High-Speed 64-Bit Binary Comparator using Three Stages with CMOS Logic Style

Anjuli (Student Member IEEE)
E&CE Department
FET-MITS (Deemed University)
Lakshmangarh, Sikar,
Rajasthan (India)

Satyajit Anand
E&CE Department
FET-MITS (Deemed University)
Lakshmangarh, Sikar,
Rajasthan (India)

ABSTRACT
High-speed 64-bit binary comparator using three stages with CMOS logic style 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 have been 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**.**

1. 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. 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].

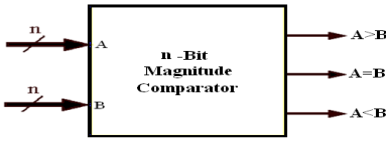

Fig 1: Block Diagram of n-Bit Magnitude Comparator

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^{o}C temperature and 50MHz frequency at 90nm technology in Tanner EDA Tool.

2. 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.

3. 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:

$$B_{Big} = \overline{A_1}\, B_1 + \left(\overline{A_1 \oplus B_1}\right).\left(\overline{A_0}\, B_0\right) \quad (1)$$

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

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:

$$C_{out} = AB + (A \oplus B).C_{in} = G + P.C_{in} \quad (3)$$

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):

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

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

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

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……..63.
Put these two values from equations (7) & (8) in equations (1) & (2).

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

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

Where j = 0……..31.
G & P signals can be further combined to form group G & P signals.

$$B_{Big\,[3:\,0]} = \overline{A_3}\, B_3 + \overline{(A_3 \oplus B_3)}.\overline{(A_2\, B_2)} + \overline{(A_3 \oplus B_3)}.\overline{(A_2 \oplus B_2)}.\overline{(A_1\, B_1)} + \overline{(A_3 \oplus B_3)}.\overline{(A_2 \oplus B_2)}.\overline{(A_1 \oplus B_1)}.\overline{(A_0\, B_0)}$$

$$B_{Big\,[3:0]} = \overline{A_3}\, B_3 + \overline{(A_3 \oplus B_3)}.\left[\overline{A_2}\, B_2 + \overline{(A_2 \oplus B_2)}.\{\overline{A_1}\, B_1 + \overline{(A_1 \oplus B_1)}.\overline{(A_0\, B_0)}\}\right]$$

$$B_{Big\,[3:0]} = G_3 + EQ_3.\{G_2 + EQ_2.(G_1 + EQ_1.G_0)\}$$

$$B_{Big\,[3:0]} = B_{Big\,[3:2]} + EQ_{[3:2]}.B_{Big\,[1:0]} \quad (11)$$

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

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

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

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

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 $j = 0 \ldots 3$, 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).

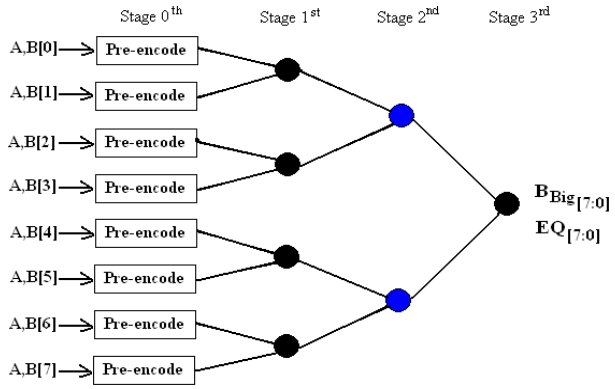

Fig 2: Tree-Diagram of 8-Bit Binary Comparator

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

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

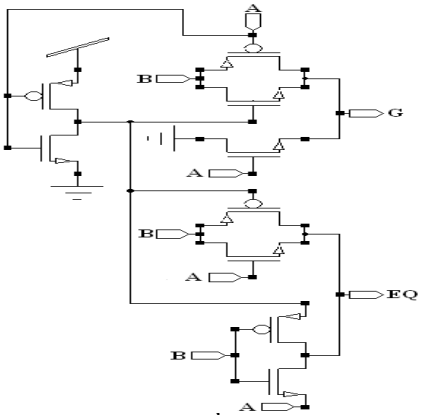

Fig 3: Schematic of Stage 0^{th} of Existing 64-Bit Binary Comparator

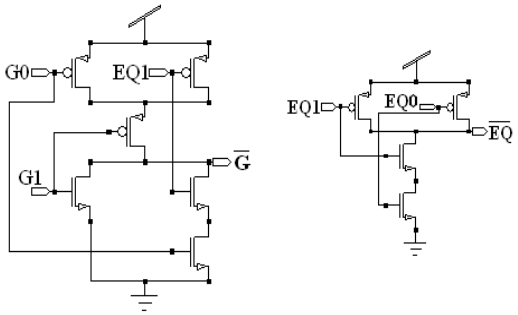

Fig 4: Schematic of Stage 1^{st} of Existing 64-Bit Binary Comparator

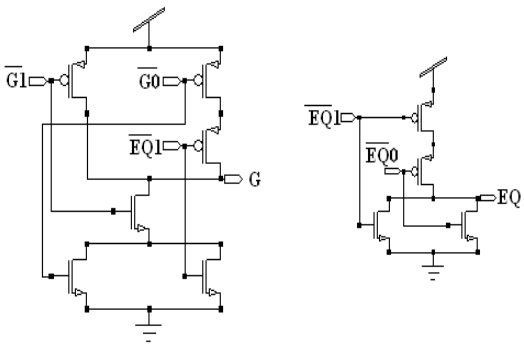

Fig 5: Schematic of Stage 2^{nd} of Existing 64-Bit Binary Comparator

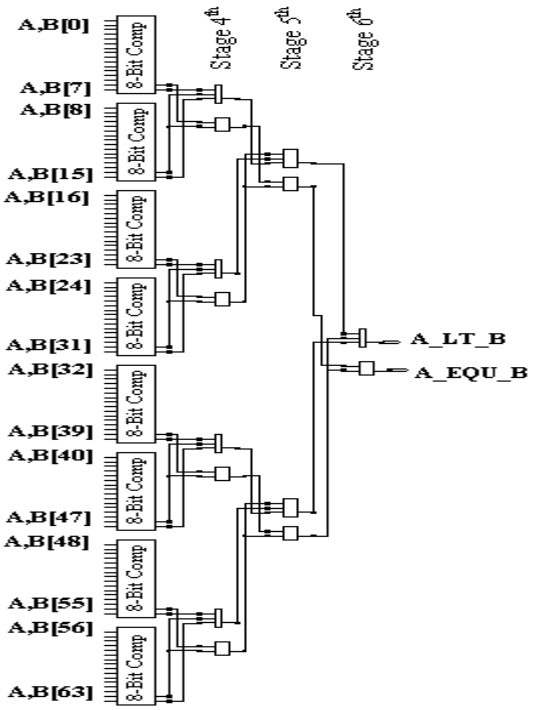

Fig 6: Schematic of Existing 64-Bit Binary Comparator

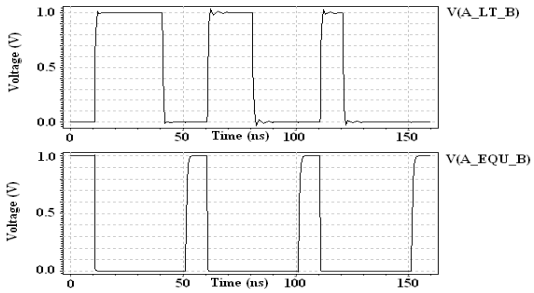

Fig 7: Waveforms of Existing 64-Bit Binary Comparator

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") at that time. Simulation results for this design are given in Table 3 – Table 5 for conclusion.

4. MODIFIED 64-BIT BINARY COMPARATOR DESIGN

Some modifications have been 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 having three stages (Stage A, Stage B and stage C) in place of tree-based structure. Fig. 8 shows logic diagram of modified 64-bit binary comparator.

In modified design, three stages have been used. In stage A, eight 8-bit comparators have been used to provide "A less than B" and "A equal to B" outputs (A_LT_B and A_EQU_B). In stage B, one NAND gate is used to provide "A equal to B" output (A_EQU_B) of 64-bit comparator design and also seven AND gate have been used to provide input for stage C. In stage C, one NOR gate has been used to provide "A less than B" output (A_LT_B) of 64-bit comparator design.

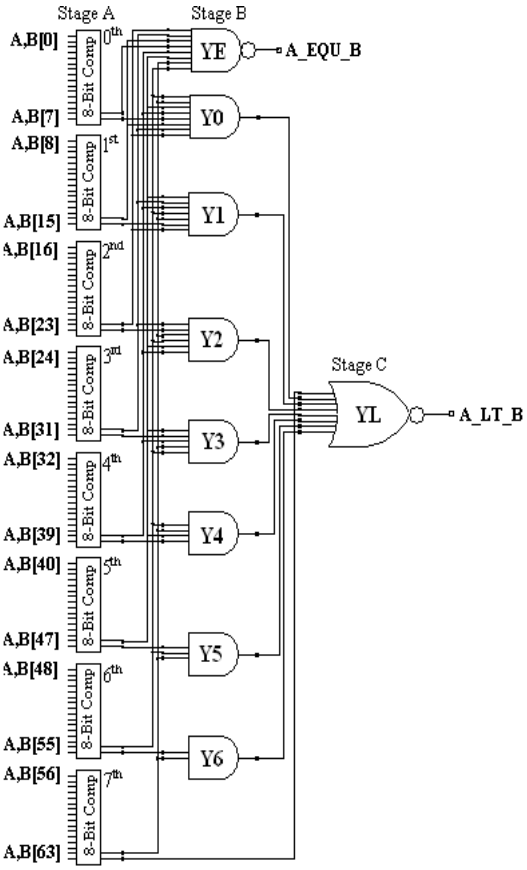

Fig 8: Logic Diagram of Modified 64-Bit Binary Comparator

For this modified design, In stage A, all three basic stages (0^{th}, 1^{st}, 2^{nd}) have been implemented using CMOS logic style. Means stage 0^{th} of modified 64-bit comparator design have been implemented using CMOS logic style (as in Fig. 9) that was implemented using modified PTL style in existing design. Remaining two stages (1^{st} & 2^{nd}) are exactly same as existing 64-bit comparator design Fig. 4 and Fig. 5. Description of this design is given in tabular form in Table 2.

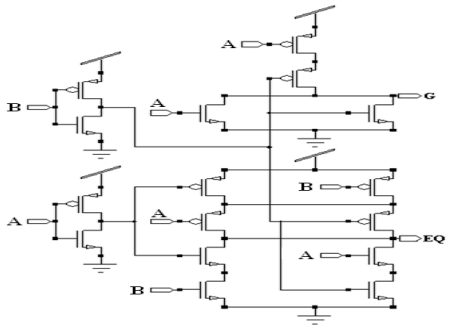

Fig 9: Schematic of Stage 0^{th} of Stage A of Modified 64-Bit Binary Comparator

In stage B, the "A less than B" output (A_LT_B) of 0^{th} 8-bit comparator and "A equal to B" outputs (A_EQU_B) of seven (from 1^{st} to 7^{th}) 8-bit comparators are given to AND gate Y0 that produces input for NOR gate YL. This 8-input AND gate has been 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. 10.

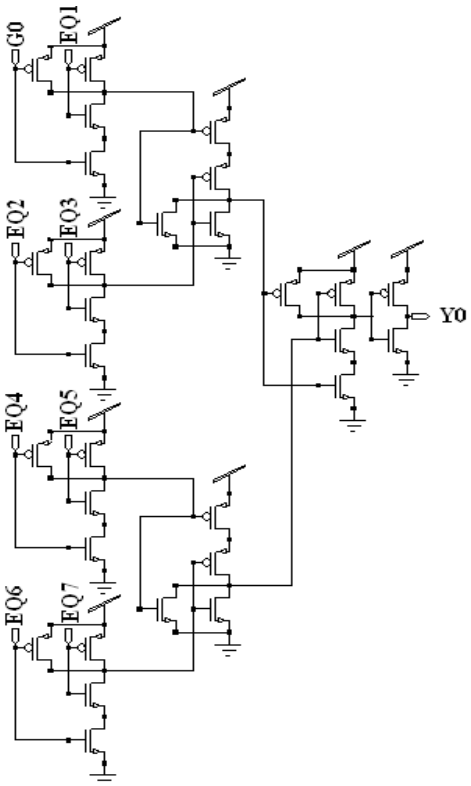

Fig 10: Schematic of AND Gate Y0 of Modified 64-Bit Binary Comparator

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 has been 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.11.

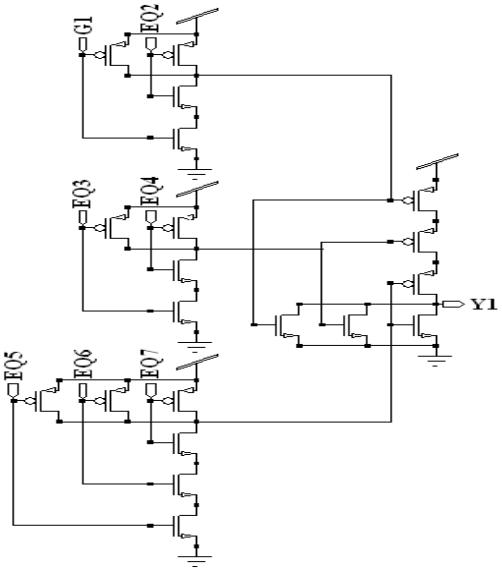

Fig 11: Schematic of AND Gate Y1 of Modified 64-Bit Binary Comparator

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 has been 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.12.

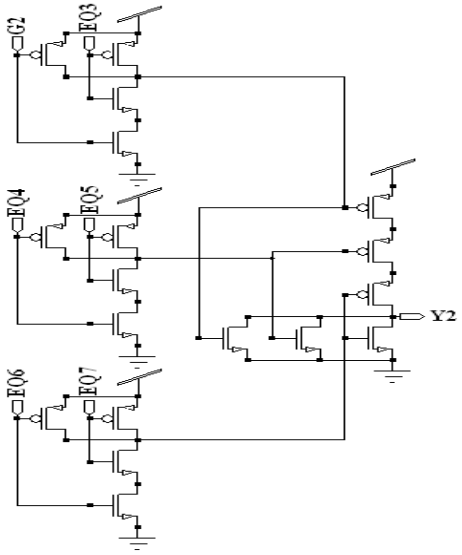

Fig 12: Schematic of AND Gate Y2 of Modified 64-Bit Binary Comparator

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 has been 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.13.

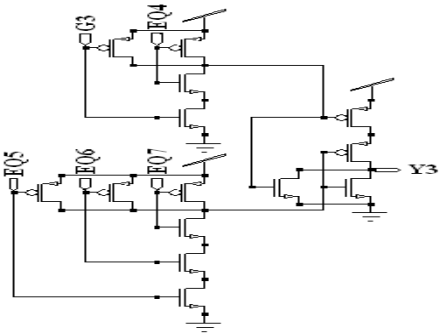

Fig 13: Schematic of AND Gate Y3 of Modified 64-Bit Binary Comparator

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 has been 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.14.

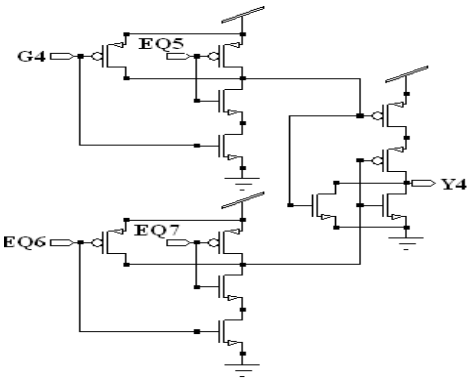

Fig 14: Schematic of AND Gate Y4 of Modified 64-Bit Binary Comparator

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 has been 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.15.

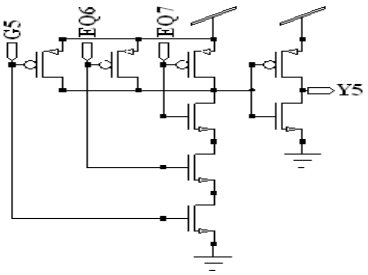

Fig 15: Schematic of AND Gate Y5 of Modified 64-Bit Binary Comparator

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 has been 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.16.

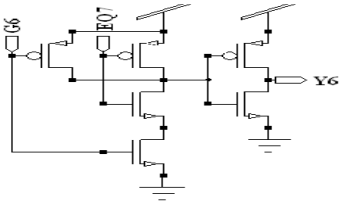

Fig 16: Schematic of AND Gate Y6 of Modified 64-Bit Binary Comparator

The "A equal to B" outputs (A_EQU_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_EQU_B) of modified 64-bit binary comparator. This 8-input NAND gate has been 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.

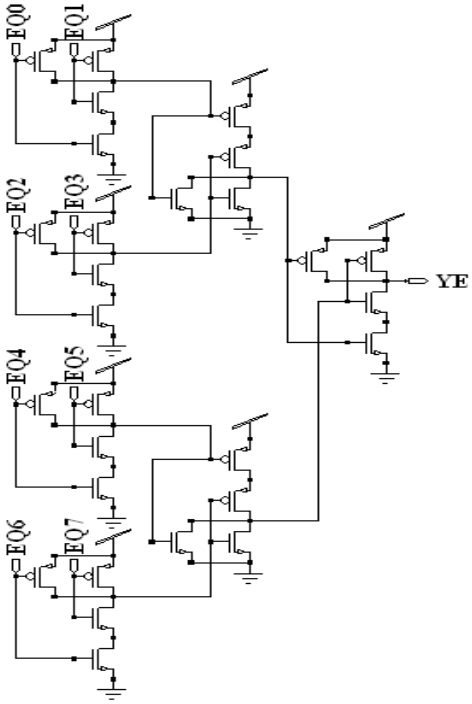

Fig 17: Schematic of NAND Gate YE of Modified 64-Bit Binary Comparator

In stage C, 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 has been 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.18.

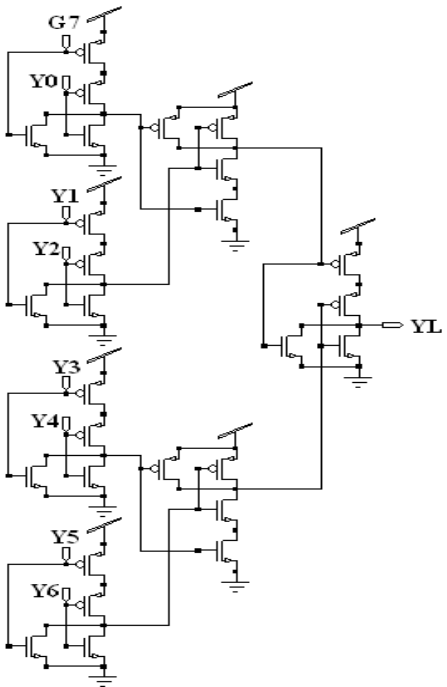

Fig 18: Schematic of NOR Gate YL of Modified 64-Bit Binary Comparator

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 1748 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.19.

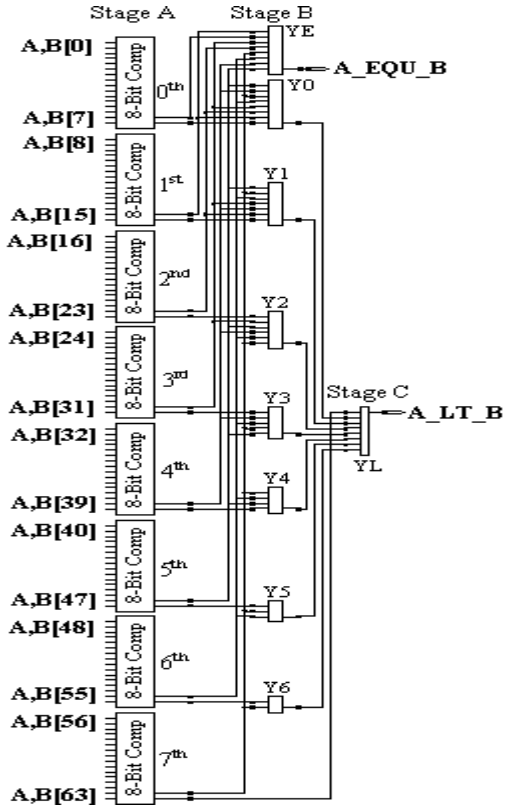

Fig 19: Schematic of Modified 64-Bit Binary Comparator

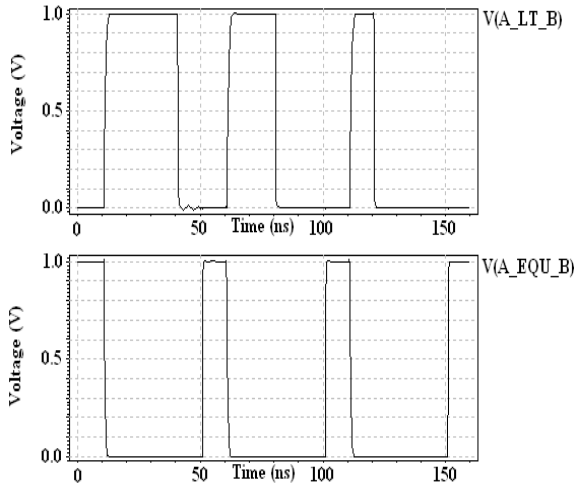

Fig 20: Waveforms of Modified 64-Bit Binary Comparator

According to input bit stream, waveforms of modified 64-bit binary comparator are obtained and shown in Fig.20. 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") at that time. Simulation results for modified 64-bit binary comparator design are given in tabular form in Table 3 –Table 5.

5. SIMULATION AND COMPARISON

After simulation of both the designs final results are obtained for delay and power consumption and are shown in Table 3 – Table 5. Simulations have been carried out at 90nm technology in Tanner EDA Tool.

Table 1. Description of Existing 64-Bit Binary Comparator design

Detail	Stage 0th	Stage 1st	Stage 2nd	Transistor Count
Design	Using MPTL Style	Using CMOS Style	Using CMOS Style	1206
Nature of output	Actual	Inverse	Actual	

Table 2. Description of Stage A of Modified 64-Bit Binary Comparator Design

Detail	Stage 0th	Stage 1st	Stage 2nd	Transistor Count
Design	Using CMOS Style	Same as Existing	Same as Existing	1584
Nature of output	Actual	Inverse	Actual	

Table 3. Simulation Data with 1volt Input Voltage

Design	Power Consumption (watt)	Delay Time (second)	
		$t_{A\_LT\_B}$	$t_{A\_EQU\_B}$
Existing	8.9563e-6	4.4290e-9	6.7628e-10
Modified	2.1059e-5	4.3226e-9	6.5424e-10

Table 4. Simulation Data with 30°C Temperature

Design	Power Consumption (watt)	Delay Time (second)	
		$t_{A\_LT\_B}$	$t_{A\_EQU\_B}$
Existing	9.1340e-6	4.4250e-9	6.7904e-10
Modified	2.1305e-5	4.3227e-9	6.6169e-10

Table 5. Simulation Data with 50MHz Frequency

Design	Power Consumption (watt)	Delay Time (second)	
		$t_{A\_LT\_B}$	$t_{A\_EQU\_B}$
Existing	9.0262e-6	4.4291e-9	6.6731e-10
Modified	2.0877e-5	4.3226e-9	6.5436e-10

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. 21 & Fig. 22. Simulated data for these graphs is given in Table 3.

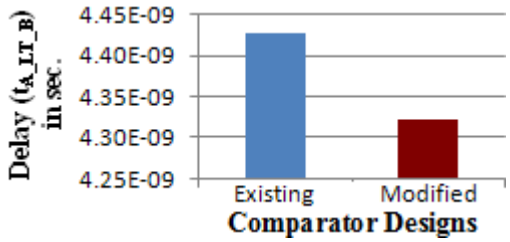

Fig 21: Delay ($t_{A\_LT\_B}$) vs Comparator Designs

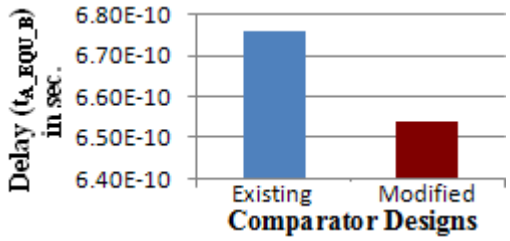

Fig 22: Delay ($t_{A\_EQU\_B}$) vs Comparator Designs

The graphs shown in Fig.21 & Fig.22 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.21, delay is reduced 2.4 %. In Fig.22, delay is reduced 3.3%.

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. 23 & Fig. 24. Simulated data for these graphs is given in Table 4.

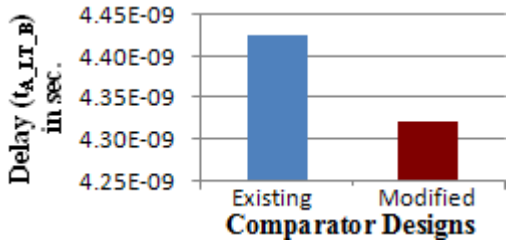

Fig 23: Delay ($t_{A\_LT\_B}$) vs Comparator Designs

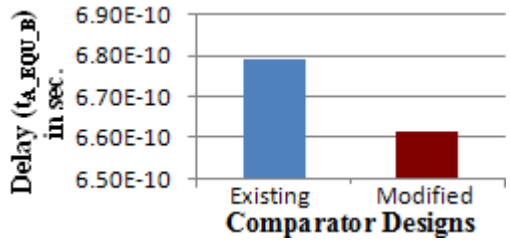

Fig 24: Delay ($t_{A\_EQU\_B}$) vs Comparator Designs

The graphs shown in Fig.23 & Fig.24 reveal that delay of modified 64-bit comparator design at 30°C temperature is remarkably reduced than existing 64-bit comparator design. In Fig.23, delay is reduced 2.3%. In Fig. 24, delay is reduced 2.6%.

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. 25 & Fig. 26. Simulated data for these graphs is given in Table 5.

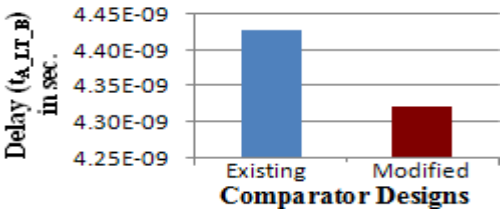

Fig 25: Delay ($t_{A\_LT\_B}$) vs Comparator Designs

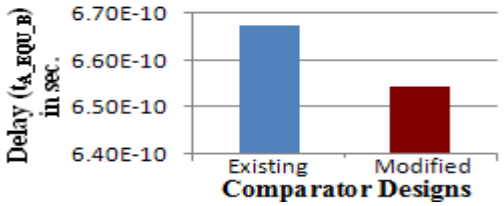

Fig 26: Delay ($t_{A\_EQU\_B}$) vs Comparator Designs

The graphs shown in Fig. 25 & Fig. 26 reveal that delay of modified 64-bit comparator design at 50MHz frequency is remarkably reduced than existing 64-bit comparator design. In Fig.25, delay is reduced 2.4%. In Fig.26, delay is reduced 2.0 %.

6. CONCLUSION

In modified design, at 1 volt input voltage delay for output "A less than B" ($t_{A\_LT\_B}$) is reduced 2.4 % and delay for output "A equal to B" ($t_{A\_EQU\_B}$) is reduced 3.3 % in comparison to existing design. Similarly, at 30°C temperature delay for output "A less than B" ($t_{A\_LT\_B}$) is reduced 2.3 % and delay for output "A equal to B" ($t_{A\_EQU\_B}$) is reduced 2.6 %. And also at 50MHz frequency delay for output "A less than B" ($t_{A\_LT\_B}$) is reduced 2.4 % and delay for output "A equal to B" ($t_{A\_EQU\_B}$) is reduced 2.0 % 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.

7. 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., pp. 317–320, 2010.

[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.