Crystal-free wireless communication with relaxation oscillators and its applications

David Burnett

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-5
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-5.html

April 12, 2019

Acknowledgement

My group, SCuM/OpenWSN, technical guides, postdoc collaborators, neighbors in the Swarm lab, interested sponsors, U of WA staff and faculty connections, my sanity-restoring Berkeley social circle, BSAC, the EECS department administration and EECS & BWRC IT, NDSEG at ASEE, my outside friends from the old days, outside mentors, and -- of course -- Kris. (Full acknowledgements in the dissertation.)

Crystal-free wireless communication with relaxation oscillators and its applications

by

David C Burnett

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 Lance J. Kriegsfeld

Spring 2019

Crystal-free wireless communication with relaxation oscillators and its applications

Abstract

Crystal-free wireless communication with relaxation oscillators and its applications

by

David C Burnett

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kristofer S.J. Pister, Chair

For the last decade, the size of complete 2.4GHz wireless modules containing everything but power have stagnated at ~1cm x 1cm. This is despite continued advances in semiconductor processes due to components needed by the core communication IC. Breaking this size barrier (which also sets a power and cost barrier) by eliminating all off-chip components is the goal of the Single-Chip Mote project, of which this dissertation is a part. The major components to be eliminated are antenna, battery, and crystal oscillator. Without these components, a complete wireless module could be the size of the RFIC silicon a few millimeters on a side, or less, instead of the size of the supporting PCB.

Once those components are eliminated, advances in process scaling will lead to another size floor defined by the size of inductors, which do not scale with process. Inductor-based oscillators also put a floor on power consumption (dictated by desired swing versus achievable inductor quality factor) and, not being representable by digital cells, cannot be part of an integrated synthesis flow. In light of these limitations, understanding to what extent relaxation oscillators (the non-resonant oscillator family including RC oscillators and ring oscillators) can be used for communication is necessary to establish inductor-free performance limits and, more importantly, is necessary for future Single-Chip Mote scaling.

Contrary to general opinion, FSK communication systems based on free-running RF ring oscillators do not exhibit catastrophically poor performance but are capable of good packet delivery rate (PDR 99% or better) with moderately higher tone spacing compared to typical low-power wireless specifications (~2x when communicating with COTS base station, ~6x when communicating to another ring-based radio). Methods to work around the high jitter of rings are established; this jitter is dominated by flicker noise and is poorly represented by simulators, making accurate simulation-based design difficult. Communication and complete sensor systems can be demonstrated, and system-level chip design at the core of this work can be taught to successive generations by adopting an industrial project-based approach.

Updates and errata to this dissertation can be found at:
`http://people.eecs.berkeley.edu/~db/dissertation/`

To my brother

For whom I hope I've cleared a path

Contents

List of Figures

List of Tables

Acknowledgments

With apologies to everyone I forgot to add as I finish this, the very last thing in my dissertation, and acknowledge:

My fellow "Pisterbot" labmates, past and present. My generation: Joey Greenspun, Brad Wheeler, Fil Maksimovic, Daniel Drew, and Dan Contreras. The next generations: Craig Schindler, Brian Kilberg, Hani Gomez, Lydia Lee, Alex Moreno, and Nathan Lambert. And the previous generations: Fabien Chraim, Mike Lorek, Travis Massey, Ankur Mehta, Kevin Weekly, Ben Cook, Lance Doherty, Mike Scott, Subbu Venkatraman, and Brett Warneke. I've had insightful, productive, educational conversations with all of you over the years and, sometimes, it feels like my dissertation is mostly the sum of those conversations. At the beginning of my first foray into graduate school I was warned that graduate work is lonely and one shouldn't undertake it unless one is prepared for and comfortable with that loneliness. Lab work involves a lot of solo time, but the overall results are absolutely a product of the network in which one is immersed.

The people, in addition to the grad students above, who are part of SCuM (or SCM or scum or $SC\mu M$...) and OpenWSN and, in doing so, make this project into a larger exciting whole: Francesco Bigazzi, Keoma Brun, Tengfei Chang, Timo Claeys, Osama Khan, Paul Kwon, Borja Martinez, Sahar Mesri, Lorenz Schmid, Ioana Suciu, Arvind Sundararajan, Pere Tuset, Xavi Vilajosana, Mališa Vučinić, Thomas Watteyne, Ziran Zhang, and Bob Zhou.

A couple of students downtown were particularly instrumental in helping me figure out the practical parts of my work: Stevo Bailey, James Dunn, Ozzy Lacaille, Nathan Narevsky, Sameet Ramakrishnan, and Andrew Townley. The software tools in this field are arcane at best and utterly unusable at worst and these folks generously shared their secrets.

My Berkeley postdoc collaborators in other groups/departments: Dr. Ben Smarr, Dr. Hossain Fahad. Ben gave us an armband application that got this work started, and Fahad worked with us to build a chemical sensor demonstration that capped it all off.

Our Boser group neighbors to the west: Behnam Behroozpour, Burak Eminoglu, Igor Izyumin, Oleg Izyumin, Mitch Kline, Pramod Murali, Efthymios Papageorgiou, Richie Przybyla, Hao-Yen Tang, and Yu-Ching Yeh. Somehow we were able to share lab space, equipment, and ideas in total harmony. I don't know how a group of people so respectful of each other's space and experiments gets assembled but I'm glad it did. Plus, everyone in that group is sharp as a tack. I learned a ton from them and I expect to keep learning things from them for the rest of my career.

The rest of the folks with whom we shared the Swarm lab and the folks who kept it running: Mila MacBain, Candy Corpus, Erin Hancock, Mark Oehlberg, Alex Palomino, Ken Lutz, Shelley Kim, Huo Chao Kuan, and students from the Arias, Kubiatowicz, and Maharbiz groups and others. Landlords and housemates usually made home life miserable, but positive lab life with these folks made up for it.

My friends at Novo Nordisk over the last couple years: Ben Vu, Brittany Berry-Pusey, and Steve Barnaby, whose input and interest gave my work a context I feel lucky to have had.

University of Washington professors and staff who stayed interested in my Berkeley progress: Linda Bushnell, Jim Peckol, John Sahr, and Pam Eisenheim. It was always great to see you at Bay Area UW alumni events or when I'd visit Seattle and your genuine interest in my progress made me proud of my recent accomplishments even if they felt lackluster to me.

The folks who comprise the rest of my Berkeley circle, giving me opportunities to stay sane and well-socialized when the alternative was just around the corner − the meangirls and associates, in particular: Joe Corea, Stefanie Garcia, Brittany Judoprasetijo, Ben Keller, Dom Labanowski, Claire Lochner, Jodi Loo, Joe Menke, Varun Mishra, Aminy Ostfeld, Maggie Payne, Adrien Pierre, Thomas Rembert, Mahsa Sadeghi, Phil Sandborn, David Schinazi, Simon Scott, Sam Smith, Evan and Katie Sparks, Rachel Traylor, Donna Weekly, Natasha Yamamoto, Tom Zajdel, and Alla Zamarayeva.

Everyone who participated in spaceships games or my east bay bike trips, with apologies to those who clamored for more while I toiled away at my research instead.

The folks who keep BSAC running: Mike Cable, Dalene Schwartz Corey, John Huggins, Mark Kasulen, Richard Lossing, and Kim Ly. I came up reading BSAC papers in college and it's been surreal to be part of the center in grad school.

EECS department folks, the stewards of How To Actually Get The Degree, which whom I interacted with in bursts whenever I was getting close to the next milestone: Shirley Salanio, Susanne Kauer, and Patrick Hernan. It's always felt like this department had things figured out and was working for students' best interests: a big relief when day-to-day things aren't going well and you need some reassurance that the big picture is going fine.

Tom Parsons and everyone we never interact with at ERSO: we probably caused you a mountain of administrative troubles over the years and you have my appreciation for making it all work out behind the scenes.

EECS instructional computing, EECS research computing, and BWRC computing folks on whom I leaned to get network/compute resources running and who sometimes needed to gently push back: Bira Coelho, Eric Fraser, Mark Kraitchman, Brian Richards, and Lars Rohrbach. I remember listening to the tech infrastructure hour on Orientation Day and coming away with the impression that IT in this department really had it together – and the last six years have confirmed it.

The NDSEG Fellowship administrators at ASEE: they were extremely clear and communicative over the duration of my tenure so I was never worried about my next paycheck. This is a big deal when you're living on a grad student stipend in the Bay Area.

My friends from the old days who have seen me less and less since I entered this program but have stayed friends all the same: Reid and Abby, Maze, Nels, Tony and Sara, Khahil, Mara and Indi, Colin, AJ, Sarah, Nate and Emily, Alisha and Andrew, Jeremy, and the SIGGRAPH GraphicsNet team.

My academic mentor from the outside world, Dr. Pauline Yu, for sanity checks, perspective, and really weird food over through the years.

And Kris, of course, for taking a chance on me in the first place and painstakingly teaching me what grad school actually is. I feel like a completely different person compared to when I started, so here's hoping all the frustration was worth it.

Chapter 1

Introduction

This dissertation pertains to research in pursuit of ultra-miniaturization of wireless systems by eliminating all off-chip components and all on-chip inductors, leading to a wireless system limited only by transistor scaling. Current complete wireless systems, usually referred to as "modules," have been stuck at approximately 1cm x 1cm in both industry [1, 2] and academia [3, 4]. This is because they require off-chip components in addition to the wireless IC, all assembled on a PCB. Elimination of the PCB requires elimination of three things current ICs requires: antenna, power, and a crystal oscillator frequency reference.

The Single-Chip Mote project encompasses more than this dissertation and includes at least four other dissertations/theses [5, 6, 7, 8] and a dozen papers and talks [9, 10, 11, 12, 13, 14, 15, 16]. We worked primarily to do what RF designers would say is impossible and eliminate the crystal. Supplying power to low-power platforms is already a popular research area, so our power efforts here were mostly limited to low-power design across the chip with an eye toward pushing peak power − and total energy consumption − below the threshold of something a chip-scale scavenged energy source can provide (it does us no good to design a fully-integrated wireless module only to power it from an energy scavenger 100x larger, for instance). As for the antenna: electrically-small antennas (much smaller than the wavelength) simply perform poorly [17, 18, 19]. Unless we can change the speed of light, as in low-temperature co-fired ceramic (LTCC) found in chip antennas, we're up against fundamental laws and have made no further attempts to miniaturize good antennas at 2.4GHz, where $\lambda/4 = 3.125cm$. Off-chip antenna elimination is probably best done at higher frequencies with shorter wavelengths.

Supposing antenna, power, and crystal are all eliminated, the size of the resultant wireless module is limited only by how tightly we can pack transistors and on-chip passives. Most of the single-chip mote's size can shrink as transistors do, but the stand-out element immune to scaling is the inductor (other passives don't scale well either, but the inductor is by far the largest and least likely for any scaling breakthroughs in the near future).

At 15GHz, $\lambda/4 = 5mm$ and we enter the domain of possible on-chip antennas. It may be that, in that domain, LC tanks can proportionally shrink so eliminating the inductors isn't as critical. But at today's most popular communication frequency, 2.4GHz, inductors remain

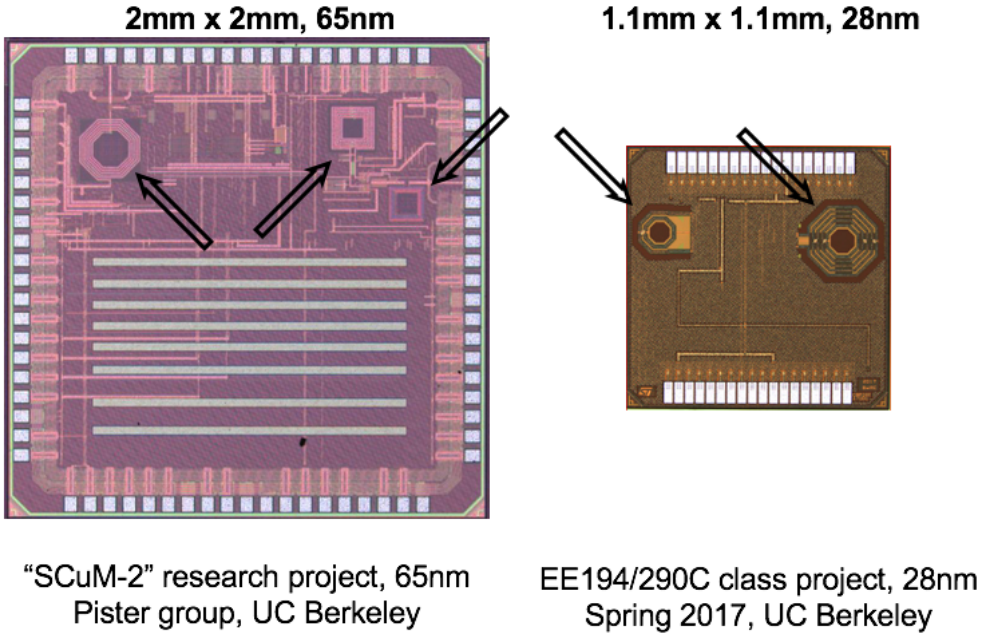


Figure 1.1: Low-power wireless ICs with planar inductors indicated by arrows. ICs are pictured to relative scale. Inductors range in value from 0.8nH to 9nH. Left: Second-generation single-chip mote IC in TSMC 65nm bulk process, which included the first microprocessor-based digital system [7]. Right: IC in STMicro 28nm FD-SOI process, designed and fabricated as part of our new course [16].

an impediment.

For example, we attempted a complete wireless module including a 2.4GHz transceiver based on an LC tank local oscillator (LO), microprocessor, RAM, and everything else in a 65nm process. That first attempt is on the left side of Figure 1.1. On that chip, the inductors are maybe 10% of the chip's active area (neglecting pads and sealring). A second attempt in 28nm is pictured on the right of Figure 1.1 (designed as part of a new Berkeley course [16]). There, the inductors are around 25%. At 14nm the inductors would probably be the majority.

Our architecture typically includes two inductors: one in the match, and one in the oscillator. The single-chip mote has always used a passive tapped-capacitor matching network for passive gain at the frontend inspired by the work of our predecessors [20], which includes an inductor. That can be replaced with a low-noise amplifier (LNA) to get the job done [21]. But the typical RF oscillator, an LC tank, remains. The alternative to the LC tank is the ring oscillator – vastly smaller, inductor-free, and potentially lower power (e.g., a 2.4GHz ring oscillator in 65nm can draw as low as ~100μA vs LC tank's 1mA or more), but exhibiting 1000x or greater phase noise for a given power as described at the end of [22]. Phase noise of all RF oscillators is usually reined in by a frequency synthesizer including a phase- or frequency-locked loop (PLL/FLL) but, without a crystal oscillator, we have no reference on

which to base the loop.

Hence we investigate wireless systems incorporating a noisy, but small and low power, RF oscillator. The result could be a complete 2.4GHz wireless module less than $300\mu m$ x $300\mu m$ needing only power and antenna and of benefit to applications able to leverage small and low mass form factors, reduced power, and lower cost (in silicon area and in the complete module's bill-of-materials). These can include medical devices to be implanted of swallowed, disposable products, chemical processing integration (e.g., stirring in these chips at the factory to create smart paint), unobtrusive telemetry packages to instrument flying insects, or tiny spacecraft where every gram counts against fuel costs.

This dissertation is the result of that investigation and consists of the following components:

- We first explore a sample application that could benefit from such miniaturization in Chapter 2. The reported device is realized with COTS parts but designed with single-chip applications in mind.
- Next we discuss the performance of our first free-running ring oscillator in the context of an FSK receiver in Chapter 3.
- Background about phase noise, jitter, Allan Variance/Deviation, etc. is described in Chapter 4 to aid in explaining observed phenomena.
- Performance of various relaxation oscillators is in Chapter 5. This performance is evaluated in the context of two low-power RF communication standards.
- An ASIC containing a re-designed ring oscillator-based transceiver, microprocessor, sensor ADC, and other accessories is tested with respect to communication (Chapter 6) and chemical sensor system (Chapter 7) capabilities.
- Finally, Chapter 8 includes results from our work to create a new course at Berkeley to help students get up to speed on chip design faster to enable more efficient research in this area.

Portions of the aforementioned ASIC is documented in Appendix A with pointers to additional sources of documentation.

Updates to this dissertation can be found at `https://people.eecs.berkeley.edu/~db/dissertation/`. That URL should serve as a pointer-for-life and redirect to wherever my web host is in the future. At time of writing it points to space hosted by UC Berkeley's excellent Open Computing Facility: `https://www.ocf.berkeley.edu/~dcb/dissertation/`.

Chapter 2

Reconfigurable, Wearable Sensors to Enable Long-Duration Circadian Biomedical Studies

In this chapter, the reported device is realized with COTS parts but designed with single-chip applications in mind. This chapter is based on our BODYNETS 2014 paper [23] which was later expanded to a journal article [24]. Since writing the paper, the device has continued to be used by the Kriegsfeld lab and our two labs continue to discuss future joint projects, particularly wireless telemetry of core body temperature implantable in rats and mice in a form factor small enough that the animal behaviors are minimally affected and data is available immediately instead of having to retrieve a data logger posthumously. These projects are discussed with an eye towards a platform able to similarly instrument large populations of humans for many years, thereby enabling physiological research impossible to undertake today.

2.1 Abstract

The last 10 years have seen the emergence of wearable personal health tracking devices as a mainstream industry; however, they remain limited by battery lifetime, specific sensor selection, and a market motivated by a focus on short-term fitness metrics (e.g., steps/day). This hampers the development of a potentially much broader application area based on optimization around biomedical theory for long-term diagnostic discovery. As new biometric sensors come online, the ideal platform enabling the gathering of long-term diagnostic data would have the built-in extensibility to allow testing of different sensor combinations in different research settings to discover what kinds of data can be most useful for specific biomedical applications. Here we present the first generation of a reconfigurable wrist-mounted sensor device measuring 7x4x2cm and weighing 51g with battery (29g without). In its current configuration, it has recorded skin temperature, acceleration, and light exposure; these three

variables allow prediction of internal circadian rhythms, as an example of the application of biological theory to enhance pattern detection. This generation is capable of operating long-term with minimal day-to-day disruption via easily exchangeable batteries, and has enough space for several months of data sampling to gather long-term diagnostic metrics. Future developments will include the addition of energy scavenging and a wireless mesh network for ambient data collection, the combination of which will allow uninterrupted data to be gathered without depending on the user.

2.2 Introduction

In the next several decades, medicine will mature from primarily intervention-focused to predictive and preventative. To accomplish this transformation, large amounts of potentially diagnostic biomedical data must be gathered across broad populations to enable the development of the predictive algorithms that will power early detection and disease avoidance in medical practice. Wearable devices could play a large role in generating these data, but to do so they must overcome hardware obstacles such as biosensor fidelity, battery life, and data accessibility. However, given the complexity of human physiological interactions with the modern environment and the role of these interactions as drivers for disease, greater difficulty may lie in discovering which data types provide the best long-term diagnostics for each medical condition in each different population. Outside of a biological framework, finding such patterns is reduced to a brute force approach. Therefore device development should be coupled with biological hypotheses to maximize efficiency of pattern discovery and validation, e.g., [25, 26, 27].

A central source of variance in biological systems is the presence of biological circadian rhythms. Circadian regulation is widespread in the human genome [28, 29], and diverse tissue systems show populations of cells with internal circadian clocks (see [30] for a review). As such, day-to-day shifts in circadian rhythms – such as those caused by artificial light at night, shift work, and jet lag – can disrupt the relative alignment of internal timing systems [31, 32, 33, 34, 35]. Such "internal desynchrony" interferes with biological processes and, when chronic, it increases the risk of many diseases, including heart attacks, cancer, obesity, depression, and reproductive dysfunction [36, 37, 38, 39, 40, 41]. In turn, disease states often show circadian symptoms, e.g., hormone concentrations might show changed waveforms; these fluctuations might not be apparent through daily one-time measurements but could be diagnostic when seen as changing daily waveforms, as has been shown for instance in adrenal hormones in circadian disrupted rats [42]. As such, using circadian biology as a starting framework for wearable device design has two advantages: in the short term, changes to daily waveforms of biological measures may assist assessment of current health state; in the long term, persistent measures of circadian stability will enhance the short-term projections and allow for the construction of personal predictive models of daily waveforms and rates of change thereto across time scales (e.g., changes in daily temperature rhythm across the female menstrual cycle, changes to blood sugar waveforms across season, etc.). By framing multivariate, high

volume wearable device-generated data in biological time, the temporal resolution of data can be increased while the unexplained variance is reduced, and the probability of successful pattern detection for diagnostic purposes is increased substantially.

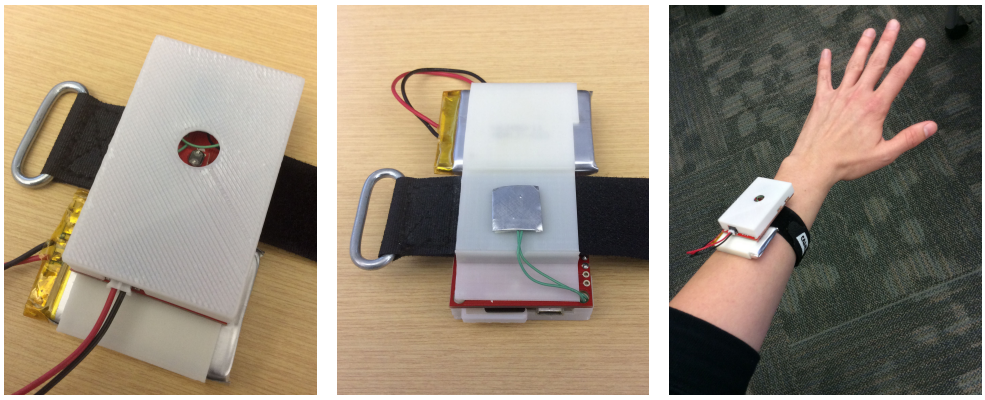
While biometric devices are in a state of rapid change, the ideal platform for interested innovators would be flexible so that new technological advances could be integrated to allow efficient evolution of biomedical device use. With such a platform, engineers could partner with biomedical researchers to efficiently integrate their specific advances into application hypothesis testing. This would speed turn-around of short-term payoffs such as instantaneous disease detection while sustaining the longer-term development of predictive diagnostic algorithms. For this platform to be capable of taking advantage of the predictable changes brought by circadian rhythms, it must be able to maintain high fidelity sensor input across several days.

In this manuscript we present a wristband device capable of continuous input from multiple sensors across more than 24 hours, with an easily exchangeable battery so that use is interrupted for only one minute/day. Internal circadian phase of core physiological functions can be predicted accurately with a combination of distal body temperature, activity, and light exposure [43, 44]. Therefore the platform presented here is equipped with a skin thermometer, accelerometers, and a light sensor, and gives output of these sensors across several days during real-world trials. This makes it viable to test biomedical hypotheses based on circadian waveform analysis. The platform is also designed to be flexible, allowing modular sensor configurations and user-reconfigurable recording parameters for modified tests as new sensor technology and biological hypotheses become available. Finally, this platform is being optimized to allow continuous, ambient data collection so that users can generate data capable of near real-time analysis without added user burden. It is our hope that the development of this and similar devices will speed the emergence of cheaper and more effective personal, predictive medicine through wearable device use.

2.3 Device Description

Hardware

The device was designed to be highly reconfigurable while taking the shortest path to collecting data relevant to circadian biology. The core of the device is the NXP LPC2148 ARM7-based Sparkfun product “Logomatic V2” PCB. Starting with an off-the-shelf product kept initial costs and build time low and allowed quick replacement when damaged, as wearables often are. The board was also well-supported by the open-source hardware community which further reduced engineering time, though this work represents a significantly more advanced use of the board than ever previously attempted. The board was enhanced to include sensors to measure the three aforementioned analytes: distal body temperature, activity, and light exposure. Sensors and their interface circuits were chosen to be as simple as possible to enable an extremely low-cost circadian logger.



(a) Top (b) Bottom (c) Worn

Figure 2.1: Photos of current device showing light port 2.1a, skin temperature plate 2.1b, and device in use 2.1c.

The PCB and sensors were combined with a 3D-printed enclosure, fastening strap, 2GB SD card, and 3.3V 1000mAh battery to form the complete device capable of logging several months of data (Figure 2.1a). Its dimensions are approximately 7x4x2cm and it weighs 51g with battery (29g without), making it appropriate for comfortable wrist wear (Figure 2.1c).

Temperature

A thermistor, TDK NTCG163JF103F, was attached to a thin 1.5x1.5cm aluminum plate with thermal epoxy and affixed with foam tape to the inside of the band. This was so the aluminum plate would make skin contact when worn (Figure 2.1b). This provided maximum thermal conductance to the thermistor from the wearer's skin while minimizing thermal sink to the rest of the device and ambient environment. The thermistor was wired as part of a resistive divider to one of the ADC ports on the PCB. The other resistor in the divider was chosen such that the thermistor's 0.1°C resolution corresponded to the voltage of one LSB of the ADC over expected skin temperature range. This allowed power and part savings by eliminating the need for an amplifier. In the next iteration the sensor will be power gated by the microcontroller, allowing further power savings.

Activity

Activity was recorded by the MMA8452 triaxial digital accelerometer. Though all three axes were logged at the beginning, it was determined that any one of the three were sufficient to infer activity level. The axis horizontally perpendicular to the arm was chosen for analysis. Acceleration along other axes were ignored. Communication to the microcontroller took place over I2C when awake to measure instantaneous acceleration, or via interrupt signal during sleep to count all instances the device experienced acceleration beyond 0.1 g-force.

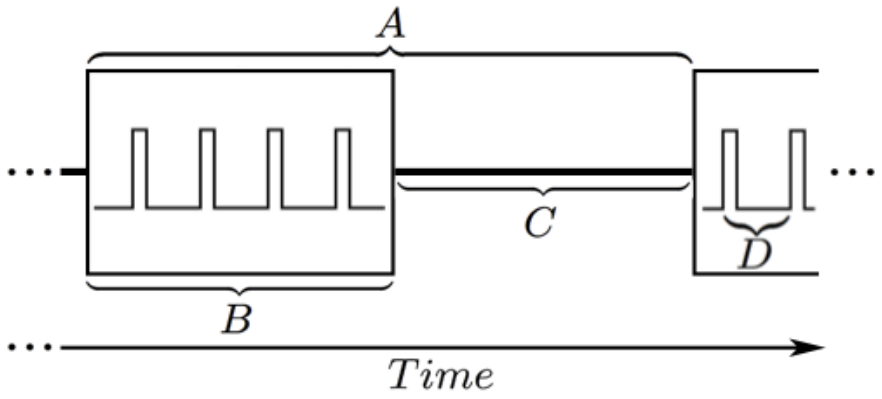


Figure 2.2: Data collection timing diagram. Each pulse represents one set of sensor data sampled and logged. Time A is the time between the start of each sampling block. Time B is the duration of each sampling block. Time C is the time spent outside of a sampling block, when the device is in sleep mode to conserve power. Time D is the time between samples. Times A, B, and D are all user-defined via an interactive Python script.

Light Exposure

The Clairex CL9P5L photoconductor was used to determine light exposure. It was attached to the surface of the device on the hypothesis that visible light reaching the wrist was a good approximation of the blue light entering the wearer's eye, stimulating the optic nerves that contribute to circadian rhythm maintenance. The data from this sensor needed only be qualitative to be useful, so the logarithmic response of the photoconductor didn't need to be linearized. Another resistive divider was built to roughly divide the range of the ADC into five brightness regions, which can be subjectively described as dark, dim, indoor night, indoor day, and full daylight. Like temperature, this sensor will also be power gated in the next iteration.

Firmware

The firmware included with the ARM7 board was augmented to include I2C support, sleep mode, interrupts, custom data logging formats, and adjustable sampling periods. Using a config file on the SD card, users can set sampling frequency, duration of sampling set, and length of sleep before another sample set (Figure 2.2). While sampling, ADC values for temperature and light are taken and the accelerometer is queried. While sleeping, temperature and light are ignored but any activity that generates an acceleration greater than 0.1 g-force sends and interrupt to the microcontroller to increment a counter.

Software

Configuration and processing code was written to ease use of the device by biomedical researchers. The firmware config file is generated by answering questions posed by an interactive Python script, adjusting the variables described in Figure 2.2 to suit the needs of the investigation. The resultant data files are also pre-processed by a Python script and then analyzed with Matlab code written to correctly interpret recorded data into useful results.

2.4 Data

Initially data from each sensor was taken individually and used to calibrate or correct artifacts arising from the sensors themselves or their integration in the central board. All three sensors saw revision of data processing and output.

Temperature

The temperature sensor was designed to provide accuracy greater than clinical thermometers in use today but was initially found to undergo significant drift. A more stable voltage reference was built around the sensor to improve stability, only to find the drift's direction, but not magnitude, was changed. Despite this small but detectable drift, the relative temperature values during each sampling block clearly still capture diurnal rhythms. The overwhelming of the observed drift by the subject's daily temperature rhythm indicates this sensor is still applicable to circadian studies.

Activity

The accelerometer installed in this device is capable of capturing both greater temporal resolution and greater dimension depth than was deemed necessary for the application of the device to discerning circadian phase. Circadian studies traditionally rely on gross measures of locomotor activity, such as 10 minute bins of acceleration counts across the day. Initially the device revealed foot steps and hand motions, but the accelerometer's output was pared down to mere 1-dimensional counts. This generated savings of storage space and power while not impacting the biological application. This choice is an example of the kinds of optimization possible when device development is coupled to a specific biological framework.

Light Exposure

Early tests with the light sensor confirmed our ability to see consistent logarithmic changes in resistance with light intensity as expected. Repeated measures outside under noon-time sun, inside under fluorescent laboratory lighting with and without daylight nearby, inside under low light, and concealed in a dark box allowed for the designation of consistent resistance ranges to different common lighting conditions. Although quality (e.g., wavelength decomposition)

of light is not a feature of this device, the greater intensity of daylight than office lighting, and regressions against known sunrise-sunset times allow confident disambiguation of most light sources (i.e., the light intensity at all but sunrise and sunset may be enough to infer the source of light as artificial or natural in most circumstances; the presence of smooth transition curves under natural light at sunrise and sunset further help differentiate the source of light at those times, when intensity alone is insufficient).

Combined Measures in the Real World

Once all three sensors had been calibrated and artifacts corrected, a series of field tests was run. A representative sample of data taken by the device is shown in Figure 2.3. In this example, the device was operated in the real world for 54 hours continuously, stopping only to briefly exchange batteries. It was configured such that all three sensors were sampled 10 times/second for 15 minutes, the system went into deep sleep for 45 minutes, then woke up to perform another 15 minute, 10Hz cycle. The sampling configuration was the result of prior tests run by the authors. These data span multiple sleep and wake cycles and persist through many kinds of physical daily activity. For this recording session subject was at home with exposure to natural and artificial light. The data show clear diurnal trends consistent across sensors, indicating that the device is likely suitable for circadian biology studies. Note for example the smooth rise in light intensity at times of sunrise, and the smooth decline at times of sunset, which transition into a plateau of artificial light into the evening on both nights, with decreased activity, validated by the self-reported sedentary evening actions of the subject (e.g., “watching a movie"). Despite a known error of $\sim$1°C thermistor drift over each 15 minute period – the source of which is under investigation – the diurnal cycle of temperature is still visually apparent, so circadian studies may not be affected. Future analysis will enable further refinement of the quantity of data taken, extending device capability and lifetime.

2.5 Future Work

For the entire history of life, availability of sunlight has shaped the evolution of behavioral “temporal niches" (e.g., diurnality in humans) and been the most reliable marker of seasonal change. Therefore, biological visual systems evolved to enable both image formation (i.e., seeing) and correctly timed daily and seasonal biological changes (e.g., increasing fat mass in shortening days). The integration of electric lights into modern life therefore disrupt biological timekeeping mechanisms. For this reason, the current urban environment presents a number of challenges to human health, such as shift work or light exposure at night, discussed in the introduction. Over a lifetime, such disruptions increase the likelihood of disease in every organ system, from cancer to arthritis to depression. However, the manifestation of circadian rhythms and the impact of artificial circadian disruptions are different in different people. Though the reality of negative health impacts from circadian disruption is now clear, studies in modern populations to identify best practices for different demographics (or individuals)

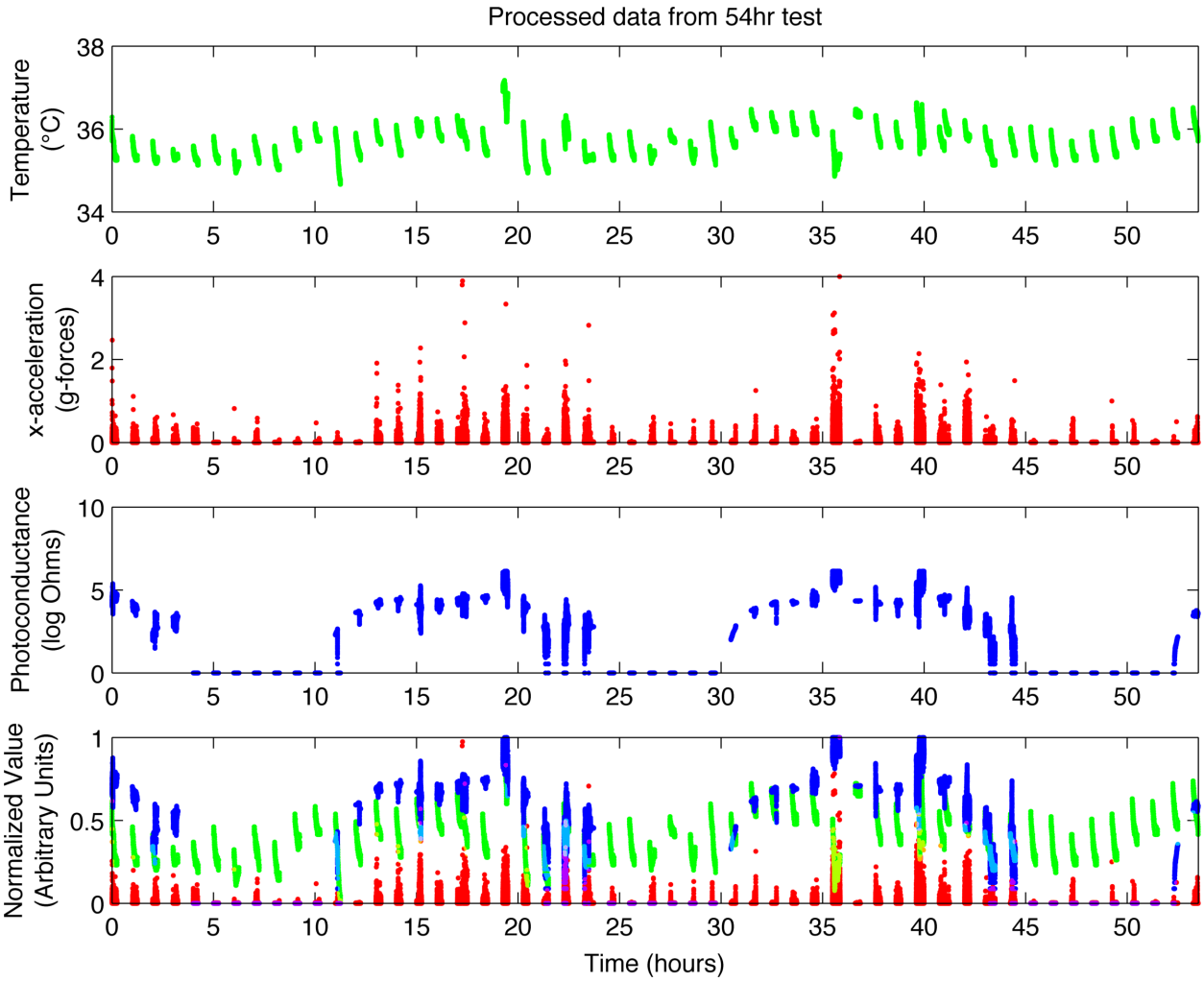


Figure 2.3: 54 hours of continuous recording shows predictable changes in temperature, activity, and light across three nights and two days. A normalized, combined overlay is also provided. Note that time to change batteries is not apparent in the data.

are blocked by a lack of accessible data. Wearables provide a non-invasive mechanism of gathering such data, but are most useful only if researchers can configure the kinds of data acquired based on knowledge of biological systems. Thus the wrist band we present here is useful for those interested in studying circadian disruptions in people living freely in their environments. Our device requires only one minute per day to change the battery and upload data. This means that as a medical device it is unobtrusive, and could realistically be used to generate accurate weekly logs of individuals' activity cycles in the context of light exposure, precise in both time and intensity.

For broader use (years of recording and/or accessibility to young children or the technically disinclined) user burden would ideally be negligible. Therefore the device presented here is also the starting point for a second generation device able to gather circadian data over long periods without user input or maintenance. Specifically, the future version of the wearable presented here is currently being developed as a fully-integrated wireless sensor system in the spirit of the Smart Dust mote concept [45]. This device will be small enough to be worn unobtrusively and use thermal gradient, kinetic, and solar scavengers to obtain power and perform sensing relevant to circadian biology. It will also take advantage of a data network utilizing existing work in wireless body-area networks (WBANs) [46, 47], and mesh networks [48]. A conceptual network layout is given in Figure 2.4. These features would allow continuous monitoring without depending on the user to remember to recharge or protect the device, or to sync data. As such, this next generation will expand the current functionality presented here to enable continuous data gathering from each user. This will maximize the potential to identify patterns and provide diagnostics for long term ailments arising from lifetimes of exposure to the modern, disrupted circadian environment.

2.6 Conclusions

A reconfigurable wearable device founded in circadian biology has been presented. It is capable of logging skin temperature, activity level, and ambient illumination for an extended period. Its flexible nature allows the user to change sensors and adjust sampling behavior to more accurately tune the device's output for a given study. Results from our experience and biomedical researcher feedback continue to inform device refinements; we aim to eventually have an integrated yet configurable platform optimally targeted toward early detection and prevention of disease states in ways not currently feasible with wearable devices.

The goal of this work was to motivate the development of chip-scale sensing systems to measure the same environmental parameters in a form factor that can be chronically worn with zero user burden (bulky size, recharging, data transfer, cost, etc.). A block diagram for such a system is given in Figure 2.5, consisting of only two mm-scale chips and antenna and capable of measuring temperature and ambient light. An accelerometer for actigraphy would need to come from elsewhere, e.g., an attached MEMS plane. Fortunately, temperature sensing is the most critical component to health outcome prediction and is already built in to the SCM IC as illustrated.

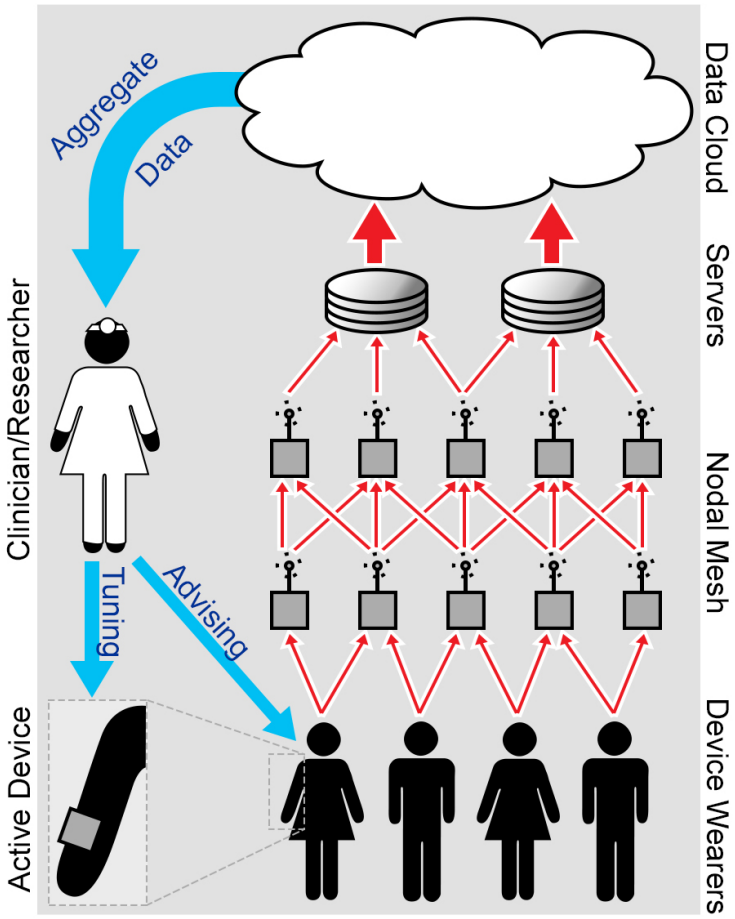


Figure 2.4: Conceptualized ambient network to collect and process wearable health monitor data.

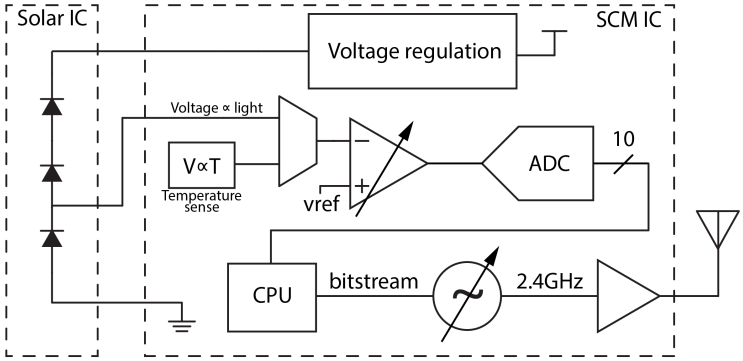


Figure 2.5: Conceptual miniaturized "armband"-style system based on 2mm x 3mm single-chip mote (SCM) platform. Current iteration of SCM includes temperature sensor and external input to PGA/ADC frontend. A 3mm x 3mm solar power IC (current iteration called "Zappy2") was developed to power SCM from a nominally 1.2V supply created from three diodes in series. The voltage across one diode could be connected to the external ADC input to get a relative indication of ambient light.

Chapter 3

Narrowband communication with free-running 2.4GHz ring oscillators

This chapter discusses performance of our first free-running ring oscillator in the context of an FSK receiver. Comparisons to similar work are particularly difficult because very few researchers use free-running ring oscillators for communication. This chapter is based on my PEMWN 2017 paper [14] and, since writing it, we understand more about accurately quantifying oscillator performance (e.g., via Allan Variance) to make more fair "apples-to-apples" comparisons. This means the apparent agreement between simulation and experiment displayed in Figure 3.2 may not be generalizable.

This work was performed at high SNR to examine the effects of high phase noise alone. It also strove to use as narrow bandwidth as possible. This was to (a) differentiate the work from other papers with 100MHz or greater receive bandwidth, and (b) attempt to work with 1MHz tone spacing and therefore enable the possibility of receiving from a commercial IEEE 802.15.4 transmitter. The results were promising but a complete test with a COTS transmitter has yet to be attempted. We later found sensitivity is poor at narrow bandwidths, so later work in Chapter 6 drops the 802.15.4 receive goal and trades tone spacing/bandwidth for substantial bit/chip error rate benefits to maximize sensitivity of ring-to-ring communication.

3.1 Abstract

Ring oscillators have area and power advantages over LC tanks, but conventional wisdom is that rings must be locked to a high-Q external reference to be useful in RF communications. In this paper we explore performance of a 2.4GHz receiver incorporating only a free-running ring as a local oscillator. Using a simple technique to compensate for frequency error, we find that a minimum-size ring fabricated in 65nm CMOS and consuming only 105μW is able to demodulate 75% of received 802.15.4 packets and, if the FSK tone deviation is doubled from 802.15.4 spec, packet receive rate exceeds 99.8%.

3.2 Introduction

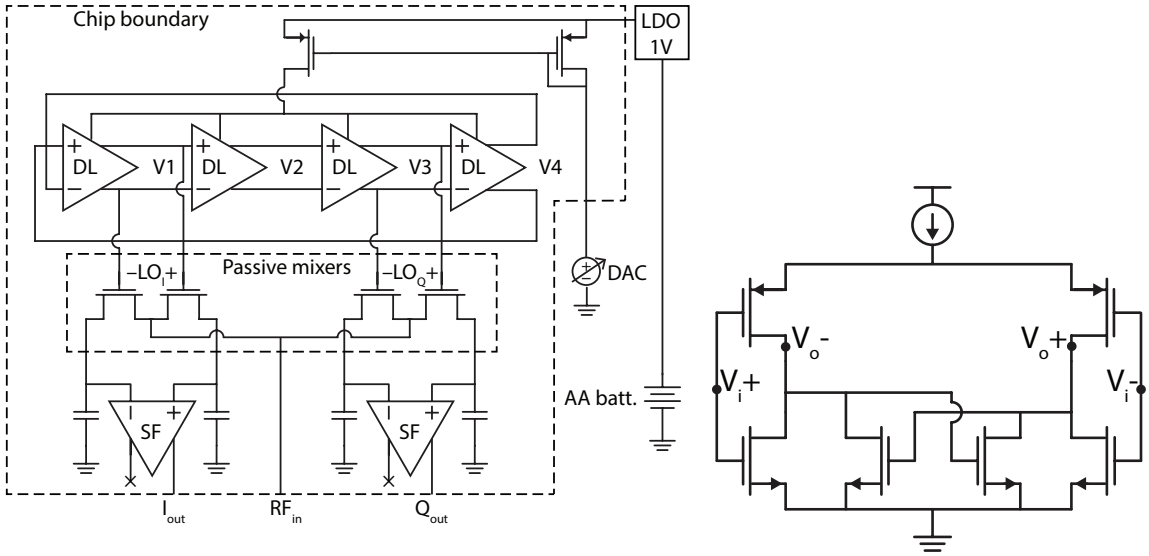
As CMOS communication circuits move from being manufactured as standalone components to being integrated alongside ASICs to add wireless communication functionality, die area and power comes at an increasing premium. Digital components benefit in area and cost from process scaling while analog and RF seldom do. This is owed partially to the large size of the passives analog and RF blocks must utilize. Often, RF designs will save area by using a ring oscillator to generate local RF signals in lieu of a large inductor as part of an LC tank. Removing the filtering capabilities of the resonant tank comes at a significant phase noise cost. Standard wisdom is that a ring oscillator's frequency error is so significant that communication is impossible unless the ring is locked to a high-quality reference via PLL, which requires off-chip references and greatly increases system complexity. In this paper we explore capabilities of a 2.4GHz FSK receiver using neither a resonant tank to reject noise nor an off-chip reference to correct it. Instead, our receiver simply uses a free-running ring oscillator to generate the local RF signal. Furthermore, we investigate narrowband & high spectral efficiency communication performing 1-2Mbit (or Mchip, in our specific case) communication in a bandwidth of a few MHz. Our primary application is satisfying, to the maximum extent possible, the IEEE 802.15.4 specification [49] but these results can be applied to any modulation involving binary frequency shift keying (BFSK, sometimes also called 2FSK). This includes the 802.15.4 standard's offset-quadrature phase shift keying with half-sine shaping (OQPSK-HSS), which is equivalent to minimum shift keying (MSK) [50] as well as Bluetooth Low-Energy's Gaussian frequency shift keying (GFSK) [51].

CMOS circuit

Free-running ring oscillators for RF communications

Design of ring oscillators themselves is well-studied [52, 53] and some standards-compliant frequency synthesizer designs have been based on free-running rings [54]. In wireless sensor nodes, free-running rings or other relaxation oscillators are often used to satisfy timekeeping requirements [55, 56, 57, 10] while some ring designs have been specifically targeted for wireless communications [58, 59]. Radios incorporating free running LC tanks have been demonstrated to demodulate OOK [60, 61] and standards-compatible FSK [12] signals, but lack the compactness and scaling benefits of ring oscillators.

When rings serve as the RF oscillator and are left free-running, the radios that incorporate them tend to fall into two modulation categories. The first is OOK [62, 63, 64] incorporating strategies to compensate for an uncertain LO frequency and typically serves the role of always-on wakeup radio for a higher performance 2.4GHz transceiver. The second is FM-UWB [65, 66] and is usually a complete standalone FSK transceiver but with very wide modulation. Both types of radios have wide IF bandwidths, leading to low sensitivity which must be augmented by significant coding gain and therefore have low data rates. By contrast, our work aims to demonstrate FSK in a bandwidth of a few MHz which can enable higher sensitivities,



(a) Overall Circuit (b) Delay Cell

Figure 3.1: Experimental oscillator & NMOS passive mixer structure, as fabricated in a 65nm bulk process, diagrammed in 3.1a. Minimum-size ring oscillator containing 4 differential delay cells (DL) in 3.1b drives single-balanced passive mixers downconverting RF signal provided by test equipment. Mixed-down intermediate frequency passes into source followers (SF) with high drive strength to pass signal off-chip. Not shown: kick-start circuit to prevent even number of ring stages from latching at power-on (only possible if cross-coupled NMOS devices in (b) are overpowered by inverters and allow $V_o+ = V_i-$).

commercial standards compatibility and, by connecting this ring to a transmitter, symmetric communication.

3.3 Background

A simple experimental ASIC in 65nm bulk CMOS was designed and fabricated consisting only of mixer, output buffers, and the oscillator under test as illustrated in Figure 3.1a. No LNA and no baseband filtering or amplification was included. The oscillator is a 4-stage differential ring oscillator; topology was chosen with an eye toward use in a differential IQ receiver. Each of the ring's four delay cells contained a pair of inverters with cross-coupled NMOS, as drawn in Figure 3.1b, to provide additional gain and enforce opposite sides to have opposite phases. Kick-start devices, not illustrated, were present to start up oscillator in case it latches at a stable point at power-on. Latch state is only possible if the cross-coupled NMOS devices are overpowered by the inverters and do not enforce opposite sides to have opposite phases.

Transistors were sized at minimum (PMOS devices at 2x minimum size to account for differences in carrier mobility). Minimum size was chosen to give worst-case phase noise

behavior at best-case current consumption, with the knowledge that the measured phase noise results could be improved with increased width & power [67]. The ring drives the NMOS mixer gates with approximately 800mV swing at 1V VDD and 105μA input current. The ring can be tuned over a range of about 1.6GHz to 2.6GHz, limited by the current mirror devices. In the vicinity of 2.4GHz, frequency can be adjusted by 100kHz by changing the current mirror bias by 101nA (or 84μV). The resultant 2.4GHz signals were connected directly to passive mixer gates. This yielded what we believe to be the absolute minimum practical power consumption for a ring of this topology in 65nm CMOS.

Simulation

Analytical derivation of the phase noise profile is dependent on enough process-specific parameters that an accurate phase noise estimate for a given ring oscillator is impractical [68]. Hence, we use simulation tools to get a good idea of the ring's performance. A transient simulation in Analog FastSPICE of the complete circuit in Figure 3.1a was performed for 1ms of simulated time. The simulation included all foundry-modeled noise sources from 1Hz to 500GHz and calculated with "conservative" error preset accuracy. A histogram of this simulation is given in the left half of Figure 3.2. The presence of frequency-independent white noise and 1/f flicker noise will result in a phase noise spectrum in the shape of a Voigt profile [69], but for the purposes of using standard tools in this estimate we will approximate it as a Gaussian.

CDF prediction

To obtain an initial estimate of performance, we first assume the Gaussian profile represents the frequency distribution when the ring is tuned to a particular frequency. By shifting the mean of that Gaussian a particular distance (to a higher or lower frequency) and overlapping it with an un-shifted one, we can build an estimate of frequency distribution over time. These Gaussians were generated from statistics in Figure 3.2 and are displayed with frequency distribution from an example data set in Figure 3.3. These statistics were calculated over time periods of 1ms and 0.1ms, whereas each bit decision is made every 1μs (1Mcps) or 0.5μs (2Mcps), both of which will have smaller standard deviations. Hence they will yield somewhat pessimistic estimates, but they nonetheless allow us to obtain a reasonable starting point with which we can compare experimental results.

We assume the ring spends about half of its time tuned to either frequency, which is a good assumption given IEEE 802.15.4 DSSS coding. The cumulative distribution function (CDF) can then give us the probability of incorrectly selecting the wrong bit (i.e., the estimated chip error rate) based on the overlap of the two Gaussians. The 802.15.4 specification defines the sensitivity as the received power which yields a 1% packet error rate (PER) for a 20 byte (or "octet" in the parlance of the specification) packet with 6 byte overhead at a tone spacing of IF$\pm$0.5MHz or 1MHz total tone separation. A 1% PER can be shown as equivalent to 6.5% chip error rate (CER) [9]. Using this tone separation and setting the CDF decision point to

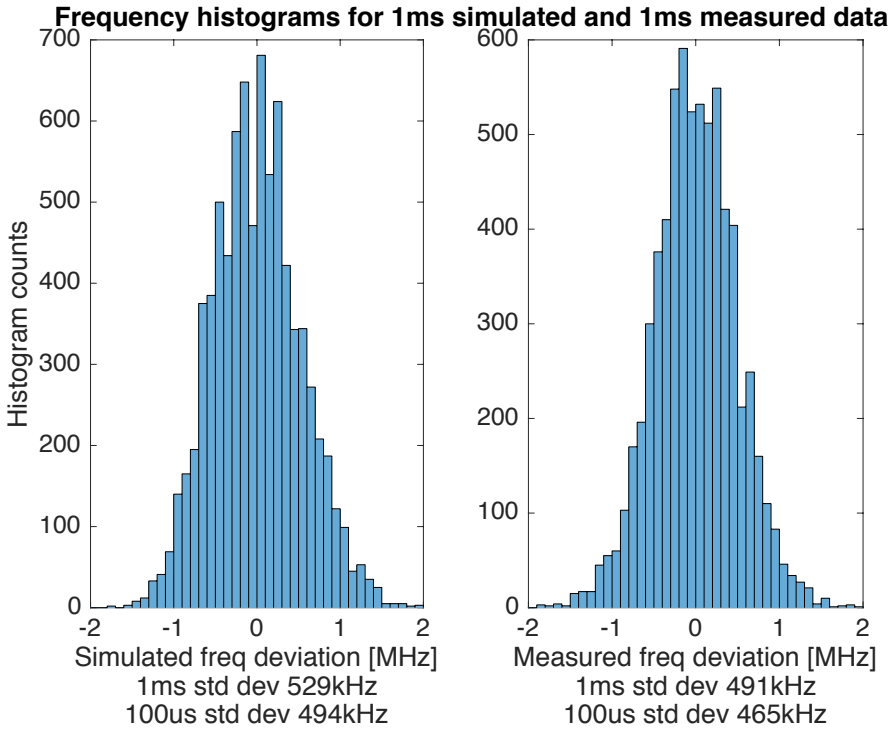


Figure 3.2: Frequency histograms for 1ms of simulated data (determined by computational limitations) and a 1ms portion of a 16ms oscilloscope capture. Both frequency distributions were obtained from nominally 8MHz intermediate frequencies resultant from an RF input derived from ideal source (simulation) or test equipment (measurement) mixed with a free-running ring oscillator. Standard deviations, or jitter, indicate that simulation and measurement are in good agreement. Standard deviation value for 1ms is calculated over full histogram period. Standard deviation value for 100μs is mean of jitter over 10 non-overlapping 100μs windows. Note: time-series of frequency data forming measured histogram on right is displayed in top half of Figure 3.9.

halfway between the two, our simulated standard deviation yields CER=17.2%, implying that no level of RF input power will yield a 1% PER. This is consistent with the data presented later in this paper. Solving for frequency separation instead, we find our simulated ring oscillator frequency distribution can meet 6.5% CER if FSK frequency shifts are increased to IF$\pm$0.80MHz or 1.60MHz total separation. Again, this is consistent with experimental data.

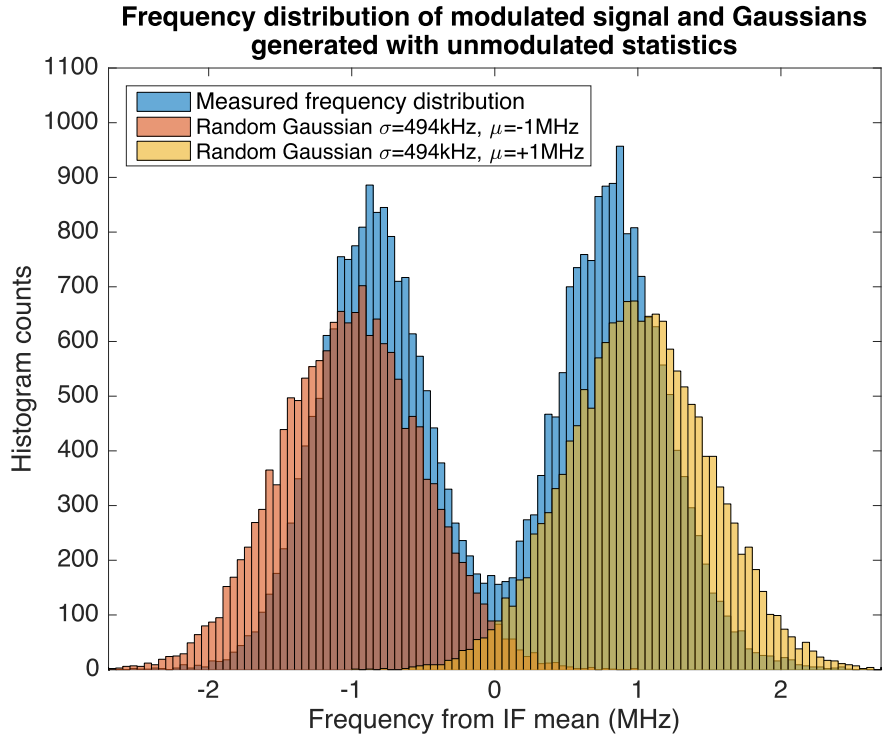


Figure 3.3: Distribution of measured frequency for 2Mchip/s and ±1MHz tone separation. Also plotted are two normally-distributed random values with measured standard deviation 494kHz, as displayed in Figure 3.2, and means of ±1MHz. Experimental data has been post-processed to subtract the windowed average (described in Section 3.5 and displayed in Figure 3.6). This can be thought of a histogram of frequency-distances-from-moving-average. The raw measured histogram looks like a single Gaussian distribution, but subtracting the (moving) average exposes the two groupings of high and low frequency resultant from FSK modulation. This subtraction is likely responsible for the reduction in standard deviation and frequency shift, which is about 375kHz and ±0.87MHz for each of the two modalities.

3.4 Experimental Methods

Setup

The RF input in Figure 3.1a was connected to a signal generator set to a frequency in the 2.4GHz band such that the intermediate frequency would be in the vicinity of 8-12MHz. Exact tuning of the IF was not possible due to the random frequency changes of the free-running oscillator. Neither signal generator nor ring oscillator frequency (i.e., ring oscillator current mirror bias) were adjusted over the several hours during which data was taken and the IC was kept in an open air lab environment. This mimics a one-time calibration of the LO. Data was collected during several of these multi-hour sessions over the course of several days. The

signal generator was set to produce enough input power to ensure the IF voltage waveform was approximately 25mV amplitude when captured. The desired intermediate frequency was determined through experimentation to be lowest possible while still ensuring a low likelihood of the ring's instantaneous frequency jumping across the incoming RF signal. A low IF was desired to maximize period measurement accuracy of the captured signal. The IF was captured by oscilloscope at 2.5GSamp/s for 16ms.

The ring's current source was biased off-chip. The frequency of a current-starved ring oscillator is controlled by its current, so any noise on the current being mirrored to the ring will directly result in additional phase noise. Initial attempts were made to generate current via SMU and dedicated current source IC but both were found to add an unsatisfactory amount of noise to the output. An appropriately-sized resistor to bias the current mirror was found to result in the narrowest practical ring frequency spectrum. Voltage was supplied by an IC LDO powered by two AAA batteries. Later a 16-bit precision DAC (Digilent PmodDA3 product using an Analog Devices AD5541A IC) powered by FPGA was found to produce practically equivalent frequency variation while being much more experimentally flexible.

Initial tests were performed in a shielded room but those results were not significantly different than an open lab environment.

Procedure

The IEEE 802.15.4 specification defines sensitivity in terms of 1% packet error rate for a packet containing a 20Byte "PHY service data unit" (PSDU) or data payload [49]. Packets were constructed per that spec, of size 26B total including 4B preamble, 1B start symbol, 1B length, and 20B of randomly-generated data. (We used random data in lieu of building a PSDU conforming to 802.15.4 MAC spec.) Data was transmitted at 1Mchip/s and 2Mchip/s over a variety of FSK tone spacings. Test hardware memory limitations dictated that 1Mchip/s transmissions be 14B in size: 2B preamble, 1B start symbol, 1B length, and 10B random data.

Using FSK with the minimum tone spacing, called Minimum Shift Keying (MSK), uses tone spacing half its data rate (e.g., 1MHz tone spacing – or ±0.5MHz carrier deviation – and 2Mcps data rate). It has a straightforward equivalent representation in the IQ domain [50, 70] whereas general FSK modulation does not. Weaver's 3rd Method [71, 72] was employed to generate equivalent IQ modulation for other non-MSK tone spacing for use with our test equipment. Said test equipment required carrier modulation inputs to be provided in the form of in-phase and quadrature signals.

These packets' 26B or 14B binary sequences were encoded using standard 4-bit-to-32-chip DSSS chip sequences from the IEEE spec (repeated in Table 3.1) for a total of 1664 26B*2*32) or 896 (14B*2*32) chips per packet, illustrated for the 26B case in Figure 3.4. These DSSS chip sequences were intended for Offset Quadrature Phase Shift Keying (O-QPSK) so the 1664- or 896-chip sequence was converted from IQ representation to FSK representation using the techniques described in [50]. Once in MSK representation, tone spacing could be

Table 3.1: IEEE 802.15.4 data-to-chip conversion for O-QPSK

Binary value $b_3...b_0$ or $b_7...b_4$	32-chip O-QPSK Chip Sequence c_0 ... c_{31}
4'b0000	11011001110000110101001000101110
4'b0001	11101101100111000011010100100010
4'b0010	00101110110110011100001101010010
4'b0011	00100010111011011001110000110101
4'b0100	01010010001011101101100111000011
4'b0101	00110101001000101110110110011100
4'b0110	11000011010100100010111011011001
4'b0111	10011100001101010010001011101101
4'b1000	10001100100101100000011101111011
4'b1001	10111000110010010110000001110111
4'b1010	01111011100011001001011000000111
4'b1011	01110111101110001100100101100000
4'b1100	00000111011110111000110010010110
4'b1101	01100000011101111011100011001001
4'b1110	10010110000001110111101110001100
4'b1111	11001001011000000111011110111000

increased (thus creating a general FSK signal) and Weaver's 3rd Method could be employed to generate arbitrary IQ signals for use with our test equipment.

The conversion procedure is as follows:

1. Assign every odd O-QPSK chip to I and every even O-QPSK chip to Q. (Ordinarily I or Q components change every other chip, alternating between an I change and a Q change, hence "Offset"-QPSK.)

2. Instantiate a new array of FSK chips of the same original length (1664 or 896 in this example)

3. For each FSK chip position (1-indexed), if the position is even, that FSK chip is **I** XOR **Q**. If the position is odd, that FSK chip is the opposite: NOT (**I** XOR **Q**).

Below is the same procedure as written in Matlab syntax:

```
I_OQPSK = packetOQPSK(1:2:end);
Q_OQPSK = packetOQPSK(2:2:end);

for ii=1:size(packetOQPSK,2)-1
     if(mod(ii,2)) % ii is odd
         packetFSK(ii) = xor(I_OQPSK(ceil((ii+1)/2)), ...
            Q_OQPSK(ceil((ii+0)/2)));
     elseif(~mod(ii,2)) % ii is even
         packetFSK(ii) = ~xor(I_OQPSK(ceil((ii+1)/2)), ...
            Q_OQPSK(ceil((ii+0)/2)));
```

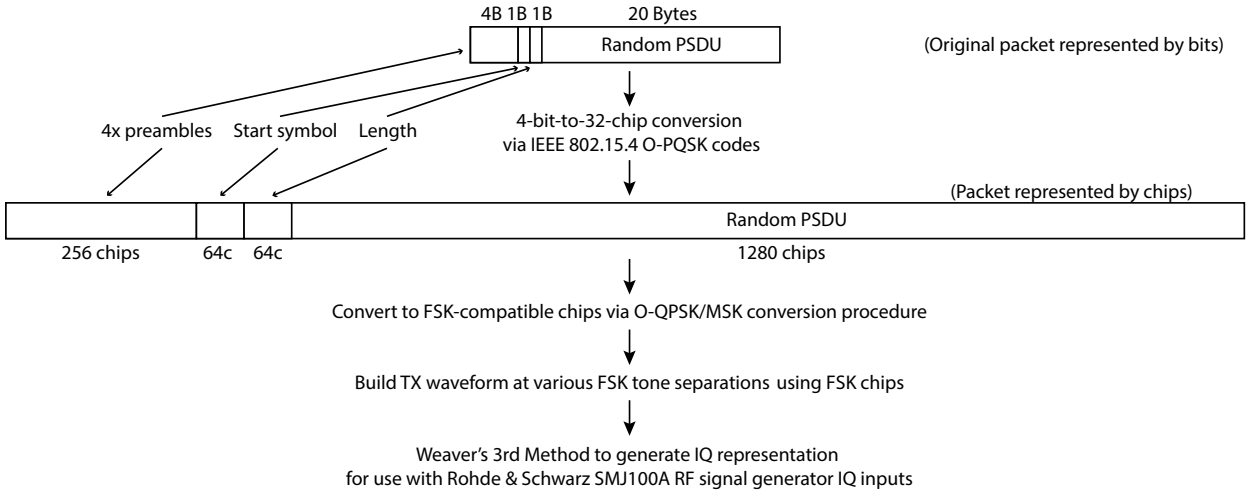


Figure 3.4: Layout of 26 Byte packet, conversion to 1664 chip packet, and following steps in flow to generate signals received from test equipment.

```
    end
end

packetFSK(size(packetOQPSK,2))=0;
```

The last line in the code above is because each FSK chip depends on the next O-QPSK chip. But there isn't another O-QPSK chip, so we arbitrarily decide the final FSK chip should be 0.

One could instead use the above procedure to pre-convert all O-QPSK chip sequences into MSK sequences. This has been done and displayed in Table 3.2. Then one could build FSK packets directly by concatenating FSK chip sequences. This has a slight disadvantage: each FSK chip depends on the next O-QPSK chip, so pre-converted FSK chip sequences are only 31 chips long and must be arbitrarily padded with 0 or 1. This means every sequence may have one chip error already built in to the 32nd chip.

The digitized IF output was analyzed for mean voltage crossings to determine cycle periods and then per-period frequencies. At the data period, the mean of these frequencies were compared to the nominal intermediate frequency and a 0 or 1 was stored. This flow from voltage to frequency to per-chip frequency is visualized in Figure 3.5. A step of clock recovery was included in data processing to ensure frequency of the IF was measured during one whole bit period instead of measuring frequency partly during one bit and partly another. This array of recovered chips was also correlated with the standard 802.15.4 DSSS chip sequences to find the 32-chip symbols the chip stream best matched.

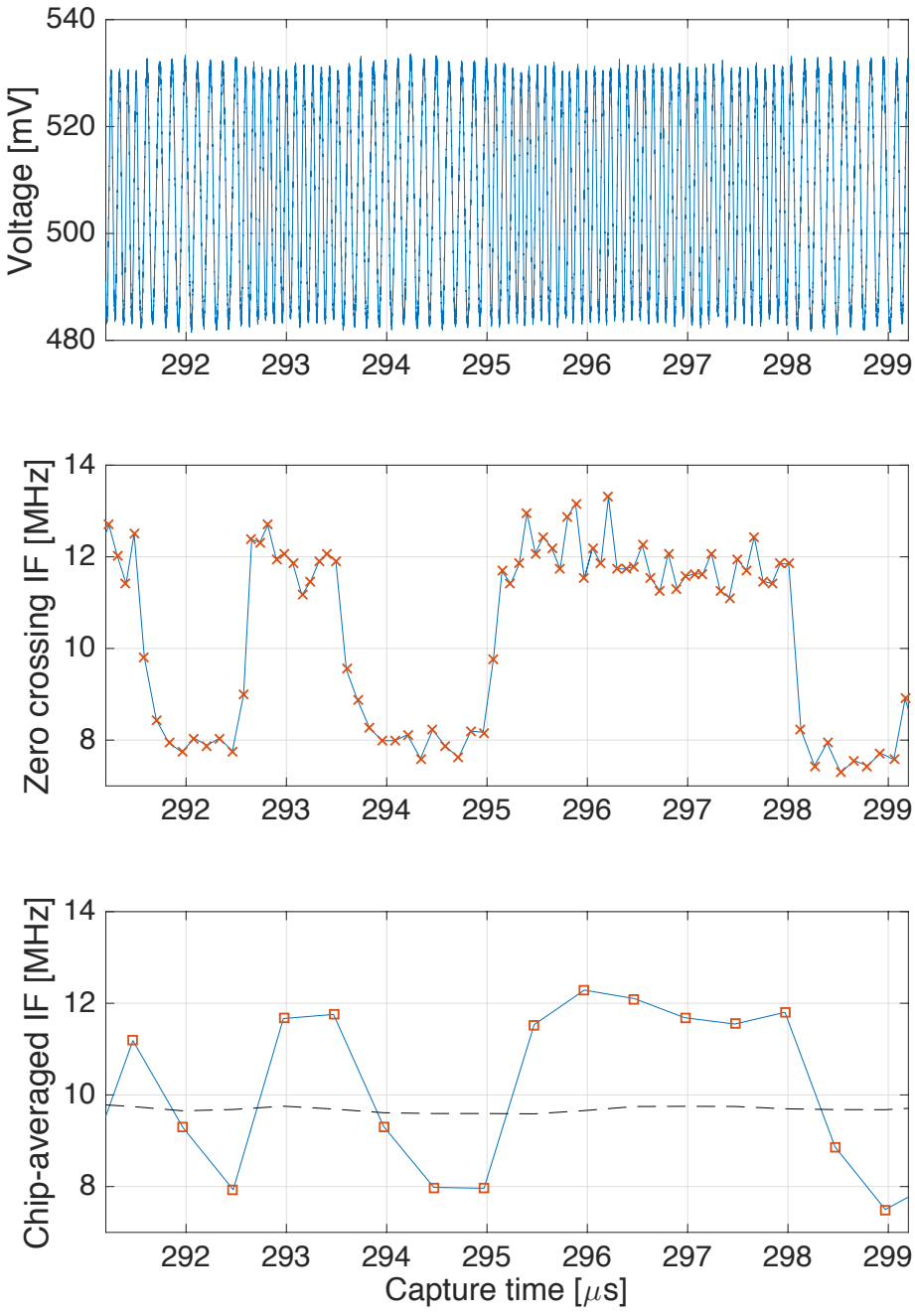


Figure 3.5: Voltage waveform after mixer (top), frequency from zero-crossing-to-zero-crossing period measurements (middle), and average frequency over the last chip period (bottom). A 16 chip sequence `1001100011111100` is displayed. Per-chip frequency is evaluated every $0.5\mu s$ for a data rate of 2Mcps in this example. Black dashed line across bottom plot: average frequency to which per-chip frequency is compared to determine 0 or 1.

Table 3.2: IEEE 802.15.4 data-to-chip conversion for MSK

Binary value $b_3...b_0$ or $b_7...b_4$	31-chip MSK Chip Sequence c_0 ... c_{30}
4'b0000	00111111000100001010001100100 11
4'b0001	01100011111100010000101000110 01
4'b0010	00100110001111110001000010100 01
4'b0011	00110010011000111111000100001 01
4'b0100	10100011001001100011111100010 00
4'b0101	00001010001100100110001111110 00
4'b0110	00010000101000110010011000111 11
4'b0111	11110001000010100011001001100 01
4'b1000	11000000111011110101110011011 00
4'b1001	10011100000011101111010111001 10
4'b1010	11011001110000001110111101011 10
4'b1111	11001101100111000000111011110 10
4'b1100	01011100110110011100000011101 11
4'b1101	11110101110011011001110000001 11
4'b1110	11101111010111001101100111000 00
4'b1111	00001110111101011100110110011 10

3.5 Results

The ring oscillator frequency varied by 1-2MHz over the course of a given 16ms oscilloscope capture. Were we to simply set a frequency as a high/low or 1/0 decision point, as described in Section 3.3, this would have yielded poor results. Instead, we established a high/low decision frequency based on a windowed average of the last several frequency measurements to track the slow frequency changes of the ring. It was found that a 15-point window recovered chips most accurately at all investigated tone spacings besides 0.5MHz ($\pm$0.25MHz), at which no chip sequences were recoverable. Up to 50-point windows were tested with negligible change in recovery rate. A sample set of frequencies during a capture is plotted in Figure 3.6 with a moving average overlaid. Readers will notice the abrupt changes in frequency in addition to the more Gaussian-spreading of frequencies evident in any given segment of time. These shifts are as-yet-unexplained, but the techniques presented successfully compensate for them. Clock and data recovery, taking place after the frequency high/low decision is made, is anticipated to be unaffected by these large jumps in frequency.

Unmodulated intermediate frequencies were first captured to establish the frequency statistics of the ring oscillator without needing to directly capture RF cycles. These data are shown in the left half of Figure 3.2 and indicate that simulation from Section 3.3 and experiment are in good agreement, giving support that our design was fabricated as intended. Experimental data has slightly better standard deviation over 1ms. This is likely due to chance; the full capture is plotted in the top half of Figure 3.9 and indicates that the measured frequency has more variance on longer time scales.

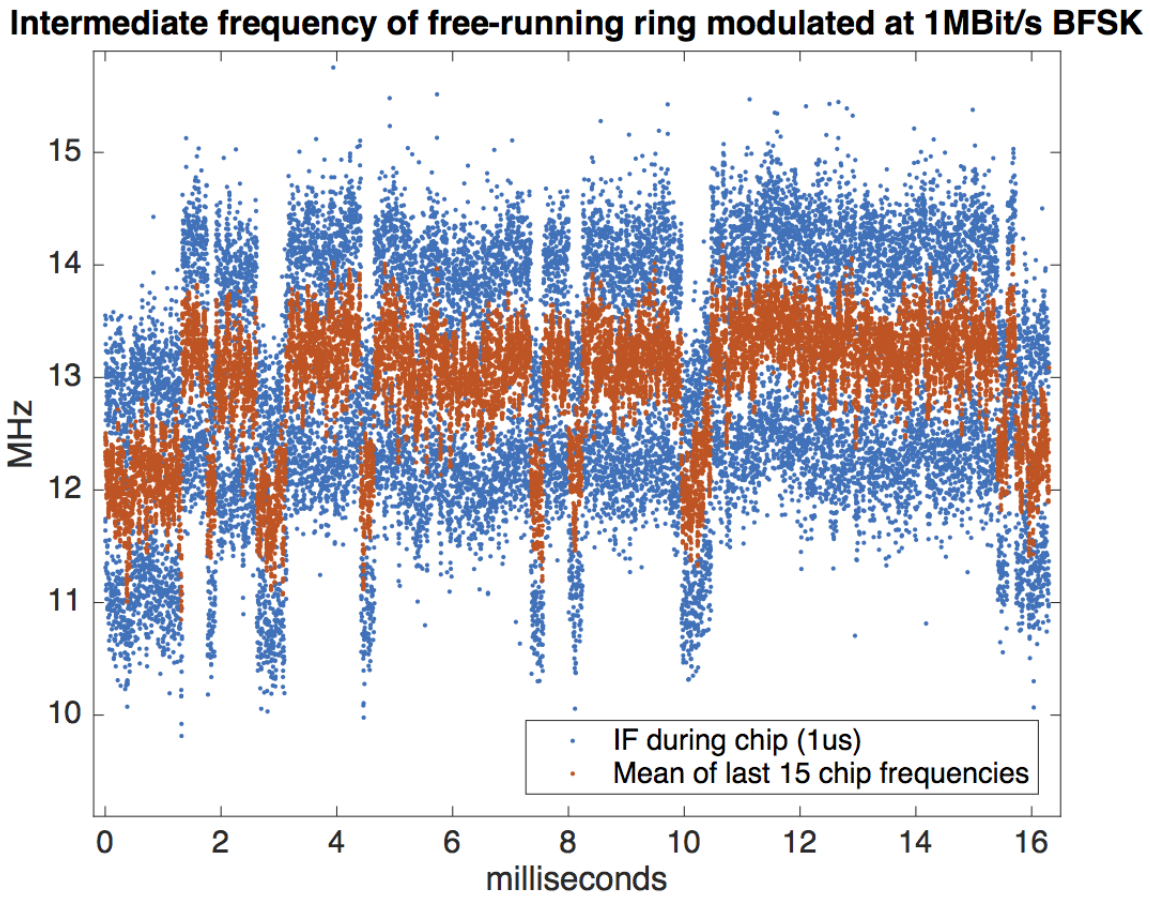


Figure 3.6: Instantaneous frequency of FSK modulated signal (blue, appearing as top and bottom traces) measured via voltage zero-crossing period over time. Also displayed: windowed average of frequency (orange, middle trace). Windowed average tracks slow frequency changes as ring oscillator drifts, sometimes abruptly, and provides decision point by which to judge high or low frequency shift. In this particular data set, the 896 chip test packet is repeated 18 times with 2MHz FSK frequency spacing. Random abrupt frequency shifts move the ring center freq by approximately 1MHz. An explanation for these shifts has yet to be found.

Table 3.3: Comparison to other published work

	This work	[63]	[64]	[62]	[65]	[66]
Frequency	2.4GHz	2.4GHz	2.0GHz	2.4GHz	4.0GHz	4.0GHz
Process	65nm	130nm	90nm	65nm	90nm	65nm
VCO power	105uW	233uW @33kbps	6uW	13uW	280uW	140uW
Ring topology	4-stage differential	Not reported	3-stage single	3-stage single	3-stage single	2-stage differential
Data rate	up to 2Mcps	up to 33kbps	100kbps	250kbps	100kbps	100kbps
Modulation	FSK	OOK	OOK assumed	OOK	FM-UWB	FM-UWB
Noise bandwidth	∼5MHz	100MHz estimated from [73]	100MHz	54MHz	500MHz	300MHz

Results are displayed in Figure 3.7 with chip error rate 6.5% line superimposed. While the 802.15.4 frequency spacing of 1MHz results in greater than 6.5% CER, relaxing tone separation requirements to 2MHz does meet error rate spec. Alternatively, if the application using a free-running ring receiver can tolerate higher packet error, the standard frequency separation will still result in the majority of chips being successfully received.

These results in comparison to other free-running ring oscillator designs are displayed in Table 3.3 and indicate that our work is capable of communicating in a uniquely narrow bandwidth compared to other free-running radios while still consuming a comparable amount of power and communicating at a comparable rate. Reference [16] deserves particular mention for its data rate vs power but is still restricted to OOK and thus lacks the potential for more higher sensitivity via reduced receiver bandwidth and advanced receiver features such as I/Q demodulation.

A photo of the die is shown in Figure 3.8.

Packet error rate

A 16ms capture represented 16000 or 32000 chips, depending on data rate, which yielded a good statistical source on which to base the results of 3.7. However, at 896 or 1664 chips per packet, this means only 18 whole packets were contained in each data set. Nonetheless, we correlated the chip stream, found preambles and start symbols, and despread the chips to recover data. All 18 packets were perfectly recovered in all frequency spacings aside from ±0.5MHz and ±0.75MHz, which recovered 14/18 and 15/18 respectively. At ±1MHz tone

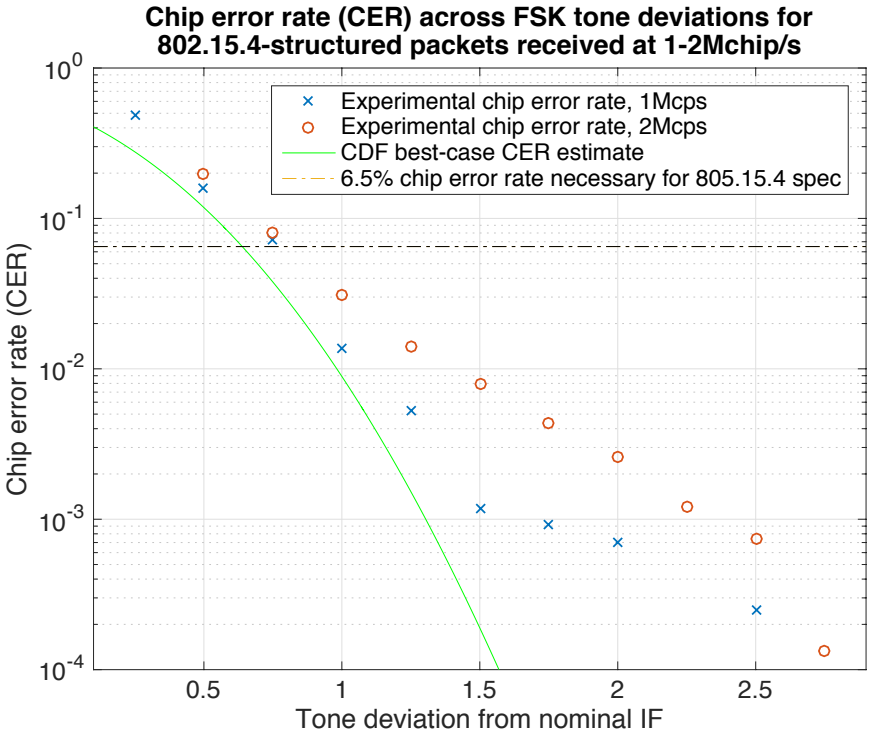


Figure 3.7: Ratio of chips successfully recovered over a variety of FSK tone deviations (tone deviation is $\pm\Delta$f; total tone separation is 2x axis label). Chips were received by circuit at 1 or 2 Megachips/s (Mcps) for 16ms, for a total of 16000 or 32000 chips per data point. Free-running rings do not meet average 1% packet error rate (6.5% chip error rate) at $\pm$0.5MHz and 2Mcps as required by 802.15.4 spec, but does at $\pm$1MHz and wider. Not pictured: A freq deviation of $\pm$3MHz (tone separation of 6MHz) and 1Mcps results in 100% of chips recovered during test.

spacing a longer capture was taken, at only 100MSamp/s for a total of 410ms of acquisition time. After recovering chips and converting to bits, and 487/488 packets were recovered perfectly for a PER of 0.2% or successful packet recovery rate of 99.8%. These results reflect only IEEE 802.15.4 DSSS codes but, were this high-noise recovery method employed with an encoding scheme with higher coding gain, we expect proportionally better PER with closer frequency spacing or higher noise.

Effect on noise bandwidth

This investigation focused entirely on the effect of high phase noise on received chip/packet error rate. This means the IF is unfiltered and the experiment was designed with high SNR to avoid confounds from other sources of noise. In a real system, maximizing SNR to maximize sensitivity is a high priority. An effective way to limit noise is to limit bandwidth.

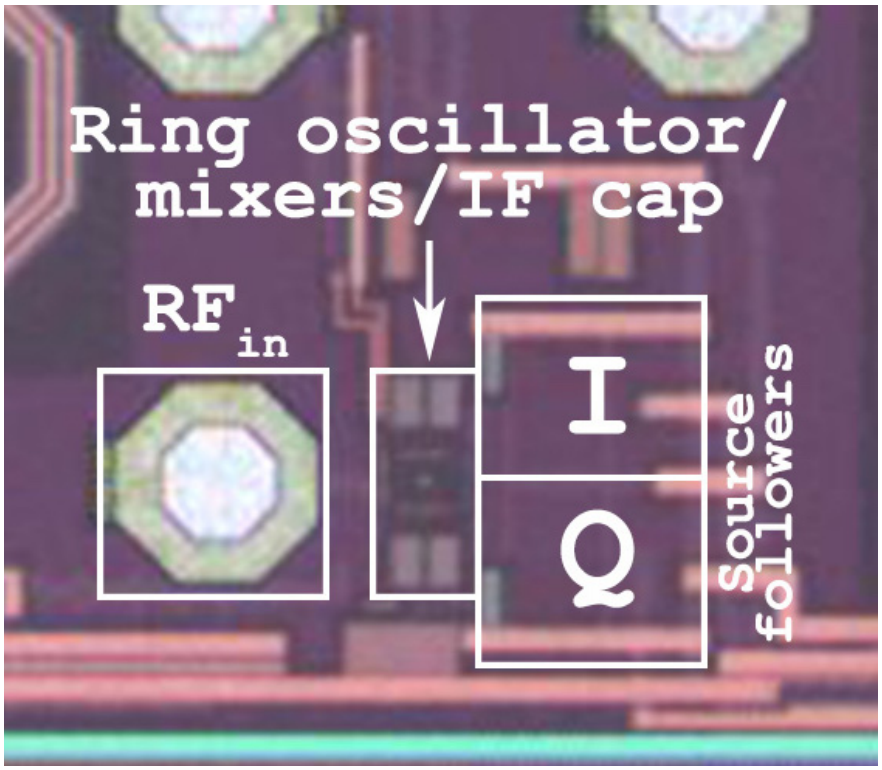


Figure 3.8: Photo of section of IC containing circuits under test. Displayed IC section is 450um wide by 400um tall. Our 4-stage differential ring oscillator is the white dot inside the box labeled to contain ring, mixers, and IF capacitors. Not shown: pads for power, current mirror bias, and IF I & Q out.

In MSK-based systems, the null-to-null width of a modulated tone is 1.5x the data rate [70], so a signal modulated at 2Mchip/s as in 802.15.4 can limit its noise bandwidth to about 3MHz. In this work, the ring oscillator's noise profile results in a large range of intermediate frequencies. This full range must be accommodated by an IF band-pass filter to recover as much received data as possible. A free-running capture is plotted in the top half of Figure 3.9 as an example, indicating approximately 5MHz to capture the majority of received power. A review of other data yields worst-case bandwidth of about 8MHz. This worst-case is observed in both 16ms and 400ms captures. Increasing IF bandwidth to 8MHz would result in SNR of about 4.3dB worse than the null-to-null required for MSK.

Much of the variance in the top half of Figure 3.9 is due to low-frequency wander of the ring oscillator. The ring was biased by precision DAC, so a feedback system was built to count IF cycles over periods of 100ns and correct the DAC code appropriately. The 100ns timer was derived from a divided FPGA clock. The intended effect was to correct slow frequency variations and reduce necessary bandwidth. The results are displayed in the bottom half of Figure 3.9 and indicate a bandwidth of only around 2MHz is needed to capture the majority of received power. Some larger frequency deviations are still present on account

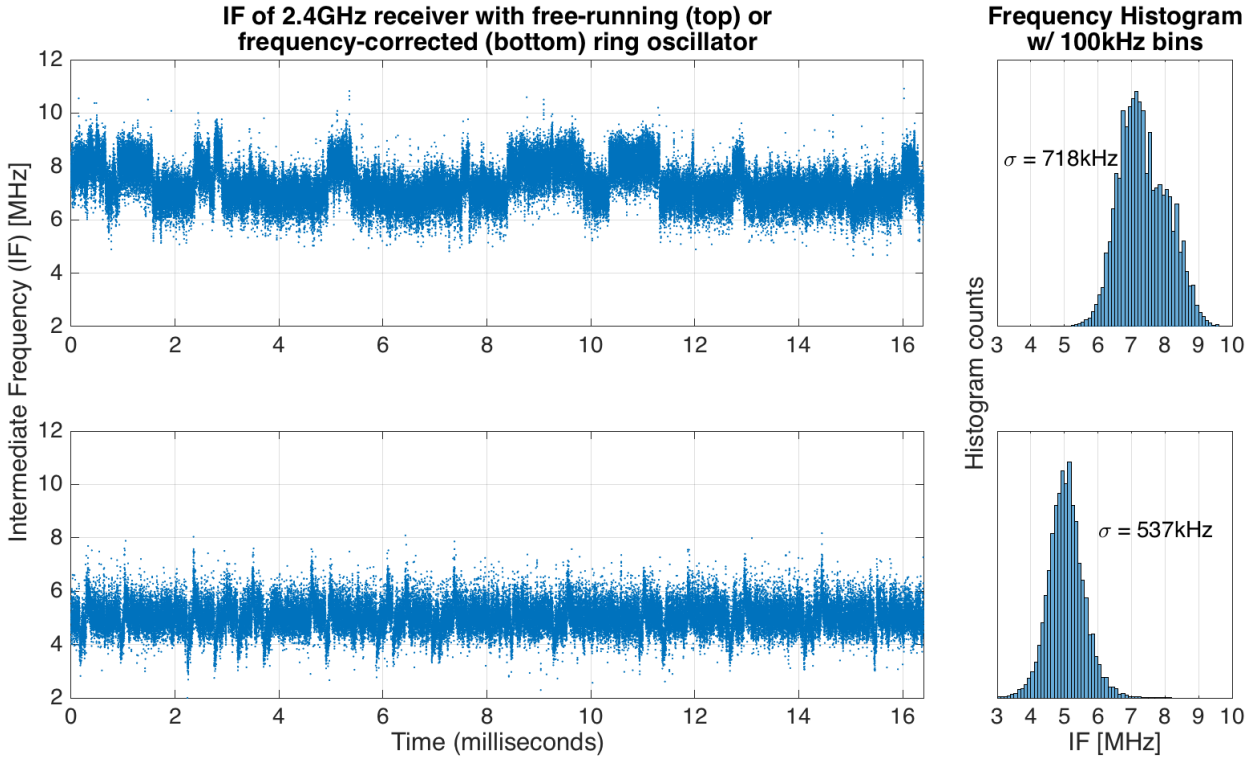


Figure 3.9: Top: IF over time, and IF histogram over all displayed time, of free-running ring mixing down an unmodulated tone showing large random frequency changes. Standard deviation is 718kHz. Bottom: IF over time, and IF histogram over all displayed time, of ring with frequency correction. Correction was based on cycle counts, applied every 100ns to center IF at 5MHz. Large random disturbances evident in any free-running time series have been corrected out. Some frequency disturbances still exist due to inconsistent cycle counter behavior. Standard deviation is 537kHz. Note: first 1ms of top plot forms basis of histogram for right-side histogram in Figure 3.2.

of inconsistent cycle counter behavior. This plot displays an unmodulated IF, so additional bandwidth is dependent on FSK tone spacing. We speculate that this distribution added to an identical one shifted 1MHz away, as in the time spacing for MSK at 2Mchip/s, would add another 1MHz of required bandwidth for a total of 3MHz, just as in typical 802.15.4.

Such a feedback system requires high initial bandwidth. A system could be devised to start with wide bandwidth to find the incoming carrier while receiving preambles, enable 100ns (for example) correction of the local oscillator, and switch in a narrower filter to reduce noise bandwidth while receiving data. Prior work has described efficacy of this concept [74]. Care must be taken with system design, as this method is susceptible to locking on to the highest-power signal in the IF bandwidth including strong interferers.

"IF Jump" Hypotheses

Some hypotheses were developed in an attempt to explain the observed ~1MHz jumps displayed in Figure 3.6 and the top half of Figure 3.9. At the transition point, we can model our differential ring oscillator as a differential RC oscillator and write an expression for the frequency accordingly per [22]:

$$f = \frac{1}{2M}\frac{I}{CV_{op}ln2} \tag{3.1}$$

Where M is number of stages, C is the capacitance of the transitioning node, I is the current available to fill C, and V_{op} is the oscillator voltage swing. Swing and current are our most likely sources of frequency jumps. The most likely inputs that could be responsible for changing swing and current are:

- V_{DD}: abrupt supply changes will change bias point.
- V_{GS}: V_t shifts and changes in mirrored current can result in changing the gate voltage of the current source device in Figure 3.1a.
- Temperature: Heating or cooling the chip will change frequency.

It seems unlikely V_t, mobility, or other changes inside the delay cells are responsible for the frequency shifts on account of the shifts only happening in two discrete levels. If all four delay cells were susceptible to random events leading to 1MHz frequency shifts, we would expect to see at least 2^4 different discrete voltage shift levels.

We investigated to what extent these inputs needed to be changed to affect a 1MHz change while the oscillator was running at 2.4GHz.

- V_{DD}: Finely adjustable benchtop voltage sources add substantial noise, obscuring measurement. For example, supplying bias current to the SCM1A ring from a Keithley 2634B SMU increases frequency jitter to approximately 10-20MHz, a roughly 10x increase over noise observed when biasing w/ resistor. In simulation, we need to raise V_{DD} by about $300\mu V$.
- V_{GS}: As mentioned earlier in Section 3.3, measurement tells us we need to change the current bias pin by 101nA or 84μV to affect 100kHz change. This means we can expect 840μV to shift by 1MHz. The current bias pin is directly connected to the gate of the, so we can be confident we are adjusting V_{GS} as a proxy for a V_t shift. In simulation, a series voltage source placed directly on the current mirror device gate must be set to 3.54mV to result in 1MHz frequency shift.
- Temperature: Sub-degree IC temperature control was not feasible in the lab, but a simulated temperature increase of 0.68°C slowed down the oscillator by 1MHz.

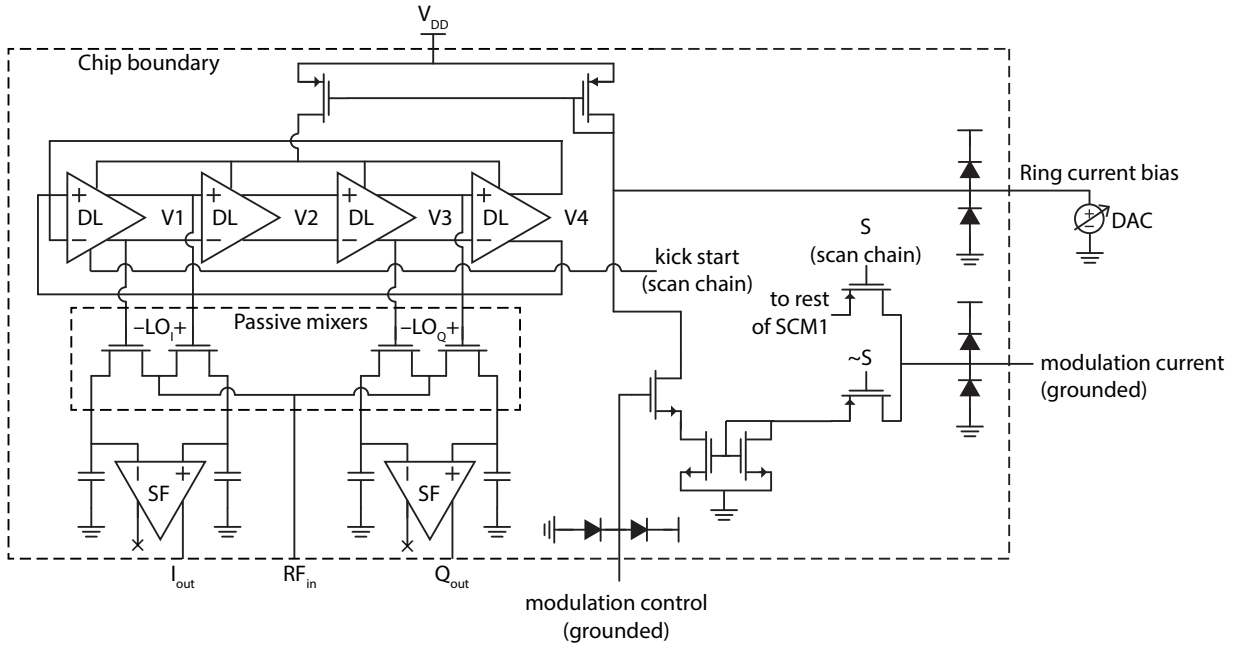


Figure 3.10: More complete schematic of SCM1A ring oscillator receiver including unused control pins and additional devices attached to current source gate bias node. Note all scan chain signals are powered from a different supply.

Unfortunately, none of these hypotheses seem like reasonable explanations. The LDO providing V_{DD} is specified at about $30\mu V$, or about 10x less than we need to cause a 1MHz shift. A 1-3mV shift in threshold voltage or gate voltage of a 65nm transistor seems unlikely, especially only in the two discrete levels observed. And temperature will vary during operation, but we expect temperature to stabilize under constant operating conditions instead of bouncing ~0.7°C.

Another hypothesis is interference from the unused components in the rest of the chip. One or more dormant circuits could charge on/discharge off through parasitics at a random rate. A more complete schematic is shown in Figure 3.10. The relevant routes from the current source are switched off externally but, regardless, some of this network is connected to other parts of the IC and the scan chain itself is powered from a different supply. But with limited control pins available, We lack ample experimental capability to test this hypothesis.

3.6 Conclusion

We have demonstrated that, contrary to popular belief, narrowband communication is possible with a 2.4GHz free-running ring oscillator burning only 105μA. A receiver using such a ring for a local oscillator is capable of recovering packets transmitted by an IEEE 802.15.4-compliant transmitter, but recovering around 75% of received packets. Alternatively, if the

FSK specification were widened from ±0.5MHz to ±1MHz, about 99.8% of packets can be received perfectly.

We speculate that we could improve performance by two different methods. The first is to burn more current in a larger ring to reduce thermal noise contributions and bring variance down, effectively shifting the data in Figure 3.7 down or left. This will work if the dominant source of noise is thermal; increased current will *increase* variance if the dominant source is flicker. The noise in this chapter's device under test is likely dominated by flicker noise, so a new design is necessary to first reduce flicker contributions. Heavy filtering of the current mirror is one way. Foundry-supplied simulation models greatly (exactly how much has yet to be determined) underestimate flicker noise so models must first be hand-tuned to match measured flicker contributions. Then, new designs to reduce flicker contributions can be tested.

The second is by putting the RF oscillator in a feedback loop that compares the IF with a reference frequency. The loop need only have bandwidth ∼10x beyond the rate of the random ∼1MHz jumps, or about 100 kHz. The magnitude of the frequency jumps is on the order of 10%, so an RC oscillator with typical ∼0.1% jitter could serve as a loop reference. With a loop bandwidth in excess of 10MHz, one could correct IF variance as well. This is above the modulation rate and one would use the VCO feedback control signal to demodulate instead of sampling the IF. Given the IF variance is over 6%, we speculate that an RC oscillator would also suffice here.

The idea of using an RC oscillator as the reference for a PLL or FLL is usually dismissed because RC oscillators have poorer frequency variance at all time scales as compared to the oscillators to which they provide correction. The ring oscillators in this dissertation have about 100-200ppm error at all tested time scales (discussed later in Chapter 5 and displayed in Figure 5.4), whereas some tested RC oscillators exhibit as low as 10ppm at 1s (Figure 5.6) or 10ms (Figure 5.5). With loop bandwidths of 1Hz or 10Hz, an RC oscillator reference would improve the LO in a traditional loop.

While the receiver is actively receiving an incoming signal, we have a better opportunity. A mixing operation is subtraction, not division, and that means we can compare a low IF with a reference of the same nominal frequency directly without needing a divider. Feedback corrections apply directly to the RF oscillator and the variance of the reference is scaled by the IF/RF ratio in the RF oscillator's point of view. For example, a 4.8 MHz IF generated by mixing incoming signal with a 2.4GHz oscillator would scale reference variance by 500. Hence, an RC oscillator with typical 0.1% (1000ppm) cycle-to-cycle jitter could have an "effective jitter," or the same RC jitter with respect to 2.4GHz, of 2ppm. Taken the other way, the 2.4GHz RF oscillator has 100ppm error, or about 240kHz. But 240kHz error on a 4.8MHz IF is 50,000ppm. Our 1000ppm RC oscillator reference is over 2 orders of magnitude cleaner by comparison. This concept is also discussed in [5, 6, 12, 75].

Either method is a feasible way to push the performance of this free-running receiver over the edge of 1% packet error rate as prescribed by the 802.15.4 specification.

Chapter 4

Jitter and Phase Noise

This chapter contains background about phase noise, jitter, Allan Variance/Deviation, etc. This material, combined with a few carefully-generated Matlab models to demonstrate how each of these concepts are related, might be the start of a course someday. Other ways to complement this material include demonstration of how these concepts are related via real measurements − with the caveat that some sources don't recommend generating Allan variance from phase noise data on account of the result being very noisy − and tying those results to theoretical best demodulation performance.

4.1 Phase noise and jitter properties

This section is the overlapping combination of several sources, some of which explain various concepts more clearly or in more detail than others [76, 77, 78, 79, 80, 81, 82, 83]. There is no one definitive, best, source from which to draw these concepts so we cite the generally-consulted works up front and make specific, possibly redundant, citations as appropriate later.

Uncertainty in the time at which intermediate frequency voltage crosses zero (or the average voltage) is the result of small phase perturbations integrated over time. Typical definitions of these perturbations, collectively called phase noise, start with a phase error term in the sinusoidal expression representing voltage output of an oscillator designed to run at carrier frequency f_c:

$$v(t) = sin(2\pi f_c t + \phi_n(t)) \tag{4.1}$$

Where $\phi_n(t)$ is the time-dependent phase error containing white noise, flicker noise, and anything else the designer wishes to include in the model. It is claimed that an ideal noiseless spectrum analyzer could measure $\phi_n(t)$ directly [77]. In practice, phase noise is measured in a few different ways with the goal of obtaining the power spectral density (PSD) $\mathscr{L}(f_m)$, which is the typical starting point for phase noise-to-jitter calculation. Below is a brief description

of basic phase noise measurement types but more advanced methods have been developed [84].

Methods of measuring phase noise

Again, the goal of various measurement methods is to obtain $\mathscr{L}(f_m)$, which is a ratio of power density per unit frequency (offset frequency from carrier) to total signal power. It has an integral of unity and is typically expressed in log units referred to carrier power, dBc/Hz. Since the integral of the PSD is unity, one interpretation of PSD is the proportion of time the oscillator spends at a given offset frequency from its designed carrier frequency.

Direct measurement: spectrum analyzer

This is the most straightforward method and directly measures power at a given frequency. Spectrum analyzer values need only to be scaled to the total carrier power and frequency to obtain $\mathscr{L}(f_m)$. This method will accrue significant error for signals with such significant phase noise that the center frequency drifts during the measurement, or for signals with large amplitude noise since the spectrum analyzer can't distinguish amplitude noise from phase noise. Typically amplitude noise can be considered not to be "large" if it is 10dB less than phase noise or smaller.

Phase detector

Phase detectors mix the signal of interest with a reference signal at the same center frequency but with much lower phase noise. A low-bandwidth PLL keeps the reference oscillator locked to the oscillator of interest and at a 90° offset such that phase deviations faster than the PLL bandwidth are still evident. The mixer output is filtered and amplified; the result is a voltage proportional to phase. This method is insensitive to amplitude noise unlike when using a spectrum analyzer.

Phase detectors produce magnitude of phase deviation at a given carrier offset frequency. We calculate phase spectral density in units of phase variance per unit bandwidth:

$$S_{\Delta\phi}(f_m) = \frac{\Delta\phi^2_{rms}(f_m)}{BW\ used\ to\ measure\ \Delta\phi_{rms}} \left[\frac{rad^2}{Hz}\right] \tag{4.2}$$

Where bandwidth (BW) must be negligible with respect to any changes in $S_{\Delta\phi}$ vs. f_m. In log, this is referred to 1 rad:

$$S_{\Delta\phi}(f_m) = 20log_{10}\frac{\Delta\phi_{rms}}{1rad}\left[\frac{dBr}{Hz}\right] \tag{4.3}$$

This can be converted to power spectral density per the NIST definition [85]:

$$\mathscr{L}(f_m) = S_{\Delta\phi}(f_m) - 3dBc \tag{4.4}$$

Equation 4.4 assumes the small angle criterion is met. This is generally taken to be true if peak phase deviations ϕ_{peak} are less than 0.2 rad per decade. We speculate that this commonly-repeated specification comes from $\phi^2_{peak} = (0.2/2\pi)^2 = 10^{-3}Hz^2$.

It follows that at $f_m = 1Hz$:

$$S_{\Delta\phi}(1Hz) = 20log_{10}\frac{\Delta\phi_{peak}}{1Hz} = 20log_{10}\frac{0.2rad}{1Hz\frac{2\pi\,rad}{1Hz}} = -30dBc \tag{4.5}$$

or less, to satisfy the criterion. If phase deviations are greater than 0.2 rad, the equality in (4.4) does not hold and the original phase spectral density measurements $S_{\Delta\phi}(f_m)$ must be used for calculations.

For frequency offsets greater than 1 Hz, maximum allowable phase error decreases by -10dB per decade. This trend is presumably obtained by scaling maximum allowable phase error by offset frequency f_m (but this is not explicitly stated in reference literature):

$$S_{\Delta\phi,Limit}(f_m) = 10log_{10}\left(\left(\frac{\Delta\phi_{peak}rad}{2\pi\,rad}\right)^2 \cdot \frac{1}{f_m}\right) \tag{4.6}$$

Frequency discriminator

A frequency discriminator measures by mixing outputs of a delay line and phase shifter, both of which are fed the signal of interest. Frequency deviations are converted into phase deviations and, by mixing, are converted into voltage deviations. Since this method requires no frequency tracking loop, it is great for oscillators that drift too quickly to keep locked during measurement. The voltage response has characteristics of a sinc function with nulls at frequency inversely proportional to the delay of the delay line, so some care must be taken to avoid those nulls during design.

The output of this measurement system is frequency deviation at a given carrier offset frequency. We calculate frequency spectral density in units of phase variance per unit bandwidth:

$$S_{\Delta f}(f_m) = \frac{\Delta f^2_{rms}(f_m)}{BW\ used\ to\ measure\ \Delta f_{rms}}\left[\frac{Hz^2}{Hz}\right] \tag{4.7}$$

Where bandwidth (BW) must be negligible with respect to any changes in $S_{\Delta f}$ vs. f_m. In log, this is referred to 1 Hz:

$$S_{\Delta f}(f_m) = 20log_{10}\frac{\Delta f}{1Hz}\left[\frac{dBHz}{Hz}\right] \tag{4.8}$$

This can be converted to power spectral density:

$$\mathscr{L}(f_m) = S_{\Delta f}(f_m) - 20log_{10}\left(\frac{f_m[Hz]}{1[Hz]}\right) - 3dB \tag{4.9}$$

Interpreting $\mathscr{L}(f_m)$

All surveyed instruments generate power spectral density plots regardless of measurement method. The plot of $\mathscr{L}(f_m)$ can be broken into regions in which different frequency-dependent noise trends dominate. The trend lines decrease in slope as one moves further from the carrier. For example, in the artificial plot in Figure 4.1, a f_m^{-4} trend is observed closest to carrier when f_m is very low (very close to carrier frequency) and proceeds through $f_m^{-3}...f_m^{-0}$ as offset frequency f_m increases. Each trend is typically called by a particular name in addition to its f_m^{-x} trend. Commonly-used names are labeled in the legend of Figure 4.1. By this naming pattern one might a;so refer to white frequency noise as "random walk phase noise" but, in practice, these names are the agreed-upon ones.

Typical sources of these noise trends are below. How these types of noise appear in various types of spectrum plots is given in Table 4.1.

- White phase noise: white noise at oscillator output from buffers, electronics, etc.
- Flicker phase noise: flicker version of above.
- White frequency noise: white noise on an oscillator control line (e.g., current source node) or coming from the oscillator devices themselves upconverted into frequency variations (white f^0 noise becomes f^{-2} noise).
- Flicker frequency noise: flicker version of above (flicker f^{-1} noise becomes f^{-3} noise).
- Random walk frequency noise: temperature drift, aging, and more effects that aren't well understood.

Some f_m^{-x} trends may be obscured in a real plot by measurement error, other sources of noise, oscillator wander during measurement, or simply a low quantity of a particular noise source. It is rare to obtain a PSD plot in which all five power law spectra regions can be recognized. Regions closer to the carrier represent oscillator behavior at time scales longer than, e.g., PLL bandwidth or packet length, so it is also rare for a designer to pay close attention to those high-drift close-to-carrier regions in the first place.

Table 4.1: Noise process trends in various types of plots (concisely available on p.14 of [86])

Noise name	Trend in Allan Deviation $\sigma_y(\tau)$	Trend in phase spectrum $S_{\Delta\phi}(f_m)$	Trend in frequency spectrum $S_{\Delta f}(f_m)$
White phase	τ^{-1} (yes, -1)	f_m^0	f_m^2
Flicker phase	τ^{-1}	f_m^{-1}	f_m^1
White freq.	$\tau^{-1/2}$	f_m^{-2}	f_m^0
Flicker freq.	τ^0	f_m^{-3}	f_m^{-1}
Random walk freq.	$\tau^{1/2}$	f_m^{-4}	f_m^{-2}

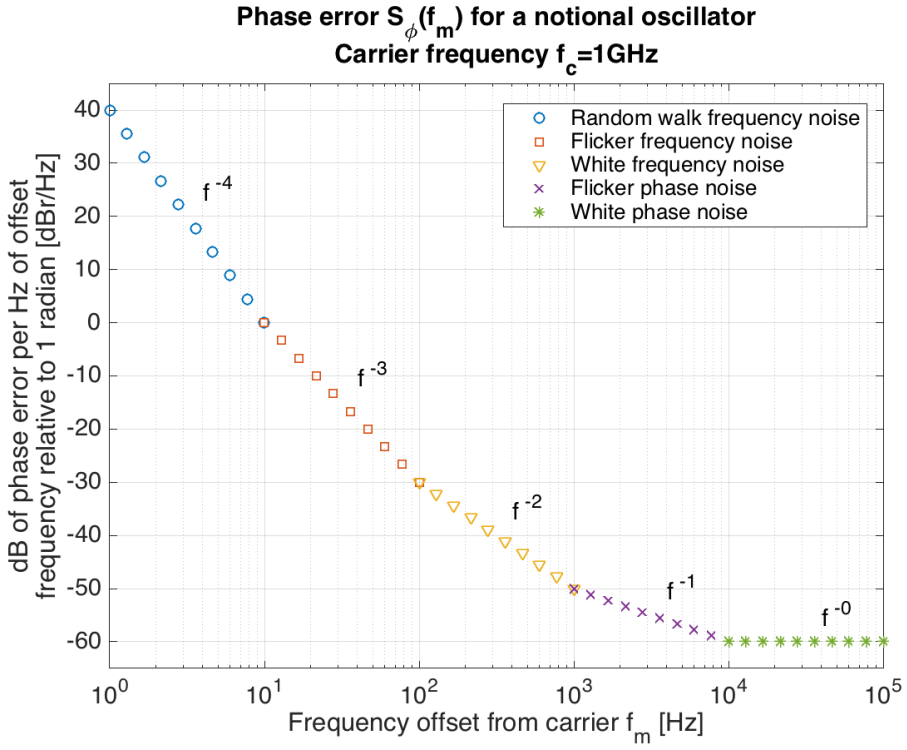


Figure 4.1: Notional phase noise profile with frequency trend of each noise type labeled. Defined regions based on Figure 2.4 of [78] and adjusted such that each region of noise occupies one frequency decade for clarity. Note: phase noise and jitter are related, but the plots in this figure and Figure 4.2 to not correspond to one another.

White frequency noise and flicker frequency noise tend to be the types most focused on by oscillator designers. These are sources inherent in the oscillator and the basis of power/noise tradeoffs, whereas lower-ordered noise is often due to measurement electronics and higher ordered noise comes from long-term drift effects. A perfect, noiseless, oscillator buffered with a realistic voltage buffer would still exhibit f^{-1} and f^0 noise due to buffer noise.

Another reason for the focus on white/flicker frequency noise is that the "knee" between white and flicker frequency noise in an Allan Deviation plot (at $\tau = 10^0$s in Figure 4.2) is also the point at which averaging no longer improves frequency error relative to the averaging window time τ. Fractional frequency error stagnates and hits a "flicker floor," later getting worse with time ($\tau = 10^1$s and beyond in Figure 4.2) due to random walk frequency noise.

Measurements greater than 0 dB

While obtaining measurements of phase noise in its various forms, it's often possible to have values greater than 0 dB close to the carrier. Depending on what is being measured, this could have a variety of meanings.

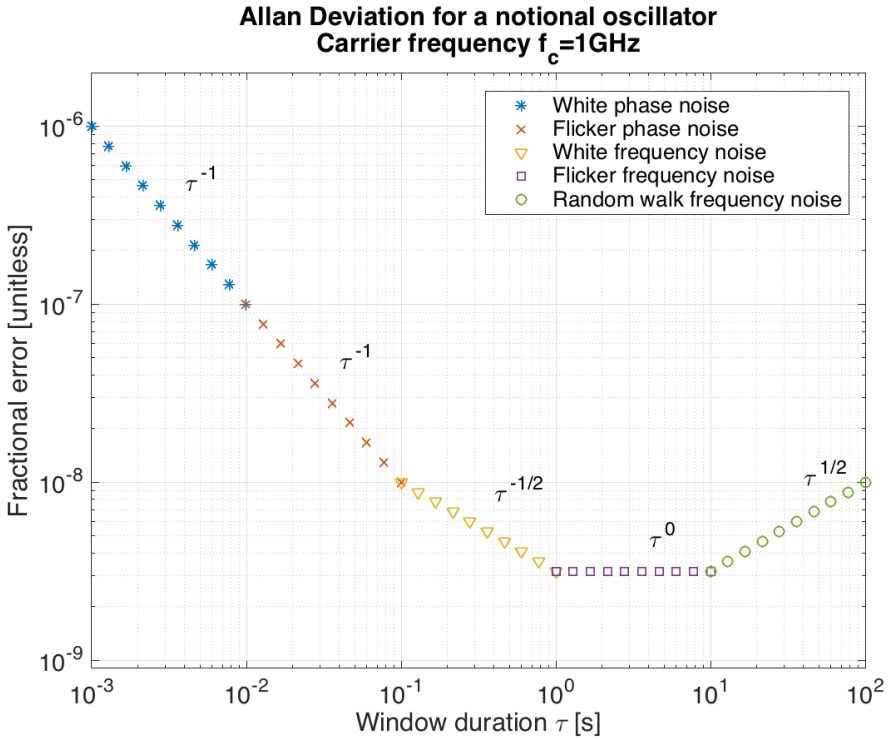


Figure 4.2: Notional Allan Deviation profile with period trend of each noise type labeled. Note: phase noise and jitter are related, but the plots in this figure and Figure 4.1 to not correspond to one another.

- Using a phase detector, frequency discriminator, or otherwise measuring the phase spectrum $S_{\Delta\phi}(f_m)$ or frequency spectrum $S_{\Delta f}(f_m)$: since these measurands are referred to 1 rad or 1 Hz, respectively, 0 dB or greater simply means phase or frequency variations are greater than 1 rad or 1 Hz, respectively [87].
- $\mathscr{L}(f_m)$ calculated from $S_{\Delta\phi}(f_m)$ or $S_{\Delta f}(f_m)$: greater than 0 dB means more power at a frequency offset f_m than total carrier power, which is impossible. This result is likely because $S_{\Delta\phi}(f_m)$ or $S_{\Delta f}(f_m)$ violate the small angle criterion, described by Equation 4.5
- You're simulating with PSS and PNOISE in spectreRF. The simulator will report results with impossibly large voltages. Use the "lorentzian=yes" option in PNOISE to fix [88]. The AFS simulator displays a Lorentzian by default [89].

4.2 Variances

This section follows along with the derivation presented in [90]. The terms "deviation" and "variance" may be used interchangeably with the knowledge that deviation (or standard

deviation) σ is the square root of variance σ^2.

An oscillating signal $y(t)$ will have some average fractional frequency deviation $\overline{y}_k$, over a period of time τ offset by some initial time t_k, defined as:

$$\overline{y}_k = \frac{1}{\tau}\int_{t_k}^{t_k+\tau} y(t)dt \tag{4.10}$$

This can also be written as the difference between frequency measurements $x(t)$ at the beginning and end of the τ-long period:

$$\overline{y}_k = \frac{x(t_k+\tau) - x(t_k)}{\tau} \tag{4.11}$$

One could determine $\overline{y}_k$ N times over and calculate the variance, as is typically done with random processes. This works fine for Gaussian white frequency noise (the result converges) but for flicker and random walk frequency noise the variance grows as one adds more measurements.

Allan Variance, Allan Deviation (AVAR, ADEV)

The Allan variance $\sigma_y(\tau)$ solves this divergence issue by calculating not the variance of $\overline{y}_k$ but instead the variance of the *difference* between two adjacent $\overline{y}_k$ values.

$$\sigma_y^2(\tau) =< \frac{1}{2}(\overline{y}_{k-1} - \overline{y}_k)^2 > \tag{4.12}$$

The angle brackets denote taking the average for infinite time. Because there are two $\overline{y}_k$ samples above, this is often called "two-sample" variance. This can also be written as the differences between frequency measurements $x(t)$ at the beginning and end of both adjacent τ-long periods (which ends up consisting of three $x(t)$ measurements at beginning, middle, and end of a 2τ-long period) for N repeated trials instead of infinite time:

$$\sigma_y^2(\tau) \approx \frac{1}{2(N-2)\tau^2}\sum_{i=1}^{N-2}(x_{i+2} - 2x_{i+1} + x_i)^2 \tag{4.13}$$

Because the above there are three $x(t)$ measurements, the Allan variance is sometimes also called "three-sample" variance. Literature is inconsistent between Allan, two-sample, or three-sample variance terms.

Finally, as the width τ of a $\overline{y}_k$ measurement varies, one obtains different Allan variance values for different averaging times. The results trace out the typical Allan Variance (or, taking the square root, Allan Deviation) plot.

Modified Allan Variance, Modified Allan Deviation (MVAR, MDEV)

The Allan Variance calculates differences between adjacent $\overline{y}_k$ values of length τ. The Modified Allan Variance calculates $\overline{y}_k$ values of length τ for as many overlapping windows as are available in the data set, not just for as many adjacent windows fit next to each other in the data set.

For example, if we have 101 $x(t)$ samples spaced τ apart for a total data set duration of 100τ, that yields 100 $\overline{y}_k$ values with which to compute AVAR at a period τ. Later, if we wanted to calcuate AVAR at 10τ, traditional AVAR says we should use adjacent measurements so our 100 τ-long data set would contain 10 $\overline{y}_k$ values. But if the windows overlap instead of abut, we can extract 91 $\overline{y}_k$ values of length 10τ from this data set. We start with zero offset and calculate with as many non-overlapping windows as will fit in the data set, then we offset by 1 measurement and calculate with non-overlapping windows again, and so on, eventually having used 91 windows to determine the result.

MVAR is represented as $\mathrm{mod}\sigma_y^2(\tau)$ in the reference on which this section is based, but more often MVAR is represented the same as AVAR, $\sigma_y^2(\tau)$, with a note in the caption or plot about what was used. Modern compute systems are very powerful compared to when AVAR was developed so MVAR is the modern standard and should be assumed unless otherwise stated. In terms of frequency measurements $x(t)$, MVAR can be calculated for N samples as follows:

$$mod\sigma_y^2(\tau) \approx \frac{1}{2\tau^2 n^2 (N-3n+1)} \sum_{j=1}^{N-3n+1} \sum_{i=j}^{n+j-1} (x_{i+2n} - 2x_{i+n} + x_i)^2 \tag{4.14}$$

Where n is the number of adjacent x_i values available in the data set for a given initial offset.

Time Variance, Time Deviation (TVAR, TDEV)

AVAR and MVAR are frequency measurements scaled by the period τ over which those measurements were taken. They are measures of fractional frequency error: error per unit time. If instead we want to know absolute error accumulated over time, we start again at Equation 4.10 but omit the τ scaling factor:

$$x(\tau) = \int_{t_k}^{t_k+\tau} y(t)dt \tag{4.15}$$

Where $x(t)$ is the same x that first appeared in Equation 4.11. The derivation for AVAR or MVAR continues but with this small change to yield TVAR. Hence, the deviation of time $\sigma_x(t)$ ends up being similar to AVAR but in units of time, not fractional error.

This also means the slopes of a TDEV plot are increased by 1: for instance, where white noise would trend as $\tau^{-1/2}$ in ADEV, the same noise trends as $\tau^{1/2}$ in TDEV. TDEV, as used in this dissertation, is mathematically expressed in terms of MDEV [91]:

$$\sigma_x(t) = mod\sigma_x(t)\frac{\tau}{\sqrt{3}}. \tag{4.16}$$

4.3 Converting phase noise to jitter

Jitter is the standard deviation of waveform period and has units of seconds, but there are many types of jitter. Common forms include:

- *Jcc* or cycle-to-cycle jitter: this is the standard deviation of waveform period measured cycle-to-cycle. For white noise sources, we expect this value to converge as we take more and more samples. This is based on two zero crossing times so it is also known as a 2-sample measurement. For flicker noise sources, we expect the difference between two adjacent waveform period measurements (e.g., the Allan Variance) to converge as we take more and more samples.
- K-period jitter (sometimes other letters besides K), which is the same as cycle-to-cycle jitter but compares periods of K cycles instead of only 1.
- *Jc* or cycle jitter: the standard deviation of the difference between the zero crossing time measured and the zero crossing time of a perfect frequency source. This is similar to, but not the same as, *Jcc*. We expect this quantity to grow with time as more and more phase perturbations are accumulated.

Since jitter is a standard deviation unless otherwise specified, one must remember that, e.g., a period specified with ±1ps of jitter will only be within 1ps 68% of the time. This corresponds to one standard deviation on either side of the mean, assuming Gaussian noise.

Computing analytically

If white noise were the only noise contribution (this is a good approximation for many use cases), we could model it directly using a Lorentzian function and curve-fitting the c parameter as described by Equation (41) in [92]. That equation is duplicated here and adapted for our notation by using f_c for carrier instead of f_0 in the original:

$$\mathscr{L}(f_m) \approx 10\ log_{10}\left(\frac{f_c^2 c}{\pi^2 f_c^4 c^2 + f_m^2}\right) \tag{4.17}$$

We can write a simplified version of the above in the region where $f_m^2 \gg \pi^2 f_c^4 c^2$

$$\mathscr{L}(f_m) \approx 10\ log_{10}\left(\frac{f_c^2}{f_m^2} c\right) \tag{4.18}$$

From this, we can see the c parameter describes the rate at which noise is accumulated as we change f_m, and that the function has a f_m^{-2} white noise trend. This simplified expression can be used with Equation 4.24 (introduced later) and integration limits $f_1 = 1/T_1$ and $f_2 = \infty$ to obtain (modulo a $1/\pi\sqrt{2}$ factor) the white-only jitter expression found in [92]:

$$\sigma^2(t) = ct \tag{4.19}$$

This expression is found in various forms in literature. For instance, [93] writes something similar with κ:

$$\sigma(\Delta T) \approx \kappa\sqrt{\Delta T} \tag{4.20}$$

The authors of [94] have a jitter expression using SSCR, or Single Sideband to Carrier Ratio, which is likely the same as $\mathscr{L}(f_m)$. Their Equation (6) is repeated below, and would arise if we solved for c in (4.18) above and substituted that into (4.19):

$$\sigma_{To}^2 = \frac{f_m^2}{f_o^3} SSCR(f_m) \tag{4.21}$$

These expressions all consist of a constant and a f_m^{-2} trend to describe how edge uncertainty grows as the time window increases (in the frequency domain: as $\mathscr{L}(f_m)$ is integrated from $f_m = \infty$ to f_m values approaching zero). They are compact and easy to work with but, as mentioned earlier, only apply to situations in which white frequency noise (f_m^{-2} trend) is the only noise source. In reality, we need to integrate $\mathscr{L}(f_m)$ to factor in non-white noise sources.

Integrating power spectral density

Frequency error $j(t)$, which can be calculated by phase error $\phi(t)$.

$$j(t) = \frac{\phi(t)}{2\pi f_c} \tag{4.22}$$

and cycle jitter $\sigma(t)$ (Jc in the list at the beginning of this section) can be computed by the root mean square of frequency error:

$$\begin{aligned}\sigma_t = \sqrt{\frac{1}{T}\int_0^T j^2(t)dt} &= \sqrt{\frac{1}{T}\int_0^T \left(\frac{\phi(t)}{2\pi f_c}\right)^2 dt} \\ &= \frac{1}{2\pi f_c}\sqrt{\frac{1}{T}\int_0^T \phi^2(t)dt} = \frac{1}{2\pi f_c}\sqrt{\int_{f_1}^{f_2} S_\phi(f_m)df_m} \\ &= \frac{1}{2\pi f_c}\sqrt{2\int_{f_1}^{f_2} \mathscr{L}(f_m)df_m}\end{aligned} \tag{4.23}$$

This derivation is taken from [79], wherein the author presumably uses (4.2) and asserts a change of variables to obtain equivalence on the second line above.

The last step assumes that the small angle approximation (Equation 4.5) is valid and therefore we can say $\mathscr{L} = S_\phi/2$. When $\mathscr{L}$ is in log units of dBc/Hz (as is the case for most test equipment) this expression also takes the form of

$$\sigma_t = \frac{1}{2\pi f_c}\sqrt{2\int_{f1}^{f2} 10^{\frac{\mathscr{L}(f_m)}{10}} df_m}. \tag{4.24}$$

This is the form favored by various jitter calculators and industry application notes [95, 96, 97, 98, 99].

Chapter 5

CMOS oscillators to satisfy 802.15.4 and Bluetooth LE PHY specifications without a crystal reference

This chapter contains performance of various relaxation oscillators, evaluated in the context of two low-power RF communication standards and is based on my CCWC 2019 paper [15]. In contrast to standard deviations measured during the preliminary work in Chapter 3, the Allan Deviation and Time Deviation measurements here provide a clearer evaluation of oscillator performance over time.

It was gratifying to have some agreement between hand calculation, simulation, and measurement of jitter for the 32kHz oscillator under test − if only we could say the same for ring oscillators operating at RF speeds.

5.1 Abstract

We compare measured CMOS relaxation oscillator performance against physical-layer specifications defined by low-power RF communication standards IEEE 802.15.4 and Bluetooth Low-Energy with the aim of satisfying those specifications without a crystal oscillator. The time and frequency aspects of these FSK-based specifications concern RF channel, modulation frequency, data rate, and sleep time. If they can be satisfied without a crystal oscillator, future wireless ICs can be designed that do not rely on a crystal reference and still perform standards-compatible communication. We find most physical-layer specifications can be satisfied by relaxation oscillators with the exception of RF carrier accuracy. The RF ring oscillator under test is dominated by flicker noise, which sets an Allan variance/jitter floor about an order of magnitude noisier than is required by current specifications.

Table 5.1: Oscillator-derived time specifications for common low-power radio standards. All values are ±.

Source of time	IEEE 802.15.4 [49] (O-QPSK PHY) mean μ	IEEE 802.15.4 [49] (O-QPSK PHY) stddev σ	Bluetooth LE [51] (uncoded PHY) mean μ	Bluetooth LE [51] (uncoded PHY) stddev σ
Channel	40ppm total $\mu + \sigma$		42ppm	20ppm over packet and 400Hz/μs
Data rate	40ppm total $\mu + \sigma$		50ppm total $\mu + \sigma$	
Sleep timer	N/A	30ppm [100]	N/A	500ppm

5.2 Introduction

Communication systems based on frequency shift keying (FSK), like IEEE 802.15.4 O-QPSK/MSK PHY [49] or Bluetooth Low-Energy [51], have physical layers defined by four senses of time. The published specification defines accuracy requirements, error bounds, and how to map information to these senses of time.

These four senses of time are illustrated in Figure 5.1: channel frequency, modulation frequency, data rate, and timer. Most of these values have some mean μ and standard deviation (jitter) σ. The exceptions are:

- Modulation frequency, which can lack σ if it is, for instance, controlled by fixed capacitors with no significant time-dependent random error. In our implementation, this quantity is not derived from an oscillator and so is not affected by using CMOS frequency sources instead of a crystal reference. We will ignore it for the remainder of this paper.
- Timer, which has μ but we don't care because its average duration is adjustable dependent on schedule. Timer σ is the important quantity because that determines receiver guard time, which determines average system energy.

The μ and σ specified for 802.15.4 and BLE are compiled in Table 5.1 to the extent possible. Some values are specified in absolute terms (Hz) but we have converted them to ppm for consistency. Some specification documents only define aggregate error, such as with data rate, or don't define timer jitter, such as with 802.15.4. In that case, we assume a jitter target to obtain 1.3ms guard time as described in existing published implementations [100].

Interpreting BLE carrier drift spec

Taken literally, BLE's carrier drift (accumulated jitter) requirement of $400Hz/\mu s$ is 0.17ppm: two orders more restrictive than a typical crystal. It is unlikely the drift is measured per

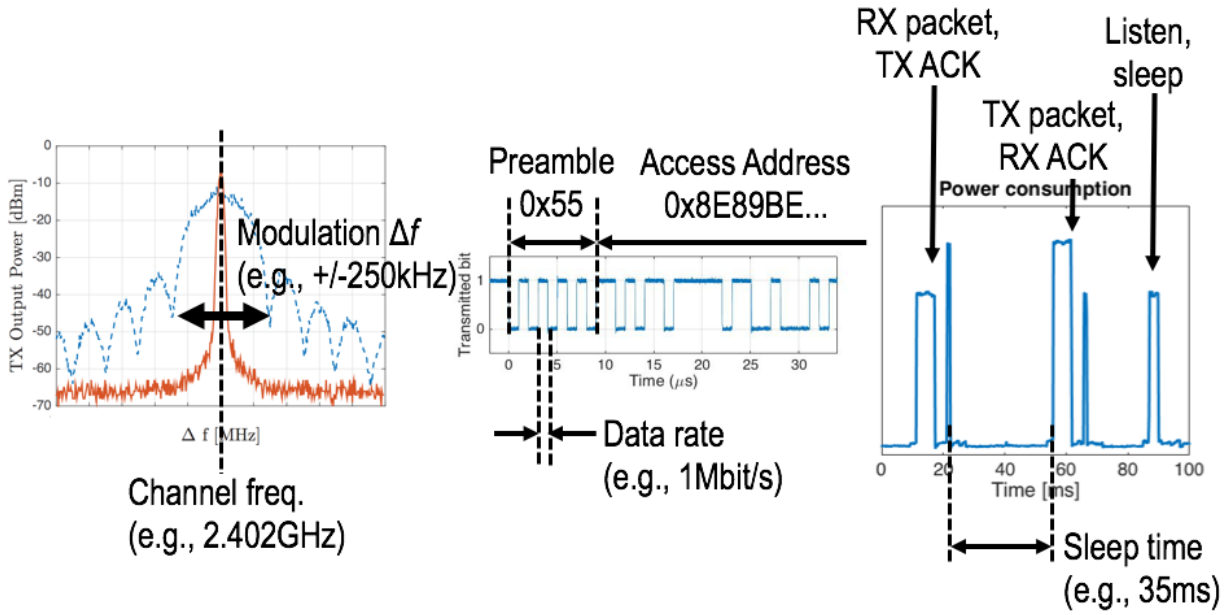


Figure 5.1: The four senses of time defining the physical layer of a FSK radio: (a) channel or center frequency (evident in unmodulated red trace) and modulation frequency Δf to define average frequency and denote 1 or 0 by transmitting faster or slower than average (evident in modulated blue trace; null-to-null spacing of modulated waveform is $1.5\Delta f$), (b) data rate ("chip rate" if data is coded, as in 802.15.4, or "bit rate" if uncoded, as in some portions of BLE) to define how often the modulation frequency will change, and (c) sleep timer to define when to wake up and, possibly, communicate.

microsecond but information on the practical evaluation of this specification is sparse. Most test equipment manufacturers state their capability to make BLE carrier drift measurements but lack detail on how the test is performed. For traditional Bluetooth, two manufacturers describe comparing average frequency of adjacent 10-bit sequences, which means a maximum accumulated drift of 4kHz over $10\mu s$ or 1.7ppm [101, 102]. Only one equipment manufacturer describes their BLE-specific tests; they evaluate carrier frequency every $50\mu s$ in the packet payload [103]. The maximum accumulated drift during that test is 20kHz or 8.3ppm, which is closer to a crystal's tolerance. If instead we use the spec defining the maximum frequency drift over an entire $376\mu s$ packet, 50kHz, the resultant 21ppm error tolerance is within typical crystal timing capabilities.

These four ways of interpreting carrier drift are summarized in Table 5.2, marked in Figure 5.2 and the least restrictive is marked in Figure 5.4. Existing BLE transceiver design literature [104] describes drift requirements as somewhat between our third and fourth interpretation, but lacks a reference for such a conclusion.

Table 5.2: Four ways of interpreting the Bluetooth Low-Energy carrier drift specification

No.	Error	Description
1	$400Hz/\mu s$ (0.17ppm)	As printed in [51]
2	$4kHz/10\mu s$ (1.7ppm)	Adjacent 10-bit sequences
3	$20kHz/50\mu s$ (8.3ppm)	Payload sequences only
4	$50kHz/376\mu s$ (21ppm)	Whole packet

5.3 Comparing Measurements to Specifications

The following sections contain measurement results of various CMOS oscillators, the performance of which is compared to wireless specifications a particular oscillator might be used to satisfy. A specification's mean μ is defined by circuit design decisions (predominantly DAC tuning resolution, unless dithering is used) and Allan variance $\sigma_y(\tau)$ plots are used to describe edge uncertainty (jitter, σ) in fractional units after averaging for a particular amount of time. Note that, for some oscillators, error starts *increasing* at long time scales. The typical instinct to average more samples to get more accuracy is invalid in all oscillators after enough time, when higher-ordered noise contributors start to dominate.

Most Allan variance figures in this paper include multiple overlapping traces. These overlapping data sets were taken to overcome sampling speed and memory size limitations of the Agilent 53230A frequency counter with 6GHz input and OCXO timebase options. Plots were generated in Timelab [91] and typically take the form of Modified Allan Variance. The Modified is similar to traditional Allan Variance but calculates variance over overlapping windows instead of adjacent windows. In the past, it was not used due to computational demands required to calculate many overlapping windows. Today, it is the generally-accepted standard method of determining relative oscillator period drift over time.

All data was taken in an indoor open-air lab benchtop environment (i.e., without a temperature chamber). All oscillators have some sensitivity to temperature – and supply voltage, age, etc. – changes, compensation for which is beyond the scope of this paper.

Crystal reference

We start with the highest-performing solution: a 10MHz oven-controlled crystal oscillator (OCXO), a class of oscillators known for sub-ppb accuracy. The Allan deviation for this crystal is plotted in Figure 5.2. All the specifications described in Table 5.1 are 30ppm or greater with the exception of BLE's carrier drift, and a line in Figure 5.2 at 30ppm indicates a crystal will meet or exceed those specs, which is expected.

As for BLE carrier drift, we plot the four ways of interpreting the drift spec as described in Section 5.2: 0.17ppm at $1\mu s$ intervals, 1.7ppm at $10\mu s$, 8.3ppm at $50\mu s$, and 21ppm at $376\mu s$ – as stars in Figure 5.2. The first and second stars are unlikely ways to interpret the drift spec.

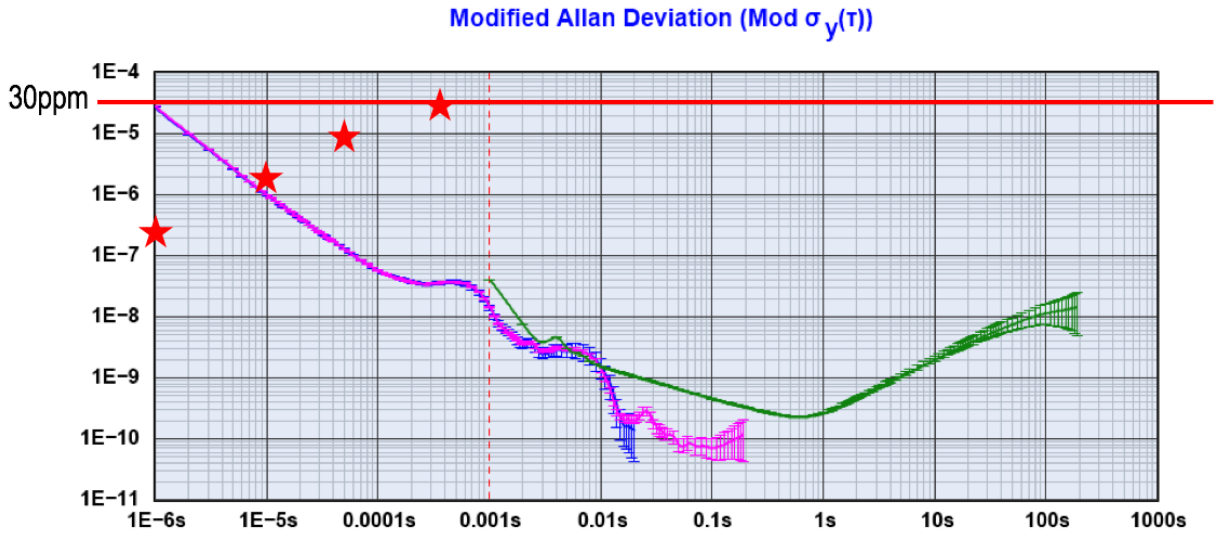


Figure 5.2: Modified Allan Deviation for a temperature-compensated crystal oscillator. Line at 30ppm denotes lower limit of most specifications from Table 5.1. Stars indicate four ways of interpreting BLE carrier drift described in Section 5.2. The leftmost traces in blue and purple, terminating near 10ms and 100ms respectively, are from frequency measurements every $1\mu s$. Green rightmost trace terminating near 100s is from measurements every 1ms.

If we assume this crystal is the reference oscillator in a PLL, the RF VCO locked to it will have the same noise profile at times longer than the loop bandwidth. Assuming a loop bandwidth of 1/10th of the 10MHz crystal or a $1\mu s$ period, all four stars are at periods equal to or greater than that period. Furthermore, the third and fourth stars are the mostly likely ways to interpret BLE drift spec and lie well above the Allan deviation line. Assuming negligible additional jitter contributions by the phase detector, loop filter, and divider, we therefore predict the VCO will meet the RF carrier drift spec and hence a crystal-based RF system passes all specifications, as expected.

CPU clock

Ring oscillators are small, low power, and often included in SoCs to provide digital clock sources. We fabricated one such oscillator: a current-starved 4-stage differential ring, with current DAC to adjust speed. It consumes $1.5\mu A$ at 20MHz in extracted simulation. The real current draw is too small to measure against our SoC's baseline. This oscillator's measured Allan deviation is plotted in Figure 5.3 with a line at the most relaxed specification: 500ppm for BLE's timer. While this clock is fine for operating a CPU, it is clearly not suitable for RF timing.

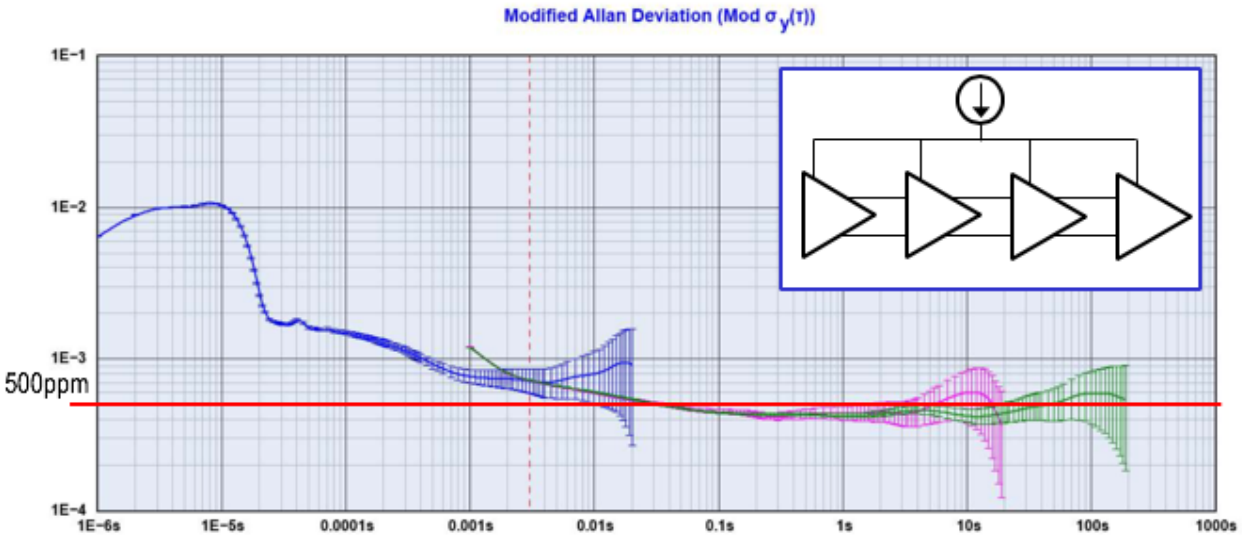


Figure 5.3: Modified Allan Deviation for a low-power CPU clock. Line at 500ppm corresponds to the least restrictive timing spec from Table 5.1; clearly, this class of clock is inappropriate for radio timing. The leftmost trace in blue is from frequency measurements every $1\mu s$; the middle and rightmost traces in purple and green, respectively, are from measurements every 1ms.

Free-running RF ring oscillator

Ring oscillators running at RF speeds often serve as the adjustable oscillator in a PLL-based frequency synthesizer. The PLL uses a crystal oscillator as a more pure frequency reference and thereby turns an ordinarily-noisy RF oscillator into something that can meet radio spec. This traditional technique is successfully used in commercial products and, more importantly, depends on an external crystal, so we consider a free-running RF oscillator in this section. The oscillator under test is an 8-stage differential current-starved ring with kick-start circuit to prevent latching. The schematic for each stage, or delay cell, is shown in the inset of Figure 5.4. It consumes 1.25mW and was tuned to 2.44GHz manually prior to measurement.

Measured results are shown in Figure 5.4. The results miss most specifications but are close enough to motivate refining future designs. We see flatness (τ^0 trend) across the majority of the plot indicating flicker noise dominates in the range of useful time metrics, as discussed in Chapter 4. The white noise-dominant region, which would follow a $\tau^{-1/2}$ trend, likely lies at time scales shorter than $1\mu s$ – out of the range of our instruments.

Data clock w/ 2MHz RC oscillator

A frequency synthesizer can potentially supply all necessary frequency sources but, by running crystal-free, we don't have that option. Instead, we designed a 2MHz RC oscillator to generate

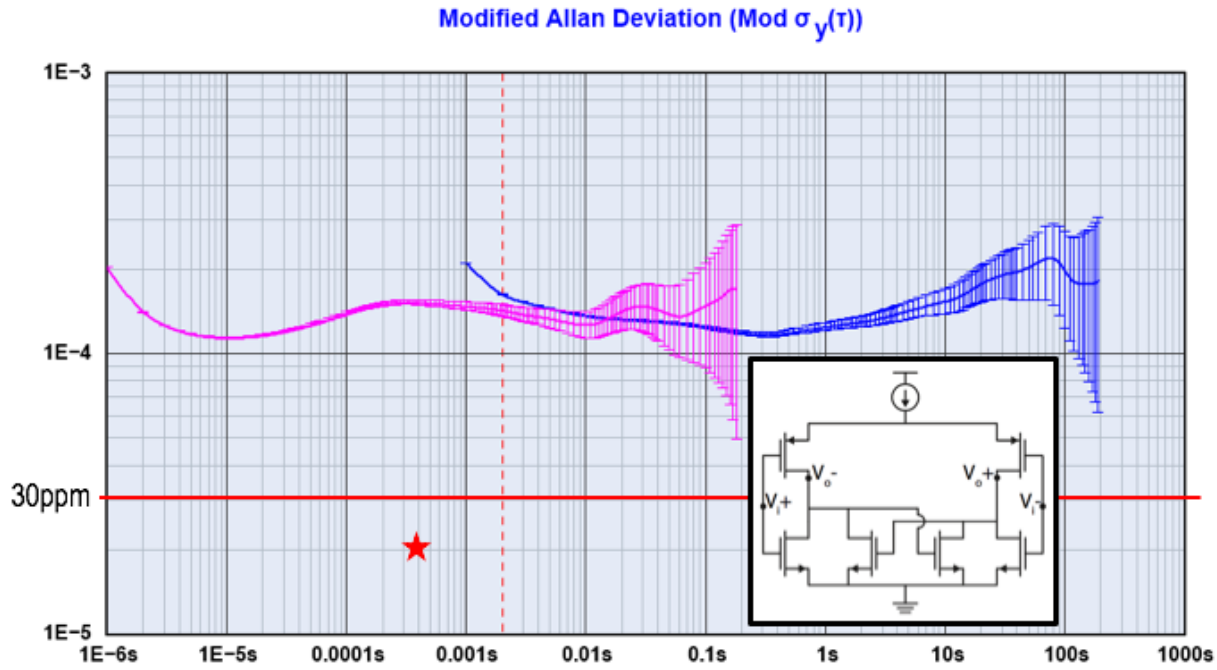


Figure 5.4: Modified Allan Deviation for a free-running 2.44GHz ring oscillator. Line at 30ppm denotes lower limit of most specifications from Table 5.1. Star indicates the most relaxed, highest-drift, of the four ways of interpreting BLE carrier drift described in Section 5.2: 21ppm at 376μs. Leftmost purple trace is from frequency measurements every 1μs. Rightmost blue trace is from measurements every 1ms. Inset: schematic of individual delay cell; eight are linked together to form oscillator.

the 802.15.4 data clock directly and BLE data clock via a fixed divide-by-2 block. It was based on the design in [57] but without the flipped replica inverter regulator to reduce complexity. It consumes 2μA, which could be a significant savings compared to deriving this frequency source from a synthesizer. This is particularly useful because the power needed to generate this clock is drawn simultaneously with LO generation and PA power during transmission, thus reducing minimum peak power. Its adjustable resistor DAC was designed to have 27ppm tuning resolution.

Its Allan deviation is plotted is Figure 5.5 with markers indicating the oscillator barely exceeds jitter specifications for 802.15.4 (0.5μs chip period) and BLE (1μs bit period). While jitter alone meets specifications, 27ppm tuning resolution means the starting frequency could be at most ±13.5ppm off. That initial static error, combined with measured jitter, may exceed 40 or 50ppm and therefore miss specifications. We are confident a simple redesign to increase tuning resolution with a finer resistive DAC, and/or sizing inverters to consume more power and thereby contribute less jitter, would meet specification.

Practically, however, off-the-shelf parts are more generous than published specifications.

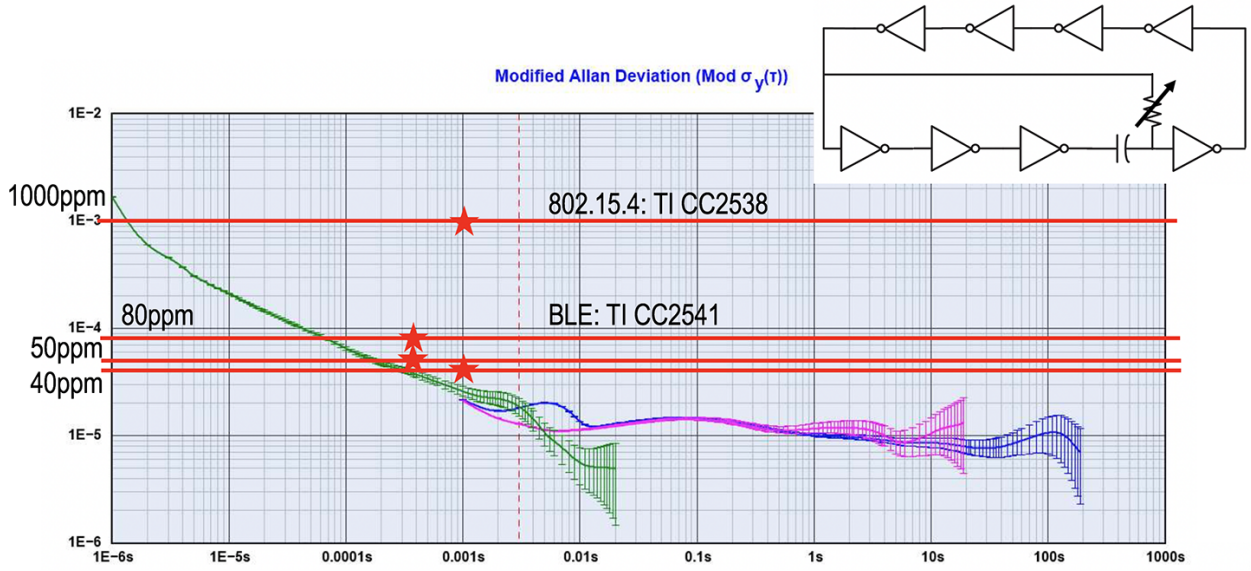


Figure 5.5: Modified Allan Deviation for a 2MHz RC oscillator based on [57]. Lines at 40 and 50ppm indicate specifications for 802.15.4 and BLE, respectively, with stars marking their respective chip/bit periods. Lines at 80ppm and 1000ppm indicate actual data sheet requirements for Texas Instruments CC2541 (BLE) and CC2538 (802.15.4), respectively [105, 106], with stars marking corresponding chip/bit periods. Leftmost trace in green is from frequency measurements every $1\mu s$. Purple and blue rightmost traces are from measurements every 1ms.

Included in Figure 5.5 are lines indicated real-world receiver datasheet limits, which this oscillator meets.

Sleep timer w/ 32kHz RC oscillator

The 2MHz oscillator in Section 5.3 would suffice for sleep timing as well, but sleep timer requirements are much more lax than a data clock so we know we have an opportunity to save power. A re-designed oscillator based on [57] was implemented for this purpose, including the additional LDO stage as in the source design, consuming simulated 350nA. The real current draw is again too small to measure against our SoC's baseline. It was targeted for 32kHz operation but lacks any tuning because the true center frequency does not matter; after calibration step to measure its frequency, e.g., receiving two packets with known interstitial time as described in Sec. III of [11], a timer can be set with an appropriate number of ticks to achieve a given sleep time.

The measured jitter of this oscillator is plotted in Figure 5.6 with relevant sleep timer specifications indicated. Note: Allan Deviation was used instead of Modified Allan Deviation in this plot to better visually identify overlap between 1000s and 10000s measurements.

Modified Allan tends to generate exaggerates upward "curls" at low τ values for reasons as-yet-unknown, where "low" is 5-10 τ so the exact value depends on the sampling rate of the data set. Traditional Allan Deviations, plotted here, instead exhibit straight lines along a $\tau^{-1/2}$ trend at low τ values.

Because of how this timer is used, it makes more sense to plot time deviation: instead of fractional error after a given time (Allan Deviation $\sigma_y(\tau)$), we are more interested in absolute error in seconds after a given time (Time Deviation $\sigma_x(\tau)$). This method of interpreting our results is plotted in Figure 5.7. Both specifications target approximately 1ms guard time. In BLE, this goal is mentioned in Section 4.2.2, "Sleep Clock Accuracy," of [51] in describing the "anchor point" for the timing of a packet exchange event. There is no fixed guard time specified in 802.15.4. The 802.15.4e amendment specifies the packet guard time *macTsRxWait* to a default of ±1.1ms and acknowledgement guard time *macTsAck* of ±0.2ms [107]. A practical implementation of OpenWSN uses ±1.3ms for the packet and ±0.5ms for the acknowledgement [100].

The plotted time deviation indicates 1ms jitter will result after waiting for about 105s. Given error bars, and the fact that jitter only describes error magnitude of only 1 standard deviation, a system using this sleep timer would likely be designed to sleep no longer than 20s to ensure reliable communication with a 1ms guard time.

Comparing this oscillator's performance to commercial 32kHz quartz or silicon resonator products is difficult. Random accumulated error over time is an important feature in high-accuracy temperature-compensated crystal oscillators (TCXOs) or oven-controlled crystal oscillators (OCXOs) so their datasheets include stability information (as opposed to information about environmental or supply sensitivity). Instead, we want to compare oscillators in this work to their low-cost equivalents found in Internet-of-Things devices but low-cost crystal datasheets seldom include Allan Variance/Deviation plots, jitter after time, or other information needed to reconstruct performance and compare. Often, "stability" in these low-cost crystal datasheets refers to how the mean frequency shifts over the specified temperature range rather than random accumulated error over time at fixed temperature and voltage conditions.

One exception was found that specified 35ns period jitter [108]. A 32.768kHz oscillator has period of $30.5\mu s$ so 35ns jitter represents 1147ppm. If we assume this oscillator accumulates frequency error (i.e., its Time Deviation increases) only through white noise, about 30s of accumulation should increase 35ns of error at 1 period to $35ns * \sqrt{30.5\mu s/30s} = 5\mu s$. The RC oscillator described here reaches that amount of error after about 6s, which compares favorably but underscores the advantages of using resonant oscillators. Other advantages may include better temperature and supply sensitivity to reduce the rate at which the oscillator must be recalibrated to meet network guard time requirements, but these downsides may be compensated for through network-level solutions [13].

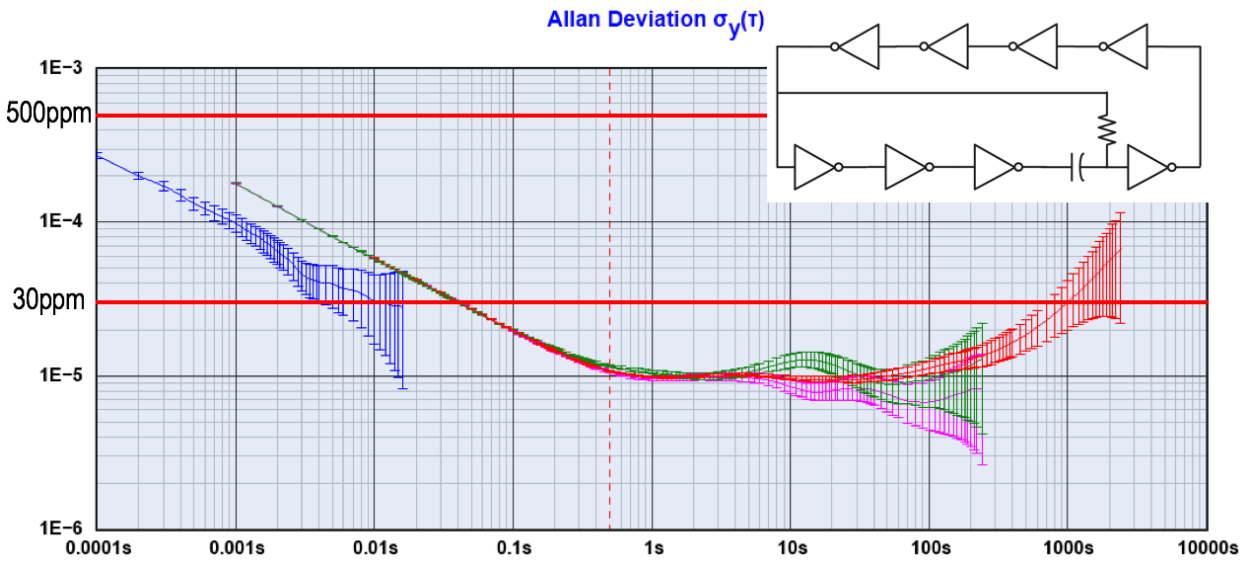


Figure 5.6: Allan Deviation for a 32kHz RC oscillator based on [57]. Lines at 30ppm and 500ppm indicate specifications from Table 5.1. Green and pink traces are from repeated 1000s lab measurements. Red trace is from lab measurement lasting 10000s; over this time period, temperature change in the room is significant enough to manifest as random walk error and causes error to grow after approx. 100s. Blue trace is from 64ms of transient noise simulation showing good, but optimistic, agreement with measurement.

5.4 Summary of Results

A summary of specifications, relaxation oscillators designed to meet to them, and whether the given oscillator meets the specification, is given in Table 5.3. Data clock and sleep timer are met, whereas the implemented RF ring oscillator is dominated by flicker noise as indicated by the relative flatness of the Allan deviation. The measured noise floor is and is at least 70ppm beyond the specification goal. Because flicker dominates the behavior of this oscillator, the instinct to spend more power to reduce white noise doesn't apply and may exacerbate observed noise levels via increased flicker noise.

Though the RF oscillator's performance misses the specifications examined in this paper, existing work on communication with free-running ring oscillators indicates relaxing IEEE 802.15.4 modulation frequency specifications from ±0.5MHz FSK to ±1MHz FSK would allow 802.15.4-quality communication (< 1% PER) [14].

5.5 32kHz sleep timer jitter calculation and simulation

The 32kHz timer oscillator is straightforward to analyze by hand, and it is the most computationally tractable to simulate by virtue of its low speed, so we use it to evaluate to what

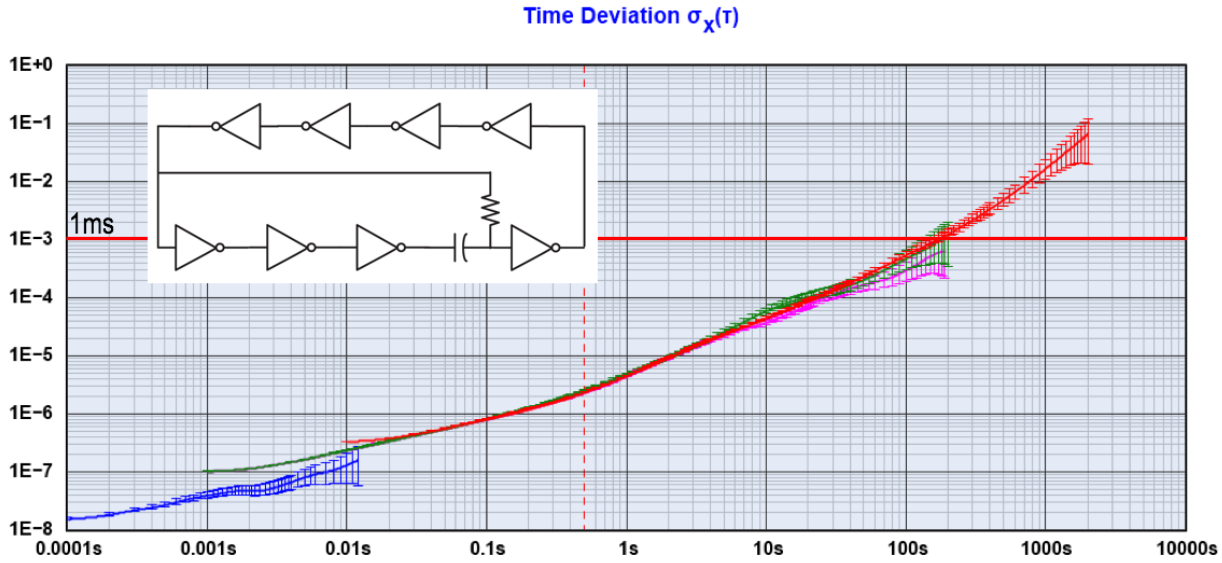


Figure 5.7: Time deviation for a 32kHz RC oscillator based on [57]. Line at 1ms indicates reasonable wakeup time error goal, or guard time, for BLE and 802.15.4 specifications. Green and pink traces are from repeated 1000s lab measurements. Red trace is from lab measurement lasting 10000s. Blue trace is from 64ms of transient noise simulation showing good, but optimistic, agreement with measurement.

Table 5.3: Summary of specification and oscillator performance comparisons

Spec type	Oscillator	Specification summary	Meets?
Channel	Ring, Sec. 5.3	40ppm $\mu + \sigma$ (802.15.4) 42ppm μ+ Sec. 5.2 (BLE)	N
Data clock	RC, Sec. 5.3	40ppm $\mu + \sigma$ (802.15.4) 50pm $\mu + \sigma$ (BLE)	Y
Sleep Timer	Sub-Thresh. RC, Sec. 5.3	1.3ms [100] σ (802.15.4) 1ms σ (BLE)	Y

extent theory and measurement agree in this experiment. Jitter was evaluated four different ways, the results of which are compiled in Table 5.4 and described below.

Hand calculations

To estimate jitter by hand, we first recognize this oscillator's performance is dominated by the relaxation of the large static R and C passives. A jitter estimate can be performed by comparing voltage noise at the RC node with slew rate at the time of transition. If we assume noise is dominated by the integrated noise on the large 10pF capacitor, $\overline{v_n^2} = kT/C = 4 * 10^{-21}/10^{-12}$ so $v_n = 20\mu V$. The RC node swings with amplitude $2V_{DD}$ and transitions at approximately $V_{DD}/2$, so we say the RC decay is about 2/3 to its final value before switching or lasts $1\tau = 1/RC$. One RC decay is about $1.5V_{DD}$ so we can say $dv/dt = 1.5 * 1/\tau * V_{DD} * 1/e = 19.7mV/\mu s$. This estimate matches well with simulation. Hence, 1σ of dt uncertainty at the RC is about 1ns, or 36ppm.

The devices in this oscillator are deep submicron transistors. Furthermore, the additional layer of voltage regulation sets their local V_{DD} to approximately 500mV, meaning the transistors are biased in sub-threshold at the critical transition voltage. Therefore we must rely on simulation to estimate gain of an inverter with input at $V_{DD}/2$: about 11V/V. This RC oscillator is effectively a single-ended ring oscillator with a dominant noise source in the middle. We know single-ended ring jitter can be estimated by calculating the jitter of a single stage [68]. Hence, we can conclude expected jitter in this rough approximation to be kT/C noise times gain of one stage: about 11ns or 360ppm.

Simulation

We performed a transient noise simulation of this oscillator, along with transistor-level schematics of current and voltage sources supplying it, for 64ms with noise bandwidth 1Hz to 100MHz. Period measurements of results have standard deviation of 14.5ns 464ppm. Period simulations were averaged into 0.1ms groups and plotted alongside measurements in Figure 5.6 and Figure 5.7. We can again choose a point in the white noise region of the time deviation plot and scale by the square root of the period ratio. Using this method, simulated jitter is estimated to be a more optimistic 7.1ns or 226ppm.

Measurement extrapolation

We can estimate real period jitter by extrapolating our time deviation data: starting from a point in the white noise region of the plot in Figure 5.7 (e.g., $0.8\mu s$ error at 0.1s period) and scaling by the square root of the period ratio, measurement period jitter is estimated to be 14.1ns or 452ppm. This is somewhat noisier than hand-calculation and about twice as noisy as simulation when comparing results extrapolated from time deviation plots. Simulated and measured data are plotted in Figs. 5.6 and 5.7 where the similarities are evident, despite the comparatively low amount of simulated data available.

Table 5.4: Comparison of 32kHz oscillator jitter determined through hand calculation, simulation, and measurement

Source	Method	Period Jitter (ns)	Period Jitter (ppm)
Calc	Voltage noise on RC node	11	360
Sim	Tran noise stddev	14.5	464
Sim	Time dev. extrapolation	7.1	226
Meas	Time dev. extrapolation	14.1	452

5.6 Conclusion

We have described timing specifications of the physical layer of two major FSK-based radio standards – IEEE 802.15.4 and Bluetooth Low-Energy – and compared measurements of various CMOS oscillators against aspects of those standards. Our goal was to determine to what extent current radio standards can be satisfied without off-chip crystal references. Sleep timing and data clock requirements can be satisfied with RC oscillators. Frequency shifts to transmit FSK data can be accomplished by, for instance, switching capacitors in and out of the oscillator, which needs no crystal even in a traditional system. RF ring oscillator results miss specifications by about an order of magnitude but we have hope for future designs if flicker noise dominance issues can be avoided. Finally, we presented an example to illustrate the comparison between theory, simulation, and measurement in relaxation oscillators which shows promising simulator results but still underscores the importance of fabrication and measurement before oscillator results can be relied upon.

If RF communication systems can relax their reliance on external reference oscillators, it is possible to design a wireless sensor mote with zero external components resulting in drastic reductions in size, power consumption, and cost. These gains can be realized while maintaining advantages of standards-compatible communications like peer-to-peer communications without a high-fidelity base station (as in, e.g., RFID) and interoperation with billions of COTS devices.

Chapter 6

Free-running 2.4GHz ring oscillator-based FSK TX/RX for ultra-small form factors

An ASIC containing a re-designed ring oscillator-based transceiver, microprocessor, sensor ADC, and other accessories is tested with respect to communication in this chapter. As mentioned in Chapter 3, work in this chapter revealed sensitivity is poor at narrow bandwidths. This chapter drops the 802.15.4 receive goal and trades tone spacing/bandwidth for substantial bit/chip error rate benefits to maximize sensitivity of ring-to-ring communication.

6.1 Abstract

We demonstrate the first communication between radios using free-running 2.4GHz ring oscillators requiring neither a frequency synthesizer nor any external components (e.g., crystal oscillator). Other work, displayed in Table 6.1 demonstrates free-running ring transmitters and receivers as individual components, but none demonstrate performance of two devices communicating with each other. After a single calibration and in an open lab environment, a free-running low-IF ring RX is able to demodulate data sent by a free-running FSK ring TX (0/1 tone separation 5.5MHz) with input power as low as −67dBm (yielding estimated 5.5m range) to better than 6.5% chip error rate (CER) / 1% packet error rate (PER). Demodulation compares zero-crossing period measurements to an IF windowed average that tracks the mixed TX and RX ring oscillators' significant accumulated jitter over time. By solely relying on ring oscillators for 2.4GHz tone generation, this work enables 600x chip area reduction when comparing a $10\mu m$ x $10\mu m$ ring oscillator (oscillator only, no current DAC) with a $250\mu m$ x $250\mu m$ 4.7nH LC tank (inductor only, no capacitor DAC) as illustrated in Figure 6.1. This work also enables 4x power reduction when comparing $105\mu A$ @ 1.0V ($105\mu W$) ring oscillator power vs. $500\mu A$ @ 0.8V ($400\mu W$) LC tank power. Note: both oscillators have the opportunity to reduce their power consumption by reducing their voltage swing to the

minimum necessary for good mixer performance, as described in more detail in Chapter 3 of [6].

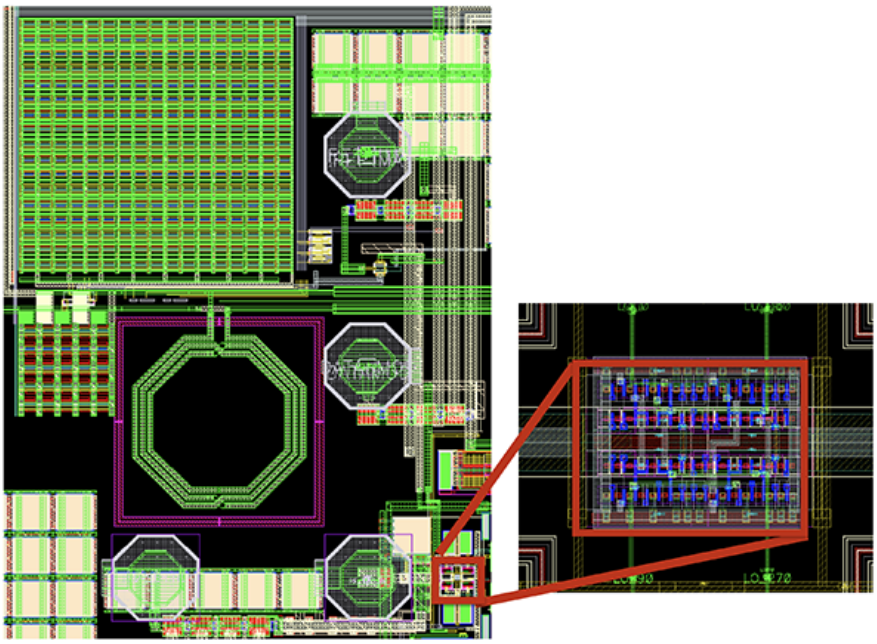


Figure 6.1: Area comparisons of LC tank oscillator (courtesy Fil Maksimovic) and ring oscillator in IC layout. The 4.7nH LC tank inductor (green octagon, center left) measures $250\mu m$ x $250\mu m$. The 2.4GHz ring oscillator core (inset) measures $10\mu m$ x $10\mu m$.

6.2 Introduction

RF ring oscillators have the advantage of very small area and low power but are typically part of a large, power-hungry frequency synthesizer requiring an external crystal reference which negates the ring oscillator's advantages. This work demonstrates a 2.4GHz FSK transmitter and receiver operable without external crystal references and free from inductor-based oscillators, using only free-running deep submicron RF rings as local oscillator (LO). Most free-running ring oscillator work is comprised of uncertain-IF on/off keyed (OOK) wake-up radios [63, 64, 62], or very wideband (100s of MHz) FM-UWB frequency-shift keyed (FSK) TX [65] and RX [66]. Narrowband (<100MHz) free-running ring oscillator-based RF transceivers—capable of communicating with each other (i.e., "peer-to-peer") instead of with a high-performance base station—have yet to be demonstrated. By leveraging ring oscillator size and power advantages, systems incorporating these ultra-small transceivers could have total form factor small/light enough to be suitable for: an unobtrusive skin implant to communicate physiological readings over a body-area network (BAN), the instrumentation of flying insects, or being added to chemical processing to make smart materials.

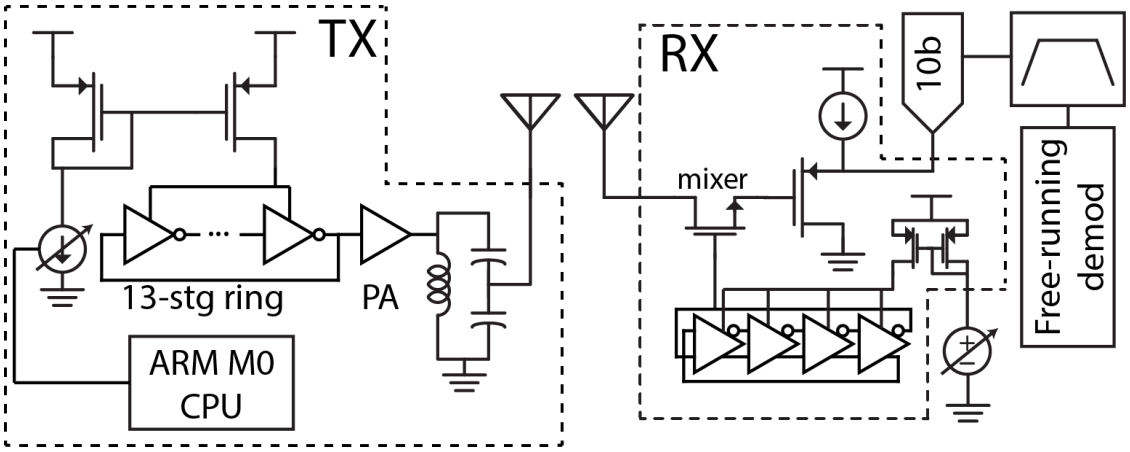


Figure 6.2: System schematic for free-running transmitter & receiver.

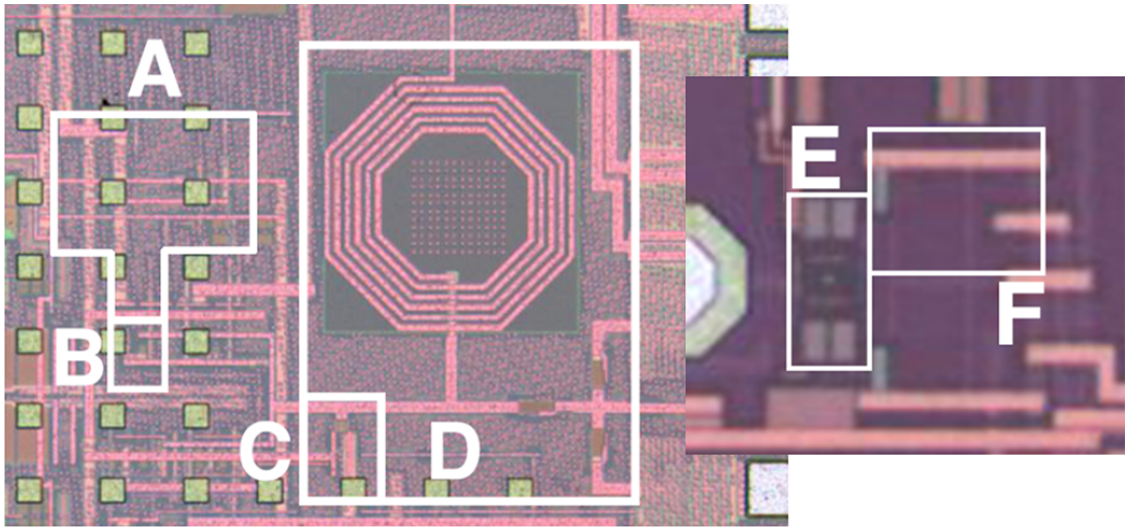


Figure 6.3: Die photo showing (A) TX oscillator current bias network, (B) oscillator/mixer, (C) PA, (D) match, (E) RX: oscillator/mixer, (F) off-chip buffer. TX/RX dice not pictured to relative scale.

6.3 Implementation

We demonstrate a 2.4GHz direct modulation transmitter and low-IF receiver using free-running ring LOs fabricated in TSMC 65nm CMOS (Figure 6.2). To compensate for high phase noise, we communicate with a 2FSK-based coded packet structure based on the IEEE 802.15.4 2.4GHz O-QPSK PHY comprised of 32-chip sequences (each representing 4-bit words) preceded by 0x5555, a Bluetooth LE-style preamble. The RF local oscillators require no external components to operate besides power/bias. Die area of relevant components are annotated below to highlight relative size (Figure 6.3).

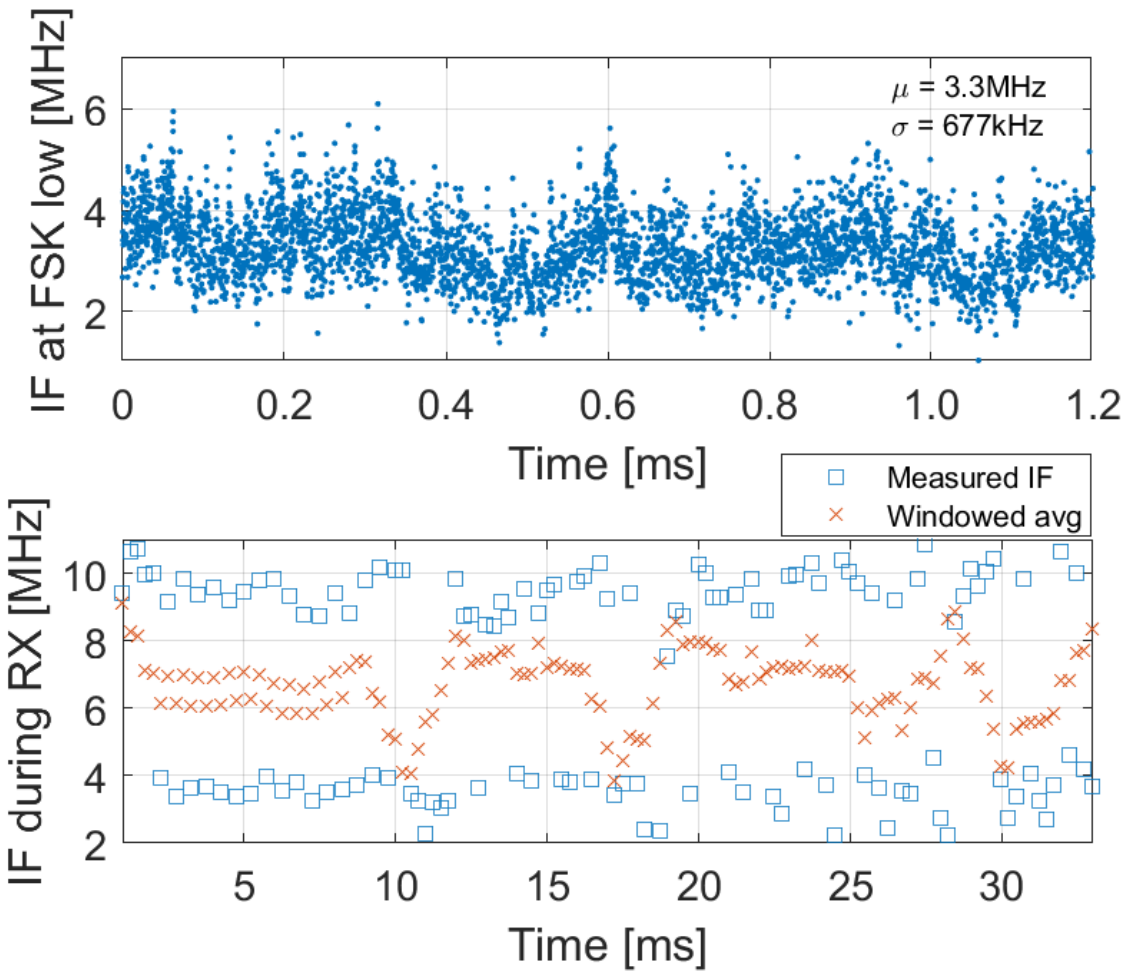


Figure 6.4: (Top) Example of unmodulated IF over 1.2ms. (Bottom) Modulated frequency measurements with moving average between.

Transmitter

The transmitter uses a 2.4GHz single-ended, 13-stage current-starved ring (0.0009mm^2) consuming 364μW from 1.2V amplified by a switching power amplifier (PA, 0.0013mm^2) consuming 1.6mW from 1.2V [5] and delivering -12dBm via passive on-chip match (0.1mm^2) to an external antenna. The oscillator is directly FSK modulated via its current DAC (0.053mm^2), controlled by on-chip ARM Cortex M0 (1.23mm^2 incl. 128kB SRAM). Ring oscillators exhibit significant phase noise/jitter (Figure 6.4) so FSK tone spacing was $\tilde{5}$.5MHz such that the demodulator could distinguish between noisy FSK tones (Figure 6.5). The bit period was 250μs, located at the knee between white & flicker noise in the IF Allan variance (Figure 6.6) to minimize jitter.

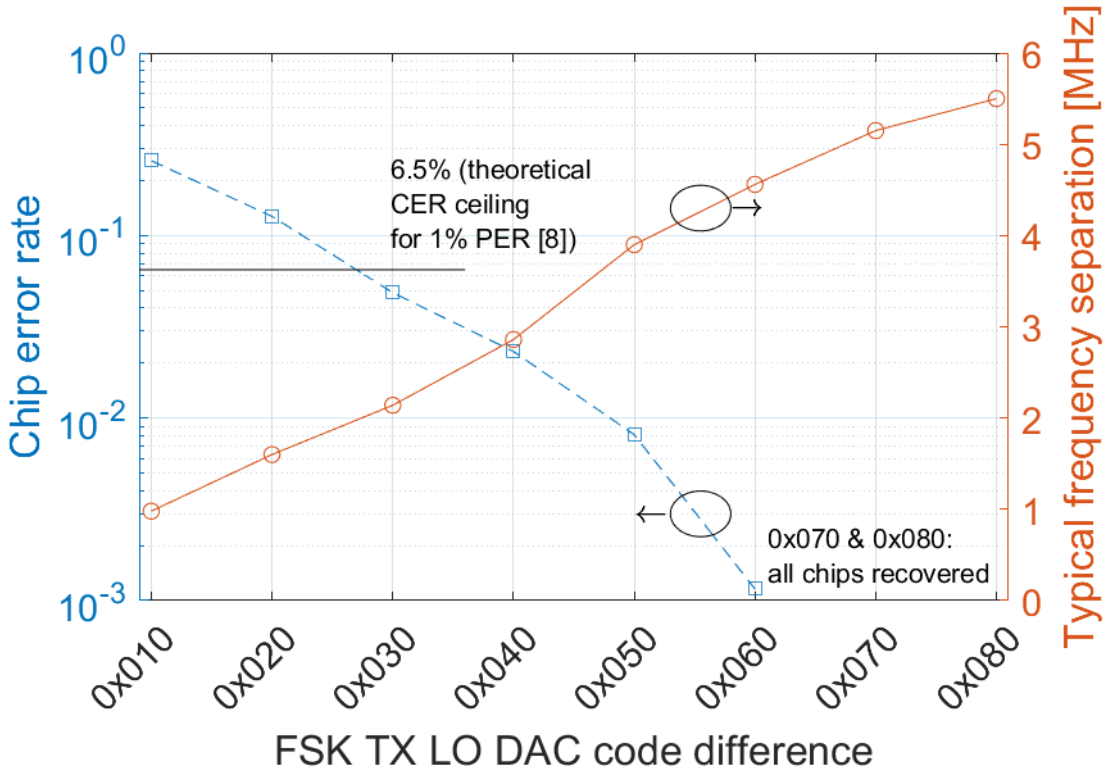


Figure 6.5: Chip error rate, under high SNR to expose phase noise effects, and FSK frequency separation vs. TX DAC code separation.

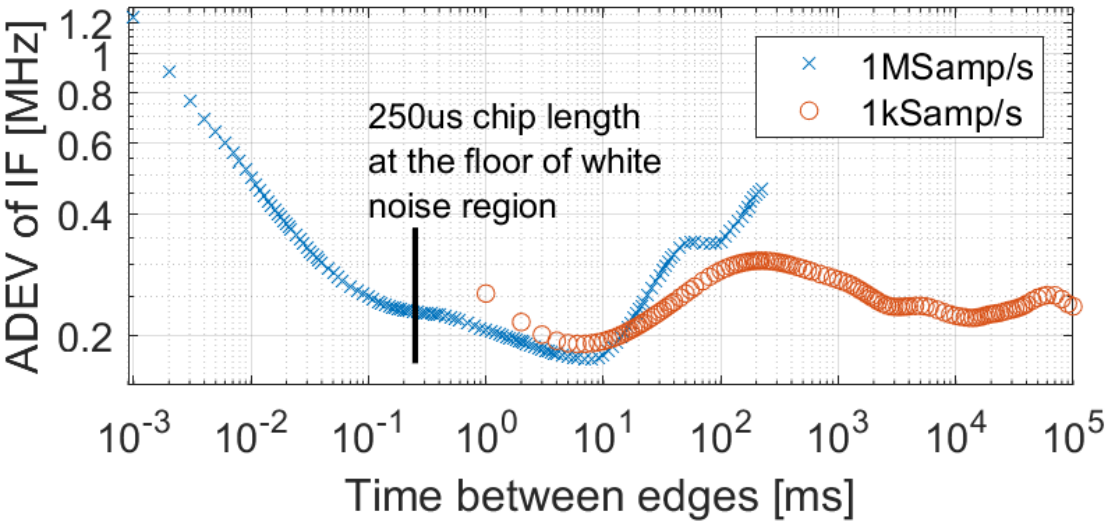


Figure 6.6: Allan deviation $\sigma_y(t)$ at IF resultant from free-running TX mixed with free-running RX.

Receiver

The receiver uses a 2.4GHz differential, 4-stage current-starved ring oscillator (0.0006mm^2) consuming 105μW from 1V with current mirror tuned via off-chip voltage DAC [14]. To enable more flexible experimentation, RF input is fed directly to the passive NMOS mixer with neither passive match nor low-noise amplifier (LNA) at RF; frontend gain is therefore estimated to be -0.5dB. The mixer gate is driven directly by the LO. The low intermediate frequency (IF), 6MHz nominal, is buffered off-chip via PMOS source follower (0.014mm^2) for digitization and experimental demodulation. A simple 4-pole band-pass Butterworth filter with passband 2MHz to 32MHz is applied digitally before free-running demodulation.

Free-running demodulation

The high phase noise/jitter inherent in ring oscillators makes traditional demodulation techniques, e.g., matched filters, perform poorly; as the ring accumulates random error, the filter templates fail to match. Instead, we sample the IF using a 10-bit ADC at 100MS/s and measure current IF frequency by comparing zero crossing times. A moving average of recent per-bit frequency measurements (typically 30 measurements of 250μs bit periods forming a 7.5ms window) follows the IF drift over time and average frequency over the last bit period is compared with the moving average to determine binary value. Bit alignment via clock/data recovery is assumed for our purposes. The IF is acquired for data length plus 12% guard time at 18s intervals in 256-chip (32-bit) blocks. When a 32-chip 802.15.4-style preamble is recognized, we de-correlate subsequent chips in groups of 32 using 802.15.4 standard codes to recover data. Separately, we compare the recovered chip stream and with original data to calculate chip error rate (CER) (Figures 6.5 and 6.7).

6.4 Performance

TX and RX local oscillators underwent a single initial calibration to 2.45GHz before several days of measurement. Temperature was not controlled except insofar as the ambient lab environment is regulated by building controls. Transmitter output was attenuated inline and fed into receiver. All data is resultant from free-running TX mixed down by a free-running RX. Comparing AVAR @ 1.2ms in Figure 6.6 with σ=677kHz in Figure 6.4 we see AVAR jitter estimates are conservative. The cumulative distribution function (CDF) tells us we need FSK tone separation >1.03MHz (code 0x010), given σ=677kHz, for 6.5% CER (target from [9]). Experiment shows 2.1MHz (code 0x030) is the minimum necessary in high SNR conditions (Figure 6.5). In practice we widened tone spacing until the point of diminishing returns at low input powers (e.g., -66dBm sensitivity is not improved for FSK separations >5.5MHz, code 0x080), adding 3.4MHz BW for 15dB improved sensitivity. Given receiver simplicity, significant performance improvements may be realizable with modest RX improvements. We see acceptable performance (<6.5% CER, <1% packet error rate/PER) down to -67dBm (Figure 6.7), when IF amplitude falls below the noise floor of our off-chip

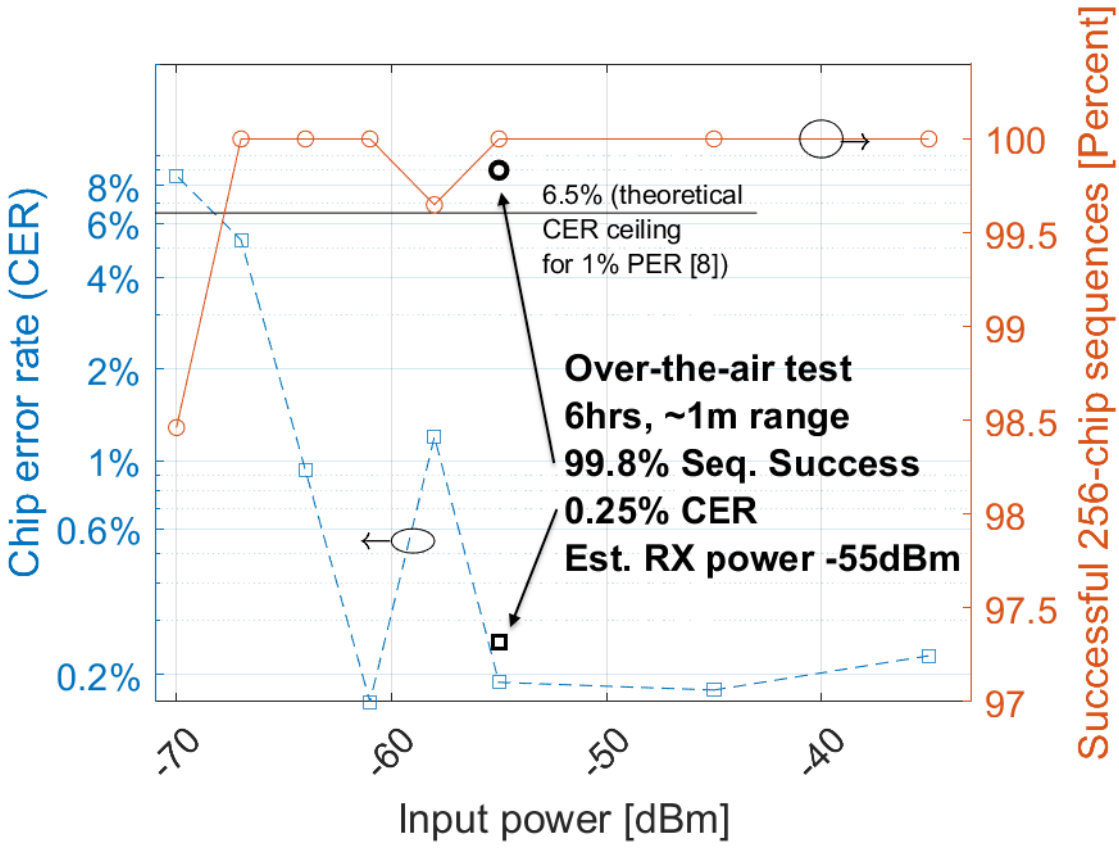


Figure 6.7: Chip error rate (CER) and percent correctly de-correlated 256-chip (32-bit) sequences (1-PER) vs. receiver input power.

ADC. This corresponds to −55dB attenuation (5.5m estimated range), demonstrating first communication between free-running deep submicron RF ring oscillators and potential for full RF transceivers to occupy <0.1mm^2 die area.

Table 6.1: Comparison to other published work

	This work	[63] Milosiu WiSNet 2015	[64] Pletcher ISSCC 2008	[62] Bryant ESSCIRC 2014	[65] Saputra JSSC 2011	[66] Kopta RFIC 2016
Frequency	2.4GHz	2.4GHz	2.0GHz	2.4GHz	4.0GHz	4.0GHz
Process	65nm	130nm	90nm	90nm	90nm	65nm
Ring Oscillator Topology	4-stage differential RX, 13-stage single-ended TX	Not reported	3-stg single	3-stg single	3-stg single	2-stg diff
Modulation	FSK	OOK	OOK	OOK	FM-UWB	FM-UWB
Data rate	4kcps (0.5kbps)	up to 33kbps	100kbps	250kbps	100kbps	100kbps
Bandwidth	30MHz	100MHz (estimated)	100MHz	54MHz	500MHz	300MHz
Oscillator Power	105μW RX 364μW TX	up to 233μW	6μW	13μW	280μW	140μW
Off-chip RF components?	None	(Unknown)	BAW RF Filter	None	Match	None
Chip-to-chip communication?	Yes, TX/RX	Wakeup RX only	Wakeup RX only	Wakeup RX only	TX only	RX only
Sensitivity	-67dBm for 6.5% CER, 1% PER	-80dBm for 1% wakeup word error	-72dBm for 10^{-3} BER	-88dBm for 10^{-3} BER	N/A (Transmitter)	-70dBm or 10^{-3} BER

Chapter 7

Demonstrated Sensor Applications

In this chapter, the aforementioned ASIC containing a re-designed ring oscillator-based transceiver, microprocessor, sensor ADC, and other accessories is combined with a CS-FET in collaboration with Hossain Fahad to demonstrate a chip-scale wireless chemical sensor system requiring no additional electronic components to operate [109]. The sensor ADC was also demonstrated with a sodium sensor from Mallika Bariya and two-terminal lactate sensor from Maggie Payne.

7.1 Two-chip Wireless H_2S Gas Sensor System Requiring Zero Additional Electronic Components

This section describes a wireless hydrogen sulfide (H_2S) gas sensor system comprised of two integrated circuits: a chemically-sensitive field effect transistor (CS-FET) sensor and a single-chip micro-mote (SCμM). The sensor IC is a bulk transistor functionalized to respond to H_2S. The SCμM IC uses an ARM Cortex M0 to digitize sensor voltage via an ADC and transmit data through a 2.4GHz FSK transmitter based on an ultra-small, crystal-free, free-running ring oscillator. The IC pair has combined volume <4mm^3, requires only a power source & antenna and no additional components, and has been demonstrated to acquire signals resulting from H_2S gas and wirelessly transmit results at 2.4GHz.

Background

Advances in fabrication techniques have allowed great reductions in sensor size. However, when a sensor is integrated into a larger system, the control, readout, and communication electronics can dwarf the sensor itself [110] putting a high floor on size (1cm x 1cm x 0.2cm or more), power consumption (>20mW), and cost (>$10). Even when commercial sensors integrate digitizing & readout electronics inside the sensor package, communication still requires a wireless module. These modules are the driver for the size/power/cost limits above

and include an RF IC, 1-2 crystal oscillators, passives, and sometimes an antenna, mounted on a PCB.

Surpassing these limitations could enable unobtrusive measurement of physiological signals [23] or wearable sensor tags to identify hazards in the environment at per-person spatial resolution. One such hazard is hydrogen sulfide gas (H_2S) with a recommended maximum exposure limit of 10ppm for 10 minutes and is immediately dangerous at concentrations above 100ppm [111]. This highly toxic gas is the example molecule we use to demonstrate this work, which combines control, readout, and communication electronics into a single IC to break through the aforementioned limits and enable ubiquitous and inexpensive environmental monitoring.

Proposed System

Overview

We demonstrate a complete H_2S gas sensor system consisting of only a chemical sensor IC and a single-chip micro-mote (SCμM) ADC/CPU/wireless transmitter IC requiring only a power source and antenna to operate and offering a higher level of integration compared to existing chip-scale sensing and communication systems [112]. The sensor IC is a chemically-sensitive FET (CS-FET): a bulk NMOS transistor controlled by gas molecules adsorbing on the gate [113]. Such a transistor was functionalized to respond to H_2S through a Au layer added to the gate and connected to the external sensor pad of the single-chip mote. The CS-FET was fabricated via custom process and SCμM was fabricated in standard TSMC 65nm CMOS. A connection diagram is given in Figure 7.1.

Operation

When power is applied, the mote biases the CS-FET through regulated voltage and a static resistor. The presence of H_2S on the CS-FET Au catalyst layer allows more current flow between source and drain in the same manner as raising gate voltage in a traditional NMOS transistor. This increased current flow develops a higher voltage on the static $95k\Omega$ resistor, which is amplified by the PGA by a factor of 2 and digitized by the 10-bit successive approximation register (SAR) ADC.

Software running on the on-chip ARM Cortex M0 CPU software controls power to the CS-FET, sets the PGA gain factor, triggers the ADC acquisition, and reads the result. Software then assembles the ADC result into a packet and transmits it by adjusting the current source of an RF oscillator to affect frequency shift keying (FSK) modulation. The oscillator is a current-starved, free-running, 8-stage 2.4GHz differential ring oscillator requiring no external crystal reference and its output is amplified by a switching power amplifier (PA).

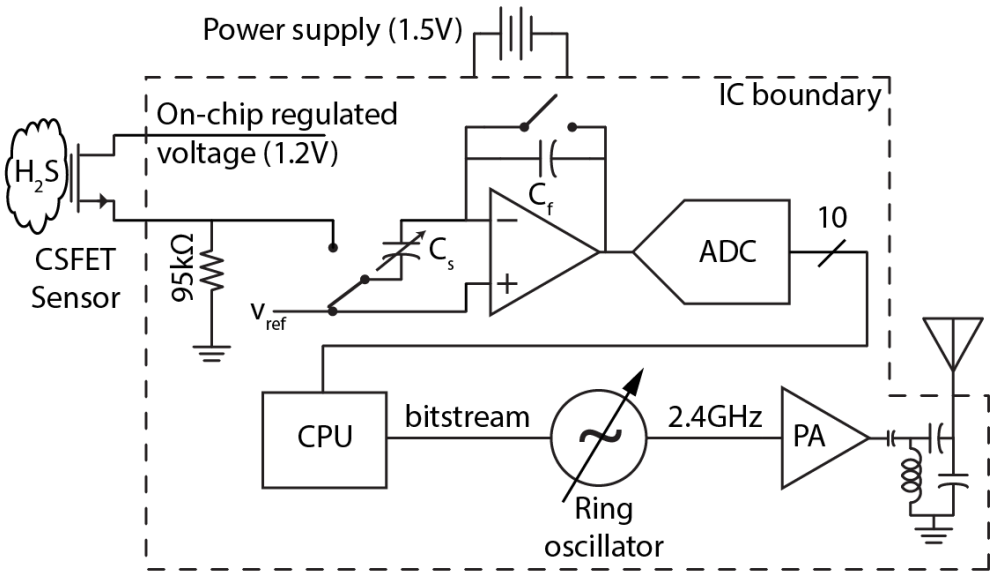


Figure 7.1: Schematic of CS-FET H2S sensor connections to Single-Chip Micro-Mote (SCμM) sensor frontend between on-chip voltage source and resistor, and major blocks of SCμM processing & RF transmitter system.

Sensor bias and amplification

CS-FET devices are expected to allow 1μA current flow at 5ppm H_2S concentration [113] (half the NIOSH recommended exposure limit [111]). Bulk CS-FETs are, ideally, biased with a minimum 1V source-drain voltage during operation [114]. SCμM generates a 1.2V sensor voltage and the current at 5ppm H_2S concentration is expected to develop 95mV across the static resistor, leaving 1.105mV to bias the CS-FET.

The developed voltage is amplified by a switched-capacitor programmable gain amplifiers (PGA). These typically use reference voltage of zero and, as a result, the voltage gain is simply the ratio of the sampling capacitor and feedback capacitor (C_s/C_f). We use nonzero Vref in our system so the expression for the PGA output voltage is:

$$V_{out} = \frac{C_s}{C_f} V_{in} + \left(\frac{C_s}{C_f} - 1 \right) V_{ref} \tag{7.1}$$

The sampling capacitor C_s is programmable between 1x and 256x larger than the feedback capacitor C_f. From this expression and with a nominal V_{ref} of 190mV generated on-chip, we can set amplifier gain (C_s/C_f ratio) to 2 and obtain a minimum V_{in}=95mV, above which V_{out} will be nonzero. This sets the minimum detectable input voltage to 95mV. At our H_2S target detection floor of 5ppm, the expected 1μA current is will develop 95mV across the static 95$k\Omega$ resistor and increase with higher gas concentration. Beyond approximately 10ppm, the source-drain voltage across the CS-FET will go below the recommended minimum. In this region of operation, the sensor remains responsive to increasing concentrations but at

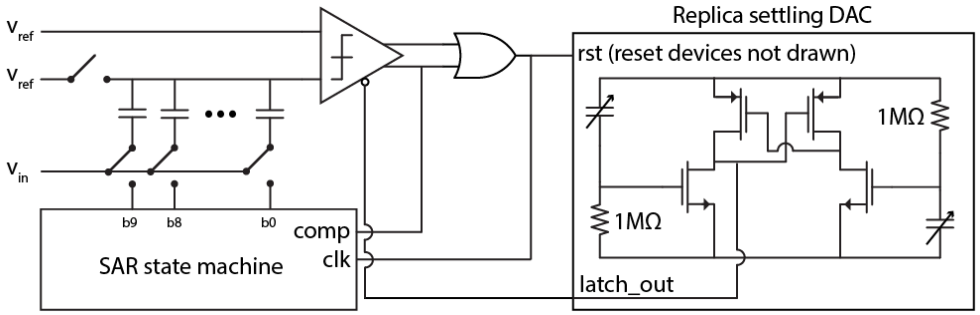


Figure 7.2: Schematic of self-timed ADC. After microprocessor triggers SAR state machine, each binary search occurs after comparator reset delay and replica settling DAC delay. Note: ADC reference voltage V_{ref} pictured here is different than PGA V_{ref} in Figure 7.1.

diminishing sensitivity and eventual railing of the input PGA voltage. This is acceptable behavior for our application, in which any detected concentration beyond 10ppm should result in immediate action to alleviate danger of H_2S poisoning.

Digital sampling

The PGA output connects to the ADC input. We take 200 ADC readings to average out thermal noise of the PGA reference voltage. ADC acquisition is split into three phases. During the first phase, the switched-capacitor PGA sampling capacitor is connected to Vin and the ADC is in reset. During the second phase, the PGA amplifies the sampled voltage and fills the ADC capacitor array. During the third phase, the ADC performs the conversion using internally-generated clocks and asserts an output when done. The CPU waits in a loop for the output to be asserted and reads the binary value from the SAR state machine.

The ADC design (Figure 7.2) is based on [115] and adds a comparator input settling timer via a replica DAC [116]. The replica DAC output is connected to the state machine to generate binary search clocks internally instead of requiring an external clock. A digital OR gate is used to prevent slowdown in case of very close comparator inputs which would result in high comparator delay [117]. The replica DAC is adjustable to produce overall ADC conversion times from $0.35\mu s$ to $1.72\mu s$.

Wireless Communication

On-chip software assembles the average of 200 ADC reading into a packet prepended with a Bluetooth LE-style preamble (0x07555555) and appended with a postscript (0x5533FF00) to assist a receiver in recovering data clock and recognizing the packet, respectively.

After assembling the packet, the CPU periodically adjusts the current source of the 2.4GHz ring oscillator to shift its frequency high or low depending on whether the next bit in the packet is a 1 or a 0. The separation between the “1” and “0” frequencies is typically small and changes fast because of the stability of crystal references. A calibrated crystal-free

radio is still able to interoperate with standards-compliant wireless devices if it is based on a resonant LC tank oscillator [118]. Ring oscillators, which are smaller than LC tanks and can also benefit from future reductions in transistor size, exhibit higher frequency variance which requires us to exaggerate the typical FSK frequency separations found in wireless standards. For instance, the IEEE 802.15.4 standard uses a frequency shift of $\pm$0.5MHz and a 0.5μs data period.

Literature indicates a free-running ring with approximately 500kHz jitter at 1ms intervals can achieve good communication performance (which we define as <1% packet error rate, or PER) using frequency shift of $\pm$1MHz and a 0.5μs data period [14]. The ring in this work has 1.52MHz simulated jitter at 1ms intervals so we expect to need to increase frequency shift to $\pm$3MHz.

Experimental Results

System Operation

Two CS-FET sensors were tested in conjunction with the SCμM IC: Sensor A was tested with 10ppm H_2S flow and Sensor B was tested with 50ppm. Tests consisted of industry-standard "bump" tests conducted by briefly flowing a pre-mixed concentration of H_2S gas and ambient air, waiting for the sensor to recover to baseline, and flowing gas again. All tests showed clear response. Continuous ADC readings for Sensor A are plotted in Figure 7.3 indicating clear response to 10ppm. The first flow was terminated before the sensor reached steady state. The second flow lasted long enough to see a steady-state value before being terminated. Both took approximately 6 seconds to reach their peak value.

Of note is that the steady-state value during the second flow is the maximum ADC code reachable by this frontend. Given a PGA gain of 2, and accounting for nonlinearities at the top of the PGA range, this indicates current flow of at least 5.7μA through the fixed 95$k\Omega$ resistor. This is in excess of the 1μA estimated and, indeed, it was found that bulk devices are more sensitive than those in SOI [119]. This result has the effect of lowering our sensitivity floor by at least 5x which is beneficial for an ambient monitoring device: 10ppm is defined as the safe upper limit but long-term exposure to 2ppm is known to cause fatigue, headache, loss of appetite, and more [120]. In applications where these sensors are worn in everyday circumstances or many are distributed across a city, a lower detection floor is beneficial in preventing low-level exposure of an individual or population.

Photos of the ICs are shown in Figure 7.4.

Data Acquisition

Each ADC conversion was measured to take 1.55μs but each acquisition cycle takes 337μs. The PGA is estimated to need at least 1.8μs to settle so the balance of time consists of software overhead required to, e.g., step the system through the three signal acquisition phases and store the result. We take 200 samples per data point for a total time of 67.4ms.

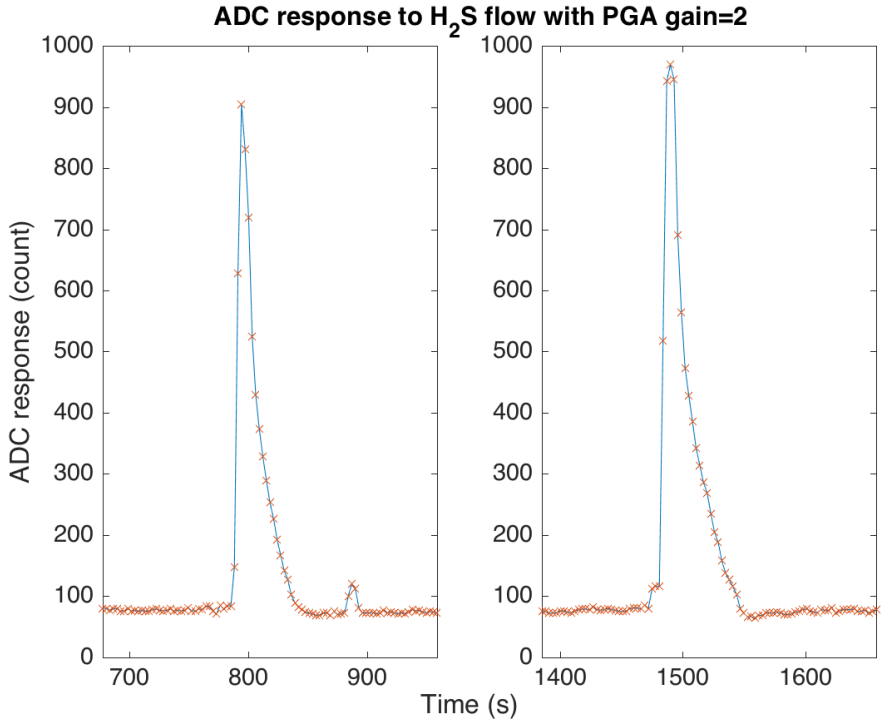


Figure 7.3: ADC values in response to two 10ppm H_2S flows (one short, one long) directed at Sensor A. Data points are marked with a red 'x' and were taken every 3s.

The system response with PGA gain set to unity was obtained by incrementing Vin in 0.1mV steps and acquiring 30,000 measurements at each step. The results are plotted in Figure 7.5.

The high DNL could indicate need for more thermal noise averaging, or a finer voltage step to more accurately determine ADC code boundaries. The wave-like plot of integral nonlinearity (INL) more clearly shows the same, opposite-phase, trend in the DNL. Its shape indicates low codes are represented by slightly more voltage than average and high codes are represented by slightly less. Fortunately our application does not demand high fidelity ADC performance and, worst-case, a lookup table to could be employed to compensate for these nonlinearities.

Data Transmission

Data was transmitted at a center frequency of 2.465GHz with an output power of -12dBm. A sample transmission is plotted in Figure 7.6 showing FSK-modulated data. Considerably more random frequency error (phase noise and jitter) was observed in the RF carrier due to a significant underestimation of flicker noise in the simulation models. The RF carrier was modulated by approximately ±7MHz to ensure high and low frequency shifts can be clearly distinguished. This necessity had an impact on data rate because the response time

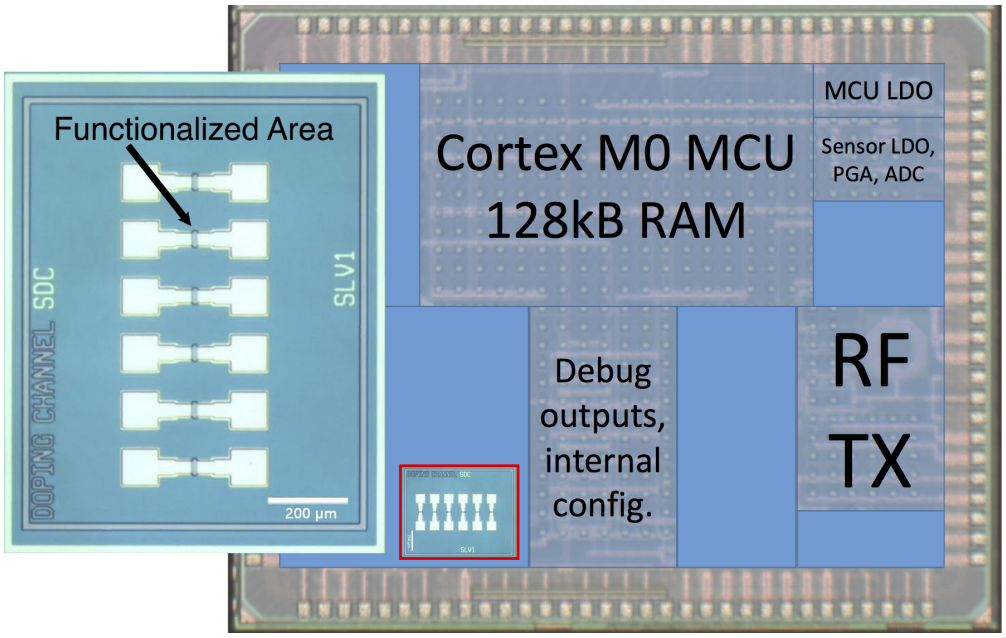


Figure 7.4: Right: die photo of Single-Chip Micro-Mote (SCμM) IC measuring 2.5mm x 3mm including microprocessor, RAM, sensor interface, and reference-free 2.4GHz RF transmitter. Lower center, red outline: CS-FET IC pictured at scale with respect to SCμM IC. Left: CS-FET IC at 4x larger-than-scale with one of six H_2S-sensitive functionalized areas indicated.

of the current DAC – which is responsible for FSK frequency shifts – is limited. A 249μs bit period was necessary to allow ample settling time for a 7MHz frequency shift. This means our proposed 12 byte packet takes 29.3ms to transmit. When added to the 67.4ms sensor sampling duration and compared with our overall 3 second sensor sampling period, we can expect a 3.0% duty cycle.

Power Consumption

The sensor, PGA, and ADC together consume 195μA from a 1.2V on-chip supply. The rest of the acquisition system including CPU, clocks, and voltage regulation consumes 604μA from a 0.8V on-chip supply. The free-running oscillator and power amplifier consume 1.08mA and 1.30mA from 1.2V, respectively, generated off-chip for experimental flexibility. Added together, the system consumes peak power of 0.71mW while sensing/acquiring and 2.85mW while transmitting. These values increase to 1.20mW and 3.57mW, respectively, if we assume supply voltages are regulated from a 1.5V battery. With 3.0% duty cycle and assuming a future version of our system consumes typical 1μA while asleep, we can expect 188μW average power and 358 hours (>2 weeks) of lifetime from a 45mAh SR41 Ag_2O button cell battery.

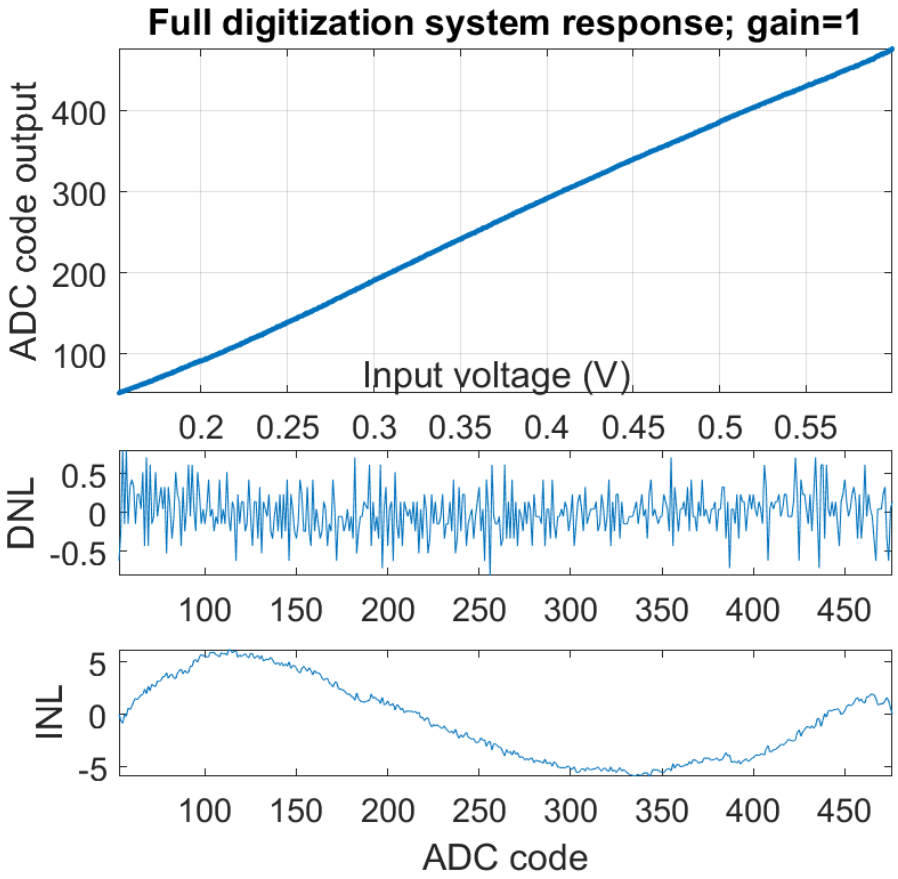


Figure 7.5: Voltage response of sensor frontend in region where system is least impacted by PGA/frontend nonlinearity and gain is set to unity. DNL extrema are [0.71, −0.81] and INL extrema are [5.4, −6.2].

Conclusions and Future Work

We have demonstrated a complete wireless hydrogen sulfide sensor consisting of sensor IC and sensor acquisition/communication IC and requiring only antenna and power source to operate. The system is clearly responsive to H_2S gas concentrations at the NIOSH recommended exposure limit. Performance is acceptable for our application but several design improvement opportunities were made clear during testing.

Software overhead is responsible for 99% of time taken and power consumed while taking a sensor reading. This underscores the need for hardware control instead.

Shifting frequency by changing ring oscillator current turned out to be unnecessarily slow. Instead, a tunable capacitive DAC could be attached to the ring oscillator to adjust ring oscillator capacitive load, and therefore frequency, rapidly. This could be done on the order of $1\mu s$, compared to this chip's $249\mu s$ limitation. This would also allow current DAC noise to

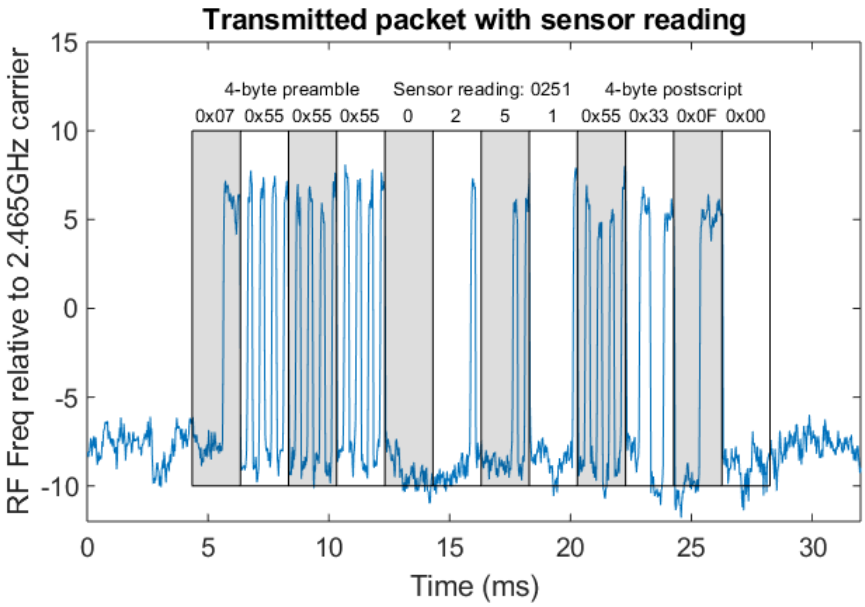


Figure 7.6: Wireless transmission of acquired sensor value during 50ppm H_2S flow over Sensor B.

be more aggressively filtered. Nominal power consumption would increase due to capacitive DAC parasitic loading but the results would be, in our opinion, well worth it given poor noise models.

Improved flicker noise models are clearly needed to allow minimum frequency shifts to be more accurately estimated in simulation. Tighter frequency spacing would reduce occupied bandwidth and increase the number of sensors capable of communicating in the 2.4GHz band.

With these straightforward improvements to shorten sensor readings to $3.5\mu s$ and packet transmission to $96\mu s$, we estimate duty cycle could be as low as 0.2% yielding an average power of $5.6\mu W$. Note the assumed $1\mu A$ sleep current is responsible for 53% of power consumed in this case. But even without such improvements, average power of $188\mu A$ paired with a scavenged energy source would enable unobtrusive, self-powered wearable toxic gas detection or cheap ubiquitously-distributable sensors capable of reporting dangerous conditions immediately.

7.2 Testing with other chemical sensors

The sensor frontend was connected to three types of chemical sensors as a proof-of-concept.

1. A bulk MOS CS-FET functionalized to respond to hydrogen sulfide gas [114]. This experiment was described in the previous section.

2. A printed, flexible sodium sensor for use in liquids [110, 121]

3. A printed, flexible lactate sensor, also for use in liquids [122].

Sodium sensor

The sodium sensor is potentiometric, so it needs no bias network. One terminal was connected to ground and the other to the ADC input. Three liquid sodium concentrations were prepared and applied to the sensor; these concentrations were expected to result in development of 20mV for each step from low to med and med to high.

The first experiment delaminated the printed sensor from its plastic substrate. Per the collaborator who provided the sensors, delamination had never been observed. The current guess is that, because 10k ADC acquisition cycles were initially used, enough positive current flowed into the sensor to force corrosion of the delaminated electrode. For the second and third experiments, fewer ADC cycles were used and delamination was not observed again.

Two-terminal lactate sensor

The lactate sensor was tested in 2-terminal configuration, as opposed to a traditional 3-terminal configuration for use with an electrochemical potentiostat. It was connected the same as the CS-FET: from supply, to ADC input, to pull-down resistor. The fixed resistor limited the range of testable lactate concentrations. Below the limit, too much voltage would be dropped across the sensor and potentially damage it. This limit was ~300mV. above the limit, so much current would flow through the sensor that the difference between supply voltage and voltage developed across the pull-down resistor was too small for the sensor to operate. With a commercial silver chloride reference electrode, this limit was ~200mV. Obviously a potentiostat would be beneficial on future SCμM revisions.

The sensor was connected and immersed in a selection of buffered lactate concentrations from 1 mol to 20 mol (this range was wider than the estimated limit) plus a buffer-only reset solution. The ADC response showed the expected exponential decay to the final value as obtained with traditional lab equipment. Tests were performed outside the estimated limit but detailed analysis of whether the electrode was damaged was not undertaken. If damage occurred, it was not catastrophic as in the sodium sensor because response curves were still observed throughout the test.

Chapter 8

Tapeout class: taking students from schematic to silicon in one semester

Transistor-level circuit design is taught widely at the university level but the actual process of realizing a chip is strange, obtuse, badly documented, with poor Internet resources (in contrast to, for instance, PCB design) and learning those skills is best likened to an apprenticeship into a secret trade. This chapter is based on my ISCAS 2018 paper [16] presented at the education session and describes our work to create a new course at Berkeley to help students get up to speed on chip design faster and more clearly.

Since this paper was written we were given the opportunity to re-manufacture the class IC with the digital components included. We submitted the revised GDS in December 2018 but, over a year later, the IC has yet to be received.

8.1 Abstract

In the spring of 2017, the UC Berkeley department of EECS introduced an innovative new course: "28nm SoC for IoT." This course went far beyond schematic-level design typical of circuits education and resulted in two chips going out for manufacturing: one analog and RF cores only, and one with all cores: analog, RF, and digital. Ten students with no prior IC experience, nine undergraduate and one graduate, designed and laid out an SoC in ST 28nm FD-SOI CMOS including a 2.4GHz transceiver, baseband filtering, ADC, Bluetooth MAC, a RISC-V CPU, and internal power regulation. The transceiver, baseband, ADC, and power regulation were sent out for fabrication and returned in the spring of 2017. The complete chip, including all cores, was sent out fabrication November 2017 and its return is pending. This paper discusses the instructors' experiences and results with this course.

8.2 Introduction

Motivation

The minimum size of a low-power standalone electronic system is usually dominated, besides its energy source, by the material needed to connect the system's components, e.g., the PCB. This also sets lower parasitic limits, which in turn set a lower bound on power consumption. Interposer advances allow tighter integration of disparate integrated circuits; however, current and future applications demand even further reductions in size. As a result, students intending to go into industry increasingly need to be prepared for extensive on-die design and integration. Present undergraduate and graduate coursework in this area is limited in its "reality" insofar as the vagaries of IC design and manufacturing are concerned.

In order to augment existing IC design coursework at UC Berkeley, we created a course to teach the development of mixed-signal wireless ICs, along with many aspects of IC development often not included in the classroom. The core goal of this course was to design and fabricate a real chip, in a real process, with a real deadline, to provide students with the most realistic SoC design experience possible.

This design experience included both interpersonal and technical elements. On the interpersonal side, we invited students to contribute to a larger team, and to deliver a design for which they alone were responsible. We emphasized clear communication of relevant performance metrics and design challenges through presentations and written documentation, as well as keeping to a semester-long project schedule. On the technical side, students with no past chip experience produced an extremely complex system that would give even accomplished IC veterans pause. Students went beyond their existing circuit education to produce designs which accounted for non-ideal power sources, included on-chip reconfigurability, designed with limitations from the outside world such as ESD and latch-up, and complied with the foundry's design rules. Along the way, we introduced students to industry-standard digital design & synthesis tools from Cadence, Mentor Graphics, and Synopsys in a real production environment which had bugs and inconsistencies; in contrast, polished academic generic process design kits tend to be free of such quirks.

Course background

This course was first offered by the University of California, Berkeley, in the Department of Electrical Engineering and Computer Sciences during the Spring 2017 semester. Titled "28nm SoC for IoT," the purpose of the course was to introduce students to the challenges of modern IC development by having them design a complete 2.4GHz transceiver and fully-featured microcontroller in a modern process, verify and integrate functionality, and send the design out to be manufactured. The chip was to be capable of satisfying Bluetooth Low-Energy (BLE) physical (PHY) and media access control (MAC) layer specifications [51] in hardware, and the rest of the BLE stack in software. The course consisted of 10 enrolled students (9 undergraduate, 1 graduate), each of whom had done well in at least one pre-existing electrical

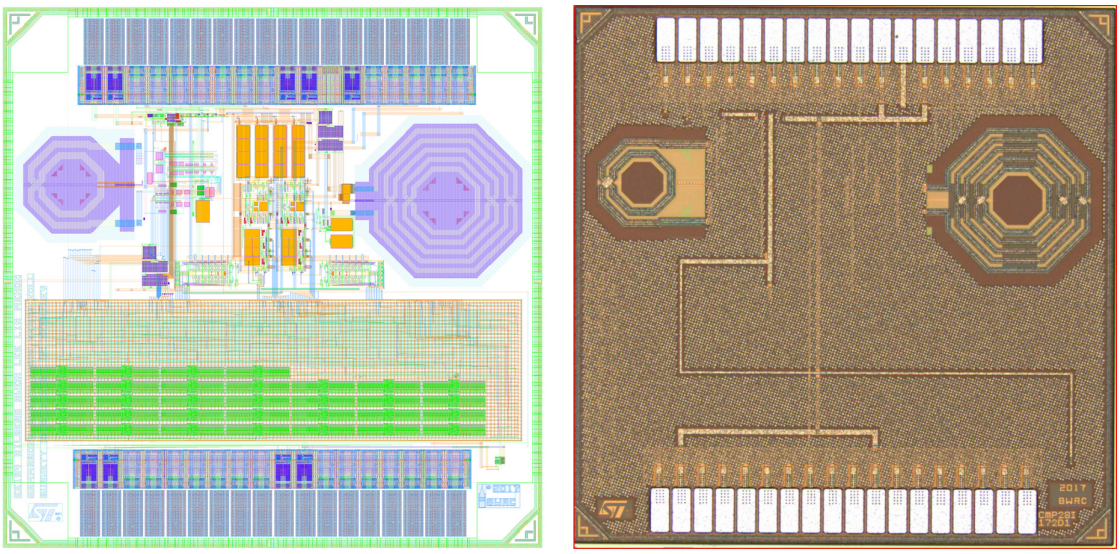


Figure 8.1: Final results of class project, measuring 1.1mm x 1.1mm. Left: screenshot of final layout including 2.4GHz transceiver & power regulation in top half, and microprocessor in bottom half. Unfortunately, synthesized digital elements in bottom half were not completed in time for first fabrication (second fabrication run included digital and is still pending). Right: photo of die produced from first accepted GDS, including transceiver and hand-built scan chain for debugging.

engineering undergraduate circuit design course: analog circuit design, digital circuit design, or RF circuit design. Three instructors were involved: David Burnett, graduate student instructor; Dr. Osama Khan, postdoctoral lecturer; and Prof. Kristofer S.J. Pister, faculty advisor.

By nature, the course was structured differently than a typical electrical engineering course. Instead of assigning every student or small group of students identical labs and projects throughout the semester, the instructors presented a high-level architecture of a wireless system-on-chip (SoC) on the first day of instruction. This architecture included everything necessary for an internet of things (IoT) device: RF components such as impedance match, mixer, and RF oscillator in an architecture based on our group's crystal-free radio work [12]; analog components such as baseband filtering & amplification, digitization, voltage regulators, and bandgap references; and digital components such as clock and data recovery, a Bluetooth Low-Energy MAC core to handle packets (based on Sahar Mesri's IEEE 802.15.4 MAC [7] and the Nordic nRF series BLE MAC [123]), and a microcontroller with RISC-V processor [124] generated with Rocket-chip [125], based on the Freedom E300 platform from SiFive, Inc [126] and written in CHISEL [127]. From the presented architecture, students volunteered to take responsibility for various blocks according to their background and interest. We were lucky: with 5 students choosing analog/RF blocks and 5 students choosing digital ones, there was enough interest and experience to handle the minimum of components to complete the

basic system.

The project was initially intended to fit in 1mm x 1mm of multi-project wafer (MPW) silicon sourced from a commercial vendor but, early in the semester, that vendor discontinued their MPW service. Fortunately, STMicroelectronics donated 1.1mm x 1.1mm of silicon in their 28nm fully-depleted silicon-on-insulator (FD-SOI) process [128] for use in the class, in the existing context of their collaboration projects with UC Berkeley. Once the project was completed, the GDS was submitted to ST for manufacturing. The resultant chip is displayed in Figure 8.1.

8.3 Overcoming student misconceptions

In a traditional engineering course, the final project is due at the end of the semester and demonstrates your skills as an engineer. In this course, by far the most challenging aspect was convincing students that their individual "project" e.g., the filter, or oscillator, or synthesized digital module, or other block, needed to be ready very early and was only the beginning of a larger integrated whole requiring significant effort to complete. This was one of many misconceptions about engineering work that academia tends to emphasize to students; examples of others are as follows:

- "99.99% is an A": students need to understand that where design rules are concerned, many foundries will simply not accept chips with nonzero design rule check (DRC) errors. We also observed students failing to investigate absolutely all layout-versus-schematic (LVS) errors or even disagreeing with the tool, where poorly-understood LVS errors can be catastrophic.

- "Performance is my only benchmark": students can benefit from a more holistic view of design with more focus on functionality and integration and less on meeting spec perfectly across process corners or accounting for rare or avoidable edge cases. The tendency to spend valuable hours practicing perfectionism resulted in weeks of timeline delays and loss of design integration and verification time.

- "My block is my only concern": it was a distinct challenge to convince students to anticipate interfaces with other blocks, or to demonstrate two or more of their blocks operating together.

- "Late is OK": the fabrication deadline is fixed. "Reduced functionality and on-time" is much more preferable to "late but fully-featured". This is the opposite to how project grading and late penalties are structured in most courses.

Our initial failure to motivate students to avoid these issues led to a severe timeline distortion. The intended design schedule was first upset by needing to change processes. Then, as the semester progressed, students were a little late with each design stage. This lateness compounded and, by the end of the semester, we had barely begun top-level chip

assembly. After the end of the semester, the teaching assistant and a few students were able to devote time to the chip between research and internship commitments until it was finally finished. We were lucky that the GDS due date was well after the end of the semester, or the tapeout would have been a total miss.

8.4 Teaching assistant to project manager

While students needed to approach this course differently, instructors also needed to modify their approach to this course in order to make it successful. This was particularly apparent in the role of teaching assistant (TA). The TA's traditional roles in guiding labs or emphasizing lecture material were replaced with responsibilities such as managing overall schedule, making high-level decisions about chip organization, and forecasting & preventing roadblocks. These roles are more traditionally associated with those of a technical project manager.

The TA was also responsible for using his design experience to advise students about upcoming tapeout steps. As discussed in Section 8.3, students initially focused predominantly on their own blocks to the exclusion of all else. Soon enough, the students began to focus on practical design considerations which most courses don't have time to explore: V_t mismatch, floorplanning, designing with real voltage, current, and frequency sources, defining interfaces and tuning, designing for test, accounting for parasitics, etc. By the end of the course, our students had learned to think about their designs in the bigger picture and to understand that schematic results are just the start of a physically-realizeable circuit.

Most of the TA's unique responsibilities were carried out continuously during the semester through weekly revisions to schedule, reminders about future steps, and emphasis on multi-block integration. Near the end of the semester it became clear that bugs in our digital synthesis tool flow would not be resolved in time. Assuming the IC would lack a microprocessor and digital peripherals, the TA led implementation of backup plans to allow testing of what had already been successfully put down in layout.

The most significant new TA challenge was a lack of expertise in the course material. The diversity of subsystems that go into a wireless SoC makes it unlikely to find a single person qualified to advise on all design aspects. Instead, the TA tapped resources outside of the course, usually other graduate students, to advise design of a particular block. We were grateful to find 11 such individuals, enumerated in the Course Acknowledgement section, to provide their expertise to our students.

8.5 Results

Our students put an incredible amount of effort into this class and it showed: at the end of the semester, we successfully taped out and submitted the design to the foundry. Issues with our digital synthesis toolchain led to synthesized components not making it in time for this semester's fabrication deadline. (The complete chip, including digital, was later resubmitted

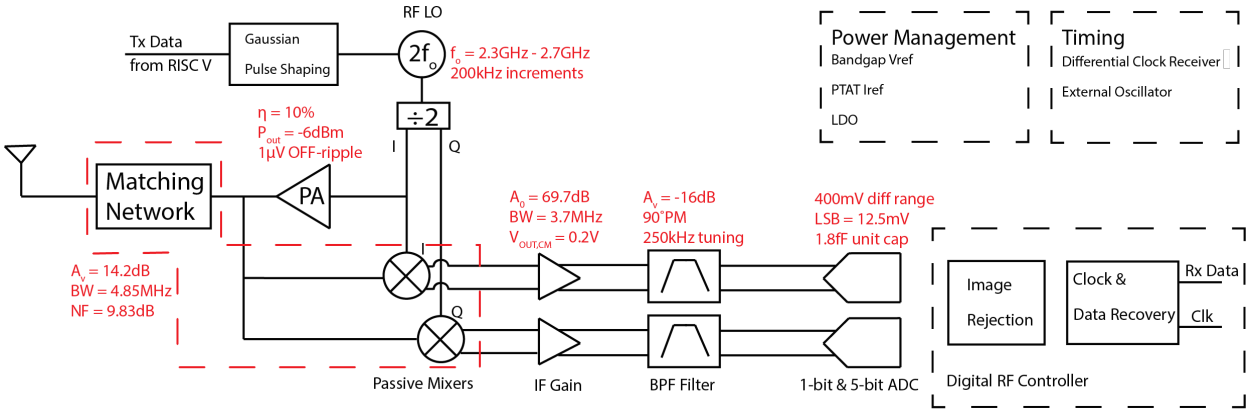


Figure 8.2: Block diagram of class project including simulated performance annotations in red. Digital blocks not pictured: RISC-V microprocessor, memory, DMA, and BLE MAC. Also included on chip but not pictured: clock receiver, courtesy of Pi-Feng Chiu.

for fabrication in November 2017.) A screenshot of the layout including synthesized digital components is shown on the left side of Figure 8.1 and a photo of the manufactured design, lacking synthesized digital, is shown on the right. A block diagram of the project's final form, though lacking detail on the digital components, is given in Figure 8.2. The components successfully taped out include:

- 2.4GHz Bluetooth low-energy compliant transceiver
  - RF frontend matched to off-chip antenna
  - Power amplifier
  - Differential IQ passive mixer & 4.8GHz LC tank
  - Low-IF baseband amplification, filtering, and ADC
- Power management of entire chip from a single unregulated supply
- Microcontroller w/ custom digital peripherals in CHISEL (DRC & LVS clean but not fabricated)
  - 32-bit RISC-V processor & 64kB SRAM
  - Bluetooth packet handling
  - GFSK oscillator modulator
  - Clock & data recovery

At time of writing, preliminary tests of the chip indicate the power regulation is functioning. Testing is ongoing.

For future tapeout courses

As we reflect on the strengths and weaknesses of the semester, we have a few specific future recommendations. It is our hope that these recommendations will help future courses like this one, including our own.

- More meetings: partway through the semester, we devoted an hour each week to coordination meetings for analog and digital teams. Progress markedly improved. These meetings should have begun on the first week.
- Foster collaboration: students will seldom meet on their own but, when called together by instructors, will stick together and continue working well after the called meeting. We fostered this tendency by scheduling the aforementioned one hour analog or digital coordination meetings but reserving the room for two hours. After the instructors left, students continued to hammer out their designs.
- Devote in-class time to progress presentations: about halfway through the semester, every student was expected to present for at least 5 minutes per week, every week. This was essential for keeping everyone up to speed on project development, while motivating students to make progress. Early in the semester, we spent a lot of time giving lectures and assigning homework in pursuit of fostering a better understanding of wireless communication. However, because all enrolled students had already demonstrated proficiency in one of the foci necessary to enable this project, much of this background was not necessary to start individual block design. In the future, theory lectures can likely be moved to later in the semester and early lectures should focus on giving students a thorough system overview.
- Grading structure: We were, perhaps, too reliant on students' intrinsic motivation and placed a low emphasis on assigning tasks with clear grades attached. Future courses should include frequent graded milestones with stiff penalties for lateness to emphasize the need for a quick, "good enough," solution instead of a late, better, one. In general, grading emphases should resemble performance reviews in industry, including timeliness, communication responsiveness, and cooperation.
- Demand deliverables: there are a few specific milestones which students should be made to take seriously via grading structure. Every student should produce:
  - System-level considerations & design: all interfaces for their block, including power, and must either terminate at a pad or match an adjacent student's interface exactly,
  - System block diagram: their concept of the whole system and where theirs fits in,
  - Design review: this semester, several blocks escaped a critical eye by accident. All should be reviewed by an expert before progressing too far in design, and

  - Performance measurements: make sure students demonstrate their block works as advertised.

- Hide buffer time in the schedule: we were able to manufacture a chip in large part because we never discussed the foundry's true due date with students. Instead we opted to build tolerance into the schedule and kept students aiming for an earlier deadline.

Regarding curriculum, as noted above, we attempted to teach wireless communication background early in the course. We speculate that early lectures could be better spent teaching ICs from the outside in to introduce the real-world issues students will need to consider in their designs. These include the voltage levels and communication protocol (commonly JTAG) of the test equipment , mechanical design considerations of the PCB, parasitics of wirebonds and pads, ESD prevention and input/output pad drivers, high-voltage IO MOSFETs vs core voltage MOSFETs, and prevention of latchup and antenna violations.

Feedback

The course has generated a significant amount of interest (and initial skepticism) from the local UC Berkeley community. We had approximately 10 extra graduate students auditing it off-and-on throughout the semester. It has since been a popular subject of conversation among the graduate and undergraduate communities, and has been added to the department's Master of Engineering program as a capstone project course for the Physical Electronics and Integrated Circuits track.

We surveyed the students at the end of the semester and asked for their anonymous feedback. First, we asked them to rate the necessity of various elements of the course from 1 (could be removed next year) to 5 (absolutely necessary). Our recommendations in Section 8.5 were in line with these student survey results, which were read only after Section 8.5 was written: weekly in-class presentations and analog/digital coordination meetings, which were rated 97% necessary and 93% necessary, respectively, were seen as much more necessary than assignments and lectures which were rated 70% and 50%, respectively.

We asked, "if you had more time on this chip, what would you do?" Responses centered around testing, fixing the design, and learning more about the tools.

Unsurprisingly, responses to "how would you change things next year?" and "what will you do differently on your next chip?" were almost exclusively about getting organized sooner and completing individual block design earlier.

Lastly, there were a couple of good student statements that summarize the technical and non-technical outcomes of this course:

> "I learned how to bridge circuit design to system design. I also learned how to design a chip, not just a circuit."

> "[I learned about] communication across function-sets of the chip, learning to set expectations, just saying no when things are unreasonable, making a decision even when the direction isn't clear (you just need *a* decision, not *the* decision)"

8.6 Conclusion

We have taught the first iteration of a unique new design course, pooling talented students successful in prior RF, analog, and digital design courses, to create a cohesive team to design and fabricate a 2.4GHz wireless sensor node.

Many educational institutions already have access to industry-standard semiconductor tools and, even without donated silicon, could offer a similar course on a modest budget. For instance, MPW space in 350nm is on the order of US$1000 per mm^2. It is our hope that any institution wishing to offer a tapeout class will benefit from our experience to more efficiently teach this material.

8.7 Course Acknowledgement

Creating this course would have been impossible without the support of numerous individuals and organizations. First we thank STMicroelectronics, and Dr. Andreia Cathelin in particular, for a generous donation of 28nm FD-SOI silicon for this course. We would also like to thank Circuits Multi Projets (CMP), and Jean-François Paillotin in particular, for assistance with preparing our final submission. The students, staff, faculty, and sponsors of the Berkeley Wireless Research Center (BWRC) were instrumental in supporting the practical side of this course with software tools, compute resources, and process expertise, especially the staff who facilitated our use of those resources: Yessica Bravo, Candy Corpus, James Dunn, and Brian Richards. Our students would not have been able to produce what they did without Berkeley graduate student guest lecturers & design mentors: Stevo Bailey, Pi-Feng Chiu, Jaeduk Han, Ben Keller, Greg Lacaille, Filip Maksimovic, Brad Wheeler, John Wright, and guest lecturers & design mentors from industry: Henry Cook and Megan Wachs from SiFive and Sahar Mesri from Qualcomm. Finally, we wish to thank the ten students who bravely enrolled in this ambitious course and thereby served as our guinea pigs: Brian Kilberg, Paul Kwon, Lydia Lee, Andrew Lin, Ramakrishnan Menon, Luke Sammarone, Alex Spaeth, Edward Wang, Ruijie Zhang, and Rachel Zoll.

Chapter 9

Conclusion

We have demonstrated a 2.4GHz communication system based off of ultra-small (4-stage differential ring measuring $10\mu m$ x $10\mu m$), low power (burning as little as $105\mu A$ from a 1V supply), crystal-free free-running RF ring oscillators – first as a receiver listening to a traditional signal with low phase noise, then listening to a transmitter with similarly high local oscillator phase noise characteristics as in the receiver. This required extensive work in understanding the nature and measurement of noise sources in these oscillators which can typically be ignored when locked to a high-quality external reference. The transceiver achieved -67dBm sensitivity using a receiver with negligible gain and approximately 10-12dB better sensitivity using a receiver with a proper matching network and some modest IF amplification. The transmitter was extended to serve as a complete wireless chemical sensor system with high potential for miniaturization. Elements of this work contributed to making the Single-Chip Mote project a success, demonstrating a system-on-chip requiring only a power source to operate. That project system later became the basis of a new ongoing course at Berkeley.

For everything that was achieved, many new questions were spawned. The most salient question is RF ring simulation vs. measurement: establishing why simulated results are so far from results in the lab. The most likely culprit is underestimation of flicker noise in the foundry's simulation models. Determining and fixing the discrepancy is vital for successful future designs. Ring oscillators' size and power consumption, in comparison to resonant LC tank or PLL-based alternatives, motivate the continuation of this work.

Many future lab experiments could be undertaken currently-available hardware:

- We have everything we need to build a complete software-based SCM3 demo showing the system going to sleep and the 32kHz timer waking it up.
- The Bluetooth LE packet assembler/disassembler cores (started by EE194 2017 students and improved by MAS-IC students afterward) and arbitrary bit FIFOs could combine to make a complete BLE MAC. The demonstration software has yet to be written.

- Demodulation of BLE packet on SCM3: this should only be a matter of spending the time to tune the right receiver parameters. Once demodulated, the packet should pour into the receive FIFO mentioned above.

- Test and refine our fabricated MEMS resonators. If a system includes a MEMS plane with sensors and actuators, it may as well have a MEMS resonator to help with frequency stability.

- Correcting the rings with an FLL was coarsely demonstrated in Chapter 2 and shown in Figure 3.9. To what extent flicker noise contributions could be mitigated over the short term? This FLL could be referenced to an external crystal, one of our fabricated-but-untested MEMS resonators, or − ideally − an internal relaxation oscillator. Our relaxation oscillators have small flicker components and have better noise performance at the loop bandwidth.

Some future projects only require theory and/or simulation:

- How does the power/phase noise tradeoff in the LO (assuming we can spend power to reduce phase noise in the first place) affect traditional noise figure tradeoffs, choice of IF , etc.? The effect of phase noise on receiver SNR has been studied [129] but assumes much lower jitter than what these ring oscillators exhibit.

- More clearly demonstrate the concepts in Chapter 4. There's still a ways to go in explaining these principles from the ground up; demonstrations in Matlab would help.

- Make comparisons between matched filter and period-based demodulation performance in general and for noisy a ring IF. Preliminary work as part of this dissertation indicates matched filters perform poorly with free-running rings because the IF drifts away from matched filter templates. Using lots of matched filter templates to bin the frequency and follow IF drift approaches the same results as measuring period from zero crossings (and might also be the same as finding the highest peak on an FFT), but could have some hardware advantages like being able to evaluate all filter templates in parallel. And, above all, the work in Chapter 6 used a 10-bit ADC at 100MSps. In a fully on-chip implementation, that would likely consume enough power to negate the ring oscillator's power savings.

- Instead of traditional FSK, try modulation better suited for low SNR e.g., JT65 [130]. A noisy LO doesn't necessarily mean a low SNR per se, but schemes using a pilot tone like JT65 are also better suited to IF drift.

Finally, some projects entail new designs:

- Re-design RF ring oscillators for less susceptibility to flicker noise. And use capacitive tuning to do frequency shifting during FSK transmit: by using current tuning, we need the current network to respond faster than one bit period. This increases the bandwidth

of the current mirrors, which may have been a source of noise. Ring oscillators are small, so use two: one only tunable slowly (current DAC-limited) for receive and one with a fast capacitive DAC for transmit.

- Eliminate all inductors by replacing the passive antenna match with an active LNA.
- Integrate ring oscillator results and an LNA with Richard Su's synthesizable PLL work (with relaxation oscillator reference) [131] to make a fully-synthesizable, inductor-free radio.
- Fix issues with the sensor ADC and mux its input with all 16 GPIOs.
- Add a potentiostat and TIA to the list of sensor ADC input options.
- Add a MEMS resonator driver/oscillator.
- Integrated antenna: At 15GHz, $\lambda/4 = 5mm$ and we enter the domain of on-chip antennas with good performance [132].

Hopefully these projects can be addressed myself, through collaborators, and with future students in the years to come.

Bibliography

[1] Bluegiga (now Silicon Labs). "BLE112 Bluetooth low energy single mode module". In: *Datasheet Version 1.0* (Dec. 2011), pp. 1–33. URL: https://www.silabs.com/documents/login/data-sheets/BLE112-DataSheet.pdf.

[2] Red Bear Labs. "Bluetooth Low Energy (BLE) 4.2 Module". In: *Model MB-N2 Data Sheet* (Nov. 2016), pp. 1–12. URL: https://github.com/redbear/nRF5x/blob/master/nRF52832/datasheet/MB-N2_Datasheet.pdf.

[3] K. Mann, T. L. Massey, S. Guha, J. P. van Kleef, and M. M. Maharbiz. "A wearable wireless platform for visually stimulating small flying insects". In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Aug. 2014, pp. 1654–1657. DOI: 10.1109/EMBC.2014.6943923.

[4] Craig Schindler, Daniel Drew, Brian Kilberg, Felipe Campos, Soichiro Yanase, and Kris Pister. "MIMSY: The Micro Inertial Measurement System for the Internet of Things". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (WF-IoT 2019)*. Limerick, Ireland, Apr. 2019.

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

[6] Brad Wheeler. "Low Power, Crystal-Free Design for Monolithic Receivers". PhD thesis. EECS Department, University of California, Berkeley, May 2019.

[7] Sahar Mesri. "Design and User Guide for the Single Chip Mote Digital System". MA thesis. EECS Department, University of California, Berkeley, May 2016. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-71.html.

[8] Lorenz Schmid. "Driving and Testing Crystal-free Radios for a Basic End-to-End Transmission". MA thesis. Section of Electrical Engineering, Electronics and Microelectronics, École Polytechnique Fédérale de Lausanne (EPFL), Aug. 2015. URL: https://lsm.epfl.ch/page-16595-en.html.

[9] O. Khan, B. Wheeler, F. Maksimovic, D. Burnett, A. M. Niknejad, and K. 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 (July 2017), pp. 777–781. ISSN: 1549-7747. DOI: 10.1109/TCSII.2016.2611643.

[10] O. Khan, B. Wheeler, D. Burnett, F. Maksimovic, S. Mesri, K. Pister, and A. Niknejad. "Frequency reference for crystal free radio". In: *2016 IEEE International Frequency Control Symposium (IFCS)*. May 2016, pp. 1–2. DOI: 10.1109/FCS.2016.7546749.

[11] O. Khan, D. Burnett, F. Maksimovic, B. Wheeler, S. Mesri, A. Sundararajan, B. L. Zhou, A. Niknejad, and K. Pister. "Time Keeping Ability of Crystal Free Radios". In: *IEEE Internet of Things Journal* (2018), pp. 1–1. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2869143.

[12] B. Wheeler, F. Maksimovic, N. Baniasadi, S. Mesri, O. Khan, D. Burnett, A. Niknejad, and K. Pister. "Crystal-free narrow-band radios for low-cost IoT". In: *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. June 2017, pp. 228–231. DOI: 10.1109/RFIC.2017.7969059.

[13] 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: *2019 IEEE INFOCOM WKSHPS: CNERT 2019: Computer and Networking Experimental Research using Testbeds (INFOCOM 2019 WKSHPS - CNERT 2019)*. Paris, France, Apr. 2019.

[14] D. C. Burnett, B. Wheeler, F. Maksimovic, O. Khan, A. M. Niknejad, and K. S. J. Pister. "Narrowband communication with free-running 2.4GHz ring oscillators". In: *2017 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*. Nov. 2017, pp. 1–6. DOI: 10.23919/PEMWN.2017.8308023.

[15] David Burnett, Brad Wheeler, Lydia Lee, Filip Maksimovic, Arvind Sundararajan, Osama Khan, and Kris Pister. "CMOS oscillators to satisfy 802.15.4 and Bluetooth LE PHY specifications without a crystal reference". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (IEEE CCWC 2019)*. Las Vegas NV, USA, Jan. 2019.

[16] D. C. Burnett, B. Kilberg, R. Zoll, O. Khan, and K. S. J. Pister. "Tapeout class: Taking students from schematic to silicon in one semester". In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2018, pp. 1–5. DOI: 10.1109/ISCAS.2018.8351506.

[17] L. J. Chu. "Physical Limitations of Omni-Directional Antennas". In: *Journal of Applied Physics* 19.12 (Dec. 1948), pp. 1163–1175. ISSN: 0021-8979. DOI: 10.1063/1.1715038. URL: https://aip.scitation.org/doi/abs/10.1063/1.1715038 (visited on 02/04/2019).

[18] H. A. Wheeler. "Fundamental Limitations of Small Antennas". In: *Proceedings of the IRE* 35.12 (Dec. 1947), pp. 1479–1484. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1947.226199.

[19] Roger F Harrington. "Effect of antenna size on gain, bandwidth, and efficiency". In: *J. Res. Nat. Bur. Stand* 64.1 (1960), pp. 1–12.

[20] B. W. Cook, A. D. Berny, A. Molnar, S. Lanzisera, and K. S. J. Pister. "An Ultra-Low Power 2.4GHz RF Transceiver for Wireless Sensor Networks in 0.13/spl mu/m CMOS with 400mV Supply and an Integrated Passive RX Front-End". In: *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*. Feb. 2006, pp. 1460–1469. DOI: 10.1109/ISSCC.2006.1696196.

[21] A. Molnar, B. Lu, S. Lanzisera, B.W. Cook, and K.S.J. Pister. "An ultra-low power 900 MHz RF transceiver for wireless sensor networks". In: *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*. Oct. 2004, pp. 401–404. DOI: 10.1109/CICC.2004.1358833.

[22] A.A. Abidi. "Phase Noise and Jitter in CMOS Ring Oscillators". In: *IEEE Journal of Solid-State Circuits* 41.8 (Aug. 2006), pp. 1803–1816. ISSN: 0018-9200. DOI: 10.1109/JSSC.2006.876206.

[23] David C. Burnett, Benjamin L. Smarr, Sahar M. Mesri, Lance J. Kriegsfeld, and Kristofer S. J. Pister. "Reconfigurable, Wearable Sensors to Enable Long-duration Circadian Biomedical Studies". In: *Proceedings of the 9th International Conference on Body Area Networks*. BodyNets '14. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 142–146. ISBN: 978-1-63190-047-1. DOI: 10.4108/icst.bodynets.2014.257078. URL: http://dx.doi.org/10.4108/icst.bodynets.2014.257078 (visited on 02/24/2015).

[24] B. L. Smarr, D. C. Burnett, S. M. Mesri, K. S. J. Pister, and L. J. Kriegsfeld. "A Wearable Sensor System with Circadian Rhythm Stability Estimation for Prototyping Biomedical Studies". In: *IEEE Transactions on Affective Computing* 7.3 (July 2016), pp. 220–230. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2015.2511762.

[25] I. Tien, S.D. Glaser, and M.J. Aminoff. "Characterization of gait abnormalities in Parkinson's disease using a wireless inertial sensor system". In: *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. Aug. 2010, pp. 3353–3356. DOI: 10.1109/IEMBS.2010.5627904.

[26] E. Jovanov, A. Milenkovic, S. Basham, D. Clark, and D. Kelley. "Reconfigurable intelligent sensors for health monitoring: a case study of pulse oximeter sensor". In: *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*. Vol. 2. Sept. 2004, pp. 4759–4762. DOI: 10.1109/IEMBS.2004.1404317.

[27] A. Burns, B.R. Greene, M.J. McGrath, T.J. O'Shea, B. Kuris, S.M. Ayer, F. Stroiescu, and V. Cionca. "SHIMMER^{TM}; A Wireless Sensor Platform for Noninvasive Biomedical Research". In: *Sensors Journal, IEEE* 10.9 (Sept. 2010), pp. 1527–1534. ISSN: 1530-437X. DOI: 10.1109/JSEN.2010.2045498.

[28] Andrey Loboda, Walter Kraft, Bernard Fine, Jeffrey Joseph, Michael Nebozhyn, Chunsheng Zhang, Yudong He, Xia Yang, Christopher Wright, Mark Morris, Ira Chalikonda, Mark Ferguson, Valur Emilsson, Amy Leonardson, John Lamb, Hongyue Dai, Eric Schadt, Howard Greenberg, and Pek Lum. "Diurnal variation of the human adipose transcriptome and the link to metabolic disease". In: *BMC Medical Genomics* 2.1 (2009), p. 7. ISSN: 1755-8794. DOI: 10.1186/1755-8794-2-7. URL: http://www.biomedcentral.com/1755-8794/2/7.

[29] Andrey A Ptitsyn, Sanjin Zvonic, and Jeffrey M Gimble. "Digital Signal Processing Reveals Circadian Baseline Oscillation in Majority of Mammalian Genes". In: *PLoS Comput Biol* 3.6 (June 2007), e120. DOI: 10.1371/journal.pcbi.0030120. URL: http://dx.plos.org/10.1371%2Fjournal.pcbi.0030120.

[30] Jennifer A. Mohawk, Carla B. Green, and Joseph S. Takahashi. "Central and Peripheral Circadian Clocks in Mammals". In: *Annual Review of Neuroscience* 35.1 (2012). PMID: 22483041, pp. 445–462. DOI: 10.1146/annurev-neuro-060909-153128. eprint: http://dx.doi.org/10.1146/annurev-neuro-060909-153128. URL: http://dx.doi.org/10.1146/annurev-neuro-060909-153128.

[31] Jeremy C. Borniger, Santosh K. Maurya, Muthu Periasamy, and Randy J. Nelson. "Acute dim light at night increases body mass, alters metabolism, and shifts core body temperature circadian rhythms". In: *Chronobiology International* (June 16, 2014), pp. 1–9. ISSN: 1525-6073. DOI: 10.3109/07420528.2014.926911.

[32] Tracy A. Bedrosian, Celynn A. Vaughn, Anabel Galan, Ghassan Daye, Zachary M. Weil, and Randy J. Nelson. "Nocturnal light exposure impairs affective responses in a wavelength-dependent manner". In: *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* 33.32 (Aug. 7, 2013), pp. 13081–13087. ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.5734-12.2013.

[33] Ilia N Karatsoreos. "Effects of circadian disruption on mental and physical health". In: *Current neurology and neuroscience reports* 12.2 (Apr. 2012), pp. 218–225. ISSN: 1534-6293. DOI: 10.1007/s11910-012-0252-0.

[34] Matthias Vogel, Tanja Braungardt, Wolfgang Meyer, and Wolfgang Schneider. "The effects of shift work on physical and mental health". In: *Journal of neural transmission (Vienna, Austria: 1996)* 119.10 (Oct. 2012), pp. 1121–1132. ISSN: 1435-1463. DOI: 10.1007/s00702-012-0800-4.

[35] Kenneth P Wright Jr, Joseph T Hull, and Charles A Czeisler. "Relationship between alertness, performance, and body temperature in humans". In: *American journal of physiology. Regulatory, integrative and comparative physiology* 283.6 (Dec. 2002), R1370–1377. ISSN: 0363-6119. DOI: 10.1152/ajpregu.00205.2002.

[36] Till Roenneberg, Karla V Allebrandt, Martha Merrow, and Céline Vetter. "Social jetlag and obesity". In: *Current biology: CB* 22.10 (May 22, 2012), pp. 939–943. ISSN: 1879-0445. DOI: 10.1016/j.cub.2012.03.038.

[37] David A. Bechtold, Julie E. Gibbs, and Andrew S. I. Loudon. "Circadian dysfunction in disease". In: *Trends in Pharmacological Sciences* 31.5 (May 2010), pp. 191–198. ISSN: 0165-6147. DOI: 10.1016/j.tips.2010.01.002. URL: http://www.sciencedirect.com/science/article/pii/S0165614710000039 (visited on 06/24/2014).

[38] Fiona C. Baker and Helen S. Driver. "Circadian rhythms, sleep, and the menstrual cycle". In: *Sleep Medicine* 8.6 (Sept. 2007), pp. 613–622. ISSN: 1389-9457. DOI: 10.1016/j.sleep.2006.09.011. URL: http://www.sciencedirect.com/science/article/pii/S1389945706006216 (visited on 07/04/2013).

[39] Horacio O de la Iglesia and William J Schwartz. "Minireview: timely ovulation: circadian regulation of the female hypothalamo-pituitary-gonadal axis". In: *Endocrinology* 147.3 (Mar. 2006), pp. 1148–1153. ISSN: 0013-7227. DOI: 10.1210/en.2005-1311.

[40] Wilbur P Williams 3rd and Lance J Kriegsfeld. "Circadian control of neuroendocrine circuits regulating female reproductive function". In: *Frontiers in endocrinology* 3 (2012), p. 60. ISSN: 1664-2392. DOI: 10.3389/fendo.2012.00060.

[41] Benjamin L Smarr, Kimberly J Jennings, Joseph R Driscoll, and Lance J Kriegsfeld. "A time to remember: The role of circadian clocks in learning and memory". In: *Behavioral neuroscience* (Apr. 7, 2014). ISSN: 1939-0084. DOI: 10.1037/a0035963.

[42] Cheryl Wotus, Travis R Lilley, Adam S Neal, Nicole L Suleiman, Stefanie C Schmuck, Benjamin L Smarr, Brian J Fischer, and Horacio O de la Iglesia. "Forced desynchrony reveals independent contributions of suprachiasmatic oscillators to the daily plasma corticosterone rhythm in male rats". In: *PloS one* 8.7 (2013), e68793. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0068793.

[43] Vitaliy Kolodyazhniy, Jakub Späti, Sylvia Frey, Thomas Götz, Anna Wirz-Justice, Kurt Kräuchi, Christian Cajochen, and Frank H. Wilhelm. "Estimation of Human Circadian Phase via a Multi-Channel Ambulatory Monitoring System and a Multiple Regression Model". In: *Journal of Biological Rhythms* 26.1 (2011), pp. 55–67. DOI: 10.1177/0748730410391619. eprint: http://jbr.sagepub.com/content/26/1/55.full.pdf+html. URL: http://jbr.sagepub.com/content/26/1/55.abstract.

[44] Vitaliy Kolodyazhniy, Jakub Späti, Sylvia Frey, Thomas Götz, Anna Wirz-Justice, Kurt Kräuchi, Christian Cajochen, and Frank H. Wilhelm. "An Improved Method for Estimating Human Circadian Phase Derived From Multichannel Ambulatory Monitoring and Artificial Neural Networks". In: *Chronobiology International* 29.8 (2012). PMID: 22891656, pp. 1078–1097. DOI: 10.3109/07420528.2012.700669. eprint: http://dx.doi.org/10.3109/07420528.2012.700669. URL: http://dx.doi.org/10.3109/07420528.2012.700669.

[45] B.A. Warneke, M.D. Scott, B.S. Leibowitz, Lixia Zhou, C.L. Bellew, J.A. Chediak, J.M. Kahn, B.E. Boser, and K.S.J. Pister. "An autonomous 16 mm3 solar-powered node for distributed wireless sensor networks". In: *Sensors, 2002. Proceedings of IEEE*. Vol. 2. 2002, 1510–1515 vol.2. DOI: 10.1109/ICSENS.2002.1037346.

[46] Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester. "A Survey on Wireless Body Area Networks". In: *Wirel. Netw.* 17.1 (Jan. 2011), pp. 1–18. ISSN: 1022-0038. DOI: 10.1007/s11276-010-0252-4. URL: http://dx.doi.org/10.1007/s11276-010-0252-4.

[47] K. S. Kwak, S. Ullah, and N. Ullah. "An Overview of IEEE 802.15.6 Standard". In: *ArXiv e-prints* (Feb. 2011). arXiv: 1102.4106 [cs.NI].

[48] 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. ISSN: 2161-3915. DOI: 10.1002/ett.2558. URL: http://dx.doi.org/10.1002/ett.2558.

[49] "IEEE Standard for Local and metropolitan area networks, Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)". In: *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)* (Sept. 2011), pp. 1–314. DOI: 10.1109/IEEESTD.2011.6012487.

[50] John Notor, Anthony Caviglia, and Gary Levy. *CMOS RFIC architectures for IEEE 802.15.4 networks - Electronic Products*. 2003. URL: http://www.electronicproducts.com/Analog_Mixed_Signal_ICs/CMOS_RFIC_architectures_for_IEEE_802_15_4_networks.aspx (visited on 12/13/2016).

[51] Bluetooth SIG. "Bluetooth Specification, Core System Package, Low Energy Controller volume". In: *Bluetooth Core Specification v5.0* (Dec. 2016), pp. 2533–2790.

[52] Liang Dai and R. Harjani. "Design of low-phase-noise CMOS ring oscillators". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49.5 (May 2002), pp. 328–338. ISSN: 1057-7130. DOI: 10.1109/TCSII.2002.801409.

[53] K. R. Lakshmikumar, V. Mukundagiri, and S. L. J. Gierkink. "A Process and Temperature Compensated Two-Stage Ring Oscillator". In: *2007 IEEE Custom Integrated Circuits Conference*. Sept. 2007, pp. 691–694. DOI: 10.1109/CICC.2007.4405826.

[54] D. J. Allen and A. L. Carley. "Free-Running Ring Frequency Synthesizer". In: *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*. Feb. 2006, pp. 1502–1511. DOI: 10.1109/ISSCC.2006.1696201.

[55] F. Sebastiano, L. J. Breems, K. A. A. Makinwa, S. Drago, D. M. W. Leenaerts, and B. Nauta. "A 65-nm CMOS Temperature-Compensated Mobility-Based Frequency Reference for Wireless Sensor Networks". In: *IEEE Journal of Solid-State Circuits* 46.7 (July 2011), pp. 1544–1552. ISSN: 0018-9200. DOI: 10.1109/JSSC.2011.2143630.

[56] A. Paidimarri, D. Griffith, A. Wang, G. Burra, and A. P. Chandrakasan. "An RC Oscillator With Comparator Offset Cancellation". In: *IEEE Journal of Solid-State Circuits* PP.99 (2016), pp. 1–12. ISSN: 0018-9200. DOI: 10.1109/JSSC.2016.2559508.

[57] D. Griffith, P. T. Røine, J. Murdock, and R. Smith. "17.8 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)*. Feb. 2014, pp. 300–301. DOI: 10.1109/ISSCC.2014.6757443.

[58] Y. C. Shih and B. Otis. "An On-Chip Tunable Frequency Generator for Crystal-Less Low-Power WBAN Radio". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 60.4 (Apr. 2013), pp. 187–191. ISSN: 1549-7747. DOI: 10.1109/TCSII.2013.2251938.

[59] X. Zhang, I. Mukhopadhyay, R. Dokania, and A. B. Apsel. "A 46-uW Self-Calibrated Gigahertz VCO for Low-Power Radios". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 58.12 (Dec. 2011), pp. 847–851. ISSN: 1549-7747. DOI: 10.1109/TCSII.2011.2172530.

[60] J. Ayers, N. Panitantum, K. Mayaram, and T. S. Fiez. "A 2.4GHz wireless transceiver with 0.95nJ/b link energy for multi-hop battery-freewireless sensor networks". In: *2010 Symposium on VLSI Circuits*. June 2010, pp. 29–30. DOI: 10.1109/VLSIC.2010.5560257.

[61] J. Tan, W. S. Liew, C. H. Heng, and Y. Lian. "A 2.4 GHz ULP Reconfigurable Asymmetric Transceiver for Single-Chip Wireless Neural Recording IC". In: *IEEE Transactions on Biomedical Circuits and Systems* 8.4 (Aug. 2014), pp. 497–509. ISSN: 1932-4545. DOI: 10.1109/TBCAS.2013.2290533.

[62] C. Bryant and H. Sjöland. "A 2.45GHz, 50uW wake-up receiver front-end with -88dBm sensitivity and 250kbps data rate". In: *ESSCIRC 2014 - 40th European Solid State Circuits Conference (ESSCIRC)*. Sept. 2014, pp. 235–238. DOI: 10.1109/ESSCIRC.2014.6942065.

[63] H. Milosiu, F. Oehler, M. Eppel, D. Früehsorger, and T. Thöenes. "A 7-uW 2.4-GHz wake-up receiver with -80 dBm sensitivity and high co-channel interferer tolerance". In: *2015 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*. Jan. 2015, pp. 35–37. DOI: 10.1109/WISNET.2015.7127409.

[64] N. M. Pletcher, S. Gambini, and J. M. Rabaey. "A 2GHz 52 μW Wake-Up Receiver with -72dBm Sensitivity Using Uncertain-IF Architecture". In: *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. Feb. 2008, pp. 524–633. DOI: 10.1109/ISSCC.2008.4523288.

[65] N. Saputra and J. R. Long. "A Fully-Integrated, Short-Range, Low Data Rate FM-UWB Transmitter in 90 nm CMOS". In: *IEEE Journal of Solid-State Circuits* 46.7 (July 2011), pp. 1627–1635. ISSN: 0018-9200. DOI: 10.1109/JSSC.2011.2144050.

[66] V. Kopta, D. Barras, and C. C. Enz. "A 420 uW, 4 GHz approximate zero IF FM-UWB receiver for short-range communications". In: *2016 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. May 2016, pp. 218–221. DOI: 10.1109/RFIC.2016.7508290.

[67] B. Razavi. "A study of phase noise in CMOS oscillators". In: *IEEE Journal of Solid-State Circuits* 31.3 (Mar. 1996), pp. 331–343. ISSN: 0018-9200. DOI: 10.1109/4.494195.

[68] A. Hajimiri, S. Limotyrakis, and T. H. Lee. "Jitter and phase noise in ring oscillators". In: *IEEE Journal of Solid-State Circuits* 34.6 (June 1999), pp. 790–804. ISSN: 0018-9200. DOI: 10.1109/4.766813.

[69] L. B. Mercer. "1/f frequency noise effects on self-heterodyne linewidth measurements". In: *Journal of Lightwave Technology* 9.4 (Apr. 1991), pp. 485–493. ISSN: 0733-8724. DOI: 10.1109/50.76663.

[70] S. Gronemeyer and A. McBride. "MSK and Offset QPSK Modulation". In: *IEEE Transactions on Communications* 24.8 (Aug. 1976), pp. 809–820. ISSN: 0090-6778. DOI: 10.1109/TCOM.1976.1093392.

[71] D. K. Weaver. "A Third Method of Generation and Detection of Single-Sideband Signals". In: *Proceedings of the IRE* 44.12 (Dec. 1956), pp. 1703–1705. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1956.275061.

[72] Murali Krishna Parekkat and Deepti Das Krishna. "On the Visualization of the Weaver's 'Third Method' for SSB Generation". In: *IETE Journal of Education* 56.1 (Jan. 2015), pp. 28–33. ISSN: null. DOI: 10.1080/09747338.2015.1050075. (Visited on 04/18/2017).

[73] H. Milosiu, F. Oehler, M. Eppel, D. Frühsorger, S. Lensing, G. Popken, and T. Thönes. "A 3-uW 868-MHz wake-up receiver with -83 dBm sensitivity and scalable data rate". In: *2013 Proceedings of the ESSCIRC (ESSCIRC)*. Sept. 2013, pp. 387–390. DOI: 10.1109/ESSCIRC.2013.6649154.

[74] A. M. Mehta and K. S. J. Pister. "Frequency offset compensation for crystal-free 802.15.4 communication". In: *The 2011 International Conference on Advanced Technologies for Communications (ATC 2011)*. Aug. 2011, pp. 45–47. DOI: 10.1109/ATC.2011.6027432.

[75] Lee Jae-Seung, Kim Joo-Myoung, Lee Jae-Sup, Han Seok-Kyun, and Lee Sang-Gug. "13.1 A 227pJ/b #x2212;83dBm 2.4GHz multi-channel OOK receiver adopting receiver-based FLL". In: *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*. Feb. 2015, pp. 1–3. DOI: 10.1109/ISSCC.2015.7063012.

[76] Agilent Technolgies Company. "Agilent Technologies E5500A/B Phase Noise Measurement System User's Guide". In: *Part number E5500-90004 Revision A.01.05* (2000), pp. 1–118. URL: http://literature.cdn.keysight.com/litweb/pdf/E5505-90003.pdf.

[77] Hewlett-Packard Company. "Residual Phase Noise and AM Noise Measurement Techniques". In: *HP Part No. 03048-90059 Edition A.0.0* (Nov. 1994), pp. 1–118. URL: http://hpmemoryproject.org/an/pdf/HP-3048A_cal_man.pdf;%20pp.%2070-191.

[78] Hewlett-Packard Company. "Phase Noise Characterization of Microwave Oscillators". In: *Product Note 11729C-2* (1985), pp. 1–54. URL: http://hpmemoryproject.org/an/pdf/pn11729C-2.pdf.

[79] Francisco Alegria. "Jitter and Phase Noise". In: *7th Summer School on ADC and DAC Metrology* (July 2007), pp. 1–95. URL: http://lesim1.ing.unisannio.it/tc4/summer2007/summer06_eng/Links/alegria/Jitter%20and%20Phase%20Noise%20-%20Alegria%20-15jun07.pdf.

[80] Keysight Technologies. "Phase Noise Measurement Application Measurement Guide". In: *Manual Part Number N9068-90002* (Feb. 2016), pp. 1–51. URL: literature.cdn.keysight.com/litweb/pdf/N9068-90002.pdf.

[81] Terry Decker and Bob Temple. "CHOOSING A PHASE NOISE MEASUREMENT TECHNIQUE: Concepts and Implementation". In: *RF & Microwave Measurement Symposium and Exhibition* (1985 [estimated]), pp. 1–63. URL: http://hparchive.com/seminar_notes/a-129.pdf.

[82] Hewlett-Packard Company. "RF & Microwave Phase Noise Measurement Seminar". In: (June 1985), pp. 1–124. URL: http://www.hparchive.com/seminar_notes/HP_PN_seminar.pdf.

[83] Algie L. Lance, Wendell D. Seal, and Frederik Labaar. "Chapter 7: Phase Noise and AM Noise Measurements in the Frequency Domain". In: *Infrared and Millimeter Waves Vol. 11* (1984), pp. 239–289. URL: https://tf.nist.gov/general/tn1337/Tn190.pdf.

[84] Keysight Technologies. "Phase Noise Measurement Solutions Selection Guide". In: *Manual Part Number 5990-5729EN* (Jan. 2018), pp. 1–24. URL: https://literature.cdn.keysight.com/litweb/pdf/5990-5729EN.pdf.

[85] J.H. Shoaf, D. Halford, and A.S. Risley. "Frequency Stability Specification and Measurement: High Frequency and Microwave Signals". In: *National Bureau of Standards Technical Note 632* (1973), pp. 1–70. URL: https://tf.nist.gov/general/pdf/47.pdf.

[86] F. Ramian. "Time Domain Oscillator Stability Measurement Allan variance". In: *Rhode & Schwarz Application Note 1EF69_E4* (Apr. 2005), pp. 1–16. URL: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ef69/1EF69_4e_TD_Osc_Stability_Allan.pdf.

[87] Kay Gheen (Agilent Technologies). "Phase Noise Analysis and Measurement". In: (2012), pp. 1–48. URL: https://www.keysight.com/upload/cmc_upload/All/PhaseNoise_webcast_19Jul12.pdf.

[88] Cadence Design Systems Inc. "Virtuoso Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide". In: *Product version 11.1* (Sept. 2011), p. 297.

[89] Berkeley Design Automation Inc. "Analog FastSPICE Platform Reference Manual". In: (Mar. 2014), p. 492. URL: http://www.berkeley-da.com.

[90] Samuel R. Stein. "Chapter 12: Frequency and Time – Their Measurement and Characterization". In: *Precision Frequency Control Volume 2: Oscillators and Standards* (1985), pp. 192–231.

[91] John Miles. "Timelab v1.30 beta". In: *ke5fx.com* (Jan. 2018). URL: http://www.miles.io/TimePod_5330A_user_manual.pdf.

[92] A. Demir, A. Mehrotra, and J. 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 (May 2000), pp. 655–674. ISSN: 1057-7122. DOI: 10.1109/81.847872.

[93] J.A. McNeill. "Jitter in ring oscillators". In: *IEEE Journal of Solid-State Circuits* 32.6 (June 1997), pp. 870–879. ISSN: 0018-9200. DOI: 10.1109/4.585289.

[94] C. Samori, A. L. Lacaita, A. Zanchi, and F. Pizzolato. "Experimental verification of the link between timing jitter and phase noise". In: *Electronics Letters* 34.21 (Oct. 1998), pp. 2024–2025. ISSN: 0013-5194. DOI: 10.1049/el:19981419.

[95] Cadence Design Systems Inc. "Jitter Measurements Using SpectreRF Application Note". In: *Product Version MMSIM6.0* (Mar. 2006), pp. 1–51. URL: http://www.3721rd.com/imgqa/smqh/me/me-3304w0tta32fht4.pdf.

[96] Wenzel Associates Inc. "Allan Variance from Phase Noise". In: *Spreadsheets/Programs* (Mar. 1998). URL: http://www.wenzel.com/wp-content/uploads/allanvariance_zip-1.zip.

[97] Sam Wetterlin Electronics. "Conversion of Phase Noise to Jitter". In: (Apr. 2006). URL: http://www.wetterlin.org/sam/SA/PhaseNoiseToJitterConversion_OldExcelVersion.xls.

[98] JitterLabs, LLC. "Spreadsheet Phase-noise Calculator". In: *Calculators* (Dec. 2004). URL: https://www.jitterlabs.com/support/calculators/PN_Spreadsheet_Template_Excel97_v1p0.xls.

[99] Valon Technology. "PN TO JITTER". In: *Phase Noise and Jitter* (Jan. 2011). URL: http://valontechnology.com/5007/PN_TO_JITTER.xls.

[100] X. Vilajosana, Qin Wang, F. Chraim, T. Watteyne, Tengfei Chang, and K.S.J. Pister. "A Realistic Energy Consumption Model for TSCH Networks". In: *Sensors Journal, IEEE* 14.2 (Feb. 2014), pp. 482–489. ISSN: 1530-437X. DOI: 10.1109/JSEN.2013.2285411.

[101] Agilent Technologies. "Performing Bluetooth(tm) RF Measurements Today". In: *Application Note 1333, 5968-7746E* (Mar. 2001), pp. 1–36. URL: http://literature.cdn.keysight.com/litweb/pdf/5968-7746E.pdf.

[102] Rohde & Schwarz. "Accurately measuring drift on Bluetooth(tm) transmitter modules". In: *News from Rhode & Schwarz, Number 169 (2000/IV)* (2000). [Accessed 18-Sep-2018], pp. 1–2. URL: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_news_from_rs/169/n169_bluetooth.pdf.

[103] Aeroflex. "Bluetooth(r) (BR+EDR+LE) Measurement Suite". In: *Part No. 46891/456, Issue 1* (Sept. 2011), pp. 1–4. URL: http://www.aeroflex-cdip.ru/pdf/05/05_002/104_Bluetooth_iss1.pdf.

[104] M. Babaie, F. W. Kuo, H. N. R. Chen, L. C. Cho, C. P. Jou, F. L. Hsueh, M. Shahmohammadi, and R. B. 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* PP.99 (2016), pp. 1–19. ISSN: 0018-9200. DOI: 10.1109/JSSC.2016.2551738.

[105] Texas Instruments. "CC2541: 2.4-GHz Bluetooth low energy and Proprietary System-on-Chip". In: *SWRS110B* (Aug. 2012), pp. 1–33. URL: http://www.ti.com/lit/ds/symlink/cc2541.pdf.

[106] Texas Instruments. "CC2538: A Powerful System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN and ZigBee Applications". In: *SWRS096A* (Apr. 2013). [Accessed 29-Dec-2013], pp. 1–26. URL: http://www.ti.com/lit/ds/symlink/cc2538.pdf.

[107] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer". In: *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)* (Apr. 2012), pp. 1–225. DOI: 10.1109/IEEESTD.2012.6185525.

[108] SiTime Corporation. "SiT1532 Datasheet Rev. 1.26". In: *SiTime 32.768kHz Oscillators* (Jan. 2018), pp. 1–12. URL: https://www.sitime.com/datasheet/SiT1532.

[109] D. C. Burnett, H. M. Fahad, L. Lee, F. Maksimovic, B. Wheeler, O. Khan, A. Javey, and K. S. J. Pister. "Two-chip wireless H_2S gas sensor system requiring zero additional electronic components". In: *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*. To appear. June 2019.

[110] Wei Gao, Sam Emaminejad, Hnin Yin Yin Nyein, Samyuktha Challa, Kevin Chen, Austin Peck, Hossain M. Fahad, Hiroki Ota, Hiroshi Shiraki, Daisuke Kiriya, Der-Hsien Lien, George A. Brooks, Ronald W. Davis, and Ali Javey. "Fully integrated wearable sensor arrays for multiplexed *in situ* perspiration analysis". en. In: *Nature* 529.7587 (Jan. 2016), pp. 509–514. ISSN: 1476-4687. DOI: 10.1038/nature16521. URL: https://www.nature.com/articles/nature16521.

[111] National Institute for Occupational Safety and Health. "NIOSH pocket guide to chemical hazards". In: *U.S. Dept. of Health and Human Services (NIOSH) Publication No. 2005-149* (Sept. 2007), p. 160. URL: https://www.cdc.gov/niosh/npg/default.html.

[112] Y. Shi, M. Choi, Z. Li, Z. Luo, G. Kim, Z. Foo, H. Kim, D. D. Wentzloff, and D. Blaauw. "A 10 mm^3 Inductive Coupling Radio for Syringe-Implantable Smart Sensor Nodes". In: *IEEE Journal of Solid-State Circuits* 51.11 (Nov. 2016), pp. 2570–2583. ISSN: 0018-9200. DOI: 10.1109/JSSC.2016.2606162.

[113] Hossain Mohammad Fahad, Hiroshi Shiraki, Matin Amani, Chuchu Zhang, Vivek Srinivas Hebbar, Wei Gao, Hiroki Ota, Mark Hettick, Daisuke Kiriya, Yu-Ze Chen, Yu-Lun Chueh, and Ali Javey. "Room temperature multiplexed gas sensing using chemical-sensitive 3.5-nm-thin silicon transistors". In: *Science Advances* 3.3 (2017). DOI: 10.1126/sciadv.1602557. URL: http://advances.sciencemag.org/content/3/3/e1602557.

[114] Hossain M. Fahad, Niharika Gupta, Rui Han, Sujay B. Desai, and Ali Javey. "Highly Sensitive Bulk Silicon Chemical Sensors with Sub-5 nm Thin Charge Inversion Layers". In: *ACS Nano* 12.3 (Mar. 2018), pp. 2948–2954. ISSN: 1936-0851. DOI: 10.1021/acsnano.8b00580. URL: https://doi.org/10.1021/acsnano.8b00580 (visited on 12/03/2018).

[115] M.D. Scott, B.E. Boser, and K.S.J. Pister. "An ultralow-energy ADC for Smart Dust". In: *IEEE Journal of Solid-State Circuits* 38.7 (July 2003), pp. 1123–1129. ISSN: 0018-9200. DOI: 10.1109/JSSC.2003.813296.

[116] R. Kapusta, J. Shen, S. Decker, H. Li, E. Ibaragi, and H. Zhu. "A 14b 80 MS/s SAR ADC With 73.6 dB SNDR in 65 nm CMOS". In: *IEEE Journal of Solid-State Circuits* 48.12 (Dec. 2013), pp. 3059–3066. ISSN: 0018-9200. DOI: 10.1109/JSSC.2013.2274113.

[117] P. J. A. Harpe, C. Zhou, Y. Bi, N. P. van der Meijs, X. Wang, K. Philips, G. Dolmans, and H. de Groot. "A 26 mu W 8 bit 10 MS/s Asynchronous SAR ADC for Low Energy Radios". In: *IEEE Journal of Solid-State Circuits* 46.7 (July 2011), pp. 1585–1595. ISSN: 0018-9200. DOI: 10.1109/JSSC.2011.2143870.

[118] F. Maksimovic, B. Wheeler, D. C. Burnett, O. Khan, S. Mesri, I. Suciu, L. Lee, A. Moreno, A. Sundararajan, B. Zhou, R. Zoll, A. Ng, T. Chang, X. Villajosana, T. Watteyne, A. Niknejad, and K. S. J. Pister. "A Crystal-Free Single-Chip Micro Mote with Integrated 802.15.4 Compatible Transceiver, sub-mW BLE Compatible Beacon Transmitter, and Cortex M0". In: *2019 Symposium on VLSI Circuits*. To appear. June 2019.

[119] H. M. Fahad. In: *Review* (2019).

[120] J. Lindenmann, V. Matzi, N. Neuboeck, B. Ratzenhofer-Komenda, A. Maier, and F.-M. Smolle-Juettner. "Severe hydrogen sulphide poisoning treated with 4-dimethylaminophenol and hyperbaric oxygen". In: *Diving and Hyperbaric Medicine* 40.4 (2010). PMID: 23111938, pp. 213–217.

[121] Mallika Bariya, Hnin Yin Yin Nyein, and Ali Javey. "Wearable sweat sensors". In: *Nature Electronics* 1.3 (Mar. 2018), p. 160. ISSN: 2520-1131. DOI: 10.1038/s41928-018-0043-y. URL: https://www.nature.com/articles/s41928-018-0043-y (visited on 01/07/2019).

[122] Margaret E. Payne, Alla Zamarayeva, Veronika I. Pister, Natasha A.D. Yamamoto, and Ana Claudia Arias. "Printed, Flexible Lactate Sensors: Design Considerations Before Performing On-Body Measurements". In: *Preparation* (2019).

[123] Nordic Semiconductor. "RADIO block diagram". In: *nRF51 Series Reference Manual, Version 3.0.1* (Dec. 2016), p. 81. URL: http://infocenter.nordicsemi.com/pdf/nRF51_RM_v3.0.pdf.

[124] Andrew Waterman, Yunsup Lee, David A. Patterson, and Krste Asanović. *The RISC-V Instruction Set Manual, Volume I: Base User-Level ISA*. Tech. rep. UCB/EECS-2011-62. EECS Department, University of California, Berkeley, May 2011. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-62.html.

[125] Krste Asanović, Rimas Avizienis, Jonathan Bachrach, Scott Beamer, David Biancolin, Christopher Celio, Henry Cook, Daniel Dabbelt, John Hauser, Adam Izraelevitz, Sagar Karandikar, Ben Keller, Donggyu Kim, John Koenig, Yunsup Lee, Eric Love, Martin Maas, Albert Magyar, Howard Mao, Miquel Moreto, Albert Ou, David A. Patterson, Brian Richards, Colin Schmidt, Stephen Twigg, Huy Vo, and Andrew Waterman. *The Rocket Chip Generator*. Tech. rep. UCB/EECS-2016-17. EECS Department, University of California, Berkeley, Apr. 2016. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html.

[126] SiFive Inc. "Freedom E300 Platform Reference Manual, Version 1.0.1". In: *Freedom SoC Documentation* (Dec. 2016), pp. 1–58. URL: https://www.sifive.com/documentation/freedom-soc/freedom-e300-platform-reference-manual/.

[127] Jonathan Bachrach, Huy Vo, Brian Richards, Yunsup Lee, Andrew Waterman, Rimas Avižienis, John Wawrzynek, and Krste Asanović. "Chisel: constructing hardware in a scala embedded language". In: *Proceedings of the 49th Annual Design Automation Conference*. ACM. 2012, pp. 1216–1225.

[128] N. Planes, O. Weber, V. Barral, S. Haendler, D. Noblet, D. Croain, M. Bocat, P. O. Sassoulas, X. Federspiel, A. Cros, A. Bajolet, E. Richard, B. Dumont, P. Perreau, D. Petit, D. Golanski, C. Fenouillet-Béranger, N. Guillot, M. Rafik, V. Huard, S. Puget, X. Montagner, M. A. Jaud, O. Rozeau, O. Saxod, F. Wacquant, F. Monsieur, D. Barge, L. Pinzelli, M. Mellier, F. Boeuf, F. Arnaud, and M. Haond. "28nm FDSOI technology platform for high-speed low-voltage digital applications". In: *2012 Symposium on VLSI Technology (VLSIT)*. June 2012, pp. 133–134. DOI: 10.1109/VLSIT.2012.6242497.

[129] M. Lont, D. Milosevic, G. Dolmans, and A. H. M. van Roermund. "Implications of I/Q Imbalance, Phase Noise and Noise Figure for SNR and BER of FSK Receivers". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 60.8 (Aug. 2013), pp. 2187–2198. ISSN: 1549-8328. DOI: 10.1109/TCSI.2013.2239151.

[130] J Taylor. "The JT65 communications protocol". In: *QEX* 232 (2005), pp. 3–12.

[131] Richard Yu-Kuwan Su and Kristofer Pister. "Towards a Synthesizable Standard-Cell Radio". PhD thesis. EECS Department, University of California, Berkeley, Dec. 2012. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-219.html.

[132] B. A. Floyd, Chih-Ming Hung, and K. K. O. "Intra-chip wireless interconnect for clock distribution implemented with integrated antennas, receivers, and transmitters". In: *IEEE Journal of Solid-State Circuits* 37.5 (May 2002), pp. 543–552. ISSN: 0018-9200. DOI: 10.1109/4.997846.

[133] R. W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29.2 (Apr. 1950), pp. 147–160. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[134] Joe Crop. "Adjustable Frequency Divider". In: *OpenCores.org* (May 2010). URL: https://opencores.org/projects/freq_div.

[135] S. K. Wadhwa, G. K. Siddhartha, and A. Gaurav. "Zero steady state current power-on-reset circuit with brown-out detector". In: *19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID'06)*. Jan. 2006, pp. 1–6. DOI: 10.1109/VLSID.2006.172.

Appendix A

SCM3 Documentation

This chapter contains SCM3/SCM3B documentation to complement the information found in other students' dissertations. Note that while this document strives to be accurate, the authoritative source of specific details is the set of original design files. Components mentioned in sparse detail are likely described more fully in the PhD dissertations of either B. Wheeler [6] or F. Maksimovic [5], or the Master's thesis of S. Mesri [7].

A.1 Background

The single-chip mote v3 and beyond has a dozen voltage domains, programming methods, signal banks, and other elements supporting operation of the chip. Understanding these supporting features and associated terminology is necessary to understand SCM3 documentation.

Voltage domains

Voltage domains are as follows; all voltages are generated by LDOs powered by VBAT, designed by various students and depend on bandgap references designed by F. Maksimovic. TYpical operation of LC tank-based SCM3 or SCM3B requires a single VBAT supply. Like the term "SCμM3" (or scm_v3, SCM3, scum3, etc.) itself, and spelling of these domains vary across documentation and design files but their Hamming distances [133] are hopefully always far enough to be unambiguous.

External power sources:

- **VBAT**: Battery voltage from which all other voltage domains are powered. Also voltage of ESD ring around chip. Nominal voltage is 1.2V-1.6v but fab documents indicate core (thin oxide) transistors have been tested up to 1.8V with lifetime degrading beyond. Our chips have survived exposure to 2.0V in lab. A re-design ensuring all transistors that touch the outside world are thick oxide (nominal 3.3V; fab documents describe

tests up to 4.0V) would make the chip more likely to survive accidental over-voltage events.

- **VDDIO**: Sets VDD for GPIO buffers. Made from thick oxide devices so can be comfortably powered at 3.3V to allow interface with, e.g., commercial sensor packages. No LDO; supplied off-chip.
- **VDD25**: Nominally 2.5V supply to power IF input/output debug chain. Not needed for regular operation. No LDO; supplied off-chip.

Internal power supplies powered by VBAT. All supplies are off by default besides VDDD, alwayson, and aux_digital:

- **VDDD aka vdd_digital aka VDDD_LDO_OUTPUT**: Powers ARM Cortex M0, 802.15.4 MAC state machine, RFTimer, plus necessary start-up functions like analog scanchain source select mux, system clock, system clock divider, and RFTimer divider. In SCM3, starts up at 1.2V (max voltage) on start-up because of a mistake (should have been approx 0.9, or what results when all but the MSB of config interface is set to on). In SCM3B, starts up at 0.8v. Since this is on by default, powering it off is performed externally via the VDDD_DISASBLE pin rather than via scanchain.
- **Vdd_alwayson**: Powers analog scanchain (ASC), optical receiver/programmer, and some fundamental control logic used at start-up. In SCM3B, this voltage nominally starts up at 0.8v and was accidentally not padded out. Since this is on by default, powering it off is performed externally via the VDD_ALWAYSON_DISASBLE pin rather than via scanchain.
- **vdd_aux_digital aka VDD_AUX_LDO_OUTPUT**: Powers non-essential digital cores like the BLE packet assembler/disassembler, counters, most dividers, and GPIO routing. In SCM3, this is off by default but powers essential components needed at power-on. In SCM3B, this is on by default (nominal startup voltage 0.8v) because we were worried we'd forgotten to move all essential components to VDDD.
- **vdd_sensoradc aka VDD_TEMPCORE aka SENSOR_ADC_LDO_OUTPUT**: Powers sensor ADC, PGA, and sensor frontend.
- **VDD_DIV**: LC tank divider power supply.
- **vdd_PA**: RF power amplifier power supply.
- **vdd_IF aka IF_LDO_OUT**: Baseband gain/filter power supply.
- **VDD_LO**: LC tank power supply.

Additional voltage domains/supplies in SCM3 but removed from SCM3B on account of cutting the ring oscillator radio to make space:

- **RFAUXVDD aka vdd_aux_rf**: Powers ring oscillator radio accessories like dividers and buffers in an attempt to keep interference from those components out of the RF/IF communication path (it didn't). No LDO; regulated off-chip.
- **RFVDD0, RFVDD1, RFVDD2, RFVDD3**: Powers each of the four ring oscillators on SCM3: S10, S1, D4, and D8, respectively. No LDO; regulated off-chip.
- **VDD25_RING**: Nominally 2.5V supply to power IF input/output debug chain for ring oscillator radio. Not needed for regular operation. No LDO; supplied off-chip.
- **VDDPA_RING**: Powers RF power amplifier for ring oscillator radio. No LDO; regulated off-chip.
- **IF_LDO_OUT_RING**: Baseband gain/filter power supply for ring oscillator radio.

Programming interfaces

Two methods of getting firmware into the chip are available. This firmware is executed after the hard-coded boot ROM finishes executing and issues a soft reset to start executing from instruction memory. In the original SCM digital design by S. Mesri there are three available boot options available to the `next_imem_mode` register: none, 3wb, and AHB. AHB has yet to be used and was likely added to accommodate wireless over-the-air programming; none does nothing; 3wb has been spliced to accept data from either three dedicated pins or the optical programmer. By default, a set of muxes receive signals from the optical programmer and pass them to the 3wb bootloading interface. The external `BOOTLOAD_SOURCE_SELECT` pin switches these muxes to receive signals from the dedicated pins.

Programming via the dedicated pins follows a typical three-wire bus (SPI-like) system: one pin for data, one pin to clock that data in serially, and one pin to latch the data that's been clocked in. When programming firmware, the latch is asserted every 32 bits. The full instruction memory must be programmed to cause the boot ROM to issue a reset, so all binaries must be zero-padded up to 64kB.

Configuration busses

Several terms get used frequently to refer to ways of getting information in, across, and out of the chip besides the firmware programming methods described above.

- **Analog Scanchain aka ASC**: A serial register 1200 bits long used for internal analog & digital component configuration. Complete register list can be found via the Matlab programming script. The ASC can be commanded from internal firmware via Cortex M0 registers or externally via dedicated pins. By default, signals are received from internal registers but can be switched to external pins via the `ASC_ext_override` pin. Control lines include data in, clock ("phi"), negative clock ("phib"), latch, write strobe

("i0o1"), and data out. Register chain is a string of latches; every other latch is made transparent by phi and the rest are made transparent by phib. Write strobe is not used. See Teensy MCU or SCM firmware code for operational details.

- **Digital Scanchain aka DSC aka "poor-man's JTAG debugger"**: A serial register to read out CPU state registers selected by hand during SCM2 design. Same pin configuration as the ASC, but write strobe is necessary to copy CPU state registers to scanchain before clocking result out. See Teensy MCU or SCM firmware code for operational details.
- **analog_rdata aka internal GPIO inputs**: Register bank used by CPU to read state from other peripherals on the chip with firmware commands. For instance, analog_rdata[31:0] can be queried to read the value of the 32kHz RC timer counter. Some analog_rdata bits are also routed out via GPIO, but this is not the default. The mapping of these bits is indexed in the big SCM3/B spreadsheet.
- **analog_cfg aka internal GPIO outputs**: Register bank used by CPU to write configuration bits to other peripherals on the chip with firmware commands. Some analog_cfg bits are also assertable externally via GPIO or via scanchain; in this case, a scanchain register typically controls which source is routed to the bit's destination. The mapping of these bits is indexed in the big SCM3/B spreadsheet.
- **GPIO aka external GPIO**: 16 dedicated pins on the IC with thick oxide tri-state buffers capable of producing digital outputs or consuming digital inputs at gnd or VDDIO.

Configuration pins

This subsection summarizes SCM configuration pins that may be needed to start up the system correctly. Left alone, these pins have appropriate pull-ups and pull-downs as appropriate such that, by default, the chip powers on in a position with relevant power supplies on (vddd, alwayson, and aux digital), primary and secondary system clocks running, 3wb firmware source set to the optical programmer, and analog scanchain set to accept commands issued via firmware.

These defaults are to accommodate the overall design goal: chip powering up via single VBAT/GND power connection, firmware being programmed via optical, then being ready for full operation without any additional connections or steps. Sometimes those steps go wrong or devices under test need to be debugged step-by-step, in which case those defaults need to be changed. All other defaults are controlled by the analog scanchain, but the settings below are more fundamental than the ASC (e.g., controlling the ASC's power) and therefore need dedicated pins.

- **VDDD_disable**: Pulled high to VBAT by default via 1.4MΩ. Pull down to gnd to switch. Disables VDDD LDO if switched.

- **ASC_ext_override**: Pulled down to gnd via 2.0MΩ by default. Pull up to VDDD to switch. Switches ASC signal sources to dedicated pins instead of firmware-controllable CPU registers and suppresses ASC POR assertion, just in case.
- **BOOTLOAD_SOURCE_SELECT**: Pulled down to gnd via 2.0MΩ by default. Pull up to VDDD to switch. witches 3wb programming signal sources to dedicated pins and disables optical programmer.
- **VDD_ALWAYSON_DISABLE**: Pulled high to VBAT by default via 1.4MΩ. Pull down to gnd to switch. Disables VDD_ALWAYSON LDO if switched.
- **System clock bypass**: Also illustrated in dashed box at bottom of Figure. A.1.

A.2 SCM3B Clock network

The original SCM3 clock network was designed such that various clock sources were routable to appropriate destinations. Testing revealed our predictions of "appropriate" proved too limiting, and often we wished to be able to connect clocks in other ways and had to resort to external wires to do the job. In SCM3B we adopted an "everything-to-everything" architecture, allowing all frequency sources to route to all destinations even when such combinations don't make sense. This universal multiplexer will usually be referred to as a crossbar.

The clock network is illustrated in Figure A.1 and, at time of writing, has been thoroughly tested on FPGA and spot checked on ASIC.

Sources

The frequency sources are as follows:

- **system_clk_pri**: Primary system clock, powered by VDDD and nominally configured to start up on initial power-on at 20MHz. Consumes approximately 1.5uA (too small wrt current baseline to measure).
- **system_clk_sec**: secondary system clock, powered by VDDD and nominally configured to start up on initial power-on at 20MHz. Consumes approximately 15uA (too small wrt current baseline to measure).
- **RC_2MHz**: transmit data clock, a 2MHz RC oscillator consuming approximately 2uA (from simulation) and tunable in 27ppm (54Hz) steps. This oscillator is designed to serve as chip clock at 2MHz for IEEE 802.15.4 or, divided by 2, bit clock at 1MHz for Bluetooth Low-Energy.
- **TIMER32k**: Sleep clock, comprised of a 32kHz oscillator consuming approximately 0.4uA (from simulation) and not tunable. The lack of tunability is by design; one needs

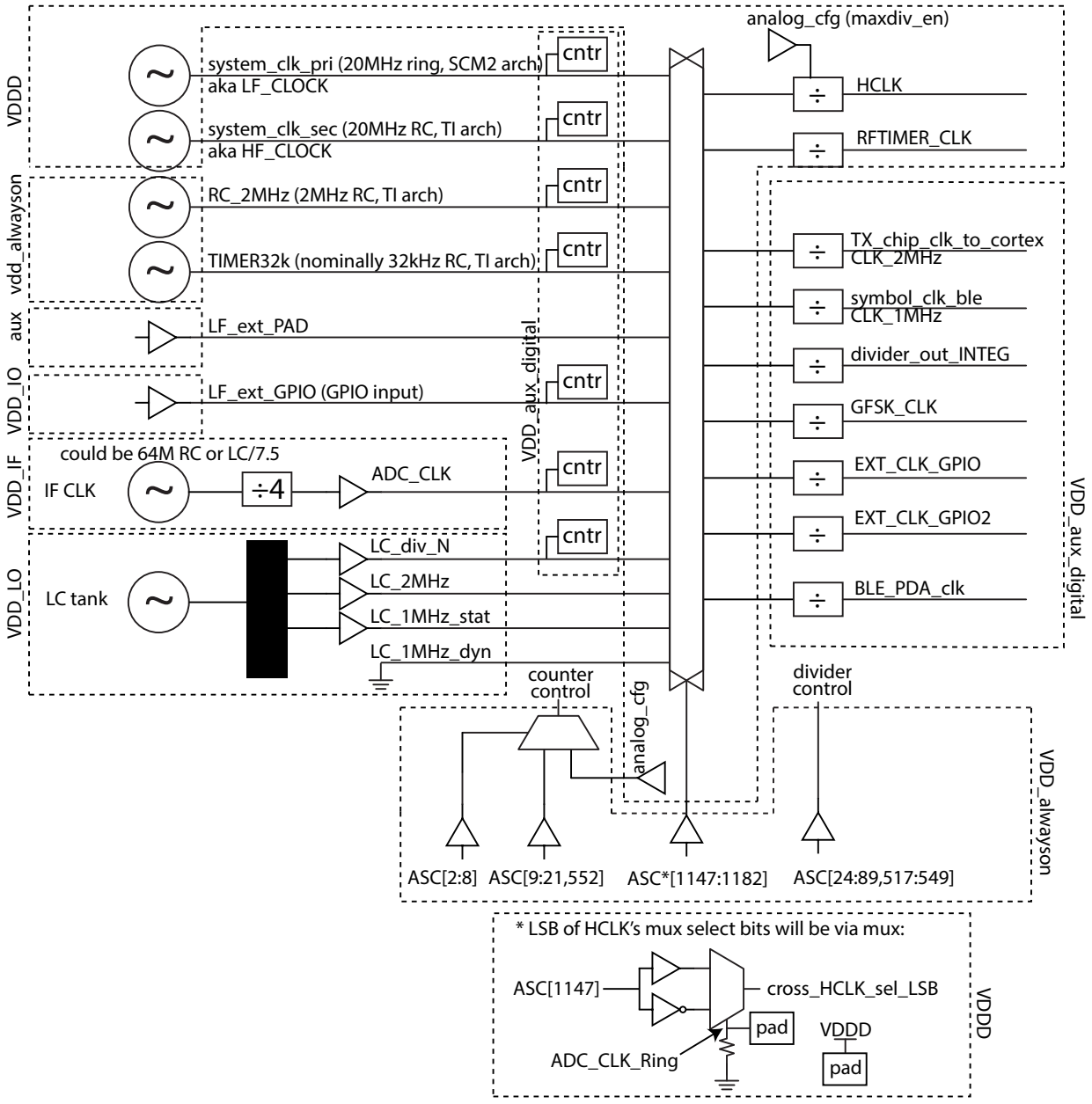


Figure A.1: Diagram of SCM3B crossbar-based clock network with voltage domains and analog scanchain (ASC) control bits annotated.

only measure the clock period during calibration to determine how many timer ticks are necessary to achieve a given sleep duration.

- **LF_ext_PAD**: Clock source input from dedicated pad, the input driver of which is driven by VDDD.
- **LF_ext_GPIO**: Clock source input from GPIO, the input driver of which is driven by VDD_IO.
- **IF_CLK**: Clock source originating in the receiver. Depending on receiver settings, this source could be derived from the LC tank after a divide-by-7.5 block, or a 10000ppm (SCM3) / 1000ppm (SCM3B) tunable 64MHz RC oscillator. This source is divided by 4 to become **ADC_CLK** before entering the clock network.
- **LC Tank** The LC tank divider network is illustrated in Figure 5.22c of [5] and produces three outputs.
  - **LC_div_N**: The LC after an arbitrary integer division.
  - **LC_2MHz**: The LC divided to approximately 2MHz for IEEE 802.15.4 data (chip) clock operations. Exact frequency depends on the channel frequency to which the LC tank is tuned.
  - **LC_1MHz_stat**: The LC divided to approximately 1MHz for Bluetooth LE data (bit) clock operations. Exact frequency depends on the channel frequency to which the LC tank is tuned.
  - The fourth output in the diagram, **LC_1MHz_dyn**, was eliminated between SCM3 and SCM3B and the vestigial clock network input is connected to ground.

Sinks

The frequency sinks are as follows:

- **HCLK**: Cortex CPU clock, called HCLK to match ARM-provided Verilog files. The divider here is default set to 4 to produce a 5MHz HCLK from **system_clk_pri** at startup. The divider producing this output has a special feature: by toggling the analog_cfg register connected to max_div_en, the divider can be set to its maximum divide ratio (255, resulting in a nominal clock frequency of 20MHz/255=78kHz) or back to its original divide ratio via atomic operations. This is to allow quick sleep/wakeup of the CPU.
- **RFTIMER_CLK**: Nominally 500kHz clock to support RF timing operations such as received packet timestamping and time-synchronized packet transmission.
- **TX_chip_clk_to_cortex** aka **CLK_2MHz**: Connects to 802.15.4 MAC to control when to clock out the next chip in the packet.

- **symbol_clk_ble** aka **CLK_1MHz**: Connects to BLE transmit FIFO to control when to clock the next bit in the packet into the GFSK modulator or, if GFSK modulator disabled, directly into the radio.
- **divider_out_INTEG**: Connects to integrator inside ZCC demodulator/CDR block to set rate at which LO feedback code is updated. LO feedback code is connected to fine DAC of ring oscillator radio and, because ring oscillator radio was removed in later revisions, is not connected in SCM3B. Integrator output value is available over software for further experimentation.
- **GFSK_CLK**: Connects to GFSK modulator, running at nominally 20MHz, to provide timing for 3-bit steps to approximate Gaussian FSK shaping.
- **EXT_CLK_GPIO**: Clock for external component clocking via GPIO.
- **EXT_CLK_GPIO2**: Clock for external component clocking via GPIO.
- **BLE_PDA_clk**: Connects to BLE packet assembler and disassembler blocks to clock gates.

Dividers

All dividers are simple 8-bit integer dividers written in Verilog and based on [134]. They are controlled by the analog scanchain. Each divider has one output: `out`, which is the divided frequency source. Outputs of all dividers are available via GPIO for debugging purposes. The divider has several inputs, and in SCM3B the polarities have been adapted to enable expected operation at startup when the analog scanchain initializes to all-0 output. This adaptation was attempted in SCM3 but had several errors, so below are the input specifications for SCM3B.

- `in`: Rising edge-triggered frequency input. Experiments at VDDD=1V have shown, for a digital block synthesized with maximum speed 20MHz, that 22MHz is the fastest input frequency before errors (cycle slipping) are observed.
- `Nin`: Divide ratio set by 8-bit word. All inputs inverted such that when the analog scanchain initializes, all dividers are set to maximum divide ratio and outputting minimum frequency to minimize startup power consumption. For HCLK divider only, `Nin` is inverted such that at analog scanchain all-0 startup, divide ratio is 4.
- `reset`: Asynchronously holds divider in reset for as long as reset is set to 1.
- `enable`: Synchronously allows divide operation to occur. Set to 0 to enable.

- `maxdiv_en`: Synchronously sets divide ratio to maximum (255) if set to 1, regardless of `Nin`, to put CPU into a lower-power state by clocking slower. At 42uA/MHz, expected CPU current consumption in 5MHz/255=78kHz low-clock mode is 3.3uA, but this is too small to experimentally determine vs. SCM3's ~200uA baseline current. Can be toggled via analog_cfg[14] to switch via atomic memory-mapped IO operation or ASC(145) if scanchain programming is desired (select with ASC(144); see scanchain script for details). Enabled only on HCLK divider and not available on other dividers.
- `passthrough_en`: Directly connects `in` to `out` if set to 1. Overrides all above settings including `enable` and `reset`.

Counters

The counters (labeled as "cntr" in Figure A.1) are very simple 32-bit counters that increment by 1 when their input sees a rising edge. They are enabled by setting `enable` to 1 and reset by setting `resetn` to 0, such that they are disabled and reset on power-on and the analog scanchain is set to all 0s. Counter output is a 32-bit word available simultaneously from all counters in software via analog_rdata registers. The lower 16 bits of any 1 counter are also available via GPIO.

Glitch/blip behavior has been observed on counter output for reasons as yet unknown. In an attempt to fix this issue, the enable signal was made registered in SCM3B to synchronize the enable signal, issued from the CPU clock domain, with the clock domain of the input frequency.

External HCLK source switch

At the bottom of Figure A.1 the HCLK source switch is illustrated. Two pads in the upper-left of SCM3B will, if connected together, switch the input of the pictured mux. The output of this mux is the LSB of the HCLK divider's source selection code. Connecting these two pads by, e.g., wirebonding, will overpower the pull-down resistor and change the HCLK divider input to **system_clk_sec** (input code `4'b0001`) instead of **system_clk_pri** (input code `4'b0000`).

A.3 Power-on reset (POR)

Power supplies never switch on instantaneously, so the power-on reset (POR) block is responsible for holding reset pin of the system in a reset state (low) while the power supply ramps up to an acceptable value. The SCuM POR topology is based on [135] and redesigned to include LVT/HVT devices as appropriate. It has been simulated across PVT, mismatch, and a very wide selection of VDD ramp rates without issue and we have yet to observe a startup failure in silicon. The original design included a resistive DAC to adjust turn-on voltage but, in practice, tuning these bits was never needed and were tied high in later silicon

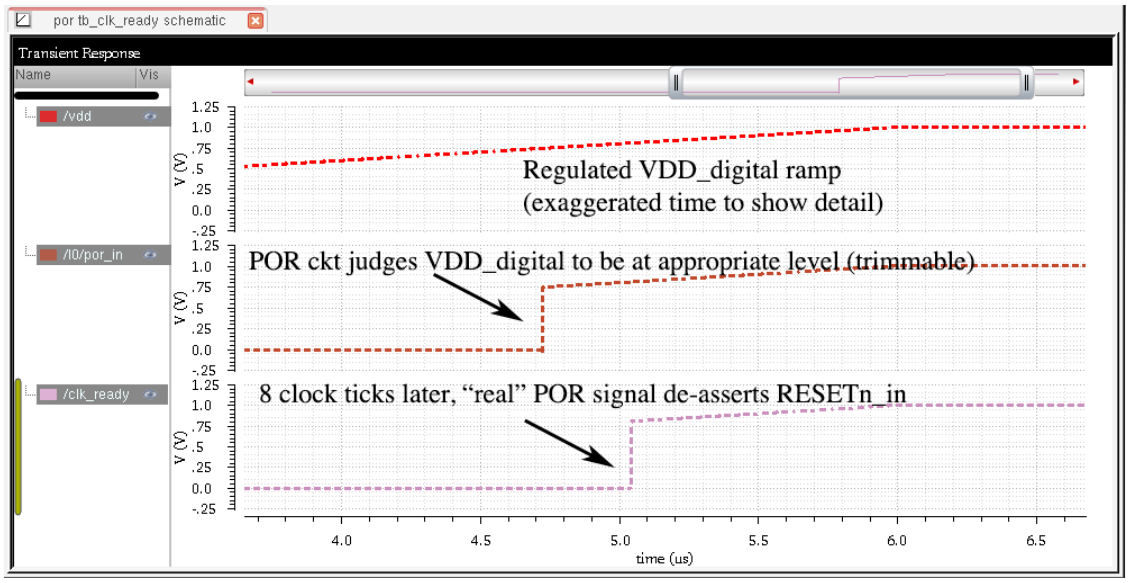


Figure A.2: POR input and output waveforms and clock ready signal. Note this plot shows clock ready after 8 clock ticks, whereas SCM-3B waits for 128 ticks.

iterations. Furthermore, due to the low-level nature of this block, the effects of adjusting this resistive DAC were difficult to observe: by the time we could see output, the POR had already fired. Further experimental investigation was never undertaken because of the observed reliability of this circuit and the lack of innovation that wold have resulted in a publishable result. (This block, like a large percentage of the work that went in to SCuM, was support engineering instead of novel research.)

This POR block appears in three locations on SCuM-3B:

1. Tied to vdd_alwayson and in control of analog scanchain (ASC) reset
2. Tied to vdd_digital and in control of CPU reset
3. Tied to VDD_IF and in control of the IF gain and filter controllers.

The brown-out detection circuit described in [135] is include in locations 1 and 2 above but, practically, it isn't necessary. In simulation, it only seems to add value in situations where the voltage rail falls fast enough such that the reservoir capacitor doesn't bleed off. We don't encounter this situation in our tests, where power is either on or off.

Location 2 also includes a "clock ready" signal. The POR fires which releases reset on a counter. The counter is incremented by the CPU clock (aka HCLK, a divided-down fraction of the system clock). After 128 counts, another reset output is released, releasing CPU system reset via Section A.4. This process is illustrated in Figure A.2.

We anticipate future iterations of SCuM will include regulation from scavenged sources (e.g., a DC/DC converter), may need to make use of brown-out detection, and could benefit

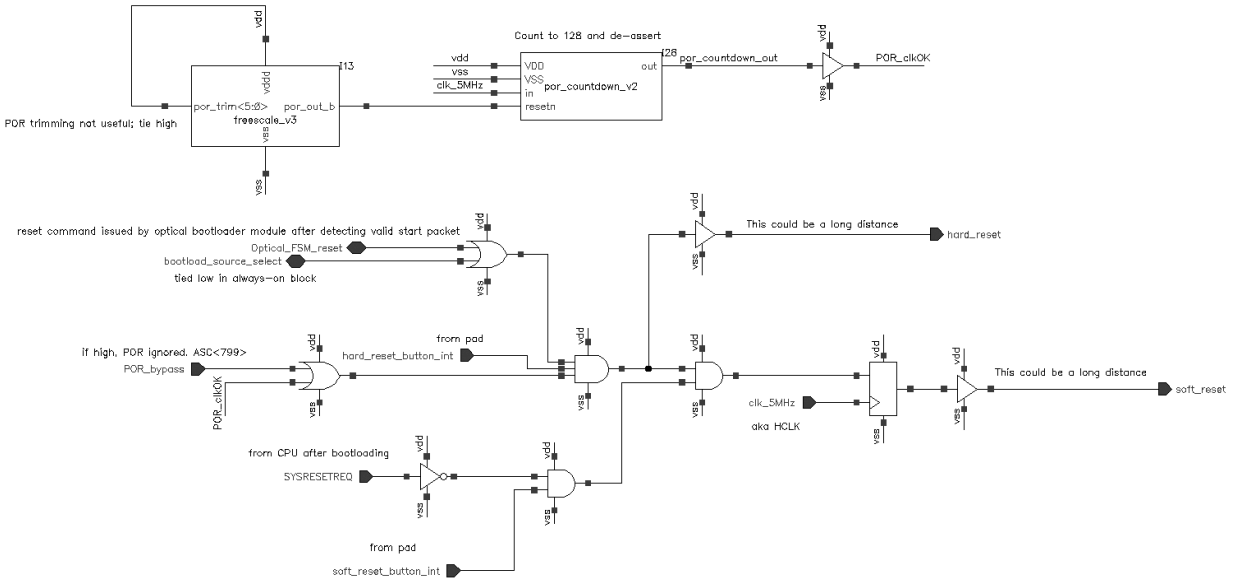


Figure A.3: Reset logic including POR and clock ready blocks as implemented in SCM-3B.

from adjustable power-on reset voltage via resistive DAC. In that case, this block should be examined and likely re-designed to suit.

A.4 Reset logic

A reset logic network was laid out to combine various POR, external hard reset, external soft reset, and internal self-reset (aka SYSRESETREQ) signals as specified by Figure 3.8 in [7] and illustrated in Figure A.3. This block includes the ability to block or disable potentially misbehaving signals (e.g., the first fabrication of the POR block) via analog scanchain or external pull-up/down.

This block has remained unchanged since SCM-2 and was laid out by hand to make absolutely sure reset signals were handled correctly. Though it has been reliable it has some less-ideal features, such as reset signals needing to be routed long distances out of the digital core and back into it after only going through a few gates. To alleviate this and integrate reset into digital synthesis, Brad and I each wrote our own reset logic modules in Verilog and verified them on FPGA. Both behave identically to each other and to the original reset logic system, and are likely safe to integrate to simply future SCM versions.

Appendix B

General lab advice

General lab advice:

1. Every time an array is appended to in Matlab, it allocates a new portion of memory the size of the new array, copies the old into the new, then deletes the old one. This can mean a 10x slowdown when dealing with arrays larger than a few tens of thousands of elements. Heed Matlab's recommendation to "preallocate for speed" by allocating the whole array up front. (This could've made my work in Chapter 3 take days instead of weeks.)

2. USB3 is 10x faster than USB2 but only if the underlying device can run that fast. Flash drive write speeds can also differ by 10x − what a cheap flash drive does in over a half hour, a fast one can do in 4 minutes.

3. Explore output file options. A 1GB CSV can be represented as an 84MB HDF5. (This is how I was able to tune and re-take the data in Chapter 6 in only a few days after making major system improvements.)