

Low-Latency Energy-Recovery Circuitry

by

Jerry C. Kao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2011

Doctoral Committee:

Professor Marios C. Papaefthymiou, Chair
Professor David Blaauw
Assistant Professor Kevin Pipe
Assistant Professor Thomas Wenisch

© Jerry C. Kao 2011
All Rights Reserved

To my family and friends for their love and support

ACKNOWLEDGMENTS

Many people have contributed to the completion of this thesis to whom I will always be grateful. My advisor, Prof. Marios Papaefthymiou, allowed me to grow intellectually over time and provided support, encouragement, and advice over the years. My Dissertation Committee, Prof. David Blaauw, Prof. Kevin Pipe, and Prof. Thomas Wenisch, provided valuable feedback and support.

Staff of the Electrical Engineering and Computer Science department, including Beth, Denise, Bert, Lauri, Steve, and Joel, were immensely helpful. Their hard work and willingness to go out of their way made my life much easier.

Another group of people that assisted me through this journey were those in Marios' research group, including Joohee, Jiyoun, Juang-Ying, Visvesh, Wei-Hsiang, and Tai-Chuan. Discussions with them accelerated my understanding in the field of energy recovery, and broadened my knowledge in other areas that I would not time to explore on my own. In addition to those in my research group, I would also like to thank Carlos and Yu-Shiang for all their support.

Finally and most importantly, I want to dedicate my Ph.D. to my family for their support. I would like to thank my wife, LingHsuan, for going through this journey with me. She prepared hot meals and took care of my daily routines, while completing her school work and giving birth to both of our children. Life at Michigan has been much more joyful and colorful with you by my side.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xi
ABSTRACT	xii
CHAPTER	
1. Introduction	1
1.1 Adiabatic Switching Principle	6
1.2 Power-Clock Generation	9
1.3 Resonant Clocking Basics	11
1.4 Summary of Contributions	12
1.5 Thesis Outline	15
2. Background	17
2.1 Reversible Logic	17
2.2 Charge-Recovery Logic	23
2.3 Resonant-Clocked Designs	33
3. Enhanced Boost Logic	41
3.1 Introduction	41
3.2 EBL Structure	43
3.3 EBL Operation	47
3.4 EBL Clock Generation	47
3.5 EBL Energy Consumption	49

3.6	Summary	53
4.	EBL FIR Filter Test Chip	54
4.1	FIR Test-Chip Overview	55
4.2	EBL Design Methodology	61
4.3	Simulation Evaluation	65
4.4	Measurement Results	70
4.5	Summary	76
5.	Dynamic Evaluation Static Latch Logic	77
5.1	Introduction	77
5.2	DESL Structure	79
5.3	DESL Operation	81
5.4	DESL Clock Generation	82
5.5	Summary	83
6.	DESL FPU Test Chips	85
6.1	FPU Architecture	86
6.2	Clock Network Overview	93
6.2.1	Resonant Clock Network	93
6.2.2	Conventional Clock Network	100
6.3	Measurement Results	102
6.4	Summary	109
7.	Conclusions and Future Directions	110
7.1	Enhanced Boost Logic	110
7.2	Resonant-Clocked Designs	111
7.3	Future Directions	112
	BIBLIOGRAPHY	115

LIST OF FIGURES

Figure

1.1	History and trends of power-supply voltage (V_{dd}), threshold voltage (V_t), and gate-oxide thickness(t_{ox}) vs. channel length for CMOS logic technologies.	2
1.2	Heat flux trend for bipolar and CMOS chips.	3
1.3	(a) Conventional switching schematic, (b) conventional switching voltage and current plots, (c) adiabatic switching schematic, (d) adiabatic switching voltage and current plots.	7
1.4	The single-end clock generator.	9
1.5	(a) H-bridge and (b) blip clock generators.	11
1.6	A resonant-clocked system with a clock generator.	12
2.1	(a) The symbol of the Feynman gate and (b) its truth table.	19
2.2	(a) The symbol of the Toffoli gate and (b) its truth table.	19
2.3	(a) The symbol of the Fredkin gate and (b) its truth table.	20
2.4	Fredkin gate implementation of a single-bit full adder	21
2.5	(a) The symbol of Peres gate and (b) its truth table.	21
2.6	The Peres gate implementation of a single-bit full adder	22
2.7	(a) The symbol and (b) the truth table of the Kerntopf gate.	22
2.8	(a) Schematics of an NMOS ADL inverter, and (b) a PMOS ADL inverter.	24

2.9	(a) Schematic and (b) operating waveforms of a 2N-2P inverter.	25
2.10	Schematic of a 2N-2N2P inverter.	27
2.11	Schematic of a PAL inverter.	27
2.12	Schematic of a PFAL inverter.	28
2.13	Schematic of a CAL inverter.	29
2.14	(a) Schematic of a PMOS TSEL inverter and (b) an NMOS TSEL inverter.	29
2.15	(a) Schematic of a PMOS SCAL inverter and (b) an NMOS SCAL inverter.	30
2.16	(a) Schematic of a PMOS SCAL-D inverter and (b) an NMOS SCAL-D inverter.	31
2.17	Simulated operating waveforms of a 2N-2P inverter at 1GHz.	31
2.18	Schematic of an inverter implemented in Boost Logic.	32
2.19	Schematic of E-R (Edge Triggered) Latch.	34
2.20	Schematic of pTERF flip-flop.	35
2.21	Schematic of SCCER flip-flop.	36
2.22	Sense-amplifier flip-flop used in resonant-clock ARM926EJ-S.	37
2.23	Schematic of level sensitive latch used in RF1: (a) H-LAT and (b) L-LAT.	38
2.24	Schematic of level sensitive latch used in RF2, B-LAT.	38
3.1	Boost Logic schematic.	43
3.2	SBL schematic.	44
3.3	EBL buffer schematic and operation.	46
3.4	Blip clock generator and its two-phase power-clock waveforms.	48

3.5	The piecewise model used to model the blip power-clock waveform.	51
4.1	FIR block diagram with clock generator and pulse generator.	55
4.2	EBL-based 4-2 compressor schematics and layout.	56
4.3	Conversion circuits between EBL and static CMOS gates.	57
4.4	Blip clock generator with frequency scaling circuits and power-clock distribution.	58
4.5	Microphotograph of the FIR core with the inductor on the side. . .	60
4.6	Microphotograph of the FIR core with the inductor over circuit. . .	60
4.7	EBL design methodology static timing delay and slew definition. . .	64
4.8	Simulated energy consumption of self-resonant EBL FIR filter. . . .	66
4.9	Simulated energy consumption per cycle of the frequency-scaled EBL FIR filter.	67
4.10	(a) Layout of conventional static CMOS FIR filter. (b) Simulated operating frequency and energy per cycle vs. supply voltage for static CMOS FIR filter.	69
4.11	Energy consumption per cycle comparison between conventional FIR and self-resonant EBL FIR filter.	71
4.12	Energy consumption per cycle comparison between conventional FIR and frequency-scaled EBL FIR filter.	71
4.13	Energy dissipation and current vs. operating frequency of the EBL FIR core with the inductor on the side.	72
4.14	Statistics and performance summary table of the EBL FIR core with the inductor on the side.	73
4.15	Simulated and measured energy consumption per cycle comparison between the FIR filter with the inductor on the side.	74
4.16	Energy dissipation and current vs. operating frequency for FIR filter with inductor over circuit.	75
5.1	Dynamic-evaluation static-latch logic with two-phase resonant clock.	80

5.2	Operating waveforms of DESL buffer.	81
5.3	Schematic of H-bridge resonant clock generator	83
6.1	(a) FPU architecture and (b) floor plan of FPU main building blocks.	88
6.2	Multiplier implementation details.	90
6.3	Schematic of the LZA indicator bit generation circuit.	93
6.4	Distributed resonant clock generator circuitry and clock distribution network.	94
6.5	(a) Full-wave 3D field solver simulation result and (b) inductance and quality factor vs. frequency.	96
6.6	Overview of the resonant clock alignment circuitry.	97
6.7	Phase measurement circuitry.	98
6.8	Variable insertion delay block for resonant clock alignment.	99
6.9	Conventional clock distribution network.	101
6.10	Die microphotograph.	103
6.11	Built-in self-test circuitry around FPU cores.	104
6.12	Measured power consumption versus clock frequency.	105
6.13	Breakdown of power consumption at resonant frequency of 1.81GHz. Clock and logic power for the resonant FPU are obtained from measurements, and total power for the conventional FPU is obtained from measurements. Clock and logic power breakdown for the conventional FPU is derived from transistor-level simulations.	106
6.14	Measured energy per cycle versus clock frequency.	107
6.15	Measured chip energy efficiency.	108

LIST OF TABLES

Table

1.1	2009 ITRS Roadmap.	3
4.1	FIR performance comparison table.	73
6.1	FPU performance summary table.	106
6.2	FPU performance comparison table.	109

LIST OF ABBREVIATIONS

EBL Enhanced Boost Logic

FIR Finite Impulse Response

DESL Dynamic Evaluation Static Latch

ABSTRACT

Low-Latency Energy-Recovery Circuitry

by

Jerry C. Kao

Chair: Marios C. Papaefthymiou

Voltage scaling has diminished with each advancement in process technologies, making power dissipation one of the primary design considerations for modern digital systems across all market segments. This dissertation describes a novel charge-recovery logic family and a resonant-clocked dynamic logic that utilize energy-recovery techniques to recycle charge from the system, effectively reducing dynamic power dissipation.

We propose a novel charge-recovery logic, called Enhanced Boost Logic, which achieves high efficiency and high performance operation through the use of aggressive voltage scaling, gate overdrive, and charge-recovery techniques. Fabricated using a $0.13\mu\text{m}$ technology with a 3nH on-chip inductor, a 14-tap 8-bit FIR filter implemented using Enhanced Boost Logic achieves operating frequency up to 600MHz with only 1.5 cycles of latency overhead compared to a static CMOS implementation. At its natural frequency of 466MHz , the FIR dissipates 39.1mW . With a figure of merit equal to $93.6\text{nW}/\text{MHz}/\text{Tap}/\text{InBit}/\text{CoeffBit}$, it achieves 29% improvement compared to previously reported FIR filter test-chips with equal or greater sampling rate than our design at its 466MHz resonant point.

We also propose a dynamic logic family, call Dynamic Evaluation Static Latch,

synchronized by a two-phase resonant clock, which achieves dynamic-logic levels performance with significant power reduction. Fabricated in a 90nm technology with an 0.41nH integrated inductor, a resonant-clocked FPU implemented using this methodology achieves clock speeds up to 2.07GHz. At its resonant frequency of 1.81GHz, it dissipates 334mW, yielding 31.5% lower power and 32% more GFLOPS/W over a conventionally-clocked version of the same FPU implemented on the same die. Relying on circuit, logic, and architectural optimizations, the resonant-clock FPU breaks new ground along several metrics. Specifically, with a total area of 0.391mm² including the on-chip inductor, it occupies the smallest footprint among competing stat-of-the-art reduced-latency FPUs. Moreover with an overall latency of 64 FO4, it achieves the shortest overall latency among state-of-the-art reduced-latency FPUs. Delivering 10.82GFLOPS/W, this resonant-clock FPU achieves the highest energy efficiency among state-of-the-art continuously data-streaming FPUs.

CHAPTER 1

Introduction

For the last several decades, the exponential growth in transistor density as described by Moore's Law has been the main driving force behind the information revolution. With each process technology advancement, faster and less power hungry devices coupled with the cost benefit of high integration have enabled the use of Very Large Scaled Integrated (VLSI) systems in a growing number of applications to the point that they have become pervasive in daily life.

As CMOS process technology advances, reducing power dissipation has become one of the primary design considerations. CMOS was picked as the prevailing VLSI technology after bipolar due to its unique characteristic of negligible standby power, which allows large numbers of transistors to be integrated in a chip. In the 1970's, Dennard et al. described a constant-electrical-field scaling methodology [1] in which the supply voltage and critical transistor dimensions (channel length, junction depth, and oxide thickness) are scaled by the same scaling factor κ to achieve speed improvement by κ and power reduction by $1/\kappa^2$, while power density remains constant. However, as channel length gets smaller, new generations of devices are scaled using a less aggressive scaling methodology, where critical dimensions are scaled at a different rate compared to supply voltage and threshold voltage. Figure 1.1 shows the historical trend of supply voltage and threshold voltage over channel length. As channel length

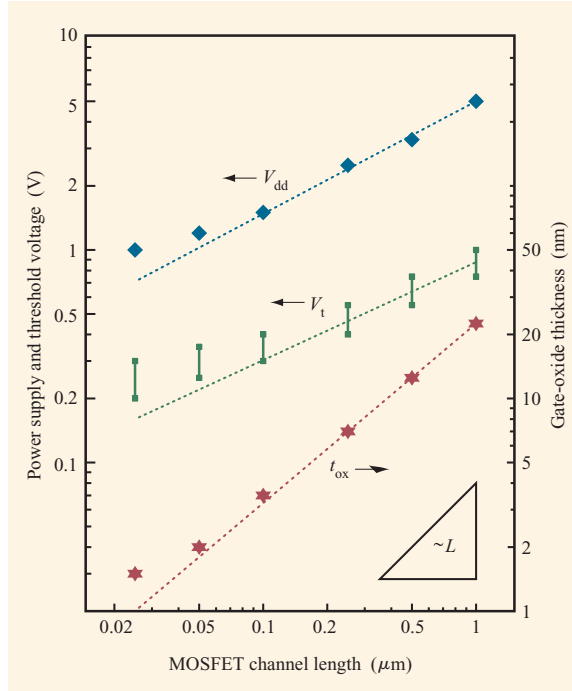


Figure 1.1: History and trends of power-supply voltage (V_{dd}), threshold voltage (V_t), and gate-oxide thickness(t_{ox}) vs. channel length for CMOS logic technologies. [2]

shrinks below $1\mu\text{m}$, supply voltage and threshold voltage are progressively scaled at a slower rate compared to channel length to combat variation in threshold voltage, yielding larger dynamic power dissipation and power density. In addition, standby power also increases exponentially over generations due to quantum mechanical effect from thinner gate oxide and shorter channel length. As we enter the era of nanoscale devices, supply voltage and threshold voltage deviate further from historical scaling trends, yielding even larger power density to the extent that power densities in some CMOS designs are approaching those of bipolar designs right before bipolar technology was phased out, as shown in Figure 1.2. This exponential growth in the power density is not sustainable, and more innovations in low power techniques are required to keep the trend of performance scaling.

In high-end microprocessors, power dissipation has become such an important design consideration that is limiting system performance. Table 1.1 lists the In-

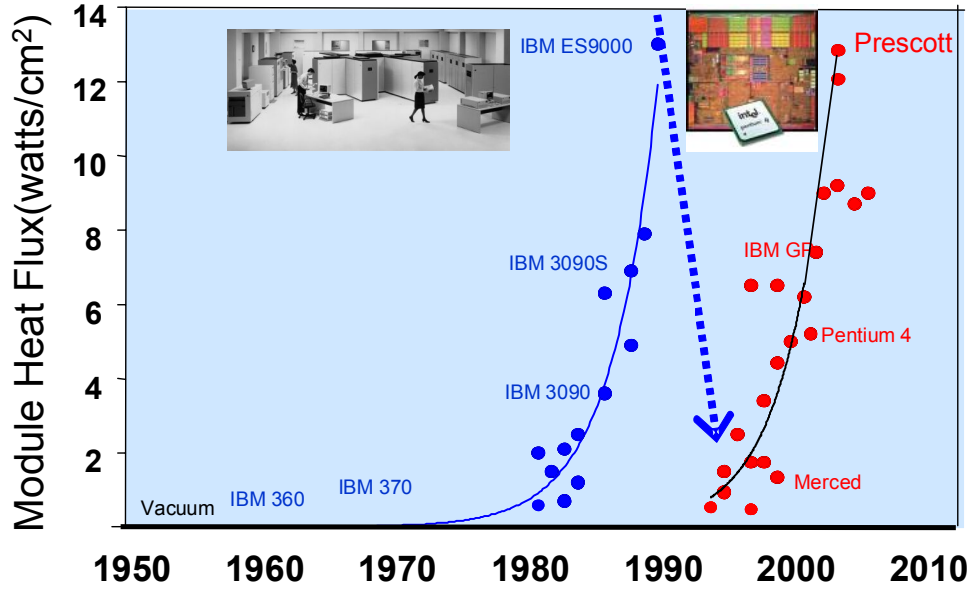


Figure 1.2: Heat flux trend for bipolar and CMOS chips. [3]

ternational Technology Roadmap for Semiconductor (ITRS) projections for high-performance microprocessors over the next few years [4]. The roadmap projects that the threshold voltage of the CMOS devices will continue to scale at a much slower rate than channel length, and thus limits the rate at which supply voltage is scaled. Due to the $100\text{W}/\text{cm}^2$ air cooling limit, the roadmap also projects that power dissipation of high-end microprocessors increases slightly in the short term but will eventually return to 2009 levels, which limits the rate at which local clock frequency is scaled. This roadmap shows that performance of high-end microprocessors is mainly constrained by power dissipation, and implies that with novel low-power circuit techniques more performance can fit within a single chip.

Year	Tech(nm)	Local Freq (GHz)	High Performance (W)	Vdd (V)	Vth (V)
2009	47	5.454	143	1.00	0.3300
2010	41	5.875	146	0.97	0.3201
2011	35	6.329	161	0.93	0.3069
2012	31	7.344	158	0.90	0.2970
2013	28	7.911	149	0.87	0.2871
2014	25	8.522	152	0.84	0.2772
2015	22	9.180	143	0.81	0.2673

Table 1.1: 2009 ITRS Roadmap.

The 2009 ITRS roadmap shows that despite technology advances in materials such as high-K gate isolator and copper interconnect, scaling trends are barely keeping up in terms of density but not in terms of performance. Therefore, more innovations are needed to keep up with performance scaling, while satisfying power constraints. At the circuit level, clock gating and power gating have successfully reduced dynamic and standby power dissipation, respectively. At the architectural level, multi-core designs have been instrumental in achieving breakthrough performance with good energy efficiency. These innovations merely extend the performance scaling of the planar CMOS structure, while maintaining operation within power dissipation bounds. Significant innovations are required to maintain performance scaling trends.

Energy recovery is a design approach that can help keep performance scaling on track. An energy-recovery system saves power by charging capacitance gradually and recycling the charge at the end of each cycle. Since energy-recovery systems operate in a different manner from static CMOS, the energy dissipation of an energy-recovery system is governed by the equation $E_{er}=(k/T)CV^2$, where k is a constant of proportionality, and T is the computation cycle period. By relying on the energy/latency tradeoff, as described in this energy dissipation equation, energy recovery is a promising circuit technique for significantly reducing dynamic power dissipation below CV^2 , the fundamental limit of static CMOS.

Energy-recovery techniques can be classified in two categories. Fine-grained energy recovery circuitry, also known as adiabatic circuitry or charge-recovery logic, deploys energy-recovery techniques at the gate level, recovering charge supplied to every gate, making it highly energy efficient for circuit with high switching activities. Coarse-grained energy recovery circuitry focuses on applying energy-recovery techniques on nets with high switching activities to achieve high energy efficiency. The most common candidate for such techniques is the clock distribution network, and this technique is known as resonant clocking. Resonant clocking can be applied at the

global distribution level, or all the way to the leaves at the timing elements. The use of resonant clocking for global clock distribution allows for some power savings and easy adaptation, while maintaining the clock gating functionality at the local clock distribution. Resonant clocking deployed all the way to the leaves of the clock network enables significantly larger power savings, since the majority of the clock-related capacitance is at the leaf level.

The contribution of this work is in the creation of novel fine-grained and coarse-grained energy-recovery circuits to achieve low-latency high-performance operation, while achieving high energy efficiency. Previous generations of fine-grained energy recovery logic reduce the number of clock phases and the complexity of the logic function in each gate to achieve operating frequencies comparable to static CMOS, yielding orders of magnitude higher latency. Our novel fine-grained energy recovery circuitry simplifies the number of power supplies, maintains the use of two-phase power clocks, and most importantly, enables the implementation of more complex logic functions in each gate to achieve a new level of energy/latency tradeoff and minimal latency overheads compared to static CMOS implementations of identical architecture. Our novel coarse-grained energy recovery circuitry combines a precharge logic with resonant clocking to achieve dynamic logic performance levels with significant reduction in clock-related power. To demonstrate the high performance and energy efficiency of this logic, we used it to implement a fused-multiply-add floating-point unit. Relying on circuit, logic, and architectural optimizations, the reduced-latency floating-point unit achieves the shortest overall latency published to date.

The remainder of this chapter is organized as follows: Section 1.1 discusses the basic principle of adiabatic switching. Section 1.2 presents generation of power-clock that is used to synchronize and power adiabatic-switching circuitry. Section 1.3 presents the basics of resonant clocking. The contributions of this work are summarized in Section 1.4, and the outline of this dissertation is covered in Section 1.5.

1.1 Adiabatic Switching Principle

Digital systems depend on the voltage levels of nodes to determine logic values. Conventional static CMOS charges and discharges a node by configuring transistors to connect the node to V_{DD} or ground, respectively. The energy dissipated in such a system is in the resistance associated with switching devices as current rushes through to charge or discharge the output loads.

In adiabatic switching, transitions are spread out over time, and energy consumption can asymptotically approach zero, as implied by the term adiabatic borrowed from thermodynamics. Circuit families that employ this principle are often referred to as *adiabatic* or *charge-recovery logic*. Since CMOS devices are not ideal switches, however, adiabatic logic cannot recover all the energy sent to its outputs and always dissipates some finite amount of energy. Chapter 2 gives more detailed background on such logic. This section discusses the adiabatic switching principle that forms the basis for its operation.

Figure 1.3(a) shows a conventional switching system, where the output load is charged and discharged by constant supplies V_{DD} or ground. Figure 1.3(b) shows the transient voltage across the output load C_L and the transient current toward the output load during a charge and discharge cycle. In the beginning of the charge or discharge cycle, the voltage drop across the resistive element is at its highest, yielding a spike in current profile. As the output node gets charged or discharged toward the voltage of the supply, the voltage drop across the resistive element drops, yielding a reduction in current until the circuit reaches an equilibrium when the output is fully charged or discharged to the voltage of the supply that it is connected to. The power dissipation of this system can be computed by integrating the instantaneous power dissipation I^2R , and the energy consumption of the conventional switching circuit

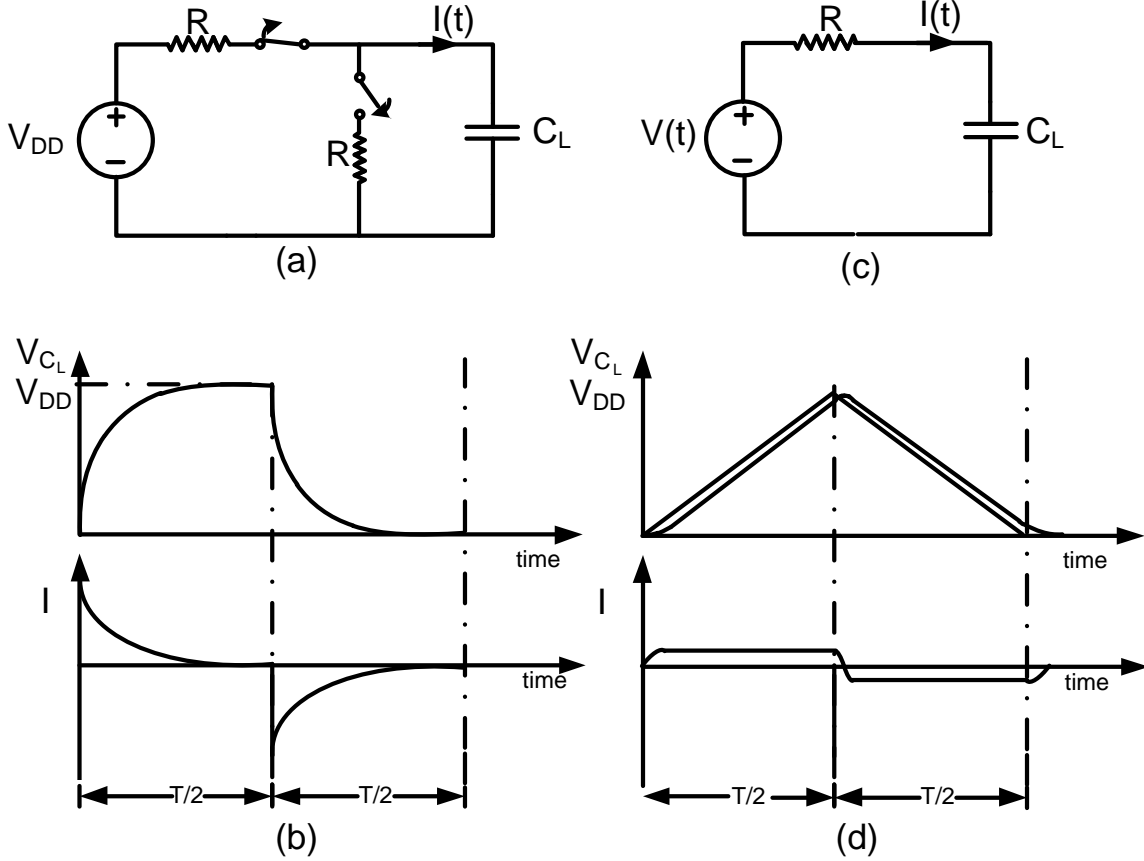


Figure 1.3: (a) Conventional switching schematic, (b) conventional switching voltage and current plots, (c) adiabatic switching schematic, (d) adiabatic switching voltage and current plots.

during each charge or discharge cycle is given by the equation:

$$E_{conv} = \frac{1}{2}CV_{DD}^2. \quad (1.1)$$

Figure 1.3(c) shows an energy-recovery system where the output load is charged and discharged by a gradually changing supply. Figure 1.3(d) shows the supply voltage and the voltage across the output load and the transient current toward the output load. For the purposes of this discussion, the gradually changing supply is a triangle waveform, where the voltage ramps up and down linearly. Since supply voltage changes gradually, the voltage across the output load follows the supply voltage

closely. The resulting current flowing toward the output load is gradual, remaining constant throughout each charge and discharge cycle, and has a magnitude that is much smaller than the peak current of a conventional switching system. Based on first order analysis, the current going toward the output load during $0 < t < T/2$ can be described as:

$$I_{er} = \frac{2V_{DD}}{T}C_L. \quad (1.2)$$

The energy consumed in such a system can be computed by integrating the instantaneous power I^2R as shown below:

$$\begin{aligned} E_{er} &= \int_0^{T/2} I_{er}^2 R dt \\ &= \int_0^{T/2} \left(\frac{2V_{DD}}{T}C_L\right)^2 R dt \\ &= \frac{RC_L}{T}C_LV_{DD}^2. \end{aligned} \quad (1.3)$$

Equation (1.3) implies that the dynamic energy consumption of an energy recovery system can asymptotically approach zero if the cycle period approaches infinity. By taking advantage of this tradeoff between energy and latency, the energy recovery system has the potential to achieve higher energy efficiency over a conventional switching system. However, due to leakage current, per-cycle energy consumption starts to increase after the operating frequency decreases beyond a certain point. Thus, the right balance between switching and leakage energy dissipation must be found to achieve minimal energy dissipation.

Another implication of Equation (1.3) is that the power dissipation $V_R I$ can be minimized by finding the optimal sizing for the switches for a fixed operating frequency. The voltage drop across the resistive element V_R depends on the size of the switches and the width of the wire, and the current I depends on the capacitance

C and the cycle period. Up-sizing the switch and the wire reduces resistance, but increases capacitance, yielding an increase in current. The goal is to find the optimal size for the switches and the wire by managing the tradeoff between V_R and I .

1.2 Power-Clock Generation

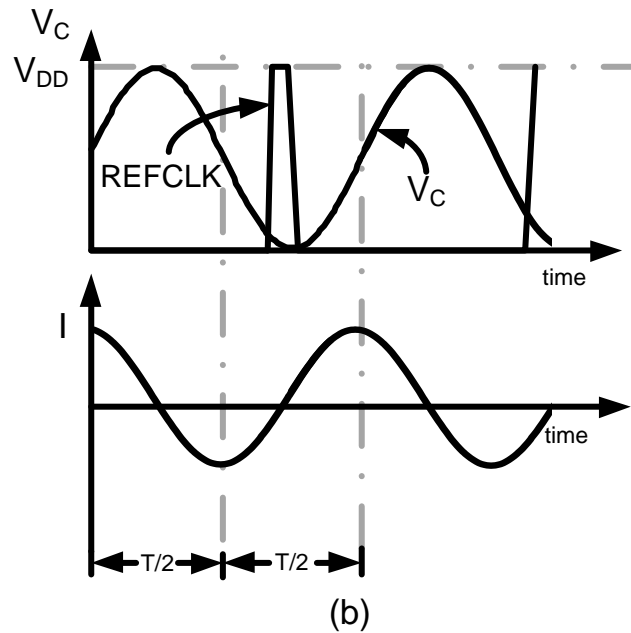
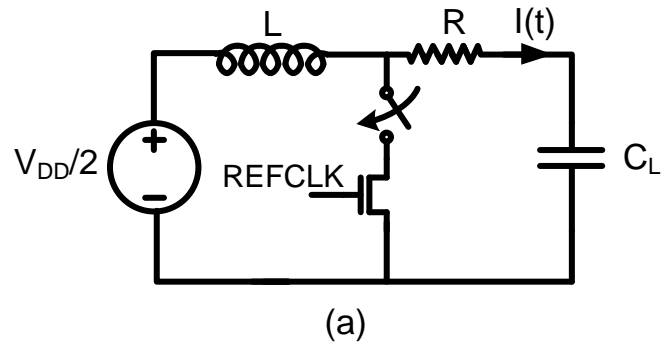


Figure 1.4: The single-end clock generator.

One possible realization of the gradually changing supplies mentioned in Section 1.1 is through an LC oscillation. Fig. 1.4 shows a single-ended clock generator,

which uses an inductor L to form an LC oscillation with the parasitic capacitance C_L . This single-ended clock generator provides a sinusoidal waveform, which has a gradual transition. It is capable of delivering and absorbing energy to and from the output loads. Moreover, it is very energy efficient, since the only loss in the system is due to the resistance associated with the generation and the distribution of the sinusoidal waveform. To sustain the oscillation, a switch driven by an reference clock source REFCLK replenishes the energy lost. The clock generator can be forced to run off-resonance and operate at the frequency of the reference clock. However, it is most energy efficient when operated at its natural oscillation frequency, which is described by Equation (1.4).

$$f = \frac{1}{2\pi} \sqrt{\frac{1}{LC_L} - \zeta^2}, \quad (1.4)$$

where C_L is the parasitic capacitance of the system, L is the inductance, and ζ is the damping factor. Since this gradual supply can provide both the powering and synchronization mechanism for charge-recovery gates, it is also known as a *power-clock*.

Variations of the basic power-clock generator design have been proposed to generate power-clocks with different characteristics. Fig. 1.5(a) shows the H-bridge clock generator that generates a two-phase clock with the crossover of the two clock phases at half V_{DD} [5]. Fig. 1.5(b) shows the blip clock generator which generates a two-phase almost-non-overlapping power-clock [6]. The clock amplitude generated by the blip clock generator is much larger than its supply voltage V_{DC} . The effective capacitance of these two power-clock generators is half of C_L , making their natural frequencies roughly $\sqrt{2}$ times larger than the single-ended clock generator.

1.3 Resonant Clocking Basics

In this section, we present the basics of resonant clocking, which is a type of coarse-grained energy-recovery technique. Unlike fine-grained energy-recovery where charge is recovered from every gate, coarse-grained energy-recovery concentrates on recovering charge from nets with high switching activities, and in the case of resonant clocking, the net of interest is the clock network which switches twice each cycle.

Resonant clocking saves energy by recycling charge at the end of each cycle. Unlike conventional clocking where charge on the clock network is dumped to ground at the end of every cycle, resonant clocking recovers the charge and stores it in a reservoir. The reservoir is usually implemented using an inductor, which enables the energy on the clock network to be stored either as magnetic energy in the inductor or as electric energy in the capacitance of the clock load and distribution network. The oscillation frequency of the resonant clock is set by LC resonance. The energy dissipated in the resonant clock is on the resistance associated with the inductor and the clock distribution network. With appropriate design of the inductor and the clock distribu-

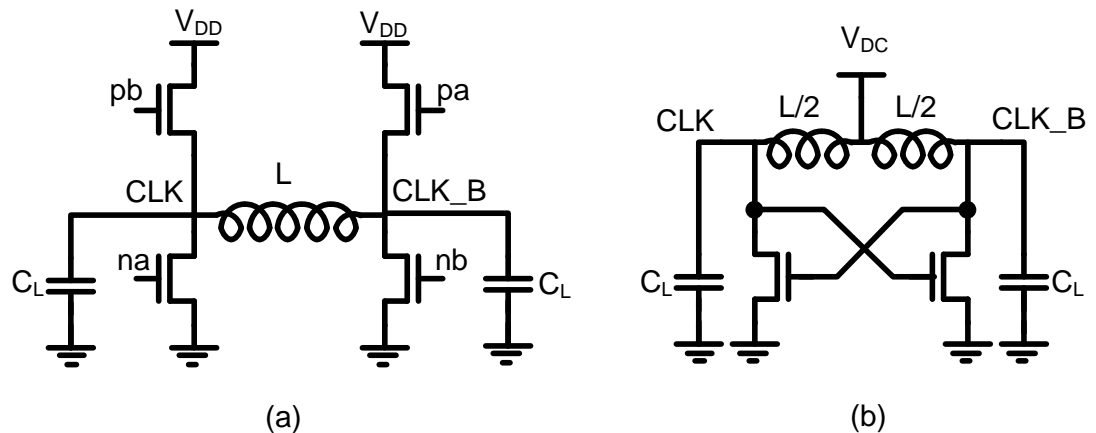


Figure 1.5: (a) H-bridge and (b) blip clock generators.

tion network, the resistance can be minimized, yielding significant savings in energy dissipation compared to conventional clocking. In contrast to fine-grained energy recovery, resonant clocking can be relatively easily applied to designs implemented with conventional clocking, since it does not require inserting buffers to balance out the delay, which increases the range of applications that resonant clocking can be applied to. In addition, since implementation of resonant clocking only requires modifying the clock distribution network, resonant clocking designs are more amenable to the use of commercial computer-aided design tools with some small modifications to the conventional design flow.

1.4 Summary of Contributions

This section outlines the contributions of this thesis proposal in the area of charge-recovery logic and resonant-clocked design. The main motivation of the work behind both areas is in the development of novel circuit structures to achieve operating frequencies beyond those achieved by previous generations of energy recovery circuits, while attaining lower power dissipation and lower or comparable overall latency com-

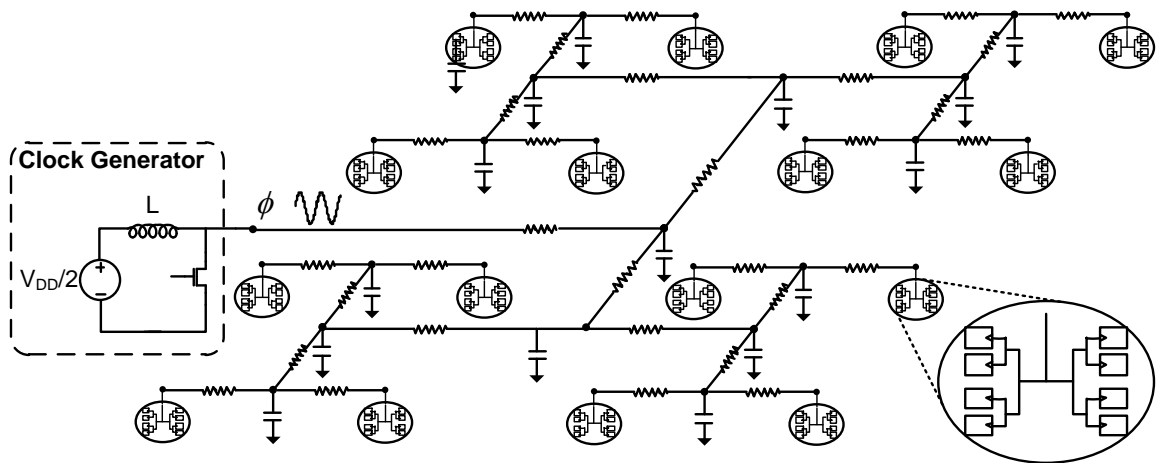


Figure 1.6: A resonant-clocked system with a clock generator.

pared to static CMOS design. These novel circuit techniques are applied to real life applications to demonstrate their low-power high-performance properties.

Recent charge-recovery logic has achieved operating frequencies that are higher or comparable to those of static CMOS. Its overall latency is at least an order of magnitude higher than static CMOS designs, however, due to the use of fewer clock phases and simpler logic functions in each gate to achieve higher operating frequencies. A 1GHz 16-bit multiplier implemented in a recent charge recovery logic has an overall latency of 24 cycles [7].

This thesis presents a novel charge recovery logic family, called Enhanced Boost Logic, to overcome the limitations of previous generations of charge-recovery logic families and to achieve a new level of energy/latency tradeoff [8, 9]. Simulations showcasing the capabilities of our charge-recovery logic are presented. A semi-custom charge-recovery design methodology is described to enable the use of commercial CAD tools to shorten design and verification time.

A 14-tap 8-bit FIR filter was designed using our charge recovery design methodology, and a FIR filter test-chip with an integrated clock generator was fabricated in IBM 0.13 μm CMOS technology to demonstrate its energy efficiency. Measurement results from our FIR filter show that our fine-grained energy-recovery circuit technique reduces power dissipation by 29% and has only 1.5 cycles of latency overhead over static CMOS, which is a significant improvement over previously published charge-recovery logic. Our FIR filter test-chip performs the most complex logic function over all previously fabricated charge-recovery designs, which implemented 16-bit adders or multipliers.

This thesis also presents contributions to resonant clocked design [10]. Most of the previously-published resonant clocked designs utilized flip-flops as the timing elements, and published operating frequencies are around the 200MHz mark [11, 12]. A coarse-grained energy-recovery design with resonant-clock latches has been recently

demonstrated at operating frequencies as high as 1GHz [13, 14]. A key factor for achieving such high performance is that level-sensitive latches are insensitive to the poor slew of the sinusoidal waveform of the resonant clock.

In this work, we propose a circuit structure to achieve low-power high-performance datapaths. The key idea behind this work is in the combination of dynamic logic and resonant clocking to achieve dynamic logic performance levels while maintaining high energy efficiency. The dynamic logic structure consists of a dynamic evaluation stage between transparent latches that are synchronized by a two-phase resonant clock. Since the entire clock network is resonated, clock-related power is greatly reduced. The transparent latch between the evaluation stages reduces the performance degradation caused by the poor slew of the resonant clock and isolates the double transition typically associated with precharge logic to the smaller capacitance on the dynamic nodes, yielding additional saving in switching power.

To demonstrate the effectiveness of our proposed circuit structure, we applied it to a real life application. In this case, a fused-multiply-add single-precision floating-point unit (FPU) was chosen as the demonstration vehicle, and a test-chip was fabricated in TSMC 90nm CMOS technology. The chip includes two designs: a conventionally clocked FPU, and a resonant clocked FPU. Both designs employed the same dynamic evaluation structure synchronized using a transparent latch. Relying on circuit, logic, and architectural optimizations, both FPUs achieve an overall latency of 64 FO4, which is the shortest overall latency for FPUs. The test-chip is the largest and fastest design presented to date with resonant clocking deployed all the way to the leaves of the clock network. It is also the first datapath that combines dynamic logic with resonant clocking.

1.5 Thesis Outline

This thesis is organized as follows. In Chapter 2, we survey previous work done in energy-recovery circuits. We first discuss the history of reversible logic, which is the origin of all energy-recovery techniques. We then cover previous work in the area of charge-recovery logic and resonant clocking, showing the evolution of circuit structures.

In Chapter 3, we present our work on fine-grain energy-recovery circuit design. Specifically, we present a novel fine-grained energy recovery logic family, called Enhanced Boost Logic (EBL). By showing the structure and operation of EBL, we demonstrate its improvements over previous charge-recovery logic.

In Chapter 4, we discuss a FIR filter implemented in this novel logic family along with an integrated clock generator. A test-chip was fabricated in IBM $0.13\mu\text{m}$ CMOS technology with two FIR cores, where one has the integrated inductor on the side, and the other has the integrated inductor over circuitry. This chapter also presents simulation and measurement results from our FIR designs. This work was published in [8, 9].

Chapter 5 discusses our work on coarse-grained energy-recovery circuit design. Specifically, we present the Dynamic Evaluation Static Latch (DESL) logic family, which combines precharge logic with level-sensitive latches to achieve dynamic-logic levels of performance. When DESL is synchronized using a two-phase resonant clock, it achieves high energy efficiency by reducing the clock-related power.

In Chapter 6, we describe a fused-multiple-add floating-point unit implemented using this technique to demonstrate its high performance and low power. We also describe a test-chip that has been fabricated in TSMC 90nm CMOS technology with two FPU cores where one is clocked conventionally, and the other is clocked by resonating the entire clock network. Measurement results from the two FPUs are presented, showing that the resonant FPU achieves 31.5% lower power consumption over its

conventional counterpart when operating at 1.81GHz.

Chapter 7 summarizes our contributions and presents directions for future research in this area.

CHAPTER 2

Background

This chapter surveys previous work in the area of energy-recovery circuits. In Section 2.1, we describe reversible logic, a class of circuit families that inspired later energy-recovery techniques. Section 2.2 covers early research in the area of charge-recovery logic, showing the evolution of circuit structures between different generations of charge-recovery logic, and provides background for our work in Chapters 3 and 4. Section 2.3 discusses previous work in the area of resonant clocking, and provides background for our work in Chapters 5 and 6.

2.1 Reversible Logic

The origins of energy-recovery techniques can be traced back to reversible logic. Reversible logic achieves potentially zero-energy operation by eliminating change in entropy during each computation. It came to light as physicists explored the fundamental physical limitation of computation. In the 1960s, Rolf Landauer observed that many operations manipulate 0s and 1s in an irreversible fashion, thus reducing the degrees of freedom represented by these bits. By the Second Law of Thermodynamics, an expansion in other degrees of freedom, in the form of entropy increase, must occur, which leads to dissipated energy, usually in the form of heat. Landauer concluded that each irreversible operation consumes at least $kT \ln 2$ [15], where k is

Boltzmann's constant, and T is the absolute temperature of the system. He also posed the question of whether irreversible operations are unavoidable to accomplish useful computation [16].

In the 1970s, Charles H. Bennett constructed the first general-purpose reversible computers in the form of a reversible Turing Machine [17]. By erasing all of intermediate results and keeping only the original input and the desired output, the reversible Turing Machine demonstrated that useful work can be done while potentially dissipating considerably less than kT of energy per operation. In the early 1980s, Fredkin and Toffoli proposed a universal reversible logic gate [18], which subsequently became known as the Fredkin gate. Since the Fredkin gate is universal, it can be used to synthesize any computing device. Besides the Fredkin gate, many reversible logic gates have been proposed, including the Toffoli gate, the Feynman gate, the Peres gate, and the Kerntopf gate [19].

For a function to be reversible, each of its outputs must have a unique input associated with it. To satisfy this one-to-one mapping between inputs and outputs, a (m, n) gate with m inputs and n outputs must have the same number of outputs and inputs, thus enabling inputs to be derived by solely observing outputs. Functions that satisfy this one-to-one mapping are also known as bijective. For example, a $(1, 1)$ inverter is a reversible gate, since it satisfies the bijective property. However, a $(2, 1)$ NAND is not a reversible gate, since there are three possible inputs ('00', '01', '10') that would result in a '0' on the output, making it impossible to determine which of them is the original input. Fig. 2.1 shows the $(2, 2)$ Feynman gate [20]. It is a reversible gate, and arrows in the truth table demonstrate the one-to-one mapping property between inputs and outputs. It can be generalized to be a k -bit CONTROL-NOT (k, k) gate, which passes the first $k-1$ bits through unchanged, while inverting the last input if all of the $k-1$ bits are 1s.

Fig. 2.2 shows the $(3, 3)$ Toffoli gate [21], which is a 3-bit CONTROL-NOT $(3,3)$

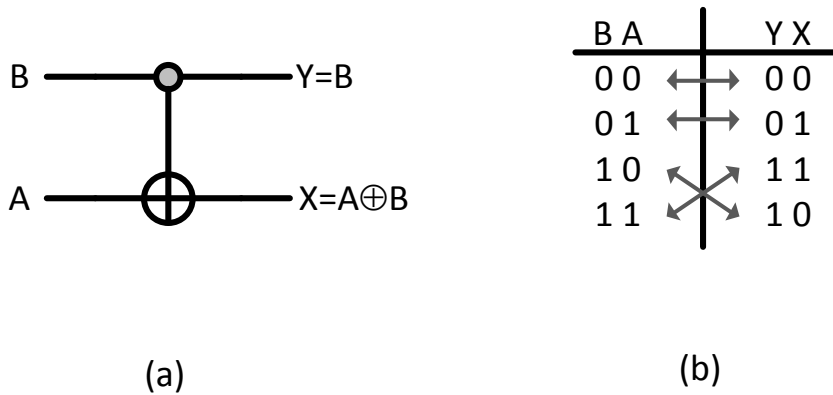


Figure 2.1: (a) The symbol of the Feynman gate and (b) its truth table.

gate. Besides being logically reversible, the Toffoli gate is also universal, since it can be programmed to function as a 2-input NAND gate by setting input C to be 1, and a fan-out gate, which duplicates its input, by setting inputs B and C to be 1 and 0, respectively. Thus, the Toffoli gate is a basic primitive that can be used to build any reversible logic circuit. In addition, the Toffoli gate is self-reversible, meaning that

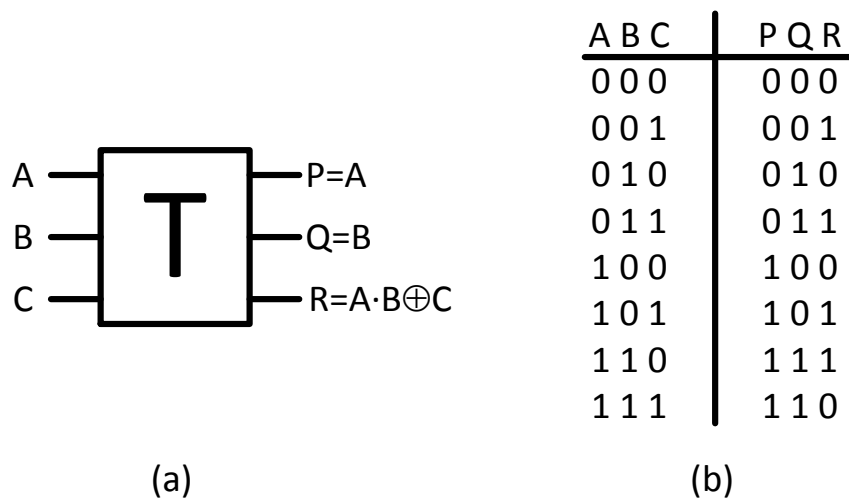


Figure 2.2: (a) The symbol of the Toffoli gate and (b) its truth table.

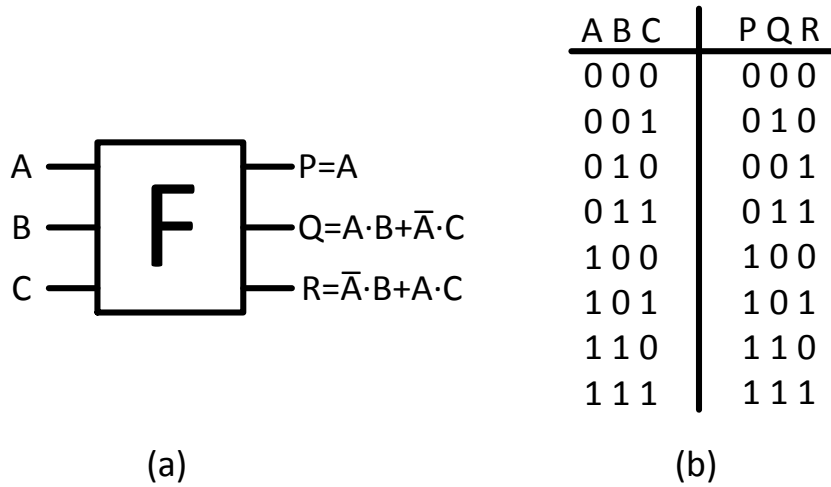


Figure 2.3: (a) The symbol of the Fredkin gate and (b) its truth table.

its boolean function is its own inverse.

In the early 1980s, Fredkin and Toffoli proposed the Fredkin gate [18]. Fig. 2.3 shows the gate symbol and the truth table of the Fredkin gate. Similar to the Toffoli gate, the Fredkin gate is bijective, universal, and self-reversible. Since its output preserves the number of 1s in its input, it is also conservative, making the construction of sequential circuits with zero internal power possible. Fig. 2.4 shows the Fredkin gate implementation of a single-bit full adder. Note that outputs marked in dash lines do not contribute to the desired outputs Sum and C_{out} but they have to be retained in order for the full adder to be reversible. These unused bits are also known as garbage bits, and they should be minimized to improve hardware efficiency.

Later generations of reversible logic focus on increasing hardware efficiency when implementing complex arithmetic operations. In 1985, Asher Peres proposed the Peres gate [22], as shown in Fig. 2.5. Even though it is not self-reversible, it has the advantage that two of them are sufficient to realize a single-bit full adder, as shown in Fig. 2.6. Moreover, the Peres gate implementation of a full adder has only 2 garbage bits and 1 input constant, making it more hardware efficient than the Fredkin-gate

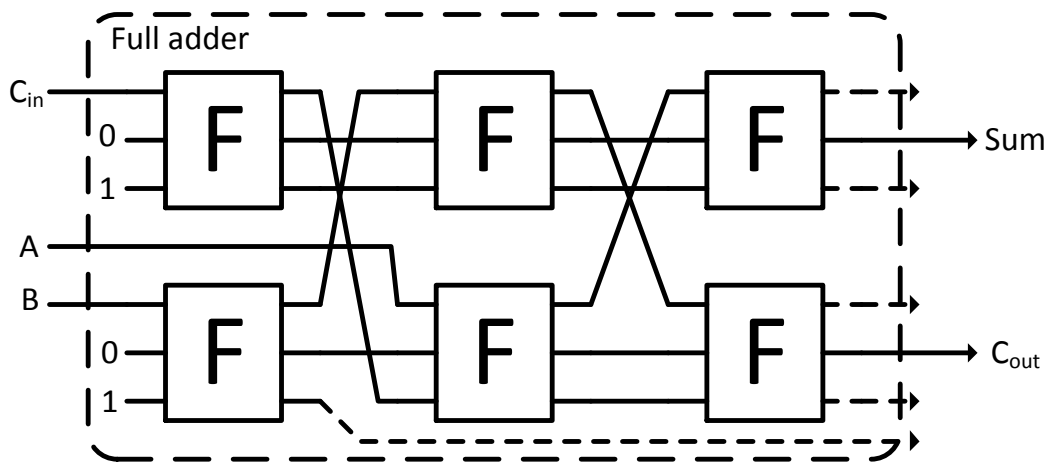


Figure 2.4: Fredkin gate implementation of a single-bit full adder

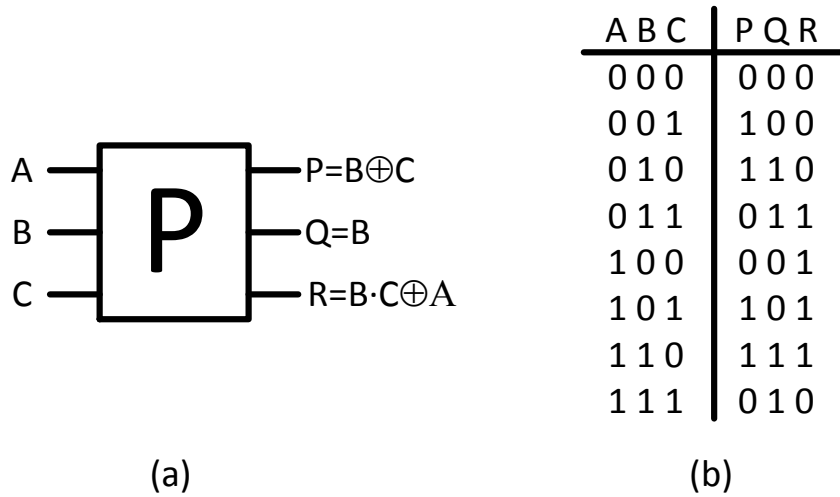


Figure 2.5: (a) The symbol of Peres gate and (b) its truth table.

implementation.

Fig. 2.7 shows the gate symbol and the truth table of a (3, 3) Kerntopf gate [23], proposed in 2004. The Kerntopf gate has the advantage of having more cofactors than previous reversible gates (18 cofactors in the (3, 3) configuration). It is, thus, able to realize more subfunctions with the same inputs and constants.

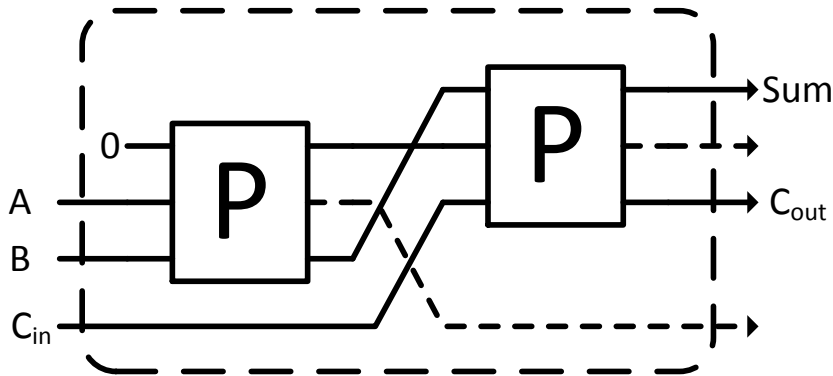


Figure 2.6: The Peres gate implementation of a single-bit full adder

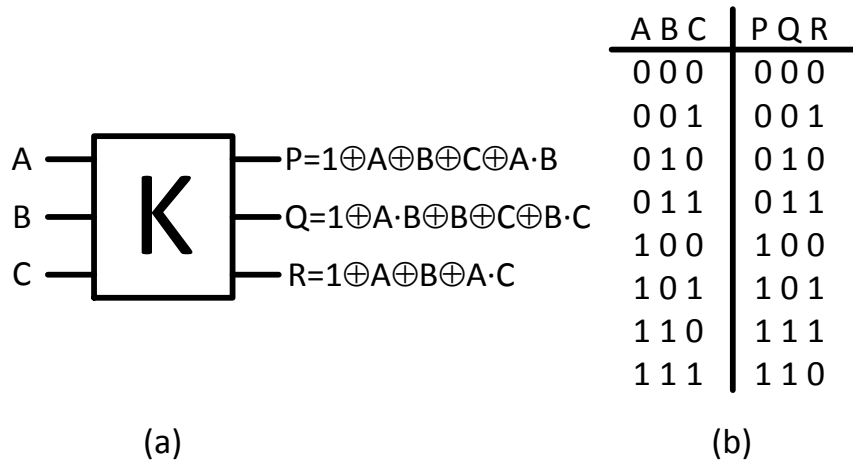


Figure 2.7: (a) The symbol and (b) the truth table of the Kerntopf gate.

Implementing reversible logic in integrated circuits has many challenges and issues. As suggested in [24], reversible computation requires all logic operations to be carried out once in the forward direction and once in the backward direction, yielding additional latency and circuit overhead. More importantly, it requires a large amount of temporary storage to maintain intermediate results until computation in the backward direction is ready. Since storing these temporary values results in energy and circuitry overheads, implementing fully logically reversible logic in CMOS is not particularly attractive.

2.2 Charge-Recovery Logic

Charge-recovery logic is a class of circuitry that recycles energy from the outputs of logic gates to achieve ultra-low power operation. Early charge-recovery logic, such as Split-level Charge-Recovery Logic (SCRL) [25, 26] and Reversible Energy Recovery Logic (RERL) [27], implemented fully logically reversible gates in CMOS. By implementing an additional inverse function in the reverse direction, each gate returns energy on computation in the forward direction back to its inputs. If computations in both directions happen "quasistatically," energy consumption can be reduced asymptotically to zero.

For a computation to satisfy the quasistatic behavior, two conditions must be followed. First, charge flow between any two nodes must occur gradually. To satisfy this condition, no device in charge-recovery circuits should turn on while there is a potential difference across it. The second condition is the elimination of nonlinear dissipative elements, such as diodes, since having a potential difference larger than the threshold voltage to generate current flow through the diode would violate quasistatic behavior.

Later work in charge-recovery logic deviates from fully reversible circuits and focuses on implementing partially reversible circuits. As pointed out in [24], the large number of temporary storage elements in a fully reversible circuit yields large circuit overheads. Since CMOS devices are not ideal switches and have leakage current, it follows that there is a tradeoff between throwing information away to keep temporary storage low and keeping information around to reduce change in entropy. Using a bit-pipeline adder as an example, the authors conclude that a partially reversible implementation has smaller circuit overhead and higher energy efficiency over a fully reversible implementation.

One of the first partially reversible logic families is the Adiabatic Dynamic Logic (ADL) proposed by Dickinson and Denker [28, 29]. Fig. 2.8 shows the structure of

NMOS and PMOS inverters in an ADL implementation. In the case of an NMOS ADL inverter, a diode precharges the output node *out* high as the power-clock rises, and the evaluation stack conditionally discharges the node out as the power clock falls. The PMOS ADL inverter works in an opposite fashion, precharging low and evaluating high. In an ADL system, ADL gates are cascaded by alternating the NMOS and PMOS ADL gates, and NMOS and PMOS gates are synchronized by a two-phase resonant clock with 180 degree phase difference to ensure that the precharge and the evaluation phase of all gates are in synch. The main advantage of ADL is the lower circuit overhead compared to fully reversible logic. However, due to its single-rail structure, it exhibits a data-dependent clock load, yielding high clock jitter.

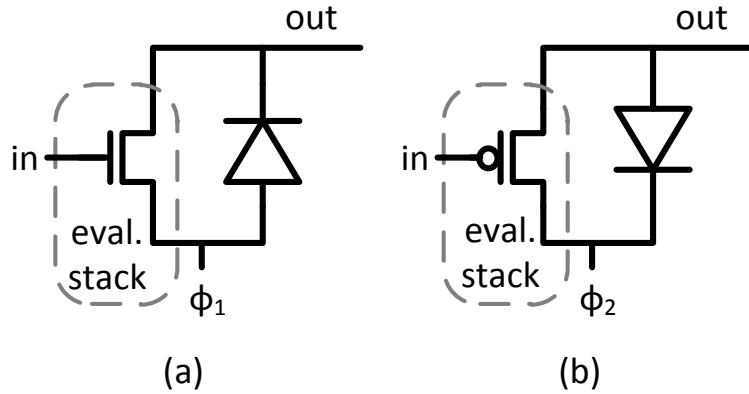
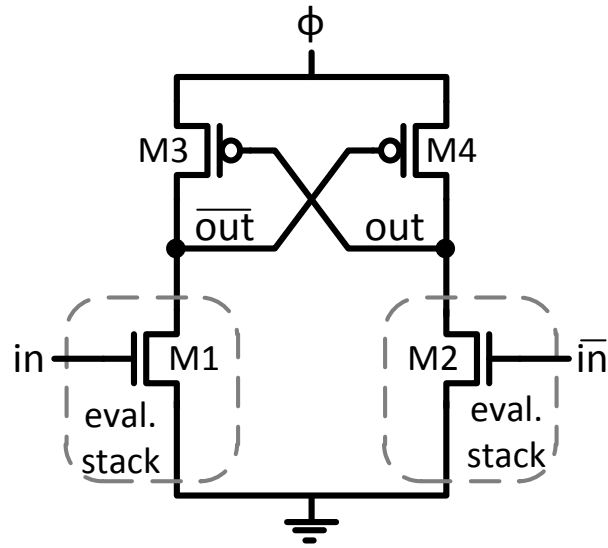
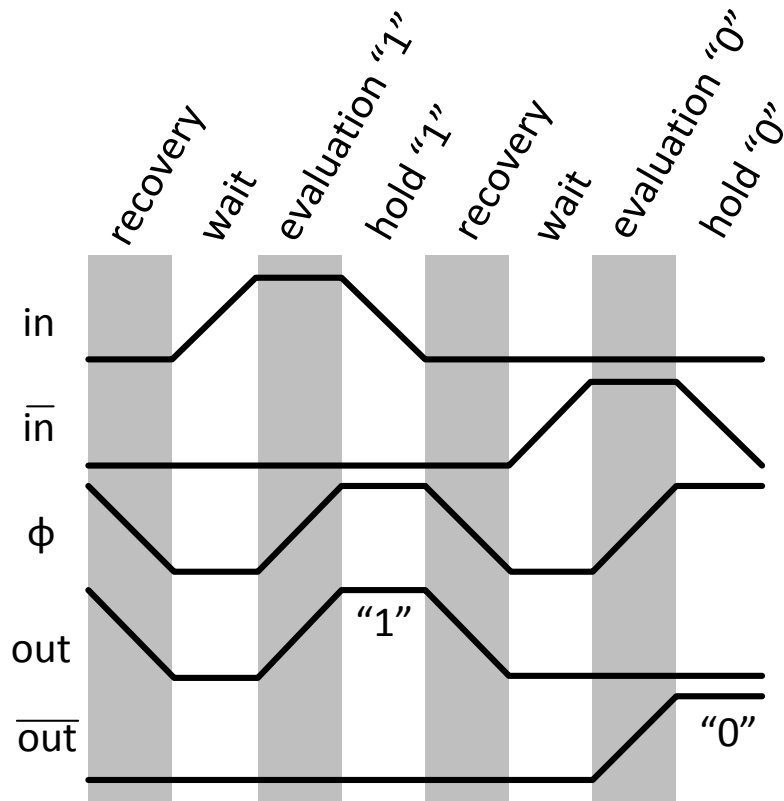


Figure 2.8: (a) Schematics of an NMOS ADL inverter, and (b) a PMOS ADL inverter.

To mitigate clock jitter due to data-dependent clock load, Kramer and Denker proposed the 2N-2P logic [30]. It is also known as Efficient Charge-Recovery Logic (ECRL) [31, 32], since Moon et al. independently proposed it at around the same time. Fig. 2.9(a) shows an inverter/buffer gate implemented in 2N-2P, which is a dual-rail logic synchronized by a four-phase power clock. It has a pair of cross-coupled PMOS devices on the top to steer current between the power clock and the two outputs, and a true and a complementary evaluation stack to perform the logic



(a)



(b)

Figure 2.9: (a) Schematic and (b) operating waveforms of a 2N-2P inverter.

operation. The per cycle energy consumed in this gate is governed by:

$$\begin{aligned}
 E_{2N-2P} &= E_{PMOS} + E_{Crowbar} \\
 &= \frac{RC}{T} CV_{DD}^2 + E_{Crowbar}.
 \end{aligned}
 \tag{2.1}$$

The energy consumption of the cross-coupled PMOS devices is similar to the derivation of the adiabatic switching principle found in Chapter 1, where resistance R is based on the size of the cross-coupled PMOS devices, and capacitance C depends on the size of the cross-coupled PMOS devices and output loads. This cross-coupled PMOS pair powered by the power-clock is a common structure among dual-rail charge-recovery logic, since it exhibits a relatively data-invariant load to the power clock.

The 2N-2P logic gate operates in four phases: evaluation, hold, recovery, and wait. Fig. 2.9(b) shows the operating waveforms of a 2N-2P gate. To simplify the explanation of 2N-2P operation, we use a trapezoidal waveform to model the sinusoidal power clock. During the evaluation phase, the cross-coupled PMOS devices steer current to the node that is left floating by the evaluation stacks, charging the node with a logical 1 to full rail. During the hold phase, the power clock remains high and each gate provides steady full rail outputs to its fanout gates. During the recovery phase, charge at outputs is recycled back to the power clock through the cross-coupled PMOS devices as the power clock falls. In the wait phase, both outputs and the power clock remain low, which is necessary for the fanout gates to perform their recovery phase operation. In a 2N-2P system, 2N-2P gates are cascaded by connecting them to one of the four clock phases, such that any two consecutive phases have 90 degrees of phase difference. Since the power-clock provides both power and synchronization to each gate, charge-recovery logic is naturally wave pipelined, meaning that in each phase, data propagate through only one gate.

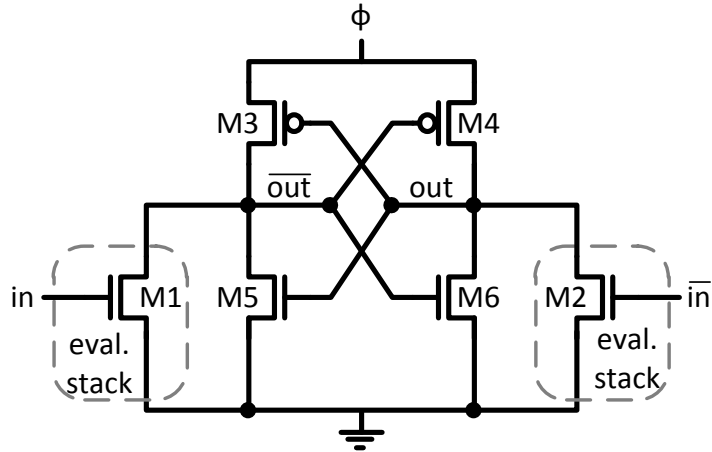


Figure 2.10: Schematic of a 2N-2N2P inverter.

Since the introduction of 2N-2P, various charge-recovery logic families have been proposed, including 2N-2N2P, PAL, PFAL, and CAL [30, 33–35]. Fig. 2.10 shows a schematic for 2N-2N2P [30], which is a variation of 2N-2P, also proposed by Kramer and Denker. By adding an additional pair of cross-coupled NMOS devices at the bottom of 2N-2P, 2N-2N2P eliminates floating outputs during the hold phase.

Pass-transistor Adiabatic Logic (PAL) [33], shown in Fig. 2.11, is a dual-rail charge-recovery logic with its evaluation stacks in parallel to the cross-coupled PMOS devices. Unlike 2N-2P, PAL operates with a two-phase resonant clock instead of a

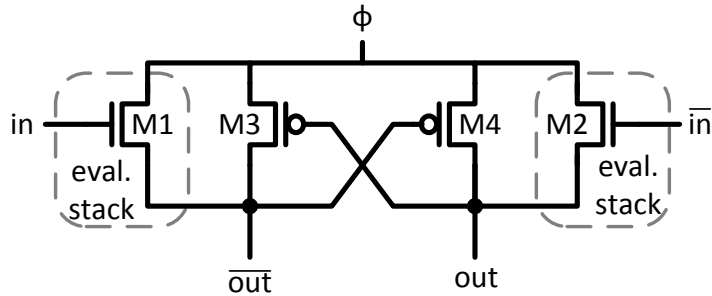


Figure 2.11: Schematic of a PAL inverter.

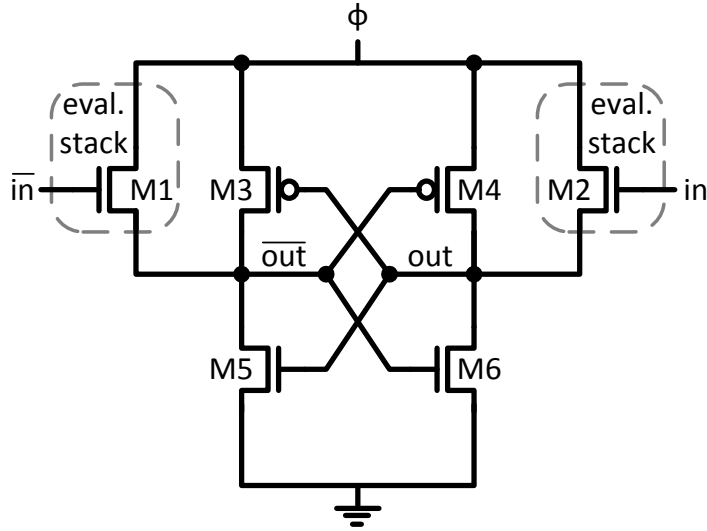


Figure 2.12: Schematic of a PFAL inverter.

four-phase one. A 1,600-stage PAL shift register has been fabricated in a $1.2\mu\text{m}$ technology, and correct operation has been verified at an operating frequency of 10MHz. Fig. 2.12 shows Positive Feedback Adiabatic Logic (PFAL) [34], a dual-rail adiabatic logic operating with a four-phase resonant clock. Compared to PAL, it has an additional pair of cross-coupled NMOS devices across its outputs. Since the inputs of a PFAL gate are at full rail during the evaluation phase, the effective resistance of its evaluation stacks is lower compared to cross-coupled PMOS devices on per- μm basis, yielding potentially higher energy efficiency over 2N-2P and 2N-2N2P.

Fig. 2.13 shows the Clocked CMOS Adiabatic Logic (CAL) [35], a dual-rail adiabatic logic operating with a single-phase resonant power-clock and two auxiliary clock phases. A test-chip with a chain of 736 CAL inverters was fabricated in a $1.2\mu\text{m}$ technology and verified at 50MHz using an externally supplied power-clock with a 5V clock amplitude [36].

To reduce the number of clock phases that a resonant system has to generate and distribute, Kim et al. have proposed three different charge-recovery logic families

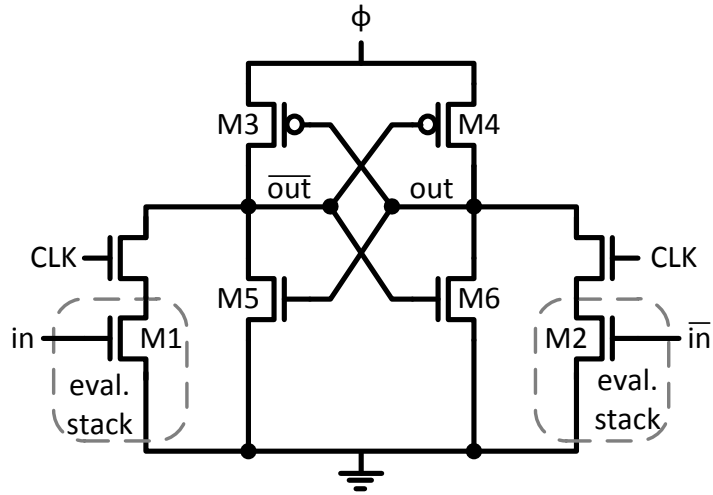


Figure 2.13: Schematic of a CAL inverter.

that operate with a single-phase resonant clock. Fig. 2.14 shows NMOS and PMOS inverter/buffer gates for True Single-phase Energy-recovery Logic (TSEL) [37], which have similar structure to CAL. Multiple TSEL gates are cascaded by alternating

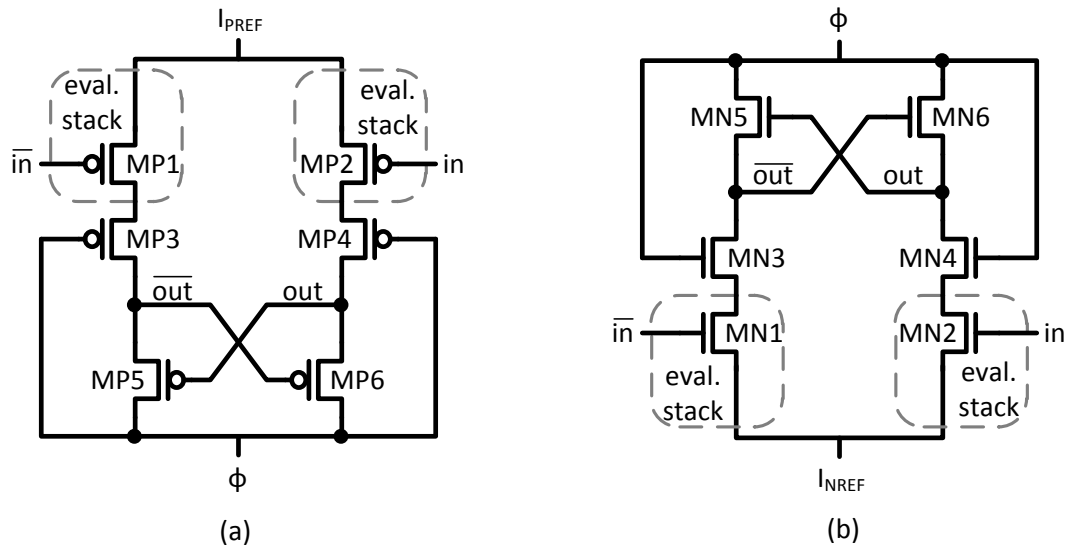


Figure 2.14: (a) Schematic of a PMOS TSEL inverter and (b) an NMOS TSEL inverter.

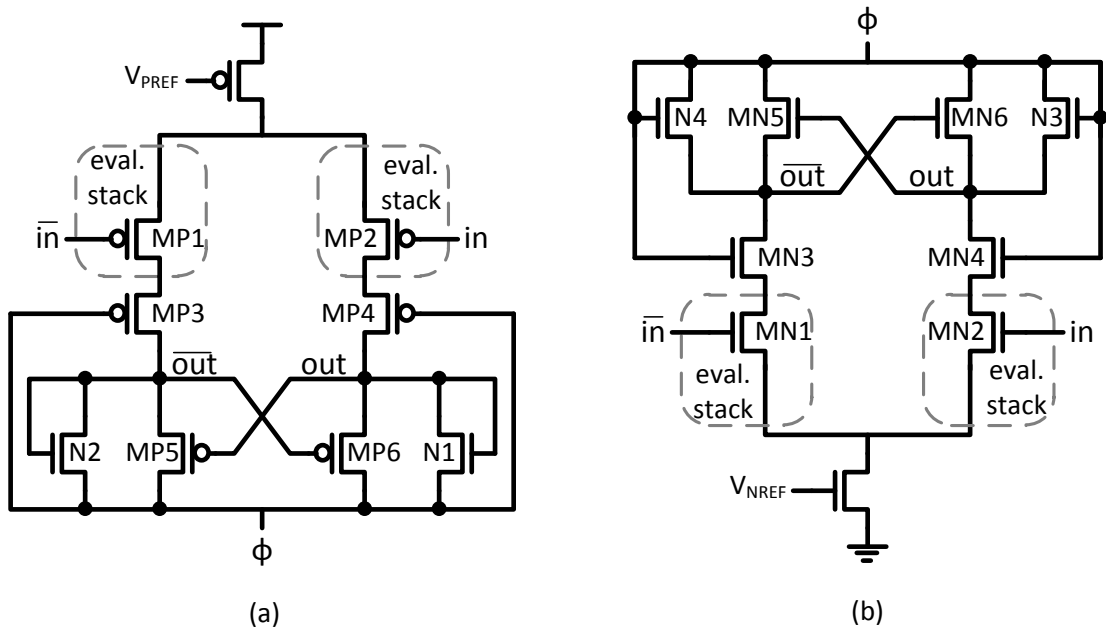


Figure 2.16: (a) Schematic of a PMOS SCAL-D inverter and (b) an NMOS SCAL-D inverter.

higher operating frequencies, they take excess amount of time to resolve differential outputs at the on-set of the power-clock. Fig. 2.17 shows the operating waveforms of

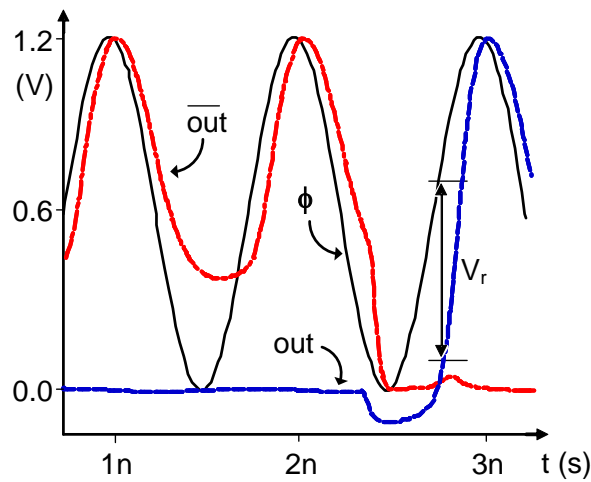


Figure 2.17: Simulated operating waveforms of a 2N-2P inverter at 1GHz.

a 2N-2P inverter at 1GHz. The voltage difference between the power-clock and the outputs of the 2N-2P gate can be as large as 0.6V. This voltage difference indicates a large IR drop across the power steering transistors, yielding poor efficiency.

To reach GHz-level operating frequencies, Sathe et al. proposed Boost Logic [43, 44], which utilizes aggressive voltage scaling, gate overdrive, and charge-recovery techniques to achieve high performance and energy efficient operation. Fig. 2.18 shows a schematic of an inverter/buffer implemented in Boost Logic. The structure of Boost Logic can be divided into two sections: Boost stage in the center and Logic stage on the two sides. Boost Logic is a dual-rail charge-recovery logic that is synchronized by a two-phase power clock. It operates in two phases: evaluation and boost. During the evaluation phase, the Logic stage develops a near-threshold voltage difference across the differential outputs, out and out_b . Due to this near-threshold voltage difference, the Boost stage quickly amplifies the differential outputs such that they follow the power-clock closely, yielding energy efficient operation. Since the Logic stage is powered by V_{DD}' and V_{SS}' , which are set to be $2/3V_{DD}$ and $1/3V_{DD}$,

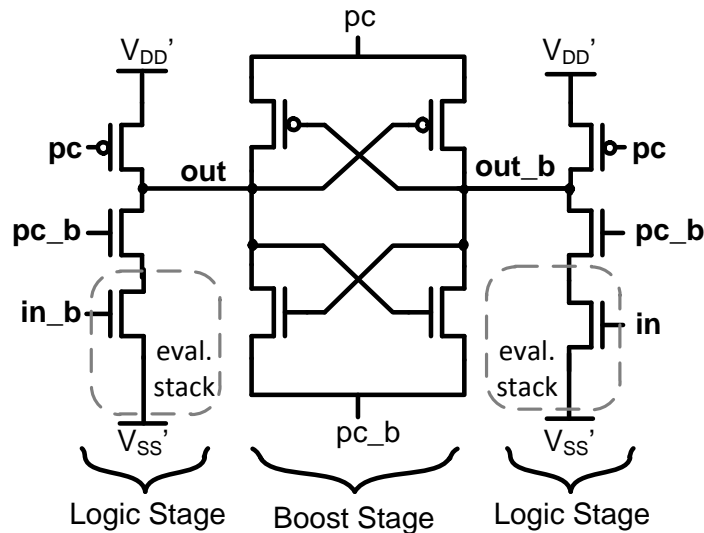


Figure 2.18: Schematic of an inverter implemented in Boost Logic.

respectively, the power dissipated in the Logic stage is extremely energy efficient due to the aggressive voltage scaling. Moreover, since inputs of Boost Logic are ramped to full rail, transistors in the logic stage are operated in the super-linear region, which results in high performance. To demonstrate the high performance and energy efficiency of Boost Logic, a test-chip with eight chains of AND, OR, XOR, and INV gates was fabricated in $0.13\mu\text{m}$ technology and correct operation was verified at operating frequencies exceeding 1GHz [7, 45].

Looking back at the evolution of existing work, many similar traits are shared among various dual-rail charge-recovery logic topologies. The first common trait is the pair of cross-coupled PMOS devices used to steer the current of the power-clock. The second common trait is that the differential outputs of a charge-recovery gate are always reset to the same voltage at the end of each cycle. The third common trait is the wave pipelining property. These common traits are likely to be carried on by future charge-recovery logic.

2.3 Resonant-Clocked Designs

Resonant clocking is an energy-recovery design methodology that focuses on recovering energy from the clock network. In resonant clocking, an inductor L and the parasitic capacitance of the clock network C form an LC oscillation that generates clocks of generally sinusoidal waveforms. Instead of dumping charge from the clock network to ground, resonant clock designs recycle energy sent to the clock network and store it in the form of magnetic energy in the inductor, potentially yielding significantly higher energy efficiency over conventional-clock designs. Since clock distribution networks have high switching activities and large capacitance, the power reduction from resonant clocking amounts to a significant portion of total power. Another advantage of resonant clocking over other alternative clocking techniques, such as the rotary clock [46, 47] and the standing wave clock [48–50], is that it distributes

a clock waveform with uniform amplitude and phase across the whole network.

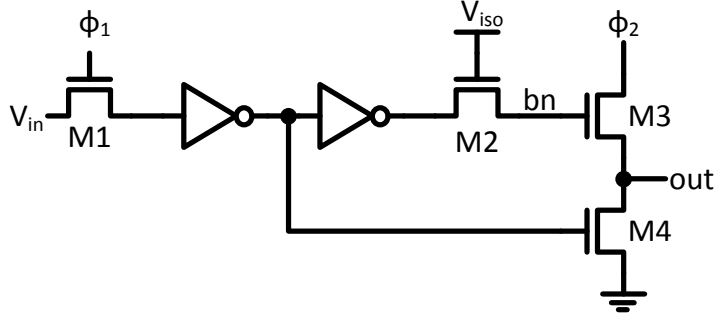


Figure 2.19: Schematic of E-R (Edge Triggered) Latch.

Early resonant-clocked designs have been inspired by charge-recovery logic. Athas et al. [51, 52] have implemented the AC-1 microprocessor using a so-called E-R latch, which is shown in Fig. 2.19. Similar to the hot-clock nMOS design [53], the E-R latch uses the booststrapping technique to improve the energy efficiency of the final driver. This latch captures input V_{in} when ϕ_1 is high, and drives output out when ϕ_2 is high. During its operation, charge is recycled from both the clock network and the outputs of the E-R latches. Since it uses NMOS pass gate structures for both M1 and M2, clock swings are set to be one threshold voltage above V_{DD} , thus eliminating the threshold voltage drop across pass transistors when transmitting a 1. Due to the gate-to-channel capacitance of transistor M3, the rise of ϕ_2 booststraps node bn to rise above V_{DD} . This voltage boost effectively reduces the resistance of transistor M3, making the E-R latch an energy efficient driver. Notice that the E-R latch exhibits a data-dependent load to the resonant system due to its use of single-end adiabatic switching, yielding increased clock jitter.

Fig. 2.20 shows the PMOS energy recovering flip-flop (pTERF) proposed by Ziesler et al. [11, 54]. This flip-flop has an energy-recovery dynamic buffer driving a pair of cross-coupled NOR gates. An inverter on the input side creates a locally-negated version of input D , and an inverter is added at output Q to increase drive strength.

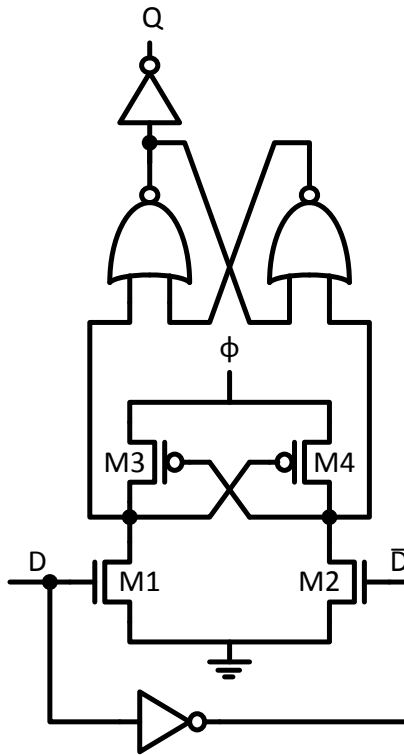


Figure 2.20: Schematic of pTERF flip-flop.

Similar to dual-rail charge-recovery logic, the cross-coupled PMOS devices efficiently steer current between the power-clock ϕ and internal nodes xf and xt using a single-phase resonant clock. A dual-mode-clocking wavelet-transform test-chip has been fabricated to compare the energy savings between a resonant-clocked pTERF design and a conventional-clocked design with master-slave flops. In the resonant clocked mode, it dissipates from 75% to 83% of the conventional clocked mode in the 115MHz to 225MHz range. However, resistance of the cross-coupled PMOS devices in pTERF degrades the quality factor of the resonating system, limiting overall potential energy savings.

More recent work has focused on resonating the parasitic capacitance of non-adiabatic flip-flops and the clock distribution network. Drake et al. [55, 56] deployed a two-phase resonant clock across three 512-bit scan-chains with master-slave flip-

flops. To maximize the capacitance of the resonant system, internal clock inverters inside the flip-flops were removed, and the two-phase resonant clock was used to drive transmission gates inside flops directly. Since resonant clocks have more gradual transition times, clocking master-slave flip-flops using sinusoidal waveforms resulted in significant performance degradation in setup times and clock-to-Q times.

One method to mitigate performance degradation of resonant-clocked flops is by inserting clock buffers to improve clock slew. Hansson et al. [57] have fabricated a 1.56GHz resonant-clock network in a $0.13\mu\text{m}$ technology with an integrated 1.2nH inductor. A single-phase resonant clock is distributed to 896 master-slave flip flops with up to 10 inverters between them, while the second phase is generated locally by an inverter. Measured with 30% switching activity at the inputs, the test-chip shows 20% reduction in total power. The introduction of local static inverters improves flop performance at the expense of increasing clock buffer crowbar current.

Another approach to mitigating performance degradation in resonant-clocked flops is by designing special flops that cope with more gradual transition times. Mahmoodi

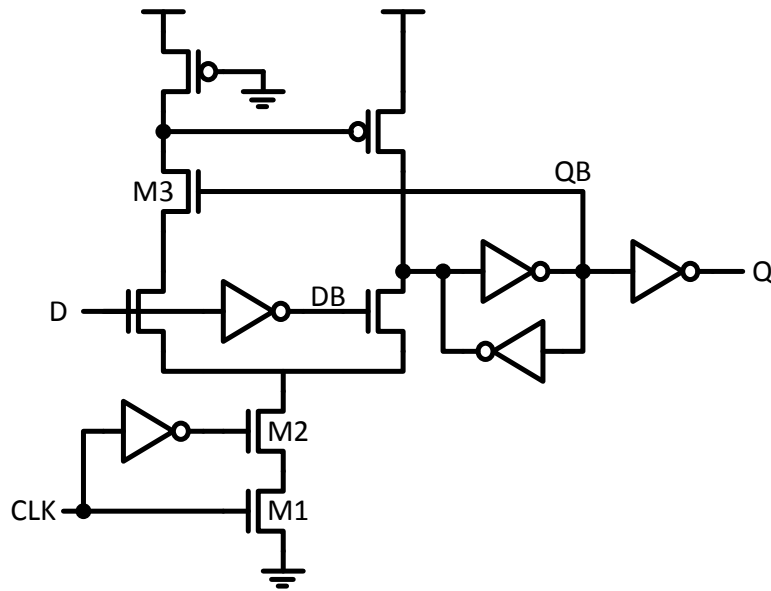


Figure 2.21: Schematic of SCCER flip-flop.

et al. [58, 59] propose the single-ended conditional-capturing energy-recovery (SC-CER) flip-flop, as shown in Fig. 2.21. Connected to different phases of a resonant clock, transistors M1 and M2 create a narrow window for evaluation, reducing crowbar current even when transition times of the resonant clock are large. Transistor M3 controlled by signal QB conditionally captures input D to further reduce flop power. The energy efficiency of SCCER-based design has been demonstrated using a 64-bit 8-cycle multiplier that deploys a single-phase resonant clock across its entire clock network with operating frequency up to 160MHz.

Ishii et al. [12] deployed resonant clocking in conjunction with sense-amplifier flip-flops, shown in Fig. 2.22, successfully demonstrating energy efficient operation of an ARM926EJ-S microprocessor. Once the sense-amplifier stage completes evaluation, one of transistors M4 and M5 shuts off to retain the correct value in the flop while reducing crowbar current. Fabricated in a 0.13um technology, test-chips from a typical corner lot have been verified with operating frequency up to 200MHz and achieve total power savings in the range of 20% to 35% depending on benchmarks.

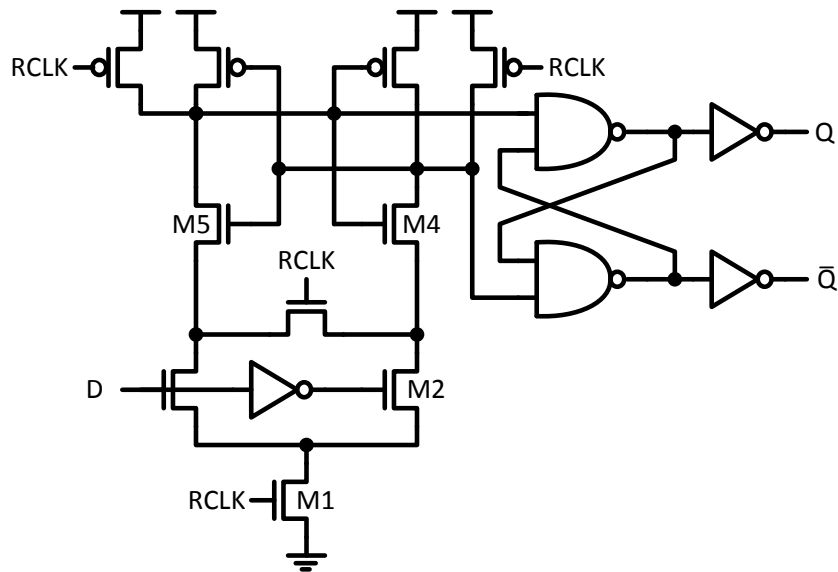


Figure 2.22: Sense-amplifier flip-flop used in resonant-clock ARM926EJ-S.

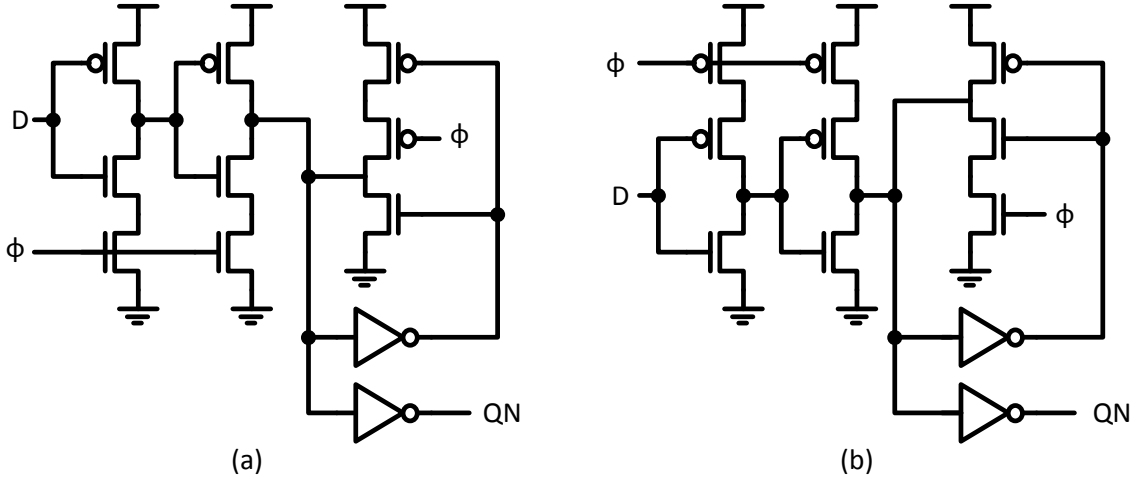


Figure 2.23: Schematic of level sensitive latch used in RF1: (a) H-LAT and (b) L-LAT.

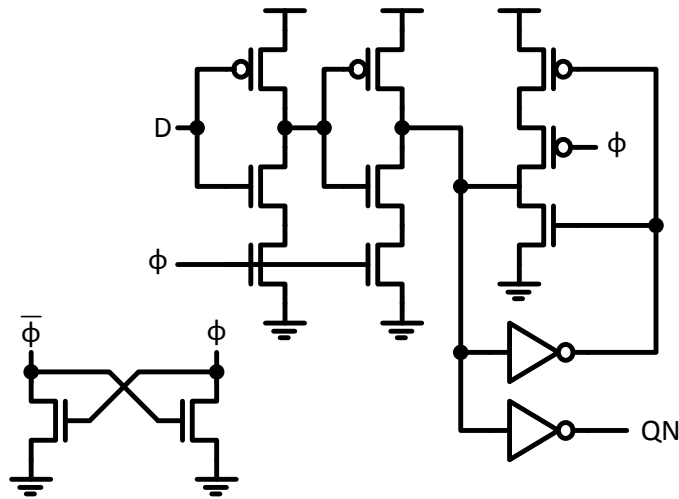


Figure 2.24: Schematic of level sensitive latch used in RF2, B-LAT.

More recent implementations of resonant-clock designs focus on improving performance of timing elements by applying resonant clocking on latch-based designs [14]. Since performance of latches depends on the voltage level of clock waveforms instead of the sharpness of clock edges, latch-based designs synchronized by resonant clocks have lower D-to-Q times than flop-based designs. Two 14-tap 8-bit finite impulse response (FIR) filter test-chips have been fabricated in a 0.13 μm technology with two

different types of latches to demonstrate the high performance and energy efficient operation of resonant clock latch-based designs. A resonant-clock FIR test-chip, called RF1 [60], uses a single phase resonant clock to synchronize 5,500 latches. Fig. 2.23 shows two latches used in RF1: H-LAT is transparent when clock is high, and L-LAT is transparent when clock is low. The single-phase resonant clock is generated by a single ended resonant clock generator with a programmable pulse width and switch size, enabling RF1 to operate efficiently across a range of operating frequencies. At its resonant frequency of 1.03GHz, RF1 dissipates 132mW, recovering 76% of clock energy compared to a conventional-clock design with an identical conventionally-clock load. Another resonant-clock FIR test-chip, called RF2 [13], has used a two-phase resonant clock to synchronize an architecture that is identical to RF1. Fig. 2.24 shows the B-LAT latch used in RF2, which has similar structure to H-LAT with additional cross-coupled NMOS devices to provide negative g_m . The blip clock generator in RF2 generates a two-phase resonant clock with a clock switch integrated into each latch. At its resonant frequency of 1.01GHz, RF2 dissipates 122mW, achieving 84% clock efficiency.

Other work in this area has focused on evaluating and characterizing variability on resonant clocks. Chan et al. [61–64] have deployed resonant clocking at the global clock distribution level to reduce clock-related power and clock skew/jitter. Two test-chips have been fabricated in 0.18 μ m and 90nm technologies with operating frequencies of 1.6GHz and 4.6GHz, respectively. Using four 6.4nH on-chip inductors with a Q of 6.8, the 0.18 μ m test-chip distributes a two-phase resonant clock across a 2mm x 2mm area and achieves 87% power savings over a non-resonant clocked design while reducing the peak on-chip jitter by 56% with 300mV of added noise on power supplies. Therefore, oscillating with a high quality factor inductor, a resonant clocked design has the potential to achieve lower skew and jitter over a conventional clocked design.

Chueh et al. [65–68] have implemented a two-phase resonant clock network over a 2mm x 2mm area using an H-tree distribution with programmable loads to evaluate the effect of imbalanced clock load on clock skew. With four on-chip inductors and programmable clock load, the test-chip operates between 900MHz and 1.2GHz, and it is 1.56X more energy efficient than [63]. When the entire flop-related capacitance is moved to one side of the H-tree distribution, measurement results show that the clock skew due to imbalanced clock loads is less than 6% of cycle time.

In 2009, Xu et al. [69, 70] implemented actively deskewed resonant clock networks to further reduce skew in clock distributions due to cross-chip variation and transient loading changes. An all-digital deskewed-control circuit around the distributed differential oscillator in each resonant clock domain locks its oscillating frequency to its neighbor’s. On-chip jitter and skew measurement circuits show that deskewed-control circuits can reduce the cross-chip skew from 20ps to 3ps across four clock domains over a 3mm x 3mm area with sixteen 3nH inductors.

Chan et al. [71, 72] later applied their global resonant clock distribution topology to a commercial microprocessor, which resonates 830 clock domains with 830 on-chip inductors at its natural frequency of 3.2GHz. Despite the reduction in power associated with global clock distribution, the potential power savings that can be achieved with this technique are limited at the chip level, since the majority of the clock capacitance is near the distribution of the local clock.

CHAPTER 3

Enhanced Boost Logic

This chapter, we present Enhanced Boost Logic (EBL), a novel charge-recovery logic family [8]. EBL achieves significantly higher energy efficiency over previous charge-recovery logic. In addition, by having the ability to compute complex logic function in each phase, designs implemented using EBL achieve low-latency overhead compared to static CMOS designs.

3.1 Introduction

Charge-recovery circuitry has the potential to reduce dynamic power consumption in digital systems with significant switching activity. To keep energy consumption to a minimum, charge-recovery circuitry is typically designed so that it maintains low voltage drops across device channels, while recovering the charge supplied to it every clock cycle. The overall energy-efficiency of charge-recovery circuitry therefore depends on the rate at which transitions occur, yielding an inverse relationship between energy consumption and clock period [73]. Relying on this energy/latency trade-off, charge-recovery circuitry can operate with energy consumption below CV^2 , the fundamental limit of static CMOS.

Early research on charge-recovery logic design focused on micropipelined dynamic circuits with multiple (four or more) clock phases for recovering charge [30, 32, 42].

These clock phases were generated by resonating the parasitic capacitance C of the circuitry through the introduction of inductors. To maximize the efficiency of recovery, the inductors were chosen so that the resulting LC tank system resonates at the target clock frequency. In these early multi-phase designs, the resulting complexity of the recovery mechanisms was considerable, especially in the case of the so-called reversible designs [74], which theoretically offer the greatest energy saving potential. Moreover, the synchronization of multiple clock phases was impeding high-speed operation.

Aimed at reducing control overheads and increasing operating speeds, several single-phase and two-phase charge-recovery families were proposed [33, 37, 38, 75]. Such micropipelined logic did achieve clock frequencies comparable with static CMOS [40, 41], but it also resulted in increased latencies, due to the reduction in the number of clock phases and, therefore, in the number of logic functions performed each clock cycle. It thus made the energy/latency trade-off of charge-recovery circuitry more manifest at the architectural level.

In recent years, a charge-recovery family that uses multiple power supply levels, called Boost Logic, was demonstrated in silicon at clock speeds exceeding 1GHz [7, 45]. Although micropipelined using a two-phase clocking scheme, Boost Logic improves upon the energy/latency trade-off of previous charge-recovery circuit families, as it relies on gate overdrive to evaluate logic functions with significantly decreased delay and with minimal short-circuit current. It thus has the potential to achieve high-speed and low-power operation with pipeline latencies that are comparable to those of static CMOS designs.

Enhanced Boost Logic (EBL), presented in this chapter, is an improved version of the basic Boost Logic that achieves shorter pipeline latencies while retaining its energy advantages over static CMOS. Similar to Boost Logic, EBL is capable of operation at high clock frequencies by developing a near-threshold voltage before the

onset of the power-clock. Evaluation devices in EBL have twice the gate overdrive compared to first-generation Boost Logic [7, 45], however, enabling the design of complex logic gates and thus decreasing total gate counts. Consequently, EBL further improves upon the energy/latency tradeoff of Boost Logic, yielding lower latency while maintaining good energy efficiency. EBL improves upon Boost Logic also with respect to implementation complexity, as it requires a smaller number of power supplies.

The remainder of this chapter is organized as follows: We discuss EBL structure and operations in Section 3.2 and Section 3.3, respectively. We also highlight improvements made in EBL to achieve more energy-efficient operation over previous charge-recovery logic. In Section 3.4, we discuss the clock generation circuitry for EBL. In Section 3.5, we present the equation that governs energy consumption of EBL. Conclusions are given in Section 3.6.

3.2 EBL Structure

The origins of EBL can be traced back to Boost Logic, shown in Fig. 3.1(a). GHz-level operation has been demonstrated in silicon on a chain of simple Boost

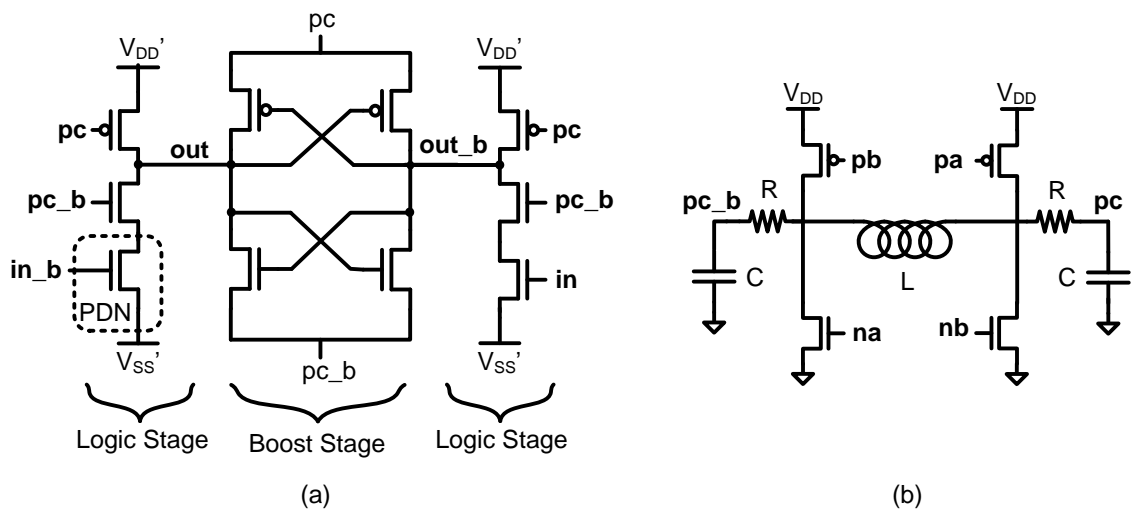


Figure 3.1: Boost Logic schematic.

Logic gates powered by a two-phase clock [7, 45]. The original Boost Logic design uses four supply levels: V_{DD} , V_{DD}' , V_{SS}' , and ground, where V_{DD}' and V_{SS}' are set at approximately $\frac{2}{3}V_{DD}$ and $\frac{1}{3}V_{DD}$, respectively. Powered by the aggressively-scaled voltage $V_{DD}' - V_{SS}'$, the Logic stage drives the dual-rail outputs conventionally with subthreshold-level energy consumption in the first half of each clock cycle. Subsequently, during the second half of each cycle, the Boost stage amplifies the near-threshold voltage between the two outputs to full rail using the two complementary clock phases **pc** and **pc_b**. These clock phases are generated using an H-bridge topology, as shown in Fig. 3.1(b). When Boost Logic gates are cascaded, the full-rail output from the Boost stage of one gate drives the Logic stage of the next gate, yielding operation in the super-linear region.

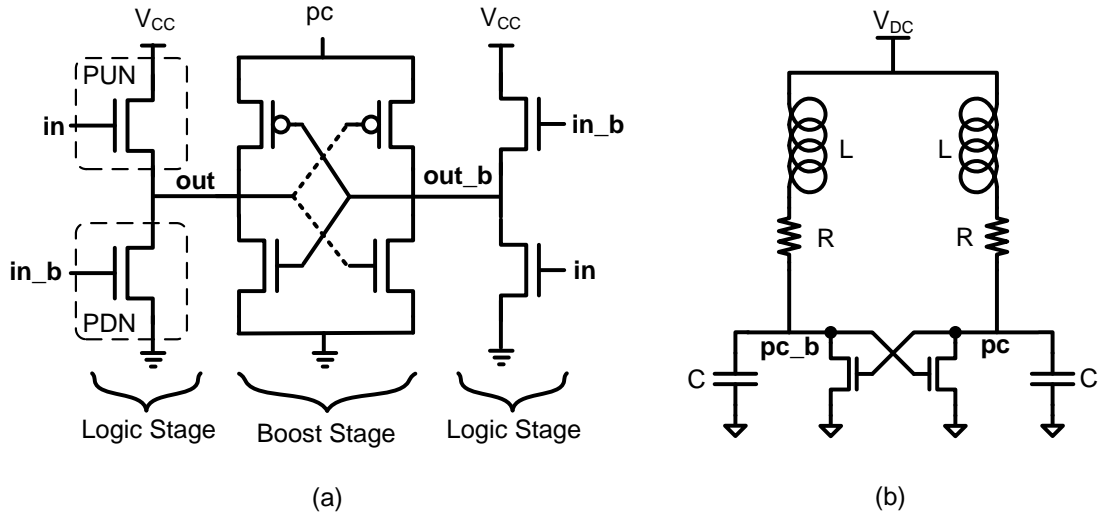


Figure 3.2: SBL schematic.

Fig. 3.2(a) shows Subthreshold Boost Logic (SBL), a variant of Boost Logic that is targeted at slower clock rates than Boost logic. The energy-efficient and multi-MHz operation of SBL with a single subthreshold supply has been demonstrated in silicon [76, 77]. Similar to Boost Logic, SBL uses aggressive voltage scaling, using a subthreshold supply V_{CC} to power the dual-rail Logic stage. Unlike Boost Logic,

however, the Logic stage has no clocked devices, and each of its two output rails is evaluated by a complementary all-NMOS stack. Another departure from Boost Logic is that the same subthreshold supply V_{CC} is used to power a "blip" clock generator, as shown in Fig. 3.2(b), producing two partially-overlapping clock waveforms **pc** and **pc_b** with peak values significantly greater than V_{CC} . The Boost stage of each gate amplifies its output voltage to the full amplitude of the corresponding clock **pc** and drives the all-NMOS Logic stage in the next SBL gate, yielding increased gate overdrive over Boost Logic. Compared to Boost Logic, SBL simplifies the number of supplies and powers each Boost stage with a single clock, yielding considerable reduction in crowbar current.

The Enhanced Boost Logic presented in this paper is another variant of Boost Logic that is aimed at pushing the iso-energy frequency point higher than SBL, while at the same time decreasing latency overhead. Fig. 3.3(a) shows a cascade of three EBL buffers. Each EBL gate has two stages: Evaluation and Boost. Similar to SBL, the Boost stage consists of a cross-coupled inverter with the source of the PMOS connected to a charge-recovering clock phase **pc**, enabling high performance through enhanced gate overdrive. Unlike SBL, however, the Evaluation stage relies on a NMOS precharge device for pull-up, instead of a complementary pull-up network, thus increasing performance by avoiding the series-connected devices in the pull-down network (PDN). The bulk of all NMOS transistors are connected to ground, and the bulk of PMOS transistors in the cross-coupled inverters are connected to the corresponding power-clock phases. From a functional point of view, each EBL gate is equivalent to a combinational logic block (Evaluation stage) that is powered by a near-threshold supply V_{CC} and drives a transparent latch synchronized by clock phase **pc** (Boost stage). Cascades of EBL gates are clocked by alternating clock phases **pc** and **pc_b**.

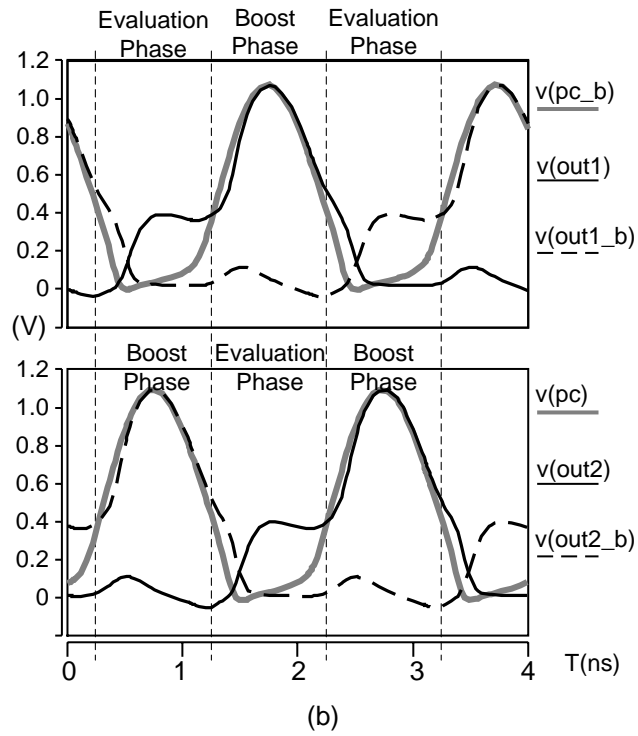
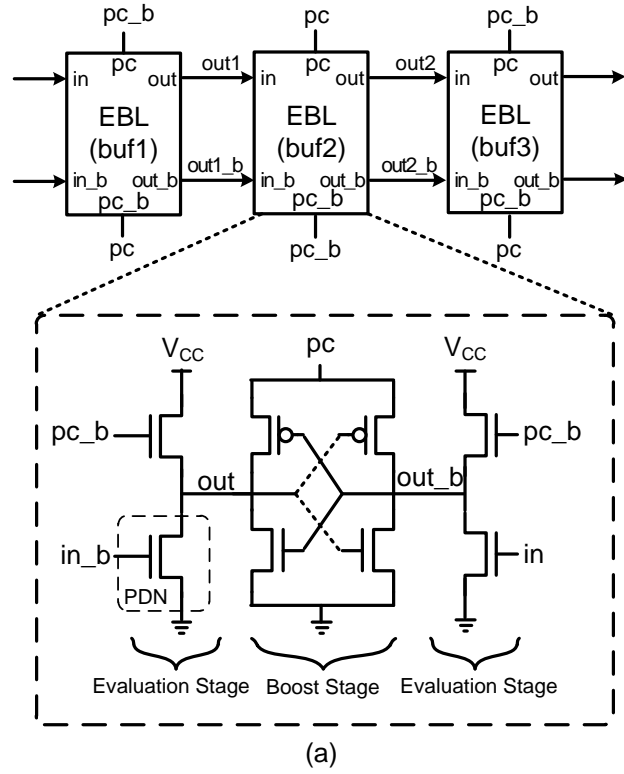


Figure 3.3: EBL buffer schematic and operation.

3.3 EBL Operation

Each EBL gate operates in two phases: Evaluation, and Boost. Fig. 3.3(b) shows the operating waveforms of EBL buffers **buf1** and **buf2** in the three-buffer cascade of Fig. 3.3(a). During the Evaluation phase of **buf2**, the clock phase **pc** ramps-up to full V_{DD} , then back to ground, while the other clock phase **pc_b** stays well below threshold voltage. As the inputs of **buf2** ramp up with clock phase **pc**, the Evaluation stage charges node **out2** toward the subthreshold supply level V_{CC} , and discharges node **out2_b** towards ground. Notice that even though the Evaluation stage is powered by a near-threshold supply, its PDN operates in super-linear mode, since its inputs are ramped to full V_{DD} . Compared to Boost Logic, EBL achieves a gate overdrive of 0.8V, yielding 2X improvement in gate overdrive. Since the Evaluation stage inputs follow clock phase **pc** to full V_{DD} , the performance of the NMOS precharge device is relatively immune to the V_{th} drop thanks to the increased gate overdrive. As inputs ramp down toward the threshold voltage level, following clock phase **pc**, the Evaluation stage is turned off. Throughout the Evaluation phase, the Boost stage is effectively shut off, since the clock phase **pc_b** is well below the threshold voltage.

During the first half of the Boost phase in the operation of **buf2**, the Boost stage amplifies the near-threshold voltage difference at **out2** and **out2_b** to full rail as clock phase **pc_b** rises to full rail. This full-rail signal is used to drive the Logic stage of **buf3**, yielding enhanced gate overdrive. During the second half of the Boost phase for **buf2**, the power-clock **pc_b** returns back to ground, recovering the charge at the output nodes of **buf2** until it reaches the near-threshold supply level V_{CC} .

3.4 EBL Clock Generation

The two clock waveforms required for EBL operation are generated using a clock generator similar to the "blip" circuit shown in Fig. 3.4 [6]. This circuit consists of

two cross-coupled RLC oscillators, using the output waveform **pc** of one oscillator to drive the NMOS switch in the other oscillator and provide negative transconductance gm , and vice versa. The frequency of the oscillation is given by the equation:

$$f = \frac{1}{2\pi} \sqrt{\frac{1}{LC} - \zeta^2}, \quad (3.1)$$

where L denotes total inductance, C denotes the capacitance of the clock distribution and output nodes, and ζ denotes the damping factor. The amplitude of the clock waveforms is determined by the clock generator supply V_{DC} . Since the damping factor varies with the amplitude of the clock driving the negative-transconductance switches, the resonant frequency has a slightly inverse relationship with the supply V_{DC} . When the clock oscillates at the full nominal V_{DD} of 1.2V, the overlap of the two clock phases occurs below the threshold voltage of the regular V_{th} NMOS device. Thus, unlike Boost Logic, EBL does not need a clocked device in the PDN of its Evaluation stage to limit short-circuit current.

In our test-chip, the inductive element has been implemented as a fully-integrated

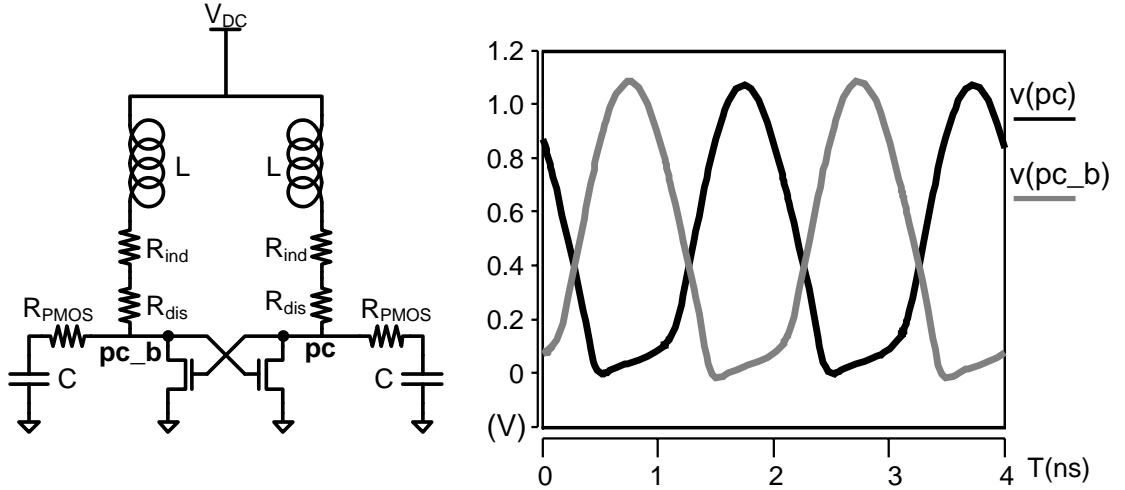


Figure 3.4: Blip clock generator and its two-phase power-clock waveforms.

center-tapped symmetrical spiral inductor, due to the relatively high target clock frequency. Moreover, the two switches of the clock generator have been implemented as a collection of smaller switches that are distributed across the clock network. A small centralized switch is also used to enable frequency-scaled operation using current injection locking. A more detailed description of our clock generator design is given in Section 4.1.

EBL improves upon Boost Logic in three ways. First, the use of a single near-threshold DC supply in the Evaluation stage reduces the number of power supplies required and doubles the gate overdrive. Second, the PDN of EBL gates enables the implementation of more complex functions. Specifically, by relying on a "blip" clock generator with two almost-non-overlapping phases, EBL eliminates the need for a clock-gated device in its PDN. Moreover, the 2X gate overdrive allows more complex functions to develop the near-threshold difference between the dual-rail outputs by the end of the Evaluation phase. Therefore, the maximum pull-down stack height of an EBL gate can be higher than in Boost Logic. (In 1GHz simulations, it can be seven NMOS devices high.) Third, the Boost stage requires a single clock phase, thus reducing the area overhead over Boost Logic by allowing minimal-sized NMOS devices. It also decreases power consumption compared to Boost Logic by reducing crowbar paths from V_{CC} to \mathbf{pc} . Due to the single precharge NMOS device, EBL has lower area overhead over SBL, and can drive its outputs faster than SBL.

3.5 EBL Energy Consumption

Per-cycle energy consumption of an EBL gate is given by the equation

$$E_{EBL} = E_{Evaluation} + E_{Boost} + E_{Crowbar}, \quad (3.2)$$

where $E_{Evaluation}$ and E_{Boost} denote the energy consumed in the two stages of EBL, and $E_{Crowbar}$ denotes the energy consumed by crowbar current during the EBL operation. The energy consumption of the Evaluation stage is given by the following equation:

$$E_{Evaluation} = \frac{1}{2}\alpha C_L V_{CC}^2, \quad (3.3)$$

where α denotes the switching activity of the Logic stage, C_L denotes the total switching capacitance at the EBL outputs, and V_{CC} is the near-threshold supply of the Evaluation stage.

To derive an expression for E_{Boost} , blip power-clock waveforms are modeled using a piecewise model, as shown in Fig. 3.5. A sinusoid with amplitude greater than V_{DD} and a slightly negative offset is used to model the pulse region of the blip clock waveform, while a linear model is used to describe the power-clock waveform when it is closed to ground. The clock waveforms in the two regions can be approximated as follows:

$$\phi_{sine} = 0.25V_a(1 + 3\sin(\omega t)) \quad (3.4)$$

$$\phi_{linear} = \frac{0.1t}{T - |t_{P1} - t_{P2}|}, \quad (3.5)$$

where $\omega = 2\pi/T$, T is the period of the blip power-clock, and t_{P1} and t_{P2} are the endpoints of the two regions, as shown in Fig. 3.5. Solving Equation 3.4 for 0.1V and 0V yields the following equation for endpoints t_{P1} and t_{P2} , respectively, of the two regions:

$$t_{P1} = \frac{\sin^{-1}\left[\frac{0.4V_a-1}{3}\right]}{\omega} \quad (3.6)$$

$$t_{P2} = \frac{\sin^{-1}\left[\frac{-1}{3}\right]}{\omega}. \quad (3.7)$$

The energy E_{sine} consumed by the Boost stage during the pulse part of the blip

power-clock is obtained by integrating I^2R over time from t_{P1} to t_{P2} , where the current I is govern by the following equation:

$$I(t) = \frac{V_{sine}(t)}{Z_L} = \frac{0.75V_a \sin(\omega t)}{\frac{1}{j\omega C_L}}, \quad (3.8)$$

and, therefore,

$$\begin{aligned} E_{sine} &= \int_{t_{P1}}^{t_{P2}} I(t)^2 R dt = \int_{t_{P1}}^{t_{P2}} \left(\frac{0.75V_a \sin(\omega t)}{\frac{1}{j\omega C_L}} \right)^2 R dt \\ &= \frac{9V_a^2 \omega^2 C_L^2 R}{32} (t - 2\cos(\omega t)) \Big|_{t_{P1}}^{t_{P2}} \\ &= \frac{9V_a^2 \pi^2 C_L^2 R}{8T^2} \left(t - \frac{1}{\omega} \cos(2\omega t) \right) \Big|_{t_{P1}}^{t_{P2}} \\ &= \frac{K \cdot 9V_a^2 \pi^2 C_L^2 R}{16T}, \end{aligned} \quad (3.9)$$

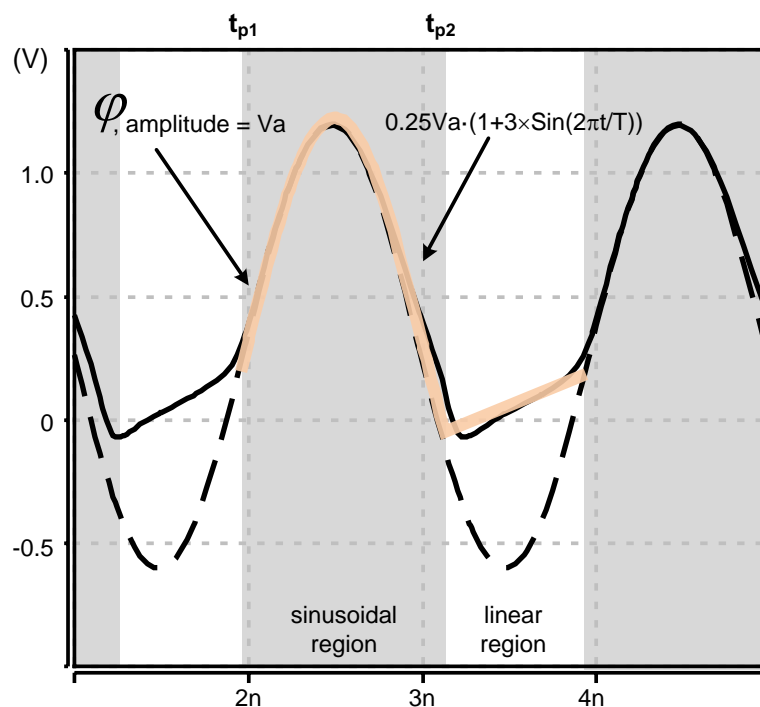


Figure 3.5: The piecewise model used to model the blip power-clock waveform.

where K denotes a constant coefficient between 0.5 and 0.6 depending on the clock amplitude, and R denotes the sum of the inductor resistance R_{ind} , the power-clock distribution resistance R_{dis} , and the resistance R_{PMOS} associated with the cross-coupled PMOS in the Boost stage. By replacing the clock amplitude V_a by the effective voltage swing in the Boost stage, $V_a - V_{CC}$, we obtain

$$E_{sine} = \frac{9K(V_a - V_{CC})^2 \pi^2 C_L^2 R}{16T}. \quad (3.10)$$

The energy E_{linear} consumed in the Boost stage of a EBL gate during the linear region of the clock waveform is given by the following equation:

$$\begin{aligned} E_{linear} &= \int_0^{T-|t_{P1}-t_{P2}|} I(t)^2 R dt \\ &= \int_0^{T-|t_{P1}-t_{P2}|} \left(C_L \frac{dV(t)}{dt} \right)^2 R dt \\ &= \int_0^{T-|t_{P1}-t_{P2}|} \left(C_L \frac{d\left(\frac{0.1t}{T-|t_{P1}-t_{P2}|}\right)}{dt} \right)^2 R dt \\ &= \frac{0.01 C_L^2 R}{T - |t_{P1} - t_{P2}|}. \end{aligned} \quad (3.11)$$

From Equation (3.10) and (3.11), the E_{linear} is less than 1% of E_{sine} . Therefore, E_{Boost} can be approximated by E_{sine} . Substituting Equation (3.3) and (3.10) in (3.2), we obtain the energy consumption of a EBL gate, which is given by

$$E_{EBL} = \frac{1}{2} \alpha C_L V_{CC}^2 + \frac{9K(V_a - V_{CC})^2 \pi^2 C_L^2 R}{16T} + E_{Crowbar}. \quad (3.12)$$

By assuming V_a to be $1.5V_{DD}$, V_{CC} to be $0.3V_{DD}$, and K to be 0.56, the energy consumption equation can be re-written as follows:

$$E_{EBL} = 0.045\alpha C_L V_{DD}^2 + \frac{0.45\pi^2 C_L R}{T} C_L V_{DD}^2 + E_{Crowbar}. \quad (3.13)$$

Notice that the energy consumed by the Evaluation stage is relatively small compared to the Boost stage, due to aggressive voltage scaling. Moreover, notice that the energy consumed by the Boost stage is not affected by the switching activity of the Evaluation stage, making charge-recovery logic more suitable for datapaths with high switching activity. However, for appropriate values of the RC_L product, an EBL design can achieve high performance and significant energy savings by trading off latency for energy.

3.6 Summary

In this chapter, we present Enhanced Boost Logic, an energy efficient charge-recovery logic family. EBL achieves high energy efficiency and high performance through the use of aggressive voltage scaling and charge-recovery techniques. Comparing to previous charge-recovery logic, EBL uses fewer supplies, yielding lower crowbar current. By achieving higher gate overdrive, EBL is capable of computing complex logic in a single phase, yielding lower latency overhead at the system level compared to previous charge-recovery logic implementations.

CHAPTER 4

EBL FIR Filter Test Chip

In this chapter, we present a 14-tap 8-bit EBL-based Finite Impulse Response (FIR) filter test-chip, a real-world application that is used to assess the performance and energy efficiency of EBL. Previous fabricated charge-recovery logic test-chips have implemented only relative simple arithmetic function such as chains of logic gates, adders, or multipliers, making our FIR filter the most complex arithmetic function implemented in charge-recovery to date. In addition, the latency of this EBL-based FIR is only 1.5 cycles longer than that of a similar-performance static CMOS design that has been implemented separately. Fabricated in a $0.13\mu\text{m}$ CMOS process, the test-chip includes a fully-integrated 3nH inductor and an integrated clock generator with frequency scaling capability. Correct operation has been experimentally validated across the 365-600MHz range. When operating at its resonant frequency of 466MHz, the FIR test-chip dissipates 39.1mW and achieves 45% efficiency in the recovery of energy through its two clock phases. The associated figure of merit equals $93.6\text{nW}/\text{MHz}/\text{Tap}/\text{InBit}/\text{CoeffBit}$, a 29% improvement over previously-reported high-performance FIRs with sampling rates above 500MHz [13, 78].

The remainder of this chapter is organized as follows: Section 4.1 provides an overview of the EBL-based FIR that we design using our semi-custom design methodology. Section 4.2 describes a semi-custom design methodology that we developed for

facilitating EBL-based circuit development. Section 4.3 gives results from Spice-level simulations of the EBL FIR filter and its static CMOS counterpart with identical architecture. In Section 4.4, we present measurement results from our EBL FIR filter test-chip. Conclusions are given in Section 4.5.

4.1 FIR Test-Chip Overview

Since each EBL gate has a built-in transparent latch, the state-intensive nature of a transpose-type FIR filter coupled with the relatively simple combinational logic between its state elements make it an ideal demonstration platform for EBL. To that end, we have used EBL to design an 8-bit 14-tap transpose-type FIR filter in a $0.13\mu\text{m}$ CMOS digital process with 7 levels of Cu and 1 ultra-thick layer of Al. This section gives an overview of our FIR test-chip.

A complete block diagram of the FIR test-chip is shown in Fig. 4.1. The FIR filter

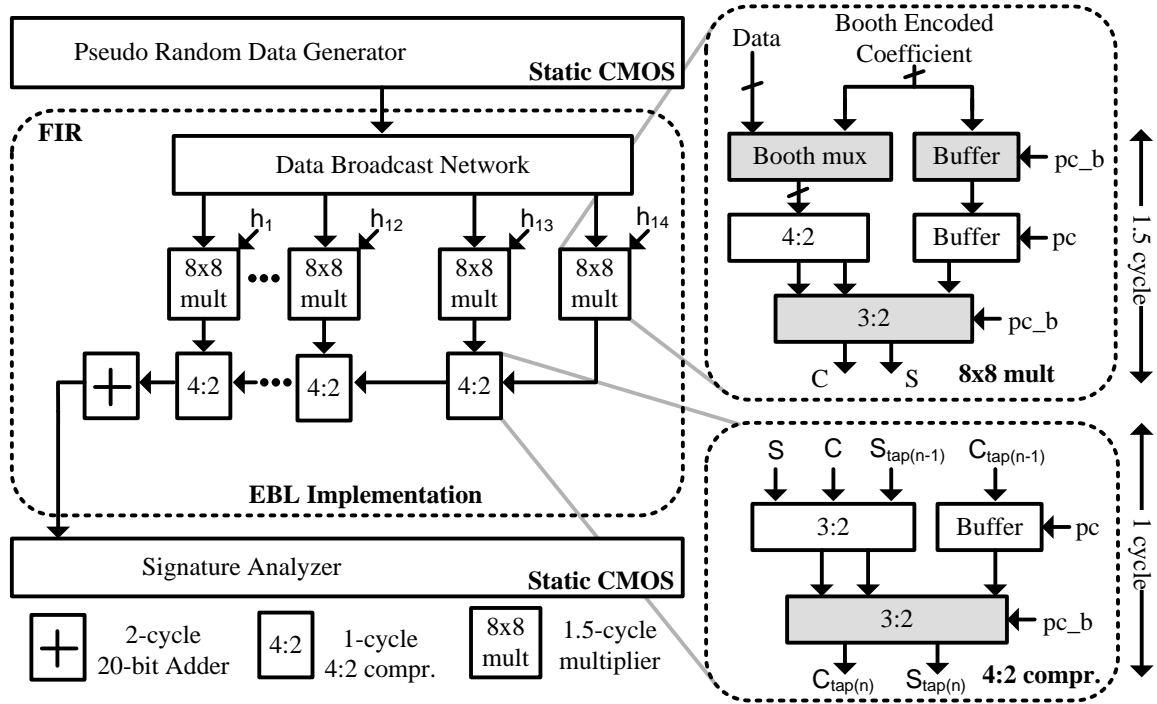


Figure 4.1: FIR block diagram with clock generator and pulse generator.

is pipelined to take advantage of EBL’s potential for low latency overhead. Input data are broadcast to each tap within 1 cycle. Each 8x8 multiplier takes 1.5 cycles to merge the partial products from the Booth mux to the sum and carry vector pairs. Each tap takes 1 cycle to merge the sum and carry vector pairs from the previous tap and the 8x8 multiplier. The vector pairs are then merged in a 20-bit hybrid carry-look-ahead/carry-select adder with 2 cycles of latency. The longest path through the EBL-based FIR has a latency of 18.5 cycles. Compared to other high-performance low-latency arithmetic implementations, the latency overhead of the EBL-based FIR is 1.5 cycles: 0.5 cycle in the 8x8 multiplier, and 1 cycle in the 20-bit adder.

EBL’s latency improvement over previous generations of charge-recovery logic is based on its ability to implement logic functions of high complexity. The single-stage schematic of an EBL 4-to-2 compressor shown in Fig. 4.2(a) highlights the capability of EBL for implementing high-complexity functions. The Sum function has an evaluation stack height of six, and the Carry function has an evaluation stack height of five. Fig. 4.2(b) shows the 121.6 μm^2 layout implementation of the 4-to-2 compressor in EBL, which has only 7.6% area overhead when compared to a static CMOS implementation.

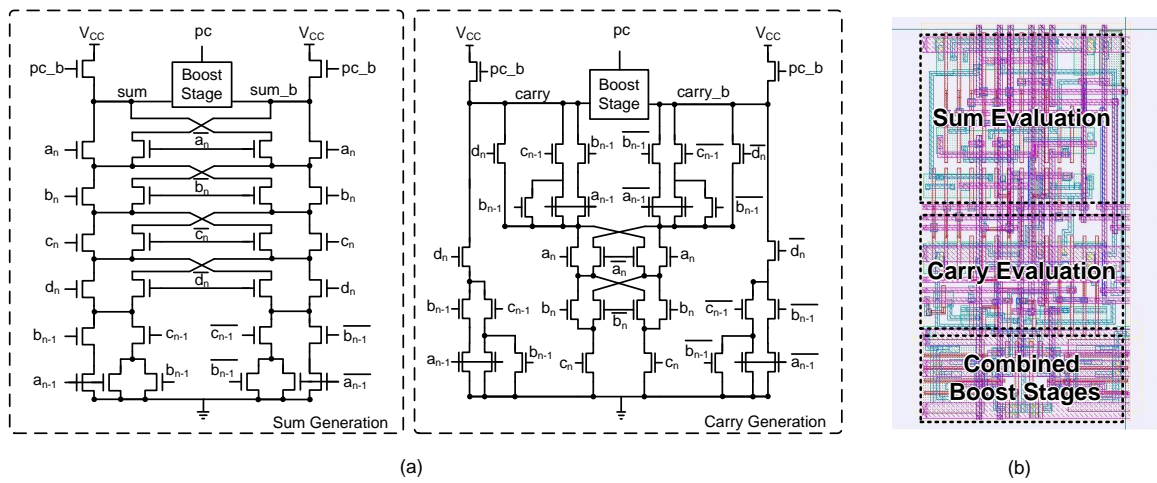


Figure 4.2: EBL-based 4-2 compressor schematics and layout.

To reduce power dissipation of simple EBL gates, the EBL gates with stack height less than 4 have been implemented with a complementary pull-up network (PUN) in their Evaluation stage, and are thus identical to SBL gates. Due to the simplicity of the logic function they perform, their true and complement PUNs are sized so that the gates have similar performance as when designed with precharge devices. The PUN-based implementation of such relatively simple gates improves energy efficiency, as it prevents the increased crowbar currents of the inherently (yet, in this case, unnecessarily) faster precharge-based EBL implementation.

The correct functionality of the FIR filter is validated through the use of Build in Self Test (BIST) circuitry. The BIST generates a pseudo-random cellular automaton sequence [79], processes the filter output using a multiple-input shift register to generate a signature vector, and captures the state of the signature at a user-defined time. When the signature vector matches a scan-in template, a single bit is inverted creating a single-bit signature output, which can be observed off chip.

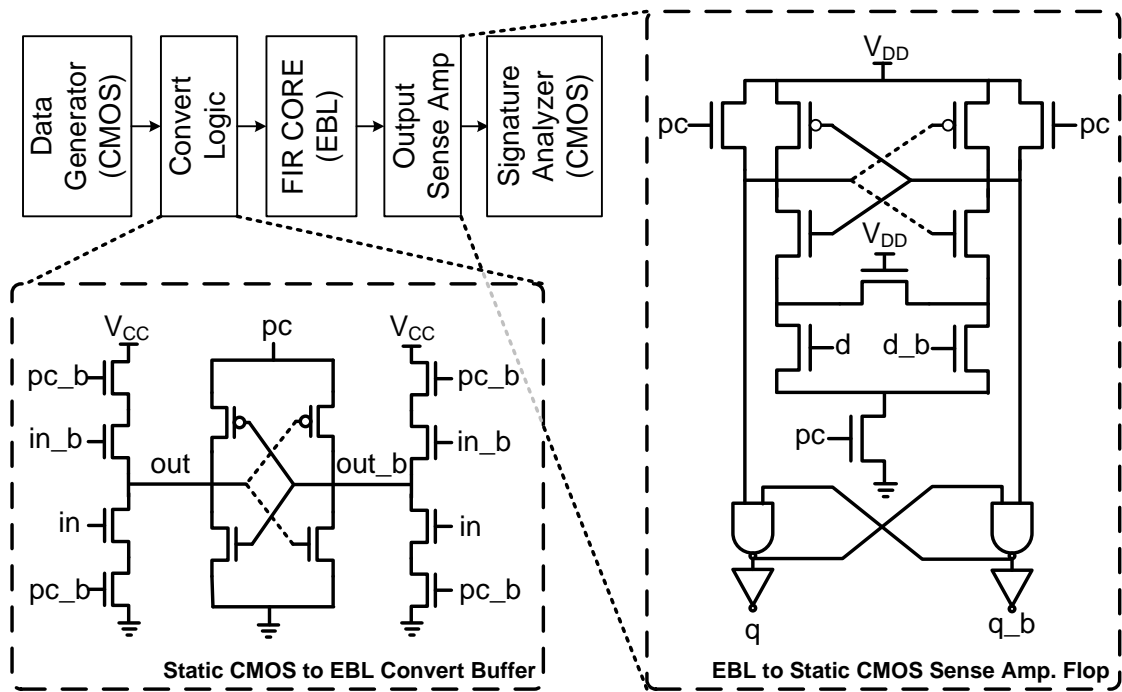


Figure 4.3: Conversion circuits between EBL and static CMOS gates.

This chip also demonstrates EBL’s ability to be seamlessly integrated with static CMOS circuits. Fig. 4.3 shows the schematic for the interface circuits between BIST circuitry implemented in static CMOS and the EBL-based FIR filter. From static CMOS to EBL, a clock-gated NMOS is inserted in both the PUN and PDN of an EBL buffer to reduce leakage paths from power-clock to V_{CC} . From EBL to static CMOS, a sense-amplifier flip-flop converts signals from an EBL gate output to a standard digital signal. The BIST circuits around the FIR filter, such as the pseudo-random sequence generator, the signature analyzer, and the signature generator, are implemented using standard cell. The output of the pseudo-random sequence generator is sent to convert buffers, and the outputs of the FIR are digitized by the sense-amplifier flip-flops before being processed by the signature analyzer.

The FIR datapath is clocked by two partially-overlapping clock phases that are generated using a blip generator. Extending the original blip design shown in Fig. 3.4,

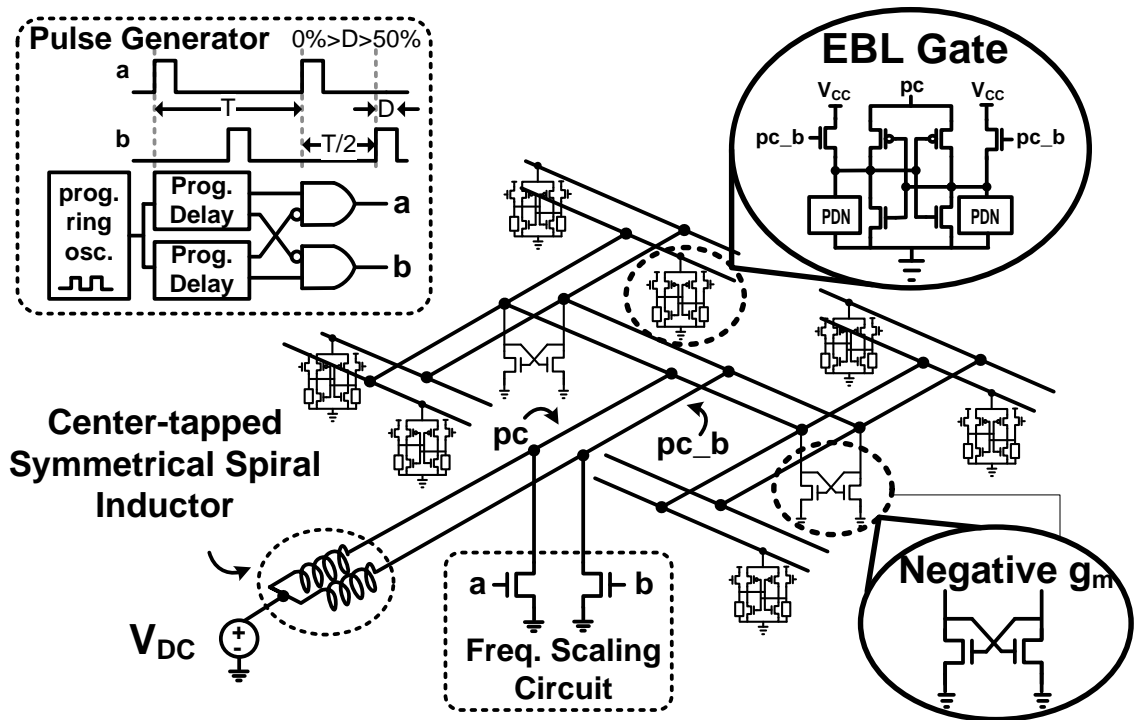


Figure 4.4: Blip clock generator with frequency scaling circuits and power-clock distribution.

our clock generator has been designed with a distributed set of switches, rather than a centralized pair of switches. Moreover, it has been supplemented by a pair of switches that are driven by an external reference clock, thus enabling frequency-scaled operation. Our blip generator is shown in Fig. 4.4. The inductor is a fully-integrated 3nH symmetrical inductor implemented on the top 2 levels, and the center-tap of the inductor is connected to clock generator supply V_{DC} . Simulation results based on the foundry-provided model show that this inductor has a quality factor of 9.65 at 466MHz. Twelve blocks of cross-coupled NMOS switches with $2400\mu\text{m}$ total active width have been distributed across the FIR filter to provide the negative trans-conductance required to maintain the clock oscillation. The pair of NMOS switches used for frequency scaling are connected to the two inductor terminals and force the clock to oscillate at the reference frequency generated by an on-chip programmable ring oscillator. The inputs **a** and **b** of the frequency scaling switches can be selectively gated off to control drive strength, yielding driver sizes W in the range $0 < W < 150\mu\text{m}$. To provide for maximal energy efficiency, the duty cycle D of **a** and **b** can be set in the range $0\% < D < 50\%$ through two programmable delays. The complementary power-clocks are routed out of the same side of the center-tap inductor and connected to the EBL gates, first through a 2-level H-tree, and then through a sparse clock grid.

In our test-chip, two EBL FIR cores was fabricated: one with the inductor on the side, and another one with the inductor over its FIR core. Fig. 4.5 shows a microphotograph of the 600MHz EBL FIR core with inductor on the side. The FIR module occupies a total area of $715\mu\text{m} \times 350\mu\text{m}$. Including BIST, the design takes $800\mu\text{m} \times 430\mu\text{m}$. The 3nH integrated inductor has a M1 stripe ground shield and lightly doped substrate directly below to improve the quality factor. Including its moat, it occupies about 0.14 mm^2 . The programmable ring oscillator and the frequency scaling clock generation circuit are placed between the inductor and the

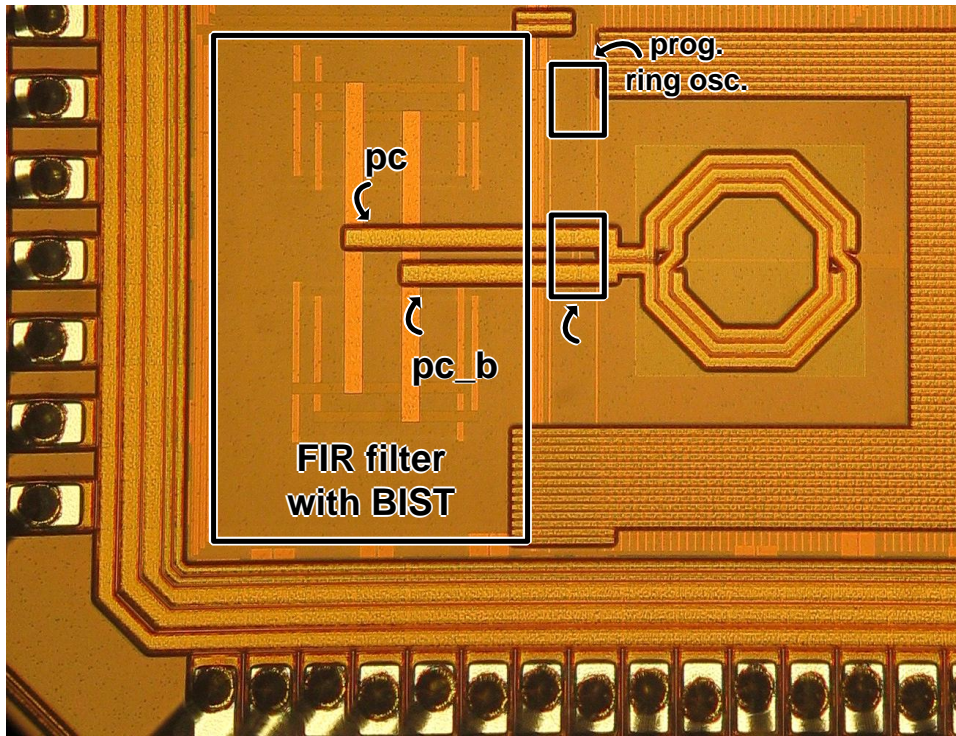


Figure 4.5: Microphotograph of the FIR core with the inductor on the side.

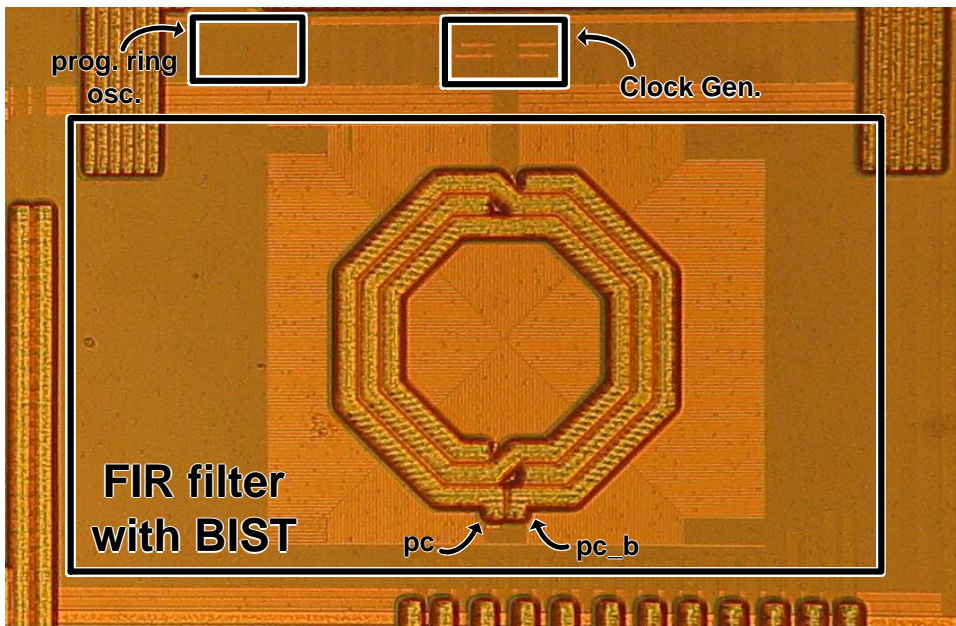


Figure 4.6: Microphotograph of the FIR core with the inductor over circuit.

FIR filter. Fig. 4.6 shows a microphotograph of the EBL FIR core with the inductor over circuitry. The 3nH integrated inductor has a M4 and M5 patterned ground shield. The layout for the FIR core and the BIST circuitry is shared between the two cores, so it also occupies $800\mu\text{m} \times 430\mu\text{m}$. However, since the inductor is directly over the circuit, this EBL FIR core achieves 29% reduction in area compared to the core with the inductor on the side.

4.2 EBL Design Methodology

Typically, charge-recovery logic has been designed using transistor-level simulation to verify functionality and electrical properties. The design and verification of large charge-recovery logic systems is therefore challenging, since the number of simulation cycles it takes to excite all possible input combinations and all possible timing arcs is at least exponential with the number of inputs. Even with the use of fast Spice programs such as Synopsys HSPICE or Cadence Ultracircuit, the computation required for such an approach is still prohibitively high.

This section presents a semi-custom design methodology for EBL that led to improvements in the performance of the FIR test-chip presented in this paper, while significantly reducing design time. This methodology enables the use of switch-level Verilog simulation. More importantly, it enables the use of industrial static timing analysis tools to verify the electrical properties of an EBL design. We first present an overview of our EBL design methodology, as applied to the FIR test-chip. We then describe the approach to switch-level netlist generation for Verilog simulation and LVS check using the same schematic. Finally, we describe our process to generate a LIBERTY format model file (.LIB) for the static timing analysis tools to verify electrical properties.

For the realization of the EBL-based FIR, we developed an EBL standard cell library with 65 EBL gates. Most of these cells are special cases of a 4-to-2 compressor

and a 3-to-2 compressor. Prior to the start of the FIR design, all the cells were verified against their behavioral Verilog models using Spice. A LIBERTY format model file was created for the EBL standard cell library based on post-layout extracted Spice results, which are described in more detail later in this section. The FIR filter was pipelined manually, and correct functionality was verified through Verilog simulation. After manual place-and-route, the final layout was extracted, and the final netlist and the extracted parasitics were sent to the static timing analysis tool for timing closure. Timing violations were fixed either by sizing up gates or through architectural modifications.

Function verification is based on switch-level Verilog simulation by converting each EBL gate to its logic equivalent, a complementary combinational logic driving transparent latches. Even though it is possible to create a behavioral Verilog model for each EBL gate, we choose to generate switch-level Verilog models from schematics, since such a bottom-up verification approach is simpler and less prone to human errors than a top-down behavioral approach. The switch-level model generation proceeds as follows. During switch-level Verilog netlist generation, the Evaluation stage is converted to complementary combinational logic. The pull-up precharge devices in the Evaluation stage are instantiated as special NMOS devices in schematic, so that they would be netlisted as weak NMOS devices in Verilog. The use of these weak devices eliminates the possibility of having contention between the precharge devices and the pull-down networks in Verilog simulation. The Boost stage is netlisted as a pair of transparent latches, one for the true output and another for its complement. To netlist the Boost stage as a pair of transparent latches, the input and the output of the Boost stage need to be separated. In the schematic, a special parameterizable Boost stage cell is created with separate input and output ports and is instantiated in all EBL gates, enabling the Verilog netlister to systematically map all Boost stages in the design.

To reduce human errors, the same schematic used to generate switch-level model is used for LVS purposes. To that end, the differential inputs and outputs in the Boost stage cell are shorted using schematic shorting elements, `cds_thru`. By adding a new property in the LVS deck for `cds_thru`, the LVS program sees the Boost stage as a cell with two differential bidirectional ports, which enables each EBL gate to be LVS clean without maintaining additional schematic.

Making the EBL gate compatible with the LIBERTY format library model file is the key enabler for running static timing analysis on an EBL design. We observe that there is only one EBL gate in each clock phase, and that in the beginning of the Boost phase, the Boost stage behaves more like a sense amplifier in a SRAM array than a transparent latch. Based on these observations, we have developed a characterization script to extract the typical gate-level parameters such as pin capacitance, propagation delay, and transition time based on a new set of definitions.

The propagation delay of an EBL gate is the time it takes the Evaluation stage to switch its output pair. It is defined as the delay between the crossover of the two clock phases and the time when the differential outputs reach a certain voltage, as shown in Fig. 4.7. The crossover point of the two clock phases is chosen as the onset of the power-clock, since its voltage level is close to the threshold voltage of Boost stage devices. For a sufficiently large output voltage difference, an EBL gate would operate correctly even if that voltage difference across the output pair is less than full V_{CC} . For reference, the rule-of-thumb voltage difference across the outputs before the onset of a sense amplifier is set to be at least 120mV. To margin for higher capacitance mismatch and process variation in our methodology, we used a 150mV voltage difference across the outputs.

Similar to conventional static CMOS standard cell characterization, the transition time (slew) parameter indicates the quality of a transition. However, since all EBL outputs track the power-clock waveforms, all transitions switch at the same rate as

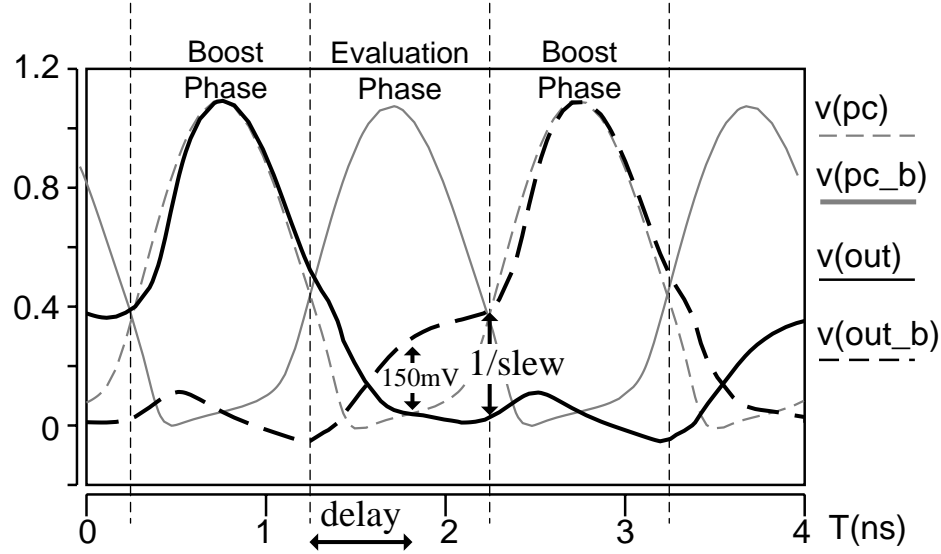


Figure 4.7: EBL design methodology static timing delay and slew definition.

the power-clock, making it meaningless to track the actual transition time using the normal 10% to 90% definition. Instead, our characterization script uses the transition time parameter to assess how well the outputs are able to track the power-clocks by redefining the transition time as the inverse of the voltage difference across the outputs at the onset of the power-clocks, as shown in Fig. 4.7. A larger voltage difference across the outputs at the onset of the power-clock implies that the Boost stage would be able to amplify the differential output pair more efficiently, in which case the outputs would track the power-clock more closely. In our design, 150mV was picked as the minimal voltage difference across the outputs at the onset of the power-clocks, since this requirement yields a delay between the power-clock and the output during the Boost phase to be less than 10% of the cycle time, yielding a smaller voltage drop across the cross-coupled PMOS devices in the Boost stage. As the voltage difference drops below the desired 150mV, the delay between the power-clock and the output increases. In extreme cases, the output pair is amplified in the wrong direction and the gate malfunctions.

During .LIB file generation, our characterization script sweeps across a range of

output loads, and generates a 1x7 table for each propagation delay and a 1x7 table for each transition time parameter. One .LIB file is generated for each target clock frequency and clock amplitude, since these two parameters affect the input pulse width and amplitude, which in turn affect the performance of the Evaluation stage. With extracted parasitics from a placed-and-routed layout, the .LIB file enables the use of static timing analysis tools to ensure timing closure and track design margin using the redefined transition time parameter described in the previous paragraph.

4.3 Simulation Evaluation

In this section, we present results from Spice-level simulations of our EBL FIR filter. For comparison purposes, we also present results from the simulation of a conventional static CMOS FIR filter that we have designed using the same architecture and a standard cell library in the same $0.13\mu\text{m}$ process technology as the EBL FIR filter. The foundry-provided inductor spice models have been used in our simulations. The simulation results in this section are compared with measurement results obtained from the EBL FIR filter core with the inductor on the side in Section 4.4.

The graphs in Fig. 4.8 give the per-cycle energy consumption of our EBL FIR filter at various clock frequencies when operating in self-resonant mode. For each frequency, the graphs give total energy consumption, energy supplied to the clock generator through V_{DC} , and energy supplied to the Evaluation stages of the EBL gates through V_{CC} . The data at each frequency point have been obtained using the inductance value indicated next to it, and with the minimum supply setting that ensured correct function. Simulations have been performed using Synopsys HSPICE with the post-layout extracted netlist based on the BSIM model and with foundry-provided parameterized inductor models. Correct operation has been confirmed from 230MHz to 800MHz, with a center-tapped symmetric spiral inductor ranging from 11nH to 0.95nH, respectively.

Our simulation results in Fig. 4.8 show that the energy consumed by the clock generator (V_{DC}) dominates the total energy consumption. They also show that total and clock generator energy requirements generally decrease, as frequency decreases. From 800MHz to 350MHz, the energy supplied to the clock generator decreases almost linearly with the operating frequency, as predicted by the expression for the term E_{Boost} in Equations (3.2) and (3.12). However, the energy consumed by the logic increases as operating frequency decreases, due to the increasing crowbar current in the clocked precharge devices of complex logic gates such as the 4-to-2 compressor. Total energy consumption increases from 350MHz to 230MHz, a phenomenon that can be explained by the choice of inductor at 230MHz. Specifically, to provide the larger inductance required for resonance at 230MHz, inductor width is reduced from $15\mu\text{m}$ to $8.5\mu\text{m}$, increasing inductor resistance and impacting efficiency in the following two ways. First, since inductor resistance is a large portion of the overall effective

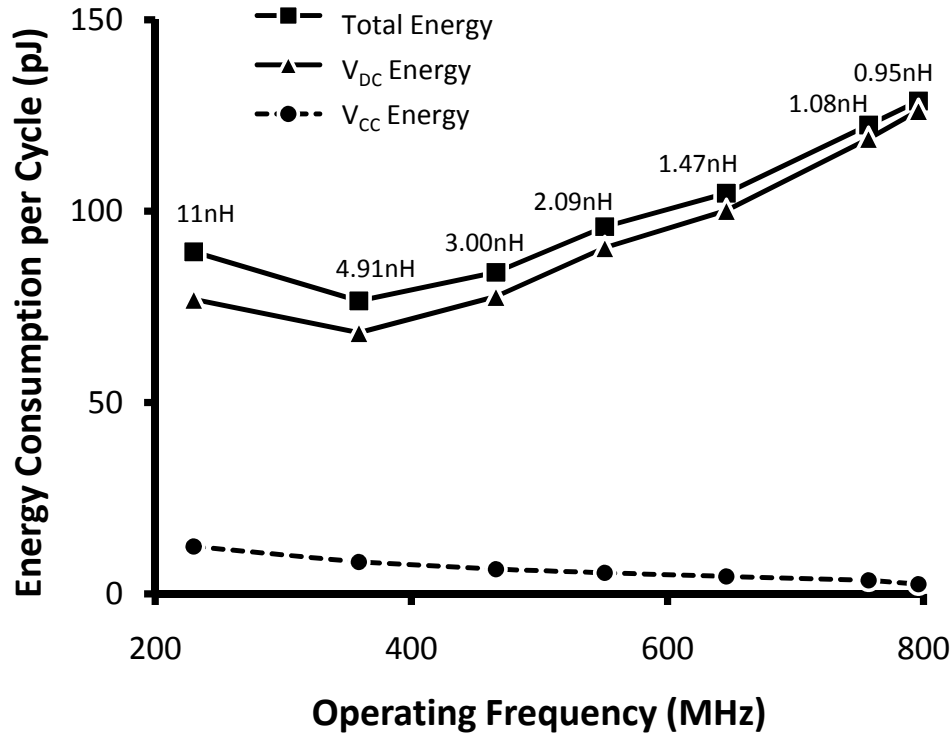


Figure 4.8: Simulated energy consumption of self-resonant EBL FIR filter.

resistance, total resistance increases by a much greater proportion than transition times, resulting in increased energy consumption over 350MHz. Second, operation at 230MHz requires an even larger inductor than implied by a straightforward resonant frequency calculation, since the increased inductor resistance results in an increased damping factor and, based on Equation (3.1), an increased resonant frequency. At frequencies below 300MHz, it appears that an off-chip discrete inductor would be the preferred choice with regard to energy savings.

The graphs in Fig. 4.9 give per-cycle energy consumption versus clock frequency when the FIR filter is operating in frequency-scaled mode with a fixed 3nH integrated inductor. Energy requirements are reported separately for the logic (V_{CC}), the clock generator (V_{DC}), and the frequency-scaling circuitry (V_{CK}). Total energy is given across the frequency range, as well as when the FIR is self-resonating with the

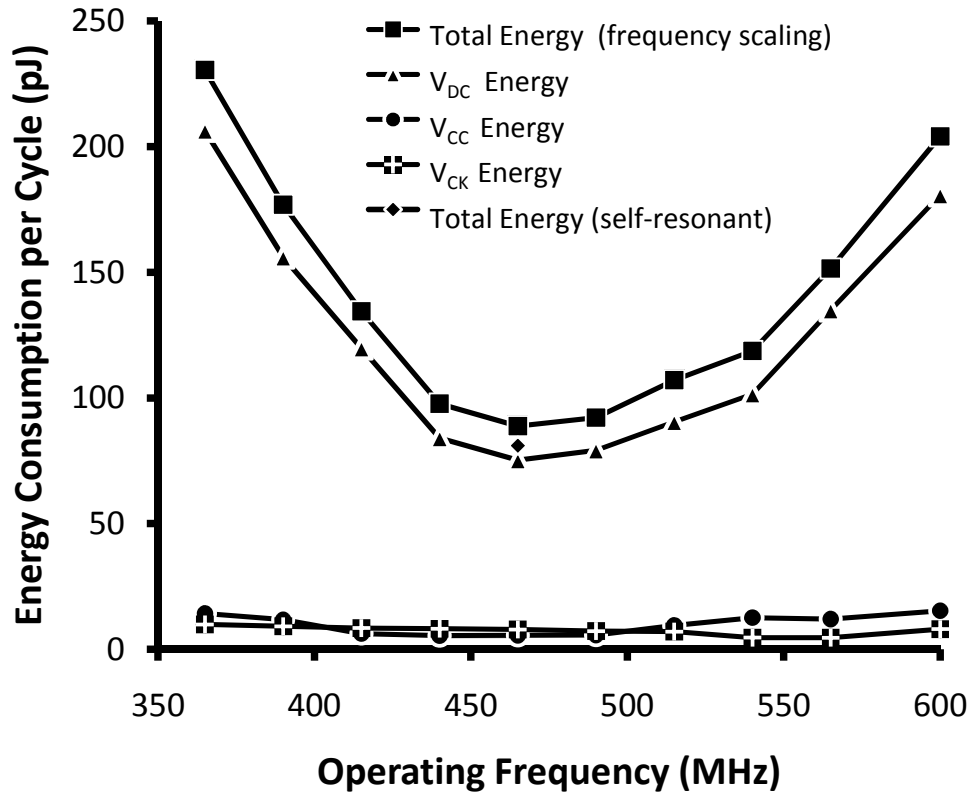


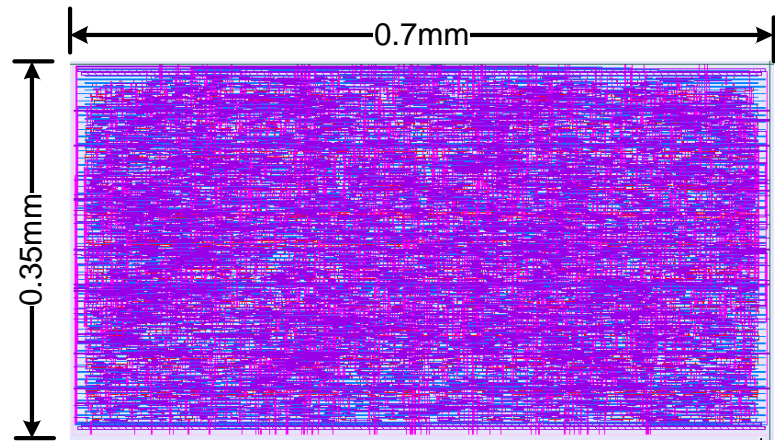
Figure 4.9: Simulated energy consumption per cycle of the frequency-scaled EBL FIR filter.

frequency-scaling circuitry turned off. The minimum energy point is achieved at the resonant frequency of 466MHz. When the frequency-scaling circuit is enabled at resonance, the energy consumed by the V_{CK} domain becomes non-zero, while the energy drawn from V_{DC} and V_{CC} remains almost the same. As operating frequency deviates from resonance, higher V_{DC} and V_{CK} supplies are required to maintain clock amplitude at the rails. As operating frequency deviates sharply from resonance, the clock waveforms become more distorted, and their overlap increases, yielding increased leakage current and increased energy consumption in the V_{CC} domain.

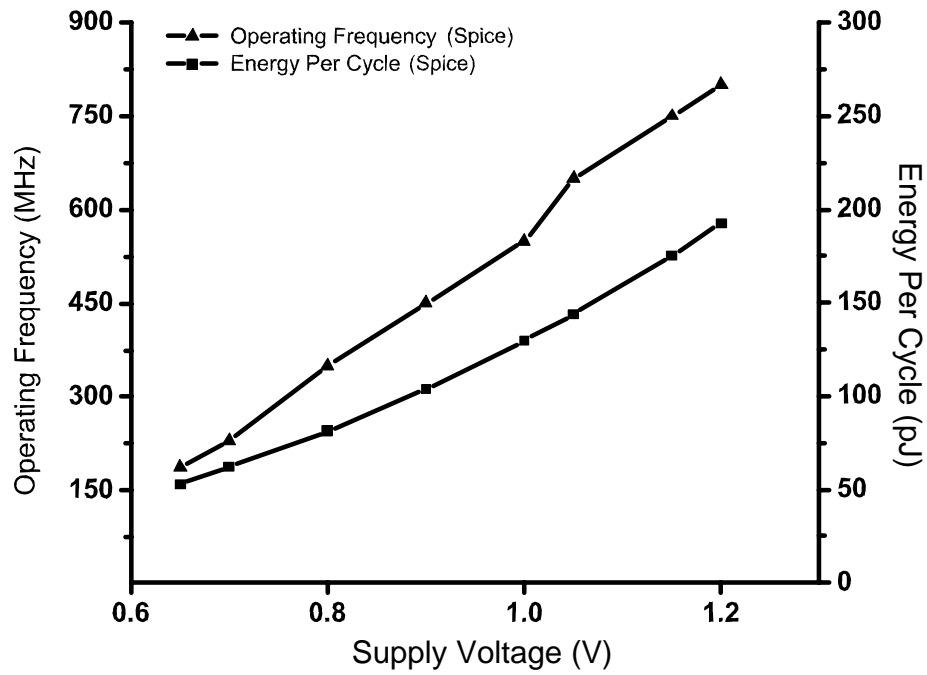
For comparison purposes, we have synthesized a conventional static CMOS FIR filter using Synopsys Design Compiler and a standard cell library in the same $0.13\mu\text{m}$ technology as the EBL-based design. The conventional FIR filter is designed with an overall latency of 19 cycles, as flip-flops do not allow for half cycles. The synthesized netlist is automatically placed and routed using Cadence Encounter with 80% area utilization and a synthesized clock tree. The layout of the resulting design is shown in Fig. 4.10(a). With a footprint of $0.35\text{mm} \times 0.7\text{mm}$, the synthesized FIR filter incurs 37% area overhead mainly due to the on-chip inductor.

Fig. 4.10(b) shows the operating frequency and the per-cycle energy consumption of the conventional FIR filter versus supply voltage. At the nominal supply of 1.2V, the conventional FIR filter achieves 800MHz with more than 80% of the standard cells at X1 or X2 drive strength. From 230MHz to 800MHz, energy consumption increases quadratically with supply voltage, as expected.

The graphs in Fig. 4.11 give the energy requirements of the voltage-scaled conventional FIR and the EBL-based FIR in self-resonant mode for a range of operating frequencies. For clock frequencies above 350MHz, the EBL FIR filter exhibits 21-34% energy savings over its conventional counterpart. Below 350MHz, the large inductors required to oscillate the system have poor quality factor due to their large dimensions and high turn counts, increasing the energy requirements of the EBL design compared



(a)



(b)

Figure 4.10: (a) Layout of conventional static CMOS FIR filter. (b) Simulated operating frequency and energy per cycle vs. supply voltage for static CMOS FIR filter.

to its conventional counterpart.

The graphs in Fig. 4.12 compare the energy requirements of the voltage-scaled conventional FIR and the EBL-based FIR in frequency-scaling mode with a fixed 3nH inductor. With the frequency-scaling circuitry enabled, the EBL FIR filter consumes 17% less energy when operating at its resonant frequency of 466MHz, and consumes less energy than the static CMOS FIR filter from 440MHz to 540MHz. When running in self-resonant mode at 466MHz, the EBL FIR filter achieves 21% energy savings compared to the conventional FIR filter running with a 0.91V supply.

4.4 Measurement Results

This section gives measurement results from the experimental evaluation of both FIR filter cores on the test-chip. It also presents a comparison of measurement and simulation results of the FIR core with the inductor on the side, showing good agreement between the two, with relative discrepancy between measurement and simulations staying within 12% for operating frequencies ranging from 365MHz to 600MHz.

The graphs in Fig. 4.13 show current and inferred per-cycle energy consumption in the EBL FIR core with the inductor on the side for operating frequencies in the 365MHz to 600MHz range. Reported energy includes the energy in the clock generator (V_{DC}), Evaluation logic (V_{CC}), and frequency scaling circuitry (V_{CK}). Each point in the plot corresponds to the minimum energy dissipation of the circuit over all possible values of V_{DC} , V_{CC} , D , and W that result in correct operation, as verified by observing the expected signature waveform. At the resonant frequency of 466MHz, the minimum energy of 84pJ is observed for $V_{DC} = 0.57V$, $V_{CC} = 0.41V$, and $V_{CK} = 1.2V$, with all frequency-scaling circuitry disabled. At 93.6nW/MHz/Tap/InBit/CoeffBit, the EBL FIR core with the inductor on the side represents a 29% improvement over previous high-performance FIR designs [13].

At self-resonance, the clock generator powered by V_{DC} is a significant source of

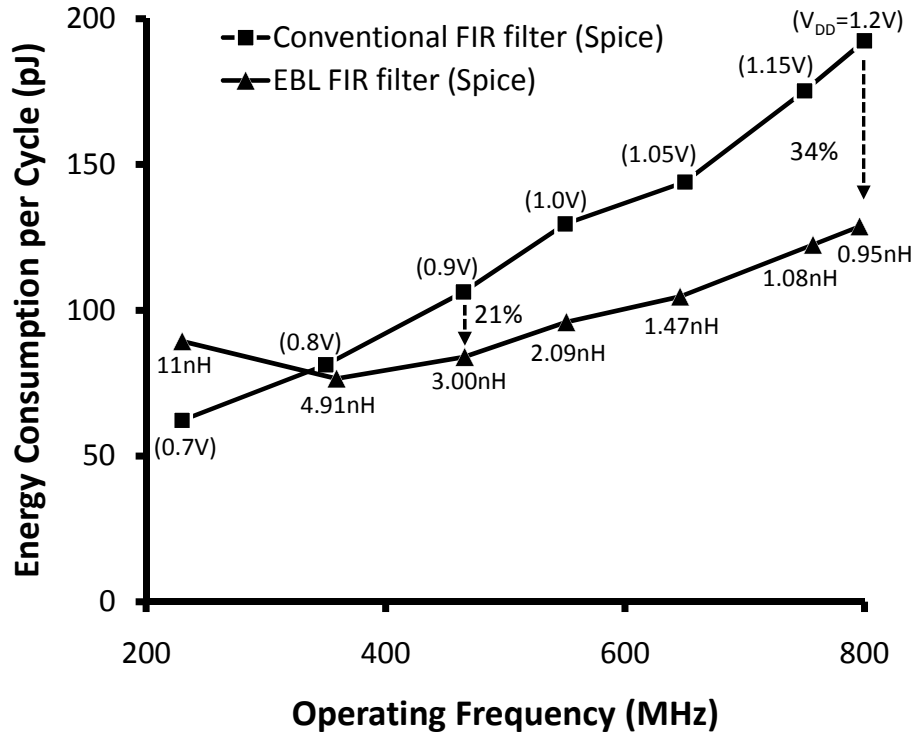


Figure 4.11: Energy consumption per cycle comparison between conventional FIR and self-resonant EBL FIR filter.

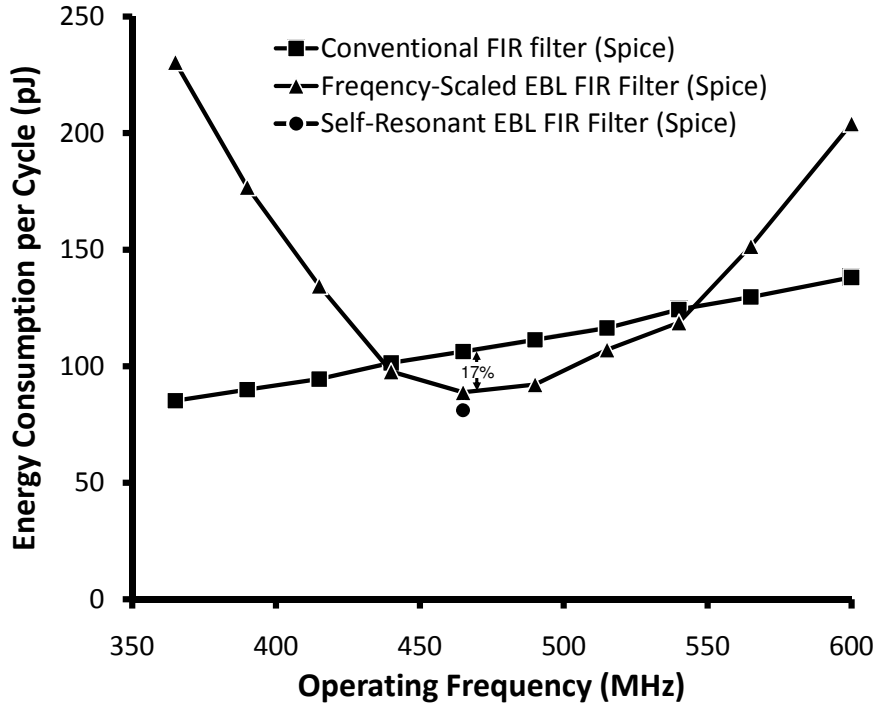


Figure 4.12: Energy consumption per cycle comparison between conventional FIR and frequency-scaled EBL FIR filter.

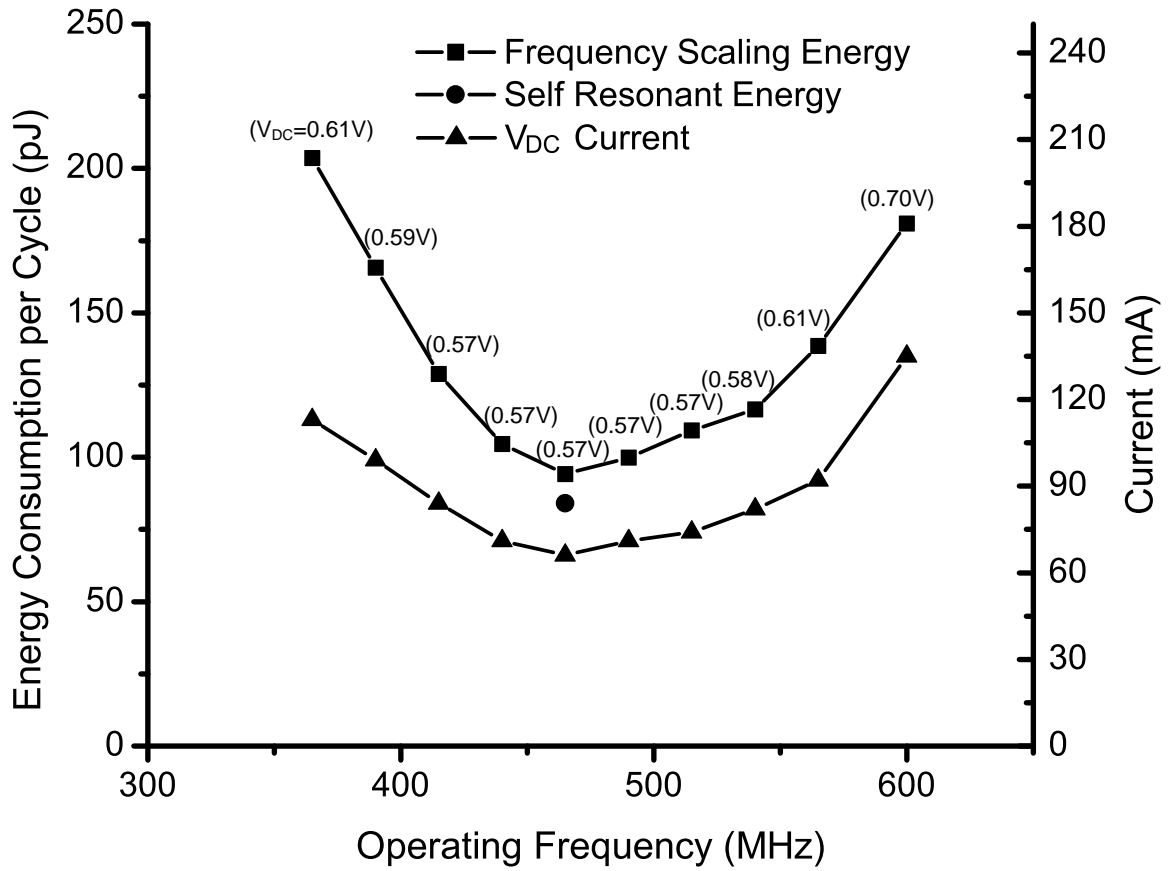


Figure 4.13: Energy dissipation and current vs. operating frequency of the EBL FIR core with the inductor on the side.

energy consumption, and the Evaluation logic remains a small percentage of the total energy requirements. With frequency scaling enabled, the energy consumption of the Evaluation logic remains relatively constant, while most of the additional energy consumption is caused by additional current in the programmable switches that drive the oscillator off resonance. The ability to scale operating frequency allows post-silicon tuning to mitigate the effects of process variation on the resonant frequency of the system. By increasing V_{DC} to 0.7V, correct operation is verified at 600MHz. Fig. 4.14 summarizes the chip statistics and measurement results, and Table 4.1 compares the performance of our EBL FIR with previously reported FIR filter test-chips with equal or greater sampling rate than our design at its 466MHz resonant

Technology	0.13 μ m 8M CMOS
Taps, in/out/coeff bit	14, 8 / 20 / 8
EBL Gate count	3330
Total/active area	0.76 / 0.34 mm ²
Frequency range	365MHz – 600MHz
VDC Supply range	0.57V – 0.63V
Resonant Frequency	466MHz
Estimated Q	1.76
Energetics @ resonance	
Power Clock supply (V _{DC})	0.57V
EBL evaluation supply(V _{CC})	0.41V
Total/logic/clock power	39.1 / 2.0 / 37.1 mW
Total/logic/clock energy	83.9 / 4.4 / 79.5 μ J
Figure of Merit	93.6nW/MHz/Tap/InBit/CoeffBit
Input switching activity	0.5

Figure 4.14: Statistics and performance summary table of the EBL FIR core with the inductor on the side.

Paper	This Work	[78]	[13]
Design Type	14-tap 8-bit FIR	8-tap 6-bit FIR	14-tap 8-bit FIR
Technology	0.13 μ m	0.18 μ m	0.13 μ m
Nominal Supply(V)	1.2	1.8	1.2
Operating Frequency(MHz)	466	225	1010
Sample rate(MSample/s)	466	550	1010
Power Dissipation(mW)	39.1	36	122.5
Area(mm ²)	0.34	0.3	0.85
Power / MHz / Tap / In-Bits / Coeff-Bits	93.6nW	230nW	133nW

Table 4.1: FIR performance comparison table.

point.

In addition to the demonstration of energy-efficient and high-performance operation, our experimental evaluation also addresses the accuracy of the Spice-level simulation results presented in Section VI. The graphs in Fig. 4.15 show simulated and measured energy requirements of the EBL FIR core with the inductor on the side that have been obtained under identical settings for V_{DC} , V_{CC} , V_{CK} , W , and D . The two graphs track each other quite closely. For operating frequencies in the 365MHz to 600MHz range, the discrepancy between simulation and measurement stays within

12%.

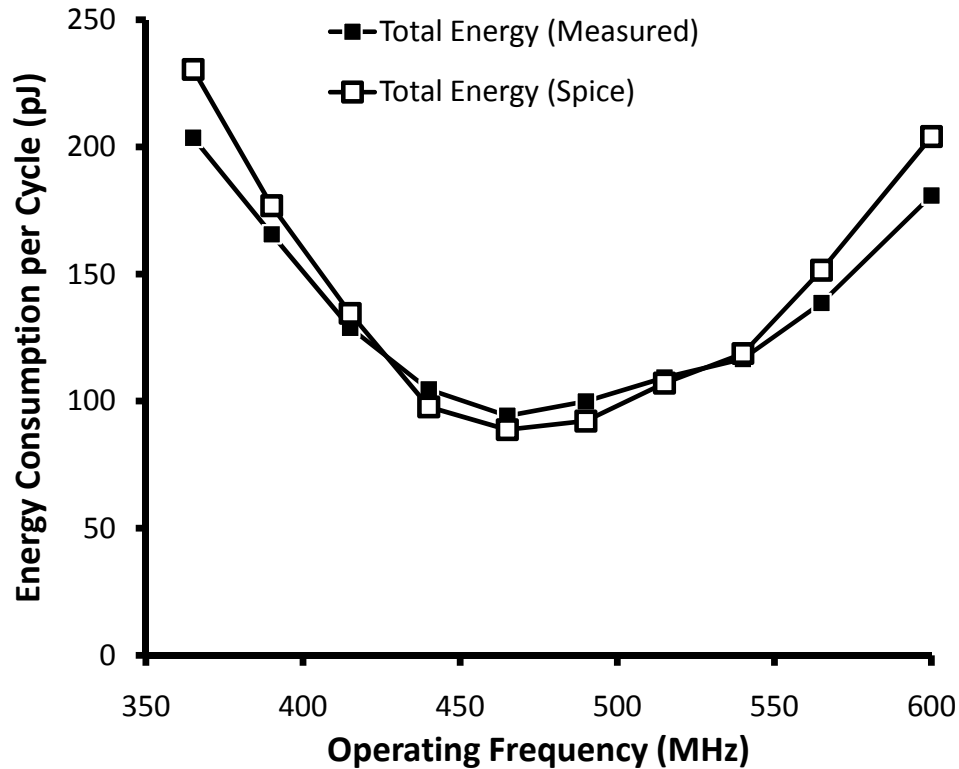


Figure 4.15: Simulated and measured energy consumption per cycle comparison between the FIR filter with the inductor on the side.

The graphs in Fig. 4.16 show current and inferred per-cycle energy consumption in the EBL FIR core with the inductor over circuitry for operating frequencies in the 365MHz to 525MHz range. At the resonant frequency of 425MHz, the minimum energy of 171.4pJ is observed for $V_{DC} = 0.72V$, $V_{CC} = 0.40V$, and $V_{CK} = 1.2V$, with all frequency-scaling circuitry disabled. The clock-related per-cycle energy measured from the V_{DC} supply is 167.7pJ, and the logic-related per-cycle energy measured from the V_{CC} supply is 3.7pJ. Since the quality factor (Q) of the inductor is lower for the inductor over circuit, the minimal V_{DC} voltage required to ensure correct functionality is much larger than the FIR core with the inductor on the side. The quality of the inductor degrades for two reasons. First, the loose power grids at levels M2 and M3 directly below the inductor hurt Q . Second, the silicon under the inductor is doped, yielding higher eddy currents. The estimated system Q is 0.89, which is 50% lower

than the core with the inductor on the side. Also notice that the core with the inductor over circuitry has a lower resonant frequency, since the M4 and M5 patterned ground shields increase parasitic capacitance. Due to the degradation in Q and lower resonant frequency, the FIR core with the inductor over circuitry achieves a figure of merit of $191.8\text{nW}/\text{MHz}/\text{Tap}/\text{InBit}/\text{CoeffBit}$, which is higher than previously-reported high-performance static CMOS designs. This experiment is our attempt to reduce the area overhead associated with integrated inductors for charge-recovery logic designs. Even though our measurement results are not as good as previously-reported results, this experiment can serve as a starting point for later work that aims at reducing the area overhead for charge-recovery logic designs.

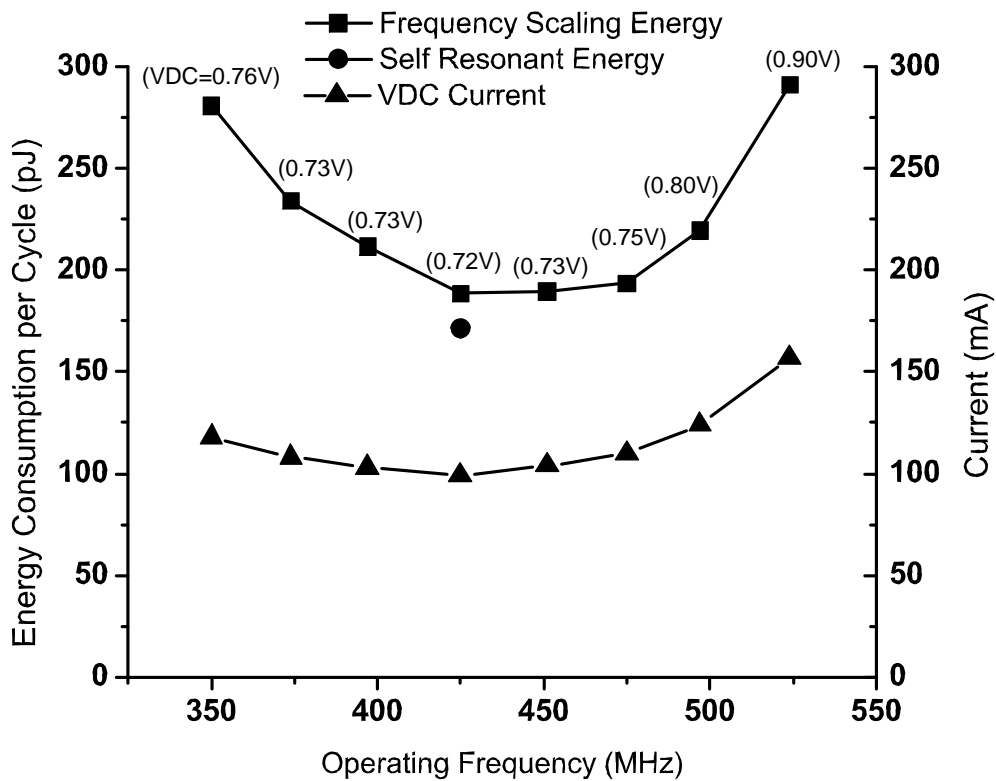


Figure 4.16: Energy dissipation and current vs. operating frequency for FIR filter with inductor over circuit.

4.5 Summary

In this chapter, we present a 14-tap 8-bit FIR filter test-chip implemented using EBL to demonstrate the high energy efficiency and low latency overhead of EBL. Post-layout simulations show that when operating in self-resonant mode, the EBL FIR filter achieves 21% to 34% lower power compared to a voltage-scaled static counterpart. A test-chip has been fabricated in the IBM 0.13 μm technology with two EBL FIR filter cores: one with the integrated inductor on the side of the FIR core, and another with the integrated inductor over the FIR core. The FIR core with the inductor on the side achieves an architecture-independent figure of merit of 93.6nW/MHZ/Tap/InBit/CoeffBit, which is a 29% improvement over previously reported FIR filter test-chips with equal or greater sampling rate than our design at its 466MHz resonant point. With an overall latency of 18.5 cycles, our EBL FIR filter has only 1.5 cycles of latency overhead compared to a static CMOS implementation. This is a major improvement over previous charge recovery logic families, which have at least an order of magnitude higher latency overhead when compare to their static CMOS implementations [7].

CHAPTER 5

Dynamic Evaluation Static Latch Logic

5.1 Introduction

In digital systems, the clock distribution network is a major source of power dissipation due to its high switching frequency and large capacitance. Clock gating has been effective in reducing dynamic power by shutting down unused local clock distribution. Its impact on peak power is limited, however, and it therefore has little effect on relaxing the cooling and power supply requirements at the system level. Resonant clocking is an alternative clock power reduction strategy targeted at reducing peak clock power. Utilizing an LC oscillation to generate a sinusoidal clock, resonant clock networks bounce energy back and forth between an inductance and the parasitic capacitance of the clock distribution. A significant portion of energy sent to the clock is recovered, also yielding potentially higher savings in average power.

Early resonant clock designs have targeted the recovery of charge from both the clock network and the outputs of their timing elements. Athas et al. [51, 52] have proposed the Energy-Recovery latch (E-R latch), which uses its clock both as a synchronization and a powering mechanism to achieve sub- fCV^2 power dissipation. Due to the single-ended nature of the E-R latch, the clock-related capacitance of E-R latch based designs exhibits significant data dependency, yielding high clock skew and jitter. The PMOS energy recovery flip-flop (pTERF) proposed by Ziesler et al. [11, 54]

recovers charge from dual-rail outputs to achieve relatively data-invariant load, yielding smaller clock jitter and skew. However, since both techniques have switches in the path of the LC oscillation, resistance associated with these switches limits the potential energy efficiency that can be achieved.

Aiming at improving energy efficiency, more recent research removes switches in the path of the LC oscillation. Drake et al. [55, 56] have implemented two-phase resonant clocking over a scan-chain network using distributed parasitic capacitance of master-slave flip-flops. However, since the sinusoidal resonant clock has a slower transition time than a conventional square-wave clock, master-slave flip-flops suffer significant performance degradation, yielding longer setup times and clock-to-Q times. To mitigate this performance degradation, multiple flip-flops have since been proposed. Mahmoodi et al. [58, 59] have proposed a single-ended conditional-capturing energy-recovery (SCCER) flip-flop and demonstrate its energy efficiency using a 64-bit multiplier that deploys a single-phase resonant clock across its entire clock network with operating frequency up to 160MHz. Ishii et al. [12] have demonstrated energy-efficient operation of an ARM926EJ-S microprocessor core, deploying a single-phase resonant clock with sense-amplifier flip-flops at clock frequencies exceeding 200MHz.

An alternative way to prevent performance degradation is by introducing clock buffers to improve clock slew. Hansson et al. [57] have implemented a 1.56GHz resonant-clock network driving flip-flops using clock buffers, improving performance at the expense of increasing clock buffer crowbar current. Chan et al. [71, 72] have implemented efficient resonant clocking in the global distribution network of a commercial processor using a distributed LC network. While they improve slew, clock buffers isolate the local clock distribution from the resonant network, thus limiting the amount of capacitance being resonated. Since the majority of capacitance is in the local clock distribution, the use of clock buffers limits the possible power savings that can be achieved.

In recent years, Sathe et al. [13, 14, 60] have addressed the performance degradation issue by proposing a latch-based resonant clock design methodology. Since latch performance is more sensitive to the voltage level of the clock waveform rather than the sharpness of the clock edge, latches improve performance by utilizing the full clock amplitude while benefiting from time borrowing. The performance and energy efficiency of latch-based resonant clock designs has been demonstrated in two FIR filter test-chips that have been fabricated in $0.13\mu\text{m}$ technology and achieve 1GHz operation with significant clock-power savings.

This chapter introduces Dynamic Evaluation Static Latch (DESL) logic, a dynamic logic with level-sensitive latches. To achieve high energy efficiency, resonant clocking is deployed across the entire clock network to reduce clock related power. DESL relies on the low D-to-Q delay of the dynamic evaluation and the time borrowing property of the static latch to achieve high performance. Similar to earlier resonant clock latch-based designs, DESL logic mitigates performance degradation due to the resonant clock waveforms by relying on the voltage level of its clock waveform instead of the sharpness of its clock edge. Moreover, the static latch provides a modest performance boost by relying on time borrowing. It also reduces dynamic power by converting dynamic signals to static signals, thus reducing switching activity on large capacitance nets.

5.2 DESL Structure

The key for achieving high performance and low power in the FPU is the DESL logic family shown in Fig. 5.1. A DESL buffer is shown in the cutout of Fig. 5.1. DESL logic is a variant of the SRAM domino read latch circuit [80], modified to work with a two-phase resonant clock. Each DESL gate consists of two stages: a dynamic evaluation stage, and a static transparent latch. The precharge and the clocked evaluation stack of the dynamic evaluation stage are clocked by clock-phase

CLK. To ensure robust operation, the NMOS evaluation stack height between the clocked precharge node DYN and the footer NMOS device is limited to 3. The static transparent latch constructed using a back-to-back pair of NAND gates is clocked by clock-phase CLK_B. An inverter amplifies the output of the transparent latch to increase drive strength.

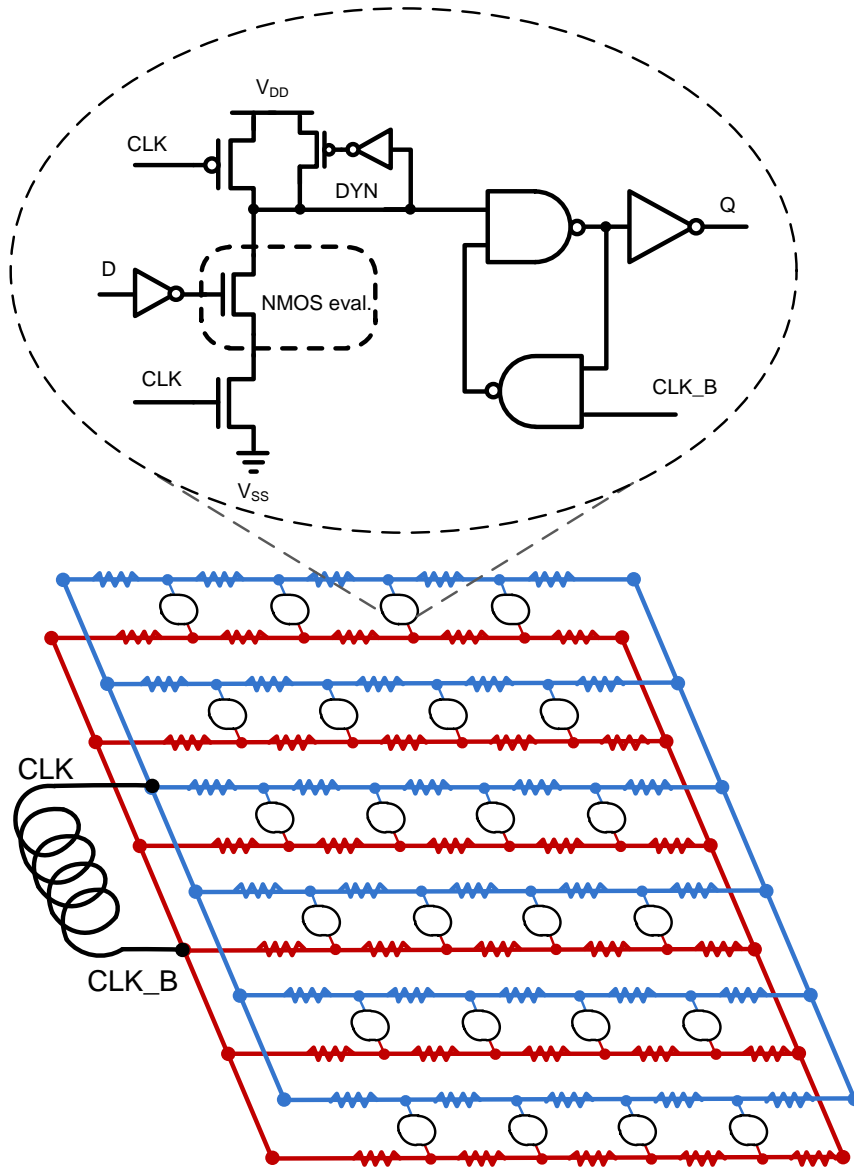


Figure 5.1: Dynamic-evaluation static-latch logic with two-phase resonant clock.

5.3 DESL Operation

Operating waveforms from the simulation of a DESL buffer are given in Fig. 5.2. When CLK is low, node DYN is precharged high, and the cross-coupled NAND keeps its previous state. When CLK rises, the gate evaluates through the clocked NMOS footer, while the pair of NAND gates clocked on CLK_B ensures that the state of node DYN is latched statically, and that output node Q remains stable through subsequent precharge/evaluate cycles that result in the same output value.

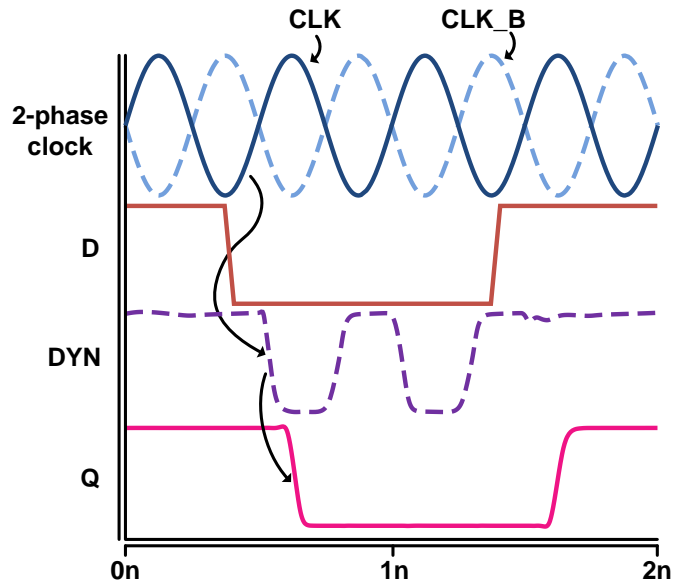


Figure 5.2: Operating waveforms of DESL buffer.

Typically, in dynamic circuits, a dynamic node performs two transitions per cycle (precharge, evaluate), driving an output buffer to amplify its state for logic evaluation in the next stage. Since the double transitions of the dynamic nodes are propagated to high-capacitance outputs, energy associated with propagating the precharge edge yields no useful computation and is wasted. DESL reduces power dissipation associated with double transitioning by inserting a static transparent latch at the output of every dynamic node DYN, converting dynamic signals to static outputs. The waveforms in Fig. 2 show that this method effectively isolates the high-capacitance output from the switching low-capacitance node DYN, thus reducing switching power

by allowing output Q to switch only once when changing state.

Another design challenge typically associated with dynamic logic is the complexity and power requirements of its clock network. To achieve maximum performance, dynamic logic usually requires multiple clock phases (four or more) in one clock cycle with numerous constraints among them. In addition, since all pull-up networks are replaced by clocked devices, clock-related power is high. In our FPU, DESL gates are synchronized by two-phase clocks which simplify clock generation and distribution. Furthermore, these two-phase clocks are amenable to low-overhead resonant implementation, resulting in maximum power savings. This combination of resonant clocking and dynamic timing elements differentiates our work from previous dynamic flip-flop designs with embedded logic [81].

Compared to static CMOS, DESL has smaller input capacitance and faster operating speed, enabling 8 FO4 per cycle, and significantly improving over commercial microprocessors whose FO4 ranges from 11 [82] to 28 [83]. To overcome the evaluation-stack height and one-logic-function-per-phase limitations, circuit, logic, and architectural optimizations are used in the design of the FPU to keep overall latency low.

5.4 DESL Clock Generation

DESL can either be clocked by a conventional "square-wave" clock or a two-phase resonant clock, but it is most energy efficient when synchronized by a resonant clock. Fig. 5.3 shows a general circuit topology for generating a two-phase resonant clock for DESL. The inductor L forms an LC tank with the parasitic capacitance C from the clock distribution network and the clock pins of DESL gates. The energy consumed on the resistance of the clock distribution network and the inductor is replenished by four switches. These switches work in two pairs controlled by four clock pulses, an , ap , bn , and bp . The switches controlled by clock pulses an and ap inject current

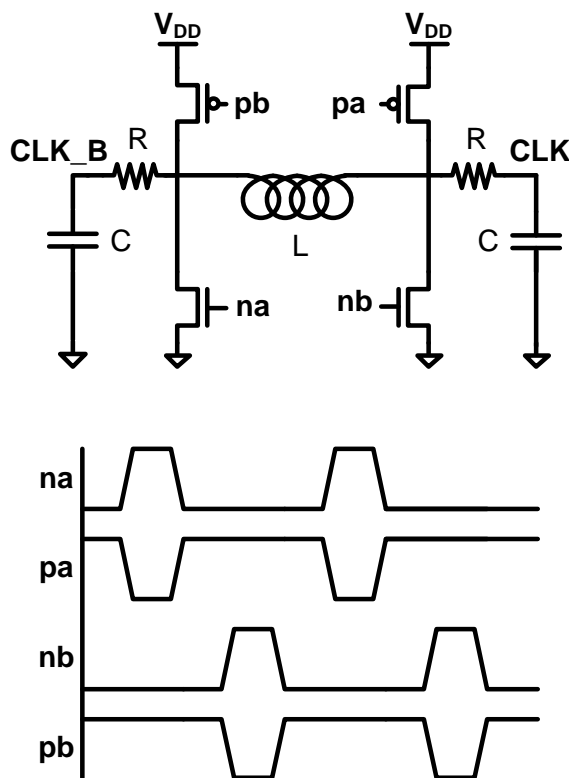


Figure 5.3: Schematic of H-bridge resonant clock generator

into the inductor with 180 degree phase difference from the switches controlled by clock pulses bn and bp . The four clock pulses are generated by an external source and have variable pulse width to achieve even higher energy efficiency. Driven by the four clock pulses, the clock generator can operate off-resonance, though it is most energy efficient when operating at the resonant frequency of the LC tank. Details of the clock generation and distribution used in the test-chip are discussed in Section 6.2.

5.5 Summary

In this chapter, we introduce resonant-clocked Dynamic Evaluation Static Latch, a novel circuit family that combines resonant clocking with dynamic logic and transparent latches. Utilizing the performance of the dynamic evaluation and timing bor-

rowing of the static latch, DESL achieves 8 FO4 per cycle, which is much shorter than the state-of-the-art microprocessor designs. Synchronized by a resonant clock waveform, DESL achieves performance levels commensurate of dynamic logic, while reducing clock-related dynamic power.

CHAPTER 6

DESL FPU Test Chips

High-performance low-power floating-point units (FPUs) are key building blocks of high performance processors. To demonstrate the energy efficiency of DESL, we have implemented a test-chip with two fused-multiply-add (FMA) single-precision FPU cores synchronized by a conventional clock and a resonant clock, respectively. In this chapter, we discuss the architecture of our FPU design and the clock networks used in each FPU core. We also show measurement results and compare them with other state-of-the-art FPUs.

Fabricated in a 90nm 9-metal Low-Power CMOS process with a nominal supply of 1.2V, the test-chip includes two FPU cores implemented in DESL. One core is synchronized by a two-phase resonant clock generated using a 0.41nH integrated inductor, while the other is synchronized by a conventional square-wave clock. Correct operation has been validated across the 1.67GHz-2.07GHz range. When operating at its resonant frequency of 1.81GHz with a 1.32V supply, the 0.391mm² resonant FPU dissipates 334mW and achieves 10.82 GFLOPS/W. Compared to its conventionally-clocked version implemented side-by-side on the same die, it yields a 31.5% decrease in power consumption and 32% improvement in GFLOPS/W. When forced to run off resonance, the resonant FPU reaches 4.14 GFLOPS at a clock frequency of 2.07GHz. An early version of this work has been published in [10].

The resonant clock FPU presented in this thesis breaks new ground on several fronts. With more than 4,000 resonant-clocked gates, it is the largest and fastest resonant-clock design ever reported with resonance deployed across the entire clock network. In previous designs with resonance deployed across the entire clock network, reported resonant frequencies are at the 1GHz mark [14]. In the IBM CELL processor with 3.2GHz resonant frequency [71], resonant clocking is deployed only in the 830 buffers of its global clock distribution network, yielding limited overall power savings. Compared to other reduced-latency FPUs, our resonant clock FPU occupies the smallest area, even when the on-chip inductor is included, and achieves the lowest overall latency [82, 84]. It also achieves the highest energy efficiency among continuous data streaming FPUs [82, 84].

The remainder of this chapter is organized as follows: Section 6.1 provides an overview of the FPU architecture that we designed using our semi-custom design methodology. Section 6.2 describes the resonant clock and the conventional clock generation and distribution networks used in each FPU core. In Section 6.3, we present measurement results from both our resonant-clock and conventional-clock DESL FPU cores. We also give a comparison of our resonant clock FPU with other state-of-the-art reduced-latency FPUs. Conclusions are given in Section 6.4.

6.1 FPU Architecture

In this section, we describe the architecture of our FPU. First, we describe our architectural optimizations for achieving lower overall latency compared to IEEE-754 compliant FPUs. Then, we describe the circuit and logic optimizations implemented in each main building block to achieve the lowest overall latency among competing FPUs.

The FPU is designed to perform the single-precision fused-multiply-add operation $A \times B + C$. To achieve high performance for multimedia applications, it deviates from

the IEEE-754 standard and supports only the round-toward-zero rounding mode, just like [82, 84], thus speeding-up the fraction datapath by simplifying the sticky bit computation and removing the fraction rounding function. Moreover, it supports denormalized numbers as inputs, but for additional performance boost, it drops support for denormalized number as output. This deviation from the IEEE standard, combined with architectural optimizations and DESL’s ability to achieve 8 FO4 per cycle, enables this FPU to achieve an overall latency of 64 FO4, which is much lower compared to 100 FO4 for other state-of-the-art IEEE-754-compliant FPUs [85].

Fig. 6.1(a) shows the FPU architecture. The FPU has been optimized to leverage the capabilities of DESL logic and achieve an overall latency of 8 cycles: The radix-4 Booth encoder and Booth mux take 1 cycle; the 6 layers of 3-to-2 compressors that compress the partial products from the Booth mux and the aligner take 3 cycles; the 75-bit end-around-carry (EAC) adder takes 2.5 cycles, and the normalizer takes 1.5 cycles. The critical paths are found in the shift amount decoder that drives the mux select line in the final stage aligner, and in the computation of the indicator bit in the leading-zero-anticipatory (LZA) logic [86].

Fig. 6.1(b) shows the floorplan and the placement of the main FPU building blocks. The data flows in the vertical direction with 10 bits on the left for exponent and 52 bits on the right for fraction. To fit within the allocated width, part of the aligner, the final layer of 3-to-2 compressors, the EAC adder, and the normalizer are folded. The solid black color in the floorplan shows the area occupied by active poly, giving some insight into how the datapath is arranged. To reduce interconnect wire length, a portion of the aligner unit (shaded with grey background) is interleaved in the multiplier. Similarly, the LZA unit (shaded with grey background) is interleaved in the fraction adder. Our FPU is designed using a semi-custom flow with 104 standard cells. With 16 tracks/bit, it is completely routed with M3 metal and below, reserving the higher level metals for clock and power distribution.

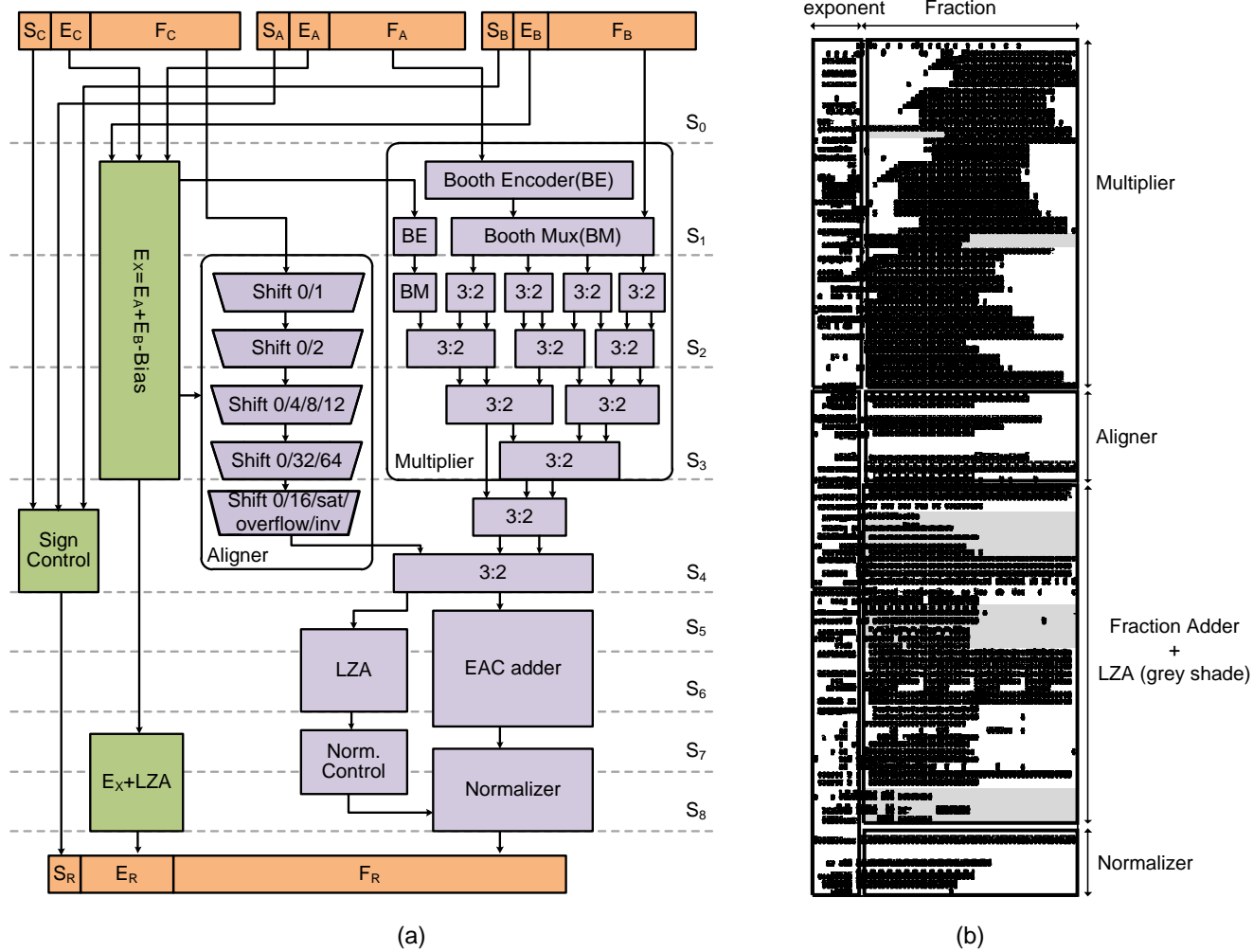


Figure 6.1: (a) FPU architecture and (b) floor plan of FPU main building blocks.

Fig 6.2 shows the implementation details of the multiplier design. Radix-4 Booth encoding reduces the number of partial products by half. To enable the use of Booth encoding, one additional bit is added in front of the fraction's implicit bit to enable two's complement number multiplication. The Booth encoder logic processes 13 3-bit segments of the A-Operand with 1-bit overlap to produce 5-bit 1-hot control signals: positive 2X (P2), positive 1X (P1), zero (Z), negative 1X (N1), and negative 2X (N2). Similar to other FPUs [82], a leading zero is added in front of the most significant bit of A-Operand, so that the most significant Booth encoding is always positive. Since

the resulting sign bit is always zero, this method reduces the partial products by one, thus reducing the complexity of the multiplier.

The B-Operand is first sent to the middle of the multiplier, and both true and complement versions of the B-Operand are sent to fan-out Booth multiplexers (Boothmux) to reduce the resistance associated with distribution in the next phase. Controlled by the result of the Booth encoder, the Boothmux logic generates $-2B$, $-B$, 0 , B , and $2B$ by shifting the true or complement version of the B-Operand, or zeroing all bits. The partial product based on the 3 most significant bits of the A-Operand is produced one clock phase later to allow enough time for the denormalized number check on the incoming A-Operand. The 13 partial products are then compressed to two output vectors using 5 layers of 3-to-2 compressors, which are arranged to reduce wire length between each layer. The 3-to-2 compressor is chosen in our design, because its evaluation logic fits elegantly in a stack of height 3.

The aligner shifts the fraction field F_R of the C-Operand to align it with the output vectors from the multiplier. Our aligner reduces hardware complexity and limits the shift amount to be between 0 and 73 by checking if the exponent of the C-Operand is much larger or smaller than the resulting exponent from the multiplication of A-Operand and B-Operand. It also reduces complexity and pre-shifts the C-Operand by 26 bits to the left, so that it only has to shift in one direction. The fraction datapath of the aligner consists of five layers of muxes. The first layer shifts by 0 or 1 bit, the second layer shifts by 0 or 2 bits, the third layer shifts by 0, 4, 8, or 12 bits, and the fourth layer shifts by 0, 32, or 64 bits. The fifth layer in the aligner takes care of the cases when the C-Operand is much larger or much smaller than the resulting exponent from the multiplication or when previous results are shifted by 16 bits. It also covers the case when C-Operand needs to be inverted when computing a subtraction operation. The last two layers compute the sticky bit based on bits that are shifted beyond the range of the aligner.

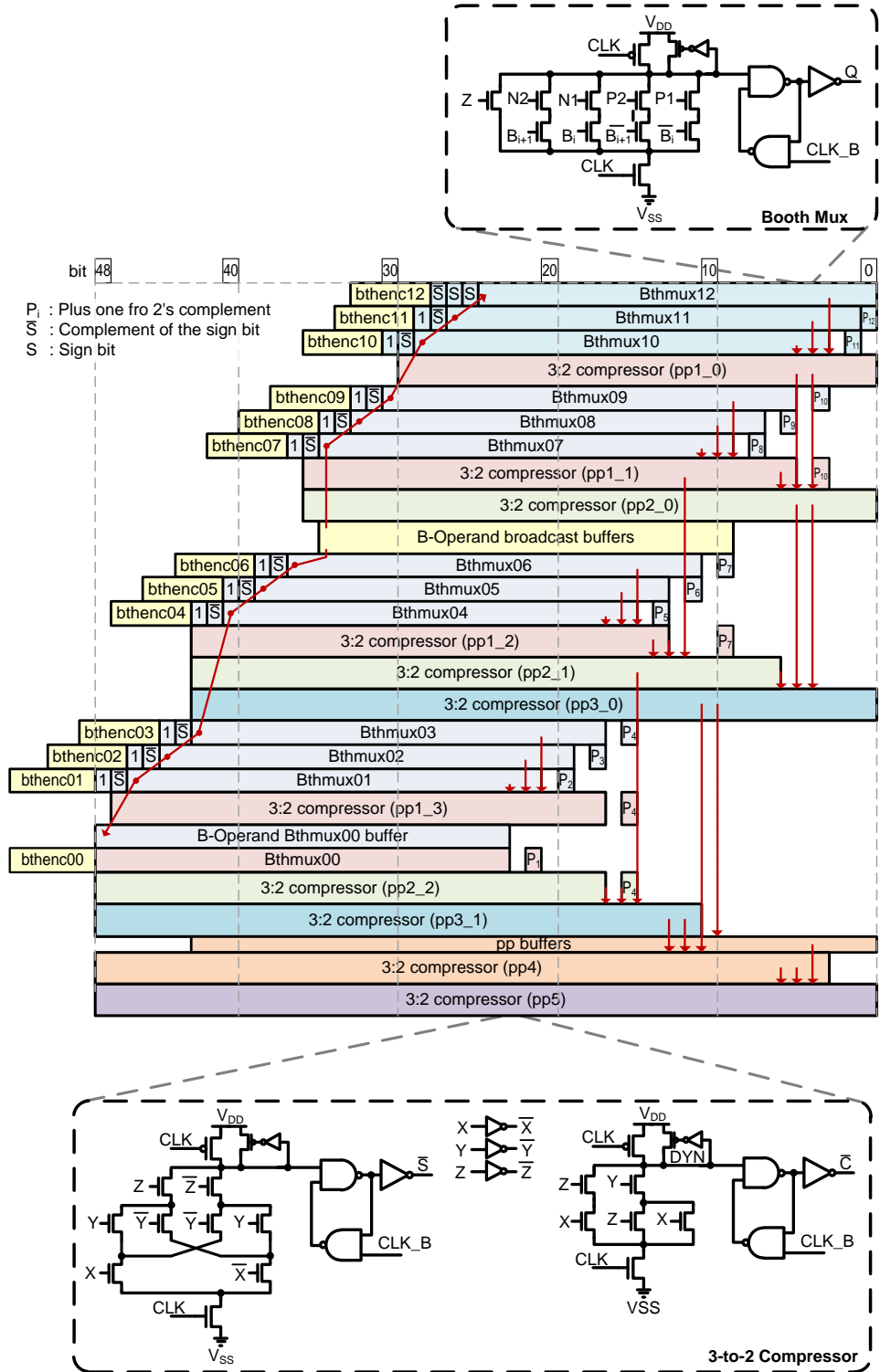


Figure 6.2: Multiplier implementation details.

The critical paths in the aligner are found in the mux-select signals. The mux-selects for the first two shifter layers are controlled directly by a pair of sum and carry bits, since encoding 2-bit one-hot mux selects would not meet the timing and stack-height-of-3 rule. Even though this decision pushes the decoding of the select signals into muxes and greatly increases the capacitance load on each select signal, the relatively small number of bits in this part of the datapath still makes non-one-hot mux-select feasible. The later shifter layers have one-hot control signals to enable wider mux implementation.

The fraction adder sums the results from the multiplier and the aligner. Since the result of the multiplier comes in a pair of sum and carry vectors, the fraction adder first merges them with the output of the aligner using a bank of 3-to-2 compressors. Due to the sign magnitude representation of the floating point number, the EAC adder is a very hardware-efficient implementation of floating-point addition, which always returns a positive magnitude result [87]. The main advantage of the EAC adder is that its result is in one's complement format, meaning a magnitude of a negative result can be generated by a simple bitwise inversion operation, which is much simpler than negating a number in two's complement system. Since the multiplier result is shorter than the aligner result, the first 26 bits of the adder are implemented as an incrementer, whereas the less significant 48 bits of the adder are implemented as a 48-bit adder. The EAC adder computes the sum ($sum0$) and sum+1 ($sum1$) to speed up the end-around carry addition. The 48-bit adder first generates propagate, P(XOR), generate, G(NAND), and kill, K(NOR) vectors, which are shared with the leading zero anticipator unit. It also computes the sum ($sum0$) and sum+1 ($sum1$) in four 12-bit segments in 1.5 cycles. In parallel, the fraction adder also computes the carry select based on the output of the 3-to-2 compressor for each of the 12-bit adders and combines the carry from each 12-bit segment with the group carry result of the incrementer. Then, a bank of 4-to-1 muxes completes both the $sum0$ and $sum1$

selection function and the coarse normalizer function by shifting the fraction result to the left by 25 bits.

Since the lengths of the three input vectors are different, the challenge of the fraction adder design is in finding a hardware-efficient method to cover the case when the multiplier result carries beyond its range. Typical two's complement arithmetic operation depends on restricting the precision of the result to ensure function correctness. Since the output of the multiplier is in the form of a pair of sum and carry vectors that has a narrower range than the output of the aligner, special attention is needed to ensure that the carry bit from the sum and the carry vectors would not corrupt the final result. An easy method would be to sign extend the sum vector of the multiplier at the cost of additional hardware. We use a more hardware-efficient method, where the sum of the multiplier is extended by one bit beyond the most significant bit. The extended bit indicates if the carry-out of the multiplier result is high, and therefore can be used to suppress the carry-out of the 48-bit adder and the end-around carry bit.

The LZA unit estimates the number of leading zeroes in parallel to the fraction adder, providing enough time to decode the control for the normalizer. Its algorithm is very similar to [86]. LZA first generates 26 indicator bits to point out the position of the first 1. Since the algorithm only estimates this position, there might be a 1-bit difference in some cases. The 26 indicator bits are then encoded into 1-hot control signals for the normalizer. The critical path of LZA is in the indicator bit generation circuit, as shown in Fig. 6.3.

The Normalizer removes all the leading zero by shifting the result of the fraction adder to the left. It consists of three layers of muxes. The first layer shifts by 24, 16, 8, or 0 bit, the second layer shift between 7 and 0 bits, and the third layer of 4-to-1 muxes covers the case when the LZA underestimates the shift amount by one bit and/or inverts the fraction when the fraction adders result is negative.

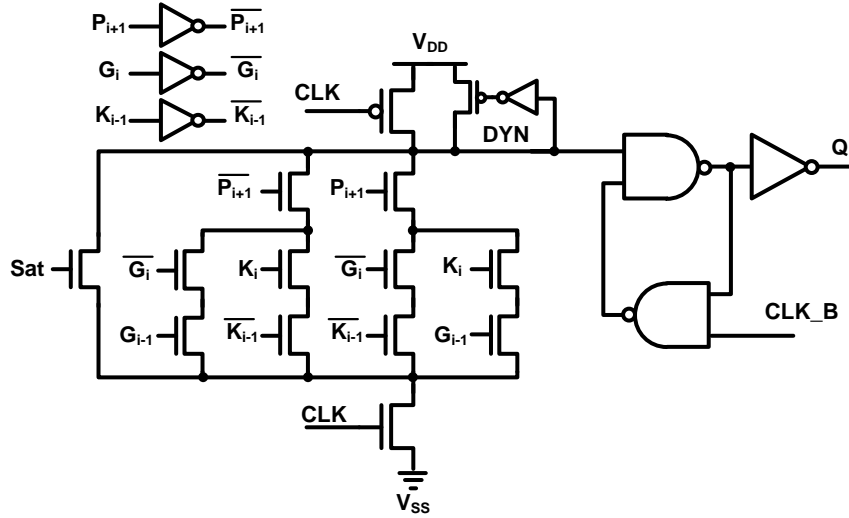


Figure 6.3: Schematic of the LZA indicator bit generation circuit.

6.2 Clock Network Overview

For comparison purposes, a FPU with identical architecture synchronized by a conventional buffered clock distribution has been implemented on the same die, side-by-side with the resonant one. The conventional-clock FPU has been derived from the same netlist as the one used for the resonant FPU. This section describes the clock distribution networks for both the resonant clock FPU core and the conventional clock FPU core.

6.2.1 Resonant Clock Network

Fig. 6.4 shows a simplified view of the two-phase resonant-clock generation and distribution network used in the resonant clock FPU. The resonant clock network consists of 114 pairs of 278 μm -long clock stripes that cover a rectangular distribution area of height 1,034 μm . Each stripe consists of a sandwich of 0.28 μm M2, 0.28 μm M4, and 0.28 μm M6 to reduce distribution resistance for higher energy efficiency. Since this FPU is designed using a semi-custom flow, all clock pins sit directly under clock stripes, further reducing distribution resistance. Top-level distribution is performed

using metal levels M9 and M8. Clock capacitance comprises the wire capacitance of the clock distribution grid and the clock-related input capacitance of all DESL gates. Together with an integrated 0.41nH inductor, this clock capacitance forms an LC tank oscillator.

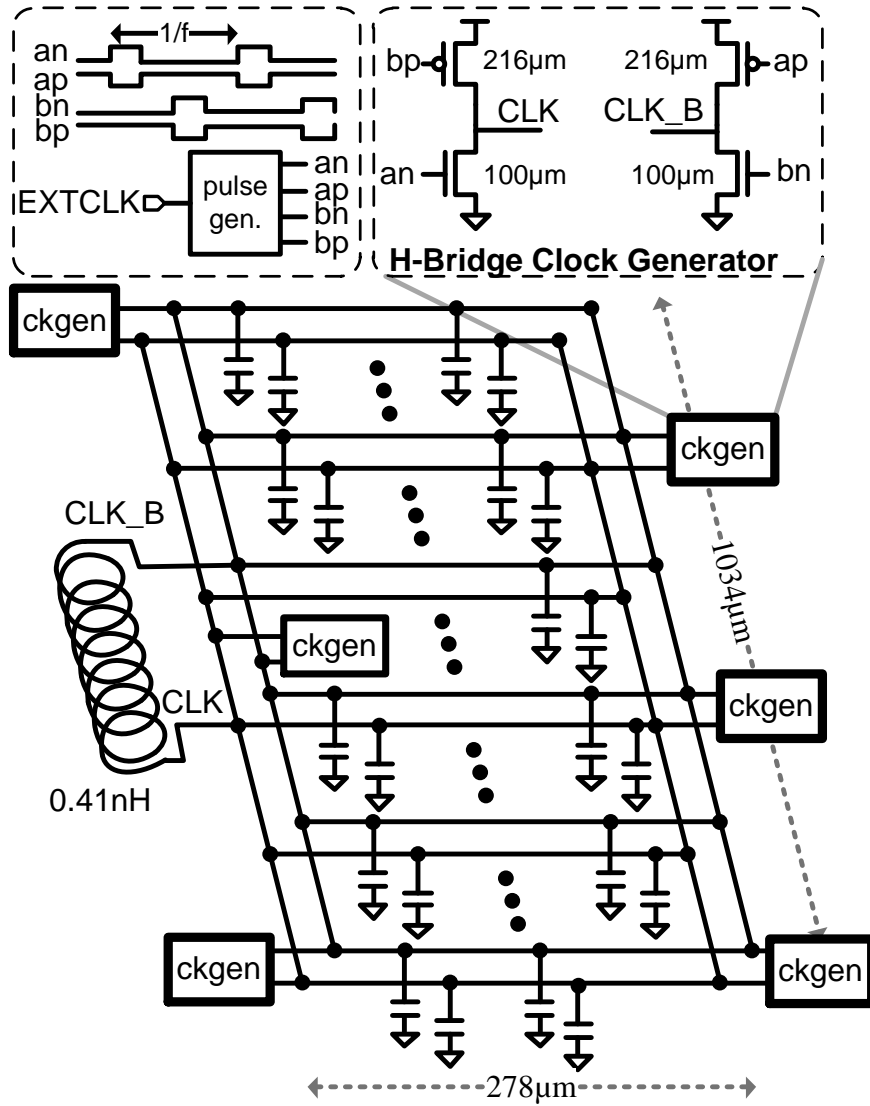


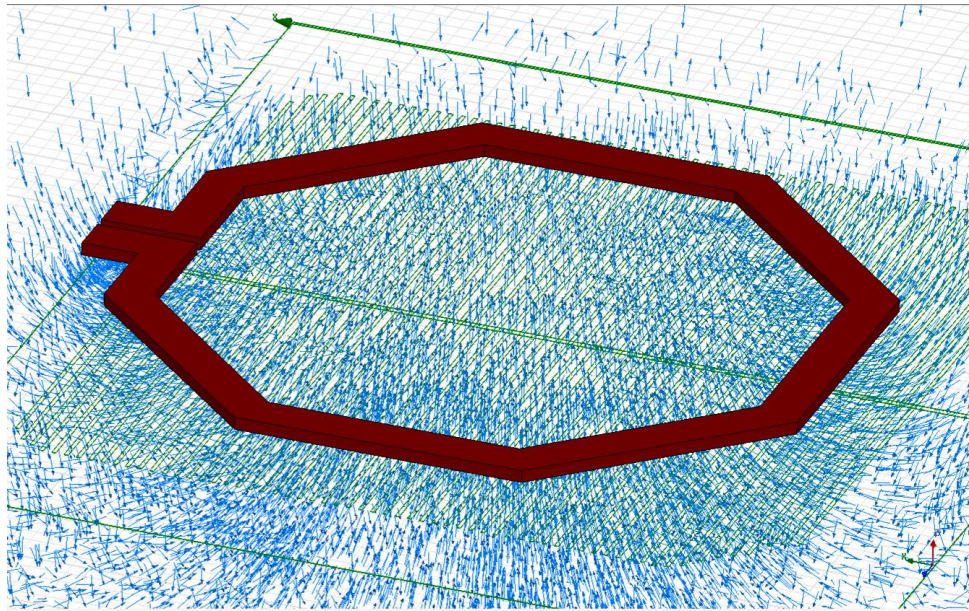
Figure 6.4: Distributed resonant clock generator circuitry and clock distribution network.

To replenish resistive losses, six H-bridge clock generator modules [5] of programmable size (up to 100μm max each) are evenly distributed across the FPU core.

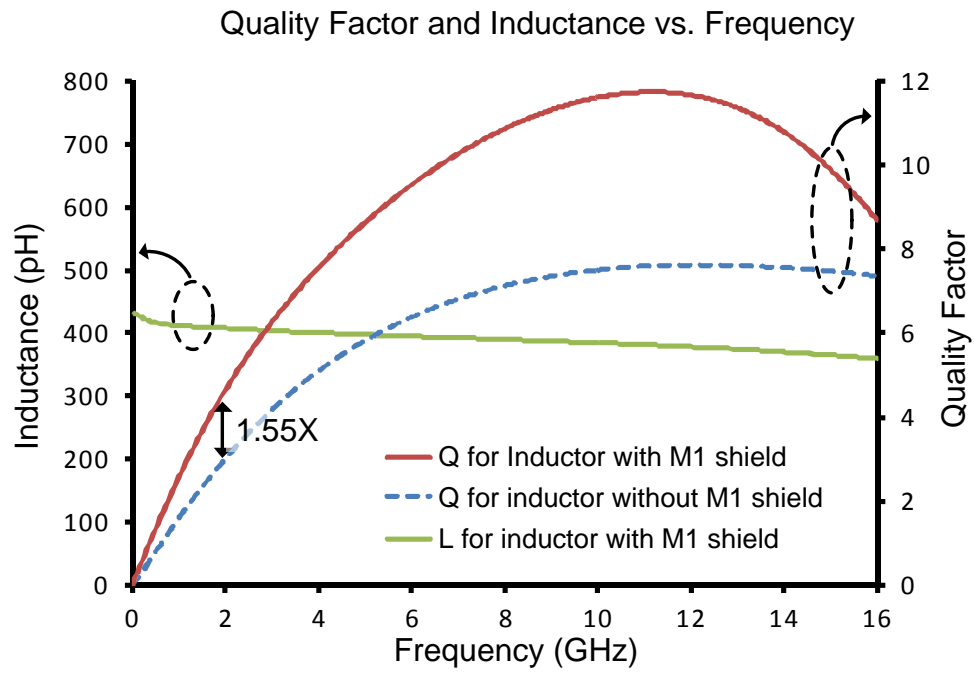
Each module comprises a pull-up/pull-down pair of devices driven by two pairs of complementary clock pulses an , ap and bn , bp with programmable duty cycle. Two pulse generators located symmetrically on two opposite sides of the FPU are used to generate these pulses. Each pulse generator drives three clock generator modules, and it is driven by the external clock source, EXTCLK, using triple-wide equal-length wire to ensure minimal clock skew. Unlike previous designs [10, 71], this clock generator topology does not rely on a half- V_{DD} supply and, thus, has no need for decoupling capacitance dedicated to half- V_{DD} supply. For reference, capacitance on half- V_{DD} supply found in [71] and [60] is about 6X and 10X of its resonating clock load, respectively. To allow for the maximum clock load to be resonated, there are no insertion buffers between the clock generator and the DESL gates. The resulting fine-grain distribution thus allows for the propagation of the resonant clock all the way to the leaves of the network.

Inductor design and evaluation was performed after extracting the clock network capacitance of the FPU design. Based on the clock capacitance loading, the inductor value was selected so that the resonant frequency is near the desired operating frequency. Inductance characteristics were evaluated using the full-wave electromagnetic simulator HFSS. The final inductor design is a 1-turn $0.19 \times 0.19 \text{mm}^2$ $12 \mu\text{m}$ -wide $3.4 \mu\text{m}$ -thick M9 metal coil with an inner diameter of $166 \mu\text{m}$ that provides 0.41nH of inductance with quality factor Q of 4.28 at 1.8GHz. Fig. 6.5(a) shows the inductor simulation setup with the magnetic field from the full-wave 3D field solver simulation result, and Fig. 6.5(b) shows the Q simulation results. A pattern of M1 strips has been added directly below the inductor to improve Q at the target operating frequency. Simulation results show that by adding the strips, the inductance increases by 0.46%, and the quality factor improves by 1.55X at 1.8GHz.

The clock network and generator have been designed to achieve a clock amplitude equal to approximately 90% of V_{DD} . When clock amplitude is higher than 90% of



(a)



(b)

Figure 6.5: (a) Full-wave 3D field solver simulation result and (b) inductance and quality factor vs. frequency.

V_{DD} , too much energy is consumed by the resistance in the clock distribution and the inductor. When clock amplitude is lower than 90% of V_{DD} , it results in significant performance degradation in D-to-Q times. Based on 2GHz simulations of two 2-input XOR DESL gates, where one is synchronized by a full- V_{DD} -swing conventional clock and another is synchronized by a two-phase resonant clock with 90% V_{DD} swing, the resonant clocked DESL gate with an ideal clock generator reduces clock-related power by 39% at the cost of 10% increase in D-to-Q times.

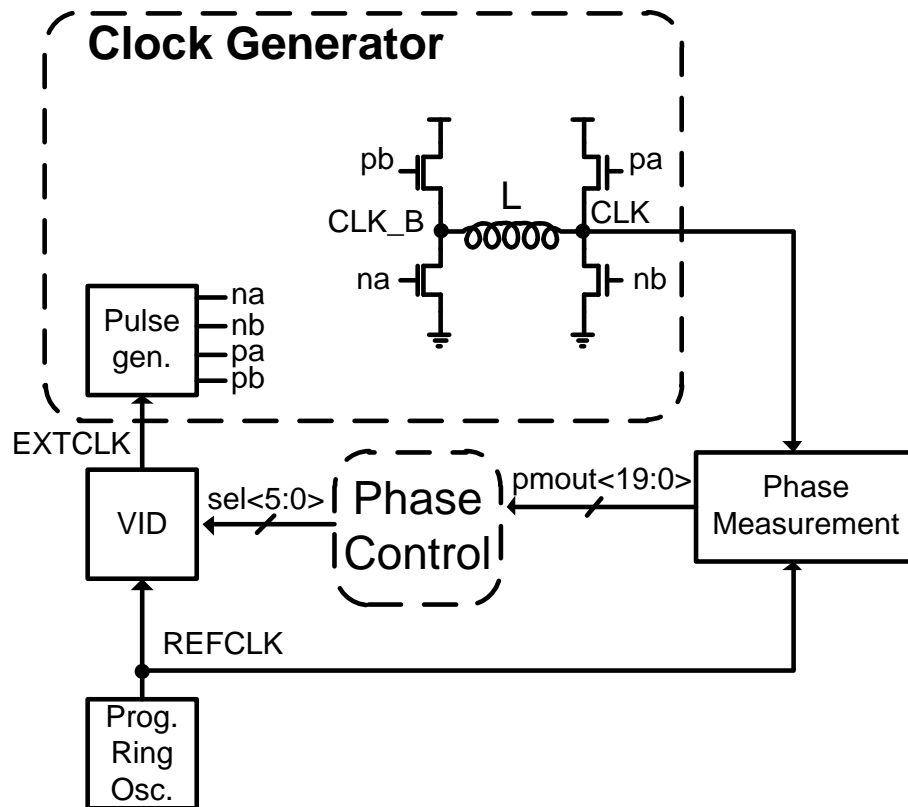


Figure 6.6: Overview of the resonant clock alignment circuitry.

Fig. 6.6 shows the resonant clock phase alignment circuitry, which adjusts the phase of the resonant clock to be synchronized with the conventional external reference clock. The phase measurement unit captures one of the resonant clock phases

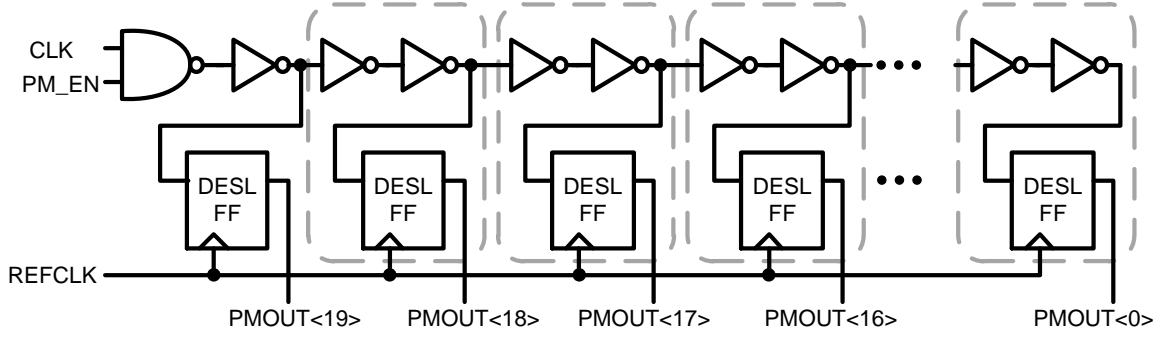


Figure 6.7: Phase measurement circuitry.

using an external clock source, REFCLK, generated by the programmable ring oscillator, and provides a 20-bit output. The pulse generator of the resonant clock generator is driven by the Variable Insertion Delay (VID) unit, which inserts a programmable amount of delay on its output, EXTCLK. The Phase Control unit looks at the output of the phase measurement unit and generates control signals for the VID unit, effectively creating a delay lock loop between the resonant clocks and the external clock source, REFCLK. In the test-chip, the phase control unit is implemented off chip with scannable control signals, but it can be implemented on-chip using a look-up table or a finite state machine in the future. Having the ability to narrow the phase difference between the two clocks enables the deployment of resonant clocking to critical power hungry datapaths instead of the whole design, thus reducing risk while achieving significant power savings.

Fig. 6.7 shows implementation details of the phase measurement circuitry, which is basically a time-to-digital converter (TDC). One of the resonant clocks is fed to a delay line consisting of a chain of inverters with a NAND gate at the front to gate off the input when the alignment circuitry is not used. DESL flip-flops constructed using two DESL inverter gates capture the delayed resonant clock from different positions of the delay line with a delay difference of two inverters between any two consecutive flip-flops, achieving a resolution of 31ps. Having two inverters between two consecutive

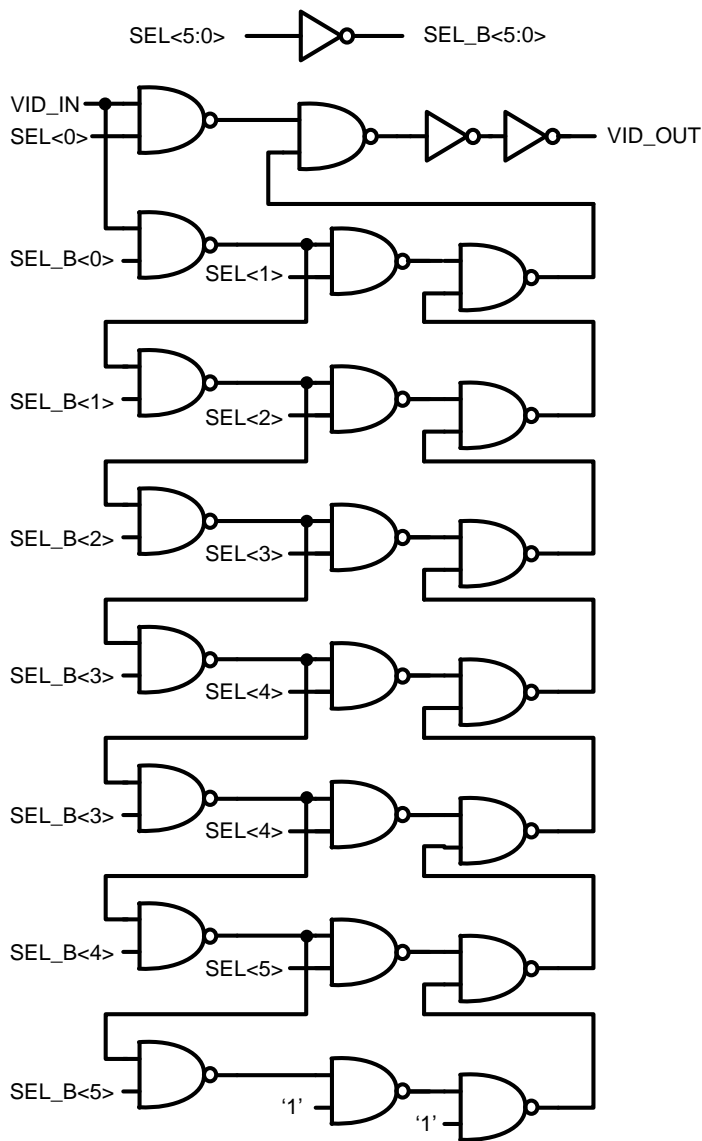


Figure 6.8: Variable insertion delay block for resonant clock alignment.

flip-flops reduces the resolution of the TDC, but also reduces the chance of sparkle or bubble errors induced by process variations. All the flip-flops are synchronized by the same phase of the external clock source, REFCLK, and the output of these flip-flops shows the phase difference between the two clocks. Assuming clock jitter is small on REFCLK, the information about the phase difference can be use to adjust the phase of the resonant clocks until the phase difference is minimized.

Fig. 6.8 shows implementation details of the VID unit, which inserts a certain amount of delay depending on the digital 6-bit thermometer-coded control signals. The VID unit is constructed using only NAND2 and INV gates, and it is therefore possible to be constructed using a normal standard cell library. Since additional delay is inserted by configuring the number of gates that the input has to travel through, the amount of delay inserted on the output is monotonic with the configuration control signals. This monotonic property is the main advantage of this type of VID unit since it keeps the control loop of the phase alignment circuit stable. When incrementing the controlled signals, SEL, the delay on the output increases by the delay of two NAND gates, resulting in a minimum granularity of 52ps. One other advantage of this VID unit is that its power dissipation depends on the amount of the delay inserted, making it more energy-efficient than other VID implementations. When neglecting the two inverters used to increase drive strength, the minimal configurable delay is only two NAND gates, which has a much lower delay overhead compared to other VID implementations.

6.2.2 Conventional Clock Network

Fig. 6.9 shows the clock distribution network in the conventional-clock FPU core. The conventional clock network has 4 stages of clock drivers. The first stage amplifies the output from the programmable ring oscillator and propagates its output to 11 distributed clock inverters along the exponent side of the conventional clock FPU. These distributed clock inverters then drive 114 local clock buffers located in the 4-bit clock bay between the 10-bit exponent and the 52-bit fraction datapath through a $0.84\mu\text{m}$ clock spine consisting of M3 and M5 layers. Two types of local clock buffers are used to generate a two-phase clock with its phases at 180 degrees: one with 4 stages of logic and another with 5 stages of logic. Outputs of the local clock buffers are propagated to fan-out DESL gates through a $0.3\mu\text{m}$ wide M2 wire. To reduce

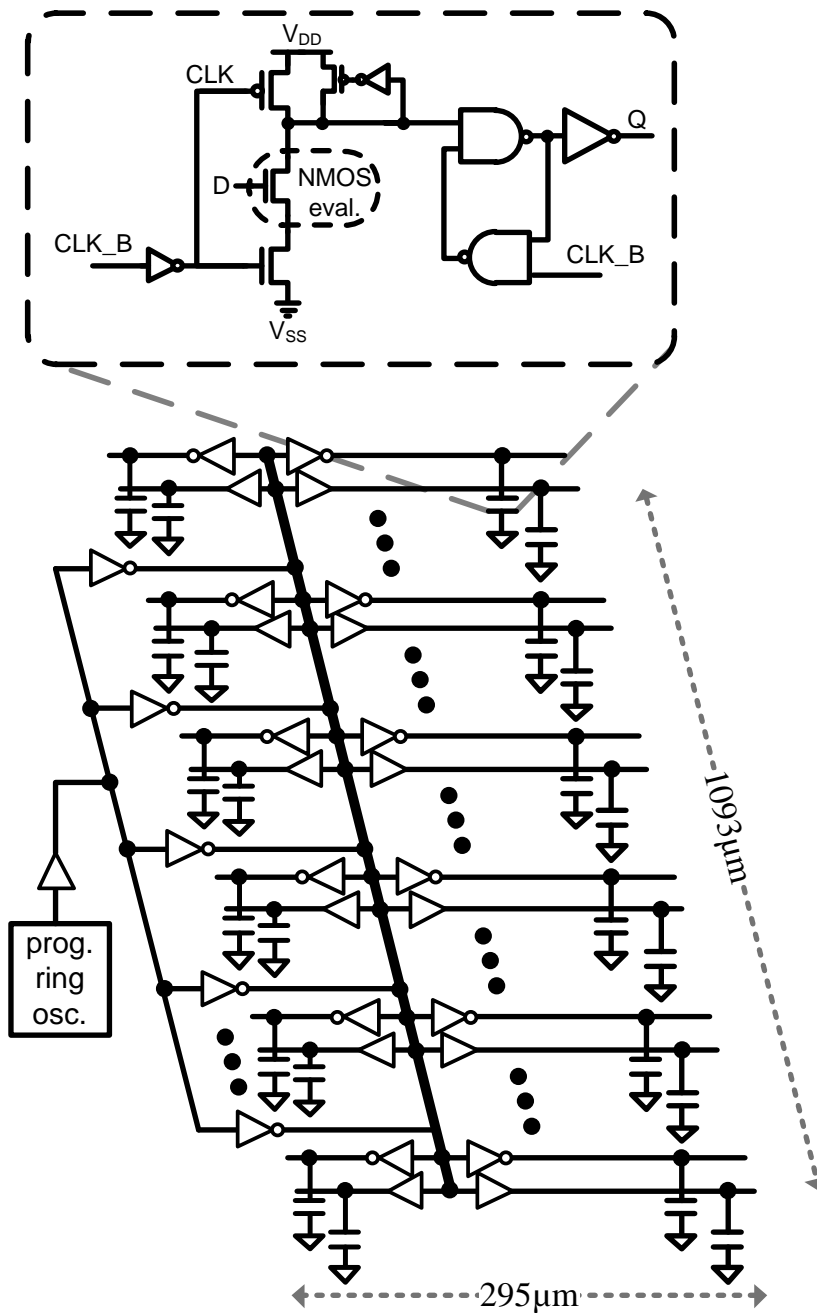


Figure 6.9: Conventional clock distribution network.

unnecessary clock capacitance, the local clock distribution network is designed to be much thinner than the resonant clock one and is similar to other conventional clock designs. Extensive simulations have been performed to ensure that the propagation

delays of two different local clock buffers are the same under the same clock load. Each local clock distribution network has been extracted, and six power levels have been created for each local clock buffer, so that both the clock skew and the clock slew of each local clock distribution would meet our design requirements. The final stage of clock inverters located in each DESL gate generates its own version of the negated clock phase to drive the dynamic evaluation stage, ensuring that static latches always capture the correct state before precharging. The addition of an inverter to every DESL gate contributes to the height difference between the two FPU cores.

6.3 Measurement Results

This section gives measurement results from the experimental evaluation of both the resonant and the conventional clock FPU cores, demonstrating the relative energy efficiency of resonant clocking. It also compares the resonant clock FPU core with other state-of-the-art reduced-latency FPUs to demonstrate the advantages of combining resonant clocking with DESL logic.

Fig. 6.10 shows a die microphotograph with the resonant FPU on the left and the conventional FPU on the right. The two FPU cores have been fabricated in a 90nm RVT low-power 9-metal technology with a nominal supply of 1.2V. The resonant FPU with the clock generation occupies $904\mu\text{m} \times 313.6\mu\text{m} = 0.283\text{mm}^2$. Including the inductor, it occupies 0.319mm^2 . The conventional FPU occupies $957.3\mu\text{m} \times 295.6\mu\text{m} = 0.283\text{mm}^2$.

Figs 6.11 shows the build-in self-test (BIST) circuitry around each FPU core. The three input operands are generated by 3 independent linear feedback shift registers (LFSR), which are initialized by scannable flip-flops. Since the input data are generated from LFSRs, they have a switching activity of 0.5, i.e., the highest possible, which helps to exercise the FPU core under the maximum power scenario. The result of the FPU is compressed in the signature analyzer. Correct operation has been

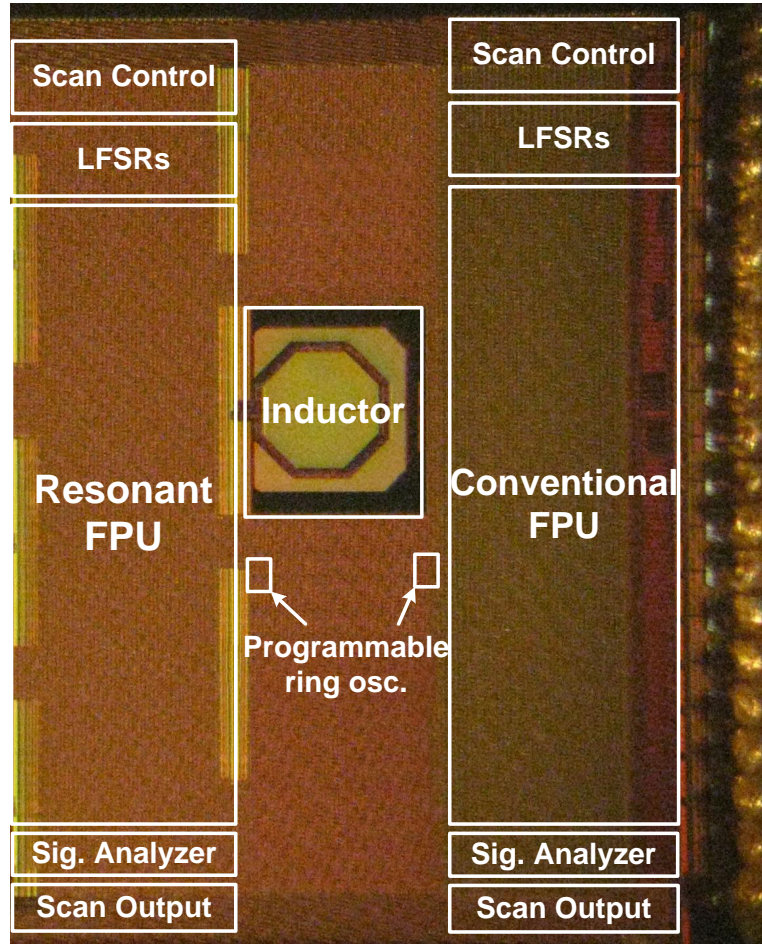


Figure 6.10: Die microphotograph.

verified by scanning out the 32-bit result in the signature analyzer or by observing the low-frequency single-bit signature output.

Fig. 6.12 shows measured power consumption versus operating frequency for the resonant FPU core and its conventional counterpart, including BIST circuitry. Correct operation has been validated from 1.67GHz to 2.07GHz with supply voltage ranging from 1.25V to 1.45V for the resonant FPU and from 1.17V to 1.35V for the conventional FPU. At its resonant frequency of 1.81GHz, the resonant FPU consumes 334mW with a 1.32V supply, yielding 31.5% lower total power than the conventional FPU. For the same operating frequency, the resonant FPU requires about 100mV higher supply than its conventional counterpart, due to the narrower effective widths

of the sinusoidal clocks. Power measurements from both cores are obtained with 50% switching activity on all LFSRs generating the three operands. In a typical application, the input switching activity is expected to be in the 10-20% range, resulting in a relatively larger percentage of clock power and, therefore, even greater relative power savings from resonant clocking.

The power breakdown of the two FPUs when running at 1.81GHz is shown in Fig. 6.13. The clock and logic power of the resonant-clock FPU are obtained from measurements using separate power supplies, V_{DD} and V_{CK} . Since the resonant clock generator circuits are located in dedicated columns on the two sides of the resonant FPU, the distribution cost of routing additional power supplies for the clock circuitry is low. In the resonant FPU, the logic power measured from the V_{DD} supply is 246mW, and the clock power measured from the V_{CK} supply is 88mW. In the conventional FPU, both clock and logic circuitry are powered by a single supply, V_{DD} , since the routing overhead of having separate supplies for the logic and the clock circuitry distributed across the chip would be prohibitively high. Based on

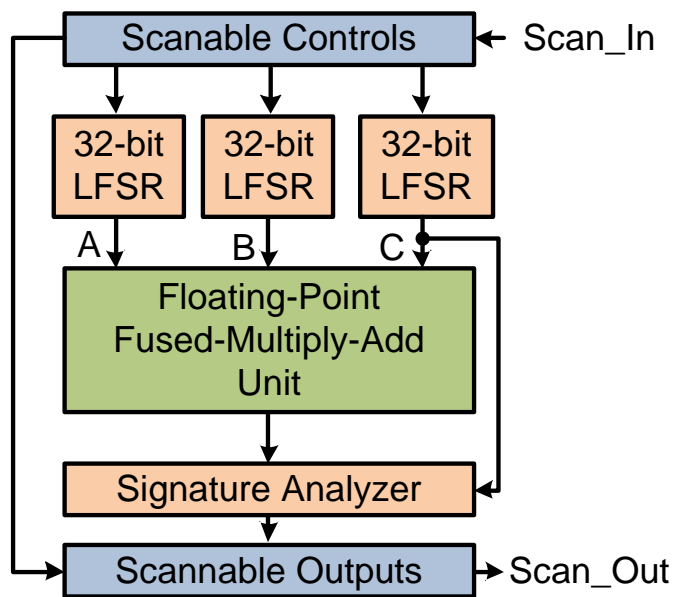


Figure 6.11: Built-in self-test circuitry around FPU cores.

measurements of the V_{DD} supply, the total power of the conventional FPU is 488mW. The power breakdown of the conventional FPU is obtained by running transistor-level simulations on an extracted netlist, yielding 262mW and 226mW for clock and logic power, respectively. Notice that the logic power of the conventional FPU is lower than that of the resonant FPU, due to the conventional supply voltage being 100mV lower. Compared to its conventional counterpart, the resonant FPU yields 66.4% reduction in clock power.

Fig. 6.14 shows the energy breakdown of both FPUs versus operating frequency. The minimum clock energy consumption of 48.65pJ is reached at 1.81GHz, indicating the natural frequency of the resonant FPU. At this frequency, the resonant FPU consumes 31.5% less energy than the conventional FPU, reaching its highest energy efficiency of 10.82 GFLOPS/W, as shown in the graph of Fig. 6.15. Driving 32pF

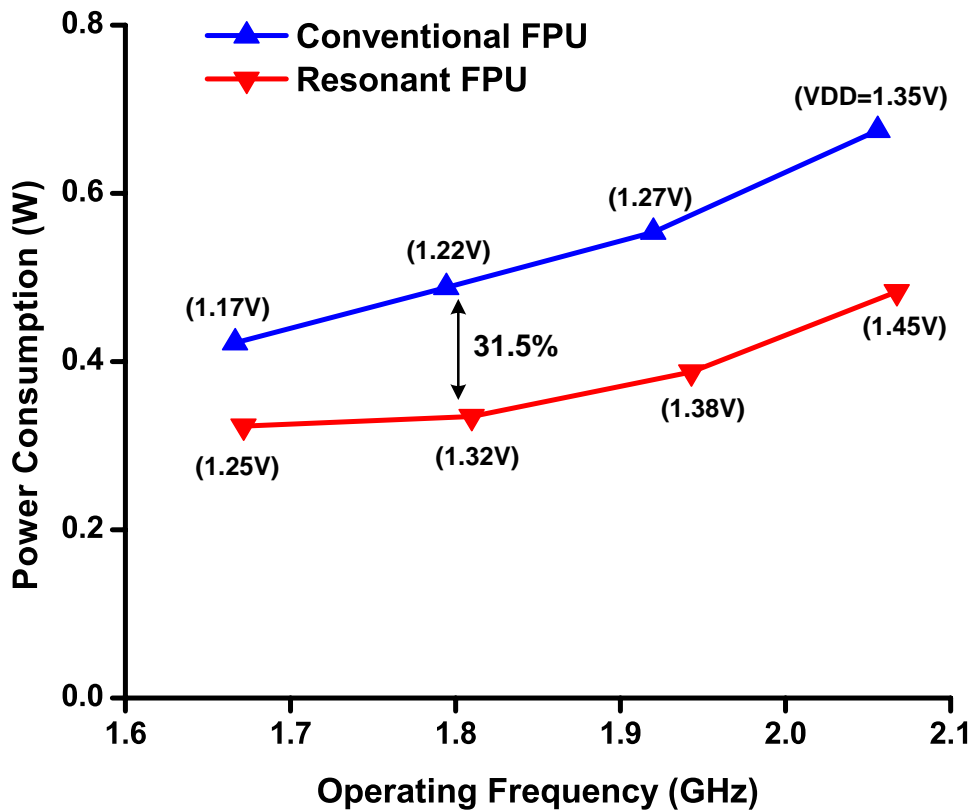


Figure 6.12: Measured power consumption versus clock frequency.

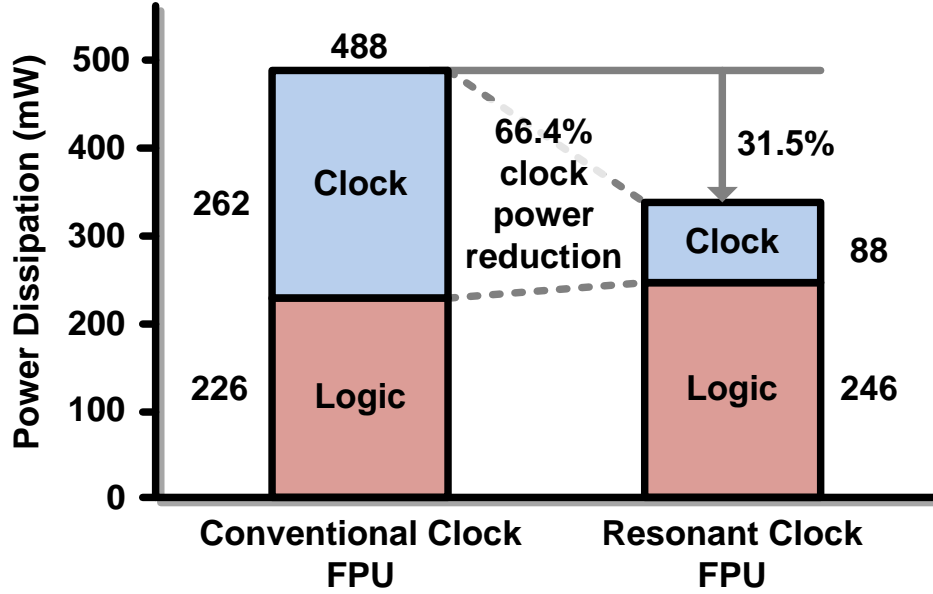


Figure 6.13: Breakdown of power consumption at resonant frequency of 1.81GHz. Clock and logic power for the resonant FPU are obtained from measurements, and total power for the conventional FPU is obtained from measurements. Clock and logic power breakdown for the conventional FPU is derived from transistor-level simulations.

Technology	90nm 9M LP (RVT)
Nominal voltage	1.2V
Transistor Count	300K
FPU Core Area	0.283 mm ²
FPU Core + Inductor Area	0.319 mm ²
FPU Core + Inductor + BIST Area	0.360 mm ²
Overall Latency	64 FO4
Resonant Frequency	1.81 GHz
Energetics @ Resonance	
Supply Voltage(V)	1.32
Clock Network Efficiency	55.5%
Total / Logic / Clock Power Diss. (mW)	334 / 246 / 88
Total / Logic / Clock Energy (pJ)	184.8 / 136.1 / 48.7
GFLOPS/W	10.82
Input Switching Activity	0.5

Table 6.1: FPU performance summary table.

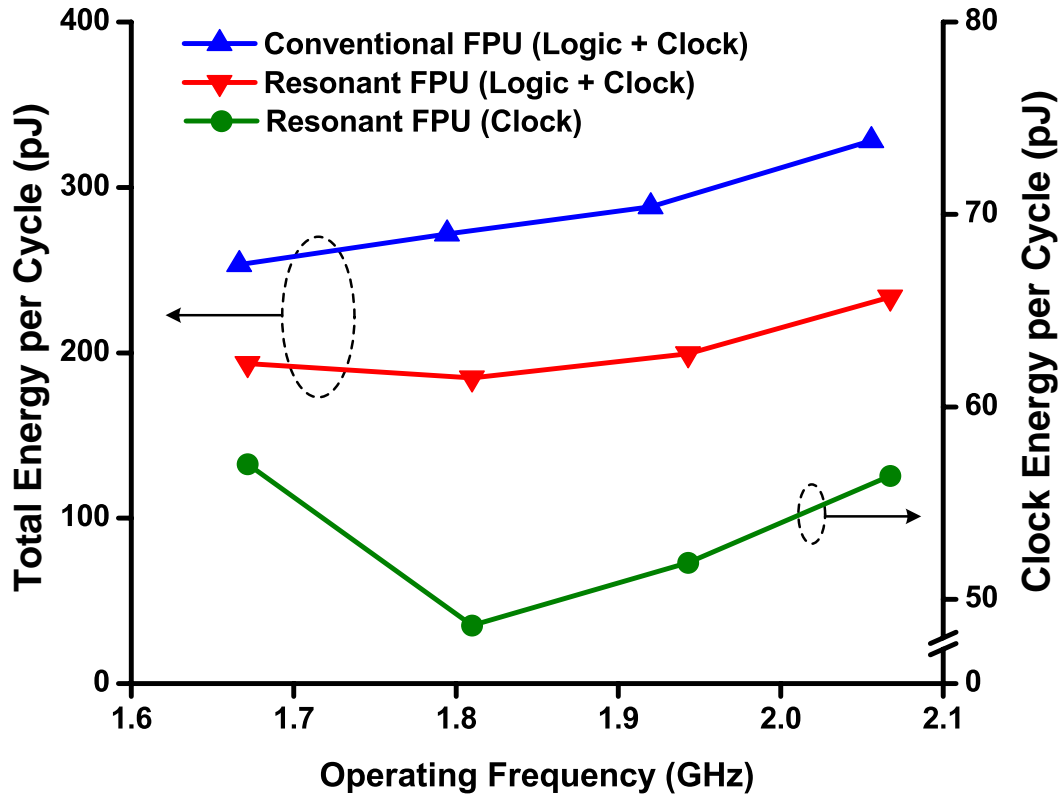


Figure 6.14: Measured energy per cycle versus clock frequency.

of clock load per phase, the resonant FPU recovers 55.5% of the CV^2f clock power, yielding a quality factor Q of approximately 2.25 for the resonant clock system. Chip performance is summarized in Table 6.1.

Resonant frequency is predicted from extractions with high accuracy. Specifically, inductance L is extracted using a commercial 2.5D Maxwell equation solver, and capacitance C of the clock network is extracted using a commercial RC extraction tool. The extracted parameters yield a resonant frequency of 1.97GHz (not including the damping factor), which is 6% away from the 1.81GHz minimum energy point in Fig. 6.14.

Table 6.2 compares our resonant FPU with two other state-of-the-art reduced-latency FPUs: a FPU from Intel [84] and a FPU from IBM [82]. All three FPUs implement reduced-latency single-precision fused-multiply-add floating architectures and are fabricated in 90nm technologies with supply voltage ranging from 1.0 to

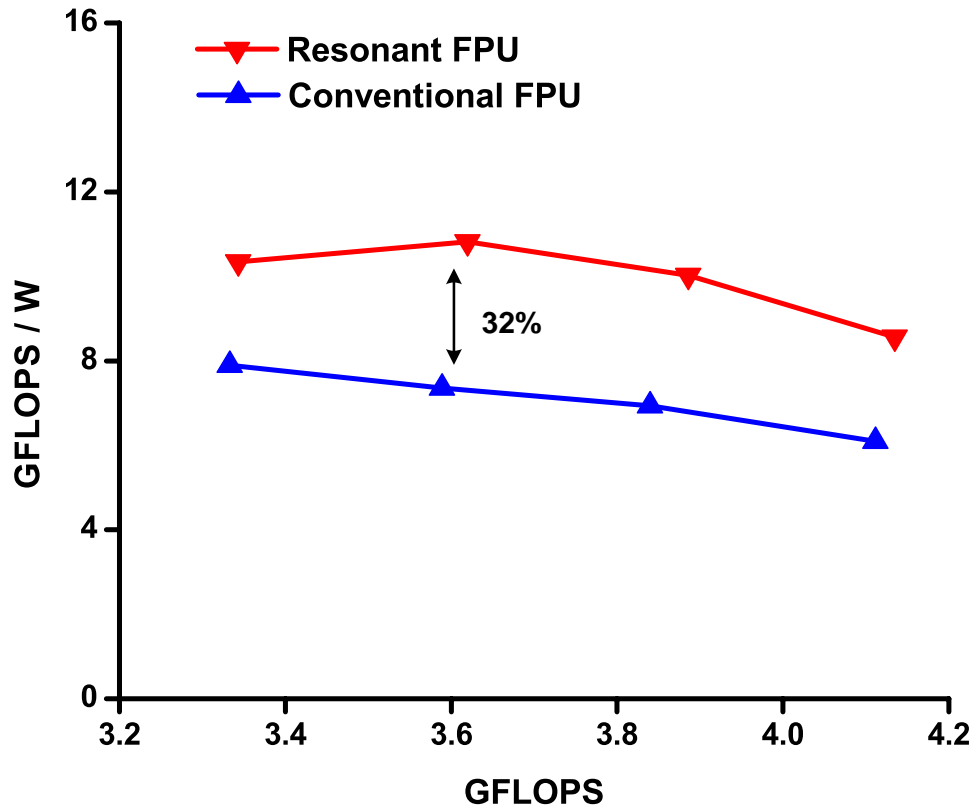


Figure 6.15: Measured chip energy efficiency.

1.2V using either semi-custom or full-custom design flows. They all support only the round-toward-zero rounding mode and drop support for denormalized number. With an overall latency of 64 FO4, our resonant FPU achieves the lowest overall latency among three FPUs, which improves on the previously best result of 66 FO4. Moreover, in spite of the inductor area overhead, the resonant FPU occupies the smallest area among the three competing FPUs.

The measurement setups for the three FPUs are quite distinct, preventing a direct comparison of energy efficiency data. The inputs of our resonant FPU are generated by an integrated LFSR, yielding a switching activity of 0.5, the highest switching activity possible. The Intel FPU has dedicated FIFOs for input pattern generation, also yielding relatively high switching activities. However, the IBM FPU is part of a large microprocessor. Since a maximum-power program is executed to extrapolate power dissipation in the IBM FPU, its switching activity and utilization rate are

unknown, and the resulting data do not lend themselves to a direct comparison. Between the remaining two continuously data streaming FPUs, the resonant clock FPU has the highest energy efficiency in GFLOPS/W, 2.1X higher than the Intel FPU, achieving the highest energy efficiency for continuously data streaming FPUs.

6.4 Summary

This chapter discuss the resonant-clocked FPU core implemented using DESL. Our test-chip measurement results show that our resonant FPU core breaks new ground in several metrics. Relying on the high performance of dynamic logic and the time borrowing property of transparent latches, our FPU core functions correctly at operating frequencies up to 2.07GHz. At its most energy-efficient operating frequency of 1.81GHz, it dissipates 334mW, achieving an energy efficiency of 10.82GFLOPS/W. Even including the inductor area, it achieves the smallest footprint compared to other state-of-the-art reduced-latency FPUs. Moreover, with an overall latency of 64 FO4, it achieves the lowest overall latency among state-of-the-art reduced-latency FPUs.

Paper	This Work	[84]	[82]
Design Type	32b FMA FPU	32b FMA FPU	32b FMA FPU
Rounding Mode Support	round-toward-zero	round-toward-zero	round-toward-zero
Denormalized Number Support	No	No	No
FO4 per cycle	8	15	11
Number of Cycle	8	11	6
Overall Latency	64 FO4	165 FO4	66 FO4
Technology	90nm CMOS LP	90nm CMOS	90nm CMOS
Design Style	semi-custom	full-custom	full-custom
Area (mm ²)	0.319	0.88	0.325
Nominal Supply Voltage	1.2	1.3	1.0
Operating Frequency (GHz)	1.81	3.1	4.0
Power Dissipation (mW)	334	1200	350
Energy Efficiency (GFLOPS/W)	10.82	5.17	22.86
Input Switching Activity	0.5	High	N/A

Table 6.2: FPU performance comparison table.

CHAPTER 7

Conclusions and Future Directions

This chapter summarizes our contributions to the areas of charge-recovery logic and resonant-clock dynamic logic. Based on our experiences with the design and evaluation of two test-chips, we present advantages of our proposed energy-recovery techniques, and discuss challenges faced when using these techniques. We also discuss future research directions aimed at the broader adoption of these techniques in mainstream commercial designs.

7.1 Enhanced Boost Logic

In this dissertation, we have presented Enhanced Boost Logic (EBL), an energy-efficient charge-recovery logic family that exhibits low latency overheads. EBL uses an aggressively-scaled near-threshold supply to perform logic evaluation with low energy consumption. Increased gate overdrive enables high-speed operation or, alternatively, the single-gate realization of complex logic functions, both of which contribute to low overall latency.

To demonstrate the performance and energy advantages of EBL, we have designed a 14-tap 8-bit FIR filter test-chip in a $0.13\mu\text{m}$ CMOS process. Unlike previously-published charge-recovery circuitry, in which overall latency is typically an order of magnitude higher than static CMOS designs [7, 88], the EBL-based FIR filter achieves

an overall latency overhead of 1.5 cycles compared to a high-performance FIR filter that we have designed using a standard cell library. In post-layout simulations, the EBL-based FIR filter running in self-resonant mode consumes 21% to 34% less energy than its voltage-scaled static CMOS counterpart from 466MHz to 800MHz while incurring a 37% area overhead due to the on-chip inductor required. With frequency scaling circuitry enabled and a fixed 3nH integrated inductor, simulation results show that the EBL FIR consumes 17% less energy at its resonant frequency of 466MHz, and consumes less energy between 440MHz and 565MHz even when forced to run off-resonance.

Fabricated in a 0.13 μ m bulk-silicon process with regular threshold voltage at 0.4V, the FIR-filter test-chip functions correctly from 365MHz to 600MHz using a 3nH on-chip symmetric spiral inductor. Clock drivers for self-resonant operation are fully integrated and distributed across the entire clock network. To support frequency-scaled operation, the clock generator includes an additional pair of small drivers that are located between the inductor and the FIR core. At its resonant frequency of 466MHz, the FIR filter is at its most energy-efficient point, dissipating 39.1mW and recovering 45% of the energy supplied through its clock generator. The corresponding figure of merit equals 93.6nW/MHz/Tap/InBit/CoeffBit.

7.2 Resonant-Clocked Designs

In this dissertation, we have also presented a reduced-latency single-precision FPU designed using Dynamic Evaluation Static Latch (DESL) logic, a high performance logic family that can be clocked either by a two-phase resonant clock or a conventional "square-wave" clock. DESL logic takes advantage of the speed of dynamic evaluation and time borrowing in static latches to achieve high performance. The static latch in DESL also reduces the complexity typically associated with the clock generation of multi-stage dynamic logic. When synchronized by a two-phase resonant clock,

DESL achieves performance levels commensurate of dynamic logic while operating with significantly reduced power consumption.

Fabricated in a 90nm low-power CMOS process, the resonant FPU functions correctly from 1.67GHz to 2.07GHz, demonstrating that resonant clocking can yield significant reductions in clock power at multi-GHz clock frequencies. At its resonant frequency of 1.81GHz, the resonant FPU achieves 10.82GFLOPS/W, yielding a 66.4% reduction in clock power and a 32% improvement in GFLOPS/W compared to its conventional-clock counterpart implemented side-by-side on the same silicon die. Relying on circuit, logic, and architectural optimizations, the resonant-clock FPU breaks new ground along several metrics. Specifically, with a total area of 0.391mm² including the on-chip inductor, the resonant clock FPU occupies the smallest footprint among competing state-of-the-art reduce-latency FPUs. Moreover with an overall latency of 64 FO4, it achieves the shortest overall latency among state-of-the-art reduced latency FPUs. Delivering 10.82GFLOPS/W, this resonant-clock FPU achieves the highest energy efficiency among continuously state-of-the-art data-streaming FPUs.

7.3 Future Directions

In this section, we discuss some of the future challenges in the implementation of charge-recovery logic and resonant clock designs. Even though the two energy-recovery techniques are slightly different, there are many similarities in future directions between the two of them.

Smaller and Higher Quality Inductors

Both energy-recovery techniques would benefit from small inductors with high quality factor. One possible way to improve the quality factor is by filling the core of the inductor with magnetic materials [89], which facilitates the fabrication of smaller

inductors with higher quality factor. Our EBL test-chip attempted to place the inductor over our FIR core, but the degradation in quality factor of the inductor eliminates all power savings. In the future, a gridless power distribution under the inductor and 3D integration of different die would hopefully reduce area overheads associated with integrated inductors. We have some initial evaluations of inductors over gridless power distribution and the results are promising. Attempts have also been made to integrate inductors with circuits using 3D stacking [89].

Multiple clock domain designs

Another challenge faced by energy-recovery techniques is the need to form a design methodology to generate, distribute, and synchronize the clock in multiple-clock-domain designs. All experimental evaluations in our test-chips have been implemented using a single clock domain. However, as designs get larger or faster, multiple-clock-domain designs become necessary to set the natural frequency of the LC oscillation to be near the desired operating frequency. Some work has already applied adaptive techniques to synchronize resonant clocks between two resonant clock domains [90]. More work is needed to expand synchronization methods to clock distribution networks with even more clock domains, reducing both clock jitter and power. This problem will become more challenging when clock imbalance, and voltage scaling and frequency scaling are put in the mix.

Design Automation

Another challenge faced by both energy-recovery techniques is the amount of time spend in manually pipelining the logic, placing gates, and routing between the gates. To make these techniques part of the mainstream design methodology, more design automation tools are needed.

In the area of charge-recovery logic, many design automation tools are still under

development. During the logic synthesis process, a tool is needed to take advantage of the state-intensive nature of charge-recovery logic while reducing the number of buffers due to wave pipelining. During the routing process, the charge-recovery logic friendly router should be able to balance the capacitive load between the dual-rail outputs, which would be helpful in reducing clock jitter. The router should also automatically use wide wire for long routes so that the resistance in the resonating system is minimized.

In the area of resonant-clocked designs, the main challenge is in the logic synthesis process. Since DESL is limited to two logic operations per cycle, a DESL friendly synthesis tool must embrace these constraints and produce a netlist with a low overall latency. Another area of interest is the automated generation of the clock network. This problem is complicated by multi-clock-domain designs, where the design automation tool needs to find the optimal number of inductors with the optimal inductance value to reduce clock jitter and power in a given clock distribution network.

With the measurement results from our test-chips, we show that energy recovery techniques are one step closer to become part of the mainstream design flows. In the near future, with the ability to place inductor over circuits, more advancement in design automation tools, and additional development in multi-domain clock networks, energy-recovery circuitry should be a promising option for low-power design.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] 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, pp. 256–268, October 1974.
- [2] Y. Taur, "CMOS design near the limit of scaling," *IBM Journal of Research and Development*, vol. 46, pp. 213–222, March 2002.
- [3] G. Shahidi, "Evolution of CMOS technology at 32nm and beyond," in *IEEE Custom Integrated Circuits Conference*, pp. 413–416, September 2007.
- [4] "International technology roadmap for semiconductors (ITRS): 2009 update," December 2009.
- [5] D. Maksimovic and V. Oklobdzija, "Integrated power clock generators for low energy logic," in *Recordings of the 26th Annual IEEE Power Electronics Specialists Conference*, vol. 1, pp. 61–67, June 1995.
- [6] W. Athas, L. Svensson, and N. Tzartzanis, "A resonant signal driver for two-phase, almost-non-overlapping clocks," in *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 129–132, May 1996.
- [7] V. S. Sathe, J.-Y. Chueh, and M. C. Papaefthymiou, "Energy-efficient GHz-class charge-recovery logic," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 38–47, January 2007.
- [8] J. C. Kao, W.-H. Ma, and M. Papaefthymiou, "A charge-recovery 600MHz FIR filter with 1.5-cycle latency overhead," in *Proceedings of the 35th European Solid-State Circuits Conference*, pp. 160–163, September 2009.
- [9] J. C. Kao, W.-H. Ma, and M. Papaefthymiou, "Energy-efficient low-latency 600MHz FIR with high-overdrive charge-recovery logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, to appear, 2011.
- [10] J. C. Kao, W.-H. Ma, S. Kim, and M. Papaefthymiou, "2.07 GHz floating-point unit with resonant-clock precharge logic," in *IEEE Asian Solid-State Circuits Conference*, pp. 160–163, November 2010.
- [11] C. Ziesler, J. Kim, V. Sathe, and M. Papaefthymiou, "A 225 MHz resonant clocked ASIC chip," in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design*, pp. 48–53, August 2003.

- [12] A. Ishii, J. Kao, V. Sathe, and M. Papaefthymiou, “A resonant-clock 200MHz ARM926EJ-STM microcontroller,” in *Proceedings of the 29th European Solid-State Circuits Conference*, pp. 356–359, September 2009.
- [13] V. Sathe, J. Kao, and M. Papaefthymiou, “RF2: A 1GHz FIR filter with distributed resonant clock generator,” in *IEEE Symposium on VLSI Circuits*, pp. 44–45, June 2007.
- [14] V. Sathe, J. Kao, and M. Papaefthymiou, “Resonant-clock latch-based design,” *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 864–873, April 2008.
- [15] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM Journal of Research and Development*, vol. 5, pp. 183–191, July 1961.
- [16] R. Landauer, “Wanted: a physically possible theory of physics,” *IEEE Spectrum*, vol. 4, pp. 105–109, September 1967.
- [17] C. H. Bennett and R. Landauer, “The Fundamental Physical Limits of Computation,” *Scientific American*, vol. 253, pp. 48–56, July 1985.
- [18] E. Fredkin and T. Toffoli, “Conservative logic,” *International Journal of Theoretical Physics*, vol. 21, pp. 219–253, April 1982.
- [19] W. Pan and M. Nalasan, “Reversible logic,” *IEEE Potentials*, vol. 24, pp. 38–41, February-March 2005.
- [20] R. P. Feynman, “Quantum mechanical computers,” *Foundations of Physics*, vol. 16, pp. 507–531–531, June 1986.
- [21] T. Toffoli, “Reversible computing,” in *Automata, Languages and Programming* (J. de Bakker and J. van Leeuwen, eds.), vol. 85 of *Lecture Notes in Computer Science*, pp. 632–644, 1980.
- [22] A. Peres, “Reversible logic and quantum computers,” *Physical Review A*, vol. 32, pp. 3266–3276, December 1985.
- [23] P. Kerntopf, “A new heuristic algorithm for reversible logic synthesis,” in *Proceedings of the 41st Design Automation Conference*, pp. 834–837, 2004.
- [24] W. Athas and L. Svensson, “Reversible logic issues in adiabatic CMOS,” in *Proceedings of Workshop on Physics and Computation*, pp. 111–118, November 1994.
- [25] S. G. Younis and T. F. Knight, Jr., “Practical implementation of charge recovering asymptotically zero power CMOS,” in *Proceedings of the 1993 Symposium on Research on Integrated Systems*, (Cambridge, MA, USA), pp. 234–250, MIT Press, 1993.

- [26] S. G. Younis, *Asymptotically Zero Energy Computing Using Split-Level Charge Recovery Logic*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1994.
- [27] J. Lim, D.-G. Kim, and S.-I. Chae, “A 16-bit carry-lookahead adder using reversible energy recovery logic for ultra-low-energy systems,” *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 898–903, June 1999.
- [28] A. Dickinson and J. Denker, “Adiabatic dynamic logic,” in *IEEE Custom Integrated Circuits Conference*, pp. 282–285, May 1994.
- [29] A. Dickinson and J. Denker, “Adiabatic dynamic logic,” *IEEE Journal of Solid-State Circuits*, vol. 30, pp. 311–315, March 1995.
- [30] A. Kramer, J. S. Denker, B. Flower, and J. Moroney, “2nd order adiabatic computation with 2N-2P and 2N-2N2P logic circuits,” in *Proceedings of the 1995 International Symposium on Low Power Design*, pp. 191–196, August 1995.
- [31] Y. Moon and D.-K. Jeong, “Efficient charge recovery logic,” in *IEEE Symposium on VLSI Circuits*, pp. 129–130, June 1995.
- [32] Y. Moon and D.-K. Jeong, “An efficient charge recovery logic circuit,” *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 514–522, April 1996.
- [33] V. Oklobdzija, D. Maksimovic, and F. Lin, “Pass-transistor adiabatic logic using single power-clock supply,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, pp. 842–846, October 1997.
- [34] A. Vetuli, S. Pascoli, and L. Reyneri, “Positive feedback in adiabatic logic,” *Electronics Letters*, vol. 32, pp. 1867–1869, September 1996.
- [35] D. Maksimovic and V. G. Oklobdzija, “Clocked CMOS adiabatic logic with single AC power supply,” in *Proceedings of the 21th European Solid-State Circuits Conference*, pp. 370–373, September 1995.
- [36] D. Maksimovic, V. Oklobdzija, B. Nikolic, and K. Current, “Clocked CMOS adiabatic logic with integrated single-phase power-clock supply: experimental results,” in *Proceedings of 1997 International Symposium on Low Power Electronics and Design*, pp. 323–327, August 1997.
- [37] S. Kim and M. Papaefthymiou, “True single-phase energy-recovering logic for low-power, high-speed VLSI,” in *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, pp. 167–172, August 1998.
- [38] S. Kim and M. Papaefthymiou, “Single-phase source-coupled adiabatic logic,” in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, pp. 97–99, August 1999.

- [39] S. Kim, C. Ziesler, and M. Papaefthymiou, "Design, verification, and test of a true single-phase 8-bit adiabatic multiplier," in *Proceedings of the 2001 Conference on Advanced Research in VLSI*, pp. 42–58, March 2001.
- [40] S. Kim, C. Ziesler, and M. Papaefthymiou, "A true single-phase 8-bit adiabatic multiplier," in *Proceedings of the 38th Design Automation Conference*, pp. 758–763, June 2001.
- [41] S. Kim, C. Ziesler, and M. Papaefthymiou, "A true single-phase energy-recovery multiplier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, pp. 194–207, April 2003.
- [42] S. Kim, C. Ziesler, and M. Papaefthymiou, "Charge-recovery computing on silicon," *IEEE Transactions on Computers*, vol. 54, pp. 651–659, June 2005.
- [43] V. Sathe, M. Papaefthymiou, and C. Ziesler, "A GHz-class charge recovery logic," in *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, pp. 91–94, August 2005.
- [44] V. Sathe, M. Papaefthymiou, and C. Ziesler, "Boost logic : a high speed energy recovery circuit family," in *IEEE Computer Society Annual Symposium on VLSI*, pp. 22–27, May 2005.
- [45] V. Sathe, J.-Y. Chueh, and M. Papaefthymiou, "A 1.1GHz charge-recovery logic," in *IEEE International Solid-State Circuits Conference*, pp. 1540–1549, February 2006.
- [46] J. Wood, S. Lipa, P. Franzon, and M. Steer, "Multi-gigahertz low-power low-skew rotary clock scheme," in *IEEE International Solid-State Circuits Conference*, pp. 400–401, 470, February 2001.
- [47] J. Wood, T. Edwards, and S. Lipa, "Rotary traveling-wave oscillator arrays: a new clock technology," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1654–1665, November 2001.
- [48] F. O'Mahony, C. Yue, M. Horowitz, and S. Wong, "10GHz clock distribution using coupled standing-wave oscillators," in *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 428–504, February 2003.
- [49] F. O'Mahony, C. Yue, M. Horowitz, and S. Wong, "Design of a 10GHz clock distribution network using coupled standing-wave oscillators," in *Proceedings of the 40th Design Automation Conference*, pp. 682–687, June 2003.
- [50] F. O'Mahony, C. Yue, M. Horowitz, and S. Wong, "A 10-GHz global clock distribution using coupled standing-wave oscillators," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 1813–1820, November 2003.

- [51] W. Athas, N. Tzartzanis, L. Svensson, L. Peterson, H. Li, P. Wang, and W.-C. Liu, "AC-1: a clock-powered microprocessor," in *Proceedings of the 1997 International Symposium on Low Power Electronics and Design*, pp. 328–333, August 1997.
- [52] W. Athas, N. Tzartzanis, L. Svensson, and L. Peterson, "A low-power microprocessor based on resonant energy," *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 1693–1701, November 1997.
- [53] C. L. Seitz, A. H. Frey, S. Mattisson, S. D. Rabin, D. A. Speck, and J. L. A. van de Snepscheut, "Hot-Clock nMOS," *Proceedings of the 1985 Chapel Hill Conference on VLSI*, pp. 1–17, 1985.
- [54] C. Ziesler, J. Kim, M. Papaefthymiou, and S. Kim, "Energy recovery design for low-power ASICs," in *IEEE International Systems-on-Chip (SOC) Conference*, pp. 424–427, September 2003.
- [55] A. Drake, K. Nowka, T. Nguyen, J. Burns, and R. Brown, "Resonant clocking using distributed parasitic capacitance," in *IEEE Custom Integrated Circuits Conference*, pp. 647–650, September 2003.
- [56] A. Drake, K. Nowka, T. Nguyen, J. Burns, and R. Brown, "Resonant clocking using distributed parasitic capacitance," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1520–1528, September 2004.
- [57] M. Hansson, B. Mesgarzadeh, and A. Alvandpour, "1.56 GHz on-chip resonant clocking in 130nm CMOS," in *IEEE Custom Integrated Circuits Conference*, pp. 241–244, September 2006.
- [58] M. Cooke, H. Mahmoodi-Meimand, and K. Roy, "Energy recovery clocking scheme and flip-flops for ultra low-energy applications," in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design*, pp. 54–59, August 2003.
- [59] H. Mahmoodi, V. Tirumalashetty, M. Cooke, and K. Roy, "Ultra low-power clocking scheme using energy recovery and clock gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, pp. 33–44, January 2009.
- [60] V. Sathe, J. Kao, and M. Papaefthymiou, "A 0.8-1.2GHz single-phase resonant-clocked FIR filter with level-sensitive latches," in *IEEE Custom Integrated Circuits Conference*, pp. 583–586, September 2007.
- [61] S. Chan, P. Restle, K. Shepard, N. James, and R. Franch, "A 4.6GHz resonant global clock distribution network," in *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 342–343, February 2004.
- [62] S. Chan, K. Shepard, and P. Restle, "Uniform-phase uniform-amplitude resonant-load global clock distributions," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 102–109, January 2005.

- [63] S. Chan, K. Shepard, and P. Restle, "1.1 to 1.6GHz distributed differential oscillator global clock network," in *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 518–519, February 2005.
- [64] S. Chan, K. Shepard, and P. Restle, "Distributed differential oscillators for global clock networks," *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 2083–2094, September 2006.
- [65] J.-Y. Chueh, C. Ziesler, and M. Papaefthymiou, "Experimental evaluation of resonant clock distribution," in *Proceedings of the 2004 IEEE Computer society Annual Symposium on VLSI*, pp. 135–140, February 2004.
- [66] J.-Y. Chueh, C. Ziesler, and M. Papaefthymiou, "Empirical evaluation of timing and power in resonant clock distribution," in *Proceedings of the 2004 International Symposium on Circuits and Systems*, vol. 2, pp. 249–52, May 2004.
- [67] J.-Y. Chueh, M. Papaefthymiou, and C. Ziesler, "Two-phase resonant clock distribution," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pp. 65–70, May 2005.
- [68] 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*, pp. 777–780, September 2006.
- [69] Z. Xu and K. Shepard, "Low-jitter active deskewing through injection-locked resonant clocking," in *IEEE Custom Integrated Circuits Conference*, pp. 9–12, September 2007.
- [70] Z. Xu and K. Shepard, "Design and analysis of actively-deskewed resonant clock networks," *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 558–568, February 2009.
- [71] S. Chan, P. Restle, T. Bucelot, S. Weitzel, J. Keaty, J. Liberty, B. Flachs, R. Volant, P. Kapusta, and J. Zimmerman, "A resonant global clock distribution for the CELL Broadband-Engine processor," in *IEEE International Solid-State Circuits Conference*, pp. 512–632, February 2008.
- [72] S. Chan, P. Restle, T. Bucelot, J. Liberty, S. Weitzel, J. Keaty, B. Flachs, R. Volant, P. Kapusta, and J. Zimmerman, "A resonant global clock distribution for the CELL Broadband Engine processor," *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 64–72, January 2009.
- [73] W. Athas, L. Svensson, J. Koller, N. Tzartzanis, and E. Ying-Chin Chou, "Low-power digital systems based on adiabatic-switching principles," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, pp. 398–407, December 1994.

- [74] S. G. Younis, T. F. Knight, and Jr., “Asymptotically zero energy split-level charge recovery logic,” in *International Workshop on Low Power Design*, pp. 177–182, April 1994.
- [75] D. Suvakovic and C. Salama, “Two phase non-overlapping clock adiabatic differential cascode voltage switch logic (ADCVSL),” in *IEEE International Solid-State Circuits Conference*, pp. 364–365, February 2000.
- [76] W.-H. Ma, J. C. Kao, V. S. Sathe, and M. Papaefthymiou, “A 187MHz subthreshold-supply robust FIR filter with charge-recovery logic,” in *IEEE Symposium on VLSI Circuits*, pp. 202–203, June 2009.
- [77] W.-H. Ma, J. Kao, V. Sathe, and M. Papaefthymiou, “187 MHz subthreshold-supply charge-recovery FIR,” *IEEE Journal of Solid-State Circuits*, vol. 45, pp. 793–803, April 2010.
- [78] R. Staszewski, K. Muhammad, and P. Balsara, “A 550-MSample/s 8-tap FIR digital filter for magnetic recording read channels,” *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1205–1210, August 2000.
- [79] K. Furuya and E. McCluskey, “Two-pattern test capabilities of autonomous TPG circuits,” in *Proceedings of International Test Conference*, pp. 704–711, October 1991.
- [80] J. Pille, C. Adams, T. Christensen, S. Cottier, S. Ehrenreich, T. Kono, D. Nelson, O. Takahashi, S. Tokito, O. Torreyter, O. Wagner, and D. Wendel, “Implementation of the CELL Broadband Engine in a 65nm SOI technology featuring dual-supply SRAM arrays supporting 6GHz at 1.3V,” in *IEEE International Solid-State Circuits Conference*, pp. 322–606, February 2007.
- [81] F. Klass, C. Amir, A. Das, K. Aingaran, C. Truong, R. Wang, A. Mehta, R. Heald, and G. Yee, “A new family of semidynamic and dynamic flip-flops with embedded logic for high-performance processors,” *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 712–716, May 1999.
- [82] H.-J. Oh, S. Mueller, C. Jacobi, K. Tran, S. Cottier, B. Michael, H. Nishikawa, Y. Totsuka, T. Namatame, N. Yano, T. Machida, and S. Dhong, “A fully pipelined single-precision floating-point unit in the synergistic processor element of a CELL processor,” *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 759–771, April 2006.
- [83] C. Webb, “IBM z10: The next-generation mainframe microprocessor,” *IEEE Micro*, vol. 28, pp. 19–29, March-April 2008.
- [84] S. Vangal, Y. Hoskote, N. Borkar, and A. Alvandpour, “A 6.2-GFLOPS floating-point multiply-accumulator with conditional normalization,” *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 2314–2323, October 2006.

- [85] N. Rohrer, M. Canada, E. Cohen, M. Ringler, M. Mayfield, P. Sandon, P. Kartschoke, J. Heaslip, J. Allen, P. McCormick, T. Pfluger, J. Zimmerman, C. Lichtenau, T. Werner, G. Salem, M. Ross, D. Appenzeller, and D. Thygesen, "PowerPC 970 in 130nm and 90nm technologies," in *IEEE International Solid-State Circuits Conference*, vol. 1, pp. 68–69, February 2004.
- [86] M. Schmookler and K. Nowka, "Leading zero anticipation and detection—a comparison of methods," in *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, pp. 7–12, June 2001.
- [87] E. M. Schwarz, "Binary Floating-Point Unit Design: The Fused Multiply-Add Dataflow," in *High-Performance Energy-Efficient Microprocessor Design*, ch. 8, pp. 189–208. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [88] A. Blotti and R. Saletti, "Ultralow-power adiabatic circuit semi-custom design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, pp. 1248–1253, November 2004.
- [89] G. Schrom, P. Hazucha, J.-H. Hahn, V. Kursun, D. Gardner, S. Narendra, T. Karnik, and V. De, "Feasibility of monolithic and 3D-stacked DC-DC converters for microprocessors in 90nm technology generation," in *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, pp. 263–268, 2004.
- [90] J.-Y. Chueh, *Resonant Clock Generation and Distribution*. PhD thesis, The University of Michigan, Ann Arbor, MI, USA, 2006.