

# Scalable Energy-Recovery Architectures

by

Tai-Chuan Ou

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Electrical Engineering)  
in The University of Michigan  
2015

Doctoral Committee:

Professor Marios C. Papaefthymiou, Chair  
Professor David Blaauw  
Associate Professor Kevin P. Pipe  
Associate Professor Zhengya Zhang

© Tai-Chuan Ou 2015  

---

All Rights Reserved

To my family and friends for their love and support

## ACKNOWLEDGMENTS

First and foremost I am grateful to my advisor Marios for guiding me throughout my career as a graduate student. Thank you for your support and patience. Even though it took me more than three years to publish our first paper when I first started, you still always seem to be so supportive and confident in me. I would like to thank my committee members: Zhengya, Prof. Blaauw, and Prof. Pipe, providing valuable feedback and comments.

Thanks to the staff of the EECS department, including Denise, Lauri, Lisa, Joel, Beth, Karen, and Steven for their immense help.

I am also grateful to folks from Marios research group including Suhwan, Joohee, Visvesh, Jerry, Wei-Hsiang, Shengshuo, and Hsi-Shou. I appreciate the opportunities working with you guys.

Thanks to my fellows in EECS department for sharing their expertise to me.

Finally and most importantly, I want to thank my family in Taiwan, and my friends both in Ann Arbor and in Taiwan. I could not have done this without all of you.

# TABLE OF CONTENTS

<b>DEDICATION</b> . . . . .	ii
<b>ACKNOWLEDGMENTS</b> . . . . .	iii
<b>LIST OF FIGURES</b> . . . . .	vii
<b>LIST OF TABLES</b> . . . . .	x
<b>ABSTRACT</b> . . . . .	xi
<b>CHAPTER</b>	
<b>I. Introduction</b> . . . . .	1
1.1 Principles of Energy Recovery . . . . .	4
1.2 LC Oscillation . . . . .	6
1.3 Charge-Recovery Logic . . . . .	7
1.4 Contributions . . . . .	12
1.4.1 Charge-Recovery LDPC Decoder with Semi-Automated Design Flow . . . . .	12
1.4.2 Charge-Recovery LDPC with In-Package Inductors .	13
1.5 Thesis Outline . . . . .	14
<b>II. Background</b> . . . . .	16
2.1 Reversible Logic . . . . .	16
2.2 Charge-Recovery Logic . . . . .	19
2.3 Summary . . . . .	25
<b>III. Design and Architecture of Charge-Recovery LDPC Decoder</b>	27
3.1 Introduction . . . . .	27
3.2 Boost Logic . . . . .	28
3.3 LDPC Decoder Architecture . . . . .	31
3.4 Power-Clock . . . . .	37
3.5 Summary . . . . .	42

<b>IV. Design Methodology</b>	44
4.1 Introduction	44
4.2 Standard-Cell Design Methodology	45
4.3 Semi-Automated Design Methodology for Charge-Recovery Logic	48
4.3.1 Front-End Design	48
4.3.2 Cell Library Design	49
4.3.3 Back-End Design	51
4.4 Comparison of the Two Design Methodologies	54
4.5 Summary	55
<b>V. Evaluation of 926MHz Charge-Recovery LDPC Decoder Test-Chip</b>	56
5.1 Introduction	56
5.2 Measurement Results	57
5.3 Summary	61
<b>VI. Design and Architecture of Charge-Recovery LDPC with In-Package Inductors</b>	63
6.1 Introduction	64
6.2 Power-Clock Generation and Distribution	66
6.3 Package Substrate Design	67
6.4 LDPC Decoder Design	71
6.5 Summary	71
<b>VII. Evaluation of 934MHz Charge-Recovery LDPC Test-Chip with In-Package Inductors</b>	73
7.1 Introduction	73
7.2 Measurement Results	74
7.3 Summary	78
<b>VIII. Conclusions and Future Research Directions</b>	80
8.1 Charge-Recovery LDPC Decoder with Semi-Automated Design Flow	80
8.2 Charge-Recovery LDPC with In-Package Inductors	81
8.3 Future Directions	82
8.3.1 Fully-Automatic Design Methodology	82
8.3.2 Efficient Charge-Recovery Designs for Wide Range of Operating Frequencies	83
8.3.3 AC-Powered Logic for IoT Devices	83

BIBLIOGRAPHY . . . . .	85
------------------------	----

# LIST OF FIGURES

## Figure

1.1	Equivalent models of (a) conventional static CMOS and (b) energy recovery. . . . .	5
1.2	Practical implementation using an inductor. . . . .	6
1.3	(a) Conventional switching schematic with voltage and current plots. (b) Charge-recovery switching schematic with voltage and current plots.	7
1.4	(a) An adiabatic dynamic logic CMOS inverter. (b) The four-phase clock waveform required for ADL gates. . . . .	9
1.5	A complex ADL gate implementing $\overline{ab} + \overline{c}$ . . . . .	9
1.6	Cascades of four ADL inverters, and their associated four-phase clocks.	10
1.7	Schematic of a resonant oscillator "clock supply" for an ADL system.	10
2.1	A (2,2) irreversible gate and its truth table. . . . .	17
2.2	The (3,3) Fredkin reversible gate, and its truth table. . . . .	18
2.3	Fredkin gate implementation of a single-bit full adder. . . . .	18
2.4	Schematic of a 2N-2P logic inverter, and the four-phase operating waveforms of the 2N-2P inverter. . . . .	20
2.5	Cascades of 4 2N-2P inverters, and their associated four-phase clocks.	21
2.6	Schematic of a PAL inverter. . . . .	21
2.7	Schematic of a CAL inverter. . . . .	22
2.8	Schematic of a PMOS SCAL inverter and an NMOS SCAL inverter.	23
2.9	Schematic of a PMOS SCAL-D inverter and a NMOS SCAL-D inverter.	23
2.10	Schematic of a boost logic inverter. . . . .	24
2.11	Schematic of a SBL inverter. . . . .	25
2.12	Schematic of an EBL inverter. . . . .	25
3.1	Schematic and operating waveforms of a boost logic gate. . . . .	28
3.2	Cascade of boost logic gates, and the corresponding output waveforms.	29
3.3	Schematic of a two-input 4-bit comparator boost logic gate. . . . .	30
3.4	Blip power-clock generator . . . . .	31
3.5	An example LDPC H-matrix and its bipartite graph representation (Tanner graph). . . . .	32
3.6	The 576-bit rate-5/6 LDPC matrix specified by IEEE 802.16e standard.	33
3.7	Bit error rate with different SNR of the implemented 576-bit rate-5/6 LDPC matrix specified by IEEE 802.16e standard. . . . .	34
3.8	LDPC matrix swapping and partitioning. . . . .	34
3.9	LDPC decoder architecture. . . . .	35



3.10	Gate-level block diagram of check node operation. . . . .	36
3.11	Power-clock generation and distribution network. . . . .	37
3.12	Simulated waveform of energy supplied to power-clock through one of the sixteen inductors. . . . .	39
3.13	Histogram of clock skew from full-block spice simulation. . . . .	40
3.14	PC skew distribution across Block 3 of LDPC die. . . . .	41
3.15	Voltage waveform of two-phase power-clock with unbalanced loading for the two phases. . . . .	41
3.16	Voltage waveform of two-phase power-clock with comparable loading for the two phases. . . . .	42
4.1	Standard-cell design methodology. . . . .	46
4.2	Inputs and outputs for synthesis. . . . .	47
4.3	Place-and-route flow. . . . .	48
4.4	Layout view of a two-input 4-bit comparator boost logic gate with weaker drive strength. . . . .	50
4.5	Layout view of a two-input 4-bit comparator boost logic gate with stronger drive strength. . . . .	51
4.6	Place-and-route flow for charge-recovery logic. . . . .	52
4.7	Simplified automatic place-and-route flow. . . . .	53
4.8	An actual layout view of the decoder chip after APR. . . . .	53
5.1	Microphotograph of the charge-recovery LDPC test-chip in 65nm CMOS. . . . .	57
5.2	Measured energy per cycle versus operating frequency. . . . .	58
5.3	Measured resonant frequency distribution at $V_{DC}=0.64V$ and $V_{CC}=0.36V$ of 16 chips. . . . .	60
5.4	Measured energy per cycle versus input SNR. . . . .	61
6.1	Charge-recovery LDPC decoder chip in 65nm flip-chip technology and custom-designed FC-BGA package substrate with 16 in-package inductors. . . . .	63
6.2	Blip power-clock generator, formed by in-package inductors, cross- coupled internal drivers, and frequency tuning circuits with pulses generated by a ring oscillator and a pulse generator. . . . .	65
6.3	Power-clock generation and distribution: clock mesh network for $PC$ and $\overline{PC}$ , 100 flip-chip bumps, 16 frequency tuning circuits, and 144 negative transconductance devices. . . . .	67
6.4	Cross-section view of the package substrate. . . . .	68
6.5	Custom-designed 6-layer FC-BGA package substrate. . . . .	69
6.6	Geometry and inductor parameters from HFSS simulations for one of the 16 in-package inductors. . . . .	70
7.1	Measured energy per cycle versus operating frequency. . . . .	74
7.2	Microphotograph of the charge-recovery LDPC test chip in 65nm CMOS. . . . .	76
7.3	Microphotograph of the top side of the custom-designed 6-layer FC- BGA package substrate. . . . .	77

7.4	Microphotograph of the bottom side of the custom-designed 6-layer FC-BGA package substrate. . . . .	78
-----	---	----

## LIST OF TABLES

### Table

2.1	Overview of charge-recovery logic families covered in this chapter. .	26
4.1	Comparison with recent silicon-demonstrated charge-recovery test-chips. . . . .	45
5.1	Chip summary and comparison with state-of-the-art designs. . . . .	62
7.1	Chip summary and comparison with state-of-the-art designs. . . . .	75

# ABSTRACT

Scalable Energy-Recovery Architectures

by

Tai-Chuan Ou

Chair: Marios C. Papaefthymiou

Energy efficiency is a critical challenge for today's integrated circuits, especially for high-end digital signal processing and communications that require both high throughput and low energy dissipation for extended battery life. Charge-recovery logic recovers and reuses charge using inductive elements and has the potential to achieve order-of-magnitude improvement in energy efficiency while maintaining high performance. However, the lack of large-scale high-speed silicon demonstrations and inductor area overheads are two major concerns.

This dissertation focuses on scalable charge-recovery designs. We present a semi-automated design flow to enable the design of large-scale charge-recovery chips. We also present a new architecture that uses in-package inductors, eliminating the area overheads caused by the use of integrated inductors in high-performance charge-recovery chips.

To demonstrate our semi-automated flow, which uses custom-designed standard-cell-like dynamic cells, we have designed a 576-bit charge-recovery low-density parity-check (LDPC) decoder chip. Functioning correctly at clock speeds above 1 GHz,

this prototype is the first-ever demonstration of a GHz-speed charge-recovery chip of significant complexity. In terms of energy consumption, this chip improves over recent state-of-the-art LDPCs by at least 1.3 times with comparable or better area efficiency.

To demonstrate our architecture for eliminating inductor overheads, we have designed a charge-recovery LDPC decoder chip with in-package inductors. This test-chip has been fabricated in a 65nm CMOS flip-chip process. A custom 6-layer FC-BGA package substrate has been designed with 16 inductors embedded in the fifth layer of the package substrate, yielding higher Q and significantly improving area efficiency and energy efficiency compared to their on-chip counterparts. From measurements, this chip achieves at least 2.3 times lower energy consumption with better area efficiency over state-of-the-art published designs.

# CHAPTER I

## Introduction

Energy efficiency has become a major design challenge not only for ultra-low power designs but also for high performance ones. Moore's Law [1] has been driving very-large-scale integration (VLSI) technology revolution for more than half a century, doubling the number of available transistors on integrated circuits approximately every two years. In 1974, Dennard's scaling [2] stated that as transistors scale, supply voltage and current scale by the same factor as critical transistor dimensions. However, this constant power density scaling trend is no longer valid for recent technology generations while Moore's Law still continues.

The breakdown of Dennard's scaling is mainly caused by the following reasons. For recent technology nodes, the scaling of the supply voltage becomes slower, or even stops to scale compared to the scaling of device size. Furthermore, as device sizes shrink, static power increases due to thinner gate oxide and shorter channel length. As a result, the power density of recent technology nodes no longer follows Dennard's scaling to remain unchanged, but increases exponentially, exacerbating heat removal issues in chip packaging. Dark silicon [3], the portion of a chip that cannot be powered on due to the packaging thermal limit, exemplifies this problem. Coupled with the incremental advances in battery technology, this exponential growth of power density has become the main challenge of today's VLSI design. For the success of next-

generation very-large-scale integration (VLSI) applications, it is therefore imperative to achieve high energy efficiency while maintaining high performance.

Many innovations have been proposed aimed at solving this critical challenge. At the circuit level, voltage scaling is one of the most effective methods for reducing energy consumption [4]. However, energy savings through voltage scaling come at the expense of performance and reliability degradation. Clock gating is another way of reducing dynamic power, but savings are limited, since clock power only accounts for part of total power consumption. Power gating provides an effective way to reduce static power dissipation, but it does not address dynamic power. At the architectural level, parallel architectures and multi-core designs have been proposed to mitigate the problem of dynamic power. For each of these innovations, improvements are limited, and fundamental innovations are required to follow Dennard's scaling trend.

Charge recovery is an alternative design approach that has the potential to achieve order-of-magnitude improvement in energy efficiency while maintaining high performance by gradually charging and discharging parasitic capacitance and recycling the charge at the end of each cycle [5–10]. Due to this fundamental difference between conventional CMOS designs and charge-recovery designs, the energy consumption characteristics of these two systems are quite different. The energy dissipation of conventional CMOS designs is governed by the equation  $E_{conv} = (1/2)CV^2$  for a complete charging or discharging cycle. For charge-recovery designs, however, energy dissipation is governed by the equation  $E_{er} = (k/T)CV^2$ , where  $k$  is a constant proportional to the  $RC$  constant of the system, and  $T$  is the duration of charging or discharging. By exploiting the energy-latency tradeoff indicated by its dissipation equation, charge-recovery logic has the potential to significantly improve energy efficiency.

Despite its promising potential, charge-recovery design has yet to be demonstrated in silicon at a large scale. Previous charge-recovery chips [11–14] are limited to small

and relatively simple designs operating at low frequencies (hundreds of MHz or lower), not fully exploring the potential of charge-recovery logic. This limitation mainly comes from the fact that charge-recovery chips are full-custom designs. Moreover, the overheads associated with the use of inductive elements further limits the scalability of charge-recovery designs. Specifically for high performance charge-recovery designs, on-chip inductors are used, resulting in silicon area overheads. For charge-recovery chips operating at lower frequencies, discrete off-chip inductors are used, resulting in area overheads on printed circuit boards and extra costs.

This dissertation focuses on scalable charge-recovery designs. We explore semi-automated design flows to enable the design of large-scale charge-recovery chips. We also explore approaches for eliminating area overheads caused by the use of integrated inductors for GHz-speed charge-recovery chips.

To enable large-scale charge-recovery designs, we present a semi-automated standard-cell-like design flow that incorporates custom-designed dynamic cells. To demonstrate the effectiveness of this design flow, we have designed and evaluated a 576-bit charge-recovery low-density parity-check (LDPC) decoder as an example prototype. LDPC is a very popular error correcting code in modern communication standards, requiring complex and power-intensive computations. 16 on-chip inductors are used to resonate the design and recover charge from gate fanouts. From device-level simulations, when self-oscillating at 866 MHz, the chip recovers 51.4% of the energy supplied to it. Clock meshes are used to distribute the two-phase power-clock, yielding a worst-case skew of 11.3 ps. The test-chip has been fabricated in a 65nm CMOS process. Functioning correctly at clock speeds ranging from 408 MHz to 1.05 GHz, this chip is the first ever silicon demonstration of a charge-recovery design of significant complexity. With over 57,000 gates, it has 32 times more devices than the largest previously-reported charge-recovery test-chips. When operating at 926 MHz, our test-chip consumes 6.4 pJ/bit/iteration with a 8.9 Gbps throughput, achieving at least 1.3 times improve-



ment in energy consumption with comparable or better area efficiency over previous state-of-the-art commercial-strength LDPCs.

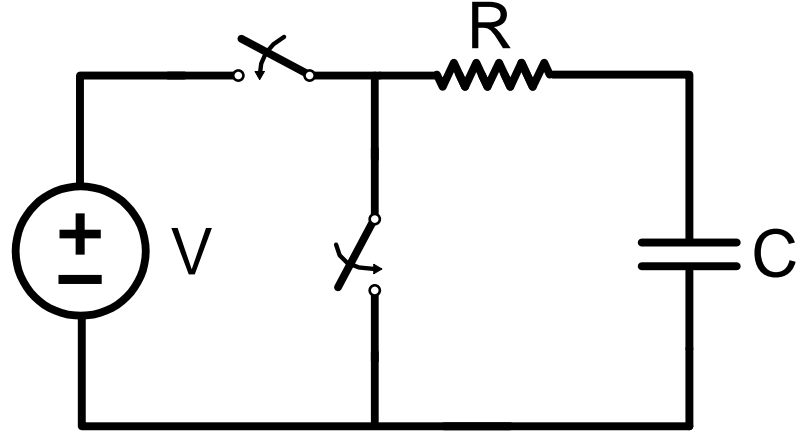
To enable area-efficient GHz-speed charge-recovery designs, we present a new architecture that uses in-package inductors, eliminating the area overheads caused by on-chip inductors. As a proof-of-concept, we have designed and evaluated a LDPC decoder including a custom designed 6-layer flip-chip–ball-grid-array (FC-BGA) package substrate with 16 in-package inductors and a test-chip fabricated in a 65nm CMOS flip-chip process. When operating at 934MHz, the decoder reaches a 9Gb/s throughput, consuming 286mW, or 3.2pJ/b/iteration, achieving at least 2.3 times lower energy consumption with better area efficiency over state-of-the-art published designs of comparable code length, complexity, and throughput.

The remainder of this chapter is organized as follows: Section 1.1 discusses the basic principles of energy-recovery systems. Section 1.2 shows how LC oscillators can be designed and used to recover charge. Section 1.3 presents the main ideas of charge-recovery logic, and provides a practical implementation as an example. The contributions in this work are summarized in Section 1.4, and the outline of this dissertation is covered in Section 1.5.

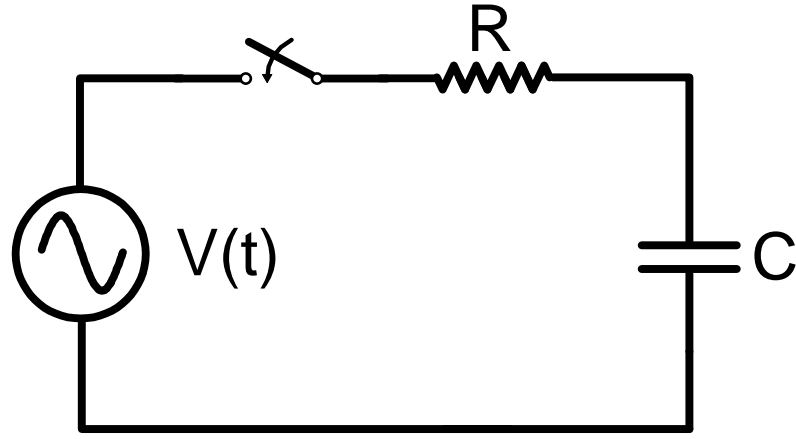
## 1.1 Principles of Energy Recovery

Energy recovery is an approach to the design of VLSI systems that achieve increased energy efficiency by exchanging energy between different subsystems. For example, just like a hybrid car can convert its kinetic energy into electric energy when braking for later use, instead of dissipating it as heat, an energy-recovery VLSI converts electric energy into magnetic energy and then back to electric energy for re-use in subsequent cycles.

In CMOS circuits, the state of a logic gate is determined by the voltage of the capacitances associated with the output nodes of that gate. If a node with capacitance



(a)



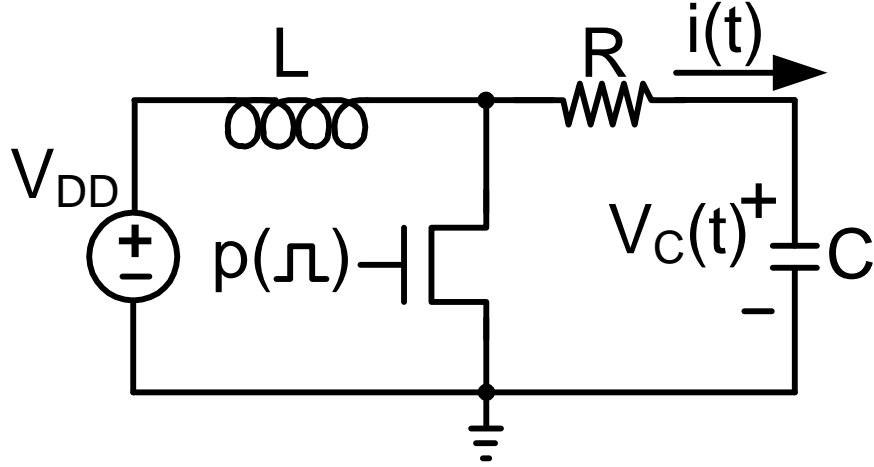
(b)

Figure 1.1: Equivalent models of (a) conventional static CMOS and (b) energy recovery.

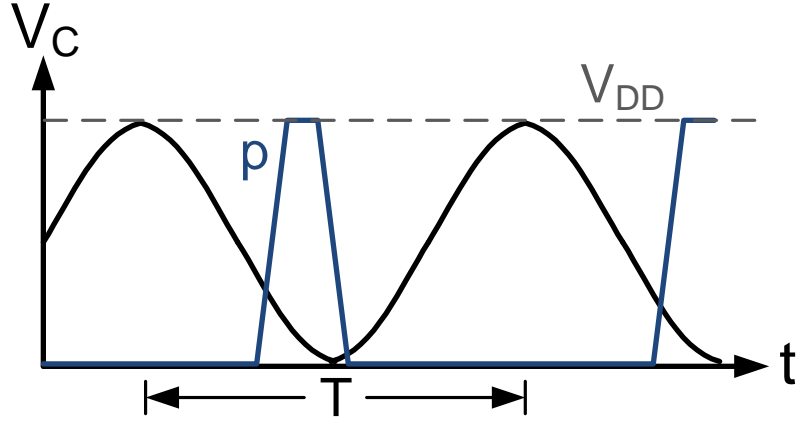
$C_L$  is charged to a required voltage  $V_{DD}$ , the energy stored in  $C_L$  is

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

When removing the stored charge to switch to another state, conventional static CMOS dissipates this stored energy by sending all charge to ground, as shown in Figure 1.1(a). Figure 1.1(b) shows that if the stored charge is instead transferred to another place (a time-varying supply in this case) for further use, the energy could be "recovered" instead of dissipated.



(a) Schematic



(b) Waveform

Figure 1.2: Practical implementation using an inductor.

## 1.2 LC Oscillation

A common approach to implementing the energy recovery principle in CMOS circuits is to employ an inductor to store the electric energy returned from the circuit into magnetic energy. Figure 1.2(a) shows a simplified model of one possible practical implementation. By adopting an inductor  $L$ ,  $C$  is periodically charged and discharged through  $LC$  resonance, and the resulting voltage waveform of  $C$  is shown in Figure 1.2(b). The only losses in the system are the losses in the parasitic resistance  $R$  of the circuit, and can be replenished and compensated by a shunt switch

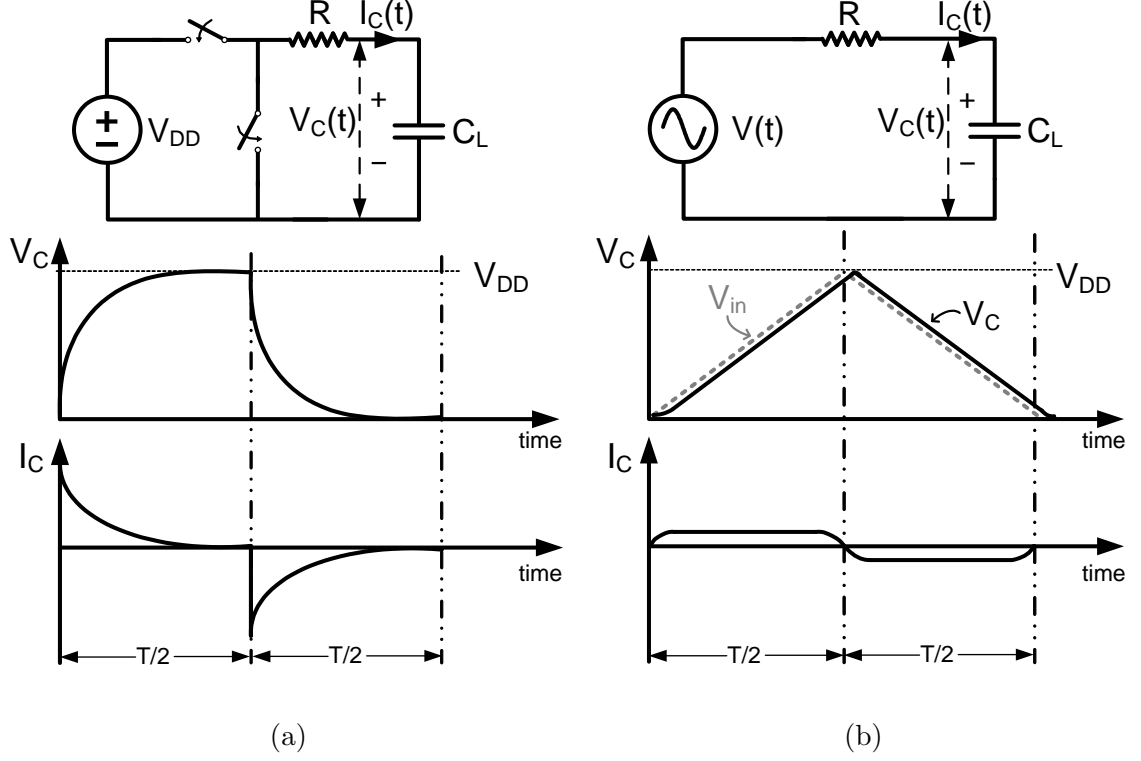


Figure 1.3: (a) Conventional switching schematic with voltage and current plots. (b) Charge-recovery switching schematic with voltage and current plots.

driven by pulse  $p$  to maintain the oscillation. The self-resonance frequency  $Fr$  of an ideal  $LC$  system is

$$Fr = \frac{1}{2\pi} \sqrt{\frac{1}{LC}} , \quad (1.2)$$

where  $C$  is the capacitance of the system, and  $L$  is the inductance. This implementation can also be forced to run off-resonance by controlling the frequency of pulse  $p$  to operate at a desired frequency. However, a distortion will be seen at the output waveform if this frequency is too far away from  $Fr$ .

### 1.3 Charge-Recovery Logic

Charge-recovery logic is a circuit family whose operation relies on energy recovery principles [5, 15]. Two systems are shown in Figure 1.3, a conventional switching system in Figure 1.3(a), and charge-recovery system in Figure 1.3(b). For conventional

switching, the output load is charged and discharged by constant supplies  $V_{DD}$  or ground. The transient voltage across the output load and the transient current toward the load are also shown during charging and discharging cycle. In the beginning of the charging or discharging cycle, the voltage drop across the resistive element is at its highest level, resulting in a spike in current profile. The energy consumption of this conventional switching circuit during each charging or discharging cycle is

$$E_{conv} = (1/2)C_L V_{DD}^2 . \quad (1.3)$$

The charge-recovery system, however, is charged and discharged by a gradually changing supply. Due to the gradual transition, the voltage across the load follows the supply closely, and the resulting current flowing toward the output load is smaller. If the supply is a resonant source, when discharging the load, the supply will be able to recover and reuse the energy.

The current flowing toward the load during the first half of a cycle is

$$I_{CR} = 2 \frac{C_L V_{DD}}{T} , \quad (1.4)$$

based on first order analysis. Therefore, the energy consumption is

$$\begin{aligned} E_{CR} &= \int_0^{\frac{T}{2}} I_{CR}^2 R \, dt \\ &= \int_0^{\frac{T}{2}} \left( 2 \frac{C_L V_{DD}}{T} \right)^2 R \, dt \\ &= \frac{R C_L}{T} C_L V_{DD}^2 . \end{aligned} \quad (1.5)$$

As can be seen from Equation (1.5), the energy consumption of the charge-recovery system has a  $T$  term in its denominator. This trade-off between energy and latency

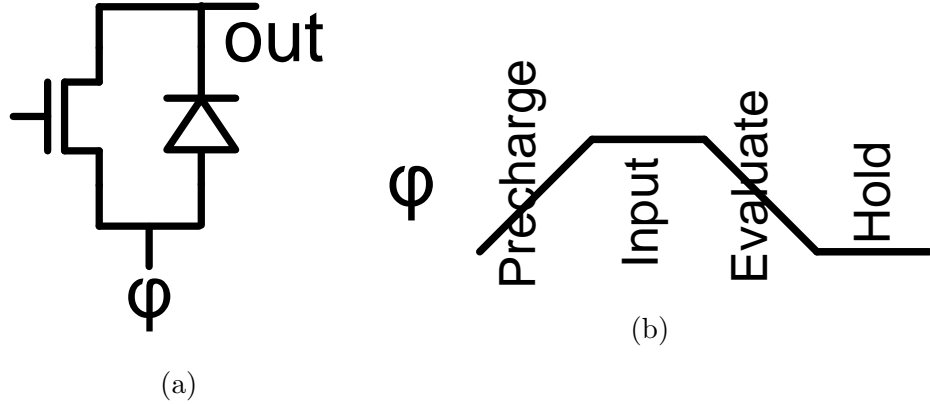


Figure 1.4: (a) An adiabatic dynamic logic CMOS inverter. (b) The four-phase clock waveform required for ADL gates.

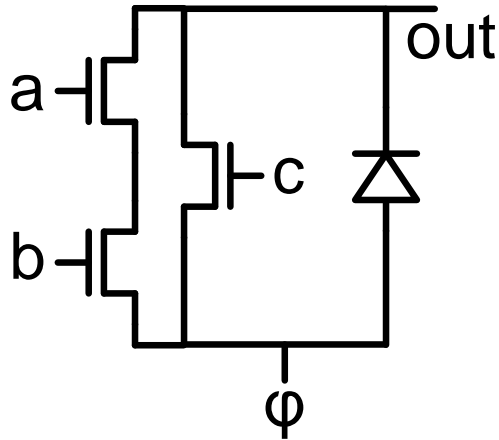


Figure 1.5: A complex ADL gate implementing  $\overline{ab + c}$ .

indicates that the more slowly the load is charged, the less energy will be consumed.

To illustrate how a practical implementation of charge-recovery logic works, adiabatic dynamic logic (ADL) [16], an early implementation of charge-recovery logic proposed by Dickinson and Denker in 1994, is briefly shown here as an example.

Figure 1.4 shows the schematic of an ADL inverter and the four-phase clock waveform  $\phi$  required by ADL gates. The operation of ADL gates can be divided into two phases: precharge and evaluate. In the precharge phase,  $\phi$  gradually transitions from 0 to  $V_{DD}$  precharging the output node through the diode. In the evaluation phase,  $\phi$  gradually transitions from  $V_{DD}$  to ground, and the NMOS conditionally discharges the

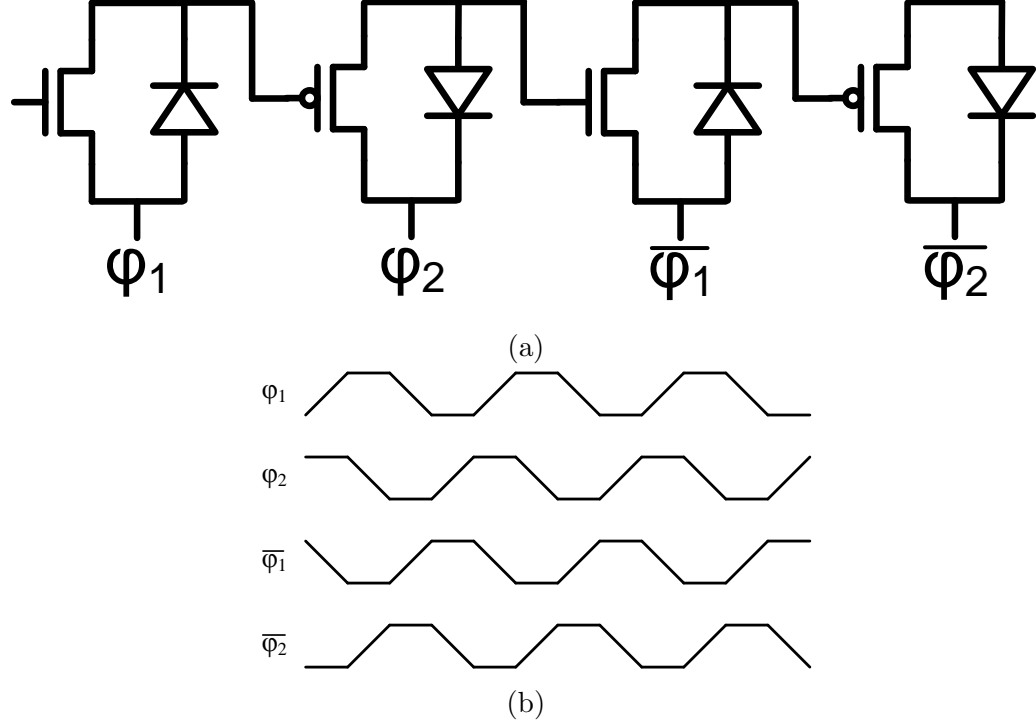


Figure 1.6: Cascades of four ADL inverters, and their associated four-phase clocks.

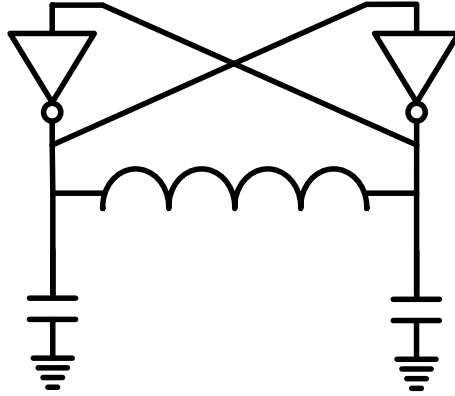


Figure 1.7: Schematic of a resonant oscillator "clock supply" for an ADL system.

output as  $\phi$  falls. ADL can also be used to implement more complex logic functions. Figure 1.5 shows the schematic of an ADL NAND/NOR gate as an example.

Cascades of ADL gates are formed by alternating NMOS ADL gates and PMOS ADL gates, clocking the gates with a four-phase clock. Four cascaded ADL gates are shown in Figure 1.6, along with the four-phase clock required by ADL gates. To generate a two-phase "clock supply" for driving two complementary clock-supply

nodes of an ADL system, resonant oscillators are used as shown in Figure 1.7. Two inverters are used to maintain the amplitude of the supply waveform. Two sets of these oscillators are used to generate the four-phase clock required for an ADL system.

An ADL test-chip with a chain of 64 ADL inverters was fabricated in  $0.9\mu\text{m}$  technology. The test-chip was tested successfully at frequencies up to 250 MHz. A factor of 15 reduction in dissipation was reported with ADL circuits over conventional CMOS. ADL gates are simple, but their single rail structure makes the clock loading data-dependent, yielding high clock jitter. Using diodes to precharge output nodes also results in a significant voltage difference across the diode, generating large current flow and degrading energy efficiency.

ADL is an early implementation of charge-recovery logic. Various different implementations of charge-recovery logic have been introduced after ADL. However, the underlying operating principles, objectives, and trade-offs are basically quite similar. A summary of previous charge-recovery designs will be discussed in Chapter II in more detail.

Like the ADL test-chip, one common characteristic of previous (including recent) charge-recovery test-chips is that these designs are all limited to small and relatively simple designs, implementing datapaths such as chains of test gates [11, 16], simple DSP processing nodes [14, 17], and FIR filters [12, 13]. Another limitation of previous charge-recovery logic comes from area overheads caused by the use of inductive elements. High performance charge-recovery designs use on-chip inductors to recover charge from gate fanout to achieve energy-efficient operations, costing significant silicon area. For designs operating at lower frequencies, discrete inductors are used, resulting in extra component cost and printed circuit board area.



## 1.4 Contributions

Charge-recovery logic has the potential to achieve order-of-magnitude improvement in energy efficiency while maintaining high performance. However, the lack of large-scale high-speed demonstration in silicon, and the area overhead caused by the need of inductors for charge-recovery logic are two major concerns. This dissertation focuses on scalable charge-recovery designs. We explore semi-automated design flows to enable the design of large-scale charge-recovery chips that operate at GHz clock rates. We also explore the elimination of the area overhead caused by the use of integrated inductors in GHz-speed high-performance charge-recovery chips through the embedding of inductors in the chip package.

### 1.4.1 Charge-Recovery LDPC Decoder with Semi-Automated Design Flow

To explore scalable charge-recovery designs, we present a semi-automated standard-cell-like design flow to enable the design of large-scale charge-recovery chips. To demonstrate this semi-automated design flow, we have designed and evaluated a 576-bit charge-recovery low-density parity-check (LDPC) decoder as an example prototype. LDPC is a very popular error correcting code in modern communication standards, requiring complex and power-intensive computations.

The chip has been fabricated in a 65nm CMOS process and relies on 16 integrated inductors to achieve energy-efficient operation by recovering charge from gate fanouts. In simulations, when self-oscillating at 866 MHz, the chip recovers 51.4% of the energy supplied to it. Clock meshes are used to distribute the two-phase power-clock with a worst-case clock skew of 11.3 ps. In terms of device count, this chip is more than an order of magnitude larger than the largest previously-reported chips with charge-recovery logic [11–14], enabled by the semi-automated design methodology we have developed. Correct functionality has been validated for clock frequencies ranging from 408 MHz to 1.05 GHz. When operating at 926 MHz, the chip achieves a throughput

of 8.9 Gbps at 6.4 pJ/bit/iteration, improving on results in previous state-of-the-art commercial-strength LDPCs [18–21] by at least 1.3 times in energy consumption with comparable or better area efficiency, even without technology scaling.

Part of this work was published in ISSCC 2014 [22].

#### 1.4.2 Charge-Recovery LDPC with In-Package Inductors

Charge-recovery circuits rely on inductive elements to recover charge from gate fanouts and achieve energy-efficient operation. On-chip inductors are used to achieve GHz-speed high-performance charge-recovery chips, at the cost of silicon area overhead and inductors with low quality factor  $Q$  due to resistive on-chip metals. For charge-recovery designs operating at lower frequencies (hundreds of MHz or lower), high  $Q$  discrete inductors are used, resulting in extra board area and extra costs. This work is the first one to explore advantages from both the high performance and the discrete ends.

To enable area-efficient high-speed charge-recovery designs and eliminate the area overhead caused by on-chip inductors, we present and evaluate a new architecture that uses in-package inductors. As a proof-of-concept, we have designed and evaluated a 576-bit charge-recovery LDPC decoder with in-package inductors. The decoder includes a custom designed 6-layer FC-BGA package substrate and a test chip fabricated in a 65nm CMOS flip-chip process. 16 inductors are designed on the fifth layer of the 6-layer FC-BGA package substrate so that the need of on-chip inductors is removed, and inductors with better  $Q$  are available, significantly improving area efficiency and energy efficiency compared to alternative implementations with on-chip inductors.

When operating at 934MHz, the decoder reaches a 9Gb/s throughput, consuming 286mW, or 3.2pJ/b/iteration, achieving at least 2.3 times lower energy consumption with similar or even better area efficiency over state-of-the-art published designs of

comparable code length, complexity, and throughput [19–22].

Part of this work will be appearing in ASSCC 2015 [23].

## 1.5 Thesis Outline

The remainder of this dissertation is organized as follows: In Chapter II, we survey and give a summary of previous work in the area of charge-recovery logic. First we discuss the origin of charge-recovery logic, which is reversible logic. Then we show the history and evolution of charge-recovery logic structures, including the boost logic family used in our silicon prototypes.

In Chapter III, we present the design and architecture of our charge-recovery LDPC decoder. Power-clock generation and distribution are evaluated through simulations.

Chapter IV explains our standard-cell-like semi-automated design flow, and compares it with standard-cell design methodology. This chapter shows the layout of one of the gates in the library of boost logic gates that we used in our prototyping efforts. It also shows layout from the decoder after place-and-route is performed.

Chapter V shows the evaluation of our charge-recovery LDPC decoder test-chip fabricated in a 65nm CMOS process. Measurement results are reported, discussed, and compared with recent state-of-the-art LDPC designs. This work was published in [22].

In Chapter VI, we present the design of a charge-recovery LDPC decoder with in-package inductors, including a test-chip fabricated in a 65nm CMOS flip-chip process and a custom-designed 6-layer flip-chip–ball-grid-array package substrate.

Chapter VII gives the evaluation results of our charge-recovery LDPC decoder with in-package inductors. The characteristics of our LDPC decoder and comparisons with recent state-of-the-art LDPC decoder are also presented here. This work will be appearing in ASSCC 2015 [23].

Chapter VIII summarizes our contributions in this dissertation, and presents directions for future research in this area.

## CHAPTER II

# Background

In this chapter we survey previous work in the area of energy recovery design. Section 2.1 describes reversible logic, which inspired charge-recovery techniques. In Section 2.2, we follow the evolution of charge-recovery logic topologies, exploring different charge-recovery techniques and challenges of early work which lead to this dissertation research.

### 2.1 Reversible Logic

Early charge-recovery logic techniques can be traced back to reversible logic. Inspired by research analyzing the theoretically fundamental limits in different fields, such as the limit presented by Shannon in 1948 on the amount of information bits that can be transmitted error-free through a noisy channel (known as Shannon limit) [24], in the 1960s Landauer tried to answer this fundamental question: what are the physical limits of the process of computation [25]? Landauer concluded that the minimum energy required to change one bit of information is  $kT\ln 2$ , where  $k$  is Boltzmann's constant and  $T$  is the absolute temperature of the system [26]. To go beyond this  $kT\ln 2$  limit, the computation will have to be reversible, because the minimum energy required is proportional to the number of bits destroyed during the computation [27]. No information is destroyed by performing the computation in a reversible manner,

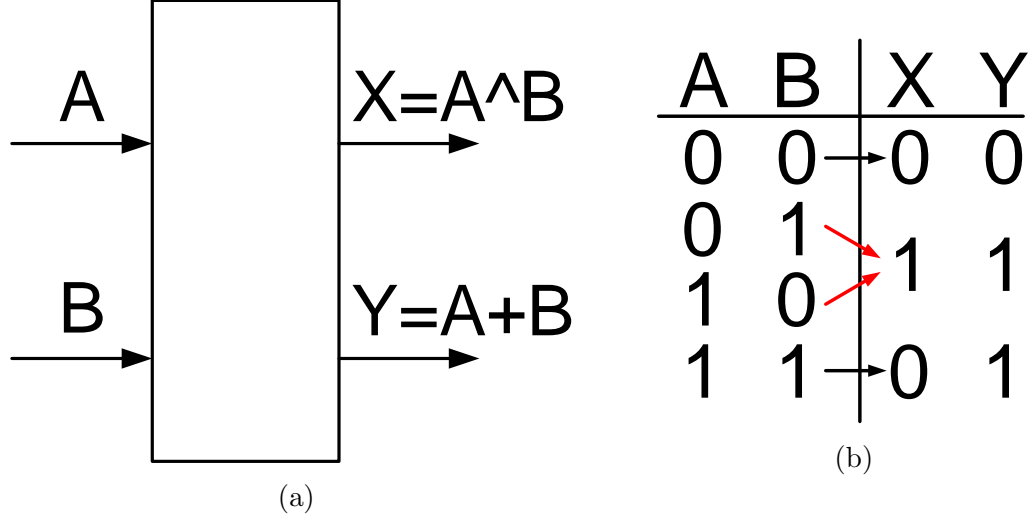


Figure 2.1: A (2,2) irreversible gate and its truth table.

and therefore theoretically zero energy would be dissipated.

For any system to be reversible, it has to be capable of operating in a backward direction, so that the inputs are reproducible from the outputs, and no information is destroyed. The logic supporting such backward operation of a system is called reversible logic [28].  $A(m, n)$  denotes a gate with  $m$  inputs and  $n$  outputs. For example, a (1,1) inverter is a reversible gate in which the output is the inverse of the input. A logic gates is reversible if the function it implements is bijective, or there is a one-to-one mapping between each input and each output. Figure 2.1 shows an irreversible (2,2) gate, since there is a 2-to-1 mapping from the input to the output.

In 1982 Fredkin and Toffoli introduced a (3,3) reversible logic gate, the Fredkin gate [29]. The gate implements  $X = A$ ,  $Y = \bar{A}B + AC$ ,  $Z = AB + \bar{A}C$ , where  $(A, B, C)$  is the input vector, and  $(X, Y, Z)$  is the output vector. Figure 2.2 shows the function of the gate and its truth table. The Fredkin gate is not only bijective but also universal, meaning that any logical operation can be constructed entirely of Fredkin gates. As an example, Figure 2.3 shows a full adder which is built with Fredkin gates.

Using reversible logic has many challenges and issues. From the above example

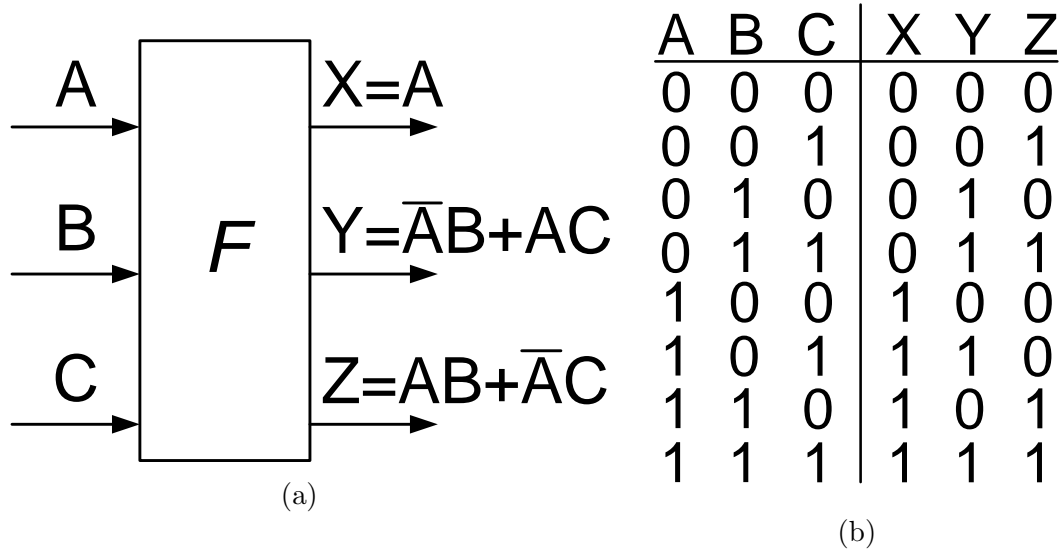


Figure 2.2: The (3,3) Fredkin reversible gate, and its truth table.

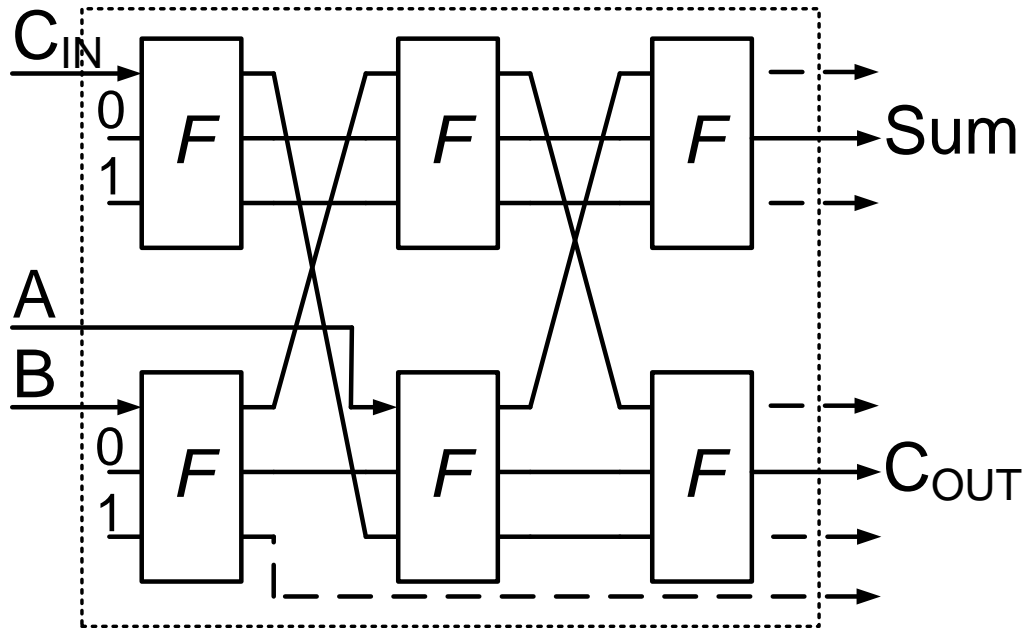


Figure 2.3: Fredkin gate implementation of a single-bit full adder.

we can find that realizing a system using reversible logic is quite different from using conventional irreversible logic. Using reversible logic creates redundant outputs that are useless in terms of implementing the desired functions. Moreover, reversible computation requires that all logical computations be carried out twice: once in the forward and once in the backward direction, resulting in additional latency and circuit

overhead [30]. Furthermore, it requires significant amount of temporary storage to keep the intermediate results until the backward computation is ready, yielding significant overhead in energy and circuit area. Consequently, the overhead needed for realizing a system using reversible logic significantly limits its use in CMOS circuits.

## 2.2 Charge-Recovery Logic

Charge-recovery logic is a circuit family that adopts energy recovery principles by gradually charging and discharging the capacitance and recycling charge at the end of each cycle.

Early work in charge-recovery logic, such as split-level charge recovery logic (SCRL) [31] and reversible energy recovery logic (RERL) [32], involved the design of reversible logic gates. This work implemented an additional inverse function in the backward direction so that after the original functional computation which moves charge towards the output, charge is returned to the beginning. However, using reversible logic requires a large amount of temporary storage to maintain intermediate results, yielding energy and circuit overheads [30].

Later work in charge-recovery logic keeps the key idea of energy recovery principles, but deviates from reversible logic. The adiabatic dynamic logic (ADL) we have shown in Chapter I is one of the first irreversible charge-recovery logic families. As mentioned in Section 1.3, the single-rail structure makes clock loading data-dependent, yielding high clock jitter. A test-chip was fabricated in 0.9  $\mu\text{m}$  technology, implementing a chain of 64 ADL inverters. Correct function was verified at 250 MHz using external power-clock source.

Kramer et al. proposed the 2N-2P logic to address this data-dependent clock loading issue [33]. Figure 2.4 shows the schematic and timing waveforms of a 2N-2P buffer/inverter. The circuit uses differential logic, and it has a pair of cross-coupled PMOS devices on the top to gradually charge and discharge output loading with



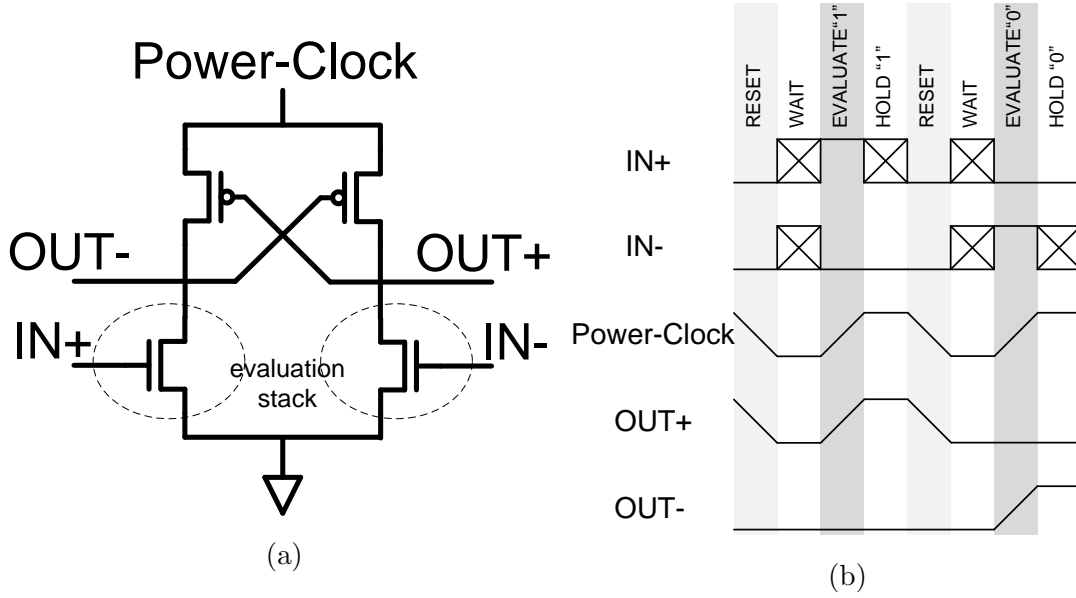


Figure 2.4: Schematic of a 2N-2P logic inverter, and the four-phase operating waveforms of the 2N-2P inverter.

power-clock, and a pair of complementary evaluation stacks to perform the logic operation. 2N-2P logic is also known as efficient charge recovery logic (ECRL) [34]. The operation of a 2N-2P gate can be divided into 4 phases: reset, wait, evaluate, and hold. In the reset phase, the power-clock ramps down, and the high output follows the power-clock so that both outputs reset to low. During the wait phase the power-clock stays low, maintaining the outputs low, and the inputs are evaluated. In the evaluate phase, the power-clock goes up, and the outputs evaluate to a complementary state so that one of the outputs stays low and the other follows the power-clock and ramps up. During the hold phase, the power-clock stays high while the inputs reset to low.

Cascades of 2N-2P gates are formed by connecting them with a four-phase power-clock such that any two connected gates are supplied by power-clocks with 90-degree phase difference. Figure 2.5 shows 4 cascaded 2N-2P inverters, and the four-phase power-clock. A shift register containing 1,000 shift stages was designed and fabricated using 2N-2P logic in 0.8  $\mu\text{m}$  CMOS, successfully tested at frequencies up to 100 MHz.

Several variants of the 2N-2P logic have been introduced since 2N-2P was pro-

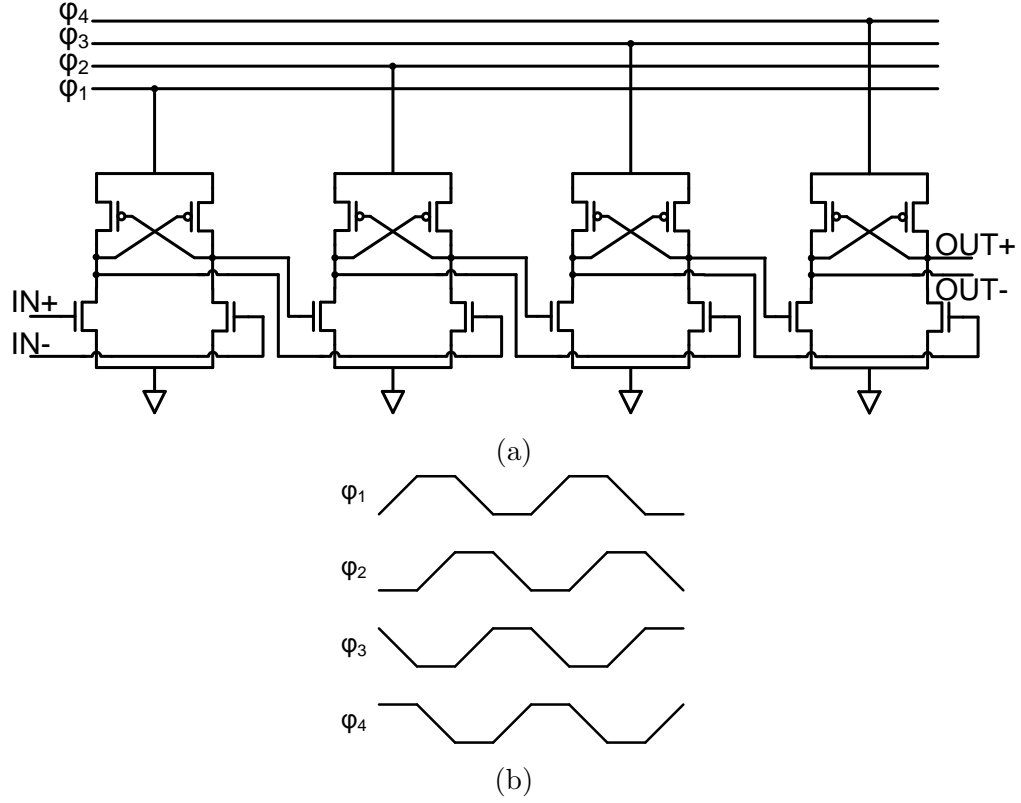


Figure 2.5: Cascades of 4 2N-2P inverters, and their associated four-phase clocks.

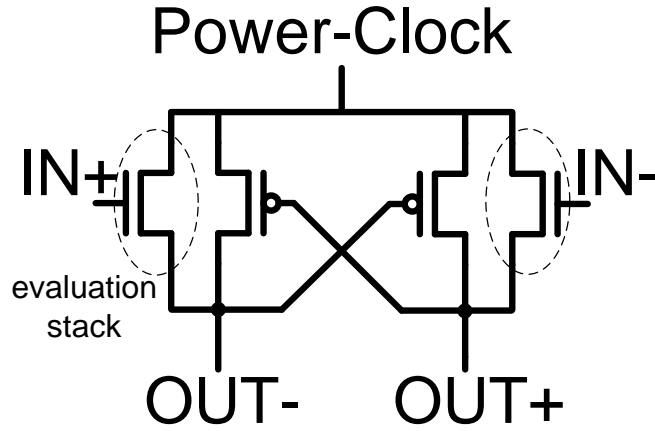


Figure 2.6: Schematic of a PAL inverter.

posed, such as pass transistor adiabatic logic (PAL) [35] and clocked CMOS adiabatic logic (CAL) [36]. Figure 2.6 shows the schematic of a PAL inverter. It retains the cross-coupled PMOS from 2N-2P logic, and moves evaluation stacks in parallel to

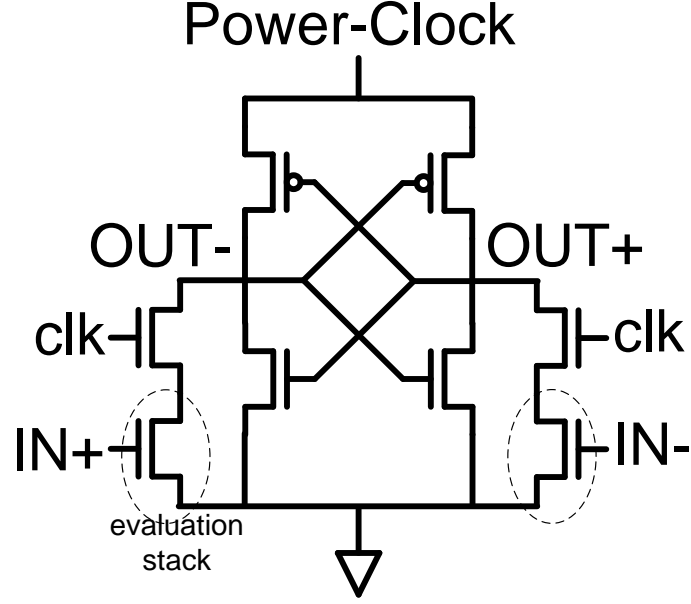


Figure 2.7: Schematic of a CAL inverter.

the PMOS devices. Unlike 2N-2P logic which requires a four-phase power-clock, PAL operates with a two-phase power-clock, and the gate complexity is also lower. A 1,600-stage PAL shift-register was fabricated in  $1.2\ \mu\text{m}$  technology and verified with 10 MHz operating frequency. Figure 2.7 shows the schematic of a clocked CMOS adiabatic logic inverter. CAL operates from a single-phase power-clock. The test chip, a chain of 736 CAL inverters, was fabricated in  $1.2\ \mu\text{m}$  technology and verified at 50 MHz.

Kim et al. proposed the source-coupled adiabatic logic (SCAL), which is a dual-rail logic using true single-phase clock [37]. Figure 2.8 shows the schematic of a SCAL inverter. The structure is similar to the structure of CAL, except that a bias transistor is added as an individually tunable current source to each gate. SCAL-D is an enhanced version of SCAL. Figure 2.9 shows the schematic of a SCAL-D inverter. In SCAL-D, the discharge time is shortened by adding the diode-connected transistors to provide additional current. An  $8 \times 8$  adiabatic multiplier was fabricated using SCAL-D logic in  $0.5\ \mu\text{m}$  technology with operating frequencies up to 130 MHz.

One common challenge shared by these charge-recovery logic families that we



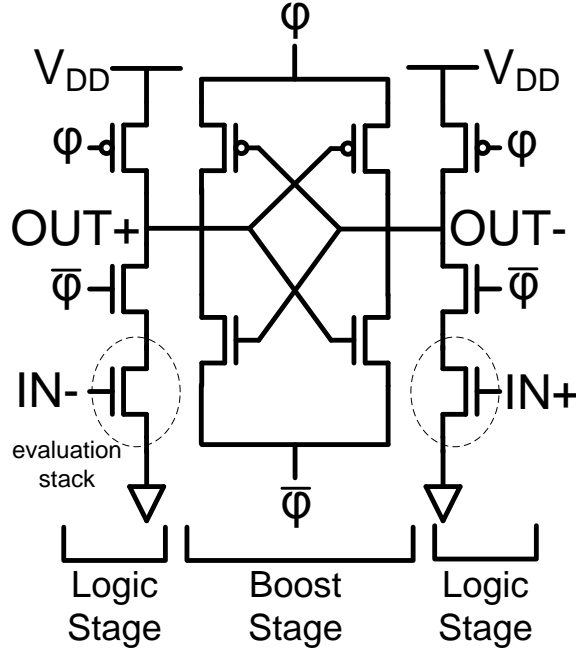


Figure 2.10: Schematic of a boost logic inverter.

stage and logic stage. Logic stage performs functional evaluation when power-clock is low, developing a near-threshold voltage difference across the differential outputs. As power-clock rises, boost stage, composed of a pair of cross-coupled inverters, amplifies this voltage difference to full-rail swing. A test-chip with eight chains of *AND*, *OR*, *XOR*, and *INV* gates was fabricated in  $0.13\ \mu\text{m}$  technology and verified at frequencies over 1 GHz.

Several variants of the boost logic have been introduced since it was proposed, such as subthreshold boost logic (SBL) [13] and enhanced boost logic (EBL) [12]. Figure 2.11 shows the schematic of a SBL inverter. Unlike Boost Logic, however, the logic stage of SBL has no clocked devices, and each of its two output rails is evaluated by a complementary all-nMOS stack, yielding reduction in crowbar current and increased gate overdrive. Figure 2.12 shows the schematic of an EBL inverter. Unlike SBL, the evaluation stage of EBL relies on a NMOS precharge device for pull-up, thus increasing performance by avoiding the series-connected devices in the pull-up network. Two FIR filter test-chips were fabricated using SBL and EBL with

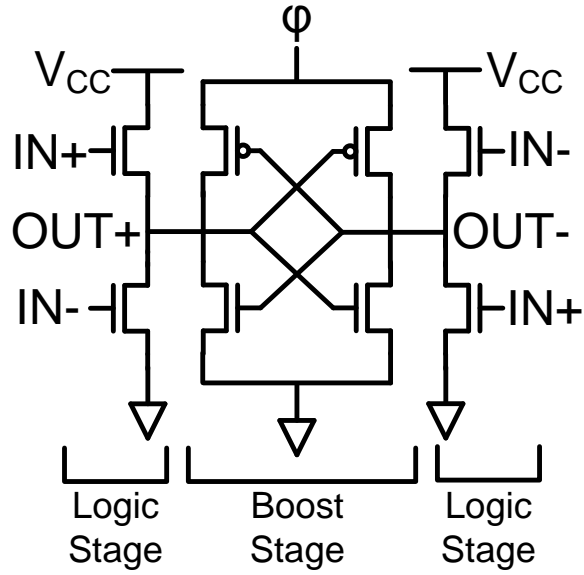


Figure 2.11: Schematic of a SBL inverter.

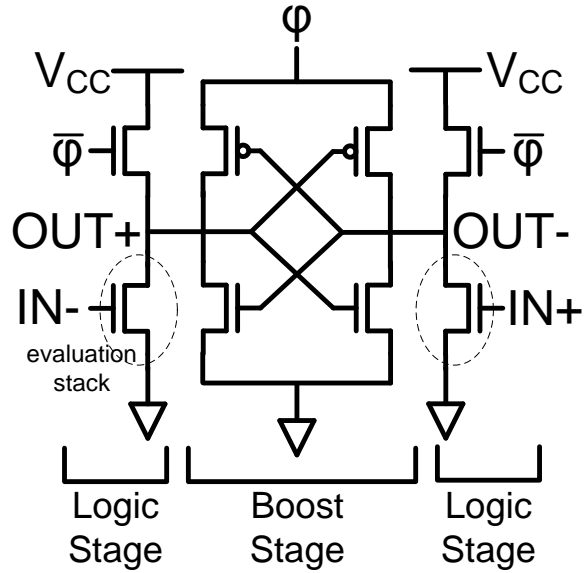


Figure 2.12: Schematic of an EBL inverter.

operating frequencies up to 187 MHz and 600 MHz, respectively.

## 2.3 Summary

Table 2.1 shows the charge-recovery logic families that have been covered in this chapter. Looking back at the evolution of prior work, several characteristics are shared

Year	1994 [16]	1995 [33]	2000 [36]	2003 [37]	2007 [11]	2009 [13]	2009 [12]
Logic Family	ADL	2N-2P	CAL	SCAL-D	Boost Logic	SBL	EBL
Technology	0.9 $\mu$ m	0.8 $\mu$ m	1.2 $\mu$ m	0.5 $\mu$ m	0.13 $\mu$ m	0.13 $\mu$ m	0.13 $\mu$ m
Core Area ( $mm^2$ )	N/A	N/A	N/A	0.47	0.1	0.38	0.34
Application	64 INV gates	1000 shift registers	732 INV gates	8b multiplier	1600 test gates	FIR	FIR
Clock Speed (MHz)	250	100	50	130	1000	187	600

Table 2.1: Overview of charge-recovery logic families covered in this chapter.

among various charge-recovery logic families. The first common trait is all previous charge-recovery test-chips were limited to small and simple designs, from chains of test gates, to adders, multipliers, and FIR filters. The limitation mainly comes from the fact that these were all full-custom designs. Another common characteristic is the area overhead created by the use of inductive elements to generate gradually-changing power-clock waveforms.

## CHAPTER III

# Design and Architecture of Charge-Recovery LDPC Decoder

### 3.1 Introduction

To explore scalable charge-recovery designs, we present a semi-automated standard-cell-like design flow that can enable the design of large-scale charge-recovery chips. To demonstrate the effectiveness of this semi-automated flow, a 576-bit rate-5/6 low-density parity-check (LDPC) decoder has been designed and evaluated using boost logic, a class of charge-recovery logic, as an example prototype.

LDPC codes, a type of capacity-approaching linear error correcting codes, have been widely used in latest communication and storage systems, including WiMAX (IEEE 802.16e) [38], 10-gigabit Ethernet (IEEE 802.3an) [39], digital video broadcasting (DVB-S2) [40], and solid-state storage [41], for their efficient and capacity-approaching performance [42, 43].

The remainder of this chapter is organized as follows: Section 3.2 presents the operation of boost logic. Section 3.3 shows the architecture and boost logic implementation of the LDPC decoder test-chip. Section 3.4 presents the on-chip power-clock generation and distribution network. A summary is provided in Section 3.5.

Part of the work covered in this chapter has been published at ISSCC 2014 [22].



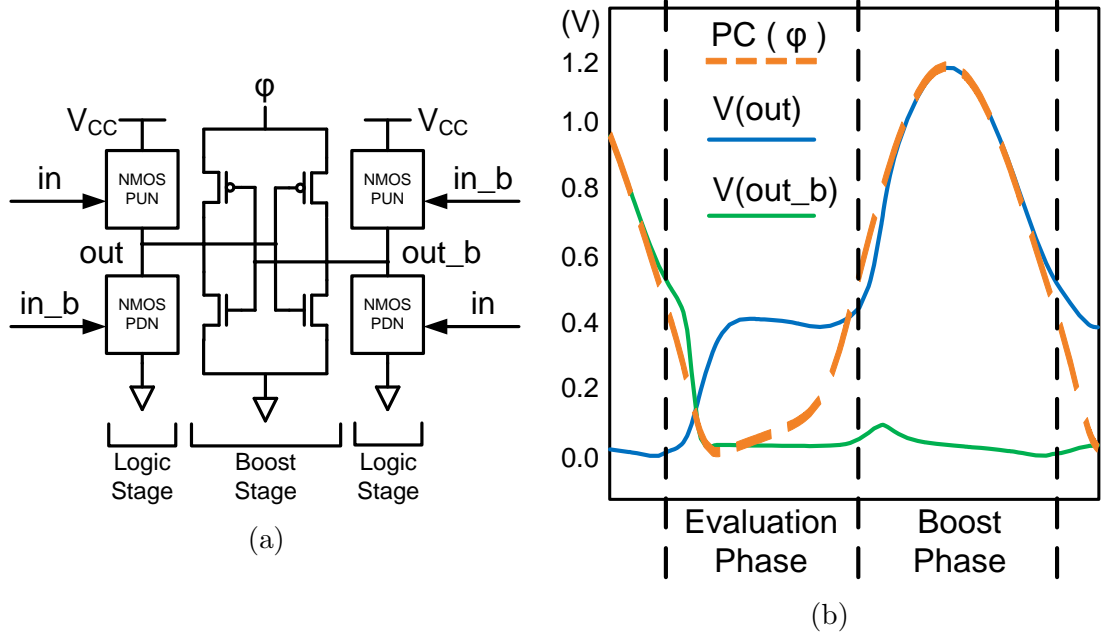


Figure 3.1: Schematic and operating waveforms of a boost logic gate.

### 3.2 Boost Logic

The LDPC test-chip has been designed using boost logic [11, 13]. This charge-recovery dynamic logic utilizes efficient signal boosting used as gate overdrive to achieve energy efficient operation at high operating frequency. Figure 3.1(a) shows the schematic of a boost logic gate. The structure of a boost logic gate can be divided into two parts, logic stage and boost stage. The logic stage has differential outputs  $out$  and  $out_b$  with a DC supply  $V_{CC}$ . Each output is driven by a pull-up network (PUN) and a pull-down network (PDN), similar to static CMOS logic, except that an NMOS PUN is used instead of PMOS to give the PUN better driving strength with subthreshold supply  $V_{CC}$ . The boost stage is composed of a pair of cross-coupled inverters connected to a charge-recovery supply power-clock ( $PC$ ) that can be generated using inductive elements. Figure 3.1(b) shows the voltage waveform of the boost logic gate. Each boost logic gate operates in two phases, evaluation and boost, which become active in turn. During evaluation,  $PC$  is low so the boost stage is off. In the meantime, logic stage performs functional evaluation, and develops

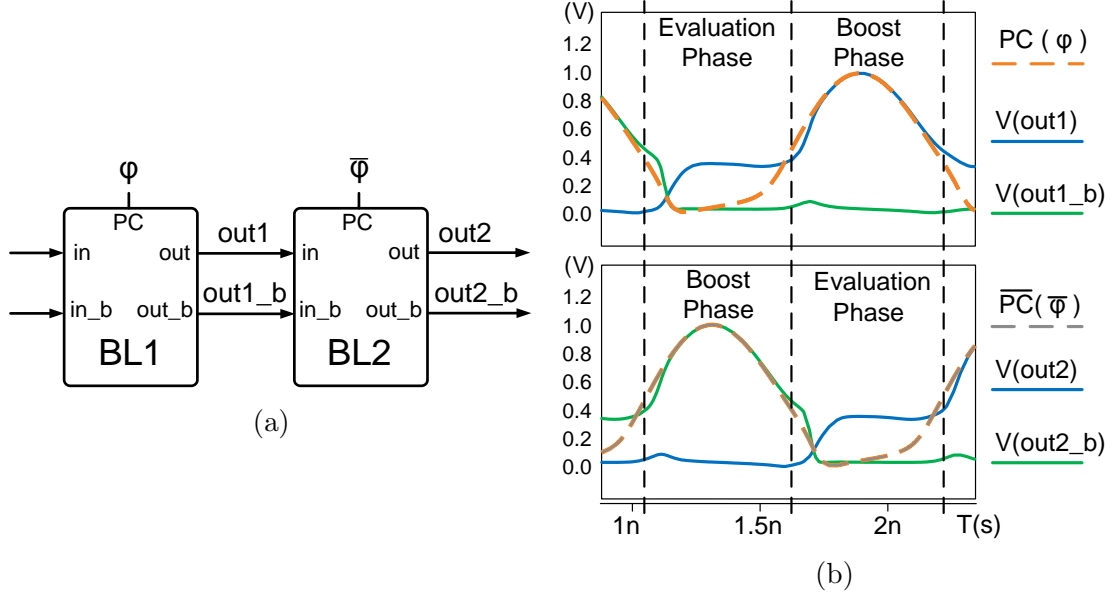


Figure 3.2: Cascade of boost logic gates, and the corresponding output waveforms.

an initial voltage difference at the output nodes. As  $PC$  rises, the gate transitions into the boost phase of its operation. During boosting, the boost stage acts as an amplifier, and one of the output nodes tracks  $PC$  and is amplified to full-rail swing. As  $PC$  falls, the charge at the output nodes is recovered through  $PC$ , and the output voltage is brought back to around  $V_{th}$ . When  $PC$  falls below  $V_{th}$ , all transistors in the boost stage are in cut-off, and the next logic evaluation phase begins.

Figure 3.2 shows the cascades of two boost logic gates and the voltage waveform of two clock phases and the output nodes. Cascades of boost logic gates are formed by clocking the gates on alternating  $PC$  phases. When  $PC$  is low, the first gate is in evaluation phase, creating an initial voltage difference. When  $PC$  rises, the first gate goes to boost phase, and  $out1$  is brought to 1V. In the meantime, the second gate is in its evaluation phase. Therefore, a significant gate overdrive can be observed at the logic stage of the second gate.

The schematic of a 2-input 4-bit boost logic comparator is shown in Figure 3.3. Each PDN and PUN has 14 transistors, and the maximum number of transistors stacked in each PDN or PUN of this gate is 5. To ensure the gates meet the target

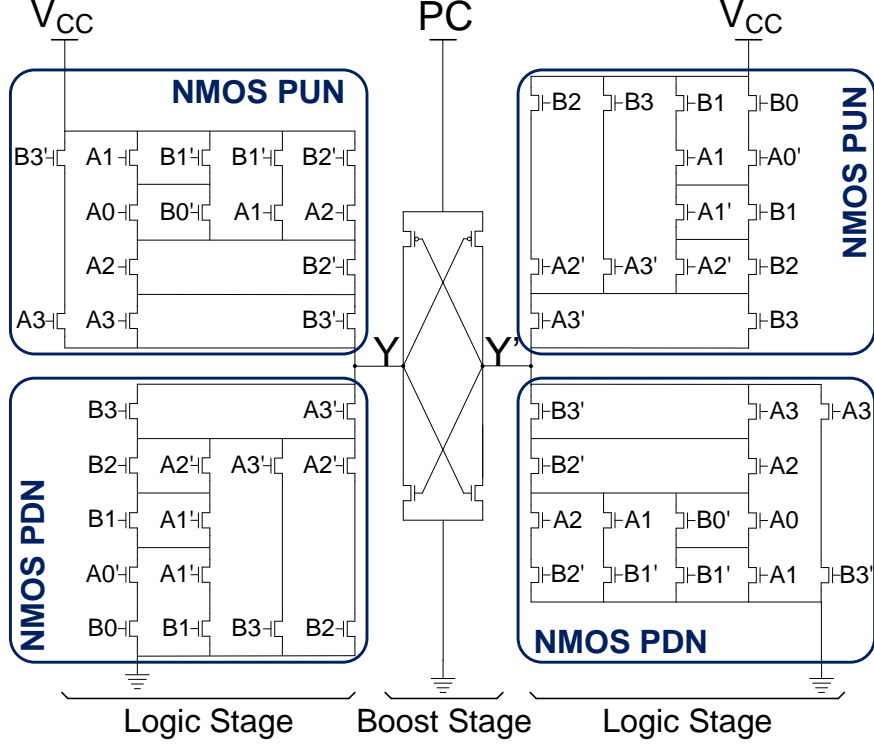


Figure 3.3: Schematic of a two-input 4-bit comparator boost logic gate.

performance efficiently without too much self-loading, the maximum stack height of a single boost logic gate is limited to 6. As a result, the test-chip has to adopt a deeply-pipelined architecture to enable the use of boost logic.

From a functional standpoint, each boost logic gate consists of a combinational logic block driving a transparent latch.  $PC$  not only provides charge to internal circuit nodes but also synchronizes the computation of the gate, which is the reason it is called power-clock.

The power-clock required by boost logic gates is generated by a power-clock generator, shown in Figure 3.4 [44]. The circuit is formed by two LC oscillators back to back, using the output waveform  $PC$  of one oscillator to drive the other, and vice versa. The capacitor comes from the parasitic cap from  $out$  and  $out.b$  nodes of boost logic gates. The cross-coupled NMOS highlighted in blue are negative transconductance devices, used to maintain the oscillation. When running off-resonance, a ring

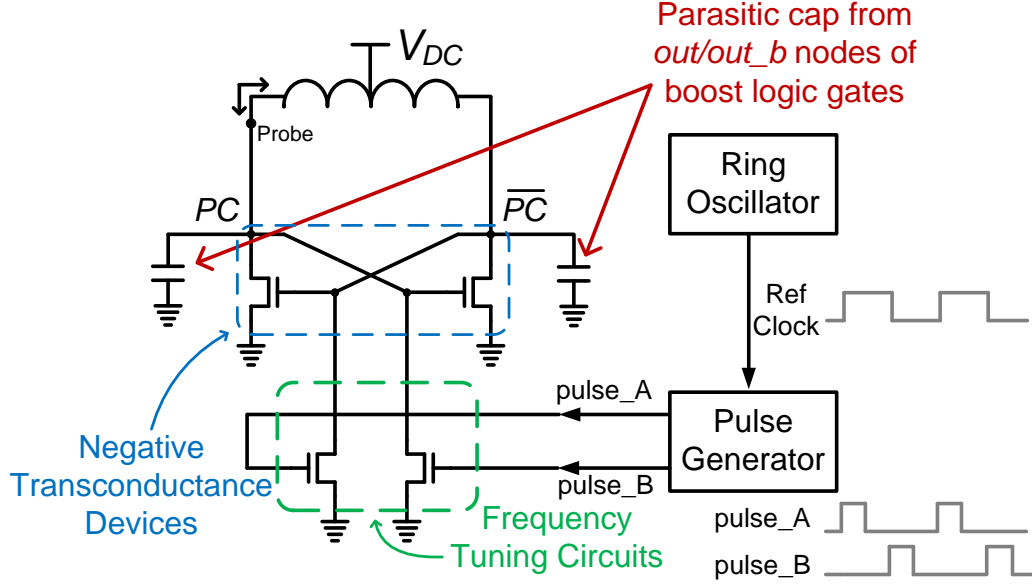


Figure 3.4: Blip power-clock generator

oscillator (RO) will first generate a reference clock at the desired frequency, feeding to a pulse generator (PG) to create two 180-degree out-of-phase pulses with programmable duty cycle. These pulses are then routed to the frequency tuning circuits, highlighted in green. Note that the power-clock generator for the decoder test-chip is entirely integrated on-chip.

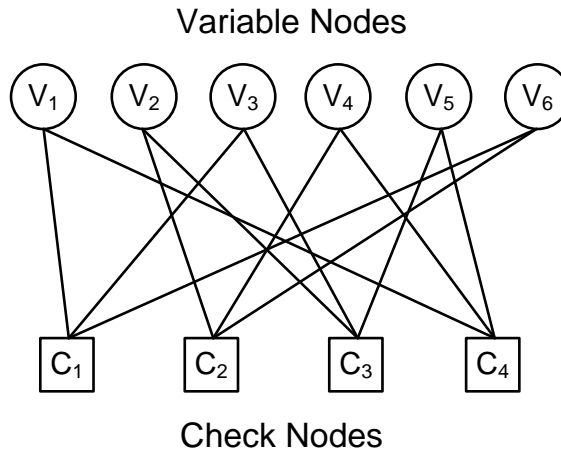
### 3.3 LDPC Decoder Architecture

To demonstrate its energy efficiency and performance potential, boost logic has been used to design a decoder for a 576-bit LDPC code.

LDPC codes are defined by a so-called H-matrix. An example of a simple H-matrix is shown in Figure 3.5(a). Each row in the matrix represents a check node (CN), and each column represents a variable node (VN). The H-matrix can be mapped to a bipartite graph called Tanner graph, shown in Figure 3.5(b), which is an equivalent representation that helps us visualize the H-matrix. There are 4 rows and 6 columns in this example matrix, and therefore 4 CNs and 6 VNs are shown in Figure 3.5(b).

Variable Nodes						
$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	
1	0	1	0	0	1	$C_1$
0	1	0	1	0	1	$C_2$
0	1	1	0	1	0	$C_3$
1	0	0	1	1	0	$C_4$

(a)



(b)

Figure 3.5: An example LDPC H-matrix and its bipartite graph representation (Tanner graph).

The matrix entries are 1s and 0s. If there is a 1, the corresponding CN and VN in the graph are connected; otherwise they are not. So, the H-matrix on the left is a one-to-one mapping of the graph on the right.

The computation of LDPC decoding is mainly done in the VNs and CNs. Each VN receives a 5-bit signal from the receiver through the channel as its initial value. Check node operation performs even parity-check and finds the minimum among the connected VNs, and variable node operation sums the outputs of the connected CNs.

0	6	13	.	11	1	.	22	21	2	21	13	20	8	1	0	9	5	1	19	20	0	.	.
.	1	.	9	10	11	3	19	11	.	10	5	3	17	3	18	0	11	12	0	0	0	0	.
12	20	20	1	16	.	5	.	7	6	22	15	20	2	21	19	15	22	16	3	.	.	0	0
12	.	12	3	.	9	3	2	2	5	13	22	7	23	14	7	21	23	2	16	20	.	.	0

Figure 3.6: The 576-bit rate-5/6 LDPC matrix specified by IEEE 802.16e standard.

Our chip implements a decoder corresponding to a rate-5/6 576-bit LDPC matrix specified by IEEE 802.16e WiMAX standard [38]. The matrix is shown in Figure 3.6. Recent LDPC applications have adopted matrices that are constructed using submatrices that are cyclic shifts of an identity matrix or a zero matrix. Each number in this matrix represents a 24-by-24 identity sub-matrix that is circular right-shifted by that number; otherwise, it is a 24-by-24 zero matrix. 24 columns are shown in this H-matrix, and each entry has 24-by-24 elements. Therefore the number of variable nodes is  $24 \times 24 = 576$ . Similarly, 4 rows are shown in the H-matrix, so the number of check nodes is 96. The rate denotes the ratio of the number of the information bits over the number of total encoded bits, including the redundant bits which are added to increase the chance of recovering from errors. Lower rate means more redundant bits are added. Therefore, a rate-5/6 576-bit LDPC code encodes a 480-bit message into 576 bits.

The decoder has been implemented using a row-parallel architecture, which provides high throughput with manageable routing complexity, allowing high energy efficiency and high area efficiency. The belief propagation algorithm has been implemented using a min-sum algorithm with offset correction [45] and flooding schedule [46]. The performance of the LDPC decoder we implemented with a 5-bit fixed-point representation for 10 decoding iterations is shown in Figure 3.7.

To reduce global communications and accommodate the deeply-pipelined architecture required by the dynamic logic (explained in Section 3.2), the decoder (or

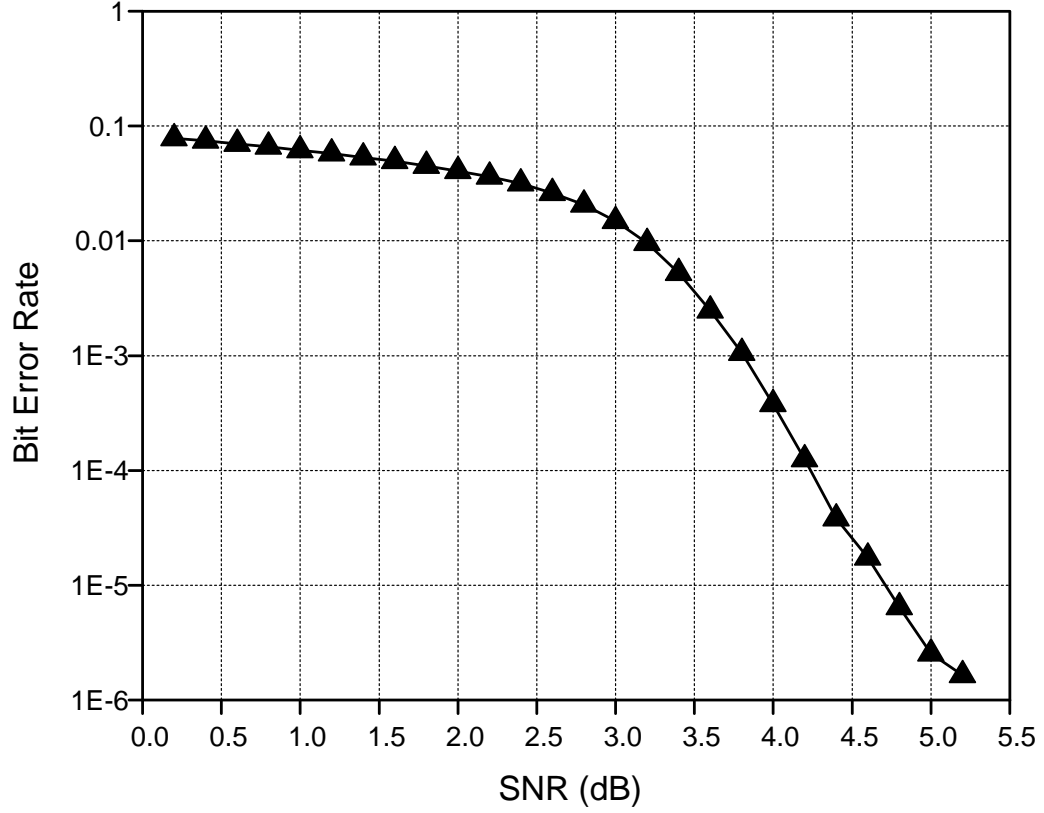


Figure 3.7: Bit error rate with different SNR of the implemented 576-bit rate-5/6 LDPC matrix specified by IEEE 802.16e standard.

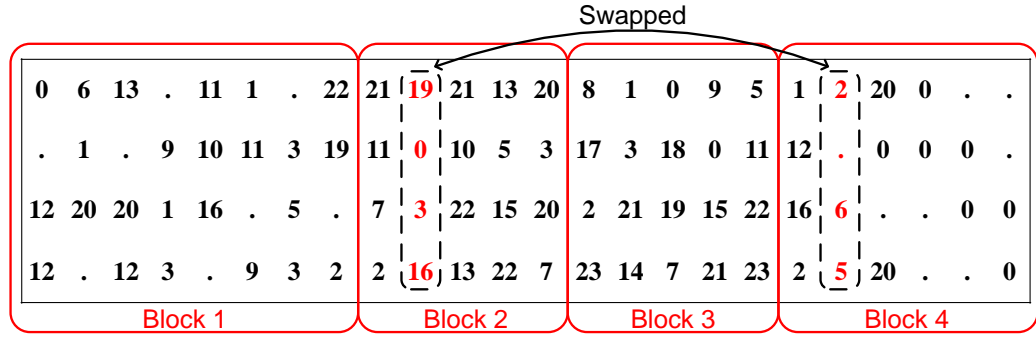


Figure 3.8: LDPC matrix swapping and partitioning.

equivalently the matrix) is partitioned into four blocks. Figure 3.8 shows the partitioned and rearranged matrix. The partitioning is done in a way that balances the computation of each block. For each of the 96 rows in the matrix, a check node operation finds the minimum value in that row. Therefore, the check node operation of Block 1 in Figure 3.8 finds the minimum among the values of 6 nonzero submatrices.

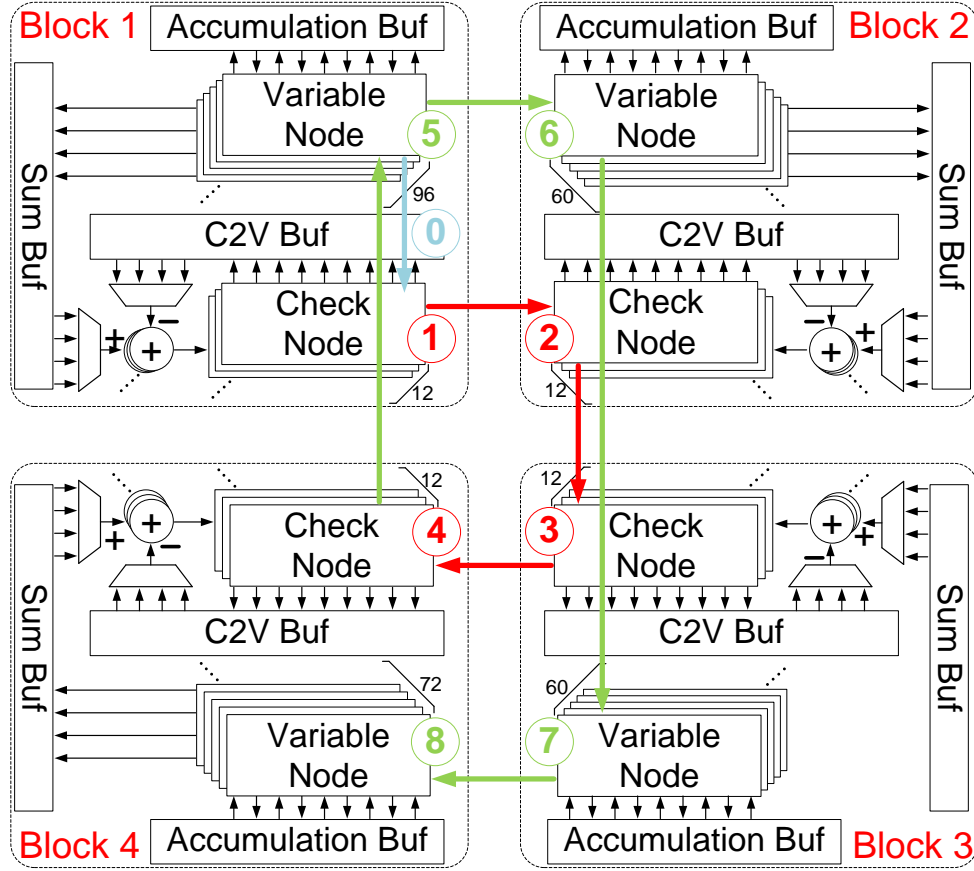


Figure 3.9: LDPC decoder architecture.

This minimum from Block 1 is then passed to Block 2, which has 5 nonzero submatrices. Therefore, the check node operation of Block 2 again finds the minimum of the values from 5 submatrices in Block 2 and the minimum from Block 1. Similarly, the check node operation in Block 3 and 4 compare 6 and 5 values, respectively. Therefore, each block performs similar amount of computation and takes similar silicon area.

To facilitate partitioning, two columns in the matrix of Figure 3.8 are swapped to increase regularity. This rearrangement ensures that within each block, the number of nonzero submatrices in each row is identical. Other than changing the in/output connections, this column-swapping does not result in any overheads in terms of operating speed or decoding performance.



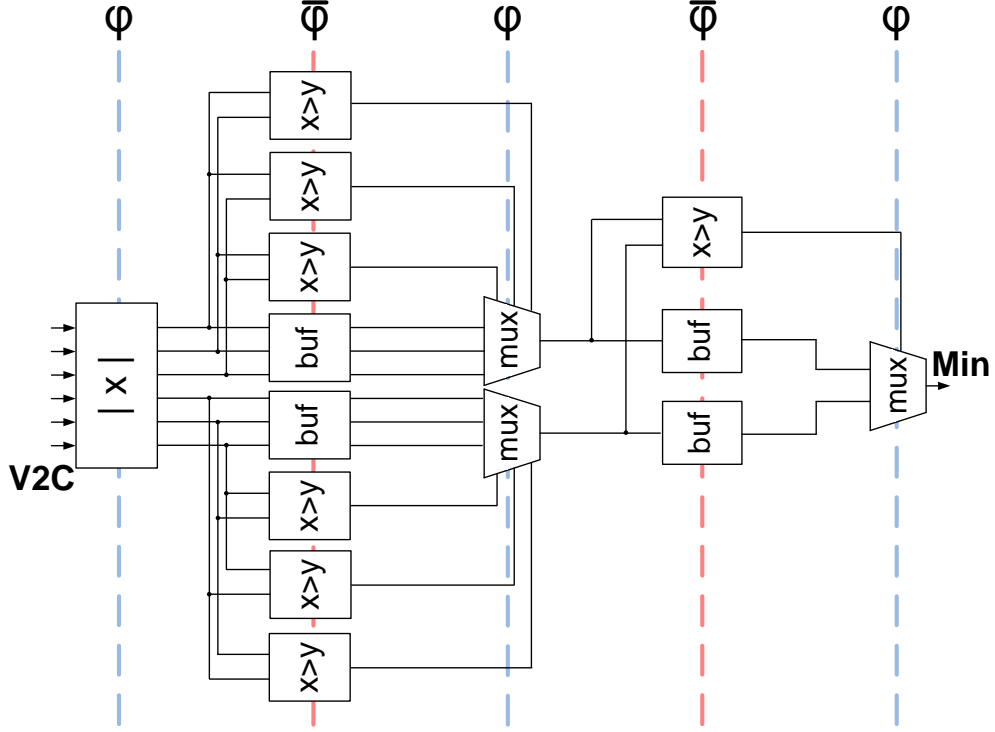


Figure 3.10: Gate-level block diagram of check node operation.

Figure 3.9 shows the block diagram of the decoder architecture. The computation of LDPC decoding is mainly done in the VNs and CNs of each block. To handle such a complex matrix, the block architecture is designed with time multiplexing. Therefore, 96 VNs and 12 CNs are shown in Block 1. The decoding operation begins from the VNs in Block 1. The VNs receive signals from the channel as their initial values, and then pass these initial VN-to-CN messages to the CNs. Block 1 then performs CN operation. Results are relayed to Block 2, Block 3, and Block 4 in order, reducing the global communication to local communication. VN operations are interleaved with CN operations, so after Block 4 finishes its CN operation, the results, which are CN-to-VN messages, are passed to the VNs in Block 1, and Block 1 computes the VN operation. The results are also relayed to Block 2, 3 and 4 in order. One complete decoding iteration of the entire H-matrix takes 24 cycles, or 48 phases. This deeply-pipelined relay architecture not only is able to accommodate our dynamic logic and reduce complex global communication but also allows us to operate with 4

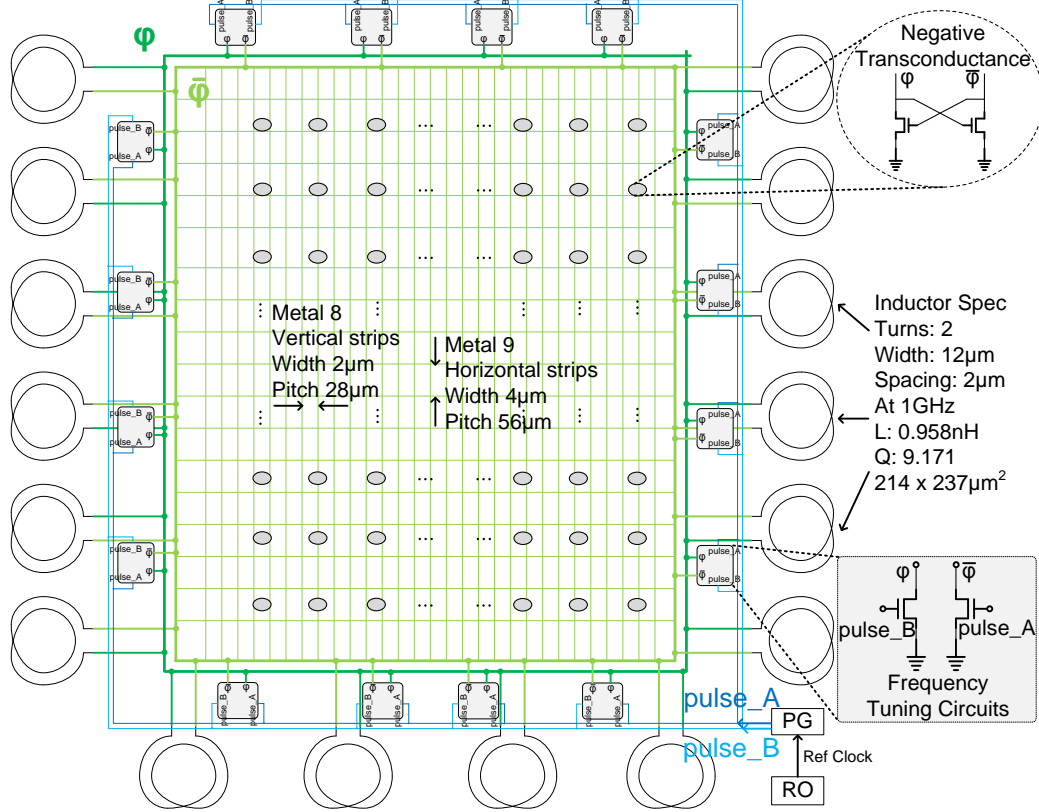


Figure 3.11: Power-clock generation and distribution network.

different sets of data in parallel without any pipeline stalls. To provide some insight into how the architecture is designed for boost logic, Figure 3.10 shows the gate level block diagram of an example CN operation in Block 1. The gates are cascaded using alternate power-clock phases,  $PC$  and  $\overline{PC}$ , and it takes 5 clock-phases, or 2.5 cycles, to complete the CN operation of a block. Note that boost logic buffers are inserted for phase alignment, as in any other dynamic logic. 54% of the gates in our decoder test-chip are boost logic buffers.

### 3.4 Power-Clock

The design of the power-clock generator and the power-clock distribution network plays a key role in the efficiency of charge-recovery chips. Figure 3.11 shows the on-chip power-clock generator and distribution network. The decoder chip is fabricated in

65nm CMOS technology with 9 metal layers. 16 center-tap on-chip inductors are used to resonate the design and recover charge from gate fanouts to achieve energy-efficient operation. Spirals are mainly formed by metal 9. Each inductor gives 0.96 nH at 1 GHz with quality factor 9.17 when simulating using a commercial 3D electromagnetic field simulator.

To minimize clock skew and enable a semi-automated design flow, clock meshes are used to distribute the two-phase power-clock. Top level metals are used for the clock meshes, metal 9 for horizontal strips, and metal 8 for vertical strips. To connect the  $PC$  pin of each boost logic gate to the mesh, for each standard-cell row, two metal-3 horizontal strips are reserved for the power-clock waveform  $PC$  and  $\overline{PC}$ . These strips are tied to clock meshes (metal 8) directly, so that we can connect each boost logic gate to the mesh in a predictable manner using commercial automated place-and-route tools while avoiding any possible large clock skew. 144 negative transconductance devices, which are pairs of cross-coupled NMOS switches ( $7,344\ \mu\text{m}$  total width for each phase), are distributed across the core to maintain the oscillation.

To operate the decoder chip off-resonance, a ring oscillator is used to first generate a reference clock signal at a desired frequency. This reference clock is then fed to a pulse generator, outputting a pair of 180-degree out-of-phase pulses with programmable duty cycle. These pulses are distributed by a tree structure with supply and ground shielding to 16 frequency tuning circuits (with programmable NMOS width from  $800\ \mu\text{m}$  to  $5,600\ \mu\text{m}$  in  $800\ \mu\text{m}$  steps) to operate the decoder at a desired frequency.

The efficiency of charge-recovery chips relies on the design of the power-clock generator and the power-clock distribution network. Figure 3.12 shows the energy consumption of the power-clock obtained from one of the 16 inductors through device-level Spice simulation of a full block ( $600\ \mu\text{m} \times 600\ \mu\text{m}$  block size) with the verified

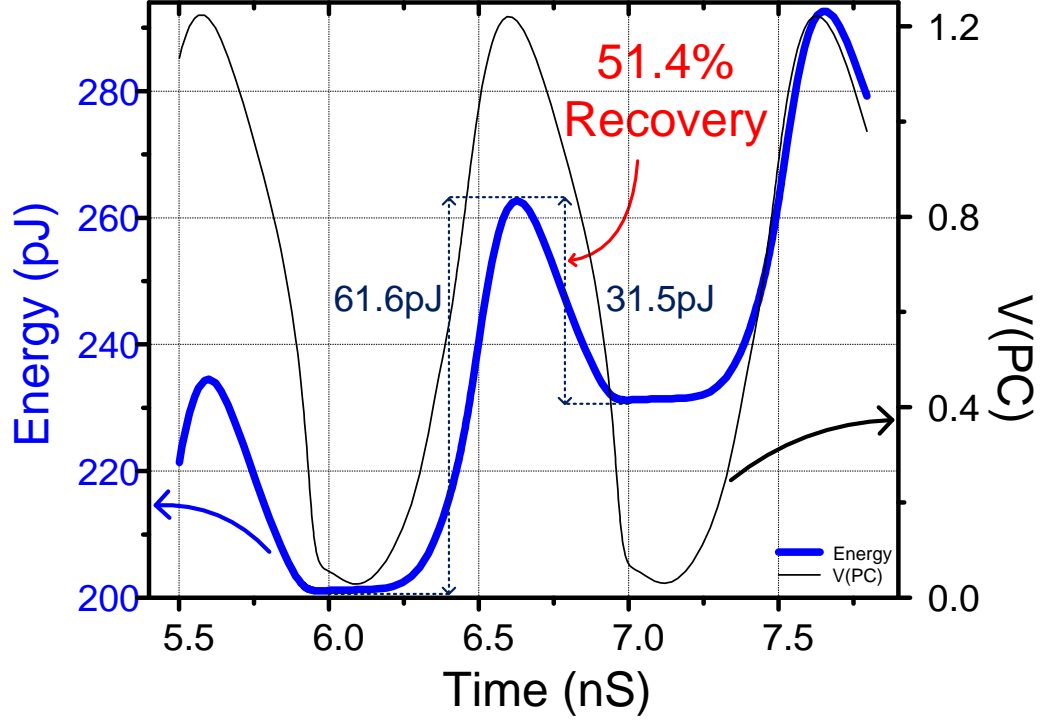


Figure 3.12: Simulated waveform of energy supplied to power-clock through one of the sixteen inductors.

inductor models from the foundry and an extracted post-layout netlist including parasitic resistance, capacitance, and grounded coupling capacitance. Probing at the node shown in Figure 3.4, as *PC* rises, 61.6 pJ of energy are delivered. When *PC* falls, 31.5 pJ are recovered, yielding a 51.4% recovery rate.

Clock skew in the power-clock mesh resulting from our semi-custom design flow has been assessed through simulations. To that end, the *PC* pins of 400 boost logic gates across one of the four blocks are probed, performing a full-block SPICE simulation with four verified inductors connected from four sides of the block and extracted post-layout netlist. Figure 3.13 shows the histogram of clock skew across the 400 probed nodes. The maximum clock skew is 11.3 ps, and the average skew is 2.55 ps, when *PC* self-oscillates at 1 GHz. For 99% of the nodes, clock skew falls below 7 ps. Figure 3.14 shows the distribution of clock skew across the block.

Balancing the loading of the two-phase power-clock is important to obtain a power-

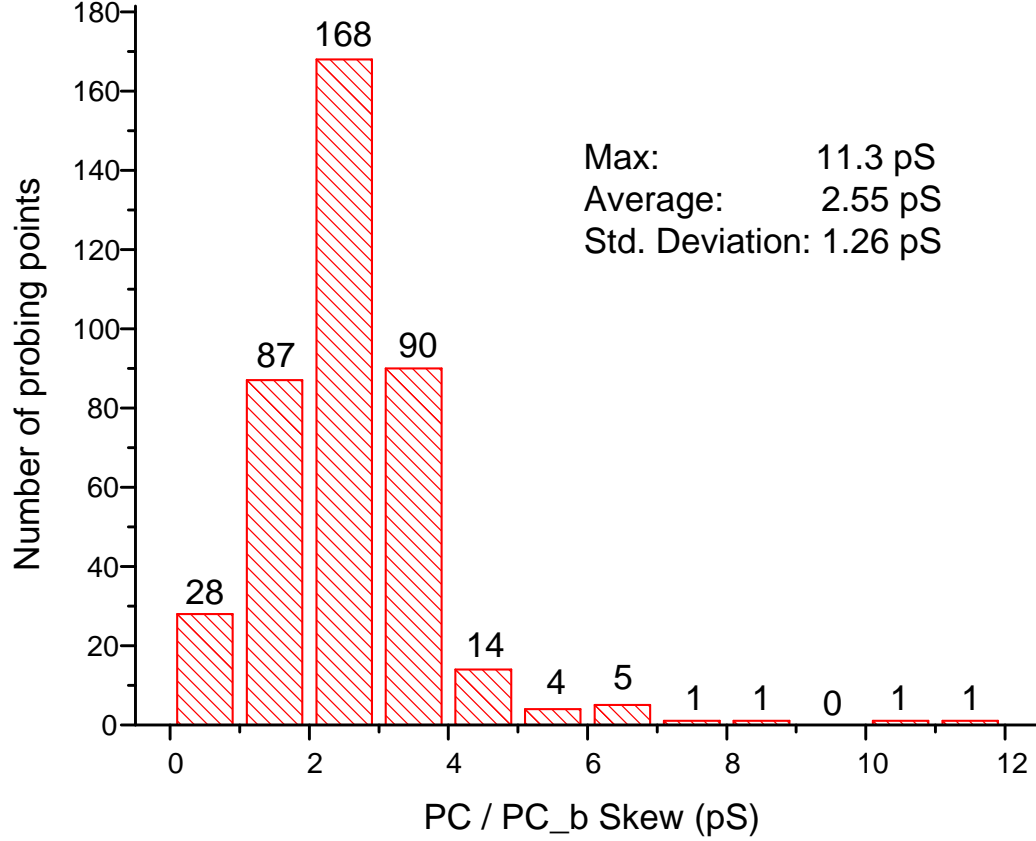


Figure 3.13: Histogram of clock skew from full-block spice simulation.

clock waveform with similar voltage swing for the two phases. Figure 3.15 shows power-clock waveforms for the two power-clock phases from a spice simulation with extracted netlists of an early implementation of a block. Loading on  $\overline{PC}$  is larger than that on  $PC$ , yielding a 5% reduction in the voltage swing of  $PC$  in comparison with  $\overline{PC}$ . This loading imbalance mainly comes from the fact that some pipeline stages have high fanouts. For example, in this unbalanced design, the 5-bit two-phase adder has 21 gates, 16 for the first phase and 5 for the second phase. To balance the loading of the two clock phases, we have repeated the logic synthesis design step and manually optimized the adder into 18 gates, 13 for the first phase and 5 for the second phase.

Figure 3.16 shows waveforms for two power-clock phases, after optimizing the design, from a spice simulation of the optimized block with extracted netlists. The voltage swing of  $PC$  and  $\overline{PC}$  differs by less than 0.7%. Note that the difference

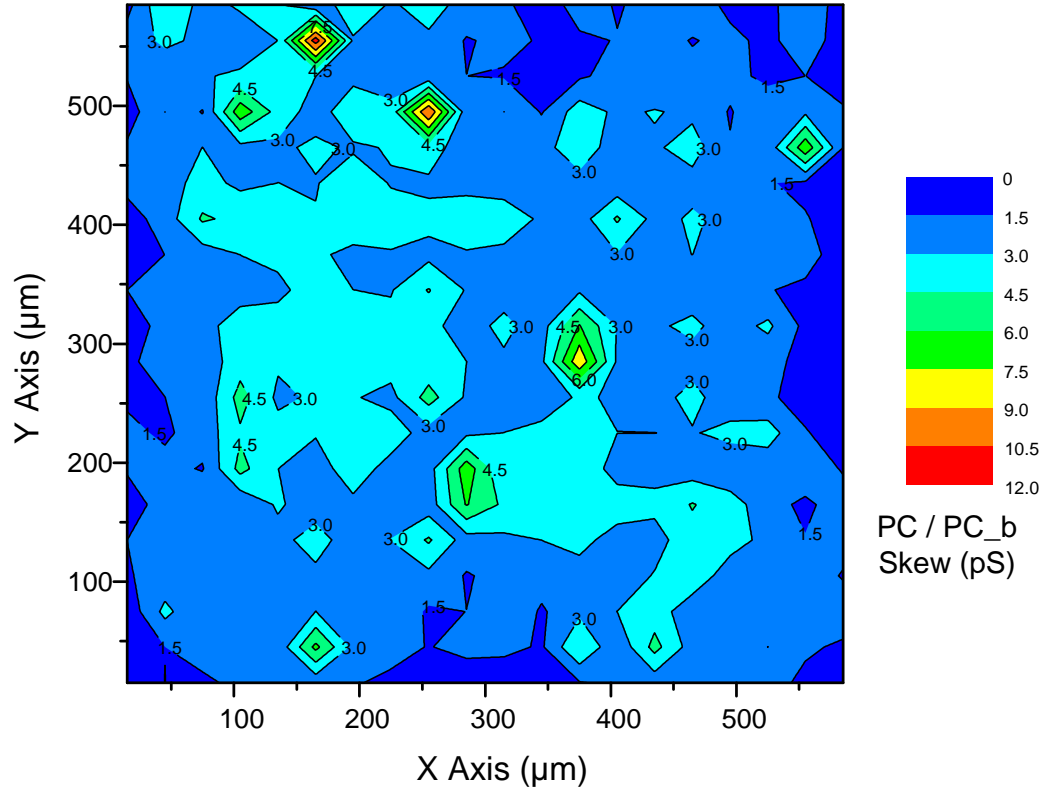


Figure 3.14: PC skew distribution across Block 3 of LDPC die.

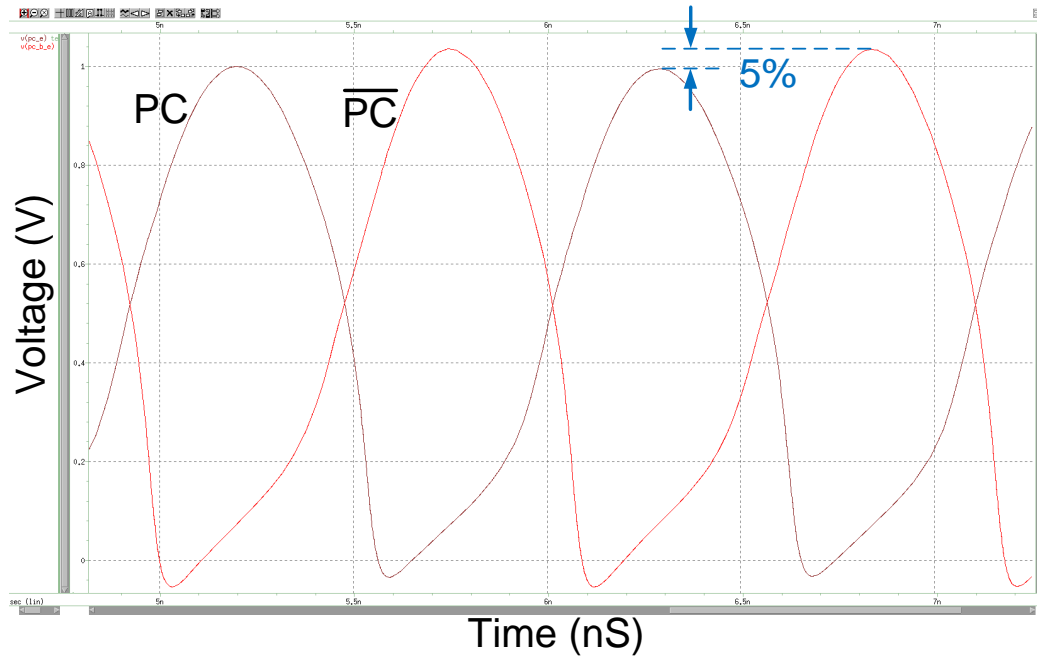


Figure 3.15: Voltage waveform of two-phase power-clock with unbalanced loading for the two phases.

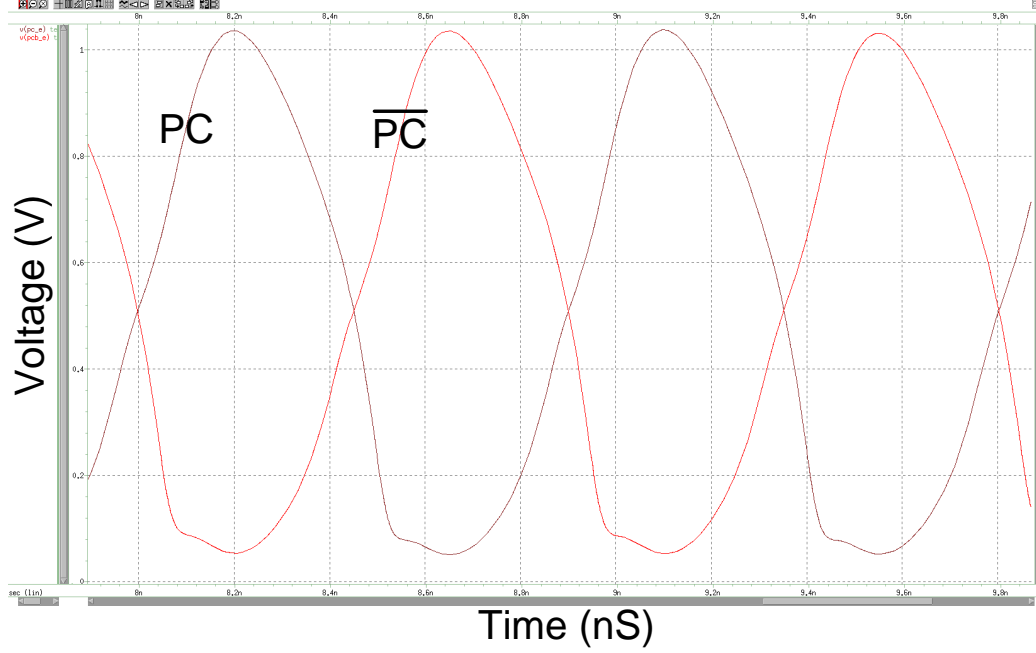


Figure 3.16: Voltage waveform of two-phase power-clock with comparable loading for the two phases.

between the bottom part of Figure 3.15 and Figure 3.16 is the result of using ideal and real negative transconductance devices in the two simulations, respectively, and it is not due to the balancing optimization.

### 3.5 Summary

This chapter presents the design and architecture of our rate-5/6 576-bit charge-recovery LDPC decoder. The decoder is designed using boost logic, a charge-recovery dynamic logic that utilizes efficient signal boosting as gate overdrive to achieve high performance energy efficient operation. To reduce global communications and accommodate the deeply-pipelined architecture required by our dynamic logic, the decoder is partitioned into four blocks, and messages are relayed between neighboring blocks. The design of the power-clock generator and the clock distribution network plays a key role in the efficiency of charge-recovery chips. 16 on-chip inductors are used to resonate the design. Clock meshes are used to distribute the two-phase power-clock

to minimize clock skew and enable a semi-automated design flow. From simulation, the chip recovers 51.4% of the energy supplied to it when self-oscillating at 866 MHz, with average clock skew of 2.55 ps and worst-case clock skew of 11.3 ps.



## CHAPTER IV

# Design Methodology

### 4.1 Introduction

The characteristics of recent silicon-demonstrated charge-recovery chips [11–14] are given in Table 4.1. One common characteristic of all these previous charge-recovery chips is that they are all limited to small and relatively simple designs. The limitation mainly comes from the fact that they are all full-custom designs.

With over 57,000 boost logic gates, the device count of our chip is more than an order of magnitude larger than the largest previously-reported charge-recovery test-chips. To manage the complexity of such a design with boost logic, we have developed a standard-cell-like semi-automated design flow that incorporates custom-designed dynamic cells from a library and a two-phase power-clock.

The remainder of this chapter is organized as follows: Standard-cell design methodology is presented in Section 4.2, showing the design flow for common digital static CMOS implementations using standard cells. Section 4.3 explains the standard-cell-like semi-automated design flow that we have developed. Section 4.4 compares the differences between two flows. A summary is provided in Section 4.5.

	This Work	JSSC '07 [11]	TVLSI '12 [12]	JSSC '10 [13]	ASSCC '11 [14]
Technology	65nm	0.13 $\mu$ m	0.13 $\mu$ m	0.13 $\mu$ m	65nm
Application	LDPC	Chains of test gates	FIR	FIR	Processing node for LDPC
Frequency Range (MHz)	408 - 1,049	700 - 1,100	365 - 600	5 - 187	404 - 609
Active Area (mm <sup>2</sup> )	1.54	0.02	0.34	0.38	0.04
Transistor Count (x1000)	1,297	N/A	N/A	41	1.64
Gate Count	55,305	1,680	3,330	N/A	N/A

Table 4.1: Comparison with recent silicon-demonstrated charge-recovery test-chips.

## 4.2 Standard-Cell Design Methodology

In digital VLSI design, a standard-cell design methodology is a method of designing integrated circuits using a standard-cell library. This methodology significantly reduces the design effort for digital designers to implement application-specific integrated circuits (ASIC) from high-level system specifications to fabrication-ready physical layout, enabling designers to scale relatively simple designs with just several hundreds transistors to complex systems with millions of devices.

Figure 4.1 shows a brief typical design flow using a standard-cell design methodology. Designers are first given system specifications, based on which they start register-transfer level (RTL) design. In RTL design, designers declare the registers of a design, describe the combinational logic indicating the possible transfers and operations, and define when to transfer and operate on data. Depending on the application, designers may need to explore possible architectures before converting specifications into RTL descriptions. For example, questions like how many pipeline stages are required, and where to insert pipeline registers have to be answered before generating RTL. Verification is performed to confirm that the RTL code is consistent

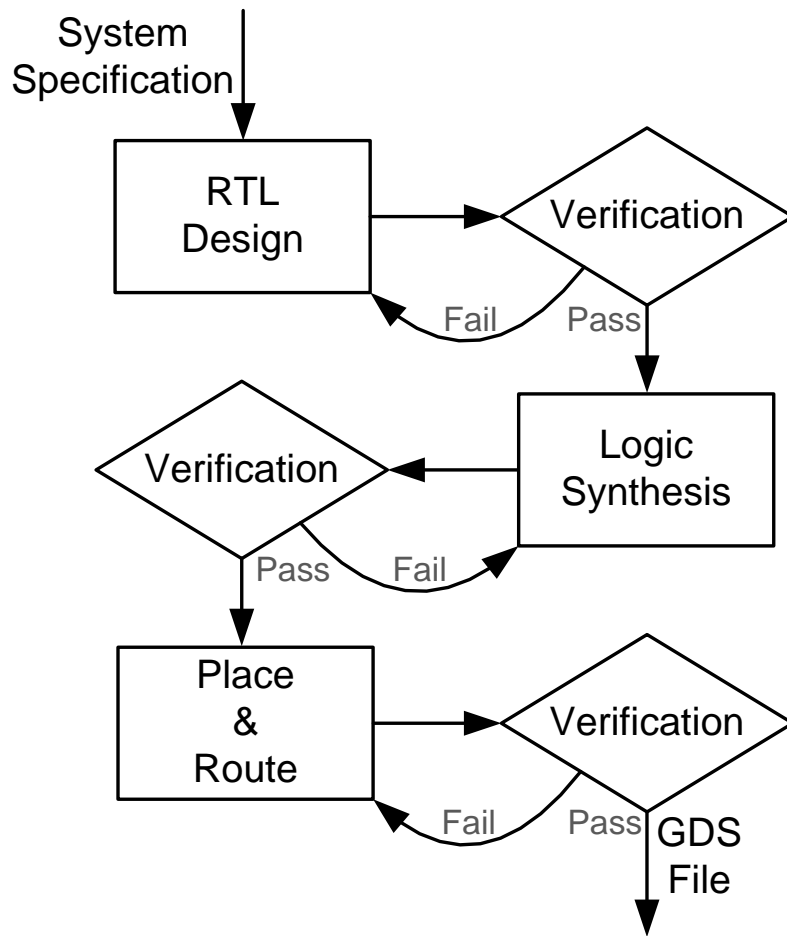


Figure 4.1: Standard-cell design methodology.

with the system specifications, before moving to next design step—logic synthesis.

Logic synthesis is the process of converting RTL code into a gate-level implementation using a standard-cell library. Figure 4.2 shows the inputs and outputs of synthesizing a gate-level netlist from RTL code using an automated logic synthesis tool. A standard-cell library is a collection of standard cells, usually developed and provided by fabrication foundries for every technology process they provide. Standard cells included in the library are basic low-level logic functions recommended for implementing ASIC, such as INV, AND, OR, latches, flip-flops, and also macro cells, such as adders or even multipliers. These standard cells are characterized and

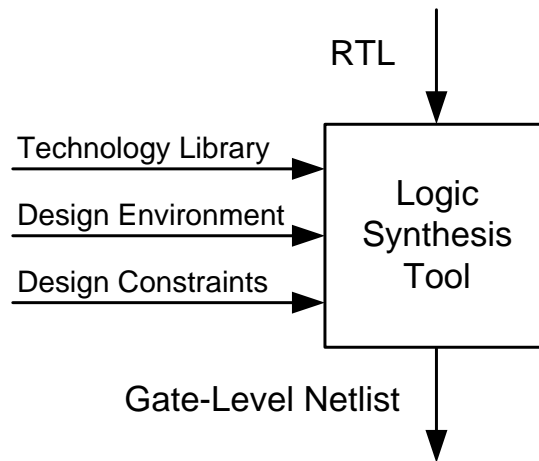


Figure 4.2: Inputs and outputs for synthesis.

provided to electronic design automation (EDA) tools, such as a synthesis tool or a place-and-route tool, as a technology library. Design environment, process parameters, operating conditions, and design constraints, such as clock period, rise and fall clock transition times, area, and power, are provided to the tool to generate a gate-level netlist from RTL code. The functionality of the generated netlist is verified before moving to the place-and-route stage.

Figure 4.3 shows the main steps of the place-and-route stage using a place-and-route tool. Before placing cells, floorplanning is performed for design partitioning, power planning, pin placement, and macro placement. Next the tool is used to place the gates in the netlist. The tool iterates trial placement and trial routing several times to find the optimum placement and meet design constraints. After placement, the tool is used to implement a clock tree based on the timing constraints of the design. Last, the tool connects and routes signals. After place-and-route, the tool performs static timing analysis to make sure all timing constraints are met. If not, it goes back to repeat previous steps and try to close timing. A final layout file is then generated by the tool.

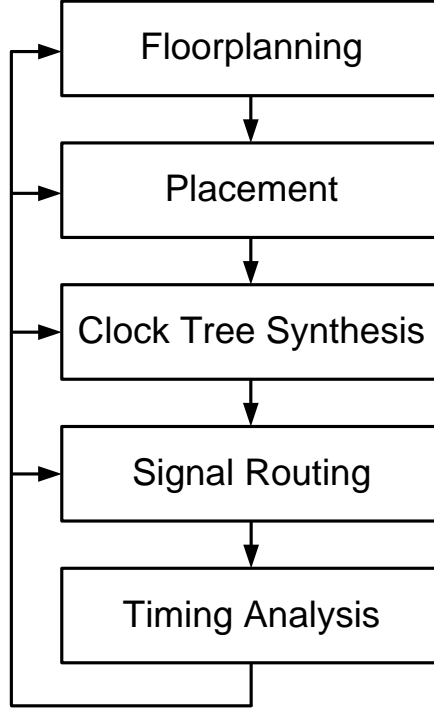


Figure 4.3: Place-and-route flow.

## 4.3 Semi-Automated Design Methodology for Charge-Recovery Logic

One common characteristic of all previous charge-recovery chips is that they are all limited to small and relatively simple designs. The limitation mainly comes from the fact that they are all full-custom designs. To manage the complexity of such a design with boost logic, we propose a standard-cell-like semi-automated design flow that incorporates custom-designed dynamic cells from a library and a two-phase power-clock.

### 4.3.1 Front-End Design

Our design methodology uses a cell library of boost gates. The functions in the library can be determined based on the target design using a commercial synthesis tool. For our silicon prototyping, we designed a library with 52 different boost logic

functions, and 2 to 6 different drive strengths per function. In our library, the PDN of a single boost logic gate can contain up to 25 transistors. The schematic of a two-input four-bit boost logic comparator is shown in Figure 3.2.

Synthesis proceeds by first developing a RTL description of the design. The RTL netlist is then partitioned into major design blocks, and a commercial synthesis tool is used to partition these major design blocks into pipeline stages. As described in Section 3.2, the boost logic cells in our design have a maximum stack height constraint, which is considered by the tool when partitioning the design into pipeline stages. Basic standard cells, such as AND, OR, and INV gates, are provided to the tool, and each cell is set with a unit delay. The tool synthesizes the design with a clock period equal to the maximum stack height (6 unit-delay if the maximum stack height is 6). Each stage of the synthesized netlist is then converted and optimized into boost logic cells manually [47]. After the design is partitioned into pipeline stages, phase-aligning buffers are inserted to balance the pipeline stages of the decoder [48].

#### 4.3.2 Cell Library Design

The layout of each boost logic cell is designed to be used by commercial back-end EDA tools. All cells share the same height of  $5\mu\text{m}$ , and the width of a cell has to be a multiple of  $0.2\mu\text{m}$ . Figures 4.4 and 4.5 show the layout view of a two-input 4-bit comparator boost logic gate with two different drive strengths. The cross-coupled inverter pair in the boost stage is placed on the right, with PMOS on the top and NMOS on the bottom, and the transistors in the logic stage are placed on the left. The transistor sizes of the stronger gate shown in Figure 4.5 are larger for both PMOS in the boost stage and NMOS in the logic stage. Metal 1, metal 2, and metal 3 are used for local cell routing. However, metal 1 has different minimum pitch and width from metal 2 and metal 3, so all signal pins are extracted with only metal 2 and metal 3 for global routing to remove any possible design rule check (DRC) violations. Each

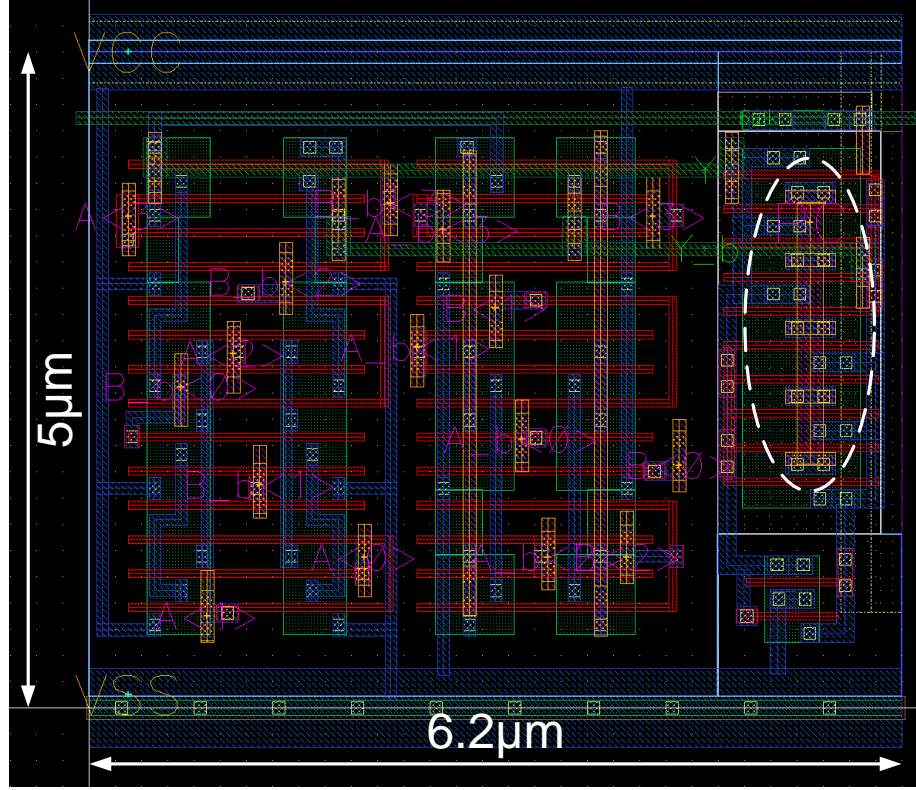


Figure 4.4: Layout view of a two-input 4-bit comparator boost logic gate with weaker drive strength.

signal pin occupies its own metal 2 or metal 3 track to make the global routing easier when using commercial EDA tools for place-and-route. To reduce the resistance of *PC* distribution network, *PC* pins, highlighted in Figure 4.4 and Figure 4.5, are connected with at least  $2\times$  minimum width, both locally within the cell and globally when performing chip-level place-and-route.

Library gates have been characterized to be used with commercial EDA tools. Each gate has been designed and characterized for a target output loading to operate at a target frequency (1 GHz). For example, the gate shown in Figure 4.4 is designed to drive up to 10 fF loading, and the gate shown in Figure 4.5 can drive up to 30 fF loading. The target output loading of each gate, along with the gate capacitance of each input, are specified in an industry-standard liberty format (.lib). This file is then provided to a commercial EDA tool for place-and-route.





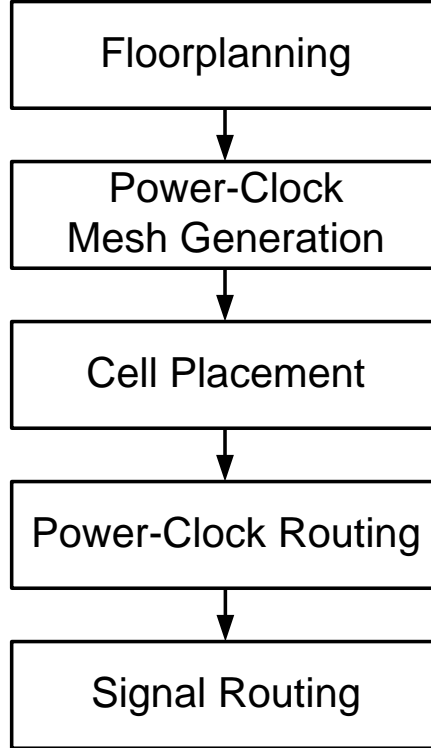


Figure 4.6: Place-and-route flow for charge-recovery logic.

connected to the mesh in a predictable manner during detailed routing while avoiding large clock skew. During place-and-route, the target output loading of each gate is set as the main constraint for place-and-route. To close timing, each function in the library has several drive strengths with different target output loading the tool can choose from. The tool iterates through trial place-and-route configurations, choosing among cells with different drive strengths to meet timing constraints.

Figure 4.7 briefly shows the place-and-route flow to incorporate the two-phase power-clock and custom-designed boost logic cells. Supply grids are first created using a commercial EDA tool,  $V_{CC}$  and  $V_{SS}$  on metal 1, and  $PC$  and  $\overline{PC}$  on metal 3. These horizontal strips are directly tied to the top level meshes. The height of a cell row is  $5\ \mu\text{m}$ . The tool is then used to do the initial placement, trial-routing, and for optimizing the initial placement. After placement, the tool is used to route the  $PC$  and  $\overline{PC}$  signals by connecting the  $PC$  pin of each cell to either the top  $PC$  horizontal

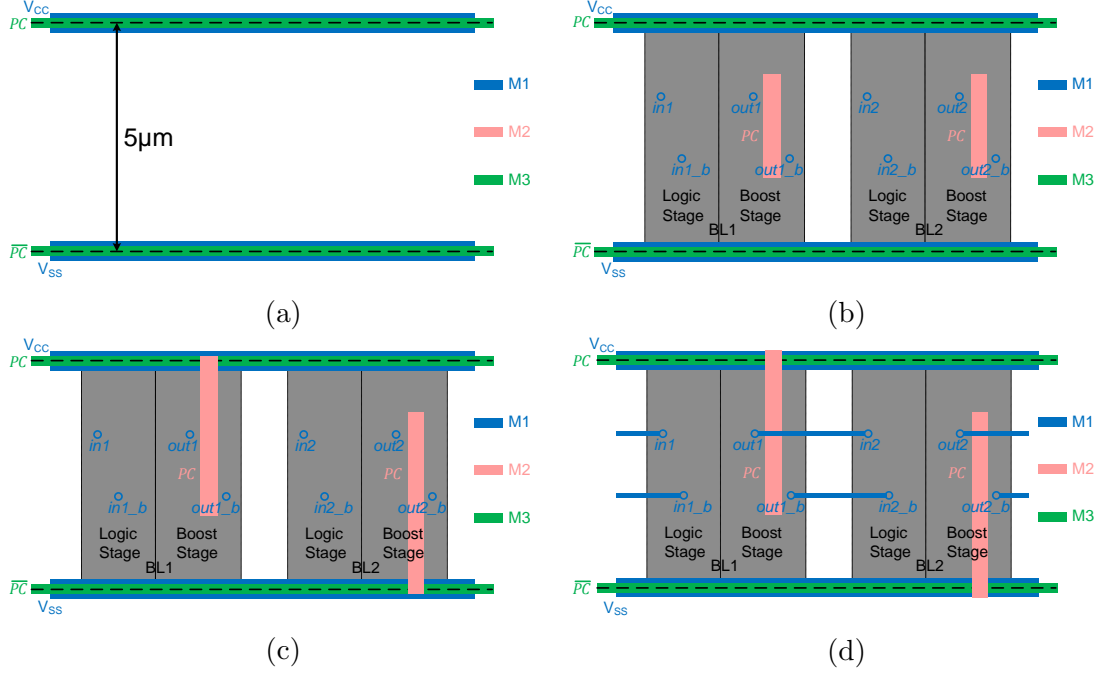


Figure 4.7: Simplified automatic place-and-route flow.

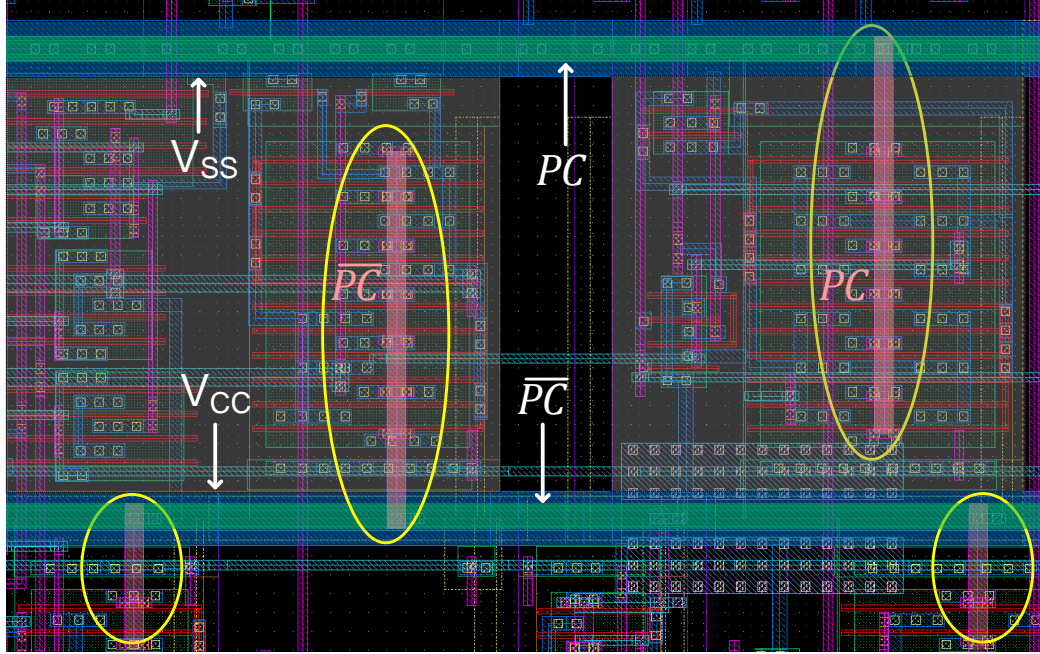


Figure 4.8: An actual layout view of the decoder chip after APR.

strip or bottom  $\overline{PC}$  strip accordingly. Last, the rest of the signals are routed.

An actual layout view after place-and-route is shown in Figure 4.8. For visibility, everything on top of metal 3 is set to be invisible here. In this cell row, the cell on

the left has its  $PC$  pin connected to the bottom  $\overline{PC}$  metal 3 strip, and the cell on the right connected to the top  $PC$  strip. The metals used to route  $PC$  and  $\overline{PC}$  signals are  $2\times$  wider than the minimum width to reduce the  $PC$  distribution resistance and improve  $Q$ .

## 4.4 Comparison of the Two Design Methodologies

To enable large-scale designs with charge-recovery boost logic, the proposed semi-automated design flow draws several elements from a standard-cell design flow. However, it still differs from a standard-cell design methodology in several aspects.

### Logic Synthesis

In a standard-cell flow, when performing logic synthesis, a tool is typically used to convert RTL code into a gate-level netlist using an existing standard-cell library. A technology library and design constraints are provided to the tool to synthesize the design.

In our semi-automated flow, the cell library is generated after a synthesis tool has been used to partition the design into pipeline stages. When partitioning the design, the number of buffers inserted for phase alignment and load balancing of the two clock phases must be taken into consideration. Each stage of the synthesized netlist is then converted and optimized into boost logic cells manually, creating a library of boost logic cells.

In a standard-cell flow, a design can be synthesized for a different target frequency by updating design constraints and rerunning synthesis, without changing the cell library. In our semi-automated flow, each library gate is designed and characterized to drive a certain amount of loading with a target frequency. Therefore, for our semi-automated flow, each boost logic gate in the library must be redesigned and characterized for the new target operating frequency.

## Place-and-Route

When performing place-and-route, a standard-cell flow performs a clock tree synthesis step after standard cells are placed. Our semi-automated flow generates a clock mesh with  $PC$  and  $\overline{PC}$  strips created over/under the cells along each cell row right after floorplanning, and  $PC$  pins of each boost logic cell are routed to either the top or bottom  $PC$  and  $\overline{PC}$  strips after placement.

For a standard-cell flow, a technology library and timing constraints are provided to an EDA tool to place-and-route the design and close timing. For our semi-automated flow, each cell is designed with a maximum output loading for operating at a target frequency. This target output loading of each gate, along with the gate capacitance of each input, are provided to the tool so that it can choose cells with the right drive strengths and meet the timing constraints.

## 4.5 Summary

This chapter discusses the standard-cell-like semi-automated design flow that we have developed. Previous charge-recovery test-chips have been limited to small and relatively simple designs, as they are full-custom designs. Our semi-automated design flow is very similar to a standard-cell design methodology. A library of boost logic cells are designed and characterized to be used with commercial back-end EDA tools. For our silicon prototypes, the library has 52 different functions, and each function has 2 to 6 different drive strengths. Incorporating custom-designed dynamic cells with a two-phase power-clock distribution, our semi-automated design flow has enabled the design of a large-scale VLSI design with 57,000 gates, as described in Chapter III, exceeding the largest previously-reported charge-recovery test-chip by more than 32 times.

## CHAPTER V

# Evaluation of 926MHz Charge-Recovery LDPC Decoder Test-Chip

### 5.1 Introduction

To demonstrate our semi-automated flow for the design of large-scale charge-recovery chips, we have designed a 576-bit rate-5/6 LDPC decoder using boost logic, as an example prototype. In this chapter, we present experimental results from the evaluation of the charge-recovery LDPC decoder.

The decoder test-chip has been fabricated in a 65nm CMOS process. When operating at 926 MHz, it consumes 564.6 mW, achieving a 6.4 pJ/b/iteration energy consumption and 5.8 Gbps/mm<sup>2</sup> area efficiency, improving on results in [18–21] by 1.3 times in energy consumption with comparable or better area efficiency, even without technology scaling.

The remainder of this chapter is organized as follows: Measurement results from our decoder test-chip are shown and discussed in Section 5.2. A summary of our charge-recovery LDPC decoder is given in Section 5.3.

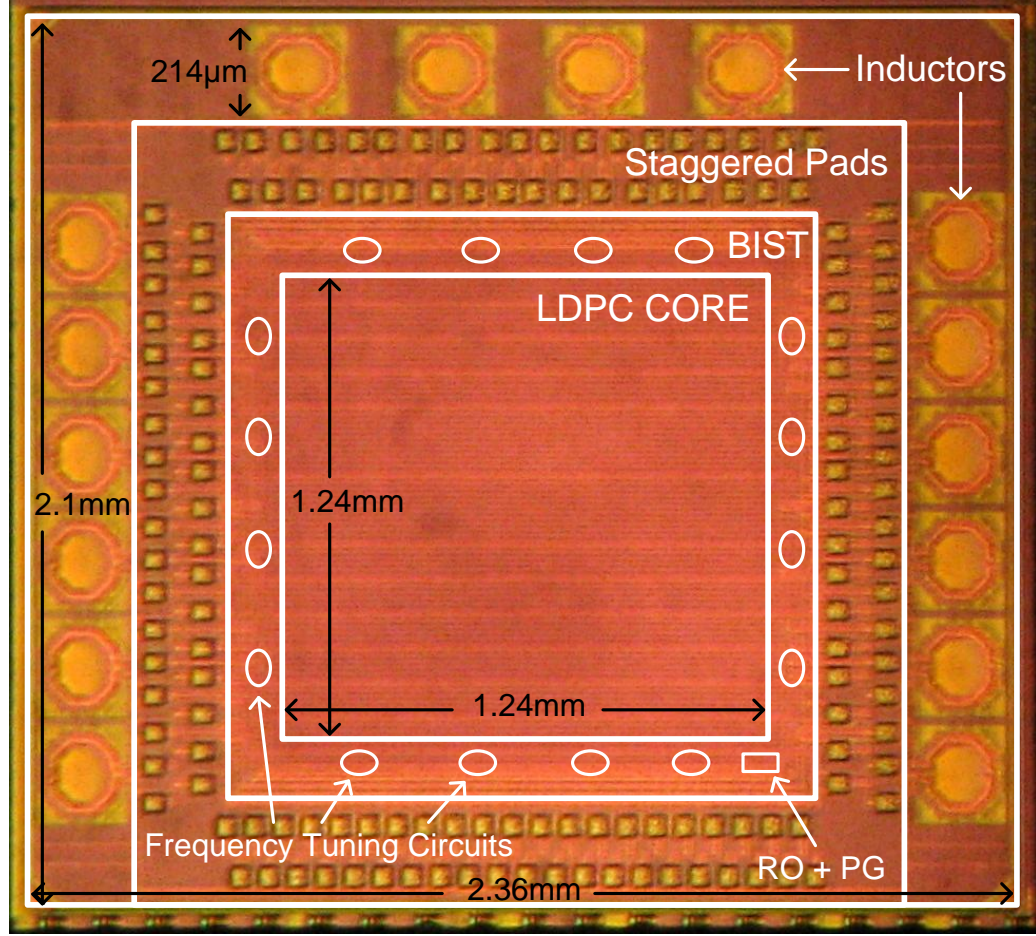


Figure 5.1: Microphotograph of the charge-recovery LDPC test-chip in 65nm CMOS.

## 5.2 Measurement Results

The decoder chip was fabricated in a commercial 65nm CMOS process. Figure 5.1 shows the chip microphotograph. The LDPC decoder logic occupies  $1.54 \text{ mm}^2$ , and it is surrounded by built-in-self-test (BIST) circuit that is used to generate and process the input and output of the decoder. 16 on-chip inductors have been placed outside the staggered pads. Since each inductor has a center tap pin connected to the supply  $V_{DC}$  from the pads, if the pads were placed outside the inductors, loops would be created around inductors, resulting in eddy currents and degrading energy efficiency.

The decoder supports two test modes: a pre-stored mode and a manual mode. In the pre-stored mode, four sets of input vectors with different signal-to-noise ratios

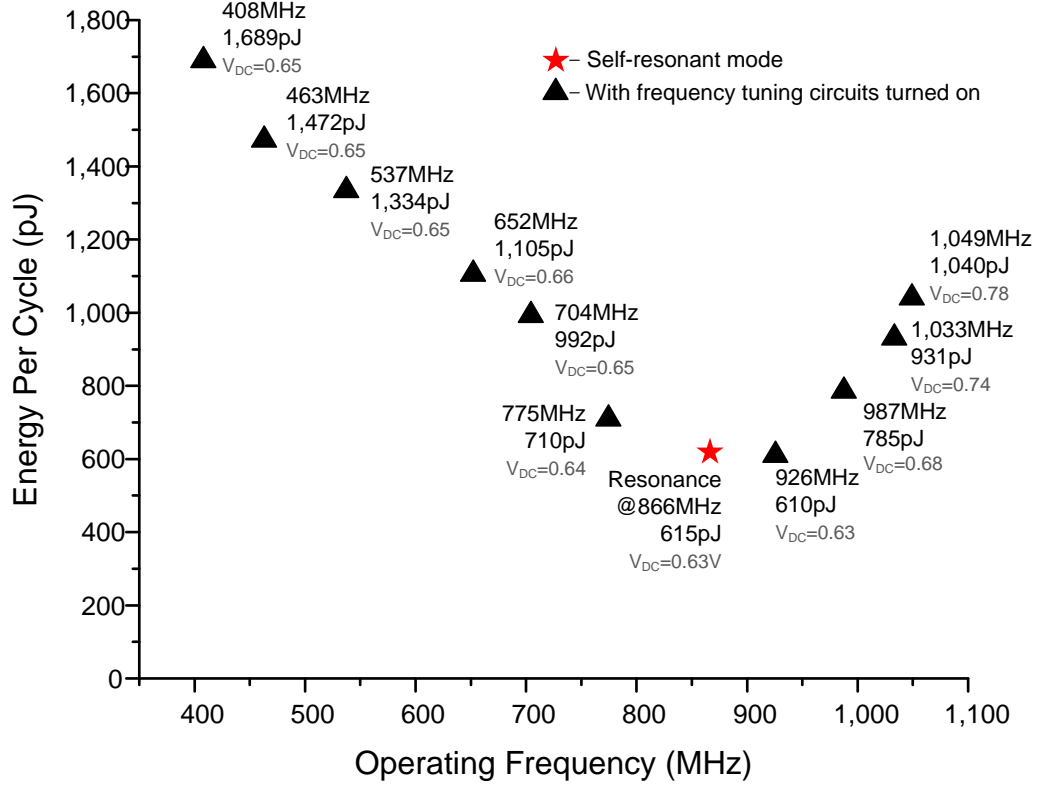


Figure 5.2: Measured energy per cycle versus operating frequency.

(SNR) are pre-stored, along with the corresponding output vectors. When running in this mode, BIST selects from one of the four sets of pre-stored inputs to feed to the decoder. The decoder runs with the selected inputs for 10 iterations, and BIST then verifies the outputs of the decoder with the corresponding set of pre-stored golden outputs. Note that after the decoder finishes the 10 decoding iterations, BIST resets the decoder to start the decoding of the same set of inputs again in order to obtain an accurate power measurement. This mode is used to verify the functionality when sweeping different supply voltages and power-clock frequencies.

In the manual mode, a scan-chain has been implemented. When operating in this mode, we can scan-in any input vector with desired SNR, the number of iterations to run, and other testing parameters to the decoder core. This mode is mainly used for debugging and testing input vectors with SNR other than the pre-stored ones.

Figure 5.2 shows energy per cycle at each operating frequency in measurement.

The supply to the power-clock generator,  $V_{DC}$ , and the supply to the logic stage,  $V_{CC}$ , were tuned to get minimum total energy at each frequency. Minimum energy consumption of our LDPC decoder chip is 610 pJ per cycle when the frequency circuits are turned on and operated at 926 MHz. Without turning on the frequency tuning circuits, the minimum energy of the chip in self-resonant mode is 615 pJ at 866 MHz. Correct function has been validated for clock frequencies ranging from 408 MHz to 1.05 GHz, over plus and minus 25% off resonance. Note that unlike [13], the minimum energy consumption of the decoder does not occur when the chip runs in self-resonant mode without turning on any frequency circuits, indicating that there are insufficient amount of negative transconductance devices. As a result, when the frequency tuning circuits are turned on, they provide additional negative transconductance, helping to replenish the energy lost from the resistance of the distribution network. The voltage swing of the power-clok is therefore increased, allowing the decoder to be operated with lower  $V_{DC}$  and  $V_{CC}$ .

Figure 5.3 shows measured resonant frequency distribution from 16 test-chips when running free with  $V_{DC}=0.64\text{V}$  and  $V_{CC}=0.36\text{V}$ . Correct function has been validated for all 16 chips, with average resonant frequency 809.6 MHz and standard deviation 5.8 MHz. The resonant frequency of these chips varies by 1.3%, well within the 25% tuning range of the clock generator circuit.

Figure 5.4 shows measured energy per cycle for input SNR ranging from 0.5 dB to 4.5 dB. Input vectors are generated by Matlab with desired SNR, and fed to the decoder core using the manual testing mode. For conventional static CMOS designs, the input SNR affects the switching activities of LDPC decoders, thus affecting energy consumption. However, for charge-recovery designs, due to the continuous switching nature of dynamic logic, the energy consumption of the charge-recovery decoder stays flat for different input SNR.

Table 5.1 summarizes the performance characteristics of our test-chip, and com-



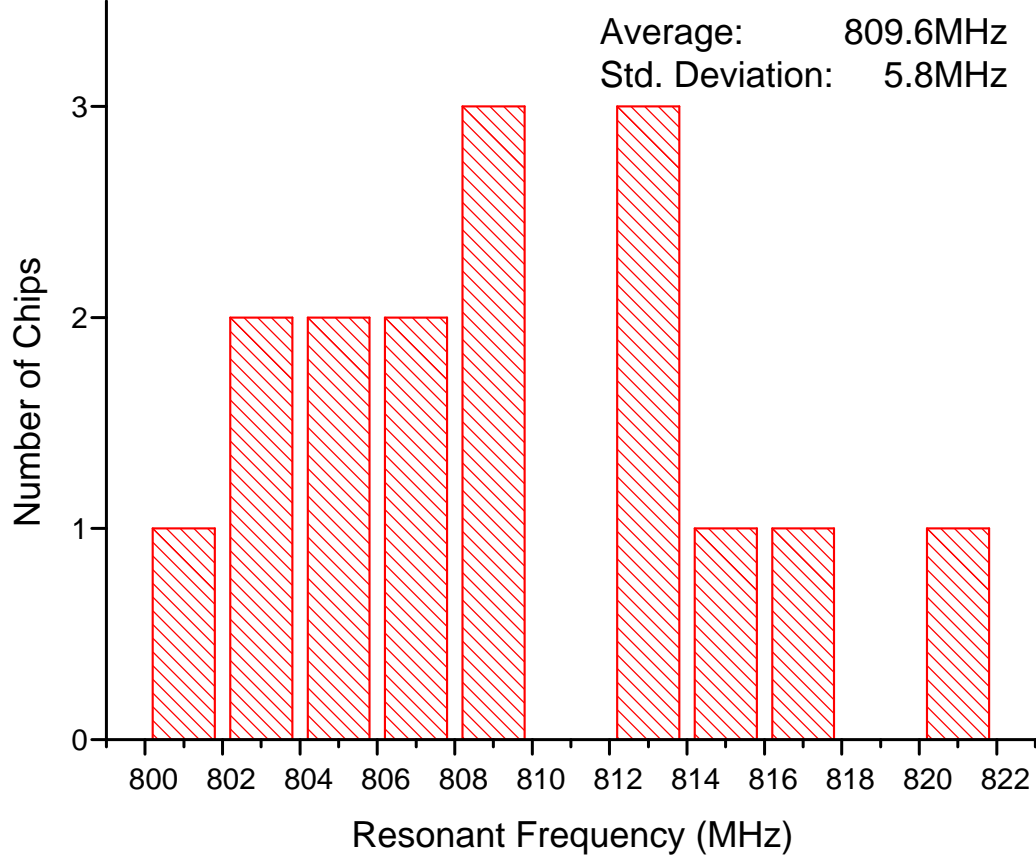


Figure 5.3: Measured resonant frequency distribution at  $V_{DC}=0.64V$  and  $V_{CC}=0.36V$  of 16 chips.

compares the performance with recent published LDPC decoders [18–21]. The 576-bit charge-recovery LDPC decoder has 1.54 mm<sup>2</sup> core area. The minimum energy consumption of the chip is 609.8 pJ when the power-clock operates at 925.9 MHz, achieving 8.9 Gbps throughput for 10 decoding iterations. The key metrics for comparing the performance of LDPC decoders are shown in the last two rows in Table 5.1, the energy consumption (measured in pJ per bit per iteration) and the area efficiency (measured in throughput per unit area). The charge-recovery LDPC decoder chip outperforms state-of-the-art designs with comparable code length and throughput. Even without technology scaling, the test-chip achieves at least 1.3 times lower energy consumption with comparable or better area efficiency.

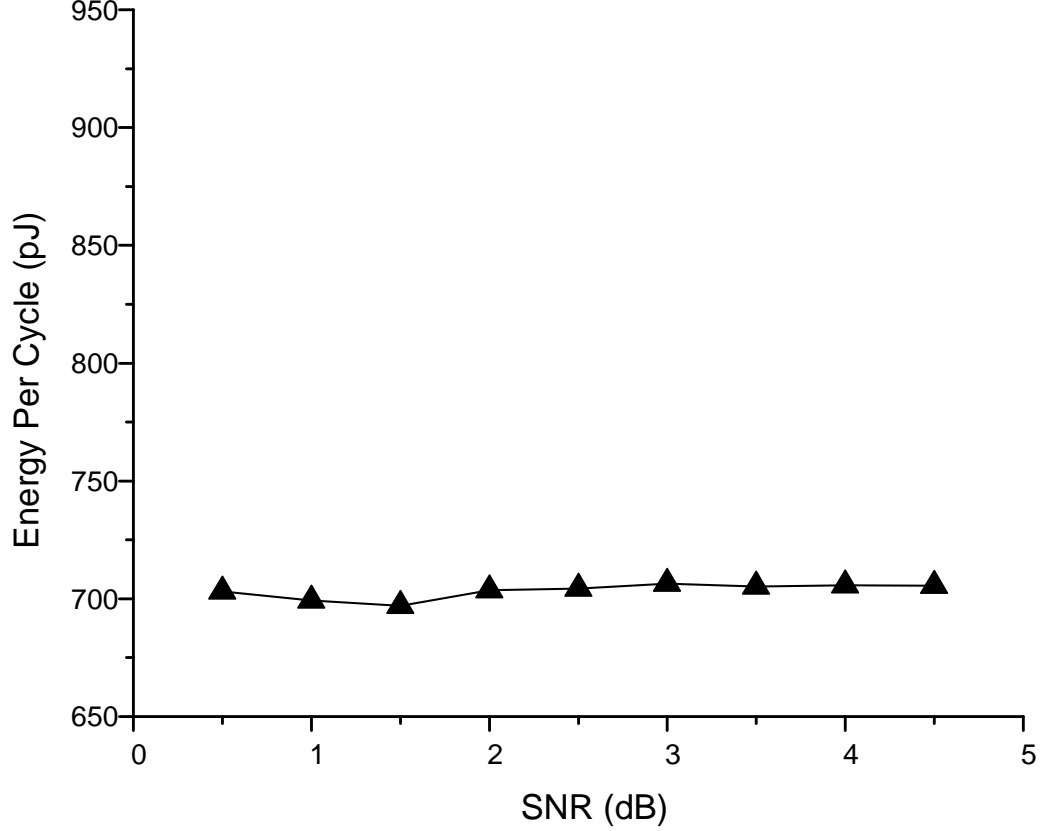


Figure 5.4: Measured energy per cycle versus input SNR.

### 5.3 Summary

To demonstrate our semi-automated flow for the design of large-scale charge-recovery chips, we have designed a 576-bit rate-5/6 LDPC decoder using boost logic, as an example prototype. The chip has been fabricated in a 65nm bulk silicon process. Correct functionality has been validated for clock frequencies ranging from 408 MHz to 1.05 GHz. When operating at 926 MHz with frequency tuning circuits turned on, the test-chip consumes 610 pJ per cycle with 8.9 Gbps throughput, achieving 6.4 pJ/bit/iteration energy consumption and 5.78 Gbps/mm<sup>2</sup>. The decoder outperforms state-of-the-art commercial-strength LDPC decoders by at least 1.3 times improvement in energy consumption with comparable or better area efficiency, even without technology scaling.

	This Work	ASSCC '11 [18]	JSSC '12 [19]	JSSC '14 [20]	ISSCC '14 [21]
Technology	65nm	65nm	65nm	65nm	28nm FDSOI
Code Length	576	576 - 2,304	672	672	672
Core Area (mm <sup>2</sup> )	1.54 (2.34 w/ ind)	3.36	1.56	1.6	0.63
Frequency (MHz)	926	110	197	540	260
Iteration	10	10	5	10	3.75
Throughput (Gbps)	8.89	1.06	5.79	9.00	12.00
Input SNR (dB)	1.0	N/A	5.5	5.0	5.0
Power (mW)	564.6	115	361	782.9	180
Energy Consumption (pJ/bit/iteration)	6.35	21.80	12.48	8.95	8.00
Area Efficiency (Gbps/mm <sup>2</sup> )	5.78 (3.80 w/ ind)	0.31	2.12 <sup>1</sup>	5.63	7.14 <sup>1</sup>

<sup>1</sup> Normalized to 10 iterations

Table 5.1: Chip summary and comparison with state-of-the-art designs.

This charge-recovery LDPC decoder chip demonstrates the potential of charge-recovery logic for energy- and area-efficient high-performance design, as well as an accompanying design methodology that leverages automated EDA tools and is applicable to large-scale DSP applications.

## CHAPTER VI

### Design and Architecture of Charge-Recovery

#### LDPC with In-Package Inductors

Charge-recovery circuits rely on inductive elements to recover charge from gate fanouts and achieve energy-efficient operation. On-chip inductors are used to achieve GHz-speed high-performance charge-recovery chips, at the cost of silicon area overheads and inductors with low  $Q$  factor due to resistive on-chip metals. For charge-recovery designs operating at lower frequencies (hundreds of MHz or lower), high  $Q$  discrete inductors are used, resulting in extra board area and extra costs. This work is the first one to explore advantages from both the high performance and the discrete ends.

To enable area-efficient high-speed charge-recovery designs, we present a new architecture that uses in-package inductors, eliminating the area overheads caused by

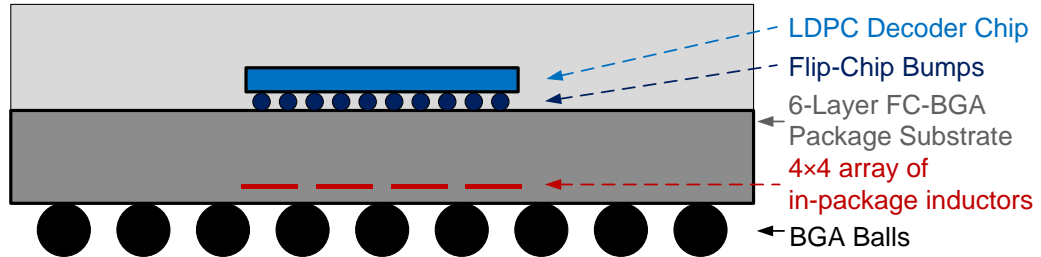


Figure 6.1: Charge-recovery LDPC decoder chip in 65nm flip-chip technology and custom-designed FC-BGA package substrate with 16 in-package inductors.

on-chip inductors. As a proof-of-concept, we have designed a 576-bit charge-recovery low-density parity-check (LDPC) decoder with in-package inductors. Figure 6.1 shows this decoder, including a custom-designed 6-layer flip-chip–ball-grid-array (FC-BGA) package substrate and a test-chip fabricated in a 65nm CMOS flip-chip process. 16 inductors are designed on the fifth layer of the 6-layer FC-BGA package substrate so that the need of on-chip inductors is eliminated, and inductor  $Q$  factors are increased, improving area efficiency and energy efficiency compared to alternative implementations with on-chip inductors.

The remainder of this chapter is organized as follows: Section 6.1 provides an overview of the charge-recovery LDPC decoder with in-package inductors. Section 6.2 presents the power-clock generation and the distribution network. Section 6.3 describes the design of the package substrate. Section 6.4 explains the architecture of the LDPC decoder. A summary is provided in Section 6.5.

Part of the work covered in this chapter will be appearing in ASSCC 2015 [23].

## 6.1 Introduction

A 576-bit, rate-5/6 low-density parity-check (LDPC) decoder has been designed using charge-recovery circuitry with in-package inductors. Charge-recovery circuits [11, 13, 49] rely on inductive elements to recover charge from gate fanouts and achieve energy-efficient operation. Previous GHz-speed high-performance charge-recovery chips have relied on integrated on-chip inductors at the cost of silicon area overheads. In addition to area overheads, on-chip inductors also suffer from relatively low quality factors ( $Q$ ), typically less than 10 [22, 50], due to resistive on-chip metals. For charge-recovery designs operating at lower frequencies (hundreds of MHz or lower), high  $Q$  discrete inductors have been used, incurring extra board area and extra costs. This thesis presents the first-ever charge-recovery test-chip with in-package inductors offering high  $Q$  inductors without area overheads.

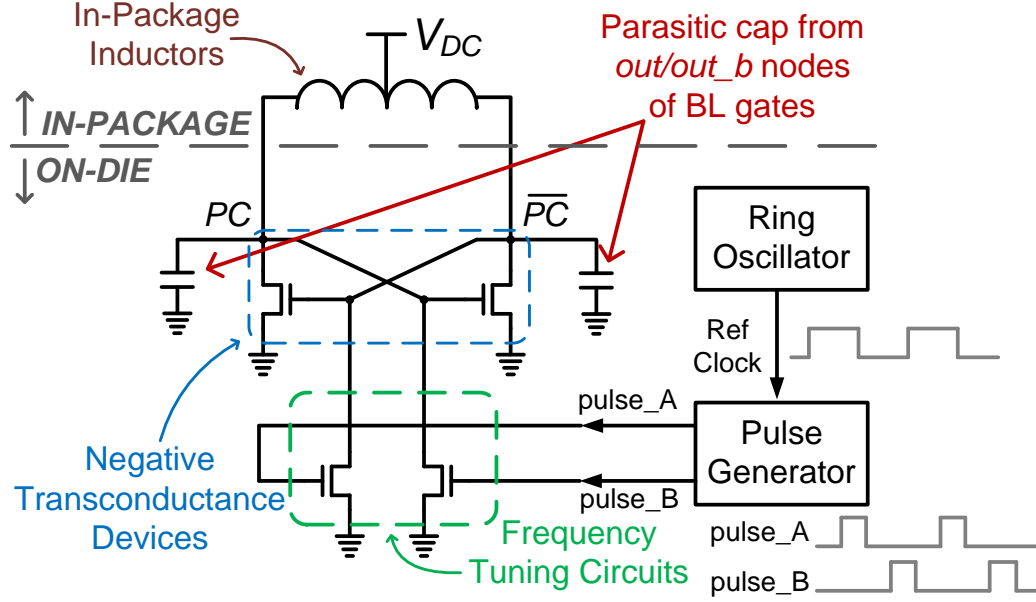


Figure 6.2: Blip power-clock generator, formed by in-package inductors, cross-coupled internal drivers, and frequency tuning circuits with pulses generated by a ring oscillator and a pulse generator.

The decoder described in this chapter is shown in Figure 6.1. The LDPC test-chip is fabricated in a 65nm flip-chip process. The inductors required by charge-recovery logic are embedded in a custom-designed 6-layer flip-chip-ball-grid-array (FC-BGA) package substrate, achieving better  $Q$  than their on-chip counterparts thanks to the availability of thicker and therefore less resistive copper in the package substrate. Inductors in the package are connected to the charge-recovery network on the die through 48 flip-chip bumps, eliminating bonding wires and their parasitics to enable efficient charge recovery. Correct operation has been verified from 624MHz to 1.08GHz. When operating at 934MHz, the chip consumes 286mW, yielding energy consumption of 3.19pJ/b/iteration and an area efficiency of 5.83Gbps/mm<sup>2</sup>, improving on results in [19–22] by at least 2.3× in energy consumption with similar or better area efficiency.

## 6.2 Power-Clock Generation and Distribution

The two-phase power-clock required by boost logic gates is generated using a distributed version of the blip power-clock generator [44], shown in Figure 6.2. The power-clock generator consists of on-die and in-package components. 144 distributed on-die negative transconductance devices, 16 on-die frequency tuning circuits, an on-die ring oscillator (RO), and an on-die pulse generator (PG), along with 16 inductors in the package are used to generate the power-clock by resonating the parasitic capacitance of the power-clock distribution network and the boost logic gate fanouts. To enable frequency tuning, the RO generates a reference clock with a desired frequency feeding to the PG, and the PG then outputs a pair of 180-degree out-of-phase pulses with programmable duty cycle, achieving frequency scaling and forcing the power-clock to run at the desired frequency.

Figure 6.3 shows the on-chip power-clock distribution network. 48 flip-chip bumps (24 for each clock phase) are used to connect the two-phase power-clock,  $PC$  and  $\overline{PC}$ , from the 16 inductors in the package substrate to the on-chip power-clock distribution network. An additional 48 bumps are for supplies,  $V_{SS}$ , and signals for testing the decoder chip.  $PC$  and  $\overline{PC}$  bumps are placed right on top of the top-level metal of the clock meshes, enabling efficient recovery. Clock meshes are employed to distribute the power-clock using top-level metals to minimize clock skew, allowing custom-designed boost logic dynamic cells to be easily connected to the two-phase power-clock using commercial EDA tools. To distribute the power-clock from the mesh to the  $PC$  pin of each boost logic gate, each standard-cell row has two metal-3 strips reserved for delivering  $PC$  and  $\overline{PC}$  to the boost logic gates. These strips are tied to top-level clock meshes, allowing the routing of the power-clock network using an automatic place-and-route tool and avoiding large clock skew. 144 negative transconductance devices are distributed across the core to maintain the oscillation. To operate the design off-resonance, a pair of 180-degree out-of-phase pulses at the target frequency

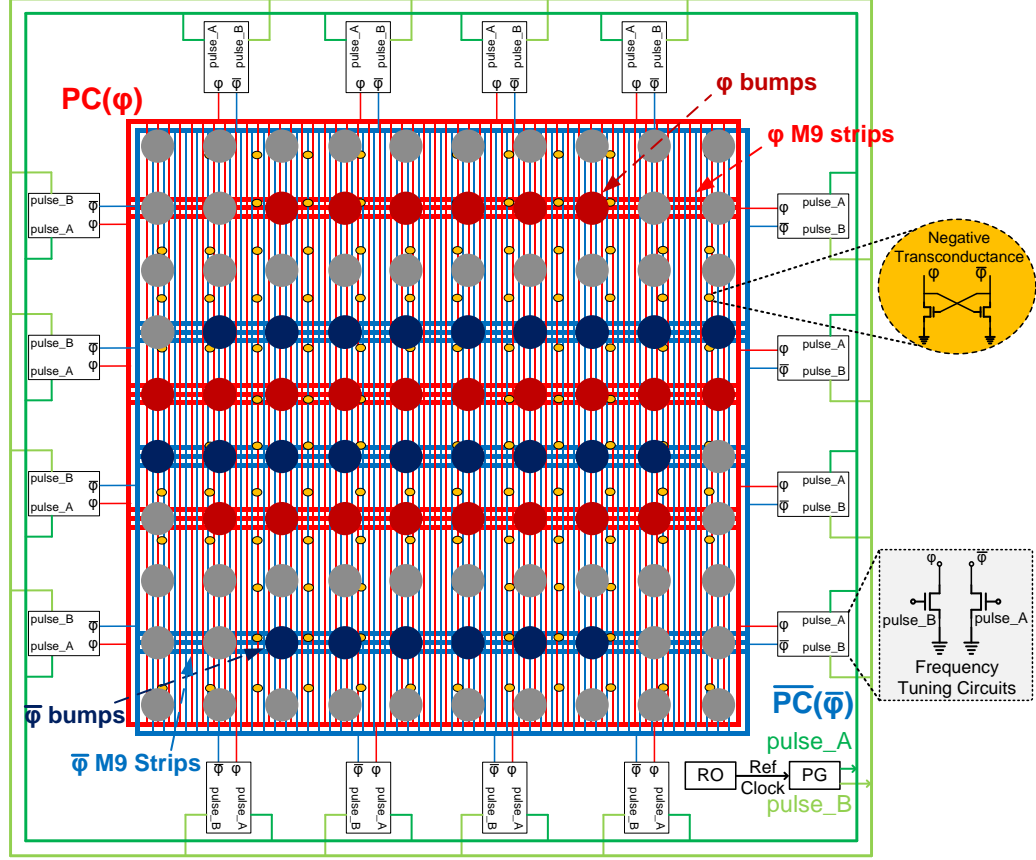


Figure 6.3: Power-clock generation and distribution: clock mesh network for  $PC$  and  $\overline{PC}$ , 100 flip-chip bumps, 16 frequency tuning circuits, and 144 negative transconductance devices.

is distributed to the 16 frequency tuning circuits around the core using a tree structure with supply and ground shielding.

### 6.3 Package Substrate Design

The FC-BGA substrate is manufactured through a 6-layer build-up 2-2-2 manufacturing process. Figure 6.4 shows the cross-section view of the package substrate. The build-up process requires a core layer that is  $1.5\times$  thicker than other layers. The thickness of the copper is  $15\mu\text{m}$ , which helps significantly with the  $Q$  of the in-package inductors compared to their on-chip counterparts ( $3.4\mu\text{m}$  for ultra-thick metal).

Figure 6.5 shows the layer-by-layer view of our package substrate design. The



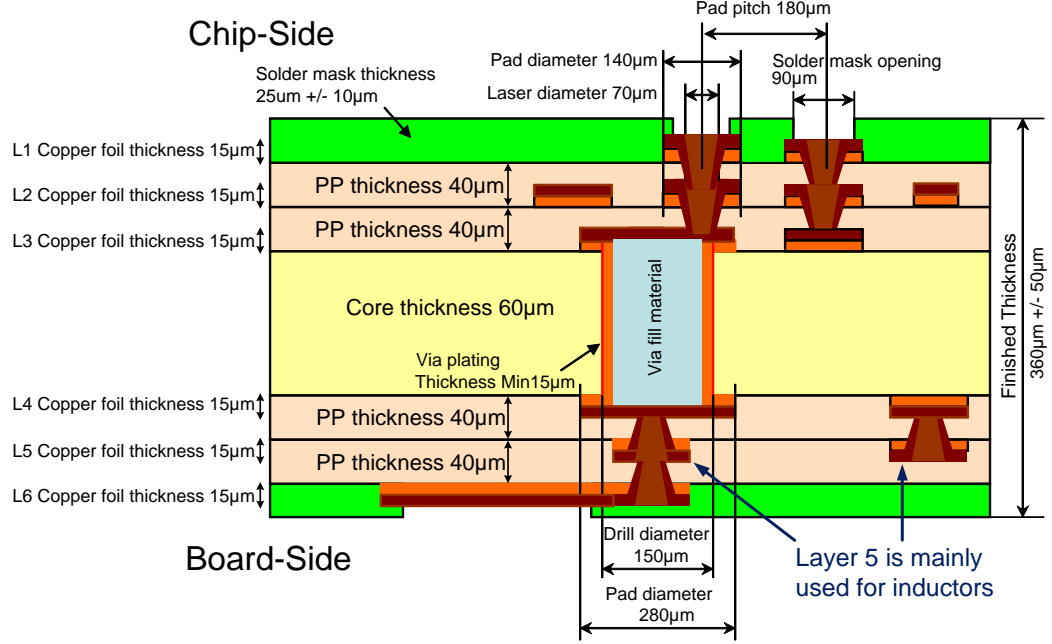


Figure 6.4: Cross-section view of the package substrate.

substrate occupies  $8\text{mm} \times 8\text{mm}$ , and the die occupies  $1.86\text{mm} \times 1.86\text{mm}$  with a  $10 \times 10$  array of flip-chip bumps. Layer 1 of the substrate is used for flip-chip bump connections. Layers 2 to 4 are mainly for routing. The diameter of vias connecting Layers 3 and 4 is larger because of this thicker core layer. 16 horizontal in-package center-tap coil inductors are designed on Layer 5. Layer 6 is reserved for BGA ball connections. Note that the size of the substrate is dominated by the number of BGA balls (for supplies,  $V_{SS}$ , and I/Os) required for the decoder, not by the inductors. The amount of area taken by the inductors on Layer 5 is comparable with the area of the decoder chip.

Inductor design plays a key role in the efficiency of charge-recovery chips. The 16 in-package coil inductors have been carefully designed and characterized using a commercial 3D full-wave electromagnetic field solver tool. Figure 6.6 shows the dimensions and the specifications of one of the 16 inductors. When simulated at 1GHz, each inductor has 969.4pH inductance with a  $Q$  factor of 33.4, achieving  $3.6 \times$  improvement in inductor  $Q$  compared to Chapter III with similar operating conditions [22].

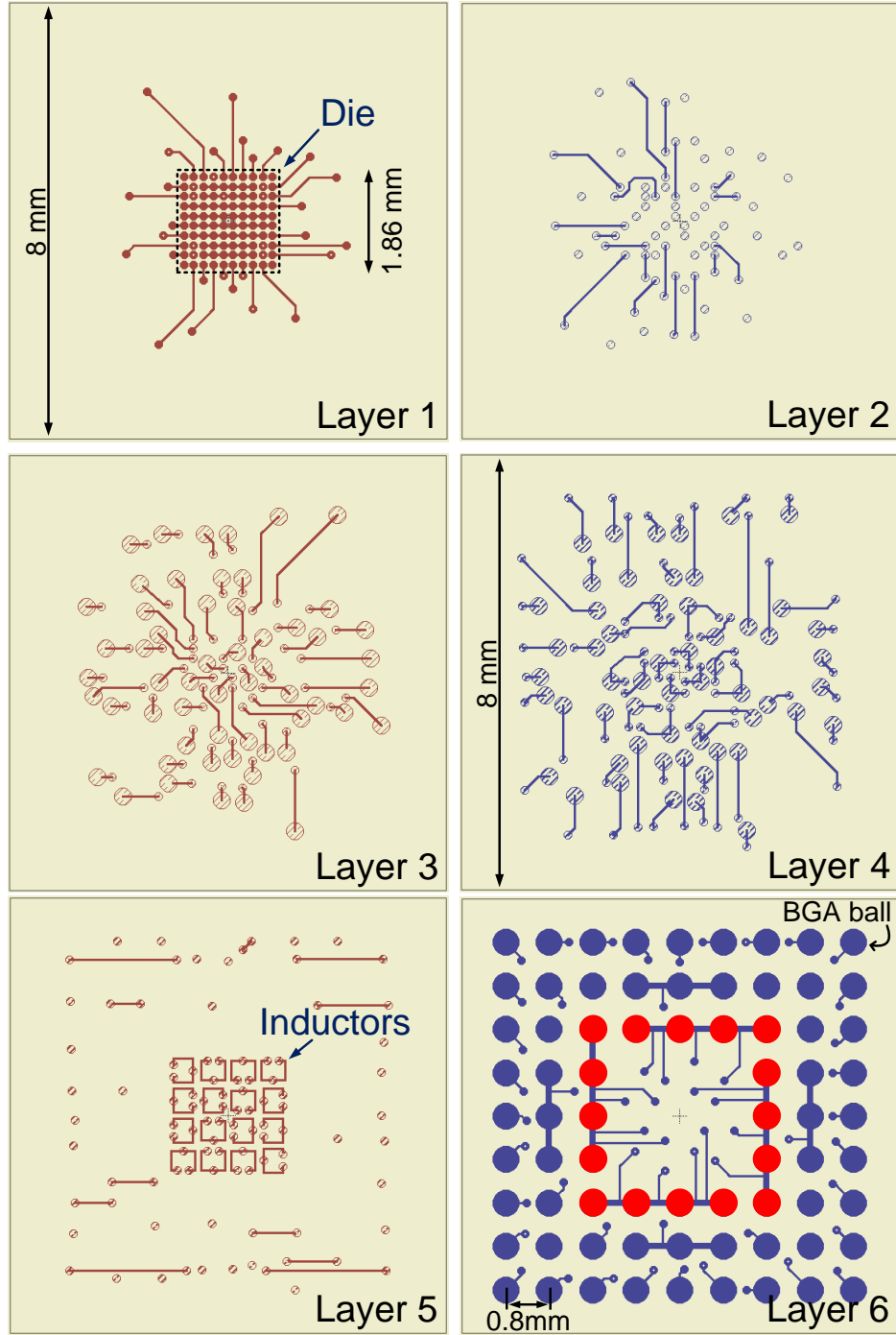


Figure 6.5: Custom-designed 6-layer FC-BGA package substrate.

Designing the coil inductors in vertical orientation has also been considered, but process variation and misalignments between layers during the build-up manufacturing process make it difficult to design coils with a precise inductance value and high  $Q$ .



## 6.4 LDPC Decoder Design

The 576-bit, rate-5/6 charge-recovery LDPC decoder adopts the block-parallel architecture described in Chapter III [22]. Figure 3.9 shows the decoder architecture for the LDPC code specified by the IEEE 802.16e standard. The code matrix, as shown in Figure 3.8, is partitioned into 4 blocks so that complex and long global interconnects are replaced with relay local interconnects between neighboring blocks. Two columns in the code matrix are swapped for regular partitioning. The min-sum decoding consists of check node operations and variable node operations. Starting from the check node operation on the first row of Block 1, the decoder then relays the results to Blocks 2, 3, and 4 in order. Following the check node operation on the first row of Block 1, the decoder performs the variable node operation on it while Block 1 begins the check node operation on the second row in parallel. For complete check node and variable node operations in all 4 blocks, one decoding iteration takes 24 cycles (48 phases). Due to this deeply-pipelined relay architecture, the decoder is able to process 4 streams in parallel without any pipeline stalls.

## 6.5 Summary

A new architecture that uses in-package inductors is presented to eliminate the area overheads caused by on-chip inductors, enabling area-efficient high-speed charge-recovery designs. A 576-bit charge-recovery LDPC decoder with in-package inductors has been designed and evaluated as an example prototype. This chapter shows the design and the architecture of the charge-recovery LDPC decoder with in-package inductors. This decoder includes a test-chip fabricated in 65 nm CMOS flip-chip technology and a custom-designed 6-layer FC-BGA package substrate. 16 inductors are embedded on the fifth layer of the 6-layer FC-BGA package substrate, improving area efficiency and energy efficiency by eliminating the use of on-chip inductors, the

improved  $Q$  of the in-package inductors, and the lower resistance of the flip-chip connections (compared to bondwires).

## CHAPTER VII

# Evaluation of 934MHz Charge-Recovery LDPC Test-Chip with In-Package Inductors

### 7.1 Introduction

To demonstrate the architecture proposed in Chapter VI which uses in-package inductors to enable area-efficient high-speed charge-recovery designs by eliminating on-chip inductors, a 576-bit charge-recovery LDPC decoder with in-package inductors has been designed as a proof-of-concept. In this chapter, we present experimental results from the evaluation of the charge-recovery LDPC decoder with in-package inductors.

The decoder includes a test-chip fabricated in a 65nm CMOS flip-chip process and a custom-designed 6-layer FC-BGA package substrate. When operating at 934MHz, the decoder reaches a 9Gb/s throughput, consuming 286mW, or 3.2pJ/b/iteration, improving on the state-of-the-art published results by at least 2.3 times in energy consumption with similar or even better area efficiency.

The remainder of this chapter is organized as follows: Section 7.2 describes measurement results from our decoder and evaluates the results. A summary of our charge-recovery LDPC decoder with in-package inductors is given in Section 7.3.

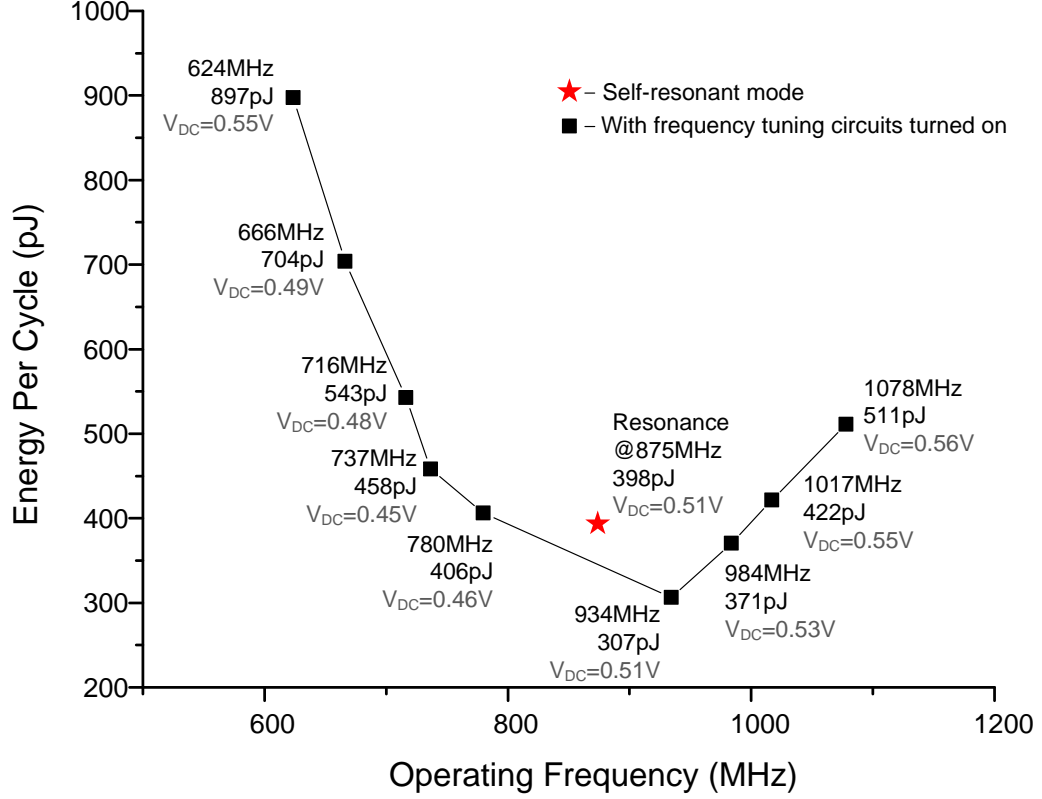


Figure 7.1: Measured energy per cycle versus operating frequency.

## 7.2 Measurement Results

The chip has been fabricated in a 65nm CMOS flip-chip process. The charge-recovery LDPC decoder logic occupies 1.54mm<sup>2</sup>. Correct function has been validated for clock frequencies ranging from 624MHz to 1.08GHz. Figure 7.1 shows measured energy per cycle versus operating frequency. Minimum energy consumption is 306.5pJ per cycle when the power-clock is operating at a frequency of 934MHz, dissipating 286.4mW of power at room temperature. With frequency tuning circuits turned off, the minimum energy consumption of the decoder in self-resonant mode is 398pJ per cycle at 875MHz. Note that, unlike [22], the minimum energy consumption operating point does not occur when the decoder chip is operating in self-resonant mode. We surmise that this might be the result of insufficiently many negative transconductance devices, resulting in reduced power-clock voltage swing and thus requiring higher  $V_{DC}$

	This Work	JSSC '12 [19]	JSSC '14 [20]	ISSCC '14 [22]	ISSCC '14 [21]
Technology	65nm	65nm	65nm	65nm	28nm FDSOI
Code Length	576	672	672	576	672
Code Rate	5/6	7/8	1/2	5/6	1/2
Core Area (mm <sup>2</sup> )	1.54	1.56	1.6	1.54 (2.34 w/ ind)	0.63
Frequency (MHz)	934	197	540	821	260
Iteration	10	5	10	10	3.75
Throughput (Gbps)	8.97	5.79	9.00	7.88	12.00
Input SNR (dB)	1.0	5.5	5.0	1.0	5.0
Power (mW)	286.4	361	782.9	576.8	180
Energy Consumption (pJ/bit/iteration)	3.19	12.48	8.95	7.32	8.00 <sup>1</sup>
Area Efficiency (Gbps/mm <sup>2</sup> )	5.83	2.12 <sup>2</sup>	5.63	5.78 (3.80 w/ ind)	7.14 <sup>2,3</sup>

<sup>1</sup> Energy consumption for 28nm (not normalized to 65nm)

<sup>2</sup> Normalized to 10 iterations

<sup>3</sup> Area efficiency for 28nm (not normalized to 65nm)

Table 7.1: Chip summary and comparison with state-of-the-art designs.

and  $V_{CC}$  values. When the frequency tuning circuits are turned on to run the decoder off-resonance, they replenish the energy lost due to the resistance of the distribution network to maintain the oscillation, acting as extra negative transconductance devices and enabling the scaling of  $V_{DC}$  and  $V_{CC}$  to lower values. The energy efficiency of this chip is higher than that of the chip reported in Chapters III and V due to the improved  $Q$  of the in-package inductors, which increases the efficiency of charge recovery, and the lower resistance of the flip-chip connections (compared to bondwires), which allows for further supply voltage scaling and thus lower energy consumption.

Table 7.1 gives the performance characteristics of the chip in this work and com-



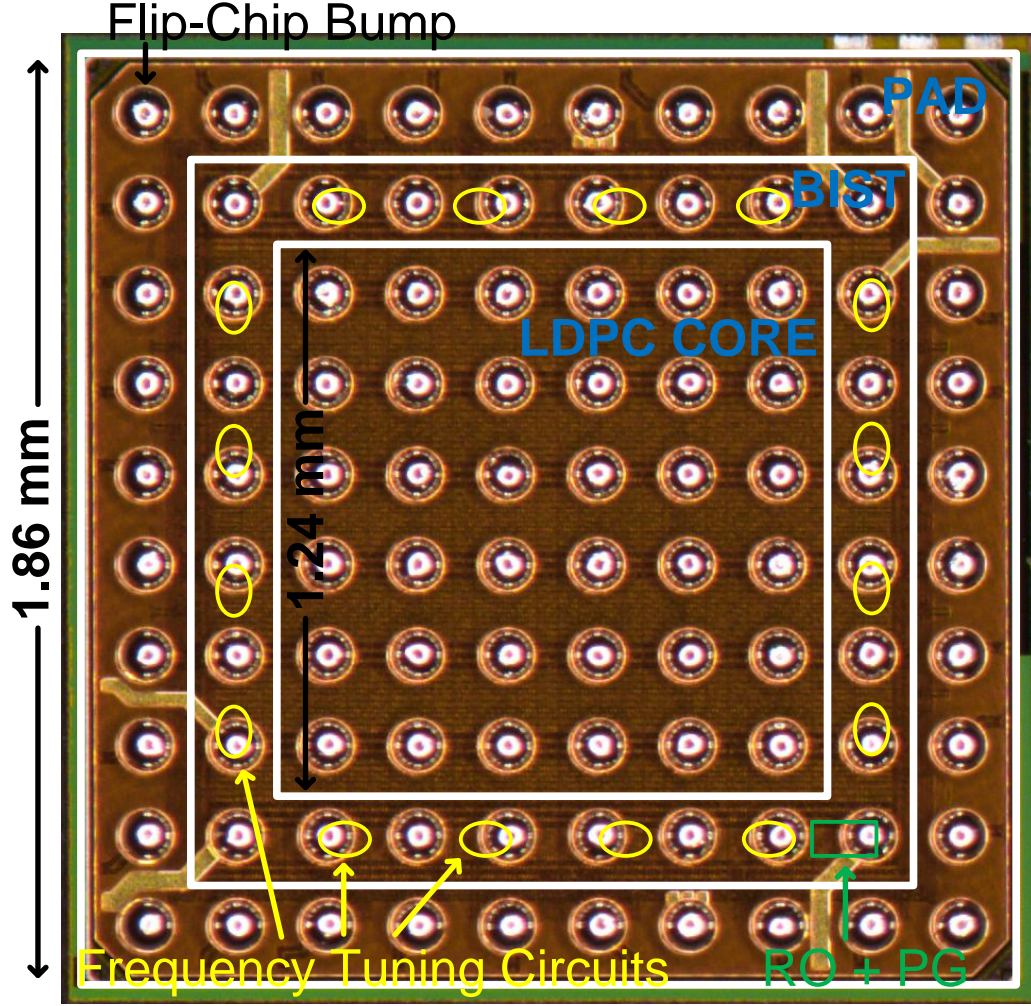


Figure 7.2: Microphotograph of the charge-recovery LDPC test chip in 65nm CMOS.

compares it with the most recently reported high-throughput LDPC decoders. The charge-recovery LDPC decoder chip in Chapters VI and VII outperforms state-of-the-art designs of comparable code length, complexity, and throughput [19–22], achieving at least 2.3 times lower energy consumption compared to all designs, while having better area efficiency compared to the ones fabricated in the same technology. Even without any normalization for different process nodes, the area efficiency of this chip is still comparable with that of the chip in [21], which was fabricated in a more advanced 28nm technology.

A die microphotograph is shown in Figure 7.2. A built-in-self-test (BIST) circuit

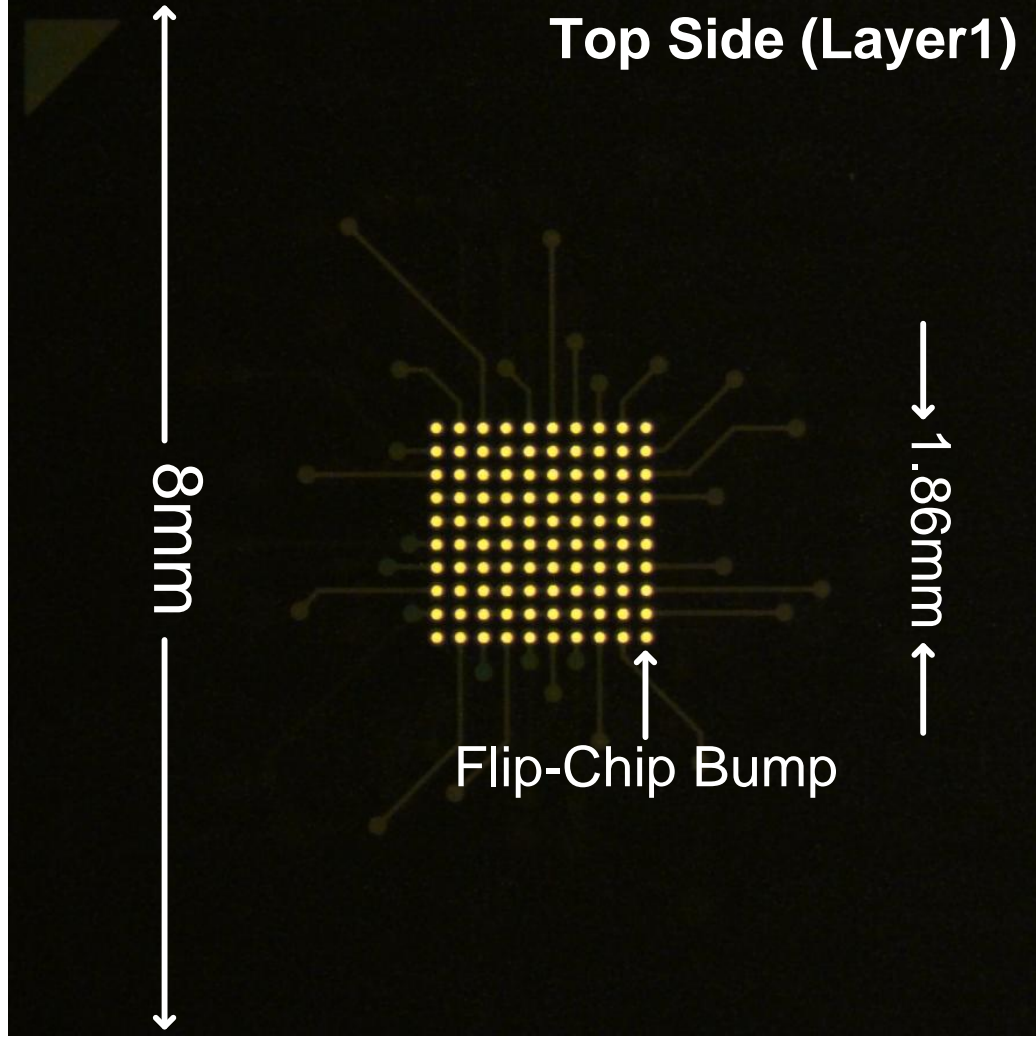


Figure 7.3: Microphotograph of the top side of the custom-designed 6-layer FC-BGA package substrate.

that is used to generate and process the input and output of the decoder, along with RO, PG, and frequency-tuning circuits are implemented with static CMOS logic and are distributed around the decoder core.

Figures 7.3 and 7.4 are microphotographs of the package substrate. Figure 7.3 shows the view from the top of the package substrate. The substrate occupies  $8\text{mm} \times 8\text{mm}$ , and the die occupies  $1.86\text{mm} \times 1.86\text{mm}$  with a  $10 \times 10$  array of flip-chip bump pads. Figure 7.4 provides the view from the bottom of the package substrate, showing 72 BGA ball pads on the 6th layer and also the inductors on the 5th layer.

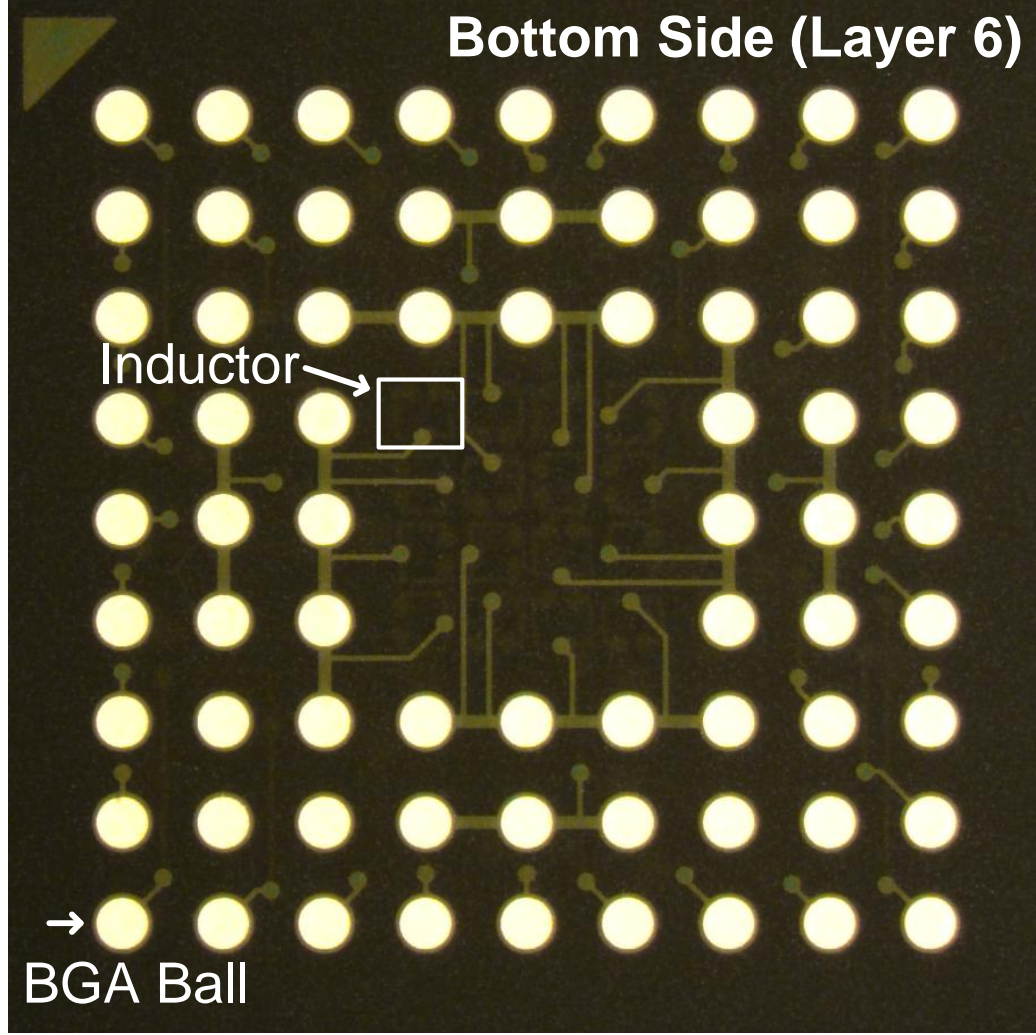


Figure 7.4: Microphotograph of the bottom side of the custom-designed 6-layer FC-BGA package substrate.

### 7.3 Summary

In Chapters VI and VII, we explore the use of in-package inductors to improve upon the quality and eliminate the area overheads of on-chip inductors in high-performance charge-recovery designs. The design of a 576-bit charge-recovery LDPC decoder with in-package inductors is presented, including a custom-designed 6-layer FC-BGA package substrate and a test-chip fabricated in a 65nm CMOS flip-chip process. 16 inductors are designed on the fifth layer of the 6-layer FC-BGA package substrate so that the need for on-chip inductors is eliminated. In-package induc-

tors also provide better quality factors than on-chip ones, improving area efficiency and energy consumption. When operating at 934MHz, the decoder reaches a 9Gb/s throughput, consuming 286mW, or 3.2pJ/b/iteration, improving on the state-of-the-art published results by at least 2.3 times in energy consumption with similar or even better area efficiency.

## CHAPTER VIII

### Conclusions and Future Research Directions

Charge-recovery logic has the potential to achieve order-of-magnitude improvement in energy efficiency while maintaining high performance. However, the lack of large-scale high-speed demonstration in silicon, and the area overheads caused by the need of inductors for charge-recovery logic are two major concerns. In this dissertation, we present ways to address these concerns by designing and evaluating two charge-recovery systems of significant complexity operating at GHz clock speed.

This chapter summarizes the contributions in this dissertation and discusses possible future research directions to take full advantage of the potential offered by charge-recovery logic.

#### 8.1 Charge-Recovery LDPC Decoder with Semi-Automated Design Flow

To explore scalable charge-recovery designs, we present a semi-automated standard-cell-like design flow to enable the design of large-scale charge-recovery chips. To demonstrate the effectiveness of this semi-custom design flow, we have designed and evaluated a 576-bit charge-recovery low-density parity-check (LDPC) decoder as an example prototype. LDPC is a very popular error correcting code in modern communication standards, requiring complex and power-intensive computations.

The LDPC decoder chip has been fabricated in a 65nm CMOS process and relies on 16 integrated inductors to achieve energy-efficient operation by recovering charge from gate fanouts. When self-oscillating at 866 MHz, the chip recovers 51.4% of the energy supplied to it from simulation. Clock meshes are used to distribute the two-phase power-clock with a worst-case skew of 11.3 ps, based on simulations when self-oscillating. In terms of device count, this chip is more than an order of magnitude larger than the largest previously-reported chips with charge-recovery logic, enabled by the semi-automated design methodology we have developed. Functioning correctly at clock speeds ranging from 408 MHz to 1.05 GHz, this prototype is the first-ever demonstration of a GHz-speed charge-recovery chip of significant complexity. When operating at 926 MHz, it achieves a throughput of 8.9 Gbps at 6.4 pJ/bit/iteration, improving on results in previous state-of-the-art commercial-strength LDPCs by at least 1.3 times in energy consumption with comparable or better area efficiency, even without technology scaling.

This charge-recovery LDPC decoder chip demonstrates the potential of charge-recovery logic for energy- and area-efficient high-performance design, as well as an accompanying design methodology that leverages automated EDA tools and is applicable to large-scale DSP applications.

## 8.2 Charge-Recovery LDPC with In-Package Inductors

To further explore the potential of area-efficient charge-recovery design at multi-GHz clock speeds, we present a new architecture that uses in-package inductors, eliminating the area overheads caused by on-chip inductors. As a proof-of-concept, we have designed and evaluated a charge-recovery LDPC decoder with in-package inductors. This decoder includes a custom designed 6-layer FC-BGA package substrate and a test-chip fabricated in a 65nm CMOS flip-chip process. 16 inductors have been embedded on the fifth layer of the 6-layer FC-BGA package substrate, improving



area efficiency and energy efficiency by eliminating the use of on-chip inductors, the improved  $Q$  of the in-package inductors, and the lower resistance of the flip-chip connections (compared to bondwires). From measurement, this chip achieves at least 2.3 times lower energy consumption with similar or even better area efficiency over state-of-the-art published designs of comparable code length, complexity, and throughput.

## 8.3 Future Directions

In Conclusion, we discuss three future possibilities and challenges in the implementation of charge-recovery logic in order to take full advantage of the potential offered by charge-recovery logic.

### 8.3.1 Fully-Automatic Design Methodology

Despite its potential to achieve energy-efficient high-performance operation, charge-recovery logic has not been widely adopted in industry as a mainstream methodology, due to the amount of time and design effort that is required to take full advantage of its potential. In this dissertation, we have developed a standard-cell-like semi-automated design flow that enables us to implement our decoder with orders of magnitude improvement in terms of device count over previous charge-recovery test-chip. However, this flow still has a lot of room for improvement.

In front-end design, when performing logic synthesis, we need a flow to automatically partition and micropipeline a design for charge-recovery logic while limiting the number of phase alignment buffers, balancing the stack height of each gate, and balancing the load of the two clock phases. This requirement is not specific to charge-recovery logic, as any dynamic logic would require this kind of support [47, 48].

In back-end design, there is a need for methods to analyze library gates for their charge-recovery characteristics, such as the shape of the sine-wave-like power-clock and the crossing point of the two-phase power-clock, so that gate models are more

accurate when performing place-and-route.

During place-and-route, there is need for automatic design methodologies that distribute charge-recovery cells as evenly as possible across the entire design to have a balanced loading on the two power-clock phases. In addition, the dual-rail signals should be routed with similar loading to reduce clock jitter.

### **8.3.2 Efficient Charge-Recovery Designs for Wide Range of Operating Frequencies**

Another challenge for charge-recovery logic is their limited range of operating frequencies. Charge-recovery test-chips are typically designed for a target operating frequency, and operate in the range of approximately  $\pm 20\%$  of the target frequency. In addition, when operating off-resonance, the energy consumption increases significantly. For applications that require a wide range of operating frequencies, this characteristic is undesirable. Therefore, novel power-clock generation techniques are essential to efficiently generate a charge-recovery supply for a wider range of operating frequencies.

### **8.3.3 AC-Powered Logic for IoT Devices**

AC-powered logic is a type of logic family that relies only on an alternating current (AC) source, without another direct current (DC) supply or ground. Much attention has been drawn to the idea of Internet of Things (IoT). One challenge to widely distribute and connect all these billions of "things" or tiny devices and collect useful data from them is how to supply power to them. Wireless powering or energy harvesting are two obvious solutions. However, the efficiency of these techniques is limited due to the use of AC-DC and DC-DC converters and voltage regulators. Using AC-powered logic becomes appealing, eliminating the use of these converters. Since charge-recovery logic utilizes sinewave-like supplies, it is potentially particularly



suitable for implementing AC-powered logic.

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] G. Moore, “Progress in digital integrated electronics,” in *International Electron Devices Meeting*, vol. 21, 1975, pp. 11–13.
- [2] R. Dennard *et al.*, “Design of ion-implanted MOSFET’s with very small physical dimensions,” *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct 1974.
- [3] H. Esmaeilzadeh *et al.*, “Dark silicon and the end of multicore scaling,” *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May 2012.
- [4] B. Zhai *et al.*, “Energy-efficient subthreshold processor design,” *IEEE Transactions on VLSI Systems*, vol. 17, no. 8, pp. 1127–1137, Aug 2009.
- [5] W. Athas *et al.*, “Low-power digital systems based on adiabatic-switching principles,” *IEEE Transactions on VLSI Systems*, vol. 2, no. 4, pp. 398–407, Dec 1994.
- [6] W. Athas *et al.*, “A low-power microprocessor based on resonant energy,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1693–1701, Nov 1997.
- [7] S. Kim and M. Papaefthymiou, “True single-phase adiabatic circuitry,” *IEEE Transactions on VLSI Systems*, vol. 9, no. 1, pp. 52–63, Feb 2001.
- [8] C. Ziesler *et al.*, “Energy recovering ASIC design,” in *IEEE Computer Society Annual Symposium on VLSI*, Feb 2003, pp. 133–138.
- [9] V. Sathe *et al.*, “A 1.1GHz charge-recovery logic,” in *IEEE International Solid-State Circuits Conference*, Feb 2006, pp. 1540–1549.
- [10] W.-H. Ma *et al.*, “A 187MHz subthreshold-supply robust FIR filter with charge-recovery logic,” in *IEEE Symposium on VLSI Circuits*, Jun 2009, pp. 202–203.
- [11] V. Sathe *et al.*, “Energy-efficient GHz-class charge-recovery logic,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 38–47, Jan 2007.
- [12] J. Kao *et al.*, “Energy-efficient low-latency 600 MHz FIR with high-overdrive charge-recovery logic,” *IEEE Transactions on VLSI Systems*, vol. 20, no. 6, pp. 977–988, Jun 2012.

- [13] W.-H. Ma *et al.*, “187MHz subthreshold-supply charge-recovery FIR,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 793–803, Apr 2010.
- [14] Y. Zhang *et al.*, “A 1pJ/cycle processing engine in LDPC application with charge recovery logic,” in *IEEE Asian Solid-State Circuits Conference*, Nov 2011, pp. 213–216.
- [15] A. Kramer *et al.*, “Adiabatic computing with the 2n-2n2d logic family,” in *IEEE Symposium on VLSI Circuits*, Jun 1994, pp. 25–26.
- [16] A. Dickinson and J. Denker, “Adiabatic dynamic logic,” *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 311–315, Mar 1995.
- [17] J. Hu *et al.*, “A new dual transmission gate adiabatic logic and design of an 8×8-bit multiplier for low-power DSP,” in *International Conference on Signal Processing*, vol. 1, Aug 2004, pp. 559–562.
- [18] X. Peng *et al.*, “A 115mW 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS,” in *IEEE Asian Solid-State Circuits Conference*, Nov 2011, pp. 317–320.
- [19] S.-W. Yen *et al.*, “A 5.79-Gb/s energy-efficient multirate LDPC codec chip for IEEE 802.15.3c applications,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 9, pp. 2246–2257, Sep 2012.
- [20] Y. S. Park *et al.*, “Low-power high-throughput LDPC decoder using non-refresh embedded DRAM,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 3, pp. 783–794, Mar 2014.
- [21] M. Weiner *et al.*, “A scalable 1.5-to-6Gb/s 6.2-to-38.1mW LDPC decoder for 60GHz wireless networks in 28nm UTBB FDSOI,” *IEEE International Solid-State Circuits Conference*, pp. 464–465, Feb 2014.
- [22] T.-C. Ou *et al.*, “An 821MHz 7.9Gb/s 7.3pJ/b/iteration charge-recovery LDPC decoder,” *IEEE International Solid-State Circuits Conference*, pp. 462–463, Feb 2014.
- [23] T.-C. Ou *et al.*, “A 934MHz 9Gb/s 3.2pJ/b/iteration charge-recovery LDPC decoder with in-package inductors,” *IEEE Asian Solid-State Circuits Conference*, Nov 2015, to be published.
- [24] C. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, Oct 1948.
- [25] C. Bennett and R. Landauer, “The fundamental physical limits of computation,” *Scientific American*, vol. 253, no. 1, pp. 38–46, Jul 1985.
- [26] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, Jul 1961.

- [27] C. Bennett, "Logical reversibility of computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, Nov 1973.
- [28] W. Pan and M. Nolasani, "Reversible logic," *IEEE Potentials*, vol. 24, no. 1, pp. 38–41, Feb 2005.
- [29] E. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, vol. 21, no. 3-4, pp. 219–253, 1982.
- [30] W. Athas and L. Svensson, "Reversible logic issues in adiabatic CMOS," in *Proceedings of Workshop on Physics and Computation*, Nov 1994, pp. 111–118.
- [31] S. Younis, "Asymptotically zero energy computing using split-level charge recovery logic," *PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA*, 1994.
- [32] J. Lim *et al.*, "A 16-bit carry-lookahead adder using reversible energy recovery logic for ultra-low-energy systems," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 6, pp. 898–903, Jun 1999.
- [33] A. Kramer *et al.*, "2nd order adiabatic computation with 2N-2P and 2N-2N2P logic circuits," in *IEEE International Symposium on Low Power Design*, 1995, pp. 191–196.
- [34] Y. Moon and D.-K. Jeong, "An efficient charge recovery logic circuit," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 4, pp. 514–522, Apr 1996.
- [35] V. Oklobdzija *et al.*, "Pass-transistor adiabatic logic using single power-clock supply," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 10, pp. 842–846, Oct 1997.
- [36] D. Maksimovic *et al.*, "Clocked CMOS adiabatic logic with integrated single-phase power-clock supply," *IEEE Transactions on VLSI Systems*, vol. 8, no. 4, pp. 460–463, Aug 2000.
- [37] S. Kim *et al.*, "A true single-phase energy-recovery multiplier," *IEEE Transactions on VLSI Systems*, vol. 11, no. 2, pp. 194–207, 2003.
- [38] "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," *IEEE Std. 802.16e*, Feb 2006.
- [39] "IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," *IEEE Std. 802.3an*, Sep 2006.

- [40] “ETSI Standard TR 102 376 V1.1.1: Digital Video Broadcasting (DVB) User Guidelines for the Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications (DVB-S2),” *ETSI Std. TR 102 376*, Feb 2005.
- [41] G. Dong *et al.*, “On the use of soft-decision error-correction codes in nand flash memory,” *IEEE Transactions on Circuits and Systems I*, vol. 58, no. 2, pp. 429–439, Feb 2011.
- [42] R. Gallager, “Low-density parity-check codes,” *MIT Press, Cambridge, MA*, 1963.
- [43] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar 1999.
- [44] W. Athas *et al.*, “A resonant signal driver for two-phase, almost-non-overlapping clocks,” *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 129–132, May 1996.
- [45] M. Fossorier *et al.*, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation,” *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 673–680, May 1999.
- [46] F. Kschischang and B. Frey, “Iterative decoding of compound codes by probability propagation in graphical models,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 219–230, Feb 1998.
- [47] K. Chu and D. Pulfrey, “Design procedures for differential cascode voltage switch circuits,” *IEEE Journal of Solid-State Circuits*, vol. 21, no. 6, pp. 1082–1087, 1986.
- [48] R. Puri *et al.*, “Logic optimization by output phase assignment in dynamic logic synthesis,” in *IEEE/ACM International Conference on Computer-aided Design*, 1996, pp. 2–7.
- [49] J. Kao *et al.*, “A charge-recovery 600MHz FIR filter with 1.5-cycle latency overhead,” *IEEE European Solid-State Circuits Conference*, pp. 160–163, Sep 2009.
- [50] V. Sathe *et al.*, “Resonant clock design for a power-efficient high-volume x86-64 microprocessor,” *IEEE International Solid-State Circuits Conference*, pp. 68–70, Feb 2012.