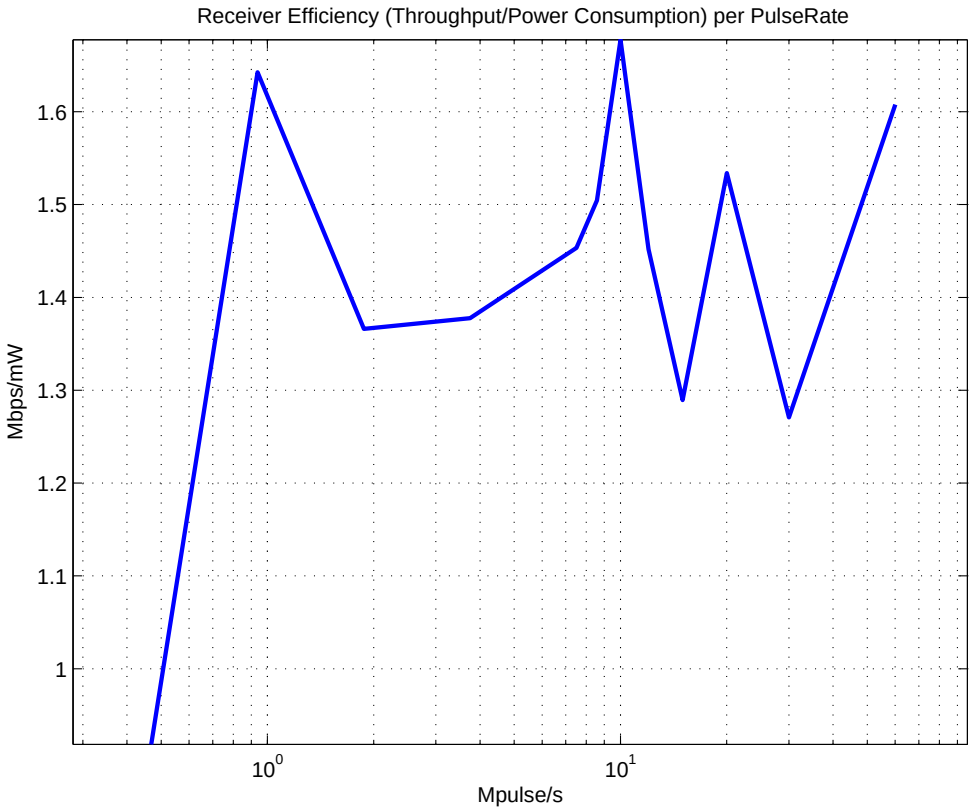

Figure 4.9: Receiver Throughput/Power vs. Pulse Rate for Constant PSD

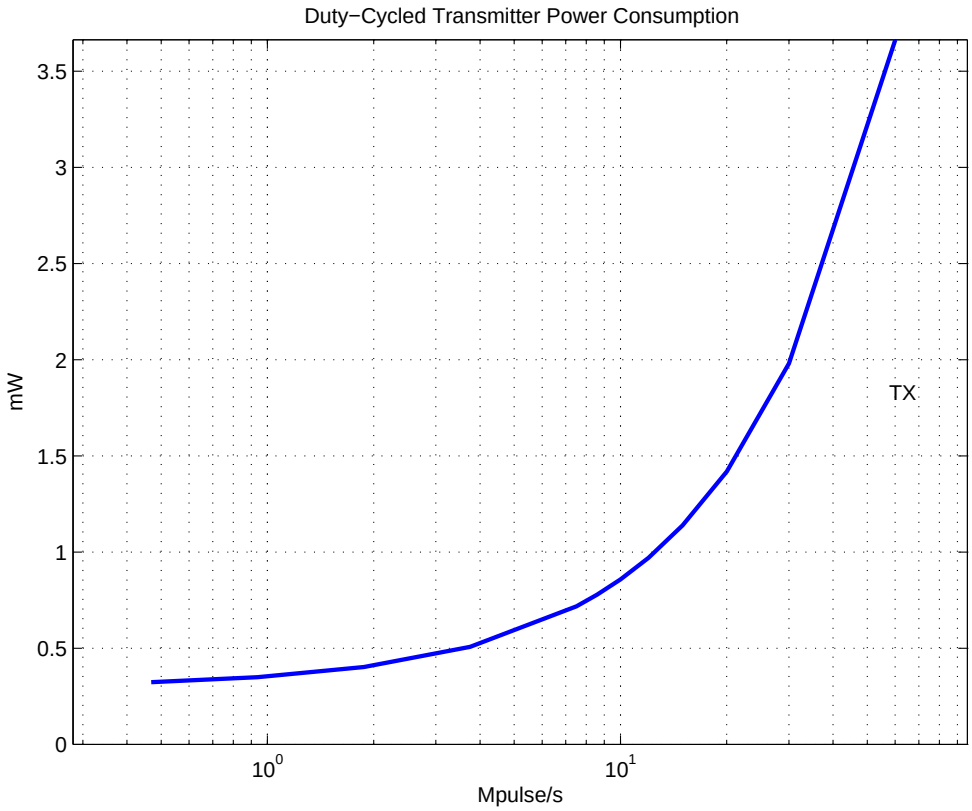

Figure 4.10: Transmitter Power Consumption vs. Duty-Cycling

4.2 Closing Remarks

The examination of impulse-UWB signaling was inspired by the need for an ultra low power, low cost radio able to operate over short distances with the possibility to performing ranging. After exploring the UWB channel characteristics and design space, a "mostly digital" architecture was selected and an extensible modeling and simulation environment was created to evaluate system specification trade-offs. System-level power conservation techniques were identified (i.e. duty-cycling between pulse reception) as well as selection criteria for low power circuit design. Suitable low power circuit topologies were determined and implemented in a standard, digital $0.13\mu m$ CMOS process. A flexible, highly integrated impulse-UWB transceiver front-end prototype was designed and power efficient communication on the order of $1.5Mbps/mW$ was demonstrated. Total power consumption was measured at less than $1mW$ for $1Mpulse/s$ for transmit and receive including digital backend power estimates. These results surpass traditional narrowband designs by over an order of magnitude. Additionally, the programmable nature of the prototype chip allows for future experimentation.

Bibliography

[1] M. S. W. Chen, "Ultra-wide-band baseband design and implementation," Master's thesis, University of California at Berkeley, Berkeley, CA, 2002.

[2] S. B. T. Wang, "Design of ultra-wideband rf front-end," Ph.D. dissertation, University of California at Berkeley, Berkeley, CA, 2005.

[3] J. R. Jauchem, M. R. Frei, K. L. Ryan, J. H. Merritt, and M. R. Murphy, "Lack of effects on heart rate and blood pressure in ketamine-anesthetized rats briefly exposed to ultra-wideband electromagnetic pulses," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 1, pp. 117–20, Jan. 1999.

[4] M. J. Ammer, "Low power synchronization for wireless communication," Ph.D. dissertation, University of California at Berkeley, Berkeley, CA, 2004.

[5] B. W. Cook, A. D. Berny, A. Molnar, S. Lanzisera, and K. S. Pister, "An ultra-low power 2.4ghz rf transceiver for wireless sensor networks in 0.13um cmos with 400mv supply and an integrated passive rx front-end," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'06)*, vol. 49, Feb. 2006, pp. 370–1.

[6] Z. Xu, S. Jiang, T. Wu, H.-Y. Jian, G. Chu, K. Ku, P. Wang, N. Tran, Q. Gu, M.-Z. Lai, C. Chien, M. F. Chang, and P. D. Chow, "A compact dual-band direct-conversion cmos transceiver for 802.11 a/b/g wlan," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'05)*, vol. 48, Feb. 2005, pp. 98–9.

[7] T. Maeda, H. Yano, T. Yamase, N. Yoshida, N. Matsuno, S. Hori, K. Numata, R. Walkington, T. Tokairin, Y. Takahashi, M. Fujii, and H. Hida, "A direct-conversion cmos transceiver for 4.9-5.95ghz multi-standard wlans," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'04)*, vol. 47, Feb. 2004, pp. 90–1.

[8] R. Ahola, A. Aktas, J. Wilson, K. R. Rao, F. Johsson, I. Hyyrylainen, a. Brolin, T. Hakala, A. Friman, T. Makiniemi, J. Hanze, M. Sanden, D. Wallner, Y. Guo, T. Lagerstam, L. Noguer, T. Knuuttila, P. Olofsson, and M. Ismail, "A single chip cmos transceiver for 802.11 a/b/g wlans," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'04)*, vol. 47, Feb. 2004, pp. 92–3.

[9] L. Perraud, C. Pinatel, M. Recouly, J.-L. Bonnot, N. Sornin, F. Benoist, M. Massei, and O. Gibrat, "A dual-band 802.11a/b/g radio in 0.18 um cmos," in *Proc. IEEE*

International Solid-State Circuits Conference (ISSCC'04), vol. 47, Feb. 2004, pp. 94–5.

[10] A. Behzad, L. Lin, Z. M. Shi, S. Anand, K. Carter, M. Kappes, E. Lin, T. Nguyen, D. Yuan, S. Wu, Y. C. Wong, V. Fong, and A. Fofougaran, "Direct-conversion cmos transceiver with automatic frequency control for 802.11a wireless lans," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 356–7.

[11] D. Su, M. Zargari, P. Yue, S. Rabii, D. Weber, B. Kaczynski, S. Mehta, K. Singh, S. Mendis, and B. Wooley, "A 5ghz cmos transceiver for ieee 802.11a wireless lan," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'02)*, vol. 45, Feb. 2002, pp. 92–3.

[12] H. Darabi, S. Khorram, Z. Zhou, T. Li, B. Marholev, J. Chiu, J. Castaneda, E. Chien, S. Anand, S. Wu, M. Pan, R. Roufoograran, H. Kim, P. Lettieri, B. Ibrahim, J. Rael, L. Tran, E. Geronaga, H. Yeh, T. Frost, J. Trachewsky, and A. Rofougaran, "A fully integrated soc for 802.11b in 0.18um cmos," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'05)*, vol. 48, Feb. 2005, pp. 96–7.

[13] W. Kluge, L. Dathe, R. Jaehne, S. Ehrenreich, and D. Eggert, "A 2.4ghz cmos transceiver for 802.11b wireless lans," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 360–1.

[14] I. Bhatti, R. Roufoogaran, and J. Castaneda, "A fully integrated transformer-based front-end architecture for wireless transceivers," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'05)*, vol. 48, Feb. 2005, pp. 106–7.

[15] K. Muhammad, D. Leipold, B. Staszewski, Y.-C. Ho, C. M. Hung, K. Maggio, C. Fernando, T. Jung, J. Wallberg, J.-S. Koh, S. John, I. Deng, O. Moreira, R. Staszewski, R. Katz, and O. Friedman, "A discrete-time bluetooth receiver in a 0.13um digital cmos process," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'04)*, vol. 47, Feb. 2004, pp. 268–9.

[16] C. Cojocaru, T. Pamir, F. Balteanu, A. Namdar, D. Payer, I. Gheorghe, T. Lipan, K. Sheikh, J. Pingot, H. Paananen, M. Littow, M. Cloutier, and E. MacRobbie, "A 43mw bluetooth transceiver with -91dbm sensitivity," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 90–1.

[17] C.-H. Park, S. Byun, Y. Song, S. Wang, C. Conroy, and B. Kim, "A low power cmos bluetooth transceiver with a digital offset canceling dll-based gfsk demodulator," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 96–7.

[18] N. Filiol, N. Birkett, J. Cherry, F. Balteanu, C. Cojocaru, A. Namdar, T. Pamir, K. Sheikh, G. Glandon, D. Payer, A. Swaminathan, R. Forbes, T. Riley, S. M. Alinoor, E. Macrobbie, M. Cloutier, S. Pipilos, and T. Varelas, "A 22mw bluetooth rf

transceiverr with direct rf modulation and on-chip if filtering," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'01)*, vol. 44, Feb. 2001, pp. 202–3.

[19] A. J. Leeuwenburgh, J. W. F. ter Laak, A. G. Mulders, A. J. Hoogstraate, P. J. M. van Laarhoven, M. Mijrolder, J. G. Pummel, and P. J. M. Kamp, "A 1.9ghz fully integrated cmos dect transceiver," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 450–1.

[20] "Tr1000 product information," http://www.rfm.com/, RF Monolithics, 2004.

[21] "Tr3000 product information," http://www.rfm.com/, RF Monolithics, 2004.

[22] P. Quinlan, P. Crowley, M. Chanca, S. Hudson, B. Hunt, K. Mulvaney, G. Retz, C. O'Sullivan, and P. Walsh, "A multi-mode 0.3-128kb/s transceiver for the 433/868/915mhz ism bands in 0.25um cmos," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'04)*, vol. 47, Feb. 2004, pp. 274–5.

[23] "Hrf-roc093xc product information," http://www.mysoiservices.com/, Honeywell, 2003.

[24] "9xcite product information," http://www.maxstream.net, MaxStream, 2003.

[25] "X2010 product information," http://www.okwelectronics.com, OKW Electronics, 2002.

[26] "Ia4320 product information," http://www.integration.com/, Integration Associates, 2003.

[27] P. Choi, H. Park, I. Nam, K. Kang, Y. Ku, S. Shin, S. Park, T. Kim, H. Choi, S. Kim, S. M. Park, M. Kim, S. Park, and K. Lee, "An experimental coin-sized radio for extremely low power wpan (ieee802.15.4) application at 2.4ghz," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 396–7.

[28] "At86rf210 product information," http://www.atmel.com/, Atmel, 2004.

[29] "Cc2420 product information," http://www.chipcon.com/, Chipcon, 2003.

[30] "Em250 product information," http://www.ember.com/, Ember, 2004.

[31] J.-Y. Chen, M. P. Flynn, and J. P. Hayes, "A fully-integrated auto-calibrated super-regenerative receiver," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'06)*, vol. 49, Feb. 2006, pp. 376–7.

[32] B. P. Otis, Y. H. Chee, and J. M. Rabaey, "A 400 uw-rx, 1.6mw-tx super-regenerative transceiver for wireless sensor networks," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'05)*, vol. 48, Feb. 2005, pp. 396–7.

[33] B. P. Otis, Y. H. Chee, R. Lu, N. M. Pletcher, and J. M. Rabaey, "An ultra-low power mems-based two-channel transceiver for wireless sensor networks," in *IEEE VLSI Circuits Digest of Technical Papers (VLSI'04)*, June 2004, pp. 20–3.

[34] G. Luff, D. Stegmeir, M. Mostafa, C. Quek, W. Roberts, D. Haab, M. Romney, B. Stutz, D. Walker, K. Tran, S. Tunser, S. Thilenius, N. Troop, D. Eddowes, E. Lee, K. Laba, D. Schwan, M. Huber, P. Brown, S. Moghe, and R. Koupal, "A low cost, highly integrated 5.8ghz low-if transceiver for 1.5 mbps streaming data applications," in *Proc. IEEE Radio Frequency Integrated Circuits Conference (RFIC'04)*, June 2004, pp. 343–6.

[35] A.-S. Porret, T. Melly, C. C. Enz, and E. A. Vittoz, "A low-power low-voltage transceiver architecture suitable for wireless distributed sensors network," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'00)*, May 2000, pp. 156–9.

[36] A.-S. Porret, T. Melly, D. Python, C. C. Enz, and E. A. Vittoz, "An ultralow power uhf transceiver integrated in a standard digital cmos process: Architecture and receiver," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, pp. 452–66, Mar. 2001.

[37] A. Vouilloz, M. Declercq, and C. Dehollain, "A low-power cmos super-regenerative receiver at 1 ghz," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, pp. 440–51, Mar. 2001.

[38] N. Joehl, C. Dehollain, P. Favre, P. Deval, and M. Declercq, "A low-power 1-ghz super-regenerative transceiver with time-shared pll control," *IEEE J. Solid-State Circuits*, vol. 36, no. 7, pp. 1025–31, July 2001.

[39] P. Favre, N. Joehl, A. Vouilloz, P. Deval, C. Dehollain, and M. J. Declercq, "A 2-v 600μa 1-ghz bicmos super-regenerative receiver for ism applications," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 2186–96, Dec. 1998.

[40] D. G. Y. Yee, "A design methodology for highly-integrated low-power receivers for wireless communication," Ph.D. dissertation, University of California at Berkeley, Berkeley, CA, 2001.

[41] *First Report and Order*, Federal Communications Commission Std. FCC 02-48, Feb. 2002.

[42] J. D. Taylor, Ed., *Introduction to Ultra-Wideband RADAR Systems*, 1st ed. Boca Raton, Florida: CRC Press, 1995.

[43] R. A. Scholtz, "Multiple access with time-hopping impulse modulation," in *Proc. Military Communications Conference (MILCOM'93)*, vol. 2, Oct. 1993, pp. 447–50.

[44] P. I. Withington and L. W. Fullterton, "An impulse radio communications system," in *Proc. of the International Conference on Ultra-Wideband, Short-Pulse Electromagnetics*, Oct. 1992, pp. 113–20.

[45] R. A. Fleming and C. E. Kushner, "Spread sprectrum localizers," U.S. Patent 5,748,891, May, 1998.

[46] G. M. Maggio, N. Rulkov, and L. Reggiani, "Pseudo-chaotic time hopping for uwb impulse radio," *IEEE Trans. Circuits Syst. I*, vol. 48, no. 12, pp. 1–12, Dec. 2001.

[47] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted reference rf communications," in *Proc. IEEE Conference on Ultra-Wideband Systems and Technologies (UWBST'02)*, May 2002, pp. 265–9.

[48] T. Terada, S. Yoshizumi, Y. Sanada, and T. Kuroda, "A cmos impulse radio ultra-wideband transceiver for 1mb/s data communication and +/-2.5cm range findings," in *IEEE VLSI Circuits Digest of Technical Papers (VLSI'05)*, June 2005, pp. 30–3.

[49] A. Tamtrakarn, H. Ishikuro, K. Ishida, M. Takamiya, and T. Sakurai, "A 1-v 299μw flashing uwb transceiver based on double thresholding scheme," in *IEEE VLSI Circuits Digest of Technical Papers (VLSI'06)*, June 2006.

[50] C. J. L. Martret and G. B. Giannakis, "All-digital pam impulse radio for multiple-access through frequency-selective multipath," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM'00)*, vol. 1, Nov. 2000, pp. 77–81.

[51] W. Namgoong, "A channelized dsss ultra-wideband receiver," in *Proc. IEEE Radio and Wireless Conference (RAWCON'01)*, Aug. 2001, pp. 105–8.

[52] H. J. Lee, D. S. Ha, and H. S. Lee, "A frequency-domain approach for all-digital cmos ultra wideband receivers," in *Proc. IEEE Conference on Ultra Wideband Systems and Technologies (UWBST'03)*, Nov. 2003, pp. 86–90.

[53] S. Hoyos, B. M. Sadler, and G. R. Arce, "Analog to digital conversion of ultra-wideband signals in orthogonal spaces," in *Proc. IEEE Conference on Ultra Wideband Systems and Technologies (UWBST'03)*, Nov. 2003, pp. 47–51.

[54] I. D. O'Donnell and R. W. Brodersen, "A highly-integrated, low-power, ultra-wideband transceiver for low-rate, indoor wireless systems," Qualifying Exam, Dept. of Electrical Engineering, University of California at Berkeley, Nov. 2000.

[55] R. Blazquez, P. P. Newaskar, F. S. Lee, and A. P. Chandrakasan, "A baseband processor for pulsed ultra-wideband signals," in *Proc. IEEE Custom Integrated Circuits Conference (CICC'04)*, Oct. 2004, pp. 587–90.

[56] I. D. O'Donnell, M. S. W. Chen, S. B. T. Wang, and R. W. Brodersen, "An integrated, low power, ultra-wideband transceiver architecture for low-rate, indoor wireless systems," in *Proc. IEEE CAS Workshop on Wireless Communications and Networking*, Pasadena, CA, Sept. 2002.

[57] F. S. Lee, D. D. Wentzloff, and A. P. Chandrakasan, "An ultra-wideband baseband front-end," in *Proc. IEEE Radio Frequency Integrated Circuits Conference (RFIC'04)*, June 2004, pp. 493–6.

[58] *Code of Federal Regulations Title 47*, Federal Communications Commission Std. Part 15, Dec. 2001.

[59] I. D. O'Donnell and R. W. Brodersen, "An ultra-wideband transceiver architecture for low power, low rate, wireless systems," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1623–31, Sept. 2005.

[60] F. J. Agee, C. E. Baum, W. D. Prather, J. M. Lehr, J. P. O'Loughlin, J. W. Burger, J. S. H. Schoenberg, D. W. Scholfield, R. J. Torres, J. P. Hull, and J. A. Gaudet, "Ultra-wideband transmitter research," *IEEE Trans. Plasma Sci.*, vol. 26, no. 3, pp. 860–73, June 1998.

[61] W. D. Prather, C. E. Baum, J. M. Lehr, J. P. O'Loughlin, S. Tyo, J. S. H. Schoenberg, R. J. Torres, T. C. Tran, D. W. Scholfield, J. Gaudet, and J. W. Burger, "Ultra-wideband source and antenna research," *IEEE Trans. Plasma Sci.*, vol. 28, no. 5, pp. 1624–30, Oct. 2000.

[62] J. D. Taylor, Ed., *Ultra-Wideband RADAR Technology*, 1st ed. Boca Raton, Florida: CRC Press, 2001.

[63] H. L. Bertoni, L. Carin, and L. B. Felsen, Eds., *Ultra-Wideband Short-Pulse Electromagnetics*, 1st ed. New York: Plenum Press, 1993.

[64] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw Hill, 1995.

[65] J. M. Cramer, R. A. Scholtz, and M. Z. Win, "Spatio-temporal diversity in ultra-wideband radio," in *Proc. IEEE Wireless Communications and Networking Conference*, Sept. 1999, pp. 888–92.

[66] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth signal propagation for indoor wireless communications," in *Proc. IEEE International Conference on Communication*, June 1997, pp. 56–60.

[67] ——, "On the energy capture of ultrawide bandwidth signals in dense multipath environments," *IEEE Commun. Lett.*, vol. 2, no. 9, pp. 245–7, Sept. 1998.

[68] D. Cassioli, M. Z. Win, and A. F. Molisch, "The ultra-wide bandwidth indoor channel: From statistical model to simulations," *IEEE J. Select. Areas Commun.*, vol. 20, no. 6, pp. 1247–57, Aug. 2002.

[69] ——, "A statistical model for the uwb indoor channel," in *Proc. IEEE International Conference on Vehicular Technology*, May 2001, pp. 1159–63.

[70] J. Yang, "Spatial channel characterization for cognitive radios," Master's thesis, University of California at Berkeley, Berkeley, CA, 2004.

[71] H. Tang, "A unified approach to wireless system design," Ph.D. dissertation, University of California at Berkeley, Berkeley, CA, 2003.

[72] D. C. Giancoli, *Physics for Scientists and Engineers with Modern Physics*, 2nd ed. New Jersey: Prentice Hall, 1988.

[73] P. M. Clarkson, *Optimal and Adpative Signal Processsing*, 1st ed. Boca Raton, Florida: CRC Press, 1993.

[74] L. W. C. II, *Digital and Analog Communication Systems*, 5th ed. New Jersey: Prentice Hall, 1997.

[75] K. Siwiak and D. McKeown, *Ultra-Wideband Radio Technology*, 1st ed. New York: John Wiley & Sons, 2004.

[76] D. M. Pozar, *Microwave Engineering*, 2nd ed. New York: John Wiley & Sons, 1998.

[77] M. G. D. Benedetto and G. Giancola, *Understanding Ultra Wide Band Radio Fundamentals*, 1st ed. New Jersey: Prentice Hall, 2004.

[78] P. P. Newaskar, R. Blazquez, and A. P. Chandrakasan, "A/d precision requirements for an ultra-wideband radio receiver," in *Proc. IEEE Workshop on Signal Processing Systems (SIPS'02)*, Oct. 2002, pp. 270–5.

[79] S. Hoyos, B. M. Sadler, and G. R. Arce, "Mono-bit digital receivers for ultra-wideband communications," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1337–44, July 2005.

[80] J. M. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan, "Picoradios for wireless sensor networks – the next challenge in ultra-low power design," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'02)*, vol. 45, Feb. 2002, pp. 200–1.

[81] I. S.-C. Lu, N. Weste, and S. Parameswaran, "Adc precision requirements for digital ultra-wideband receivers with sublinear front-ends: A power and performance perspective," in *VLSI Design*, 2006, pp. 575–80.

[82] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Select. Areas Commun.*, vol. 17, no. 4, pp. 539–50, Apr. 1999.

[83] G. V. der Plas, S. Decoutere, and S. Donnay, "A 0.16pj/conversion-step 2.5mw 1.25gs/s 4b adc in 90nm digital cmos process," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'06)*, vol. 49, Feb. 2006, pp. 566–7.

[84] S. W. M. Chen and R. W. Brodersen, "A 6b 600ms/s 5.3mw asynchronous adc in 0.13um cmos," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'06)*, vol. 49, Feb. 2006, pp. 574–5.

[85] L. Y. Nathawad, R. Urata, B. A. Wooley, and D. A. B. Miller, "A 20ghz bandwidth, 4b photoconductive-sampling time-interleaved cmos adc," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'02)*, vol. 45, Feb. 2003, pp. 320–1.

[86] X. Jiang, Z. Wang, and M. F. Chang, "A 2gs/s 6b adc in 0.18μm cmos," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. 46, Feb. 2003, pp. 322–3.

[87] P. Scholtens and M. Vertregt, "A 6b 1.6gsample/s flash adc in 0.18μm cmos using averaging termination," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'02)*, vol. 45, Feb. 2002, pp. 168–9.

[88] C. Donovan and M. P. Flynn, "A "digital" 6-bit adc in 0.25-μm cmos," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 432–7, Mar. 2002.

[89] J. A. Crawford, *Frequency Synthesizer Design Handbook*. Norwood, MA: Artech House, 1994.

[90] Q. Huang and P. Basedau, "A 200 μ a, 78mhz cmos crystal-oscillator digitally trimmable to 0.3ppm," in *Proc. IEEE International Symposium on Low Power Electronics and Design (ISLPED'96)*, Aug. 1996, pp. 305–8.

[91] B. Razavi, "Prospects of cmos technology for high-speed optical communications circuits," *IEEE J. Solid-State Circuits*, vol. 37, no. 9, pp. 1135–1145, Sept. 2002.

[92] I. D. O'Donnell and R. W. Brodersen, "A 2.3mw baseband impulse-uwb transceiver front-end in cmos," in *IEEE VLSI Circuits Digest of Technical Papers (VLSI'06)*, June 2006.

[93] ——, "A flexible, low power, baseband, impulse-uwb transceiver front-end," Proc. IEEE International Conference on Ultra-Wideband (ICUWB'06), Sept. 2006, submitted.

[94] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd ed. New York: John Wiley & Sons, 1993.

[95] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 1st ed. Cambridge, UK: Cambridge University Press, 1998.

[96] B. Razavi, *RF Microelectronics*, 1st ed. New Jersey: Prentice Hall, 1998.

[97] ——, *Design of Analog CMOS Integrated Circuits*, 1st ed. New York: McGraw Hill, 2001.

[98] E. A. Vittoz and J. Fellrath, "Cmos analog integrated circuits based on weak inversion operation," *IEEE J. Solid-State Circuits*, vol. 12, no. 3, pp. 224–31, June 1977.

[99] D. Aebischer, H. J. Oguey, and V. R. von Kaenel, "A 2.1-mhz crystal oscillator time base with a current consumption under 500 na," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 999–1005, July 1997.

[100] E. A. Vittoz, M. G. R. Degrauwe, and S. Bitz, "High-performance crystal oscillator circuits: Theory and application," *IEEE J. Solid-State Circuits*, vol. 23, no. 3, pp. 774–83, June 1998.

[101] Q. Huang and P. Basedau, "Design considerations for high-frequency crystal oscillators digitally trimmable to sub-ppm accuracy," *IEEE Trans. VLSI Syst.*, vol. 5, no. 4, pp. 408–16, Dec. 1997.

[102] B. Kim, T. C. Weigandt, and P. R. Gray, "Pll/dll system noise analysis for low jitter clock synthesizer design," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'94)*, June 1994, pp. 31–4.

[103] T. C. Weigandt, B. Kim, and P. R. Gray, "Analysis of timing jitter in cmos ring oscillators," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'94)*, June 1994, pp. 27–30.

[104] J. G. Maneatis, "Low-jitter process-independent dll and pll based on self biased techniques," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1723–32, Nov. 1996.

[105] A. A. Abidi and R. G. Meyer, "Noise in relaxation oscillators," *IEEE J. Solid-State Circuits*, vol. 18, no. 6, pp. 794–802, Dec. 1983.

[106] B. S. Kim, "High speed clock recovery in vlsi using hybrid analog/digital techniques," Ph.D. dissertation, University of California at Berkeley, Berkeley, CA, 1990.

[107] I. A. Young, J. K. Greason, and K. L. Wong, "A pll clock generator with 5 to 110 mhz of lock range for microprocessors," *IEEE J. Solid-State Circuits*, vol. 27, no. 11, pp. 1599–1607, Nov. 1992.

[108] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of mos transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–40, Oct. 1989.

[109] G. Wegmann, E. A. Vittoz, and F. Rahali, "Charge injection in analog mos switches," *IEEE J. Solid-State Circuits*, vol. 22, no. 6, pp. 1091–7, Dec. 1987.

[110] R. Brederlow, W. Weber, J. Sauerer, S. Donnay, P. Wambacq, and M. Vertregt, "A mixed-signal design roadmap," *IEEE Des. Test. Comput.*, vol. 18, no. 6, pp. 34–46, Nov./Dec. 2001.

[111] G. Roubik, *Introduction to CMOS OP-AMPs and Comparators*, 1st ed. New York: John Wiley & Sons, 1999.

[112] J. Yuan and C. Svensson, "New single-clock cmos latches and flipflops with improved speed and power savings," *IEEE J. Solid-State Circuits*, vol. 32, no. 1, pp. 62–9, Jan. 1997.

[113] B. J. McCarroll, C. G. Sodini, and H.-S. Lee, "A high-speed cmos comparator for use in an adc," *IEEE J. Solid-State Circuits*, vol. 23, no. 1, pp. 159–65, Feb. 1988.

[114] B. Razavi and B. A. Wooley, "Design techniques for high-speed, high-resolution comparators," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1916–26, Dec. 1992.

[115] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. New Jersey: Prentice Hall, 2003.

[116] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*, 1st ed. Cambridge, UK: Cambridge University Press, 1998.

[117] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power cmos digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–84, Apr. 1992.

[118] D. Marković, "Analysis and design of low-energy clocked storage elements," Master's thesis, University of California at Berkeley, Berkeley, CA, 2000.

[119] T. Burd, "Low power cmos library design methodology," Master's thesis, University of California at Berkeley, Berkeley, CA, 1994.

[120] I. D. O'Donnell, "Digital circuit and board design for a low power, wideband cdma receiver," Master's thesis, University of California at Berkeley, Berkeley, CA, 1996.

[121] H. W. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic*, 1st ed. New Jersey: Prentice Hall, 1993.

Appendix A

BIT Register list

This appendix documents the registers in the UWB BIT chip.

Format:

Address: Bit [31:0] Name
Description

Registers:

Addr: 00 [21] DLLobsvctrl
Addr: 00 [20] DLLalwaysconnect
Addr: 00 [19] DLLalwaysmirror
Addr: 00 [18] DLLalwaysbiascp
Addr: 00 [17:12] DLLcpbias[5:0]
Addr: 00 [11:08] BSpbias[3:0]
Addr: 00 [07:04] BSpmirr[3:0]
Addr: 00 [03:00] BSnbias[3:0]

BIAS_CORE:

pbias sets # of lsb's = Iout at RPSET
pmirr sets # of lsb's mirror to nmos bias line
Note if pmirr is 0, then Inlsb is set by optional Irnset current.
nbias sets # of lsb's for nmos
So, for Irpset, Iplsb = Irpset/pbias and
Inlsb = Irpset*(pmirr/pbias)/nbias

DLL_CORE:

cpbias sets # of Iplsb's of bias for charge pump

alwaysbiascp runs Iplsb bias current for charge pump continuously (no duty-cycling on endllbias)

alwaysmirror continuously mirrors pbias vctrl through current mirror to nbias vctrl (no duty-cycling on endllbias)

alwaysconnect continuously connects nbias vctrl to the diode mirror. (If you duty-cycle the mirror current, you should disconnect the nbias vctrl line from the diode, or it will perturb the value (i.e. drain charge off the nbias vctrl cap.) If duty-cycled, controlled by endll.)

obsvctrl opens pass-gate to VCTRL pin of the pbias vctrl node (summing node in DLL)

Addr: 01 [30:00] _GMR0_trim

GAIN_CORE:

_trim0_out turns on PMOS switches to change the Rload seen on gmr0's _OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 02 [30:00] _GMR0trim

GAIN_CORE:

_trim0out turns on PMOS switches to change the Rload seen on gmr0's OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 03 [30:00] _GMR1trim

GAIN_CORE:

_trim1out turns on PMOS switches to change the Rload seen on gmr1's OUT node. Note this is like a thermometer

code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 04 [30:00] _GMR1_trim

GAIN_CORE:

_trim1_out turns on PMOS switches to change the Rload seen on gmr1's _OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 05 [23:19] GMR3bias[4:0]
Addr: 05 [18] GMR3alwasy
Addr: 05 [17:13] GMR2bias[4:0]
Addr: 05 [12] GMR2always
Addr: 05 [11:07] GMR1bias[4:0]
Addr: 05 [06] GMR1always
Addr: 05 [05:01] GMR0bias[4:0]
Addr: 05 [00] GMR0always

GAIN_CORE:

gmr[0:3]always turns on bias to GMR[0:3] stages respectively (i.e., no duty-cycling, always is on)

bias[0:3]gmr sets #*Inlsb as bias current in respective stage

---

Addr: 06 [31:27] TIAtrim[4:0]
Addr: 06 [26:22] TIA_trim[4:0]
Addr: 06 [21:20] TIAbiasptrim[1:0]
Addr: 06 [19:15] TIAbiasp[4:0]
Addr: 06 [14:13] TIAbiasplsb[1:0]
Addr: 06 [12:07] TIAbiasn[5:0]
Addr: 06 [06:01] TIA_biasn[5:0]
Addr: 06 [00] TIAalways

GAIN_CORE:

TIAalways turns on bias to TIA always (no duty-cycling)
biasntia sets #*2*Inlsb in I1 nmos for TIA OUT (_VIN)
biasn_tia sets #*2*Inlsb in I1 nmos for TIA _OUT (VIN)
biastiaplsb sets 0 to 3 Iplsb's*(10/8) as p bias lsb
biasptia sets I2 in both branches as # * p bias lsb above
biastiaptrim sets 0 to 3 Iplsb's*(2/8) as p trim lsb
trimtia pushes # * p trim lsb's into TIA OUT
trim_tia pushs # * p trim lsb's into TIA _OUT

Addr: 07 [00] _GMR2_trim

GAIN_CORE:

_trim2_out turns on PMOS switches to change the Rload seen on gmr2's _OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 08 [30:00] _GMR2trim

GAIN_CORE:

_trim2out turns on PMOS switches to change the Rload seen on gmr2's OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 09 [30:00] _GMR3trim

GAIN_CORE:

_trim3out turns on PMOS switches to change the Rload seen on gmr3's OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 10 [30:00] _GRM3_trim

GAIN_CORE:

_trim3_out turns on PMOS switches to change the Rload seen on gmr3's _OUT node. Note this is like a thermometer code. Default is all on (all zero), then to adjust offset, set 1 to next nearest msb. I.e. 0x00000000, 0x80000000, 0xC0000000, 0xE0000000, 0xF0000000, 0xF8000000. Finer adjust on offset can be achieved by killing the lsb's one at a time if needed/desired.

Addr: 11 [29:24] OUT_bias[5:0]
Addr: 11 [23:18] OUTbias[5:0]
Addr: 11 [17:13] PRE_trim[4:0]
Addr: 11 [12:08] PREtrim[4:0]
Addr: 11 [07:06] PREbiasptrim[1:0]
Addr: 11 [05:01] PREbias[4:0]
Addr: 11 [00] OUTalways

GAIN_CORE:

outalways turns on output 50-driver. This can't be duty-cycled, so this is the enable (also enables preamp)
biaspre sets #*2*Inlsb as bias current for driver pre-amp
biaspretrim sets 0 to 3 Iplsb/8 as pre trim lsb current
trimpre pulls # of pre trim lsb's from buffer PRE
trim_pre pulls # of pre trim lsb's from buffer _PRE
Note: the 50-driver is a NMOS source follower preceeded by the same buffer cell used to drive the ADC as a preamp. The preamp, thus, can used to adjust offset in OUT,_OUT.
biasout sets #*10*Inlsb as Ibias in OUT leg of driver.
bias_out sets #*10*Inlsb as Ibias in _OUT leg of driver.
Note, coarse offset reduction in OUT,_OUT may be done by changing the leg bias currents.

Addr: 12 [26] OUTobs
Addr: 12 [25] OUTbuf
Addr: 12 [24] OUTtia
Addr: 12 [23] OBSbuf
Addr: 12 [22] OBSgmr3
Addr: 12 [21] OBSgmr2

Addr: 12 [20] OBSgmr1
Addr: 12 [19] OBSgmr0
Addr: 12 [18] OBStia
Addr: 12 [17:13] BUF_trim[4:0]
Addr: 12 [12:08] BUFtrim[4:0]
Addr: 12 [07:06] BUFbiasprtrim[1:0]
Addr: 12 [05:01] BUFbias[4:0]
Addr: 12 [00] BUFalways

GAIN_CORE:

bufalways turns on bias to buffer stage always (no duty-cycle)
biasbuf sets #*2*Inlsb as bias current for buffer
biasbuftrim sets 0 to 3 Iplsb/8 as buf trim lsb current
trimbuf pulls # of buf trim lsb's from buffer OUT
trim_buf pulls # of buf trim lsb's from buffer _OUT
OBStia sends TIA diff output to {OBS,_OBS} pins
OBSgmr0 sends gmr0 diff output to {OBS,_OBS} pins
OBSgmr1 sends gmr1 diff output to {OBS,_OBS} pins
OBSgmr2 sends gmr2 diff output to {OBS,_OBS} pins
OBSgmr3 sends gmr3 diff output to {OBS,_OBS} pins
OBSbuf sends buf diff output to {OBS,_OBS} pins

Note: Only one OBS should be on at any one time unless you want to *dc* short outputs (you don't.) This is to allow for *dc* VOS measurements off chip and calibration. Path to OBS can't handle freq > 10-ish MHz

OUTtia sends tia diff output to {OUT,_OUT} buffer/driver
OUTbuf sends buf diff output to {OUT,_OUT} buffer/driver
OUTobs sends obs diff input to {OUT,_OUT} buffer/driver

Note: Only one OUT should be on at any one time unless you want to short the stages (you don't.) Output buffer will drive 50Ω load (when biased properly) and should be able to handle up to $\sim 1GHz$ bandwidth. OBS,_OBS may be sent to OUT,_OUT for purposes of calibrating the driver independently.

Addr: 13 [23:14] Cchkshift[9:0]
Addr: 13 [13] Ccyset
Addr: 13 [12] Ccyrst
Addr: 13 [11:02] Cdldelay[9:0]
Addr: 13 [01] Cdlset
Addr: 13 [00] Cdlrst

CTRL_CORE:

dlrst forces delay counter to reset (turns off 'GO')
dlset forces delay counter to be set (like 'GO')
dldelay counts down # of clk cycles before issuing 'go'
to cycle counter. Used to control delay from
'GO' to pulse gen or start of pulse reception
to compensate for time-of-flight or whatever.
cyrst forces cycle counter to reset (turns of sampling/pulsing)
cyset forces cycle counter to start
chkshift value of cycle counter to check (clock in) digital
backend shift (and direction) control signals. If
shift is signalled, it will happen at next cycle=
chkplus or chkminus as appropriate.

Addr: 14 [29:20] Cchkplus[9:0]
Addr: 14 [19:10] Cchkminus[9:0]
Addr: 14 [09:00] Cchkload[9:0]

CTRL_CORE:
chkload value of cycle counter to trigger reload
chkminus value of cycle counter to execute shift back
signal from digital backend
chkplus value of cycle counter to execute shift forward
signal from digital backend

Addr: 15 [29:20] Crplus[9:0]
Addr: 15 [19:10] Crminus[9:0]
Addr: 15 [09:00] Crload[9:0]

CTRL_CORE:
rload value of cycle counter to reload to on chkload
rminus value of cycle counter to shift to on chkminus
rplus value of cycle counter to shift to on chkplus

Addr: 16 [26] TXenobs
Addr: 16 [25] TXmode
Addr: 16 [24] TXCLKreg
Addr: 16 [23] TXCLKoverride
Addr: 16 [22] DIGSHIFTreg
Addr: 16 [21] DIGDIRreg
Addr: 16 [20] DIGSHIFToverride
Addr: 16 [19] DIGCLKreg

Addr: 16 [18]	DIGCLKoverride
Addr: 16 [17]	CselDIGCLK
Addr: 16 [16]	Cobsencalclk
Addr: 16 [15:13]	Cobsaddr[2:0]
Addr: 16 [12]	Cinvclkobs
Addr: 16 [11]	Cenclkobs
Addr: 16 [10:09]	CselOBSCLK[1:0]
Addr: 16 [08:04]	Cpppasscnt[4:0]
Addr: 16 [03]	Cppenobs
Addr: 16 [02]	Cpprstp2p
Addr: 16 [01]	Cpprstcnt
Addr: 16 [00]	Cadrst

CTRL_CORE:

adrst reset state in ctrl_adc – turns DLL phases in ADC sampling phases and generates some control for the parallel to parallel converter (p2p)

pprstcnt force p2p index to reset (next sampled stored at 0)

pprstp2p force p2p registers to reset (all p2p values go to 0)

ppenobs allow clkin to grab input values into obs registers (obs registers go to mux tree to output so the ADC outputs may be inspected in this way.) They can also be inspected using obsaddr.

pppasscnt[0:7] one-hot encoded state for length of p2p conversion. 0x01 is one bank (32-bits), 0x02 is 2 (64-bits), 0x80 is 8 banks (256-bits – the max)

selOBSCLK[0:1] choses which CLK is sent to the OBSCLK pin:
- 0: chose DIGCLK to display
- 1: chose TXCLK to display
- 2: choses OBSCLK from CTRL_CORE (useful with the obsaddr[*] bus below to flop ctrl state.)
- 3: same as 2, high bit choses OBSCLK from CTRL

enclkobs turn on output clk (OBSCLK) that should give a rising edge when OBS data is valid for grabbing. OBS data may be viewed through the 32-bit interface by reading the correct addr and setting obsaddr

_invclkobs invert ouptut clk (OBSCLK) in case falling edge or other rising edge is desired. (inv when = '0')

obsaddr choose bank of 32 control signals in CTRL_CORE for output viewing. Roughly, the addresses are:
- 0: delay and cycle counter state
  - [31] = cyshiftdone
  - [30] = cyshift
  - [29] = cyatminus

[28] = cyatplus
[27] = cyplusminus
[26] = cyshiftstate
[25] = cyatshift_d
[24] = cyload
[23] = cycle[9]
[22] = cycle[8]
[21] = cycle[7]
[20] = cycle[6]
[19] = cycle[5]
[18] = cycle[4]
[17] = cycle[3]
[16] = cycle[2]
[15] = cycle[1]
[14] = cycle[0]
[13] = cyactive
[12] = dltrigger
[11] = delay[9]
[10] = delay[8]
[09] = delay[7]
[08] = delay[6]
[07] = delay[5]
[06] = delay[4]
[05] = delay[3]
[04] = delay[2]
[03] = delay[1]
[02] = delay[0]
[01] = dlactive
[00] = clk

1: cycle counter state + generated control signals with digital shift/dir info.
[31] = dltrigger
[30] = dlactive
[29] = gnd
[28] = gnd
[27] = engainbias
[26] = enclkin
[25] = endllbias
[24] = endll
[23] = sampadc
[22] = clkb
[21] = cyshiftdone
[20] = cyshift
[19] = cyatminus

[18] = cyatplus
[17] = cyplusminus
[16] = cyshiftstate
[15] = digdir
[14] = digshift
[13] = cyatshift_d
[12] = cyload
[11] = cycle[9]
[10] = cycle[8]
[09] = cycle[7]
[08] = cycle[6]
[07] = cycle[5]
[06] = cycle[4]
[05] = cycle[3]
[04] = cycle[2]
[03] = cycle[1]
[02] = cycle[0]
[01] = cyactive
[00] = clk

2: ctrl_adc and p2p state
[31] = clkb
[30] = OBSCLK
[29] = reset
[28] = DIGCLKraw
[27] = gs0always
[26] = ppenDigAO
[25] = ppdone
[24] = gnd
[23] = ppenobs
[22] = gnd
[21] = ppenbank[4]
[20] = ppenbank[3]
[19] = ppenbank[2]
[18] = ppenbank[1]
[17] = ppenbank[0]
[16] = adenDIGDC
[15] = adclk12
[14] = aden12
[13] = adclk14
[12] = aden14
[11] = adclk34
[10] = aden34
[09] = adenQd[2]
[08] = adenQd[1]

[07] = adenQd[0]
[06] = adenadc[4]
[05] = adenadc[3]
[04] = adenadc[2]
[03] = adenadc[1]
[02] = adenadc[0]
[01] = sampadc
[00] = clk

3: p2p obs data
[31] = P2P_OBS[31]
[30] = P2P_OBS[30]
[29] = P2P_OBS[29]
[28] = P2P_OBS[28]
[27] = P2P_OBS[27]
[26] = P2P_OBS[26]
[25] = P2P_OBS[25]
[24] = P2P_OBS[24]
[23] = P2P_OBS[23]
[22] = P2P_OBS[22]
[21] = P2P_OBS[21]
[20] = P2P_OBS[20]
[19] = P2P_OBS[19]
[18] = P2P_OBS[18]
[17] = P2P_OBS[17]
[16] = P2P_OBS[16]
[15] = P2P_OBS[15]
[14] = P2P_OBS[14]
[13] = P2P_OBS[13]
[12] = P2P_OBS[12]
[11] = P2P_OBS[11]
[10] = P2P_OBS[10]
[09] = P2P_OBS[9]
[08] = P2P_OBS[8]
[07] = P2P_OBS[7]
[06] = P2P_OBS[6]
[05] = P2P_OBS[5]
[04] = P2P_OBS[4]
[03] = P2P_OBS[3]
[02] = P2P_OBS[2]
[01] = P2P_OBS[1]
[00] = P2P_OBS[0]

4: re-aligned ADC data Q[0:31]
[31] = Q[31]
[30] = Q[30]

[29] = Q[29]
[28] = Q[28]
[27] = Q[27]
[26] = Q[26]
[25] = Q[25]
[24] = Q[24]
[23] = Q[23]
[22] = Q[22]
[21] = Q[21]
[20] = Q[20]
[19] = Q[19]
[18] = Q[18]
[17] = Q[17]
[16] = Q[16]
[15] = Q[15]
[14] = Q[14]
[13] = Q[13]
[12] = Q[12]
[11] = Q[11]
[10] = Q[10]
[09] = Q[9]
[08] = Q[8]
[07] = Q[7]
[06] = Q[6]
[05] = Q[5]
[04] = Q[4]
[03] = Q[3]
[02] = Q[2]
[01] = Q[1]
[00] = Q[0]

5: un-aligned, raw ADC output data A[0:31]
[31] = A[31]
[30] = A[30]
[29] = A[29]
[28] = A[28]
[27] = A[27]
[26] = A[26]
[25] = A[25]
[24] = A[24]
[23] = A[23]
[22] = A[22]
[21] = A[21]
[20] = A[20]
[19] = A[19]

[18] = A[18]
[17] = A[17]
[16] = A[16]
[15] = A[15]
[14] = A[14]
[13] = A[13]
[12] = A[12]
[11] = A[11]
[10] = A[10]
[09] = A[9]
[08] = A[8]
[07] = A[7]
[06] = A[6]
[05] = A[5]
[04] = A[4]
[03] = A[3]
[02] = A[2]
[01] = A[1]
[00] = A[0]
6: DLL phases [0:31]
[31] = DLL[31]
[30] = DLL[30]
[29] = DLL[29]
[28] = DLL[28]
[27] = DLL[27]
[26] = DLL[26]
[25] = DLL[25]
[24] = DLL[24]
[23] = DLL[23]
[22] = DLL[22]
[21] = DLL[21]
[20] = DLL[20]
[19] = DLL[19]
[18] = DLL[18]
[17] = DLL[17]
[16] = DLL[16]
[15] = DLL[15]
[14] = DLL[14]
[13] = DLL[13]
[12] = DLL[12]
[11] = DLL[11]
[10] = DLL[10]
[09] = DLL[9]
[08] = DLL[8]

[07] = DLL[7]
[06] = DLL[6]
[05] = DLL[5]
[04] = DLL[4]
[03] = DLL[3]
[02] = DLL[2]
[01] = DLL[1]
[00] = DLL[0]
7: all phases are = CLK (for calibration of delay)
[31:00] = CLK

obsencalclk - if set, passes CLK to addr7 of obsaddr mux above. Can be used to measure/calibrate path delays on each phase relative to a common/known starting pt. If set to '0', and obsaddr=7, then no ctrlobs activity.

selDIGCLK choses which ctrl subblock generates the digital clock. '1' for the p2p convertor's 'done', or '0' for adc ctrl's A*(!B) on delayed 'sampadc.' Hmmm. Why two options? Dunno. Probably an oversight. I should think that '1' would always work, but YMMV.

overrideDIGCLK set true if you want to drive the digital clock from regDIGCLK, instead of from ctrl_core. Useful for independently debugging digital backend. (Also use regdin[0:159:1] registers below to override the input.)

regDIGCLK register holds current value of DIG CLK override, like for TXCLK above.

overrideShift overrides digital SHIFT and DIR feedback signals with the value of registers 'regSHIFT' and 'regDIR.'

regDIR overrride value for DIGDIR

regSHIFT override value for DIGSHIFT

overrideTXCLK set true if you want to drive the TX clock from regTXCLK, instead of from ctrl_core. Useful for independently debugging TX.

regTXCLK register holds current value of TX CLK (i.e. write 0, then write 1, to get a rising edge.) Note that this is aligned to OSCCLK's rising edge internally. This allows for CLKIN to override OSCCLK and keep timing between the digital interface and whatever is overriding (i.e. microprocessor or FPGA.)

modeTX holds a 1-bit value indicated which 'mode' the TX is in. This affects the timing of the drive signals to the H-bridge and whether to quench an inductive current or dump built-up capacitive charge after sending a pulse. A '0' gives a simple pulse, i.e. $\pm A$ for T. A '1' gives a Manchester

pulse with *dc* value approx. zero, i.e.
$\pm A$ for T then $\mp A$ for T.

enobsTX If '0', AND's TX observation bus (32-bits) to zero so no edges/transition. If '1', then passes bus unharmed.

Addr: 17 [26] O2Cenobsbias
Addr: 17 [25] O2Cselrise
Addr: 17 [24] O2Cenoscbuf
Addr: 17 [23] O2Coverrideclkin
Addr: 17 [22] O2Cenclkout
Addr: 17 [21:14] OSCc2trim[7:0]
Addr: 17 [13:06] OSCc1trim[7:0]
Addr: 17 [05:00] OSCbiasp[5:0]

OSC_CORE:

pbias sets # of Iplsb's of current for osc PMOS source
c1trim sets C1 (Cgate) value for osc
c2trim sets C2 (Cdrain) value for osc

OSC2CLK_CORE:

enclkout sends clk to output pin CLKOUT
overrideclkin uses CLKIN for clk instead of OSCG
enoscbuf turns on oscin to CLK bias circuitry
oscselrise choses rising or falling edge of oscin as system clk's rising edge

---

Addr: 18 [30] Cgs4set
Addr: 18 [29] Cgs4rst
Addr: 18 [28] Cgs3set
Addr: 18 [27] Cgs3rst
Addr: 18 [26] Cgs2set
Addr: 18 [25] Cgs2rst
Addr: 18 [24] Cgs1set
Addr: 18 [23] Cgs1rst
Addr: 18 [22] Cgs0set
Addr: 18 [21] Cgs0rst
Addr: 18 [20:11] Cgs0stop[9:0]
Addr: 18 [10:01] Cgs0start[9:0]
Addr: 18 [00] Cgs0always

CTRL_CORE:

gs0always turns on control signal as soon as cycle counter is activated and keeps it always on
gs0start cycle count value to turn on control signal

gs0stop cycle count value to turn off control signal
gs0rst turns off control signal 0 (sampadc)
gs0set turns on control signal
gs1rst turns off control signal 1 (endll)
gs1set turns on control signal
gs2rst turns off control signal 2 (endllbias)
gs2set turns on control signal
gs3rst turns off control signal 3 (enclkin)
gs3set turns on control signal
gs4rst turns off control signal 4 (engainbias)
gs4set turns on control signal

Addr: 19 [20:11] Cgs1stop[9:0]
Addr: 19 [10:01] Cgs1start[9:0]
Addr: 19 [00] Cgs1always

CTRL_CORE:
gs1always turns on control signal as soon as cycle counter is activated and keeps it always on
gs1start cycle count value to turn on control signal
gs1stop cycle count value to turn off control signal

Addr: 20 [20:11] Cgs2stop[9:0]
Addr: 20 [10:01] Cgs2start[9:0]
Addr: 20 [00] Cgs2always

CTRL_CORE:
gs2always turns on control signal as soon as cycle counter is activated and keeps it always on
gs2start cycle count value to turn on control signal
gs2stop cycle count value to turn off control signal

Addr: 21 [20:11] Cgs3stop[9:0]
Addr: 21 [10:01] Cgs3start[9:0]
Addr: 21 [00] Cgs3always

CTRL_CORE:
gs3always turns on control signal as soon as cycle counter is activated and keeps it always on
gs3start cycle count value to turn on control signal
gs3stop cycle count value to turn off control signal

Addr: 22 [20:11] Cgs4stop[9:0]
Addr: 22 [10:01] Cgs4start[9:0]
Addr: 22 [00] Cgs4always

CTRL_CORE:
gs4always turns on control signal as soon as cycle counter is activated and keeps it always on
gs4start cycle count value to turn on control signal
gs4stop cycle count value to turn off control signal

Registers below (with addresses ¿ 23) may be read, but not written:

Addr: 23 UNUSED

UNUSED:
Read should return all 0. (Clamped to zero.)

Addr: 24 [31:0] Cppout[31:0]

CTRL_CORE:
ppout[31:0] - lower word of parallel to parallel converter block in controller. (Can aggregate up to 5 words of ADC output and read stored results.)

Addr: 25 [31:0] Cppout[63:32]

CTRL_CORE:
ppout[63:32] - second word of parallel to parallel converter block in controller. (Can aggregate up to 5 samples of ADC output and read stored results.)

Addr: 26 [31:0] Cppout[95:64]

CTRL_CORE:
ppout[95:64] - third word of parallel to parallel converter block in controller. (Can aggregate up to 5 samples

of ADC output and read stored results.)

Addr: 27 [31:0] Cppout[127:96]

CTRL_CORE:

ppout[127:96] - lower word of parallel to parallel converter block in controller. (Can aggregate up to 5 samples of ADC output and read stored results.)

Addr: 28 [31:0] Cppout[159:128]

CTRL_CORE:

ppout[159:128] - upper word of parallel to parallel converter block in controller. (Can aggregate up to 5 samples of ADC output and read stored results.)

Addr: 29 UNUSED

UNUSED:

Read should return all 0. (Clamped to zero.)

Addr: 30 [06] inp2
Addr: 30 [05] inp1
Addr: 30 [04] inn2
Addr: 30 [03] inn1
Addr: 30 [02] input
Addr: 30 [01] mode
Addr: 30 [00] clk

TX_CORE:

'obs' (observation/debug) outputs from TX block.

clk is incoming clock (can be overridden in Addr: 16, otherwise clk is generated in CTRL_CORE by load signal in ctrlcycle block – see chkload in Addr: 14 and rload in Addr: 15)

mode is TXmode (set in Addr: 16)

input is input bit to TX block, comes from external pin to chip called 'txin' also called 'address[7].'

inn1 seems to be delayed signal that drives 'N' pulldown of H-bridge driver side '1.'

inn2 seems to be delayed signal that drives 'N'
pulldown of H-bridge driver side '2.'
inp1 seems to be delayed signal that drives 'P'
pulldown of H-bridge driver side '1.'
inp2 seems to be delayed signal that drives 'P'
pulldown of H-bridge driver side '2.'

Addr: 31 [31:00] ana_rout[31:0]

ANA_CORE:
32-bit word output 'obs' from analog block, selected by 'obsaddr' in Addr: 16. May be DLL phases; raw, unaligned ADC outputs; control state: i.e. delay cycle count, p2p state, etc.; etc.

Appendix B

BIT Testboard

The purpose of this appendix is to document the testboard setup, some aspects of its design, and the calibration testboards used for S-parameter measurements. The pins from the die are generally brought straight out to a connector (i.e. SMA or IDC) or a circuit component (i.e. bias potentiometer or crystal.) The power supplies are all separated, but derived through bypass and inductive filtering from one of three input sources: receive, digital, and transmit. In particular the analog receive power supplies have a separate supply for each block (and for the padring.) The digital I/O bus (and few analog receive block I/O's) are grouped together into three 20-pin headers at the top of the testboard.

A picture of the overall test setup is shown in figure B.1. A laptop PC, running a C-program for RS232 serial port communication was used to communicate through the RS232 level-translation board to a Xilinx Spartan-3 FPGA. The Xilinx FPGA implements a serial to parallel UART for programming (and downloading) the state of the impulse transceiver testchip on the testboard, and implements any digital backend functionality. A

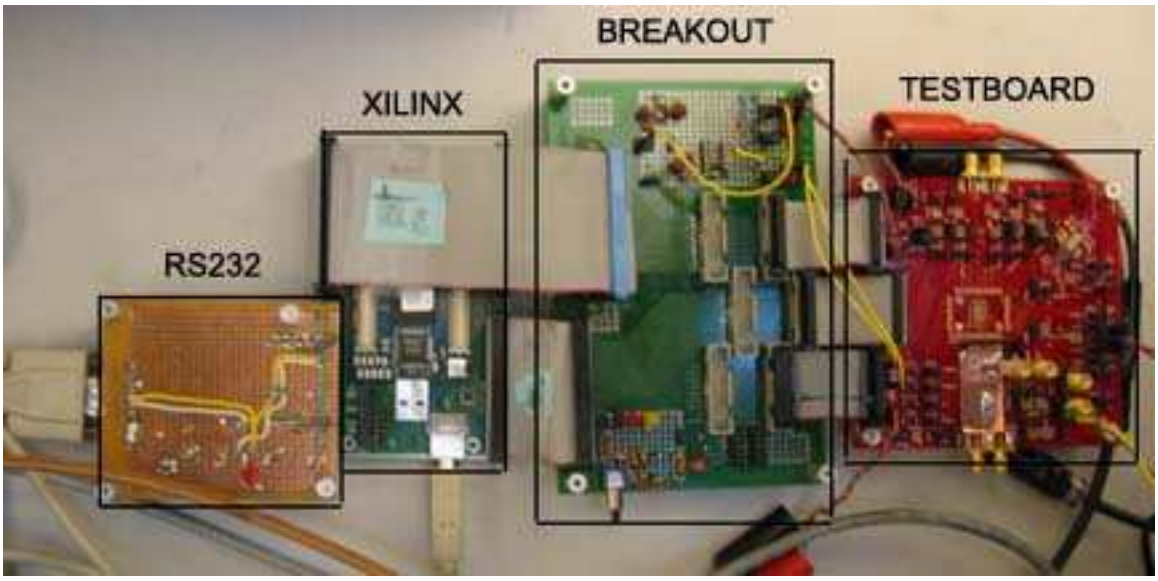

Figure B.1: Testboard Setup

breakout board was inserted in between the Xilinx digital backend and testchip to allow for signal inspection (via logic analyzer or oscilloscope probe) or generation (e.g. a global reset push button switch.) A photo of the testboard is shown in figure B.2. The testboard setup is based on a "daughter card" concept. As chip-on-board (COB) bonding directly bonds the die to a board, if a die fails (or is accidentally injured), the entire board may be scrapped. To reduce the loss for such an occurrence, an effort was made to only include the necessary connection and bias circuitry on the testboard itself. Signals are brought out to headers and generated on a separate board.

All of the reported measurements were from dies bonded COB to the testboard as shown in figure B.3. Chips were also packaged in a 128-pin OCP plastic package (14x20x2.7mm.) These packaged chips were expected to have a faster turn-around time than COB bonding, and were intended to be used to help bring up the test setup. Unfor-

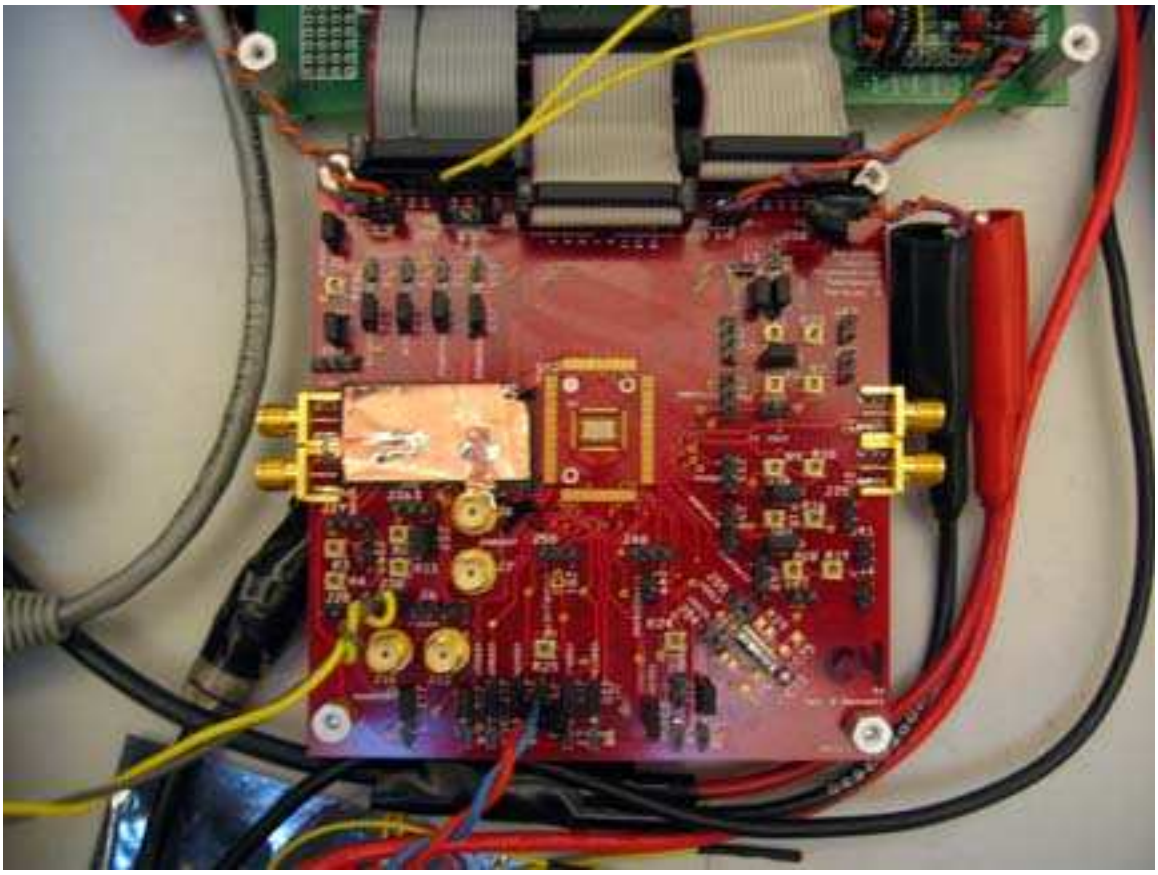

Figure B.2: Testboard

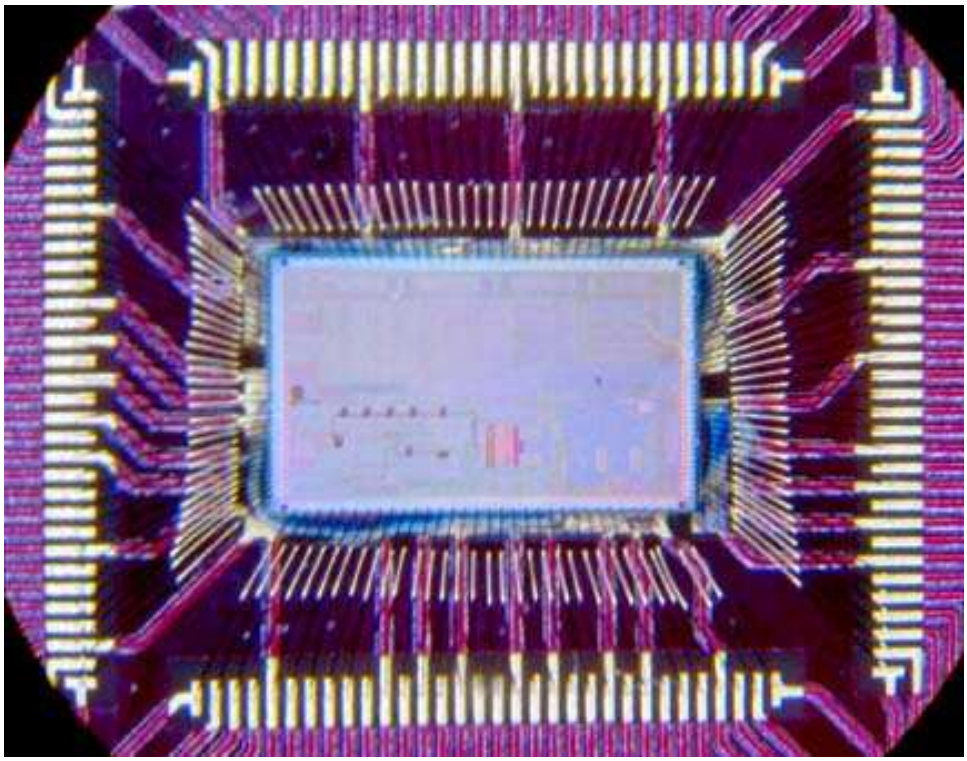

Figure B.3: Chip-On-Board Mounting of Die to Testboard

tunately due to a shipping mistake and other unexpected delays, they arrived many weeks later than the COB testboards. As performance was expected to be superior for COB testboards (vs. packaged parts) due to lower parasitic inductance, only the COB testboards were measured. A photo of a testboard with a packaged die is shown in B.4.

A photo of the TEM horn antenna used for correlation TX/RX measurements is shown in figure B.5. This antenna has good low frequency response and spans $100MHz$ to $1GHz$. It was built specially for this project.

One interesting aspect of the testboard design is the proper selection of bypass capacitor components. Due to series inductance in the packaging, on-board capacitors exhibit self-resonance and stop functioning as capacitors beyond that frequency. In fact, they may

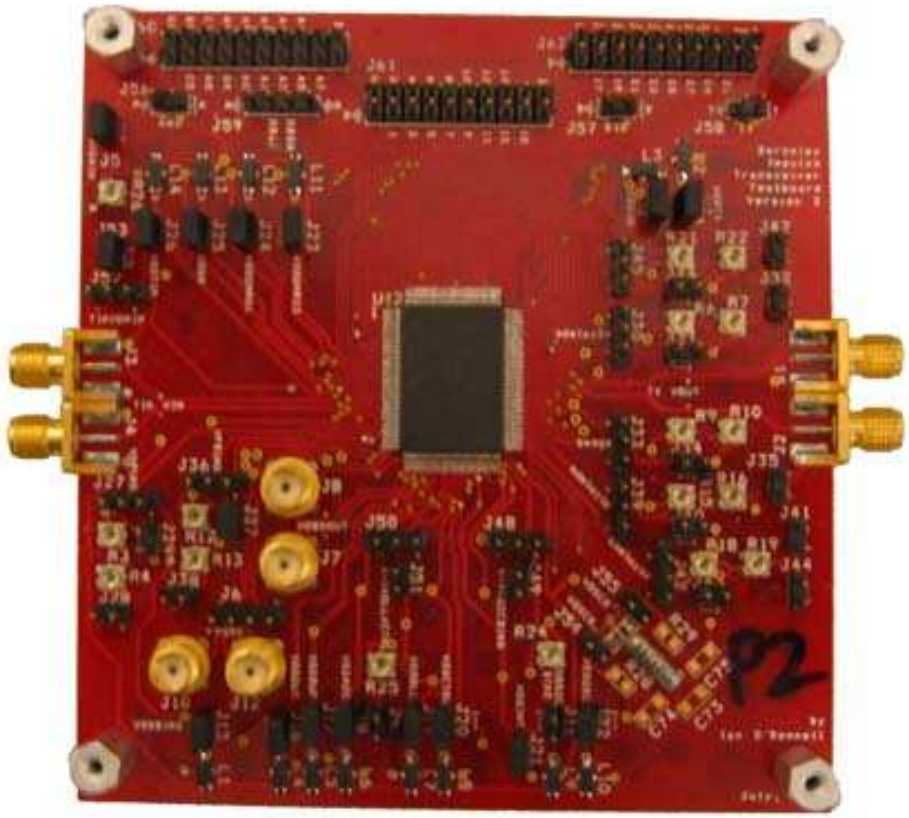

Figure B.4: Packaged Chip Version of Testboard

Figure B.5: TEM Horn Used for Transceiver Measurements

appear inductive and, at a particular frequency resonate with other bypass capacitors, thus increasing the impedance and reducing the bypass effect. Due to the large frequency range spanned during operation (on the order of a MHz to $1GHz$), the potential for this occurrence is high. Additionally, debugging the effect from these types of problems can be time consuming and difficult. As such, care must be taken when selecting components to ensure that no damaging parastic resonances are likely to occur. An example is shown in figure B.6 using three capacitors in parallel (one "fast," one "medium," and one "slow") with 1nH series packaging inductance. Note that there exists a strong peak near $20MHz$ where the bypass impedance is nearly $1000\times$ larger than at $2MHz$. Care was taken to ensure that the overall impedance stays beneath 10Ω (based on simulations of supply current spikes to limit the maximum voltage noise generated.) Otherwise, particular divider ratios for pulse rate generation may create supply-induced noise large enough to significantly impair operation. A good reference for high-frequency board design and bypass/decoupling concerns may be found in [121].

In order to more accurately measure the on-chip circuitry, calibration boards were also created to allow for de-embedding of the testboard characteristics for S-parameter measurements. To calibrate, four boards are needed for each input/output 2-port pair: open, short, load and through. The testboard has only one output (the 50Ω output buffer), but may have two inputs: from the TIA input, or from the observation/debug input. (The observation input may connect directly to the 50Ω output buffer, allowing us to separate the output driver from the internal circuitry.) For each calibration board, an attempt was made to make it as similar to ideal as possible. In the "through" case a small trace of

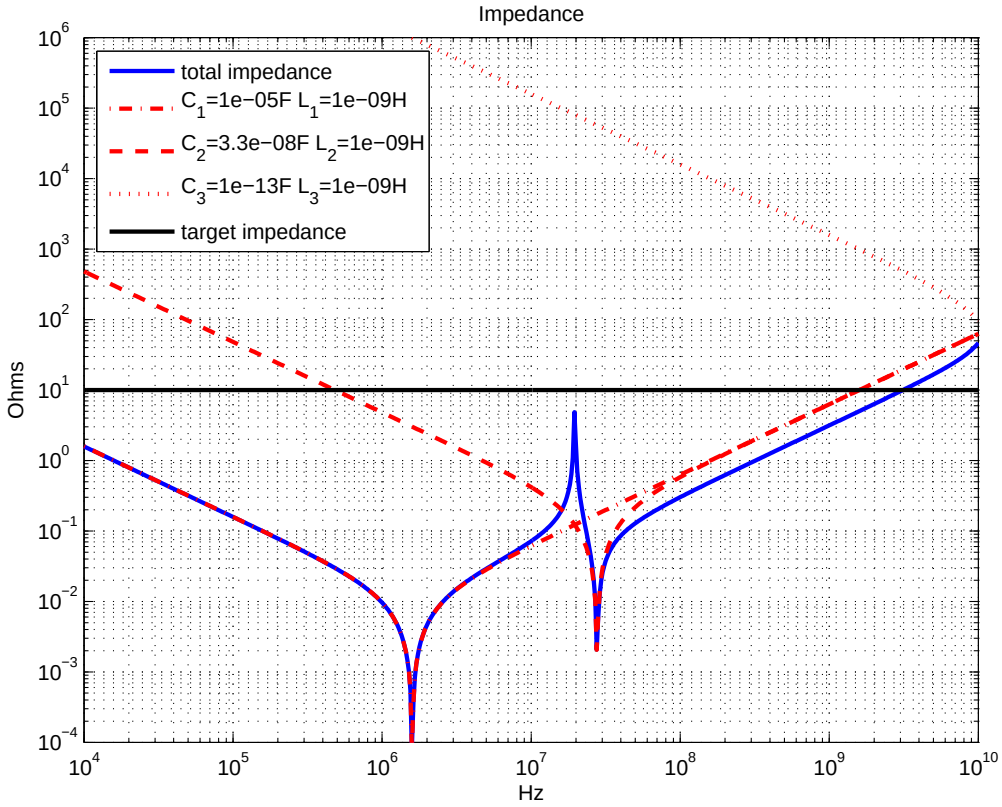

Figure B.6: Bypass Capacitor Impedance vs. Frequency

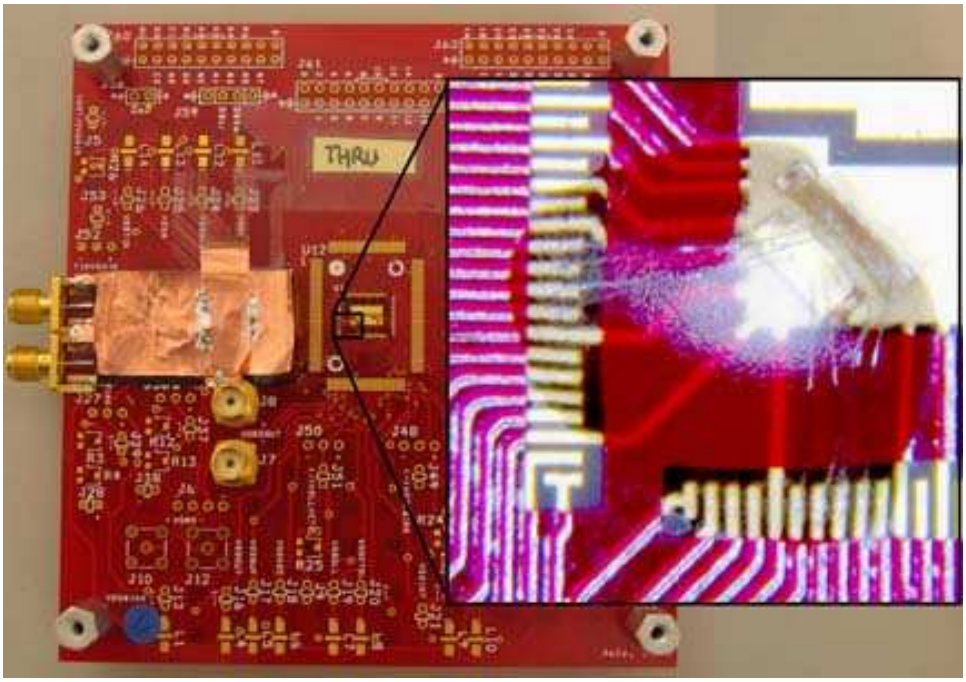

Figure B.7: Calibration Board Through for TIA Input to Output Buffer

metal (over an insulator; part of a route from an unused package) was glued to the COB bonding site and the input and output traces were bonded to this trace (shown in figure B.7 for the TIA to output buffer and in figure B.8 for the observation input to output buffer.) Termination was achieved using 0201 surface mount resistors of 50Ω value and bonding to one side of the pad, while grounding the other side (shown in figure B.9.) Short was directly bonded from the input/output pads to the COB bonding site (which is grounded.) This is shown in figure B.10. Open was simply left open, as shown in figure B.11. In hindsight, open is not quite accurate as the bond wires are not included. It would have been better to bond to an unconnected pad. While this oversight is regrettable, it is not believed that this would drastically change the overall results.

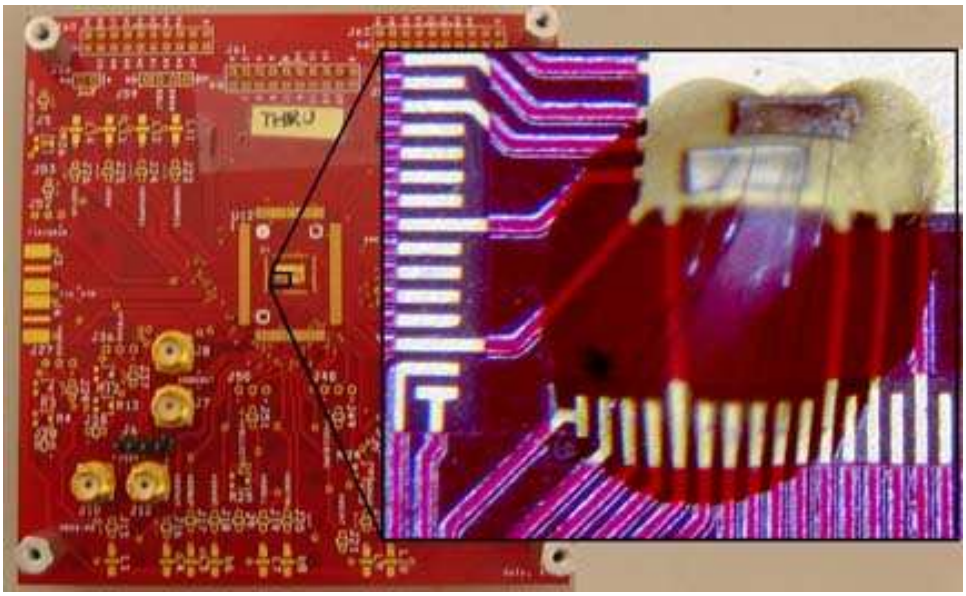

Figure B.8: Calibration Board Through for TIA Input to Output Buffer

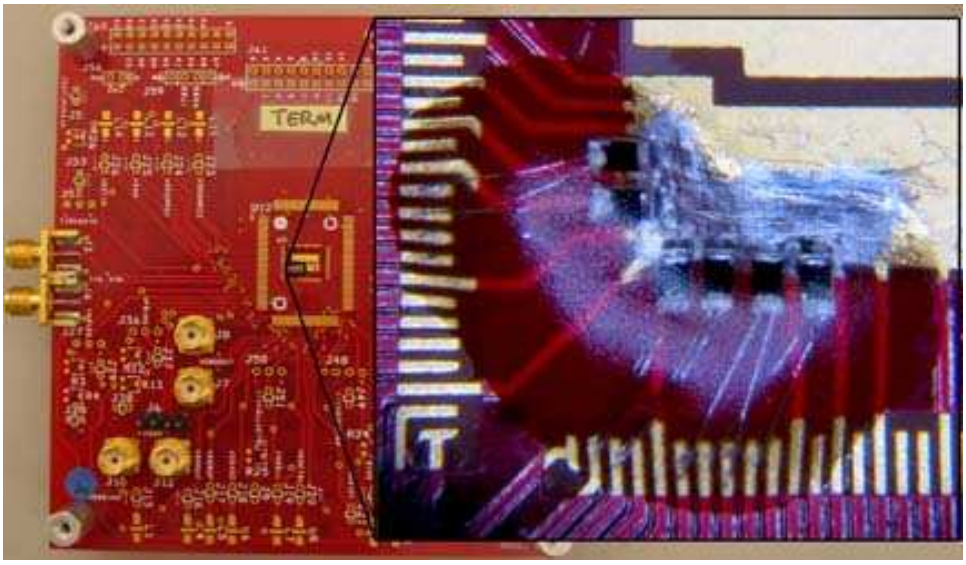

Figure B.9: Calibration Board Load/Termination

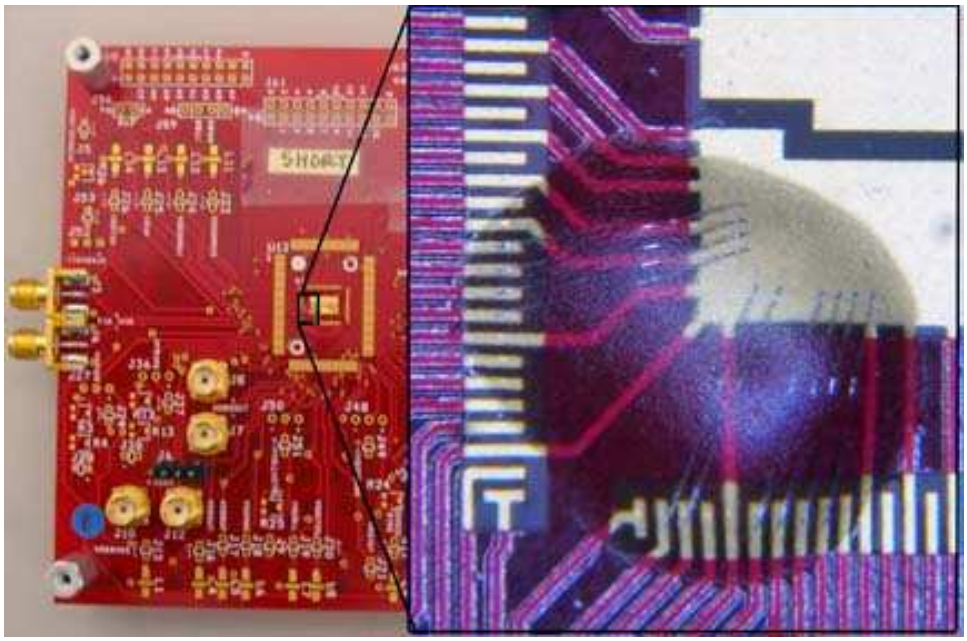

Figure B.10: Calibration Board Short

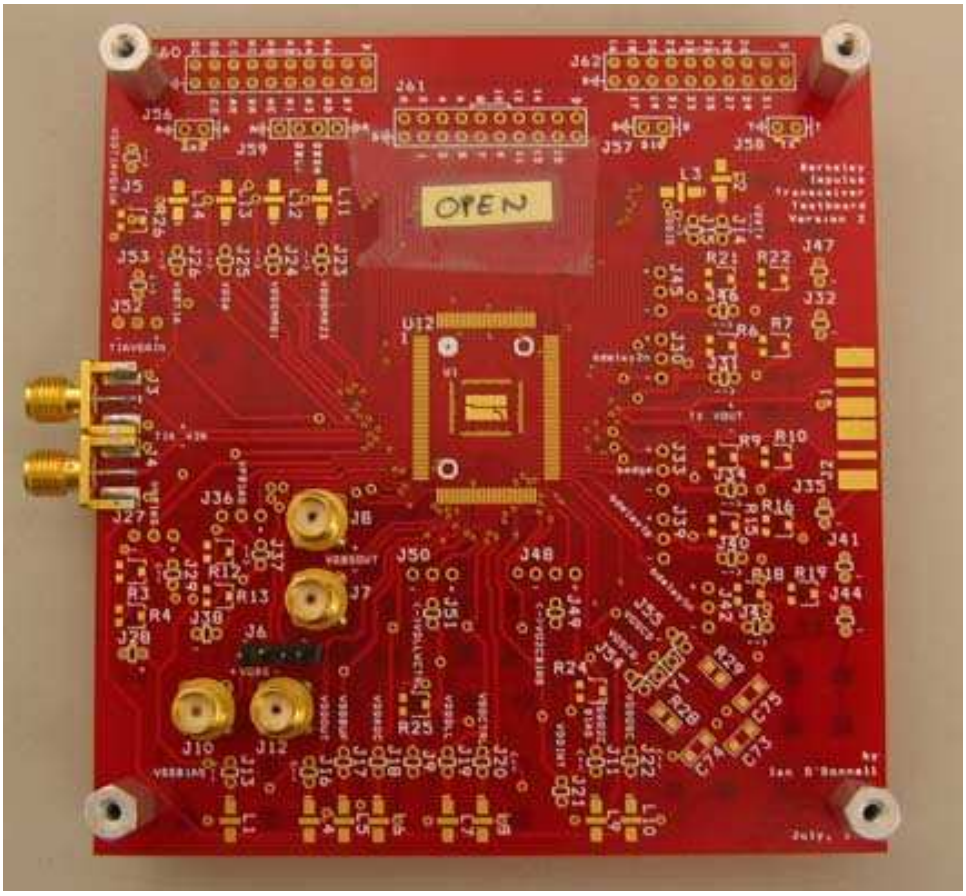

Figure B.11: Calibration Board Open

Appendix C

BIT Pinout

Figure C.1 shows the pad locations on the die. Pin #1 is in the lower right-hand corner and pad numbering proceeds in a clockwise order. Note that in the tables below an asterick (*) indicates that a pad exists but is also connected to its neighbor and hence not bonded to pinout as it is redundant. (This was necessary due to the peculiarity of use of VDD/GND pads and I/O, Core and Padring power supplies.)

Table C.1: Berkeley Impulse Transceiver (BIT) Pinout Receiver

No.	Name	No.	Name	No.	Name
1	IO_OSCD	**19**	IO_VDDADC	**37**	IO_VNBIAS
2	IO_OSCG	**20**	IO_GNDA	**38**	I_TIAVGAIN
3	IO_VDDOSC	**21**	IO_VDDBUF	**39**	IO_VDDTIA
4	IO_GNDA	**22**	IO_GNDA	**40**	IO_GNDA
5	IO_O2CBIAS	**23**	IO_VDDOUT	**41**	IO_VIN+
6	I_CLKIN	**24**	O_GOUT+	**42**	IO_VIN-
*****	IO_VDDA	**25**	O_GOUT-	**43**	IO_VIN+
7	IO_VDDA	**26**	O_GOUT+	**44**	IO_VIN-
8	IO_GNDA	**27**	O_GOUT-	**45**	IO_GNDA
*****	IO_GNDA	*****	IO_GNDA	**46**	IO_VDDTIA
9	O_CLKOUT	**28**	IO_GNDA	*****	IO_GNDA
10	IO_GNDA	**29**	IO_VDDA	**47**	IO_GNDA
11	IO_VDDINT	*****	IO_VDDA	**48**	IO_VDDA
12	O_OBSCLK	**30**	IO_OBS+	*****	IO_VDDA
13	IO_GNDA	**31**	IO_OBS-	**49**	IO_GNDA
14	IO_VDDCTRL	**32**	I_GO	**50**	IO_VDDGMR01
15	IO_GNDA	**33**	I_RESETA	**51**	IO_GNDA
16	IO_VDDDLL	**34**	IO_VDDBIAS	**52**	IO_VDDGMR23
17	IO_DLLVCTRL	**35**	IO_GNDA	**53**	IO_GNDA
18	IO_GNDA	**36**	IO_VPBIAS		

Table C.2: Berkeley Impulse Transceiver (BIT) Pinout Interface

No.	Name	No.	Name	No.	Name
54	I_RESETD	*	IO_VDDD	**91**	IO_DATA24
55	I_WRCLK	**73**	IO_VDDD	**92**	IO_DATA25
55	I_WRCLK	**74**	IO_GNDD	**93**	IO_DATA26
56	I_WREN	*	IO_GNDD	**94**	IO_DATA27
*	IO_VDDD	**75**	IO_DATA12	**95**	IO_DATA28
57	IO_VDDD	**76**	IO_DATA13	**96**	IO_DATA29
58	IO_GNDD	**77**	IO_DATA14	*	IO_VDDD
*	IO_GNDD	**78**	IO_DATA15	**97**	IO_VDDD
59	IO_DATA00	**79**	IO_DATA16	**98**	IO_GNDD
60	IO_DATA01	**80**	IO_DATA17	*	IO_GNDD
61	IO_DATA02	*	IO_VDDD	**99**	IO_DATA30
62	IO_DATA03	**81**	IO_VDDD	**100**	IO_DATA31
63	IO_DATA04	**82**	IO_GNDD	**101**	I_ADDR0
64	IO_DATA05	*	IO_GNDD	**102**	I_ADDR1
*	IO_VDDD	**83**	IO_DATA18	**103**	I_ADDR2
65	IO_VDDD	**84**	IO_DATA19	**104**	I_ADDR3
66	IO_GNDD	**85**	IO_DATA20	*	IO_VDDD
*	IO_GNDD	**86**	IO_DATA21	**105**	IO_VDDD
67	IO_DATA06	**87**	IO_DATA22	**106**	IO_GNDD
68	IO_DATA07	**88**	IO_DATA23	*	IO_GNDD
69	IO_DATA08	*	IO_VDDD	**107**	I_ADDR4
70	IO_DATA09	**89**	IO_VDDD	**108**	I_ADDR5
71	IO_DATA10	**90**	IO_GNDD	**109**	I_ADDR6
72	IO_DATA11	*	IO_GNDD	**110**	I_TXIN

Table C.3: Berkeley Impulse Transceiver (BIT) Pinout Transmitter

No.	Name
111	IO_GNDT
112	O_TX-
113	O_TX+
114	O_TX-
115	O_TX+
116	IO_GNDT
117	IO_VDDT
118	IO_GNDT
119	IO_VDDT
120	IO_GNDT
121	IO_VDDT
122	IO_GNDT
123	IO_VDDT
124	I_BTDELAY1
125	I_BTDELAY2
126	I_BTEDGE
127	I_BDEQP
128	I_BDEQN

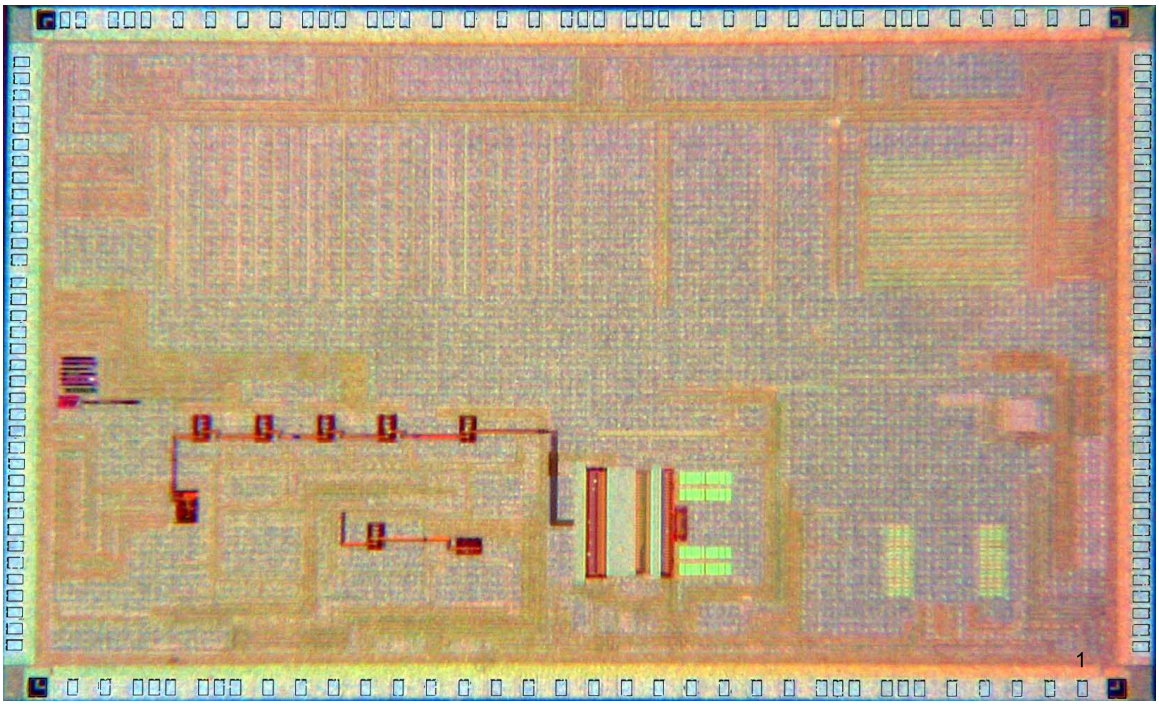

Figure C.1: Pinout Die Photo