Advanced Clocking and Power Management Techniques for Microprocessors

Fahim ur Rahman

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Visvesh Sathe, Chair

Chuan Shi

Keith Bowman

Arijit Raychowdhury

Program Authorized to Offer Degree:
Electrical and Computer Engineering
University of Washington

University of Washington

Abstract

Advanced Clocking and Power Management Techniques for Microprocessors

Fahim ur Rahman

Chair of the Supervisory Committee:
Assistant Professor Visvesh Sathe
Electrical and Computer Engineering

Ensuring high performance and low-power is the goal for almost all system-on-chips (SoCs). With the recent slow-down in technology scaling, innovations in circuits and systems are playing a significant role to improve the power-performance metric. This thesis proposed different techniques in clocking and power management of microprocessors to enable high-performance and energy-efficient operation while ensuring robustness across process, voltage and temperature variation. We proposed new circuit techniques, computational control or innovations in system-level that demonstrated substantial improvement in several key areas.

Switching energy is clock distribution constitutes a significant portion of power budget of SoCs.To reduce the clocking power in a system,resonant clocking technique was proposed. But resonant-clocking has limitations in frequency scaling. We propose quasi-resonant clocking (QRC), a new technique to achieve frequency scalability in resonant clocked system. QRC made resonant clocking viable for voltage-frequency scalable systems.

In modern microprocessors, latency in clock generation and frequency switching affects the processor start-up time and causes delay in resolving cache coherence request in multi-core systems. A fast-locking PLL can improve the latency and performance of multi-core systems. We analyzed the reason for longer lock-time in PLLs and came up with computational-locking, a new approach to achieve lock in PLL that improves the lock-time substantially. Measurement result from 65nm

test-chips demonstrate significant improvement in lock-time.

To further improve the power-performance metric, we analyzed the advantages and imitations of existing clock power architectures and implemented unified clock and power architecture (UniCaP). One of the limitations of traditional architecture is the system needs significantly large voltage margin against supply droop and temperature variation. In this thesis we show how UniCaP drastically reduces the voltage margin while ensuring performance regulation. We demonstrated UniCaP in a switched-capacitor based sub-threshold system, fabricated 65nm CMOS process. Measurement result demonstrated significant reduction of voltage-margin, resulting in significant energy savings.

We minimized the energy dissipation of the ultra-low power UniCaP system further by implementing a self-energy minimization feature. The system operates in minimum energy-per-cycle point while satisfying the performance requirements. We analyzed the energetics of switched-capacitor and derived the expression for energy-per-cycle (EPC). Then we constructed a system that will estimate the EPC at different voltages and through comparison will approach the minimum energy-per-cycle point. We implemented the system in 65nm test-chip and measurement result demonstrated MEP-operation across temperature and load variation with less than 5mV of error.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I started my PhD in University of Washington in 2014 with almost no experience in VLSI circuit design. It is the journey of last four and half-years that taught me everything I know about circuit design, reshaped my though process and spurred my innovative instincts. I would not be in my current position without the help of some extra-ordinary people.

First of all I would like to thank my father, Khalilur Rahman, my mother, Fowzia Begum and my sister, Ishrat Jabeen. All my achievements in my life was possible because of their constant support. Even being eight thousand miles away, they constantly pray for my well-being and support me whenever they feel I need any help. I am anxiously waiting for my commencement to see their smiling faces in the audience seats.

I would like to thank all my mentors in University of Washington. I am truly grateful to the dissertation committee members for their constant guidance and support. I am really lucky to look into many interesting problems during my PhD. Credit goes to the committee members for guiding me to such interesting problems, directly and indirectly. I thank them for taking the time to review my work. They are one of the most innovative circuit designers I have ever met.

My researches would not have been possible without the financial support I received throughout the years. I would like to thank Intel Corporation, Qualcomm Tech, Inc and Semiconductor Research Corporation for the financial grants. I am also grateful to Analog Devices for the Outstanding Designer Award and Solid-state Circuit Society for the SSCS Pre-doctoral Achievement Awards, they really inspired me to pursue my researches.

I am really grateful to work in collaboration with some of the finest designers in the semiconductor industries. Special thanks to Greg Taylor from Intel Corp. for working with us in the computationally-locked PLL project. His guidance and technical expertise were crucial to the

implementation of computationally-locked PLL. Keith Bowman from Qualcomm collaborated with us in the researches in clock-power management systems. His immense knowledge, attention to the finest details helped me in my researches and subsequent publications. My internship manager at ARM, Bal Sandhu also deserves special thanks. I would always remember his humble and supportive personality.

Alvin Loke, my internship mentor from Qualcomm deserves a special mention. I never knew I would meet my best mentor at the very end of my formal education. He is the smartest and the most humble person I have ever met. During my internship in Qualcomm I learnt so many things from him about circuit design and process technology. I will miss our morning hikes, trying out different foods and all the discussions. I will keep on trying to follow his problem solving style, versatility in thinking and work ethic. I would also like to thank Joohwa Kim, Thanh Tran and my internship manager Rajit Chandra from Qualcomm for their supports during my internship.

I am really grateful to befriend some really supportive people during my PhD. I would really like to thank Jabeom Koo for being there. I have never met anyone so helpful and polite. I have gone through really difficult time during my first year. If it hadn't been for him, I would have quit my PhD long time ago. When I used to have a really hard time in the lab, I remember him staying late just to walk and chat on our way back to make me feel normal. He taught me many so many things without losing patience for a moment, helped me debugging codes, reviewed my layouts in the middle of his busiest schedules. An honest and original person like him is rare and I am grateful that one such person was in UW VLSI during my initial years. I would also like to thank my lab-mate Patrick Howe for his supports. I miss our hang-outs at the Ave, HUB, Waterfronts and the chit-chats during our walk back home around 2am in the morning. My heartiest thanks to Tong Zhang from FAST-lab for teaching me so many things and always helping me in my researches. Even two years after graduating, he is my go-to person whenever I have any queries.

Apart from the people in UW, I am glad to have some amazing friends in Seattle. Reuniting with my school friend Mishan and Fuad in Seattle was an unexpected gift for me. I thank them for

so many loving memories throughout my PhD, the last two years have been really great. I am also thankful to Shamsi Tamara Iqbal and Konok Bhaia, my local guardians. They have gone above and beyond numerous times to help us out. I would also like to thank Professor Fahad Khalil, Nandini Abedin, Golam Rabbani for so many fond memories.

Thanks Aladin Gyrocery for serving food after midnight. It was always refreshing to catch up with my friend Mushal from Aladin. I wish him all the best.

Thanks to my beloved team Arsenal FC for their constant poor performance that kept me away from watching football over the last four years.

Finally I would like to thank my best friend, my loving wife Hannah. Words cannot express how much she sacrificed for my PhD. She has been constantly by my side, in the struggles, failures and in the triumphs. I can only hope to provide her the same through the rest of my life.

DEDICATION

To my father, mother, sister and my dear wife, Hannah

Chapter 1

INTRODUCTION

Over the last few decades, we have witnessed a drastic improvement in computational engines resulting in different kinds of system-on-chips (SoCs). Continuously improving the speed (performance) while reducing the power of SoCs has resulted in state-of-the art microprocessors for PCs, laptops and mobile phones (Figure 1.1). At the same time the demand for more efficient computing resources are increasing. With the recent revolution in the field of artificial intelligence and data science factoring in the exponential increase in users, we need to accelerate the improvement rate in SoC power-performance for processing the enormous amount of data generated every moment.

The demand for high-performance and low-power circuits have been met by scaling of process nodes [1–11], as governed by Dennard scaling [12]. As mentioned in [12], by reducing the supply voltage along with the critical dimension of CMOS transistors, a chip scaled from one technology generation to the next could integrate roughly twice the transistors for the same die area, run them at a higher frequency, and still maintain roughly constant total power. But, as the boundary of

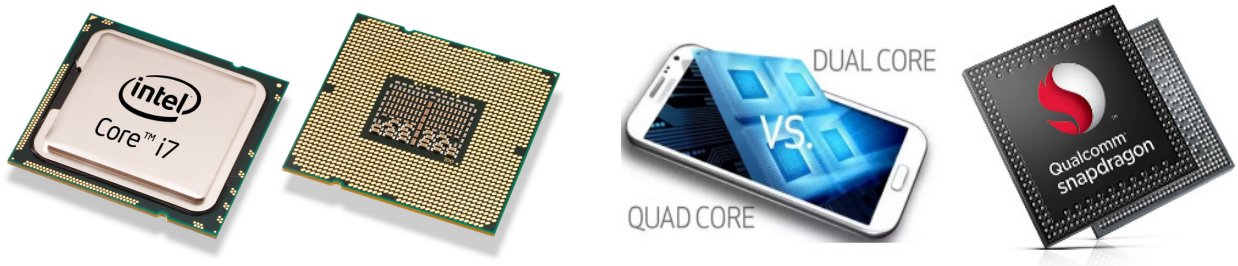


Figure 1.1: High performance microprocessors in commercial PCs and mobile phones

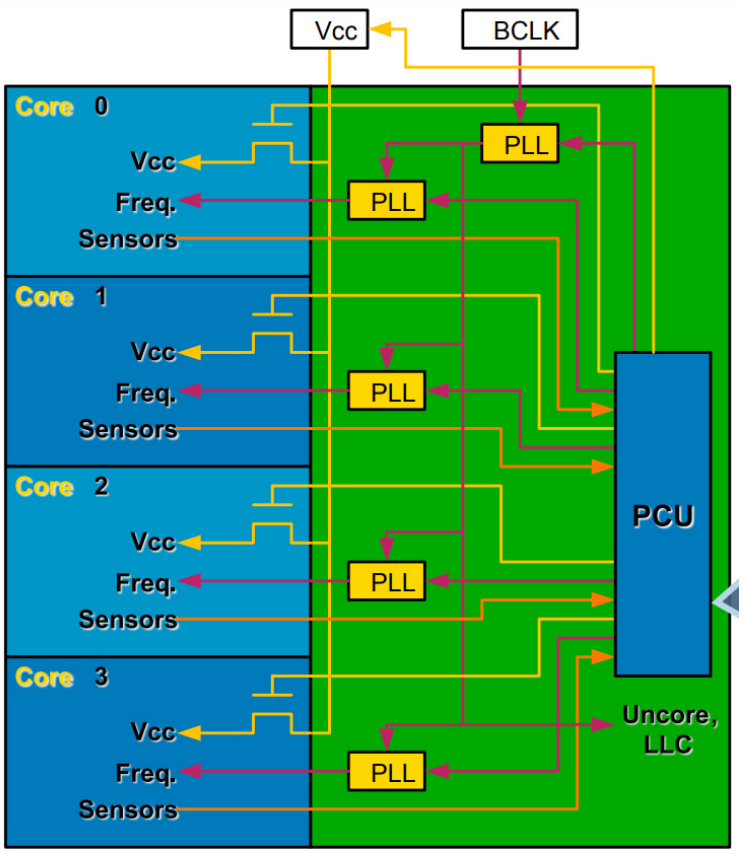


Figure 1.2: Multi-core procesor architecture

technology scaling was pushed near the atomic lengths, the improvement in power and performance did not follow Dennard's scaling law [13]. With the death of Dennard's sclaing law, innovations in circuits and systems play the major part in pushing forward the power-performance limit of SoCs.

To improve the SoC power-performance, multi-core architecture in processor was proposed. By exploiting parallellism in computing, multiple independent processors can operate at lower frequency to perform the tasks. Adopting multi-core architecture has proven to be an effective solution. This resulted in graphical processor unit (GPU) and multi-core processor [14]. At present, almost all commercial PCs and cell-phone processors have shifted into multi-core paradigm.

Figure 1.2 shows a simplified architecture of a multi-core processor. Here several cores (core0-4) are running independently at different voltages (VCCs) and frequencies. If inactive, each core can

be turned off. For each core, a voltage regulator is needed to regulate the operating voltage. A PLL is needed in each core to generate the clock. During operation, the voltage and frequencies are scaled according to the computation load, a technique known as Dynamic-voltage-frequency-scaling or DVFS. A top level controller (PCU in Figure 1.2) orchestrates the communication between cores and asserts the voltage and frequency of each core. Adopting multi-core architecture came up with new design challenges. High-performance multi-core processors tend to dissipate a significant amount of power in their clock distribution network. The clock generation latency directly impacts the latency in starting up the core and DVFS. More importantly, the latency in waking up a core results in delay resolving cache-coherence request in multi-core architecture.

Variability in device and circuit parameters adversely affects the performance and energy efficiency of micro-processors across all market segments, ranging from the small embedded core in a system-on-chip (SoC) to large multi-core servers. Dynamic variations like supply voltage droops, temperature changes, and transistor aging changes the transistor speed in the processor. The variability is taken into account by applying a voltage margin, which results in additional energy dissipation (explained in chapter 4). Droops result from abrupt changes in switching activity, inducing large current transients in the power delivery system. Fore multi-core architecture droops are even worse because of the combined effect of load fluctuation across all the cores. Reducing the voltage margin can drastically improve the power-performance metric.

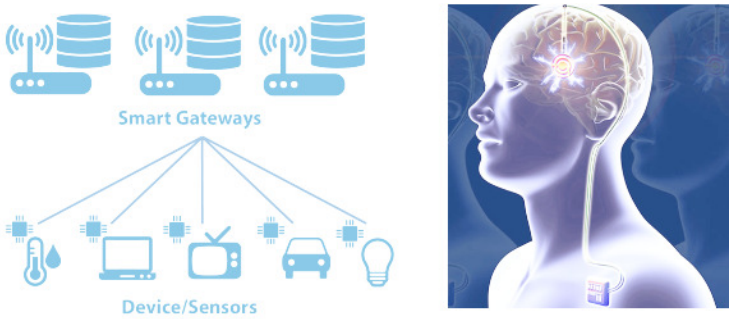


Figure 1.3: Application of ultra-low-power SoCs in edge computing and sensor applications in Internet-of-things (IoT) and medical implants. Image courtesy: (1) OpenFog Reference Architecture for Fog Computing, (2) Eliza Strickland, IEEE spectrum

To the other-end of high performance computing trend, we have witnessed a significant drive towards ultra-low-power SoCs (Figure 1.3. With the proliferation of Internet-of-Things (IoT), the sensors and devices on the edge need their own processor for computing. These devices are either powered by energy harvesting or portable batteries. For energy harvested devices, ultra-low power and energy efficient processors can perform more computation within the energy budget. For battery powered devices, more efficiency results in improvement in battery life or smaller battery sizes. For medical implants, where battery life is a major concern, improving efficiency is extremely important.

For ultra-low-power applications a significant amount of energy is dissipated in the voltage guard-band. The reason is, for ultra-low-power operation, the devices operate in near/sub-threshold region where sensitivity to process,voltage and temperature variation (PVT) is higher. Also ensuring operation at the optimum the supply voltage and frequency can minimize energy dissipation.

1.1 Research objective

This dissertation will focus on investigating, analyzing and implementing novel system/circuit level techniques in clocking and power management to improve SoC performance. The following approaches are used to improve the power-performance metric:

- To reduce clock power dissipation while ensuring frequency scalability, we propose Quasi-Resonant-Clocking (QRC).
- For improving the latency of processor, we focused on reducing the clock generation latency. We analyzed PLL locking dynamics and came up with computational locking, a computationally enabled PLL will drastically reduced locktime (16 reference clock cycles).
- We investigated the voltage margin problem in SoCs and found out that clock power architecture is the key reason for having high voltage margin. We implemented a unified clock and power architecture (UniCaP) that drastically reduces the voltage margin while ensuring the performance regulation. We demonstrated UniCaP ona a ultra-low-power SoC where voltage margin problem was acute.
- To minimize the energy dissipation in ultra-low-power devices, we focused on ensuring system

operation at minimum energy per cycle. We implemented a system that always operates on the optimum point while maintaining the performance requirements

1.2 Organization

The rest of the thesis is organized as follows:

Chapter 2: This chapter describes Quasi-Resonant-Clocking (QRC) [15, 16], a resonant clocking architecture capable of efficiently enabling Dynamic Voltage and Frequency Scaling, and achieving continuous voltage-frequency scalability. The use of runtime control by QRC is central to ensuring robust, efficient operation across variations in Process, Voltage and Temperature. QRC exhibits instantaneous wide-range duty-cycle control, required for a broader class of clocking applications involving large capacitive loads. Two QRC variants, QRC_{footer} and QRC_{pass}, are presented with test-chip measurements under DVFS operation in 65nm CMOS. The QRC test-chip demonstrates a maximum energy reduction over conventional clocking of 47% at 132MHz. Across both implementations, energy measurements based on runtime DVFS in the 0.7V-1.2V range indicate energy reduction in the 32%-47% range.

Chapter 3: This chapter describes 'Computational Locking' [17], a new methodology for achieving fast frequency and phase acquisition in All-digital PLL(ADPLL) with the targeted application of system clocking. Without affecting the steady-state peformance, computational locking enables accelerated phase-locking during 'cold-start' and 'relock'; improving the overall performance of microprocessors. We also present a novel TDC architecture that provides wide input range, low resolution and low resolving time. Implemented in 65nm CMOS technology, a computationally locked 1-2GHz PLL demonstrates a mean 'relock'-time of 12 $T_{REFCLK}$and 'cold-start' locktime of 16 T_{REFCLK}measured over 50,000 relock and cold-start experiments.

Chapter 4: In this chapter, we present an all-digital Unified Clock and Power (UniCaP-SC) architecture that combines switched-capacitor (SC) based voltage control and clock frequency regulation into a single loop to significantly reduce required V_{dd} guardbands [18, 19]. A UniCaP-SC test-chip consisting of a near-threshold voltage (NTV) ARM Cortex-M0 processor was fabricated in 65nm CMOS. The fully-integrated system enables all-digital construction, aggressive V_{dd} margin

reduction, and continuous V_{dd} scalability using SC-based voltage converters while no additional decoupling capacitance (decap). Test-chip measurements demonstrate a 16% Vdd reduction corresponding to a 94% V_{dd} margin recovery or an equivalent 3.2× increase in the operating clock frequency (f_{clk}).

Chapter 5: Here we a fully-autonomous self-energy minimizing system suitable for ultra-low power applications under performance constraints [20]. The switched-capacitor regulator based system can measure the total energy-per-cycle(EPC) (including the regulator losses) at run-time without using any off-chip components. A minimum EPC tracking loop finds the minimum energy-per-cycle point(MEP) by comparison of EPC at different operating points. The system is built upon robust-efficient unified clock and power (UniCaP) architecture which ensures performance regulation with minimum voltage guard-band. Measurement results from the 65nm CMOS test-chip demonstrate V_{MEP} tracking with less than 5mV of error within the operating voltage of 0.38-0.58V. Measurement result also demonstrates MEP tracking with temperature variation from -30°C to 120°C and different loading condition and 20 different test-chips.

Chapter 2

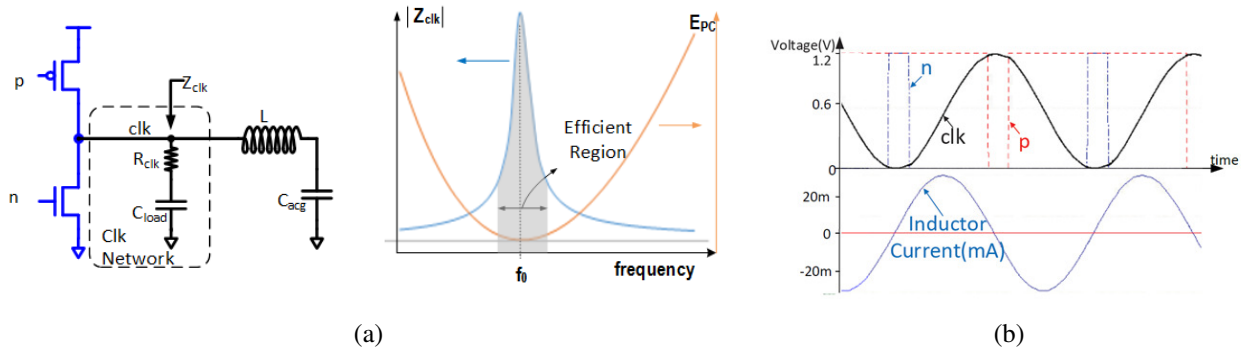
QUASI-RESONANT CLOCKING : CONTINUOUS VOLTAGE-FREQUENCY SCALABLE RESONANT CLOCKING SYSTEM FOR DYNAMIC VOLTAGE-FREQUENCY SCALING SYSTEMS

Power dissipation continues to play a central role in impacting computing performance, mobility and cost over a broad range of digital systems, from high-performance microprocessors [21–23] to ultra low-power systems employing aggressive pipelining [24]. Fine-grained clock gating has significantly reduced clock power [23, 25], but the large global clock distribution driving these clock-gates accounts for a sizable fraction of total system power [26, 27]. Enabling clock power reduction therefore, remains critical to the efficient implementation of digital systems.

Resonant clocking has been proposed as an efficient approach to global clock distribution [28–39]. Figure 2.1 illustrates the central idea behind resonant-clocking – introducing inductance (L) into the clock network, and enabling efficient LC resonance between the clock load (C_{load}) at frequencies close to the natural (or resonant) frequency, $f_0 = {}^{1}/_{(2\pi\sqrt{LC})}$. In contrast with conventional CV^2 per-cycle energy dissipation, the energy delivered by the supply to a resonant clock system each cycle needs to only compensate for the I^2R losses in the LC tank. High quality-factor (Q) designs, achieved through reduced R_L and R_{clk} sustain clock oscillations with a significantly lower energy dissipation per cycle(E_{PC}): $E_{PC} = \frac{\pi}{4Q} C_{load} V_{dd}^2$ [35].

Recent implementations have overcome several outstanding processor integration challenges. Two such important challenges are the design of inductors in-line with power trunks [26, 27, 34, 40–42], and addressing system-Q degradation due to power grid eddy-current flow [26]. The result of these efforts has been processor designs with significant clock power reduction, positively impacting power-constrained performance and battery-life.

Meanwhile, the increased significance of Dynamic Voltage and Frequency Scaling (DVFS) [43],



(a) (b)

Figure 2.1: Traditional resonant clocking : (a)Simplified resonant clock schematic with lumped global clock drivers and load. Driving the resulting tank circuit yields efficiency around the natural frequency f_0. C_{acg} is a large dc-current blocking capacitor. (b)Schematic simulation waveform of traditional resonant clocking using a pulse driver [29, 33, 35] to generate sinusoidal waveforms.

and a trend toward its increasingly aggressive use across a broader voltage-frequency range dilutes resonant clocking efficiency benefits. Resonant clock efficiencies, attractive at clock frequencies close to f_0 (Figure 2.1), quickly diminish along with Z_{tank} outside a small frequency range away from f_0. Although power-constrained performance benefits remain, the broader energy-efficiency advantages of the technique are diminished as systems spend a smaller portion of time within this narrow frequency range.

Recent works have attempted to address this challenge using techniques suitable for their specific applications [27, 44]. However, these techniques face challenges of limited frequency-tunability [27], limited duty cycle control [44], or clock waveforms that transition to $-V_{dd}$, limiting scalibility [44].

We present Quasi Resonant Clocking (QRC) [15, 16], a DVFS-compliant technique that achieves continuous voltage and frequency scalability. The proposed approach produces rail-to-rail clock waveforms with near-arbitrary duty-cycle control. With the exception of [44] (itself restricted to sub-threshold designs) the proposed approach is the only architecture capable of achieving *uniform* energy efficiency continuously across the 0 to f_0 frequency range. A key enabler of efficient, Process-Voltage-Temperature (PVT) robust operation, and DVFS support is *autonomous* timing

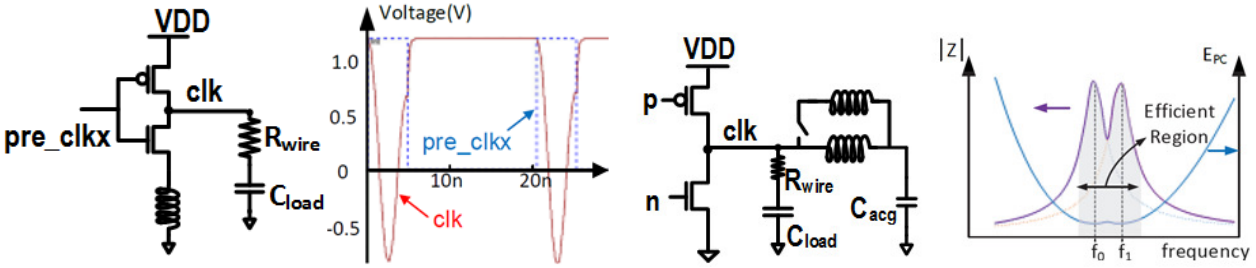
control of QRC related circuits which can track slewing supply-voltages during DVFS events, and temperature variations. The proposed technology is capable of a frequency range of 0–f_0, and sub-threshold to nominal-V_{dd} operation. We present circuits and architectures for two different variants of QRC, notably $QRC_{pass}$and QRC_{footer}which offer different capabilities and efficiencies. We also present an analysis of QRC energy and its dependence on key QRC parameters. This model is used to determine optimal design parameters, and examine the fundamental trade-off between efficiency and rise/fall times exhibited by all resonant-clock systems. The two proposed QRC variants are validated through test-chip demonstrations of a DVFS system consisting of an 8-way Multiply-Accumulate (MAC) array in 65nm CMOS. Lastly, we present test-chip measurements obtained from each of these two implementations.

The remainder of this chapter is organized as follows: In Section 2.1, we review related work in the area of frequency-scalable resonant clocking. In Section 5.2, we provide an overview of the QRC architecture and the salient aspects of its operation. Circuit and architecture descriptions for each variant of the QRC implementation are discussed in Sections 2.3 and 2.4. An analysis of QRC, including the examination of energy-optimal design parameters is presented in Section 2.5. Test-chip architecture, evaluations and measurements of both variants, $QRC_{footer}$and QRC_{pass}, are presented in Section 2.6.

2.1 Related Work

Several prior works in the literature have focused on addressing the limited frequency range of resonant clocking. These approaches range from disabling resonant clocking at frequencies outside a narrow range [26], to alternative resonant clocking implementations which extend resonant operation outside its narrow operating range.

A technique originally proposed and demonstrated in [45, 46] for I/O pad and display drivers, and re-implemented more recently in [47] involves using switched capacitor banks to transition a clock load through a sequence of intermediate voltage levels. Although slightly less efficient than inductor-based techniques, the switched-capacitor based technique offers notable advantages: (1) Simpler logic control, (2) Ease of integration and (3) Natural voltage scalability. However, though



(a) (b)

Figure 2.2: (a)Intermittent resonance enables continuous frequency control of a clock "blip" for sub-threshold applications, (b)Tuned-inductor based resonant clocking extends the range of efficient resonance through programmable inductance of the LC tank.

effective for its originally intended application of driving adiabatic logic or LCD drivers [45, 46], this approach is not well suited to system clocking – Employing the technique efficiently results in clock waveform "shelving" [46] (the clock voltage levels off at intermediate voltages during) that impacts efficiency and race-immunity among timing elements. In more severe situations, non-monotonic clock waveforms near mid-rail voltages [47] can result in runt clock pulses.

More recently, intermittent resonant clocking, a novel technique achieving uniform efficiency resonant clocking from 0–f_0 has been proposed [44]. Illustrated in Figure 2.2a, the technique presents a novel topology to deliver an *RLC* oscillation enabled clock "blip" from V_{dd} to $-V_{dd}$. The Pmos can hold *clk* at V_{dd} for an arbitrary duration before commencing the next "blip". Frequency scaling is achieved by varying the duration of time that the *clk* net remains at V_{dd}. The proposed architecture, developed for sub-threshold operation, does not scale to higher voltages by construction. Non-compliant CMOS voltages - the clock waveform transitions below 0V to a body-drain diode-drop voltage of approximately 700mV - prevent operation at nominal voltages. Further, operation at higher voltages also results in the dissipative turn-on of the Nmos diode turn-on as clock transitions below -700mV, shunting the *RLC* system. Finally, the inability to control duty cycle limits the approach to use in flip-flop based system clocking applications.

Bandwidth extension through additional programmable inductors to modulate total induc-

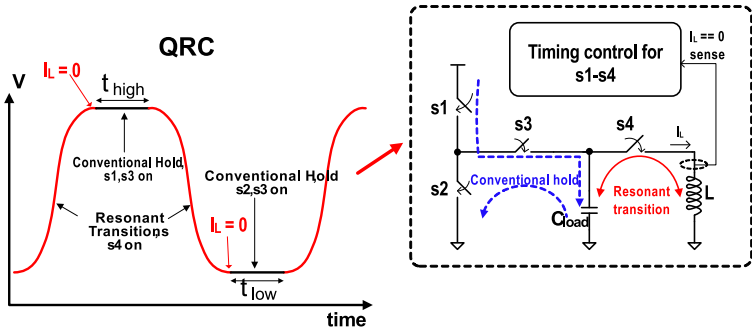


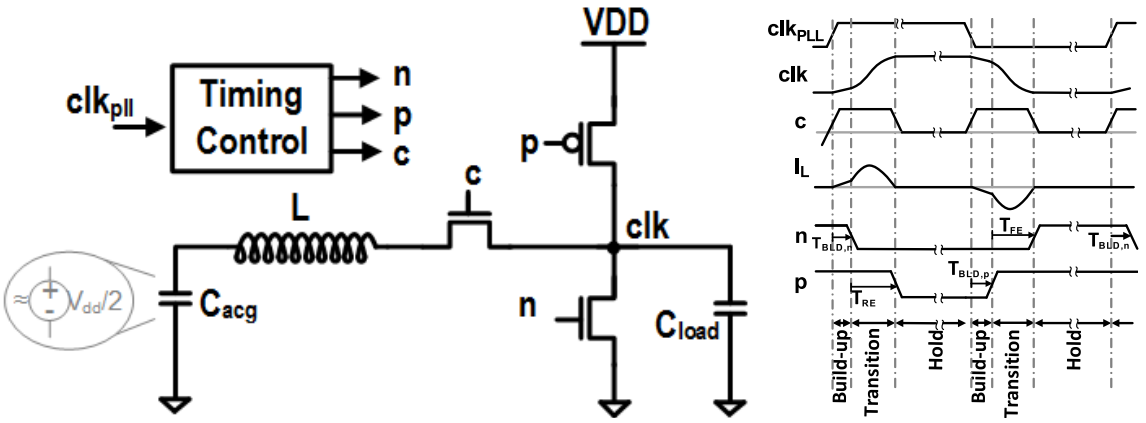
Figure 2.3: Quasi-Resonant Clocking: Key rationale.

tance (Figure 2.2b) and shift f_0 has been demonstrated [27], and recently implemented in a production processor [42]. The proposed approach caters to its target application, but does not provide continuous Voltage-frequency scalability across a broad range. It is therefore not well-suited to applications seeking a truly broad resonant-clock operating frequency range. Furthermore, the placement of additional inductors in an already physically and electrically constrained environment places additional inductor design challenges and complexity, and relies on technology to deliver inductors of sufficient quality-factor.

2.2 QRC Overview

Quasi-resonant clocking achieves continuous frequency scalability by effectively interleaving conventional and resonant modes of clock operation. Figure 2.3 illustrates the rationale behind the QRC architecture. The time instant when the clock voltage is at its maximum or minimum is ideal for disconnecting the inductor from the clock domain (discussed further in this section). Disconnecting the inductor safely and efficiently enables the clock to be held to the supply and ground rail for an arbitrary duration of time (τ_{high} and τ_{low} respectively), readily achieving frequency and duty-cycle control.

A simplified circuit diagram for the proposed QRC clocking scheme is shown in Figure 2.4a. Conduction switch M_c conditionally disconnects the inductor from the clock network at the end of each resonant transition, twice every cycle. C_{acg} blocks DC inductor current flow and is large



(a) QRC: simplified schematic

(b) QRC: simplified timing diagram

Figure 2.4: The QRC architecture relies on precise timing of n, p, c signals provided by the Timing Control Module to achieve interleaved conventional and hold operation.

enough to act as an AC ground connection.

Figure 2.4b shows a simplified QRC timing diagram, identifying signal transitions for n, p and c, controlled precisely through the *Timing Control* module. Consider the steady-state operation of the QRC system. The voltage across C_{acg} ($C_{acg} \approx 30 \cdot C_{load}$) remains steady at $V_{dd}/2$ (for a 50% duty-cycle clock). At the start of the clock cycle, V(clk_{PLL}) = 0, V(clk) = 0. As clk_{PLL} transitions to V_{dd}, c is asserted and connects the inductor to the resonant system, effecting an RL current build-up in the inductor. $T_{BLD,n}$ and $T_{BLD,p}$ correspond to the inductor-current build-up times before the rising and falling transitions of the resonant clock respectively. Use of $T_{BLD,p}$ and $T_{BLD,n}$ are motivated by two factors:(1) A longer build-up provides a sharper slew for the output clocks, essential for reliable digital systems. (2) Suitably tuned build-up durations provide improved efficiency [26, 35]. The optimal build-up duration varies depending on a number of factors including the resistance in the build-up and resonant paths and the load capacitance.

After duration $T_{BLD,n}$, n transitions to 0, turning off the hitherto conducting M_n, starting the LC transition of clk from 0 to V_{dd}. After a delay (T_{RE}) determined by the natural frequency of the LC network, clk reaches its peak voltage. A comparator detects this peak, prompting the timing

module to de-assert c and disconnect the inductor from the network precisely when inductor current $I_L = 0$ with help from a timing-control module. Concurrently, p transitions to 0, turning on M_p, and driving clk to V_{dd} rail. At the end of this sequence, clk has transitioned to V_{dd} and the inductor is disconnected from the grid, enabling M_p to hold clk to V_{dd} indefinitely until the next transition of clk toward 0. The sequence of events during the falling clk edge is conceptually identical to the rising edge sequence with a reversal in I_L direction. The occurence of clk maxima or minima coincides with $I_L = 0$, which also results in a reversal of polarity of the conducting M_c switch. 'clk'maxima or minima detection is therefore performed by comparing the potential difference accross M_c.

The clock cycle time, T_{clk} can be written as : $T_{clk} = T_{BLD,p} + T_{RE} + T_{\tau\_high} + T_{BLD,n} + T_{FE} + T_{\tau\_low}$ Frequency and duty cycle programmability is achieved by varying T_{τ_high} and T_{τ_low} independently in QRC [48]. T_{RE} and T_{FE} are determined by the natural frequency of the LC system: $T_{RE} + T_{FE} = \frac{1}{f_0} = 2\pi\sqrt{LC}$. Consequently, the maximum operating frequency using QRC is therefore limited by f_0, corresponding to ordinary resonant operation.

DVFS events require charging C_{acg} to $\frac{V_{dd}}{2}$. However, at prevalant levels of capacitance ($\approx 30 \cdot C_{load}$), the charge-discharge durations amount to a few clock cycles, and do not pose an overhead for DVFS. Additionally, any settling time resulting from the voltage transition has a minor temporary impact on clock slew and efficiency.

To provide PVT–robust, efficient clocking compatible with runtime voltage and frequency transitions, QRC faces two important challenges – Timing control, and the efficiency of the M_c switch. Turning M_c off while $I_L \neq 0$ results in wasteful dissipation and voltage ringing that exceeds the supply rail, degrading both efficiency and reliability. Furthermore, precise timing must be maintained across variation in PVT, and across the DVFS-enabled V_{dd} range. An all-digital runtime control system relying on a sense-amplifier and dual-edge triggered DLL provides the necessary current detection.

The second challenge of ensuring robust and efficient operation arises from the quiescent $V_{dd}/2$ voltage across C_{acg}, motivating design of a gate driver to meet the resulting gate-overdrive needs of M_c. These challenges manifest separately in $QRC_{footer}$and $QRC_{pass}$and are therefore addressed individually through different gate-driver circuits.

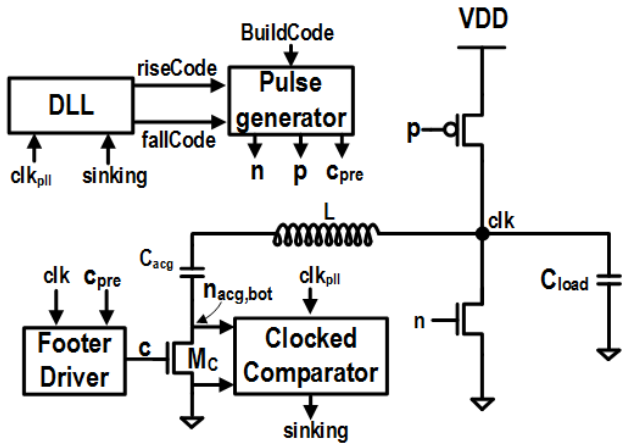


Figure 2.5: QRC_{foot} simplified schematic. Use of M_c as a footer requires a footer-driver to prevent current backflow. A clocked comparator provides all-digital zero-I_L detection.

2.3 QRC_{footer} **Design**

In this Section, we discuss the architecture and circuits that constitute the QRC_{footer}65nm test-chip implementation. In particular we describe two key enabling sub-systems: A DLL-enabled *Timing Control* module and the footer-driver circuit.

Figure 2.5 illustrates the QRC_{footer}architecture [15]. M_c is connected as a footer device between the bottom-plate of C_{acg} and ground. A footer arrangement provides full gate overdrive ($V_{gs} = V_{dd}$). The *Timing-control* module uses a comparator sampling the source-drain terminals of M_c, and clocked by c to provide feedback to the *Timing Control* module to ensure carefully timed assertions and de-assertions of p, n, and c_{pre}. Although M_c has full gate-overdrive when c = V_{dd}, a driver circuit is required to enforce device cutoff when $V(clk)$ is in conventional mode, as discussed further in Section 2.3.2.

2.3.1 Timing Control Module

The *Timing Control* module orchestrates the assertion/de-assertion of n, p and c to enable robust, efficient operation under slewing V_{dd} and f_{clk} conditions expected in DVFS . Timing requirements

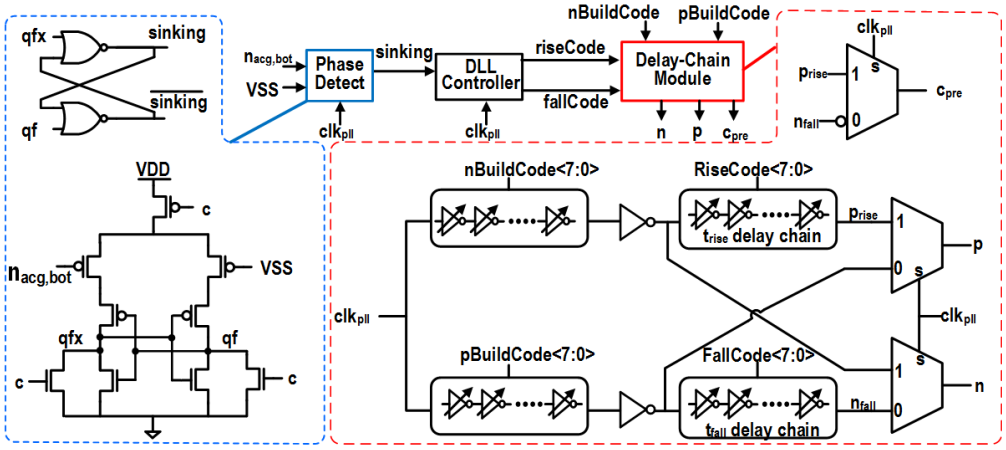
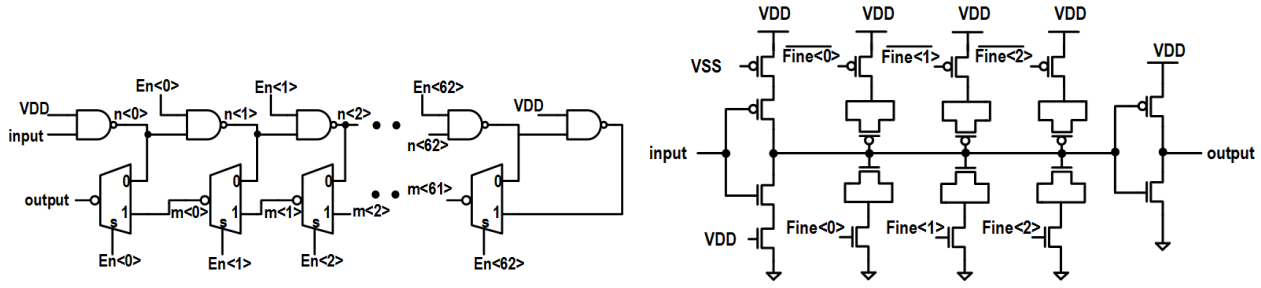


Figure 2.6: The Timing Control Module(shown for QRC_{foot}, Figure 2.5) employs a "dual-DLL" architecture, controlling two delay chains with a single front-end phase-detector, precisely timing signals n, p and c. MUXes provide glitchless, context-dependent delay.

for n, p and c are stringent to ensure the inductor is disconnected from the clock domain when $I_L = 0$ at the end of each resonant transition to avoid wasteful, reliability-degrading under-damped voltage oscillations. The assertion and de-assertion delays of n and p are determined by frequency of the output clock (f_{clk}), f_0, $T_{BLD,n}$, $T_{BLD,p}$ (Figure 2.5). Finally c switches twice every cycle, and is asserted for durations T_{RE} and T_{FE} of each resonant transition of clk.

Figure 2.6 summarizes the implementation of the *Timing Module*. A DLL implementation aligns two events – the de-assertion of c (the gate-driver output, which disconnects the inductor from the network), and the advent of zero current flow in the inductor. The need to align these events twice every cycle (one for each clock edge) when transitioning from resonant to conventional mode motivates a 'dual-DLL' architecture, with a shared front-end phase-detector and two delay-chains to achieve the desired timing control for each transition. Independent delay chains provide separate control of current buildup durations ($T_{BLD,n}$, $T_{BLD,p}$) and clock transition times (T_{RE}, T_{FE}) (Figure 2.5) for rise and fall clock edges to support non 50% duty-cycles.

A p-type strong-ARM latch, triggered on the de-asserting (for an Nmos M_c) edge of c, performs phase-detection, determining current flow direction through M_c by sampling the relative polarity



(a) Coarse delay chain

(b) Fine delay chain

Figure 2.7: Delay chain design ensures the range and precision required at across the entire V_{dd} range.

of its source and drain terminals twice every cycle. Thus, the phase-detector determines whether c is de-asserted too early or late relative to the $I_L = 0$ event based on the direction of current flow (with C_{acg} either *sourcing* or *sinking* current). This phase-detector output provides the required early/late signals to the dual-DLL. The DLL controller updates the rise and fall delay chains, with the $rise\_delay$ ($fall\_delay$) chain updated based on the comparator sample when the clk voltage is V_{dd} (0). Delay-code update-driven glitches are avoided by updating delays when delay-chain outputs are not observed–Updates to the $rise\_delay$ ($fall\_delay$) chain occur when clk is 0 (V_{dd}). Glitch-less, context dependent delay for n, p and c is provided by using selection MUXes.

By starting resonant clock transitions a fixed delay ($T_{BLD,n}$ and $T_{BLD,p}$) after clk_{PLL}, QRC is transparent to duty-cycle changes made at its input, on clk_{PLL}. This feature is beneficial for optimizing phase-paths in some digital designs. Further, any changes in input clock duty-cycle are instantaneously referred to the output, a key benefit in IVR applications involving runtime modulation of clock duty-cycle.

Programmable delay is achieved through a telescopic delay chain [49] for coarse resolution, combined with a programmable load inverter chain for fine delay control (Figure 2.7). A thermometer code controls the coarse telescopic delay chain [49], and MOS-based capacitors offer programmable load in the fine delay chain. The delay chain was designed to provide sufficient

resolution at $V_{dd} = 0.7V$, and range at $V_{dd} = 1.2V$ for robust, efficient QRC operation.

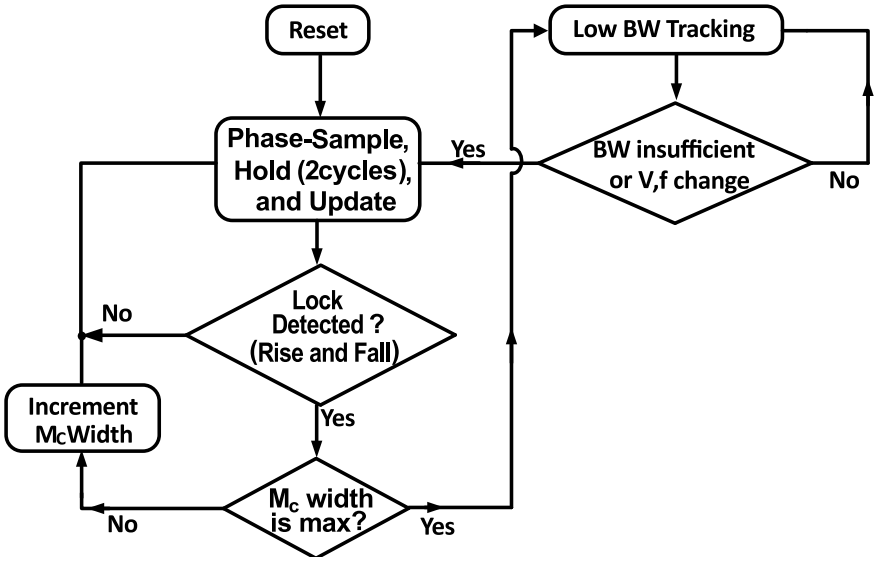
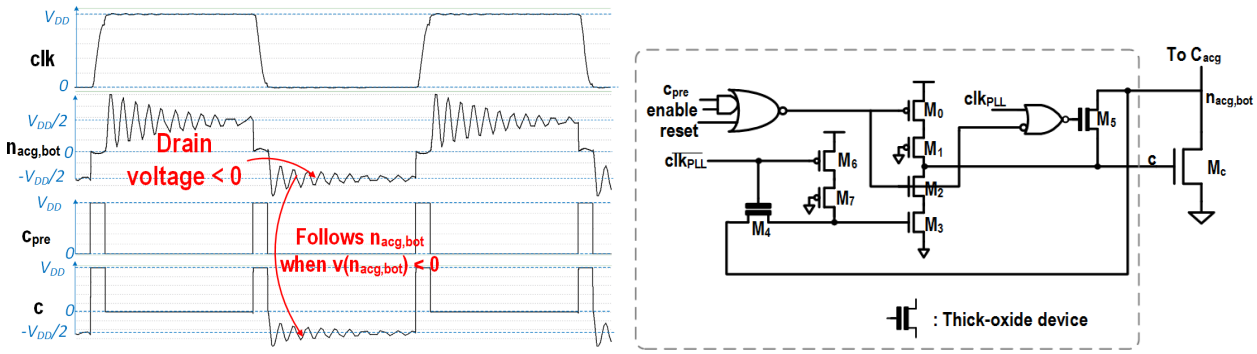


Figure 2.8: DLL lock acquisition procedure, avoiding reliability challenges before DLL-lock, and finally transitioning to a low-bandwidth mode sufficient to track temperature variation.

Figure 2.8 summarizes the operation of the DLL controller. Upon cold-start, the system first safely determines PVT-dependent delay-code settings. Reliability degradation due to initially mistimed c signals upon start-up are managed by first locking with only a fraction of the M_c device, limiting I_L, and throttling any under-damped voltage oscillations. After achieving DLL lock for both edges (rising and falling), the controller increases M_c width and returns to locking mode, making any necessary delay adjustments to re-lock. This interleaved lock and M_c-width update is repeated until the target M_c width is reached. Once locked, the DLL transitions into a low-power mode, with a sampling rate (kHz range) required to track temperature changes on the die, rendering DLL controller power negligible.



(a) (b)

Figure 2.9: (a)The $n_{acg,bot}$ node (see Fig. 2.5) transitions below 0V, settling towards $-Vdd/2$. Holding c to 0V is insufficient to ensure cutoff of M_c. The footer driver ties the gate of M_c to $n_{acg,bot}$, ensuring cutoff. (b)Footer driver schematic. The driver ties the gate of M_c to $n_{acg,bot}$ through M_5. M_4 is fed-back to the pull-down stack to avoid current backflow through $M_2 - M_3$.

2.3.2 Footer Driver circuit

The QRC_{footer}implementation employs a footer for efficient operation through full gate overdrive. However, when $V(clk) = 0$, the $V_{dd}/2$ voltage across the C_{acg} results in the footer drain voltage reaching $-V_{dd}/2$ (Figure 2.9). Using traditional drivers to set the gate voltage c to 0V is therefore insufficient, and results in current back-flow from the ground terminal which discharges C_{acg}. A footer driver is therefore designed to drive c to achieve full gate overdrive when on, and ensure cut-off when non-conducting. When the network is in resonant mode with $V(c_{pre}) = V_{dd}$, the driver drives c to V_{dd}, allowing M_c to conduct in the linear mode regardless of clock polarity. When clk is held at V_{dd} in conventional mode with $V(c_{pre}) = 0$, the drain voltage of M_c ($n_{acg,bot}$ transitions to $V_{dd}/2$, and the driver drives c to 0V. However, when $V(clk) = 0$ in conventional mode, the drain of M_c transitions to $-V_{dd}/2$, requiring that the driver set c to $-V_{dd}/2$ to ensure M_c is in cut-off.

We implemented a footer driver (Figure 2.9b) to efficiently enable this functionality. The pull-up network consisting of M_0 and M_1 sets c to V_{dd} during build-up and resonant transition. In conventional mode, disabling M_c when $V(clk_{PLL}) = V_{dd}$ similarly involves a conventional pull-down

through conducting devices M_2 and M_3. M_4 and M_5 are in cutoff and do not interfere in the action of the pull-down network. When $V(clk_{PLL}) = 0$ in conventional mode, $n_{acg,bot}$, the erstwhile drain of M_c is $-V_{dd}/2$, M_5 conducts to ensure that c is connected to n, ensuring $V_{gs} = 0$. M_4 is employed to connect the gate terminal of M_3 to n, thereby avoiding current back-flow in the pull-down stack. Thick oxide devices are required for M_4 and M_5 to withstand higher oxide stress when conducting. The higher V_{th} of these devices is offset by the increased gate overdrive of $3/2V_{dd} - V_{th}$ during conduction. M_1 and M_7 are protection devices employed to prevent oxide stress related degradation in M_0 and M_6 respectively.

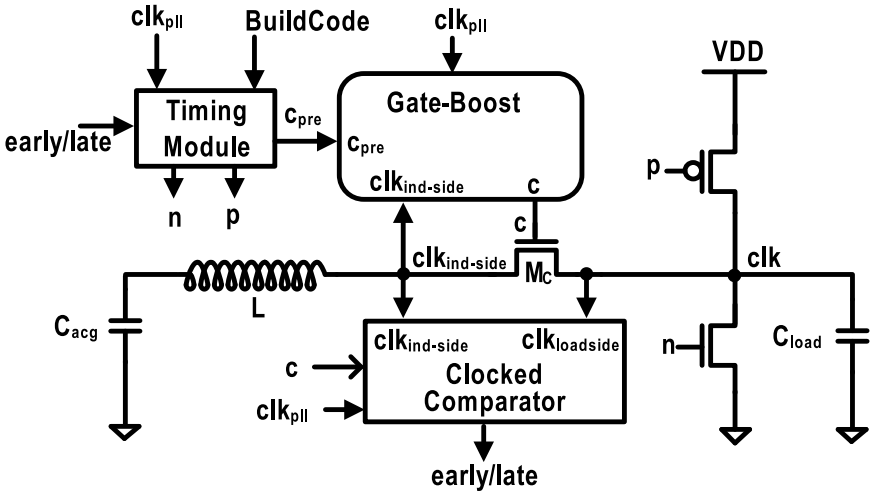


Figure 2.10: QRC_{pass} simplified schematic. M_c is employed as a Nmos pass-gate. Gate-overdrive is employed to ensure a uniform 1.2V gate overdrive throughout conduction.

2.4 QRC_{pass} **Design**

In this Section, we discuss circuits and architectures that constitute the QRC_{pass}design. The QRC_{pass}topology enables a single inductor to be shared between multiple clock load domains for a Single Inductor Multiple Output (SIMO) capability. This feature is promising for a variety of applications including driving multi-phase bridges of an Integrated Buck converter. Similar to

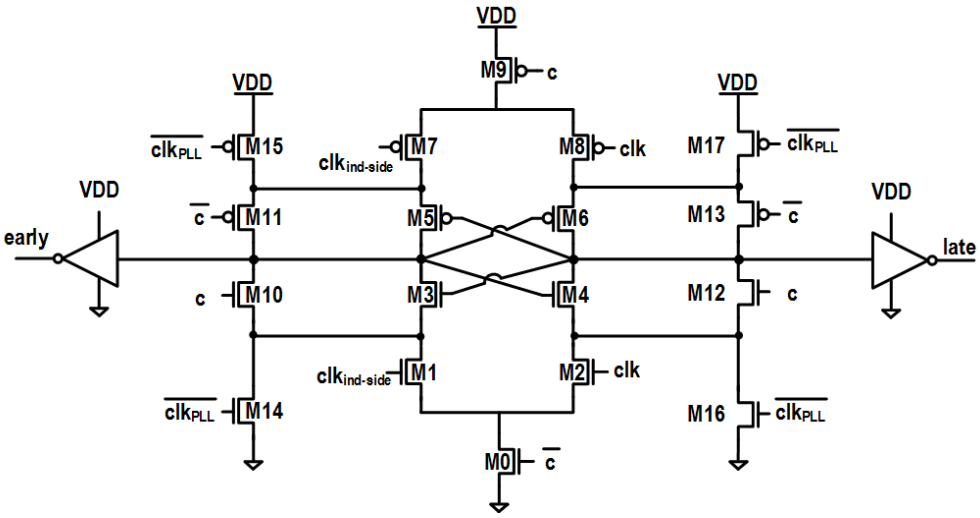


Figure 2.11: QRC_{pass} clocked comparator schematic. The comparator operates with alternating input common mode voltages of 0V and V_{dd}, twice every cycle.

QRC_{footer}, QRC_{pass}relies on the same *Timing Control* module for precise signal timing. In this section, we focus on modules that are distinct from the QRC_{footer}implementation, namely the sense-amplifier and the gate-overdrive circuits.

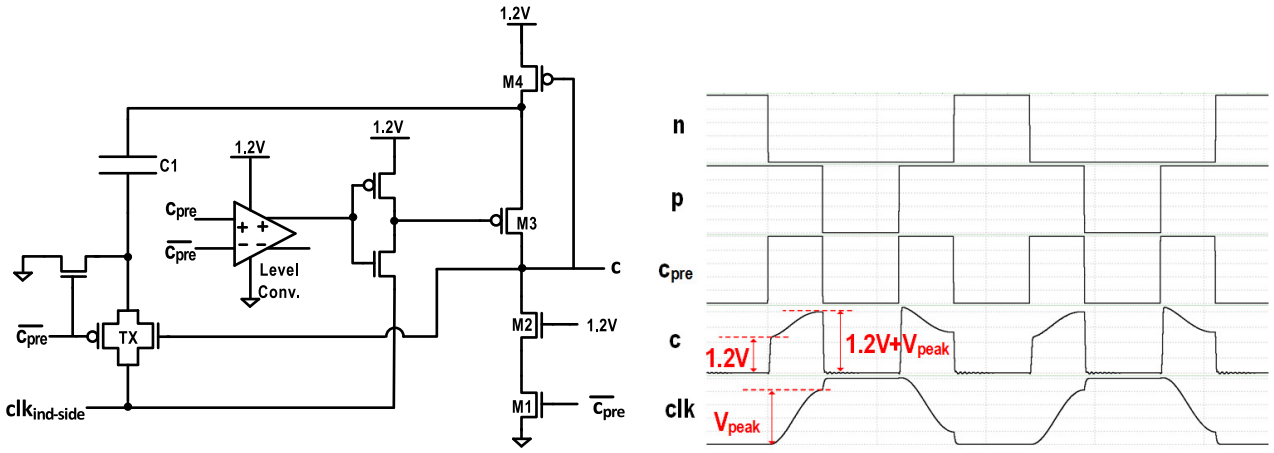
QRC_{pass}employs a gate-boosted Nmos pass-gate (M_c) to serve as the conduction switch (Figure 2.10), resulting in a set of current-sensing and gate-overdrive challenges distinct from those in QRC_{footer}. Unlike QRC_{footer}, where the sense-amplifier measures M_c source-drain polarity around a 0V common mode, QRC_{pass}requires polarity sensing with a common-mode of V_{dd} and 0V after the rising and falling edges of *clk* respectively. In addition, the $V_{dd}/2$ voltage across C_{acg} results in reduced gate overdrive for M_c.

2.4.1 Sense-amplifier Latch

Figure 2.11 shows the proposed sense amplifier (sense-amp) implemented for QRC_{pass}. In order to sample twice every cycle, with common mode voltages of 0 and V_{dd} respectively, the proposed sense-amp exhibits alternating operation of pull-down and pull-up sense-amplifier sections. When

$clk_{PLL} = 1$, the sense-amp amplifies the differential current drive in M_1 and M_2 (enabled by footer device M_0) to determine current flow direction in M_c. Devices $M_3 - M_6$ provide the positive feedback to resolve the differential current drive, $M_{10} - M_{13}$ serve to pre-charge sense-amp nodes during both clock transitions. Finally M_{15}, M_{17} (M_{14}, M_{16}) provide the necessary power (ground) connection to the sense-amp when $clk = 1$ ($clk = 0$). Devices $M_1 - M_8$ are up-sized, and employ non-minimum channel length devices to mitigate random mismatch between differential devices.

2.4.2 Gate Boosting



(a) (b)

Figure 2.12: (a)Schematic circuit for pass-gate boosting. The proposed topology provides a constant 1.2V V_{gs} to M_c while using only logic devices while maintaining reliability, (b)Pass-gate boosting simulation waveforms.

Placing M_c between the inductor and clk allows multiple domains to share a single inductor, but results in degraded gate overdrive. A standard 2X voltage doubler [50–52] applied to M_c provides the necessary gate overdrive but is not feasible due to the $2V_{dd}$ gate-source voltage that will result as clk approaches 0. We propose a gate boosting topology suited to the QRC_{pass}implementation (Figure 2.12a) that relies on boot-strapping the gate of M_c with the transition of $clk_{ind-side}$. The resulting

system provides full gate overdrive ($V_{dd} - V_{th}$) for M_c throughout the resonant clk transition between the supply rails. The proposed circuit ensures that c the gate signal provided to M_c is 0V when $c_{pre} = 0$. When $c_{pre} = 1$, the boosting circuit delivers a voltage of $1.2V + V_S$ (where V_S is the source voltage of M_c), consistently ensuring full gate-overdrive for M_c.

Figure 2.12b, illustrates the operation of the proposed circuit. In conventional mode ($c_{pre} = 0$), series devices M_1 and M_2 (a gate-oxide protection device) drive c to 0V, ensuring cutoff. The boosting capacitor (C1) pre-charges to 1.2V. In resonant mode, as $c_{pre} = 1$, turning M_c, TX conducts ($c_{pre} = 1$) connecting the bottom-plate of C to n_{Lside}. M_3 conducts, driving out to $V(n_{Lside}) + 1.2V$. This gate overdrive is maintained during the entire resonant transition, both rising and falling. Notably, while maintaining a 1.2V gate-source and gate-drain voltage during operation, the gate voltage safely transitions up to 2.4V (When V_{dd}=1.2V) while M_c conducts. The gate-oxide of M_c is protected by the conducting channel while conducting.

An off-chip 1.2V supply voltage was used to aid test and characterization of the proposed circuit. An integrated implementation of the supply using charge-pumps is relatively straightforward due to the low-current draw of the 1.2V supply. Alternatively, the 1.2V supply can be replaced with V_{dd}, resulting in a voltage doubler gate-oxide stress compliant design at the cost of reduced efficiency at low voltages.

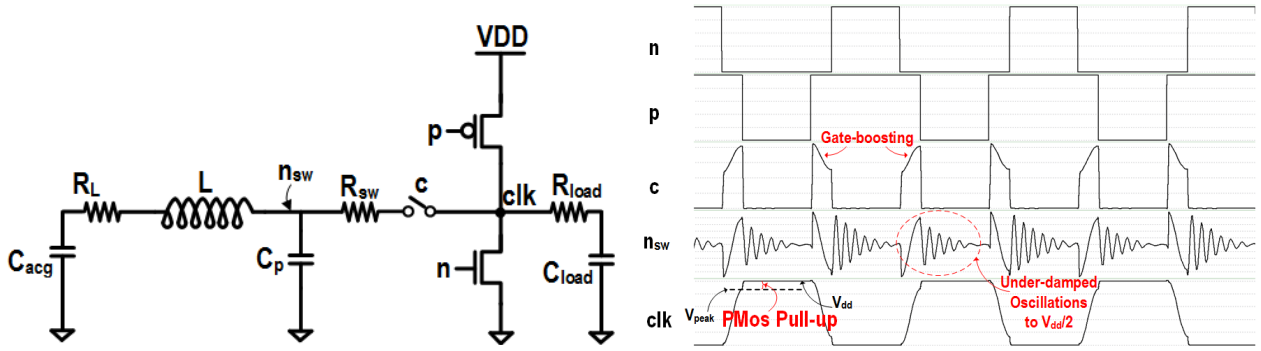
In QRC_{pass}, turning off of the pass gate causes an underdamped RLC oscillation at node n_{sw} as the drain capacitance of the pass-gate transitions from either V_{dd}or 0 to $\frac{V_{dd}}{2}$ (Figure 2.13b). This underdamped oscillation is unavoidable, but does not affect the reliability of devices in the QRC system.

2.5 Analysis

2.5.1 Energy Dissipation Analysis

In this section we present simulations and analytical models for QRC energy dissipation consisting of switching and conduction loss components. Simulations are used to examine aspects of QRC design that are either onerous or impossible to demonstrate through measurement of the test-chip.

Given the similarity of the analysis between $QRC_{pass}$and QRC_{footer}, we restrict our focus on the QRC_{pass}implementation for brevity. We use the simplified schematic in Figure 2.13a to analyze the energetics of Quasi-resonant Clocking.



(a) (b)

Figure 2.13: (a)Equivalent circuit used for QRC_{pass} energy dissipation analysis. (b)QRC_{pass} simulation waveforms relevant to energy dissipation.

The energy-per-cycle (EPC) dissipation of a resonant clocking system modeled as an equivalent R, L and C system can be approximated as ([35])

$$EPC_{res} = \frac{\pi}{4Q} CV_{dd}^2, \tag{2.1}$$

where Q is the system quality factor.

To analyze QRC energy dissipation, we start by noting that a lossless LC resonant clock transition is sinusoidal, with magnitude $V_{dd}/2$ around a DC voltage level $V_{dd}/2$. The total capacitance $C_{total} = C_{load} + C_p$ stores a charge of $C_{total}V_{dd}/2$, corresponding to an energy storage of $1/2 \cdot C_{total}(V_{dd}/2)^2$. In practice, C_{load} charges to a lower value, V_{peak} (Figure 2.13b), the difference accounting for the conduction losses in the system. V_{peak}, the voltage of clk at the end of its resonant transition, and

E_{cond}, the per-cycle conduction loss can be determined using known values of Q:

$$\begin{aligned} V_{peak} &= \frac{V_{dd}}{2}(1+e^{(\frac{-\pi}{\sqrt{4Q^2-1}})}) \\ E_{cond} &= (C_{load}+C_p)V_{dd}(V_{dd}-V_{peak}) \\ &= (C_{load}+C_p)V_{dd}[V_{dd}-\frac{V_{dd}}{2}(1+e^{\frac{-\pi}{\sqrt{4Q^2-1}}})]. \end{aligned} \tag{2.2}$$

An additional source of energy loss arises from conventional charge and discharge of parasitic capacitance, C_p on the inductor side of the switch. Twice every cycle, this capacitance transitions from $V_{dd}/2$ to the supply rails at the onset of the resonant transition and returns to the mid-rail voltage after the transition. The resulting energy dissipation, $E_{sw,parasitic}$ can be modeled as:

$$E_{sw,parasitic} = 4 \cdot \frac{1}{2} C_p \frac{V_{dd}^2}{4} = \frac{1}{2} C_p V_{dd}^2. \tag{2.3}$$

Similarly, the system incurs switching losses in the driver twice, once for each transition

$$E_{sw,driver} = 2 \cdot \frac{1}{\eta_{boost}} \cdot C_{Mc} \cdot (V_{dd}+1.2)^2, \tag{2.4}$$

where C_{Mc} is the capacitance of the M_c switch and η_{boost} is the efficiency of the gate boosting module. Adding Equations 2.2, 2.3 and 2.4 provides the expression for total losses per-cycle:

$$\begin{aligned} E_{tot} = (C_{load}+C_p)V_{dd}(V_{dd}-V_{peak}) + \frac{1}{2}C_p V_{dd}^2 \\ + \frac{2}{\eta_{boost}} \cdot C_{Mc} \cdot (V_{dd}+1.2)^2. \end{aligned} \tag{2.5}$$

The tradeoff between switching and conduction losses is apparent in M_c width selection (Figure 2.14): Increasing M_c width reduces resistance,and E_{cond}, at the expense of increased $E_{sw,driver}$. Including the resistive contribution of M_c in the I^2R losses in Equation 2.2, and using the resulting V_{peak} to obtain E_{tot}, the resulting analytical EPC results derived from Equations 2.3, 2.4 and 2.5 match well with simulation results (with parasitic capacitance).

Figure 2.14 shows the existance of an optimal M_c width, trading off switching andconduction losses in equation 2.5. As seen in Equation 2.2, increased resistance due to the clock distribution, the inductor, or reduced series switch width, degrades system Q, increasing conduction losses. On the other hand, larger switches increase switching losses. In our proposed implementation of

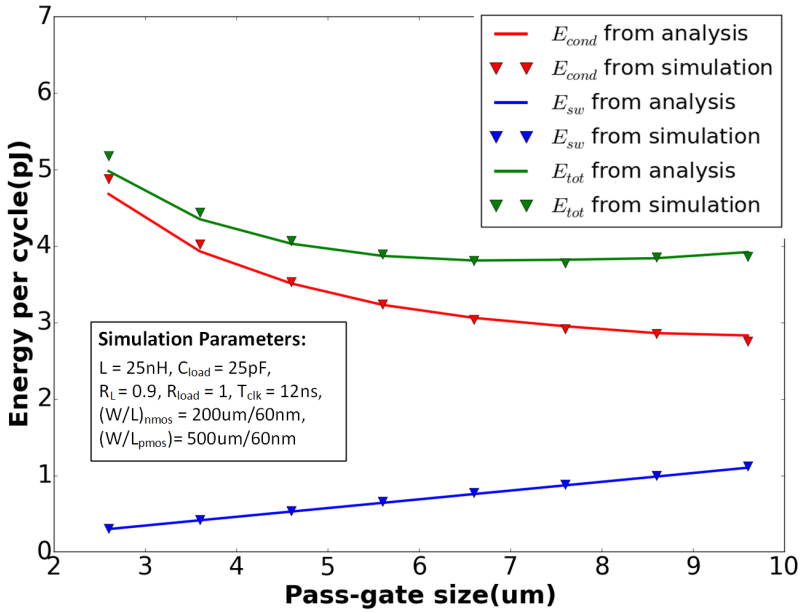


Figure 2.14: Comparison of post-layout simulations of QRC with analytical results from Equation 2.5.

$QRC_{pass}$and QRC_{footer}, the switch-width was designed to be tunable by having several banks of MOSFET in parallel. In post-silicon, a one time calibration was run to fix the optimal switch-width.

2.5.2 f_{max}

For QRC, clock frequency is modulated by inserting a hold-high/hold-low duration between the resonant transition and changing the duration of them. The maximum frequency is possible when $\tau_{holdhigh} = \tau_{holdlow} = \tau_{buildup} = 0$, which corresponds to a sinusoidal clock waveform operating at its natural frequency, f_0.

Increased f_0 delivers broader operating range but at the expense of reduced energy savings (Equation(2.1)):

$$\begin{aligned} EPC_{res} = \frac{\pi}{4Q} CV_{dd}^2 &= \frac{\pi}{4}(2\pi\ f_0\ RC)CV_{dd}^2 \\ &= EPC_{conv}(\frac{1}{2}\pi^2\ f_0\ RC). \end{aligned} \tag{2.6}$$

f_{max} can be calculated from the targeted energy savings:

$$Energy\_efficiency, \eta_{res} = 1 - \frac{EPC_{res}}{EPC_{conv}}, \qquad f_0 = \frac{2(1-\eta_{res})}{\pi^2 RC}. \tag{2.7}$$

From Equation(2.7) it can be seen that increasing f_0 degrades efficiency. This trend imposes a trade-off between energy-efficiency and maximum achievable frequency. In designs with extremely high clock loading (global clock distributions of microprocessors), multiple distributed inductors have been shown to be effective, with each inductor resonating with a portion of the total clock load ([26, 27]).

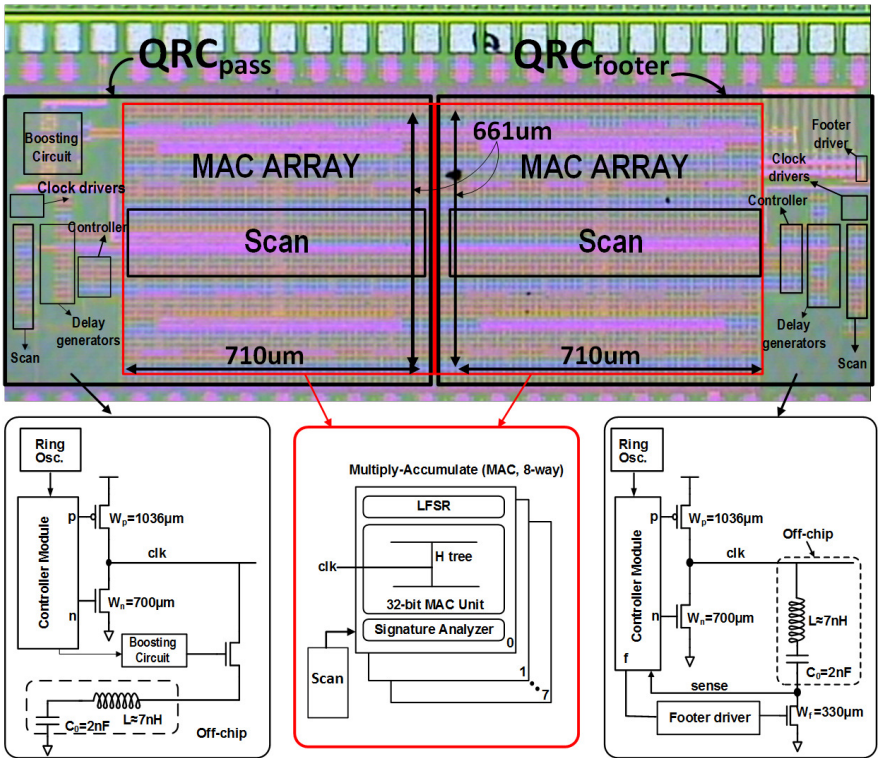


Figure 2.15: Dieshot of the QRC_{pass} and QRC_{foot} test-chip.

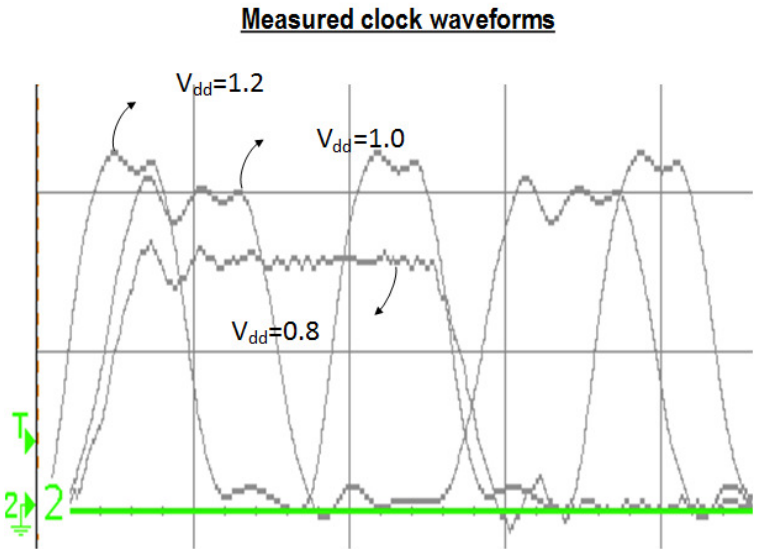


Figure 2.16: Oscilloscope traces of the QRC *clk* waveforms across V_{dd} and operating frequency.

2.6 Test-chip Architecture and Measurements

In this section, we present the architecture and implementation of the QRC test-chip featuring the $QRC_{footer}$and QRC_{pass}variants, and their test-chip measurements.

2.6.1 Test-chip Architecture

QRC_{footer} and QRC_{pass} are implemented separately as DVFS compliant clock distributions for an 8-way MAC array datapath in 65nm CMOS (Figure 4.7). Both datapaths employ Build-in Self Test (BIST) and scan-chains for functional and parametric test. The 8-way MAC was implemented using synthesis, auto-place and route (SAPR). The SAPR flow was augmented to produce a tapered H-Tree wire clock distribution driving a 3-level H-Tree driving a clock mesh on metal layers M4 and M5. The clocked comparators, footer-driver, charge-pump, delay circuits and the clock drivers were custom developed. Resonant clocks typically exhibit relatively lower skew due to low network resistance [26, 27]. The inherent slew degradation in resonant clocks however, causes significant PVT-induced skew in post-gater clock distributions, leading to potential setup and hold-

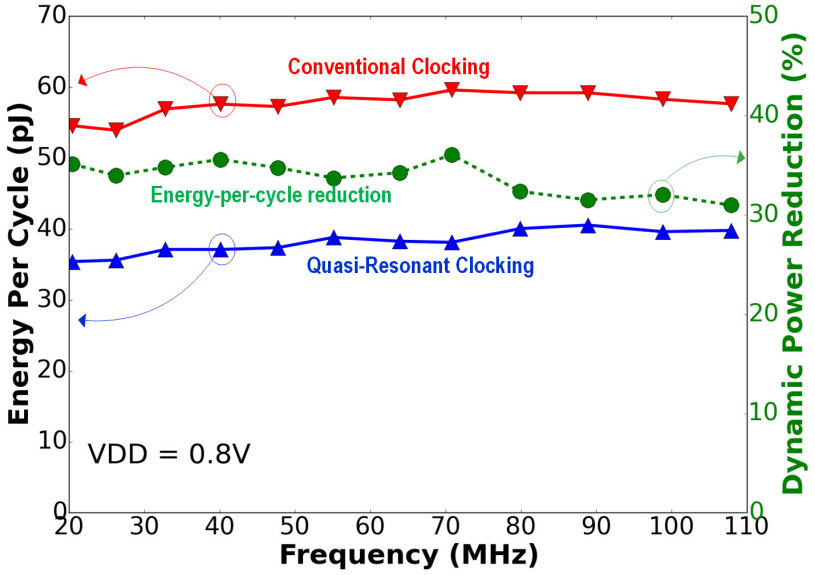


Figure 2.17: Energy-per-cycle measurements of QRC_{foot} at V_{dd}=0.8V vs. operating frequency.

time degradation. Of the several techniques available to mitigate this degradation [26, 27, 35], we adopt latch-based design [35] approach. C_{acg} is implemented as an off-chip electrolytic capacitor for both variants. This test-chip used an off-chip inductor in the implementation of $QRC_{footer}$and QRC_{pass}. While on-chip inductors can be used for QRC, the choice of off-chip inductors was driven by the desire to explore the impact of inductor selection for slew-rate and efficiency trade-offs, and be able to readily capture resulting QRC clock waveforms (Figure 2.16). Notably, use of on-chip inductors does not significantly degrade system efficiency because M_c losses largely dominate system Q ($\approx$ 1.3 for QRC_{pass}, $\approx$ 1.2-1.8 for QRC_{footer}). Robust system operation was ensured through the design of the resonant-clock latch-based design methodology [35]. Runtime voltage-frequency scaling was performed by using a ring-oscillator powered by the logic supply as a clock source. The energy measurements reported in this paper include the dissipation from all constituent QRC sub-systems. In $QRC_{footer}$and QRC_{pass}, the value of C_{acg} and L are 5nF and 7nH respectively. The extracted value of capacitance, C_{load} is approximately 180pF.

As mentioned in Section 5.2, an energy-optimal selection of $T_{BLD,p}/T_{BLD,n}$ exists. In the

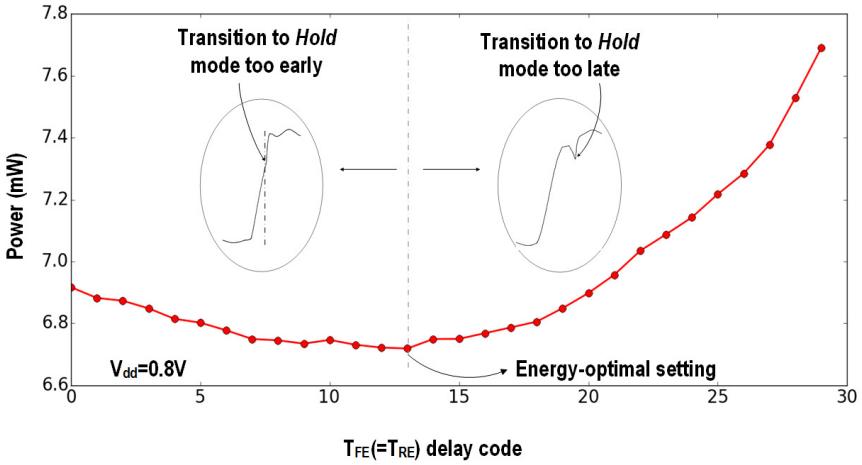


Figure 2.18: Energy-per-cycle measurements of QRC_{foot} over a range of delay values for c. The autonomous timing control module was overridden to enable the delay code sweep.

$QRC_{pass}$and QRC_{footer}implementations, $T_{BLD,p}/T_{BLD,n}$ was implemented with tunable delay chains. After a one-time calibration, the optimum $T_{BLD,p}/T_{BLD,n}$ duration was found to be 80ps at 1V.

2.6.2 QRC_{footer} and QRC_{pass} Test-chip Measurements

Figure 2.17 shows the uniform energy dissipation per-cycle (EPC) for the QRC_{footer} across a range of operating frequencies with V_{dd} fixed at 0.8V. Conventional clocking dissipation was conservatively estimated by downsizing the programmable clock driver to allow EPC measurements at iso-clock slew. This comparison also ignores any electromigration challenges associated with conventionally driving the entire clock load through a central buffer, and ignores the clock power dissipation that would be incurred in distributed clock buffers. The resulting energy efficiency – achieved QRC_{footer} energy reduction compared to conventional clocking is also shown in Figure 2.17. Measurements at 0.8V indicate EPC savings of 32-39% compared to the traditional clocking. Since gate-overdrive was not employed for M_c in QRC_{footer}, higher voltages yield improved energy efficiency due to the more significant reduction in conduction losses.

To demonstrate importance of precisely timed gate control of M_c, a manual override was used to

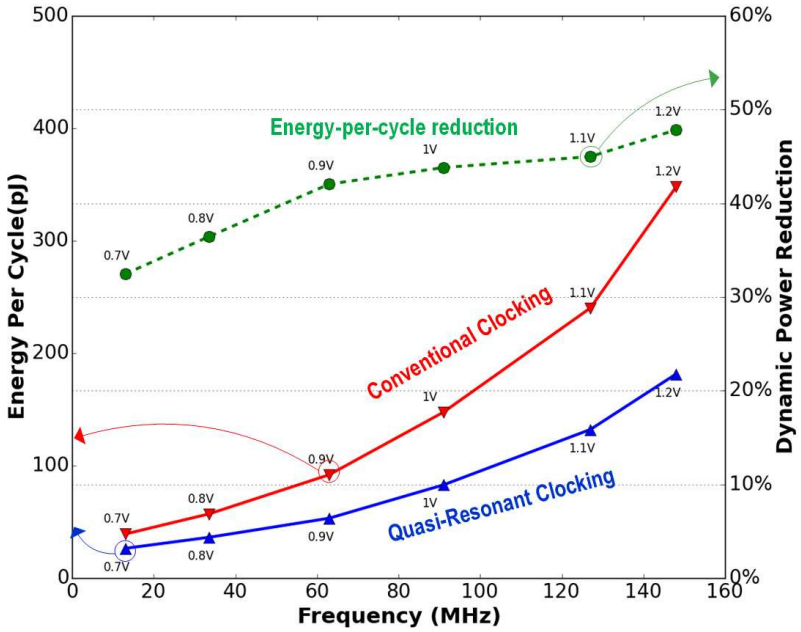


Figure 2.19: Energy-per-cycle measurements of QRC_{foot} operating under a Dynamic Voltage-Frequency Scaling range of 0.7V–1.2V.

sweep the delay codes for the t_{rise} and t_{fall} delay chains (Figure 2.6) across a range of values and the resulting EPC reduction was measured. Figure 2.18 demonstrates the impact of M_c timing on energy-efficiency. Low delay codes result in premature de-assertion of M_c, while late delay codes result in late M_c disconnection, both resulting in excessive conduction losses.

The energy efficiency of QRC_{footer}under DVFS is shown in Figure 2.19. The core supply voltage was swept from 1.2V to 0.7V, and the on-chip core-supply powered ring oscillator provided the accompanying frequency scaling. QRC_{footer} achieves an energy reduction of 32% -47%(across 0.7V–1.2V). QRC does not fundamentally limit scaling $V_{dd}$to 0.7V or set a lower bound on the operating frequency in any way, as verified through simulation. The test-chip lower bounds in frequency and voltage in this test chip were the result of a design oversight pertaining to the use of level converters.

Figure 2.20 shows measured EPC vs. frequency at V_{dd}=0.8V. Figure 2.21 shows EPC dissipation of QRC_{pass}across a V_{dd} range of 0.7V-1.2V. Also included for reference are the conventional EPC

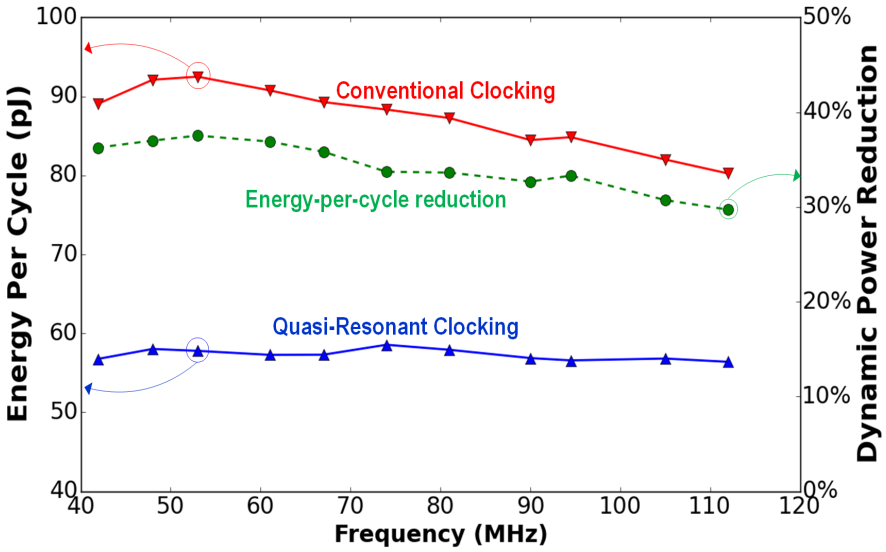


Figure 2.20: Energy-per-cycle measurements of QRC_{pass} at V_{dd}=0.7V vs. operating frequency.

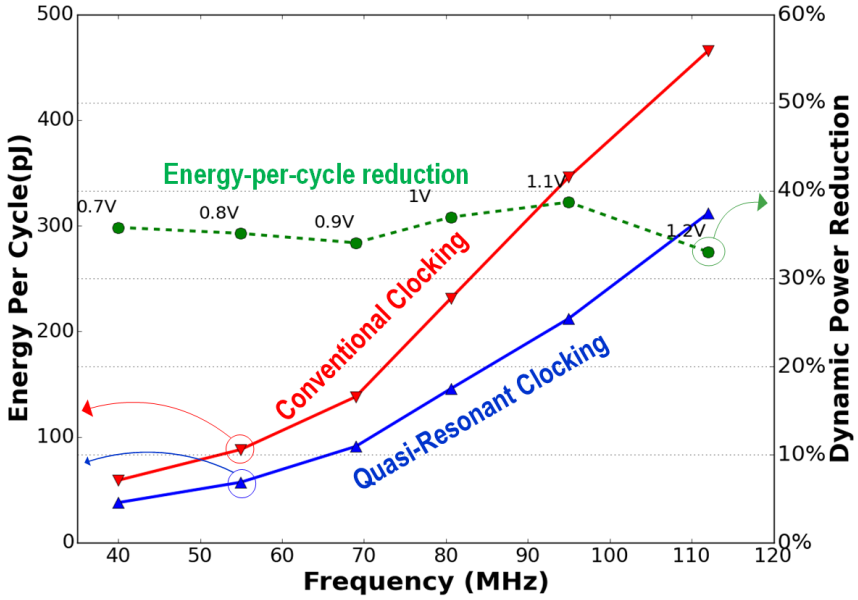


Figure 2.21: Energy-per-cycle measurements of QRC_{pass} operating under a Dynamic Voltage-Frequency Scaling range of 0.7V–1.2V.

	JSSC'13[6]	ISSCC'14[7]	ISSCC'13[20]	This work	
				QRC_{footer}	QRC_{pass}
Process Technology	32nm CMOS	22nm SOI	40nm CMOS	65nm CMOS	65nm CMOS
System Resonant frequency	3.3GHz	≈ 3.1GHz,4GHz (two modes)	Always Resonant	Always Resonant	Always Resonant
Voltage-Frequency scalable resonance	No	No	No	Yes	Yes
Voltage Range	1.0V-1.2V	0.75-1.05V	0.37V*	0.7V-1.2V	0.7V-1.2V
Frequency Range	2.4-4GHz	2.5GHz - 5GHz	DC - 0.98MHz	DC – 152MHz	DC – 132MHz
Duty Cycle Control	Limited	Limited	No	Yes	Yes
Dynamic Power Reduction	15%-30%	25%-33%	36%	32%-47%⁺	34%-38%⁺
Inductor	On-Chip (1nH-3nH each)	On Chip (0.3nH-2.5nH each)	Off-chip 7µH	Off-chip ~7nH	Off-chip ~7nH
SIMO-capable	No	No	No	No	Yes

Figure 2.22: Comparison of the proposed QRC architecture and test-chip measurements with related works.

numbers at corresponding voltages. Compared to traditional resonant clocking, QRC_{pass}efficiency is relatively insensitive to frequency. Consistent with simulations, measured EPC values are lower than conventional clocking, but higher than QRC_{footer}. Increased QRC_{pass}EPC over $QRC_{footer}$is largely attributable to the additional switching losses in the boosting circuit. Notably, QRC_{pass}efficiency remains approximately constant across voltage and frequency scaling, a trend enabled by gate-boosting M_c.

Figure 2.22 compares key design metrics of both QRC variants with related works. Overall, QRC demonstrates scaling across a wide voltage-frequency range. Near-uniform efficiency is achieved over a broad range of frequencies. Another important capability of QRC is its ability to achieve a broad duty-cycle range and the ability to service multiple clock domains using a single inductor.

2.7 Conclusion

We presented the Quasi-resonant Clocking (QRC) architecture and demonstrated the first-ever continuous voltage-frequency scalable resonant clocking system. Autonomous timing control of QRC circuits plays a central role in enabling robust, efficient QRC operation. We presented test-chip implementation of two variants of QRC, implemented in 65nm CMOS. Energy-efficiency measurements from the test-chip validate the ability of QRC to achieve uniform efficiency across frequency scaling. DVFS measurements in the 0.7V-1.2V range indicate Energy-per-Cycle reduction of 32%–47%, and 34%–38% using QRC_{footer} and QRC_{pass} respectively. The ability to achieve 0-cycle latency duty-cycle control offers a promising opportunity for QRC to be employed in a broader range of applications driving large capacitive loads beyond system clocking.

Chapter 3

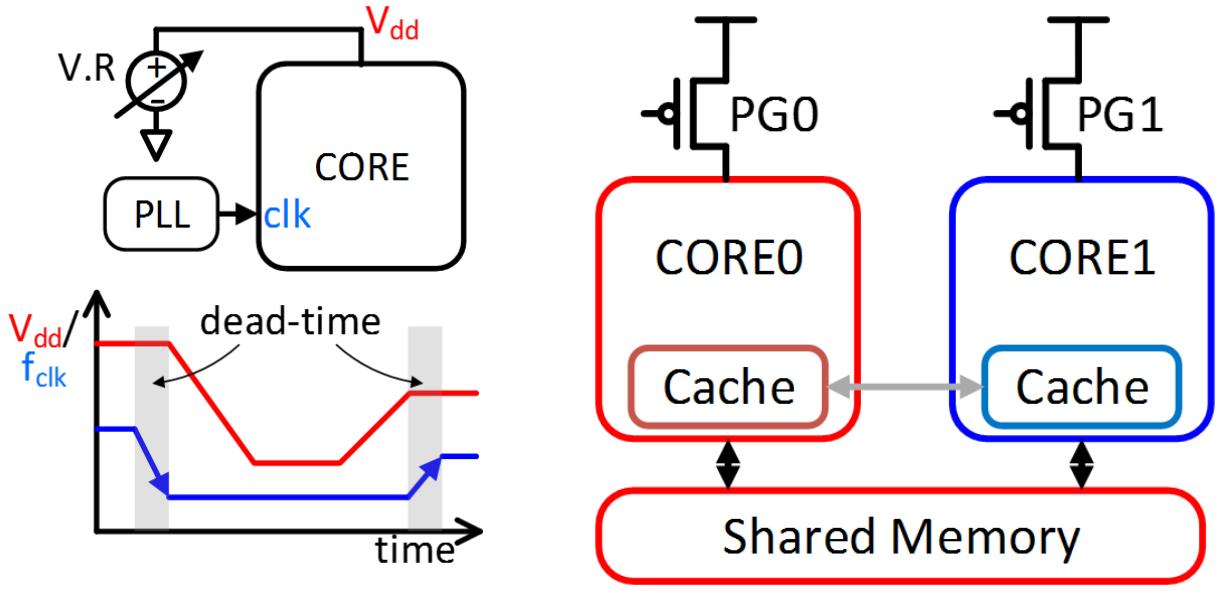
APPLYING COMPUTATIONAL CONTROL IN ACCELERATING PLL LOCK-TIME : COMPUTATIONAL-LOCKING PLL

PLLs play a central role in enabling synchronous communication and computation in digital systems by providing a frequency-scalable timing-reference, stable across PVT variation [49]. All-digital PLLs (ADPLLs) in particular, provide several advantages, including improved scalability across process technology nodes, and portability between designs. ADPLLs have been widely adopted in system-clocking [53–56] and more recently, even data-transfer [57–61] applications.

The sustained focus on energy-efficiency in digital systems continues to drive advances in low-power techniques and design methodologies. Two salient low-power techniques, that also have implications on PLL design, are Dynamic Voltage Frequency Scaling (DVFS) and Power Gating (Figure 3.1). DVFS involves varying the system clock frequency (f_{clk}) through PLL re-lock to meet performance requirements, providing runtime supply-voltage (V_{dd}) scaling opportunities for low-energy design [62–64]. During each DVFS event, the duration of time during PLL re-lock corresponds to "dead-time", where the processor does not operate to avoid circuit timing violations. Power-gating on the other hand allows disconnecting a module from the power supply through a header switch to drastically reduce leakage power [64]. Re-enabling the module however, requires the PLL to lock from a power-off state (cold-start).

Increasingly frequent and rapid V_{dd} transitions enabled by integrated voltage regulation therefore motivate accelerating re-lock time.

Recent trends in Integrated Voltage Regulation (IVR) involve more frequent DVFS transitions and more rapid V_{dd} changes (Fig. 4.21). PLL re-lock latencies–previously hidden by the relatively long V_{dd} transition times associated with off-chip Power Management Units (PMUs)–now need to be significantly lowered to avoid impacting system performance. This motivation is particularly



(a) (b)

Figure 3.1: (a)V_{dd} and f_{clk} scaling during DVFS events. System operation is typically stalled during PLL re-lock to the target frequency. (b) Power-gated cores with shared memory and local caches. Turning on a core often involves re-locking the PLL from its power-off state

acute for domains driven by Low-Droput Regulators (LDOs). In addition, the combination of power-gating and the prevalence of multi-core and heterogeneous systems has also motivated PLL T_{lock} reduction. Cold-start lock-times contribute to power-domain wake-up delay. For latency-sensitive tasks, higher T_{lock} degrades system performance. For instance, multi-core systems with individually power-gated cores (Figure 3.1b) need to rapidly wake-up to service cache-coherency requests [65] from active cores.

Existing PLLs used in system clocking feature T_{lock} values amounting to hundreds of reference-clock ($REFCLK$) cycles. These longer lock-times lead to considerable dead-times during DVFS transitions and wake-up latencies during wake-up [66], impacting system performance. One approach to reduce wait-times involves employing two PLLs operating in ping-pong fashion [67], incurring significant area and power overhead. More direct efforts toward lock-time reduction have

made progress over the years. However significant advances in T_{lock} reduction are limited due to a reliance on the traditional PLL model for design and optimization, which treats the PLL as a linear time-invariant (LTI) system and assumes a linear relationship between phase-error and time-delay between clocks even during lock. While these simplifications, discussed in more detail in Section 3.1, are adequate for analyzing the performance of locked-PLLs, they fall short of accurately modeling locking behaviour. Achieving significant advances in T_{lock} will require a different, more accurate analysis and design model for PLLs.

Several efforts in T_{lock} reduction for PLLs have been reported in the literature [68–72]. While these techniques are often well-tailored for their specific use-case, they are either not suitable for system clocking [68], remain vulnerable to PVT variation [73], degrade steady-state performance in the form of excessive jitter [69], or do not support large changes in frequency divider ratios to avoid cycle-slipping limitations [70, 74]. Most importantly, to the best of our knowledge, although exhaustive statistical characterization of T_{lock} is common (and required) in production design, availability of statistical information across multiple lock iterations has been wanting. In addition T_{lock} variability information across multiple parts and under variations in supply-voltage and temperature has not been available in the literature.

In this work, we propose Computational Locking (C-lock) to substantially accelerate PLL-lock. The technique departs from traditional assumptions of PLL linearity, capture-range, and phase-time equivalence. C-lock adopts a runtime-solver approach that not only avoids the performance-degrading assumptions of traditional PLLs, but is also readily amenable to techniques that adapt to PVT variation, and avoid cycle-slipping during lock. Acquiring lock using C-lock involves relying on a digital "solver" module to iteratively perform runtime resolution of a system of equations constructed to accurately model PLL behaviour. The solver produces a sequence of Digitally Controlled Oscillator (DCO) codes each $REFCLK$ cycle until lock-acquisition. The operation performed by the solver resembles evaluation of the root of a function iteratively, using a gradient-descent approach. The solver is used only during lock. Once locked, the solver is *seamlessly* detached from the PLL and a simpler (traditional) digital loop filter (DLF)-based controller takes control of the loop. In this manner, lock acceleration does not interfere with steady-state PLL

operation.

We demonstrate C-lock by retrofitting a solver into an integer-N ADPLL for system-clocking in a 65nm standard CMOS technology. While we believe that the C-lock technique is equally applicable to fractional-N PLLs, the demonstration described in this paper starts from this simpler implementation. Given the statistical nature of lock-times, we obtained T_{lock} measurements from over 50,000 iterations of re-lock and cold-start across multiple parts, $\pm 5\%$ V_{dd} variation, and 0°C–90°C temperature variation. An integrated built-in self test (BIST) module obtains lock-time measurements, snapshots of TDC code traces during lock acquisition, and intermediate solver results for a detailed examination of the lock process. C-lock measurements indicate the lowest PVT-robust T_{lock} reported for both re-lock ($12T_{REFCLK}$) or cold-start ($16T_{REFCLK}$) [17].

The chapter is organized as follows: In section 3.1 we provide an overview of traditional PLLs and discuss the non-idealities that result in excessive locktime. In Section 5.2, we elaborate upon the key idea of C-lock, its top-level architecture, and explain its detailed operation while acquiring lock. Section 3.3 presents detailed phase-frequency update equations, derived in this work to enable C-lock. Implementation details of the DCO, TDC, solver module, and the Built-in Self Test (BIST) module are described in Section 3.4. Test-chip architecture and measurements are presented in Section 3.5.

3.1 Limitations of the traditional PLL model

In this section, we identify simplifying assumptions that are made in traditional PLL design, and examine how these simplifications contribute to large lock times.

Traditional analysis and design models adopt a linear, time-invariant (LTI) model for the ADPLL, enabling a discrete-time (z-domain) representation [75], sampled at a rate determined by *REFCLK*. Figure 4.1 shows the block diagram of a traditional discrete-time ADPLL model. A phase-frequency detector (PFD) and time-to-digital converter (TDC) together quantify the phase difference between *REFCLK* (Φ_{REF}) and the divided DCO clock, *FBCLK* clock (Φ_{FBCLK}). The phase error is processed by a digital loop filter (DLF) before producing the required DCO code for the next *REFCLK* cycle. The output clock is divided and fed back as *FBCLK* to the PFD-TDC for comparison with *REFCLK*.

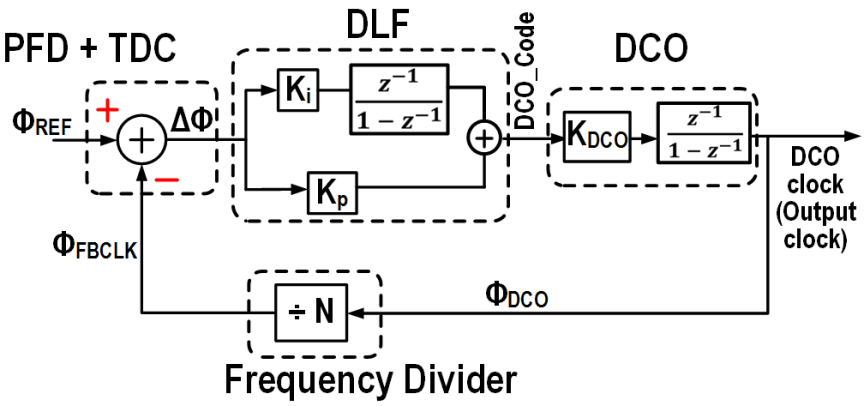


Figure 3.2: Traditional ADPLL block diagram with discrete-time (z-domain) representation

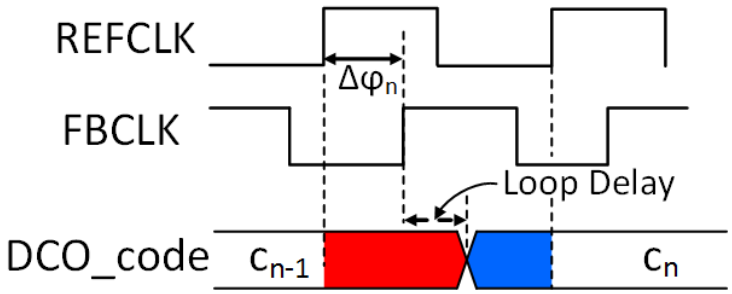


Figure 3.3: Typical ADPLL waveforms during lock. Delayed application of the DCO code due to TDC latency and non-zero phase error leads to a weighted-average application of DCO codes. The dependence on the proportion of c_{n-1} and c_n on phase-error leads to significant non-linearity

The z-domain model makes several simplifying assumptions about the PLL that, while justifiable for analyzing tracking performance (when the PLL is locked), significantly hinder further advances in T_{lock} performance. These assumptions are that: (1) the PLL is linear, without finite-PFD capture range limitations; (2) the time-difference between *REFCLK* and Φ_{DCO} is a good estimator for phase-error, allowing the TDC output to be interpreted as phase-error; (3) the DLF produces the resulting DCO code which is to be applied with zero-latency; (4) the TDC quantizes phase-error with zero-latency; (5) the quantized phase error is available at each rise edge of *REFCLK*, even if a *FBCLK* has not even arrived, and consequently (6) the PLL applies the DCO code to be applied for

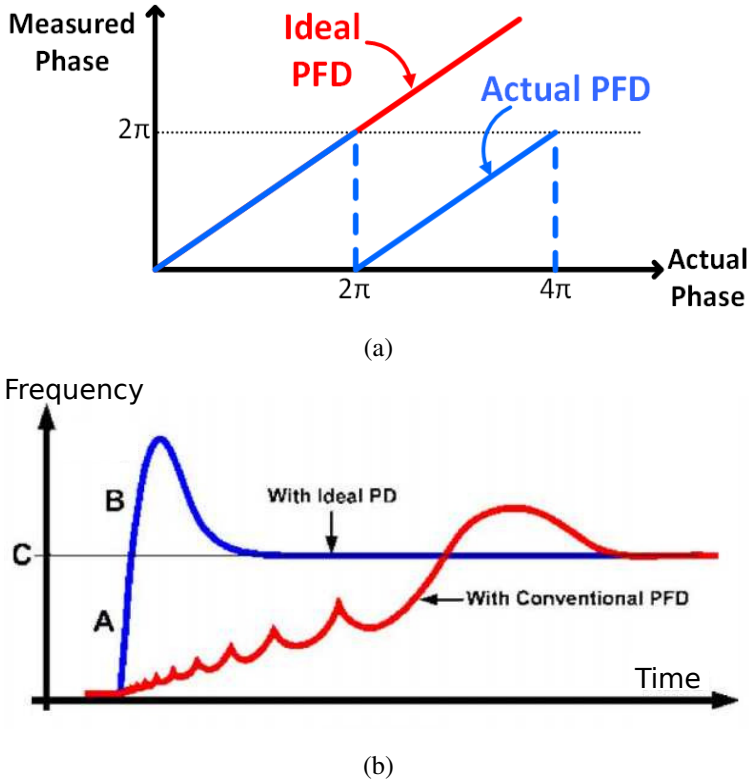


Figure 3.4: (a) Transfer curve of PFD, (b) Impact of cycle-slipping on T_{lock} [76]

a constant duration that lasts entire *REFCLK* cycle.

The above assumptions don't hold true however, particularly when modeling PLL lock. Quantizing phase-error between Φ_{REF} and Φ_{FBCLK} incurs TDC delay that can be a significant fraction of T_{REFCLK}, especially in system clocking applications when the core operates at a lower frequency (low N). Similarly, the DLF must incur computational latency in producing the output DCO code. Furthermore, unlike under tracking conditions, arrival times Φ_{REF} and Φ_{FBCLK} are not restricted to within a small fraction of T_{REFCLK}. A combination of these effects result in a latency in producing the DCO code to be applied in the current cycle. Figure 3.3 shows a typical *REFCLK* and *FBCLK* scenario during lock-acquisition. The effective DCO code, applied over a *REFCLK* cycle, is a weighted average of the DCO code in the current and prior cycles (c_n and c_{n-1}) respectively. The weighing factor depends on the latency in computing c_n, and on the time-varying arrival time

difference between $REFCLK$ and $FBCLK$. In contrast, traditional models assume application of c_n for the complete cycle, resulting in considerable error.

The finite capture range of PFD-TDCs significantly limit the linearity of PLLs (Figure 3.4a). By construction, PFDs detect time-delays with a limited range of T_{REFCLK} (or equivalently 2π). Errors exceeding this range are simply calculated in a modulo 2π fashion (Figure 3.4a). However, PLLs must acquire frequency or phase lock regardless of initial frequency or phase error. Initial frequency errors in particular, often lead to phase-errors that easily exceed the range of the PFD-TDC, leading to well-known cycle-slipping behaviour (Figure 3.4). This cycle-slipping significantly degrades PLL lock trajectory and contributes to increased lock-time. PLL designs using counter-based TDCs to avoid the cycle-slipping limitation have been previously demonstrated [74, 77]. These PLLs do not have the problem of cycle-slipping. However, combining the cycle count with the TDC code without error requires tracking the PVT induced variation in the TDC code that corresponding to one DCO cycle during lock acquisition. Any error in calibration contributes to longer lock-time. We chose to avoid this complexity.

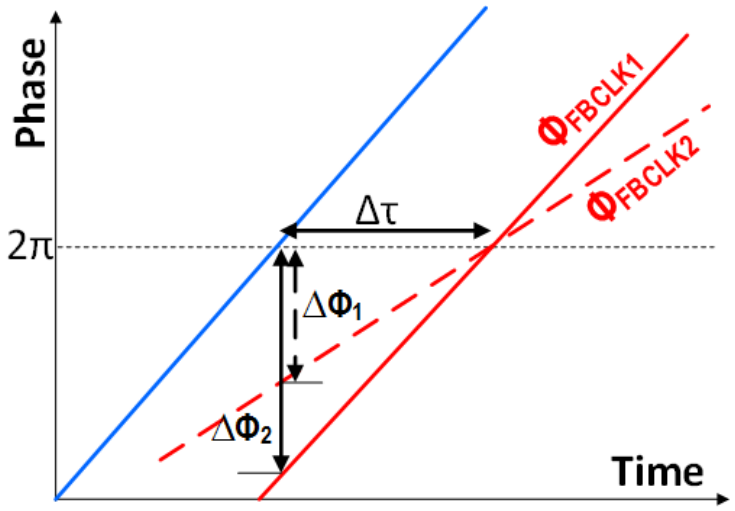


Figure 3.5: Identical time-delay errors can correspond to different phase errors depending on DCO frequency for non-frequency locked systems

Phase-time equivalence is another significant simplification employed by traditional PLL models. Compensator design relies on the use of phase as the measurement and control variable (Figure 4.1). The PFD-TDC quantizes time-delay error between *REFCLK* and *FBCLK*. Phase-time equivalence assumes a linear relationship between the two to simplify analysis. However, this linear relationship between time-delay and phase-error only holds valid for PLLs that are already frequency locked. The actual correspondence between measured time-delay and the corresponding phase-error in a PLL that is not frequency locked, depends on the (unknown and transient) DCO frequency during each cycle. Figure 3.5 illustrates this observation. A given TDC code ($\Delta\tau$) corresponds to different phase errors ($\Delta\phi_1$ and $\Delta\phi_2$) depending on the frequency of the DCO clock. Assuming phase-time equivalence, while reasonable for locked-PLLs, therefore holds less validity for lock acquisition. A more precise relationship between phase and time errors is described in Appendix 4.6. We summarize the result below:

$$\Delta\Phi_n = \frac{\Delta\tau_n}{T_{REFCLK} + \Delta\tau_n - \Delta\tau_{n-1}}, \tag{3.1}$$

where $\Delta\Phi_n$ is the phase error (normalized to 2π), $\Delta\tau_n$ is the time-delay between *REFCLK* and Φ_{FBCLK}, $\Delta\tau_{n-1}$ is the time-delay in the previous cycle and T_{REFCLK} is the time period of *REFCLK*. Note that if the PLL is frequency-locked $\Delta\tau_n$ and $\Delta\tau_{n-1}$ are equal, leading to the more familiar expression: $\Delta\Phi_n = {}^{\Delta\tau_n}/_{T_{REFCLK}}$.

Another source of large prevailing T_{lock} in PLLs is the guard-band required during loop compensation to account for PVT variation. PVT variation significantly affects the gains of the DCO (particularly for ring-based topologies) and TDC. This change in gain subsequently alters loop gain and the resulting closed-loop pole locations. While calibration can address performance degradation due to process variation, temperature variation continues to pose a problem for PLL lock characteristics, particularly for PLLs that rely on low-gate overdrive circuits for wide dynamic range.

The combined effects of delayed DCO code update, phase-time non-equivalence, cycle slipping, and PVT variation contribute significant inaccuracies in the traditional PLL model, and hinder efforts for rapid lock.

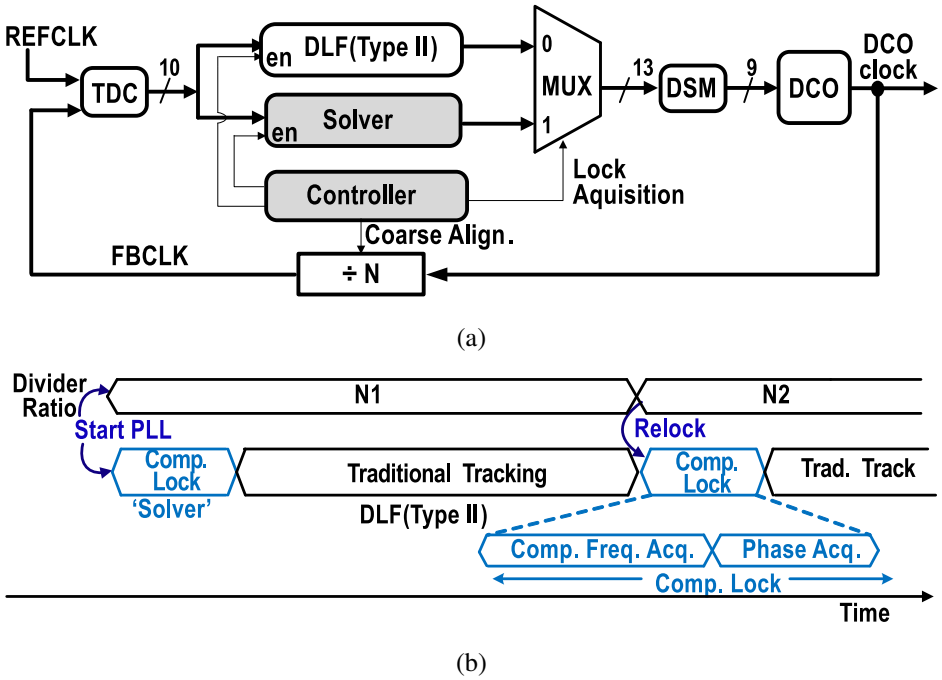


Figure 3.6: (a) Top level architecture, (b) Operation of the proposed PLL

3.2 Computational Locking: Overview

Figure 3.6a shows a block diagram of a PLL employing C-lock. The *Solver* module, which controls the PLL during lock acquisition, is placed in parallel with DLF module. At the onset of 'cold-start' or a frequency change (re-lock), the *controller* transfers control of the loop from the DLF to the *Solver* (Figure 3.6b). The *Solver* is tasked with computing the sequence of DCO codes that will result in phase-lock. A digital delta-sigma modulator (DSM), placed between the DCO and the *Solver* allows for finer DCO resolution through dithering. Once phase-lock is achieved, the controller is tasked with transferring loop control back to the DLF module for type-II control *seamlessly*, without causing transient phase-error (Figure 3.6b). Although more dissipative than the DLF, the *Solver* is gated-off after each (infrequent) cold-start or re-lock event, minimizing its contribution to overall power dissipation.

C-lock occurs in three stages: re-snap, frequency-acquisition, phase-acquisition. Figure 3.7

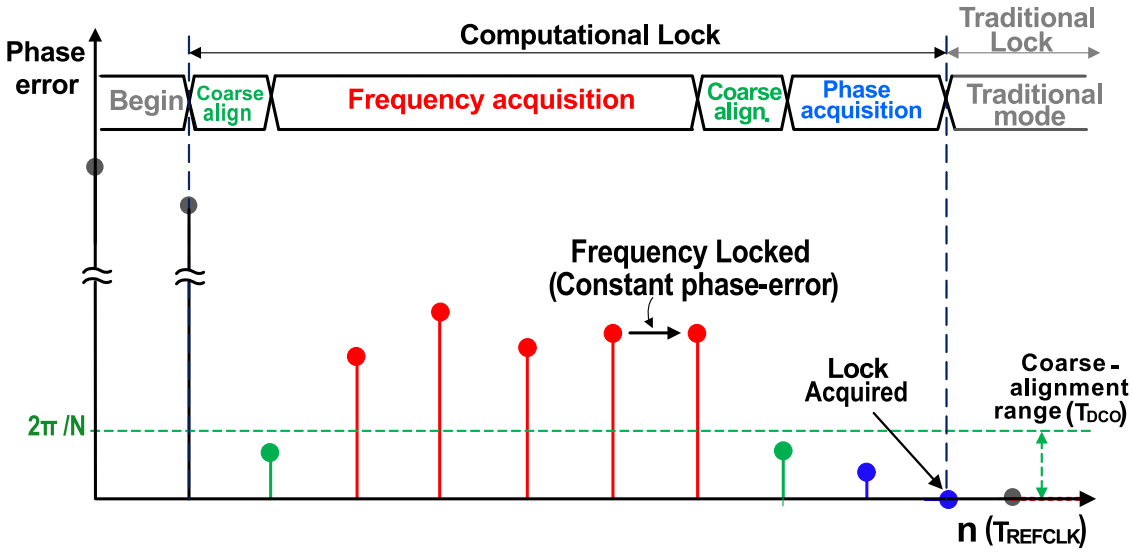


Figure 3.7: Computation Locking steps: Phase error is shown at different steps

provides a graphical representation of phase error during a computational lock process for a cold-start. C-lock begins with a coarse alignment of the *REFCLK* and *FBCLK* to within a single DCO clock cycle ($2\pi/N$). This coarse-alignment is performed in the *re-snap* phase (described in additional detail in Section 3.3.1).

After re-snap the *Solver* begins frequency acquisition, iteratively solving for the DCO code required for frequency lock (Figure 3.7). The iterative approach adopted by the *Solver* is reminiscent of gradient-descent, and incorporates solving accurate phase-frequency update equations that describe the PLL at runtime. The *Solver* assumes a nominal loop-gain, which depends on PVT susceptible DCO and TDC gains. The effects of the inaccuracy of this nominal gain value are mitigated during subsequent iterations of the *Solver* operation, allowing the approach to adjust for runtime PVT variation. Determination of frequency-lock is based on achieving an operating frequency that lies within a pre-defined frequency error band around the target frequency.

After frequency-acquisition, the *Solver* first performs another re-snap to limit the phase error accumulated during frequency-acquisition, before transitioning to phase-acquisition (Figure 3.7). Phase acquisition is tasked with rapidly eliminating the residual phase-error using a sequence of

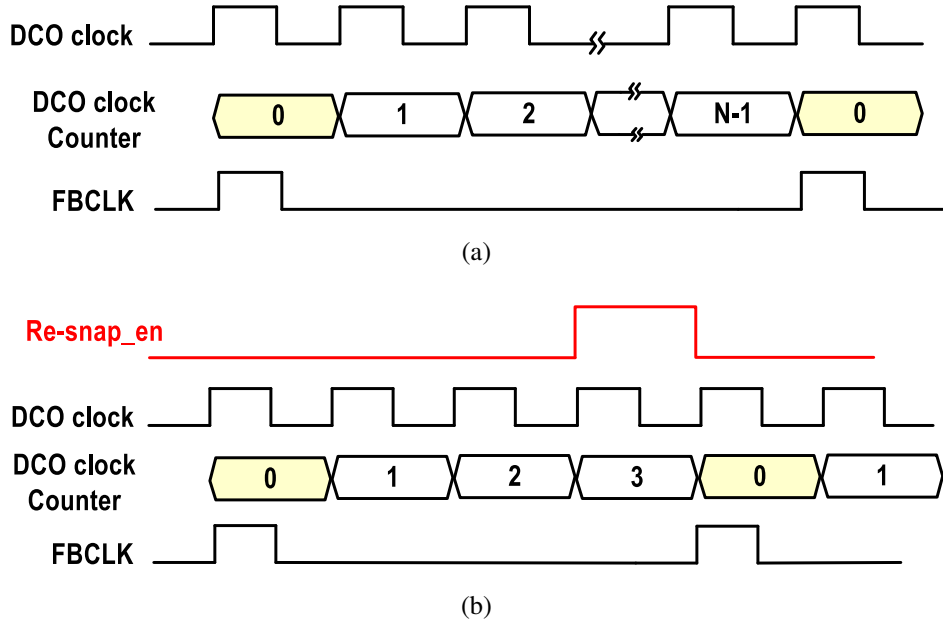


Figure 3.8: (a) Frequency divider operation under normal conditions, (b) Re-snap operation: Counter-reset and *FBCLK* generation triggered by *REFCLK* (simplified view, not accounting for retiming delay)

'frequency bumps': short pulses of frequency deviations away from the target frequency to overcome the remaining phase difference. Once the observed phase error is within the tolerance limit required for the application, the system is considered to be phase-locked. The *Solver* finally provides the DLF with the accumulated code to be stored in the integral portion of its controller to ensure seamless transition to traditional phase-tracking.

3.3 Computational Locking: Details

This section provides additional details on the operation of the three key operations that are performed by the *Solver* to enable C-lock.

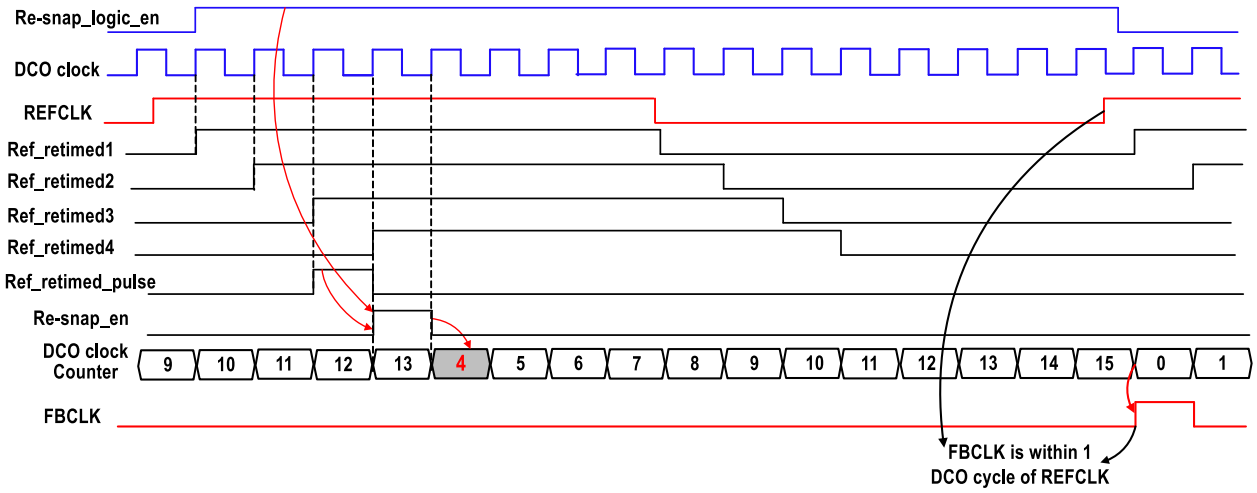


Figure 3.9: Managing clock domain crossing during Re-snap (N=16): waveform of *REFCLK*, DCO clock and retimed signals.

3.3.1 Re-snap

Re-snap effectively resets the clock divider on the DCO clock edge that immediately proceeds a *REFCLK* edge. As a result, the *FBCLK* rising transition is less than one DCO cycle after the *REFCLK*. Re-snap plays an important role in enabling aggressive lock-time reduction with C-lock in several ways: (1) it allows frequency acquisition to begin with a small phase-error, preventing cycle slipping even during lock acquisition; (2) it reduces the DCO code-update latency in the PLL, allowing for the current-cycle DCO code to be applied sooner; (3) it expedites phase-lock during the phase-acquisition stage by reducing the initial phase error in the PLL to at most one DCO clock. Re-snap is employed in C-lock during frequency acquisition (if the TDC output is higher than a specified value indicating large phase difference) and before phase acquisition stages. As mentioned in section 5.2, C-lock performs frequency acquisition first before phase-acquisition. During frequency acquisition the PLL can accumulate a significantly large phase difference. Performing Re-snap at that moment can reduce the phase error by reducing the TDC output to be within one DCO time-period. This accelerates the phase-acquisition.

The Re-snap mechanism is implemented using a clock gater-based frequency divider consisting

of a modulo-N counter (that counts DCO clock edges). Under regular operation, an $FBCLK$ edge is generated after the feedback counter reaches a count of $N-1$ (Figure 3.8a). The Re-snap mechanism consists of an additional signal called 'Re-snap_en' that dynamically updates the feedback counter, resetting the counter to 0, as shown in Figure 3.8b. Appropriate timing of 'Re-snap_en' with appropriate resetting of the value of the counter ensures arrival of $FBCLK$ within one DCO cycle of $REFCLK$.

Figure 3.9 demonstrates how appropriate assertion of the 'Resnap_en' signal with an appropriate counter resetting value can ensure $FBCLK$ to be within one DCO cycle of $REFCLK$. When the system decides to perform Re-snap, it asserts a 'Re-snap_logic_en'. But the signal cannot be directly used to reset the counter. Appropriate timing and clock-domain crossing are important factors for practical implementation. Since $REFCLK$ and the DCO clock are relatively asynchronous, $REFCLK$ is first synchronized (re-timed) using four flip-flops clocked by the DCO clock before being used to perform Re-snap. From the re-timed $REFCLK$ signals a signal called 'REF_retimed_pulse' is generated with a pulse-width of one DCO time-period and time-period of T_{REFCLK}. 'REF_retimed_pulse' is asserted on the rising edge of the 3rd DCO cycle after $REFCLK$. If 'Re-snap_logic_en' is asserted during that time the 'Re-snap_en' signal is asserted on the next DCO cycle, which resets the counter value in the following DCO cycle. Since 'Re-snap_en' signal is generated 4 DCO cycles after the rising edge of $REFCLK$, the counter is reset at the 5th DCO cycle after the rising edge of $REFCLK$. Therefore, the counter value is reset to '4' instead of '0' to ensure that the next $FBCLK$ will be within one DCO cycle of the $REFCLK$.

Resetting the counter value in Re-snap can cause the absence of $FBCLK$ or two $FBCLK$ edges in the corresponding cycle. This may cause a temporary error (existing until the arrival of the next $REFCLK$) in the PFD. Therefore, the system has to wait one T_{REFCLK} after Re-snap before using the TDC output.

3.3.2 Frequency Acquisition

The Frequency Acquisition stage is tasked with frequency-locking the PLL to $REFCLK$. During this stage, the *Solver* first measures the time-period corresponding to N DCO cycles, where N

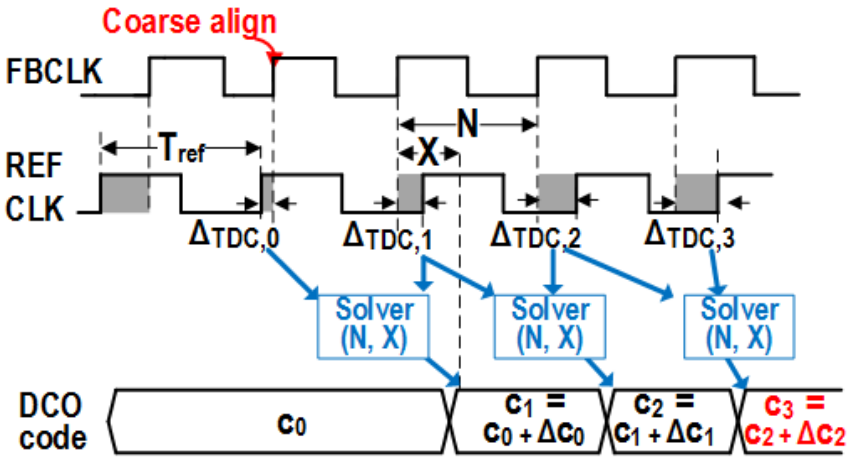


Figure 3.10: *REFCLK*, *FBCLK* and DCO codes during frequency acquisition

is the feedback divider ratio. The *FBCLK* time-period that results in cycle n, depends on both c_n and c_{n-1} due to PLL update latency. Therefore the solver first extracts $T_{DCO,n}$, the cycle time corresponding to c_n in order to suppress update latency effects. Knowledge of $T_{DCO,n}$ allows the system to determine subsequent DCO codes required to achieve frequency lock. In this approach, the very first DCO code update iteration requires knowledge of $T_{DCO,0}$, obtained by maintaining c_0 over an entire *REFCLK* cycle.

Denoting TDC outputs during cycle n and cycle $n-1$ as $\Delta_{TDC,n}$ and $\Delta_{TDC,n-1}$, the operation of *Solver* proceeds with a measurement of $T_{meas,n}$, the measured time-period of the PLL over cycle n of the *REFCLK*:

$$T_{meas,n} = T_{REFCLK} + g_{TDC}(\Delta_{TDC,n} - \Delta_{TDC,n-1}), \tag{3.2}$$

where g_{TDC} corresponds to the TDC gain. Once frequency locked, T_{meas} = T_{REFCLK}. To account for the impact of loop-delay, the *Solver* model uses ΔTDC codes across successive cycles to determine the applied DCO codes (Figure 3.10). During cycle n, the DCO operates on code c_{n-1} for X DCO cycles, the latency incurred in calculating c_n. Subsequently, c_n is applied for the remaining $N-X$ cycles. Therefore, $T_{meas,n}$ does not correspond to $T_{DCO,n}$ which would be observed under locked operation. To extract this time-period from T_{meas}, we first express it as a function of $T_{DCO,n-1}$ and

$T_{DCO,n}$:

$$T_{meas,n} = X \cdot T_{DCO,n-1} + (N - X) \cdot T_{DCO,n}. \tag{3.3}$$

Simplification of Equation 3.3 yields,

$$N \cdot T_{DCO,n} = T_{meas,n} + \frac{X}{(N-X)}(T_{meas,n} - N \cdot T_{DCO,n-1}). \tag{3.4}$$

The time-period error between $REFCLK$ and $FBCLK$ can be written as,

$$\Delta T_n = T_{REFCLK} - N \cdot T_{DCO,n}. \tag{3.5}$$

Simplification of Equations 3.4 and 3.5 yields,

$$\begin{aligned} \Delta T_n = {} & g_{TDC}(\Delta_{TDC,n} - \Delta_{TDC,n-1}) + \\ & \frac{X}{(N-X)}(g_{TDC}(\Delta_{TDC,n} - \Delta_{TDC,n-1}) - \Delta T_{n-1}). \end{aligned} \tag{3.6}$$

Note that $X = 0$ for $n = 0$ because c_0 is applied for an entire cycle to accurately obtain $T_{DCO,0}$. Consequently,

$$\Delta T_1 = g_{TDC}(\Delta_{TDC,1} - \Delta_{TDC,0}). \tag{3.7}$$

To obtain the frequency error ($\Delta f_{DCO,n}$), we rely on the approximation:

$$\frac{\Delta f_{DCO,N}}{f_{DCO,N}} \approx \frac{\Delta T_n}{T_{REFCLK}}, \tag{3.8}$$

where

$$f_{DCO,N} = N \cdot f_{REF} = \frac{N}{T_{REFCLK}} \tag{3.9}$$

is the frequency target of the DCO. Combining Equations 3.8 and 3.9 we obtain,

$$\Delta f_{DCO,n} \approx \frac{N \cdot \Delta T_n}{T_{REFCLK}^2}, \tag{3.10}$$

where $T_{DCO,n}$ is obtained from Equation 3.4. Once $\Delta f_{DCO,n}$ is known from Equation 3.10 and 3.6, the code adjustment Δc_n can be obtained as,

$$\Delta c_n = \Delta f_{DCO,n} \cdot \mu_n \cdot g_{DCO}, \tag{3.11}$$

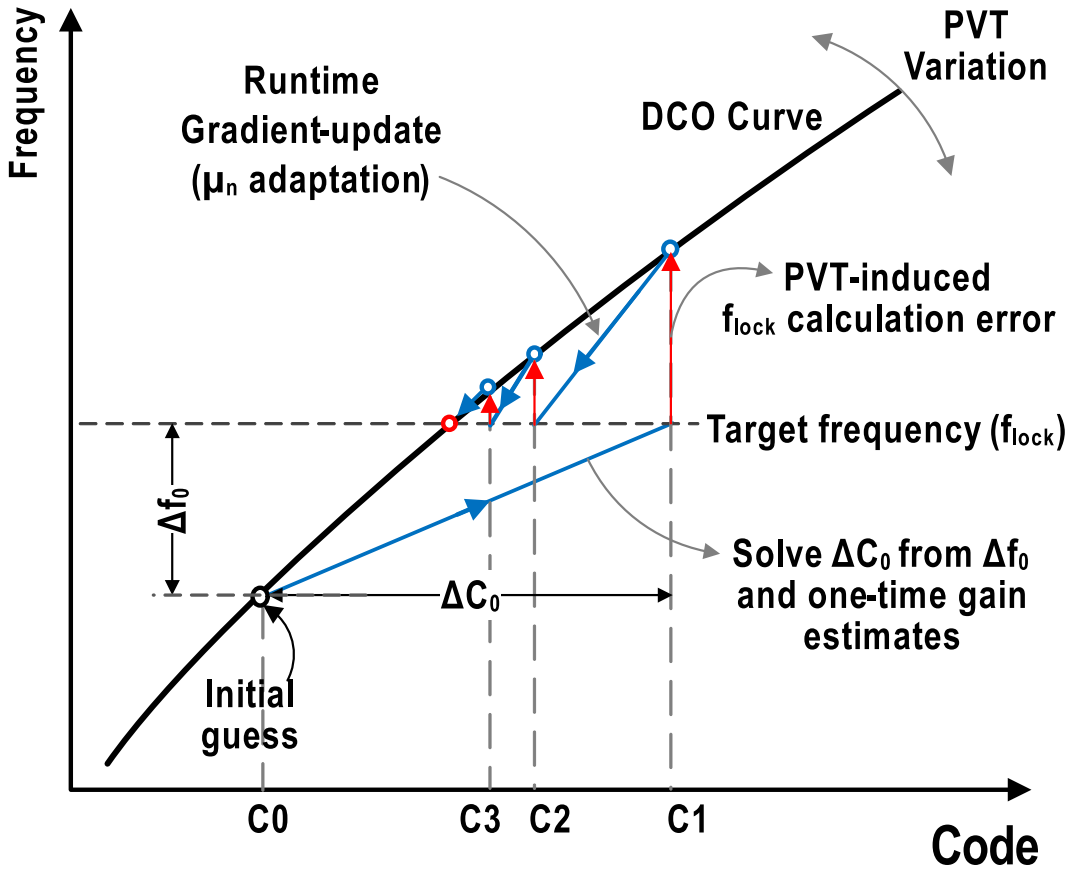


Figure 3.11: Frequency acquisition using an iterative gradient descent-like approach

where g_{DCO} is the gain of the DCO and μ_n is the step-size of each iteration. During frequency acquisition the *Solver* solves Equation 3.10 and 3.4 in each cycle and applies the resulting DCO code. This iterative process continues until ΔT_n, as obtained from Equation 3.6, is found to be within a user-specified tolerance limit. A final full-cycle measurement is conducted to confirm the results of the calculation. This measurement is obtained by maintaining the solved DCO code c^* in the next cycle, allowing T_{meas} to directly represent $T_{DCO} \cdot N$ for the subsequent *REFCLK* cycle.

The second term in the right hand side of Equation 3.6, consisting of the fractional coefficient $\frac{X}{N-X}$ accounts for the contribution of the previous-cycle DCO code to the current N-cycle time-period. A low value of X helps reduce the quantization errors caused by a fixed-point arithmetic

evaluation of Equation 3.6, and their propagation to subsequent cycles. One practical implementation of C-lock relies on pre-stored constant terms $\frac{1}{N-X}$ for the target range of values of N, while X is determined at run-time every cycle. The value X is determined by TDC resolution time, and computation time within the *Solver* module. Four DCO cycles are allotted to complete all of the computations for each iteration of lock acquisition to provide sufficient timing slack. After the TDC quantizes the delay between the incoming clocks, it asserts a 'TDC_done' signal (details in section 3.4.2). The TDC code is sampled by the *Solver* module at the rising edge of 'Retimed_TDC_done', a 2-flop synchronized version of the 'TDC_done' signal (section 3.4.3) to avoid metastability. The number of cycles incurred in resolving the TDC time (X) is also sampled by the DCO counter on cue from the 'Retimed_TDC_done' signal. If X sampled in this fashion exceeds 12 due to high phase error, another Re-snap is performed and frequency acquisition is restarted using the most recent DCO code, but with a smaller phase difference that reduces X.

The PLL loop-gain, used by the *Solver* to determine successive DCO code values for lock, varies with changing PVT conditions. Use of a poor gain-estimate by the *Solver* leads to an excessively larger number of iterations, and in extreme cases prevents frequency acquisition altogether. To address this challenge, the *Solver* updates this estimate by adjusting μ_n (Equation 3.10). Since each iteration of frequency acquisition seeks to eliminate frequency error, any observed frequency error, $\Delta f_{DCO,N}$, is largely the result of an imprecise loop gain model. Thus when required, μ_n is readily updated based on $\Delta f_{DCO,N}$. Figure 3.11 illustrates the operation of this gain adjustment that is performed only during the first iteration of frequency acquisition.

3.3.3 Phase acquisition

Phase acquisition represents the final phase of the C-lock process. At the onset of this phase, the system is frequency locked, with knowledge of c_N^*, the DCO code required for lock. A Re-snap is first performed to limit the initial phase error to within a DCO cycle (Figure 3.12). The system then determines the time-delay error between $FBCLK$ and $REFCLK$ as measured by the TDC. This error is used by the *Solver* to determine the magnitude of the excess-frequency "pulse"–deviations in the DCO code from c_N^*–that is to be applied to the DCO to cause targeted

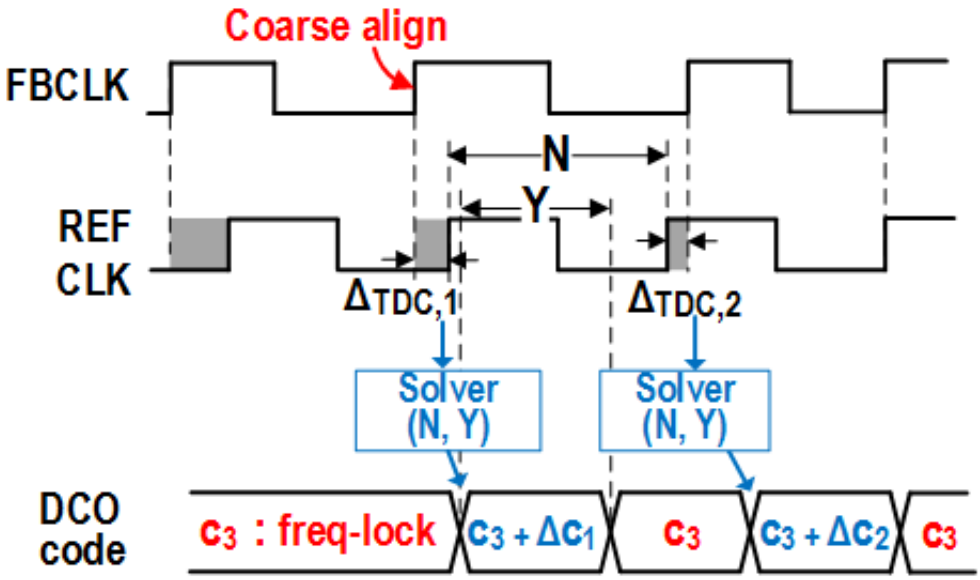


Figure 3.12: Waveform of REFCLK, DCO clk and DCO code during phase acquisition

phase-adjustments (Figure 3.12). This frequency pulse is applied for a fixed Y cycles within each T_{REFCLK}, ensuring that the PLL operates at c_N^* at the beginning of every *REFCLK*, allowing a seamless transition to phase-lock without update-latency problems. For a frequency pulse duration of Y DCO cycles, the applied frequency correction in each cycle can be derived to be,

$$\Delta f_{DCO,n} = f_{DCO,n-1} \cdot \frac{g_{TDC} \cdot \Delta_{TDC,n} \cdot N}{T_{REFCLK} \cdot Y}. \tag{3.12}$$

The corresponding DCO code c_n, can be obtained using Equation 3.11. Note that μ_n is neither performed nor necessary at this stage since gain-adaptation is performed during frequency acquisition.

3.4 PLL Implementation

3.4.1 DCO structure

The DCO used in the test-chip implementation consists of a ganged inverter-based programmable ring oscillator (Fig. 3.13). Modulating the drive strength of each stage of the inverter by introducing

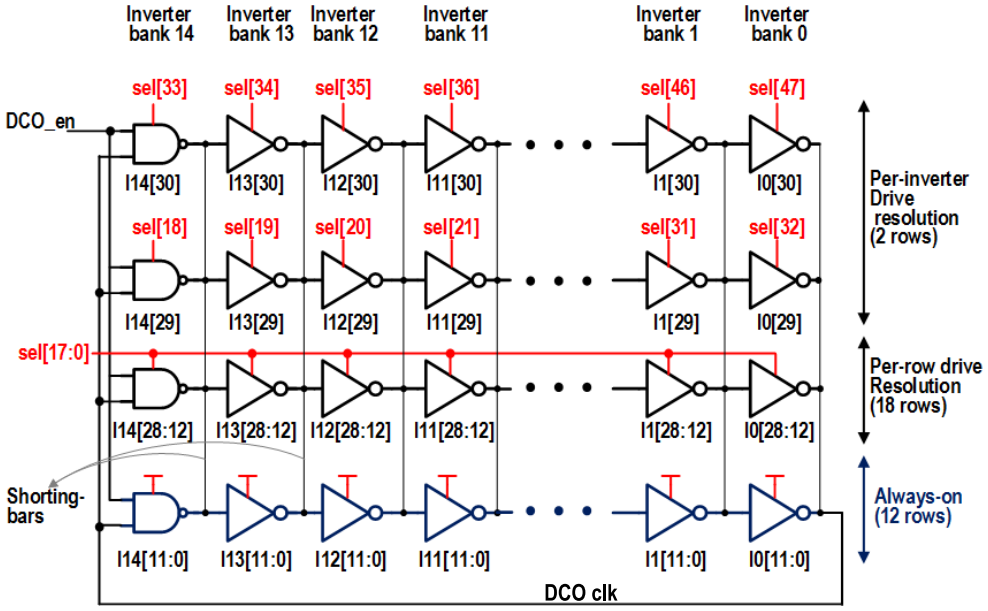


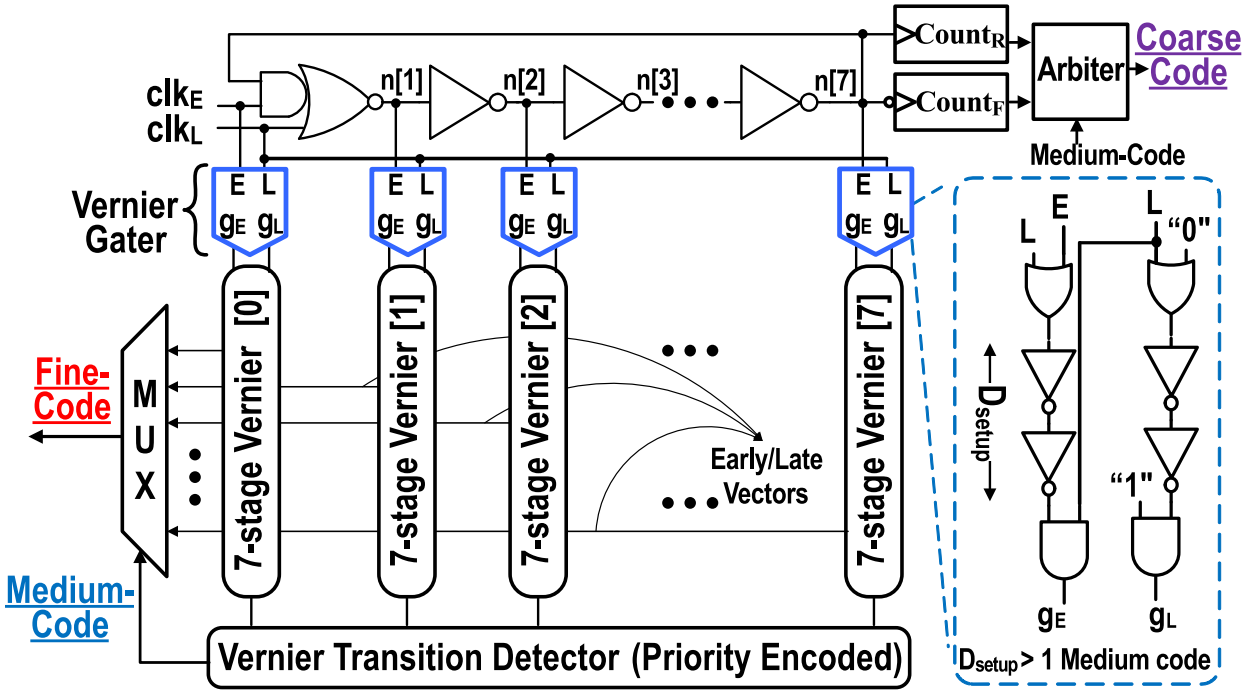
Figure 3.13: Digital Controlled Oscillator (DCO) structure. Frequency modulation is performed by tuning the strength of ganged inverter stages

.

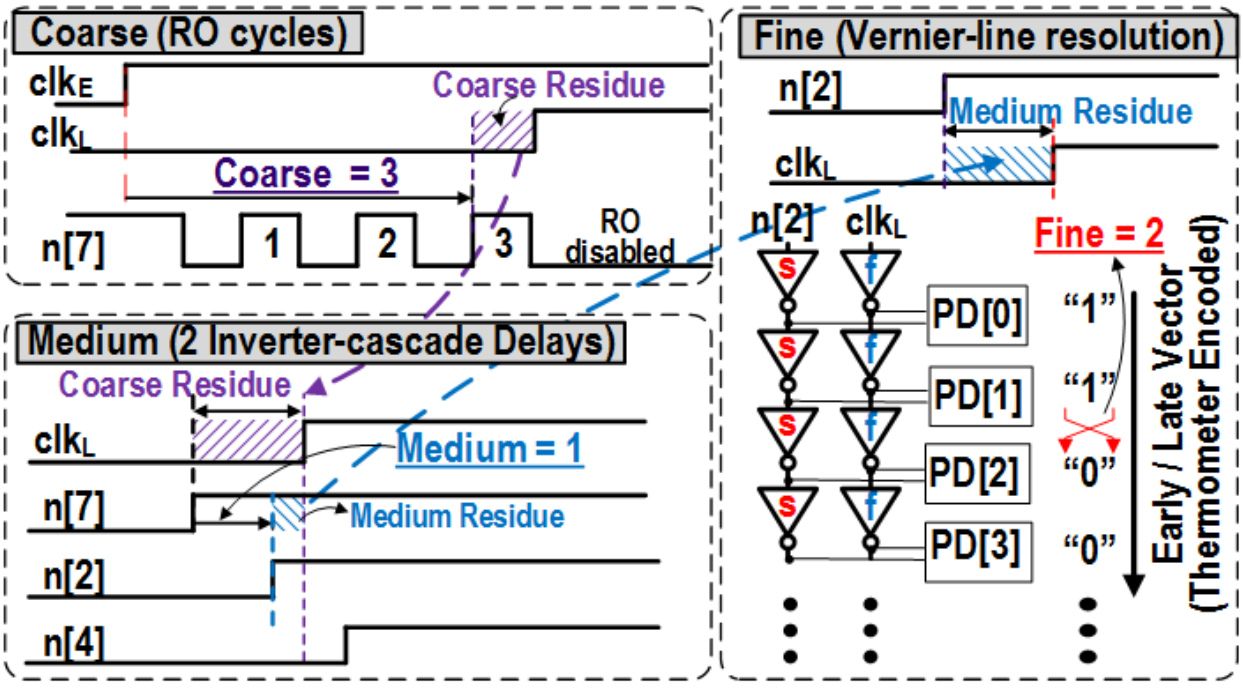
parallel inverter banks provides the required frequency modulation. The 15-stage oscillator is designed with each stage capable of adding a total of 20 inverters in parallel, requiring a total of 15*20 = 300 tunable inverters. Eighteen out of the twenty programmable inverters are organized as programmable rows that can be added to the DCO. The remaining two rows are designed to be individually programmable for improved frequency resolution.

3.4.2 TDC

Improved TDC resolution further reduces PLL lock-time. Various techniques have been proposed for high-resolution TDCs [78–82] but these techniques enhance resolution at the cost of latency. TDC latency is constrained in PLL applications by $REFCLK$, and the desire to minimize DCO code update latency. In the case of vernier-chain based TDCs, improving resolution for a given dynamic range increases power dissipation due to longer chain-length. To address the dual challenge of



(a)



(b)

Figure 3.14: (a) Proposed 3-step TDC Architecture, (b) TDC output in terms of coarse, medium and fine codes

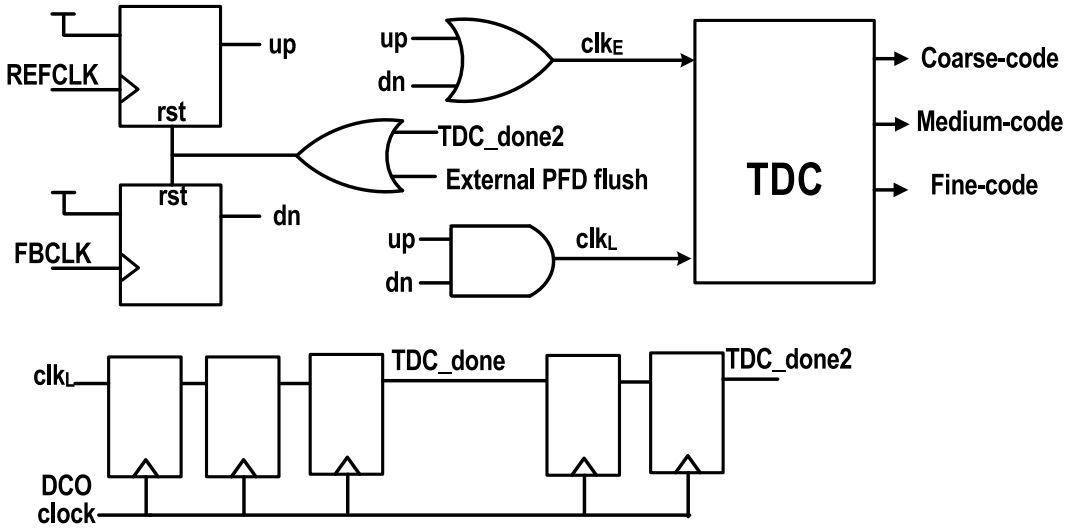


Figure 3.15: Phase-Frequency Detector (PFD) with TDC.

resolution and latency, we employed a 3-stage TDC [17] that provides wide dynamic range (8ns), high resolution (5ps) and low-resolution time (around 100ps worst case).

Fig. 3.14a shows the architecture of the TDC which digitizes the time differences between early clock (clk_E) and late clock (clk_L) at successively finer time-resolution. A 7-stage ring oscillator (RO) connected to the counter logic provides coarse-resolution (*Coarse Code*) with a wide dynamic range. A 7-stage vernier chain is connected to each RO node. Identifying the vernier chain that experiences a "cross-over" in arrival between the clk_E and the clk_L signal chains provides information on the location of clk_E within the RO at the arrival of clk_L at the resolution of one RO stage (*Medium Code*), while the index of the cross-over point offers sub-gate resolution (*Fine Code*). Thus, the TDC enjoys the wide dynamic range offered by the RO-clocked counters, while achieving sub-gate resolution offered by the vernier chain. In this architecture, the vernier chain length is driven by the need to achieve a dynamic range of only two RO inverter-stages.

Two counters, clocked by opposite phases of the clock, are used in conjunction with the *Medium Code* by arbitration logic to provide the *Coarse Code* while addressing the asynchronous arrival of clk_L relative to the counter clocks. A Vernier Gater (V-G) is employed at the input of each

vernier-chain to ensure that the vernier chains trigger only after the arrival of clk_L. The *OR* gate in the V-G is used to prevent the resetting RO edges after the arrival of clk_L from toggling the vernier chain. The phase-detector outputs from each vernier-chain are processed by the *Vernier Transition Detector* to select the vernier chain outputs where the cross-over between E and L edges (V-G inputs) occurs. The index within this chain where the cross-over occurs yields the *Fine Code*.

The TDC is used in conjunction with a PFD similar to [83] to resolve the time difference between $REFCLK$ and $FBCLK$, as shown in Figure 3.15. The inputs of the TDC, clk_E and clk_L, are generated by 'OR' and 'AND'ing the outputs of the flops in the PFD. clk_L is retimed using 3 DCO cycles to generate a synchronized 'TDC_done' signal. By the time 'TDC_done' is asserted, the TDC outputs are resolved and ready to be used by the controller. 'TDC_done2', which is a 2-cycle retimed version of 'TDC_done', resets the flip-flops in the PFD to prepare the PFD for the next measurement.

3.4.3 Solver

The *Solver* performs the necessary computations to calculate the DCO-code, perform adaptive gain-adjustment, asserts signals for Re-snap and hands over the PLL control to traditional mode. As mentioned in Section 3.3, with C-lock, the PLL computation cannot be materialized with a Type-I or Type-II Digital Loop Filter. The controller uses different equations to calculate DCO code, DCO code is sometimes halted, sometimes a DCO code is applied for fraction of a full T_{REFCLK}. To incorporate all these effects into a fully-autonomous system, the *Solver* is implemented as a finite state machine (FSM). Figure 3.16 shows the flowchart of the FSM in the *Solver*. C-lock starts with an initial Re-snap followed by two wait cycles ("Wait_after_resnap" and "Wait_init2") for the PFD to settle and to capture two successive TDC outputs before starting frequency acquisition. Frequency acquisition starts at the "Meas_update_freq_first" state. This state represents represents the first iteration of frequency acquisition. Because the DCO code remains unaltered over three cycles during this process, loop latency does not factor into the phase-frequency update equations, allowing for an unambiguous starting point for the frequency acquisition process. Equation 3.7 is used to calculate the DCO code in this first iteration. If the TDC output is excessive ($>$ 6ns), a Re-snap is performed

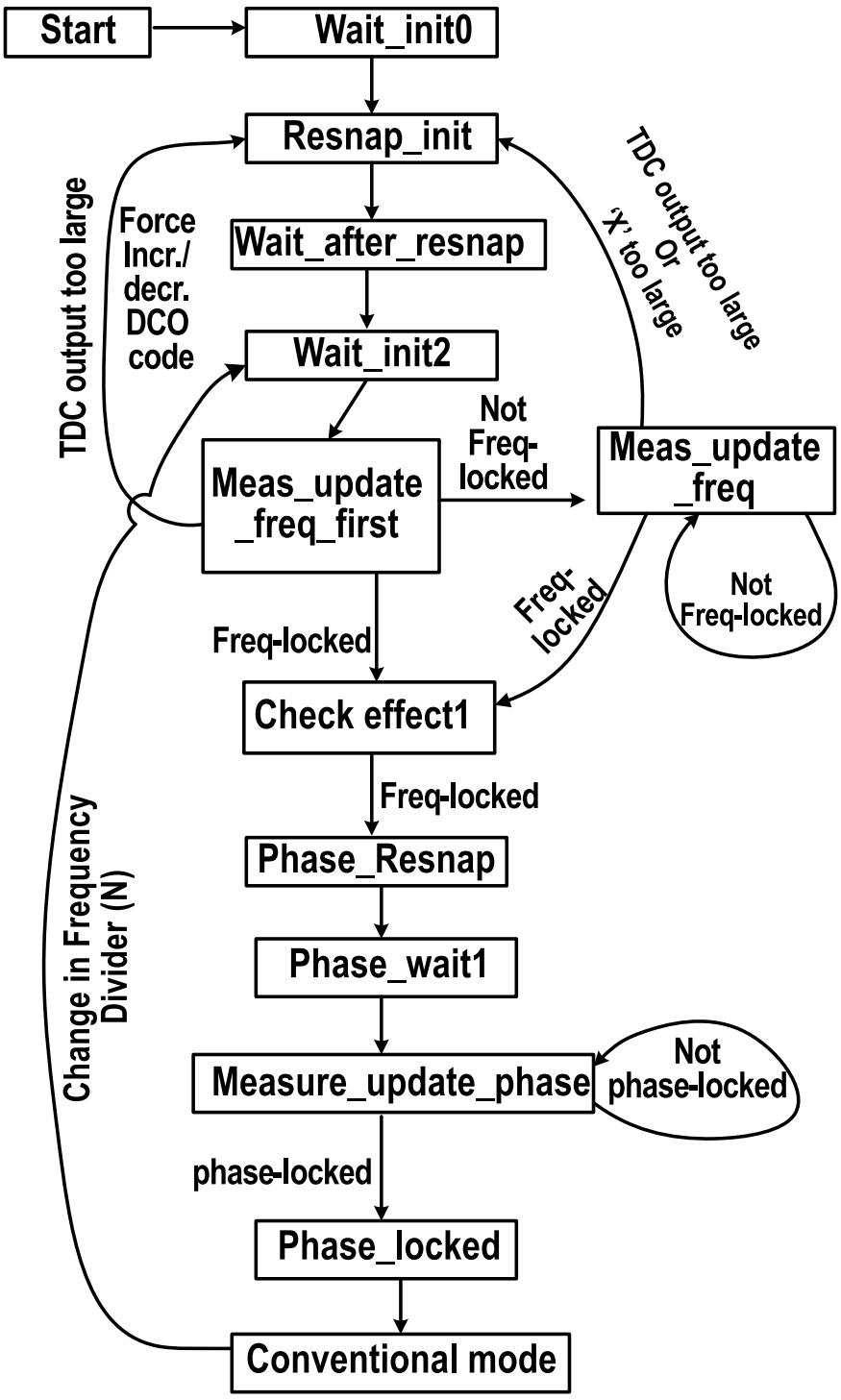


Figure 3.16: FSM states of *Solver*

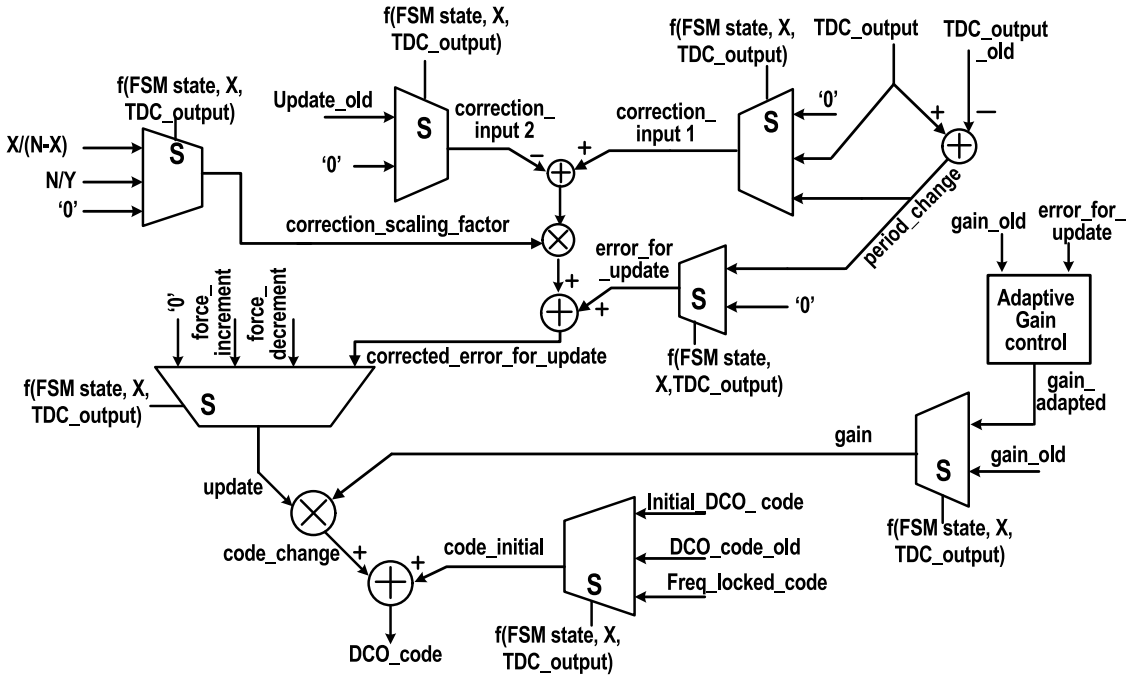


Figure 3.17: Datapath for DCO code calculation

to prevent imminent saturation of TDC while a force increment/decrement (10% of the DCO code) is applied depending on the polarity of the TDC code. If the TDC error is within the limit and the PLL is not frequency-locked, FSM goes to next state ”Meas_update_freq” where the loop latency effects are taken into account (Equation 3.6) to calculate the DCO code. An additional checking for the value of X is performed (mentioned in Section 3.3.2). If the TDC output is too large or X is too large, the system performs another Re-snap. Otherwise, the FSM stays in ”Meas_update_freq” and keeps computationally solving and updating the DCO code for frequency-locking. Either in ”Meas_update_freq_first” or ”Meas_update_freq”, if the time period error is within the tolerance limit (freq-locked), the system waits one more cycle and measures the time period error to confirm achieving frequency lock. Then another Re-snap is performed in the ”Phase_resnap” state to truncate the phase errors (in a coarse scale), followed by one cycle wait (”Phase_wait1”). Then the FSM starts performing phase acquisition in ”Measure_update_phase” state using Equation 3.12. Once the TDC output is within the phase-lock tolerance limit, the system transitions into the ”phase_locked”

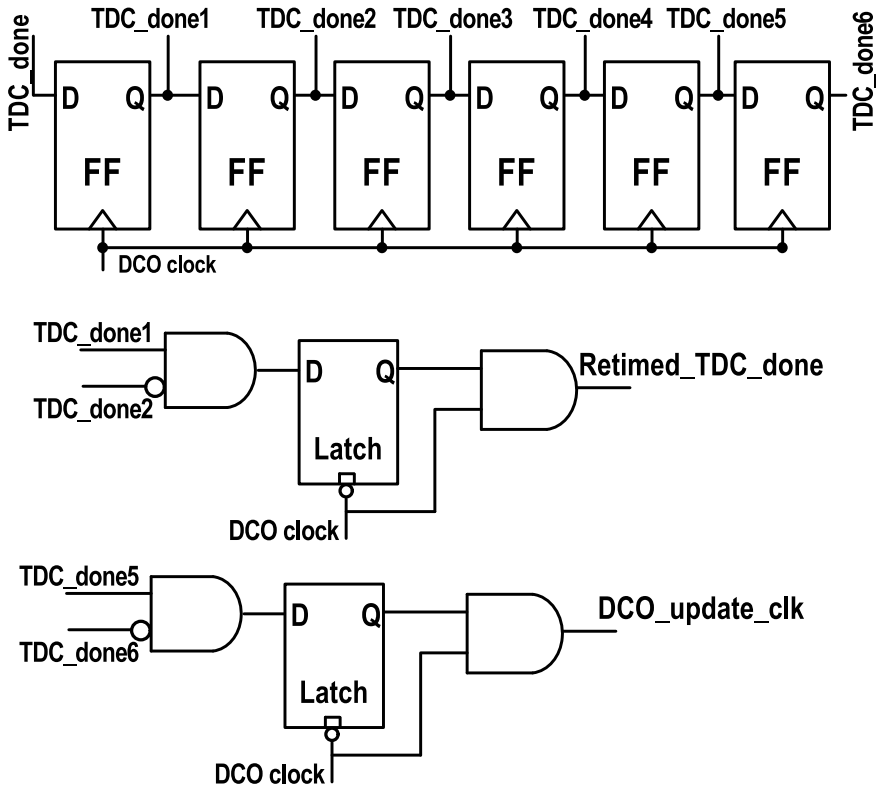


Figure 3.18: Gated clocks to control FSM update (concurrently with TDC reset) and DCO update

state where the accumulator within the traditional type-II controller is updated with the DCO code to enable a smooth transition into traditional mode. Finally, the FSM goes to ”conventional mode”, allowing the system to operate as a traditional type-II PLL. The FSM continues to monitor any change frequency divider (N) for the next locking action. Any change in N transitions the FSM into ”wait_init2”, starting another round of computational lock.

The tolerance level for frequency lock must be less than half of the tolerance level for phase lock to ensure phase-locking in all cases. In the implementation, the tolerance level for frequency lock and phase-lock was 20ps (for combined N DCO cycles) and 60ps (for combined N DCO cycles), respectively.

Figure 3.17 shows the datapath for DCO code calculation during C-lock. To ensure runtime reconfigurability, several MUXes are placed in the datapath. The selection logic for the MUXes are

derived from the FSM state, X (for determining Re-snap with/without force-increment/decrement) and TDC_output (for deciding Re-snap with/without force-increment/decrement, phase-lock). In the figure, 'update_old', 'period_change_old', 'gain_old', 'DCO_code_old' are the values of 'update', 'period_change', 'gain'. 'DCO_code' in the previous cycle. 'correction_input1', 'correction_input2', 'correction_scaling_factor' are used to incorporate the loop delay effect of Equation 3.6 and temporary phase update in equation 3.12.

Figure 3.18 illustrates the mechanism for generating the gated clocks that the *Solver* relies on to update the TDC and the DCO. The *Solver* works on 2 clocks: 'Retimed_TDC_done' and 'DCO_update_clk'. 'Retimed_TDC_done' is generated by retiming TDC_done using two DCO cycles, and is used to update the FSM state and flop the TDC decoder output. 'DCO_update_clk' is generated as a 4-cycle retimed version of 'Retimed_TDC_done' and is used to capture the DCO code at the output of data-path. The *Solver* obtains the parameters needed to calculate the DCO code at the rising edge of 'Retimed_TDC_done'. Using four synchronizing flops to generate retimed 'DCO_changing_edge' accordingly extends the time available to perform *Solver* computations for the DCO code, greatly relaxing timing constraints. The frequency divider places a more stringent timing constraint on the design than the multi-cycle datapath of the *Solver*. Further implementation details are available as an openly accessible Verilog repository [84].

3.4.4 Frequency divider

The frequency divider generates the $FBCLK$ with frequency f_{DCO}/N, where f_{DCO} is the frequency of DCO clock and N is the PLL divider ratio. $FBCLK$ is then provided to the PFD-TDC module to quantize the time difference relative to $REFCLK$. Figure 3.19 shows the implementation of the frequency divider, which is based on a modulo-N counter with clock-gating. When Re-snap is not enabled (Re-snap_en = 0), the counter value (freq_counter) is incremented in every DCO cycle. The counter is reset to 0 after reaching '$N-1$'. To support Re-snap functionality (Section 3.3.1), the counter used to generate $FBCLK$ can be loaded to a *Solver*-determined value at Re-snap.

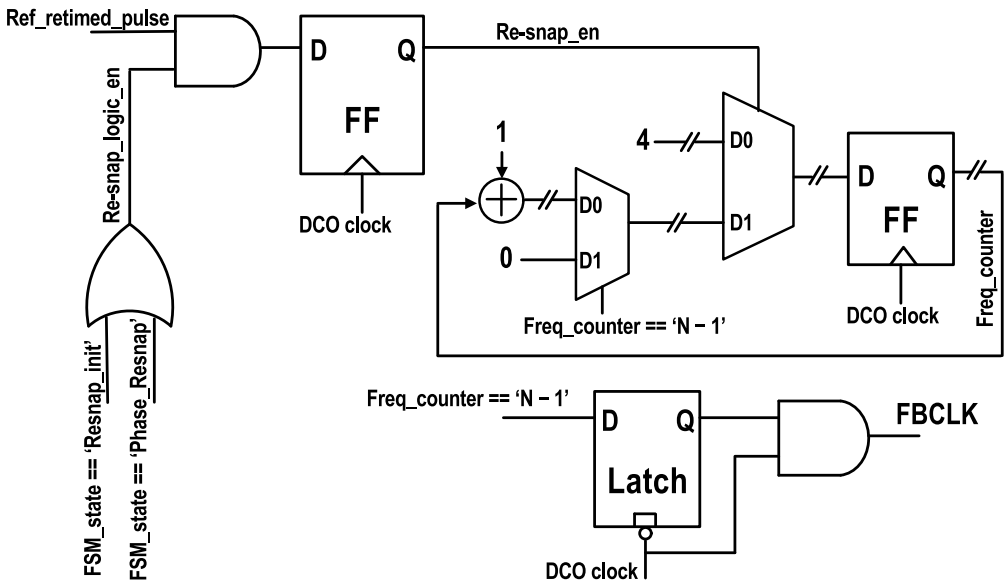


Figure 3.19: Frequency divider schematic

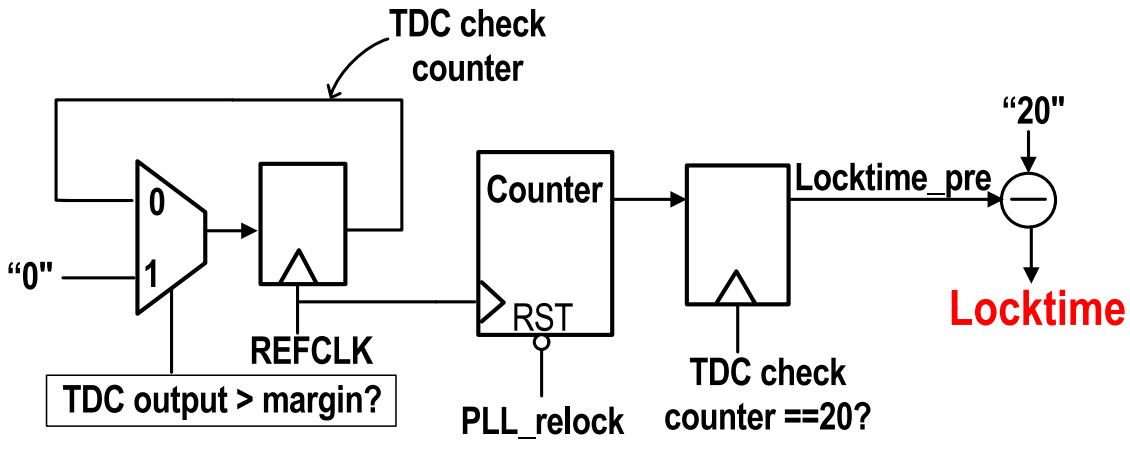


Figure 3.20: Lock-time counter module in BIST

3.4.5 Built-in self test (BIST) circuits

Observability of the PLL lock trajectory during post-silicon test is critical to validate C-lock. Parameters that describe the PLL trajectory are (a) FSM state; (b) TDC code; (c) measured frequency error; and (d) DCO code. A programmable FIFO was implemented to periodically capture these parameters at run-time to provide observability of key state variables during each iteration of phase-lock. An on-chip lock-time measurement unit provides lock-time in units of *REFCLK*. The lock-time measurement counter 3.20 begins counting at the onset of cold-start or re-lock events. The TDC code is measured at each *REFCLK* cycle. If the phase-error is within a user-defined tolerance and the *Solver* is in the phase-lock state, the counter value is captured. Subsequently, the TDC code continues to be checked for 20 cycles to qualify the sampled lock time.

3.5 Test Chip implementation and Measurements

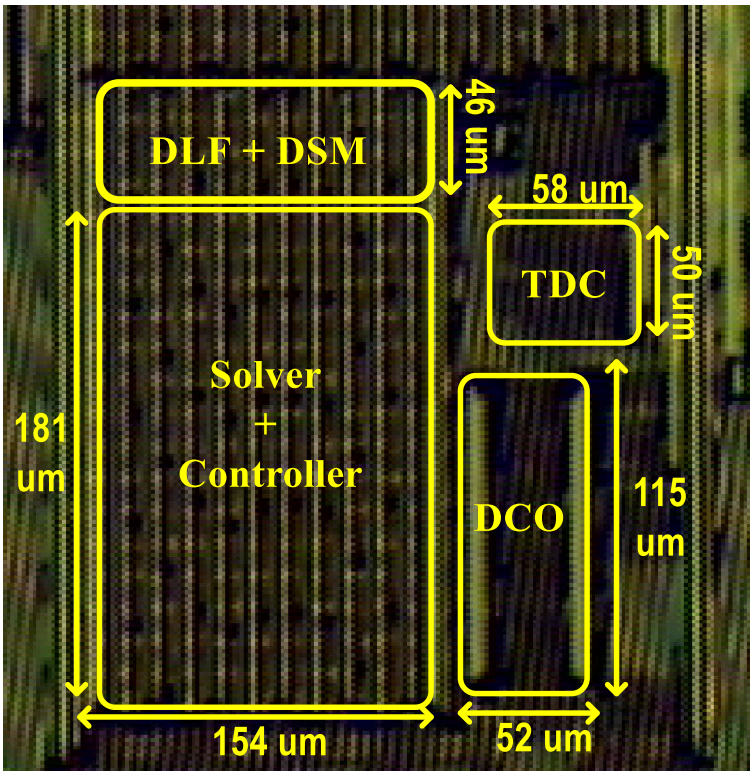


Figure 3.21: Die photo

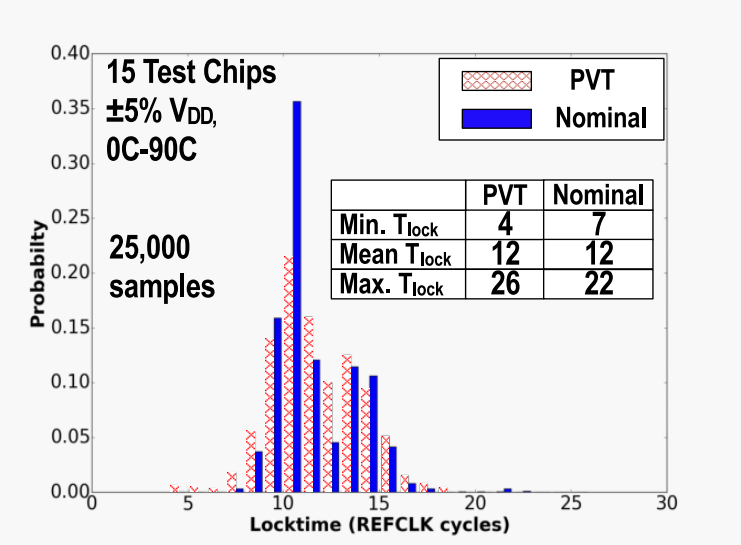


Figure 3.22: Histogram of T_{lock} for re-lock, in units of T_{REFCLK}, obtained from over 25,000 re-lock iterations across all from-to frequency combinations, with and without PVT

The computational PLL test chip was fabricated in 65nm CMOS technology (Fig. 4.7). The controller module, BIST and scan modules were all implemented through SAPR. C-lock incurs a 34% area overhead, which can be improved through a structured synthesis design approach, and fabrication in a more advanced process node. The energy overhead of C-lock does not significantly affect total PLL power due to being used only during lock acquisition: the *Solver* remains gated during steady-state operation.

PLL lock-time is frequently reported as a single number. However, because lock acquisition is non-deterministic in nature, a statistical representation of T_{lock} using mean and standard-deviation is more suitable. Thus, 50,000 iterations of PLL cold-start and re-lock were performed across 15 test chips, with $\pm$5% V_{dd}, and 0°C-90°C variation to incorporate the effects of PVT. An 'CSZ EZT-430i' temperature chamber was used to perform temperature sweeps.

Re-lock performance characterization involved 25,000 measurements across all possible supported frequency transitions. Measurement results with and without PVT variation are summarized by the histogram in Figure 3.22. Even under PVT variation, the mean and worst-case T_{lock} is found

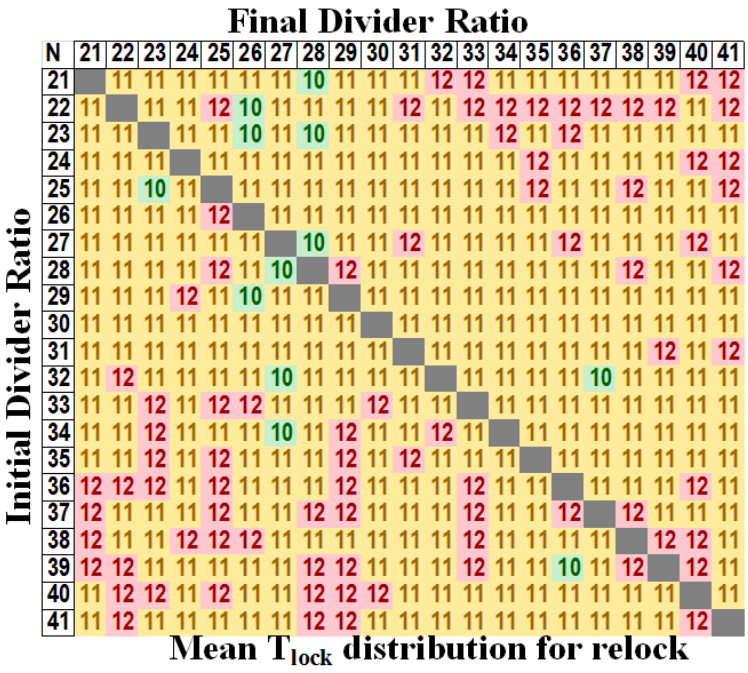
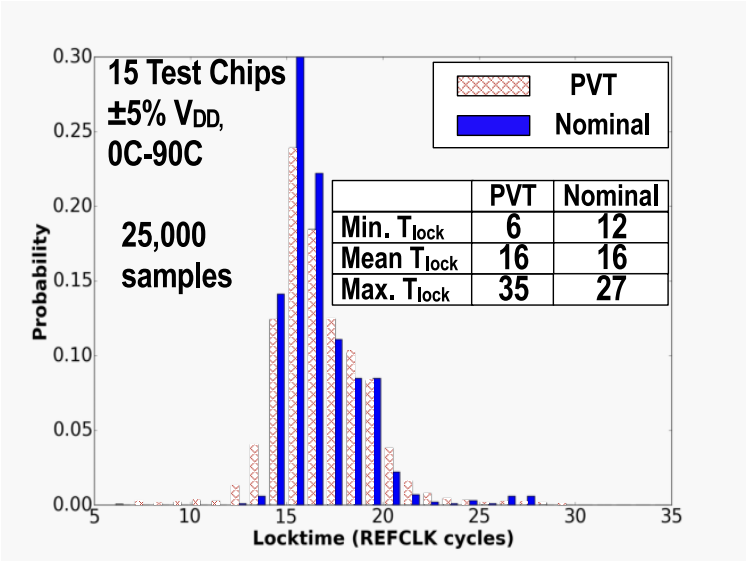


Figure 3.23: Mean T_{lock} obtained for each from-to pair of PLL frequency transitions. Each entry is determined from its corresponding T_{lock} histogram.



n

Figure 3.24: Histogram of T_{lock} for cold-start, in units of T_{REFCLK}, obtained from over 25,000 cold-start iterations across all frequencies, with and without PVT

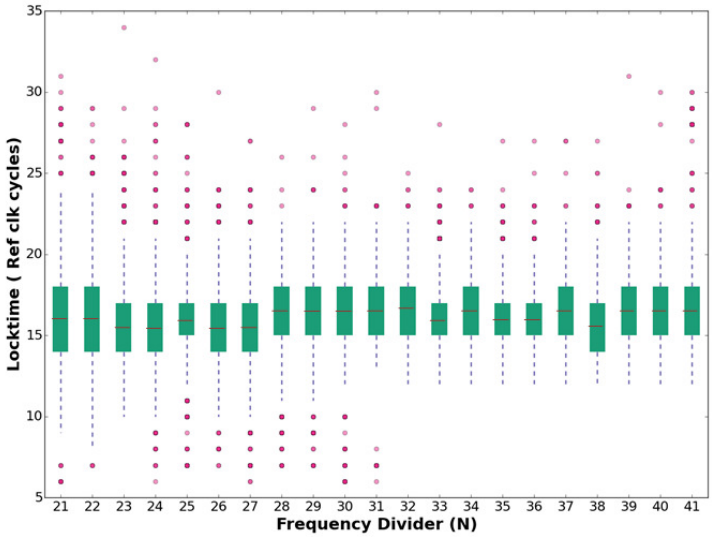


Figure 3.25: The statistics of cold-start locktime at all frequencies

to be 12 and 26 cycles respectively. Figure 3.23 summarizes the result of re-lock measurements. Matrix entries in row i and column j correspond to the mean measured re-lock time resulting from changing N from i to j. Matrix entries show that T_{lock} does not depend significantly on re-lock from-to frequencies.

For cold-start T_{lock} characterization, 25,000 measurements were obtained covering all possible target frequencies under nominal and PVT variation conditions. The resulting histogram (Figure 3.24) indicates a mean (worst case) locktime of 16 (35) *REFCLK* cycles. Figure 3.25 summarizes T_{lock} statistics specific to each target frequency. Under cold start, the PLL demonstrates no significant dependence between T_{lock} and target frequency.

As seen from Figure 3.22 and Figure 3.24 T_{lock} for cold-start exceeds that for re-lock. This increase in lock-time is attributed to additional FSM states that the *Solver* goes through for cold-start operation before starting frequency measurements.

Figure 3.26 shows a run-time trace of the time-delay between *FBCLK* and *REFCLK* as measured by the TDC at each *REFCLK* cycle during lock acquisition. C-lock begins with Re-snap (cycle 1) which limits the time-delay error to within one DCO cycle. In the following frequency acquisition

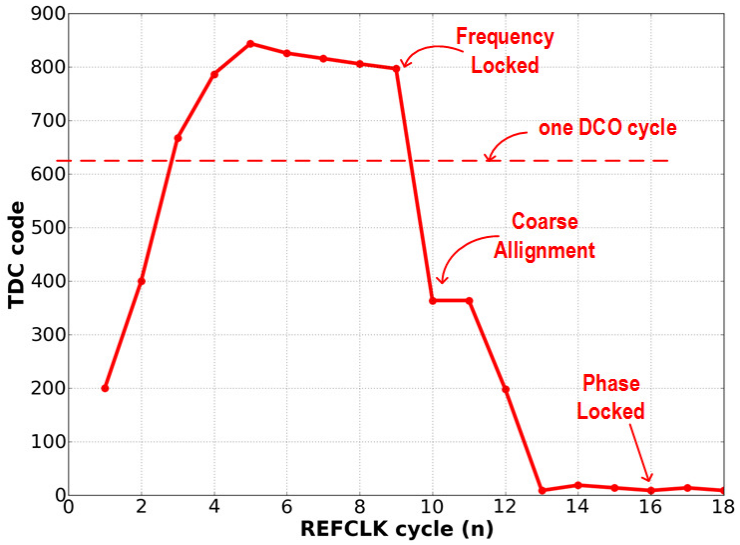


Figure 3.26: Post-silicon demonstration of C-lock in action : traces of TDC code during C-lock

Table 3.1: Mean T_{lock} (*REFCLK* cycles) at different *REFCLK* frequencies

F_{REF} (MHz)	25	30	35	40	45	50
Mean T_{lock} (*REFCLK* cycles)	13.3	11.8	11.6	12.8	12.8	12

mode, the solver module generates the sequence of DCO codes until the difference between successive TDC codes is below a threshold (at cycle 9, this instance). Another re-snap is performed after frequency acquisition (cycle 10) before the system begins phase acquisition. Here, time-delay is corrected using temporary (sub *REFCLK* cycle) frequency adjustments until the system is phase-locked (cycle 12). Once locked, the resulting DCO code is loaded into the counter in the integral path of the type-II controller. This code-transfer allows the subsequent loop-control hand-over to occur seamlessly, without any deviation in phase or frequency (cycle 13).

PLL lock-time was also tested over a range of *REFCLK* frequencies (F_{REF}). Table 3.1 shows the measured mean T_{lock} for re-lock across different F_{REF}. As seen from the table there is no significant

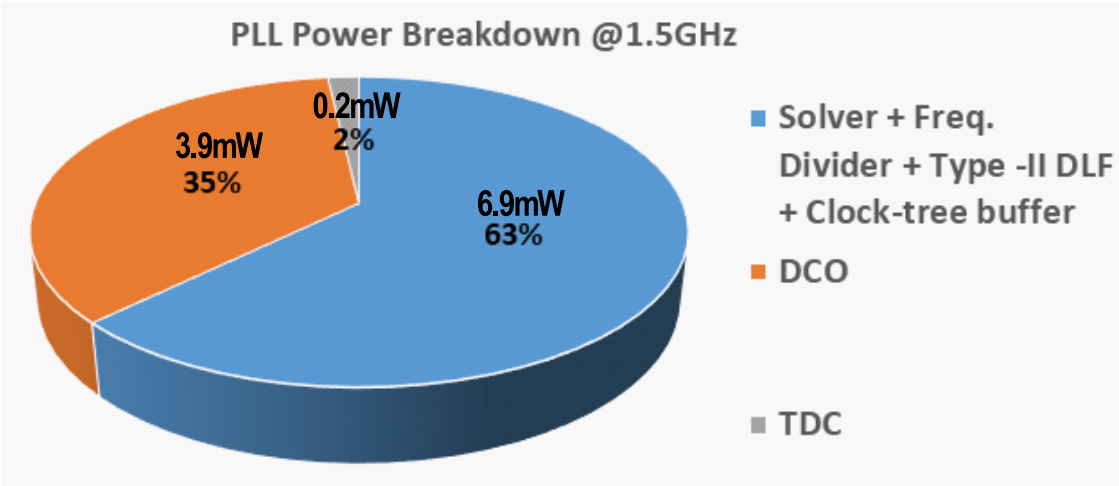


Figure 3.27: Steady-state PLL power break-down at 1.5GHz. Because the *Solver* is disabled in this mode, it does not contribute to total power

Table 3.2: Comparison Table

	[73]	[71]	[74]	[70]	**This work**
Process Technology	0.18μm CMOS	14nm CMOS	16nm CMOS	0.18μm CMOS	65nm CMOS
PLL Arch.	ADPLL	Analog PLL	DPLL	ADPLL	ADPLL
Output Freq. (GHz)	0.03-0.45	0.15-5	0.5-9.5	0.254-1.37	1-2
F_{REFCLK} (MHz)	0.22-8	19.2, 100	60	19.53	50
Jitter (ps)	70 (peak) @446MHz	3.765 (RMS) @1.6GHz	0.45 (RMS)	8.88 (RMS) 32.5 (peak) @1.25GHz	3.09 (RMS) 7.55 (peak) @1.5GHz
Power	14.5mW @446MHz	0.65mW @ 1.6GHz	7.1mW	35mW 1.25GHz	10.8mW @1.5GHz
Best-case T_{lock} (T_{REFCLK})	2	NR	NR	NR	**4 (relock) 6 (cold-start)**
Mean T_{lock} (T_{REFCLK})	NR	100	75	57	**12 (relock) 16 (cold-start)**
Worst-case T_{lock} (T_{REFCLK})	NR	NR	NR	NR	**27 (relock) 35 (cold-start)**
PVT tolerant	NO	YES	YES	YES	YES

variation of T_{lock} across a range of F_{REF} from 20-50MHz. The number of $REFCLK$ cycles required to achieve lock remains largely unchanged. However, the absolute lock-time (in units of seconds) will vary with T_{REF}. The measurement results from the table show that C-lock application is not restricted to a specific F_{REF}.

The PLL consumes 10.8mW of power in steady-state at 1.5GHz. Figure 3.27 shows a pie-chart of this power breakdown. The type-II DLF, solver, clock-tree buffers and frequency divider consume 6.9mW. When active the TDC ring oscillator consumes 3mW plus an additional 3pJ of energy consumption in the vernier tree flops, arbiters and decoders. The PLL oscillator (DCO) consumes 3.9mW of power. In steady-state the PLL is locked, and the total power consumption of the TDC is always less than 0.2mW.

Table 3.2 compares key design metrics of the proposed PLL with prior related work. It is important to note that although the *Solver* dissipates a considerable amount of power, this dissipation does not lead to a significant change in PLL power due to its infrequent use only during lock-acquisition. Furthermore, C-lock does not impact steady state PLL performance (jitter, power dissipation), which are determined by the properties of traditional PLL control. The reported measurements for steady-state PLL jitter and power measurements are included for completeness, however, and are not the result of computational locking. We report best-case, mean and worst-case T_{lock} , recognizing the statistical nature of lock-acquisition, and its implications on variable T_{lock} . Even considering worst-case lock-time, C-lock compares favorably among prior efforts that have been designed to be robust to PVT variation.

3.6 Conclusion

We proposed Computational Locking (C-lock), a technique for robust, rapid lock acquisition in ADPLLs. The technique relies on solving cycle accurate equations that govern PLL phase and frequency update. We motivated the effort by identifying the causes behind longer locktime in traditional closed loop ADPLLs, and subsequently explained how C-lock achieves rapid frequency and phase acquisition during both cold-start and re-lock acquisition. We presented test-chip results obtained by implementing C-lock on an ADPLL. The proposed architecture presents a negligible

power overhead. Lock-time statistics gathered over 25,000 iterations of cold-start and re-lock time indicate mean T_{lock} values of 12 and 16 *REFCLK* cycles respectively.

The goal of this effort was to demonstrate computational locking as an approach to accelerating lock times in PLLs. For ease of implementation, we demonstrated this work using an integer-N PLL. However, the broader principles of C-lock can be applied on fractional-N PLLs, an outline of the implementation is provided in Appendix B at the end of this chapter.. Furthermore, although the PLL implemented in this work is intended for system-clocking applications, we anticipate C-lock to be applicable to PLL designs for wire-line and wireless applications in the future.

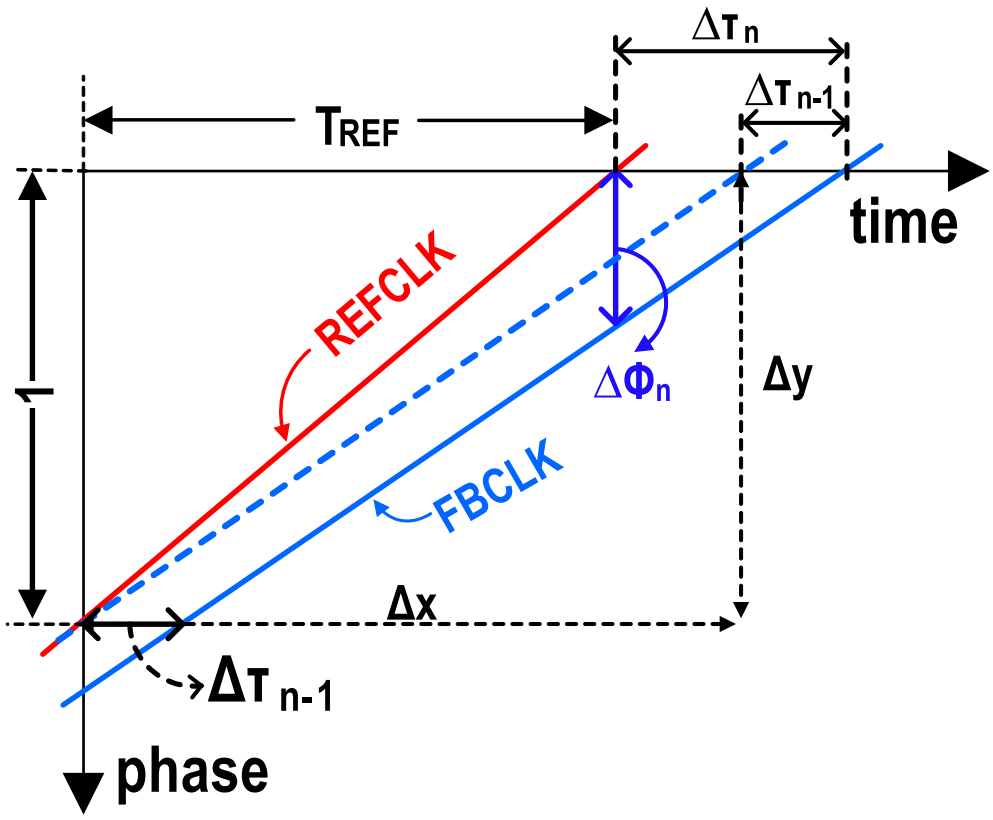


Figure 3.28: Deriving the relation between phase difference ($\Delta\Phi_n$) vs time difference ($\Delta\tau_n$)

3.7 Appendix A: Relationship between phase and time error

Figure 3.28 shows a phase (normalized to 2π) vs time plot for $REFCLK$ and a non-frequency locked $FBCLK$. The discrete-time (z-domain) model of the PLL relies on T_{REFCLK} as the sampling time-period. The Phase difference in cycle n, $\Delta\Phi_n$, is measured at the rising edge of $REFCLK$, while the PLL obtains time-delay errors between $REFCLK$ and the $FBCLK$ clock ($\Delta\tau_n$). To obtain $\Delta\Phi_n$, we scale $\Delta\tau_n$ with the gradient of the $FBCLK$ phase-time line (its frequency, f_{FBCLK}) as shown in Figure 3.28. From the figure,

$$\frac{\Delta\Phi_n}{\Delta\tau_n} = \frac{\Delta y}{\Delta x} = f_{FBCLK} = 1/T_{FBCLK}. \tag{3.13}$$

where T_{FBCLK} is the time period of the $FBCLK$ cycle.

Also as seen from the figure, T_{FBCLK} relates to T_{REFCLK}, and the measured time-delay errors τ_n and τ_{n-1}:

$$T_{FBCLK} = T_{REFCLK} + \Delta\tau_n - \Delta\tau_{n-1}, \tag{3.14}$$

Equation 3.13 can be rewritten as,

$$\Delta\Phi_n = \frac{\Delta\tau_n}{T_{FBCLK}}, \tag{3.15}$$

which, combined with Equation 3.14 yields the relationship shown in Equation 3.1 clarifying the time-varying nature of the phase-time relationship during lock-acquisition, as T_{FBCLK} transitions to T_{REFCLK}.

3.8 Appendix B: Fast-locking Frac-N PLL

For fractional-N frequency synthesis, the frequency divider in the PLL usually uses a delta-sigma modulator to enable fractional frequency division in the feedback path. The delta sigma modulator is clocked by REFCLK and because of that the PLL has to operate in low-bandwidth to filter out the quantization noises. The low-bandwidth operation of fractional-N PLL results in very long lock-time.

To digitize fractional frequency, the TDC can be used. Lets assume the frequency divider N consists of integer part N_{int} and fractional part N_{frac}. If $T_{REF,TDC}$ is the REFCLK time-period

measured in terms of TDC code then the time period error corresponding to the fractional frequency is,

$$T_{frac,TDC} = \frac{T_{REF,TDC} \cdot N_{frac}}{N}. \tag{3.16}$$

As for frequency acquisition, equation 3.6 needs to be modified to,

$$\begin{aligned} \Delta T_n = {} & g_{TDC}(\Delta_{TDC,n} - \Delta_{TDC,n-1}) + \\ & \frac{X}{(N-X)}(g_{TDC}(\Delta_{TDC,n} - \Delta_{TDC,n-1}) - \Delta T_{n-1}) - T_{frac,TDC}. \end{aligned} \tag{3.17}$$

For phase acquisition, a delta-sigma quantizer‘(DSM) should be used like traditional frac-N PLLs. But the system will also use the fractional part of the DSM ($DSM_{n,frac}$) to enable the fractional phase offset correction, $\Delta_{TDC,n,frac}$ with the following equation:

$$\Delta_{TDC,n,frac} = \frac{T_{REF,TDC} \cdot DSM_{n,frac}}{N}. \tag{3.18}$$

For phase-acquisition, equation 3.12 will be modified to,

$$\Delta f_{DCO,n} = f_{DCO,n-1} \cdot \frac{g_{TDC} \cdot (\Delta_{TDC,n} + \Delta_{TDC,n,frac}) \cdot N}{T_{REFCLK} \cdot Y}. \tag{3.19}$$

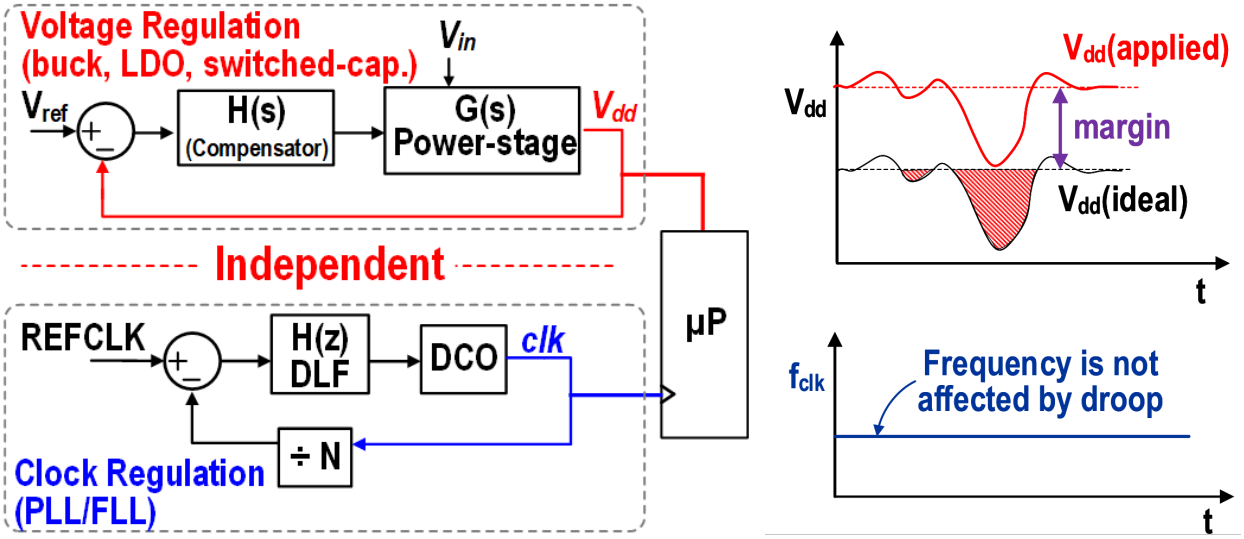
Chapter 4

A UNIFIED CLOCK AND SWITCHED-CAPACITOR-BASED POWER DELIVERY ARCHITECTURE FOR VARIATION TOLERANCE IN LOW-VOLTAGE SOC DOMAINS

Traditional digital systems regulate V_{dd} and f_{clk} separately using two independent control loops (Fig. 4.1a). A Phase-Locked Loop (PLL) powered by a regulated supply typically produces a stable fixed-frequency clock that locks to a reference (*REFCLK*). A separate voltage regulator loop maintains the target V_{dd}. Despite recent advances in voltage regulator design [85–87], load transients and regulator input supply voltage (V_{in}) fluctuations lead to significant V_{dd} droops (or surges). Because V_{dd} and f_{clk} loops operate in isolation, the f_{clk} loop has no knowledge of, or ability to adapt to either V_{dd} droop or temperature changes that degrade timing-slack and cause failure (Fig. 4.1b). Consequently, significant V_{dd} guard-bands (margins) are introduced to ensure robust operation at target f_{clk}. Voltage guard-bands prevent timing failure during relatively rare, worst-case logic-slowdown events, but significantly increases dynamic and leakage power across the entire design. These margins are particularly detrimental to sub- or near-threshold voltage (NTV) applications where delay sensitivity to V_{dd} variation is significantly higher.

Our work elaborates upon a recently demonstrated Unified Clock and Power (UniCaP) architecture [18, 88] ideally suited to the design of robust, efficient sub- and near-threshold systems. In general, UniCaP [88–91] affords several advantages over the traditional two-loop approach, most notably including aggressive V_{dd} margin reduction. UniCaP relies on a single, unified clock and voltage control loop (Figure 4.2), where voltage conversion–performed either by a switched-inductor, switched-capacitor, or linear-regulator–is contained within the clock regulation loop enabled by a PLL.

In UniCaP, the PLL employs a V_{dd}-powered tunable-replica oscillator (TRO) which can be



(a) (b)

Figure 4.1: Conventional digital systems with (a) Independent control loops for voltage and frequency regulation. (b) Voltage margins are needed to maintain timing-slack under voltage droop conditions.

tuned to match the delay and V_{dd} sensitivity of the system critical path. Use of a V_{dd}-powered TRO ensures that supply and temperature disturbances, however rapid, affect the critical path and the oscillator time-period identically, regardless of the control loop bandwidth: a V_{dd} droop causes a TRO cycle-time increase identical to the delay of the critical path (Fig. 4.2). A V_{dd}-powered oscillator-based PLL incurs significant jitter, but this jitter is deliberate and correlated with the critical path delay.

Previous works have demonstrated 'elastic clock' approaches for V_{dd} droop margin mitigation in production silicon[92–98]. In [93–96], the detection of a V_{dd} droop triggers a short-term f_{clk} reduction to maintain timing slack. Detection latency, however, significantly degrades both recoverable margin and performance. Furthermore, these techniques do not take advantage of V_{dd} surges that accompany any V_{dd} droop, yielding a net performance loss. Techniques aimed at injecting V_{dd} noise directly into the oscillator [97–100] avoid this detection latency. However, open-loop operation [97]

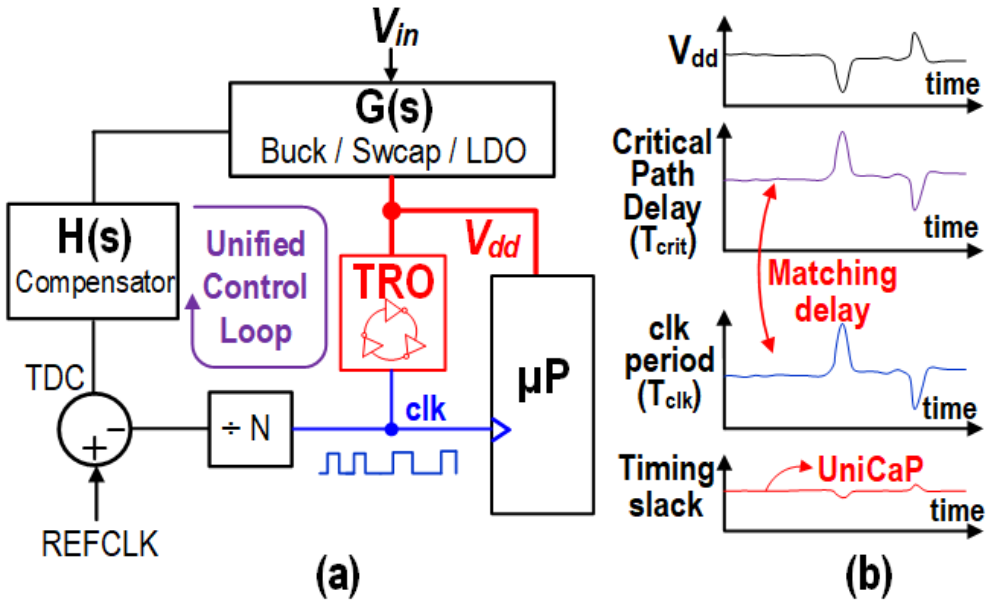


Figure 4.2: (a) Unified Clock and Power(UniCaP) architecture regulates f_{clk}. (b) an elastic clock maintains timing-slack in the event of a V_{dd} droop.

is not well-suited to many real-world applications, and PLL-based techniques [98, 100] are unable to reject V_{dd} noise that occurs within the PLL bandwidth and do not track temperature variation, degrading efficiency. In [101], a ring-oscillator was employed as a voltage-to-time converter to enable all-digital construction, but without a unified clock and power, the approach does not offer any V_{dd} droop margin reduction.

A unified control loop allows UniCaP to reconcile two opposing goals: (1) adjusting f_{clk} in response to V_{dd} droop or temperature changes in the short-term while (2) simultaneously regulating the clock to a fixed reference in the longer-term. The key enabling idea behind UniCaP is incorporating the voltage converter stage into a PLL which uses V_{dd} control for f_{clk} modulation. Such an approach ensures that V_{dd} changes driven by the loop affect the TRO and the critical path identically, thus canceling any timing-slack impact.

UniCaP is expected to significantly benefit low-voltage, cost-sensitive systems for IoT and sensor applications where voltage regulator performance and low operating voltages (potentially

NTV) require even more conservative V_{dd} guard-bands. Furthermore, although generally applicable across voltage converter technologies, UniCaP is particularly well-suited to designs using switched-capacitor voltage converters (SCVCs). Compared to switched inductor and linear regulator alternatives, fully-integrated SCVCs present a favorable trade-off between form-factor, cost and energy-efficiency for sensor and IoT applications. However, SCVC-based systems are hindered by a stringent trade-off between continuous voltage-scalability and transient response, which requires additional voltage margins, limiting efficiency and applicability. An SC-based UniCaP architecture (UniCaP-SC) avoids this trade-off, enabling continuous voltage scalability without the accompanying voltage droop margin requirements.

Unifying the clock and V_{dd} control loops into offers several additional benefits. In addition to V_{dd} droop, UniCaP-SC also tracks temperature variation, obviating wasteful temperature-variation related margins significant at low V_{dd}. V_{dd} droop immunity provided by the elastic clock decouples V_{dd} droop margins from voltage regulator bandwidth. The elastic clocking provided by UniCaP-SC allows it to withstand rapid V_{dd} droops even with using a low-bandwidth feedback loop. Bandwidth requirements are determined by how frequency or phase lock need to be restored. Supply-droop tolerance also allows UniCaP-SC systems to be designed with little or no additional decoupling capacitance (decap), reducing area and cost. Relying on the *REFCLK* as the sole reference, UniCaP-SC allows true all-digital system construction, without the need for even a voltage reference. All these benefits are obtained while delivering a system that regulates system operating frequency. Importantly, the proposed architecture does not interfere with digital-design methodologies for timing closure, affecting only post-silicon methodologies to guard-band V_{dd}.

We present the design and implementation of a UniCaP-SC system in a 65nm CMOS process. A fully-integrated 8-way phase-interleaved 2:1 SCVC featuring bottom-plate charge-recycling was implemented to power a fully functional NTV Cortex-M0 microprocessor and an FFT accelerator. With V_{in}=1.2V, the UniCaP-SC system demonstrated continuous voltage scalability in the 0.44V-0.56V range without any associated increase in supply guard-bands or use of series LDOs [102, 103], typically employed when scaling conventional SCVC systems. The test-chip was verified for functional correctness throughout its operating range, and achieved 150mV of V_{dd} margin reduction,

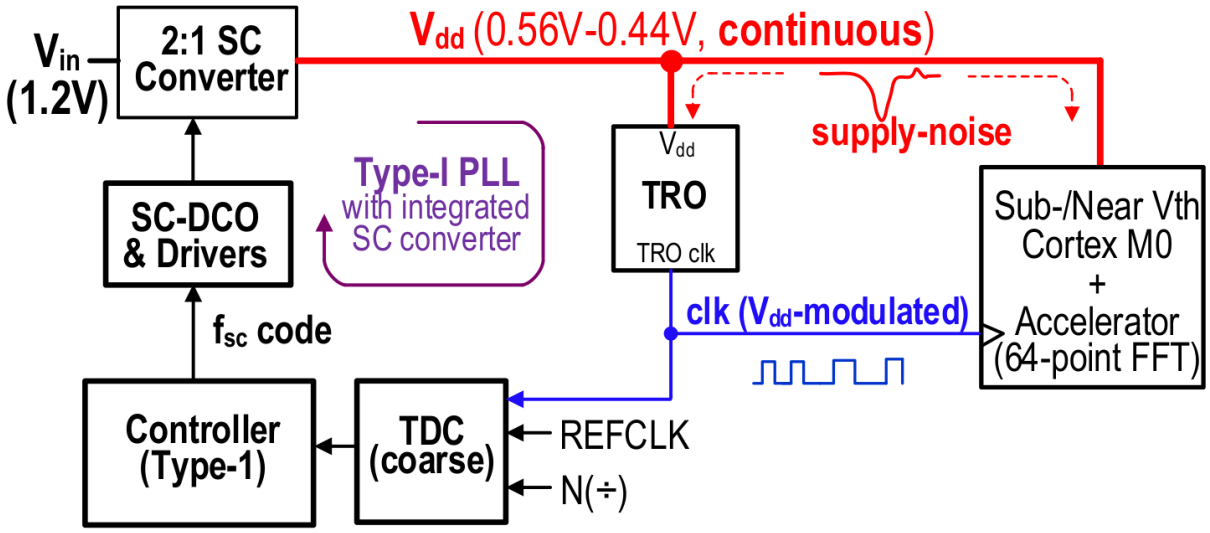


Figure 4.3: Block diagram of the UniCaP-SC architecture implemented in 65nm CMOS.

amounting to an average of 94% V_{dd} margin recovery over its operating range.

The remainder of this chapter is organized as follows: Circuit and architectural details of UniCaP modules are discussed in Section 4.1. Circuit implementations specific to the test-chip, including the wide capture-range time-to-digital converter (TDC), SCVC, and TRO are presented in Section 4.2. We present test-chip measurements in Section 4.3. Finally, a brief discussion of notable UniCaP-SC considerations are provided in Section 5.6.

4.1 UniCaP-SC

This section describes the UniCaP-SC structure, and its advantages over conventional SC-based design using the prototype implementation as an example (Figure 4.3). A V_{dd}-powered TRO clocks the Cortex-M0 processor, the FFT module, and a frequency divider. Figure 4.4 illustrates an implementation of the UniCaP-SC loop compensation architecture. A Time-Digital Converter (TDC) quantizes the delay between the arrival of rising edges of the divided feedback clock and *REFCLK*. A Digital Loop Filter (DLF) subsequently processes per-cycle phase errors. Unlike traditional ADPLLs which use the DLF output to directly modulate the clock frequency using a digitally

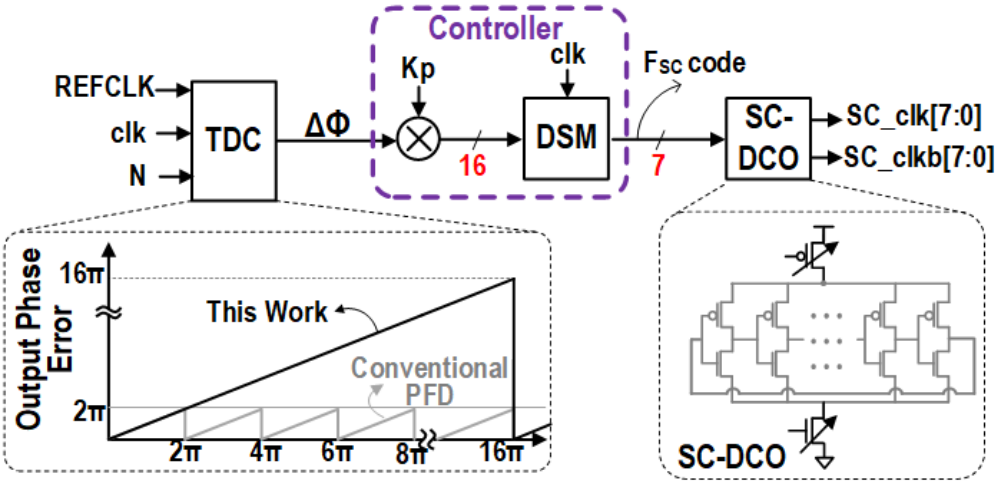


Figure 4.4: FLL control path with wide-range TDC, compensator, Delta-sigma modulator(DSM) and Switched-capacitor Oscillator(SC-DCO) providing 8 phases (SC_clk[7:0]) for the phase interleaved SCVC.

controlled oscillator, UniCaP-SC relies on the DLF to modulate the switching frequency of an 8-way phase-interleaved SCVC through the SC-DCO. Thus, UniCaP-SC relies on SCVC output impedance (R_{out}) control to modulate V_{dd}, and cause the desired TRO frequency variation to maintain or restore lock.

Similar to existing SC-based regulators [85, 104, 105], V_{dd} control is performed through *implicit* linear-regulation: modulating SCVC R_{out} to adjust the IR drop at its output to achieve the desired output voltage (Figure 4.5). R_{out} adjustment is performed by modulating the switching frequency of the SCVC (f_{sc}). Enabling fine-grain V_{dd} adjustment therefore requires high resolution f_{sc} control. The complexity of a high-resolution SCVC-DCO was avoided by exploiting the significant frequency difference between f_{sc} (>5MHz) and the system loop bandwidth (<20kHz), which allows for the use of a digital Delta-Sigma Modulator (DSM). The DSM allows the 16-bit DLF output to be asserted through a dithering 7-bit SCVC-DCO.

Using a V_{dd} powered oscillator results in significant deviation in phase and frequency due to the noisy supply. Furthermore, DVFS transitions can cause significant phase error as the system undergoes a transient change in f_{clk} to lock to a new target frequency. These deviations can

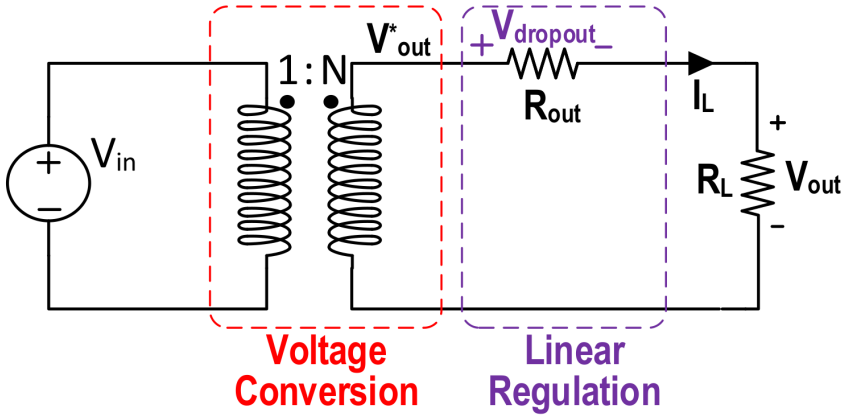


Figure 4.5: Equivalent circuit model of an SCVC. Transformer windings model the ideal voltage conversion ratio achieved by the converter. The non-zero output resistance, R_{out} is an inherent consequence of charge transfer between flying capacitors with non-zero potential difference

easily exceed the limited 2π capture range typical of traditional TDCs (1 *REFCLK* cycle) [106]. The prototype UniCaP-SC implementation therefore employed a TDC with an 8-cycle capture range (Figure 4.4). TDC implementation details are discussed in Section 4.2.

UniCaP-SC provides additional benefits over those offered by the generic UniCaP architecture, by overcoming a key challenge facing practical SC-regulator based systems : poor transient response [102], which requires additional V_{dd} margin. Efforts to achieve continuous V_{dd} scalability by modulating R_{out} in Figure 4.5 only exacerbates these voltage droop challenges, requiring larger margins, and thus undermining the benefits of continuous V_{dd} scaling. A more detailed treatment of droop degradation with scaled V_{dd} is provided in Appendix 4.6. An alternative approach involves using a series LDO to provide the required transient response [102, 103, 107]. However, the voltage overhead required to support LDO operation negates fine-grained V_{dd} scalability benefits, particularly in low-voltage applications. As we demonstrate in this paper, by largely decoupling V_{dd} margin requirements from the supply droop magnitude, UniCaP-SC enables efficient, continuous V_{dd} scaling without either the associated supply noise margins, or the need for a series LDO.

Figure 4.6 shows a linearized system block diagram of the UniCaP-SC FLL. H_{SC} and H_{TRO}

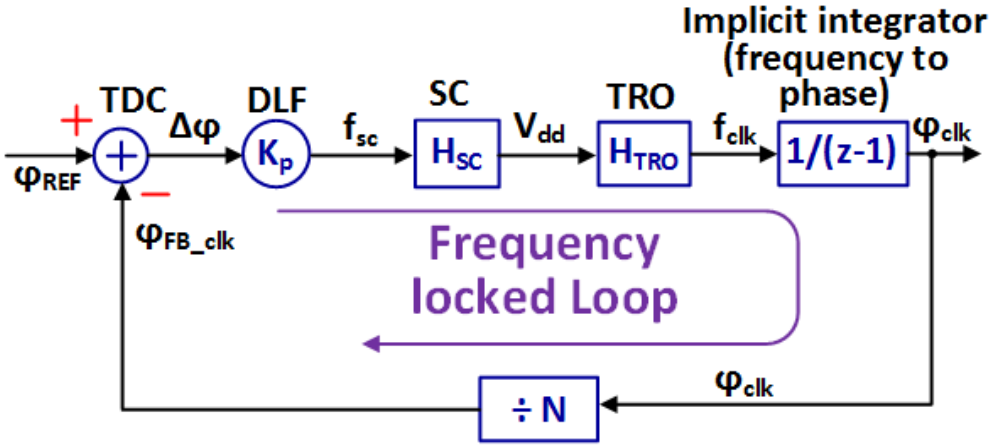


Figure 4.6: Simplified transfer function block diagram of UniCaP-SC loop

represent the transfer function of SCVC (f_{SC} to V_{dd}) and TRO (V_{dd} to f_{clk}) respectively. At steady state, the FLL guarantees f_{clk} = N· f_{REF}. The system response at the output of the SCVC, modeled as a single-pole system:

$$H_{SC}(z) = \frac{K_{SC} \cdot z}{(z-p)}, \tag{4.1}$$

with p being determined by the decap and resistance seen at V_{dd} and K_{SC} indicating the DC gain. In the current implementation, the settling time of SCVC is found to be negligible compared to T_{REFCLK}, resulting in $p \to 0$. The resulting pole-zero cancellation allows H_{SC} to be further approximated as a multiplier with gain K_{SC} for relatively moderate loop gain values. Similarly, H_{TRO} can be simplified as a constant with gain K_{TRO}.

Simplification of H_{SC} and H_{TRO} yields the following expression for the forward gain:

$$G_1(z) = \frac{\phi_{clk}(z)}{\Delta\phi(z)} = \frac{K_P \cdot K_{SC} \cdot K_{TRO}}{(z-1)}. \tag{4.2}$$

The closed-loop transfer function can therefore be written as:

$$\frac{\phi_{clk}(z)}{\phi_{REF}(z)} = \frac{G_1(z)}{1+N \cdot G_1(z)} = \frac{K_P \cdot K_{SC} \cdot K_{TRO}}{z-(1-N \cdot K_p \cdot K_{SC} \cdot K_{TRO})}. \tag{4.3}$$

Root-locus analysis of the resulting system indicates a significant range of loop gain values for which the system provides a stable over-damped response.

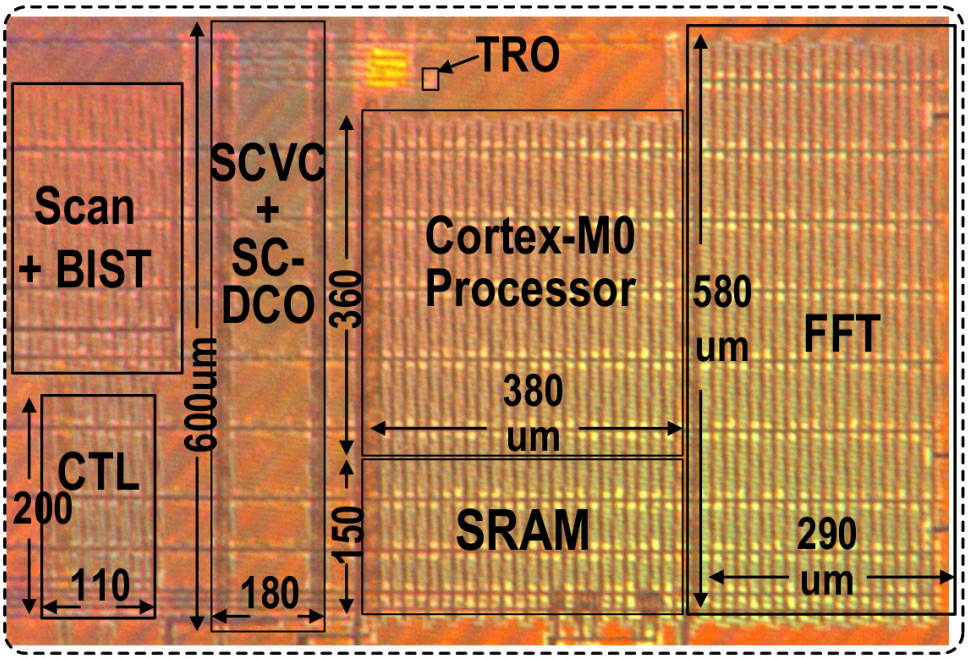
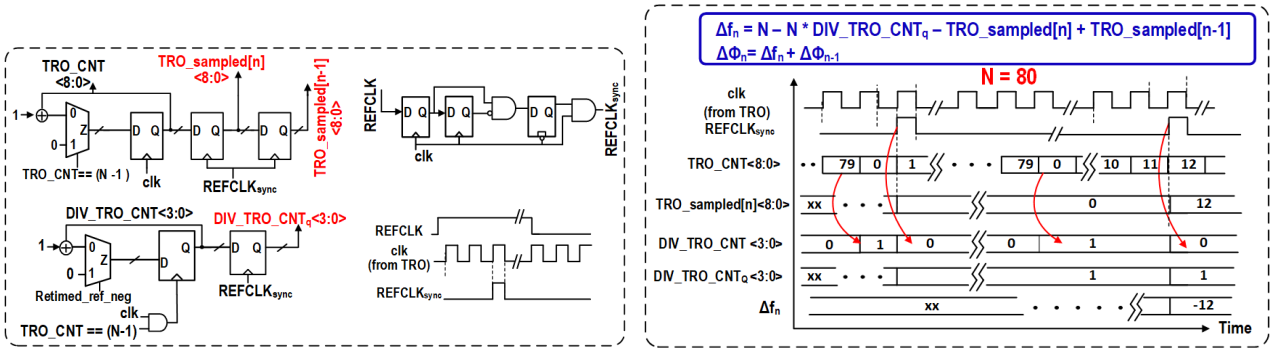


Figure 4.7: Die photograph of the UniCaP-SC test-chip.

A PI-control strategy can also be employed in UniCaP-SC, realizing a type-II PLL. Phase-lock enables recovery of lost cycles during droop events, and is typically required in several real-time applications. Achieving phase-lock, however, requires TDC design with a wide locking range to avoid cycle-slipping resulting from a noisy supply. Additional considerations associated with achieving phase-lock are outside the scope of this work and are demonstrated in [91].

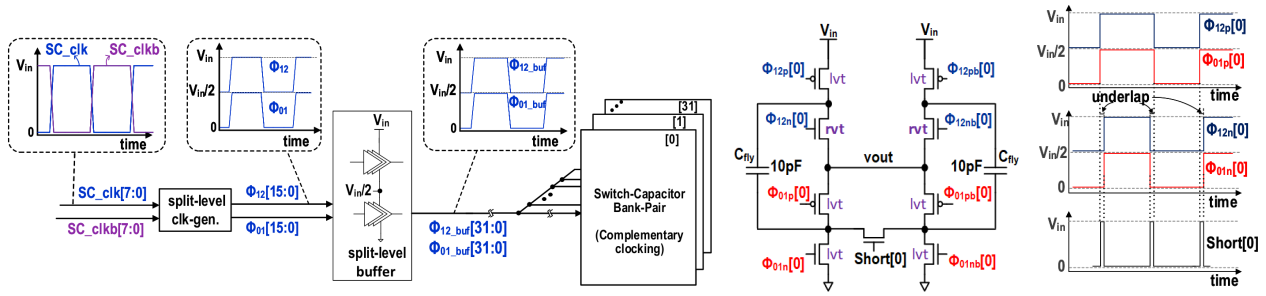
4.2 Test Chip Architecture

The UniCaP-SC architecture was implemented in a 65nm CMOS test-chip (Figure 4.7). This section describes the design of key test-chip modules. To better characterize UniCaP-SC impact on low-power computing, the test-chip included a fully-integrated SCVC, and a Cortex-M0 processor core with an FFT accelerator. The SCVC input (V_{in}) was fixed at 1.2V. Modulating SCVC R_{out} yielded V_{dd} scalability in the 0.44V-0.56V range with approximately 16 bits of precision, corresponding to fine-grained f_{clk} control over the 5MHz-30MHz range. *REFCLK* frequency was fixed at 192kHz,



(a) (b)

Figure 4.8: (a) Gate-level schematic of the proposed wide-range TDC (b)Timing waveform of different registers in the TDC with equation for output phase difference.



(a) (b) (c)

Figure 4.9: (a) Overall organization of the SCVC with 32 bank-pairs and a split-level clock generator. (b) Schematic of a switched-capacitor bank-pair (c) Waveform of gate signals in switched-capacitor bank.

informed by the operating range of the processor in NTV (tens of MHz), and the availability of a crystal reference. Integrated Built-in Self Test (BIST) was used to record snapshots of internal state variables in successive clock cycles during transient events, and to simplify calibration of the TRO and SCVC-DCO. Importantly, no explicit decap modules were introduced or required, either on- or off-chip in the design of this prototype. The quantity of implicit decoupling capacitance (including

the capacitance from FFT) is approximately 250pF.

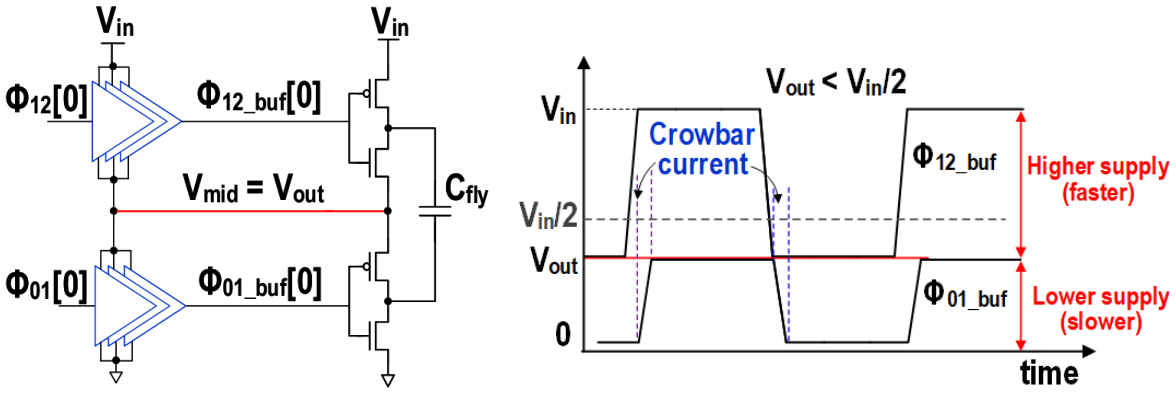
The remainder of this section examines the design of key components of the UniCaP-SC test-chip.

4.2.1 Time-Digital Converter

The TDC plays a key role in enabling the UniCaPcontrol loop. Wide capture range is a key requirement to acquire and maintain lock with a noisy oscillator supply (V_{dd}). Figure 4.8a outlines the design of the TDC. We employed a coarse-grained cycle counting TDC with a readily extensible capture range. More complex TDCs providing a similar capture range at higher-resolution may be used for applications with more stringent requirements [17]. The cycle-counting TDC relies on a re-timed version of *REFCLK* ($REFCLK_{SYNC}$) to determine the number of *clk* cycles that occur in one *REFCLK* cycle. $REFCLK_{SYNC}$ is obtained by sampling *REFCLK* using two flip-flops timed by *clk*. Retiming *REFCLK* is essential for avoiding metastability arising from the asynchrony between *REFCLK* and the TRO clock during lock. The feedback counter value at each $REFCLK_{SYNC}$ rising edge is captured as $TRO\_sampled[n]$. The captured feedback counter value, combined with the divider-ratio N is used to determine phase error within 1 DCO cycle. In the event that the phase error exceeds one *REFCLK* cycle, the cycle counter (TRO_CNT) reaches $N-1$. At this time, TRO_CNT is reset, and a coarser counter, DIV_TRO_CNT is advanced to quantize the time-delay between the divided TRO clock and *REFCLK* in units of T_{REFCLK} (Figure 4.8b). DIV_TRO_CNT is sampled, and subsequently reset after the arrival of every new $REFCLK_{SYNC}$ edge. This sampled value of DIV_TRO_CNT ($DIV\_TRO\_CNT_q$) is used along with $TRO\_sampled$ to determine the phase error.

4.2.2 Switched Capacitor Voltage Converter (SCVC)

Figure 4.9 shows the organization of the SCVC module. The voltage converter consists of 32 SC bank-pairs. A distributed SCVC-DCO enables efficient 8-way phase interleaving, with each phase driving four bank-pairs. Each bank-pair contains two 10pF flying capacitors, amounting to a total of 640pF for the converter. Figure 5.14a shows the simplified schematic of one bank-pair. The

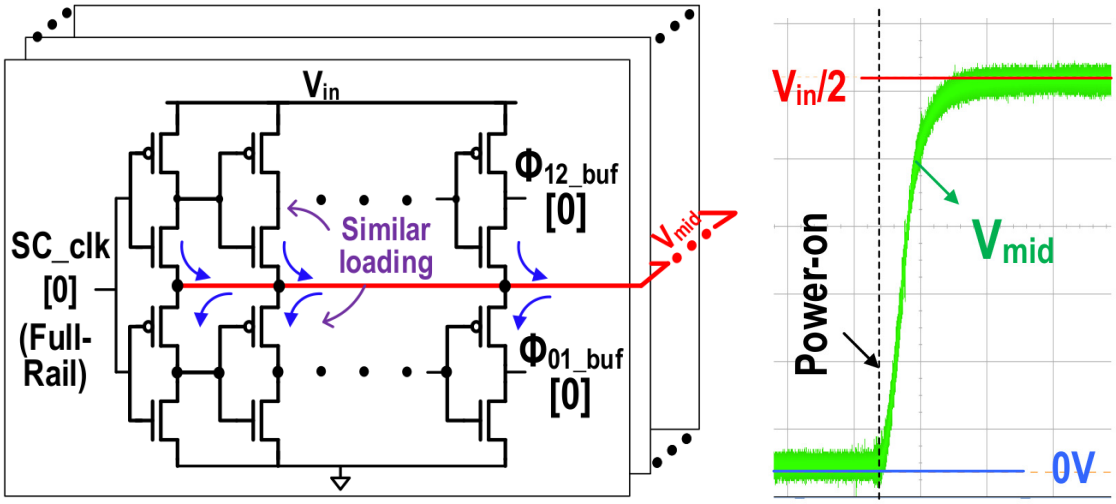


(a) (b)

Figure 4.10: (a) Simplified schematic using switched-capacitor V_{out} as V_{mid} and (b) resulting inter-level clock skew that leads to overlapping conduction phases.

bank-pairs are driven by complementary sets of clocks, with each bank clocked by under-lapping split-level clock phases ($\phi_{12p}, \phi_{12n}, \phi_{01p}, \phi_{01n}$) to avoid shorting the capacitor at the onset of clock transitions (Fig. 5.14b). Bottom-plate charge recycling was employed [108, 109] to reduce the considerable switching losses due to parasitic capacitance on the bottom-plates of C_{fly}. The shorting switch is pulsed during a dead-time when neither of the complementary banks is connected to the load. The SCVC switches nominally employ low V_{th} (LVT) devices to provide reduced switching losses at comparable resistance to their nominal V_{th} (RVT) counterparts. However, the NMOS transistor of the upper split-rail (driven by ϕ_{12n}, ϕ_{12nb}) was implemented as an RVT device to avoid any notable leakage current increase at lower V_{dd}.

Distributing the switched capacitor clock across the 32 bank-pairs of the SCVC is a significant source of switching power dissipation. To avoid distributing a full-swing (0-V_{in}) clock, split-level clock distribution enabled by a mid-rail voltage (V_{mid}) is commonly used [104]. In this system, clocks ϕ_{12p} and ϕ_{12n} are distributed using the $V_{in} - V_{mid}$ supply, while ϕ_{01p} and ϕ_{01n} are distributed using the $0 - V_{mid}$ supply (Figure 4.9a). A common approach to avoiding an external power supply for V_{mid} involves using V_{dd} as an approximate $V_{in}/2$ connection for the mid-rail in 2:1 SCVC designs [104] (Figure 4.10a). However, this approach is inefficient for run-time V_{dd}-scaling applications. Figure 4.10b illustrates the condition when $V_{dd} \neq V_{mid}$. An imbalanced mid-rail



(a) (b)

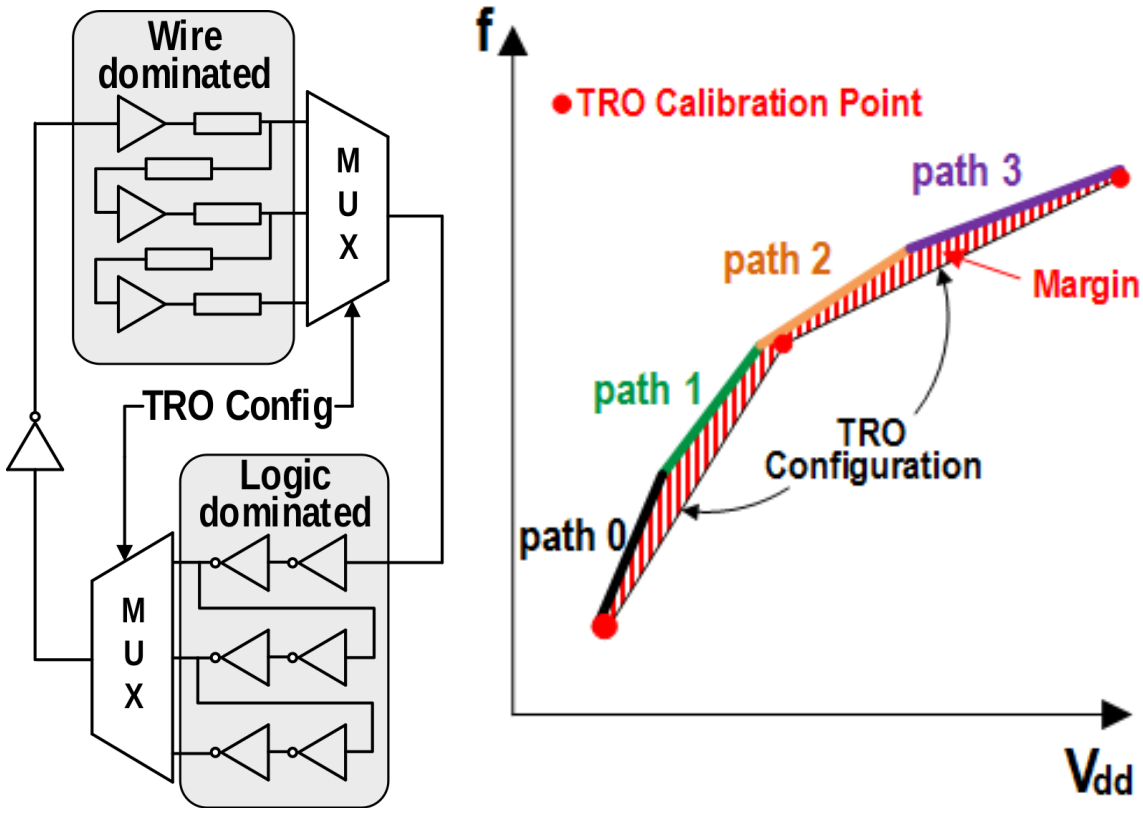
Figure 4.11: (a) Proposed split-rail charge-recycling with floating V_{mid} connected across all phases, (b) Measured oscilloscope trace of floating V_{mid} scheme.

voltage allows ϕ_{12p} and ϕ_{12n} to be distributed faster than ϕ_{01p} and ϕ_{01n}, leading to inter-level clock skew. The resulting clock overlap between the phases causes short-circuit current in the converter, leading to efficiency degradation.

We implement a straightforward alternative to setting V_{dd} as V_{mid}: The mid-rail is connected between all 8 phase-pairs of the clock distribution, and left to float (Figure 4.11a). Charge-recycling between similar capacitive load in the upper and lower voltage domains maintains $V_{mid} \approx V_{in}/2$ independently of V_{dd}. A similar approach has been independently proposed in [110]. Figure 4.11b shows an oscilloscope waveform capture of the mid-rail ramping up to $V_{in}/2 = 0.6V$ upon power-up in a test-setup where V_{dd}=0.4V.

4.2.3 Tunable Replica Oscillator (TRO)

The TRO is designed to match the delay and voltage sensitivity of the critical path. Although UniCaP-SC heavily reduces V_{dd} margins, it does not eliminate them altogether. Margining is performed by introducing additional delay into the TRO, which implicitly increases V_{dd} through the



(a) (b)

Figure 4.12: (a) Schematic of TRO consisting of a programmable number of logic-dominated delay cells and wire-dominated cells. (b) TRO calibrated to match V_{dd}-dependent critical paths (path 0–3). Limited V_{dd} characterization leads to imperfect modeling and requires a modest amount of additional margin.

action of the PLL to allow a slower (margined) TRO to oscillate at f_{clk}, guaranteeing sufficient slack for the critical path. The TRO construction, outlined in Figure 4.12a, is conceptually identical to that used in [93]. MUXs allow the TRO to be configured with a programmed quantity and balance of logic-dominated (high V_{dd}-sensitivity) and wire-dominated (low V_{dd}-sensitivity) gates to provide variable delay and sensitivity. Different paths within the design become critical at different V_{dd} points (paths 0–3 in Figure 4.12b). Practical considerations involving test-time and cost require the TRO to be configured to match the critical path at only a limited number of V_{dd} values, leading to a

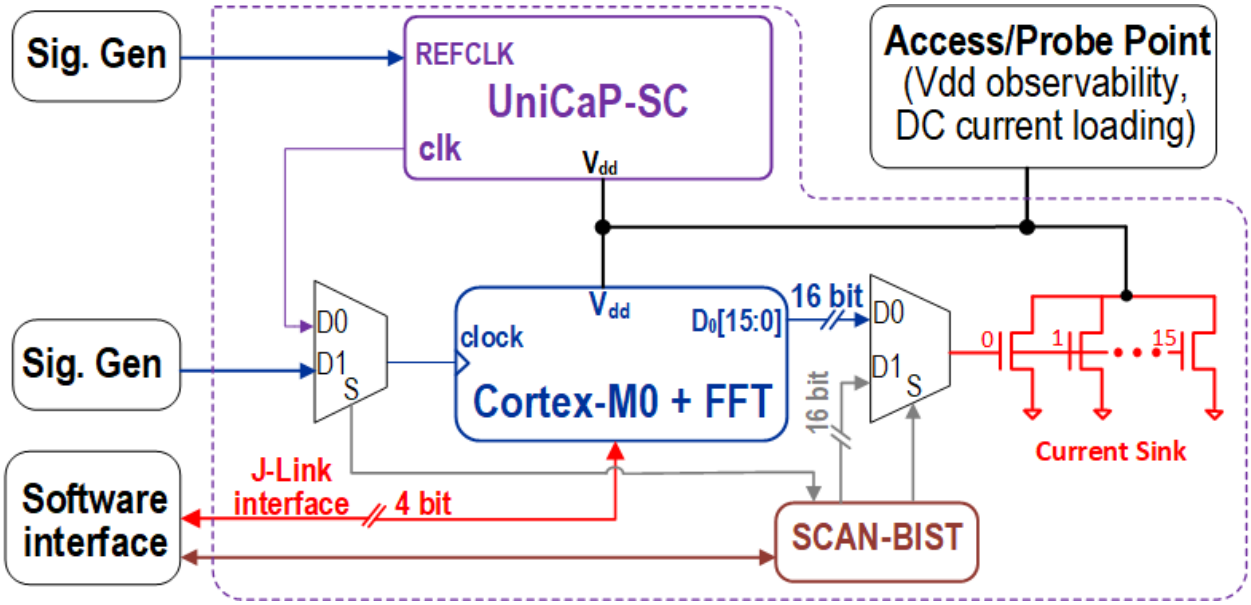
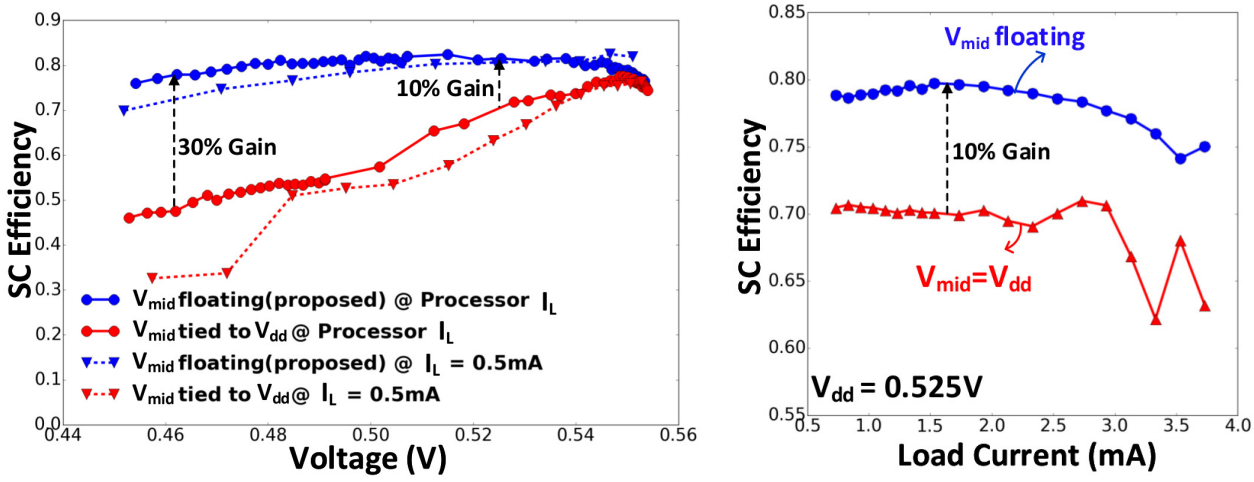


Figure 4.13: Instrumentation and test-setup for the UniCaP-SC test-chip.

less-precise but conservative representation of the critical path (Figure 4.12b). Although modest, some margin is still required to account for delay and sensitivity deviation between the TRO and the critical path away from these characterization voltages. Another factor requiring some additional margin is the limited resolution in configuring TRO delay and sensitivity. Managing the efficiency and cost of TRO characterization is expected to be identical to existing production designs relying on tunable replica circuits [42, 93, 111].

4.2.4 Digital Load

Current loading, efficiency and performance characterization features are provided by a Cortex-M0 processor, a dedicated FFT accelerator module, and a synthetic load for directed load current transients (Fig. 4.13). Although the TRO clocks both, the processor and the FFT module, system critical paths were found to lie within the processor over the entire targeted V_{dd} range of 0.44V-0.56V. Typical of sub-threshold and NTV systems, a separate 0.8V supply is used to power on-chip SRAM, required for instruction and data memory. A separate power supply allows V_{dd} to scale below the



(a) (b)

Figure 4.14: Measured SC-converter efficiency vs. (a) V_{dd} under both fixed and V_{dd}-dependent I_L, and (b) I_L using two split-rail clock distribution variants: (i) proposed with V_{mid} floating and (ii) V_{mid} connected to V_{dd}.

minimum operating supply voltage required for correct SRAM operation (V_{min}). The synthetic load consists of binary-weighted NMOS switches connected between V_{dd} and 0. The load is memory mapped to the processor, allowing targeted load-current waveforms to be generated through processor program execution, which performs the desired sequence of memory write operations to the synthetic load register. The load module is also scan-programmable, readily enabling external dc-loading and current-step response measurements. In addition to the integrated BIST and synthetic load modules, an external access point was used to deliver precise, V_{dd}-independent loading to perform some of the reported experiments.

4.3 Measurement Results

All reported test-chip measurements in this section are qualified by validating functional correctness of the processor while executing *fmax and speed-indicative* benchmarks provided by ARM. Instead of detailed TRO calibration at multiple V_{dd} points, a single-point V_{dd} calibration was performed at 0.5V to incorporate margins associated with imperfect TRO-critical path-tracking across V_{dd}.

4.3.1 SCVC Efficiency

Maintaining converter efficiency across the entire target V_{dd} range is a major objective for voltage-scalable systems. Efficiency is defined as the ratio of output power to total input power:

$$\eta_{SC} = P_{out}/P_{in} = V_{dd} * I_L/(V_{in} * I_{in} + P_{Overhead}), \tag{4.4}$$

where, $P_{Overhead} = P_{DCO} + P_{DSM} + P_{DLF} + P_{TDC} + P_{TRO}$. P_{DCO} and P_{DSM} are the SCVC-DCO and delta-sigma power dissipation respectively. We conducted experiments to evaluate η_{SC} across the operating V_{dd} range, and separately across I_L. In this test chip, I_L accounts for the total load current, inclusive of the *TRO, TDC, DLF* and *Controller* modules in addition to the processor and FFT modules.

Measurements for η_{SC} across V_{dd} were performed under two different current loading scenarios: constant 0.5mA I_L, and the more realistic load current due to processor execution. Two different SCVC clock distribution configurations were also evaluated: with a floating mid-rail and with V_{mid} assigned to V_{dd}. Figure 4.14a shows the four resulting SCVC converter efficiency curves corresponding to different configurations, across a V_{dd} range of 0.45V-0.55V. All curves exhibit a characteristic η_{SC} maximum, determined by the optimal balance between increased switching losses at higher V_{dd} (where maintaining the needed low R_{out} requires increasing SCVC switching frequency), and increased conduction losses at lower V_{dd}. As expected, constant I_L curves decline faster at lower V_{dd} compared to the processor loading curves. This trend is due to increased conduction losses as load current does not decrease with scaled V_{dd}. In contrast, the efficiency decline is more gradual for the case of processor-driven I_L due to conduction loss relief provided by near-exponential I_L reduction with V_{dd}. The figure also shows the significant η_{SC} improvement achieved by the floating mid-rail over the V_{dd} mid-rail configuration. Optimal η_{SC} for the V_{dd} mid-rail also occurs at a much higher V_{dd}, before sharply reducing due to the efficiency degradation from crowbar current in the converter from slight inter-rail voltage mis-match (Figure 4.10b). Overall, the floating mid-rail approach provides an efficiency benefit of up to 30% as V_{dd} scales in this test-chip.

Figure 4.14b shows SCVC efficiency versus I_L under two configurations: floating mid-rail, and one with $V_{mid} = V_{dd}$. The SCVC input (V_{in}) and output voltages (V_{dd}) were maintained at 1.2V and

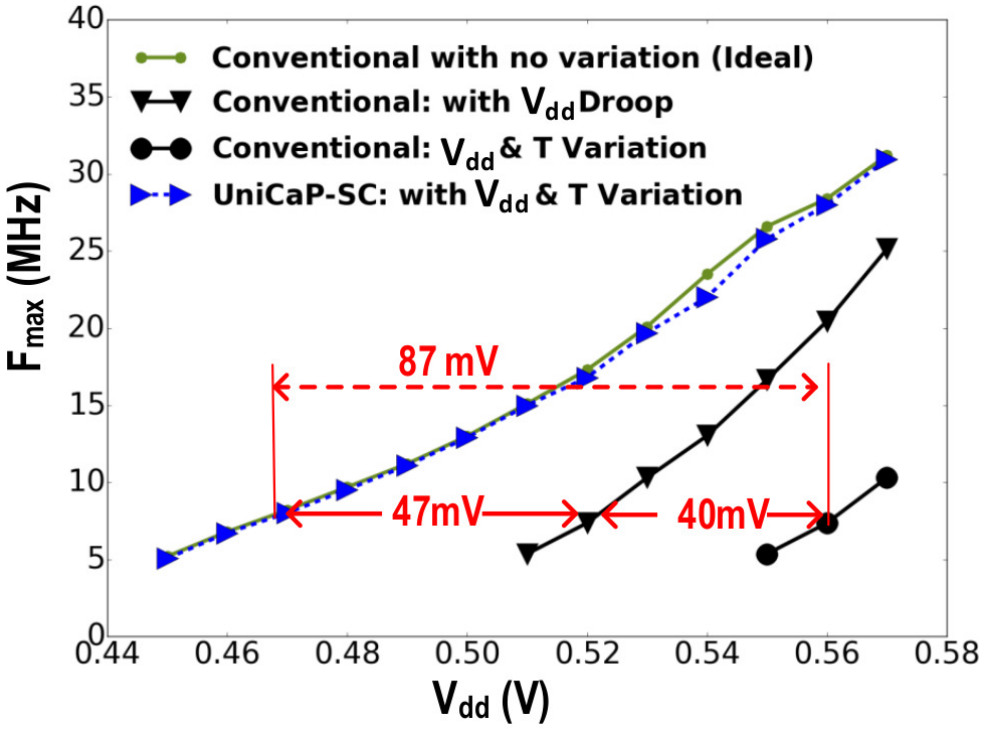


Figure 4.15: Measured maximum f_{clk} (f_{max}) vs V_{dd}.

0.525V respectively. Both curves exhibit an overall downward trend as I_L increases, stemming from both, increased conduction losses and switching losses associated with higher SCVC switching frequencies to reduce R_{out}, required to maintain V_{dd} despite increased I_L. The observed efficiency improvement with the floating mid-rail is consistent with that observed in Figure 4.14a. For SCVC efficiency measurement the current from TRO, TDC, DLF, Controller are all included in I_L.

4.3.2 f_{max} vs. V_{dd}

Aggressive reduction of V_{dd} margins for supply noise and temperature variation are the most significant contribution of UniCaP-SC. To quantify these benefits on energy dissipation or performance, we performed system f_{max} tests, identifying the maximum frequency for which the system remains functional across the operating V_{dd} range (Figure 4.15). Measurements were obtained under two configurations: (1) *Conventional*, where the test-chip operates with the traditional two-loop ap-

proach using an external fixed-frequency clock source (Figure 4.13); and (2) *UniCap*, using the unified control loop. Measurements were also obtained for each configuration under V_{dd} droop conditions (corresponding to a modest 100% increase in current load) and temperature variation in the range from -15°C to 45°C. f_{max} measurements under an ideal configuration, with no supply noise or temperature variation are also reported. As seen in the figure, supply-voltage and temperature fluctuations require a significant V_{dd} margin in the conventional system: At 7MHz, supply noise margins of 47mV (corresponding to 8.39% of V_{dd}), and temperature margins of 40mV (7.14% of V_{dd}) are required. In contrast, the UniCaP-SC curve nearly overlays the *ideal* curve, recovering nearly all of the associated supply margins. On average 94% margin recovery is achieved.

Figure 4.15 also reveals another important and counter-intuitive observation about switched-capacitor converters: Voltage margins due to supply droop *increase* with reduced V_{dd}, in both absolute and percentage terms. This observation contradicts expectations of reduced margins at lower V_{dd} arising from a super-linear reduction in load current (exponential in the case of near- and sub-threshold circuits) at scaled supply voltages. Indeed, even measurements by the authors on using buck converter-regulated processor test-chips [91] corroborate the conventional wisdom that V_{dd} margins reduce with supply-voltage. This apparent contradiction in the case of SC converters is attributable to the use of resistance control in the SC to arrive at the target V_{dd} as explained in more detail in the Appendix.

4.3.3 Temperature Tracking

Figure 4.16 shows the impact of temperature variation on required supply margins. Data for the *Conventional* curve was obtained by identifying the minimum V_{dd} required for correct operation of the processor at 9.2MHz across a temperature range from -30°C to 105°C. UniCaP-SC data-points were obtained by recording V_{dd} as determined by the control loop for error-free processor operation at 9.2MHz over the same temperature range. A conventional two-loop system must operate at the maximum recorded V_{dd} to operate across temperature variation. However, by autonomously tracking temperature, UniCaP-SC affords opportunistic energy-savings by down-scaling V_{dd} at higher temperatures. Over an operating range from -10°C to 40°C, UniCaP-SC provides 40mV of

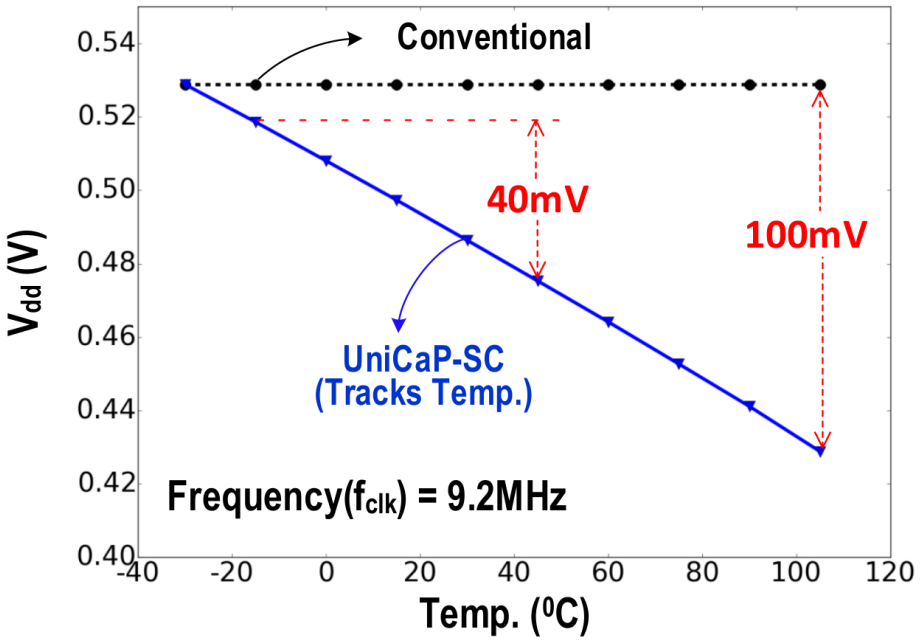


Figure 4.16: Measured V_{dd} vs. Temp. for a target f_{clk} of 15MHz across the entire Temp. range.

opportunistic V_{dd} margin reduction, approximately 8% of V_{dd}.

4.3.4 Minimum Total Energy-per-Cycle (EPC)

Figure 4.17 shows the minimum EPC obtained for the complete system (load + regulator + PLL) versus f_{clk} in both, *Conventional* and UniCaP-SC configurations. Energy measurements were taken only under supply noise conditions at a fixed temperature. The curves illustrate the interactions between switching losses, conduction losses and the EPC contribution of the processor load, as a function of V_{dd} (and therefore f_{clk}). UniCaP-SC provides a minimum EPC improvement of 7% over conventional design, with more significant savings at higher operating frequencies. The measured reduction in the Minimum Energy Point (MEP) energy is observed to be 12.3%. EPC savings below 6MHz are not measured because the conventional system is not functional at the corresponding V_{dd}.

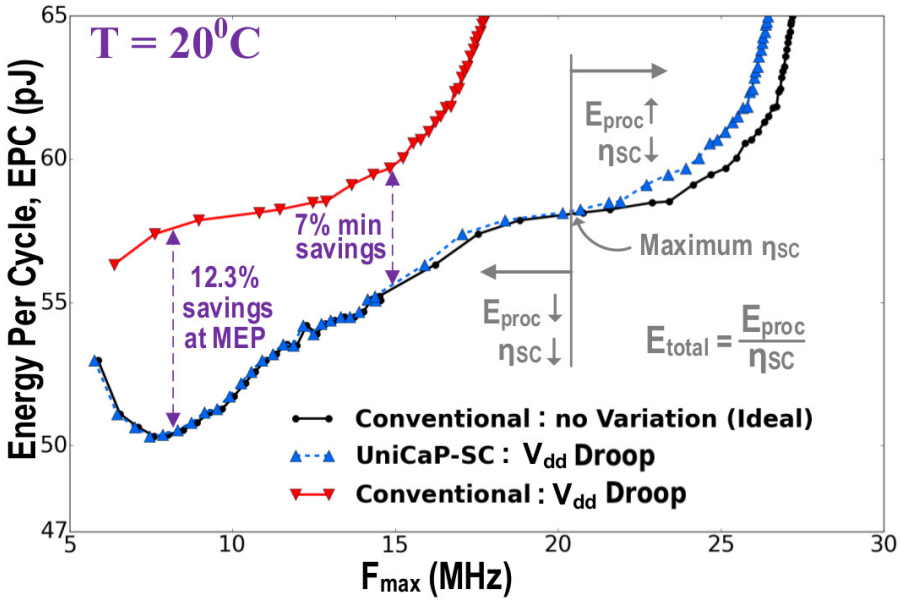


Figure 4.17: Measured Energy per Cycle(EPC) of the system vs f_{clk} plot.

4.3.5 Transient Response

Figure 4.18 shows captured V_{dd} oscilloscope waveforms in response to a 1mA/1ns current ramp-up and ramp-down. Stable operation is observed during both transients. The asymmetry in the voltage response between ramp-up and ramp-down transients is attributable to a non-linear TRO-V_{dd} versus f_{clk} characteristic, which in-turn impacts loop gain. Reduced-loop gain translates to slower transient response, leading to increased droop before the system recovers f_{clk} through V_{dd} control. Operation under random current loading was also evaluated by generating random numbers in the processor, and writing them into the synthetic load register. Figure 4.19 shows a section of an oscilloscope-captured V_{dd} waveform. Stable system operation was observed during random loading in an experiment that lasted over 10 million processor cycles. Figure 4.20 shows measured waveforms of V_{dd} and the *clk* in the event of a voltage droop, demonstrating an instantaneous clock stretch during a supply droop.

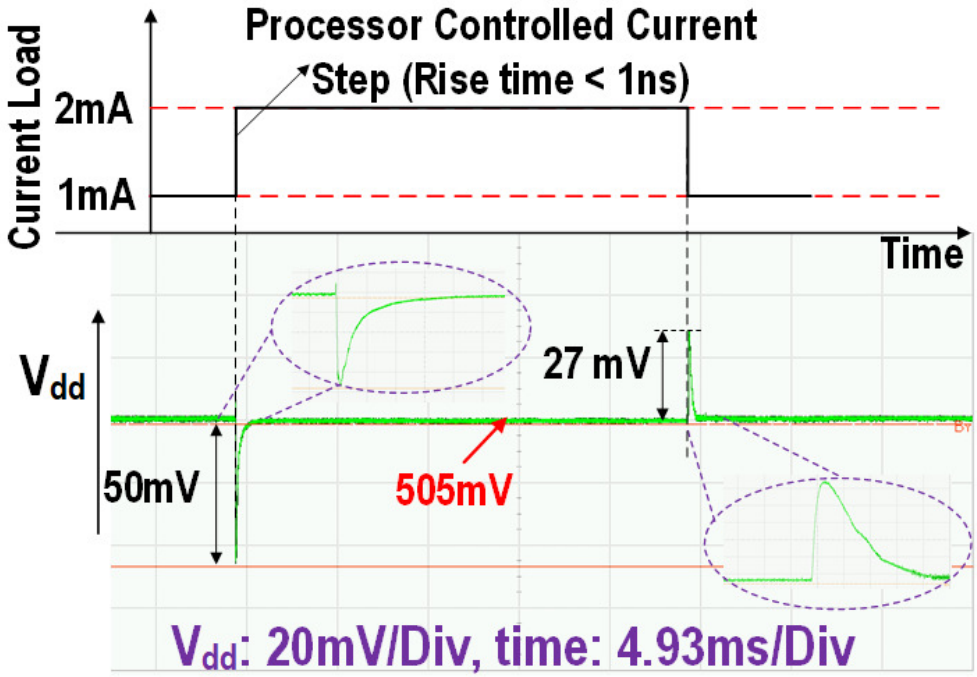


Figure 4.18: Measured oscilloscope trace demonstrating transient V_{dd} droop and surge response from a current step of 1mA.

4.3.6 On-the-fly DVFS

One of the key benefits afforded by UniCaP-SC is its ability to support on-the-fly DVFS, allowing the processor to operate un-interrupted while the system transitions to a new target frequency. Unlike traditional DVFS however, transition between performance states in UniCaP-SC only involves selection of the new target f_{clk}: V_{dd} is determined autonomously by the system. Instead of a traditional voltage-frequency table therefore, UniCaP-SC typically relies on a frequency-TRO configuration table, where characterized delay and sensitivity settings for the TRO are pre-programmed, analogous to a voltage-frequency table in conventional systems. Figure 4.21 shows oscilloscope-captured V_{dd} waveforms during DVFS transition as the system is provided with a sequence of different f_{clk} targets during continuous processor operation through changes in N. UniCaP-SC is seen to adjust V_{dd}, so as to lock to the new frequency target after each transition. At

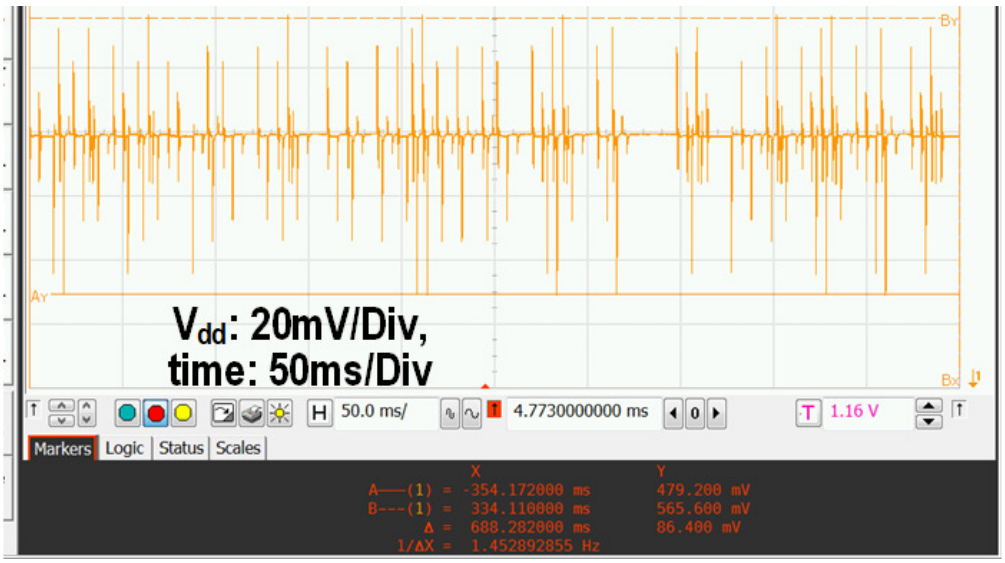


Figure 4.19: Measured oscilloscope trace demonstrating system transient response to random frequent supply droop and surge.

the end of these transitions, the processor was verified to have performed all computations without error.

4.4 Discussion

In this section, we discuss some important observations, features, and existing limitations of the proposed UniCaP-SC architecture and test-chip.

Although UniCaP-SC has demonstrated aggressive V_{dd} margin reduction due to supply noise and temperature variation, the need for a relatively modest, but essential guard-band to ensure correct operation persists due to several factors. Limited precision of the TRO module requires some conservatism in establishing TRO settings. For systems experiencing very large droop, V_{dd} may transition to regions where new critical paths are exercised, without an accompanying TRO configuration update. The resulting mismatch also contributes to some additional voltage margin. Within-die PVT variation between the critical path and the TRO cannot be addressed by UniCaP-SC.

UniCaP-SC effectively addresses chip-mean variation through use of the TRO to track the critical

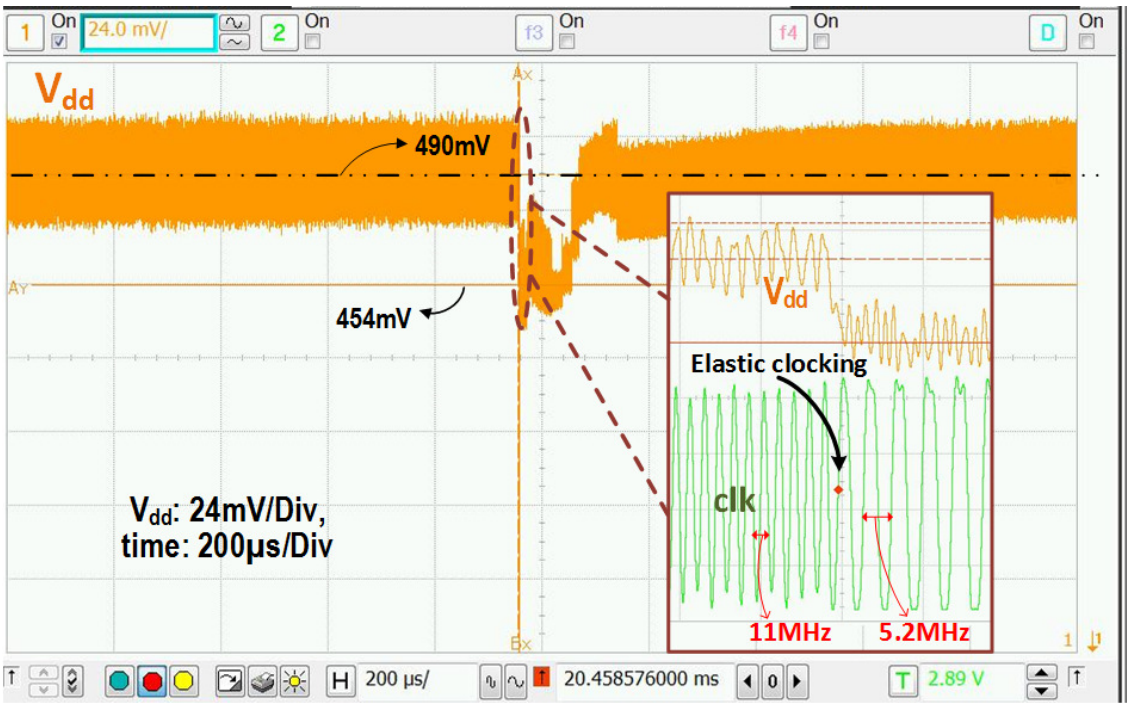


Figure 4.20: Measured oscilloscope capture demonstrating instantaneous clock-stretching during supply droop.

path on a chip. A one-point calibration, commonly performed on production silicon, will further allow the TRO to track a significant portion of on-die variation. Another factor that influences the need for additional guard-bands when using UniCap-SC is the latency of the clock distribution. If the clock distribution delay sensitivity to V_{dd} is similar to critical-path delay sensitivity to V_{dd}, then the clock distribution latency does not require any additional guard-bands [93]. If however, these relative sensitivities to V_{dd} are significantly different, a small additional guard-band may be required to account for the resulting mismatch in the modulated clock period at the sink-points of the distribution and the critical path delay.

UniCaP-SC matches the delay and sensitivity of critical logic paths to avoid timing failure. However, designs that employ a shared voltage domain for logic and memory face an additional restriction: V_{dd} cannot fall below V_{min}. The UniCaP-SC test-chip does not face this limitation since memory is powered by a separate power supply, typical of many NTV or sub-threshold

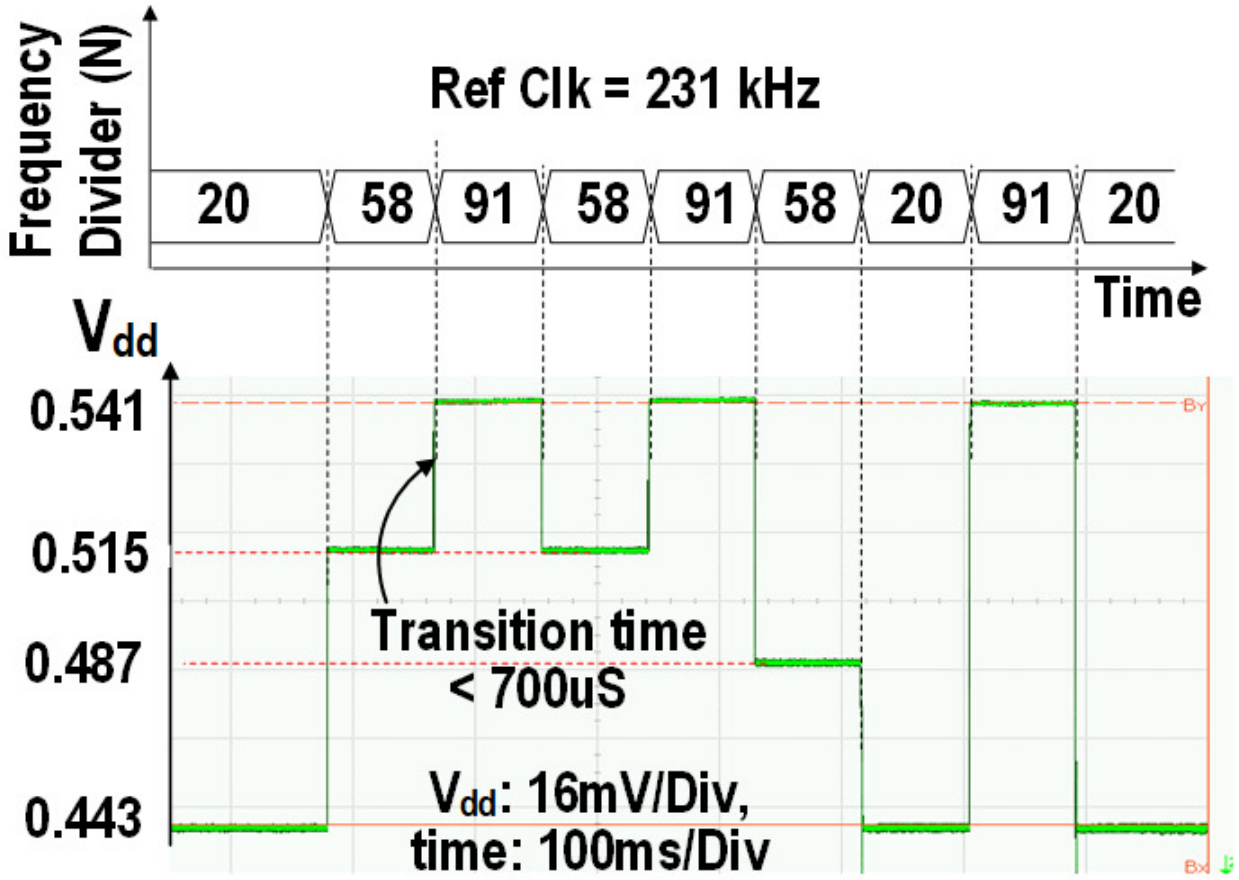


Figure 4.21: Measured oscilloscope trace demonstrating on the fly DVFS in UniCaP-SC system by changing N.

computing applications. Applications requiring a shared power supply must however, either address or circumvent the V_{min} challenge.

Phase interleaving, with individual banks synchronized by evenly spaced clock phases effectively reduces SCVC ripple [85, 105]. Ripple voltages contribute voltage overhead since the converter output must operate at a higher average voltage (with increased power dissipation) to guarantee timing slack at the minimum voltage. Although the UniCaP-SC test-chip employed an 8-way phase-interleaved converter, this choice was governed by the availability of pre-existing SCVC modules. Based on supply droop margin measurements, UniCaP-SC is expected to be compatible

with a more efficient and less complex single-phase SCVC design, requiring little or no additional V_{dd} margin to endure the accompanying increased ripple. The implications of an elastic clock on SCVC ripple have been discussed in [112].

Although UniCaP rejects the impact of V_{dd} droop and aggressively relaxes V_{dd} margins, avoiding functional failure of circuits at extremely low V_{dd} independently of timing slack motivates techniques for improved transient droop voltage response. Two options are readily available to the designer in this regard: (1) use of a more complex, variable-ratio SC converter to provide wide-range V_{dd} scalability without degrading the output impedance of the SC stage and (2) use of a higher f_{REFCLK} which, at the expense of higher switching losses, allows an increase in the control loop bandwidth and therefore transient response.

Finally, although no additional adjustment of TRO settings was required to achieve on-the-fly DVFS in the UniCaP-SC test-chip, enabling this capability on systems with a wider operating range, or a more diverse set of critical paths requires a sequence of measures: (1) Adjust TRO settings to reflect the critical path delay and sensitivity at the target frequency. A small additional "transition margin" in the form of TRO delay must be added. This margin addresses delay or voltage sensitivity mismatch that may exist between the newly selected TRO settings, and the current and intermediate critical paths that will surface as V_{dd} transitions to its steady-state value. (2) Change the PLL divider ratio, allowing the system to transition V_{dd} and f_{clk} to lock to the new target frequency, and (3) Once re-lock is achieved, remove the additional delay guard-band to remove the transition margin.

4.5 Conclusion

We presented UniCaP-SC, an architecture that unifies SC-based voltage regulation and clock regulation into a single control loop, producing a clock that adapts to across-chip supply droop and temperature variation, while maintaining system clock frequency. By efficiently tolerating supply droop, UniCaP-SC addresses transient response degradation with V_{dd} scaling in SC-based regulators to achieve efficient continuous V_{dd} scalability without a series LDO. A true all-digital 65nm UniCaP-SC test-chip consisting of an integrated 2:1 SCVC, an ARM Cortex-M0 processor, and an FFT accelerator was shown to recover an average of 94% of supply droop and temperature

margin, enabling a 16% reduction in V_{dd}.

4.6 Appendix: Voltage Droop at scaled V_{dd}

Figure 4.5 illustrates a simplified equivalent circuit for an SCVC. SCVCs *inherently* consist of two sequentially connected stages: voltage conversion and implicit linear regulation. The discrete, topology-driven SCVC voltage ratios ($N = V^*_{out}/V_{in}$) provided by voltage conversion stage are modeled by a transformer with the appropriate turns ratio (N). The subsequent linear regulation stage results from the non-zero R_{out} of the SCVC, determined by switch and flying capacitor sizes, and switching frequency [108, 113, 114].

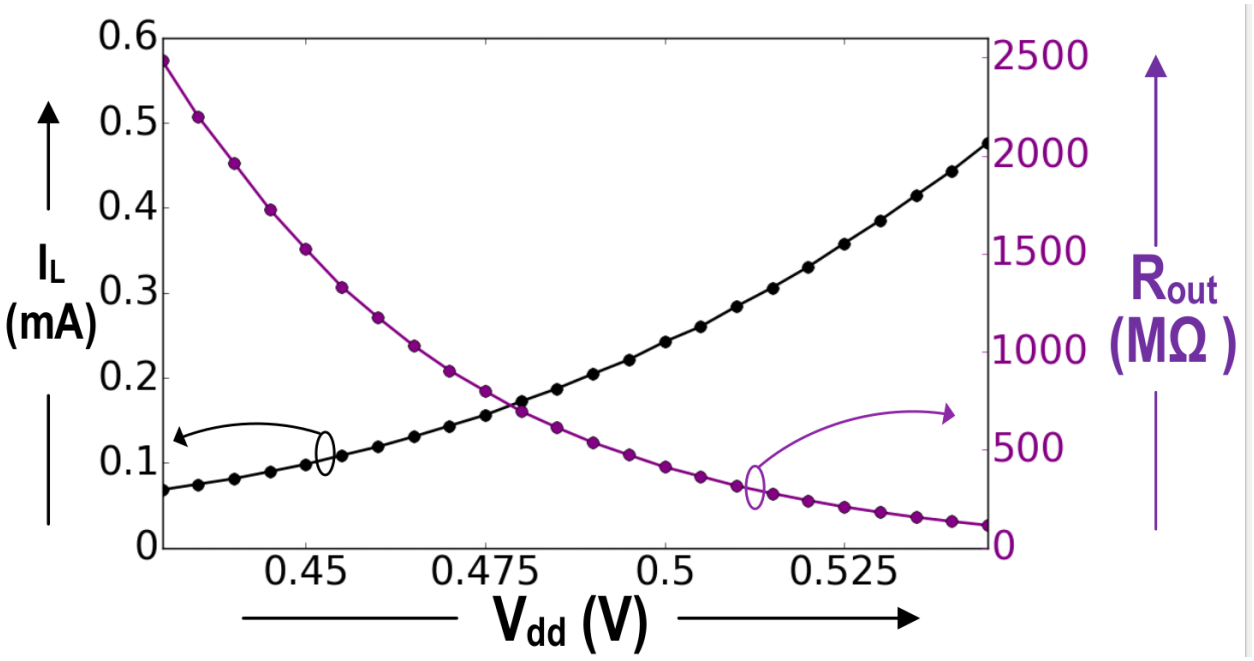


Figure 4.22: Measured R_{out} and I_L vs V_{dd} for Cortex-M0 microprocessor.

The output voltage of an SCVC can be controlled at runtime through two mechanisms: (1) dynamically varying flying capacitor (C_{fly}) configurations [104, 114, 115], effectively changing the turns ratio of the inductor model in Figure 4.5, and (2) linear regulation, by varying R_{out} based on the load current (I_L) to achieve a targeted $V_{dropout}$ so that $V^*_{out} - I_L R_{out} = V_{out}$.

Runtime re-configuration of C_{fly} adds significant complexity while still providing insufficient fine-grained voltage control. Exploiting the implicit linear-regulation capability of SCVCs is an attractive alternative for continuous voltage control, but incurs prohibitively worsening voltage droop as V_{dd} scales. Analysis shows that this trend is expected for regulators like SCVCs and Digital LDOS that rely on varying R_{out} to achieve a target V_{dd} operating point.

Figure 4.22 shows measured load current from a Cortex-M0 microprocessor across its operational V_{dd} range. Also shown is the R_{out} that the SCVC must achieve to meet the target V_{dd}. As V_{dd} scales, $V_{dropout} = I_L R_{out}$ increases, requiring an increase in R_{out} to meet the target V_{dd}. Unlike switched-inductor converters or analog LDOs, whose output impedance can be largely decoupled from their operating point, SCVCs are unable to decouple the relationship between R_{out} and their DC operating point. The same increased R_{out} required to achieve the target V_{dd} is also presented to the load-transient, worsening supply droop. The extent of the droop degradation depends on the amount of available decap and the loop response time. Nevertheless, the initial part of the droop, governed by the IR drop across the regulator, is significantly worsened. Importantly again, unlike buck converters or analog LDOs, declining I_L with reduced V_{dd} does not mitigate the effect of the droop. As seen in Figure 4.22, R_{out} must increase more precipitously with reduced I_L so as to maintain a desired $V_{dropout}$. The observations of worsening droop based on the above analysis are corroborated by measurement results presented in Figure 4.15 where required supply droop margins increase with reduced V_{dd}.

Chapter 5

COMPUTATIONALLY ENABLED MINIMUM ENEGY-PER-CYCLE TRACKING WITH PERFORMANCE REGULATION

Over the last few years there has been a substantial increase in the demand for ultra-low power devices in sensors and edge-computing in internet of things(IoT) and medical implants. These applications mostly rely on battery power or energy harvesting to ensure portability. For battery-powered devices minimizing the total energy can improve the battery life, size and weight. For energy harvesting, energy minimization is necessary to ensure improved performance with the scavenged energy. In both cases minimizing the total energy consumption is critical. One essential constraints for almost all real world system is maintaining the required performance. While the constraints can be different for different applications, guaranteeing operation over a specific frequency is necessary for almost all applications. Therefore energy minimization under performance constraints is the goal for ultra-low-power applications.

Supply voltage control is one of the most effective means of energy minimization. As the voltage V_{dd} scales down, the switching energy per cycle, which is proportional to ${V_{dd}}^2$, scales down quadratically [116]. This reasoning lead to aggressive V_{dd} scaling and sub-threshold operation [97, 117–120] where the circuits are operated below the threshold voltage (V_{th}). But aggressively scaling down V_{dd} does not always minimize the total energy per cycle(EPC). Researches [121, 122] have pointed out that as the V_{dd} scales down the leakage EPC begins to increase. In sub-threshold region, the leakage EPC becomes the dominant factor in total EPC and offsets the decreasing switching EPC as V_{dd} scales down 5.1. Because of these two opposing trend the total EPC becomes minimum at a specific voltage, V_{MEP}. System operation at voltage V_{MEP} and the corresponding frequency f_{MEP} (obtained from the voltage-frequency curve of the system) results in the minimum total EPC. Prior works [121, 123] have shown that operation in MEP point (voltage V_{MEP} and frequency

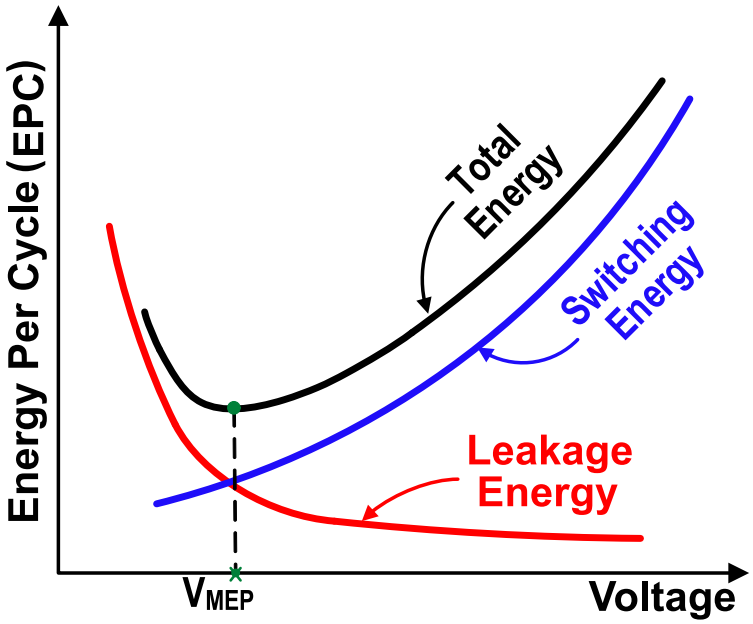


Figure 5.1: Existence of Minimum Energy-per-cycle Voltage (V_{MEP}) from the trade-off of leakage energy-per-cycle and switching energy-per-cycle

f_{MEP}) with duty cycled on time enabled with power gating can result in total minimum energy of the system. The MEP point of the system is not fixed over run-time. Researches [121, 122] have shown dependence of V_{MEP} on switching factor, process and temperature variation. Therefore, a run-time tracking of MEP point is necessary.

Several efforts in run-time MEP tracking have been reported in literature [124, 125]. One implementation [124] demonstrates MEP tracking across switching activity and temperature variation in a buck-converter based system. This approach relies on a large capacitor to run the system while finding the MEP. The capacitor is first charged to a specific voltage, which is sampled. The system then runs N cycles on the capacitor charge and the capacitor voltage is sampled again. The difference between the two sampled voltages indicates the energy consumed over N cycle. Through successive comparison of these energy measurements at different voltage, the system approaches MEP. Another work was focused on obtaining MEP by measuring the ratio of leakage power and switching power [125]. The *optimal leakage ratio* used in the work is dependent on temperature and process

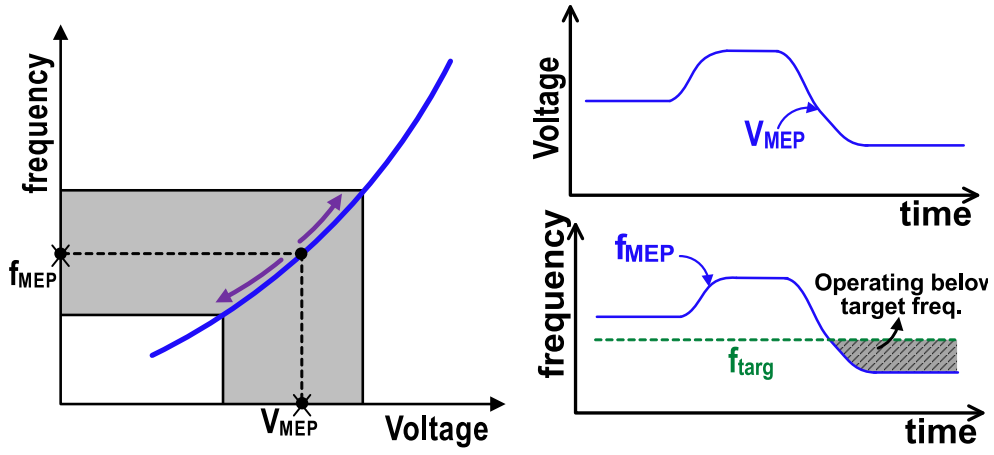
parameter and the work used the same *optimal leakage ratio* across different temperature for MEP tracking. Also in deriving the ratio, this work assumes frequency to be independent of V_{dd}, which is not valid assumption for most cases. Both [124] and [125] do not take into account the regulator energy losses and the proposed approaches will not work for a converter with non-uniform efficiency across voltage. Moreover, these works were not focused on building a performance constrained system, which is essential for almost all real-world applications.

We propose a fully-integrated system that can autonomously track the MEP point, resulting in minimum total EPC [20]. Our proposed system is a switched-capacitor(SC) regulator based ultra-low power system with operating voltage of 0.38-0.58V. We computationally determine the total EPC of the system (including the regulator losses) by measuring the necessary parameters. The system reaches MEP point through successive comparison of the total EPC at different V_{dd}s. The proposed methodology also tracks the changes in MEP due to variation in temperature, process and switching activity of the system. The proposed system is built upon robust-efficient Unified clock and power Architecture (UniCaP) [19, 126–128] that ensures performance regulation. The UniCaP architecture also drastically reduces the voltage guard-band for supply and temperature variation, reducing the energy consumption in guard-band. The performance constraint total EPC minimization has been demonstrated on a 65nm LP CMOS test-chip.

The rest of the chapter is organized as follows: In section 5.1, the considerations and challenges in building a true MEP tracking self-optimized system are addressed. The overview of the proposed system architecture and operation is presented in section 5.2. Section 5.3 discusses how MEP is searched at run-time. Implementation detail of the test-chip is described in section 5.4 before presenting the measurement results in section 5.5. Finally a brief discussion of some aspects of the proposed MEP tracking methodology is presented in section 5.6.

5.1 Challenges/Considerations in practical low-power systems operation in MEP

To implement an MEP-tracking system for real world application, several important factors must be taken into consideration. The EPC of a system can vary significantly during run-time. The leakage energy can change due to a change in the temperature. On the other hand, change in system load



(a) (b)

Figure 5.2: (a) V_{MEP} and f_{MEP} can change during run-time (b) f_{MEP} below the target frequency results in failure to meet performance requirement

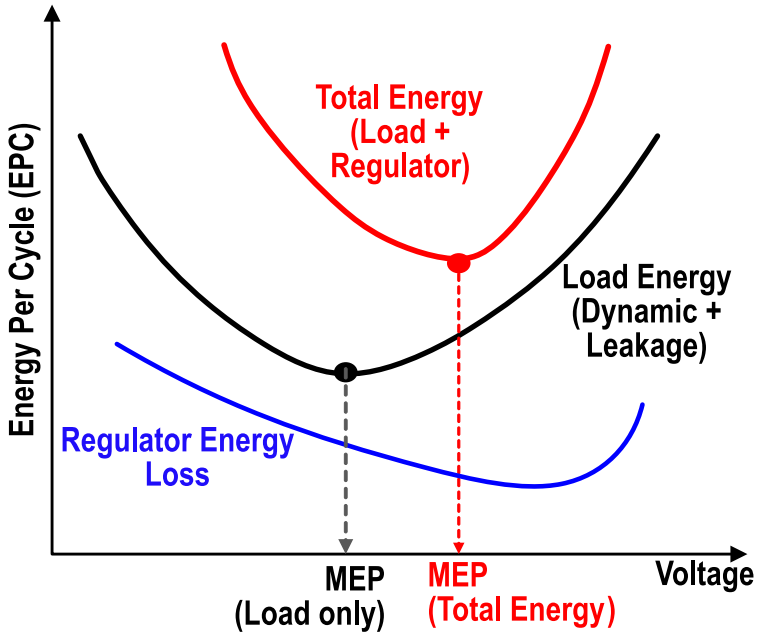


Figure 5.3: Non-uniform regulator efficiency can shift V_{MEP} by a significant amount

will directly impact the switching energy. For example, measurement results in section 5.5 will show that turning on a FFT accelerator can cause a significant variation in MEP. Also, device aging can cause change in the leakage energy and shift the MEP. The combined effect of these effects can cause a significant variation in V_{MEP} and f_{MEP} in run-time (Figure 5.2a). Therefore run-time tracking is essential to ensure system operation at MEP.

Implementing MEP-tracking system while maintaining performance requirements is a big challenge. Depending on the application, performance requirement can be different. Some system requires maintaining a target frequency with additional constraints for phase lag or lead, some system requires the system to run above the target frequency averaged over a period of time and some system requires communicating the frequency to the system on top of that. Ensuring performance regulation is therefore, essential for almost any real world application. Running the system in V_{MEP} and f_{MEP} can interfere with meeting target frequency. As shown in Figure 5.2b as V_{MEP} and f_{MEP} varies during run-time, f_{MEP} can go below the target frequency (f_{targ}) and running at MEP will prevent the system from maintaining the frequency requirement.

Another significant factor in energy minimization that is not taken into account in previous works is the regulator losses [124, 125]. Figure 5.3, shows a qualitative plot of the EPC vs operating voltage plot for a system. The EPC corresponding to the load energy of the system becomes minimum at a specific point denoted by MEP(Load only in figure 5.3. However, minimizing the EPC of load energy does not result in minimizing the EPC of the total system. The regulator losses constitute a significant portion of the total system energy. If the regulator losses are not uniform across voltage, they can cause a significant change in the curvature of the total EPC (tEPC) vs voltage plot. The MEP corresponding to the total system energy or tMEP can change by a significant amount. Minimizing the EPC corresponding to load energy only will result in failure to minimize the total EPC. For low-voltage operations and linear regulators (SC and Low-Dropout Regulator) the regulator efficiency degrades significantly at low voltage. These voltage dependent regulator efficiencies cause a significant change in the MEP and regulator losses must be taken into account.

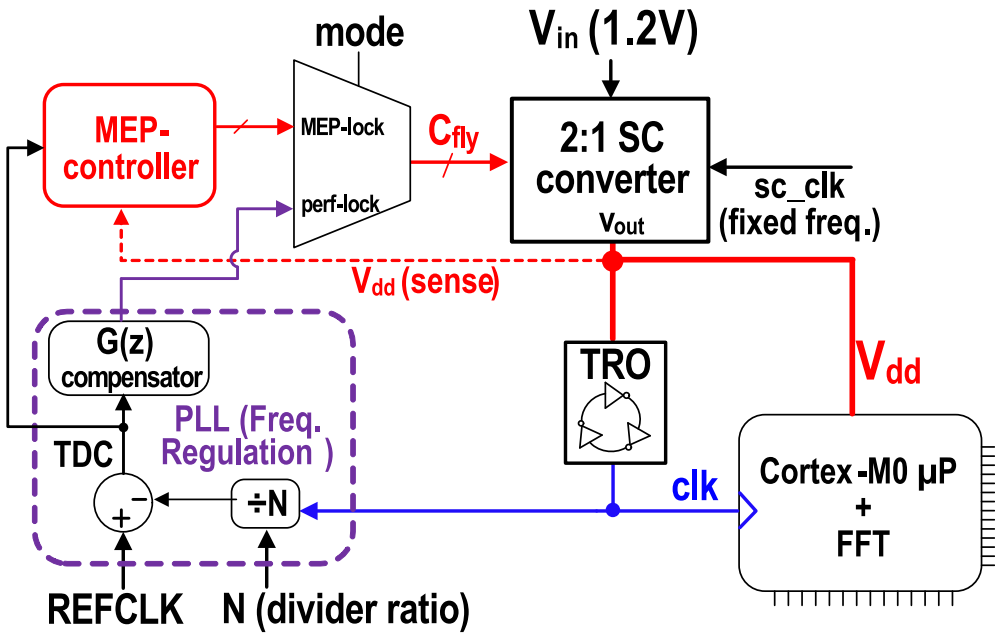


Figure 5.4: Proposed system architecture.

5.2 Comp-enabled tMEP tracking: Overview

We present a SC-regulator based system with on-chip tMEP tracking capability and performance guarantee. The proposed system ensures system operation at the most energy efficient point, while maintaining the performance requirement.

Maintaining the system performance requirements while tracking V_{MEP} is a challenge. Moreover, the large voltage margin in low-voltage SC-operation, causes substantial power loss in the guard-band. To implement a frequency regulated system with very little voltage margin, we adopt the UniCaP architecture [19, 127, 128].

Figure 5.4 shows the proposed system architecture. This architecture enables two modes of operation: (a) perf-lock, where the system is frequency locked, (b) MEP-lock, where the system operates at MEP. The appropriate mode is selected by the controller through a 2:1 MUX. A 2:1 SC-converter provides the system voltage V_{dd} . The system load consists of a Cortex-M0 microprocessor and FFT accelerator. The system clock(clk) is generated from a tunable replica oscillator(TRO).

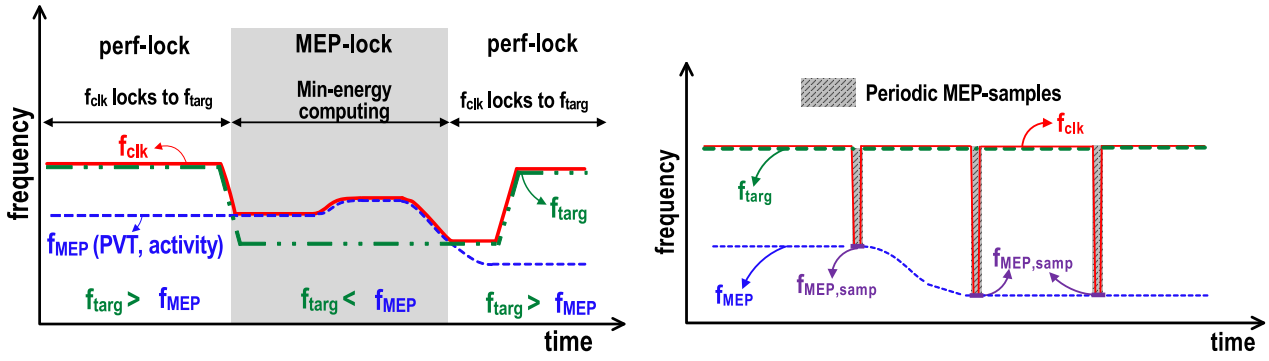


Figure 5.5: (a) Proposed system operation, (b) Periodic MEP sampling to keep track of change in f_{MEP} during performance-locked mode

TRO is constructed to match the time period and the voltage sensitivity of the critical path delay of the system. The TRO and the system load are powered by V_{dd}. Having the shared V_{dd}-powered TRO ensures timing slack in the presence of supply and temperature variation. During supply droop, as the critical path delay of the microprocessor increases, the TRO time period also increases by the same amount. This elastic clocking then helps maintain the timing slack. At near/sub-threshold operation in SC-based system, supply droop margin is very large. The elastic clocking provided by the TRO running on shared V_{dd} drastically reduces the margin.

To, regulate the frequency of *clk*, UniCaP constructs a phase-locked-loop (PLL). When the MUX in figure 5.4 selects perf-lock mode, the PLL regulates the *clk* frequency (f_{clk}) to N times the REFCLK frequency. The PLL is constructed with a frequency divider ($\div$ N), a time-to-digital converter(TDC) which digitizes the time difference between the REFCLK and the frequency-divided TRO *clk* and a compensator. The compensator output is used to modulate the f_{clk} by changing the V_{dd}. V_{dd} is tuned by capacitance modulation [114, 128], changing the number of flying capacitor banks (C_{fly}) in the SC-converter. The frequency of switching clock (sc_clk) of the SC is constant.

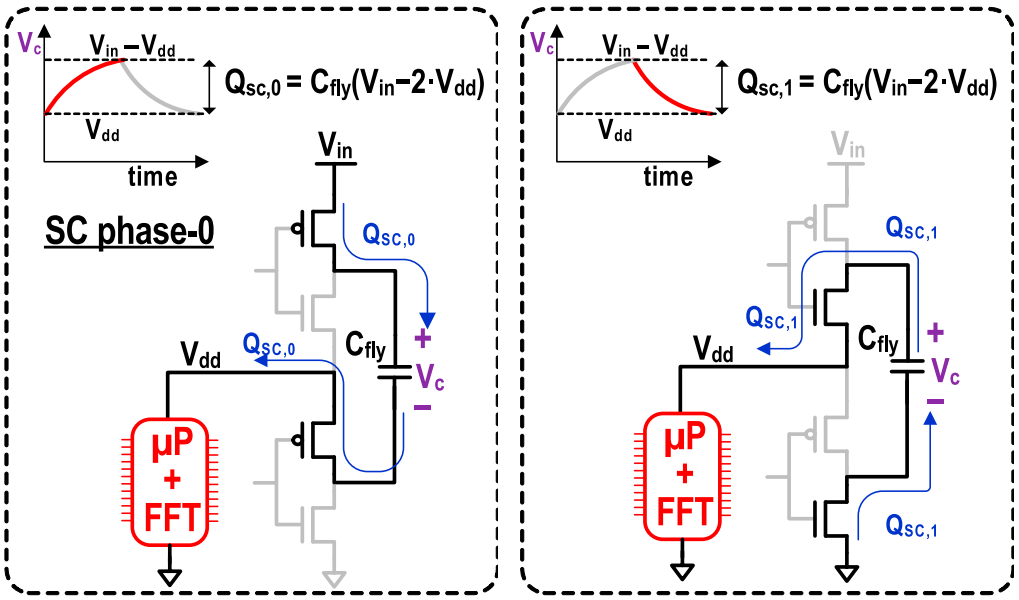
Changing f_{clk} by tuning V_{dd} prevents the system from timing failure because any change in V_{dd} will impact the TRO time period and critical path of the processor identically. The PLL ensures the target frequency ($N \cdot f_{clk}$) is maintained and the shared V_{dd} and elastic clocking enables drastic

reduction of voltage margin.

During MEP-lock mode, the selection MUX selects the output from the MEP controller which helps the system search and track V_{MEP} and f_{MEP}. The MEP controller make the system run at a particular V_{dd}; it then extracts the V_{dd} value (through V_{dd}-sensing) and other system parameters to quantify the total energy-per-cycle (EPC) (detail in next section). During MEP-lock, the MEP-controller runs the system at different V_{dd}s by changing C_{fly} and compares the EPC at those candidate V_{dd} point to find V_{MEP}. f_{MEP} is set by V_{MEP} through TRO in this mode. In this mode f_{MEP} is not regulated; therfore the compensator is not used. The TDC is re-purposed to measure the frequency f_{clk} and feed that information to MEP-controller for determining EPC (discussed in Section 5.3.2).

The combined operation of perf-locked and MEP-locked mode ensures total EPC minimization with performance constraints. Figure 5.5a shows the proposed system operation. To maintain frequency requirement, system frequency f_{clk} has to be higher than target frequency (f_{targ}). When f_{targ} is higher than f_{MEP} , system operates in per-locked mode where f_{clk} is set to f_{clk} by the frequency-regulated loop in UniCaP. If f_{targ} goes below f_{MEP}, the system autonomously transitions into MEP-locked mode. In MEP-locked mode, the system operates in f_{MEP} and system V_{dd} is set to V_{MEP} through the operation of MEP-controller. The MEP-controller also tracks any change in V_{MEP} and f_{MEP} in this mode. When f_{targ} exceeds f_{MEP} again, the system control is again reverted back to perf-lock mode. Therefore, the system always operates in the most energy-efficient point while maintaining the frequency requirement.

The system needs the value of f_{MEP} to compare with f_{targ} to decide the operating mode. In MEP-locked mode f_{MEP} is known. To keep track of changes in f_{MEP} in perf-locked mode, Periodic MEP-sampling is used, as shown in Figure 5.5b. From perf-lock mode, the system periodically goes to small duration of MEP-lock mode to sample the latest f_{MEP}. The re-purposed TDC in MEP-lock mode is used to obtain the f_{MEP}. The system performance is uninterrupted during the brief duration of MEP sample because of the elastic clocking feature. However, upon exiting from MEP-sampling, depending upon the frequency requirement, the system needs to run slightly higher than the required frequency to account for the lost performance.



(a) (b)

Figure 5.6: Charge transfer in (a) Phase 0 and (b) Phase 1 in a 2:1 switched-capacitor voltage converter

5.3 Computationally-enabled tMEP search and tracking

Our proposed tMEP searching and tracking methodology consists of two main parts : (a) computationally determining the tEPC, (b) Obtaining tMEP by successive comparison of tEPC at different operating points. In this section we discuss these part in detail

5.3.1 Derivation of total Energy-per-cycle(EPC)

For computationally determining the total EPC of system, we first come up with the expression of tEPC by analyzing the energetics of the switched-capacitor voltage converter (SCVC). Figure 5.6 shows the charge flow of a 2:1 SCVC. The SCVC works on supply voltage V_{in} to deliver I_L current to the load at output voltage V_{dd}. The SCVC has a flying capacitor (C_{fly}) and the switching frequency is f_{sc}. The SCVC works in 2-phases. In phase-0 (Figure 5.6a), the supply delivers charge to C_{fly}. In the

steady state operation in phase-0, the voltage across C_{fly} changes from V_{dd} to $V_{in} - V_{dd}$. The amount of charge delivered in the phase ($Q_{sc,0}$) can be calculated by multiplying the capacitance (C_{fly}) with the change in voltage across the capacitor:

$$Q_{sc,0} = C_{fly} \cdot (V_{in} - 2 \cdot V_{dd}). \tag{5.1}$$

In phase-1 (Figure 5.6b), the supply is disconnected and the voltage across C_{fly} discharges from $V_{in} - V_{dd}$ to V_{dd}. With this discharge, the capacitor delivers to the output charge $Q_{sc,1}$ which has the same value as $Q_{sc,0}$.

In a full-cycle of SC, the supply delivers charge only in phase-0. Therefore, the total charge delivered in a SC-cycle is $Q_{sc,0}$. Multiplying the charge with the supply voltage V_{in} will result in the total energy delivered in a SC-cycle, $E_{tot,sc}$,

$$E_{tot,sc} = V_{in} \cdot Q_{SC,0}. \tag{5.2}$$

Multiplying $E_{tot,sc}$ with the SC switching frequency (f_{sc}) will provide the total power delivered by the supply, P_{tot},

$$P_{tot} = E_{tot,sc} \cdot f_{sc}. \tag{5.3}$$

Dividing P_{tot} with the microprocessor frequency (f_{clk}) will result in total energy delivered per cycle (tEPC),

$$tEPC = \frac{P_{tot,}}{f_{clk}}. \tag{5.4}$$

Substituting the values from Equation 5.1, 5.2, 5.3 into Equation 5.4 will give us the expression of $tEPC$,

$$tEPC = \frac{V_{in} \cdot C_{fly} \cdot (V_{in} - 2 \cdot V_{dd}) \cdot f_{sc}}{f_{clk}}. \tag{5.5}$$

5.3.2 On-chip estimation of total EPC

Equation 5.5 provides us the expression for the total EPC (tEPC) of the system. To use the expression to estimate tEPC, all the parameters on the right hand side of the equation: V_{in}, C_{fly}, V_{dd}, f_{sc}, f_{clk}, needs to be known at run-time.

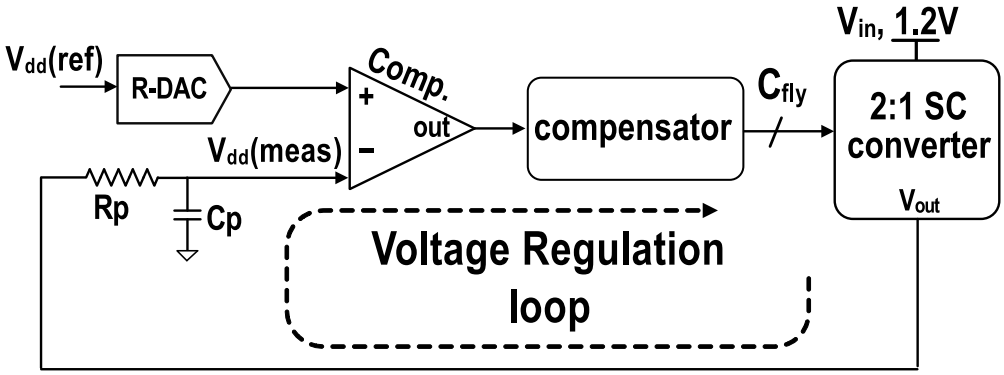
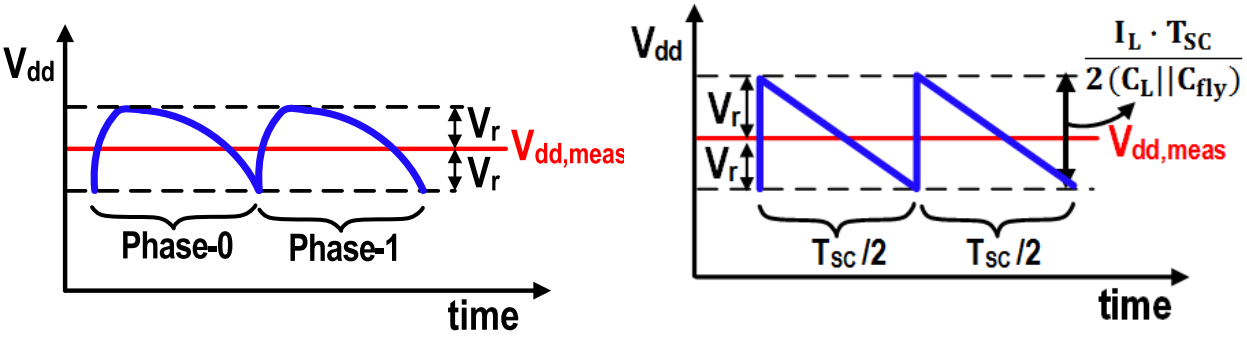


Figure 5.7: Voltage regulation loop for measuring V_{dd}

Because tMEP is found by comparing tEPC at different points, the constant term V_{in}can be ignored (for multiplication). Also, as mentioned in section 5.2, SC is tuned by capacitance-modulation. Therefore the switching frequency of SC, f_{sc} is also constant and can be ignored in computation.

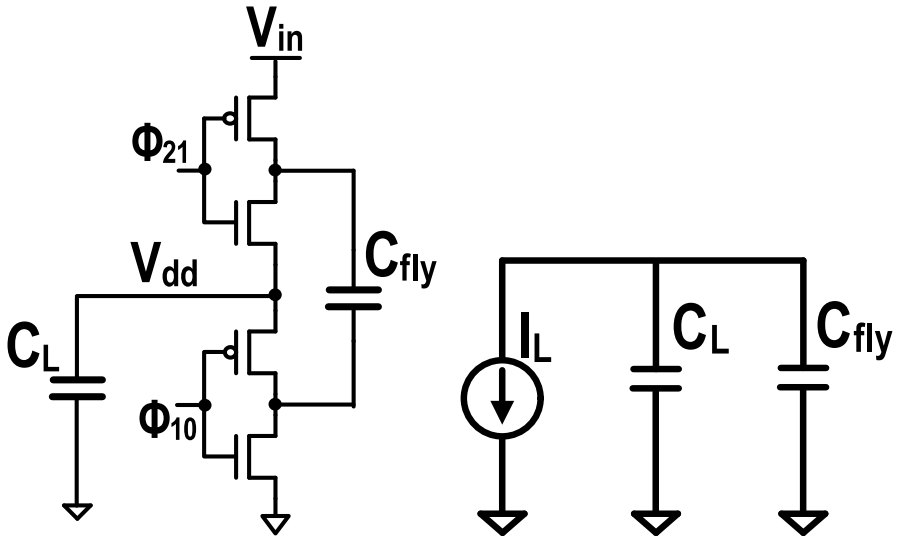
To measure the V_{dd}, the system relies on a voltage regulation loop shown in Figure 5.7. V_{dd} is set by the MEP-controller in terms of V_{ref} $code$ as a known parameter. A Resistive Digital-to-Analog Converter (RDAC) takes the V_{ref} $code$ and generates corresponding voltage V_{ref}. The V_{ref} is used as a reference by the voltage regulation loop to regulate the system V_{dd} to V_{ref}. The voltage regulation loop has a clocked comparator to compare the V_{dd} with V_{ref}. A RC filter (consisting of R_P and C_P) is used to remove the ripple from V_{dd}before comparison. The comparator is used by a compensator to change C_{fly}. Through the operation of the regulation loop V_{dd} approaches V_{ref}. Once, V_{dd} is equal to V_{ref}, the corresponding C_{fly} is obtained from the compensator output. The values of V_{ref} and C_{fly} are used as V_{dd} and C_{fly} to estimate tEPC.

The TDC and the frequency divider from the UniCaP architecture provides N_{clk}, the ratio of f_{clk} and the REFCLK frequency (f_{ref}). Because $f_{clk} = N_{clk} \cdot f_{ref}$ and f_{ref} is constant, N_{clk} can be used in the place of f_{clk} for tEPC estimation.



(a) (b)

Figure 5.8: (a) Ripple effect in EPC estimation, (b) simplified analysis of ripple effect



(a) (b)

Figure 5.9: (a) Charge redistribution C_{fly} and C_L causes ripple, (b) equivalent circuit model to calculate ripple voltage due to discharge

5.3.3 *Finer EPC estimation: Additional factors*

For accurate estimation of tEPC, some additional factors, not considered in Equation 5.5, should be taken into account.

First, Equation 5.5 ignores the losses in driving SC banks. The switching losses are proportional to the number of active banks and by construction (Section 5.4.2) proportional to C_{fly}. A one-time calibration can be used to obtain the proportionality constant, K_{sc}. After incorporating the switching losses, the expression for tEPC becomes:

$$tEPC = \frac{C_{fly} \cdot V_{in} \cdot \{(V_{in} - 2 \cdot V_{dd}) + K_{sc}\} \cdot f_{sc}}{f_{clk}}. \tag{5.6}$$

Secondly, the ripple effect is not taken into account in Equation 5.5. The voltage regulation loop measures the average value of V_{dd} ($V_{dd,meas}$ in Figure 5.8a) after removing the ripple through RC filtering. But tEPC estimation requires the minimum value of V_{dd} to calculate the discharge amount in C_{fly} (Section 5.3.1). Therefore, half of the ripple voltage, V_r needs to be subtracted from measured V_{dd} as shown in Figure 5.8a. With the correction of ripple factor, Equation 5.6 becomes:

$$tEPC = \frac{C_{fly} \cdot V_{in} \cdot \{(V_{in} - 2 \cdot V_{dd} + 2 \cdot V_r) + K_{sc}\} \cdot f_{sc}}{f_{clk}}. \tag{5.7}$$

Ripple effect arises due to charge redistribution in C_{fly} and the capacitance at the output node, C_L as shown in Figure 5.9a. In every phase of SC, V_{dd} first rises up due to charge redistribution of C_{fly} and C_L. Then V_{dd} decreases as the load current I_L causes discharge from C_{fly} and C_L (Figure 5.9b). To estimate V_R, we estimate instant increase of V_{dd} at the start of an SC-phase, followed by the linear discharge, as shown in Figure 5.8b. Charge loss in one SC-phase (for duration $T_{sc}/2$)is $I_L \cdot T_{sc}/2$. Dividing the charge loss by capacitance value will result in voltage decrease, $2 \cdot V_r$. Therefore,

$$2 \cdot V_r = \frac{I_L \cdot T_{sc}}{2(C_L || C_{fly})}. \tag{5.8}$$

I_L can be estimated from the equivalent circuit model [113, 129] as follows:

$$I_L = \frac{V_{in} - 2 \cdot V_{dd}}{2 \cdot R_{out}}. \tag{5.9}$$

Where, R_{out} is the output resistance of the SC. For slow-switching limit [113, 129], R_{out} can be estimated as,

$$R_{out} = \frac{1}{4 \cdot f_{sc} \cdot C_{fly}}. \tag{5.10}$$

Substituting values from Equation 5.9, 5.10 in Equation 5.8 we get,

$$2 \cdot V_r = \frac{(V_{in} - 2 \cdot V_{dd}) \cdot C_{fly}}{C_L + C_{fly}}. \tag{5.11}$$

Finally, substituting the value of V_r from Equation 5.11 in Equation 5.7 yields,

$$tEPC = \frac{C_{fly} \cdot V_{in}\{(V_{in} - 2V_{dd})(1 + \frac{C_{fly}}{C_{fly}+C_L}) + K_{sc}\} f_{sc}}{f_{clk}}. \tag{5.12}$$

C_L can be estimated by post-layout capacitor extraction. Equation 5.12 gives a more accurate estimation of tEPC and is used in for tMEP tracking.

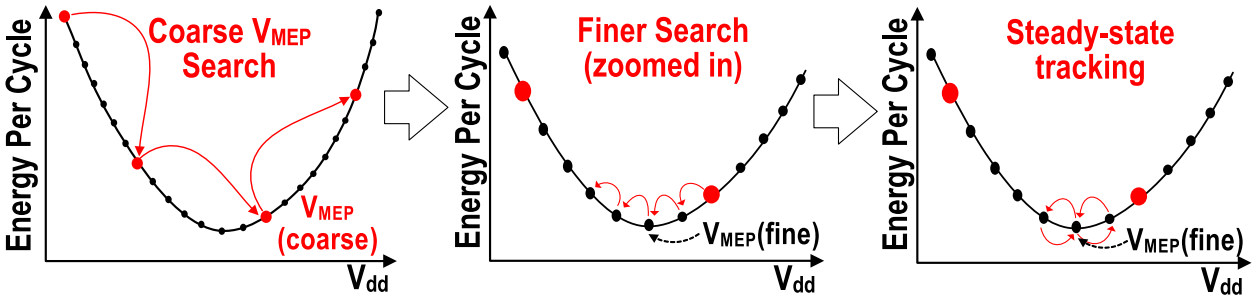


Figure 5.10: tMEP search methodology using successively finer steps

5.3.4 Finding MEP through EPC comparison

tMEP is found by comparison of tEPC at different V_{dd}s. For each V_{dd}, the MEP controller sets the appropriate V_{ref} *code*, the voltage regulation loop then locks V_{dd} to $V_{ref}$and the MEP controller then extracts the parameters and computationally estimates tEPC by Equation 5.12 (ignoring the multiplication with constant terms V_{in} and f_{sc}).

As shown Equation 5.12, tEPC computation involves a division with f_{clk}. Performing arithmetic division is expensive in terms of energy. Since, tMEP searching involves comparing two tEPC

values , instead of comparing tEPCs, we can compare the cross-multiplied terms and avoid division altogether.

The system reaches tMEP by successive comparison of tEPCs at different points. To expedite the searching process, our proposed methodology exploits convexity in tEPC search space to perform successive finer search. As shown in Figure 5.10, MEP-search is carried on in several phases. In first phase, the controller carries on a extensive search where the candidate V_{dd} points are coarsely spaced. Through successive comparison among the candidate V_{dd}s a minima of tEPC is found. Second phase searching starts off from the minima of the first search and goes to the opposite direction with a finer step and smaller search-window. The same pattern for subsequent phases and finally the tMEP controller ends up searching with the finest step within the smallest window to track any changes in V_{MEP}. Implementation details of the tMEP searching and tracking are provided in Section 5.4.3.

5.4 Implementation

The top level architecture of the proposed system was shown in Figure 5.4. A top level controller (built with synthesis,auto place and route or SAPR) selects whether the system would operating in MEP-lock or perf-lock mode. For performance-lock mode, the TDC, the frequency divider and the compensator are used. These components similar to those used in [19, 20]. The tunable replica oscillator TRO is also reused from [20] which is based on the work of [93]. For MEP-lock mode, MEP-controller is used.

Figure 5.11 shows different components that perform tMEP-searching and tracking. MEP-controller module orchestrates the process by setting the appropriate V_{dd}s and then measure and compare tEPC at those V_{dd}s. As mentioned in section 5.3, the voltage regulation loop consisting of RDAC, comparator, compensator sets the appropriate V_{dd} before tEPC estimation and comparison. The compensator either increases or decreases the switched-capacitor output voltage by tuning the flying capacitance C_{fly} until the comparator output changes to the appropriate value depending on the tMEP searching direction (implementation details in Section 5.4.3). To obtain fine grained control of the output V_{dd}, the precision of 5-bit control of the flying capacitors are augmented by

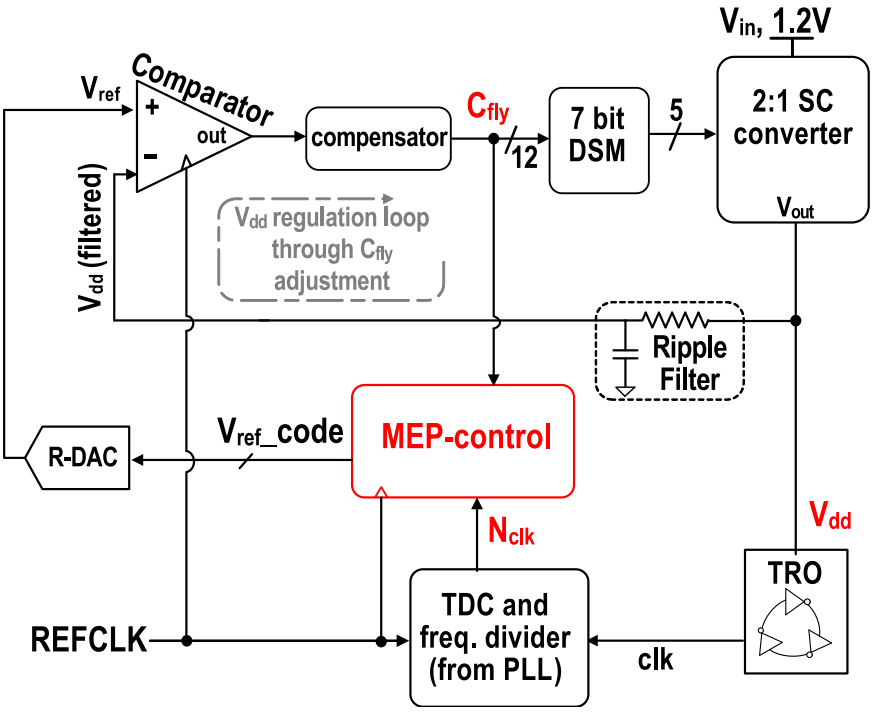


Figure 5.11: tMEP-searching and tracking circuits

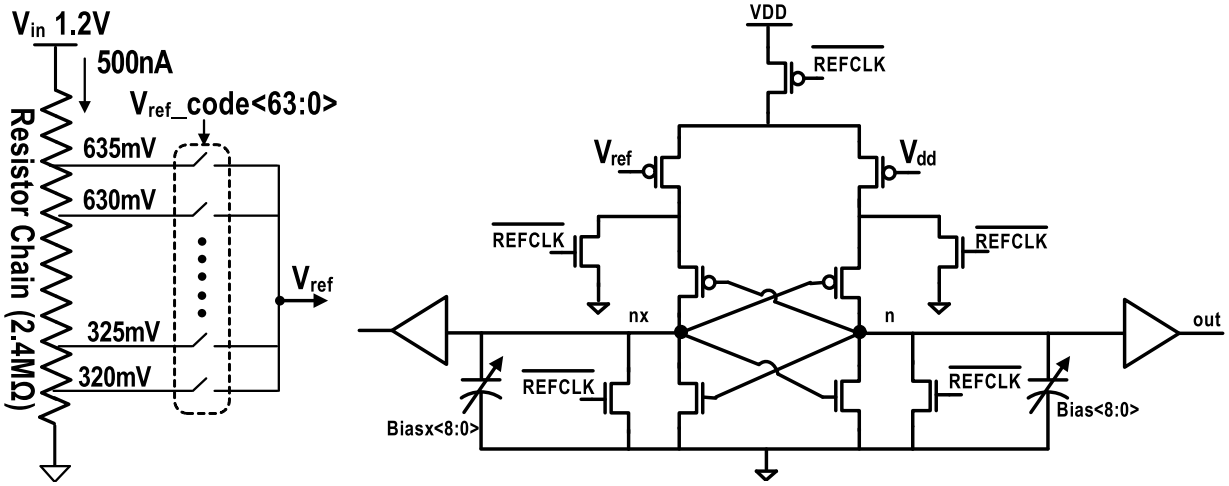


Figure 5.12: Circuits of (a) RDAC and (b) clocked comparator.

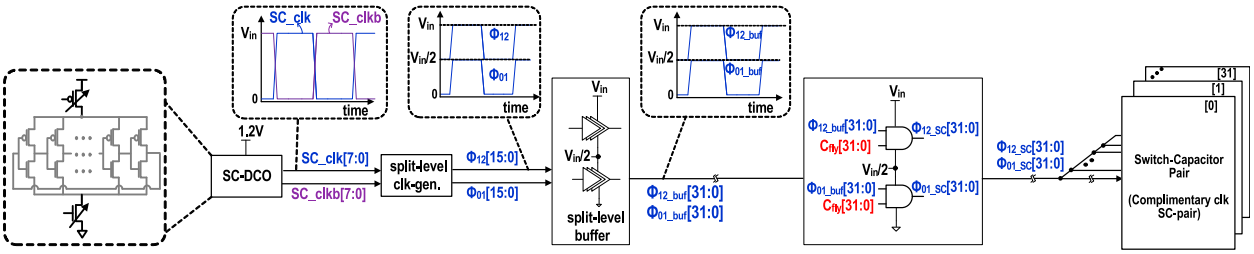


Figure 5.13: Driver for Switched-capacitor voltage converter

using a 7-bit delta-sigma modulator (DSM), running with the faster SC-clk. The compensator, MEP-controller and DSM are powered by the V_{dd}. The RDAC and comparator are powered by the 1.2V supply voltage. The ripple filter in the voltage regulation loop has a resistance of 26kohm (p+ poly resistor without Salicide) and capacitance of 30pF (nmos capacitor). After changing C_{fly}, the comparator waits one *REFCLK*cycle for the output of the RC filter to settle before measurement.

5.4.1 RDAC and comparator

The RDAC takes the V_{ref} *code* and sets the voltage reference for comparing with V_{dd}. Figure 5.12a shows the schematic of RDAC. RDAC consists of a resistor chain connected to 1.2V. P+ poly resistor without Salicide are used in the resistor chain. The combined value of resistors is 2.4MΩ. 64 tap points are selected at appropriate positions to provide voltages from 320mV to 635mV with 5mV step. A 64:1 MUX (implemented with transmission gates with 1-hot encoder) selects the V_{ref} corresponding to the V_{ref} *code*. The resistor chain dissipates 500nA current which is negligible compared to the minimum load power.

A clocked comparator or sense-amplifier is used to compare V_{dd} with V_{ref}. Figure 5.12b shows the schematic of the comparator. It is based on the sense-amplifier used in [15, 130]. The cross-coupled inverter pair arbitrates between the 2 input voltages at the rising edge of *REFCLK*. For offset cancellation in the comparator digitally tunable capacitors(implemented with nmos) are placed on the output nodes with 'bias' and 'biasx' digital input. The comparator output value is flopped by the

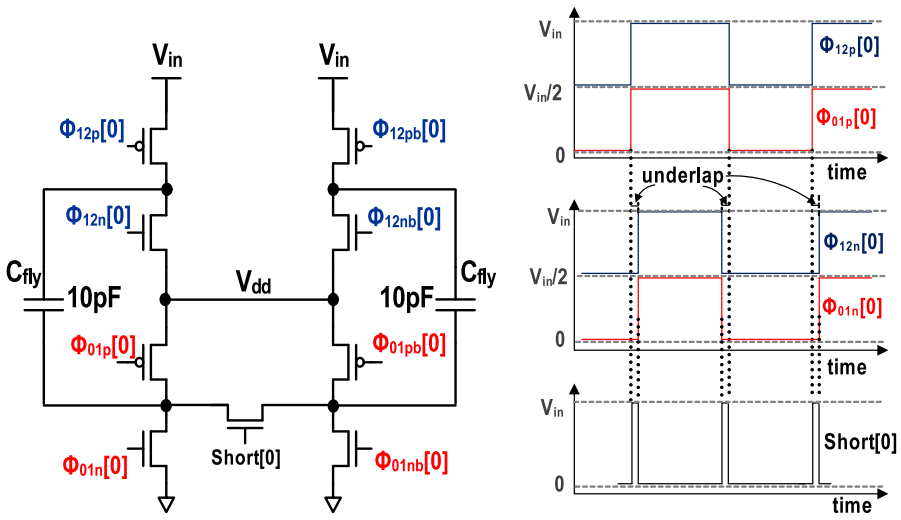


Figure 5.14: (a) Schematic of a SC bank pair. (b) Waveform of gate signals in SC bank

negative edge of *REFCLK*. This allows the sense-amplifier output half *REFCLK* period to settle. The sense-amplifier is actually clocked by a delayed version of *REFCLK* so that the flip-flop output is not setting to '0' due to the negative of *REFCLK* in sense-amplifier.

5.4.2 Switched-capacitor(SC) voltage converter

A 2:1 SC is used as the voltage converter. A current starved digitally controlled oscillator(SC-DCO) is used to drive the clock. Figure 5.13 shows the driver circuit for SC. The driver circuit is mostly similar to the one used in [128]. Floating split-rail mechanism [110, 128] is used to obtain the mid-rail. The key difference between this implementation and the SC-driver in [128] is capacitance-modulation is used to tune the output voltage here instead of frequency-modulation. Therefore, AND gates with 32-bit of enable signals for changing $C_{fly}$is added in the driver path. The resulting output signals, ϕ_{12_sc} and ϕ_{01_sc} are used to drive the SC-pair units. Figure 5.14a shows the schematic of a SC-pair unit. Bottom plate charge recycling (enabled by the short[0] signal in Figure 5.14a) is used to reduce the parasitic loss [109, 131].The signals at the gates are all generated

from ϕ_{12_sc} and ϕ_{01_sc}. The signals are appropriately timed (Figure 5.14b with under-lapping to prevent crowbar currents.

5.4.3 MEP controller

The MEP controller orchestrates the MEP-searching and tracking process during MEP-lock mode. The MEP controller is powered by the shared V_{dd} . The controller runs REFCLK on re-timed with sc_clk. For searching or tracking MEP includes (a)running the system at a known V_{dd} value, (b)measuring the EPC at that voltage and frequency, (c) comparing the EPC with the MEP at that point and update MEP if the EPC of the current operating point is lower.

Figure 5.15 shows the flowchart for MEP searching and tracking algorithm which is implemented inside the MEP controller using finite state machines(FSM). As mentioned in Section 5.3.4, MEP searching and tracking goes through 4 phases. The phases are denoted by n in Figure 5.15. Specific parameters needs to be defined for each phase. In a particular phase, MEP is searched in a set of voltages or candidate V_{dd}s. They are defined by V_{ref}_code and V_{window}. V_{ref}_code represents the distance between the candidate V_{dd}s in units of 5mV (smallest resolution in RDAC). V_{window} represents number of candidate V_{dd}s. There is one additional parameter m which represents the direction of search. m = 1 corresponds to a search with descending and m = -1 corresponds to ascending search.

For running the system at particular V_{dd} value, the controller increases or decreases C_{fly} in steps of $C_{step,mag}$. For descending search (m = -1), the controller keeps decreasing C_{fly} by $C_{step,mag}$ until the comparator result shows that $V_{dd} < V_{ref}$. The same goes for ascending search, with the polarities of variables and comparator result reversed. Once the V_{dd} is reached, EPC corresponding to the V_{dd} (V_{ref}), C_{fly} (Cap), f_{clk} (N_{clk}) is compared with the EPC corresponding to V_{MEP}, N_{MEP}, C_{MEP}, which are the V_{dd}, f_{clk} and C_{fly} for the hitherto known MEP points. If the current EPC is found to be lower then, V_{ref}, Cap, $N_{clk}$are loaded as the new values of V_{MEP}, f_{MEP}, C_{MEP}.

At the start of the searching process, V_{MEP}, f_{MEP}, $C_{MEP}$are set to values that will result in very high EPC so that they gets updated right after the comparing with the first operating point. For the initial phase, m = 1, V_{ref}_code, V_{window} and $C_{step,mag}$ are set as high values. As different phases

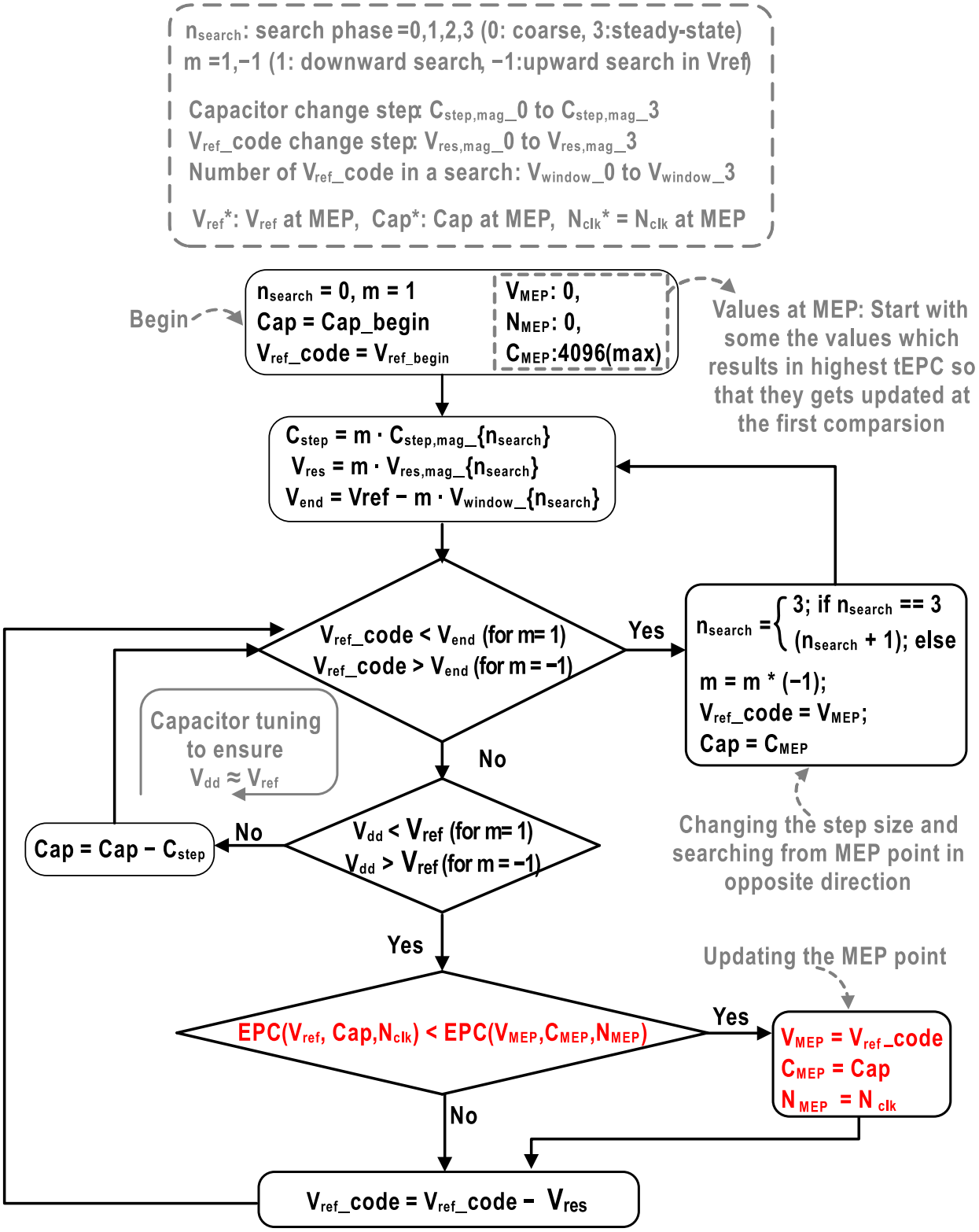


Figure 5.15: Flowchart of the MEP-searching and tracking algorithm

goes on, the polarity of m keeps reversing and smaller values of V_{ref_code}, V_{window} and $C_{step,mag}$ are used. For the steady-state track, denoted by n=3, V_{ref_code} is set to '1', V_{window} is set to '2' to track any changes in MEP.

5.4.4 System load

We used a ARM-cortex M0 processor and FFT accelerator as the system load. The cortex-M0 processor has 2 SRAM banks (for instruction memory and data memory) of 32b and 512 rows. The SRAM operates on a separate voltage rail set to 0.7V due minimum voltage requirement (V_{Min}) in SRAM. Level converters were used at the interface between SRAM and Cortex-M0 processor. The ARM-cortex M0 processor can be interfaced by Jlink adapter to run different programs. The Cortex M0 processor's critical path dictates the configuration of the TRO. ARM provided 'fmax' program was run on the processor for configuring the TRO setting.

The 32 bit FFT accelerator is the another part of system load. The FFT accelerator runs on the TRO clock (clk). To generate different loading conditions of FFT during testing, a clock gater (synthesized) is used to either run the FFT at f_{clk} or different fractions of f_{clk}. The different loading conditions are used to test the MEP-tracking at different switching factors of the total system load. FFT accelerator's correctness was checked throughout the measurement experiments.

5.5 Measurements

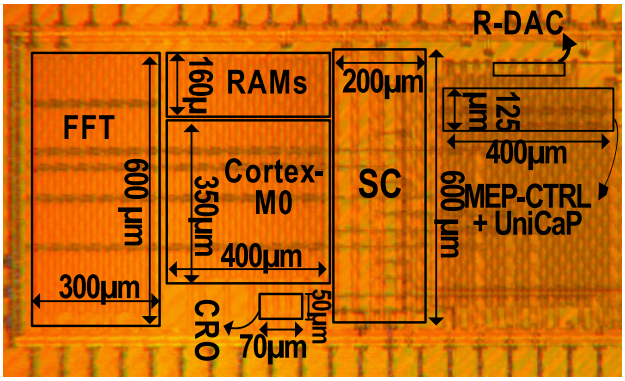


Figure 5.16: Die-photo of the test-chip

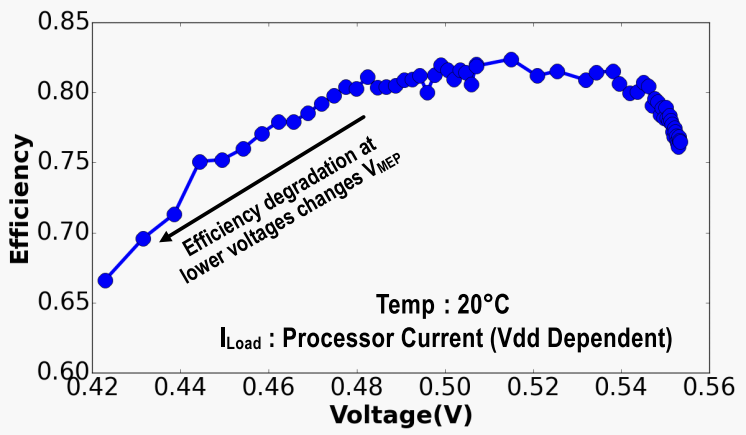


Figure 5.17: Measured switched-capacitor efficiency vs V_{dd} curve

Figure 5.16 demonstrates the 65nm LP CMOS test-chip. The SC converter runs with a 1.2V supply voltage. The system operates at V_{dd} of 0.38-0.58V, provided by the SC converter. Within the V_{dd} range, the system frequency, f_{clk} is 1.1-38MHz. The REFCLK frequency is fixed at 96kHz. Integrated built-in self test (BIST) was used to record snapshots of internal state variables in successive clock cycles during transient events and to simplify calibration of the TRO. Importantly, no explicit decap modules were introduced or required, either on- or off-chip in the design of this prototype. The quantity of implicit decap (including the capacitance from FFT) is approximately 250 pF. The MEP controller has an area overhead of 0.043mm^2. For the measurements, the Cortex-M0 processor and FFT accelerator functionality was verified during the corresponding experiments. For the demonstration of combined system operation (section 5.5.5 Altera FPGA was used to send the command word for DVFS transition and MEP sampling.

5.5.1 SC-converter efficiency

Figure 5.17 shows the measured SC-converter efficiency across different V_{dd}s. As shown in the figure, at lower V_{dd}s, the efficiency degrades significantly. The non-uniform SC-efficiency causes significant shift in the tMEP. Measurement results show that with the cortex-M0 processor as load at 20°C, the tMEP is at 542mV, whereas the MEP from load EPC only (without considering regulator

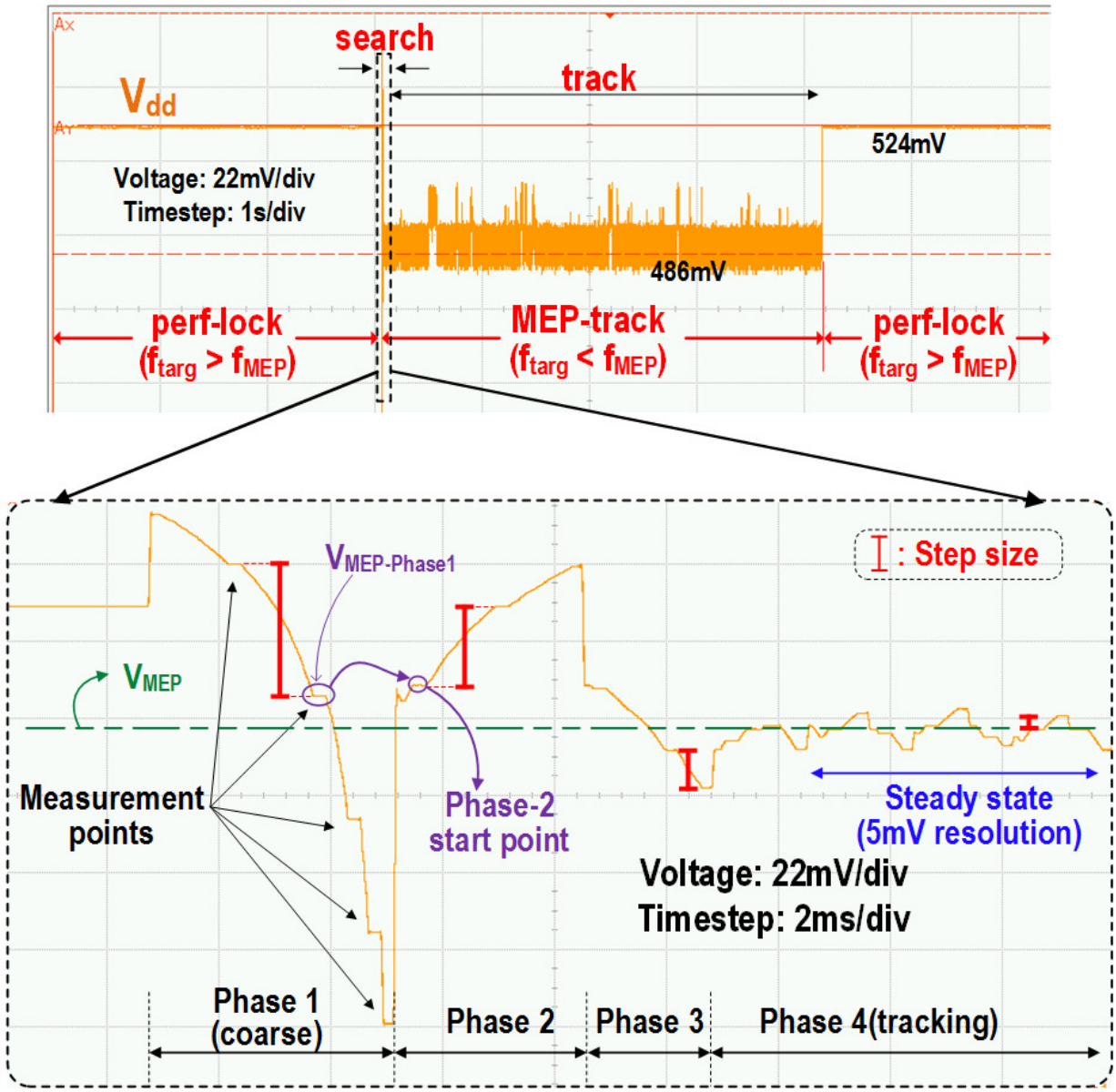


Figure 5.18: Measured V_{dd} waveforms during transitions perf-lock and MEP-lock modes. (Inset) Phases of MEP search, and corresponding V_{dd} transitions demonstrating proposed MEP-searching algorithm in action

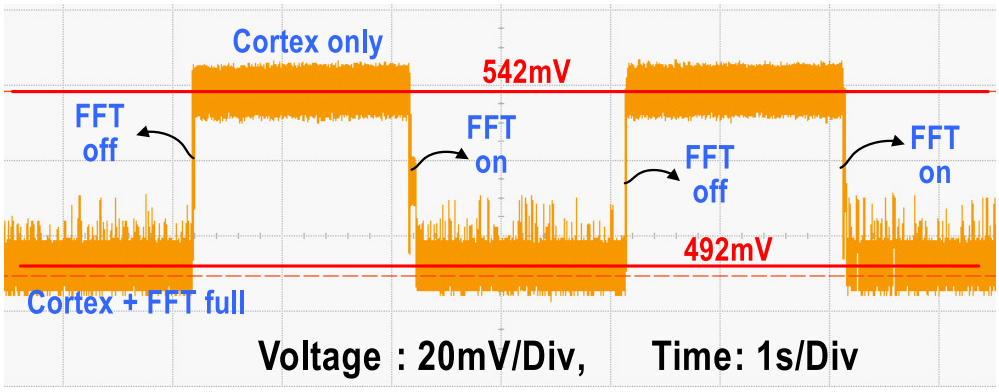


Figure 5.19: Measured V_{dd} waveform under MEP-lock during run-time changes in switching activity by FFT on-off operation

losses) is at 460mV. Therefore, considering SC-converter losses is essential for accurate MEP tracking. As will be shown in 5.5.4, our proposed MEP tracking method takes into account regulator losses and accurately tracks the tMEP.

5.5.2 tMEP-searching and tracking

Figure 5.18 presents the oscilloscope capture of V_{dd} traces demonstrating the proposed system operation in perf-lock and MEP-lock mode. When $f_{targ} > f_{MEP}$, the system operates in perf-lock mode and V_{dd} is set by the PLL in UniCaP to ensure $f_{clk} = f_{targ}$. When f_{targ} goes below f_{MEP}, system autonomously transitions into MEP-lock mode. At first the system performs MEP-search. When tMEP is found, system carries on MEP-tracking to adjust to any changes in tMEP. A closer look into the MEP-search part, shown in the inset in figure 5.18, demonstrates the MEP-searching algorithm (Section 5.3.4) in action. The traces of V_{dd}during MEP-search shows different searching phases. In phase-1, a coarse search is carried on with coarsely spaced measurement points. For each measurement, the measurement voltage is set by RDAC as a reference; the regulator then changes the voltage until V_{dd} reaches the reference. Once V_{dd} reaches the reference, the tEPC is estimated and comparison is done (during the small valleys in the V_{dd} traces in the inset). Phase-2 starts from the minimum tEPC point among the measured point in phase-1 and goes in the opposite direction

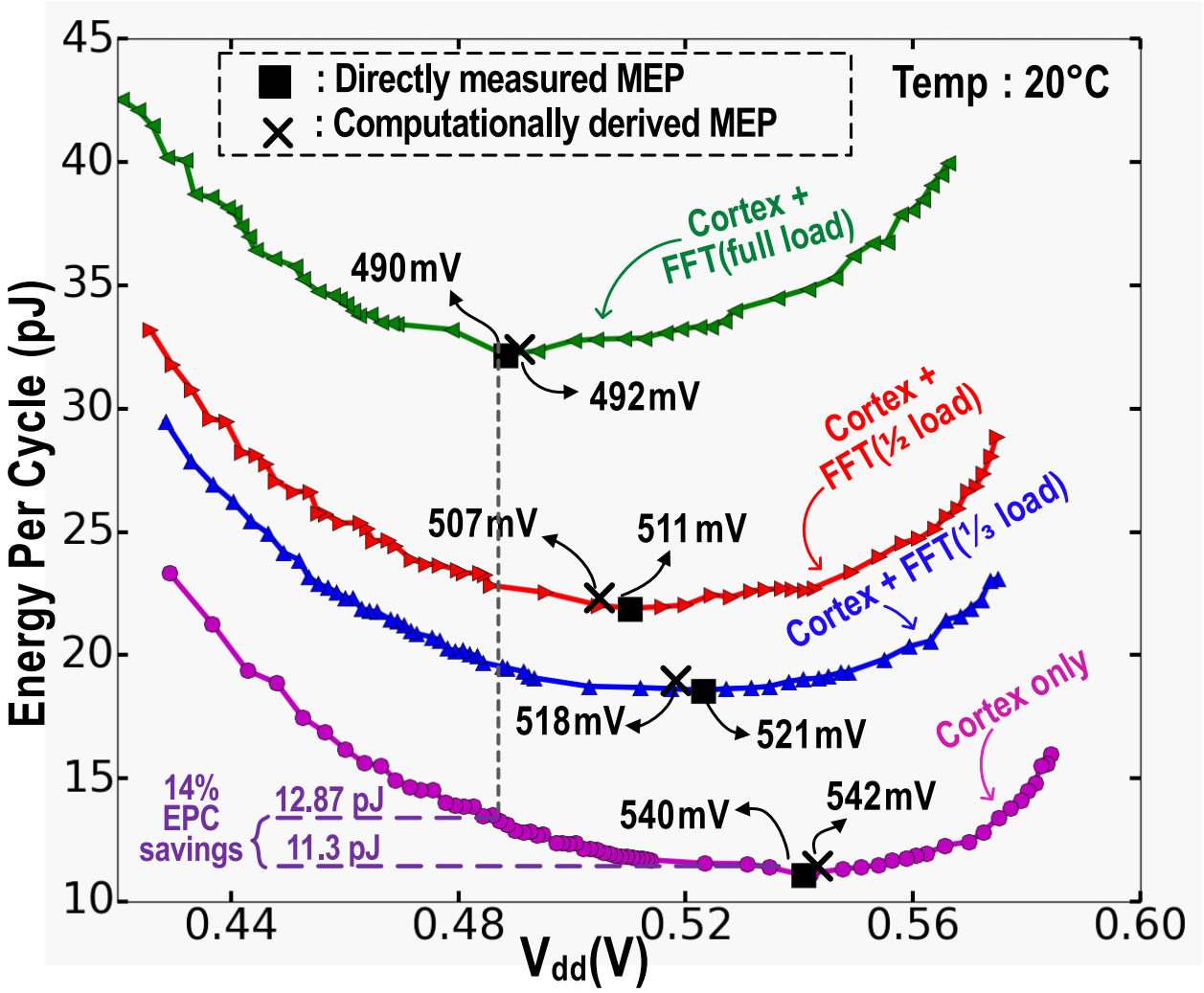
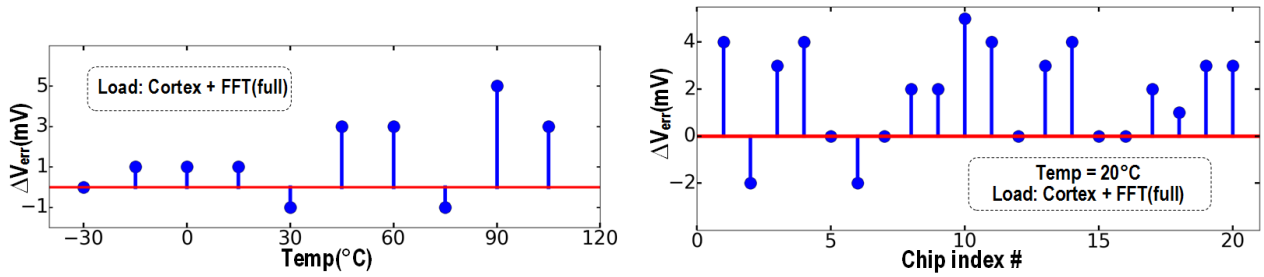


Figure 5.20: EPC vs V_{dd} sweep at different loading condition. Computationally derived V_{MEP} shown in the plots



(a) (b)

Figure 5.21: (a) Measured V_{MEP} error across different temperature, (b)Measured V_{MEP} error across different chips

with a finer step. Phase-3, starts with an even smaller step. Finally, in steady state, V_{dd} keeps changing by the smallest step(5mv) around the V_{MEP} to adjust to any run-time changes.

5.5.3 Run-time tracking of switching activity variation

Figure 5.19 shows oscilloscope traces of V_{dd} during MEP-tracking in the presence of switching activity variation. Switching activity variation is demonstrated by turning on/off the FFT accelerator in the load. In the experiment, at first the FFT accelerator is running and the tMEP is tracked at 492mV. When FFT is turned off, the system has less switching EPC and the tMEP shift to a higher voltage of 542mV. Thanks to the MEP-tracking feature, the system is able to adjust V_{dd} to the new V_{MEP}, as shown in the figure. Throughout the experiment, FFT was turned on and off and the system was always able to adjust to the new tMEP.

5.5.4 MEP tracking accuracy

To show the accuracy in determining the tMEP of the proposed system across different load, we present figure 5.20. The figure plots the measured tEPC of the system as the supply V_{dd} is swept. tEPC is measured externally through ammeter reading at the supply. The measurement was taken for the proposed system running under different loading condition, implemented by differnt clock

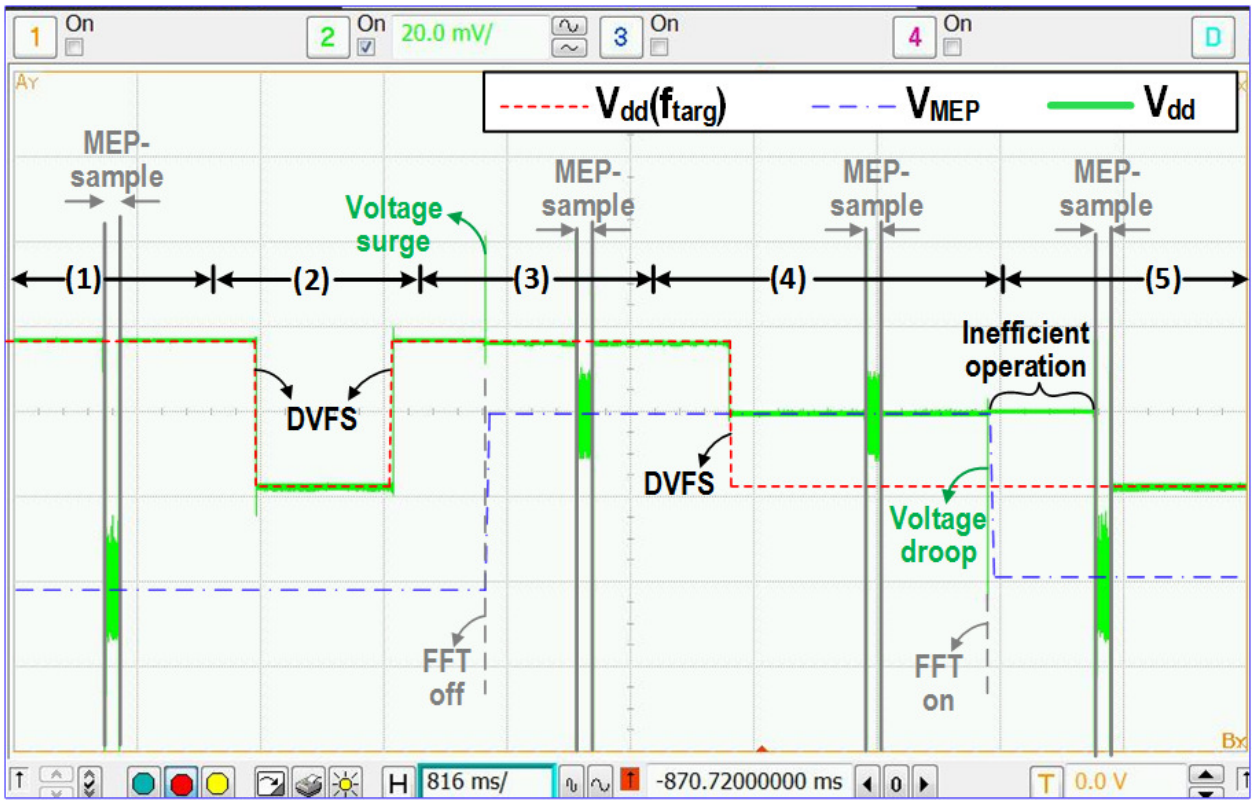


Figure 5.22: Measured V_{dd} waveform under MEP-lock during run-time changes in switching activity by FFT on-off operation

gating duty cycle of the FFT. For each load, the minima of the tEPC vs V_{dd} curve is the actual tMEP. For each load, tMEP obtained from our proposed MEP tracking methodology is also shown (with the symbol $\times$). Form figure 5.19 we see that V_{MEP} derived from on-chip tracking is always within 5mV of the actual V_{MEP}. Moreover, turning on-off FFT causes a shift in the V_{MEP} and by tracking the change accurately our proposed algorithm provides 14% energy savings. Figure 5.21a shows the measured V_{MEP} error across temperature. Across the temperature range from -30°C to 105°C, the error is always within 5mV. We also measured V_{MEP} error across 20 different chips and the result is shown in figure 5.21b across different chip indexes. Across different chips, the V_{MEP} error is also within 5mV.

5.5.5 *Demonstration of combined system operation*

We conduct an experiment to show the combined operation of the performance-constrained MEP-tracking system running under one possible operating scenario. The 65nm CMOS test-chip can operate in perf-lock and MEP-lock mode fully autonomously, a top level controller is needed to send out the corresponding command word (turn on/off FFT, Perform DVFS, MEP sampling). To run this experiment, we used Altera FPGA to send different command words to demonstrate different aspects of a performance-regulated MEP tracking system.

Figure 5.22 shows the oscilloscope traces of V_{dd} during the experiment. For explanation purpose, traces corresponding to V_{MEP} and V_{dd} (f_{targ}) is drawn. V_{dd} (f_{targ}) corresponds to the V_{dd} needed to run the system at target frequency (f_{targ}), which is set by UniCaP operation. Also different regions in the oscilloscope capture in annoted ((1) to (5)) for description purpose. At the start of the experiment Cortex-M0 processor and FFT both are running. The system is operating in per-lock mode with periodic MEP sample to measure the unknown f_{MEP} (region (1)). As shown in the oscilloscope trace, V_{dd} settles to V_{MEP} during MEP sampling. The system also undergoes through two DVFS (region (2)), executed by changing the divider ratio N in the PLL in UniCaP. As shown in the figure V_{dd} is changed to the new values of V_{dd} (f_{targ}). Thanks to the elastic clocking feature of UniCaP, the system was able to run un-interrupted and flawlessly during DVFS. After that, the FFT accelerator is turned off (region (3)). This result in voltage surge due to sudden decrease in load current. The voltage surge is recovered by UniCaP to restore the frequency. Turning off FFT all causes a shift in V_{MEP} and the new f_{MEP} is sampled in the next MEP-sampling (region (3)). Then the system undergoes another DVFS, this time the target frequency is less than f_{MEP} (region (4)). Having the value of f_{MEP} sampled, the system finds out that f_{targ} is lower than f_{MEP}. In one variant, as described in section 5.2, the system can choose to operate in MEP-lock mode. Here we construct a system where the operating frequency is regulated and known to the top-level. The system operates in performance-lock mode with f_{MEP} as the target frequency instead of f_{targ}. System undergoes through another MEP-sampling. After that, towards the end of region (4), FFT is turned-on again. This results in voltage droop which is recovered by the performance regulated loop. The elastic

clocking feature prevents any timing failure during the V_{dd} droop. Turning on FFT also results in V_{MEP} changing to a lower value. f_{MEP} goes below f_{targ} again. The system is unaware of the change because it hasn't sampled the latest f_{MEP} and keeps operating in perf-lock mode with the previously sampled f_{MEP} as the frequency. This temporary inefficient operation (region (5)) ends when the system goes through another MEP-sampling and f_{MEP} is updated. With the updated f_{MEP} value the system finds out that f_{MEP} is lower than f_{targ} and the system continues to operate with f_{targ} as the target frequency.

5.6 Discussion

In this section, we discuss some important observations, features and existing limitations of the proposed methodology and test chip. The derivation of tEPC assumes the operation of slow-switching region in the switched-capacitor. The equation of tEPC will be inaccurate for switched-capacitors operating in fast-switching region. However, for ultra-low power system, around the region of V_{MEP}, switched-capacitor usually operates in slow-switching limit. So, slow-switching limit assumption is valid for tMEP tracking.

In our implementation, we used capacitance modulation for tuning SC. Frequency modulation, which is also a common technique in SC converters [104, 115, 128], can also be used in the SC for performance constrained MEP-tracking. In that case f_{sc} will not be constant and run-time measurement of f_{sc} is needed to estimate tEPC.

The voltage regulation loop in our implementation tunes V_{dd} with small step. This results in longer search-time of tMEP and a sudden voltage droop or surge can cause even longer time. A voltage regulation with faster response time [104] can result in faster-search time.

MEP-sampling allows the system to be aware of the latest f_{MEP}. A frequent MEP-sampling will provide more up-to-date knowledge of f_{MEP}, but the system will lose performance during these more frequent episodes f_{MEP} and needs to run at higher frequencies more often to compensate the lost performance. On the other hand, less frequent MEP-sampling will result in delayed update in $f_{MEP}$and the system will end up operating in inefficient region for longer duration. The appropriate choice of MEP-sampling frequency depends on the application and variation of temperature and

loading of the system.

With the proposed method, MEP-tracking bandwidth depends on the window of tEPC estimation, which is time period of the reference clock in the current implementation. In a system where frequency of load-change is higher than the MEP-tracking bandwidth, the system will end up operating in the average V_{MEP} over the estimation window.

Finally, like previous work in MEP-tracking [125, 132], the scope of this work was on implementation and demonstration of MEP-tracking. To minimize the total energy of the system, the system needs to operate with power gating [121] and operation at MEP will then result in minimum total energy dissipation.

5.7 Conclusion

We presented a fully-integrated, all-digital switched-capacitor regulator based ultra-low-power system that autonomously operates in the MEP while maintaining the performance-constraint. Ensuring operation at MEP is the pathway to minimizing the total energy of the system and maintaining the performance constraints makes the system viable for real world applications. The system was designed on UniCaP architecture which reduces the voltage margin and substantially reduce the energy dissipation. Our proposed approach also takes into account the regulator energy losses and minimizes the total energy-per-cycle. Post-silicon demonstration from the 65nm CMOS test-chip shows MEP tracking with less than 5mV error in V_{MEP}, while maintaining the performance constraints. Measurement result also shows tracking MEP across temperature variation of -30°C-120°C and load variation with less than 5mV error.

Chapter 6

CONCLUSION

6.1 Summary of Research Contributions

The dissertation is focused on improving the speed of microprocessors while minimizing the power dissipation. We investigated the barriers to achieving high power-performance metric and tried to overcome those barriers by novel circuits and systems.

Our solutions targeted several areas of SoC architecture. Interestingly, in all of these problems, the key breakthrough comes from digital control system techniques. The reason is, in many cases the key challenge is to build a system that is robust across PVT variation. Constructing a control system by extracting the suitable parameters ensured adaptability to PVT variations. For the Quasi-resonant clocking we leveraged the orthogonal relationship in voltage and current waveform in LC circuit and constructed a delay-locked loop based system that ensures proper timing for gating. In computational PLL, the existing model did not account for loop delay. We found out a way to measure the loop delay during run-time and incorporate the effect in the control loop. Also to account for the PVT variation, we implemented adaptive gain adjustment. For UniCaP architecture, the key idea was to absorb to uncoupled control loops into one. The connecting link was the tunable replica oscillator. Constructing a performance regulated loop with voltage regulation absorbed inside the loop truly enabled drastic voltage margin reduction while maintaining performance regulation. And finally in MEP tracking, we constructed a control system that will take system operating towards MEP through successive comparisons.

The types of digital control system techniques that were applied in the projects ranges from Simple Bang-Bang Control system technique (in QRC and MEP-tracking loop) to advanced computational control (Computational PLLs).

All of the proposed techniques are demonstrated in 65nm CMOS test-chips. The measurement

results shows significant improvement compared to state-of-the art solution. QRC demonstrated for the first time, DVFS(while maintaining the duty cycle) with DVFS measurements in the 0.7V-1.2V range with Energy-per-Cycle reduction of 32%–47%. Computational locking demonstrated lock-time of sub-16 T_{REFCLK}. In UniCaP-SC, measurement results demonstrated 16% reduction in $V_{dd}$by enabling 94% reduction of the voltage margin. And finally for efficient ultra-low-power operation our test-chip demonstrates MEP-tracking with less than 5mV error while maintaining the performance requirements.

While we have demonstrated significant improvement in power-performance metric of processors, they cannot be applied simultaneously to all classes of microprocessors. Quasi-resonant clocking is not suitable for a system with a large clock distribution network because in that case the resistive losses will impact the resonant energy savings. Computationally-locked PLL significantly improves the power-gating, cache-coherency and DVFS latency and is most suitable for high-speed multi-core processors. UniCaP is a generalized solution to improving the voltage margin of microprocessors and can be applied to all digital systems that need clock and power regulation. UniCaP-SC, the SC-based system that we implemented, is particularly suitable for ultra-low power systems for limited current density of SC converters. Similarly the self-autonomous performance-constrained MEP tracking system is suitable for low power operation. For extremely low power operation($<< 1\mu$W) the MEP controller power may dominate and in that case using MEP-controller module will not be a feasible solution.

6.2 Future Research Direction

In addition to improving the power-performance metric of SoCs, several of our proposed techniques also opened up new research directions that are outside the scope of this work.

Our proposed computational locking approach is a general solution. We demonstrated computational locking for system clocking. But computational locking approach can also be used for other PLLs. For example, PLLs for SerDes application, BLE, spectrum sensing and LPDDR (DRAM interfacing). For burst-mode communication, PLL lock-time usually constitutes a significant portion of the on-time [68]. Reducing the PLL lock-time can save substantial power by reducing the

on-time.

Also, the key insights of computational locking can be applied in voltage regulator or other control system applications. Researches have been going on to implement fast-locking low-dropout-regulator (LDO) that can benefit from the techniques used in fast-locking PLL.

UniCaP architecture has been previously implemented in LDOs [133] and buck converters [91]. For multi-core architecture application, UniCaP can be very effective in tolerating the high supply droop in single-inductor multiple output (SIMO) buck converter.

We demonstrated MEP-tracking in switched-converter based regulator. The computational approach can also applied for MEP-tracking in buck converters and LDOs. The expression for EPC will be different and require extraction of different parameters. Implementation of performance-constrained MEP tracking in buck converter and LDO based system is subject to further researches.

BIBLIOGRAPHY

[1] S.-Y. Wu, J. Liaw, and Others, "A highly manufacturable 28nm CMOS low power platform technology with fully functional 64Mb SRAM using dual/tripe gate oxide process," in *2009 Symposium on VLSI Technology*. IEEE, june 2009.

[2] S.-Y. Wu, C. Y. Lin *et al.*, "A 16nm FinFET CMOS technology for mobile SoC and computing applications," in *2013 IEEE International Electron Devices Meeting*. IEEE, dec 2013, pp. 9.1.1–9.1.4.

[3] S. U. Z. Khan, M. S. Hossain, F. U. Rahman, R. Zaman, M. O. Hossen, and Q. D. M. Khosru, "Impact of high-κ gate dielectric and other physical parameters on the electrostatics and threshold voltage of long channel gate-all-around nanowire transistor," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 28, no. 4, pp. 389–403, jul 2015.

[4] S. U. Z. Khan, M. S. Hossain, M. O. Hossen, F. U. Rahman, R. Zaman, and Q. D. M. Khosru, "Analytical modeling of gate capacitance and drain current of gate-all-around $In_xGa_{1-x}As$ nanowire MOSFET," in *2014 2nd International Conference on Electronic Design (ICED)*. IEEE, aug 2014, pp. 89–93.

[5] S. U. Z. Khan, M. S. Hossain, F. U. Rahman, R. Zaman, M. O. Hossen, and Q. D. M. Khosru, "Uncoupled mode space approach towards transport modeling of Gate-All-Around $In_xGa_{1-x}As$ nanowire MOSFET," in *8th International Conference on Electrical and Computer Engineering*. IEEE, dec 2014, pp. 100–103.

[6] Q. D. M. Khosru, S. U. Z. Khan, M. S. Hossain, F. U. Rahman, M. O. Hossen, and R. Zaman, "Capacitance-Voltage Characteristics of Gate-All-Around InxGa1-XAs Nanowire Transistor," *ECS Transactions*, vol. 53, no. 1, pp. 169–176, may 2013.

[7] F. U. Rahman, M. S. Hossain, S. U. Z. Khan, R. Zaman, M. O. Hossen, and Q. D. M. Khosru, "Characterization of interface trap density of In-rich InGaAs Gate-all-around nanowire MOSFETs," in *2012 7th International Conference on Electrical and Computer Engineering*. IEEE, dec 2012, pp. 674–677.

[8] R. Zaman, S. U. Z. Khan, M. S. Hossain, F. U. Rahman, M. O. Hossen, and Q. D. M. Khosru, "Self-consistent determination of threshold voltage of In-rich Gate-All-Around InxGa1xAs nanowire transistor incorporating quantum mechanical effect," in *2012 7th International Conference on Electrical and Computer Engineering*. IEEE, dec 2012, pp. 678–681.

[9] Md. Shafayat Hossain, Saeed Uz Zaman Khan, Md. Obaidul Hossen, Fahim Ur Rahman, R. Zaman, and Q. D. M. Khosru, "Analytical modeling of potential profile and threshold voltage for rectangular gate-all-around IIIV nanowire MOSFETs with ATLAS verification," in *2012 IEEE International Conference on Electron Devices and Solid State Circuit (EDSSC)*. IEEE, dec 2012, pp. 1–3.

[10] Md. Obaidul Hossen, Md. Shafayat Hossain, Saeed Uz Zaman Khan, Fahim Ur Rahman, R. Zaman, and Q. D. M. Khosru, "Ballistic performance limit and gate leakage modeling of Rectangular Gate-all-around InGaAs Nanowire Transistors with ALD Al2O3 as Gate Dielectric," in *2012 IEEE International Conference on Electron Devices and Solid State Circuit (EDSSC)*. IEEE, dec 2012, pp. 1–3.

[11] A. L. S. Loke, D. Yang, T. T. Wee, J. L. Holland, P. Isakanian, K. Rim, S. Yang, J. S. Schneider, G. Nallapati, S. Dundigal, H. Lakdawala, B. Amelifard, C. Lee, B. McGovern, P. S. Holdaway, X. Kong, and B. M. Leary, "Analog/mixed-signal design challenges in 7-nm CMOS and beyond," in *2018 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, apr 2018, pp. 1–8.

[12] R. Dennard, F. Gaensslen, V. Rideout, E. Bassous, and A. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, oct 1974.

[13] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, power, and the future of CMOS," in *IEEE International Electron Devices Meeting, 2005. IEDM Technical Digest.* IEEE, 2005, pp. 9–15.

[14] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded Sparc Processor," *IEEE Micro*, vol. 25, no. 2, pp. 21–29, mar 2005.

[15] F. U. Rahman and V. S. Sathe, "Voltage-scalable Frequency-independent Quasi-resonant Clocking Implementation of a 0.7-to-1.2V DVFS System," in *2016 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, Feb 2016, pp. 334–335.

[16] F. ur Rahman and V. Sathe, "Quasi-Resonant Clocking: Continuous Voltage-Frequency Scalable Resonant Clocking System for Dynamic Voltage-Frequency Scaling Systems," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 3, pp. 924–935, mar 2018.

[17] F. ur Rahman, G. F. Taylor, and V. S. Sathe, "Computational locking: Accelerating lock-times in all-digital PLLs," in *2017 Symposium on VLSI Circuits*. IEEE, Jun 2017, pp. C184–C185.

[18] F. ur Rahman, S. Kim, N. John, R. Kumar, X. Li, R. Pamula, K. A. Bowman, and V. S. Sathe, "A Unified Clock and Switched-Capacitor-Based Power Delivery Architecture for Variation Tolerance in Low-Voltage SoC Domains," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1173–1184, apr 2019.

[19] F. U. Rahman, S. Kim, N. John, R. Kumar, X. Li, R. Pamula, K. A. Bowman, and V. S. Sathe, "An All-Digital Unified Clock Frequency and Switched-Capacitor Voltage Regulator for Variation Tolerance in a Sub-Threshold ARM Cortex M0 Processor," in *2018 IEEE Symposium on VLSI Circuits*. IEEE, jun 2018, pp. 65–66.

[20] F. ur Rahman, R. Pamula, A. Boora, X. Sun, and V. Sathe, "19.1 Computationally Enabled Total Energy Minimization Under Performance Requirements for a Voltage-Regulated 0.38-

to-0.58V Microprocessor in 65nm CMOS," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. IEEE, feb 2019, pp. 312–314.

[21] P. J. Restle *et al.*, "A clock distribution network for microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 5, pp. 792–799, May 2001.

[22] N. Kurd *et al.*, "Next generation intel micro-architecture (nehalem) clocking architecture," in *2008 IEEE Symposium on VLSI Circuits*, June 2008, pp. 62–63.

[23] T. Fischer *et al.*, "Design solutions for the Bulldozer 32nm SOI 2-core processor module in an 8-core CPU," in *2011 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2011, pp. 78–80.

[24] D. Jeon *et al.*, "A Super-Pipelined Energy Efficient Subthreshold 240 MS/s FFT Core in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 1, pp. 23–34, Jan 2012.

[25] N. Kurd *et al.*, "Haswell: A Family of IA 22 nm Processors," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 49–58, Jan 2015.

[26] V. S. Sathe *et al.*, "Resonant-Clock Design for a Power-Efficient, High-Volume x86-64 Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 140–149, Jan 2013.

[27] P. Restle *et al.*, "Wide-frequency-range resonant clock with on-the-fly mode changing for the POWER8 microprocessor," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, Feb 2014, pp. 100–101.

[28] W. Athas *et al.*, "A low-power microprocessor based on resonant energy," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1693–1701, Nov 1997.

[29] C. Ziesler *et al.*, "A 225 MHz resonant clocked ASIC chip," in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design -ISLPED '03.* ACM, Sep 2003, pp. 48–53.

[30] S. Chan *et al.*, "A 4.6GHz resonant global clock distribution network," in *2004 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, Feb 2004, pp. 342–343.

[31] S. Chan, K. Shepard, and P. Restle, "Distributed Differential Oscillators for Global Clock Networks," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 9, pp. 2083–2094, Sep 2006.

[32] V. S. Sathe, J. C. Kao, and M. C. Papaefthymiou, "RF2: A 1GHz FIR Filter with Distributed Resonant Clock Generator," in *2007 IEEE Symposium on VLSI Circuits*. IEEE, June 2007, pp. 44–45.

[33] V. Sathe, J. Kao, and M. Papaefthymiou, "A 0.8-1.2GHz Single-Phase Resonant-Clocked FIR Filter with Level-Sensitive Latches," in *2007 IEEE Custom Integrated Circuits Conference*. IEEE, Sep 2007, pp. 583–586.

[34] S. C. Chan *et al.*, "A resonant global clock distribution for the cell broadband engine processor," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 64–72, Jan 2009.

[35] V. S. Sathe, J. C. Kao, and M. C. Papaefthymiou, "Resonant-clock latch-based design," in *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, Apr 2008, pp. 864–872.

[36] V. Sathe, "Hybrid resonant-clocked digital design," *PhD Dessertation*, 2007.

[37] V. Sathe, J. Y. Chueh, and M. Papaefthymio, "A 1.1GHz charge-recovery logic," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers (ISSCC)*. IEEE, Feb 2006, pp. 1540–1549.

[38] A. T. Ishii, J. C. Kao, V. S. Sathe, and M. C. Papaefthymiou, "A resonant-clock 200MHz ARM926EJ-S microcontroller," in *2009 Proceedings of ESSCIRC*. IEEE, Sep 2009, pp. 356–359.

[39] J.-y. Chueh, V. Sathe, and M. Papaefthymiou, "900MHz to 1.2GHz Two-Phase Resonant

Clock Network with Programmable Driver and Loading," in *IEEE Custom Integrated Circuits Conference 2006*. IEEE, Sep 2006, pp. 777–780.

[40] K. Gillespie *et al.*, "5.5 steamroller: An x86-64 core implemented in 28nm bulk cmos," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2014, pp. 104–105.

[41] R. Groves *et al.*, "Optimization and modeling of resonant clocking inductors for the power8tm microprocessor," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, Sep 2014, pp. 1–4.

[42] E. J. Fluhr, S. Baumgartner *et al.*, "The 12-Core POWER8 Processor With 7.6 Tb/s IO Bandwidth, Integrated Voltage Regulation, and Resonant Clocking," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 10–23, jan 2015.

[43] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," in *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (ISSCC)*. IEEE, Feb 2000, pp. 294–295,.

[44] H. Fuketa *et al.*, "Intermittent resonant clocking enabling power reduction at any clock frequency for 0.37V 980kHz near-threshold logic circuits," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, Feb 2013, pp. 436–437.

[45] L. J. Svensson and J. G. Koller, "Driving a capacitive load without dissipating fCV^2," in *Proceedings of 1994 IEEE Symposium on Low Power Electronics*, Oct 1994, pp. 100–101.

[46] R. Lal, W. Athas, and L. Svensson, "A low-power adiabatic driver system for AMLCDs," in *2000 Symposium on VLSI Circuits. Digest of Technical Papers*, June 2000, pp. 198–201.

[47] L. G. Salem and P. P. Mercier, "A 0.4-to-1V 1MHz-to-2GHz switched-capacitor adiabatic clock driver achieving 55.6% clock power reduction," in *2017 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2017, pp. 442–443.

[48] V. S. Sathe, "Quasi-resonant clocking: A run-time control approach for true voltage-frequency-scalability," in *Proceedings of the 2014 International Symposium on Low power Electronics and Design - ISLPED '14*. ACM Press, Aug 2014, pp. 87–92.

[49] T. Xanthopoulos, *Clocking in modern VLSI systems*. Springer Science + Business, 2009.

[50] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T Subthreshold SRAM Employing Sense-Amplifier Redundancy," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 141–149, Jan 2008.

[51] S.Basu and G. C. Temes, "Simplified clock voltage doubler," *electronic Letters*, vol. 35, no. 22, pp. 1901–1902, Oct 1999.

[52] Y. Nakagome *et al.*, "An experimental 1.5-V 64-Mb DRAM," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 224, pp. 465–472, Apr 1991.

[53] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, "An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 4, pp. 412–422, Apr 1995.

[54] N. August, H.-J. Lee *et al.*, "A TDC-less ADPLL with 200-to-3200MHz range and 3mW power dissipation for mobile SoC clocking in 22nm CMOS," in *2012 IEEE International Solid-State Circuits Conference*. IEEE, Feb 2012, pp. 246–248.

[55] C.-C. Chung and C.-Y. Ko, "A Fast Phase Tracking ADPLL for Video Pixel Clock Generation in 65 nm CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 10, pp. 2300–2311, Oct 2011.

[56] D. M. Moore, T. Xanthopoulos *et al.*, "A 0.009 mm^2 Wide-Tuning Range Automatically Placed-and-Routed ADPLL in 14-nm FinFET CMOS," *IEEE Solid-State Circuits Letters*, vol. 1, no. 3, pp. 74–77, Mar 2018.

[57] G. Shu, W. S. Choi *et al.*, "A 4-to-10.5 Gb/s Continuous-Rate Digital Clock and Data Recovery With Automatic Frequency Acquisition," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 2, pp. 428–439, Feb 2016.

[58] X. Chen, J. Breiholz *et al.*, "A 486 μW All-Digital Bluetooth Low Energy Transmitter with Ring Oscillator Based ADPLL for IoT applications," in *2018 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE, Jun 2018, pp. 168–171.

[59] H. Liu, Z. Sun *et al.*, "An ADPLL-centric bluetooth low-energy transceiver with 2.3mW interference-tolerant hybrid-loop receiver and 2.9mW single-point polar transmitter in 65nm CMOS," in *2018 IEEE International Solid - State Circuits Conference*. IEEE, Feb 2018, pp. 444–446.

[60] H. Okuni, A. Sai *et al.*, "26.1 A 5.5mW ADPLL-based receiver with hybrid-loop interference rejection for BLE application in 65nm CMOS," in *2016 IEEE International Solid-State Circuits Conference*. IEEE, Jan 2016, pp. 436–437.

[61] S. Zheng and H. C. Luong, "A WCDMA/WLAN Digital Polar Transmitter With Low-Noise ADPLL, Wideband PM/AM Modulator, and Linearized PA," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 7, pp. 1645–1656, Jul 2015.

[62] T. Burd *et al.*, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, Nov 2000.

[63] S. T. Kim, Y. C. Shih *et al.*, "Enabling wide autonomous DVFS in a 22 nm graphics execution core using a digitally controlled fully integrated voltage regulator," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 18–30, Jan 2016.

[64] P. A. Meinerzhagen, C. Tokunaga *et al.*, "An Energy-Efficient Graphics Processor in 14-nm Tri-Gate CMOS Featuring Integrated Voltage Regulators for Fine-Grain DVFS, Retentive Sleep, and VMIN Optimization," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 144–157, Jan 2019.

[65] J. L. Hennessy, D. A. Patterson, and K. Asanovic, *Computer architecture : a quantitative approach*. Morgan Kaufmann/Elsevier, 2012.

[66] H.-J. Lee, A. M. Kern *et al.*, "A scalable sub-1.2mW 300MHz-to-1.5GHz host-clock PLL for system-on-chip in 32nm CMOS," in *2011 IEEE International Solid-State Circuits Conference*. IEEE, Feb 2011, pp. 96–97.

[67] S. Geissler, D. Appenzeller, E. Cohen *et al.*, "A low-power RISC microprocessor using dual PLLs in a 0.13μm SOI technology with copper interconnect and low-k BEOL dielectric," in *2002 IEEE International Solid-State Circuits Conference*, vol. 2. IEEE, Feb, pp. 112–425.

[68] T. Anand, M. Talegaonkar *et al.*, "3.7 A 7Gb/s rapid on/off embedded-clock serial-link transceiver with 20ns power-on time, 740uW off-state power for energy-proportional links in 65nm CMOS," in *2015 IEEE International Solid-State Circuits Conference*. IEEE, Feb 2015, pp. 64–66.

[69] T. Watanabe and S. Yamauchi, "An all-digital PLL for frequency multiplication by 4 to 1022 with seven-cycle lock time," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 198–204, Feb 2003.

[70] J.-M. Lin and C.-Y. Yang, "A Fast-Locking All-Digital Phase-Locked Loop With Dynamic Loop Bandwidth Adjustment," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 10, pp. 2411–2422, Oct 2015.

[71] K.-Y. J. Shen, S. F. S. Farooq *et al.*, "19.4 A 0.17-to-3.5mW 0.15-to-5GHz SoC PLL with 15dB built-in supply noise rejection and self-bandwidth control in 14nm CMOS," in *2016 IEEE International Solid-State Circuits Conference*. IEEE, Jan 2016, pp. 330–331.

[72] C.-C. Li, M.-S. Yuan *et al.*, "All-Digital PLL for Bluetooth Low Energy Using 32.768-kHz Reference Clock and $\leq$ 0.45-V Supply," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 12, pp. 3660–3671, Dec 2018.

[73] Chia-Tsun Wu, Wen-Chung Shen *et al.*, "A Two-Cycle Lock-In Time ADPLL Design Based on a Frequency Estimation Algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 6, pp. 430–434, Jun 2010.

[74] F. Ahmad, G. Unruh, A. Iyer *et al.*, "A 0.59.5-GHz, 1.2us Lock-Time Fractional-N DPLL With 1.25% UI Period Jitter in 16-nm CMOS for Dynamic Frequency and Core-Count Scaling," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 21–32, Jan 2017.

[75] R. B. Staszewski and P. T. Balsara, *All-Digital Frequency Synthesizer in Deep-Submicron CMOS*. Wiley-Interscience, 2006.

[76] C.-S. Lin, T.-H. Chien *et al.*, "An Edge Missing Compensator for Fast Settling Wide Locking Range Phase-Locked Loops," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 11, pp. 3102–3110, Nov 2009.

[77] R. Staszewski and P. Balsara, "Phase-domain all-digital phase-locked loop," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 3, pp. 159–163, Mar 2005.

[78] S. Henzler, S. Koeppe *et al.*, "A Local Passive Time Interpolation Concept for Variation-Tolerant High-Resolution Time-to-Digital Conversion," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 7, pp. 1666–1676, Jul 2008.

[79] J. Yu, F. F. Dai, and R. C. Jaeger, "A 12-Bit Vernier Ring Time-to-Digital Converter in 0.13um CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 830–842, Apr 2010.

[80] Y. Park and D. D. Wentzloff, "A Cyclic Vernier TDC for ADPLLs Synthesized From a Standard Cell Library," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 7, pp. 1511–1517, Jul 2011.

[81] M. Lee, M. E. Heidari, and A. A. Abidi, "A Low-Noise Wideband Digital Phase-Locked Loop Based on a CoarseFine Time-to-Digital Converter With Subpicosecond Resolution," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 10, pp. 2808–2816, Oct 2009.

[82] B. Kim, H. Kim, and C. H. Kim, "An 8bit, 2.6ps two-step TDC in 65nm CMOS employing a switched ring-oscillator based time amplifier," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, Sep 2015, pp. 1–4.

[83] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2001.

[84] F. ur Rahman, "Computationally locked PLL: Verilog code." [Online]. Available: https://github.com/rfahimur26/Comp_lock_adpll.git

[85] Y. Lu, J. Jiang, and W.-H. Ki, "A Multiphase Switched-Capacitor DCDC Converter Ring With Fast Transient Response and Small Ripple," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 579–591, feb 2017.

[86] F. Yang and P. K. T. Mok, "A Nanosecond-Transient Fine-Grained Digital LDO With Multi-Step Switching Scheme and Asynchronous Adaptive Pipeline Control," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 9, pp. 2463–2474, sep 2017.

[87] M. Huang, Y. Lu *et al.*, "An Analog-Assisted Tri-Loop Digital Low-Dropout Regulator," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 20–34, jan 2018.

[88] F. U. Rahman, S. Kim *et al.*, "An All-Digital Unified Clock Frequency and Switched-Capacitor Voltage Regulator for Variation Tolerance in a Sub-Threshold ARM Cortex M0 Processor," in *2018 Symposium on VLSI Circuits*, 2018, pp. 65–66.

[89] D. Bol, J. De Vos *et al.*, "A 25MHz 7μW/MHz ultra-low-voltage microcontroller SoC in 65nm LP/GP CMOS for low-carbon wireless sensor nodes," in *2012 IEEE International Solid-State Circuits Conference*. IEEE, feb 2012, pp. 490–492.

[90] S. Gangopadhyay, D. Somasekhar *et al.*, "A 32 nm Embedded, Fully-Digital, Phase-Locked Low Dropout Regulator for Fine Grained Power Management in Digital Circuits," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 11, pp. 2684–2693, nov 2014.

[91] X. Sun, S. Kim, F. ur Rahman *et al.*, "A combined all-digital PLL-buck slack regulation system with autonomous CCM/DCM transition control and 82% average voltage-margin reduction in a 0.6-to-1.0V cortex-M0 processor," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*. IEEE, feb 2018, pp. 302–304.

[92] K. Wilcox, R. Cole *et al.*, "Steamroller Module and Adaptive Clocking System in 28 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 24–34, jan 2015.

[93] K. A. Bowman, S. Raina *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*, vol. 51, no. 1, pp. 8–17, jan 2016.

[94] A. Grenat, S. Pant, R. Rachala, and S. Naffziger, "Adaptive Clocking System for Improved Power Efficiency in a 28nm x86-64 Microprocessor," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*. IEEE, 2014, pp. 106–107.

[95] M. S. Floyd, P. J. Restle *et al.*, "26.5 Adaptive clocking in the POWER9 processor for voltage droop protection," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, feb 2017, pp. 444–445.

[96] J. M. Hart, H. Cho *et al.*, "A 3.6 GHz 16-core SPARC SoC processor in 28 nm," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 19–31, 2014.

[97] J. Myers, A. Savanth *et al.*, "A 12.4pJ/cycle sub-threshold, 16pJ/cycle near-threshold ARM Cortex-M0+ MCU with autonomous SRPG/DVFS and temperature tracking clocks," in *2017 Symposium on VLSI Circuits*. IEEE, jun 2017, pp. C332–C333.

[98] N. Kurd *et al.*, "Next Generation Intel Core Micro-Architecture (Nehalem) Clocking," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1121–1129, Apr 2009.

[99] X. Zhang, T. Tong *et al.*, "Supply-noise resilient adaptive clocking for battery-powered aerial microrobotic System-on-Chip in 40nm CMOS," in *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*. IEEE, sep 2013, pp. 1–4.

[100] D. Jiao, B. Kim, and C. H. Kim, "Design, Modeling, and Test of a Programmable Adaptive Phase-Shifting PLL for Enhancing Clock Data Compensation," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10, pp. 2505–2516, oct 2012.

[101] S. R. Sridhara, M. DiRenzo *et al.*, "Microwatt Embedded Processor Platform for Medical System-on-Chip Applications," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 4, pp. 721–730, apr 2011.

[102] G. Patounakis, Y. Li, and K. Shepard, "A Fully Integrated On-Chip DCDC Conversion and Power Management System," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 443–451, mar 2004.

[103] Y.-H. Lee, Y.-Y. Yang *et al.*, "A DVS Embedded Power Management for High Efficiency Integrated SoC in UWB System," *IEEE Journal of Solid-State Circuits*, nov 2010.

[104] R. Jain, B. M. Geuskens *et al.*, "A 0.451 V Fully-Integrated Distributed Switched Capacitor DC-DC Converter With High Density MIM Capacitor in 22 nm Tri-Gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 917–927, apr 2014.

[105] S. Bang, J. S. Seo *et al.*, "A Low Ripple Switched-Capacitor Voltage Regulator Using Flying Capacitance Dithering," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 919–929, apr 2016.

[106] W.-H. Chen, M. E. Inerowicz, and B. Jung, "Phase Frequency Detector With Minimal Blind Zone for Fast Frequency Acquisition," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 12, pp. 936–940, dec 2010.

[107] Y. Lu, W.-H. Ki, and C. Patrick Yue, "An NMOS-LDO Regulated Switched-Capacitor DCDC Converter With Fast-Response Adaptive-Phase Digital Control," *IEEE Transactions on Power Electronics*, vol. 31, no. 2, pp. 1294–1303, feb 2016.

[108] M. D. Seeman, "A Design Methodology for Switched-Capacitor DC-DC Converters," Ph.D. dissertation, 2009.

[109] N. Butzen and M. S. J. Steyaert, "Scalable Parasitic Charge Redistribution: Design of High-Efficiency Fully Integrated Switched-Capacitor DCDC Converters," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 12, pp. 2843–2853, dec 2016.

[110] J.-S. Seo, A. Young *et al.*, "Deep Trench Capacitors for Switched Capacitor Voltage Converters Outline Motivation," in *3rd International Workshop for Power Supply on Chip*, 2012.

[111] K. A. Bowman, J. W. Tschanz *et al.*, "A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, jan 2011.

[112] B. Zimmer, Y. Lee *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*, vol. 51, no. 4, pp. 930–942, apr 2016.

[113] V. S. Sathe and J.-s. Seo, "Analysis and optimization of CMOS switched-capacitor converters," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, jul 2015, pp. 327–334.

[114] Y. K. Ramadass, A. A. Fayed, and A. P. Chandrakasan, "A Fully-Integrated Switched-Capacitor Step-Down DC-DC Converter With Digital Capacitance Modulation in 45 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2557–2565, dec 2010.

[115] H.-P. Le, S. R. Sanders, and E. Alon, "Design Techniques for Fully Integrated Switched-Capacitor DC-DC Converters," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 9, pp. 2120–2131, sep 2011.

[116] N. H. E. Weste and D. M. Harris, *CMOS VLSI design : a circuits and systems perspective*. Addison Wesley, 2011.

[117] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw, "Exploring Variability and Performance in a

Sub-200-mV Processor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 881–891, apr 2008.

[118] Bo Zhai, D. Blaauw, D. Sylvester, and S. Hanson, "A Sub-200mV 6T SRAM in 0.13μm CMOS," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. IEEE, feb 2007, pp. 332–606.

[119] J. Myers, A. Savanth, D. Howard, R. Gaddh, P. Prabhat, and D. Flynn, "8.1 An 80nW retention 11.7pJ/cycle active subthreshold ARM Cortex-M0+; subsystem in 65nm CMOS for WSN applications," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*. IEEE, feb 2015, pp. 1–3.

[120] D. Bol, M. Schramme, L. Moreau, T. Haine, P. Xu, C. Frenkel, R. Dekimpe, F. Stas, and D. Flandre, "19.6 A 40-to-80MHz Sub-4μW/MHz ULV Cortex-M0 MCU SoC in 28nm FDSOI With Dual-Loop Adaptive Back-Bias Generator for 20μs Wake-Up From Deep Fully Retentive Sleep Mode," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. IEEE, feb 2019, pp. 322–324.

[121] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, jan 2005.

[122] B. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, sep 2005.

[123] A. Wang and A. Chandrakasan, "A 180mV FFT processor using subthreshold circuit techniques," in *2004 IEEE International Solid-State Circuits Conference (IEEE Cat. No.04CH37519)*. IEEE, pp. 292–529.

[124] Y. K. Ramadass and A. P. Chandrakasan, "Minimum Energy Tracking Loop With Embedded

DCDC Converter Enabling Ultra-Low-Voltage Operation Down to 250 mV in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 256–265, jan 2008.

[125] J. Lee, Y. Zhang, Q. Dong, W. Lim, M. Saligane, Y. Kim, S. Jeong, J. Lim, M. Yasuda, S. Miyoshi, M. Kawaminami, D. Blaauw, and D. Sylvester, "19.2 A 6.4pJ/Cycle Self-Tuning Cortex-M0 IoT Processor Based on Leakage-Ratio Measurement for Energy-Optimal Operation Across Wide-Range PVT Variation," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. IEEE, feb 2019, pp. 314–315.

[126] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J.-D. Legat, "A 25MHz 7μW/MHz ultra-low-voltage microcontroller SoC in 65nm LP/GP CMOS for low-carbon wireless sensor nodes," in *2012 IEEE International Solid-State Circuits Conference*. IEEE, feb 2012, pp. 490–492.

[127] D. Bol, v. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J.-D. Legat, "SleepWalker: A 25-MHz 0.4-V Sub-mm^2 7μW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 20–32, jan 2013.

[128] F. ur Rahman, S. Kim, N. John, R. Kumar, X. Li, R. Pamula, K. A. Bowman, and V. S. Sathe, "A Unified Clock and Switched-Capacitor-Based Power Delivery Architecture for Variation Tolerance in Low-Voltage SoC Domains," *IEEE Journal of Solid-State Circuits*, pp. 1–12, 2019.

[129] M. Seeman, "A design methodology for switched-capacitor dc-dc converters," *PhD Dessertation, Electrical Engineering and Computer Science, University of California, Berkeley*, 2009.

[130] F. ur Rahman and V. Sathe, "Quasi-Resonant Clocking: Continuous Voltage-Frequency Scalable Resonant Clocking System for Dynamic Voltage-Frequency Scaling Systems," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 3, pp. 924–935, mar 2018.

[131] T. M. Andersen, F. Krismer, J. W. Kolar, T. Toifl, C. Menolfi, L. Kull, T. Morf, M. Kossel, M. Brandli, P. Buchmann, and P. A. Francese, "A 4.6W/mm² power density 86% efficiency on-chip switched capacitor DC-DC converter in 32 nm SOI CMOS," in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*. IEEE, mar 2013, pp. 692–699.

[132] Y. K. Ramadass and A. P. Chandrakasan, "Minimum Energy Tracking Loop with Embedded DC-DC Converter Delivering Voltages down to 250mV in 65nm CMOS," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. IEEE, feb 2007, pp. 64–587.

[133] S. Gangopadhyay, S. B. Nasir *et al.*, "UVFR: A Unified Voltage and Frequency Regulator with 500MHz/0.84V to 100KHz/0.27V operating range, 99.4% current efficiency and 27% supply guardband reduction," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. IEEE, sep 2016, pp. 321–324.