

A High-Speed 64-Bit Binary Comparator

Anjuli, Satyajit Anand

E&CE Department, FET-MITS, Lakshmangarh, Sikar, Rajasthan(India)

Abstract : A high-speed 64-bit binary comparator is proposed in this brief. Comparison is most basic arithmetic operation that determines if one number is greater than, equal to, or less than the other number. Comparator is most fundamental component that performs comparison operation. This brief presents comparison of modified and existing 64-bit binary comparator designs concentrating on delay. Means some modifications are done in existing 64-bit binary comparator design to improve the speed of the circuit. Comparison between modified and existing 64-bit binary comparator designs is calculated by simulation that is performed at 90nm technology in Tanner EDA Tool.

Keywords - Binary comparator, digital arithmetic, high-speed.

I. INTRODUCTION

In digital system, comparison of two numbers is an arithmetic operation that determines if one number is greater than, equal to, or less than the other number [1]. So comparator is used for this purpose. Magnitude comparator is a combinational circuit that compares two numbers, A and B, and determines their relative magnitudes (Fig.1). The outcome of comparison is specified by three binary variables that indicate whether $A > B$, $A = B$, or $A < B$.

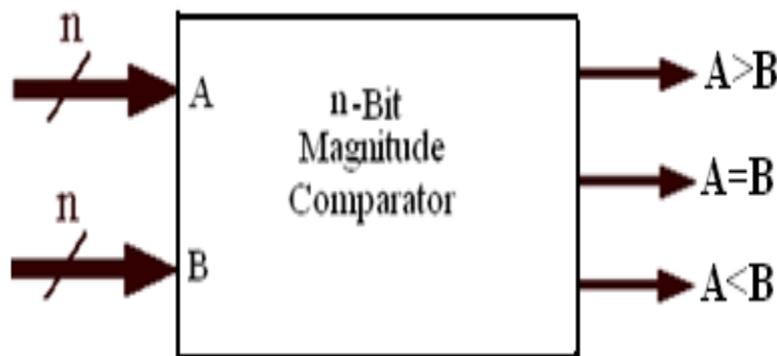


Figure 1. Block Diagram of n-Bit Magnitude Comparator

The circuit, for comparing two n-bit numbers, has 2n inputs & 2^{2n} entries in the truth table. For 2-bit numbers, 4-inputs & 16-rows in the truth table, similarly, for 3-bit numbers 6-inputs & 64-rows in the truth table [1].

In recent year, high speed & low power device designs have emerged as principal theme in electronic industry due to increasing demand of portable devices. This tremendous demand is due to popularity of battery operated portable equipments such as personal computing devices, wireless communication, medical applications etc. Demand & popularity of portable electronic devices are driving the designers to strive for higher speed, smaller power consumption and smaller area.

The logic style used in logic gates basically influences the speed, size, power dissipation, and the wiring complexity of a circuit [2]. Circuit size depends on the number of transistors and their sizes and on the wiring complexity [3]. The wiring complexity is determined by the number of connections and their lengths. All these characteristics may vary considerably from one logic style to another and thus proper choice of logic style is very important for circuit performance [4].

In order to differentiate both the designs existing and modified, simulations are carried out for delay and power consumption with 1 volt input voltage (and supply voltage), 30°C temperature and 50MHz frequency at 90nm technology in Tanner EDA Tool.

II. 64-BIT BINARY COMPARATOR

64-bit binary comparator compares two numbers each having 64 bits (A_{63} to A_0 & B_{63} to B_0). For this arrangement truth table has 128 inputs & 2^{128} entries. By using comparator of minimum number of bits, a comparator of maximum number of bits can be design [5], [6], [7] with the help of tree-based structure logic [8] and also with other useful logic styles.

III. EXISTING 64-BIT BINARY COMPARATOR DESIGN

64-bit comparator in reference [8], [9], [10] represents tree-based structure which is inspired by fact that G (generate) and P (propagate) signal can be defined for binary comparisons, similar to G (generate) and P (propagate) signals for binary additions.

Two number (each having 2-bits: A_1, A_0 & B_1, B_0) comparison can be realized by:

$$B_{Big} = \overline{A_1}B_1 + (\overline{A_1} \oplus \overline{B_1}) \cdot (\overline{A_0}B_0) \quad (1)$$

$$EQ = (\overline{A_1} \oplus \overline{B_1}) \cdot (\overline{A_0} \oplus \overline{B_0}) \quad (2)$$

For $A < B$, " B_{Big}, EQ " is "1,0". For $A = B$, " B_{Big}, EQ " is "0,1". Hence, for $A > B$, " B_{Big}, EQ " is "0,0". Where B_{Big} is defined as output A less than B (A_{LT_B}). A closer look at equation (1) reveals that it is analogous to the carry signal generated in binary additions. Consider the following carry generation:

$$\begin{aligned} C_{out} &= AB + (A \oplus B) \cdot C_{in} \\ &= G + P \cdot C_{in} \end{aligned} \quad (3)$$

Where A & B are binary inputs C_{in} is carry input, C_{out} is carry output, and G & P are generate & propagate signals, respectively.

After comparing equations (1) & (3):

$$G_1 = \overline{A_1}B_1 \quad (4)$$

$$EQ_1 = (\overline{A_1} \oplus \overline{B_1}) \quad (5)$$

$$C_{in} = \overline{A_0}B_0 \quad (6)$$

C_{in} can be considered as G_0 . Since for static logic, equation (1) requires tall transistor stack height, hence, an encoding scheme is employed to solve this problem. For this, encoding equation is given as:

$$G_{[i]} = \overline{A_{[i]}} B_{[i]} \quad (7)$$

$$EQ_{[i]} = \overline{A_{[i]} \oplus B_{[i]}} \quad (8)$$

Where $i = 0, \dots, 63$.

Put these two values from equations (7) & (8) in equations (1) & (2).

$$B_{Big[2j+1:2j]} = G_{[2j+1]} + EQ_{[2j+1]} \cdot G_{[2j]} \quad (9)$$

$$EQ_{[2j+1:2j]} = EQ_{[2j+1]} \cdot EQ_{[2j]} \quad (10)$$

Where $j = 0, \dots, 31$.

G & P signals can be further combined to form group G & P signals.

$$\begin{aligned} B_{Big[3:0]} &= \overline{A_3}B_3 + (\overline{A_3} \oplus \overline{B_3}) \cdot \overline{A_2}B_2 + (\overline{A_3} \oplus \overline{B_3}) \cdot (\overline{A_2} \oplus \overline{B_2}) \cdot \overline{A_1}B_1 + (\overline{A_3} \oplus \overline{B_3}) \cdot (\overline{A_2} \oplus \overline{B_2}) \cdot (\overline{A_1} \oplus \overline{B_1}) \cdot \overline{A_0}B_0 \\ &= \overline{A_3}B_3 + (\overline{A_3} \oplus \overline{B_3}) \cdot [\overline{A_2}B_2 + (\overline{A_2} \oplus \overline{B_2}) \cdot \{\overline{A_1}B_1 + (\overline{A_1} \oplus \overline{B_1}) \cdot \overline{A_0}B_0\}] \end{aligned}$$

$$\begin{aligned} &= G_3 + EQ_3 \cdot \{G_2 + EQ_2 \cdot (G_1 + EQ_1 \cdot G_0)\} \\ B_{Big[3:0]} &= B_{Big[3:2]} + EQ_{[3:2]} \cdot B_{Big[1:0]} \end{aligned} \quad (11)$$

$$EQ_{[3:0]} = EQ_{[3:2]} \cdot EQ_{[1:0]} \quad (12)$$

Similarly, for 64-bit comparator, B_{Big} & EQ can be computed as:

$$B_{Big[63:0]} = G_{63} + \sum_{k=0}^{62} (G_k \cdot \prod_{m=k+1}^{63} EQ_m) \quad (13)$$

$$EQ_{[63:0]} = \prod_{m=0}^{63} EQ_m \quad (14)$$

Fig. 2 shows 8-bit version of existing tree-based comparator structure and Fig. 3 -Fig. 5 shows corresponding circuit schematics for each logic block of each stage. Pre-encoding circuitry is aimed to minimize the number of transistors. Hence, modified pass transistor logic style is employed to reduce the number of transistors up to 9. In above 8-bit example circuitry, the first stage comparison circuit implements equations (9) & (10) for $j = 0, \dots, 3$, whereas the second stage generates $B_{Big[3:0]}$, $B_{Big[7:4]}$ and $EQ_{[3:0]}$, $EQ_{[7:4]}$ according to equations (11 & 12). Finally, $B_{Big[7:0]}$ and $EQ_{[7:0]}$ are computed in third stage according to equations (13 & 14).

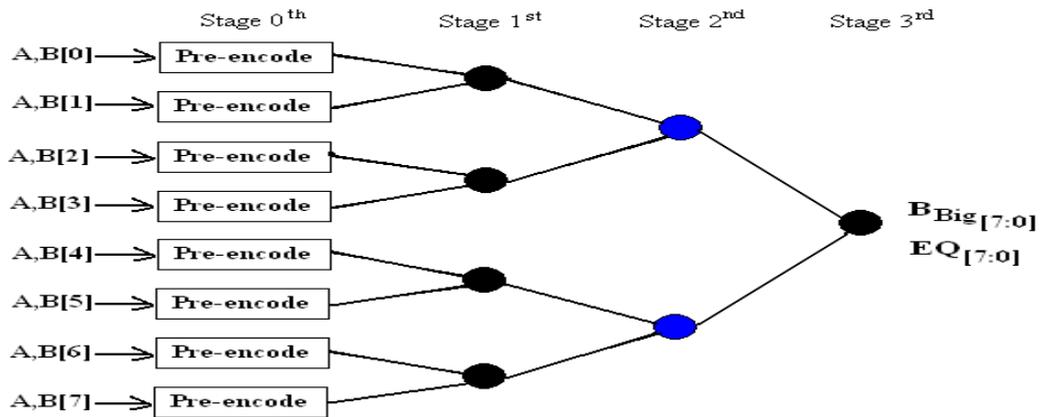


Figure 2. Tree-Diagram of 8-Bit Binary Comparator

Stage 0th is implemented using modified pass transistor logic style giving output in actual form, Stage 1st is implemented using CMOS logic style giving output in inverse form, Stage 2nd is also implemented using CMOS logic style but giving output in actual form.

64-bit comparator is here designed by using 7 stages (from 0th to 6th). In stage 0th, modified pass transistor logic style circuitry (as in Fig. 3) is employed to produce “less than” & “equal to” outputs. Outputs of stage 0th act as inputs of stage 1st. In stage 1st, CMOS circuitry (as in Fig. 4) is employed to produce inverse inputs for stage 2nd. In stage 2nd, again CMOS circuitry (as in Fig. 5) is employed to produce actual inputs for stage 3rd. Now, according to tree structure given in Fig. 2, again circuitry of stage 1st is used for stage 3rd. Similarly, for stage 4th, circuitry of stage 2nd is employed. For stage 5th circuitry of stage 1st is employed. For stage 6th circuitry of stage 2nd is employed. Accordingly schematic of Existing 64-bit binary comparator is drawn and shown in Fig. 6. Existing design requires 1206 transistor count for 64-bit binary comparator.

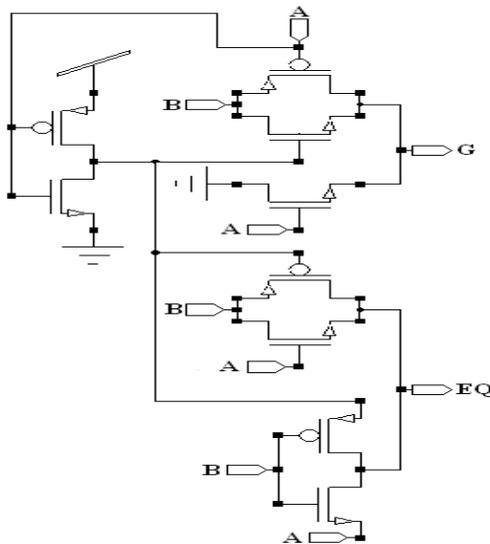


Figure 3. Schematic of Stage 0th of Existing 64-Bit Binary Comparator

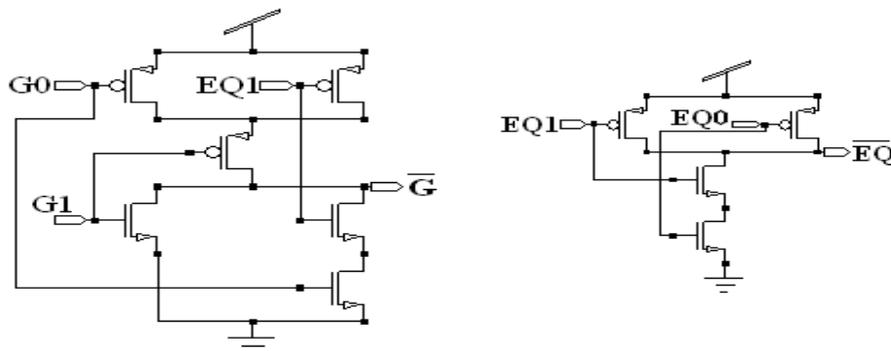


Figure 4. Schematic of Stage 1st of Existing 64-Bit Binary Comparator

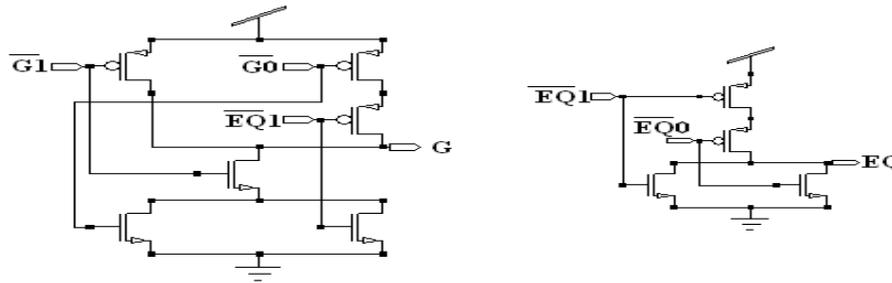


Figure 5. Schematic of Stage 2nd of Existing 64-Bit Binary Comparator

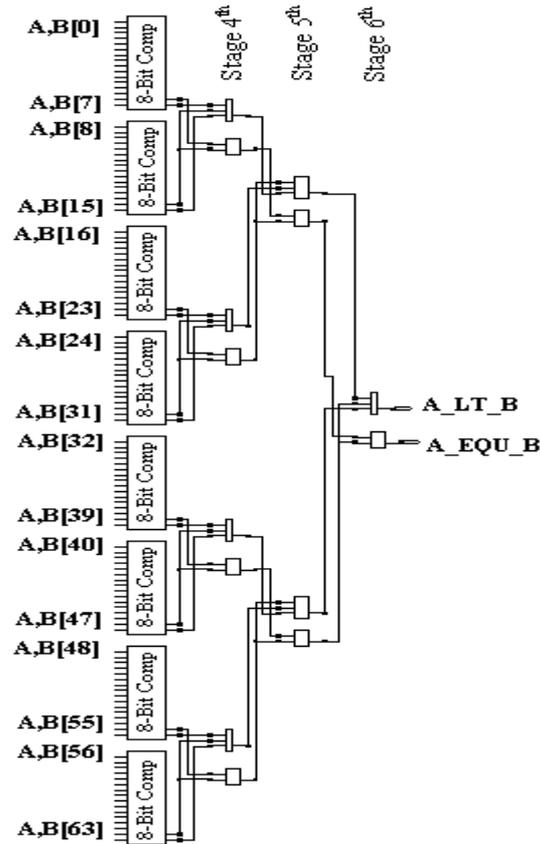


Figure 6. Schematic of Existing 64-Bit Binary Comparator

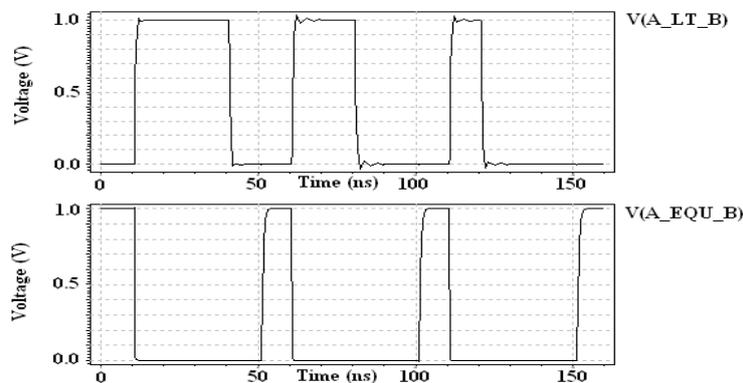


Figure 7. Waveforms of Existing 64-Bit Binary Comparator

According to input bit stream, waveforms of existing 64-bit binary comparator are obtained and shown in Fig. 7. Waveforms show that only one output is high (“1”) at a time. When both the outputs “less than” & “equal to” (A_LT_B & A_EQU_B) are low (“0”), then waveforms represent that “greater than” output is high (A_GT_B is “1”). Simulation results for this design are given in Table I - III for conclusion

IV. MODIFIED 64-BIT BINARY COMPARATOR DESIGN

Some modifications are done in existing 64-bit binary comparator design [8] to improve the speed of the circuit. Existing 64-bit binary comparator design [8] follows tree-based structure from 2-bit to 64-bit circuitry. But modified design follows tree-based structure from 2-bit to 8-bit circuitry only. After 8-bit to 64-bit circuitry, modified design follow simple logic structure in place of tree-based structure. Fig. 8 shows logic diagram of modified 64-bit binary comparator.

4.1 Logic Diagram

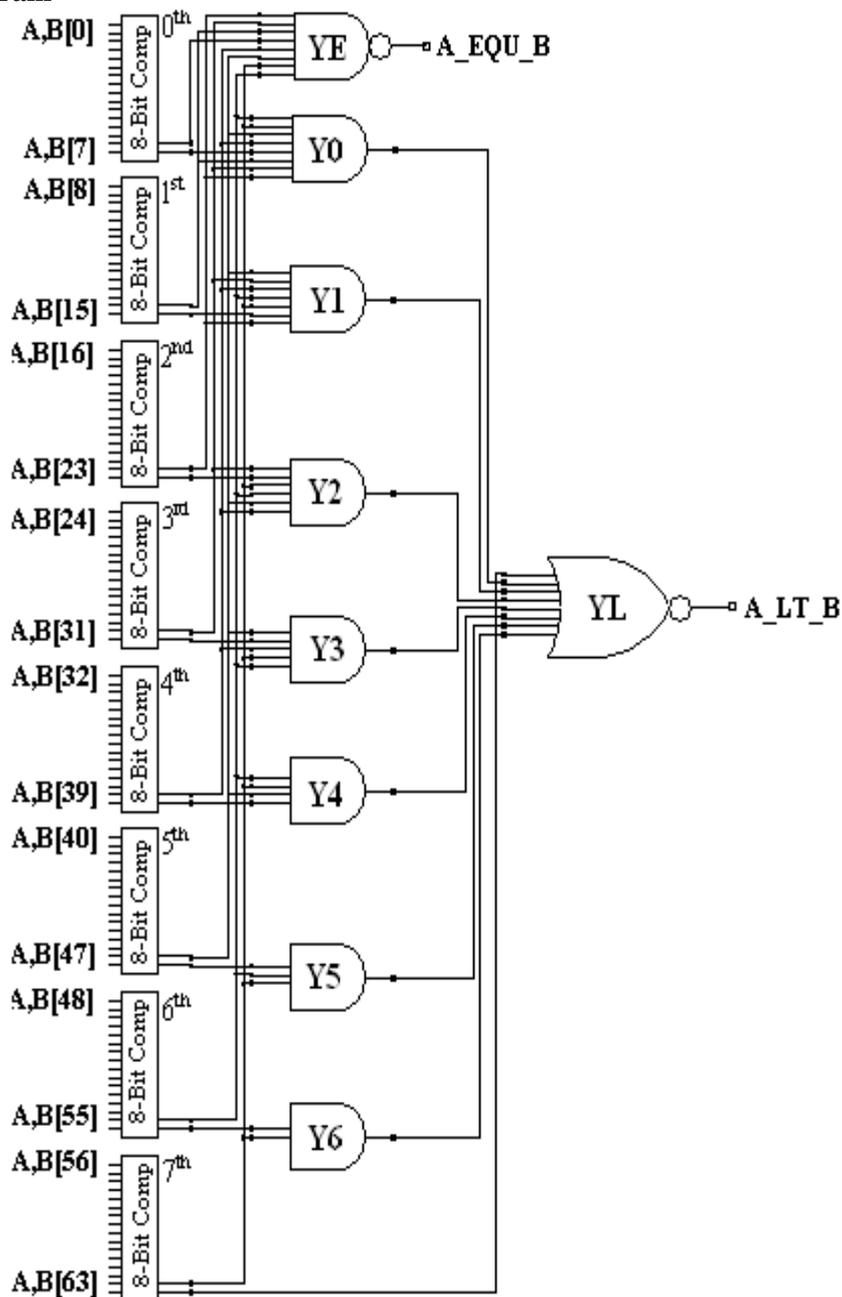


Figure 8. Logic Diagram of Modified 64-Bit Binary Comparator

4.2 Schematic of Different Logic Gates

The “A less than B” output (A_LT_B) of 0th 8-bit comparator and “A equal to B” outputs (A_EQU_B) of seven (from 1st to 7th) 8-bit comparators are given to AND gate Y0 that produces input for NOR gate $\bar{Y}L$. This 8-input AND gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 8-inputs are NANDed through four 2-input NAND gates and then NORed through two 2-input NOR gates then finally ANDed through one 2-input AND gate. Hence, one 8-input AND gate has been implemented using 30 transistors count. Schematic of AND Gate Y0 of modified 64-bit binary comparator is shown in Fig. 9.

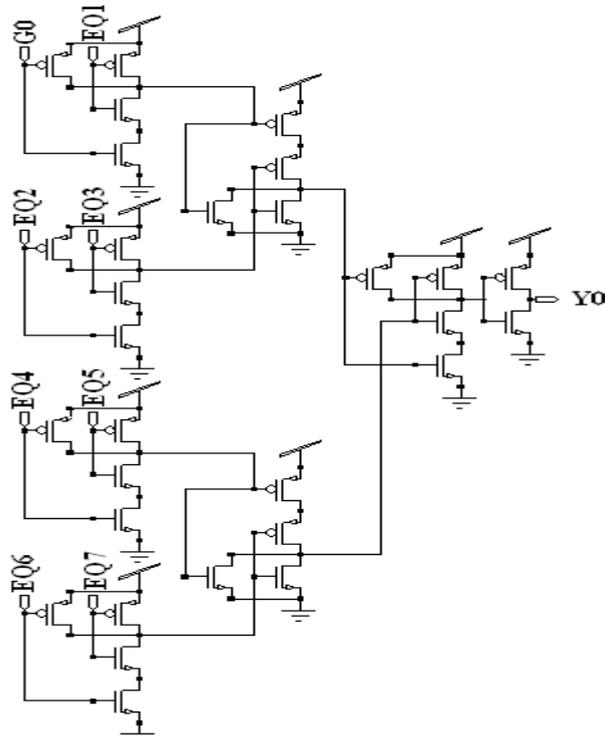


Figure 9. Schematic of AND Gate Y0 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 1st 8-bit comparator and “A equal to B” outputs (A_EQU_B) of six (from 2nd to 7th) 8-bit comparators are given to AND gate Y1 that produces input for NOR gate YL. This 7-input AND gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 7-inputs are NANDed through two 2-input NAND gates and one 3-input NAND gate and then finally NORed through one 3-input NOR gate. Hence, one 7-input AND gate has been implemented using 20 transistors count. Schematic of AND Gate Y1 of modified 64-bit binary comparator is shown in Fig.10.

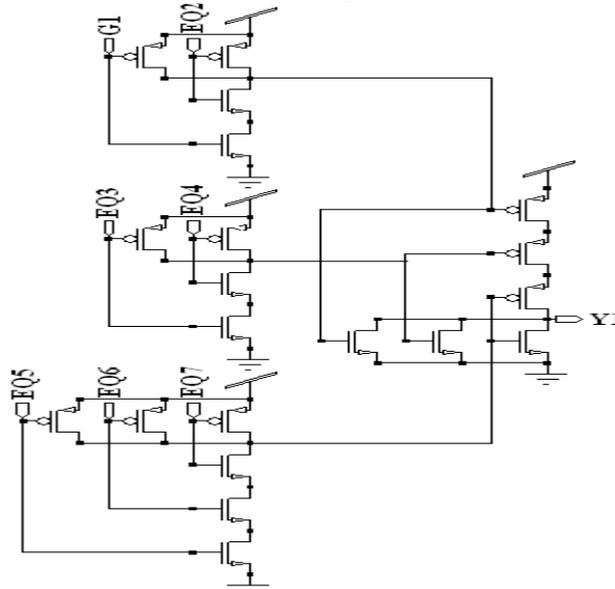


Figure 10. Schematic of AND Gate Y1 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 2nd 8-bit comparator and “A equal to B” outputs (A_EQU_B) of five (from 3rd to 7th) 8-bit comparators are given to AND gate Y2 that produces input for NOR gate YL. This 6-input AND gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 6-inputs are NANDed through three 2-input NAND gates and then finally NORed through one 3-input NOR gate. Hence, one 6-input AND gate has been implemented using 18 transistors count. Schematic of AND Gate Y2 of modified 64-bit binary comparator is shown in Fig.11.

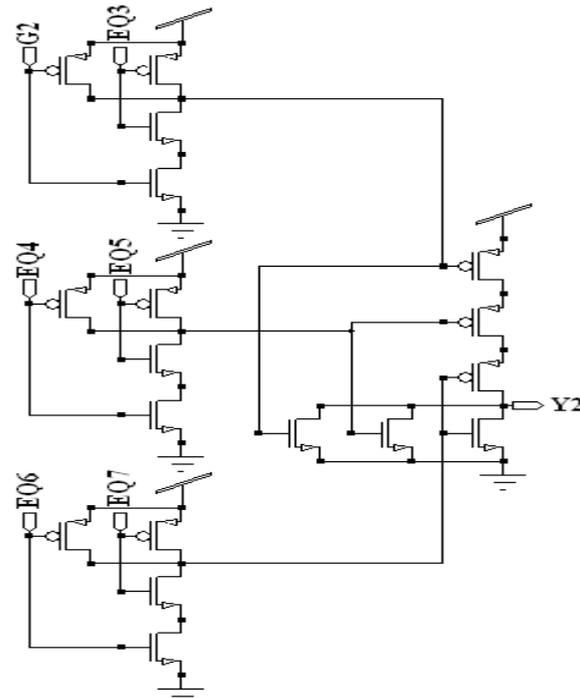


Figure 11. Schematic of AND Gate Y2 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 3rd 8-bit comparator and “A equal to B” outputs (A_EQU_B) of four (from 4th to 7th) 8-bit comparators are given to AND gate Y3 that produces input for NOR gate YL. This 5-input AND gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 5-inputs are NANDed through one 2-input NAND gate and one 3-input NAND gate then finally NORed through one 2-input NOR gate. Hence, one 5-input AND gate has been implemented using 14 transistor count. Schematic of AND Gate Y3 of modified 64-bit binary comparator is shown in Fig.12.

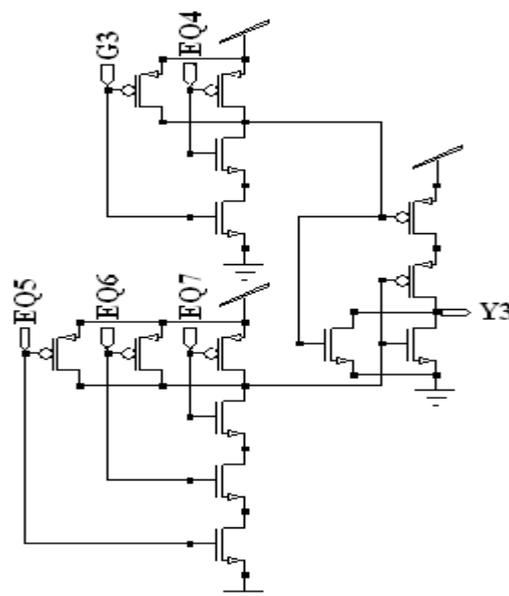


Figure 12. Schematic of AND Gate Y3 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 4th 8-bit comparator and “A equal to B” outputs (A_EQU_B) of three (from 5th to 7th) 8-bit comparators are given to AND gate Y4 that produces input for NOR gate YL. This 4-input AND gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 4-inputs are NANDed through two 2-input NAND gate and then finally NORed through one 2-input NOR gate. Hence, one 4-input AND gate has been implemented using 12 transistor count. Schematic of AND Gate Y4 of modified 64-bit binary comparator is shown in Fig.13.

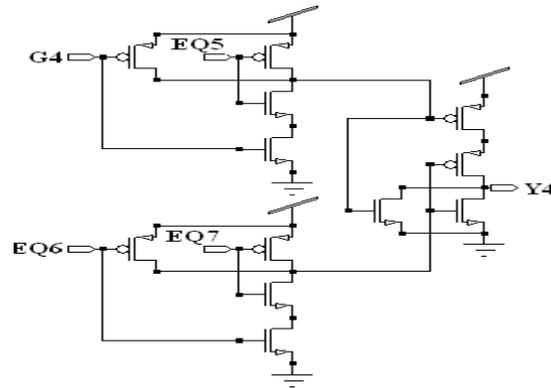


Figure 13. Schematic of AND Gate Y4 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 5th 8-bit comparator and “A equal to B” outputs (A_EQU_B) of two (6th & 7th) 8-bit comparators are given to AND gate Y5 that produces input for NOR gate YL. This 3-input AND gate is implemented using CMOS logic style. Hence, one 3-input AND gate has been implemented using 8 transistor count. Schematic of AND Gate Y5 of modified 64-bit binary comparator is shown in Fig.14.

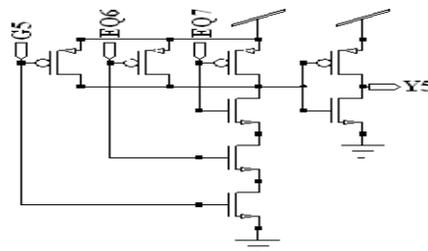


Figure 14. Schematic of AND Gate Y5 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 6th 8-bit comparator and “A equal to B” output (A_EQU_B) of 7th 8-bit comparator are given to AND gate Y6 that produces input for NOR gate YL. This 2-input AND gate is implemented using CMOS logic style. Hence, one 2-input AND gate has been implemented using 6 transistor count. Schematic of AND gate Y6 of modified 64-bit binary comparator is shown in Fig.15.

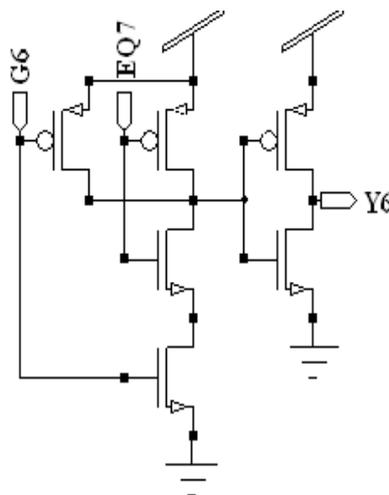


Figure 15. Schematic of AND Gate Y6 of Modified 64-Bit Binary Comparator

The “A less than B” output (A_LT_B) of 7th 8-bit comparator and outputs of seven AND gates (from Y0 to Y6) are given to NOR gate YL that produces final “A less than B” output (A_LT_B) of modified 64-bit binary comparator. This 8-input NOR gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 8-inputs are NORed through four 2-input NOR gates and then NANDed through two 2-input NAND gates then finally NORed through one 2-input NOR gate. Hence, one 8-input NOR gate has been implemented using 28 transistors count. Schematic of NOR Gate YL of modified 64-bit binary comparator is shown in Fig.16.

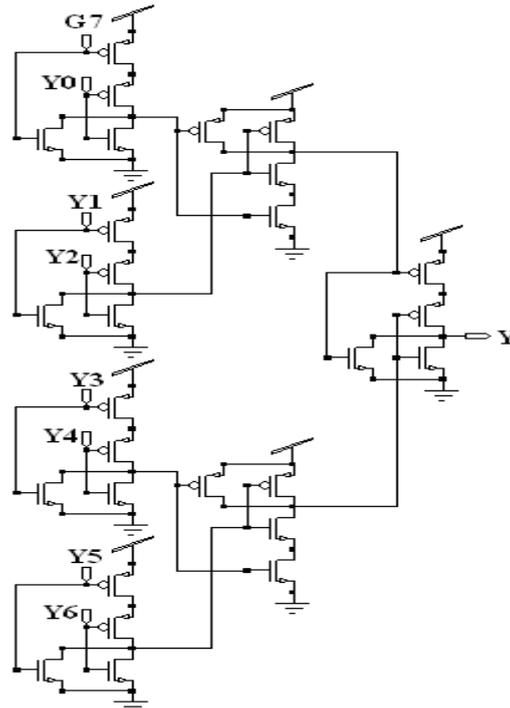


Figure 16 Schematic of NOR Gate YL of Modified 64-Bit Binary Comparator

The “A equal to B” outputs (A_EQUAL_B) of eight (from 0th to 7th) 8-bit comparators are given to NAND gate YE that produces final “A equal to B” output (A_EQUAL_B) of modified 64-bit binary comparator. This 8-input NAND gate is implemented using CMOS logic style. In order to avoid large transistor stack height, 8-inputs are NANDed through four 2-input NAND gates and then NORed through two 2-input NOR gates then finally NANDed through one 2-input NAND gate. Hence, one 8-input NAND gate has been implemented using 28 transistors count. Schematic of NAND gate YE of modified 64-bit binary comparator is shown in Fig.17.

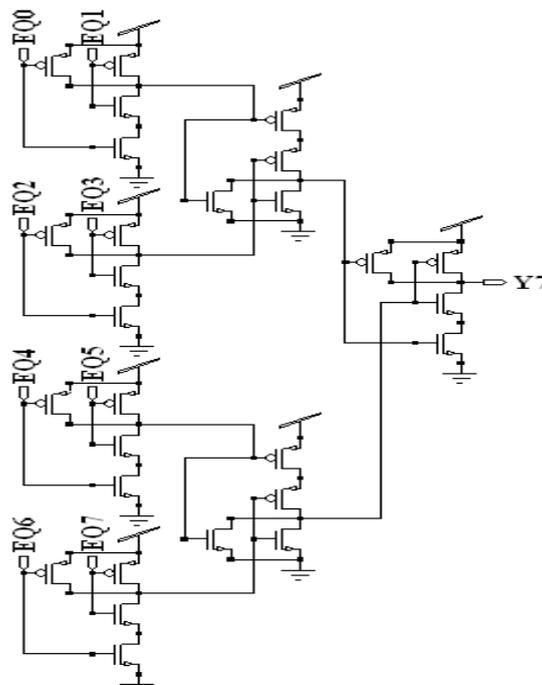


Figure 17. Schematic of NAND Gate YE of Modified 64-Bit Binary Comparator

Since output of 8-bit comparators are obtained in inverse form so NOR and NAND gates are used in place of OR and AND gates to produce final output of “A equal to B” and “A less than B” in actual form. This design requires 1300 transistor count for 64-bit comparator. Schematic (using instances of each section) of modified 64-bit binary comparator design is drawn and shown in Fig.18.

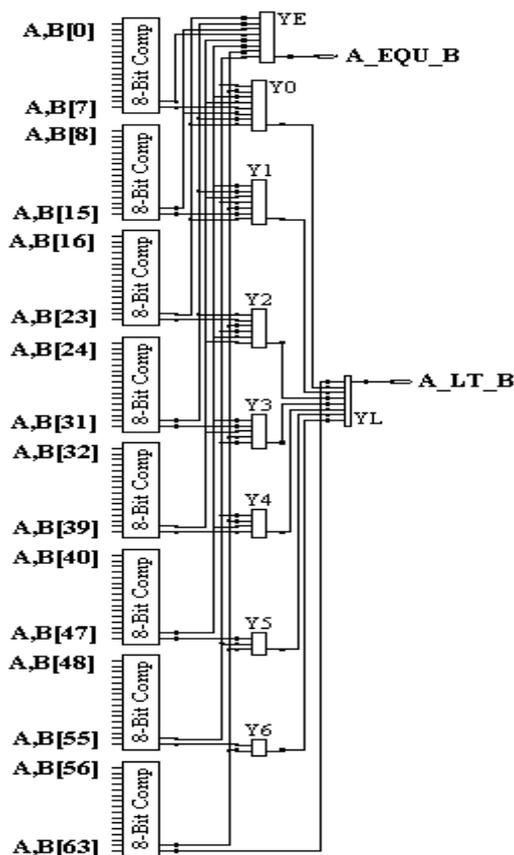


Figure 18. Schematic of Modified 64-Bit Binary Comparator

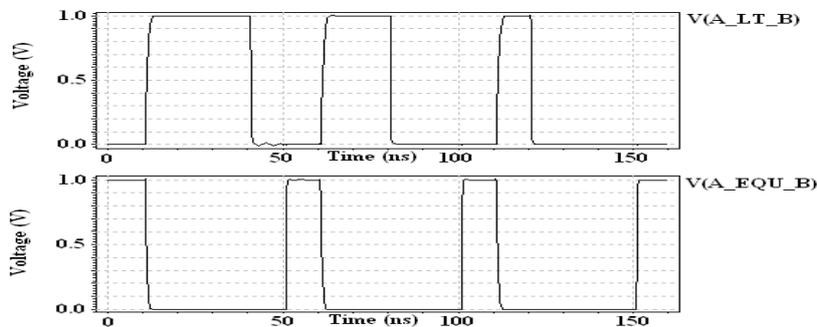


Figure 19. Waveforms of Modified 64-Bit Binary Comparator

According to input bit stream, waveforms of modified 64-bit binary comparator are obtained and shown in Fig.19. Input bit stream for modified design is same as in existing design of 64-bit comparator. Output waveforms of modified design produce same position of 1,s and 0,s as in waveforms of existing design for each input bits. Waveforms show that only one output is high (“1”) at a time. When both the outputs “less than” & “equal to” (A_LT_B & A_EQU_B) are low (“0”), then waveforms represent that “greater than” output is high (A_GT_B is “1”). Simulation results for modified 64-bit binary comparator design are given in tabular form in Table I - III.

V. SIMULATION AND COMPARISON

After simulation of both the designs final results are obtained for delay and power consumption and are shown in Table I-III. Simulations have been carried out at 90nm technology in Tanner EDA Tool..

Table I. Simulation Data with 1volt Input Voltage

Design	Power Consumption (watt)	Delay Time (second)	
		$t_{A_LT_B}$	$t_{A_EQU_B}$
EXISTING	8.9563e-006	4.4290e-009	6.7628e-010
MODIFIED	1.0024e-005	4.3682e-009	6.2350e-010

Table II. Simulation Data with 30°C Temperature

Design	Power Consumption (watt)	Delay Time (second)	
		$t_{A_LT_B}$	$t_{A_EQU_B}$
EXISTING	9.1340e-006	4.4250e-009	6.7904e-010
MODIFIED	1.0186e-005	4.3649e-009	6.2793e-010

Table III. Simulation Data with 50MHz Frequency

Design	Power Consumption (watt)	Delay Time (second)	
		$t_{A_LT_B}$	$t_{A_EQU_B}$
EXISTING	8.7920e-006	4.4277e-009	6.6736e-010
MODIFIED	9.8400e-006	4.3679e-009	6.2166e-010

After simulation of both the designs final results are obtained for delay and power consumption with 1 volt input voltage. Delay comparison of modified and existing 64-bit comparator designs is shown in Fig. 20 & Fig. 21. Simulated data for these graphs is given in Table I.

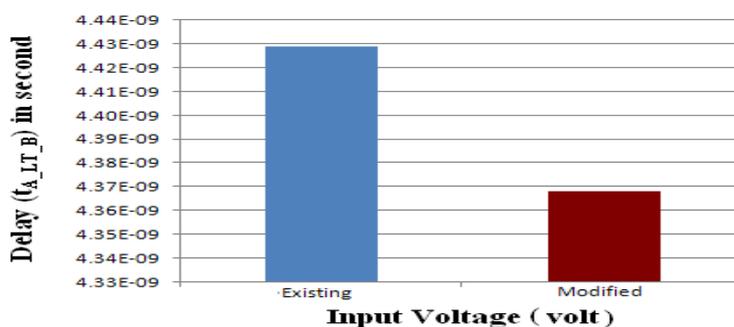


Figure 20. Delay ($t_{A_LT_B}$) with Input Voltage

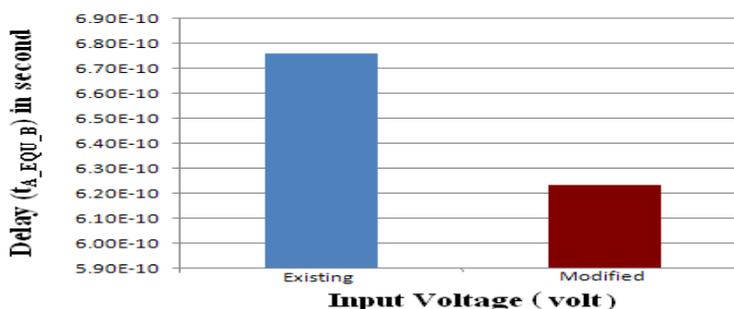


Figure 21. Delay ($t_{A_EQU_B}$) with Input Voltage

The graphs shown in Fig. 20 & Fig. 21 reveal that delay of modified 64-bit comparator design at 1 volt input voltage is remarkably reduced than existing 64-bit comparator design. In Fig. 20, delay is reduced 1.4 %. In Fig. 21, delay is reduced 7.8 %.

After simulation of both the designs final results are obtained for delay and power consumption with 30°C temperature. Simulation with temperature has been done at 1 volt input voltage. Delay comparison of modified and existing 64-bit comparator designs is shown in Fig. 22 & Fig. 23. Simulated data for these graphs is given in Table II.

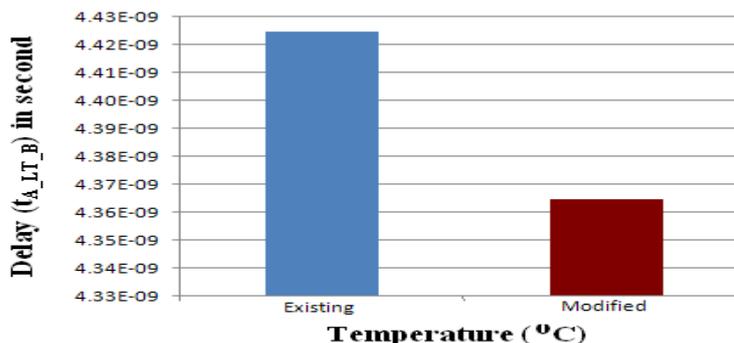


Figure 22. Delay ($t_{A_LT_B}$) with Temperature

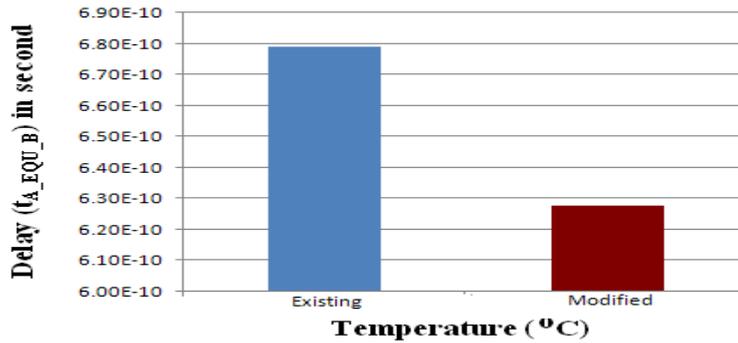


Figure 23. Delay ($t_{A_EQU_B}$) with Temperature

The graphs shown in Fig. 22 & Fig. 23 reveal that delay of modified 64-bit comparator design at 30°C temperature is remarkably reduced than existing 64-bit comparator design. In Fig. 22, delay is reduced 1.4%. In Fig. 23, delay is reduced 7.5%.

After simulation of both the designs final results are obtained for delay and power consumption with 50MHz frequency. Simulation with frequency has been done at 1 volt input voltage. Delay comparison of modified and existing 64-bit comparator designs is shown in Fig. 24 & Fig. 25. Simulated data for these graphs is given in Table III.

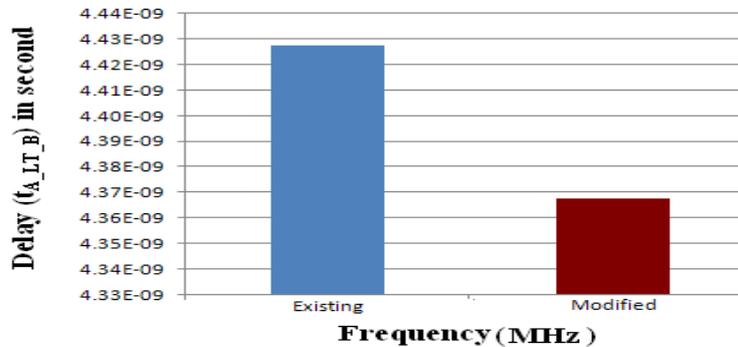


Figure 24. Delay ($t_{A_LT_B}$) with Frequency

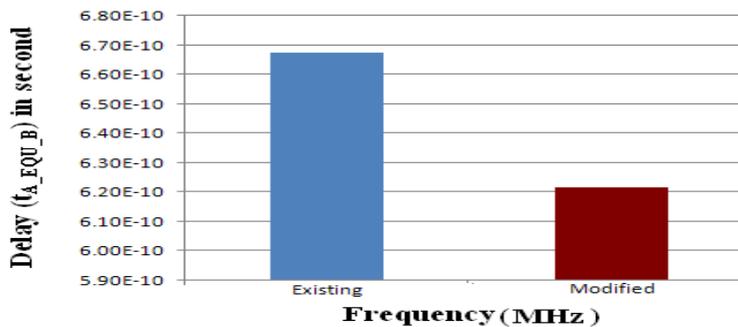


Figure 25. Delay ($t_{A_EQU_B}$) with Frequency

The graphs shown in Fig. 24 & Fig. 25 reveal that delay of modified 64-bit comparator design at 50MHz frequency is remarkably reduced than existing 64-bit comparator design. In Fig. 24, delay is reduced 1.4%. In Fig. 25, delay is reduced 6.8%.

VI. CONCLUSION

In modified design, at 1 volt input voltage delay for output “A less than B” ($t_{A_LT_B}$) is reduced 1.4% and delay for output “A equal to B” ($t_{A_EQU_B}$) is reduced 7.8% in comparison to existing design. Similarly, at 30°C temperature delay for output “A less than B” ($t_{A_LT_B}$) is reduced 1.4% and delay for output “A equal to B” ($t_{A_EQU_B}$) is reduced 7.5%. And also at 50MHz frequency delay for output “A less than B” ($t_{A_LT_B}$) is reduced 1.4% and delay for output “A equal to B” ($t_{A_EQU_B}$) is reduced 6.8% in comparison to existing design. Hence, superiority of modified design is maintained for temperature and frequency also. All of the reduction in delay is obtained after sacrificing power consumption and transistor count. But still modified design gives better result (for delay) than existing design. Therefore, modified 64-bit binary comparator design can be better option for high-speed applications.

REFERENCES

- [1] M. Morris Mano “*Digital Design*”. (Pearson Education Asia. 3rd Ed, 2002).
- [2] R. Zimmermann and W. Fichtner, “Low Power Logic Styles: CMOS Versus Pass Transistor Logic” *IEEE Journal of Solid State Circuits*, Vol.32, No.7, pp1079-1090, July 1997.
- [3] S. Kang and Y. Leblebici “*CMOS Digital Integrated Circuit, Analysis and Design*” (Tata McGraw-Hill, 3rd Ed, 2003).
- [4] A. Bellaouar and Mohamed I. Elmasry, “*Low Power Digital VLSI Design: Circuits and Systems*” (Kluwer Academic Publishers, 2nd Ed, 1995).
- [5] C.-H. Huang and J.-S. Wang, “High-performance and power-efficient CMOS comparators,” *IEEE J. Solid-State Circuits*, vol. 38, no.2, pp. 254–262, Feb. 2003.
- [6] H.-M. Lam and C.-Y. Tsui, “High-performance single clock cycle CMOS comparator,” *Electron. Lett.*, vol. 42, no. 2, pp. 75–77, Jan. 2006.
- [7] H.-M. Lam and C.-Y. Tsui, “A MUX-based high-performance single cycle CMOS comparator,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 7, pp.591-595 July 2007.
- [8] Pierce Chuang, David Li, and Manoj Sachdev, Fellow, IEEE “A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Binary Comparator” *IEEE Transactions on Circuits And Systems— II: Express Briefs*, Vol. 59, No. 2, February 2012.
- [9] F. Frustaci, S. Perri, M. Lanuzza, and P. Corsonello, “A new low-power high - speed single – clock -cycle binary comparator,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 317–320.
- [10] S. Perri and P. Corsonello, “Fast low-cost implementation of single-clock-cycle binary comparator,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 12, pp. 1239–1243, Dec. 2008.