Towards Very Large Scale Analog (VLSA): Synthesizable Frequency Generation Circuits

by

Muhammad Faisal

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2014

Doctoral Committee:

Associate Professor David D. Wentzloff, Chair
Professor Michael P. Flynn
Yorgos Palaskas, Intel Corporation
Associate Professor Sara A. Pozzi

To my family and friends

ACKNOWLEDGEMENTS

I'd like to start off by thanking my advisor, David Wentzloff, who has been a great coach and mentor throughout my PhD. Throughout the past five years, the extent of his guidance extended beyond just setting the direction of my Ph.D research. Not too many advisors can race their first year students down the sleeping bear dunes – sorry about your hamstring by the way. I would also like to thank Prof. Michael Flynn, Prof. Sara Pozzi & Prof. (☺) Yorgos Palaskas for serving on my Ph.D committee and providing valuable suggestions to refine further improve my thesis.

I must give credit to all the educators who inspired me to pursue my ambitions, academic or otherwise. In particular, Professor Adel Sedra, who's is the reason I am a circuit designer today, and to this day he remains a strong advocate of me and my work – instilling confidence in me. He used to tell us that circuits are good for our soul, something I hope to understand someday. I must also thank Mark Spera, my high school chemistry teacher, who once said, "Muhammad, choose your path carefully. You've got what it takes to achieve anything in life. It better be the right thing". His words echo in my head to this day.

I also want to thank the entire Radio Integration Lab at Intel. Their encouragement and genuine interest in my research motivated me to push forward in this direction.

I want to thank my parents, Razzaq and Sajida, for their unconditional love and support, my siblings, Umair, Waqas, Saima, Shaista and my brothers in law, Asif and Azam, for their constant encouragement and curiosity about my school.

I'd like to thank the past and present members of the WICS mafia who enriched my life in more ways than I can imagine. In particular, Nathan Roberts, who has been a friend, colleague and a football buddy since the first of grad school. Osama Khan, who didn't have to think twice before helping me with technical matters and off the field personal situations. Sangwook Han, without whom, I would not have finished my first tapeout. Jonathan Brown, who convinced me to come to the University of Michigan. Kuo-Ken Huang, who was my go-to car guy and helped me pick out two cars. Dae Young Lee, who constantly encouraged me to keep my head up when my first chip wasn't behaving. Seunghyun Oh, who's efficiency and intelligence amazed me every day. Youngmin Park, who blazed the VLSA trails for me. Ryan Rogel, who added a new dimension to the group dynamics. David Moore, who is the greatest protégé on the planet and will go on to do some amazing things. Elnaz Ansari, who taught me Persian dance moves. Hyeongseok Kim, who was quiet but had a lot of interesting stories to share. Yao Shi, whose knowledge of random things surprised me a few times. Michael Kines, who found the level of my corruption quite amusing. Avish Kosari, who has promised to turn me into a violinist someday. Armin Alaghi, the honorary member of WICS group who could probably be a very successful comedian.

I would also like to thank a number of past and present students from the rest of Michigan Integrated Circuits Lab. I'd like to start off with Jorge Pernillo, who enhanced my social life in Ann Arbor. Jeffrey Fredenburg, who makes generating innovative ideas seem effortless. Mo Barangi, who embarrassed me in front of friends and strangers more times than I want to remember. I'd also like to thank Chunyang, Jaehun, Mohammad, Batu, Nick, Hyungil, David, Ben, Bharan, Dave, Zhiyoong, Yoonmyung, Gyuho, Hassan and Greg.

A number of non-UM friends played a crucial role in keeping me sane throughout my PhD. Emily Grekin was my social key and yoga buddy (John's pretty cool too). My Canadian friends, Sunny, Mintu, Yashar, Kundan, Anand, George, Sep and Jorge, made sure that I didn't miss non-Ph.D life too much. Only these guys would travel 4200 kilometers on a whim to hang out with Mo and then spend three days sardined up in an SUV, listening to "the big bad wolf" while fighting and arguing over who gets to take the shot gun seat.

Big thumbs up to the Ann Arbor coffee shops and libation venues for providing the essential fuel!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Driven by advancement in integrated circuit design and fabrication technologies, electronic systems have become ubiquitous. This has been enabled powerful digital design tools that continue to shrink the design cost, time-to-market, and the size of digital circuits. Similarly, the manufacturing cost has been constantly declining for the last four decades due to CMOS scaling. However, analog systems have struggled to keep up with the unprecedented scaling of digital circuits. Even today, the majority of the analog circuit blocks are custom designed, do not scale well, and require long design cycles.

This thesis analyzes the factors responsible for the slow scaling of analog blocks, and presents a new design methodology that bridges the gap between traditional custom analog design and the modern digital design. The proposed methodology is utilized in implementation of the frequency generation circuits – traditionally considered analog systems. Prototypes covering two different applications were implemented. The first synthesized all-digital phase-locked loop was designed for 400-460 MHz MedRadio applications and was fabricated in a 65 nm CMOS process. The second prototype is an ultra-low power, near-threshold 187-500 kHz clock generator for energy harvesting/autonomous applications. Finally, a digitally-controlled oscillator frequency resolution enhancement technique is presented which allows reduction of quantization noise in ADPLLs without introducing spurs.

Chapter 1

Introduction

In the summer of 1948, a tiny electronic device called a transistor was announced at the headquarters of Bell Labs in New York at a press conference. The invention failed to create a buzz and barely received any attention from the local newspaper, the New York Times [1]. Fast forward to October 2013, and the entire world tuned in to watch the launch of the latest generation of the iPhone. The event was streamed live on the internet, and millions of people were able to watch the unveiling of the latest Apple gadgets in the comfort of their homes across the globe. The entire infrastructure that allows such incredible connectedness and the sophistication of electronic gadgets is built on what's called integrated circuits. Even today the smallest unit of electronic circuits is a transistor - precisely the invention that gained little attention from the general public 65 years earlier. Integrated circuits have enabled a stunning transformation of every aspect of human life. They have turned room sized computers and bulky furniture sized radios to hand held devices. The electronic revolution is driven by the unprecedented ability to scale and miniaturize transistors at an incredible pace as encapsulated by Moore's law [2].

1.1. More than Moore

In 1965, Gordon Moore suggested, based on empirical data that the *number of transistors in integrated circuits doubles every two years.* This was later modified by David House to state the chip performance, facilitated by an increased number of transistors and the ability to design that quicker, doubles every 18 months [2]. The semiconductor industry has remarkably followed Moore's law. The deflationary effect of Moore's law has played a crucial role in wide spread proliferation of electronics in every aspect of our lives. The tremendous scale of this deflationary effect can be understood by comparing the cost per transistor. For example, the average price of an integrated transistor was $5.52 in 1954, and it has dropped to about one billionth of a dollar in 2013 [3]. However, the process scaling is running into some fundamental physical limits, and the speed of Moore's law is predicted to slow down by 2020.

The extreme scaling suggested by Moore's law works well for microprocessors, and memories, but not for analog components which interface with the external world. Even “all-digital” systems require some sort of analog components on-chip [3]. For example, an all-digital microprocessor requires a clock generator as well as a power management system, and both of these systems are inherently analog in nature; therefore, do not scale according to Moore's law. However, these analog components add significant value for the consumers. A microprocessor cannot function without power management or the clock generator. As we know, analog circuits scale significantly slower than their digital counterparts. Some empirical evidence shows that the analog circuits double in performance approximately every five years, while digital circuits double in performance every 18 months [4]. Therefore, significant innovation at the block and architectural levels along with a paradigm shift in the design procedure for analog circuits must take place in order to keep up with the growing need for higher performance at a lower cost in consumer electronics.

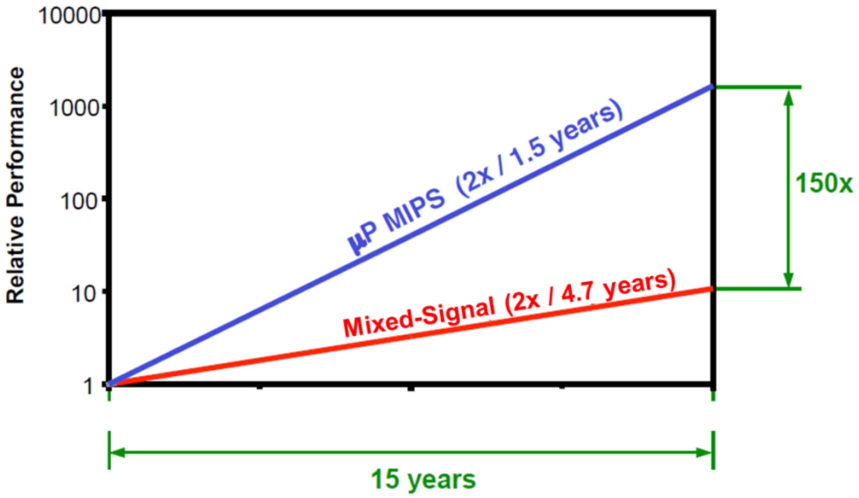


Figure 1.1: Analog vs. Digital scaling [4]

Figure 1.1 shows how the digital domain has followed Moore's law. However, the analog domain has scaled at a much slower pace. A number of reasons can be attributed to the slow scaling of analog circuits [5]:

- Scaling leads to smaller transistors, which have worse control over their current due to second order short channel effects that appear in deep sub-micron technologies. This can be compensated with more sophisticated circuits and calibration techniques which lead to larger area.
- Scaling leads to reduced supply voltage, limiting headroom as well as the dynamic range of various analog blocks. This leads to exacerbated non-linearities in circuits as well as lower voltage operations require a proportionally lower noise levels.
- Passive components such as capacitors and inductors are often required in analog blocks, which have not scaled with process.

Therefore, a strong emphasis has been placed on developing digitally-assisted or all-digital architectures for traditionally analog components. Regardless of relatively slower scaling of analog components, new classes of computing are appearing at a rapid pace as described by Bell's law [4][6].

1.2. Moore to Bell

Bell's Law states that *roughly every decade a new, lower priced computer class forms based on a new programming platform, network, and interface resulting in new usage and the establishment of a new industry*. This has been proven true since the 1960s when the mainframe computer used to be the size of a room then came the era of work station computers in the 1970s. However, computers were still considered a luxury and were reserved for educational and research purposes, which changed with the appearance of Apple Computers and the PC, followed by laptops and now smartphones. This trend has continued to this day, and there are no signs of stopping. As illustrated by Figure 1.2, Bell's Law for the next class of computing seems to be becoming true. Several millimeters scaled computing systems have already been reported [7]. This new class of ubiquitous computing will give birth to what's dubbed as Internet of Things. The University of Michigan is leading the way with its millimeter cubed project as well as its strong research in ultra-low power near threshold digital and analog/RF circuits[7].

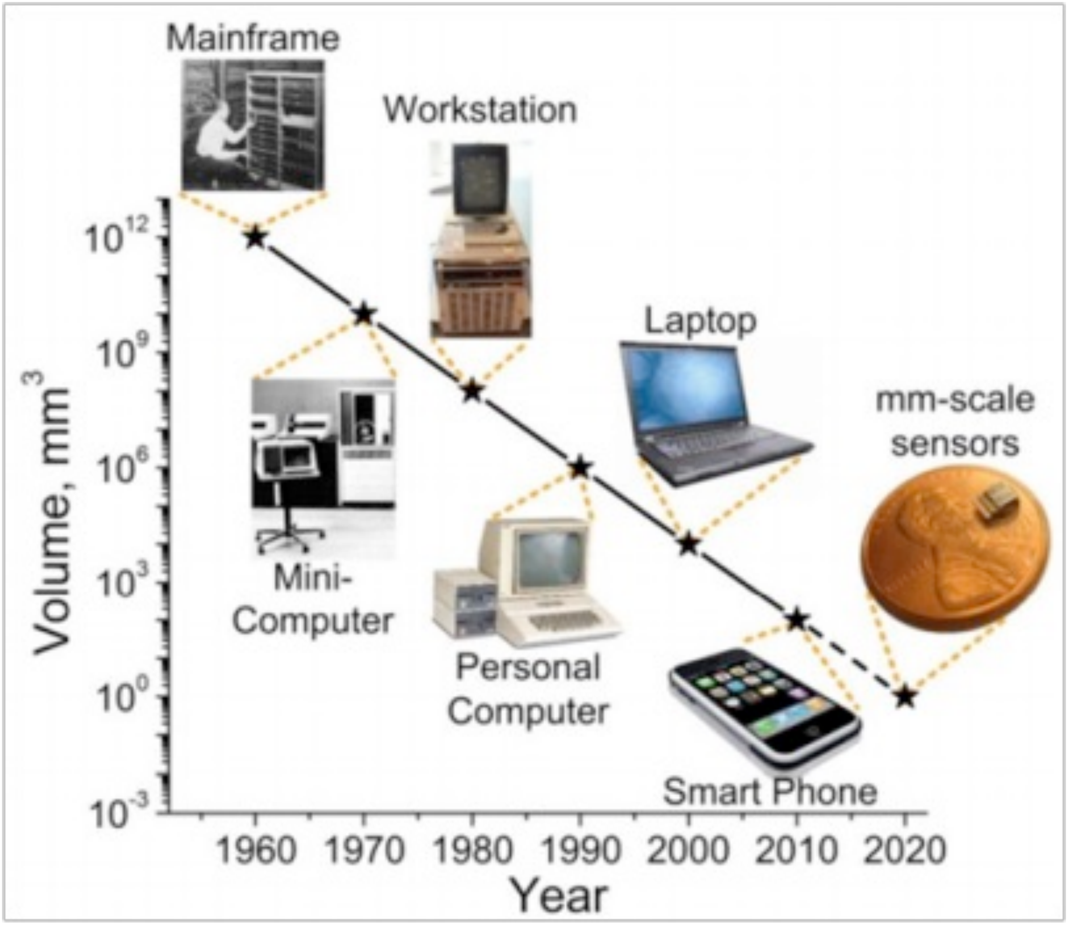
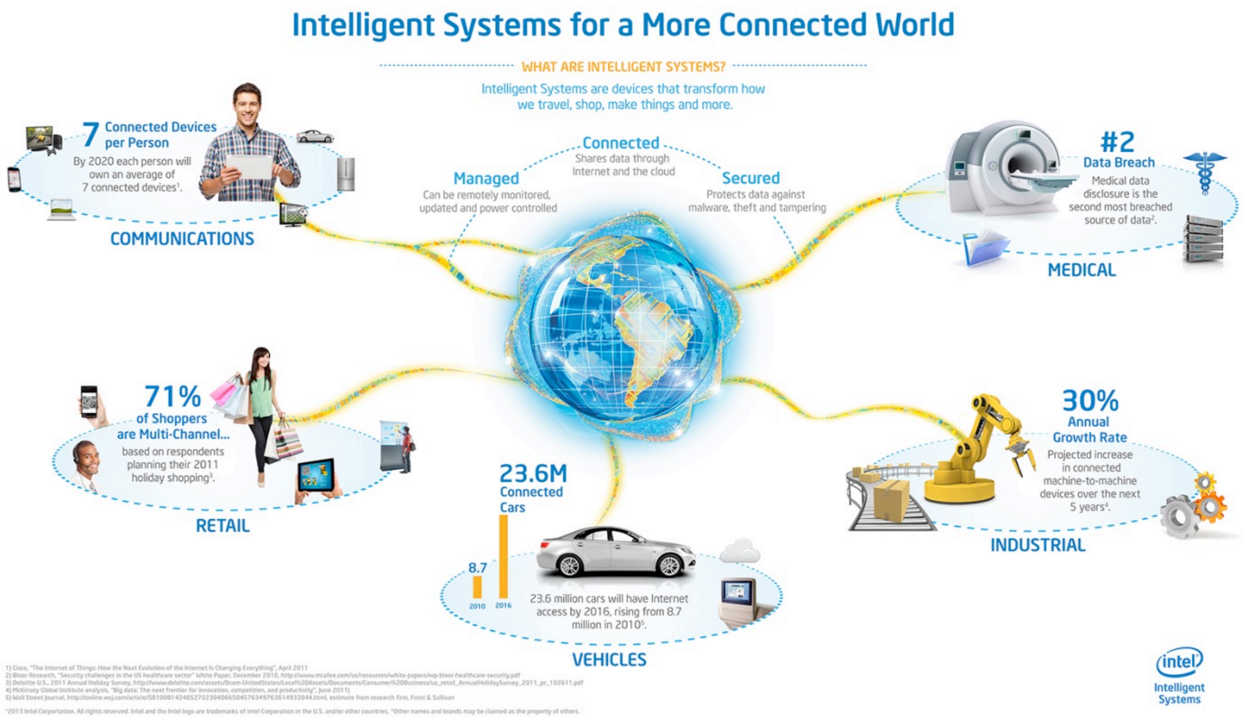


Figure 1.2: Bell's Law [7]

1.3. Internet of Things

The Internet of Things is loosely defined as virtually every object can have the ability to sense, process and communicate information [8]. It has been predicted by many major technological players that the Internet of Things will penetrate every aspect of human life within the current decade. As shown in Figure 1.3, the ability to sense, process and communicate is already affecting our experience of shopping, transportation, communication and health care. It has been repeatedly predicted, as listed below, that the trend of an increased number of connected devices will only continue as the technology progresses to facilitate wide spread proliferation.

- Intel predictions 50 billion connected devices by 2020 [9].
- Cisco predicts 1 trillion connected devices by 2025 [10].
- Bosch predicts that the average person will carry up to 1000 sensors by 2025 [11].


Intelligent Systems for a More Connected World
WHAT ARE INTELLIGENT SYSTEMS?
Intelligent Systems are devices that transform how we travel, shop, make things and more.
7 Connected Devices per Person
By 2020 each person will own an average of 7 connected devices1.
COMMUNICATIONS
Connected
Shares data through Internet and the cloud
Managed
Can be remotely monitored, updated and power controlled
Secured
Protects data against malware, theft and tampering
#2 Data Breach
Medical data disclosure is the second most breached source of data2.
MEDICAL
71% of Shoppers are Multi-Channel...
based on respondents planning their 2011 holiday shopping3.
RETAIL
23.6M Connected Cars
8.7
2010
2016
23.6 million cars will have Internet access by 2016, rising from 8.7 million in 20105.
VEHICLES
30% Annual Growth Rate
Projected increase in connected machine-to-machine devices over the next 5 years4.
INDUSTRIAL
intel
Intelligent Systems

1) Cisco, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything", April 2011
2) Bloor Research, "Security challenges in the US healthcare sector" White Paper, December 2010, http://www.mcafee.com/us/resources/white-papers/wp-bloor-healthcare-security.pdf
3) Deloitte U.S., 2011 Annual Holiday Survey, http://www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/Consumer%20Business/us_retail_AnnualHolidaySurvey_2011_pr_102611.pdf
4) McKinsey Global Institute analysis, "Big data: The next frontier for innovation, competition, and productivity", June 2011)
5) Wall Street Journal, http://online.wsj.com/article/SB10001424052702304066504576349763614933844.html, estimate from research firm, Frost & Sullivan

*2013 Intel Corportation. All rights reserved. Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

Figure 1.3: Internet of Things, Intel's view [9]

Upon closer observation, one can argue that the technical capabilities required to make the IoT a reality already exist. For example, technologies such as wireless communication, embedded processing and sensing already exist. Then why hasn't the IoT become a reality yet?

Though the required technologies exist, significant innovation is still required in order to make the IoT feasible. Figure 1.4 shows the ecosystem and how various different aspects of technology will interact to make up the overall IoT environment. More efficient and cost effective applications, data management and software are required. However, the real bottle neck is still the hardware that cannot yet support the wide spread use of various things due a number of reasons [12]:

- **Power consumption** is too prohibitive in various different components of a computing system. Long battery life or energy autonomy will be required for IoT since changing batteries on 1 trillion devices every day will be an impossible task to achieve.
- Electronics **cost** in its current state is too high. The only way to make 1000 devices per person a reality would be if the ICs in these devices are prices at cents per unit.
- The current state of electronics is too bulky, and significant efforts must be focused on **miniaturization** of existing systems as well as development of new small form factor systems.
- **Quicker design** and manufacturing cycles will be required in order to meet the predicted high demand for the connected devices.

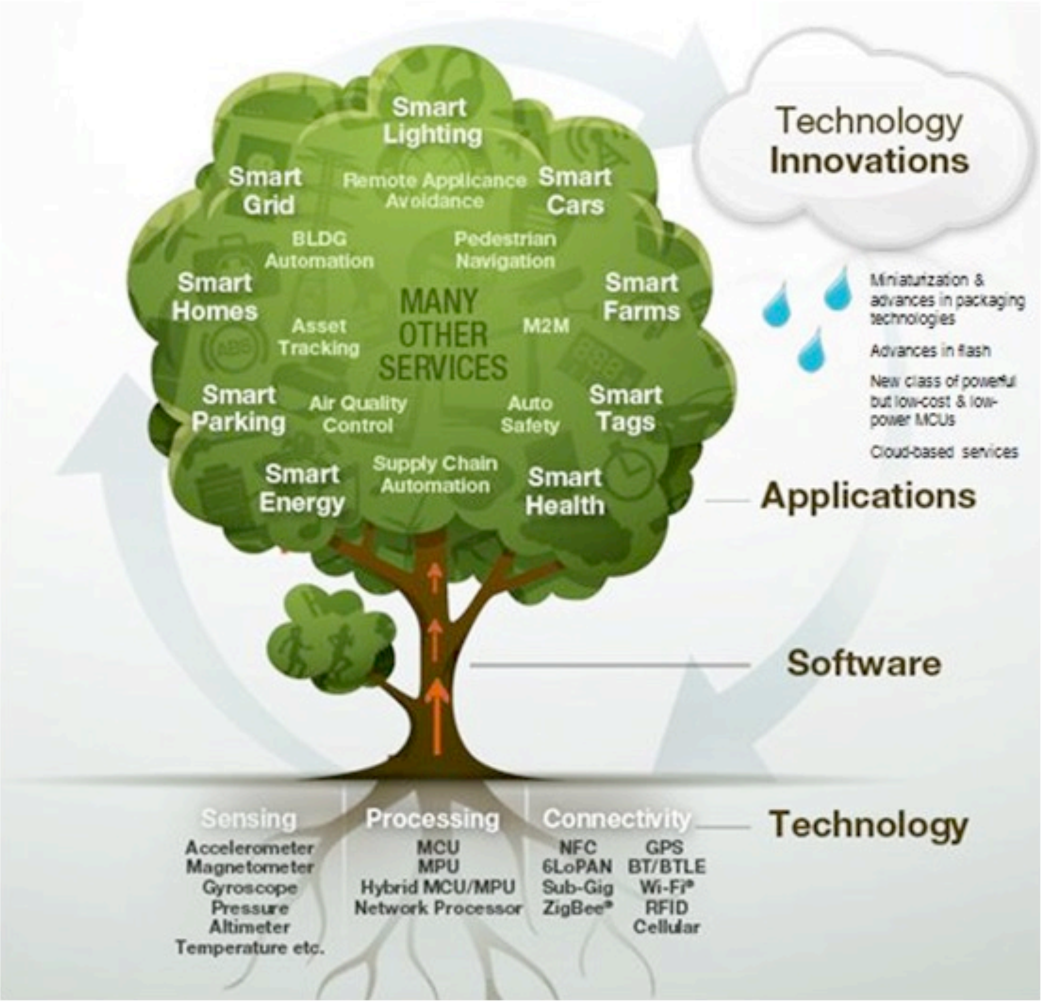


Figure 1.4: Internet of Things ecosystem [12]

One of the key contributions of the presented research is the development of a design methodology as well as innovative architectures to address the issues that are stalling the realization of the Internet of Things. One of the major analog components that can be found virtually in every electronic system is a frequency generation circuit. Frequency generation circuits serve as the heartbeat of every electronic system, and are often based on phase-locked loops. An overview of the phase-locked loop (PLL) applications is given in the following section.

1.4. Applications of Phase-Locked Loops

There are widespread applications of frequency generation circuits. As mentioned earlier, they serve as the heartbeat of electronic systems. They could be used as clock generators for embedded processing applications or local oscillator (LO) generation for radios, for modulation and demodulation. Some examples of applications are given below.

Clock generator:

One of most wide spread applications of frequency generation circuits is for clock generation in digital systems. Typically, the clock generators consist of a reference signal, and an on-chip programmable frequency multiplier block [13]. The frequency multiplier block could either be a phase-locked loop, frequency-locked loop or delay-locked loop based system. The block diagram of a typical PLL-based clock generator is shown in Figure 1.5 below.

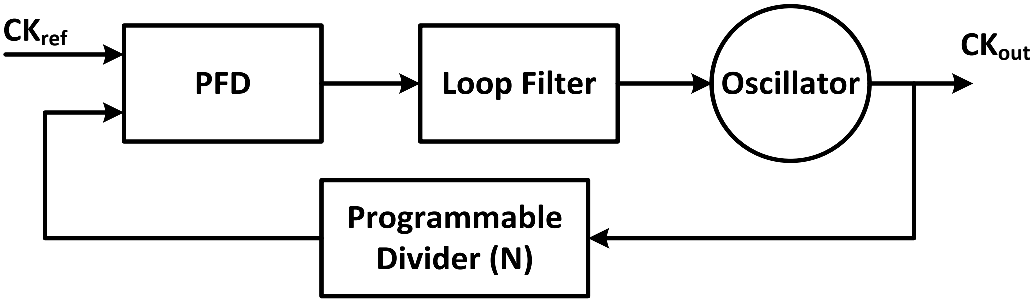


Figure 1.5: Typical PLL-based clock generator

This particular clock generator multiplies the frequency by the divider ratio *N*. Therefore, the output clock frequency is *N* times that of the reference clock.

Clock and Data Recovery:

In many interconnect applications such as USB and VGA, the data is transmitted without an accompanying timing reference signal. For example, in optical communication, information only travels on a single fiber optics channel. However, the receiver must process this data synchronous to some timing reference. Therefore, the receiver must not only recover information but also the clock signal from the received data. Most clock and data recovery systems are based on some sort of frequency generation/phase-locked loop [13]. Figure 1.6 shows a simple block diagram of a typical clock and data recovery circuit.

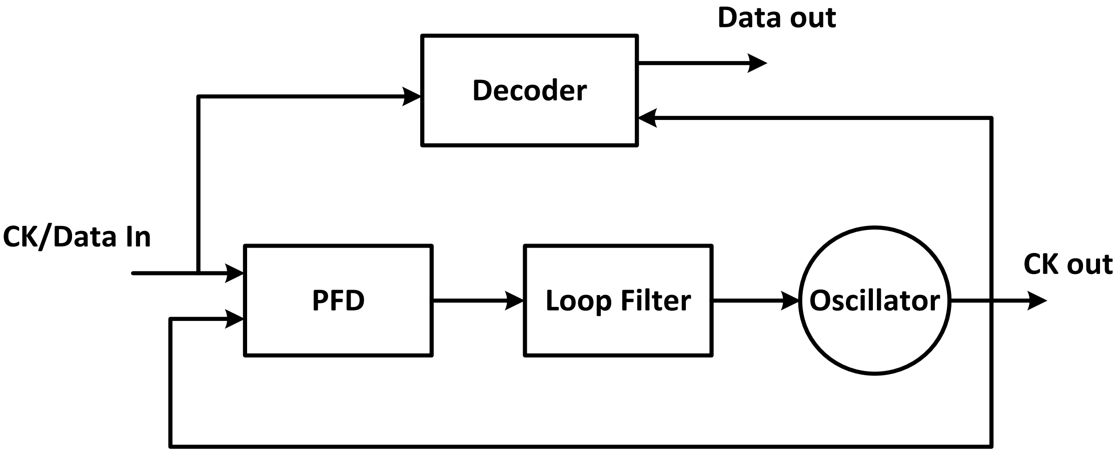


Figure 1.6: A typical clock and data recovery system

Modulation and Demodulation:

When a phase-locked loop is in the lock state, the control of the oscillator (either a voltage or binary number) is proportional to the output frequency. Therefore, by observing the oscillator control, one can demodulate either phase or frequency modulated data. Figure 1.7 shows a typical block diagram of a PLL-based demodulator.

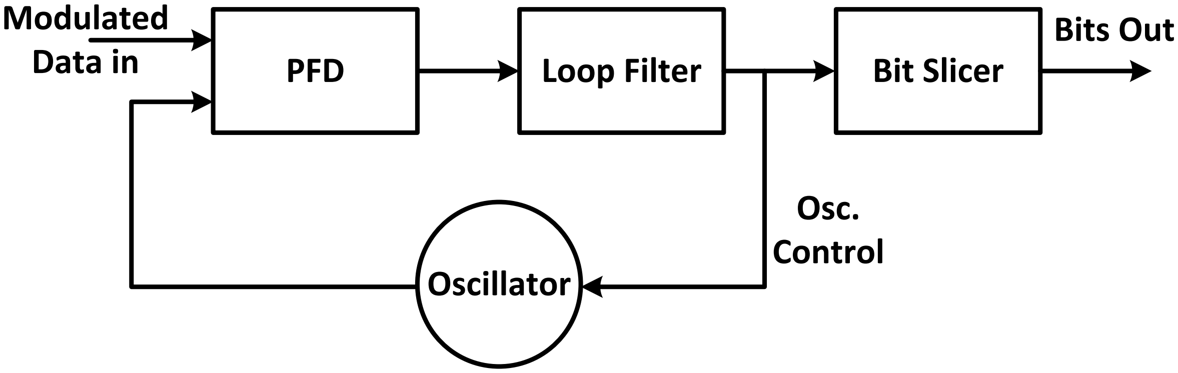


Figure 1.7: A PLL based demodulator

Similarly, the PLL can also be used as a modulator, as illustrated in the Figure 1.8 below.

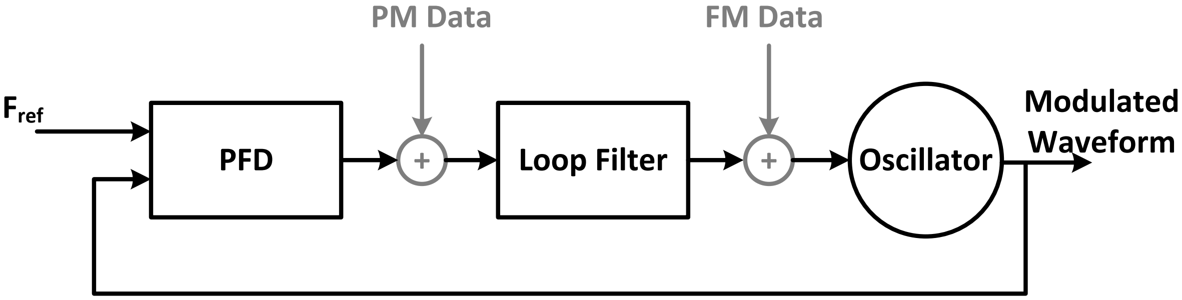


Figure 1.8: A PLL-based demodulator

Phase modulated bits can be fed into the output of the phase frequency detector (PFD), and the output of the oscillator appears to be phase modulated. Similarly, if frequency modulation is required, the data can be directly fed into the control word or control voltage of the oscillator [14].

Local Oscillator Generation for Receivers:

Almost all receiver architectures require a local oscillator (LO) that is used to down convert the signal to either low-IF or baseband frequency which then is digitized and processed in the digital domain. A typical LO generator based on a PLL is often one of the most challenging blocks to design in a receiver. Figure 1.9 and Figure 1.10 show typical architectures of both low IF heterodyne and direct conversion homodyne receivers. LOs are always the crucial part of the receivers, and they are often PLL based [14].

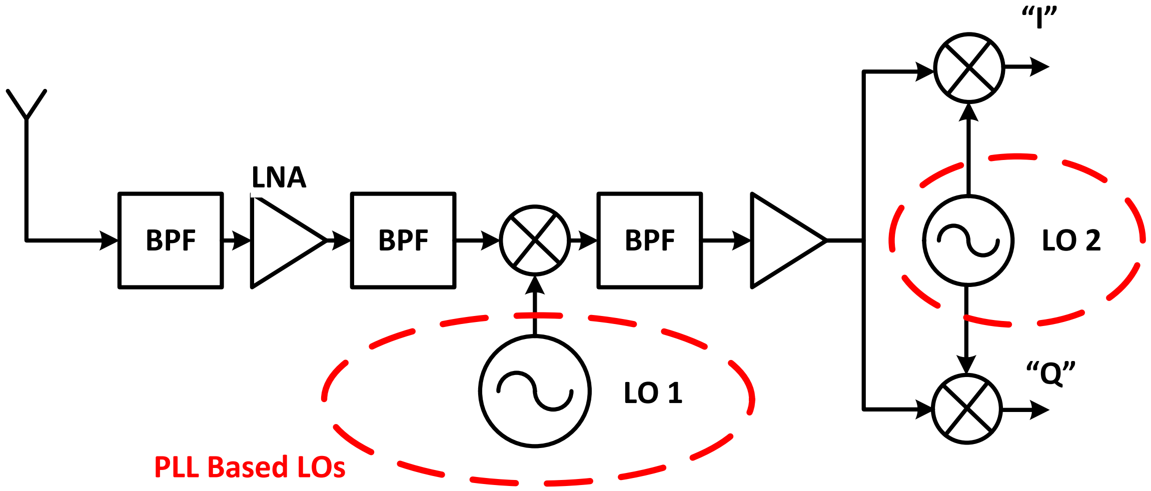


Figure 1.9: A heterodyne receiver requires a PLL to generate LO

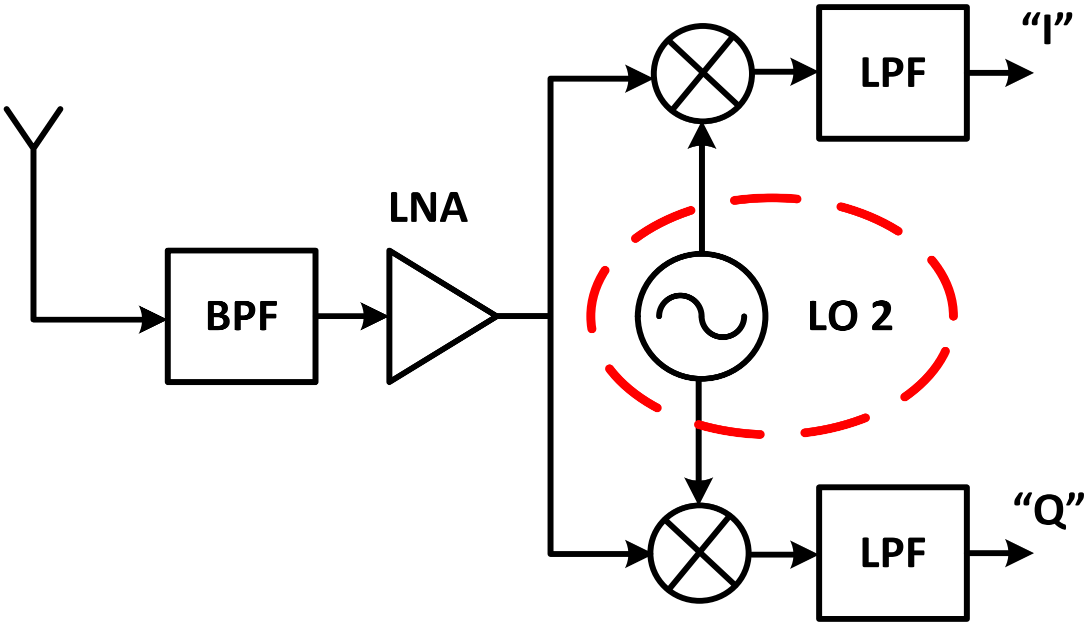


Figure 1.10: A direct conversion receiver with a PLL based LO

Based on the applications discussed above, it is clear that addressing the previously mentioned issues of analog scaling for frequency generation circuits would be a significant contribution towards making IoT a reality [14].

1.5. All-Digital PLLs to Synthesized PLLs

All-digital phase locked loops (ADPLLs) have been the focus of major research efforts since the 90s since they eliminate the need for bulky loop filters and offer increased flexibility and configurability [13]-[19]. However, the overall design of ADPLLs has barely changed since their inception. Figure 1.11 shows a block diagram of a typical ADPLL which consists of a time-to-digital converter (TDC), a digital loop filter (DLF), a frequency divider and a voltage controlled oscillator (VCO) which is typically controlled by a digital to analog converter (DAC) [20].

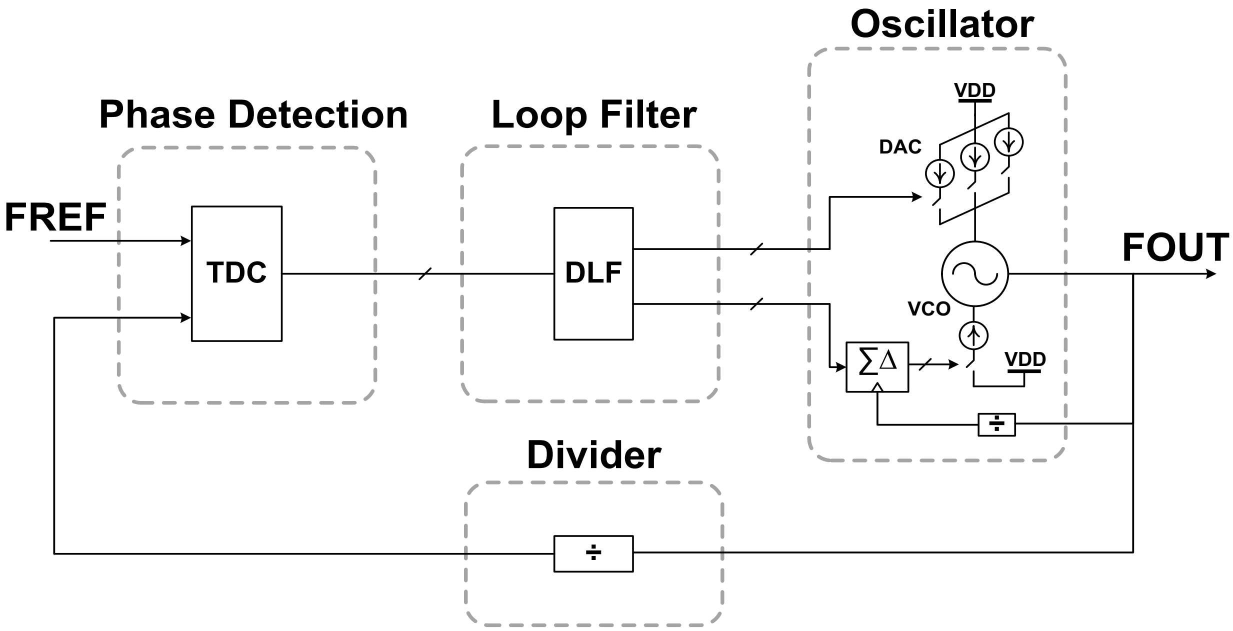


Figure 1.11: Typical ADPLL architecture

Regardless of its classification as an all-digital PLL, a significant amount of analog design goes into the VCO, DAC, and often the divider and TDC. These blocks are all-analog blocks and suffer from the issues highlighted in section 1.1 as process scales, and therefore these ADPLLs do not scale as well as the digital circuits. The next logical step is to implement the ADPLLs using traditional digital design methodologies, with ideally full-swing signals (like digital CMOS signals). This will allow faster design cycles as well as the potential to take advantage of process scaling.

1.6. PLL Design for Internet of Things

From discussions in prior sections, it can be inferred that PLLs designed for IoT applications will be required to address the following issues:

Energy constraint

IoT applications will require either long battery life or energy autonomy. This problem can be addressed from two directions: (i) improving the energy source or (ii) reducing energy consumption. Improving the energy source or batteries is outside of the scope of this research work, and significant research efforts are being focused on power harvesting [16]-[17]. However, power harvesting systems produce very low voltage levels and cannot source large currents that are often required in analog components. Achieving robust operation in voltage- and energy-constrained environments is a major requirement for IoT electronics.

Size

The size of PLLs for IoT applications must be as small as possible because the weight and size of the 10s of connected devices (as predicted) could become prohibitive.

Cost

Cost will play an important role in IoT development. Therefore, PLLs for IoT applications must be as cost effective as possible. All-digital PLL architectures that occupy small silicon area, and do not require external components or expensive bias generation circuitry will be the key.

Accelerated Design

With 1 trillion devices deployed, an unprecedented demand will be placed on new connected devices. Therefore, the ADPLLs designed for IoT must have short design cycles in order to meet the market demand. A design methodology was developed as part of this project to allow synthesis of ADPLLs.

1.7. Thesis Contributions

This thesis builds upon research conducted by Dr. Youngmin Park [18]. Dr. Park's work was mainly focused on utilizing digital standard cells only to implement traditionally analog components. He successfully implemented an ultra-wide band transmitter, time-to-digital converters, and all-digital phase-locked loop [18]. However, this research differs from Dr. Park's contributions in the following ways:

1. ADPLL Performance

This thesis is only concerned with ADPLLs, and explored implementations for ADPLLs for various applications. Dr. Park proved the merits of cell-based design for analog/RF blocks using only digital standard cells. Although the design time can be significantly shortened with utilizing digital standard cells only, it's extremely challenging to match the performance of custom designed ADPPLs. Therefore, extensive calibration techniques are required.

This thesis is focused on bridging the gap between performance and automated cell-based design. Thereby implementing ADPLLs efficiently for high performance applications.

2. ADPLL Architectures

This research also proposes new ADPLL architectures as a means to reduce power consumption, and overcome some of the non-idealities of automated layout. These architectures remove the divider, and use an embedded TDC architecture – resulting in significant power reductions.

3. Low Voltage Applications

This research also extends the cell-based design approach to low-voltage and ultra-low power applications. This was not previously explored in Dr. Park's research.

This thesis explores innovative architectures for ADPLLs, and design methodologies to accelerate design by utilizing existing digital tools, to implement low power PLLs without compromising the performance. The contributions are briefly described as follows.

1. ***Design Methodology for Synthesizable ADPLLs***

   A new design methodology for accelerated design of PLLs was developed in this project. This design methodology utilizes existing digital and analog design flows to implement ADPLLs. The ADPLL designs are portable across process nodes since they are represented by a hardware description language (HDL). Further details of the methodology are provided in Chapter 2 .

2. ***Synthesizable 400-460 MHz ADPLL***

   Secondly, a novel ADPLL architecture that was implemented entirely using digital design flows is presented. This ADPLL was developed for a 400-460MHz frequency range, and for wireless body area network applications. Details of this ADPLL, including a number of architectural contributions, are discussed in Chapter 4 .

3. ***Synthesizable Ultra-low Power Clock Generator***

   In order to prove feasibility of the new design approach in a variety of applications, a near-threshold, ultra-low power clock generator for energy autonomous/harvesting application was designed and implemented. This system is the smallest known clock generator in sub-MHz frequency ranges. Further details are provided in Chapter 5 .

4. ***DCO Resolution Enhancement Technique***

   Lastly, a novel digitally controlled oscillator resolution enhancement technique was developed. This technique allows frequency tuning in DCOs with small steps without a sigma-delta modulator, and do not introduce any undesirable spurious tones at the output of the ADPLL. The circuits that make up this technique can be entirely implemented using digital standard cells, and the digital design flows. This technique linearizes the resolution-power tradeoff which is a hyperbolic relationship in traditional frequency tuning approaches. Further details of this frequency tuning and resolution enhancement technique are discussed in Chapter 3 .

Chapter 2

A Design Methodology for Accelerated ADPLL Design

In the 1970s and early 1980s, any CAD research focused on analog circuit design optimization and automation was considered antiquated and too slow to adjust to the sweeping wave of all things digital. Traditional analog functions were rapidly being implemented with digital circuits, and analog CAD research was considered intellectually insignificant [21]. However, the digital design methodologies progressed at an unprecedented pace, and it became possible to perform logic-to-gate mapping and automatically generate layout from standard cell libraries for complex digital systems such as microprocessors by mid 1980s. This provided a reliable and safe path for designers to quickly go from an idea to silicon implementation for a large array of digital systems [22].

The emergence of application specific integrated circuits (ASICs) in the late 80s exposed the lack of analog synthesis and layout tools which kick-started a new wave of research efforts into analog circuit design tools. In particular, many of the leading integrated circuits research groups around the world began to explore the possibilities of synthesizing analog circuits the same way digital circuits are done. These analog CAD research efforts have resulted in various fragmented analog design assistance tools, many of which have been absorbed by popular CAD tools of today, such as Cadence and Synopsys. However, even after thirty years of research, analog design is still considered “black magic”, and the dream of analog synthesis has yet to be realized [23]. In this chapter, current analog and digital design methodologies are briefly contrasted, various factors that have hindered the realization of an analog synthesis tool are discussed, and finally a design methodology that significantly accelerates design of some traditionally analog systems is proposed.

2.1. Analog vs. Digital Design Flow

In order to do a fair comparison of the current state of analog design with digital design flows, Gajski's chart (or the Y-chart) will be used. Gajski's chart was proposed in 1983 to illustrate various steps and levels of abstraction required for VLSI design [24]. Though it was proposed for digital systems only, it will be adapted for analog circuit design for illustration purposes. The main goal of this chart is to map out the various hierarchical steps that are required in order to effectively solve complex design problems. Figure 2.1 shows the Y-chart for a typical digital system design of present day. The Y-chart consists of three different domains: functional domain, structural domain and physical design. Digital system design begins with functional/behavioral specifications, and after some initial mathematical modeling, the design moves to the structural domain through some mapping method or tool, and finally the design concludes in the physical domain. In addition to the design domains, Gajski also highlighted that there are various layers of abstraction in digital system design – depicted by concentric circles in the Y-chart. Design of complex and large digital systems can be simplified significantly as highly specialized designers are only responsible for one layer of abstraction [25]. For example, a digital systems designer can use well optimized cells for their designs without diving into the details of device sizing etc. Therefore, a top down design process that starts with system level modelling coupled with specialized transistor and cell level design can significantly reduce the design complexity. This also opened the possibilities of automating a large portion of digital design. The shaded area in Figure 2.1 shows the portion of design that's done with CAD tools today. However, transistor and cell level design still requires significant human capital. Typically, foundries provide cells that have been optimized for performance, and from a digital designer's point of view, the design flow is completely automated.

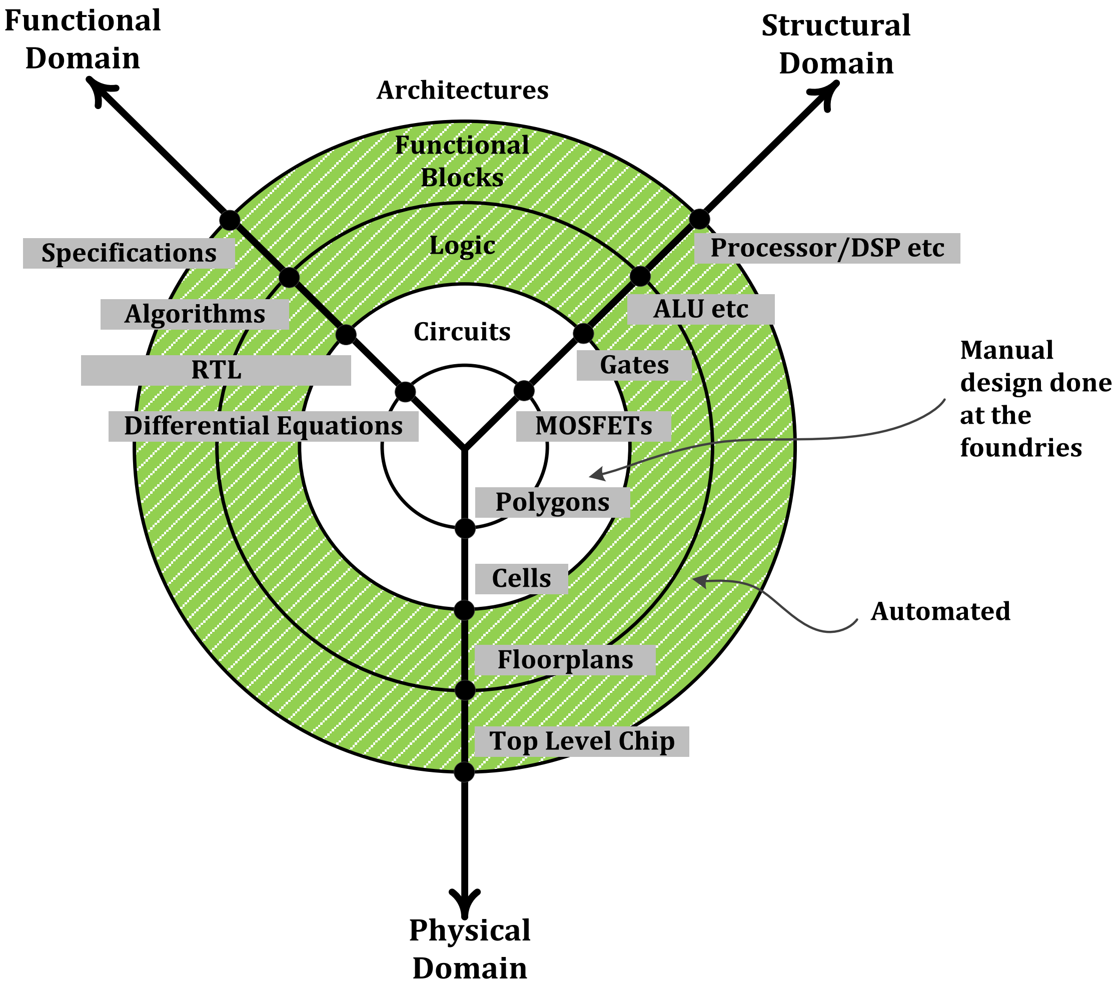


Figure 2.1: Gajski's chart of general purpose VSLI design [25]

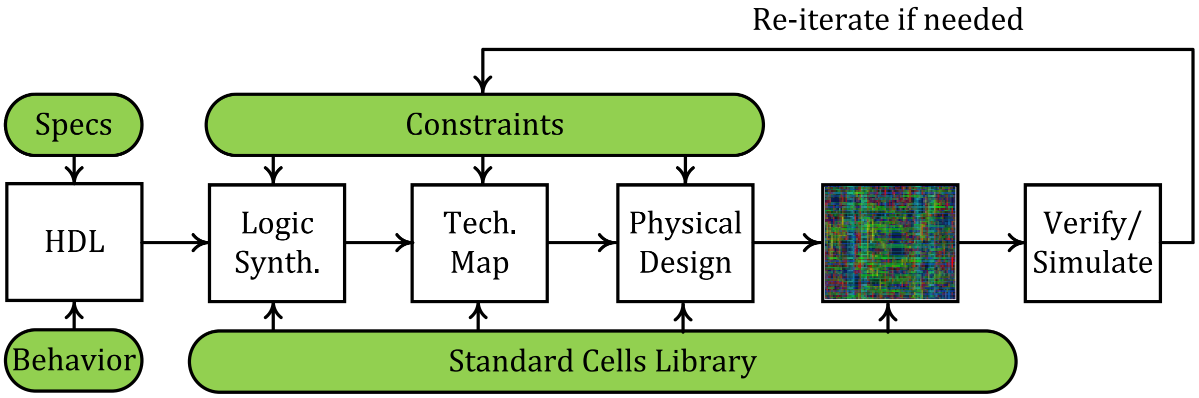


Figure 2.2: Linear representation of digital design flow [26][27]

A more linear representation of today's digital design flow is given in Figure 2.2. The entire process of digital design past the cell design is automated. A digital designer's job is to describe the desired circuit behavior using a hardware description language (HDL) and define the constraints which are typically determined by the performance requirements of the design. The next step after HDL entry is to use the standard cells library, and logic synthesis tools such as Synopsys Design Compiler to convert behavior description to a structural logic description. The final step prior to final verification is the physical design. Current physical design tools have the ability to optimize placement and routing of digital designs to meet specifications [26]-[32].

Similarly, the Y-chart can be used to represent analog design. Figure 2.3 shows the Y-chart for a PLL design. The design of a typical analog system begins with mathematical modeling which helps determine the specifications of the overall system as well as the specifications of the sub-blocks [31][33]. After mathematical modeling, analog designers select circuit topologies as well as types of devices to be used in the structural domain. Extensive simulations are then conducted in order to finalize the device sizes at which point physical design can begin. Analog designs are significantly more susceptible to parasitics and mismatch caused by routing and inefficient placement, and therefore verification at each level of hierarchy is typically required.

One of the main reasons why analog synthesis hasn't yet become a reality is because analog design is often quite recursive and requires simultaneous changes at multiple levels of hierarchy in order to arrive at a robust design. Figure 2.4 shows a typical progression of analog design. Simulations are required at every hierarchical level, and typically the design is tweaked manually by analog designers.

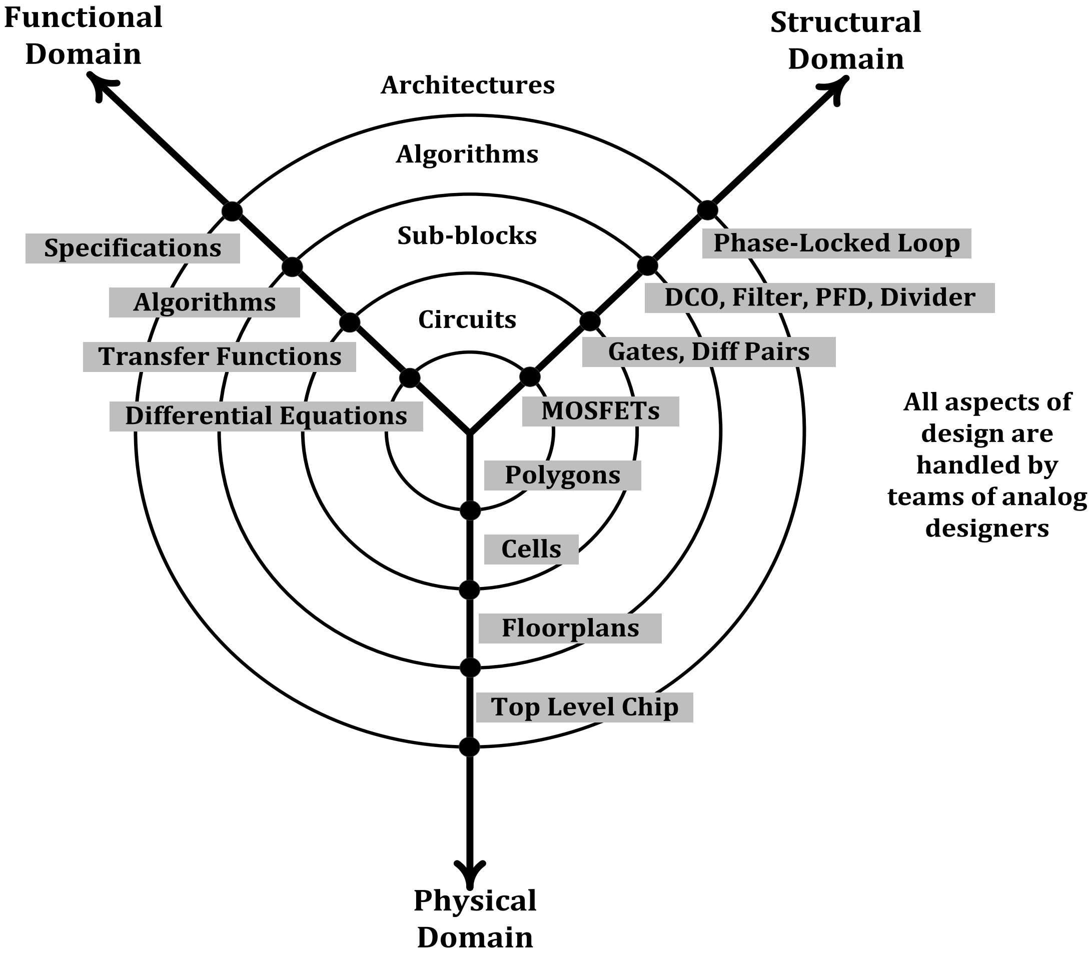


Figure 2.3: Analog design flow Y-chart

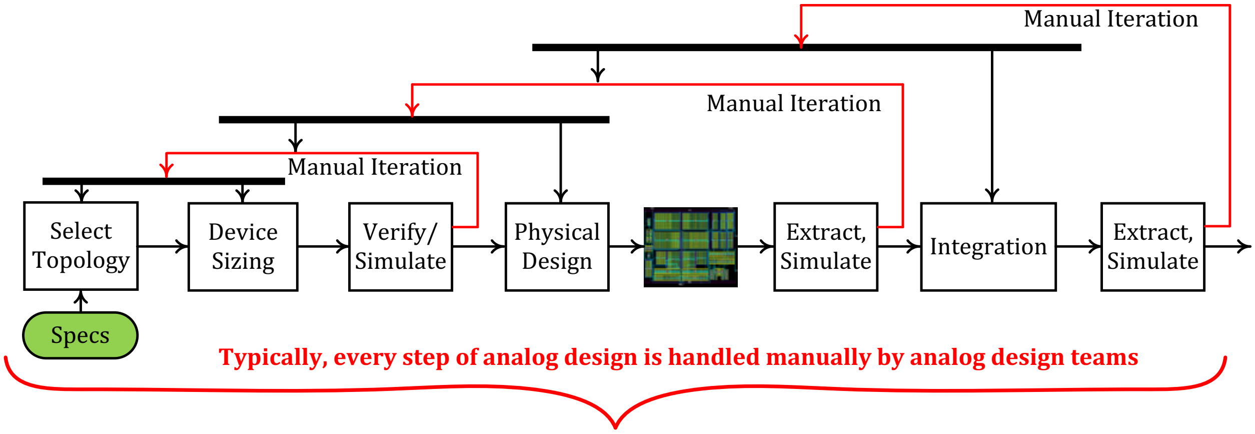


Figure 2.4: Typical progression of analog system design [30]

2.2. Why isn't Analog Synthesis a Reality Yet?

Automated analog integrated circuit design is slowly becoming a viable solution for improving design productivity for critical analog components. Over the past decade or so, analog design automation has significantly progressed. There are commercially viable tools that allow device sizing, automatic layout, and can design components with 10 to 100 devices. However, these tools face serious issues in terms of productivity since they aren't able to scale from the component level to the system level. Regardless of the success of these tools, analog design still lacks abstraction, and cannot be automatically synthesized. Some of the reasons are discussed below [27]-[29][32].

1. ***Non-Linear Design:*** Analog design is a very non-linear process which makes hierarchical abstraction challenging. Many parameters must be optimized simultaneously in order to meet specifications with robust designs and it is a recursive process that often requires a number of iterations and often changes must be made at multiple hierarchical levels simultaneously.

2. ***Bifurcation of Analog Design:*** Analog circuits do not scale as well as digital circuits. Some of the challenges that analog designs face with CMOS scaling: reduced voltage headroom, exacerbated mismatches, increased leakage and second order effects in transistor behavior. Two philosophies have emerged in analog design. According to the first school of thought, due to poor scaling of analog circuits, it is better to implement analog blocks in older processes, and use multi-die integration techniques to advance the overall performance of mixed signal systems. Secondly, there is a group of researchers who are pushing the limits of analog design in the advanced CMOS processes, and are inventing digitally intensive or all-digital architectures for rationally analog blocks. Due to this division, standardization and therefore development of standardized tools has been slow [31].
3. ***Variations:*** In digital circuits, process, voltage and temperature (PVT) variations are often addressed by overdesigning the entire digital block. They are often designed for the worst case scenario. However, PVT variations cannot be mitigated in analog design by simply overdesigning because designs that are not tuned precisely may result in a complete breakdown. For example, an overdesigned amplifier may become unstable.
4. ***Highly Individualized Skills:*** Today, the analog design skills are highly individualized where skills are highly specialized and stored in the memory of analog designers. These skills do not necessarily translate to algorithms or simple scripts or pieces of code.

5. ***Lack of Architectural Innovation:*** In the last four decades, not much architectural innovation has occurred in analog designs. Simpler architectures typically lend themselves nicely to design automation, and often amenable to levels of abstraction.

This research presents a different approach to analog synthesis. The design methodology is discussed in the following section in detail.

2.3. A Brief Survey of Analog Synthesis

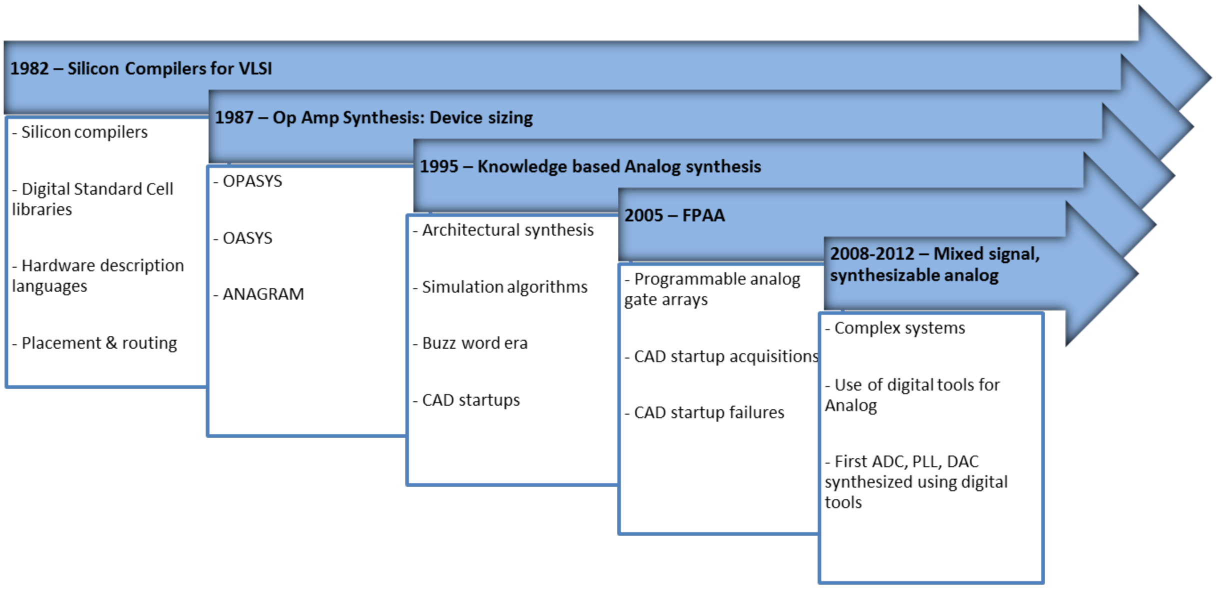

1982 – Silicon Compilers for VLSI
- Silicon compilers
- Digital Standard Cell libraries
- Hardware description languages
- Placement & routing
1987 – Op Amp Synthesis: Device sizing
- OPASYS
- OASYS
- ANAGRAM
1995 – Knowledge based Analog synthesis
- Architectural synthesis
- Simulation algorithms
- Buzz word era
- CAD startups
2005 – FPAA
- Programmable analog gate arrays
- CAD startup acquisitions
- CAD startup failures
2008-2012 – Mixed signal, synthesizable analog
- Complex systems
- Use of digital tools for Analog
- First ADC, PLL, DAC synthesized using digital tools

Figure 2.5: History of analog synthesis summarized

The advances in integrated circuits have been driven by unprecedented advances in computer aided design. The CAD tools have come so far that a majority of the digital design is completely automated today. However, almost all of the analog/RF circuits are custom designed and manually laid out. Typically, in a mixed-signal SoC, analog components make up about 20% of the circuits while requiring orders of magnitude greater design effort compared to their digital counter parts [27]. Therefore, significant research efforts have been focused on developing analog synthesis tools. This section briefly highlights the history of analog synthesis – as illustrated in Figure 2.5.

Silicon Compilers for VLSI (1982 – Present)

The appearance of silicon compilers became regular in 1982 as a result of research on digital design automation that began in the late 1970s. Some of the first reported design automation tools, known as silicon compilers, for VLSI were reported by Gajski in 1981 [24]-[25]. These tools had the ability to translate behavioral description into a structural netlist, and select an appropriate architecture for digital blocks based on delay specifications. In addition, Gajski reported algorithms for automatic layout of digital circuits. This was the beginning of a CAD tool revolution that propelled the digital integrated circuits to become so prevalent in society.

Automatic Device Sizing (1987 – Present)

Automatic synthesis and layout tools were quickly adopted for digital design, and by the 1980s, researchers began to develop analog synthesis tools. The University of California at Berkeley, and Carnegie Melon University led the way in analog synthesis and layout tools. By 1987, researchers began to develop analog standard cell libraries [34]. However, due to a much wider range of applications, specifications and possibly standard cells, a truly general purpose analog standard cell library was never developed. This caused a shift in focus by the late 80s towards the development of efficient simulation tools that would allow automatic transistor sizing and development of standard cells such as opamps. Two of the earliest reported tools for automatic opamp design were Opasys, and Oasys [34]-[38].

Automatic Hierarchical Knowledge Based Design (1995 – Present)

After relatively mediocre success with analog standard cell libraries, researchers shifted focus towards abstraction, and hierarchical design automation for analog circuits. Applications of complex algorithms such as genetic algorithms began as an attempt to replace the intuition of analog designers. This approach was dubbed knowledge based analog synthesis, which gave birth to wide spread IP reuse. However, one of the major drawback of this approach is that often multi-level hierarchical optimization is required for analog systems which was not possible with such tools. However, a large portion of these tools were integrated into the simulation tools that are utilized even today for example circuit optimizer in Spectre is a combination of algorithms developed since the late 80s [39]-[43].

Field Programmable Analog Array (1998 – Present)

By early 1990s, field programmable gate arrays for digital applications had become quite common, which triggered a wave of new research towards development of the analog equivalent of a general purpose programmable sea of blocks. This was dubbed the field programmable analog array (FPAA). However, FPAA researchers faced the same challenges as the tool developers for analog synthesis. The sheer number of parameters to be optimized simultaneously, and the trade-off of performance and reconfigurability is too costly. Research on FPAAs is still going strong, and is showing promise with the increase of digitally assisted analog circuits. [44]-[49]

Analog design with Digital Tools (2008 – Present)

In 2008, the first ever analog/RF blocks completely implemented using digital standard cells and design flows was reported at the University of Michigan. This started a trend towards migrating traditionally analog blocks to all-digital, synthesizable architectures which then could be implemented using standard digital synthesis and place-and-route tools. A number of different blocks such as ultra-wide band transmitters, time-to-digital converters, all-digital PLLs, and ADCs, have been reported to date [50]-[52][65][66][33]. This approach shows promise, and is the focus of this thesis.

Digital vs. Analog Synthesis in Industry

To further understand the disparity between analog synthesis tools versus the digital tools, a brief study was conducted on the current state of commercial analog tools, and the digital synthesis tools. As can be seen in Figure 2.6, there are barely any tools that allow analog synthesis. The only tools that the authors were able to locate that claim analog synthesis, and are commercially available are filter design tools.

Figure 2.6: Current state of commercial digital vs. analog tools [53][57]

However, these tools have severe limitations, and often accompany chip manufacturers development kits. On the other hand, there is a plethora of digital tools available. Very complex digital systems can be completely implemented using automated tools [53][57].

2.4. Current State of Commercially Available Analog Layout Tools

A number of companies offer “automatic” analog layout tools that come with a wide range of capabilities. The common goal, however, is to increase productivity and shorten the physical design cycle. Three of the most popular analog layout tools are discussed in this section.

Synopsys Helix – Device Level Placement for Custom Design [54]

Synopsys offers an analog layout tool known as Helix which automates layout of analog blocks such as PLLs, ADCs, SerDes [54]. This design tool depends on prior generation of parametric cells that require an additional suite of design tools (PyCells or Sagantec). Moreover, Helix is meant to be used to generate an initially estimated floor plan, and estimate the parasitics, and the designs are DRC-aware which result in design time savings.

Cadence Virtuoso –Electrically Aware Design (EAD) [55]

The layout design tool offered by Cadence Systems offers a unique real-time DRC and parasitics feedback to the designer as the designer lays out an analog block. This results in significant design time savings by reducing the number of iterations that are often required to fix DRC and layout parasitics. However, the analog blocks are still custom laid out by a designer.

Tanner EDA – HiPer Design Suit [56]

Tanner EDA offers a powerful analog layout tool that comes with automated device generation, place and route. However, this tool is severely limited in it’s capabilities with advanced technologies nodes. To date, this tool has only been reported to work with technologies nodes up to 90nm.

Proposed Design Methology vs. Commercial Analog Layout Tools

A detailed comparison of the above mentioned design tools to the proposed design methodology is given in **Error! Reference source not found.**.

Table 2.1: Commercial Layout Tools vs. Proposed Design Methodology

Comparison Metric	Synopsys Helix	Cadence EAD	HiPer	Proposed Design Methodology
Device Level Layout Automation	Yes	No	Yes	No
Cell to Top level layout automation	No	No	Yes	Yes
Utilizes Existing Design Tools	No	No	No	Yes
Readily integrated with Digital	No	No	No	Yes
DRC/Parasitic Aware	Yes	Yes	No	Yes

It can be seen from the table above, the layout automation tools that are available in the market today only solve a portion of the problem. One significant step forward is the real-time parasitics feedback to layout design. This can reduce the number of design iterations. However, the proposed design methodology is differentiated from the existing analog layout automation tools in two ways: 1) No additional tools or training is required 2) designs are readily integrated with digital.

2.5. A New Analog Synthesis Philosophy

The proposed analog synthesis philosophy is highlighted in Figure 2.7. The main goal of the idea behind this philosophy is to pay special attention to innovation at every level of the hierarchy throughout the design process while keeping in mind what the limitations of existing digital tools are. This allows for an accelerated design, and significantly reduces the design time without compromising the performance of analog circuits. Various hierarchical steps of the design methodology are described below:

Device Sizing: Analog design starts at the very bottom where the size of the transistors that make up various circuits is determined. Today, an average analog designer believes that digital synthesis only allows minimum length devices, and limited sets of widths. However, this is not the case. It is possible to design circuits with unconventional aspect ratios, abstract these as custom cells, and integrate them with the standard cell libraries. The first step towards synthesizing analog blocks is the ability to design with unconventional aspect ratios.

Cell Level Design: In order to bridge the gap between completely synthesized circuits and the analog performance, new cells must be designed and added to the existing digital standard cell libraries. If the cells are designed with analog performance in mind, and laid out at the standard cell grid, one can make the trade-off between performance, and the length of the design cycle [33].

Architectural Design: As mentioned earlier, one of the major challenges a synthesized analog circuit faces is that the performance does not compare to those of fully custom hand crafted designs. Though it might not be possible to achieve equal performance, many decisions made at the architectural level can mitigate the performance loss greatly. An approach similar to logic simplification should be applied to analog architectures. Often, the functionality can be maintained by removing a few of the non-essential blocks by migrating the complexity to the digital domain. For example, the divider can be eliminated from a PLL without losing programmability or performance of a PLL [33]. Further details of this simplification will be discussed in the later chapters.

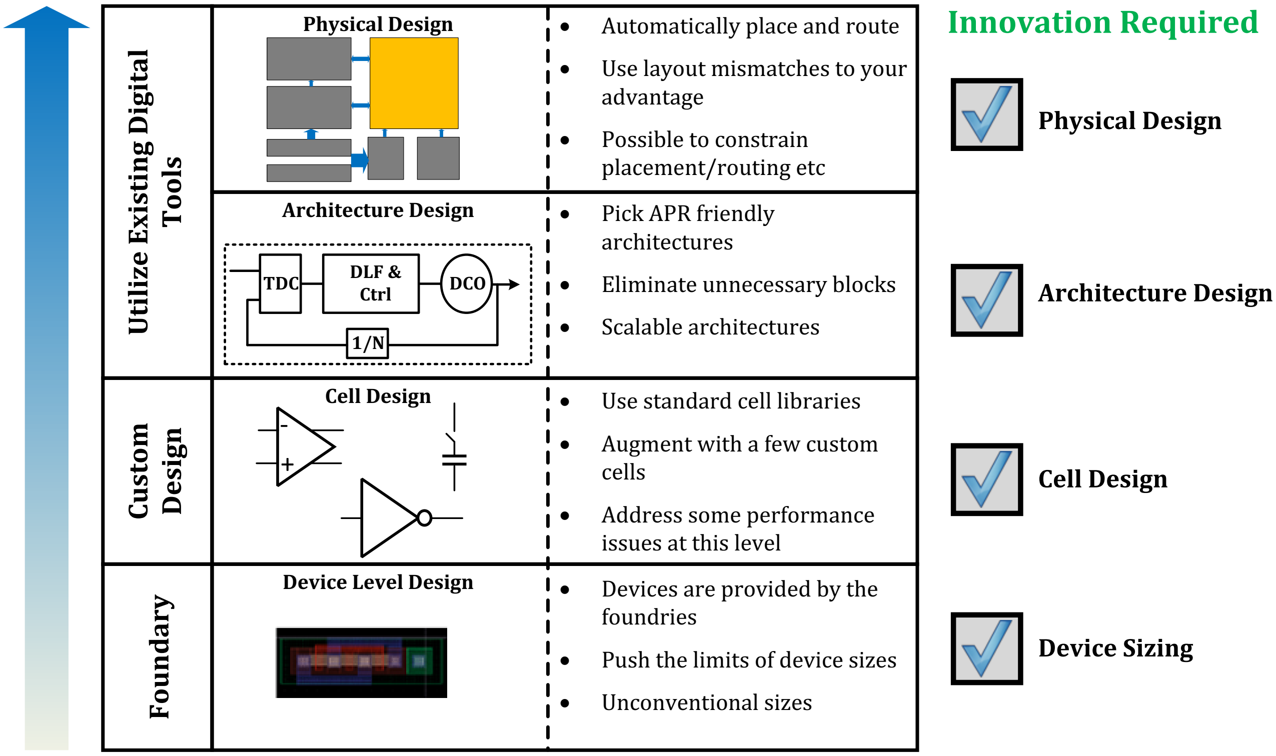

Utilize Existing Digital Tools
Custom Design
Foundary
Physical Design
Automatically place and route
Use layout mismatches to your advantage
Possible to constrain placement/routing etc
Architecture Design
TDC
DLF & Ctrl
DCO
1/N
Pick APR friendly architectures
Eliminate unnecessary blocks
Scalable architectures
Cell Design
-
+
Use standard cell libraries
Augment with a few custom cells
Address some performance issues at this level
Device Level Design
Devices are provided by the foundries
Push the limits of device sizes
Unconventional sizes
Innovation Required
Physical Design
Architecture Design
Cell Design
Device Sizing

Figure 2.7: Analog Circuit Design Innovation Chain

2.6. The Design Methodology

The steps of the analog synthesis methodology are discussed below.

Step 1: Unit Cell Design

The first step is to augment the digital standard cells library with some analog unit cells that are optimized for analog performance. For example, in a PLL design, the most crucial component is the oscillator. The unit cells that make up the oscillators (e.g. tunable delay cell) are custom designed. These cells must be laid out in accordance with the standard cell grid and integrated with the existing digital design flows [33]. This is illustrated in Figure 2.8 below.

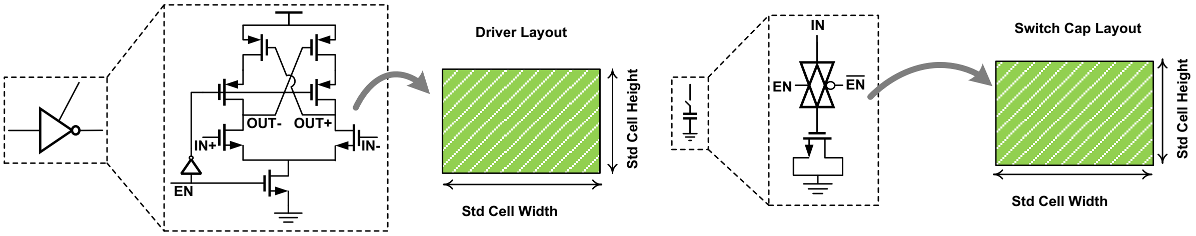
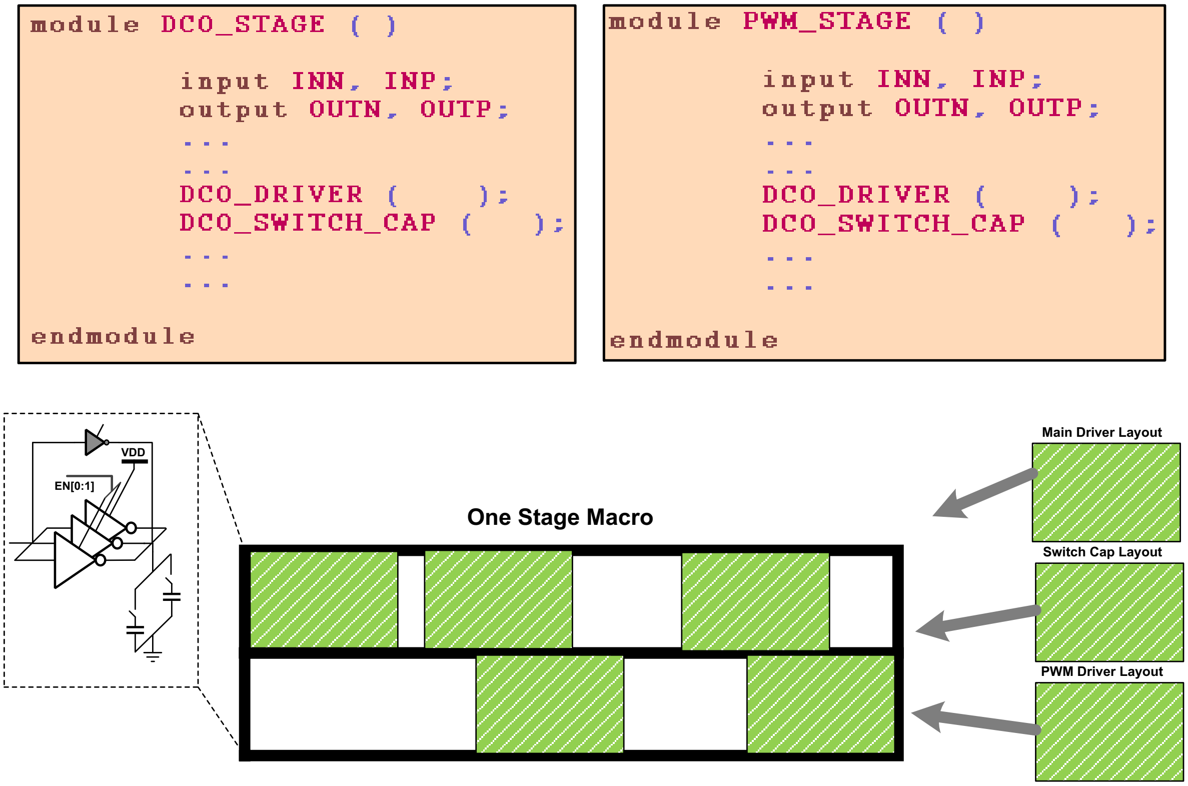


Figure 2.8: Cell design step [33]

Step 2: Macro/Sub-block Design



```
module DCO_STAGE ( )

        input INN, INP;
        output OUTN, OUTP;
        ...
        ...
        DCO_DRIVER (     );
        DCO_SWITCH_CAP (    );
        ...
        ...

endmodule
```

```
module PWM_STAGE ( )

        input INN, INP;
        output OUTN, OUTP;
        ...
        ...
        DCO_DRIVER (     );
        DCO_SWITCH_CAP (    );
        ...
        ...

endmodule
```

Figure 2.9: Illustration of Macro Design

Once the unit cells have been designed, the next step is to design a unit macro that will allow a moderate level of layout matching. This step simply instantiates the cells designed in the previous step. HDL code is written for one stage of critical blocks such as the oscillator or a delay line in a PLL. Moreover, the placement is forced in order to achieve matching. This is all done with a script and can be reiterated very easily [33]. An illustration of this step is given in Figure 2.9.

Step 3: Verification & Iterations

Once the macro has been designed, it must be simulated and verified to make sure it will meet specifications. Figure 2.10 shows the logical flow of the various steps. Depending on how far the performance of the macros is from the requirements, designer may need to go back to either the cell design or macro design steps to iterate. Because all the design steps are scripted beyond the cell design step, the iterations tend to be significantly quicker than those in traditional analog design [33].

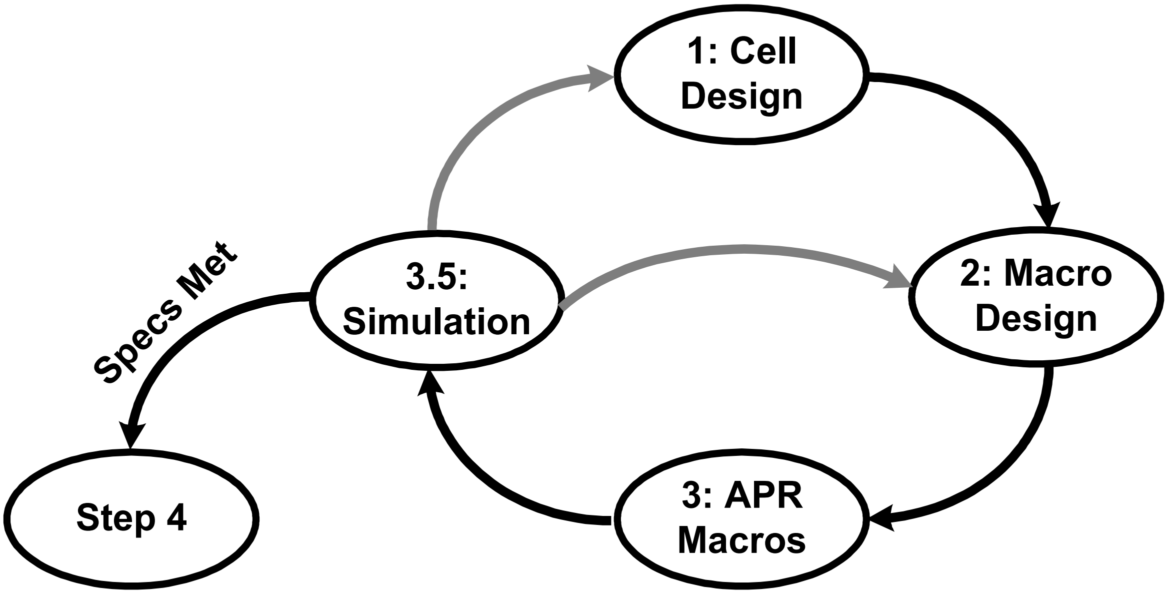
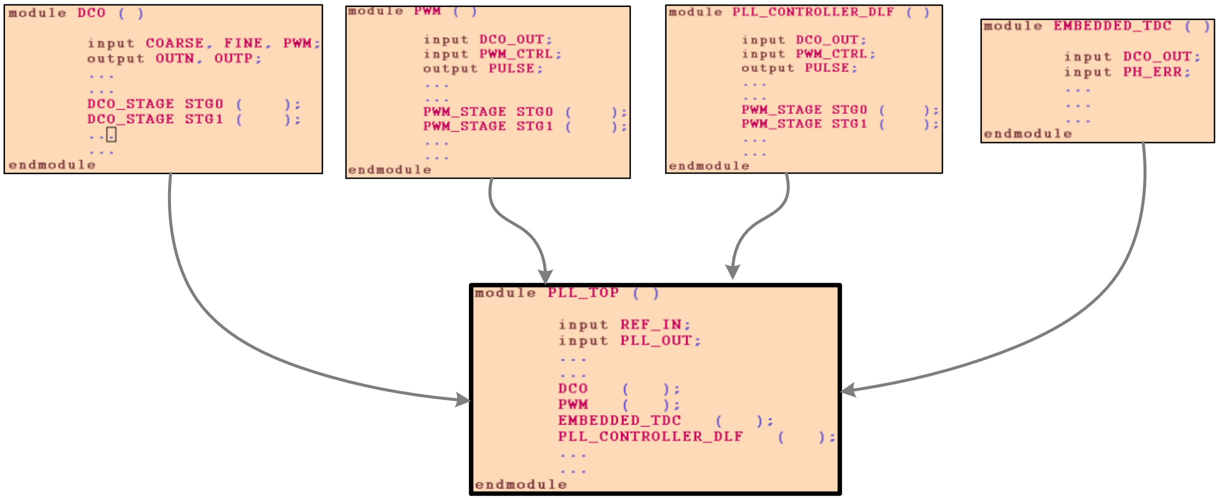


Figure 2.10: Verification and iteration step

Step 4: Top Level Design and Integration

Once the macros have been verified, the top level design and integration can begin. In this step, the entire design is brought together with HDL code, and integrated (see Figure 2.11).

The digital components can be verified using digital simulations tools, and the results can then be augmented with analog models [33].



```
module DCO ( )

        input COARSE, FINE, PWM;
        output OUTN, OUTP;
        ...
        ...
        DCO_STAGE STG0 (     );
        DCO_STAGE STG1 (     );
        ...
        ...
endmodule
```

```
module PWM ( )

        input DCO_OUT;
        input PWM_CTRL;
        output PULSE;
        ...
        ...
        PWM_STAGE STG0 (     );
        PWM_STAGE STG1 (     );
        ...
        ...
endmodule
```

```
module PLL_CONTROLLER_DLF ( )

        input DCO_OUT;
        input PWM_CTRL;
        output PULSE;
        ...
        ...
        PWM_STAGE STG0 (     );
        PWM_STAGE STG1 (     );
        ...
        ...
endmodule
```

```
module EMBEDDED_TDC ( )

        input DCO_OUT;
        input PH_ERR;
        ...
        ...
        ...
endmodule
```

```
module PLL_TOP ( )

        input REF_IN;
        input PLL_OUT;
        ...
        ...
        DCO    (    );
        PWM    (    );
        EMBEDDED_TDC    (    );
        PLL_CONTROLLER_DLF    (    );
        ...
        ...
endmodule
```

Figure 2.11: Top level design and integration

Step 5: Top Level Physical Design

The last step in the design methodology is to perform automatic place and routing of the top level design. In this step, the top level design is run through an APR tool such as Encounter. The result is a top level PLL, as illustrated by in Figure 2.12 [33].

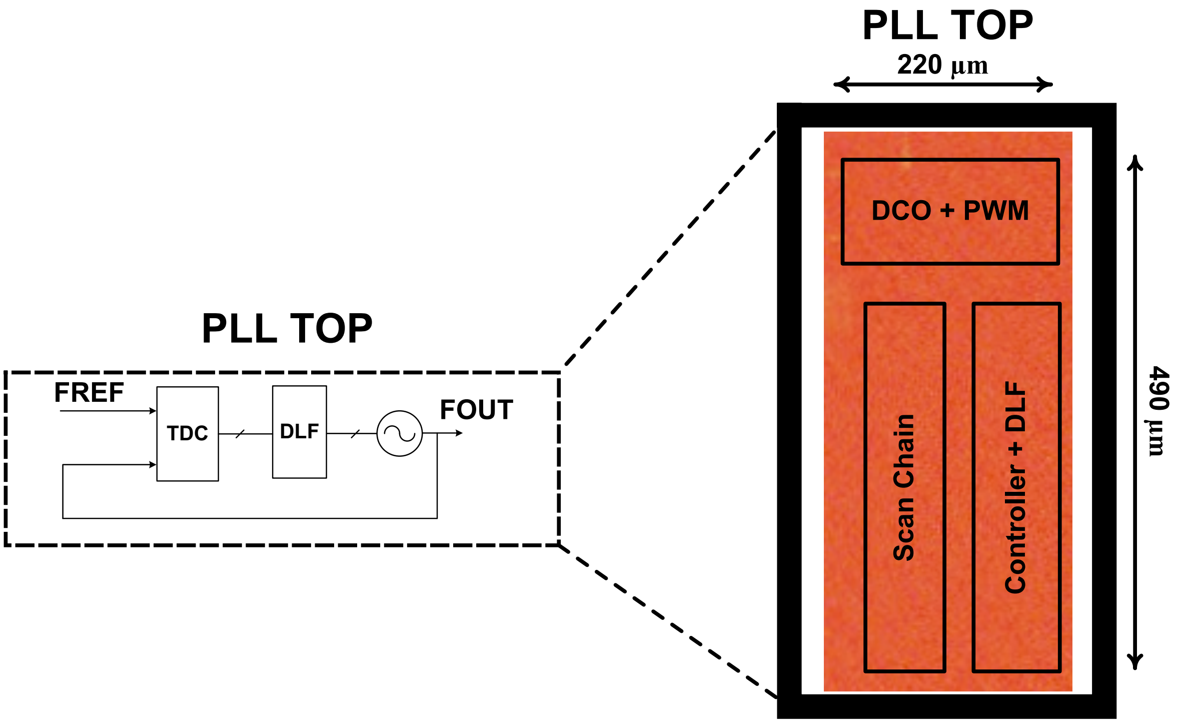


Figure 2.12: Top level physical design

2.7. Conclusion

In conclusion, the main goal of this methodology is to make the analog design process conform to the digital design process in order to use automatic design tools only available in the digital CAD flow as much as possible to accelerate the design time. This results in a significant reduction in non-recurring engineering cost as well as allows better integration. Figure 2.13 shows what the analog design looks like when done using this methodology. Compared to Figure 2.1, the steps that are automated in the digital design flow are also automated here.

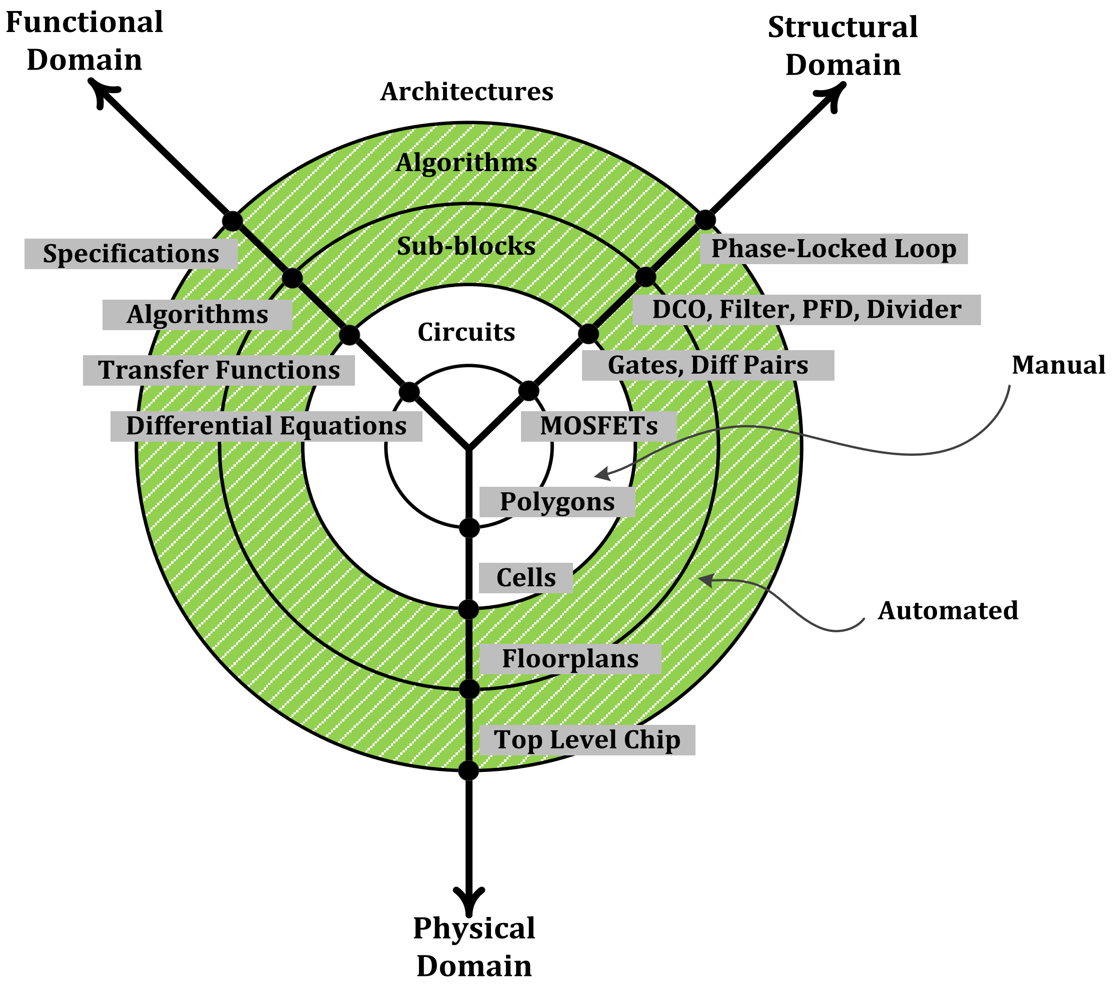


Figure 2.13: Analog design Y-chart using the proposed design methodology

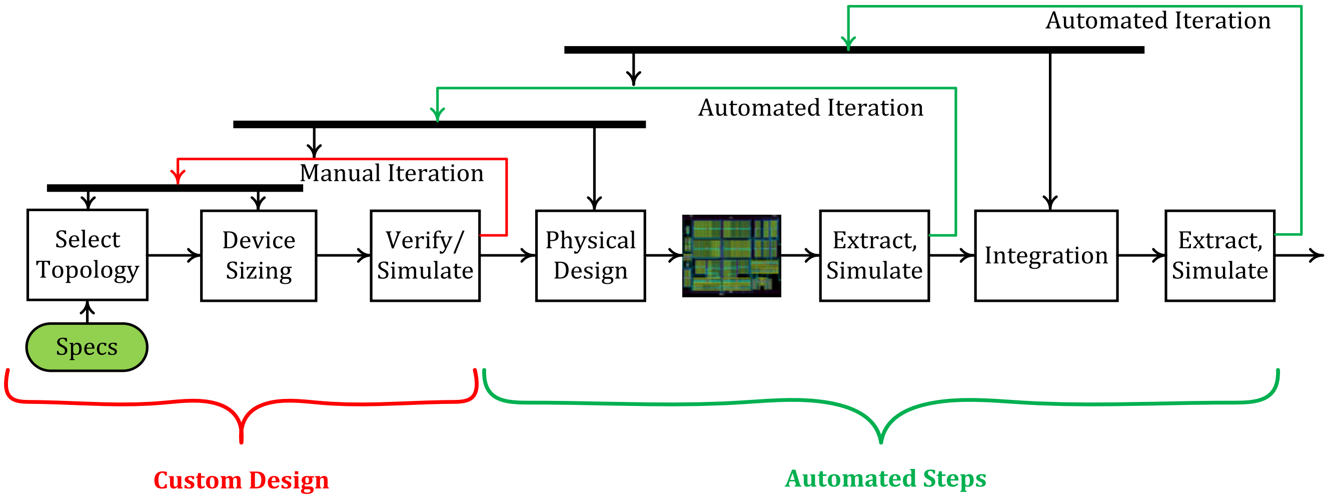


Figure 2.14: Analog design representation using the proposed design methodology

Figure 2.14 above shows what the design process for a typical analog block looks like when the proposed design methodology is utilized. The entire design process beyond the initial cell design is migrated to digital design tools. This allows quick design iterations and a significant reduction in design time. The ability to quickly design mixed-signal circuits will be of utmost importance with increased electronics demand.

This methodology has been proven in silicon, and a number of research prototypes were fabricated. Two of the prototypes are discussed in the following Chapter 4 & Chapter 5 .

Lastly, automated layout design methodologies for analog have been proposed before. Significant research resources have been spent on developing analog synthesis and layout tools with no avail. However, there are some automatic analog layout tools available in the market today. Each of these tools require additional training for the designers, are expensive and the resultant layout cannot be readily integrated with digital designs.

Chapter 3

Enhancing DCO Resolution

We know from [58] that the dominant source of phase noise in ADPLLs is the quantization noise in the TDC and the DCO. This chapter explores possibilities for resolution enhancement while optimizing power, that are amenable to design and synthesis in a digital CAD flow. It is worth mentioning here that only truly digitally controlled ring oscillators are considered in this chapter, and this analysis doesn't apply to LC oscillators. In other words, all control voltages are assumed to have a value of 0 or VDD, and no voltage tuning via a digital-to-analog converter or otherwise is used. Therefore, the frequency can only be tuned with discrete steps, and the oscillator responds instantaneously to a change in the frequency control word.

3.1. Assumptions & Specifications

The design of digitally controlled ring oscillators is a challenging and recursive process which is often facilitated by powerful simulation tools. Therefore, a number of simplifying assumptions will be made here in order to gain design insight without getting buried in complicated mathematics. The assumptions, design requirements, and specifications are summarized in Table 3.1. In addition, a number of constraints are set by the process technology, the design methodology, and the DCO architecture selection. Those limitations are as following:

1. Tunable delays must be amenable to implementation in a digital standard cell form factor in order to facilitate automatic placing and routing of the ring oscillator. This primarily constrains the size and/or speed of these elements to be on the same order as that of a standard cell (e.g. a minimum sized inverter).
2. Only digitally tunable switch capacitors & buffers are considered for frequency tuning of the DCO. There are several options and implementation variants of tunable oscillators. Upon evaluating several options, we concluded these two were the most practical for implementation in a VLSA flow, and therefore restrict the analysis to these elements.

Table 3.1: Assumptions and design specifications

Specification/Assumption	**Value**	**Comment**
Number of stages	K	K-stage digitally controlled ring oscillator
Tuning steps per stage	N	Total tuning steps: $N \cdot K$ for large N
Linearity	-	Assuming we are designing for fine control and frequency tuning will be linear.
Delay model	-	We will be using a simplified capacitor charge/discharge model for delay/frequency calculations.
DCO Frequency tuning range (Δf)	$f_{min}\ to\ f_{max}$	$\frac{1}{2Kt_{dmax}}\ to\ \frac{1}{2Kt_{dmin}}$
Delay tuning range	$t_{dmin}\ to\ t_{dmax}$	As capacitors are added, the delay will shift away from the original specifications. Therefore, more current must be injected in order to bring it back to the original specifications.
Nominal minimum delay per stage	t_{d0}	$t_{d0} = \frac{C_{L\_\min\ \_buffer} V_{DD}/2}{I_{DSAT\_\min\ \_buffer}}$
Minimum sized buffer's intrinsic capacitance	$C_{int\_\min\ \_buffer}$	$C_{int\_\min\ \_buffer} \propto (WL)_{\min\ \_buffer}$ $C_{int\_min\_buffer}$: C_{diff} & C_{gate} lumped together
External fixed capacitance	C_{ext_fixed}	Consists of wiring capacitance, and any parasitics that are added when turning buffers/switches are added
Minimum current	$I_{DSAT\_\min\ \_buffer}$	$k' \left(\frac{W}{L}\right)_{\min\ \_buffer} \left[(V_{DD} - V_T)V_{DSAT} - \frac{{V_{DSAT}}^2}{2}\right]$
Relative aspect ratios	$\left(\frac{W}{L}\right)_{n,p}$	$\left(\frac{W}{L}\right)_p = 2\left(\frac{W}{L}\right)_n$
Voltage supply	$V_{DD} = 1$	

3.2. Tuning Delay with a Switched Capacitor

One way to tune delay (and frequency) digitally is to add digitally controlled switched-capacitor elements at the output node of the ring oscillator buffers – shown in Figure 3.1. This allows linear tuning of delay since the delay is linearly proportional to the load capacitance. We will assume that we are adding N switched capacitors per stage.

The delay, t_d, in Figure 3.1 can be expressed as [59]:

$$t_d = t_{d0}\left(1 + \frac{C_{ext\_fixed} + n \cdot C_{ext\_tunable}}{C_{int\_\min\ \_buffer}}\right) \tag{3.1}$$

Where t_{d0} is the intrinsic delay of the buffer without any external loading (i.e. without the switch capacitor), C_{ext_fixed} is the fixed external capacitance loading the buffer (e.g. wiring capacitance & the parasitic capacitance added by the switch) that will vary non-linearly with N, $C_{ext\_tunable}$ is the incremental capacitance added by one switched capacitor stage being activated, n is the digital control word and ranges from 0 to N, and $C_{int\_min\_buffer}$ is the internal fixed capacitance of the minimum sized buffer.

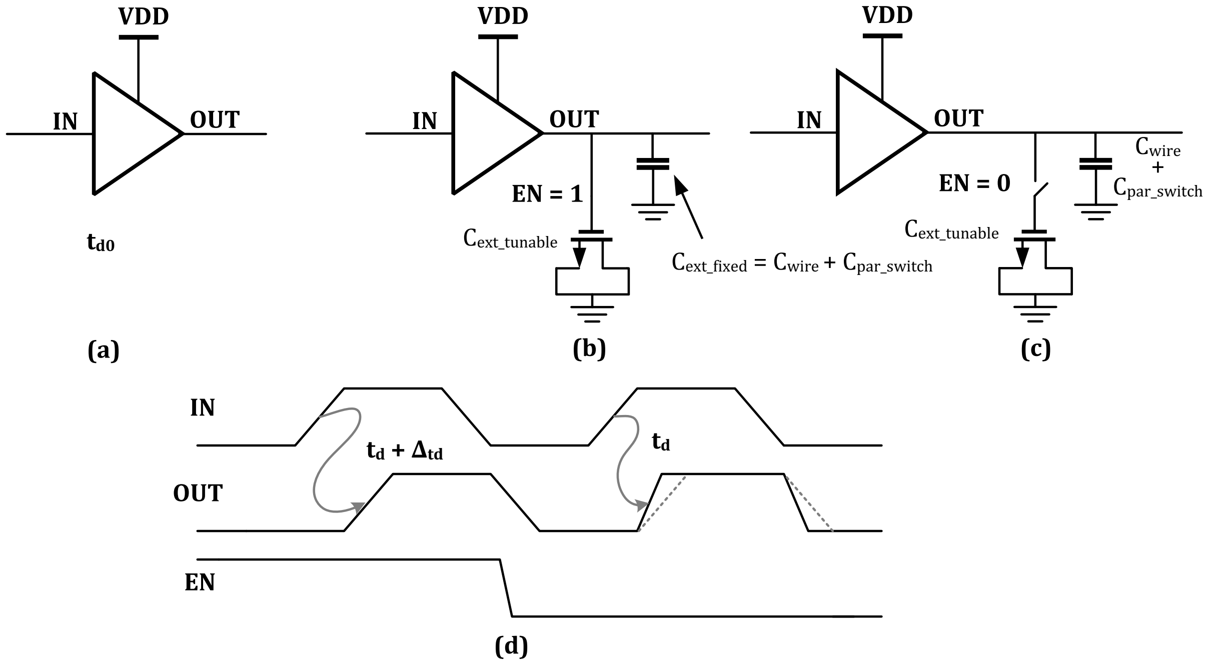


Figure 3.1: (a) Unloaded buffer (b) Switch capacitor on (c) off (d) delay waveform

Here we will make some further simplifying assumptions. First, we assume we are targeting high frequencies, therefore the buffers in the ring oscillator will use transistors that are near minimum-sized for the given process to achieve the smallest nominal gate delay. Second, a switched capacitor load element (designed as shown in Figure 3.1) would also use minimum-sized transistors in order to achieve the finest possible resolution control. These assumptions result in the loading capacitance of a single switched capacitor element being on the order of the intrinsic capacitance of the buffer ($C_{int\_min\_buffer}$). More specifically, we assume the minimum size NMOS device adds a tunable capacitance value $C_{ext\_tunable}$. We must also note that C_{ext_fixed} is a non-linear function of N, the total number of tunable switched-capacitors.

Now we can compute t_d(n) (from Figure 3.1) as follows:

$$t_d(n) \ = t_{d0}\left(1 + \frac{C_{ext\_fixed}(N) + n\,C_{ext\_tunable}}{C_{int\_\min\ \_buffer}}\right) \tag{3.2}$$

$$t_d(n) = t_{d0}\left(1 + \frac{C_{ext\_fixed}(N)}{C_{int\_\min\ \_buffer}}\right) + t_{d0}\left(\frac{n\,C_{ext\_tunable}}{C_{int\_\min\ \_buffer}}\right) \tag{3.3}$$

$$\Delta t_d = t_{d0}\left(\frac{n\,C_{ext\_tunable}}{C_{int\_\min\ \_buffer}}\right) \tag{3.4}$$

Therefore, the resolution is determined by the external tunable capacitance relative to the fixed capacitance (made up of wiring, switch parasitics and intrinsic capacitance of the buffer). For f_{min}, we can set n=N in equation 3.3, and for f_{max}, n=0.

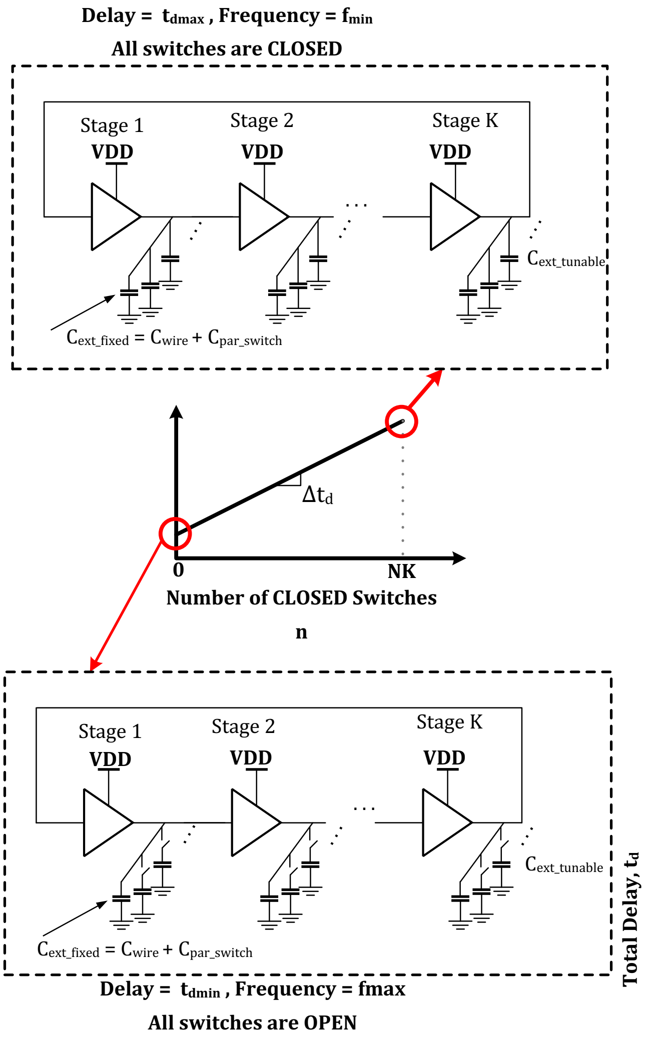

Delay = t_{dmax} , Frequency = f_{min}
All switches are CLOSED
Stage 1
VDD
Stage 2
VDD
Stage K
VDD
$C_{ext\_tunable}$
C_{ext_fixed} = C_{wire} + C_{par_switch}
Δt_d
0
NK
Number of CLOSED Switches
n
Total Delay, t_d
Stage 1
VDD
Stage 2
VDD
Stage K
VDD
$C_{ext\_tunable}$
C_{ext_fixed} = C_{wire} + C_{par_switch}
Delay = t_{dmin} , Frequency = fmax
All switches are OPEN

Figure 3.2: Maximum and minimum frequency scenarios

Frequency Resolution

As a specific example, let's assume that we tune $C_{ext\_tunable}$ with a minimum sized NMOS devices and that $C_{ext\_tunable}$ = (1/3) $C_{int\_min\_buffer}$. This means that the delay can be tuned with a step size of t_{d0}/3. The intrinsic delay in a 65 nm CMOS technology (used to fabricate the first prototype) is approximately 10ps. Therefore, a switch capacitor allows tuning delay with ~3ps steps – which translates to approximately 500 kHz step @ 400 MHz f_{rosc}. This **frequency step is too large**, and alternative methods of delay/frequency tuning are discussed in the following sections.

From 3.3, we can calculate the frequency of the DCO:

$$f(n) = \frac{C_{int\_\min\ \_buffer}}{2Kt_{d0}}\left(\frac{1}{C_{int\_\min\ \_buffer}+C_{ext_{fixed}}(N)+nC_{ext_{tunable}}}\right) \tag{3.5}$$

$$\frac{d}{dn}f(n) = \frac{C_{int\_\min\ \_buffer}}{2Kt_{d0}}\left(\frac{-C_{ext\_tunable}}{(C_{int\_\min\ \_buffer}+C_{ext\_fixed}(N)+nC_{ext\_tunable})^2}\right) \tag{3.6}$$

Therefore, if large frequency tuning range is required, this is a very non-linear method of frequency tuning.

Power Consumption

The power of a ring oscillator using static CMOS buffers is:

$$P = CV_{DD}^2 f \tag{3.7}$$

As illustrated in Figure 3.2, Maximum frequency is achieved by opening all N switches. Therefore, the external capacitance is at its minimum in all K stages of the ring oscillator.

$$P_{switches\_off} = K(C_{int\_\min\ \_buffer} + C_{ext\_fixed}(N))V_{DD}^2 f_{max} \tag{3.8}$$

When all switches are turned on:

$$P_{switches\_on} = K\left[C_{ext\_fixed}(N) + C_{L\_\min\ \_buffer} + NC_{ext\_tunable}\right]V_{DD}^{2}f_{min} \tag{3.9}$$

From equation 3.3,

$$\frac{1}{f_{max}} = t_{dmin} = 2Kt_d(0) = 2Kt_{d0}\left(1 + \frac{C_{ext\_fixed}(N)}{C_{int\_\min\ \_buffer}}\right) \tag{3.10}$$

$$\frac{1}{f_{min}} = t_{dmax} = 2Kt_d(N) = 2Kt_{d0}\left(1 + \frac{C_{ext\_fixed}(N)}{C_{int\_\min\ \_buffer}}\right) + 2KN\, t_{d0}\left(\frac{C_{ext\_tunable}}{C_{int\_\min\ \_buffer}}\right) \tag{3.11}$$

Therefore, maximum and minimum power levels are:

$$P_{switches_{off}} = \left[C_{int\_\min\ \_buffer} + C_{ext_{fixed}}(N)\right](V_{DD})^2 \frac{1}{2t_{d0}\left(1 + \frac{C_{ext_{fixed}}(N)}{C_{int\_\min\ \_buffer}}\right)} \tag{3.12}$$

$$= C_{int\_\min\ \_buffer}(V_{DD})^2 \frac{1}{2t_{d0}} = V_{DD} I_{DSAT\_\min\ \_buffer}$$

$$P_{switches\_on} = \left[C_{ext\_fixed}(N) + C_{int\_\min\ \_buffer} + NC_{ext\_tunable}\right](V_{DD})^2 \frac{1}{2t_{d0}\left(1 + \frac{C_{ext\_fixed}(N) + N\, C_{ext\_tunable}}{C_{int\_\min\ \_buffer}}\right)} \tag{3.13}$$

$$= C_{int\_\min\ \_buffer} \frac{1}{2t_{d0}} = V_{DD} I_{DSAT\_\min\ \_buffer}$$

From 3.12 & 3.13 we can conclude that power stays constant.

3.3. Tuning Delay with Parallel Buffers

Next, we will discuss delay tuning by switching parallel buffers on/off - as shown in Figure 3.3. We begin with a minimum sized buffer (Figure 3.3a), and maximum delay is achieved by turning all the switchable buffers off (Figure 3.3b), and the minimum delay is achieved when all the buffers are turned on (Figure 3.3c). The delay is to be tuned between t_{d2} and t_{d1} with N steps.

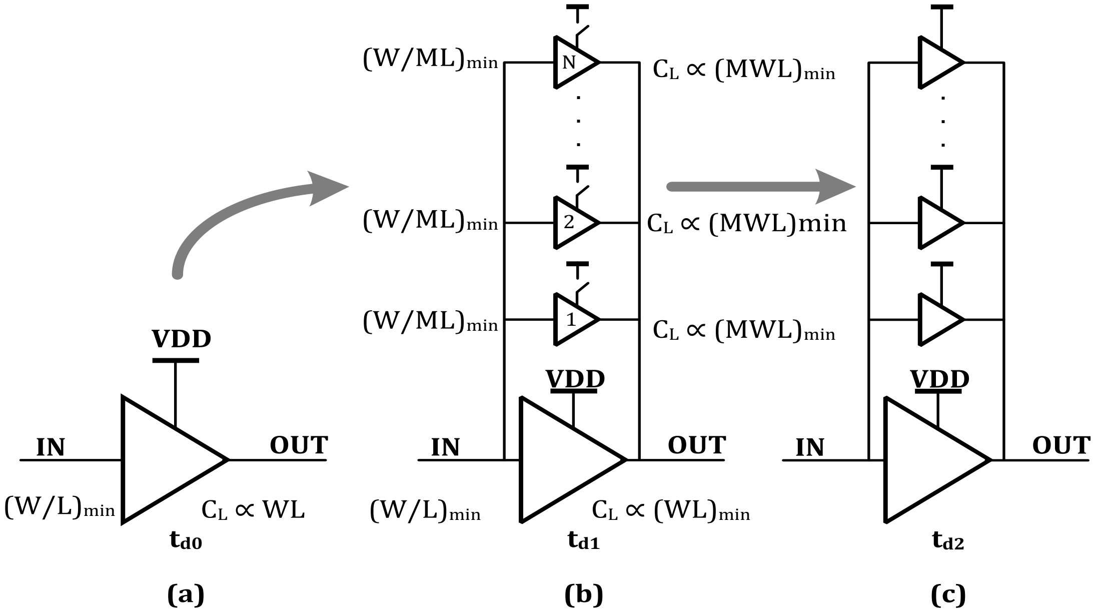


Figure 3.3: Tuning delay with parallel buffers

Simplified expression for the delay of a minimum sized buffer is as following:

$$t_{d0} = \frac{C_{int\_\min\ \_buffer} V_{DD}/2}{I_{DSAT\_\min\ \_buffer}} \tag{3.14}$$

Note: The assumptions discussed in section 3.1 are valid in this analysis as well.

We are to design a digitally tunable delay element by adding N parallel buffers to the minimum sized buffer. The desired delay tuning range is t_{dmin} to t_{dmax} which is achieved by turning off buffers. Each of the additional buffers has $C_L \propto (MWL)$ since the aspect ratio of additional buffers is $\propto \frac{W}{ML}$. This results in total overall C_{ext_fixed}:

$$C_{ext\_fixed} \propto (WL)_{min} + (MWL)_{\min[1]} + (MWL)_{\min[2]} \dots + (MWL)_{\min[N]} \tag{3.15}$$

$$C_{ext\_fixed} \propto (MN+1)(WL)_{min} \propto (MN+1)C_{int\_\min\ \_buffer} \tag{3.16}$$

$$t_d(n) = \frac{(MN+1)C_{int\_\min\ \_buffer} V_{DD}/2}{I_{DSAT\_\min\_buffer} + nI_{DSAT\_tunable}} \tag{3.17}$$

$$f(n) = \frac{I_{DSAT\_\min\_buffer} + nI_{DSAT\_tunable}}{2K(MN+1)C_{int\_\min\ \_buffer}V_{DD}/2} \tag{3.18}$$

We know that $I_{DSAT\_tunable}$ by design is:

$$I_{DSAT\_tunable} = \frac{I_{DSAT\_\min\_buffer}}{M} \tag{3.19}$$

Therefore, the frequency resolution is:

$$\frac{d}{dn}f(n) = \frac{I_{DSAT_{tunable}}}{K(MN+1)C_{int\_\min\ \_buffer}V_{DD}} = \frac{I_{DSAT\_\min\ \_buffer}}{KM(MN+1)C_{int\_\min\ \_buffer}V_{DD}} \tag{3.20}$$

We can see from 3.19 that the frequency resolution is a strongly non-linear function of the number of tunable buffers as well as the relative size of the tunable buffer – captured by M.

To calculate the minimum frequency, n = 0:

$$\frac{1}{f_{min}} = 2Kt_d(0) = \frac{K(MN+1)C_{int\_\min\ \_buffer}V_{DD}}{I_{DSAT\_\min\ \_buffer}} \tag{3.21}$$

The maximum frequency can be calculated by setting n=N:

$$\frac{1}{f_{max}} = 2Kt_d(N) = \frac{K(MN+1)C_{int\_\min\ \_buffer}V_{DD}}{I_{DSAT\_\min\ \_buffer} + NI_{DSAT_{tunable}}} \tag{3.22}$$

The power levels when all buffers are on/off can be expressed as following:

$$P_{buffers\_off} = K(MN+1)C_{int\_\min\ \_buffer}V_{DD}^2 f_{min} \tag{3.23}$$

$$P_{buffers\_on} = K(MN+1)C_{int\_\min\ \_buffer})V_{DD}^2 f_{max} \tag{3.24}$$

By substituting the values of f_{min}, and f_{max} into 3.23 & 3.24:

$$P_{buffers\_off} = I_{DSAT\_\min\ \_buffer}V_{DD} \tag{3.25}$$

$$P_{buffers_{on}} = (I_{DSAT\_\min\ \_buffer} + NI_{DSAT_{tuning}})V_{DD} \tag{3.26}$$

If we substitute 3.22 into 3.26, then

$$P_{buffers_{on}} = \left(1 + \frac{N}{M}\right) I_{DSAT\_\min\ \_buffer} V_{DD} \tag{3.27}$$

Frequency Resolution

High delay/frequency resolution can be achieved using this approach. However, the power consumption is inversely proportional to the resolution. The frequency resolution of this approach is given by 3.19.

Power Consumption

As expresses in 3.27, the power consumption of this approach proportional to the total number of added of buffers, N, and the relative size of the added buffers.

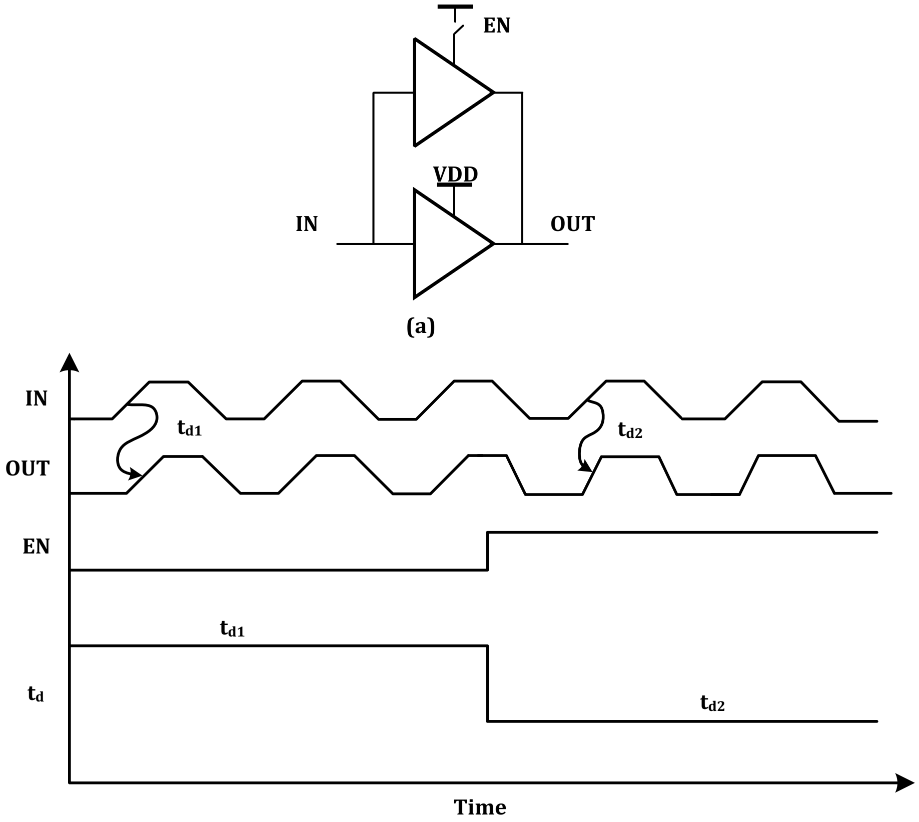
3.4. Pulse Width Modulated Delay/Frequency Tuning

So far we have discovered two major issues with the switch capacitor tuning and parallel buffer tuning:

1. The minimum achievable frequency step is too large with switched capacitors, and resolution vs. switched-capacitor is a highly strong non-linear function which could cause problems if used in a PLL. Nonlinear frequency tuning could cause the PLL to be either unstable or cause spurs.
2. The power consumption could be too high with parallel buffers approach

Now we present a new technique for frequency tuning which is based on pulse width modulation that achieves the same high resolution of buffer switching without the non-linearities, but at the power consumption levels of switched capacitor tuning.

Instead of having a large bank of switchable buffers or capacitors, consider using only one switchable buffer per stage, and then turning it on/off only for a portion of the DCO period. For example, Figure 3.4 illustrates what the delay would look like when a single parallel switchable delay element is turned on/off for the entirety of the DCO period. In this case, the delay resolution is limited (or set) by the size of switchable delay element.

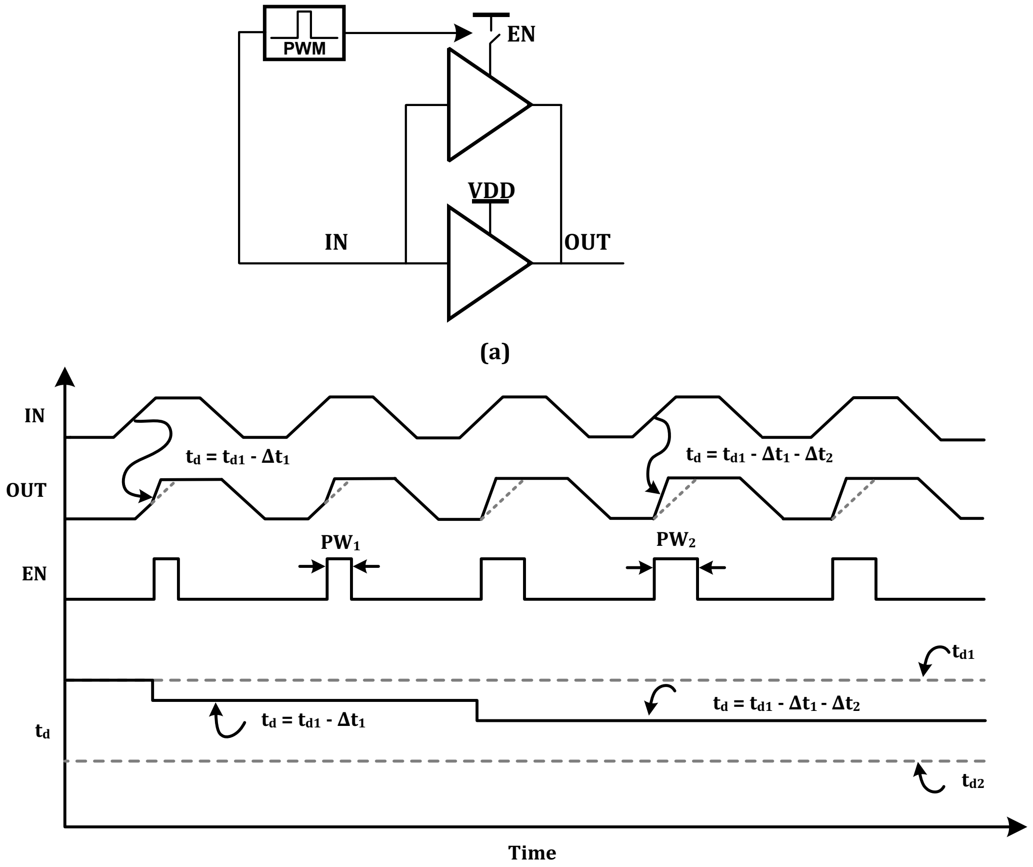


(a)

(b)

Figure 3.4: Tuning delay by turning on/off a buffer for the entirety of the period

The delay can be tuned with much finer steps than what a single switchable buffer can provide by turning on the switchable buffer only for a portion of the period and varying (tuning) that portion of time. By only enabling the switched buffer for a fraction of the period, the charge/discharge current of the stage is increased for only a fraction of time. As this fraction of time is tuned, the time it takes to charge/discharge the load capacitance varies with it, ultimately tuning the propagation delay t_d. Tuning this on-time of the switch buffer can be achieved with a pulse width modulator driving the enable input – as illustrated in Figure 3.5.


PWM
EN
VDD
IN
OUT
IN
$t_d = t_{d1} - \Delta t_1$
$t_d = t_{d1} - \Delta t_1 - \Delta t_2$
OUT
PW_1
PW_2
EN
t_{d1}
$t_d = t_{d1} - \Delta t_1 - \Delta t_2$
$t_d = t_{d1} - \Delta t_1$
t_d
t_{d2}
Time

(a)

(b)

Figure 3.5: Delay as a function of pulse width

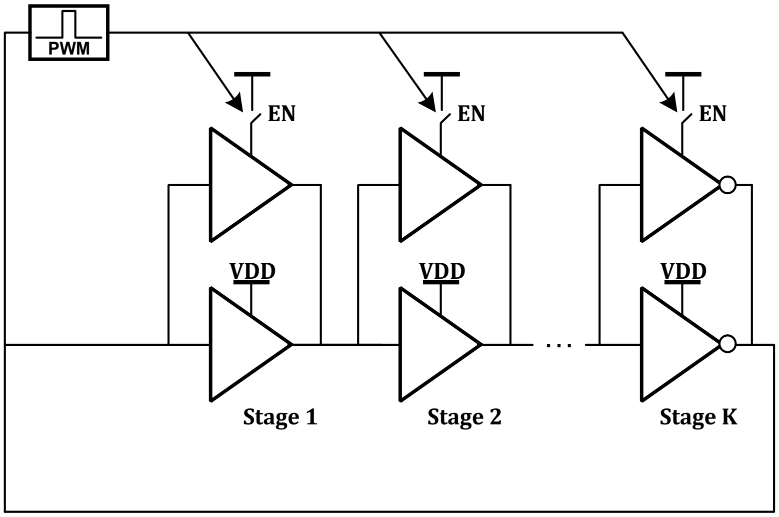


Figure 3.6: Pulse Width Modulator Based Frequency Tuning in an RDCO

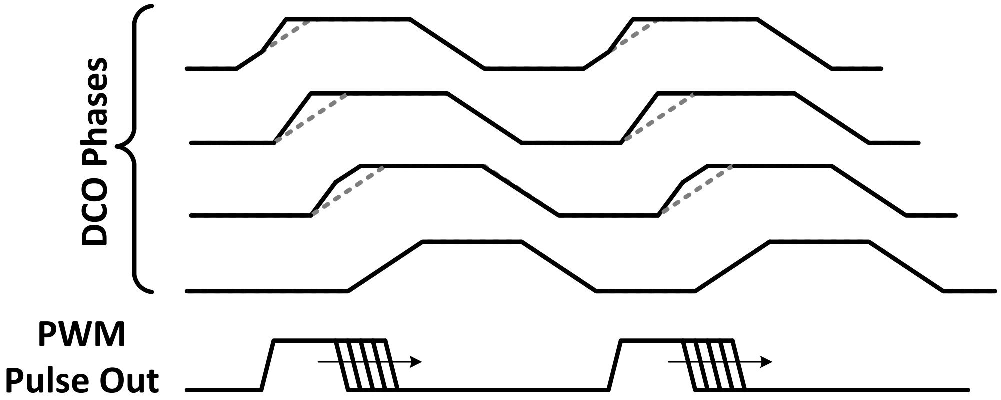


Figure 3.7: Pulse width modulator in a ring oscillator

A ring oscillator built using delay elements that are tuned with this PWM technique is shown in Figure 3.6.

The function of the PWM-based frequency tuning in a ring oscillator can be further understood by studying Figure 3.7. The phase transitions in the ring oscillator that overlap the pulse are slightly sped up, while the rest of the transitions stay the same. This slight change in the rise/fall times causes a very small delay change, and by tuning the pulse width, the frequency of the ring oscillator can be tuned with fine steps.

Frequency Resolution

The frequency can be incremented with extremely small steps using this approach since the buffer being controlled by the PWM can be sized to have an arbitrarily small aspect ratio. In depth discussion of the resolution is given in section 3.5.

Power Consumption

In addition to the power from the two minimum sized buffers per stage, the power from the PWM block that produces the signal must also be included. The PWM, as shown in Figure 3.8, is switched at the DCO frequency and the power is dominated by the power of a tunable delay as well. However, we can utilize the power efficient method of tuning delay (i.e. switched capacitors) in the PWM without affecting the frequency of the oscillator. Increasing the resolution of the PWM does not affect the DCO frequency as long as the PWM is designed to function at the highest possible DCO frequency.

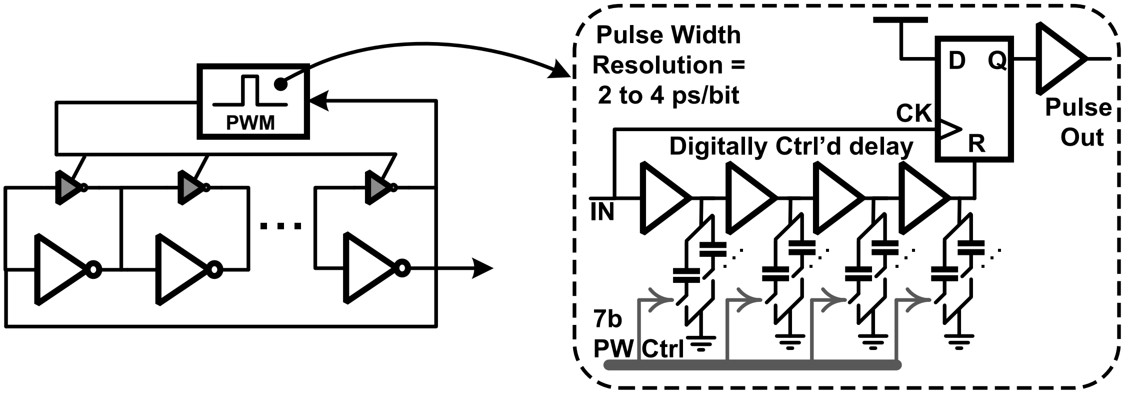


Figure 3.8: The pulse width modulator architecture in a ring DCO

The power consumption of the PWM is dominated by the tunable delay which is responsible for modulating the pulse width. Therefore, the power consumption of the entire DCO constructed using the PWM approach is the sum of the switched-capacitor tunable delay and the additional buffers that are controlled by the PWM.

$$P_{total} = P_{PWM} + P_{ring} \tag{3.28}$$

P_{PWM} can be computed from 3.10 & 3.11 and the ring power can be computed from 3.26 where N=M =1:

$$P_{maximum} = V_{DD} I_{DSAT\_\min\ \_buffer} + 2V_{DD} I_{DSAT\_\min\ \_buffer} \tag{3.29}$$

Similarly, at lowest frequency, the entire PWM and the additional buffer in the ring oscillator will be off. Therefore, the minimum power consumption is:

$$P_{minimum} = V_{DD} I_{DSAT\_\min\ \_buffer} \tag{3.30}$$

3.5. Resolution using PWM Technique

To gain further understand of the PWM-based frequency/delay tuning technique, we will write an analytical expression for the delay resolution. We can write a closed form expression for the delay resolution that is achievable by utilizing the PWM technique by considering the maximum average current that a switchable buffer provides. From section 5.4, the switchable buffer is sized to be twice as big as the minimum sized buffer because the pulse width modulator can only provide pulse width tuning up to half of the period. One way to look at the PWM functionality is that the PWM-controlled buffer is slightly increasing the drive strength of the overall stage by injecting extra current into the load capacitance for a fraction of the period. This is illustrated in Figure 3.9.

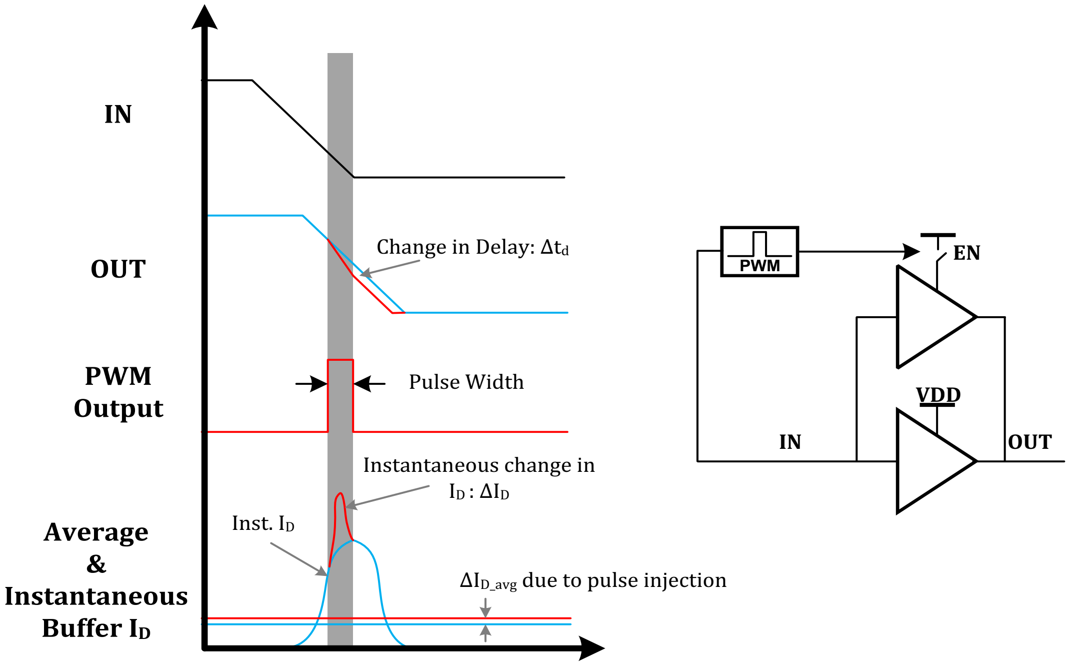

IN
OUT
PWM Output
Average & Instantaneous Buffer I_D
Change in Delay: Δt_d
Pulse Width
Instantaneous change in I_D : ΔI_D
Inst. I_D
ΔI_{D_avg} due to pulse injection
PWM
EN
VDD
IN
OUT

Figure 3.9: Delay resolution as a function of the pulse width

This increases the average current slightly. The change in current and the delay can be expressed as:

$$\Delta I_{DSAT\_w\_PWM} = \frac{\Delta PW}{T_{dco}} I_{DSAT\_\min\ \_buffer};\ \Delta PW\ is\ the\ PWM\ LSB \tag{3.31}$$

$$t_d(n) = \frac{2C_{int\_\min\ \_buffer}V_{DD}/2}{I_{DSAT\_\min\ \_buffer} + n\frac{\Delta PW}{T_{dco}} I_{DSAT\_\min\ \_buffer}} \tag{3.32}$$

$$f(n) = \frac{I_{DSAT\_\min\ \_buffer} + n\frac{\Delta PW}{T_{dco}} I_{DSAT\_\min\ \_buffer}}{2KC_{int\_\min\ \_buffer}V_{DD}} \tag{3.33}$$

$$\frac{d}{dn} f(n) = \frac{\Delta PW I_{DSAT\_\min\ \_buffer}}{2KT_{dco}C_{int\_\min\ \_buffer}V_{DD}} \tag{3.34}$$

3.6. Drawbacks of PWM-based Frequency Tuning

As it has been discussed in the previous two sections that using the PWM-based delay tuning technique offers significant improvements over the switch-capacitor only or switched-buffer only techniques. However, there are a number of drawbacks that must be considered when designing a PWM-tuned oscillator. Those issues are discussed in this section.

1. Initial Pulse Width Offset

One of the issues with tuning frequency/delay using the PWM is that the PWM, as shown in Figure 3.8 has an initial PW offset. Therefore, when designing frequency/delay tuning, the designer must keep the PWM enabled in order to avoid any jumps in frequency when the PWM tuning starts.

2. Overlap in Transitions is required for robust function

One of the major challenges with this technique is that there must be sufficient overlap between transitions of adjacent stages in a ring oscillator.

As shown in Figure 3.10, when there is insufficient overlap in the transitions of adjacent phases of the oscillator, there is a strong non-linearity in the frequency vs. PW curve. This could cause unnecessary spurs when used in a PLL. Therefore, the designer must ensure that there is sufficient overlap between transitions.

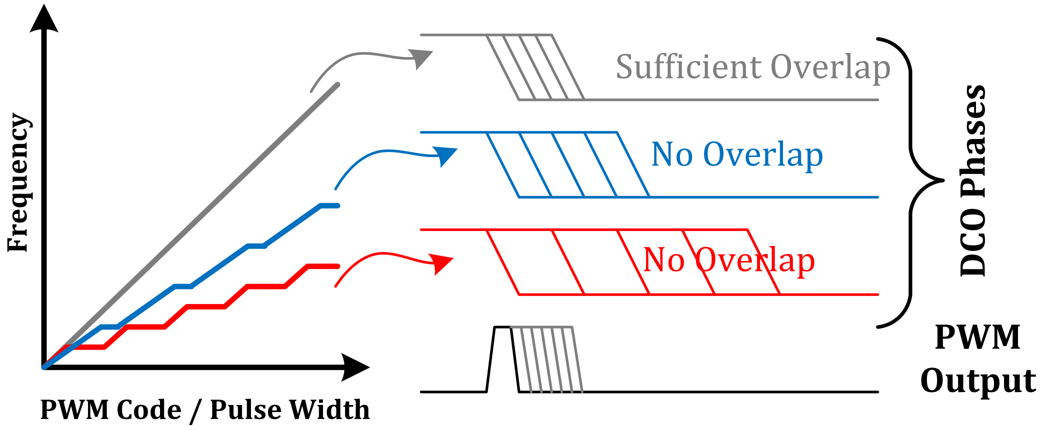


Figure 3.10: Different scenarios of PWM-based frequency tuning

3.7. Measured Results

The proposed PWM-based frequency tuning technique was fabricated and utilized in a 400-460MHz ADPLL that will be discussed in Chapter 4 . With the help of this technique, we were able to reduce the frequency tuning LSB size by 20 times (from 1.2 MHz to 60 kHz).

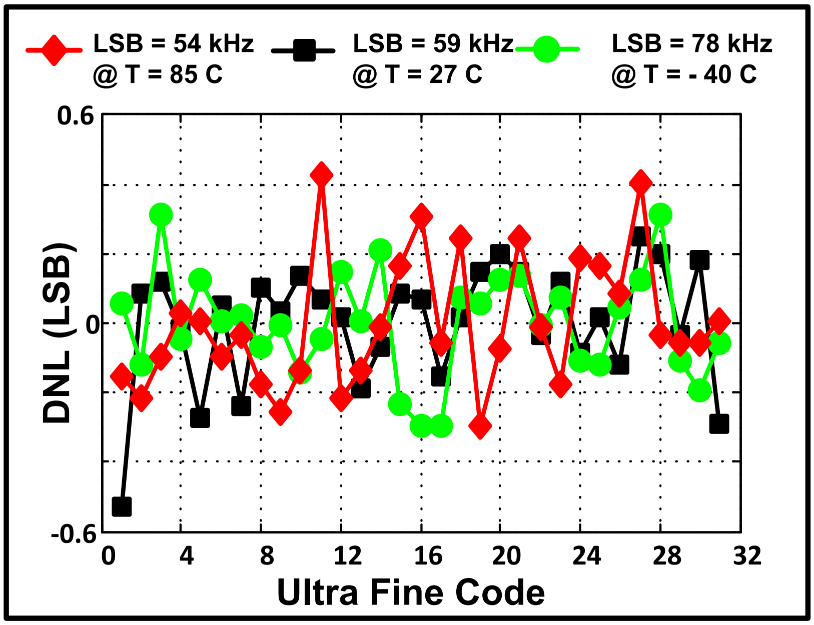


Figure 3.11: Differential non-linearity of the PWM frequency control

The measured DNL of ultra-fine (PWM) tuning as a function of temperature is presented in Figure 3.11. The worst case DNL at 403MHz is -0.55 LSB. Finally, this coarse/fine tuning combined with PWM control decouples the resolution versus tuning range tradeoff, and the DCO is able to have small quantization error as well as a wide tuning range. Since this is a newly proposed technique, the performance as a function of V_{DD} changes, and temperature are also shown in Figure 3.11.

We must emphasize here that the PWM-based DCO resolution enhancement technique is an excellent alternative to sigma-delta modulator based DCO resolution enhancement because of: (1) lower power consumption, (2) no spurs are introduced. The PWM based technique in this particular prototype offers resolution improvements that are equivalent to a 3rd order sigma-delta modulator[60]. The power consumption of a typical 3rd order sigma-delta modulator in 65nm CMOS process is > 3mW, and the PWM based technique only consumes 700 μW. Moreover, no spurs are introduced since the PWM technique works at the DCO frequency.

3.8. Conclusion

A number of frequency/delay tuning techniques were analyzed in this chapter. The

power consumption of each of the techniques is compared in Table 3.2.

Table 3.3 compares the frequency resolution of each of the approaches.

Table 3.2: Power comparison of the frequency/delay tuning techniques

Frequency Tuning Technique	Minimum Power	Maximum Power	Comment
Switched-capacitor	$V_{DD}I_{DSAT\_\min\ \_buffer}$	$V_{DD}I_{DSAT\_\min\ \_buffer}$	Resolution is limited by technology.
Parallel-buffer	$I_{DSAT\_\min\ \_buffer}V_{DD}$	$\left(1+\frac{N}{M}\right)I_{DSAT\_\min\ \_buffer}V_{DD}$	Power consumption is prohibitively large if large N is desired
PWM-based	$V_{DD}I_{DSAT\_\min\ \_buffer}$	$3V_{DD}I_{DSAT\_\min\ \_buffer}$	This technique linearizes the trade-off between resolution and power instead of a quadratic relationship.

Table 3.3: Frequency resolution comparison

Frequency Tuning Technique	Frequency Resolution
Switched-capacitor	$\frac{C_{int\_\min\ \_buffer}}{2Kt_{d0}}\left(\frac{-C_{ext\_tunable}}{(C_{int\_\min\ \_buffer}+C_{ext\_fixed}(N)+nC_{ext\_tunable})^2}\right)$
Parallel-buffer	$\frac{I_{DSAT\_\min\ \_buffer}}{KM(MN+1)C_{int\_\min\ \_buffer}V_{DD}}$
PWM-based	$\frac{\Delta PWI_{DSAT\_\min\ \_buffer}}{2KT_{dco}C_{int\_\min\ \_buffer}V_{DD}}$

We have shown that the power-resolution trade-off is a complex parabolic function when switchable-buffers are used, and the frequency resolution is too coarse when only minimum sized switched-capacitance elements are used. However, the PWM-based frequency tuning technique linearizes the relationship – resulting in significant power reductions while maintaining fine frequency resolution.

Chapter 4
An Automatically Placed and Routed 400-460 MHz ADPLL

In 2009, the federal communications commission (FCC) added additional frequency bands to the existing medical implant communication service (MICS) band. A group of five frequency bands between 400 and 457MHz are defined as MedRadio bands. These frequency bands are: 401-106 MHz, 413-419 MHz, 426-432 MHz, 438-444 MHz and 451-457 MHz [61][62]. This has opened to new possibilities for development of wireless communication devices for wireless health monitoring and medical implantable devices. However, these applications require ultra-low power consumption and are typically size constrained. One of the key components of a wireless transceiver, as highlighted in Chapter 1 , is the local oscillator (LO). The LO (typically a phase-locked loop based system) is often responsible a significant portion of power consumption and the overall chip area. At sub-GHz frequencies, the required silicon area of LC-PLLs can be prohibitive and often external components are required. Therefore, all-digital PLL architectures are becoming an increasingly popular way of frequency generation.

All-digital phase-locked loops (ADPLLs) are preferred for frequency generation over traditional analog PLLs to take advantage of process scaling [62]-[65]. ADPLL architectures offer area savings by eliminating large loop filters, reconfigurability of the loop gain and bandwidth, and are mostly portable across processes. However, ADPLL performance inherently suffers due to TDC and DCO quantization errors which contribute to the in-band and out-of-band phase noise [67][66]. Moreover, most ADPLLs use DACs and Δ∑ modulators (DSM) to improve the DCO resolution, which require carefully matched custom design [67]-[76].

The next logical step for ADPLLs is to utilize digital synthesis and automatic place-and-route (APR) flows to simplify the design phase and facilitate easier integration with SoCs. Some traditionally mixed-signal systems such as ADCs and ADPLLs are already being implemented with digital synthesis tools [64][65]. A sub-sampling, integer-N ADPLL is presented in this chapter. What distinguishes this ADPLL from others is that it was completely designed and APR-ed using digital design flows. Secondly, a pulse-width-modulator (PWM) based DCO resolution enhancement technique that replaces the traditional DAC and DSM is discussed. The PWM technique has the advantage of introducing no spurs, and allows DCO tuning with 59kHz steps. An adaptive digital loop filter (DLF) [71] is implemented to allow a large lock-in range as well as have low bandwidth to suppress TDC noise. The ADPLL FoM is -218dB at 403MHz, and covers the entire MedRadio range (401 to 457MHz).

In addition to the innovative approach to reducing the DCO quantization noise, the following architectural modifications were made for further power and area savings.

1. The divider was removed, making it a sub-sampling ADPLL architecture. The benefits of removing the divider are twofold: a) better phase noise since the loop phase noise is typically multiplied by N^2 where N is the divider ratio, b) lower power and silicon area.

2. The ring-based DCO was used as the delay line for the TDC – further reducing the area and power. This is also called an embedded TDC architecture.

4.1. Sub-sampling ADPLL Model

Figure 4.1 shows a model of a typical ADPLL. It consists of a phase detection scheme (typically a time to digital converter), digital loop filter, digitally controlled oscillator, and a divider. In this simplified model, the phase noise can be separated in two parts: 1) DCO noise which dominates the out-of-band phase noise, and 2) loop noise which typically consists of reference noise, TDC noise, and the divider noise.

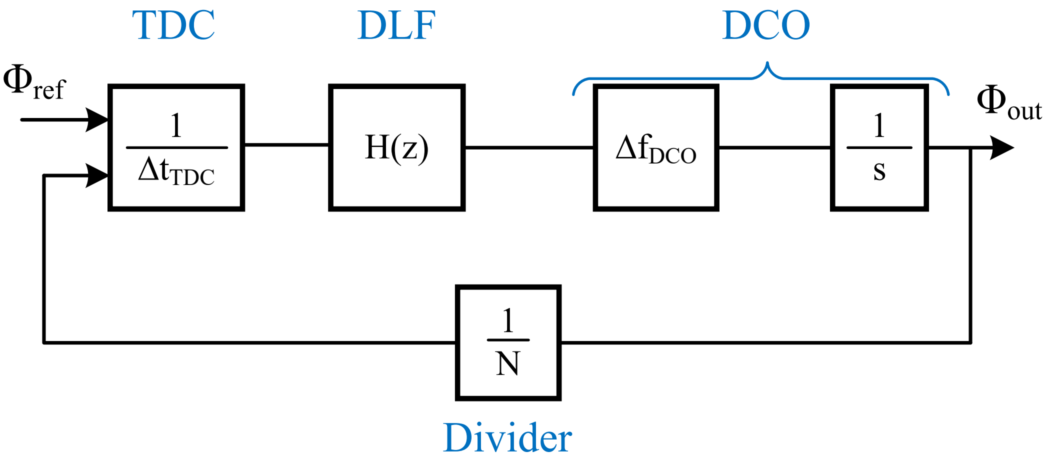


Figure 4.1: Typical ADPLL model

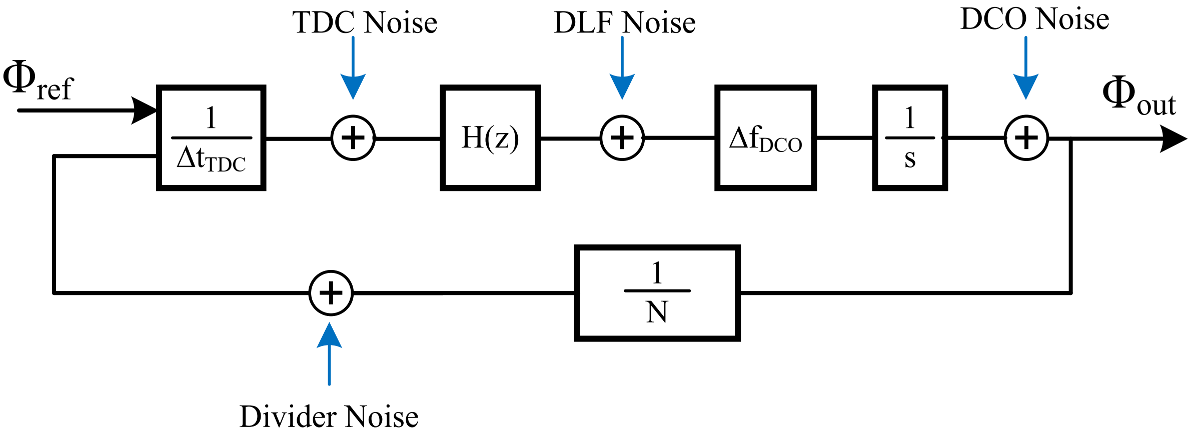


Figure 4.2: ADPLL model with phase noise contributors

Figure 4.2 illustrates how the noise sources in each of the sub-blocks of PLL appear in the loop. Before continuing further, the following assumptions are made in order to simplify the noise analysis.

1. Reference noise is negligible compared to the TDC and divider noise
2. DLF noise is negligible if designed properly (i.e. large bit width)
3. Quantization noise of the TDC and DCO are the two dominant sources of noise

As mentioned earlier, the noise sources can be separated into the in-band loop noise, and out-of-band DCO/VCO noise. Figure 4.3 shows what the individual noise sources look like versus frequency offset, and the overall PLL noise, where f_c is the bandwidth of the PLL. According to [77], an optimized PLL typically is optimized with both noise sources in mind.

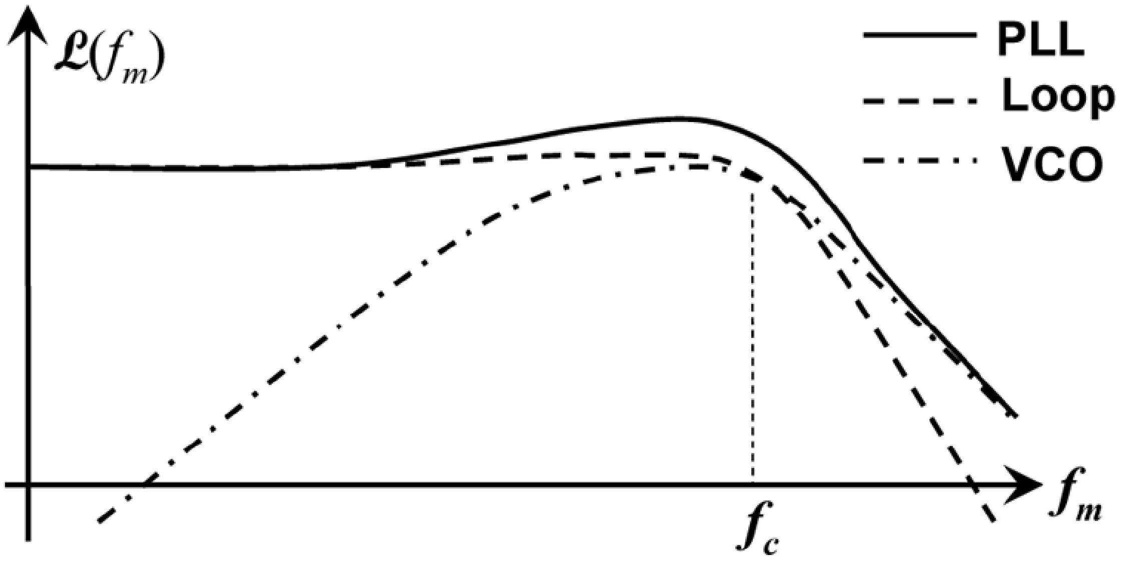


Figure 4.3: Breakdown of two dominant phase noise sources[77]

Many noise reduction techniques have been studied extensively in literature [78]-[80]. As shown in Figure 4.2, phase noise at the input of the divider directly appears at the output of the ADPLL. Therefore, we can calculate the divider-input referred single side band loop noise power as following:

$$\mathcal{L}_{loop} = \frac{1}{2}N^2.\left(S_{ref\ noise} + S_{div\ noise} + S_{TDC\ noise} + \ \Delta t_{TDC}{}^2.S_{DLF\ noise}\right) \tag{4.1}$$

If the divider is removed, the ADPLL becomes a sub-sampling ADPLL as illustrated in Figure 4.4.

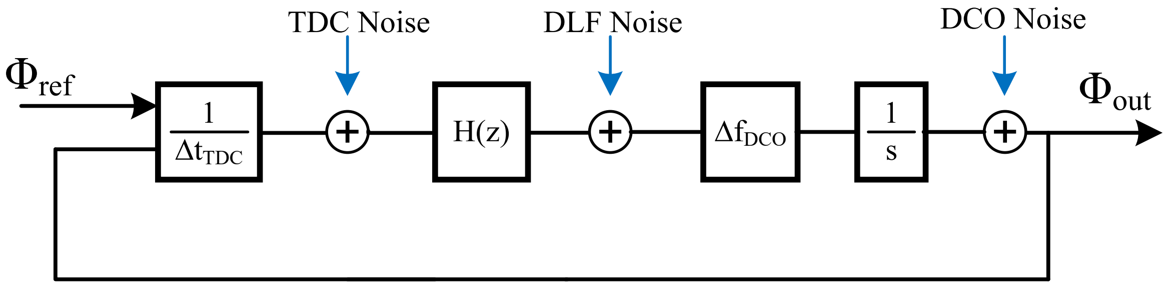


Figure 4.4: Noise model of a dividerless ADPLL

Assuming that the reference noise, and the DLF noise are negligible, then the loop noise power of a dividerless ADPLL becomes:

$$\mathcal{L}_{loop} = \frac{1}{2}.\left(S_{TDC\ noise}\right) \tag{4.2}$$

Therefore, the in-band phase noise can now be optimizing by simply optimizing the TDC quantization noise. The effect of TDC quantization has been studied in literature in great detail. The design equation for ADPLL in-band phase noise due to uniformly distributed TDC quantization noise can be written as [58]:

$$S_{PLL\ in-band} = \left(2\pi.\frac{\Delta t_{TDC}}{T_{DCO}}\right)^2.\frac{1}{12f_{REF}} \tag{4.3}$$

In the equation above, Δt_{TDC} is the resolution of the TDC, and T_{DCO} is the period of the digitally controlled oscillator.

Similarly, the out-of-band phase noise that is dominated by the uniformly distributed DCO quantization noise can be described as:

$$S_{PLL\ out-of-band} = \left(2\pi.\frac{\Delta f_{DCO}}{f_{REF}}\right)^2.\frac{1}{12f_{REF}} \tag{4.4}$$

Therefore, a sub-sampling ADPLL that does not have a divider can be designed using equations 3.3 and 3.4.

TDC Quantization Noise:

The TDC resolution is a function of delay per stage. However, in this particular ADPLL, the ring oscillator is re-used as the delay line. Therefore, the resolution of the TDC is limited by the number of stages of the DCO, and can be expressed as [58]:

$$\Delta t_{TDC} = \frac{T_{DCO}}{2.N_{dco\ stages}} \tag{4.5}$$

However, in an ADPLL, the TDC quantization noise can be further reduced by averaging the output of the TDC.

DCO Quantization Noise:

The DCO quantization noise can be reduced by improving the resolution of the DCO. A number of architectural solutions already exist, e.g. dithering. However, they all come at the expense of added power, and complexity. A novel approach to enhancing the DCO resolution using pulse width modulation (PWM) was developed as a part of this thesis and is further discussed in the later sections.

Disadvantages of Dividerless ADPLL:

A sub sampling PLL is unable to distinguish between different harmonics of the reference frequency which makes it quite challenging to lock and tune the frequency of a sub sampling ADPLL. Therefore, the ADPLL may lock to a false and undesired division ratio. A temporary (transient mode) frequency lock loop solves this problem. The FLL can be shut off once the correct harmonic has been captured. The solution to this problem is discussed in Chapter 5 .

Moreover, the pull-in range of the ADPLL is a function of the reference frequency. In order to collect sufficient information from the sub sampled output frequency, the error frequency must be Nyquist sampled. The relationship between pull-in range and the reference frequency can be expressed as followed:

$$\text{Pull} - \text{in Range} = Nf_{ref} \pm \frac{f_{ref}}{2} \tag{4.6}$$

Where *N* is the division ratio.

Therefore, if an output frequency of 403 MHz is desired, and the reference frequency is 40.3 MHz, the ADPLL can correctly lock to the desired harmonic if the initial output frequency is anywhere between 382.85 MHz and 423.15 MHz. In this ADPLL presented in this chapter, it was possible to set the initial DCO control word to ensure that the correct harmonic was being acquired during locking operation.

4.2. Overall Architecture of the ADPLL

Figure 4.5 shows the ADPLL architecture with PWM-based resolution enhancement technique. It consists of a ten stage ring DCO, embedded TDC, adaptive DLF and DCO controller. The adaptive DLF observes the TDC output (Φ_{err}) over a programmable measurement window, and then a decision by the DCO controller state machine is made whether to increment or decrement the frequency. The default setting for the measurement window is 100 reference cycles, but can be programmed through a scan chain. The DCO controller sends a 20b coarse, 20b fine and 7b ultrafine frequency control word to the DCO. The coarse and fine control bits are thermometer encoded and the ultrafine frequency bits are binary encoded.

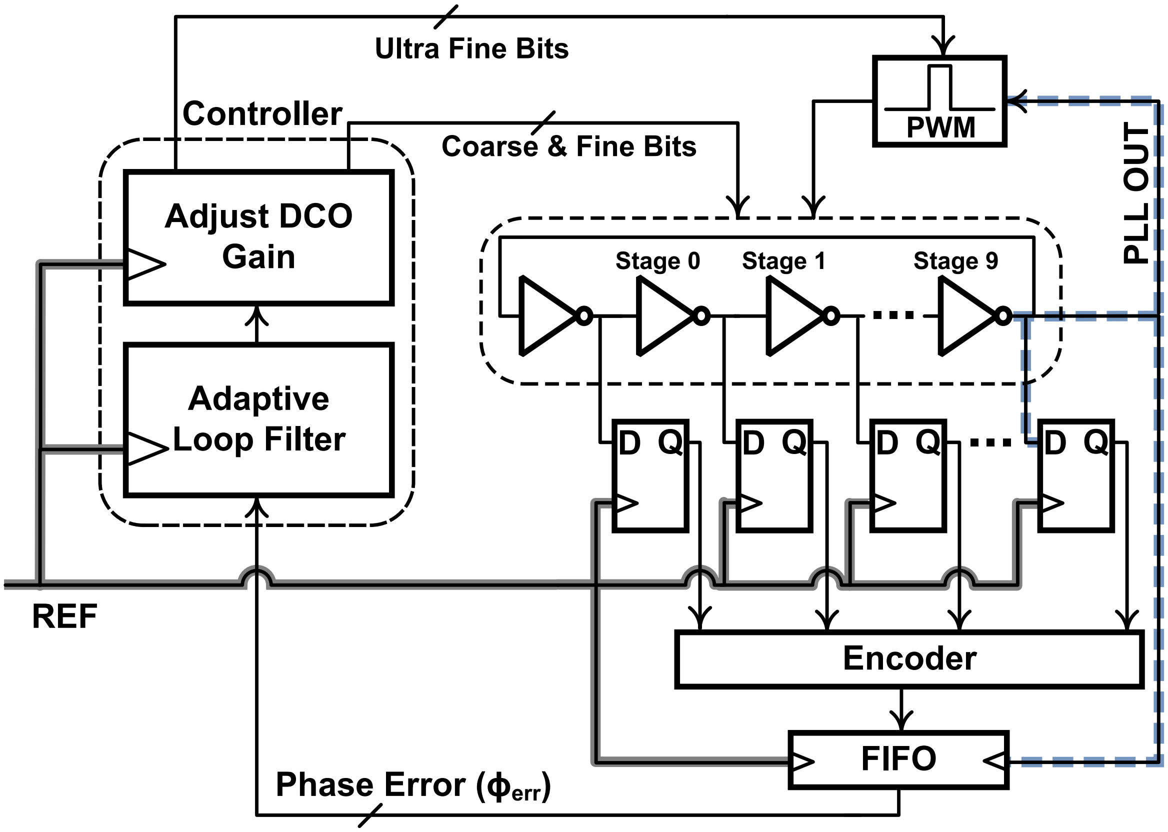


Figure 4.5: The overall ADPLL architecture

The entire ADPLL is cell-based and the layout APR-ed, including the DCO and TDC, which introduces systematic mismatch in wiring capacitance. The most critical is the stage-to-stage mismatch that causes a bounded differential non-linearity in the TDC. The TDC output is processed in the DLF in order to mitigate the effect of mismatches as discussed later.

4.3. Sub-block Design Details

A. Phase Detection Scheme: Embedded TDC

The phase detection scheme is based on an embedded TDC [63]. The embedded TDC samples all ten phases of the DCO every rising edge of the reference, and encodes the error signal into a 5b output (Φ_{err}). If there is a difference between f_{ref} and f_{dco} (Δf), the internal edges of the DCO will slide with respect to the reference edge, and the Φ_{err} observed by the TDC will be a cyclic phase measurement. The slope of Φ_{err} represents the magnitude and sign of Δf. When Δf is small, the TDC resolution is further enhanced by counting (8b counter) the number of f_{ref} cycles for which Φ_{err} stays at one state, and the resultant TDC LSB becomes $T_{\mathrm{dco}}/2^{13}$ (~300 fs at 403MHz). The 5b embedded TDC combined with 8b phase counter in the controller effectively provide 13b phase resolution. Typically, ADPLLs require the TDC step to be normalized to T_{dco}. This is because the TDC step is independent of the DCO frequency, and the loop gain varies as a function of DCO frequency – an undesirable non-linearity. However, in the embedded TDC architecture, the TDC step size depends on the delay per stage of the DCO and therefore, the step size tracks the DCO period and eliminates the need for TDC step normalization.

B. Digitally Controlled Oscillator

Figure 4.6 illustrates the architecture of the DCO with PWM circuit driving the control of all ultra-fine drivers together. The detailed schematic of the cells used in the DCO and the PWM are shown in Figure 4.7 and Figure 4.8. The unit main driver cell is a differential pair with cross coupled PMOS loads which can be turned on/off by the EN signal. The unit switch-capacitor cell is a transmission gate loaded with an NMOS device. The unit ultra-fine driver cell is a 40X weaker version of the main driver. These three are the only custom cells in the entire ADPLL, and have the same pitch as the standard cells. A 7b PWM signal is generated using the same driver and switch-capacitor cells as in the DCO. The DCO features three different step sizes; coarse, fine and ultra-fine. It can be tuned with 9MHz/bit coarse steps, and 1.2MHz/bit fine steps. The coarse steps are set by turning parallel main drivers on/off while the fine tuning is done by enabling parallel switch-capacitor cells. The enable of each coarse/fine cell is independently indexed by the controller. The details of the PWM circuit were discussed in Chapter 3 .

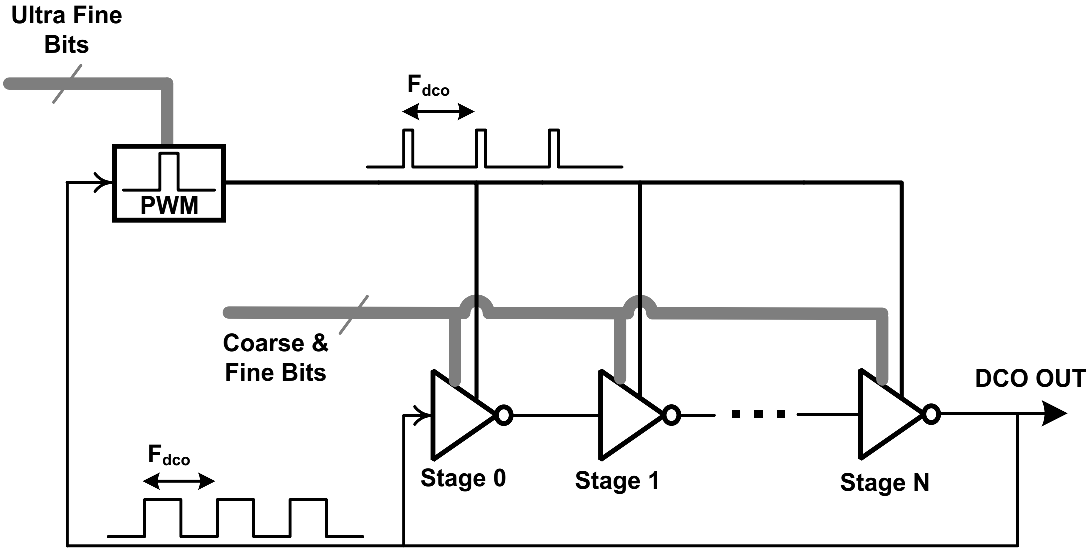


Figure 4.6: Ring DCO circuit diagram

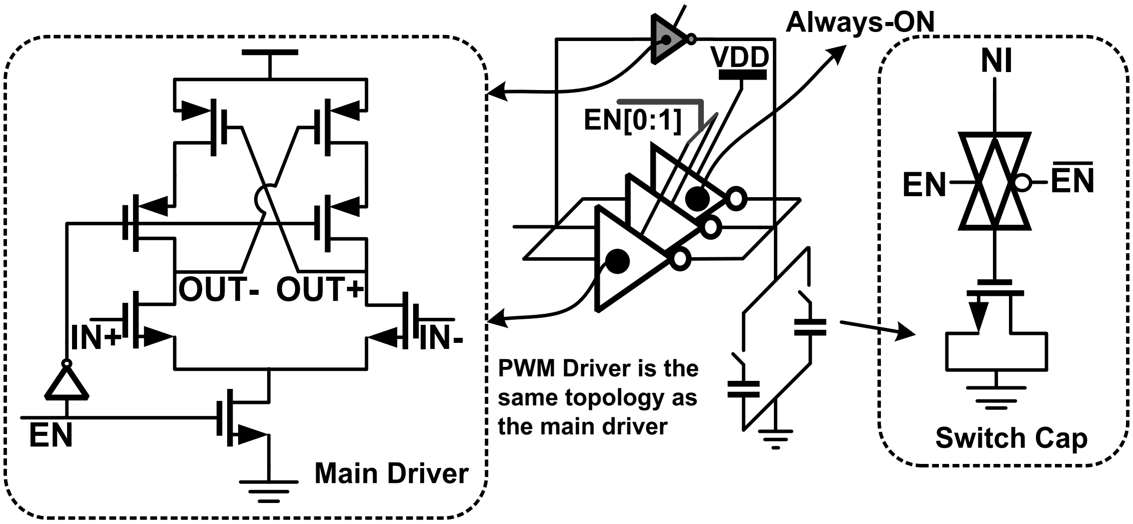


Figure 4.7: Details of one DCO stage

C. Adaptive Digital Loop Filter

The adaptive DLF has low and high gain modes. For large Δf (difference between f_{ref} and f_{dco}), the loop operates in the high gain mode and Δf is measured and used to adjust the loop gain. The loop switches to low gain mode when Δf reaches a target value, after which the gain is adjusted based on the phase measurement. The operation of the adaptive DLF and DCO controller is illustrated by a timing diagram in Figure 3.5. The DLF observes Φ_{err} over a measurement window, and determines the gain mode automatically based on Δf. When Δf is large, larger bandwidth is desired to settle fast. The loop filter measures the number of transitions (N_{trans}) in Φ_{err}, essentially differentiating Φ_{err} to obtain Δf. N_{trans} represents the magnitude, and the direction of the transition (up/down) represents the sign of Δf. Based on N_{trans}, the DLF performs a linear search for the appropriate DCO step size. A programmable N_{thresh} defines the boundary between low gain and high gain modes. Small Δf is defined by the number of Φ_{err} transitions in one measurement window being less than 10. In this case, higher resolution is desired and the DLF automatically switches to low gain mode. In low gain mode, the phase error is measured by counting the number of reference cycles between two transitions of Φ_{err} to get the phase width (Ph_{width}). This represents how long it takes for the DCO edge to slide from one phase to the next; therefore, it represents Δf. Once the Ph_{width} is known, a linear search chooses one of four small DCO step sizes.

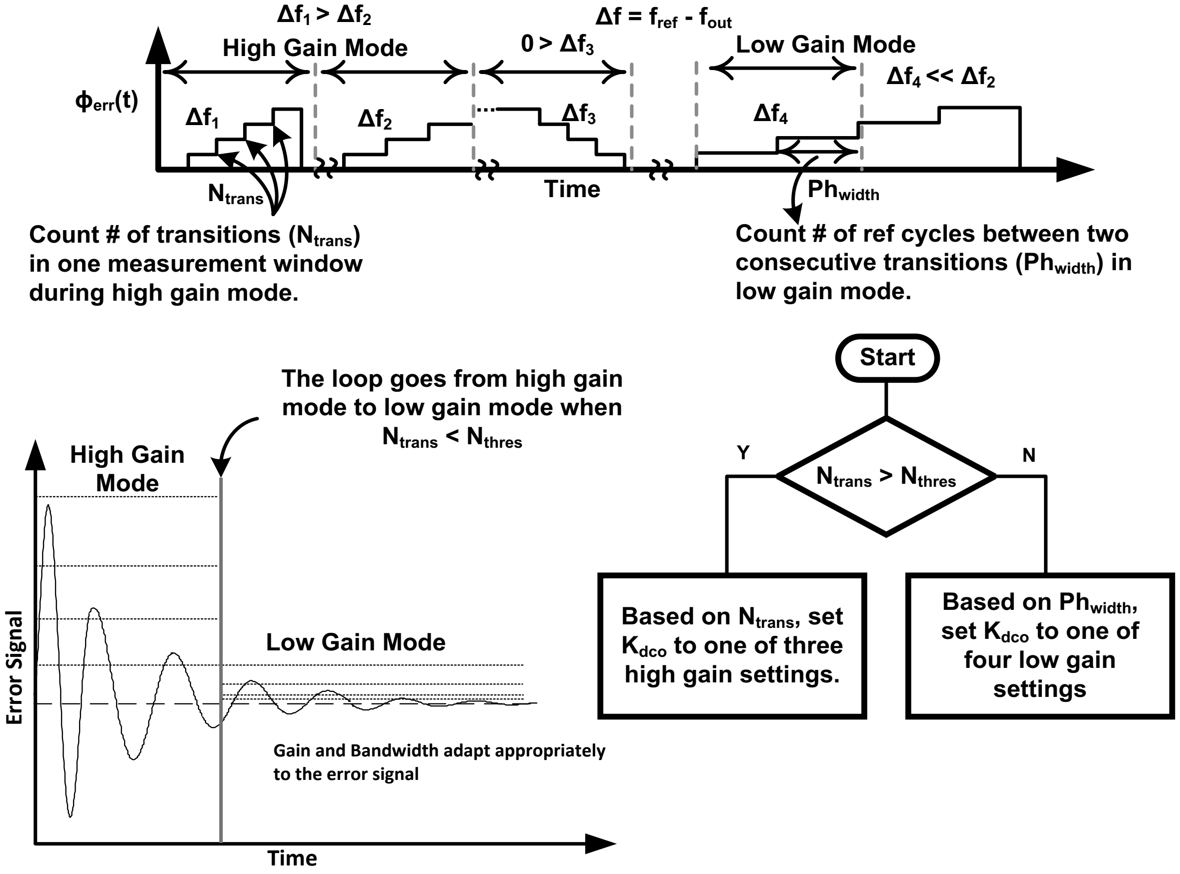

Δf1 > Δf2
High Gain Mode
Δf = fref - fout
0 > Δf3
Low Gain Mode
Δf4 << Δf2
Φerr(t)
Δf1
Δf2
Δf3
Δf4
Ntrans
Time
Phwidth
Count # of transitions (Ntrans) in one measurement window during high gain mode.
Count # of ref cycles between two consecutive transitions (Phwidth) in low gain mode.
The loop goes from high gain mode to low gain mode when Ntrans < Nthres
High Gain Mode
Low Gain Mode
Error Signal
Gain and Bandwidth adapt appropriately to the error signal
Time
Start
Y
N
Ntrans > Nthres
Based on Ntrans, set Kdco to one of three high gain settings.
Based on Phwidth, set Kdco to one of four low gain settings

Figure 4.8: Adaptive Loop Filtering Technique

The entire DLF is implemented on chip. This adaptive DLF allows a lock-in range of ±30 MHz. When the PLL output is locked to the reference, the Φ_{err} behaves like a bang-bang phase detector and frequency is controlled with an ultra-fine LSB (59 kHz) around the desired frequency.

In the high gain mode, the effect of stage-to-stage mismatch is alleviated by measuring the slope of Φ_{err} (N_{trans}) over multiple reference cycles. This way, any differential non-linearity in the DCO due to stage-to-stage mismatch is averaged because the total delay (sum of individual delays) always equals T_{dco}.

4.4. DCO Resolution Enhancement Technique

A novel resolution enhancement technique was used in this ADPLL. The details of the technique were discussed in Chapter 3 . This technique allowed 20X improvement in resolution, but only at 50% power penalty.

4.5. Design Methodology

The design methodology used to implement this ADPLL is discussed has been discussed in detail in Chapter 2 . We utilized automated digital flows to accelerate the design phase. The first step is to identify the core, tri-state unit cells that will be arrayed to form a digitally tunable delay element, and design and layout these unit cells with the standard cell pitch. The unit cells are roughly the size of a D flip-flop standard cell. These cells are then integrated with the synthesis and APR flows. Once the cells are integrated, an HDL description of the entire ADPLL is used to synthesize and APR it. The tools are also used to create macros for sub-blocks to achieve a moderate amount of matching in the layout. For example, in this ADPLL, a macro for one stage of the DCO was first APR'd that included three main drivers and two switch capacitor cells. This macro was then instantiated in HDL ten times to create a ten-stage DCO. This methodology significantly accelerates the design phase because most of the design decisions are made at the architectural level; therefore design iterations are completely automated. Moreover, the number and complexity of required design rule checks grows exponentially with scaling, but with this methodology, this challenge is mostly handed over to the tools.

4.6. Measurement Results

This ADPLL performs integer-N synthesis without a divider by subsampling the TDC output for division ratios greater than one. The division ratio (N) can be programmed by a frequency control word in the controller. The in-band phase noise is -98dBc/Hz for f_{ref}=403MHz (N=1 and BW = 140 kHz) and -87dBc/Hz for 40.3MHz (N=10 and BW = 40 kHz).

Figure 4.9 shows the output spectrum for f_{out}=403MHz for 403MHz and 40.3MHz reference frequencies. The PLL output in lock state with and without PWM-based resolution enhancement is shown. This results in 14dB and 11dB improvements in in-band phase noise for 403MHz and 40.3MHz reference frequencies, respectively, with a measured rms period jitter of 7.9 ps and 13.3 ps. According to equation 4.4, the improvements should be 24 dB & 21 dB. This discrepancy can be attributed to ring oscillator device noise becoming a dominant factor at far out frequencies. Table 4.1 shows a detailed comparison of various measured performance metrics to state-of-the-art work. The measured phase noise of the ADPLL is given in Figure 4.10. The model represented the ADPLL very closely since the calculated in-band phase noise was -95 dBc/Hz (-89dBc/Hz measured), and out-of-band was -150 dBc/Hz (-141 dBc/Hz measured). In Figure 4.10, measured phase noise is overlaid with the mathematic model.

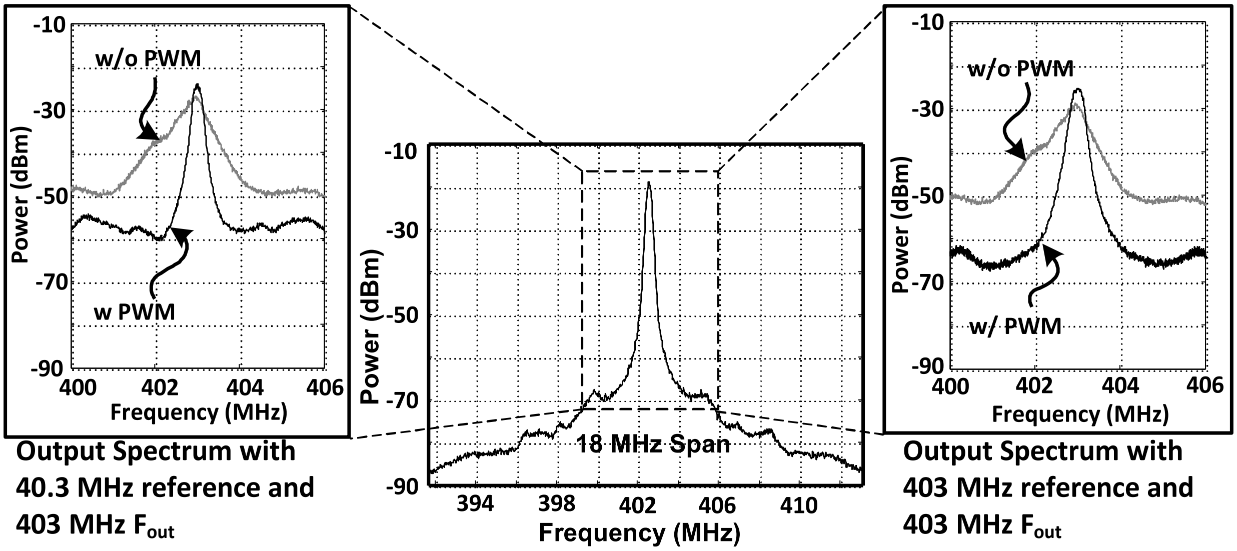


Figure 4.9: The output spectrum of the PLL with and without the PWM

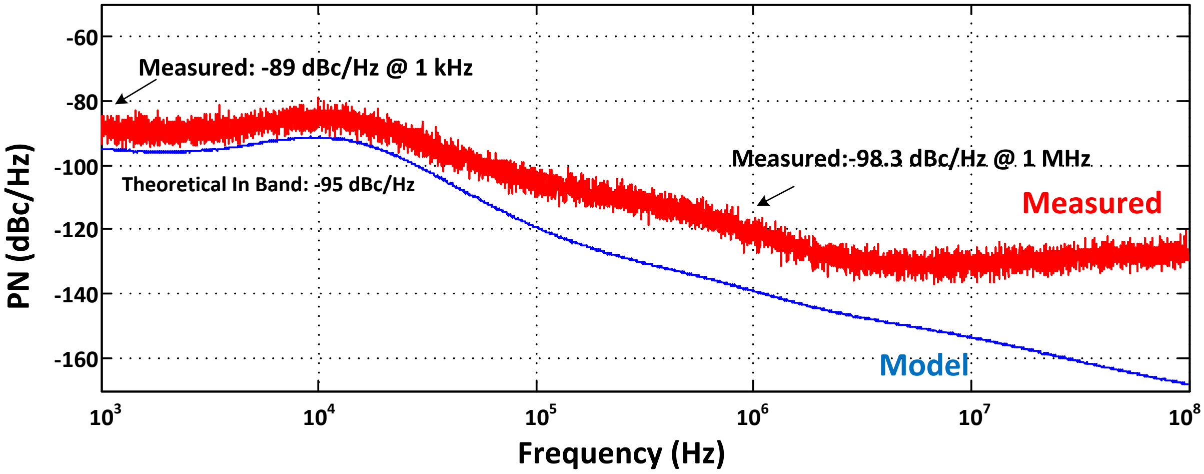


Figure 4.10: Phase noise of the ADPLL with 403MHz Fref

The ADPLL is implemented in a 65nm CMOS process, and occupies an active area of 0.1 mm². It covers the MedRadio bands and consumes 2.1mA and 3.3mA from a 1.0V supply for *N* of 10 and 1 respectively. The PWM block consumes 770μW and occupies 93x110μm. The overall FoM of this PLL is compared with state-of-the-art ADPLLs in Figure 4.11. The die photo is also shown in Figure 4.12. Table 3.1 compares more specific performance metrics of this work to some recently published state-of-the-art ADPLLs.

Table 4.1: Performance comparison with state-of-the-art work

	This Work		[63]	[64]	[67]
F_{REF} (MHz)	403	40.3	26	544	108/72/36
F_{OUT}(MHz)	403	403	800	0.7 - 3.5	3100
RMS Jitter	7.9 ps	13.3 ps	21.5 ps	1.6 ps	1.01 ps
PN (dBc/Hz)	-98 @ 1MHz	-87 @ 1MHz	-98 @ 1MHz	-116 @ 1MHz	-
Area	0.1 mm^2	0.1 mm^2	0.05 mm^2	0.36 mm^2	0.32 mm^2
Power	3.3 mA	2.1 mA	2.66 mA	1.6 mA	27.5/26.8/
VDD	1	1	1.1-1.3	1	1.2
Architecture	ADPLL	ADPLL	ADPLL	Highly Digital	ADPLL
DAC & ΔΣ	No	No	DAC & ΣΔ	Multiple DACs	DAC
Technology	65 nm	65 nm	65 nm	90 nm	65 nm

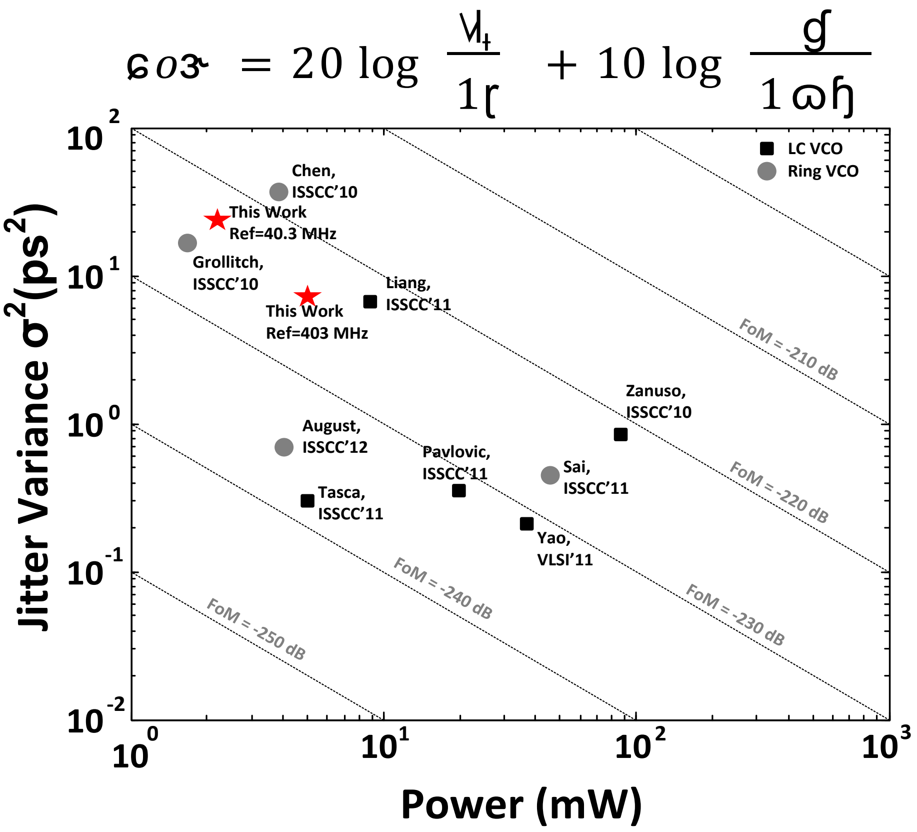


Figure 4.11: Figure of Merit Comparison

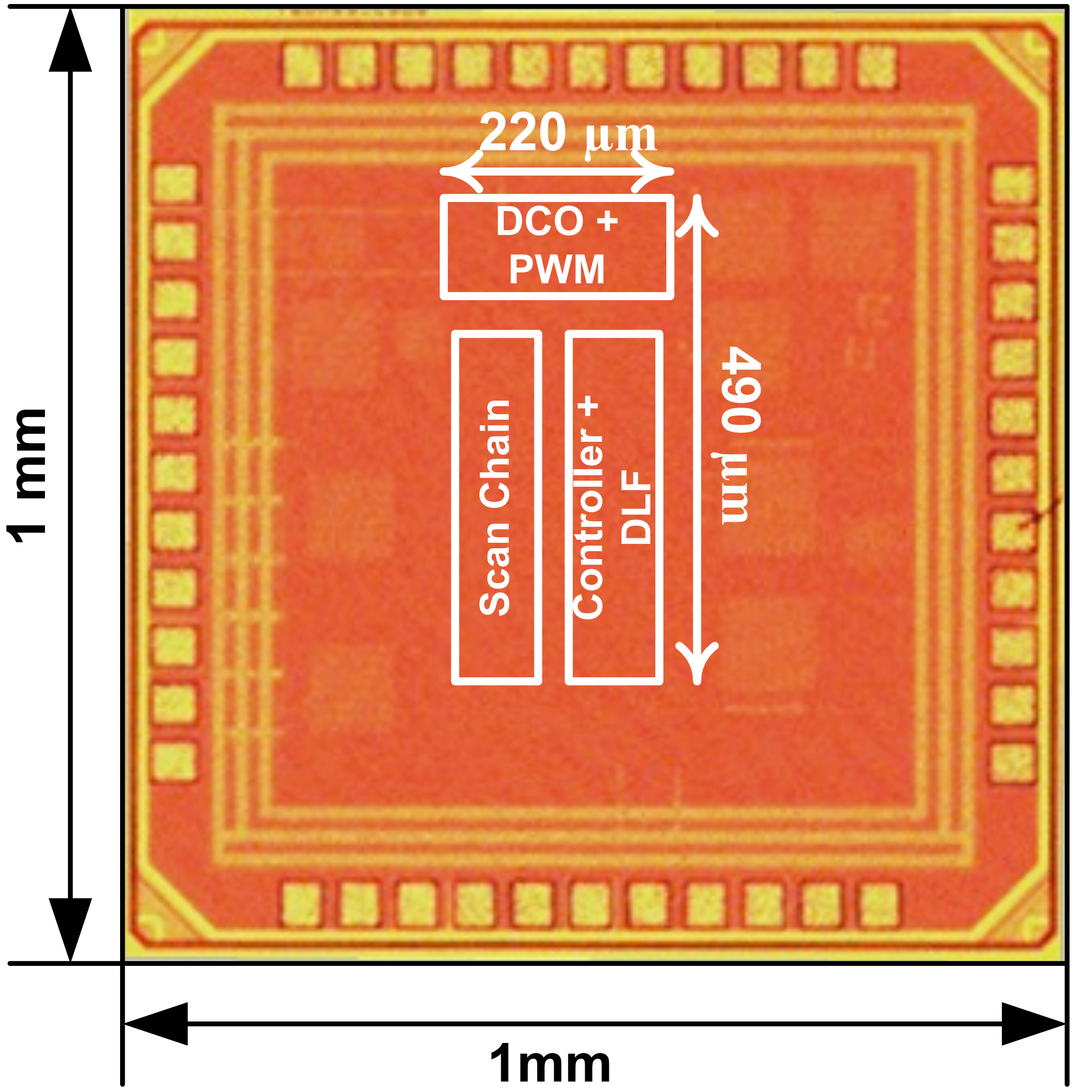


Figure 4.12: The die photo of the ADPLL

4.7. Conclusion

A subsampling integer-N ADPLL that was completely designing used digital design flows was presented. This methodology significantly simplifies the design phase for ADPLLs and is amenable to process scaling. A PWM based DCO resolution enhancement technique is introduced which improves the DCO resolution to 59 kHz/LSB. This resolution enhancement is implemented at the DCO frequency, unlike traditional dithering, and therefore does not introduce any spurs. The FoM of this APR'd ADPLL is -214 dB, which is comparable to state-of-the-art ADPLLs not using digital design flows.

Chapter 5

An Ultra-Low Power Near-Threshold Clock Generator

Significant research efforts are being focused on ultra-low power (ULP), small form factor mobile devices for applications such as health monitoring and the internet of things. These applications seek to extend battery life and/or achieve energy autonomy through energy harvesting and ULP design. This places significant constraints on the active and sleep power consumption of every component, requires higher levels of integration, and ULP design [81],[83]. Reducing the supply voltage (V_{DD}) of digital circuits to the minimum energy point, typically near or below V_{th}, is an effective way to save power. In near-V_{th} computing (NTC), the clock frequencies are typically below 1 MHz, however exponential variation in gate delays of sub-V_{TH} digital circuits exacerbates timing violations, leading to large design margins that offset the power benefits of NTC operation. This makes NTC circuit design particularly challenging since traditional design techniques don't produce the same robust operation that they do in super-V_{th} operation. An architectural solution to this is *dynamic voltage and frequency scaling* based on workload and PVT variations. For NTC, this requires a low-voltage, low-power, programmable and stable clock in the sub-1MHz range. A number of sub-μW clock generator (CKGEN) solutions have already been reported, but they all lack programmability and therefore cannot offer dynamic frequency scaling [84]-[88].

A popular solution for clock programmability in microcontrollers is to generate the highest desired frequency with a crystal oscillator and then use a divider to generate lower frequencies. However, this is not a low-power solution, and cannot achieve the best possible performance as the phase noise degrades proportional to N^2, where N is the divider ratio. IoT applications will demand a low-cost solution, which for IC design translates to small form factor, ease of integration and test, and minimal off-chip components. For these reasons, all-digital architectures leveraging the digital design flow are highly desirable. In this chapter, we present a 187kHz to 500kHz ADPLL-based CKGEN that consumes 300nW from a 0.5V VDD, has a jitter <0.1% and was implemented in a 0.13μm process. The entire ADPLL was completely implemented using standard digital design flows and automatic place and route (APR). Moreover, an integrated crystal oscillator (31.25 kHz) is included and serves as the F_{ref} for the PLL. Therefore, this is a complete CKGEN solution for ULP NTC platforms.

5.1. Clock Generator Architecture

Figure 5.1 shows the overall architecture of the CKGEN which consists of an off-chip crystal with an integrated on-chip oscillator, and a dual loop ring-based ADPLL which was completely implemented using a digital CAD flow and APR. The ADPLL features a coarse frequency acquisition loop and a fine phase locking loop.

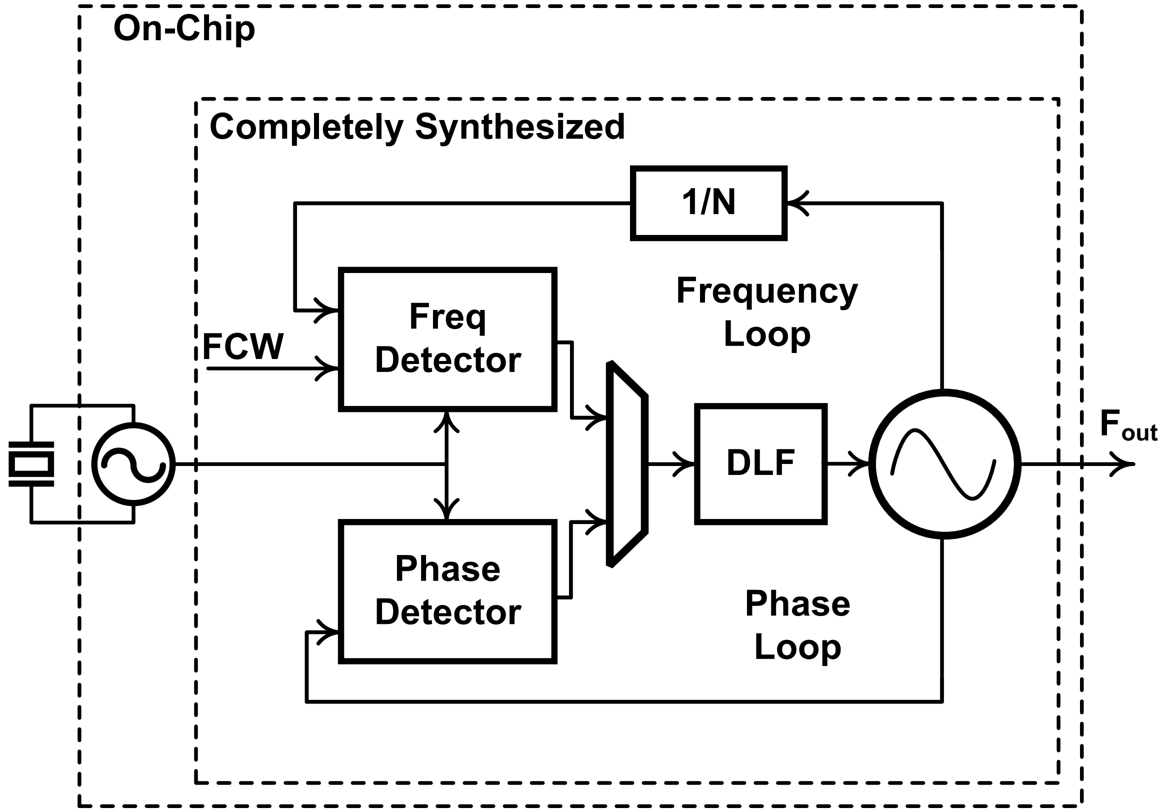


Figure 5.1: The overall architecture of the clock generator

The feedback loops are mutually exclusive, and the loop-select state machine, as illustrated in Figure 5.2, toggles between the loops as needed. The coarse loop consists of an edge combiner to multiply the DCO frequency by 8 and a fast frequency counter converts the DCO frequency to a digital number. The frequency detector logic compares the counter value with the desired frequency control word and sends a correction to the DCO controller. The loop select state machine monitors the previous four corrections and determines which loop should be enabled. When the frequency error is below 4 kHz, the phase loop is enabled and the frequency loop is power-gated to save power. Since the phase loop is dividerless, the phase error is sub sampled which could result in locking to an incorrect harmonic. The oversampling in the frequency loop ensures that the loop converges to the correct harmonic before switching over to the phase loop. A minimum oversampling ratio of 4 is required for correct PLL operation, but oversampling by 8 was implemented by making a power/speed trade-off and to guarantee correct locking. The phase loop consists of an embedded TDC which reuses the ring oscillator as the delay line, resulting in power and area savings. The phase controller compares the phase to the F_{ref} and sends the correction to the DCO controller. In the lock state, only the phase loop is on, which does not have a divider, allowing further power savings without compromising performance or programmability of the ADPLL.

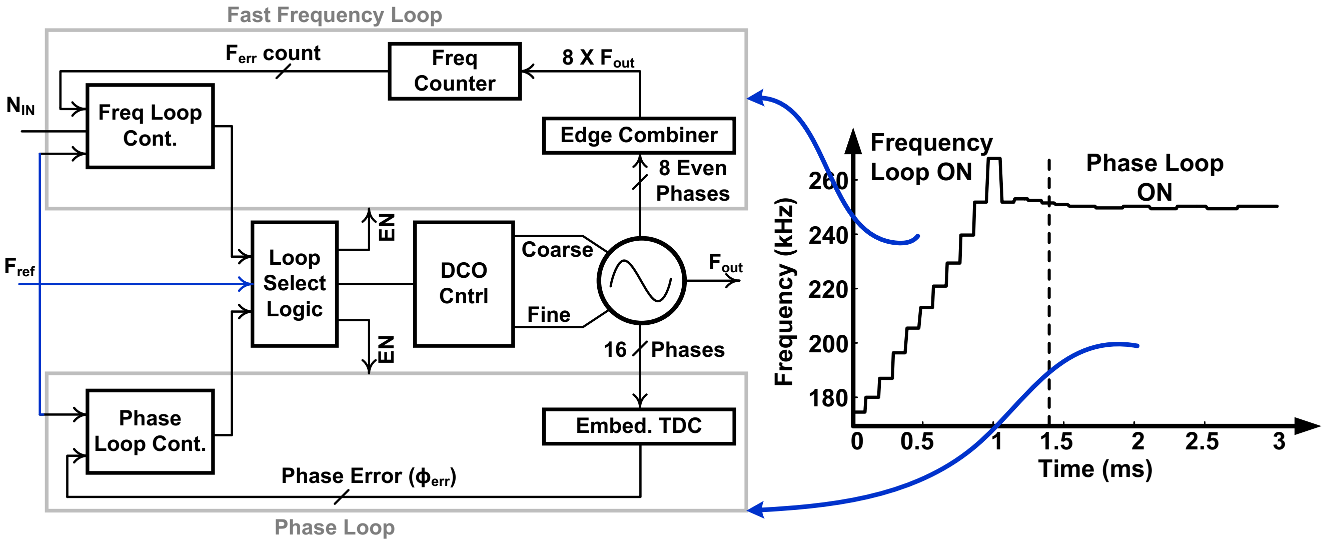
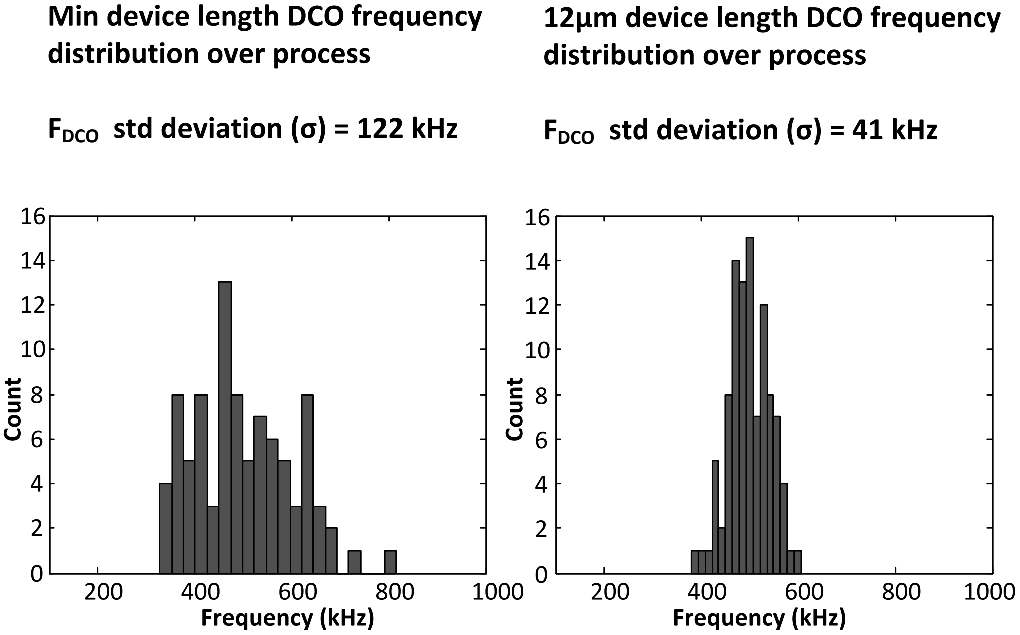


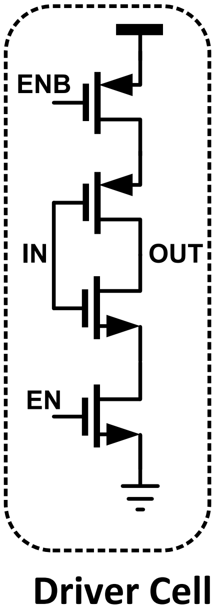
Figure 5.2: Details of frequency and phase loops and the step response

The fast locking algorithm first performs coarse tuning within F_{ref} cycles. The combination of an edge combiner and the frequency counter provides an oversampled version of the DCO period. In other words, the frequency information handed to the frequency controller each F_{ref} period has a resolution of $F_{ref}/8$. Therefore, when the frequency loop finishes acquisition, the output frequency is within 4kHz of the desired output frequency, at which point the phase loop takes over and can achieve lock within 4 cycles. In total, the locking time for the largest frequency step is 38-45 F_{ref} cycles. The PLL step response is shown in Figure 5.2 which illustrates how lock is achieved with frequency and phase loops.

5.2. Near Threshold Design for Ultra Low Power

A major goal of any CKGEN is to provide sufficient tuning range to account for PVT variations in the oscillator and provide a stable, desired frequency. PVT variations are magnified in near or below V_{th} operation, and large PVT variations require an oscillator with large tuning range, which results in higher power consumption. Therefore, it's desirable to reduce the impact of PVT variations on the DCO frequency as much as possible to shrink the required DCO turning range, thus lowering power. The PVT variations mainly impact the overdrive voltage ($V_{ov} = V_{GS}- V_{th}$), and therefore the drive strength of the delay cells in a ring oscillator. One option is to use zero-V_{th} devices if available in a process, but this causes an undesirable increase in the leakage power, a problem when the device is in sleep mode. Another technique that simultaneously improves performance while reducing leakage power is termed *dynamic V_{th}*, where the body and the gate of the devices are tied together. This technique allows low ON-state V_{th} and high OFF-state V_{th}. However, this technique requires a triple well technology which makes integration of custom cells in to the digital standard cell library challenging, and therefore the ADPLL cannot be completely synthesized. Hence, a different solution is required. It is known that V_{th} decreases as the channel length is increased, and that longer channel length results in lower leakage power. Figure 5.3 lists the squared overdrive voltage, V_{ov}^2, in two extreme operating corners (SS, 0.45V, 85°C and FF, 0.55V, -40°C). As shown Figure 5.3, V_{ov}^2 is 20 times larger in the fast corner than in the slow corner for a minimum length device. However, V_{ov}^2 only increases by 1.8X between these corners when 12 μm length devices are used. Figure 5.3 also compares the frequency distribution over process for two different near-V_{th} 500kHz DCOs. The frequency spread over process for a DCO with minimum sized inverters is 3 times as large as that of a 12 μm inverter based DCO. This means that the DCO tuning range to calibrate out the PVT variations can shrink by roughly 3X by using long-channel devices, resulting in significant power savings.





Parameter	@ minL		@ L= 12 μm	
	Slow	Fast	Slow	Fast
$V_{ov}^2 = (V_{GS}\text{-}V_{th})^2$	0.0036	0.0729	0.1369	0.2421
V_{ov}^2 Ratio	20.25		1.8	

Figure 5.3: Effect of increasing transistor length on the PVT variations

5.3. Sub-block Design Details

Further details of the DCO, TDC and the edge combiner are given in Figure 5.4. The DCO is a 17 stage single ended ring oscillator, and each stage consists of an always-on inverter and one switchable inverter for coarse frequency tuning. Each stage also has four switch-cap cells which allow fine tuning with 1 kHz steps on average. These unit tri-state inverter and switch cap cells were integrated with the standard cell library using the same pitch as the digital cells. The rest of the ADPLL only uses standard cells. The entire ADPLL was then described using structural and behavioral Verilog, synthesized, and APR'd using standard digital CAD tools.

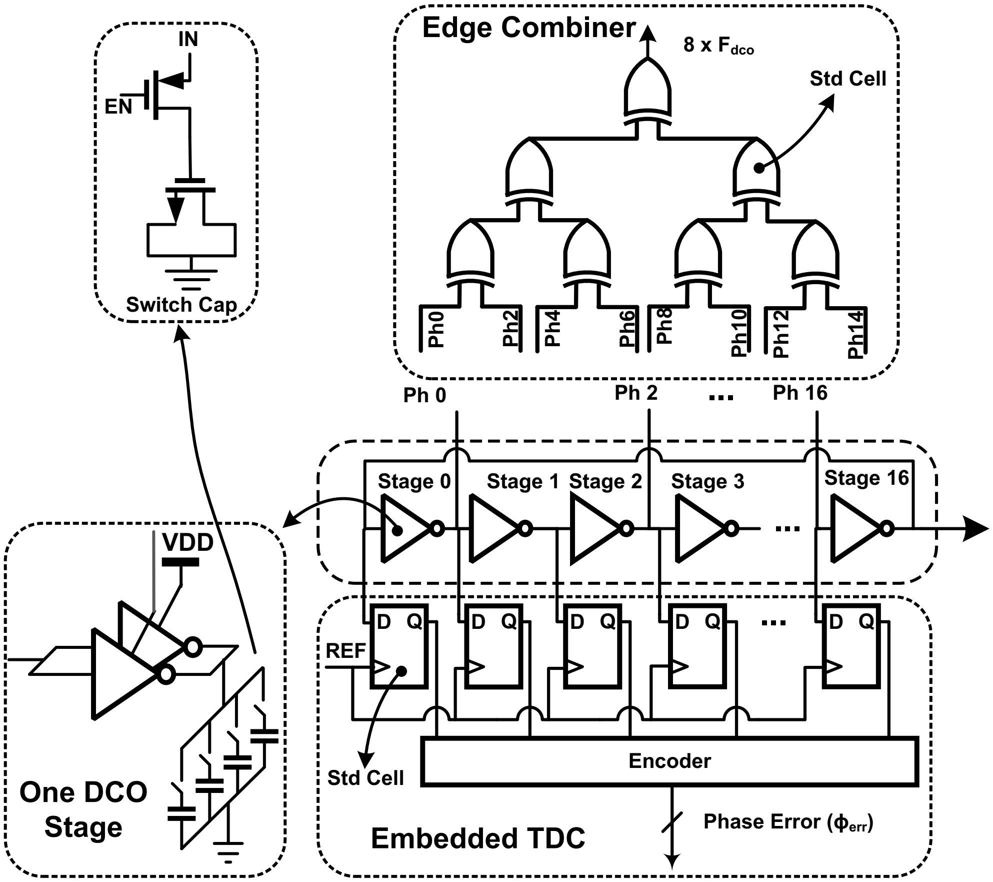


Figure 5.4: Details of DCO, TDC and the edge combiner

An integrated crystal oscillator serves as the F_{ref} for this PLL (31.25 kHz), making this a complete CKGEN solution. The integrated crystal oscillator consumes 24 nA, using a technique to automatically tune the feedback amplifier to just cancel the loss term of the crystal [57].

5.4. Measurement Results

Figure 5.5 shows a sample measured output spectrum of the clock for N=11. As can be seen in Figure 5.5, there are no in-band spurs. The reference spurs are 55 dB below the carrier.

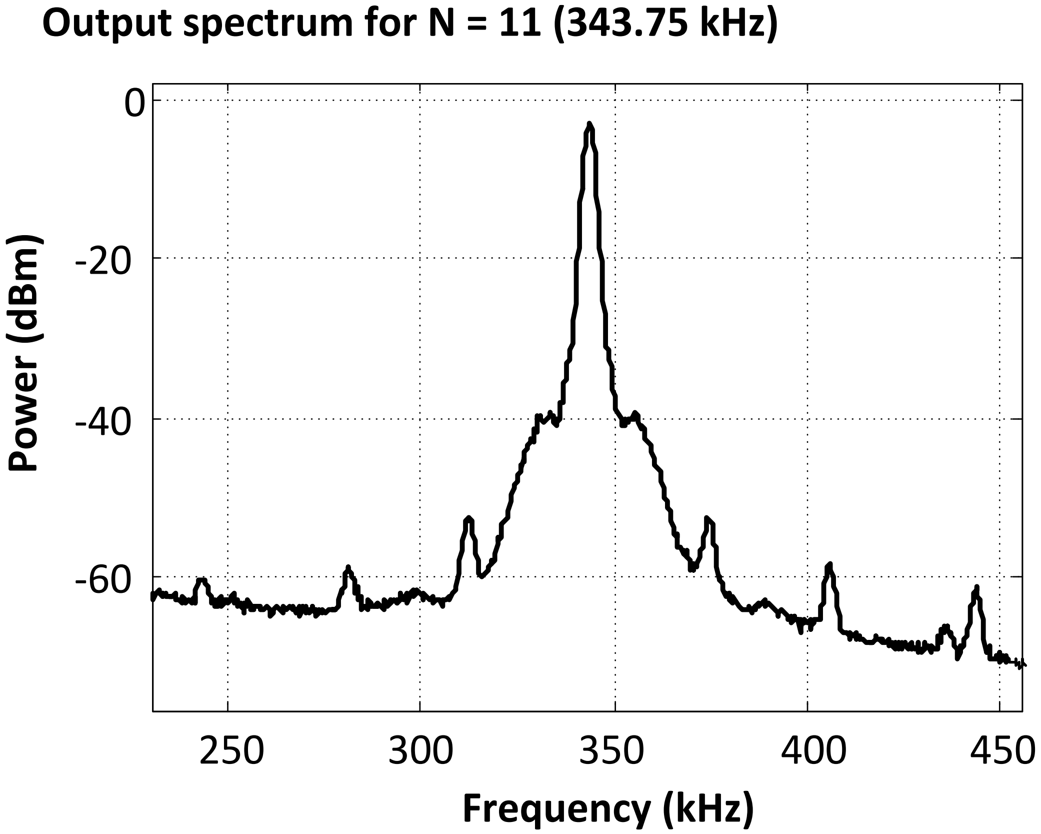


Figure 5.5: Output spectrum of the clock generator for N = 11

In addition to measuring the output spectrum, power and jitter as a function of frequency (or division ratio N) was also measured. Since near-threshold designs are particularly prone to PVT variations, the clock generator was also tested across different voltage and temperature operating conditions.

Power vs. Frequency

Figure 5.6 shows the power of the entire clock generator as a function of frequency. The three curves represent power measurements at typical, fast, and slow operation conditions. In the fast condition, the voltage supply was increased by 10% and the system was cooled down to 0 C. Similarly, in the slow condition, the supply voltage was reduced by 10% and the system was headed up to 85 C. The nominal or typical condition is 0.5V supply and room temperature. The power in the typical condition ranges from 302 nW (@ 187.5 kHz) to 590 nW (@ 500 kHz).

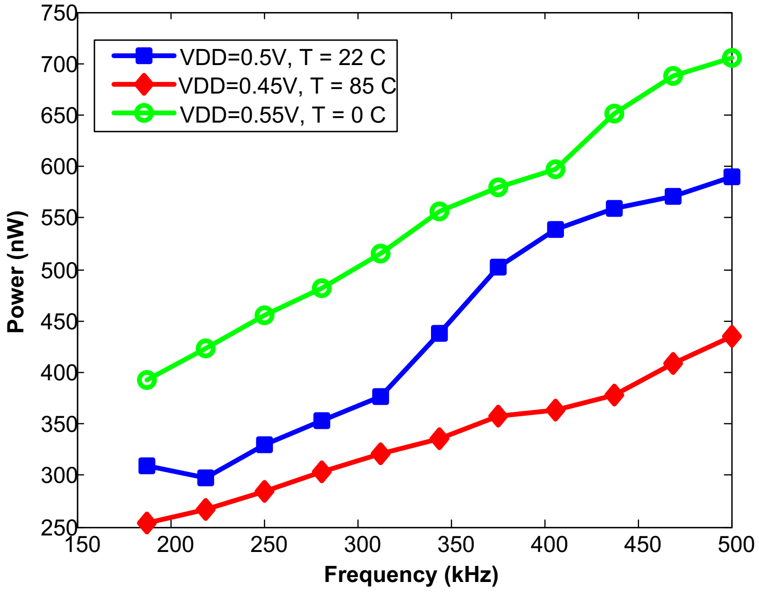


Figure 5.6: Power vs. frequency for the entire frequency range

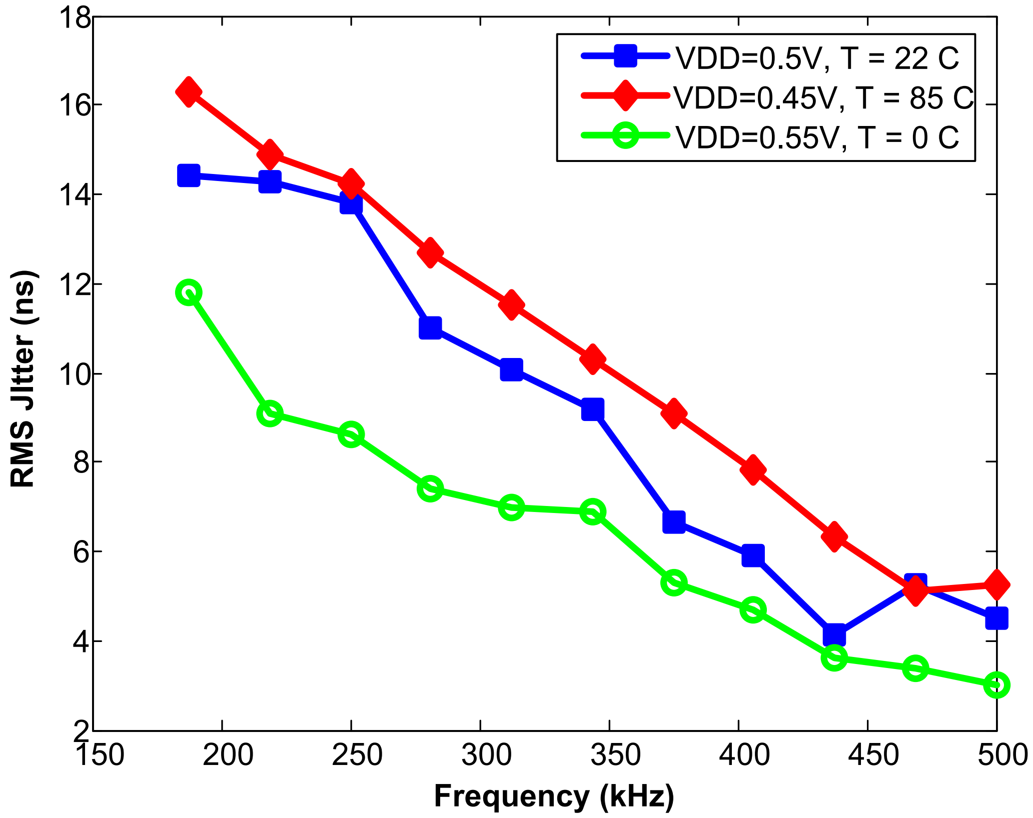


Figure 5.7: RMS Jitter vs. Frequency of the Clock Generator

Jitter vs. Frequency

The RMS jitter is a measure of the quality of a clock signal by quantifying the uncertainty in the time of each edge of the clock. Figure 5.7 shows the rms jitter measurements over three different corners for the entire frequency range. In addition, the measured peak-to-peak jitter over the same operating conditions is shown in Figure 5.8. It is worth noting here that the jitter measurements are those of the entire CKGEN system (crystal + PLL) and the RMS jitter at the lowest frequency is 0.025%.

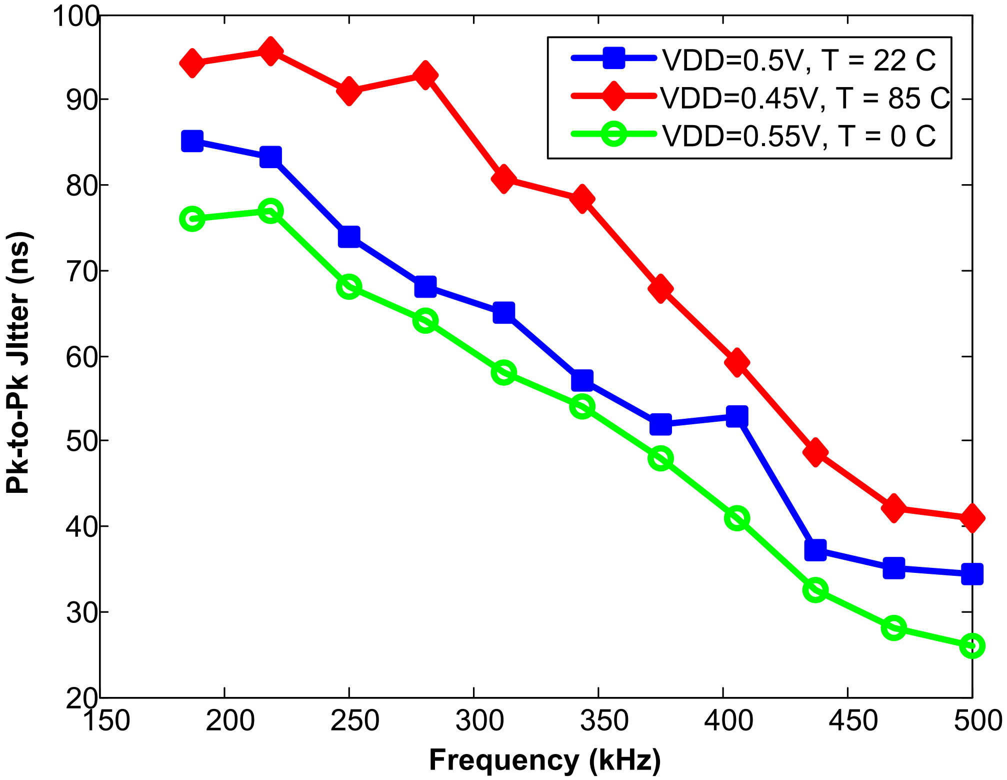


Figure 5.8: Peak-to-Peak Jitter vs. Frequency of the Clock Generator

Jitter Over Process

Since PVT variations are the biggest threat to NTC, six different chips were tested to observe any process variations. The measured peak-to-peak and RMS jitter for six chips are given in Figure 5.9.

Measured phase noise of the clock generator is shown in Figure 5.10.

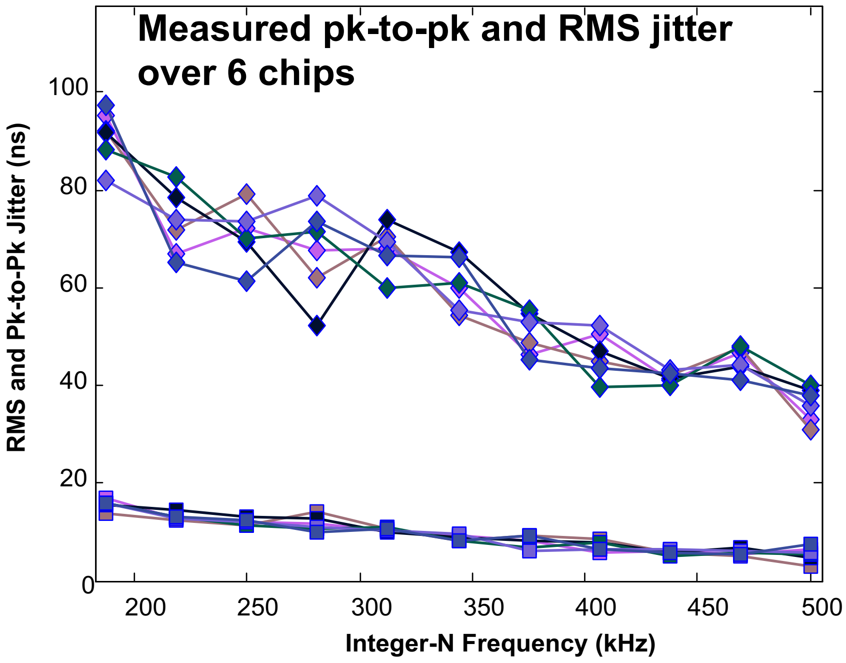


Figure 5.9: Pk-Pk and RMS jitter measured for six chips

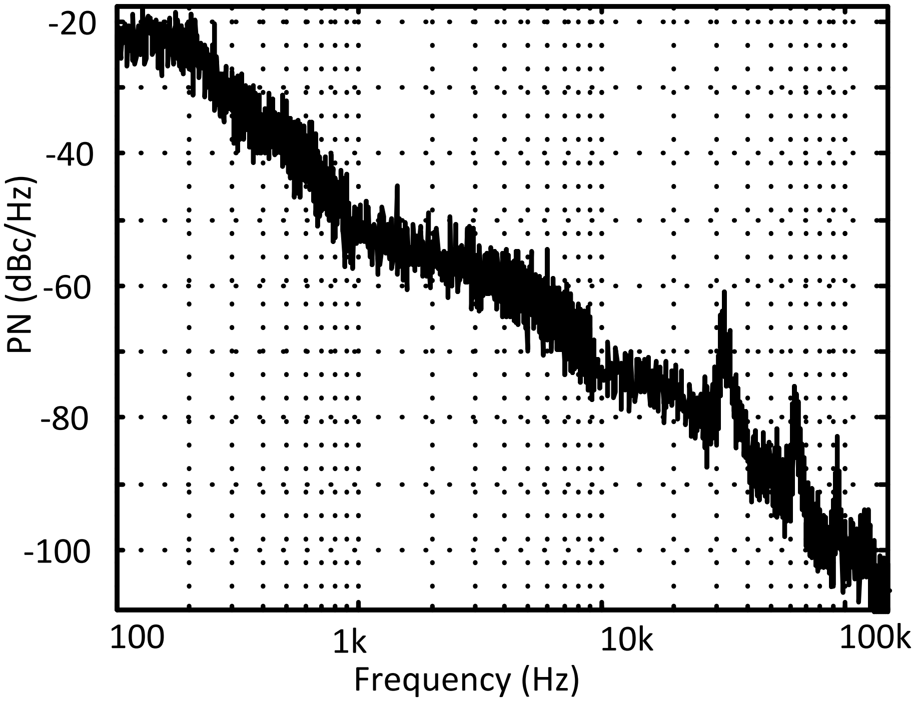


Figure 5.10: Clock Generator phase noise with N = 11

Comparison with the State-of-the-art

Finally, various performance metrics of this work are compared to the state-of-the art published in recent conferences. Table 4.1 shows that this CKGEN is the most efficient at 1.1pJ/cycle.

Table 5.1: Comparison of the CKGEN to the state-of-the-art work

Performance Metric	This Work		[53]	[59]	[61]
	@ Min F_{out}	@ Max F_{out}			
VDD	0.5 V		1	-	1
Process	0.13 μm		90 nm	55 nm	90nm
Area (mm^2)	0.07 (PLL) + 0.13(XO)		0.27	0.16	0.037
F_{out}	187.5 kHz	500 kHz	5 MHz	216 MHz	2GHz
RMS Jitter	12.3 ns	4.7 ns	49.7 ps	8.05 ps	1.6 ps
Power	300 nW	570 nW	11.3 μW	10.5 mW	7 mW
Energy/Cycle (pJ)	1.6	1.1	2.26	48.6	3.5
Architecture	ADPLL, APR-ed, + XO		DCO only	-	DLL
F_{ref}	31.25 kHz		N/A	27 MHz	500 MHz
Reference Included	Yes		No Reference	No	No

Lastly, the die photo is given in Figure 5.11. The PLL occupies 300μm x 300μm, and the integrated crystal oscillator occupies 200μm x 65μm.

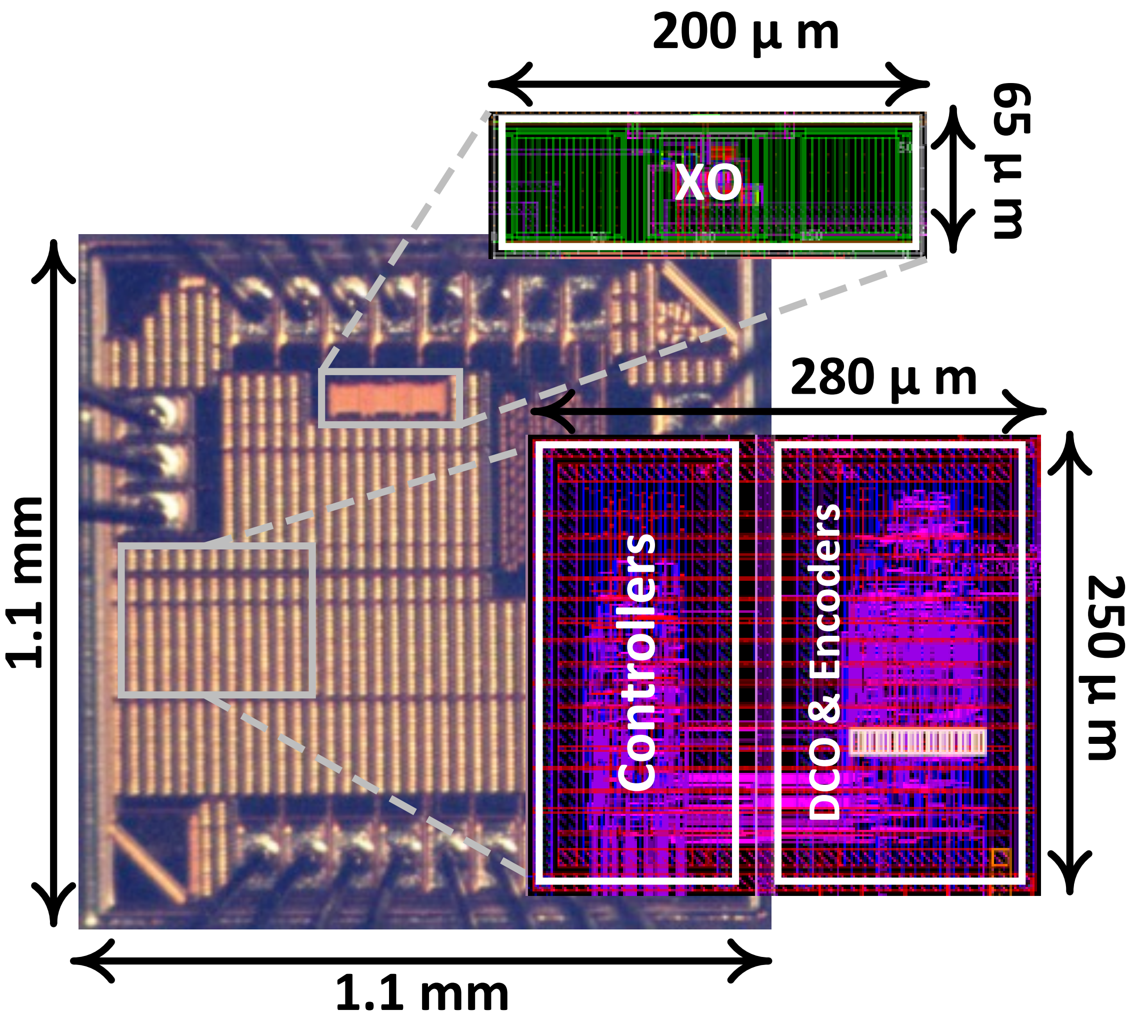


Figure 5.11: The die photo of the CKGEN

5.5. Conclusion

An ultra-low power clock generator for internet of things applications is presented. The clock generator is a PLL based clock generator and consumes 300 nW @ 200 kHz with 0.025% jitter. Moreover, the entire design was implemented using digital design flows. Implementation of this clock generator proves that the proposed design methodology is promising for the shrinking time-to-market of future electronics.

Chapter 6
Conclusions

6.1. Thesis Summary & Conclusions

ADPLLs have become the preferred way of clock generation, frequency conditioning, synchronization, and LO generation in a wide variety of applications from ultra-low power microcontrollers to complex SoCs. This has been a result of unprecedented innovation in ADPLL architectures, and advances in the process technology. However, the ADPLLs are still manually designed by highly skilled analog designers, and then laid out by expensive mask designers. This results in increased non-recurring engineering cost, and the custom designed blocks often occupy large silicon area – resulting in higher per unit manufacturing cost as well. In addition, with the on-set of the Internet of Things, time-to-market, cost and form factor of electronics in general must be reduced. Therefore, utilizing digital CAD tools to design and implement traditionally analog functions is an excellent candidate for taking that next step towards electronics ubiquity.

This thesis proposes a design methodology that uses existing digital design flows to implement traditionally analog blocks such as all-digital phase locked loops. This design methodology significantly reduces the design time, and results in smaller silicon area. In addition to the design methodology, novel ADPLL architectures are presented which are amenable to automatic implementation in digital design flows.

One of the main challenges with all-digital ADPLLs is that the performance does not match with those of all-analog PLLs due to the quantization effects. A number of architectural solutions that involve analog design of digital-to-analog converters, and voltage controlled oscillators with complex sigma-delta modulators have already been proposed in research. However, those solutions aren't amenable to automatic implementations with digital design tools. This thesis presents a novel technique for DCO quantization noise reduction. This technique linearizes the power-resolution tradeoff, and does not require any additional custom design.

In conclusion, the design time, cost, power, and area must all be considered when designing electronic systems for Internet of Things applications. This thesis presents a design methodology as well as two prototypes to demonstrate the value of designing traditionally analog blocks using the proposed methodology. The performance of the two PLLs is competitive with prior work. The first prototype is a 400-460 MHz ADPLL, and the second prototype is an ultra-low power clock generator that consumes sub microwatt power, and occupies minimal area.

6.2. Future Work

One of the major advantages of utilizing automatic design tools to reduce design efforts that it frees up the designers and creates room for innovation at an architectural level. The author proposes applications of this design methodology to blocks other than ADPLLs. Some of the work has already begun at the University of Michigan - some of which is discussed below.

1. Elnaz Ansari, a doctoral student at the University of Michigan, has been exploring the applications of this design methodology to other traditionally analog blocks.

She has already implemented a 2GS/s 12b digital-to-analog converter using this methodology. This DAC has already been tested for functionality and performance.

2. Typically, a large portion ICs is digital design with some supporting analog circuitry. For example, a microprocessor can be completed implemented using automated tools. However, the power management and the clock generation are still custom design portions of microprocessors. The analog blocks in such applications often account for 20-40% of the design effort, but they only play an auxiliary role. Therefore, the author suggests that synthesizable architectures for power management blocks should be explore in order to take digital ICs to the next level.

REFERENCES

[1] “The News of Radio”, *New York Times*, July 1, 1948, p. 46, col. 3.

[2] “2012 Update Overview”, *International Technology Roadmap For Semiconductors*, 2013, p. 1-5.

[3] W. Arden, *et al “*More-than-Moore”, *International Technology Roadmap For Semiconductors, p. 8-9.*

[4] B. Murmann, “Digitally Assisted Analog Circuits”, 2003

[5] “What is ‘More than Moore’?”, Mixed-Signal Foundry Experts, www.more-than-moore.com, retrieved: Nov 01,2013.

[6] G. Bell, “Bell’s Law for the birth and death of computer classes,” Commun. ACM, vol. 51, no. 1, pp. 86–94, Jan. 2008.

[7] Yoonmyung Lee, *et al,* "A Modular 1mm^3 Die-Stacked Sensing Platform with Low Power I2C Inter-die Communication and Multi-Modal Energy Harvesting", IEEE Journal of Solid-State Circuits, Vol. 48, No. 1, Jan. 2013.

[8] “An Introduction to the Internet of Things (IoT)”, Lopez Research, Nov 2013

[9] “Rise of the Embedded Internet”, Intel Embedded Processors, Intel Corporation, 2010.

[10] Dave Evans, “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything”, Cisco Internet Business Solutions Group, April 2011.

[11] Janusz Bryze, “Emergence of a $Trillion MEMS Sensor Market”, SensorCon 2012, California

[12] K. Karimi, G. Atkinson, “What the Internet of Things (IoT) Needs to Become a Reality”, June 1013

[13] Amr Fahim, “Clock Generators for SoC Processors: Circuits and Arhictectures”, Kluwer Academic Publisher, 2005

[14] Vaucher, C., “Architectures for RF Frequency Synthesis”, Kluwer Academic Purblishers, 2002

[15] Terng-Yin Hsu; Bai-Jue Shieh; Chen-Yi Lee, "An all-digital phase-locked loop (ADPLL)-based clock recovery circuit," *Solid-State Circuits, IEEE Journal of* , vol.34, no.8, pp.1063,1073, Aug 1999

[16] Jen-Shiun Chiang; Kuang-Yuan Chen, "A 3.3 V all digital phase-locked loop with small DCO hardware and fast phase lock," *Circuits and Systems, 1998. ISCAS '98.* May-3 Jun 1998

[17] Andrea Lacaita, *et al , "*Integrated Frequency Synthesizers for Wireless Systems", Cambridge University Press, 2007.

[18] Y. Park, "A Cell-Based Design Methodology for Synthesizable RF/Analog Circuits", Doctoral Thesis, University of Michigan, 2011

[19] Pandey, J.; Yu-Te Liao; Lingley, A.; Parviz, B.; Otis, B., "Toward an active contact lens: Integration of a wireless power harvesting IC," *Biomedical Circuits and Systems Conference, 2009. BioCAS 2009. IEEE* , vol., no., pp.125,128, 26-28 Nov. 2009

[20] Soltani, N.; Fei Yuan, "A High-Gain Power-Matching Technique for Efficient Radio-Frequency Power Harvest of Passive Wireless Microsystems," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.57, no.10, pp.2685,2695, Oct. 2010

[21] Rutenbar, R.A., "Analog design automation: Where are we? Where are we going?," Custom Integrated Circuits Conference, 1993., Proceedings of the IEEE 1993 , vol., no., pp.13.1.1,13.1.7, 9-12 May 1993

[22] B. Preas and P. Karger, "Automatic Placement: A Review of Current Techniques," Proc. DAC. June 1986

[23] Rutenbar, R.A. , "CAD Techniques to Automate Analog Cell Design", ACM/DAC June 2001

[24] Gajski, D. and Kuhn, R. "New VLSI Tools", Computer, December 1983.

[25] D. Gajski, "Silicon Compilers", Addison-Wesley, 1987

[26] Phelps, R.; Krasnicki, M.J.; Rutenbar, R.A.; Carley, L.R.; Hellums, J.R., "A case study of synthesis for industrial-scale analog IP: redesign of the equalizer/filter frontend for an ADSL CODEC,"

[27] Rutenbar, R.A. " Analog Synthesis (and Verification) Revisited: What's Missing", SMACD, September 2012

[28] Gielen, G. G E; Rutenbar, R.A., "Computer-aided design of analog and mixed-signal integrated circuits," Proceedings of the IEEE , vol.88, no.12, pp.1825,1854, Dec. 2000

[29] Hongzhou Liu; Singhee, A.; Rutenbar, R.A.; Carley, L.R., "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," Design Automation Conference, 2002. Proceedings. 39th , vol., no., pp.437,442, 2002

[30] Bizjak, L.; Da Dalt, N.; Thurner, P.; Nonis, R.; Palestri, P.; Selmi, L., "Comprehensive Behavioral Modeling of Conventional and Dual-Tuning PLLs," Circuits and Systems I: Regular Papers, IEEE Transactions on , vol.55, no.6, pp.1628,1638, July 2008

[31] Phelps, R.; Krasnicki, M.J.; Rutenbar, R.A.; Carley, L.R.; Hellums, J.R., "A case study of synthesis for industrial-scale analog IP: redesign of the equalizer/filter frontend for an ADSL CODEC," Design Automation Conference, 2000. Proceedings 2000 , vol., no., pp.1,6, 2000

[32] Chang, H.; Sangiovanlli-Vincentelli, A.; Balarin, F.; Charbon, E.; Choudhury, U.; Jusuf, G.; Liu, E.; Malavasi, E.; Neff, R.; Gray, P.R., "A Top-down, Constraint-driven Design Methodology For Analog Integrated Circuits," Custom Integrated Circuits Conference, 1992., Proceedings of the IEEE 1992 , vol., no., pp.8.4.1,8.4.6, 3-6 May 1992

[33] M. Faisal, D. D. Wentzloff, "An Automatically Placed-and-Routed ADPLL for the MedRadio Band using PWM to Enhance DCO Resolution," IEEE Radio Frequency Integrated Circuits Symposium (RFIC), June 2013, pp. 115-118

[34] Laber, C.A.; Rahim, C.F.; Dreyer, S.F.; Uehara, G.T.; Kwok, P.T.; Gray, P.R., "Design considerations for a high-performance 3-μm CMOS analog standard-cell library," Solid-State Circuits, IEEE Journal of , vol.22, no.2, pp.181,189, Apr 1987

[35] Harjani, R.; Rutenbar, R.A.; Carley, L.R., "A Prototype Framework for Knowledge-Based Analog Circuit Synthesis," Design Automation, 1987. 24th Conference on , vol., no., pp.42,49, 28-1 June 1987

[36] Harjani, R.; Rutenbar, R.A.; Carley, L.R., "Analog circuit synthesis and exploration in OASYS," Computer Design: VLSI in Computers and Processors, 1988. ICCD '88., Proceedings of the 1988 IEEE International Conference on , vol., no., pp.44,47, 3-5 Oct 1988

[37] Carley, L.R.; Rutenbar, R.A., "How to automate analog IC designs," Spectrum, IEEE , vol.25, no.8, pp.26,30, Aug. 1988

[38] Harjani, R., "OASYS: a framework for analog circuit synthesis," ASIC Seminar and Exhibit, 1989. Proceedings., Second Annual IEEE , vol., no., pp.P13,1/1-4, 25-28 Sep 1989

[39] Fung, A.H.; Chen, D.J.; Li, Y.-N.; Sheu, B.J., "Knowledge-based analog circuit synthesis with flexible architecture," Computer Design: VLSI in Computers and Processors, 1988. ICCD '88., Proceedings of the 1988 IEEE International Conference on , vol., no., pp.48,51, 3-5 Oct 1988

[40] Chowdhury, M.F.; Massara, R.E., "Knowledge-based analogue VLSI layout synthesis," Algorithmic and Knowledge Based CAD for VLSI, IEE Colloquium on , vol., no., pp.11/1,11/6, 6 Nov 1989

[41] Odet-Allah, A.; Hassoun, M., "An algorithm for symbolic and numeric architecture determination in a knowledge-based analog-to-digital converter synthesis environment using fuzzy membership functions," Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on , vol.5, no., pp.607,611 vol.5, 1999

[42] Iskander, R.; Galayko, D.; Louerat, M.; Kaiser, A., "Knowledge-aware synthesis using hierarchical graph-based sizing and biasing," Circuits and Systems, 2007. NEWCAS 2007. IEEE Northeast Workshop on , vol., no., pp.984,987, 5-8 Aug. 2007

[43] Mishra, B. K.; Save, S., "Novel CAD Design Methodology for Two Stage Opamp with Noise-Power Balance," Signal Acquisition and Processing, 2010. ICSAP '10. International Conference on , vol., no., pp.287,290, 9-10 Feb. 2010

[44] Zhiming Pan; Hongyong Li, "Circuit design for a neuron based on FPAA," Electronics, Communications and Control (ICECC), 2011 International Conference on , vol., no., pp.2584,2587, 9-11 Sept. 2011

[45] Fernandez, D.; Martinez-Alvarado, L.; Madrenas, J., "A Translinear, Log-Domain FPAA on Standard CMOS Technology," Solid-State Circuits, IEEE Journal of , vol.47, no.2, pp.490,503, Feb. 2012

[46] Baskaya, I.F.; Reddy, S.; Sung-Kyu Lim; Anderson, D., "Hierarchical placement for large-scale FPAA," Field Programmable Logic and Applications, 2005. International Conference on , vol., no., pp.421,426, 24-26 Aug. 2005

[47] Wen-Hui Fu; Jun Jiang; Xi Qin; Ting Yi; Zhi-Liang Hong, "A Reconfigurable Analog Processor Based on FPAA with Coarse-Grained, Heterogeneous Configurable Analog Blocks," Field Programmable Logic and Applications (FPL), 2010 International Conference on , vol., no., pp.211,216, Aug. 31 2010-Sept. 2 2010

[48] Martinez-Alvarado, L.; Madrenas, J.; Fernandez, D., "Translinear signal processing circuits in standard CMOS FPAA," Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on , vol., no., pp.715,718, 13-16 Dec. 2009

[49] Schlottmann, C.R.; Hasler, P.E., "A Highly Dense, Low Power, Programmable Analog Vector-Matrix Multiplier: The FPAA Implementation," Emerging and Selected Topics in Circuits and Systems, IEEE Journal on , vol.1, no.3, pp.403,411, Sept. 2011

[50] Y. Park, D.D. Wentzloff, "All-digital synthesizable UWB transmitter architectures," IEEE International Conference on Ultra-Wideband (ICUWB), Sep. 2008, pp. 29-32

[51] Y. Park, D. D. Wentzloff, "IR-UWB Transmitters Synthesized from Standard Digital Library Components," IEEE International Symposium on Circuits and Systems (ISCAS), June 2010, pp. 3296-329

[52] Y. Park, D. D. Wentzloff, "An All-Digital 12pJ/pulse 3.1-6.0GHz IR-UWB Transmitter in 65nm CMOS," IEEE International Conference on Ultra-Wideband, Sep. 2010

[53] Webench Design Center, http://www.ti.com/lsds/ti/analog/webench/ overview.page?DCMP=sva-web-schematic-en&HQS=sva-web-schematic-pr-lp-webenchdesigncenter-en, Texas Instruments, retrieved March 20, 2014.

[54] “Helix: Device Level Placement for Custom IC Design”, http://www.synopsys.com/Tools/Implementation/CustomImplementation/Pages/helix-ds.aspx, Retrieved: April 15, 2014

[55] “Virtuoso Electrically Aware Design (EAD) – A New Approach to Custom/Analog Layout”, http://www.cadence.com/Community/blogs/ii/archive/2013/07/10/virtuoso-electrically-aware-design-ead-a-new-approach-to-custom-analog-layout.aspx?CMP=cn_issue36, Cadence Systems, Retrieved April 22, 2014.

[56] "Industry-leading Productivity for Analog, Mixed Signal and MEMS Layout from Tanner EDA", http://www.tannereda.com/products/l-edit-pro, Tanner EDA, Retrieved April 22, 2014.

[57] "Accelerating IC Design Innovation", http://www.synopsys.com/Tools/Pages/default.aspx, Synopsys, Retrieved March 20, 2014.

[58] Madoglio, P.; Zanuso, M.; Levantino, S.; Samori, C.; Lacaita, A.L., "Quantization Effects in All-Digital Phase-Locked Loops," Circuits and Systems II: Express Briefs, IEEE Transactions on , vol.54, no.12, pp.1120,1124, Dec. 2007

[59] Rabaey, J. et al, "Digital Integrated Circuits: A design perspective", 2nd edition, Prentice Hall, 2003.

[60] Wen-Lin Yang; Wen-Hung Hsieh; Chung-Chih Hung, "A third-order continuous-time sigma-delta modulator for Bluetooth," VLSI Design, Automation and Test, 2009. VLSI-DAT '09. International Symposium on , vol., no., pp.247,250, 28-30 April 2009

[61] Farlow, C., "An Overview of the Medical Device Radiocommunications Service (MedRadio) and Future Telemetry Considerations", BANTA 2011, June 2011

[62] "Medical Device Radiocommunication Service (MedRadio)", Federal Communications Commission, http://www.fcc.gov/encyclopedia/medical-device-radiocommunications-service-medradio, Retrieved 8/09/2012

[63] Chen, M.S.-W., Su, D., Mehta, S., "A calibration-free 800MHz fractional-N digital PLL with embedded TDC," ISSCC Digest of Technical Papers, pp.472-473, Feb. 2010

[64] Wenjing Yin, et al, "A 0.7-to-3.5 GHz 0.6-to-2.8 mW Highly Digital Phase-Locked Loop With Bandwidth Tracking," JSSC, pp.1870-1880, Aug. 2011

[65] Weaver S., Hershberg, B., Un-Ku Moon; , "Digitally synthesized stochastic flash ADC using only standard digital cells," VLSI Circuits, pp.266-267, June 2011

[66] Youngmin Park, Wentzloff, D.D. , "An all-digital PLL synthesized from a digital standard cell library in 65nm CMOS," CICC , Sept. 2011

[67] Sai, A.; Yamaji, T.; Itakura, T., "A 570fsrms integrated-jitter ring-VCO-based 1.21GHz PLL with hybrid loop," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International* , vol., no., pp.98,100, 20-24 Feb. 2011

[68] I-Ting Lee; Kai-Hui Zeng; Shen-Iuan Liu, "A 4.8-GHz Dividerless Subharmonically Injection-Locked All-Digital PLL With a FOM of 252.5 dB," *Circuits and Systems II: Express Briefs, IEEE Transactions on* , vol.60, no.9, pp.547,551, Sept. 2013

[69] Liangge Xu; Lindfors, S.; Stadius, K.; Ryynanen, J., "A 2.4-GHz Low-Power All-Digital Phase-Locked Loop," *Solid-State Circuits, IEEE Journal of* , vol.45, no.8, pp.1513,1521, Aug. 2010

[70] Takinami, K.; Strandberg, R.; Liang, P.C.P.; Le Grand De Mercey, G.; Wong, T.; Hassibi, M., "A Distributed Oscillator Based All-Digital PLL With a 32-Phase Embedded Phase-to-Digital Converter," *Solid-State Circuits, IEEE Journal of* , vol.46, no.11, pp.2650,2660, Nov. 2011

[71] Opteynde, F., "A 40nm CMOS all-digital fractional-N synthesizer without requiring calibration," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International* , vol., no., pp.346,347, 19-23 Feb. 2012

[72] Deyun Cai; Haipeng Fu; Junyan Ren; Wei Li; Ning Li; Hao Yu; Kiat Seng Yeo, "A Dividerless PLL With Low Power and Low Reference Spur by Aperture-Phase Detector and Phase-to-Analog Converter," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.60, no.1, pp.37,50, Jan. 2013

[73] Temporiti, E.; Weltin-Wu, C.; Baldi, D.; Tonietto, R.; Svelto, F., "Insights into wideband fractional All-Digital PLLs for RF applications," *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE* , vol., no., pp.37,44, 13-16 Sept. 2009

[74] I-Ting Lee; Yen-Jen Chen; Shen-Iuan Liu; Chewn-Pu Jou; Fu-Lung Hsueh; Hsieh-Hung Hsieh, "A divider-less sub-harmonically injection-locked PLL with self-adjusted injection timing," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International* , vol., no., pp.414,415, 17-21 Feb. 2013

[75] Wooseok Kim; Jaejin Park; Jihyun Kim; Taeik Kim; Hojin Park; DeogKyoon Jeong, "A 0.032mm^2 3.1mW synthesized pixel clock generator with 30psrms integrated jitter and 10-to-630MHz DCO tuning range," *Solid-State Circuits Conference Digest of*

Technical Papers (ISSCC), 2013 IEEE International , vol., no., pp.250,251, 17-21 Feb. 2013

[76] L. Xiu, *et al*, "A novel all-digital PLL with software adaptive filter," JSSC, March 2004

[77] X. Gao, E. A. M. Klumperink, P. F. J. Geraedts, and B. Nauta, "Jitter analysis and a benchmarking figure-of-merit for phase-locked loops," IEEE Trans. Circuits Syst. II, vol. 56, pp. 117–121, Feb. 2009.

[78] P. Andreani and A. Fard, "More on the ————————————□phase noise performance of CMOS differential-pair LC tank oscillators," IEEE J. Solid-State Circuits, vol. 41, no. 12, pp. 2703–2712, Dec. 2006.

[79] D. Ham and A. Hajimiri, "Concepts and methods in optimization of integrated LC VCOS," IEEE J. Solid-State Circuits, vol. 36, no. 6, pp. 896–909, Jun. 2001.

[80] E. Hegazi, H. Sjoland, and A. Abidi, "A filtering technique to lower LC oscillator phase noise," IEEE J. Solid-State Circuits, vol. 36, no. 12, pp. 1921–1930, Dec. 2001.

[81] Zhang, Y., *et al*, "A Batteryless 19 W MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications," *JSSC* 01/13

[82] Shrivastava, A. *et al*, "A 150nW, 5ppm/^{o} C, 100kHz On-Chip clock source for ultra low power SoCs," *CICC '12*

[83] Hoppner, S.; Haenzsche, S.; Ellguth, G.; Walter, D.; Eisenreich, H.; Schuffny, R., "A Fast-Locking ADPLL With Instantaneous Restart Capability in 28-nm CMOS Technology," Circuits and Systems II: Express Briefs, IEEE Transactions on , vol.60, no.11, pp.741,745, Nov. 2013

[84] Wong, C.-H.; Lee, T.-C., "A 6-GHz Self-Oscillating Spread-Spectrum Clock Generator," Circuits and Systems I: Regular Papers, IEEE Transactions on , vol.60, no.5, pp.1264,1273, May 2013

[85] Bernard, S.; Bol, D.; Valentian, A.; Belleville, M.; Legat, J.-D., "A robust and energy efficient pulse generator for ultra-wide voltage range operations," Quality Electronic Design (ASQED), 2013 5th Asia Symposium on , vol., no., pp.80,84, 26-28 Aug. 2013

[86] Kyungho Ryu; Dong-Hoon Jung; Seong-Ook Jung, "All-digital process-variation-calibrated timing generator for ATE with 1.95-ps resolution and a maximum 1.2-GHz test rate," ESSCIRC (ESSCIRC), 2013 Proceedings of the , vol., no., pp.41,44, 16-20 Sept. 2013

[87] Wei Deng; Musa, A.; Siriburanon, T.; Miyahara, M.; Okada, K.; Matsuzawa, A., "A 0.022mm2 970μW dual-loop injection-locked PLL with −243dB FOM using synthesizable all-digital PVT calibration circuits," Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International , vol., no., pp.248,249, 17-21 Feb. 2013

[88] Iizuka, T.; Miura, S.; Ishizone, Y.; Murakami, Y.; Asada, K., "A true 4-cycle lock reference-less all-digital burst-mode CDR utilizing coarse-fine phase generator with embedded TDC," Custom Integrated Circuits Conference (CICC), 2013 IEEE , vol., no., pp.1,4, 22-25 Sept. 2013

[89] Sung W.H., *et al*, "A frequency accuracy enhanced sub-10μW on-chip clock generator for energy efficient crystal-less wireless biotelemetry applications," *VLSIC '10*

[90] Park, P., *et al*, "An all-digital clock generator using a fractionally injection-locked oscillator in 65nm CMOS," *ISSCC '*12

[91] Mesgarzadeh, *et al*, "A Low-Power Digital DLL-Based Clock Generator in Open-Loop Mode," *JSSC* 07/09

[92] Oporta, H., "An Ultra-low Power Frequency Reference for Timekeeping Applications,", 12/08

[93] Li, Y.W., Ornelas, C., Hyung Seok Kim, Lakdawala, H., Ravi, A., Soumyanath, K., "A reconfigurable distributed all-digital clock generator core with SSC and skew correction in 22nm high-k tri-gate LP CMOS," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International* , vol., no., pp.70,72, 19-23 Feb. 2012

[94] Jin-Han Kim, Young-Ho Kwak, Mooyoung Kim, Soo-Won Kim, Chulwoo Kim, "A 120-MHz to1.8-GHz CMOS DLL-Based Clock Generator for Dynamic Frequency Scaling," *Solid-State Circuits, IEEE Journal of* , vol.41, no.9, pp.2077,2082, Sept. 2006

[95] Mesgarzadeh, B., Alvandpour, A., "A Low-Power Digital DLL-Based Clock Generator in Open-Loop Mode," *Solid-State Circuits, IEEE Journal of* , vol.44, no.7, pp.1907,1913, July 2009