High-Speed Tree-based 64-Bit Binary Comparator using New Approach

Anjuli
M.Tech. VLSI Design
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 tree-based 64-bit binary comparator using new approach 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.

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 [6]. 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) [6]. The outcome of comparison is specified by three binary variables that indicate whether A>B, A=B, or A<B.

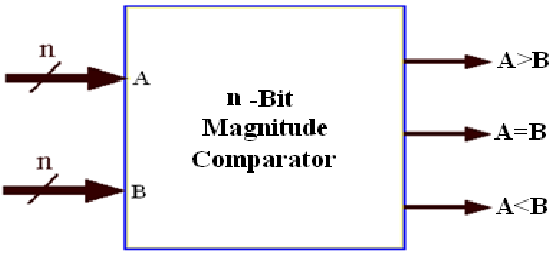

Fig 1: Block Diagram of n-Bit Magnitude Comparator

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 [6].

The logic style used in logic gates basically influences the speed, size, power dissipation, and the wiring complexity of a circuit. Circuit size depends on the number of transistors and their sizes and on the wiring complexity. 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 [7].

In order to differentiate all three designs existing and modified, simulations are carried out for power and delay at 1 volt supply voltage (and input voltage) 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 using tree structure logic.

3. EXISTING 64-BIT BINARY COMPARATOR DESIGN

64-bit Comparator in reference [1], [2], [3] 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:

$$B_{Big} = \overline{A}_1 B_1 + \left(\overline{A_1 \oplus B_1}\right) \cdot \left(\overline{A}_0 B_0\right) \quad (1)$$

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

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:

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

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):

$$G_1 = \overline{A}_1 B_1$$
(4)

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

$$C_{in} = \overline{A}_0 B_0$$
(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:

$$G_{[i]} = \overline{A}_{[i]} B_{[i]}$$
(7)

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

Where i = 0………..63.

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

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

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

Where j = 0………..31.

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

$$\begin{aligned} B_{Big\,[3:0]} = {} & \overline{A}_3 B_3 + \left(\overline{A_3 \oplus B_3}\right) \cdot \left(\overline{A}_2 B_2\right) + \left(\overline{A_3 \oplus B_3}\right) \\ & \cdot \left(\overline{A_2 \oplus B_2}\right) \cdot \left(\overline{A}_1 B_1\right) + \left(\overline{A_3 \oplus B_3}\right) \\ & \cdot \left(\overline{A_2 \oplus B_2}\right) \cdot \left(\overline{A_1 \oplus B_1}\right) \cdot \left(\overline{A}_0 B_0\right) \\ = {} & \overline{A}_3 B_3 + \left(\overline{A_3 \oplus B_3}\right) \\ & \cdot \left[\overline{A}_2 B_2 + \left(\overline{A_2 \oplus B_2}\right) \right. \\ & \left. \cdot \left\{\overline{A}_1 B_1 + \left(\overline{A_1 \oplus B_1}\right) \cdot \overline{A}_0 B_0\right\}\right] \\ = {} & G_3 + EQ_3 \cdot \{G_2 + EQ_2 \cdot (G_1 + EQ_1 \cdot G_0)\} \end{aligned}$$

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

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

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

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

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

Fig. 2 shows 8-bit version of existing tree-based comparator structure and Fig. (3, 4 & 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 (including inverters).

In above 8-bit example circuitry, the first stage comparison circuit implements equations (9, 10) for *j* = 0. . . 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 the third stage according to equations (13 & 14).

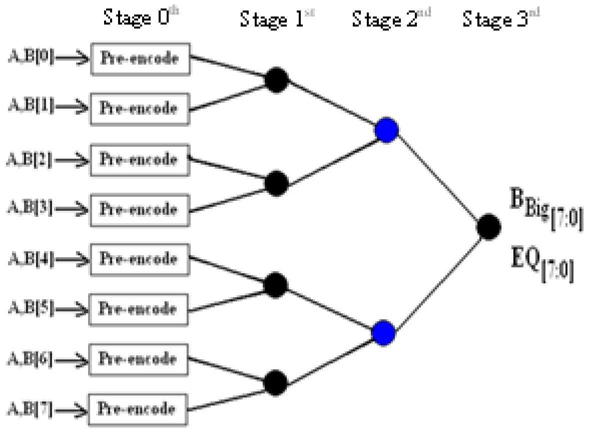

Fig 2: Tree-Diagram of 8-Bit Binary Comparator

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 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 Figure 3) is employed to produce "less than" & "equal to" outputs. Outputs of stage 0th are 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 Figure 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.

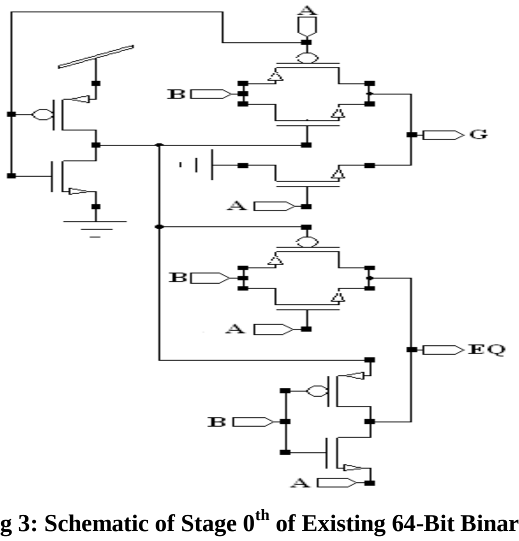

Fig 3: Schematic of Stage 0th of Existing 64-Bit Binary Comparator

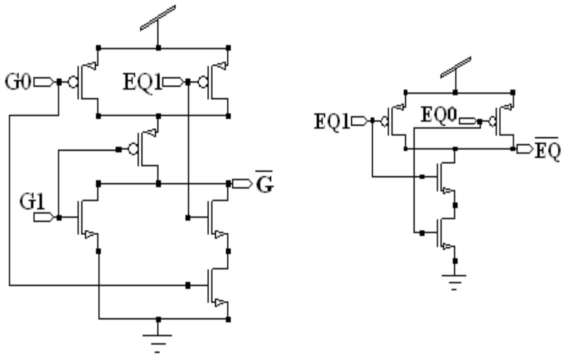

Fig 4: Schematic of Stage 1st of Existing 64-Bit Binary Comparator

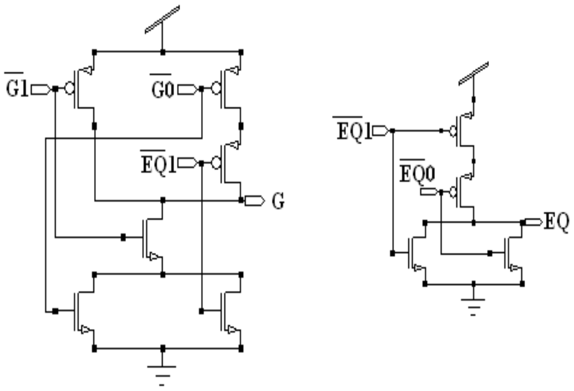

Fig 5: Schematic of Stage 2nd of Existing 64-Bit Binary Comparator

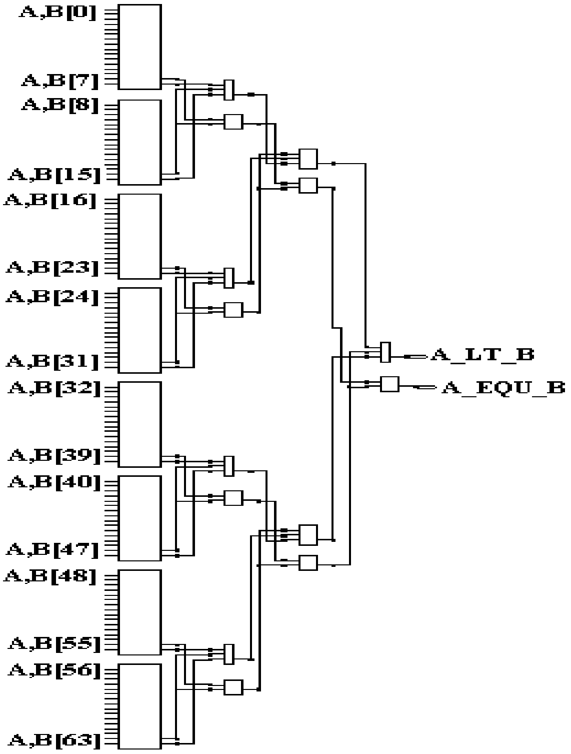

Fig 6: Schematic of Existing 64-Bit Binary Comparator

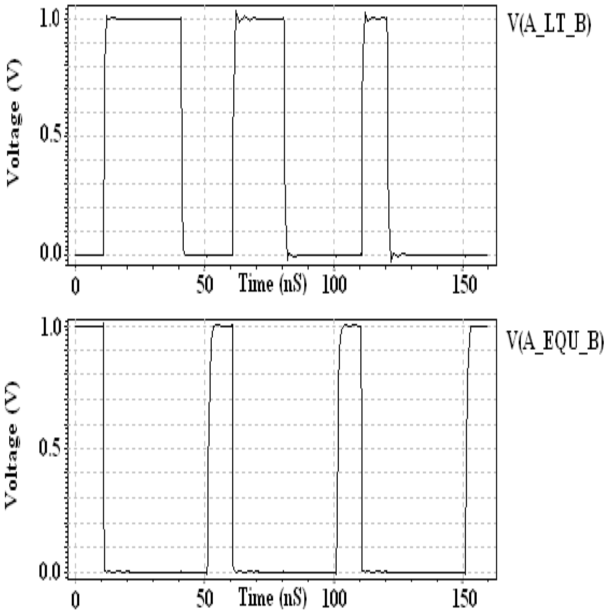

Fig 7: Waveforms of Existing 64-Bit Binary Comparator

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”). Simulation results for this design are given in table 1 for conclusion.

4. MODIFIED 64-BIT BINARY COMPARATOR DESIGN

Some modifications are done for stage 0th, stage 1st & stage 2nd in existing 64-bit comparator design [1] to improve the speed of the circuit.

Main idea behind PTL (pass transistor logic) is to use purely NMOS Pass Transistors network for logic operation [5]. The basic difference of pass-transistor logic style compared to the CMOS logic style is that the source side of the logic transistor networks is connected to some input signals instead of the power lines. In this design style, Transistors are act as switch to pass logic levels from input to output [4].

For this design, stage 0th is same as existing 64-bit comparator design & implemented using modified PTL style (Fig. 3) giving output in actual manner. Stage 1st is implemented using modified pass transistor logic style (MPTL) giving output in actual manner as in Fig. 8. MPTL means extra PMOS circuitry is used in pass transistor logic style circuitry to pass logic high (“1”) from input to output. Stage 2nd is same as stage 1st of existing 64-bit comparator design & implemented using CMOS logic style giving output in inverse manner as in Fig. 9.

Final output for 64-bit comparator of this design is obtained in inverse form. So, at the end of schematic design of 64-bit comparator two inverters (Fig. 10) are required to provide actual form of output waveform. Schematic (using instances of each stages) of modified 64-bit comparator design (Fig. 11) is same as existing 64-bit comparator design with only the difference of including inverters.

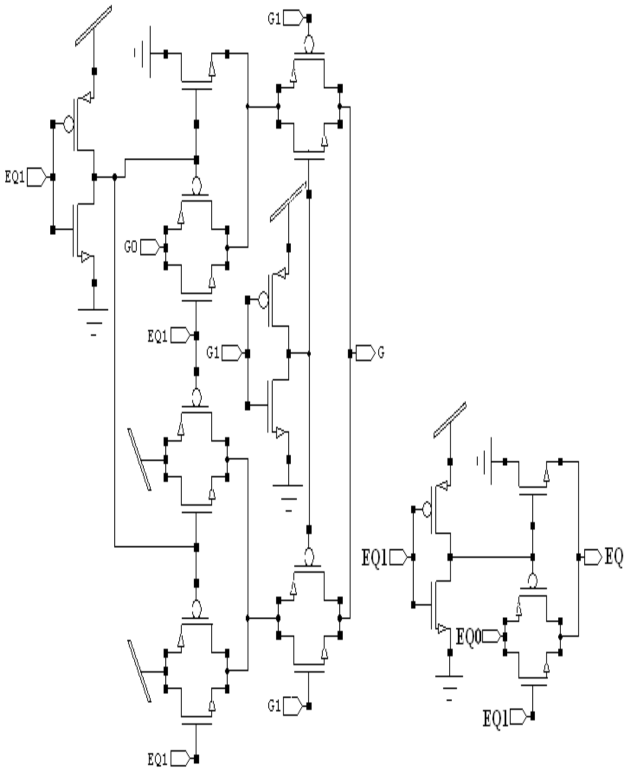

Fig 8: Schematic of Stage 1st of Modified 64-Bit Binary Comparator

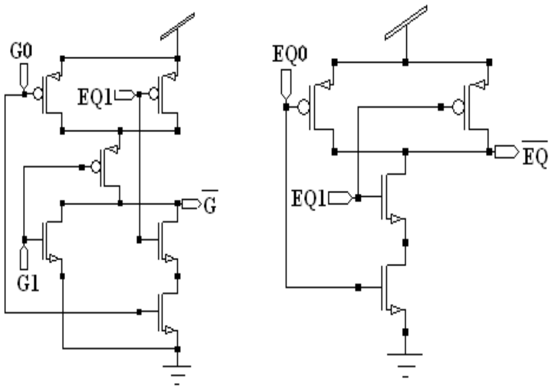

Fig 9: Schematic of Stage 2nd of Modified 64-Bit Binary Comparator

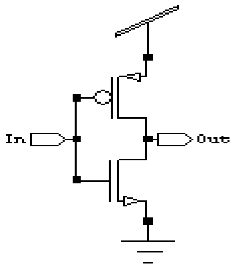

Fig 10: Schematic of Inverter of Modified 64-Bit Binary Comparator

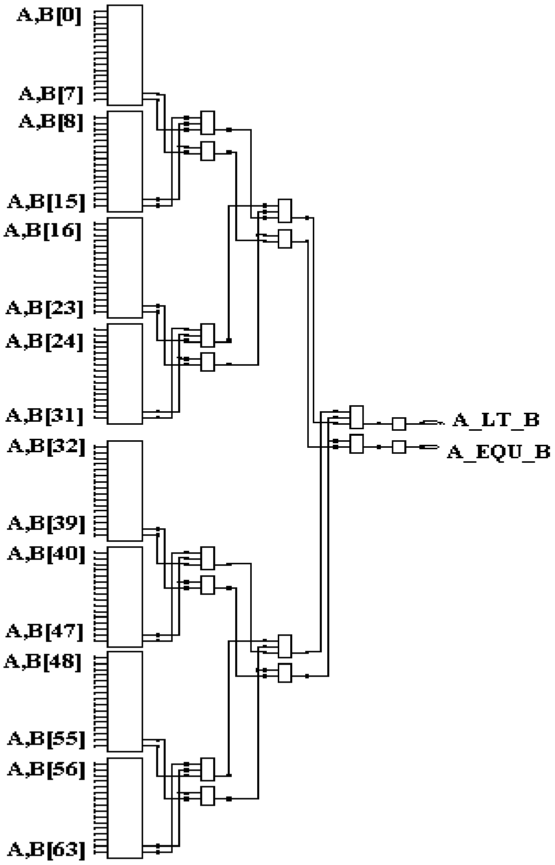

Fig 11: Schematic of Modified 64-Bit Binary Comparator

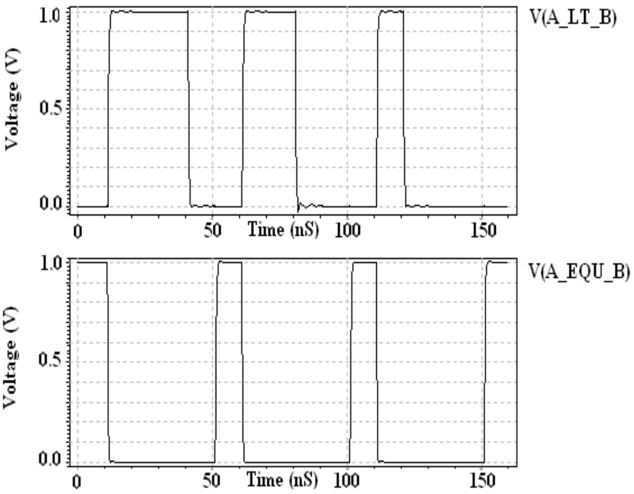

Fig 12: Waveforms of Modified 64-Bit Binary Comparator

According to input bit stream, waveforms of modified 64-bit binary comparator are obtained and shown in Fig. 12. Input bit stream for this design is same as in existing design of 64-bit comparator. Output waveforms of this design produce same position of 1,s and 0,s as in waveforms of existing design for each input bit. Simulation results for this design are given in table 1 for conclusion.

Table 1. Simulation results for both the designs

Designs	Power Consumption (watts)	Delay (seconds)		Transistor Count
		$t_{A\_LT\_B}$	$t_{A\_EQU\_B}$	
Existing	8.9563e-006	4.4290e-009	6.7628e-010	1206
Modified	1.2183e-005	4.3413e-009	6.5118e-010	1630

5. SIMULATIONS AND COMPARISION

After simulation of both the designs final results are obtained for power consumption and delay and are shown in table 1.

In modified design, delay ($t_{A\_LT\_B}$) is reduced from 4.4290e-009 to 4.3413e-009 means 2.0 % reduction in comparison to existing design and delay ($t_{A\_EQU\_B}$) is reduced from 6.7628e-010 to 6.5118e-010 means 3.7 % reduction in comparison to existing design.

6. CONCLUSION

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 64-bit binary comparator design. In modified design, delay ($t_{A\_LT\_B}$) is reduced 2.0 % and delay ($t_{A\_EQU\_B}$) is reduced 3.7 % in comparison to existing design.

7. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] S. Kang and Y. Leblebici "CMOS Digital Integrated Circuit, Analysis and Design" Tata McGraw-Hill, 3rd Ed, 2003.

[6] M. Morris Mano "Digital Design". Pearson Education Asia. 3rdEd, 2002.

[7] A. Bellaouar and Mohamed I. Elmasry, Low Power Digital VLSI Design: Circuits and Systems," Kluwer Academic Publishers, 2nd Ed, 1995.