

# An Ultra-Low-Power 75mV 64-Bit Current-Mode Majority-Function Adder

by

Manuchehr Ebrahimi

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada 2012

© Manuchehr Ebrahimi 2012

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

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

Manuchehr Ebrahimi

## Abstract

Ultra-low-power circuits are becoming more desirable due to growing portable device markets and they are also becoming more interesting and applicable today in biomedical, pharmacy and sensor networking applications because of the nano-metric scaling and CMOS reliability improvements. In this thesis, three main achievements are presented in ultra-low-power adders. First, a new majority function algorithm for carry and the sum generation is presented. Then with this algorithm and implied new architecture, we achieved a circuit with 75mV supply voltage operation. Last but not least, a 64 bit current-mode majority-function adder based on the new architecture and algorithm is successfully tested at 75mV supply voltage. The circuit consumed 4.5nW or 3.8pJ in one of the worst conditions.

## Acknowledgements

First I would like to have my special thanks to my supervisor, Professor Manoj Sachdev and express my deepest gratitude to him for his constant help and support throughout my research. Without his advice and encouragement this work would not have been possible. It was honor and privilege to be his student. Moreover, I would like to thank my committee members for their comments and inputs that helped adding the value of my thesis. Also I want to thank Phil Regier for his constant help and support with tools, equipments and technology access.

I would like to dedicate my final words to my father, mother and sister. Without your words, support and encouragement I wouldn't be able to have any word in here. To you with my best appreciation!

# Contents

List of Tables	ix
List of Figures	x
List of Acronyms	xii
<b>Chapter 1</b>	<b>1</b>
<b>Introduction</b>	<b>1</b>
1.1. Motivation	1
1.2. Thesis Organization	2
<b>Chapter 2</b>	<b>3</b>
<b>Power Consumption in CMOS Circuits</b>	<b>3</b>
2.1. Introduction	3
2.2. Power Dissipation	4
2.2.1 Static power	5
2.2.2. Dynamic Power	8
2.2.3. Short Circuit	10
2.3. Low Power and Low Energy Circuits Ideas	11
2.3.1. Low Voltage and Sub-Threshold Circuits	12

2.3.2. Pipelined and Self-Timed Circuits	16
2.3.3. Adiabatic-Switching	20
2.3.4. Winner-Take All circuits	22
2.4. Summary	24
<b>Chapter 3</b>	<b>25</b>
<b>Adder Architectures</b>	<b>25</b>
3.1. Introduction	25
3.2. Boolean Logic Full Adder Function	26
3.3. Boolean Logic Full Adder Architectures	26
3.3.1 Ripple Carry Adder	27
3.3.2. Carry Skip Adder	29
3.3.3. Carry Select Adder	29
3.3.4. Carry Save Adder	29
3.3.5. Brent-Kung Adder	30
3.3.6. Kogge-Stone Adder	31
3.3.7. Han-Carlson Adder	31
3.3.8. Lander-Fischer Adder	31
3.3.9. Parallel Adder Taxonomy Revisited	33
3.4. Low Power Full Adder Architectures Comparison	33
3.5. Majority Function	36
3.6. Majority Function Architectures	36

3.7. Summary	39
<b>Chapter 4</b>	<b>40</b>
<b>Ultra low power Current-Mode Majority Function Full-Adder</b>	<b>40</b>
4.1. Introduction	40
4.2. Project overview and guidance	40
4.3. Weak inversion transistor sizing and characteristics	42
4.4. Current-Mode Majority Function FA implementation	49
4.4.1 Carry circuit	53
4.4.2 Sum circuit	55
4.4.3 Pulse Generating	56
4.5. Self-Timed Circuit implementing	60
4.6. Circuit Simulation and Analysis	61
4.6.1. A Single Bit CMMF-FA Simulation	62
4.7. A 64-Bit Pipeline CMMF-FA Test and Simulation	67
4.8. Conventional Full Adder Test with 75mV	77
<b>Chapter 5</b>	<b>79</b>
<b>Conclusion</b>	<b>79</b>
5.1. Project Review	79

5.2. Future works

81

## References

82



# List of Tables

Table 2.2.1 Technology Scaling Trends	<a href="#">13</a>
Table 3.4.1 Adder Architectures	<a href="#">40</a>
Table 3.4.2 Adder Architectures characteristics at 1.2V, 27°C.	<a href="#">41</a>
Table 4.4.1 Carry generation in Majority Function full adder.	<a href="#">55</a>
Table 4.4.2 Sum generation in Majority Function full adder.	<a href="#">55</a>
Table 4.4.3 Simplified Sum generation in Majority Function full adder.	<a href="#">56</a>

# List of Figures

Figure (2.2.1) Conventional CMOS circuit.	<a href="#">4</a>
Figure (2.2.2) Leakage Current Components.	<a href="#">7</a>
Figure (2.2.3) Short circuit model.	<a href="#">11</a>
Figure (2.3.1) Transistor current characteristics.	<a href="#">14</a>
Figure (2.3.2) Gated Clocks basics.	<a href="#">17</a>
Figure (2.3.3) TSPC pipeline.	<a href="#">18</a>
Figure (2.3.4) Most common pipeline architectures.	<a href="#">18</a>
Figure (2.3.5) Comparison of (a) Synchronous and (b) Asynchronous circuit structure.	<a href="#">19</a>
Figure (2.3.6) SCRL NAND gate.	<a href="#">21</a>
Figure (2.3.7) 2LAL Buffer.	<a href="#">21</a>
Figure (2.3.8) Standard Winner-Take-All Network	<a href="#">22</a>
Figure (2.3.9) Two Channel Winner-Take-All Network.	<a href="#">23</a>
Figure (3.3.1) 4-bit Ripple Carry Adder.	<a href="#">28</a>
Figure (3.3.2) 4-bit Carry Skip Adder.	<a href="#">28</a>
Figure (3.3.3) 4-bit Carry Select Adder.	<a href="#">28</a>
Figure (3.3.4) 4-operands Carry Save Adder.	<a href="#">30</a>
Figure (3.3.5) 16-bit Brent-Kung Adder.	<a href="#">32</a>
Figure (3.3.6) 16-bit Kogge-Stone Adder.	<a href="#">32</a>
Figure (3.3.7) 16-bit Han-Carlson Adder.	<a href="#">32</a>
Figure (3.3.8) 16-bit Lander-Fischer Adder.	<a href="#">32</a>
Figure (3.3.9) Parallel Adder Taxonomy Revisited.	<a href="#">39</a>
Figure (3.4.1) 32-bit Low-Power Adder comparison.	<a href="#">35</a>
Figure (3.5.1) Majority Function in Voltage Mode.	<a href="#">37</a>
Figure (3.5.2) Majority Function Full Adder implementing using VMMF[6].	<a href="#">37</a>
Figure (3.5.3) Average Power Comparison.	<a href="#">38</a>
Figure (3.5.4) PDP Curves comparison.	<a href="#">38</a>
Figure(4.3.1) The “L” size effect on the $I_D$ in voltage variation when “W” is Min.	<a href="#">43</a>
Figure(4.3.2) The “L” size effect on leakage in voltage variation when “W” is Min.	<a href="#">43</a>
Figure(4.3.3) The “W” size effect on the $I_D$ in voltage variation when “L” is Min.	<a href="#">44</a>
Figure(4.3.4) The “W” size effect on leakage in voltage variation when “L” is Min.	<a href="#">44</a>
Figure(4.3.5) Total width of transistors comparison.	<a href="#">45</a>
Figure(4.3.6) Effect of “F” parameter on Transistors current.	<a href="#">46</a>
Figure(4.3.7) Effect of “F” parameter on Transistors leakage.	<a href="#">46</a>
Figure(4.3.8) Effect of greater “F” parameter on Transistors current.	<a href="#">47</a>
Figure(4.3.9) VTC vs. PMOS sizes at 75mV.	<a href="#">48</a>

Figure(4.3.10) VTC vs. PMOS sizes at 200mV.	<a href="#">48</a>
Figure(4.4.1) Current mode majority function basic.	<a href="#">52</a>
Figure(4.4.2) Current mode majority function FA concept.	<a href="#">53</a>
Figure(4.4.3) The Carry circuit in current mode majority function FA.	<a href="#">54</a>
Figure(4.4.4) The Sum circuit in current mode majority function FA.	<a href="#">55</a>
Figure(4.4.5) The PCi circuit in current mode majority function FA.	<a href="#">56</a>
Figure(4.4.6) The AND gate which is been used in the PCi circuit.	<a href="#">57</a>
Figure(4.4.7) The PSi circuit in current mode majority function FA.	<a href="#">58</a>
Figure(4.4.8) The AND gate which is been used in the PCi circuit.	<a href="#">59</a>
Figure(4.5.1) Asynchronous pipeline pulse generating in CMMF.	<a href="#">60</a>
Figure(4.6.1) Single Bit CMMF Full Adder Test Circuit.	<a href="#">62</a>
Figure(4.6.2) A Single Bit CMMF Full Adder Circuit.	<a href="#">63</a>
Figure(4.6.3) CMMF-Driver circuit (Left) and MF-INV1 Circuit (Right).	<a href="#">63</a>
Figure(4.6.4) A Single Bit CMMF-FA Simulation at 200mV/27°C.	<a href="#">64</a>
Figure(4.6.5) A Single Bit CMMF-FA Simulation at 75mV/27°C.	<a href="#">64</a>
Figure(4.6.6) A Single Bit CMMF-FA Simulation at 200mV/57°C.	<a href="#">65</a>
Figure(4.6.7) A Single Bit CMMF-FA Simulation at 75mV/40°C.	<a href="#">65</a>
Figure(4.7.1) A 64 Bits CMMF-FA circuit.	<a href="#">67</a>
Figure(4.7.2) A 64 Bits CMMF-Adder Test circuit.	<a href="#">68</a>
Figure(4.7.3) CMMF-BUF64 and CMMF-Load circuits.	<a href="#">70</a>
Figure(4.7.4) Test Results of Bits 1 to 12.	<a href="#">71</a>
Figure(4.7.5) Test Results of Bits 13 to 24.	<a href="#">72</a>
Figure(4.7.6) Test Results of Bits 25 to 36.	<a href="#">73</a>
Figure(4.7.7) Test Results of Bits 37 to 48.	<a href="#">74</a>
Figure(4.7.8) Test Results of Bits 49 to 60.	<a href="#">75</a>
Figure(4.7.9) Test Results of Bit 61 to 64.	<a href="#">76</a>
Figure(4.7.10) Current consumption in 64 Bit adding operation.	<a href="#">76</a>
Figure(4.8.1) A Single Bit Conventional Full Adder based on TGL.	<a href="#">77</a>
Figure(4.8.2) A Single Bit Conventional Full Adder Test Results at 75mV.	<a href="#">78</a>
Figure(4.8.3) A 5 Bits Conventional Adder Test Results at 75mV.	<a href="#">78</a>

# List of Acronyms

PDP	Power-Delay Product
DIBL	Drain-Induced Barrier
GIDL	Gate-Induced Drain Leakage
TSPC	True Single Phase Clock
CDPD	Clock and Data Precharged Circuit Dynamic
DFF	D Flip-Flop
PVT	Process, Voltage and Temperature variation
SCRL	Split-Level Charge Recovery Logic
2LAL	Two Level Adiabatic Logic
CP	Carry Propagate
CLA	Carry Look-Ahead
RCA	Ripple Carry Adder
CSK	Carry-Skip Adder
CSA	Carry-Select Adder
KS	Kogge-Stone Adder
MF	Majority-Function
VMMF	Voltage-Mode Majority-Function
CMMF	Current-Mode Majority-Function
FA	Full Adder

# Chapter 1

## Introduction

### 1.1. Motivation

Power consumption is a key limitation in many electronic systems, ranging from mobile telecom to portable and desktop computing systems. Power is also a show stopper for many emerging applications like ambient intelligence and sensor networks. Consequently, new design techniques and methodologies are needed to control and limit power consumption. From sophisticated handheld devices to bioelectronic circuits and nano-satellites, all require low power design. Due to scaling, circuits are becoming more capable, use more transistors to implement complicated functions and offer new applications to customers. But this means more power consumption. In some cases, low power design is required to avoid over heating. There are other applications like bioelectronics where the circuit would be implanted inside the body and has to work either with small battery or using power harvesting techniques. Similar to that, RFID and growing sensor networking circuits also have to consume very low power because of available power limitation. In some cases we may consider low-power design a second priority, but in those applications lower-power design is critical. So either source power limitation or, over heating concern and battery life consideration, low power design is the answer.

In digital processing, a full adder is one of the main elements; an ALU, DSP and digital filtering in any microprocessor/microcontroller are based on it. Therefore, to have low power digital processing, a low-power full adder is desired.

In terms of power reduction techniques and comparison there are few papers and references available. At the architecture level, some solutions like adiabatic circuits have been introduced to reduce power consumption. However, some of these solutions, like adiabatic, may not be practical due to the number of transistors they require. Some of these techniques like pipeline structures or asynchronous timing becoming more attractive and getting more attention than other solutions. This is beside the original and main solution to reduce the supply voltage.

The aim of this research is to explore different solutions along with circuit techniques and to achieve a practical low-power architecture that is applicable and suitable for 64-bit low-power addition.

## 1.2. Thesis Organization

In chapter 2, we review power consumption in CMOS circuits, which is followed by solutions that are introduced to lower power consumption. First, we quickly review the CMOS sources and design consideration, theory to implementation, to have low power circuits.

Chapter 3 provides background information on existing adder architectures. It compares some of the architectures in terms of power consumption and introduces suitable low power architecture.

In chapter 4, we recall results of chapter 2 and 3 and propose a new architecture. Chapter 5, is the conclusion and summary of achievements followed by future works.

# Chapter 2

## Power Consumption in CMOS Circuits

### 2.1. Introduction

Low-power circuit operation is becoming an increasingly important metric for future integrated circuits. As technology continues to scale into the sub-micron regime, massively parallel architectures are increased and being constrained by power considerations. Low power and low energy have captivated circuit designers for the past few years in the quest for enhancing performance and extending battery lifetime. The increasing demand for integrating more functions with faster speeds is met by a slow increase in the capacity of batteries. The increasing power dissipation for fixed supply devices is almost equally challenging as for portable devices. As technology feature size is reduced, the number of transistors on the chip is increased and more power is dissipated. According to Moore's law, the number of transistors quadruples every two to three years. Expensive packing techniques are essential for dissipating such extensive power consumption from that large number of transistors. Also, increased power dissipation has an impact on device reliability. The terms of low power and low energy, although have different definitions, both serve to achieve the same objective. Power is defined as the average product of supplied voltage to a chip from the power supply and its consumed current and it is measured in watts. Meanwhile, the term of energy refers to the energy dissipated per operation and is measured in joules. In fact, energy can be expressed in terms of the Power-Delay Product (PDP), which is the product of power consumption and delay. In general, reducing power will increase delay time; however performance is a product of these two parameters. There are some methods

and techniques for power and energy reduction. Most of the techniques in low power design are not really new ideas or concepts but mainly they are revisited due to transistors scaling which is a source of leakage currents.

## 2.2. Power Dissipation

In most digital CMOS integrated circuits, power consumption can be attributed to three different components: short circuit, leakage, and dynamic switching power. Short circuit currents occur in CMOS circuits during switching transients when both NMOS and PMOS devices are “on” but usually are small in well designed circuits. Dynamic switching power is the dominant component of power consumption today and it is result of the gate and interconnect capacitances charging and discharging during the switching of signals. The third component of power consumption is the leakage which is also considered as static power dissipation.

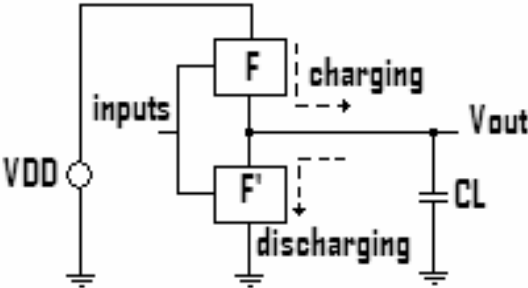


Figure (2.2.1) Conventional CMOS circuit [2].

Basic energy and charge conservation principles explain the switching energy and power dissipation on static fully restoring CMOS logic. In generic a CMOS gate that is shown in figure (2.2.1) and is loaded with a capacitor  $C_L$ . The load capacitor refers to the lumped parasitic input capacitances of the next logic stage. It is connected to supply voltage VDD through a pull-up network composed of “P” channel MOSFETs and same way is connected to the GND through a pull-down network of “n” channel MOSFETs. So  $C_L$  charges to VDD when pull-up network is tied and pull-down cut and will discharge when networks conditions swap. Consider  $Q$  is the charge size in process of charging then:



$$Q = C_L \cdot V_{DD} \quad (2.1)$$

So the energy that is supplied to  $C_L$  is:

$$E_C = \frac{1}{2} C_L \cdot V_{DD}^2 \quad (2.2)$$

Because energy is conserved, the other half must be dissipated by pull-up network regardless to the make-up of the resistance of the switches (PMOS), network and the time that is required to complete the charging. Similarly, during the discharge all of the signal energy stored on the capacitor is inevitably dissipated in the pull-down network. This is because no signal energy can enter to the GND rail ( $Q \cdot V_{GND} = Q \cdot 0$ ). The energy of charge or given energy from supplier is:

$$E_{VDD} = C_L \cdot V_{DD}^2 \quad (2.3)$$

However, the energy dissipated when a signal is cycled and it is fixed at twice the signal energy, hence the only way to reduce energy dissipation in conventional CMOS circuits is to reduce the signal energy and this leads to have more background noise sensitivity and thus the probability of malfunction. Consider figure (2.2.1) once again when NMOS and PMOS are substituted in pull-down and pull-up network receptively and consider resistor “R” for channel resistance and using constant charge current, so the dissipation through the channel resistance of pull-up(down) would be:

$$E_{dis} = P \cdot T = I^2 \cdot R \cdot T = \left( \frac{C_L \cdot V_{DD}}{T} \right)^2 \cdot R \cdot T = \frac{R \cdot C_L}{T} \cdot C_L \cdot V_{DD}^2 \quad (2.4)$$

Equation (2.4) shows dynamic charge/discharge power dissipation and it is guidance to low power and energy circuit design. Now we will look at those three different components of power dissipation individually and more in detail.

## 2.2.1 Static power

Technology scaling is one of the driving forces behind the tremendous improvement in performance, functionality and the power in integrated circuits over the past several years. However, as scaling continues for future technologies, the impact of sub-threshold leakage currents will become increasingly large.

In industry, the standard scaling methodology has been constant field scaling with 30% reduction of all dimensions per generation as summarized in table (2.1). In general, using constant field

scaling, physical dimensions ( $W, L, t_{\text{gox}}, X_j$ ) all scale by a factor  $1/S$ , substrate doping scales by  $S$ , and voltages ( $V_{CC}, V_{\text{tn}}, V_{\text{tp}}$ ) scale by  $1/S$ , where  $S$  is greater than unity. Consequently, device currents scale by  $1/S$ , gate capacitances scale by  $1/S$ , and intrinsic gate delays scale by  $1/S$ . Thus with 30% scaling of physical parameters, one can achieve close to a 50% improvement in frequency from generation to generation, although this will be degraded by worsening interconnect dominated delays [40].

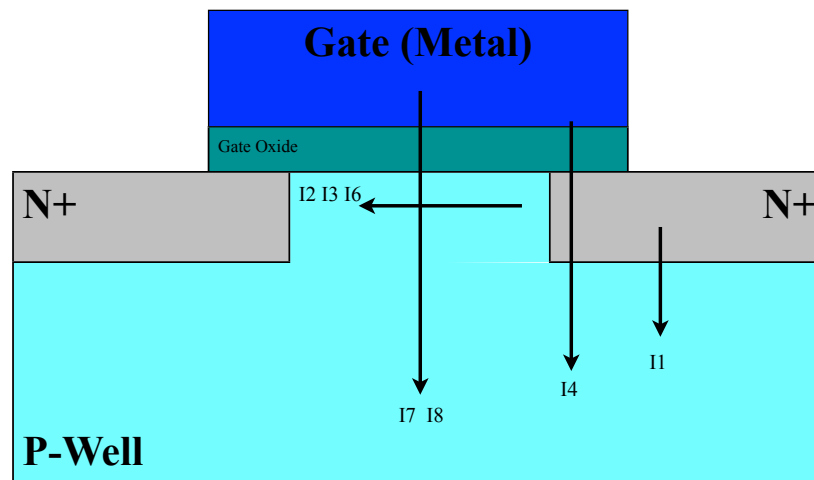
**Table 2.2.1 Technology Scaling Trends[40]**

<b>Scaling Parameter</b>	<b>1/S Constant Field Scaling</b>	<b>30% Scaling Field Scaling</b>
$W, L, T_{\text{gox}}, X_j$	$1/S$	0.7
Substrate doping	$S$	1.43
$V_{CC}, V_{\text{tn}}, V_{\text{tp}}$	$1/S$	0.7
$C_{\text{gate}}, I_{\text{max}}$	$1/S$	0.7
Propagation Delay	$1/S$	0.7
Frequency	$S$	1.43
Chip Dimension	$1/S^2$	0.5
Dynamic Power	$1/S^2$	0.5
Leakage Power	exponential	exponential
<b>Constant Die Assumption</b>		
Chip Dimension	1	1
Functionality	$S^2$	1.43
Dynamic Power (Constant Die)	1	1
Leakage Power (Constant Die)	exponential	exponential

The switching energy dissipated per event scales by  $1/S^3$  because of  $1/S$  constant field scaling, when the operating frequency increasing with scaling results the switching power dissipation scales by  $1/S^2$ . However, on the constant die size, dynamic power dissipation result of switching currents remains relatively constant with scaling. This is because of the number of switching elements that are used in the same die size which are increased by a factor of  $S^2$ .

On the other hand, leakage currents increase exponentially with a reduction in  $V_t$ , and furthermore the total effective width of the devices will increase by a factor of  $S$  [40].

Leakage current consumption is considered as static power consumption. Major elements of leakage current are shown in figure (2.2.2).



**Figure (2.2.2) Leakage Current Components [3].**

- $I_1$  is the reverse-bias p-n junction leakage caused by barrier emission and minority carrier diffusion and band-to-band tunneling. However this current has minimal contribution to total OFF current.
- $I_2$  is sub-threshold conduction current. This is Drain-Source current when Gate-Source voltage is lower than  $V_{TH}$ . This is a dominant component in leakage current and we will talk more in detail later in sub-threshold circuit section.
- $I_3$  results from the drain-induced barrier lowering (DIBL) effect. In general and ideally, DIBL does not change the sub-threshold slope but does lower  $V_{TH}$ .
- $I_4$  is gate-induced drain leakage (GIDL). The  $I_4$  is a result of the applied high electric field under gate-drain overlap region which causing a thinner depletion region of drain to well junction. GIDL is small for normal supply voltage but its impact rises at higher supply voltages (near burn-in).

- $I_5$  is channel punch-through. A punch-through current is a consequence of source and drain depletion regions merging into a single depletion region when channel current in sub-gate region is out of the gate voltage control.
- $I_6$  is the Narrow-Width effect current.
- $I_7$  is oxide leakage.
- $I_8$  is the gate current due to hot carrier injection.

In general all above currents are participating in two kinds of leakage current, first, ON leakage ( $I_7$  and  $I_8$ ) and second, OFF leakage currents which includes  $I_1$  through  $I_6$ . The main concern in terms of leakage is about the OFF current and therefore, the focus is on the current components  $I_1$  through  $I_6$ .

So the total leakage current assumption will be,

$$I_L = I_1 + I_2 + I_3 + I_4 + I_5 + I_6 = I_0 \cdot e^{\frac{V_{GS} - V_{th} + \eta V_{DS}}{nV_T}} \cdot [1 - e^{-\frac{V_{DS}}{V_T}}] \quad (2.5)$$

where right equation presents sub-threshold current in MOSFET moreover  $\eta$  and “n” are DIBL and sub-threshold slope coefficient.

So the total static power consumption would be,

$$P_S = I_L \cdot V_{DD} \quad (2.6)$$

### 2.2.2. Dynamic Power

Dynamic (switching) power is the main contributor to total the CMOS power consumption and mainly related to architecture and circuit speed requirements. Ever since the 0.5 $\mu$ m generation, the gate dielectric oxide thickness, supply voltage and threshold voltage have scaled with device dimensions to limit the growth of dynamic power consumption while improving performance which led to exponential increase in static leakage power. Looking at dynamic power we see whenever a capacitor, which represents parasitic or controlling-charge element, charges or discharges, there is power dissipation and equations (2.2) to (2.4) are applicable. Equation (2.4) clearly shows the effect of time and advantages of using constant current charge/discharge to control storage energy and power dissipation versus constant voltage.

In the charge process as we saw that in figure (2.2.1),  $C_L$  draws an energy equal to  $C_L \cdot V_{DD}^2$  from the power supply where  $C_L$  is the average total on-chip capacitance switched per cycle. Half of this energy is stored on  $C_L$ , while the other half is dissipated immediately as heat on the network (in the PMOS transistors and the capacitor). The discharge process similarly draws the stored energy in  $C_L$  (equation 2.2) and dissipates that on the NMOS pull-down network. Hence, the total dynamic power dissipation is also function of charge/discharge event probability ( $P_e$ ).

So the consumed power by switching of the capacitor over the period of  $T$  is,

$$P = \frac{E}{T} = \frac{C_L \cdot V_{DD}^2}{T} = C_L \cdot V_{DD}^2 \cdot f \quad (2.7)$$

Where  $f = 1/T$  is charge/discharge speed.

Considering activity factor ( $\alpha$ ) or event probability in equation (2.7), the total dynamic (switching) power becomes,

$$P_D = \alpha \cdot C_L \cdot V_{DD}^2 \cdot f \quad (2.8)$$

Equation (2.8) is a general function for dynamic power dissipation and there are some other sources which are hidden inside of parameters. One of the most common and major one which has an impact on  $P_D$  and increases dynamic power consumption is a glitch. A glitch mostly is hidden inside of probability of event. A glitch highly depends on the circuit architecture and signal timing in the circuit. The other item which has an impact on dynamic power is technology scaling. From table (2.2.1) we can see that dynamic power is scaled by  $(1/S^2)$  when  $V_{DD}$  scaled by  $1/S$ . But this is partially true and in reality, there is other fact that has an impact on total dynamic power. This is beyond the architecture affect like the glitch. This is about MOSFET properties and controlling charges; however it has consequence in architecture. From the theory of charge control devices, charges can be distinguished as either controlling or controlled charge. In a MOSFET, controlling charge is the required charges for the gate to do the switching while the controlled charge flows through the channel. In a digital circuit, logic levels (0 and 1) are related parameters to  $I_{ON}/I_{OFF}$ . The  $I_{OFF}$  is the leakage current and as we saw that in static power review, this current is increased due to scaling. So to have a valid logic, the  $I_{ON}$  also must increase in a same order. This mean we are in positive loop, because the channel current ratio

(controlled charges) is related directly to the controlling charge and  $C_g$ . We can see that in equation (2.9) and (2.10).

$$V_G = \frac{Q}{C} = \frac{I_{ON} - I_{OFF}}{C_g \cdot t_{pd}} \quad (2.9)$$

In saturation region and  $V_S=0$ , MOSFET current is,  $I_{DS} = \frac{K}{2} \cdot (V_G - V_{th})^2$  (2.10)

Hence, to have stronger  $I_{ON}$  or  $I_{DS}$ , transistors must consider to be stronger and that leads to larger capacitors and it requires more charge or current and this loop will continue.

The last but not least hidden component in dynamic power is power loss in wires and conductances. Lower voltage due to scaling in conjunction with higher current cause voltage drop on the internal resistors of wires and conductances. This voltage drop and power loss in most cases needs compensation to avoid logic level distortion and have better SNR.

### 2.2.3. Short Circuit

Short circuit power is consequence of signal rise and fall time. The fact is during those periods PMOS and NMOS or in general, pull-up and pull-down networks are ON, so there will be a path from  $V_{DD}$  to  $V_{SS}$  (GND). Short circuit power is part of dynamic power consumption due to its dependency on signal transition and it may presents differently in different digital logic structure (e.g. Static and Dynamic logic).

Basically, CMOS cells have a minimal period of short circuit current flow, but due to the slower operation in low voltage circuits, this period increases. Thereby, the short circuit power is a factor of the supply voltage and as it is shown in equation (2.11), it will consume less when voltage decrease. Note that the  $t_r$  and  $t_f$  parameters will increase because of  $V_{DD}$  reduction but not in linear fashion.

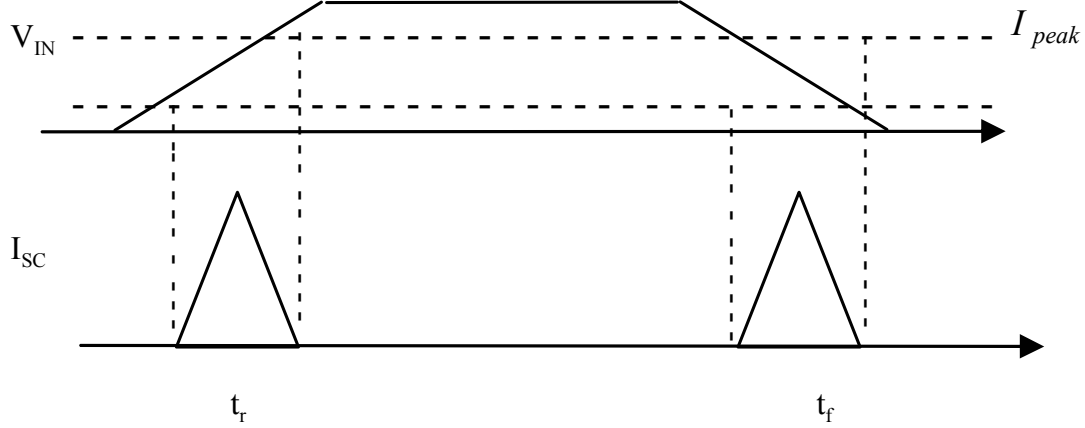
Consider short circuit spikes, approximately be a triangle and  $V_{DD}$  is bigger than  $V_{th}$ , as it is depicted in figure (2.2.3), hence, we can write,

$$P_{sc} = V_{DD} \left[ \frac{I_{pr} \cdot t_r}{2} + \frac{I_{pf} \cdot t_f}{2} \right] \cdot f \quad (2.11)$$

where  $I_{pr}$  : Pick current during the rise-time.  
 $I_{pf}$  : Pick current during the fall-time.  
 $t_r$  : Rise-time period.  
 $t_f$  : Fall-time period.  
 $f$  : circuit switching frequency.  
 $I_p$  : Saturation current.

If we consider  $I_{pr} = I_{pf}$  and apply switching activity factor ( $\alpha$ ) in equation (2.11), then we can rewrite that equation and simplify it to,

$$P_{sc} = \alpha \cdot V_{DD} \cdot I_p \frac{t_r + t_f}{2} \quad (2.12)$$



**Figure (2.2.3) Short circuit model [3].**

Thereby short circuit current can given by,

$$I_{sc} = I_p \cdot \frac{t_r + t_f}{2f} \quad (2.13)$$

### 2.3. Low Power and Low Energy Circuits Ideas

In last section we discussed briefly about power consumption and its sources in CMOS technology. We saw that in most digital circuit where there is no need for biasing, then switching power, or in general, dynamic power is major source for power dissipation. We also saw that

leakage currents which are sources for static power dissipation in non biased digital circuits also are growing because of technology scaling. From equations (2.6), (2.8) and (2.12), it is very obvious that supply voltage ( $V_{DD}$ ) has major role in both static and dynamic power dissipation; note that short circuit and glitch power dissipation are included in dynamic power dissipation. Hence supply voltage reduction is one of the most efficient and attractive solutions for low power circuit. For dynamic power, smaller capacitors help to improve power dissipation and it leads to optimum speed or frequency because of its effect on current of transistors and load reduction at the same time. Last but not least parameter that has direct effect on power, is activity factor ( $\alpha$ ). Controlling activity factor also helps to reduce static power when each block turns ON/OFF in its own turn. Using a pipeline architecture applies activity factor control idea and it makes parallelism more attractive. In this section we are looking more in detail about these parameters and their interaction with each other.

### 2.3.1. Low Voltage and Sub-Threshold Circuits

Lowering supply voltage is our goal but the challenge is the minimum applicable supply voltage which circuit can operate correctly. History of minimum voltage refers to as early as 1962, when Keyes published papers about the limitations of performance and power dissipation of digital circuits. He concluded that the minimum possible voltage limit is not much higher than the thermal voltage ( $KT/q = 25$  mV) but ultimately voltage must be above 500 mV for performance. Then Menial and Swanson in 1971 pushed voltage lower and showed CMOS circuits have the best power-speed product in comparison with TTL and ECL. Indeed that was true when leakage was low then. They showed a ring oscillator in 1972 which could work with 100mV. In 2001 another minimum voltage operation theory emerged. To achieve the lowest possible voltage, NMOS and PMOS, off-currents must be equalized and with this condition the ideal limit that they proposed was  $2nKT/q = 57$  mV. Another group presented an inverter using 180nm technology that could operate only at 70 mV. However they used a feedback to control the voltage to the wells to match NMOS and PMOS current. In 2002, Ono derived another minimum voltage limit by equating the NMOS and PMOS threshold voltages. They used triple well process and well voltage control and presented a SRAM bit that could operate at 175mV [1].



Transistor operation region depends on the applied supply voltage. Lowering supply voltage shifts the operation region from strong inversion to moderate and finally to weak inversion. The strong inversion region, also known as the super-threshold regime is characterized by large current drive and supply voltage substantially above the threshold voltage of the transistor,  $V_{th}$ . The moderate inversion however, has lower current drive in compare with the super-threshold regime. In this case, moderate inversion, transistors operate close to threshold voltage,  $V_{th}$ . Unlike the other two regions, the weak inversion region, which is known as the sub-threshold regime, is characterized by small current drive and supply voltage is below the  $V_{th}$ .

In sub-threshold operation, channel of transistor is not inverted and the source for the transistor current is diffusion. So from charge-based current models, transistor current in sub-threshold is given by [1],

$$I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{nV_T}} (1 - e^{\frac{-V_{DS}}{V_T}}) \quad (2.3.1)$$

where  $I_0$  is  $I_{DS}$  when  $V_{GS} = V_{th}$  and is given by [1],

$$I_0 = \mu_{eff} C_{OX} (n - 1) \frac{W}{L_{eff}} V_T^2 \quad (2.3.2)$$

Parameter “n” is sub-threshold slope factor and is given by [1],

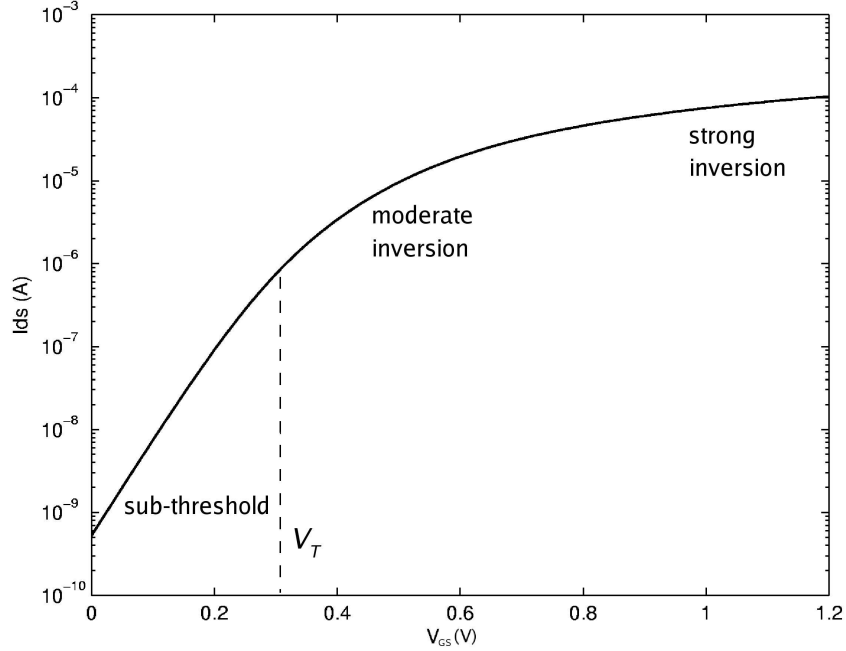
$$n = 1 + \frac{C_d}{C_{ox}} \quad (2.3.3)$$

Considering DIBL (Drain-Induced Barrier Lowering) effects in transistor current which was shown in equation (2.3.1), gives right model for transistor current in very weak inversion. This total current is given by,

$$I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}+\eta V_{DS}}{nV_T}} (1 - e^{\frac{-V_{DS}}{V_T}}) \quad (2.3.4)$$

where  $\eta$  is DIBL coefficient.

Figure (2.3.1) shows the logarithmic transistor current vs.  $V_{GS}$  in all three regions, sub-threshold moderate inversion and in the super-threshold regimes.



**Figure (2.3.1) Transistor current characteristics [36].**

The slope of  $I_D$  vs.  $V_{GS}$  in millivolts per decade of current changes represents  $1/S$  where  $S$  is slope factor and is given by,

$$S = nV_T \ln 10 \quad (2.3.5)$$

Results of transistor current ( $I_{on}$ ) in sub-threshold regime shows that the current is exponentially dependent on  $V_{GS}$ ,  $V_{th}$  and supply voltage. Hence the propagation delay and current matching between transistors are exponentially dependent on the voltages. Hence, voltage variation due to exponential dependence will be a major concern in sub-threshold design. For process variation, that can fall into global and local variations. Global variations affect all devices on a wafer similarly (i.e. discrepancies in alignment) with an effect seen in the sub-threshold region as strong PMOS or weak NMOS, or vice versa but local variations affect devices on the same wafer differently and consist of both systematic and random components. Typically, global variations is of most concern in digital CMOS design. However device mismatching is a consequence of local variation and threshold voltage ( $V_{th}$ ) variation models that. The standard deviation of threshold voltage approximately is proportional to  $(WL)^{-1/2}$  [1].

Temperature variation and its effects also has an impact on propagation and current mismatch. Two major temperature consequences on threshold voltage and mobility are given by [33],

$$V_{th}(T) = V_{th}(T_0) - K_c T \quad (2.3.6)$$

$$\mu(T) = \mu(T_0) \left(\frac{T}{T_0}\right)^{-M} \quad (2.3.7)$$

Where  $T_0 = 300^\circ\text{K}$  and  $K_c$  is the threshold voltage coefficient which typically is about  $2.4 \text{ mV}/^\circ\text{K}$  moreover  $M$  is the mobility temperature exponent with typical value around 1.5. In a strong inversion, lower mobility dominates in high temperatures and slows circuits but a lower  $V_{th}$  dominates in high temperatures and results in a lower delay in the sub-threshold region.

So as much as voltage variation in sub-threshold has an impact on speed of transistors, in comparison with the other regions, temperature variation in sub-threshold decreases delay. This is not fully compensated mechanism to keep the delay constant and in fact it causes some disorientation on the timing when synchronization is matter. Because of timing matter and anomalies delay in sub-threshold regime, glitching is common in combinational circuits. Consequences of these glitches are power dissipation and possible false signal generation. Following the previous section about CMOS power consumption, consider that,

$$E_{Total} = P_{Total} \cdot T \quad (2.3.8)$$

If we model entire circuit with  $C_{eff}$  then dynamic energy consumption will be,

$$E_{dyn} = C_{eff} \cdot V_{DD}^2 \quad (2.3.9)$$

Consider well-known delay  $t_d$  in an inverter which is given by [1],

$$t_d = \frac{K C_g V_{DD}}{(V_{DD} - V_{th})^\alpha} \quad (2.3.10)$$

Also we can rewrite operational frequency of “ $f_{op} = 1/T_{op}$ ” based on the depth of critical path “ $L_{DP}$ ” so the operating period is given by,

$$T_{op} = t_d \cdot L_{DP} \quad (2.3.11)$$

The static energy consumption is given by,

$$E_{st} = I_{leak} V_{DD} T_{op} \quad (2.3.12)$$

$$E_{st} = W_{eff} K C_g L_{DP} V_{DD}^2 e^{-\frac{V_{DD}}{n V_T}} \quad (2.3.13)$$

$$E_{total} = E_{st} + E_{dyn} \quad (2.3.14)$$

$$E_{total} = V_{DD}^2 [C_{eff} + W_{eff} K C_g L_{DP} e^{-\frac{V_{DD}}{n V_T}}] \quad (2.3.15)$$

Assuming a standard technology where  $V_{th}$  is fixed (i.e. no triple wells for body biasing), main task would be finding an optimum  $V_{DD}$  and related operational time (or frequency) to minimize the energy for a given design. Each design and architecture has its own critical path depth and so it requires different minimum  $V_{DD}$ . Regarding to (2.3.15) and “Lambert W” function and its constraints,  $V_{DD}(\text{optimum})$  is given by [1],

$$\frac{dE_{total}}{dV_{DD}} = 0 \quad (2.3.16)$$

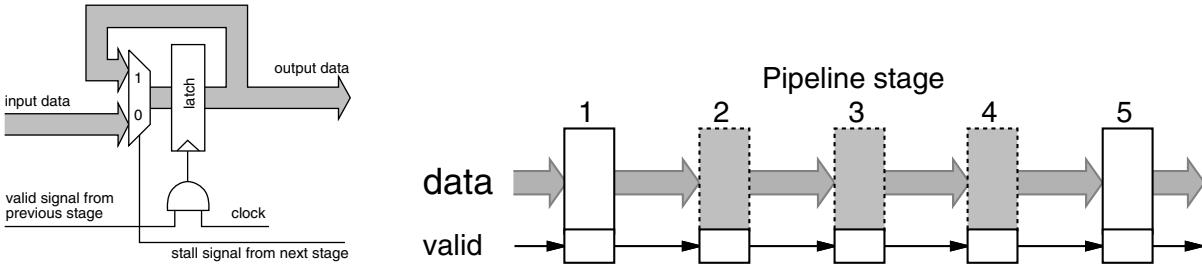
$$V_{DD}(\text{optimum}) = nV_T [2 - LambertW(\frac{-2C_{eff}}{W_{eff}KC_gL_{DP}}) e^2] \quad (2.3.17)$$

So in combinational circuit the optimum supply voltage is defined by (2.3.17). This defines optimum  $V_{DD}$  and it is not the minimum  $V_{DD}$  but it consider a reference point when minimum voltage is desired.

### 2.3.2. Pipelined and Self-Timed Circuits

Pipeline and parallelism were proposed to reduce power consumption by increasing the throughput of logic blocks and processors to reduce frequency and supply voltage. A pipelined execution unit presents a shorter stage delay than a non-pipelined execution unit [2]. It is therefore possible to work at the same operating frequency while reducing the supply voltage.

Pipelining is a technique to improve the resource utilization by forcing them to work in a given defined period. The main elements for the pipeline implementing are the gated clocks and the latch-based design data path. The idea is to provide and prepare an activation signal to be used in data path. So it consists of an AND gates, validation signal generators and global clock. Figure (2.3.2) shows the gated clock basics. Clock gating can be performed at many different levels of granularity. At the unit level, all pipeline stages of the unit are clocked as long as there is any instruction present in any stage of the unit. At the stage level, only the pipeline stages where instructions are present are clocked. Intuitively, finer grain clock gating result in larger power savings, but are also more complex to implement.



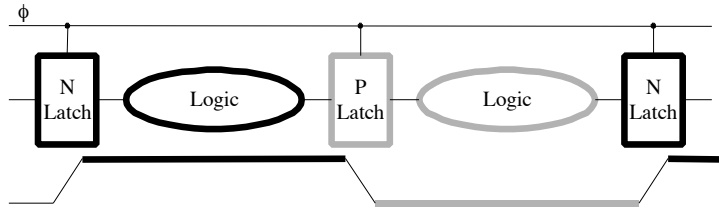
**Figure (2.3.2) Gated Clocks basics [41].**

The main limitations for the application of clock gating is timing on the clock gating signal and the ability to group latches with identical gating conditions. Some latch groups may be too small to be considered for clock gating due to design complexity and power overhead of the associated clock gating logic. With increasing wire delays, placement of latches close to the cone of logic feeding the data input may conflict with the placement necessary to group a set of latches for the purpose of clock gating. Also the required logic to compute when a latch must be clock-gated would become more complex and of course more power hungry.

The clock gating signal may also have to fan out to many clock drivers when the latch group is large. These delays may make it difficult to reach to the timing closure. Another problem is the inductive noise ( $Ldi/dt$ ) on supply voltage rails which is caused by clock-gating. To terminate surge currents which result from clock-gating, designers use on-chip decoupling capacitors that can contribute significantly to leakage power, thereby eroding some of the savings achieved through clock-gating [41].

The TSPC (True Single Phase Clock) and CDPD (Clock and Data Precharged Circuit Dynamic) are the most high throughput CMOS gated clock circuit techniques [2].

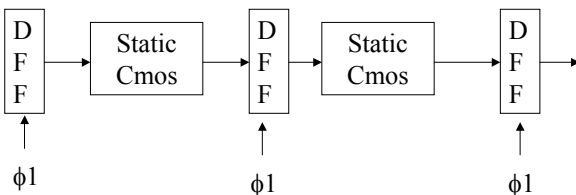
Short setup, hold and propagation delay time of TSPC contribute to high speed. The TSPC requires N and P-Blocks as it is depicted in figure (2.3.3). So the P block consists of a p-type latch which may embed logic, associated with the complementary logic gates before and after the p-latch and it is the same for n-block and using n-type latch. These blocks must be connected with N and P type latches alternately. The CDPD is an alternative solution for a fast one clock cycle decision and in the same time it reduces the power consumption [2].



**Figure (2.3.3) TSPC pipeline [45].**

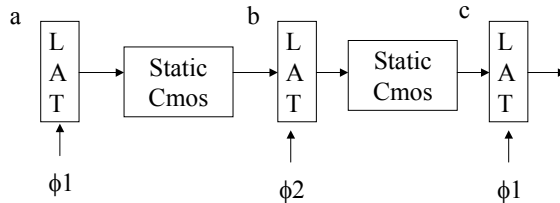
Domino logic often have been used for logic calculation. Figure (2.3.4) shows the most common architectures based on static and dynamic CMOS circuits.

**Pipelined DFF System (pulsed Latches) with Static CMOS**



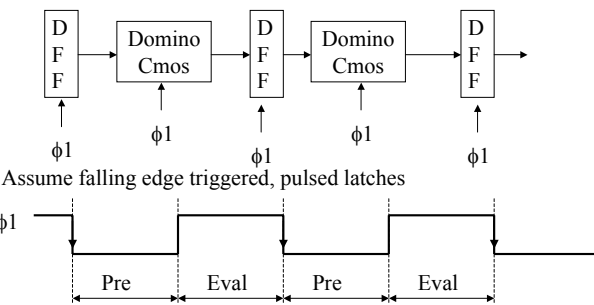
$$\text{Clock period} = T_{cq} + T_{cl} (\text{max})$$

**Pipelined Latch System with Static CMOS**



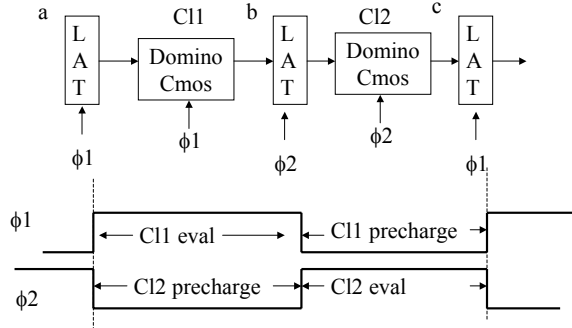
$$\text{Clock Period} = 2 * T_{c2q} + T_{cl} (\text{max path over both logic blocks})$$

**Domino Logic + FF system**



Assume falling edge triggered, pulsed latches  
A poor match, we are wasting time doing precharge. If Domino block same evaluation time as Static block, then slower than Static CMOS. Precharge time adds to clock period.

**Domino Logic + Latch system**

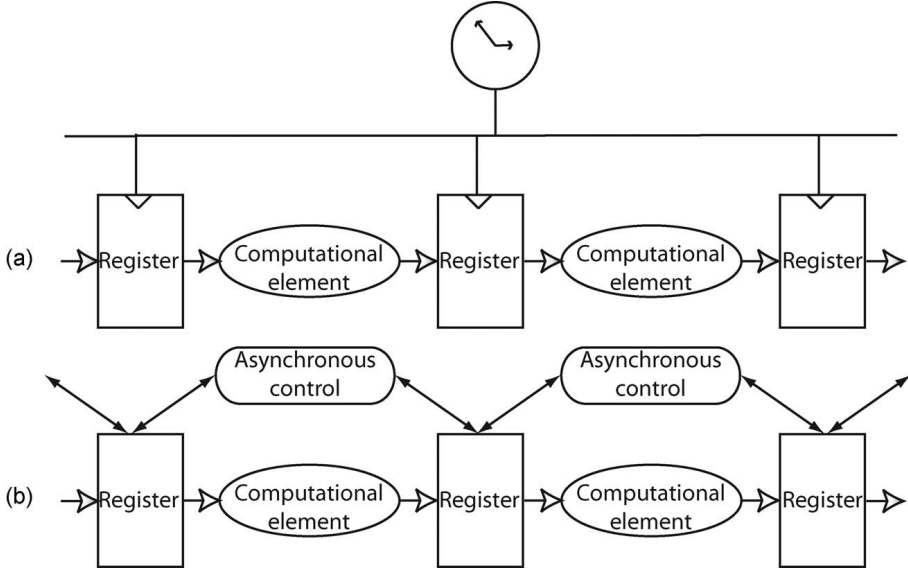


Clock period =  $2 * T_{c2q} + T_{cl}$  (max path), same delay as static CMOS system. Notice that precharge time is hidden.

**Figure (2.3.4) Most common pipeline architectures [45].**

In general, pipeline design falls into Synchronous and Asynchronous circuits. In synchronous, the clock is the main element to validate a block timing. In the other hand self-timed or asynchronous architectures have been proposed to reduce power consumption by removing the

clock tree which is known to be a relatively larger consumer. The asynchronous or clock-less circuits are conceptually similar to synchronous designs, in the sense that both circuits have registers for storing the inputs and results of a calculation and computational elements for transforming the data flowing in a circuit. In a synchronous design, the sequencing of the data from register to register is controlled by a (usually) global clock. However in clock-less circuit design, the sequencing of the data from register to register through the computation elements is controlled by some other means, an asynchronous control. So the components in an asynchronous circuit operates autonomously. They are not governed by clock circuitry or a global clock signal, but instead need only wait for the signals that indicate completion of instructions and operations. These signals are specified by simple data transfer protocols. The data through the stages propagate by means of handshake signals that signal propagation of the data. Figure (2.3.5) compare synchronous and asynchronous design structure.



**Figure (2.3.5) Comparison of (a) Synchronous and (b) Asynchronous circuit structure [45].**

There are several advantages and disadvantages of using asynchronous versus synchronous. Some of them are,

- Robust operation across PVT (Process, Voltage and Temperature) variations due to the elimination of the clock.

- Logically determined circuit design. Circuits are designed to function independent of the timing assumptions normally inherent in synchronous design approaches.
- Power management with very low latency.
- Low EMI and crosstalk.
- Modular composition and delay insensitive interfacing. The ability for individual blocks to automatically self-synchronize their data rates permits the designer to concentrate on the logical structure of the data flow.
- Complicated design approaches.
- Area/performance penalties.

### 2.3.3. Adiabatic-Switching

Adiabatic Logic is the term given to low-power electronic circuits that implement reversible logic. The term comes from the fact that an adiabatic process is one in which the total heat or energy in the system remains constant. Research in this area has mainly been retrieved by the fact that as circuits get smaller and faster, their energy dissipation greatly increases, a problem that adiabatic circuits promises to solve. Most research has focused on building adiabatic logic out of CMOS. However, current CMOS technology, though fairly energy efficient compared to similar technologies, dissipate energy as heat, mostly when switching. The fundamental reasons are, never turn on a transistor when there is a voltage difference between the drain and source and never turn off a transistor that has current flowing through it.

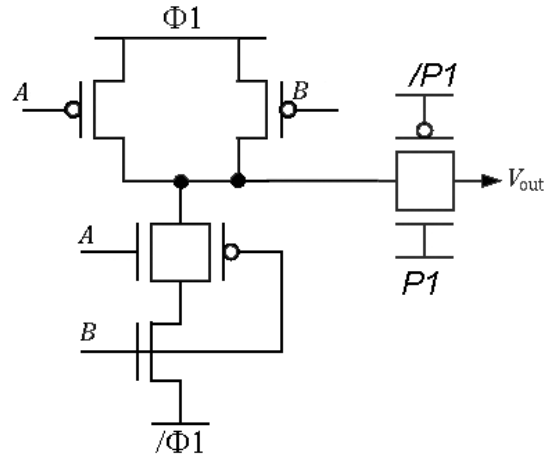
Several designs of adiabatic CMOS circuits have been developed. Some of the more interesting ones include split-level charge recovery logic (SCRL) [12] and Two Level Adiabatic Logic or 2LAL [13]. Both rely heavily on the transmission gates, use trapezoidal waves to clock the circuit and can be fully pipelined. CMOS transistors dissipate power when they switch. The main part of this dissipation is due to the need to charge and discharge the gate capacitance “C” through a component that has some resistivity “R”. The energy dissipated when, charging of the gate is equal to  $E = \frac{RC}{T}.CV^2$ ,

where “T” is the time it takes the gate to charge or discharge. In non-reversible circuits, the charging time “T” is proportional to the “RC”. Reversible logic uses the fact that a single clock

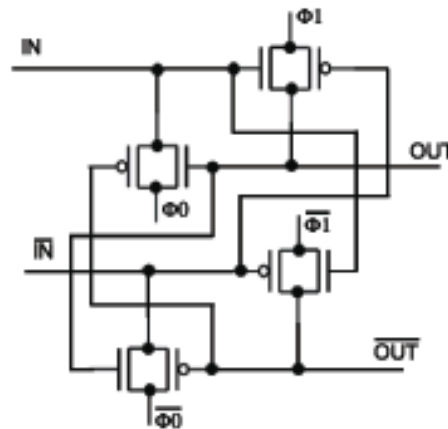


cycle is much longer than RC and thus attempts to spread the charging of the gate over the whole cycle and thus reduces the energy dissipated.

The SCRL NAND gate and 2LAL Buffer are shown in figure (2.3.6) and (2.3.7) respectively.



**Figure (2.3.6) SCRL NAND gate [46].**



**Figure (2.3.7) 2LAL Buffer [46].**

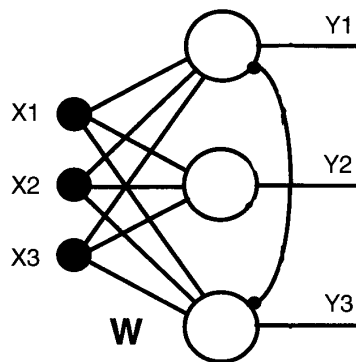
Figure (2.3.6) is very similar to a conventional NAND; however, one of the main differences is that the top and bottom rails are driven by trapezoidal clocks ( $\phi_1$  and  $\overline{\phi_1}$ ) rather than  $V_{DD}$  and  $V_{SS}$ . At the beginning the entire circuit is set to  $V_{DD}/2$  except for  $P1$  which is set to  $V_{SS}$  and  $\overline{P1}$  which is set to  $V_{DD}$  so that the transmission gate is off. In the next step, the transmission gate is turned on by gradually switching the  $P1$  and  $\overline{P1}$ . Following,  $\phi_1$  and  $\overline{\phi_1}$  which were at  $V_{DD}/2$  are split to  $V_{DD}$  and  $V_{SS}$  respectively. At this point, the gate computes the NAND of  $a$  and  $b$  like a non-adiabatic gate would. Once the output is used by the next gate, the transmission gate can be

turned back off gradually. Then  $\phi^1$  and  $\overline{\phi^1}$  are gradually returned to  $V_{DD}/2$  and now the input can change and the next cycle can begin. It is important not to change the input until the rails are back to  $V_{DD}/2$  so that a transistor is not turned on when there is a potential difference thus violating the first rule.

Figure (2.3.7) shows the basic buffer element of 2LAL which consist of two sets of transmission gates.  $\Phi_1$  and  $\Phi_0$  are both trapezoidal clocks but  $\Phi_1$  is a quarter cycle behind  $\Phi_0$ . Initially all the nodes are at 0. As the input gradually raises to 1 (if it is 1) or stays at 0,  $\Phi_0$  changes to 1. On the next step,  $\Phi_1$  changes to 1 which sets the output to 1 if the input was one and otherwise leaves it at 0 which that reduces the power dissipation because no charge passes through the transistor. On the third step  $\Phi_0$  goes back to 0 resetting the input to 0. Finally  $\Phi_1$  returns to 0 and the output is restored to 0 by the following gate in order to accommodate for full pipelining and thus the circuit is ready to process a new input.

### 2.3.4. Winner-Take All circuits

Winner-take-all nets are useful because they form part of the underlying basis of many well known neural network algorithms such as vector quantization and coding, optimization, self organizing feature maps, and adaptive resonance theory. The standard winner- take-all network is shown in Figure (2.3.8).



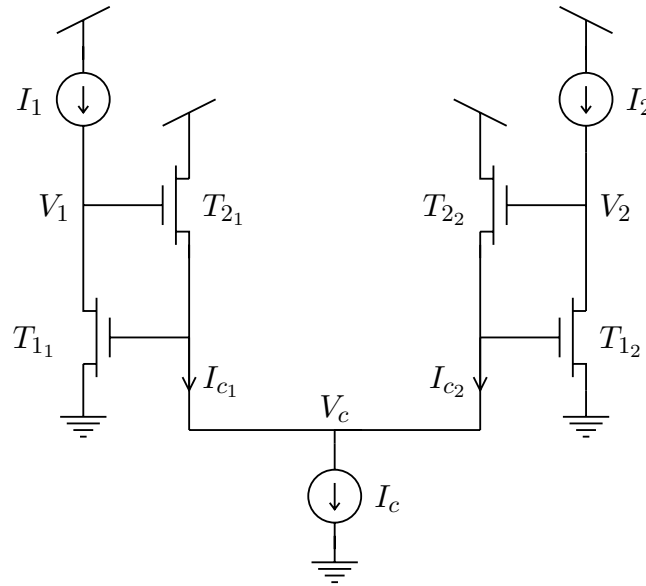
**Figure (2.3.8) Standard Winner-Take-All Network [19].**

The basic function of a winner-take-all net is to select the neuron that has the largest dot product of its weights and the incoming signals. In the standard network, this is done through the use of

lateral inhibition. The neuron with the largest initial activation (i.e. the neuron that has the largest dot product of its weights and the incoming signal) will inhibit the other neurons in the network the most. The result of this inhibition is the selection of one and only one neuron as the "winner". Figure (2.3.9) shows a schematic diagram of a two-neuron winner-take-all circuit. To understand the behavior of the circuit, first consider the input condition  $I_1 = I_2 \equiv I_m$ . Transistors  $T_{11}$  and  $T_{12}$  have identical potentials at gate and source, and are both sinking  $I_m$ ; thus, the drain potentials  $V_1$  and  $V_2$  must be equal. Transistors  $T_{21}$  and  $T_{22}$  have identical source, drain, and gate potentials, and therefore must sink the identical current  $I_{c1} = I_{c2} = I_c/2$ .

In the sub-threshold region of operation, the equation  $I_m = I_0 \text{EXP}(V_c/V_0)$  describes transistors  $T_{11}$  and  $T_{12}$ , where  $I_0$  is a fabrication parameter, and  $V_0 = kT/q\kappa$ . So the  $V_m$  is given by,

$$V_m = V_0 \ln\left(\frac{I_m}{I_0}\right) + V_0 \ln\left(\frac{I_c}{2I_0}\right) \quad (2.3.18)$$



**Figure (2.3.9) Two Channel Winner-Take-All Network [47].**

## 2.4. Summary

Designing for power and energy efficient designs has become a necessity for modern VLSI technologies. Constant electrical field scaling which cause leakage current to increase exponentially along with increasing integration capacity are the main sources of growing static power dissipation. Reviewing power consumption in CMOS circuits shows dynamic power dissipation result of switching still dominates. It was confirmed that supply voltage has major role in both static and dynamic power dissipation and voltage reduction is the most efficient solution for low power circuits.

Lowering the voltage causes transistors to operate in sub-threshold region. Increasing propagation delay is one of the main characteristics of transistor in sub-threshold region. This will be source of glitching which not only increases power dissipation but also can generate false signal. Pipelined and self-timed circuits are proposed to improve the resource utilization and efficiency of circuits. Synchronous and asynchronous are two main categories of pipelining architectures. Each one has advantages and disadvantages, however, asynchronous is more attractive for low-power design because it is clock less.

Then there is specific circuit implementation that use reversible logic to save energy. Adiabatic logic is the term given to this circuits. It requires several clock pulses with different phases to transfer energy from one point to the others.

# Chapter 3

## Adder Architectures

### 3.1. Introduction

Addition is one of the fundamental arithmetic operations and it has been used extensively in many VLSI systems such as microprocessors, DSP and other specific application architectures. In addition to its main task, which is adding two numbers, it is the nucleus of many other useful operations such as, subtraction, multiplication, address calculation and etc. It is also the speed limiting and more power consuming element as well. The design of faster, smaller and more efficient adder architecture has been aim and goal for many research efforts and has resulted in a large number of adder architectures. Each architecture provides different insight and thus suggests different implementations.

The power consumption and propagation delay are two most important properties of the adder circuit architectures which basically are against each other. That is knowing, lowering the power causes longer propagation delay and vice versa, hence, most architectures referring to one of those important properties. Nevertheless, in some cases they booth may compromised to achieve to low energy consumption. All architectures provide different insight and therefore require different implementation. This chapter provides overall and essential information and abstract of the most adder architectures in system level. In general full Adder function can introduce either using boolean logic function (conventional architecture) or the majority-function.

## 3.2. Boolean Logic Full Adder Function

A full adder boolean logic function is based on three inputs,  $(A_i, B_i, C_{i-1})$  and provides two outputs  $S_i$  and  $C_i$ . Equation (3.2.1) and (3.2.2) are sum and carry outputs with respect to their input.

$$S_i = A_i \oplus B_i \oplus C_{i-1} \quad (3.2.1)$$

$$C_i = A_i B_i + B_i C_{i-1} + A_i C_{i-1} \quad (3.2.2)$$

However it is most common and practical to use denoted characters  $P_i$  (Carry Propagate) and  $G_i$  (Carry Generate) and rewrite (3.2.1) and (3.2.2) by replacing (3.2.3) and (3.2.4).

$$P_i = A_i + B_i \quad (3.2.3)$$

$$G_i = A_i B_i \quad (3.2.4)$$

So sum and carry outputs is given by:

$$C_i = G_i + P_i C_{i-1} \quad (3.2.5)$$

$$S_i = P_i \oplus C_{i-1} \quad (3.2.6)$$

There are different solutions to implement n-bits full adder. First, architecture must be defined based on speed or power consumption. Then logic cell implementing will take place to finish the design cycle. In following section we will have a quick look at the most common n-bit full adder architectures.

## 3.3. Boolean Logic Full Adder Architectures

Since today many types and architectures are introduced for the full adder boolean logic but all they can conceptually categorized into two major groups:

(a) Carry Propagate (CP)

(b) Carry Look-Ahead (CLA)

In first group (group “a”), generated carry propagates from first to the last digit or it may skips some blocks. Based on equation (3.2.6), result of sum ( $S_i$ ) depends on this propagation. This group consider to have a linear base structure which requires less logic gates and so less power consuming. But it has long delay in compare with group “b”. Glitch in this group also is a

consequence of such a delay or propagation time in big adder and it has an impact on dynamic power consumption. Most full adder architectures design fall in to this group (“a”). Some of the very popular architectures in this group are:

- a.1- Ripple Carry Adder.
- a.2- Carry Skip Adder.
- a.3- Carry Select Adder.
- a.4- Carry Save Adder.

The second group (group “b”) however calculates the carry in advance to avoid that propagation time delay. For every bit, the ( $S_i$ ) is independent of last sum result so the ripple effect has thus been terminated. Because of that termination, the number of bits wont change addition time and it will be independent than bits numbers. However, due to increasing number of overhead gates in these circuits, propagation delay increases but still this group is faster than the first one in operation under certain condition. As a result of increasing number of gates, this group also consider high power consuming again in certain condition that we will see later. Some of the most popular architectures in this group are:

- b.1 Keogh-Stone Adder.
- b.2 Brent-Kung Adder.
- b.3 Han-Carlson Adder.
- b.4 Ladner-Fischer Adder.
- b.5 Ling Adder.

### 3.3.1 Ripple Carry Adder

The simplest addition architecture is based on a linear array of a full adder cell as it is depicted in figure (3.3.1). This architecture which also known as RCA has been subjected to be the smallest and the lowest power consuming. However according to the experimental results in this model, they show the average activity overhead (glitch) is about 50% [2]. The worst case delay or the critical delay path in N-Bit RCA is given by:

$$t_p = (N - 1)t_{carry} \quad (3.3.1)$$

where  $t_{\text{carry}}$  is the carry propagation delay from the input to the output.

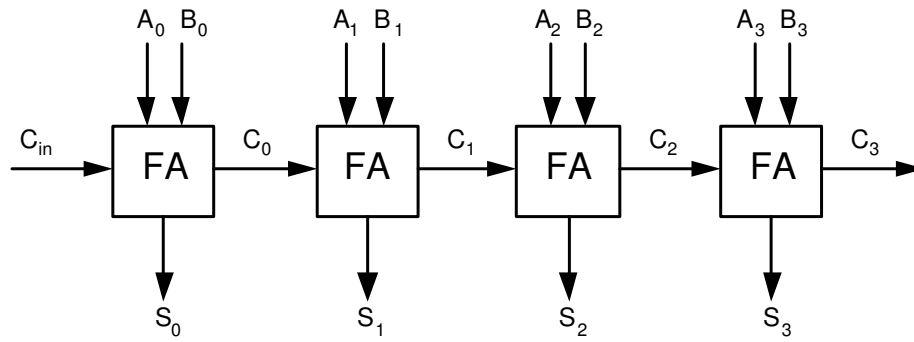


Figure (3.3.1) 4-bit Ripple Carry Adder [3].

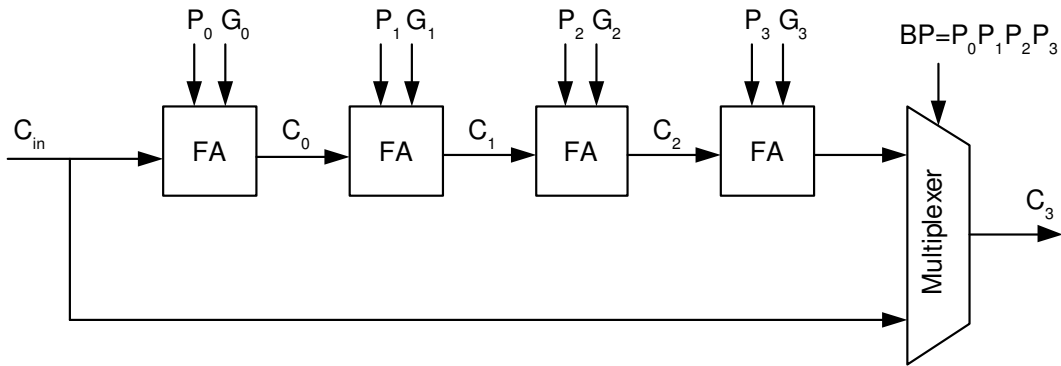


Figure (3.3.2) 4-bit Carry Skip Adder [3].

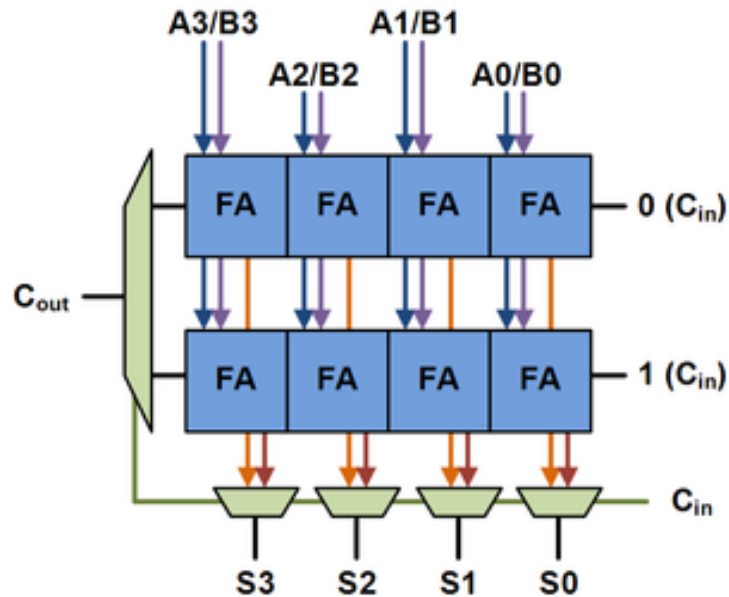


Figure (3.3.3) 4-bit Carry Select Adder [44].



### 3.3.2. Carry Skip Adder

The carry skip adder like RCA is based on linear structure but it takes advantage of extra gates to skip from designated blocks that logic gates are leading to make long critical path  $C_{in}$  to  $C_{out}$  shorter. Hence the critical propagation delay path in N-bit full adder when it is divided to M-bits groups is given by,

$$t_p = t_{pg} + Mt_{carry} + \left(\frac{N}{M} - 1\right)t_{skip} \quad (3.3.2)$$

Where  $t_{pg}$  is the required time to generate “P” and “G” signals and  $t_{carry}$  is the carry propagation delay in each group and  $t_{skip}$  is the multiplexer propagation delay. Note that multiplexer propagation delay in equation (3.3.2) consists of select signal generating delay (N/M bits AND logic) and data path delay.

### 3.3.3. Carry Select Adder

The main idea in a carry select adder is to split a sequential adder into two parts and performing the computation of most significant bit (MSB) part with considering the two possibilities for carry-in bit in parallel. The right generated carry then will be selected using the carry-out bit of the least significant bit (LSB). In this case the critical delay path in N-bit full adder when it is divided to M-bit group is given by,

$$t_p = Mt_{carry} + \left(\frac{N}{M}\right)t_{multiplexer} \quad (3.3.3)$$

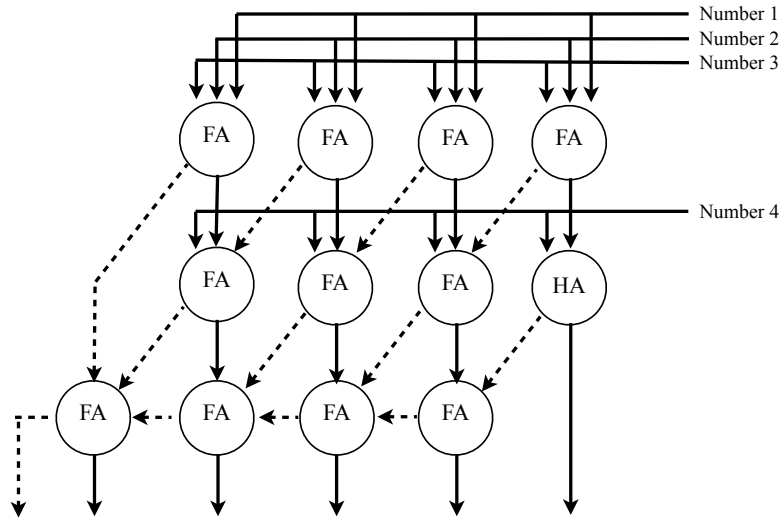
$$M = \sqrt{\frac{N}{2}} \quad (3.3.4)$$

Where  $t_{carry}$  is the carry propagation delay in each group and  $t_{multiplexer}$  is the multiplexer propagation delay. Figure (3.3.3) shows the idea of carry select adder.

### 3.3.4. Carry Save Adder

There are many cases where it is desired to add more than two numbers together. The straightforward way of adding together N numbers (all M bits wide) is to add the first two, then add that sum to the next, and so on. This requires a total of (N – 1) additions, for a total gate delay of (N.log M). Using carry save adder, the delay can be reduced further still. The idea is to

take these three numbers that we want to add together,  $X + Y + Z$ , and convert it into two numbers  $C + S$  such that  $X + Y + Z = C + S$ . The carry save adder consists of a ladder of stand alone full adders, and carries out a number of partial additions. The principal idea is that carry has a higher power of two and thus is routed to the next column. Doing additions with Carry save adder saves time and logic. Figure (3.3.4) shows the general idea of carry save adder with four operands.



**Figure (3.3.4) 4-operands Carry Save Adder [43].**

Carry save adder which also known as CSA comparison with the other standard adder is not straightforward. CSA systems require more bits than the other for the same interval of representable value. This leads to storage or additional bus resources, hence for a N-digit CSA,  $2N$  bits are required [9].

### 3.3.5. Brent-Kung Adder

This is the adder from group “b” or carry look ahead. The main idea of carry look ahead (CLA) is an attempt to generate all incoming carries in parallel and avoid waiting until the correct carry propagates from the first stage. A new Boolean operator which is called “Dot operator or  $(.)$ ” is introduced as,

$$(G, P).(G', P') = (G + PG', PP') \quad (3.3.5)$$

### 3.3.6. Kogge-Stone Adder

Kogge-stone adder is similar to Brent-Kung adder in principle. The only difference is that it uses the idempotent property. In this architecture, the adjacent bits are grouped based on the cell sizes and they are reused by adjacent nodes. Therefore fan-out is equal to the cell size and it has the least number of levels comparing to other structures. If the number of inputs is  $K$ , then the total cost or the number of used cells is  $K \log_2 K - (K - 1)$  and number of levels is  $\log_2 K$  [9].

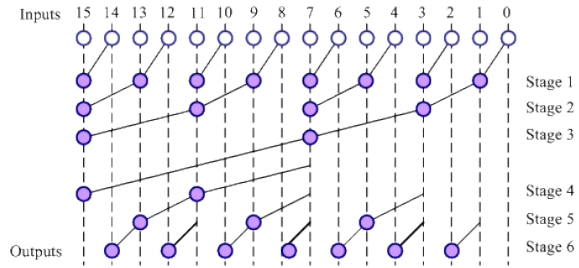
Figure (3.3.6) shows 16-bit Kogge-Stone Adder implementation.

### 3.3.7. Han-Carlson Adder

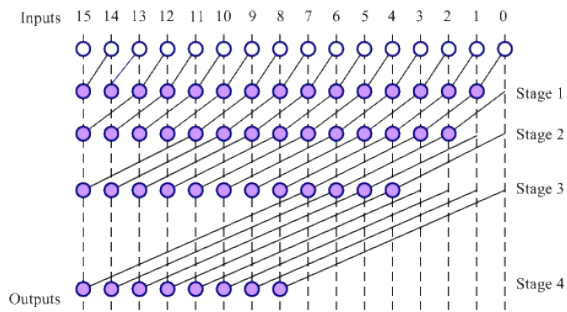
Han-Carlson adder is another architecture of prefix adders which is similar to architecture of Kogge-Stone, but it has area-time trade-off. In other words, it increases the logic depth for a reduction in fan-out. In this architecture, at the first level, bits are grouped based on the cell sizes and at the second level, the nodes have  $N$  number of inputs from previous level results based on the cell sizes ( $N$ ) [9]. Figure (3.3.7) shows 16-bit Han-Carlson Adder implementation.

### 3.3.8. Lander-Fischer Adder

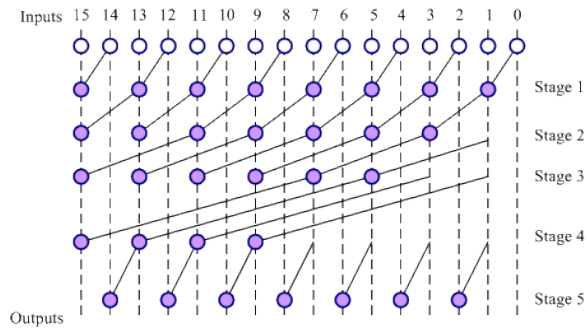
Ladner-Fischer adder is an improved version of Sklansky adder, where the maximum fan-out is reduced. Ladner-Fischer formulated a parallel prefix network design space which included this minimal depth case. In general this adder structure has logic depth similar to Han-Carlson where that is equal to  $(\log_2 n + 1)$  and it is higher than Sklansky and Kogge-Stone which they are limited to  $(\log_2 n)$ . In terms of fan-out however, it still has a large fan-out requirement up to  $n/2$  in compare with the others techniques. Figure (3.3.8) shows 16-bit Lander-Fischer Adder implementation. The number of computation nodes is similar to Sklansky and Han-Carlson is given by  $(n/2)(\log_2 n)$ .



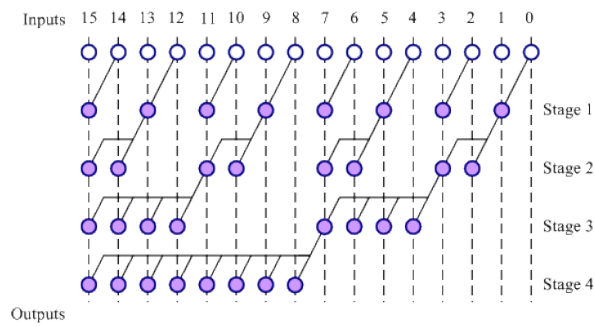
**Figure (3.3.5) 16-bit Brent-Kung Adder [43].**



**Figure (3.3.6) 16-bit Kogge-Stone Adder [43].**



**Figure (3.3.7) 16-bit Han-Carlson Adder [43].**



**Figure (3.3.8) 16-bit Lander-Fischer Adder [43].**

### 3.3.9. Parallel Adder Taxonomy Revisited

There are many other kinds of parallel adder which can fit in each axes of the parallel adder tree taxonomy. The taxonomy is three dimensional graph based on logic depth, wire tracks and fan-out and it helps to summarize group “b”. Figure (3.3.9) shows some of the above adders in that tree.

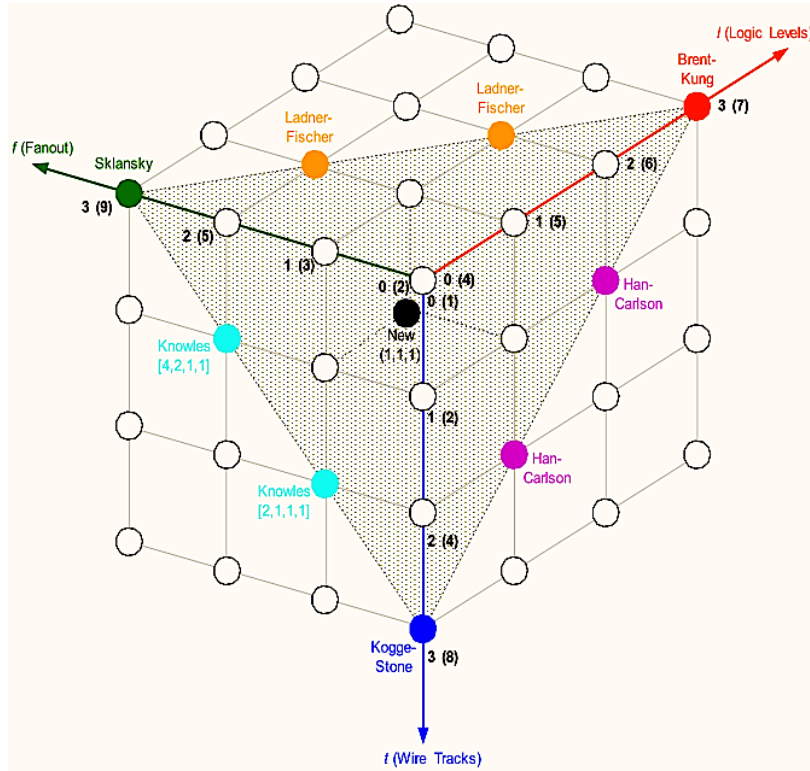


Figure (3.3.9) Parallel Adder Taxonomy Revisited [42].

### 3.4. Low Power Full Adder Architectures Comparison

Low power designs have often been compared based on area or total gate count. But gate count does not show the impact of transistor sizing and supply voltage scaling on energy and delay. Different arithmetic algorithms have been proposed in order to improve computational efficiency in terms of speed, area, and regularity of structures. In low power applications however, evaluating the energy efficiency of the algorithm is crucial. Research for low power adders lacks the framework for analyzing and quantifying the energy ramifications of different algorithm

choices and their implementations. Delay estimation of designs was initially based on the number of logic levels. The notion of fan-in and fan-out considerations for the delay was expanded into a comprehensive method known as “Logical Effort”. Table (3.4.1) compares some of the best known adder architectures in terms of gates count and complexity [11].

Adder Type (32-bit)	Gate Count	Complexity
RCA	161	208
CSK	197	245
VBA	209	254
CSA	248	423
CLSK	272	338
KS	404	461

**Table 3.4.1 Adder Architectures comparison.**

Where in this table RCA refers to Ripple-Carry Adder, CSK means Carry-Skip Adder, VBA is Variable Block Adder, CSA presents Carry-Select Adder, CLSK refers to Carry-Lookahead-Skip Adder and KS presents Kogge-Stone Adder architecture. There are many architectures for the adder and more than that, there are many different implementations for each architecture but not many references to compares architectures using best implementation in terms of power or energy consumption. Hence table (3.4.1) just gives an idea about the differences. Neither gate count nor complexity can be used as a figure of merit for energy efficiency because they do not consider impact of switching activity, parasitics and wiring and gate sizing on energy. To evaluate design in above table, the following definition of merit for “efficiency” is used as:

$$Efficiency = \frac{10,000}{D \times T} \quad (3.4.1)$$

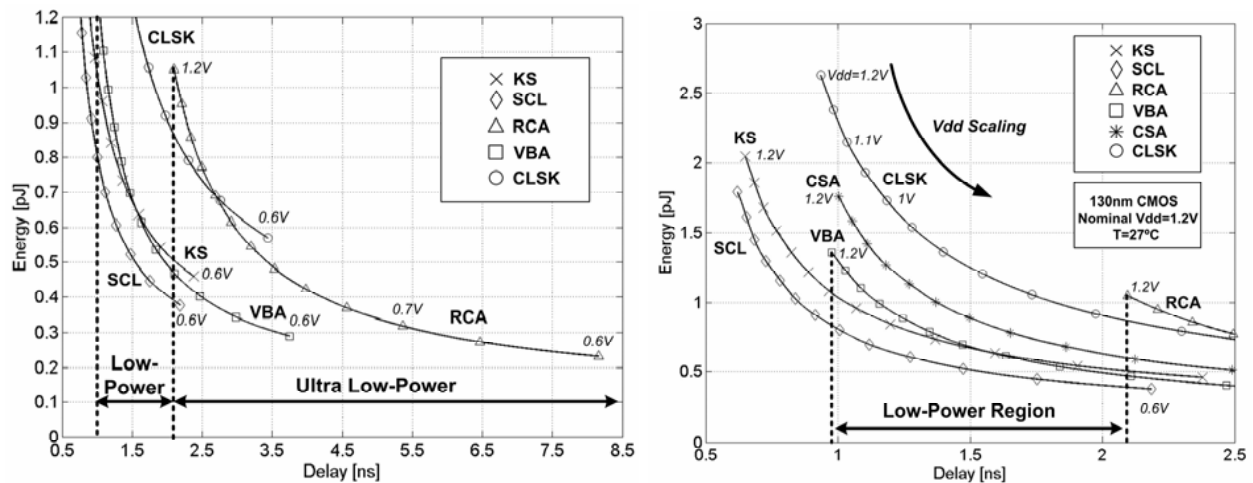
Where D refers to worst case delay or critical path and T represents the average number of gate transitions per addition [11]. In that comparison experience all above architectures are simulated using technology 130nm with applied 1.2V at 27°C. Note that in that simulation SCL (Spares Carry-Lookahead) adder was used instead of CSK which is initially RCA with improved critical path and results are shown in the table (3.4.2).

Adder Type (32-bit)	Delay (ns)	Av. Energy (pJ)	EDP(pJ/GHz)	Gate Count
RCA	2.1	1.1	2.31	161
VBA	0.98	1.38	1.35	209
CSA	1	1.78	1.78	248
CLSK	0.94	2.63	2.47	272
SCL	0.62	1.78	1.1	315
KS	0.65	2.04	1.3	404

**Table 3.4.2 Adder Architectures characteristics at 1.2V, 27°C.**

In first glance Gate Count vs. Av. Energy may give an impression about gate count impact on power or energy consumption. Average energy was measured on a set of 500 random input test vectors. The delay of each adder is obtained from simulation of the critical path vectors [11].

All adders in that experience had been sized for minimal energy. Figure (3.4.1) shows above adder architectures when supply voltage was swept from 1.2V to 0.6V with 50mV span.



**Figure (3.4.1) 32-bit Low-Power Adder comparison[11].**

The energy-delay results demonstrate that when designing for low power a comparison of designs at a single voltage or a comparison based on gate count is insufficient for determining the optimal structures [11]. However there are many other parameters that have not considered in

this experiment such as synchronous and asynchronous, so for energy consideration, time delay was multiplied by consumed average power. It also shows that in ultra low power RCA may consider as an competitive option where power or energy has higher priority than delay or speed.

### 3.5. Majority Function

Majority-function (also called the median operator), is a function from “n” inputs to one output. The value of the operation is false when “n/2” or more arguments are false, and it is true otherwise. Alternatively, representing true values as “1” and false values as “0”, we may use equation (3.4.2),

$$Majority(P_1, \dots, P_n) = \left[ \frac{1}{2} + \frac{\sum_{i=1}^n (P_i - \frac{1}{2})}{n} \right] \quad (3.4.2)$$

Where the “-1/2” in the formula serves to break ties in favor of zeros when “n” is even; a similar formula can be used for a function that breaks ties in favor of ones. For n = 3 the ternary median operator can be expressed using conjunction and disjunction as,

$$MF(x, y, z) = xy + yz + zx \quad (3.4.3)$$

Remarkably this expression denotes “carry” in logical addition. So the generated carry and sum in logical addition is given by,

$$C_{out} = MF(x, y, z) \quad (3.4.4)$$

$$S = \overline{MF(\overline{MF(x, y, \bar{z})}, C_{out}, \bar{z})} \quad (3.4.5)$$

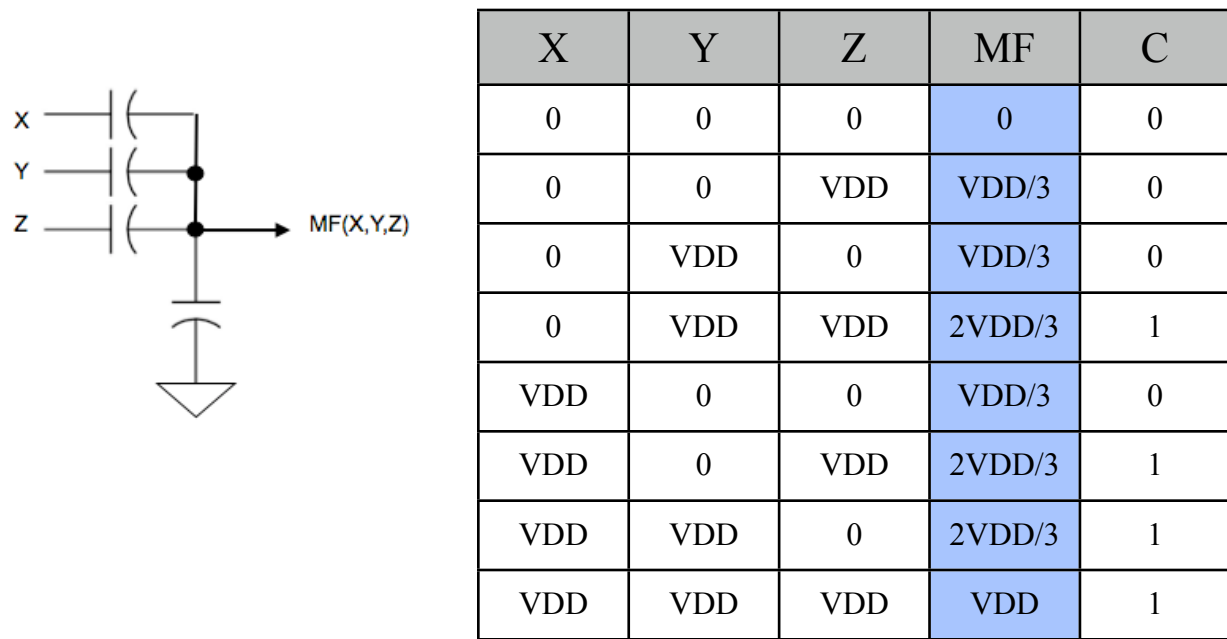
Where “z” represents  $C_{in}$  in equations above.

Logical circuit implementing using majority-function will be attractive because of the “XOR”. The fact this gate has long propagation delay in compare with other standard logical gates and it is in the critical path in conventional addition but majority-function doesn’t need or use this gate.

### 3.6. Majority Function Architectures

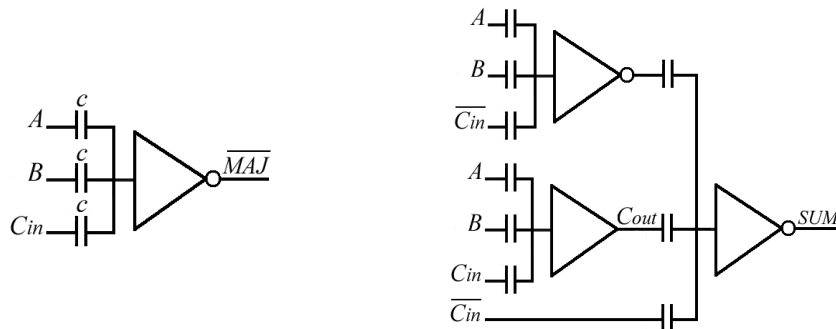
There is only one architecture so far has been introduced for a logical full adder using majority function. In that only architecture, a network of passive capacitors are introduced to present majority function which is depicted in figure (3.5.1).





**Figure (3.5.1) Majority Function in Voltage Mode.**

In figure (3.5.1) we can see that, “Carry = 1” when MF is greater than  $VDD/2$  and “Carry = 0” when MF is less than  $VDD/2$ . Note that voltage swing or voltage precision in that circuit is as small as  $VDD/3$  and that is an advantage of this architecture, but it also limits the minimum supply voltage to three time higher than the other architectures. Figure (3.5.1) also shows simple idea of MF implementation circuit. Thereby, carry generation is one step operation and propagation delay has less impact on the results. Equation (3.4.5) is depicted in figure (3.5.2) which is showing VMMF (Voltage-Mode Majority-Function). Note that XOR gate has not been used to generate carry [6].



**Figure (3.5.2) Majority Function Full Adder implementing using VMMF [6].**

Figure (3.5.3) shows the average power consumption comparison between CPL (Complementary Pass Logic), conventional and VMMF adder [7]. The PDP comparison is depicted in figure (3.5.4). Simulations are performed in  $0.18\mu\text{m}$  and voltage varies from 1.8V to 0.4V at room temperature.

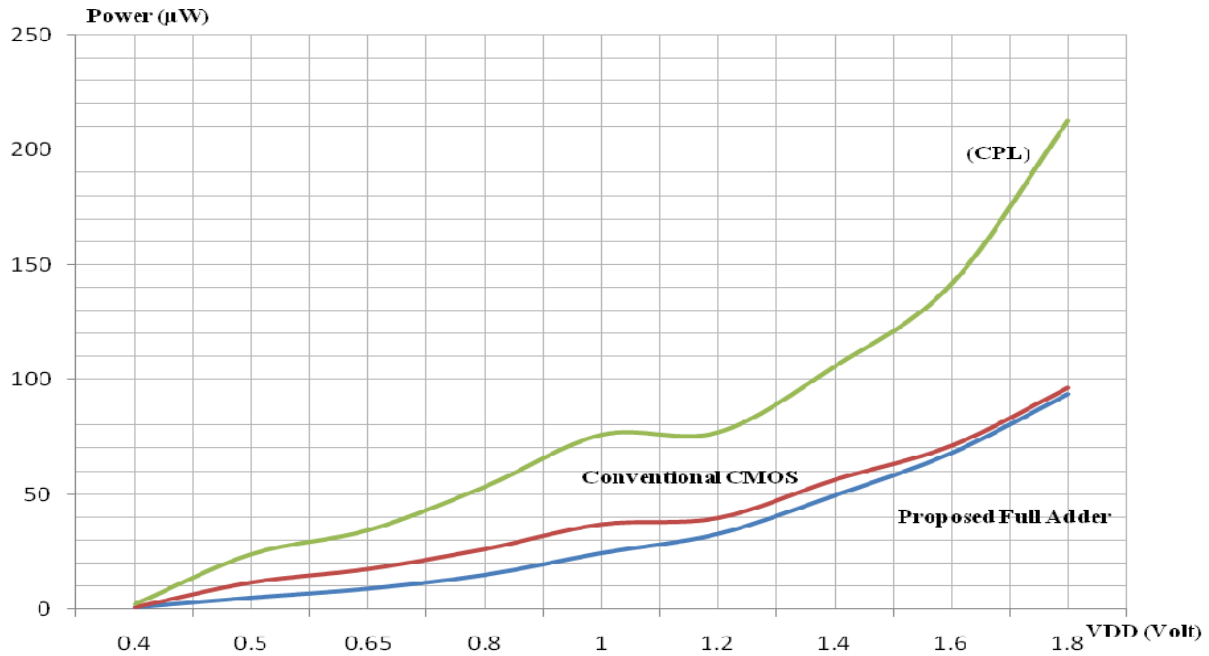


Figure (3.5.3) Average Power Comparison [7].

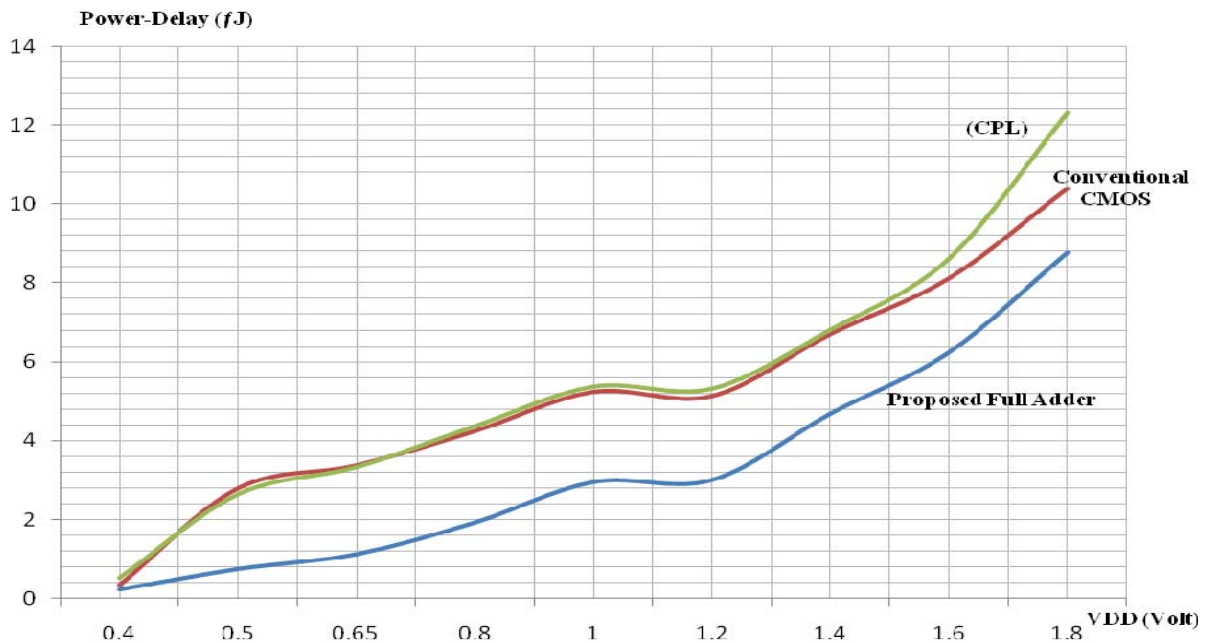


Figure (3.5.4) PDP Curves comparison [7].

### 3.7. Summary

Addition is one of the fundamental arithmetic operations and it has been used extensively in many VLSI systems. In addition to its main task, which is adding two numbers, it is the nucleus of many other useful operations such as, subtraction, multiplication, address calculation and etc. It is also the speed limiting and more power consuming element as well. The power consumption and propagation delay are two most important properties of the adder circuit architectures which basically are against each other.

Two main group of full adder architectures are, carry propagation and carry look-ahead. Result of propagation delay comparison clearly presents carry look-ahead group has less delay than the first group, carry propagation. However, comparison of power consumption in both groups confirms that the first group, carry propagation, specifically ripple carry adder from that group, consume less power.

Along with boolean logic full adder architectures, majority-function full adder is proposed which it is not based on conventional logic. Majority-function, is a function from “n” inputs to one output. That value of the operation is false when “n/2” or more arguments are false and it is true otherwise. Proposed voltage-mode majority-function full adder is based on front-end capacitor network to realize the majority-function. This algorithm is attractive because it doesn't need XOR gate which has longest propagation delay compare with other logical gates. Also it provides direct calculation, so logic depth wont be matter in this architecture.

# Chapter 4

## Ultra low power Current-Mode Majority

### Function Full-Adder

#### 4.1. Introduction

In last three chapters we discussed power consumption in CMOS circuits and low power and energy design techniques. Then we reviewed some of the full adder architectures and their comparison in terms of power and energy consumption. Now in this chapter, results of our last discussions will use as a guidance. These results like pieces of puzzle will picture desired architecture and implementation when we put them together.

#### 4.2. Project overview and guidance

First and very obvious step regarding to chapter 2 is to drop the supply voltage as low as possible and that leads to sub-threshold operation. Section (2.3.1) showed the history of tried and proposed minimum voltage. Consider an inverter which is the basic digital gate in sub-threshold and if temporary we assume that there is no DIBL or Early-Voltage effects in transistors, then the maximal gain of the inverter,  $A_{inv}$ , occurs at the switching threshold of  $V_{DD}/2$  is given by,

$$A_{inv} = \frac{g_m^{NMOS} + g_m^{PMOS}}{g_d^{NMOS} + g_d^{PMOS}} \quad (4.1.1)$$

Where  $g_m$  and  $g_d$  are defined as the corresponding partial derivation of  $i_{ds}$  vs. input voltage and output voltage respectively [5]. Considering sub-threshold current equation we can rewrite (4.1.1) to,

$$A_{inv} = \left(\frac{k_n+k_p}{2}\right) \frac{1-e^{-\frac{V_{DD}}{2V_T}}}{e^{-\frac{V_{DD}}{2V_T}}} \quad (4.1.2)$$

Hence, in order to have  $A_{inv} > 1$  the minimum supply voltage is given by [5]

$$V_{DD}^{Min} = 2V_T \ln\left(\frac{k_n+k_p+2}{k_n+k_p}\right) \quad (4.1.3)$$

Considering  $k_n = k_p = k = 1/n$  and  $n$  is given by

$$n = 1 + \frac{C_d}{C_{ox}} \quad (4.1.4)$$

Therefore, minimum  $V_{DD}$  in (4.1.3) can simplified and rewritten as,

$$V_{DD}^{Min} = 2V_T(n + 1) \quad (4.1.5)$$

Using (2.3.5) in (4.1.5) gives,

$$V_{DD}^{Min} = 2V_T + 0.87S \quad (4.1.6)$$

where  $S$  is slope factor and  $V_T$  is thermal voltage.

However in most practical circuits that voltage supply in excess of 100 mV, the gain of CMOS inverter (in sub-threshold) is set by DIBL effects rather than by saturation effects and it is given by,

$$A_{inv} = \frac{k_n+k_p}{\eta_n+\eta_p} \quad (4.1.7)$$

where  $\eta$  is DIBL coefficient.

The other result of chapter two was about pipeline architecture advantages in order to eliminate activity factor and improve the resources utilization. Also we saw that the advantage of asynchronous and self timing in comparison with synchronous clocking in a pipeline architecture. So the goal is to utilize asynchronous pipelined in sub-threshold circuit.

In chapter three most important full adder architectures in each group are reviewed with their power and energy consumption comparison. The result in table (3.4.2) showed that the RCA (Ripple Carry Adder) consume less power and energy in average. Note that its total energy consumption was higher than the other due to its delay. In other words the total energy was given by,

$$E_{total} = t_{total} \times P_{total} = T_{Delay} \sum_{i=1}^n P_i \quad (4.1.8)$$

where, “ $n$ ” is the total number of bits,  $T_{Delay}$  is a propagation delay from fist to last bits and  $P_i$  is the power consumption in each stage. Using an asynchronous pipeline architecture to rebuild

RCA changes equation (4.1.8) and breaks the big propagation delay to small pieces where they contribute with only related stages. So total energy will be,

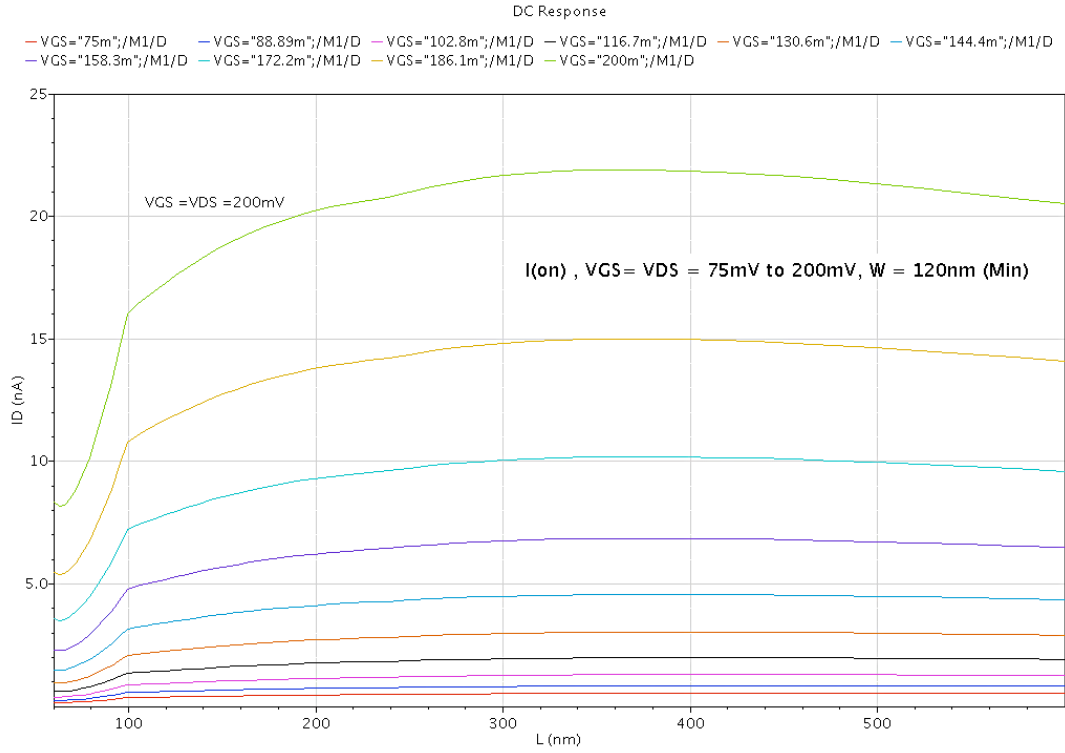
$$E_{total} = \sum_{i=1}^n t_i \times P_i \quad (4.1.9)$$

The last part in our puzzle and design cycle is suitable architecture to implement. In very low voltage circuits like the full adder, critical path, which in this case is carry generation, is the key point. Figure (3.5.4) indicates that majority function has better result in compare with the other circuits in that comparison. We can easily expect that due to majority function structure and the fact that carry generation doesn't require complicated and multi-stage gates. So our aim is to introduce sub-threshold asynchronous pipeline RCA based on majority-function. In next section and first, we look at transistors, specifically NMOS, characterization in sub-threshold and weak inversion.

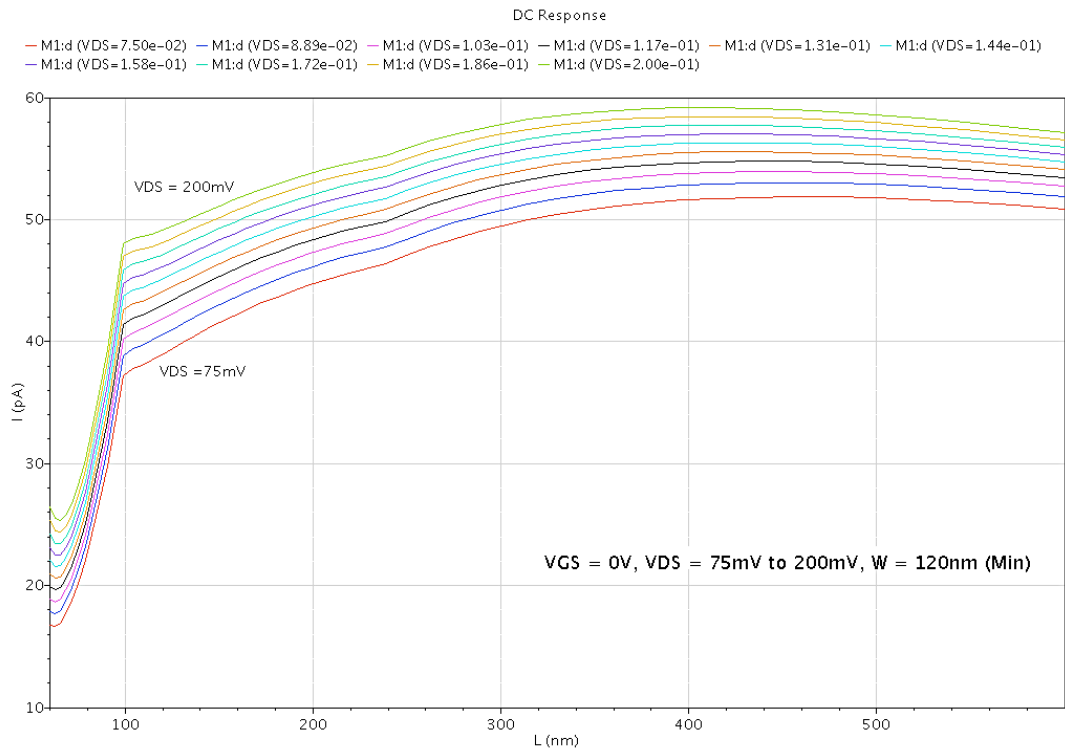
### 4.3. Weak inversion transistor sizing and characteristics

One of the first important step prior to design is to test transistors (e.g. NMOS) characterizations in sub-threshold and in weak inversion. The effects of sizing "L" on the "ON" and "OFF" or leakage currents are showed in figures (4.3.1) and (4.3.2) respectively. Results show that increasing "L" doesn't do much neither on "ON" nor "OFF" current in weak inversion. In this experience "W" was kept as minimum and supply voltage varied from 75 mV to 200 mV.

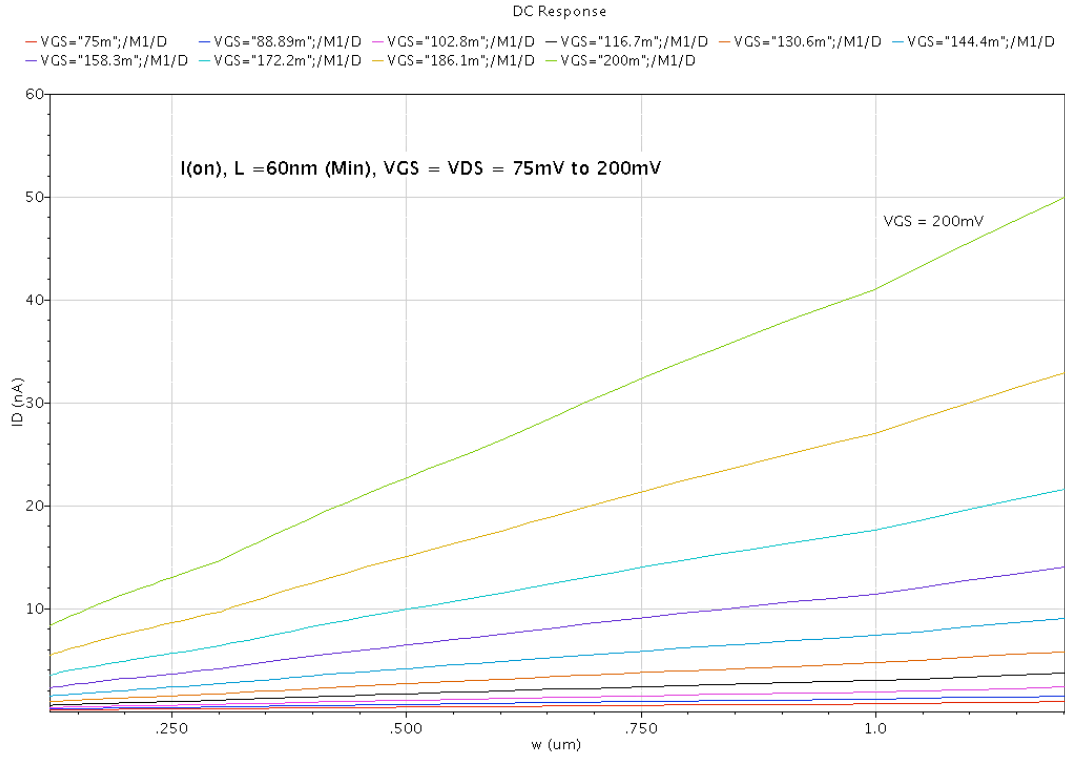
Also results show that leakage current variation due to supply voltage variation is very small in compare with "ON" current variation. The other result from this simulation was nonlinearity on both "ON" current as well as leakage current when "L" was linearly increased. The leakage current doesn't exactly follows the same pattern in "ON" current, hence sizing "L" may is not applicable on weak inversion. The effects of sizing "W" on "ON" and "OFF" or leakage currents are showed in figures (4.3.3) and (4.3.4) respectively. Unlike increasing "L" size, results show that increasing size of "W" linearly, it will increase "ON" current almost linearly as well as leakage current. Hence, due to the results of sizing "L" and "W", practically "W" sizing will be applicable in weak inversion when linearity is matter. Also in different operation point where "W" is not minimum, the pattern due to varying "L" size may be different than when "W" is minimum.



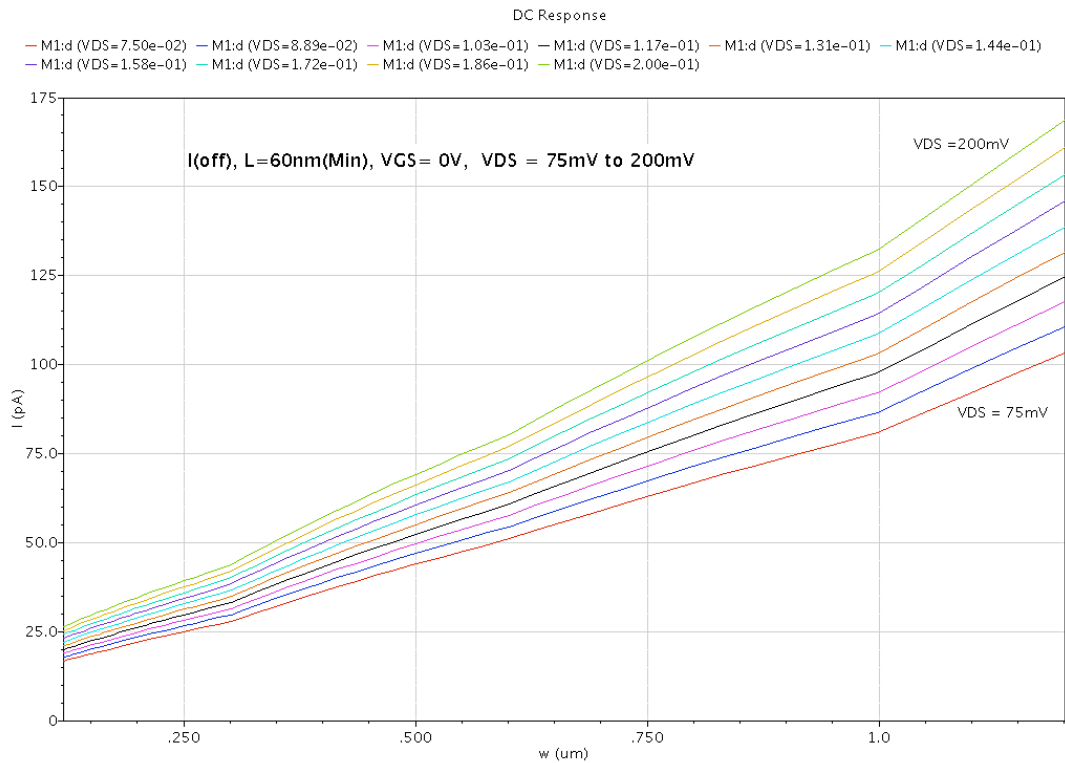
**Figure(4.3.1) “L” size effect on the  $I_D$  in voltage variation when “W” is Min.**



**Figure(4.3.2) “L” size effect on leakage in voltage variation when “W” is Min.**



**Figure(4.3.3) “W” size effect on the  $I_D$  in voltage variation when “L” is Min.**

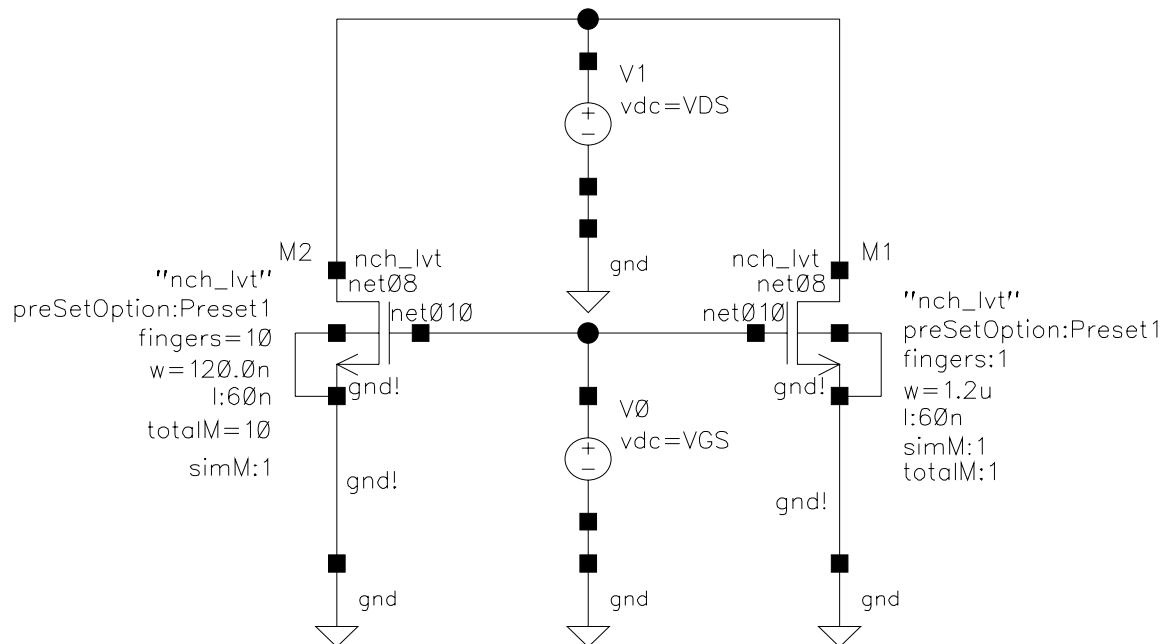


**Figure(4.3.4) “W” size effect on leakage in voltage variation when “L” is Min.**



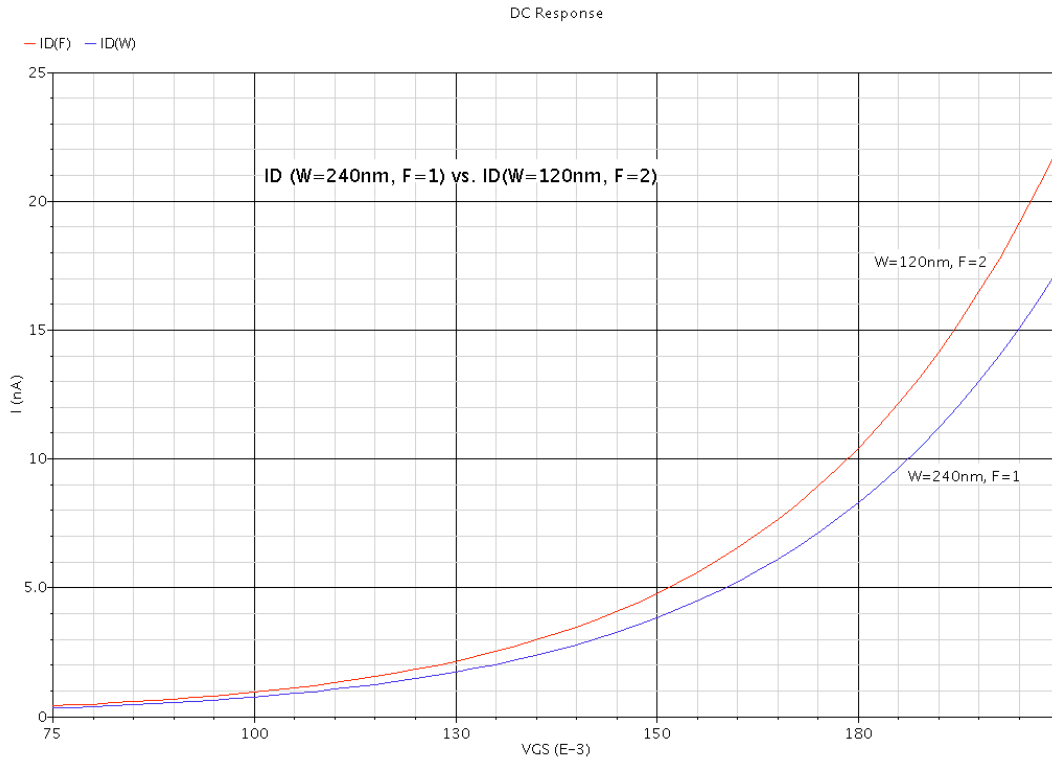
The other important parameter in transistor sizing is a parameter called finger number or “F”. Low operating current is the one of the sub-threshold drawback, hence transistor has to be sized big enough in order to have sufficient “ON” current. Transistor total width can increase either by increasing “W” or increasing finger number “F” and it is given by,  $W_{Total} = W \times F$ .

A general assumption is to have almost identical current on two transistors when both have identical total size in an identical condition. However in sub-threshold and weak inversion that is not the case. Figures (4.3.5) shows two identical transistors, one with  $W=240$  nm and  $F=1$  which gives  $W_{Total} = (240 \text{ nm} \times 1) = 240$  nm and the other one with  $W=120$  nm and  $F=2$  which gives also  $W_{Total} = (120 \text{ nm} \times 2) = 240$  nm.

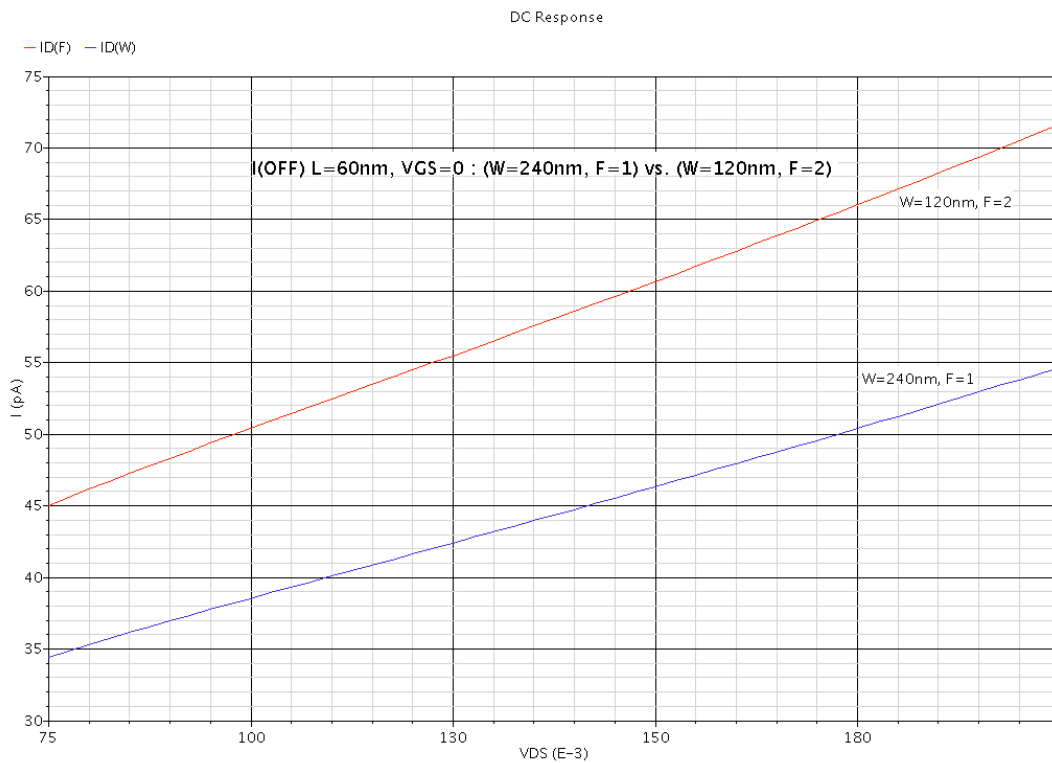


**Figure(4.3.5) Transistor parameters (Total width) test and comparison.**

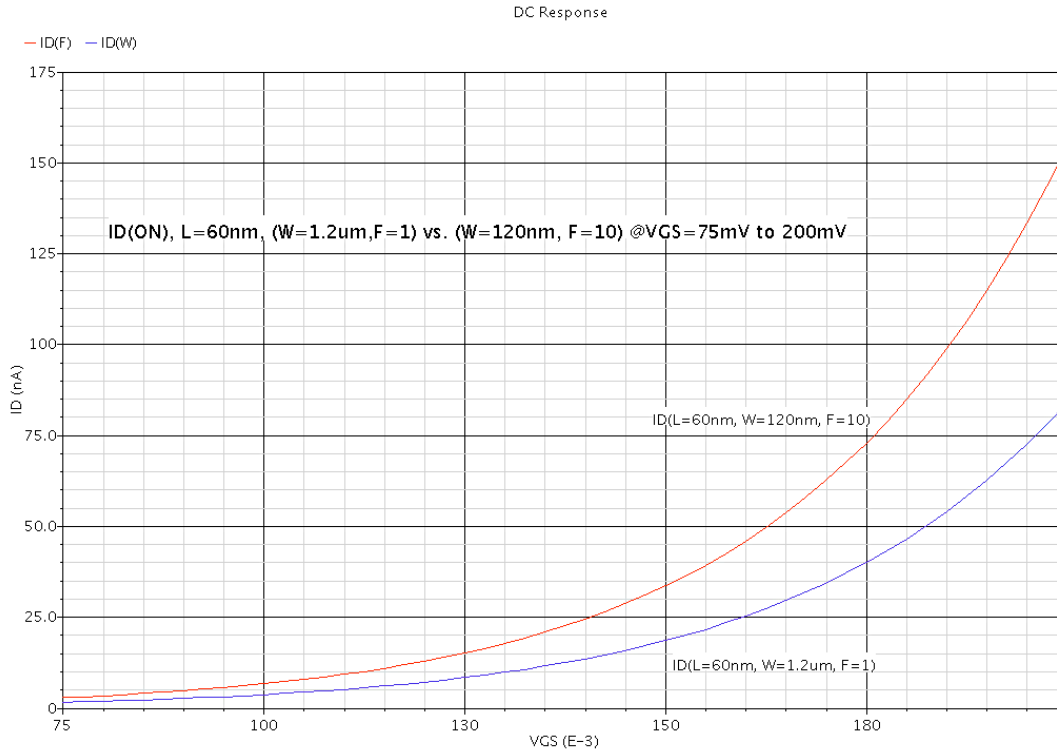
Results of this simulation are depicted in figure (4.3.6) and (4.3.7). Figure (4.3.6) shows “ON” current and figure (4.3.7) presents leakage current. Results show that at the same condition transistor with greater “F” parameter provides better conductivity than higher “W”. This difference on the  $I_D$  between two transistors will be bigger as “F” increases. Figure (4.3.8) compares  $I_D$  of two transistors with identical  $W_{Total}$  but one with  $L=60$  nm,  $W=1.2$   $\mu\text{m}$ ,  $F=1$  and the other with  $L=60$  nm,  $W=120$  nm,  $F=10$ .



**Figure(4.3.6) Effect of “F” parameter on Transistors current.**

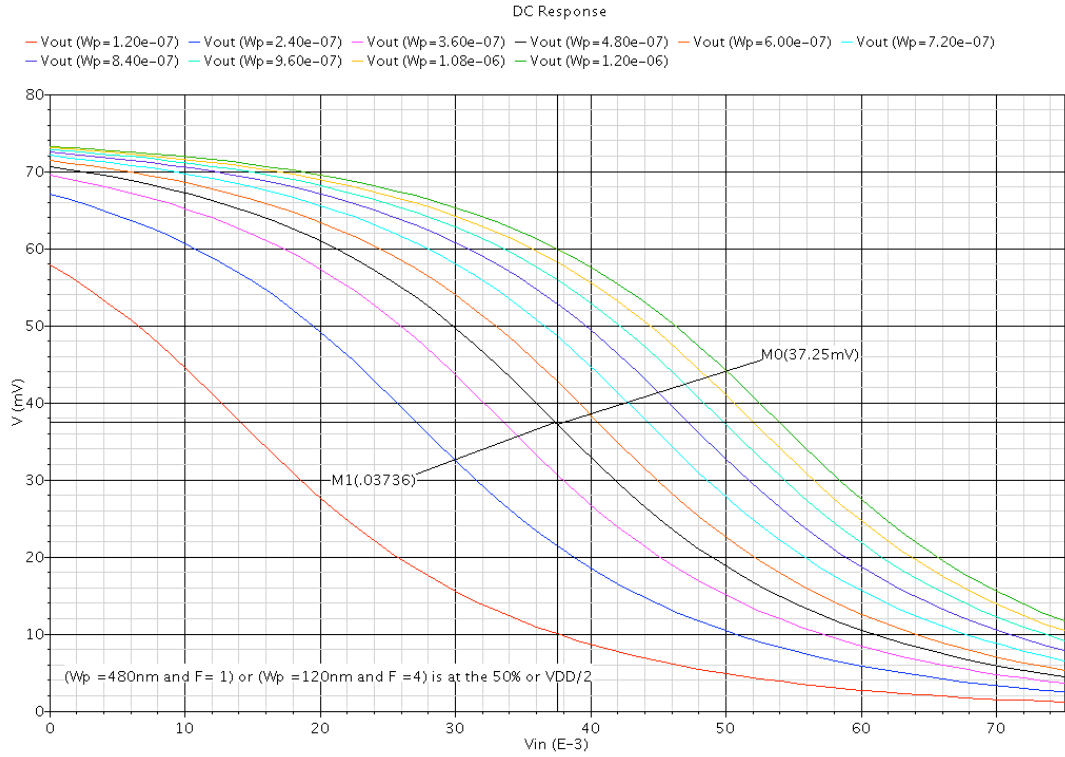


**Figure(4.3.7) Effect of “F” parameter on Transistors leakage.**

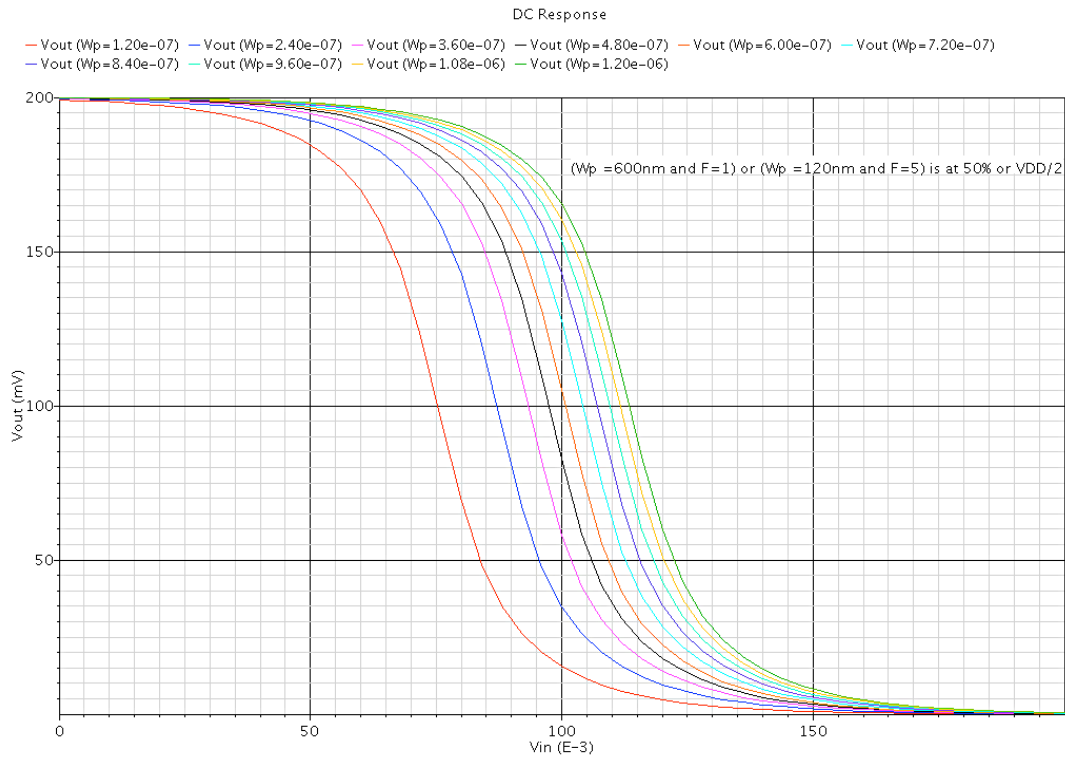


**Figure(4.3.8) Effect of greater “F” parameter on Transistors current.**

However, less flexibility will be the consequence of using “F” versus “W” for transistor sizing. In terms of using “F” step size will become bigger because “F” has to be an integer. In fact, transistor size increase by integer factor of minimum “W”. This discrepancy on the current becomes critical when we use both “F” and “W” to size and ratio two transistors. Nevertheless, using “F” improves slope factor and helps transistor works more efficient in sub-threshold and weak inversion. All the results are applicable to PMOS and the current discrepancy is worse than NMOS. The ratio of NMOS and PMOS in sub-threshold is the other important fact. An inverter is an important and basic element in digital logic and its VTC shows the best ratio of NMOS and PMOS in sub-threshold. A minimum NMOS size was considered in an INV circuit to achieve minimum energy consumption. Using VTC gives PMOS sizes and ratio to maintain 10% to 90% voltage swing as they are depicted in figures (4.3.9) and (4.3.10) when operating voltage varies from 75 mV to 200 mV respectively.



**Figure(4.3.9) VTC vs. PMOS sizes at 75mV.**



**Figure(4.3.10) VTC vs. PMOS sizes at 200mV.**

Considering test results, transistor parameters and their sizing impact, parameter “F” takes place instead of “W” whenever it is applicable and “L” sizing wont be used due to its nonlinear impacts and at last, the ratio of three will use due to general voltage swing and weakness of PMOS. It also pushes inverter switching voltage,  $V_m$ , below  $V_{DD}/2$  and that helps to use less power during the evaluation which we will see that later.

#### 4.4. Current-Mode Majority Function FA implementation

In sections (3.5) and (3.6) we discussed about the majority-function full adder and its so far only implemented circuit that it has been introduced. The circuit was based on passive capacitors network and taking advantage of lowering voltage swing to  $V_{DD}/3$  but also it suffers from minimum applicable supply voltage because of that lowering. The capacitors must be sized such that transistor parasitic capacitors have less impact on the output voltage of network. Relatively big capacitors in front-end network has impact on speed and energy consumption of circuit. Using current instead of voltage can help to improve circuit operation where a single node in circuit perform analog current addition. Table (4.4.1) and (4.4.2) show carry and sum generation in majority-function respectively. In table (4.4.1), carry generation, consider logic “1” is represented by reference current “I” and vice versa and no current refers to logic “0”. Thereby  $I_x$  column in table (4.4.1) presents analog sum of input signals  $A_i$ ,  $B_i$  and  $C_{i-1}$  in current-mode. It also presents a single node current where adding all incoming and outgoing currents must becomes zero. Table (4.4.1) confirms that,

$$C_i = 0 \quad \text{if} \begin{cases} I_x = 0, \\ I_x = I, \end{cases}$$

$$C_i = 1 \quad \text{if} \begin{cases} I_x = 2I, \\ I_x = 3I, \end{cases}$$

So the threshold current of “(1.5)I” is a boundary condition where “I” refers to input logic “1” and the statements above can result,

$$C_i = \begin{cases} 0 & \text{if } I_x < (\frac{3}{2})I, \\ 1 & \text{if } x > (\frac{3}{2})I. \end{cases} \quad (4.4.1)$$

Therefor a single current comparator can generates the carry with respect to majority-function.

$A_i$	$B_i$	$C_{i-1}$	$C_i$	$I_x$	Logic
0	0	0	0	0	I <sub>x</sub> < 1.5I
0	0	1	0	I	
0	1	0	0	I	
1	0	0	0	I	
0	1	1	1	2I	I <sub>x</sub> > 1.5I
1	0	1	1	2I	
1	1	0	1	2I	
1	1	1	1	3I	

**Table 4.4.1 Carry generation in Majority-Function full adder.**

$\overline{C}_i$	$A_i$	$B_i$	$C_{i-1}$	$S_i$	$I_y$	Logic
1	0	0	0	0	mI	I <sub>y</sub> < G.I
0	0	1	1	0	2I	
0	1	0	1	0	2I	
0	1	1	0	0	2I	
1	0	0	1	1	(m+1)I	I <sub>y</sub> > G.I
1	0	1	0	1	(m+1)I	
1	1	0	0	1	(m+1)I	
0	1	1	1	1	3I	

**Table 4.4.2 Sum generation in Majority-Function full adder.**

Sum generation is depicted in table (4.4.2) with regard to same assumption. It converts logic to the current and vice versa. The only difference in here is that  $\overline{C}_i$  is replaced by  $m \times I$  current in total current  $I_y$ . Note that all conditions below must be considered in table (4.4.2).

$$\begin{cases} (m+1)I > 2I, \\ (m+1)I > mI, \\ mI < 3I. \end{cases} \quad (4.4.2)$$

Equation (4.4.2) results,

$$\begin{cases} m > 1, \\ I > 0, \\ m < 3. \end{cases} \quad (4.4.3)$$

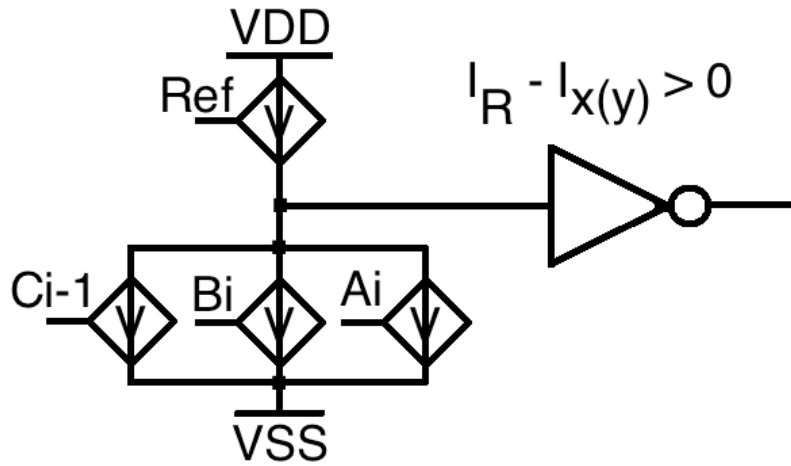
Thereby,  $1 < \mathbf{m} = \mathbf{2} < 3$  and considering  $m=2$ , we can replace threshold current “G.I” to “(2.5)I” and table (4.4.2) will change to table (4.4.3). Similar to carry generation, a single current comparator generates sum with respect to majority-function.

$\overline{C}_i$	$A_i$	$B_i$	$C_{i-1}$	$S_i$	$I_y$	Logic
1	0	0	0	0	2I	I <sub>y</sub> < 2.5I
0	0	1	1	0	2I	
0	1	0	1	0	2I	
0	1	1	0	0	2I	
1	0	0	1	1	3I	I <sub>y</sub> > 2.5I
1	0	1	0	1	3I	
1	1	0	0	1	3I	
0	1	1	1	1	3I	

**Table 4.4.3 Simplified Sum generation in Majority Function full adder.**

The easy way of implementing a current-mode majority-function is depicted in figure (4.4.1). It requires a single node and a current comparator but with the cost of increasing static current and power consumption. Using duality and converting current to voltage at very last step however, helps to improve static current consumption and take the advantage of speed improvement. In figure (4.4.1) reference current competes with inputs current ( $A_i$ ,  $B_i$  and  $C_{i-1}$ ) and result of that keeps lumped capacitor  $C_p$  either charged or discharged. So if we refer inputs current to  $I_x$  ( $I_y$ )

and reference current to  $1.5I$  ( $2.5I$ ) then the output of figure (4.4.1) generates table (4.4.1) and (4.4.4).



**Figure(4.4.1) Current-mode majority-function basis.**

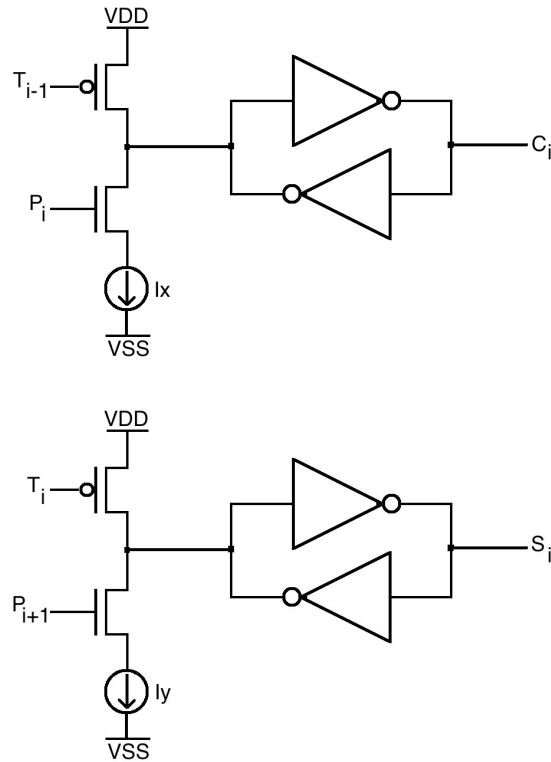
Figure (4.4.1) requires constants and static current and we need different way to avoid this constants current and static power consumption. A capacitor converts current to voltage and save applied power. In sub-threshold where operation current is so small, parasitic capacitor along with node capacitor will be noticeable. Consider “ $C_p$ ” is lumped model of that capacitor then,

$$I_{C/D} = C_p \times \left(\frac{dV}{dt}\right) \quad (4.4.4)$$

Where  $I_{C/D}$  is charge or discharge current and it is a result of all inputs. So if we could charge this capacitor during the initialization and then evaluate inputs in period of “ $t$ ”, which is required time period to discharge that capacitor with reference current, then there wont be that static constants current. Figure (4.4.2) pictures this idea for both tables (4.4.1) and (4.4.4) by simply using  $I_x$  and  $I_y$ . The circuit is initialized by signal(s)  $T_{i-1}(T_i)$  and initializing signal will turn off during the evaluation and evaluating pulse  $PC_i(PS_i)$  with defined period of ‘ $t$ ’ applies to the circuit. If  $I_x$  ( $I_y$ ) was bigger than reference current then in defined period it can discharge capacitor,  $C_p$ , and brings voltage below switching threshold voltage  $V_M$ . So as we can see, not only static current path is closed but also evaluation is limited to  $V_{DD}/2$  or  $V_M$  (when it is considered to be in midpoint). Note that using PMOS/NMOS ratio equal to three pushes  $V_M$  to be lower than  $V_{DD}/2$  and evaluating voltage swing becomes lower.



Low voltage swing was advantage of voltage-mode majority-function by using front-end capacitor network architecture which is applicable in this current-mode architecture as well.

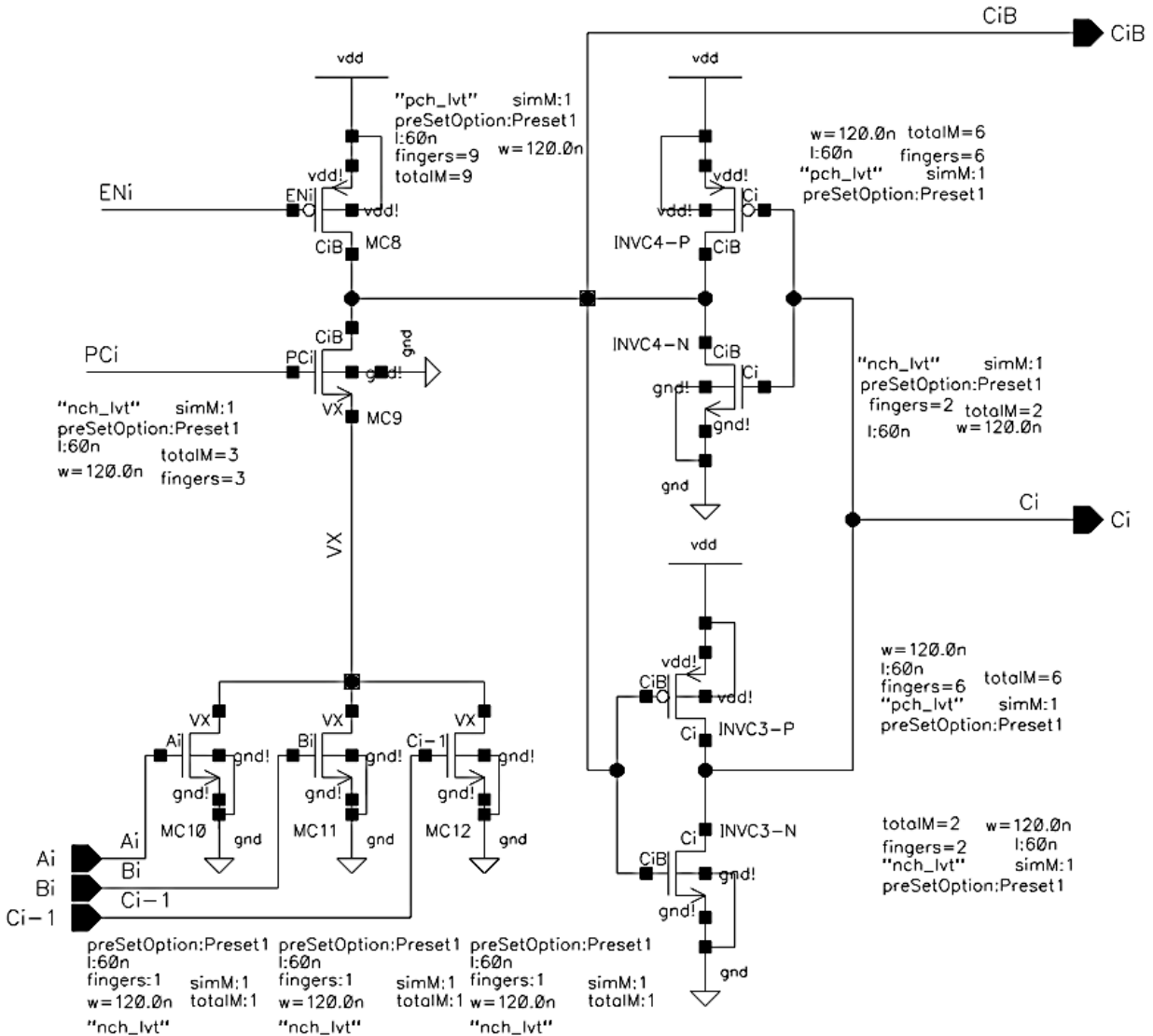


**Figure(4.4.2) Current mode majority function FA concept.**

Hence, we took advantage of both current-mode and voltage-mode in this architecture. As we discussed and have seen it in table (4.4.1) and (4.4.3) both carry and sum follow the identical idea so their architectures are similar with only difference on sizes to maintain  $1.5I$  and  $2.5I$  for carry and sum respectively. Also from table (4.4.3) we saw that sum is a product of carry from same stage so sum has to wait for its carry to calculate first before it evaluates the input signals. More about timing and synchronization will discuss in coming section.

#### 4.4.1 Carry circuit

Figure (4.4.3) shows carry circuit implementing the way it is discussed in figure (4.4.2). At the beginning when circuit is not enabled and enable line ( $EN_i$ ) is low, transistor MC8 is ON and it charges input capacitor of back to back inverter pair INVC3 and 4.

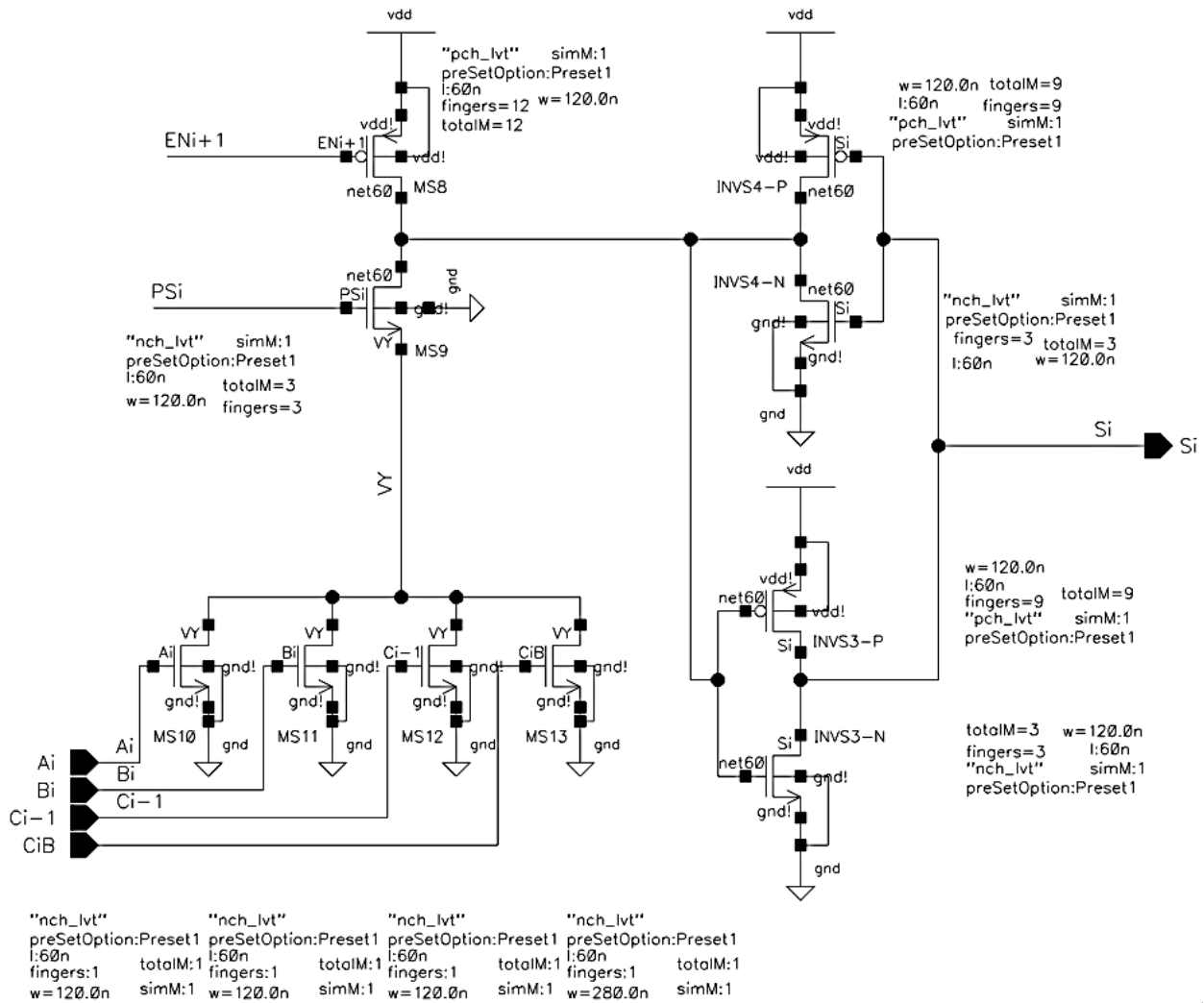


**Figure(4.4.3) Carry circuit in current-mode majority-function FA.**

Note that transistors MC10, MC11 and MC12 are the sources for majority function and are sized equally to provide current of “I”. At the time that enable line (ENi) becomes high, transistor MC8 will turn “OFF” and at the same time MC9 will turn “ON” because of PCi. Input capacitor of back to back inverter pair (INVC3 and 4) discharges through MC9 and input transistors MC10, 11 and 12 based on their ON/OFF conditions during period time of PCi. So PCi either must be generated from ENi or synchronized with that.

## 4.4.2 Sum circuit

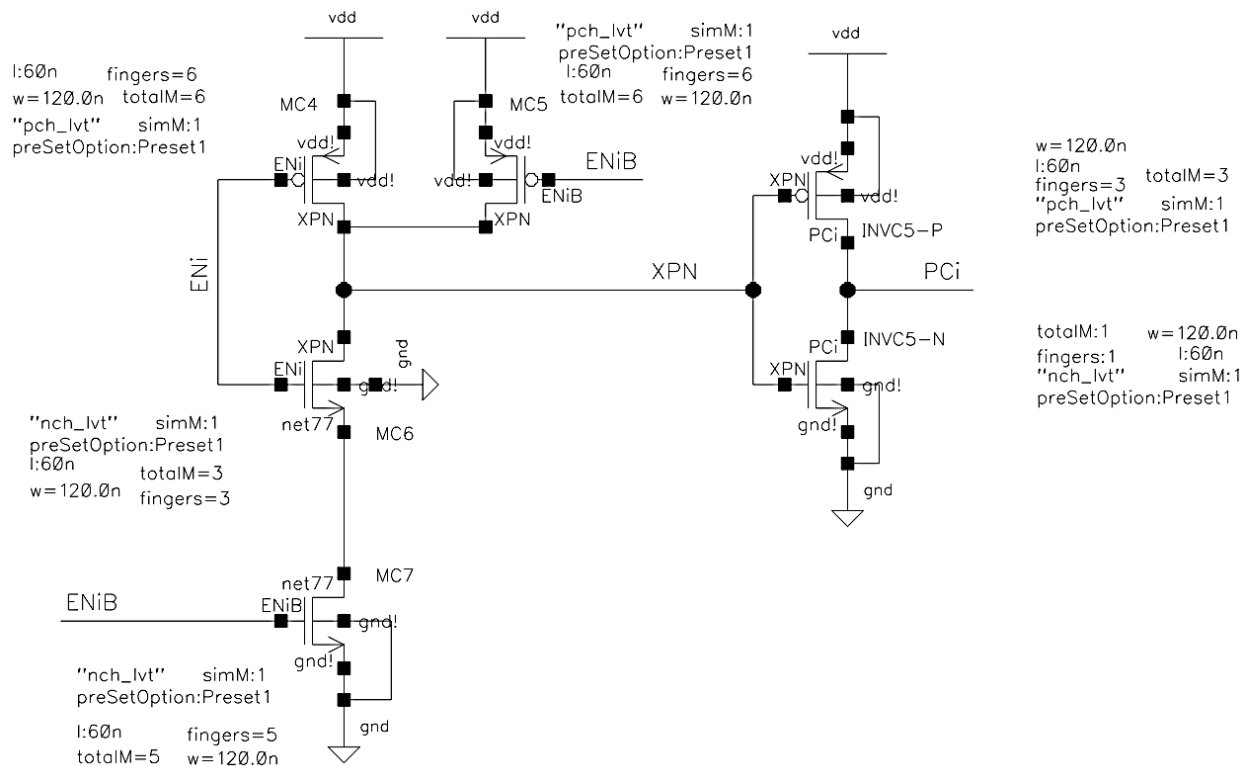
The other advantage of this idea is that sum circuit is similar to carry circuit with only difference in size and an extra input signal. Figure (4.4.4) shows sum circuit implementing. Like carry circuit, transistors MS8 charges input lumped capacitor at initial time, when  $EN_{i+1}$  is low. When  $EN_{i+1}$  goes high, MS8 will turn “OFF” and MS9 turns “ON” as a result of  $PS_i$ . So that lumped input capacitor of back to back inverter pair (INVS3 and 4) discharges through MS9 and MS10, MS11, MS12 and MS13. Like the  $PC_i$ ,  $PS_i$  must also be generated either from  $EN_{i+1}$  or synchronized with that.



Figure(4.4.4) Sum circuit in current-mode majority-function FA.



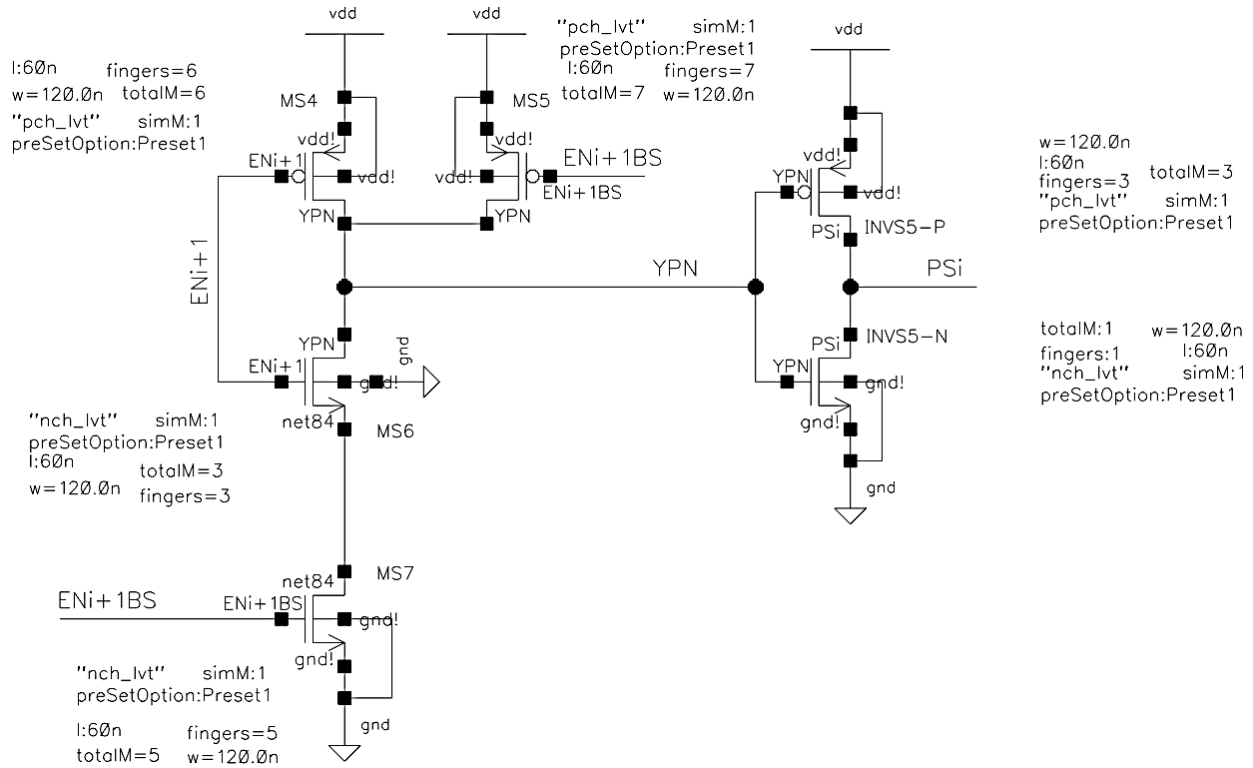
(1.5I). So size of MC9 must be close to 1.5 times bigger than MC10, MC11 or MC12. Following our experiment in section (4.3) all transistors are considered to have minimum “L” and sized “W” to meet the requirements, so  $W_{MC3}$  is almost 1.5 times bigger than  $W_{MC10}$ ,  $W_{MC11}$  and  $W_{MC12}$ . Note that following section (4.3), the source for the minor difference in here from our calculation is because of nonlinearity in transistor current vs size. Like carry circuit,  $PC_i$  is low when  $EN_i$  is low and MC1 is “ON”, so input lumped capacitor similar to carry circuit charges through the MC1. When  $EN_i$  turns “ON” or it goes high, it turns “OFF” MC1 and turns “ON” MC2. MC2 and MC3 provide starving mechanism to discharge input lamp capacitor like what MC9 does with combination of MC10, MC11 and MC12 in carry circuit. MC3 presents reference current and generates discharge period or delay time equal to reference current so that  $EN_{i+1}$  will go high after this delay. This delay time is maximum required time for evaluation on each stage and must be realized on each stage.



**Figure(4.4.6) The AND gate which is been used in the PCi circuit.**



and next stage and sum circuit can use  $C_i$  signal as their input signal. Because of that  $PS_i$  uses  $EN_{i+1}$  to generate its pulse,  $PS_i$ . Similar to  $PC_i$  these signals  $EN_{i+1}$  and  $EN_{i+1}B$  which refers to the starving node in figure (4.4.7) are applied to an AND gate which is pictured in figure (4.4.8). Result of the AND gate is  $PS_i$  with pulse width that presents reference current,  $I_y$ .

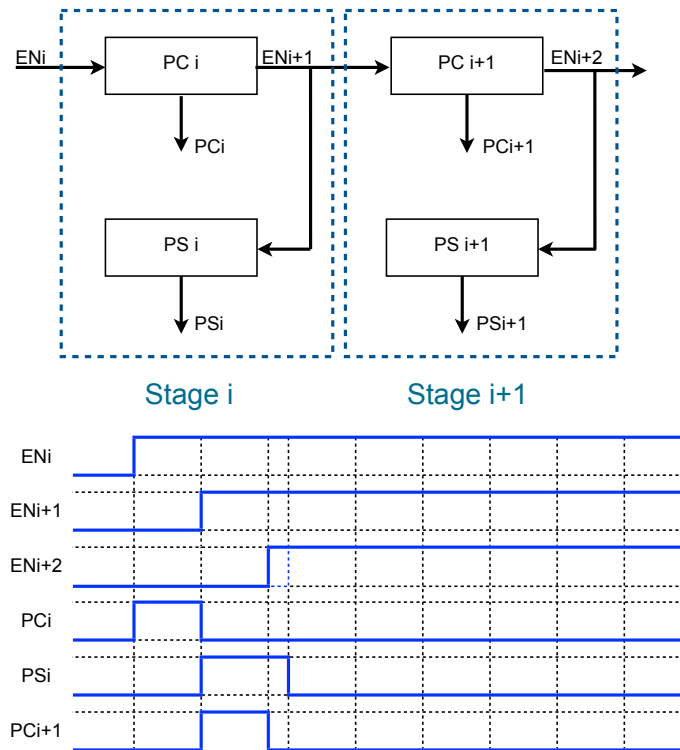


**Figure(4.4.8) AND gate which is been used in the PCi circuit.**

So all required elements for a single bit full adder based on current-mode majority-function are presented. Before we look at test circuit and result on a single bit, it is important to take a look at synchronization and review pipeline architecture which has been used in this design.

## 4.5. Self-Timed Circuit implementing

In chapter two we discussed advantages of using pipeline and self-timed circuits. Proposed current-mode majority-function full adder uses pipeline architecture with self-timed circuit. In table (4.4.3), we saw sum is function of  $A_i$  and  $B_i$ , input signals to each stage,  $C_{i-1}$  from previous stage and  $C_i$  which is carry result of the same stage. In all carry ripple adder circuit we have to wait for  $C_{i-1}$ , so using pipeline architecture can help to save energy which is consumed in the format of a glitch. This glitch propagates from first to last stage in group one (“a”) of full adder (Carry propagate group in chapter 3). Therefore pipeline can prioritize each stage to avoid generating a glitch. We also saw advantage of asynchronous architecture in chapter two which has been used in this design and Figure (4.5.1) pictures asynchronous pipeline implementing in this design.



**Figure(4.5.1) Asynchronous pipeline pulse generating in CMMF.**

At the beginning  $EN_i$  signal is “Low” hence all other stages are “Low”. When  $EN_i$  goes “High” and after period of  $t_{pc}$ , which based on reference current,  $I_x$ , as it was discussed in section (4.4.3),



$EN_{i+1}$  becomes “High”.  $PC_i$  or applied pulse with period of  $t_{pc}$  to carry circuit, is a result of  $EN_i$  and  $EN_{i+1}$ . In the next stage, carry pulse generator enables when  $EN_{i+1}$  becomes “High” and so. Note that, pulse for sum circuit must be presented when  $C_i$  is ready, therefore  $EN_{i+1}$  is considered to generate  $PS_i$  with period of  $t_{ps}$ .  $PS_i$  circuit follows the same idea that has been used for  $PC_i$ .  $PS_i$  and  $PC_{i+1}$  both are synced and generated from  $EN_{i+1}$  which refers to falling edge of  $PC_i$  and that means carry evaluation in last stage is completed and carry signal is valid. Because of different required pulse widths for carry and sum, two separate pulse generator has been used in this design. If we consider  $t_{pc} = t_{ps}$ , then we could use  $PC_{i+1}$  instead of  $PS_i$  to make circuit smaller and use less transistors. However this requires to have size ratio of 1.7 between back to back inverters (or (input lumped capacitor) in carry circuit and sum circuit. Note that in sub-threshold size of transistors are big to maintain minimum current and considering back to back inverters larger to achieve 1.7 time bigger lump capacitor leads to enlarge entire sum circuit as we discussed it earlier (i.e. Impact of parasitic capacitors).

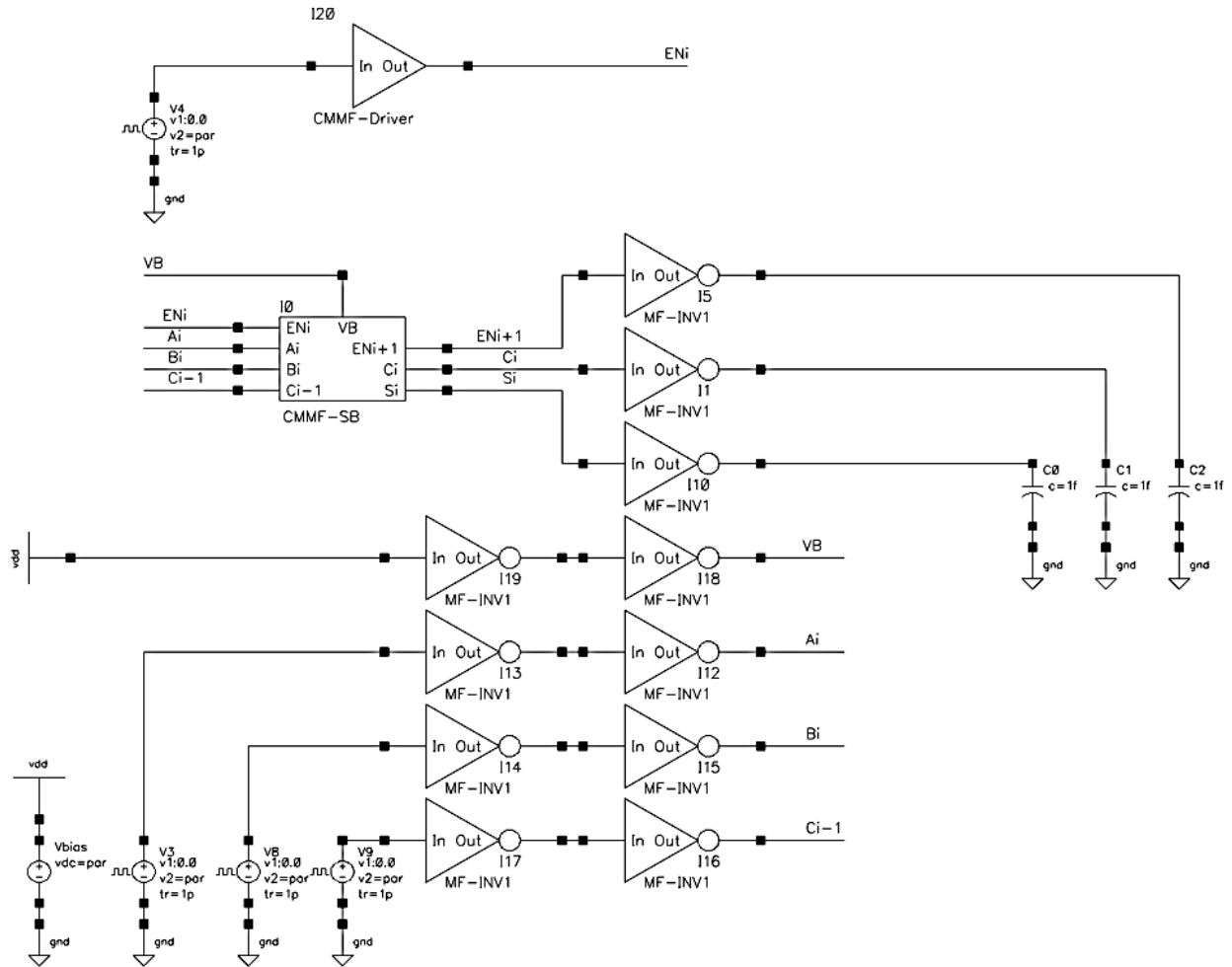
We could also generate two different pulse PC and PS at the beginning and then pass them through each stage with delay equal to pulse width of PC. That approach uses more transistors where in this architecture asynchronous pipeline pulse generators are merged in evaluating circuit to use less transistors and consume less power. Also at 75 mV, required propagation time or the delay to evaluate input signals was measured and determined to be around 10  $\mu$ s. Thereby,  $EN_{i+1}$  goes “High” right 10  $\mu$ s after  $EN_i$  goes “High” from “Low” or that means 10  $\mu$ s requires to evaluate input signals at 75 mV.

## 4.6. Circuit Simulation and Analysis

Designed architecture for CMMF full adder is an independent circuit. This means carry is generated in each stage and also has been used internally to generate sum result. Therefor carry and sum outputs of each stage are independent from next and other stages, so all applicable tests in single bit full adder circuit, conceptually are valid in structural N-bit circuit.

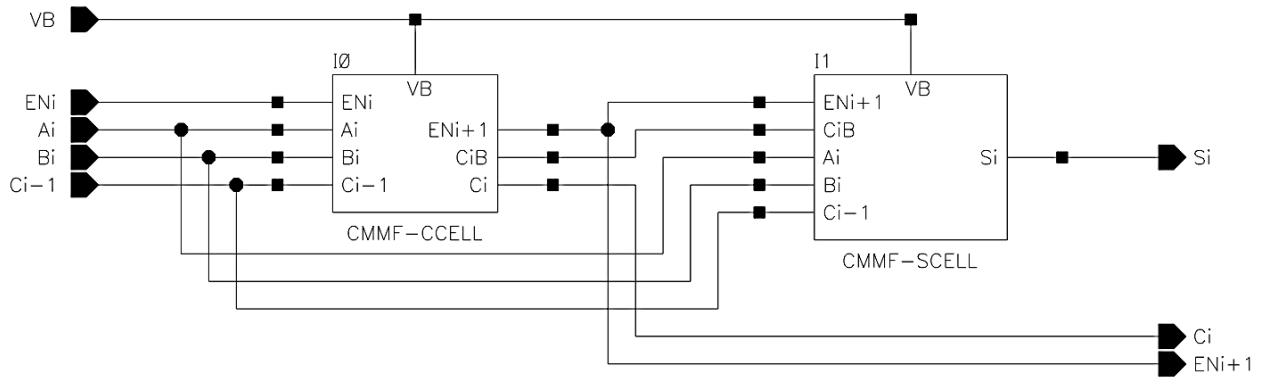
### 4.6.1. A Single Bit CMMF-FA Simulation

In this test all inputs and outputs to or from the CMMF-FA single bit unit, are driven by identical line buffer/Inverter. Figure (4.6.1) presents applied tests to the CMMF full adder circuit and its interaction with I/O units. Three major elements in test circuit are the “CMMF-SB”, “MF-INV1” and “CMMF-Driver” units.



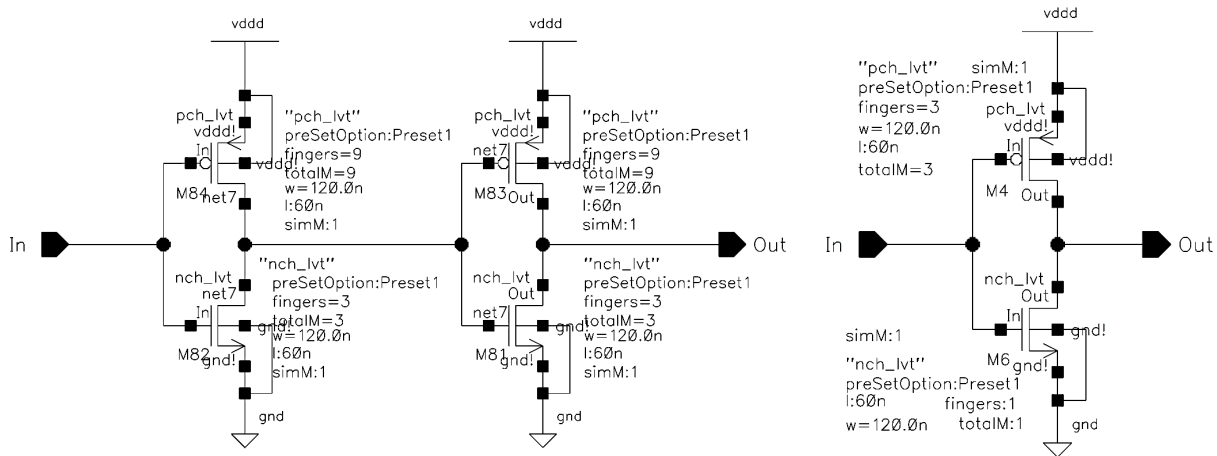
Figure(4.6.1) Single Bit CMMF Full Adder Test Circuit.

Figure (4.6.2) pictures detail of CMMF-SB circuit. This is a main unit which will use to provide an 64-bit adder circuit in later tests. Input signals can connect directly to pulse generators but buffer and driver circuits are used to provide realistic condition and consider any impact result of drivers and buffers on final results.



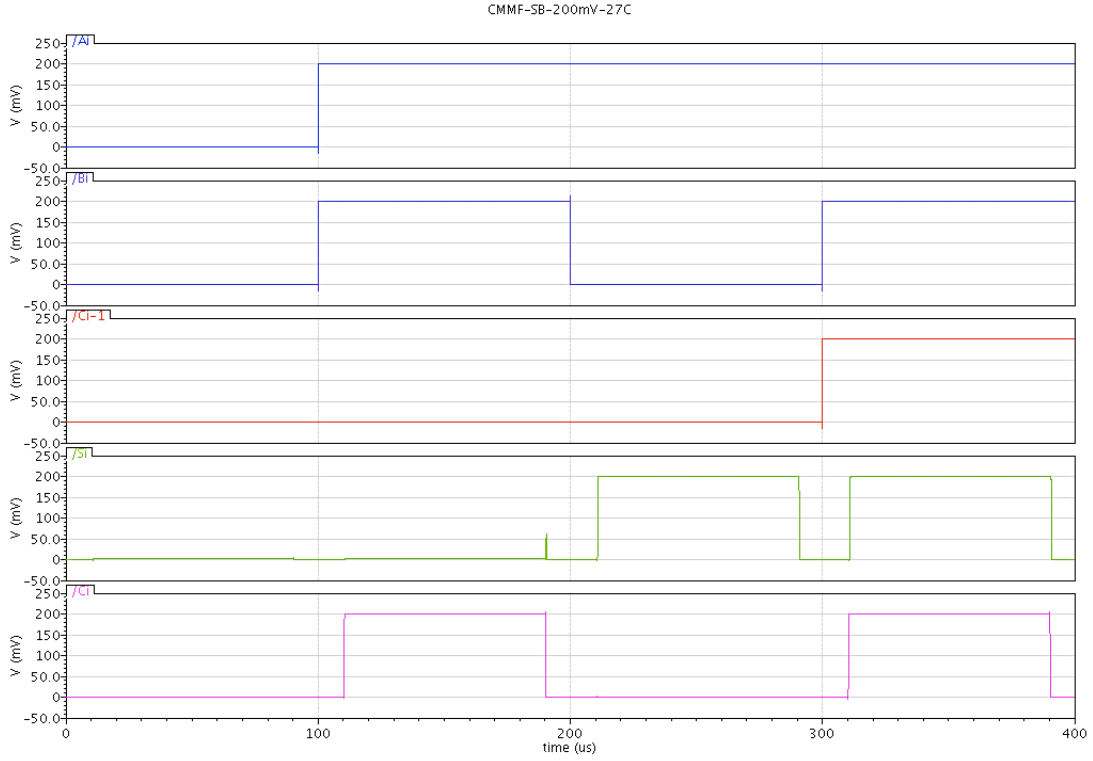
**Figure(4.6.2) A Single Bit CMMF Full Adder Circuit.**

Figure (4.6.3) shows “CMMF-Driver” and “MF-INV1” circuits in detail. These units also will be used in 64-bit CMMF adder test circuit.

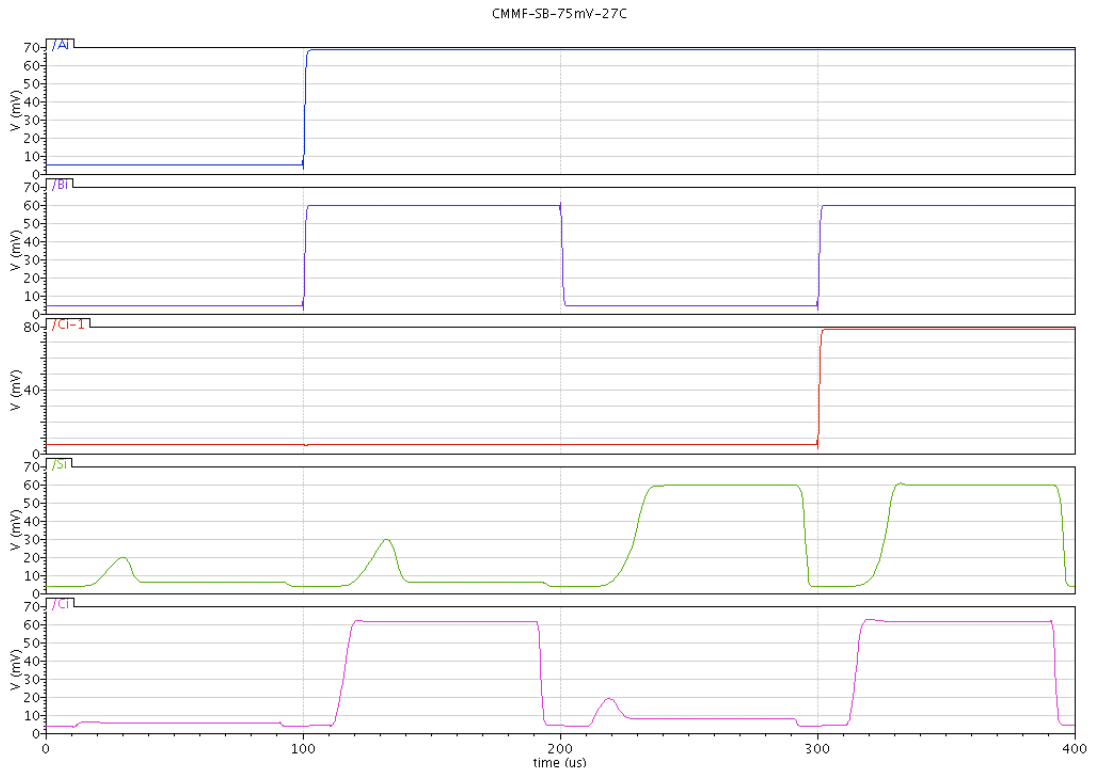


**Figure(4.6.3) CMMF-Driver circuit (Left) and MF-INV1 Circuit (Right).**

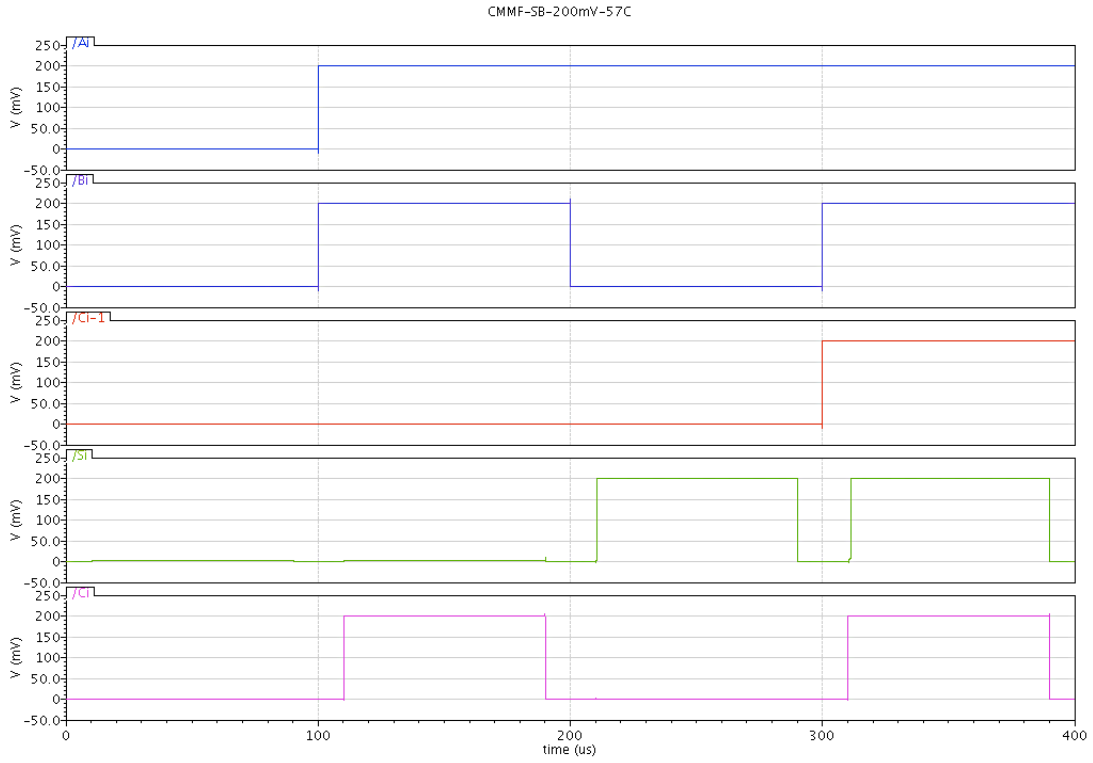
$A_i$ ,  $B_i$  and  $C_{i-1}$  are applied input signals along with  $EN_i$ . They generate all conditions in look-up table of full adder. Note that evaluation begins right after enable signal applied. First test is to operate circuit at 200 mV and in 27°C. Results are pictured in figure (4.6.4). Then voltage gradually dropped to reach to the minimum applicable voltage at the same temperature. Figure (4.6.5) shows the test results at 75 mV and 27°C.



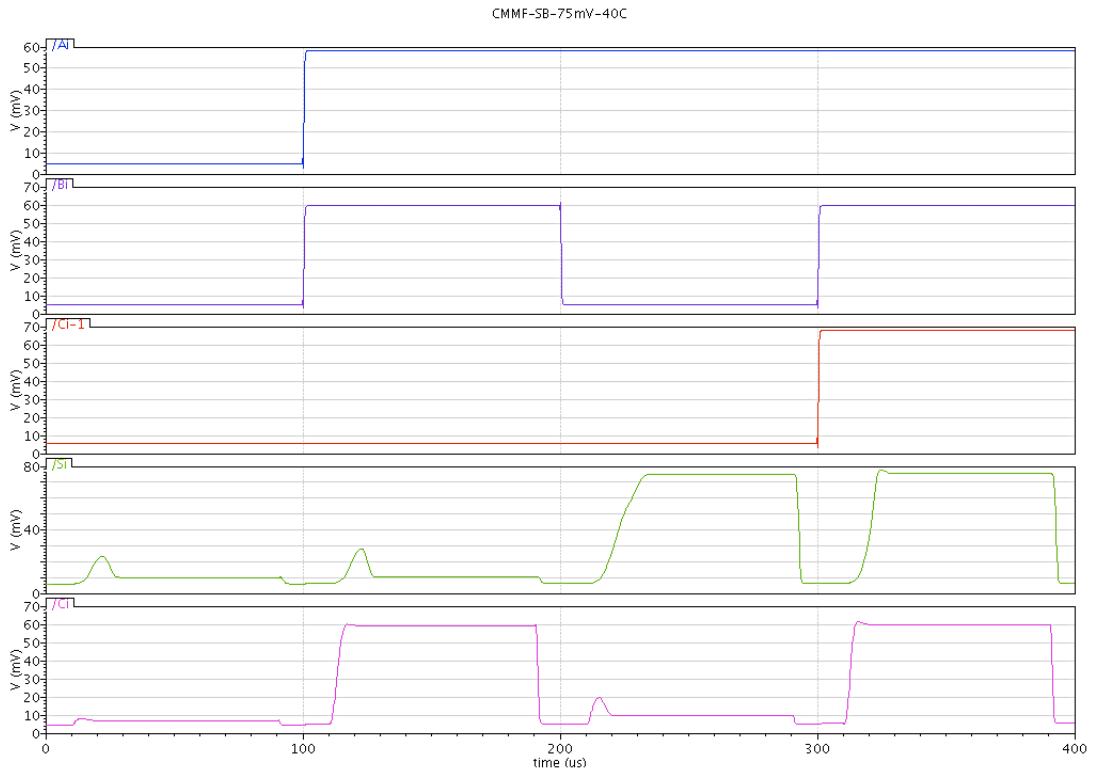
**Figure(4.6.4) A Single Bit CMMF-FA Simulation at 200mV/27°C.**



**Figure(4.6.5) A Single Bit CMMF-FA Simulation at 75mV/27°C.**



**Figure(4.6.6) A Single Bit CMMF-FA Simulation at 200mV/57°C.**



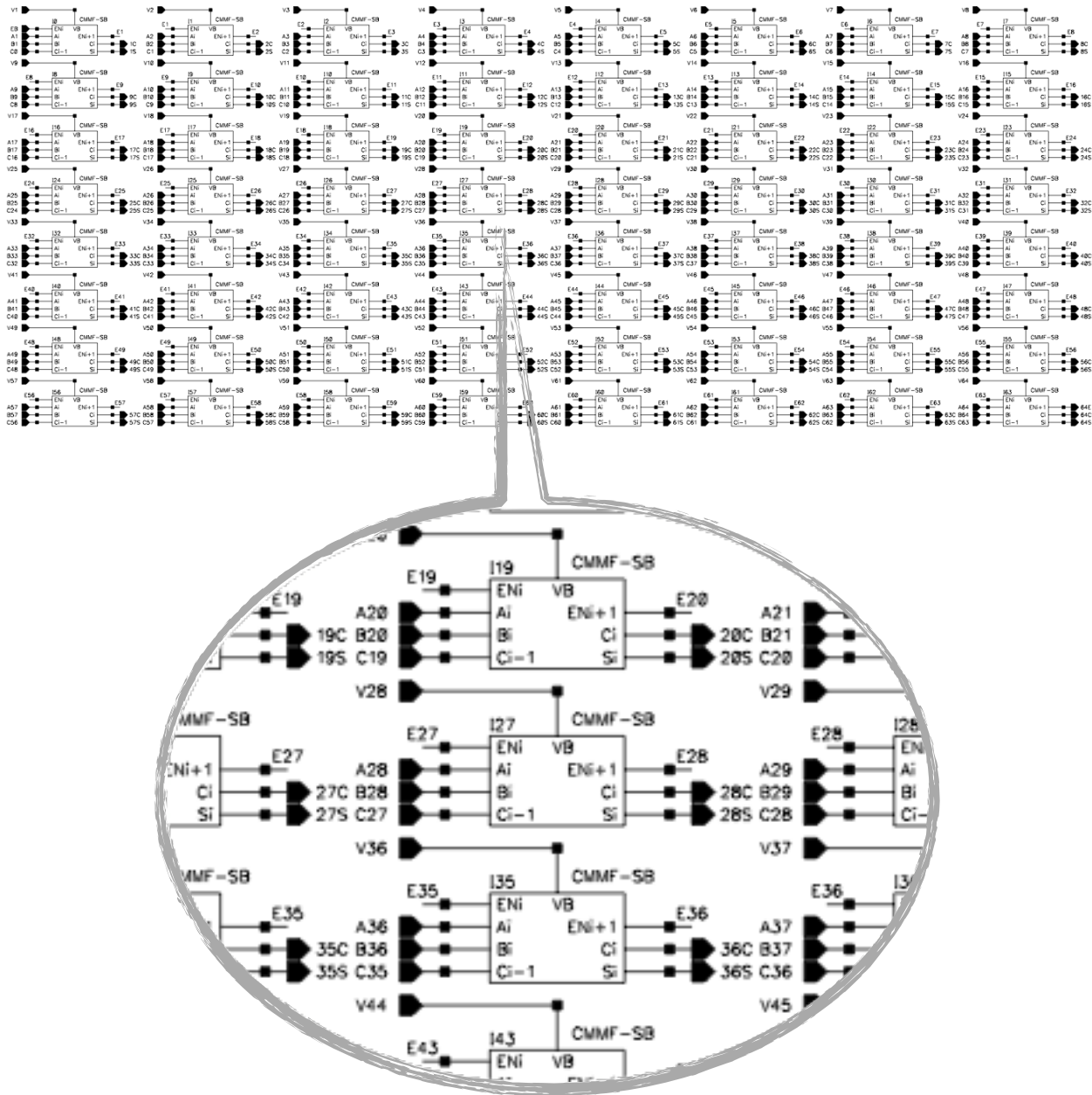
**Figure(4.6.7) A Single Bit CMMF-FA Simulation at 75mV/40°C.**

Test results also show the impact of voltage variation and minimum applicable voltage at room temperature, 27°C. The minimum applicable voltage of 75 mV was recorded and confirmed. Temperature variation at minimum and maximum applicable voltage (i.e 200 mV and 75 mV) also tried to confirm maximum operating temperatures in each case. Figures (4.6.6) and (4.6.7) present test results at 200 mV and 57°C followed by 75 mV and 40°C. Comparing figure (4.6.7) and figure (4.6.5) shows that speed of circuit has been improved when temperature was raised. That was expected and discussed earlier in section (2.3.1). Unlike strong inversion where lower mobility dominates in high temperatures and slows circuits, in sub-threshold region, a lower  $V_{th}$  dominates in high temperatures and results in a lower delay. Following equation (2.3.7), Temperature has an exponential effect on the mobility in sub-threshold and that causes nonlinearity impact on the circuit. This condition must be considered in any circuit that required wide operating temperature and it is based on timing.

Above tests confirm operation of single bit CMMF full adder architecture. Most architectures pass single bit tests but problem happens when they are used to generate higher bits. There are many reason for that but the major ones are significant voltage and speed drop. Hence to confirm the single bit CMMF full adder cell design, it must be used in adder with large number of bits.

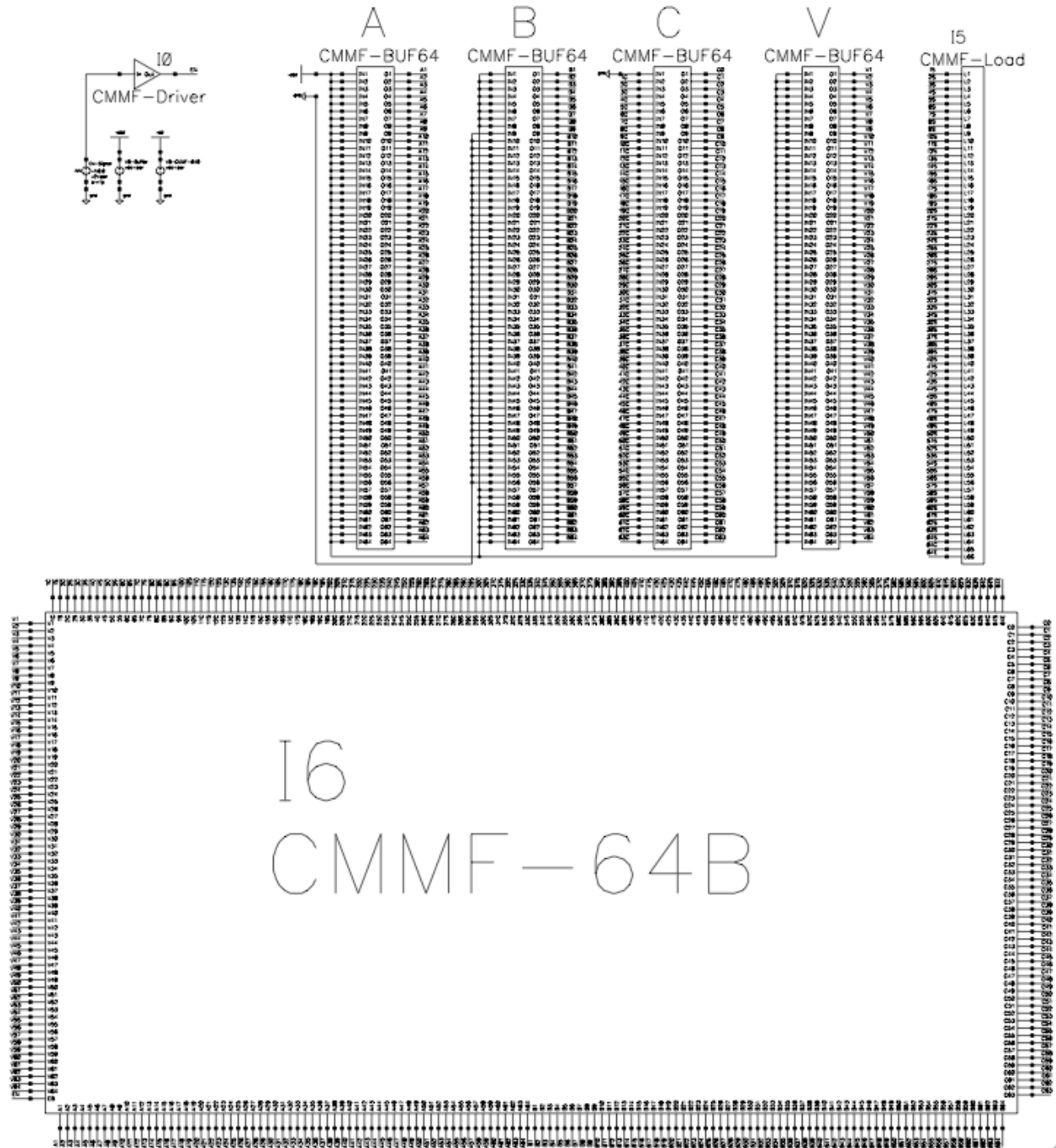
## 4.7. A 64-Bit Pipeline CMMF-FA Test and Simulation

A 64-bit array of the single bit CMMF full adder that was tested in previous section are put together to generate a 64-Bit Pipeline CMMF adder. Single bit units are pipelined and connected as they are shown in figure (4.7.1) to perform an 64-Bit adder.



Figure(4.7.1) A 64 Bits CMMF-FA circuit.

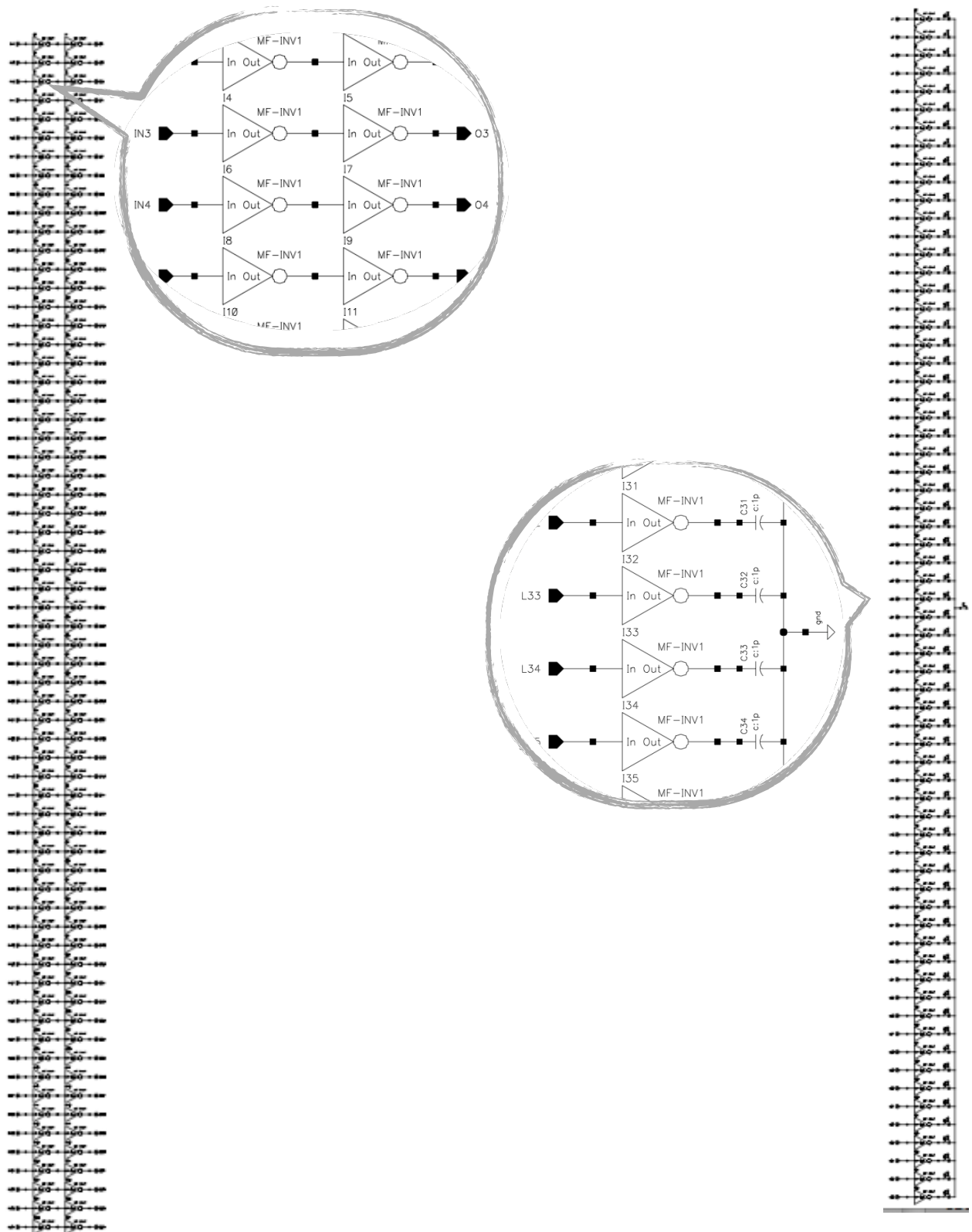
Generated 64-Bit CMMF adder, then is used in test circuit which is presented in figure (4.7.2) to add two numbers, “FFFF,FFFF...FFFF,FFFF” and FFFF,0000...0000,FFFF”. These two numbers are selected because they generate longest carry propagation and consuming almost maximum power because of sum result.



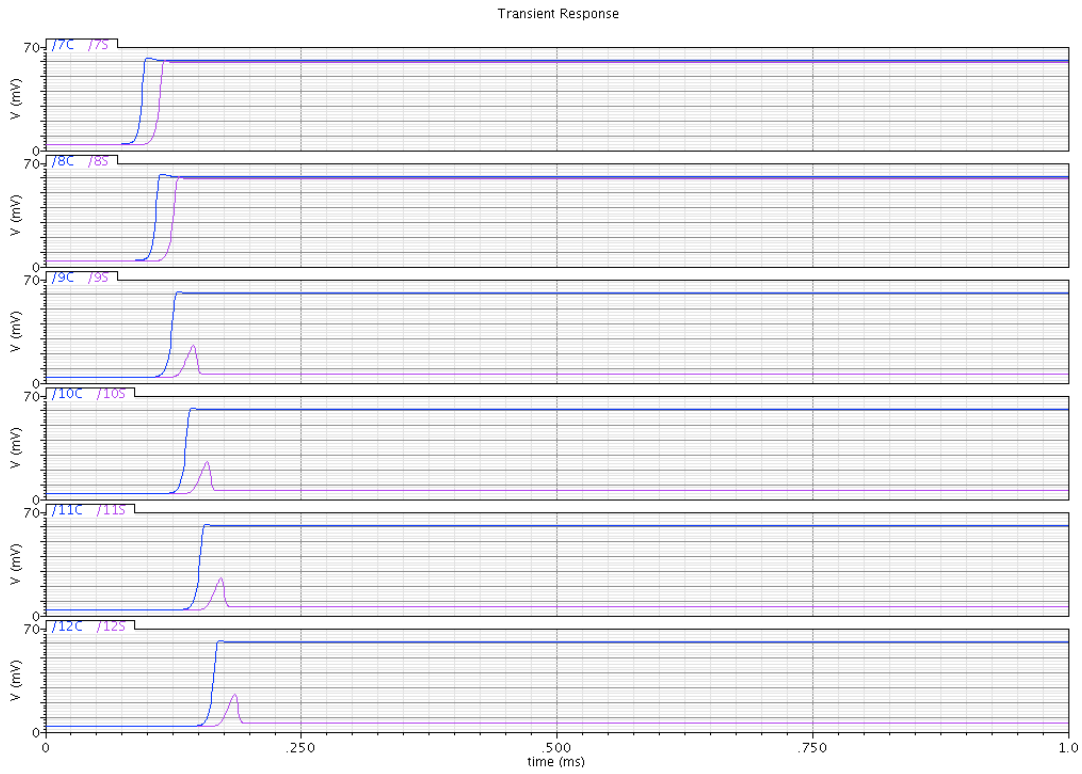
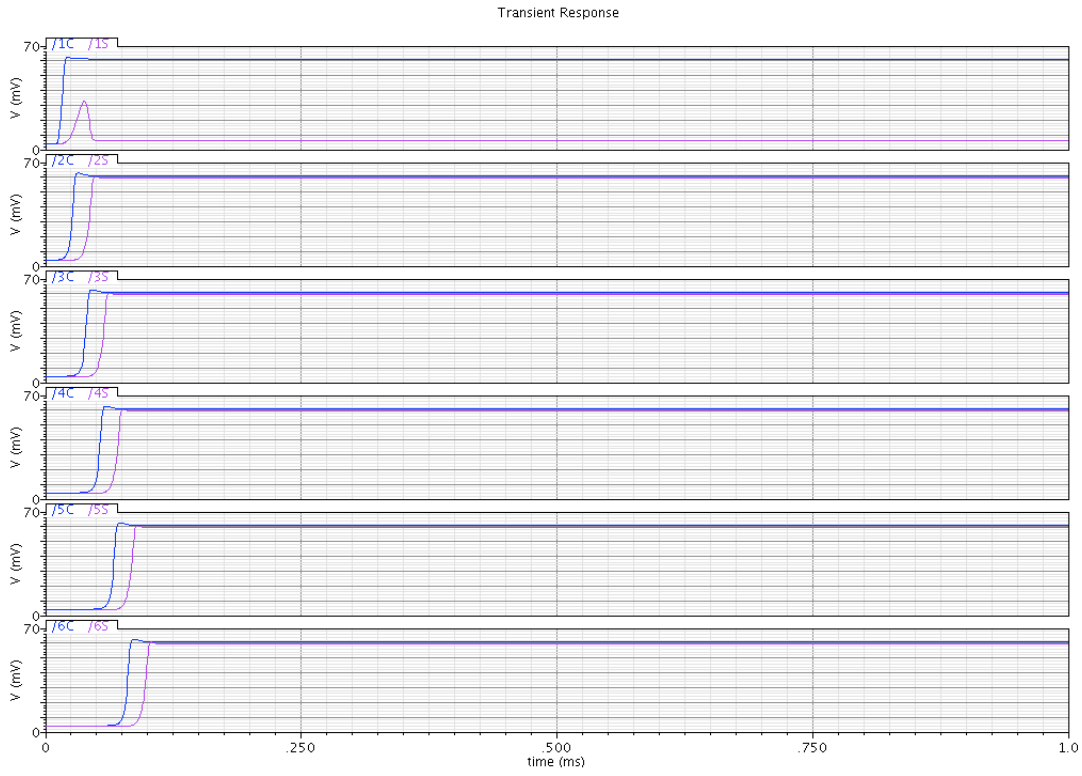
Figure(4.7.2) A 64 Bits CMMF-Adder Test circuit.



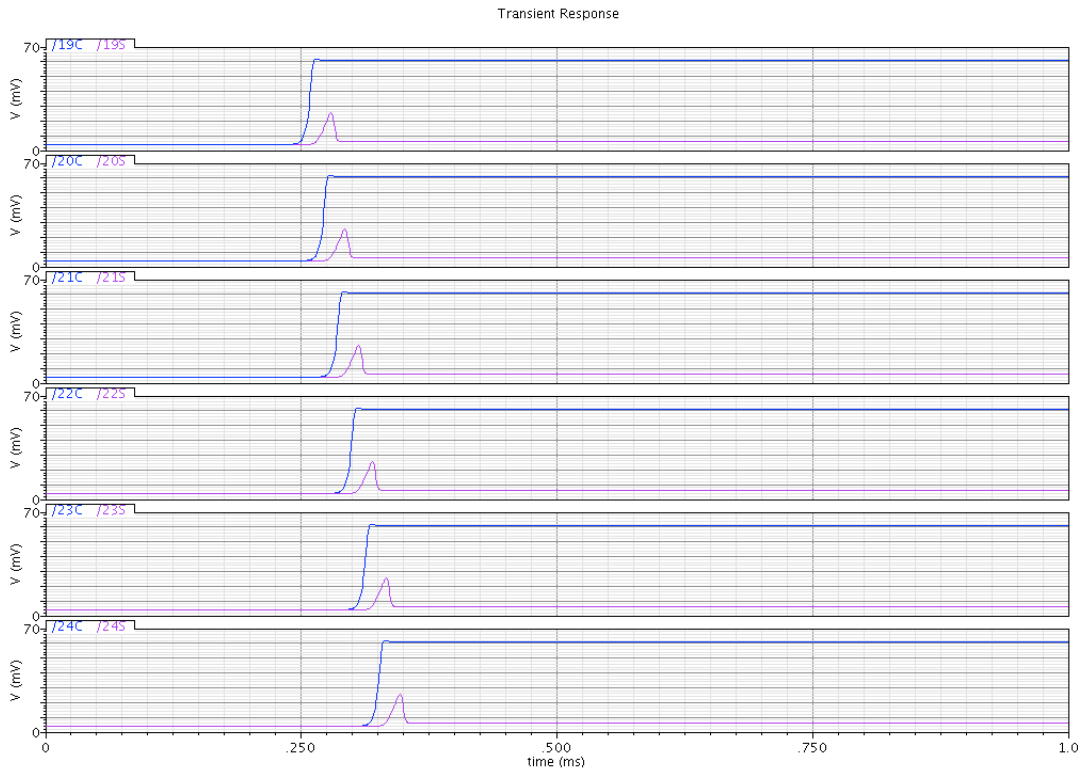
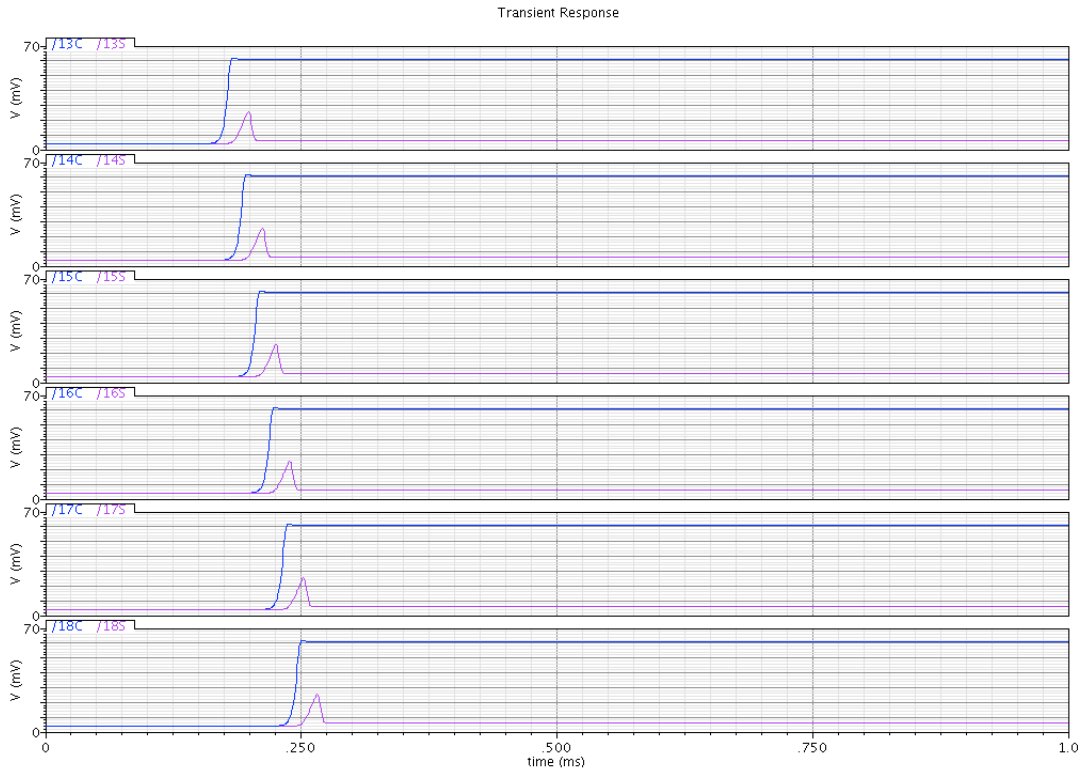
Same concept to use buffers and drivers to provide realistic conditions and consider any impact from drivers and buffers on final results that we considered in a single bit CMMF full adder test has been taken in this test. The “CMMF-BUF64” and “CMMF-Load” circuits in detail are shown in figure (4.7.3). Both circuits use “CMMF-Driver” circuit which was already shown in figure (4.6.3). Last element in this test is “CMMF-Driver” which was shown in figure (4.6.3) too. Carry and sum result of all bits are pictured in transient mode to present signals conditions in each stage. All carry and sum results are pictured in figure (4.7.4) to (4.7.9). The total operation time for a 64 bits adding at 75 mV applied voltage and 27°C is captured as 860  $\mu$ s. The power and energy consumption in this operation was 4.5 nW or 3.8 pJ. Figure (4.7.10) shows current consumption before and during the operation. Current consumption is raised almost linearly as the stages are enabled one by one and the delta current was about 20 nA, from first to the last stage. Recorded power consumption in this test is close to maximum power consumption for this circuit. This is because dynamic power consumption is result of two major elements. First, evaluating and second, is level changing. It has been discussed earlier that evaluating uses minimum possible power which is based on  $V_M < V_{DD}/2$  where  $V_M$  is the switching threshold voltage of inverters. So regardless to the inputs and output result this much power has to be taken to evaluate any input signal. The second part however, it consumes more than the half of voltage which was used to evaluate and it is used to flip back to back inverters output. So whenever carry or sum outputs becomes high, the second power consumption element has been used. In this test all carry and 25% of sum outputs became high.



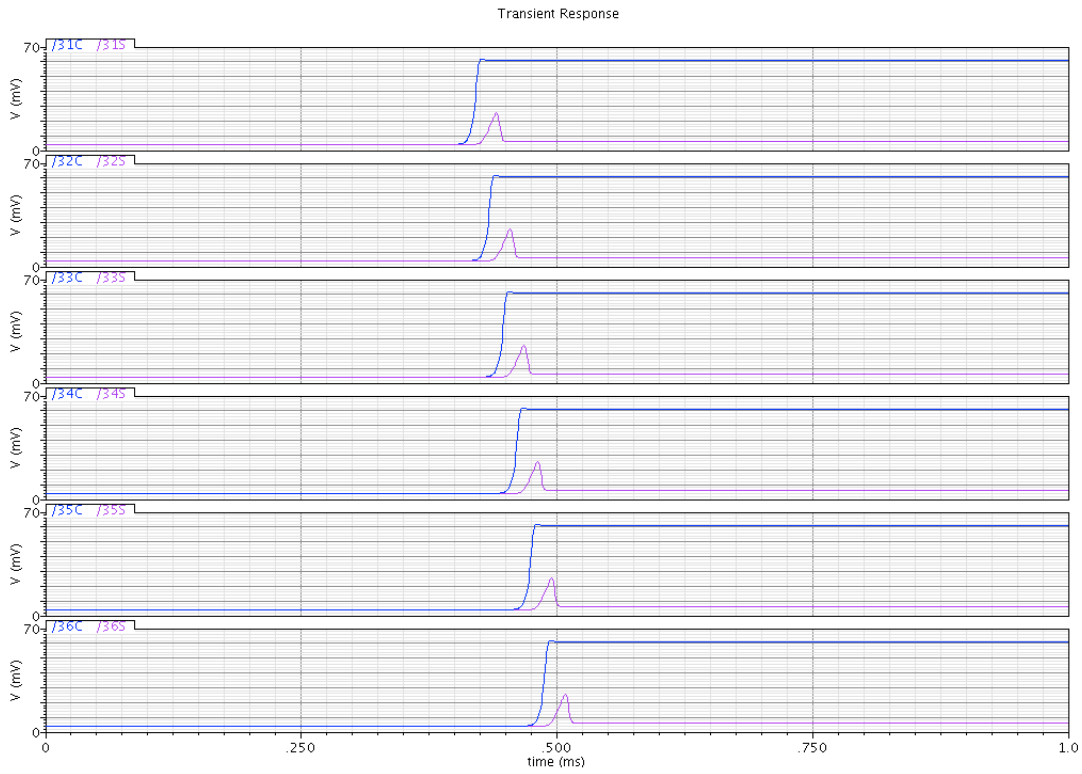
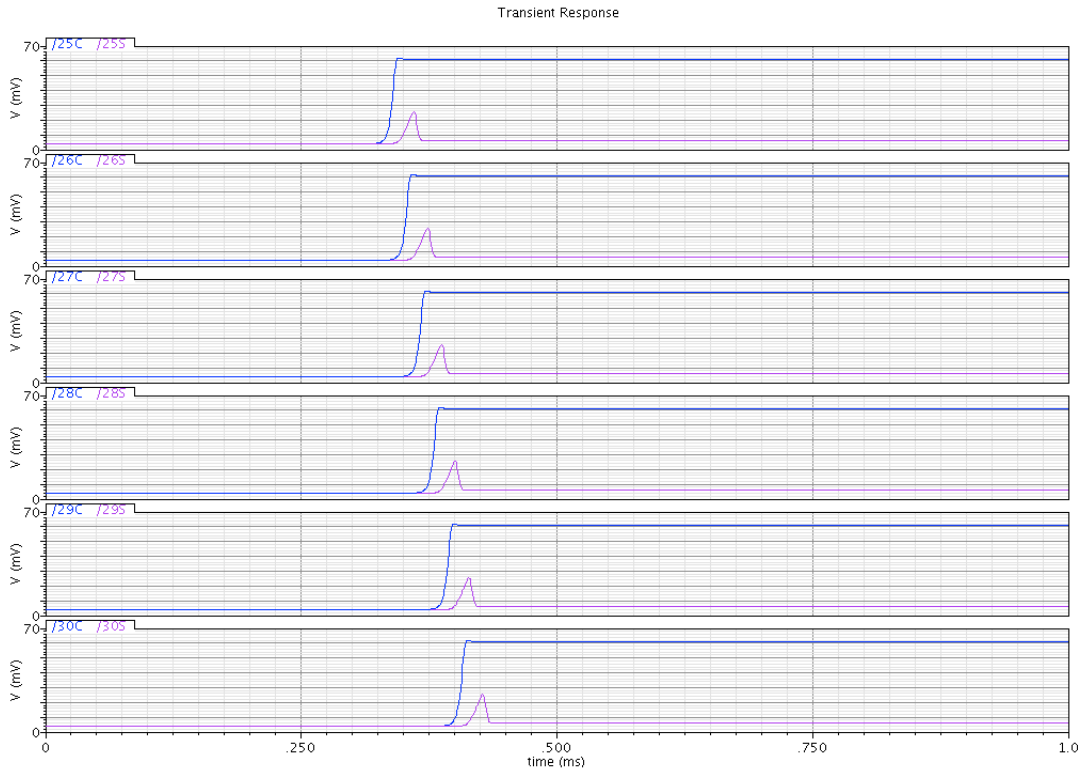
Figure(4.7.3) CMMF-BUF64 and CMMF-Load circuits.



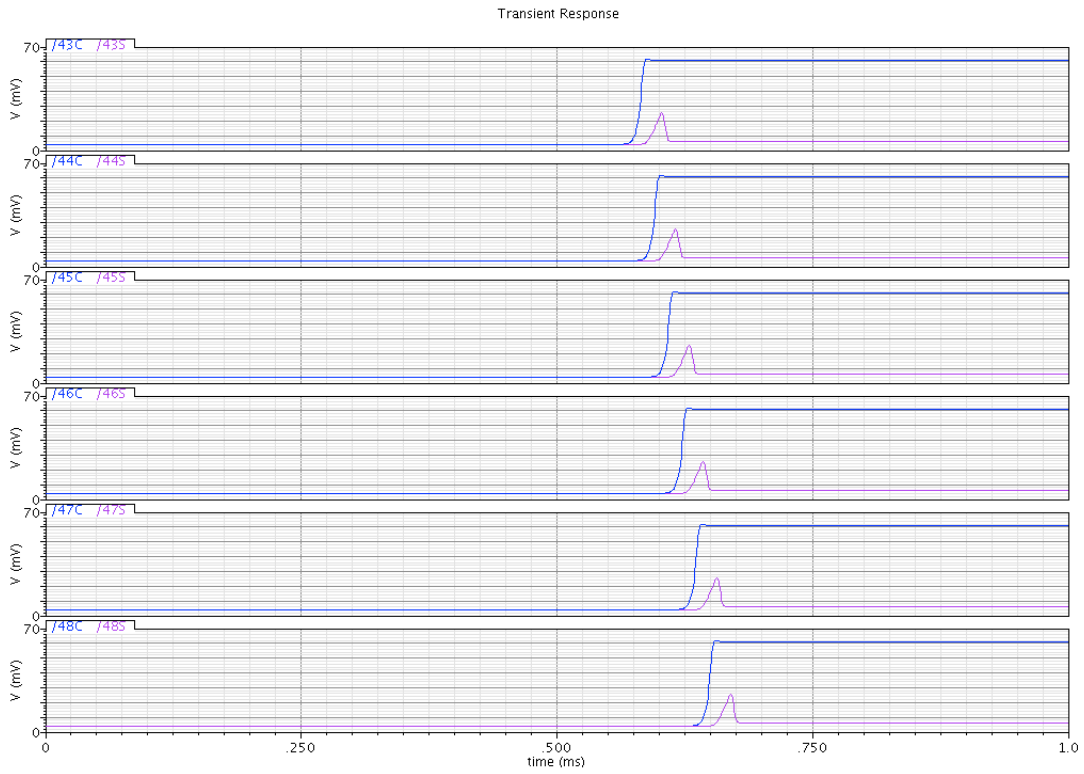
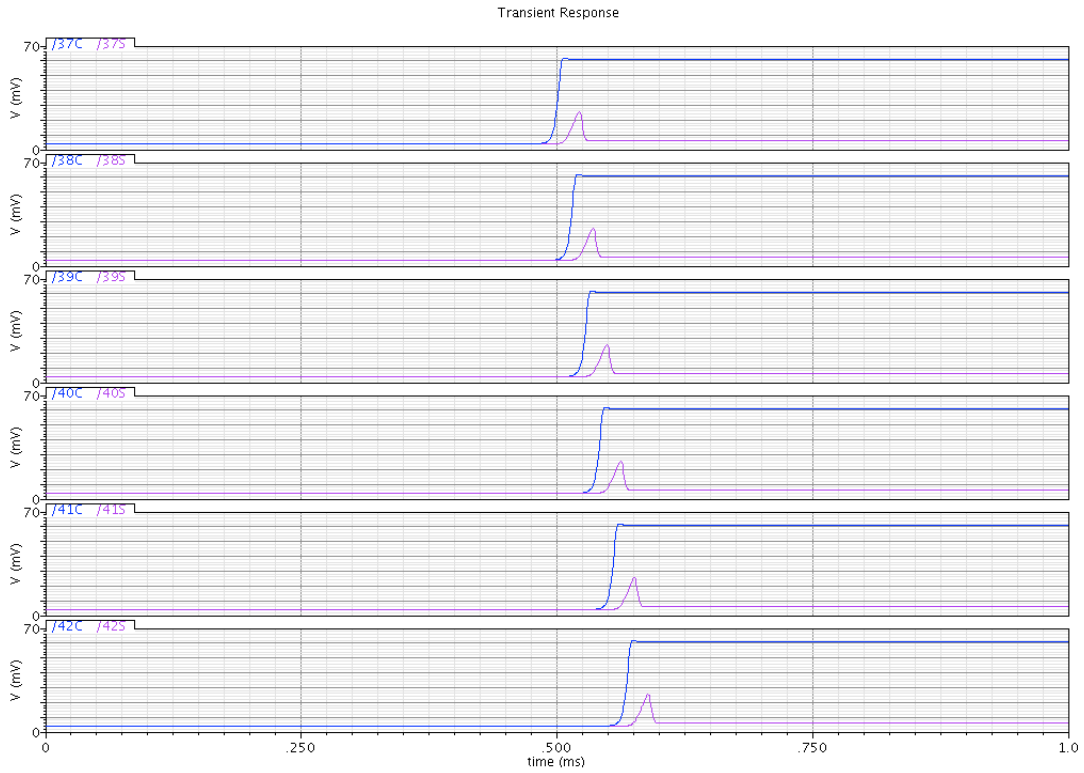
**Figure(4.7.4) Test Results of Bits 1 to 12.**



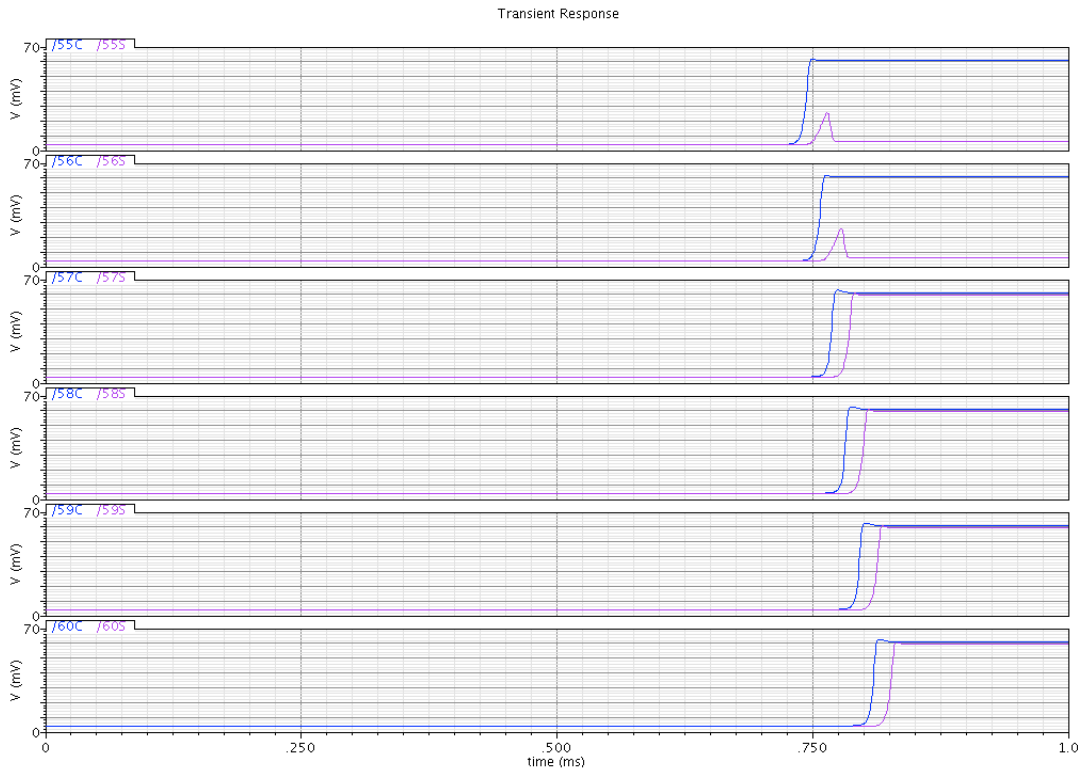
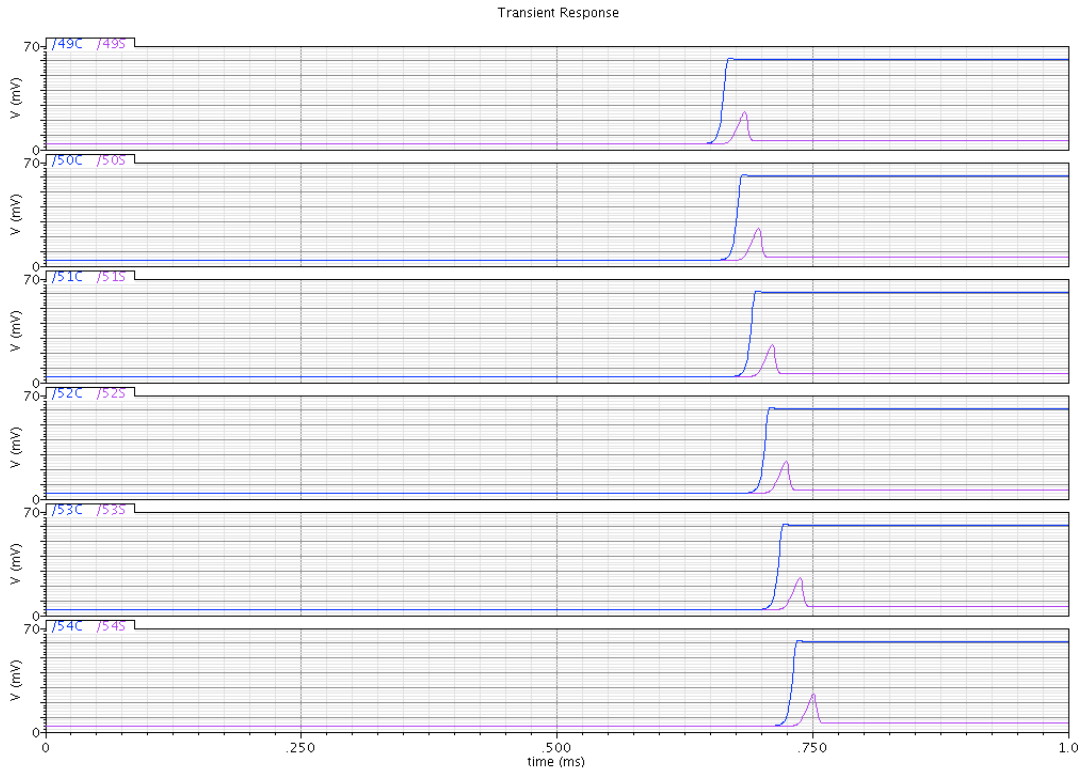
**Figure(4.7.5) Test Results of Bits 13 to 24.**



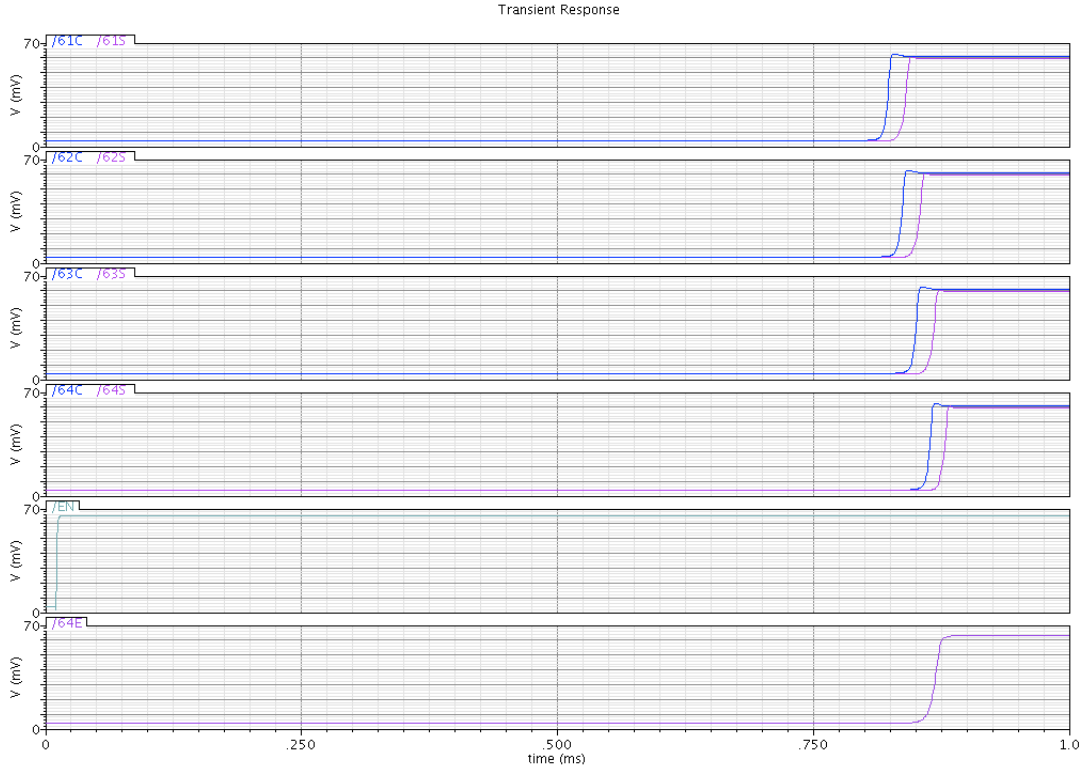
**Figure(4.7.6) Test Results of Bits 25 to 36.**



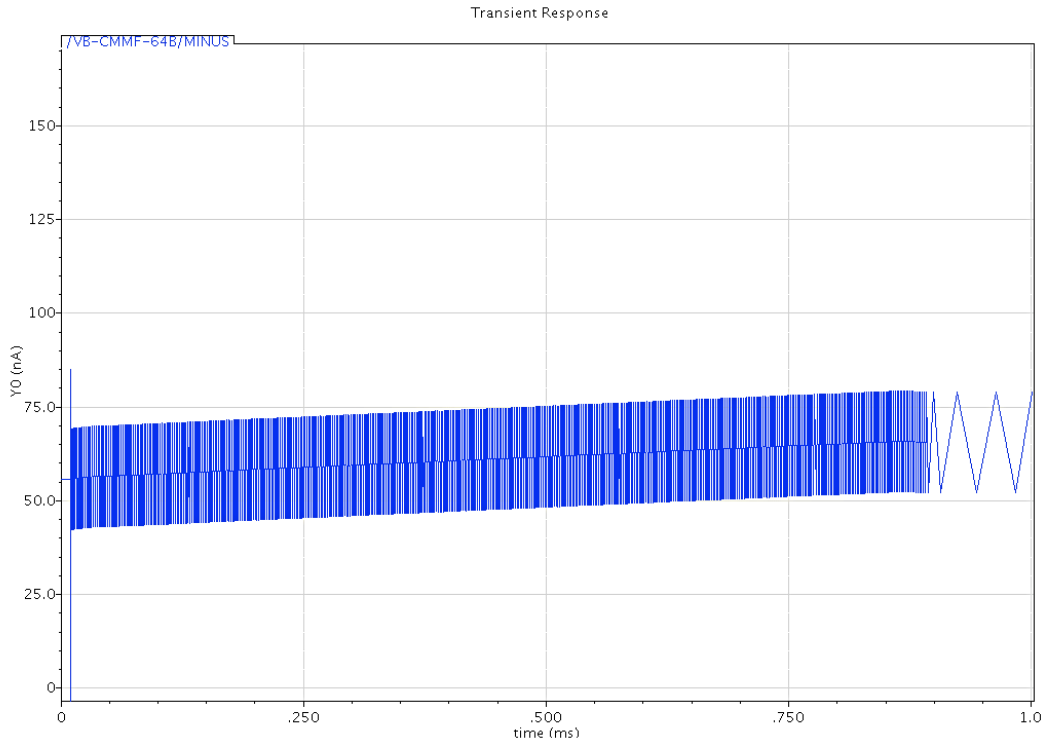
**Figure(4.7.7) Test Results of Bits 37 to 48.**



**Figure(4.7.8) Test Results of Bits 49 to 60.**



**Figure(4.7.9) Test Results of Bit 61 to 64.**

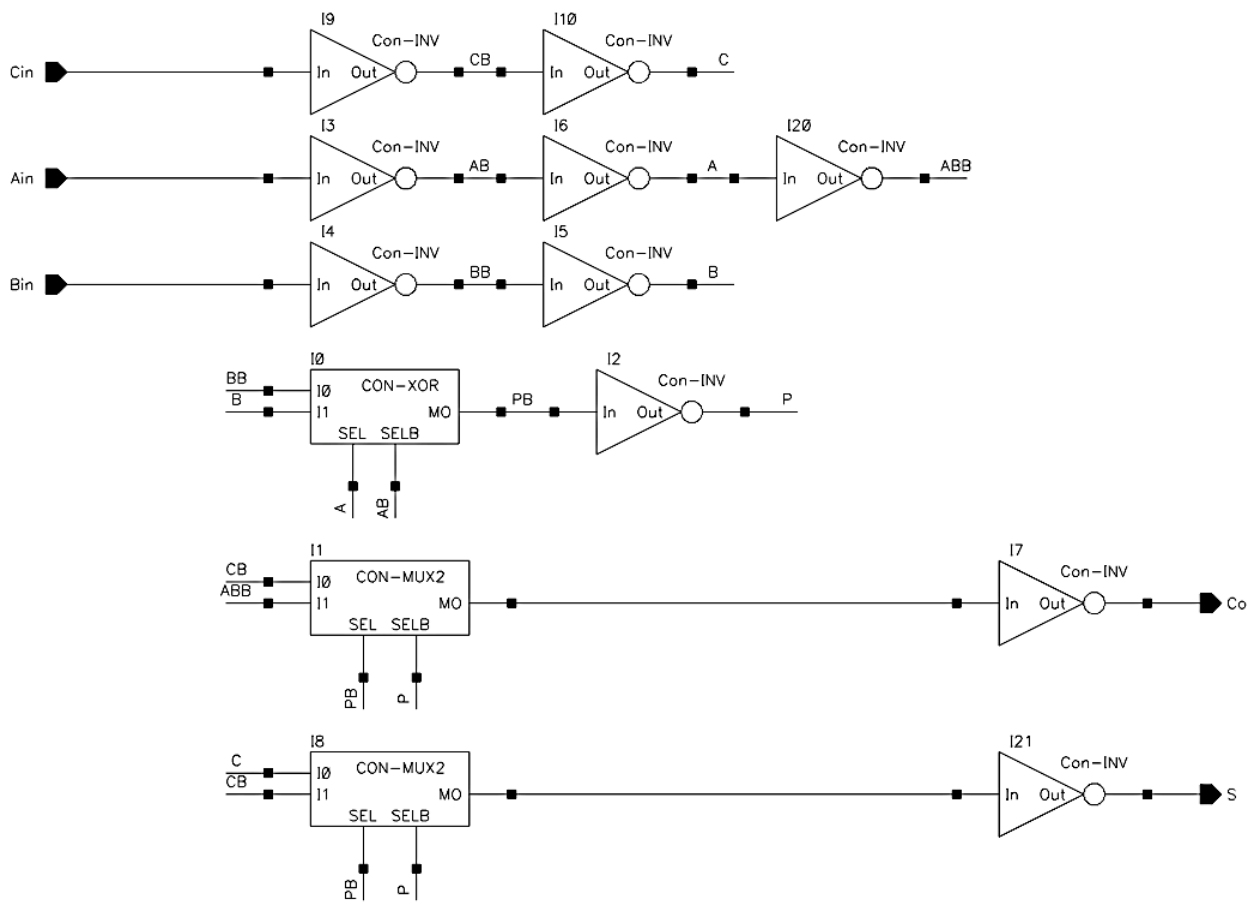


**Figure(4.7.10) Current consumption in 64 Bit adding operation.**



## 4.8. Conventional Full Adder Test with 75mV

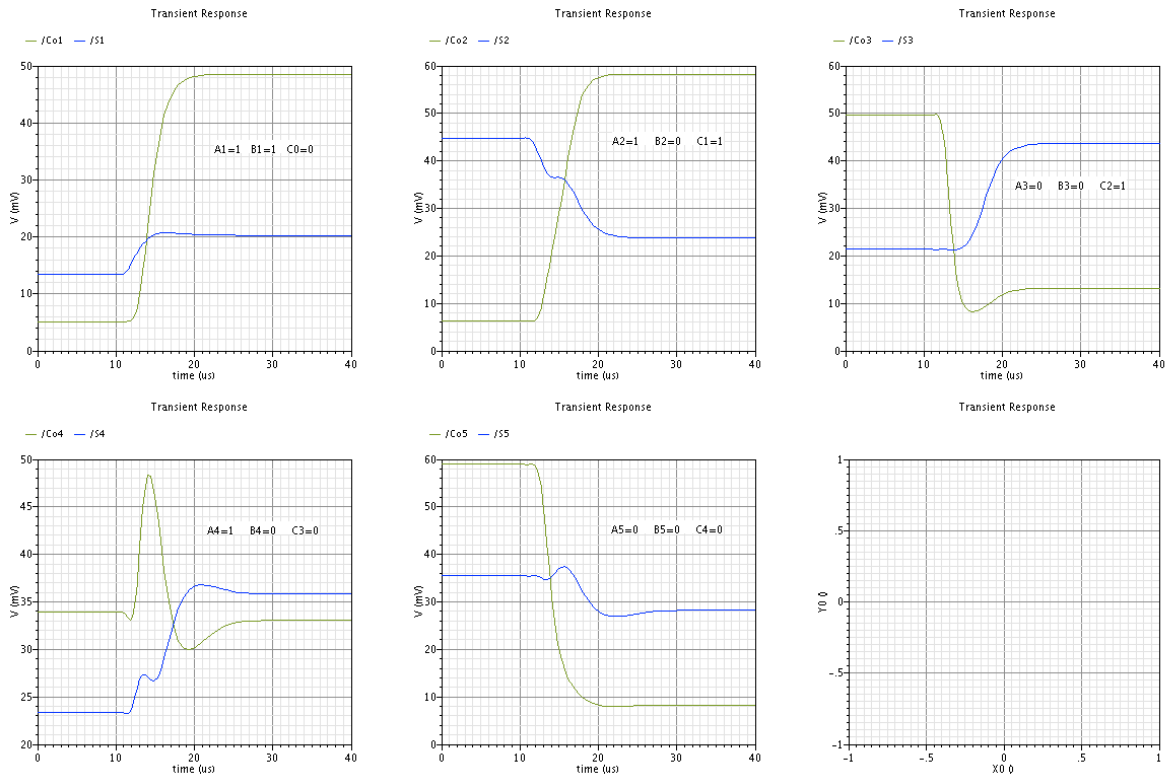
A conventional single bit full adder based on transmission gate logic (TGL) as it is shown in figure (4.8.1) is tested with applied 75mV voltage supply. The test results of a single bit full adder that is shown in figure (4.8.1) is pictured in figure (4.8.2). All the input signals are buffered same reason and way that it was discussed earlier. Results of a single bit are acceptable even we could see the result of propagation delay. The next test was done on a 5 bits conventional adder based on the single bit full adder which was discussed and tested. Result of each stage are shown in transient mode to present the signals conditions in each stage and is shown in figure (4.8.3). Results are showing that signal levels are not acceptable especially after fourth bit which causing false result.



**Figure(4.8.1) A Single Bit Conventional Full Adder based on TGL.**



**Figure(4.8.2) A Single Bit Conventional Full Adder Test Results at 75mV.**



**Figure(4.8.3) A 5 Bits Conventional Adder Test Results at 75mV.**

# Chapter 5

## Conclusion

### 5.1. Project Review

Low and ultra low-power circuits are becoming more desirable as portable devices markets are growing and they also become more interested and applicable in biomedical, pharmacy and sensor networking application because of the CMOS reliability improvement and the nano-metric scaling.

Architectures like adiabatic, winner takes all and pipelined are introduced along with obvious solution to reduce the speed and supply voltage or using multiple supply voltage. However still there is more references need to do the comparison in each and every cases similar to the high speed application. Reducing voltage leads to sub-threshold and weak inversion operation along with asynchronous pipeline architectures which are the most efficient solutions for ultra low-power consumption. But having architecture that use all these efficient solution is a challenge.

Proper algorithm is the answer to win that challenge and have an appropriate architecture, the architecture that implies precise asynchronous pipeline technique in sub-threshold and ultra low voltage. Majority-function algorithm and two synthesis method of this algorithm which was the voltage-mode (VMMF) and current-mode discussed and reviewed. VMMF because of the capacitor network on front and dividing operating voltage to  $VDD/3$  wasn't an ideal case for the sub-threshold. However using current mode (CMMF) and converting it to voltage immediately was the answer. In this thesis current mode majority function adder is presented for the first time. Also for the first time ultra low voltage of 75 mV was implied. Therefor three major areas in this

thesis are, current-mode majority-function algorithm, achieving to 75 mV operating voltage and the technique to implement 64-bit adder.

In summary, the majority-function algorithm provides advantages like,

- 1- It provides direct and quick way to calculate the carry and the sum by a simple comparison.
- 2- Because of using current mode it is fast even in ultra low voltages.
- 3- It requires and implies pipeline architecture.
- 4- It works perfectly with asynchronous self-timed circuit.
- 5- The input signals evaluating needs less than  $V_{DD}/2$  or  $V_M$  and because of that it consumes low power.
- 6- There are many ways to implement the algorithm. The one which was the best suitable for the sub-threshold and ultra low power circuits only introduced.
- 7- It requires less transistors to implement in compare with others.

Proposed circuit can operate voltage from 200 mV to as low as 75 mV. Total 64-bit calculation took 860  $\mu$ s at 75 mV applied voltage. This means about 13  $\mu$ s time requires for each stage or bit. The operation consumed 4.5 nW or 3.8 pJ with 75 mV applied voltage and defined input signals “FFFF,FFFF,.....,FFFF,FFFF, and “FFFF,0000,.....,0000,FFFF”. Input signals are meant to realize maximum and worse case condition for both operating time and power consumption. Delta current consumption was about 20 nA and it linearly increased from first stage/bit to the last. Total leakage current was about 55 nA before operation begins, before applying EN signal (enable signal). 64-bit adder did respond correctly and returned output signals are consistent and have acceptable voltage level.

The conventional circuit failed at 75 mV applied voltage on the fourth bit. Signal levels are poor and unacceptable.

## 5.2. Future works

For the future, it is important to experience transistor exact behavior in ultra low voltage like 75mV as we saw and discussed about some uncertainty and discrepancy in Cadence simulation in that region. The current architecture is very easy and uses easy and straight forward logics. The key parts perhaps are the starving mechanism which requires more investigation for better linearity. Current architecture uses two pulse generators for the carry and the sum circuits, however it is possible to ratio the inverters and use only one pulse generator. This can reduce number of transistors almost 30%. Using other architectures to implement majority factor and doing more tests and comparison would be very suitable.

# References

1. A. Wang; B. H. Calhoun, A. P. Chandrakasan, "Sub-Threshold Design For Ultra Low-Power Systems," Springer, 2006.
2. C. Piguet, "Low-Power CMOS Circuits Technology, Logic Design and CAD Tools," Taylor & Francis Group, 2006.
3. J. M. Rabaey. "Digital Integrated Circuits: A design Perspective". Prentice-Hall, 1996.
4. C. C. Nez; E. A. Vittoz. "Charge-Based MOS Transistor Modeling". John Wiley, 2006.
5. R. Sarpeshkar, "Ultra Low Power Bioelectronics: Fundamentals, Biomedical Applications, and Bio-Inspired Systems," Cambridge, 2010.
6. K. Navi; M. H. Moaiyeri; T. Nikoubin, "New High Performance Majority Function Based on Full Adders," 14<sup>th</sup> International CSI Computer Conference, page(s): 100-104, 2009.
7. K. Navi; M. M. Naeini, "A New Full-Adder Based on Majority Function and Standard Gates," Journal of Communication and Computer, vol. 7, no. 5, page(s): 1-7, May 2010.
8. V. Fortunate; K. Navi; M. Haghparast. "A New Low Power Dynamic Full Adder Cell Based on Majority Function," World Applied Sciences Journal, vol. 4, no. 1, page(s): 133-141, 2008.
9. S. Naraghi, "Reduced Swing Domino Techniques for Low Power and High Performance Arithmetic Circuits," A thesis of Master of Applied Science, University of Waterloo, 2004.
10. R. D. Jorgensen; L. Sorensen; D. Elet; M. S. Hagedorn; D. R. Lamb; T. Hal Friddell; W. P. Snapp. "Ultralow-Power Operation in Subthreshold Regimes Applying Clockless Logic," Proceedings of the IEEE, vol. 98, no. 2, page(s): 299-314, Feb. 2010.
11. M. Vratonjic; B. R. Zeiden; V. G. Oklobdzija, "Low-and Ultra Low-Power Arithmetic Units: Design and Comparison," Proceedings of the IEEE International Conference on Computer Design: VLSI in computers and processors, page(s): 249-252, 2005.
12. S. G. Younis, "Asymptotically Zero Energy Computing using Split- Level Charge Recovery Logic," MIT, Technical Report AITR-1500, June 1994.
13. M. P. Frank, "Physical Limits of Computing. Lecture #24 Adiabatic CMOS," Spring 2002.

14. M.H. Moaiyeri; R.F. Mire; K. Navi; T. Nikoubin, "New High-Performance Majority Function Based Full Adders," 14<sup>th</sup> International CSI Computer Conference, page(s): 100-104, 2009.
15. O.C. Akgun; J. Rodrigues; J. Spars, "Minimum-Energy Sub-Threshold Self-Timed Circuits- Design Methodology and a Case Study," IEEE Symposium on Asynchronous Circuits and Systems, page(s): 41-51, May 2010.
16. S.F. Al-Sarawi. "Low Power Schmitt trigger circuit," Electronic Letters, vol. 38, page(s): 1009-1010. Aug 29, 2002.
17. H. Peak; A. Yousif; J.W. Haslett, "A CMOS Integrated Linear Voltage-to-Pulse-Delay-Time Converter for Time Based Analog-to-Digital Converters," Proceedings of the IEEE International Symposium on Circuits and Systems, page(s): 2373- 2376, 2006.
18. J.P. Kulkarni; K. Keejong; K. Roy "A 160 mV Fully Differential Robust Schmitt Trigger Based Sub-threshold SRAM," IEEE Journal of Solid State Circuits, vol. 42, page(s): 2303-2313, Oct. 2007.
19. P. Hylander; J. Meader; E. Frie "VLSI Implementation of Pulse Coded Winner Take All Networks," Proceedings of the 36th Midwest Symposium on circuits and systems, vol.1, page(s): 758-761, Aug 1993.
20. B. Sekerkiran. "A High Resolution CMOS Winner-Take-All circuit," Proceedings of the IEEE International Conference on Neural Networks, vol. 4, page(s): 2023-2026, Nov./Dec. 1995.
21. H. Hating; K. Sakaue; K. Naruke, "CMOS shift register circuits for radiation-tolerant VLSI's," IEEE Transactions on Nuclear Science, page(s): 1034-1038, Oct. 1984.
22. M.H. Moaiyeri; R.F. Mire; K. Navi; T. Nikoubin, "New High-Performance Majority Function Based Full Adders," 14<sup>th</sup> International CSI Computer Conference, page(s): 100-104, 2009.
23. N. Ghobadi; R. Maid; M. Tehran; A. Afzali-Kusha. "Low Power 4-Bit Full Adder Cells in Subthreshold Regime," 18th Iranian Conference of Electrical Engineering (ICEE), page(s): 362-367, May 2010.
24. H. Peak; A. Yousif; J.W. Haslett, "A CMOS Integrated Linear Voltage-to-Pulse-Delay-Time Converter for Time Based Analog-to-Digital Converters," Proceedings of the IEEE International Symposium on Circuits and Systems, page(s): 2373- 2376, 2006.

25. J.P. Kulkarni; K. Keejong; K. Roy “A 160 mV Fully Differential Robust Schmitt Trigger Based Sub-threshold SRAM,” IEEE International Symposium on Low Power Electronics and Design, page(s): 171-176, 2007.
26. A. Wang; A. Chandrakasan “A 180-mV Subthreshold FFT Processor Using a Minimum Energy Design Methodology,” IEEE Journal of Solid State Circuits, vol. 40, page(s): 310-319, Jan 2005.
27. A. P. Chandrakasan; S. Sheng; R. W. Brodersen. “Low-Power CMOS Digital Design,” IEEE Journal of Solid-State Circuits, vol. 27, no. 4, April 1992.
28. O.C. Akgun; Y. Leblebici; E. A. Vittoz. “Design of Completion Detection Circuits for Self-timed Systems Operating in Subthreshold Regime,” PhD. Research in Microelectronics and Electronics Conference, page(s): 241-244, July 2007.
29. B. H. Calhoun; A. Wang; A. Chandrakasan. “Device Sizing for Minimum Energy Operation in Subthreshold Circuits,” Proceedings of the IEEE on Custom Integrated Circuits Conference, page(s): 95-98, Oct 2004.
30. A. Wang; A. P. Chandrakasan; S. V. Kosonocky. “Optimal Supply and Threshold Scaling for Subthreshold CMOS Circuits,” Proceedings of the IEEE Computer Society Annual Symposium on VLSI, page(s): 5-9, April 2002.
31. D. Blaauw; B. Zhai. “Energy Efficient Design for Subthreshold Supply Voltage Operation,” Proceedings of the IEEE International Symposium on Circuit and Systems, page(s): 29-32, May 2006.
32. J. Wkong; Y. K. Ramadass; N. Verma; A. P. Chandrakasan. “A 65 nm Sub-threshold Microcontroller With Integrated SRAM and Switched Capacitor DC-DC Converter,” IEEE Journal of Solid-State Circuits, vol. 44, no. 1, page(s): 115-126, Jan 2009.
33. B. C. Paul; A. Raychowdhury; K. Roy. “Device Optimization for Digital Subthreshold Logic Operation,” IEEE Transactions on Electron Devices, vol. 52, no. 2, page(s): 237-247, Feb. 2005.
34. H. Soelernan; K. Roy. “Ultra-Low Power Digital Subthreshold Logic Circuits,” Proceedings of the International Symposium on Low Power Electronics and Design, page(s): 94-96, Aug, 1999.



35. S. K. Gupta; A. Raychowdhury; K. Roy. "Digital Computation in Subthreshold Region for Ultralow-Power Operation: A Device–Circuit–Architecture Codesign Perspective," Proceedings of the IEEE, vol. 98, no. 2, page(s): 160-190, Feb 2010.
36. M.J. Turnquist "Sub-threshold Operation of a Timing Error detection latch," Ph.D Research in Microelectronics and Electronics, page(s): 124-127, July 2009.
37. W. C. Athas; L. J. Svensson; J.G. Koller; N. Tzartzanis; E. Ying-Chin Chou. "Low-Power Digital Systems Based on Adiabatic-Switching Principles," IEEE transaction on VLSI System, vol. 2. no. 4, page(s) 398 -407, December 1994.
38. P. D. Khandekar; S. Subbaraman. "Low Power Inverter and Barrel Shifter Design Using Adiabatic Principle," Advances in Computational Sciences and Technology, vol. 3, no.1, page(s) 57–65, 2010.
39. L. Varga; F. Civics; G. Hosszu "An Efficient Adiabatic Charge-Recovery Logic," Proceedings of the IEEE, SoutheastCon, page(s): 17-20, 2001.
40. J. T. Ako, "Subthreshold Leakage Control Techniques for Low Power Digital Circuits," PhD thesis, MIT University, May 2001.
41. H. Jacobson; P. Bose; Z. Hu; A. Buyuktosunoglu; V. Zyuban, "Stretching the Limits of Clock-Gating Efficiency in Server-Class Processors," 11<sup>th</sup> International Symposium on High-Performance Computer Architecture , page(s): 238 -242, Feb. 2005.
42. D. Harris; "Lecture 11," Harvey MUDD college, Spring 2004.
43. <http://www.aoki.ecei.tohoku.ac.jp/arith/mg/algorithm.html>.
44. [http://en.wikipedia.org/wiki/Carry-select\\_adder](http://en.wikipedia.org/wiki/Carry-select_adder).
45. [http://www.ece.msstate.edu/~reese/EE8273/lectures/domino\\_pipe.pdf](http://www.ece.msstate.edu/~reese/EE8273/lectures/domino_pipe.pdf)
46. <http://www.scribd.com/doc/44035390/Adiabatic-Logic>.
47. <http://en.wikipedia.org/wiki/Winner-take-all>.