

2014

16-bit Digital Adder Design in 250nm and 64-bit Digital Comparator Design in 90nm CMOS Technologies

Naga Venkata Vijaya Krishna Boppana
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all



Part of the [Electrical and Computer Engineering Commons](#)

Repository Citation

Boppana, Naga Venkata Vijaya Krishna, "16-bit Digital Adder Design in 250nm and 64-bit Digital Comparator Design in 90nm CMOS Technologies" (2014). *Browse all Theses and Dissertations*. 1385. https://corescholar.libraries.wright.edu/etd_all/1385

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

16-bit Digital Adder Design in 250nm and 64-bit Digital
Comparator Design in 90nm CMOS Technologies

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

By

N.V. Vijaya Krishna. Boppana
B.E., Andhra University, 2011

2014
Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

Dec 29, 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY N.V. Vijaya Krishna. Boppana ENTITLED “Design of Fast, Low Power and Area Efficient Static and Dynamic Digital Designs in 250nm and 90nm CMOS Technologies” BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

Saiyu Ren, Ph.D.
Thesis Director

Brian D. Rigling, Ph.D.
Chair, Department of Electrical Engineering
College of Engineering and Computer Science

Committee on Final Examination

Saiyu Ren, Ph.D.

Raymond E. Siferd, Ph.D.

John M. Emmert, Ph.D.

Robert E. W. Fyffe, Ph.D.
Vice President of Research and Dean of the
Graduate School

ABSTRACT

Boppana, N.V. Vijaya Krishna. M.S.Egr., Department of Electrical Engineering, Wright State University, 2014. Design of Fast, Low Power and Area Efficient static and dynamic digital designs in 250nm and 90nm CMOS Technologies.

High speed, low power, and area efficient adders and comparators continue to play a key role in hardware implementation of digital signal processing applications. Adders based on Complimentary Pass Transistor Logic (CPL) are power and area efficient, but are slower compared to Square Root Carry Select (SQRT-CS) based adders. This thesis demonstrates a unique custom designed 16-bit adder in 250-nm CMOS technology to obtain fast and power/area efficient features by combining CPL and CS logic. Comparing the results obtained for proposed 16-bit Linear CPL/CS adder with the BEC (Binary Excess-1 Code) based low power SQRT-CS adder, the delay is reduced by approximately one thirds, power is reduced by 19.2%, and the number of transistors is reduced by 23.4%. Also, new tree-based 64-bit static and dynamic digital comparators are presented in this thesis to perform high speed and low power operations. This tree-based architecture combines a new approach of designing dynamic comparator using a low duty cycle clock to reduce the short circuit power consumption in pre-charge (or pre-discharge) mode. This work also introduces a new sizing strategy and load balancing techniques to improve self-pipelining tendency of a tree based design. A resource sharing technique is also integrated in both static and dynamic comparator designs. At 1.2V

power supply in CMOS 90nm technology, worst path delay and worst power are 374ps and 822 μ W, respectively for low cost static design with 1244 (768+476) transistors in total. 768 transistors are used for resource sharing. The proposed full and partially dynamic designs show superior power efficiency compared to recent state of art designs. The worst power consumptions at 5GHz and 25% (50ps) duty cycle clock for the 64-bit full and partially dynamic comparator designs are 5.00mW and 2.78mW, respectively. 769 (320+449) transistors includes 320 transistors for resource sharing, and 1217 (768+449) includes 768 transistors for resource sharing for full and partial dynamic comparators, respectively.

TABLE OF CONTENTS

1	Introduction	1
1.1	Static designs.....	1
1.2	Dynamic designs	4
1.3	Motivation	5
1.4	Thesis Organization.....	7
2	16-bit Low-Power, High Speed CPL-CSA Adder using 250nm CMOS Technology	8
2.1	Introduction	8
2.2	CPL implementation in 16-bit Carry Select Adder (CSA) design.....	8
2.2.1	Carry Select Adder.....	8
2.2.1.1	Square root -CSA using RCA and BEC [4]	9
2.2.2	Carry Select Adder using CPL.....	13
2.2.2.1	4-Bit Regular CPL Adder	14
2.2.2.2	16-Bit SQRT_CPL_CSA using RCA/BEC.....	17
2.2.2.3	4-Bit Proposed CPL CS Adder	17
2.2.2.4	16-Bit SQRT/Linear_CPL_CSA using proposed design.	22
2.3	Limitations for proposed 4-bit CPL CSA	24

2.4	Results comparison	24
3	A High Speed and Low Power 64-Bit Digital Comparator using 90nm CMOS Technology	25
3.1	Introduction	25
3.2	Operating Principal and Design Methodology.....	27
3.3	Static and Dynamic XE logic	29
3.4	Design of proposed 64-bit static comparator using Level-1 & Level-2 sub blocks	34
3.4.1	Level-1 4-Bit Static Comparator Sub Block Design.....	34
3.4.2	Level-2 4-bit Static Comparator Sub-block Design.....	37
3.4.3	Delay and Power Optimization Strategies	39
3.4.4	Tree Based 64-bit Low Cost Static Comparator Design.....	40
3.4.5	Schematic Simulation Results.....	43
3.5	Proposed 64-bit dynamic comparator design using Level-1 & Level-2 sub blocks	44
3.5.1	Level-1 and Level-2 4-Bit dynamic sub block design	44
3.5.2	Proposed 64-bit, low cost, full dynamic, self-pipelined comparator design using radix-4 structure	47
3.5.2.1	Sizing and Pre-charging Strategies:.....	48
3.5.3	Proposed 64-bit partially dynamic comparator design in radix-4 structure	49
3.6	Simulation Results and Discussion	51

4	Conclusion & Future Work	59
4.1	Conclusion.....	59
4.2	Future Work	60
5	References	61

LIST OF FIGURES

Figure 1.1 AND logic [1] [2]using (a) CMOS, (b) CPL, and (c) DPL	3
Figure 1.2 Examples of footed dynamic [2]logic design (a) NAND, and (b) NOR	5
Figure 2.1 Intermediate block of 16 bit SQRT CSA, (a) using RCA+RCA, (b) RCA+BEC [2].....	10
Figure 2.2 (a) 1-bit 28T full adder, (b) N-bit RCA, (c) N-bit BEC using CPL_XOR and CPL_AND, and (d) 2-to-1 Transmission gate Multiplexer.	12
Figure 2.3 1-bit CPL adder cell [8] [9].	14
Figure 2.4 4-bit regular CPL adder [9].	15
Figure 2.5 Internal block structure of regular CPL adder: (a) P-block, (b) Q-block, (c) R-block, (d) S-block.....	16
Figure 2.6 4-bit proposed CPL adder cell.....	18
Figure 2.7 Internal block structure of modified CPL adder: (a) Mux, (b) P-block, (c) Q-block, (d) R-block.	19
Figure 2.8 Schematic implementation of 4-bit CPL adders in 250nm CMOS technology (a) Regular (b) Proposed/modified.	21
Figure 2.9 16-Bit CPL CSA using 4-bit proposed design: (a) SQRT, (b) Linear.....	23
Figure 3.1 (a) Basic principal for binary data comparison, (b) Comparator design modification from traditional 3-bit output to encoded 2-bit output.....	28
Figure 3.2 N-Bit Comparison	29

Figure 3.3 XE block (a) 12T-static [24], (b) 5T-dynamic.	31
Figure 3.4 Dynamic XE block performance using (a) traditional clock (D=50%) and (b) clock with reduced duty cycle (D<50%).	33
Figure 3.5 Level-1 static 4-Bit comparator sub-block.....	36
Figure 3.6 Level-2 static 4-Bit comparator sub-blocks	39
Figure 3.7 Proposed 64-bit static comparator using radix-4 tree structure.....	41
Figure 3.8 Worst and best path delay representation in radix-4 structure.	42
Figure 3.9 Input vectors to measure (a) Worst path delay, (b) Worst power, and (c) Maximum power consumption.	43
Figure 3.10 Level-1 dynamic 4-bit comparator sub-block	46
Figure 3.11 Level-2 dynamic 4-bit comparator sub-block	47
Figure 3.12 Clock strategy for proposed radix-4 64-bit full dynamic comparator design	49
Figure 3.13 Clock strategy for proposed radix-4 64-bit partially dynamic comparator design	50
Figure 3.14 Schematic designs for 64-bit digital comparators in 90nm CMOS technology (a) Static (b) Full dynamic (c) Partially dynamic	52
Figure 3.15 Worst case vectors (a) for delay, (b) for power at 5 GHz clock.....	53
Figure 3.16 Simulation results showing worst case delay at 40ps (20%) duty cycle (D) clock for 64-bit dynamic comparator.....	54

LIST OF TABLES

Table 2.1 4-bit CPL adder results comparison at 500-Mhz activity	22
Table 2.2 16-bit SQRT-CSA results comparison at 100-Mhz activity	23
Table 3.1 Simulation results of static and Dynamic XE blocks	34
Table 3.2 Simulation results for L1 and L2 static comparator sub blocks at an activity (Data rate) of 500MHz (1GHz).....	40
Table 3.3 Performance Comparisons for 64-bit Static and Dynamic Comparator	55

ACKNOWLEDGMENT

I would like to thank my advisor Dr. Saiyu Ren for her continuous support through-out my graduation career at Wright state university. Dr. Ren's suggestions and guidance in working on this thesis and writing are irreplaceable. I would like to thank my VLSI professors for sharing their knowledge. I would also like to thank Mike Van Horn for the technical assistance.

I would like to thank my parents and teachers for their love, patience and support.

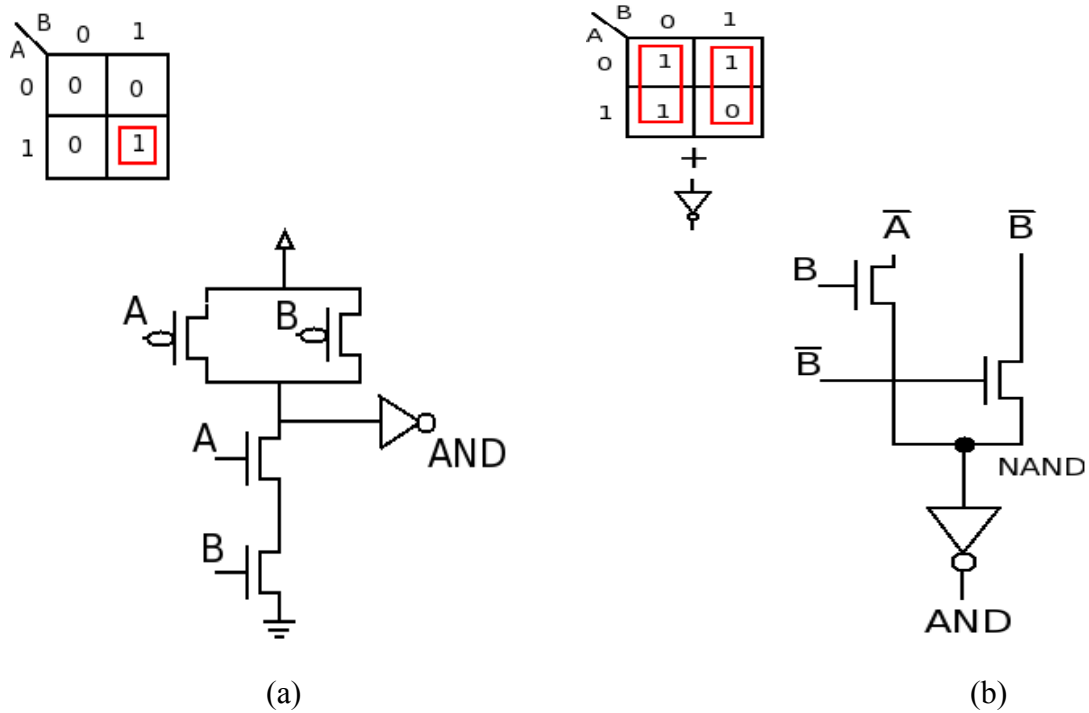
1 Introduction

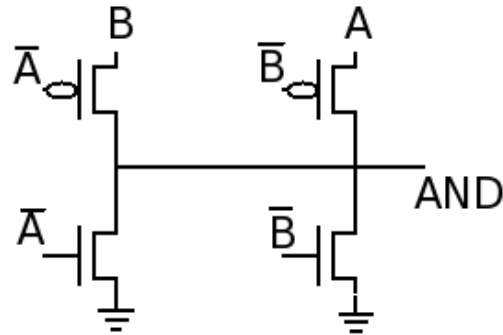
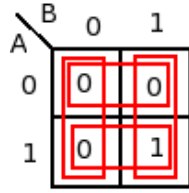
Digital designs have variety of data processing, controlling and data storing applications. Combinational and sequential designs, such as adders, sub-tractors, latches, flip-flops, multiplexers, shifters, encoders, decoders, counters, etc., are the sub-components of large scale digital design applications, such as microprocessors, microcontrollers, digital signal processors etc. Digital sub-components are built by using standard cells, such as NAND, NOR, AND, OR, INVERTER, XOR, XNOR. These standard cells can be designed in static or dynamic design methodology.

1.1 Static designs

Static designs can be designed either by using Complementary Metal Oxide Semiconductor (CMOS) logic or by using Pass Transistor Logic (PTL) [1] [2]. In CMOS logic, either pull up network (p-MOS) or pull down network (n-MOS) will be ON at a time. Number of transistors in pull up and pull down networks are equal in CMOS designs. In case of pass transistor designs, input signals drive the gate as well as pass through the FET. Pass transistor based designs require less number of transistors and power consumption is also less when compare with the CMOS designs. Power consumption in PTL is less because this logic requires less number of transistors and does not require power and ground connections to implement standard logic compared with CMOS static logic. But it has disadvantage that pass logic generates degraded signal to drive other gates and passes noise from input to output.

Complimentary Pass transistor Logic (CPL) and Double Pass transistor Logic (DPL) are two typical types of pass transistor logic implementation. Figure 1.1 demonstrates the difference of implementing AND2 gate by using CMOS logic, CPL and DPL, respectively.





(c)

Figure 1.1 AND logic [1] [2] using (a) CMOS, (b) CPL, and (c) DPL

As shown in Figure 1.1(a), 6 transistors are needed to implement an AND2 function of $Y = ((AB)') + \text{inv}$ with the attached Karnaugh map (K-map). It can be seen that the number of n-MOS and p-MOS transistors are always equal and in complement to each other, which results in that only one network (either pull up or pull down) is on and another is off at any stable state. This logic has been the most robust logic to generate strong signals with low noise and unique feature of automatic noise error recovery. The relative low speed and large number of transistors triggered other logic architecture developed.

Figure 1.1(b) shows the implementation of AND2 using complimentary pass transistor logic (CPL). It needs 8 transistors (3 inverters and 2 pass transistors) to design AND2 by designing complement $((AB)')$ of the function in pass logic using K-map

shown in Figure 1.1(b) and an inverter should be added at the output of pass logic to yield strong and noise free output for AND2. Though AND2 logic implementation seems it requires more number of transistors and power compared to static CMOS, CPL designs have advantages while designing complex, complimentary and dual functions which share the input inverters to design all the functionalities.

In other hand, DPL implementation of AND2, using 2 inverters + 4 transistors, shown in Figure 1.1(c) has an advantage of speed when compare with the CPL design. According to the K-map shown in Figure 1.1(c) every logic should come from two transistors in parallel (n-n or p-p or n-p) which reduces the delay when compare with CPL. CPL is preferable to DPL when high output drive capabilities required.

1.2 Dynamic designs

Dynamic designs [2] are useful in designing some high speed applications with power trade off. Dynamic designs need less number of transistors with adding a clock control as shown in Fig. 1.2. Any logic function will be implemented only by pull down network (n-MOS transistors). Single p-MOS transistor is used to pre-charge the output node to be '1'. This logic gets rid of the large capacitance contribution of PUN and reduces the delay. Figure 1.2 shows footed dynamic logics for dynamic NAND and NOR designs. In pre-charge phase of a footed design, when clock is low, the output node of containing self-load and gate load capacitances is charged to high. In evaluation phase, if pull down network is not ON, the output will keep high ('1'); if PDN is on, the output will be pulled to low ('0'). It can be seen that transistor count and input/output node capacitance are reduced compared to static CMOS logic. Reduction in capacitance at

input output decreases the delay which results in high speed performance. The tradeoff of this logic is high power consumption and non-recovering of noise resulted error.

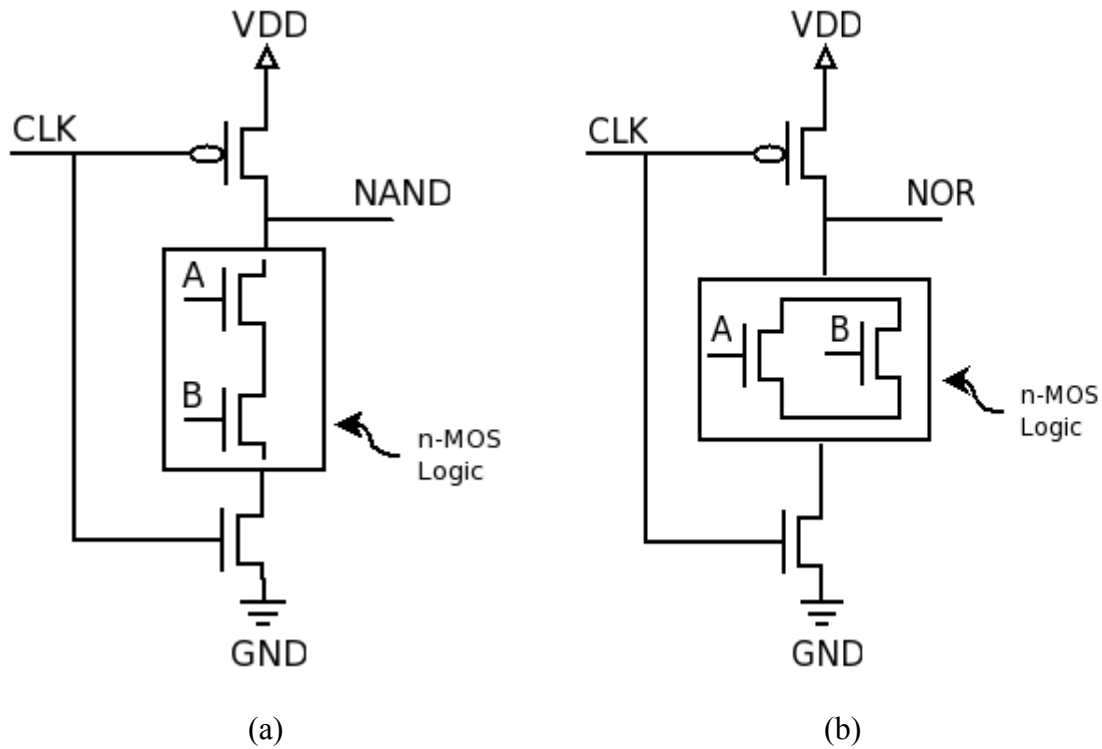


Figure 1.2 Examples of footed dynamic [2]logic design (a) NAND, and (b) NOR

1.3 Motivation

With the increasing requirement for huge number of functionalities performed by a single chip, semiconductor industries are trying to encompass billions of transistors in a small space. Speed of the operations is in great need while reducing the power consumption. The goal for every designer is to design a low power, high speed, and low cost chip. Some applications may emphasize speed more, others power more or cost more. As mentioned earlier, adders, comparators are the major basic building blocks of any processor or controller device. Reducing size, power consumption and increasing speed

in micro level is one way of achieving portability, power efficiency and high speeds in macro level.

Performance specifications in digital designs can be achieved by scaling the transistor size. Technology is scaling down to atomic sizes. Technology scaling along with novel and efficient techniques of implementation of a digital design helps reducing the size and power consumption while keeping the speed of operation. There are many works dealing with designing high speed and cost efficient adders and comparators for different applications. For example, Ripple Carry, Carry Look-Ahead (CLA), Carry Bypass, Carry Select, Carry Save, Carry Skip, Look-Ahead Carry, Carry Complete etc., are the different techniques to implement adders and all-N-transistor (ANT), Priority Encoding (PE), Multiple Output Domino Logic, Multi Level Look Ahead, Parallel MSB Checking, Bit-wise Competition Logic, Single Clock Cycle Tree structure, Constant Delay logic based comparator etc., are the different techniques used to design digital comparators.

The motivation of reducing power consumption and portability of designs leads to designing a 16-bit static carry select adder design using CPL and CMOS styles. Also, the requirement of high speed and low power 64-bit digital comparator operation leads to introducing a novel design for static, dynamic and partially dynamic designs with new techniques of power and delay optimizations using with clock with reduced duty cycle and improving self-pipelining tendency of the design along with resource sharing.

1.4 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 discusses the implementation of a new architecture using static CMOS and Complimentary Pass Transistor (CPL) logic to design low cost and high speed 16-bit Carry Select Adders (CSA). Designs in this chapter are realized using 250nm CMOS technology. Chapter 3 presents the design and implementation of a new radix-4 tree structured 64-bit digital comparator in static, full dynamic, and partially dynamic modes suitable for low cost and high speed applications. This chapter introduces a new concept of using clock with reduced duty cycle to reduce the short circuit power consumption of a dynamic design and a concept of improving self-pipelining of the tree based design. All the designs in this chapter are realized in 90nm CMOS technology. Finally, conclusions and details about future work are included in Chapter 4.

2 16-bit Low-Power, High Speed CPL-CSA Adder using 250nm CMOS Technology

2.1 Introduction

Digital adders are the key components in microprocessor design [3], digital signal processors (DSP), etc. Though feature size is reducing gradually, increase in need for multifunctional processors enlarge the chip area and increase the power consumption. Scaling of transistors is reaching almost atomic levels and at some point of time VLSI designs might need inventions at atomic levels. But now with the current feature sizes there is a requirement for reducing chip area and power consumption of a particular functional design by using some novel techniques of implementation. There are many adder architectures [2] designed with emphasis on reducing delay (D), lowering transistor count/area (A) and power consumption (P). Each architecture has limitation and trade-off for the above three specifications. As mentioned earlier, the basic motive behind this section of work is to design and implement a novel technique for a small, fast and low power dissipating 16-bit static adder using Complimentary Pass transistor Logic (CPL) and Carry Select Adder (CSA) architecture.

2.2 CPL implementation in 16-bit Carry Select Adder (CSA) design

2.2.1 Carry Select Adder

Ripple carry multi-bit adder takes more time to finish the computation because of carry propagation through all one bit adder blocks. General carry select adder

architecture [4] consists of two sets of ripple carry adders and a multiplexer. It has two architectures, conventional/linear carry select (LCS) and square root carry select architecture (SQRT CS). LCSA design for multi bit addition use 'p' number of 'm' bit carry select blocks to get $n (= p*m)$ bit addition. SQRT CSA design use 'm' bit carry select block as first block of carry select. In later stages of carry select, the number of bits of carry select addition in each block increases by one bit until it reaches 'm+p-1' bits for p^{th} block. An n bit addition is obtained with p blocks, where $n (= p^2/2 + [p*m - (p/2)])$. The main advantage of using carry select adder is the sum and carry out of any sub block are predicted. In carry select adder, from second sub block to final sub block, the sum outputs and carry outs are selected. So, the design doesn't add any carry propagation delay other than initial sub block set up time (internal ripple carry) and the delay in later stages caused by fan out, selection at multiplexers, for preceding block carry out.

2.2.1.1 Square root -CSA using RCA and BEC [4]

SQRT CSA architecture is faster than Linear CSA architecture because processing time to get m bit addition in p^{th} block is nearly less than or equal to processing time of (m-1) bit addition plus processing time of selecting outputs of (m-1) bit addition through MUX. The sum and carry out bits waiting at MUX are selected by preceding block's carry. In case of SQRT CSA, there is almost no inactive or waiting time between any two adjacent blocks to select the sum and carry out. Thus the delay is less for SQRT CSA and the difference in delay between SQRT CSA and Linear CSA increases as the number of bits of addition is increased. There are different techniques in implementing square root CSA [5] [6] [7]. The following discussion is about the two different SQRT CSA designs, regular and modified design using BEC (Binary Excess-1 Code).

General/regular design uses two sets of RCA (RCA1 and RCA2). Low power design proposed in [5] uses one set of RCA and BEC. Figure 2.1(a) and (b) shows the design of 3 bit ($m=3$) intermediate block of 16-bit square root adder using regular design and BEC design respectively.

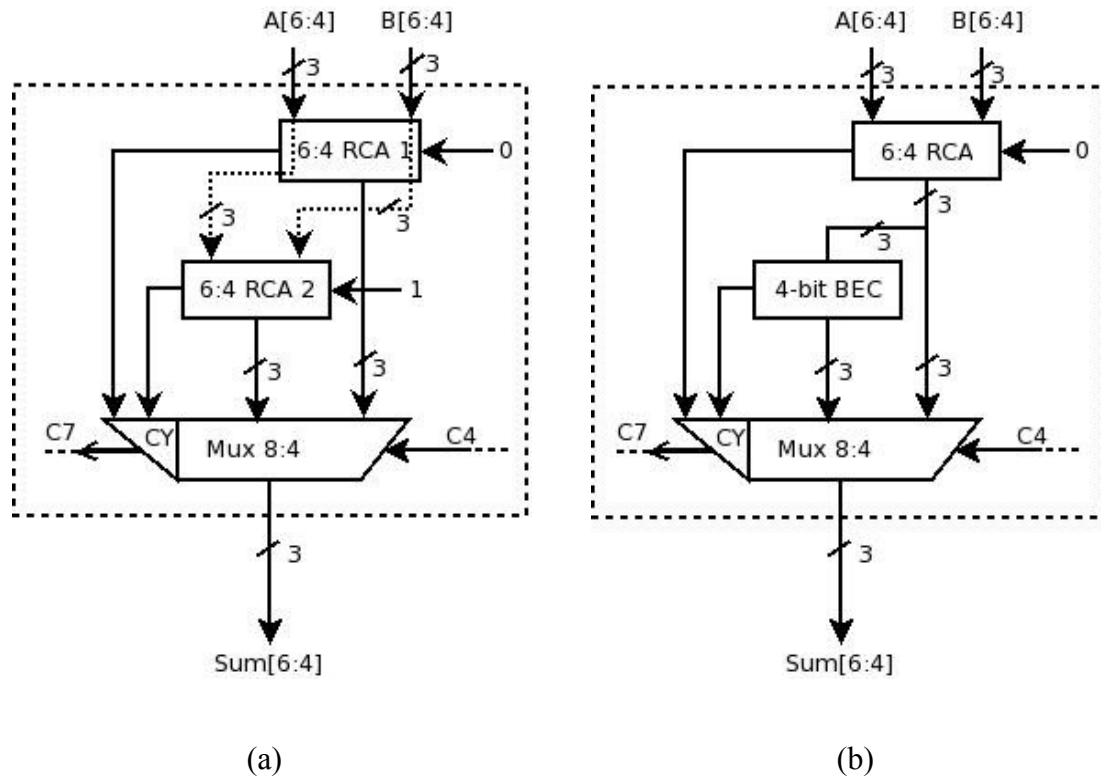
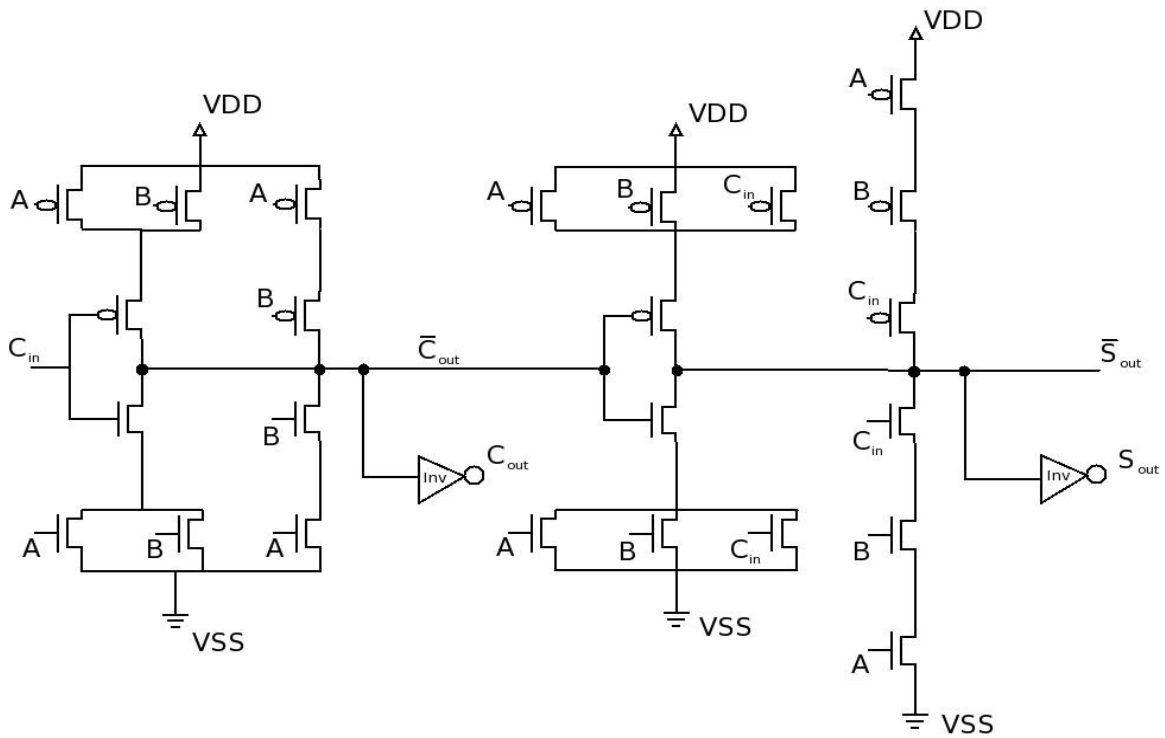


Figure 2.1 Intermediate block of 16 bit Sqrt CSA, (a) using RCA+RCA, (b) RCA+BEC [2].

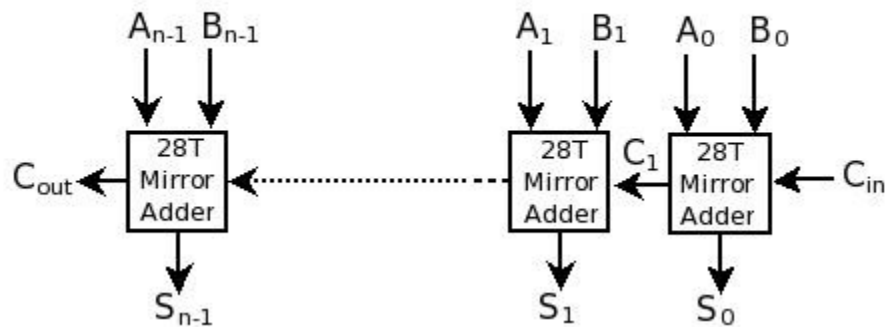
Sqrt CSA using BEC is low power consuming design than using 2 sets of RCA because BEC use less number of components than RCA. However, the delay is high for design with BEC. This is because the computation of sum and carry out from the ripple carry adder has to be done with '0' carry in and then the output has to be processed through BEC to get Excess one output; which replaces the computation needed through RCA 2 (shown in Figure 2.1 (a)) to predict the outputs with an assumption of carry input

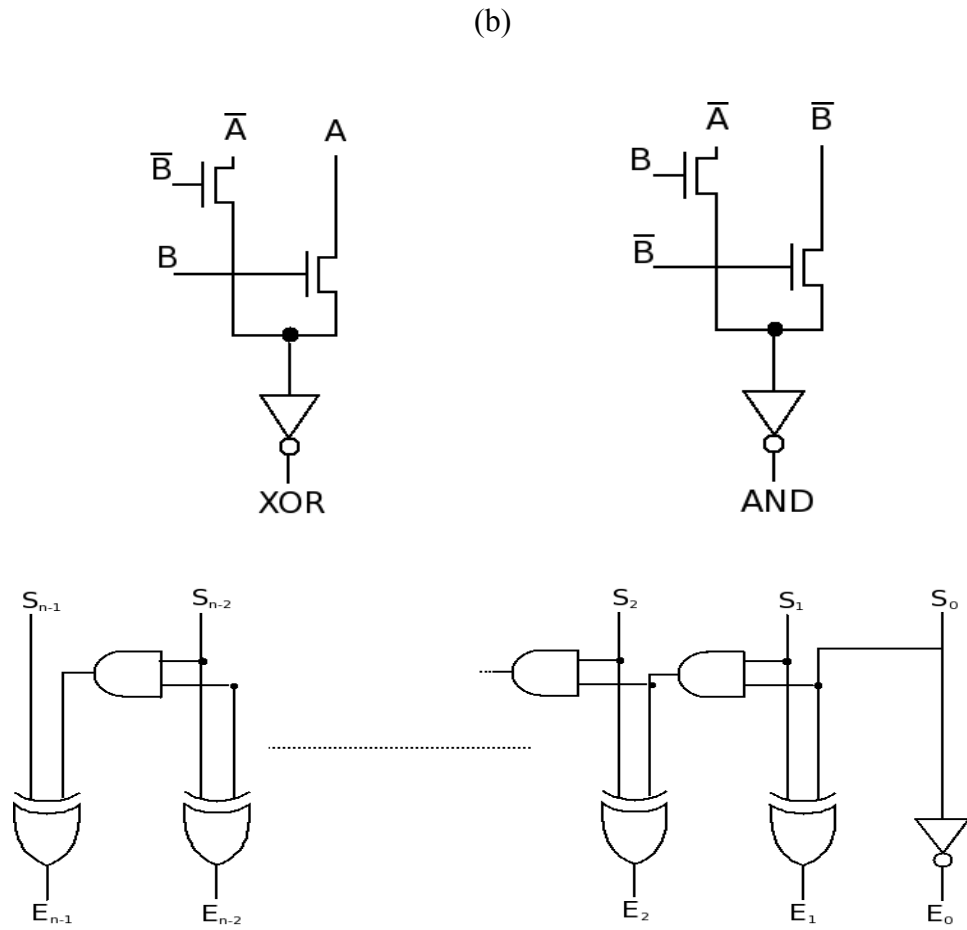
as logic “1”. As shown in Figure 2.2, Mux in regular and BEC based designs are designed by CMOS switch (transmission gate logic) and BEC is designed by using CPL based standard cells XOR and AND. RCA in regular square root carry select adders is designed by using 28T full adder.



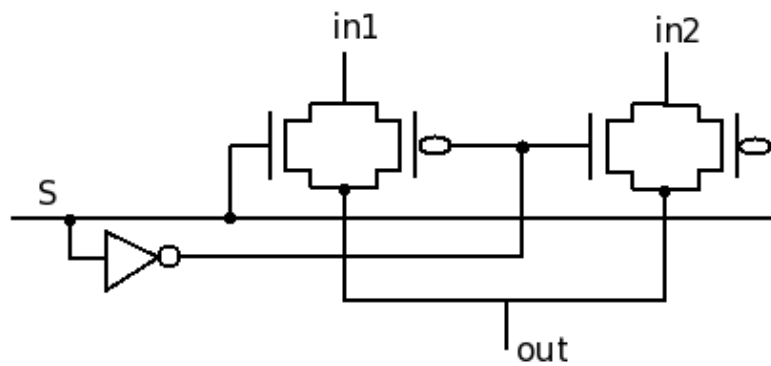
(a)

RCA :





(c)



(d)

Figure 2.2 (a) 1-bit 28T full adder, (b) N-bit RCA, (c) N-bit BEC using CPL_XOR and CPL_AND, and (d) 2-to-1 Transmission gate Multiplexer.

2.2.2 Carry Select Adder using CPL

This paper proposes designing SQR T CSA/LCSA using complimentary pass transistor logic (CPL) to get fast and low power operations. As shown in Figure 2.3, CPL adder cell [8] [9] consume less power because input signals drive the gates as well as pass through the channel of the transistors unlike static CMOS logic. There is no need for VDD and GND connections to implement the adder logic except where an inverter is required; at inputs to generate complimentary signals; at outputs to strengthen the pass signals; in some intermediate stages to strengthen a weak “1” output of an n channel pass gate.

As mentioned in Section 1.1, CPL designs have great flexibility in designing complimentary logic functions ($S0 \Leftrightarrow S0'$, $C1 \Leftrightarrow C1'$, $C0 \Leftrightarrow C0'$) without changing circuit topology but just by inverting the pass signals (complementarity principle) as seen in Figure 2.3. It has a flexibility of designing its dual logic function ($C1 \Leftrightarrow C0$, $C1' \Leftrightarrow C0'$) just by inverting the gate signals. But CPL has disadvantage of signal degradation (voltage level and rise/fall times) if the signal pass through long rail of series transistors. Using the complimentary nature of CPL designs, if the signal is getting weak, the signal can be strengthened by using inverter as a buffer instead of a non-inverting buffer which consumes more power; however, the signal and its compliment signal pins must be switched.

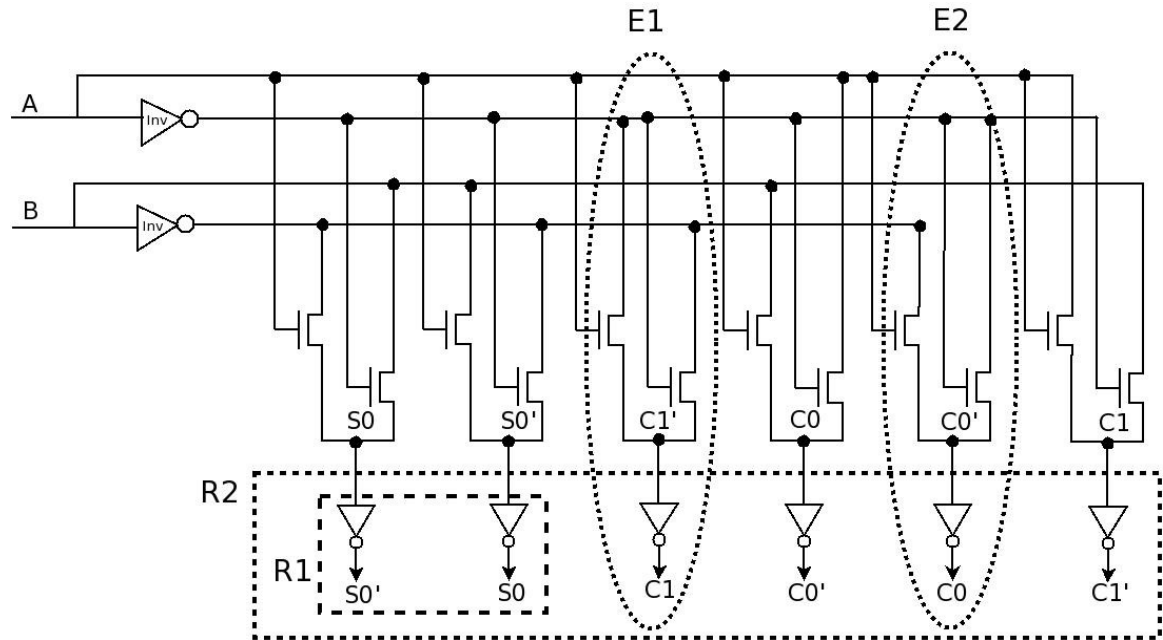


Figure 2.3 1-bit CPL adder cell [8] [9].

$$\begin{array}{ll}
 S0 \Rightarrow A \text{ (XOR) } B & S0' \Rightarrow A \text{ (XNOR) } B \\
 C1 \Rightarrow A \text{ (OR) } B & C1' \Rightarrow A \text{ (NOR) } B \\
 C0 \Rightarrow A \text{ (AND) } B & C0' \Rightarrow A \text{ (NAND) } B
 \end{array}$$

2.2.2.1 4-Bit Regular CPL Adder

CPL adder cell shown in Figure 2.3 works as a Half Adder (HA). It generates sum (S0), carry (C0) and its dual (C1) as well as the complimentary signals (S0', C0', C1'). P, Q, R, S Sub blocks of Block1 and Blocks 2 enclosed in left side dotted rectangle of 4 bit Regular CPL adder [9] shown in Figure 2.4 generates S[0], S[1], C[1], C[1]' respectively. Structure of Block 3 and Block 4 enclosed in second dotted rectangle follow same structure as Block 1 and Block 2 respectively in first enclosed section. Figure 2.5(a), (b), (c), and (d) shows P, Q, R, S sub block structures respectively. Sum and carry are generated with inputs as A[0], A[1], B[0], B[1] through Block 1 and 2 using P, Q, R, S sub blocks. Sum and carry are selected either by primary carry input or by its

complimentary signal (C_{in} or C_{in}'). Second enclosed section with Block 3 and 4 generates sum and carry for next 2 bit addition and the signals wait at final stage multiplexers P1, Q3, R3, S3. These signals are selected by Carry out and its complimentary signal ($C[1]$, $C[1]'$) generated by first enclosed section with Block 1 and 2. Carry out signals ($C[3]$, $C[3]'$) generated by second enclosed section from Block 4 acts as carry input signals for the next section and this goes on to design an n (even) bit adder.

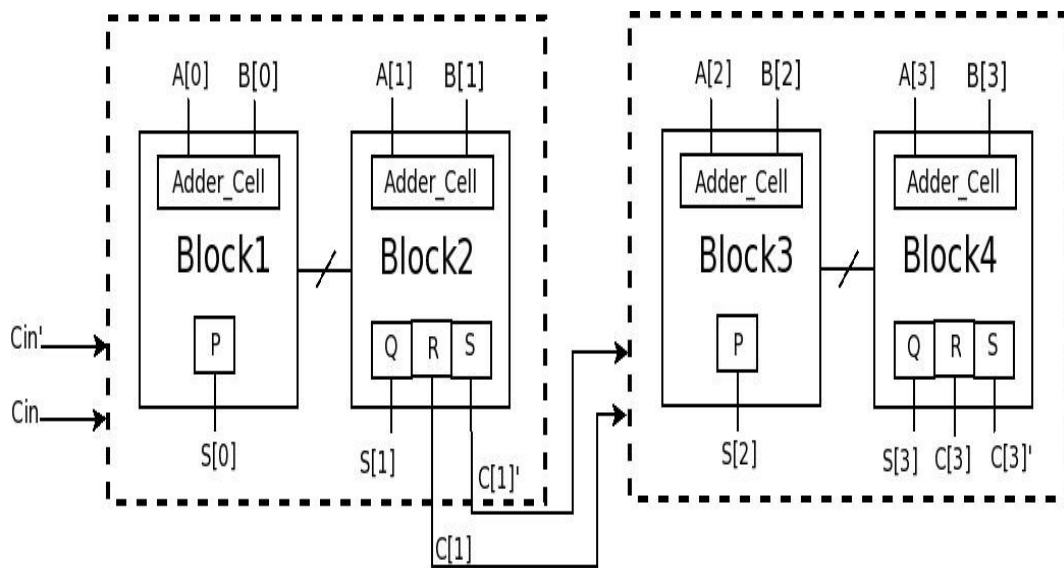


Figure 2.4 4-bit regular CPL adder [9].

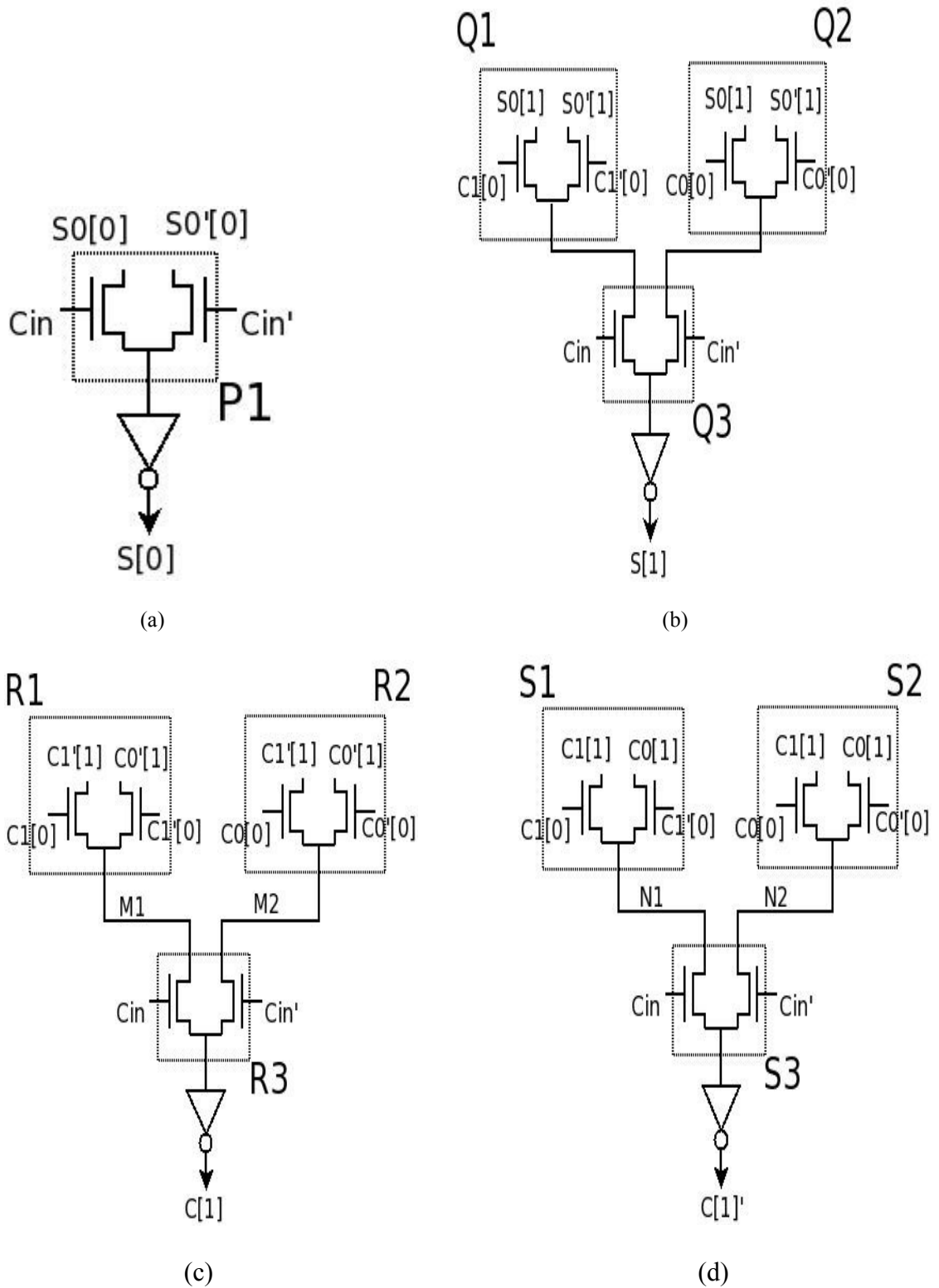


Figure 2.5 Internal block structure of regular CPL adder: (a) P-block, (b) Q-block, (c) R-block, (d) S-block.

2.2.2.2 16-Bit SQRT_CPL_CSA using RCA/BEC

16 bit SQRT CSA using 4 bit regular CPL adder yields high speed operation but it has more number of transistors and consume high power when comparing the results of work 3 and 4 with traditional SQRT CSA, work 1 and 2, at an activity of 100MHz as shown in Table 2.2. This is because of using two sets of regular CPL adders in regular design (or) one set of regular CPL adder and one set of CPL BEC block in design using BEC. Regular CPL adder [9] takes more number of transistors to design the functionality when compare with the traditional designs. Then this problem leads to designing of an area efficient, fast and low power design by modifying internal structure of 4 bit regular CPL adder to eliminate second block (RCA/CPLA/BEC) in carry select adder.

2.2.2.3 4-Bit Proposed CPL CS Adder

This proposed design concentrates on three specifications to reduce power delay area (PDA) product. This design shown in Figure 2.6 works on the principle of internal carry selection to get better results for design specifications. Block 1 has the similar structure as in regular design with only one modification of removing inverters enclosed in dotted rectangle R1 of adder cell 1 shown in Figure 2.1 and flip the sum signals. Removal of inverters leads to reduction in power consumption. Remaining output signals of adder cell are buffered through inverters because they need to drive the gates of multiplexers of next block (Block 2 in Figure 2.6).

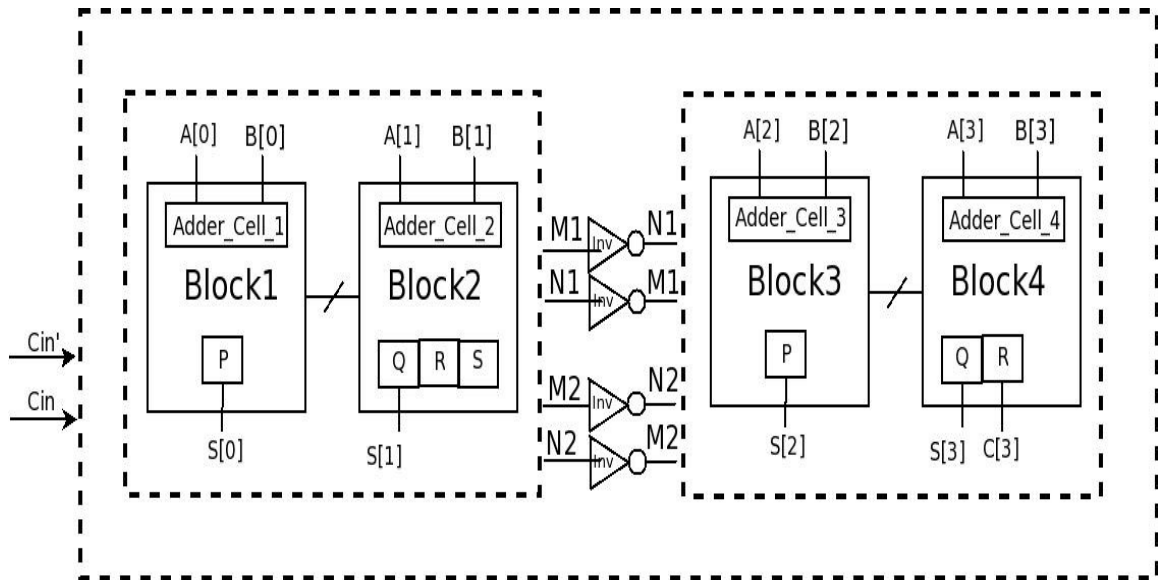


Figure 2.6 4-bit proposed CPL adder cell.

Block 2 plays a key role in reducing power and delay. Q, R, S sub blocks in Block 2 of modified design shown in Figure 2.6 follows the similar structure as regular design shown in Figure 2.4 except one modification for its sub blocks, i.e., R3 and S3 multiplexer blocks are removed. The key idea of designing internal carry selection involves assuming $Cin = 1$ to select N1 and M1 signals shown in Figure 2.7(a), (b) to act as Cout and Cout' from 2 bit addition performed by Block 1 and 2. Similarly, assumption of $Cin' = 1$ yields N2 and M2 as Cout and Cout'. All inverters at the output of adder cell 2 of modified design are removed to reduce the power consumption and output pin names are switched.

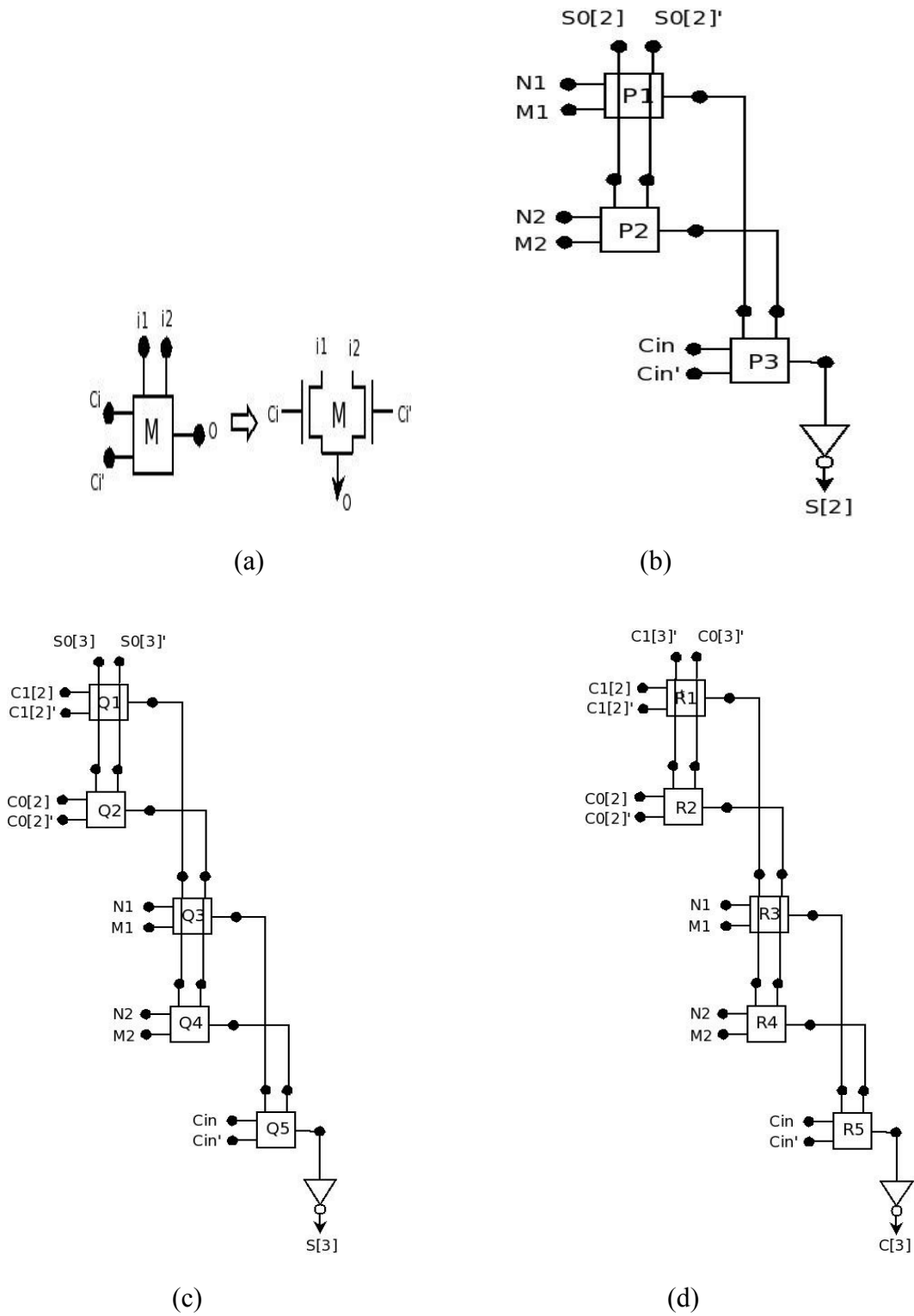
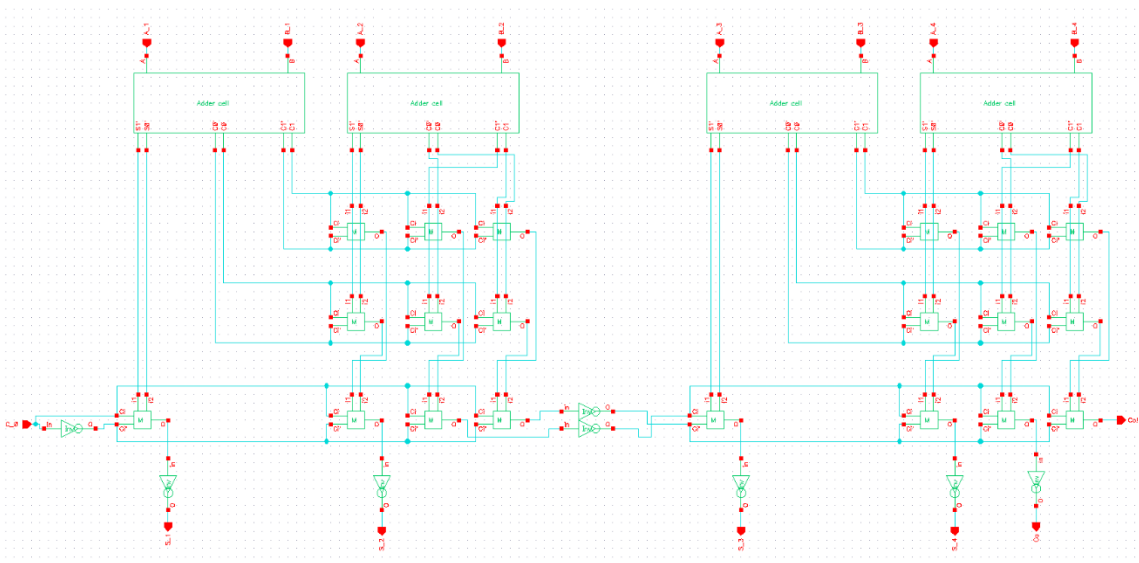
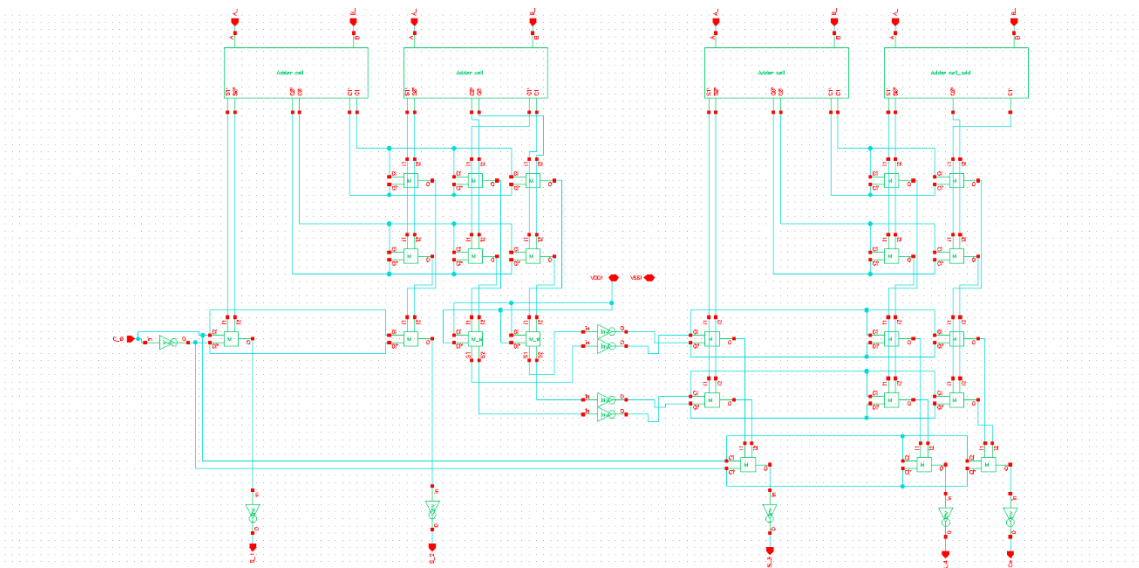


Figure 2.7 Internal block structure of modified CPL adder: (a) Mux, (b) P-block, (c) Q-block, (d) R-block.

The input signals (A, B and their compliment signals) pass through only 2 to 3 stages of series n MOSFETs, before they get buffered/regenerated by inverters shown in Figure 2.5(c), (d). By assumption of primary carry in as '1', N1 and M1 are chosen as carry in and its compliment signal to select sum and carry from Block 3 and Block 4 as shown in Q (Q3) and R (R3) sub blocks shown in Figure 2.7(c), (d). Similarly N2 and M2 act as a carry signals by assumptions of primary carry in as '0' (i.e. $C_{in} = 1$), which select the sum and carry signals of Block 3 and 4 using Q4 and R4 multiplexers shown in Figure 2.7(c), (d). Finally all sum and carry signals predicted by assuming carry in as '1' and '0' wait at P3, Q5, R5 multiplexers of Block 3 and 4. Immediately after getting carry from previous 4 bit adder block all out put signals, S[0], S[1], S[2], S[3], C[3], wait at P1, Q3 of first enclosed section and P3, Q5, R5 from second enclosed section respectively are selected at same time. As shown in Table 2.1, 4 bit modified CPL adder with inbuilt carry selection structure meets all three specifications of reducing power, area and delay of computations when compare with the regular design specifications. Figure 2.8 shows the schematic implementation of regular and modified/proposed 4-bit CPL adders in 250nm CMOS technology.



(a)



(b)

Figure 2.8 Schematic implementation of 4-bit CPL adders in 250nm CMOS technology (a) Regular (b) Proposed/modified.

Table 2.1 4-bit CPL adder results comparison at 500-Mhz activity

Type of Adder	Delay (ps)		Avg. power (mW)	# of Transistors (Avg. # of transistors per bit addition)
	Sum	C _{out}		
Regular_CPL	696	650	3.112	166 (~42)
Modified_CPL	675	405	2.685	148 (37)

2.2.2.4 16-Bit SQRT/Linear_CPL_CSA using proposed design.

4-bit Regular CPL adder design shown in Figure 2.4 generates carry out signals for every 2 bit addition performed in rectangular sections enclosed. So, it takes 8 carry ripples for 16 bit adder implementation. In proposed design all sum out signals and final carry out signals are selected at same time. So, 16 bit Linear CSA design using 4 bit CPL modified design takes 4 carry ripples as shown in Figure 2.9(a) and C_{out} signal is coming with just 405ps delay as shown in Table 2.1, which is better than regular CPL adder design. High speed carry out play a major role in reducing the delay but not sum. In case of 16 bit SQRT CSA as shown in Figure 2.9(b), this design does not use the proposed logic till 3rd block, which is a 3 bit CPL modified design, and in case of 5th block also this design uses the proposed technique till 4th bit addition after that it does not. Though SQRT CSA logic is more beneficial in reducing the delay of computation, because of 5 ripples needed for computation and all block level additions does not use proposed technique completely, SQRT design computation delay is more than linear design. In simulation results, delay is measured from C_{in} to worst case C_{out} or sum signal. Delay and average power for all 16 bit adders mentioned in this paper, in Table 2.2, are measured at an activity of 100MHz with 2.5V supply using spectre simulator on schematic designs in

0.25 μm CMOS technology in Virtuoso schematic editor. 10ns, 20ns, 40ns are the pulse periods for input signals A, B, Cin respectively with 0.1ns as rise and fall times.

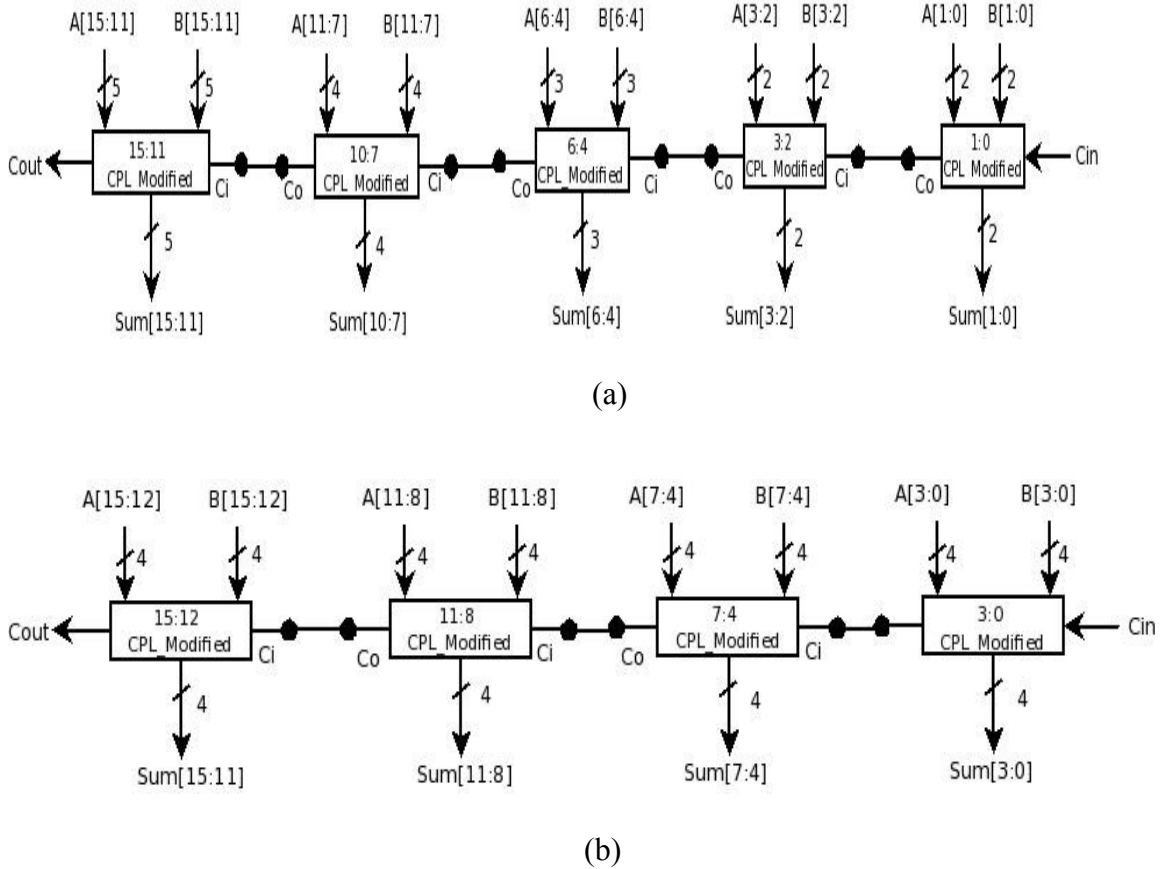


Figure 2.9 16-Bit CPL CSA using 4-bit proposed design: (a) SQRD, (b) Linear.

Table 2.2 16-bit SQRD-CSA results comparison at 100-Mhz activity

Work #	Type of Adder	Delay (ns)	Avg. Power (mW)	PDP (mW*ns)	# of Transistors	Avg. # of transistors per bit addition
1	RCA+RCA	2.54	8.643	21.95	974	~61
2	RCA+BEC	3.64	7.056	25.68	752	47
3	SQRD_RCA_CPL	1.58	10.9	17.22	1182	~74
4	SQRD_RCA_BEC_CPL	1.98	7.909	15.65	880	55
5	Proposed_SQRD_CPL	1.54	5.402	8.31	566	~35

6	Proposed_Linear_CPL	1.25	5.698	7.12	576	36
---	---------------------	------	-------	------	-----	----

2.3 Limitations for proposed 4-bit CPL CSA

Prediction circuit proposed has an ability to predict and select 4 bit addition outputs (four sum outputs and one carry out). Extending the prediction logic to find more than 4 bit addition results, with single carry in as selecting signal, needs large number (more stages) of multiplexers. As shown in Table 2, proposed 4 bit design holds no good delay values for 16 bit SQRT_CPL_CSA (work 5) adder when compare with 16 bit Linear CPL_CSA (work 6).

2.4 Results comparison

The proposed 16 bit SQRT or Linear designs saves significant number of transistors (area), power and reduces the delay when compared with regular CPL adder designs as shown in Table 2.2. From these results in table, proposed Linear CSA using modified CPL design has 18.83% less delay with an increase of 5.4% power and with slight increase in transistor count of 1.77% when compared with the results of proposed SQRT CSA using modified CPL. Either one of the designs can be selected according to the power and speed requirements. The proposed design in work 6 has 50.7% and 65.6% reduced delay, 34% and 19.24% reduction in power and 40.8% and 23.4% of reduced transistor count when compared with regular square root adder designs in work 1 and work 2.

3 A High Speed and Low Power 64-Bit Digital Comparator using 90nm CMOS Technology

3.1 Introduction

Binary comparator is an electronic device capable of performing an arithmetic operation of comparing two digital input signals. A simple single bit comparator compares two input digits, a digit is either logic '1' or logic '0', and yields three different possible outputs: "greater", "less" or "equal". A multi-bit comparator compares two multi-bit words. A multi-bit input digital comparator is widely used in computing and controlling devices, such as microcontrollers, microprocessors, digital image processors, encryption devices etc. Modern electronic computing devices are capable of working with binary word lengths of 32 bits (4 Bytes) and 64 bits (8 Bytes). Portability, computing speed and power efficiency are in great need for computing devices. A variety of comparator designs have been proposed to achieve the design specifications such as low power consumption, less delay (high speed) and less number of transistors (low cost and portability) [10]-[23].

A parallel tree structured 64-bit comparator using all-N- transistor (ANT) dynamic logic was proposed in [10] and demonstrated the improvements of performance and transistor count over conventional designs using domino sub-tractors; however, the 3.5 clock cycle pipelining process made the design less attractive for some applications. A priority encoder based comparator was first proposed in [11] to reduce the circuit complexity and demonstrated a significant cost improvement along with significant

enhancements in speed. Multiple output domino logic (MODL) implementation to decrease the power consumption and multi-level look-ahead technique to reduce the path delay was also implemented in [11]. High fan in dynamic logic implementation was proposed in [12] and demonstrated improvements in delay and transistor count over [10] and [11]. A high speed static design was proposed in [13] using 100nm CMOS technology. This static design can compete with the dynamic comparator designs for high-speed. In [14] parallel MSB checking with dynamic NOR gates was proposed to demonstrate the improvements in delay with area trade off over priority encoder based design proposed in [11].

An enhanced priority encoder and MUX based high fan in design was proposed in [15] using 0.35 μ m technology to demonstrate the improvement in speed of operation with an area trade off over [14]. A bit wise competition logic was proposed in [16] to demonstrate the improvement in delay and area over [10] [11] [17]. It was designed using less number of transistors (962) which shows a 38% improvement over a previous best design for transistor count proposed in [17]. Single clock cycle high performance designs were proposed in [18] [19] [20] using 90nm CMOS technology. Tree based single clock cycle comparators were proposed in [21] [22] [23]. Designs proposed in [21] and [23] were realized in 65nm CMOS technology. Constant delay logic to improve the speed of 64 bit radix-4 tree based comparator was proposed in [23].

In this work, a unique high speed, power and area efficient design for 64-bit static and dynamic comparator operation is demonstrated using 90nm-1.2V CMOS technology. Generally, dynamic designs consume more power than static designs. This work emphasizes the use of clock with reduced duty cycle to reduce the pre-charge (or

pre-discharge) time, which reduces the short circuit power consumption in dynamic designs, while increasing the evaluation time. Also, this work demonstrates a new approach for improving the self-pipeline nature of a digital design by adjusting the worst and best delays to be equal. The equal delay can be achieved using the pre-charging and sizing strategies discussed in Section 3.5.2.1. This work also introduces the design methodology for resource sharing. Some portion of the static or dynamic logic blocks, XE (XOR/XNOR) blocks described in Section 3.3, can be utilized to run other important arithmetic operations such as addition and multiplication. A similar approach of using XOR and XNOR to design comparator and adder was used in [19] and [24], respectively.

3.2 Operating Principal and Design Methodology

The proposed 64 bit comparator design implementation is not based on the traditional way of generating Boolean equations using truth table and K-maps. It is designed based on the general working principal of comparing binary data. This process is illustrated using Figure 3.1(a) where the MSB bits of Data A and Data B are not equal. Then the comparator ignores the comparison at rest of the bit positions. According to this principal, the comparator always progresses from MSB to LSB of a multi-bit binary data. Unlike some previously proposed designs with 3 outputs, AG, BG, and EQ, our proposed design has only two encoded outputs AG (or BG) and EQ, as shown in Figure 3.1(b). This method of design modification from regular 3-bit output to 2-bit encoded output, shown in Figure 3.1(b), is represented by logic functions 3.1, 3.2, and 3.3.

$$AG = A \cdot \overline{B} \quad (3.1)$$

$$EQ = \overline{A \oplus B} \quad (3.2)$$

$$BG = \overline{A} \cdot B \xrightarrow{\text{Modified}} BG = \overline{AG} \cdot \overline{EQ} \quad (3.3)$$

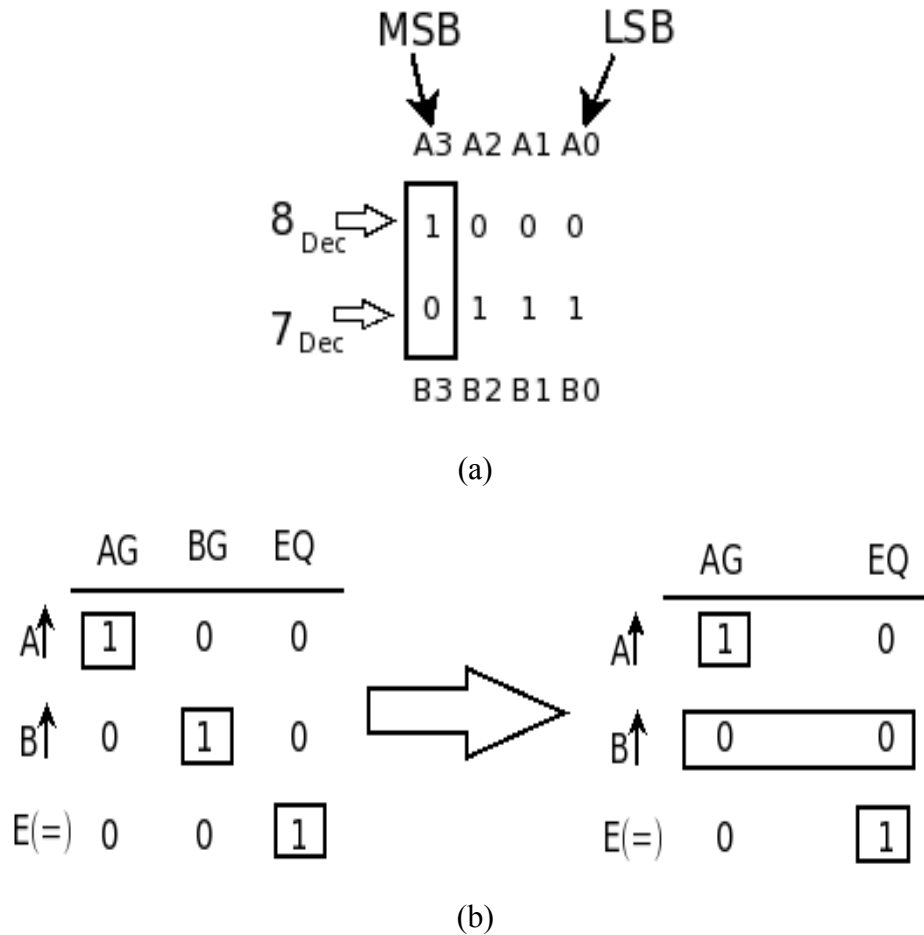


Figure 3.1 (a) Basic principal for binary data comparison, (b) Comparator design modification from traditional 3-bit output to encoded 2-bit output.

For an N-bit binary comparison, the comparator starts comparing from MSB bit, (N-1)th bit, and it proceeds to the next bit, (N-2)th bit, for comparison if and only if the MSB bits of two data are equal. As shown in Figure 3.2, this process continues until it gets an unequal (X) bit pair on its way of comparison towards LSB bit position. When it reaches the first unequal bit pair, it stops comparing the rest of the bits and yields an output of logic ‘1’ at AG (A Greater) or BG (B Greater). If both data are equal then it

yields logic '1' at EQ and logic '0' at both AG and BG. X and E logics used are realized by equations 3.4 and 3.5, respectively.

$$X = A (XOR) B \tag{3.4}$$

$$E = A (XNOR) B = \overline{X} \tag{3.5}$$

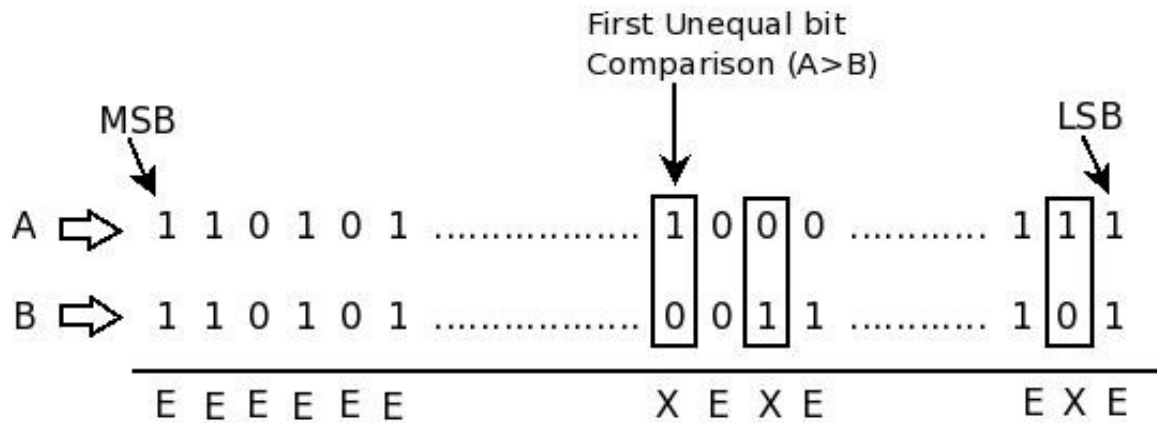


Figure 3.2 N-Bit Comparison

Implementation of the above design procedure using hardware requires three main sub blocks in different stages. One is XE block, second one is Level-1 comparison block, and the third one is Level-2 comparison block. XE block performs single bit comparison and gives outputs through X and E output pins, as shown in Figure 3.3, which is discussed in Section 3.3. Level-1 comparison block includes XE block to perform tier-1 comparison. Level-2 comparison block perform tier-2 to tier-M comparison. Here M, the total number of tiers, depends on comparison tree structure.

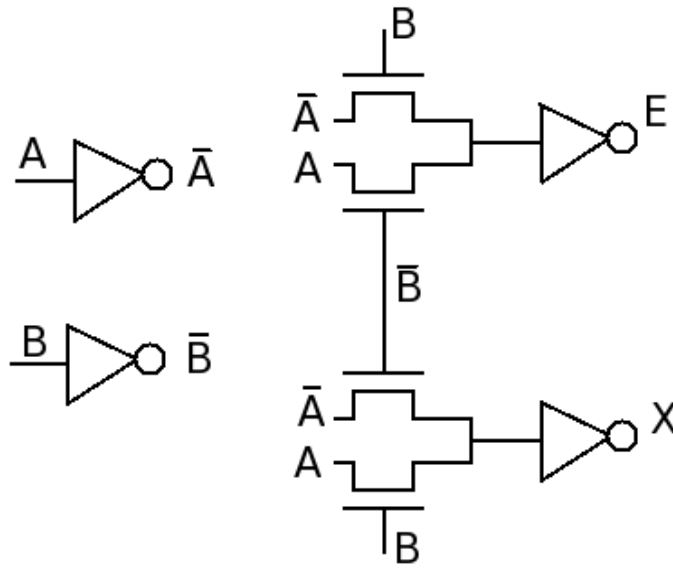
3.3 Static and Dynamic XE logic

The XE block takes A and B bits to generate X and E output signals. This XE block is designed in such a way that it yields logic '1' at X (XOR) if both A and B input

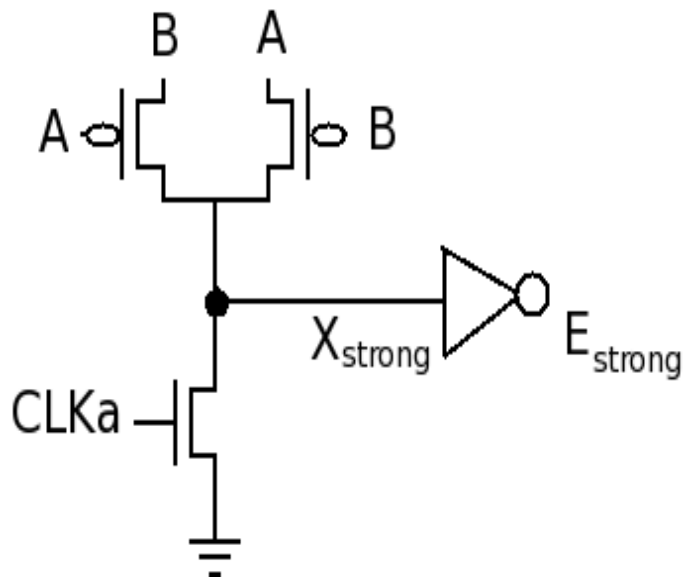
bits are unequal (0 1 or 1 0). If both A, B input bits are equal (0 0 or 1 1), then E (XNOR) takes a value of logic '1'. These are the key logic blocks which helps in designing many arithmetic designs. In microprocessor designing, this XE block can be used as a common logic block for resource sharing to implement other low power and high speed arithmetic operations. As shown in Figure 3.3(a) and (b), static XE block design using complimentary pass transistor logic requires 12 transistors [9], whereas, the proposed dynamic XE block needs 5 transistors. The proposed dynamic XE design resembles a static 5 transistor XOR-XNOR logic proposed in [25]. Static 5 transistor XE design in [25] is less attractive with high power consumption, especially in low frequency operations, and gives weak logic. Proposed dynamic XE block operating functions are illustrated using equations 3.6 and 3.7.

$$X_{strong} = (A \cdot \bar{B} + \bar{A} \cdot B) \cdot \overline{CLK_a} \quad (3.6)$$

$$\begin{aligned} E_{strong} &= \overline{CLK_a \cdot (A \cdot B + \bar{A} \cdot B)} \\ &= \overline{CLK_a \cdot \overline{A \oplus B}} \\ &= \overline{CLK_a \cdot \overline{X_{strong}}} \end{aligned} \quad (3.7)$$



(a)



(b)

Figure 3.3 XE block (a) 12T-static [24], (b) 5T-dynamic.

This proposed 64 bit comparator needs 64 XE blocks. More than half of the area and power of the 64-bit static comparator are utilized by XE blocks. To reduce the power consumed by dynamic block design, the proposed design approach uses a clock with reduced duty cycle to decrease the pre-discharge/pre-charge time. Reduction in pre-discharge/pre-charge time reduces the short circuit current while increasing the evaluation time. Dynamic XE block with traditional 50% duty cycle clock, as shown in Figure 3.4(a), causes a high power consumption when clock is high and two input signals are opposite in logic. In this case, one of the p-MOS transistors in pull up network and clocked n-MOS transistor in pull down network go to ON state. Then, there is a continuous flow of short circuit current between VDD and GND rails, which is a major contribution for power consumption in dynamic designs. In effort to improve the power efficiency and speed of operation, we performed the simulations on dynamic XE block with reduced duty cycle (D), which is shown in Figure 3.4(b). This proposed method yields an excellent improvement in power and delay reduction, as summarized in Table 3.1, with 25% and 10% duty cycle clock. All results obtained are based on simulations performed in Cadence Analogue Design Environment (ADE) on designs using standard V_t transistors in 90nm CMOS technology.

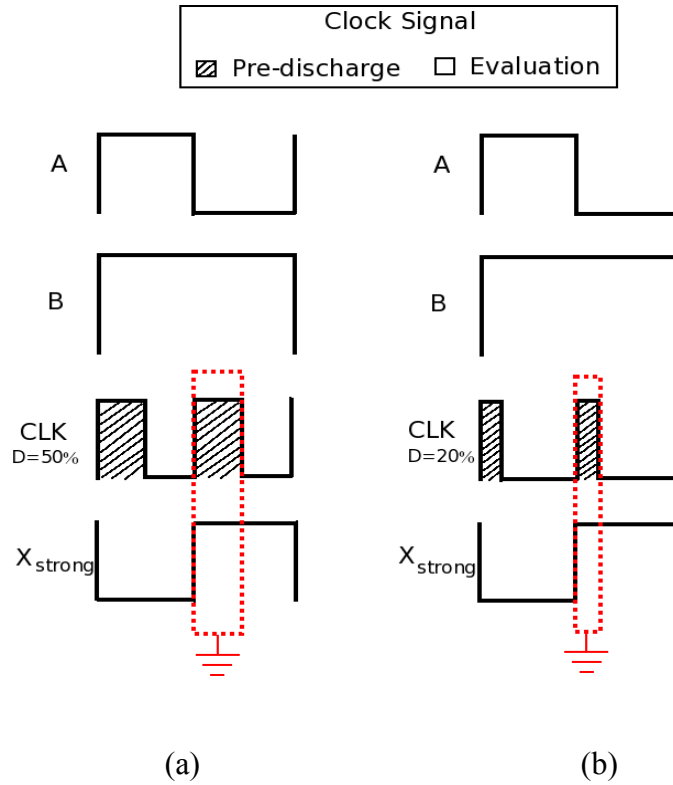


Figure 3.4 Dynamic XE block performance using (a) traditional clock (D=50%) and (b) clock with reduced duty cycle (D<50%).

As shown in Figure 3.4, simulations on static and dynamic XE designs are performed using 1.0GHz and 500MHz pulse signals as Data A and Data B, respectively. For dynamic designs, clock frequency is double the frequency of fastest input signal. As summarized in Table 3.1, at 2GHz and 10% duty cycle clock with 1.2V and 1.0V supply, average power consumption by dynamic design is nearly equal to the average power consumption by static design for the same vector. Reduction in duty cycle of the clock from 50% to 10% also increases the evaluation time for the dynamic design. Pre-charging/pre-discharging strategies, discussed in Section 3.5, used in designing Level-1 and Level-2 dynamic sub-blocks explains the improvement in the overall speed of the proposed 64-bit full dynamic comparator design while reducing the transistor count.

Table 3.1 Simulation results of static and Dynamic XE blocks

Mode	Activity/clock @VDD	Duty Cycle	Avg. Power (μ W)	Delay (ps)	# of Transistors
Static	1GHz@1.2V	N/A	9.37	39	12
	1GHz@1.0V	N/A	5.85	80	
Dynamic	2GHz@1.2V	50%	38.53	14	5
		25%	21.24	14	
		10%	10.83	14	
	2GHz@1.0V	50%	19.83	22	
		25%	11.26	22	
		10%	5.99	22	

3.4 Design of proposed 64-bit static comparator using Level-1 & Level-2 sub blocks

3.4.1 Level-1 4-Bit Static Comparator Sub Block Design

The proposed 4 bit comparator works on the previously mentioned design principal illustrated in Figure 3.1 and Figure 3.2. As shown in Figure 3.5, Level-1 (L1) 4-bit comparator sub block needs 4 ‘XE’ blocks and 3 chains of transistors for performing the Greater and Equal operations. This 4 bit block clearly shows proposed logic for comparison between two 4 bit words A and B. First, all XE blocks perform comparison at the respective positions in parallel and generates outputs X (bits are not equal) and E (bits are equal). Then, the actual 4 bit word magnitude comparison takes place using Chains 1,

2, and 3. Chain 1 and 2 generates output at AG (A greater) and Chain 2 and 3 generates output at EQ. Chain 2 is a common logic generator for both AG and EQ outputs.

As mentioned in Section 3.2, proposed design compares the first unequal bits of A and B to check which word is greater. If any bit pair is unequal then it sets logic '1' on X which sets and pass through the respective parallel transistor in chain 3 and sets a weak logic '1' at node N2. This weak logic '1' sets a strong logic '0' at EQ. If some intermediate unequal bit pair is found, then the first unequal bit pair sets X to logic '1' and E to logic '0' at that position. This active low signal at E breaks the n- MOS chain to ignore the comparison results at rest of the bit positions. The AG and EQ logic functions for Level-1 sub-block are developed in 3.8 and 3.9 respectively.

$$AG = \overline{(X_3B_3) + E_3(X_2B_2) + E_3E_2(X_1B_1) + E_3E_2E_1(X_0B_0) + E_3E_2E_1E_0} \quad (3.8)$$

$$EQ = \overline{X_3 + X_2 + X_1 + X_0 + E_3E_2E_1E_0} \quad (3.9)$$

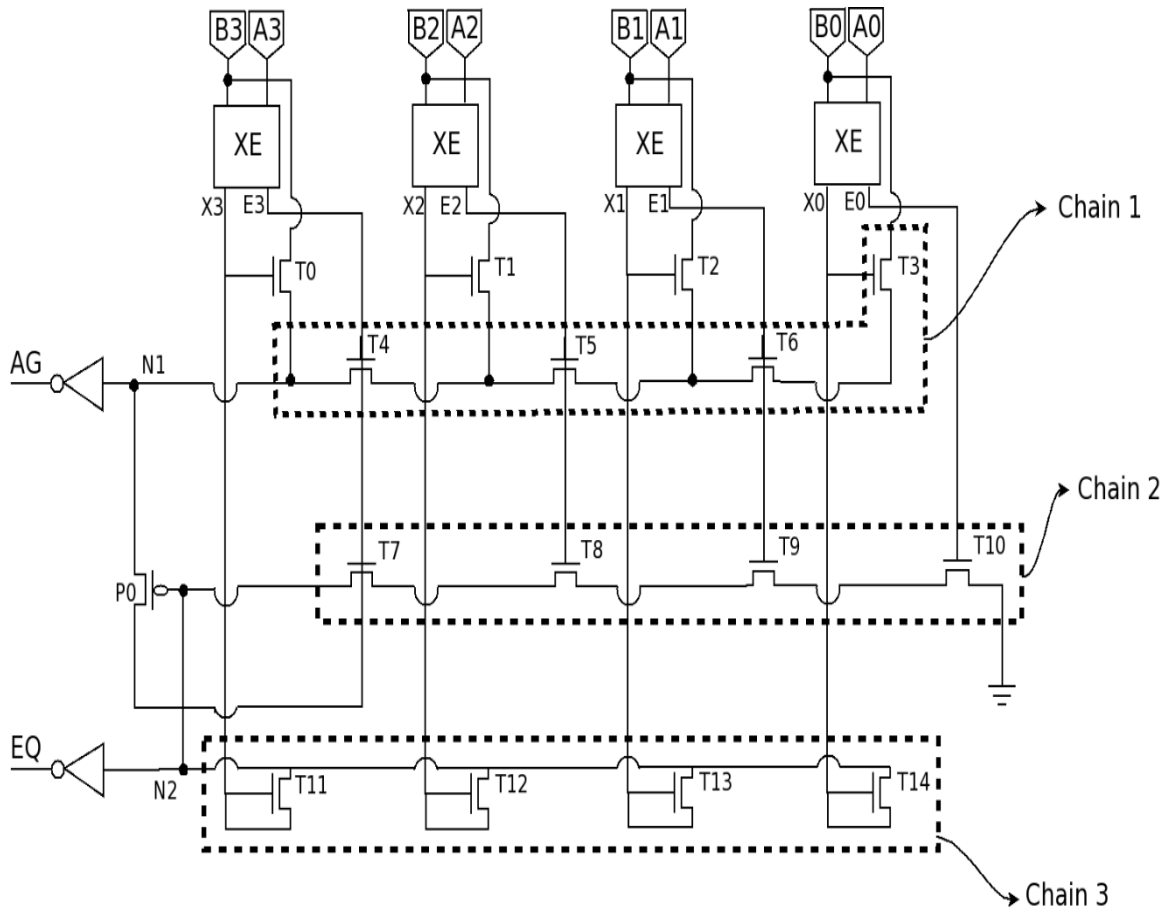


Figure 3.5 Level-1 static 4-Bit comparator sub-block

For instance, if MSB bits of A3 and B3 are not equal (1 0 or 0 1), then outputs of XE block, X3 and E3, set to logic '1' and logic '0' respectively. Then, logic '0' at E3 breaks rest of the Chain 1 by setting the transistor T4 to OFF mode and logic '1' at X3 sets the transistor T0 to ON mode. As shown in Figure 3.5, one end (drain) of the N-MOS transistor T0 is connected to input bit B3. In this example, if bit B3 is logic '0', then this implies that A3 input is logic '1' and the word A is greater. So, logic '0' from B3 pass through transistor T0 and sets logic '0' on node N1. Then, this logic '0' gets inverted at inverter 1 to yield logic 1 at AG.

Chain 2 performs operation in one of the worst cases that is when all bit pairs are equal. At this case, all E outputs from XE blocks sets to logic '1', which sets all transistors in chain 2 to ON state. Then, chain 3 passes a strong '0' to node N2 either from ground or from X3. Strong '0' on N2 sets two outputs AG and EQ at same instance. Logic '0' at node N2 passes through inverter 2 to set logic '1' at EQ. It also sets the P-MOS transistor P0 to ON mode, which sets logic '1' on node N1 by passing logic '1' from E3, which in turn sets an output of logic '0' on AG. Most actions performed by chains 1, 2 and 3 do not need any ground (VSS) or power (VDD) connections except for substrate connections. This type of designing reduces the power consumption by stack effect. Only one ground connection is used by T10 to pass logic '0' though chain 2. Here there is an alternative to avoid this ground connection by connecting the free end of the T10 transistor to X0. It works well because chain 2 function only if all E's are logic '1' and in this case all X's set to logic '0', which is helpful to avoid the ground connection with an increase in delay. It has a design flexibility that switching connections between transistors (T0, T1, T2, T3) and input bits from B (B3, B2, B1, B0) to input bits from A (A3, A2, A1, A0) switches output AG to work as BG.

3.4.2 Level-2 4-bit Static Comparator Sub-block Design

For multi-bit (N-bit) input data comparison, single tier comparison using Level-1 yields outputs with large delay at worst case comparisons. This is because of N number of series transistors in Chain 1 and Chain 2 for single tier 64 bit comparator architecture. Though the single tier architecture seems to utilize low power and less number of transistors, it does not meet challenging speeds with single tier architecture. So, the design needs multi-tier architecture.

Level-2 comparator sub blocks are useful in designing different multi-tier architectures. We can observe the clear resemblance of operation between Level-1 and Level-2 sub blocks. Level-1 (L1) compares input bits from MSB to LSB and in a similar way Level-2 (L2) compares outputs of Level-1 sub blocks from most significant block to least significant block. As shown in Figure 3.6, Level-2 sub block uses 4 pairs of EQ and AG outputs from 4 Level-1 sub blocks as inputs and generates BG and EQ as outputs. Level-2 sub-block also has 3 chains to perform the greater and equal operations. Chain 1 and 2 performs operations to contribute to the BG output. Chain 2 and 3 contribute to the EQ output. As discussed earlier, both Level-1 and Level-2 blocks yields inverted outputs. In a 64-bit comparator, if the architecture is designed with odd number of tiers (or stages) and if transistors T0, T1, T2, and T3 in Level-1 sub block are connected to input bits of B, then the output will be AG. And, if the input connections are flipped, from B to A, then the output of the comparator switches from AG to BG. This consideration for outputs will be altered for comparator with even number of stages. BG and EQ logic functions for level-2 are developed in 3.10 and 3.11, respectively.

$$BG = \overline{(\overline{EQ_3}AG_3) + EQ_3(\overline{EQ_2}AG_2) + EQ_3EQ_2(\overline{EQ_1}AG_1) + EQ_3EQ_2EQ_1(\overline{EQ_0}AG_0) + EQ_3EQ_2EQ_1EQ_0} \quad (3.10)$$

$$EQ = \overline{\overline{EQ_3} + \overline{EQ_2} + \overline{EQ_1} + \overline{EQ_0} + EQ_3EQ_2EQ_1EQ_0} \quad (3.11)$$

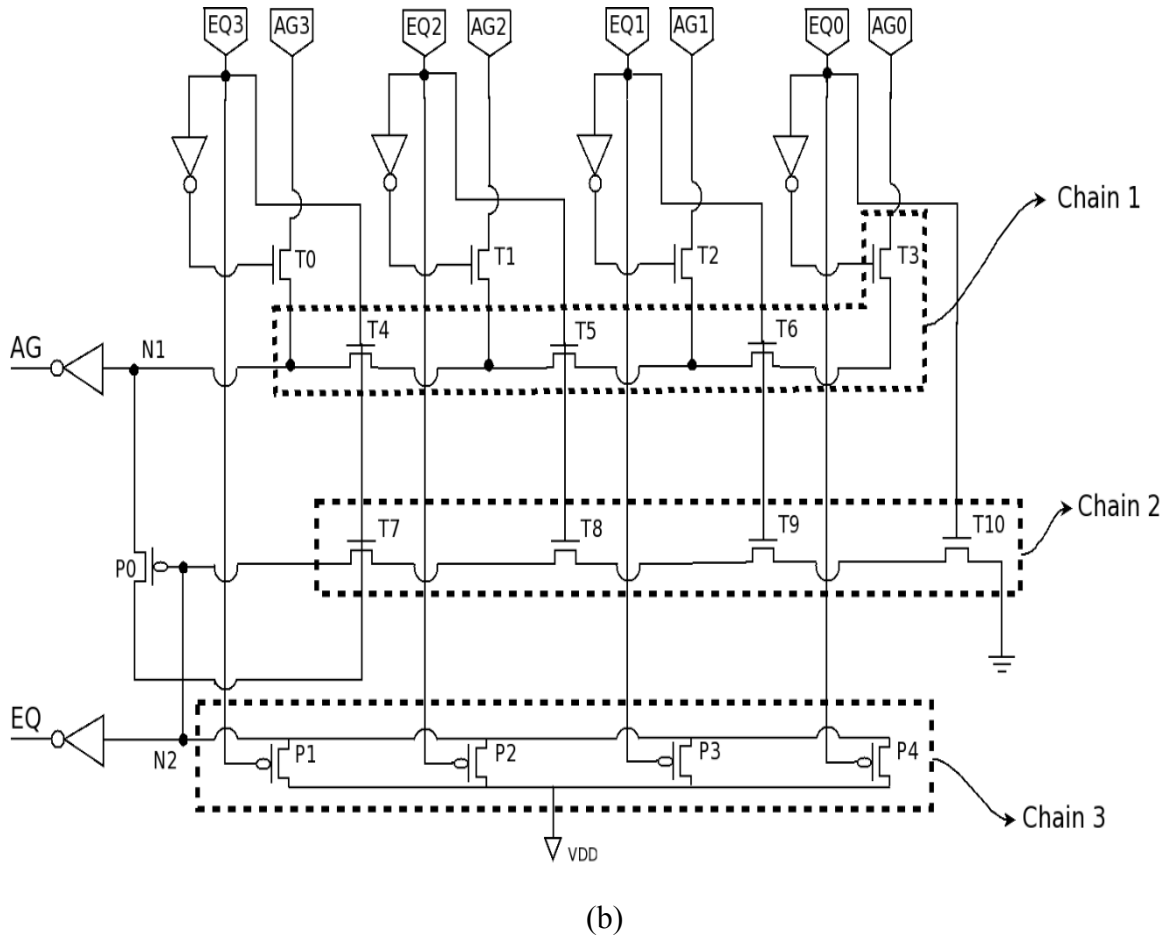


Figure 3.6 Level-2 static 4-Bit comparator sub-blocks

3.4.3 Delay and Power Optimization Strategies

Low power and high speed are the objectives of this proposed design. Analyzing Figure 3.5 and Figure 3.6, the worst delay path is Chain 1 when only the LSB pair bits are not equal. The delay path is T3, T6, T5, and T4. Two techniques are used to optimize this worst path delay. One is adjusting the transistors sizes; another is balancing the load. Increase Chain 1 transistor widths progressively from MSB side, to worst case input side, LSB side, to reduce Chain 1 delay. Chain 1 and Chain 2 structures are similar. Both chains are stacked ones. But, Chain 2 is designed with minimum width transistors.

So, delay optimization procedure for both blocks is similar. The simulations are performed on different low cost sub-blocks and results are tabulated in Table 3.2. Low cost design is intended for smaller area and maximum width of transistor used in designing is 480nm. As shown in Table 3.2, Level-1 and Level-2 low cost sub-block delays are 130ps and 123ps, respectively.

Table 3.2 Simulation results for L1 and L2 static comparator sub blocks at an activity (Data rate) of 500MHz (1GHz).

Design	No. of Transistors	Low Cost			
		Power (μ W)	Delay (ps)		Max. width of transistor
			EQ	AG/BG	
Level-1	68	28.21	130	117	480nm
Level-2	28	5.99	57	123	480nm

Load balancing is another important factor in delay optimization of 64-bit tree based comparator. Outputs from Level-1 sub blocks in tier-1 has to drive the gates of Level-2 sub block in tier-2 and then Level-2 sub block outputs drive Level-2 sub block in tier-3 and this process goes on till the last tier. EQ output signal from Level-1 has larger load to drive than AG. So, to set the overall effective delays of final output signals nearly equal, by considering loads, delay of EQ signal is made less than the AG/BG signal in Level-2 sub-blocks.

3.4.4 Tree Based 64-bit Low Cost Static Comparator Design

In this work, a 64-bit static comparator is designed in radix-4 tree structure using the proposed 4-bit comparator design. Radix-4 is a 3 tier architecture, shown in Figure 3.7. There are other tree based structures like radix-2 and radix-8. The number of stages (2) needed to implement a 64-bit comparator in radix-8 is smaller; however, the

stack height (8) is too high for delay optimization. Radix-2 structure is also not good for delays, because it is a 6 tier structure with 12 transistors and 6 inverters in its worst path. In case of radix-4 design, worst path from LSB side includes 12 blocks of delay and best path from MSB includes 3 blocks. As shown in Figure 3.8, radix-4 structure has 13 transistors and 5 inverters in worst path, and 4 transistors and 5 inverters in best path. Delays of signals coming from all paths should be nearly equal to maintain the self-pipelining of the design. This design needs 4 buffers at 'Bg' outputs shown in Figure 3.7 from stage 2 for worst case delay improvement.

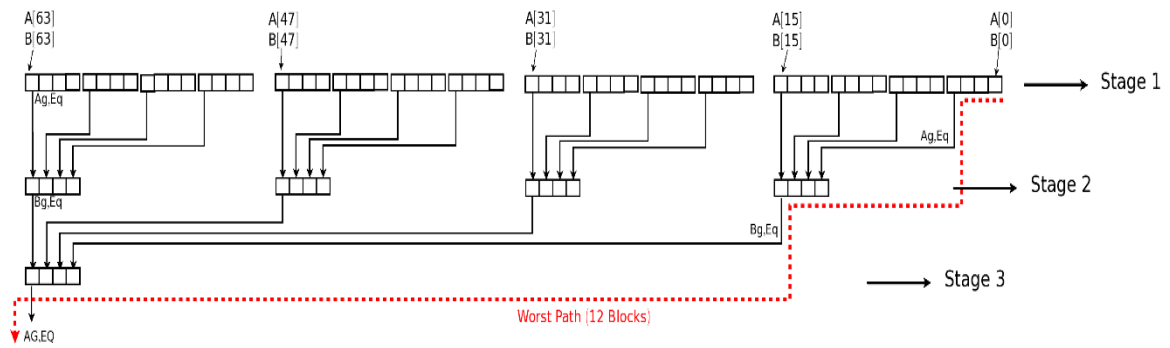


Figure 3.7 Proposed 64-bit static comparator using radix-4 tree structure

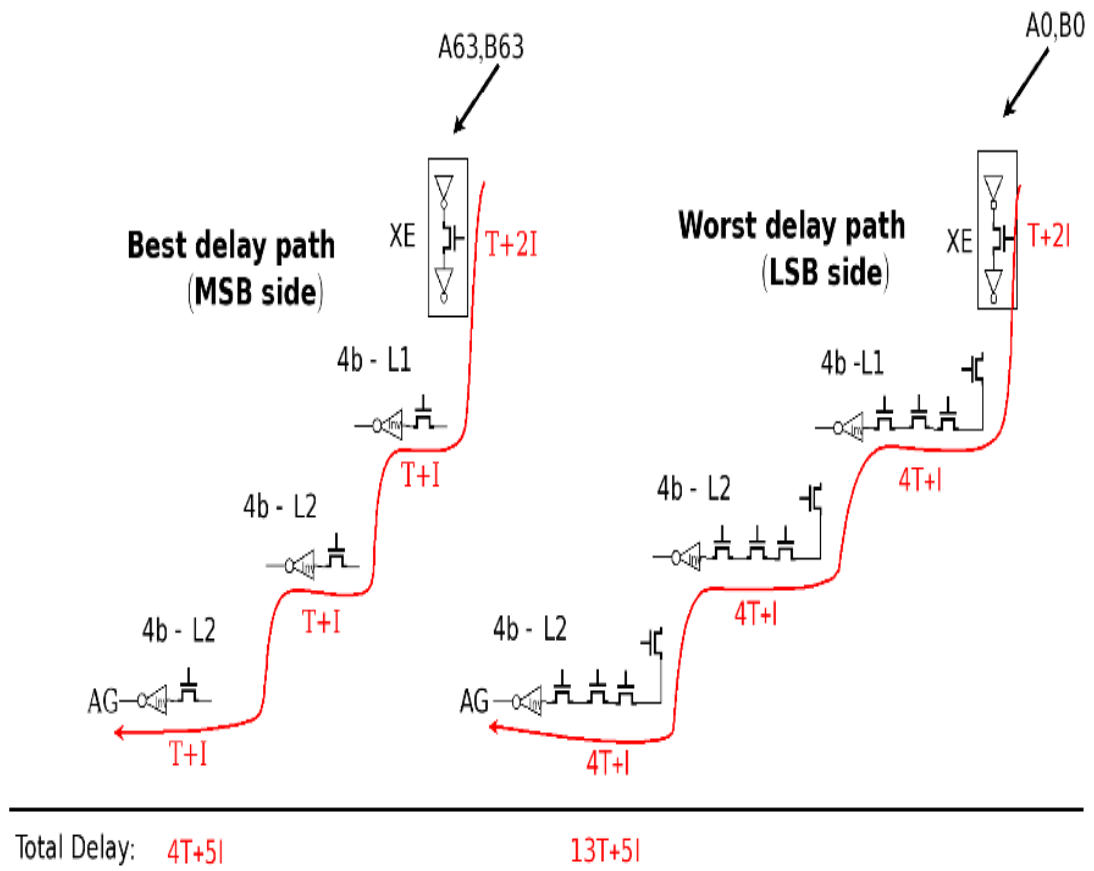


Figure 3.8 Worst and best path delay representation in radix-4 structure.

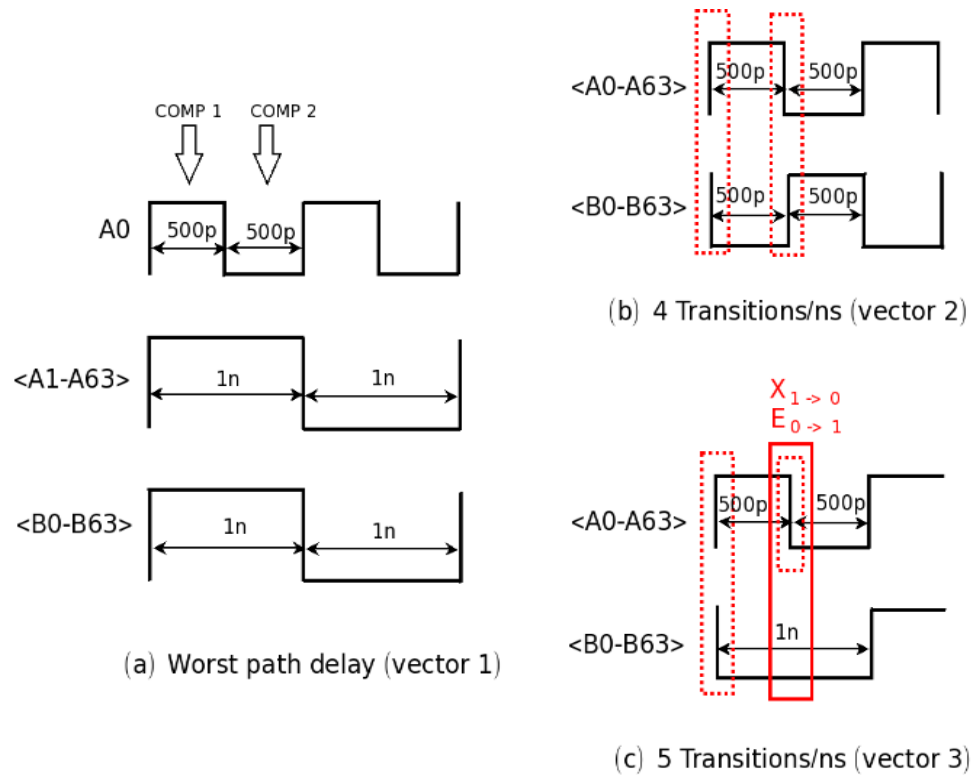


Figure 3.9 Input vectors to measure (a) Worst path delay, (b) Worst power, and (c) Maximum power consumption.

3.4.5 Schematic Simulation Results

Average power at worst case delay vector, worst case delay, worst power and maximum power consumptions are measured by triggering input vectors, shown in Figure 3.9, in Cadence Spectre simulator. As mentioned in Section 3.3, simulations are performed on designs using standard- V_t transistors. In low cost mode, worst case path delay of 64-bit comparator using vector 1 is 374ps and average power consumption at this vector is 232 μ W. Maximum power consumption of 822 μ W was measured by triggering vector 3. With input vector 3, the design consumes maximum power (more than when triggering by vector 2). In this proposed design, XE block is the major power consuming block. This proposed 64 bit comparator has 64 XE blocks and the maximum

power consumption was measured by triggering maximum activity involving all XE blocks. As shown in Figure 3.9(c), vector 3 causes a maximum of 5 transitions per 1 ns period at each XE block. But vector 2, which seems to be the worst power vector, triggers 4 transitions per 1ns period at XE block. With input vector 2, worst average power consumption is 647 μ W, which is less than the power consumption, 822 μ W, caused by triggering vector 3. The number of signal transitions in XE block is calculated using the formula in 3.12.

$$\frac{\# \text{ of Trans.}}{1 \text{ ns}} = \frac{2}{T_A} + \frac{2}{T_B} + \frac{\# \text{ of Trans. on } (X + E)}{1 \text{ ns}} \quad (3.12)$$

Where T_A & T_B are periods for A and B signals in ns.

Switching frequency can also be represented by activity factor “ α ” which is estimated by using 3.13 [23].

$$\alpha = \frac{\# \text{ of signal transitions}}{\# \text{ of input signals} \times \# \text{ of clock cycles}} \quad (3.13)$$

3.5 Proposed 64-bit dynamic comparator design using Level-1 & Level-2 sub blocks

3.5.1 Level-1 and Level-2 4-Bit dynamic sub block design

Dynamic sub blocks are designed by modifying static sub-blocks. This section demonstrates the dynamic low cost radix-4 tree structured 64 bit comparator design using reduced duty cycle clock. In this work, clock signal is given in stimuli of the simulator. Measurement of average power, delay and transistor count do not include clock tree. This

work demonstrates mainly the impact of proposed approach of using the non-traditional clock with reduced duty cycle. The Radix-4 64-bit dynamic design is similar to static design with some internal modifications in Level-1 and Level-2 sub-blocks. Both Level-1 and Level-2 dynamic sub blocks are designed for low cost purpose where the maximum width of transistor used is 480nm. Level-1 sub block is modified from static to dynamic by adding dynamic XE block in place of static XE block to reduce the transistor count.

As shown in Figure 3.10, a clock gated n-MOS transistor, T7, and a p-MOS transistor, P5, are added at node N2 of chain 3 and node N1 of Chain 1 respectively to pre-discharge and pre-charge; pre-discharge on node N2 substitute the functionality of chain 2; pre-charge on node N1 reduces the worst case delay through chain 1. Level-2 dynamic sub-block can be designed in a similar way as Level-1 dynamic sub-block designed to reduce the transistor count. But, that makes the EQ signal quicker than that of AG/BG output signal. To set the AG/BG signal delay and EQ signal delay equal, to retain the self-pipelining, Level-2 dynamic comparator sub block is designed by adding a clocked p-MOS pre-charge transistor, shown in Figure 3.11, at node N1 on Chain 1 of a low cost sub-block. In this work, self-pipelining is a special tendency of the design to yield the synchronized outputs without the need for adding any D Flip Flop (DFF). To improve the self-pipelining nature of the design, AG/BG output signals and EQ signals of sub blocks should come with same delay. Worst, moderate and best case signal propagation delays should be close. Maximum width of the transistor used in designing low cost comparator sub blocks is 480nm, which is discussed in Section 3.4.3.

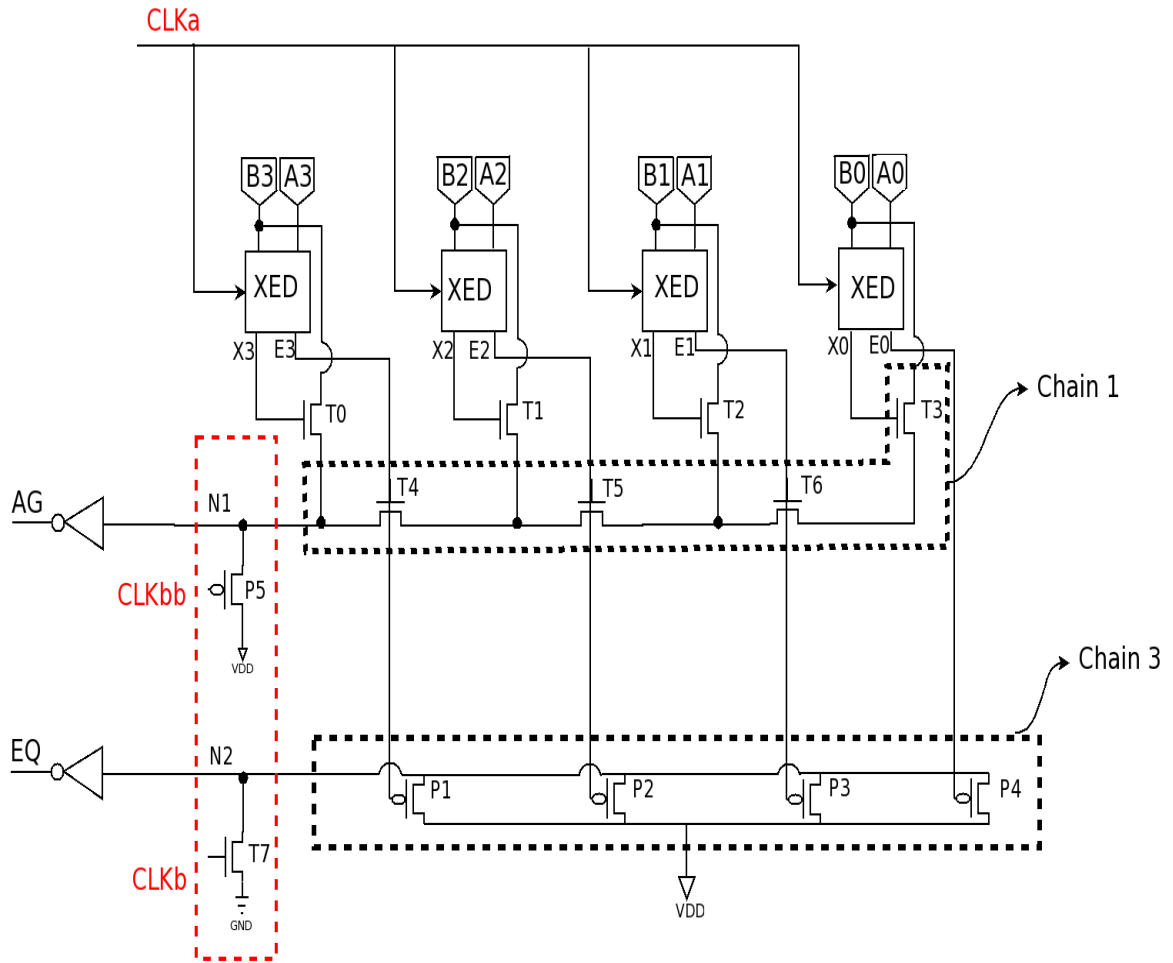


Figure 3.10 Level-1 dynamic 4-bit comparator sub-block

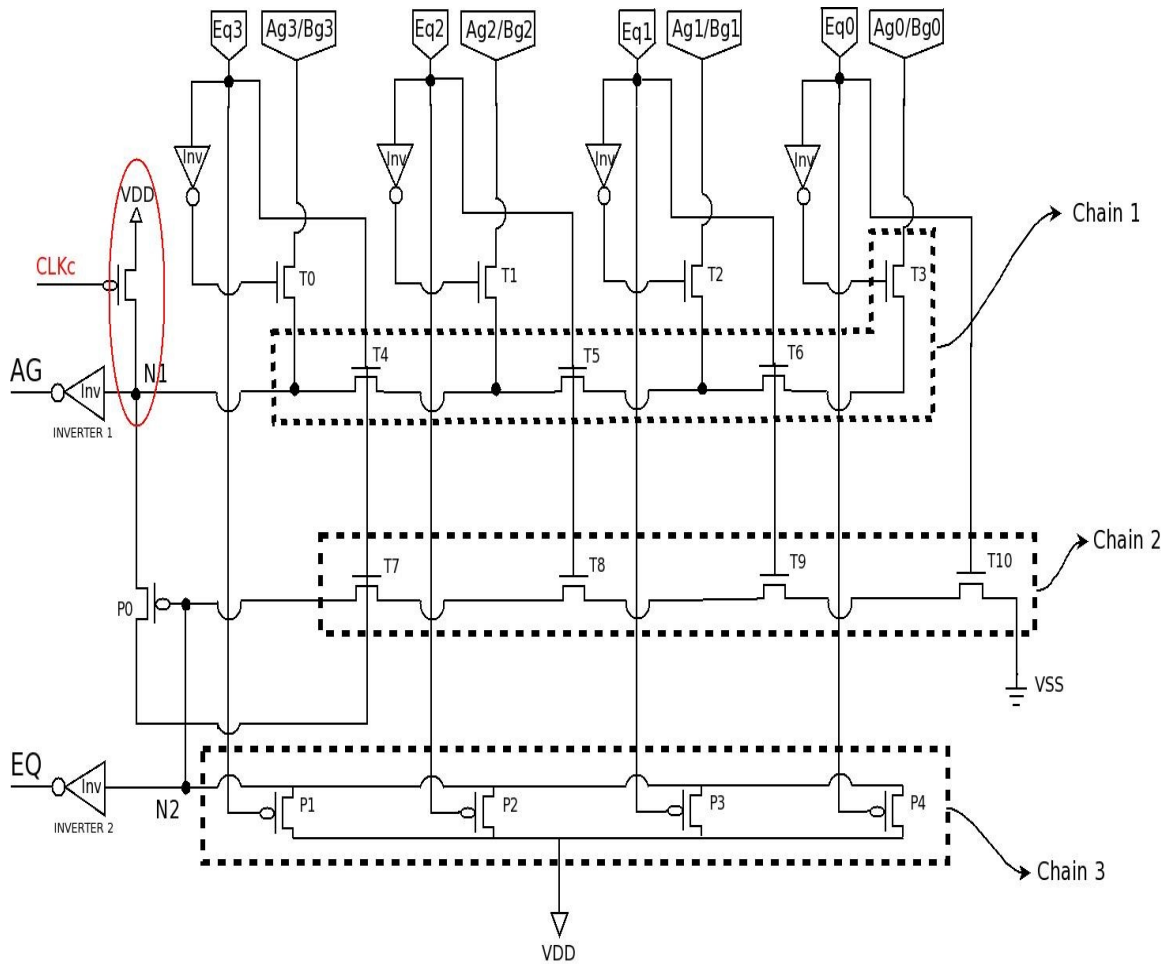


Figure 3.11 Level-2 dynamic 4-bit comparator sub-block

3.5.2 Proposed 64-bit, low cost, full dynamic, self-pipelined comparator design using radix-4 structure

Though the proposed dynamic and static designs are designed using similar structure, dynamic designs need more effort to make the design operate with self-pipelining. Level-1 and Level-2 sub blocks used in static and dynamic designs are of similar size. Duty cycle of clock and sizing strategy for sub blocks at different levels plays a major role in improving the pipelining of the proposed design.

3.5.2.1 Sizing and Pre-charging Strategies:

Worst path and best path delays are made approximately similar by reducing the transistor widths in best path, when unequal bit pairs occur in MSB position. Initially, worst path delay is optimized and then signal delays from best delay paths increased to match the worst delay. This proposed approach helps improving the self-pipelining nature of any tree structure. In a sub-block design, widths were increased progressively from MSB side to LSB side.

Second important factor in improving speed and self-pipelining is proper arrangement of pre-charging transistors. In Level-1 dynamic sub-block design, N1 and N2 nodes are pre-charged quickly to increase the speed through worst path. But, in second and third stages, chain 3 does not need any pre-charge transistor. EQ signal passing through second stage should come with moderate delay. If Level-2 sub-block also has a pre-charge transistor, EQ signal comes first to the last stage and causes invalid logic. Reduced duty cycle clocks strategy is proposed to eliminate the pre-charge overlap issues. As shown in Figure 3.12, reduced duty cycle clocks are given at different stages from CLK_a to CLK_c to improve the power efficiency while maintaining its speed.

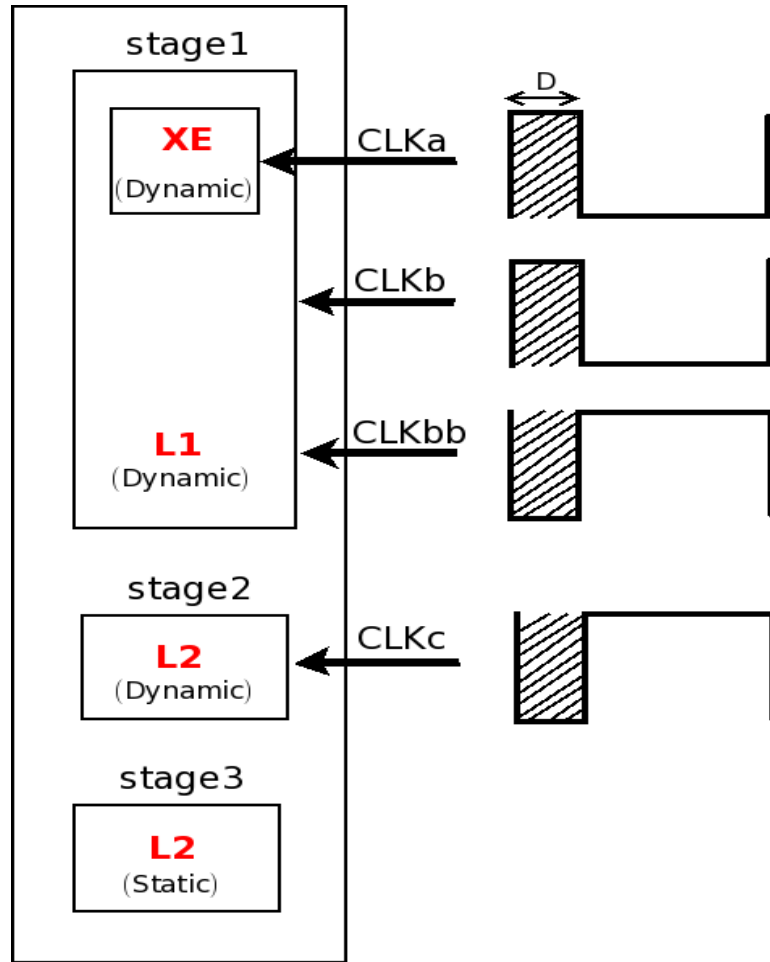


Figure 3.12 Clock strategy for proposed radix-4 64-bit full dynamic comparator design

3.5.3 Proposed 64-bit partially dynamic comparator design in radix-4 structure

In an attempt to further reduce the power consumption at high speeds of operation, proposed dynamic design (full dynamic) is modified as a partially dynamic design. In full dynamic design, maximum power is consumed by dynamic XE block (XED). In order to reduce the power consumption, dynamic XE block (5 transistors) belong to Level-1 sub-block of stage 1 is replaced by static XE block (12 Transistors),

which increases the transistor count. Full dynamic design requires 64 clock inputs to drive 64 dynamic XE block, where partial dynamic design require no clock inputs to drive XE logic. So, partial dynamic design requires smaller clock tree. To improve the speed of AG/BG and EQ signals, Level-1 and Level-2 sub-blocks in stage 1 and stage 2 are given clocks CLK_b , CLK_{bb} , and CLK_c as shown in Figure 3.13.

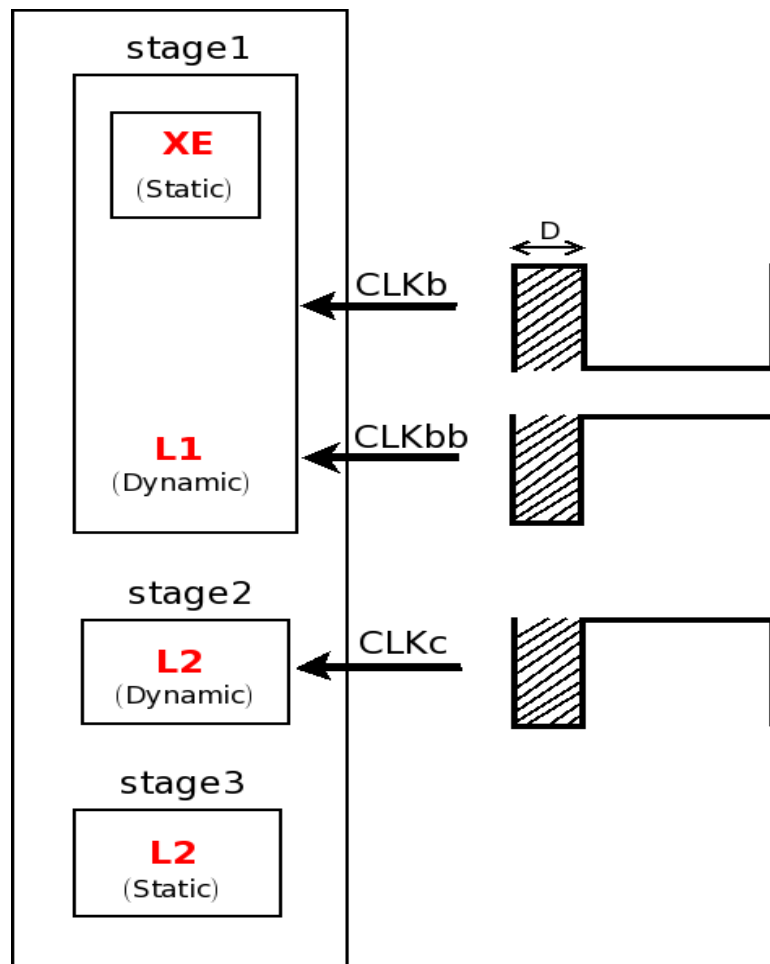
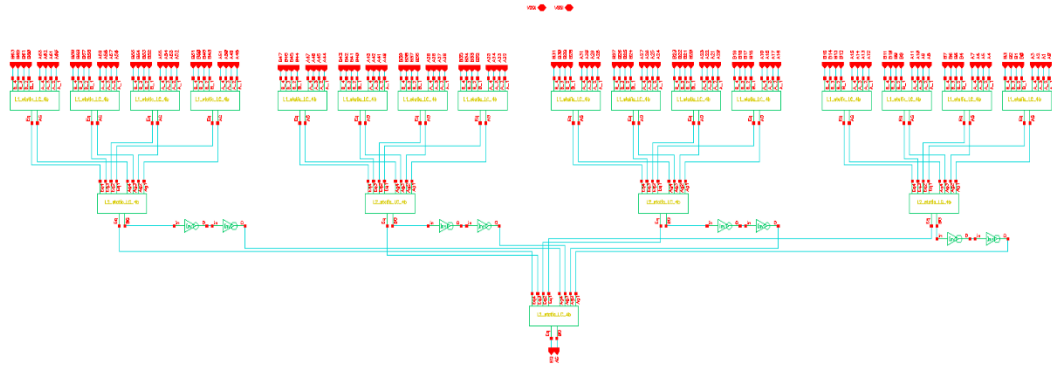


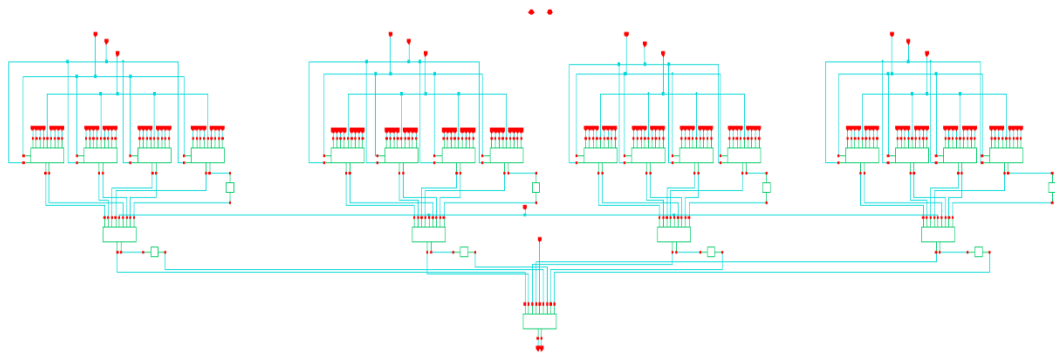
Figure 3.13 Clock strategy for proposed radix-4 64-bit partially dynamic comparator design

3.6 Simulation Results and Discussion

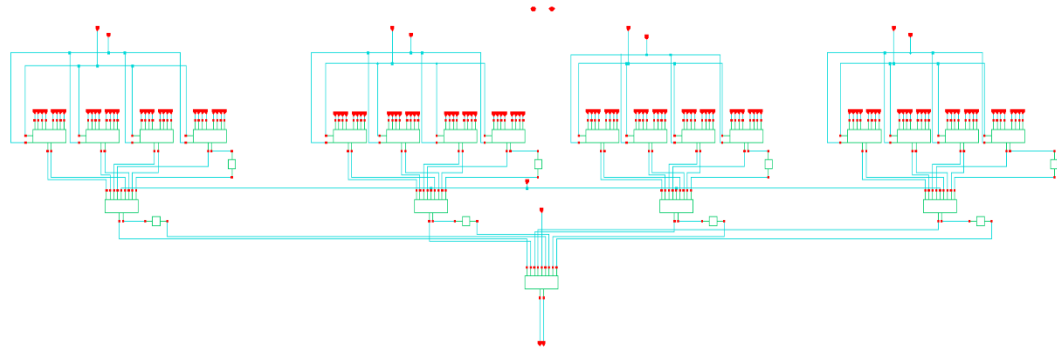
As mentioned earlier, all designs are developed in Cadence-Virtuoso schematic editor using 90nm-1.2V CMOS technology. All designs are designed using standard- V_t transistors. Figure 3.14 shows the schematic implementation diagrams of 64-bit digital static, full dynamic and partially dynamic comparators. Simulations are performed in Cadence-ADE L using Spectre simulator and simulation results for dynamic design do not include clock tree. All input signals, A[63:0] and B[63:0], and clock signals were given in simulator to test the performance of the proposed designs.



(a)



(b)



(c)

Figure 3.14 Schematic designs for 64-bit digital comparators in 90nm CMOS technology (a) Static (b) Full dynamic (c) Partially dynamic

As mentioned in Section 3.4.5, the average power consumption when measuring the worst path delay for static design are obtained by triggering input vectors seen in Figure 3.9(a). The proposed low cost static design works at a worst path delay of 374ps, which is a moderate value when compared with [18] [21] [22] [26]. Though the delay values are moderate for this proposed static design, average power consumption at worst case delay vector and worst power vector, 232 μ W and 822 μ W respectively, are significantly lower compared to the previous work. Also these proposed designs are intended for resource sharing purpose. Total number of transistors required to build the static design are 1244 (768+476) of which 768 transistors, 64 XE block transistors (64 \times 12), can be used to design other functionalities such as adder and multiplier.

The proposed full and partially dynamic design's delay and power consumption at full activity are obtained by triggering input data vectors shown in Figure 3.15. Clocks are given for full and partially dynamic designs according to the clock strategies illustrated in Figure 3.12 and Figure 3.13 respectively. The duty cycle of the clock is

given as 20% (40ps) and 25% (50ps), and then the delay and average power consumption are measured. As summarized in Table 3.3, while varying the duty cycle, the proposed dynamic design with self-pipelining works at 5GHz with worst case delays varying from 268ps to 278ps for full dynamic design, and 254ps and 256ps for partially dynamic design.

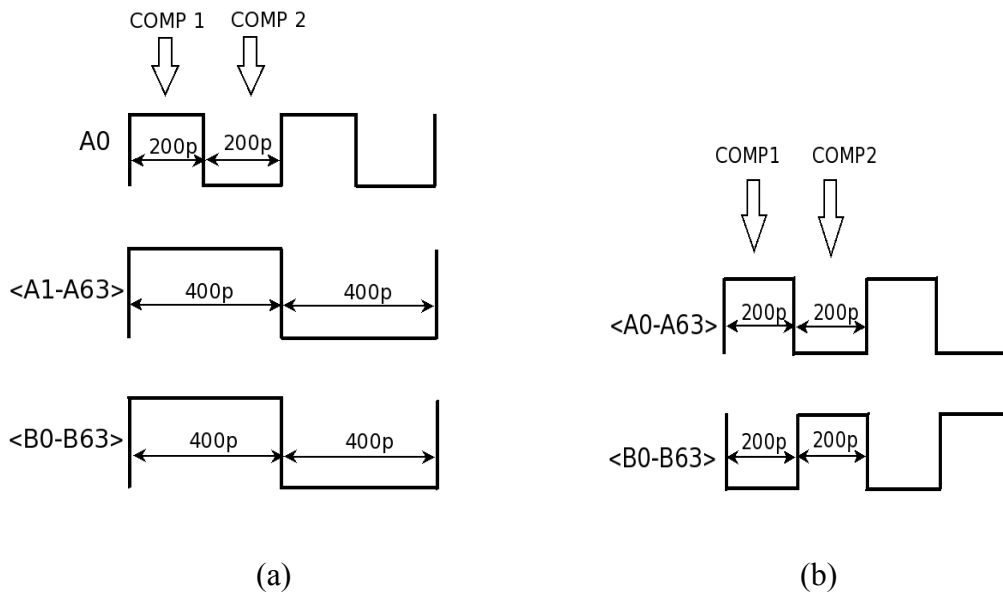


Figure 3.15 Worst case vectors (a) for delay, (b) for power at 5 GHz clock.

Figure 3.16 shows the simulation results for worst case delay measurement with 20% duty cycle clock for the full dynamic design. Delay values for partial dynamic design are nearly unchanged with variation in duty cycle. These delay values can be further lowered by designing with larger transistor widths. At 5 GHz clock (comparison rate), while triggering the worst path delay vector, the design works with very low average power consumption results, shown in Table 3.3. While varying the duty cycle of

the clock from 20% to 25%, the power consumption varies from $156\mu\text{W}$ to $164\mu\text{W}$ for full dynamic design and $639\mu\text{W}$ to $643\mu\text{W}$ for partially dynamic design. Though the power consumption for partial dynamic design is higher at worst path delay vector, worst power consumption (2.57mW to 2.78mW) is far below the worst power consumption (4.21mW to 5.00mW) by full dynamic design. Partial dynamic designs' worst power consumption varies slightly with the clock duty cycle when compared with the variation for full dynamic design.

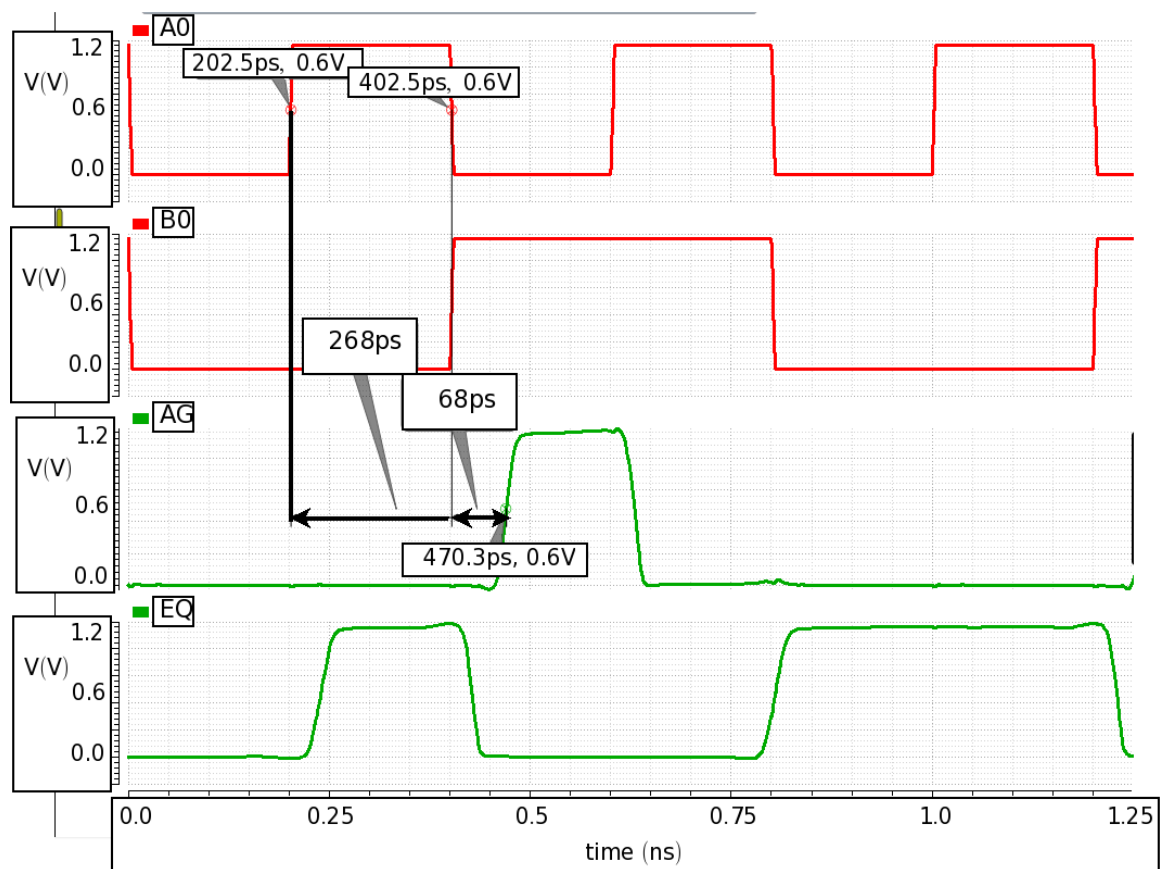


Figure 3.16 Simulation results showing worst case delay at 40ps (20%) duty cycle (D) clock for 64-bit dynamic comparator.

Table 3.3 Performance Comparisons for 64-bit Static and Dynamic Comparator

Design	Technology	Clock/input Frequency		Power (mW)	Delay (ps)	Transistor count	PDP (pJ)
C.C. Wang 1997 [10]	0.6um 5V	1GHz (Dynamic)		0.111mW/MHz [11]	6300 [11]	1890	111
C. Huang 2003 [11]	0.6um 3V	185MHz (Dynamic)		0.023mW/MHz	5400	1640	23
Boroujeni 2005 [13]	100nm 1V	1GHz (Static)	HS	0.614	302	N/A	0.185
			LP	0.150	570	N/A	0.085
J.Y. Kim 2007 [16]	0.18um 1.8V	200MHz (Static)		2.53	1120	964	2.83
Menendez 2007 [26] [20]	100nm 1.2V	N/A (Dynamic-LP)		0.025	8471	>1000	0.214
		N/A (Dynamic-HS)		1.25	462	>1000	0.577
S. Perri 2008 [18]	90nm 1V	2.7GHz (Static)		4.4	373	N/A	1.64
		4.3GHz (Dynamic)		1.0uW/MHz	230	1051	0.989
F. Frustaci 2010 [19]	90nm 1V	N/A (Dynamic)		0.77uW/MHz	258	1365 [21]	0.198pJ/GHz (EDP)
M. Larijani 2011 [20]	90nm ptm 1.2V	5.4GHz (Dynamic)		13.6 ⁽³⁾	222	724	3.0 ⁽³⁾
	UMC 90nm 1.2V	4.2GHz (Dynamic)		3.3 ⁽³⁾	310	724	1.02 ⁽³⁾
P. Chuang 2012 [21]	90nm 1V	100MHz (Dynamic)		0.207 (worst power)	240	1206	0.049
	65nm 1V	100MHz (Dynamic)		0.189 (worst power)	166	1206	0.031
S.A. Hafeez 2012 [22]	0.15um 1.5V	1GHz (Static)		7.76	860	4000	6.67
P. Chuang 2013 [23] ⁽²⁾⁽⁵⁾	65nm 1V	500MHz (Static)		1.87 ⁽⁶⁾ ($\alpha=12.5\%$)	203	N/A	0.38 (2.15) ⁽⁴⁾⁽⁶⁾
		500MHz (CD logic)		2.34 ⁽⁶⁾ ($\alpha=12.5\%$)	167	N/A	0.39 (2.2) ⁽⁴⁾⁽⁶⁾
Proposed Radix-4 Low Cost Static, Dynamic and Partially Dynamic designs' simulation results							
Static	90nm 1.2V	1 GHz		0.232(0.822) ⁽⁴⁾	374	768 ⁽¹⁾ +476 =1244 ⁽¹⁾	0.086(0.307) ⁽⁴⁾
Dynamic	90nm 1.2V	5GHz D=20%		0.156(4.21) ⁽⁴⁾	268	320 ⁽¹⁾ +449 =769 ⁽²⁾	0.042(1.13) ⁽⁴⁾
		D=25%		0.164(5.00) ⁽⁴⁾	278		0.046(1.39) ⁽⁴⁾
Partially-Dynamic	90nm 1.2V	5GHz D=20%		0.639(2.57) ⁽⁴⁾	254	768 ⁽¹⁾ +449 =1217 ⁽²⁾	0.162(0.65) ⁽⁴⁾
		D=25%		0.643(2.78) ⁽⁴⁾	256		0.164(0.71) ⁽⁴⁾

LC-Low Cost, HS- High Speed, LP-Low Power

¹ for resource sharing

² results include clock tree

³ Average power consumption measured at worst case delay vector

⁴ Numbers in brackets related to maximum power consumption (full activity)

⁵ Silicon results

⁶ Values are extrapolated in [23] to full speed by using the formula: $\frac{1}{(\text{comparator delay} \times 500\text{MHz})}$

The power-delay product (PDP) at worst case delay vector for full and partially dynamic designs varies from 0.042pJ to 0.046pJ and 0.162pJ to 0.164pJ respectively with clock duty cycle changing from 20% to 25%. Worst PDP values for full and partially dynamic designs at worst power vector varies from 1.13pJ to 1.39pJ and 0.65pJ to 0.71pJ respectively, which are far lower than the simulated/measured results obtained from the previously proposed designs. Worst PDP values are calculated by multiplying worst power and worst delay for the proposed designs. There is a comparative analysis done using results summarized in Table 3.3. The transistor count of 769 (320+449) for fully dynamic design is lower than many designs proposed previously but this full dynamic design requires large clock tree. If we consider the resource sharing, sharing of 64 dynamic XE blocks, for other functionality, the transistor count is even lower.

There is a comparative analysis done on normalized delay and PDP values obtained in this work and [23]. Results obtained in current work using 90nm-1.2V are normalized to 65nm-1V process to compare the performance of the proposed design with recently proposed design [23]. To make proper comparisons, PDP values from [23] are extrapolated from its full speed (5.98GHz frequency or 167ps delay) to the proposed designs operating speed (5GHz). Normalization formula for delay and extrapolated normalization formula for PDP (Energy) used in [23] are used in this work to make comparisons as shown below.

$$t_{d_{norm}} = t_d \times \left(\frac{65nm}{tech.}\right) \quad (3.14)$$

$$PDP_{norm} = PDP_{operating} \times \left(\frac{65nm}{tech.}\right) \times \left(\frac{1V}{V_{tech}}\right)^2 \quad (3.15)$$

$$PDP_{norm-extra} = PDP_{norm} \times \left(\frac{5GHz}{f_{operating}}\right) \quad (3.16)$$

Comparisons are depicted in Figure 3.17 and Figure 3.18 for normalized delay and PDP respectively. The normalized delay values for the proposed full and partial dynamic designs are 16% to 29% and 9.5% to 10.7% higher than the delay value of 167ps measured in [23], normalized-extrapolated energy values illustrated in Figure 3.18 are far lower than the normalized and extrapolated (to 5GHz) values in [23]. Partially dynamic design shows a superior power improvement than full dynamic design with the transistor count trade-off. Delays can be further decreased by designing a high speed design using larger width transistors with a slight power trade-off. Proposed dynamic designs works with a superior improvement in worst case energy values with 69% and 62% improvement for full dynamic design and 82% and 80% improvement for partial dynamic design when compare with the normalized-extrapolated silicon results measured in [23], which includes clock tree.

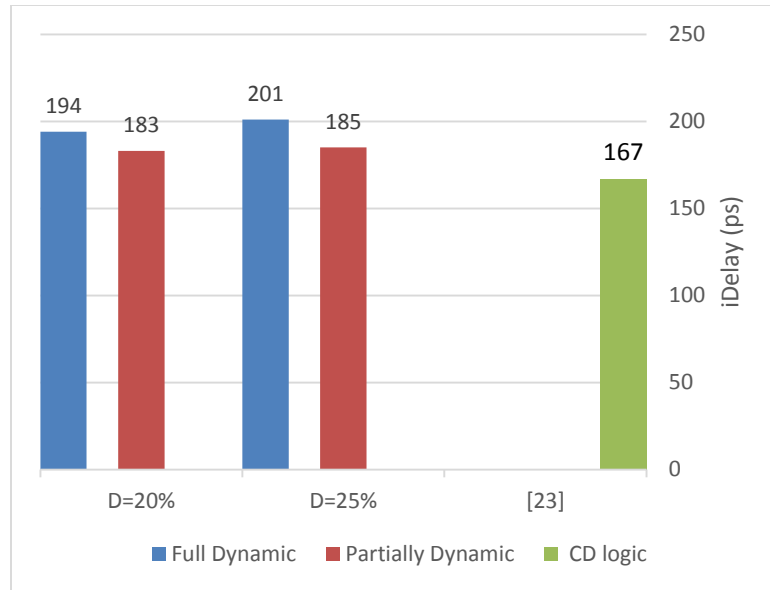


Figure 3.17 Comparison of normalized (to 65nm-1V) delay results for proposed 64-bit low cost dynamic design at 5GHz clock with various D and silicon results measured in [23].

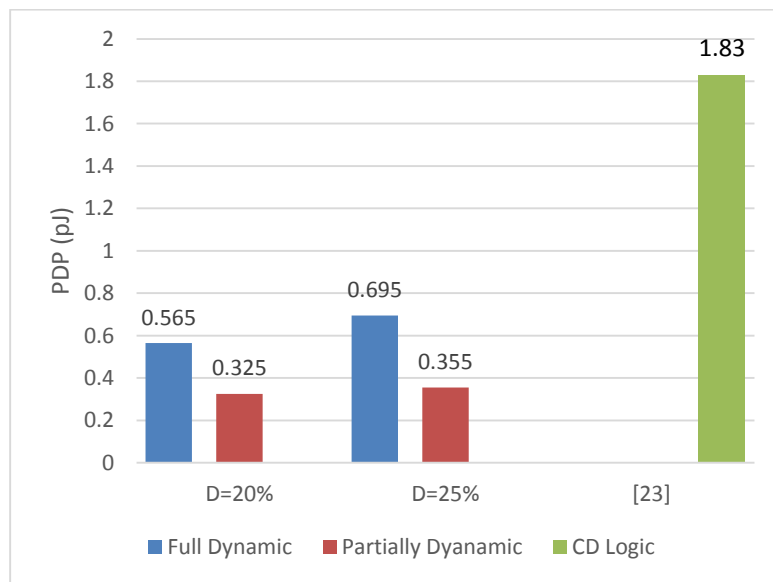


Figure 3.18 Comparison of normalized-extrapolated (to 65nm-1V, 5GHz) PDP results for proposed 64-bit low cost dynamic design at 5GHz clock with various D and silicon results measured in [23].

4 Conclusion & Future Work

4.1 Conclusion

- The proposed technique for the 4-bit CPL adder block, sub-block of proposed 16-bit carry select adder, results in significant reduction in power, area, and delay for the square root and linear carry select adder designs when compare with the regular designs.
- One set of the proposed design replaces the requirement of two sets of RCA or RCA and BEC to save the area and power.
- New static and dynamic 64-bit comparator designs are presented in this work.
- The proposed design approach using a reduced duty cycle clock for pre-charge and pre-discharge along with a strategy of width adjustment in comparator sub-blocks increase the self-pipelining of the design to make the design work at 5GHz clock speed.
- Proposed designs (static, full dynamic, and partially dynamic) maintain competitively lower transistor count and worst power consumptions.
- Full dynamic design is suitable for low transistor count applications while partially dynamic design is apt for low power consuming applications.

- The proposed reduced duty cycle clock usage in dynamic designs with reducing power consumption and sizing strategy for radix-4 tree structure, is applicable for many tree based structures to improve the self-pipelining.
- In addition to low power and high speed applications, the proposed designs in static and dynamic modes can share some portion of resources for other important functionalities, such as addition, multiplication, etc.

4.2 Future Work

Future work include:

- Developing new techniques to design a 64 bit and a 128 bit fast, low-power consuming and area efficient adders.
- Improving the power efficiency and speed (>5GHz to 10GHz) of the operation by implementing with different low power and high speed XOR-XNOR designs. This design can reach even more speeds by making the high speed sub blocks and adding registers for high speed pipelining.
- Silicon implementation and exhaustive functional testing.
- Testing the robustness of the designs by doing PVT variations and Monte Carlo analysis.
- Testing signal integrity and power integrity while packaging and placing on printed circuit board (PCB).

5 References

- [1] V. Oklobdzija, *High-Performance System Design: Circuits and Logic*, Wiley-IEEE Press, 1999.
- [2] D. M. Morris and N. H. W. , *CMOS VLSI Design - A Circuits and System Perspective*, 4th, Ed., Addison Wesley, 2010.
- [3] M. M. M, "Computer System Architecture", 3rd ed., Upper Saddle River, NJ: Prentice-Hall, 1993.
- [4] O. J. Bedrij, "Carry-Select Adder," Vols. EC-11, no. 3.
- [5] B. R. a. H. Kittur, "Low-Power and Area-Efficient Carry Select Adder," *IEEE Trans. on VLSI Systems*, vol. 20, pp. 371-375, February 2012.
- [6] C. T. Y and H. M. J, "Carry-Select adder using single ripple-carry adder," *Electronics Letters*, vol. 34, pp. 2101-2103, October 1998.
- [7] L. Shanigarapu and B. P. S. , "Low-power and high speed carry select adder," *Internations Journal of Scientific and Research Publications*, vol. 3, no. 8, August 2013.
- [8] R. Z. and W. F. , "Low-power logic styles: CMOS versus pass transistor logic," *IEEE Journal of Solid-State circuits*, vol. 32, pp. 1079-1090, July 1997.
- [9] M. E. Elrabaa, I. S. Abu-Khater and M. I. Elmasry, *Advanced Low-Power Digital Circuit Techniques*, Springer, 1997.
- [10] C. C. Wang, C. F. Wu and K. C. Tsai, "1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking," in *IEEE Proc. on Computers and Digital Techniques*, 1998.
- [11] C. H. Huang and J. S. Wang, "High-performance and power-efficient CMOS comparators," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 254-262, February 2003.
- [12] C. Wang, P. Lee, C. Wu and H. Wu, "High Fan-In Dynamic CMOS Comparators with Low Transistor Count," *IEEE Transactions on circuits and systems I: fundamental theory and applications*, vol. 50, September 2003.

- [13] B. Kheradmand-Boroujeni, K. Shojaee and A. Afzali-Kusha, "Design and Simulated Annealing Optimization of Static Comparator for Low-Power High-Speed CMOS VLSI," in *International Conference on Microelectronics*, 2005.
- [14] H. M. Lam and C. Y. Tsui, "High-performance single clock cycle CMOS comparator," in *IEEE Electronics Letters*, 2006.
- [15] H. M. Lam and C. Y. Tsui, "A MUX-based high-performance single-cycle CMOS comparator," *IEEE Transactions on Circuits and Systems II, Exp. Briefs*, vol. 54, pp. 591-595, July 2007.
- [16] J. -Y. Kim and H.-J. Yoo, "Bitwise competition logic for compact digital comparator," in *in proc. of IEEE Asian Solid-State Circuits Conference*, 2007.
- [17] S.-W. Cheng, "A High-Speed Magnitude Comparator with Small Transistor Count," in *IEEE Proc. of International Conference on Electronics, Circuits and Systems*, 2003.
- [18] S. Perri and P. Corsonello, "Fast Low-Cost Implementation of Single Clock-Cycle Binary Comparator," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 55, no. 12, pp. 1239-1243, December 2008.
- [19] F. Frustaci, S. Perri, M. Lanuzza and P. Corsonello, "A new low-power high-speed single-clock-cycle binary comparator," in *in Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010.
- [20] M. Larijani and M. B. Ghaznavi-Ghoushi, "A 4.2GHz, 3.3mW, 1.1pJ, 90nm 64-bit new digital comparator with 1.2 clock pipeline delay," in *Iranian Conference on Electrical Engineering (ICEE)*, 2011.
- [21] P. I. -J. Chuang, D. Li and M. Sachdev, "A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Comparator," *IEEE Trans. on Circuits and Systems II*, vol. 59, no. 2, pp. 108-112, December 2012.
- [22] S. A. Hafeez, A. Gordon-Ross and B. Parhami, "Scalable Digital CMOS Comparator Using a Parallel Prefix Tree," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 21, no. 11, pp. 1989-1998, December 2012.
- [23] P. I. -J. Chuang, M. Sachdev and V. C. Gaudet, "A 167-ps 2.34mW Single-Cycle 64-Bit Binary Tree Comparator with Constant-Delay Logic in 65-nm CMOS," *IEEE Trans. on Circuits and Systems*, vol. 61, no. 1, pp. 160-171, December 2013.
- [24] K. H. Cheng and C. S. Huang, "The novel efficient design of XOR/XNOR function for adder applications," in *Proc. of the 6th IEEE International Conference on Electronics, Circuits and*

Systems (ICECS), 1999.

- [25] R. Kumar and V. K. Pandey, "A new 5-transistor XOR-XNOR circuit based on the pass transistor logic," in *World Congress on Information and Communication Technologies (WICT), 2011.*
- [26] E. M. e. al., "CMOS Comparators for High-Speed and Low-Power Applications," in *International Conference on Computer Design, 2007.*

Copyright By

N.V. Vijaya Krishna. Boppana

2014