IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Sequential circuit implementation using electron tunneling and TLG Technology

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

ABSTRACT

The two types of devices, Single Electron tunneling devices (SEDs) and Threshold Logic Gates (TLGs), both have the power of controlling the transport of an electron through a tunnel junction at a certain time. A single electron has the charge which is adequate to store an information in a SED. The switching delay of a Threshold Logic Gate is very small and speed of the processing of TLG based devices will be in the order of 10^9. For implementing logic gates and a sequential circuit, TLG, of course, will be a best candidate to fulfill the necessities. When an Ultra-low noise is considered, TLG based circuit will be the best selection for implementing our desired tunneling circuits. Different TLGs like 2-input AND/NAND, 2-input XOR/XNOR, RS Flip-flop, T-Flip-flop have been implemented with the help of concept of linearly separable threshold logic gate of multiple inputs. Almost in every instances, the threshold logic equations, Truth tables and simulated results for them are provided in parallel in due places.

Key words: Electron-tunneling, XOR, T Flip-flop, Coulomb-blockade, sequential circuit

1. INTRODUCTION

Single Electron tunneling based device is one of such an equipment by which we can draw all Boolean logic gates and more complex circuits to be implemented. A single electron can passes through the tunnel junction only when the junction voltage, under the action of bias voltage and multiple input voltages connected to the islands via small capacitors, exceeds the threshold voltage. For implementing a sequential circuit, TLG based logic gates will be a suitable candidate. And with the help of the principle of threshold linear equations, someone will be able to build more complex logic circuits easily.

2. BUFFER

The buffer or inverter [2, 3, 4, 8, 10] is a circuit, that complement its own input is depicted in Fig. 1(a). It is made up of two single electron transistors (SET1 and SET2) connected in series. The two input voltages having the same values are directly coupled to the islands of the SET1 and SET2 [6, 7, 9] through two true capacitors C_1 and C_2 bearing the same values of capacitances. The island-sizes of the both SETs have the dimension of 10 nm diameter of gold and their capacitances should be close to 10^{-17} F. The output terminal V_0 is connected to a point in between SET1 and SET2 and to the ground through a capacitance C_L for the purpose of putting down charging effects.

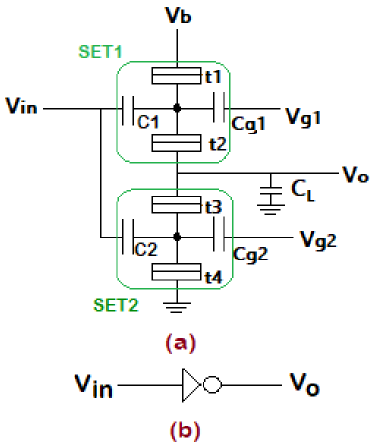


Fig. 1 (a) Buffer (b) Symbol of Buffer

To build the buffer, the parameter values we have 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=50KΩ. For simulation purpose, the value of the unit capacitance C will be taken as 1aF.

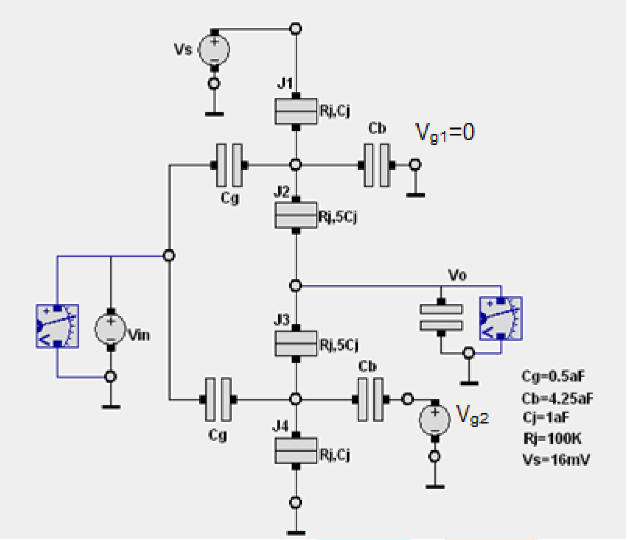
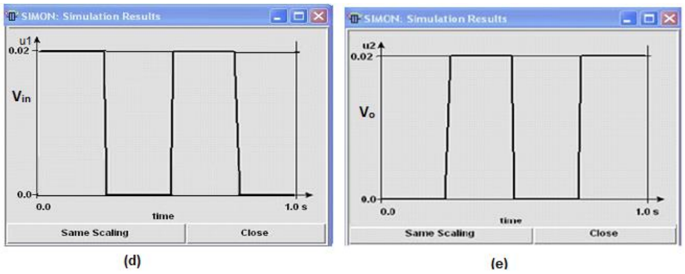


Fig. 1(c) Simulation set of Buffer



(d)

(e)

Fig. 1(d), (e) Simulation result of Buffer input and output

The operation of the buffer is being described as: - the output voltage V_0 will be high until the input voltage V_{in} becomes high and the V_0 value will be low as soon as the input voltage is high. To have this target, we must set the voltages $for\ V_{g1} = 0\ V\ and\ V_{g2}$ =16mV along with the input voltages, at present, V_{in} both for SET1 and SET2. SET1 is in conduction mode only when V_{in} set low and the SET2 is in Coulomb blockade [2, 3, 4, 12]. This arrangement connects the output voltage V_0 to V_b and therefore the output voltage is high. As the high voltage=16mV (logic 1) is applied to the input terminal(s), it causes the induced charge on each of the islands of the two SETs to transfer by a fraction of an electron charge and enforces the SET1 in Coulomb blockade and the SET2 in conducting mode. Therefore, the output shifts from high to low (logic 0).

In this work, we have assumed for the Boolean logic inputs "0" =0 Volts and logic "1"=0.1×$\frac{q_e}{C}$. For different simulation cases, we will consider that C=1aF and Logic "1"= 0.1×$\frac{1.602\times10^{-19}}{1\times10^{-18}}$=$0.1 \times 1.602 \times 10^{-2}$=$16.02 \times 10^{-3}$ =16.02 ≅ 16 mV.

3. Multiple input threshold logic gate

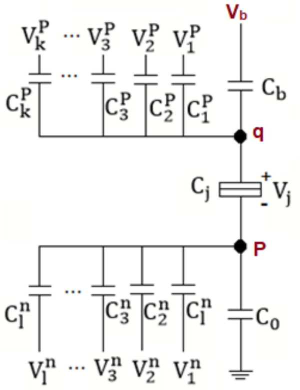


Fig. 2 Multiple input TLG

The figure given in Fig.3 is n-input multiple threshold logic gate [1, 2, 3, 4, 9, 10] consisting of a tunnel junction. The tunnel junction has its capacitance C_j and resistance R_j. Each input voltage V_k^P above the tunnel junction is connected to the point "q" through their corresponding capacitors $C_k^P s$; and each input voltage V_l^n below the tunnel junction is connected to the point "p" through their respective capacitors C_l^ns. The Bias voltage V_b is connected to point the "q" through a true capacitor C_b too. Junction capacitor C_j is connected at point "p" that is connected to the ground through a true capacitor C_0 which takes an active role for determining the input voltage. LTGs can be implemented by means of a function which is represented by the signun function of h(x) expressed by equations (1) and (2).

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

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

where x_k and w_k being the n-Boolean inputs and their corresponding n integer weights respectively and θ being the threshold.

The LTG compares two parts (i) the weighted sum of the inputs $\sum_{k=1}^{n}(w_k \times x_k)$ and (ii) the threshold value θ. If weighted sum-value in the input side is greater than or equal to the critical voltage or threshold value θ , the logic output of the LTG will be high else output will be low (logical "0").

Two basic circuit elements in a LTG are being the tunnel junction capacitance C_j and the capacitance C_0 . The input signal voltages $V_1^P, V_2^P, V_3^P, \dots, V_k^P$ weighted by their corresponding capacitances $C_1^P, C_2^P, C_3^P, \dots, C_k^P$, are added to the junction voltage, V_j. On the other hand, the input signal voltages $V_1^n, V_2^n, V_3^n, \dots, V_l^n$ weighted by their corresponding vector capacitances $C_1^n, C_2^n, C_3^n, \dots, C_l^n$, are subtracted from the voltage, V_j.

To enable tunneling action, the critical voltage V_c is essential, and this V_c acts as the intrinsic threshold of the tunnel junction circuit. The bias voltage V_b is being used to adjust the gate threshold to the desired value θ. A tunneling event initiates, only when an electron starts passing through the junction from p to q as directed by a green arrow in Fig. 2.
The following expressions we will use for the rest our discussion.

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

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

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

Whenever all the voltage sources in Fig. 3 are connected to ground, the circuit is considered that it is made up of only three capacitors namely, C_Σ^P, C_Σ^n and C_j connected in series. C_T *is* assigned to be the sum of all 2-term products of the three capacitances C_Σ^P, C_Σ^n and C_j.

4. 2-input AND gate

When we are trying to make the threshold logic gate of AND, we are to draw the Table-1 of an AND gate and compare the weights of variables w_A and w_B of two variables A and B respectively with the threshold θ [1, 2,3,8,13,14].We can write the threshold logic equation of a 2-input AND gate having two variables A and B as.

$$AND(A,B) = sgn\{w_A.A + w_B.B - (\theta)\} \cdots\cdots\cdots (6)$$

Table-1

A	B	F(A,B)	θ	Eqn. no.
0	0	0	0<θ	(1)
0	1	0	$w_B < \theta$	(2)
1	0	0	$w_A < \theta$	(3)
1	1	1	$w_B + w_A \geq \theta$	(4)

After solving the 4 equations in 4th column of Table 1, we have one set of solution w_B=1, w_A=1 and θ=2. So the Threshold logic equation for AND gate can be written in equation (6a) and its corresponding threshold logic gate is drawn in Fig. 3(a)

$$AND(A,B) = sgn\{A + B - 2\} \cdots\cdots\cdots (6a)$$

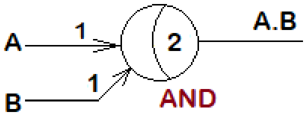


Fig. 3(a) Threshold logic AND gate

For correct operation of an AND gate we should involve an inverter. So, instead of taking an AND gate like in Fig. 3(a), we first draw an NAND gate and append an inverter in series for obtaining the modified AND gate. With the assistance of a truth table-2 of NAND gate we will be able to draw the threshold equation with respect to threshold θ and the coefficients w_A *and* w_B of A and B respectively.

$$NAND(A,B) = sgn\{w_A.A + w_B.B - (\theta)\} \cdots\cdots\cdots (7a)$$

Table-2

A		B	NAND(A,B)	θ	Eqn. no.
0		0	1	0≥ θ	(1)
0		1	1	$w_B \geq \theta$	(2)
1		0	1	$w_A \geq \theta$	(3)
1		1	0	$w_B + w_A < \theta$	(4)

After solving the 4 equations in 4th column of Table 2, we have got one set of solution $w_B = -1$, w_A=-1 and $\theta = -1.5$. Hence, the threshold logic equation for an NAND gate will be as in equation (7b) and its corresponding threshold logic gate is drawn in Fig. 3(b).

$NAND(A, B) = sgn\{-A - B - (-1.5)\}$………… (7b)

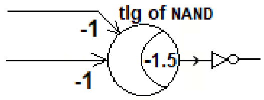
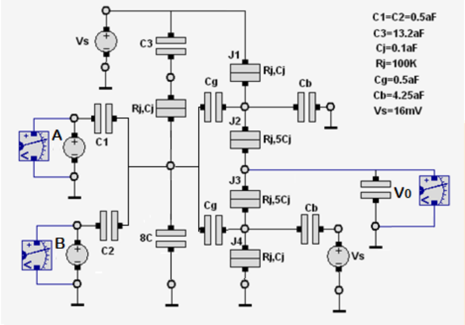


Fig. 3(b) AND gate (NAND plus buffer)

To implement the AND gate, we will use the parameters $C_1^n = C_2^n = 0.5\text{aF}$, $\boldsymbol{C_{b1}} = \boldsymbol{C_{b2}} = 4.25\text{aF}$, $C_{g1} = C_{g2} = 0.5aF, C_L = 9aF, C_0 = 8aF,\ R_j = 10^5\ \Omega$. The simulation set for the AND threshold logic is given in Fig. 3(c) and after simulating the result we get is given in Fig. 3(d).



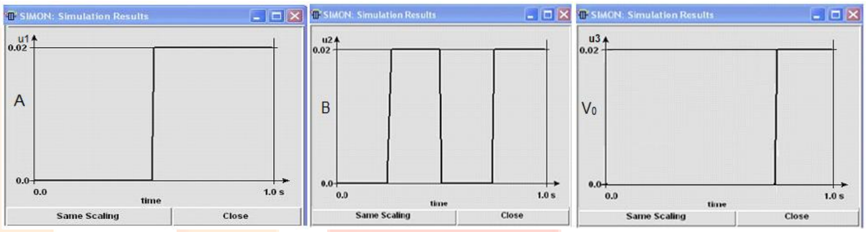


Fig.3 (c) AND Gate Fig. 3(d) Simulation result of AND gate

5. XOR Gate

The logic function of XOR gate, for two variables A and B, is defined as Y=A.B̄ +Ā.B. The space plot diagram of Y in 2D space is shown in Fig. 4. From this figure it is crystal transparent that no linear separating line can separate the red and colorless bubbles. Hence, the Boolean logic function Y=A.B̄ +Ā.B is not linearly separable. As a result, we cannot draw a threshold logic gate representing the equation Y=A.B̄ +Ā.B.

For representing the Boolean function Y=A.B̄ +Ā.B by a threshold logic gate, we assign P = (A.B̄), as it is a AND logic equation, we can draw a threshold gate-based equation as given in equation (8).

P = $sgn\,\{A + \bar{B} - 2\}$………………… (8)

As we know B + B̄ =1 or B̄= $-$ B + 1, the equation (8) can be written by the equation (9). The equivalent Threshold gate is provided in Fig. 4(a) or 4 (b). For the verification of the equation (9), the truth Table-3 is given.

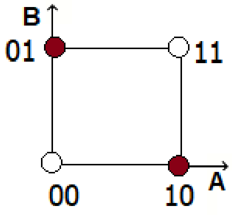
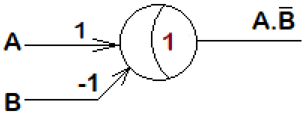


Fig. 4 space plot of Y

P(A,B̄) = $sgn\,\{\,A - B - (1)\,\}$…………………… (9)



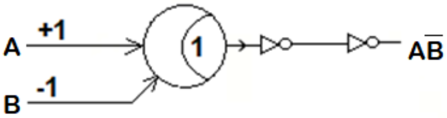


Fig. 4(a) Threshold gate of P = A.B̄ Fig. 4(b) Threshold gate of A.B̄ using double buffers

Table-3
Truth table of $P(A,\bar{B})$

A	B	P
0	0	0
0	1	0
1	0	1
1	1	0

We can modify the threshold equation of P as we want to apply a buffer, we know that the buffer inverts its input signal, so when a buffer is appended with Fig. 4(a), it calculates $\bar{P}=\bar{A}+B$. Its corresponding truth table is given in Table-4. From the table, one can easily derive the threshold logic equation as $\bar{P}$=sgn $\{-A+B-(-0.5)\ \}$. The corresponding threshold gate using a buffer is Fig. 4(d).

Table-4
Truth table of $\bar{P}$

A	B	$\bar{P}$	θ
0	0	1	$0 \geq \theta$
0	1	1	$w_b \geq \theta$
1	0	0	$w_a < \theta$
1	1	1	$w_a + w_b \geq \theta$

Whenever we are interested in drawing the threshold logic gate of $\bar{P}$ we require the logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^n = C_1^P = \frac{1}{2}C$ =0.5aF, $C_b = 10.24aF,\ C_j = 0.25aF,\ C_L = 9aF, C_0 = 9.5aF,\ R_j$=$10^5$ Ω,. V_b = 0.95 e/C = 15.2mV ($as\ e/C = 160mV$) which is close to 16mV, therefore, V_b =V_s=16mV.

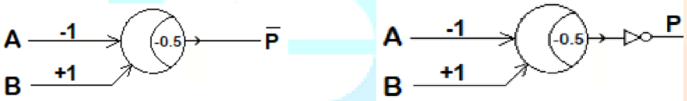


Fig. 4(c) Threshold logic gate of $\bar{P}$ 4(d) of P

Assuming, P= $\bar{A}$.B, we can express the Boolean expression Y=A.$\bar{B}$ +$\bar{A}$.B as Y= P + $\bar{A}$.B.

$$Y = P + \bar{A}.B \text{ .... } (10)$$

To get the threshold gate logic of equation (10), we draw the truth Table-5 with the assistance of Table-3 and from Table-5 we solve the equations to get the weight values.

Table-5

A	B	P	Y	θ	Eqn. no.
0	0	0	0	$0<\theta$	(1)
0	1	0	1	$W_2 \geq \theta$	(2)
1	0	1	1	$W_1 + W_3 \geq \theta$	(3)
1	1	0	0	$W_1 + W_2 < \theta$	(4)

After solving the conditional equations in Table-5, we have obtained a solution set $\{W_1, W_2, W_3\ ; \theta\}$ = {-1, 1, 2 ; 1}.

Hence the Threshold equation for the Y or for an XOR is
Y = sgn $\{-A + B + 2P - (1)\}$................ (11)

And its corresponding Threshold logic gate is depicted in Fig. 5. For correct operation, we are to apply a buffer in series to obtain an XNOR which is also shown in Fig. 5. If we again add another buffer in series we obtain an XOR gate shown in Fig. 5 as well.

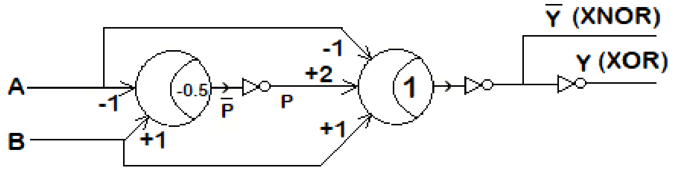


Fig. 5 XOR/ XNOR gate of two inputs

For making the threshold logic gate of an XOR, we have chosen, logic input “0”=0V, logic “1” = 16mV, C=1aF, $C_1^n = C_1^p = 0.5C_2^p = \frac{1}{2}C$ =0.5aF, $C_b = 8.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, $V_b = 15.81mV$ that is close to 16mV, so V_b =V_s=16mV. The simulation set and simulation result of XOR gate are given in Fig. 6 and 7 respectively.

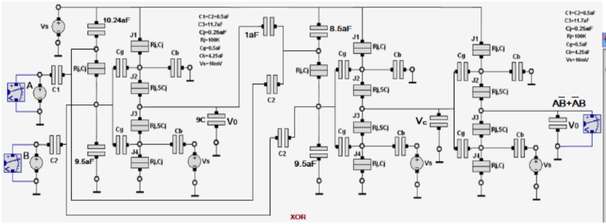


Fig. 6 Simulation set of XOR(A,B)

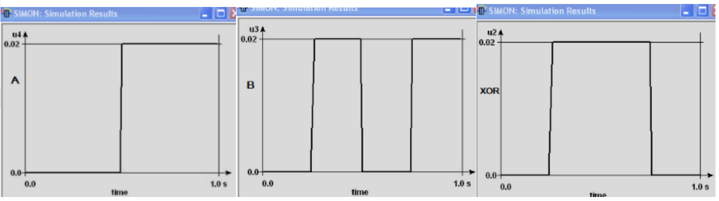

SIMON: Simulation Results
u4
0.02
A
0.0
0.0
1.0 s
time
SIMON: Simulation Results
u3
0.02
B
0.0
0.0
1.0 s
time
SIMON: Simulation Results
u2
0.02
XOR
0.0
0.0
1.0 s
time

Fig.7 Simulation result of XOR(A,B)

6. RS Flip-flop

RS Flip-flop based on Boolean logic is shown in Fig.8. The output of new Q^t is Q^{t+1}=S+$\bar{R}Q^t$. The Truth table of the flip-flop is given in Table-6.

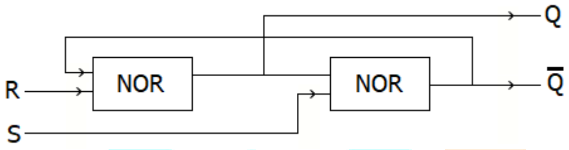


Fig. 8 RS Flip-flop

Table-6
Truth Table of R-S Flip-flop

R	S	Q	$\bar{Q}$	Description
0	0	last Q	last $\bar{Q}$	Keeps current o/p values of Q *and* $\bar{Q}$
0	1	1	0	o/p Q is set to 1 and $\bar{Q}$ to 0
1	0	0	1	Q is reset to 0 and $\bar{Q}$ to 1
1	1	?	?	Forbidden or Unspecified input combination

RS Flip-flop is a sequential circuit, as its outputs are being feedback to inputs. This flip-flop is made up of two NOR gates, two input terminals (A) and (B), and two outputs (Q) and ($\bar{Q}$) as depicted in Fig. 8. The input-output relationship shown in the Table-6 is as follows. *When* $R = S = 0$, the output of the Flip-flop remains the same. When the inputs $R = 1$ *and* $S = 0$, the output Q is reset to $Q = 0$ *and* $\bar{Q} = 1$. If inputs R=0 *and* S=1, the output Q is reset to Q =1 and $\bar{Q}$ = 0. On the other hand when the inputs are both 1, the output becomes unstable, this is why we are to avoid this input condition.

A Boolean gate-based implementation for the RS Flip-flop usually requires two cross-coupled gates which constitute a feedback loop. When R and S are “0”, the two *NOR gates* behave as chained inverters and they constitute bi-stable elements (where R=0 and S=1 & R=1 and S=0 are being two stable states). Now if we set R=0 and S=1, the output of the first NOR gate is forced to 1 and as a chained inverter, we get Q=1 and $\bar{Q}$= 0. In the same way when the inputs R=1 and S=0, the output of the second NOR gate is forced to 1 and so we have Q=0 and $\bar{Q}$ =1. If both the input levels are high i.e., R=1 and S=1, then both the NOR gates are force to deliver the low outputs i.e., Q=0 and $\bar{Q}$ =0. Now, if we set both the inputs as 0s concurrently, the output of the Flip-flop either switches to Q=1 and $\bar{Q}$ =0 or Q=0 *and* $\bar{Q}$ =1, it is observed that they even oscillates between these two situations which is not desired. So these input are always to be avoided.

When we are going to implement the RS Flip-flop, we need two buffered Boolean NOR gates. For testing RS Flip-flop, we initiate with the inputs R=0 and S=0 given that Q=0 and $\bar{Q}$ =1. Now, as we set S=1 the output of first NOR will be high i.e., Q=1 and $\bar{Q}$ =0. These two values are memorized when S returns to 0. Next, when the input value R becomes high (=1), the outputs are Q=0 and $\bar{Q}$ =1. These two values are stored when the value of R is returned to 0. At the last step, when we set both the inputs high=1, then both the outputs are forced to set 0s.Now, if R=0 and S=0, the simulator determines the possible tunnel events and after resolving situations, the simulator shows the output results as Q =1 and $\bar{Q}$ =0. So, at last, we conclude that the RS Flip-flop gives the correct behavior shown in Table-6.

7. Characteristic equation of RS Flip:

For writing the characteristic equations of the RS Flip-flop, we are to write the truth table of it, then with the help of the Karnaugh map the characteristic equations of Q^{t+1} and $\bar{Q}^{t+1}$ are written in the equations (12) and (13) respectively. The truth table is given in Table-7.

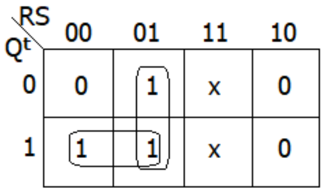
$$Q^{t+1} = \bar{R}Q^t + \bar{R}S \quad \text{.....................} (12)$$

$$\bar{Q}^{t+1} = \bar{S}\bar{Q}^t + R\bar{S} \quad \text{.....................} (13)$$

Table-7

Truth table for RS Flip-flop

		Previous		New	
R	S	Q^t	$\bar{Q}^t$	Q^{t+1}	$\bar{Q}^{t+1}$
0	0	1	0	1	0
0	1	1	0	1	0
1	0	1	0	0	1
1	1	1	0	invalid	
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	0	1	invalid	



Q^t \ RS	00	01	11	10
0	0	1	x	0
1	1	1	x	0

$$Q^{t+1} = \bar{R}Q^t + \bar{R}S$$

It is clear that Boolean gate based RS Flip-flop keeps the unstable behavior when both of the inputs R and S are high i.e., R=S=1. This instability happens only for the cause of the cross-connection of the two NOR gates. If we are able to build an RS Flip-flop without cross-coupling gates, the instability problem will be resolved. In this circumstances, when we are intended in designing the RS Flip-flop, it will cost more number of gates. But when we are going to make the same thing with the help of threshold gate design, the gates needed will be lesser.

A threshold gate based design of an RS Flip-flop can be built by means of a single threshold gate only. By a small modification [7, 8] of the equations (12) and (13) we obtain the Threshold gate equations (14) and (15) respectively, where there will not be any occurring events of uncertainty.

$$Q^{t+1} = \bar{R}Q^t + \bar{R}S + SQ^t \quad \text{.....................} (14)$$

$$\bar{Q}^{t+1} = R\bar{Q}^t + R\bar{S} + \bar{S}\bar{Q}^t \quad \text{.....................} (15)$$

For the verification of equations (14) and (15), we draw a truth table without giving the input condition R=1 and S=1 in Table-8.

Table-8

		Previous		New	
R	S	Q^t	$\bar{Q}^t$	Q^{t+1}	$\bar{Q}^{t+1}$
0	0	1	0	1	0
0	1	1	0	1	0
1	0	1	0	0	1
1	1	1	0	1	0
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	0	1	0	1

Now for implementation of RS Flip-flop, we must verify that the functions (14) and (15) are satisfying the specified function of RS Flip-flop. If we set R=0 and S= 0, we obtain $Q^{t+1}=Q^t$ (Hold). If we set R=0 and S= 1, we have Q^{t+1}= 1 (Set). If we set R=1 and S= 0, we get Q^{t+1}= 0 (Reset). If we set R=1 and S= 1, we have $Q^{t+1}=Q^t$ (Hold). And for all cases, the output $\bar{Q}^{t+1}$ becomes the complement of output Q^{t+1}. So we decide that equation (14) and (15) satisfy the characteristics of the RS Flip-flop without any forbidden input combinations R=1, S=1or R=0, S=0 and both of these combinations corresponding to Hold function.

When we are interested in implementing the Boolean gate based logic circuit of the equation (14) we require three 2-input AND gates and one 3-input OR gate. Whereas when we want to implement it by means of threshold logic gate based circuit, a 3-input threshold gate suffices to implement the same Boolean equation (14).

Table-9

$\bar{R}$	S	Q^t	output Q^{t+1}	θ	Eqn. no.
1	0	1	1	$w_1+w_3 \geq \theta$	(1)
1	1	1	1	$w_1+w_2+w_3 \geq \theta$	(2)
0	0	1	0	$w_3<\theta$	(3)
0	1	1	1	$w_2+w_3 \geq \theta$	(4)
1	0	0	0	$w_1<\theta$	(5)
1	1	0	1	$w_1+w_2 \geq \theta$	(6)
0	0	0	0	$0<\theta$	(7)
0	1	0	0	$w_2<\theta$	(8)

We has assumed the positive logic, for this cause we will take all the coefficients of $\bar{R}$, S, Q^t i.e, w_1, w_2, w_3 and θ are positive. If we put, $w_1=1$, $w_2=1$, $w_3=1$ and $\theta=2$, the all the 8 conditions in the 5th column in Table-9 are satisfied. So, one solution set is $\{w_1, w_2, w_3; \theta\}=\{1,1,1;2\}$. The threshold logic equation of equation (14) is given in equation (16).

$Q^{t+1}=\mathrm{sgn}\{\bar{R}+S+Q^t-2\}$.................... (16)

We know that $R+\bar{R}=1$ or $R=-\bar{R}+1$ or $\bar{R}=-R+1$, so the above equation becomes

$Q^{t+1}=\{-R+S+Q^t-1\}$.................... (17)

According to the threshold logic equation (17), the threshold logic gate is drawn in Fig. 9.

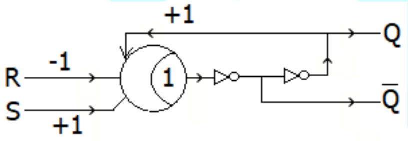


Fig.9 TLG based RS Flip-flop for eqn. (17)

Table-10

Verification of the correctness of equation (17)

R	S	Q^t	Q^{t+1}	function
0	0	0	0	Previous value
0	0	1	1	
0	1	0	1	Set
0	1	1	1	
1	0	0	0	Reset
1	0	1	0	
1	1	0	0	Previous value
1	1	1	1	

The RS Flip-flop we have proposed has been theoretically proved and verified by using the simulator and the simulation results are shown in the Fig. 10 and Fig. 10(a). We have obtained the Simulation results by using the circuit parameters given below for the threshold logic gate logic input "0"=0V, logic "1" = 16mV, $C_1^p(w_1=1)=C_2^p(w_2=1)$=$0.5aF$, $C_1^n(w_3=-1)$ =0.4aF, $C_3=12.7aF$, $C_L=9aF, C_0=8.6aF$, R_j= 10^5 $\Omega, V_s=V_b=16mV$. For the buffer, the parameters taken are, $C_{g1}=C_{g2}=0.5aF, C_1=C_4=0.5aF$, $C_2=C_3=0.1aF$, $\boldsymbol{C_{b1}}=\boldsymbol{C_{b2}}=4.25aF$v, $C_L=9aF$, R_j=50KΩ , C_j=0.1aF.

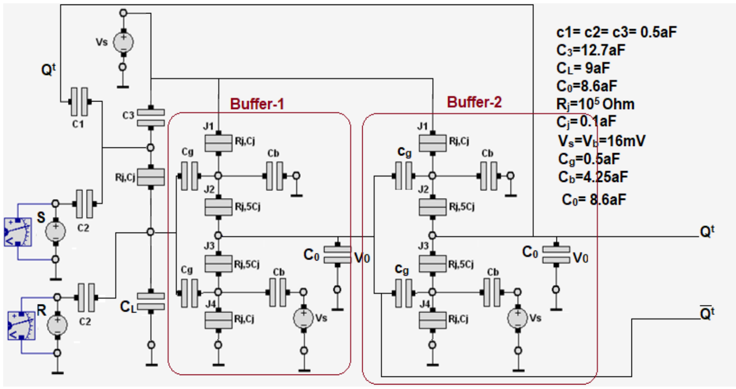


Fig. 10. Threshold Logic based RS Flip-flop

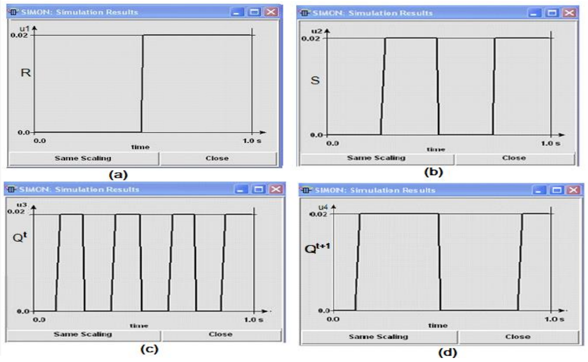


Fig. 10(a) R-input (b)S-input (c) Previous Q^t (c) Q^{t+1}=New Q^t

8. T Flip-flop

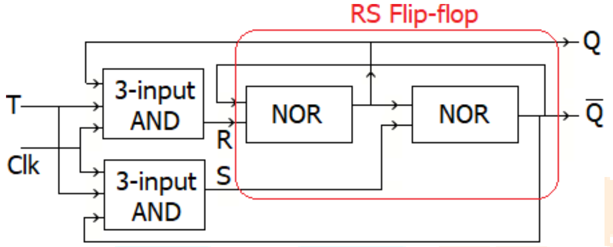


Fig. 11 Flip-flop based on Boolean gates

T Flip-flop can be used as a memory cell or memory unit or memory element for retrieving or storing the information. T Flip-flop consists of two 3-input AND gates, Two NOR gates, two input terminals T and Clk, and two output terminals Q and Q̄. The behavior of triggered T Flip-flop shown in Table-11 is as follows. For every positive clock pulse, a transition happens. If T and current Q are low, then the output Q i.e., Q^{t+1} is low, If T and current Q are different values i,e., one is high and another is low, then Q^{t+1} is high. And when both T and current Q are high, then the output Q i.e., Q^{t+1} is low. Unlike RS Flip-flop, a T Flip-flop is having no Forbidden or Unspecified input combination(s).

Table-11
Truth Table of T Flip-flop

Clk	T	Q	Q^{t+1}	$\bar{Q}^{t+1}$	Function
$^1_0$ ⎍	0	0	0	1	If both T and Q values are "0", then Q(t+1)=0
$^1_0$ ⎍	1	0	1	0	If T and Q values are different then Q(t+1)=1
$^1_0$ ⎍	0	1	1	0	If T and Q values are different then Q(t+1)=1
$^1_0$ ⎍	1	1	0	1	If both T and Q values are "1", then Q(t+1)=0

One possible T Flip-flop implementation based on Boolean logic gates is drawn in Fig.11. The operation of the circuit is as follows. The two cross-coupled NOR gates forms an RS Flip-flop. If T=0, then irrespective of Q(current), both R and S will be 0 and value of Q^{t+1} will be last Q. Now if T=1 then the value of R and S will be depended upon the present value of Q and Q̄. If Q=0 and Q̄=1, the value of Q^{t+1} will be complement of last Q i.e., Q^{t+1} =1, and if Q=1 and Q̄ =0, the value of Q^{t+1} will be complement of last Q i.e., Q^{t+1} =0.
In no case, the values of R and S will be high (=1) simultaneously.

9. Implementation of T Flip-flop using Threshold Logic Gate

The logic function of T Flip-flop on the basis of Boolean gate can be written as

$$Q^{t+1}=T\bar{Q}^t + TQ^t \qquad (18)$$

and

$$\bar{Q}^{t+1}=TQ^t +T\bar{Q}^t \qquad (19)$$

We know that $\overline{Q}^{t+1}$=NOT (Q^{t+1}). We are familiar with the expression Y=A.$\overline{B}$ +$\overline{A}$.B and it is similar to the equation (18). Now for representing the Boolean function Q^{t+1}=T.$\overline{Q}^{t}$ +T.Q^{t} using threshold logic gate, first we take P= (T.$\overline{Q}^{t}$); as it is an AND function, so using equation (8) we can write for P as.

P = *sgn* {T+ $\overline{Q}^{t}$—2}.............................. (20)

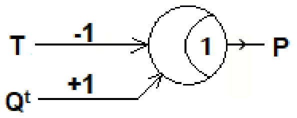
As we know $\overline{Q}^{t}$ + Q^{t} =1 or $\overline{Q}^{t}$= — Q^{t} + 1, the equation (20) can be written as equation (21).

P = *sgn* { T- Q^{t} - (1) }........................... (21)

The threshold Logic gate of equation (21) is drawn in Fig. 12(a). For stability of this gate we can connect two buffers in series to the output point of P shown in Fig. 12(b). The truth table of equation (21) is given in Table-12.

Table-12
Truth table of equation (23)

T	Q^t	P
0	0	0
0	1	0
1	0	1
1	1	0



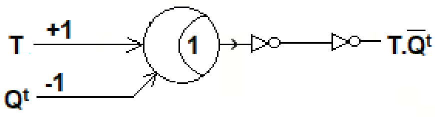


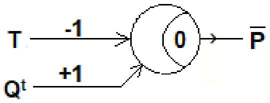
Fig. 12(a)

threshold Logic gate of equation (23) Fig. 12(b) Modification of TLG of equation (23)

For correct operation, we shall modify the threshold equation of P in equation(21), since the buffer inverts its input signal, so that it calculates $\overline{P}$=sgn {−T+Q^{t}−(0) } and truth table regarding this expression is given in Table-13. The threshold logic gate of equation $\overline{P}$ is drawn in Fig. 13(a) and its complement in Fig.13 (b).

Table-13
Truth table of $\overline{P}$

T	Q^t	$\overline{P}$
0	0	1
0	1	1
1	0	0
1	1	1



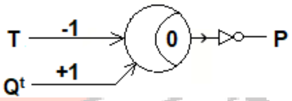


Fig. 13(a) Threshold logic gate of $\overline{P}$ Fig. 13(b) Complement of Fig. 14(a)

Now is the time for us to express the Boolean expression Q^{t+1}= T.$\overline{Q}^{t}$ +T.Q^{t}. We put P= T.$\overline{Q}^{t}$

So, Q^{t+1}= T.$\overline{Q}^{t}$ +T.Q^{t} = P + T.Q^{t}.

⟹Q^{t+1} = P + $\overline{T}$.Q^{t} .... (22)

For having the threshold gate logic of equation (22), we draw the truth Table-14 with the help of Table-13 and from which we solve the minimum value solution to get the weight values.

Table-14

P	T	Q^t	Q^{t+1}	θ
0	0	0	0	$0<\theta$
0	0	1	1	$W_3 \geq \theta$
1	1	0	1	$W_1 + W_2 \geq \theta$
0	1	1	0	$W_2 + W_3 < \theta$

After solving the conditional equations in the 5th column in Table-14 we obtain a solution set which is { w_1 , w_2 , w_3 ; θ} = {2, -1, 1; 1}

Hence the Threshold equation for the T Flip-flop is

Q^{t+1} = sgn { 2P — T + Q^{t}—(1) }................. (23)

And its corresponding Threshold logic gate is drawn in Fig. 14**.**

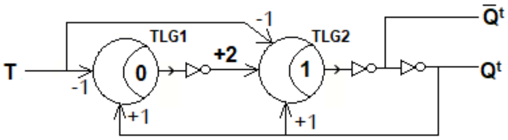


Fig. 14 Threshold logic gate based T Flip-flop

For the simulation purpose of the T Flip-flop, we have chosen related parameters. For Threshold logic gate1 (TLG1):$C_0 = 8.1aF$, R_j=$10^5\Omega$, $V_s = V_b = 16mV$,$C_1^p(w_1 = 1)$=$0.5aF$,$C_1^n(w_3 = 1)$ =0.4aF, For the Threshold Logic gate 2 (TLG2) :$C_1^p(w_1 = 2) = 1aF$,$C_2^p(w_2 = 1)$=$0.5aF$,$C_1^n(w_3 = -1)$ =0.4aF, $C_3 = 13.4aF$, $C_L = 9aF$,$C_0 = 8.6aF$, R_j=10^5 Ω,$V_s = V_b = 16mV$. For the buffer, the parameters taken are, $C_{g1} = C_{g2} = 0.5aF$,$C_1 = C_4 = 0.5aF$,$C_2 = C_3 = 0.1aF$, $\boldsymbol{C_{b1}} = \boldsymbol{C_{b2}} = 4.25aF$v, $C_L = 9aF$, $R_j =$ 50KΩ, C_j=0.1aF. The simulation set and its simulated results are shown in Fig. 15 and Fig. 15(a) respectively.

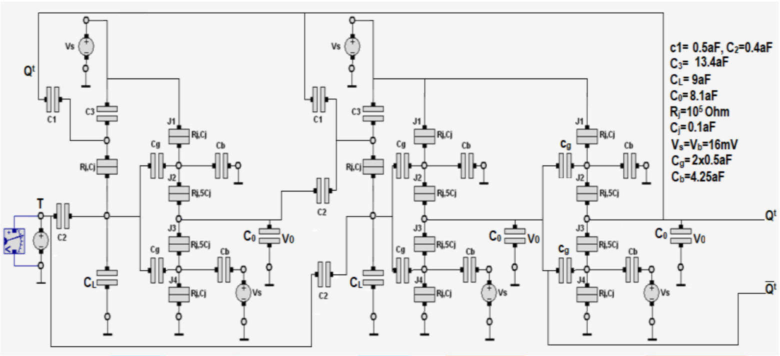


Fig. 15 T-Flip-flop Threshold logic gate

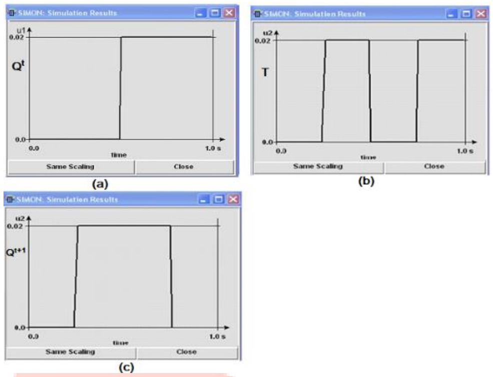


Fig.15 (a) Q^t-input (b) T-input (c) Q^{t+1}=New Q^t

Till now, we have been able to implement a T-Flip-flop without trigger signal. When anyone wants to make a register or counter he can add a clock input signal for synchronization purpose.

10. A sequential circuit

Our motive is to construct a sequential circuit by using threshold logic gates. The state table of this circuit is produced in Table -15. The table represents two flip-flops X_1 , X_2 , one input "a" and output Y.

Table-15

Initial state		Input	Next state		output
X_1	X_2	a	X_1	X_2	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

Next-states and output Y can be written directly from Tabble-15. The logic equations for the next-state and output are obtained from the Table-15 as below:

$X_1(t+1)\ \sum(4,6)$ ……………………………… (24)

$X_2(t+1) = \sum(1,2,5,6)$………………… (25)

$Y(X_1,X_2,a)$ =$\sum(3,7)$ …………………… (26)

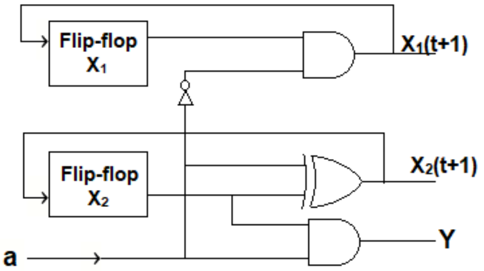
All the minterms are being the variables for X_1, X_2 and "a" which are really the initial-state and input variables. We can simplify the functions given in terms of minterms by means of Karnaugh maps to provide the following.

$X_1(t+1) = X_1\bar{a}$……………………………… (27)

$X_2(t+1) = X_2 \oplus a$ ……………………… (28)

Y (X_1,X_2,a) = $X_2\ a$ …………………… (29)

With the help of the 3 equations (27), (28) and (29), we have drawn the Boolean logic circuit of the sequential circuit as given in Fig. 16. The same circuit can also be drawn on the basis of threshold logic gate is drawn in Fig. 17.



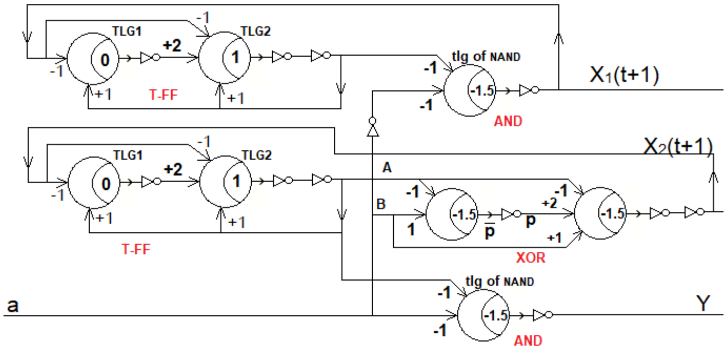
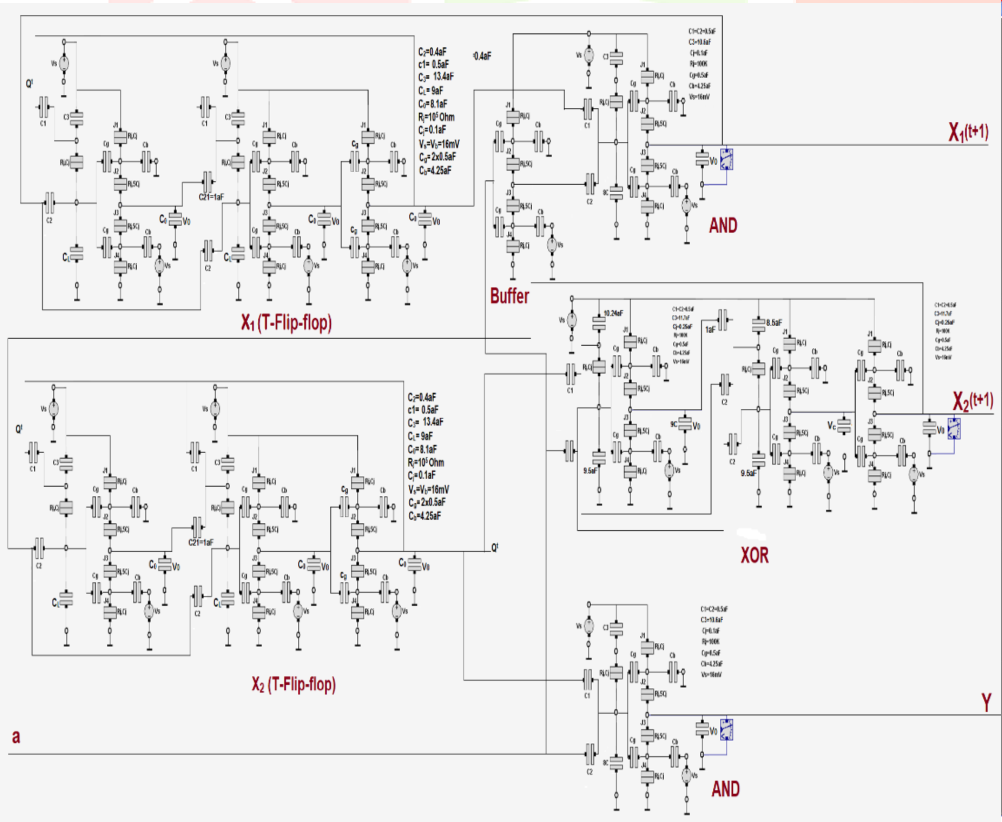


Fig. 16. Sequential circuit equations (27)--(29)

Fig. 17. Sequential circuit using TLGs

On the background of the circuit drawn in Fig. 17, we have tried to implement it using the simulation set and the simulation results have been collected by using simulation software SIMON and the simulation result is given in the Fig. 18(b). Noted that all the parameters we have chosen for the sequential circuit are given below as.

For T Flip-flop, we have chosen related parameters as. For Threshold logic gate1 (TLG1):$C_0 = 8.1aF$, R_j=$10^5\Omega$, $V_s = V_b = 16mV$,$C_1^p(w_1 = 1)$=$0.5aF$,$C_1^n(w_3 = 1)$ =0.4aF, For the Threshold Logic gate 2 (TLG2) :$C_1^p(w_1 = 2) = 1aF$,$C_2^p(w_2 = 1)$=$0.5aF$,$C_1^n(w_3 = -1)$ =0.4aF, $C_3 = 13.4aF$, $C_L = 9aF$,$C_0 = 8.6aF$, R_j= 10^5 Ω,$V_s = V_b$ = $16mV$. For the buffer, the parameters taken are, $C_{g1} = C_{g2} = 0.5aF$,$C_1 = C_4 = 0.5aF$,$C_2 = C_3 = 0.1aF$, $\boldsymbol{C_{b1}} = \boldsymbol{C_{b2}} = 4.25aF$v, $C_L = 9aF$, R_j = 50KΩ, C_j=0.1aF. **For XOR**, we have chosen, logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^n = C_1^P = 0.5C_2^p = \frac{1}{2}C$ =0.5aF, $\boldsymbol{C_b} = 8.5aF$, $\boldsymbol{C_j} = 0.25aF$, $C_L = 9aF$,$C_0 = 9.5aF$, R_j= 10^5 Ω,V_b = $15.81mV$ that is close to 16mV, so V_b =V_s=16mV. **For AND gate,** we have used the parameters $\boldsymbol{C_1^n} = \boldsymbol{C_2^n}$ = 0.5aF, $\boldsymbol{C_{b1}} = \boldsymbol{C_{b2}}$ = 4.25aF, $C_{g1} = C_{g2} = 0.5aF$,$C_L = 9aF$,$C_0 = 8aF$, R_j = 10^5 Ω.



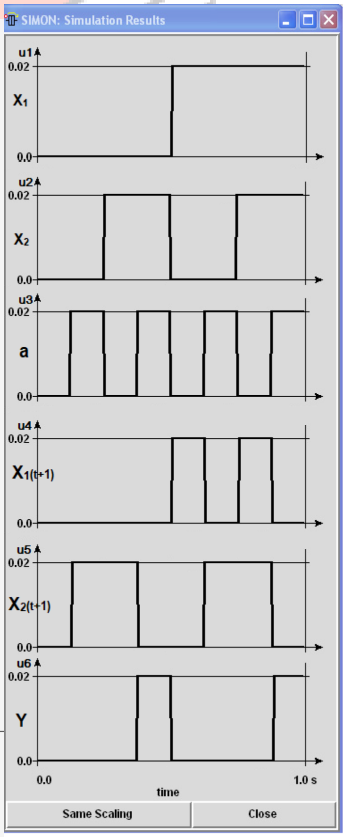


Fig. 18(a) Simulation set of sequential circuit

Fig. 18(b) simulation result of Fig. 18(a)

11. Delay

Every electronic devices have some processing delays when they are being used. All the LTG based circuits we have drawn in this paper must have some delays. We have analyzed regarding 2-input AND/XOR, Buffer, RS Flip-flop, T-Flip flop and a sequential circuit based on LTGs and simulation sets. Now, we shall light on the analysis of their speed. For finding out the processing delays of the logic circuits discussed above, we shall involve the critical voltage V_c and the tunnel junction capacitance C_j. However, we are to consider the temperature at T = $0K$, the processing delay of any logic gate can be determined by means of the approaches [8, 9, 10] given below.

Delay = $-(e|\ln(P_{error})|R_t)$ / ($|V_j| - V_c$) ……. (30)

Where V_j is the junction voltage, V_c is the critical voltage and R_t is the junction resistance.

Whenever the critical voltage V_c, of course, is less than the tunnel junction voltage V_j, i.e., $V_c < |V_j|$, then tunneling event happens. This happens, for example, in a 2-input AND gate in Fig-3(c) when V_{in1} is 16mV, giving V_j =11.8mV, the critical voltage of the tunnel junction V_c is 11.58mV. Given that the probability of error P_{error} is equal to 10^{-12} and tunnel resistsnce $R_t = 10^5\Omega$. By calculating, we obtain a gate delay equal to 0.062| ln(P_{error}) |ns = 1.71ns. In the same way, we can find out all the circuit delays written in Table-17. Whenever an electron goes through the tunnel junction, the amount of total energy being changed in the circuit after the tunneling events. The difference between the energy levels before and after the tunneling event is determined by the equation (31).

$$\Delta E = \boldsymbol{E}_{before\ tunnel} - \boldsymbol{E}_{after\ tunnel}$$
$$= -e(V_c - |V_j|) \cdots\cdots\cdots\cdots (31)$$

This is the amount of switching/tunneling energy (ΔE) which is consumed whenever a tunnel event occurs in the tunneling circuit.

We have had curves of the switching delay Vs. switching error probability in Fig. 19(a) and the switching delay Vs. the unit capacitance C shown in Fig. 19(b).

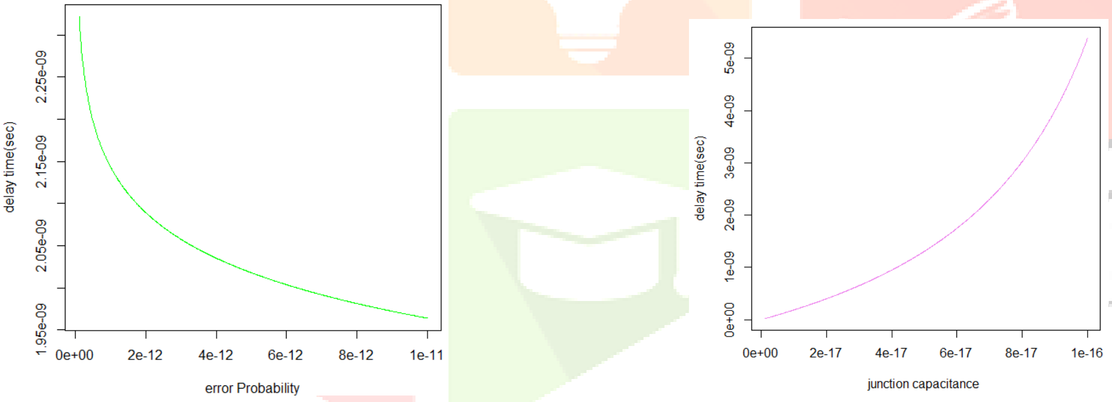


Fig. 19(a) Delay vs. Error Probability Fig. 19(b) Delay Vs. capacitance

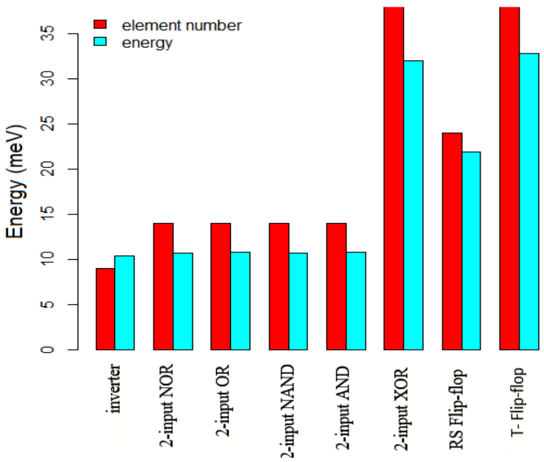


Fig. 20 comparison of Elements and Switching energy od different gates

We have counted the number of elements in every gates or circuits, their switching delays, and the switching energies consumed by the corresponding LTGs by using the same methodology as adopted for the Boolean gates. All the calculated parameters are provided in tabular form in Table-16.

The element numbers and how much energy they are consuming, are shown by vertical bars w.r.t. individual gates in Fig. 20.

Table-16

Gate/Device	SET-based delay	LTG-based delay
inverter	8	0.60ns
2-input NOR	4	1.67ns
2-input OR	4	1.71ns
2-input NAND	4	2.21ns
2-input XOR	20	4.87 ns
RS Flip-flop	16	3.06 ns
T- Flip-flop	20	4.7 ns
Sequential circuit	40	9.5 ns

Next, we have given the attributes collected from this work regarding element numbers, delays, and switching energy in Table-16.

The processing delays for 2-input AND gate the witching delay is 0.080|ln(P_{error})| ns, for 2-input AND gate it becomes 0.062|ln(P_{error}) | ns, and for sequential circuit it is 0.346|ln(P_{error})| ns. The value of P_{error} is considered to be 10^{-12}, so the time after which 1st output bit of the sequential circuit will fan out is 0.346|ln(P_{error})| ns=9.56ns. i.e., after every 9.56 ns, the next output bit will be taken from the sequential circuit. Therefore time/duration of the clock signal (not given in the sequential circuit) should be greater than or equal to 9.56 ns. In this situation, the speed of the sequential circuit will be $1/9.56ns$ = 1.04GHz.

Table-17

Gate/Device	elements	Delay	Switching Energy
inverter	09 elements	0.022$\lvert\ln(P_{error})\rvert$ ns	10.4 meV
2-input NOR	14 elements	0.072$\lvert\ln(P_{error})\rvert$ ns	10.7 meV
2-input OR	14 elements	0.062$\lvert\ln(P_{error})\rvert$ ns	10.8 meV
2-input NAND	14 elements	0.080$\lvert\ln(P_{error})\rvert$ns	10.7 meV
2-input AND	14 elements	0.062$\lvert\ln(P_{error})\rvert$ns	10.8 meV
2-input XOR	38 elements	0.173$\lvert\ln(P_{error})\rvert$ ns	32.0 meV
RS Flip-flop	24 elements	0.111$\lvert\ln(P_{error})\rvert$ ns	21.9 meV
T- Flip-flop	38 elements	0.173$\lvert\ln(P_{error})\rvert$ ns	32.8 meV
Sequential circuit	151 elements	0.346$\lvert\ln(P_{error})\rvert$ ns	129.6 meV

We want to know the circuit delays of CMOS, SET-based and LTG-based gates. The processing delay or switching delay for a CMOS logic gate like AND, NOR, NAND, XOR is 12ns [15, 16], whereas time required for tunneling through a single electron transistor (SET) [9, 10] is close to 4ns [2, 3, 4, 5, 7, 8].

Switching delays of SET and LTG

Assuming the error probability is 10^{-12} , then the delay for the 2-input OR gate will be 1.71ns and similarly the delays for the other gates can be calculated and are all produced in Table-16. When comparing, we observe that LTG based circuit is faster than the SET based circuit when P_{error}=10^{-12}. The comparison of delays between SET and LTG based circuits is represented by a bar diagram depicted in Fig.21.

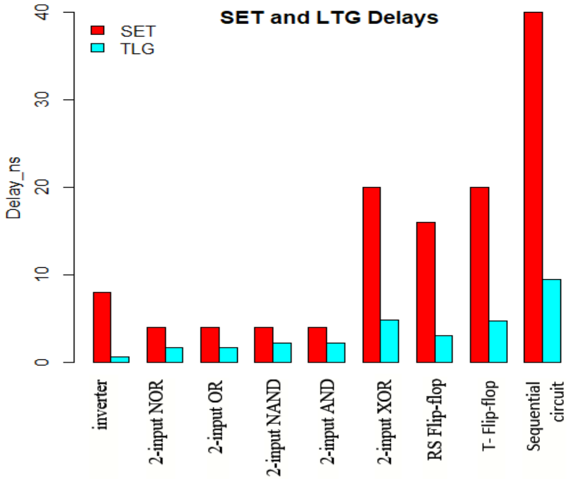


Fig. 21 Delays of SET and LTG

12. Conclusion

For the purpose of implementing the sequential circuit we need AND gate, XOR gate, memory cells (T-Flip-flops) and buffers. All of them are discussed in due places. For implementing different gates or circuits we have used the buffer many times for complementing action. A generic Linear Threshold logic gate implementation is discussed from which a family of logic gates has been implemented. All the threshold logic gate circuits for sequential circuit are drawn with the help of their threshold logic equations. Last, a threshold logic gate circuit for sequential circuit is shown citing the parameters required for implementing it. All the gates or circuits presented in this work have been implemented and are verified by means of simulation using SIMON [14]. The number of elements requiring for logic gates, and other circuits, their processing delays, power consumption by them are given. It is seen that the threshold logic gates are at least 2-times faster than SET based logic gates. The temperature of the operation is to be maintained at 0K in real situation.

13. REFERENCES

[1] Anup Kumar Biswas, "Application of single electron threshold logic gates and memory elements to an up-down Counter" International Journal of Creative Research Thoughts (IJCRT) | Volume 9, Issue 6 June 2021 | ISSN: 2320-2882

[2] Anup Kumar Biswas, -"Implementation of A 4n-Bit Comparator based on IC Type 74L85 using Linear Threshold Gate Tunneling Technology" International Journal of Engineering Research & Technology ISSN: 2278-0181, Vol. 10 Issue 05, May-2021 pp.299-310,

[3] 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-2021pp.325-336, ISSN: 2278-0181

[4] Anup Kumar Biswas, "Design of A Pipeline for A Fixed-Point Multiplication using Single Electron Tunneling Technology", International Journal of Engineering Research & Technology ISSN: 2278-0181, Vol. 10 Issue 04, April-2021 pp. 86-98,

[5] 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 IJEST Vol. 1, No. 1, 2009, pp. 61-73

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

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

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

[9] 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 TRANS. ON NANOTECHNOLOGY, VOL. 3, NO. 2, JUNE 2004

[10] C. Lageweg, S. Cot¸fan˘a, and S. Vassiliadis, "A linear threshold gate implementation in single electron technology," IEEE Computer Society VLSI Workshop, Apr. 2001, pp. 93

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

[12] J. R. Tucker,- "Complementary digital logic based on the Coulomb-blockade", Journal of Applied Physics., vol. 72, no. 9, pp. 4399–4413, November 1992.

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

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

[15] Jacob. Millman and C. C. Halkias; "Integrated Electronics- Analog and Digital Circuits and Systems -second edition" McGraw Hill Education;

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

BIOGRAPHY

ANUP KUMAR BISWAS, born in 1971, he defended his Ph.D.[Engg.] thesis in 2005 in the stream of Electronics and Telecommunication Endineering at Jadavpur University. He was a Senior Research Fellow in Faculty of Engineering and Technology (*FET*) from 2002 to 2005 at the Department of Electronics &TC. Dr. Biswas has published 25 papers in national, international journals and conferences. The area of his scientific interests includes computer arithmetic, parallel architectures, nanotechnology, and computer-aided design. He is engaged in research activities and teaching last 16 years.