Modeling and Design for Low Power and Variation Tolerance in Integrated Circuits

---

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Divya Akella Kamakshi

December 2017

APPROVAL SHEET

This Dissertation
is submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Author Signature: DAK

This Dissertation has been read and approved by the examining committee:

Advisor: Dr. Benton H. Calhoun

Committee Member: Dr. Mircea R. Stan

Committee Member: Dr. Nicholas Barker

Committee Member: Dr. Brucek Khailany

Committee Member: Dr. John Lach

Committee Member: Dr. John Stankovic

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

December 2017

Abstract

Modern integrated circuits ranging from ultra-low power internet-of-things devices to high-performance processors cater to a wide spectrum of applications and notably aid in revolutionizing human lifestyle. For instance, wearable technology is a ubiquitous internet-of-things application that has made great strides in the healthcare and fitness domain. High-performance chips such as graphics processing units enable an efficient computation platform for both graphics and non-graphics applications. However, along with the capability to support an ever-increasing scope of applications, these integrated circuits are also faced with a multitude of design challenges. In this work, we aim to address two such challenges: power consumption and tolerance to circuit or environmental variations, in two key markets of the semiconductor industry, namely internet-of-things and high-performance computing.

Low-power operation is a crucial requirement in modern integrated circuits design. In self or battery-powered internet-of-things devices that are required to have a long lifetime, ultra-low power circuit design is of utmost importance due to limited battery capacities and efficiency of current energy harvesting technology. Ultra-low power operation increases the prospects of their sustainable or even perpetual operation. In the high-performance domain, advanced technologies and faster clock frequencies have led to an increase in static and dynamic power consumption. Low power consumption is necessary for longer battery lifetimes in portable devices like tablets and laptops. It is also crucial to reduce the thermal hot-spots and the need for heat-sinks in such processors. In this work, we investigate low power circuits and systems especially targeting the self/battery-powered applications.

Tolerance to the effects of process and environmental variations are also crucial to both ultra-low power and high-performance designs. In ultra-low power designs, lowering the supply voltage for sub-threshold operation of circuits is a popular technique to reduce energy consumption. However, a critical obstacle to using this method is the high sensitivity of sub-threshold circuits to process, supply voltage, and temperature variations. These variations affect the circuit delays and thereby limit product yield. Similarly, in high-performance designs, high workload variations can cause power supply noise and temperature gradients in the chips. In this work, we investigate design techniques to achieve immunity or tolerance to such variations, which is key to reliable operation and increased chip yield.

Toward the above high-level goals of lowering power and tolerating variations, we explore design techniques for self-powered, internet-of-things systems-on-chips that provide a flexible platform capable of gathering, processing, and transmitting data. These chips have many tightly integrated components, but we specifically target two crucial needs: reliable and ultra-low power circuits for data and clocking. To achieve this, we design circuits such as a direct memory access, a general purpose input/output interface, and an on-chip clock source. Next, we build a modeling framework to enable variation tolerance analysis of integrated circuits. In the ultra-low power domain, we analyze the potential of a post-silicon hold time closure technique to overcome the impact of variations in digital circuits. In the high-performance domain, we analyze the potential of a fine-grained globally asynchronous locally synchronous scheme to overcome the effects of power supply noise. We also present ultra-low power designs of a temperature sensor and a supply voltage monitor, which are crucial to enabling variation tolerance in ultra-low power chips. Finally, we delve into tool flows and strategies to make the design of variation tolerant digital components feasible.

The ultra-low power and variation tolerant design and modeling methodologies presented in this thesis, aim to push the boundaries of chip design for internet-of-things and high-performance computing. In this thesis, we detail the approaches and contribution toward the above goals.

To Mom, Dad, and Vishnu.

Acknowledgments

I owe my deepest gratitude to my advisor, Prof. Benton H. Calhoun, for his endless support and mentorship during the course of my graduate study. His enthusiasm and quest for excellence will continue to be an inspiration. I sincerely thank the other members of my thesis committee: Prof. Mircea Stan, Prof. Nicholas Barker, Dr. Brucek Khailany, Prof. John Lach, and Prof. John Stankovic, for their encouragement and insightful comments.

It has been a privilege to be a part of Ben-group and watch it grow over the last 5 years. I thank Aatmesh Shrivastava, whose guidance helped shape my earlier projects. I appreciate the support from all my Ben-group colleagues of the present and the past:- Rishika Agarwala, Seyi Ayorinde, Arijit Banerjee, Peter Beshay, Henry Bishop, Jacob Breiholz, Jim Boley, Kyle Craig, Chuhong Duan, Patricia Gonzalez, Sumanth Kamineni, Alicia Klinefelter, Kevin Leach, Shuo Li, Ningxi Liu, Christopher Lukas, Harsh Patel, He Qi, Abhishek Roy, Yousef Shakhsheer, Daniel Truesdell, Dilip Vasudevan, Farah Yahya, and Yanqing Zhang. I also thank Xinfei Guo of HPLP group for his contributions to this work.

I sincerely thank Matt Fojtik and Brucek Khailany of NVIDIA for a rewarding internship. I am grateful to Kaushik Mazumdar for his constant support as a friend and a mentor. I will cherish the warm and cheery friendship with Ritambhara and Avinash, during the many ups and downs in this journey. My heartfelt gratitude goes to my teachers, all friends who helped me make memories at UVA, my adorable friends from far away whom I can always count on, and to Terry Tigner, for always being there.

I am most indebted to my parents and my husband for their unrelenting love and care.

Acronyms

ADC	Analog to Digital Converter
ADPLL	All-digital Phase Locked Loop
AFE	Analog Front End
ASIC	Application Specific Integrated Circuits
BJT	Bipolar Junction Transistor
BSN	Body Sensor Node
BWCM	Bit-Weighted Current Mirror
$\mathbf{C}_{ox}$	Gate Oxide Capacitance
CCO	Current Controlled Oscillator
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CTAT	Complementary To Absolute Temperature
DCO	Digitally Controlled Oscillator
DFS	Dynamic Frequency Scaling
DTMOST	Dynamic Threshold MOS Transistor
DVS	Dynamic Voltage Scaling
ECG	Electro Cardiogram
FDSOI	Fully Depleted Silicon On Insulator
FIFO	First-In First-Out
FSK	Frequency Shift Key

GALS	Globally Asynchronous Locally Synchronous
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
HP	High-Performance
IC	Integrated Circuit
IoT	Internet of Things
$\mathbf{I}_{OFF}$	Off-current
$\mathbf{I}_{ON}$	On-current
L	Channel Length
LCU	Lightweight Control Unit
μ_0	Carrier Mobility
MIM	Metal Insulator Metal
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MSB	Most Significant Bit
MSps	Mega Samples per second
n	Sub-V_T Slope Factor
NVM	Non-volatile Memory
OMSP	OpenMSP430
$\mathbf{OSC}_{cmp}$	Temperature-compensated DCO
$\mathbf{OSC}_{diode}$	Diode-connected-transistor-based DCO
$\mathbf{OSC}_{ucmp}$	Temperature-uncompensated DCO
ϕ_t	Thermal Voltage
PDN	Power Distribution Network
PLL	Phase-locked loops
PMU	Power Management Unit
PTAT	Proportional To Absolute Temperature

PVT	Process, Voltage, Temperature
REF_CLK	Reference clock
RF	Radio Frequency
RMS	Root Mean Square
RX	Receive
SAR	Serial Approximation Register
SiP	System in Package
STA	Static Timing Analysis
SoC	System on Chip
SPI	Serial Peripheral Interface
Sub-V$_T$	Sub-threshold Voltage
SVT	Standard Threshold Voltage
TC	Temperature Coefficient
TX	Transmit
UI	Unit Intervals
ULP	Ultra Low Power
V$_{BE}$	Base to Emitter Voltage
V$_{DD}$	Supply voltage
V$_{DD-VIRTUAL}$	Virtual Power Rail
V$_{DS}$	Drain to Source Voltage
V$_{GS}$	Gate to Source Voltage
VLSI	Very Large Scale Integration
V$_T$	Threshold voltage
W	Channel Width
WSN	Wireless Sensor Node
XTAL	Crystal

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

There has been a tremendous growth in the variety of integrated circuits (ICs) and systems in the last few decades. Very large scale integration (VLSI) makes the design of systems-on-chip (SoCs) and processors a reality, by integrating components of different natures such as digital, analog, mixed-signal, radio-frequency (RF) etc. on the same substrate. Such ICs can be categorized based on many metrics such as power, performance, and the application space that they cater to. They can range anywhere from performance-relaxed, ultra-low power (ULP) chips to high-performance (HP), power-hungry chips. In this thesis, we motivate this work using two high-impact IC segments in the current semiconductor market. The main focus is on ULP and variation tolerant SoCs for internet-of-things (IoT) applications. The other segment includes power supply noise tolerant HP processors for computation-intensive applications.

IoT is a rapidly evolving space in the semiconductor industry. It refers to a network of devices that gather, share, analyze, and utilize information for a common purpose. IoT devices may be of different types ranging from personal electronics to sensors for monitoring environmental, physiological, industrial, structural signals, etc. as shown in Figure 1.1.

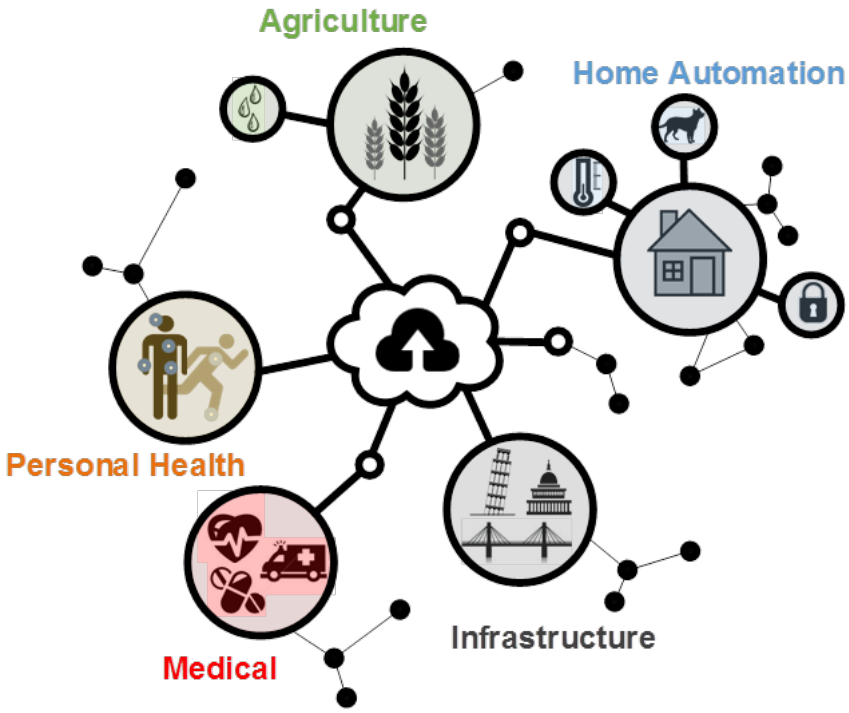


Figure 1.1: IoT node network contains sensors that are embedded into different environments and are required to sustain without human intervention.

IoT devices such as wireless sensor nodes (WSNs) and body sensor networks (BSNs) can gather information from different sources, which is then analyzed and utilized to improve and revolutionize human lifestyle. Many IoT SoCs have previously demonstrated health monitoring [1][2][3][4] and environmental monitoring [5][6]. The vision of a trillion IoT node network calls for embedding such sensors into a number of environments. The requirement in each IoT node regarding form factor, power, energy, reliability, throughput, latency, etc. varies with its application. For a large variety of these IoT nodes such as body sensors and infrastructure monitors, reduced device access is a crucial obstacle. For such IoT devices that are untethered and unobtrusively placed, battery replacement or recharging is a very expensive task. Therefore, battery-less, self-powered technology is increasingly gaining attention for the maintenance-free operation of IoT devices. Therefore, ULP SoC design techniques are crucial to the large-scale deployment of IoT nodes. In this thesis, we focus on some of the design challenges faced by such ULP IoT SoCs such as form-factor, power, and reliability.

Another segment of ICs that we focus on in this thesis is HP processors. HP computing is a rapidly growing market that promises to provide solutions to many complex problems. HP computing involves efficient parallel processing using powerful architectures such as graphics processor unit (GPUs). For instance, GPUs have a massively parallel architecture that consists of multiple small, efficient cores that can carry out multiple tasks simultaneously. It can speed up many applications that are parallel in nature. Therefore, GPUs are increasingly becoming a viable platform for non-graphic general-purpose applications [7], in addition to the traditional graphics applications. An example of one such non-graphic application is molecular dynamics, which is a technique used to track the movement of atoms over different time intervals [8]. This is of great significance in scientific fields such as physics, chemistry, and biology. GPUs also enable a multitude of other applications in the fields of entertainment, finance, medicine, defense, data science, fluid dynamics, artificial intelligence, numerical analysis, machine learning, etc. as shown in Figure 1.2. Therefore, such HP processors promise to make major contributions to scientific innovation, industrial advancements, and the quality of human life. In this thesis, we also focus on the challenges faced by such HP

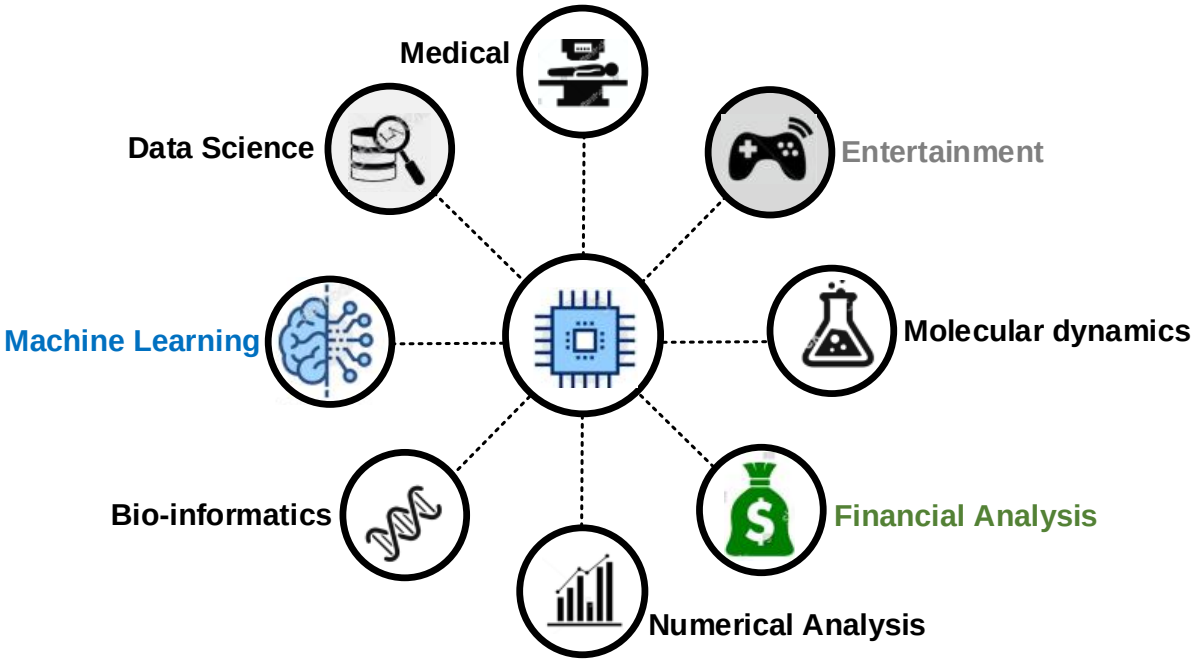


Figure 1.2: HP processors cater to a multitude of applications involving HP computing.

processors such as power and reliability.

In summary, we focus on two high-impact segments of the current semiconductor market, IoT and HP computing, which promise to make a tremendous impact on the quality of human life. Along with its wide range of applications, they also come with design challenges. In the next section, we discuss two crucial challenges, low power consumption and tolerance to circuit and environmental variations, which are motivated by both the IC segments ULP IoT SoCs and HP processors.

1.2 Challenges

Low Power Operation

In most modern IC designs, low power consumption is a crucial requirement for different reasons based on the application. In battery/self-powered IoT SoCs, ULP design techniques are crucial to sustainable operation and long lifetimes. In HP processors, higher power consumption leads to increased heat dissipation and on-chip hot spots. This necessitates higher cooling or heat-sink demands to maintain an optimal operational temperature for its reliable operation. Higher energy efficiency and lower power operation are key to devices such as laptops and tablets to enable longer battery-lives. Optimizations for low power can be done starting at the transistor-level all the way up to the system or the architecture-level. One of the most powerful and popular approaches to decreasing power consumption in SoCs and processors is lowering the supply voltage (V_{DD}). This is because the active power consumption in digital circuits is, to a first order, proportional to V_{DD}^2. This, however, comes at the cost of performance.

In highly performance-relaxed systems that are also required to be ULP [9][10], the circuits are operated in the sub-threshold (sub-V_T) region of transistors, in which V_{DD} is less than the transistor threshold voltage V_T. This has drastically revolutionized the power budgets of ULP systems such as sensor nodes. They are now targeting power consumption in the range

of 100s of nWs to a few μWs, thereby enabling self-powered operation. Recently, in addition to digital circuits, analog circuits are also being operated at lower supply voltages [11][12] for ULP operation.

Low power operation is challenging because it involves more than just operating the system in sub-V_T. An efficient system architecture is needed for additional power savings and energy-efficient operation. There are many tightly-integrated components in ULP IoT chips such as [13][9] that are required to be optimized for power. For instance, the clock source is a critical component in such ULP designs that must run continuously for timekeeping and synchronization and must consume low power. It is important to note that there are other higher voltage domains in an IoT SoC, in addition to the sub-V_T domain. Therefore, it is crucial to look for opportunities in circuit-level optimizations, besides low V_{DD} operation.

In this thesis, we mainly focus on challenges related to achieving low power operation in ULP IoT systems. In addition to low power consumption, IoT devices are also required to have a small form-factor to reduce bulkiness and system cost. Therefore, we find great value in our solutions that especially eliminate off-chip components while still achieving low power. In this thesis, we also discuss a HP processor scheme that enables power savings.

Reliable Operation

Another challenge facing the designers of the above-discussed IC segments, is the high impact of process, voltage, and temperature (PVT) variations. PVT variations and transistor mismatch cause the currents in transistors to vary in both sub-V_T and super-V_T domains. This affects the performance of both analog and digital components, and is prominent in both ULP and HP chips, which are the two segments of focus in this thesis.

Despite being a highly effective approach to reduce energy consumption, sub-V_T operation makes circuits highly prone to PVT variations [14] in ULP chips. In sub-V_T, the on-current to off-current (I_{ON}/I_{OFF}) ratio of a transistor is very low. The drive current of a sub-V_T transistor is exponentially dependent on V_{DD}, V_T, and temperature. Therefore, PVT

variations can lead to further degradation of I_{ON}/I_{OFF}. This leads to a huge spread in gate delays and currents and affects the functionality of sub-V_T analog and digital circuits. The problem of variations is further worsened in scaled technologies. In the current market, a large-scale and real-world deployment of ULP IoT circuits is hindered and made expensive by PVT variations at lower V_{DD}s, and the subsequently low chip yield.

In HP processors, process variations and transistor mismatches are high in some scaled technologies and large chip sizes. Voltage and temperature variations can be caused by workload switching patterns. In this thesis, we exclusively focus on the effects of workload induced voltage variations, which is referred to as power supply noise. A typical power delivery network in HP processors includes many RLC parasitics. Workload current variations in the presence of these parasitics cause V_{DD} to fluctuate. V_{DD} noise is classified into static IR drop and dynamic LdI/dt droop. These power supply noise effects can cause circuit timing errors and therefore, lead to severe performance degradation in HP processors.

In this thesis, we focus on a few challenges in achieving variation tolerant operation by ULP SoCs and HP processors. We present modeling and design techniques toward this goal.

1.3 Summary of Contributions

This thesis addresses two key challenges in the ULP IoT SoC and HP processor IC segments: maintaining ultra-low power and tolerance to PVT variations. Figure 1.3 shows a summary diagram of the design space in focus, and how their challenges and our contributions relate within this space.

Toward maintaining ULP and energy efficiency in self-powered IoT chips, we focus on power optimization opportunities at the system and circuit level. Among the many possibilities of power optimization in a tightly integrated SoC such as [9][10], we focus on circuits related to data and clocking. These supportive components, that are crucial to IoT platform chips, regardless of their application, have not received adequate attention in the ULP context. We

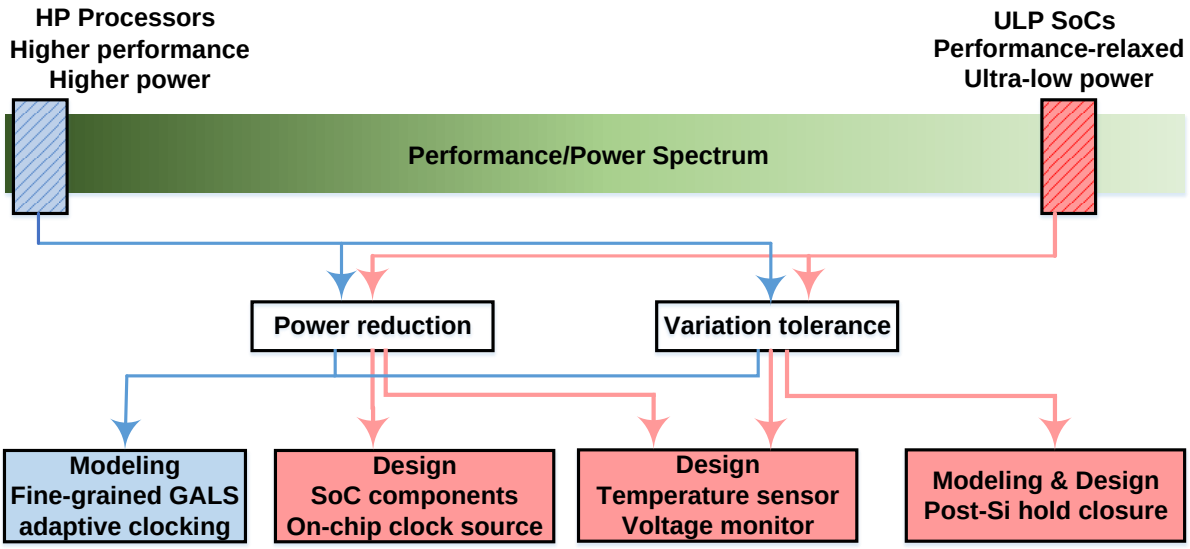


Figure 1.3: Summary of domains of interest, challenges, and contributions in this thesis.

implement a sub-V_T, 2-channel, direct memory access (DMA) module that achieves efficient intra-chip data-transfer on an ULP IoT chip. An IoT chip is required to communicate with other nodes and components. Therefore, we also design and implement a general-purpose input/output (GPIO) interface on an ULP IoT chip. It has effective power-on-reset (POR) circuits that avoid unnecessary power consumption. Finally, clock generators can be a source of a high percentage of ULP system power. Therefore, we design, implement, and test a fully on-chip, flexible clock platform that can trade-off stability for power savings and is highly suitable for BSN IoT devices.

Toward achieving variation tolerance, we propose techniques to mitigate the impact of different sources of variations. We quantify the benefits and costs using modeling techniques. In HP processors, we show using models and analysis that a fine-grained globally asynchronous locally synchronous (GALS) adaptive clocking scheme is beneficial to combat the effects of power supply noise. In ULP performance-relaxed IoT chips, we show that post-silicon timing closure can effectively deal with the increased impact of variations in sub-V_T designs.

We also implement circuits to aid variation tolerance in ULP IoT chips. We design the tunable-buffer and its delay controlling bias generator circuit, which aid in the aforementioned

tunable-buffer post-silicon hold time closure methodology. In this thesis, we present the design and simulations of an ULP temperature sensor. We also design and test a V_{DD} monitor circuit implemented on a test-chip.

Finally, we develop a design strategy and tool-flow for the tunable-buffer based post-silicon hold time closure scheme. The validity of this novel scheme is demonstrated using test-chip results, which show the feasibility of robust and reliable operation of the circuits.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 presents design techniques for ULP IoT SoCs. We identify the needs related to data-flow and clocking in such SoCs. We discuss the design of a DMA module for efficient on-chip data-transfer and POR circuits for a reliable GPIO interface. We discuss the design of a fully on-chip, flexible clock source platform that can generate a stable clock signal with ULP operation and present test-chip measurement results.

Chapter 3 presents modeling and analysis for variation tolerance methodologies in HP and performance-relaxed SoCs. We discuss the impact of voltage variations on HP processors. We model a fine-grained GALS adaptive clocking scheme to effectively tolerate power supply noise. We present the system-level analysis to quantify the benefits of the above scheme. We also the discuss the impact of PVT variations on sub-V_T, performance-relaxed ULP chips. We analyze a data-path tunable-buffer insertion scheme to enable post-silicon hold time correction and estimate the benefits and costs of this scheme.

Chapter 4 presents design techniques of circuits that aid tolerance to PVT variations in sub-V_T, ULP SoCs. We discuss the design of an ULP, sub-V_T temperature sensor design, which includes techniques to resist process-induced current variations and a programmable digital control for resolution-power trade-off in IoT devices with different sampling rate and energy needs. We present the design of an ULP V_{DD} monitor for low-frequency ripple and

voltage variations in ULP IoT SoCs whose area and power can be traded-off for different V_{DD} monitoring needs of SoCs. We also discuss circuit techniques for the variation tolerant post-silicon hold time closure technique introduced in Chapter 3. Toward this, we present the designs of a tunable-buffer circuit and an ULP bias voltage generator to control the delay of the tunable-buffer.

Chapter 5 presents the tunable-buffer insertion strategy for performance-relaxed ULP SoCs. We present a physical design tool-flow for tunable-buffer insertion using standard industry tools. We show simulation and measurement results that verify the concept. The above-mentioned tunable-buffer insertion in data-paths for post-silicon hold time closure involves steps ranging from cost-benefit analysis, the design of building blocks for this scheme, and a tool-flow for automatic design. We split these different topics across Chapter 3 (modeling and analysis), Chapter 4 (design of circuits), and Chapter 5 (methodology and tool-flow) to fit into the flow of this thesis.

Chapter 6 concludes the thesis with insights on future work.

Chapter 2

Design for Ultra-Low Power Internet-of-Things Systems-on-Chip

2.1 Background

[1] In this chapter, we discuss design techniques for ULP IoT SoCs, which provide a flexible platform capable of gathering, processing, and transmitting data. Such SoCs can be used for a broad spectrum of IoT applications such as health monitoring (e.g., blood pressure, electrocardiogram (ECG), etc.), infrastructure and environmental monitoring, home automation, etc. For many applications, such devices are required be deployed in remote or inaccessible locations. It is expensive and time-consuming to periodically replace or recharge batteries in such devices, especially while headed toward the vision of a trillion IoT nodes. For this reason, self-powered operation is becoming an increasingly attractive option. Therefore, these SoCs must operate in an energy-constrained environment where every small amount of stored energy is essential for its sustained and perpetual operation. In this chapter, we discuss the contributions toward this goal with respect to two such ULP and self-powered systems designs [9][10]. These self-powered SoCs are highly integrated designs that include

[1] This chapter derives content from [DAK4][DAK7][DAK8][DAK9]

many components such as memory, data-transfer modules, clocking module, accelerators, inter-chip communication interfaces, power management unit (PMU) etc., to cater to a broad range of applications.

Figure 2.1 shows the main building blocks of such a BSN SoC [9]. In summary, this SoC has two main sensing interfaces: a 4-channel analog front-end (AFE) with an 8-bit analog-to-digital converter (ADC) and variable voltage SPI output pads (0.4 V to 3.3 V) for commercial sensor compatibility. The Opencores MSP430 (OMSP) processor [15] and a multitude of accelerators can execute numerous biomedical and environmental signal processing algorithms (e.g., filtering, peak detection, histograms). A lightweight control unit (LCU) can manage chip data and control while the OMSP is off. A low-power crystal is the clock source for the chip's clocking flexible clocking unit that contains a programmable all digital phase locked loop (ADPLL). The sub-V_T digital blocks run on a 0.5 V supply delivered by the PMU. It is apparent that there are multiple opportunities for ULP techniques in such SoCs.

In this chapter, we discuss ULP design techniques that specifically focus on two crucial needs of these SoCs: data-flow and clocking. The functionality of the digital circuits in an SoC is strongly dependent on its architecture. It is required that the data-flow be efficient and ULP within the SoC (between the peripherals and the memory). For efficient and multi-mode transfer capabilities without involving the LCU, we design a DMA module. Next, it is also important to have commercial sensor compatibility in these nodes, and we design a GPIO interface for such data transfers. Finally, we discuss the design of an ULP, fully on-chip, flexible clock source platform for the clocking needs of BSN SoCs.

2.2 On-Chip Data-Transfer

Within the above-described ULP IoT SoCs as shown in Figure 2.1, blocks of data are needed to be transferred between the different peripherals and/or the memory. In this section, we describe the design of a DMA module, which aids efficient on-chip data transfer without

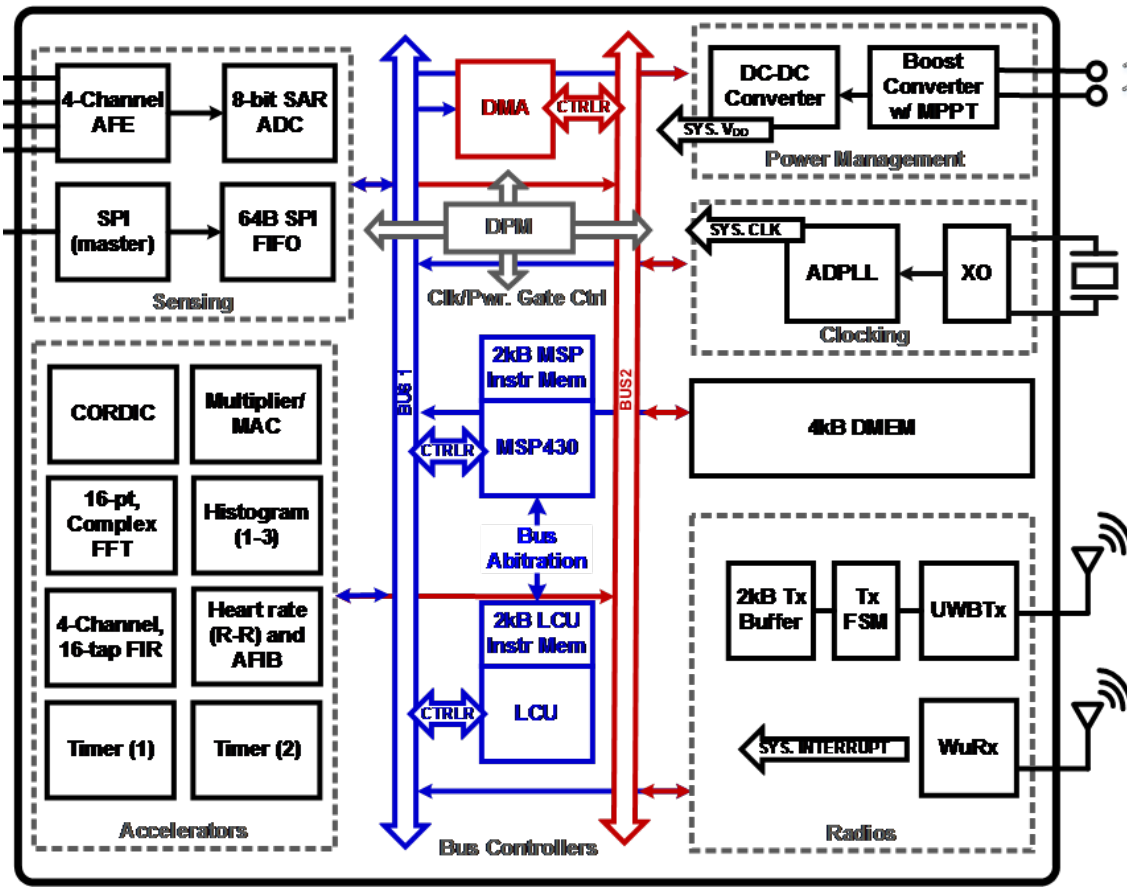


Figure 2.1: System block diagram for an IoT SoC.

constantly engaging the LCU. To achieve this, the chip uses two 16-bit independent buses, each controlled by a separate controller. Bus 1 is controlled by either the OMSP or LCU, while Bus 2 is controlled by a two-channel DMA. All blocks on the chip are memory-mapped peripherals that allow simultaneous access via either bus. At startup, the LCU is the main controller on the chip and can configure the OMSP as the main controller of Bus 1 or as a bus peripheral that is used only for ALU or background operations. Since chunks of data in this SoC needs to be transferred between the peripheral blocks and the on-chip memories, we design the DMA to manage these transfers efficiently on Bus 2 without constantly engaging the Bus 1 controller (LCU or OMSP). This is essentially an ULP architectural solution for a highly efficient data transfer. This also provides an opportunity to the LCU to perform simultaneous operations.

2.2.1 Approach

We discuss the design of the DMA module for a highly efficient and ULP data transfer. The core of the DMA has two channels: Channel0 and Channel1. They are similar channels functionally, but Channel0 always has a higher priority over Channel1.

- When Channel0 is transferring data, it continues to transfer data even if Channel1 is enabled. Once Channel0 is done transferring, control moves to Channel1.
- If Channel0 is enabled while Channel1 is transferring data, Channel1 is put to pending state. Channel0 takes control and finishes its transfer. After that, Channel1 then regains control and continues the pending transfer. Hence, Channel0 has a higher priority over Channel1.

Therefore, both channels can be active at once, but one has higher priority to resolve conflicts.

The DMA module can support four transfer modes for peripheral address:

- Static mode: The address is kept unchanged after every transfer.
- Increment mode: The address is incremented by one after every transfer.
- Decrement mode: The address is decremented by one after every transfer.
- Round robin mode: The address keeps cycling between two, three, or four addresses.

The DMA supports only one mode for memory address:

- Static mode: The memory address is kept unchanged after every transfer.

Therefore, four types of data-transfer are possible:

- Peripheral to Peripheral: It takes 1 cycle to complete every transfer; It takes 1 cycle to simultaneously read from a peripheral address and write into a peripheral address.
- Peripheral to Memory: It takes 1 cycle to complete every transfer; It takes 1 cycle to simultaneously read from a peripheral address and write into a memory address.
- Memory to Peripheral: It takes 2 cycles to complete every transfer; It takes 2 cycles to read from memory.

- Memory to Memory: It takes 2 cycles to complete every transfer; It takes 2 cycles to read from memory.

Therefore, in addition to the ability to resolve conflicts, the DMA module is highly programmable for source and destination addresses, the number of transfers, the transfer type, intervals between each transfer, and the transfer mode. In addition to architectural efficiency, the DMA was designed to operate in sub-V_T to achieve nW power consumption. It operates at 0.5 V V_{DD} with an energy per cycle of ∼3 nJ. In addition to sub-V_T operation, it also includes an independent block reset, clock gating, power gating, and power mode settings for additional power savings.

2.3 GPIO Interface

In the previous section, we discussed the design of a DMA module to enable a highly efficient on-chip data-transfer. Equally important is the ability of an ULP SoC to interact with other sensors or hardware. In this section, we discuss the design of GPIO interfaces, mainly focusing on its POR additions. Figures 2.2 shows the block diagram of another recent self-powered SoC designed as part of a system-in-package (SiP) including a frequency-shift key (FSK) transmitter (TX) and non-volatile memory (NVM) to leverage the advantages of different technologies and also to increase flexibility and reduce cost. Therefore, the SoC needs reliable and ULP sensing interfaces. Therefore, this SoC was built to have two (SPI, GPIO) sensing interfaces. Flexible custom interfaces allow the SoC to communicate efficiently with SiP components and reduce their power. In this section, we discuss the design of a GPIO interface.

2.3.1 Approach

The GPIO interface consists of a bidirectional pad that can be used as both input and output. The GPIO pad consists of a tri-stated input and output buffers with inbuilt level

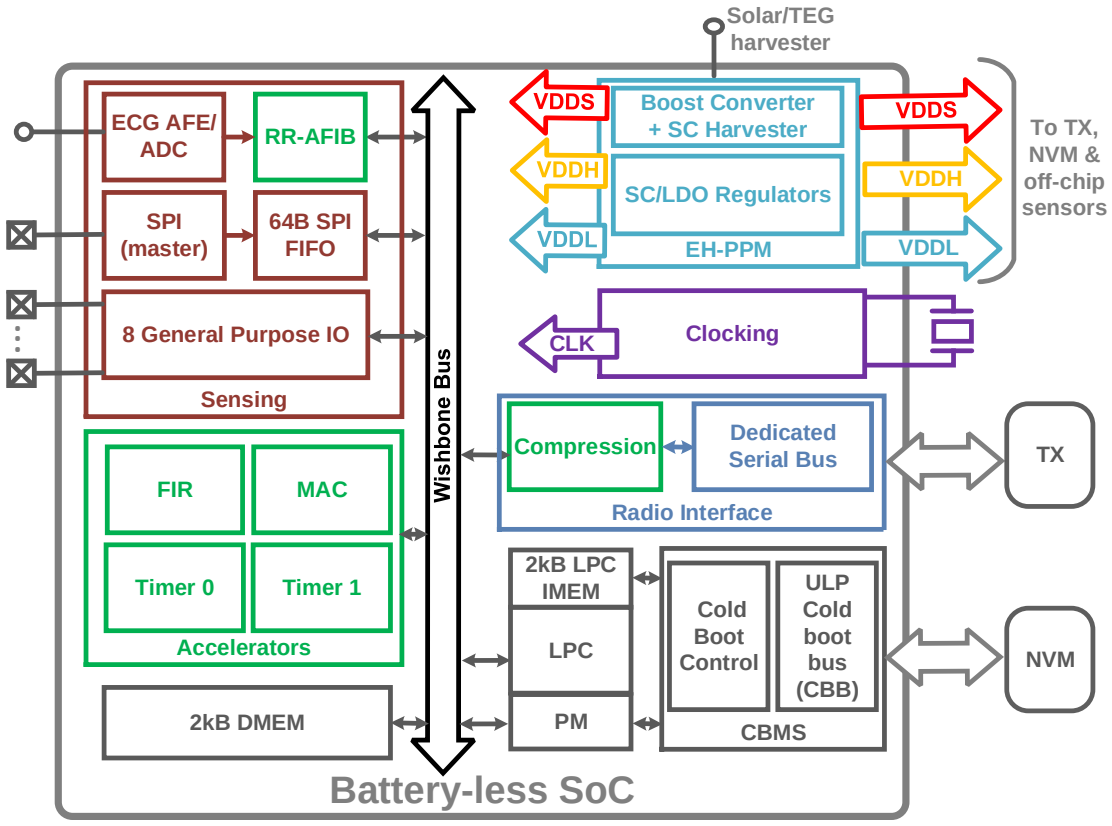
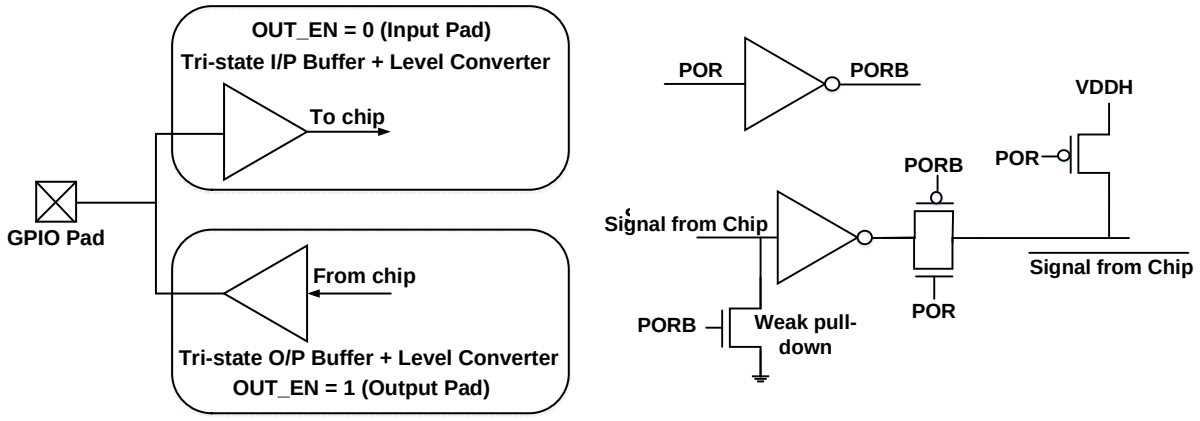


Figure 2.2: Block diagram of IoT SoC designed as part of a SiP.

converters as shown in Figure 2.3a. The GPIO pad is configurable to make it contention free.

There are 2 configuration bits: OUT_EN and PD_EN for each pad coming in from the



(a) (b)

Figure 2.3: (a) GPIO pad consisting of an input and output buffer with level converters. (b) POR circuits for GPIO.

chip in the 0.4 V domain. OUT_EN controls the input/output configuration. When OUT_EN = 0, the GPIO pad acts as an input; when OUT_EN = 1, it acts as an output. In this section, we emphasize on the POR-related circuits to reduce power, which is crucial for the self-powered IoT application space. This capability is configured by PD_EN control bit.

The need for POR configurations arises due to potentially large short-circuit currents and a subsequently unreliable start-up of the SoC. Before the PMU generates a POR signal that indicates stable supply voltage rails, there will be signals coming from the chip to the GPIO interface (data, OUT_EN, PD_EN), which are still in an unknown state (the core 0.4 V domain is one of the later ones to settle in the POR start-up sequence). These floating signals cause high unwanted short-circuit currents in voltage domains associated with the GPIO pad. This, in turn, interferes with the proper start-up of the PMU and the SoC is unable to boot. To solve this issue, we design a POR circuitry that makes the GPIO interface reliable by avoiding the short-circuit currents. Therefore, before POR, the floating inputs to the pad are pulled down/up using a weak pull-up/down structure as shown in Figure 2.3b to mitigate short-circuit currents.

PD_EN is a pull-down enable/disable configuration bit. This signal, along with the data input in the output pad (OUT_EN = 1) configuration comes from the chip in the 0.4 V domain. When POR = 0 (before system POR), the pad will be forced to be configured as an input pad. OUT_EN is weak pulled down to 0, PD_EN is weak pulled up to high, which causes a weak pull-down in input pad (DATA_IN from the chip) as shown in Figure 2.4. When POR = 1, the pad is configured by the chip as the 0.4 V domain becomes stable.

The weak pull-down POR circuits add minimal power overhead. When pull-down is enabled, pull-down current is $\sim$2 nA and when pull-down is disabled it is only $\sim$3 pA.

To summarize this section, we discussed circuits that enable ULP and energy-efficient data-transfer and also the overall SoC operation. These circuits were implemented and tested in the 130 nm bulk complementary metal oxide semiconductor (CMOS) technology [9][10]. In the next section, we focus on another crucial, power-hungry, and high-cost component

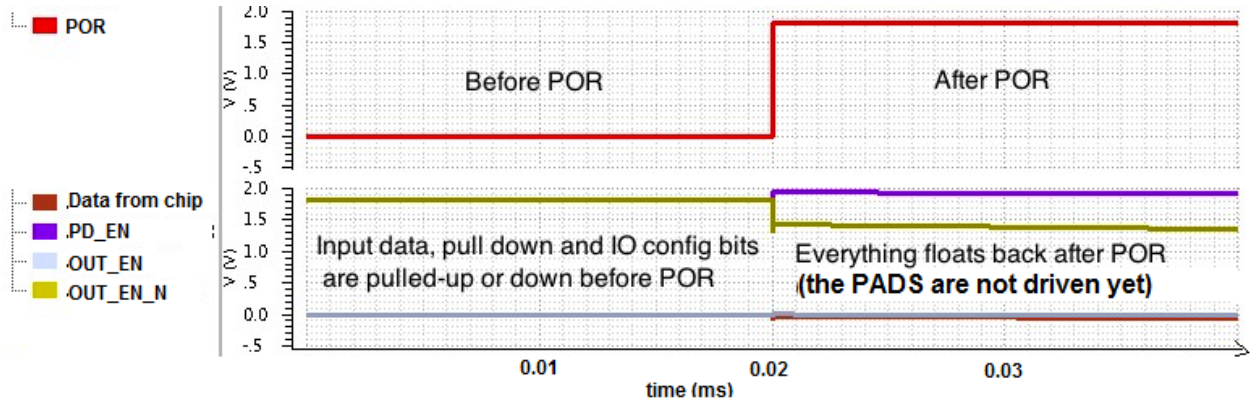


Figure 2.4: Demonstration of reliable POR functionality in GPIO pads.

catering to the SoC clocking requirements: the clock source. We discuss the need for on-chip solutions and our contribution to this challenge.

2.4 On-Chip Clock Source

The clock is a critical component in ULP designs that must run continuously for time-keeping and synchronization. It is essential for such a clock source to consume low power while providing stable frequencies as well as having a small form-factor. Crystal (XTAL) oscillators with high-temperature stability are conventionally used in ULP IoT systems [16]. In a recent implementation [9], the total power consumption of the clock source including an off-chip XTAL, an integrated XTAL oscillator, and an all-digital phase-locked loop was 300 nW at 187.5 kHz. However, a 32.768 kHz XTAL oscillator design in [17] achieves a power consumption of 5.58 nW by lowering the oscillation swing and another oscillator in [18] consumes only 1.5 nW at 0.3 V supply. Although such XTAL designs have achieved low power consumption recently, their biggest disadvantage is form-factor. To achieve the trillion-node IoT vision, it becomes crucial to avoid off-chip components to lower the system volume and cost. Therefore, the design of completely on-chip oscillators, which enable a high-stability clock and ULP consumption in IoT nodes, is necessary. In this chapter, we present a fully on-chip, flexible clock source platform for ULP IoT devices. This clock source

can be programmed and tuned according to the systems power and stability needs. The clock source system targets a lower temperature range that is compatible with BSN applications such as sensor patches that do not experience harsh environmental conditions.

2.4.1 Approach

A fully integrated clock source system is shown in Figure 2.5. The system consists of a high stability temperature-compensated digitally controlled oscillator (DCO) implemented in [19] (OSC_{cmp}), a low-power temperature-uncompensated, diode-connected-transistor-based ULP DCO (OSC_{diode}) that is capable of being frequency locked to OSC_{cmp} and acts as the system clock, and a digital block that can perform locking using a counter-based scheme or a temperature-drift prediction-based mode. For a uniform rate of increasing temperature with time, OSC_{cmp} accumulates error at a slower rate than OSC_{diode}. OSC_{diode} is locked to OSC_{cmp} at a rate that is fast relative to environmentally caused changes in the clock frequency so that its effective long-term stability stays within the stability bound of OSC_{cmp}. The higher-power OSC_{cmp} is disabled between locking events.

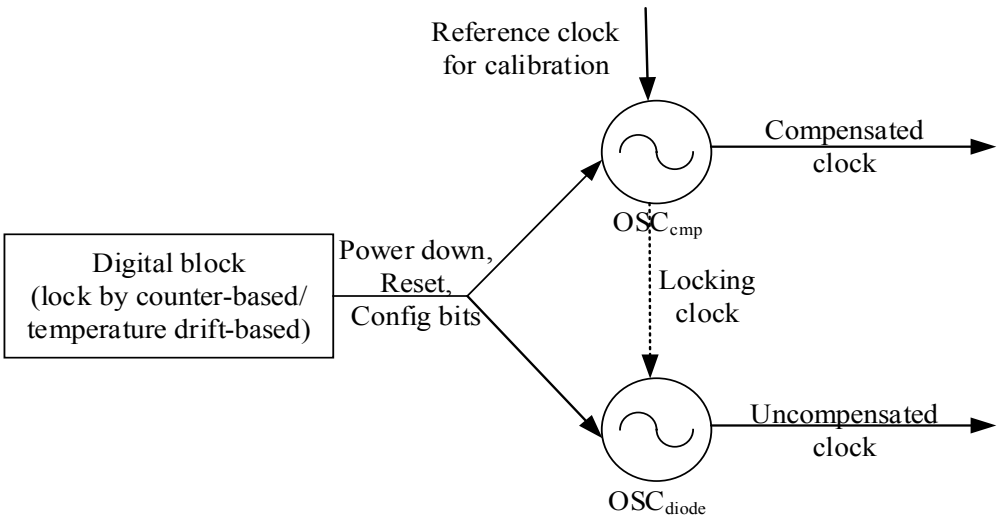


Figure 2.5: Design of an on-chip clock-source platform.

Temperature-Uncompensated Diode-Connected-Transistor Oscillator

In this section, we describe the ULP temperature-uncompensated oscillator for the clock source platform, OSC_{diode}. Diode-connected metal oxide semiconductor field effect transistor (MOSFET) devices are used to generate a virtual power rail ($V_{DD-VIRTUAL}$) from V_{DD}. The oscillator is powered by $V_{DD-VIRTUAL}$ as shown in Figure 2.6. The diode strength is a function of the width of the diode transistor. Diode-connected transistor stacks sized in a binary-weighted fashion are turned on/off by a 23-bit control signal. This controls the value of $V_{DD-VIRTUAL}$ to obtain different frequencies. For a higher 23-bit value, $V_{DD-VIRTUAL}$ increases and hence raises the oscillation frequency. Thus, setting the 23 calibration bits tunes the oscillator to a specific frequency.

The diode-connected transistors operate in sub-V_T. The sub-V_T drain current is given by:

$$I_{DSUB} = I_0 \; exp((V_{GS} - V_T)/n\phi_t) \; (1 - exp(-V_{DS})/\phi_t) \tag{2.4.1}$$

I_0 is the current when $V_{GS} = V_T$, and n is the sub-V_T slope factor. In the diode-connected transistors, $V_{DS} > 3\phi_t$ and Equation (2.4.1) can be approximated as:

$$I_{DSUB} = I_0 \; exp((V_{GS} - V_T)/n\phi_t) \tag{2.4.2}$$

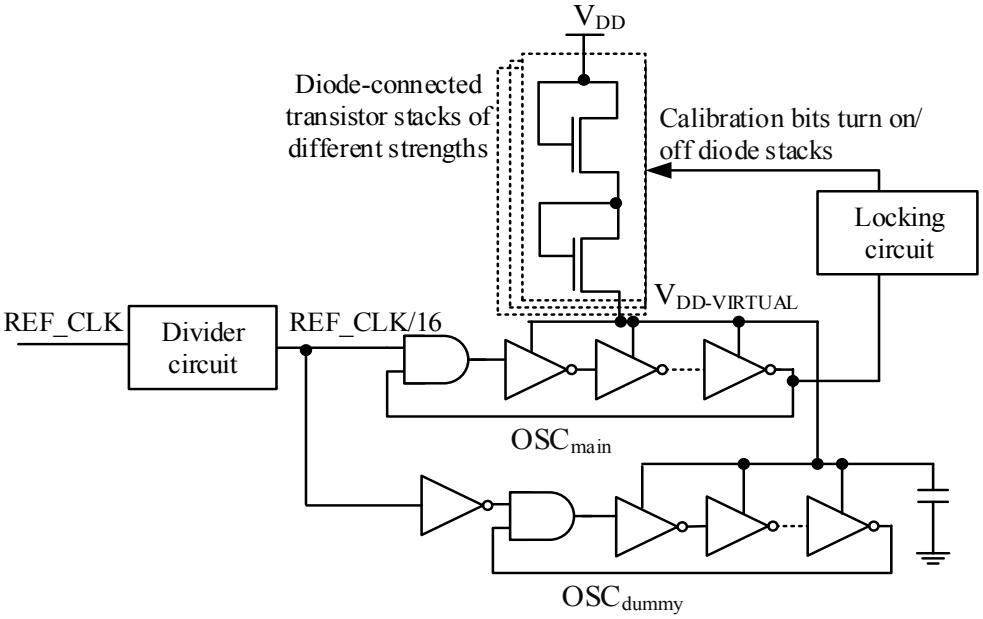


Figure 2.6: Design of a diode-connected transistor-based OSC_{diode}.

This is sub-V_T MOSFET saturation, in which I_{DSUB} becomes independent of V_{DS}.

From the above equations the sub-V_T current temperature coefficient (TC) can be derived as [20]:

$$TC = (1/I_{DSUB})(dI_{DSUB}/dT) = (2-m)/T + (\kappa - (V_{GS} - V_T)/T)/n\phi_t \tag{2.4.3}$$

From 2.4.3, we observe that as V_{GS} is lower (weaker inversion), the TC increases. In diode-connected transistors in OSC_{diode}, $V_{GS} = V_{DS}$ and in the uncompensated oscillator OSC_{ucmp} in [19], $V_{GS} = 0$. Therefore, the TC for our design is lower than OSC_{cmp} [19].

OSC_{diode} uses the DCO architecture presented in [19]. It comprises of an oscillator, a locking circuit, and digital storage for the 23 calibration bits. OSC_{diode} can lock to the frequency of a reference clock (temperature-compensated oscillator OSC_{cmp}). The locking circuit consists of a frequency comparator (to compare the frequency of OSC_{diode} and the reference clock) and a successive approximation register (SAR) logic (to set all the calibration bits). During locking, the instantaneous frequency of OSC_{diode} is affected. Therefore, the re-locking can take place during the idle times of the sensor operation.

Two main techniques are used to stabilize OSC_{diode}. Firstly, we give OSC_{diode} sufficient time to stabilize after every change in the calibration bits (after each bit set and before the next comparison). Secondly, OSC_{diode} includes both a primary oscillator (OSC_{main}) and a dummy oscillator (OSC_{dummy}) as shown in Figure 2.6, with the clock output derived from OSC_{main}. OSC_{dummy} improves the load mismatch on the $V_{DD-VIRTUAL}$ rail. During comparison, OSC_{main} is enabled and consumes a specific amount of current. After comparison, OSC_{main} does not oscillate, and its current consumption reduces, causing $V_{DD-VIRTUAL}$ to increase. This causes OSC_{diode} to finally settle at the wrong frequency. As a remedy, OSC_{dummy} is enabled when OSC_{main} is disabled and vice-versa, which helps to maintain a roughly constant current draw from $V_{DD-VIRTUAL}$ at all times.

Temperature-Compensated Oscillator

OSC_{cmp} is a current-controlled DCO implemented in [19] that is used as a temperature-compensated oscillator in the system. OSC_{cmp} frequency is determined by a constant current source I_o and the capacitance C_L as shown in Figure 2.7. The constant current source I_o is obtained by adding currents from a proportional to absolute temperature (PTAT) source and a complementary to absolute temperature (CTAT) source. In the PTAT source, the current increases with an increase in temperature. In the CTAT, the current decreases with an increase in temperature. The sum current I_o of PTAT and CTAT stays constant and it varies by only 1% over a 100 ^{o}C range across different process corners. C_L is a metal insulator metal (MIM) cap and also has very small temperature variation. The stability of this DCO was measured to be 7 ppm/^{o}C from 20 ^{o}C to 40 ^{o}C at 0.7 V.

Digital Control Block

A low-power digital control block was implemented to automate the locking of OSC_{diode} to OSC_{cmp}. It controls the time interval between successive locks of OSC_{diode} to OSC_{cmp}. We describe two locking modes: (a) a periodic (counter-based) locking scheme; and (b) a prediction (temperature-drift-based) locking scheme in which an algorithm is used to optimize the number of locks in the event of temperature drift.

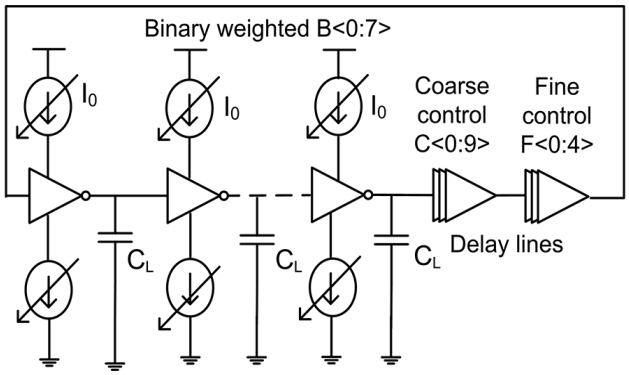


Figure 2.7: Temperature-compensated oscillator OSC_{cmp}.

In the counter-based scheme, locking is achieved through a 32-bit programmable counter. After counting the programmed number of cycles, the digital block issues a signal to enable the locking of one DCO to another. A 32-bit count register implies the capability to count 2^{32} cycles. After locking, a power-down signal is asserted to disable OSC_{cmp} and save power. Its SAR bits are retained to preserve calibration and frequency lock settings. The start-up times of OSC_{cmp} is in the range of a few microseconds, which has to be considered during its power up for the next locking event. The digital block takes this into account through a "settle" register in each counter. A power-up signal is issued at a programmable number of cycles prior to the commencement of the next lock.

In the temperature-drift-based scheme, the temperature dependence of the SAR calibration bits is considered. As temperature drifts, the frequency of OSC_{diode} drifts. When it is re-locked to OSC_{cmp}, the difference in the current and previous value of SAR bits of OSC_{diode} indicates the amount of drift in clock frequency and thereby serves as a proxy for the change in the temperature since the last lock. An algorithm implemented for a temperature drift-prediction based locking scheme is shown in Figure 2.8.

Each state has a corresponding programmable counter threshold, which represents the time interval between successive locks. A minimal threshold (11) is the shortest interval

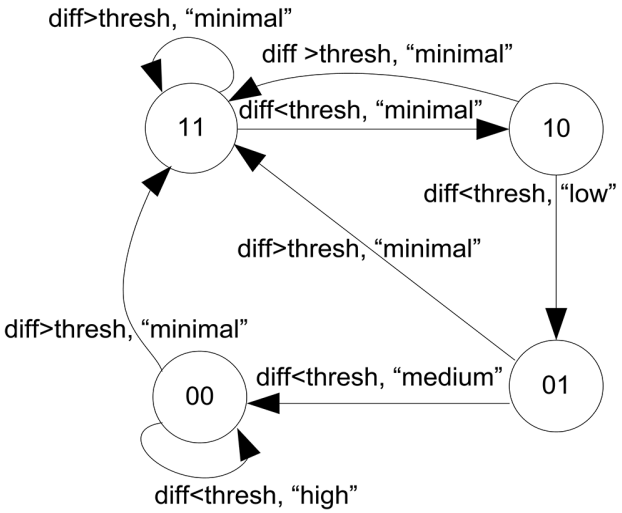


Figure 2.8: A temperature-drift-based locking algorithm.

between successive locks, and a high threshold (00) is the longest locking interval. The SAR bits are monitored every time locking is performed. The initial locking state is "11", because initially there is no information on the temperature drift. In all the successive locks, the difference between current SAR bits and previous SAR bits (diff) determines the locking interval. If diff exceeds a preprogrammed threshold value (thresh), it means that a significant temperature drift occurred. If diff >thresh, a lock is initiated and the state is reset to 11, causing the next locking to take place after a *minimal* interval. If diff <thresh, the locking state is decremented by 1, and the next locking happens after an increased time interval than before. The digital block is operational at voltages ranging from 0.5 V to 1.1 V at 100 kHz frequency.

2.4.2 Results

To calibrate for a specific clock frequency during system start-up, the on-chip DCOs are initially locked to an external stable clock source, such as an XTAL oscillator. The system frequency is programmable, which makes it suitable for dynamic frequency scaling (DFS) technique to save SoC processing power. In this section, we present important metrics of the clock source platform namely power, jitter, and stability.

Power

The average power consumption of the individual components of the proposed system at room temperature (∼27 ^{o}C) is shown in Table 2.1. The total power consumption of the system is 36 nW at 0.7 V V_{DD}, with the digital block operating in the counter-based locking scheme at 1 min locking interval. The prediction mode power is from simulations.

Jitter and Stability

The measured jitter is 0.0023 UI_{rms} (RMS unit intervals) for OSC_{cmp} and 0.0027 UI_{rms} for OSC_{diode}, which is the clock output. It is better than [21] at 0.025 UI_{rms}, and [22] at

Table 2.1: Power consumption of clocks source components at room temperature (∼27 ^{o}C)

Clock Source	V_{DD} (V)	Frequency (kHz)	Power (nW)
OSC_{cmp}	0.7	100	423
OSC_{diode}	0.7	100	20
Digital block (counter mode)	0.7	100	12
Digital block (prediction mode)	0.7	100	5.1

0.024 UI_{rms}, which were designed for sensor systems. Digital blocks operating in the kHz frequency can be operational with the above clock when they are designed to meet the timing constraints. Jitter is also highly crucial in data converter applications. For high-speed data converters operating at mega-samples per second speed (MSps), the clock jitter is required to be in the ps range. But in the ULP application space such as wearable technology, signals such as ECG (having bandwidth of up to 100 Hz) are sampled at the speed of few kilo samples per second (kSps). The data converters are able to operate at acceptable signal-to-noise ratios with RMS jitter numbers in the ns range.

According to our locking scheme, the uncompensated clock (system clock) attains an average long-term stability of the compensated clock. Therefore, the effective average stability of this clock source equals the stability of OSC_{cmp}, which is measured to be 7 ppm/^{o}C in the BSN compatible temperature range of 20 ^{o}C to 40 ^{o}C at 0.7 V. The clock source system targets body sensor applications that experience only limited temperature variation.

The jitter of the oscillators follow a Gaussian distribution with a mean jitter of 0 as a result of deviation from the ideal edge in both positive and negative directions. This jitter is critical to the locking setup, because OSC_{diode} locks to OSC_{cmp} within the accuracy of the jitter on OSC_{cmp}. The number of edges of OSC_{cmp} during a comparison cycle may vary because of the jitter and cause the locked clock (OSC_{diode}) frequency to be slightly higher or lower than the reference. Since the mean of random jitter is 0, the locked frequency OSC_{diode} will on an average match the OSC_{cmp} frequency. A measured plot of the average frequency variation with respect to the reference clock for 100 locks is shown in Figure 2.9. Initially, OSC_{diode} locks to OSC_{cmp} with an accuracy of ∼200 ppm due to the effect of jitter. Due to deviation of jitter in both positive and negative directions, the average frequency variation of

OSC_{diode} with respect to OSC_{cmp} can also be have positive (frequency of $OSC_{diode} > OSC_{cmp}$) and negative values (frequency of $OSC_{diode} < OSC_{cmp}$). The average frequency variation approaches 0 as the number of locks increases, because the jitter averages to 0 with higher number of samples. This achieves a high average long-term stability for clock.

Locking Scheme

To demonstrate the locking scheme, the temperature profile of a thermal chamber enclosing the chip was set to vary by 1 ^{o}C/min. With a 4-min locking interval, the uncompensated oscillators drift away and then re-lock to the compensated oscillator. The counter-based locking scheme is periodic in nature. However, when there is no significant drift in temperature and the oscillator frequency, it unnecessarily consumes extra power during locking events. To address this, the temperature-drift is predicted using an algorithm as shown in Figure 2.8 and the savings are analyzed. Figure 2.10 shows a sample temperature profile over 75 min for temperatures between 20 ^{o}C and 30 ^{o}C.

In the counter-based locking scheme, the locking is performed every minute; hence, the number of locks is 74. With the drift-based algorithm, the number of locks is reduced to 41, because the uncompensated oscillator is not re-locked when there is no temperature drift. This implies ∼1.8x fewer locks than the counter-based scheme. This implies further power savings in an energy-constrained system. The number of locks is optimized, and the savings

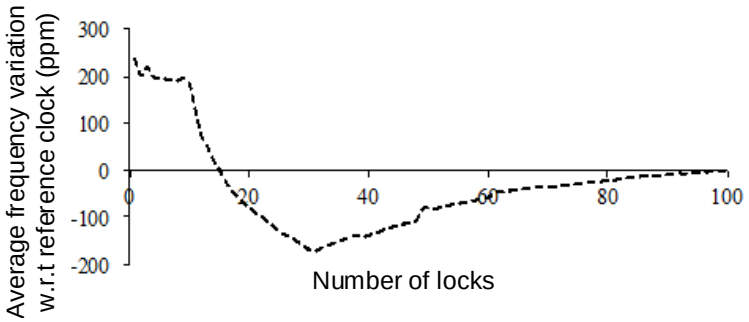


Figure 2.9: Measured average frequency variation w.r.t the reference clock vs. number of locks: Over a number of locking samples the average frequency variation tends to be 0.

heavily depend on the temperature change profile.

Power Supply Noise

The sensitivity of frequency to power supply variation of both OSC_{cmp} (temperature-compensated clock) and OSC_{diode} (clock source system output clock) is 0.1 %/mV. It is comparable to the supply sensitivity in +0.08/0.04 %/mV in [23] (+4%/2% at +/50 mV offset). It is comparable to the voltage stability of [21] (0.5% per 1% V_{DD} at 600 mV that is equivalent to 0.083 %/mV). It is better than 0.42 %/mV in [24], but also necessitates voltage regulation using an ultra-low power voltage reference [25] as described in [24]. In [26], the supply sensitivity is as less as 0.09 %/V due to the regulated local supply. The proposed clock source also similarly requires a very stable supply and a regulated local supply voltage such as in [26] to improve the line sensitivity of the system. Such a system can also be easily powered using an ULP unity gain buffer, providing a very stable low noise supply. This technique is used in [11] to provide a stable band-gap reference voltage.

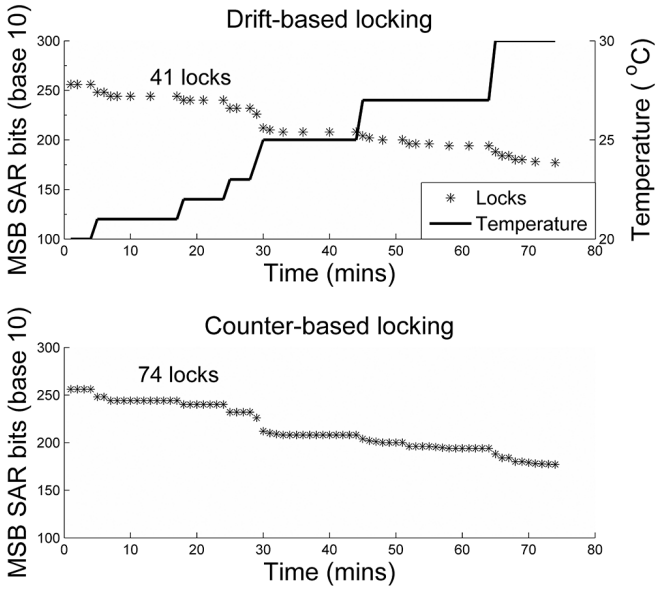


Figure 2.10: The number of locks performed using the counter-based scheme was 74 (1 lock/min) and with the drift-based locking scheme, it was reduced to 41.

Comparison with State-of-the-Art On-chip Clock Sources

Table 2.2 shows the comparison of this work with state-of-the-art, relaxation, ring, RC, and gate-leakage-based on-chip clock sources. This work targets applications in environments that are not harsh, such as the human body.

Table 2.2: Comparison to prior state-of the-art, on-chip clock sources

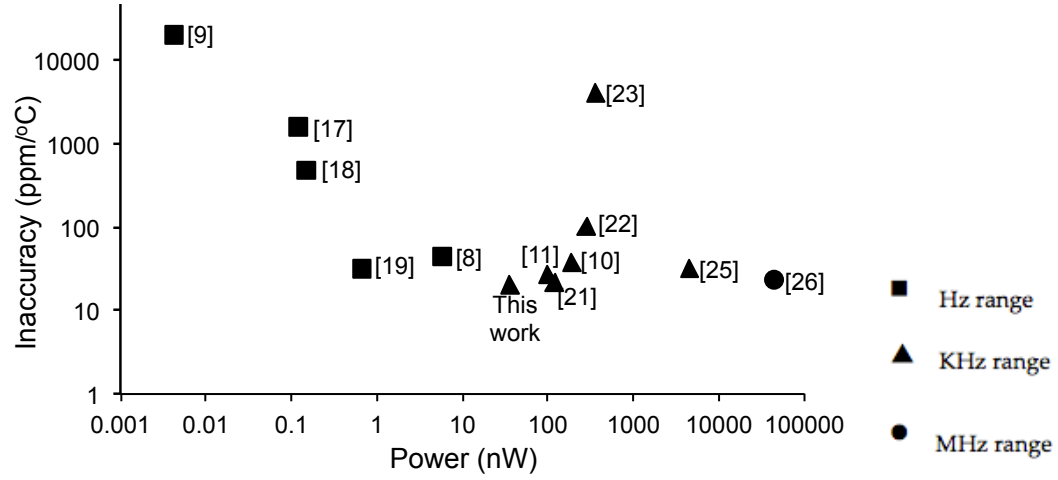
Ref.	Tech. (nm)	Area (mm^2)	Frequency (Hz)	Temp. (^{o}C)	Inaccuracy (ppm/^{o}C)	Line Sensitivity (%/mV)	Power
[27]	180	0.24	11	-10-90	45	0.001@1.2-2.2 V	5.8 nW
[28]	180	NA	18	-30-60	20,000	2.6@0.5-0.6 V	4.2 pW
[26]	65	0.015	33 k	-20-90	38.2	0.00009@1.15-1.45 V	190 nW
[29]	180	0.26	70.4 k	-40-80	27.4	0.0005@1.2-3.2 V	99.4 nW
[30]	130	0.00048	<0.09	0-80	1600	0.04@0.6 V	120 pW
[23]	130	0.019	11.1	0-90	490	0.08/-0.04@0.550.65 V	150 pW
[24]	130	0.015	20	-20-60	31	0.42@0.650.75 V	660 pW
[31]	65	0.032	18.5 k	-40-90	38.5	0.001@1.53.3 V	120 nW
[32]	90	0.12	100 k	-40-90	104	0.0093@0.7250.9 V	280 nW
[33]	180	0.0162	31.25 k	-45-80	4000	0.005@1.8 V	360 nW
[34]	60	0.048	32.8 k	-20-100	32.4	0.000125@ 1.6-3.2 V	4.48 μW
[35]	180	0.04	14 M	-40-125	23	0.0016@1.7-1.9 V	45 μW
This work	130	0.269	12-150 k	20-40	7	0.1@0.65-0.75 V	36 nW @100 kHz

Figure 2.11(a) is a plot of inaccuracy vs. power consumption of the on-chip clock sources and our clock source has the lowest power consumption among the kHz oscillators. Figure 2.11(b) is a plot of inaccuracy vs. energy per clock cycle and our work consumes the least energy per clock cycle among kHz range on-chip oscillators and is comparable to the Hz range timer [28]. Energy per cycle is an important metric that indicates the energy-efficiency of an oscillator and is crucial in battery-operated or self-powered IoT devices. We conclude that our on-chip clock source is highly energy-efficient and is a good candidate for IoT SoCs.

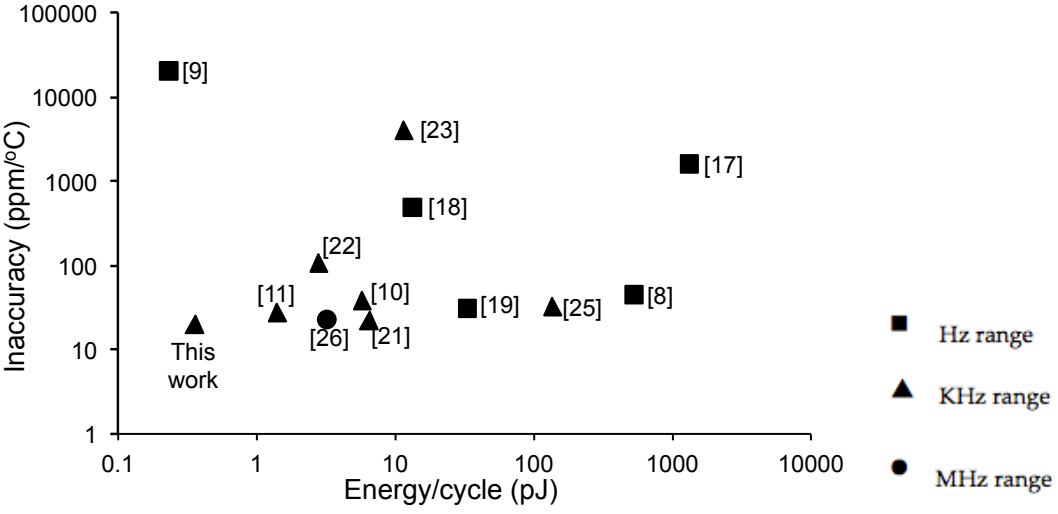
The on-chip clock source system avoids the use of off-chip XTAL, which makes the system cost lower.

2.5 Conclusions

In this chapter, we discuss different design techniques for ULP IoT SoCs. Efficient data-flow and stable clocking are critical for reliable and ULP operation of circuits in such SoCs. Therefore, we discuss circuit techniques specifically targeting data-flow and clock. For efficient on-chip data-transfer between peripherals and memory, we design a flexible, ULP DMA module. The DMA is highly programmable, capable of different modes of transfer and operates in sub-V_T. Sensing interfaces are another crucial requirement in IoT SoCs and



(a)



(b)

Figure 2.11: (a) Plot of inaccuracy vs. power for state-of-the-art on-chip clocks. (b) Plot of inaccuracy vs. energy per cycle for state-of-the-art on-chip clocks.

we present the design of a GPIO interface with emphasis on POR circuits for its reliable operation. The above designs were implemented and tested in IoT SoC platforms.

Finally, we discuss the design of a completely on-chip clock source platform suitable for ULP BSN IoT SoCs. A stable clock signal is required for reliable circuit operation. Although XTAL oscillators are capable of generating a high stability clock at low power consumption, we delve into the design of an on-chip clock source to avoid off-chip components. We emphasize the design of a diode-connected transistor based uncompensated oscillator design. The on-chip clock source is able to achieve the lowest power and energy consumption per cycle compared to prior on-chip kHz frequency oscillators and a long-term average stability of 7 ppm/^{o}C between 20 ^{o}C and 40 ^{o}C at 0.7 V supply voltage. We have two modes of locking: periodic counter based locking scheme as well as a temperature-drift-based scheme for additional power savings. A stability-power trade-off and a wide range of programmable frequencies can be achieved using this system, providing opportunities for further power savings. This design was implemented as a test chip separate from the IoT SoC platforms.

We conclude the discussion of design techniques targeted toward efficient data-flow and clocking for ULP IoT SoCs.

Chapter 3

Modeling for Variation Tolerance in Integrated Circuits

3.1 Background

[1]The impact of PVT variations in an IC varies with its application space. The two IC segments of our focus are HP processors and performance-relaxed ULP SoCs. Whereas power consumption is one of the most important metrics for these ICs, their tolerance to PVT variations is also crucial for reliable operation. In this section, we discuss a few certain variations that drastically affect HP processors and ULP SoCs. In the remainder of this chapter, we introduce and model solutions that tolerate variations and discuss analysis results.

High-performance processors

The power delivery network (PDN) of a typical microprocessor is composed of multiple stages of parasitics originating from the voltage regulator module, motherboard, package, and on-chip power grid as shown in Figure 3.1. Abrupt switching of workload currents in the presence of PDN parasitics introduces spikes and droops in the voltage rails. This is

[1]This chapter derives content from [DAK1][DAK11]

referred to as power supply noise, which is a form of V_{DD} variation. Power supply noise can be classified into IR drop (static and dynamic) and inductive LdI/dt droop. Supply noise fluctuations are caused due to different resonance peaks of the PDN impedance curve as depicted in Figure 3.2(a). The LdI/dt droop is categorized into different types [36] as illustrated in Figure 3.2(b). The third droop has a duration of a few microseconds and is affected by the bulk capacitors, the second droop is caused by the interaction of the board and the package and lasts a few hundred nanoseconds, and the first droop with a duration of a few nanoseconds is determined by the package inductance and on-die capacitance. The first

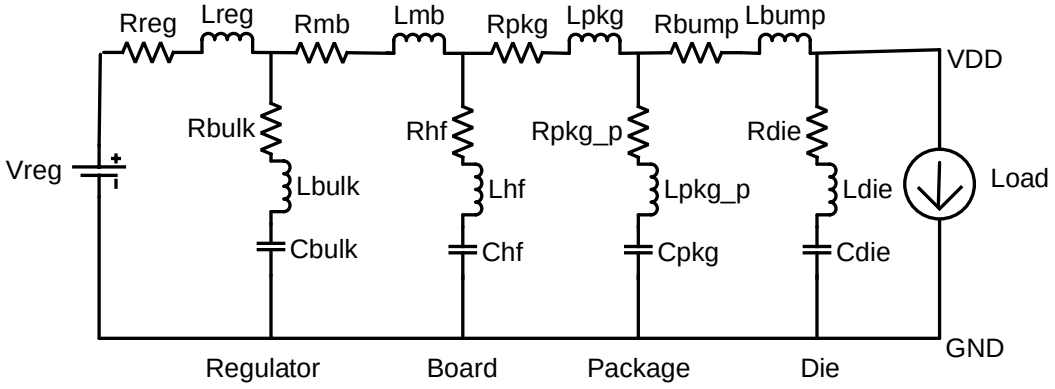


Figure 3.1: A typical PDN model applicable to HP processors.

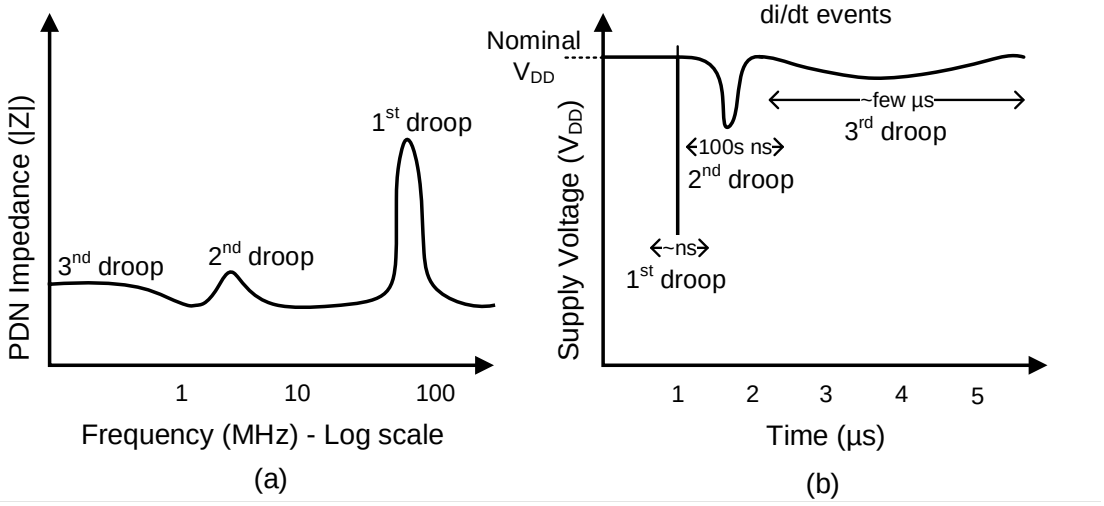


Figure 3.2: (a) Illustration of different resonances of a practical PDN impedance. (b) Illustration of different time constants causing droops of different durations.

droop impedance, and therefore the first droop is typically of a higher magnitude [37]. V_{DD} fluctuations increase with workload currents and therefore tremendously affect the performance of HP processors by causing circuit errors.

Traditionally, to tolerate the worst-case power supply noise, sufficient operating margin or guard-bands are allocated to prevent timing failures. However, this costs additional power and it is also challenging to determine a realistic, worst-case scenario of the power supply noise and the required margins. Therefore, power supply noise needs to be mitigated or compensated for using other techniques. Careful motherboard design and package routing reduce the effect of PDN parasitics to some extent. Usage of decoupling capacitors enables filtering out power supply noise. However, it is typically impractical to place enough capacitance on the SoC to achieve the characteristics of an ideal PDN, due to area constraints. Alternative methodologies involve increasing the operational supply voltage during low processor activity and potential supply droops [38], avoiding voltage emergencies arising from architectural events [39], adaptive and reactive clocking schemes [40][37][41] etc. In this chapter, we consider the baseline to be a traditional adaptive clocking scheme. We introduce the concept of a fine-grained GALS adaptive clocking scheme to thoroughly compensate for the effects of power supply noise. We model and analyze the benefits of a fine-grained GALS adaptive clocking scheme for HP processors. Next, we discuss one of the many negative impacts of PVT variations in the domain of performance-relaxed ULP SoCs.

Performance-Relaxed Ultra-low Power SoCs

Sub-V_T is an attractive option for ULP in performance-relaxed IoT SoCs [9]. However, sub-V_T currents are highly sensitive to PVT variations due to an exponential dependence on PVT parameters as compared to a quadratic dependence in super-V_T. Therefore, PVT variations have a higher impact in the sub-V_T domain as compared to the super-V_T domain. For instance, simulations indicate that the FO4 X1 inverter delay (130 nm CMOS bulk, 27^{o}C) varies by $\sim$16X across slow to fast corners at 0.3 V (sub-V_T) and only by $\sim$2X at 1 V

(super-V_T), as shown in Figure 3.3. Similarly, voltage and temperature variations also have a high impact on delay. Such wide delay distributions in sub-V_T can have a negative impact on circuits of all natures such as analog, digital, etc. In this thesis, we focus on the significant impact of PVT variations on timing closure in sub-V_T digital systems.

Traditionally, to achieve tolerance in the presence of variations, additional timing margins are allocated during design-time. However with an increasing impact of variations and uncertain environmental fluctuations, estimation of realistic timing margins is becoming challenging. The impact of variations becomes even more critical in hold time closure. In a flip-flop-based design, the equation for the hold time constraint is: $t_{hold} \leq t_{clock-q} + t_{logic} - t_{skew}$, where t_{hold} is the flip-flop hold time, $t_{clock-q}$ is the flip-flop clock-to-q delay, t_{logic} is the logic delay, and t_{skew} is the clock skew (clock arrival time at the capture flip-flop, t_{clk2}, minus clock arrival time at the launch flip-flop, t_{clk1}). Hold time violations can be caused by insufficient t_{logic} or excessive t_{skew}. Additional margins are needed to satisfy the above constraint across PVT variations. It directly translates to minimizing t_{skew} during the clock-tree synthesis and buffer-insertion in data paths to increase t_{logic} as shown in Figure 3.4.

This traditional-buffer insertion is relatively easy and tool-friendly, but it requires a realistic yet worst-case estimation of the timing margins across PVT variations. An overestimation of hold time margins leads to an overhead in power and area, whereas an underestimation may lead to circuit failure. It is especially critical because unlike setup-time, hold failures cannot

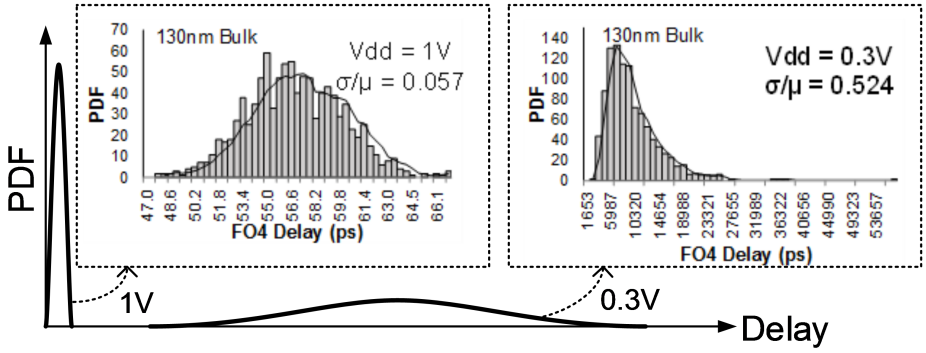


Figure 3.3: Impact of process variations in sub-V_T.

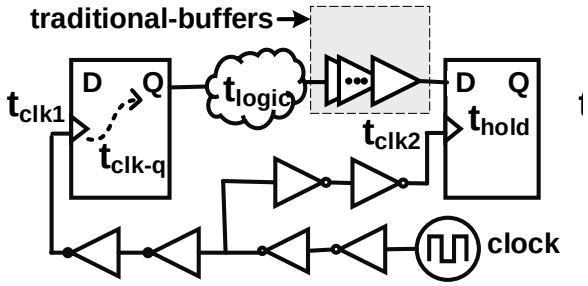


Figure 3.4: Traditional-buffers are inserted in the data-path after timing margin estimation.

be corrected post-silicon in flip-flop based designs. This makes post-silicon hold correction an attractive solution. In this chapter, we consider the baseline to be a traditional-buffer insertion scheme for hold time closure. We introduce the concept of a post-silicon tunable hold time closure scheme using tunable-buffers in hold-critical data-paths, to tolerate the effect of PVT variations in digital components. Compared to traditional-buffering techniques, the tunable-buffer insertion technique seeks to overcome the difficulties of estimating realistic timing margins during design-time. We model and analyze our post-silicon hold time closure technique to estimate its benefits and cost.

In Section 3.2, we analyze a power supply noise scheme for HP processors and in Section 3.3, we analyze a variation tolerant digital timing closure scheme for performance-relaxed ULP SoCs. We conclude the chapter in Section 3.4.

3.2 High-Performance Processors

As introduced in Section 3.1, we focus on the issue of power supply noise in HP processors. Modern HP computing units enable a wide variety of applications for the growing IoT era. For instance, there has been groundbreaking development in the areas of imaging, video and speech. It enables running of many data center applications and the widespread use of machine learning. In this thesis, we focus on high-performance, high-end GPU-like processors that have large areas and high power consumption. Although we present the modeling and analysis in the context of a GPU-like system, it is applicable to other large chips such as multi-

core central processing unit (CPU) server processors, application specific server accelerators such as tensor processing units (TPUs), which have similar power density requirements.

In such HP processors, adaptive clocking schemes have grown to become highly popular solutions that aim to improve performance by adapting to and compensating for supply noise incidents. In [42], the adaptive clocking system contains three PLLs running at independent frequencies with a multiplexer to switch between them using dynamic algorithms. The adaptive clocking design in [43] consists of a delay lock loop-based voltage droop detector and a digital frequency synthesizer that slows the clock as soon as a droop is detected. In the Razor-based 32-bit ARM processor [44], an adaptive controller tunes the operating frequency in response to timing error rates to contribute to the system energy savings. The adaptive clocking scheme in [45] continuously tracks the supply voltage using a tunable replica circuit, which mimics the critical path delay. As illustrated in Figure 3.5, in a fixed clock scheme, the maximum operational frequency (F_{max}) is set to tolerate the worst-case supply noise. However, an adaptive clock tracks supply noise by scaling the operational frequency in response to noise events and potentially attain a higher average system frequency. Adaptive clocking in [43] reduced the operational voltage for a given frequency by 3-6% and thereby, an increase of core power efficiency from 7-19%. An adaptive PLL demonstrated in [46] achieves up to a 15.6% improvement in maximum processor frequency or a 9.8% reduced

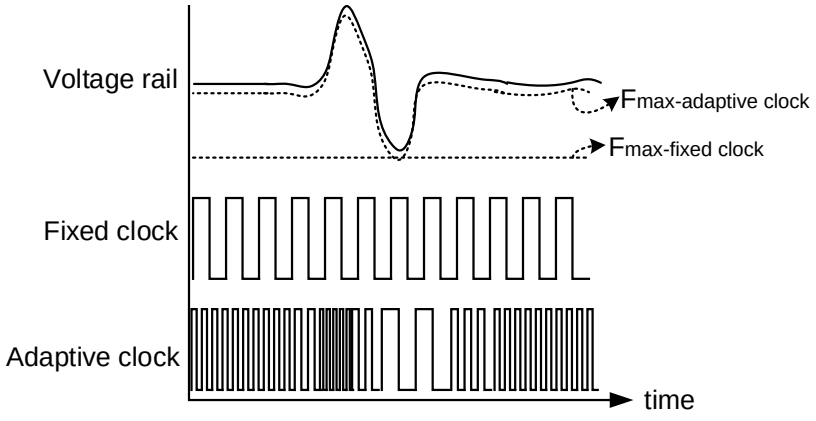


Figure 3.5: In a fixed clocking scheme, F_{max} is set to tolerate worst-case supply noise. In an adaptive clocking scheme, the frequency is varied dynamically with voltage variations.

dynamic power consumption for realistic supply noise. The above adaptive clocking schemes are however limited in their benefits as they cannot compensate for certain local effects of clock-tree insertion delay and spatial voltage variations. In this chapter, we propose to overcome them using adaptive clocking in a fine-grained GALS floorplan.

In the traditional GALS design scheme [47], synchronously clocked islands communicate asynchronously with each other. Recently, commercial designs are adopting GALS methodologies to reduce the design effort [48][49]. Multi-core designs spanning tens of mm^2 of physical area typically have one clock per core. The baseline is such a traditional GALS design with an adaptive clock in each core as shown in Figure 3.6(a). The fine-grained GALS design, on the other hand, is one in which the SoC is partitioned into a large number of synchronous sub-blocks, each spanning only a few mm^2 of physical area with its own local clock as shown in Figure 3.6(b). In this chapter, we discuss the benefits of the fine-grained GALS adaptive clocking scheme to compensate for secondary local effects of power supply noise such as the effects of clock-tree insertion delay and spatial workload variations. However, system partitioning and clock domain crossings may incur a performance penalty. IC testability and generation of local clocks are crucial issues in the adoption of fine-grained GALS. In this thesis, we also present a comprehensive overview of the potential challenges and overheads of the fine-grained GALS adaptive clocking scheme.

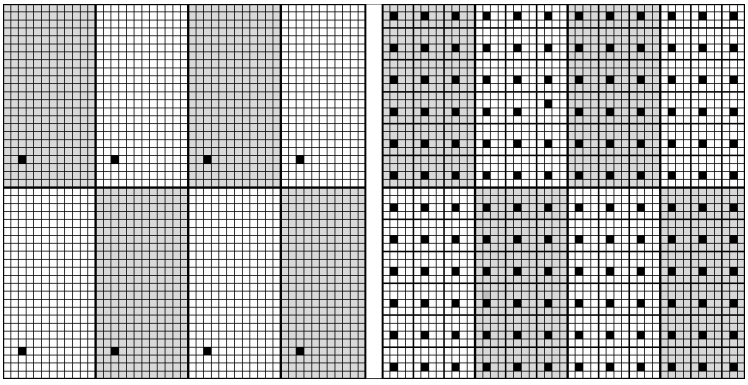


Figure 3.6: (a) Baseline traditional synchronous adaptive clocking scheme (each black dot represents an adaptive clock). (b) Fine-grained GALS adaptive clocking scheme.

3.2.1 Approach

A traditional adaptive clock source scheme can dramatically reduce voltage guard-band or margin and guarantee failure-free operation in the presence of power supply noise. For instance, the operational voltage was reduced by 3-6% with the adaptive clocking scheme in [43]. However, certain local effects of power supply noise cannot be compensated by this scheme. In this section, we describe these local effects of clock-tree insertion delay and spatial workload variations. We also present a modeling methodology to compare the behavior of traditional synchronous adaptive clocking scheme with the proposed fine-grained GALS adaptive clocking scheme in the presence of the above local effects.

Issues in a Traditional Adaptive Clocking scheme

Effect of Clock-tree Insertion Delay: Clock-tree insertion delay is the delay between the clock root at the source to the clock leaf at the flip-flops in load circuits. As illustrated in Figure 3.7, when the physical region in which the adaptive clock generator is present experiences a power supply noise event, it responds with a change in frequency. However, it takes a time equal to the clock-tree insertion delay (Δt), for the stretched clock pulses to reach the load circuits.A clock-tree with a high insertion delay experiences varied voltage fluctuations across its length. It also takes a longer time for the load circuits to see the stretched clock pulses. On the other hand, the load circuits instantaneously sense any fluctuations in the supply voltage. Therefore, there is a time difference (Δt) between the voltage change and the change in the operating frequency as seen by the load circuits. In modern SoCs, this time difference due to the clock-tree insertion delay is in the range of one to a few clock cycles (1-2 ns) [36]. This clock-tree insertion delay effect requires additional margin for failure-free circuit operation, which cannot be fully eliminated with the use of traditional synchronous adaptive clocks.

Effect of Spatial Workload Variations: Spatial variations in the switching current can cause voltage fluctuations of different magnitudes in different sections of the chip. Figure

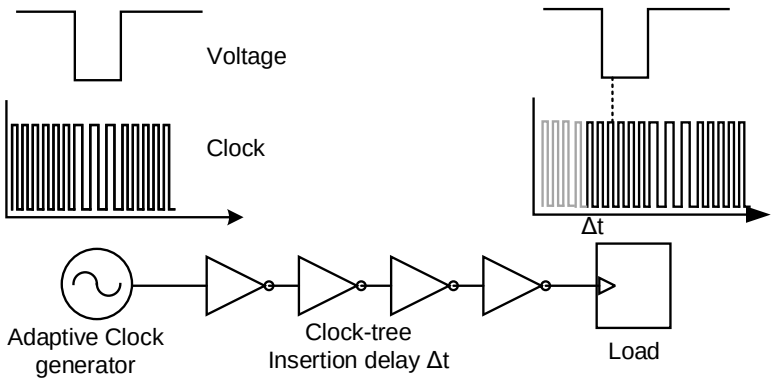


Figure 3.7: Illustration of the effect of insertion delay: The stretched clock is delayed by the clock-tree insertion delay Δt. This requires additional margin for failure-free operation.

3.8 illustrates the effect of varied voltage fluctuations across the chip. For instance, a local supply droop near the load circuits can cause them to slow down, but the adaptive clock source may be sensing a local supply voltage overshoot that increases the clock frequency. Therefore, a faster clock frequency reaches slower circuits. This is an example scenario in which spatial workload variations can cause a timing failure. The traditional synchronous adaptive clocking scheme does not compensate for this localized effect. Additional margins are required to overcome this effect of spatial workload variations and guarantee failure-free operation.

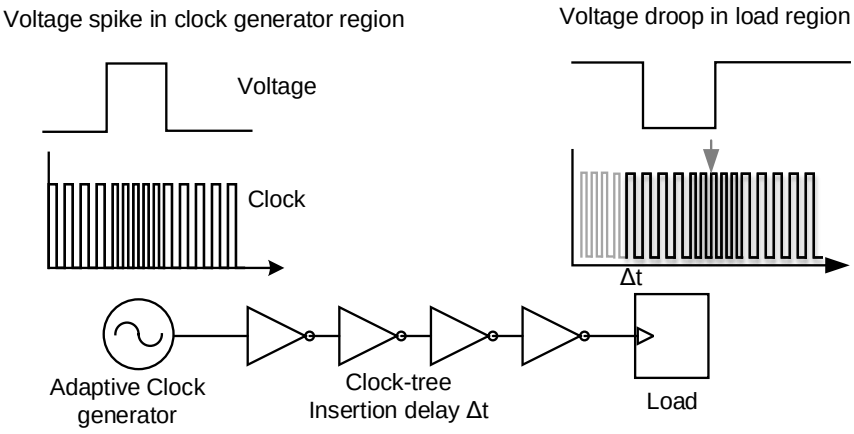


Figure 3.8: Illustration of the effect of spatial voltage variations: The clock source and the load circuits may respond differently to local voltage variations leading to circuit failure.

Benefits of a Fine-grained GALS Adaptive Clocking Scheme

The proposed fine-grained GALS adaptive clocking scheme involves a design that is composed of myriad synchronous islands of areas as small as a mm^2. A fine-grained GALS floorplan mitigates the above-discussed local effects of insertion delay and spatial workload variations as illustrated in Figure 3.9. The insertion delay for such fine-grained synchronous islands is only a few hundred picoseconds as compared to a few nanoseconds in traditional synchronous islands. Moreover, due to the close proximity of the clock generator and load circuits, they also tend to experience similar voltage fluctuations. Therefore, fine-grained GALS adaptive clocks can reduce the additional margins associated with mitigating the local effects of clock-tree insertion delay and spatial workload variations.

Metric: To quantify the savings in margin with fine-grained GALS, we first discuss the metric *uncompensated voltage noise*. The clock frequency at the load circuits is impacted by many factors such as the clock-tree insertion delay and spatial voltage variations at the clock generator compared to the load circuits. V_{req} is the actual voltage that is required for failure-free operation of the load circuits at the system clock frequency. V_{mean} is the average

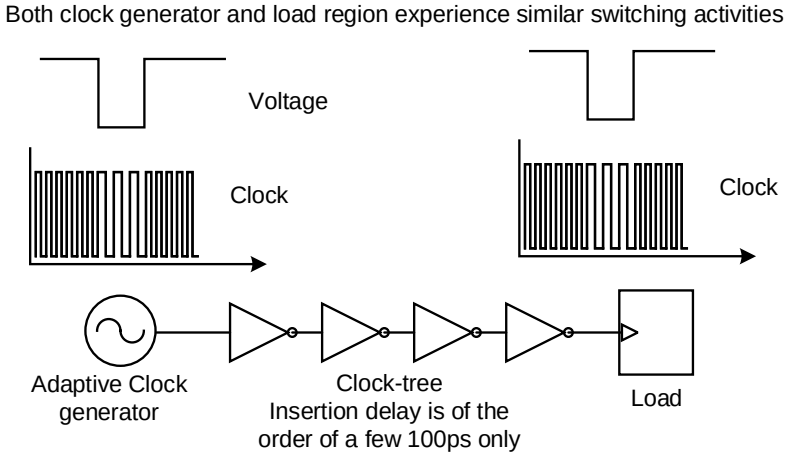


Figure 3.9: Illustration of benefits of the fine-grained GALS adaptive clocks: A lower insertion delay as compared to traditional synchronous GALS scheme reduces the effect of insertion delay and the close proximity of the clock and the logic reduces the effect of spatial workload variations.

voltage present at the load circuits over a clock cycle. When V_{mean} is less than V_{req} the load circuits will incur timing failure. To overcome this, an additional margin is required, which is the difference between V_{mean} and V_{req}. This is referred to as *uncompensated voltage noise*. The system-level experimental setup is shown in Figure 3.10 to quantify the proposed *uncompensated voltage noise* metric for different local effects. The PDN is simulated using a publicly available analysis tool, Voltspot [50]. Verilog-A was used to model the adaptive clocking scheme as well as the clock-tree insertion delay.

Experimental Setup: To demonstrate the spatial effect of workload variations, a distributed PDN model is required. We use Voltspot [50], a trace-level simulator, to model a distributed PDN. For the analysis in this paper, we divide the chip area (23.5 mm x 23.5 mm) into a 47 x 47 array of architectural units of approximately 0.5 mm x 0.5 mm. We modified Voltspot to emulate a distributed package in addition to the on-chip PDN distribution as shown in Figure 3.11. The worst-case supply noise occurs at around 30 MHz (PDN resonance point) workload frequency. To demonstrate the effect of clock-tree insertion delay, a uniform current distribution is assumed across the PDN array. To demonstrate the effect of spatial workload variations, a non-uniform current distribution is assumed across the PDN array.

We model the adaptive clock generator to respond to voltage variations with a single cycle

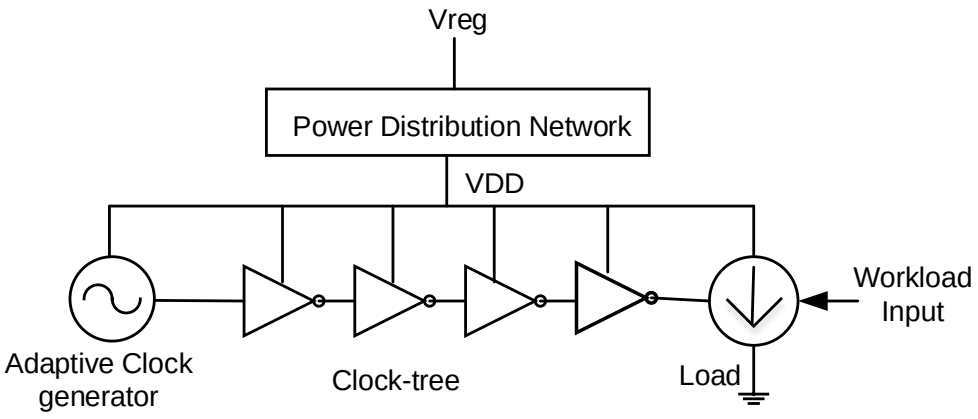


Figure 3.10: A system-level experimental setup.

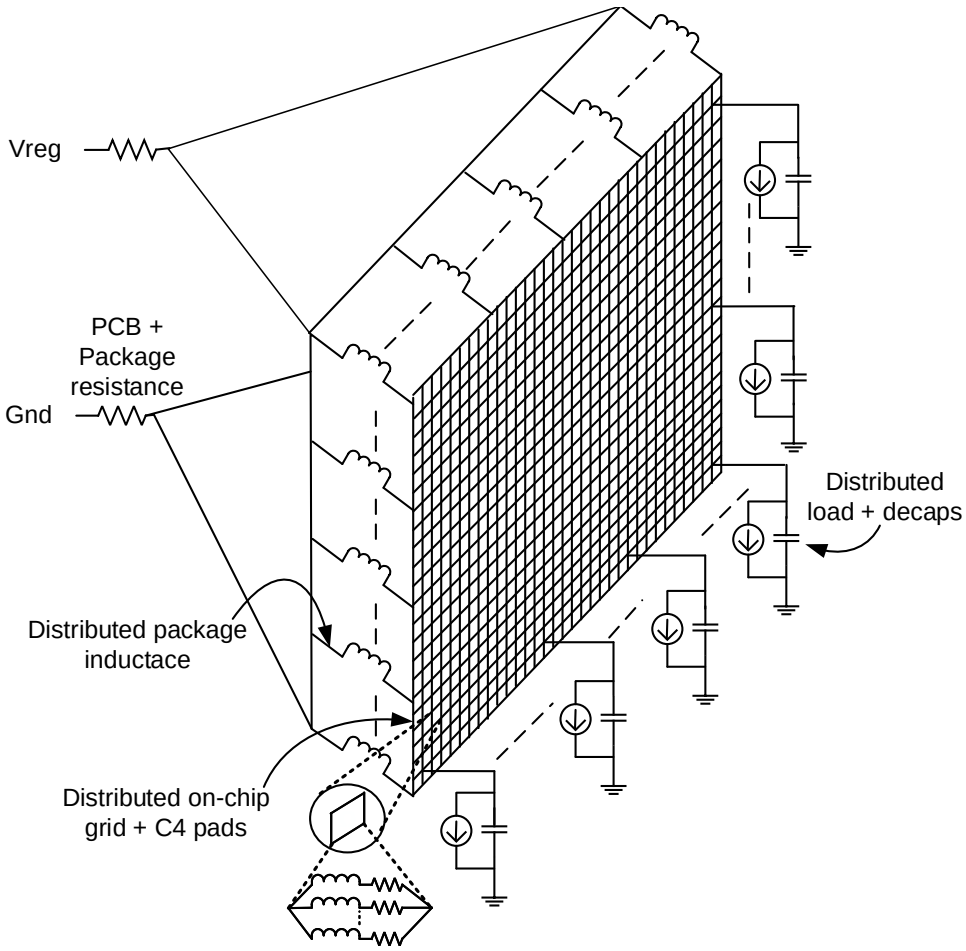


Figure 3.11: Distributed PDN model implemented using Voltspot.

latency. Firstly, a VF curve was obtained by modeling a critical path including low, regular and high-V_T devices using the FreePDK 45 nm kit. The data and clock path models are shown in Figure 3.12. The circuit was simulated for maximum frequency, F_{max}, at different V_{DD} and insertion delays. The global clock path drives long wires and includes both device and wire delays. This is used to generate a VF curve to determine the frequency of the adaptive clock at a given supply voltage. The load rail voltage is averaged over the previous cycle to get V_{mean}. The frequency of the current cycle is determined by V_{mean} and the pre-characterized VF curve.

The clock-tree is also modeled in Verilog-A to include the characteristics of both the global and the local clock distribution. The device and wire parasitics are derived using post-layout

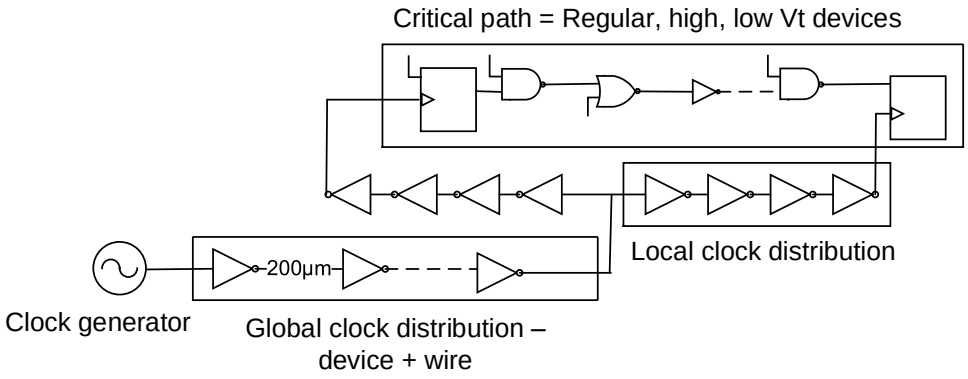


Figure 3.12: Clock-tree and critical data-path modeling to obtain the voltage-frequency (VF) relationship used in the adaptive clock generator model.

HSPICE simulations. It is assumed that the entire clock-tree sees the same supply voltage. The clock-tree insertion delay variation with supply voltage is obtained for different cases of nominal insertion delay. Similar trends and curve-fit polynomials are obtained for nominal insertion delays 1.5 ns, 0.9 ns, 0.6 ns and 0.3 ns. For instance, the polynomial for 1.2 ns insertion delay is: $D(ns) = -22.9V^6 + 130.3V^5 - 301.7V^4 + 361.9V^3 - 232.8V^2 + 73.1V - 6.6$

Switching activity in the workload in the presence of PDN parasitics causes noise in the power supply rail. In [39], the current profile characteristics in SPEC benchmarks were broadly classified into three categories: step, pulse, and resonating currents. Among these, resonating currents (repeating activity patterns), have the worst effect on supply noise. A resonating workload waveform is shown in Figure 3.13. We choose a 10 cycle slew rate for the dynamic current switching from 10 A to 90 A in our analysis and sweep the workload frequency up to 40 MHz. The nominal supply voltage is 1 V.

3.2.2 Results

For the analysis, we divide the PDN area into a 47 x 47 array of architectural units as shown in Figure 3.14, each unit of size 0.5 mm x 0.5 mm. We group the 2209 units into nine partitions: three 12 x 20 and one 11 x 20 partitions along the top and the bottom, and one 47 x 7 partition along the middle of the die. We focus on one such 12 x 20 partition for

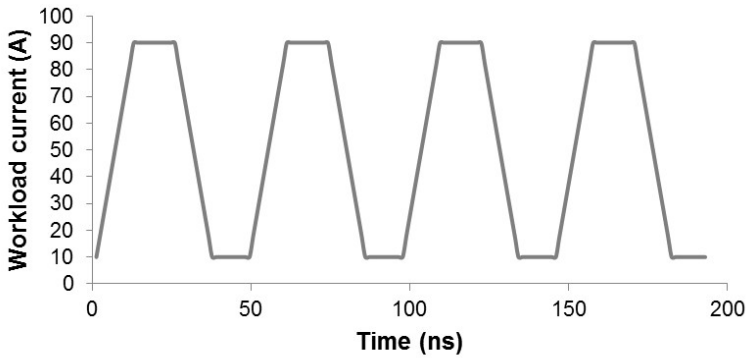


Figure 3.13: Resonating workload associated with repeating activity patterns.

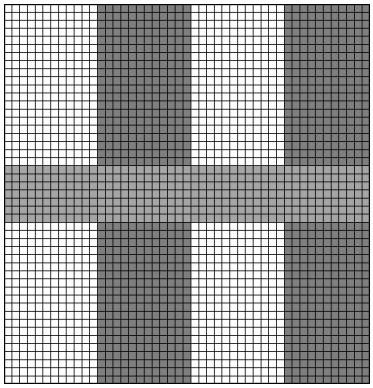


Figure 3.14: PDN area is divided into a 47 x 47 array of units.

our analysis. We present the simulations to show the effect of clock-tree insertion delay and spatial workload variations in the traditional synchronous GALS scheme and their mitigation using the fine-grained GALS adaptive clocking scheme.

Effect of Clock-tree Insertion Delay

To analyze the effect of clock-tree insertion delay (Δt), a uniform resonating workload current as shown in Figure 3.13 is applied throughout the PDN partition. *Uncompensated voltage noise* is measured by sweeping workload frequencies between 10 MHz to 40 MHz for different designs with insertion delays ranging from 0.3 ns to 1.5 ns. As expected, the worst-case *uncompensated voltage noise* occurs at higher insertion delays (1.5 ns in Figure 3.15) at workload frequency near the PDN resonant frequency of 30 MHz.

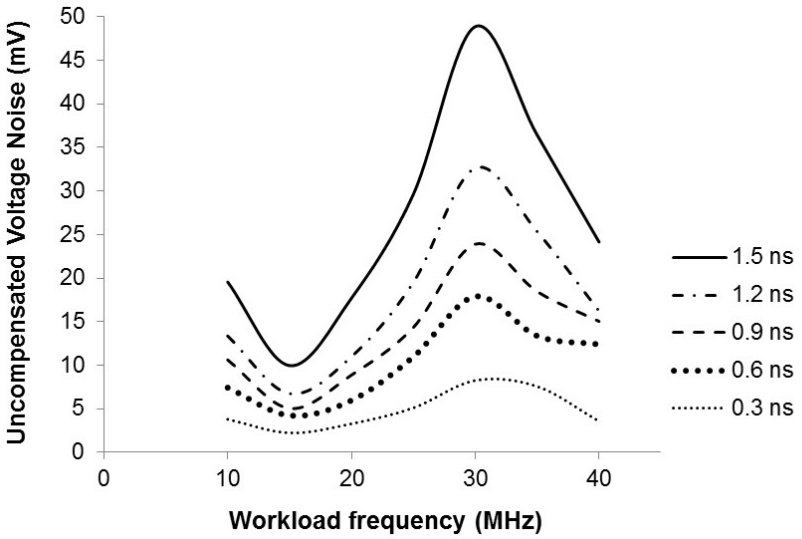


Figure 3.15: *Uncompensated voltage noise* vs. workload frequency for various insertion delays. *Uncompensated voltage noise* increases with insertion-delay increase.

Using the traditional synchronous adaptive clocking scheme with a clock-tree insertion delay of 1.5 ns, the *uncompensated voltage noise* at 30 MHz workload frequency is 49 mV. On the other hand, the fine-grained GALS adaptive clock scheme with a low insertion delay such as 300 ps demonstrates an even lower *uncompensated voltage noise* of 8 mV. Therefore, fine-grained GALS adaptive clock mitigates the effect of clock-tree insertion delay and further reduces the voltage margin.

Effect of Spatial Workload Variations

To analyze the effect of spatial voltage variations in addition to the effect of insertion delays, we apply a non-uniform workload across a PDN partition as illustrated in Figure 3.16. Figure 3.16(a) represents the baseline traditional GALS adaptive clocking scheme. It is equivalent to a large synchronous island/core operated using its own local clock. The clock generator (clock unit) is assumed to be in the top right corner. The *uncompensated voltage noise* is measured in the lower-left corner (measurement unit) that is farthest from the clock unit for worst-case measurements. Figure 3.16(b) represents the fine-grained GALS adaptive clocking scheme. We assume a 2 mm x 2 mm size for each GALS unit with its own local

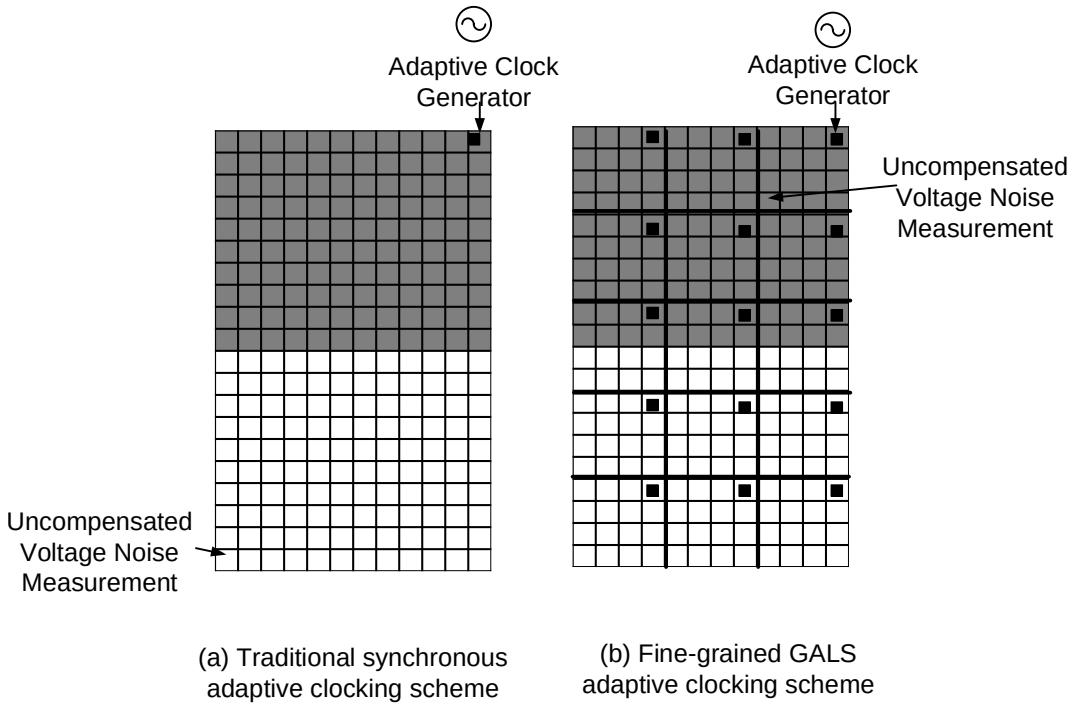
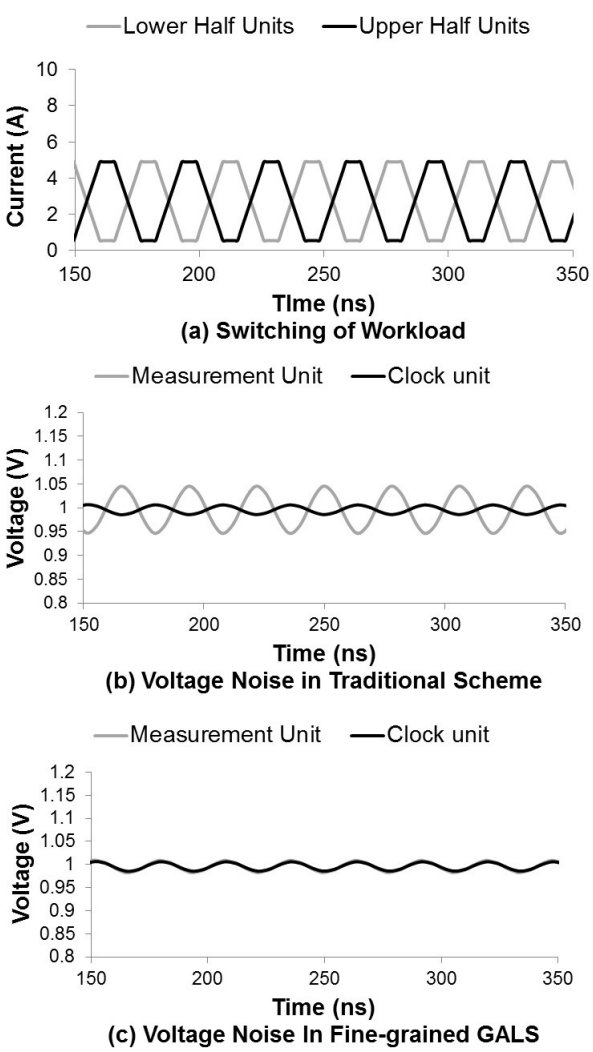


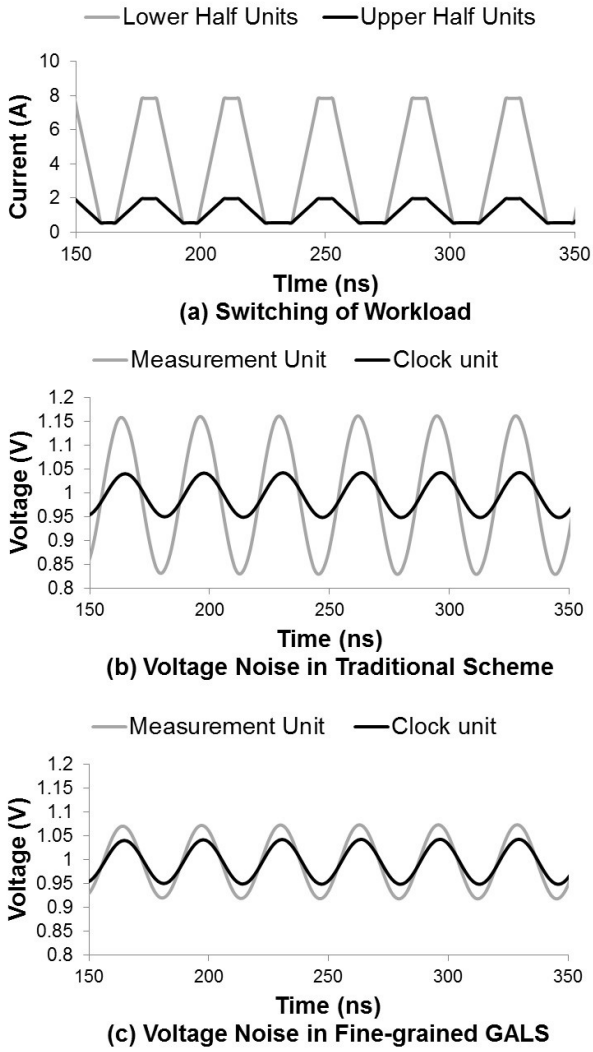
Figure 3.16: (a) In the traditional scheme, *uncompensated voltage noise* is measured at a unit farthest from the clock unit. (b) In the proposed scheme, a clock generator is present in every GALS unit.

clock. We present two cases of worst-case workload variations to demonstrate the mitigation of the effect of spatial workload variations using fine-grained GALS adaptive clocks.

1) Case 1: All units of the partition consume equal current switching at 30 MHz frequency, which is chosen to generate a worst-case supply noise. However, the workload current switches at a phase difference of 180^o between the lower and the upper half of the partition as shown in Figure 3.17a(a). The remaining seven partitions of the SoC are assumed to be idle. Figure 3.17a(b) shows the difference in voltage fluctuations between the clock and the measurement units. The units from the idle partitions that are in close proximity to the clock unit cause it to have lower supply noise than the measurement unit. For a traditional GALS design with an insertion delay of 1.5 ns, the *uncompensated voltage noise* is 59 mV. For a fine-grained GALS adaptive clocking scheme, Figure 3.17a(c) shows that there is very little difference in the voltage fluctuations between clock and measurement units and the *uncompensated voltage noise* is 5 mV. Therefore, the *uncompensated voltage noise* savings provided by the



(a)



(b)

Figure 3.17: (a) Case 1 and (b) Case 2 for the effect of spatial workload variations. The current profile, supply noise variation in clock and measurement units for the traditional clocking and the fine-grained GALS clocking scheme are shown for the 2 cases.

fine-grained GALS scheme is 54 mV.

2) Case 2: To demonstrate a higher supply noise variation, the lower half units of the partition is assumed to consume 80% of the total partition power and the upper half units consume the remaining 20%. This current profile of the partition is illustrated in Figure 3.17b(a). The remaining seven blocks of the SoC are idle and a workload switching frequency of 30 MHz is chosen for the worst-case supply noise scenario. As expected, the supply noise

is higher in the measurement unit (lower half) than the clock unit for the traditional GALS clock scheme as shown in a Figure 3.17b(b). A higher difference in supply noise gives an *uncompensated voltage noise* of 111 mV for the traditional synchronous GALS design with a 1.5 ns insertion delay. For the fine-grained GALS adaptive clocking scheme, the difference in supply noise fluctuation between clock and measurement is lower as shown in Figure 3.17b(c). The *uncompensated voltage noise* in this case is 33 mV. There could potentially be other worst-case workloads, especially when the remaining partitions are also switching, which could lead to a higher *uncompensated voltage noise* savings. In this case, the voltage margin savings provided by the fine-grained GALS adaptive clocking scheme is 78 mV.

GALS Partition Area

We used the same worst-case workload profiles as in Case 2 above, and studied the effect of the GALS area partition on *uncompensated voltage noise*. As expected, the savings in *uncompensated voltage noise* increases with lower GALS partition area as shown in Figure 3.18. The corresponding power savings (in %) from *uncompensated voltage noise* savings with lower GALS partition area is also shown in Figure 3.18. It is to be noted that the overhead from inter-partition data-transfer and local clock generators are not considered in this plot.

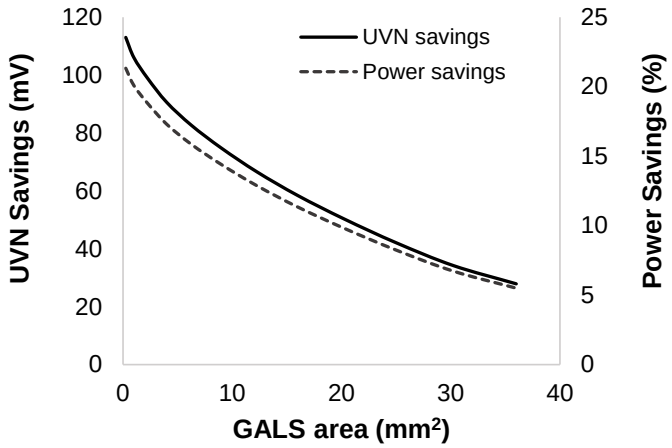


Figure 3.18: GALS area vs. *uncompensated voltage noise* and corresponding power savings.

The lowest GALS area considered in the plot is 0.25 mm^2. However, the realistic area of the GALS partition is determined by architectural factors and decisions. The realistic power savings are also determined by the impact on performance caused by asynchronous interfaces between GALS partitions. The potential overheads are discussed in the next section.

Overheads

The sizes of the fine-grained GALS partitions determine the *uncompensated voltage noise* savings and therefore, the power savings due to reduced voltage margins. Lower dimensions of the GALS units lead to higher *uncompensated voltage noise* savings. However, there is a limit to which the size of a fine-grained GALS unit can be reduced. In this section, we discuss how the optimal size of each GALS unit can be determined. The optimal size of each clock domain in fine-grained GALS depends on the trade-off between the savings from the power supply noise tolerance and the overheads from the interfaces crossing asynchronous clock domains and the many local clock generators.

Data-transfer Overhead: Data-transfer on processors inevitably results in latency that affects performance. In [51][52][53], a series of micro-benchmarks are used to quantify the access latency of a GPUs global and shared memory. Global memory latency is the full-time in accessing data located in DRAM/L2 or L1 caches, including the page table lookup latency. In the fine-grained GALS clocking scheme, there is an additional latency penalty from asynchronous interfaces at the clock domain crossings (CDC). The signals crossing the boundaries of the various asynchronous clock domains are required to be synchronized to avoid operational failure. A simple implementation of a synchronizer called the brute-force synchronizer involves a series of flip-flops that samples a signal from one clock domain to another. This synchronizer is able to reduce the probability of metastability by inserting the extra clock cycles, but this also means that it comes with a synchronization latency penalty of a few clock cycles. Brute-force synchronizers can also be combined with first-in

first-out (FIFO) queues [54] for efficient transfer of data across the boundaries. A more fitting candidate for GALS schemes with adaptive clocks is potentially the pausible bisynchronous FIFO [54] that integrates well into standard tools. It has a very low average latency of only 1.34 cycles as compared to a 1 cycle latency of a synchronous crossing, while incurring a minimal area and power overhead.

In our fine-grained GALS architecture, each traditionally synchronous core was divided into 15 partitions of approximate area 4 mm^2. [51] discusses the global access latency for different access patterns on multiple GPU architectures. For instance, in pattern P1, data is repeatedly loaded in a cache line so that every memory access is a cache hit. The global access latency for P1 on a GTX780 (Kepler architecture) is 198 cycles. Suppose that in the fine-grained GALS scheme, an L2 cache access has 15 additional clock-domain crossings. If a traditional brute-force synchronizer is used, a penalty of 3 cycles from each clock domain interface as compared to synchronous crossings translates to a $\sim$23% increase in latency for an access pattern P1 of [51]. On the other hand for an approximately half-cycle penalty for low-latency asynchronous crossings such as [54] compared to synchronous crossing, the latency penalty is less than 4%. Table 3.1 lists the average latencies and latency penalties of state-of-the-art synchronizers compared to synchronous crossing.

Figure 3.19 shows the percentage increase in latency penalties for different numbers of GALS partitions per core compared to a baseline core using synchronous boundary interfaces with 1 cycle latency. The corresponding GALS partition area (in mm^2) is also indicated. The trend for both brute-force synchronizer and the low-latency pausible bisynchronous synchronizer is shown. As expected, the latency penalties increase with increasing number of

Table 3.1: Average latency of synchronizers

Synchronizer	Average latency	Latency penalty for L2 access on GTX780
Brute-force [54]	4	23%
Pausible Bisynchronous [54]	1.34	4%
Synchronous	1	0%

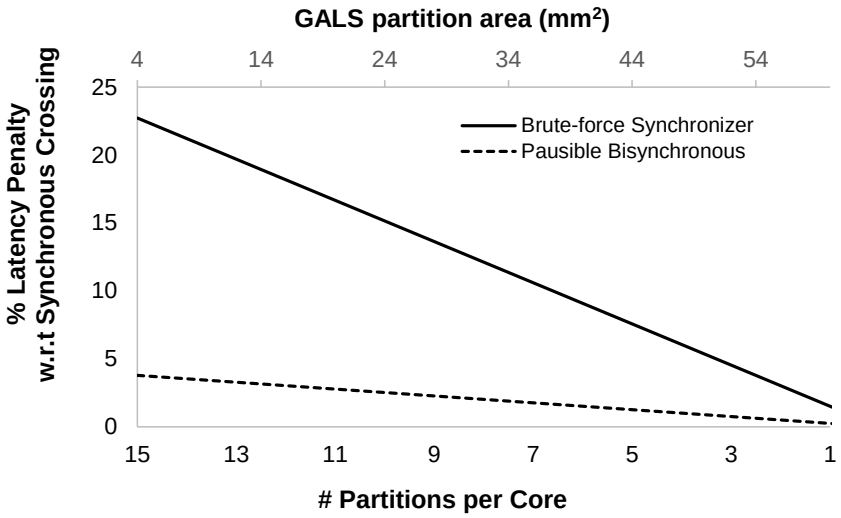


Figure 3.19: Latency penalties versus partitions per core (and versus GALS partition area).

GALS partitions.

Finally, we qualitatively examine the effect of latency penalties on processor performance. A quantification of the performance penalties is possible using architectural level simulators, which is outside the scope of this thesis. In [55], the authors demonstrate the effect of latency on performance for various mesh networks for GPUs. It is shown that the performance is not affected dramatically with small variations in latency if the interconnection network provides sufficient bandwidth. This is possible because GPUs are specifically designed to tolerate high memory latencies. For instance, in a network-on-chip (NOC) scenario, an extra 4-cycle latency per router of a 2-cycle baseline latency is well-tolerated across different benchmarks and causes an average performance loss of only 3.5%. As described previously and shown in Figure 3.19, with an estimated increase in latency by less than 4% with recent fast-crossing interface circuits such as [54] and the inherent latency-tolerance of GPUs, we safely expect the impact on GPU performance to be very low.

Clock Generator Overhead: In the floorplan used in our analysis (with 2 mm x 2 mm GALS units), the total number of local clock generators across the SoC adds up to 138.

Digitally controlled oscillators are capable of generating clocks in the MHz-GHz frequency range for such HP processors. These clock generator circuits respond to critical-path replica circuits powered by the noisy supply voltage [56][57] and they make good candidates for such local clock generators. The power consumption of such circuits are usually only a few mWs [58] and constitute only a small percentage (less than 1%) of the SoC power consumption. A recent DCO architecture proposed in [59] is designed to drive such GALS units and is capable of operating at multiple frequencies up to 1 GHz and consumes approximately 200 μW. Its area in a 65 nm technology is approximately 850 μm^2, which is less than 0.025% of a 2 mm x 2 mm GALS unit.

To summarize the above section, we present models and quantitatively estimate the benefits of fine-grained GALS adaptive clocking scheme for HP processors compared to the traditional adaptive clocking scheme. We also qualitatively examine its data and clock overheads. We conclude that a fine-grained GALS adaptive clocking scheme exhibits more tolerance to the local effects of power supply noise. A 78 mV savings in *uncompensated voltage noise* translates ~15% savings in power for the same performance.

Next, we analyze a variation tolerant design technique for performance-relaxed ULP SoCs.

3.3 Performance-Relaxed Ultra-Low Power Systems-on-Chip

As discussed in Section 3.1, we focus on the effect of PVT variations in the digital design domain of performance-relaxed ULP IoT SoCs. Traditionally, hold time closure is performed at design-time. It involves adding long chains of traditional-buffers in hold-critical data-paths. However, with the traditional technique, an underestimation of the hold time margins can lead to chip failure, and an overestimation can cause an increase in power and area. The design effort toward hold margin estimation worsens with a higher impact of PVT variations in sub-V_T chips. Therefore, the above solution has limited control and flexibility over the

chip functionality, reliability, and yield. This gives rise to the need for a post-silicon solution for hold time closure, which can mitigate the design effort in estimating timing margins, guarantee a high chip yield, and potentially reduce the power and area overheads due to traditional overestimation of margins.

Traditionally, post-silicon V_{DD} and frequency tuning mechanisms to maximize energy efficiency and to avoid setup time failures [60] are available. Recently, authors in [61] demonstrated post-silicon tunability for hold time as well, but for latch-based designs. It involves the complexity of generation and distribution of two-phase clocks. Latch-based designs also require additional re-timing and verification efforts compared to flip-flop-based designs. In flip-flop-based designs, previous work proposed the use of post-silicon tunable-buffers in the clock path for clock-skew (t_{skew}) minimization [62]. However, tuning t_{skew} for one path may impact the timing in other paths. Therefore, this technique requires a robust architecture for optimal tunable-buffer placement in the clock-tree. Phase-detectors to observe t_{skew} and control mechanisms to tune t_{skew} add to the complexity, power, and area, especially in low-power designs. In this chapter, we introduce a post-silicon hold time closure technique for performance-relaxed ULP flip-flop-based designs. It involves the insertion of tunable-buffers in the hold-critical data-paths as shown Figure 3.20. We analyze and discuss the potential benefits of the scheme and the costs involved.

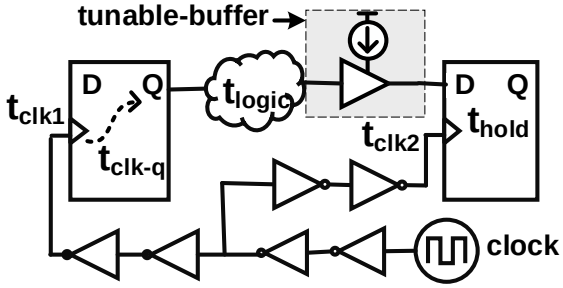


Figure 3.20: Tunable-buffers are inserted in the data-path for post-silicon hold time closure.

3.3.1 Approach

The robustness of the traditional-buffer insertion technique for hold time closure depends on the hold slack target decided by the designer at the time of design. Hold slack is indicative of the amount of additional margin that a design has, to overcome the impact of unpredictability in PVT. A positive hold slack of a timing path implies reliable functionality, and a negative hold slack implies circuit failure. The number of traditional-buffers inserted, and therefore, the circuit reliability increases with the amount of hold slack target specified at the time of design. Therefore, the challenge lies in determining the optimal amount of hold slack target. To overcome this design effort involved in estimating optimal timing margins, we introduce the concept of using tunable-buffers in the data-path instead of traditional-buffers as shown in Figure 3.20. The delay of these tunable-buffers can be controlled post-silicon. Therefore, compared to traditional-buffer insertion, the tunable-buffer technique mitigates the effort for hold margin estimation at the time of design by enabling post-silicon hold correction.

A tunable-buffer will have a wide delay range that can be varied during operation for post-silicon tunability. However, we still need to insert these tunable-buffers at the time of design. Therefore, a fixed value of the tunable-buffer delay must be chosen so that they can be inserted into hold-critical data-paths using standard commercial tools. In this section, we discuss how to arrive at this fixed delay value of the tunable-buffer for design-time. This chosen value impacts the trade-off between the number of tunable-buffers inserted and the post-silicon tunability. In this section, we analyze the above trade-off to a first order.

Factors That Affect the Number of Tunable-Buffers and Post-Silicon Tunability

Distribution of Hold-Critical Paths: The number and the length of hold-critical paths vary across different blocks. This is illustrated in Figure 3.21, which shows the examples of two cases. In block#1, most hold-critical paths are of similar lengths. Therefore, a similar number of tunable-buffers will be inserted in each path. On the other hand, block#2 has hold-critical paths of a wide variety of lengths and therefore, different numbers of tunable-buffers will

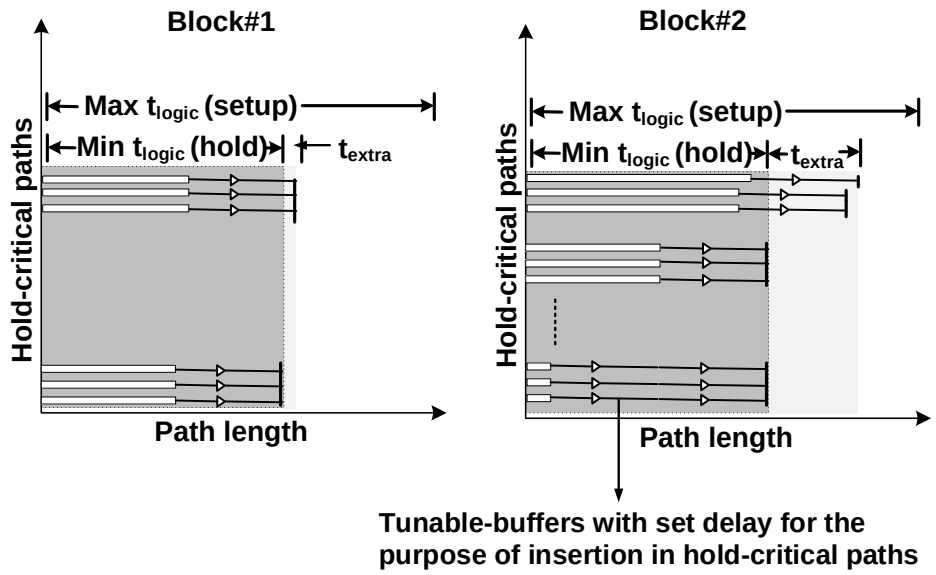


Figure 3.21: Depending on the block, the hold-critical data-paths can be of different lengths. The number of tunable-buffers inserted depends on this hold path distribution.

be inserted in different paths. The number of tunable-buffers inserted depends on this hold path distribution. It is to be noted that the impact on setup time is not a concern in this approach, because our target SoCs operate in the kHz range frequencies. As illustrated in Figure 3.21, even after tuning the tunable-buffer and increasing data-path delays, the system satisfies the setup time constraint.

Nature of the Tunable-buffer: The tunable-buffer will require a control signal to "tune" its delay. The control signal may be a digital or an analog signal, depending on the tunable-buffer implementation. In the Chapter 4, we derive that the tunable-buffer controlled by an analog voltage is a good candidate for this scheme. Therefore, we assume that the control is an analog voltage signal ($Vctrl$) for the purpose of our analysis. As discussed previously, to be able to insert these tunable-buffers with a tool-flow using standard commercial tools, a set value for $Vctrl$ needs to be chosen at design-time. We call this value $Vctrl_{design}$.

The chosen $Vctrl_{design}$ value impacts the number of tunable-buffers and post-silicon tunability inserted in each path. If a lower value is chosen, it corresponds to a lower tunable-buffer delay during insertion. Therefore, more number of tunable-buffers will be inserted to

meet a preliminary hold slack target specified by the user, but it also gives more opportunity for post-silicon tunability. If a higher $Vctrl_{design}$ is chosen, it corresponds to a higher tunable-buffer delay during insertion, but it gives a lesser opportunity for post-silicon tunability. Therefore, it is crucial to choose an optimal value. A first-order experimental setup and analysis to derive this is value explained in the next section.

Experimental Setup

A first-order experimental setup to derive $Vctrl_{design}$ is shown in Figure 3.22. In this section, we describe the inputs, outputs and equations for the estimation models. The outputs of this model enable derivation of an optimal $Vctrl_{design}$ value.

Inputs

(a) Pre-closure hold slack: This input describes the *distribution of hold-critical paths* in a block before hold time closure. As mentioned previously, hold slack is indicative of the available hold timing margin. It indicates if the hold time criteria are satisfied for each data-path. Negative hold slack indicates failure. Ultimately, the goal of the tool is to make the hold slack of each data-path to be greater than the hold slack target provided by the designer. Pre-closure hold slack refers to the available hold slack of each data-path before hold time closure. Essentially, this input file contains the hold slack numbers of all the data-paths after clock tree synthesis and before adding hold buffers. This input is depicted using histograms as shown in Figure 3.23a and 3.23b. A 32-bit shift register (SR) in which data is shifted from one register to the next, and all the data-paths are of approximately similar lengths, the pre-closure hold slack is of similar values. However, a finite impulse

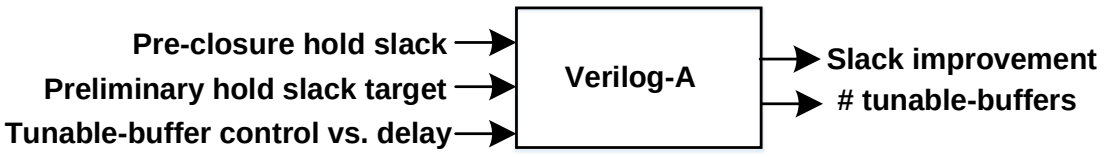
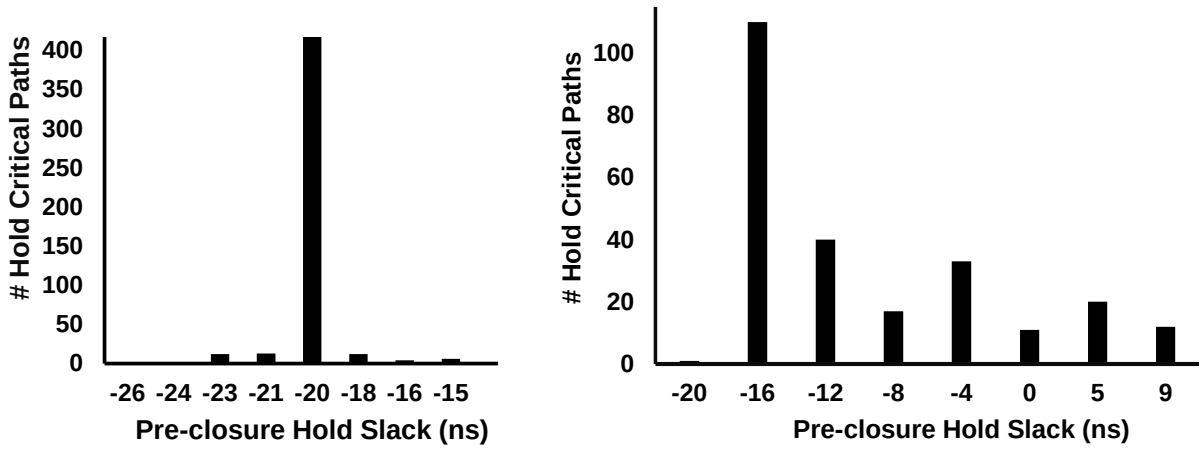


Figure 3.22: Cost-benefit analysis for tunable-buffer insertion in data-paths. From the outputs of the model, we are able to estimate an optimal $Vctrl_{design}$ value.



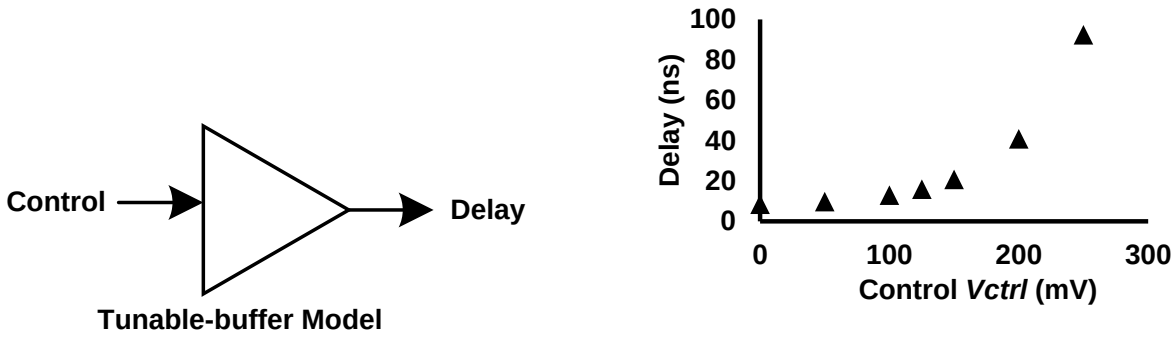
(a) (b)

Figure 3.23: Pre-closure hold slack (before hold buffer insertion) (a) for shift register. (b) for FIR block.

response (FIR) filter block, in which data-paths are all of different lengths, has a wide range of values for the pre-closure hold slack.

(b) Preliminary hold slack target: This is the preliminary hold target number that the designer specifies to the tool. The block is required to satisfy this requirement at design time. With the use of tunable-buffers instead of traditional-buffers, the design will be capable of post-silicon tunability and therefore a higher post-silicon hold slack than the preliminary hold slack target.

(c) Tunable-buffer control vs. delay: This input describes the *nature of tunable-buffers*. Figure 3.24a shows the abstract model of the tunable-buffer. Therefore, we generate a control (i.e., *Vctrl*) vs. delay dependence from simulations. The tunable-buffer delay is dependent on the process, temperature, input slew rate, and load capacitance. We run the simulations in which hold timing closure is to be performed (we chose FF:25^{o}C, the process corner which showed worst-case variations in hold slack due to data-path delays and clock skew). We observe that with varying typical input slew rates, the tunable-buffer delay variation is less than 5%. Therefore, we ignore this variable. The load capacitance is also a varying factor for the tunable-buffers in different hold-critical data-paths. It depends on the standard-cell



(a) (b)

Figure 3.24: (a) Tunable-buffer model. (b) Control ($Vctrl$) vs. tunable-buffer delay

.

and the interconnect that it is driving. Since it is a challenge to extract path-dependent load capacitances, we use a value at the lower end of the spectrum in this technology (4 fF). Lower capacitance implies lower tunable-buffer delay, and it, therefore, adds pessimism to the number of tunable-buffers added according to our model. The control ($Vctrl$) vs. tunable-buffer delay is shown in Figure 3.24b. The tunable-buffer delay is chosen as the lower value between the rise and fall delay for pessimism in the number of tunable-buffers added.

Outputs

(a) Slack improvement: This output is indicative of the amount of post-silicon tunability that can be obtained with different combinations of the inputs. The extent of post-silicon tunability is evaluated in terms of *post-Si hold slack*, which is the maximum post-silicon hold slack possible by tunable-buffer insertion. Moreover, to normalize this value, we represent it as *slack improvement*, which is the percentage improvement of the *post-Si hold slack* over the hold slack achieved by traditional-buffer insertion. This is a first-order estimate of the benefits of the post-silicon hold time closure scheme.

*slack improvement = (post-Si hold slack - traditional hold slack)/traditional hold slack*100*

(b) # tunable-buffers: This output indicates the number of tunable-buffers inserted in a block corresponding to different input combinations. This is a first-order estimate of the costs

associated with this scheme. To understand the actual savings/overhead in the number of buffers, a percentage increase or decrease in # of tunable-buffers over # of traditional-buffers must be derived. However, the buffer savings/overhead are dependent on the tunable-buffer circuit structure. Therefore, we model # tunable-buffers in this section. We estimate the buffer savings based on the tunable-buffer structure as derived in Chapter 4.

*buffer savings/overhead = (#tunable-buffers - #traditional-buffers)/#traditional-buffers*100*

By analyzing the above two outputs of the model, we can derive an optimal $Vctrl_{design}$ to be used for the design of any block using tunable-buffer insertion.

Model

The amount of possible post-silicon tunability (*slack improvement*) and the number of inserted tunable-buffers in a block (*#tunable-buffers*) are dependent on the three inputs of the model: *pre-closure hold slack*, *preliminary hold slack target*, and the *tunable-buffer control vs. delay* profile. Although the exact values for *#tunable-buffers* and *slack improvement* can be precisely determined with extensive repetition of the design process for different combinations of the inputs, we save a lot of design effort by deriving a first-order estimate. The following equations represent our first-order model.

(a) Actual hold slack target: This is the hold slack target that needs to be achieved by the tool by inserting tunable-buffers after the clock-tree synthesis. We calculate the *actual hold slack target* for a given block containing N hold-critical paths as:

actual hold slack target[N:1] = preliminary hold slack target - pre-closure hold slack[N:1]

(b) #tunable-buffers: Next, we calculate the number of tunable-buffers inserted in each of the N hold-critical paths to meet their respective *actual hold slack target*, for different tunable-buffer control signal (*Vctrl*) values. This varies because the tunable-buffer delay is a function of *Vctrl* (Figure 3.24b).

#tunable-buffers[N:1] = CEILING(actual hold slack target[N:1]/tunable-buffer delay)

The total number of tunable-buffers inserted in the given block is given by the sum of tunable-buffers in all the N hold critical paths.

$$\#tunable\text{-}buffers: \Sigma\#tunable\text{-}buffers[N:1]$$

The CEILING function introduces a pessimism (higher number) in *#tunable-buffers*, because the value is rounded to the next highest integer. We observe that a more optimistic *#tunable-buffers* is obtained when the CEILING function is not used (fractional values of *#tunable-buffers[N:1]* are summed to obtain *#tunable-buffers*). Therefore, we obtain an optimistic and pessimistic curve for *#tunable-buffers* using this model, for different values of inputs and *Vctrl*.

(c) Post-Si hold slack: Finally, we calculate the maximum possible post-silicon tunability, for different *Vctrl*. *Max tunable-buffer delay* is the maximum *tunable-buffer delay* at the maximum possible *Vctrl*.

$$post\text{-}Si\ hold\ slack = MIN(pre\text{-}closure\ hold\ slack[N:1]+\#tunable\text{-}buffers[N:1]*max\ tunable\text{-}buffer\ delay)$$

The minimum function indicates the lowest value of post-silicon tunability amongst all the N hold-critical paths. As a result of both pessimistic and optimistic curves for *#tunable-buffers*, we obtain two such curves for *slack improvement* also. A pessimism in *#tunable-buffers* (higher number) implies an optimism in *slack improvement* (more tunability).

In addition to the approximation of tunable-buffer delay based on the load capacitance and input slew rates, this first-order estimation also neglects the tweaks made by the commercial tools during place-and-route (standard-cell resizing, the addition of design rule violation buffers, etc.), which also impact the values of different hold-slacks.

3.3.2 Results

Using the above first-order model, we can estimate the benefits of the post-silicon hold time closure approach (*slack improvement*) and its cost (*# tunable-buffers*). The model also

enables determination of the inputs to the tool (*preliminary_hold_slack_target* and $Vctrl_{design}$) for the optimal design of any block.

Figure 3.25 shows the outputs of the model, *slack improvement* and *# tunable-buffers*, for the SR block across different values of *preliminary_hold_target* (0, 10 ns and 20 ns) and different values on *Vctrl*. The legend of the plot follows the naming convention with respect to the *#tunable-buffer*. Pessimism implies a higher number of tunable-buffers and a higher amount of post-silicon tunability. The plots are intended to be discrete scatter plots for different values of *Vctrl*, but the dots are connected for the sake of clarity. As discussed previously, the SR block has hold-critical data-paths of similar lengths as shown in Figure 3.23a. The number and distribution of hold-critical paths heavily impact *slack improvement* and *#tunable-buffers*. Therefore, it is valuable to run a pre-design analysis for every block using this model.

As expected, we observe that the *#tunable-buffers* increases for higher *slack improvement*. The model estimates optimistic and pessimistic curves for both *slack improvement* and *#tunable-buffers*. We also indicate the real values in the plot, which are derived from the actual design methodology using standard tools (discussed in Chapter 5). The real values lie in between the optimistic and pessimistic curves. Therefore, our model enables a good pre-design estimate to save a lot of design effort to determine the trade-offs for every block.

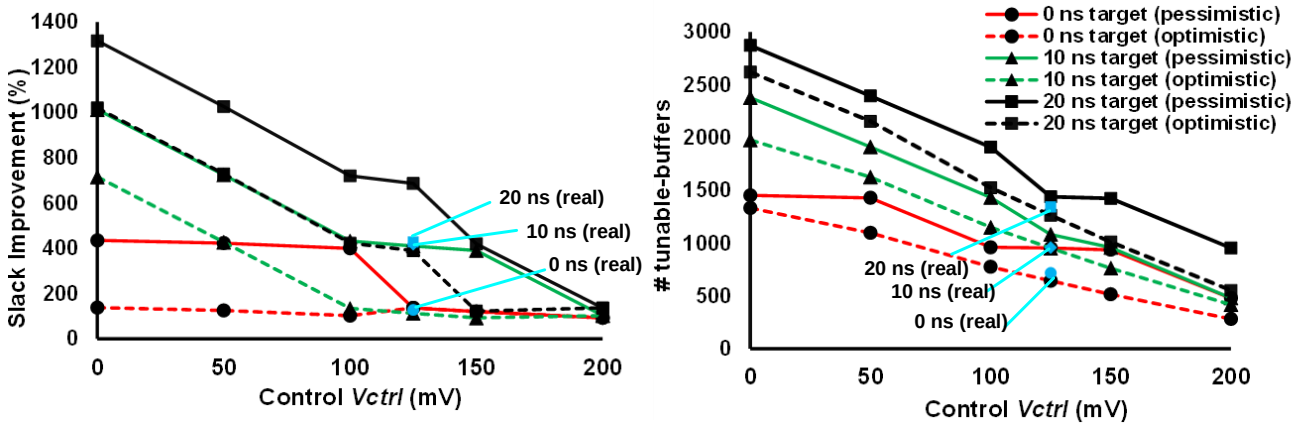


Figure 3.25: Slack improvement and #tunable-buffers for SR.

It is useful to examine the plot for the amount of post-silicon tunability that the designer requires, and then use the corresponding values of *preliminary_hold_slack_target* and $Vctrl_{design}$ in the tool-flow, because this will enable a higher savings or lower overhead in the number of tunable-buffers. However, the actual cost (overhead/savings) in the number of buffers depends on the tunable-buffer design. In Chapter 4, we will discuss the details of a good candidate for the tunable-buffer design. Using this tunable-buffer structure, our model estimates the buffer savings to be ∼10 to 30% (0 ns target and $Vctrl = 125$ mV) for the SR block. In chapter 5, we see savings in this range using the actual design methodology.

Figure 3.26 shows the outputs of the model for FIR block. It follows different trends compared to SR due to a different hold path distribution. We observe that the pessimistic slack improvement is negative (does not meet timing) with *preliminary_hold_target* of 0ns and 10ns. This indicates that this may not be sufficient for the FIR design. Such observations are crucial to determining an estimate for the inputs to the tool *preliminary_hold_target* and $Vctrl_{design}$ to lower the design effort involving input selection from a wide range of combinations.

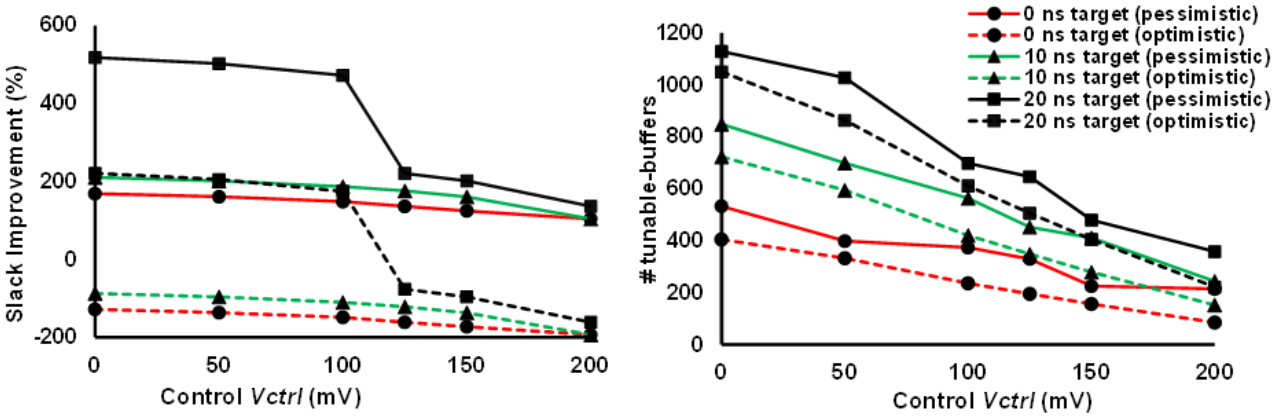


Figure 3.26: Slack improvement and #tunable-buffers for FIR.

3.4 Conclusion

In this chapter, we discussed the impact of voltage variations on HP processors. Traditional synchronous adaptive clocking schemes cannot fully compensate for the effects of power supply noise in HP processors. We propose the fine-grained GALS adaptive clocking scheme to tolerate power supply noise. We present the system-level models and analysis to quantify the benefits of the above scheme. From our experimental setup for GPU-like chips of large areas, we obtained a 78 mV savings in *uncompensated voltage noise* at a nominal supply voltage of 1 V, with the use of the fine-grained GALS adaptive clocking scheme compared to the traditional synchronous adaptive clocking scheme. This is an equivalent of 15% savings in power consumption for the same processor performance. We also qualitatively discuss the potential buffer overheads or savings in the implementation of the proposed scheme.

We also the discuss the impact of PVT variations on performance-relaxed ULP chips. The effects of PVT variations are much higher in sub-V_T circuits. Traditional buffer insertion is challenging due to the need to correctly estimate the timing margins. Therefore, we propose the tunable-buffer insertion scheme to enable post-silicon hold time correction. We present the analysis to quantify the benefits and potential overheads of the above scheme. It allows us to make a first-order estimate of *preliminary_hold_slack_target* and $Vctrl_{design}$ inputs to the tool-flow that can provide the maximum *slack improvement* with minimum *#tunable-buffers*.

Chapter 4

Design for Variation Tolerance in Internet-of-Things Systems-on-Chip

4.1 Background

[1] Among the issues faced by the designers of ULP circuits, the issue of variations is a critical one. Sub-V_T operation, in which circuits are operated at V_{DD} less than the transistor V_T, is an attractive option for ULP IoT sensor nodes in which speed is not the primary concern. In digital circuits, dramatic energy reductions can be achieved due to a quadratic dependence of active energy on V_{DD}. Energy-harvesting IoT nodes such as [13][9] demonstrate benefits from the use of such sub-V_T digital designs. However, along with the ULP benefits of sub-V_T operation, the negative impacts of PVT variations are high at low V_{DD}.

Process variations are an inevitable consequence and limitation of current chip fabrication technologies. These variations and mismatches in transistors' parameters cause wide variations in their currents, even more so in sub-V_T. For instance, simulations show that the FO4 X1 inverter delay (130 nm bulk CMOS, 27 ^oC) varies by ∼16X across slow to fast corners at 0.3 V V_{DD} and only by ∼2X at 1 V V_{DD}. Such wide delay distributions as shown in Figure 3.3

[1] This chapter derives content from [DAK2][DAK3][DAK5][DAK11]

make reliable circuit operation challenging in sub-V_T. Similar to process variations, variations in ambient temperature and supply voltage are also possible in such SoCs that also have a high impact on both digital and analog circuits.

Below is an example of how voltage variations can occur in such SoCs. For example, dynamic voltage scaling (DVS) across a wide V_{DD} range and partitioning the SoC into multiple V_{DD} domains, are key power management techniques for achieving energy efficiency in emerging IoT chips. DC-DC converters such as low drop-out, switched-capacitor, and single-input multiple-output designs are typically used for power delivery in such SoCs [9]. However, using smaller on-chip capacitors instead of large off-chip capacitors, varying output load, cross-regulation issues, and slow-transient response contribute to V_{DD} variations and low-frequency ripple. Sub-V_T circuits in such SoCs are highly sensitive to V_{DD} variations.

Therefore, circuits that target PVT variation tolerance are crucial in recent ULP IoT sensor systems. As a step toward this, in this chapter, we first propose the design of sensors that detect such variations. We discuss a temperature sensor design that is operational in the sub-V_T region for ULP operation. V_{DD} monitoring is also crucial in V_{DD} regulators [63], and in circuit techniques such as adaptive clock generation [64] and distribution [65], timing error prevention [66] etc. In this chapter, we present a ULP voltage monitor architecture for monitoring low-frequency V_{DD} variations. We emphasize on sub-V_T, low-V_{DD} monitoring. Finally, we present the design of circuits namely tunable-buffer and bias generator, for the post-silicon hold time closure technique discussed in Chapter 2, to overcome the effect of PVT variations. The circuits discussed in this chapter aid variation tolerance in such SoCs targeting ULP consumption.

4.2 Temperature Sensor

Various types of temperature sensors based on both bipolar junction transistors (BJTs) and MOSFETs have been traditionally designed. However, BJT-based temperature sensors

such as [67] have power consumption in the μW range, which is not suitable for devices operating from harvested energy. BJT-based temperature sensor architectures also require a higher power supply (1.5 V to 2 V in [67]) that limits the minimum supply voltage in IoT devices. On the other hand, MOSFET-based temperature sensors [68][69][70][71] are growing in popularity these days for ULP systems. These temperature sensors consume less power in the tens to hundreds of nWs range as compared to the BJT-based sensors at the expense of accuracy. In [68], a temperature sensor operational from 0.85 V V_{DD} was implemented using dynamic threshold MOS transistors (DTMOSTs) as temperature-sensing devices and an inverter-based zoom ADC to enable a power consumption of 600 nW. In [69], the MOSFET-based temperature sensor consumes only 220 nW while operating continuously. The work in [69] demonstrates techniques to lower power consumption such as sub-V_T design, use of frequency-to-digital conversion instead of ADC, an on-chip time reference, etc. that makes this implementation suitable for embedded passive RFID application space. In [70], a CMOS temperature sensor employs a serially connected sub-V_T MOS as a sensing element that is operational from 0.5 V V_{DD}, thereby enabling a power consumption of 119 nW. In [71], a new temperature sensor topology and changes to the conventional voltage-to-current converter and current mirror structures were proposed to achieve a power consumption of 71 nW.

A high V_{DD} is traditionally used in analog circuits for greater headroom requirements. However, in recent ULP systems, analog circuits are being operated at a lower V_{DD} for power savings and also to directly power the circuits using energy-harvesting methodologies such as thermoelectric, solar, etc. For example, in [12] a 2.4 GHz RF receiver was designed to operate at 300 mV V_{DD}. In [72], it was demonstrated that lowering the amplitude of oscillation reduces the power consumption of a XTAL oscillator, and it was designed to operate at voltages as low as 300 mV. In this chapter, we propose to further lower the power consumption of a temperature sensor system with the use of a sub-V_T sensor core that consumes very low power and energy and is operational down to 0.2 V V_{DD} [73].

4.2.1 Approach

A simplified block diagram of the proposed temperature sensor is shown in Figure 4.1. A current proportional to ambient temperature is generated using a sub-V_T temperature to current converter. The current is subsequently translated to frequency using a current-controlled oscillator (CCO) and then into a representative digital code using a digital block, which is usable by any SoC. Modifications to the conventional current mirror structure enable reduced power consumption in cases such as fast process corners in which power consumption is usually higher than desired.

The proposed temperature sensor core consists of (a) a sub-V_T PTAT current element that generates a current proportional to temperature. (b) a bit-weighted current mirror (BWCM) that mirrors the PTAT current to starve a ring oscillator. Process variations can cause a high amount of variations in the sub-V_T PTAT current output. A BWCM consisting of 8 current mirror arms can be enabled or disabled to control the PTAT current that is mirrored to the subsequent stages (1x implies same current mirrored, 1/2x implies half the current mirrored, etc.). (c) a CCO that generates a clock frequency proportional to the mirrored PTAT current. The sensor core constituting the above components is operational at 0.2 V V_{DD}. Additionally, a digital block converts the CCO oscillation frequency to a digital code for processing by any SoC component. The digital block operates at 0.5 V V_{DD}, a higher voltage rail that is

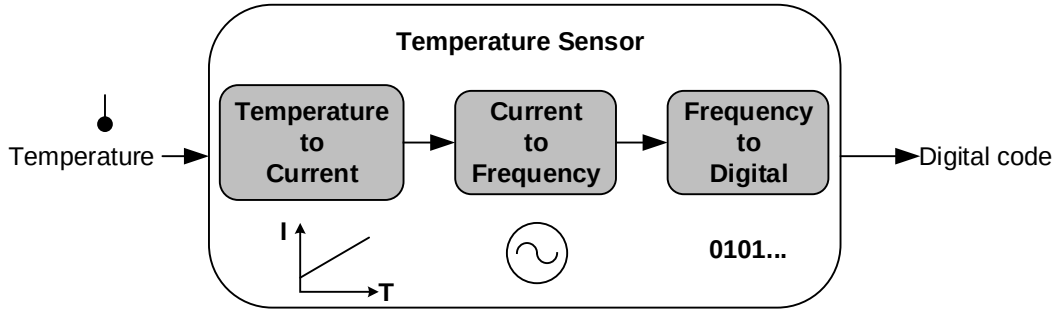


Figure 4.1: A block diagram of the proposed temperature sensor system.

typically available for digital processing on SoCs such as [9]. The 0.2 V analog supply can be generated from the 0.5 V supply using a switched-capacitor DC-DC regulator or low-dropout (LDO) regulator.A detailed description of these components is presented in the next section.

PTAT Current Circuit

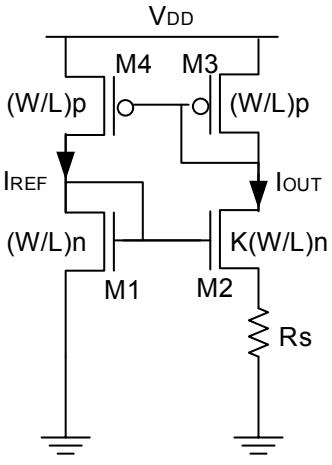
Figure 4.2a shows the PTAT current source using a single resistor designed to operate down to 0.2 V V_{DD}. It generates current that increases linearly with temperature. Transistors M1 to M4 operate in the sub-V_T saturation region. Sub-V_T current is given by:

$$I_{DSUB}=I_o\exp((V_{GS}\text{-}V_T)/\mathrm{n}\phi_t)(1\text{-}\exp(\text{-}V_{DS})/\phi_t) \tag{4.2.1}$$

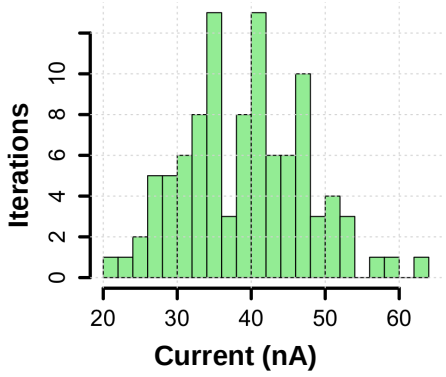
where I_o is drain current when gate-source voltage (V_{GS}) equals V_T, μ_o is the carrier mobility, C_{ox} is the gate oxide capacitance, W and L are the channel width and length, and n is the sub-V_T slope factor. When drain to source voltage V_{DS} >3ϕ_t (ϕ_t=kT/q), the term exp(-V_{DS})/ϕ_t in (4.2.1) starts becoming negligible and (4.2.1) can be approximated as:

$$I_{DSUB}=I_o\exp((V_{GS}\text{-}V_T)/\mathrm{n}\phi_t) \tag{4.2.2}$$

This is called the sub-V_T MOSFET saturation region, in which the drain current becomes independent of V_{DS}. To overcome low headroom at 0.2 V, thin-oxide standard-V_T (SVT) devices with V_T of ∼0.2 V are used. Long channel lengths are used to reduce short-channel



(a)



(b)

Figure 4.2: (a) Sub-V_T PTAT current source operational down to 0.2 V V_{DD}. (b) PTAT current variation due to process variations.

effects. To obtain the equation for PTAT current, Kirchoffs voltage law is applied to transistors M1, M2 and resistor R_S. V_{GS1} and V_{GS2} are the gate to source voltages and I_{DSUB1} and I_{DSUB2} are the drain currents of transistors M1 and M2 respectively.

$$V_{GS1} = V_{GS2} + I_{DSUB2}R_S \tag{4.2.3}$$

Substituting (4.2.2) in (4.2.3) and assuming I_{DSUB1}= I_{DSUB2}= I_{OUT} and V_{T1}= V_{T2}:

$$I_{OUT}=\mathrm{n}\phi_t \; log_e\mathrm{K}/R_S \tag{4.2.4}$$

The direct temperature dependence of I_{OUT} on ϕ_t(=kT/q) gives a proportional to absolute temperature current. Due to the small values of I_{OUT} current, a smaller resistor as compared to prior PTAT designs can be used. However, such sub-V_T circuits are highly sensitive to process variations. Process variations can cause output current variations in the sub-V_T PTAT. Figure 4.2b shows a plot of the PTAT current at 27 ^{o}C in 100-point Monte Carlo simulations. We observe that the mean PTAT current is 39 nA, and the 3σ variation of 25 nA is very high. The PTAT current is mirrored into the subsequent current controlled oscillator stage. Therefore, higher PTAT current may cause higher total power dissipation than desired. Therefore, a bit-weighted current mirror (BWCM) is proposed to decrease such process-induced power dissipation.

Bit-weighted Current Mirror

The PTAT current is mirrored using a BWCM to starve the transistors of a CCO , which generates frequencies proportional to temperature.

The BWCM is shown in Figure 4.3. The BWCM consists of eight current mirror arms, each of them mirroring different fractions of PTAT current to the CCO. For instance, 1x implies that PTAT current is entirely mirrored, 1/2x implies that 50% of the PTAT current is mirrored and so on, while still maintaining linearity. The current mirror arms can be enabled or disabled by bits B<7:0> in the switches. Simple PMOS or NMOS switches cannot be used to enable or disable the current mirror arms at 0.2 V V_{DD} due to effects of high leakage current that interferes with the PTAT current thereby altering its linear

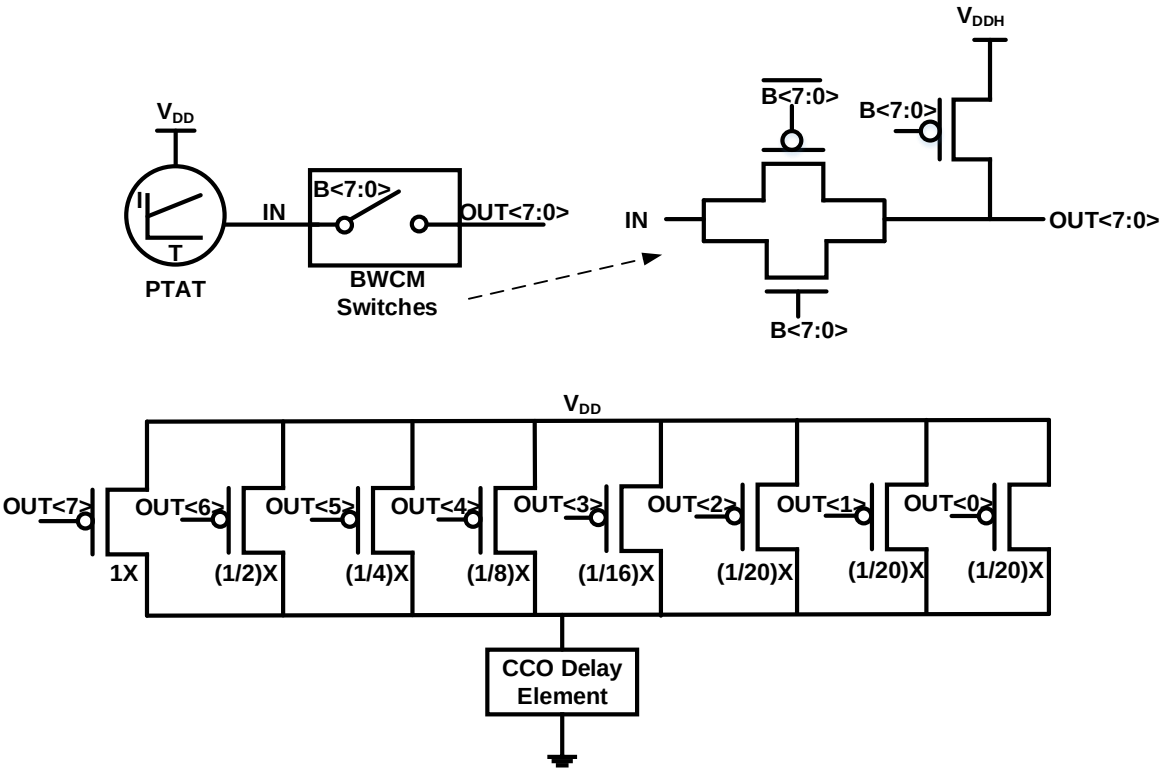


Figure 4.3: Bit-weighted current mirror circuit.

characteristics. Figure 4.3 shows the architecture of the BWCM switches used to turn on and off each bit-weighted current mirror arm. To enable an arm, the corresponding transmission gate switch is enabled by one bit of B<7:0>. To disable an arm, the output of the switch and therefore the gate of the PMOS of the corresponding current mirror branch is pulled high to turn it off. However, when pulling the PMOS gate to only 0.2 V V_{DD}, the leakage of the off PMOS transistor is high, thereby causing the effective BWCM current to be non-linear. To overcome the leakage, the PMOS of the branch to be disabled is turned off by tying its gate to 0.5 V V_{DDH} (available for digital operation [9]).

Current Controlled Oscillator

A CCO is starved by the current from the BWCM (I_{BWCM}). Due to a low headroom availability of 0.2 V, an NMOS-only CCO was implemented. The CCO output frequency is

dependent on I_{BWCM} and C_L. I_{BWCM} is linearly proportional to temperature and C_L is a Metal-Insulator-Metal capacitor of very small temperature variation. Therefore, the effective CCO frequency is primarily determined by current I_{BWCM}. The drive strength of the NMOS devices is controlled using process trimming bits. The BWCM bits and CCO process bits are set during the initial calibration phase.

Digital Block

The digital block has a programmable fixed and a variable counter synthesized using low leakage high-V_T logic as shown in Figure 4.4. The fixed counter asserts a Done signal after counting a preset number of reference clock (e.g. system clock) cycles. This gives a fixed time window for temperature sampling and conversion into digital code. The variable counter counts the CCO cycles until Done is asserted high and outputs a code that is representative of the CCO frequency and therefore the ambient temperature. Prior to the next temperature fetch, both the counters are reset.

4.2.2 Results

PTAT Current Circuit

The linearity of the PTAT current is denoted by R^2 (a measure of goodness of fit in linear regressions). The linearity histogram for a 100-point Monte Carlo simulation is shown in Figure 4.5a. The mean R^2 is 0.9993 and the 3σ variation is 0.0024.

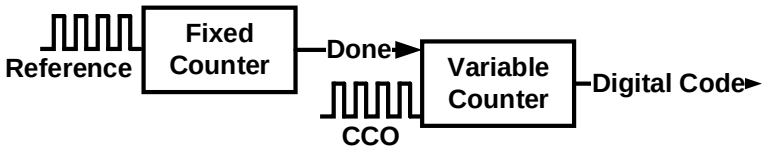


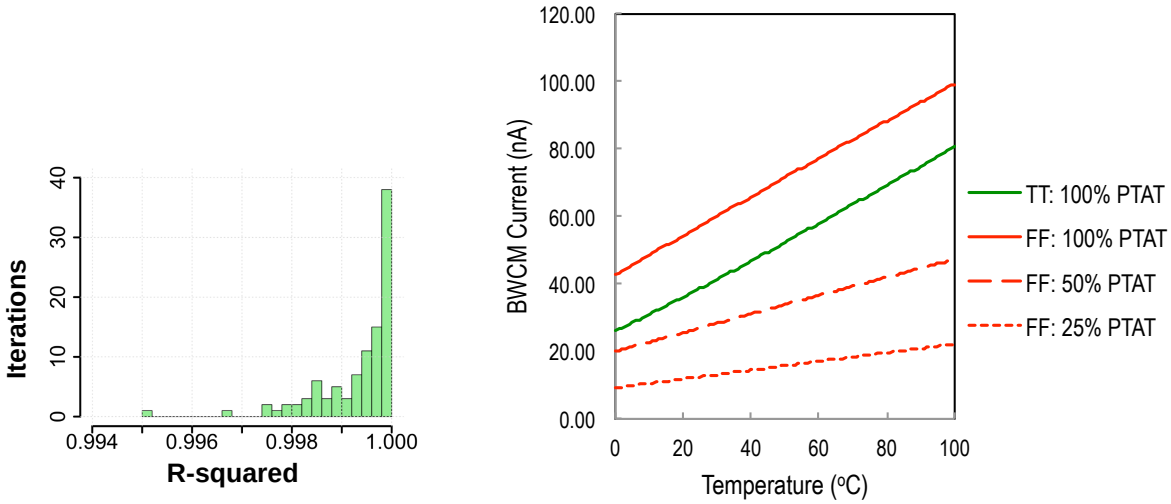
Figure 4.4: Digital block consisting of two counters to convert temperature to digital code.

Bit-weighted Current Mirror

The PTAT current is mirrored to starve the CCO using a BWCM. Figure 4.5b demonstrates the advantage of using a BWCM. In fast process corners such as FF, the PTAT current is higher than desired and when mirrored to subsequent CCO stages this causes higher power consumption than desired. Figure 4.5b shows the cases when 100% of the PTAT current is mirrored. For fast process corners, only a fraction of the PTAT current is required to be mirrored to control the power consumption of the temperature sensor system. The plot also shows the scaled BWCM current (50% and 25% PTAT) for different bit-configurations of B<0:7>for fast process corners.

Inaccuracy, Power Supply Sensitivity, and Power

Figure 4.6 shows the histogram of the temperature sensor inaccuracy of the 15 data points taken from the 100 Monte Carlo simulation points (determined after processing trimming of the CCO delay elements). The mean inaccuracy is +1.0/-1.2^{o}C and maximum inaccuracy is +1.5/-1.7^{o}C. The frequency range of the CCO gives a resolution of 0.008 ^{o}C/LSB, although



(a) (b)

Figure 4.5: (a) Linearity (R^2) histogram. (b) Different fractions of BWCM current in faster process corners saves power.

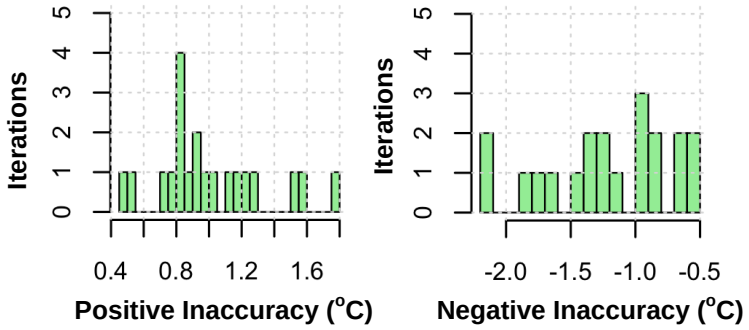


Figure 4.6: Inaccuracy histogram (15 points) after trimming. The mean inaccuracy is +1.0/-1.2 ^{o}C, the maximum inaccuracy is +1.5/-1.7 ^{o}C.

the thermal noise floor limits the practical resolution. Programmable counters enable different sampling windows for temperature conversion making resolution-power trade-off possible.

A variation of power supply because of variation of reference voltage due to temperature will add additional inaccuracy in the temperature sensor. Such a supply noise variation was simulated to be 0.032^{o}C/mV. It can be improved using decoupling capacitors (∼20 pF of area ∼50x50 μm^2). To effectively eliminate the effect of supply noise that is 1 kHz or higher, a low-pass filter (off-chip 50 kΩ and 10 nF) can be used. The sensor presents a low load, so it can be operated from a well-controlled supply using an LDO, too.

To measure the power consumption, the BWCM bit value for the typical process corner was set to 1000 0000. For faster process corners the bits were set to scale the current accordingly while maintaining high linearity of the mirrored current. The average power consumption of the analog core at 0.2 V is 18 nW. The total system power (including the locking circuit, level shifter [74] and digital block at 0.5 V) is 23 nW. A lower sampling rate further saves power as the digital blocks consume only 190 pW of leakage between samples.

Comparison with Prior-Art

Table 4.1 compares the proposed temperature sensor design with other recent low-power designs of varying circuit topologies with sub-μW power consumption. This work focuses

Table 4.1: Comparison with prior-art temperature sensors

Sensor	Tech. (μm)	V_{DD} (V)	Fully-Integrated	Inaccuracy (^{o}C)	Temp (^{o}C)	Power (nW)	Energy/Conversion (nJ)
This work	0.13 CMOS bulk	0.2,0.5	Yes	+1.5/-1.7	0-100	23	0.23
[68]	0.16 CMOS bulk	0.85	No	+/-0.4(3σ)	-40-125	600	3.6
[69]	0.18 CMOS bulk	1	Yes	+3/-1.6	0-100	220	22
[70]	0.18 CMOS bulk	0.5,1	No	+1/-0.8	-10-30	120	3.6
[71]	0.18 CMOS bulk	1.2	Yes	+1.5 /-1.4	0-100	71	2.2
[75]	0.18 CMOS bulk	1.2	No	+1/-0.8	0-100	405	0.41

on lowering the power consumption while maintaining similar inaccuracy and temperature range compared to prior-art. Compared to a recent ULP temperature sensor [71] designed in the same technology, the proposed temperature sensor system has 3x lower power and comparable inaccuracy. The area of the analog core is 150x100 μm^2 and the total system area is 250x250 μm^2.

We conclude the discussion on the ULP temperature sensor that senses the ambient temperature and aids tolerance against temperature variations. Next, we discuss the design of an ULP supply voltage monitor that aids tolerance to voltage variations.

4.3 Supply Voltage Monitor

Supply voltage monitoring is important in recent ULP IoT sensor systems to provide tolerance to supply voltage variations. Various types of supply voltage monitors sensors based on ring oscillators, ADCs, etc. have been traditionally designed for this purpose. In [65], the authors proposed on-die dynamic voltage monitoring and an adaptive clock distribution scheme to enable tolerance to power supply variations for HP processors. In [76], on-die sensors are employed to monitor high-frequency V_{DD} variations. Such droop detectors have traditionally consumed higher quiescent currents and therefore are not suitable for energy-constrained, sub-V_T IoT designs such as [13].

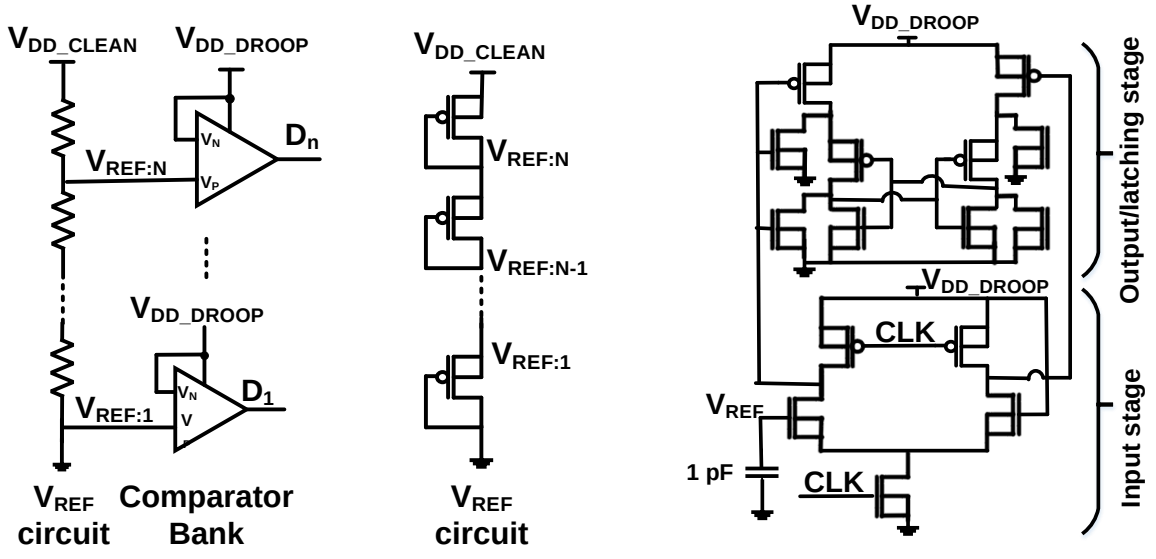
In this chapter, we design to further lower the power consumption of a voltage monitor design by operating the comparator core in the V_{DD} domain to be monitored compared to

the traditional super-V_T operation of circuits. The voltage monitor consumes very low power and energy and is operational down to 0.3 V V_{DD}.

4.3.1 Approach

A simplified block diagram of the proposed voltage monitor is shown in Figure 4.7a. It is a flash ADC style supply voltage monitor. The comparator bank compares V_{DD_DROOP}, which is the monitored supply voltage, with a set of reference voltages generated by a reference (V_{REF}) circuit.

The comparators operate at V_{DD_DROOP} and the V_{REF} circuit operates at V_{DD_CLEAN}, which is the clean supply voltage typically available on SoCs for analog modules [9]. The number of comparators in the bank and the V_{REF} voltages are predetermined during design depending on the IoT chip requirements. With the availability of many V_{REF} voltages, the voltage monitor can be designed to operate at a wide V_{DD_DROOP} range (0.3 V to 1



(a) (b) (c)

Figure 4.7: (a) A flash-ADC style voltage monitor (b) A diode-connected V_{REF} circuit. (c) A comparator design capable of low-V_{DD} operation.

V). It outputs a thermometer code (D_n:1) that can be directly used in an SoC. A detailed description of these components is presented in the next section.

V_{REF} Circuit

The V_{REF} circuit consists of stacked diode-connected transistors of equal sizes operating at V_{DD_CLEAN} (1 V) as shown in Figure 4.7b. It is a voltage divider circuit that gives many reference voltages ($V_{REF(N:1)}$). The use of many transistors of long length (L) mitigates process mismatch and temperature effects as observed in simulations. We use 50 stacked transistors (L=5 μ) for this reason and for a 20 mV resolution. The transistors operate in sub-V_T and consume low static power, which leads to a minimal impact on V_{DD_CLEAN} and makes the circuit suitable for an ULP voltage monitor. The target bias current is a few nAs to lower the impact of switching noise from the comparator bank. In [63], a configurable V_{REF} was generated by multiplexing high-impedance nodes of a diode-connected divider. However, such nodes cannot sufficiently drive a multiplexer and this can cause deviation from ideal divider characteristics. The V_{REF} variation in [63] is 64 mV. To avoid this, we implement a non-configurable V_{REF} circuit.

Comparator Circuit

The clocked comparator [77] is based on the double-tail architecture [78] as shown in Figure 4.7c. The clock (CLK) can be derived from the system clock (tens of kHz) of the IoT chips [9] to sufficiently monitor low-frequency (kHz) V_{DD} variation. When CLK is low, it is in the reset mode and when CLK is high, it latches the differential output voltage depending on the inputs. This design requires only a single-phase clock, unlike [78]. Less stacking in the input stage and use of regular-V_T devices enable low-V_{DD} operation. It has better offset [77][79] than the conventional design [78], which makes it suitable for low-V_{DD} operation. Isolation of the input stage from the latching stage lowers the kick-back noise to the high-impedance V_{REF} circuit. The comparator operates at V_{DD_DROOP}, unlike a traditional fixed-bias (V_{DD_CLEAN})

operation. This avoids switching transients on V_{DD_CLEAN} and achieves lower power at lower V_{DD_DROOP}. Clock gating enables further power savings. A metal-insulator-metal (MIM) capacitor (C_{REF}=1 pF) at its V_{REF} node reduces switching noise. It is created over the comparator devices to avoid area overheads.

4.3.2 Results

The number of comparators in the bank depends on SoC needs. We quote the metrics of the voltage monitor with 2 comparators for similarity with traditional window-detection designs with 2 comparators [63][80]. Although the monitor architecture is capable of wide range of operation, we discuss sub-V_T, low-V_{DD_DROOP} monitoring.

V_{REF} Circuit

Figure 4.8a shows the results of a 250-pt Monte Carlo simulation for the V_{REF} nodes (0.5, 0.4, 0.3 V) of the V_{REF} circuit. The worst V_{REF} variation (across 0-100^{o}C), which occurs at the middle of the stack, is 4 mV. This is an improvement over the 64 mV variation quoted in [63]. From simulations, we also observed that its static current can be in fA range such as in [63]. However, we size the transistors to have a few nAs of current across all process corners to lower the switching noise at the high-impedance V_{REF} nodes from the comparator. The average measured power consumption of the V_{REF} circuit is 4 nW. Higher bias current, a higher number of transistors with a long L, and avoiding multiplexing the high-impedance V_{REF} nodes contribute to low V_{REF} variations. The V_{REF} circuit area is 4200 μm^2.

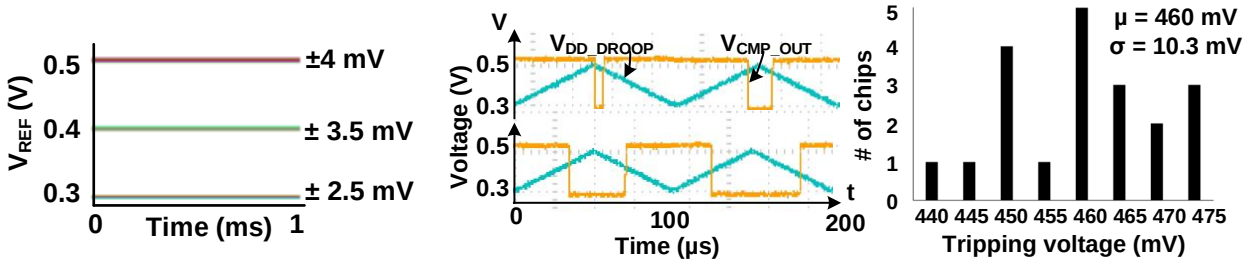
Comparator circuit

Figure 4.8b shows the measured standalone comparator output at a 0.5 V V_{DD_DROOP} for a sweep of decreasing V_{REF}. Figure 4.8c shows the measured tripping point of a comparator operating at a nominal V_{DD_DROOP} of 0.5 V with a V_{REF} of 0.46 V from the V_{REF} circuit, across 20 chips. The average tripping point (μ) is 0.46 V with a standard deviation (σ) of

10.3 mV, which is indicative of the process variation and mismatch effects on both the V_{REF} circuit and the comparators. The voltage monitor can also be implemented with a calibration technique [77] depending on the accuracy requirement. The area of each comparator and C_{REF} is 990 μm^2.

Power and Area

Figure 4.9 is a measured power plot of a comparator across 5 chips showing lower power at lower V_{DD_DROOP} and CLK frequencies. The total measured average power of the voltage monitor with 2 comparators at 0.5 V V_{DD_DROOP} and the V_{REF} circuit at 1 V V_{DD_CLEAN} at a 100 kHz CLK is 28 nW. The voltage monitor area is 6180 μm^2.



(a) (b) (c)

Figure 4.8: (a) Monte Carlo simulation plot shows low variation in V_{REF} nodes 0.5 V, 0.4 V, and 0.3 V. (b) Measured comparator output (V_{DD_DROOP} = 0.5 V) for different V_{REF}. (c) Comparator tripping voltage variation across 20 chips.

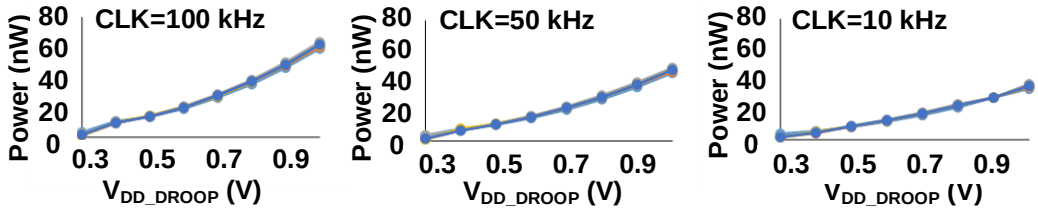


Figure 4.9: Measured power of the comparator across 5 chips showing lower power at lower V_{DD_DROOP} and CLK frequencies.

Comparison with Prior-Art

In Table 4.2, we compare the proposed design with prior V_{DD} monitors/droop detectors. This design can monitor up to 10 kHz frequency V_{DD_DROOP} variations at 0.5 V. It consumes lower power than [63]. Most of the area is incurred by the robust design choices of the V_{REF} circuit. This work can also be extended to a configurable design like [63] with lesser V_{REF} variations, by using switches to vary the number of active transistors in the V_{REF} circuit to generate variable V_{REF} without a multiplexer. The droop to be measured is predetermined during design. For a nominal 0.5 V V_{DD_DROOP}, the comparators are operational up to 0.3 V and so the droop is quoted as 200 mV. It is capable of wider V_{DD} operation than the oscillator-based design [81] and capable of low-V_{DD} monitoring compared to [80] used in energy-harvesting circuits.

We conclude the discussion on the design of an ULP voltage monitor circuit. Finally, in the next section, we discuss the design of circuits for the post-silicon hold time closure scheme introduced in Chapter 3.

4.4 Circuits for Post-Silicon Timing Closure

Hold time closure in the presence of PVT variations is traditionally achieved by allocating extra design margins. This translates to lowering clock-skew (t_{skew}) by clock-tree optimization and increasing data-path delay (t_{logic}) by traditional-buffer insertion. However, an underestimation of hold margins may lead to circuit failure and overestimation increases power and

Table 4.2: Comparison with prior-art voltage monitors

Sensor	Tech. (μm)	V_{DD} (V)	Droop (mV)	Area (μm^2)	Power (nW)	Type/Freq
This work	0.13 CMOS bulk	1-0.3	200	6180	28 (0.5 V)	Analog/low
[63]	0.065 CMOS bulk	1.2	tunable	191	50	Analog/low
[80]	0.180 CMOS bulk	1.8	300	N/A	7.2	Analog/low
[81]	0.130 CMOS bulk	0.5-0.8	44-170	7100	900(0.75V)	Digital/low
[76]	0.090 CMOS bulk	1.8-0.7	270(1V)	N/A	N/A	Analog/high
[65]	0.016 FinFET	0.95-0.7	90(0.9V)	2590	2.5M(0.9V)	Digital/high

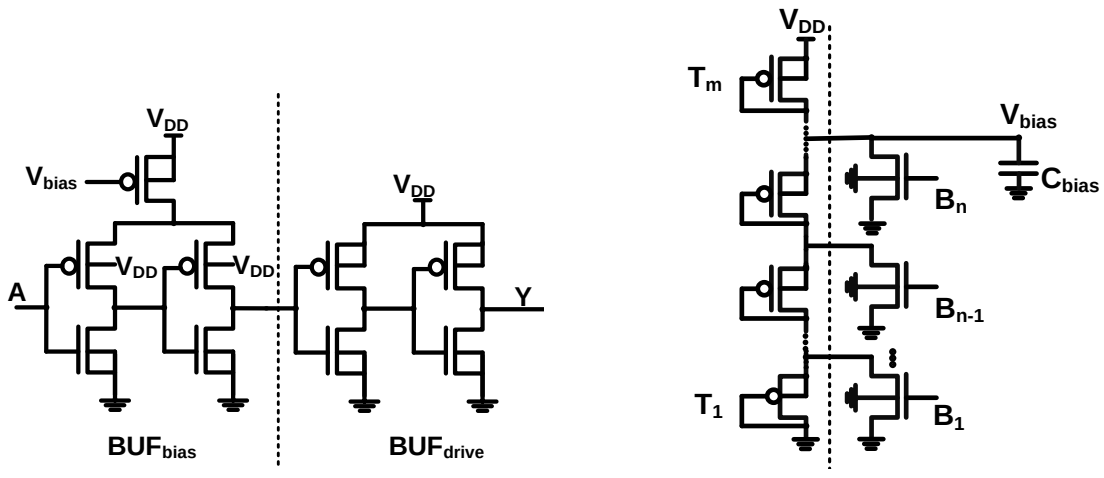
area. This necessitates a post-silicon knob for hold time closure in sub-V_T designs as described in Chapter 2. We propose a post-silicon hold time closure technique for low-performance and energy-efficient flip-flop-based designs using tunable-buffers in the data-path instead of traditional-buffers. Compared to traditional-buffer insertion, the tunable-buffer technique mitigates the effort for hold margin estimation by enabling post-silicon hold correction.

The two critical circuits required for this approach are: the tunable-buffer delay and the delay control mechanism.

4.4.1 Approach

Tunable-buffer Circuit

Many tunable-buffer structures have been previously explored for different applications. For instance, they use tunable output capacitances [82] or configurable switches or current mirrors at the inverter pull-down/pull-up [83], to generate variable delays. However, capacitances may not scale well with technology. Capacitances and digital switches also have high areas, which makes such tunable-buffers unsuitable for insertion in data-paths. Therefore, we implement tunable-buffers with an analog control as shown in Figure 4.10a.



(a) (b)

Figure 4.10: (a) Tunable-buffer structure. (b) Bias voltage generator.

The tunable-buffer in this work has two units, BUF_{bias} and BUF_{drive}. BUF_{bias} is current-starved using a PMOS header (biased with voltage V_{bias}) to provide a variable delay. BUF_{drive} operates at V_{DD} to maintain a sharp output. Its lowest delay (when $V_{bias} = 0$ V) is comparable to the delay of two traditional-buffers. The substrates of all the PMOS devices are tied to V_{DD}, which makes their insertion possible with physical design tools. An alternative tunable-buffer structure consists of only 2 inverters in series. The first inverter is current-starved using both a PMOS header and an NMOS footer, each biased using 2 separate voltages. The second inverter is tied to V_{DD}. Although this leads to a smaller area and the lowest delay comparable to one traditional-buffer, it involves the additional overhead of generation, control, and routing of 2 bias voltages. Therefore, we choose the tunable-buffer shown in Figure 4.10a with a single V_{bias}. Current-starved BUF_{bias} consumes lower power than a full-V_{DD} swing buffer. We present results in the next section from which we infer that the tunable-buffer power is always lower than a chain of traditional-buffers for the same delay, despite the short-circuit current due to the low-swing BUF_{bias}. This confirms its feasibility for our technique.

Bias Generator Circuit

The tunable-buffer requires a bias voltage (V_{bias}). Bias generators based on digital-to-analog converters, charge pumps, etc. have been previously explored [84][85] for applications such as fine-grained body-biasing [86]. A bias generator targeted toward energy-efficient systems needs to consume low power. Figure 4.10b shows the voltage-divider based V_{bias} generator.

A voltage divider is built using a stack of equally-sized diode-connected PMOS transistors ($T_{m:1}$), with their bulks tied to their sources for similar bias. V_{bias} is tapped at the center of the stack. On-chip capacitor, C_{bias}, mitigates switching noise from the tunable-buffers. $T_{m:1}$ operate at sub-V_T, which enables an ultra-low static power. The target bias current is a few nAs to lower switching noise at V_{bias}. In our design, m = 20 and V_{DD} = 0.5 V. Therefore, when $T_{20:1}$ are all active in the divider circuit, V_{bias} = 0.25 V. A control logic of NMOS switches with a thermometer code-input $B_{n:1}$ (n=10 in this paper) ties different nodes

of the divider to ground (V_{SS}). This is to vary the active number of transistors and generate different V_{bias}.

The design decisions for the bias-generator: m, n, and the nodes of the divider at which NMOS switches are placed, are governed by the required range and granularity of the tunable-buffer delay. The tunable-buffer range is its maximum delay, and granularity is its different delay steps. Whereas in traditional synthesis flow, additional hold margin leads to a longer buffer chain, it translates to a wider tunable-buffer range in the proposed technique. In low-performance SoCs, the tunable-buffer delay range is more critical than its granularity.

4.4.2 Results

Tunable-buffer Circuit

We allow the maximum tunable-buffer range (V_{bias} = 0.25 V). This range is equivalent to the delay of 20 traditional-buffers at a typical corner (TT:27°C). This high additional margin enables failure-free operation at extreme PVT conditions. At $V_{bias} > 0.25$ V, the tunable-buffer starts approaching its limit of reliable operation.

Figure 4.11 shows the simulated power of the tunable-buffer for different V_{bias}, compared to a traditional-buffer chain of same delay in both 28 nm fully depleted silicon on insulator (FDSOI) and 130 nm bulk CMOS nodes, at sub-V_T (0.4 V, 100 kHz). The power of the tunable-buffer is lower compared to the chain of traditional-buffers of equivalent delay (for e.g., 2.61X lower in 130 nm for V_{bias} = 0.167 V) for values of V_{bias} up to 0.25 V, which we decide to be the maximum V_{bias} for reliable operation. This allows the use of our tunable-buffer without power overheads.

Bias Generator Circuit

A granularity of 10 (n=10) is chosen as a sufficient coarse-grained tunability for different PVT conditions. The tunable-buffer is non-linear owing to its current-starved nature and non-linear V_{bias} generation, but this is not critical in low-performance designs.

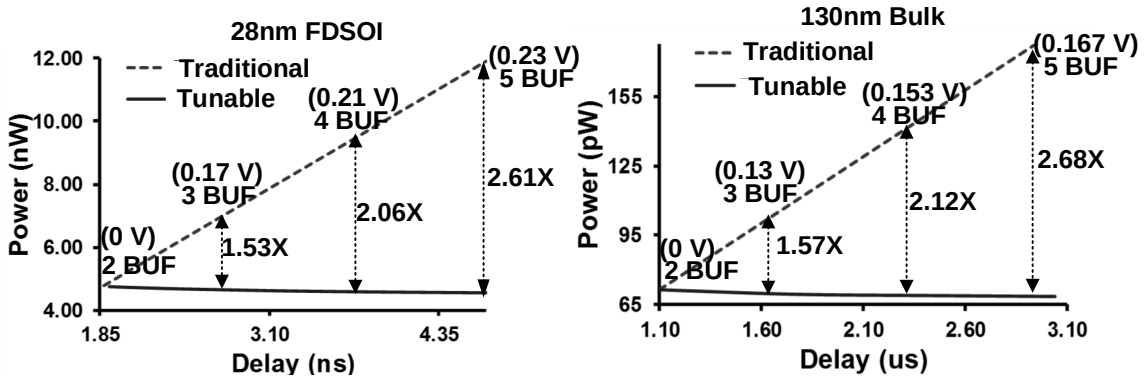


Figure 4.11: Tunable-buffer is lower power than the traditional-buffer chain of same delay.

A control logic of NMOS switches with a thermometer code-input $B_{n:1}$ ties different nodes of the divider to ground (V_{SS}) to vary the active number of transistors and generate different V_{bias} as shown in Table 4.3.

The V_{bias} generator has an average measured power of only 4 nW at 0.5 V.

4.5 Conclusion

In this chapter, we discuss the design of circuits that aid tolerance to PVT variations in sub-V_T, ULP IoT SoCs. In the ULP sub-V_T temperature sensor design, the core is operational down to 0.2 V. Together with the digital logic at 0.5 V, the total system power consumption is 23 nW, and the maximum inaccuracy is +1.5/-1.7 ^{o}C in the temperature range of 0 ^{o}C to 100 ^{o}C with a 2-point calibration. A BWCM architecture resists process-induced power variations, process trimming bits in the current-controlled oscillator control its drive strength, and a programmable digital control based on counters supports a resolution-power trade-off for IoT applications with different sampling rate and energy needs.

Table 4.3: Thermometer code $B_{10:1}$ vs. V_{bias}

$B_{10:1}$	V_{bias} (mV)	$B_{10:1}$	V_{bias} (mV)	$B_{10:1}$	V_{bias} (mV)
1111111111	0	1111110000	143	1100000000	222
1111111110	45	1111100000	167	1000000000	237
1111111100	83	1111000000	188	0000000000	250
1111111000	115	1110000000	206	-	-

The ULP voltage monitor design monitors low-frequency ripple and voltage variations in ULP IoT SoCs. It can be designed to operate under a wide V_{DD} range. The power consumption is only 28 nW at 0.5 V for a 2 comparator monitor design. The comparators operate at the monitored voltage to enable lower power at lower voltages. Its area and power can be traded-off for different V_{DD} monitoring needs of SoCs.

Finally, we discuss circuit techniques for the variation tolerant post-silicon hold time closure technique. We design the tunable-buffer circuit that is a good candidate for this solution. Despite the possibility of short circuit current in the tunable-buffer standard cell, it is still ULP and consumes lower power than a chain of traditional-buffers of equivalent delay. We also design an ULP bias generator circuit to control the delay of the tunable-buffer.

Chapter 5

Methodology and Tool-Flow for Variation Tolerant Digital Design

5.1 Background

[1]Chapter 3 introduces a novel variation tolerant technique for flip-flop based digital designs targeting performance-relaxed, energy-efficient, and ULP IoT SoCs. As mentioned in the introduction, we categorize the different steps involved in this technique across Chapter 3, 4, and 5 to fit into the flow presented in this thesis (modeling and analysis, circuit design, and tool-flow). To summarize this technique, tunable buffers are inserted in the data-path instead of traditional-buffers for hold time closure as shown in Figure 3.20. These tunable-buffers are controlled by an analog voltage from a bias generator, which enables a wide range of delay. Compared to traditional-buffer insertion, the tunable-buffer technique mitigates the effort for hold margin estimation by enabling post-silicon hold correction. The tunable-buffer has a wide delay range and hence can reduce the need for long chains of traditional-buffers in the data-path. In Chapter 3, we analyze the above technique for a first-order estimation of its benefits and costs. Chapter 4 presents the design of circuits that are crucial to making the

[1]This chapter derives content from [DAK11]

above technique feasible. The circuits include a tunable-buffer circuit and a bias generator circuit. Finally, in this chapter, we present the physical implementation of the above circuits and the design methodology or strategy for inserting the tunable-buffers in hold critical data paths. We also present the tool-flow for automating the above technique and timing sign-off.

5.2 Approach

First, we discuss the physical implementation of the circuits to making post-silicon hold time closure feasible, namely the tunable-buffer and the bias generator.

Tunable-Buffer and Bias Generator Implementation

The tunable-buffer implementation must be such that it fits into a tool-flow for automatic digital design with standard tools. For that reason, we implement the tunable-buffer as shown in Figure 5.1 (the layout depicted is in a 130 nm CMOS bulk technology). The overhead introduced by the tunable-buffer compared to a conventional delay cell, which consists of two back-to-back buffers, is a PMOS header for delay tunability and a V_{bias} port. The PMOS header that is controlled by V_{bias} is added between the two buffers as shown in Figure 5.1. Metal layers M1 through M3 are typically used for routing between standard cells at the block level. Therefore, we reserve the next available horizontal layer, M5, for routing V_{bias} above the V_{DD} track. The height of the tunable-buffer is the same as that of all the standard cells and its overall area is only 1.27X higher than 2 back-to-back traditional-buffers due to the PMOS header. However, a single tunable-buffer can mitigate the need for a chain of traditional-buffers in hold-critical paths. Therefore, in hold-critical data-paths containing long chains of traditional-buffers (>2), this technique can potentially enable area savings.

The bias generator design is not a digital block and is, therefore, not automatically placed in the tool-flow. Its area is 0.0022 mm^2 (only ∼1% of a 130 nm OMSP core) and we use a single V_{bias} to tune many tunable-buffers as discussed in the next section. The bias generator

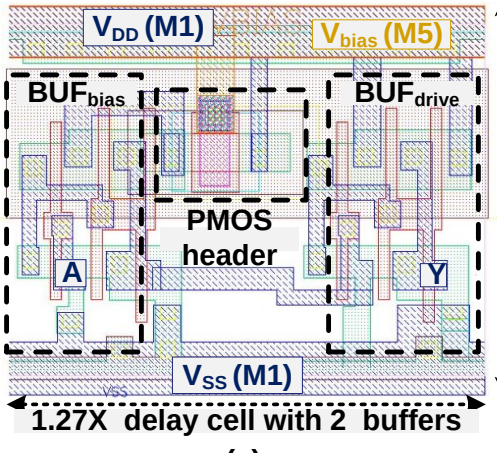


Figure 5.1: Tunable-buffer implementation.

is custom-placed and routed to the V_{bias} of the digital blocks. We use a MIM capacitor for C_{bias} (a 10 pF capacitor is of area 0.002 mm^2), which can mitigate the noise from 1k simultaneously switching tunable-buffers at 100 kHz. A MIM cap can be placed over the digital area to avoid area overhead.

Next, we describe the tunable-buffer insertion methodology or strategy.

Tunable-buffer Insertion Methodology

In traditional-buffer insertion, hold-critical paths (min-paths) with lower t_{logic} (logic path delay) are inserted with a higher number of traditional-buffers than hold-critical paths with higher t_{logic}.

Instead of traditional-buffers, we propose to insert tunable-buffers in the hold-critical paths (min-paths) in low-performance IoT SoCs. They are all tuned with the same V_{bias} despite different data-path delays (t_{logic}). Figure 5.2 illustrates an example scenario. The tunable-buffer delay is tuned with a certain V_{bias} to meet hold time in the most-critical paths (e.g. Bin A). The same delay (due to same V_{bias}) also resolves the hold issues in the less critical paths (e.g. Bin C), which gain extra hold slack (t_{extra}). Despite t_{extra}, the data-path delay remains within the t_{logic} limit to meet setup time. For instance, in an OMSP core designed to operate at 32 kHz in sub-V_T (0.5 V V_{DD}), the setup slack is ∼45% of the cycle

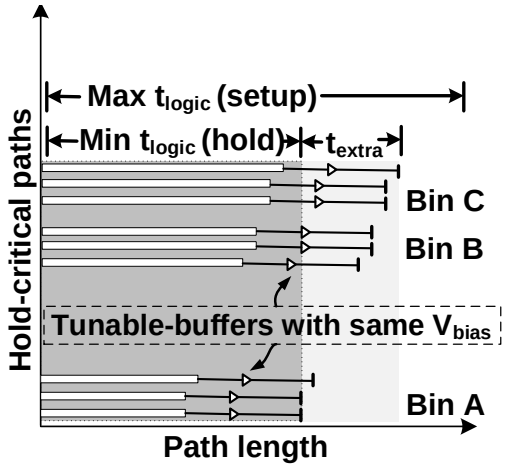


Figure 5.2: Illustration of tunable-buffer insertion strategy in low-performance SoCs.

time (14 μs). Extra hold slack (t_{extra}) of tens of ns (in 130 nm) will not cause failure with above setup slack. The static timing analysis (STA) is presented in Section 5.3. Therefore, tuning all the buffers in unison is acceptable as illustrated in Figure 5.2.

Finally, we describe the tool-flow for automatic insertion of tunable-buffers using standard commercial tools.

Tool Flow for Implementation

First, the RTL for a design is synthesized as done traditionally. The tunable-buffers are then inserted during the below modified physical design tool-flow for hold time closure. The steps involved in the physical design flow are shown in Figure 5.3(a):

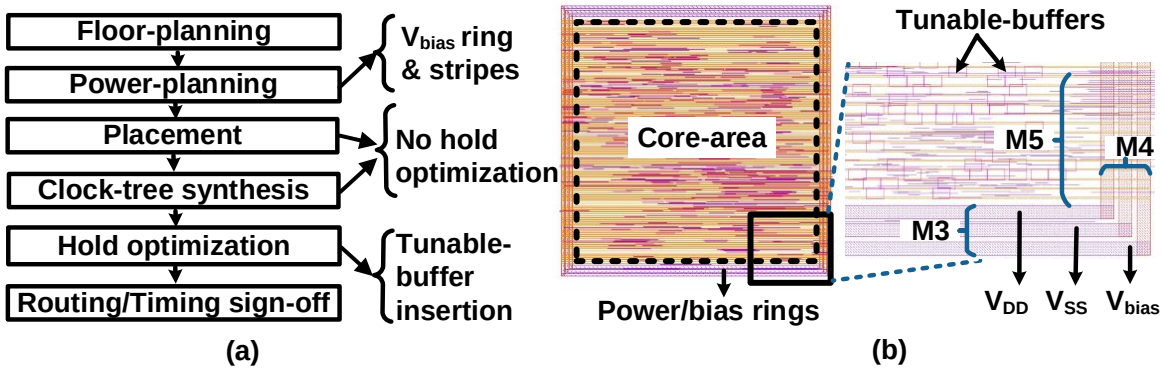


Figure 5.3: (a) Flow-chart of physical implementation using tunable-buffers. (b) Layout of an OMSP design with tunable-buffers.

(a) *Floor-planning*: We decide the design dimensions here.

(b) *Power-planning*: This step involves creating V_{DD} and V_{SS} rings using metal M3 and M4, and adding metal M1 stripes over standard-cell rows to pre-connect V_{DD} and V_{SS}. A V_{bias} ring using metal M3 and M4 is also created, and metal M5 stripes are added to pre-connect the tunable-buffers to V_{bias}. M5 is reserved for V_{bias} (Figure 5.1(a)) and this indeed creates M5 blockage within the block. However, this can be optimized in the future by placing M5 stripes only over tunable-buffer rows.

(c) *Placement*: This step involves standard-cell placement and their location, size, and design rule violations (DRV) optimization. Hold time is not optimized in this step.

(d) *Clock-tree synthesis*: This step involves balancing a clock-tree or minimizing clock skew by inserting buffers or inverters in the clock-paths.

(e) *Hold optimization:* This is the step that is critical to the modified physical design tool-flow. We discussed in Chapter 3, how we need to freeze the tunable-buffer to have a specific delay during its insertion. In the tunable-buffer structure described in the thesis, its delay is controlled by V_{bias}, and we need it to be tuned to a particular value of V_{bias} ($V_{bias\_insertion}$) during insertion in data-paths. For this, we generate timing information (.lib) for the tunable-buffer corresponding to different V_{bias} using a characterization tool. A preliminary target hold margin input to the physical design tool identifies the hold-critical paths. We restrict the hold-buffers to only the tunable-buffer.

Depending on the desired amount of post-silicon tunability, the tunable-buffer .lib corresponding to different V_{bias} can be used in the flow. For maximum tunability, the .lib generated at $V_{bias} = 0$ can be used. However, it can have no impact on the mitigation of the traditional-buffer chains (no buffer savings) and potentially cause area and power overhead. In this chapter, we use the tunable-buffer .lib generated at $V_{bias} = 0.125$ V in the tool-flow, to demonstrate a trade-off of post-silicon tunability and the number of tunable-buffers. We verify this implementation in the Results section. To iterate, the biggest advantage of this technique lies in the fact that, in case of underestimation in the preliminary target hold

margin, the tunable-buffers can be tuned post-silicon for hold correction. Figure 5.3(b) shows the layout of an OMSP design implementation with tunable-buffers in the data-path.

(f) *Routing and timing sign-off*: After routing, STA is performed using the tunable-buffer .lib at different V_{bias} as discussed in Section 5.3. Finally, the bias generator is custom-placed and routed to the V_{bias} ring. V_{bias} is shared between all the tunable-buffers in a block and is, therefore set to solve the worst-case hold violations and yet not affect the setup timing requirements in performance-relaxed IoT SoCs.

5.3 Results

Timing Simulations

An STA tool is used for timing simulations. Figure 5.4 shows the hold slack histograms in a fast-fast corner (FF:25^{o}C) with aggressive on-chip variation (OCV), for a 32-bit/16-stage shift-register (SR) designed using the proposed technique, across different V_{bias}. Negative hold slack means timing failure. The number of paths failing hold time decrease as V_{bias} is increased from 0 V to 0.2 V (Figure 5.4(a)-(c)). This demonstrates hold correction with V_{bias} tuning at a worst-case PVT condition. V_{bias} = 0.25 V is the reliable limit for our tunable-buffer.

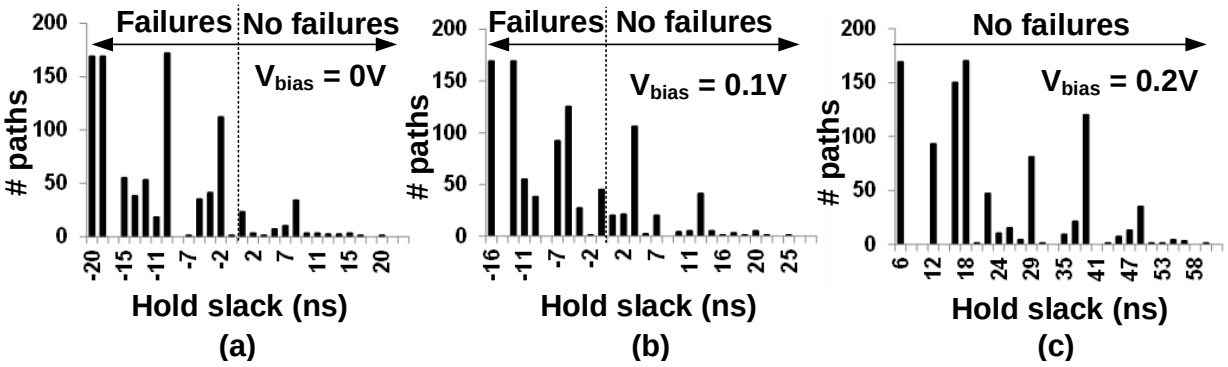


Figure 5.4: The number of paths vs. hold slack at FF:25^{o}C from STA simulation: Negative slack (failures) at $V_{bias} = 0$ is made positive by increasing V_{bias}.

Table 5.1 summarizes the hold-related benefits of 3 blocks (32-bit/16-stage SR, 8-bit/4-tap FIR, 2-channel DMA) designed with the proposed technique to operate at 0.5 V V_{DD} and 32 kHz frequency. Post-silicon tunability enables hold-slack improvement, which is maximum at V_{bias} = 0.25 V. We compare the maximum hold-slack improvement for these blocks (at FF:25^{o}C with aggressive OCV) with their traditionally-designed counterparts. We designed the baseline traditional blocks for ∼25 ns of hold-slack at this corner.

The SR and FIR blocks have different natures of hold-critical paths as discussed in Chapter 3. The SR sees a 195% increase in hold slack availability and a setup-critical FIR sees up to a 103% increase. There can be added benefits in blocks with hold-critical paths needing long chains of tunable-buffers (29% for SR and 18% in DMA). We observe only a small buffer overhead for setup-critical blocks such as FIR for a 103% increase in hold slack. This is because of the presence of many long paths in such blocks in which only 1 traditional-buffer is inserted using the traditional buffer-insertion flow. On the other hand, a tunable-buffer is effectively 2 back-to-back traditional buffers as described in Section 4.4.1. The # of tunable-buffers and subsequently hold-buffer savings or overheads are determined by the distribution of the hold-critical paths in the specific block as analyzed in Section 3.3.

Finally, we note that all the above blocks meet setup time also at V_{bias} = 0.25 V. The worst-case setup slack for the blocks with setup-critical paths (FIR and DMA) at a slow-slow corner with OCV is greater than ∼45% of the clock cycle with a much higher setup-slack for hold-critical SR (∼90%), which is ample for failure-free operation at 32 kHz.

Table 5.1: Hold-benefits in blocks with data-path tunable-buffers

Design	Max Hold-Slack Increase (% w.r.t traditional)	#Hold-Buffers (% w.r.t traditional)
SR	195%	-29%
FIR	103%	3%
DMA	152%	-18%

Experimental Setup and Measurements

We implemented the above SR and FIR blocks with tunable-buffers and a V_{bias} generator. To verify post-silicon hold correction, we also implemented an on-chip experimental setup as shown in Figure 5.5. This was necessary because hold failure occurrence is unpredictable due to variations and the typical corner characteristics of our chip. To overcome this challenge, we implemented a variable delay line in the clock-paths of a few hold-critical timing paths. We sweep this delay line to generate different values of t_{skew} and therefore, cause hold failures by emulation. We are able to correct these failures by increasing V_{bias} of the tunable-buffers by tuning bits $B_{10:1}$ of the V_{bias} generator. Therefore, with this setup, we verify post-silicon hold correction in both the SR and FIR.

With this experimental setup, we are also able to measure t_{skew} of a few hold critical hold paths in both the test blocks. Figure 5.6(a) demonstrates post-silicon hold correction in the SR across 5 chips. Up to t_{skew} of ∼100 ns, there are no hold failures in the hold-critical path and therefore, V_{bias} is tuned to 0 V. After that, as t_{skew} increases, hold failures start occurring. Each data point corresponds to the measured t_{skew} in a hold-critical path and the corresponding V_{bias} set to correct the subsequent hold failure. We demonstrated hold correction for up to 800 ns of t_{skew} (32X our STA sign-off hold slack of 25 ns at FF:25^{o}C), which proves the scheme's high potential for hold-slack improvement.

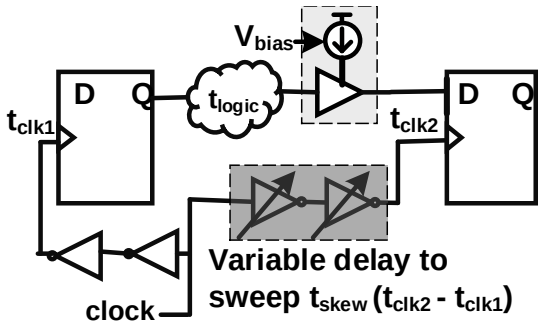


Figure 5.5: Experimental setup: variable delay in clock-path to increase t_{skew} and cause hold failure.

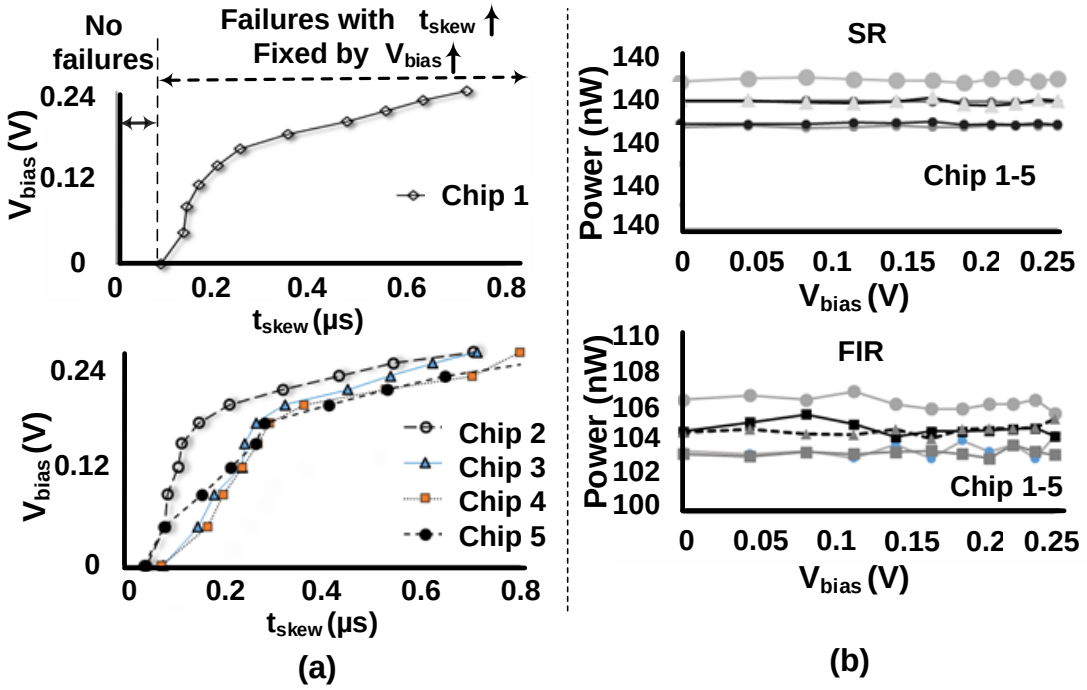


Figure 5.6: (a) Post-silicon hold correction: hold failure due to t_{skew} corrected by V_{bias} tuning. (b) Negligible (∼1%) variation of power with V_{bias} tuning.

Power and Area

Figure 5.6(b) shows the measured average power of the SR and FIR across 5 chips for different V_{bias}. The power across V_{bias} stays within ∼1% variation. The low magnitude of power variation can be attributed to the inherent variability during the measurement process. We infer that the impact of tunable-buffer short-circuit current is negligible. The V_{bias} generator has an average measured power of only 4 nW at 0.5 V.

We expect the area overhead from V_{bias} ring at block-level and routing at the chip-level to be similar to the fine-grained body-biasing techniques [87], which is about 2%. Owing to the ultra-low power and small area of the bias generator, it has the potential to be used as local bias generators to reduce V_{bias} routing overhead. The area and power overhead/savings from the tunable-buffers are design-dependent. For instance, we show only a small increase in the FIR, which may be compensated by savings from other blocks in an SoC. We conclude that post-silicon hold correction can be achieved with minimal area and power impact.

5.4 Conclusion

In the earlier chapters, we introduce a post-silicon hold time closure technique for sub-V_T designs and design circuits that aid this technique. This technique mitigates the effort for hold margin estimation by enabling post-silicon hold correction with minimal impact on area and power. In this chapter, we discuss the implementation styles of the tunable-buffer and the bias generator circuits. We also present the tunable-buffer insertion strategy for performance-relaxed IoT SoCs and a modified physical design tool-flow for tunable-buffer insertion using standard commercial tools. We demonstrate both silicon and simulation results that verify this post-silicon hold time closure scheme.

Chapter 6

Conclusions

In this chapter, we summarize the work in this thesis, discuss contributions, and provide insights to potential future work opportunities. In this thesis, we focus on two current high impact segments in the semiconductor industry: performance-relaxed ULP IoT SoCs and HP processors for compute-intensive applications.

IoT devices cater to a wide variety of applications, which call for deployment in remote areas. Therefore, energy-efficient and ULP circuit design techniques become necessary. In this thesis, we address a subset of opportunities for power reduction in such SoCs by focusing on efficient data-flow and clock-related circuits. Another critical issue that impedes widespread deployment of IoT SoCs is their high sensitivity to PVT variations. We address a subset of issues in this category to promote widespread ULP IoT device deployment in the future. Beyond the scope of IoT SoCs, we also focus on processors that enable high-performance computing and drastically impact human lifestyle. Power reduction and tolerance to supply noise (voltage variations) are critical requirements of this segment of ICs also, which we address in this thesis.

6.1 Project Contributors

The two IoT SoCs described in Chapter 1 [9] [10] were built a collaboration of all the co-authors listed in the publications. The DMA module design was my individual contribution, and I was responsible its design and implementation with guidance from Yousef Shaksheer. I and Christopher Lukas co-designed the GPIO interface for [10]. My other contributions for the IoT SoCs include the implementation of a XTAL oscillator circuit originally designed by Nathan Roberts at the University of Michigan and a JTAG module to aid the extensive testing of the IoT SoC in [10]. The clock source presented in Chapter 1 used the concept originally presented in [19]. My contributions include the implementation of a complete and flexible clock platform. The high-stability, temperature-compensated clock source was designed by Aatmesh Shrivastava. My individual contributions with guidance from Aatmesh Shrivastava include the design, implementation, and testing of the diode-transistor based, uncompensated clock source, a calibration scheme, and a digital controller to demonstrate multiple locking schemes.

The HP processor methodology presented in Chapter 3 was a collaboration between the University of Virginia and NVIDIA research. It was partially done during my 2015 summer internship at NVIDIA and was completed at UVA. The project was executed with guidance from Brucek Khailany and Matt Fojtik. The other contributors of the project were Sudhir Kudva, Yaping Zhang, and Prof. Calhoun.

The idea for post-silicon hold time closure methodology and design discussed in Chapters 3, 4, and 5 was mainly formulated and driven by me. Xinfei Guo helped with the tunable-buffer simulations, helped during brainstorming sessions, and during working toward a publication. Harsh Patel assisted with the tape-out and testing of the test-chip. Prof. Calhoun guided me through the process.

The temperature sensor and supply voltage monitor was primarily driven by me. Aatmesh Shrivastava guided the temperature sensor project and helped with the simulations. The base design for the comparator, which I tweaked for use in the supply voltage monitor,

was created by Abhishek Roy. Harsh Patel assisted with the tape-out of the supply voltage monitor.

6.2 Open Research Questions

With this section, we end the thesis with suggestions of open research questions that are related to the projects described here, for future work.

Chapter 2 (Design for ULP IoT SoCs)

We discussed the power impact of clock sources in ULP self-powered SoCs and the need for on-chip solutions to reduce system cost. Firstly, this platform can support any DCO solution that follows a certain architecture. This presents an opportunity for incorporation of other stable ULP designs into such a platform. Secondly, we present a temperature-drift prediction-based locking scheme. The system can be further improved by the use of a temperature sensor such as the one described in Chapter 4. Finally, we also primarily focused on a ring-oscillator based solution. However, relaxation-based oscillators are also increasingly gaining attraction for ULP SoCs and can potentially replace the temperature-compensated stable clock solution on this platform.

Chapter 3 (Modeling Methodologies for Variation Immunity in ICs)

We quantitatively evaluated the benefits of a fine-grained GALS adaptive clocking scheme. However, a similar quantitatively evaluation of overheads requires more than Verilog-A modeling. It is valuable to analyze an architectural model of a GPU-like system to analyze data-transfer overheads. Much work has been done toward this [51][52] and it is interesting to incorporate some of these models with our work. There are also many adaptive clock source designs, however, characterizing them for each chip is expensive in terms of cost and effort. Low-characterization adaptive clock solutions are very attractive for this reason.

Chapter 4 (Design for Variation Immunity in IoT SoCs)

Sub-V_T circuits are highly sensitive to PVT variations. However, analog, sub-V_T solutions require trimming or characterization bits for reliable operation across process corners. Reducing the characterization efforts required in the highly sensitive temperature sensor circuit is crucial for low-cost deployment. We implemented the supply voltage monitor circuit to track low-frequency V_{DD} variations and correct subsequent errors in SRAM circuits. In addition to aiding variation tolerance in circuits, an interesting opportunity lies in using the monitor information to improve the efficiency of the power management unit. For instance, we can allow the V_{DD} to vary or ripple as long as the circuits can tolerate it, which can enable higher efficiency is switched-capacitor DC-DC converters. Only when V_{DD} variation crosses a threshold, the converter needs to be switched and compromised on efficiency.

Chapter 5 (Variation Immune Digital Design Methodology and Implementation)

Although the design methodology and implementation for a post-silicon hold time closure technique are presented in this work, we control the post-silicon tuning bits manually for proof of concept. Therefore, an automatic tuning mechanism for the post-silicon control bits is a valuable addition to reduce the characterization time after fabrication. For this, implementation of error detection mechanisms or built-in self-test (BIST) schemes that can enable detection of functional violations due to hold time failures can potentially enable the automatic setting of the post-silicon tunability bits.

Research into the above open questions can potentially push the boundaries and mitigate limitations faced by ICs in achieving low power and variation tolerance.

Appendix A

Publications

A.1 Completed

[DAK1] **D. Akella**, et al.: "Modeling and Analysis of Power Supply Noise Tolerance with Fine-Grained GALS Adaptive Clocks" IEEE Asynchronous Circuits and Systems, 2016.

[DAK2] **D. Akella**, et al.: "A 0.2 V, 23 nW CMOS Temperature Sensor for Ultra-Low-Power IoT Applications" Invited paper to the special issue, selected papers from IEEE S3S Conference 2015, Journal of Low Power Electronics and Applications, 2016.

[DAK3] **D. Akella**, et al.: "A 23 nW CMOS Ultra-Low Power Temperature Sensor Operational from 0.2 V" IEEE SOI-3D Subthreshold Microelectronics Technology Unified Conference, 2015.

[DAK4] **D. Akella**, et al.: "A 36 nW, 7 ppm/^{o}C On-Chip Clock Source Platform for Near-Human-Body Temperature Applications" Journal of Low Power Electronics and Applications, 2016.

[DAK5] **D. Akella**, et al.: "A 28 nW CMOS Supply Voltage Monitor for Adaptive Ultra-Low Power IoT Chips" IEEE SOI-3D Subthreshold Microelectronics Technology Unified Conference, 2017.

[DAK6] A. Shrivastava, [et al. including **D. Akella**]: "A 1.5 nW, 32.768 kHz XTAL Oscillator Operational from a 0.3 V Supply" IEEE Journal of Solid-State Circuits, 2016.

[DAK7] F. Yahya, [et al. including **D. Akella**]: "A Battery-less 507 nW SoC with Integrated Platform Power Manager and SiP Interfaces" Symposia on VLSI Technology and Circuits, 2017.

[DAK8] A. Roy, [et al. including **D. Akella**]: "A 6.45 W Self-Powered SoC With Integrated Energy-Harvesting Power Management and ULP Asymmetric Radios for Portable Biomedical Systems" IEEE Transactions on Biomedical Circuits and Systems, 2015.

[DAK9] A. Klinefelter, [et al. including **D. Akella**]: "A 6.45 W Self-Powered IoT SoC with Integrated Energy-Harvesting Power Management and ULP Asymmetric Radios" IEEE International Solid-State Circuits Conference, 2015.

[DAK10] **D. Akella** et al.: " Enabling Post-Silicon Hold Time Closure by Tunable-Buffer Insertion" Work-in-Progress Poster at IEEE Design Automation Conference, 2017.

A.2 Planned

[DAK11] **D. Akella**, et al. "A Post-Silicon Hold Time Closure Technique using Data-Path Tunable-Buffers for Variation-Tolerance in Sub-threshold Designs", 2018 IEEE International Symposium on Quality Electronic Design.

[DAK12] H. Patel, [et al. including **D. Akella**]: "Adapting PVT Variations using a Digital Controller for Reliable and Energy Optimal IoT Applications ", 2018 IEEE Design Automation and Test in Europe.

Bibliography

[1] T. H. Teo et al. A 700 μW Wireless Sensor Node SoC for Continuous Real-Time Health Monitoring. *IEEE Journal of Solid-State Circuits*, 45(11):2292–2299, Nov 2010.

[2] Y. H. Tu et al. A body sensor node SoC for ECG/EMG applications with compressed sensing and wireless powering. In *2017 International Symposium on VLSI Design, Automation and Test*, pages 1–4, April 2017.

[3] F. Zhang et al. A batteryless 19 μW MICS/ISM-band energy harvesting body area sensor node SoC. In *2012 IEEE International Solid-State Circuits Conference*, pages 298–300, Feb 2012.

[4] H. Bhamra et al. A 24 μW, Batteryless, Crystal-free, Multinode Synchronized SoC Bionode for Wireless Prosthesis Control. *IEEE Journal of Solid-State Circuits*, 50(11):2714–2727, Nov 2015.

[5] C. M. Nguyen et al. Wireless sensor nodes for environmental monitoring in Internet of Things. In *2015 IEEE MTT-S International Microwave Symposium*, pages 1–4, May 2015.

[6] K. I. K. Wang et al. Miniaturized wireless sensor node for earthquake monitoring applications. In *7th IEEE International Symposium on Industrial Embedded Systems*, pages 323–326, June 2012.

[7] M. Garland et al. Parallel Computing Experiences with CUDA. *IEEE Micro*, 28(4):13–27, July 2008.

[8] M. Daga et al. Towards accelerating molecular modeling via multi-scale approximation on a GPU. In *2011 IEEE 1st International Conference on Computational Advances in Bio and Medical Sciences*, pages 75–80, Feb 2011.

[9] A. Klinefelter et al. A 6.45 μW self-powered IoT SoC with integrated energy-harvesting power management and ULP asymmetric radios. In *IEEE International Solid-State Circuits Conference*, pages 1–3, Feb 2015.

[10] F. Yahya et al. A Self-Powered 539 nW SoC with Integrated Platform Power Manager and SiP Interfaces. In *IEEE VLSI Symposium*, Jun 2017.

[11] A. Shrivastava et al. A 32 nW bandgap reference voltage operational from 0.5 V supply for ultra-low power systems. In *2015 IEEE International Solid-State Circuits Conference*, pages 1–3, Feb 2015.

[12] F. Zhang et al. Design of a 300 mV 2.4 GHz Receiver Using Transformer-Coupled Techniques. *IEEE Journal of Solid-State Circuits*, 48(12):3190–3205, Dec 2013.

[13] A. Roy et al. A 6.45 μW Self-Powered SoC With Integrated Energy-Harvesting Power Management and ULP Asymmetric Radios for Portable Biomedical Systems. *IEEE Transactions on Biomedical Circuits and Systems*, 9(6):862–874, Dec 2015.

[14] B. H. Calhoun et al. Modeling and sizing for minimum energy operation in subthreshold circuits. *IEEE Journal of Solid-State Circuits*, 40(9):1778–1786, Sept 2005.

[15] OpenCores. OpenMSP430 Overview http://opencores.org/project,openmsp430. 2017.

[16] Y. Zhang et al. A Batteryless 19 μW MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications. *IEEE Journal of Solid-State Circuits*, 48(1):199–213, Jan 2013.

[17] D. Yoon et al. A 5.58 nW 32.768 kHz DLL-assisted XO for real-time clocks in wireless sensing applications. In *2012 IEEE International Solid-State Circuits Conference*, pages 366–368, Feb 2012.

[18] A. Shrivastava et al. A 1.5 nW, 32.768 kHz XTAL Oscillator Operational From a 0.3 V Supply. *IEEE Journal of Solid-State Circuits*, 51(3):686–696, March 2016.

[19] A. Shrivastava and B. H. Calhoun. A 150 nW, 5 ppm/^{o}C, 100 kHz on-Chip clock source for ultra low power SoCs. In *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference*, pages 1–4, Sept 2012.

[20] K. Ueno. CMOS voltage and current reference circuits consisting of subthreshold MOSFET. *Solid State Circuits Technologies; Intech Press: Rijeka, Croatia*, 2010.

[21] D. W. Jee et al. Digitally Controlled Leakage-Based Oscillator and Fast Relocking MDLL for Ultra Low Power Sensor Platform. *IEEE Journal of Solid-State Circuits*, 50(5):1263–1274, May 2015.

[22] W. H. Chen. A 0.5 V, 440 μW frequency synthesizer for implantable medical devices. *IEEE Journal of Solid State Circuits*, 47(7):1896–1907, Jul 2012.

[23] Y. S. Lin et al. A 150 pW program-and-hold timer for ultra-low-power sensor platforms. In *2009 IEEE International Solid-State Circuits Conference*, pages 326–327,327a, Feb 2009.

[24] Y. Lee et al. A 660 pW multi-stage temperature-compensated timer for ultra-low-power wireless sensor node synchronization. In *2011 IEEE International Solid-State Circuits Conference*, pages 46–48, Feb 2011.

[25] M. Seok et al. A 0.5 V 2.2 pW 2-transistor voltage reference. In *2009 IEEE Custom Integrated Circuits Conference*, pages 577–580, Sept 2009.

[26] D. Griffith et al. A 190 nW 33 kHz RC oscillator with +/-0.21and 4 ppm long-term stability. In *2014 IEEE International Solid-State Circuits Conference*, pages 300–301, Feb 2014.

[27] S. Jeong et al. A 5.8 nW CMOS Wake-Up Timer for Ultra-Low-Power Wireless Applications. *IEEE Journal of Solid-State Circuits*, 50(8):1754–1763, Aug 2015.

[28] P. M. Nadeau et al. 4.2 pW timer for heavily duty-cycled systems. In *2015 Symposium on VLSI Circuits*, pages C240–C241, June 2015.

[29] M. Choi et al. A 99 nW 70.4 kHz resistive frequency locking on-chip oscillator with 27.4 ppm/^{o}C temperature stability. In *2015 Symposium VLSI Circuits*, pages 17–19, June 2015.

[30] Y. S. Lin et al. A sub-pW timer using gate leakage for ultra low-power sub-Hz monitoring systems. In *2007 IEEE Custom Integrated Circuits Conference*, pages 397–400, Sept 2007.

[31] A. Paidimarri et al. A 120 nW 18.5 kHz RC oscillator with comparator offset cancellation for +/-0.25% temperature stability. In *2013 IEEE International Solid-State Circuits Conference*, pages 184–185, Feb 2013.

[32] T. Tokairin et al. A 280 nW, 100 kHz, 1-Cycle Start-up Time, On-chip CMOS Relaxation Oscillator Employing a Feedforward Period Control Scheme. In *2012 Symposium on VLSI Circuits*, pages 16–17, Jun 2012.

[33] L. Joonhyung et al. Ultra low power RC oscillator for system wake-up using highly precise auto-calibration technique. In *2010 Proceedings of European Solid State Circuits Conference*, pages 274–277, Sept 2010.

[34] K. J. Hsiao. A 32.4 ppm/^{o}C 3.2-1.6 V self-chopped relaxation oscillator with adaptive supply generation. In *2012 Symposium on VLSI Circuits*, pages 14–15, June 2012.

[35] Y. Tokunaga et al. An On-Chip CMOS Relaxation Oscillator With Voltage Averaging Feedback. *IEEE Journal of Solid-State Circuits*, 45(6):1150–1158, June 2010.

[36] K. L. Wong et al. Enhancing microprocessor immunity to power supply noise with clock-data compensation. *IEEE Journal of Solid-State Circuits*, 41(4):749–758, April 2006.

[37] N. Kurd et al. Next Generation Intel Core Micro-Architecture (Nehalem) Clocking. *IEEE Journal of Solid-State Circuits*, 44(4):1121–1129, April 2009.

[38] X. Zhang et al. A novel VRM control with direct load current feedback. In *2004 19th Annual IEEE Applied Power Electronics Conference and Exposition*, volume 1, pages 267–271 Vol.1, 2004.

[39] M. S. Gupta et al. An Event-guided Approach to Reducing Voltage Noise in Processors. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 160–165, 2009.

[40] A. Grenat et al. Adaptive clocking system for improved power efficiency in a 28 nm x86-64 microprocessor. In *2014 IEEE International Solid-State Circuits Conference*, pages 106–107, Feb 2014.

[41] J. Cortadella et al. Reactive clocks with variability-tracking jitter. In *2015 33rd IEEE International Conference on Computer Design*, pages 511–518, Oct 2015.

[42] J. Tschanz et al. Adaptive Frequency and Biasing Techniques for Tolerance to Dynamic Temperature-Voltage Variations and Aging. In *2007 IEEE International Solid-State Circuits Conference*, pages 292–604, Feb 2007.

[43] K. Wilcox et al. Steamroller Module and Adaptive Clocking System in 28 nm CMOS. *IEEE Journal of Solid-State Circuits*, 50(1):24–34, Jan 2015.

[44] D. Bull et al. A Power-Efficient 32 bit ARM Processor Using Timing-Error Detection and Correction for Transient-Error Tolerance and Adaptation to PVT Variation. *IEEE Journal of Solid-State Circuits*, 46(1):18–31, Jan 2011.

[45] B. Zimmer et al. A RISC-V vector processor with tightly-integrated switched-capacitor DC-DC converters in 28 nm FDSOI. In *2015 Symposium on VLSI Circuits*, pages C316–C317, June 2015.

[46] B. Kim et al. A Supply-Noise Sensitivity Tracking PLL in 32 nm SOI Featuring a Deep Trench Capacitor Based Loop Filter. *IEEE Journal of Solid-State Circuits*, 49(4):1017–1026, April 2014.

[47] D. M. Chapiro. *Globally-asynchronous Locally-synchronous Systems (Performance, Reliability, Digital)*. PhD thesis, Stanford, CA, USA, 1985.

[48] E. J. Fluhr et al. POWER8TM: A 12-core server-class processor in 22nm SOI with 7.6Tb/s off-chip bandwidth. In *2014 IEEE International Solid-State Circuits Conference*, pages 96–97, Feb 2014.

[49] S. Rusu et al. Ivytown: A 22 nm 15-core enterprise Xeon processor family. In *2014 IEEE International Solid-State Circuits Conference*, pages 102–103, Feb 2014.

[50] R. Zhang et al. Architecture Implications of Pads As a Scarce Resource. In *Proceeding of the 41st Annual International Symposium on Computer Architecture*, pages 373–384, 2014.

[51] X. Mei and X. Chu. Dissecting GPU Memory Hierarchy Through Microbenchmarking. *IEEE Transactions on Parallel and Distributed Systems*, 28(1):72–86, Jan 2017.

[52] X. Mei et al. Benchmarking the Memory Hierarchy of Modern GPUs. In *2014 11th International Conference Network and Parallel Computing*, pages 144–156, 2014.

[53] H. Wong et al. Demystifying GPU microarchitecture through microbenchmarking. In *In International Symposium on Performance Analysis of Systems and Software*, pages 235–246, 2010.

[54] B. Keller et al. A Pausible Bisynchronous FIFO for GALS Systems. In *2015 21st IEEE International Symposium on Asynchronous Circuits and Systems*, pages 1–8, May 2015.

[55] A. Bakhoda et al. Analyzing CUDA workloads using a detailed GPU simulator. In *2009 IEEE International Symposium on Performance Analysis of Systems and Software*, pages 163–174, April 2009.

[56] R. Jevti et al. Per-Core DVFS With Switched-Capacitor Converters for Energy Efficiency in Manycore Processors. *IEEE Transactions on Very Large Scale Integration Systems*, 23(4):723–730, April 2015.

[57] R. Mullins and S. Moore. Demystifying Data-Driven and Pausible Clocking Schemes. In *13th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 175–185, March 2007.

[58] J. Zhao and Y-B.Kim. A Low-power Digitally Controlled Oscillator for All Digital Phase-locked Loops. *VLSI Design*, 2010:2:1–2:11, January 2010.

[59] G. Heck et al. A New Local Clock Generator for Globally Asynchronous Locally Synchronous MPSoCs. *Analog Integrator Circuits Signal Processing*, 89(3):631–640, December 2016.

[60] M. S. Golanbari et al. Post-fabrication calibration of Near-Threshold circuits for energy efficiency. In *IEEE International Symposium on Quality Electronic Design*, pages 385–390, Mar 2017.

[61] Y. Zhang et al. Hold time closure for subthreshold circuits using a two-phase, latch based timing method. In *2013 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference*, pages 1–2, Oct 2013.

[62] S. Rusu and S. Tam. Clock generation and distribution for the first IA-64 microprocessor. In *IEEE International Solid-State Circuits Conference*, pages 176–177, Feb 2000.

[63] A. Savanth et al. A 50 nW Voltage Monitor Scheme for Minimum Energy Sensor Systems. In *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems*, pages 81–86, Jan 2017.

[64] B. Zimmer et al. A RISC-V Vector Processor With Simultaneous-Switching Switched-Capacitor DC-DC Converters in 28 nm FDSOI. *IEEE Journal of Solid-State Circuits*, 51(4):930–942, April 2016.

[65] K. A. Bowman et al. A 16 nm All-Digital Auto-Calibrating Adaptive Clock Distribution for Supply Voltage Droop Tolerance Across a Wide Operating Range. *IEEE Journal of Solid-State Circuits*, 51(1):8–17, Jan 2016.

[66] M. Hiienkari et al. Ultra-wide voltage range 32-bit RISC CPU with timing-error prevention in 28 nm CMOS. In *2014 SOI-3D-Subthreshold Microelectronics Technology Unified Conference*, pages 1–2, Oct 2014.

[67] K. Souri et al. A CMOS temperature sensor with a voltage-calibrated inaccuracy of +/-0.15^{o}C (3σ) from -55 to 125 ^{o}C. In *2012 IEEE International Solid-State Circuits Conference*, pages 208–210, Feb 2012.

[68] K. Souri et al. A 0.85 V 600 nW all-CMOS temperature sensor with an inaccuracy of +/-0.4^{o}C (3σ) from -40 to 125 ^{o}C. In *2014 IEEE International Solid-State Circuits Conference*, pages 222–223, Feb 2014.

[69] Y-S. Lin et al. An ultra low power 1 V, 220 nW temperature sensor for passive wireless applications. In *2008 IEEE Custom Integrated Circuits Conference*, pages 507–510, Sept 2008.

[70] M. K. Law et al. A Sub-μW Embedded CMOS Temperature Sensor for RFID Food Monitoring Application. *IEEE Journal of Solid-State Circuits*, 45(6):1246–1255, June 2010.

[71] S. Jeong et al. A Fully-Integrated 71 nW CMOS Temperature Sensor for Low Power Wireless Sensor Nodes. *IEEE Journal of Solid-State Circuits*, 49(8):1682–1693, Aug 2014.

[72] A. Shrivastava et al. A 1.5 nW, 32.768 kHz XTAL Oscillator Operational From a 0.3 V Supply. *IEEE Journal of Solid-State Circuits*, 51(3):686–696, March 2016.

[73] D. A. Kamakshi et al. A 23 nW CMOS ultra-low power temperature sensor operational from 0.2 V. In *2015 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference*, pages 1–3, Oct 2015.

[74] S. C. Luo et al. A Wide-Range Level Shifter Using a Modified Wilson Current Mirror Hybrid Buffer. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(6):1656–1665, June 2014.

[75] M. K. Law and A. Bermak. A 405 nW CMOS Temperature Sensor Based on Linear MOS Operation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(12):891–895, Dec 2009.

[76] A. Muhtaroglu et al. On-die droop detector for analog sensing of power supply noise. In *2003 Symposium on VLSI Circuits*, pages 193–196, June 2003.

[77] M. Miyahara et al. A low-noise self-calibrating dynamic comparator for high-speed ADCs. In *2008 IEEE Asian Solid-State Circuits Conference*, pages 269–272, Nov 2008.

[78] D. Schinkel et al. A Double-Tail Latch-Type Voltage Sense Amplifier with 18 ps Setup+Hold Time. In *2007 IEEE International Solid-State Circuits Conference*, pages 314–605, Feb 2007.

[79] P. F. Chiu et al. A double-tail sense amplifier for low-voltage SRAM in 28 nm technology. In *2016 IEEE Asian Solid-State Circuits Conference*, pages 181–184, Nov 2016.

[80] B. Mishra et al. A sub-μA power management circuit in 0.18 μm CMOS for energy harvesters. In *2013 Design, Automation Test in Europe Conference Exhibition*, pages 1197–1202, March 2013.

[81] A. Roy and B. H. Calhoun. Exploring circuit robustness to power supply variation in low-voltage latch and register-based digital systems. In *2016 IEEE International Symposium on Circuits and Systems*, pages 273–276, May 2016.

[82] G. Geannopoulos and X. Dai. An adaptive digital deskewing circuit for clock distribution networks. In *IEEE International Solid-State Circuits Conference*, pages 400–401, Feb 1998.

[83] M. Maymandi-Nejad and M. Sachdev. A monotonic digitally controlled delay element. *IEEE Journal of Solid State Circuits*, 40(11):2212–2219, Nov 2005.

[84] N. Kamae et al. A body bias generator with wide supply-range down to threshold voltage for within-die variability compensation. In *IEEE Asian Solid State Circuit Conference*, pages 53–56, Nov 2014.

[85] M. Meijer et al. A forward body bias generator for digital cmos circuits with supply voltage scaling. In *IEEE International Symposium on Circuits and Systems*, pages 2482–2485, May 2010.

[86] A. Islam and H. Onodera. On-chip monitoring and compensation scheme with fine-grain body biasing for robust and energy-efficient operations. In *2016 21st Asia and South Pacific Design Automation Conference*, pages 403–409. IEEE, 2016.

[87] S. Narendra et al. 1.1 V 1 GHz Communications Router with On-chip body bias in 150 nm CMOS. In *IEEE International Solid-State Circuits Conference*, pages 218–482, Feb 2002.