A Constant Delay Logic Style - An Alternative Way of Logic Design

by

Pierce I-Jen Chuang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

High performance, energy efficient logic style has always been a popular research topic in the field of very large scale integrated (VLSI) circuits because of the continuous demands of ever increasing circuit operating frequency. The invention of the dynamic logic in the 80s is one of the answers to this request as it allows designers to implement high performance circuit block, i.e., arithmetic logic unit (ALU), at an operating frequency that traditional static and pass transistor CMOS logic styles are difficult to achieve. However, the performance enhancement comes with several costs, including reduced noise margin, charge-sharing noise, and higher power dissipation due to higher data activity. Furthermore, dynamic logic has gradually lost its performance advantage over static logic due to the increased self-loading ratio in deep-submicron technology (65nm and below) because of the additional NMOS CLK footer transistor. Because of dynamic logic's limitations and diminished speed reward, a slowly rising need has emerged in the past decade to explore new logic style that goes beyond dynamic logic.

In this thesis a constant delay (CD) logic style is proposed. The constant delay characteristic of this logic style regardless of the logic expression makes it suitable in implementing complicated logic expression such as addition. Moreover, CD logic exhibits a unique characteristic where the output is pre-evaluated before the inputs from the preceding stage is ready. This feature enables performance advantage over static and dynamic logic styles in a single cycle, multi-stage circuit block. Several design considerations including appropriate timing window width adjustment to reduce power consumption and maintain sufficient noise margin to ensure robust operations are discussed and analyzed. Using 65nm general purpose CMOS technology, the proposed logic demonstrates an average speed up of 94% and 56% over static and dynamic logic respectively in five different logic expressions. Post layout simulation results of 8-bit ripple carry adders conclude that CD-based design is 39% and 23% faster than the static and dynamic-based adders respectively. For ultra-high speed applications, CD-based design exhibits improved energy, power-delay product, and energy-delay product efficiency compared to static and dynamic counterparts.

Acknowledgements

There are so many people that I am thank for. Without them, there is no way I can accomplish this work.

I would like to thank my supervisor Professor Manoj Sachdev for first encouraging me to go for graduate studies and then patiently mentor and helping me during every step of this journey. Without him, this research experience will definitely be not as enjoyable and rewarding. I respect him both as a researcher and a person who really thinks for and takes care of his students.

I have been very fortunate to work with all CDR group members for keeping me company and for all the random discussions and helpful hints in the office. Thanks to Dr. David Rennie for all the useful suggestions and for helping me throughout the chip tape-out process. Especially thanks to David Li who introduced me to the CDR group and helped me countless times throughout the last two years. Thanks to Jaspal, and Adam for taking me to Tim Horton and C&D during coffee breaks and share their interesting life stories.

A special thanks must go to Xiaoxia. You have always been on my side and I cannot tell you how much difference you have made in my life. Finally, I cannot express enough gratitude to my parents. Mom and Dad thank you for being supportive of my academic decisions, even though this means that we are half an earth apart.

To My Parents

Contents

List of Tables

List of Figures

Chapter 1

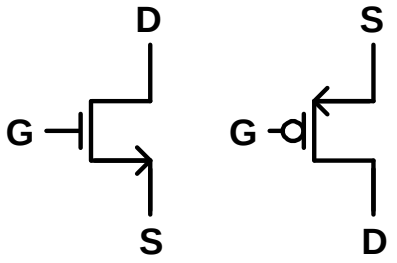
Introduction

The integrated circuit (IC) has become a vital element in today's life as it enables the realization of virtually all electronic devices that were unthinkable a few decades ago. This is thanks to Moore's Law, which states that the number of transistors for a given die area doubles every eighteen months. The most important drive horse that carries on Moore's Law in the past forty years is the scaling of Complementary Metal Oxide Semiconductor (CMOS) technology. The down scaling of CMOS devices to molecule-level dimensions allows more components to be packed in a given area, hence improving density, performance and power consumption. However, as CMOS technology further scales down to allow faster IC with less power consumption, continuous circuit innovation, in particular, logic implementation, is a necessity in order to avail its benefits.

1.1 Complementary Metal Oxide Semiconductor Transistors

Conventional CMOS technology contains two types of transistors, n-channel (NMOS) and p-channel (PMOS), on the same silicon material. These two types of transistors are vital in digital circuits such as logic implementations because they are responsible for passing logic "0" and "1". The basic schematic of NMOS and PMOS transistor is shown in Figure 1.1.

Two types of representations are conventionally used to distinguish between NMOS and PMOS devices. In the area of digital design, PMOS transistor is often represented with a circle at its gate terminal (G) while NMOS transistor does not have the circle. In terms of analog circuits, NMOS and PMOS transistors are identified by an arrow pointing away and toward the gate terminal (direction of the current flow) respectively. Since this thesis is mainly focused on the digital logic design, only the first method of representation (circle) is used. In the following section, a detail description of NMOS transistor's behaviours under different condition is provided while PMOS transistor follows the same equations and arguments.



(a) (b)

Figure 1.1: Schematic of (a) NMOS and (b) PMOS transistor.

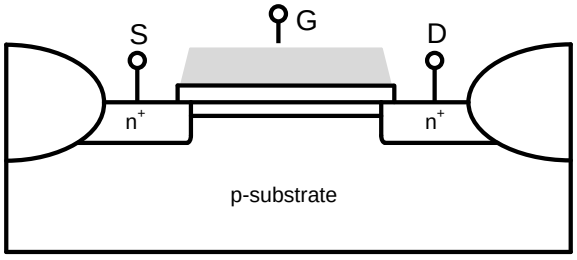


Figure 1.2: Cross section of a NMOS device.

1.2 NMOS Transistor

A cross section of a NMOS device on a silicon wafer is shown in Figure 1.2. Depending on the gate (G) to source (S) voltage difference (V_{gs}) and drain (D) to S voltage difference (V_{ds}), the transistor can work at three different regions, namely subthreshold, triode(linear) and saturation regions.

1.2.1 Subthreshold Region

Subthreshold (weak-inversion) conduction takes place when V_{gs} is less than the threshold voltage (V_t). The weak current conducting in the transistor channel in this region is conventionally treated as leakage current. However, in recent years it has been demonstrated that very high energy efficient circuit system can be achieved when the supply voltage is scaled below V_t and is suitable for applications where speed is not the primary concern [2, 3]. The current in this region can be approximated by the following expression:

$$I_d = I_s e^{\frac{V_{GS}-V_t}{nkT/q}} \left(1 - e^{-\frac{V_{DS}}{kT/q}}\right)(1 + \lambda V_{DS}) \tag{1.1}$$

where I_s and n are empirical parameters with n typically in the range of 1 to 1.5, kT/q is the thermal voltage and is equal to $26mV$ at 300K, and λ is an empirical parameter called channel-length modulation. Based on Equation 1.1, the subthreshold current is exponentially dependent on V_{GS}. Moreover, if V_{DS} is sufficiently large ($>$ 100mV) and assume that λV_{DS} is much smaller than 1, then $e^{\frac{V_{DS}}{kT/q}}$ can be neglected and the current is now independent of V_{DS}. This suggests that the transistor behaves like a current source, generating a constant current which is entirely dependent on V_{GS}.

1.2.2 Triode (Linear) Region

NMOS transistor enters this region when $V_{GS} > V_t$ and $V_{DS} < V_{GS} - V_t$. The current in this region can be approximated by the following expression:

$$I_d = \mu_n C_{ox} \frac{W}{L} \left[(V_{GS} - V_t) V_{DS} - \frac{{V_{DS}}^2}{2} \right] \tag{1.2}$$

where μ_n is the charge-carrier effective mobility and C_{ox} is the gate oxide capacitance per unit area. $\mu_n C_{ox}$ is also known as the process transconductance and is often denoted as kn'.

In this region the transistor is turned on and a channel has been created which allows current to flow between the source and drain terminal. When V_{DS} is small enough, $\frac{{V_{DS}}^2}{2}$ from Equation 1.2 can be neglected and the current is linearly proportional to V_{GS}. Hence transistor behaves like a resistor in this region.

1.2.3 Saturation Region

Transistor operates at the saturation region when the following conditions are satisfied: $V_{GS} > V_t$ and $V_{DS} > V_{GS} - V_t$. In this case, the current is no longer a linear function of V_{DS}; instead, it now has a squared dependency with respect to the gate-source voltage. The current in this region can be approximated by the following expression:

$$I_d = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{GS} - V_t)^2 \tag{1.3}$$

Equation 1.3 suggests that the transistor in the saturation region behaves like a perfect current source. The current flowing between source and drain terminal is constant regardless of the value of V_{DS} and is only dependent on V_{GS}. This simplified assumption is not entirely correct since V_{DS} modulates the current as well. A more accurate description of the MOS transistor current in the saturation region then becomes:

$$I_d = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{GS} - V_t)^2 (1 + \lambda V_{DS}) \tag{1.4}$$

where the channel-length modulation effect takes place, similar to the current equation in the subthreshold region.

As transistor's channel length continues to shrink due to technology scaling, current behaviours begin to deviate considerably from Equation 1.4 and new physical phenomenon,

also known as short-channel effects, begin to influence transistor's current behaviours. Among all the short channel effects, the main culprit for this deviation is the velocity saturation.

Velocity Saturation

Velocity saturation effect happens due to high lateral (horizontal) electric field between source and drain terminal. Consider the empirical Equation 1.5 which states that the average carrier drift velocity is directly proportional to carrier mobility μ and electric field E, which is the voltage difference between drain and source terminal (V_{DS}) divided by the channel length L.

$$\nu_n = \mu E \tag{1.5}$$

This simplified assumption only holds at low electric field. At high lateral field strength, the average carrier drift velocity does not follow this linear model but saturates at a constant value due to carrier scattering. In this regard, increasing electric field, hence the voltage difference between drain and source, no longer improves transistor's current output. Instead, transistor's current is saturated at I_{DSAT}, and the current behaviour is better approximated by the following expression:

$$I_d = \mu_n C_{ox} \frac{W}{L} \left[(V_{GS} - V_t) V_{DSAT} - \frac{{V_{DSAT}}^2}{2} \right] (1 + \lambda V_{DS}) \tag{1.6}$$

where V_{DSAT} is the velocity saturation voltage. Equation 1.5 and 1.6 lead to three observations:

- Velocity saturation effect is more prominent in short channel devices because at shorter channel length L, lower V_{DS} is required before the carrier drift velocity v_n saturates.
- Shorter channel devices therefore experience an extended saturation region, and tend to operate more in saturation conditions.

- The saturation current I_{DSAT} is linearly dependent on the gate to source voltage V_{GS} in the velocity saturation region instead of the squared dependence in the original saturation current expression. This reduces the amount of current a transistor can deliver for a given V_{GS}.

1.3 Digital Design Performance Merits

1.3.1 Delay

Delay is one of the most important performance merits for digital designers. From a digital system's perspective, the delay determines how fast, hence the operating frequency, a particular digital circuit can run on. The delay metric is further defined as the propagation delay, t_p, which measures how quickly a system responds to a change at its input(s) and is measured between the 50% transition points of the input and output waveforms, as shown in Figure 1.3. t_{pHL} defines the response time of a system for a high (input) to low (output) transition while t_{pLH} refers to a low to high transition. The propagation delay t_p is defined as the average of the two and is expressed as:

$$t_p = \frac{t_{pLH} + t_{pHL}}{2} \tag{1.7}$$

In addition, the propagation delay is also a function of the slopes of the input and output signals, as shown in Figure 1.3. Therefore, two more merits, namely the rise and fall time, t_r and t_f, are introduced to measure the transition time between 10% and 90% of the rise and fall waveforms respectively.

1.3.2 Dynamic Power Dissipation

Another important digital system performance merit is the power consumption. Power consumption measures how much power a particular digital circuit needs to consume under specific condition. The power consumption of MOS transistors can be further categorized as dynamic and static (leakage) power consumption.

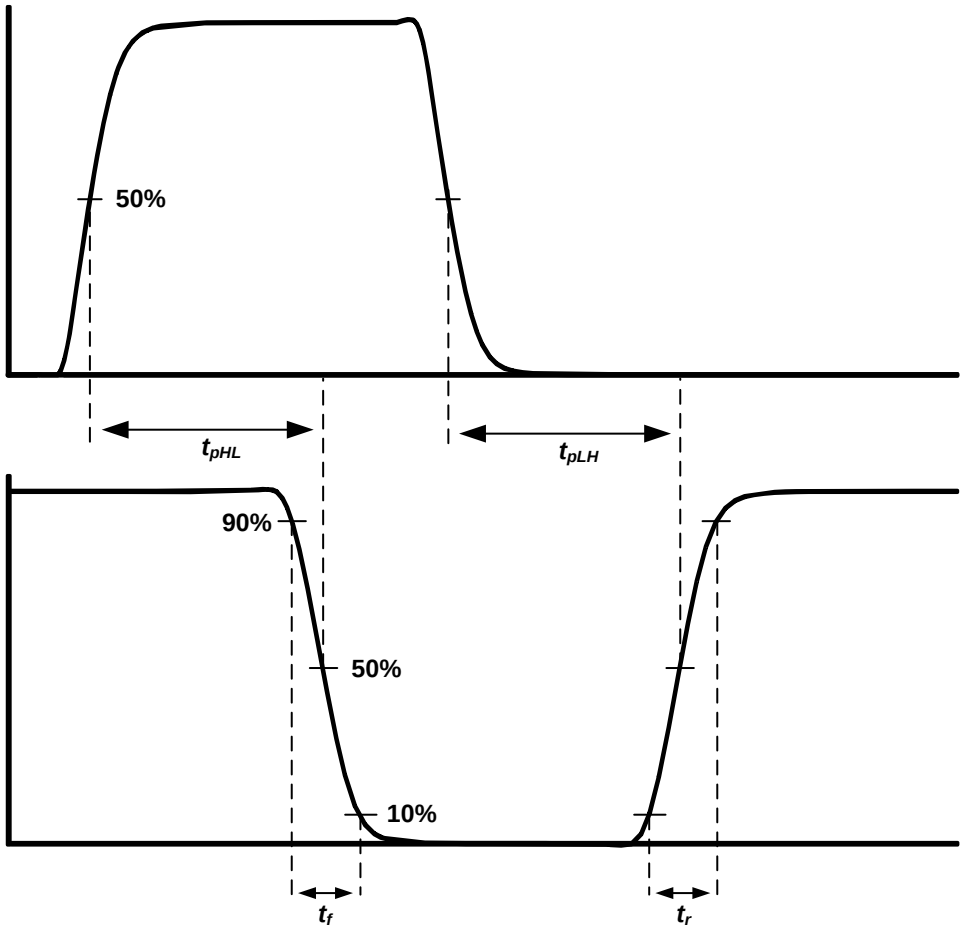


Figure 1.3: Definition of propagation delay and rise and fall time

Dynamic power consumption is defined as the energy a transistor requires to charge or discharge the capacitor. For instance, each time a capacitor is charged through the PMOS transistor, its voltage rises from GND to VDD and a certain amount of energy is drawn from the power supply, and vice versa for the NMOS transistor. The amount of energy E_{supply} taken from the supply during this transition can be derived by integrating the instantaneous power over the period of transition:

$$E_{supply} = \int_0^\infty i_{VDD}(t) V_{DD}\, dt = V_{DD} \int_0^\infty C_L \frac{dv_{out}}{dt}\, dt = C_L V_{DD} \int_0^{V_{DD}} dv_{out} = {C_L V_{DD}}^2 \tag{1.8}$$

while the amount of energy E_C stored on the capacitor at the end of the transition is

$$E_C = \int_0^\infty i(t)V_{out}\,dt = \int_0^\infty C_L \frac{dv_{out}}{dt} v_{out}\,dt = C_L \int_0^{V_{DD}} v_{out}\,dv_{out} = \frac{C_L {V_{DD}}^2}{2} \tag{1.9}$$

where E_C is the amount of energy stored in the capacitor at the end of the transition and C_L is the load capacitor. Equation 1.8 and 1.9 suggest that only half of the energy supplied by the power source is stored in C_L, while the other half has been dissipated by the transistor, regardless of the transistor dimension. Also, E_C has a squared dependency on the supply voltage V_{DD}, hence reducing the voltage supply is the most effective way of reducing the dynamic energy consumption due to this quadratic relationship.

The capacitor charging or discharging event and hence the dynamic energy consumption only takes place when the device is switched. In order to compute the dynamic power consumption, it is necessary to take into account how often the device is switched. Denotes the number of times a device is switched per second as f_{switch}, then the dynamic power consumption is given by

$$P_{dynamic} = C_L {V_{DD}}^2 f_{switch} \tag{1.10}$$

Since $P_{dynamic}$ is linearly proportional to f_{switch}, advances in technology scaling results in even-higher dynamic power consumption due to higher f_{switch} (since t_p decreases). Even though C_L of an individual transistor reduces as technology scales due to smaller dimension, the total capacitance on the chip actually increases as more and more gates are placed on a single die. In addition, as explained in the following section, circuit style, such as dynamic logic, suffers from higher $P_{dynamic}$ due to higher data activity (f_{switch}) compared to static logic.

1.3.3 Static Power Dissipation (Leakage)

The other major source of power consumption is the static power dissipation and is expressed by the following relation:

$$P_{static} = I_{static} V_{DD} = I_{leakage} V_{DD} \tag{1.11}$$

where $I_{leakage}$ is the leakage current (subthreshold current) that flows between the supply rails when the device is not switching (turned off) and can be approximated by the subthreshold current expression shown in Equation 1.1. In larger CMOS manufacturing processes such as 0.25μm static power dissipation is not a major concern because dynamic power dissipation dominates the overall energy consumption. However, as technology scales down to nano-scale, the thin oxide thickness (for aggressive nanometer CMOS process, the thickness can be as thin as only few hydrogen molecules) along with other short channel effects significantly increases the leakage. Figure 1.4 illustrates the dynamic vs. static power dissipation of static random access memory (SRAM) as technology scales [1]. Clearly, as technology continues to shrink, leakage power becomes the dominant source of power consumption.

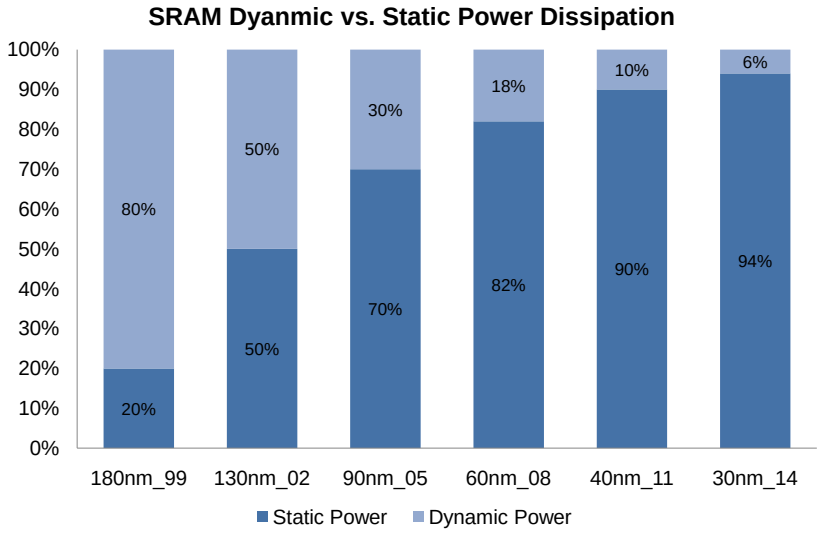


Figure 1.4: SRAM dynamic vs. static power dissipation at different CMOS process [1]

1.4 Logic Implementations: Circuit Families

1.4.1 Static Logic

Static logic is the most widely used logic style in CMOS technology and its basic structure is shown in Figure 1.5. It consists of a NMOS pull-down network (PDN) and a PMOS pull-up network (PUN). The primary advantages of static logic are robustness, low power dissipation especially at low data activity factor, and adequate performance with no static power dissipation. Its most distinct characteristic is that at any given time, the gate output is connected to either VDD or GND via a low-resistance path. While this unique feature ensures static logic's robustness, it is also a major drawback since static CMOS requires both NMOS and PMOS transistors on each input. During a falling output transition, PMOS transistors do not contribute to the pull-down transition current but only add significant capacitance. Hence, static CMOS has a relatively large logical effort and area penalty and is slow when implementing complicated logic expression such as 4-input XOR. The schematic of a two-input static NAND and NOR gate is shown in Figure 1.6. For the

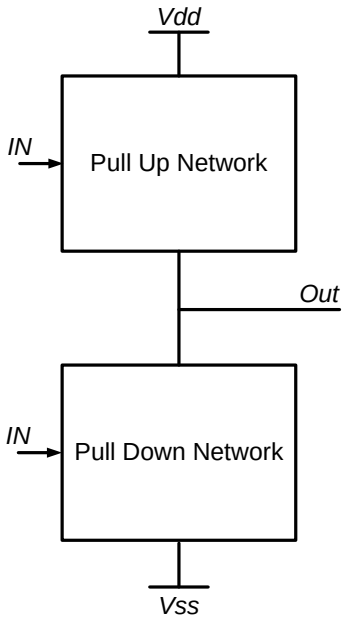
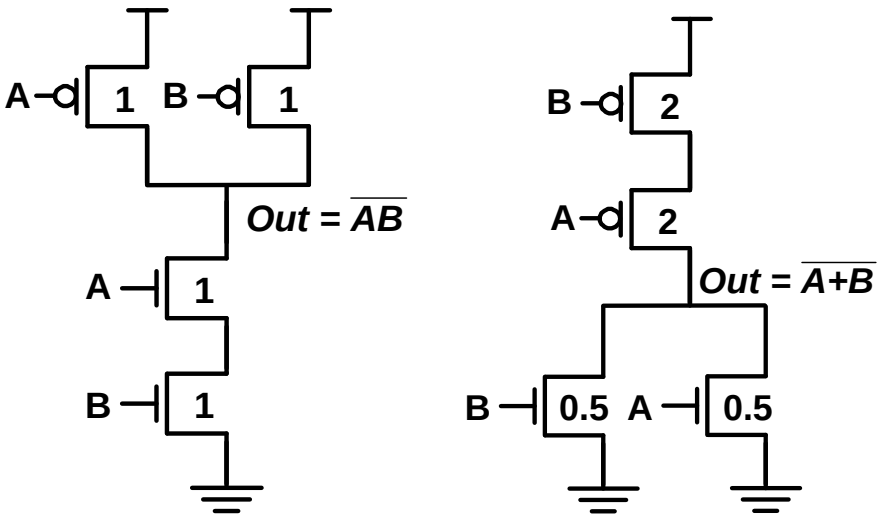


Figure 1.5: Static Logic as a combination of a pull up and down network.

two-input NAND gate, Out is connected to VDD when either A or B is logic "0" and is only connected to GND when both A and B are at logic "1". On the other hand, Out is connected to VDD only when both A and B are logic "0" and is connected to GND for the rest of the time in a two-input NOR gate. At every point in time, Out is not floating and is computed as the value of the boolean function implemented by the PDN and PUN. The PDN and PUN are implemented using NMOS and PMOS devices because they can pass strong logic "0" and "1" respectively. PMOS devices are typically sized up two times larger than NMOS devices to provide equal rise and fall delay due to lower hole mobility. Therefore, PMOS transistors have to be up sized four times larger than NMOS transistors to achieve equivalent rise and fall delay for the two-input NOR gate. The up-sized PMOS transistors contribute input capacitance for both transitions, while only helping the rise delay. In this regard, PMOS devices become the area bottleneck for static CMOS logic style when implementing NOR gate (PMOS devices in series). Furthermore, the up-sizing technique provides diminished rising delay improvement due to self-loading effect, since the additional drain capacitance introduced by up-sizing gradually offsets the performance enhancement contributed by higher pull-up current as a result of larger width device.



(a) (b)

Figure 1.6: Schematic of a two-input static (a) NAND gate and (b) NOR gate.

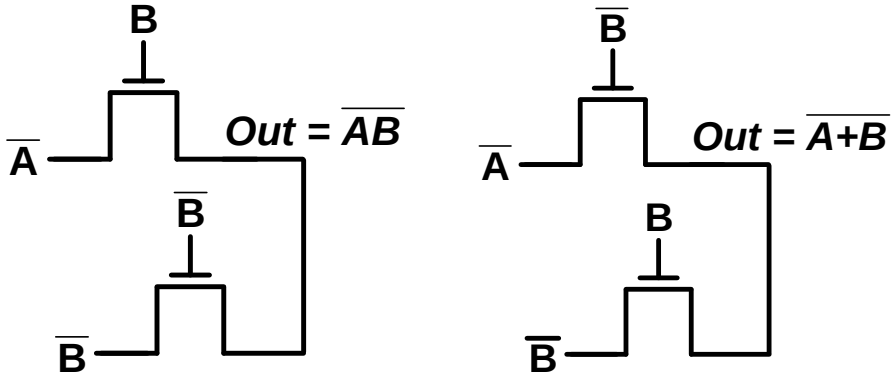


Figure 1.7: Schematic of a two-input pass transistor (a) NAND gate and (b) NOR gate.

1.4.2 Pass Transistor Logic

In the previous section, static CMOS logic is described where the logic inputs are only applied to the gate terminals of the transistors. In this section, another conventional logic style is introduced where inputs are also applied to the source/drain diffusion terminals of the transistors to reduce area and power while not sacrificing performance. This type of logic style is called pass transistor logic (PTL). Figure 1.7 depicts the schematic of a two-input pass transistor NAND and NOR gate. Compared to static logic, fewer transistors (hence lower capacitance) are required to implement the same function. For instance, the implementation of static NAND gate requires four transistors while PTL NAND gate only requires two transistors. Nevertheless, PTL requires complementary signals which are often generated using additional inverters in a single-ended system. This introduces additional hardware overhead and compromises the area advantage. In realistic designs the complementary signals are often shared among several pass transistor gates, hence the additional area overhead can be minimal, depending on the type of application.

The output of PTL should be protected by an inverter(buffer) before driving the next stage load. In other words, PTL logic cannot be cascaded by connecting the output of a PTL to the gate input of another PTL. This is because NMOS transistor can only deliver a weak logic "1" ($VDD - V_{th}$). Consider a pass transistor schematic as shown in Figure

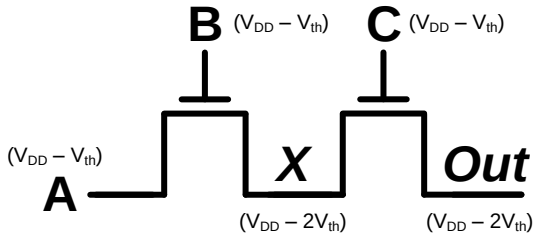


Figure 1.8: Multiple threshold voltage drops at the output of a pass transistor gate

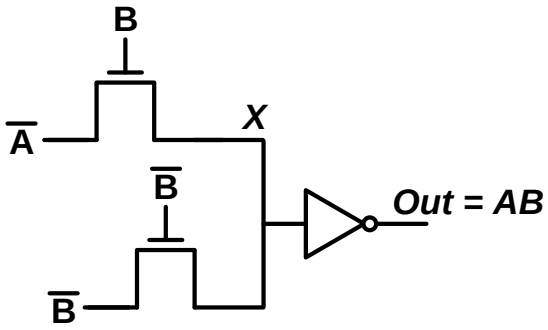


Figure 1.9: Schematic of a two-input pass transistor AND gate

1.8, where signal A, B and C all come from other pass transistor gates directly without protective inverters. If all signals are logic "1", then the maximum voltage at both X and Out will be $VDD - 2V_{th}$, since both A and B only have a maximum voltage swing of $VDD - V_{th}$. Hence in real practice an inverter is always inserted at the output of every PTL before driving the next stage logic.

However, the addition of a protective inverter at the output of PTL causes direct power dissipation because the inverter's PMOS transistor is not completely turned off. Consider a two-input pass transistor AND gate with an inverter at the output, as shown in Figure 1.9. When A is logic "0" and B is logic "1", X is charged up to VDD - V_{th} and Out is discharged to GND through the inverter's NMOS transistor. While PTL successfully evaluates in this case, the gate to source voltage (V_{gs}) of the inverter's PMOS transistor is equal to V_{th} instead of zero. In this regard, the PMOS transistor is not completely off, and a direct current path exists (static power dissipation).

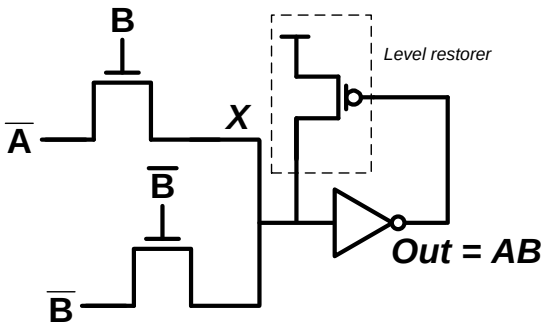


Figure 1.10: Schematic of a two-input pass transistor AND gate with feedback level restorer

A common solution to the voltage drop problem is the use of a level restorer. Figure 1.10 illustrates the schematic of a two-input AND gate with a feedback level restorer circuit. The single PMOS transistor's gate is connected to the output of the inverter and the drain terminal is connected to the input of the inverter. Consider the situation where A is logic "0" and B is logic "1". X is initially charged up to VDD - V_{th} and Out is discharged to GND. Once Out is at GND, the level restorer PMOS transistor turns on, continues to charge up X to full VDD, and eliminates the problematic static current path. Furthermore, no direct current path can exist through the level restorer and the pass transistor logic, since the PMOS transistor is only active when Out is low, which implies that X must be at logic "1".

While this solution mitigates the problem of static power dissipation, it contributes additional capacitance, adds layout complexity, and most importantly, makes pass transistor logic a ratioed logic. When X is to make a logic "1" to "0" transition, the NMOS pull down path now has to fight against the PMOS level restorer because initially the restorer is on. Therefore, the NMOS pull-down path must be stronger than the restorer and careful transistor sizing is necessary in order to make the circuit function properly.

1.4.3 Transmission Gate Logic

Another widely used solution to mitigate the threshold voltage drop problem associated with pass transistor logic is the use of transmission gates (TG). In a TG design, a PMOS

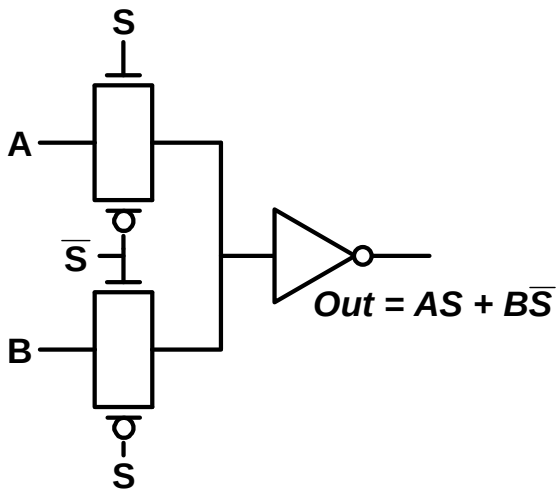


Figure 1.11: Schematic of a transmission gate based multiplexer

device is often placed in parallel with a NMOS device to deliver both strong logic "1" and "0". The schematic of a TG multiplexer is shown in Figure 1.11. This configuration does not have V_{th} drop problem because the parallel PMOS device provides full voltage swing, at the expense of additional transistors and control signals. If transmission gates are used, it is common practice to size the PMOS and NMOS transistors approximately the same width rather than using a double-width PMOS to NMOS size ratio. This is because both transistors are passing the signal in parallel, and the primary objective of the PMOS device is to provide full voltage swing.

1.4.4 Dynamic & Compound Domino Logic

The invention of the dynamic logic in the 80s is one of the answers to the request of ever increasing IC operating speed as it allows designers to implement high performance circuit block, i.e., arithmetic logic unit (ALU), at an operating frequency that the traditional static and pass transistor CMOS logic styles are difficult to achieve [4]. A generalized schematic of a dynamic gate with footer CLK transistor is shown in Figure 1.12. The operation of dynamic logic is as follows: When CLK is low (precharge period), transistor M1 is on, and NMOS PDN is off because M2 is off. X is charged to VDD by transistor M1 and Out is

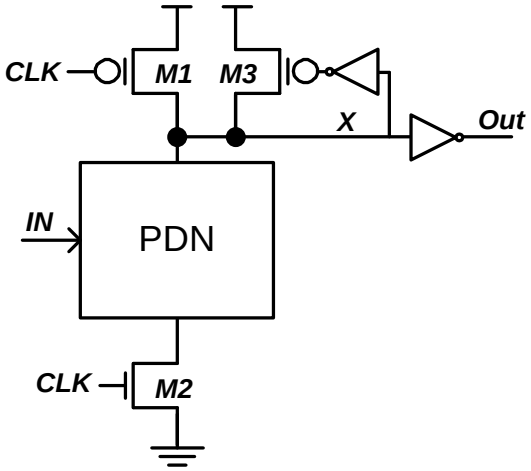


Figure 1.12: Dynamic logic with a footer transistor.

maintained at GND. Dynamic logic enters evaluation period when CLK rises to high. In this case, depending on the input patterns two possible scenarios can take place. If NMOS PDN is off, X will be floated because both M1 and PDN are off. Therefore, a small PMOS keeper (M3) is required to fight against the leakage and to help maintaining the voltage of node X at VDD. On the other hand, if NMOS PDN is on, then X is quickly discharged to GND and Out is charged up to VDD via the inverter. Dynamic logic does not have the problem of static power dissipation because when X is at GND (Out is at VDD), PMOS keeper M3 is guaranteed to be off. When Out is discharged, it cannot be charged again until the next precharge period begins. Thus the inputs to the gate of NMOS PDN can make at most one transition during evaluation. In summary, the unique characteristics of dynamic logic are:

- The logic function is implemented with NMOS transistors only.
- The number of transistors for complicated logic expression implemented with dynamic logic is substantially lower than the static case.
- Dynamic logic has faster switching speed because less number of transistors (especially without any PMOS logic transistors) contributes to less load capacitance.
- It only consumes dynamic power since no static current path ever exists between

VDD and GND. However, the overall power consumption can be significantly higher than the static design because of the higher switching activity.

The performance enhancement comes with several costs however, including reduced noise margin, charge-sharing noise, and higher power dissipation due to higher data activity. In a traditional dynamic logic, an output inverter is required between dynamic logics to satisfy the data monotonicity requirement and to ensure proper logic evaluation [5]. This not only increases the overall delay but also the power consumption as well. Two variations of the dynamic logic have been proposed to mitigate this problem. NP domino, or also known as NORA domino [6] [7], replaces this inverter with pre-discharged dynamic gates using PMOS logic [8]. However, NORA is extremely susceptible to noise and has not been used extensively. Zipper domino [9] attempts to achieve the same objective by a slightly different implementation, but is never widespread in the VLSI industry [8], [10].

Furthermore, dynamic logic has gradually lost its performance advantage over static logic due to the increased self-loading ratio in deep-submicron technology (65nm and below) because of the additional NMOS CLK footer transistor (Figure 1.12). This phenomena has been demonstrated in [11], which concludes that at processes such as 180nm and 130nm, the optimal adder architecture is radix-4 (5 transistors in series, including the footer transistor); however, radix-2 (3 transistors in series, including the footer transistor) configuration becomes optimal at 65nm technology and beyond because the increased self-loading ratio has made radix-4 architecture slower than radix-2, even though radix-2 configuration requires more number of stages to complete the addition.

Compound domino logic (CDL) where dynamic and static CMOS gates alternating between each other mitigates the two aforementioned problems and has become the most popular logic style in high performance circuit block, i.e., 64-bit adder in modern central processing unit (CPU) [12][13][14][15]. In this design, the output inverter is replaced with a more complex inverting static CMOS gates (Figure 1.13), i.e., NAND or NOR, such that the monotonicity requirement is satisfied while conducting complex logic operations without wasting the one inverter delay [16]. Moreover, all the dynamic stages except the first stage can be footless (the footer transistor is eliminated) in CDL, thus reduce the total stack height by one. However, this implementation comes at the expense of increased

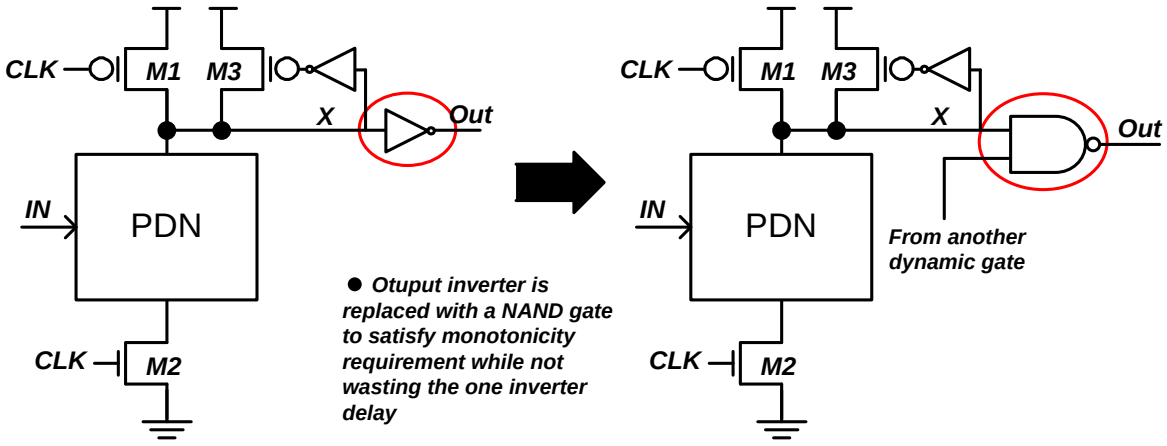


Figure 1.13: Dynamic Logic vs. Compound Domino Logic

power consumption due to the direct path current from VDD to GND during the precharge period [8]. While CDL offers higher performance and reduced power consumption over pure static and dynamic logic style respectively [17], its noise margin is significantly degraded as in a CDL design, the output of the dynamic logic without any buffer is required to drive the next stage via a long interconnect and with other signal wires running in parallel. The crosstalk of the adjacent wire can potentially flip the state of the dynamic logic, and results in false logic evaluation [11]. As a result, extra distance among wires running in parallel has to be enforced in laying out such a design at the expense of increased total wire length. In the extreme case, power rails are placed in between adjacent wires to eliminate the crosstalk problem. This technique nevertheless, causes significant performance degradation and increased power consumption as a result of increased parasitic capacitance. Because of this reliability concern, CDL is regarded as a less robust logic style and is not considered in this thesis.

1.4.5 Source Couple Logic

Source-coupled logic (SCL) [18] (CMOS equivalent of current-mode logic) has shown superior performance that is difficult to achieve by any other logic styles. SCL circuits are also ideal for integration into low-noise environments, due to their differential nature and their

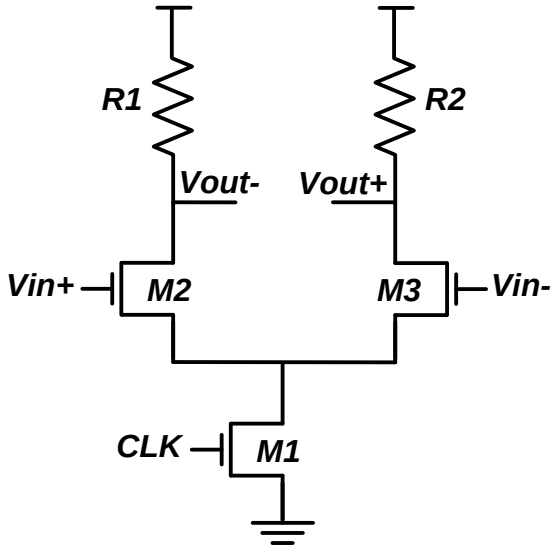


Figure 1.14: Schematic of a SCL buffer

lack of switching noise. The schematic of a SCL buffer is shown in Figure 1.14. SCL suffers from high power dissipation due to the concept of current-steering with constant current draw and its differential characteristic requires complementary signals. Therefore, SCL is rarely used in designing digital processor logic core but has found extensive applications in multi-Gbs multiplexer, I/O, and frequency dividers for high frequency communication system [19][20][21][22].

1.4.6 Feedthrough Logic

A significant research effort has been dedicated to explore new logic styles that go beyond dynamic logic and CDL. In recent years, a new way of logic operation, also known as feedthrough type logic (FTL) style [23, 24], has been proposed and demonstrated its high performance capability. Consider a conventional NMOS dynamic logic (Figure 1.12), the PMOS clock transistor's functionality is to restore the logic state to logic "1" during the precharge period and the NMOS clock transistor is to act as a footer which disables the discharge path when CLK is low. During the evaluation period, the logic state of the dynamic logic before the inverter can either retain at logic "1" or discharge to logic

"0" through NMOS logic transistors, depending on their inputs. In this manner, logic transistors are the critical path as they determine the amount of time that the dynamic logic needs to complete a logic evaluation. In FTL style however, the role of the clock and logic transistors are interchanged. Logic transistors are no longer the critical path; instead, their main functionality is to act as "keepers" to maintain the correct logic level. On the other hand, clock transistor is now the critical path and determines the delay of FTL.

The first generation of FTL logic exhibits many shortcomings including excessive power dissipation and reduced noise margin in cascaded stages, which are explained in the next section. An improved FTL design which applies a global window technique to reduce the power consumption and uses a keeper to maintain sufficient noise margin at each stage is proposed [25]. However, this circuit topology exhibits similar crosstalk problem as that of CDL design. The global window technique implies that all FTL gates will use the same window duration. This is clearly not an optimized design since window width should be proportional to the total internal and parasitic interconnect capacitance that a gate drives.

A new high performance logic, that is called constant delay (CD) logic, is described in this thesis. CD logic mitigates the aforementioned problems by applying a local window technique and a self-reset circuit which enables robust logic operation with minimized power consumption compared to previous designs. Simulation results with back-annotated lump capacitance in 65nm CMOS technology indicate that CD logic is on average 94% and 56% faster than static and dynamic logic respectively in five different logic expressions. Furthermore, 8-Bit ripple carry adder (RCA) implemented with the proposed CD logic style is more energy (power $\times$ total simulation time), power-delay product (PDP, power $\times$ delay), and energy-delay product (EDP) efficient when performance is the primary concern.

1.5 Thesis Organization

The thesis is organized as follows: Chapter 2 reviews the history of feedthrough logic and introduces the proposed CD logic. Chapter 3 characterizes the proposed CD logic and analyzes the impacts of window width on performance and robustness. Chapter 3 also compares CD logic with other logic styles in different logic expression to demonstrate CD

logic's performance superiority and power consumption drawback. Chapter 4 shows the simulation results of three 8-Bit RCA implemented with static, dynamic, and CD logic. Finally, Chapter 5 concludes with some final remarks and comments.

Chapter 2

Evolution of Constant Delay Logic

2.1 FTL Logic

Feedthrough type logic (Figure 2.1) in CMOS technology was first introduced in [23, 24]. Its basic operation is as follows: when CLK is high, the predischarge period begins and Out is pulled down to GND through M2. When CLK becomes low, M1 is on, M2 is off and the gate enters the evaluation period. If inputs (IN) are logic "1", Out enters the contention mode where M1 and transistors in the NMOS pull down network (PDN) are conducting current simultaneously. If PDN is off, then the output quickly rises to logic "1". In summary, the main advantages of FTL gate are:

- It only requires NMOS transistor logic expression
- The critical path is constant regardless of the logic expression
- The output is pre-evaluated (ratioed logic) before the inputs from the preceding stage is ready

Despite its performance advantage, FTL suffers from reduced noise margin, excess direct path current, and non-zero nominal low output voltage which are all caused by the contention between M1 and NMOS PDN during the evaluation period. Furthermore,

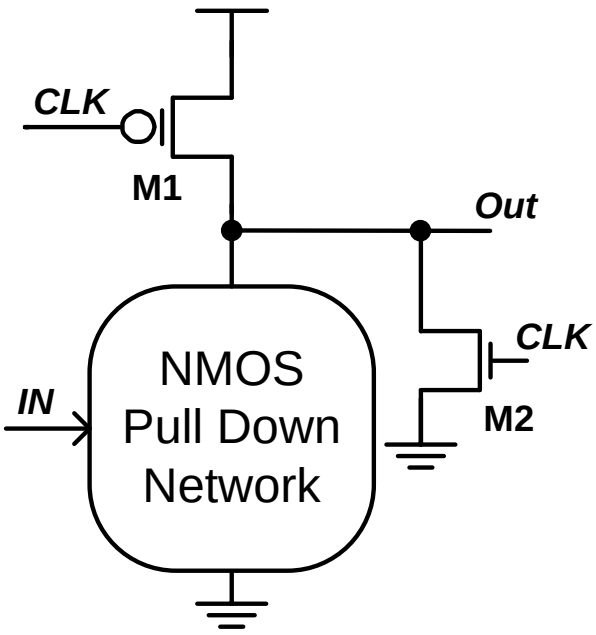


Figure 2.1: Schematic of feedthrough logic (FTL).

cascading multiple FTL stages together to perform complicated logic evaluations are not practical. Consider a chain of inverters implemented in FTL are cascaded together and driven by the same clock. When CLK is low, M1 of every stage turns on, and the output of every stage begins to rise. This will result in false logic evaluation at even number (i.e, 2, 4, 6, etc) stages since initially there is no contention between M1 and NMOS PDN because all inputs to NMOS transistors are reset to logic "0" during the reset period. Figure 2.2 demonstrates this problem and shows the simulated waveform of a chain of inverters implemented in FTL. At even number stages, the output will initially pull up, and then settle back to a non-zero low output voltage. As the number of stages increases, the unwanted glitch also aggravates and takes longer time to settle back.

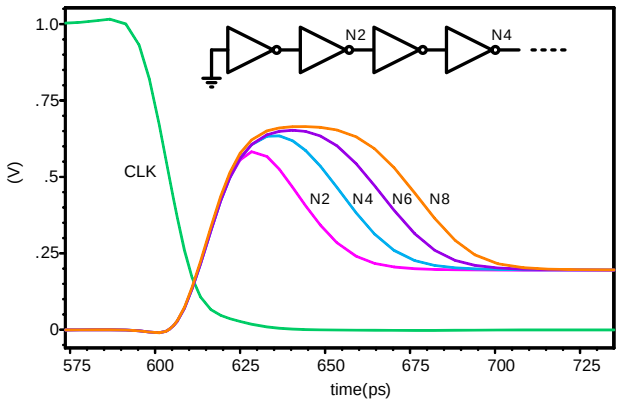


Figure 2.2: Simulated unwanted glitch at different logic depth in a chain of inverters with FTL implementation

2.2 Dynamic Feedthrough Logic

In order to mitigate the aforementioned problems, we proposed a dynamic feedthrough logic (DFTL) style (Figure 2.3) where FTL logic is used in conjunction with dynamic logic [25]. The output of FTL logic does not drive another FTL stage directly. Instead, a dynamic logic is inserted between two FTL stages to act as a buffer while performing logic operation (the concept is similar to compound domino logic). When multiple DFTL stages are cascaded together, transistor M2 in the subsequent stage is no longer required. This is because during the predischarge period (CLK is high), the inputs to the NMOS PDN which come from the dynamic gates of the preceding stage are always charged to logic "1" because the dynamic gates are in the precharge period (driven by $\overline{CLK}$). DFTL eliminates the problem of false logic evaluation associated with cascaded FTL because the inputs to FTL's NMOS PDN are always at logic "1" when first entering the evaluation period. Therefore, FTL gates first enter the contention mode and conditionally make a low to high transition depending on the inputs during the evaluation period.

In [25], a global timing window technique was also proposed to reduce the power consumption. However, this caused the output of the FTL stage to be floated when both the window and NMOS PDN are off during the evaluation period. Therefore, a keeper is required at the output of every FTL stage, which increases the layout complexity and re-

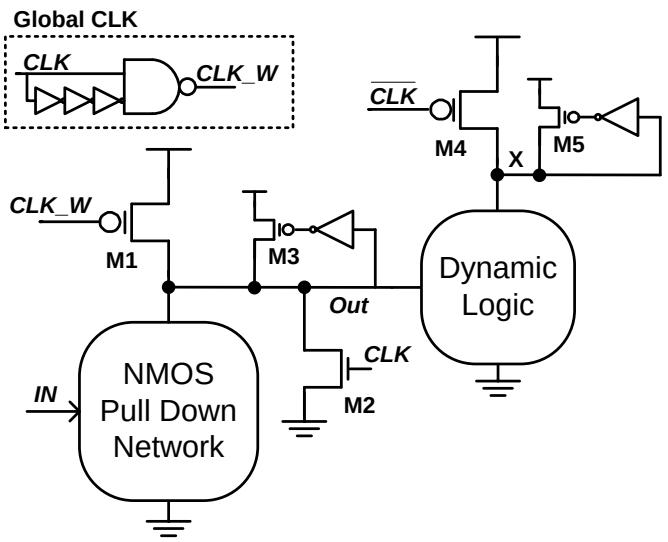


Figure 2.3: Schematic of dynamic feedthrough logic (DFTL).

duces the performance advantage. The width of this global window determines the FTL's functionality, since an insufficient window width will cause the FTL to evaluate incorrectly. It is then extremely important to ensure that the width is sufficiently long to ensure proper logic evaluation of all FTL logic gates under temperature, process, and load variation. This is clearly not the most optimized design since the output capacitance of every FTL logic can vary significantly, depending on the interconnect wire length and the next stage load.

The keeper (M5 in Figure 2.3) at the output of every dynamic stage also needs to be sized up to maintain sufficient noise margin because of the inevitable glitch caused by the contention in the previous FTL stage. The outputs of both FTL and dynamic logic are heavily influenced by the load and interconnect parasitic capacitance (similar to the problem of dynamic logic' output in compound domino logic). This raises a reliability problem in a design with many signals laying out in parallel, as the crosstalk of one signal can potentially flip the state of other signals and results in false logic evaluation. Clearly, while DFTL alleviates some of FTL logic's problems, it still requires further work to make this logic style practical.

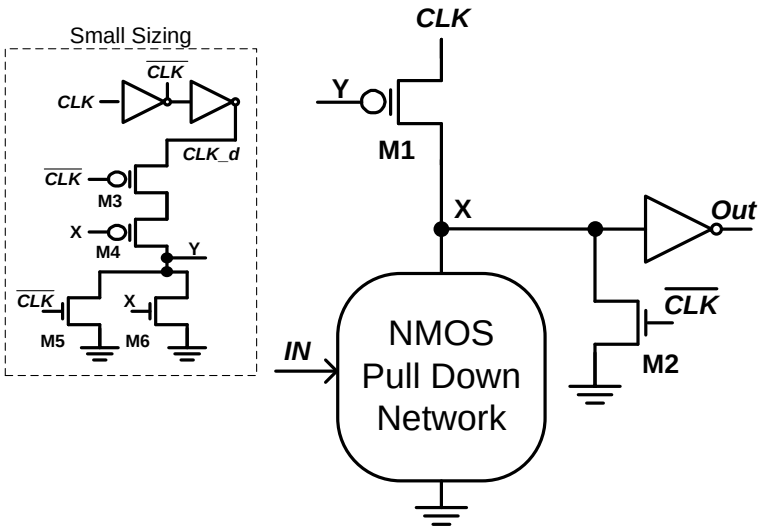


Figure 2.4: Schematic of Constant Delay (CD) Logic

2.3 Constant Delay (CD) Logic

A new constant delay (CD) logic family is proposed. CD logic provides a local adjustable window with self-reset circuitry and maintains the unique characteristic of constant critical path regardless of the logic expression.

CD Logic Operation

Figure 2.4 shows the schematic of a CD gate and Figure 2.5 and Figure 2.6 depict the corresponding timing diagram and flow chart respectively. For simplicity we assume that IN always restore to logic "1" during the predischarge mode and conditionally make a high to low transition prior or during the evaluation period. When CLK is low, CD logic enters predischarge mode and both nodes X and Y are discharged to GND through transistors M1, M2, M5 and NMOS PDN. A unique characteristic of CD gate is that three possible scenarios, naming the contention mode, C-Q delay mode, and D-Q delay mode, can take place when the logic enters evaluation. When CLK becomes high while IN remain at logic

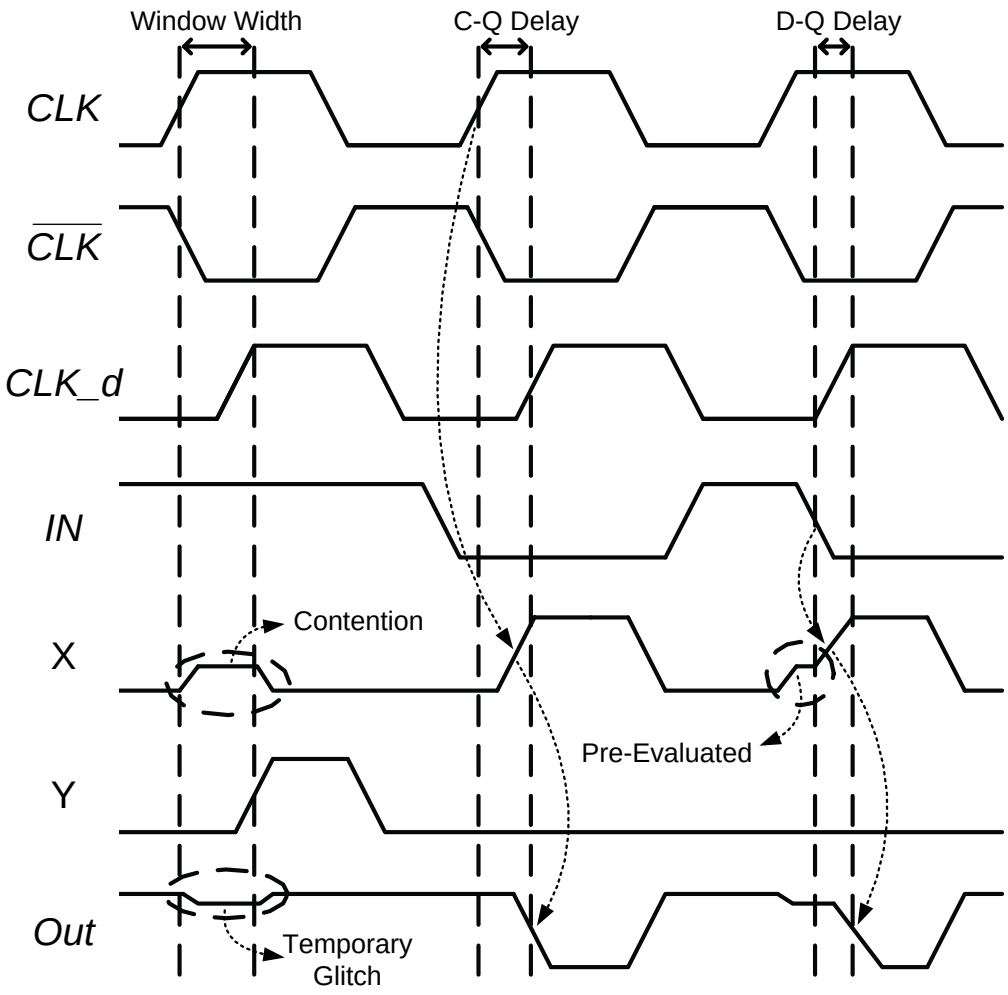


Figure 2.5: Timing diagram of proposed CD Logic

"1", node X enters contention mode and settles at a non-zero voltage level which in term causes Out to experience a temporary glitch. The duration of this glitch is determined by the local widow width, which is determined by the delay between CLK and CLK_d. When CLK_d becomes high, and if node X remains low, then node Y rises to logic "1", and turns off transistor M1. Thus the contention period is over, and the temporary glitch at Out is eliminated.

The second scenario takes places when IN make a transition from high to low before CLK becomes high. In this case, node X rises to logic "1" and node Y remains at logic

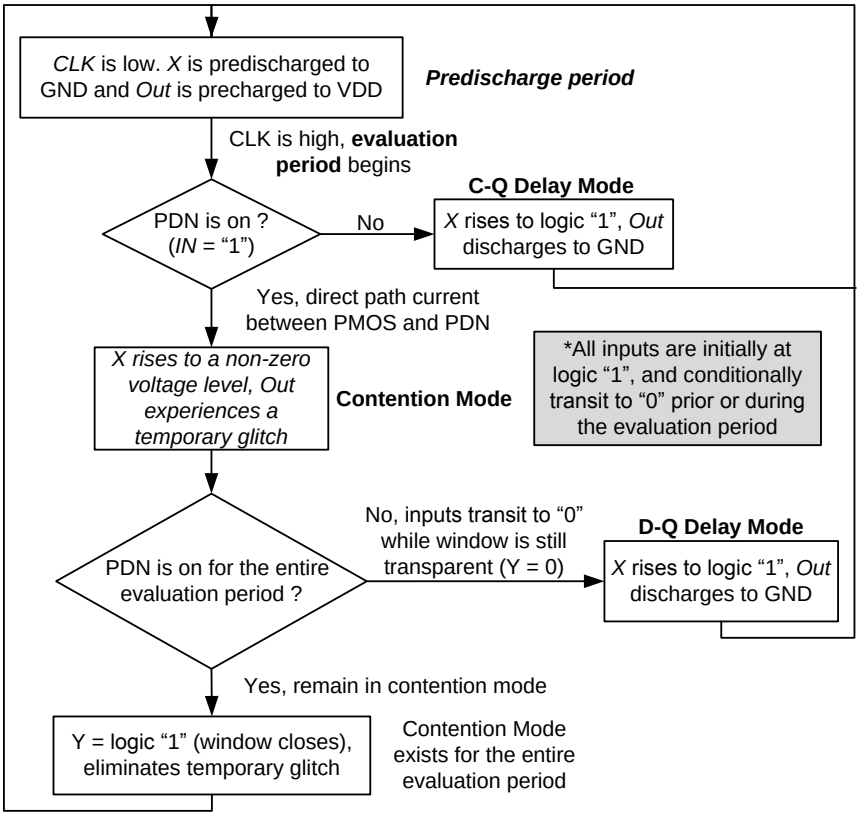


Figure 2.6: Flow chart of proposed CD Logic

"0" for the entire evaluation cycle. In this operation, the delay is measured by the rising edge of the CLK to the falling edge of the output signal, hence the name C-Q delay mode.

The D-Q delay mode utilizes the pre-evaluated characteristic of CD logic to enable high performance operation. In this mode, CLK rises from low to high before IN transit. In this regard, node X initially enters contention mode and rises to a non-zero voltage level. As soon as IN become logic "0", while node Y is still low (window is still transparent), then node X quickly rises to logic "1". A race condition exists in this case between node X and CLK_d. If CLK_d rises much earlier than node X, node Y will go to logic "1", turn off transistor M1, and result in false logic evaluation. If CLK_d rises slightly slower than node X, then node Y will initially rise (thus slightly turns off transistor M1) but eventually settle back to logic "0". CD logic can still perform correct logic operation in this case; however, its performance is severally degraded because of M1's reduced current

Predischarge Mode	Evaluation Mode		
	Contention Mode	C-Q Delay Mode	D -Q Delay Mode
CLK = low. *X* and *Out* are predischarged and precharged to GND and VDD respectively.	*CLK* = high and *IN* = "1" for the entire evaluation period. Direct path current flows from PMOS to PDN. *X* rises to a non-zero voltage level and *Out* experiences a temporary glitch.	*IN* goes to "0" before *CLK* transits to high. *X* rises to logic "1" and *Out* is discharged to VDD. Delay is measured from *CLK* to *Out.*	*IN* goes to "0" after *CLK* transits to high (while window is still transparent). X initially enters contention mode and later rises to logic "1". Delay is measured from *IN* to *Out.*

Figure 2.7: Summary of CD logic's operation

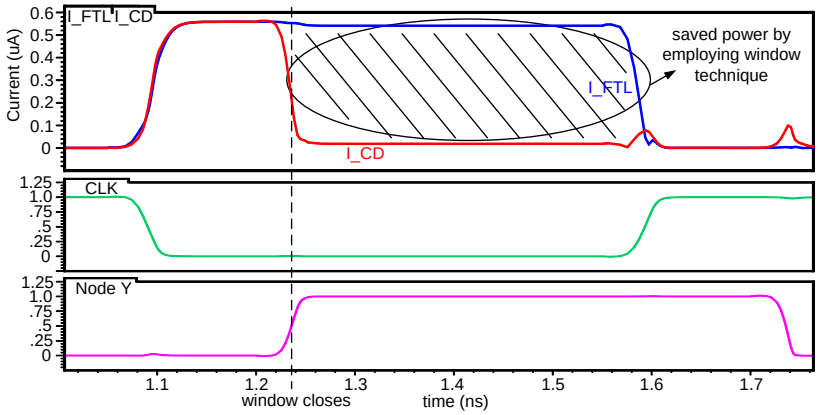


Figure 2.8: Simulated current consumption of FTL and CD Logic. The window technique effectively reduces the current consumption of CD logic (shown by the shaded and circled area).

drivability. Therefore, it is extremely important to maintain a sufficient window width under temperature and process variation to ensure optimal performance. A summary of CD logic's different operation is shown in Figure 2.7.

The window technique reduces CD logic's power consumption during the contention mode, as shown in Figure 2.8. The shaded and circled area indicates the amount of power that is saved by turning off the PMOS transistor, thus shutting down the direct current path. Unlike [25], the local window technique in the proposed CD gate allows designers to customize the window width for different logic expression to achieve minimal power dissipation while not sacrificing performance. For instance, a multiple input NAND gate will require longer window width than NOR gate because of the higher internal capacitance

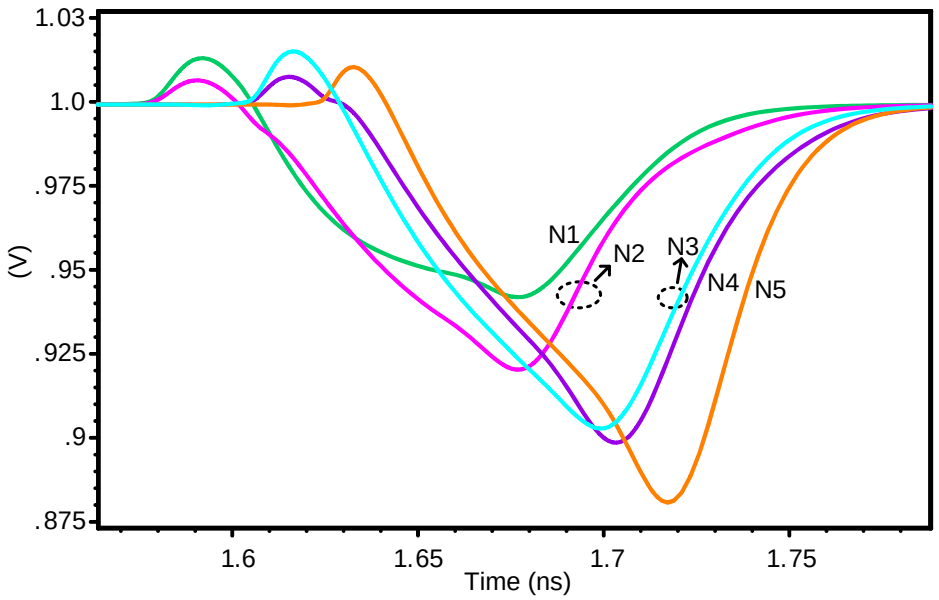


Figure 2.9: Simulated output glitch of a five stage two-input AND gate in CD logic implementation

due to the stacked transistors. Another advantage of CD logic is that the internal node is always connected to either VDD or GND rail, thus making the robustness of CD logic comparable to conventional static CMOS, except during the contention mode. In summary, the advantages of CD logic over DFTL are:

- Local window technique allows designers to optimize individual gate
- CD logic does not require keeper and node X is always connected to either VDD or GND rail except during the contention mode
- The inverter protects the internal node and makes the total internal capacitance predictable

CD Logic Implementation in High Speed Application

The window width plays an important role in determining CD logic's performance and therefore should be investigated in detail. In addition, CD gate experiences temporary

Table 2.1: Average activity factor of various logic styles

Logic Style	Activity Factor (α)
Static	0.1
Dynamic	0.5
CD	0.5*

* Even though CD's activity factor is 0.5, it always consumes power during evaluation period.

glitch which reduces the noise margin during the contention mode. Figure 2.9 shows the simulated output waveform of a five stage 2-input AND gate implemented in CD logic. When the inputs are logic "1" and CLK is high, it is evident that the glitch level aggravates as it traverses though each stage. This raises a robustness concern when multiple CD gates are cascaded together to perform complicated logic evaluation. Therefore, the maximum number of CD logics that can be cascaded together depends on the maximum tolerable unwanted glitch of a given design.

Another drawback of CD logic is the higher data activity (α) compared to both static and dynamic logic style. Table 2.1 summarizes the average α for various logic styles. Static logic has an empirically α of 0.1 [8]. Dynamic logic has an activity factor of 0.5 because every node in dynamic logic must be precharged to VDD every clock cycle. This implies that some dynamic nodes are precharged only to be immediately discharged again in the next evaluation period, leading to a higher activity factor [26]. CD logic is even worse in this case because it always consumes power when it enters evaluation period, even though its α is also 0.5. During the evaluation period, CD logic always dissipates power via either dynamic power dissipation (internal node X goes to VDD and Out is discharged to GND) or direct path dissipation. (CD logic enters contention mode, and the direct current between PMOS and NMOS transistors exists until the window closes)

Clearly, this problem places CD logic in a disadvantageous position compared to static and dynamic logic in terms of power consumption. However, while it is true that CD logic will consume more power than static and dynamic logic, we believe that the high speed system employing CD logic is still an attractive design because: i) CD logic is used only

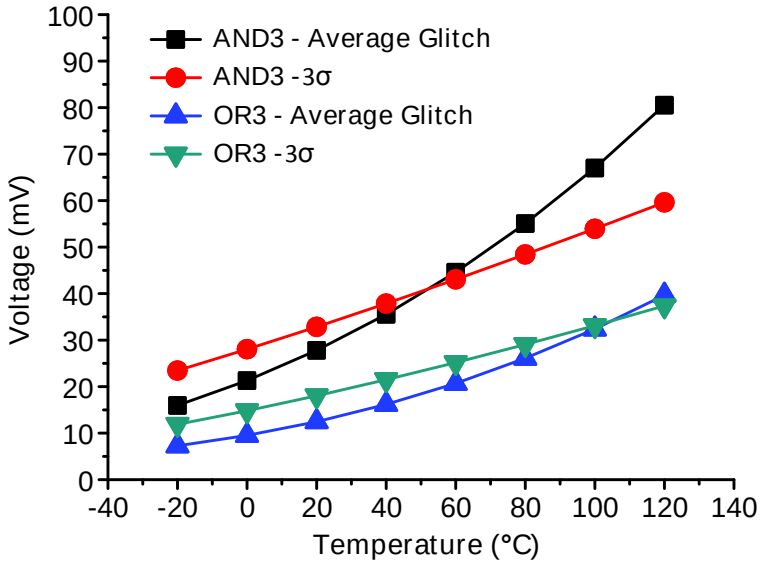


Figure 2.10: Temporary glitch mean and standard deviation at the output of 3-input CD AND and OR gate vs. temperature in a monte-carlo simulation with 7500 iterations.

to replace the critical path instead of the entire circuit and ii) clock gating can reduce CD logic's power consumption.

Clock Gating Technique

Clock gating is a power management technique where the clock connection to idle module (i.e., input data remain the same) is turned off (gated), which in turn reduces the dynamic power consumption of the target module [27] and has been implemented in many modern processors [28, 29]. In other words, since the module's clock only toggles when there is a change in the input data pattern, the effective input data activity increases and the module is more likely to operate at higher input data activity environment. In the rest of the thesis, it is assumed that clock gating technique is employed and the minimum input data activity is at 50%.

Figure 2.10 illustrates the temporary glitch mean and standard deviation (σ) at the output of a 3-input CD AND and OR gate vs. temperature in a monte-carlo simulation with 7500 iterations. The mean and 3σ of a 3-input AND gate's unwanted glitch at 20°C are only

20mV and 25mV, respectively. However, as temperature increases, the worst case glitch level within 99.7% of the cases (assume Gaussian distribution) can reach approximately 130mV at 120°C. Therefore, designers who wish to employ CD logic need to enforce more strict design guidelines in order to sustain the system's reliability, i.e. simulate the circuits under extreme temperature and corner conditions. In this thesis, unless otherwise specified, all the CD logics are designed such that the average unwanted glitch level at nominal VDD (1V) is less than 5% (50mV) at 110°C in a monte-carlo simulations with 7500 iterations. In other words, the average glitch level of CD logic at 110°C needs to satisfy the following condition:

$$\frac{\sum_{i=1}^{7500} \frac{(V_{DD}-Vout_i)}{V_{DD}}}{7500} \times 100\% \leq 5\% \tag{2.1}$$

where $Vout_i$ is the ith iteration's temporary glitch level at Out (Figure 2.4).

Chapter 3

CD Logic Characterization

The simulation test bench for CD logic characterization is shown in Figure 3.1. All simulation runs are done in Cadence environment using 65nm CMOS technology at 1V nominal supply voltage at 27°C with tNtP corner, and with the maximum temporary glitch requirement as described in the previous section. In the first two parts of the analysis, two types of logic, 3-input AND gate and OR gate, are used because of their serial and parallel fan-in characteristics respectively. All gates are sized for minimum PDP using Cadence optimization tool. The input drivers are sized to drive a load of FO4. The clock and data frequencies are set to 2GHz. The window duration (width) is defined as the 50% point of the rising edge of CLK to the 50% point of the rising edge of node Y.

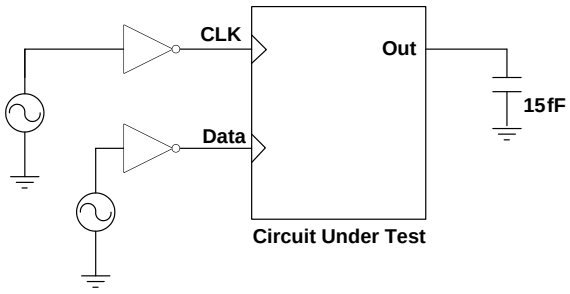


Figure 3.1: Simulation test bench

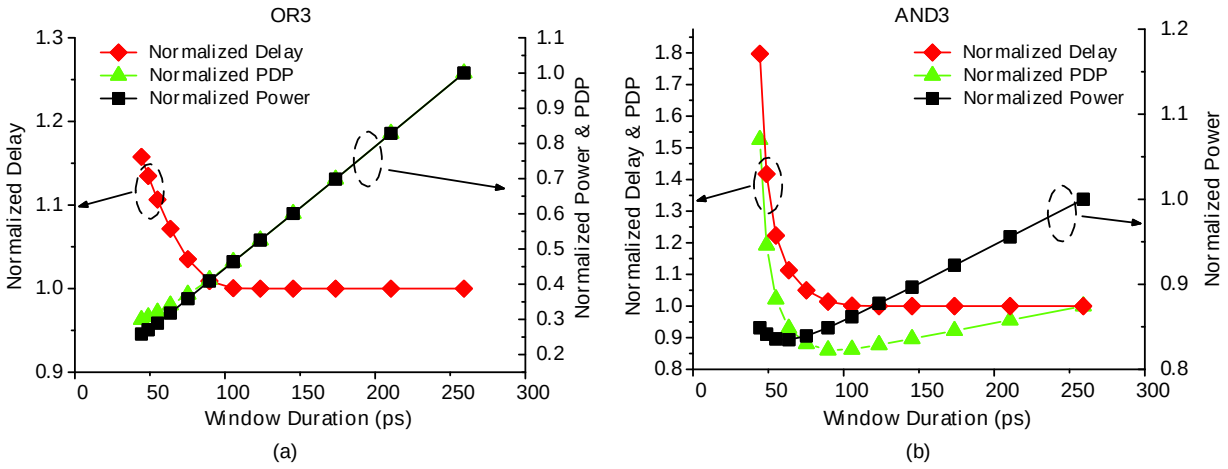


Figure 3.2: Simulated normalized delay, power, and PDP vs. window duration for a 3-input (a) OR gate and (b) AND gate implemented in CD logic

3.1 Performance vs. Window Width

Figure 3.2 shows the normalized delay, power and PDP vs. window duration for a 3-input AND and OR gate. As window width increases, the delay initially decreases but soon reaches a saturated value. This is because when the window duration is too short, the race condition, as mentioned in the earlier section, takes places between node X and Y, thus increases CD logic's delay. When the width is sufficiently long, node X can rise to logic "1" without racing against node Y, and CD logic reveals full performance advantage. One observation can be made from Figure 3.2: window width plays a more dominant role in AND than OR gate in terms of performance. A maximum of 80% performance discrepancy is observed in AND gate whereas a maximum of only 17% is observed in OR gate. This indicates that OR gate is less sensitive to window duration than AND gate because of OR gate's lower internal drain capacitance compared to AND gate. The power consumption of both gates is directly proportional to window duration; hence, the window width should be minimized to reduce the power consumption.

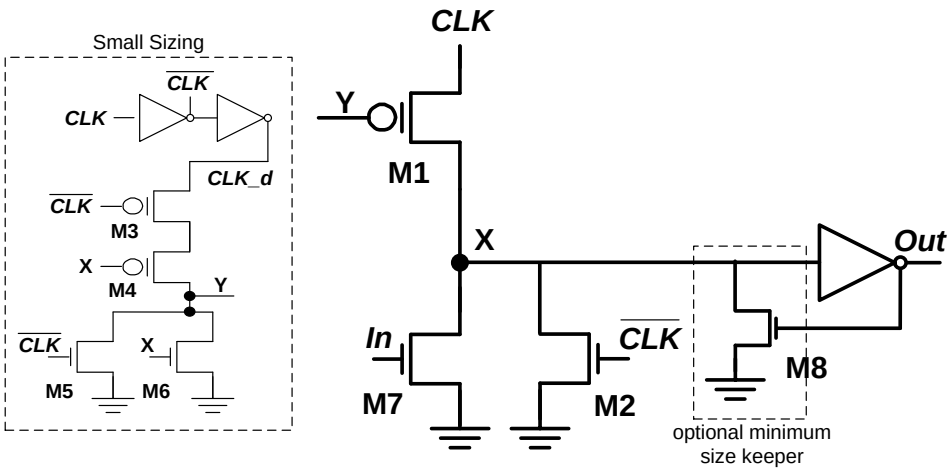


Figure 3.3: Schematic of a buffer with optional keeper implemented with CD logic

3.2 Noise Margin vs. Window Width

Noise margin is defined as the dc-noise level at the input generating a false logic evaluation at the output of the same gate and can be computed based on the following formula:

$$Noise\ Margin = |V_{original} - V_{noise}| \tag{3.1}$$

where $V_{original}$ is the expected voltage level without any input noise interference and V_{noise} is the input dc-noise voltage that causes the false logic evaluation. For CD logic, two types of noise margin are defined: Logic "1" and "0" noise margin. Logic "1" noise margin refers to the input dc-noise level that causes CD logic to fail to remain in the contention mode. Consider a buffer implemented in CD logic, as shown in Figure 3.3; if the input which is supposed to be at full VDD is now degraded due to noise, then the glitch level at node X may be too high such that Out is falsely discharged. In this case, the noise margin can be calculated as $1V - V_{in}$, where V_{in} is V_g of $M7$. Similarly, logic "0" noise margin refers to the input dc-noise level that causes CD logic to fail evaluating. In this case, if an input which is supposed to be at GND is now much higher due to noise, the contention between M1 and M7 will cause node X to settle at an intermediate voltage instead of VDD. When CLK_d rises to VDD (window closes), node Y will also be charged up through M3 and M4,

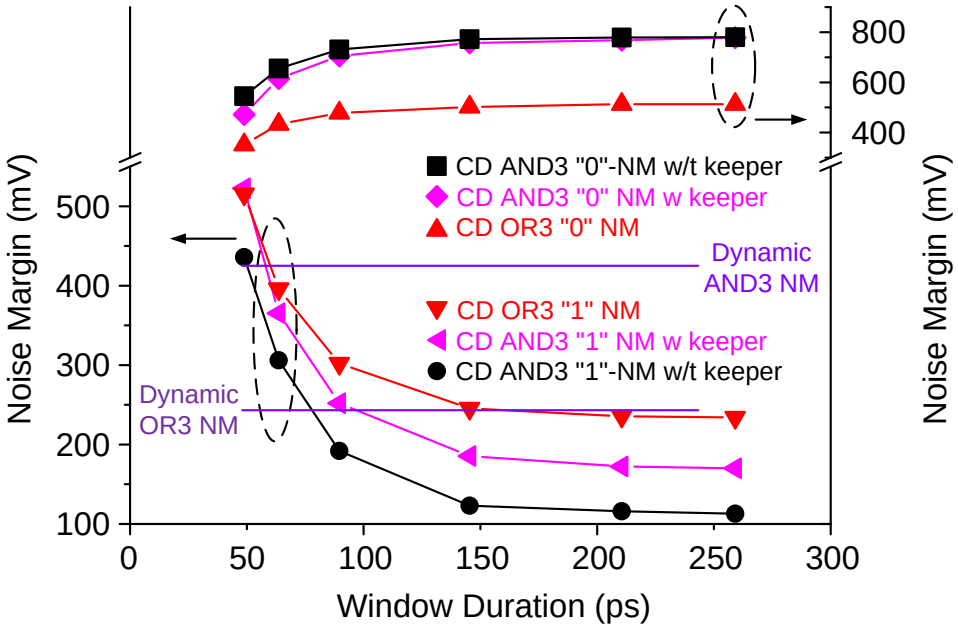


Figure 3.4: Simulated logic "1" and "0" noise margin vs. window duration for a 3-input AND gate and OR gate

since M3 is on and M4 is partially on because node X is not at VDD. If the voltage level at node X is too low, then node Y will be charged to VDD and node X will be discharged to GND through M7 which is driven by the noise source.

Figure 3.4 shows the simulated worse case logic "1" and "0" noise margin for a 3-input AND gate and OR gate implemented with CD and dynamic logic. CD logic's logic "0" noise margin is always much higher than logic "1" noise margin, suggesting that CD logic is more robust during C-Q delay and D-Q delay modes than contention mode. Moreover, as window width prolongs, logic "0" noise margin improves while logic "1" noise margin degrades for both logic types. Therefore, reducing the widow duration not only minimizes the power consumption but also improves CD logic's overall robustness[1]. Based on Figure 3.4, CD logic is not as robust as dynamic logic in a 3-input AND configuration. On the other hand, CD logic exhibits better noise margin than that of dynamic logic for a 3-input

[1]As long as logic "0" noise margin is higher than logic "1" noise margin, improving logic "1" noise margin is equivalent to improving CD logic's overall robustness.

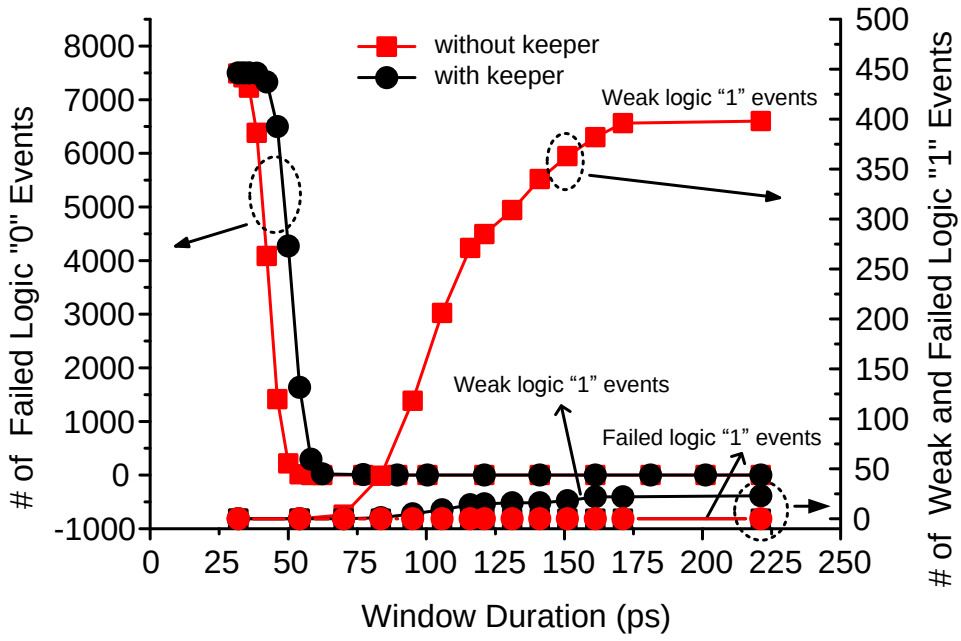


Figure 3.5: Number of weak and failed “1” and failed “0” events for a 3-input AND gate implemented in CD logic in a monte-carlo simulation with 7500 iterations.

OR configuration when the window width is less than 150ps. If CD logic is to be used in an application where additional noise margin is necessary, an optional minimum size NMOS keeper can be inserted to improve its overall robustness, as shown in Figure 3.4. The minimum size keeper improves logic “1” noise margin by approximately 60mV with virtually no degradation in logic “0” noise margin.

Figure 3.5 illustrates the number of weak and failed “1” and failed “0” events vs. window duration for a 3-input AND gate with and without a keeper in a monte-carlo simulation with 7500 iterations. Weak and failed logic “1” events are defined as the cases where the maximum temporary output glitch is more than 5% and 20% of the VDD respectively. Failed “0” event represents the case where the output fails to evaluate. Extremely short window duration (<50ps) leads to a majority of logic “0” evaluation failures, which corresponds to a poor logic “0” noise margin. As the window duration increases beyond 50ps, the number of failed logic “0” events quickly drops to zero while the number of weak logic “1” events begins to increase, corresponding to a decreasing logic “1” noise margin.

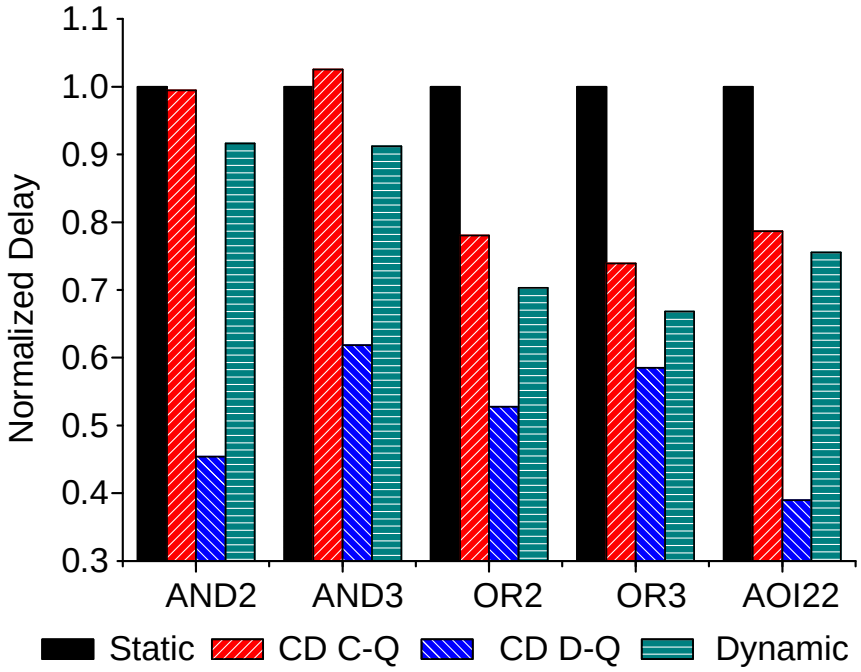


Figure 3.6: Normalized delay of five logic expressions implemented in static, dynamic, and CD logic

Despite the increased number of weak logic "1" events, the number of failed logic "1" events remains at zero even at window duration of 225ps. Moreover, the minimum size keeper effectively reduces the number of weak "1" events from 400 to less than 25 when the window width is at 225ps. These results closely resemble to Figure 3.4, and indicates that either over tightening or stretching the window duration can compromise CD logic's overall robustness.

3.3 Static, Dynamic, and CD Logic Performance Comparison

Figure 3.6 and 3.7 illustrate the normalized delay and average power of static, dynamic, and CD logic in five logic expressions at 50% and 100% input data activity. The average power is calculated by summing up the power consumption of every possible input vector

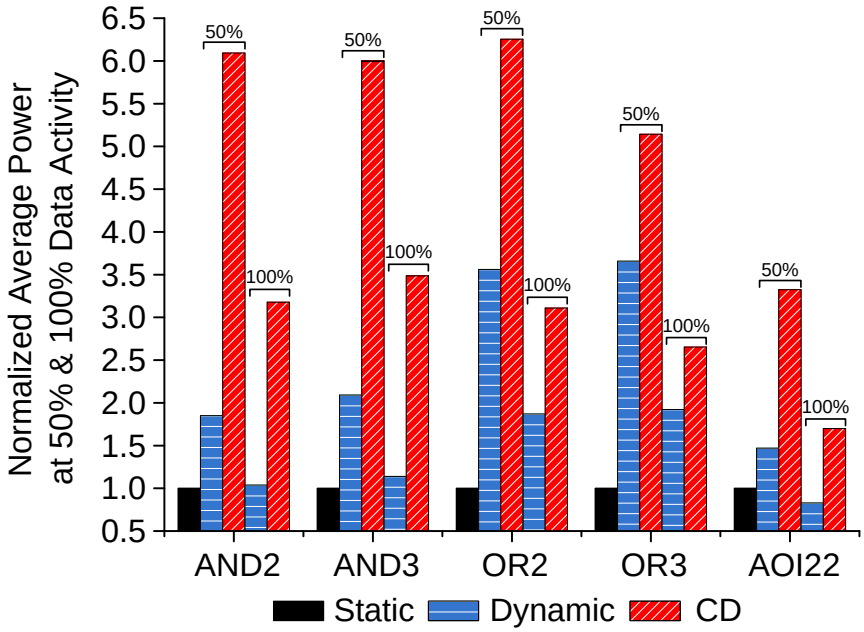


Figure 3.7: Normalized average power of five logic expressions at 50% and 100% data activity implemented in static, dynamic, and CD logic

then dividing by the number of input vector combinations. The data activity is defined as

$$data\ activity = \frac{\#\,of\ signal\ transitions}{\#\,of\ signals \times \#\,of\ clock\ cycles} \tag{3.2}$$

CD logic demonstrates superior performance, especially for complicated logic expressions, such as $Y = AB+CD$ (AOI22), in D-Q mode due to the pre-evaluated characteristic. On the other hand, CD logic's performance is only approximately the same or even worse than dynamic logic during C-Q mode. Therefore, it is advantageous to implement CD logic in a multiple-stage, single-cycle datapath since then can the pre-evaluated feature of CD logic be fully utilized. In this regard, designer needs to ensure that CD logic always operates in D-Q mode. Alternatively, C-Q mode should be avoided. The power consumption of CD logic at 50% data activity is at least 3X and 5X higher than that of static logic in AOI22 and the rest of logic expressions respectively. This suggests that CD logic should only be used to replace the timing critical path in any circuit block, i.e., carry propagation in an adder, since it is not energy efficient to implement any system with only CD logic.

Table 3.1: Area comparison of five logic expressions implemented with Static, Dynamic, and CD Logic

	Total Transistor Width (μm)		
	Static	Dynamic	CD
AND2	7	7.7	10.4
AND3	10.5	11.4	13.1
OR2	8	7.2	9.8
OR3	13.5	8.2	10.6
AOI22	15	10.7	12.5
Average	10.8	9.04	11.28

Five different logic expressions' total transistor width implemented in static, dynamic, and CD logic style are summarized in Table 3.1. Despite CD logic's additional transistors overhead, the average area of CD logic is 4.4% and 25% larger than that of static and dynamic logic respectively. For complex logic expression such as AOI22, CD logic occupies less area than static logic.

Chapter 4

8-Bit Ripple Carry Adders Analysis

As shown in the previous section, CD logic demonstrates speed advantage over static and dynamic logics only during D-Q mode, which only takes places in a single cycle, multi-stage system. In this chapter, three 8-Bit ripple carry adders (RCAs) using static, dynamic, and CD logic style are laid out and simulated to compare their performance.

4.1 Addition

Addition is the most frequently used arithmetic operation and is often the speed-limiting element of arithmetic logic unit in modern CPUs. Addition forms the basis for many processing operations, including counting, multiplication, and digital signal filtering. As a result, adder circuits that add two binary numbers are of great interest in the field of digital IC and are often used as a test bench to compare different logic style.

Table 4.2 shows the truth table of a binary full adder. A and B are the adder inputs, C_i is the carry input, S is the sum output, and C_{out} is the carry output. Based on this truth table, the boolean expression for S and C_{out} can be summarized as:

$$S = A \oplus B \oplus C_i = A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i + \bar{A}\bar{B}C_i + ABC_i \tag{4.1}$$

$$C_{out} = AB + BC_i + AC_i \tag{4.2}$$

Table 4.1: Full adder truth table

A	B	C_i	S	C_{out}	Carry Status
0	0	0	0	0	delete
0	0	1	1	0	delete
0	1	0	1	0	propagate
0	1	1	0	1	propagate
1	0	0	1	0	propagate
1	0	1	0	1	propagate
1	1	0	0	1	generate/propagate
1	1	1	1	1	generate/propagate

A full adder (FA) unit can be implemented with the above boolean equations using CMOS transistors. To construct a N-bit adder, N FA units can be cascaded in series, connecting C_{out} of a FA cell to the next stage FA C_i input. This configuration is known as ripple-carry adder, since the carry bit "ripples" from one stage to the other. A block diagram of a N-bit ripple carry adder (RCA) is shown in Figure 4.1. For a RCA, the worst case delay is the propagation of carry signal from the least significant carry bit (C_i) to the most significant carry bit. Furthermore, an additional stage is required to produce the sum based on this carry signal. In the case of RCA, the delay is then proportional to the number of bits in the input words N and is approximated by

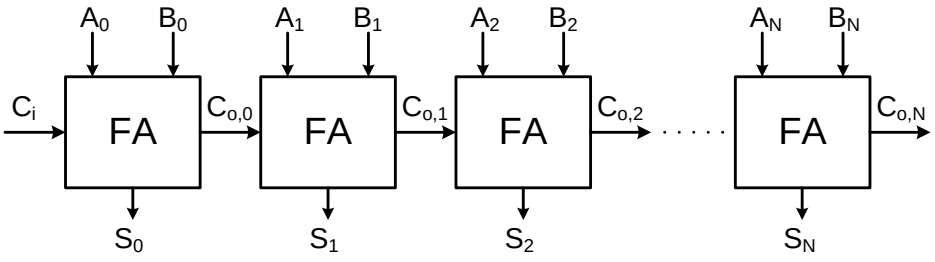


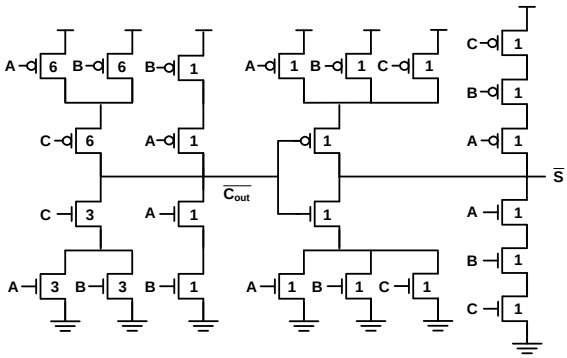
Figure 4.1: N-bit ripple carry adder block diagram

$$t_{adder} = (N-1)t_{carry} + t_{sum} \tag{4.3}$$

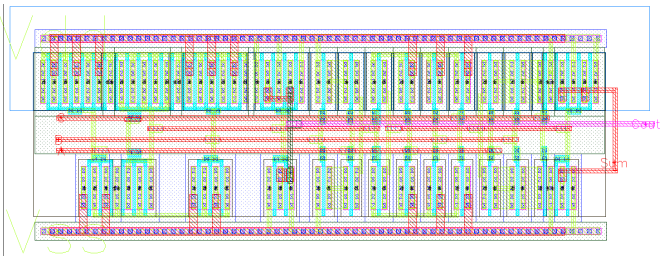
In the case of a 8-Bit ripply carry adder, the worse case delay is then $t_{adder} = 7t_{carry} + t_{sum}$.

4.2 8-Bit ripple carry adders

The basic full adder (FA) is implemented with an energy efficient, mirror structured 28T cell with sizing strongly favoured to compute C_{out}, as shown in Figure 4.2 [8]. PMOS and NMOS transistors in the carry generation are sized to be six times and three times larger than the sum generation transistors respectively. Since the critical path of RCAs



(a)



(b)

Figure 4.2: 28 transistors full adder cell (a) schematic and (b) layout implemented in static logic.

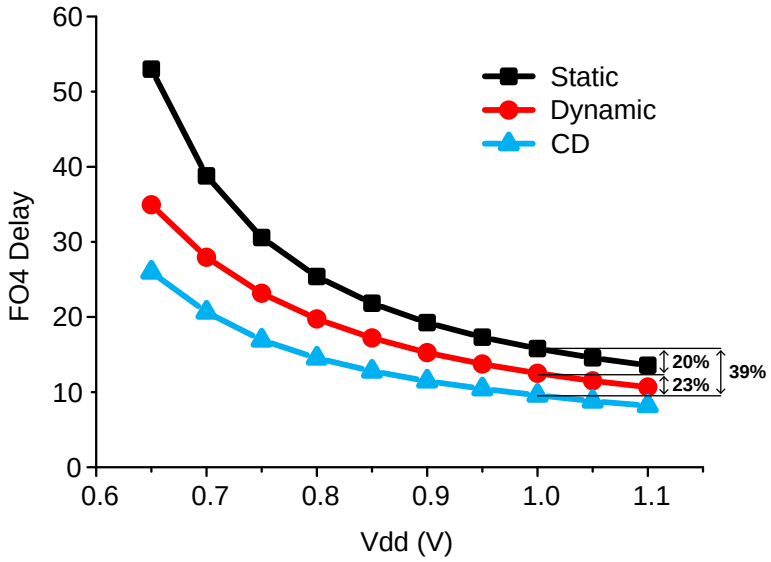


Figure 4.3: Normalized delay comparison of 8-Bit RCAs for various logic styles

goes from C_{in} to C_{out}, in dynamic and CD logic implementations only the carry generation circuitry is replaced while the sum computation remains static. During this experiment we observe that the parasitic capacitance contributed by both internal (within a cell) and global (stage to stage) interconnects are critical and can increase the total delay by up to 40% from schematic to post-layout simulation. Therefore, to ensure fair comparison we first laid out each RCA, then extracted the corresponding parasitic capacitance at every node, and finally back-annotated the extracted lump capacitance value to the schematic. Compared to post layout simulations, extensive studies reveal that schematic simulations with back-annotated capacitance achieve a result difference of less than 2% but are at least 3X faster in terms of simulation time. Therefore, all the data are generated from simulations with back-annotated lumped capacitance for the rest of the comparisons.

The test bench setup is similar to Figure 3.1. Clock and data frequencies are set to 500MHz to accommodate for the worse case static RCA delay at lower supply voltages. Similar to previous section, the average power consumption is measured with 50% data activity. Instead of calculating the average power by measuring the power consumption of every input vector (2^{16} combinations for 8-Bit RCA), we use an alternative power measurement scheme where the input vectors are generated by a 16-bit binary random number

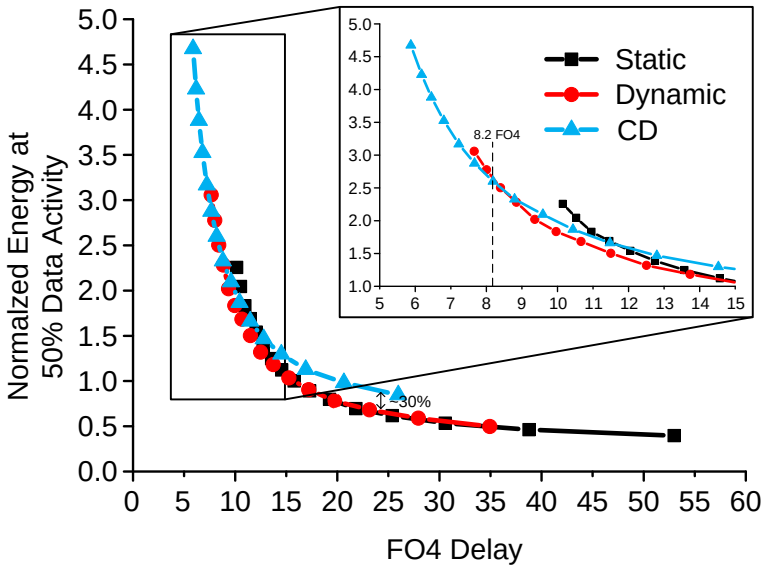


Figure 4.4: Normalized energy vs. delay curve of 8-Bit RCAs at 50% data activity

generator implemented in Verilog. We then conduct the simulation and constantly calculate the average power consumption until it converges to a stable value. Simulation results indicate that the power consumption of three RCAs quickly converges with less than 1% variation after 10000 random input vectors, which corresponds to 20μs simulation time. Therefore, all the average power measurements conducted in this section are simulated with at least 10000 random input vectors to ensure accurate and fair results. Similar to previous section, the delay is calculated from the 50% point of the rising edge of Data (static) or CLK (dynamic and CD) to the 50% point of the rising edge of the final output sum using the worst case delay vector. The clock timing is designed in such a way that all the CD logics except the first stage are operated in D-Q mode with window duration of approximately 115ps.

Normalized fanout of 4 (FO4) delay [30] comparison of 8-Bit RCAs implemented using static, dynamic, and CD logic style is illustrated in Figure 4.3. At the 1V nominal voltage, CD-based RCA is approximately 39% and 23% faster than the identical adder implemented using static and dynamic logic respectively. As the supply voltage is varied from 0.65V to 1.1V, CD-based RCA achieves approximately the same performance advantage over the other two adders. Figure 4.4 illustrates the normalized energy (power $\times$ total simulation

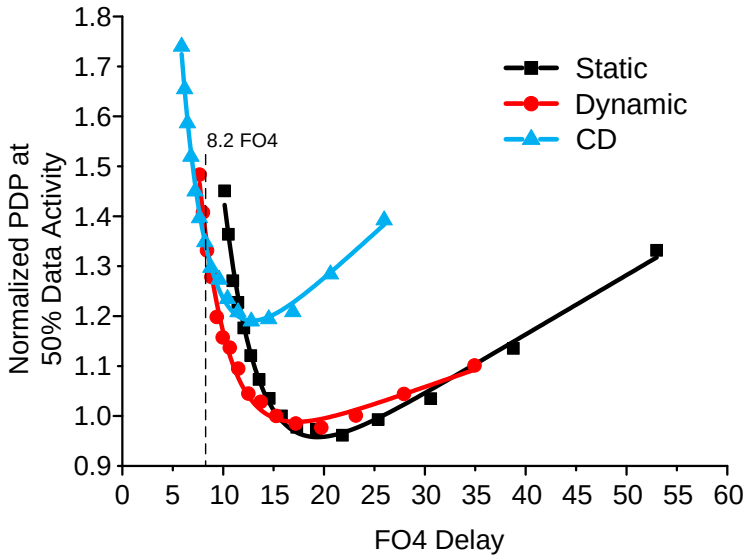


Figure 4.5: Normalized power-delay product of 8-Bit RCAs vs. delay at 50% data activity

time) vs. delay tradeoff curve for the three RCAs. Clearly, if longer delay can be tolerated, static and dynamic are the more energy efficient design. However, CD logic becomes more energy efficient when performance constraint is set at 8.2 FO4. For a delay constraint of 7.5 FO4 delay, CD logic is roughly 20% more energy efficient than dynamic logic, while the static-based adder cannot reach this delay constraint. CD logic provides a speed advantage that logic styles such as static and dynamic are difficult to reach. Therefore, CD logic is suitable in a system where performance is the most critical factor. In the current setup, the critical path accounts for approximately 50% of the entire circuitry. It is expected that CD logic will demonstrate more energy efficiency in a more complicated system with a unique critical path that accounts for smaller portion of the entire circuitry, i.e., carry generation in a 64-bit parallel prefix adder.

Figure 4.5 shows the normalized power-delay product (PDP, power $\times$ delay) vs. delay curve for the three RCAs. Contrary to the energy vs. delay curve in Figure 4.4 where energy continue to decrease as delay increases, the minimum PDP point for all three configurations lies in between 10 to 20 FO4 delay. This is because at longer delay constraint, the increment in delay dominates the PDP calculation. From Figure 4.5, CD-based RCA becomes more PDP efficient than the static and dynamic-based RCAs at delay constraint of 13 FO4 and

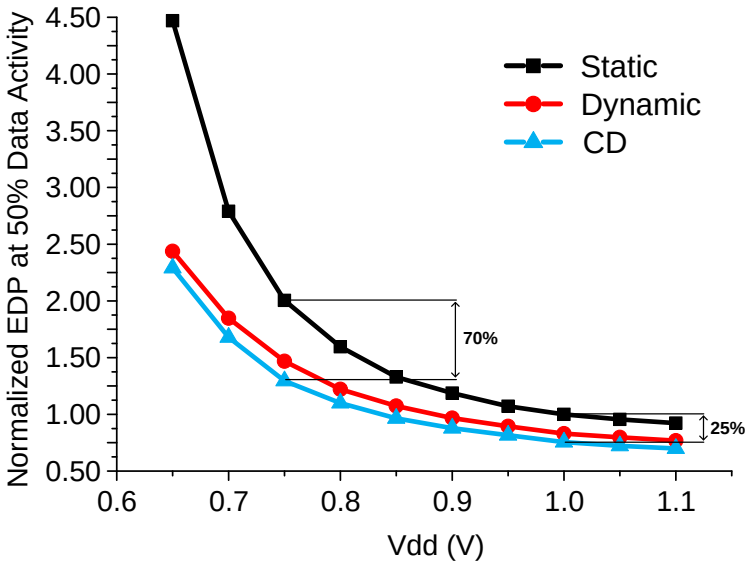


Figure 4.6: Normalized energy-delay product of 8-Bit RCAs vs. supply voltage

8.2 FO4 respectively. At longer delay targets, CD-based RCA's PDP is worse than that of the other two designs.

Figure 4.6 illustrates the normalized energy-delay product (EDP) vs. supply voltage curve. CD-based RCA's EDP is less than that of static and dynamic-based RCAs across all supply voltages. At the supply voltage of 1V and 0.75V, CD-based RCA is 25% and 70% more EDP efficient than static-based RCA respectively. Figure 4.7 shows the normalized variation of delay and power consumption for the three RCAs at different temperature with tNtP corner. The normalized variation is calculated by dividing each design's simulated delay and power results at different temperature to its simulated results at 20°C. As the temperature varies from -20°C to 120°C, CD-based RCA shows lower delay sensitivity with a maximum deviation of 2%. In terms of power sensitivity, CD-based adder lies in between static and dynamic-based RCAs with a maximum deviation of 18%. Finally, CD-based RCA evaluates correctly under all temperature conditions.

Figure 4.8 illustrates the normalized glitch mean and σ of CD-based 8-Bit RCA employing condition (2.1) vs. different supply voltages at 110°C in a monte-carlo simulation with 7500 iterations. Two observations can be drawn from Figure 4.8: i) a CD-based

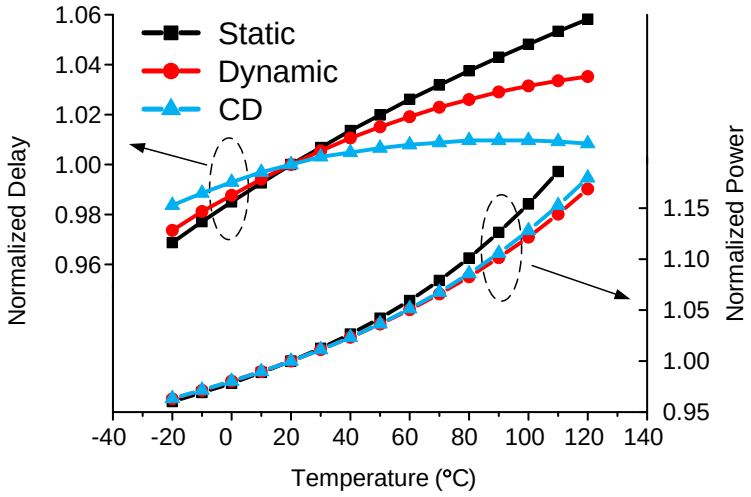


Figure 4.7: Normalized delay and power consumption variation of 8-Bit RCAs at different temperature

circuitry which is designed to satisfy the glitch constraint at 1V is not sufficient since at higher supply voltage (1.05V and 1.1V), the glitch average is increased to 5.2% and 5.8% respectively and ii) designers only need to ensure that the target glitch specification is met at the maximum supply voltage that the circuit operates at. Finally, Table 4.2 summarizes the delay and power consumption of three RCAs vs. different corners with worst case delay vector at 110°C. CD-based RCA successfully evaluates under all conditions.

4.3 Summary

In this section, 8-Bit ripple carry adders implemented with static, dynamic, and CD logic are laid out and analyzed. CD-logic based RCA shows superior performance and energy-delay product across all supply voltages. When performance is the primary concern, i.e., the delay of the RCA has to been less than 7.5 FO4 delay, CD-based design is 20% more energy efficient than dynamic-based design while static-based design cannot reach this constraint. In-depth analysis reveals that the unwanted temporary glitch associated with CD-logic only aggravates as supply voltage scales up; therefore, when designing CD-based circuitry designers only need to ensure that the target glitch specification is satisfied at the maximum

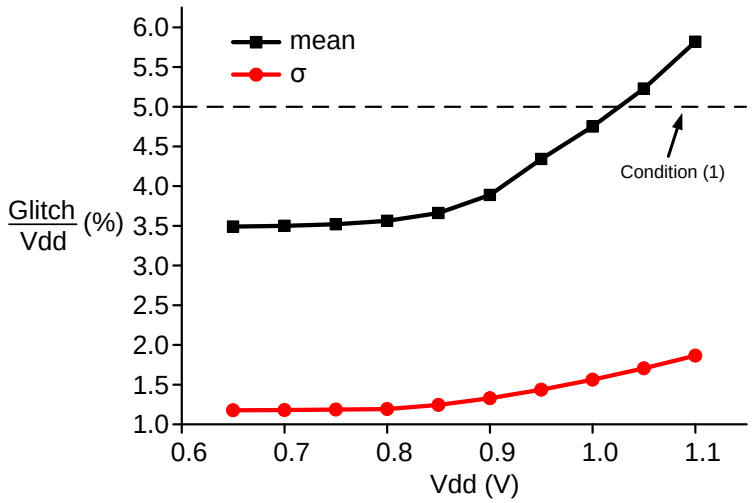


Figure 4.8: Normalized glitch mean and standard deviation of CD-based 8-Bit RCA with respect to VDD at different supply voltage at 110°C using monte-carlo simulations with 7500 iterations.

supply voltage that the circuit will operate at. Finally, CD logic exhibits similar delay and power variation compared to static and dynamic logic across different temperature levels and successfully evaluates under all corners at the worst case temperature of 110°C.

Table 4.2: Power and delay comparison of three RCAs vs. different corners with worst case delay input vector at 110°C

	Static		Dynamic		CD	
Corner	delay(ps)	power(uW)	delay	power	delay	power
tNtN	369	413	293	376	228	477
fNfP	278	453	216	420	168	525
fNsP	320	440	244	408	212	524
sNfP	314	446	251	409	182	508
sNsP	484	404	389	359	308	469

Chapter 5

Conclusion

In this thesis, we describe the evolution of constant delay type logic and have proposed a new high performance logic style with constant delay characteristic and self-reset circuitry. The pre-evaluated feature of this logic style makes it particularly suitable in a circuit block where a unique critical path exists and performance is the primary concern. Several advantages of CD logic over previously proposed feedthrough type logic styles have been explored, including better noise margin, ability to cascade multiple stages to perform complicated logic evaluation, and reduced power consumption with the option of optimizing individual gate's window width via a local timing window technique.

Design issues of appropriate timing window width adjustment to reduce power consumption without sacrificing performance is discussed and analyzed. Since CD logic's power consumption is directly proportional to the window duration, this window width should be as small as possible. However, gates with higher stack height such as AND gate should employ longer window width to achieve maximum performance potential. Also, over tightening or stretching the window width can reduce logic "0" and "1" noise margin respectively, thus compromising CD logic's overall robustness.

The performance advantage of CD logic has been demonstrated in five different logic expressions compared to static and dynamic logic. In D-Q mode, CD logic achieves an average speed up of 94% and 58% with an average area overhead of 4.4% and 25% respectively. The excessive power consumption of CD logic as a result of inevitable direct path

current during the contention mode indicates that CD logic should only be used to replace a system's critical path, since implementing an entire circuit block with purely CD logic is not energy efficient.

8-Bit ripple carry adders are implemented and laid out using static, dynamic, and CD logic style to further demonstrate CD logic's performance advantage. Simulation results indicate that CD-based RCA is more energy and PDP efficient than the other two configurations at shorter delay targets. At the 1V nominal VDD, CD-based RCA is 39% and 23% faster than static and dynamic-based RCA respectively. Moreover, CD-based RCA is 25% and 70% more EDP efficient than static-based RCA at the 1V and 0.75V supply respectively. Finally, as long as the targeted glitch specification is satisfied at the maximum supply voltage that the circuit is operating at, this specification will also be fulfilled at all lower supply voltage configurations.

The performance discrepancy between dynamic and static logic is less obvious in next generation CMOS process, i.e, 45nm and below, because interconnect capacitance instead of drain capacitance begins to dominate the delay. However, it is expected that CD logic will remain similar performance enhancement in the future since the improved performance is mainly contributed from both the constant number of transistors in the critical path and the pre-evaluated characteristic. Furthermore, the excessive leakage problem which begins to dominate the overall power consumption in deep-submicron technologies will also help CD logic since CD logic is in a disadvantageous position compared to other logic styles primarily in terms of dynamic power consumption. Hence, in advanced processes such as 45nm and below CD logic is expected to exhibit additional energy efficiency compared to its counterparts. Because of its high performance capability, CD-Logic is particularly suitable for high performance digital application, such as high speed parallel-prefix adder in a high performance arithmetic logic unit in a modern CPU.

Bibliography

[1] "International Technology Road Map for Semicondcutors 2001," May 2010 [Online]. Available: `http://www.itrs.net/teams.html`. ix, 9

[2] N. Verma and A. Chandrakasan, "A 65nm 8T Sub-Vt SRAM Employing Sense-Amplifier Redundancy," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp. 328 –606, 11-15 2007. 3

[3] M. Sinangil, N. Verma, and A. Chandrakasan, "A Reconfigurable 8T Ultra-Dynamic Voltage Scalable (U-DVS) SRAM in 65 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 44, pp. 3163 –3173, nov. 2009. 3

[4] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *Solid-State Circuits, IEEE Journal of*, vol. 32, pp. 1079 –1090, July 1997. 15

[5] R. Krambeck, C. Lee, and H.-F. Law, "High-speed compact circuits with CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 17, pp. 614 – 619, June 1982. 17

[6] N. Goncalves and H. De Man, "NORA: a racefree dynamic CMOS technique for pipelined logic structures," *Solid-State Circuits, IEEE Journal of*, vol. 18, pp. 261 – 266, June 1983. 17

[7] V. Friedman and S. Liu, "Dynamic logic CMOS circuits," *Solid-State Circuits, IEEE Journal of*, vol. 19, pp. 263 – 266, April 1984. 17

[8] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective (3rd Edition)*. Addison Wesley, 3 ed., May 2004. 17, 18, 31, 44

[9] C. Lee and E. Szeto, "Zipper CMOS," *IEEE Circuits and Systems Magazine*, pp. 10 – 16, May 1986. 17

[10] K. Bernstein, K. Carrig, C. M. Durham, P. R. Hansen, D. Hogenmiller, E. J. Nowak, and N. J. Rohrer, *High Speed CMOS Design Styles*. Springer, 1st ed., Aug. 1998. 17

[11] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy Delay Optimization of 64-Bit Carry-Lookahead Adders With a 240 ps 90 nm CMOS Design Example," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 2, pp. 569–583, 2009. 17, 18

[12] S. K. Mathew, R. K. Krishnamurthy, M. A. Anders, R. Rios, K. R. Mistry, and K. Soumyanath, "Sub-500-ps 64-b ALUs in 0.18μm SOI/bulk CMOS: design and scaling trends," *Solid-State Circuits, IEEE Journal of*, vol. 36, no. 11, pp. 1636–1646, 2001. 17

[13] S. Mathew, M. Anders, R. Krishnamurthy, and S. Borkar, "A 4 GHz 130 nm address generation unit with 32-bit sparse-tree adder core," in *VLSI Circuits Digest of Technical Papers, 2002. Symposium on*, vol. dders high-speed integrated circuits microprocessor chips, pp. 126–127, 2002. 17

[14] S. K. Mathew, M. A. Anders, B. Bloechel, N. Trang, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 1, pp. 44–51, 2005. 17

[15] S. Wijeratne, N. Siddaiah, S. Mathew, M. Anders, R. Krishnamurthy, J. Anderson, S. Hwang, M. Ernest, and M. Nardin, "A 9GHz 65nm Intel Pentium 4 Processor Integer Execution Core," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pp. 353–365, 2006. 17

[16] I. Sutherland, R. F. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, 1st ed., Feb. 1999. 17

[17] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and K. Ram, "Energy-delay estimation technique for high-performance microprocessor VLSI adders," in *Computer*

Arithmetic, 2003. Proceedings. 16th IEEE Symposium on, vol. dders delay estimation 0.10 micron 0.13 micron 32 bit 64 bit CMOS technology VLSI adder topology energy-delay space estimation energy-delay trade-off, pp. 272–279–, 2003. 18

[18] S. Kiaei, S.-H. Chee, and D. Allstot, "CMOS source-coupled logic for mixed-mode VLSI," pp. 1608 –1611 vol.2, May 1990. 18

[19] J.-O. Plouchart, J. Kim, V. Karam, R. Trzcinski, and J. Gross, "Performance Variations of a 66GHz Static CML Divider in 90nm CMOS," pp. 2142 –2151, Feb. 2006. 19

[20] H. Sarbishaei and M. Sachdev, "ESD protection circuit for 8.5Gbps I/Os in 90nm CMOS technology," pp. 697 –700, Sept. 2009. 19

[21] Z.-D. Huang, C.-Y. Wu, and B.-C. Huang, "Design of 24-GHz 0.8-V 1.51-mW Coupling Current-Mode Injection-Locked Frequency Divider With Wide Locking Range," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 57, pp. 1948 –1958, Aug. 2009. 19

[22] L. Rodoni, G. von Buren, A. Huber, M. Schmatz, and H. Jackel, "A 5.75 to 44 Gb/s Quarter Rate CDR With Data Rate Selection in 90 nm Bulk CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 44, pp. 1927 –1941, July 2009. 19

[23] V. Navarro-Botello, J. A. Montiel-Nelson, and S. Nooshabadi, "Low Power Arithmetic Circuit in Feedthrough Dyanmic CMOS Logic," *49th IEEE International Midwest Symposium on Circuits and Systems*, vol. 1, pp. 709–712, 2006. 19, 22

[24] V. Navarro-Botello, J. A. Montiel-Nelson, and S. Nooshabadi, "Analysis of High-Performance Fast Feedthrough Logic Families in CMOS," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 54, no. 6, pp. 489–493, 2007. 19, 22

[25] P. Chuang, D. Li, and M. Sachdev, "Design of a 64-bit low-energy high-performance adder using dynamic feedthrough logic," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 3038–3041–, 2009. 20, 24, 29

[26] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Springer, 1 ed., June 1995. 31

[27] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits : a design perspective*, vol. 2nd. Upper Saddle River, N.J.: Pearson Education, 2003. 32

[28] N. Kurd, J. Barkarullah, R. Dizon, T. Fletcher, and P. Madland, "A multigigahertz clocking scheme for the Pentium(R) 4 microprocessor," *Solid-State Circuits, IEEE Journal of*, vol. 36, pp. 1647 –1653, Nov. 2001. 32

[29] S. Vangal, N. Borkar, E. Seligman, V. Govindarajulu, V. Erraguntla, H. Wilson, A. Pangal, V. Veeramachaneni, M. Anders, J. Tschanz, Y. Ye, D. Somasekhar, B. Bloechel, G. Dermer, R. Krishnamurthy, S. Narendra, M. Stan, S. Thompson, V. De, and S. Borkar, "5GHz 32b integer-execution core in 130nm dual-VT CMOS," vol. 2, pp. 334 –535, 2002. 32

[30] M. Horowitz, "VLSI Scaling for Architects," *Presentation slides, Computer Systems Laboratory, Stanford University*, April 2010 [Online]. Available: `http://www-vlsi.stanford.edu/papers/VLSIScaling.pdf`. 46