Low Power, Crystal-Free Design for Monolithic Receivers

by

Bradley Wheeler

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 Kristofer S. J. Pister, Chair
Professor Ali M. Niknejad
Professor Steven D. Glaser

Spring 2019

Low Power, Crystal-Free Design for Monolithic Receivers

Abstract

Low Power, Crystal-Free Design for Monolithic Receivers

by

Bradley Wheeler

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kristofer S. J. Pister, Chair

Predictions of the proliferation of hundreds of billions of connected wireless devices have yet to come true. The economics of such deployments becoming feasible require that current wireless modules become smaller, cheaper, and use less power. A typical wireless device combines a RF System-on-Chip with multiple frequency references, passive components, an antenna, and a battery on a printed circuit board. The Single Chip Mote project aims to reduce the size, weight, power, and cost of these devices by eliminating the off-chip frequency references and passives. The ultimate goal being to form a 2.4 GHz wireless node by attaching only an antenna and energy source to a single CMOS die. Of particular interest is the range of applications this could enable where the size and weight of current wireless devices has prohibited their use.

This work implements a crystal-free IEEE 802.15.4 receiver that covers the data path from RF to bits. The receiver utilizes a passive front-end to reduce power and quadrature down-conversion followed by on-chip filtering and digitization. Integrated digital baseband is included for demodulation and clock recovery as well as built-in estimation of the errors in the RF channel frequency and data rate. Initial frequency calibration is performed simultaneously with bootloading using contact-less optical programming. Operation across the 0 - 70 C commercial temperature range has been demonstrated while inter-operating with commercial off the shelf IEEE 802.15.4 devices. The analog portion of the receiver, including the free-running LO, consumes 1.03 mW from a 1.5 V battery while achieving a sensitivity of -83 dBm.

To Cam

Thanks for putting up with all of this.

Contents

Acknowledgments

The Single Chip Mote was an ambitious undertaking for a small academic team and would not have been possible without the contributions of many people. The project never would have made it if the core SCM team of Filip Maksimovic, David Burnett, Osama Khan, and myself had not held together for the five years it took to get here. Fil deserves a shout-out as our fates were intertwined from early on. Rarely is it a good idea to put the success of any part of your project in the hands of another graduate student, but for us I think it worked out better in the end.

Special thanks to my advisor Kris Pister whose relentless optimism and drive got SCM to where it is today (how hard could it be?). Every advisor rolls the dice when they take new students and I'm forever grateful he took a chance on me. Six years later and it still feels like blind luck that I've ended up here. Thanks also to Ali Niknejad for the technical guidance and inspiration along the way.

A tremendous amount of the digital backbone of SCM exists because of the tireless efforts of Sahar Mesri. Thanks to Bob Zhou and Jonathon Wang for all the help with Verilog development and testing. Lydia Lee deserves props for the thankless job of slugging it out with the digital tool-flow setup. The development of the optical bootloader for SCM would not have happened without Andy Ng. A long list of people have contributed in various ways to the testing and development of SCM, thanks to Alex Moreno, Brian Kilberg, Nima Baniasadi, Hall Chen, Paul Kwon, Arvind Sundararajan, and Lorenz Schmid. The feedback of the OpenWSN team and other software developers, Tengfei Chang, Thomas Watteyne, Xavi Vilajosana, and Ioana Suciu was instrumental in steering SCM toward a usable implementation.

Lastly, thanks to the students and staff of both the Berkeley Sensor and Actuator Center and the Berkeley Wireless Research Center. These two organizations are filled with some of the most amazing people I have ever met.

Chapter 1

Overview

1.1 The Single Chip Mote

This work fits into the broader context of a project called the Single Chip Mote (SCM). The overall goal of this project is to create a low power wireless node that operates with zero external components. This problem essentially distills down to the questions of how to design a radio without a frequency reference, how to integrate an antenna, and how to power the system. The decision was made for this generation of hardware to focus on inter-operation with existing hardware which typically operates in lower frequency bands such as 2.4 GHz. At these frequencies external antennas are still required for reasonable performance. Thus this generation of the Single Chip Mote is intended to operate with two external components, a battery and an antenna. While current hardware is intended to operate from a battery, design consideration is given to minimizing power to enable future operation from scavenged energy sources. Future iterations of the project are likely to address antenna integration by moving to higher frequencies where on-chip antennas become feasible. Note that throughout this manuscript the term crystal-free is used in a manner as to apply to any off-chip frequency reference, whether that be an actual quartz crystal or any other physical resonator such as MEMS.

This project is somewhat unique in an academic setting in that it is very much a system level project aimed at designing a SoC. Not only is the goal to explore the design space of crystal-free radio, but perhaps even more-so, the goal is to produce a mote that can be used in other research projects. The ability to add wireless connectivity where it is previously impossible due to the mass and power requirements of off the shelf hardware enables a significant number of new research possibilities. To qualify as usable, the chip needs to truly operate in a bits-in bits-out fashion with no external components beyond the battery and antenna. Design aspects that might typically be relegated off-chip (like biasing, LDOs, digital implementation on FPGAs, etc) in the interest of increasing the academic rate of publishing cannot be ignored in this case. Thus the project requires a balancing of resources between designing for leading-edge performance and the more mundane aspects of raw functionality. There is simply not the time or personnel in an academic setting to optimize every sub-component. These unique design requirements also demand the contribution of many students, of both graduate and undergraduate experience levels. The Single Chip Mote often demands as much focus on project and personnel management as it does on circuit design.

1.2 Choosing a Wireless Standard

There are many architectural possibilities when it comes to designing a low power, crystal-free transceiver. For the purposes of demonstrating academic proof of concept one could simply build a low-complexity proprietary transceiver ASIC, place the non-critical parts off-chip on a PCB, conduct error rate measurements, and stop there. While this approach is likely the fastest and most flexible way to explore the fundamental problem, it is the farthest from useful in terms of having a Single Chip Mote for further research work. This option does have the advantage of allowing one to make more optimal architectural choices such as lowering energy/bit or mitigating the impact of performance degradation due to lack of a frequency reference. A major downside to a proprietary communication scheme is that you can only communicate among your own devices without access to existing networks. Deploying transceivers in real-world environments presents many challenges that extend far beyond architectural choices such as carrier frequency and modulation scheme. Managing medium access, networking, and reliable packet delivery are just a few of the many practical problems in creating a usable network of wireless nodes. Rather than try to implement these tasks from scratch it makes considerably more sense to leverage what existing hardware and software has already accomplished in this domain. These considerations lead to a review of current wireless standards for one that is a good target for implementing a network containing many low power crystal-free nodes. While targeting an existing standard simplifies many of the choices in a design, it also imposes specifications that may not be well suited for a crystal-free mote. To facilitate the success of the overall project an emphasis is thus placed on being compatible with existing hardware with less concern on being compliant with every aspect of a given standard.

There are many wireless standards available to choose from due to the rapid expansion of consumer electronics. They can be coarsely divided into the two categories of Internet of Things devices and personal electronics, with some standards arguably encompassing both. Wi-Fi and the 3G/4G/5G cellular standards such as LTE are designed for high performance in a device that has many times over the volume and battery capacity of a Single Chip Mote. While the ubiquity of devices with these types of radios would be a boon to any Single Chip Mote deployment, the advanced, spectrally efficient modulations are ill suited for crystal-free implementation. The IoT class of devices generally have simpler radios and are more focused on lowering power consumption. Examples of popular standards in this class are Bluetooth Low Energy and IEEE 802.15.4. It should be noted that IEEE 802.15.4 here refers to the PHY layer specification and not the sometimes more familiar Zigbee MAC. Both standards employ simple FSK type modulations and are relatively similar in their overall implementation.

While the hardware implementations of BLE and 802.15.4 transceivers can both be of relatively low complexity and low power, there are still other aspects of the communication system to consider. Simply totaling the page count of the relevant standards declarations for BLE (2822 pages) and IEEE 802.15.4(e) (539 pages) gives an idea of the relative complexity involved in implementing full systems. Not only is it advantageous for the Single Chip Mote project to leverage existing hardware standards, but perhaps even more so it should leverage existing software utilizing these types of radios. Prior to the development of the Single Chip Mote, over a decade of work in this research group has focused on software development tackling the hard problems that exist in wireless sensor networking. One result of that work is a project called OpenWSN [1] which is a standards based, open source software platform that provides IPv6 connectivity to wireless nodes running the IEEE 802.15.4 PHY. Plugging into this existing ecosystem provides an opportunity to very rapidly progress to usable networks once Single Chip Mote hardware is developed. Interoperation with existing hardware platforms that support OpenWSN also provides an opportunity to leverage the capabilities of commercial hardware to bolster the performance of a Single Chip Mote in a network. Ultimately this led to the decision that the main focus of this work would be on implementing a Single Chip Mote that is compatible with off the shelf IEEE 802.15.4 transceivers in the 2.4 GHz ISM band running OpenWSN.

1.3 OpenWSN and Time Synchronized Channel Hopping

Two common approaches to medium access in low power wireless networks are wake-up based and time synchronized scheduling (i.e. TDMA). In the general wake-up based approach a low power receiver listens for a signal to tell the device to turn on its main receiver and process an incoming transmission. The goal is to reduce the latency of packet exchange by being free to immediately transmit data as desired instead of waiting for a scheduled opportunity to do so.

The wake-up approach has several drawbacks: 1) In order to consume very little power since it is always on, the wake-up receiver sensitivity is generally much poorer than the main receiver. This effectively limits the overall sensitivity of the device as it will not turn on its main receiver if it didn't hear the wake-up message. 2) If the wake-up receiver is listening on only one channel then it is subject to interference and multipath effects [2]. 3) Without any mechanism to prevent transmissions from occurring at the same time, collisions and contention for medium access become problematic for high density networks.

A Time Synchronized Channel Hopping (TSCH) network attempts to overcome these issues by scheduling communication events. Since the receiver knows when to expect an incoming packet it only needs to be turned on in the proximity of the expected reception and can be left off otherwise. This leads to a very low duty cycle of operation allowing a higher power and performance receiver to be used for better sensitivity. The low duty cycle scheduled approach also allows for the use of frequency diversity to overcome persistent interference and channel fading. Sequential communication events between two nodes can be scheduled to occur on different channels which has been shown to mitigate these issues [3]. Scheduling also reduces medium access contention between devices that are in the same network and thus cooperating on the same schedule. This reduces the amount of potential collisions to those from other devices in the area on other networks. An often cited downside to the scheduled approach is that the worst case latency bounds can be rather large due to needing to wait for scheduled transmission slots before re-transmitting dropped packets. There are however ways to address these latency issues through diversity and scheduling when the network is built with bounded latency in mind [4] [5].

OpenWSN is an open source implementation of a full, standards-based protocol stack built on IEEE 802.15.4(e) using TSCH medium access which provides IPv6 connectivity to the Internet. It was designed to support multiple hardware platforms and is thus portable to many implementations of 802.15.4 hardware, such as the Single Chip Mote. Given that OpenWSN originated in the same research group it is an obvious strategy to leverage it as the software stack for Single Chip Mote. The implication being that when a working hardware prototype is created, it is a relatively quick process to port the software stack and achieve global IPv6 connectivity on a crystal-free node. The diverse hardware supported by OpenWSN creates the opportunity to intermix nodes of various types and capabilities. For example a handful of COTS nodes could be added to a network of Single Chip Motes to bolster the performance. SCM radios communicating with COTS radios benefit from larger link budgets due to the higher transmit power and better sensitivity of high performance commercial radios.

There is another advantage to TSCH when it comes to adding crystal-free transceivers to an existing network. Due to the scheduled nature of the network, there is a shared sense of time inherent across all nodes in the network. Feedback about discrepancies in expected packet arrival times can be used to correct for clock drifts [6]. Crystal oscillator based wireless nodes already exploit this network feedback to achieve very tight time synchronization of $< 1\mu s$ [7]. This network-level sense of time can be transferred to a crystal-free mote once it has managed to join the network, either through some initial calibration or search process. Once in the network, the crystal-free nodes sense of time can be compared against the occurrence of scheduled events and used to calibrate free-running oscillators [8].

1.4 IEEE 802.15.4

OpenWSN is built on top of the IEEE 802.15.4 PHY specification [9] and the IEEE 802.15.4(e) amendment that specifies TSCH aspects of the MAC. The standard is aimed at enabling low power communication which is reflected in its relatively lax specifications compared to higher performance standards. The 2.4 GHz OQPSK-HSS PHY is used which provides a 250 kbps user data rate. Direct Sequence Spread Spectrum (DSSS) with an 8:1

Specification	Value	Tolerance
Frequency	2.4GHz - 2.485 GHz	
Channel Spacing	5 MHz	±40*ppm*
User Data Rate	250 kbps	±40*ppm*
Chip Rate	2 MHz	±40*ppm*
Modulation	OQPSK-HSS	
Sensitivity	-85 dBm @ 1% PER	
Interference	0dB/5MHz +30dB/10MHz	
Max Input Power	-20dBm	

Table 1.1: Receiver related specifications for IEEE 802.15.4.

spreading is used to provide robustness and improve sensitivity. The relevant specifications are shown in Table 1.1.

The constant envelope nature of the signal allows for the use of a nonlinear PA and simplifies the receive chain. A further simplification of both the transmitter and receiver can be made by recognizing that Offset-QPSK with half-sine shaped baseband data is mathematically equivalent to Minimum Shift Key (MSK) modulation. MSK is a special class of frequency shift key where the data rate is equal to one half of the tone separation. A simple calculation is used to convert an OQPSK-HSS bitstream into its equivalent MSK format as given in [10]. For even indexed chip intervals Equation 1.1 is used whereas an inversion is added for odd chip intervals as in Equation 1.2.

$$k_{even} = I \oplus Q \tag{1.1}$$

$$k_{odd} = not(I \oplus Q) \tag{1.2}$$

An example of the conversion and corresponding waveforms is shown in Figure 1.1. This conversion is advantageous in that it allows a simpler direct modulation FSK transmitter to be used rather than relying on up-conversion of IQ baseband data. In the receiver this also enables the trade-off of allowing a simpler FSK demodulator to be used, albeit at the expense of reduced performance.

Interference tolerance in IEEE 802.15.4 is specified as 0 dB at the adjacent and 30 dB at the alternate channels (1 and 2 channels away respectively, 5 MHz channel spacing) with the desired signal at 3 dB above minimum sensitivity (i.e. -82 dBm). This is illustrated in Figure 1.2 which depicts an equal power modulated interferer at a 5 MHz offset and a 1000x larger interferer at a 10 MHz offset. In both cases the packet error rate (PER) should not exceed 1% which is also the benchmark for the minimum sensitivity test.

The OQPSK-HSS 2.4 GHz PHY employs Direct Sequence Spread Spectrum (DSSS) coding to improve sensitivity and robustness to interference. Bits to be transmitted are grouped into 4-bit symbols which are then mapped to 32-bit chip sequences. In receive the opposite process occurs using correlators to identify the best symbol match for each 32-bit

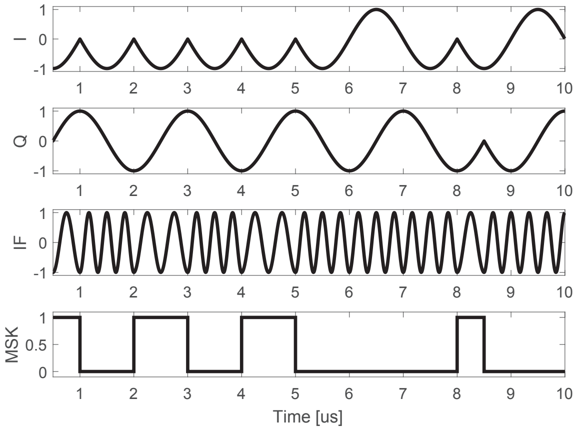


Figure 1.1: Equivalence between half-sine shaped IQ data (top) and Minimum Shift Key (bottom).

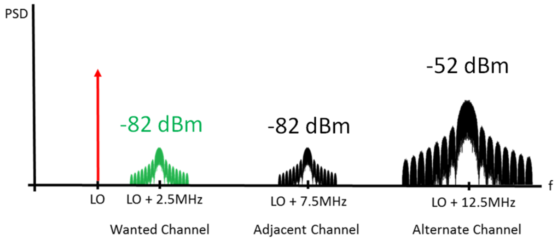


Figure 1.2: Depiction of interference specifications for IEEE 802.15.4 in a low-IF receiver.

Preamble (8 symbols)	SFD (2 symbols)	Length (2 symbols)	Payload (≤250 symbols)	CRC (4 symbols)

Figure 1.3: IEEE 802.15.4 packet structure.

chip sequence received. It should be noted that the OQPSK to MSK translation needs to be applied to both the transmitted chip sequence as well as to the correlation sequences used in the receiver. The translation process results in the receiver correlations only needing to be done with 31 bits since each 32nd bit becomes dependent on the symbol that follows it.

The packet structure of a IEEE 802.15.4 PHY later packet is shown in Figure 1.3. A preamble consisting of 8 '0' symbols is translated after spreading to a 256 chip long sequence with a duration of 128 μs. The preamble is followed by a one byte Start Frame Delimiter symbol with the value 0x7A. The length field is one byte long with the MSB as a reserved bit. The payload follows the length field and can be a maximum of 125 bytes. At the end of the packet a two byte CRC is appended. Note that the length field includes the two bytes of the CRC and thus has a maximum value of 127. Due to the detailed steps involved in packet generation (especially the CRC) it is recommended to consult a source such as [11] when attempting to generate valid packets for test purposes.

1.5 Crystal-Free Challenges

When attempting to implement a crystal-free 802.15.4 transceiver there are several factors to consider: 1) The RF local oscillator's (LO) ability to tune to a channel 2) The RF LO's impact on modulation accuracy 3) The accuracy and variation of the data rate 4) The accuracy and variation of the scheduling timer.

Without a good frequency reference like a crystal, the free-running oscillators that are responsible for these functions will drift due to noise as well as environmental factors such as temperature. As alluded to in Section 1.4 it is possible to perform periodic corrections on these oscillators using timing information inherent to the TSCH network, but their free-running performance impact still needs to be assessed. The remainder of this section serves to highlight the issues that will be of primary concern when designing a crystal-free radio and that will be discussed throughout this work.

RF LO

For the LO, the primary concerns are dealing with signal degradation from phase noise, tuning to the RF channel center frequency, and modulation accuracy. Many of these parameters also vary with temperature which further complicates the design by demanding some temperature compensation mechanism. Even after employing a calibration scheme there will always be some residual error in the RF channel center frequency. While the 802.15.4 standard specifies this allowable error as ±40 ppm, COTS receivers were found to implement a tolerance greater than this as seen in Section 1.6. The receive chain and demodulator in the Single Chip Mote should take this error source into account and be tolerant of it as well. Regardless of any periodic corrections to its center frequency, a free-running RF LO is also going to have significantly higher phase noise than a PLL-based synthesizer in a typical transceiver. The system-level impact of this increased phase noise is of significance interest when designing a crystal-free radio and discussed in depth in Section 2.2. The modulation accuracy of the transmitter is not only affected by phase noise, but also by the choice to implement the modulation as Minimum Shift Key. The tone spacing used in the modulation will have some error which will degrade the error vector magnitude of the modulated signal. The effects of this tone spacing error on the receiver are similar to those of phase noise and RF channel accuracy and can thus be modeling in a similar manner.

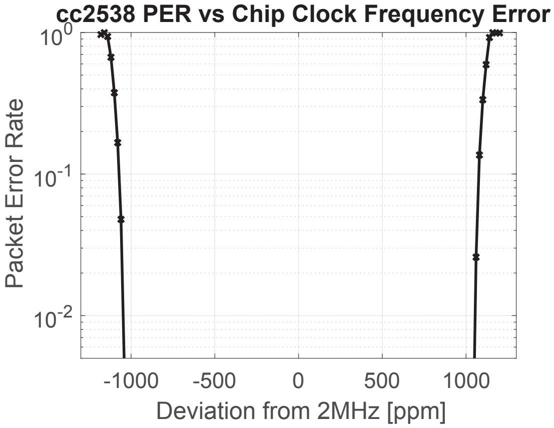


Figure 1.4: Error rate vs chip clock center frequency inaccuracy for the cc2538.

Chip Clock

IEEE 802.15.4 specifies a chip rate of 2 MHz with a tolerance of ±40 ppm, which dictates a high quality source for this clock. In the absence of a crystal the highest quality on-chip clock available is the RF LO. While a divider could be used to generate this chip clock from the RF source, it is not a desirable situation from a power perspective. Furthermore the RF LO suffers from temperature effects that require calibration before an accurate 2 MHz clock could be derived. It was found that commercial receiver's tolerance on this clock rate are well in excess of the value specified by the standard. This enables the Single Chip Mote to use a lower power, lower quality clock source and still communicate with other 802.15.4 hardware. Since the receiver clock recovery for SCM is under our control, chip clock rate tolerance is a design parameter that should be considered.

Sampling Clock

While in the transmitter the chip clock exists as its own independent oscillator, in the receiver the chip clock is derived from the ADC sampling clock. A typical receiver oversamples the incoming data stream at a rate which is an integer multiple of the data rate. This gives the receiver multiple samples per symbol with which to perform functions like demodulation, clock recovery, etc. If the sampling clock of the receiver is inaccurate or varying, then so is the chip clock. Whether the inaccuracy originates in the transmit or receive chip clock is indistinguishable in terms of performance. The receiver clock and data recovery must be designed with tolerance of chip clock inaccuracies in mind.

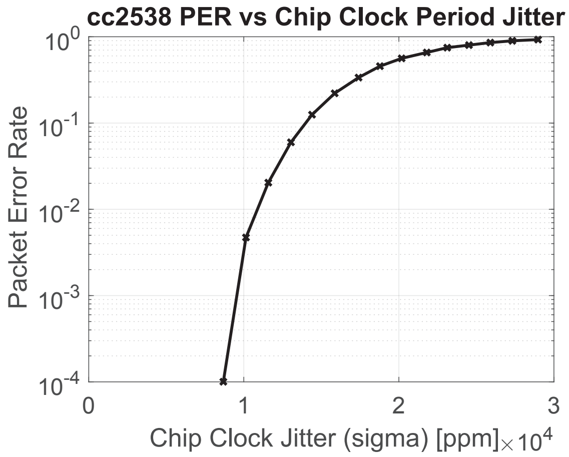


Figure 1.5: Error rate vs chip clock jitter for the cc2538.

1.6 COTS Hardware

As previously mentioned there are a variety of hardware options for IEEE 802.15.4 that OpenWSN already supports. The TI CC2538 is used in the OpenMote [12] and is thus a good candidate for inter-operation with the Single Chip Mote. It is of interest to investigate the response of this SoC to the types of errors introduced by a crystal-free system in order to determine the limits of crystal-free operation. The datasheet [13] for this part specifies that it will tolerate an RF channel error of ±150 ppm and a chip clock error of ±1000 ppm. Since the chipping clock is of particular interest, measurements were taken to confirm these values as well investigate the tolerance to jitter on the chipping clock are shown in Figures 1.4 and 1.5.

1.7 SCM System Overview

In general this work aims to implement a receiver that meets the relatively modest standards set forth by IEEE 802.15.4, do so without using a crystal reference, and to consume as little power as possible. The required sensitivity of -85 dBm allows for a relatively high noise figure which reduces the power requirements of the receiver. Combined with a low power transmitter output power of -10 dBm and assuming a few dB of antenna loss on both ends still results in a link margin in excess of 70 dB. While range is very dependent on the antennas used and the propagation environment, with this link budget one could expect 1-10 m indoor range and 10s of meters outdoors. If the RF link is between a Single Chip Mote and a commercial IEEE 802.15.4 transceiver, then an additional 10-15 dB of link margin is gained by the higher output power and better sensitivity of the high performance COTS device. The interference tolerance of the mote is another area where power can be saved at the expense of performance. Linearity costs power, and the system level approach taken here is to allow the network to re-transmit packets that are dropped due to interference rather than spending considerable amounts of power to reduce the chances of problems due to interference.

The Single Chip Mote is built around an ARM Cortex-M0 microprocessor with 128 kB of SRAM equally split between data and instruction memory. The instruction memory is programmed via an on-chip optical receiver as discussed in Appendix B. Dedicated hardware FSMs were developed to support IEEE 802.15.4 transmit/receive actions independent of processor control. A bank of dedicated TSCH timer interrupts trigger the radio actions and DMA provides direct movement of packets to and from SRAM. A detailed guide to the digital system is given in [14]. Integrated power management includes multiple LDO voltage regulators referenced to a programmable fractional bandgap. A variety of free-running clock sources are integrated to serve the functions of RF LO, chipping, sampling, and system clocks. A flexible bank of counters and dividers are also included to accommodate various calibration schemes between these oscillators. The remainder of this work focuses on the receiver implementation, which encompasses everything from EM waves reaching the antenna to packets being delivered to the application layer. The design of the frontend, down-conversion, filtering, and digitization are covered in Chapter 3 while the digital baseband is discussed in Chapter 4. Chapter 5 presents measurement results for the overall receiver.

Chapter 2

Crystal-Free Receiver Design and Modeling

Even after narrowing down to focus on an IEEE 802.15.4 transceiver there is still a tremendous amount of flexibility in implementation. The design is even further confounded by considering the complications added by crystal-free operation.

The design considerations here will be subdivided into two circuit domains: analog (how to get from RF to bits) and digital (turning those bits into packets). There is a considerable amount of flexibility in where to draw the boundary between these analog and digital portions. Process scaling has made it attractive to push as much signal processing as possible into the digital domain at the expense of increasing the performance demands on the ADC. This work targets a 65 nm process in which digital processing operating in the 10s of MHz range can consume a significant fraction of the overall system power. Therefore a very complex digital baseband approach is not likely to yield the lowest power system solution. It is difficult to analyze this trade-off between analog and digital in advance since detailed power estimates for digital blocks generally require them to have already been written and synthesized. The approach taken here is to balance the design between analog and digital by placing signal down-conversion and the bulk of filtering in the analog domain, with the digital domain containing further filtering, demodulation, and clock and data recovery (CDR).

The remainder of this chapter focuses on the early stages of a design where there are primarily two tasks to complete: 1) Make architectural decisions that best support the system level goals required by the receiver. 2) Model the receiver performance in the face of these decisions in order to set circuit level specifications.

2.1 Architectural Decisions

A receiver implementation can be broadly broken down into three steps: 1) down-convert the signal from the RF carrier used for transmission 2) perform signal filtering and digitization and 3) recover the payload information that was encoded in the data stream.

These steps involve a series of design choices which may seem rather arbitrary at the time they are made but are inextricably linked in the overall system. The goal is to make the best educated choices possible in a timely manner, then re-evaluate them later as the design complexity increases and the trade-offs are better understood. This approach helps to avoid slowdowns caused when the number of seemingly free parameters makes the design space appear paralyzingly complex. There will never be enough time to optimize every last parameter, and the functionality of the entire Single Chip Mote system will be prioritized over block level performance.

Down-Conversion

Phase Coherence

The first choice to be considered is whether to do coherent or non-coherent down-conversion. For an FSK demodulator there is approximately a 2.5 dB theoretical difference in the required E_b/N_o for demodulation between the two. Obtaining this extra performance however requires a carrier tracking loop to estimate the phase of the incoming signal which adds extra design time, complexity, power, and points of failure. The use of free-running oscillators with moderate amounts of phase noise will also degrade the ability of a tracking loop to estimate the carrier phase and thus the entire performance gain is unlikely to be realized in a crystal-free radio. Given the costs to implement what is likely to be a small benefit a non-coherent architecture was selected.

Local Oscillator

There are two categories for fully on-chip free-running oscillators which can be used as the RF LO (local oscillator). The first is resonant oscillators which use an LC tank to produce an oscillation at a specific frequency. The second is ring oscillators which are constructed from delay elements placed in a loop. It is shown in [15] that for a ring oscillator to achieve the same phase noise performance as an LC tank it requires more than Q^2 times the current, which clearly makes the LC preferable from a low power standpoint. There is still some argument to be made in favor of the ring oscillator if area is of substantial concern. The size of the inductor in the LC tank for a 2.4 GHz LO is in the nH range and will measure 100s of μm on a side, which does not change with process scaling. While the ring oscillator itself can be much smaller, its poorer performance and sensitivity require a substantial amount of circuitry for supply conditioning and tuning which will reduce the area benefit. Measurement results on free-running oscillators also show frequency stability issues that appear related to flicker noise which are substantially worse than an LC tank [16].

Given these considerations an LC tank oscillator is chosen as the LO for the Single Chip Mote, and the circuit level design is discussed extensively in [17]. This work is primarily only concerned with the LO insofar as its phase noise affects receiver performance. Circuit simulators provide performance data related to phase noise on a given LO implementation, but are too slow to evaluate entire systems over the span of many packets which is required for statistical performance verification. System level behavioral modeling approaches are a much faster way to perform this verification, but may not support time domain behavior of a noisy oscillator. What is needed is a way to bridge the gap and implement behavioral system level models with the phase noise behavior predicted by circuit level simulations. This approach will be discussed in further detail in Section 2.2.

Direct Conversion vs Low IF

After having decided on a non-coherent receiver using an LC tank LO the next decision involves the frequency plan for down-conversion. Direct conversion receivers use an LO that is at the same frequency as the RF carrier and shift the modulated signal down to baseband for further processing. While this eliminates the image problem and its corresponding filter requirements found in heterodyne receivers, it introduces a few other problems. Since the signal is now centered at DC, any offsets within the receive chain must be reduced or eliminated without affecting the integrity of the desired modulation. Any DC tracking loop introduced for this purpose is also going to require additional power and design time. Flicker noise from signal processing circuits may also be significant in the frequency region occupied by the baseband signal.

An alternative is the low intermediate frequency (IF) topology where the signal is shifted down to a carrier of a few MHz. By choosing the IF above the flicker corner both noise and DC offset can be removed without affecting the signal. This approach comes at the cost of increasing the power consumption of the signal processing chain as it needs to operate at higher frequency than the baseband data rate. The rejection of image band frequencies also becomes a concern.

In either case LO pulling can be a concern for high input powers. IEEE 802.15.4 specifies the maximum signal input power as -20 dBm, which after RF gain can cause significant pulling or even injection locking of the LO. One possible solution is to implement the LO at twice the desired channel frequency and divide it by two. This avoids the pulling issue but the divider logic can consume significant power, especially in a 65 nm process. Another problem that affects either architecture choice is frequency offset from the desired RF channel center frequency. This error can be lessened through system level approaches as discussed in Section 5.13, but will never be eliminated and should thus be considered when evaluating demodulator performance. To target low power and low complexity it was decided to utilize a low IF architecture using an LO at 2.4 GHz for the Single Chip Mote.

Filtering

The two main tasks required of the receive filter are to limit the noise bandwidth of the signal and to remove interference. Disregarding interference for a moment, the choice of what bandwidth to use for the filter involves trade-offs between signal power, noise power,

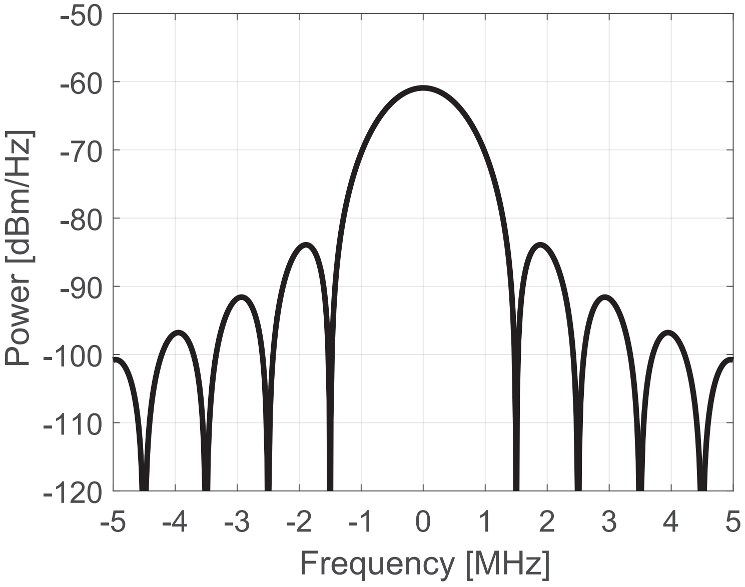


Figure 2.1: Spectrum of an MSK modulated signal with 2 MHz data rate.

and inter-symbol interference. The spectrum of an MSK modulated signal is shown in Figure 2.1. While the null-to-null bandwidth is 1.5 times the data rate, 99% of the signal power is contained in a bandwidth of 1.2 times the data rate. A wider filter bandwidth captures more of the signal power but also lets in more noise power. Too narrow of a bandwidth will not only reduce both the signal and noise power, but will also begin to cause inter-symbol interference. When the overall system performance is considered the optimum bandwidth is dependent on many factors, and for initial design considerations is chosen here to be equal to the data rate of 2 MHz.

The steepness of the filter roll-off is not especially critical for noise considerations since as filters increase above third order there are only relatively small changes in equivalent noise bandwidth. The roll-off is however of critical concern when it comes to removing unwanted interference signals at various frequency offsets. IEEE 802.15.4 does not specify an especially stringent set of interference requirements as outlined in Section 1.4. To reiterate those requirements here, a $<$ 1% packet error rate should be maintained for a -82 dBm desired signal while 1) a -82 dBm modulated interferer is present 5 MHz away on the adjacent channel and 2) a -52 dBm modulated interferer is present 10 MHz away on the alternate channel. A system level model can be used in order to validate that these conditions are met for a given filter within the overall system. The model used for evaluating these filter requirements in discussed in Section 2.3 and the actual circuit implementation is discussed in Chapter 3.

Image Rejection

In addition to the filtering requirements imposed by the standard there are additional requirements introduced by the choice of a low IF architecture. While the choice of an IF in the low MHz range is a reasonable trade-off between avoiding flicker noise and limiting bandwidth requirements in the filter, it potentially causes the image band to lie in the adjacent or alternate channels. IEEE 802.15.4 specifies no image rejection requirements, so the standard adjacent/alternate channel rejection values will be applied here. The image rejection can be accomplished in either the analog domain using image reject mixers or digitally using complex bandpass filters. In either case the amount of rejection that can be achieved is limited by the amplitude and quadrature phase accuracy of the local oscillator signals. The extent to which the rejection is compromised can be analytically described for the analog rejection approach but other factors come into play with a digital complex bandpass filter. The factors are those that influence the nominal stop-band rejection at the image frequency and include the number of taps in the digital filter and quantization effects. The digital complex bandpass approach is adopted for the Single Chip Mote and is discussed further in Chapter 4.

There are several ways that the required quadrature LO signals can be generated, each with its own advantages and disadvantages. One option is to use CMOS logic to create phase shifted LO signals with varying duty cycles. Depending on the process used this logic operating at RF can consume significant power. If a divide by two LO architecture is already being used then it can be combined with the quadrature generation. Quadrature signals can also be generated by locking two separate oscillators into a quadrature relationship. This requires doubling the power consumption by essentially having two LOs. A third option is to place a passive polyphase filter between the LO and mixer to generate the required phase shift. This is the approach taken for the Single Chip Mote and the RC implementation is discussed in [17]. There is some amplitude loss across the passive network so more current must be used in the LO to reclaim the lost swing, but this is still the most attractive solution in this 65 nm process node. Correcting for quadrature inaccuracies due to process mismatch in the passive polyphase is difficult due to the small component values used in the filter. Rather than attempting to implement correction approaches in the analog domain with components operating at RF, a digital approach can be used to correct for quadrature inaccuracy in the digital domain [18] and is discussed further in Chapter 4. It should also be noted that by filtering out the image channel a 3 dB improvement in SNR is obtained by effectively halving the amount of thermal noise within the receiver's bandwidth. This 3 dB increase in SNR does not come for free however as image rejection requires I and Q channels which doubles the power consumption. The same SNR improvement could be obtained by spending twice the power in the I channel alone, although image rejection is then lost completely.

Demodulators and Packet Acquisition

There are many MSK demodulation schemes of varying complexity and performance. Two of these methods are implemented in the Single Chip Mote and are briefly introduced here. A more detailed discussion on demodulators and digital baseband can be found in Chapter 4. The first demodulator is a very straightforward implementation of an FSK demodulator which will be referred to as the Zero Crossing Counter (ZCC). The ZCC attempts to discriminate between the two modulation tones by measuring the timing between successive zero crossings in the IF waveform. This can be done by sampling the IF waveform at a high rate and counting how many clock cycles occur between successive zero crossings. A threshold on the count value then yields an estimate of whether the IF was the fast or slow FSK tone as shown in Figure 2.2. A high sampling frequency is required to obtain the necessary resolution between count values for the two tones. It is also advantageous to use as low of an IF center frequency as possible since that increases the count ratio between the two tones given that their spacing is fixed at 1 MHz.

The ZCC benefits from only requiring a single bit input from a comparator rather than a full multi-bit ADC. Since this demodulator operates solely on a real-valued IF signal, the receiver can also save power by shutting down its quadrature receive path at the expense of image rejection. Additional filtering could be digitally implemented on the 1-bit comparator output prior to the ZCC demodulator, but this is not done in the Single Chip Mote to reduce complexity. The simplicity of the ZCC is not without its drawbacks. Since zero crossings are the method of demodulation the ZCC requires relative strict duty cycle control to avoid corrupting the FSK information. This translates into a strict offset limit on the comparator, which means an offset calibration loop is likely required. The ZCC performance is also inhibited by the location of zero crossings relatively to bit boundaries. With no carrier recovery mechanism in the receiver there is no guarantee that bit transitions occur at the zero crossings of the IF. When the modulation changes in the middle of a count value the demodulator is more likely to incur errors. With no way to know or correct either the transmitter or receiver LO phase then the average performance of the ZCC must be evaluated across a uniform random distribution of 2π for the LO phase.

The second demodulator implementation is based on matched filtering, which is known to provide the optimal demodulator for non-coherent FSK [19]. As shown in Fig 2.3, templates for the two FSK tones are correlated with the incoming signal and the magnitude of their output is compared. Whichever template has a higher correlation value is considered the better match and that data bit is then selected for output. Since carrier phase is not known, quadrature templates are used so that the phase of the incoming IF waveform is irrelevant. Ideally the two filter templates are orthogonal for the best selectivity, but this is not always possible, as is the case here since the 1 MHz tone separation is not a free variable to be modified.

Convolution with the matched filter templates is effectively implementing a digital bandpass filter centered around each of the modulation tones and thus leads to inherently better selectivity than the ZCC approach. The matched filter can also be operated with lower sam-

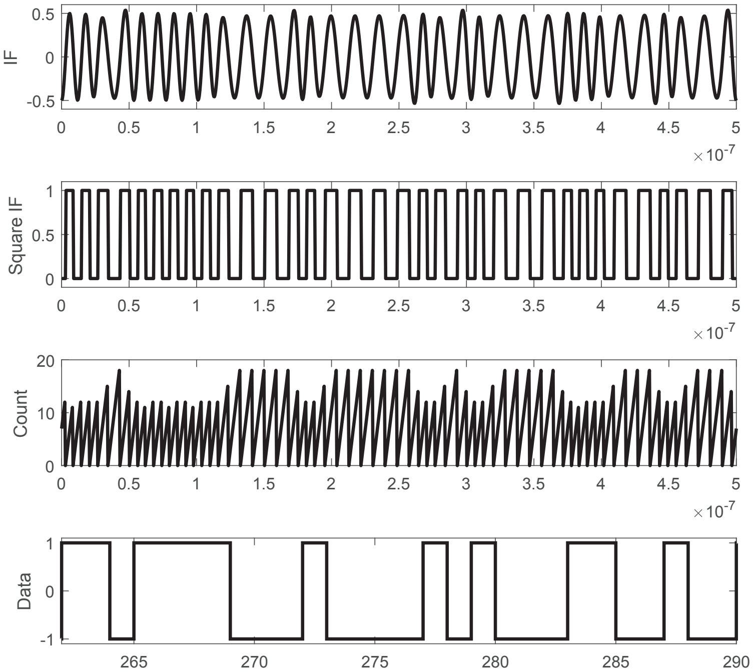


Figure 2.2: Transient waveform of zero crossing counter demodulator behavior. At every zero crossing of the IF waveform, both rising and falling, the counter is reset to zero. If the counter exceeds a threshold then the output bit is deemed a '0' otherwise if the count value remains below the threshold the output is '1'.

ple rates as there is no longer the requirement to differentiate between small time intervals. A drawback of the matched filter demodulator over the simpler ZCC is the requirement for a multi-bit input, which can complicate the design space by now requiring a more complex ADC. Clock recovery is another very important part of the receiver following the demodulator and will be discussed in depth in Chapter 4. For now CDR performing near its ideal limit will be used for system modeling. Further details on demodulator implementations are also discussed in Chapter 4.

Despreading

All steps following CDR utilize the work implemented in [14] and will be summarized here as this work is concerned with the entire data path from antenna to packet storage in memory. The process of converting received chip sequences back into symbols is done via a bank of correlators. Each set of 32 received chips is compared against the 16 possible sequences outlined in [9] and the best match is selected based on smallest Hamming distance. Note that since the transceiver is being implemented as MSK the values used in the correlators are the MSK transformed versions rather than the OQPSK ones defined in the standard. Single

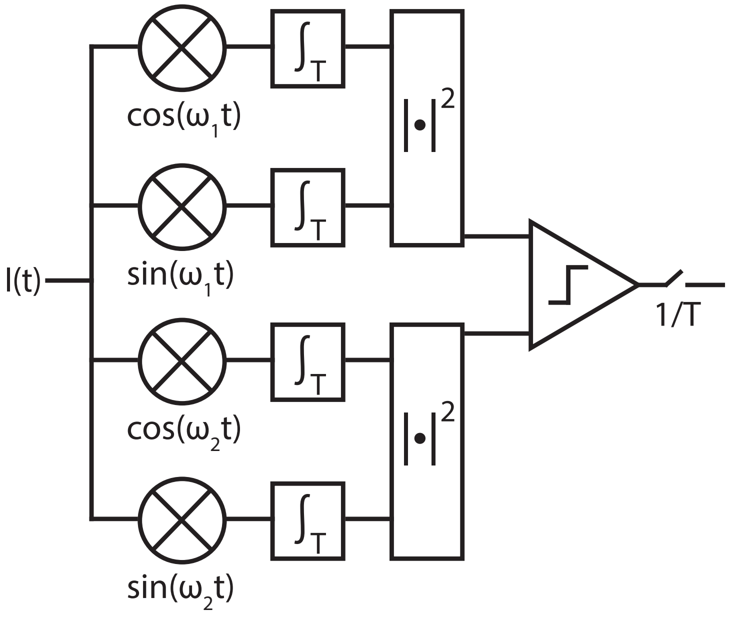


Figure 2.3: Block diagram of the matched filter demodulator for non-coherent FSK.

Chip Mote utilizes hard decisions to minimize the correlator complexity but implementing soft decisions using multi-bit output from the demodulator could improve the performance by approximately 2 dB [20]. There also exists an opportunity to exploit the cyclically shifted nature of the different chip sequences to simplify the correlator hardware implementation if desired [21].

Packet Acquisition

After bits have been recovered there is still the task of temporally locating the start of the packet in the presence of noise and interference. IEEE 802.15.4 contains both a preamble and a start frame delimiter (SFD) for this purpose as discussed in Section 1.4. Both [14] and [13] implement packet detection by first searching for some number of preambles followed by the SFD. An adjustable threshold is used for how many errors are acceptable in these correlations. There is a trade-off to be made here in how sensitive the packet detection mechanism is. If the receiver accepts matches with high error rates then it is less likely to miss a packet, but is more likely to falsely trigger on noise. If instead the receiver requires a very accurate match before it will begin storing packet data then it will miss many packets near its minimum detectable signal level. How long the receiver will be listening to random noise before the packet is expected to arrive also plays a role. If the time between when the receiver is turned on and when the packet actually arrives is very short, as say in a very tightly time synchronized TSCH network, then the receiver can afford to relax its threshold settings

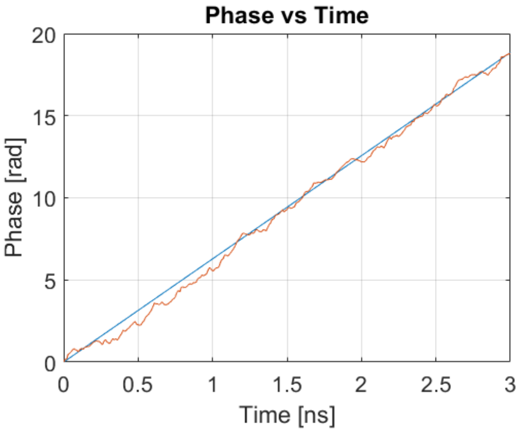


Figure 2.4: Exaggerated example of phase deviation due to noise.

since there are simply fewer opportunities for false triggering. Given this set of trade-offs, it is imperative that as much control over this detection process be given to the application software as possible to allow for optimizing the detection for the situation. Consideration should also be given to the mechanism for aligning the boundaries of the 32-bit long chip sequences that are fed into the correlators. Ideally the receiver is able to investigate possible alignment matches while simultaneously monitoring for better matches in order to reduce acquisition errors.

2.2 Time Domain Phase Noise Modeling

Understanding how the phase noise of an open loop oscillator affects transceiver performance is critical to the design of a crystal-free radio. The effect of phase noise on a receiver is heavily dependent on the communication protocol and the specifics of implementation. Therefore it is desirable to develop a behavioral model of the complete system in order to derive circuit level specifications. There are many ways to simulate transceiver performance, each with its own advantages and disadvantages. For this work MATLAB was chosen as the system level simulator due to its flexibility. In order to implement a time domain behavioral model assessing the impact of phase noise, it is necessary to understand how to generate discrete-time sequences which mimic the behavior of a free-running oscillator.

An ideal oscillator linearly accumulates phase over time whereas any real oscillator will have perturbations in its phase due to noise. An exaggerated depiction of this is shown in Figure 2.4. In the time domain this variation appears as a random variation in frequency known as jitter. In the frequency domain it results in a spreading of power across a range of frequencies resulting in what are often termed “skirts” when viewed on a spectrum analyzer. There is extensive literature already discussing the mechanisms and mathematics of phase noise [22] [15]. The goal of this section is not to provide an exhaustive tutorial on phase noise, but rather to provide an overview on generating time domain sequences that can be used to model the impact of phase noise on a receiver.

A generic oscillation with phase noise can be expressed as shown in Equation 2.1 where $\phi(t)$ represents a random fluctuation in phase. There can be multiple power law noise processes present in $\phi(t)$ although oscillators of interest predominately consist of white and flicker noise.

$$x(t) = cos(\omega t + \phi(t)) \tag{2.1}$$

A discrete time representation of $\phi(t)$ that accurately captures its spectral properties is required in order to generate a time series for simulation in MATLAB. The required discretization can be obtained from [23] [24] and is given in Equation 2.2. The frequency of oscillation is f_0, $w(n)$ is a random variable with Gaussian distribution, Δt is the simulation time step, and c is a scalar constant that represents the rate at which the phase variance grows due to white noise.

$$\phi_w(k) = 2\pi f_0 \sum_{n=0}^{k-1} w(n)\sqrt{c\Delta t} \tag{2.2}$$

It is shown in [22] that the power spectrum of an oscillator in the presence of only white phase noise is Lorentzian in shape and is described by Equation 2.3. This is the single sided power spectral density and f_m represents the frequency offset from the center frequency of oscillation f_0. It should be observed that a single scalar constant is sufficient to describe both the time domain properties of the generated $\phi(k)$ sequence as well as the resulting power spectrum of the oscillation.

$$\mathcal{L}(f_m) \approx 10 \cdot log_{10}(\frac{f_0^2 c}{\pi^2 f_0^4 c^2 + f_m^2}) \tag{2.3}$$

The constant c has further intuitive physical meaning as it also describes the cycle-to-cycle jitter of the oscillation as given in Equation 2.4. Since a low-IF receiver architecture is being designed it is also of interest to know how the cycle-to-cycle jitter of the RF oscillator will translate to jitter of the intermediate frequency after down-conversion by a mixer. This translation is given in [25] and is shown in Equation 2.5. This simple set of equations now provides a very useful link between circuit and system level simulations. Error rate simulations can provide specifications on the value of c which can be extracted from transistor-level simulations to assess a potential oscillator.

$$\sigma_{RF}^2 = \frac{c}{f_0} \tag{2.4}$$

$$\sigma_{IF}^2 = \sigma_{RF}^2 \cdot (\frac{f_0}{f_{IF}})^3 \tag{2.5}$$

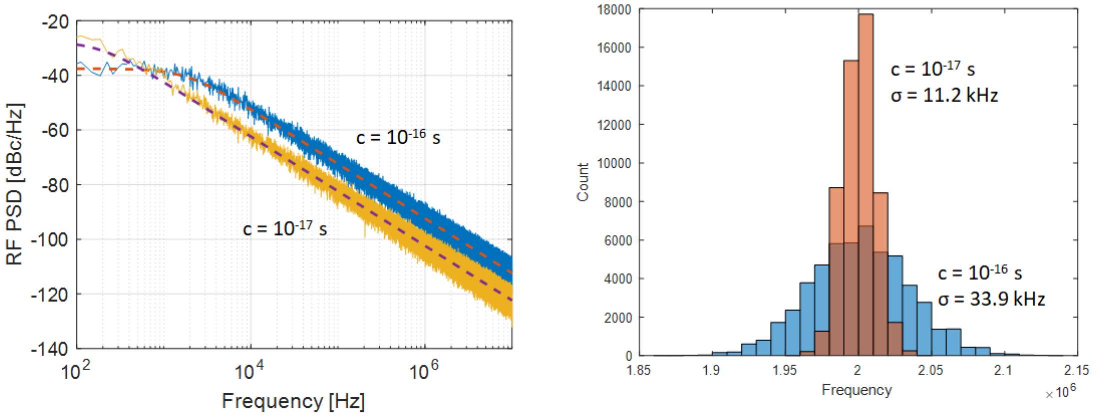


Figure 2.5: Left: PSD with white phase noise values of c=1e-17 and c=1e-16 s. Right: Histogram of variation at IF.

Figure 2.5 shows two example oscillators with different values of c and their corresponding distributions of cycle-to-cycle jitter after mixing down to an IF of 2 MHz. Note that in both cases the frequency variation due to the white phase noise is significantly smaller than the 1 MHz Δf tone spacing used by the IEEE 802.15.4 MSK modulation.

The above treatment of phase noise only considered the impact of white noise. Depending on the design of the oscillator and communication system it may also be important to assess the impact of flicker noise. This work will follow the approach used in [23] which is given in Equation 2.6. Another constant denoted c_f is used to describe the magnitude of the flicker noise component and $w_f(n)$ is another Gaussian random variable (distinct from the random variable used for white noise). In order to generate a sequence that has the appropriate spectral properties of flicker noise, $h(k)$ is generated based on a recursive fractional differencing approach as given in [26] and shown in Equation 2.7. This approach can be used for various power law noise processes by choosing the value of β, which for flicker noise is $\beta = -3$. It should be noted that the required convolution operation imposes a heavy computational burden for long time domain simulations. For simulations of extended duration it becomes imperative to compute this convolution using the FFT rather than doing so in the time domain.

$$\phi_f(k) = 2\pi f_0 \cdot \Delta t \sqrt{2\pi c_f} \sum_{n=0}^{k} (h_{k-n} \cdot w_f(n)) \tag{2.6}$$

$$h(k) = (k - 1 - \frac{\beta}{2}) \cdot \frac{h_{k-1}}{k} \quad \text{where} \quad h(0) = 1 \tag{2.7}$$

The overall equation for generating a discrete-time representation of an oscillator with both white and flicker phase noise is shown in Equation 2.8. The spectrum is no longer

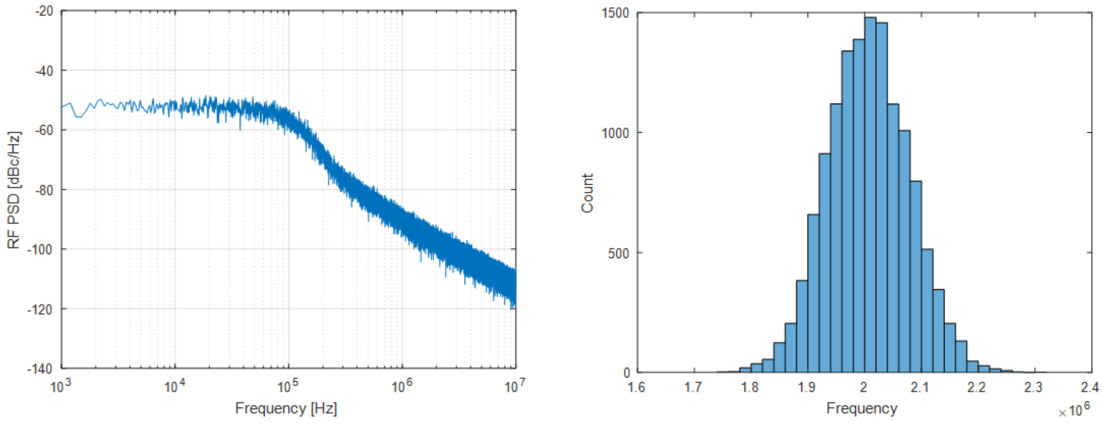


Figure 2.6: Left: PSD with white and flicker noise. Right: Histogram of variation at IF.

described by a Lorentzian profile, but it is still possible to extract values for c for c_f based on curve fitting for comparison to transistor-level oscillator simulations. This can be accomplished by first adjusting the value of c in Equation 2.8 so that the resulting phase noise at large offsets matches the simulated value (since the phase noise at large offsets will be dominated by white noise and c describes only the white noise component in the model). Once the phase noise at large offsets matches then the amount of flicker noise can be adjusted using c_f to match the shape of the phase noise curve at lower frequency offsets. The end result is that the discrete time sequence produced by Equation 2.8 with the two adjusted constants produces a waveform that has the same phase noise properties as that predicted by a circuit level simulation of the real oscillator. An example power spectral density and the cycle-to-cycle frequency variation after down-conversion to IF are shown in Figure 2.6. Depending on the time scale of observation relative to the level of flicker noise, the histogram of cycle-to-cycle frequency may no longer be Gaussian. The flicker noise essentially causes the mean frequency after down-conversion to wander. If this mean wander is slow compared to the duration of a packet, then the receiver will perceive it as a center frequency error with the cycle-to-cycle variation being predominately due to white noise. The MATLAB script for generating a time domain sequence of the noisy oscillator in Figure 2.6 is given in Appendix A.

$$\phi(k) = 2\pi f_0 \cdot (\sum_{n=0}^{k-1} w(n)\sqrt{c\Delta t} + \Delta t\sqrt{2\pi c_f}) \sum_{n=0}^{k} (h_{k-n} \cdot w_f(n)) \tag{2.8}$$

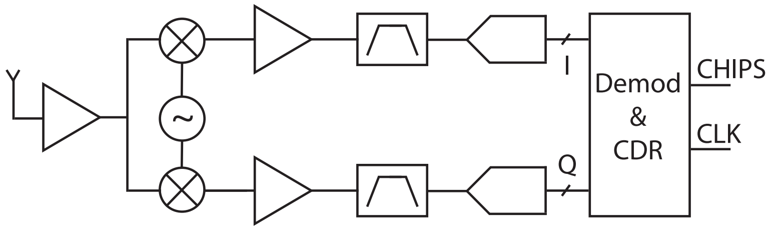


Figure 2.7: Block diagram of the main components of the received used to develop the initial performance model.

2.3 Receiver Design and Modeling

Based on the decisions outlined earlier in this chapter a block diagram of the receiver to be modeled is shown in Figure 2.7. The receiver will use a free-running quadrature LO to down-convert to a low IF. Gain, filtering, and digitization will be implemented on-chip and both types of demodulators introduced previously will be implemented. There are many interdependent parameters to be determined which affect the trade-off between the overall performance of the receiver and its power consumption. In order to reduce the complexity of the design space the error rate performance will first be assessed only in the presence of noise sources within the receiver. The design can then be extended to consider the effects of interference and how to meet the standard requirements.

At this stage it is difficult to set concrete design limits on the power consumption of the receiver without knowing more about the required circuit performance. However some information is available from previous hardware generations which can serve to provide the basis for a rough estimate. The LC local oscillator in [27] consumed 1 mW of power. Its performance appeared to be substantially better than required for implementing an 802.15.4 receiver so it was assumed to be possible to reduce this power. However that design did not include on-chip biasing and supply regulation so its performance is likely to change once those circuits are integrated. This work is not concerned with the circuit design details of the LO (see [17]) but at the time that the receiver design began it seemed reasonable to assume that the LO power could be reduced by at least a factor of two to 500 μW. It was also known from previous hardware measurements that the Cortex-M0 operating at 5 MHz consumed 250 μW. This power consumption did not include any of the digital baseband circuitry which now must be added (Chapter 4). Assuming that the digital baseband at most doubles the digital power consumption, and that the analog portion of the radio including both the LO and the rest of the receiver can be limited to less than 1 mW, the total power consumption in receive mode would be somewhere between 1 and 1.5 mW.

To assist in the design and verification of the receiver it is very useful to have a behavioral model that mimics the functionality of the real receiver and includes various non-idealities. There is a trade-off here between how much depth and detail is implemented in the model versus how much time investment it takes to implement the modeling. On one hand a very detailed simulation that takes into account the complete receiver and all known non-idealities increases the likelihood that problems will be caught before tape-out. On the other hand though one could spend forever modeling and verifying and never actually end up taping anything out. The designer must decide how to find the appropriate balance with the obvious goal being to reach working silicon as quickly as possible. There are many ways to go about modeling a transceiver: MATLAB, Verilog-A, as well as many commercial system simulator packages like Simulink or SystemVue. Each option has its advantages and disadvantages. System modeling packages allow a designer to implement rather complex models quickly using a building block approach, but this can remove the requirement that the designer actually understands what is going on inside these blocks. Implementing each of these blocks from the ground up in MATLAB forces the designer to gain an understanding of how each subsystem works. Verilog-A has the advantage of tight integration with the circuit simulator environment so that idealized model blocks can be replaced with real circuit implementations to assess their impact on the model performance. For this work MATLAB was chosen mostly for its flexibility and ability to implement blocks from the ground up, although Verilog-A also looks very attractive if this design were to be repeated from the beginning. Regardless of the simulation method chosen the process is going to be iterative as ideal blocks are replaced with more realistic representations and requirements are re-evaluated.

Choosing an Intermediate Frequency

There are many things to consider when choosing what IF to down-convert to. The matched filter templates are implemented as sine and cosine waves of the two MSK tones that the transmitter alternates between (see Section 4.1). Convolution of these templates with the incoming signal acts like a bandpass filter around each frequency tone. The lowest frequency of which a full cycle can be captured in the 500 ns chip duration is 2 MHz. The consequence being that templates for frequencies lower than this will implement a low pass filter rather than a bandpass which will have some impact on performance which must be modeled. The choice of IF also has an impact on the deviation caused by phase noise as seen in Equation 2.5, but as will be discussed shortly, the impact of phase noise overall is minimal so this is not of great concern. The power consumption of analog filters is dependent on the bandwidth they must achieve. By moving to a higher IF, the amplifiers in the filter will need to become faster and thus consume more power. Flicker noise in the analog portion of the receiver must be carefully managed if a very low IF is considered, although this could be mitigated by using large devices in layout or using techniques such as chopping and double correlated sampling. For the ZCC demod a low IF is again preferred as it maximizes the frequency ratio between the two FSK tones for a fixed tone spacing. The preceding constraints all point to a low IF possibly located in the single digit MHz range. Given this set of considerations an initial choice of an IF of 2.5 MHz was chosen for the system modeling going forward. This is the lowest choice for which the matched filter templates will still contain an entire cycle of the 2 MHz and 3 MHz tones that result from the choice of 2.5 MHz. This value could be revisited as the model is refined and circuit implementations are solidified, although it remained satisfactory throughout the SCM design phase. No detrimental side effects of this choice were discovered during modeling and no obvious better choice was found.

Receiver Noise Sources

The main source of thermal noise within the receiver is the circuit noise added by the RF frontend. Additional noise will also be added from other downstream circuitry and collectively the total amount of noise is captured in the design metric known as noise figure. The required receiver sensitivity of -85 dBm, the chosen filter bandwidth of 2 MHz, and the noise figure are related by Equation 2.9. The remaining unknown term is the minimum amount of SNR required at the demodulator input to achieve the specified 1% packet error rate. The SNR requirement can be derived from a system level model which will then allow a minimum noise figure to be calculated.

$$-174dBm + 10log10(2e6) + NF + SNR_{min} = -85dBm \tag{2.9}$$

In addition to amplifier noise, ADC quantization noise also contributes to in-channel degradation of the SNR. High performance receivers generally use a large enough number of bits that the quantization noise floor is much lower than the thermal noise floor. The number of bits chosen for the ADC presents an opportunity for trading off power and performance. Using more bits increases both the dynamic range and the achievable SNR but is also more costly in terms of power, both for the ADC itself and also the following digital baseband. The SNR requirements for a demodulator are generally quite modest in comparison to the SNR achievable by an ADC even with only a few bits. The required dynamic range depends on how much interference must be tolerated at the input to the ADC. An ADC with a large number of bits is able to digitize the desired signal alongside other larger signals and then filter out the interference digitally. Given the interference requirements for IEEE 802.15.4, it is assumed here that any unwanted signals present have been sufficiently filtered prior to the ADC so that their amplitude is less than or equal to the desired channel. This allows the dynamic range requirements to be relaxed and a smaller number of bits can be used for the ADC to save power.

ADC sampling rate affects the quantization noise density as well as interference folding. Oversampling is beneficial in both cases as it reduces the in-band noise density and relaxes the requirements for anti-aliasing prior to the ADC. The signal can always be down-sampled at a later stage to reduce power consumption if desired. The trade-offs between sample rate, the number of bits, and the achievable SNR considering only quantization noise are shown in Figure 2.8. This plot assumes that a digital channel select filter has been used to band-limit the noise to the same 2 MHz bandwidth as was chosen for the signal filter. It can be seen that SQNR in excess of 20 dB is easily achieved and thus quantization noise is not likely to be a substantially limiting factor even for ADCs with small numbers of bits.

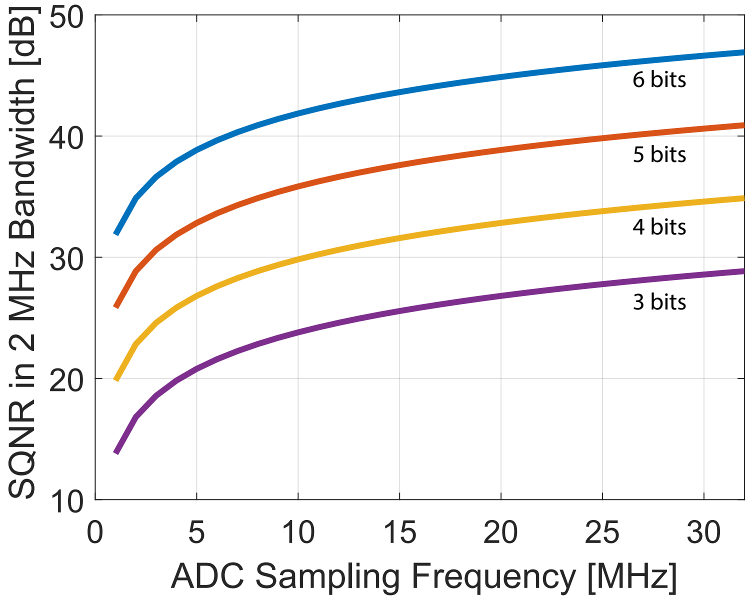


Figure 2.8: The achievable signal to quantization noise ratio for a given sampling frequency and number of bits when the signal is digitally band-limited to 2 MHz.

The matched filter demodulator and CDR require an integer number of samples per symbol in order to perform their respective functions. This translates to the requirement that the ADC clock rate is an integer multiple of the chip rate of 2 MHz. Furthermore the algorithm used for CDR, discussed in Chapter 4, requires a power of two for the number of samples per symbol. This effectively reduces the choices for ADC sampling rate to 8, 16, or 32 MHz. The choice is ultimately driven by a trade-off between power consumption and where interference aliases to after being sampled. Considering these trade-offs, a frequency of 16 MHz is used for the ADC clock in the system level modeling to follow. The source of the ADC clock should also be taken into account from both a tuning and jitter perspective. While the clock jitter is generally only a problem when targeting SNR values far in excess of those discussed here, the derivation of the chip clock from this source means that its tuning accuracy is quite critical. For initial modeling this clock is assumed to be ideal and these issues will be discussed further in Chapters 3 and 4.

The zero crossing counter demod requires a much higher sample rate due to its attempt to measure the time between zero crossings. A high speed clock is needed to ensure the difference between high and low FSK tones can be distinguished. Using too high of a clock speed is also undesirable from a power perspective as the comparator output bit stream must be processed by the digital implementation of the zero crossing demod. A first estimate about what clock rate to use is chosen to be bounded by 50 MHz on the low end and 100 MHz as the upper speed limit. A third intermediate value of 76 MHz is also added as another data point. Since the 2 MHz chip rate is recovered from the same clock that operates the demodulator, the clock rate must be a multiple of two and thus 75 MHz is rounded up to 76 MHz.

Phase noise manifests itself in the receiver as an additive noise source. Phase noise becomes dominant at high SNR values when the input signal is much higher power than the thermal noise added by the receiver. The frequency deviation caused by the LO phase noise is indistinguishable from the modulation and effectively places a floor on the achievable error rate where further increases in SNR no longer help. Using the phase noise model from Section 2.2 to curve fit to the measured phase noise data in [27] results in a white noise coefficient of $c = 1.06\text{e-}18$ s and a flicker coefficient of $c_f = 2\text{e-}13$ s. Using these coefficients to generate a time domain waveform with these phase noise characteristics and mixing it down to a low IF in MATLAB results in a 1σ frequency deviation of 5.5 kHz. If flicker noise is omitted from the model the deviation becomes 4 kHz. These predicted IF deviation values are much smaller than the 1 MHz MSK tone spacing and as previously mentioned, this 1 mW oscillator could easily meet the requirements of 802.15.4. It can be noted that the effect of flicker noise was not that significant and the majority of the phase noise in this LC LO is due to white noise. Including this low level of phase noise in a receiver model for this work would not be especially informative as the phase noise will have essentially no impact on performance.

Instead, in order to give the reader some intuition as to the impact of phase noise a set of simulations will also be included here with phase noise that is greatly exaggerated from the values expected in a realistic LC LO. This analysis will assume that white phase noise continues to dominate over flicker. To visualize the link between the white phase noise modeling coefficient c and the amount of frequency variation after down-conversion to a 2.5 MHz IF, Equation 2.5 is plotted in Figure 2.9 for a range of values. The exaggerated values chosen are $c = 0.4\text{e-}15$ which corresponds to a 1σ variation of about 75 kHz and $c = 1.2\text{e-}15$ which corresponds to a variation of about 130 kHz.

The following plots are generated from a MATLAB model used to evaluate the impact of the considerations discussed above on the error rate performance of the system. The error rates given here are chip error rates as there has been no de-spreading back to symbols or bits. The goal of this system level model is to

- Specify a limit on acceptable phase noise
- Evaluate how many ADC bits are required
- Compare demodulator performance (and find SNR_min)
- Set a limit on acceptable noise figure
- Assess filtering requirements

Before discussing the model results it is useful to establish the link between the chip error rate and packet error rate which can be found in [20]. The 802.15.4 standard specifies the

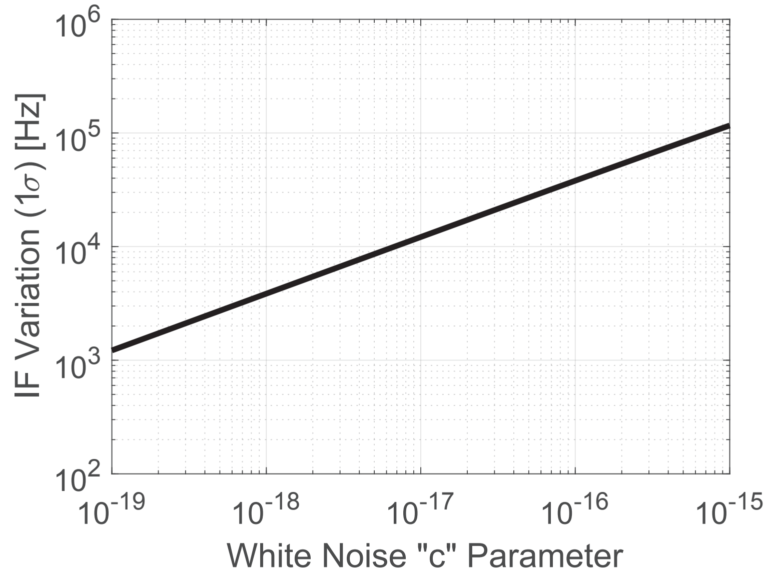


Figure 2.9: Cycle-to-cycle variation in a down-converted 2.5 MHz IF signal vs the amount of white phase noise in a 2.44 GHz LO.

sensitivity test should achieve < 1% packet error rate with 20 byte payloads which corresponds to 52 symbols per packet. Each symbol contains 32 chips and the spread spectrum code set used has a minimum distance of 5 and mean distance of 8 between code words. The probability of a symbol error can be calculated with Equation 2.10 for a given chip error rate. A symbol error occurs when the number of chip errors is high enough that the received data has a lower Hamming distance to an incorrect symbol than the symbol that was sent. The symbol error rate can be converted to a packet error rate by multiplying by 52 which is the number of symbols per test packet. The packet error rate vs chip error rate is plotted in Figure 2.10 and it can be seen that a 1% packet error rate requires approximately a 6.5% chip error rate.

$$SER = 1 - \sum_{e=0}^{8} \binom{32}{e} CER^{e}(1 - CER)^{(32-e)} \tag{2.10}$$

To allow fair comparison between demodulators, it is assumed that thermal noise has been band-limited to 2 MHz prior to the ADC. This band-limiting serves to mimic the channel select filter which has not yet been designed at this point. This results in optimistic values for SNR_min given that this filter has sharper roll-off than what can actually be achieved in the eventual analog filter implementation, but suffices for the explanation here. This is fine for initial comparison sake as the filter in the model will be updated as the design iteration occurs. No further digital filtering is performed after quantization in the matched filter case, which should be kept in mind as that will further improve the performance. This

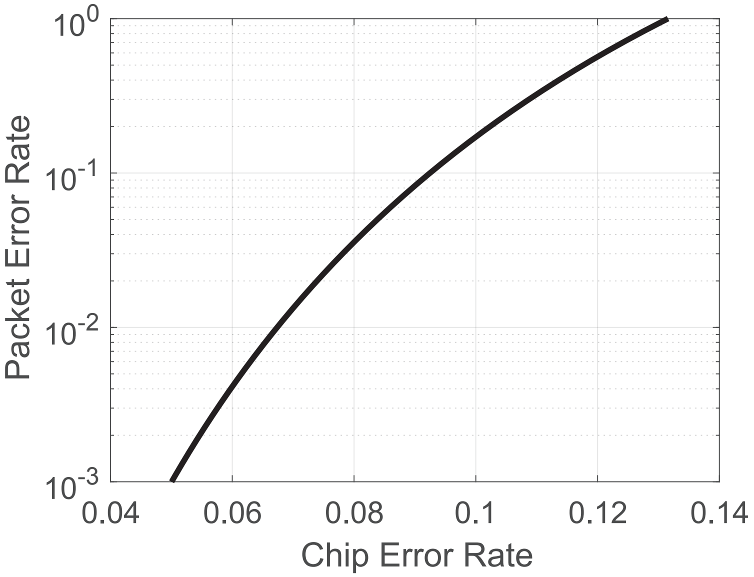


Figure 2.10: Relationship between chip error rate and packet error rate for the 802.15.4 DSSS coding.

improvement for the matched filter could be even further extended if image band noise is digitally filtered out. The ZCC does not have this benefit as it operates solely on the in-phase channel. The SNR axis is defined as the signal to noise ratio prior to the ADC input on either of the in-phase or quadrature channels. A realistic CDR implementation is used for the MF which performs very near its ideal limit, while the ZCC uses an ideal CDR for initial evaluation. Automatic gain control for the matched filter case will be discussed in Chapter 4 and for now is assumed to operate ideally such that the input amplitude is always scaled to match the full-scale ADC range. Payload lengths of 20 Bytes are used for all simulations.

Figure 2.11 shows the modeled performance of the ZCC demodulator operating at the intermediate sample rate of 76 MHz with and without phase noise. Recall that these values of phase noise are exaggerated from those expected in a typical LC LO but serve to demonstrate the effect of phase noise on the demodulator. The minimum required SNR for a 6.5% chip error rate (CER) with no phase noise is about 6 dB. The large amount of phase noise associated with c = 1.2 fs increases the SNR_min by about 1 dB and sets a chip error rate floor at high SNR of around 0.6%. As seen from Figure 2.10 a 0.6% chip error rate results in a packet error rate that is very low so even this highly pessimistic amount of phase noise has negligible effects on the receiver. Figure 2.12 shows the effect of changing the sample rate for the ZCC when a phase noise value of c = 0.4 fs is used. The simulation shows a significant penalty for using low sample rates, although there is essentially no difference in minimum SNR for 76 Msps vs 100 Msps.

Figure 2.13 shows the simulated matched filter performance with and without phase

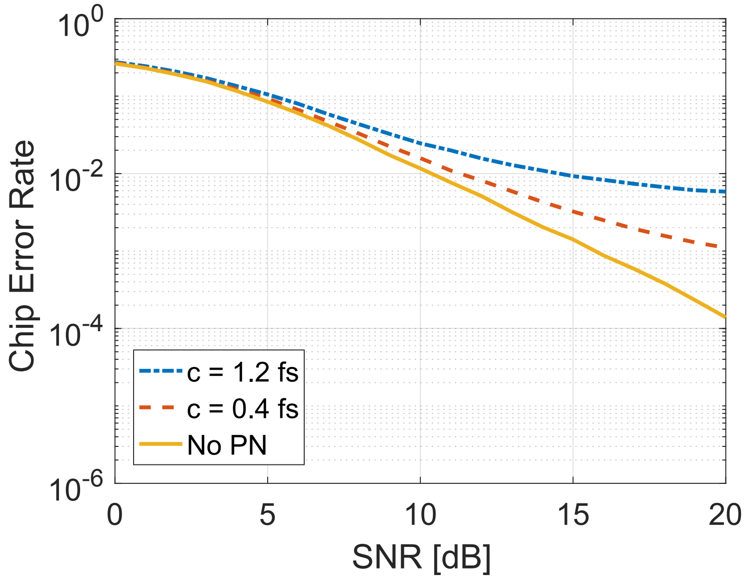


Figure 2.11: Simulated zero crossing counter demodulator performance while operating at a sample rate of 76 MHz with and without phase noise. The white phase noise values correspond to approximately 75 kHz of 1σ IF deviation for c = 0.4 fs and 150 kHz for c = 1.2 fs.

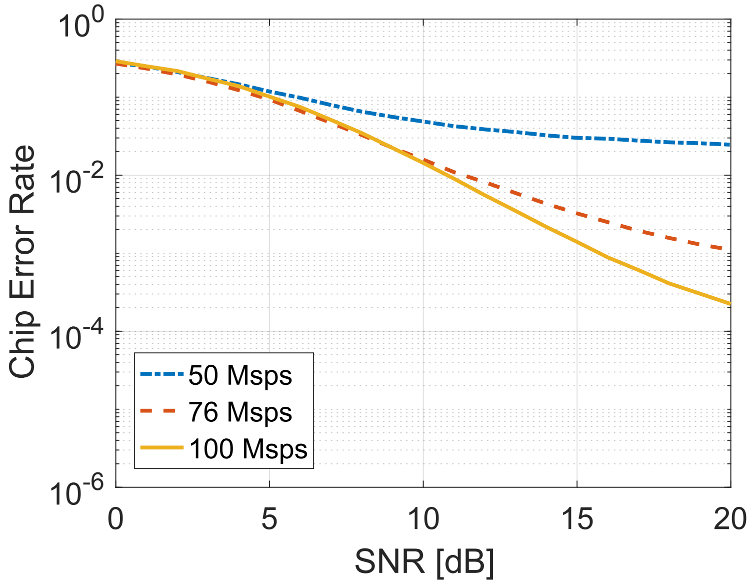


Figure 2.12: Effect of varying sample rate on zero crossing counter demodulator when a white phase noise value of c = 0.4 fs is used.

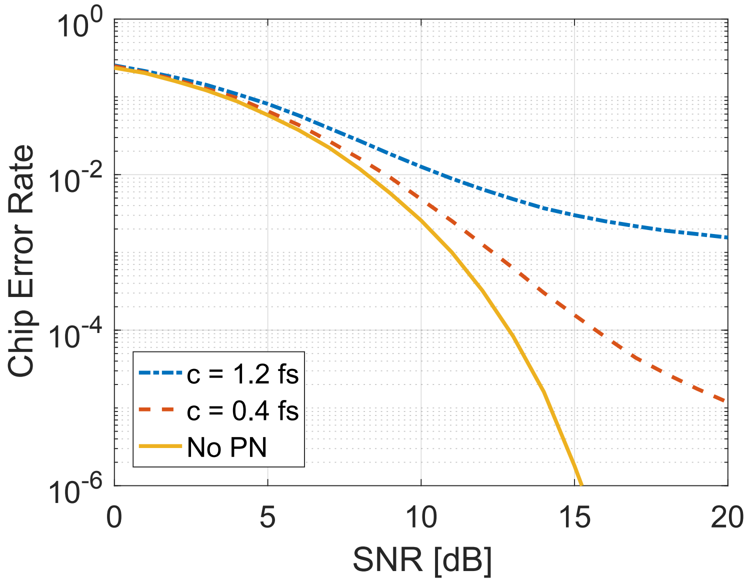


Figure 2.13: Simulated matched filter demodulator performance for varying levels of phase noise. The white phase noise values correspond to approximately 75 kHz of 1σ IF deviation for $c = 0.4$ fs and 150 kHz for $c = 1.2$ fs.

noise. Quantization by the ADC is not yet considered in this plot. In all cases the MF outperforms the ZCC. With no phase noise the minimum SNR is reduced by 1 dB to about 5 dB. Again the degradation from a large amount of phase noise is about 1 dB, and the error rate floor is reduced by about a factor of approximately three compared to the ZCC. The effect of phase noise is even more negligible in the matched filter case than it was for the ZCC.

Figure 2.14 shows the effect of quantizing the signal prior to being demodulated by the matched filter. There are minimum SNR penalties of about 3.5 dB, 1 dB, and 0.25 dB for the 3, 4, and 5 bit cases respectively. The worst case CER floor in the 3-bit case is about 3%. All three quantization cases are still able to achieve the required error rate performance although there is a considerable minimum SNR penalty in the 3-bit case.

Comparing the error rate performance of the two demodulators, it would appear that the mid-level performance scenarios are approximately equal under either case (ie, the 76 Msps ZCC and the 4-bit MF are very similar). While this may be the case in this initial simulation, there are several other factors still to take into account. First, a realistic CDR needs to be considered for the ZCC case which will reduce the performance. Second, the multi-bit input to the MF could benefit from digital filtering much more so than the 1-bit ZCC demodulator. Third, interference has not yet been considered which lends yet another advantage to the matched filter. These are also implementation differences between the two demodulators that favor the matched filter due to its lower sampling rate and relaxed comparator offset requirements.

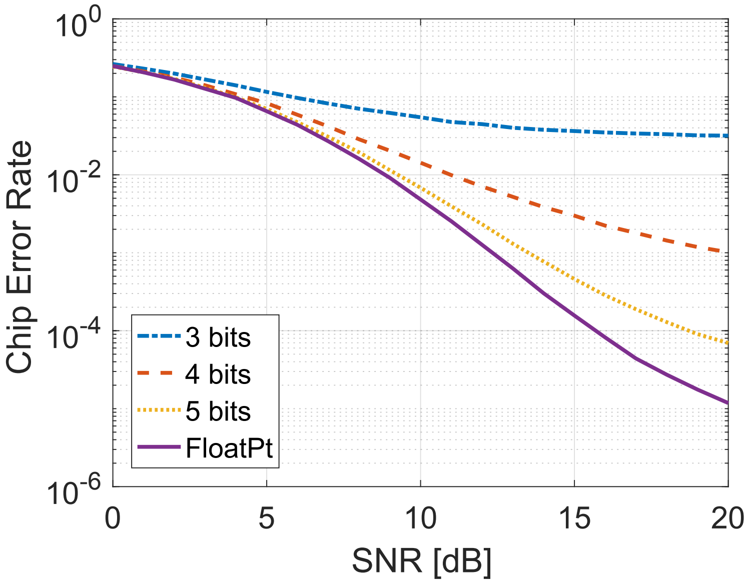


Figure 2.14: Effect of ADC quantization on matched filter demodulator performance. A white phase noise value of $c = 0.4$ fs is used for all cases.

Based on these results it would seem reasonable at this stage to choose four bits for the matched filter ADC resolution as a balance between performance and power. The use of 16 MHz for the ADC sample rate continues to be acceptable at this stage as these simulations have unearthed no compelling reason for changing it. The mid-range value of 76 MHz for the ZCC sampling rate also seems like a reasonable choice moving forward, although it is relatively easy to build in support for covering a wide range.

The main takeaway from this section is that the levels of phase noise found in an integrated LC LO are not going to be the limiting factor on the implementation of a crystal-free radio. It would appear that even much larger levels of phase noise are still acceptable, although one should consider that this is based on the assumption that white phase noise dominates in the oscillator. If flicker noise begins to play a significant role then the modeling process must also include it. It is also pertinent to note that variation from flicker noise occurs on timescales that are much larger than the length of the packet so one should be cautious to not be overly optimistic by only considering flicker noise over the very short time span of one packet. Related work on ring oscillator LOs [28] encounters these issues with flicker noise and cautions against relying on simulator models to accurately capture the importance of flicker phase noise.

Filtering Requirements

At this stage of the design the goal is to establish an initial idea of what filter order and type are required to meet the interference specifications. That information can then drive decisions about how best to implement that filter at the circuit level. Circuit implementation considerations will likely bring about changes to the desired filter profile, and the system model can then be updated to account for these new choices. There are entire books filled with the intricate details of filter design [29] [30] and the emphasis here is on not getting bogged down by the details, but rather to find a starting point for design iteration.

The matched filter demodulator with four ADC bits will be used as the starting point for determining filter requirements. It was noted in the previous section that no digital filtering was applied to either demodulator in the system model. This leaves out a significant advantage of the multi-bit ADC over the single-bit ZCC approach, which is that digital filtering can readily be applied prior to the matched filter. This allows both for filtering the ADC quantization noise to improve SNR as well as implementing image rejection. Digital filtering could also be applied to the 1-bit input case as is the norm in sigma-delta ADCs, but without quantization noise shaping to increase the dynamic range, the benefit is less significant.

As was the case with analog filter design, digital filter design is also a very deep and broad subject. The filters used in this work were designed via *fdatool* in MATLAB and the reader is referred to one of the many DSP books for a refresher on digital filter fundamentals [31] [32]. The coefficients of a prototype complex bandpass FIR filter are shown in Figure 2.15 and the corresponding frequency response is given in Figure 2.16. The use of complex coefficients allows the pass-band to be non-symmetric about the zero frequency axis which enables it to perform image rejection. It can be difficult to analyze the trade-off between how many taps and to what quantization accuracy an FIR filter should be designed. While more taps and smaller quantization steps will result in better stop-band rejection, the internal filter computations will be more complex and thus consume more power. The filter shown here compromises by using 8 taps to achieve roughly 20 dB of rejection.

To compare the effect of adding the complex bandpass filter, Figure 2.17 compares the 4-bit matched filter case from the previous section with and without the additional filtering. It can be seen that a significant improvement in performance is obtained by filtering the quantization noise after digitization. The SNR required to reach the 6.5% error rate goal has decreased to approximately 2 dB (recall that SNR here refers to the signal to noise ratio prior to the ADC and assumes a very sharp filtering of noise prior to the ADC). This decrease in minimum SNR is about 4 dB which comes about due to the filtering of the image channel noise and reduction of quantization noise outside the channel bandwidth. To check the image rejection, an input signal with an SNR approximately 3 dB over the minimum detectable is simulated with an equal amplitude interferer placed in the image channel. The resulting CER of 2.2% confirms that the receiver still operates with less than 1% packet error rate under this condition.

Of the remaining interference specifications to investigate, the most stringent is the al-

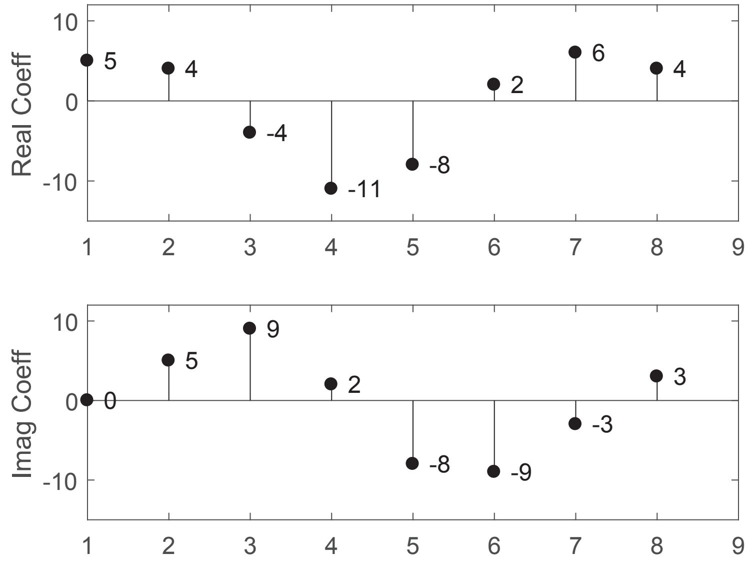


Figure 2.15: Coefficients for digital complex bandpass image rejection filter.

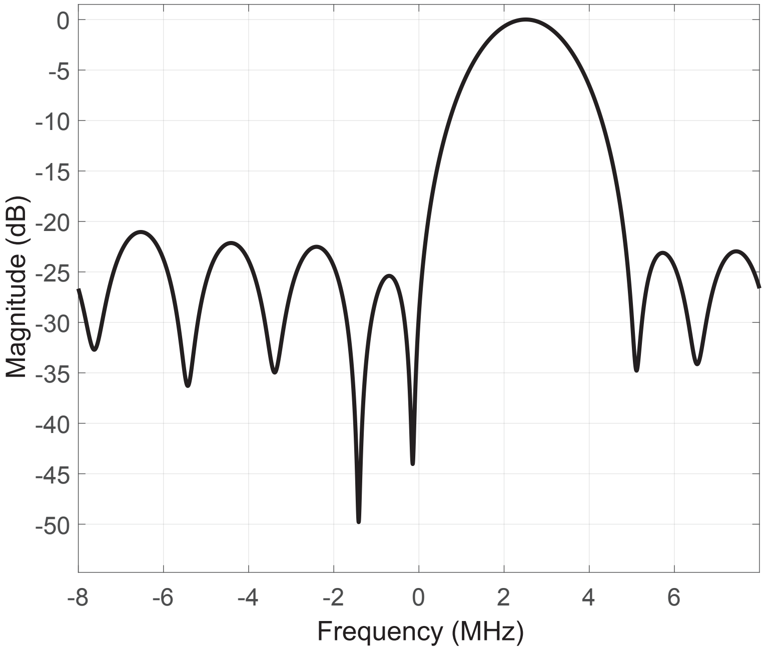


Figure 2.16: Frequency response of example image rejection filter with floating point internal calculations.

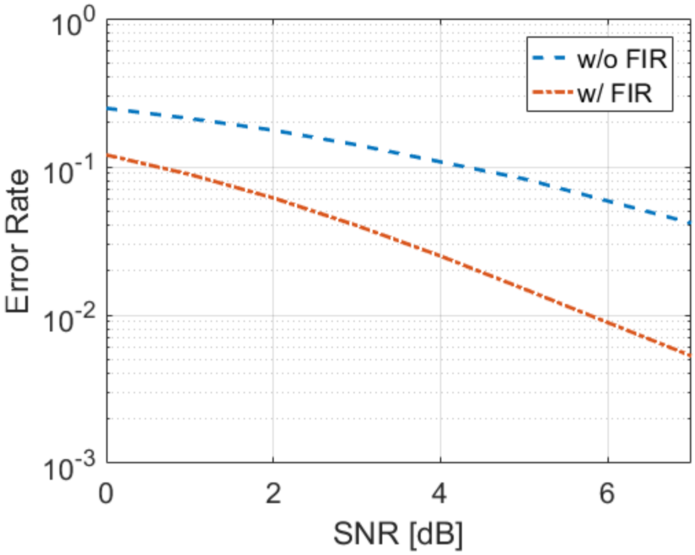


Figure 2.17: Comparison of matched filter demodulator performance with and without complex bandpass filtering after digitization.

ternate channel rejection when the interferer is on the opposite sideband of the LO than the desired channel. In this scenario, the blocker which is 10 MHz away from the desired channel, gets down-converted to 7.5 MHz due to the choice of a 2.5 MHz IF. If this most stringent situation can be filtered, then all other scenarios should not present issues.

Two standard analog filter prototypes will be used here to obtain an approximate idea of what level of filtering is required. The Butterworth response is maximally flat in the pass-band which causes it to roll-off less sharply than the Chebyshev response. The cost to the sharper roll-off of the Chebyshev is tolerating pass-band ripple and requiring higher Q poles for implementation [29]. These are only two of many options for filter profiles, but will be used for a first evaluation of the required roll-off. A modulated blocker is added to the simulation that is 30 dB larger than the desired channel and located 10 MHz away on the other side of the LO. The combined signals after down-conversion are filtered by second, third, and fourth order Butterworth and Chebyshev filters and the CER results are given in Table 2.1. All filters have their corner frequency set to 3.5 MHz, and the Chebyshev is specified to have 0.5 dB pass-band ripple. A SNR 3 dB over minimum detectable is again used for the desired signal, and the simulation is carried out using the 4-bit matched filter with complex bandpass and $c = 0.4$ fs phase noise coefficient.

Several conclusions can be drawn based on these simulated results. A second order filter is not likely to provide sufficient filtering regardless of how steep its roll-off is. The third order Chebyshev provides very nearly the required attenuation, while the Butterworth is slightly worse. This indicates that a properly designed third or fourth order filter is probably sufficient, and that pole locations should be chosen which generate as steep a roll-

Profile	CER
Butterworth 2nd	15.5%
Butterworth 3rd	8.7%
Butterworth 4th	10.4%
Chebyshev 2nd	25%
Chebyshev 3rd	6.8%
Chebyshev 4th	11.6%

Table 2.1: Error rate performance for various filter types and order.

off as possible. The higher order filters actually result in worse performance here, given that no consideration has been given to their group delay effect on inter-symbol interference. Higher order filters can provide better suppression of unwanted signals, but care must be taken during their design to not adversely affect the modulated signal.

In general the matched filter demodulator outperforms the zero crossing counter when interference is considered. The matched filter's convolution with filter templates is essentially an additional bandpass filter which makes it more frequency selective. The ability to add more robust digital filtering to the multi-bit ADC case also lends a considerable performance increase. In either design case interference should be attenuated as much as possible prior to the ADC but this is especially important for the ZCC due to the small dynamic range of its 1-bit digitization. While the main receiver path for the Single Chip Mote will implement the matched filter approach for these benefits, the ZCC will also be included as a lower power, lower performance mode.

The simulations performed here are a good starting point for modeling the receiver but still do not provide all the required information to complete the design. In order to more accurately set specifications on noise limits in the receiver more details need to be known about the specific implementation of blocks like the analog filters. Including simulation of the actual Verilog implementation of digital baseband blocks (such as presented at the end of Chapter 4) as they are developed will also serve to hone the accuracy of the model. As such the modeling process is iterative as the analog and digital portions of the design progress. The design of the analog and digital parts of the receiver are discussed in Chapters 3 and 4 respectively and the result of an updated behavioral model including those designs is shown in Figure 2.18. This is the point where the modeling was stopped for the SCM receiver and the required noise performance for the receiver frontend was extracted from here. It can be seen that approximately 6 dB of SNR is required to achieve the 6.5% that translates to the required 1% packet error rate. This implies from Equation 2.9 that a maximum noise figure of 20 dB is acceptable for the receiver. This value is quite high in comparison to many receivers and it is typical to implement some margin to account for other non-idealities that were not considered. Noise figure will be discussed further during the frontend design in Chapter 3.

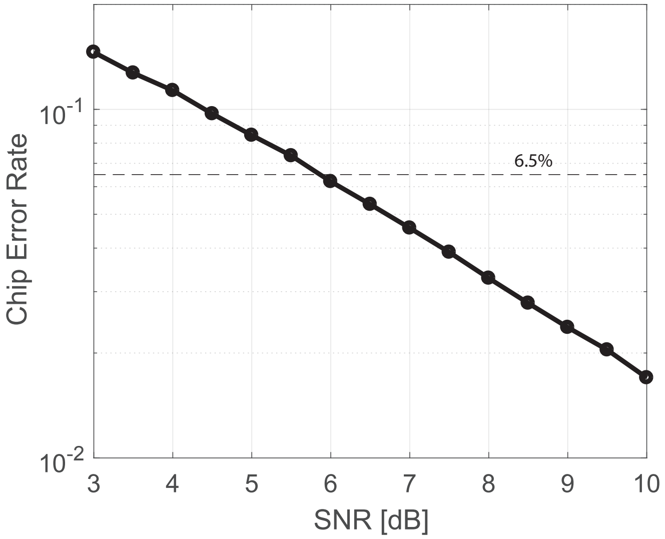


Figure 2.18: Result of an updated behavioral model which accounts for the parts of the design introduced in later chapters. The minimum SNR to achieve a 1% packet error rate is approximately 6 dB.

Chapter 3

Receiver Analog Design

The analog portion of the Single Chip Mote receiver design encompasses everything between the antenna and the digitized samples going into the digital baseband. The tasks that must be performed by this portion of the receiver can be broadly broken down into three functions: RF gain and down-conversion, low frequency gain and filtering, and digitization as shown in the generic receiver diagram in Figure 3.1. This chapter addresses each of those three functions in turn as well as the additional supporting hardware blocks required such as clock generation and supply regulation.

This chapter focuses on the circuit level implementation of the low IF topology introduced in Chapter 2. The implementation strategy was to utilize proven block level topologies from literature and incorporate them into a complete receiver. Block level performance and power were traded off in an attempt to meet the required 802.15.4 design specifications with as little power as necessary. The crystal-free nature of the overall transceiver was also a consideration in the choices of circuit topologies to use. The resulting receiver implements quadrature down-conversion to enable image rejection and performs the bulk of the interference filtering in the analog domain. Digitization occurs with 4-bit ADCs operating at a sample rate of 16 MHz and a full-scale amplitude of 50 mV was used to limit the impact of non-linearity by lowering the gain requirements. While the primary receiver path will be based on a matched filter it was also decided to support a zero crossing counter mode as a backup demodulator which introduces additional constraints on clock speeds and the comparator implementation.

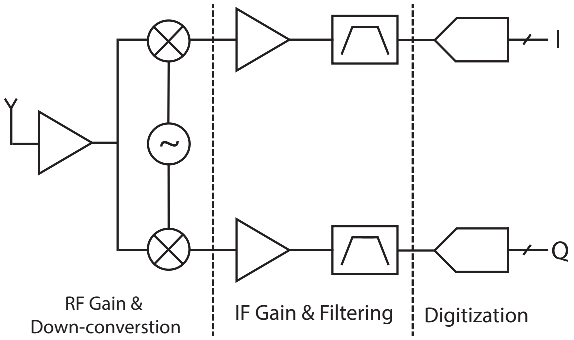


Figure 3.1: Generic IQ receiver indicating the three primary functions that must be performed.

3.1 RF Frontend

Traditional receivers use a low noise amplifier (LNA) very soon after the antenna to provide RF gain. Having gain early in the signal path allows the receiver to have very good overall noise performance. While the use of a LNA enables a low noise figure it costs power to generate linear gain at RF in this manner. The design requirements outlined in Chapter 2 are relatively relaxed in comparison to many receiver designs. This presents the opportunity for the LNA to be omitted and to trade performance for lower power in this design. While eliminating the LNA seems like an obvious path to a lower power implementation consideration must still be given to the receiver power budget as a whole. The LNA and the LO are typically the largest power consumers in a receiver and their combined contribution to the power budget must be considered. Simply eliminating the LNA to save power is a losing strategy if it tremendously increases the power requirements of the LO.

An alternative to using a LNA is the passive mixer first topology [33] [34]. In this type of receiver the mixer switches are directly connected to the antenna without any intermediary gain stage. Since the down-conversion takes place without using any active circuit elements this type of receiver can be highly linear. Another advantage is the N-path filtering effect which is created by the bi-directionality of the mixer switches. There is no isolation from the low frequency side of the mixer back to the RF port so whatever impedance is seen at baseband is effectively up-converted to RF [35]. This not only provides filtering benefits but also allows for tuning the impedance seen at the RF port by adjusting the baseband impedance. While this effect forms a bandpass filter around the LO frequency the achievable filtering with this technique is limited by the on resistance of the mixer switches. At frequencies far away from the LO the achievable rejection reaches a floor of $R_{switch}/(R_{antenna} + R_{switch})$ [36]. This requires the mixer switches to have very low on resistance which implies a large capacitive load and increased LO power requirements for driving them. The multi-phase LO waveform generation for this type of mixer tends to consume a significant amount of power as multiple non-overlapping square waveforms must be generated at RF. While low noise figures can be achieved by the passive mixer first topology, the relatively low amount of gain achievable in the mixer shifts the noise burden to the first baseband amplifiers. Research is ongoing in search of ways to further improve this topology and take advantage of its linearity benefits such as exploiting the up-conversion of baseband impedances to lower the required mixer gate size and thus further reduce the requirements on LO power [36].

There are two methods utilized in the SCM receiver to reduce the power demands of the passive mixer first architecture. Both of these techniques originate from [37] which forms the basis for the topology that will be used for the SCM frontend. The first method is to exploit the voltage gain achievable through passive impedance transformation. At 2.4 GHz it is possible to fully integrate on chip either matching networks or transformers that will boost the 50Ω antenna impedance up to a higher value. As a consequence of this increase in source impedance some passive voltage gain is achieved which improves the total receiver noise performance in the same manner as having gain in a LNA. The passive gain network is implemented using a tapped capacitor impedance transformation with an inductor used to make the structure resonant at 2.4 GHz [37]. A single ended implementation was chosen to ease the interface to test equipment. The second method to lower power is to utilize sine waves directly from the LO to drive the mixer switches rather than using LO buffers to create square waves. This allows the mixer gate capacitance to be included in the resonant tank and removes the high power requirement to generate complex LO waveforms using digital gates. Overlap in conduction between mixer switches must now be avoided by adjusting the DC level of the LO waveform relative to the switch threshold voltages. Quadrature generation in [37] was accomplished by coupling two resonant tanks together. Rather than doubling the power and area by adding a second tank the SCM receiver utilizes a passive RC polyphase filter to generate the quadrature phases [17].

Matching Network

A single ended matching network topology was chosen for ease of interfacing with antennas and test equipment. The first SCM generation [27] utilized differential RF inputs and the difficulties introduced by the requirement for an off-chip balun discouraged that approach going forward. The RF port is also shared between RX and TX in order to require only one antenna and one matching network. The design requirements for these two situations are fundamentally at odds with one another. The receiver benefits from a large step-up ratio of impedance across the passive the network to enable a large passive voltage gain. The transmitter however requires the opposite when trying to deliver a large amount of power efficiently to the antenna. Due to output power requirements of SCM it is possible to compromise and share the same matching network at the expense of performance, the burden of which was primarily shifted to the transmitter (see [17] for further details). From the receiver standpoint the primary issue to be considered when sharing the matching network is the additional capacitance introduced at the mixer input by the capacitor providing the coupling to the PA output.

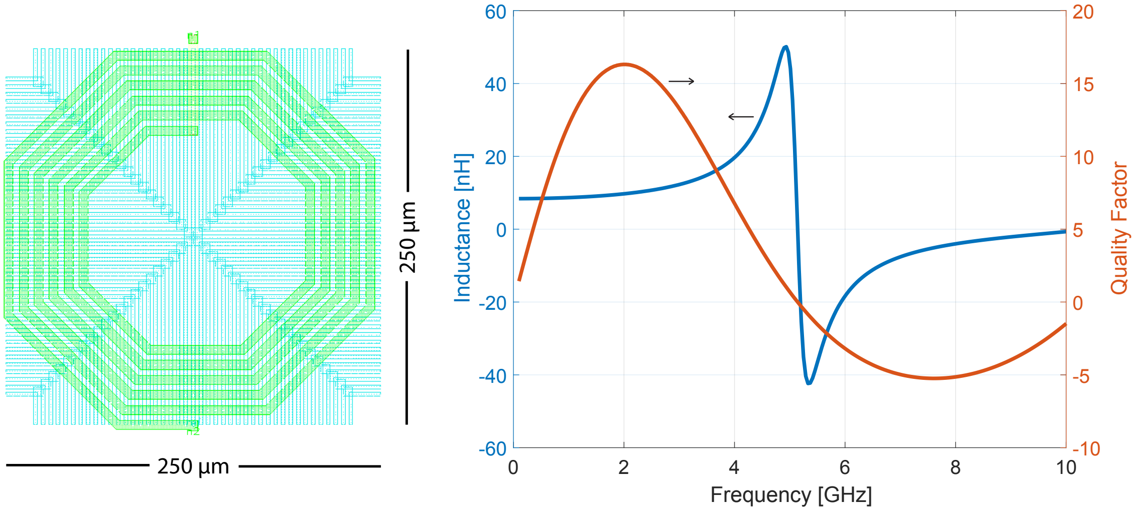


Figure 3.2: Left: Layout view of the inductor used in the RF matching network. Right: EMX simulation results for the inductance and quality factor of the inductor.

The inductor is the critical component in the passive frontend topology used here. To maximize voltage gain of the passive network it is desirable to have the largest LQ product possible [37]. An area restriction of 250 μm on a side was chosen for the inductor to balance the achievable gain with the amount of design area required. The metal stackup for this 65 nm process provides an ultra-thick top metal layer which was used to construct the inductor. The EM tool Lorentz Peakview was used to explore the design space and produce the inductor shown on the left in Figure 3.2 along with the simulated performance obtained using Integrand EMX on the right. The 5.5 turn inductor has a SRF of 5.1 GHz and L/Q values at 2.44 GHz of 10.5 nH and 15.5 respectively. A M1 patterned ground shield and intrinsic substrate doping are used under the inductor to reduce the degradation of quality factor due to substrate currents. Rather than request metal density waivers from the foundry for such a large area a custom metal fill pattern was utilized. This pattern had much lower metal content than the automated metal fill scripts and was simulated to impact the inductor Q by only approximately 1%.

Previous hardware generations had included tuning of the match center frequency in order to correct for process variations. Including tuning requires the addition of more capacitors and switches at the very critical node at the mixer input. In transmit mode and when receiving high input powers this node can experience significant voltage swings which will potentially activate diodes in these additional tuning transistors. Due to the relatively low Q of the passive network the impact of variation in the match frequency is not expected to cause severe degradation in performance. For these reasons it was decided to omit tuning on this hardware iteration as the risk was higher to include it than it was to accept a small performance loss if the match was slightly off center.

Design Optimization

With a passive network providing gain which is loaded by a passive mixer that is being driven directly from the LO via a polyphase network the design becomes very interwoven between the various components. It is not possible to design one block in isolation without considering its effects on the other blocks and thus the overall performance. The design task is further complicated by the goal of sharing the antenna port between both receive and transmit modes. In order to examine the design of the RF frontend consider the simplified model in Figure 3.3 where a passive gain network is loaded by a mixer which then drives further baseband circuits with some input referred noise. The achievable gain in a passive matching network largely depends on the quality factor of the available inductor as shown on the left in Figure 3.4. A matching network on its own however is useless without connecting it to additional circuits which propagate the signal through the receiver. These additional circuits will introduce loading to the matching network which can substantially reduce its gain. The plot on the right in Figure 3.4 shows that attaching a mixer with a 1 $k\Omega$ input impedance to a matching network built around an inductor with Q = 15 reduces the achievable gain by more than 6 dB. The input impedance of the mixer can be increased by reducing the size of the switching devices but this comes at the expense of increased noise as seen in Figure 3.5. Clearly there is an optimum design point that balances these two opposing trade-offs to result in the best overall receiver noise performance.

The mixer is the most complicated block to analyze due to its non-linear, frequency translating nature. The passive mixer references discussed previously provide some basis for hand analysis for the mixer but are generally based on many assumptions and are good for reaching an initial point in the design space. Ultimately the design needs to be verified with simulation and it would be useful to be able to explore the effect of parameter variation on performance. This is difficult to do directly in simulation due to the interdependence of the many portions of the design. It is not possible to simply sweep a variable like mixer switch width as the matching network needs to be redesigned every time the mixer changes to provide the optimal voltage gain. The approach taken here is to reduce the mixer design space to three primary variables: switch width, LO DC level relative to threshold, and LO amplitude. In order to evaluate the impact of the mixer on the rest of the design, there are three quantities that must be known: the mixer input impedance, input referred noise, and voltage gain. This becomes a tractable simulation where it is possible to quantify these three outputs against the three parameters of interest over a modestly sized design space. It is then possible in MATLAB to design the optimum matching network for each of these mixer designs and calculate an overall system noise figure given an assumed input referred noise of the amplifiers following the mixer.

An example of the result of this series of simulations and calculations is shown in Figure 3.6 for three different LO amplitudes. In a CMOS LC VCO amplitude is proportional to current so using the smallest swing that can achieve the required performance is beneficial. The input referred noise of the baseband amplifiers is assumed to be $10nV/\sqrt{Hz}$ which is approximately the attainable noise level when biasing the amps with on the order of 10 μA.

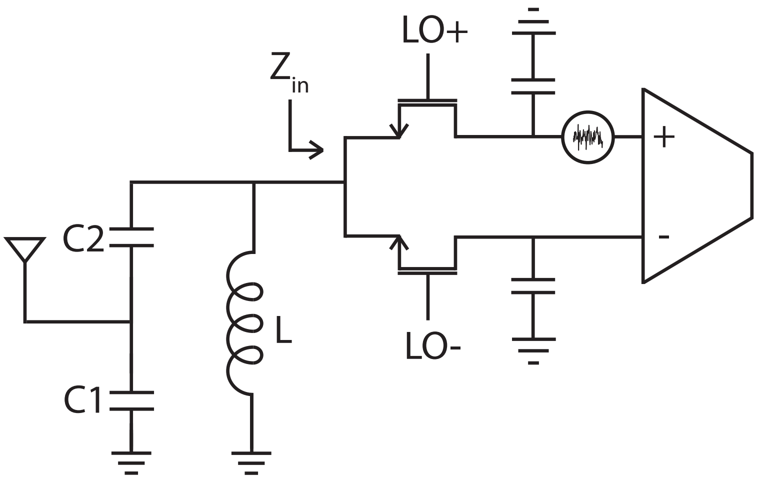


Figure 3.3: Circuit model used for optimizing the tradeoffs in the frontend in order to design the mixer. Simulated mixer performance is used in a MATLAB model to find the best overall system noise figure.

It is also assumed that the noise in the image channel will later be removed by digital filtering and thus 3 dB can be subtracted from the calculated noise figure. The minimum achievable noise figures in the 100 mV, 200 mV, and 300 mV LO amplitude cases are 15.4 dB, 12.9 dB, and 12.3 dB respectively. While in all cases the required noise performance can be achieved there is a 2.5 dB advantage of using 200 mV amplitude over 100 mV. The difference of only 0.6 dB between the 200 mV and 300 mV cases shows the diminishing returns of spending more power in the LO. The resulting 12.9 dB NF (15.9 dB if image noise is not filtered) is assumed to have an acceptable level of margin over the maximum 20 dB noise figure that was derived from the receiver model in Section 2.3. Thus this bias current assumption for the first IF amplifiers seems acceptable and 200 mV is chosen as the target LO amplitude to be delivered to the mixer in the SCM receiver.

Mixers

The use of a single ended matching network leads to the choice of single balanced mixers for this design which results in fewer mixer switches to drive and simpler LO distribution. The lack of LO isolation from the RF port will potentially create DC offsets created by self-mixing, but those can be dealt with by AC coupling the mixer output to the following amplifiers. While generating multi-phase non-overlapping waveforms to drive the mixers can increase the performance of the receiver, the power required to generate these additional LO waveforms can be significant [38]. This work trades performance for lower power dissipation by not including active buffers between the LO and mixer gates. Quadrature LO signals are generated by a single-stage passive RC polyphase filter that is directly connected to the

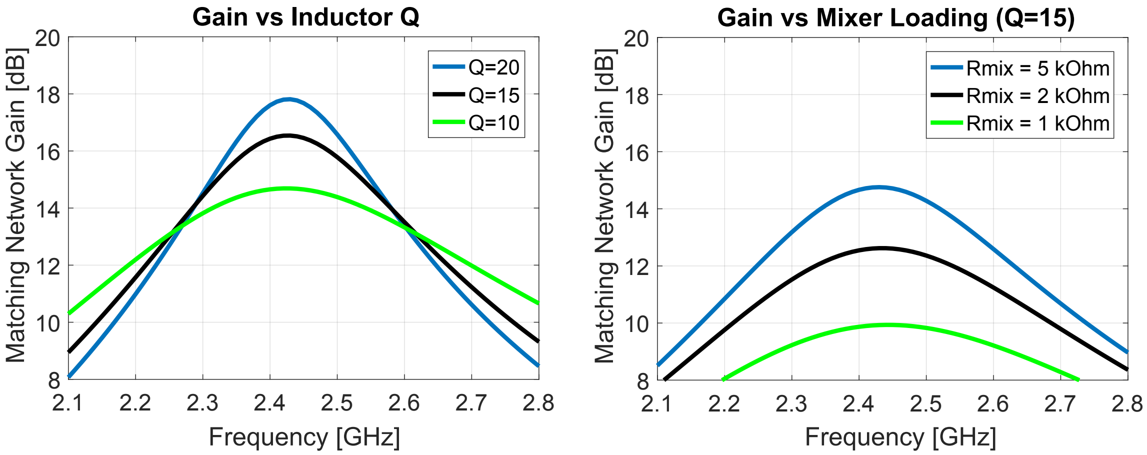


Figure 3.4: Left: The achievable gain of an unloaded tapped capacitor matching network based on the quality factor of the inductor. Right: The reduction in gain of the matching network as a function of the load impedance introduced by the mixer.

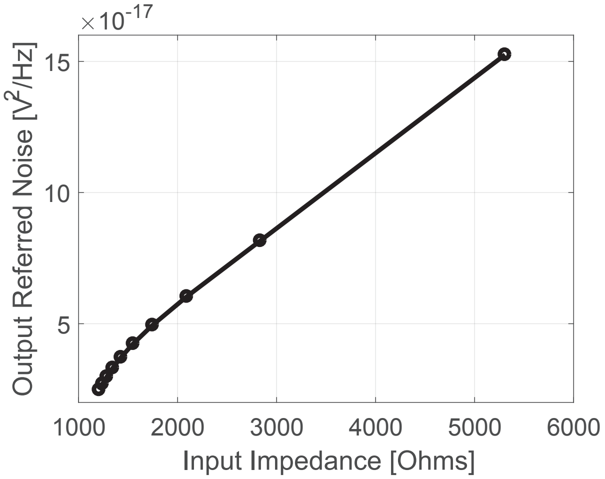


Figure 3.5: While larger mixer devices reduce the input referred noise of the mixer they also lower its input impedance which reduces the gain of the passive frontend.

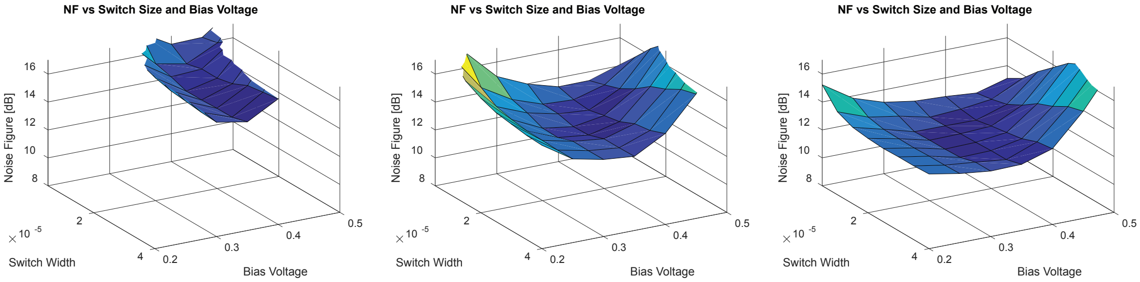


Figure 3.6: Result of optimization model for mixer design with LO amplitude of 100 mV (left), 200 mV (center), and 300 mV (right). The Z-axis cutoff is at the maximum noise figure of 18 dB. A LO swing of 200 mV can achieve a noise figure of 12.9 dB which is only 0.6 dB worse than increasing the LO power for a 300 mV swing.

LC tank and loaded by the mixers with their associated bias network. For detailed design information on the polyphase filter and LO in general see [17]. The design target was to deliver 200 mV of swing amplitude to each mixer gate based on the previous analysis of the effect of swing on noise figure.

The DC bias of each mixer switch is set independently to allow for trimming of mismatch in threshold voltages. The DC voltages are generated using four copies of a binary weighted current DAC routed through a variably sized diode connected load as shown in Figure 3.7. The polyphase filter outputs are AC coupled to the mixer gates and a large resistance is added to reduce the effect of the bias loading on the polyphase filter. The sources of all four mixer switches are connected together and DC biased to ground by the match inductor. AC coupling capacitors are inserted between the mixer and TIA to prevent DC offsets from saturating the following amplifier. To prevent the large swings present at the mixer input node during transmit mode from turning on source-body diodes in the mixer switches deep N-wells were used that could be switched to a high impedance state during transmission. The mixer devices were sized by following the approach outlined at the start of this chapter. Devices were first characterized in SPICE simulation for their noise, gain, and input impedance vs layout dimensions. A MATLAB lumped element model of the frontend was then used to find the optimum sizing to minimize the achievable noise figure for a given input referred noise in the first IF amplifier.

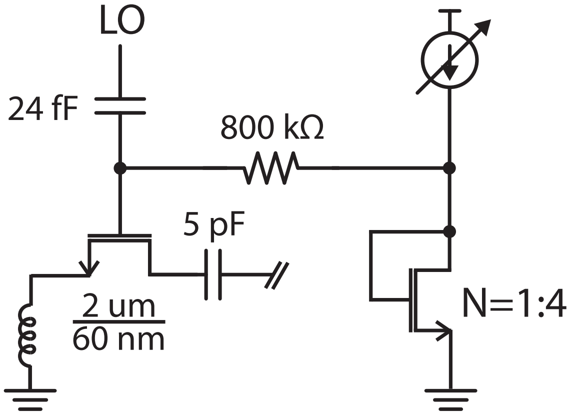


Figure 3.7: The DC bias of the sinusoidal LO swing is set individually for each mixer switch using a variable current source into a diode connected load. The sources of all mixer switches are DC biased to ground via the match inductor.

TIA

Due to the relatively low gain and moderate noise of the passive frontend the first active amplifier becomes the critical noise limitation. Inverters are used to construct the TIA in order to take advantage of the doubled transconductance for the same bias current. Constructing a pseudo-differential amplifier from a pair of inverters in resistive feedback provides a very simple way to implement this first amplifier. The self-biased nature of this structure removes the need for additional bias and common mode feedback circuitry which speeds up the design process. The single stage implementation also eliminates issues with stability.

A downside to a simple pair of inverters in resistive feedback is their lack of common mode rejection. The amplifier responds equally to differential and common mode inputs. A simple modification of adding two additional transistors as first demonstrated in [39] can alleviate this issue. These two additional transistors, labeled as M3 in Figure 3.8, serve to degenerate the gain for common mode signals while retaining the differential gain. The common mode gain is reduced to gm_1/gm_3 while the differential gain remains $(gm_1 + gm_2)(r_{o1}//r_{o2})$.

Gain control is implemented by including a variable load resistance between the differential outputs. The elements are sized to provide twenty gain steps of approximately 1 dB each. This gain control method was chosen rather than adjusting the feedback resistors to reduce the effect of changing the load impedance on the mixer. Since the passive mixer is bidirectional any change in its load impedance will be up-converted to RF and affect the input matching. The overall TIA which follows the mixers is shown in Figure 3.8. Changing the amplifier gain will also change its bandwidth so variable feedback capacitors are included to enable the capability to adjust the bandwidth if desired. The combined TIA and passive frontend provide about 35 dB of gain from RF to IF. The complete implementation of the frontend signal path is shown in Figure 3.9.

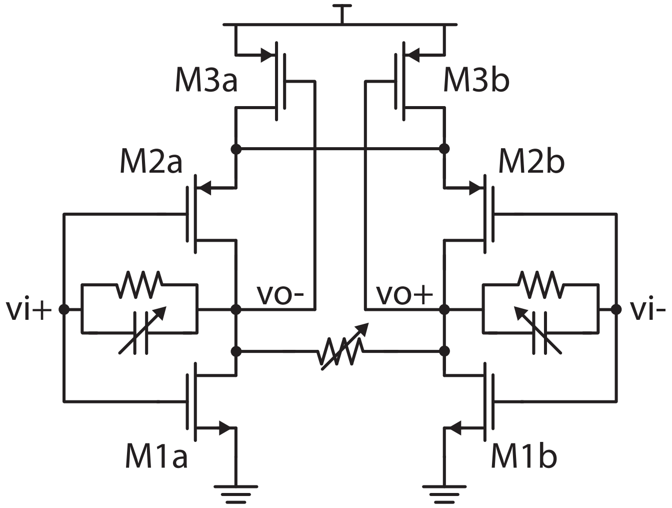


Figure 3.8: The inverter-based amplifier used in the frontend TIA is based on [39]. This structure reduces the common mode gain compared to a pseudo-differential pair of inverters and self-biases without the need for common-mode feedback or other biasing. Gain control is implemented with a variable resistor connected between the differential outputs.

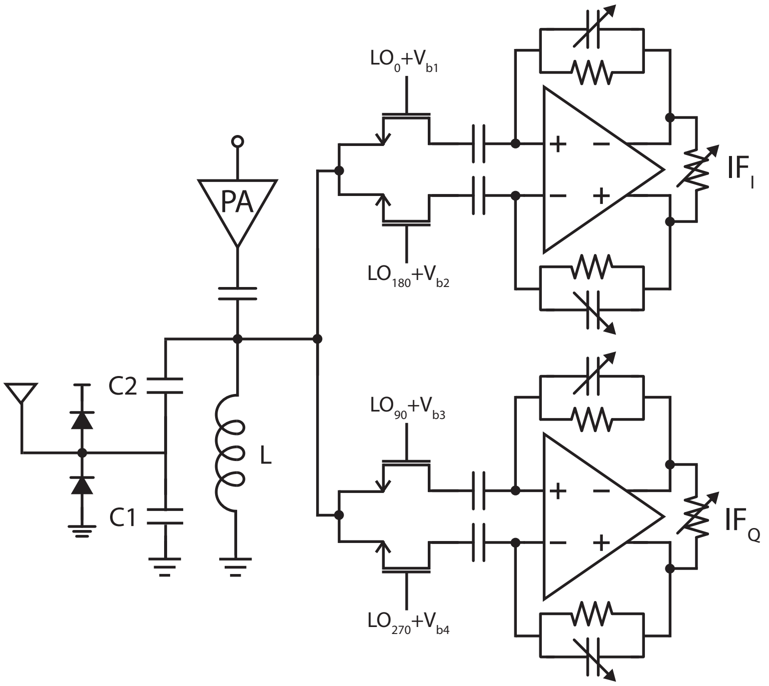


Figure 3.9: Schematic of the complete frontend from the antenna pad to quadrature IF outputs including the connection for sharing the RF port with the transmitter.

3.2 Filters

Simulation results from Chapter 2 indicated that a third or fourth order filter should be sufficient to meet the adjacent and alternate channel filtering requirements of 802.15.4. In this section the choices made for filter implementation will be discussed which can then be inserted back into the previous MATLAB model to verify that the requirements are met. In addition to providing filtering these stages must also provide voltage gain to increase the signal amplitude to a level appropriate for digitization. With a 50 mV amplitude full scale range for the ADC and a minimum signal power of -85 dBm on the 50Ω antenna a total voltage gain of approximately 70 dB is required. With 35 dB of gain in the RF frontend the filters must contribute at least another 35 dB of gain. Furthermore the gain should be made programmable to facilitate interfacing with an automatic gain controller. The filter has a substantial amount of gain preceding it so noise will not be a major design consideration. Low power operation is desirable which is fundamentally at odds with linearity. Given that the SCM receiver is intended to operate in a battery constrained environment, preference will be given to low power operation over high linearity.

The three main options for implementing the required filtering in the low MHz range are opamp-RC, gm-C, and switched capacitor. The opamp-RC can have good linearity due to the feedback used, but requires low output impedance opamps, relatively large component values at these frequencies, and must deal with component variation effects. Gm-C filters lessen the gain-bandwidth requirements by operating open loop and driving only capacitive loads. The open loop operation of gm-C reduces the linearity performance and the component variation effects are still present as well. Closed loop switched capacitor circuits benefit from their transfer functions being dictated by capacitor ratios and clock frequencies which can both be well controlled. The closed loop nature again improves linearity, but the amplifiers must be designed to be fast enough to settle to acceptable accuracy levels. All of the above options also require at least one amplifier per pole implemented in the transfer function which is undesirable for power consumption. An alternative option is to utilize switched capacitor circuits that operate with open loop active elements as in [40] which implements a 7th order lowpass filter using a single active amplifier.

A schematic of a simplified version of this type of filter is shown in Figure 3.10. The filter operates by integrating charge proportional to the input voltage onto a holding capacitor C_h during one clock phase, then passively charge sharing with a sampling capacitor C_s in a second clock phase. In discrete time this is implementing an IIR filter with the transfer function given by Equations 3.1 and 3.2. This charge sharing process can be extended to an arbitrary number of holding capacitors to implement higher order filters using one active element as in [40]. The DC gain of the filter is given in Equation 3.3 and depends on the transconductance of the active device, the integration time window, and the size of the sampling capacitor. The passive network of switches and capacitors is amenable to process scaling and is very linear which makes the transconductor the primary limitation on linearity.

$$H(z) = \frac{1-a}{1-az^{-1}} \tag{3.1}$$

$$a = \frac{C_h}{C_h + C_s} \tag{3.2}$$

$$A_v = \frac{gmT_s}{Cs} \tag{3.3}$$

A downside to the use of a discrete time filter is the effect of aliasing interference beyond the Nyquist rate back into the bandwidth of the filter. A benefit of the windowed integration operation in the previously described filter is that it inherently provides some level of anti-alias filtering. A rectangular windowed integration in time produces a sinc response in the frequency domain which will help to attenuate interference beyond the sampling rate. When combined with the one real pole of the frontend mixer and TIA the achievable anti-alias filter response is shown in Figure 3.11. This response has nulls at multiples of the filter clock frequency and is applied to the signal prior to the discrete time transfer function of the filter itself. Additional filtering for out of band signals is also obtained by the frequency selectivity of the passive matching network. Another benefit of the windowed integration operation is its tolerance of clock jitter [41] which will be higher in a crystal-free system utilizing free running clocks.

A downside to the topology used in [40] is that it can only implement poles on the real axis. This results in a slower roll-off in the transition region which is undesirable for attenuating close-in interference. Complex conjugate poles are needed to generate steeper filter roll-off but require a feedback path to implement. Several works have proposed solutions by adding active feedback paths to obtain the desired complex poles but the addition of more active elements increases power and adds noise [42] [43]. An alternative implementation [44] shown in Figure 3.12 achieves the required negative feedback passively using polarity inversions in the switched capacitor network. The discrete time transfer function of this topology is given in Equation 3.4 where Q_{in} is the amount of charge integrated during the sampling period. Having a gain of less than unity in the feedback path places restrictions on the Q of the poles achievable. On the left in Figure 3.13 is a plot of the possible pole locations obtained by sweeping the values of C_1 and C_2. A fourth order filter can easily be formed by cascading two of these structures and a target gain of 20 dB per stage satisfies the overall gain requirements for the receiver. A MATLAB analysis was carried out to determine the capacitor ratios for the best performance of a cascade of two sections given the restrictions on pole placement. An example of the transfer functions of such an arrangement are shown on the right in Figure 3.13.

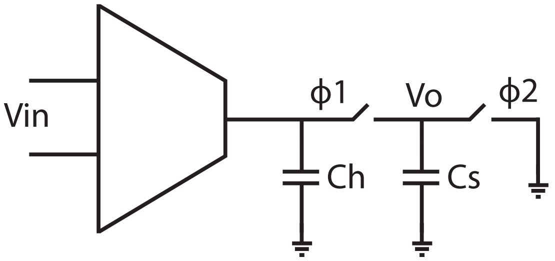


Figure 3.10: Schematic of a simple discrete time IIR filter implementing one pole.

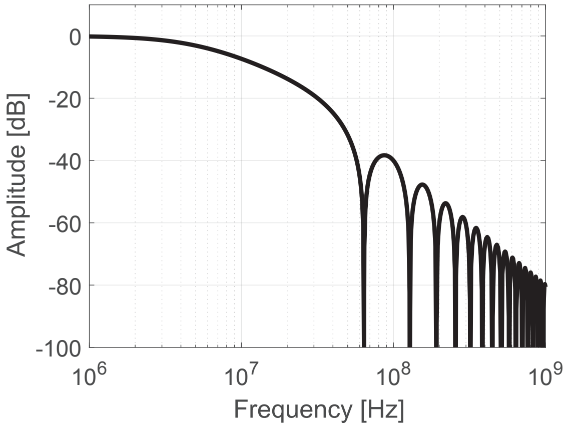


Figure 3.11: Transfer function of the anti-aliasing achieved by the combined filtering of the TIA and the sinc response of the windowed integration.

$$\frac{V_o(z)}{Q_{in}(z)} = \frac{1}{C_s}\left(\frac{z(1-a_1)}{(z-a_1)}\right)\left(\frac{z(1-a_2)}{(z-a_2)}\right) \tag{3.4}$$

$$a_1 = \frac{C_1}{C_1 + C_s} \tag{3.5}$$

$$a_2 = \frac{C_2}{C_2 + C_s} \tag{3.6}$$

The design of the transconductor is critical to the performance of the discrete time analog IIR filter. The impedance of the switched capacitor load should be much lower than the output impedance of the amplifier to reduce non-linearities and ensure that the transfer function is defined by the ratios of capacitors and switching frequency [45]. A low amplifier output impedance also degrades the null depth of the anti-aliasing filter obtained from the windowed integration [41]. The achievable in-band gain of the filter is dependent on the size of the sampling capacitor C_s as shown in Equation 3.3. More gain can be obtained

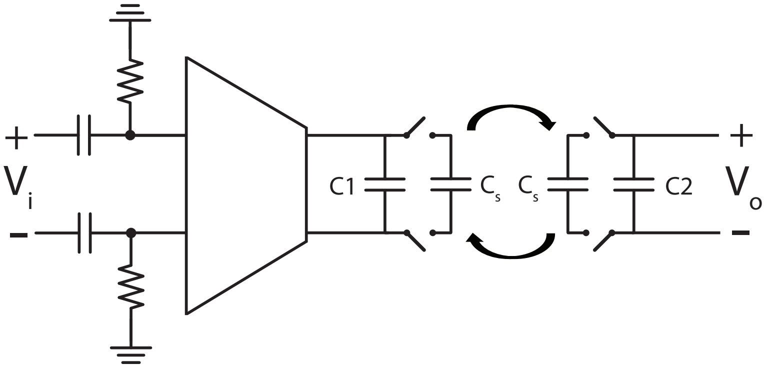


Figure 3.12: The topology used in [44] is able to implement complex conjugate poles using polarity inversions in the switched capacitor network.

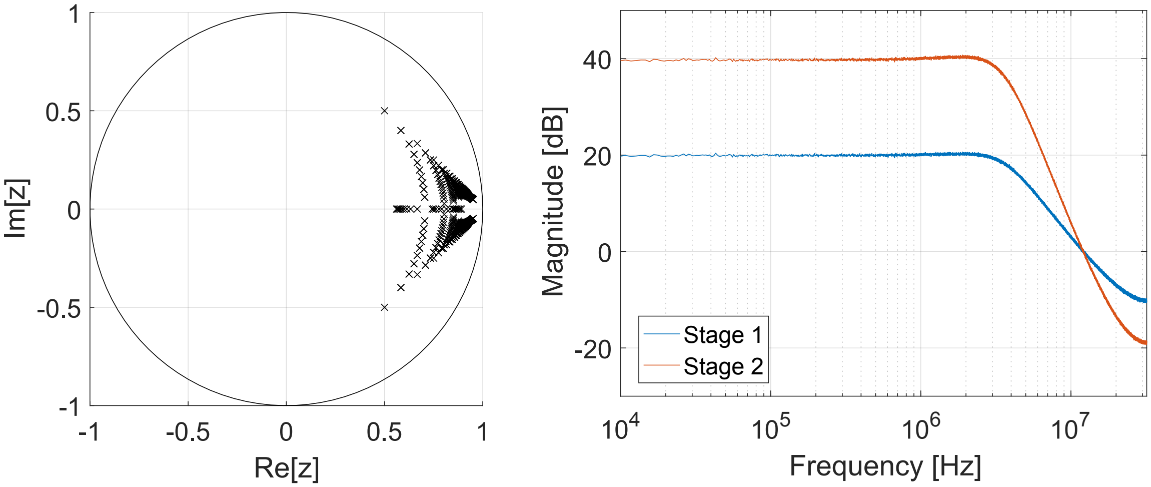


Figure 3.13: Left: Possible Z-plane pole locations for the topology in Figure 3.12 assuming integer capacitor ratios. Right: Example transfer functions for a cascade of two filter stages implementing a fourth order filter.

by raising gm at the cost of more power or lowering the sample rate at the detriment of aliasing performance. It is thus desirable to use as small of a value for C_s as feasible to achieve the required amount of gain. Output capacitance of the transconductor appears in parallel with the C_1 capacitor and thus impacts the transfer function if these values are of similar magnitude. Therefore it is desirable to construct an amplifier with as high output impedance as possible while minimizing the size of its output parasitic capacitance. Many of the implementations of this type of filter in the previous referenced works use inverter based transconductors due to their efficient use of current and amenability to process scaling. This is possible in a high performance design that burns considerable power and thus is not constrained to using very small capacitor values. For scaling down the performance to the level required by the SCM receiver however, it is difficult to achieve the required output characteristics using an inverter based amplifier.

For these reasons a fully differential folded cascode topology with small output devices is used to implement the transconductor. A schematic of the amplifier is shown on the left in Figure 3.14 along with the common mode feedback circuit on the right used to set the output DC level to mid-rail. A PMOS input pair is used so the input can be DC biased to ground to ease interfacing between stages and AC coupling between stages prevents DC offsets from saturating the following amplifiers. The input branch is biased at a low overdrive voltage with 4 μA of current to give each input device a gm of 40 μS. The constant-gm bias and startup circuits are shown in Figure 3.15. In order to change the DC gain of the filter the resistance in the bias circuit is adjusted to change the gm of the amplifier and thus the gain of the filter passband. This implementation of gain control was chosen over the more common approach of using a DAC for choosing the size of the sampling capacitor in order to minimize the parasitics at the critical output node of the filter. A drawback to this approach is that changes in gain cause bias point changes in the amplifier which then require time for the common mode feedback to readjust. The startup circuit is implemented by placing a FET between the two nodes of the constant gm bias and capacitively coupling its gate to the supply rail. During a startup transient the gate of this FET will be pulled high ensuring that current begins to flow in the bias circuit which in turn causes the startup FET to be turned off. The sampling capacitors in the switching network are kept at a fixed size to limit parasitics as previously mentioned. Multiple sampling capacitors are used to pipeline the design so that a new output is available at every cycle to produce a full-rate output similar to [40]. Some degree of tunability is instead built into the C_1 and C_2 capacitors to allow for tuning of the transfer function. These capacitor DACs are implemented differentially using the schematic shown in Figure 3.16.

3.3 ADC

The last step in the analog signal path is to digitize the received waveform before entering the digital baseband. Based on the analysis in Chapter 2 this design implements 4-bit ADCs clocked at a rate of 16 MHz. Flash and SAR are both reasonable architectural choices

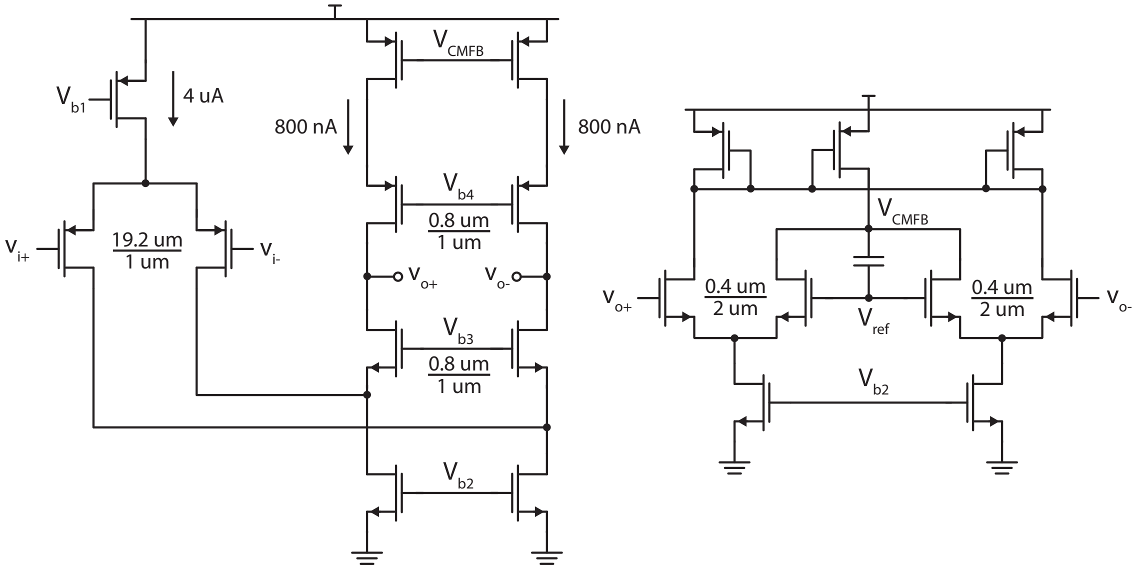


Figure 3.14: Left: A folded cascode amplifier is used to achieve high output impedance for the transconductor. Right: The implementation of the common-mode feedback used for fully differential operation.

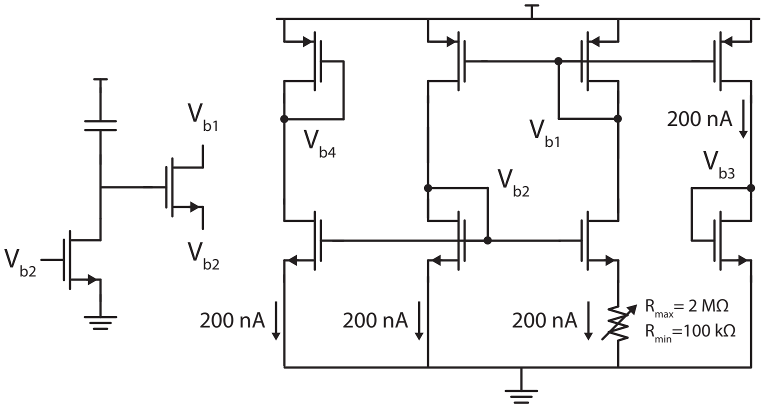


Figure 3.15: A constant gm circuit is used to bias the transconductor. Since the filter gain is proportional to *gm* the gain is adjusted with a variable resistor in the bias circuit. The startup circuit uses a switch connected between the two nodes of the bias circuit with its gate capacitively coupled to the supply rail.

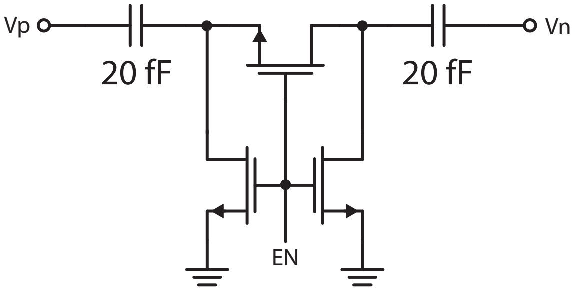


Figure 3.16: The programmable capacitor array used in the filter are implemented with differential unit cells activated by switches. Additional pulldown devices are used to prevent large swings at the output nodes from affecting the DC bias of the center switch.

for implementing an ADC with these specifications. It was previously established that the system level goal was to utilize a small full-scale range for the ADC to reduce gain and linearity requirements of the preceding stages. For a flash architecture this implies that the comparators will need to have small offset voltages which results in larger layout area and power. It was largely this reason that drove the choice of a SAR architecture for implementation of the ADCs. In order to drive the capacitor DAC of the SAR without affecting the transfer function of the main filter, a third copy of the individual filter stage was used to serve as an ADC driver.

To match the 64 MHz output rate of the filter to the ADC rate the signal is downsampled by a factor of four. A synchronous SAR ADC that is making four bit evaluations at an output rate of 16 MHz must operate at a minimum clock rate of 64 MHz. Conveniently this means the filter clock can also be used to drive the ADC, but that does not provide any time window for loading or resetting the SAR DAC. To overcome this issue multiple DACs were used in a time-interleaved manner so that while one DAC was being evaluated by the SAR logic others could be conducting the loading and reset operations as shown in Figure 3.17. Due to the low 4-bit resolution of the ADC there are no significant issues caused by interleaving the DACs given the matching level that can be achieved with proper DAC layout. This approach allows for a relatively simple way to match the filter and ADC rates with low complexity in the clock generation network.

The SAR ADC itself is based on the implementation in [46] and Figure 3.18 is a schematic of the core of the ADC with one capacitor DAC shown. The DACs rotate through loading (LD), evaluating (EVL), and reset (RST) phases. A fourth idle state is also inserted after evaluation to simplify the clocking requirements. A common mode bias was added for the comparator due to the small full-scale reference voltage as it was found that larger overdrive was required for the input pair to operate correctly. The SAR FSM implementation is shown in Figure 3.20 along with the clock generation in Figure 3.19. Generation of the signals for controlling the DAC interleaving are discussed in the next section.

A schematic of the Strongarm comparator used in the ADC is shown in Figure 3.21.

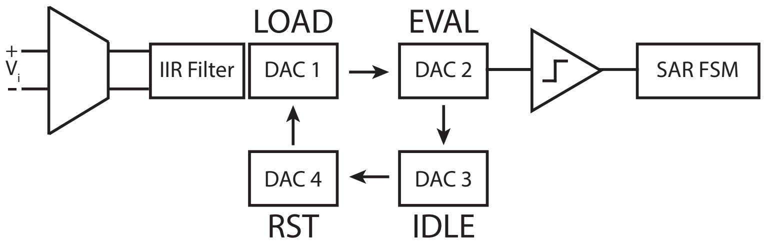


Figure 3.17: Four capacitor DACs are used in the ADC which rotate between sampling, evaluating, resetting, and idle phases.

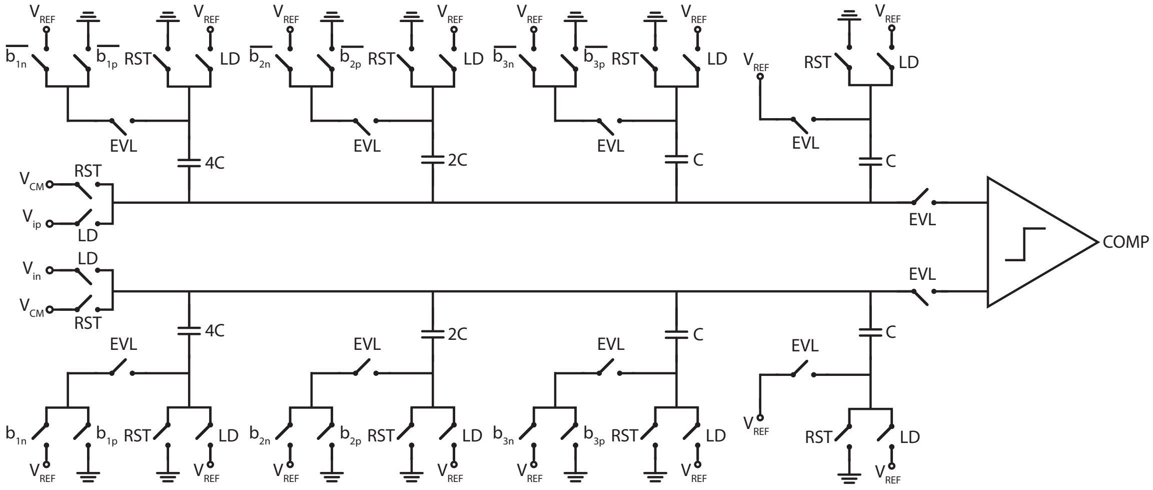


Figure 3.18: Schematic of the core ADC showing one capacitor DAC with both interleaving and SAR FSM control connections.

The comparator output is followed by a SR latch to shield the following logic from the reset phase of the comparator. This same comparator is used to support zero crossing counter demodulator mode in the receiver. To switch to that configuration the SAR DAC is bypassed and the signal is routed directly to the comparator inputs. The comparator then operates at the same clock rate as the filter producing a full-rate output waveform. Since the duty cycle of the IF waveform is very important for zero crossing mode, offset trim is built into the comparator with the adjustable capacitor DACs C_p and C_n. These DACs are implemented using small custom fringing capacitor arrays. Any threshold mismatch in the M1 transistors can be counteracted by increasing the size of the capacitive load that the stronger device must discharge.

The ADC requires two voltage references, one for the full-scale reference and one for the comparator common mode bias. Both of these references are derived using switched capacitor voltage dividers which are then buffered by amplifiers in unity gain feedback. The buffer amplifier uses the same folded cascode implementation as the filter amplifiers but has

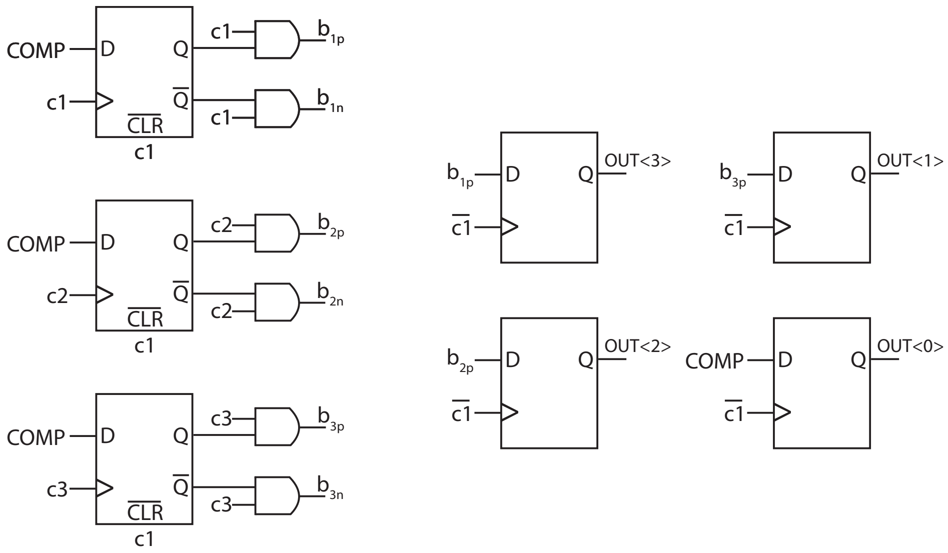


Figure 3.19: The finite state machine that derives the logic outputs for sequential switching of the capacitor DAC to perform the analog to digital conversion.

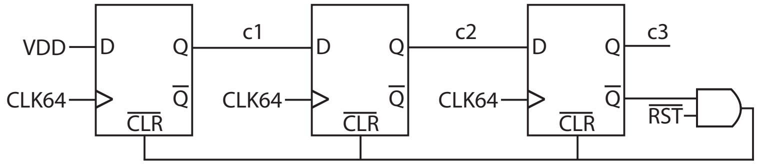


Figure 3.20: Clock generation for the ADC FSM.

a single ended output rather than fully differential. A schematic of the reference generation is shown in Figure 3.22. The divider is implemented with the switched cap resistors formed by C1 and C3 which generate the desired voltage on C2. The same topology is used to generate both required reference voltages by adjusting the value of C3. A low pass filter formed by M1 and C4 filters the ripple and the output is buffered to drive the ADC DAC.

3.4 Clocking

A diagram showing the required clock phases for driving the filters and ADC is shown in Figure 3.23. The 4x interleaving in the filter requires four phases offset by one fourth of the overall period. Similarly the ADC uses 4x interleaving but due to the decimated sample rate operates at 4x lower frequency. These clock phases are generated from a single master 64 MHz clock input which is fed to the flip flop based shift register shown in Figure 3.24. A POR signal is used to initialize the shift registers to ‘1000’ at power on. Each successive

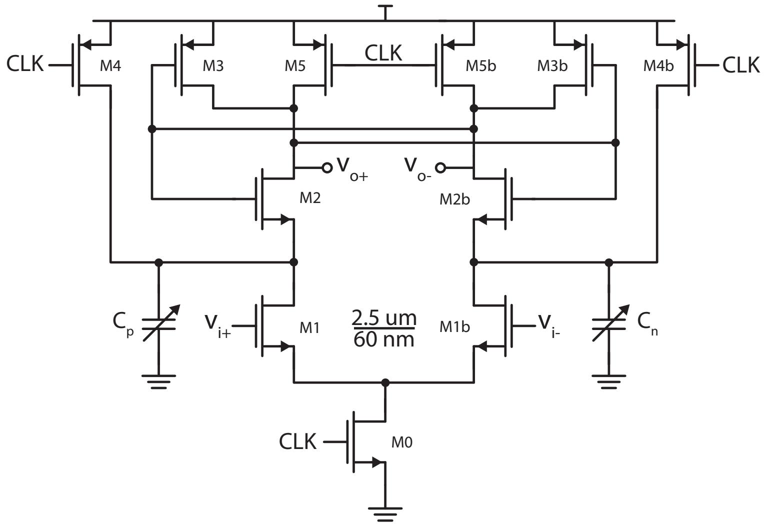


Figure 3.21: The comparator used in the ADC is a Strongarm topology with offset cancellation implemented with variable capacitor DACs attached to the drains of the input devices.

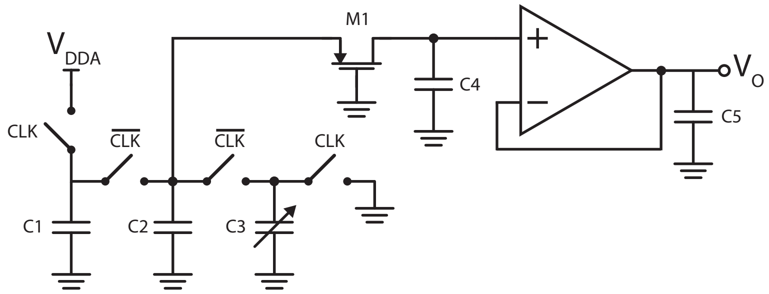


Figure 3.22: Switched capacitor voltage divider used to generate reference voltages for the ADC.

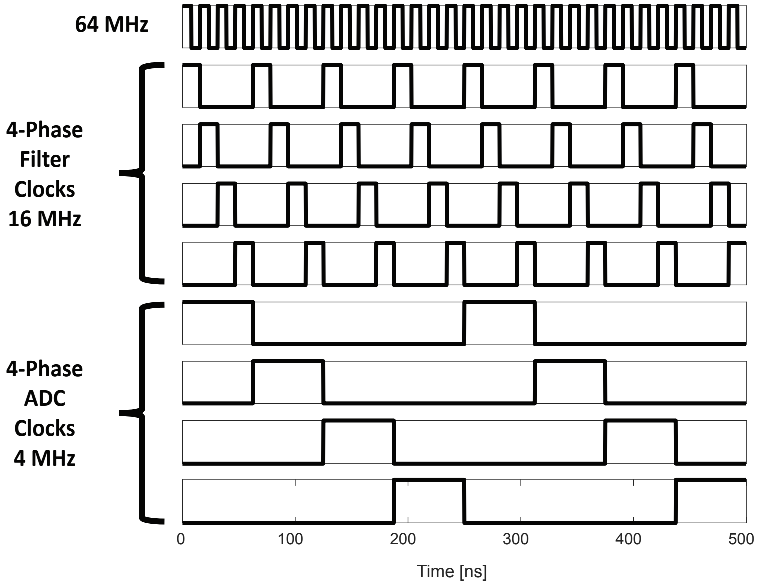


Figure 3.23: Timing diagram for the filter and ADC interleaving generated from a single master 64 MHz clock.

edge of the 64 MHz input then causes the ‘1’ to cycle through the register and produce the required phases.

Three possible clock source options were included to provide the required 64 MHz input: an on-chip RC oscillator, a divided version of the LC LO, and an external pad for testing. While the divided LC is the highest quality clock available on-chip it suffers from two main drawbacks. First the LO divider must be activated during reception in order to run the filter and ADC which incurs a significant power penalty. The second drawback is that the LO frequency must change for different channels while the 64 MHz requirement remains constant. This results in varying frequency error based on the current channel and limits the precision with which the required 64 MHz clock can be derived. Thus the on-chip RC oscillator is intended for use as the primary clock source for the receiver.

The on-chip 64 MHz RC oscillator is implemented using a replica of [47]. The schematic and resistor DAC tuning are shown in Figure 3.25. At low frequencies the output impedance of the inverters is much smaller than the resistor value that sets the time constant of oscillation. However when the resistor and capacitor values are reduced to scale this topology into the 10s of MHz range the inverters’ output impedance begins to become more critical. In order to continue to have the time constant set by the passive component values the inverter driving node Va should have a large drive strength. This was achieved by using low-Vt devices and raising the supply voltage. The original topology utilized a supply regulator which generated a lower local supply voltage that tracked threshold variation in order to reduce temperature sensitivity. This local regular was omitted in order to utilize the maximum available supply voltage. It is expected that periodic calibration is going to be necessary for

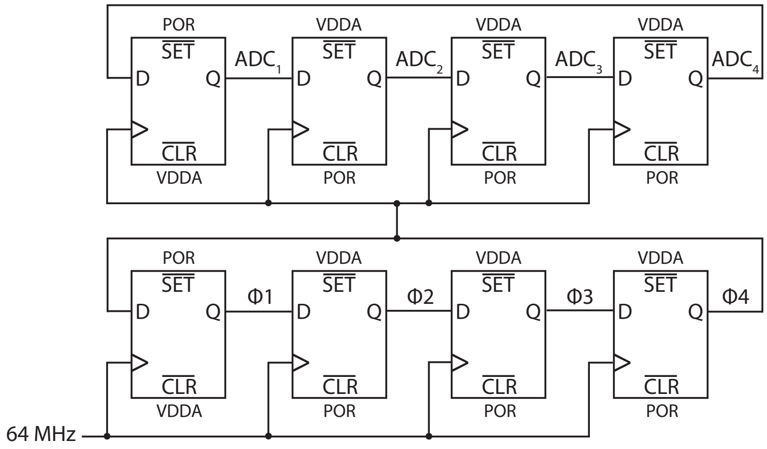


Figure 3.24: Circuit used to generate the waveforms in Figure 3.23. A power-on-reset signal is used to initialize the flip-flops to the correct state.

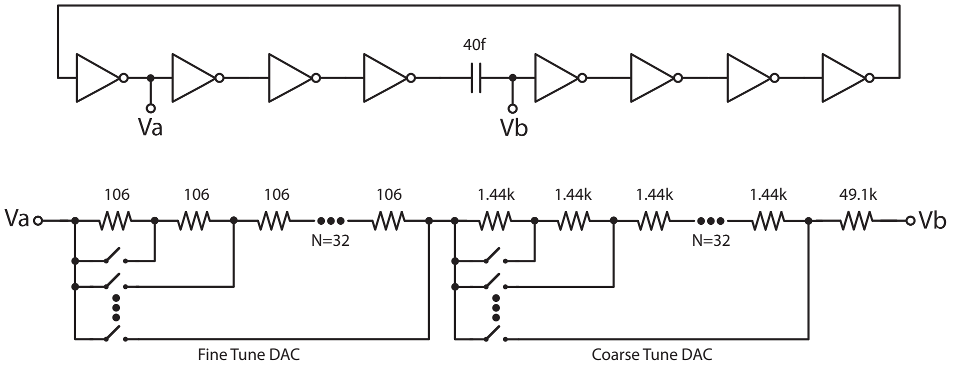


Figure 3.25: The RC oscillator used to generate the 64 MHz master clock is based on [47]. Frequency tuning is implemented with coarse and fine resistor DACs.

the crystal-free system as a whole so some level of temperature sensitivity was tolerated at the circuit level. Note that from a software standpoint it is desirable that any temperature dependence in the clocks be as linear as possible to ease calibration.

Frequency tuning is implemented with a resistor DAC containing 5 bits of coarse and 5 bits of fine adjustment. The tuning resistors are connected to node Va in order from smallest to largest. This minimizes the variation in V_{gs} experienced by the pass gate switches due to IR drop. The DACs are physically implemented using a thermometer structure and then separate digital binary to thermometer decoders were used for coarse and fine.

3.5 Supply Regulation

The receiver operates from its own independent LDO, the schematic of which is shown in Figure 3.26. The LDO is constructed as a basic two stage opamp operating from the battery voltage domain. The reference voltage for the LDO is generated by driving a current from a bandgap circuit through a variable resistor (see [17] for further details).

When designing for stability in the LDO there are two options: either the dominant pole is placed at the output node or at the stage one output. The primary difference between the two choices is the transfer function from the battery voltage to the regulated output. Since the battery is shared with other circuits on the chip, including the entire digital system, it is desirable to achieve as much rejection as possible to avoid introducing spurious frequency components into the signal path. The typical solution for stabilizing a two stage opamp would be to use Miller compensation to split the first and second stage poles and make the stage one output dominant. This however introduces a zero in the V_o/V_{bat} transfer function which can degrade the supply rejection as shown in Figure 3.27 [48]. An intuitive way to think about this issue is that at frequencies above the pole of the first stage the gate of M3 begins to look like a virtual ground. This then causes M3 to behave like a common gate amplifier for any stimulus on the battery node.

The alternative of making the LDO output pole dominant is also not without issues. Depending on how much current is being drawn the output impedance of the LDO may be quite low which will require a large capacitor to make the output pole dominant. This makes it desirable to have the first stage pole at the highest frequency possible to ease the capacitance requirements at the output. Since the rejection of external signals on the battery node was of primary concern (particularly those from digital switching) an output pole dominant LDO was chosen. Using a small bias current in the first stage to save power results in small transistor sizes which will then contribute significant mismatch and affect the accuracy of the LDO output voltage. In order to make the amplifier stable 100 pF of decoupling capacitors were distributed throughout the layout to provide enough load capacitance.

To enable flexibility in controlling the turn-on of the various LDOs in the transceiver, several control options were implemented. The primary method for testing and debugging was to utilize a scan chain to individually control LDOs. This method however is very slow as bits must be serially clocked by the microprocessor into a shift register in excess of 1000 bits long. To enable faster control for TSCH operations turn on signals were also connected to memory mapped registers and to signals that allowed the hardware radio FSM controllers to directly activate the transceiver. A GP input to the chip gave yet another option for activating the radio. The choice of which control signal to use is configured via scan chain, but only needs to be setup once so speed is not an issue.

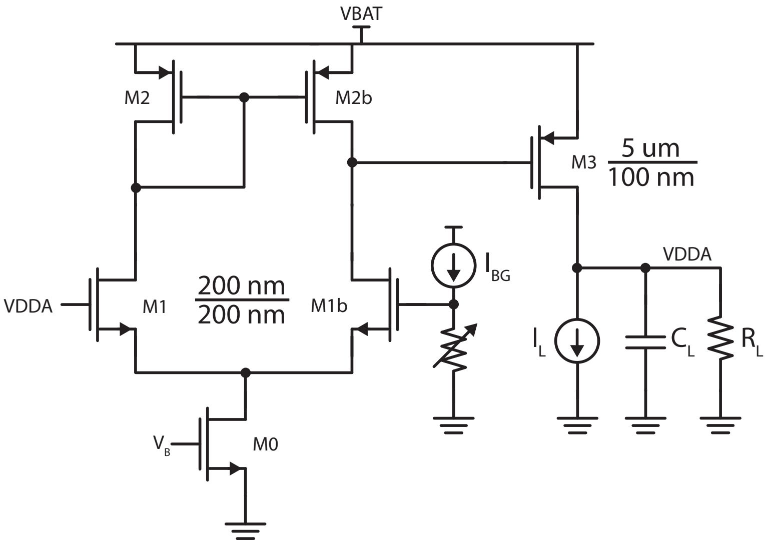


Figure 3.26: Schematic of the LDO voltage regulator used to generate the supply for the receiver. A current from a bandgap is routed through a variable resistor to implement a programmable output voltage [17].

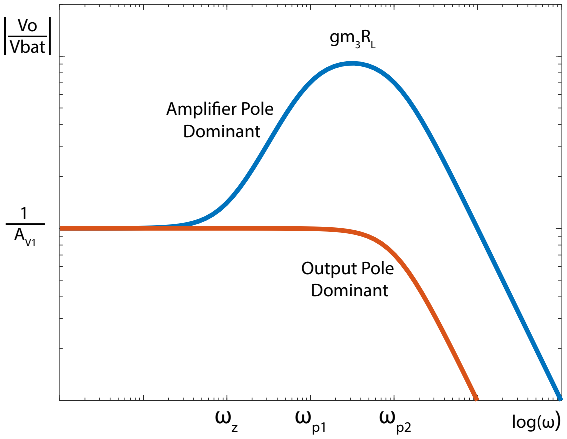


Figure 3.27: The magnitude of the transfer function from the battery supply to the regulated output depends on which pole is dominant in a two stage LDO.

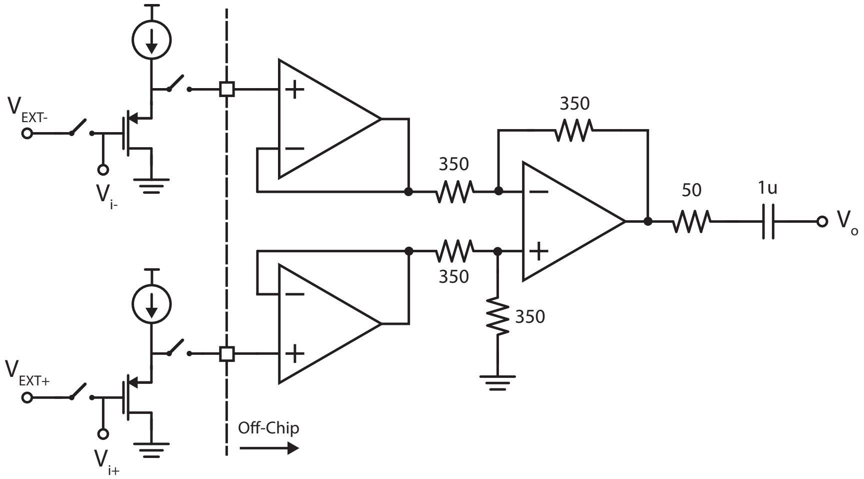


Figure 3.28: The analog test harness used to observe internal nodes and provide external stimulus to the chip. On-chip nodes are buffered with source followers to provide a pseudo-differential output which then attaches to an off-chip instrumentation amplifier to drive test equipment.

3.6 Analog Test Harness

To facilitate testing and debugging of the analog portion of the receiver several access points were built into the design. These access points were located at the output of the mixer, the output of the TIA, and at the outputs of the first and second filter stages. This debug harness provided the ability to both monitor signals at internal nodes as well as to independently provide an external stimulus. Critical internal analog nodes were isolated using source followers to buffer signals before driving them pseudo-differentially to pads. Signals were multiplexed between the various access points to reduce the number of pads required for debugging. The same debug harness was used for both I and Q channels to enable simultaneous visibility. This resulted in each channel requiring two differential input pads and two output pads for a total of eight overall debug pads. The conversion to single ended signals for interfacing to test equipment was accomplished using PCB components. Three ADA4817 opamps were used to construct an instrumentation amplifier that was used to drive 50Ω test equipment loads. Single ended input stimuli were converted to differential inputs using LTC6406 amplifiers. A schematic of the debug observation path is shown in Figure 3.28.

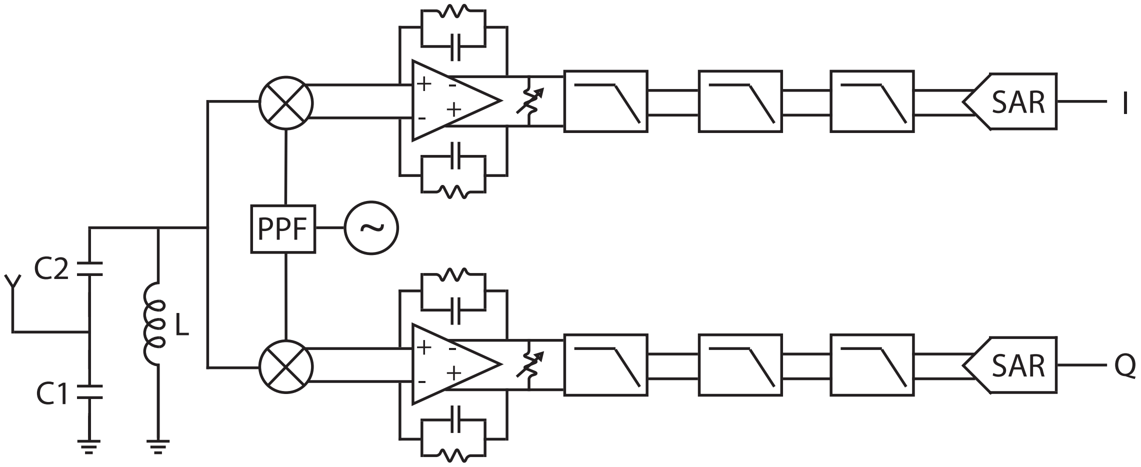


Figure 3.29: Block diagram of the full receiver from antenna through digital outputs.

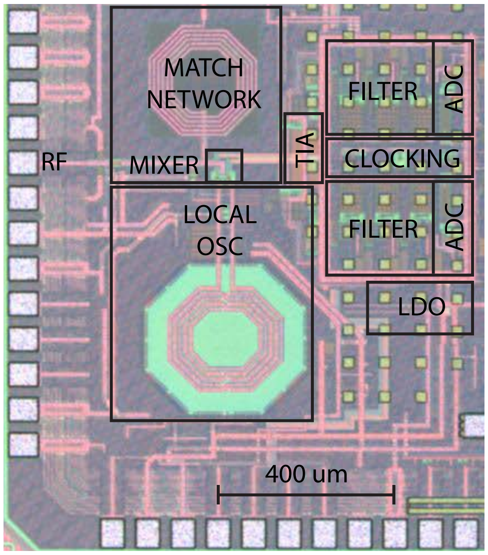


Figure 3.30: Die photo of the transceiver showing the layout floorplan of the RF frontend, I/Q signal paths, and local oscillator.

3.7 Full Receiver

A block diagram of the complete analog portion of the receiver is shown in Figure 3.29. The design exploits as much symmetry and reuse as possible to minimize the layout effort. The filter stages are identical copies and I/Q branches are mirrored images of one another. The layout floorplan of the analog portion of the full transceiver is shown in Figure 3.30.

Chapter 4

Digital Baseband Design

This chapter outlines the block level components used to construct the 802.15.4 digital baseband implemented in the Single Chip Mote. This portion of the design consists of everything between the receiver ADC outputs and packet storage in system SRAM. Further details on the Cortex-M0 microprocessor, the packet assembly/disassembly, and the radio finite state machines can be found in [14].

4.1 Demodulation

The primary method used in the Single Chip Mote to extract modulated information from the incoming IF signal is the matched filter as described in Section 7.7 of [19] and shown in Figure 4.1. The optimal detector for a non-phase-coherent FSK receiver is shown to be implemented by correlating the incoming signal samples with quadrature templates representing the two tones that are transmitted. The time domain correlations effectively bandpass filter the signal using filters centered at the high and low FSK frequencies. The use of quadrature templates accounts for the lack of information about the phase of the input signal. By comparing the magnitude at the output of the correlations the receiver is able to estimate the digital value of the current received chip. An example transient of the matched filter operation is shown in Figure 4.2.

In an ideal matched filter implementation the two FSK tones are orthogonal to one another so that one correlation is zero while the other is maximized. Since this matched filter will be utilized in a low-IF receiver the choice of frequency for the IF will directly affect the difference in magnitude of the correlations. Given the restriction to 1 MHz tone spacing dictated by the requirements of 802.15.4 there is no choice of IF that results in orthogonality.

An alternative demodulation method is also implemented in the Single Chip Mote. This secondary method relies on a high speed clock to measure the time between subsequent zero crossings in the down-converted waveform. By doing so it is able to discriminate between the two FSK tones and is thus referred to as the zero crossing counter demodulator. This demodulator path is discussed further in Section 4.10.

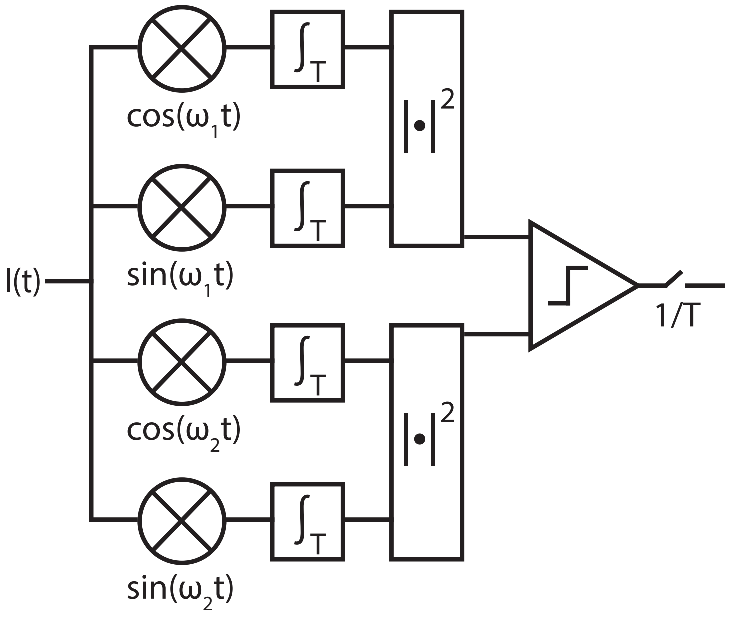


Figure 4.1: Block diagram of the non-coherent matched filter demodulator.

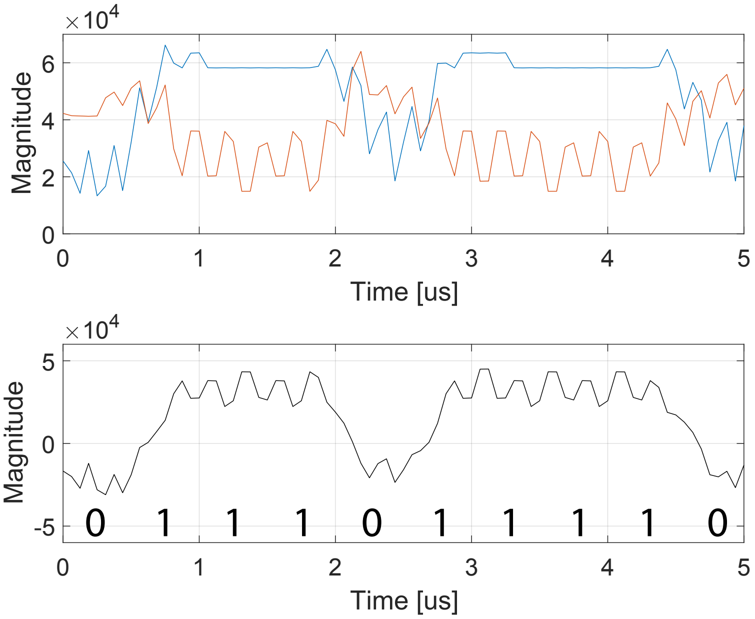


Figure 4.2: Example transient waveforms of the matched filter output showing the result of correlating with the two templates (top) and the difference between the two (bottom) which is used for making data decisions.

4.2 Clock and Data Recovery

Once the signal has been demodulated the receiver must still choose when to evaluate the demodulator output to make a chip decision. The ideal sampling instant is in the middle of a chip duration as the correlation values will be at their extrema. Any deviation from this ideal point results in a loss of SNR which will degrade performance. The purpose of the clock and data recovery (CDR) module is to attempt to locate this ideal sampling point as accurately as possible and track it throughout reception. A crystal-free receiver faces the additional burden of needing to be able to accomplish this task while tolerating much larger frequency errors in the chip rate than found in typical receivers. The CDR has no way to distinguish between rate errors which originate in the transmitter versus the receiver itself. Regardless of the source of rate inaccuracy the CDR must determine whether the incoming chip stream is arriving faster or slower than expected and adjust accordingly.

The timing recovery algorithm implemented in the Single Chip Mote is based on the timing error detector introduced in Section 9.4 of [49] which is derived from [50]. This method is based on a fourth order non-linearity targeted at continuous phase modulated signals, specifically MSK. The expression for the timing error detector is given in Equation 4.1. The chip period is represented by T, T_s is the sampling period, k is the index of the current sample, x is the series of complex valued input samples, and τ is the current estimate of the correct sampling phase. The four samples used to calculate the error estimate are obtained one sample before and after the start and end of the current estimate of chip period. There is no intuitive explanation of why this equation implements a suitable MSK timing error detector, so [49] resorts to plotting its transfer function to explain its behavior. The response of the error detector to a MSK signal for various sampling points is shown in Figure 4.3 assuming eight samples per chip as is the case in the SCM implementation. Having more samples per chip allows for finer adjustment of the sampling point but also has an effect on the shape of the detector curve [49]. The detector output can be seen to have two zero crossing locations, but they have opposite slopes. The error detector is enclosed in a feedback loop which is designed to make the stable operating point the one which samples in the middle of the chip period.

$$\begin{aligned} e(k) = {} & Re\{x^2(kT - T_s + \tau_{k-1})x^{*2}((k-1)T - T_s + \tau_{k-2})\} \\ & - Re\{x^2(kT + T_s + \tau_k)x^{*2}((k-1)T + T_s + \tau_{k-1})\} \end{aligned} \tag{4.1}$$

The implementation of the feedback loop around the error detector is shown in Figure 4.4. The first piece of the CDR is a shift register that holds 11 I and Q samples comprising one chip duration in addition to the samples before and after the chip period as required by the error detector. These four complex-valued samples are fed into the timing error detector to produce an estimate of how well the samples in the shift register actually align to the temporal boundaries of a chip. This error estimate is then scaled and accumulated in a high resolution register. Since the smallest unit of adjustment that can be made to the sample point is a single sample, the accumulated value is scaled in reference to the step

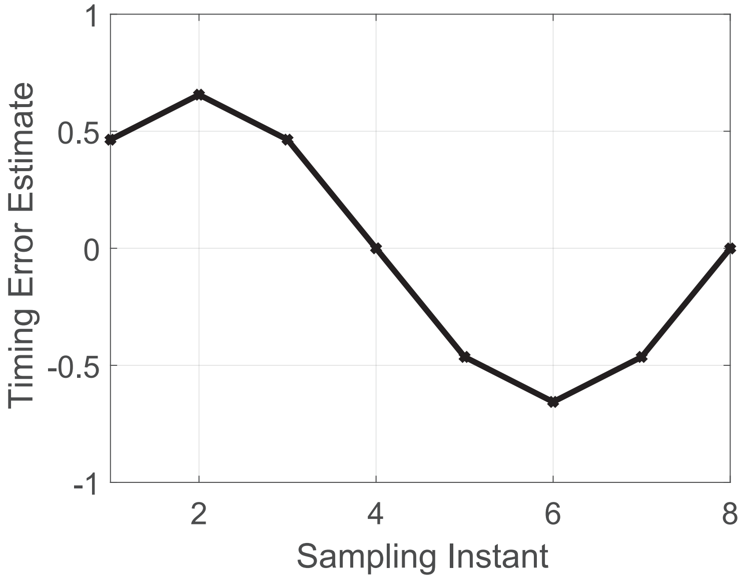


Figure 4.3: Response of the timing error detector to a MSK signal with eight samples per symbol for each choice of chip sample point.

size of a single sample point adjustment. This produces the value labeled τ in Figure 4.4 which corresponds to the CDR's current estimate of what the appropriate sample point is. Every time a new chip has been clocked into the shift register this process is repeated and τ is updated. If there is no perceived frequency error in the incoming chip rate then τ will settle to a constant value as shown in Figure 4.5 (some residual chattering around the ideal sample point is inevitable due to noise). After the initial transient the chip duration has been correctly aligned in the shift register. The CDR will then continuously clock in eight new samples before reevaluating the timing error. With no frequency error present, each new set of eight samples correctly aligns with the next chip duration.

In the more realistic scenario where there is a chip rate discrepancy then the error detector will respond with a linear drift in τ as shown in Figure 4.6 which was generated with a 1000 ppm chip rate error. The CDR compensates for the frequency error by altering the number of samples it clocks into the shift register before re-evaluating the timing error. It does this by taking the difference in the current and previous τ estimates and adjusting the number of samples clocked into the shift register by that amount. For example, if the difference between the current cycle's estimate of τ and the previous cycle's estimate differed by one, then the CDR would clock in nine new samples instead of eight before calculating another timing estimate and outputting a chip. This implementation was chosen specifically to tolerate large chip rate discrepancies on the order of thousands of ppm. An alternative solution that instead relies on clocking a fixed number of samples into a shift register and then updating an index pointer to the correct sample point suffers from rollover issues that limit its tracking range.

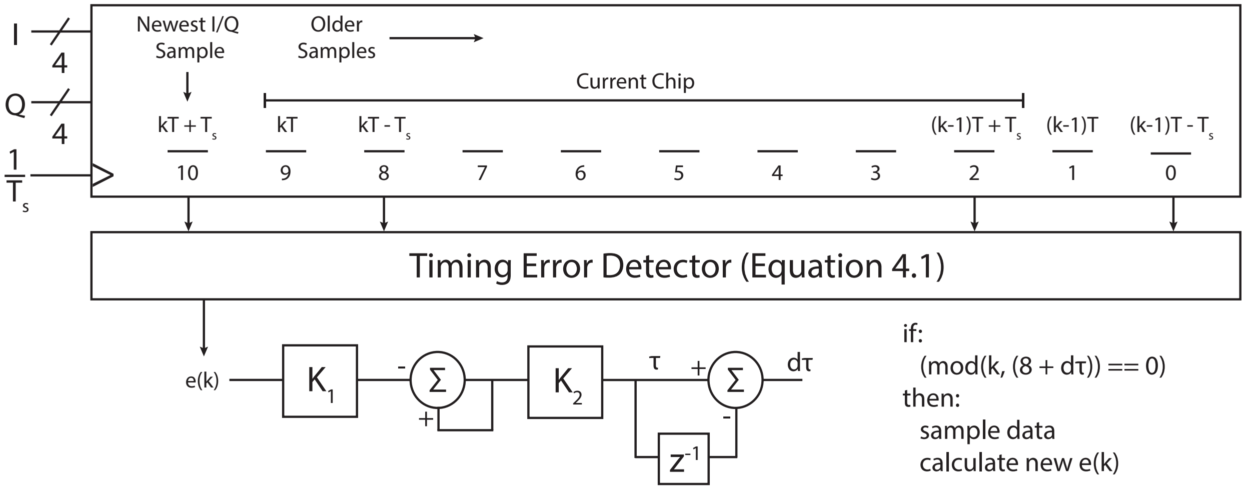


Figure 4.4: Block diagram of the CDR including the timing error detector and the feedback loop used to correct sampling phase and frequency. Chip rate adjustments are made by changing the number of samples clocked into the shift register before a chip decision is made.

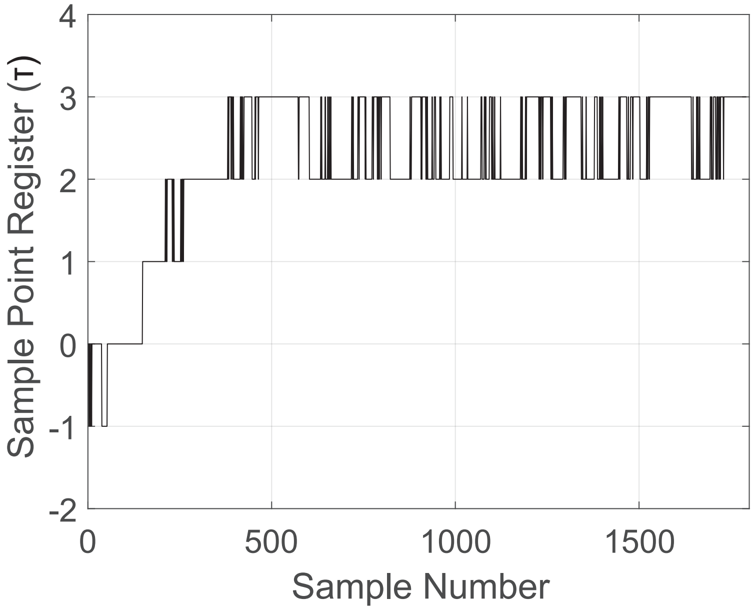


Figure 4.5: Transient behavior of the CDR as it settles to the appropriate sampling phase.

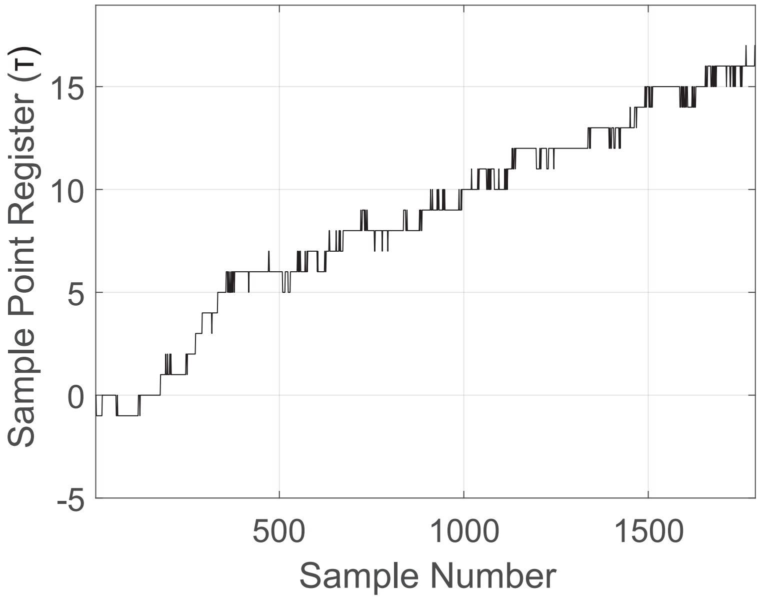


Figure 4.6: Transient behavior of the CDR in the presence of a 1000 ppm chip rate frequency error.

4.3 Complex Bandpass Filter

Image rejection in the Single Chip Mote is implemented using a digital complex bandpass FIR filter. This approach was chosen over implementation of image reject mixers in the analog domain in order to limit the complexity of the analog RF design. The filter also serves to set the noise bandwidth of the signal by filtering out ADC quantization noise. Some additional interference tolerance is also gained for signals that were not filtered out in the analog domain although the small dynamic range of the 4-bit ADCs limits the amount of additional rejection achievable.

The number of taps used in the filter affects the achievable filter profile while the quantization of the coefficients and internal registers limits the stop-band rejection. A large number of taps results in a more idealized filter shape but can influence the performance of the receiver through intersymbol interference (ISI). For the choice of 2.5 MHz as the IF for the receiver the image is located one 5 MHz channel away. The 802.15.4 standard specifies that the receiver should be able to tolerate an interferer signal of equal power on the adjacent channel. A target image attenuation of 20 dB was chosen to reduce the amplitude of the image channel to the point of having minimal effect on the desired signal. A filter length of eight taps was chosen for use on the Single Chip Mote since only a small number of coefficients is sufficient for the rather relaxed requirements needed. Furthermore this restricts the length of the filter to be equal to one chip duration and thus does not introduce significant ISI issues into the design process.

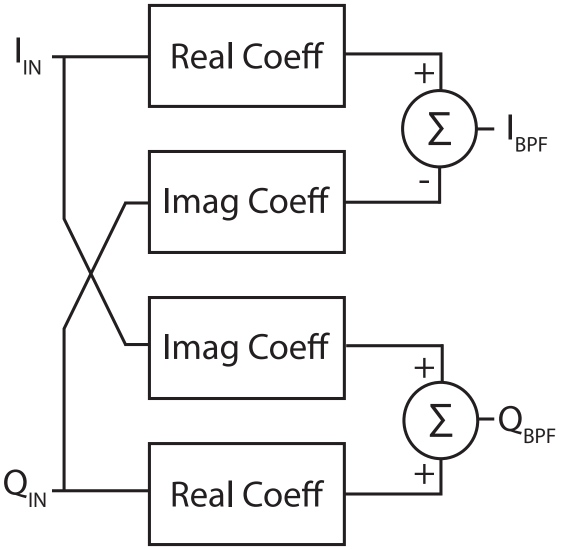


Figure 4.7: Structure for breaking down complex-valued filter computation into real-valued operations [51].

.

To achieve the targeted 20 dB stop-band rejection an eight tap FIR filter was designed using MATLAB and quantized to 5-bit coefficients. The coefficients are the same as those used for the example filter shown in Figure 2.15 in Section 2.3 along with the ideal frequency response in Figure 2.16. In order to implement the complex filter using only real valued coefficients and computations the structure shown in Figure 4.7 was utilized. The measured filter response is shown in Figure 4.8 which was obtained by generating a sequence of 4-bit I and Q samples consisting of white noise, injecting those into an FPGA implementing the filter in Verilog, and then taking the complex FFT of the output bit stream. Multiple FFTs were averaged to more accurately reveal the shape of the filter transfer which can be seen to have a passband at the desired 2.5 MHz IF with close to 20 dB rejection at the image location.

4.4 I/Q Mismatch Correction

The impact of mismatch in the quadrature receive paths is dependent on the mechanism used to conduct the image rejection. The target level of image rejection is also a consideration as very high rejection ratios require much more accurate matching. Rather than attempting to address mismatch in the RF frontend where it originates, an alternative approach is to apply a correction in the digital domain. This approach eases the design of the polyphase filter interface and moves the burden to a portion of the design that does not require time consuming RF simulations to verify. The correction algorithm used here is given in [18] where

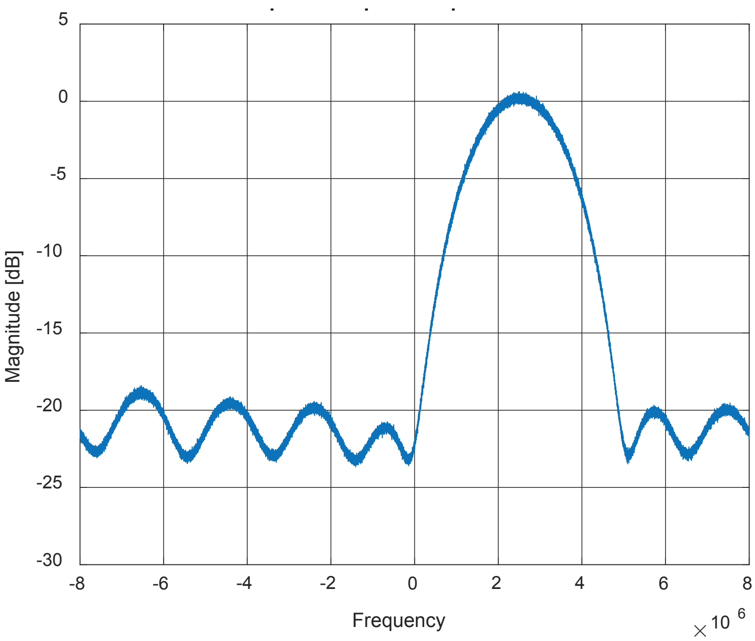


Figure 4.8: Response of complex bandpass filter after implementation in Verilog on a FPGA. The filter is implemented using the coefficients first shown in Figure 2.15 which has the unquantized response shown in Figure 2.16.

a simple 1-tap FIR filter in a feedback loop is used to correct for imbalances. The input signal is transformed as shown in Equation 4.2 and the FIR filter is updated according to Equation 4.3. The parameter M determines the settling time and stability of the compensation loop. The reader is referred to [18] for the mathematical details of the compensation algorithm.

$$y(n) = x(n) + w(n) * x^*(n) \tag{4.2}$$

$$w(n+1) = w(n) - M * y^2(n) \tag{4.3}$$

For the complex bandpass filter introduced in Section 4.3 the impact of mismatch can be evaluated by injecting non-ideal signals into the filter in MATLAB. For this task the image rejection ratio is defined as the difference in magnitude between the desired and image signals at the I and Q outputs of the filter. Since the amplitude difference can vary between the I and Q channels depending on the input parameters such as signal phase, the worst case between I and Q is taken here. The result of sweeping mismatch magnitude and phase with and without IQ correction can be seen in Figures 4.9 and 4.10 respectively. The result indicates that there is a significantly larger penalty for uncorrected amplitude mismatches than there is for phase mismatch when this complex bandpass filter is used for image rejection.

Measurements from early generations of SCM hardware indicated that I/Q mismatch was not substantial enough to cause serious issues. Therefore the I/Q correction block was

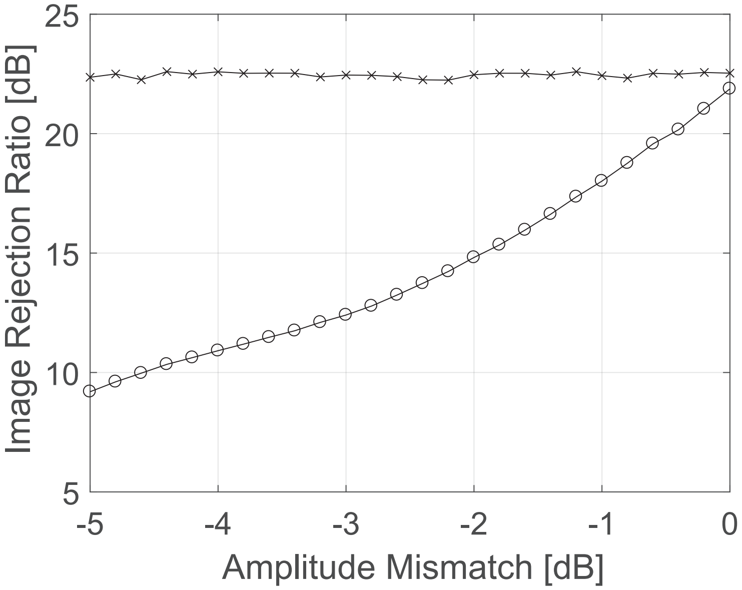


Figure 4.9: The effect of quadrature amplitude mismatch on image rejection for the complex bandpass filter specified in Section 4.3.

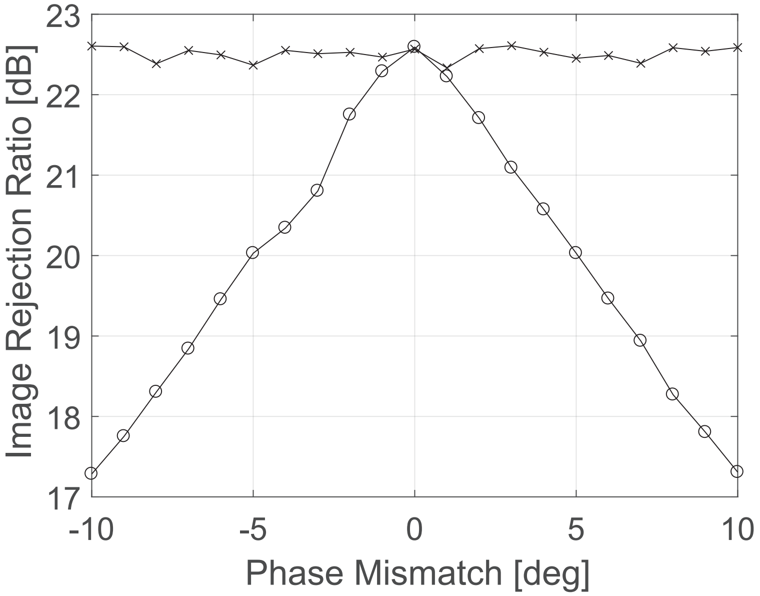


Figure 4.10: The effect of quadrature phase mismatch on image rejection for the complex bandpass filter specified in Section 4.3.

omitted from subsequent tapeouts to reduce complexity and potential failure points. The simulated results here only consider the changes in amplitude produced by the correction algorithm. If the I/Q compensator is to be implemented in a full system then its impact on packet error rate must also be considered. This requires that detailed modeling be done for the compensator's response to modulated signals, transient settling behavior, and quantization issues.

4.5 Intermediate Frequency Estimation

A major challenge in the operation of a crystal-free radio is the frequency tuning of the RF oscillator across varying environmental conditions. The ability to share timing information among nodes in the network is critical in order to maintain links as nodes experience different drifts due to variations in temperature and supply voltage. One of the primary mechanisms that the Single Chip Mote uses to derive timing information from network traffic is by estimating the error in the average value of the down-converted intermediate frequency. This estimate is made by observing the I channel samples from the complex bandpass filter and counting how many times the signal crosses zero in a 100 μs period. The IF estimator block continuously updates this measurement value every 100 μs until the radio state machine indicates that packet reception is complete. At this point the estimator freezes the value and sets a flag indicating its validity. The microprocessor can then read this value via memory mapped register and use it to make decisions about whether an update to the RF frequency setting is required.

The duration of the 100 μs time period is defined in relation to the receiver ADC clock rate; i.e. the IF estimator counts for 1600 ticks of the 16 MHz clock. This clock is generated from a free-running RC oscillator and is not perfectly accurate which will introduce some error into the IF estimate. However, the receiver chip clock is also derived from the receiver ADC clock so it must already be calibrated to within a few thousand ppm in order for the receiver to work at all. Thus the accuracy requirement for the chip clock rate directly translates to a bound on the error in the 100 μs measurement interval. For example, if there is a 1000 ppm error in the receiver clock rate that translates to an error of 100 ns in the estimate of a 100 μs time duration. For an average intermediate frequency of 2.5 MHz this 1000 ppm error in measurement interval would introduce an insignificant error of 2.5 kHz into the IF estimate.

4.6 DSSS Despreading

The next step after chip decisions have been made is to reverse the 802.15.4 Direct Sequence Spread Spectrum coding. The transmitter sends only 16 possible sequences based on what the current data symbol is. The receiver should correlate the incoming chip sequence against these 16 possibilities and choose the one that is the best match. In the case where each chip has already been evaluated as a digital one or zero this process is referred to as hard decisions. This is the implementation used in [14] and Hamming distance is used as the decision metric. This value is simply the number of chip inversions and can be generated by XOR'ing the input chip stream with each possible sequence and counting the number of resulting ones. Whichever symbol has the lowest Hamming distance is the one that was most likely transmitted.

The implementation of the OQPSK-HSS modulation of 802.15.4 as MSK leads to a requirement for a conversion before de-spreading can occur. The chip sequences listed in the 802.15.4 standard definition are to be used when the radio is traditionally implemented with I and Q data streams. As discussed in Section 1.4 the conversion to an equivalent MSK implementation simplifies the overall transceiver design. Demodulating the signal as MSK means that the receiver should use the converted chip sequences for correlation rather than the ones specified in the standard document. A consequence of this conversion process is that the last chip of each converted 32-bit sequence is dependent on the first chip of the next sequence. Since this last chip could be either a zero or a one it provides limited additional information for use during the correlation. It is for this reason that the Single Chip Mote correlator only utilizes 31 of the received chips for correlation and discards the last.

An alternative way to implement the correlator is using soft decisions. Here instead of assigning a digital value of one or zero for each chip prior to correlation, a multi-bit value is used. The Hamming distance calculation is then replaced by the Euclidean distance in determining the best symbol match. By adding this extra resolution to the bit decisions it adds additional probability information for making decisions. This information effectively tells the correlator how sure the demodulator is in a given chip estimate. If the quality of the estimate is rather poor and the resulting output value is very near the decision threshold then the demodulator is essentially guessing. With hard decisions there is no way to factor that information into the symbol estimate while soft decisions provides a means to do so. The theoretical improvement in sensitivity gained by moving from hard to soft decisions is approximately 2 dB for 3-bit demodulator outputs [20].

4.7 Packet Detection

Before the receiver can begin storing received data into memory it must first locate the start of the packet. The 802.15.4 packet format includes a preamble and start of frame delimiter (SFD) to aid in this process. The preamble consists of eight repetitions of the 0x0 symbol while the SFD is two symbols long and is transmitted as the value 0x7A. The packet detection implemented in [14] for the Single Chip Mote operates as follows:

- The correlator initially runs continuously and calculates a new Hamming distance to each symbol for every new chip.

- If the closest symbol match is 0x0 and the Hamming distance is less than a user defined threshold setting then investigate this as a possible packet, else keep evaluating every new chip.

- When a possible match is found, the state machine will wait until 32 new chips have arrived and then evaluate what symbol is the closest match to this new set of chips. It is this step that sets the alignment of symbols to the boundaries of the correlator.
- If this new symbol correlates to another 0x0 preamble symbol then the state machine continues to repeat this process. If instead the symbol matches to the first part of the SFD then the state machine will wait and check for the second SFD symbol. If both SFD symbols are found the receiver extracts the packet length from the next two symbols and proceeds to store the appropriate number of received bits into a FIFO.
- If at any stage of this packet investigation process the closest symbol match is neither a 0x0 preamble or the expected SFD symbols, then the state machine knows it has falsely investigated the start of a packet and returns to the initial search process.

An issue with this approach is the possibility of completely missing packets due to false detections. If a random chip sequence is close enough to a preamble symbol to have a Hamming distance within the threshold value the state machine will stop its search process and wait for the next symbol to be aligned in the correlator. If during this wait time the actual packet arrives it will potentially be missed unless the state machine resets in time and is able to trigger on a subsequent preamble symbol. One possible strategy to mitigate the impact of false detections is to continue to operate the correlator even while waiting for the next symbol to align with the correlator boundaries. If a better Hamming distance match occurs during the wait time then the receiver should likely investigate that match instead.

Another possible change to the detection algorithm is to limit reliance on the preamble and instead search only for the SFD. Depending on the specific implementation of a receiver there may be a significant amount of transient behavior when a packet first arrives as various control loops settle (such as automatic gain control, carrier tracking loops, clock and data recovery, etc). These transient events occur while the preamble is being received and the chip error rate is likely to be higher than after these loops have settled. This leads to a scenario where the first preamble symbols are likely to have more errors than the latter ones. With a poor choice of threshold setting this can increase the potential for false lock investigations leading to missed packets. The SFD-only approach can furthermore be extended by doing the packet search using both symbols of the SFD and doing a 64-bit correlation. This not only eliminates the variable chip error rate that occurs during the preamble symbols but also increases the certainty of frame alignment by checking across a wider timespan. This frame synchronization method was found to have a slight advantage over the original implementation and is used in all measurements in Chapter 5.

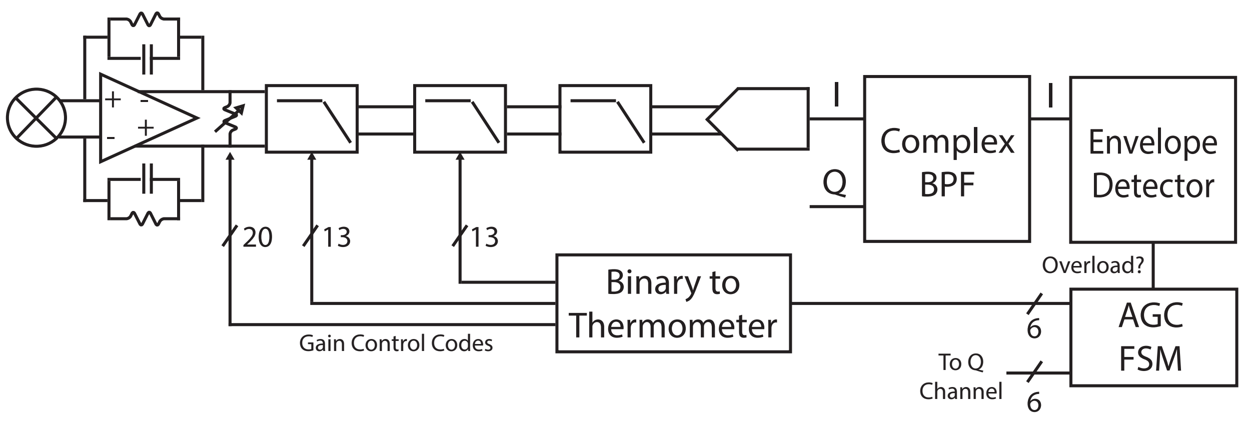


Figure 4.11: Block diagram of the automatic gain controller showing only the I channel for simplicity. The signal envelope at the output of the complex bandpass filter is monitored by the finite state machine which uses binary search to adjust the gain of the analog amplifiers.

4.8 Automatic Gain Control

The automatic gain control (AGC) algorithm chosen for the Single Chip Mote is based on binary search using a digital envelope detector. The envelope detector operates on the output of the complex bandpass filter and asserts a flag when the envelope of the signal exceeds a value the user can program via memory mapped register. A block diagram of the controller is shown in Figure 4.11. The envelope measurement is updated every 11 samples (which is approximately every 700 ns with a 16 MHz sample rate). Upon reset the controller begins at its maximum gain setting in order to be able to detect small signals. The controller will remain at maximum gain until the envelope detector indicates that the user threshold has been exceeded. At this point the controller will begin its binary search by first cutting the total receiver gain in half. The AGC then waits a user programmable amount of time before again checking the envelope detector output. The wait time is required to allow the analog amplifiers to settle after the gain adjustment. If the envelope still exceeds the user defined threshold then the controller will again reduce the gain setting by half to one quarter. Otherwise if the threshold is not exceeded the gain is increased to three quarters of the full range. This binary search process is repeated for all six bits of the gain control code. If at any point during this sequence the radio controller FSM indicates that it has found the start of a packet the gain adjustment process is stopped and the current value is held.

The 6-bit output code of the AGC is used to control the gain of three amplifier blocks in the analog domain. The frontend TIA has 20 gain settings while the first and second filter amplifiers have 13 settings each. A mapping takes place to convert the 6-bit AGC code to the appropriate signals to control the gain of the three amplifiers. The best choice for where in the signal path to begin reducing the receiver gain is dependent on the current reception scenario. Linearity and noise performance are fundamentally at odds with one another when determining where to begin making gain reductions. The earliest amplifiers in the signal path set the noise performance limit and their gain should be maximized to make the noise contributions of subsequent stages insignificant. The limitation on linearity however is the later stages in the signal path which face the largest amplitude signals in the receiver due to the large amount of gain preceding them. By beginning the gain reduction with the last amplifier stages the noise performance of the receiver remains at its best and it is still able to receive small RF signals. This however does not help with linearity in the presence of large input signals (which must be present if the gain needs to be turned down). If the desired input signal is large and thus well above thermal noise limits then beginning gain reduction near the antenna and proceeding toward the ADC makes sense. Since there is a large amount of SNR available, the noise performance of the receiver is not critical in this scenario and it is more beneficial to maximize linearity. If the desired signal is of only moderate power however then turning down the frontend gain may drop it below the minimum detectable level. Since the low power nature of the SCM receiver results in limited linearity to begin with, the adjustment order of starting with the frontend and moving toward the ADC was chosen to improve linearity.

The AGC has two separate gain control modes that determine which of the analog amplifiers are adjusted. In the TIA-only mode the filter amplifiers are held constant at maximum gain and only the frontend TIA is controlled. The TIA-only mode provides about 20 dB of gain control and allows for a short wait time to enable fast settling. The full-range mode covers the entire gain range of the receiver. In either case the user defined threshold setting should remain above the envelope of the thermal noise at the filter output to avoid false triggers. The gain controller also provides the ability to add a gain offset to the I and Q signal paths to correct for amplitude imbalances. The correction value and which channel to apply it to are set via memory mapped registers from software. There is also the ability to bypass the AGC and directly set the I and Q gain values from software. The operation of the automatic gain controller inherently provides a value that can be used as a Received Signal Strength Indicator (RSSI). The microprocessor can read the final gain value that the controller has settled to via a memory mapped register and then use that value as an indicator as to how large the input RF signal was.

4.9 Link Quality Indicator

A metric for tracking the historic performance of a link between two nodes is a very useful piece of information for deciding how to arrange links in a wireless mesh network. This allows the network to prefer paths that are more likely to result in successful packet transfer and thus require fewer re-transmissions. While RSSI provides an estimate of the strength of the incoming RF signal it is not necessarily a good indicator of the likelihood of successful packet transfer between two nodes. One drawback of RSSI is the large amount of variation that will occur due to multipath. Another example of the drawbacks of solely using RSSI as a link quality index is when large interference signals are present. While the radio may report a large value of RSSI due to the high power of the interferer, the desired signal may be considerably smaller and experiencing a large number of reception errors due to the interference.

In order to give the microprocessor access to additional link quality information an estimate of the chip error rate can be extracted from the correlator. The correlator operates by comparing the incoming chip stream to the 16 possible combinations that are used in the Direct Sequence Spread Spectrum of 802.15.4. The correlator chooses the symbol which is the best match to incoming data and outputs that symbol to be stored in memory. In doing so the correlator has already calculated how many errors there were in comparison to the best matching symbol. By accumulating this error count over some known number of chips a direct estimate of the chip error rate can be obtained. This number provides a much better indication of the failure rate for a given link and is provided to the microprocessor via a memory mapped register which can be read when the packet reception finished interrupt executes.

4.10 Zero Crossing Counter Demod

An alternative demodulator is also implemented on the Single Chip Mote which attempts to discriminate between high and low FSK tones by measuring the amount of time between subsequent zero crossings in the IF waveform. This simplifies the design requirements for the analog portion of the receiver as only a single bit comparator can be used for this purpose instead of requiring a complete ADC. Image rejection is also discarded in this mode which removes the requirement to have both quadrature receiver paths in operation. The demodulator operates using a counter to determine how many of the comparator clock ticks occur between zero crossings and a user defined threshold dictates whether the resulting count value constitutes a digital one or zero. An example of the transient operation of the demodulator is shown in Figure 4.12. A high clock frequency is needed to obtain enough time resolution to make accurate bit decisions. The duty cycle of the IF waveform is also important if both positive and negative zero crossings are used to provide count values as any DC imbalance in the waveform duty cycle will degrade performance by skewing the counter values. This necessitates a comparator offset trim mechanism capable of fine adjustment steps. While such an offset trim mechanism is relatively easy to implement and manually tune, it is more difficult to deal with this issue at the system deployment level where an automated calibration mechanism is required to deal with offsets that vary over time due to environmental factors. The zero crossing counter also tends to be less robust to interference than the matched filter. Interference that is not sufficiently filtered out prior to the comparator will introduce additional zero crossings into the IF waveform and thus more errors.

The zero crossing counter is followed by an oversampling clock and data recovery module. The CDR operates at the same clock rate as the demodulator and uses a counter to attempt to locate the middle of the chip period for sampling. At every edge in the data the CDR

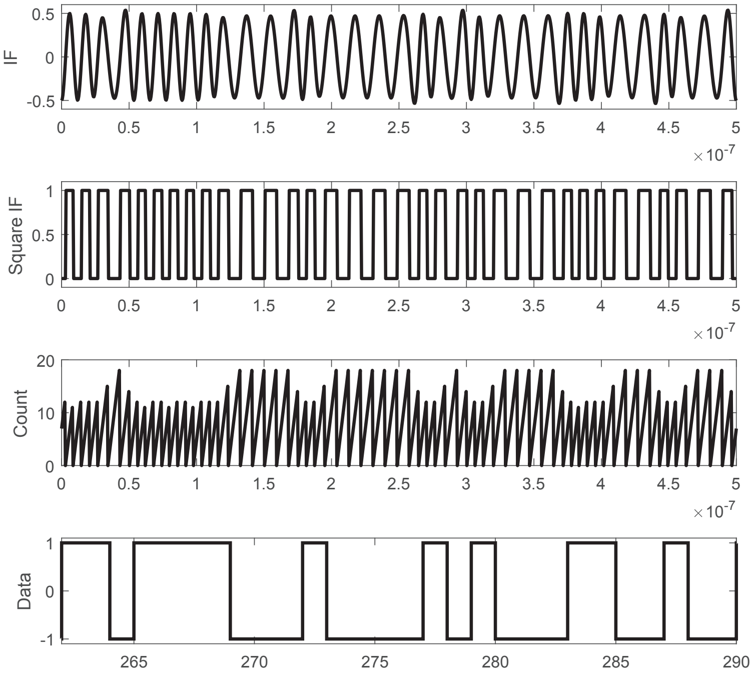


Figure 4.12: Example of the transient behavior of the zero crossing counter demodulator. The analog IF waveform (top) is sampled at a high rate with a comparator to produce a 1-bit digital waveform (2nd). A counter is then used to find the time between zero crossings of the square wave (3rd). A threshold is applied to the counter values to produce a demodulated output data stream (bottom).

counter is reset in order to track frequency error in the data rate. While this allows the CDR to tolerate the frequency errors inherent in a crystal-free receiver it also introduces additional problems. The pulse widths output from the demodulator tend to be of varying width for ones versus zeros due to the necessity of exceeding a set threshold before making a bit decision and toggling the output. This asymmetry can cause locking issues with the CDR as narrow data pulses have edges that are close together and thus appear to be arriving at faster than the data rate. Furthermore this same issue causes the CDR to be prone to chip slips which are catastrophic to packets since all of the de-spreading correlations are off by one.

Overall the zero crossing counter demodulator provides a simpler, lower performance alternative to the main demodulator path. It provides more flexibility than the matched filter path in that it can be used to demodulate FSK signals with many different tone spacings and data rates rather than solely the MSK equivalent of 802.15.4. It has a very ad-hoc nature in that its theoretical performance limits are not well defined and there are many implementation tweaks that can be made to improve its performance. It should also be noted that the zero crossing demod is capable of providing similar information about the average intermediate frequency value since it is continuously measuring this value during its normal course of operation.

4.11 Arbitrary Receive Mode

An arbitrary receive mode was included in order to enable the flexibility to receive more than just standard 802.15.4 formatted packets. This mode can be used for various purposes such as receiving other FSK variants or experimenting with different DSSS code sets for 802.15.4. The output of either demodulator path can be routed to the input of a 32-bit shift register used in this mode. There is no FIFO so the microprocessor is required to retrieve the data every time 32 new bits are available. An interrupt is asserted every 32 bits and the data is latched into a separate register to ease the timing requirements of reading by the microprocessor. In order to aid in synchronizing to the start of packets a 32-bit target start value can be programmed via a memory mapped register along with a Hamming distance threshold. If the Hamming distance between the current contents of the shift register and the target value are less than the programmed threshold then an interrupt is asserted and the shift register index reset such that another interrupt occurs every time 32 new bits are available. While the microprocessor is required to be actively involved during the packet reception, this search value interrupt enables it to sleep for the majority of the time and be alerted via an interrupt when it needs to shuffle the packet into memory. The data rate of the incoming bits is ultimately limited by how fast the microprocessor is able to execute a read of the memory mapped register followed by a write to memory.

4.12 FPGA Verification

The block diagram for the complete matched filter digital baseband is shown in Figure 4.13. The Verilog implementation of this baseband and the zero crossing counter were verified on a FPGA in order to enable the large number of trials needed to generate error rate statistics. Digital test vectors were generated using MATLAB and then transferred to a microcontroller which injected them into the FPGA. Various non-idealities were added to the test vectors in order to assess their impact on the overall system via packet error rate. SNR for this section is defined as the signal to noise ratio measured at the end of the I and Q channels prior to the ADC.

Matched Filter

The packet error rate vs SNR for the matched filter is shown in Figure 4.14 for two levels of phase noise. The lower phase noise level (labeled 1x PN) corresponds to an estimate of the phase noise of a single free running LC tank LO. This curve is applicable when receiving from a crystal-based node where the phase noise of SCM will dominate. The higher phase noise curve considers the SCM to SCM communication case where both the transmitter

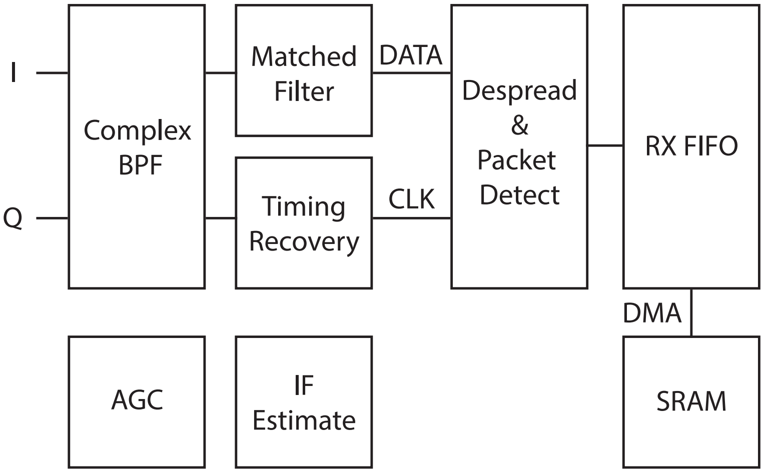


Figure 4.13: Block diagram for the primary matched filter digital baseband.

and receiver have free running LOs whose phase noise will add together. The impact on minimum detectable signal due to the higher level of phase is about 0.5 dB. The resulting minimum point where the higher phase noise crosses the 1% packet error rate threshold is approximately 7.5 dB.

The impact of chipping rate error on the matched filter can be seen in Figure 4.15. The effect of designing the CDR to tolerate the large chip errors possible in a crystal-free system can be seen in the overall 1% span of the tracking range. Another crystal-free concern is inaccuracy in the tuning of the RF LO which results in an offset in the center frequency of the IF waveform. This will cause additional demodulation errors as shown in the matched filter's response to RF channel errors in Figure 4.16 at a fixed SNR of 7 dB. Near the minimum sensitivity level an offset of 100 kHz is enough to triple the packet error rate. The digital baseband's image rejection was also verified as shown in Figure 4.17. The horizontal axis specifies the magnitude of the image signal relative to the desired channel prior to digitization with a 4-bit ADC. All trials occurred at a desired signal SNR of 10 dB. For the 0 dBFS (signal amplitude relative to ADC full scale range) case the image can be in the range of 10 dB larger than the desired channel without experiencing significant increase in packet error rate. For the 6 dBFS case the ADC is being overdriven with a signal amplitude twice the size of its full scale range. This results in saturation which decreases the tolerable image size to about 6 dB larger than the desired channel. The impact of I/Q mismatch was also characterized by adding a 6 dB decrease in the amplitude and a 5° phase error to the Q channel. In both the 0 and 6 dBFS cases the degradation in image tolerance is one dB or less.

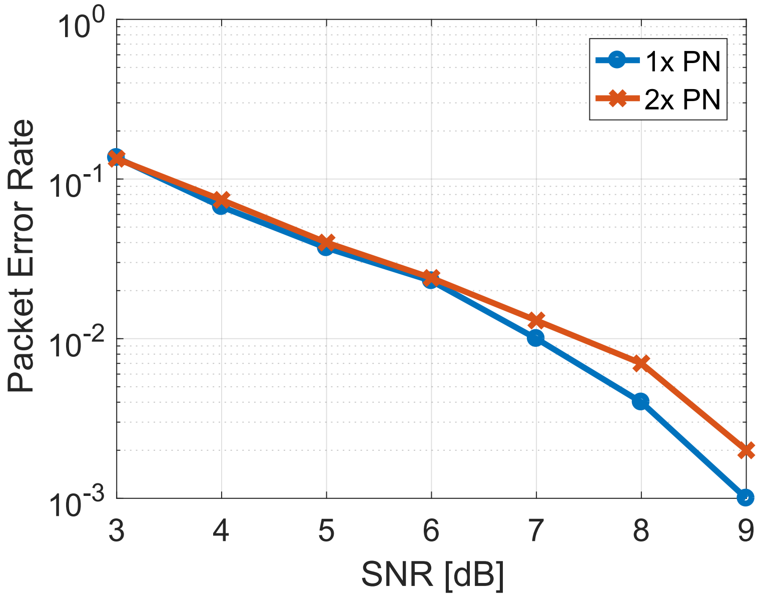


Figure 4.14: Packet error rate vs signal to noise ratio for the matched filter demodulator. SNR is defined as the signal to noise ratio at the output of the I and Q channels prior to the ADC. Two levels of phase noise are shown indicating the SCM-to-crystal-node (1x) and SCM-to-SCM (2x) cases.

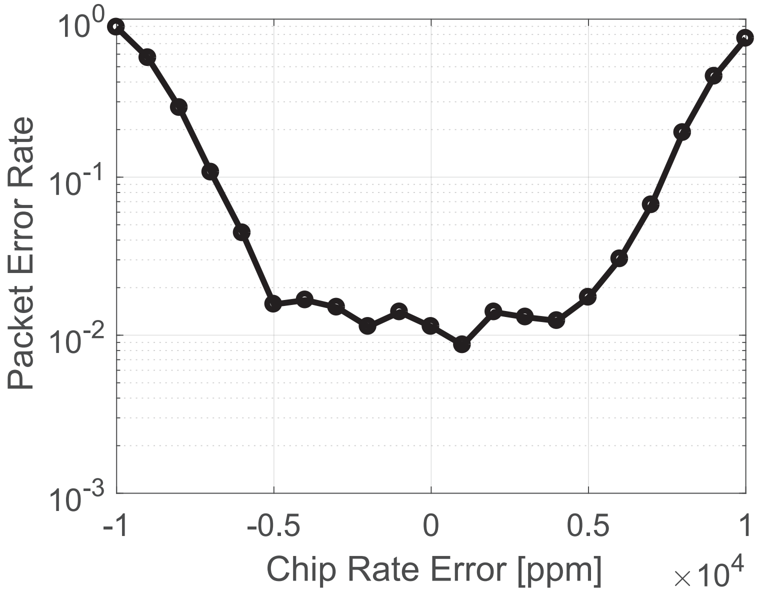


Figure 4.15: Impact of error in chip rate on the matched filter demodulator at a SNR of 7dB. There is minimal degradation in performance seen up to ±5000 ppm error.

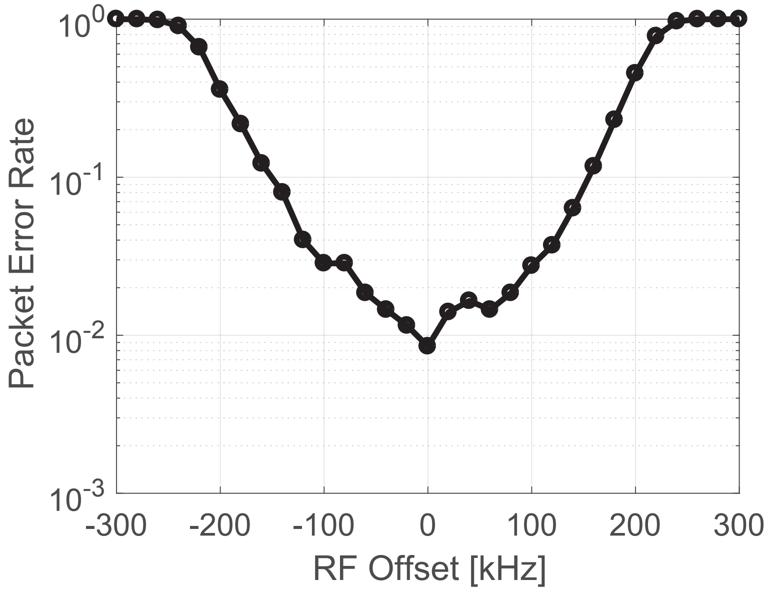


Figure 4.16: Performance of the matched filter demodulator in the presence of an RF channel offset at a SNR of 7 dB.

Zero Crossing Counter

The packet error rate vs SNR for the zero crossing counter (ZCC) demod is shown in Figure 4.18 for the same two levels of phase noise as in the matched filter case. It can be seen that the ZCC requires approximately 10 dB more SNR to achieve the same packet error rate in the single LC phase noise case. When SCM to SCM communication is considered the performance is even worse and approaches a packet error rate floor near 1% regardless of SNR.

The effect of chip rate inaccuracy on the ZCC is shown in Figure 4.19 at a fixed SNR of 18 dB. The CDR does not abruptly fail in the face of rate errors and instead demonstrates degradation in packet error rate performance that is rather gradual. Note that there is some asymmetry present in the oversampling CDR's ability to track rate errors. For a given ppm rate of error it is better for the chip rate to be too fast rather than too slow.

A downside to the ZCC demodulator approach is its sensitivity to duty cycle of the IF signal which can be seen in Figure 4.20 at a fixed SNR of 18 dB. A few percent variation is enough to cause substantial degradation in packet error rate. Care should be taken to trim the comparator offset in order to remove any DC bias in the IF waveform and monitor for shifts due to changes in environmental conditions.

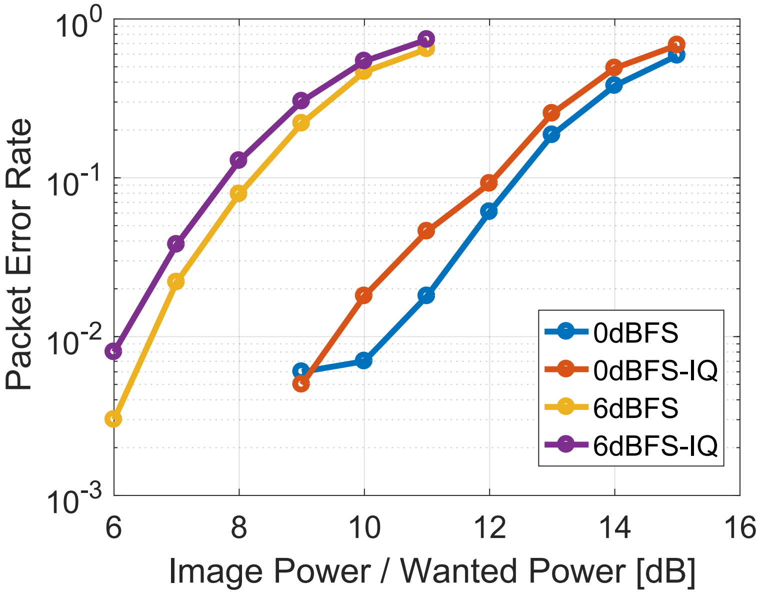


Figure 4.17: Verification of the digital baseband's image rejection comparing the tolerable image power to the desired signal power. Two conditions for total signal amplitude were considered: one where the combined signals were scaled to match the ADC full scale range, and one where the combined signals were twice the full scale range. I/Q mismatch was also considered by adding a 6 dB amplitude and 5° phase error to the Q channel. The desired signal SNR was 10 dB for all trials.

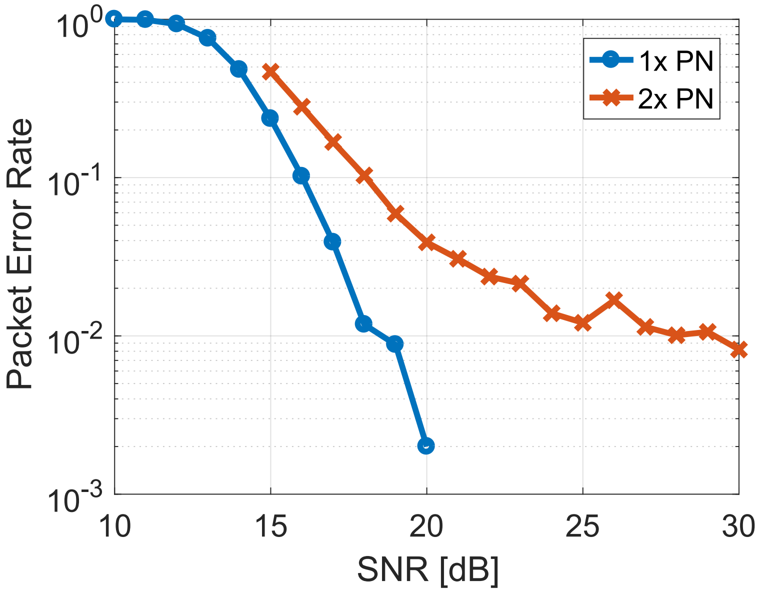


Figure 4.18: Packet error rate vs signal to noise ratio for the zero crossing counter demodulator. SNR is defined as the signal to noise ratio at the output of the I and Q channels prior to the ADC. Two levels of phase noise are shown indicating the SCM-to-crystal-node (1x) and SCM-to-SCM (2x) cases.

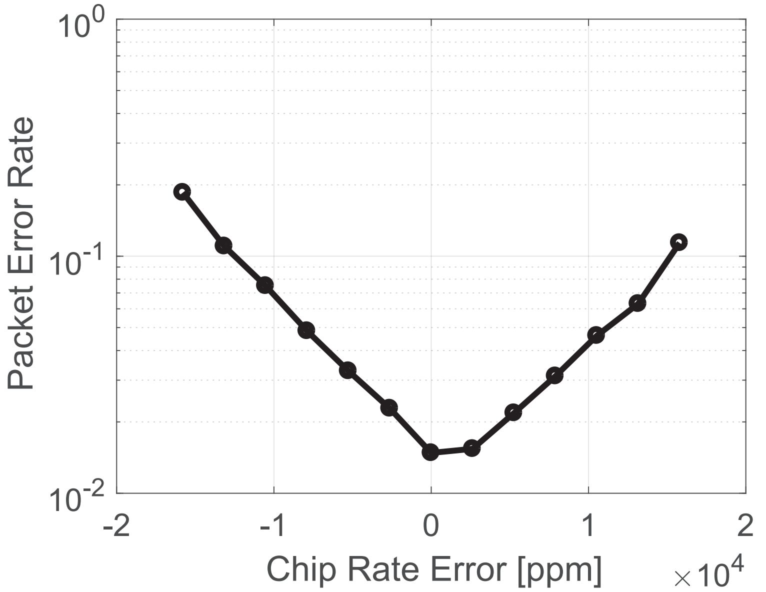


Figure 4.19: Packet error rate vs chip rate inaccuracy for the zero crossing counter demodulator at a fixed SNR of 18 dB.

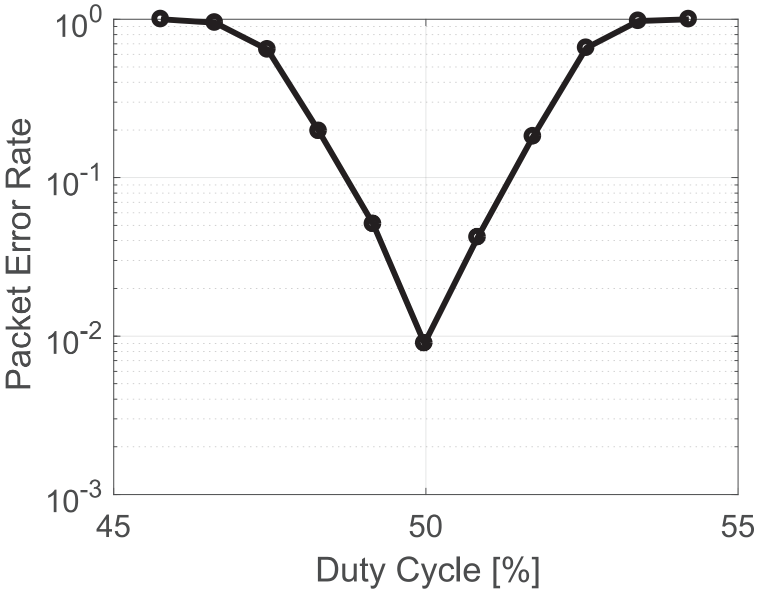


Figure 4.20: Packet error rate vs IF waveform duty cycle variation for the zero crossing counter demodulator at a fixed SNR of 18 dB.

Chapter 5

SCM Measurement Results

5.1 Overview

The pursuit of a functional Single Chip Mote (SCM) prototype spanned several generations of IC tapeouts of increasing complexity. This chapter focuses on the latest generation which was built upon the results of the hardware generations that came before it. The results reported here are taken from the hardware generation referred to as SCM3B and are also applicable to the subsequent SCM3C IC. The primary goal and focus of this chapter was implementation of a crystal-free IEEE 802.15.4 transceiver. The IC was designed to be a complete system such that after attaching an antenna and battery, the mote could be loaded with OpenWSN software and join an 802.15.4 network.

The SCM3B chip was taped out in TSMC 65LP and measures 3 mm * 2 mm as seen in the die photo in Fig 5.1. A high-level block diagram of the system is shown in Figure 5.2. The chip is built around a digital core which contains an ARM Cortex-M0 with 128 kB of SRAM and dedicated hardware for handling 802.15.4 packets and TSCH (time synchronized channel hopping). See [14] for extensive details on the core digital portion of the mote. All clocks are generated on-chip from free-running CMOS-only oscillators, including the RF LO (local oscillator). For details on the RF LO and transmitter see [17]. The chip also includes an optical receiver which is used for contact-less bootloading and is discussed further in Appendix B.

The following results focus primarily on the receiver which is constructed from the analog and digital components discussed in Chapters 3 and 4. The general design strategy was to select proven block-level circuit architectures from literature with an emphasis on their suitability for use in a crystal-free system. Unfortunately due to timing issues in the digital synthesis flow, the on-chip digital circuits were not functional. The following results therefore combine the on-chip analog portions of the receiver with the digital portion being implemented in a FPGA.

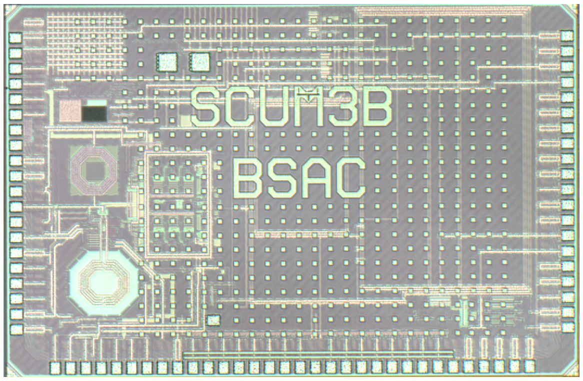


Figure 5.1: Die photo of SCM-3B IC which measures 3 mm by 2 mm and was designed to operate as a complete crystal-free 802.15.4 transceiver.

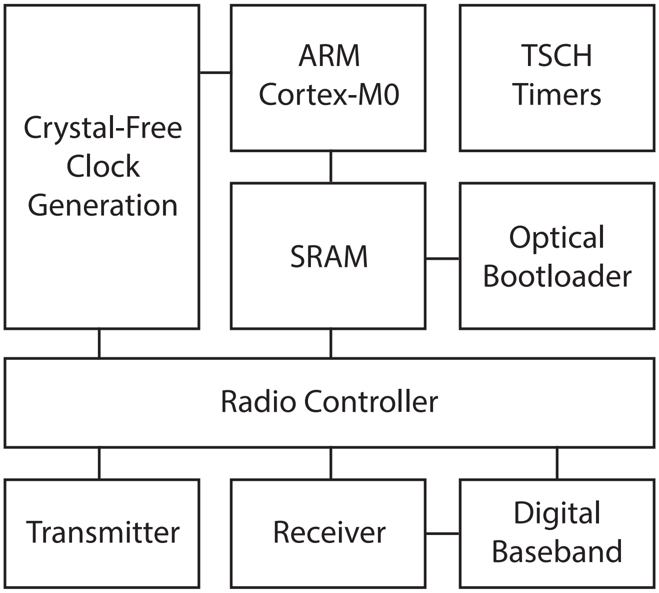


Figure 5.2: Block diagram showing the major components which make up the Single Chip Mote.

5.2 RF Frontend

The frontend is based largely on [37] with its use of passive voltage gain instead of a LNA and passive down-conversion mixers. The mixer was designed by attempting to optimize the trade-off of input impedance reducing the passive gain versus noise contributed by the mixer. The design also had to take into account the loading introduced by attaching the transmitter PA to the same matching network. The RF frontend design is discussed in detail in Section 3.1.

The measured transfer function of the I path TIA is shown on the left in Figure 5.3. The noise figure at the output of the TIA was measured to be 20.6 dB, which includes the 3 dB noise increase from the image channel. This is 4.5 dB worse than expected and is suspected to originate from incorrect simulation of the RF LO to mixer interface. A substantial 6 dB amplitude discrepancy, shown on the right in Figure 5.3, was also measured between the I and Q channel outputs. The source of which is suspected to be related to the increase in system noise figure.

This interface between the LO and mixer is the boundary between the blocks of two different designers and thus is inherently prone to issues going undiscovered. Attempts were made by both designers to perform extracted simulations of the combined LO and mixer prior to tape-out. However the results of these simulations differed, thus indicating that either one or both were incorrect. The more pessimistic result indicated that the LO swing being delivered across the polyphase network to the mixer was below the design target of 200 mV. This could be the case if the mixer loading were substantially different than previously assumed. As is typical this combined verification simulation was not completed until late in the tape-out and the choices were either to risk rushed changes at that stage or accept the possibility of reduced performance. Avoiding the risk of introducing catastrophic changes at the last minute took precedence. The lesson learned was to give more consideration to how to verify such cross-designer interfaces and to conduct the verification much earlier in the tape-out cycle. Seemingly minor choices like how the layout hierarchy is implemented can have profound consequences for the ease with which parasitic extraction and verification can be completed.

Gain control is implemented with a resistor DAC placed between the differential outputs of the TIA. This approach was chosen instead of varying the feedback resistors to limit the change in load impedance at the mixer output. Since the passive mixer is bidirectional, any change in its load impedance will be up-converted to RF and ultimately affect the input matching of the antenna. The gain control steps are shown in Figure 5.4. The relatively low output impedance of the TIA allows the amplifier to respond quickly to changes in gain setting. The linearity of these gain steps is not especially critical as long as it does not interfere with the operation of the automatic gain controller. To verify this is the case it is recommended to perform mixed signal co-simulations between the digital gain controller and the analog blocks which it controls. This can be accomplished with either a full system simulation or by emulating the rest of the signal path and ADC with ideal circuit component models to isolate the impact of the TIA.

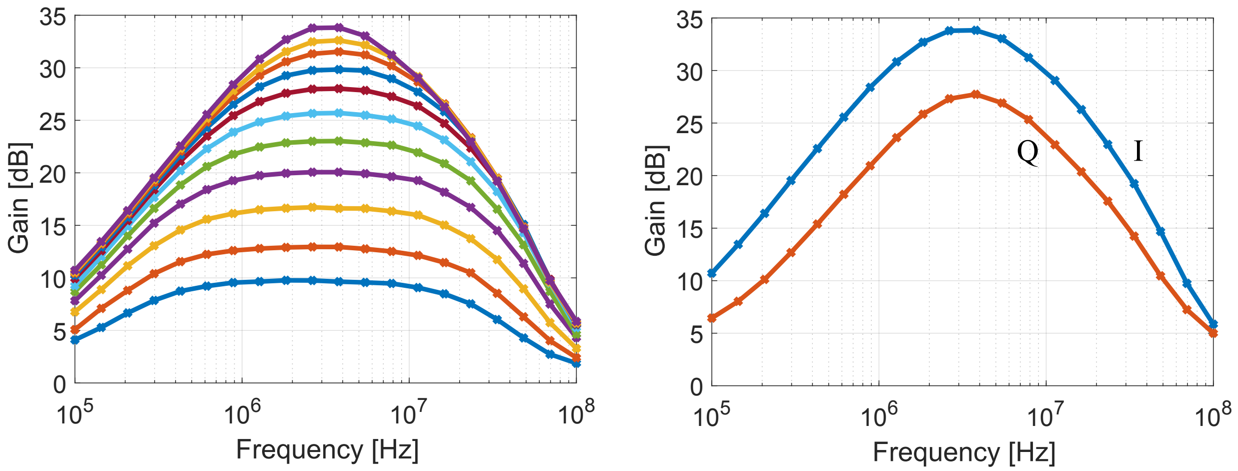


Figure 5.3: Left: Transfer function from RF input to output of I channel TIA for various gain control settings. Right: Amplitude discrepancy between I and Q channels which is suspected to stem from LO to mixer interfacing issues.

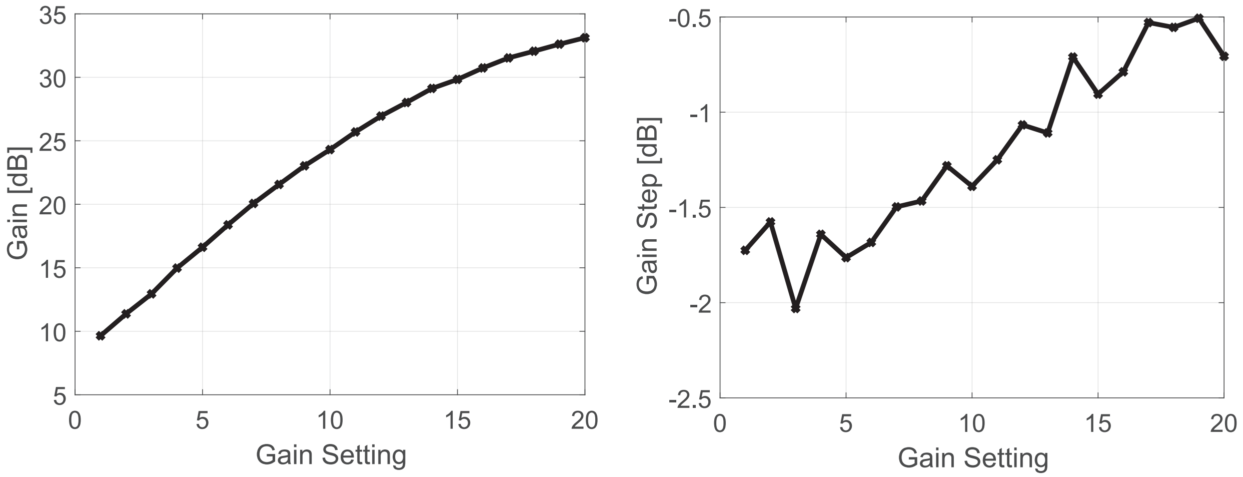


Figure 5.4: Left: TIA gain vs input code. Right: Step size for each change in gain code. It is recommended to co-simulate the final amplifier circuit with the digital automatic gain controller to ensure correct operation.

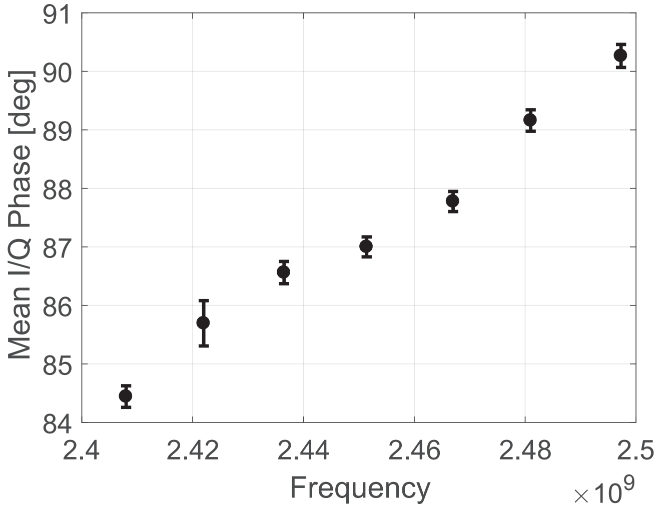


Figure 5.5: Quadrature phase error across the 2.4 GHz band, measured at the mixer output.

For image reject in a low-IF receiver the matching of the in-phase and quadrature LO components limits the achievable performance. The I/Q phase matching across the 2.4 GHz band is shown in Figure 5.5. This measurement was taken by applying a high quality RF tone from a signal generator, mixing it down to IF, and measuring the phase difference between I and Q captures of the mixer output on an oscilloscope. Earlier hardware generations had included a digital I/Q mismatch correction block as discussed in Section 4.4. Measurements from those earlier chips however indicated that the I/Q phase error was less than 3° and thus the compensation block was omitted on this generation of hardware. Measurements indicate that the phase error is now as much as double that original measurement. It is possible that the increase in phase error stems from the same underlying LO to mixer interface problem as the noise figure issues discussed above.

The phase noise of the free running LO will degrade the receiver's ability to decipher between the high and low MSK modulation tones. The LO phase noise, measured by down-converting to IF with a much better reference LO, is shown on the left in Figure 5.6. This phase noise measurement was taken by down-converting to IF, capturing the waveform with an oscilloscope, then digitally mixing down to baseband and taking the FFT. The quantization noise of the oscilloscope in this approach results in the floor that begins to appear around the 500 kHz frequency offset. Simulation data obtained from [17] is shown for comparison (solid line). Similar to earlier hardware generations [27] the simulation appears to underestimate the amount of flicker noise which causes the discrepancy at lower offset frequencies while the white noise appears to match well. While this flicker phase noise is higher than expected it still appears to contribute little to degradation in overall performance as originally discussed in Section 2.3. This same information can also be viewed as a histogram of variation in IF frequency as shown on the right in Figure 5.6 which provides a more in-

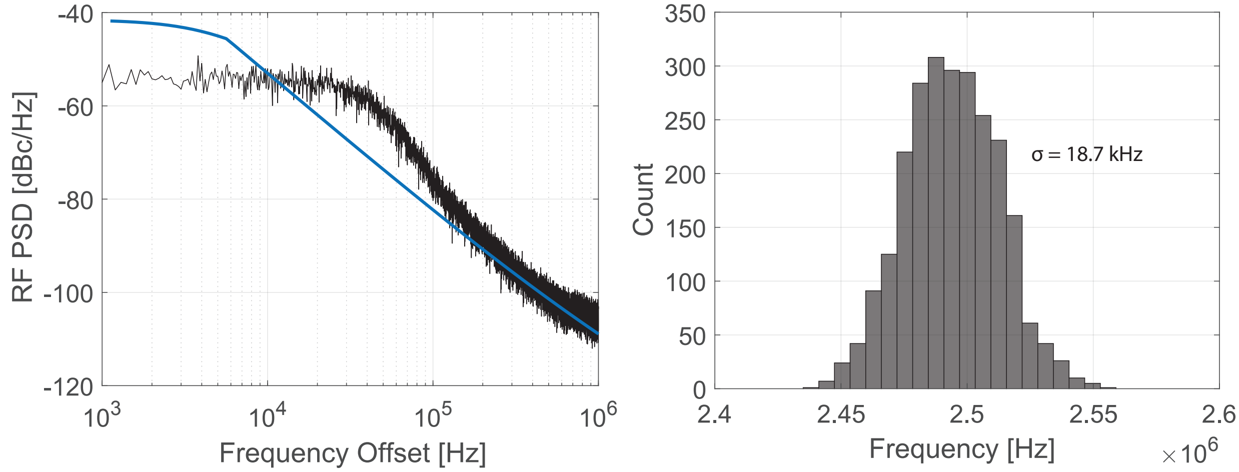


Figure 5.6: Left: Measured vs simulated LO phase noise. Right: Histogram of cycle-to-cycle frequency during 1 ms after down-conversion to 2.5 MHz intermediate frequency.

tuitive way to assess the impact of phase noise on the MSK demodulator. The IF has a one sigma variation in period of 18.7 kHz (over a measurement interval of 1 ms) which is small compared to the 1 MHz MSK tone spacing of the 802.15.4 modulation. Note that this measurement also captures deviation in the IF from the noise of the down-conversion process as well as from the sample rate of the oscilloscope used during acquisition so the deviation from phase noise alone will be less.

This passive frontend structure is inherently linear but is limited by the low power dissipation of the TIAs following the mixer. IIP3 and IIP2 were measured at the output of the TIA with the receiver gain set to maximum and are shown in Figure 5.7. A Rohde & Schwarz FSP spectrum analyzer was used for this measurement with a instrumentation amplifier interfacing the on-chip differential signals to the 50Ω input impedance of the analyzer with near-unity gain. The IIP3 measurement was conducted with a LO frequency of 2405 MHz and RF input frequencies of 2407.4 MHz and 2407.6 MHz. For the IIP2 measurement the RF input frequencies were 2413.5 MHz and 2416 MHz. The resulting IIP3 is -12.9 dBm and IIP2 is -3.34 dBm. The relatively low value of IIP2 is likely due to imbalance in the mixer device thresholds as the same bias settings were used for all four mixer switches in these tests. Reducing the gain setting of the TIA will improve its linearity as well as reduce the signal amplitude at the inputs of the following filter stages. A separate linearity test showed that the 1 dB gain compression point of the TIA increased by 4 dB when the gain was reduced to its minimum setting. While these linearity numbers are unimpressive compared to those achieved by purely passive frontends that burn much more power, they are more than sufficient for the range of performance targeted by SCM. The downstream filters will be much more of a limiting factor in overall linearity due to the large amount of gain preceding them.

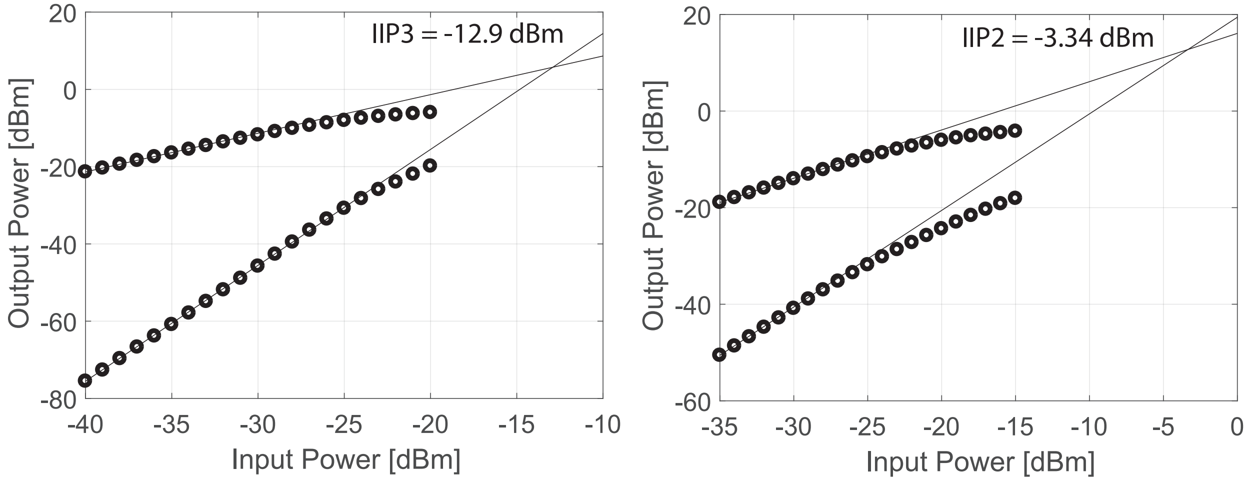


Figure 5.7: IIP3 (left) and IIP2 (right) as measured at the output of the TIA.

5.3 Filters

The primary filtering in the receiver is implemented using a cascade of two identical copies of the discrete time analog filter blocks discussed in Section 3.2. Each filter implements a pair of complex conjugate poles and were together designed to generate a bandpass filter around the IF of 2.5 MHz. The high pass portion of the bandpass is provided by inter-stage AC coupling which also removes DC offsets and flicker noise. The nominal capacitor ratios for the filter were determined using a MATLAB model of the receiver. Since the capacitor ratios must be integers, the design space is small enough to evaluate the majority of the possible options and implement the DACs to cover PVT variation around the nominal value which results in the best error performance. A third copy of the filter serves as an ADC driver to isolate the SAR DAC capacitance from loading the filter. The filters are clocked by a free running RC oscillator of the type discussed in Section 3.4 running at 64 MHz. The frequency inaccuracy of this free running oscillator directly translates to an error in the received chip rate. The receiver implements a tracking mechanism in the CDR to correct for this rate error which was discussed in Section 4.2. The measured tolerance to errors in this clock rate is shown in Figure 5.24 and shows that this clock only needs to be accurate to within about ±5000 ppm.

The discrete time analog nature of the filter outputs requires additional consideration during measurement. A high bandwidth observation path was constructed using on-chip source followers which output a differential signal to a high speed off-the-shelf instrumentation amplifier (INA). The INA converted the signal to single ended and was able to drive the 50Ω impedance of test equipment. The bandwidth of this signal chain was sufficiently high to allow the sampled and held nature of the filter output to be captured by an oscilloscope. The filter clock was simultaneously captured to be used for sampling at the correct time

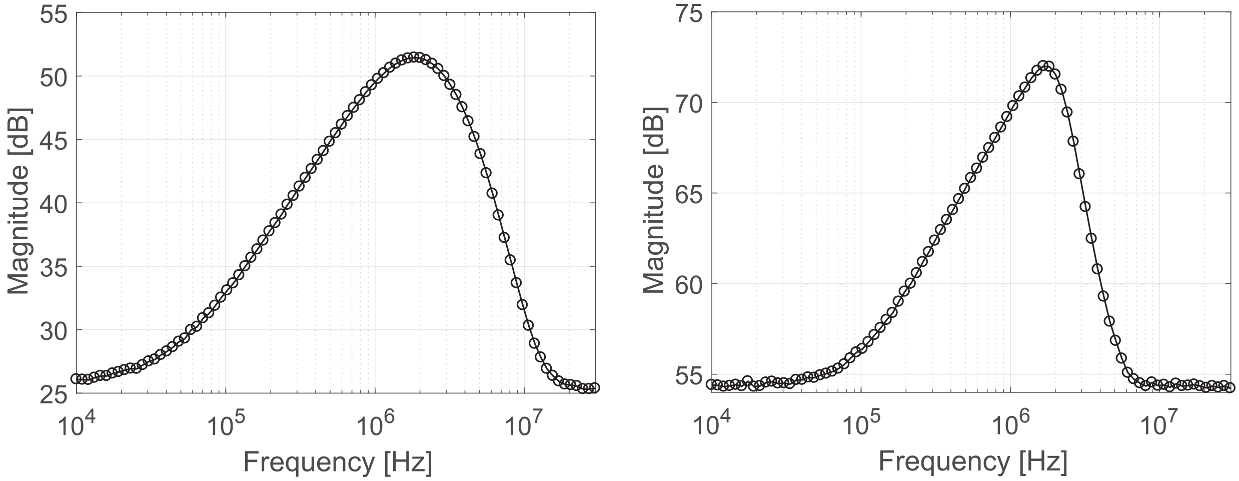


Figure 5.8: Transfer function from RF to the output of first filter stage (left) and second filter stage (right).

instants. This step was necessary as the free running RC oscillator serving as the filter clock has jitter and frequency drift which prevent a simplistic periodic sampling of the output. MATLAB was then used to sample the continuous time waveform at the rising edges of the filter clock thus producing a stream of samples corresponding to the discrete time analog output of the filter. Corrections were applied for cable loss and gain of the debug observation path as were appropriate. All inputs were applied at RF and down-converted by the frontend prior to reaching the filter inputs. For all filter measurements the LO was injection locked with an external stimulus to reduce the effect of phase noise on these measurements and isolate the filter performance.

Transfer Functions

The transfer functions from a RF input to the output of each filter stage is shown in Figure 5.8. Limitations of the discrete time analog test setup limit the dynamic range of the measurement and are the cause of the apparent floor in the transfer functions. The peak in-band gain at the output of the first filter is 51.5 dB and 72 dB at the output of the second stage. The peak of the passband is lower than expected by about 600 kHz, likely resulting from the use of sampling capacitor values that were on the same order as parasitics within the design. The 3 dB bandwidth of the second stage extends from about 900 kHz to 2.5 MHz. The gain control steps for each filter stage are shown in Figure 5.9. The reduced bandwidth of the filters is attributable to the design goal of spending as little power in the filters as possible. Spending slightly more power here to reduce the impact of amplifier non-idealities and layout parasitics on the transfer function would be a good thing to consider for a future iteration.

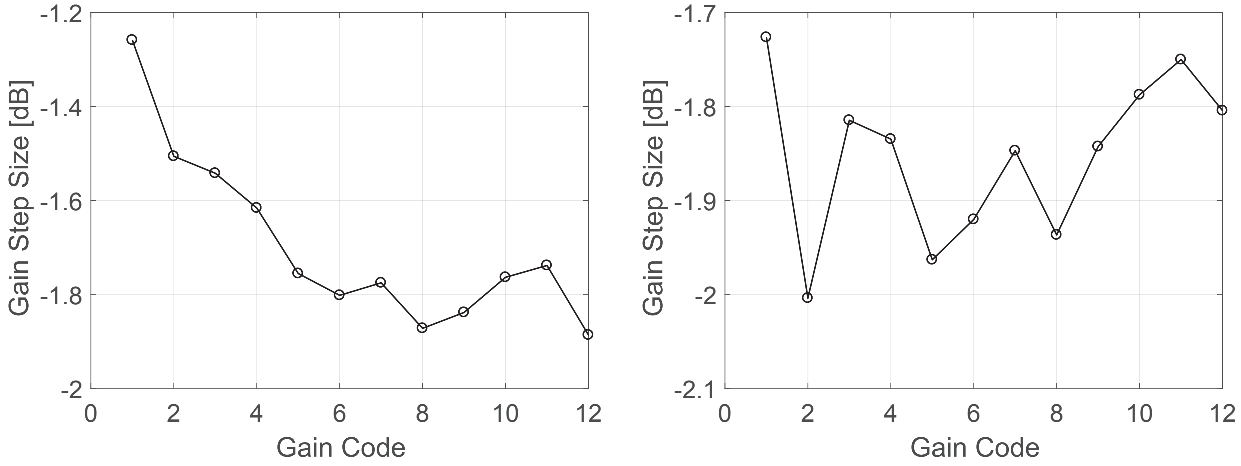


Figure 5.9: Gain control step sizes for first (left) and second (right) stage filters.

Linearity

The low power and open-loop nature of the filter amplifiers substantially limits the linearity of the receiver in comparison to the RF frontend. This type of filter can be exceptionally linear at the cost of power [52], but in this implementation linearity is sacrificed in favor of reducing power consumption. A plot of output amplitude vs input power for stage one is shown in Figure 5.10. A linear fit reveals a 1 dB gain compression point of -53.5 dBm when the receiver gain is set to its maximum value to obtain the best sensitivity. Given the additional gain from stage one prior to stage two, the 1 dB compression point for stage two at the maximum gain setting is further reduced to -75.5 dBm as shown on the left in Figure 5.11. When tolerance of larger input signals is required, decreasing the gain of the RF frontend to its minimum value results in a 1 dB gain compression point for stage two of -48 dBm as shown on the right in Figure 5.11. The use of the automatic gain controller can help somewhat to mitigate these linearity issues by not operating the receiver at its maximum gain setting, although the linearity is still not very good even at reduced gain. If higher linearity is desired, more power must be spent in the filter stages to obtain it.

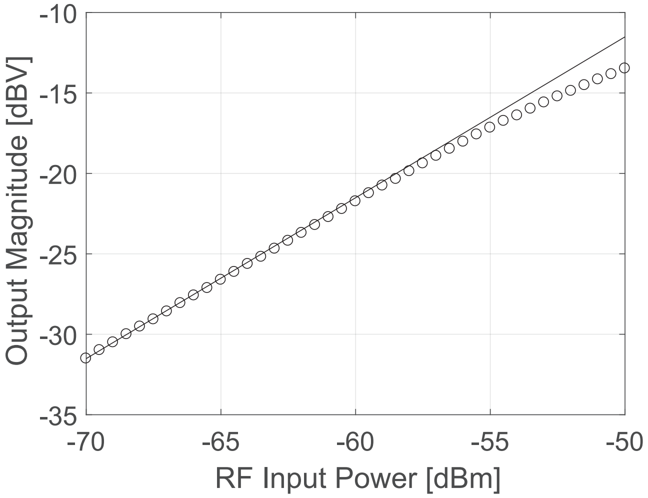


Figure 5.10: The 1 dB gain compression point for the first filter stage at maximum gain is -52.5 dBm.

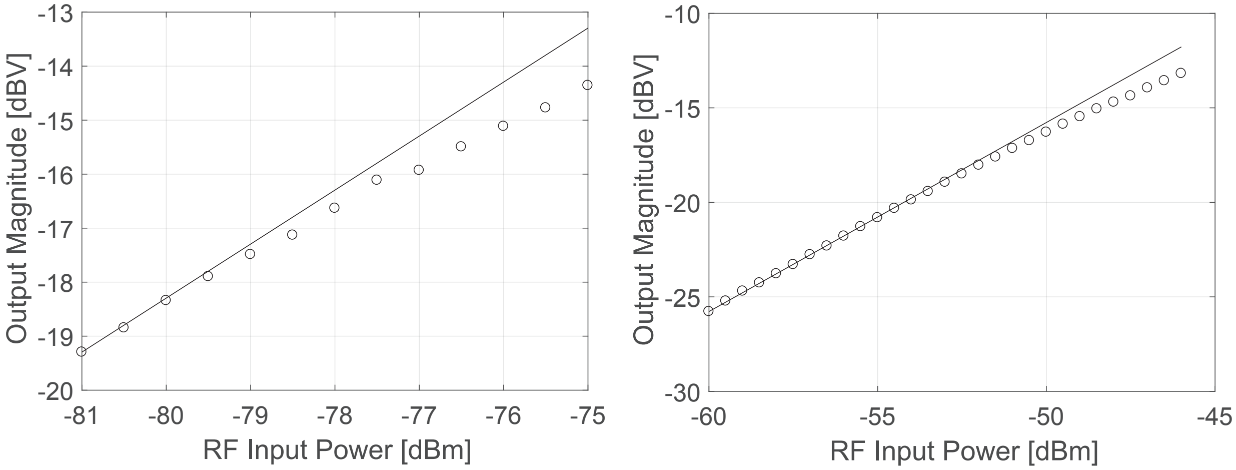


Figure 5.11: The 1 dB gain compression point for the second filter stage at maximum gain is -75.5 dBm (left) and increases to -48 dBm (right) when the frontend gain is reduced to its minimum value.

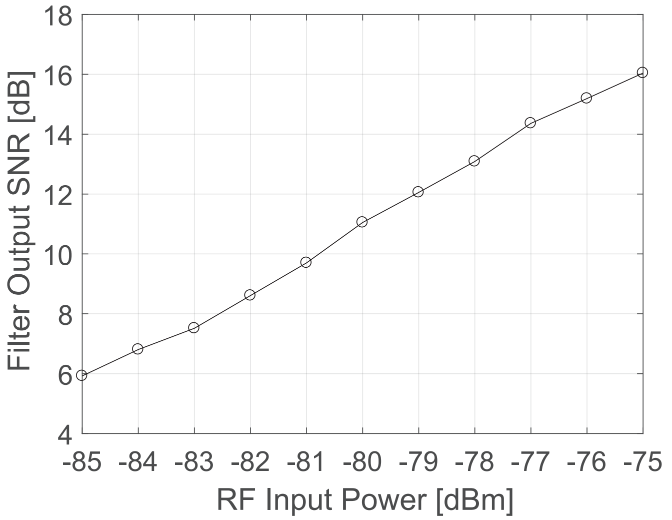


Figure 5.12: Signal to noise ratio measured at the output of the second filter stage vs RF input power indicates that the filter does not degrade the receiver noise figure of 20.6 dB.

Noise

While the filters limit the overall linearity of the receiver, the noise performance is expected to be dominated by the RF frontend. The large amount of gain in front of the filters will reduce their input referred noise to negligible levels even for very low power consumption. The SNR measured at the output of filter stage two vs RF input power is shown in Figure 5.12. From this measurement the noise figure of the receiver up to this point can be calculated to be approximately 20.5 dB. Recall that this is the same value that was measured in Section 5.2 directly at the TIA output. This indicates that the dominant noise contributor is indeed the RF frontend and that the noise contributed by the filters is irrelevant in comparison.

Aliasing

Since the filter is implemented as a discrete time structure, aliasing and frequency folding are fundamentally unavoidable. The receiver's primary protection from aliasing is provided by the one pole roll-off of the TIA in the RF frontend and the sinc filter resulting from the windowed integration operation of the first filter stage as shown in Figure 3.11. The passive match will also provide some level of attenuation at large frequency offsets, but due to its low Q the rejection is limited.

The worst case for aliasing occurs for frequencies that are offset from the LO by the filter operating frequency of 64 MHz. This is the lowest possible frequency for aliasing to occur which means the frontend and sinc attenuation are at their worst. Any frequency injected at this location will appear in-channel after aliasing and potentially corrupt desired signals.

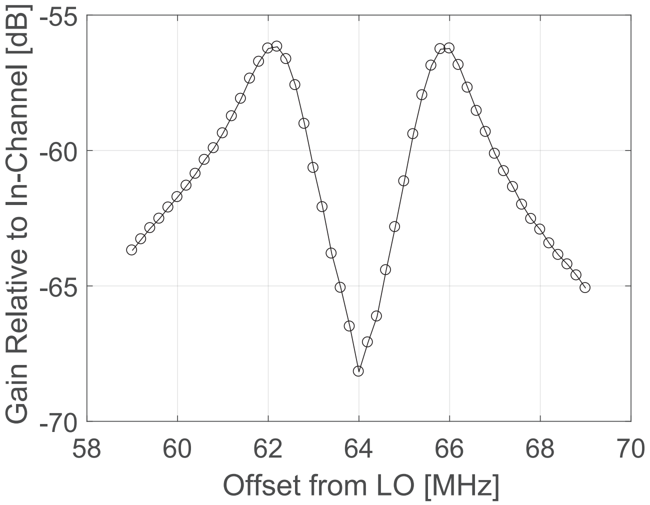


Figure 5.13: The worst case frequency for aliasing is offset from the LO by the filter sampling rate of 64 MHz and is measured to have an attenuation of 56.2 dB relative to signals applied at an IF offset from the LO.

Figure 5.13 shows the magnitude of the transfer function from a RF input with an offset from the LO near the sampling clock to an in-channel output of the second filter stage. The gain is plotted relative to the peak gain experienced by a desired signal injected near the LO frequency and down-converted without aliasing. As expected the result is a symmetric mirror image of the filter transfer function. The worst case attenuation of 56.2 dB occurs at an offset from the sample rate equal to the peak passband frequency. This value exceeds the expected attenuation value of 48 dB calculated by only considering the measured 8 MHz 3 dB corner of the TIA roll-off combined with the attenuation of a sinc function.

To capture the out of band performance of the anti-aliasing a RF tone was swept from 1.6 GHz to just below the 2.4 GHz band and the output amplitude of the second filter stage was plotted. The sweep was broken into two ranges with the first being from 1.6 GHz to 2.25 GHz using a RF input power of -10 dBm. The resulting transfer function is shown on the left in Figure 5.14 and is again plotted relative to the peak gain experienced by a RF input that is an IF offset away from the LO. The second sweep range covered from 2.25 GHz to 2.38 GHz and was conducted using a lower RF input power of -20 dBm to avoid gain compression which becomes more significant as the sweep approaches the LO frequency. The result of the second sweep is shown on the right in Figure 5.14. The LO frequency used during these sweeps was 2.41 GHz and the worst case rejection is observed 64 MHz below the LO frequency.

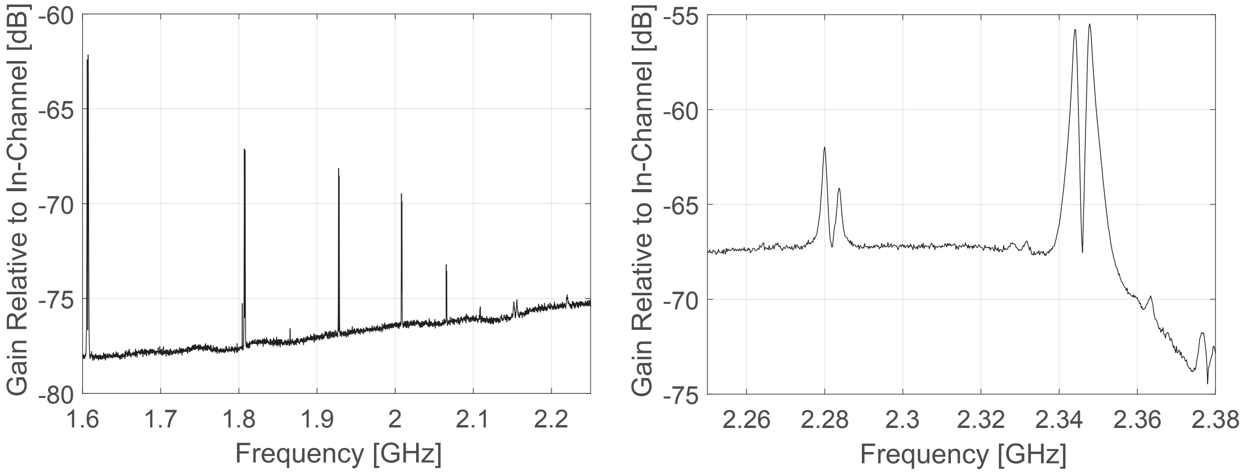


Figure 5.14: Out of band aliasing was evaluated by sweeping the RF input tone and measuring the amplitude of the signal that aliased to within the channel. Gain is plotted relative to that of an un-aliased in-channel signal. Left: A RF input power of -10 dBm was used between 1.6 GHz and 2.25 GHz. Right: The RF input power was reduced to -20 dBm near the LO frequency of 2.41 GHz to avoid gain compression issues.

5.4 ADC

The complex FFT of the 4-bit quadrature ADC outputs for a -83 dBm RF input is shown in Figure 5.15. The impact of I/Q mismatch can be seen by the presence of the signal in both sidebands with a relative difference in amplitude of approximately 5 dB. Figure 5.15 also shows the result of passing the ADC values through the complex bandpass filter. Note that the effect of the LO phase noise can be seen as the tone does not fall in a single FFT bin but rather is smeared across several bins by the phase noise.

To measure the SNR at the output of the bandpass filter vs RF input power, the LO was first injection locked to an external source so that during the subsequent FFT the input signal would fall in one frequency bin. A RF input tone resulting in an IF output of 2.5 MHz was then input to the receiver and the power was swept with the receiver gain set to maximum. The resulting SNR was calculated individually for each ADC and is shown in Figure 5.16. Until the ADC begins to saturate near -70 dBm, the noise figure remains the same as previous measurements at earlier points in the signal path.

Earlier measurements at the filter output indicated that the passband was centered slightly lower than expected. To evaluate the effect of sweeping the intermediate frequency on SNR, a -83 dBm RF signal was applied and its frequency varied to change the frequency of the down-converted product. Figure 5.17 shows the result of calculating the SNR of the bandpass filters outputs at various intermediate frequencies. The result indicates that SNR could be improved at a slightly lower IF value of 2 MHz, which is nearer to the observed peak in the filter passband. While it may appear that this could directly lead to an improvement

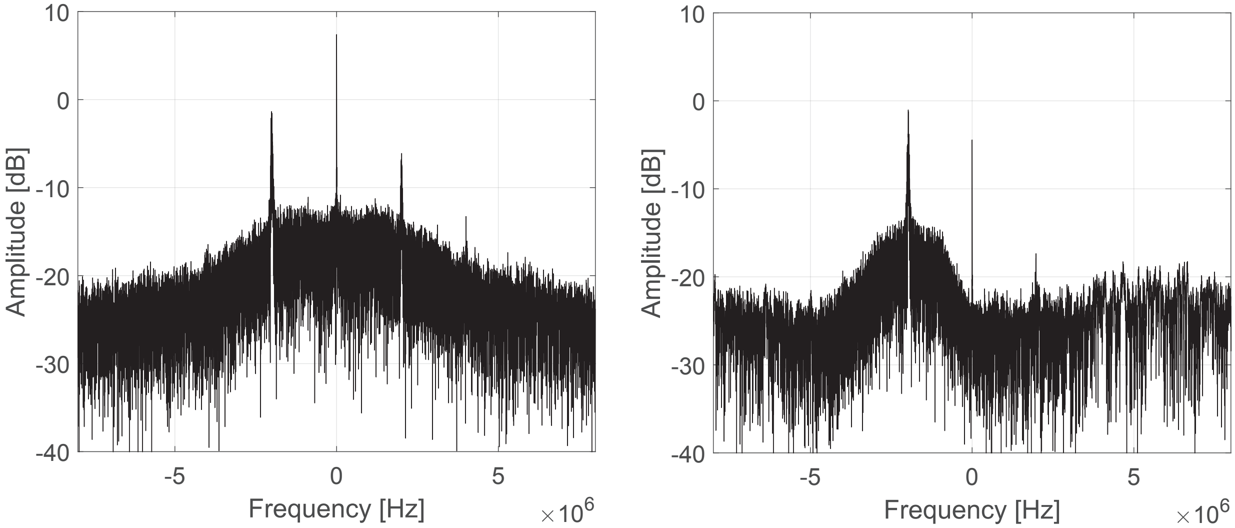


Figure 5.15: Complex FFT showing a 2.5 MHz IF output for raw ADC outputs (left) and after passing through the digital complex bandpass filter (right).

in sensitivity there are other factors to consider at the system level. Changing the IF requires changing both the bandpass filter coefficients and the matched filter templates. Both of these changes affect the performance of the digital baseband and attempts at updating these parameters on a FPGA were unsuccessful in improving sensitivity. If supporting a variable IF is desired it would seem that should be accounted for in the beginning of the design stage to verify the expected performance across the desired IF range. Another consequence of not planning to support a variable IF is that in the synthesized version of the digital baseband the bandpass filter coefficients and the matched filter frequency templates are hard coded. In the future one might consider making these values programmable from either memory mapped registers or scan chain to enable some flexibility.

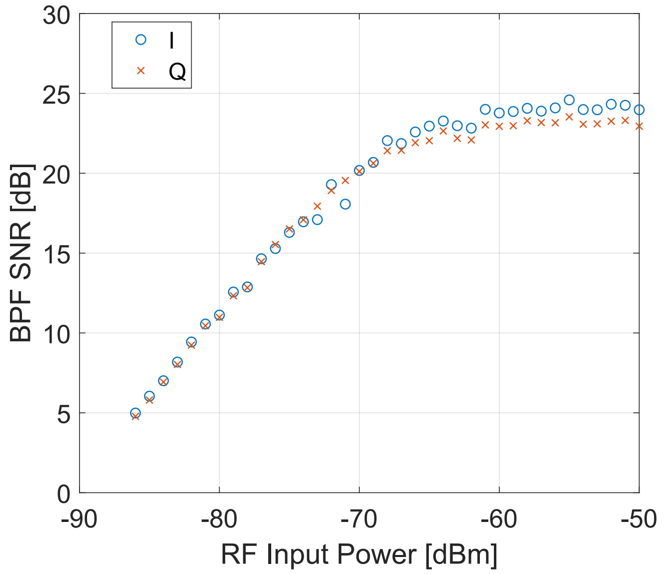


Figure 5.16: Signal to noise ratio at the output of the complex bandpass filter shows that the SNR begins to saturate above -70 dBm if the receiver is left at the maximum gain setting.

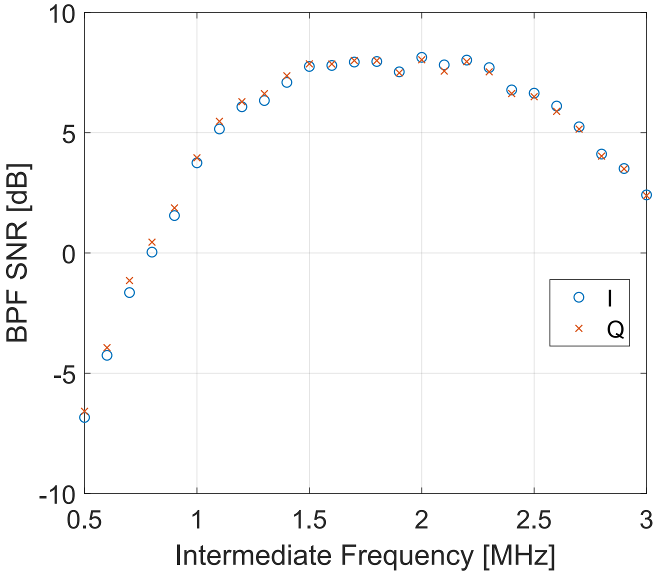


Figure 5.17: Sweeping the intermediate frequency at a constant input power of -83 dBm shows that the peak SNR occurs at a slightly lower IF due to the shift in filter passband discussed earlier.

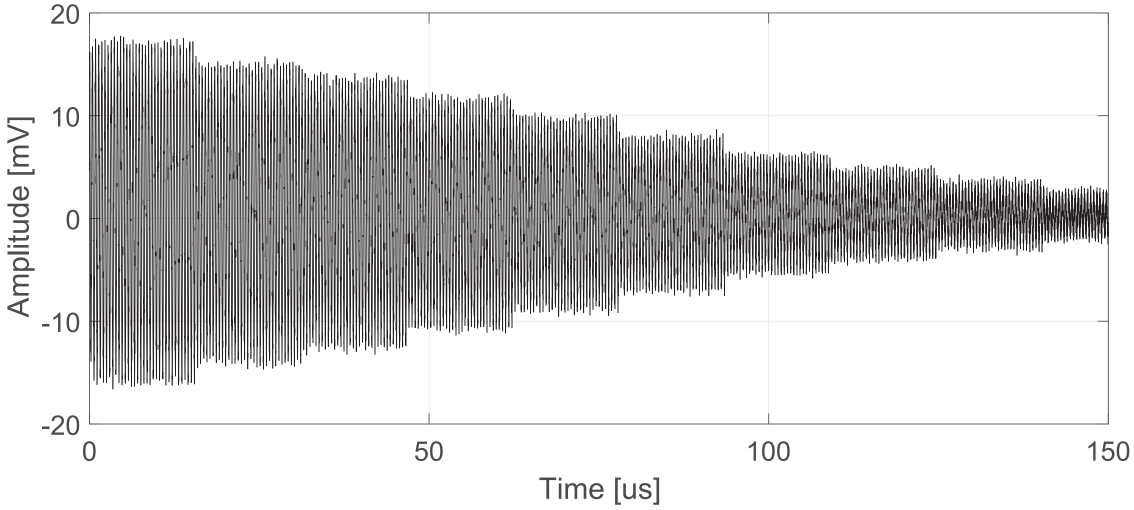


Figure 5.18: Transient output of the TIA showing its response to changes in gain setting across its maximum to minimum range in steps of two codes.

5.5 Automatic Gain Control

The automatic gain controller in the receiver is implemented using the binary search algorithm described in Section 4.8. The controller monitors the output envelope of the bandpass filter and will reduce the gain when the envelope exceeds a software programmable threshold. There are two AGC modes, one which gives the controller the ability to adjust the gain of both the TIA and the filters and a reduced range mode where only the TIA gain is automatically adjusted. The TIA responds very quickly to changes in gain due to the implementation of the resistor DAC across its relatively low impedance output. Figure 5.18 shows the transient response of the TIA as its gain is sweep from maximum to minimum in steps of two codes. The filters are slightly slower to respond due to the change in bias point induced by changing the gm of the amplifier to reduce the gain.

Figure 5.19 shows the transient response of the gain controller in TIA-only mode for a RF input power of -70 dBm. The behavior for the full gain range AGC mode is similar, but requires a longer wait time between adjustments due to the slower response of the filter amplifiers to changes in gain setting. The TIA-only mode settles in less than 40 μs while full gain mode requires nearly 50 μs. To evaluate the performance of the gain controller across a range of input amplitudes the RF input power was swept and the baseband reported RSSI values were recorded for 1024 packets. The RSSI value corresponds to the current gain setting and has a maximum value of 63. Therefore small inputs are expected to have near maximum gain and there should be an approximately linear in dB relationship between the RSSI value and input power. The results of this test for both TIA-only and full range modes are shown in Figure 5.20. For the TIA-only case the gain setting is restricted to a floor at code 43 which the controller has reached by -60 dBm. The larger variation that occurs near

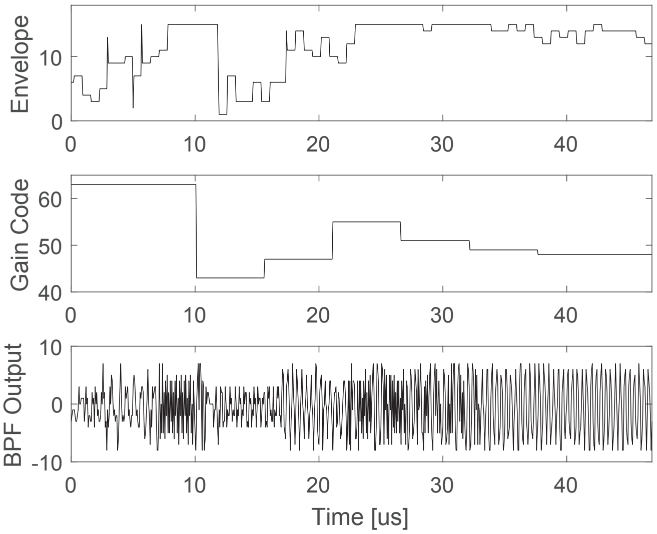


Figure 5.19: Capture of digital baseband waveforms showing the operation of the automatic gain controller. Top: the controller measures the envelope size of the bandpass filter output to make decisions about gain adjustments. Middle: The binary search algorithm settles to a gain setting of 48 for a -70 dBm input. Bottom: The controller attempts to utilize the full scale range at the bandpass filter output.

-50 dBm in the full range case is due to the transition from reducing gain in the first to the second filter amplifier.

In general the receiver can be operated at the maximum gain setting without significant adverse effects. The ADC will compress at larger input powers but will still have well in excess of the SNR requirements for successful demodulation. When interference is larger than the desired signal at the input to the ADCs then the dynamic range of the desired signal is reduced. While the complex bandpass filter can further reduce this interference, the envelope of the desired signal will be reduced and cause issues with the AGC controller. In this scenario the gain controller tends to erroneously respond by maximizing the gain in an attempt to increase the envelope of the signal. However since the ADC's dynamic range is already being used up by other interference the signal envelope cannot be made any larger. When there is a single interference source that is large enough to pass the filters and dominant the ADC dynamic range, there is no substantial benefit to be gained from operating the AGC. However, if there are multiple interference sources that could potentially intermodulate and corrupt the desired channel then AGC is more likely to be beneficial. Another benefit of operating with the gain controller enabled is the RSSI output from the baseband which provides more information about the link. However, like in other receivers, the RSSI value can be misleading due to multipath or interference and the LQI is likely a better choice of link metric. All measurements in this chapter outside this section were taken with the AGC disabled and the gain set to maximum unless otherwise noted.

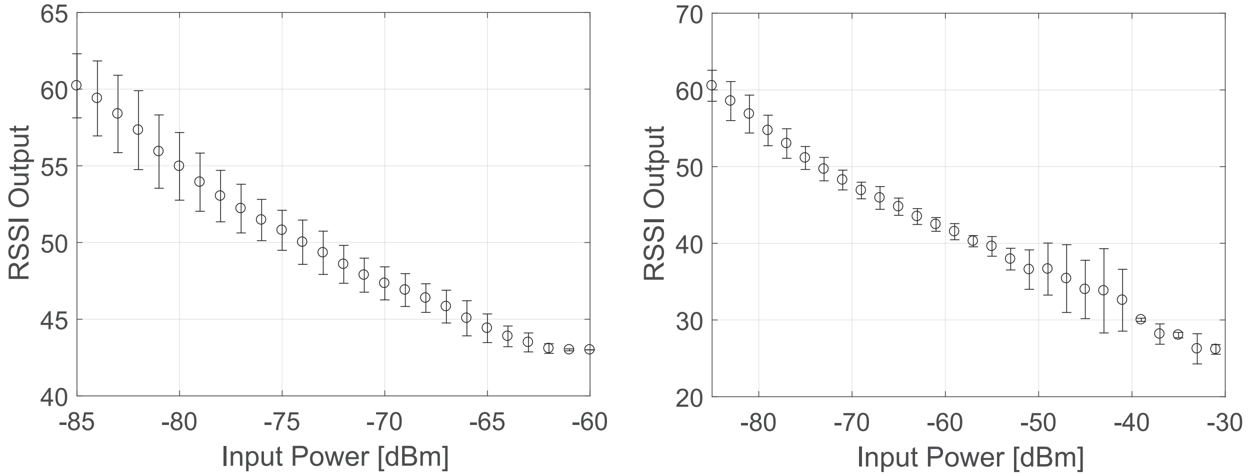


Figure 5.20: The AGC reports its final gain setting to the microprocessor for use as a RSSI measurement. Left: TIA-only mode restricts the controller to only changing the TIA gain which leads to a minimum output code of 43 at high input powers. Right: The full range of the gain controller reports an RSSI that is approximately linear in dB with RF input power across the receiver's usable input range.

5.6 Clocks

Receiver Chip Clock

The discrete time portion of the receiver was designed to use a 64 MHz clock, which due to the crystal-free nature of the chip must be derived from CMOS oscillators. The highest quality clock that could be obtained is from dividing down the RF LO to the appropriate frequency. This approach has the advantages of producing a clock with a low amount of jitter that tracks frequency changes of the RF LO. If a temperature compensation and channel tuning scheme is implemented for the RF LO then any other receiver clock derived from it is also inherently corrected with no further effort. A downside is that the RF LO frequency is required to change for different channels while the receiver clock should remain fixed at 64 MHz. This negates the ability to use a fixed LO divider and instead a variable divider must be implemented at a higher cost in power. Furthermore the use of any divider is going to incur a power penalty while it is being operated. Alternatively a low frequency RC oscillator can be used to generate this clock at significantly reduced power.

The ability to tune this clock is also very important since the receiver chip clock is derived from this source. The frequency step size of the tuning should be smaller than the tolerable chip rate error of the CDR in the digital baseband. The measured tuning curve of the oscillator is shown in Figure 5.21. The mean fine code step size varies depending on the oscillation center frequency, but in the region of interest near 64 MHz the step size is generally 2000 ppm or smaller. The schematic design for this oscillator and its tuning DACs

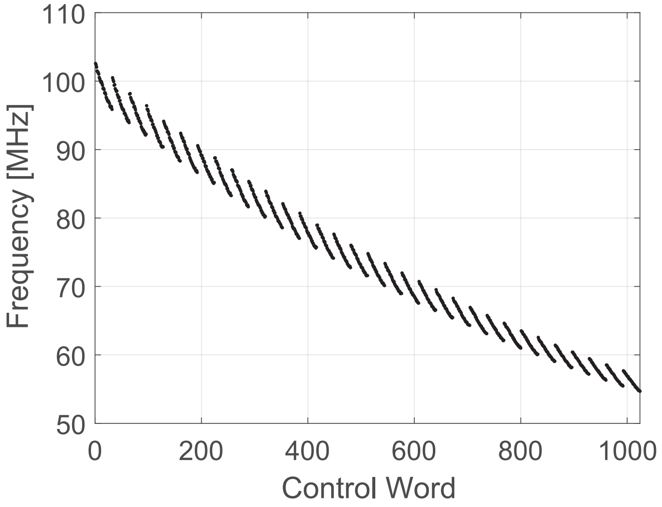


Figure 5.21: Receiver clock tuning curve which has 5-bit coarse and 5-bit fine tuning DACs.

was discussed in Section 3.4.

Transmitter Chip Clock

While this work is concerned only with the receiver portion of the Single Chip Mote, a note on the chip clock rate for the transmitter is warranted. Per the IEEE 802.15.4 standard, the TX chip clock accuracy should be better than ±40 ppm although receivers will typically tolerate a much larger error. For example the TI cc2538 [13] tolerates ±1000 ppm as shown in the measurement in Figure 1.4. Since the tolerable chip rate error of other 802.15.4 receivers is not under our control, this clock must have fine enough frequency steps to meet those tighter specifications. On SCM the transmit chip clock is derived from another copy of the RX oscillator which is scaled down in frequency to 2 MHz. A total of five 5-bit binary weighted DACs were implemented to provide fine tuning steps as well as cover PVT variation. Three coarse DACs cover broad tuning while fine and super-fine DACs have frequency step sizes of approximately 80 ppm and 30 ppm respectively. While these step sizes are more than sufficient to tune within the acceptable limits of other commercial 802.15.4 receivers, it should be noted that managing the tuning of this oscillator with five separate, overlapping DACs from software is rather cumbersome. Giving some thought during the design phase as to how this oscillator will need to be controlled and tuned from software during actual radio operation is worthwhile.

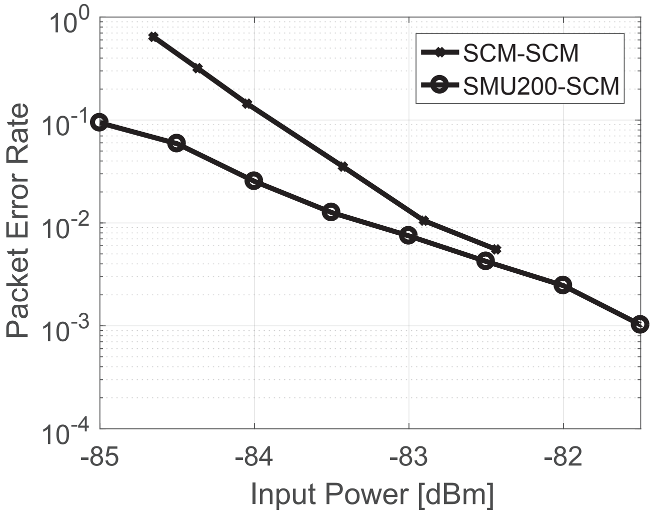


Figure 5.22: IEEE 802.15.4 receiver packet error rate vs input power for input signals generated from RF test equipment and from a SCM transmitter.

5.7 Receiver Performance

The receiver performance was first measured using a Rohde & Schwarz SMU200A to generate 2.4 GHz 802.15.4 packets with a payload length of 20 bytes. The digital MSK chip stream used to modulate the RF source was generated on another FPGA running the digital system. In order to isolate the performance of the receiver from the issues associated with crystal-free frequency tuning, the receiver clocks (including the RF LO) were tuned as close as possible to their ideal values by using counters referenced to a crystal-based source. The TX chipping clock source was provided by an accurate 2 MHz crystal-based source for this initial test. The resulting waterfall curve is shown in Figure 5.22 and indicates a -83 dBm input power is required to achieve a packet error rate of $< 1\%$. Due to the increase in system noise figure previously discussed in Section 5.2 this falls short of the target 802.15.4 sensitivity by 2 dB.

To assess the impact of having the phase noise and channel inaccuracy of a free-running RF LO in both the receiver and transmitter, a test was also conducted where the SMU200A was replaced by another SCM. The transmit chip clock was now sourced from an on-chip 2 MHz RC oscillator. Again all clocks are first tuned by comparison against crystal-clock referenced counters. The resulting waterfall curve can be seen in Figure 5.22 in comparison to the curve where the incoming packets are generated via test equipment. A degradation in minimum detectable signal of about 0.5 dB is observed as well as reduced performance at low input power levels.

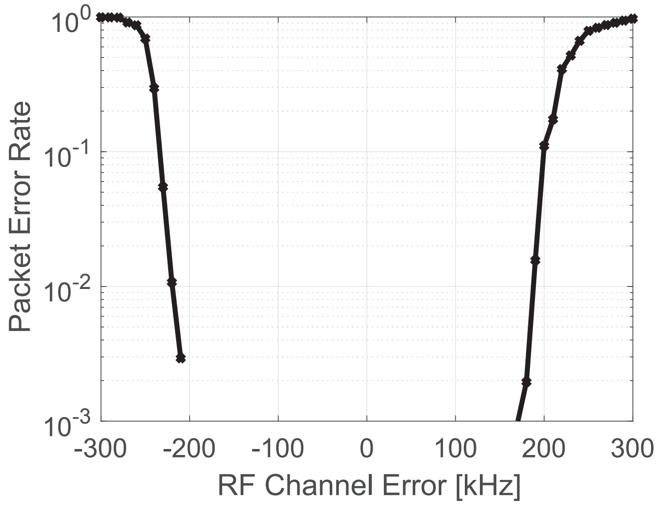


Figure 5.23: Packet error rate versus error in RF channel frequency at an input power of -70 dBm.

RF Channel Accuracy

Since SCM nodes do not have a very accurate sense of frequency it is important to know the limits on how closely they must tune to a channel to send and receive packets. There is a fundamental trade-off between channel accuracy tolerance and noise bandwidth in the receiver. While uncertain-IF architectures are able to find an unknown incoming signal by searching in a wide bandwidth, their sensitivity suffers due to the larger noise input. SCM nodes are designed as a narrow-band architecture but inherently have some channel error tolerance due to the matched filter demodulator. The packet error rate vs RF channel error for SCM is shown in Figure 5.23 at an input power of -70 dBm. It should be noted in the SCM to SCM case that consideration must be given to RF LO frequency relationships between TX/RX due to the coarse quantization of the LO tuning steps. Each tuning step is on the order of 75-100 kHz and if the RX LO is off in one direction while the TX LO is off in the other, the additive result can be a substantial RF channel error outside the tolerable bounds.

Chip Rate Tolerance

It is also useful to know the bounds on tolerable chip error rate in the receiver. To the receiver it is not possible to distinguish between error in the transmit chip clock vs the recovered chip clock. The receiver is only able to sense and correct for a difference between the two rates. To characterize the packet error rate vs difference in chip rate, RF packets were sent from a R&S SMU200A at -70 dBm input power. The digital MSK chip stream was again generated by a FPGA running the digital system and the transmit chipping clock

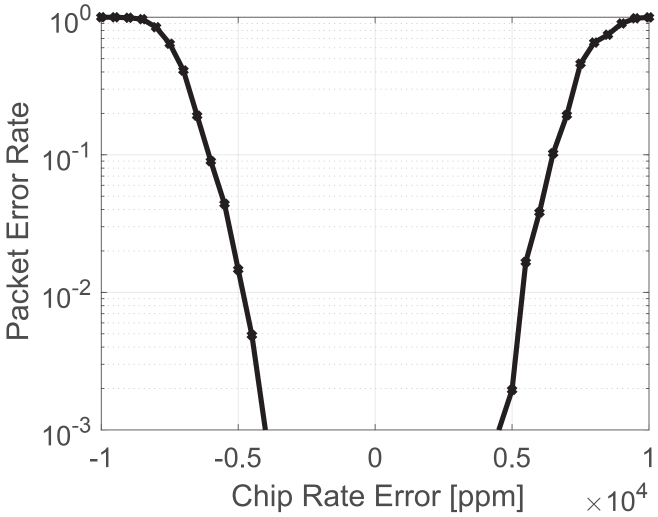


Figure 5.24: Packet error rate vs error in chip clock rate for an input power of -70 dBm. This sets the accuracy to with which the 64 MHz RC receiver clock must be tuned.

was provided by a function generator which was varied around the ideal value of 2 MHz by ± 10,000 ppm. The resulting packet error rate for the worst case condition of maximum length packets is shown in Figure 5.24 and indicates that chip rate discrepancies up to ± 5000 ppm are tolerable.

Maximum Input Power

The packet error rate for large input signals is shown in Figure 5.25 and falls significantly short of the 802.15.4 mandated -20 dBm input power tolerance. The primary failure mechanism is LO pulling due to the input signal which begins occurring around -30 dBm input power. This can be observed on a spectrum analyzer by monitoring the LO leakage while a RF tone is being applied. As the power of the RF tone is increased the characteristic spurs of the injection pulling process will begin to appear [53] followed by eventually collapse of the LO to the same frequency as the external tone. Insufficient consideration was given to the problem of large input signals during the design phase. While there is gain control in the IF amplifiers, there is no way to adjust or reduce the gain of the RF frontend. Thus any large RF signal will be further amplified by the passive matching network and affect the LO frequency through pulling. One possible solution would be to add the ability to shunt the matching network with a low impedance to reduce the RF gain. Another alternative would be to use a LO at twice the carrier frequency followed by a divide by two.

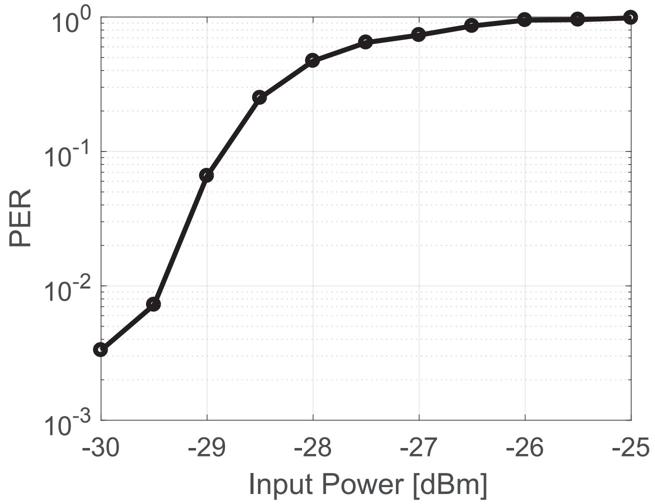


Figure 5.25: Packet error rate for large input power signals. Above -30 dBm pulling begins to occur of the free-running LO followed by injection locking which prevents packet reception.

5.8 Interference Tolerance

Interference testing was first conducted by applying a -82 dBm input power desired signal and injecting an 802.15.4 modulated interferer at various channel offsets. This input power value is 3 dB over the standard specified minimum detectable signal (MDS) of -85 dBm, but is only 1 dB over the receiver's actual MDS. The power of the modulated interferer was swept and the resulting packet error rate rates are shown in Figure 5.26. The same test was also conducted for a larger desired input signal power of -70 dBm and the results are shown side by side for comparison.

The nomenclature used in the 802.15.4 standard defines the 'adjacent' channel to be ±5 MHz away from the desired channel and the 'alternate' to be ±10 MHz away. The standard specifies that the receiver should tolerate an equal power interferer on the adjacent channel. The more stringent 802.15.4 interference specification is the alternate channel rejection which dictates tolerance of an interferer which is 30 dB larger. Due to the low IF architecture the channels at +5 MHz (adjacent) and -10 MHz (alternate) offsets both down-convert to 7.5 MHz and thus have essentially the same rejection. At a -82 dBm input power level 23 dB of rejection is provided for these two cases, which meets the adjacent channel requirement but falls short of the alternate. The +10 MHz alternate channel down-converts to 12.5 MHz and at this higher frequency the receiver is able to provide 37 dB of rejection. In the -70 dBm input power case these two rejection numbers increase to 28 dB and 38 dB respectively.

In both cases the image rejection (the -5 MHz curve) is substantially worse than expected. The original design goal was to tolerate an image signal of equal RF power to the desired channel, but measurements indicate that the desired channel needs to be approximately 10 dB larger than the image to enable reception. The FPGA verification in Section

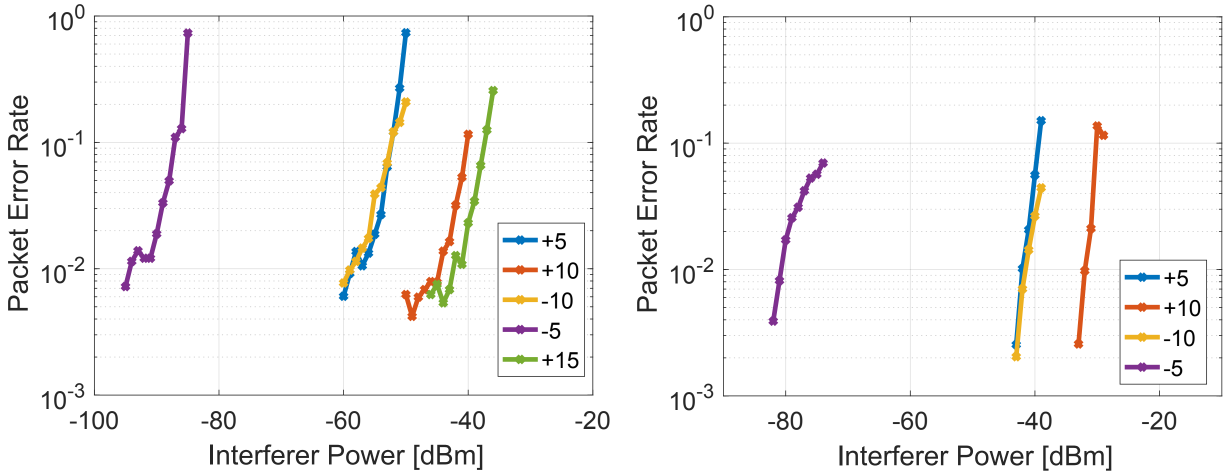


Figure 5.26: Packet error rate in the presence of interference at various 802.15.4 channel offsets. Desired signal power of -82 dBm (left) and -70 dBm (right).

4.12 indicated that the digital baseband was capable of exceeding this equal image power requirement even in the presence of substantial I/Q mismatch. While I/Q mismatch and ADC saturation are likely playing a role in the reduced image tolerance, the primary issue is suspected to be the linearity of the filter stages. When both an image and desired signal are present they both down-convert to within the filter's passband. This situation where two signals fall directly in the passband is the worst case linearity scenario and results in intermodulation between the signals which can prohibit packet reception.

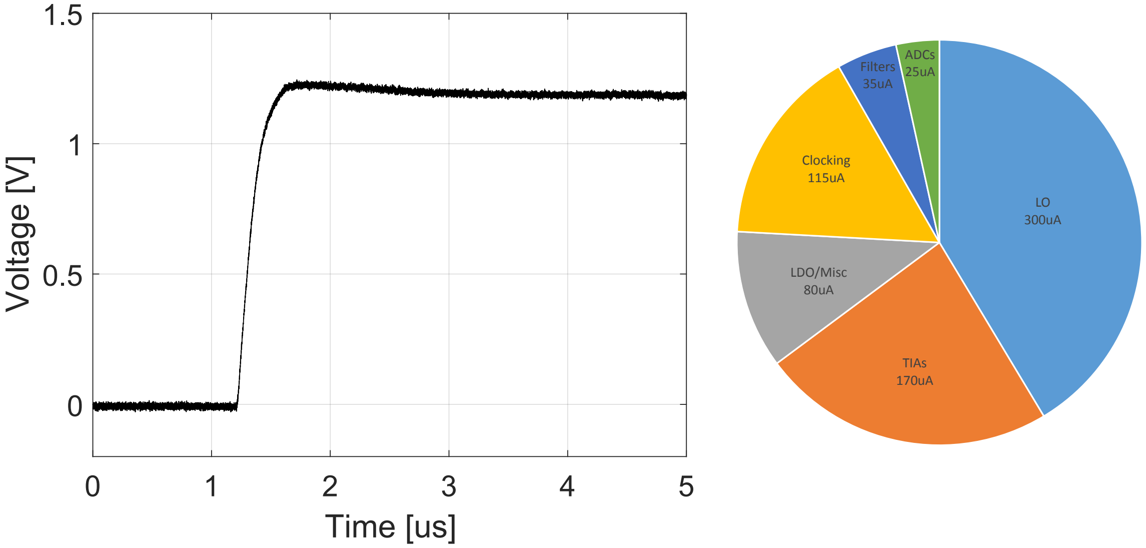


Figure 5.27: Left: Transient startup waveform for the receive chain LDO voltage regulator. Right: Current consumption breakdown of the analog portion of the receiver when operating from a 1.5 V battery.

5.9 Power

The IF chain has its own dedicated LDO for power management which regulates from a 1.5 V battery to 1.2 V. This LDO turns on in about 225 ns which can be seen in Figure 5.27. The RF LO and TX PA are each on their own separate LDO. To enable flexibility for using the system, there are several control methods for enabling the transceiver LDOs. The primary control is via memory mapped register from the microprocessor which allows it to individually turn on the radio LDOs. While this is a very flexible approach it requires the microprocessor to be actively involved. An alternative control method was also included which allows the hardware FSMs that control the radio to directly activate the analog block LDOs. While this method requires the least intervention from the microprocessor it does not provide any flexibility in varying startup times for frequency settling. A third method allows for a single external pin to control the activation of specific LDOs. Due to the failure of the on-chip digital this is the power control method that was utilized in the following sections.

In a time synchronized wireless network the energy dissipation is tightly coupled to the amount of energy required to perform three primary actions:

- Idle listen when a packet reception is expected but never occurs
- Reception followed by transmitting an acknowledgement
- Transmission followed by receiving an acknowledgement

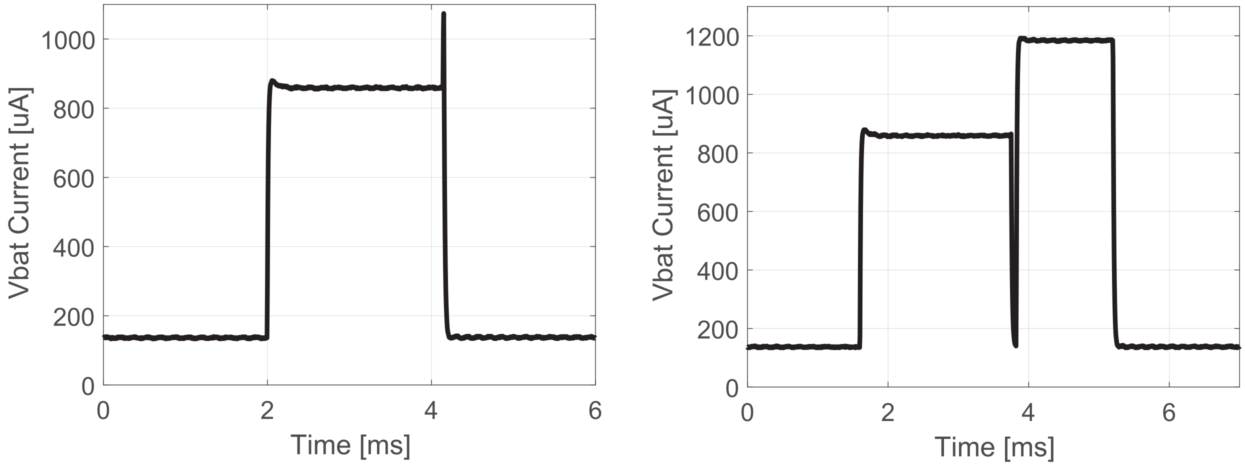


Figure 5.28: Left: Transient current waveform for an idle listen where the radio turns on to receive a packet, but one is not detected before the guard time expires. Right: Transient current waveform for a packet reception followed by transmission of an acknowledgement.

Examples of the first two events are shown in Figure 5.28 which were captured with a National Instruments NI-9203 current measurement card. Integrating the current above the 135 μA baseline for the idle listen with 1 ms guard time results in a charge consumption of 1.57 μC. The same integration for reception followed by an acknowledgement transmission yields 3.01 μC. Since the digital portion of the design is implemented on a FPGA, these measurements correspond only to the analog power consumption. Of the 725 μA of current consumed by the receiver during the idle listen, approximately 300 μA is consumed by the LO with the remainder dissipated in the rest of the receive chain as shown on the right in Figure 5.27.

5.10 Zero Crossing Counter Mode

The receiver is also capable of shutting down its quadrature path and operating in zero crossing counter mode. The packet error rate vs input power for this mode is shown in Figure 5.29 for a sampling frequency of 76 MHz. As shown in Section 4.12 this type of demodulator has issues with duty cycle variation, for which there is a limited amount of trim in the receiver. The result is that this mode requires substantially higher input power to achieve the same packet error rates as matched filter mode. The interference tolerance for this mode is shown in Figure 5.30. As expected zero crossing mode is again outperformed by the matched filter due to the latter's inherent frequency selectivity in the demodulator.

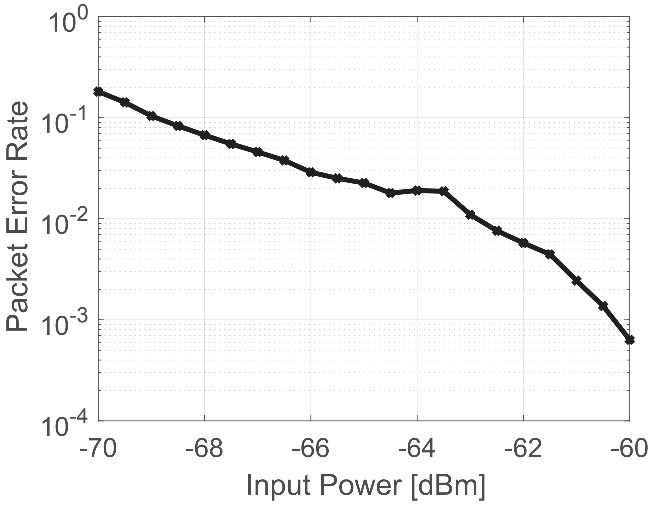


Figure 5.29: Packet error rate versus input power for zero crossing counter demodulator mode.

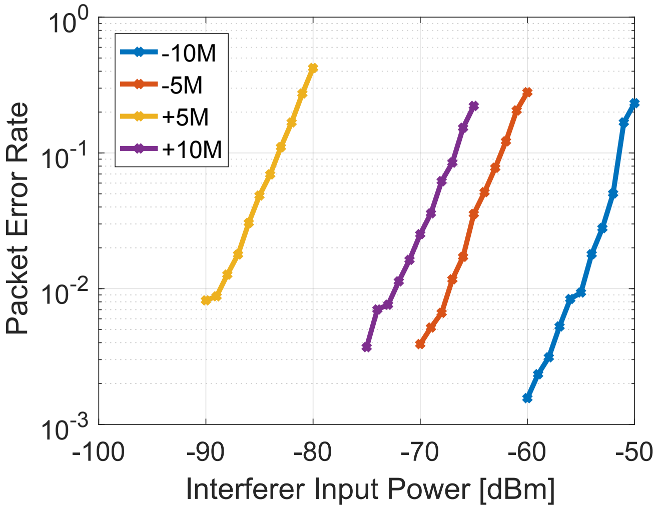


Figure 5.30: Packet error rate versus interferer power at various channel offsets for zero crossing counter demodulator mode with a desired signal input power of -60 dBm.

5.11 Baseband Outputs

There are several pieces of information obtainable from the digital baseband which can assist in the operation of a crystal-free network. The first is the estimate of the average frequency of the IF which was introduced in Section 4.5. By monitoring the number of zero crossings in a 100 μs period the baseband can tell the microprocessor on the receiver an approximation of the frequency error between the TX and RX LOs. It is then up to the software stack to determine which end of the communication link has the more trustworthy estimate of frequency and update the other accordingly. Over a span of 100 μs a 2.5 MHz IF signal will have an average of 500 zero crossings which means each tick of the reported estimate value is approximately 5 kHz. Figure 5.31 shows the reported IF estimate as input power is increased. The estimate tends to be less accurate at low input powers and will also tend to report that the IF is too high if there is interference present as that will introduce additional zero crossings. To reduce the occurrence of incorrect adjustments the average IF should be tracked across multiple packets and adjustments made on the accumulated statistics rather than any one individual packet. This frequency estimation is useful for correcting LO drift due to temperature as discussed in Section 5.13. IF frequency estimation can also be performed while operating in zero crossing demod mode and was used for temperature correction in [27] as well as for assistance in searching for RF channels during uncalibrated startup in [54].

A similar measurement can be made for the receiver chip clock rate. During zero-drift operation the CDR shifts in eight new samples and outputs a strobe to the demodulator to make a decision. To correct for chip rate error the CDR periodically clocks in either more or less than eight samples before a decision gets made (see Section 4.2 for details). By keeping track of how many extra samples are added or dropped over the course of the packet, an estimate can be obtained of the difference between the TX and RX chip rates. As with the mean IF frequency this information only applies to a relative error between the TX and RX chip rates. It is again up to the software stack to determine where to apply corrections. Figure 5.32 shows the reported number of sample point adjustments vs the error between TX and RX chip rates at -70 dBm input power for a packet length of 125 payload bytes. At an ADC sampling rate of 16 MHz each sample point is spaced by 62.5 ns. The conversion between the reported number of sample point adjustments and an estimated error value in ppm is given in Equation 5.1.

$$error\_in\_ppm = \frac{1e6 * num\_adjustments * 62.5ns}{packet\_length\_in\_bytes * 64chips/byte * 500ns/chip} \tag{5.1}$$

Another useful baseband output is the link quality indicator. In the case of a hard decision correlator the receiver knows exactly how many chip errors there were in a given packet assuming there are few enough errors that the packet was received with a correct CRC. This information can be used by the microprocessor to make decisions about the reliability of various links in the network. The baseband records the number of chip errors seen in the first four bytes of the payload data (which comes immediately after the length field) and

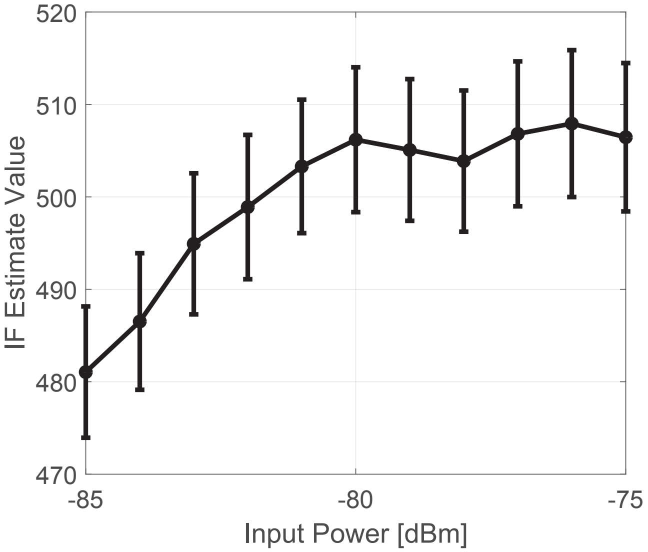


Figure 5.31: The baseband's estimate of average intermediate frequency vs input power. Each unit value corresponds to 5 kHz in frequency. The estimated value tends to be less accurate at low input powers or in the presence of interference. The ideal value of 500 corresponds to an IF of 2.5 MHz. The granularity of LO tuning steps limits how close to the ideal IF value the receiver can be tuned. In this measurement the IF can be seen to be 7-8 ticks above 500 which indicates the LO is off from its ideal value by approximately 35-40 kHz.

makes this number available to the microprocessor. The measurement is recorded during the payload rather than during the preambles so that the various feedback loops (CDR, AGC) will have settled and an accurate steady state measurement of error rate can be made. A plot of reported LQI value vs input power is shown in Figure 5.33. To convert the reported LQI value to an approximate chip error rate, divide by 256 (4 bytes is 8 payload symbols with 32 chips per symbol).

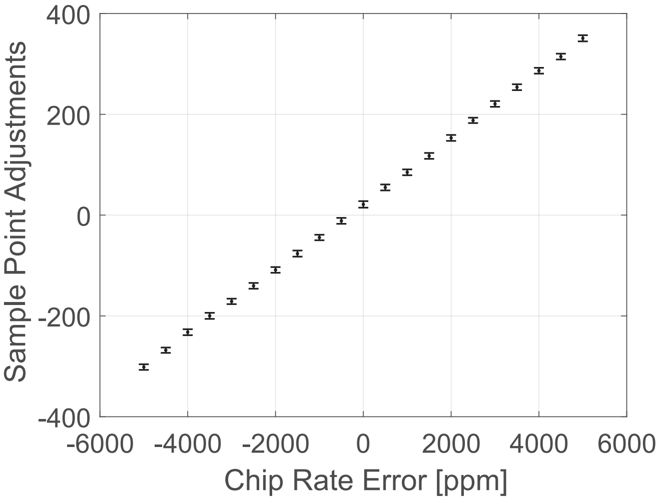


Figure 5.32: The output of the digital baseband clock and data recovery block which provides an estimate of the error between the TX and RX chip rates. This value can be converted to a ppm error estimate by taking into account the length of the packet using Equation 5.1. This sweep was performed with maximum length 125 byte payloads. Error bars are 1σ variation.

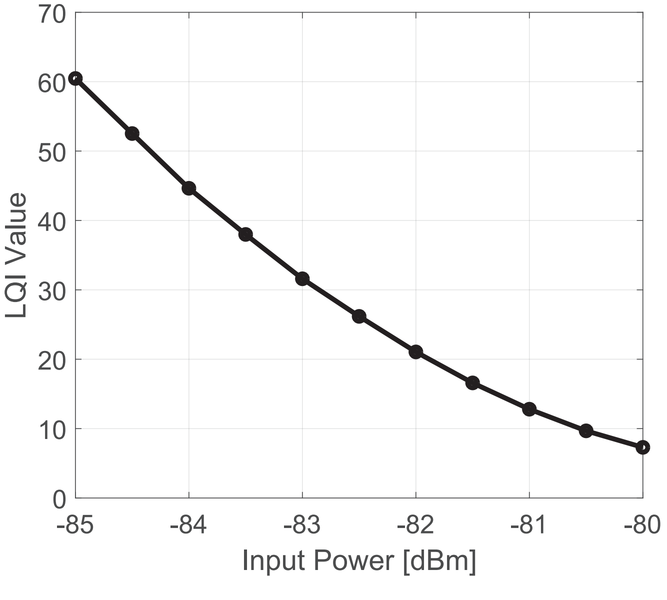


Figure 5.33: The baseband LQI value vs input power provides an estimate of the chip error rate. The LQI value is obtained by counting the number of chip errors in the first four bytes of the payload.

5.12 Frequency Calibration

Before a mote can begin to exchange packets it must first tune its clocks close enough to the correct values. Nominal settings can be applied in software so that the mote starts up with what is likely to at least be in the vicinity of the right frequencies. If calibration on a per-mote level is tolerable then these initial settings can be tailored to specific motes to counter chip to chip variation. A temperature calibration could also be done on each chip so that it has some idea of how to compensate if the environmental conditions have changed when it powers on. However it is undesirable to require this level of individualized tuning and calibration if any significant number of motes are to be deployed. It would be much more preferable to have a generalized startup algorithm that can be used on any mote to get it to join a network.

There are four clocks that the mote needs to calibrate in order to fully participate in a time synchronized channel hopping network. The first is the RF local oscillator. To be a fully functioning member of the network the mote needs to know how to tune to all 16 of the 802.15.4 channels in both transmit and receive modes. The second and third clocks which must be correct are the transmit chip clock and the receiver clock which generates the recovered chip clock. The last clock is the timer used to schedule when the radio should be turned on for communication so that it can remain in an off state the majority of the time to save power. The accuracy to with which these clocks need tuned depends on what other device the node is communicating with. The limits of communicating with the commercial 802.15.4 OpenMote [12] transceiver for example are given below. In the SCM to SCM case it is only the relative error that affects the ability to communicate. The absolute frequencies don't necessarily matter other than for adhering to the defined 802.15.4 channels and data rate. The required accuracy on the scheduling clock is dependent on the size of the guard time around the communication windows.

- SCM TX to an OpenMote:
  RF channel: ±150 ppm
  Chip rate: ±1000 ppm

- SCM RX from an OpenMote:
  RF channel: ±80 ppm
  Chip rate: ±5000 ppm

- SCM TX/RX to/from another SCM:
  RF channel: ±80 ppm
  Chip rate: ±5000 ppm

One possible solution to the initial calibration problem is to exploit the timing information that can be extracted during the bootload process. The current SCM hardware generation does not have any non-volatile memory so it must be reprogrammed every time it loses power. The mote is programmed using a contact-less optical bootloader which is discussed in detail in Appendix B. The optical transmitter used for programming is based on a microcontroller and thus has access to an accurate crystal based time reference. SCM can use its on-chip counters to compare the frequency of its free-running oscillators to the very accurate data rate of the optical programmer. The result is that at the end of the bootloading process SCM can have a relatively good calibration on all of its clock sources.

Once an initial calibration has been obtained, whether that was via the optical programmer calibration or some other method, the mote can then begin to use network traffic to keep itself synchronized. The IF estimate and receiver chip rate error estimator which were discussed in Sections 4.5 and 5.11 can be used to update the receiver chip clock and LO receive channel settings. For tuning the LO transmit channel settings and transmit chip rate the mote has a few options. The same network of on-chip counters that was used during the initial frequency calibration can also be used to periodically compare the frequency of various oscillators. For example if the receiver chip clock rate has been accurately calibrated by successfully receiving from a crystal-based node, then the TX chip rate oscillator can be calibrated against the receiver clock or divided LO. Feedback from the IF estimator on other nodes can also be fed back and used as an input for making LO tuning decisions (note that the OpenMote can measure frequency error in a manner similar to the way SCM does IF estimation). The known timing of the network schedule can also be exploited to calibrate the TSCH scheduling timer [8]. As the mote continues to exchange packets with other nodes in the network it can periodically apply updates as necessary to correct for changes in environmental conditions.

Eventually SCM may integrate NVRAM in a new hardware revision and relying on bootloading for initial calibration will become less practical. One possible solution is to power on and execute a search for network beacons that are occurring on a known channel. At initial startup the mote will need to find the right tuning codes for its LO and receive clock to be able to receive on at least this one channel. An implementation of this algorithm is discussed in [54] where the mote starts up and systematically searches for network beacons which it uses to determine when it has correctly tuned its receiver. Once the mote has found the settings to acquire packets on one channel then it can proceed to leverage timing information from network traffic to dial in its calibration and find other channels.

5.13 Temperature Tracking

A significant challenge with a crystal-free transceiver is the large temperature coefficient of its free-running oscillators. In the absence of frequency errors, the performance of the receiver itself degrades by approximately 1 dB at 70 C as seen in Figure 5.34. However any change in temperature will introduce significant drift in frequency which will result in complete transceiver failure if left uncorrected. The LO has a temperature coefficient of -40 ppm/C [17] and the two RC oscillators (Section 3.4) which provide the receiver clock and the transmit chip clock have slopes of 340 ppm/C and 160 ppm/C respectively.

To demonstrate the mote's ability to adapt to changing environmental conditions, a

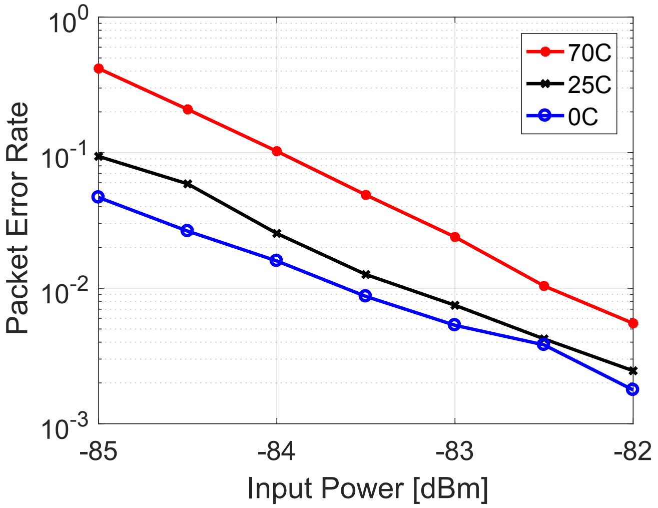


Figure 5.34: Packet error rate across the commercial 0-70 C temperature range.

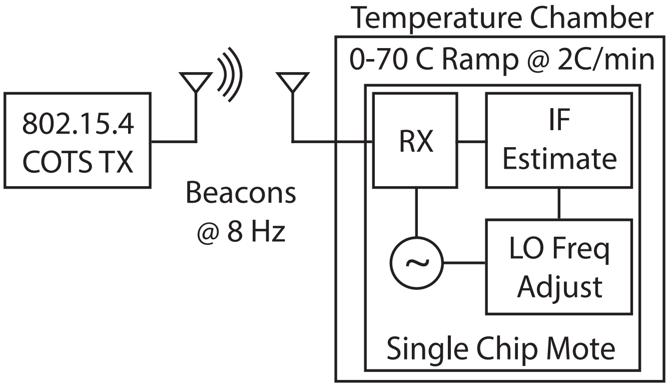


Figure 5.35: Experimental setup used to validate a SCM node's ability to use beacons from a commercial 802.15.4 device to adjust its free-running oscillators in the present of a large temperature variation.

test was conducted where the mote experienced a 2C/min temperature ramp across the commercial 0-70 C range. The test setup is depicted in Figure 5.35. A commercial Openmote [12] 802.15.4 transceiver was used to transmit beacons at a rate of 8 Hz on a single channel. This node was placed two rooms away at a straight-line distance of approximately 10 meters in an office environment with active Wi-Fi traffic. A SCM node was placed inside of a temperature chamber and conducted an initial frequency calibration at 0 C when it powered on.

After initial calibration SCM began to continuously listen for beacons. After acquiring the first beacon packet SCM used its timers to schedule and track reception of subsequent incoming packets. This allows SCM to leave its radio off for the majority of the time and only wake up to receive packets. For each received packet that had a valid CRC SCM obtained estimates of the average IF and chip rate error from the digital baseband (Section 5.11). Since the OpenMote uses a crystal based frequency reference it can be assumed that it has a much more accurate RF channel frequency and chip rate than SCM. Thus SCM assumes that all of the estimated frequency errors are occurring in its own clocks and it will make adjustments accordingly.

Due to noise there can be considerable variation in the IF estimate from the baseband as can be seen in Figure 5.31. With an 8 Hz packet rate, a -40ppm/C LO temperature coefficient, and a 2C/min temperature ramp the receiver will hear 10s of beacon packets before the uncorrected LO drift would become significant enough to begin dropping packets. This means that the receiver can use the information obtained from several packets to get a better idea of the mean value of the error estimates before it makes frequency corrections. Low pass filtering these multiple IF estimates can help to reduce incorrect LO adjustments which might occur if decisions were to be made on the basis of a single packet. The chip rate error estimate also has variation which can be filtered in the same manner.

For this test a 10-tap low-pass FIR filter was applied to the historical error estimates on the SCM node and then periodically an update was applied to the LO frequency and receiver clock rate as necessary. The number of taps to use is a tradeoff between how ideal the low-pass filter shape is, the filter latency, and how much memory and computation to dedicate to the filtering. The beacon rate, the rate of temperature change, and the number of channels being used also must be considered. These IF estimates need to be tracked individually per channel since each channel is a different LO setting and thus has its own error. The error in receiver chip rate is the same regardless of channel and so requires less complexity in managing historical estimates and updates. Ultimately the filter must be responsive enough to allow the receiver to track changes in environmental conditions before it begins dropping packets. The use of 10 taps here is based on the conditions for this experiment, but was a somewhat arbitrary choice.

The FIR filter shown in Figure 5.36 was designed using $fdatool$ in MATLAB and has a corner frequency of 0.5 Hz. The sampling rate is set by the rate of beacon packets which each provide a new sample of the IF estimate and chip rate error. If a packet is missed then the estimates for the last valid packet are re-used to prevent missing input samples. Only estimates from packets with valid CRCs are used and LQI is used to screen out packets with large numbers of chip errors as these likely encountered interference which will degrade the quality of their estimates. This is probably not the ideal filter response to use, but worked sufficiently well in this test.

The detailed results from this experiment are shown in Figure 5.37. As the temperature ramped the node was able to track the transmitted beacons and use that information to hold its own frequencies constant in the presence of changing environmental conditions. The residual error in the LO frequency is shown in (a) after the node's feedback mechanism has used the average intermediate frequency information shown in (d) to correct for drift. The data in (a) was obtained by measuring the average frequency of the LO divider output over

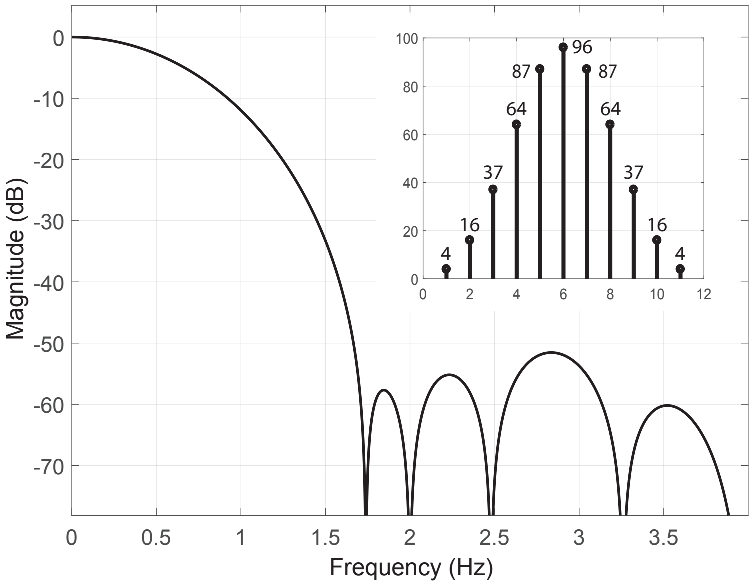


Figure 5.36: The response and coefficients of the low pass filter used to smooth out variations due to noise in the IF and chip rate error estimates. The 3 dB corner is 0.5 Hz and the sampling rate of 8 Hz is set by the beacon packet rate.

the course of the test. The data in (d) is obtained directly from the IF estimator block in the digital baseband. The minimum step size of the LO tuning is on the order of 100 kHz so the receiver is not able to always tune to exactly the right IF as can be seen by the slight DC shift above 2.5 MHz in (d). Except for a few large excursions (which will be discussed further below) and at the upper ends of the temperature range the LO mostly stays within ±40 ppm of its ideal value.

The residual error in the RX chip rate is shown in (b) after applying updates based on the CDR's reported chip rate error. The RC oscillator that generates the receiver chip clock has a step size of about 2000 ppm so the frequency is updated whenever the error reaches approximately 1000 ppm. As seen in Figure 5.24 this level of error is well within the rate tolerance of the CDR and is thus expected to not cause any degradation in performance.

The changing LO control code can be seen in (c) and has a mostly linear response with temperature. A significant challenge of tuning the LO is building software control that is monotonic and as linear as possible. The RF LO tuning is implemented with multiple DACs to provide coarse, medium, and fine range tuning which overlap to prevent gaps. The requirement to use multiple tuning DACs makes maintaining monotonicity challenging, especially across the entire temperature range. It is also not possible to have perfectly linear tuning even with a single DAC so the feedback mechanism must be able to tolerate some level of non-linearity in frequency step size. It can be observed that the larger excursions of LO error in (a) correspond to the abnormalities with monotonicity and linearity of the tuning in (c).

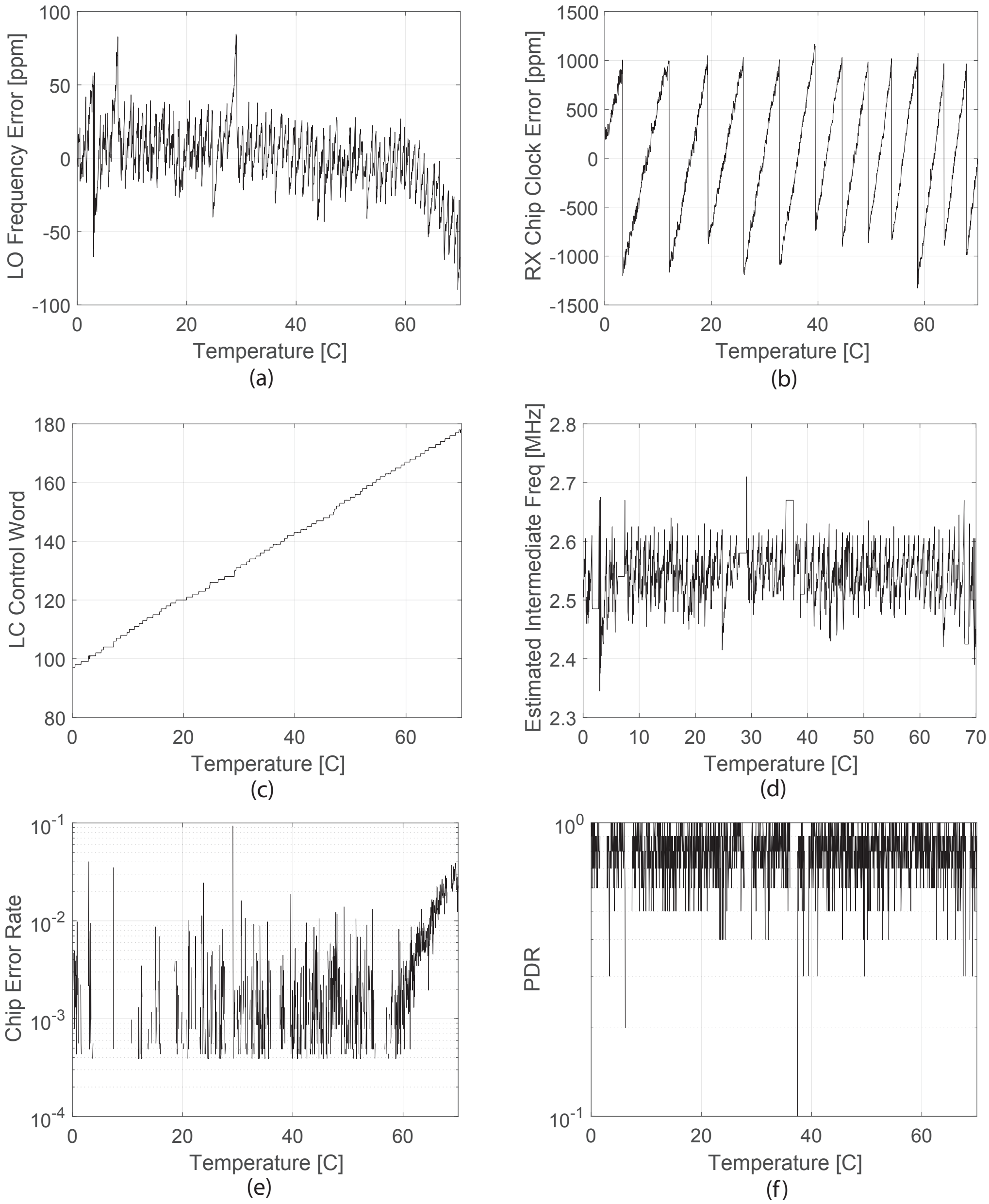


Figure 5.37: (a) RX LO frequency error after correction. (b) RX chip clock error after correction. (c) The RX LO control code is adjusted by feedback to counteract frequency drift over temperature. (d) The baseband provided estimate of the average intermediate frequency serves as the input to the LO feedback loop which attempts to hold this value constant at 2.5 MHz. (e) The resulting chip error rate over temperature as reported by the baseband LQI module. (f) The packet delivery rate over the course of the test which has an average value of 73%.

An estimate of the chip error rate is given in (e) which is obtained by dividing the LQI value from the baseband by 256 to convert to chip error rate. The receiver performance degradation at the upper end of the commercial temperature range can be seen in (e) when the chip error rate begins to degrade above 60 C. This causes a corresponding increase in the LO error as seen in (a) which does not get reflected in the IF error estimate of (d). While individual receiver blocks were designed and tested at elevated temperature, the cause of the degradation above 60 C is suspected to be due to the combined effect of many blocks reaching the end of their operational range rather than one single failure point.

The packet delivery rate over time and temperature is shown in (f) and has a mean value of 73% over the temperature range indicating that this approach works even with significant amounts of dropped packets. Given that 1) The IF estimate remains mostly within the ±200 kHz acceptable bounds of the receiver's channel tolerance, and 2) The receive chip clock error remains well within the tolerable bounds of ±5000 ppm it would seem to indicate that this PDR is likely a result of factors other than frequency tuning accuracy. It is difficult to say conclusively this is the case, but given the complex RF environment in which this test took place and use of only a single RF channel it seems reasonable. This test was conducted with the receiver set at its maximum gain setting so there is no RSSI estimate available to provide further insight.

Chapter 6

Conclusion

While the Single Chip Mote has come a long way, there is a still a tremendous amount of work that could be undertaken. Much of that work is software oriented, but there is still considerable hardware development that could be done to improve the system. While the current generations of SCM were made in a 65 nm process it likely makes sense to move to a more advanced process node at some point. When that transition occurs it will provide a reset point where many aspects of the design could be re-approached with the knowledge gained to this point. The following is a collection of thoughts on what one might do differently for future Single Chip Motes, viewed with all the benefits of hindsight.

The general theme with this version of the Single Chip Mote is that while it was interoperable with off the shelf IEEE 802.15.4 hardware it fell short of many of the specifications set forth in the standard document. Some of these shortcomings were resultant from no real design effort being placed on achieving certain specifications, such as the minimum output power level or the maximum receiver input power. These were largely ignored because they were deemed not that important in the underlying goal of achieving an operational crystal-free transceiver. If considered at the beginning of the design cycle there is no fundamental reason these could not be met, although LO pulling is a serious consideration for a free running LO and discussed further later in this chapter. Other specifications that were considered during the design process also encountered shortcomings, such as the receiver sensitivity and some of the interference scenarios. These issues stemmed from problems that could have been identified if more time was dedicated to full system level post-layout simulations. Again there is no fundamental limitation here due to the crystal-free implementation that will prevent these specifications from being met.

It becomes somewhat difficult to declare success or failure for the 802.15.4 specifications regarding frequency accuracy (which are ±40 ppm for both the RF channel and the chip rate). Without any calibration or network correction SCM is certainly not able to achieve frequency accuracy of this level with free running oscillators. If network based frequency updates are considered however then SCM does come much closer to being able to meet these specifications as well. The results in Section 5.13 showed promise for maintaining the RF LO within ±40 ppm of its intended value. The critical enabling factor was the ability to estimate the average intermediate frequency and use it to perform updates. The observed excursions outside ±40 ppm for this work were mostly attributable to the software implementation of the tuning function which mapped the coarse, medium, and fine control DACs into a single software control knob. More forethought into this implementation issue will likely yield a better solution that can maintain the RF LO within the specified frequency bounds. Maintaining the chip rate within ±40 ppm was found to not be necessary due to receivers (both SCM and COTS) which tolerate much larger errors. If one wants to achieve this level of accuracy on the chip clock for the sake of claiming standards compatibility then much design effort will need to be focused on the tuning of the free running RC oscillators which generate both the TX and RX chip clocks. It was shown with the tuning in the current transmit chip clock that the types of RC oscillators used in this work could achieve tuning with steps smaller than ±40 ppm. Temperature however remains the primary issue as this level of tuning granularity must be achievable across a wide range of temperature. Furthermore it may become difficult to detect such small errors using the CDR based chip rate error estimator presented here.

The majority of the difficulty in using the SCM crystal-free transceiver is, not surprisingly, tuning of the various clocks in the system to the correct frequencies. While various calibration and correction mechanisms can be applied in software there are a few hardware changes that could ease the burden. In the design used for SCM the required LO frequency for each channel varies depending on whether the mote is in the direct modulation transmit mode or the low IF receive mode. Furthermore frequency pulling occurs when activating the power amplifier, the polyphase filter, or the divider which affects the tuning of the LO. These effects also vary across temperature and RF channel which increases the difficulty of implementing a tuning mechanism. One solution might be to approach the design from the beginning with the intention of only needing to know one LO tuning word per channel. This for example could be achieved by moving to a direct conversion receiver and implementing the transmit modulation in such a way that in an un-modulated state the LO is at the center of the channel. This solves the inherent discrepancy between the required TX and RX LO frequencies but the pulling issues still exist. It may be possible to reduce the pulling issues by buffering the LO at the expense of additional power. An architecture based around a 2x LO that is divided down would also serve to isolate the pulling issues.

For SCM all of the frequency tuning is implemented via software. Various counters are used for comparing different clock sources and adjustments are made either via scan chain or memory mapped registers. This approach is not ideal as it requires oscillators to run for considerable amounts of time to accumulate enough clock ticks for sufficient resolution. It might make more sense to implement some of these tuning tasks in hardware using a PLL/FLL that is only activated during calibration. There is precedence for this in recent literature where a PLL is used to set the right LO frequency and is then deactivated allowing the LO to free run during the packet event [55]. While that work used an external reference for its PLL, an on-chip reference could be used as well. A viable strategy could be to tune only one on-chip clock via network traffic [8] and then reference all other clocks to the one whose calibration is being updated. That concept can be extended even further by sourcing as many on-chip clocks from the same physical oscillator as possible. For example, in SCM the microprocessor clock and the transmit chip clock come from two separate oscillators which means that both of them must be calibrated. Instead if these different clocks can be generated by multiple dividers off of a single physical oscillator then the calibration can be simplified. While this can increase the difficulty of the oscillator design due to the frequency and tuning resolution requirements, it is still a relative easy design and greatly simplifies the frequency management which must be performed by software.

While producing a transceiver that consumes as little power as possible was always a goal, it was secondary to producing a functional wireless node. This was largely a function of the magnitude of the project as well as not having a designer dedicated specifically to chip-wide power management. As a consequence SCM does not have a low power sleep mode and consumes in excess of 100 μA of DC current at all times. A significant fraction of this leakage is due to lack of consideration for a sleep mode early in the analog portions of the design. The current consumption of things like bandgap references and LDO amplifiers that are not able to be powered down begin to add up when there are multiple copies of them across the chip. For a future hardware generation there is a significant amount of power management work that could be applied to SCM, such as incorporating efficient switching regulators to lower the battery voltage as opposed to the linear regulators used on the current hardware. To be successful in implementing low power modes it has to be thought about from the very beginning of the chip design.

Pay attention to the software implications of hardware decisions. Small hardware changes can have a big impact on the ease of software development, such as seemingly simple things like the endianness of memory mapped control registers. Stick to a convention and document it well. It's incredibly frustrating in the software design to have to decipher undocumented, nonsensical hardware control registers. That's not to say that the hardware design complexity should be increased at the benefit of making programmer's lives easier. There are many times in general that it makes sense to push the complexity to software, but at least make that a conscious decision. A prime example of a common hardware to software interfacing issue is with oscillator tuning. In hardware the tuning will likely be implemented with multiple DACs of varying step sizes to cover PVT variation yielding multiple control knobs which are overlapping and likely nonlinear. From the software standpoint it is desirable to map this to a single control knob that is monotonic and as linear as possible. Giving this problem some consideration in the hardware design phase may lead to a more usable system overall.

In a project of this scope, verification is paramount. Paranoia makes working ICs; question everything. Some of the best advice I ever received from a senior grad student was "If you're not absolutely sure it's going to work, it's not." Assuming something works without aggressively testing it prior to tape-out is a recipe for wasted time and re-spins. It is far more costly in time and resources to debug silicon on the lab bench than it is to spend some extra time on verification. Build in more time for verification than you think you'll need, especially if you are incorporating blocks made by other students or undergraduates. Most designers don't skimp on verification on purpose. They underestimate their time-lines and run out of time as the tape-out deadline approaches. Pay special attention to the interfaces where circuits from different designers come together. Many of the failures I have seen over the course of this project could have been prevented by more detailed attention to these boundaries. One individual should be responsible for the overall combined circuit and a verification plan should be considered early in the design phase. Run transient noise simulations as a final sign-off and cross check any Periodic Steady State results against transient simulations. Prototype with COTS hardware when you can; why build it if you can buy it. Use FPGAs to check digital logic and always run post place-and-route simulations to check timing and functionality. You're never going to catch every last bug, but with proper planning and a little more time invested up front you'll reach your goals with less frustration and likely get there faster.

One of the next major steps for the Single Chip Mote project is getting users excited about applications. While the obvious first deployments are to simply replace existing wireless sensor nodes with SCM, this really isn't leveraging all the benefits of going crystal-free. Applications that require very large numbers of nodes to be deployed are a good candidate for SCM implementation due to their reduced part count and cost, but there is likely a considerable amount of software and networking development that needs to be done before large scale deployment becomes feasible. The more immediate impact of SCM is likely to be in applications where the size and weight of commercial wireless modules has prohibited their use. Platforms like insect sized robots (or even insects themselves) have limited payload capacity but could potentially carry a SCM chip, power source, and antenna. Adding sensors such as inertial measurement units, cameras, and microphones would then result in a mobile platform that can engage with and move about its environment. As seen with the explosive growth of software, the major source of innovation is the user community. Giving the capabilities of SCM and the required development tools to as many people as possible is the route that likely leads to the most groundbreaking use cases. A key component of that is an easy to use software development kit that allows developers to focus on software development without dealing with all the issues associated with trying to use a single silicon die by itself.

If a new SCM team were to undertake a future hardware iteration, the architecture used here could serve as an acceptable starting point although a few changes likely make sense. Sufficient noise performance to meet the 802.15.4 standard can be obtained with the passive frontend architecture used here. The crystal-free nature of the receiver does not affect the sensitivity assuming the oscillator frequencies have been appropriately tuned. Thus the standard noise vs power tradeoffs in a receiver frontend still apply to a crystal-free radio. If better performance than achievable with a passive frontend is desired and a few hundred μW of power are available for an LNA, then a design like [56] could be considered as an alternative. For the LO, in a newer process node a 4.8 GHz LO with a divide by two is probably a better choice. This mitigates the issues experienced with pulling as well as moves away from directly coupling the mixer to the LO which complicates the design process. The frontend design is still fairly interconnected between the matching network, mixer, LO, and PA so it is recommended to have either one designer handle these blocks or be very rigorous in the cross-designer verification. Dedicating one designer to the filter path and digitization would also be recommended if possible. For this work, covering from the antenna all the way through CDR with one person results in attention being spread across a wide range of sub-blocks and thus inherently it is easier to overlook issues. Also do not overlook implications of non-linearity in the receive path. For this design a limited amount of effort was put toward linearizing the receiver, instead assuming that dropped packets would be handled at the network level by re-transmission. This likely led to several missed opportunities to improve performance at only slightly higher power levels. The rest of the previous advice in this chapter is also worth heeding, especially the points related to oscillator tuning. And above all else, verify, verify, verify.

Appendix A

MATLAB Phase Noise Modeling

```
close all
clear all
clc
tic

%% Constants & Parameters
N = 2^25;           % Number of points
cn = 1e-16;         % 'c' value for white noise
cfn = 1e-10;        % 'c' value for flicker noise
Fs = 5e9;
Fc = 2.4e9;         % Carrier frequency

dF = Fs/N;
dT = 1/Fs;          % Timestep
t = 0:dT:(N-1)*dT;  % Time vector
sim_length = t(end);

%% Generate the discrete pulse response, h(k) of H(z) = 1/(1-z^-1)^(3/2)
B = -3;
h = ones(1,N);
for k = 2:1:N
    h(k) = (k - 1 - B/2) * h (k-1) / k;
end

% Iterate and average spectral densities to smooth result
total = 0;
for num_avg_fft = 1:1:5

%% Generate two independent Gaussians, w and wp
w = randn(1,N);
wp = randn(1,N);

%% Generate the white portion of phi(t)
s_white = cumsum(sqrt(cn * dT) * w);

%% Generate the flicker portion by calculating the convolution sum via DFT
% Must pad inputs with zeros since fft convolution is circular
result_length = length(h) + length(wp) - 1;
h_padded = [h zeros(1,result_length - length(h))];
wp_padded = [wp zeros(1,result_length - length(wp))];
```

```
h_fft = fft(h_padded);
wp_fft = fft(wp_padded);
result_fft_conv = ifft(h_fft .* wp_fft);

% Take the first N points
result_fft_conv = result_fft_conv(1:N);

s_flicker = sqrt(2 * pi * cfn) * dT * result_fft_conv;


%% Combine white + flicker
s_phi = 2*pi*Fc*(s_white + s_flicker);


%% Create LO with phase noise
LO = sin(2*pi*Fc*t + s_phi);

%% Mix down to baseband for plotting phase noise
% Factor of 2 accounts of gain of 1/2 through mixing operation
x = 2 * LO .* sin(2*pi*Fc*t);

% Filter off high frequency mixing product
[b,a] = butter(2,20e6/(Fs/2));
noisyIF = filter(b,a,x);

%% Calculate PSD
N = length(x);
dF = Fs/N;
freq = -Fs/2:dF:Fs/2-dF;
xdft = fft(x,N)/N;
xdft = fftshift(xdft);
psdx = abs(xdft).^2;
psdx_db = 10*log10(psdx);

total = total + psdx_db;

end

psdx_db = total / num_avg_fft;
```

```
%% Plot the one-sided PSD
set(gca,'FontSize',14)
semilogx(dF:dF:(N/2+1)*dF,6+psdx_db((length(psdx_db)/2):end) - 10*log10(dF))
title('PSD of Noisy LO')
xlabel('Frequency [Hz]')
ylabel('RF PSD [dBc/Hz]')
axis([1e3 1e7 -140 -20])
grid on

toc
```

Appendix B

Optical Bootloader Design

B.1 Introduction

Integrated systems require some way to transfer configuration and program data to on-board memory. Wired programmers are inconvenient and become less practical as systems scale down in size to monolithic proportions. Radio based bootloaders provide a wireless solution, but at the expense of significant power. The addition of an integrated optical receiver requires much less power than a RF receiver and consumes little additional area.

A small, low power optical receiver is presented here for transferring program data to SRAM. An integrated photodiode is combined with an inverter-based amplifier design to generate a digital waveform from optical input. A pulse width modulation scheme is used to allow clock and data recovery (CDR) to operate without any on-chip clock source. The receiver is able to achieve a data rate of 320 kbps for active and standby powers of 1.52 μW and 640 nW, respectively, while occupying 16,900 μm^2.

Significant credit for the implementation of the analog portion of the optical bootloader goes to Andy Ng who did both the schematic design and layout. Without his contributions the ability to optically program Single Chip Motes would likely not have happened.

B.2 System Design

The use case goal for the Single Chip Mote is to enable 64 kB of program memory to be wirelessly programmed in on the order of 1 s, thus requiring a data rate in the 100s of kbps. The goal was to transfer the majority of the programming burden to an external USB-powered optical programmer in order to reduce the complexity and power required on-chip. A self-clocked approach to CDR was used to eliminate the requirement for a receiver clock which simplifies the receiver and allows more of the power budget to be spent in the frontend amplifier. The receiver is required to work from a default configuration at cold start with no trim and to reject DC offsets in bright lighting conditions. Optical programming for this design is only required to be done under normally varying lab conditions near room temperature.

B.3 Clock and Data Recovery

The self-clocked CDR shown in Figure B.1 is based on [57] where a pulse width modulation scheme is used to implement an optical wake-up receiver. In this scheme short and long pulses are time delayed and then used as clock edges to sample the incoming pulse stream. The time delay in [57] is implemented by sampling the incoming data stream with a clock running at less than 1 kHz and then delaying for an integer number of clock cycles. The use of such a low clock frequency reduces power consumption but limits data rate to below 100 bps which is insufficient for this application. Instead of increasing the frequency this work avoids an on-chip clock for sampling by implementing a self-clocking strategy based on an analog delay. The pulse stream coming from the RX frontend is passed through a delay cell

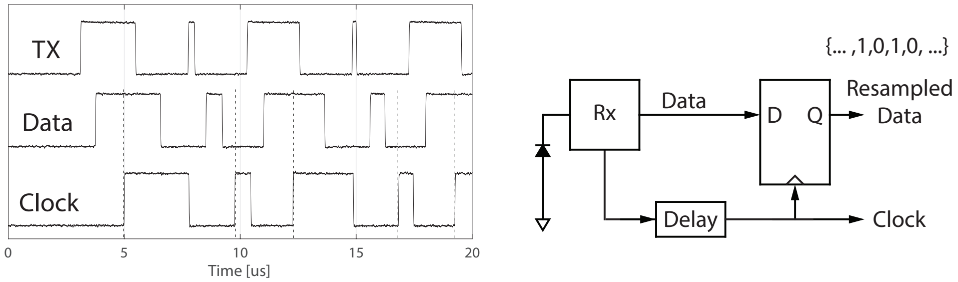


Figure B.1: Block diagram and measured transient waveforms for the RX frontend and self-clocked data recovery.

built from a chain of current-starved inverters and then combined with a single flip-flop to form the complete CDR.

B.4 Circuit Design

Diode

The frontend uses a deep N-well to substrate diode with an area of 2510 μm^2. The diode was measured to have a responsivity of 0.21 A/W to IR light at 850 nm. The diode is held at a reverse bias of $\mathrm{V_{DD}}/2$ by the self-biased inverter transimpedance amplifier (TIA). The simulated junction capacitance at this bias was 2.26 pF which agrees within 10% of the measured value for this junction type of 1 fF/μm^2. The external optical programmer used in this system is based on a SFH4555 IR emitter with irradiance of 550 mW/sr and has a target range of less than 10 cm.

Transimpedance Amplifier

A TIA architecture was chosen based on [58] and was designed using a self-biased inverter for its simplicity and current reuse. The feedback simplifies biasing of the photodiode and allows for higher data rates than a high impedance design as in [57]. The current limiting devices for both the amplifier and the analog delay cells are biased using constant g_m circuits. DC offsets from ambient light are reduced by AC coupling the TIA to the second gain stage, which is based on the baseband amplifier design in [37]. The resistor used in the AC coupling is implemented with a FET which has a small-signal resistance at DC of nearly 1 $G\Omega$, however the large voltage swings at its terminals make this number not especially informative. The simulated small-signal transfer functions of the first and second amplifier stages are shown on the left in Figure B.3 along with the simulated noise at the second stage output on the right. The transfer function peaks at 154 dB of transimpedance and the output referred noise integrated from 1 Hz to 10 MHz is 10.7 mV. The sizing of the first inverter following

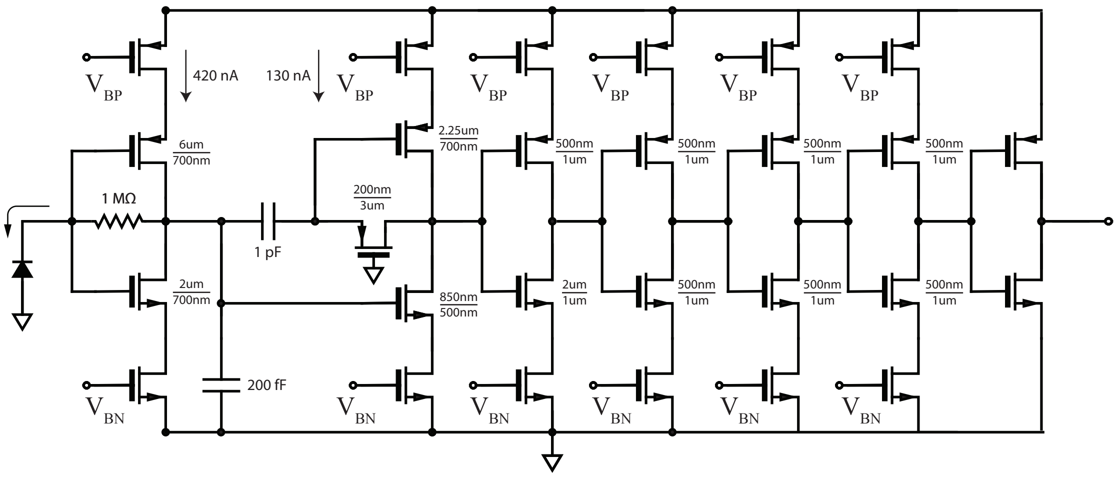


Figure B.2: Schematic of frontend TIA, bandpass filter, and biased inverters.

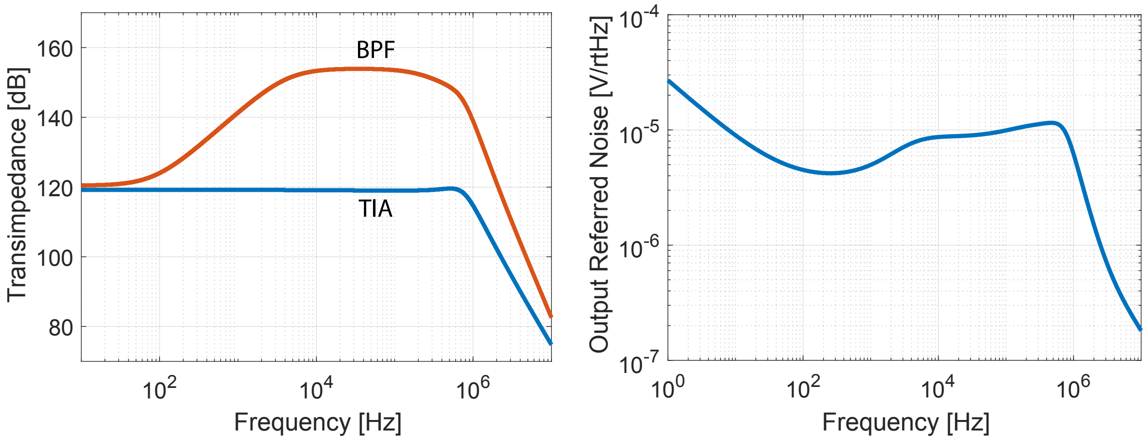


Figure B.3: Left: Simulated transfer function for the schematic of the first and second amplifier stages which convert the photodiode current to an output voltage. Right: Output referred noise simulation of the two combined amplifier stages.

the bandpass filter stage is intentionally skewed so that when no input light is present the output of the inverter is near ground. This causes the following stages to operate as digitally switched inverters rather than linear gain stages. Not only does this save power by avoiding the biasing of additional devices in the linear region of operation, it also reduces the amount of output toggling that occurs due to noise as only excursions above the trip point of the skewed inverter propagate to the output. It should be noted that transient simulations are essential to the design and verification of this frontend due to the highly non-linear large signal operation.

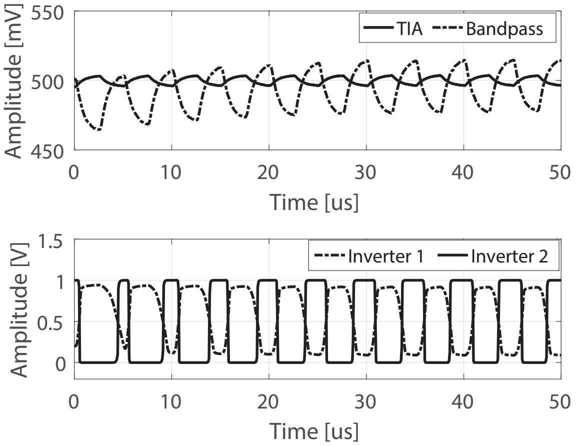


Figure B.4: Example of the pulse width expansion that occurs as the AC coupling in the frontend settles to a new value for a long string of unchanging input data values.

Delay Cell

The analog delay cell consists of 16 current-starved inverters like those in Figure B.2. The nominal target delay value t_d is 1 μs to enable data rates in the 100s of kbps. Only one corner of devices was received from fabrication so measurement results are unavailable for process variation, but simulations indicate that the process-dependent shift in t_d tracks the shift in pulse widths, minimizing the effect on the CDR.

B.5 Digital Backend

The recovered clock and data bit stream is routed to a digital block which provides the programming interface to the microprocessor. This block performs three main functions: 1) Search for a start sequence which indicates that the digital system should execute a hard reset to prepare for bootloading 2) Perform 4B/5B decoding on the incoming data stream to recover the original program data and 3) Convert the serial bit stream to a format equivalent with the pre-existing three-wire programming bus which loads program data into instruction memory.

The 4B/5B coding step is used to DC balance the incoming data stream to avoid transient issues caused by the AC coupling in the receiver settling to data-dependent values. Compiled program binaries often have long strings of unchanging bits which cause changes in the pulse width outputs of the frontend as shown in Figure B.4. If left uncorrected this pulse expansion will cause incorrect reception of long strings of zero bits as the pulse width will expand beyond the delay value and begin registering as ones instead of zeros.

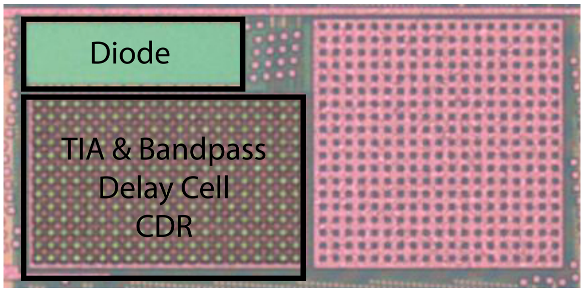


Figure B.5: Die photo of the optical receiver which occupies 16,900 um^2.

B.6 Measurement Results

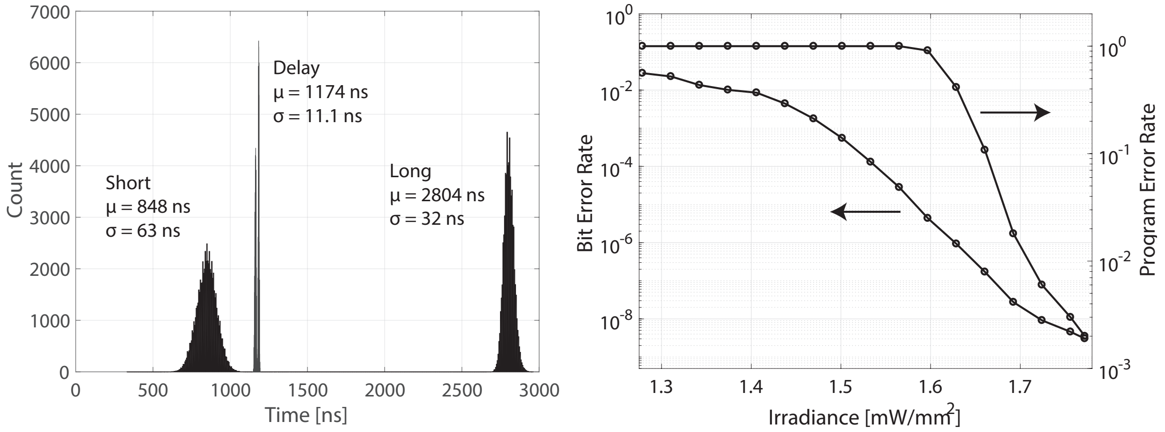
The receiver was fabricated in TSMC 65LP and the die photo is shown in Figure B.5. The total design including the photodiode, amplifiers, delay cell, and CDR occupies 16,900 μm^2. A FPGA-based TDC [59] was used to measure individual pulses' width and delay. The optical programmer was built using a SFH4555 IR emitter LED driven by a micro-controller able to generate pulses with widths controllable in increments of 5.6 ns. Long strings of unchanging data cause shifts in DC bias which lead to changes in pulse width over time. The following measurements use 4B5B encoding to avoid this issue by DC balancing the data stream.

The active power consumption was measured to be 1.52 μW and the standby power was 640 nW. Figure B.6(a) is an example histogram obtained via the TDC setup showing the distribution of short and long pulses as well as the delay time. The mean delay is about 20% longer than the simulated value of 1 μs. The distributions are approximately as expected from simulation although it is difficult to simulate long enough transient sequences to gather similar amounts of data. Figure B.6(b) shows the measured Bit Error Rate and Program Error Rate (PER defined as the occurrence of a single bit error in a 64 kB payload) for a given input irradiance. An irradiance of 1.7 mW/mm^2 is sufficient to ensure fewer than one payload bit error per one hundred programming cycles.

Supply Voltage and Temperature

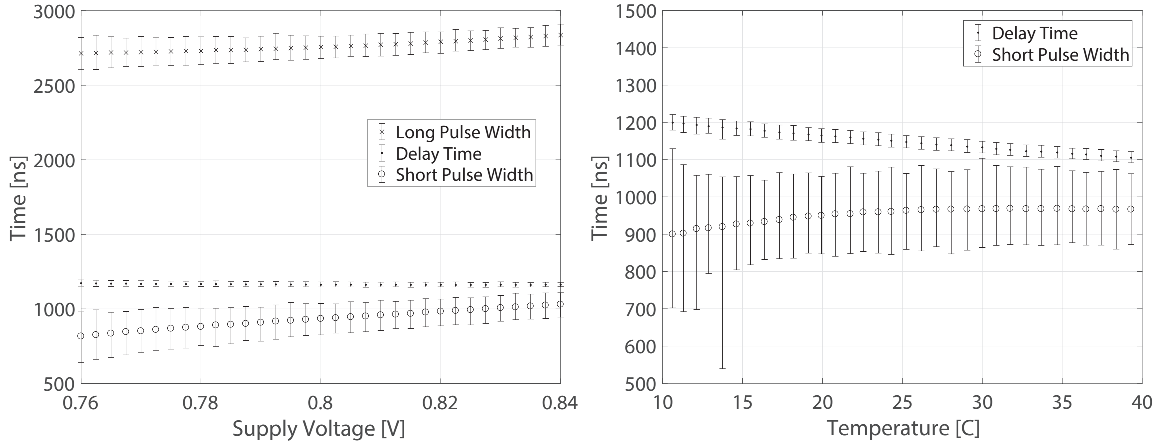
The receiver is designed to operate at a nominal supply voltage of 0.8 V. The effect of varying supply voltage on pulse widths and delays is shown in Figure B.7(a). Error bars indicate the maximum and minimum values recorded during a 4 kB 4B5B encoded payload. The slope of the change in delay time is 125 ns/V which is insignificant when compared to the variation in pulse widths from the TIA. Further excursions away from nominal supply point can be tolerated by adjusting the pulse widths from the transmitter. The receiver's tolerance of interference on the supply node is shown in Figure B.8.

Figure B.7(b) shows the variation of the short pulse widths and delay times across temperature. Again the variation of the pulse widths from the TIA are the limiting factor and



(a) (b)

Figure B.6: (a) Histogram showing measurement results for short and long pulses and delay time. (b) BER and PER vs irradiance for 64 kB 4B5B payloads.



(a) (b)

Figure B.7: Pulse and delay variation due to (a) supply voltage and (b) temperature.

will cause zeros to get clocked as ones at elevated temperatures when the width of the short pulse exceeds the delay time. The slope of the delay variation with temperature is 3.4 ns/°C.

B.7 Future Iterations

Possible future improvements to the optical bootloader fall roughly into two categories: additional features and performance improvements. This section outlines a few ideas that surfaced during the course of the design and testing of the current hardware generation that could be explored in the future.

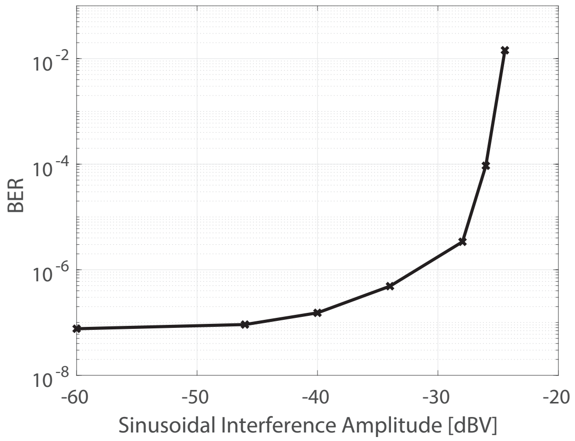


Figure B.8: Measured bit error rate as a function of sinusoidal interference injected at the supply node.

The three primary trade-offs involved in designing an optical bootloader are power, speed, and programming range. The choice was made during the design of this generation of the optical bootloader to target as low power operation as possible while still retaining on the order of one second programming time. The consequence is that a relatively high power LED is needed for the programmer and the range is short (<10 cm). If it is desired to improve speed or programming range then the receiver could simply be redesigned to meet those goals at the expense of more power. Since the optical receiver needs to essentially always be listening in order to perform its function, it is undesirable to burn a considerable amount of power. However, during an actual programming event the receiver is much less power constrained as programming happens relatively infrequently. One possible way to break the power/speed/range trade-off is to split the optical bootloader into separate wake-up and high-performance modes. In this approach a low power, low data rate optical receiver could run at all times and search for a start sequence. When the start sequence is found the high-performance receiver could be activated by the backend state machine for the duration of the payload reception. These two modes could be physically distinct receivers or it may be possible to combine the two by having the ability to activate additional inverters in the frontend to increase the transconductance. The combined approach benefits from requiring area for only one photodiode.

Another potential source for performance improvements is to consider the similarities to other high speed serial link designs. The pulse width modulation scheme used for the optical bootloader is dependent on receiving narrow pulses. The bandwidth of these input light pulses generally exceeds the bandwidth of the receiver and as a consequence some pulse width expansion occurs. Similar issues are encountered in high-speed wireline designs where it is desirable to restore the shape of incoming pulses that are distorted by the limited bandwidth of the channel. High-speed wireline designs restore the pulse shapes using equalization which enables higher data rates to be achieved by the link. It may be possible to adopt similar feedback equalization techniques in an optical bootloader implementation to reduce the width of the output pulses despite the limited bandwidth of the frontend TIA.

The current optical receiver operates by transferring the full 64 kB contents of the instruction memory during every program event. When the compiled binary is smaller than this size it is padded with zeros prior to transfer. This essentially wastes time by transferring data padding which serves no purpose other than simplifying the on-chip state machine. Adding the ability for the digital backend to handle variable payload lengths and perform the zero padding on-chip could reduce the programming time in many cases.

One additional feature that could potentially be useful is the ability to program the analog scan chain with the optical receiver. This would eliminate the need to connect the six analog scan chain pads (in, out, clk, clk_bar, out, mux_select) in order to externally configure the scan chain. Of course if the complete system is functioning as expected then software can be bootloaded which will perform the desired analog scan chain programming directly from the Cortex-M0. Having an alternative contact-less path for programming the scan chain provides a backup way to interface to the chip in the event the boot mechanism is non-functional. If this feature were to be pursued, extreme care should be given to ensure that no analog scan chain problems are inadvertently introduced as this forms the critical control core of the entire chip.

Another additional feature that may be worth considering is to add hardware support for using the optical receiver to assist in localization using a system like the HTC Lighthouse. This system uses timed sweeps of infrared lasers to localize devices used in virtual reality environments. Since the optical bootloader already contains an IR receiver it may be possible to re-purpose it for this additional task without significant hardware modifications. The primary piece of hardware to consider adding would likely be connecting hardware timers and control logic to the optical receiver output so pulse timing information could be captured and provided to the microprocessor.

Regardless of whether improvements are being made to the design or it is just being ported to a new process, it is highly encouraged to prototype the design in hardware prior to tape-out when possible. The bootload process is one of the most critical aspects of the SoC and requires a large number of steps to go right in order. Prototyping the complete bootload system in advance ensures that as many issues as possible are identified and fixed prior to silicon. The analog portion of the design can either be built using similar COTS parts or by re-using the current frontend on existing chips. The optical backend FSM and full digital system should be implemented on FPGA and interfaced with the analog frontend to undergo a full end-to-test from programmer to SRAM.

Bibliography

[1] Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, Qin Wang, Steven Glaser, and Kris Pister. "OpenWSN: a standards-based low-power wireless development environment". In: *Transactions on Emerging Telecommunications Technologies* 23.5 (2012), pp. 480–493.

[2] Thomas Watteyne, Ankur Mehta, and Kris Pister. "Reliability through frequency diversity: why channel hopping makes sense". In: *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM. 2009, pp. 116–123.

[3] Thomas Watteyne, Steven Lanzisera, Ankur Mehta, and Kristofer SJ Pister. "Mitigating multipath fading through channel hopping in wireless sensor networks". In: *Communications (ICC), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1–5.

[4] Brian Kilberg, Craig B Schindler, Arvind Sundararajan, Alex Yang, and Kristofer SJ Pister. "Experimental Evaluation of Low-Latency Diversity Modes in IEEE 802.15.4 Networks". In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. IEEE. 2018, pp. 211–218.

[5] Alex Yang, Arvind Sundararajan, Craig B Schindler, and Kristofer SJ Pister. "Analysis of low latency TSCH networks for physical event detection". In: *Wireless Communications and Networking Conference Workshops (WCNCW), 2018 IEEE*. IEEE. 2018, pp. 167–172.

[6] Tengfei Chang, Thomas Watteyne, Kris Pister, and Qin Wang. "Adaptive synchronization in multi-hop TSCH networks". In: *Computer Networks* 76 (2015), pp. 165–176.

[7] Analog Devices. *LTC5800-IPM*. original document from Analog Devices. 2013. URL: `http://www.analog.com/media/en/technical-documentation/data-sheets/5800ipmfa.pdf`.

[8] Osama Khan, Brad Wheeler, David Burnett, Filip Maksimovic, Sahar Mesri, Kris Pister, and Ali Niknejad. "Frequency reference for crystal free radio". In: *Frequency Control Symposium (IFCS), 2016 IEEE International*. IEEE. 2016, pp. 1–2.

[9] "IEEE Standard for Low-Rate Wireless Networks". In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (Apr. 2016), pp. 1–709.

[10] John Notor, Anthony Caviglia, and Gary Levy. "CMOS RFIC architectures for IEEE 802.15.4 networks". In: *Cadence Design Systems, Inc* 41 (2003).

[11] Rohde & Schwarz. *Generation of IEEE 802.15.4 Signals.* URL: `https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1gp105/1GP105_1E_Generation_of_IEEE_802154_Signals.pdf`.

[12] Xavier Vilajosana, Pere Tuset, Thomas Watteyne, and Kris Pister. "OpenMote: Open-source prototyping platform for the industrial IoT". In: *International Conference on Ad Hoc Networks*. Springer. 2015, pp. 211–222.

[13] Texas Instruments. *CC2538 SoC*. 2013. URL: `http://www.ti.com/lit/ds/symlink/cc2538.pdf`.

[14] Sahar Mesri. "Design and user guide for the single chip mote digital system". MA thesis. EECS Department, University of California, Berkeley, 2016.

[15] Asad A Abidi. "Phase noise and jitter in CMOS ring oscillators". In: *IEEE Journal of Solid-State Circuits* 41.8 (2006), pp. 1803–1816.

[16] David C Burnett, Brad Wheeler, Filip Maksimovic, Osama Khan, Ali M Niknejad, and Kristofer SJ Pister. "Narrowband communication with free-running 2.4 GHz ring oscillators". In: *Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), 2017 International Conference on*. IEEE. 2017, pp. 1–6.

[17] Filip Maksimovic. "Monolithic Wireless Transceiver Integration". PhD thesis. EECS Department, University of California, Berkeley, 2018.

[18] Lauri Anttila, Mikko Valkama, and Markku Renfors. "Blind compensation of frequency selective I/Q imbalances in quadrature radio receivers: Circularity-based approach". In: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. Vol. 3. IEEE. 2007, pp. III–245.

[19] Edward A Lee and David G Messerschmitt. *Digital communication*. Springer Science & Business Media, 2012.

[20] Steven Lanzisera and Kristofer SJ Pister. "Theoretical and practical limits to sensitivity in IEEE 802.15.4 receivers". In: *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*. IEEE. 2007, pp. 1344–1347.

[21] Thomas Kho. *Steganography in the 802.15.4 physical layer*. Tech. rep. 2007.

[22] Alper Demir, Amit Mehrotra, and Jaijeet Roychowdhury. "Phase noise in oscillators: A unifying theory and numerical methods for characterization". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47.5 (2000), pp. 655–674.

[23] David C Lee. "Modeling timing jitter in oscillators". In: *Proc. Forum Design Languages*. 2001, pp. 3–7.

[24] Osama Khan, Brad Wheeler, Filip Maksimovic, David Burnett, Ali M Niknejad, and Kris Pister. “Modeling the impact of phase noise on the performance of crystal-free radios”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 64.7 (2017), pp. 777–781.

[25] Alfio Zanchi. “How to Calculate the Period Jitter σT from the SSCR L (f n) with Application to Clock Sources for High-Speed ADCs”. In: *Application Note* (2003).

[26] N Jeremy Kasdin and Todd Walter. “Discrete simulation of power law noise (for oscillator stability evaluation)”. In: *Frequency Control Symposium, 1992. 46th., Proceedings of the 1992 IEEE*. IEEE. 1992, pp. 274–283.

[27] Brad Wheeler, Filip Maksimovic, Nima Baniasadi, Sahar Mesri, Osama Khan, David Burnett, Ali Niknejad, and Kris Pister. “Crystal-free narrow-band radios for low-cost IoT”. In: *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE. 2017, pp. 228–231.

[28] David Burnett. “Crystal-free wireless communication with relaxation oscillators and its applications”. PhD thesis. EECS Department, University of California, Berkeley, 2019.

[29] Rolf Schaumann and Mac Elwyn Van Valkenburg. *Design of analog filters*. Vol. 1. Oxford University Press New York, 2001.

[30] Hercules G Dimopoulos. *Analog electronic filters: Theory, design and synthesis*. Springer Science & Business Media, 2011.

[31] John G Proakis and Dimitris G Manolakis. *Digital Signal Processing: Principles, Algorithms, and Edition*. 1995.

[32] Steven W Smith et al. “The scientist and engineer’s guide to digital signal processing”. In: (1997).

[33] Caroline Andrews and Alyosha C Molnar. “A passive-mixer-first receiver with baseband controlled RF impedance matching, <6dB NF, and >7dBm wideband IIP3”. In: *2010 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2010, pp. 46–47.

[34] Caroline Andrews and Alyosha C Molnar. “A passive mixer-first receiver with digitally controlled and widely tunable RF interface”. In: *IEEE Journal of solid-state circuits* 45.12 (2010), pp. 2696–2708.

[35] Caroline Andrews and Alyosha C Molnar. “Implications of passive mixer transparency for impedance matching and noise figure in passive mixer-first receivers”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 57.12 (2010), pp. 3092–3103.

[36] Sashank Krishnamurthy, Filip Maksimovic, and Ali M Niknejad. “580μW 2.2-2.4 GHz Receiver with +3.3 dBm Out-of-Band IIP3 for IoT Applications”. In: *ESSCIRC 2018-IEEE 44th European Solid State Circuits Conference (ESSCIRC)*. IEEE. 2018, pp. 106–109.

[37] Ben W Cook, Axel Berny, Alyosha Molnar, Steven Lanzisera, and Kristofer SJ Pister. "Low-power 2.4-GHz transceiver with passive RX front-end and 400-mV supply". In: *IEEE Journal of Solid-State Circuits* 41.12 (2006), pp. 2757–2766.

[38] Dong Yang, Caroline Andrews, and Alyosha Molnar. "Optimized design of N-phase passive mixer-first receivers in wideband operation". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.11 (2015), pp. 2759–2770.

[39] Yuanching Lien, Eric Klumperink, Bernard Tenbroek, Jon Strange, and Bram Nauta. "A high-linearity CMOS receiver achieving +44dBm IIP3 and +13dBm B1dB for SAW-less LTE radio". In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE. 2017, pp. 412–413.

[40] Massoud Tohidian, Iman Madadi, and Robert Bogdan Staszewski. "A 2mW 800MS/s 7th-order discrete-time IIR filter with 400kHz-to-30MHz BW and 100dB stop-band rejection in 65nm CMOS". In: *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*. IEEE. 2013, pp. 174–175.

[41] Ahmad Mirzaei, Saeed Chehrazi, Rahim Bagheri, and Asad A Abidi. "Analysis of first-order anti-aliasing integration sampler". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 55.10 (2008), pp. 2994–3005.

[42] P Payandehnia, H Maghami, M Kareppagoudr, and GC Temes. "Passive switched-capacitor filter with complex poles for high-speed applications". In: *Electronics Letters* 52.19 (2016), pp. 1592–1594.

[43] P Payandehnia, H Maghami, H Mirzaie, M Kareppagoudr, S Dey, M Tohidian, and GC Temes. "A 0.49–13.3 MHz tunable fourth-order LPF with complex Poles achieving 28.7 dBm OIP3". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.8 (2018), pp. 2353–2364.

[44] Sevil Zeynep Lulec, David A Johns, and Antonio Liscidini. "A simplified model for passive-switched-capacitor filters with complex poles". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 63.6 (2016), pp. 513–517.

[45] Renaldi Winoto. "Downconverting Sigma-Delta A/D Converter for a Reconfigurable RF Receiver". PhD thesis. EECS Department, University of California, Berkeley, 2009.

[46] Chun-Cheng Liu, Soon-Jyh Chang, Guan-Ying Huang, and Ying-Zu Lin. "A 10-bit 50-MS/s SAR ADC with a monotonic capacitor switching procedure". In: *IEEE Journal of Solid-State Circuits* 45.4 (2010), pp. 731–740.

[47] Danielle Griffith, Per Torstein Røine, James Murdock, and Ryan Smith. "A 190nW 33kHz RC oscillator with ±0.21% temperature stability and 4ppm long-term stability". In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE. 2014, pp. 300–301.

[48] Elad Alon. "Measurement and Regulation of On-Chip Power Supply Noise". PhD thesis. EE Department, Stanford University, 2006.

[49] Umberto Mengali. *Synchronization techniques for digital receivers.* Springer Science & Business Media, 2013.

[50] Aldo N D'Andrea, Urnberto Mengali, and Ruggero Reggiannini. "A digital approach to clock recovery in generalized minimum shift keying". In: *IEEE Transactions on Vehicular Technology* 39.3 (1990), pp. 227–234.

[51] Pei-Hsueh Lee, Ho-Ching Chao, Wei-Lung Mao, Hen-Wai Tsao, and Fan-Ren Chang. "The Effects of I/Q Imbalance and Complex Filter Mismatch on GPS/Galileo System". In: *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007).* 2001, pp. 543–550.

[52] Massoud Tohidian, Iman Madadi, and Robert Bogdan Staszewski. "A 2mW 800MS/s 7th-order discrete-time IIR filter with 400kHz-to-30MHz BW and 100dB stop-band rejection in 65nm CMOS". In: *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers.* IEEE. 2013, pp. 174–175.

[53] Behzad Razavi. "A study of injection locking and pulling in oscillators". In: *IEEE journal of solid-state circuits* 39.9 (2004), pp. 1415–1424.

[54] Ioana Suciu, Filip Maksimovic, David Burnett, Osama Khan, Brad Wheeler, Arvind Sundararajan, Thomas Watteyne, Xavier Vilajosana, and Kris Pister. "Experimental Clock Calibration on a Crystal-Free Mote-on-a-Chip". In: *IEEE International Conference on Computer Communications. CNERT: Computer and Networking Experimental Research using Testbeds.* 2019.

[55] Masoud Babaie, Feng-Wei Kuo, Huan-Neng Ron Chen, Lan-Chou Cho, Chewn-Pu Jou, Fu-Lung Hsueh, Mina Shahmohammadi, and Robert Bogdan Staszewski. "A fully integrated Bluetooth low-energy transmitter in 28 nm CMOS with 36% system efficiency at 3 dBm". In: *IEEE Journal of Solid-State Circuits* 51.7 (2016), pp. 1547–1565.

[56] Mustafijur Rahman and Ramesh Harjani. "A sub-1V, 2.8 dB NF, 475μW coupled LNA for internet of things employing dual-path noise and nonlinearity cancellation". In: *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC).* IEEE. 2017, pp. 236–239.

[57] G Kim, Y Lee, S Bang, I Lee, Y Kim, D Sylvester, and D Blaauw. "A 695 pW standby power optical wake-up receiver for wireless sensor nodes". In: *Custom Integrated Circuits Conference (CICC), 2012 IEEE.* IEEE. 2012, pp. 1–4.

[58] Brian S Leibowitz, Bernhard E Boser, and Kristofer SJ Pister. "CMOS smart pixel for free-space optical communication". In: *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II.* Vol. 4306. International Society for Optics and Photonics. 2001, pp. 308–319.

[59] Tian Xiang, Lei Zhao, Xi Jin, Tianqi Wang, Shaoping Chu, Cong Ma, Shubin Liu, and Qi An. "A 56-ps multi-phase clock time-to-digital convertor based on Artix-7 FPGA". In: *Real Time Conference (RT), 2014 19th IEEE-NPSS.* IEEE. 2014, pp. 1–4.