Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

Implementation of A 4n-Bit Comparator based on IC Type 74L85 using Linear Threshold Gate Tunneling Technology

Anup Kumar Biswas
Assistant Professor,
Department of Computer Science and Engineering
Kalyani Govt. Engineering College, Kalyani,
Nadia-741235, West Bengal, India

***Abstract*:- In this paper we will focus on the design of a 4n-bit comparator based on threshold logic functions using the specific behavior of a tunnel junction, i.e., the ability of controlling the transport of individual electron through a tunnel junction. We introduce an inverter based on single electron transistor, a novel design of an n-input linear threshold gate accommodating both positive and negative weights with 1 single tunnel junction and n+2 true capacitors, different gates like ANDs, OR, NOR and XOR. Every gates with their output waveforms are also introduced. All the time delays, their component values for different components are presented in tabular forms.**

Key words: linear threshold gate, electron-tunneling, comparator

1. INTRODUCTION

It is clear for those who are involved in research in semiconductor technology that the ever decreasing feature size and the corresponding increase in density of transistors facilitated many improvements in semiconductor based designs. We are able to realize that such improvement will eventually come to an end. For ensuring further feature size reduction, possible successor technologies with greater scaling potential like single electron tunneling are currently under the investigation of researchers. Single electron tunneling based circuits are centered regarding tunnel junctions, through which individual electrons can be tunneled in a controlled manner.

In our present era, high operating speed, low power consumption, and high integration density-based devices are financially indispensable in all the sectors of engineering and technology. Single Electron tunneling based device is one such equipment by which all Boolean logic gates can be implemented. It is a single electron which is adequate to store information in the single electron tunneling based device in the atmosphere of 0K. Power being consumed in the single electron tunneling circuits is very low when compared with (CMOS) circuits. The processing speed of a threshold Logic gate (TLG) will be nearly close to the electronic speed. The single electron transistor (SET) and LTG draws the attention of researchers, scientists or technologists to design and implement large scale circuits for the sake of the consumption of ultra-low power and their small size in different phases of applications. Tunneling events for TLG-based circuits happen when only a single electron tunnel through the tunnel junction under the proper bias voltage and multiple input voltages connected to the islands via small capacitors. For implementing a 4n-bit comparator circuit, LTG would be a best candidate for this case. Ultra-low noise is produced while tunneling an electron through the LTG based circuit. Multiple input logic gates, an XOR, are implemented, and based on them, a "comparator of 4-bits" is implemented and above all, with the help of this comparator a multiple of 4-bit i.e. 4n-bit comparator is presented in this paper.

2. INVERTER

Almost for every complex circuit, an inverter [3, 7, 8] is essential. An inverter is depicted in Fig. 1(a) which is made up of two single-electron transistors connected in series. The input voltage V_{in} is directly coupled to the islands (indicating by small dots) of the SET1 and SET2 through two capacitors C_1 and C_2 respectively. The island of each SET has a size smaller than 10 nm diameter of a superconductor metal like gold and their capacitances must be less than 10^{-17} F. The output terminal V_0 is connected to the common channel in between them and to the ground via a capacitor C_L to put down charging effects.

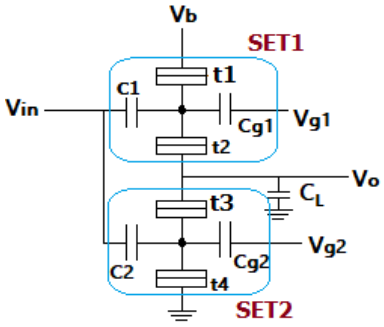

Fig.1 (a) an Inverter

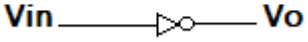

Fig.1 (b) Symbol of an Inverter

For the inverter, the parameters values chosen are: V_{g1}=0, V_{g2}=0.1× $\frac{q_e}{C}$, $C_L = 9C$, $t_4 = \frac{1}{10}C$, $t_3 = \frac{1}{2}C$, $t_2 = \frac{1}{2}C$, $t_1 = \frac{1}{10}C$, $C_1 = \frac{1}{2}C, C_2 = \frac{1}{2}C, C_{g1} = \frac{17}{4}C$ and $C_{g2} = \frac{17}{4}C$, R1 =R2=100KΩ. For simulation purpose, the value of C is taken 1aF in this work.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

The normal operation of the inverter is described as: - the output V_0 value will be high when the input voltage V_{in} is low and V_0 value will be low when the input voltage is high. To obtain this goal, we set the voltages $for\ V_{g1} = 0\ and\ V_{g2}$ =16mV along with the tuning gate voltages V_{in} both for SET1 and SET2 are applied. When V_{in} is low, the SET1 goes in conduction mode and the SET2 in Coulomb blockade. This effectively connects the output to voltage V_b and causes the output voltage to become high. Coulomb blockade interacts on the steady flow of current because whenever the high voltage (logic 1) is applied to the input, it causes to shift the induced charge on each of the islands of the two SETs by a fraction of an electron charge and keeps the SET1 in Coulomb blockade [4, 5, 11, 14, 16] and the other one (SET2) in conducting mode. As a consequence, the output shifts from high to low. The simulation result of an inverter is presented in Fig. 1(d).

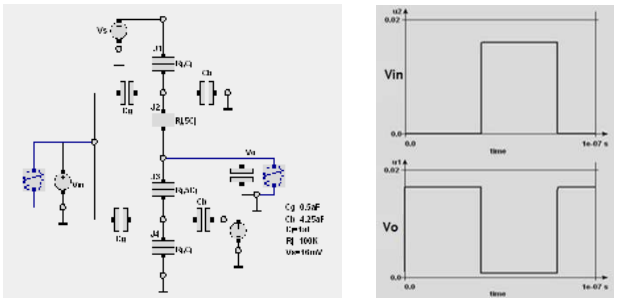

Fig. 1(c) Simulation set of Inverter (d) Simulation result

In this work we assume the Boolean logic inputs corresponding to the voltages like: logic "0" =0 Volts and logic "1"=0.1× $\frac{q_e}{C}$. We assume, for simulation and other purposes,

C=1aF then Logic "1"= $0.1\times\frac{1.602\times10^{-19}}{1\times10^{-18}}$ =$0.1\times1.602\times10^{-2}$=$16.02\times10^{-3}$ =16.02 ≅ 16 mV.

3. MULTIPLE INPUT THRESHOLD LOGIC GATE

The multiple input threshold logic gate [1, 2, 5, 7, 8, 9] required to build gates consists of a tunnel junction, two multiple inputs connected to the points 'a' and 'b'. Each input voltage V_k^P, for the upper side regarding the tunnel junction is connected to point "b" through the capacitance $C_k^P$and each input voltage V_l^n, for the lower side is connected to point "a" below the tunnel junction through the capacitor C_l^n. Bias voltage V_b is connected to the point "b" through a true capacitor C_b . Point "a" is grounded through a capacitor C_0 as shown in Fig. 2. The capacitor of tunnel junction is C_j. Using this capacitor based circuit and changing the suffix value of capacitors, we will be able to implement the linear threshold gate (LTG) which is the base element in constructing all gates and combinational circuits. The linear threshold gate is presented by the signum function of g(x) expressed by equations (1) and (2).

$$f(x) = \mathrm{sgn}\{g(x)\} = \begin{cases} 0, & if\ g(x) < 0 \\ 1, & if\ g(x) \geq 0 \end{cases} \quad \cdots\cdots (1)$$

$$g(x) = \sum_{k=1}^{n}(w_k \times x_k) - \emptyset \quad \cdots\cdots (2)$$

where x_k are being the n Boolean inputs and w_k are their corresponding n integer weights. The LTG compares the weighted sum of the inputs $\sum_{k=1}^{n}(w_k \times x_k)$ with the threshold value $\emptyset$. If the weighted sum value is more than or equal to the threshold or critical voltage value then the logic output of the LTG is "1", otherwise it is "0".

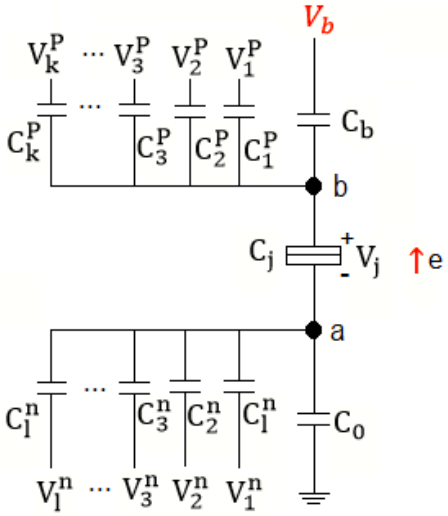

Fig. 2 Multiple input threshold logic gate

The tunnel junction capacitance C_j and the output capacitance C_0 are considered to be the two fundamental circuit elements in a LTG. The input signal vector elements $\{V_1^P, V_2^P, V_3^P, \dots, V_k^P\}$ are weighted by their corresponding vector element capacitances $\{C_1^P, C_2^P, C_3^P, \dots, C_k^P\}$ and added to the junction voltage, V_j. On the contrary, the input signal vector elements $\{V_1^n, V_2^n, V_3^n, \dots, V_l^n\}$ which are weighted by their respective vector capacitances $\{C_1^n, C_2^n, C_3^n, \dots, C_l^n\}$ are subtracted from the junction voltage, V_j, across the junction.

The critical voltage $V_c\ is$ needed to enable tunneling action. V_c acts as the intrinsic threshold of the tunnel circuit. The bias voltage V_b connected to tunnel junction through the true capacitance, C_b, is used to adjust the gate threshold to the desired value $\emptyset$. When a tunneling phenomenon occurs though the tunnel junction, an electron passes through the junction from "a" to "b" in upward direction.

We will use the following notations for the rest our discussion.

$$C_\Sigma^P = C_b + \sum_{k=1}^{g} C_k^P \quad \cdots\cdots (3)$$

$$C_\Sigma^n = C_0 + \sum_{l=1}^{h} C_l^n \quad \cdots\cdots (4)$$

$$C_T = C_\Sigma^P C_j + C_\Sigma^P C_\Sigma^n + C_j C_\Sigma^n \quad \cdots\cdots (5)$$

When we consider that all of the voltage sources (in Fig. 2) are 0 (or connected to ground), the circuit can be thought of combination of three capacitors , namely C_Σ^P, C_Σ^n and C_j , connected in series,. Here, C_T is represented by the sum of all 2-term products of these three capacitances C_Σ^P, C_Σ^n and C_j.

We are now interested in finding out the expression of critical voltage V_c of the tunnel junction. Suppose that the capacitance of the tunnel junction is C_j and the remainder of the circuit

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

bearing the equivalent capacitance is C_e, as observed from the tunnel junction's perspective, we can measure the critical voltage[7,8] for the tunnel junction as given below.

$$V_c = 0.5\frac{e}{C_j + C_e} \quad \cdots\cdots (6)$$

$$V_c = 0.5\frac{e}{C_j + (C_\Sigma^P || C_\Sigma^n)} = 0.5\frac{e}{C_j + \frac{(C_\Sigma^P) * (C_\Sigma^n)}{(C_\Sigma^P + C_\Sigma^n)}}$$

$$= 0.5\frac{e(C_\Sigma^P + C_\Sigma^n)}{C_j * (C_\Sigma^P + C_\Sigma^n) + (C_\Sigma^P) * (C_\Sigma^n)}$$

$$= 0.5\frac{e(C_\Sigma^P + C_\Sigma^n)}{C_T} \quad \cdots\cdots (7)$$

When we are able to calculate the voltage across the junction as V_j , a tunnel event will happen through this tunnel junction if and only if the following condition is satisfied.

$$|V_j| \geq V_c \quad \cdots\cdots (8)$$

If the junction voltage is less than the critical voltage i.e. $|V_j| < V_c$ there being no tunneling events through the tunnel junction. As a result, the tunneling circuit remains in a $stable\ state$.

Theoretically, threshold values are being integer numbers but for our purpose we can take it as real numbers, if necessary. And the threshold logic equations for two inputs AND, OR, NAND and NOR gates can be written in the following sections.

4. THRESHOLD LOGIC EQUATION / FUNCTION (TLF)

A **threshold logic function** (TLF) is a Boolean **function** that can be implemented by using a single TLG. The TLF can also be called **linearly** separable. An **unate function** is a Boolean **function** represented by a formula such that each variable appears either in the positive or in the negative form throughout the formula. As the equations (12), (16), (17), (18), (19), (20) and (21) have their variables either positive or negative for particular cases, so they are unate functions.

Threshold logic equation for AND gate: F(A,B)=A.B

Table-1

A	B	F(A,B)	t (threshold)
0	0	0	0
0	1	0	$t > w_B$
1	0	0	$t > w_A$
1	1	1	$w_B + w_A$>t

$t > w_B$ ····· (9)
$t > w_A$ ····· (10)
$w_B + w_A > t$ ····· (11)

As AND is positive logic we assume all the weights w_A, w_B and the threshold value t are positive.

If we assume w_A=1, w_B=1 and t=1.5, then all the equations from (9) to (11) are satisfied.

So we can write

g(x)= $\sum_{k=1}^{n}(w_k \times x_k)$ -∅

⟹g(A,B)={A. $w_A + B.w_B - t$} and the Threshold logic equation for AND gate is

$$AND(A,B) = sgn\{A + B - 1.5\} \quad \cdots\cdots (12)$$

Threshold logic equation for OR gate

Table-2

A	B	F(A,B)	t
0	0	0	0
0	1	1	$w_B > t$
1	0	1	$w_A > t$
1	1	1	$w_B + w_A > t$

For positive logic we assume weights of A and B are 1 each. Then from the three equations

$w_B > t$ ····· (13)
$w_A > t$ ····· (14) and
$w_B + w_A > t$ ····· (15)

If we assume w_B=1, w_A=1 and t=0.5, then the three equations in 4th column in Table-2 are satisfied. Hence the Threshold logic equation for OR gate is

$$OR(A,B) = sgn\{A + B - 0.5\} \quad \cdots\cdots (16)$$

Threshold logic equation for NAND gate

Table-3

A	B	F(A,B)= (AB)'	t
0	0	1	$0 \geq t$
0	1	1	$w_B \geq t$
1	0	1	$w_A \geq t$
1	1	0	$w_B + w_A < t$

As NAND is negative logic we assume that all the weights w_A, w_B and the threshold value t are negative. If we take the values $w_B = -1$, w_A=−1 and t=−1.5 $(-2 < t < -1)$ then the four equations in the 4th column of the table are satisfied. Hence the Threshold logic equation for NAND gate is

$$NAND(A,B) = sgn\{-A - B - (-1.5)\}$$
$$= sgn\{-A - B + 1.5\} \quad \cdots\cdots (17)$$

Threshold logic equation for NOR gate

Table-4

A	B	F(A,B)= (A∨B)'	t
0	0	1	$0 + 0 \geq t$
0	1	0	$w_B < t$
1	0	0	$w_A < t$
1	1	0	$w_B + w_A < t$

As NOR is negative logic we assume that all the weights w_A, w_B and the threshold value t are negative. If we take the values assume $w_B = -1$, $w_A = -1$ and t=-0.5 $(-1 < t < 0)$ then the four equations in the 4th column are satisfied. Hence the Threshold logic equation for NOR gate is

$$NOR(X,Y) = sgn\{-1A - 1B - (-0.5)\}$$
$$= sgn\{-A - B + 0.5\} \quad \cdots\cdots (18)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

Threshold logic equation for 3-input AND gate

Table-5

A	B	C	F(A,B,C)	t
0	0	0	0	0
0	0	1	0	$t > w_C$
0	1	0	0	$t > w_B$
0	1	1	0	$t > w_B + w_C$
1	0	0	0	$t > w_A$
1	0	1	0	$t > w_A + w_C$
1	1	0	0	$t > w_A + w_B$
1	1	1	1	$w_A + w_B + w_C \geq t$

As AND gate is a positive logic we shall assume that all the values of w_A, w_B, w_C and t are positive. If we take $w_A = 1$, $w_B = 1$, $w_C = 1$ and $t = 2.5$ (or any value in the range $2 < t < 3$), then all the inequality equations from are satisfied.

So the Threshold logic equation for 3-input AND gate is

$$AND(A,B,C) = sgn\{A + B + C - 2.5\} \cdots\cdots (19)$$

Threshold logic equation for 4-input AND Gate

Similarly, for 4-input AND gate threshold logic equation will be

$$AND(A,B,C) = sgn\{A + B + C + D - 0.5\} \cdots\cdots (20)$$

Threshold logic equation for 4-input OR Gate

Table-6

A	B	C	D	F(A,B,C,D)	t
0	0	0	0	0	0
0	0	0	1	1	$w_D \geq t$
0	0	1	0	1	$w_C \geq t$
0	0	1	1	1	$w_C + w_D \geq t$
0	1	0	0	1	$w_B \geq t$
0	1	0	1	1	$w_B + w_D \geq t$
0	1	1	0	1	$w_B + w_C \geq t$
0	1	1	1	1	$w_B + w_C + w_D \geq t$
1	0	0	0	1	$w_A \geq t$
1	0	0	1	1	$w_A + w_D \geq t$
1	0	1	0	1	$w_A + w_C \geq t$
1	0	1	1	1	$w_A + w_C + w_D \geq t$
1	1	0	0	1	$w_A + w_B \geq t$
1	1	0	1	1	$w_A + w_B + w_D \geq t$
1	1	1	0	1	$w_A + w_B + w_C \geq t$
1	1	1	1	1	$w_A + w_B + w_C + w_D \geq t$

As OR gate is a positive logic we shall assume that all the values of w_A, w_B, w_C and t are positive. If we take $w_A = 1$, $w_B = 1$, $w_C = 1$, $w_D = 1$ and $t = 0.5$ (or any value in the range $0 < t < 1$), then all the equations in column 5 of Table-6 are satisfied.

So the threshold logic equation for 4-input OR gate is

$$OR(A,B,C,D) = sgn\{A + B + C + D - 0.5\} \cdots\cdots (21)$$

5. THRESHOLD GATE-BASED IMPLEMENTATION

The threshold gate-based implementations of the Boolean Logic gates have the same basic circuit topology and we will draw and explain them in the subsequent sections. The threshold gate is built up of a bias capacitance C_b, a tunnel junction capacitance C_j , and an output capacitance C_0 . The two input AND and OR gates contain two input true capacitors. For both the AND and OR gates, the two input capacitors we are taking are $C_1^P = C_2^P = 0.5C$ for positively weighted inputs. Whereas, for the NAND and NOR gates, two capacitors hold the values as $C_1^n = C_2^n = 0.5C$ for negatively weighted inputs.

Every threshold gates is augmented with an inverter (Fig.1 (a) or 1(c)). The function of the inverter is to invert/buffer the output of a threshold gate. The logic function done by the buffered threshold gate is that the inverse of "which is done by the threshold gate itself". For instance, a buffered AND gate implements the $NAND$ function. For the remaining part of our discussion, when we refer to a logic function such as AND or OR, we imply the logic function performed by the entire gate (i.e., threshold gate plus an output buffer).

The parameters used for the implementations and simulations of different gates like AND, OR, NAND and NOR gates and other combinational or sequential circuits are given in Table-7 [6, 7].

Table-7

TLG	2-input AND	2-input OR	2-input NAND	2-input NOR	3-input OR
C_b	10.6C	11.7C	13.2C	11.7C	11.8C
C_0	8C	8C	9C	9C	7.8C
C_1^P	-	-	0.5C	0.5C	0.4C
C_2^p	-	-	0.5C	0.5C	-
C_1^n	0.5C	0.5C	-	-	0.5C
C_2^n	0.5C	0.5C	-	-	-
C_j	0.1C	0.1C	0.1C	0.1C	0.1C
$C_{g1} = C_{g2}$	0.5C	0.5C	0.5C	0.5C	0.5C
$C_2 = C_3$	0.1C	0.1C	0.1C	0.1C	0.1C
$C_2 = C_3$	0.5C	0.5C	0.5C	0.5C	0.5C
$C_1 = C_4$	4.25C	4.25C	4.25C	4.25C	4.25C
C_L	9C	9C	9C	9C	9C
$logic\ 0 = 0\ V,\ logic\ 1 = 16m\ V, R_j$=$10^5\Omega$, C_j=0.1aF, other capacitance values are in terms of C, where $C = 1\ aF$					

6. AND GATE

For implementing the AND gate we will use the parameters C_3=10.6C , $C_1^n = C_2^n$=0.5aF, $C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$, $C_0 = 8aF$, R_j=10^5 Ω in Fig. 4(a) and accordingly after running the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

simulator the output we get is given in Fig. 4(b).

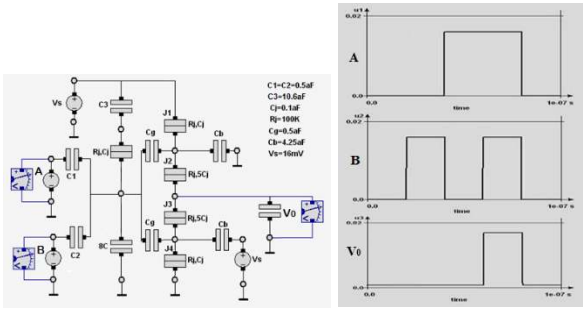

Fig.3 (a) AND Gate (b) Simulation result of AND gate

7. NAND GATE

For implementing the NAND gate, we will use the parameters $C_1^P = C_2^P$=0.5aF, $C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_b = C_3$ = 13.2aF, $C_L = 9aF$, $C_0 = 8aF$, R_j=10^5 Ω in Fig. 5(a) and accordingly after running the simulator the output we get is given in Fig.4(b).

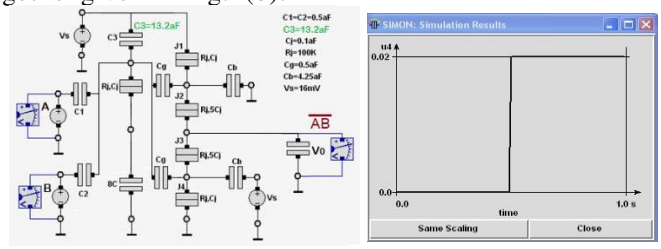

Fig.4 (a) NAND Gate

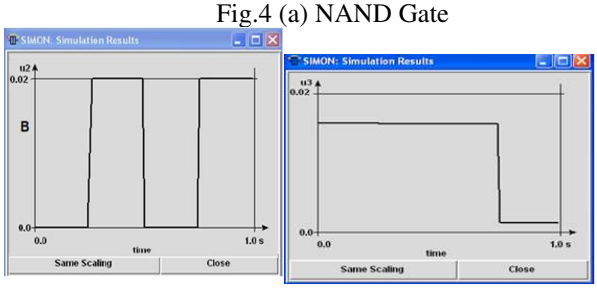

Fig.4 (b) Simulation result of NAND gate

8. 3-INPUT AND GATE

For implementing the 3-input AND gate we will use the parameters C_3=10.6C , $C_1^n = C_2^n$=0.5aF, $C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$, $C_0 = 8aF$, R_j= 10^5 Ω in Fig. 5(a) and accordingly after running the simulator the output we get is given in Fig. 5(b).

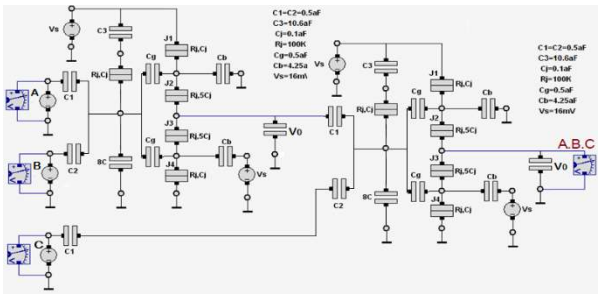

Fig. 5(a) 3-input AND gate

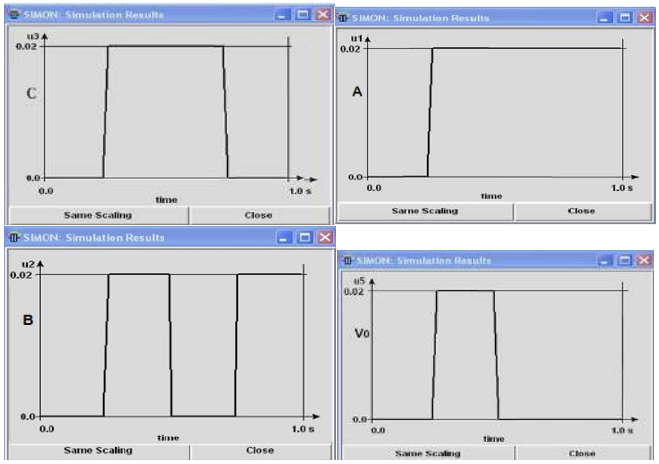

Fig.5(b) Simulation result of 3-input AND gate

9. FIG. 4-INPUT AND GATE

When we are implementing a 4-input AND gate we will use the parameters C_3=10.6C , $C_1^n = C_2^n$=0.5aF, $C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$, $C_0 = 8aF$, R_j=10^5 Ω in Fig. 6(a) and accordingly after running the simulator the output we get is given in Fig. 6(b).

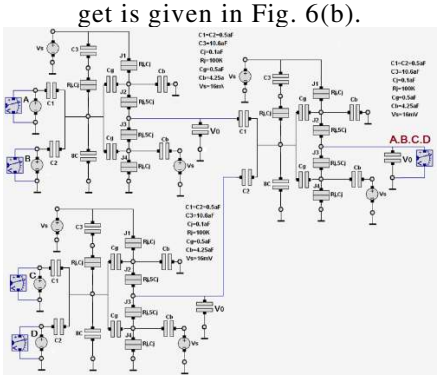

FIG. 6(A) 4-INPUT AND GATE

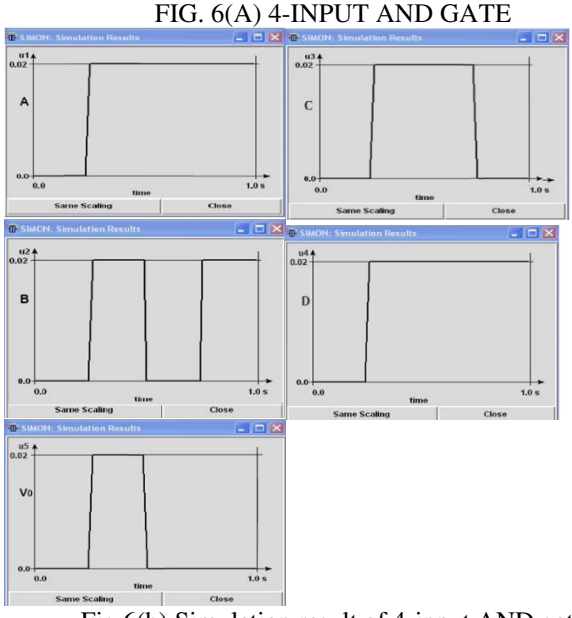

Fig.6(b) Simulation result of 4-input AND gate

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

10. OR GATE

For implementing the AND gate we will use the parameters $C_1^n = C_2^n$=0.5aF,$C_3 = 11.7aF$, $C_b = C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$,$C_0 = 8aF$, R_j= 10^5 Ω,$V_s = 16mV$ and accordingly after running the simulator the output we get is given in Fig. 7(b).

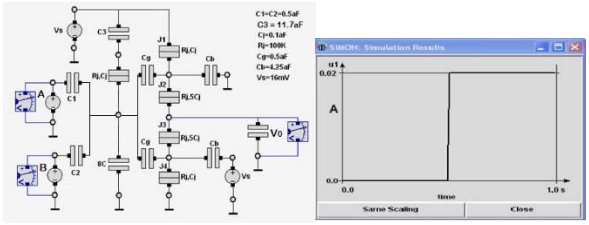

Fig. 7(a) OR gate

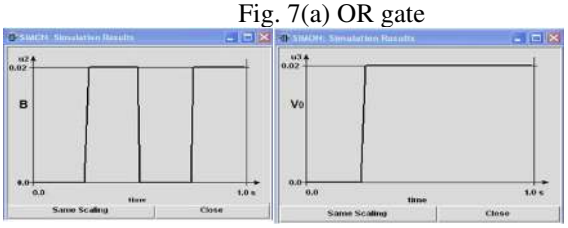

Fig. 7(b) simulation result of OR gate

11. 3-INPUT OR GATE

For implementing the 3-input AND gate we will use the parameters $C_1^n = C_2^n$=0.5aF,$C_3 = 11.7aF$, $C_b = C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$,$C_0 = 8aF$, R_j= 10^5 Ω,$V_s = 16mV$ and accordingly after running the simulator the output we get is given in Fig. 8(b).

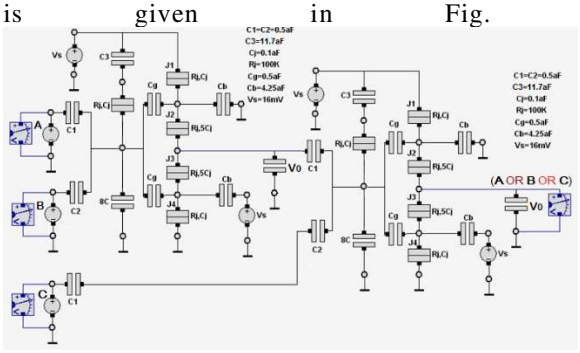

Fig. 8 3-input OR gate

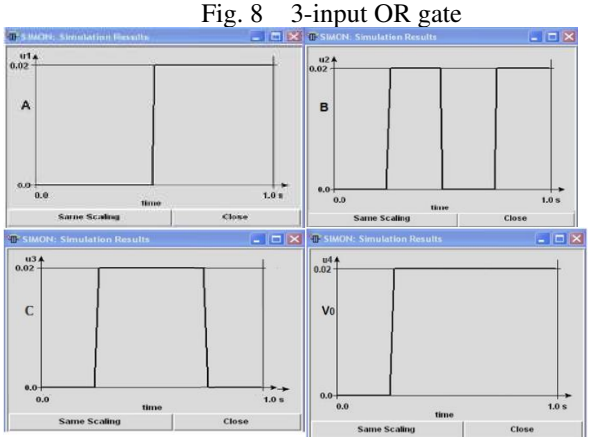

Fig.8 (b) Simulation result of 3-input OR gate

12. 4-INPUT OR GATE

Combining the three 2-input OR gate, a 4-input OR gate is built To implementing this gate we will use the parameters $C_1^n = C_2^n$=0.5aF,$C_3 = 11.7aF$, $C_b = C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$,$C_0 = 8aF$, R_j= 10^5 Ω,$V_s = 16mV$ and accordingly after running the simulator the output we get is given in Fig. 9(b).

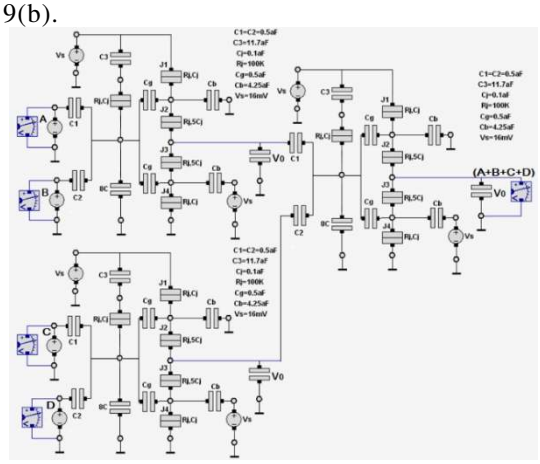

Fig. 9(a) 4-input OR gate

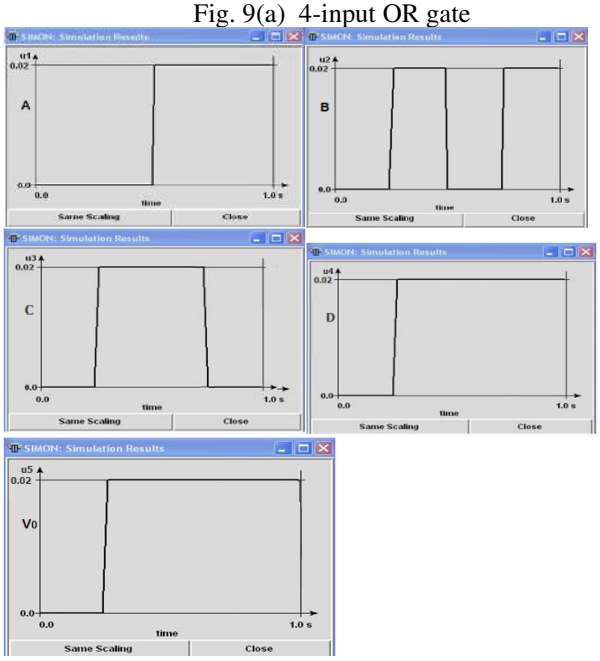

Fig.9 (b) Simulation result of 4-input OR gate

13. NOR GATE

For implementing the NOR gate we will use the parameters $C_1^n = C_2^n$=0.5aF,$C_3 = 11.7aF$, $C_b = C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF$,$C_0 = 9aF$, R_j= 10^5 Ω,$V_s = 16mV$ and accordingly after running the simulator the output we get is given in Fig. 10(b).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

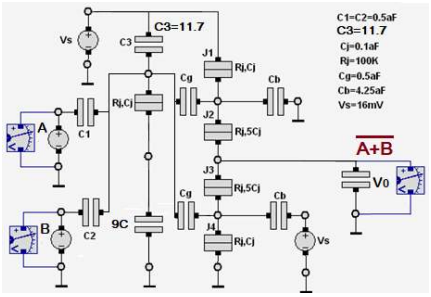

Fig. 10(a) NOR gate

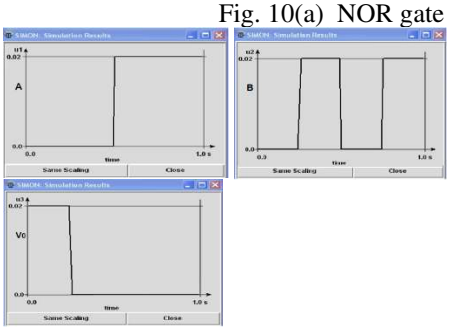

Fig.10 (b) Simulation result of NOR gate

14. EXCLUSIVE OR (XOR) GATE

For implementing the XOR gate we will use the parameters $C_1^n = C_2^n = C_1^P = C_2^P = C_g =$ 0.5aF, $C_3 = 11.7aF$, $C_{b1} = C_{b2} = C_b = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$, $C_L = 9aF, C_0 = 8aF, C_j =$ 0.1aF, $R_j =$ 10^5 $\Omega, V_s = 16mV$ and accordingly after running the simulator the output we get is given in Fig. 11(b).

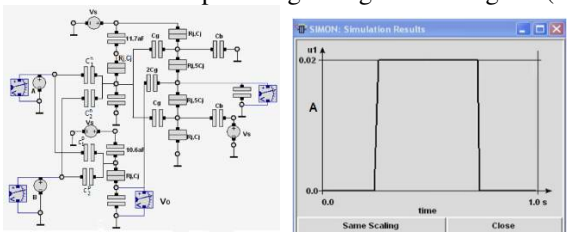

Fig. 11(a) XOR gate

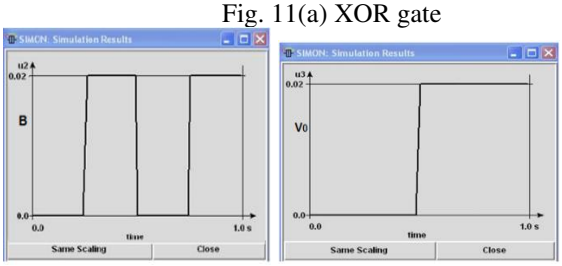

Fig. 11(b) simulation result of XOR gate

15. DESIGN OF A 4-BIT COMPARATOR

Here we consider two binary numbers A and B of bit lengths 4 which are given as follows:

$$\left.\begin{aligned} A &= A_3A_2A_1A_0 \\ B &= B_3B_2B_1B_0 \end{aligned}\right\} \quad \text{.....(22)}$$

where each suffixed letter represents one of the digits in the number. Both the two numbers are equal if all pairs of corresponding bits are equal i.e., if A_3=B_3, A_2=B_2 and A_1=B_1 and A_0=B_0 . As the numbers are binary, the bits are either 1 or 0 and the equality relation of each pair of bits can be expressed logically by an equivalence function given in equation (23):

$$x_i = A_iB_i + A'_iB'_i \quad i = 0, 1, 2, 3 \quad \text{.....(23)}$$

where x_i=1 only if the pair of bits in position i are equal, i.e., if both are 1's or both are 0's.

The equality of the two numbers, A and B, is displayed in a combinational circuit by an output binary variable which we designate by the symbol (A = B) shown in Fig. 12(a)[*given in a separate page extra page due to its large size*]. This binary variable is equal to 1 if the input numbers, A and B, are equal, and it is equal to 0 if they are not equal. If the equality condition exist, then all x_i variables must be equal to 1. This indicates an AND operation [20] of all variables:

$$(A=B) = x_3\, x_2\, x_1\, x_0 \quad \text{.....(24)}$$

the binary variable (A = B) is equal to 1 if and only if all pairs of digits of the two numbers are equal.

When we want to examine whether A is greater than or less than B, we investigate the relative magnitudes of the pairs of corresponding bits commencing from the most significant bit position. If the two bits are equal, we compare the next lower significant pair of bits. This comparison continues until a pair of unequal bits is faced. If the corresponding bit of A is 1 and that of B is 0, we conclude that A >B. If the corresponding bit of A is 0 and that of B is 1, we decide that A<B. The sequential comparison can be expressed logically by the following two Boolean functions (25) and (26) as follows.

$$(A > B) = A_3B'_3 + x_3A_2B'_2 + x_3\, x_2A_1B'_1 + x_3\, x_2\, x_1A_0B'_0 \quad \text{.....(25)}$$

$$(A < B) = A'_3B_3 + x_3A'_2B_2 + x_3\, x_2A'_1B_1 + x_3\, x_2\, x_1A'_0B_0 \quad \text{.....(26)}$$

The symbols (A > B) and (A < B) are binary output variables which are equal to 1 when (A > B) and (A < B) respectively. The circuit drawn in the blue box in Fig. 12(b) with the assistance of the equations (24), (25) and (26).

[Note: Fig. 12(a) A 4-bit Comparator based on LTG representing IC type 74L85 is given in **the last separate page due to its large size.]**

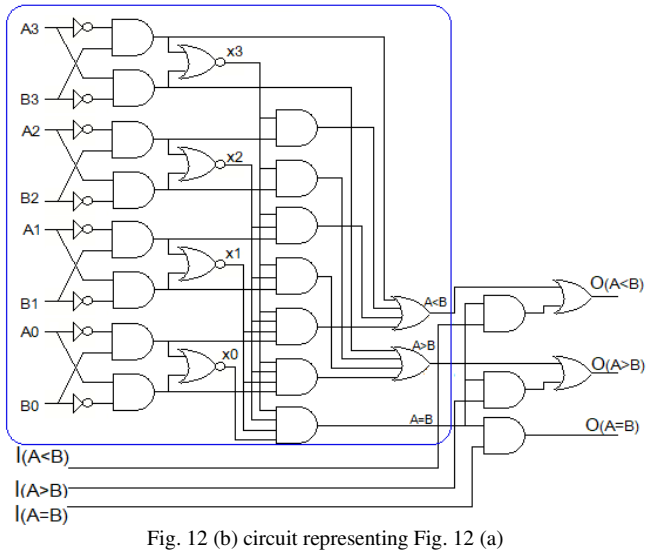

Fig. 12 (b) circuit representing Fig. 12 (a)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

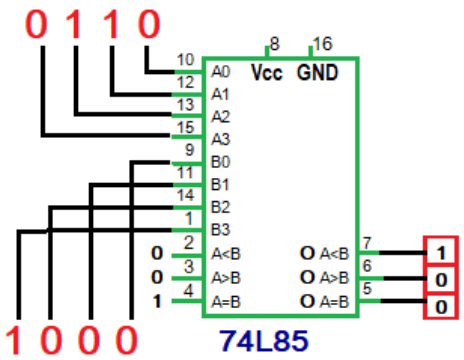

Fig. 12 (c) chip form of Fig. 13(a) (called 74L85)

16. DESCRIPTION OF 74L85 COMPARATOR

We have elaborately tried to implement the circuit of a 4-bit comparator of IC type 74L85 and which is depicted in Fig. 12(a). The Fig. 12(a) is pictorially represented by Fig. 12 (b). The IC type 74L85 is functionally implemented with the help of this Fig. 12 (b).

When we are interested in making the chip executable, first work is to provide power supply. As we know, the circuits like adder, multiplier start executing from LSB, on the contrary a comparator comparing more than 2-bits data commence comparing from the MSB and goes towards LSB. We take two data A and B having the same word length. The data values of A and B may be different or the same, whatever may be, the comparator will provide the decision like a signum function. For the case of an IC 74L85, the data inputs will be of two parts with 4-bits each and the output will be of 3-bits. In the input side data can be any binary numbers of four bits whereas in the output will be of only one bit *high* and other two bits will be of *low*. All the three bits are expressing the output depending upon the inputs.

17. 4-bit Comparator

A Comparator is a combinational circuit which compares two binary numbers for comparing whether one binary number is equal to, less than or greater than the other number. We logically design a circuit which is having two input data A and B of 4 bits each, and three output terminals, one for (A > B) condition, one for(A = B) condition and one for (A < B) condition.

When we are considering four bit comparison, the only Fig. 13(a) is sufficient to manipulate the data in the input side to provide output. In this case the three cascading input terminals indicated by input(A<B), input(A>B) and input(A=B) are kept in don't care condition, the only exception is that when the comparing data all are equal, we set input(A<B)=L, input(A>B)=L and input(A=B)=H. The input-output relationship is shown in Table-1. The marks '×' in the Table-8 indicate "don't care".

Table-8

Inputs being compared				Cascading Inputs			Output of 4-bit comparator		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	×	×	×	×	×	×	H	L	L
A3<B3	×	×	×	×	×	×	L	H	L
A3=B3	A2>B2	×	×	×	×	×	H	L	L
A3=B3	A2<B2	×	×	×	×	×	L	H	L
A3=B3	A2=B2	A1>B1	×	×	×	×	H	L	L
A3=B3	A2=B2	A1<B1	×	×	×	×	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	×	×	×	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	×	×	×	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H

Before Cascading: When we are interested in what happens if we take the all possible inputs for cascading inputs i.e., for input(A<B), input(A>B) and input(A=B).The input-output combinations we get from Fig. 12(a) /Fig.12(c) is written in Table-9. From the Table-9, we observe that when the comparing input data A and B are equal, the outputs (in the red colour box) are not acceptable, because at the same time two or three output levels can't be HIGH or three output levels can't be LOW for a comparator. Only one output level will be HIGH and other two will be LOW for a comparator. As the comparator output condition levels are violating for the cases indicating by the red color box in Table-9, so all the cascading inputs in the red color box will not be taken in consideration. So the modified table to be accepted when two or more 4-bit comparators are in cascaded connection is given in Table-10.

Table-9

Inputs being compared				Cascading Inputs			Output of 4-bit comparator		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	×	×	×	×	×	×	H	L	L
A3<B3	×	×	×	×	×	×	L	H	L
A3=B3	A2>B2	×	×	×	×	×	H	L	L
A3=B3	A2<B2	×	×	×	×	×	L	H	L
A3=B3	A2=B2	A1>B1	×	×	×	×	H	L	L
A3=B3	A2=B2	A1<B1	×	×	×	×	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	×	×	×	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	×	×	×	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	L	H	H	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	H	H	H	H
A3=B3	A2=B2	A1=B1	A0=B0	L	H	H	L	H	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	L	H	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	L	L	L	L

×= don't care, H= high level, L=Low level

We are considering the elements in the box as don't care for simplification of the table. For the expected working circuit two or three cascading inputs can't be H at the same time. Three of the cascading inputs can't be L at the same time also.

Table-10

Inputs being compared				Cascading Inputs			Output of 4-bit comparator		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	×	×	×	×	×	×	H	L	L
A3<B3	×	×	×	×	×	×	L	H	L
A3=B3	A2>B2	×	×	×	×	×	H	L	L
A3=B3	A2<B2	×	×	×	×	×	L	H	L
A3=B3	A2=B2	A1>B1	×	×	×	×	H	L	L
A3=B3	A2=B2	A1<B1	×	×	×	×	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	×	×	×	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	×	×	×	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H

×= don't care, H= high level, L=Low level

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

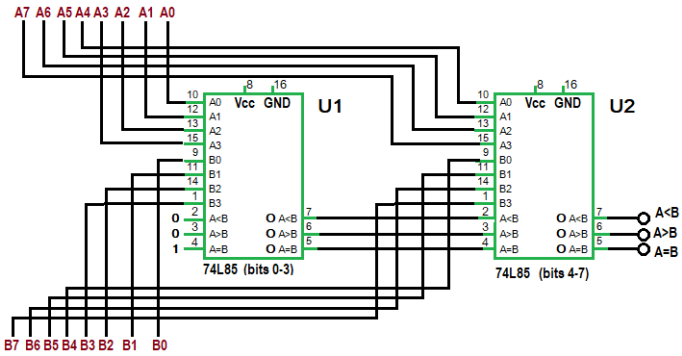

Fig. 13 8-bit comparator

18. 8-BIT COMPARATOR

After Cascaded connection: Two 8-bit numbers compared by cascading two 4-bit comparators U1 and U2 is shown in Fig.13. The output of the lower order comparator (U1) [i.e, when inputs A = (A3, A2, A1, A0) and B=(B3, B2, B1, B0)] at terminals 5(O A=B), 6(O A>B), and 7(O A<B) are connected respectively to the corresponding cascading inputs of the higher-order comparator (U2). In the lower order comparator (U1), the A=B cascading input (terminal 4) must be connected to HIGH level input. And the other two cascading inputs A>B (terminal 3) and A< B (terminal 2) must be connected to LOW. The outputs of the higher-order comparator (U2) will be the outputs of the 8-bit comparator.

If anybody is interested in making a circuit for comparing more than 8-bits then he/she can connect the circuit like Fig.14 for comparing 4×n bits, where n=3,4,5… . So he /she can use a 4n-bit comparators.

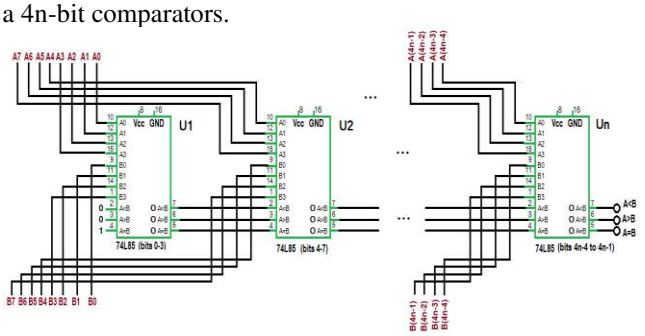

Fig. 14 a 4n-bit Comparator

19. DISCUSSION

Now we are interested in determining the processing delay of any two-/three-/four-input logic gates, combinational and sequential circuits based on the LTG gates. To find out the delay of a logic gate we have to take the consideration of involving critical voltage V_c given in equations (6) and (7), as well as the tunnel junction capacitance C_j. However, assuming at $T = 0K$, the switching/processing delay of a logic gate can be calculated by using the approach [1,7, 8, 9].

$$\text{delay} = -(e|\ln(P_{error})|R_t) / (|V_j| - V_c) \quad \cdots\cdots\cdots\cdots (21)$$

where V_j is the junction voltage and V_c is the critical (threshold) voltage

The slowest switching happens whenever the tunnel junction voltage V_j has the value greater than the critical voltage V_c, i.e., $|V_j| > V_c$, but very close to it. This must happen, for instance, for the case of a 2-input NOR gate, when only V_{in1} is logic 1, resulting V_j =11.8mV, the critical voltage of the tunnel junction voltage V_c is 11.58mV. We consider that the probability of error change P_{error} =10^{-12} and the junction resistance $R_t = 10^5\Omega$. We obtain a gate delay equal to 0.07281|ln (P_{error})|ns = 1.675 ns. In the same manner, we are capable of calculating the circuit delays and they are tabulated in Table-11.

Table-11

Gate	Area	Delay	Switching Energy
inverter	09 elements	0.022|ln(P_{error}) | ns	10.4 meV
2-input NOR	14 elements	0.072|ln(P_{error})| ns	10.7 meV
2-input OR	14 elements	0.062|ln(P_{error})| ns	10.8 meV
2-input NAND	14 elements	0.080|ln(P_{error}) |ns	10.7 meV
2-input AND	14 elements	0.062|ln(P_{error}) |ns	10.8 meV
2-input XOR	20 elements	0.102|ln(P_{error})| ns	21.2 meV
D latch	65 elements	0.342|ln(P_{error})| ns	53.2meV
3-input OR	28 elements	0.104|ln(P_{error}) | ns	21.6 meV
4-input OR	42 elements	0.104|ln(P_{error}) | ns	32.4 meV
3-input AND	28 elements	0.104|ln(P_{error}) | ns	21.6 meV
4-input AND	42 elements	0.104|ln(P_{error}) | ns	32.4 meV
4-bit Comparator	604 elements	0.528|ln(P_{error}) | ns	491.9 meV
8-bit Comparator	1208 elements	1.056|ln(P_{error}) | ns	983.8 meV

When an electron passes through the tunnel junction, the amount of total energy in the circuit changes after the tunneling. So the difference between the energies before and after the tunneling event is calculated by the equation (22)

$$\Delta E = E_{before\ tunnel} - E_{after\ tunnel} = -e(V_c - |V_j|) \quad \cdots\cdots\cdots\cdots (22)$$

and this ΔE is the amount of switching energy consumed while a tunneling event occurs in the tunneling circuit.

Here we have drawn the curves of the switching delay as a function of the switching error probability in Fig. 15(a) and the switching delay as a function of the unit capacitance C which is drawn in Fig. 15(b).

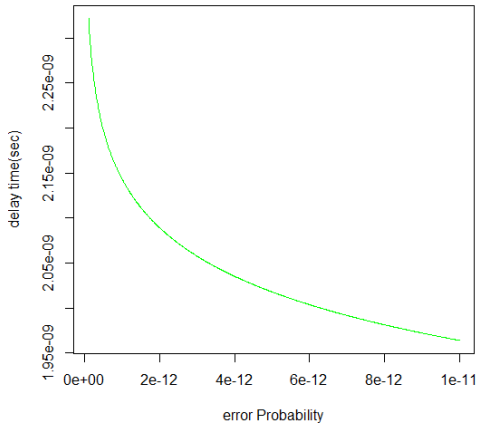

Fig. 15(a) Delay vs. Error Probability

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

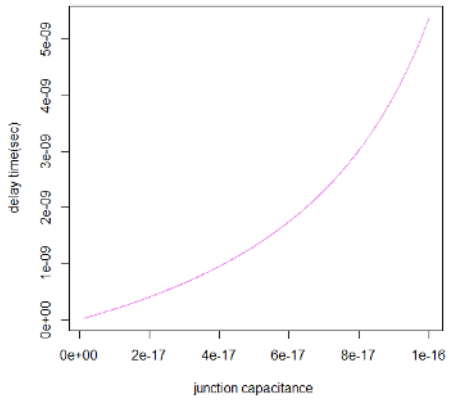

Fig. 15 (b) Delay Vs. capacitance

We have also found out the area/element numbers for all gates, their switching delay, and switching energy consumptions for the corresponding individual linear threshold gates (using the same methodology as accepted for the Boolean gates). The switching energy vs. elements diagram regarding our present LTG based circuits is shown in Fig. 16.

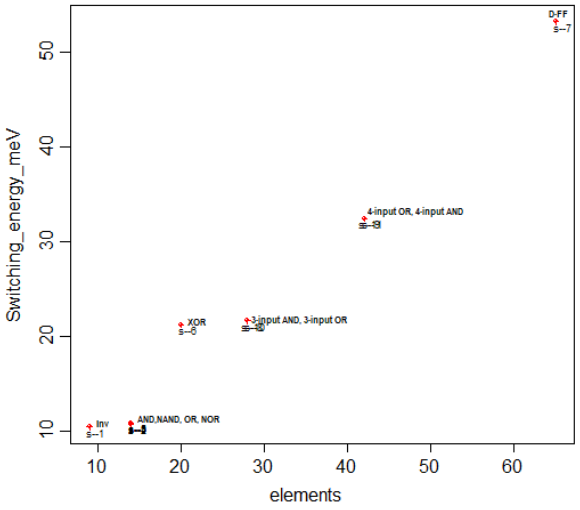

Fig. 16. Switching energy vs. elements

From the Table-11 we observe that the time delay for the 4-bit comparator as well as IC Type 74L85 is 0.528|ln(P_{error}) | ns , for 8-bit comparator time costs for processing is 1.056|ln(P_{error}) | ns and in general for a 4n-bit comparator the processing delay will be 0.528n|ln(P_{error}) | ns. In this case time delay is directly proportional to number of 4-bit comparators or IC Type 74L85.

Given that the value of P_{error} equals to 10^{-16}, so the time delay for a 8-bit comparator to provide the first decision about the two 8-bit input data being compared is 19.45ns. Therefore the fan-out rate is $=\frac{1}{19.45\text{ns}} \approx 51.413881\text{GHz}$.

Table-12

Gate or circuit	SET logic circuits	TLG based delays When $P_{error}=10^{-16}$
inverter	8	0.81ns
2-input NOR	4	2.65ns
2-input OR	4	2.28 ns
2-input NAND	4	2.94ns
2-input AND	4	2.28 ns
2-input XOR	4	3.76 ns
D latch	32	12.6 ns
3-input OR	8	3.83ns
4-input OR	8	3.83ns
3-input AND	8	3.83ns
4-input AND	8	3.83ns
4-bit Comparator	40	19.45ns
8-bit Comparator	80	39.90ns

20. SWITCHING DELAYS OF LTG AND SET

The time delay or the processing delay for a CMOS logic gate like NAND, NOR, XOR is 12ns [20], whereas the time required for tunneling through a single electron transistor (SET) is approximately close to 4ns [4, 5, 16, 18]. The XOR gate using conventional logic circuits needs 16 transistors, whereas the same function can be implemented with the help of just one SET [3, 5, 6, 11] i.e. number of nodes can be reduced to 1 instead of 16.

Assuming that that error probability is 10^{-16} then the delay for the 2-input OR gate will be 2.28ns and similarly the other delays for the other gates used in implementing the IC chip 74L85 can be calculated and are all shown in Table-12. It is transparent that the LTG based circuit is faster than the SET based circuit when $P_{error}=10^{-16}$. The comparison of delays for SET and LTG gate based circuits is drawn by a bar diagram in Fig.16.

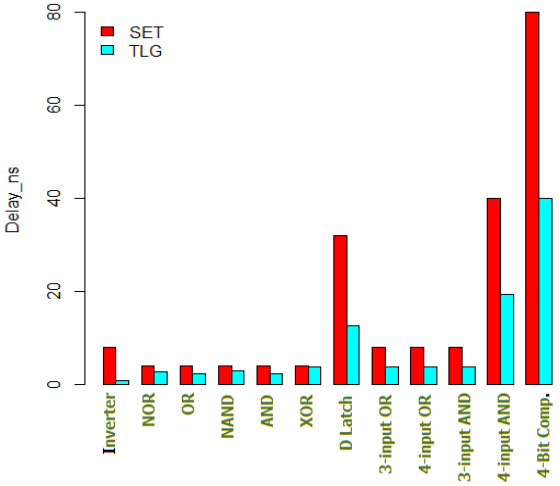

Fig.16 Delay comparison of SET and LTG

21. CONCLUSION

We discussed a novel design of threshold logic gates implemented in single electron transistor technology, where the basic operating principle is single electron transport phenomenon. A generic Linear Threshold Logic Gate implementation is elaborately discussed and from which we

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

have obtained a family of logic gates like AND, NAND, OR. We have also derived the threshold logic equations for different logic gates. All the 2-input logic gates along with 3-input AND gate and 4-input AND/OR gates have been implemented and are verified by means of simulation using SIMON. The number of elements like true capacitors, junction capacitor for logic gates, their delays, power consumed by them are shown in a tabular form and their related curves or bar diagram are produced in due places. With the help of the different logic gates we have implemented a 4-bit comparator and by this single comparator unit we have been able to build 8-bit or 4n-bit comparator. In single electron tunneling technology, we observe that the threshold logic gates are at least 3 times faster than CMOS based logic gates. The operating temperature is kept very close to 0K in real operation. At last we find that the implemented comparator providing the fan-out is the order of $O(10^9)$.

22. REFERENCES

[1] Anup Kumar Biswas,"State Transition Diagram for A Pipeline Unit based on Single Electron Tunneling".International Journal of Engineering Research & Technology (IJERT) Vol. 10 Issue 04, April-2021

[2] Anup Kumar Biswas, "Design of A Pipeline for A Fixed-Point Multiplication using Single Electron Tunneling Technology", International Journal of Engineering Research & Technology (IJERT), Vol. 10 Issue 04, April-2021 pp. 86-98

[3] Souvik Sarkar1, Anup Kumar Biswas2, Ankush Ghosh1, Subir Kumar Sarkar1 "Single electron based binary multipliers with overflow detection", International Journal of Engineering, Science and Technology Vol. 1, No. 1, 2009, pp. 61-73

[4] A. K. Biswas and S. K. Sarkar: "An arithmetic logic unit of a computer based on single electron transport system": Semiconductor Physics, Quantum Electronics & Opt-Electronics. 2003. Vol 6. No.1, pp 91-96

[5] A.K. Biswas and S. K. Sarkar: "Error Detection and Debugging on Information in Communication System Using Single Electron Circuit Based Binary Decision Diagram." Semiconductor Physics Quantum electronics and opt electronics, Vol. 6, pp.1-8, 2003

[6] Alexander N. Korotkov, "Single-electron logic and memory devices" INT. ELECTRONICS, 1999, Vol. 86, No. 5, 511- 547

[7] Casper Lageweg, Student Member, IEEE, Sorin Cot¸ofan˘a, Senior Member, IEEE, and Stamatis Vassiliadis, Fellow, IEEE "Single Electron Encoded Latches and Flip-Flops" IEEE TRANSACTIONS ON NANOTECHNOLOGY, VOL. 3, NO. 2, JUNE 2004

[8] C. Lageweg, S. Cot¸fan˘a, and S. Vassiliadis, "A linear threshold gate implementation in single electron technology," in IEEE Computer Society VLSI Workshop, Apr. 2001, pp. 93– A. Korotkov, "Single-electron logic and memory devices," Int. J. Electron., vol. 86, no. 5, pp. 511–547, 1999.

[9] K. Likharev, "Single-electron devices and their applications," Proc. IEEE, vol. 87, pp. 606–632, Apr. 1999.

[10] A. Korotkov, R. Chen, and K. Likharev, "Possible performance of capacitively coupled single-electron transistors in digital circuits," J. Appl. Phys., vol. 78, pp. 2520–2530, Aug. 1995.

[11] J. R. Tucker, "Complementary digital logic based on the "Coulomb blockade"," J. Appl. Phys., vol. 72, no. 9, pp. 4399–4413, Nov. 1992.

[12] A. Korotkov and K. Likharev, "Single-electron-parametron-based logic devices," J. Appl. Phys., vol. 84, no. 11, pp. 6114–6126, Dec. 1998.

[13] C. Wasshuber, H. Kosina, and S. Selberherr, "SIMON—A simulator for single-electron tunnel devices and circuits," IEEE Trans. Computer- Aided Design, vol. 16, pp. 937–944, Sept. 1997.

[14] A. B. Zorin, S. V. Lotkhov, H. Zangerle, and J. Niemeyer, "Coulomb blockade and cotunneling in single electron circuits with on-chip resistors: Toward the implementation of the R pump," J. Appl. Phys., vol. 88, no. 5, pp. 2665–2670, Sept. 2000.

[15] T. Oya, T. Asai, T. Asai, T. Fukui, and Y. Amemiya, "A majority-logic device using and irreversible single-electron box," IEEE Trans. Nanotechnol.,vol. 2, pp. 15–22, Mar. 2003.

[16] Z. Durrani, A. Irvine, and H. Ahmed, "Coulomb blockade memory using integrated single-electron transistor/metal–oxide–semiconductor transistor gain cells," IEEE Trans. Electron Devices, vol. 47, pp. 2334–2339, Dec. 2000.

[17] J. Millman and C. C. Halkias; Integrated Electronics- Analog and Digital Circuits and Systems_ McGraw Hill Education; 2 edition

[18] Kai Hwang and Naresh Jetwani, " Advanced Computer Architecture parallelism, scalability, programability" chapter 6, 2nd edition McGraw Hill

[19] Millman's Electronic Devices & Ciruits 4th Edition (English, Paperback, Millman Jacob)

[20] H. MORRIES MANO, Digital Logic and Computer Design. Prentice Hall of India Chapter-5, 17th Indian Edition, 1999

BIOGRAPHY

Anup Kumar Biswas is an Assistant Professor in the Department of Commuter Science and Engineering in Kalyani Govt. Engineering College. He is awarded his PhD[Engg.] degree in the stream of Electronics and Telecommunication Engineering from Jadavpur University in the year 2006. He has engaged in teaching and research activities since the last 16 years. His Specialization field is nanotechnology especially single electron tunneling technology. Dr. Biswas has published several papers in various national, international conferences and journals.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

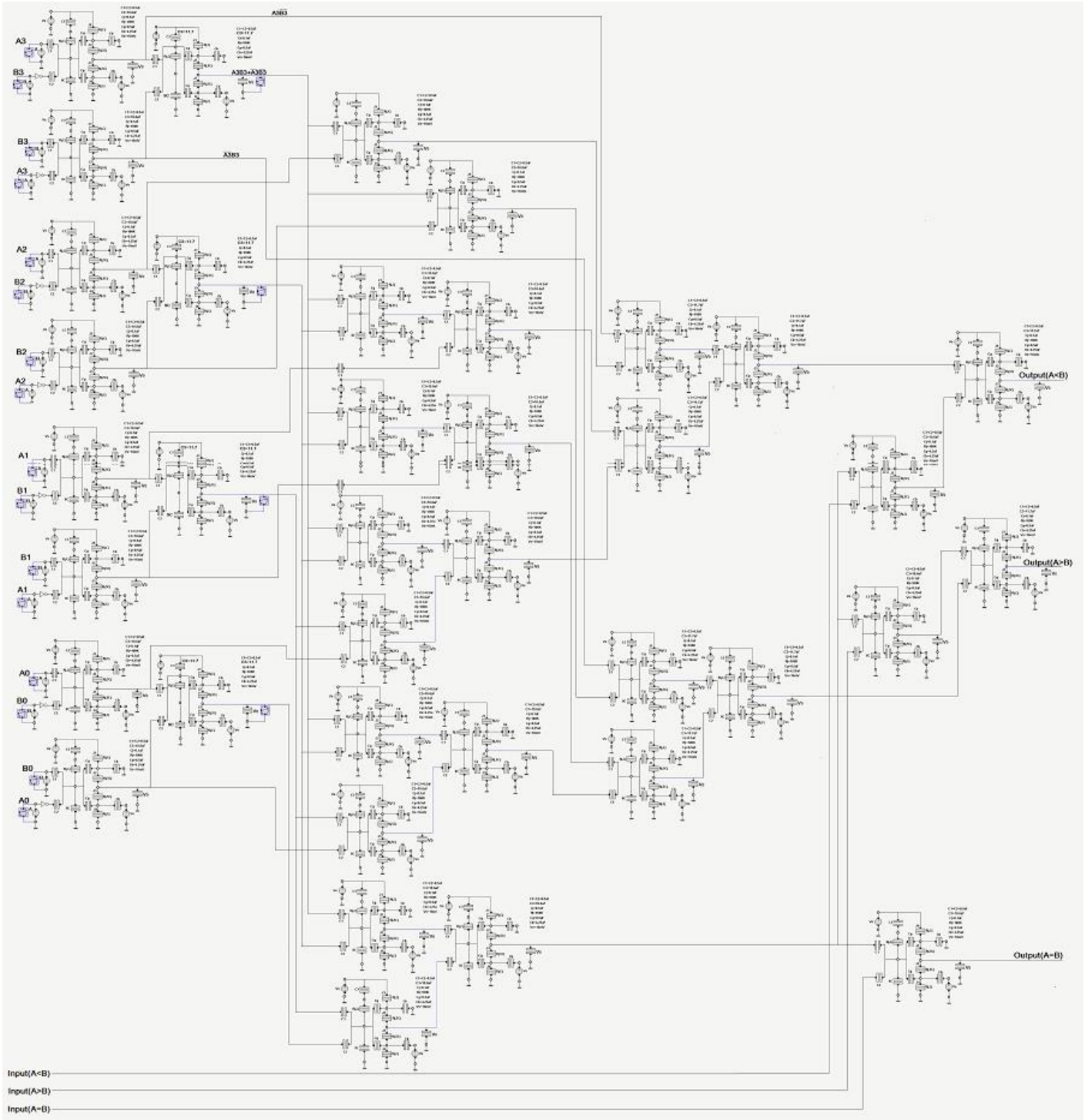

Fig. 12(a) A 4-bit Comparator based on LTG representing IC type 74L85