UNIVERSITY OF CALIFORNIA
SANTA CRUZ

CURRENT-MODE CLOCKING AND SYNTHESIS CONSIDERING LOW-POWER AND SKEW

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

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Riadul Islam

September 2017

The Dissertation of Riadul Islam
is approved:

Professor Matthew R. Guthaus, Chair

Professor Jose Renau

Professor Ken Pedrotti

Dean Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Dedicated to my parents,

Abul Farrah and Hena Farrah

Abstract

Current-Mode Clocking and Synthesis Considering Low-Power and Skew

by

Riadul Islam

Over the past decade, power associated with the Clock Distribution Network (CDN) has played an increasingly important role in the global integrated circuit industry. Since Complementary Metal Oxide Semiconductor (CMOS) technology continues to shrink, new physical phenomena are added to device/transistor behaviour. However, less attention has been given to add more features to the interconnect materials.

In order to reduce the power associated with interconnect, researchers introduced some efficient low power techniques like low-swing clock signaling, clock gating, and resonant energy recovery clocking. Another very attractive signaling scheme, namely Current-Mode (CM) signaling, can save significant power while maintaining high-frequency operation. However, a true CM clocking methodology for local and global CDNs has not been explored.

I propose a new paradigm for clock distribution that uses current, rather than voltage, to distribute a global clock signal with reduced power consumption. While CM signaling has been used in one-to-one signals, this is the first usage in a one-to-many CDN. To accomplish this, I create a new high-performance current-mode pulsed flip-flop with enable (CMPFFE) using a representative $45nm$ CMOS technology. When the CMPFFE is combined with a CM transmitter, the first CM clock distribution network exhibits 45.2% lower average power compared to traditional voltage-mode (VM) clocks.

In addition, I propose the first CM clock synthesis (CMCS) methodology to reduce overall clock network power with low skew. The method can integrate with traditional clock routing followed by transmitter and receiver sizing. I validate the proposed methodology using ISPD 2009 and 2010 industrial benchmarks. This methodology saves $39 - 84\%$ average power with similar skew on the benchmarks using $45nm$ CMOS technology simulation of clock frequencies range from 1-3GHz. In addition, the CMCS methodology takes $2.4 - 9.1\times$ less running time and consumes $20 - 26\%$ less transistor area compared to synthesized, buffered VM clock distributions.

Acknowledgments

First of all, I would like to thank Professor Matthew R. Guthaus, my advisor, for his guidance, enthusiasm, and constant encouragement throughout the years of my doctoral study. From him I learned not only the need for profound research, but also the importance of looking at the bigger pictures. His insights have helped me greatly in several crucial stages of this research. I would like to thank my committee members, Professor Jose Renau and Professor Ken Pedrotti for reviewing this thesis and providing critical suggestions. I would also like to thank all fellow colleagues of our VLSI-DA research group, particularly, Rajsaktish Sankaranarayanan, Ping-Yao Lin, Hany Fahmy, Brian Chen, Rebecca Rashkin, Jeff Butera, Benjamin Lacara, and Jie Zhang for both technical discussions and all sorts of fun discussions. I am very grateful to my colleague Hunter Nichols for proofreading my thesis.

It has been a long and winding 7 years since I left my home country and embarked on the journey of further educating myself. This thesis to a certain extent provides an evidence of these efforts. I would like to thank all my siblings, Abu Hena Mostafa Kamal, Shahana Akhter, and Thouhidul Islam for always inspiring me. I am thankful to my brother's wife for their encouragements. I am deeply indebted to my parents, Abul Farrah and Hena Farrah for their love and moral support. Most importantly, I thank them for pointing me in this direction. I am forever indebted to all my family members for sending me all the beautiful gifts all the way from Bangladesh.

During my Ph.D. study I was blessed with a baby girl (Mahveen Riad), I thank her for being my inspiration. After her birth she became one of the main source I could recharge myself and work harder on my research.

Finally, but mostly, I would like to thank my wife, Mohana Mahfuz Prema, who more than anyone has given me the strength to finish this work. I thank her for the understanding, tolerance and the much needed inspiration.

This work was supported in part by the National Science Foundation under grant CCF-1053838.

Chapter 1

Introduction

Portable electronic devices require long battery lifetimes which can only be obtained by utilizing low-power components. Recently, low-power design has become quite critical in synchronous Application Specific Integrated Circuits (ASICs) and System-on-Chips (SOCs) because the because the on-chip wires or interconnect in scaled technologies are consuming an increasingly significant amount of power. Researchers have demonstrated that the majority of this power is consumed by global buses, Clock Distribution Network (CDN), and synchronous signals in general [28]. For example, the microprocessor dissipates 40%-70% of its total power in its clock distribution network and latches [2,58].

In addition to power, interconnect delay poses a major obstacle in high frequency operation. Technology scaling reduces transistor and local interconnect delay while adversely affecting global interconnect delay [43]. Moreover, the conventional structures of CDNs are becoming increasingly difficult for multi-GHz Integrated Circuits (ICs) because skew, jitter, and variability are often proportional to large latencies [23,26].

Since the CDN consumes significant power, can limit maximum operation frequency and can even cause functional failure, there has been a great deal of work on optimization of clock distributions including resonant clocking [12,22,29,48,65]. There has also been an equally impressive amount of work on different circuit designs or signaling techniques [16,18,28,43,45,66]. Among different optimization techniques, resonant energy recovery clocking is very promising. However, its implementation to microprocessors is limited due to the size of inductor.

Prior to and in early Complementary Metal-Oxide Semiconductor (CMOS) technologies, Current-Mode (CM) logic was an attractive high-speed signaling scheme [5, 78]. CM logic, however, consumes significant static power to offer these high speeds. Because of this, standard CMOS Voltage-Mode (VM) signaling has been the *de facto* standard logic family for several decades.

Low-swing and current-mode signaling, however, are highly attractive solutions to help address the interconnect power and variability problems [18, 28, 33, 35, 36, 43, 66, 80]. In global CM interconnect, the static power is often significantly less than VM dynamic power and latency is significantly improved over VM. CM signaling schemes also offer higher reliability since they are less susceptible to single-event transient upsets due to the absence of buffers with source/drain diffusion areas that can be hit by high-energy particles.

Previous CM schemes have been used for long global wires or, more commonly, off-chip signals [82]. Standard logic signals, however, have remained VM to benefit from the low static power of CMOS logic. However, it is not practical to make each individual point-to-point segment of the CDN CM, but the clock signal should still benefit from the power and reliability of CM signaling. In this thesis, the power savings is maximized by creating a high-fanout symmetric or near-symmetric clock distribution that feeds many CM Flip-Flop (FF) receivers. Logic signals on the FF receivers retain VM compatibility with low-power VM CMOS logic in the remainder of the chip.

1.1 Thesis Contributions and Outline

This thesis focuses on various issues associated with synchronous Very-Large-Scale Integration (VLSI) interconnection and CDN for high-speed applications. These include parasitic resistance and capacitance, power, skew, crosstalk, and electromigration. The chapters of this thesis are structured such that they can be read independently of one another, while, at the same time, being part of one coherent unit. The brief description of the content of the six chapters of this thesis are as follows:

Chapter 1: Introduction – Problem Statement and Research Motivation
In this Chapter, I present the bottleneck of present synchronous VLSI issues, particularly associated with clock networks. I introduce the motivation of this research and a possible solution to the existing problem.

Chapter 2: Background – Preliminary Concepts of Interconnect Parasitics and Different Signaling Schemes

In this Chapter, I present the interconnect trend, wire parasitics that come with interconnect (resistance, capacitance). I also introduce wire parasitic analytical expressions and traditional wire models. In order to measure the efficiency of different signaling scheme at different frequencies, I also present analysis using a single test network applying different signaling schemes.

Chapter 3: Current-Mode Clocking – Suitable for Symmetric Clock Networks

In this Chapter, I present the existing CM clocking schemes with the pros and cons of these schemes considering architecture, power, performance, and noise robustness. I introduce the first CM Pulsed Flip-Flop with Enable (CMPFFE) as Receiver (Rx) circuit, and pulsed current Transmitter (Tx) . I also introduce the previous high-performance VM FFs and compare with the proposed CMPFFE considering the different aspect of FF. The result of this chapter motivates the need to introduce an algorithm to work with CM clocking.

Chapter 4: Differential Current-Mode Clocking – Suitable for Symmetric Clock Networks

In this Chapter, I present the existing differential CM clocking schemes along with their benefits and pitfalls considering architecture, power, performance, and noise robustness. I then introduce the first Differential CM Pulsed Flip-Flop (DCMPFF), and Differential Pulsed Current Tx (DPCTx). I also present the differential CM clock distribution by integrating DCMPFF and DPCTx.

Chapter 5: CMCS – Current-Mode Clock Synthesis

In this Chapter, I address the CM clock routing problem. I first present a new methodology to ensure the proper functionality of CM clocking in an asymmetric network and improve the global clock skew applying in an example network. I then introduce the proposed Current-Mode Clock Synthesis (CMCS) methodology. I also compare the proposed CM methodology with the existing VM methodologies to show the effectiveness of the proposed scheme.

Chapter 6: Conclusion – Major Contributions and Future Work

In this chapter, I summarize the major contributions of this dissertation and the possible direction of research to extend this work.

Chapter 2

Background

The modern digital electronic system emerged with the invention of the transistor in the year 1947 at Bell Laboratories. Next, the bipolar junction transistor was devised in 1949 which led to the first bipolar logic gate in 1956. After two years, Jack Kilby created the first bipolar IC at Texas Instrument. However, due to the large power consumption associated with the bipolar transistor family, Metal-Oxide Semiconductor Field-Effect Transistor (MOSFET) took over. As a result of low-power and scalability, the modern semiconductor industry progressed based on CMOS technology.

Due to the advancement of CMOS technology, the density of transistor and speed of integrated circuits have gone through a tremendous revolution in the last five decades [59]. The silicon industry follows Moore's law where the number of transistors integrated in a single die grows exponentially [51]. The huge number of transistors in a die have lead the silicon industry to integrate all the digital, analog, and data communication modules in a single chip. However, most SOCs are interconnect limited since interconnect does not scale as well as the transistors. Hence, designing interconnect with optimal power and performance budget has become very critical for modern microprocessors.

The modern Central Processing Unit (CPU) that has defined the performance of a computer for many years is facing several challenges. These bottlenecks can be identified as a memory bottleneck (the bandwidth of the channel between the computer's memory and CPU), the power wall (the chips overall temperature handling capacity and power consumption) the Instruction Level Parallelism (ILP) wall (the availability of enough discrete parallel instructions for a multi-core chip). The latter bottleneck is mostly dependent on the computer's Instruction

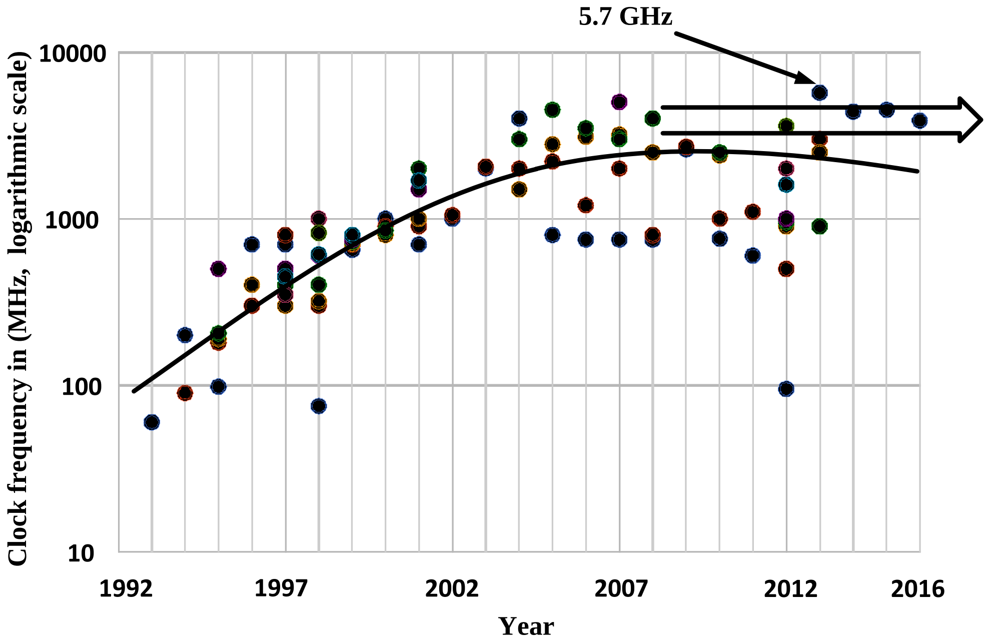


Figure 2.1: Due to the bottleneck of power wall, battery capacity and the existing cooling system the moderen microprocessor clock frequency is settled down in the range of 3-5GHz [37].

Set Architecture (ISA) and the availability of the resources and beyond the scope of this thesis. However, the earlier two bottlenecks are depended on the low-level design issues (circuits, interconnect). If we closely observe the clock frequency trend of the microprocessor in the last twenty-four years, we can easily predict that in the upcoming years the processor speed will settle down in the frequency range of 3-5GHz as shown in Figure 2.1 [37]. This bandwidth limitation is largely because of overall power consumption of the system and the available cooling system. The situation is even more critical for portable, mobile devices, such as laptops, tablets, and phones, where the power wall hits much earlier. This is primarily due to the constraints of battery capacity and limited or often fanless cooling.

2.1 Interconnect Trends

In order to ensure cost-effective advancements in the performance of integrated circuits, the International Technology Roadmap for Semiconductors (ITRS) projects the CMOS technology advancement as shown in Table 2.1.

Due to the increasing number of devices and die size, the ITRS projected the require-

Table 2.1: ITRS projects the CMOS technology advancement in terms of supply voltage, transistor size, and metal pitch scaling [3].

$Calendar\ year$	2013	2016	2020	2024	2026
$Interconnect\ one\ half\ pitch\ (nm)$	26.76	18.92	11.92	7.51	6.0
$MOSFET\ physical\ gate\ length\ (nm)$	20.17	15.34	10.65	7.4	6.2
# $of\ interconnect\ levels$	13	13	14	15	16
$Supply\ voltage\ (V)$	0.85	0.77	0.68	0.61	0.57
$Power\ index\ (\frac{W}{GHz-cm^2})$	$2-2.4$	$2.1-2.7$	$2.5-3$	$2.7-3.5$	$3.1-4$

ment of 12-16 levels of metal interconnect. However, the number of metal layers has not grown much past 10-12. This is mostly due to the overhead of vias in lower metal layers. As a result, the interconnect delay often exceeds the gate delay. On the other hand the total interconnect power can vary depending on the application of a system. Figure 2.2 shows the different power breakdown of microprocessor and an Field-Programmable Gate Array (FPGA) design [58]. Clearly, from Figure 2.2, interconnect and clocks constitute majority of the power budget for each class of devices.

Generally, interconnect scaling is broken down into global and local scaling. According to the ITRS report in 2012, the deep-submicron (less than or equal to the 16nm node) IC industry is facing the following design issues/challenges with interconnect [3]:

- The rapid introduction of new materials/processes to meet high-conductivity and low-dielectric permittivity requirements.
- Manufacturable integration including integration complexity, Chemical Mechanical Planarization (CMP) damage, resist poisoning, dielectric constant degradation.
- New chip reliability issues with new materials and processes.
- Three dimensional control of interconnect features.

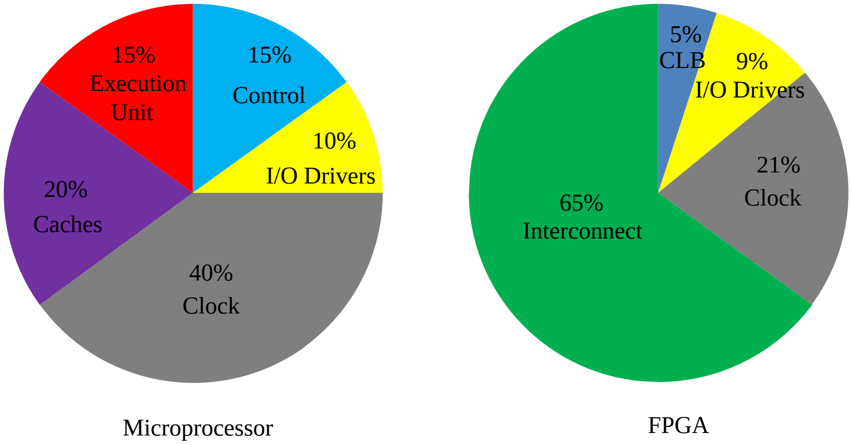


Figure 2.2: Interconnect and clocks constitute majority of the dynamic power budget for microprocessor and FPGA [58].

- Manufacturability and defect management that meet overall cost/performance requirements.

2.1.1 Local Interconnect Trends

In general, local interconnect are the lowest level of interconnects. Local interconnect can be defined as wires that span within a functional block/unit [69]. They usually connect transistor gates, sources, drains, and bodies in a CMOS technology. Since local interconnects do not travel too far, they can be smaller and can afford high resistivity.

Figure 2.3 shows the representative diagram of closely packed interconnects in an IC [62]. The parasitic resistance of an interconnect of dimensions $(length = L, width = W, height = H)$ can be written as

$$R = \rho\frac{L}{A} = \rho\frac{L}{WH} = R_{\Box}\frac{L}{W}$$

where ρ is the resistivity of the material, A is the cross-sectional area of the interconnect, and $R_{\Box}$ is the resistance per square. In general, we calculate two components associated with inter-

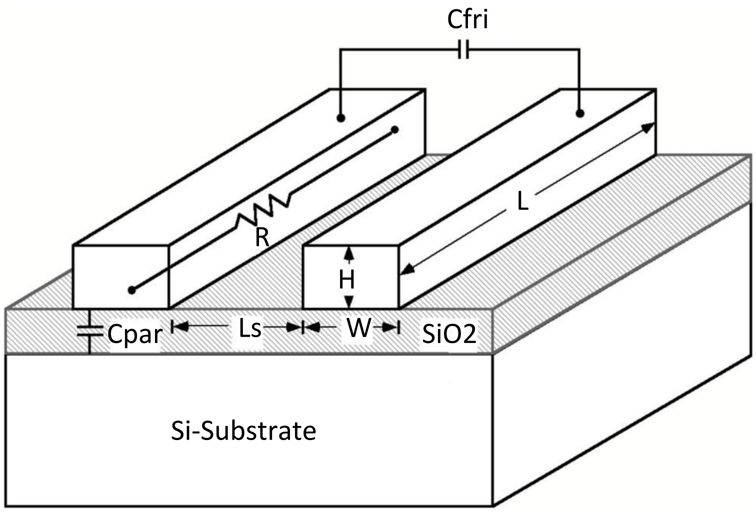


Figure 2.3: Large density of interconnects introduce fringing capacitance (C_{fri}) in ICs [62].

connect parasitic capacitance. The parallel plate capacitance can be realized by

$$C_{par} = K_{ox}\varepsilon_o \frac{WL}{X_{ox}} \tag{2.1}$$

where K_{ox} and X_{ox} are the dielectric constant and oxide thickness, respectively, and ε_o is the permittivity of the free space. The other component is fringing capacitance associated with neighboring metals and can be written as

$$C_{fri} = K_{ox}\varepsilon_o \frac{HL}{L_s} \tag{2.2}$$

where L_s is the spacing between two wires.

As technology progresses, it is projected that local interconnect delay will reduce at the same pace as gate delay. Table 2.2 shows the two common scenarios for scaling of local interconnect, considering S (< 1) as scaling factor. Due to the reduction of transistor size and increase of design complexity, more interconnect are needed for signal transmission/communication. Therefore, interconnect width and spacing are reduced by a factor of S with each generation, resulting in constant per unit capacitance. On the other hand the re-

Table 2.2: In local wires, ideal scaling translates to a constant and quasi-ideal scaling translates to slightly lower RC delay, respectively [69].

$Interconnection\ Parameter$	$Ideal\ Scaling$	$Quasi-ideal\ Scaling$
$Linewidth, Length, and\ Spacing$	S	S
$Wire, and\ ILD\ Thickness$	S	$\sqrt{S}$
$Resistance\ (per\ unit\ length)$	$\frac{1}{S^2}$	$\frac{1}{S^{3/2}}$
$Capacitance\ (per\ unit\ length)$	1	1
$RC\ Delay$	1	$\sqrt{S}$
$Line\ Current\ Density$	$\frac{1}{S}$	$\frac{1}{\sqrt{S}}$

duction of wirelength directly translates to constant RC delay for local wires. In the case of quasi-ideal scaling the vertical dimensions (wire and inter- layer dielectric (ILD) thickness) are scaled slower than the horizontal dimensions (linewidth, length, and spacing), resulting in a tall and narrow wire. The quadratic increase in per unit resistance is compensated by the slightly larger interconnect thickness, resulting in slightly lower (i.e., as a factor of $\sqrt{S}$) RC delay.

2.1.2 Global Interconnect Trends

Global interconnect connects separate functional units and can have significantly longer wirelength compare to local wirelength. While clock has both global and local interconnect. Unlike local wirelength, global wirelength is determined by the size of functional unit and the size of chip.

Since the complexity of high-performance microprocessors, SOCs, and Network-on-Chips (NOCs) continue to increase, the need for identifying the critical interconnect path and floorplan become ever more critical. The chip performance and power are determined by this critical interconnect (i.e., global interconnect) performance and power metrics. In order to understand the overall impact of global interconnect on the performance of future technology nodes, it is imperative to consider the scaling scenario of global interconnect.

Table 2.3 shows the common scaling scenarios of global interconnect considering ideal and constant-dimension scaling. Since the chip sizes and functional unit sizes are not

Table 2.3: The global RC delay increases as a scaling factor of $\frac{1}{S^3}$ and $\frac{1}{S}$ in case of ideal and constant dimension scaling, respectively [69].

$Interconnection\ Parameter$	$Ideal\ Scaling$	$Constant - dimension\ Scaling$
$Linewidth, and\ Spacing$	S	1
$Wirelength$	$\frac{1}{\sqrt{S}}$	$\frac{1}{\sqrt{S}}$
$Wire, and\ ILD\ Thickness$	S	1
$Resistance\ (per\ unit\ length)$	$\frac{1}{S^2}$	1
$Capacitance\ (per\ unit\ length)$	1	1
$RC\ Delay$	$\frac{1}{S^3}$	$\frac{1}{S}$
$Line\ Current\ Density$	$\frac{1}{S}$	S

shrinking in scaled technologies, the wirelength of global interconnect is not reducing. Hence, the global RC delay increases as a scaling factor of $\frac{1}{S^3}$ and $\frac{1}{S}$ in case of ideal and constant dimension scaling, respectively.

2.2 Different Signaling Schemes

The advancement of CMOS technology with innovative circuit topologies increases the speed of synchronous ASICs and SOCs. This creates a demand for high global clock frequency. In general, CMOS signaling schemes are VM. Since CMOS signals are terminated in a MOSFET gate, resulting nearly infinite impedance at the final node. On the other hand, in a CM signaling scheme, we process the current instead of voltage signal. The CM signals usually terminate at the node with low-impedance, makes them less sensitive to Electrostatic Discharge (ESD) stress compared to VM signals.

In the subsequent subsections, I discuss different VM, and CM signaling schemes, reported in the literature. These were primarily point-to-point signaling.

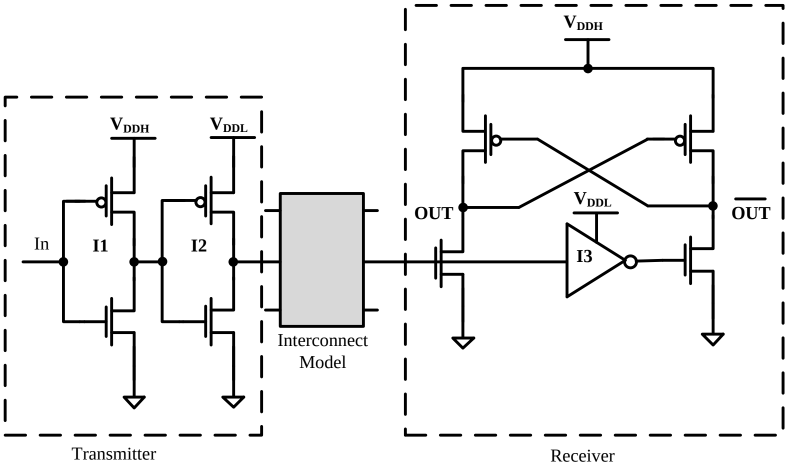


Figure 2.4: Conventional low-swing scheme utilizes dual-supply voltage and low-swing interconnect to save power [52].

2.2.1 Low-Swing Signaling

Reduced or low-swing signaling is very attractive in low-power design, in particular this method directly improves the energy/bit of an interconnect or data transmission system. A signaling scheme is considered to be an efficient design scheme and determined by the dynamic switching energy, design complexity or area/routing, delay, and reliability (i.e., robustness to process variation, voltage supply noise, and crosstalk noise).

A low-swing signaling scheme consists of a Tx/driver and a Rx circuit, and an interconnect between them. Generally, low-swing schemes are based on level-converter circuits. A conventional level converter circuit is shown in Figure 2.4 [52]. This scheme utilizes dual-Supply Voltage (V_{DD}) , where the high-V_{DD} denoted as V_{DDH} and low-V_{DD} denoted as V_{DDL}. This scheme is based on a Differential Cascode Voltage Switch Logic (DCVSL) based Rx circuit. The Tx circuit is a simple buffer, the first inverter (I1 of Figure 2.4) has a high-supply voltage while the second inverter (I2 of Figure 2.4) drives the interconnect line using a low-supply voltage. The Tx generates a slow and energy efficient signal. A DCVSL-type Rx circuit generates a full-swing output signal. This scheme enables large power savings compared to traditional full-swing signaling schemes. However, it is highly susceptible to crosstalk noise.

Pseudodifferential interconnect is another low-swing on-chip interconnect [83]. Fig-

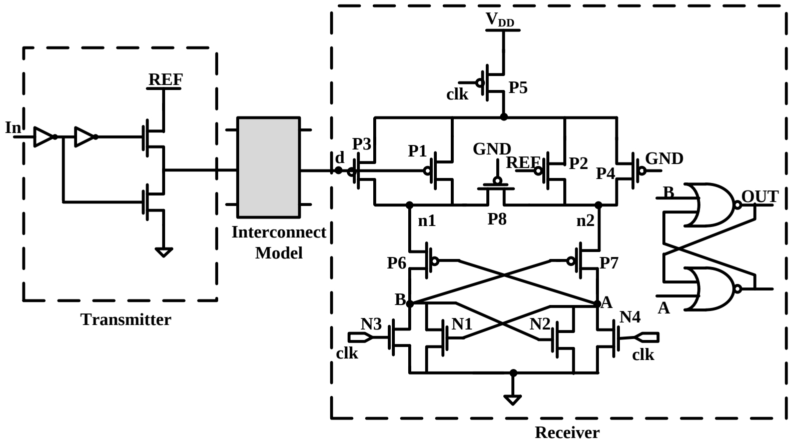


Figure 2.5: Pseudodifferential interconnect uses single wire; however, exhibits most advantages of differential sense amplifier such as low input-offset and good sensitivity [83].

ure 2.5 shows the pseudodifferential signaling scheme [83]. The Tx or the interconnect driver uses NMOS transistors for both pull-up and pull-down. The Rx circuit is a clocked sense amplifier followed by a static Set-Reset (SR) latch. It has dual-pair of PMOS ($P1$-to-$P4$) input transistors, with the gates of $P1$ and $P3$ connected to node d (i.e., termination end of interconnect), while the gates of $P2$ and $P4$ are being biased with REF and GND, respectively. An equalizing PMOS ($P8$) transistor biased with GND connected between $n1$ and $n2$ nodes. Similar to a traditional CMOS inverter sense amplifier circuit, this circuit has a cross-coupled inverter pair ($N1 - P6$ and $N2 - P7$).

The circuit operation of pseudodifferential interconnect can be explained with the reference of Figure 2.5. When node d reaches the desired level, the receiver circuit is enabled by negative clk signal. If d is low the current drive of $P1$ is larger than that of $P2$, while current drive of $P2$ and $P4$ are equal. Hence, B is pulled high and A pulled down to low by utilizing cross-coupled inverter pair. An opposite transition is occurred when the node d is high (i.e., equal to REF signal). When the sense amplifier is pre-charged, the following static FF retains the data value at the NOR based storage cell.

The major advantage of this scheme is it has a single interconnect wire, however, its Rx exhibits most of the advantages of a differential sense amplifier such as low input-offset and good sensitivity [28]. This scheme has high energy efficiency. However, the mismatch between

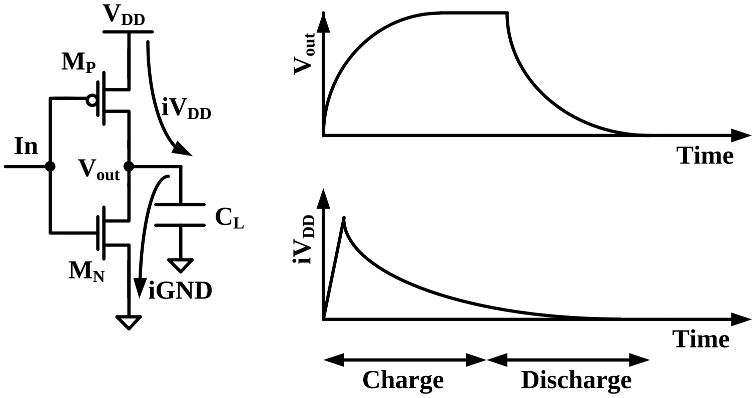


Figure 2.6: The CMOS gate consumes dynamic power in the process of charging-discharging the output capacitance (C_L).

distant REF and local REF signal can cause functional failure or use more energy. Moreover, the stacked PMOS transistors require a large area to implement the receiver circuit.

2.2.2 Resonant Energy Recovery Clocking

Resonant energy recovery clocking has emerged with great potential to reduce active clock power and release stringent timing budgets of high-performance digital ICs [60]. Resonant clocking uses the CDN capacitance and an on-chip inductor to resonate at a fundamental frequency.

In order to understand how energy recovery works and saves energy, I explore the dynamic energy dissipation of a traditional CMOS logic gate. Figure 2.6 shows the output voltage (V_{out}) and charging-discharging currents ($i_{V_{DD}}$ - i_{GND}) of a standard CMOS gate. We can precisely measure the dynamic energy dissipation of each cycle by using

$$E_{V_{DD}} = \int_0^{\infty} \mathrm{i_{V_{DD}}(t)} V_{DD}\,\mathrm{d}t = V_{DD}\int_0^{\infty}(C_L \frac{\mathrm{d}V_{out}}{\mathrm{d}t})dt = C_L V_{DD} \int\limits_0^{V_{DD}} \mathrm{d}V_{out} = C_L V_{DD}^2, \tag{2.3}$$

Similarly, we can calculate the value of the energy E_S, stored on the capacitor at the end of the

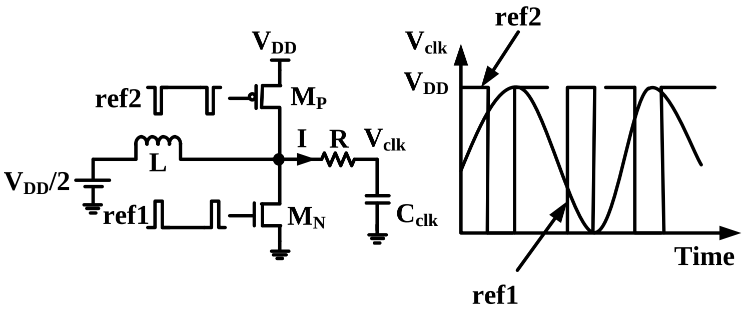


Figure 2.7: In resonant clocking scheme narrow positive and negative pulses are applied to sustain oscillation and to avoid overshoot, respectively [65].

transition, by integrating instantaneous power over the period using

$$E_s = \int_0^{\infty} \mathrm{i}_{\mathrm{V_{DD}}}(\mathrm{t}) V_{out}\, \mathrm{d}t = \int_0^{\infty} (C_L \frac{\mathrm{d}V_{out}}{\mathrm{d}t}) V_{out} dt = C_L \int\limits_0^{V_{DD}} \mathrm{V}_{out} dV_{out} = \frac{C_L V_{DD}^2}{2}, \quad (2.4)$$

Clearly, from Equation 2.3 and Equation 2.4, half of the energy (i.e., $\frac{C_L V_{DD}^2}{2}$) dissipates in the process of charging load capacitance at pull-up network (M_P of Figure 2.6) while the other half of the energy dissipates in the pull-down network (M_N of Figure 2.6). Hence, energy recovery clocking re-cycles the latter half of the energy that dissipates in the pull-down network using a lumped inductor.

A traditional resonant clock circuit that is capable of absorbing the energy stored in the electric field of circuit capacitance is shown in Figure 2.7. A lumped inductor is employed to store and recycle the returned energy from the circuit capacitance into magnetic energy. Hence, C_{clk} periodically charges and discharges through LC resonance, resulting in a resonant clock output at V_{clk} node. In order to sustain oscillation and replenish I^2R, the losses due to the resistance of the circuit, a narrow positive pulse is applied to the NMOS (M_N of Figure 2.7). While, a narrow negative pulse is applied to the M_P (PMOS) when the V_{clk} is going low-to-high to avoid overshoot and kept the output at V_{DD} level while maintaining the oscillation. This scheme effectively removes the short-circuit current of the driver circuit. Now, we can write the natural oscillation frequency (f_c) of the LC network, ignoring the damping factor as

$$f_c = \frac{1}{2\pi}\sqrt{\frac{1}{LC_{clk}}} \tag{2.5}$$

Considering the resulting sinusoidal voltage at node (V_{clk}), we can write the average current (I_{avg}) and average energy dissipation in each cycle as

$$I_{avg} = \frac{\omega VC}{2} \tag{2.6}$$

and

$$E_r = \frac{1}{2}{I_{avg}}^2RT = \frac{1}{2}(\frac{\omega VC}{2})^2RT = \frac{1}{2}(\frac{2\pi f_c VC}{2})^2RT = \frac{1}{2}\pi^2 f_c V^2C^2R, \tag{2.7}$$

respectively. Utilizing the resulting energy dissipation in Equation 2.7 and Equation 2.5 and understanding the quality factor ($Q = \frac{\sqrt{\frac{L}{C}}}{R}$) of a series ($RLC$) network, we can write the energy dissipation of a resonant circuit as

$$E_r = \frac{\pi}{4Q}CV_{DD}^2 \tag{2.8}$$

Comparing the energy dissipation of CMOS network (Equation 2.4) and resonant energy recovery network (Equation 2.8) with the same capacitive load, we can come to a break even point of the quality factor of a (RLC) network as

$$Q_{min} > \frac{\pi}{2} \tag{2.9}$$

Hence, Equation 2.9 represents the minimum Q of a resonant energy recovery network to use less power than a traditional CMOS network and also gives the direction to choose the appropriate inductor (L) model for tank circuit.

In the past decade, a number of test chips successfully demonstrated resonant clocking implementation [12, 14, 65]. In the early years, resonant clocking was used in adiabatic circuits, where charge stored in interconnect parasitic and internal dynamic logic gate nodes has recycled into discrete or integrated on-chip inductors [4, 14].

In a similar approach, Chan et. al introduced uniform-phase, uniform-amplitude res-

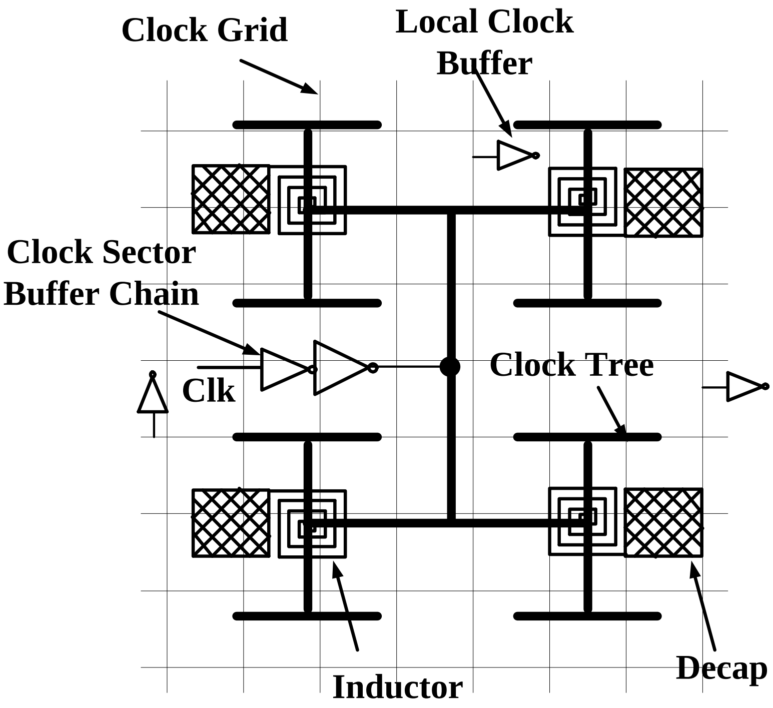


Figure 2.8: Components and topology of a resonant clock sector, local sector buffers, and global clock grid [14].

onant load global clock distribution network [14]. In this paper, on-chip spiral inductors were used to resonate the clock sector/local clock network capacitance, while the global clock distribution utilized buffers to drive the clock grids. Figure 2.8 shows the components and topology of a resonant clock sector [14]. At 4.6 GHz resonant clock frequency, this scheme observed up to 20% saving of clock power at 90nm node.

There have been equally impressive amounts of work done to attain more energy efficiency. Researchers have applied global resonant clocking to global clock distribution networks and used resonant clocking to drive the final sinks (i.e., FFs/latches) [48, 64].

Another prior approach applied resonant clocking to produce a standing wave clock using an inductively loaded standing wave oscillator for global CDN [63]. Unlike a conventional standing wave CDN, this network provide uniform phase and approximately uniform amplitude clock signal throughout the network. However, this method utilized distributed LC network,

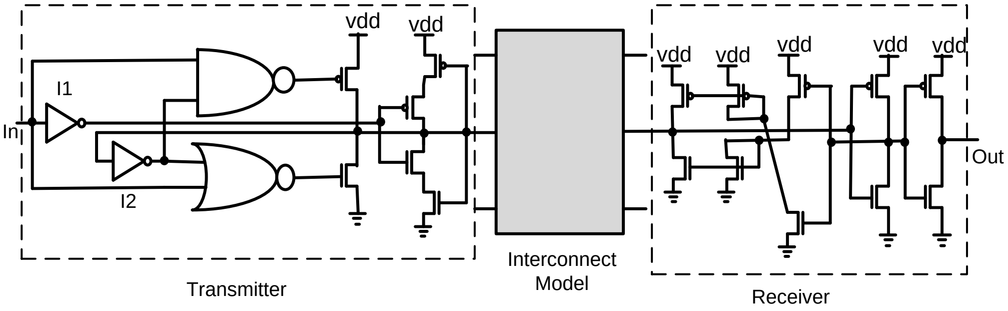


Figure 2.9: Dynamic over-driving transmitter based CM scheme uses feedback connection at receiver circuit to tackle V_{CM} shift, but voltage variation at long interconnect source and sink nodes can result rise-time and fall-time mismatch in the output [43].

which plugged in with area overhead, and amplitude variation in the clock signal may cause functional failure of the final timing elements driven by these clocks.

Another attractive approach applied resonant energy recovery clocking to produce traveling wave clock [17]. This paper is based on Rotary Traveling Wave Oscillator (RTWO) that creates a traveling wave within a closed loop differential transmission line [77]. In RTWOs, distributed inverters placed to sustain oscillation and to ensure rotational clock. However, the clock phases are $180\,^{\circ}$ out of phase creates design complications and requires clock recovery circuit in order to have uniform phase clock output.

All the energy recovery resonant clocking schemes are frequency limited. In recent years, the narrow frequency range of conventional resonant clocking was eliminated in Intermittent Resonant Clocking (IRC) [8, 9, 24]. However, resonant clocking still requires an extra inductor and inherently has high signal rise/fall time due to the sinusoidal output signal. We provide same with more evidence on that in Section 2.3.

2.2.3 Current-Mode Signaling

CM signaling scheme offers low-energy with higher reliability. Moreover, this scheme is less susceptible to high-energy particle induced single-event transients. Nearly 20% of overall sequential soft error rate is reported due to a clock node upset [32, 50].

In a CM signaling scheme, a transmitter utilizes a VM input signal to transmit a

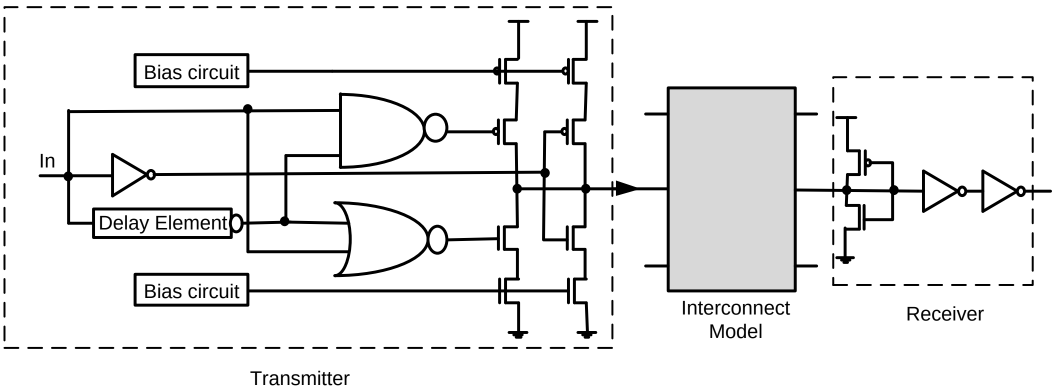


Figure 2.10: Expensive variation tolerant CM signaling scheme consumes large static and dynamic power when compared to the other CM techniques [18].

current with minimal voltage swing into an interconnect (transmission line), while a receiver converts current-to-voltage providing a full swing output voltage. In the early stages CM signaling scheme applied to current sense amplifiers for CMOS Static Random-Access Memories (SRAM) [66]. This paper modeled long interconnects bit line as an RC network and analytically showed the significant improvement of CM signaling delay compared to VM signaling.

Another interesting CM signaling scheme reported by Katoch and his colleagues [43], as depicted in Figure 2.9. The scheme is based on a dynamic over-driving transmitter with a strong and weak driver. The strong driver turns ‘ON’ for short duration depends on the switching threshold of feedback inverter (I_2 of Figure 2.9), while the weak driver provides a small static current to the line. The receiver circuit consists of a low-gain inverter amplifier and a controlled current source. The feedback from the amplifier provides necessary bias voltage at the input node of the receiver network for appropriate current sensing. Moreover, this schemes tackles the problem associated with the receiver line Common-Mode Voltage (V_{CM}) swing by using feedback at the receiver circuit. In order to compensate for V_{CM} shift, a feedback connection from the output node injects appropriate amount of current to the receiver input node. However, due to the long transmission line it is likely to have different voltage at transmitter output and receiver input nodes, resulting rise-time and fall-time mismatch in the output [18].

Other researchers presented a variation tolerant CM signaling scheme for on-chip interconnects [18]. The major contribution of this work is to design a variation tolerant CM Tx with corner-aware bias-circuitry. Figure 2.10 shows the variation tolerant CM scheme including receiver and transmitter circuits [18]. Similar to the dynamic over-driving Tx based CM scheme approach, this Tx has a strong and weak driver. However, one of the NAND-NOR gates input is fed from a delayed version of clock signal instead of that feedback connection (see Figure 2.9 and Figure 2.10). Unlike the dynamic over-driving Tx based CM scheme [43], it has a variation tolerant bias network for the Tx circuit. In this scheme, the Rx circuit provides low-impedance to the ground and holds the terminal point at the switching threshold. However, this comes at the expense of large static and dynamic power when compared to the other CM techniques and makes it unattractive compared to existing VM clock signaling.

2.3 Preliminary Experiments

2.3.1 Experimental Setup

In order to compare the propagation delay (t_{pd}), power consumption, and average rise/fall time (t_{rf}) of different schemes, I implemented a test network using a $5mm$ interconnect wire as shown Figure 2.11. The interconnect RC parasitics were extracted from Predictive Technology Model (PTM) [31] considering top global metal layer using interconnect $width = 0.8\mu m$, $spacing = 0.8\mu m$, $thickness = 2\mu m$, $height = 1\mu m$, $dielectric - constant = 2.5$.

In this experiment, I utilized the conventional level converter circuit based low-swing scheme [52], a resonant energy recovery signaling [4], a dynamic over-driving Tx based CM signaling scheme [43], and a traditional buffered VM scheme implemented with a 45nm CMOS technology model [55]. The power-performance of each scheme was evaluated considering frequencies from $1 - 3GHz$ (typical global clock frequency range) and a 1V supply voltage. However, for the low-swing scheme in addition to a regular supply voltage, I used a low-supply voltage ($V_{DDL} = 0.8V$).

2.3.2 Results and Comparisons

Table 2.4 shows the propagation delay, power (static and dynamic), and average rise-fall time comparison of the different signaling scheme on a $5mm$ interconnect. As expected, the repeater based buffered VM signaling scheme consumes the highest power at all the frequencies

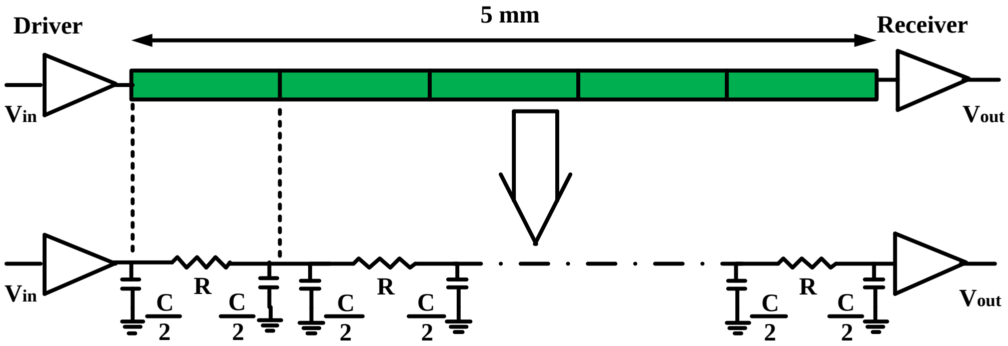


Figure 2.11: In order to measure the power-performance of different signaling scheme, I used $5mm$ interconnect line modeled as distributed $R-C$ network driven by respective Tx circuits and the final node of the interconnect is connected to the Rx circuits that provides a full-swing output voltage.

due to the full $(0 - to - V_{DD})$ voltage swing in the interconnect. On the other hand, the dynamic over-driving Tx based CM scheme consumes the lowest power due to the negligible voltage swing on the interconnect. In particular, the CM scheme consumes 58$\%$ to 72$\%$ less power compared to the buffered VM signaling scheme from 1-3GHz frequency as shown in Figure 2.12.

Among all the signaling schemes, resonant energy recovery signaling has lowest propagation delay and buffered scheme has largest propagation delay. At $3GHz$, the CM signaling scheme has 38$\%$ higher propagation delay than the resonant signaling scheme due to the large Rx circuit delay. In addition, the low-swing signaling scheme has the lowest rise time and resonant energy recovery scheme has the highest rise time.

2.4 Summary

In this Chapter, I presented, the interconnect trends considering local and global wire routing. I also presented different low-power VM and CM signaling schemes. In order to compare efficiency, I performed analysis on a $5mm$ interconnect line by implementing the different signaling schemes. Among all the schemes dynamic over-driving Tx based CM scheme consumes lowest power [43], while traditional buffered based VM scheme consumed highest power. In addition, the resonant energy recovery signaling has 62% to 65% lower t_{pd} compared to the buffered signaling scheme. However, the most important observation of this preliminary experiments were the power consumption of CM scheme increases proportional to frequency at a much slower rate than the other signaling schemes as shown in Figure 2.12.

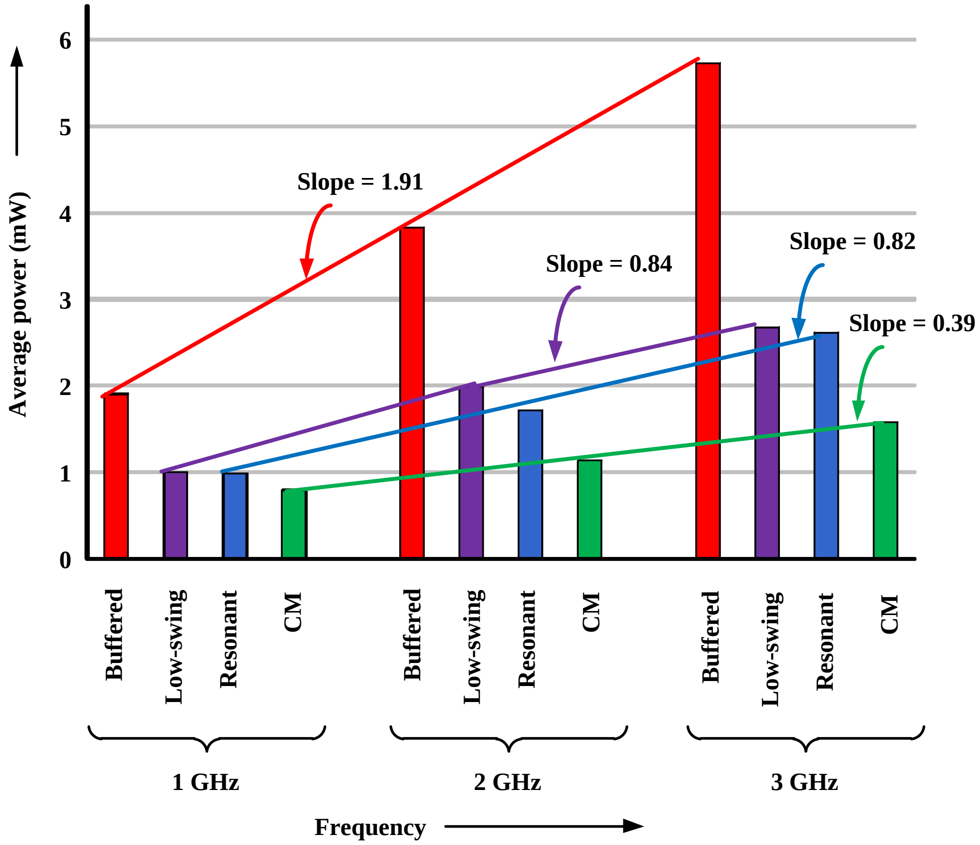


Figure 2.12: The CM scheme consumes 58% to 72% less power compared to the buffered VM signaling scheme from $1 - 3GHz$ frequency.

Table 2.4: The buffered VM signaling scheme consumes the highest power due to the full $(0 - to - V_{DD})$ voltage swing in the interconnect and the dynamic over-driving transmitter based CM scheme consumes the lowest power due to the negligible voltage swing on the interconnect at $1 - 3GHz$ frequency.

$Sig.\ schemes$	$Frequency$								
	$1GHz$			$2GHz$			$3GHz$		
	$t_{pd}(ps)$	$t_{rf}(ps)$	$Power\ (mW)$	$t_{pd}(ps)$	$t_{rf}(ps)$	$Power\ (mW)$	$t_{pd}(ps)$	$t_{rf}(ps)$	$Power\ (mW)$
$Buffered$	156.2	23.2	1.91	156.6	23.3	3.83	155.8	22.7	5.73
$Low - swing$ [52]	146.3	16.3	1.0	144.0	16.4	1.99	143.0	15.4	2.68
$Resonant$ [4]	59.6	110.2	0.99	56.4	100.2	1.72	54	100.0	2.62
CM [43]	73.6	70.5	0.8	68	30.3	1.14	86.4	21.0	1.58

Chapter 3

Current-Mode Clocking

In this Chapter, I present the first true CM CDN and a new CM pulsed D-type FF where the Clock (CLK) input is a CM Rx and the data input (D), an active low enable ($\overline{EN}$), and output (Q) are VM. This enables CM clocking with VM logic. In particular, the key contributions of this Chapter are the first demonstration of a CM clocked FF, the effective integration of the CM FF with VM CMOS logic, the power consumption comparison of CM CDN and VM CDN at different frequencies and the noise and variability analysis of CM and VM CDN.

3.1 Existing Current-Mode Clocking

CM clocking is very attractive for high-performance, high-noise immunity, and low-power operation. However, CM CDNs have been researched very little. According to my best exploration, only two previous work proposed CM clocking schemes [42, 53]. One method utilized CM signaling on symmetric H-tree [42], while the other applied CM signaling on a symmetric NOC design [53].

The representative CM scheme in Figure 3.1 uses a CMOS inverter as the Tx while the Rx is based on a transimpedance amplifier [53]. The amplifier output node ‘X’ drives the large NMOS M6 to saturation, resulting in low impedance path to ground for the input current sourced by the driver. The diode connected PMOS M7 regulate the transconductance of M5 while, M3 provide the negative feedback and modulate the input impedance. The two output inverters aided the amplifier to provide a rail-to-rail output voltage. This scheme provides significant delay improvement over VM schemes, but the receiver line voltage swings around the (V_{CM}).

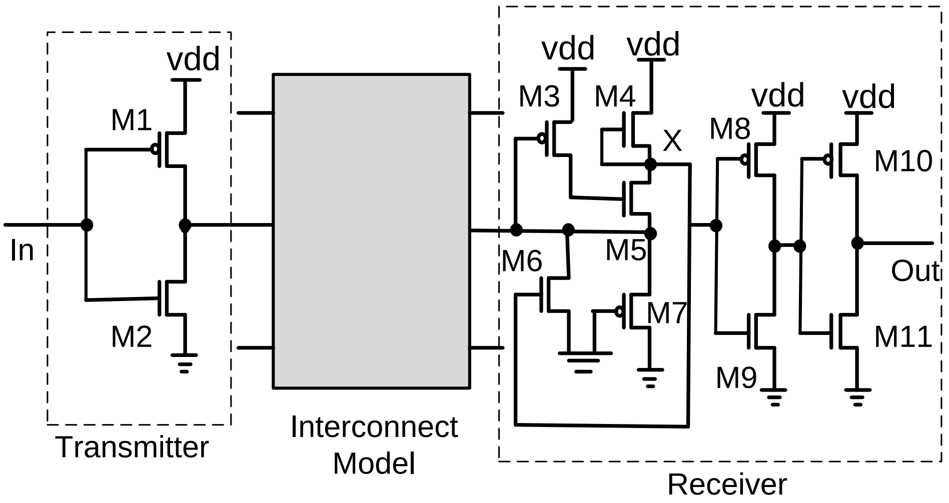


Figure 3.1: Expensive transimpedance amplifier receiver CM scheme exhibits significant skew due to V_{CM} shift if applied to CDNs [53].

The shift of V_{CM} causes a large skew in a CDN [42].

3.2 Overview of Existing Pulsed Flip-flops

The proposed CM clocking uses an Rx circuit as an edge-triggered pulsed FF. Hence, in this section, I would like to discuss the Traditional Pulsed FF (TPFF) and a recently reported high-performance pulsed type FF. The basic idea of a pulsed FF is to generate a small transparent window to latch data at the rising or falling edge of the input CLK signal as shown in Figure 3.2. A pulsed type FF inherently exhibits negative setup time, makes it more attractive than a regular master-slave FF. The delay between the rising edge of the CLK and the pulsed signal (CLKG) can be defined as the negative setup time.

Figure 3.3 shows the schematic of a TPFF. It has an input stage to generate the pulsed signal (less than 50% duty cycle), a register stage, and a storage cell. When the CLK signal is low node ‘A precharged to high. At the rising edge of the CLK, the two inputs of AND gate is high for a brief period of time resulting a voltage pulse (CLKG) at the buffer (X1-X2) output.

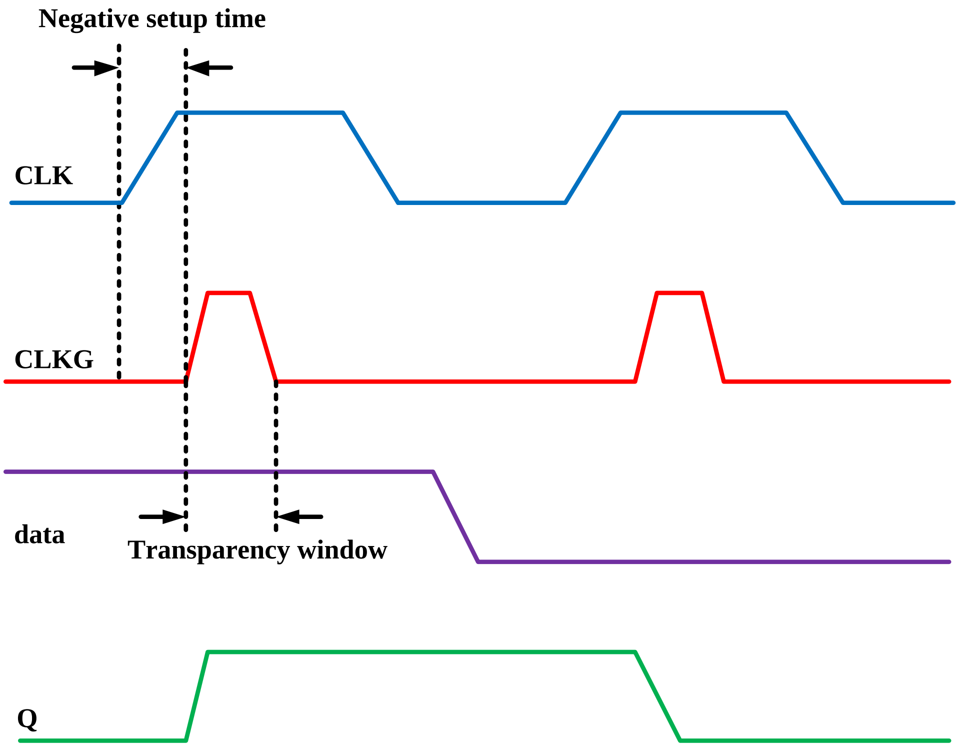


Figure 3.2: The Pulsed FF creates a transparency window after a certain delay to latch data at the storage cell; that delay can be defined as the pulsed FF negative setup time.

For TPFF the accumulated time delay of the AND gate and the buffer (X1-X2) is the negative setup time. The generated CLKG signal triggers data to the storage cell as shown in Figure 3.2. This FF consumes low power at low-frequencies and has a large negative setup time. However, this FF consumes high-dynamic power at high-frequencies and will be discussed in detailed in Section 3.5.3.

In a recent work, Dual Dynamic node hybrid Pulsed FF (DDPFF) has a low Clock-to-Q (CLK-Q) delay [1]. The schematic diagram of DDPFF is shown in Figure 3.4. Unlike TPFF, this FF has two storage cells (I1-I2 and I3-I4). The DDPFF creates a transparency window in the overlap of CLK and CLKB signal at the rising edge of the CLK signal. This FF consumes low power, however, requires large silicon area.

Another recent work uses Conditional Pulse Enhancement technique in a implicit-

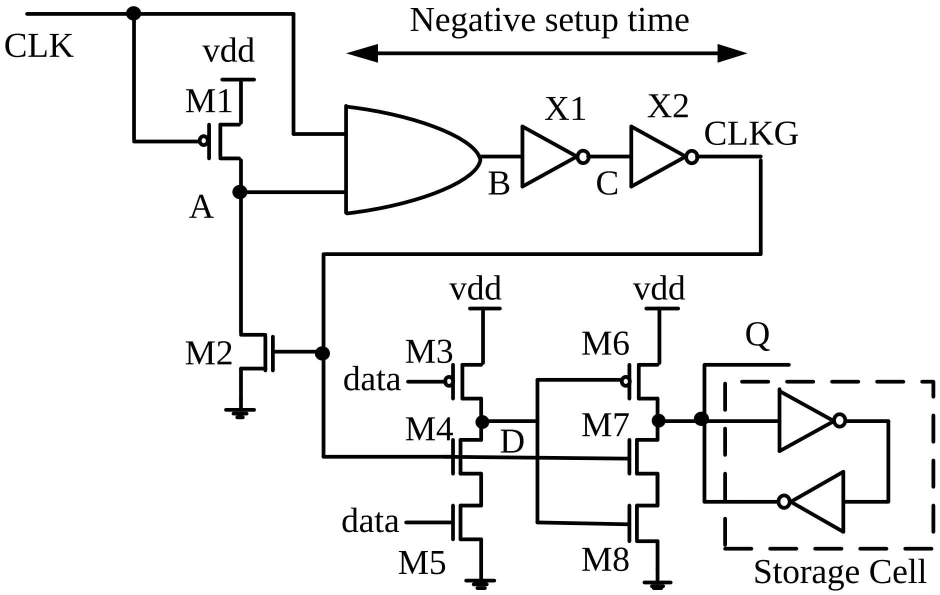


Figure 3.3: At the rising edge of the CLK signal the TPFF explicitly generates the pulsed signal (CLKG) that triggers data at the register stage to store at storage cell.

pulsed trigger FF (CPEFF) [30]. This FF has very low CLK-Q delay and exhibits low-power operation. However, the CPEFF has large hold time due to the pulse-enhancement architecture.

3.3 Challenges

Traditionally, CM signaling schemes offer low-power and high-performance operation. Hence, devising a new CM clocking scheme that can save more power with maintaining the same performance is our primary challenge. However, the traditional CM clocking schemes used a receiver circuit as current-to-voltage converter and buffers to drive highly capacitive sinks. So, designing a CM FF that can eliminate the requirement of extra buffers and efficiently work with traditional VM scheme is our primary goal.

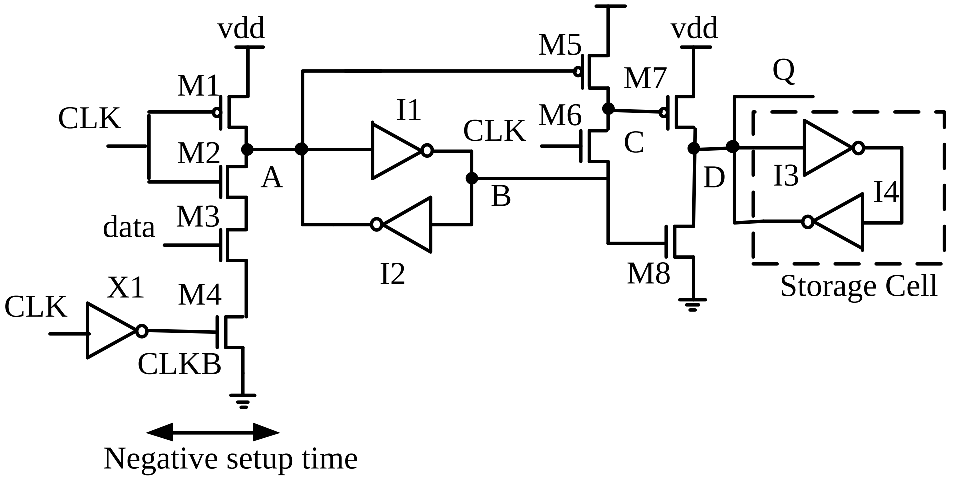


Figure 3.4: Using the rising edge of the CLK and delayed version of inverted CLK (CLKB) signal the DDPFF implicitly generates the pulsed signal that triggers data at the register stage to store at storage cell [1].

3.4 Proposed Current-Mode Clocking

All of the previous CM clocking schemes perform current-to-voltage conversion and then use the buffered VM signal. However, driving the lowest level of a CDN with a full-swing voltage result in large dynamic power in addition to significant buffer area to drive the clock pin capacitances. Hence, a new high-performance CM scheme without final buffers can reduce overall power consumption and silicon area of a CDN.

3.4.1 Proposed Current-Mode Pulsed Flip-flop with Enable (CMPFFE)

Figure 3.5 and Figure 3.6 show the circuit and simulation data of the proposed current-mode pulsed DFF with enable (CMPFFE). The CMPFFE uses an input Current-Comparator (CC) stage, a register stage, and a static storage cell. The CMPFFE also uses an active-low enable ($\overline{EN}$) signal. The CC stage compares the input push-pull current with a reference current and conditionally amplifies the CLK to a full-swing voltage pulse that triggers the data to latch at the register stage. The feedback pulsed FF is in stark contrast to the previous CM schemes which utilized expensive Rx circuits and buffers to drive the final FFs .

The choice of push-pull current enables a simple Tx circuit (discussed further in Section 3.4.2) while maintaining a constant (or at least low-swing) bias voltage on the CDN interconnect. The CMPFFE in Figure 3.5 is only sensitive to unidirectional push current which provides the positive edge trigger operation of the FF. This design is easily modified using a complementary current comparator into negative clock edge FF using the pull current.

In order to efficiently receive an input pulse current, a CM Rx requires a low input impedance (Z_{in}). A small signal analysis at the input of the proposed CMPFFE ensures the low Z_{in} according to

$$Z_{in} = \frac{1}{g_{m1} + g_{m2}} \tag{3.1}$$

where g_{m1} and g_{m2} are the transconductance of transistor M1 and M2, respectively. The input impedance of the proposed CM FF is also identical to the previously reported variation-tolerant CM signaling Rx [18].

Traditionally, CM Rx/logic circuits consume a significant amount of static power even when the circuits are in sleep mode. Our CMPFFE incorporates an active-low enable ($\overline{EN}$) signal that, when low, connects PMOS (M4) to vdd for normal operation. On the other hand, it disables the static current I1 in stand-by mode when high. Since internal node B is decoupled in this stand-by mode, an additional transistor M7 is required to ground the internal CLK node and prevent any unintentional latching of input data. Transistor M7 is disabled during normal operation. Adding an extra ‘OFF’ transistor will introduce a stacking effect in the CC [61]. Since the leakage of a two-transistor stack is an order of magnitude less than the leakage in a single transistor, resulting significantly lower leakage current in M4 [79]. The peak CMPFFE leakage current is $2.4\mu A$, significantly smaller than the peak switching current of $134\mu A$ in active mode. However, global $\overline{EN}$ routing requires extra metal resources. Since the proposed CM scheme does not require buffers in the CDN, it is not difficult to globally route $\overline{EN}$.

In the input stage, the reference voltage generator (Mr2-Mr3) creates a reference current (Iref1) that is mirrored by M4 and generates I1. Similarly, the M1-M2 pair creates the FF reference current (Iref2) which is combined with the input current (i_in); this current is then mirrored by M5 to I2. A PMOS (Mr1) is added to replicate the voltage drop of M3.

It is possible to use a local or global reference voltage generator for the input gate voltage of M4. Using a global reference can increase the robustness by reducing transistor mismatch between FFs. Hence, I used a global reference voltage generator that distributed across

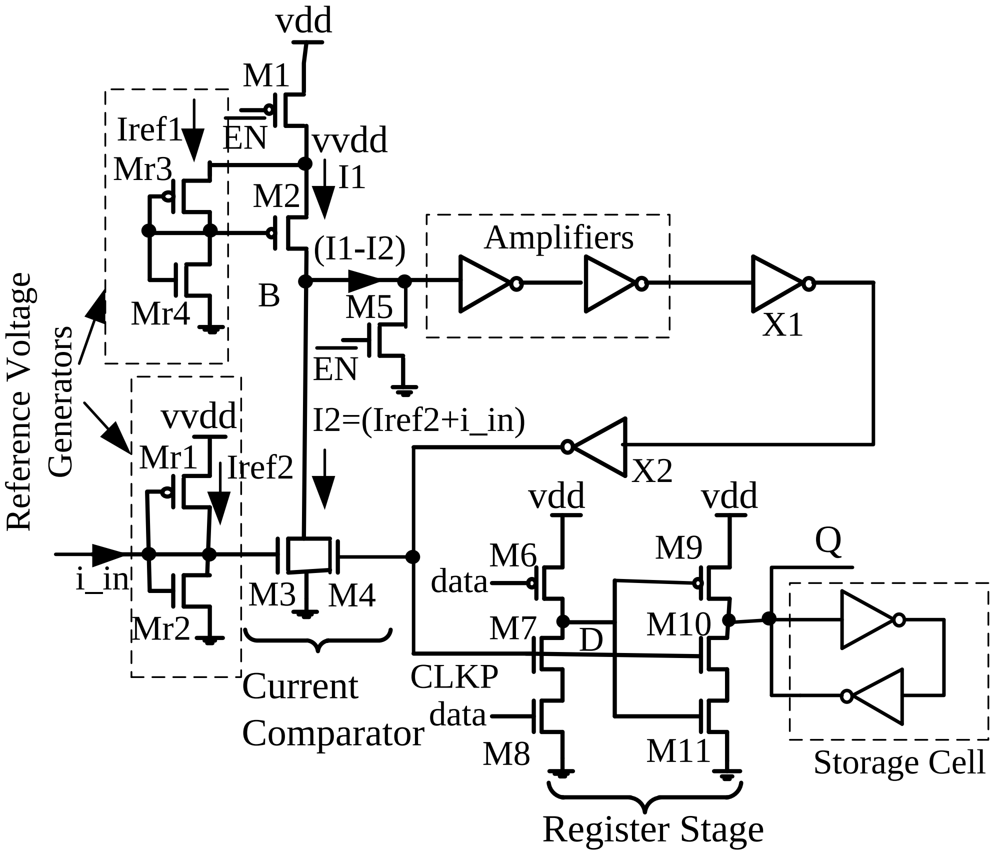


Figure 3.5: The Proposed CMPFFE uses current-comparator and feedback connection to generate a voltage pulse that triggers a register stage to store data in the storage cell.

the whole chip, when I integrate the CMPFFE with the CM CDN. This also saves two transistors per FF and reduces static power with a negligible performance penalty. Unlike corner-aware reference voltage generators [18], I used a simple three transistors global reference voltage generator as shown in Figure 3.5. In addition, CM signaling eliminates the requirement of CDN buffers, which reduces significant active area and makes easier global reference routing.

The mirrored currents I1 and I2 are compared using the inverting amplifier (A1) at node B and further extended to a CMOS logic level at node C by another inverting amplifier (A2). The inverter pair (X1-X2) generate the required voltage pulse duration before the feed-

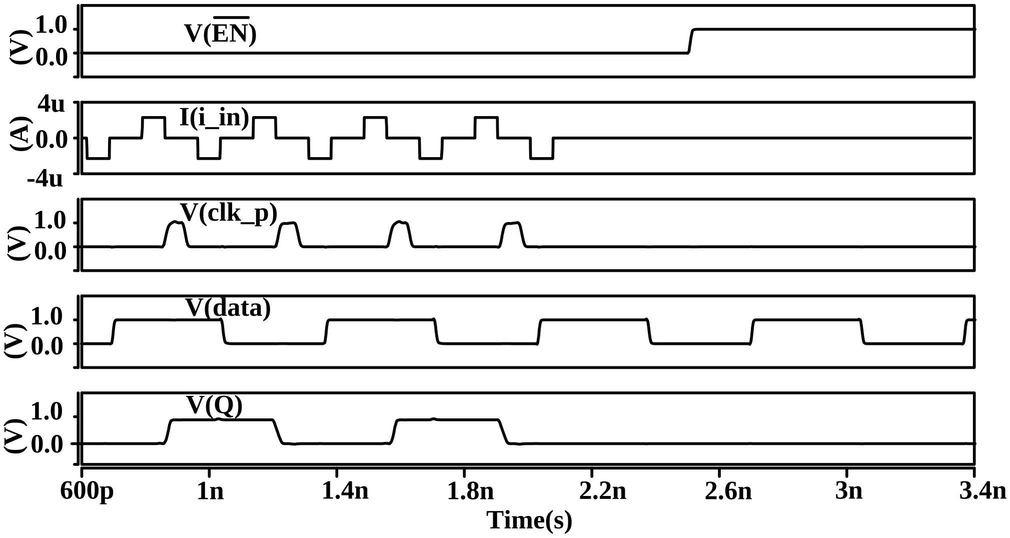


Figure 3.6: Simulation waveforms confirm the internal current-to-voltage pulse generation (clk_p) that triggers input data capture.

back connection in M6.

The feedback connection from the generated voltage pulse with M6 quickly pulls down the current comparator node B which facilitates generating a small voltage pulse and results in fewer transistors in the register stage. In addition, I properly size the X2 inverter so that it can efficiently drive the clock capacitance of register stage without affecting circuit performance.

Figure 3.7 shows the transfer characteristics of the proposed CMPFFE based on input current and voltage pulse (clk_p) generation. Figure 3.7 identifies three regions of operation of the proposed FF. In region 1, the input current is ≤ 0, and node B starts discharging from steady state resulting in a high voltage (very low swing $980mV - 850mV$) at the A1 output. Hence, the clk_p signal stays at 0. In region 2, the input current is ($0 < i\_in < 1.5\mu A$), and node B starts moving towards steady state to high. However, the swing is not large enough resulting in a low clk_p signal. In region 3, the input current is $\geq 1.5\mu A$, and the voltage swing at node B is large enough so that the amplifiers and inverter chain can generate required voltage pulse (clk_p goes low to high Figure 3.6) for the register stage.

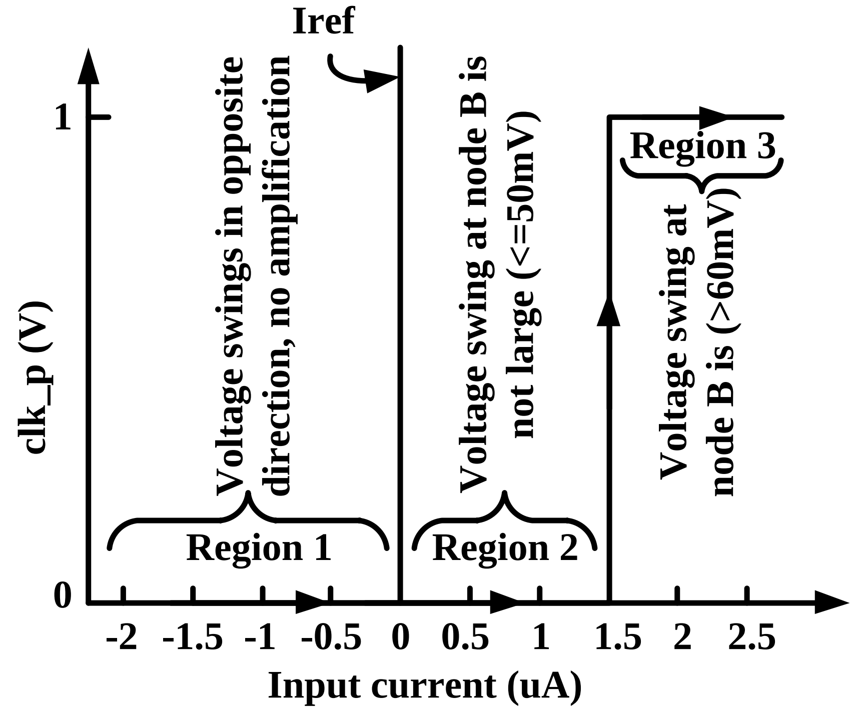


Figure 3.7: The proposed CMPFFE generates an output voltage pulse depending on the input current and also complementing the edge-triggered operation.

The register stage is similar to a single-phase register [81], but requires fewer transistors and has a reduced clock load compared to other pulsed FFs. The current-generated voltage pulse triggers storing data in the output storage cell.

The sizing of M6 is critical to the voltage pulse; I use a minimum sized NMOS transistor with unity aspect ratio. The width of the generated clk_p is also sensitive to the width and amplitude of input current (i_in). The amplitude of i_in strongly affects the FF performance by changing the operating point of M5 and adding extra delay to generated clk_p signal. In order to achieve minimum CLK-Q delay, the ideal input current has a $\pm 2.3\mu A$ amplitude and $70ps$ pulse width. This can be guardbanded to tolerate noise and variation.

3.4.2 Current-Mode Transmitter and H-Tree Distribution

In order to integrate the CMPFFE, I need a reliable transmitter that can provide a push-pull current into the clock network and distribute the required amount of current to each CMPFFE. Figure 3.8 shows the possible current Tx circuit and also can fulfill the present purpose. The basic idea is to construct a short positive (negative) pulse around the rising (falling) edge of the clock. The transmitter receives a traditional voltage CLK from a Phase-Locked Loop (PLL) or a CLK divider at the root of the H-tree network and supplies a pulsed current to the interconnect which is held at a near constant voltage. The clock distribution is a symmetric H-tree with equal impedances in each branch so that current is distributed equally to each CMPFFE leaf node.

The pulsed current Tx in Figure 3.8 is similar to previous transmitter circuits [18,43], but I have used an NAND-NOR design. The NAND gate uses the CLK signal and a delayed inverted CLK signal, clkb, as inputs to generate a small negative pulse to briefly turn on M1. Hence, the PMOS transistor briefly sources charge from the supply while the NMOS is 'OFF'. Similarly, the NOR gate utilizes the negative edge of the CLK and clkb signals to briefly turn 'ON' M2. Hence, the NMOS transistor briefly sinks current while the M1 is off. The non-overlapping input signals from the NAND-NOR gates remove any short circuit current from the transmitter.

The Tx M1 and M2 device sizes are adjusted to supply/sink charge into the CDN. The root wires of the CDN carry the current that is distributed to all branches so the sizing of CDN wires is critical for both performance and reliability. If the resistance of the wire is too high, the current waveform magnitude and period will be distorted and affect the performance of the CMPFFEs. The wire width must also consider electromigration effects while carrying a total current to drive all the FFs with the required current amplitude and duration.

The current transmitter is simple and occupies small silicon area due to a small number of gates. Moreover, I can easily fix the output voltage to a constant level, by changing the size of M1 (PMOS) and M2 (NMOS) transistors. Depending on the bias requirement of the proposed receiving CMPFFE, I can adjust the transistor size in order to supply/sunk appropriate charge at the output node. It is worth mentioning that I can also vary the generated current pulse width by changing the number of inverters inside the delay element.

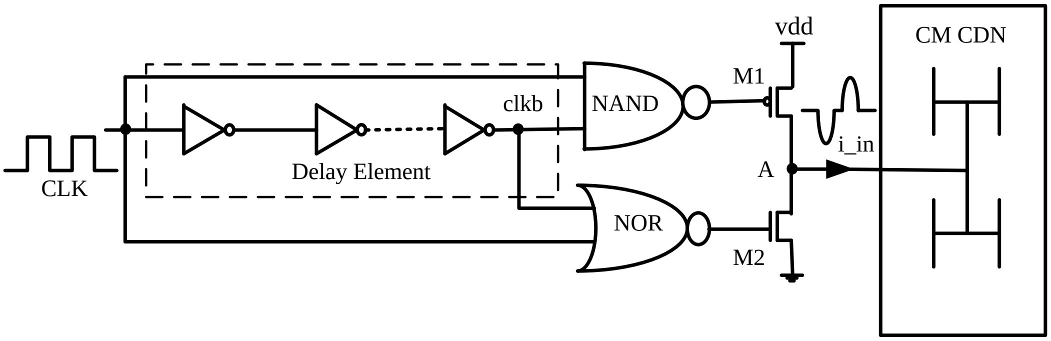


Figure 3.8: The proposed CM transmitter converts an VM input signal to a push-pull current and distributes current equally to the CM CDN.

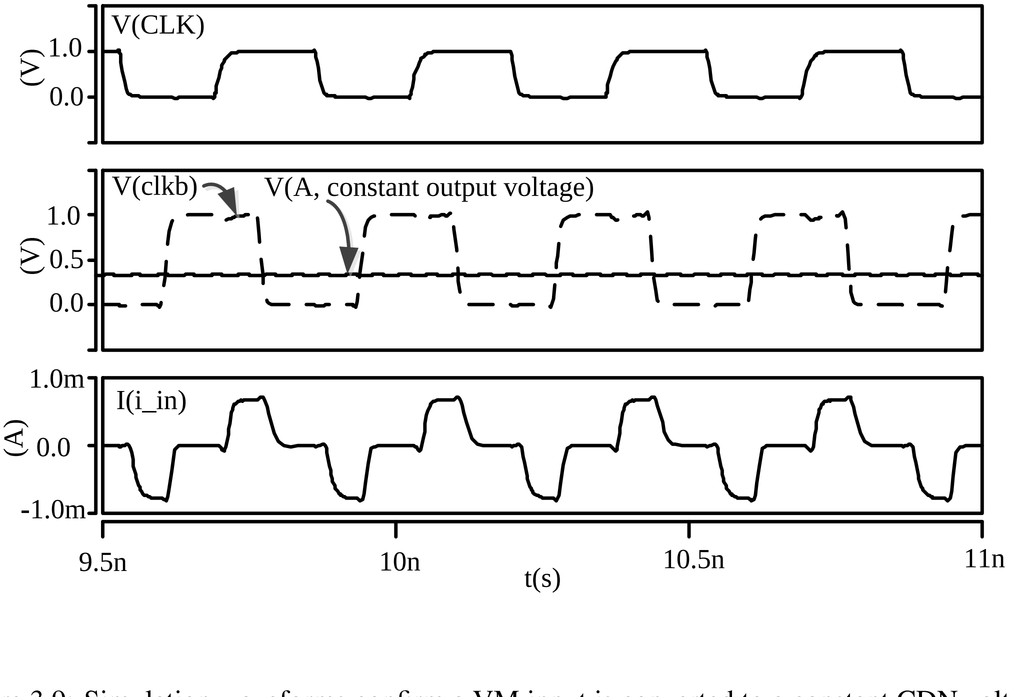


Figure 3.9: Simulation waveforms confirm a VM input is converted to a constant CDN voltage and a representative push-pull current at the output.

3.5 Experiments

3.5.1 Experimental Setup

I implemented the proposed CMPFFE, a traditional VM Master-Slave DFF (MSDFF), a TPFF [44], a high-performance CPEFF [30], and a recently reported low-power DDPFF [1] in FreePDK 45nm CMOS technology [55]. Each FF is compatible with a standard cell library height of 12 horizontal M2 tracks. The layout areas, maximum CLK-Q delay, Setup Times (t_s), Hold Times (t_h), and total power are listed in Table 3.1. The performance of the FFs was evaluated using post-layout SPICE simulation at CLK frequencies from $2-5GHz$ with less than $10ps$ slew and a $1V$ supply voltage. The power considers input data at 100% activity and 4 minimum size inverter load.

In order to validate the functionality of the CM Tx and the proposed CMPFFE in a CDN, I implemented a symmetric H-tree network spanning $1.2mm \times 1.2mm$. Each branch of clock tree is modeled as a lumped 3-component Π-model and then connected together to make a distributed CDN model [59, 76]. The interconnect unit capacitance and resistance values are as suggested by 2009-2010 ISPD Clock Synthesis contest [70, 71]. In addition, it is reasonable to model clock network as RC wires instead of RLC wires as suggest by 2010 ISPD Clock Synthesis contest [71]. The primary reason is the total clock network resistance is much higher than the total inductive reactance [84] for nominal global clock frequency range ($\leq 5GHz$). The functional simulation results with the resulting output current are shown in Figure 3.9.

3.5.2 CMPFFE Analysis

The CMPFFE consumes 5.3% and 26% less silicon area compared to the recently reported CPEFF and DDPFF, respectively. The proposed FF uses 25 transistors and the VM TPFF and MSDFF use 26 and 20 transistors, respectively. While CPEFF and DDPFF use 23 and 22 transistors, respectively. In order to work in all process corners, I used 4 extra transistors in the pulse generation of the later 2 FFs. Figure 3.10 shows the layout of a MSDFF, a TPFF [44], and the proposed pulsed FF.

The CLK-Q delays of the FFs are measured under relaxed timing conditions – the data is stable sufficiently before the arrival of the clock edge. This applies both to the rising edge of the VM signal and the current pulse for the CM clock. In a VM FF, I considered 50%

Table 3.1: The proposed CMPFFE is 87% faster and similar area compared to the TPFF but consumes more static power.

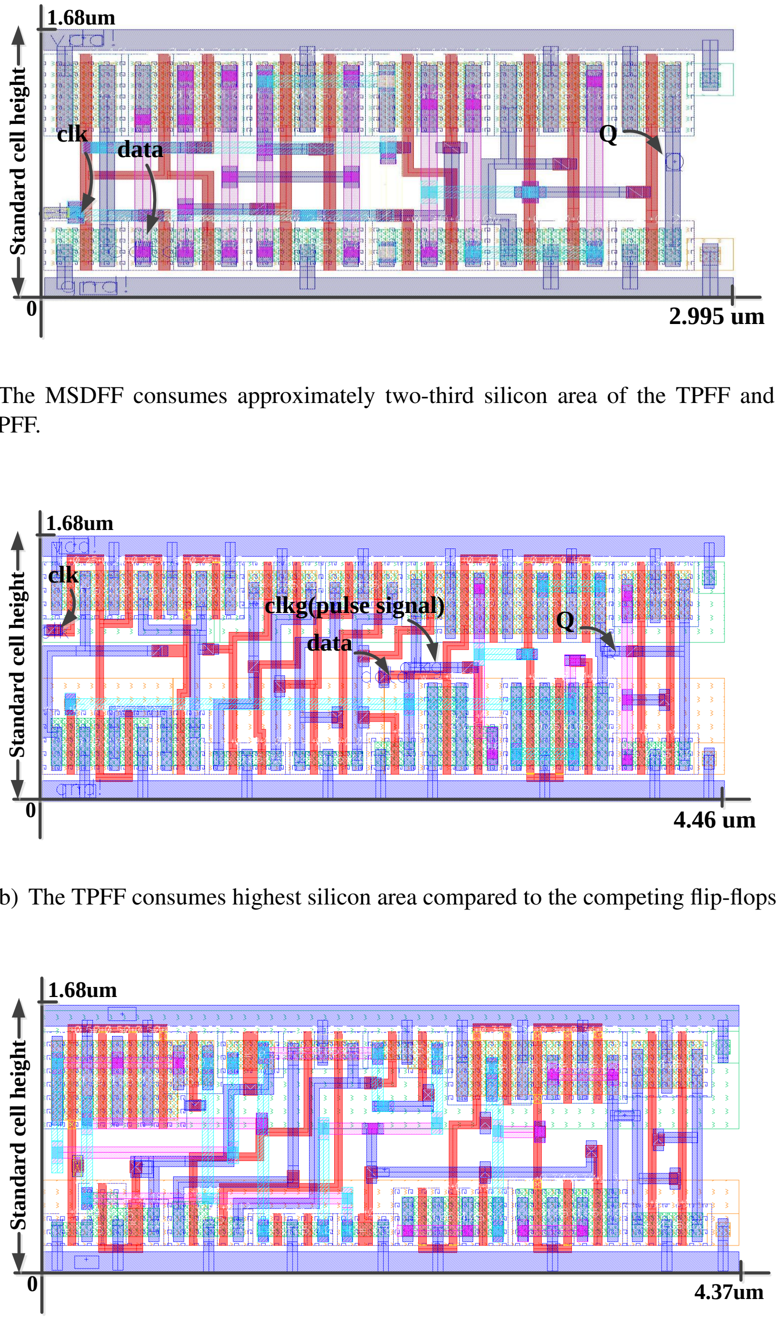
$TypesofFF$	$Area$ (μm^2)	$Delay\ (ps)$			$Total\ Power\ (static + dynamic)\ (\mu W)$			
		$CLK - Q$	t_s	t_h	$2GHz$	$3GHz$	$4GHz$	$5GHz$
$MS\ DFF$	5.03	37.0	21.0	5.0	49	73	98	122
$TPFF$ [44]	7.48	75.5	-46.0	87.0	77	103	137	171
$CPEFF$ [30]	7.75	25.0	-10.0	130	60	89	117	149
$DDPFF$ [1]	9.86	33.0	-5.0	14	62	95	123	155
$CMPFFE$	7.34	40.3	-15.8	46.6	141	151	168	183

input CLK transition to 50% FF output (Q) transition as the CLK-Q delay of a VM FF. Similar to a VM FF, in CM case I considered 50% ideal input current ($2.3\mu A$) transition to 50% Q transition as the CLK-Q delay of CM FF. Table 3.1 shows the maximum CLK-Q delay for both high-to-low and low-to-high Q transitions. Among all the FFs, the CPEFF has the lowest CLK-Q delay. However, low CLK-Q delay and negative setup time also introduce large hold times for an FF. Clearly, the CMPFFE has a lower CLK-Q delay than the TPFF but is only slightly slower than the MSDFF. The DDPFF has 18% lower CLK-Q delay than the proposed FF, but the proposed FF has 13% lower data-to-Q delay.

Figure 3.11 shows the Monte-Carlo (MC) simulations of CLK-Q delay of the proposed CMPFFE under varying process and mismatch conditions at $25°C$. The MC simulations results demonstrated the resiliency of the proposed CM clocking scheme due to process variation and mismatch.

I also measured the t_s and t_h times for each FF. These use the common definition as the time margin that causes a CLK-Q delay increase of 10% beyond nominal. The t_s and t_h of the CMPFFE are $-15.8ps$ and $46.6ps$, respectively. The setup time of the CMPFFE is $1.75\times$ lower than the traditional MSDFF. In addition, recently reported CPEFF has $2.8\times$ more t_h compared to the proposed CMPFFE. The CMPFFE has $3.2\times$ better t_s, but also has $3.3\times$ more t_h compared to the DDPFF.

Table 3.1 presents the total power including both static and dynamic. At low frequen-



(a) The MSDFF consumes approximately two-third silicon area of the TPFF and the CMPFF.

(b) The TPFF consumes highest silicon area compared to the competing flip-flops.

(c) The CMPFFE consumes 2.9% less silicon area compared to the TPFF.

Figure 3.10: The layout of an MSDFF, a TPFF, and the proposed CMPFFE.

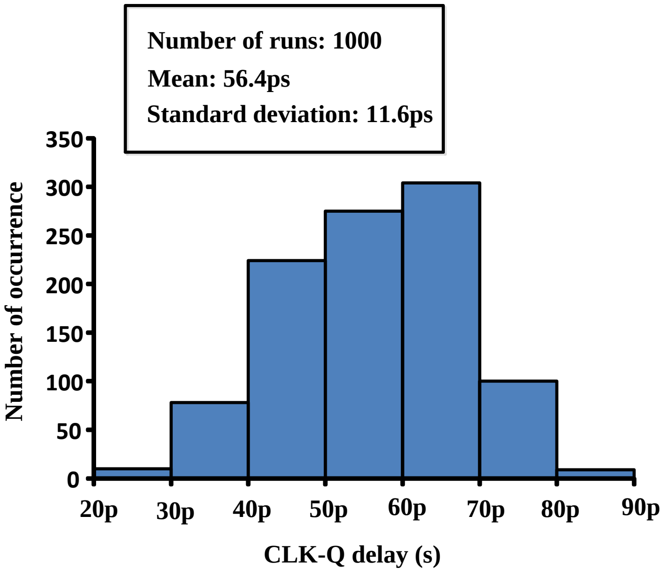


Figure 3.11: The resiliency of the proposed CM scheme is demonstrated through non-uniform Monte-Carlo process variations and mismatch simulations.

cies, the CMPFFE consumes higher power than the TPFF, CPEFF, DDPFF, and MSDFF due to a high static power overhead. However, the dynamic power of the CMPFFE increases proportionally to the frequency at a slower rate than the other VM FFs. At high frequencies, the power consumption of the CMPFFE is comparable to the TPFF and the CPEFF.

The FF power, however, does not represent the overall power consumption of a CDN because interconnect and buffers are major contributors. In Section 3.5.3, I show that the power savings in the CDN is worth the increase in CMPFFE total power despite the additional static power.

3.5.3 CM CDN Analysis

The total system power consumption of a CDN includes the CDN interconnect, buffer power and the FF power consumption. When measuring the total power consumption, I have considered different number of sinks distributed in different size chips followed by the references provided by 2009-2010 ISPD Clock Synthesis contest (i.e. sinks per unit area is the same in each case) [71]. In order to supply the required amount of current to each sink, I used dif-

Table 3.2: The relative sizing of a current-mode transmitter at Figure 3.8 increases $6x$ in each case.

No. of sinks	Chip-edge (mm)	Txs relative sizing
4	0.48	$W_{M1} = 1, W_{M2} = 1$
16	0.96	$W_{M1} \approx 6, W_{M2} \approx 6$
64	1.92	$W_{M1} \approx 36, W_{M2} \approx 36$
256	3.84	$W_{M1} \approx 216, W_{M2} \approx 216$
1024	7.69	$W_{M1} \approx 1296, W_{M2} \approx 1296$

ferent size Txs depending on the size of chip and number of sinks. Table 3.2 presents the Tx sizing for different number of load and chip size. Theoretically, the Tx size should increase $4\times$, since I am increasing number of sinks in the same manner. However, the chip size also doubled in each case, resulting approximately $6\times$ increase of Tx size. The control circuitry in the Tx may require size increases or buffers to drive a larger capacitive load when M1/M2 sizes in Figure 3.8 are increased.

In a VM CDN, the dynamic switching power of the interconnect and CLK load capacitances along with CLK buffers dominate the power consumption. In a CM CDN, the power due to small fluctuations in V_{CM} and the Tx power contribute, but the static power of the CMPFFE dominates. In both cases, the number of sinks and chip dimensions increase the total power consumption.

I use the same H-tree model in both the CM and VM CDN, but buffers drive the VM CDN instead of the CM Tx circuit. The VM buffered network is optimized for an output clock signal with less than $20ps$ slew from $2 - 5GHz$. Since the proposed CM FF is pulsed by nature, the VM CDN considers several pulsed FFs (TPFF [44], CPEFF [30], DDPFF [1]) and also considers the MSDFF as a reference. In order to facilitate normal CMPFFE operation, I used an active low ($\overline{EN}$) signal and also included the required routing power in the CM CDN power calculation.

Table 3.3 shows the power breakdown of the VM and CM CDN's simulation of clock frequencies ranging from $2 - 5GHz$. The total power consumption of CMPFFE system including $\overline{EN}$ signal routing, global reference routing, CM Tx, CMPFFEs power, and CM CDN power. On average, the CM CDN consumes less power than the VM CDN for all sizes of CDN at different frequencies. This is due to the large dynamic power consumption due to the voltage swing (0-to-V_{DD}) in the VM CDN, whereas the CM CDN has negligible voltage swing as shown in Figure 3.9.

Among different FF systems, the CM FFs consume higher power than the other VM FFs. However, VM interconnect power dominates the CM FF power even at small sizes. The real advantage is that the CM CDN power does not increase like the VM CDN power over frequency. Since the fluctuation of V_{CM} is relatively small, the dynamic power consumption of the CM CDN is negligible. At a low $2GHz$ clock frequency, the CM CDN system with the number of CMPFFEs ranging from 4 to 1024 exhibits total power savings of 9% to 32% compared to an MSDFF system. At the same frequency, the proposed system with 1024 sinks shows a total power savings of 33% and 38% compared to the TPFF system and CPEFF system, respectively.

As expected and suggested by Table 3.1, I observed a linear increase in total power savings with the increase of frequency using CM CDN compared to a VM CDN as in Figure 3.12. At $5GHz$ in particular, the CM CDN system exhibits 51% to 67% total power savings considering 4 to 1024 sinks. The primary reason behind that is at high frequencies the relative power consumption of the VM FFs and CMPFFE is nearly equal. At $2GHz$ the CM CDN system saves up to 33% average power compared to other VM CDN. While at $5GHz$ the CM CDN system saves 59% to 62% average power compared to other VM FFs (MSDFF, CPEFF, TPFF, and DDPFF) system as shown in Figure 3.12.

In addition to the dynamic power consumption of VM and CM CDN, I also measured the static power consumption of the largest CDN network with 1024 sinks. The total static power consumption for CM CDN with no clock activity is $154\mu W$. In the same conditions, the total static power consumption of the VM CPEFF system is $186\mu W$. The results are nearly the same and the difference is negligible compared to the dynamic power consumption of each CDN.

Table 3.3: Power saving increases with the increase of frequency utilizing Our CM CDN compared to other VM CDNs.

Fre. (GHz)	# of sinks	CE[1] (mm)	Total CDN power (mW)		Total power consumption including FFs and CDNs (mW)					% Saving compared to			
			VM CDN	CM CDN	MSDFF sys.	TPFF sys.	CPEFF sys.	DDPFF sys.	CMPFFE sys.[2]	MSDFF sys.	TPFF sys.	CPEFF sys.	DDPFF sys.
2	4	0.48	0.72	0.26	0.92	1.03	0.96	0.97	0.83	9.82	19.63	13.95	14.66
	16	0.96	3.03	0.54	3.81	4.26	3.99	4.02	2.79	26.8	34.49	30.03	30.58
	64	1.92	10.62	1.13	13.75	15.54	14.46	14.58	10.16	26.14	34.66	29.74	30.36
	256	3.84	44.28	4.52	56.82	63.99	59.64	60.15	40.62	28.52	36.53	31.9	32.48
	1024	7.69	184.90	18.25	235.07	263.74	246.34	248.38	162.63	30.81	38.34	33.98	34.52
3	4	0.48	1.07	0.30	1.36	1.48	1.43	1.45	0.90	33.89	39.23	36.85	37.89
	16	0.96	4.23	0.58	5.40	5.88	5.66	5.76	3.00	44.42	48.96	46.94	47.82
	64	1.92	15.56	1.15	20.23	22.15	21.26	21.64	10.72	47.00	51.59	49.55	50.45
	256	3.84	66.51	4.91	85.19	92.88	89.29	90.83	43.64	48.77	53.01	51.12	51.95
	1024	7.69	270.02	19.78	344.77	375.49	361.16	367.30	174.71	49.33	53.47	51.62	52.43
4	4	0.48	1.13	0.33	1.52	1.68	1.60	1.62	1.00	34.26	40.37	37.38	38.3
	16	0.96	4.24	1.04	5.81	6.43	6.11	6.21	3.73	35.81	42.04	39.00	39.95
	64	1.92	21.88	1.20	28.15	30.65	29.37	29.75	11.94	57.58	61.03	59.34	59.86
	256	3.84	89.38	5.29	114.47	124.45	119.33	120.87	48.30	57.81	61.19	59.63	60.04
	1024	7.69	361.37	20.03	461.72	501.66	481.18	487.32	192.06	58.40	61.71	60.09	60.59
5	4	0.48	1.70	0.34	2.19	2.38	2.30	2.32	1.06	51.46	55.45	53.75	54.22
	16	0.96	7.18	1.38	9.13	9.92	9.56	9.66	4.31	52.83	56.56	54.96	55.40
	64	1.92	27.14	1.62	34.95	38.08	36.68	37.06	13.33	61.85	64.99	63.65	64.03
	256	3.84	112.05	5.78	143.28	155.83	150.19	151.73	52.65	63.26	66.21	64.95	65.30
	1024	7.69	453.62	20.37	578.55	628.72	606.20	612.34	207.76	64.09	66.95	65.73	66.07

CE[1]: Chip-edge, CMPFFE sys.[2]: Total power consumption of CMPFFE system including $\overline{EN}$ signal routing, global reference routing, CM Tx, CMPFFEs power, and CM CDN power.

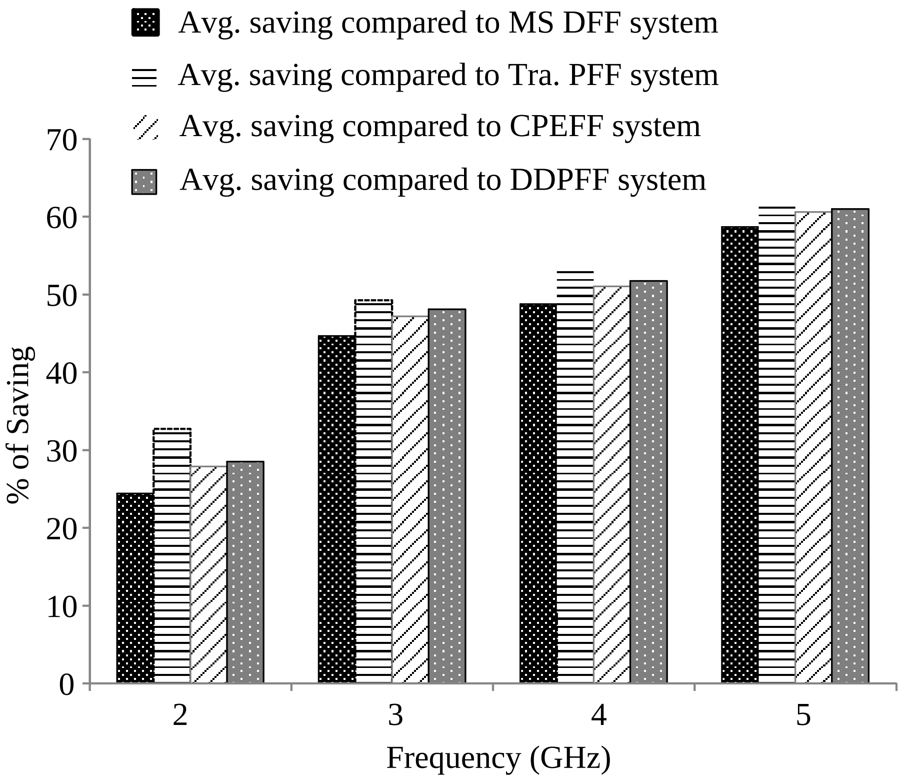


Figure 3.12: The average power savings of the CM CDN system increases proportional to the frequency compared to the other VM FF based CDN scheme Table 3.3.

3.5.4 Reliability Analysis

Unlike an exponentially tapered H-tree [20, 41, 57], I used homogeneous wire sizing from the root to each sink, and verified the maximum current density of CM CDN in the root wire to be $0.275MA/cm^2$ which is less than VM CDN, $0.53MA/cm^2$. This more than satisfies the ITRS suggestion that current density be limited to $1.5MA/cm^2$ [3]. Therefore, electromigration is not a problem for the demonstrated sizes.

3.5.5 Noise Analysis

In order to measure the noise immunity, I compare crosstalk noise simulations for both CM and VM. Figure 3.13 shows the testbench to analyze the effects of crosstalk noise on

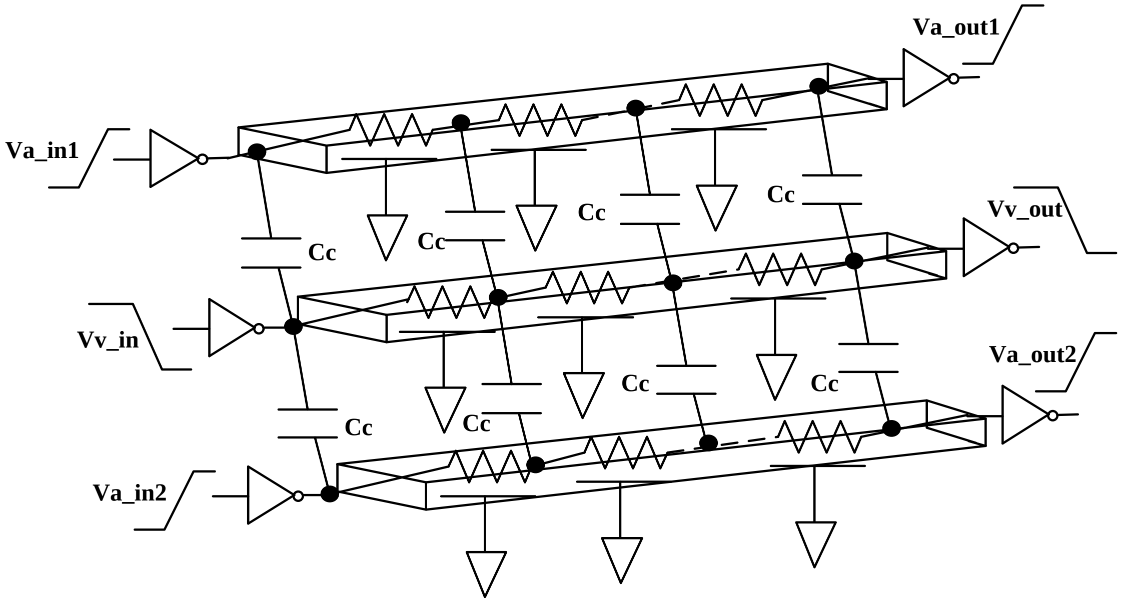


Figure 3.13: Traditional VM schemes are most susceptible to crosstalk noise, when the aggressors are 180° out of phase compared to the victim line.

traditional VM buffer driven interconnects. This experiment is commonly used to quantify the effect of coupling capacitance on dynamic delay due to the switching activity of neighboring nets that have significant coupling to the original circuit. In scaled technologies, traditional VM schemes are most susceptible when the aggressors are 180° out of phase compared to the victim line.

Figure 3.13 mimics the worst case crosstalk by considering 3 parallel interconnections ($5mm$ long) driven by variable impedance drivers/buffers (VM). Hence, the victim line experiences an effective capacitance which is double than the original coupling capacitance. Each $5mm$ interconnect line was buffered/segmented every $1mm$. In this case, the simulation shows that victim line delay can increase up to 35%.

In the CM design, two aggressors are driven by VM buffers, while the victim line is a CM Tx. Simulations suggest that the CM scheme exhibits negligible performance penalty and more robustness to noise because the CM victim line has a much larger capacitance without buffering. This means that the relatively short neighbouring VM aggressor lines have less crosstalk coupling and therefore less influence on CM delay. Unlike VM CDN, the CM CDN requires a global reference voltage and active low enable ($\overline{EN}$) signal routing for the CMPFFEs.

Since the centralized reference voltage and $\overline{EN}$ signal both are the constant voltage, these have a minimum effect due to crosstalk noise. In addition, the wire capacitance is large so it is not affected much.

3.5.6 Variability Analysis

Transistor Threshold Voltage (V_{TH}) may be affected by variations in doping concentration, gate oxide thickness, gate length effective dimension, etc. [19]. Unlike crosstalk noise, V_{TH} mismatch can introduce large skew in a clock network. Hence, quantifying V_{TH} induced clock skew is very critical for the reliability of the clock network.

We considered the worst case corner for both the CM and VM CDN. For CM, this is with V_{TH} variation only in the CM Tx and CM FFs because it does not use other buffers. However, the CM Tx is shared and adds zero skew. For VM, this includes variation in the VM FFs and the clock buffers. Traditionally, CLK skew is measured at the CLK pins of the FFs. However, I wanted to include the impact of variability on the new FF so skew is measured at the FF output. This effectively includes CLK-Q variation in addition to normal CLK skew variation. Figure 3.14 shows an example to calculate skew due to V_{TH} variation at Slow-Slow (ss)-Fast-Fast(ff) corner. In CM CDN, I calculated the time delay considering input CLK signal transition of the CM Tx and the output of both CMPFFEs with ss V_{TH} and ff V_{TH}. The delay difference is the skew in CM case. Similarly, I calculated the skew in VM CDN considering CLK transition at the root buffer to the output of VM FFs with ss V_{TH} and ff V_{TH}.

Table 3.4 shows the effect of worst corner V_{TH} variation on different CDN skews. The traditional VM MSDFF, CPEFF, and DDPFF based CDN show comparable skew at all corner variations. In the ff-ss corner, the CM CDN clock has $17ps$ skew while classic MSDFF based VM CDN has $33ps$. In addition, the proposed CMPFFE-based CM CDN exhibits 51% and 60% less skew compared to the CPEFF and TPFF based CDN, respectively. This is due to fact that the VM CDNs uses buffers to distribute the highly capacitive clock to the sinks.

As mentioned earlier, the performance of CMPFFE is sensitive to the width and amplitude of its input current (i_in). I performed numerous simulations aimed at determining the sensitivity of the clock to output delay of the CMPFFE as a function of the input current. Figure 3.15 shows the variation of this CLK-Q delay relative to input current amplitude and Pulse Width (PW) variations. I define the current sensitivity of the CLK-Q delay as the slope of the

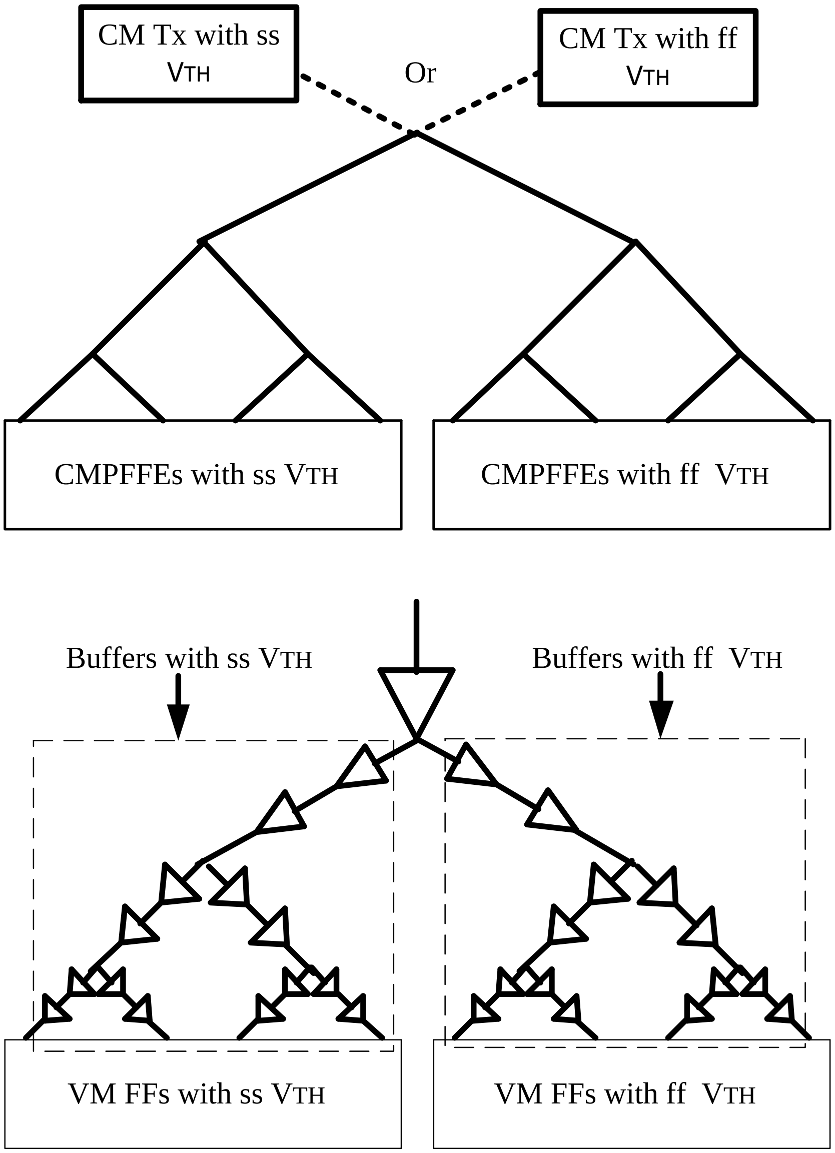


Figure 3.14: In ss-ff corner, the proposed CM CDN has up to 60% less skew compared to other VM CDNs.

approximated linear trendline of the CLK-Q delay curves. I utilized the minimum input current (i.e. $\pm 2.3\mu A$) and varied it up to $2\times$ considering different PW.

At $PW = 70ps$, the current sensitivity on the CLK-Q delay is the highest and while providing the lowest CLK-Q delay compared to the other PWs. On the other hand, at $PW =$

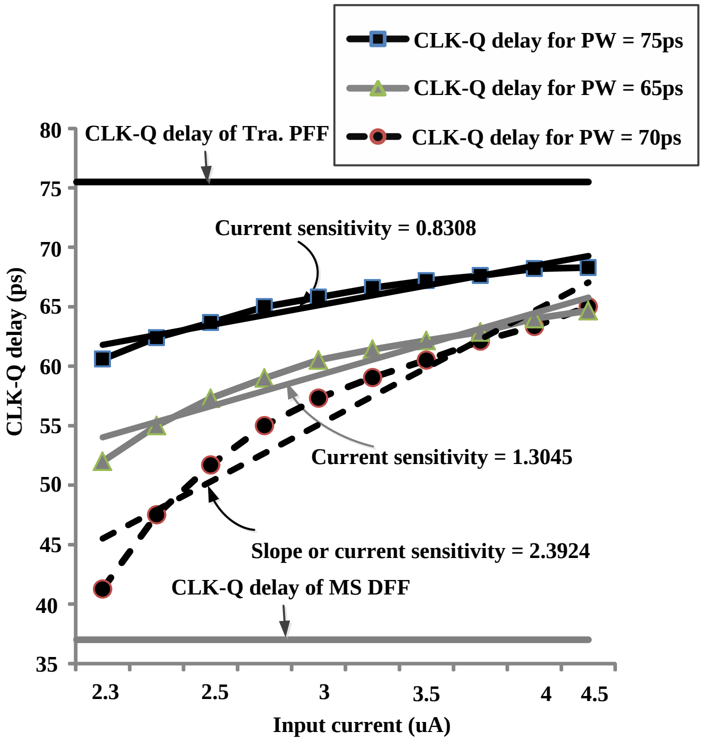


Figure 3.15: The CMPFFE current sensitivity on CLK-Q delay is within the nominal CLK-Q delay of traditional VM MSDFF and TPFF.

$75ps$ the current sensitivity of CLK-Q delay is the lowest but provides the highest CLK-Q delay in comparison to other PWs. The delay variation, however, is within the nominal CLK-Q delay of traditional VM MSDFF and TPFF. Hence, the proposed CMPFFE has a wide input current range while maintaining the optimal performance. This current sensitivity analysis is helpful towards understanding the performance tradeoffs in the proposed CMPFFE with respect to the input current and guides the early stage design of the current Tx.

3.5.7 Supply Voltage Fluctuation

Due to the spatial variation, it is possible that the power supply or V_{dd} could vary at different locations of the chip. Traditionally, designers utilize $\pm10\%$ supply voltage fluctuation from the nominal value. Table 3.4 shows the effect of the supply voltage fluctuation ($\pm10\%$ deviation from 1V supply) on the various CDNs' performance. Similar to the V_{TH} variation, I considered performance metric of CDNs considering the delay variation from root to FFs

Table 3.4: The proposed CM CDN has lower skew due to supply voltage and threshold variation compared to recently reported pulsed FF based VM CDN schemes.

CDN with	Skew (ps)		
	Supply voltage variation		Threshold voltage variation
	$V_{dd} = 0.9V$	$V_{dd} = 1.1V$	ff-ss
MSDFF	10	-18	33
TPFF [44]	12	-21	43
CPEFF [30]	11	-17	35
DDPFF [1]	13	-17	34
CMPFFE	-4	15	17

output. When the supply voltage is low $(0.9V)$, the VM CDN and VM FFs have a positive skew from the nominal supply. The primary reason is the lower overdrive voltage $(V_{GS} - V_{TH})$, where V_{GS} is the gate-to-source voltage of a transistor.

On the other hand, applying high supply voltage $(1.1V)$ in VM CDNs exhibits a negative skew from the nominal case. However, at $0.9V$ supply the proposed CM CDN shows a negative skew compared to the nominal supply voltage. While at $1.1V$, the proposed scheme exhibits a positive skew. This is due to the operating point variation of the CMPFFE and also validates our current sensitivity analysis. Overall, the proposed CM CDN has a lower or comparable skew compared to the other VM CDNs.

3.6 Summary

In this Chapter, I presented the first true CM FF and its usage in a fully CM CDN. The proposed CMPFFE is 87% faster, requires similar silicon area and consumes only 7% more power compared to a traditional VM pulsed FF at $5GHz$. Better yet, the CMPFFE enables a 24% to 62% power reduction on average when used in a CM CDN compared to conventional VM CDNs. The CMPFFE also eliminates the need for complex CM Rx circuitry and/or local VM buffers to drive highly capacitive clock sinks as in previously proposed CM signaling schemes.

Chapter 4

Differential Current-Mode Clocking

In this Chapter, I extend the *de novo* CM clocking concept of Chapter 3 to implement and analyze the first Differential Current-Mode (DCM) clock distribution and a new DCM pulsed D-type FF. Similar to the CMPFFE the CLK input to the FF is a CM receiver and the data input (D) and output (Q) are VM.

We can categorize signaling as differential or non-differential (single-ended). Differential clocks use two wires to send a pair of complementary clock signals. Differential signaling has higher reliability to electromagnetic interference, supply voltage fluctuations, and other sources of common-mode noise compared to single-ended signaling [13, 36, 39]. In addition, DCM signaling has better noise immunity compared to a single-ended CM scheme [46, 54, 67]. However, this comes at the cost of double wiring resources and approximately $2\times$ numbers of buffers compared to a single-ended scheme.

4.1 Existing Differential Signaling Schemes

Unlike traditional buffer-based interconnect signaling, DCM signaling uses a differential CM Tx that sends complementary current pulses at a very low-voltage swing into a pair of interconnecting wires. The interconnect is held at roughly the same voltage and is unbuffered. At the receiving end, a differential CM Rx senses the two complementary currents and ideally converts them into two differential voltages or a single-ended, full-swing output voltage.

A typical non-clock DCM signaling scheme is shown in Figure 4.1 [54]. This scheme uses a self-level-converted driver circuit that limits the output voltage swing from PMOS threshold voltage (V_{tp}) to $(V_{DD} - V_{tn})$. The level converting inverters uses the reverse orientation of a regular inverter where PMOS and NMOS sources are connected to ground and V_{DD}, respectively. Finally, two diode-connected transistor pairs drive two differential interconnect lines and also controls the output voltage swing. However, this kind of driver does not provide sufficient driving capability for large loads and is highly sensitive to noise [25].

This DCM scheme uses a low-swing differential CM Rx circuit called the Modified Asymmetric Source Driver Level Converter (MASDLC) as shown in Figure 4.1 [54]. The MASDLC receives two differential input signal that are transmitted over the interconnect. The receiver converts the received currents into a voltage and amplifies the differential voltage into a full-swing output voltage. In order to increase the robustness of the design, the Rx uses both a common-gate and a common-source amplifier configuration. However, the Rx consumes a significant amount of static power due to double current-mirror stages.

Another prior art uses differential current-sensing for interconnect signaling is shown in Figure 4.2 [46]. The scheme is based on a Modified Clamped Bit-Line Sense Amplifier (MCBLSA) Rx [46]. It utilized the traditional "Fanout Of Four" (FO4) sizing rule for a CMOS buffer chain to design the driver. However, there is no real guideline to design the Tx for differ-

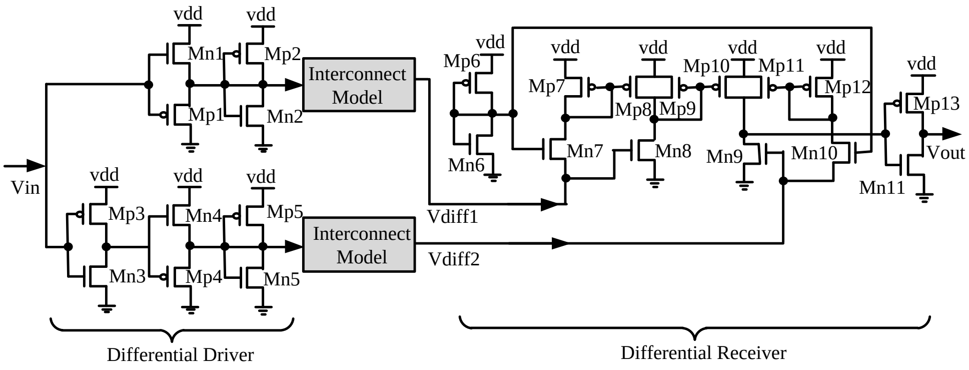


Figure 4.1: A self level converted driver circuit using traditional level converting inverters that limits the output voltage swing from PMOS threshold voltage (V_{tp}) to $(V_{DD} - V_{tn})$ and the MASDLC Rx uses both the common-gate and the common-source amplifier configuration to convert differential input currents to voltage [54].

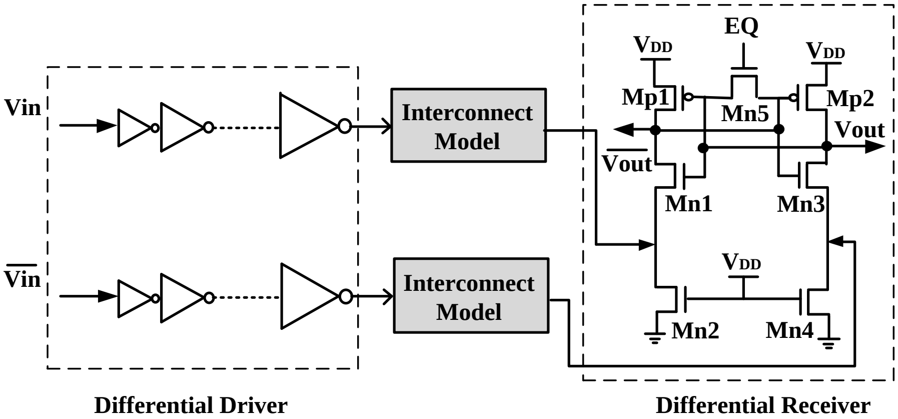


Figure 4.2: The clamped bit-line sense amplifier Rx based DCM scheme uses factor of four sizing rule in cascaded inverters that drive the long interconnect [46].

ent sized interconnects. Moreover, the Tx drives static current into the interconnect while the current is useful during only a fraction of the cycle which results in additional power consumption. The Rx circuit requires an Equalizing (EQ) signal that creates a metastable phase, while the differential input currents break this metastability and help the Rx to produce two complementary outputs. However, this scheme suffers significant static power loss in the metastable phase and also may switch next stages buffer or latches [47].

A point-to-point or N-to-1 differential CM signalling scheme was proposed based on CM sense amplifier [75]. This scheme used two control signals (ten and sen) at Tx and Rx circuits, respectively as shown in Figure 4.3 that solves static power problem associated with the other approaches [46, 54]. In addition to enable the Tx, the Tx control signal (ten) discharges two interconnect lines into the same potential resist unwanted signal communication. Depending on the data input, the Tx injects current into one differential line while the other line draws zero current. The Rx circuit is based on a typical CM sense amplifier that previously used in SRAM read operation [66]. Due to the similarities to SR latch, the Rx can also store the received data and the setup time and hold time of the latch depends on the amplitude of input current. This scheme achieved significant delay and energy improvement over traditional repeater based data transmission scheme. However, the requirement of two internal signals for Tx and Rx circuit increases the complexity of the design and also requires valuable extra metal-routing resources.

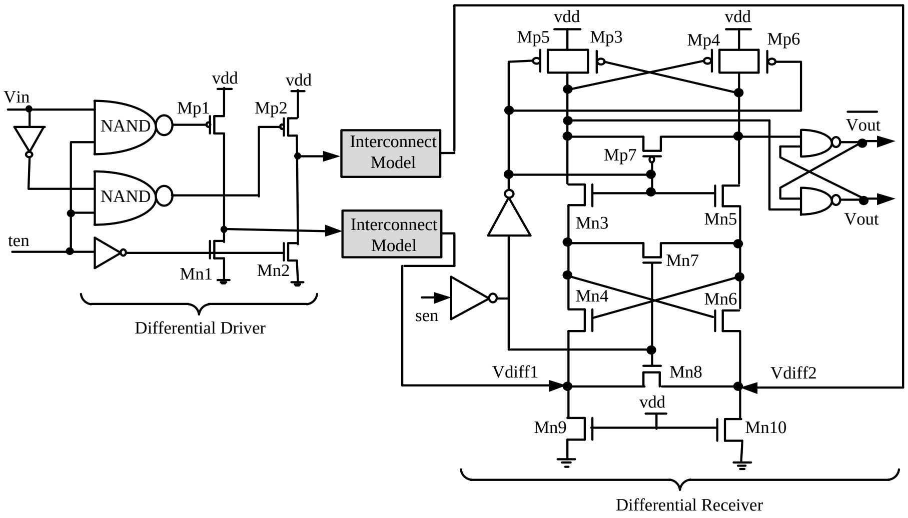


Figure 4.3: The point-to-point or N-to-1 DCM scheme has a Tx that uses a control signal to reduce static power consumption and the CM sense amplifier based Rx latch can store the received data; the amplitude of the input current determines the setup time and hold time of the latch [75].

4.2 Differential Current-Mode Pulsed Flip-Flop

I propose the first differential CM pulsed FF (DCMPFF) in Figure 4.4. The DCMPFF extends the proposed single input CMPFFE [35, 36] in Chapter 3 to have two complementary input currents, I(IN+) and I(IN-). These inputs can be either positive or negative depending on the current direction, however, the DCMPFF is only sensitive when I(IN+) has a push-current and I(IN-) has a pull-current to mimic an edge-triggered behavior.

The DCMPFF has a CC with two reference voltage generators, an inverter-amplifier, an output stage, and a static storage cell. An enable ($\overline{EN}$) signal activates the DCMPFF while the CC uses the push-pull current as input CLK to provide a full-swing output voltage depending on the data input.

A reference voltage generator is built using a diode-connected PMOS-NMOS pair (or polysilicon resistors) as shown in Figure 4.4. The two reference voltage generators create two static currents in PMOS M2 and NMOS M3 and also provide a low-impedance input. The CC

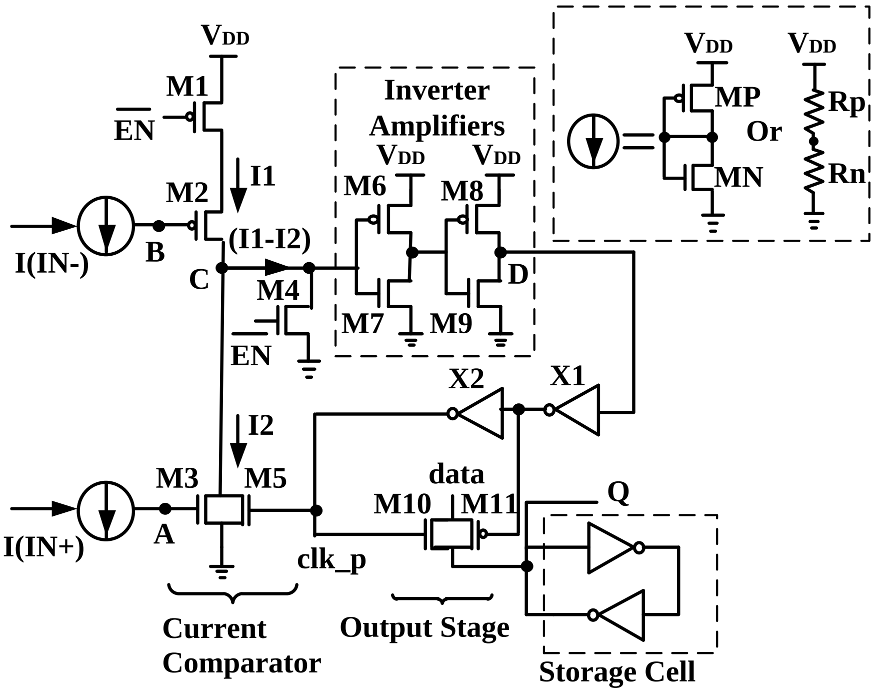


Figure 4.4: The input stage compares the complementary input currents and amplifies the difference to generate a voltage pulse (clk_p) that triggers a register stage to store data.

compares the differential current using inverting amplifier (M6-M7) at node C. After the two-stage amplification, a buffer provides required drive to generate full-swing local CLK pulse (clk_p) that activates the output stage. A feedback connection to M5 limits the clk_p pulse to less than 50% duty cycle. A transmission gate output stage latches data into a storage cell.

The use of a differential input current is more robust to noise compared to a single-ended scheme which will be discussed and analyzed further in Section 4.4. The complementary push-pull currents also helps simplify the design of the current Tx which can generate the currents from a single input voltage.

The CC compares two complementary currents which are combined using an inverter amplifier that enables smaller transistors in the CC (M2-M3) compared to the prior single-ended CMPFF CC [35]. Due to the lower logical effort of M2-M3, the DCMPFF requires less input current and consumes less power.

The representative simulation waveforms of the proposed DCMPFF are shown in

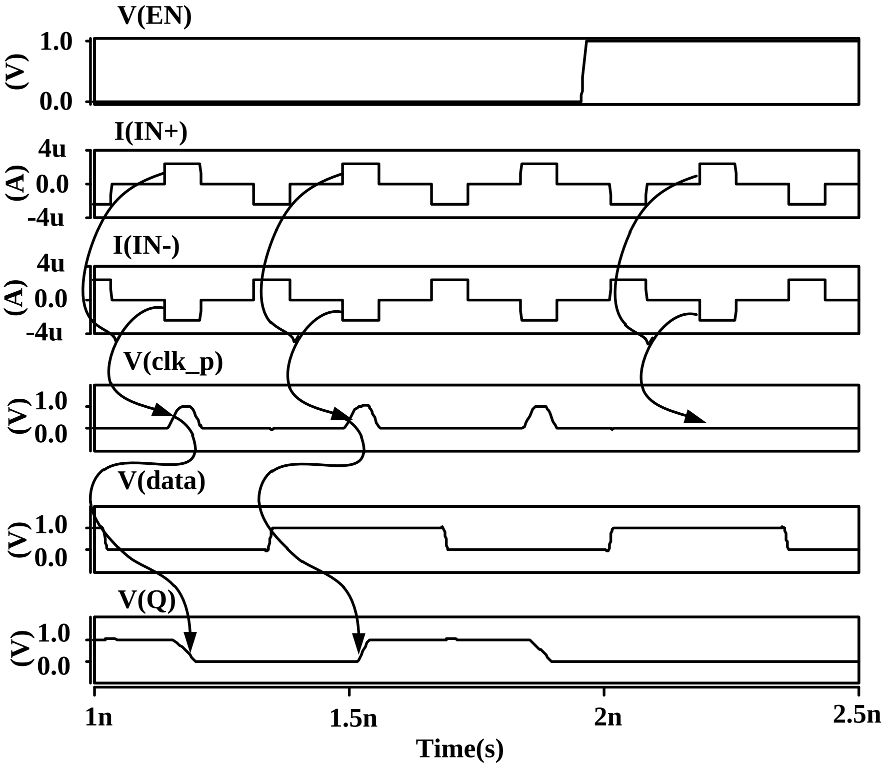


Figure 4.5: Simulation waveforms confirm the complementary current-to-voltage pulse generation (clk_p) that triggers the input data capture.

Figure 4.5 and confirm the internal current-to-voltage conversation. The internally-generated clk_p signal triggers the data storage which is enabled with $\overline{EN}$. The amplitude of the two input currents affect the FF performance by changing the operating point of M2-M3.

4.3 Differential Pulsed Current Transmitter and Distribution

A differential clocking scheme requires a Differential Pulsed Current Tx (DPCTx) that can efficiently provide differential push-pull current into the interconnect and distribute enough current to each sink. The DPCTx is a voltage-to-current converter that receives a traditional VM CLK from a PLL and converts it into a complementary push-pull current signal with minimal voltage swing in the interconnect line. The entire proposed scheme with the DCMPFF,

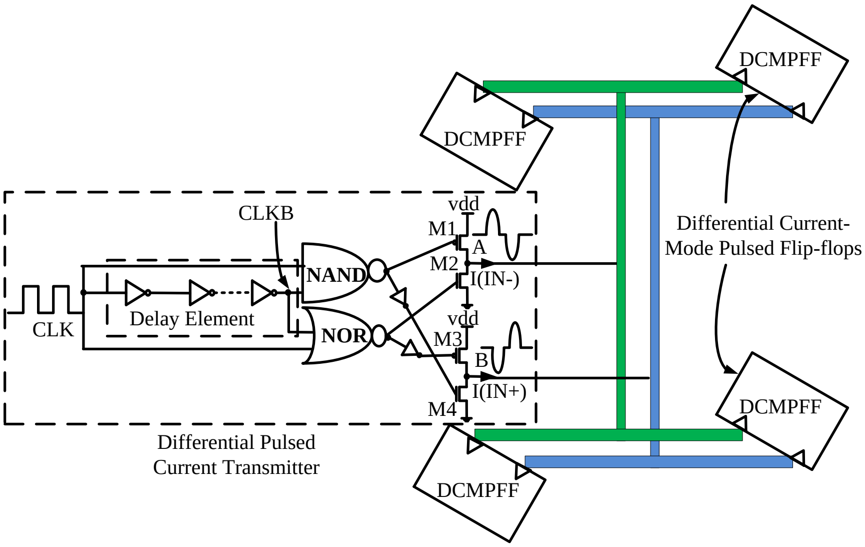


Figure 4.6: The proposed DCM Tx and CDN converts an VM input signal to complementary pulse currents with minimal interconnect voltage swing and distributes current equally to the DCMPFFs.

DPCTx, and CDN is shown in the Figure 4.6. The DCM scheme is based on a CDN that has a similar impedance at each branch resulting equal current to each DCMPFF.

The proposed DPCTx extends the pulsed current Tx discussed in Chapter 3, Section 3.4.2 by using two extra inverters and an extra driver circuit (M3-M4) to generate two complementary currents. The second (differential) current has the same amplitude with one inverter delay of phase difference.

In order to have equal differential current, the DPCTx uses similar sizes for M1-M2 and M3-M4 drivers. The driver sizes are adjusted for current-loss in the long transmission line and supply the required amount of current to each sink. It is important to have appropriate sizing of the wires for both reliability and performance of the CDN. A narrow or highly resistive network will produce distorted output current while a wide network would be low resistance and not have electromigration problems.

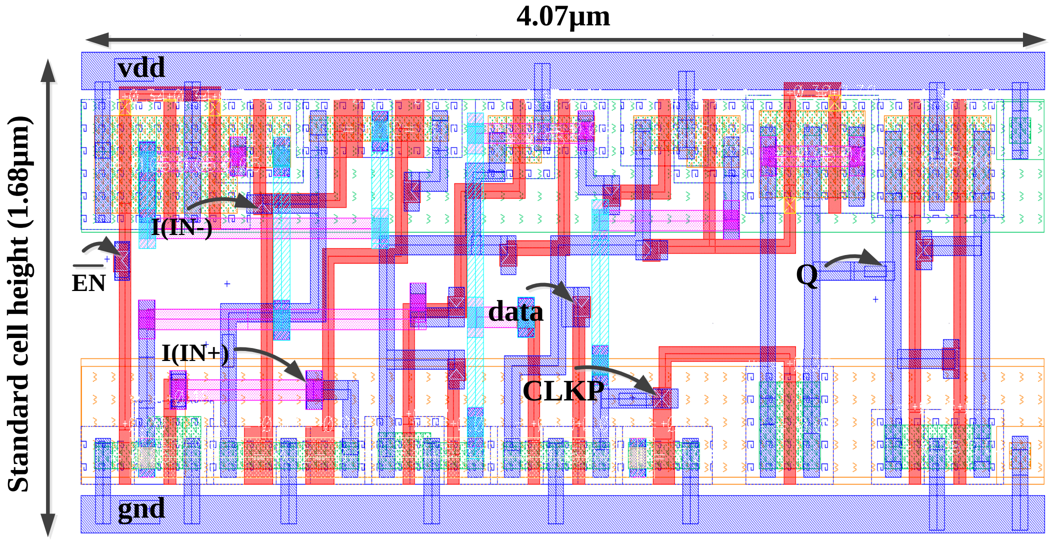


Figure 4.7: The proposed DCMPFF designed with standard cell height and consumes lower silicon area compared to the previous CM FF [35].

4.4 Experiments

4.4.1 Experimental Setup

The circuits are simulated in HSpice with a 45nm CMOS technology model [55]. In order to compare the power, performance, and area, I implemented several designs in layout: an MS DFF, a Tra. PFF [44], a CM Pulsed FF (CMPFF) [35] without enable, and the proposed DCMPFF. The layout areas, nominal CLK-Q delay, data-to-Q (D-Q) delay, and total power are listed in Table 4.1. The performance of the FFs was evaluated considering clock frequencies from 1-5GHz and a 1V supply voltage. The power considers input data at 100% activity with a four FF load.

4.4.2 DCMPFF Results

The DCMPFF consumes 6% less silicon area compared to the CMPFF and uses 23 transistors while the MS DFF and CMPFF use 20 and 25 transistors, respectively. Figure 4.7 shows the layout of the proposed DCMPFF. The CLK-Q delays of the FFs are measured under

Table 4.1: The proposed DCMPFF is 47% faster and consumes 7% more area compared to the Tra. PFF, but more power efficient in the higher frequency range.

Types of FF	Normalized Area	Delay (ps)		Normalized Power (static + dynamic)				
		CLK-Q	D-Q	1 GHz	2 GHz	3 GHz	4 GHz	5 GHz
MS DFF	1.00	37.0	58.0	1.00	1.00	1.00	1.00	1.00
Tra. PFF	1.49	75.5	29.5	1.50	1.57	1.41	1.40	1.40
CMPFF [35]	1.45	45.0	15.0	3.50	3.37	2.47	1.91	1.61
DCMPFF	1.36	39.7	19.7	1.66	1.65	1.21	1.09	0.94

relaxed timing conditions for both the VM and CM instances. In other words, the data is stable sufficiently before the arrival of the VM clock edge or the CM input current pulse.

Table 4.1 shows the nominal CLK-Q delay for both high-to-low and low-to-high Q transitions. Compared to the single-ended CMPFF [35] input current of $\pm 2.3\mu A$ amplitude, the nominal CLK-Q delay of DCMPFF requires only $\pm 1.8\mu A$ and $70ps$ pulse width. Clearly, the DCMPFF has a lower CLK-Q delay than the CMPFF but is only slightly slower than the MS DFF. For each FF, we measured the t_s and t_h. These use the common definition as the time margin that causes a CLK-Q delay increase of 10% beyond nominal. The t_s and t_h of the DCMPFF are $-20ps$ and $95ps$, respectively. The setup time of the DCMPFF is $1.95\times$ lower than the traditional MS DFF while the t_h of the DCMPFF is $1.34\times$ higher than the CMPFF. I also measure the D-Q delay of each FF. The D-Q of the DCMPFF is 66% faster than the VM MS DFF.

I measured the total power consumption of each FF considering the input clock and data switching. For VM FFs, I used a traditional approach [56]. For CMPFFs/DCMPFFs, I used a CM Tx that can produce the required amount of current and the bias voltage to drive the CM FF. First, I measure the total power consumption including the Tx and CMPFFs or DCMPFFs. Then I remove the FFs to measure the Tx power. The difference between these two results is the CM FF power.

In the power measurement, I also consider both static and dynamic power of VM and CM FFs. At a 2GHz clock frequency, the DCMPFF consumes 39.3% and 4.6% more power compared to the MS DFF and Tra. PFF, respectively. However, the power consumption of

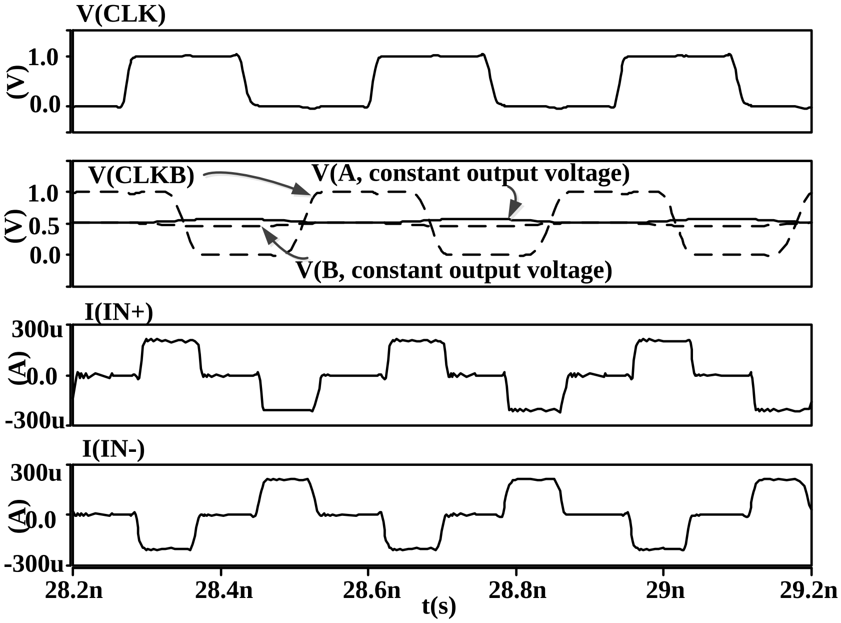


Figure 4.8: Simulation waveforms confirm a VM input is converted to constant CDN voltages and representative complementary current distribution.

the DCMPFF is comparable to an MS DFF at 5GHz. At the same frequency, the DCMPFF consumes 33% and 41% less power compared to the Tra. PFF and CMPFF, respectively. At low frequencies, the DCMPFF consumes higher power than the VM Tra. PFF and MS DFF due to a high static power overhead. However, the dynamic power of the CM FFs increases proportionally to the frequency at a slower rate than the VM FFs.

4.4.3 H-Tree Distribution

In order to validate the functionality of the DPCTx and the proposed DCMPFF in a CDN, I implemented an equal-impedance binary-tree network spanning $1mm \times 1mm$. Each branch of clock tree is modeled as a lumped 3-component Π-model and then connected together to make a distributed CDN model. The interconnect unit capacitance and resistance values are for 45nm CMOS technology [55]. The functional simulation results with the resulting output current are shown in Figure 4.8.

For initial results, our CDN analysis uses a 5-level H-tree distributed in $7.69mm \times$

$7.69mm$ area for both the single-ended CM and VM CDN, but buffers drive the VM CDN instead of the CM Tx circuit. In order to minimize later stages short-circuit power and any timing violation, the VM buffered network is optimized for an output clock signal slew with less than 10% of the minimum operating clock period. In the differential CDN, two such tree networks are routed. All CDNs drive 1024 FFs.

Figure 4.9 shows the total power consumption of the VM, CM, and DCM CDNs simulation of clock frequencies ranging from 1-5GHz. On average, our DCM CDN consumes less power than both the single-ended CM and VM CDN for all frequencies. The obvious reason for more power consumption of VM CDN compared to the other CM/DCM CDNs is due to the voltage swing (0-to-Vdd) in the VM CDN, whereas the CM/DCM CDN has negligible voltage swing as shown in Figure 4.8. The proposed DCM CDN consumes less power than the CM CDN due to the high static power consumptions in the CMPFFs.

As expected at low frequency, the total power of the DCMPFF system is comparable to the VM cases. This is because at low-frequencies the DCMPFF consumes higher power than the VM FFs. However, at high frequencies, the power of DCMPFFs is lower than both the VM FFs. While the power of CMPFFs is higher than the proposed DCMPFFs, due to the large static power consumptions. The VM interconnect power dominates the CM/DCM FF power even at low frequencies. The real advantage, however, is that the DCM CDN power does not increase with frequency like the VM CDN power. Since the fluctuation of common-mode voltage is relatively small, the dynamic power consumption of the DCM CDN is negligible. At 1GHz in particular, the DCM CDN system exhibits 3% to 16% total power savings compared to different single-ended CM/VM CDN. As expected, the power saving increases to 21% to 72% at the high 5GHz clock frequency.

4.4.4 Supply Voltage Fluctuation

I studied the response of the proposed DCM scheme to supply voltage variation. I considered a $\pm 10\%$ voltage fluctuation from the nominal supply voltage. The delay variation for traditional buffered VM scheme ranges from -21ps to 12ps compared to the nominal delay. The delay variation in single-ended CM scheme ranges from -23ps to 28ps. The proposed DCM has delay variation from -23ps to 22ps compared to the nominal voltage delay.

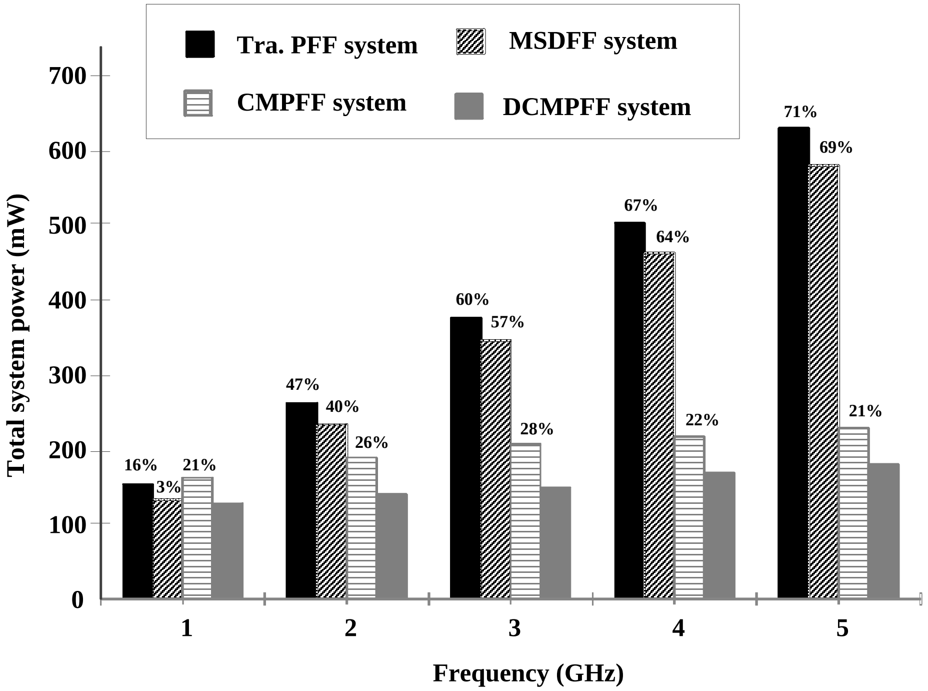


Figure 4.9: The proposed DCM CDN saves 3% to 71% power on average compared to other VM and CM CDNs @ 1-5 GHz CLK.

4.4.5 Electromigration

Since I used homogeneous wires from root-to-sinks for all the clock networks, the root wire carries the maximum current. The VM CDN maximum current density is $0.53MA/cm^2$. As expected, the proposed DCM CDN requires less current compared to the single-ended CM CDN. The maximum current density of DCM CDN in root wire to be $0.24MA/cm^2$ less than single ended CM CDN, $0.275MA/cm^2$. This more than satisfies the ITRS suggestion that current density is limited to $1.5MA/cm^2$ and relieves electromigration threat to proposed CDN wire sizing.

4.4.6 Process Sensitivity

It is impossible to analytically predict the behaviour of a large network due to the combination of the mismatch errors of individual devices. These variations make the modeling of even a small SRAM cell or FF behaviour an intractable task. However, using Monte-Carlo

(MC) simulation the impact of these random parameter variation on FF functionality and performance can be studied. Hence, the resiliency of the proposed DCM scheme is demonstrated through non-uniform MC simulation of process variation and mismatch. The result of this experiment is shown in Figure 4.10. The proposed DCMPFF has a mean CLK-Q delay of 48ps with a standard deviation of 7ps in 1000 runs. This result is much better compared to the pulsed CMPFF. The CMPFF has a mean CLK-Q delay of 55ps with standard deviation of 7.4ps in 1000 runs.

4.4.7 Loading effect

I studied the loading effect of different FFs by changing the driving load of each FF. For any reliable design, it is expected that the FF power-performance with linearly increase with the increase of FF load. Figure 4.11 shows the result of this experiments. Figure 4.11(a) and

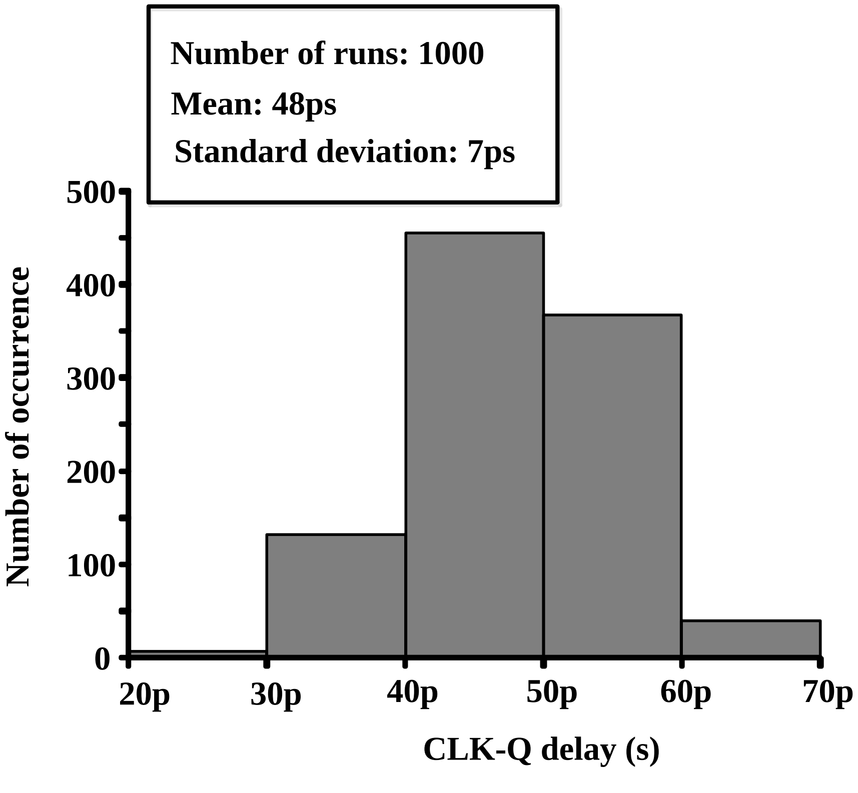


Figure 4.10: Monte-Carlo simulation results ensure the correct functionality and performance of the proposed DCM FF.

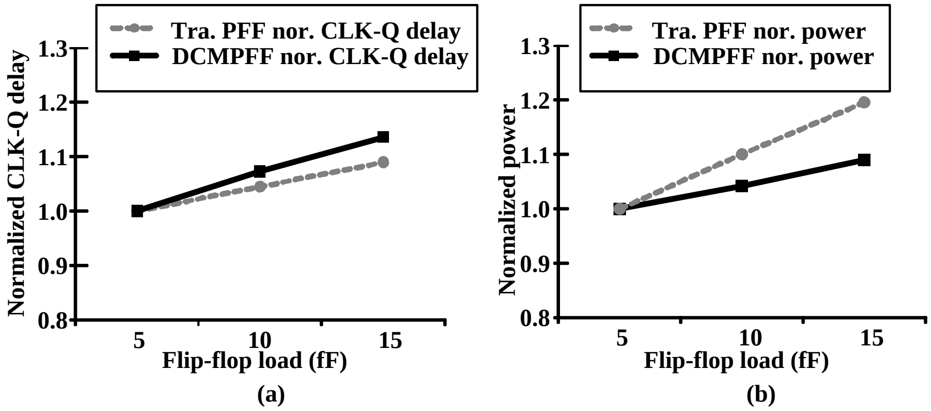


Figure 4.11: The proposed DCMPFF CLK-Q delay and power increase linearly with the increase of FF load and ensure the scalability of the proposed design.

Figure 4.11(b) show the CLK-Q delay and power consumption of proposed DCMPFF and Tra. PFF, respectively. Clearly, the proposed DCMPFF CLK-Q delay and power increases linearly with the increase of FF load and ensures the scalability of the proposed design.

4.5 Summary

In this Chapter, I presented a DCM distribution as an alternative to conventional repeater based VM or CM distribution. The proposed DCM scheme uses a new DCMPFF which is 47% faster, consumes 33% less power and requires 9% less silicon area compared to a traditional PFF at $5GHz$. When applied to symmetric H-tree network, the proposed DCM scheme saves 3% to 71% power compared to a traditional single-ended VM clock at $1-5GHz$ and consumes 21% less power on average compared to a previously reported single-ended CM scheme. In addition, the DCM scheme exhibits 21% less delay variation due to supply voltage fluctuation.

Chapter 5

CMCS: Current-Mode Clock Synthesis

In this Chapter, I present a brief description of previously reported different clock routing techniques and the motivation of the CM clocking issues. Then a Deferred-Merge Embedding (DME) based methodology is extended to route and tune CM clocks along with a thorough analysis of CM pulsed flip-flop properties discussed in Chapter 3 and design using them. In particular, this Chapter demonstrates the CM clocking in asymmetric clock networks using industrial benchmarks [70, 71] and CM latch/FF sizing to minimize global skew.

5.1 Issues in Clock Networks

In a synchronous design, clock signals provide the timing reference for the sequence of data operation. Irrespective of sink locations, the clock waveforms must arrive at various parts of the system at the specified time or time range. In order to distribute the clock signal, an interconnect clock tree and clock buffers or drivers are used. However, the characteristics of the clock signals are primarily impacted on the physical locations of the sinks.

The reliability of a clock network depends on the few key parameters of a clock signal: clock latency or insertion delay, clock skew, clock phase delay or jitter, clock slew rate, and clock pulse width.

Clock latency: Latency is defined as the maximum time/delay a clock signal take to propagate through the clock tree to the sinks. In Figure 5.1 clock latency is shown as the 50% transition of input CLK signal at Root to 50% transition of signal transition of furthest sink sa from the Root.

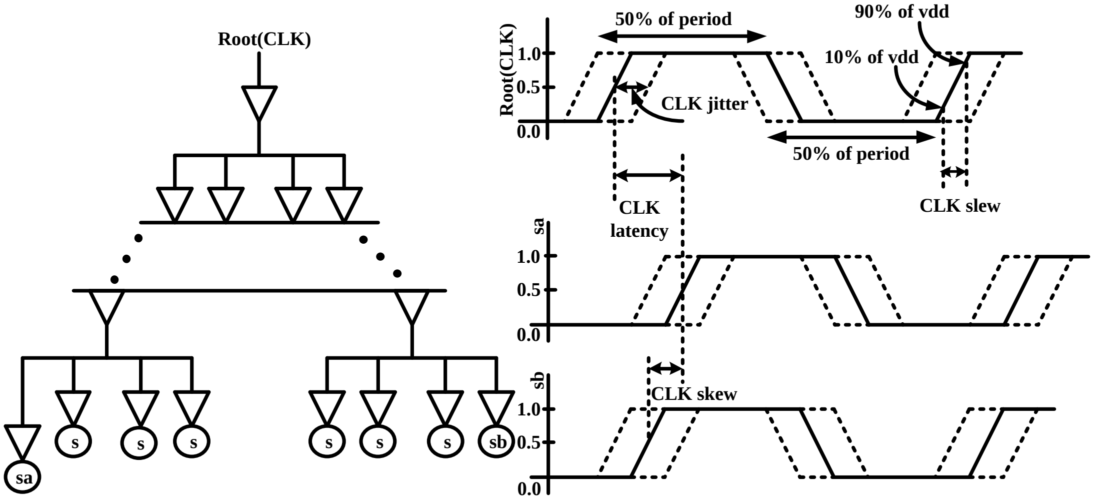


Figure 5.1: The reliability of a clock network depends on the clock latency, clock skew, clock jitter, clock slew rate, and clock pulse width.

Clock skew: Skew is defined as the maximum time difference of arrival time of all the logically connected sinks. In other words, clock skew is the time difference between maximum and minimum clock latency in a CDN. In Figure 5.1 clock skew is shown as the 50% transition of furthest sink signal sa from Root to 50% transition of signal transition of closest sink sb from the Root.

Clock jitter: Jitter is defined as the short-term variations of a signal with respect to its ideal position in time. It can be a positive or negative value. In an on-chip CDN, the cycle to cycle period and duty cycle can change slightly due to the power supply noise and interconnect coupling. The latter attribute can be minimized using proper shielding or increasing the spacing between interconnect tracks. While the power supply noise can be modeled as a source variation similar to process variation.

Clock slew-rate: Slew-rate is defined as the rise time and fall time of a clock signal. Traditionally, clock slew rate is the 10% to 90% for the rising edge or 90% to 10% transition of the supply voltage for the falling edge of a signal as shown in Figure 5.1.

Clock pulse width: is defined as the pulse width of a clock signal (T_{period}, inverse of clock frequency) and specifies the duration of repeated high and low pattern. In CMOS logic value is V_{DD} (1 volts, duration 50% time of T_{period}) and the low value is 0 volts (duration 50% time of T_{period}).

5.2 Motivation

The trip current of a CMPFFE is the minimum current to deposit enough charge at a CMPFFE input so it can store a new value. The clock tree itself remains steady-state at roughly $\frac{V_{dd}}{2}$ and the current pulse arrives nearly instantaneously. Therefore, delay induced skew is not a major issue, unlike VM clocks. In a CM clock, however, an equal amount of current is needed at each FF to prevent timing skew within the CMPFFE. The main complication is that the duration and peak, and hence total charge, of the current pulse, must be within bounds. In addition, the Tx at the root determines the steady-state voltage of the clock network which defines the bias point of the FF clock input.

Balancing the impedance at each wire branch is not a trivial task because it depends on the input impedance of the FF inputs. Prior VM methods could decouple downstream impedance using buffers but CM has an advantage in performance and power by not using buffers. In addition, the Tx at the root determines the steady-state voltage of the clock network which defines the bias point of the FF clock input.

The FF input impedance changes depending on the input current and the bias point set by the Tx, which effectively means that the CMPFFE changes input impedance during a typical clock pulse when there are slight bias fluctuations. The current steered at each branching point depends on each branch's impedance but this, in turn, depends on the downstream FFs and the current that is steered to them. Because of this challenge, previous CM clocking has been restricted to symmetric H-trees [18, 35].

As a result of trip current mismatch, the internal CMPFFE voltage pulse (clk_p) can vary in the time-domain and result in clock skew. This inaccuracy can increase quickly in larger asymmetric networks with large variation in current at the sinks. In the worst case, a CMPFFE may not respond if the trip current is insufficient which can result in a functional failure. Hence, it is desirable to use an automated synthesis tool not only for automation of the routing and impedance balancing, but also to ensure the electrical correctness and functionality.

VM clock synthesis techniques typically use Elmore delay models [21] for initial clock routing and then insert and balance buffers to constrain the network's slew rates. Since the

Elmore delay model is based on the charging/discharging of a capacitance through a resistance, it is not suitable for CM synthesis because CM clocking maintains a steady-state voltage in the entire clock network. Elmore-delay-based clock routing balances delays in clock branches which is not the same as balancing impedances. However, it is a reasonable starting point and can be compensated for by appropriately sizing the Tx and the Rx circuitry in the CMPFFE.

To demonstrate the skew improvement after proposed Tx sizing and CMPFFE sizing stages, I performed synthesis and simulation of different routing techniques in Figure 5.2 on a four sink, asymmetric CM clock distribution using the CM Tx and FF circuits [36] discussed in Chapter 3. Since a symmetric H-tree network doesn't work well with asymmetric distributions, it routes to a fixed location depending on the size of the H-tree. This results in a large $19.1ps$ skew as shown in Figure 5.2(a) [18, 36]. Using a Deferred Merge-Embedding (DME) methodology and CM clocking, I observed a better, but, still considerable $14.8ps$ skew as shown in Figure 5.2(b). The skew improvement is due to the balanced RC product in each sub-tree. Using the proposed iterative Tx sizing methodology with a DME tree, I observe improvement to $3.1ps$ skew as shown in Figure 5.2(c). Sizing the Rx in the CMPFFE further improves the impedance matching and compensates for skew using the clock-to-internal voltage pulse (CLK-clk_p) delay of the CMPFFE. Using this technique along with the DME tree and Tx sizing, the skew is $1.6ps$ as shown in Figure 5.2(d). This research provides an automated methodology for this Tx and CM FF Rx sizing [34].

In addition to skew, it is expected to have lower-jitter induced timing uncertainty in CM clocking compared to a VM scheme due to the absence of buffers in CM CDN and jitter due to crosstalk will be reduced since the net capacitance is larger.

This research provides an automated methodology for the Tx and CMPFFE Rx sizing. It is worth mentioning that the proposed methodology is in stark contrast to the existing impedance balancing VM schemes [49, 68] where clustering and load balancing was achieved using wire and/or buffer sizing [49]. Even timing model independent schemes utilized extra wires and dummy sinks to balance the network [68], but these schemes are only suitable for buffered VM clocking, since the CMPFFE also have varying input impedance.

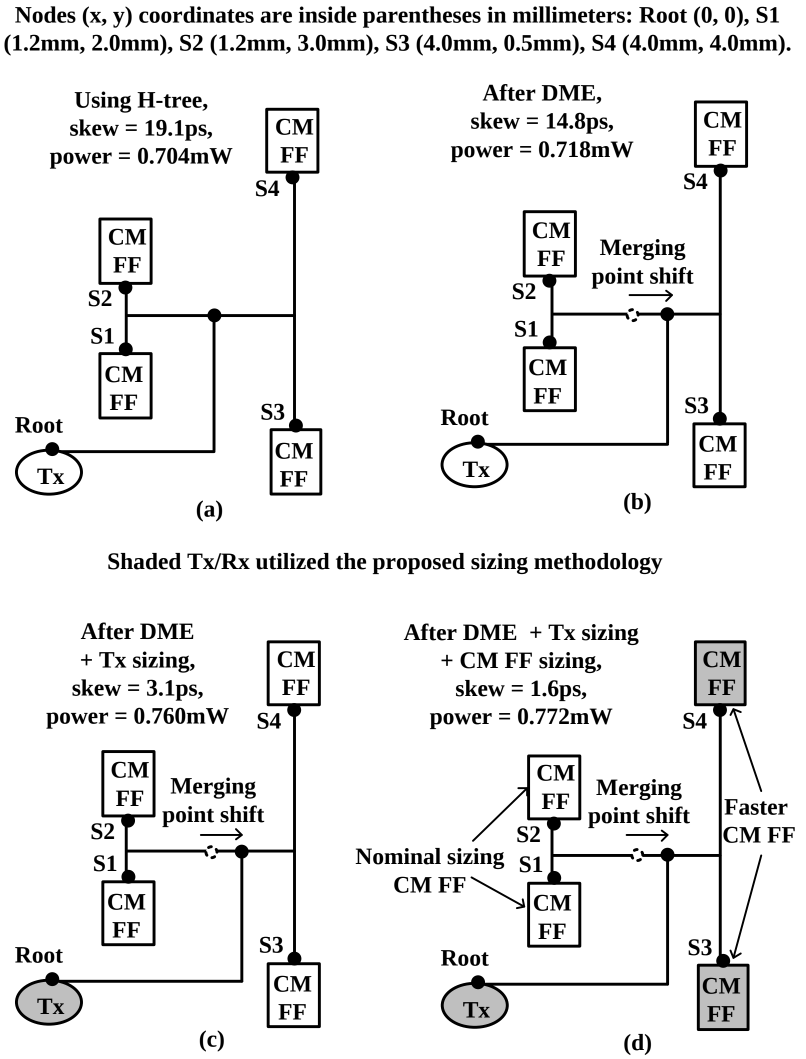


Figure 5.2: Both symmetric and DME VM synthesis techniques introduce large skews ($19.1ps$ and $14.8ps$, respectively) when directly applied to asymmetric CM clock distributions, however, DME with Tx or combined Tx/Rx sizing methodology can improve the clock skew to $3.1ps$ and $1.6ps$, respectively, with almost equal power consumption in each case.

5.3 Overview of Existing Clock Routing Techniques

In a physical design flow, Clock Tree Synthesis (CTS) is performed after placement of macros and standard cells. Hence, it is possible to identify the exact physical location of cells which is needed to establish the tree structure in the design. In general, the CTS process is carried out before routing. There have been many clock routing algorithms proposed throughout the years [38, 49]. The primary goal is to provide a reliable CDN with minimal cost in terms of skew, power, and wirelength [26, 27]. However, minimizing clock skew is the prime aspect and has been studied widely by a number of researchers.

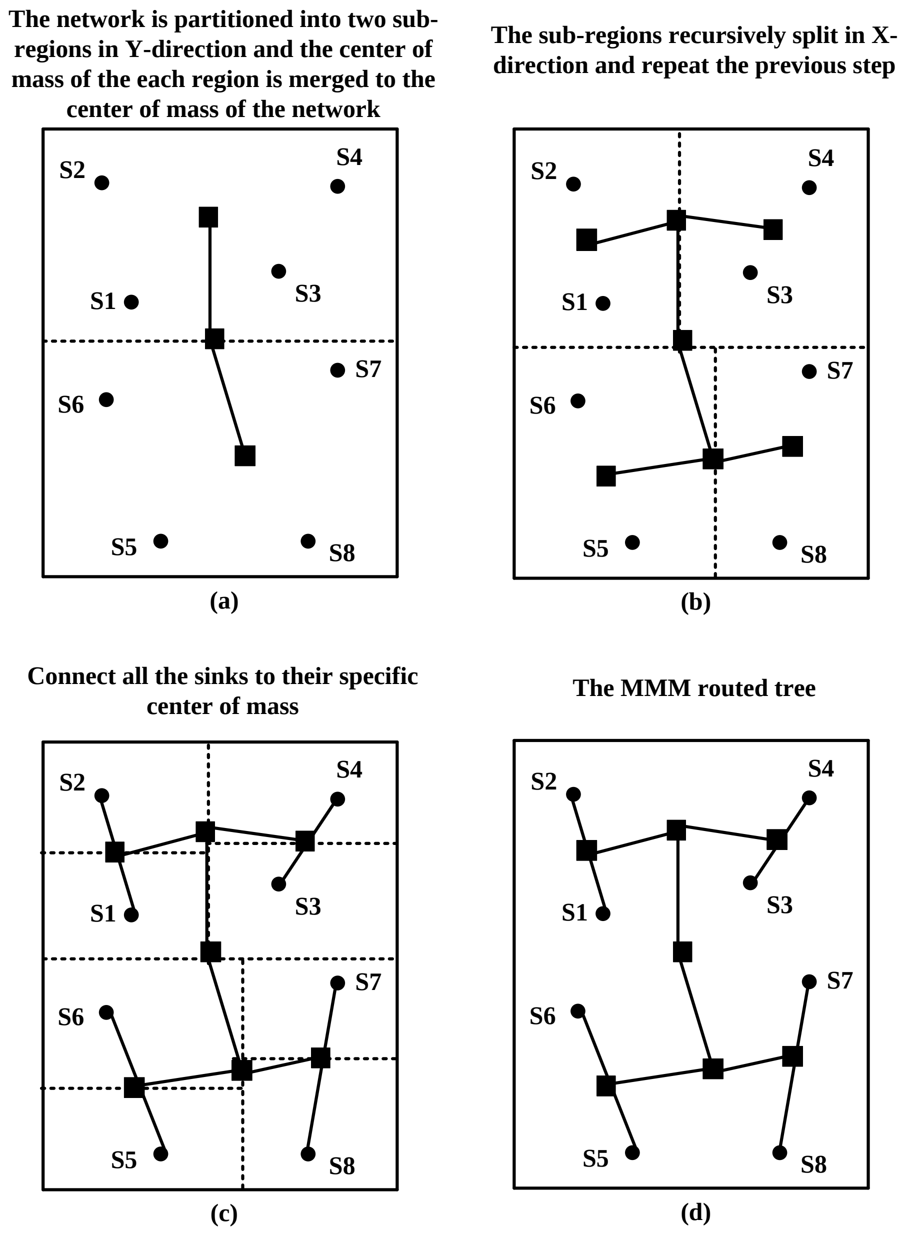
H-tree clock routing is considered as one of the earliest and a perfect synchronization between the clock signals can be achieved before the arrival of clock to the sub-blocks or clock sinks (i.e., FFs) [7]. Previously, H-tree routing is widely used in the IC industry [6, 72] and the proposed CM clocking also adapted this technique in Chapter 3 and Chapter 4. However, H-tree application is not suitable for modern IC or SOC design due to the physical asymmetry of cell locations and blockages.

Another prior art called Method of Mean and Median (MMM) is a top-down approach similar to an H-tree routing algorithm [38]. This algorithm recursively partitions the network into two sets of equal size (median). Then, connect the center of mass of the whole network to the centers of mass of the two sub-networks (mean) to produce a non-linear tree. It keeps partitioning until each network contains only one sink.

Figure 5.3 shows the MMM routing scheme on a 8 sinks region ($s1 - s8$). The network is partitioned into two sub-regions in Y-direction and the center of mass of the each region is merged to the center of mass of the network as shown in Figure 5.3a. The algorithm recursively partitions the regions in altering between X and Y-direction and merge the center of mass to their parent region to build the whole network as shown in Figure 5.3b-Figure 5.3d. This scheme significantly reduce wire length compared to H-tree routing and worst-case timing complexity of $O(nlogn)$ for n clock sinks. However, this algorithm does not ensure zero skews.

The Geometric Matching Algorithm (GMA) is another interesting approach which used the recursive bottom-up method to construct the clock tree [40]. This Algorithm minimizes the total wire length by constructing a set of $\frac{n}{2}$ segments connecting the n endpoints in a pair such that no two segment share endpoint. In order to reduce skew and edges intersection GMA may apply H-flipping. Similar to the MMM algorithm, GMA does not guarantee zero skews and GMA has a worst-case time complexity of $O(n^2logn)$ for n clock sinks.

All these heuristic algorithms (H-tree [7], MMM [38], and GMA [40]) tried to balance wire length to minimize skew and did not consider balancing clock delay with the presence of clock load. Hence, these algorithms are not efficient for tight clock skew optimization for high-



The network is partitioned into two sub-regions in Y-direction and the center of mass of the each region is merged to the center of mass of the network

(a)

The sub-regions recursively split in X-direction and repeat the previous step

(b)

Connect all the sinks to their specific center of mass

(c)

The MMM routed tree

(d)

Figure 5.3: Method of means and medians algorithm has low wire length and consumes low power; however, does not ensure zero skews for all the networks.

performance design. The Zero-Skew clock routing Algorithm (ZSA) [10, 15, 74] significantly improved the clock delay by considering uneven loading and buffer effects. The ZSA is based

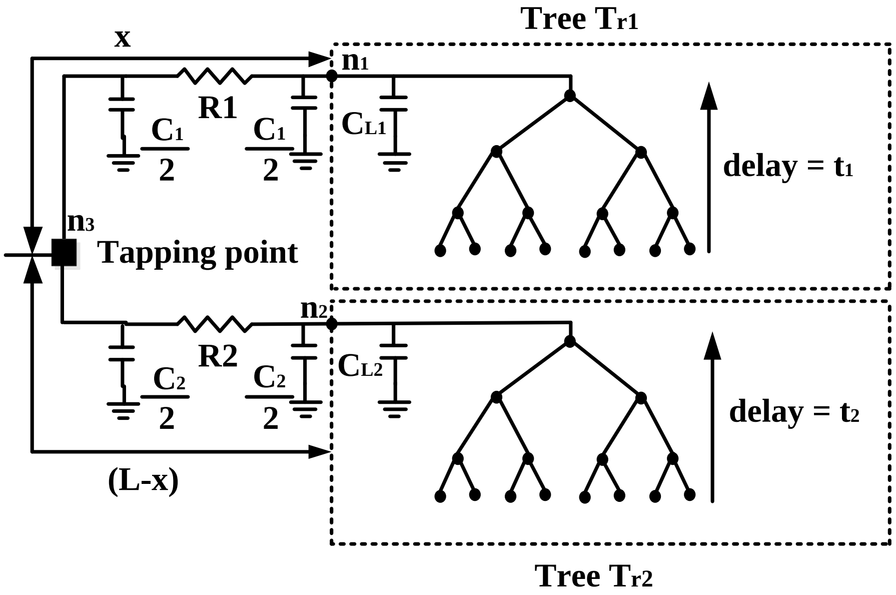


Figure 5.4: The Zero-Skew clock routing algorithm utilizes Elmore delay model to calculate merging distance or the tapping point.

on DME Algorithm and used Elmore delay model to minimize clock skew and wire length.

An example of how ZSA finds the tapping point or merging distance is shown in Figure 5.4. The ZSA merges the two subtrees (T_{r1} connected at n1 and T_{r2} connected at n2) in a tapping point (n3) in by equating the Elmore delay

$$R_1(\frac{C_1}{2} + C_{L1}) + t_1 = R_2(\frac{C_2}{2} + C_{L2}) + t_2 \tag{5.1}$$

where (R_1, C_1) and (R_2, C_2) are the resistance and capacitance of segment x and (L-x, L is the length of wire from node n1 to n2), respectively; C_{L1} and C_{L2} are the input capacitive impedance of subtree T_{r1} and T_{r2}, respectively; and t_1 and t_2 are the propagation delay of subtree T_{r1} and T_{r2} from their sinks to node n1 and n2, respectively. Solving the Equation 5.1, I have

$$x = \frac{(t_2 - t_1) + \alpha L(C_{L2} + \frac{\beta L}{2})}{\alpha L(\beta L + C_{L1} + C_{L2})} \tag{5.2}$$

where α and β are the per unit resistance and capacitance values, respectively; $R_1 = \alpha x L$,

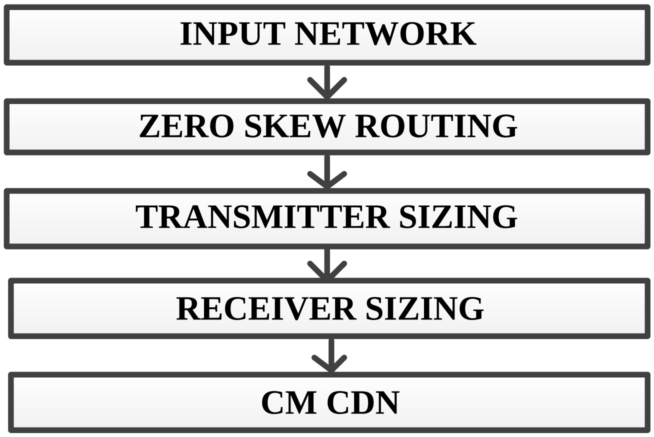


Figure 5.5: The flowchart of the proposed CMCS scheme uses a zero-skew unbuffered clock routing along with stages to set the bias voltage with Tx sizing and Rx sizing to minimize skew and maintain correct functionality.

$C_1 = \beta x L$, $R_2 = \alpha(1-x)L$, $C_2 = \beta(1-x)L$. However, the value of x must be bounded by 0-to-1. Otherwise, this Algorithm requires snaking to find the tapping points.

5.4 Proposed Current-Mode Clock Synthesis (CMCS)

The reliability and overall performance of a CM clocking scheme depends greatly on the Tx and Rx/CM FF circuits and their transistor sizes. The advantage, however, is a tremendous amount of power savings with similar skews compared to existing buffered VM clocking methodologies.

The overview of proposed CMCS scheme is shown in Figure 5.5 which starts with a traditional DME tree construction. While this is not exactly optimal for impedance matching, it generally is a good starting point. It is followed by a stage of Tx sizing to determine the appropriate bias voltage of the network and then an iterative skew improvement through Rx sizing in the CM FFs.

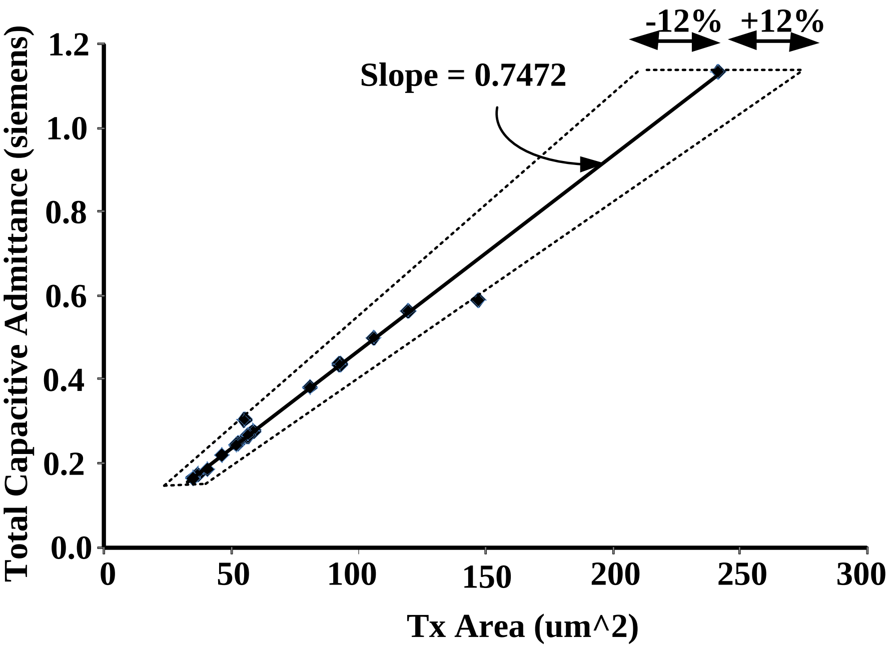


Figure 5.6: CM Tx sizing varies linearly with the total capacitance of the clock network which allows linear fitting for a starting Tx size.

5.4.1 CM Pulsed Current Transmitter Sizing

The proposed CM clock networks are unbuffered and driven at the root by a CM Tx [36]. The CM Tx generates a push/pull current and the devices are sized so that the network maintains a steady-state bias voltage. Since the Tx is large, it may have several exponentially tapered stages of buffers driving it, which are included in our later results. The detailed algorithm for our CM pulsed current Tx sizing is presented in Algorithm 1.

I performed a wide range of simulations on different size and topology networks to relate the Tx sizing with the total capacitive admittance (Y_T) of the network. The result of these experiments are shown in Figure 5.6. The relationship is highly linear between the Y_T and the Tx size.

In order to relate the total driving load/impedance with the Tx size, I calculate the total impedance of the network. However, it is tradition to use admittance, which is simply the inverse of impedance, for parallel networks. The total admittance of a network is proportional to the current as shown in Figure 5.6. I calculate the total admittance of a CDN by considering the total FF load and the RC network. The input admittance of a CM FF is

$$Y_{in} = g_{m1} + g_{m2} = C_{ox} \cdot AR \cdot V_{OV} \cdot (\mu n + \mu p) = \alpha C_{ox} \tag{5.3}$$

where g_{m1}, g_{m2} are the transconductance of the receiving transistors, μn, μp are the mobility of NMOS and PMOS transistors, and C_{ox} is the gate oxide capacitance. The aspect-ratio (AR = W/L = width/length) of Mr1-Mr2 in Figure 5.7 determine the input admittance. V_{OV} is the overdrive voltage of transistor which depends on the bias point. This equation can be simplified using a variable α and assuming all the capacitance in the CDN are in parallel (connected from $\frac{V_{DD}}{2}$ to ground). Now we can write the Y_T of an entire clock network with the FFs as

$$Y_T = \beta(\sum_{i \in sinks} \alpha_i C_{ox} + \sum_{j \in wires} C_{w,j}) \tag{5.4}$$

where $C_{w,j}$ is the wire capacitance of wire j, α_i is the admittance factor of sink i and β is a constant. We can utilize the linearity of Y_T and Tx size to parameter fit β as a starting point. The error bounds suggestion that a $\pm 12\%$ range around the starting point should be considered during optimization. The α_i values are optimized later in Section 5.5.2 when we select CM FF library cells with varying AR sizes. The first part of the Equation 5.4 ensures the total required current at each sink while the latter part helps the Tx to sustain $\frac{V_{DD}}{2}$ voltage and the fraction of energy loss due to non-ideal voltage swing on the interconnect.

Empirically the Tx sizing is convex, so we used steepest descent search to find the best size. The Tx sizing algorithm first calculates the Y_T of the network (Line 3) in the $totalAdmittance(Tree)$ method which applies Equation 5.4. Then it determines the initial Tx sizing (T_{init}) of the network (Line 4) using $sizeTransmitter(Y_T)$. It runs a transient simulation ($simulateTransient()$) and uses $calculateSkew()$ to measure the initial skew (S_{init}) (Lines 5-6). T_{best} and S_{best} are set to the initial values (T_{init} and S_{init}), respectively (Line 7). The T_{init} value is also stored in two temporary variables (T_{newUp} and $T_{newDown}$).

After this, the algorithm sweeps up and down from T_{init} with a step size of δs which is assumed to be 1% of T_{init} using two independent loops (Lines 8-24). The change in Tx device sizes also changes the network bias voltage and the input current of a CM FF that effectively changes the CLK-clk_p delay of the FF in Figure 5.7. In addition, the DME based tree does not guarantee equal impedance of each branch resulting CLK-clk_p delay mismatch.

Algorithm 1 Current transmitter sizing

```
Input: Zero skew routed tree (Tree);
Output: Properly sized transmitter
$Y_T = totalAdmittance(Tree)$
$T_{init} = sizeTransmitter(Y_T)$
$simulateTransient()$
$S_{init} = calculateSkew()$
$S_{best} = S_{init}, S_{new} = S_{init}, T_{best} = T_{newUp} = T_{newDown} = T_{init}$
while $S_{new} \leq S_{best}$ do                              ▷ repeat if improvement or equal
    $T_{newUp} = T_{newUp} + \delta s$                       ▷ $\delta s$ is the 1% of $T_{init}$, sizing up
    $simulateTransient()$
    $S_{new} = calculateSkew()$
    if $S_{new} < S_{best}$ then
        $S_{best} = S_{new}, T_{best} = T_{newUp}$
    end if
end while
$S_{new} = S_{init}$
while $S_{new} \leq S_{best}$ do                              ▷ repeat if improvement or equal
    $T_{newDown} = T_{newDown} - \delta s$                                         ▷ sizing down
    $simulateTransient()$
    $S_{new} = calculateSkew()$
    if $S_{new} < S_{best}$ then
        $S_{best} = S_{new}, T_{best} = T_{newDown}$
    end if
end while
```

This can change the skew of the network and it is imperative to calculate the new skew with the resized Tx. During each iteration, the algorithm compares the new simulated skew (S_{new}) with the previous best skew and retains the best skew (S_{best}) along with corresponding Tx size (T_{best}). The algorithm terminates if there is no improvement in skew. This proposed Tx sizing methodology has worked with any network and our experimental results in Section 5.5.3 will show the quality.

5.4.2 Receiver/CM FF sizing Methodology

To aid skew optimization, I utilize a small set of pre-designed CMPFFE library cells with different input impedances. The input impedance is changed by varying the AR of the input reference voltage generator (Mr1-Mr2) diode-connected inverter circuits in Figure 5.7 as modeled in Equation 5.3. However, it is necessary to have equal AR for both the input reference voltage generator and local reference voltage generator (Mr3-Mr4) to measure the correct trip current of a CM FF. Because of that we change the AR of both voltage generators simultaneously. This results in a voltage variation at the input of the current-comparator and can move the bias-point. The variation of bias voltage also varies the CLK-clk_p delay of CMPFFE. These results are shown later in Section 5.5.2.

The proposed CMPFFE sizing methodology balances the root to sink admittance of an unbalanced tree by selecting among the available CMPFFE library cells. Since these cells have different admittance, they have differing internal CLK-clk_p delays which can be used to balance any skew. I approach the CM FF sizing problem by starting with a median CLK-clk_p delay FF and replacing those that have lower or higher impedance (with faster or slower versions), respectively.

The detailed Algorithm for the CMPFFE sizing is shown in Algorithm 2. The FFs are initially set to the median size to allow them to be made faster/slower. After a transient simulation, the algorithm calculates the S_{init} (Line 3-4) and sets S_{best} as S_{init} (Line 5). The

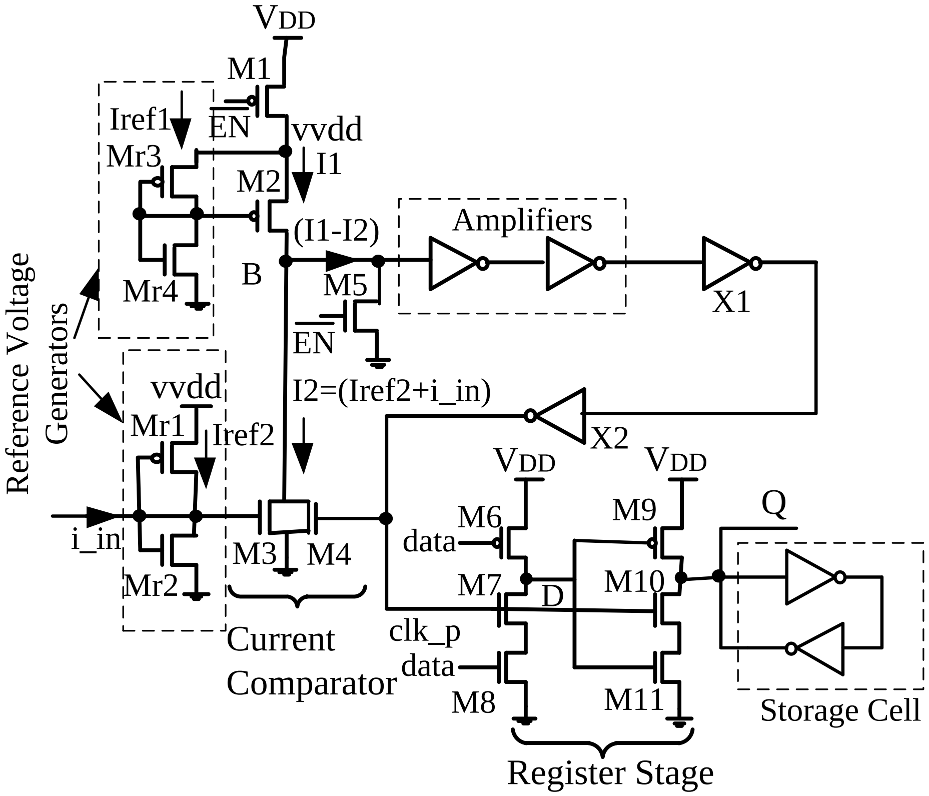


Figure 5.7: Sizing of CM FF reference-voltage generators changes the FF internal CLK-clk_p time resulting in faster or slower FF with no impact on FF timing constraint [36].

Algorithm 2 CM Pulsed FF sizing

```
1: **Input:** Zero skew routed tree (Tree) without buffers; properly sized Tx; a set of CMPFFE library cells (L); skew bound ($SB$)
2: **Output:** Properly sized CM pulsed FFs that meets the skew requirement
3: $simulateTransient()$
4: $S_{init} = calculateSkew()$
5: $S_{best} = S_{init}$
6: **while** $S_{best} > SB$ **do**
7:     $(critSinks, goodSinks) = findCriticalSinks()$
8:     $d_{max} = maxDelay(goodSinks)$
9:     $d_{min} = minDelay(goodSinks)$
10:    **for all** $s_i \in critSinks | d_{s_i} < d_{min}$ **do**
11:        $slowerFlipflop(s_i)$                              ▷ replace with slower FFs
12:    **end for**
13:    **for all** $s_i \in critSinks | d_{s_i} > d_{max}$ **do**
14:        $fasterFlipflop(s_i)$                              ▷ replace with faster FFs
15:    **end for**
16:    $simulateTransient()$
17:    $S_{new} = calculateSkew()$
18:    **if** $S_{new} < S_{best}$ **then**
19:        $S_{best} = S_{new}$
20:    **else**
21:        $break$                                            ▷ terminate if no improvement
22:    **end if**
23: **end while**
```

Algorithm search over the sinks' timing information and determine the set of sinks that need improvement in $findCriticalSinks()$ (Line 6). Then, the Algorithm iteratively resizes the critical CMPFFEs until its meet the skew bound (SB) (Lines 7-23).

The $findCriticalmethod()$ function identifies the largest cluster of FFs in any skew bound window as the "good" sinks. Algorithm 3 does this by iterating over a list of sinks sorted by their delay (D_{in}) (Line 3) and counting the number of sink delays d_j within a skew bound (SB) from sink i with delay d_i (Lines 7-12). The largest number of sinks in a window ensures that the fewest CMPFFEs will be returned in the critical sink set C and need to be adjusted in Algorithm 2. These "critical" sinks are outside the optimal window can be either too fast or too slow.

Algorithm 3 has a worst case runtime complexity of $O(n^2)$, where n is the number of sinks. However, the SB is small and we only look into the set of sinks within a skew bound, which severely limits the second n. This makes the proposed Algorithm linear in practice. In addition, using linear time maximal sum Algorithm [11], the proposed Algorithm 3 could be speed-up to $O(n)$. However, the runtime is dominated by simulation and not the Algorithm itself so we did not do this.

During each iteration of Algorithm 2, I calculate the maximum delay (d_{max}) and minimum delay (d_{min}) of the "good" sinks (Line 8-9). Then two consecutive loops iterate over the fast and slow critical sinks, respectively, and choose a faster/slower CM FF from the library cells (Lines 10-15). A transient simulation calculates the new skew (S_{new}) and stores the minimum value to S_{best} after comparison (Lines 16-22).

The proposed CMPFFE sizing algorithm converges to a minimum skew after either no skew improvement is seen or the skew bound is achieved. It is worth mentioning that the CMPFFE are sized to meet the SB for a fixed Tx size, which was determined in the previous stage. The Tx is not sized after the receivers. So there is no need to size the Tx again. In addition, the CM FFs are very fast and Algorithm 1 ensures proper functionality of each FF by properly sizing the CM pulsed current Tx. FF metastability is usually due to the input arriving during a clock transition. The proposed CM FF still has setup and hold times like VM FFs to avoid any such problems.

Algorithm 3 Finding critical sinks

```
Input: Time delay of each sink ($D_{in}$); set of sinks ($S$); skew bound ($SB$)
Output: Set of critical sinks ($C$) beyond the skew bound and good sinks set ($G$)
$D = sort(D_{in})$                                ▷ sort all the sinks based on their time delay
$index_{best} = 0, cnt_{best} = 0, i = 0$
while $i++ < |D|$ do
    $j = i + 1, cnt = 0$
    while $d_j < (d_i + SB)|(d_i, d_j) \in D$ do
        $cnt++, j++$
        if $cnt > cnt_{best}$ then
            $index_{best} = i, cnt_{best} = cnt$
        end if
    end while
end while
$G = s|d_{index_{best}} \leq d_i \leq d_{index_{best}+cnt_{best}}$        ▷ store all the good sinks in $G$
$C = S \backslash s$                                      ▷ store all the critical sinks in $C$
```

5.4.3 Impact of CMPFFE Sizing on Timing Constraint

In this Section, I will discuss the impact of CMPFFE sizing on conventional edge-triggered clocking. Consider the sequential circuit shown in Figure 5.8. Assume that as a result

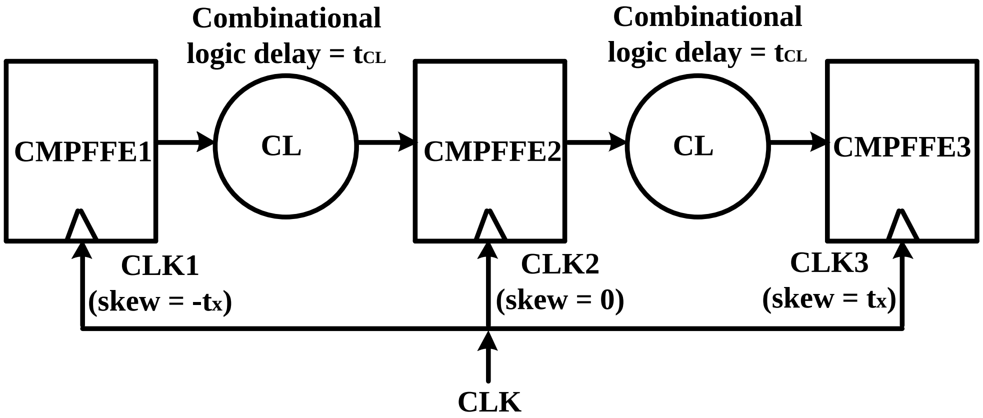


Figure 5.8: Although the proposed CMCS scheme uses nominal, faster, and slower CM FFs by adjusting $CLK - clk\_p$ delay, it has no impact of FF timing constraints.

of the clock distribution, there are static skew $(-t_x)$, zero, and $(+t_x)$ in left, middle, and right CMPFFE, respectively. Ignoring the jitter effect, I can write the timing constraint related to clock period (T_{CLK}^{FFi}) for the each FF to determine the minimum available time to perform the required computation in the combinational logic as

$$T_{CLK}^{FF1} \geq t_{CLK1-Q} + t_{su} + t_{CL} - t_x \tag{5.5}$$

$$T_{CLK}^{FF2} \geq t_{CLK2-Q} + t_{su} + t_{CL} \tag{5.6}$$

$$T_{CLK}^{FF3} \geq t_{CLK3-Q} + t_{su} + t_{CL} + t_x \tag{5.7}$$

where t_{CLKi-Q} is the CLKi-to-Q delay of FFi, t_{CL} is the combinational logic delay, t_{su} is the FF setup time. However, the t_{CLK-Q} delay of CMPFFE is the combination of $t_{CLK-clk\_p}$ and t_{clk_p-Q} delays. In order to have a zero skew for Figure 5.8 sequential design, I can utilize methodology presented in Section 5.4.2. The proposed FF sizing methodology adjust the clock skew for CMPFFE1 by using a slower FF as

$$t_{CLK1-Q} = t_{CLK1-clk\_p} + t_x + t_{clk\_p-Q} \tag{5.8}$$

On the other hand, it uses a faster FF for CMPFFE3 and adjust the timing as

$$t_{CLK3-Q} = t_{CLK3-clk\_p} - t_x + t_{clk\_p-Q} \tag{5.9}$$

In addition, The proposed methodology uses a nominal CLK-Q delay CMPFFE as CMPFFE2, resulting in a zero skew design.

5.5 Experiments

5.5.1 Experimental Setup

I implemented the proposed CMCS scheme in C++ and Python. Simulations were run on an Intel Core i5-3570 Ivy Bridge 3.4GHz quad-core processor. I validated the proposed methodology using 45nm ISPD 2009 and 2010 industrial Benchmarks [70, 71]. ISPD 2009 benchmarks are derived from real IBM ASIC designs. These benchmark circuits are distributed in $50.4 - 275.6mm^2$ area and consists of 81-623 evenly/unevenly distributed sinks with equal or unequal sink capacitances. ISPD 2010 benchmarks are derived from real IBM and Intel Microprocessor designs. The 2010 benchmark circuits are distributed in $1.4 - 91.0mm^2$ area and consists of 981-2249 nonuniformly distributed sinks with different loading. The designs were optimized for 1V supply voltage and clock frequencies range from 1-3GHz. Traditionally, 5-10% of the clock period is allocated for clock skew, so we used a clock skew bound of 70 ps for 1 GHz clock frequency. Traditionally worst case slew rate is defined as 10% of the clock period. For the proposed CM clocking schemes, we used 10% slew bound. It is worth mentioning that at steady state the CM clock tree remain roughly around $\frac{V_{dd}}{2}$, hence we only considered worst case slew rate at the clk_p signal of CM FF. The CM Tx and Rx/FF [36] were designed using the FreePDK 45nm CMOS technology [55]. I used HSPICE to measure power and performance for all results.

The clock tree is routed with minimum wire length by incorporating Balanced Bipartition (BB) with DME [10, 74] and the final tree nodes are connected to the CMPFFEs. The clock tree and the CMPFFEs are driven by a single pulsed current Tx. In addition, I followed ISPD 2010 High Performance Clock Network Synthesis Contest guideline to model the clock network as a distributed RC model [70, 71]. The CM Tx, tree, and the CMPFFEs compose the entire CM network. Figure 5.9 shows the resulting DME routed bufferless CM CDN for the

ISPD 2010 Benchmark circuit 06. In proposed CMCS scheme, the total power consumption includes the CM pulsed Tx power, parasitic power, and the total CMPFFE power.

The CMCS methodology uses library cells of CMPFFE with different AR and hence input impedance and CLK-CLKP delay resulting in “slower” and “faster” FFs. Here “faster” and “slower” refers to the smaller and larger CLK-clk_p delays, respectively. I calculate global clock skew at the FF’s internal clock pins (clk_p), so that changes in CLK-clk_p delay are included in the skew component of timing constraints and do not change the setup time and hold time.

It would be interesting to compare the CMCS results with the ISPD 2009 and ISPD 2010 winners results. But, the winning teams consider local skew minimization resulting in wire snaking and extra buffers. For example, using the 01 benchmark circuit, the ISPD 2010 winning team used $198.3pF$ capacitance, while the implemented VM network requires $93.7pF$ capacitance. Overall, ISPD 2009 and ISPD 2010 winners consume significantly more capacitance resulting more than double power consumption compared to our implemented buffered VM networks, hence in the final comparison I eliminated ISPD winners result.

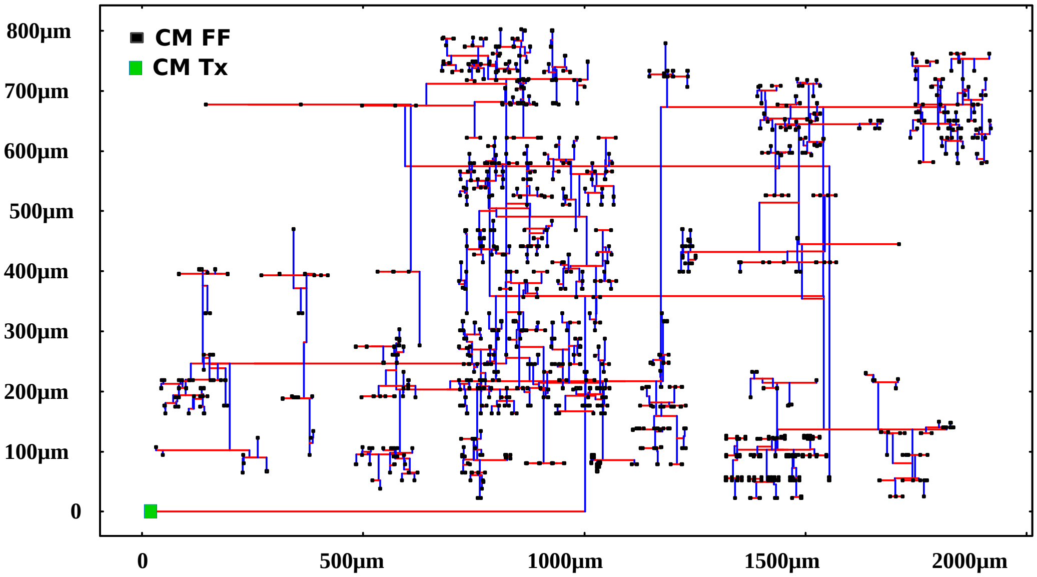


Figure 5.9: The resulting DME routed bufferless CM CDN for the ISPD 2010 Benchmark circuit 06.

Since the previous Tx sizing methodology [36] does not work with asymmetric networks, I used a state-of-the-art buffered VM methodology for comparison. The VM tree is routed using a common industry method with minimum wire length [10, 74] and the buffers are inserted to meet the skew and slew constraints (10% of the clock period) [73]. For the VM buffered network, the total power consumption includes CDN buffer power, clock tree parasitic power, and VM pulsed FF [1] power. Both the VM and CM schemes receive a traditional voltage clock from a PLL/CLK divider at the root. The input CLK signal slew rate is 10% of the CLK period.

5.5.2 CM FF Library Cells

Similar to a VM FF, in the CM case I considered 50% ideal input current ($3\mu A$) transition to 50% Q transition as the CLK-to-Q delay of CM FF. For setup (t_s) and hold time (t_h) times I used the common definition as the time margin that causes a CLK-to-Q delay increase of 10% beyond nominal. The t_s and t_h of the median size CM FF are $-15.8ps$ and $46.6ps$, respectively. Figure 5.10 shows an analysis of the CMPFFE library cells with the nominal input current of $\pm 3\mu A$ and $70ps$ pulse width. In this analysis, I vary the AR of CMPFFE reference voltage generators and measure the corresponding CLK-clk_p delay. I observed a linear relationship between CLK-clk_p delay with AR. Particularly, the CLK-clk_p delay of the CMPFFE increases with the increase of AR by increasing input impedance as shown in Equation 5.3. Hence, I utilized this characteristic to build our CMPFFE library cells with different CLK-clk_p delay. It is worth mentioning that, similar to a FF output (Q) signal, the clk_p act as both terminal and voltage pulse.

The proposed CMCS utilized 13 CMPFFE library cells (a median size and 6 faster and 6 slower) with $\pm 30ps$ CLK-clk_p delay variation from the nominal delay value.

In order to tackle skew issues, the proposed CMCS utilized 13 CM FF library cells (a median size and 6 faster and 6 slower) with $\pm 30ps$ CLK-clk_p delay variation from the nominal delay value. It is expected that the use of different sizing CMPFFE requires different FF area and may add area overhead to the overall design. However, It is possible to have zero area overhead for different size FF. Figure 5.11 shows the layout of fastest, median, and slowest CLK-clk_p delay CMPFFE. In Figure 5.11, the P_n and N_n indicates the sizing reference of PMOS and NMOS, respectively; corresponding to reference voltage generator of median size

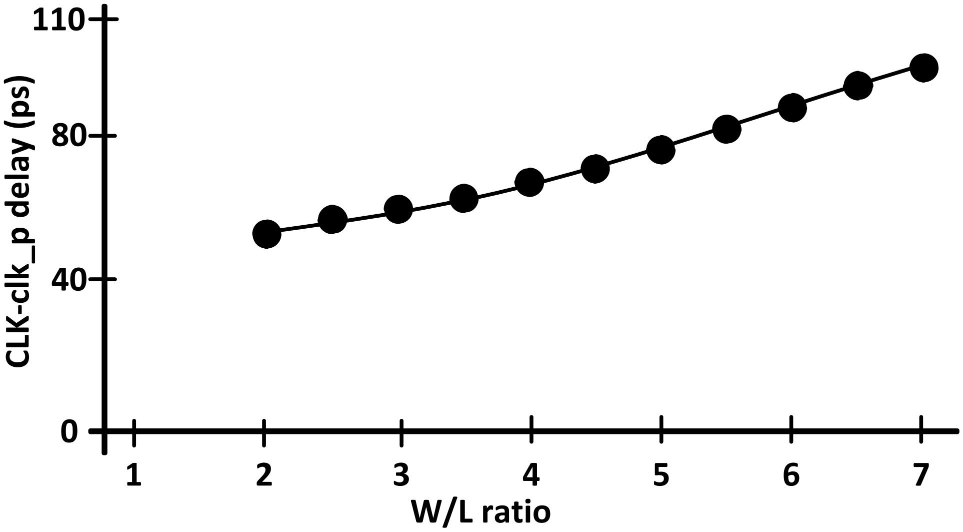


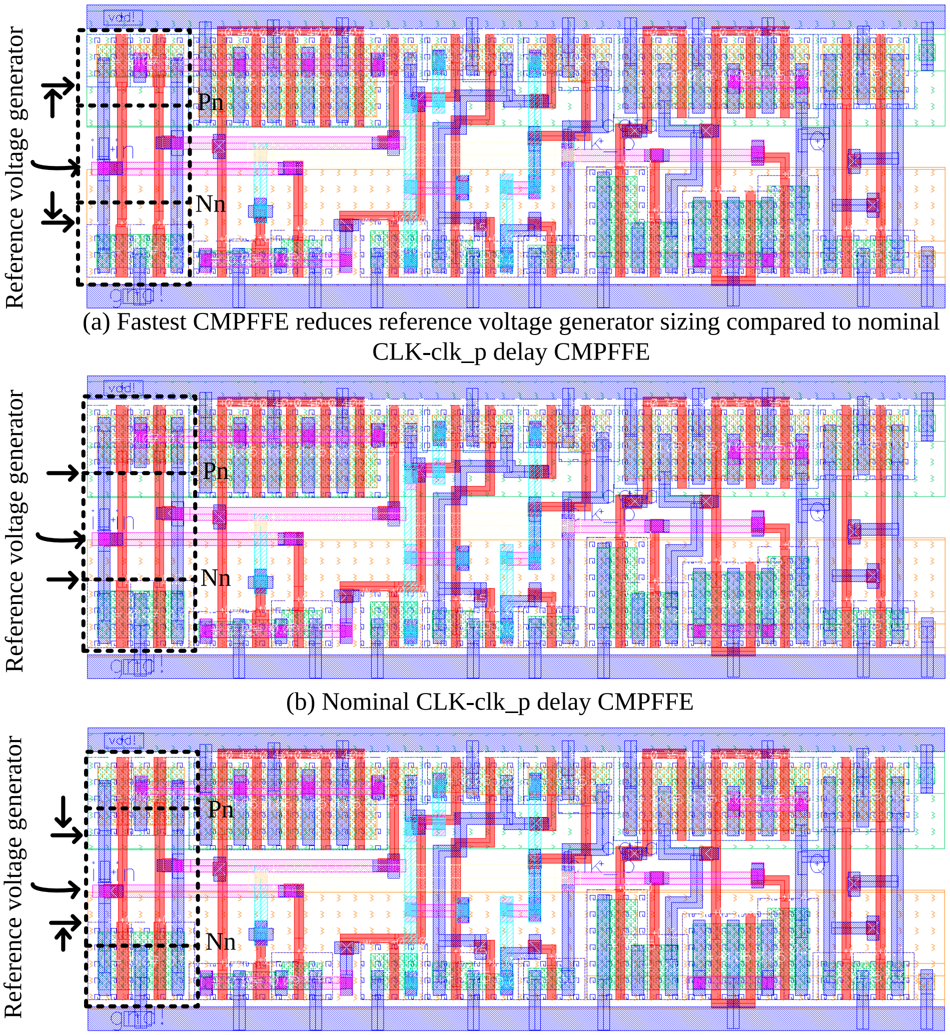
Figure 5.10: CMPFFE library cells are built based on the characteristics that the CM pulsed FF CLK-clk_p delay increases with the increase of aspect-ratio ($\frac{W}{L}$).

CM FF. I laid out the CM FF in such a way that I can adjust the sizing of CMPFFE reference voltage generator without changing the CMPFFE overall area. Since, each FF used standard cell height, I can adjust the AR by using vertical empty space for slower CM FF (larger transistors) or decrease transistors size in the opposite direction (for faster CM FF) as shown in Figure 5.11 (c) and Figure 5.11 (a), respectively. This requires no placement legalization.

5.5.3 Results and Comparisons

I characterized the register stage of each CMPFFE considering maximum driving load. In addition, the clk_p signal has fixed loading from transistors M4, M7, and M10 as shown in Figure 5.7. If the clk_p signal meets a slew rate, there is no slew rate violation at the CMPFFE output (Q) signal.

Table 5.1 shows the power, skew, and run time comparison on the ISPD 2009 benchmarks while Table 5.2 shows the ISPD 2010 benchmark networks. I extracted all the results considering the final Tx and CMPFFE sizes for CM networks.


Reference voltage generator
vdd!
Pn
Nn
gnd!
Reference voltage generator
vdd!
Pn
Nn
gnd!
Reference voltage generator
vdd!
Pn
Nn
gnd!

(a) Fastest CMPFFE reduces reference voltage generator sizing compared to nominal CLK-clk_p delay CMPFFE

(b) Nominal CLK-clk_p delay CMPFFE

(c) Slowest CMPFFE increases reference voltage generator sizing compared to nominal CLK-clk_p delay CMPFFE

Figure 5.11: CMPFFE library cells are built based on the characteristics that the CM pulsed FF CLK-clk_p delay increases with the increase of aspect-ratio ($\frac{W}{L}$).

5.5.3.1 Power Comparison

Table 5.1 and Table 5.2 show the power breakdown of the VM and CM FFs and total CDN simulation power at 1 GHz clock frequency. At 1GHz the CM FFs consume 24% and 20% more average power than VM FFs using ISPD 2009 and ISPD 2010 testbenches, respectively. On the other hand, the VM TxVM consumes 97% and 92% lower average power compared

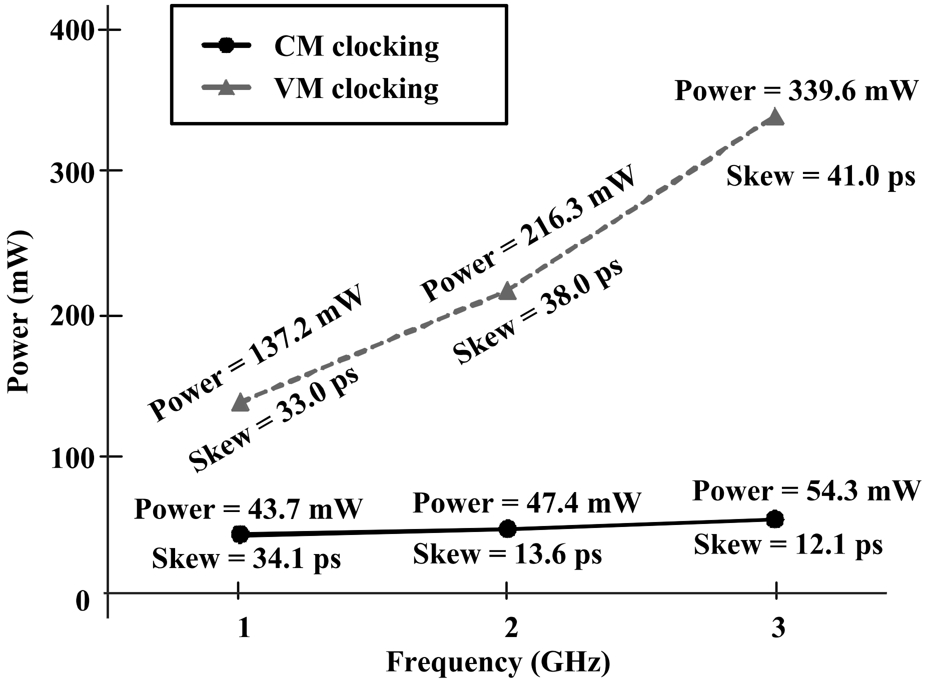


Figure 5.12: The CM clocking is highly insensitive to frequency, as a result it exhibits more power saving at higher frequencies; for example using ISPD 2009 benchmark s4r3 circuit, the power saving of CM methodology increases from 68% (at 1GHz) to 84% (at 3GHz) compared to VM scheme.

to VM buffers power on ISPD 2009 and ISPD 2010 networks, respectively. This is due to the full voltage swing ($0 \rightarrow V_{dd}$) in the VM CDN, whereas the CM CDNVM has negligible voltage swing. Overall, using proposed CMCS methodology, the CM clocking consumes lower power than the traditional buffered VM clocking on all the ISPD 2009 and 2010 benchmarks. Specifically, the CM clocks save 68-90% power compared to the VM buffered networks as shown in Table 5.1 and up to 67% power in Table 5.2.

In a CM scheme, most of the power is static power consumed by the CM FFs and there are no CDN buffers so it is highly insensitive to frequency [36]. Because of this, CM clocking save quadratically more power at higher frequencies which is extremely important in multi-GHz designs. Figure 5.12 shows the evidence of the proposed CMCS methodology efficiency compared with VM buffered scheme at higher frequencies using ISPD 2009 benchmark circuit s4r3. In particular, the power saving of CM methodology increases from 68% (at 1GHz) to 84% (at 3GHz) compared to VM scheme.

5.5.3.2 Skew Comparison

The proposed algorithm reduces skew by Tx and CM FF sizing while ensuring correct functionality. The CMCS methodology resulted in proper functionality in all of the asymmetric networks. The skew slightly degraded on average in both the 2009 and 2010 benchmarks, but the skew results were better on some benchmarks as shown in Table 5.1 and Table 5.2. These skew levels are well within tolerable limits of 5-10% of the clock period and are therefore not a concern especially considering the large power consumption savings. In addition, each scheme uses a different methodology the response to optimization is not predictable. This is common with any sort of heuristic optimization algorithm which may end up in a solution that is closer or further from optimal. However, overall the proposed CM scheme has only $3.3ps$ and $3.9ps$ average skew difference compared to VM scheme for ISPD 2009 and ISPD 2010 testbenches, respectively.

5.5.3.3 Run-Time Comparison

Most high-performance CDNs use HSPICE simulation instead of approximate analytical models such as Elmore delay in traditional clock tree synthesis (CTS) algorithms. However, HSPICE simulation requires significant simulation time compared to a traditional CTS algorithm. Table 5.1 and Table 5.2 show the results based on accurate HSPICE simulation for both VM and CM methodologies for fair comparison of quality of results and run-time.

The run time of the CMCS methodology is significantly less than the VM methodology. This is because, the proposed scheme requires fewer iterations since it doesn't use buffers that need to be sized. Overall, the run time of the benchmarks are $2.4 - 9.1\times$ less on average as shown in Table 5.1 and Table 5.2.

5.5.3.4 Silicon Area Comparison

Similar to previous CM clocking systems, the proposed CMCS scheme uses a bufferless CDN. However, the Tx circuit has a few buffers for the internal delay chain and to drive the large Tx transistors. Figure 5.13 shows a representative comparison of VM buffered total area compared to CM total area. The CM CDN includes the overhead of the resized FFs and Tx to compute the Tx and CM FF area. When considers CM Tx and VM buffers area, the CM clocking saves up to 73% transistor area compared to the VM scheme. Overall, using proposed CMCS methodology in ISPD 2009 and ISPD 2010 benchmarks, the CM clocking saves 21% average silicon area compared to VM scheme as shown in Table 5.3.

Table 5.1: The proposed CMCS scheme enables more than 82% average power saving using CM clocking and $9\times$ lower average run-time(CPU) with only $3.3ps$ skew degradation compared to the traditional VM buffered scheme using 2009 ISPD benchmarks

Benchmark				VM Buffered network			CM network			**CM compared to VM**		
Name	Sink	Chip area	Total cap	Power	Skew	Run time	Power	Skew	Run time	Power	Skew	Run time
	(#)	(mm^2)	(pF)	(mW)	(ps)	(hr)	(mW)	(ps)	(hr)	(%)	(ps)	(VM/CM)
s1r1	81	69.4	27.6	38.5	14.0	0.31	6.0	20.9	0.11	84.5	-6.9	2.8
s2r1	88	54.6	26.3	37.9	20.0	0.50	6.5	21.1	0.12	82.9	-1.1	4.1
s3r1	131	165.6	48.4	69.4	30.0	1.60	9.7	67.8	0.42	86.1	-37.8	3.8
s4r3	623	120.7	79.5	137.2	33.0	4.95	43.7	34.1	0.62	68.2	-1.1	8.0
f11	121	109.2	42.2	60.2	14.0	0.53	9.2	16.7	0.06	84.7	-2.7	9.0
f12	117	91.2	39.5	57.4	20.0	1.43	8.5	28.3	0.14	85.1	-8.3	10.5
f21	117	133.3	43.9	62.0	28.0	0.54	8.6	18.5	0.06	86.1	9.5	9.2
f22	91	50.4	26.1	37.8	12.0	0.27	6.7	19.1	0.12	82.2	-7.1	2.3
f31	273	275.6	89.7	131.7	37.0	3.61	19.8	19.1	0.22	85.0	17.9	16.6
f32	190	269.0	69.5	100.3	23.0	6.03	10.1	18.3	0.31	89.9	4.7	19.6
Avg.	183	133.9	49.3	73.2	23.1	1.98	12.9	26.4	0.22	**82.4**	**-3.3**	**9.1**

Table 5.2: Using more dense clock sinks ISPD 2010 benchmarks the CMCS scheme consumes 39% lower average power and $2.4\times$ lower average run-time(CPU) however, experienced $3.9ps$ skew degradation compared to the VM scheme

Benchmark				VM Buffered network			CM network			**CM compared to VM**		
Name	Sink (#)	Chip area (mm^2)	Total cap (pF)	Power (mW)	Skew (ps)	Run time (hr)	Power (mW)	Skew (ps)	Run time (hr)	Power (%)	Skew (ps)	Run time (VM/CM)
01	1107	64.0	93.7	157.5	32.0	8.0	56.4	42.7	1.58	64.2	-10.7	5.1
02	2249	91.0	180.4	305.8	32.0	11.0	128.3	20.2	4.02	58.0	11.8	2.7
03	1200	1.4	42.5	90.8	33.0	2.5	71.3	23.2	2.03	21.5	9.8	1.2
04	1845	5.7	69.5	128.1	33.0	5.0	126.3	53.2	2.46	1.5	-20.2	2.0
05	1016	5.8	29.6	67.0	26.0	2.0	50.4	25.8	1.38	24.8	0.2	1.4
06	981	1.5	34.9	147.5	22.0	4.0	48.5	37.1	1.32	67.1	-15.1	3.0
07	1915	3.5	60.7	132.8	30.0	4.2	129.0	39.4	2.47	2.9	-9.4	1.7
08	1134	2.6	38.9	92.0	32.0	3.9	75.5	29.9	1.32	18.0	2.1	3.0
Avg.	1431	21.9	68.8	140.2	30.0	5.1	85.7	33.9	2.07	**38.9**	**-3.9**	**2.4**

Table 5.3: The proposed algorithm saves $53 - 73\%$ silicon area as a result of bufferless clock routing using ISPD 2009 and ISPD 2010 benchmarks.

	VM Buffered network				CM network				**Area saving**
Ben.	Buff (#)	Buff. area (um^2)	DDPHFF [1] area (um^2)	Total area (um^2)	Tx Buff. (#)	Tx area (um^2)	CMPFFE area (um^2)	Total area (um^2)	(%)
s1r1	150	86.4	798.7	885.1	7	36.8	644.8	681.5	23.0
s2r1	168	93.3	867.7	960.9	7	34.9	700.5	735.4	23.5
s3r1	355	189.0	1291.7	1480.6	7	55.2	1042.8	1097.9	25.8
s4r3	785	387.5	6142.8	6530.3	14	105.9	4959.1	5065.0	22.4
f11	291	155.6	1193.1	1348.7	7	56.3	963.2	1019.4	24.4
f12	299	158.2	1153.6	1311.8	7	52.6	931.1	983.9	25.0
f21	288	156.1	1153.6	1309.7	7	58.5	931.3	989.8	24.4
f22	171	93.3	897.3	990.5	7	34.9	724.4	759.3	23.3
f31	712	371.4	2691.8	3063.2	21	121.5	2173.1	2292.6	25.2
f32	533	279.6	1873.4	2153.0	14	92.7	1512.4	1605.1	25.5
01	846	409	10915.0	11324.0	28	147.1	8811.7	8958.8	20.9
02	1579	769.5	22175.1	22944.6	42	241.6	17902.0	18143.7	20.9
03	377	173.6	11832.0	12005.6	7	56.6	9552.0	9608.6	20.0
04	427	198.1	18191.7	18389.8	14	92.7	14686.2	14778.9	19.6
05	203	94.5	10017.8	10112.2	7	40.5	8087.4	8127.8	19.6
06	239	111.2	9672.7	9783.9	7	46.3	7808.8	7855.1	19.7
07	438	204.7	18881.9	19086.6	14	80.9	15243.4	15324.3	19.7
08	298	141.2	11181.2	11322.43	7	51.9	9026.6	9078.5	19.8
Avg.	453	226.2	7273.9	7500.2	12	78.0	5872.3	5950.3	20.7

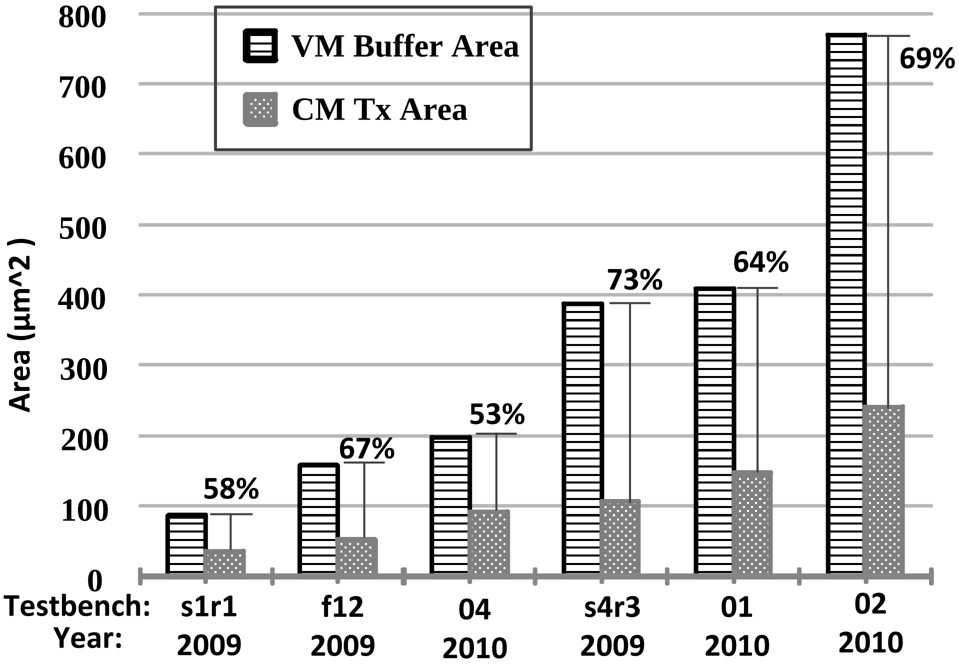


Figure 5.13: The proposed algorithm saves 53% to 73% silicon area as a result of bufferless clock routing using ISPD 2009 and ISPD 2010 benchmarks.

5.6 Summary

I have presented the first current-mode clock synthesis (CMCS) methodology. The proposed methodology used Tx and Rx sizing in the CM FFs to ensure correct functionality and reduce skew. The proposed methodology saved $39 - 82\%$ average power with similar skews on industrial benchmarks. In addition, the methodology used $2.4 - 9.1\times$ less run-time up to 73% lower silicon area compared to the buffered VM networks.

Chapter 6

Conclusions and Future Work

In modern synchronous VLSI, interconnect, in particular CDN design is growing in importance as it is significantly affecting the power-performance trade-offs. The traditional CDN design approaches are solely based on VM signaling due to the compatibility with the logic blocks and existing automated routing tools. However, using VM clocks requires charging/discharging of the large global CDN capacitance which consumes significant power. While CM clocks are a promising technique to reduce the total power consumption by sending information through current at a nearly constant voltage swing on the interconnect, CM applications were limited to off-chip signal transmission. However, according to my best exploration, there are few clocking and no design automation techniques that have considered CM clocking. The traditional CM clocking/signaling schemes are small hand tuned designs and only restricted to simplified H-tree or regular CDNs. This thesis is the first systematic research on CM clocking and design automation including the verification of proposed scheme on industrial testbenches.

6.1 Thesis Contributions

In order to propose a new paradigm of clock distribution, in this thesis, I made several contributions that advance the CM clocking methodology in the VLSI design, with the ultimate goal is to meet the power budget at the target frequency range. The key contributions of this thesis are:

Current-Mode Pulsed Flip-flop and Current-Mode Clock Distribution In Chapter 3, I present the first CM clocked FF and the effective integration of the CM FF with VM

CMOS logic.

Differential Current-Mode Pulsed Flip-flop and Differential Current-Mode Clock Distribution In Chapter 4, I demonstrate the first differential current-mode clocked FF. and the effective integration of the DCM FF with VM CMOS logic.

CMCS: Current-Mode Clock Synthesis In Chapter 5, I present the first and only clock synthesis methodology to handle CM clock signaling. This Chapter also demonstrates the first CM clocking in asymmetric clock networks using industrial benchmarks. In addition, I present the first CM latch/FF sizing to minimize global skew.

6.2 Future Work

In this thesis, I presented a complete CM clocking scheme and CM clock synthesis methodology. However, the experimental results are based on multi-level symmetric H-tree distribution and ISPD 2009-ISPD 2010 industrial testbenches using only HSPICE simulation and analytical modeling. A possible direction of research would implement a microprocessor and applying both single-ended and differential CM clocking to investigate more accurate results.

The proposed CMCS methodology utilized DME algorithm to route the clock tree, however, it would be interesting to consider CM clocking using other heuristic algorithms (MMM, GMA). In addition, none of these algorithm ensures zero skews for CM clocking, hence a possible direction of research is to implement a new algorithm that ensures zero skews to CM clocking scheme. The proposed CMCS methodology is based on time-dependent HSPICE simulation. A time-independent equal impedance CDN for CM clocking using wire/Tx/FF sizing is identified as a possible direction of research.

The reliability of CMCS methodology can be improved by considering noise related issues. For example, the effect of process variation on the CM CDN circuitry, wire parameters (RC) values, supply-voltage induced noise, clock jitter could introduce clock skew. A possible direction of research is to model all those noise attributes and integrate into CMCS to have more accurate results.

It is common practice to reduce the supply voltage to near the device threshold voltage in many-core designs. However, the circuit robustness often reduces due to the random parameter variation and supply voltage scaling. As a result, it becomes increasingly difficult to build energy efficient SOCs. However, asynchronous VLSI techniques using quasi-delay-insensitive (QDI) asynchronous logic and chip-to-chip/local handshaking protocols identified as very promising for building low-power microprocessor. Asynchronous VLSI design is identified as a possible direction of research.

Bibliography

[1] K. Absel, L. Manuel, and R.K. Kavitha. Low-power dual dynamic node pulsed hybrid flip-flop featuring efficient embedded logic. *TVLSI*, 21(9):1693–1704, Sept 2013.

[2] C.J. Anderson, J. Petrovick, J.M. Keaty, J. Warnock, G. Nussbaum, J.M. Tendier, C. Carter, S. Chu, J. Clabes, J. DiLullo, P. Dudley, P. Harvey, B. Krauter, J. LeBlanc, Pong-Fei Lu, B. McCredie, G. Plum, P.J. Restle, S. Runyon, M. Scheuermann, S. Schmidt, J. Wagoner, R. Weiss, S. Weitzel, and B. Zoric. Physical design of a fourth-generation power ghz microprocessor. *ISSCC*, pages 232 – 233, Feb 2001.

[3] Semiconductor Industry Association. The international technology roadmap for semiconductor. 2012 edition.

[4] W. Athas, N. Tzartzanis, W. Mao, R. Lai, K. Chong, L. Peterson, and M. Bolotski. Clock-powered cmos vlsi graphics processor for embedded display controller application. *ISSCC*, pages 296–297, Feb 2000.

[5] Y. Bai, Y. Song, M. N. Bojnordi, A. Shapiro, E. G. Friedman, and E. Ipek. Back to the future: Current-mode processor in the era of deeply scaled CMOS. *TVLSI*, 24(4):1266–1279, April 2016.

[6] D.W. Bailey and B.J. Benschneider. Clocking design and analysis for a 600-mhz alpha microprocessor. *JSSC*, 33(11):1627–1633, Nov 1998.

[7] H. Bakoglu, J. T. Walker, and J. D. Meindl. A symmetric clock distribution tree and optimized high speed interconnections for reduced clock skew in ULSI and WSI circuit. In *ICCD*, pages 118–122, Oct 1986.

[8] I. Bezzam, S. Krishnan, T. Raja, and C. Mathiazhagan. Low power low voltage wide frequency resonant clock and data circuits for power reductions. In *LASCAS*, pages 1–4, Feb 2013.

[9] I. Bezzam, C. Mathiazhagan, T. Raja, and S. Krishnan. An energy-recovering reconfigurable series resonant clocking scheme for wide frequency operation. *TCASI*, 62(7):1766–1775, July 2015.

[10] K.D. Boese and A.B. Kahng. Zero-skew clock routing trees with minimum wirelength. In *ASIC*, pages 17–21, Sep 1992.

[11] G. S. Brodal and A. G. Jørgensen. A linear time algorithm for the k maximal sums problem. *Mathematical Foundations of Computer Science*, pages 578–586, Aug 2007.

[12] S.C. Chan, P.J. Restle, T.J. Bucelot, J.S. Liberty, S. Weitzel, J.M. Keaty, B. Flachs, R. Volant, Peter Kapusta, and J.S. Zimmerman. A resonant global clock distribution for the cell broadband engine processor. *JSSC*, 44(1):64 – 72, Jan 2009.

[13] S.C. Chan, Kenneth L. Shepard, and P.J. Restle. Distributed differential oscillators for global clock networks. *JSSC*, 41(9):2083–2094, Sept 2006.

[14] S.C. Chan, K.L. Shepard, and P.J. Restle. Uniform-phase uniform-amplitude resonant-load global clock distributions. *JSSC*, 40(1):102 – 109, Jan 2005.

[15] T.-H. Chao, Yu-Chin Hsu, and Jan-Ming Ho. Zero skew clock net routing. In *DAC*, pages 518–523, Jun 1992.

[16] S. Chen, H. Li, and P. Y. Chiang. A robust energy/area-efficient forwarded-clock receiver with all-digital clock and data recovery in 28-nm CMOS for high-density interconnects. *TVLSI*, 24(2):578–586, Feb 2016.

[17] V.H. Cordero and S.P. Khatri. Clock distribution scheme using coplanar transmission lines. In *DATE*, pages 985–990, 2008.

[18] M. Dave, M. Jain, S. Baghini, and D. Sharma. A variation tolerant current-mode signaling scheme for on-chip interconnects. *TVLSI*, PP(99):1 – 12, Jan 2012.

[19] J.P. de Gyvez and R. Rodriguez-Montanes. Threshold voltage mismatch (ΔVT) fault modeling. In *VLSITS*, pages 145–150, April 2003.

[20] M.A. El-Moursy and E.G. Friedman. Exponentially tapered h-tree clock distribution networks. *TVLSI*, 13(8):971–975, Aug 2005.

[21] W. C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *JAP*, 19(1):55 – 63, Jan 1948.

[22] E.J. Fluhr, J. Friedrich, D. Dreps, V. Zyuban, G. Still, C. Gonzalez, A. Hall, D. Hogenmiller, F. Malgioglio, R. Nett, J. Paredes, J. Pille, D. Plass, R. Puri, P. Restle, D. Shan, K. Stawiasz, Z.T. Deniz, D. Wendel, and M. Ziegler. 5.1 power8tm: A 12-core server-class processor in 22nm soi with 7.6tb/s off-chip bandwidth. In *ISSCC*, pages 96–97, Feb 2014.

[23] E.G. Friedman. Clock distribution networks in synchronous digital integrated circuits. *IEEE*, 89(5):665–692, May 2001.

[24] H. Fuketa, M. Nomura, M. Takamiya, and T. Sakurai. Intermittent resonant clocking enabling power reduction at any clock frequency for near/sub-threshold logic circuits. *JSSC*, 49(2):536–544, Feb 2014.

[25] J.C. Garcia, J.A. Montiel-Nelson, and S. Nooshabadi. Adaptive low/high voltage swing cmos driver for on-chip interconnects. In *ISCAS*, pages 881–884, May 2007.

[26] Matthew R. Guthaus, Gustavo Wilke, and Ricardo Reis. Revisiting automated physical synthesis of high-performance clock networks. *TODAES*, 18(2):31:1–31:27, April 2013.

[27] M.R. Guthaus, N. Venkateswarant, C. Visweswariah, and V. Zolotov. Gate sizing using incremental parameterized statistical timing analysis. In *ICCAD*, pages 1029–1036, Nov 2005.

[28] H. Zhang and V. George and J. M. Rabaey. Low swing on-chip signaling techniques: effectiveness and robustness. *TVLSI*, 8(3):264 – 272, Jun 2000.

[29] Xuchu Hu and M.R. Guthaus. Distributed lc resonant clock grid synthesis. *TCASI*, 59(11):2749–2760, Nov 2012.

[30] Yin-Tsung Hwang, Jin-Fa Lin, and Ming hwa Sheu. Low-power pulse-triggered flip-flop design with conditional pulse-enhancement scheme. *TVLSI*, 20(2):361–366, Feb 2012.

[31] Nanoscale Integration and Modeling (NIMO) Group at ASU. Predictive technology model (PTM). `http://ptm.asu.edu/`.

[32] R. Islam. High-speed energy-efficient soft error tolerant flip-flops. *M.A.Sc.Thesis*, 2011.

[33] R. Islam, H. Fahmy, Ping-Yao Lin, and M.R. Guthaus. Differential current-mode clock distribution. In *MWSCAS*, pages 1–4, Aug 2015.

[34] R. Islam and M. R. Guthaus. CMCS: Current-mode clock synthesis. *TVLSI*, 25(3):1054–1062, March 2017.

[35] R. Islam and M.R. Guthaus. Current-mode clock distribution. In *ISCAS*, pages 1203–1206, June 2014.

[36] R. Islam and M.R. Guthaus. Low-power clock distribution using a current-pulsed clocked flip-flop. *TCASI*, 62(4):1156–1164, Apr 2015.

[37] ISSCC. ISSCC 2013 supplement. `http://isscc.org/index.html`.

[38] M.A.B. Jackson, A. Srinivasan, and E.S. Kuh. Clock routing for high-performance ICs. In *DAC*, pages 573–579, Jun 1990.

[39] A.P Jose, G. Patounakis, and K.L. Shepard. Near speed-of-light on-chip interconnects using pulsed current-mode signalling. In *VLSIC*, pages 108–111, June 2005.

[40] A. Kahng, J. Cong, and G. Robins. High performance clock routing based on recursive geometric matching. In *DAC*, pages 322–327, Jun 1991.

[41] A.B. Kahng, Seokhyeong Kang, and Hyein Lee. Smart non-default routing for clock power reduction. In *DAC*, pages 1–7, May 2013.

[42] N. K. Kancharapu et al. A low-power low-skew current-mode clock distribution network in 90nm cmos technology. In *ISVLSI*, pages 132–137, Jul 2011.

[43] A. Katoch, H. Veendrick, and E. Seevinck. High speed current-mode signaling circuits for on-chip interconnects. In *IEEE ISCAS*, pages 4138 – 4141, May 2005.

[44] S. Kozu et al. A 100 mhz, 0.4 w risc processor with 200 mhz multiply adder, using pulse-register technique. In *ISSCC*, pages 140–141, 1996.

[45] E. A. Kusse. Analysis and circuit design for low power programmable logic modules. *M.S.Thesis*, 1997.

[46] A. Maheshwari and W. Burleson. Differential current-sensing for on-chip interconnects. *TVLSI*, 12(12):1321–1329, Dec 2004.

[47] A. Maheshwari and W. Burleson. Current-sensing and repeater hybrid circuit technique for on-chip interconnects. *TVLSI*, 15(11):1239–1244, Nov 2007.

[48] H. M. Mahmoodi, M. Cooke, and K. Roy. Ultra low-power clocking scheme using energy recovery and clock gating. *TVLSI*, 17(1):33 – 44, Jan 2009.

[49] A.D. Mehta, Yao-Ping Chen, N. Menezes, D.F. Wong, and L.T. Pileggi. Clustering and load balancing for buffered clock tree synthesis. In *ICCD*, pages 217–223, Oct 1997.

[50] G. C. Messenger. Collection of charge on junction nodes from ion tracks. *TNS*, 29(6):2024 – 2031, Dec 1982.

[51] Gordon E. Moore. Cramming more components onto integrated circuits. *SSCSN*, 38(8):114, April 1965.

[52] Y. Nakagome, B.S. Kiyoo Itoh, M. Isoda, K. Takeuchi, and M. Aoki. Sub-1-v swing internal bus architecture for future low-power ulsis. *JSSC*, 28(4):414–419, Apr 1993.

[53] A. Narasimhan, S. Divekar, P. Elakkumanan, and R. Sridhar. A low-power current-mode clock distribution scheme for multi-ghz noc-based socs. In *ICVD*, pages 130–135, Jan 2005.

[54] A. Narasimhan, M. Kasotiya, and R. Sridhar. A low-swing differential signalling scheme for on-chip global interconnects. In *ICVD*, pages 634–639, Jan 2005.

[55] NCSU. FreePDK45. `http://www.eda.ncsu.edu/wiki/FreePDK45`.

[56] N. Nedovic, M. Aleksic, and V.G. Oklobdzija. Conditional techniques for low power consumption flip-flops. In *ICECS*, volume 2, pages 803–806, 2001.

[57] S. Pullela, N. Menezes, and L.T. Pillage. Low power ic clock tree design. In *CICC*, pages 263–266, May 1995.

[58] J Rabaey. Low power design essentials. second edition. *Springer Science and Business Media*, Jan 2009.

[59] J Rabaey, A Chandrakasan, and B Nikolic. Digital integrated circuits: A design prospective. second edition. *Prentice Hall*, Jan 2003.

[60] J. Rosenfeld and E.G. Friedman. Design methodology for global resonant h-tree clock distribution networks. In *ISCAS*, pages 2073–2076, May 2006.

[61] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *IEEE*, 91(2):305–327, Feb 2003.

[62] K. C. Saraswat and F. Mohammadi. Effect of scaling of interconnections on the time delay of vlsi circuits. *JSSC*, Sc-17(2):275 – 280, Apr 1982.

[63] M. Sasaki. A high-frequency clock distribution network using inductively loaded standing-wave oscillators. *JSSC*, 44(10):2800–2807, 2009.

[64] V. S. Sathe, J. C. Kao, and M. C. Papaefthymiou. Resonant-clock latch-based design. *JSSC*, 43(4):864 – 873, Apr 2008.

[65] V.S. Sathe, S. Arekapudi, A. Ishii, C. Ouyang, M.C. Papaefthymiou, and S. Naffziger. Resonant-clock design for a power-efficient, high-volume x86-64 microprocessor. *JSSC*, 48(1):140 – 149, Jan 2013.

[66] Evert Seevinck, P. J. V. Beers, and H. Ontrop. Current-mode techniques for high-speed vlsi circuits with application to current sense amplifier for cmos srams. *JSSC*, 26(4):525 – 536, Apr 1991.

[67] D.C. Sekar. Clock trees: differential or single ended? In *ISQED*, pages 548–553, March 2005.

[68] Xin-Wei Shih and Yao-Wen Chang. Fast timing-model independent buffered clock-tree synthesis. In *DAC*, pages 80–85, June 2010.

[69] D. Sylvester and C. Hu. Analytical modeling and characterization of deep-submicrometer interconnect. *IEEE*, 89(5):634 – 664, May 2001.

[70] C N Sze. ISPD 2010 High Performance Clock Network Synthesis Contest. In *ISPD*, Mar 2010.

[71] C N Sze, P Restle, G J Nam, and C J Alpert. Clocking and the ISPD'09 clock synthesis contest. In *ISPD*, pages 149 – 150, Mar 2009.

[72] Simon Tam, S. Rusu, U. Nagarji Desai, R. Kim, Ji Zhang, and Ian Young. Clock generation and distribution for the first ia-64 microprocessor. *JSSC*, 35(11):1545–1552, Nov 2000.

[73] G.E. Tellez and M. Sarrafzadeh. Minimal buffer insertion in clock trees with skew and slew rate constraints. *TCAD*, 16(4):333–342, Apr 1997.

[74] R.-S. Tsay. Exact zero skew. In *ICCAD*, pages 336–339, Nov 1991.

[75] N. Tzartzanis and W.W. Walker. Differential current-mode sensing for efficient on-chip global signaling. *JSSC*, 40(11):2141–2147, Nov 2005.

[76] N H E Weste and D M Harris. Cmos vlsi design: A circuits and systems perspective. third edition. *Pearson Addision-Wesley*, Jan 2004.

[77] J. Wood, T.C. Edwards, and S. Lipa. Rotary traveling-wave oscillator arrays: a new clock technology. *JSSC*, 36(11):1654–1665, 2001.

[78] M. Yamashina and H. Yamada. An MOS current mode logic (MCML) circuit for low-power sub-GHz processors. *IEICE Transactions on Electronics*, E75-C(10):1181–1187, 1992.

[79] Y. Ye, S. Borkar, and V. De. A new technique for standby leakage reduction in high-performance circuits. In *VLSIC*, pages 40–41, June 1998.

[80] Fei Yuan. CMOS current-mode circuits for data communications. *Springer*, Apr 2007.

[81] J. Yuan and C. Svensson. High-speed cmos circuit technique. *JSSC*, 24(1):62–70, 1989.

[82] J. L. Zerbe, P. S. Chau, C. W. Werner, T. P. Thrush, H. J. Liaw, B. W. Garlepp, and K. S. Donnelly. 1.6 gb/s/pin 4-pam signaling and circuits for a multidrop bus. *JSSC*, 36(5):752–760, May 2001.

[83] H. Zhang and J. Rabaey. Low-swing interconnect interface circuits. In *ISLPED*, pages 161–166, Aug 1998.

[84] L. Zhang, J.M. Wilson, R. Bashirullah, L. Lei, J. Xu, and P.D. Franzon. Voltage-mode driver preemphasis technique for on-chip global buses. *TVLSI*, 15(2):231–236, Feb 2007.

Index