Comparative Analysis of Near-Threshold and Charge Recovery Circuits for Energy Efficiency and Performance

A Thesis Presented

by

Ziqi Zhang

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Electrical Engineering

Stony Brook University

May 2013

Stony Brook University

The Graduate School

Ziqi Zhang

We, the thesis committee for the above candidate for the
Master of Science degree, hereby recommend
acceptance of this thesis.

Dr. Emre Salman - Thesis Advisor
Assistant Professor, Electrical and Computer Engineering Department

Dr. Peter Milder - Second Reader
Assistant Professor, Electrical and Computer Engineering Department

This thesis is accepted by the Graduate School

Charles Taber
Interim Dean of the Graduate School

Abstract of the Thesis

Comparative Analysis of Near-Threshold and Charge Recovery Circuits for Energy Efficiency and Performance

by

Ziqi Zhang

Master of Science

in

Electrical Engineering

Stony Brook University

2013

Two existing energy-efficient circuit design methods, near-threshold and charge-recovery circuits (adiabatic switching), are compared for energy efficiency and performance. Near-threshold logic circuits work in the region where the supply voltage is around the threshold voltage of a transistor, which has slightly higher energy dissipation than sub-threshold logic circuits, but significantly less performance penalty. Two energy-recovery circuit families (efficient charge recovery logic and clocked CMOS adiabatic logic) that implement adiabatic switching are also considered. In adiabatic switching, a single AC signal source is used as both power and clock signals. Thus, this AC signal is typically referred as power-clock signal. A comprehensive comparative analysis is performed on these circuits. Trade-offs between energy efficiency and performance are quantified. To explore the potential

of charge recovery circuits, in addition to the normal operating regime, two other possibilities are explored. It is demonstrated that performance of charge recovery circuits can be enhanced by exploiting the timing relationship between data and power-clock signals. Furthermore, low voltage charge recovery circuits are also included in the comparative analysis to demonstrate the effect of voltage scaling on these type of circuits. Specifically, when the peak voltage of the power-clock signal is scaled from 2.5 V to 1.5 V, the energy delay product of charge recovery circuits is approximately 73% less than near-threshold circuits. This situation occurs if the input data signal transitions when the power-clock signal reaches half of the peak voltage.

To my dearest parents,
Yiqun Zhang and Huimin Chen,
and my love, Xinyun Huang.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

The last two years will remain a wonderful memory in my life that I will never forget. I will cherish for what I've learned and achieved at Stony Brook University.

I will owe my gratitude to my advisor, Professor Emre Salman for providing me an opportunity to work with him and finish my master's thesis under his continuous patience and guidance. His rigorous attitude to his work and research has made a good example not only for the last 15 months, but for the rest of my career. Working with him helped me get a better understanding of my academic knowledge and a more comprehensive way of thinking and studying. I appreciate everything you've done for me professor. I should also thank Professor Peter Milder for reading my thesis and providing valuable comments.

I am grateful to be a member of Nanoscale Circuits and Systems Laboratory, thanks to all the members in the lab for the help and great company in learning. A special thanks to Hailang Wang, also a member in NanoCAS Laboratory, for being a great friend and offering me endless help no matter at work or in everyday's life. His advice and help on me have been very valuable, I appreciate that again.

Xinyun Huang, I am so lucky to have you as my wife, even though we have been in different countries, your unconditional love and endless support have meant a lot to me.

Lastly and most importantly, I'd like to thank my parents. Without their continuous love and support, I cannot get anything at all. I will do everything to return the love. Thank you.

Chapter 1

Introduction

Over the past decades, integrated circuits (ICs) have gained higher density and performance, following Moore's Law [1]. Power consumption, however, has become a primary issue for ICs in almost any application. Mobile devices, such as portable computers and cellular phones, are operating faster at reduced form factors. While battery technology has been improved over the years, a battery with a certain size has limited energy capacity [2]. To obtain a better battery performance while maintaining (or reducing) the size and weight, low power ICs are required.

Alternatively, high performance platforms that are permanently connected to power supply consume high power. An estimate shows that in 2006, data centers consumed about 61 billion kilowatt-hours (kWh), roughly 1.5% of total U.S. electricity consumption or about $4.5 billion in electricity costs [3]. Alarmingly, assuming these trends continue, the total energy consumption for United States by servers and data centers is expected to nearly double in five years.

In 1980s, CMOS replaced NMOS-based technologies with its high noise immunity and lower power consumption. However, the overall power consumption of computing devices has been increasing at about 22% per year [3]. This increase

in consumption comes even though the energy consumed by a single CMOS logic gate has fallen exponentially with Moore's Law [4]. Thus, novel circuit design techniques are required to reduce power consumption.

Multiple power management techniques have been developed during the past decades. One approach is voltage scaling. A popular technique referred to as sub-threshold design represents an ultimate point of voltage scaling, which obtains as much as 10X energy reduction [5], [6]. Another technique, referred to as adiabatic logic [7], [8], [9], [10], also achieves lower energy dissipation than conventional CMOS logic by having a different charging/discharging pattern. Unfortunately, these power reduction technologies face a penalty of reduced speed.

To evaluate and compare the energy-efficiency and performance of these circuits, several transistor-level gates with different channel widths have been designed, including an inverter, NAND and NOR. Four primary criteria have been considered [11]:

- Energy per switching,
- Energy over 300 cycles,
- Delay,
- Energy delay product.

Despite the significant existing work on low-voltage [5] and charge recovery circuits [12], [13], a comprehensive comparison of these two circuit design methods has not received much attention. The primary purpose of this thesis is to evaluate multiple criteria for both near-threshold and charge recovery circuits. For charge

recovery circuits, in addition to the normal operating regime, two other possibilities are explored that achieve a different balance between performance and energy efficiency. Furthermore, voltage scaling is applied to charge recovery circuits.

The rest of the thesis is organized as follows. The primary principles of near-threshold logic and charge recovery circuits are summarized in Chapter 2, in which several charge recovery logic styles are also described. Comparative analysis of near-threshold and charge recovery circuits is addressed in Chapter 3. Simulation results are also presented in this chapter. Finally, the thesis is concluded in Chapter 4.

Chapter 2

Background

Both near-threshold logic and charge recovery circuits have potential to achieve low power with a penalty of reduced speed. In this chapter, basic principles of near-threshold computing and charge recovery circuits are summarized. Also, two kinds of charge recovery circuit families are described.

2.1 Near-threshold Computing

Modern CMOS circuits consume significant energy during charging and discharging process of their internal nodes. Thus, lowering supply voltage (voltage scaling) can reduce the energy consumption effectively. CMOS circuits can operate at very low voltages, even at a voltage that is lower than or close to the threshold voltage (V_{th}). Operation at near-threshold voltage has drawn interest during the last five years to widen the application scope of sub-threshold circuits [5], [14].

The operation region of a transistor can be divided into three parts:

- $V_{GS} > V_{th}$, super-threshold regime;
- $V_{GS} \sim V_{th}$, near-threshold regime;

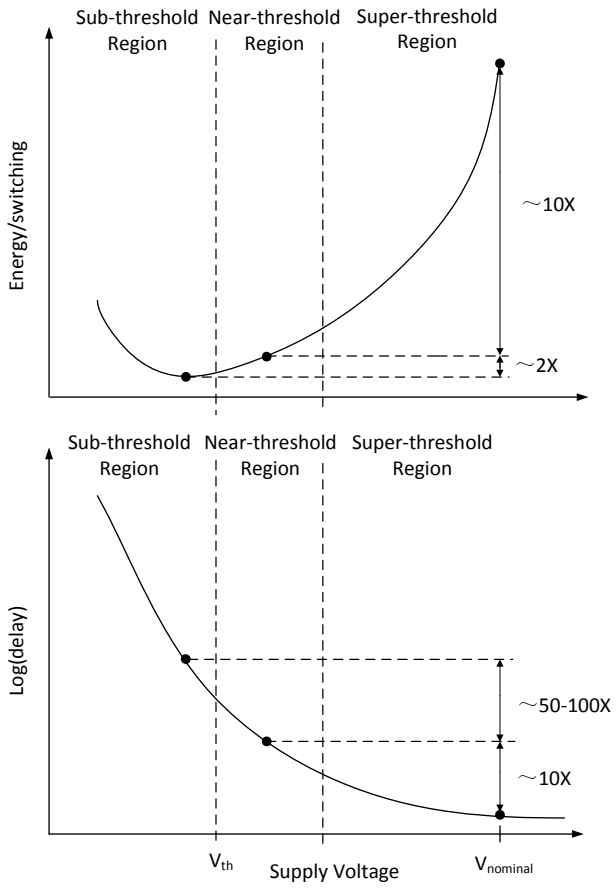

Figure 2.1: Energy and delay characteristics at different voltage supplies.

- $V_{GS} < V_{th}$, sub-threshold regime.

Figure 2.1 shows the energy and delay characteristics in different supply voltages. Both sub-threshold and near-threshold regimes achieve significant energy reduction. However, energy reduces only by 2X when V_{dd} is scaled from near-threshold regime to the sub-threshold regime, while the delay rises by 50-100X. Thus, near-threshold regime, which has a delay penalty that is relatively more acceptable, is a more reasonable choice for applications where speed is important. Therefore, there has been a growing interest on near-threshold logic [5], [14], [15].

2.2 Adiabatic Logic

Adiabatic logic refers to a set of low-power electronic circuits that implement reversible logic [10]. A system is reversible if no information of its state is lost over the entire transformation process [16]. However, since reversible logic requires no loss during operation, it is not practical to implement from the circuit design point of view. Some circuits, which partially apply the principles of adiabatic switching, achieve significantly low (but not zero) energy consumption during computation. Since some of the energy is recovered in these circuits, they are referred to as *charge recovery* or *charge recycling circuits* [17]. The term *charge recovery circuits* is used in this thesis. Adiabatic switching lowers power consumption by following two fundamental rules [10]:

- Never turn on a transistor when there is a voltage difference between the drain and source;
- Never turn off a transistor while there is current flowing through it.

The rest of the section explains the principle of adiabatic switching. Two representative circuit families are introduced in Section 2.2.2 and Section 2.2.3.

2.2.1 Principle of adiabatic switching

To better understand adiabatic switching, comparison between conventional and adiabatic switching is helpful [17], [18], [19]. Figure 2.2 shows the energy dissipation process during one switching cycle in conventional CMOS circuits [17]. The LOW-to-HIGH transition on a circuit node can be modeled as charging an *RC* tree through a switch. *C* is the capacitance of this node and *R* represents the resistance

of the switch and interconnect. When the switch is closed, V_{dd} ($2V_A$) is directly applied to R and current flows through R instantaneously. C is charged to full V_{dd} within a short period of time. Over the entire process, the energy taken from the power source is $CV_{dd}{}^2$. Half of this energy, $\frac{1}{2}CV_{dd}{}^2$, is dissipated by R.

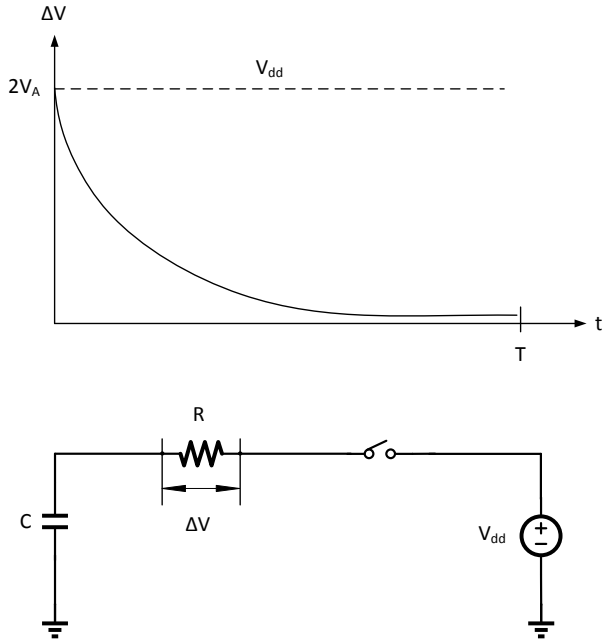

Figure 2.2: Charging an RC tree through a switch when $V_{dd} = 2V_A$.

Alternatively, adiabatic circuits exhibit a different pattern in the charging process [9]. Figure 2.3 shows the energy dissipation during one switching cycle in an adiabatic logic circuit. To follow the two fundamental rules mentioned before, a time-varying voltage source (AC) is used with a peak-to-peak voltage of V_{dd}. This voltage source behaves as both power and clock signals. Thus, it is referred to as power-clock signal. When the switch is closed, approximately zero voltage is applied across R. Voltage gradually increases in a slow pace, and the voltage dif-

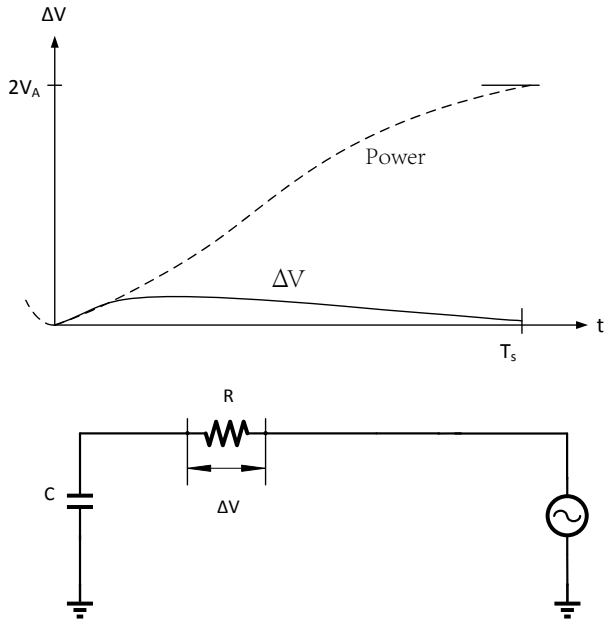

Figure 2.3: Adiabatic charging of an *RC* tree.

ference across R is maintained small throughout the entire charging process. Furthermore, by spreading out the transition more evenly, the peak current is greatly reduced. These factors significantly reduce the energy dissipation by R. The overall energy dissipated in this transition can be calculated as follows [7].

Suppose the voltage across C is $v_C(t)$, and the voltage across the voltage source is $v(t)$. As mentioned before, the voltage increases in a slow rate. Thus, $v_C(t)$ is considered to instantly follow $v(t)$,

$$v_C(t) \approx v(t). \tag{2.1}$$

Therefore the current flow through R can be calculated by

$$i(t) = C\frac{dv(t)}{dt} = \frac{CV_{dd}}{T_s}, \tag{2.2}$$

where T_s is the transition time over the entire switching process.

The energy dissipation within this switching can be calculated by integrating the power, $p(t)$, over the transition time T_s,

$$E = \int_0^{T_s} p(t)\mathrm{d}t = \int_0^{T_s} v(t) \cdot i(t)\mathrm{d}t = \int_0^{T_s} [v_R(t) + v_C(t)]i(t)\mathrm{d}t. \tag{2.3}$$

Since capacitance does not dissipate any energy, the integral of $v_C(t)i(t)$ over one cycle is zero. Also, $v_R(t)$ can be replaced by $i(t)R$. Combining (2.2) and (2.3) leads to

$$E = \int_0^{T_s} R\frac{C^2{V_{dd}}^2}{{T_s}^2}\mathrm{d}t = \frac{RC}{T_s}C{V_{dd}}^2. \tag{2.4}$$

Since an adiabatic switching cycle includes charging and recovering, the total energy consumption is

$$E = 2\frac{RC}{T_s}C{V_{dd}}^2. \tag{2.5}$$

According to 2.5, a longer T_s reduces the overall energy dissipation. With an indefinitely extended computation time, the circuit can achieve vanishingly low energy dissipation [17]. A variety of circuit families has been developed such as 2N-2N2D, 2N-2P, and positive feedback adiabatic logic (PFAL) [20], [21], [22], [23], [24], [25]. In this section, two representative circuit families, efficient charge recovery logic (ECRL) and clocked CMOS adiabatic logic (CAL), are reviewed.

2.2.2 Efficient Charge Recovery Logic

A representative member of multiple-phase charge-recovery families is the efficient charge recovery logic (ECRL) [26]. Figure 2.4 illustrates the schematic of an ECRL

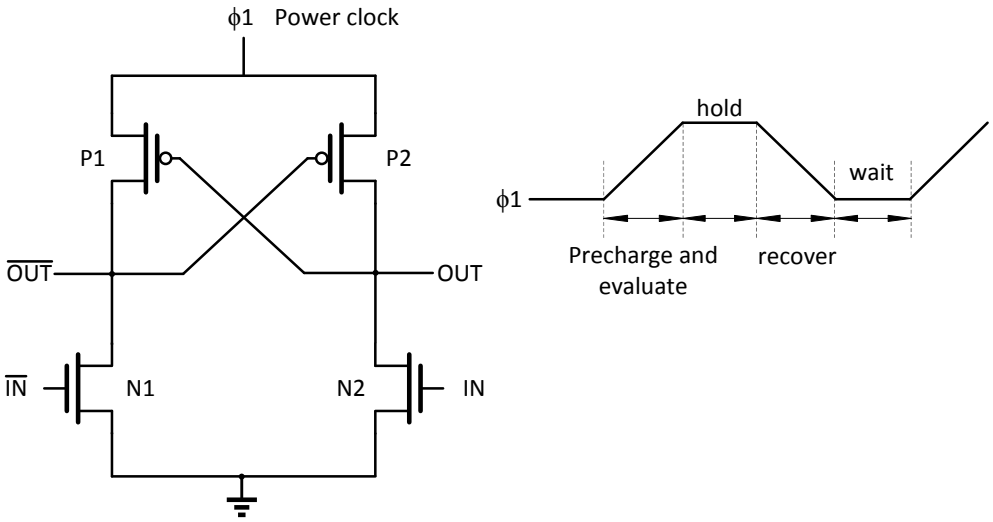

Figure 2.4: ECRL inverter.

inverter. The two cross-coupled PMOS transistors act as memory to store the circuit status, while the functionality is achieved by NMOS devices.

Notice that ECRL generates dual outputs. In contrast to conventional CMOS logic, this pattern does not consume significantly more area. Consider, for example, a NAND gate. A conventional static CMOS 2-input NAND gate consists of two NMOS transistors and two PMOS transistors, while an ECRL 2-input NAND gate consists of four NMOS and two PMOS devices. However, the ECRL gate produces two output signals and simultaneously performs two functions, a NAND and an AND. To achieve the same result, a conventional CMOS NAND gate needs another inverter that has another pair of transistors. As the complexity of the functions increases, the advantage of this dual-rail structure is more visible.

Adiabatic logic needs an oscillating power supply, known as the power-clock signal. Different adiabatic families operate with different power-clock signals. For

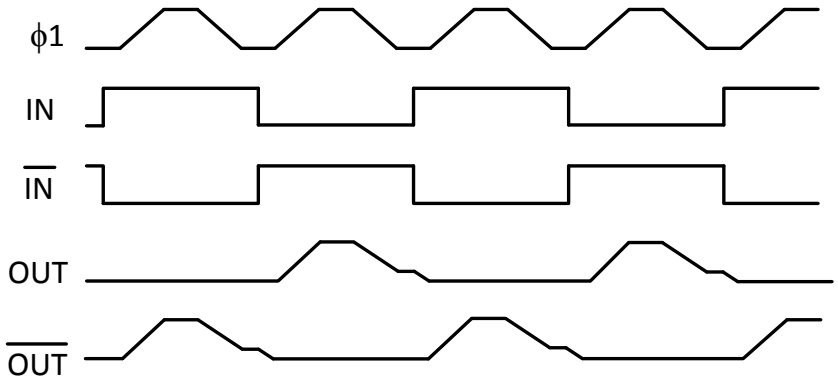

Figure 2.5: Waveforms for ECRL inverter.

example, ECRL employs a four-phase power-clock signal as depicted in Figure 2.4. Within one power-clock cycle, there are four intervals:

- Evaluate interval (E), during which the circuit generates output from the stable input;
- Hold interval (H), during which the output remains stable to provide stable input for the next stage;
- Recover interval (R), during which energy is recovered;
- Wait interval (W), which is inserted for symmetry reasons.

To understand the basic operation of this circuit (see Figure 2.5), initially assume the input IN is high and $\overline{IN}$ is low. As Figure 2.4 shows, at the beginning of this process, power-clock signal φ1 starts from 0 and rises towards V_{dd}. N2 conducts and OUT is connected to ground, while N1 is disabled. Since gate of P1 is connected to OUT, when φ1 exceeds $V_{th,p}$, P1 turns on and connects $\overline{OUT}$ to power-clock signal φ1. $\overline{OUT}$ thus follows φ1 through P1. These values remain

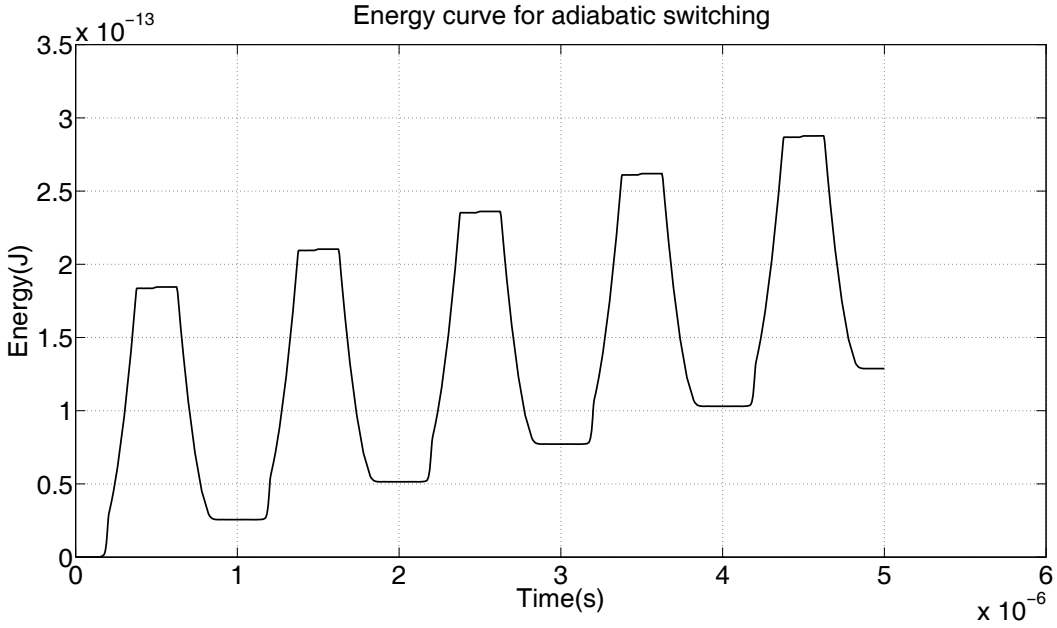

Figure 2.6: Energy waveform of ECRL circuit.

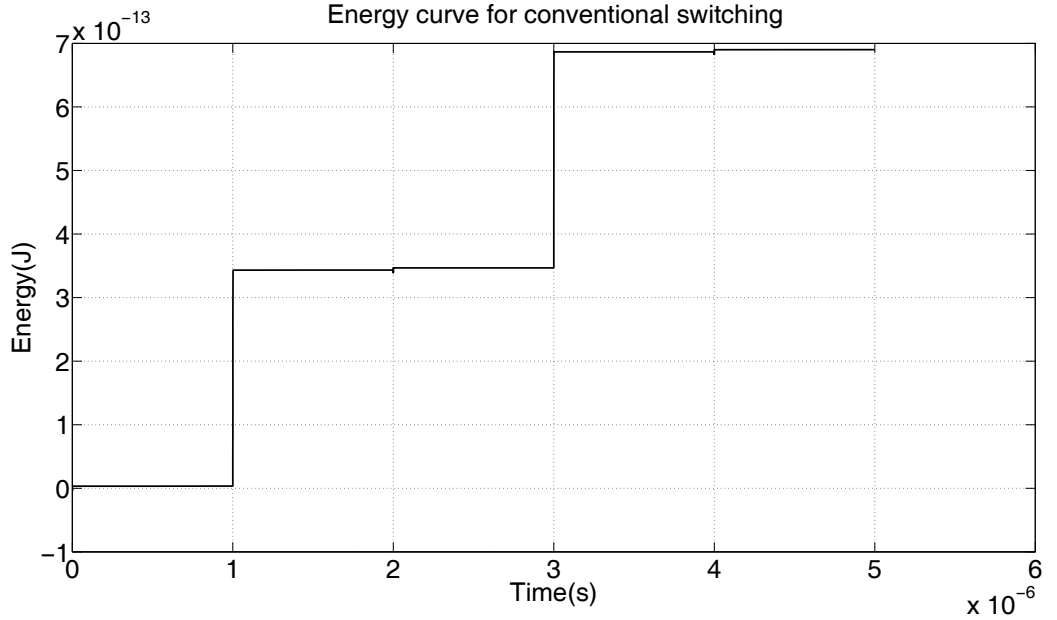

Figure 2.7: Energy waveform of conventional CMOS logic circuit.

constant during hold phase and act as the input used in the evaluation phase of the next stage. When hold phase ends, φ1 falls from V_{dd} to zero, during which $\overline{OUT}$ returns its energy to φ1. Thus, charge is recovered. As long as $\overline{IN}$ remains low, $\overline{OUT}$ is at least $V_{th,p}$ to keep P1 turned on. Thus, a fraction of energy $\frac{1}{2}C_{out}V_{th,p}^2$ remains on that node. Depending on the input signals, this energy may be discharged or reused in the next cycle. This energy loss is significantly smaller as compared to that of conventional static CMOS logic. Figure 2.5 shows how the output changes when the input is driven by a step function.

The energy dissipated over five clock cycles of ECRL is shown in Figure 2.6. For comparison, energy consumption of conventional static CMOS circuit is also provided in Figure 2.7. Energy is recovered during the recover interval for ECRL circuits. Thus, its energy consumption for one transition is approximately $30fJ$. Alternatively, a conventional static CMOS gate consumes approximately $350fJ$ for a single transition. This is more than an order of magnitude greater than an ECRL gate.

To cascade logic gates, each clock has a 90° phase lag with the next gate. The phase relationship is shown in Figure 2.8 [26]. With this pattern, when the next stage is in evaluate interval, it can use the stable input from the previous stage, which is in hold phase.

2.2.3 Clocked CMOS Adiabatic Logic

Another charge recovery circuit family, which is a variant of ECRL, is referred to as clocked CMOS adiabatic logic (CAL). CAL also adopts a dual-rail logic, but it only operates with a single-phase power-clock signal, which is different from ECRL [27, 28].

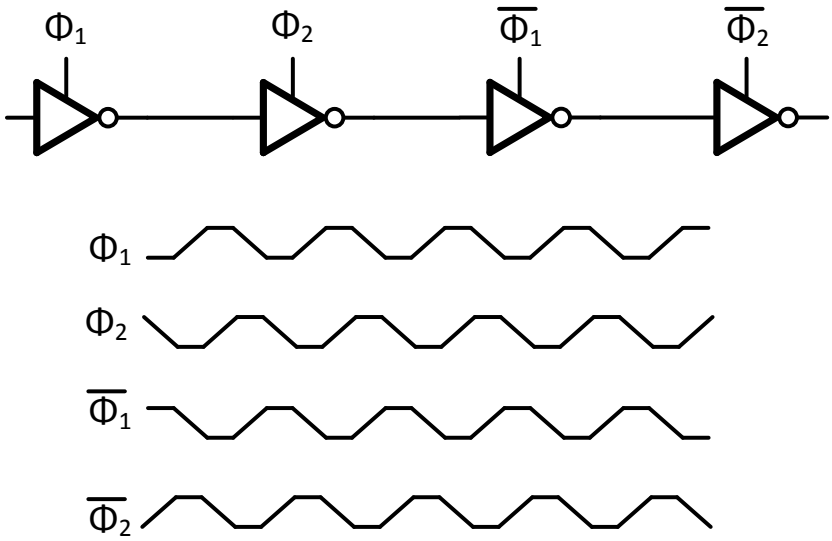

Figure 2.8: Inverter chain and four-phase power-clock signal.

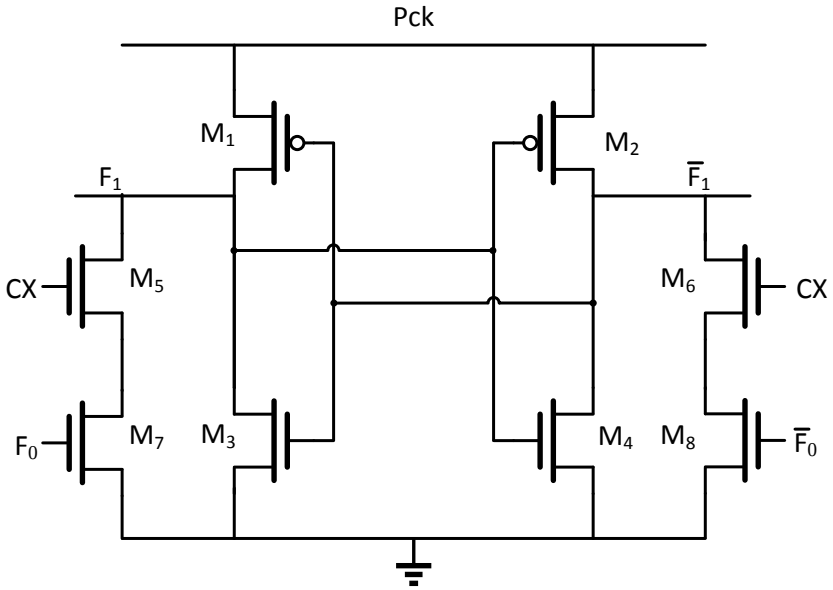

Figure 2.9: CAL inverter.

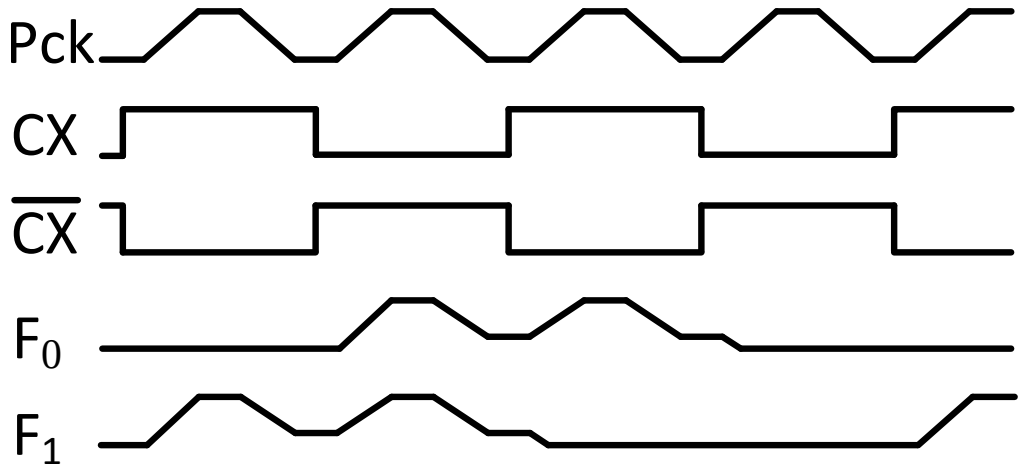

Figure 2.10: Waveforms for CAL inverter.

Figure 2.9 shows a CAL inverter. Unlike ECRL, which consists of two cross-coupled PMOS transistors to store node voltage, CAL employs a pair of cross-coupled CMOS inverters (transistors M_1-M_4) to provide the memory function. Functional unit is constructed by NMOS transistors (M_7 and M_8 in Figure 2.9). In addition to the power-clock signal, an auxiliary clock signal is added to the circuit as a path control switch. In Figure 2.9, M_5 and M_6, in series with the functional part, are signaled by this auxiliary clock signal.

Ideal waveforms for CAL inverter are shown in Figure 2.10. Power-clock signal (Pck) is a trapezoidal waveform. CX represents the auxiliary clock signal. Assume initially that CX is high, F_0 is low and its complement input $\overline{F_0}$ is high, so M_7 remains disabled and M_8 conducts. Thus $\overline{F_1}$ is connected to ground since CX turns on M_5 and M_6. $\overline{F_1}$ thus turns on M1. F_1 follows Pck. In the next cycle, CX changes to zero, disconnects both input signals. During this cycle, the output F_1 and its complement $\overline{F_1}$ will store the values from previous cycle. As a result, CAL produces

an output pattern with presence or absence of a pair of subsequent pulse waves.

To design a chain of logic gates, unlike ECRL which needs accurate timing relationship (because it employs a four-phase power-clock signal), the CAL gates are operated with a single-phase power-clock signal. This simplifies the process of clocking. The logic evaluation in the chain is enabled by alternating auxiliary clock CX and its complement $\overline{CX}$ (Figure 2.10).

Chapter 3

Comparative analysis of near-threshold and charge-recovery circuits

With growing interest on both near-threshold and charge recovery circuits, a fair comparison between these circuit techniques is worth exploring. This chapter starts with introducing analysis approach employed in this work, including the analysis environment, voltage scaling, and the comparison method. Section 3.2 presents the results obtained from transient analysis.

3.1 Analysis approach

3.1.1 Simulation setup

The entire analysis process, including the circuit design and transient analysis, is accomplished within Cadence environment using TSMC 0.25 μm CMOS technology. For a comprehensive comparison, three gates (Inverter, NAND, and NOR) with three sizes (450 nm, 4.5 μm, 9 μm), are designed for each logic style. At the input, step functions are applied, and a 50 fF load capacitance is added to the out-

put. For near-threshold logic, a 500 mV DC voltage source is applied as the power supply voltage. For charge recovery circuits, a sinusoidal signal (AC source) is used as the power-clock signal.

Each charge recovery gate (either ECRL or CAL) consists of a memory element and functional blocks, which is similar to a conventional static CMOS gate with a latch, so the charge recovery circuit is inherently pipelined, and the delay consists of the data-to-clock delay and the clock-to-output delay where clock is the power-clock signal. Thus, the arrival time of data is important. With this sinusoidal power-clock signal, both ECRL and CAL logic are analyzed with different input patterns:

- Case 1: Data transitions when power-clock signal is at zero voltage,
- Case 2: Data transitions when power-clock signal is at $\frac{1}{2}V_{dd}$,
- Case 3: Data transitions when power-clock signal is at V_{dd}.

Figure 3.1 shows the timing relationship described above. Case 1 is the conventional mode for charge recovery circuits, mentioned in Chapter 2, in which the circuit recovers the maximum energy. Only a fraction of $\frac{1}{2}CV_{dd}{}^2$ is dissipated depending on the input signal. Furthermore, in Case 1, the delay is constant and the largest, which is approximately $\frac{1}{4f}$ (since output follows power-clock signal), where f is the frequency of the power-clock signal.

For case 2 and case 3, since the data transitions when the power-clock signal is not 0, the energy consumption increases while delay is reduced. Thus, these cases have the potential to widen the application space of conventional charge recovery circuits. With these three cases, the performance and energy trade-offs of charge recovery circuits can be further explored, since previous research work primarily focuses on case 1.

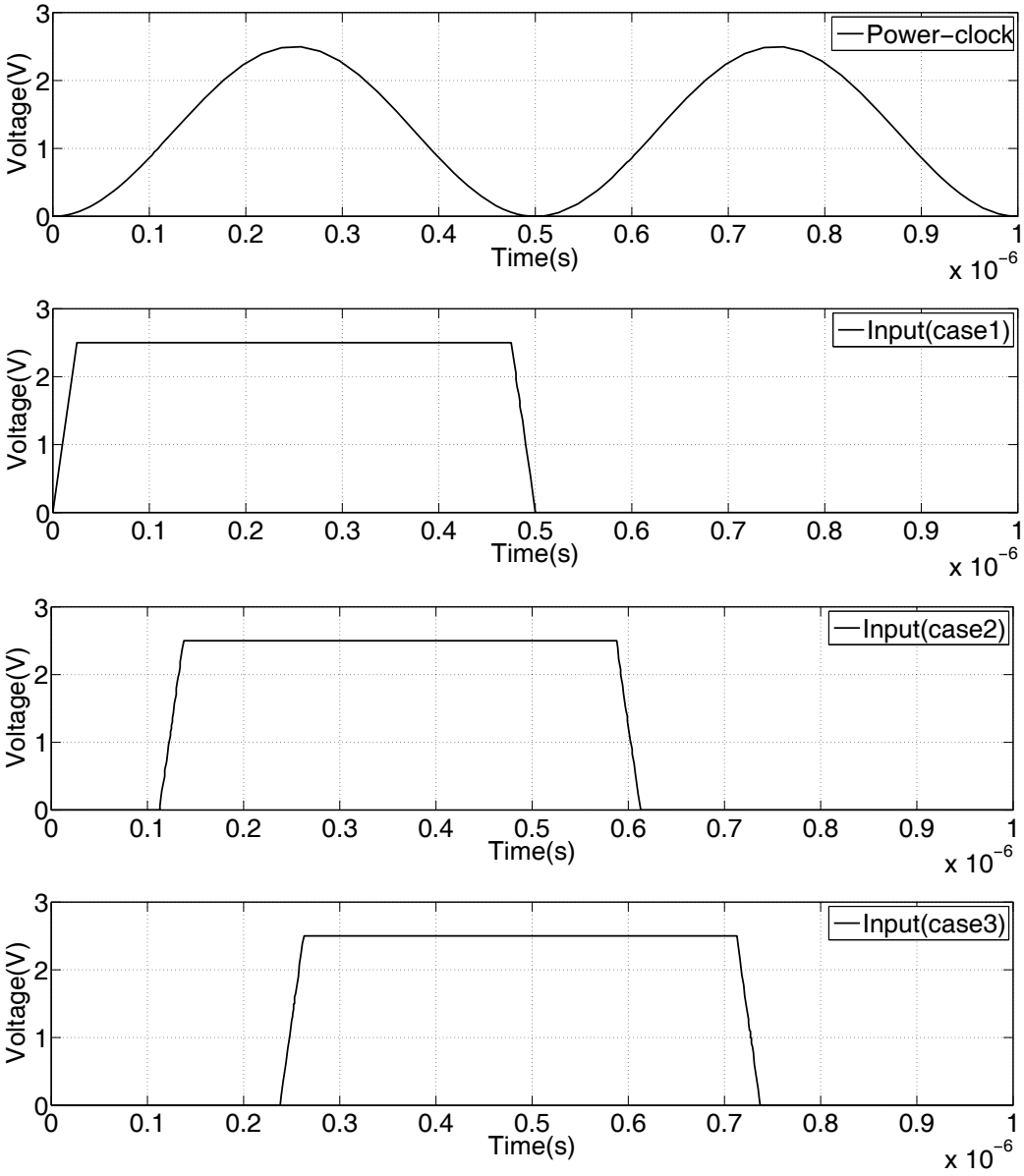

Figure 3.1: Three input patterns illustrating the three cases used to explore performance and energy efficiency trade-offs.

3.1.2 Low voltage charge-recovery circuits

The energy dissipation in charge recovery circuits is calculated by (2.5). Based on the assumption that the term $\frac{RC}{T_s}$ remains unchanged, scaling the amplitude of supply voltage (peak-to-peak voltage in AC source) also conserves energy. Thus, it is interesting to explore the efficiency of voltage scaling on charge recovery circuits. In this work, the peak-to-peak voltage of the power-clock signal of charge recovery circuits is scaled from 2.5 V to 1.5 V, and compared with near-threshold circuit operating at 500 mV DC voltage [29].

3.1.3 Evaluation criteria

To compare the aforementioned low power logic circuits, several criteria are analyzed:

- Maximum operating frequency,
- Energy per switching,
- Delay,
- Energy delay product,
- Energy over 300 cycles.

These results are obtained in Cadence using Spectre. To explore the trade-offs among the three cases of charge recovery circuits described in Section 3.1.1, the results from near-threshold logic and conventional static logic circuits are compared separately with each case of adiabatic logic circuit. The analysis is performed for both ECRL and CAL logic. Furthermore, the ECRL and CAL logic after voltage scaling is also included in the comparison [30].

3.2 Transient analysis

Comparison is achieved on all gates for each logic (Inverter, NAND, NOR). In this section, transient results for inverter are presented with three transistor widths for NMOS: 450 *nm*, 4.5 μm, and 9.0 μm. Results for other gates are included in the Appendix. For charge recovery circuits, the frequency of the power-clock signal is 2 MHz when analyzing design criteria mentioned in Section 3.1.3 except maximum operating frequency.

3.2.1 Maximum operating frequency

Table 3.1 shows the maximum operating frequency for each inverter. Near-threshold logic can work at a low frequency due to low supply voltage. For both charge recovery circuits, maximum operating frequency are lower compared to conventional static CMOS logic, but still have better performance than that of near-threshold logic. 40 MHz is the smallest maximum frequency among the results of charge recovery circuits (450 *nm* of CAL), yet it is twice the highest maximum frequency of near-threshold circuit (20 MHz). Note that with transistor sizing, up to 400 MHz clock frequency can be achieved with charge recovery circuits.

Voltage scaling affects the maximum operating frequency of charge recovery circuits. With a reduced voltage supply, current is smaller. Hence the capability to drive the transistor is also decreased, which leads to a smaller operating frequency range. For example, at 4.5 μm, both charge recovery circuits have a maximum operating frequency that is approximately 50% less.

3.2.2 Case 1 of charge recovery circuits

- **Energy per switching**

Maximum operating frequency (MHz)	450nm	4.5μm	9.0μm
ECRL 2.5V	62.5	250	400
ECRL 1.5V	40	125	250
CAL 2.5V	40	200	250
CAL 1.5V	25	125	200
Near-threshold	4	10	20
conventional static CMOS	800	4000	8000

Table 3.1: Comparison of maximum operating frequency.

Figure 3.2 shows the energy consumption per switching for each logic in case 1 of charge recovery circuits. Conventional static CMOS inverter is also included in the analysis. Note that in case 1, charge recovery logic circuits work in the conventional mode, which saves the most energy. The energy consumed in conventional static CMOS logic is significantly higher. At 450 nm width, conventional logic consumes approximately 1.2 pJ, while other logic circuits consume approximately 20 fJ.

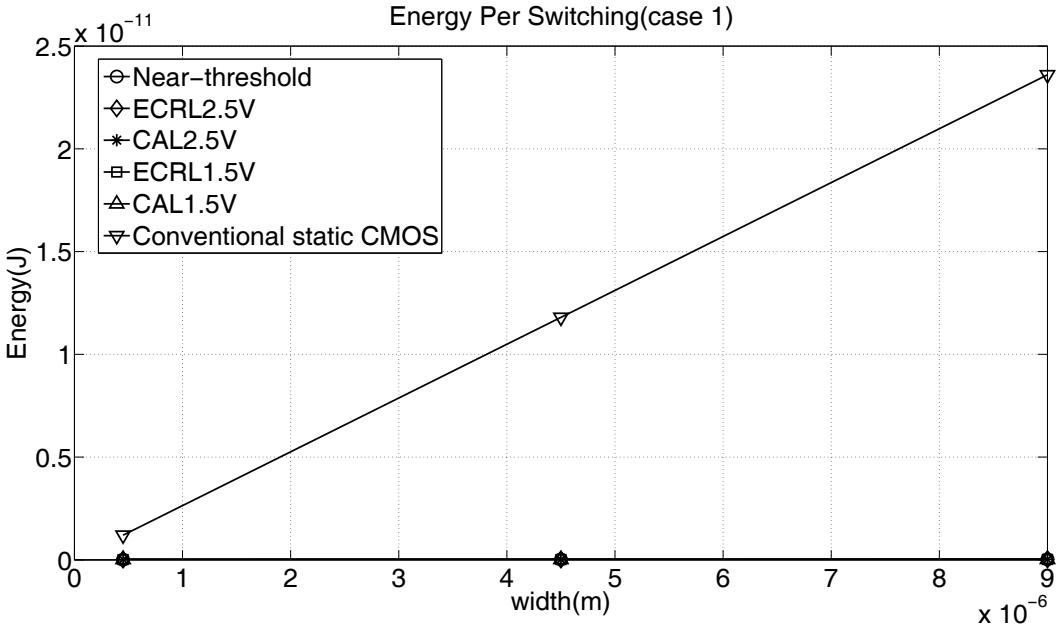

Figure 3.2: Comparison of energy per switching in case 1.

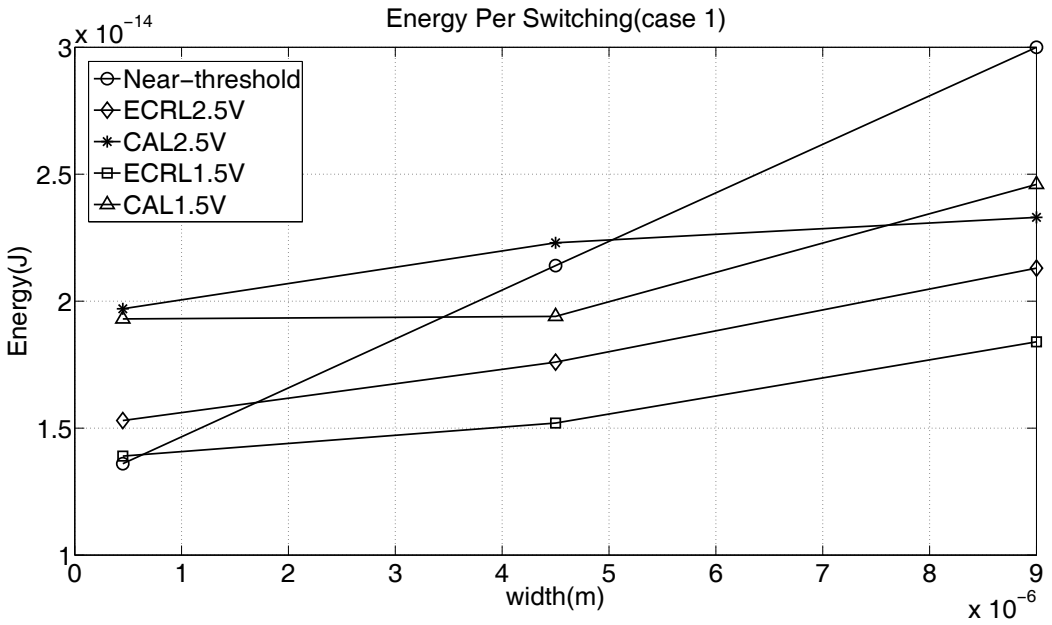

Figure 3.3: Energy per switching comparison in case 1 (conventional static CMOS is removed).

Figure 3.3 continues the comparison between the rest of the circuits. CAL and ECRL have comparable energy reduction. Near-threshold logic inverter has the smallest energy consumption with a small transistor channel width (450 nm). However, with the increase of transistor width, it gradually loses the advantage. At 9 μm, its energy consumption is 30 fJ, the highest among low power circuits. The reason is that charge recovery circuit in its normal mode only dissipates a fraction of $\frac{1}{2}C_{out}V_{th,p}^2$. Assume the load capacitance and PMOS transistor threshold voltage are fixed, a change of its own transistor size does not have much influence on the energy dissipation. Thus, when the transistor size increases, charge recovery circuits do not consume significantly more energy while near-threshold logic does. Also, with a scaled voltage supply, both CAL and ECRL logic do not achieve a considerable reduction in energy consumption per switching due to the same rea-

son. For example, at 450 *nm*, after voltage scaling, ECRL achieves only around 9% reduction.

- **Delay**

As mentioned in Section 3.1.1, delay of charge recovery circuits in case 1 is approximately $\frac{1}{4f}$, where f is the frequency of the power-clock signal. In this work, a 2 MHz clock frequency is used, so theoretically the delay for both ECRL and CAL inverter are 125 *ns* in case 1. Due to the rise/fall times, the delay is approximately 113 *ns*, as shown in Figure 3.4.

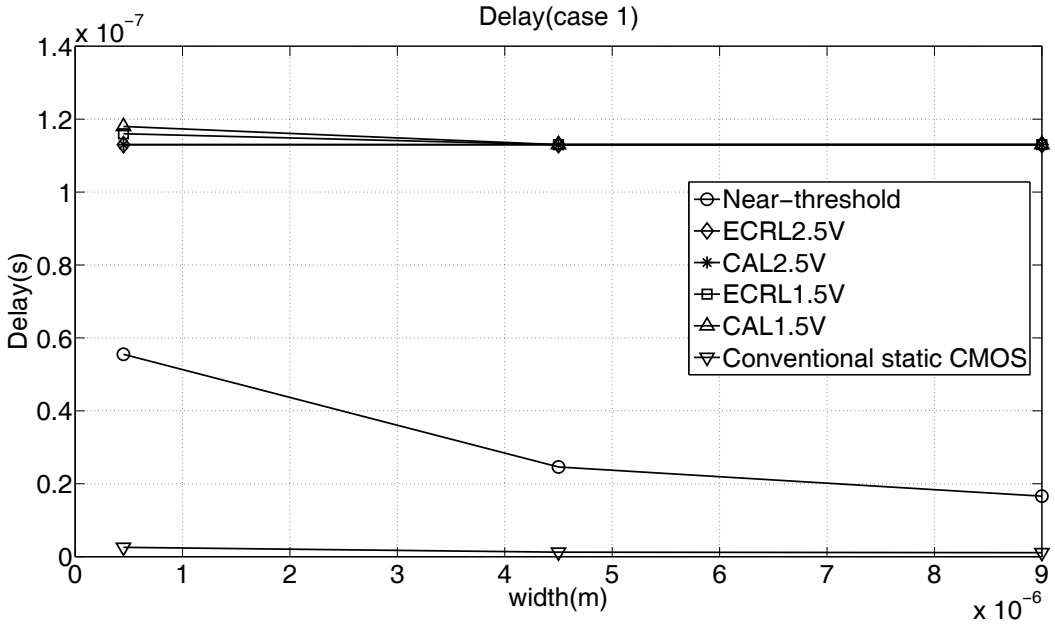

Figure 3.4: Delay comparison when charge recovery circuits are in case 1.

Results for conventional static inverter and near-threshold inverter are also given in Figure 3.4. Delay of near-threshold inverter is less than half of charge recovery inverter, but much greater than conventional static inverter.

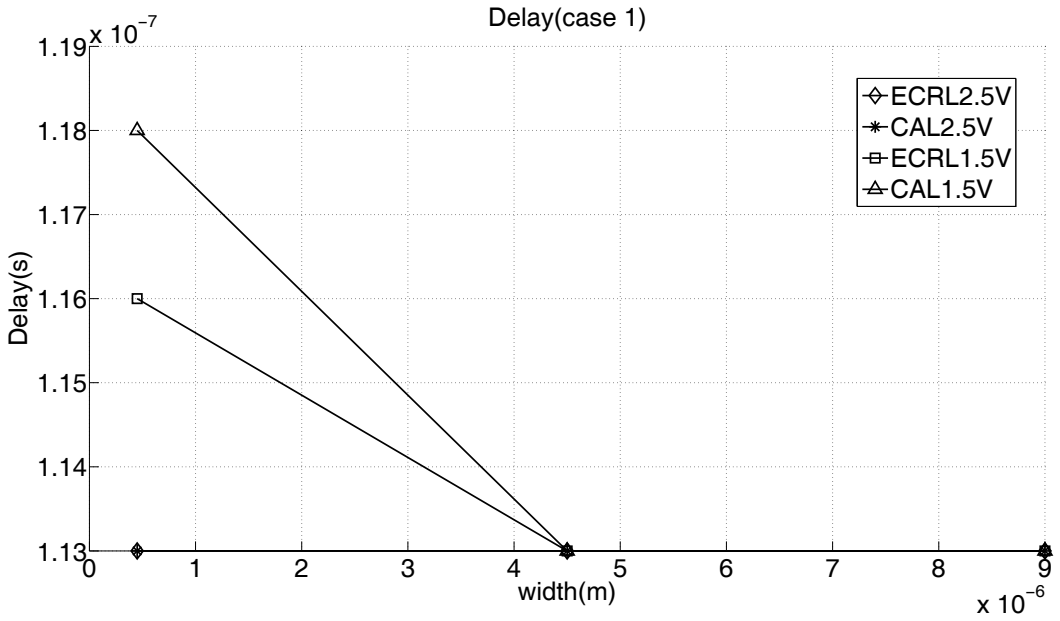

Figure 3.5: Voltage scaling effects on delay when charge recovery circuits operate in case 1.

The effect of voltage scaling is explored in Figure 3.5. With a small transistor size such as 450 *nm*, energy consumption does not decrease significantly with voltage scaling. Delay is affected by the smaller supply voltage since with a smaller current, the speed of charging or discharging slows down. With a larger width, delay does not decrease significantly.

- **Energy delay product**

Two figures show the energy delay product (EDP) comparison in case 1 of charge recovery circuits. Figure 3.6 shows graphs for all logic. It demonstrates that EDP of conventional static CMOS inverter is the highest. With the increase of the transistor width, the difference between conventional static CMOS logic and other low-power circuits is greater. At 4.5 μm, EDP of conventional static CMOS circuit is approximately 5 times of other logic circuits, and at 9 μm, it increases to 10

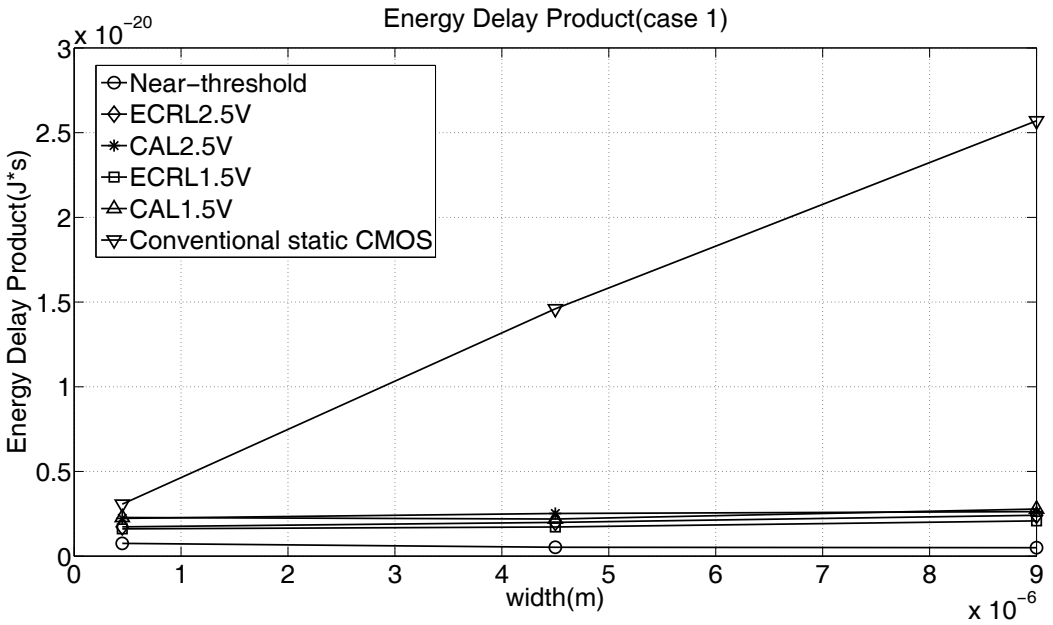

Figure 3.6: Energy delay product comparison in case 1.

times. Figure 3.7 removes the graph of conventional static CMOS inverter to better observe other graphs. Near-threshold logic inverter achieves the lowest EDP for all widths, which remains below $10^{-21}\ J \cdot s$. EDP of the two charge recovery inverters, which exceed $1.5X10^{-21}\ J \cdot s$, are slightly greater than that of near-threshold logic. After the supply voltage is scaled to $1.5V$, the energy consumption is reduced and the delay is increased, so the EDP results before and after voltage scaling are comparable.

- **Energy over 300 cycles**

Figure 3.8 shows energy accumulated over 300 cycles for each logic. The result has the same trend as energy per switching for case 1 (Figure 3.2). Conventional static CMOS consumes significantly larger energy than other circuits.

A more detailed comparison between other logic circuits is achieved in Fig-

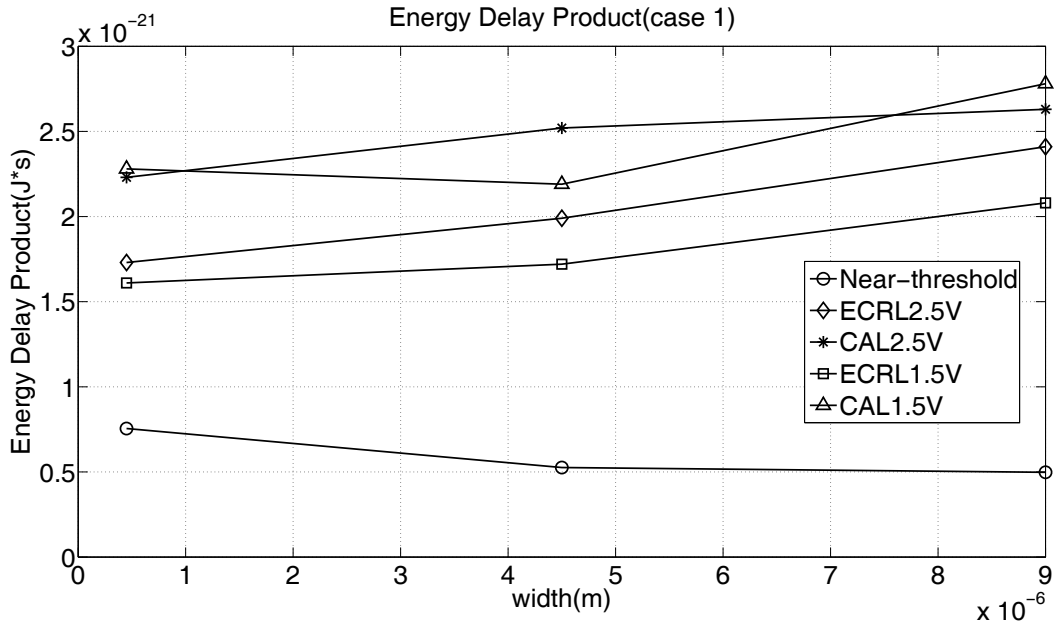

Figure 3.7: Energy delay product comparison in case 1 (conventional static CMOS is removed).

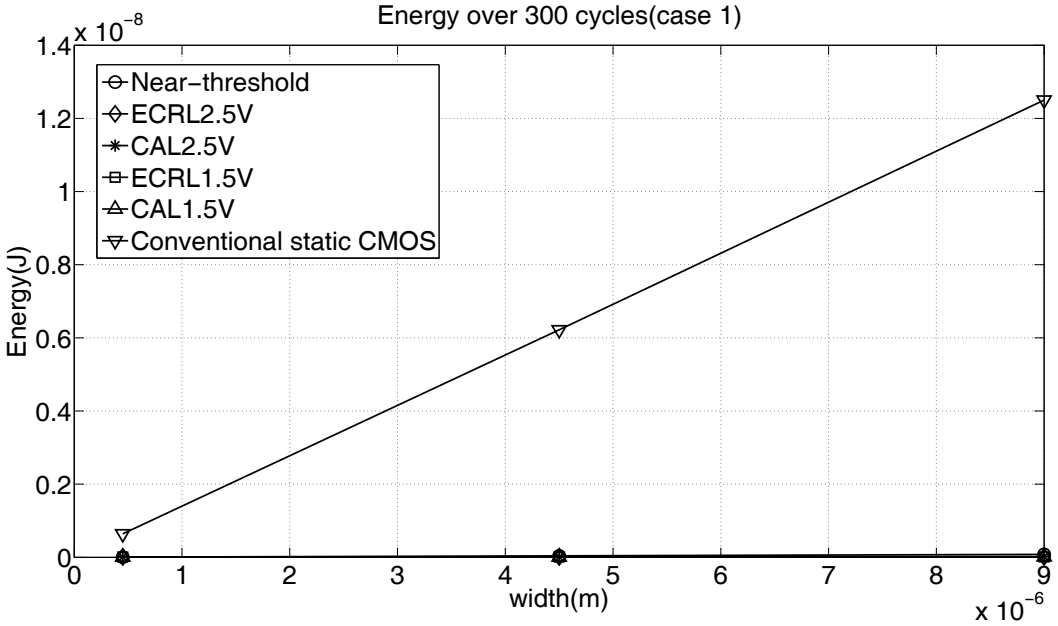

Figure 3.8: Comparison of energy consumption over 300 cycles in case 1.

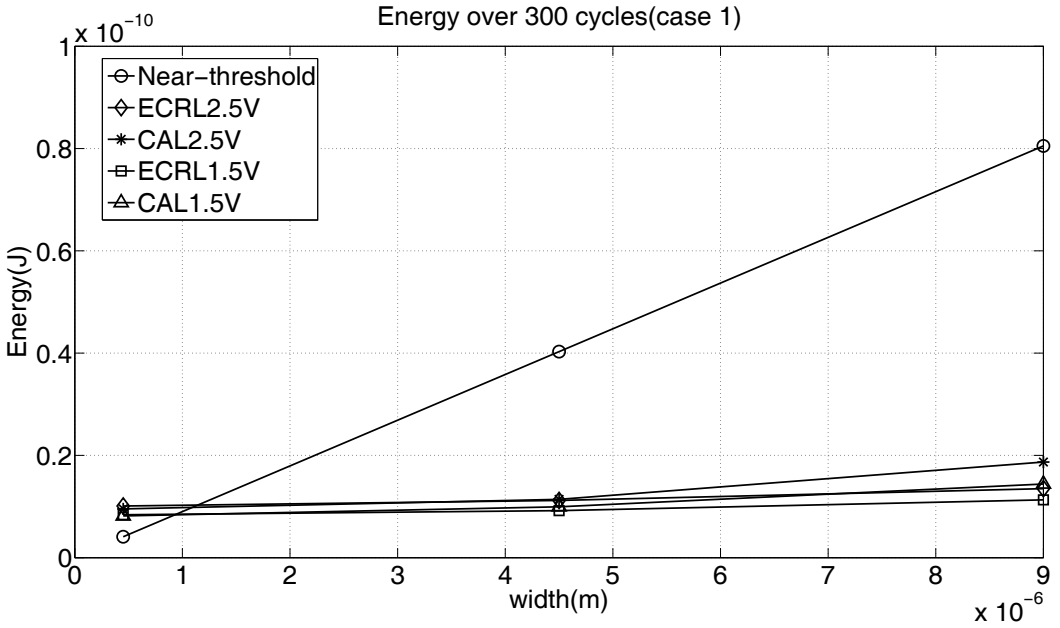

Figure 3.9: Comparison of energy consumption over 300 cycles in case 1 (conventional static CMOS is removed).

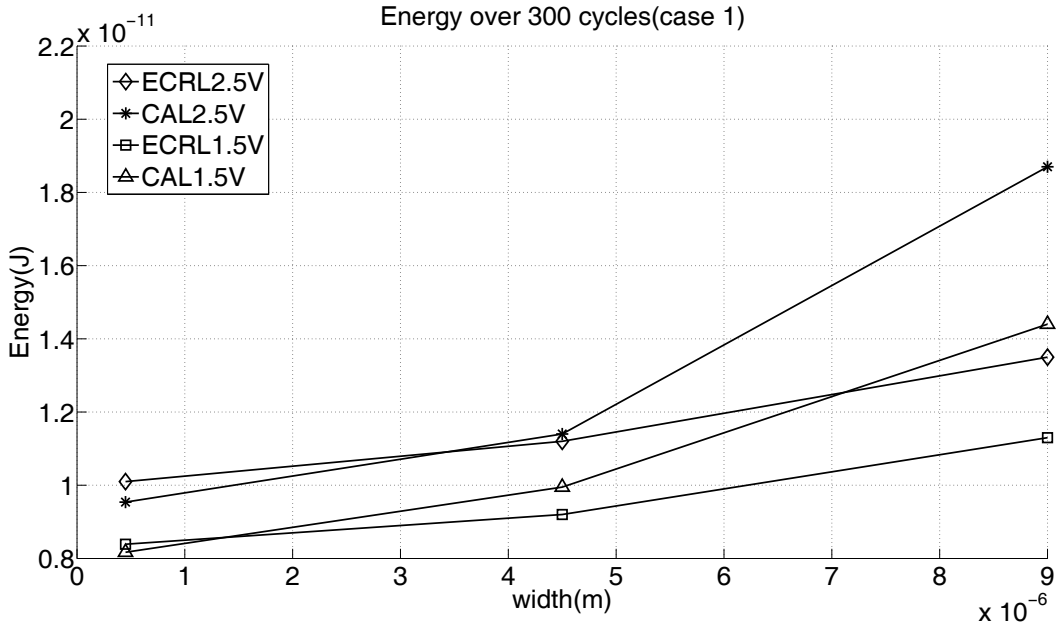

Figure 3.10: Comparison of energy consumption over 300 cycles in case 1 (near-threshold logic is removed).

ure 3.9 and Figure 3.10. Both near-threshold and charge recovery circuits have significant energy reduction compared to conventional static CMOS logic. Near-threshold logic saves the most energy with a 450 *nm* transistor width. However, with increasing device size, the advantage of charge recovery circuits becomes dominant. Figure 3.10 shows that both charge recovery circuits do not achieve a considerable reduction by scaling the supply voltage due to the operating regime of the charge recovery circuits in this case.

3.2.3 Case 2 of charge recovery circuits

- **Energy per switching**

In case 2, shown in Figure 3.11, both CAL and ECRL logic have an increase in energy consumption and energy is no longer a fraction of $\frac{1}{2}C_{out}V_{th}^2$ since in this case, data transitions when the power-clock signal is at $\frac{1}{2}V_{dd}$ instead of zero. In other words, transistor turns on when the voltage across it is not zero. Despite this increase, conventional static CMOS logic inverter still has greater energy consumption.

Figure 3.12 continues the comparison after removing conventional static CMOS inverter. In case 2, voltage scaling makes a greater contribution in reducing energy dissipation of charge recovery circuits, as illustrated in Figure 3.12. For example, at 4.5 *um*, CAL with 2.5 V supply voltage consumes more than 2 pJ, which is more than 10 times of the energy consumption after voltage scaling, which is approximately 0.2 pJ. Near-threshold logic consumes approximately 20 fJ, which is the least energy consumption among these low power circuits.

- **Delay**

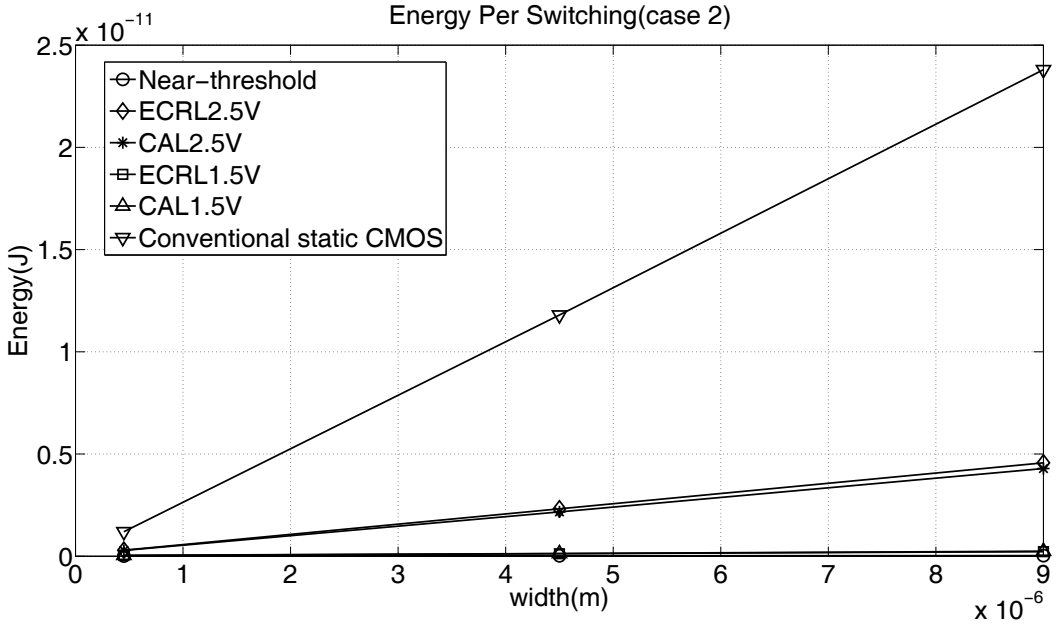

Figure 3.11: Energy per switching comparison in case 2.

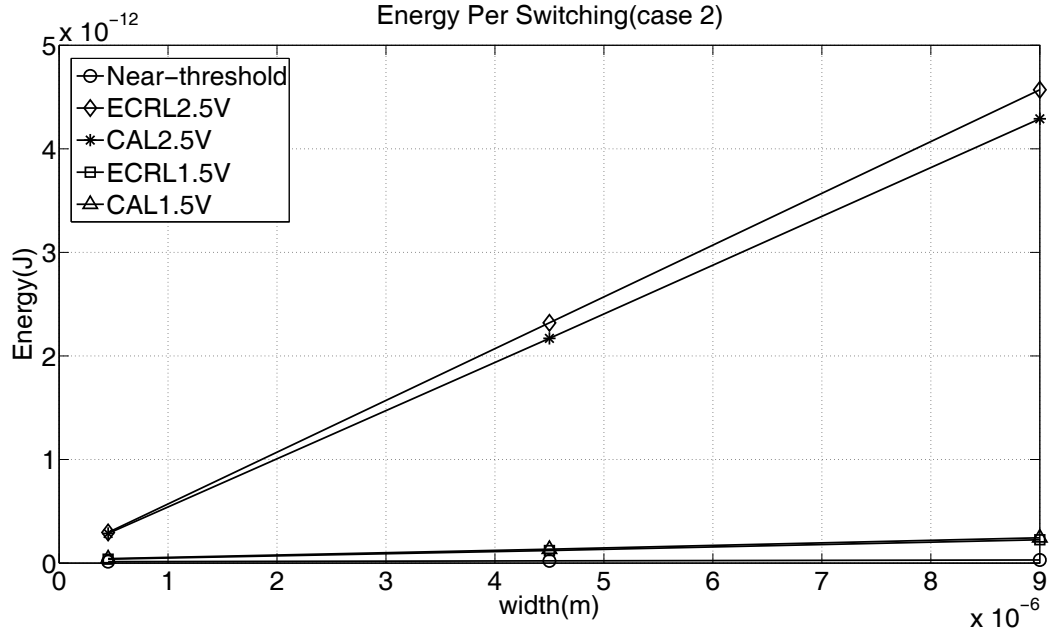

Figure 3.12: Energy per switching comparison in case 2 (conventional static CMOS is removed).

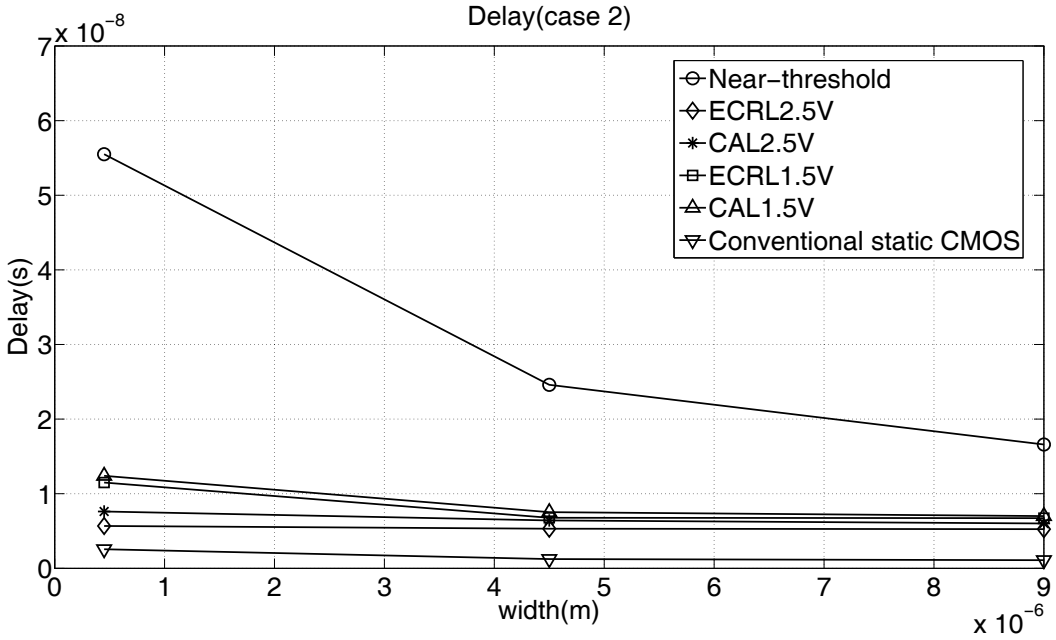

Figure 3.13: Delay comparison when charge recovery circuits are in case 2.

In case 2, when transistor turns on or off, voltage across it is not zero, so energy consumption increases, as indicated above. However, with a higher turn-on voltage, the speed of a transistor also increases. Figure 3.13 demonstrates this characteristic. While near-threshold logic remains slow, delay for both ECRL and CAL are significantly less than near-threshold logic. For example, at 450 *nm* width, delay for near-threshold logic is approximately 56 *ns*, while delay for both charge recovery circuits is smaller than 10 *ns*.

With the graph of near-threshold logic removed, Figure 3.14 focuses on charge recovery circuits. Even though conventional static CMOS logic still operates faster, charge-recovery inverters are operating faster than before, where the delay has decreased by approximately 90%. Also, voltage scaling has an affect on speed. Delay of charge-recovery inverters with voltage scaling is higher. The difference becomes clearer as the transistor width is reduced. At 450 *nm*, the delay rises by around 60%.

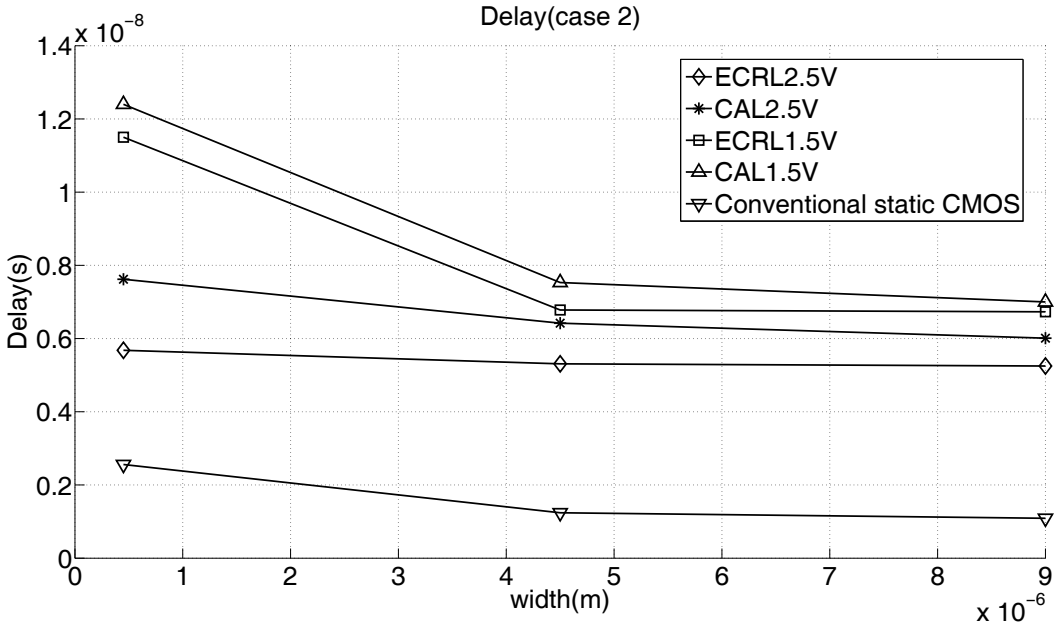

Figure 3.14: Delay comparison when charge recovery circuits are in case 2 (near-threshold logic is removed).

- **Energy delay product**

Figure 3.15 shows the EDP in case 2. EDP of both charge-recovery inverters (operating at full voltage) is comparable to conventional static CMOS logic. Alternatively, if voltage is scaled for charge recovery circuits, EDP is not only an order of magnitude less than before, but also lower than near-threshold logic. This result demonstrates that voltage scaling reduces the energy consumption of charge recovery circuits while not significantly affecting speed. Note that this characteristic is true only in case 2.

- **Energy over 300 cycles**

When energy per switching is analyzed in case 2, unlike case 1, transistor no longer turns on with approximately zero voltage difference across it, so the circuit consumes more energy. Over 300 cycles, the situation is similar. Figure 3.16

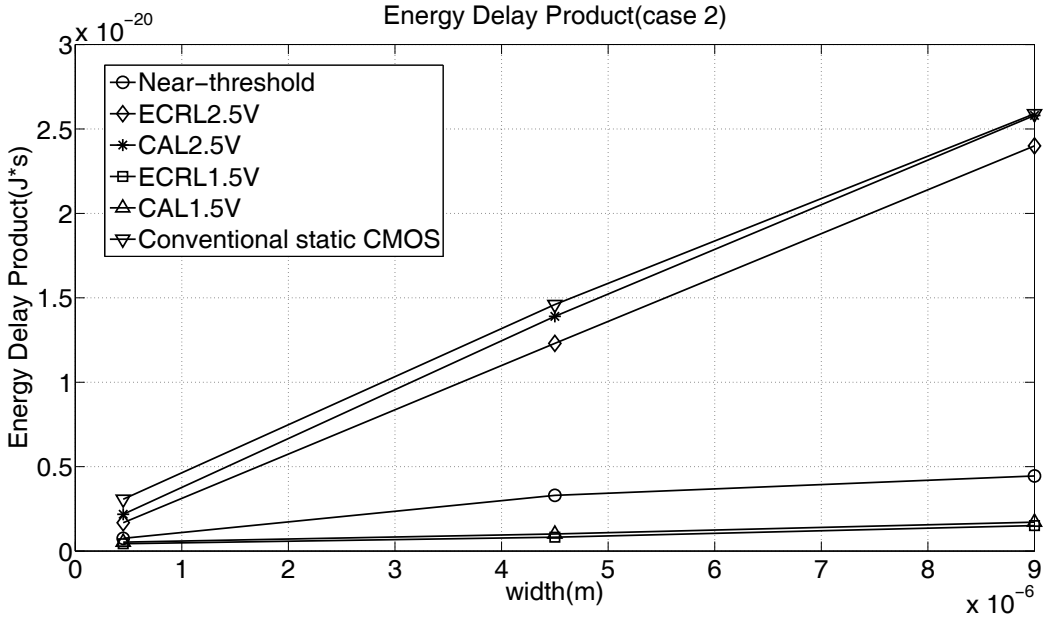

Figure 3.15: Energy delay product comparison in case 2.

presents these data for all of the inverters. Conventional static CMOS inverter consumes the most energy. Both charge recovery inverters consume more energy (approximately 10 times more) than the near-threshold logic.

After removing the graph of conventional static CMOS inverter, a clearer view about the influence of voltage scaling is illustrated in Figure 3.17. By scaling the peak voltage of the power-clock signal to 1.5 V, unlike case 1, charge recovery circuits achieve approximately 10X reduction in energy dissipation.

3.2.4 Case 3 of charge recovery circuits

- **Energy per switching**

In case 3, data transitions when power-clock signal is at V_{dd}. Energy consumption for charge recovery circuits exceed conventional static CMOS logic. Only by scaling its supply voltage to 1.5 V, power can be smaller, yet the number is still

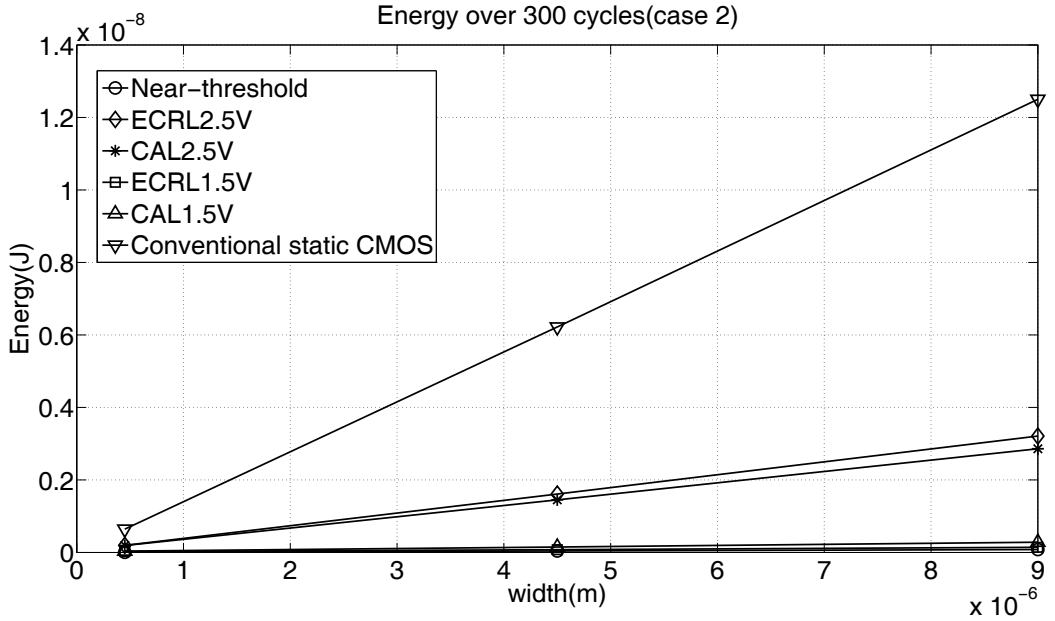

Figure 3.16: Comparison of energy over 300 cycles in case 2.

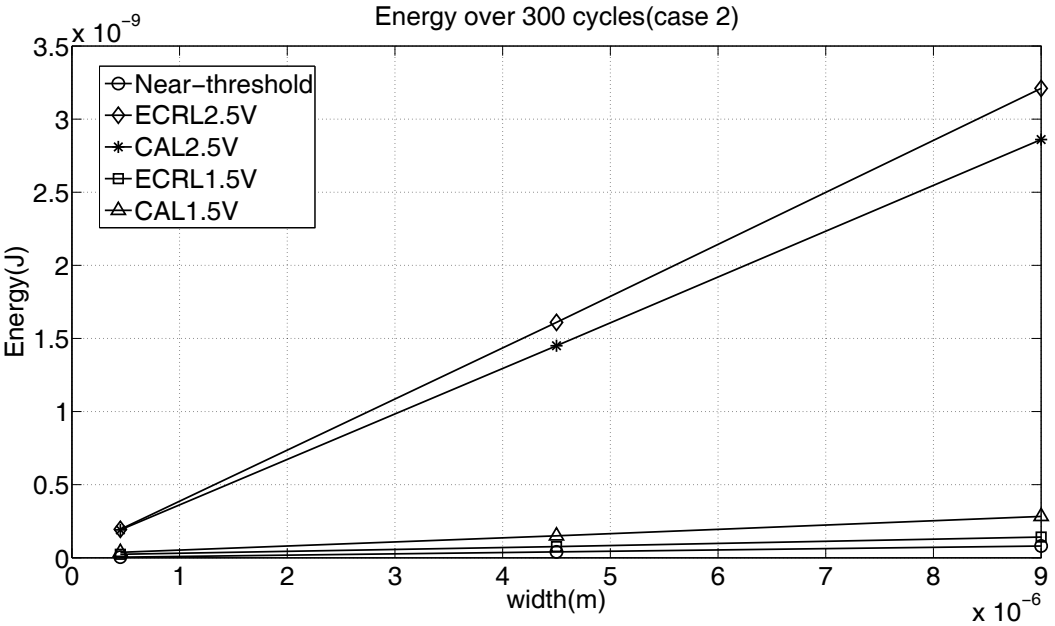

Figure 3.17: Comparison of energy over 300 cycles in case 2 (conventional static CMOS is removed).

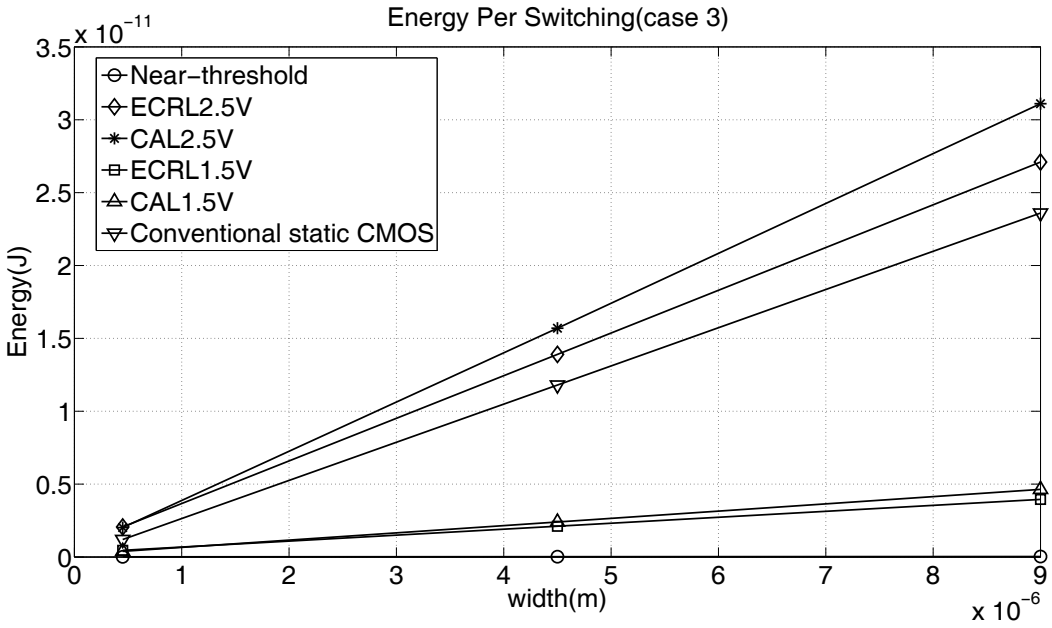

Figure 3.18: Energy per switching comparison in case 3.

comparable with conventional static CMOS inverter. This situation is not desirable. Thus, in case 3, near-threshold logic is more advantageous than charge recovery circuits.

- **Delay**

Figure 3.19 and Figure 3.20 provide the delay characteristics in case 3. As Figure 3.19 shows, near-threshold logic is the slowest among all circuits. Speed of charge recovery circuits is comparable to conventional static CMOS logic.

Figure 3.20 excludes near-threshold logic. In charge recovery circuits, as the transistor turns on and off with a V_{dd} voltage difference across it (instead of zero in case 1), the large current that flows through the transistor causes the least delay of all cases.

- **Energy delay product**

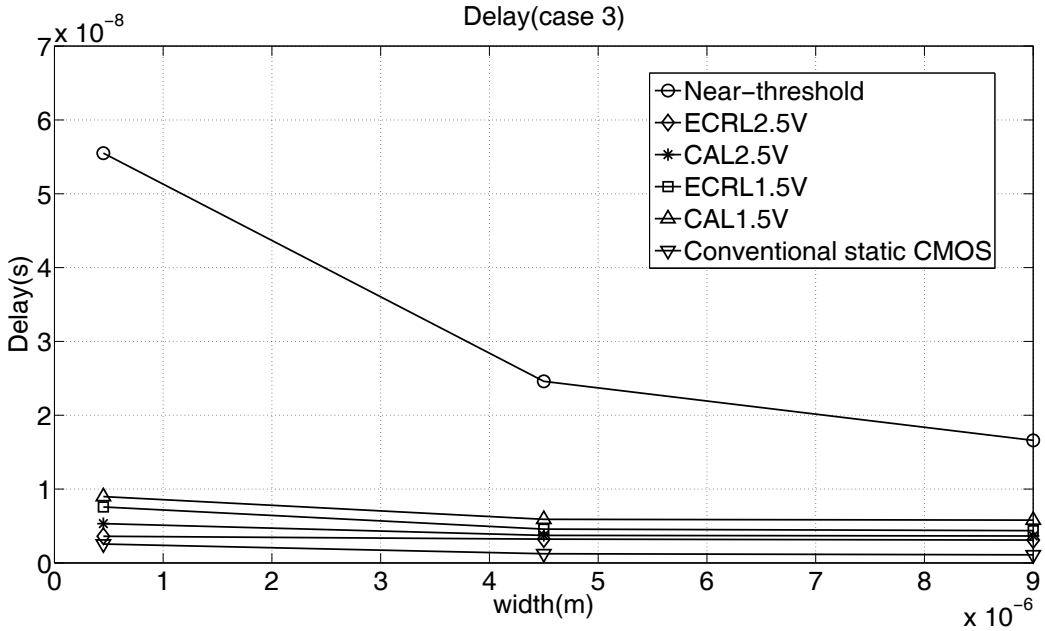

Figure 3.19: Delay comparison when charge recovery circuits are in case 3.

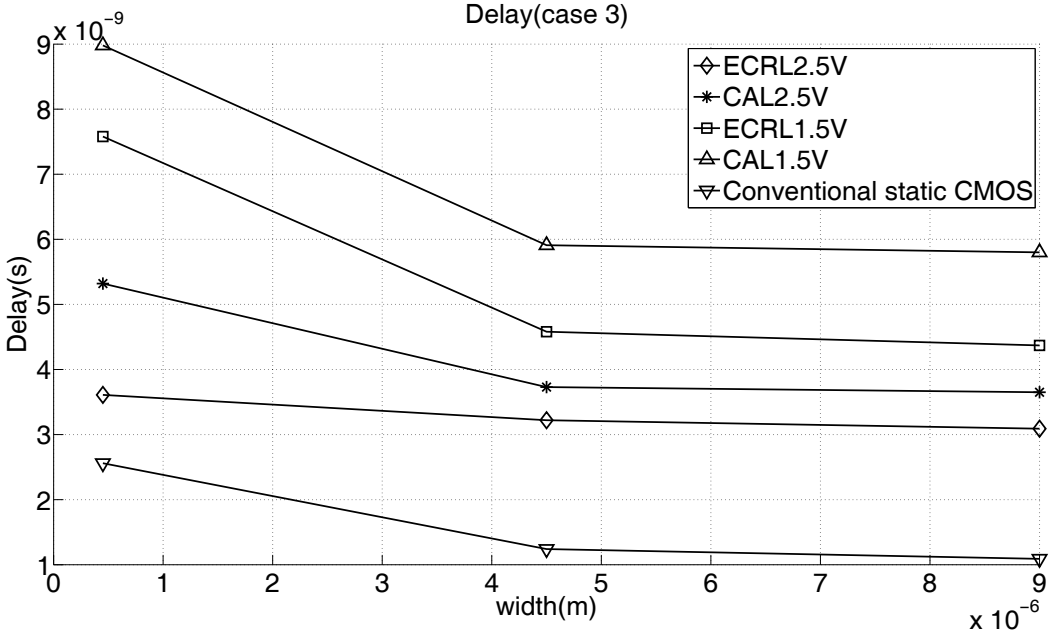

Figure 3.20: Delay comparison when charge recovery circuit are in case 3 (near-threshold logic is removed).

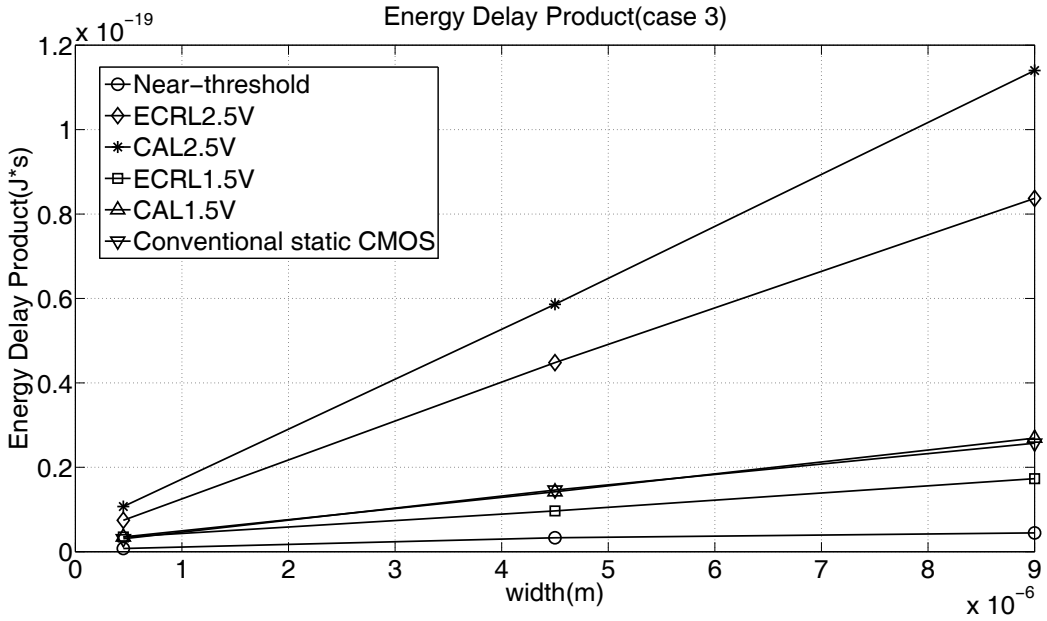

Figure 3.21: Energy delay product comparison in case 3.

Figure 3.21 shows the EDP in case 3. Energy delay product of charge recovery circuits is more than three times of conventional static logic. Thus, charge recovery circuits working in case 3 are not practical.

- **Energy over 300 cycles**

In case 3, energy consumption for charge recovery circuits become greater, as shown in Figure 3.22. Over 300 cycles, ECRL energy consumption exceeds conventional static CMOS logic. Thus, in this case, charge recovery circuits no longer improve energy efficiency.

Similar to case 2, voltage scaling in this case achieves approximately 80% reduction in energy consumption over 300 cycles in charge recovery circuits.

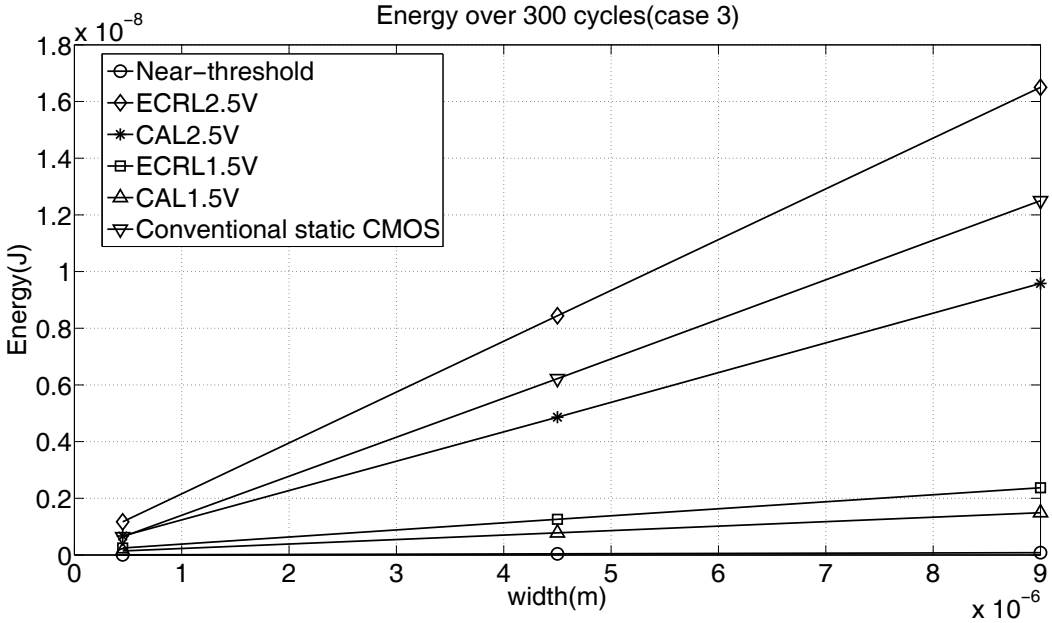

Figure 3.22: Comparison of energy consumption over 300 cycles in case 3.

Chapter 4

Conclusion

Two existing low power circuit design techniques, near-threshold logic and charge recovery circuits, are considered. Near-threshold computing is preferable to sub-threshold computing for a wide range of applications since speed penalty for sub-threshold logic is significant. Also, two kinds of charge recovery circuit families, ECRL and CAL, are considered.

A comprehensive comparison has been achieved among conventional static CMOS logic, near-threshold logic, and two kinds of charge recovery circuits. For charge recovery circuits, in addition to the traditional timing characteristic (case 1), two other timing relationships are investigated.

Five design criteria have been considered in this comparison: Maximum operating frequency, energy per switching, delay, energy delay product, and energy over 300 cycles. Results are provided in Section 3.2. Table 3.1 demonstrates that charge recovery circuits have a maximum operating frequency that is significantly greater than near-threshold logic.

In case 1, all of the low-power circuits consume significantly less energy than conventional static CMOS logic. With the increase of transistor width, the ad-

vantage of charge recovery circuits becomes more dominant. In case 2, due to a different mode of operation, the energy consumption of charge recovery circuits increases, and even exceeds conventional static CMOS logic in case 3. Delay of charge recovery circuits in case 1 is fixed to approximately $\frac{1}{4f}$, which makes charge recovery circuits the slowest at small widths. However, when the operation mode changes to case 2 and 3, both charge recovery circuits operate faster. Near-threshold logic still has advantage on energy delay product in case 1. While in case 2, even though charge recovery circuits with regular supply voltage has an EDP close to static CMOS logic, with scaled voltage, EDP drops below near-threshold logic. Case 3 is not suitable for charge recovery circuits due to its high EDP, even with voltage scaling.

Charge recovery circuits have the ability to conserve significant energy, while achieving a higher maximum operating frequency than near-threshold logic. Also, with a different operation mode (case 2), charge recovery circuits have the potential to achieve both low energy consumption and relatively high speed, as demonstrated by the lowest energy delay product.

Bibliography

[1] E. Salman and E. G. Friedman, *High Performance Integrated Circuit Design*. McGraw-Hill, 2012.

[2] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 2010.

[3] R. Brown *et al.*, "Report to congress on server and data center energy efficiency: Public law 109-431," 2008.

[4] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, Vol. 91, No. 2, pp. 305–327, 2003.

[5] R. Dreslinski, *et al.*, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 253-266, 2010.

[6] C. H. Bennett and R. Landauer, "The fundamental physical limits of computation," *Scientific American*, Vol. 253, No. 1, pp. 48–56, 1985.

[7] P. Teichmann, *Adiabatic logic: future trend and system level perspective*, 2011.

[8] J. Denker, "A review of adiabatic computing," *IEEE Symposium in Low Power Electronics*, pp. 94-97, 1994.

[9] A. Dickinson and J. Denker, "Adiabatic dynamic logic," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 282-285, 1994.

[10] B. Gojman, "Adiabatic Logic," California Institute of Technology, Tech. Rep., 2004.

[11] T. J. Hogan and G. Edwards, "Simulation of Energy Dissipation for Adiabatic Switching of CMOS based Reversible Logic Circuit," University of Bolton, Tech. Rep., 2010.

[12] Y. Ye and K. Roy, "Energy recovery circuits using reversible and partially reversible logic," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 43, No. 9, pp. 769-778, 1996.

[13] Y. Ye and K. Roy, "Low-power circuit design using adiabatic switching principle," *Proceedings of the Midwest Symposium on Circuits and Systems*, pp. 1189-1192, 1995.

[14] A. Markovic *et al.*, "Ultralow-Power Design in Near-Threshold Region," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 237-252, 2010.

[15] F. Chang *et al.*, "Practical Strategies for Power-Efficient Computing Technologies," *Proceedings of the IEEE*, Vol. 98, No. 2, pp. 215-236, 2010.

[16] C. J. Vieri, "Pendulum: A reversible computer architecture," Ph.D. Dissertation, Massachusetts Institute of Technology, 1995.

[17] S. Kim, C. Ziesler, and M. Papaefthymiou, "Charge-recovery computing on silicon," *IEEE Transactions on Computers*, Vol. 54, No. 6, pp. 651-659, 2005.

[18] S. Kim and M. Papaefthymiou, "True single-phase adiabatic circuitry," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 9, No. 1, pp. 52-63, 2001.

[19] J. Koller and W. Athas, "Adiabatic Switching, Low Energy Computing, And The Physics Of Storing And Erasing Information," *Workshop on Physics and Computation*, pp. 267-270, 1992.

[20] A. Kramer *et al.*, "Adiabatic computing with the 2N-2N2D logic family," *Symposium on VLSI Circuits*, pp. 25–26, 1994.

[21] A. Kramer, J. Denker, B. Flower, and J. Moroney, "2nd order adiabatic computation with 2N-2P and 2N-2N2P logic circuits," *Proceedings of the International Symposium on Low Power Design*, pp. 191–196, 1995.

[22] A. Vetuli, S. Pascoli, and L. Reyneri, "Positive feedback in adiabatic logic," *Electronics Letters*, Vol. 32, No. 20, pp. 1867-1869, 1996.

[23] A. Blotti, S. Di Pascoli, and R. Saletti, "Simple model for positive-feedback adiabatic logic power consumption estimation," *Electronics Letters*, Vol. 36, No. 2, pp. 116-118, 2000.

[24] S. Li *et al.*, "Quasi-static energy recovery logic with single power-clock supply," *IEEE International Symposium on Circuits and Systems*, pp. 2124–2127, 2007.

[25] V. G. Oklobdzija, D. Maksimovic, and F. Lin, "Pass-transistor adiabatic logic using single power-clock supply," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,*, Vol. 44, No. 10, pp. 842-846, 1997.

[26] 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, 1996.

[27] D. Maksimovic, V. Oklobdzija, B. Nikolic, and K. Wayne current, "Clocked CMOS adiabatic logic with integrated single-phase power-clock supply: experimental results," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 323-327, 1997.

[28] D. Maksimovic and V. G. Oklobdzija, "Clocked CMOS adiabatic logic with single AC power supply," *IEEE Solid-State Circuits Conference*, pp. 370–373, 1995.

[29] J. Fischer, P. Teichmann, and D. Schmitt-Landsiedel, "Scaling trends in adiabatic logic," *Proceedings of the 2nd ACM conference on Computing frontiers*, pp. 427–434, 2005.

[30] R. Yadav *et al.*, "Four phase clocking rule for energy efficient digital circuits: An adiabatic concept," *International Conference on Computer and Communication Technology (ICCCT)*, pp. 209-214, 2011.

Appendix A

Schematic figures

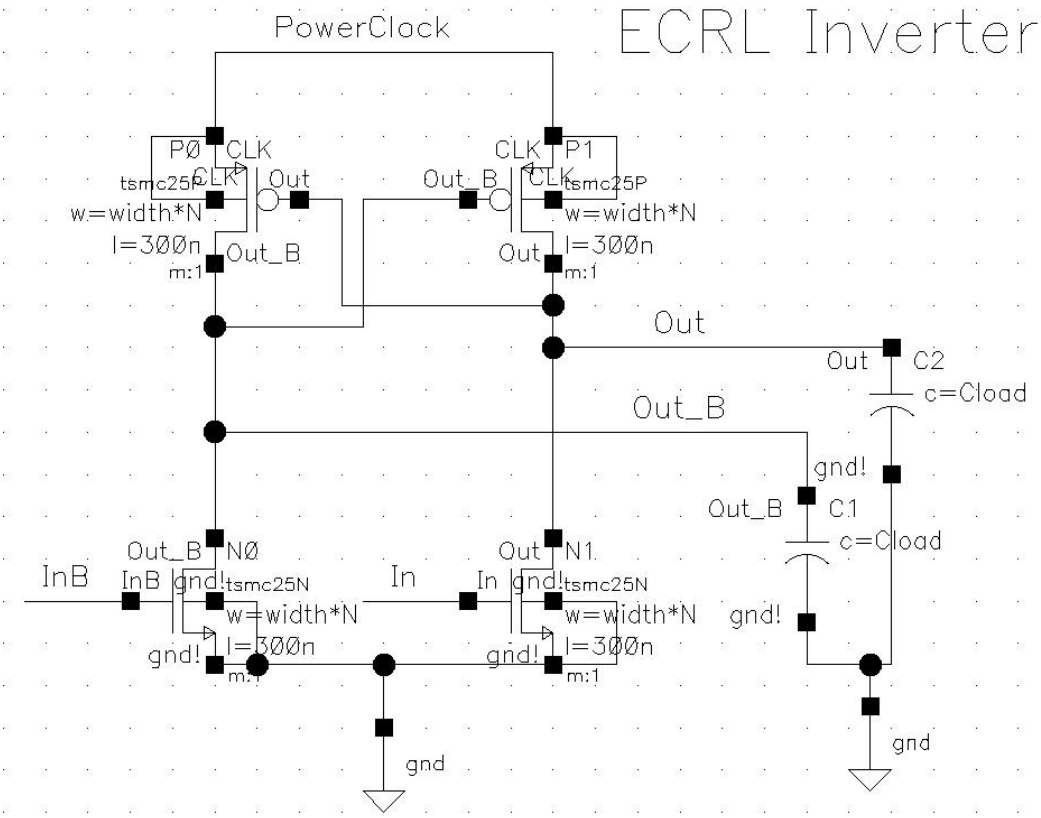

Figure A.1: ECRL Inverter schematic.

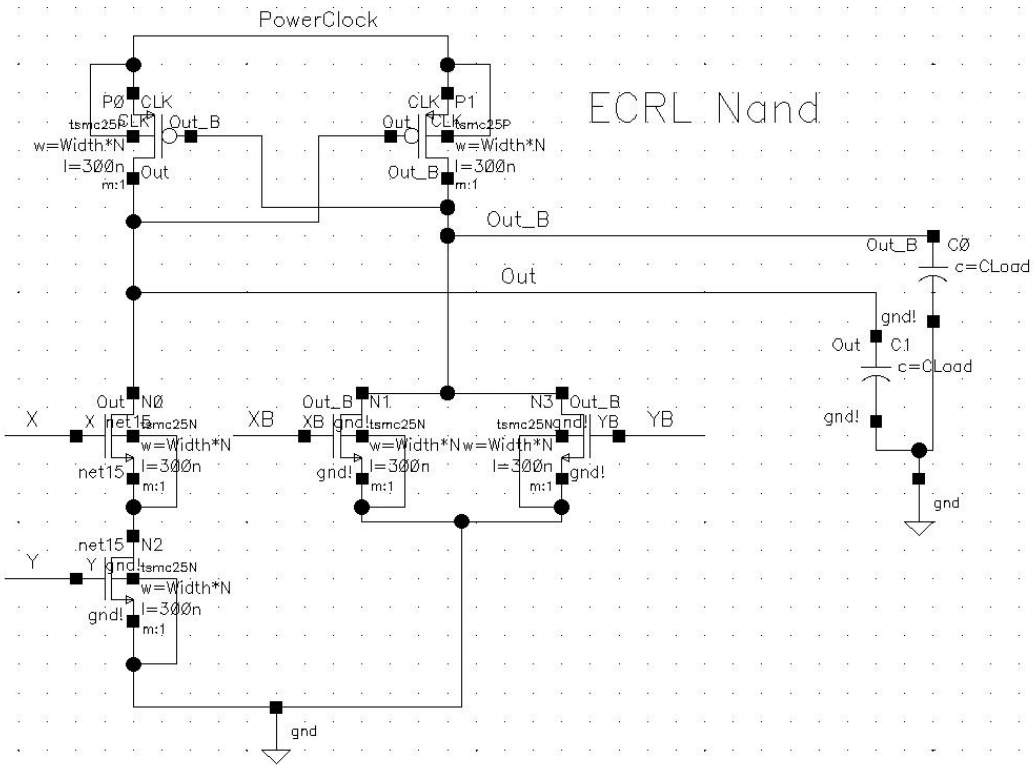

Figure A.2: ECRL NAND schematic.

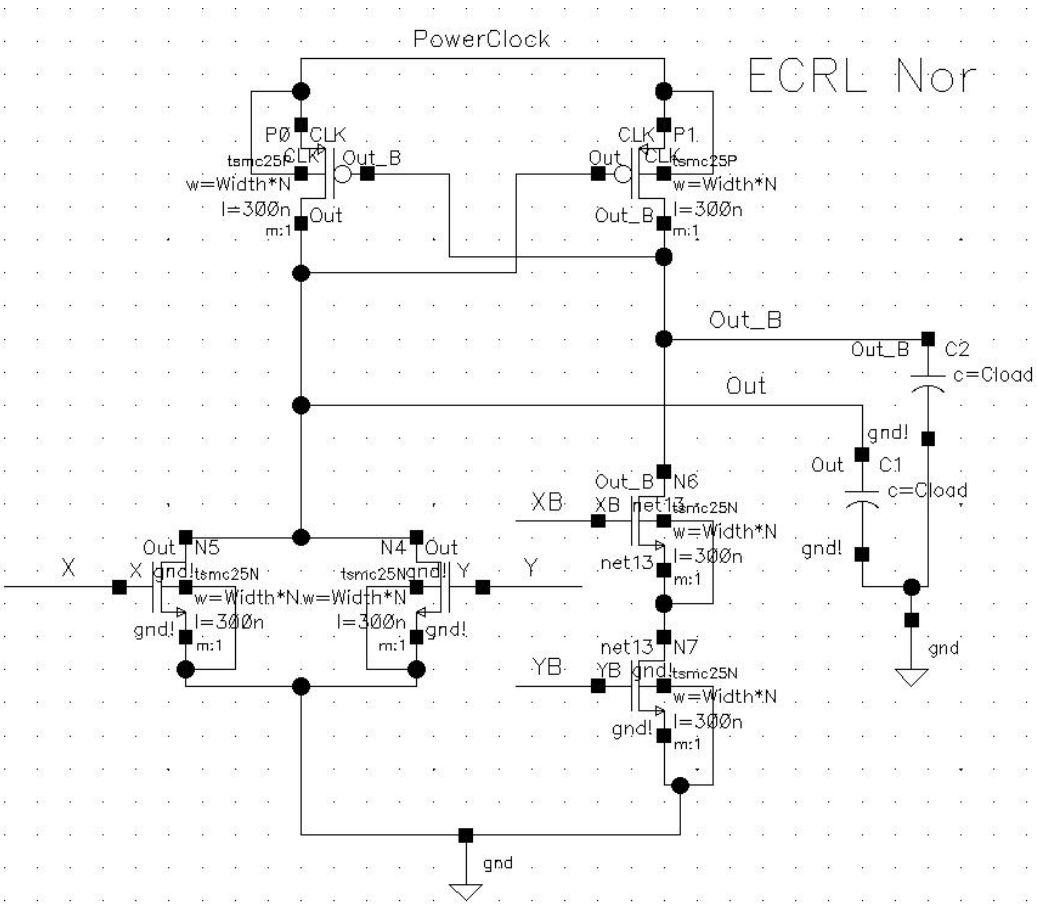

Figure A.3: ECRL NOR schematic.

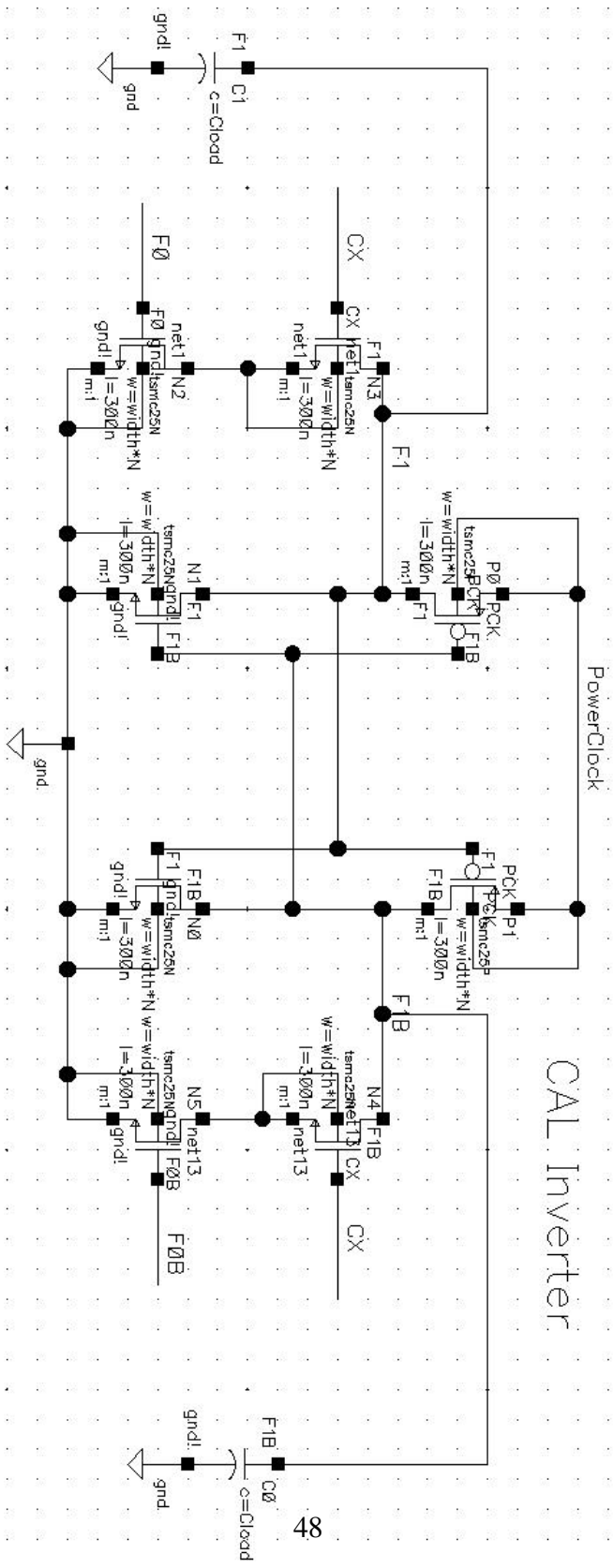

Figure A.4: CAL Inverter schematic.

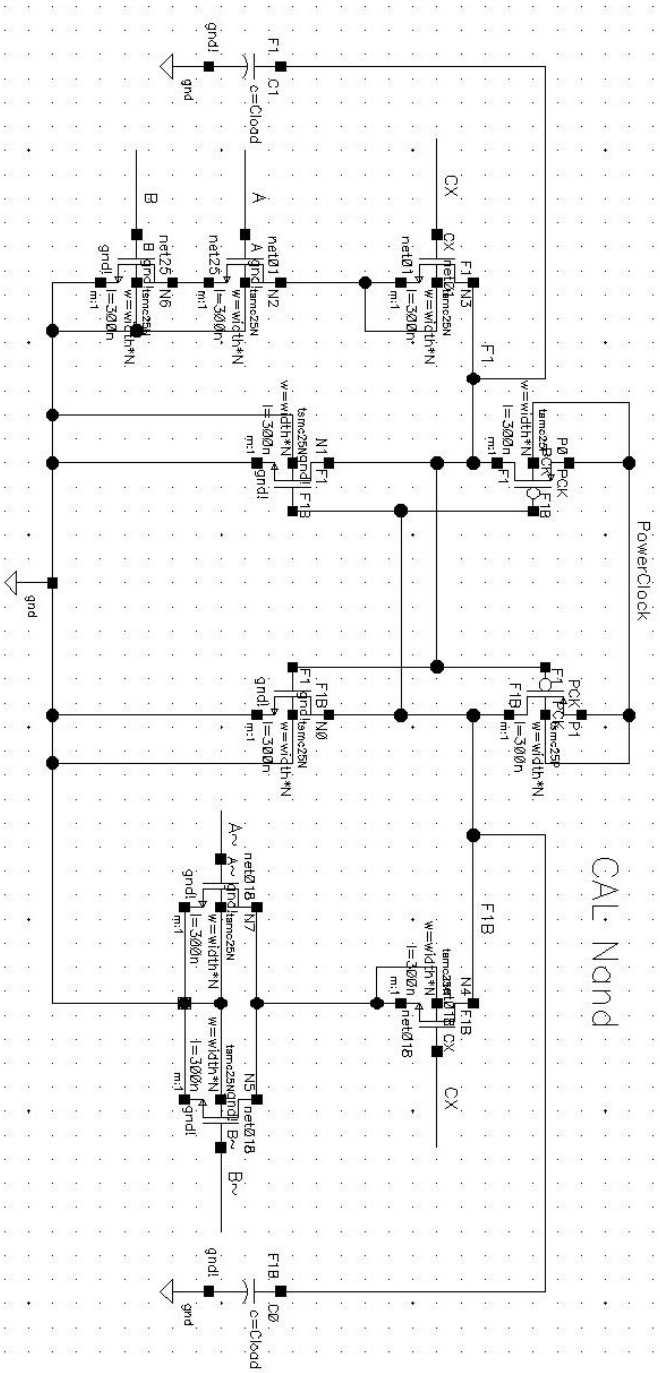

Figure A.5: CAL NAND schematic.

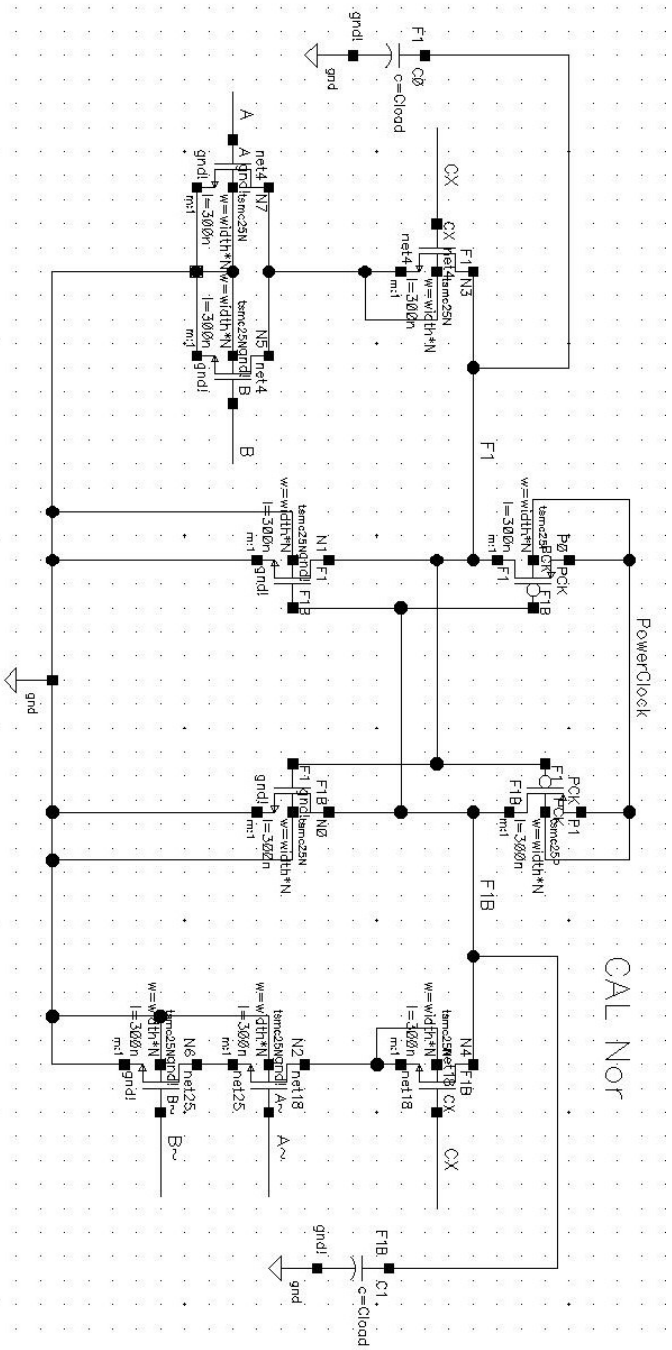

Figure A.6: CAL NOR schematic.

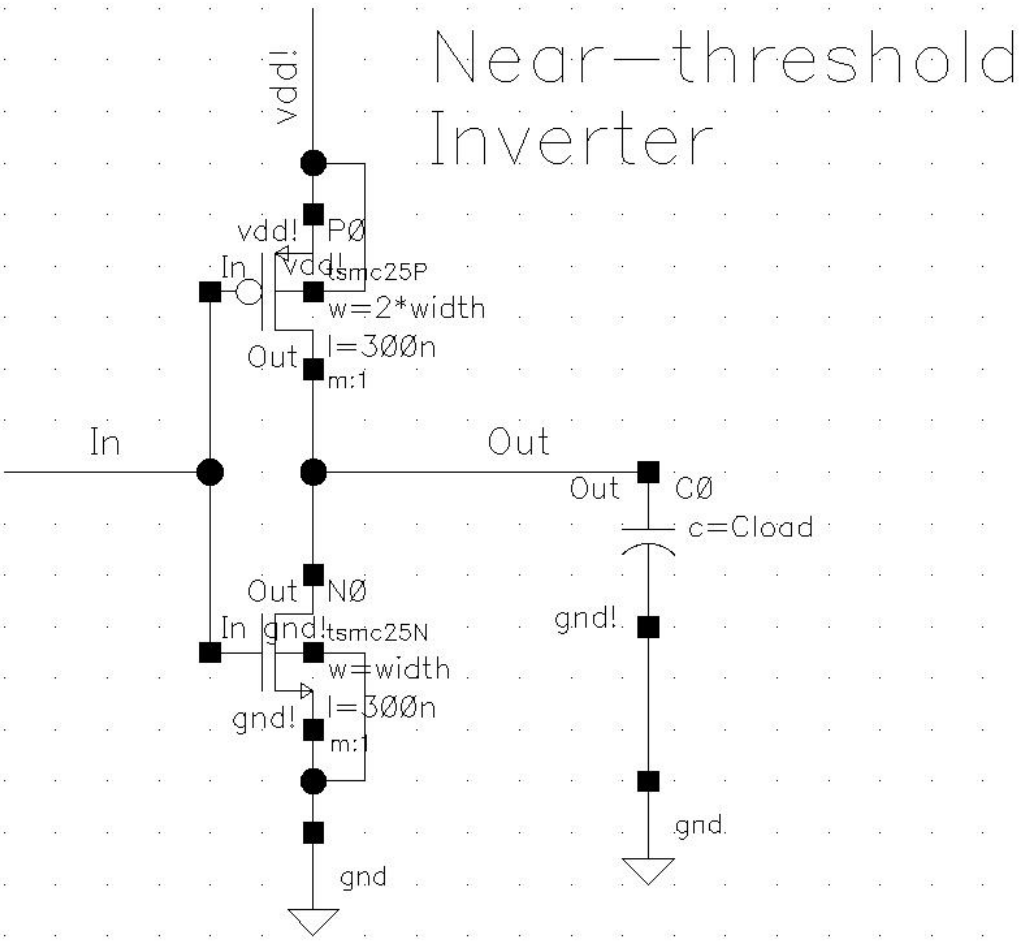

Figure A.7: Near-threshold Inverter schematic.

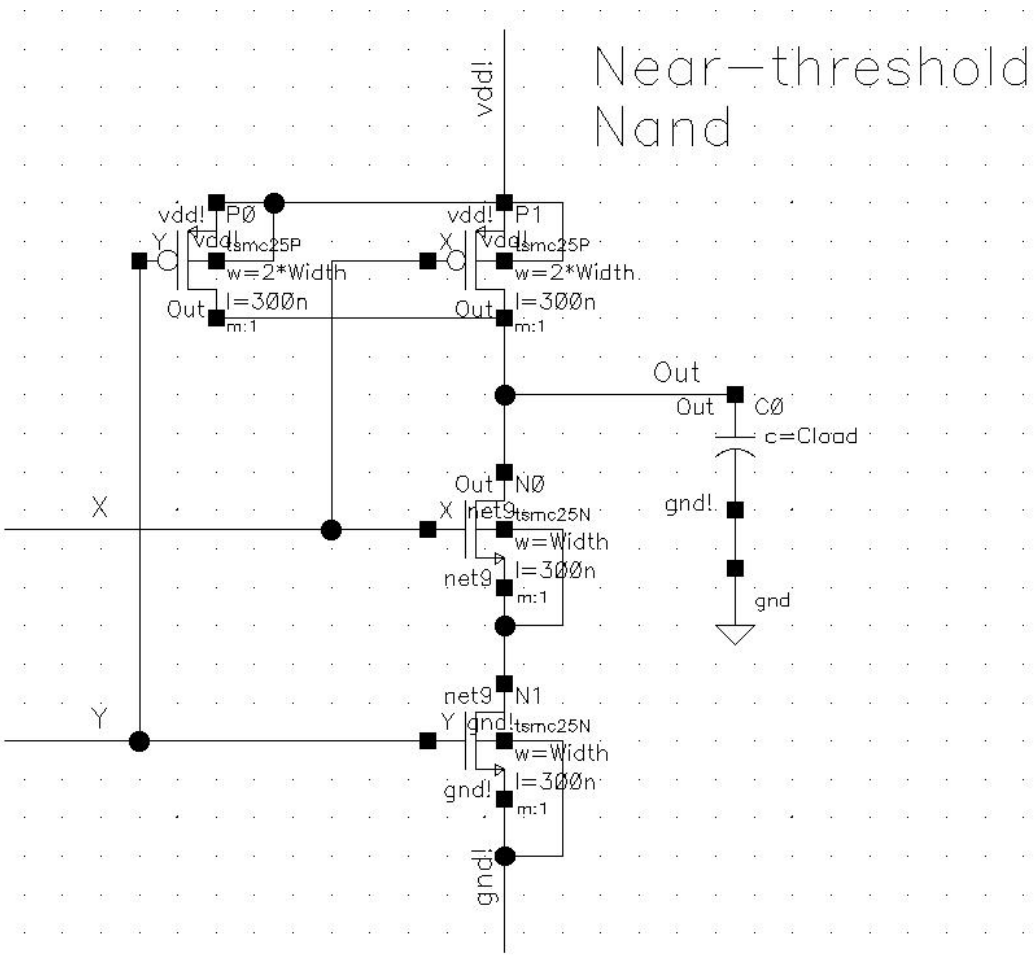

Figure A.8: Near-threshold NAND schematic.

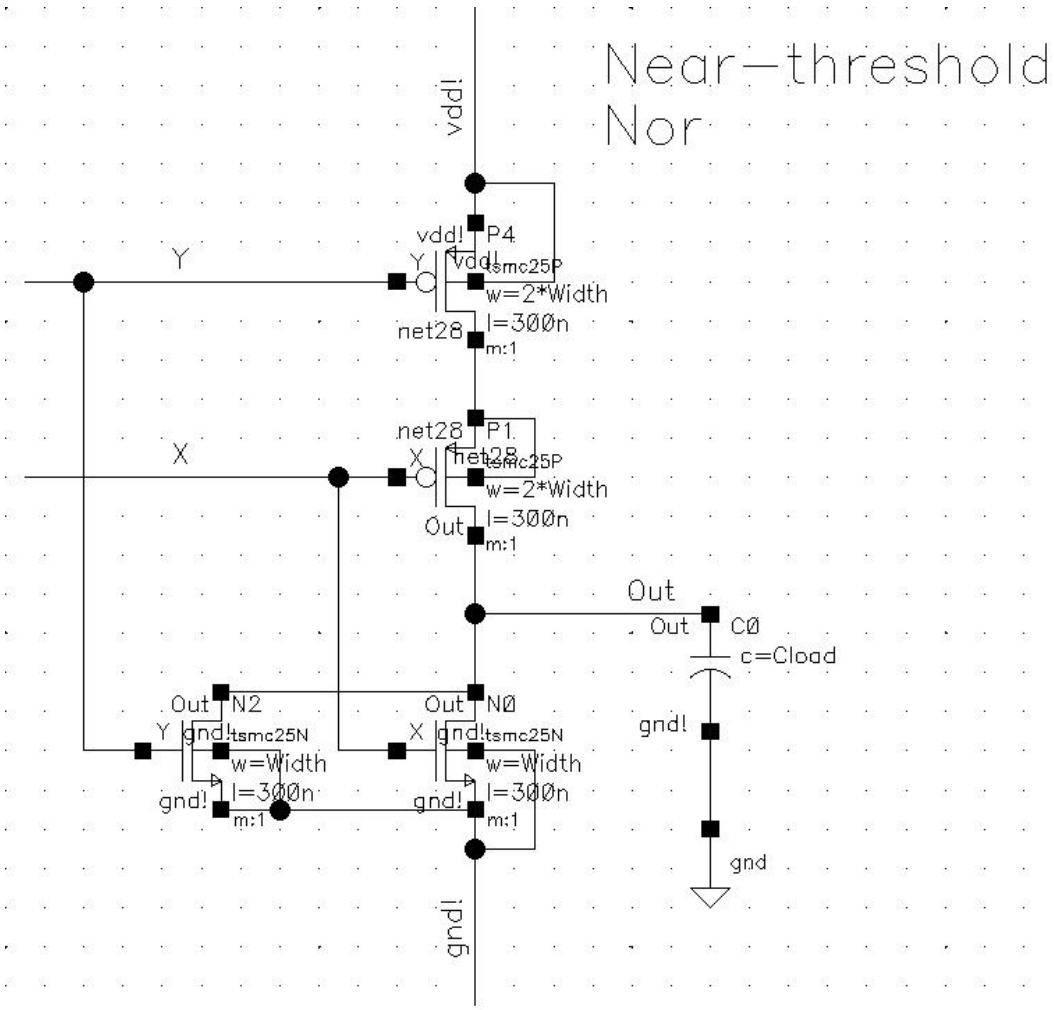

Figure A.9: Near-threshold NOR schematic.

Appendix B

Inverter waveform

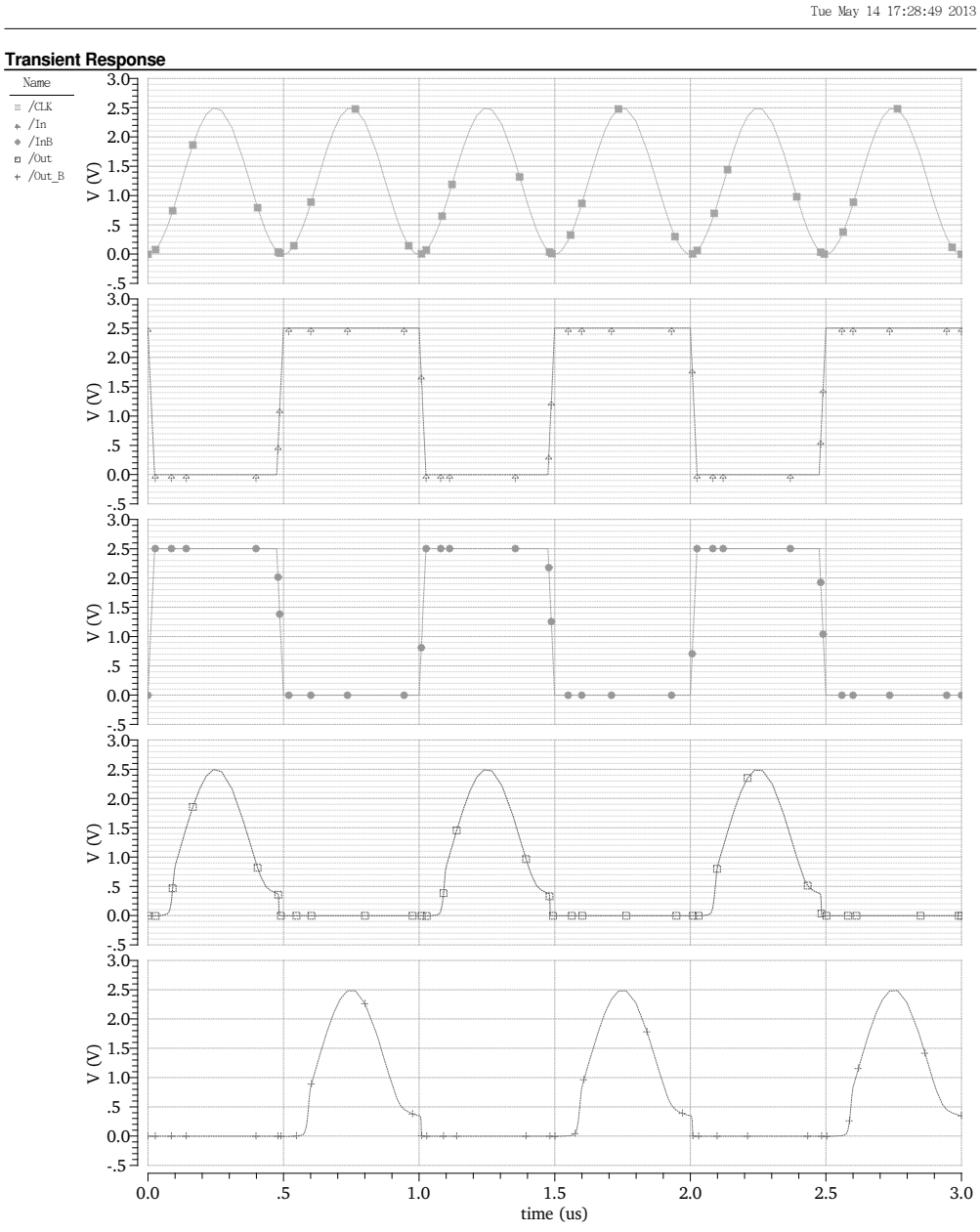

Figure B.1: ECRL Inverter waveform in Case 1.

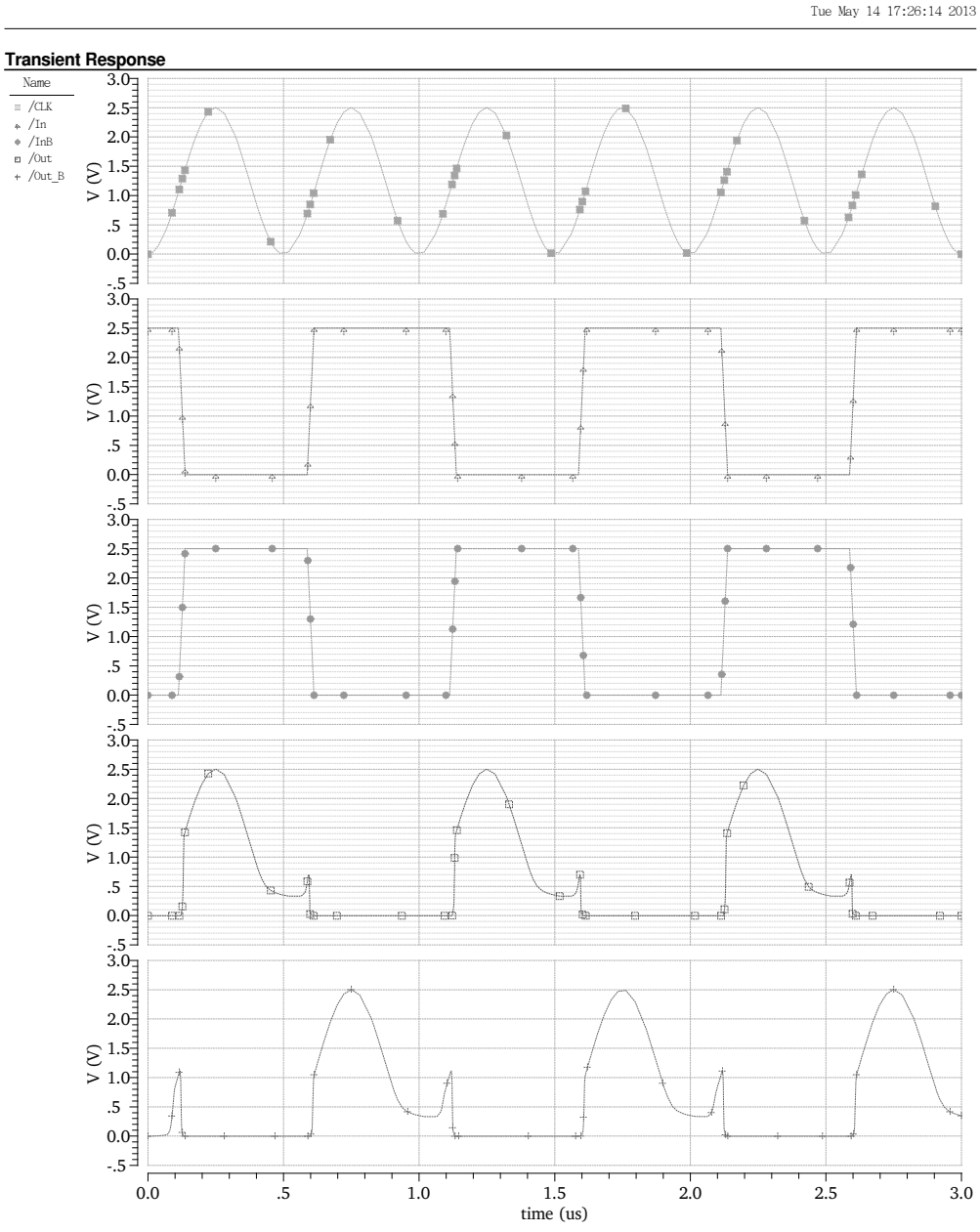

Figure B.2: ECRL Inverter waveform in Case 2.

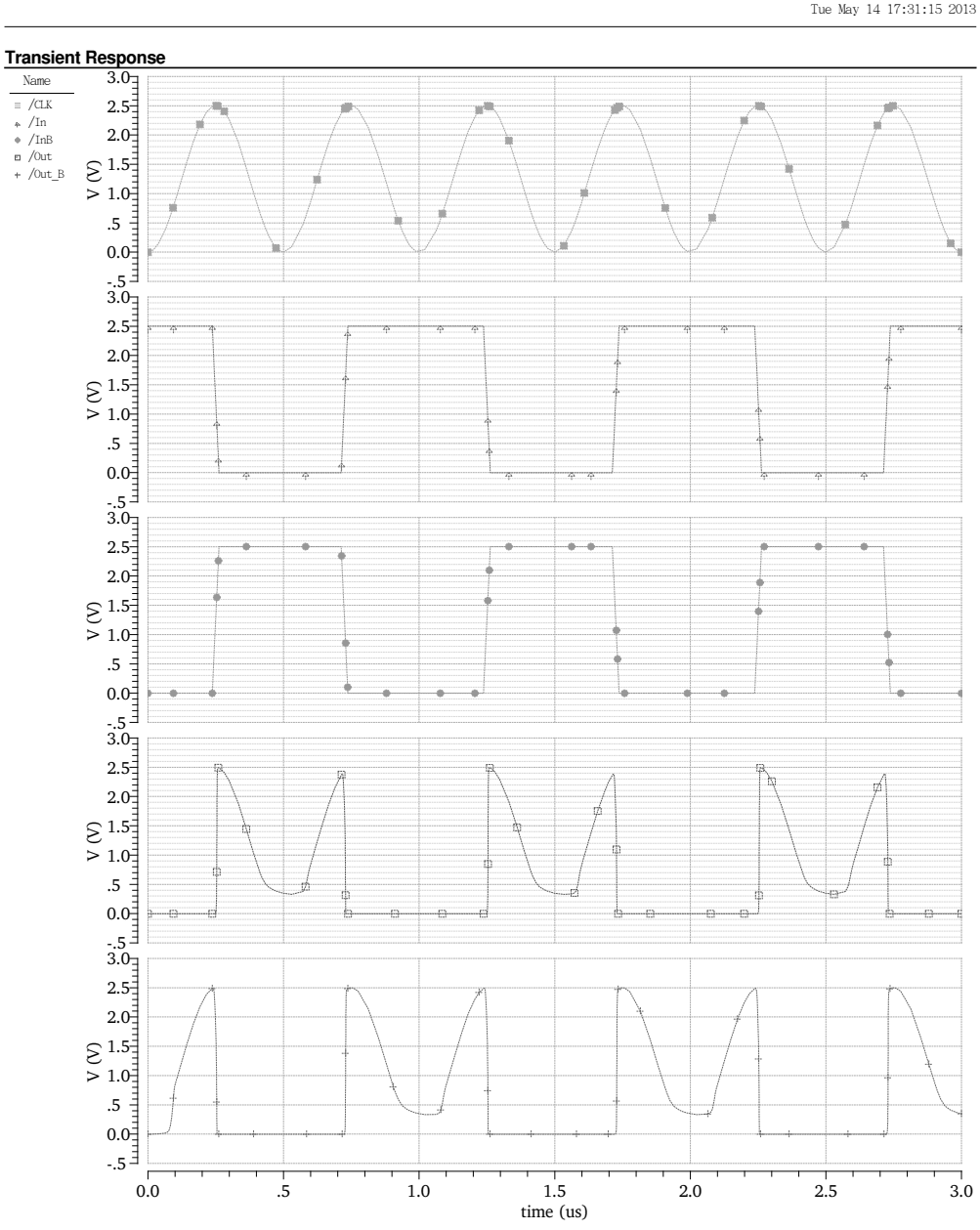

Figure B.3: ECRL Inverter waveform in Case 3.

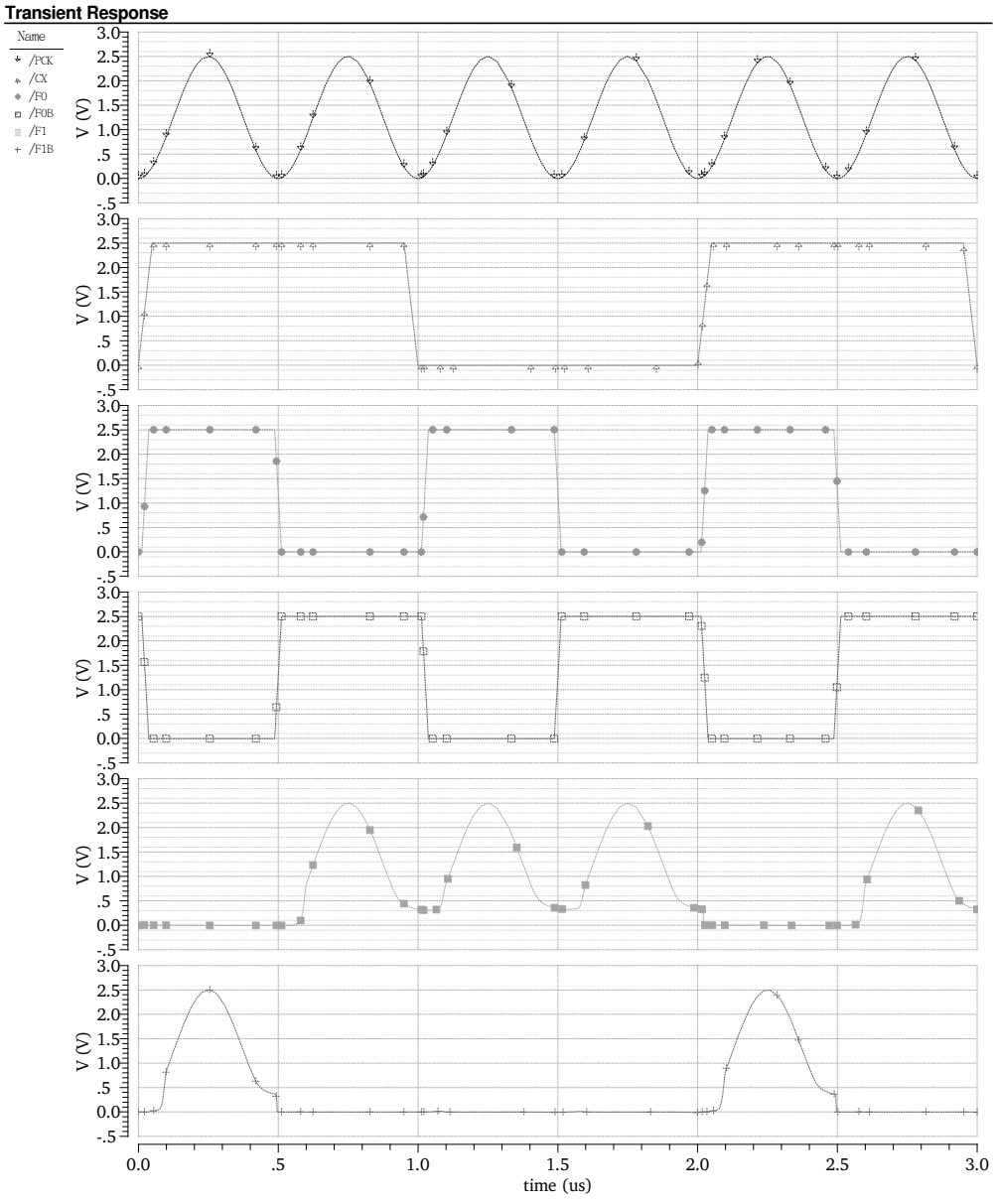

Figure B.4: CAL Inverter waveform in Case 1.

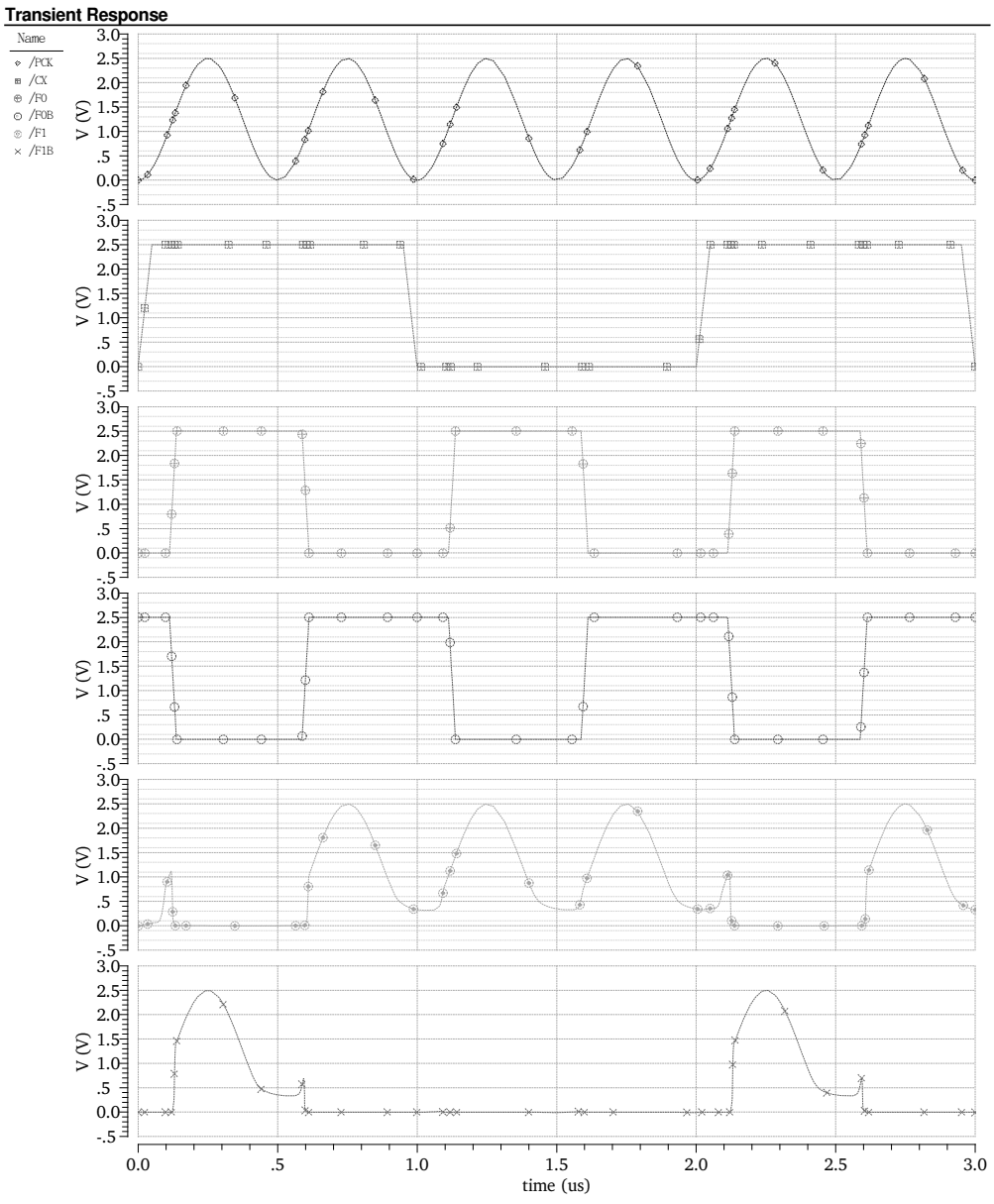

Figure B.5: CAL Inverter waveform in Case 2.

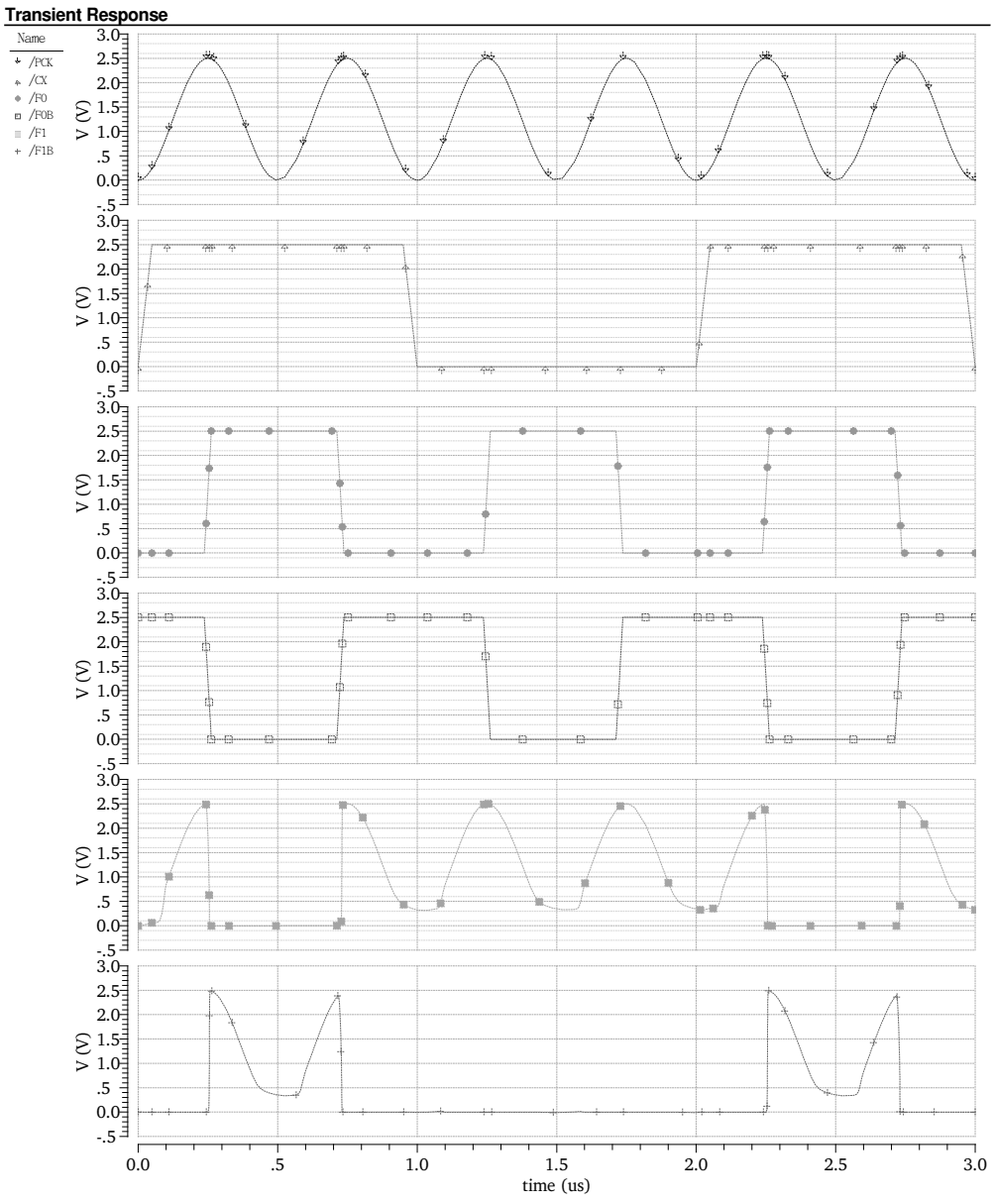

Figure B.6: CAL Inverter waveform in Case 3.

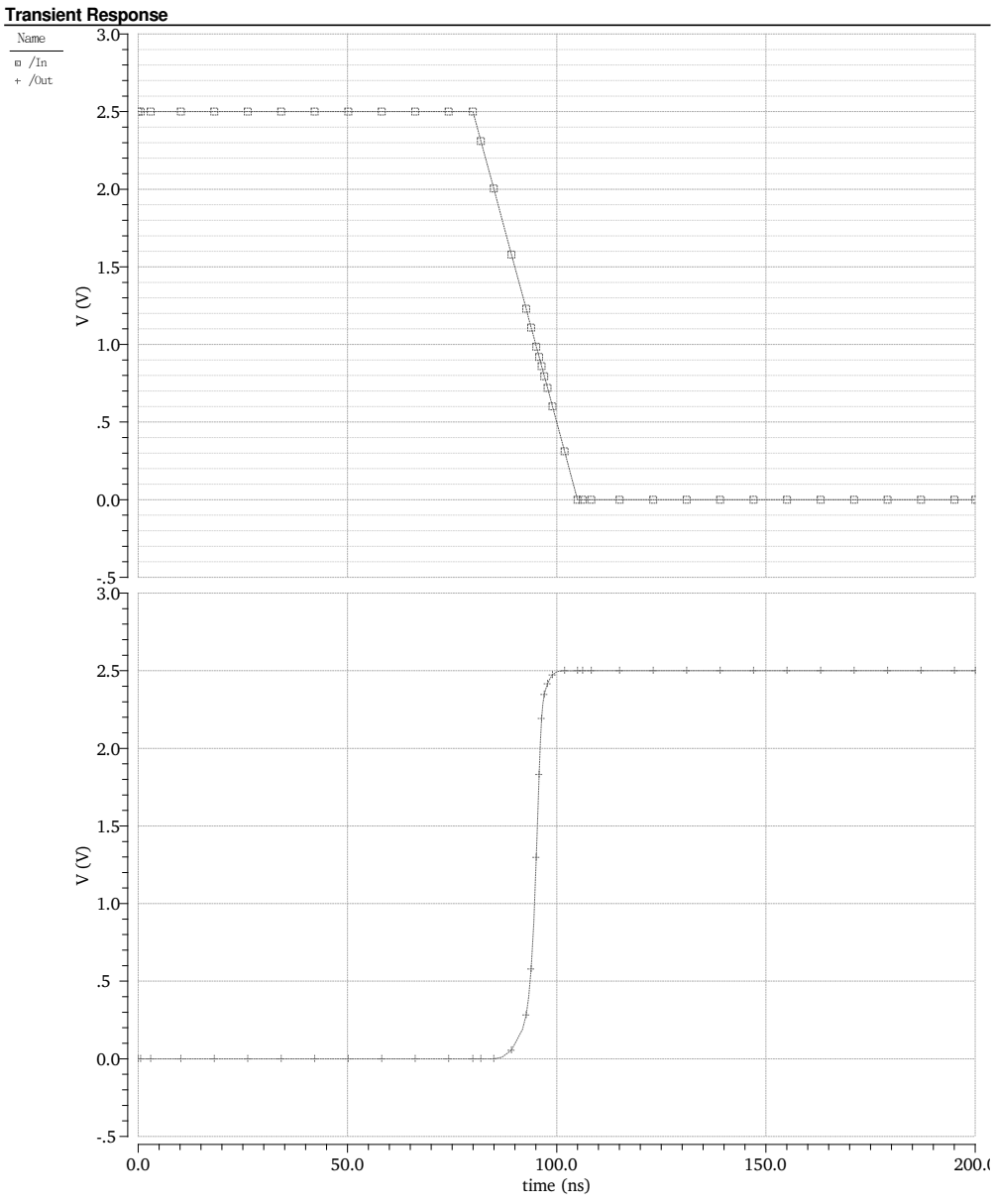

Figure B.7: Near-threshold Inverter waveform.

Appendix C

Transient results

Energy per switching (case 1) (J)	450nm	4.5μm	9.0μm
Near-threshold	13.6f	21.4f	30f
CAL 2.5V	24.2f	28.2f	31f
ECRL 2.5V	15.9f	16.4f	21.7f
CAL 1.5V	21f	24.8f	34.7f
ECRL 1.5V	14.4f	16.2f	18.9f
Conventional static CMOS	1.2p	11.8p	23.6p
Energy per switching (case 2) (J)	450nm	4.5μm	9.0μm
Near-threshold	13.6f	21.4f	30f
CAL 2.5V	288f	2.17p	4.31p
ECRL 2.5V	296f	2.31p	4.56p
CAL 1.5V	43.4f	135f	247f
ECRL 1.5V	37.8f	121f	225f
Conventional static CMOS	1.2p	11.8p	23.6p
Energy per switching (case 3) (J)	450nm	4.5μm	9.0μm
Near-threshold	13.6f	21.4f	30f
CAL 2.5V	2.11p	15.7p	31.1p
ECRL 2.5V	1.3p	8.34f	16.2p
CAL 1.5V	352f	2.16p	4.14p
ECRL 1.5V	336f	1.75p	3.26p
Conventional static CMOS	1.2p	11.8p	23.6p
Delay (case 1) (s)	450nm	4.5μm	9.0μm
Near-threshold	55.5n	24.6n	16.6n
CAL 2.5V	113n	113n	113n
ECRL 2.5V	113n	113n	113n
CAL 1.5V	118n	113n	113n
ECRL 1.5V	116n	113n	113n
Conventional static CMOS	2.56n	1.24n	1.09n
Delay (case 2) (s)	450nm	4.5μm	9.0μm
Near-threshold	55.5n	24.6n	16.6n
CAL 2.5V	7.62n	6.42n	6.01n
ECRL 2.5V	5.68n	5.31n	5.25n
CAL 1.5V	12.4n	7.53n	7.00n
ECRL 1.5V	11.5n	6.78nn	6.73n
Conventional static CMOS	2.56n	1.24n	1.09n
Delay (case 3) (s)	450nm	4.5μm	9.0μm
Near-threshold	55.5n	24.6n	16.6n
CAL 2.5V	5.32n	3.73n	3.65n
ECRL 2.5V	3.61n	3.22n	3.09n
CAL 1.5V	8.98n	5.91n	5.80n
ECRL 1.5V	7.58n	4.59n	4.37n
Conventional static CMOS	2.56n	1.24n	1.09n

Table C.1: Transient results of inverter (a)

Energy delay product (case 1) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.755a	0.526a	0.498a
CAL 2.5V	2.73a	3.18a	3.51a
ECRL 2.5V	1.8a	1.85a	2.45a
CAL 1.5V	2.48a	2.80a	3.92a
ECRL 1.5V	1.67a	1.83a	2.13a
Conventional static CMOS	3.07a	14.6a	25.7a
Energy delay product (case 2) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.755a	0.526a	0.498a
CAL 2.5V	2.19a	13.9a	25.9a
ECRL 2.5V	1.68a	12.3a	23.9a
CAL 1.5V	0.538a	1.01a	1.73a
ECRL 1.5V	0.435a	0.822a	1.51a
Conventional static CMOS	3.07a	14.6a	25.7a
Energy delay product (case 3) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.755a	0.526a	0.498a
CAL 2.5V	11.2a	58.5a	114a
ECRL 2.5V	4.69a	26.9a	49.9a
CAL 1.5V	3.16a	12.8a	24.0a
ECRL 1.5V	2.55a	8.01a	14.2a
Conventional static CMOS	3.07a	14.6a	25.7a
Energy over 300 cycles (case 1) (J)	450nm	4.5μm	9.0μm
Near-threshold	4.10p	40.3p	80.5p
CAL 2.5V	9.54p	11.4p	18.7p
ECRL 2.5V	10.1p	11.2p	13.5p
CAL 1.5V	8.17p	9.95p	14.4p
ECRL 1.5V	8.39p	9.20p	11.3p
Conventional static CMOS	642p	6.22n	12.5n
Energy over 300 cycles (case 2) (J)			
Near-threshold	4.10p	40.3p	80.5p
CAL 2.5V	191p	1.45n	2.89n
ECRL 2.5V	195p	1.61n	3.21n
CAL 1.5V	37.2p	150p	283p
ECRL 1.5V	23.8p	76.7p	142p
Conventional static CMOS	642p	6.22n	12.5n
Energy over 300 cycles (case 3) (J)			
Near-threshold	4.10p	40.3p	80.5p
CAL 2.5V	668p	4.86n	9.58n
ECRL 2.5V	1.17n	8.84n	16.5n
CAL 1.5V	141p	778p	1.49n
ECRL 1.5V	245p	1.26n	2.37n
Conventional static CMOS	642p	6.22n	12.5n

Table C.2: Transient results of inverter (b)

Energy per switching (case 1) (J)	450nm	4.5μm	9.0μm
Near-threshold	13.7f	23.0f	33.2f
CAL 2.5V	24.7f	29.6f	43.4f
ECRL 2.5V	21.6f	23f	25f
CAL 1.5V	21.1f	25.9f	36.4f
ECRL 1.5V	19.1f	19.7f	23.7f
conventional static CMOS	1.32p	12.7p	25.7p
Energy per switching (case 2) (J)	450nm	4.5μm	9.0μm
Near-threshold	13.7f	23.0f	33.2f
CAL 2.5V	522f	4.28p	8.5p
ECRL 2.5V	358f	4.72p	9.43p
CAL 1.5V	52f	245f	468f
ECRL 1.5V	46f	157f	298f
Conventional static CMOS	1.32p	12.7p	25.7p
Energy per switching (case 3) (J)	450nm	4.5μm	9.0μm
Near-threshold	13.7f	23.0f	33.2f
CAL 2.5V	2.52p	18.3p	36.2p
ECRL 2.5V	1.52p	10.4p	20.1p
CAL 1.5V	576f	3.85p	7.55p
ECRL 1.5V	421f	2.58p	4.92p
Conventional static CMOS	1.32p	12.7p	25.7p
Delay (case 1) (s)	450nm	4.5μm	9.0μm
Near-threshold	58.1n	20.2n	17.4n
CAL 2.5V	113n	113n	113n
ECRL 2.5V	113n	113n	113n
CAL 1.5V	118n	113n	113n
ECRL 1.5V	116n	113n	113n
Conventional static CMOS	2.96n	2.05n	1.85n
Delay (case 2)(s)	450nm	4.5μm	9.0μm
Near-threshold	58.1n	20.2n	17.4n
CAL 2.5V	11.7n	10.5n	10.5n
ECRL 2.5V	8.47n	6.55n	6.50n
CAL 1.5V	13.9n	8.05n	7.60n
ECRL 1.5V	13n	7.75n	7.21n
Conventional static CMOS	2.96n	2.05n	1.85n
Delay (case 3) (s)	450nm	4.5μm	9.0μm
Near-threshold	58.1n	20.2n	17.4n
CAL 2.5V	8.33n	6.40n	6.34n
ECRL 2.5V	7.23n	5.81n	5.68n
CAL 1.5V	11.1n	7.56n	7.39n
ECRL 1.5V	11.5n	6.56n	6.31n
Conventional static CMOS	2.96n	2.05n	1.85n

Table C.3: Transient results of NAND (a)

Energy delay product (case 1) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.796a	0.466a	0.579a
CAL 2.5V	2.80a	3.33a	4.88a
ECRL 2.5V	2.45a	2.59a	2.81a
CAL 1.5V	2.48a	2.92a	4.10a
ECRL 1.5V	2.22a	2.22a	2.68a
Conventional static CMOS	3.92a	26.1a	47.5a
Energy delay product (case 2) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.796a	0.466a	0.579a
CAL 2.5V	6.08a	45.0a	89.2a
ECRL 2.5V	3.03a	30.9a	61.3a
CAL 1.5V	0.723a	4.03a	7.60a
ECRL 1.5V	0.6a	1.21a	2.15a
Conventional static CMOS	3.92a	26.1a	47.5a
Energy delay product (case 3) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.796a	0.466a	0.579a
CAL 2.5V	21.0a	117a	229a
ECRL 2.5V	10.9a	60.1a	114a
CAL 1.5V	6.42a	29.1a	55.8a
ECRL 1.5V	4.85a	16.9a	31.1a
Conventional static CMOS	3.92a	26.1a	47.5a
Energy over 300 cycles (case 1) (J)	450nm	4.5μm	9.0μm
Near-threshold	4.27p	7.21p	10.4p
CAL 2.5V	11.6p	21.1p	25.8p
ECRL 2.5V	8.98p	10.4p	11.6p
CAL 1.5V	8.75p	12.1p	18.9p
ECRL 1.5V	8.39p	9.42p	9.80p
Conventional static CMOS	1.11n	10.2n	20.4n
Energy over 300 cycles (case 2) (J)			
Near-threshold	4.27p	7.21p	10.4p
CAL 2.5V	328p	2.71n	5.38n
ECRL 2.5V	209p	1.75n	3.5n
CAL 1.5V	48.1p	247p	478p
ECRL 1.5V	22.2p	93.1p	177p
Conventional static CMOS	1.11n	10.2n	20.4n
Energy over 300 cycles (case 3) (J)			
Near-threshold	4.27p	7.21p	10.4p
CAL 2.5V	753p	5.94n	11.8n
ECRL 2.5V	717p	5.31n	10.4n
CAL 1.5V	137p	854p	1.67p
ECRL 1.5V	143p	811p	1.55n
Conventional static CMOS	1.11n	10.2n	20.4n

Table C.4: Transient results of NAND (b)

Energy per switching (case 1) (J)	450nm	4.5μm	9.0μm
Near-threshold	14.0f	24.9f	36.8f
CAL 2.5V	23.6f	27.8f	37.7f
ECRL 2.5V	18.4f	22.0f	22.8f
CAL 1.5V	20.5f	24.9f	34.6f
ECRL 1.5V	18.7f	19.6f	23.5f
Conventional static CMOS	1.02p	7.87p	15.6p
Energy per switching (case 2) (J)	450nm	4.5μm	9.0μm
Near-threshold	14.0f	24.9f	36.8f
CAL 2.5V	545f	4.48p	8.9p
ECRL 2.5V	475f	4.01p	8.03p
CAL 1.5V	52.9f	240f	464f
ECRL 1.5V	56.7f	216f	408f
Conventional static CMOS	1.02p	7.87p	15.6p
Energy per switching (case 3) (J)	450nm	4.5μm	9.0μm
Near-threshold	14.0f	24.9f	36.8f
CAL 2.5V	3.79p	60.2p	59.9p
ECRL 2.5V	2.27p	14.0p	27.2p
CAL 1.5V	688f	4.6p	6.04p
ECRL 1.5V	426f	2.24p	4.24p
Conventional static CMOS	1.02p	7.87p	15.6p
Delay (case 1) (s)	450nm	4.5μm	9.0μm
Near-threshold	68.2n	27.0n	22.2n
CAL 2.5V	113n	113n	113n
ECRL 2.5V	113n	113n	113n
CAL 1.5V	117n	113n	113n
ECRL 1.5V	116n	113n	113n
Conventional static CMOS	3.92n	2.46n	2.29n
Delay (case 2) (s)	450nm	4.5μm	9.0μm
Near-threshold	68.2n	27.0n	22.2n
CAL 2.5V	11.8n	10.5n	10.5n
ECRL 2.5V	7.08n	5.81n	5.78n
CAL 1.5V	14.1n	7.89n	7.39n
ECRL 1.5V	13n	5.75n	5.21n
Conventional static CMOS	3.92n	2.46n	2.29n
Delay (case 3) (s)	450nm	4.5μm	9.0μm
Near-threshold	68.2n	27.0n	22.2n
CAL 2.5V	8.48n	6.58n	653n
ECRL 2.5V	4.31n	2.28n	2.12n
CAL 1.5V	11.3n	7.72n	7.56n
ECRL 1.5V	7.76n	4.86n	4.64n
Conventional static CMOS	3.92n	2.46n	2.29n

Table C.5: Transient results of NOR (a)

Energy delay product (case 1) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.956a	0.674a	0.819a
CAL 2.5V	2.67a	3.13a	4.24a
ECRL 2.5V	2.08a	2.47a	2.57a
CAL 1.5V	2.40a	2.81a	3.90a
ECRL 1.5V	2.18a	2.47a	2.65a
Conventional static CMOS	4.01a	19.3a	35.7a
Energy delay product (case 2) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.956a	0.674a	0.819a
CAL 2.5V	6.43a	47.1a	93.3a
ECRL 2.5V	3.36a	23.3a	46.4a
CAL 1.5V	0.748a	1.89a	3.43a
ECRL 1.5V	0.739a	1.24a	2.13a
Conventional static CMOS	4.01a	19.3a	35.7a
Energy delay product (case 3) ($J \cdot s$)	450nm	4.5μm	9.0μm
Near-threshold	0.956a	0.674a	0.819a
CAL 2.5V	32.1a	198a	391a
ECRL 2.5V	9.81a	31.9a	57.6a
CAL 1.5V	7.75a	35.5a	68.4a
ECRL 1.5V	3.31a	10.9a	19.7a
Conventional static CMOS	4.01a	19.3a	35.7a
Energy over 300 cycles (case 1) (J)	450nm	4.5μm	9.0μm
Near-threshold	4.27p	7.21p	10.4p
CAL 2.5V	11.6p	21.1p	25.8p
ECRL 2.5V	8.98p	10.4p	11.6p
CAL 1.5V	8.75p	12.1p	18.9p
ECRL 1.5V	8.39p	9.42p	9.80p
Conventional static CMOS	1.11n	10.2n	20.4n
Energy over 300 cycles (case 2) (J)			
Near-threshold	4.27p	7.21p	10.4p
CAL 2.5V	328p	2.71n	5.38n
ECRL 2.5V	209p	1.75n	3.5n
CAL 1.5V	48.1p	247p	478p
ECRL 1.5V	22.2p	93.1p	177p
Conventional static CMOS	1.11n	10.2n	20.4n
Energy over 300 cycles (case 3) (J)			
Near-threshold	4.27p	7.21p	10.4p
CAL 2.5V	753p	5.94n	11.8n
ECRL 2.5V	717p	5.31n	10.4n
CAL 1.5V	137p	854p	1.67p
ECRL 1.5V	143p	811p	1.55n
Conventional static CMOS	1.11n	10.2n	20.4n

Table C.6: Transient results of NOR (b)