IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Linear Threshold Tunneling Technology based 4-bit Binary-cum-BCD Adder

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

ABSTRACT

Single Electron tunneling based threshold gate is one approach by that we can implement all logic gates, sequential and combinational circuits. Threshold Logic Gates (TLGs) and Single Electron tunneling devices (SEDs) keep the capabilities of controlling the transport of an electron through a tunnel junction at a particular time. A single electron containing the charge is adequate to store an information in a SED. Power required in the single electron tunneling circuits is low in comparison with the (CMOS) circuits. The processing delay is very low and speed of the processing of TLG based devices will be close to electronic speed. The single-electron transistor (SET) and TLG are attracting scientists, technologists and researchers to design and implement for the consuming of ultra-low power and their small sizes. All tunneling events in a TLG-based circuit happen when only a single electron tunnels from one conductor to other through the tunnel junction when the proper voltages are applied. For implementing logic gates, Full adder and a binary-cum-BCD adder, TLG would be a best candidate to fulfill the necessities requiring their implementations. So far as an Ultra-low noise is concerned, TLG based circuit can be considered as a best selection for implementing the desired tunneling circuits. Different TLGs like 2-input AND, 3-input OR, Full adder, and a binary-cum-BCD adder have been implemented by using linear threshold logic gates or devices. Threshold network, truth table and simulated results of them are given in parallel in due places.

Key words: BCD adder, Electron-tunneling, Coulomb-blockade, linear threshold gate

1. INTRODUCTION

From the perspective of semiconductor technology that the ever decreasing feature size and the corresponding increasing in density of transistors facilitates many improvements in semiconductor based designs. But one day such improvement will come to an end in the long run. For ensuring further feature size reduction, possible emerging technologies with greater scaling potential like single electron tunneling technology is currently under the investigation of the researchers/scientists. Threshold logic gate based circuits are made with tunnel junctions, through which an electron can be passed in a controlled manner.

Single Electron tunneling based device is such an equipment by which all logic gates and more complex circuits can be implemented. Tunneling events happen when a single electron can pass through the tunnel junction under the action of bias voltage and multiple input voltages connected to the islands via small true capacitances. For implementing a binary-cum-BCD adder, TLG would be a suitable candidate.

2. Multiple input threshold logic gate

A gate called "multiple input threshold logic gate" [1-7, 9, 10] is a gate that is made up of a tunnel junction possessing a capacitance C_j and a resistance R_j, two multiple input-signals V_k^Ps and V_l^ns connected at two points 'p' and 'q' shown in Fig.1. Each input voltage V_k^P, for the top left side, is connected to the point "q" through their corresponding capacitances $C_k^P s$ and each input voltage V_l^n, in the bottom left side, is connected to the point "p" through their corresponding capacitances C_l^ns. Supply voltage or Bias voltage V_b is connected to the point "b" through a true capacitor C_b as well. Junction capacitor C_j is connected to point "p" which has been grounded through another capacitor C_0. We will be able to implement the LTGs with the assistance of a function presented by the signun function(x) of h(x) expressed by equations (1) and (2).

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

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

where x_k being the n-Boolean inputs and w_k being their corresponding *n* integer weights.

The LTG will compare the weighted sum of the inputs $\sum_{k=1}^{n} (w_k \times x_k)$ with the threshold value θ, if the weighted sum-value is greater than or equal to the threshold or critical voltage value θ then the logic output of the LTG would be high (logical "1"), otherwise it will be low (logical "0").

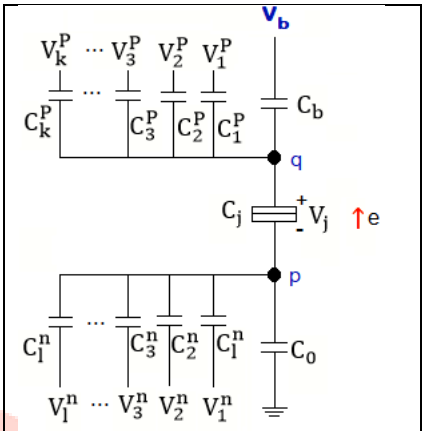


Fig. 1 Multiple input TLG

The tunnel junction capacitance C_j and the capacitance C_0 connected in series are being considered as the two basic circuit elements in a LTG. The input signal voltages $V_1^P, V_2^P, V_3^P, \ldots, V_k^P$, which are weighted by their corresponding vector capacitances $C_1^P, C_2^P, C_3^P, \ldots, C_k^P$, are added to the junction voltage, V_j. Whereas, the input signal voltages $V_1^n, V_2^n, V_3^n, \ldots, V_l^n$ (which are weighted by their corresponding vector capacitances $C_1^n, C_2^n, C_3^n, \ldots, C_l^n$), are being subtracted from the voltage, V_j.

The critical voltage V_c is essential to enable tunneling action, and which functions as the intrinsic threshold of the tunnel junction circuit. The supply or bias voltage V_b connected to tunnel junction through the capacitance, C_b, is used to adjusting the gate threshold to the desired value θ. A tunneling event happens though the tunnel junction, when an electron goes through the junction from p to q as directed by an arrow in Fig. 1.

The following notations will be followed for the rest our discussions.

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

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

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

When we assume that all voltage sources in Fig. 1 are connected to ground, then the circuit can be considered that it is made up of three capacitances namely, C_Σ^P, C_Σ^n and C_j, connected in series. The symbol C_T is assigned to the sum of all 2-term products of C_Σ^P, C_Σ^n and C_j.

Now we must find out the expression regarding the critical voltage V_c of the tunnel junction. We assume the tunnel junction capacitance to be C_j and the remaining part of the circuit has the equivalent capacitance as C_e. As observed from the point of view of tunnel junction, we can measure the threshold or critical voltage [1,2, 8,9,10] for the tunnel junction as below.

$$V_c = \frac{e}{2(C_j + C_e)} \quad \cdots\cdots (6)$$

$$V_c = \frac{e}{2[C_j + (C_\Sigma^P || C_\Sigma^n)]}$$

$$= \frac{e}{2[C_j + \frac{(C_\Sigma^P) * (C_\Sigma^n)}{(C_\Sigma^P + C_\Sigma^n)}]}$$

$$= \frac{e(C_\Sigma^P + C_\Sigma^n)}{2[C_j * (C_\Sigma^P + C_\Sigma^n) + (C_\Sigma^P) * (C_\Sigma^n)]}$$

$$= \frac{e(C_\Sigma^P + C_\Sigma^n)}{2C_T} \quad \cdots\cdots (7)$$

When the voltage of the junction is V_j , a tunneling event happens through this tunnel junction if and only if the condition below is satisfied.

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

From this equation it is decided that if the critical voltage is greater than the junction voltage i.e. $V_c > |V_j|$, then no tunneling events through the tunnel junction happens. As a consequence, the tunneling circuit remains in its *stable state*.

Theoretically, the thresholds are integer numbers though it can be taken as a real number. The threshold logic equations for 2-input AND, 3-input OR gates are shown in sections from 4 and 5.

3. Buffer

The buffer or inverter [1, 2, 3, 4, 8, 9, 10] depicted in Fig. 2(a) is made up of two single electron transistors (SETs) connected in series. The two input voltages of same values are directly coupled to the islands of the SET1 and SET2 [1, 2, 8, 9, 10] through two true capacitors C_1 and C_2 of same values respectively. The islands of each SETs have a size close to 10 nm diameter of gold and their capacitances should be less than 10aF. The output terminal V_0 is connected to the common channel between SET1 and SET2 and to the ground through a load capacitor C_L to put down charging effects.

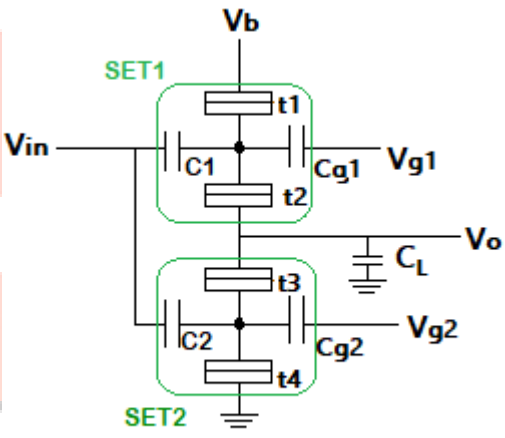
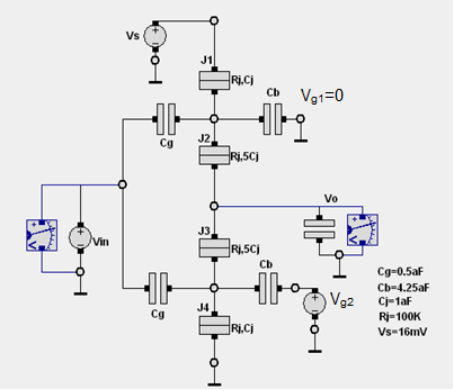


Fig.2 (a) an Inverter

Fig.2 (b) Symbol of an Inverter

For the buffer, the parameter 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=50KΩ. For the purpose of simulation, the value of C is taken as 1aF.



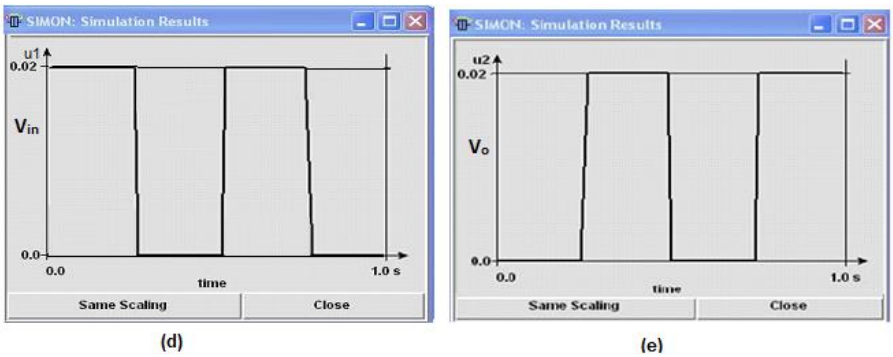


Fig. 2(c) Simulation set of Buffer **Fig.2 Simulation result of Buffer (d) input (e) output**

The operation of the buffer will be discussed as: - the output V_0 value will be high in case the input voltage V_{in} is low and the V_0 value will be low in case the input voltage is high. To achieve this target, we must set the voltages $V_{g1} = 0\ and\ V_{g2}$ =16mV along with the gate voltages, at present, V_{in} both for SET1 and SET2. SET1 will be in conduction mode if V_{in} is set to low and the SET2 is in Coulomb blockade [2, 3, 4, 13]. This results the output voltage V_0 is connected to V_b and therefore the output voltage becomes high. Coulomb blockade disturbs the steady flow of current, as the high voltage (logic 1) is applied to the input terminal(s), it makes shift the induced charge on each of the islands of the two SETs by a fraction of an electron charge and causes the SET1 to keep in Coulomb blockade and the SET2 in conducting mode. So, the output V_0 shifts from high to low (logic 0).

In this work, we assume the Boolean logic inputs "0" =0 Volts and logic "1"=$0.1\times \frac{q_e}{C}$.

For the cases of simulation and other purposes, will consider that C=1aF and Logic "1"= $0.1\times \frac{1.602\times10^{-19}}{1\times10^{-18}} = 0.1 \times 1.602 \times 10^{-2}$=$16.02 \times 10^{-3}$ =$16.02 \cong 16$ mV.

4. 2-input AND gate

For making the threshold logic gate of an AND gate, we first draw the truth table Table-1 of AND gate and compare the weights w_A and w_B of two variables A and B respectively to the threshold value θ [1,2,3,9,11].

Table-1

A	B	F(A,B)	θ	Eqn. No.
0	0	0	$0<\theta$	(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)

For positive logic, we assume weights of A and B are positive 1 each. Then from the above four inequalities in Table-1, if we assume w_B=1, w_A=1 and θ=2, then the four inequalities or equations in 5th column in this table are satisfied. Hence the Threshold logic equation for 2-AND gate is given in equation (9) and its corresponding threshold logic gate is drawn in Fig. 3(a)

$$AND(A,B) = sgn\{A + B - 2\} \quad \text{.....(9)}$$

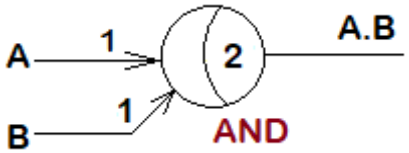


Fig. 3(a) Threshold logic AND gate

For implementing purpose, the AND gate we will use has the parameters $C_1^n = C_2^n$=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. 3(a) and accordingly after simulation the result we get is given in Fig. 3(c)and the simulation set is shown in Fig. 3(b).

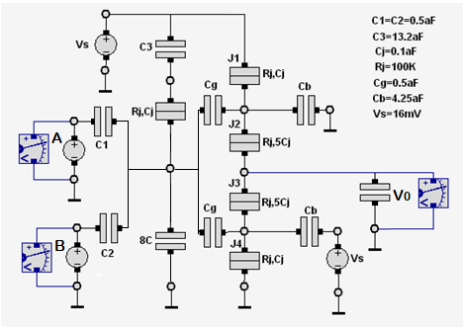


Fig. 3(b) AND Gate

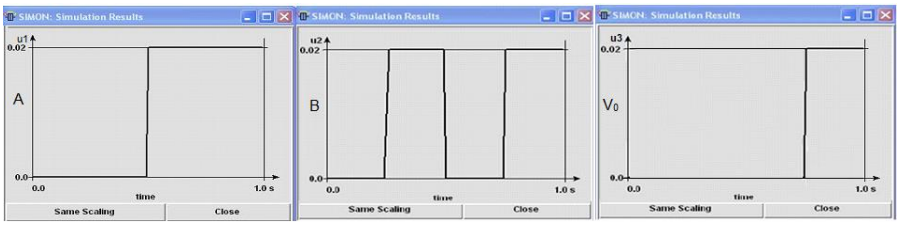


Fig.3(c) Simulation result of AND gate

5. 3-input OR/NOR Gate

Table-2
(3-input OR gate)

A	B	C	F(A,B,C) =A+B+C	θ	Eqn. no.
0	0	0	0	0< θ	(1)
0	0	1	1	$w_C \geq \theta$	(2)
0	1	0	1	$w_B \geq \theta$	(3)
0	1	1	1	$w_B + w_C \geq \theta$	(4)
1	0	0	1	$w_A \geq \theta$	(5)
1	0	1	1	$w_A + w_C \geq \theta$	(6)
1	1	0	1	$w_A + w_B \geq \theta$	(7)
1	1	1	1	$w_A + w_B + w_C \geq \theta$	(8)

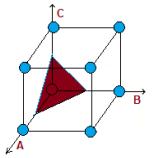
Threshold logic equation of a 3-input OR gate is defined as

$$\text{OR (ABC)} = \text{sign}\{w_A.A + w_B.B + w_C.C - (\theta)\} \quad \text{(10)}$$

For 3-input OR gate, F(ABC)= A+B+C is a Boolean logic equation. 3-input OR is a positive input logic gate and we are inspired to assign the coefficient values of $w_A = w_B = w_C$=1. For the sake of the conditional equation 0< θ in the equation of the Table-1, we can take any value for θ in the range $0 < \theta \leq 1$. Taking an integer is better, hence θ=1. By putting the values of $w_A = w_B = w_C$=1 and θ=1 in the 8-inequalities in Table-2, we found that all the equations are satisfied. Hence, one solution set becomes $\{ w_a, w_b, w_C; \theta \} = \{1, 1, 1; 1\}$ and accordingly the threshold logic equation (10) will be:

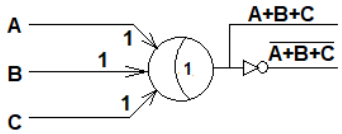
$$\text{OR (ABC)} = \text{sign}\{1.A + 1.B + 1.C - (+1)\}$$
$$= sgn\{A + B + C - (1)\} \quad \text{(11)}$$

And the state space solution diagram for F(ABC)= A+B+C has been shown in Fig. 4(a). The solution points indicating by blue small circles have the values equal to 1 for (A+B+C). It is clear from the figure that the 3-input OR gate equation is linearly separable, so a linear threshold logic gate can be drawn without any hesitation, and its corresponding TLG is depicted in Fig. 4(b) below.



(a)

Fig. 4(a) State space diagram of (A+B+C)



(b)

Fig. 4(b) 3-in OR/NOR gate

It is to be informed that for the correct operation of a 3-input OR gate, we must include an inverter. To do so, we connect a 3-in NOR gate and an inverter in series and we will find out a threshold logic equation for that 3-in NOR gate. Consider the Table-3.

Table-3
(3-input NOR gate)

A	B	C	F(A,B,C) $=\overline{A+B+C}$	θ	Eqn. no.
0	0	0	1	$0 \geq \theta$	(1)
0	0	1	0	$w_C < \theta$	(2)
0	1	0	0	$w_B < \theta$	(3)
0	1	1	0	$w_B + w_C < \theta$	(4)
1	0	0	0	$w_A < \theta$	(5)
1	0	1	0	$w_A + w_C < \theta$	(6)
1	1	0	0	$w_A + w_B < \theta$	(7)
1	1	1	0	$w_A + w_B + w_C < \theta$	(8)

In the Table-3 there are 8 inequalities. 1st inequality tells us that θ must be a non-positive number, we consider $\theta = -0.5$. As NOR gate is negative logic so their weights must be negative, one can take the values of $w_A = w_B = w_C = -1$ and when he/she will put the values to the 8 inequalities in Table-3, he will observe that all the equation are satisfied. Hence one solution set is$\{w_A, w_B, w_C ; \theta\} = \{-1, -1, -1 ; -0.5\}$. Depending upon this solution set, a threshold logic equation is given in equation (12) and its threshold logic gate is depicted in Fig. 5.

NOR (ABC) = $sgn\{-1.A - 1.B - 1.C - (-0.5)\}$
$= sgn\{-A - B - C - (-0.5)\}$ ..... (12)

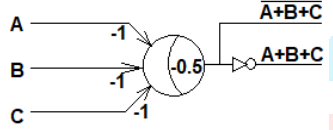


Fig. 5 A 3-input TLG NOR/OR

When we refer to a logic function such as 3-input OR, we will indicate the logic function performed by the whole gate (i.e., threshold gate output + buffer). So, 3-input OR = (3-in NOR + buffer) is to be accepted and is drawn in Fig.5(a) with its simulated waves in Fig. 5(b), (c), (d), and (e). For the purpose of implementing 3-input OR, the simulation the parameters to be taken are as: logic input "0"=0V, logic "1" = 16mV, $C_1^n = C_2^n$=0.5aF,$C_b = 11.7aF, C_b = C_{b1} = C_{b2} = 4.25aF,\ \ C_{g1} = C_{g2} = 0.5aF,\ \ C_L = 9aF, C_0 = 8aF,\ R_j$=$10^5\ \Omega, V_s = V_b\ = 16mV$.

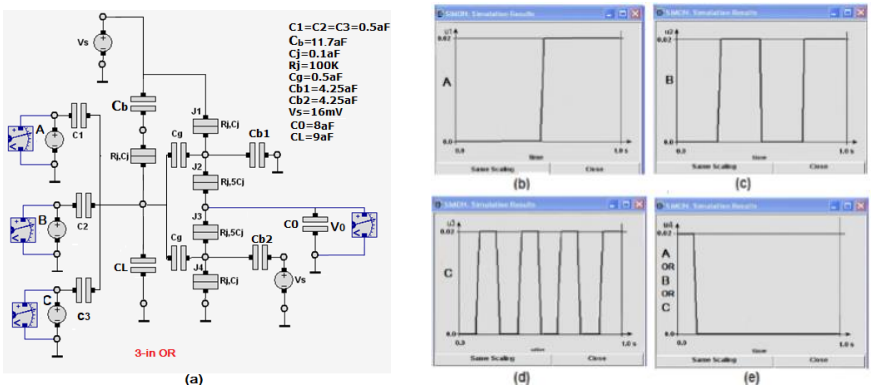


Fig. 5(a) 3-in OR gate; (b), (c), (d) and (f) are simulation results

6. Regarding Full Adder

We consider two inputs of addition "a" and "b", and carry is "c_0". Boolean expression of sum of a Full adder is S=a⊕b⊕c_0=$\bar{a}bc_0$+$a\bar{b}c_0$+$ab\bar{c}_0$+abc_0 which is not linearly separable, because the solution space can't be

separated by a plane into two parts of green-colored indicating 1-points and colorless circle points indicating 0-points as shown in Fig. 6(a) below.

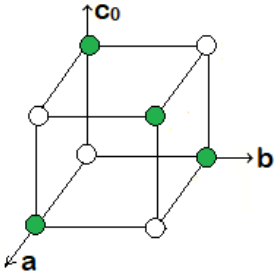


Fig. 6 (a) space plot diagram of S

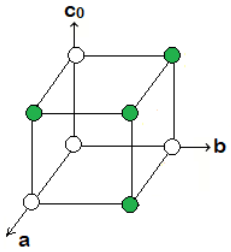


Fig. 6(b) space plot of $c_1=ab+bc_0+c_0a$

From the carry-out equation $C_1=ab+bc_0+c_0a$, if we draw the space solution diagram in Fig. 6(b), it will be clear to us that the 1-points are linearly separable from 0-points. So, the C_1 can be turned into a linear threshold gate. Consider the equation (13) below,

$$C_1=sgn\ \{w_1.a+ w_2.b+ w_3.C_0-\theta\}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(13)$$

As C_1 is a positive logic, we must take the weights as low positive integer values i.e., $w_1=w_2=w_3=1$. From the first inequality of the Table-4, the value of θ must be positive. Since we have taken $w_1=w_2=w_3=1$ and $w_1=w_2=w_3<\theta$ (from the inequalities (2), (3) and (5) in Table-4), we take the minimum integer value of θ equal to 2. After analyzing logically, we can take the parameter values as: $w_1=w_2=w_3=1$ and $\theta=2$. Now, if we put these values in the conditional equations in the 5th column in Table-4, all the inequalities are satisfied. So, one solution set of the parameters will be as $\{w1,w2,w3;\theta\}=\{1,1,1;2\}$. So the threshold equation of Carry C_1 is shown in equation (14)

$$C_1= sgn\{a+b+c_0-2\}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(14)$$

Table-4

Inputs			output	Threshold Condition	Eqn.. No.
a	b	c_0	c_1	θ	
0	0	0	0	$0<\theta$	(1)
0	0	1	0	$w3<\theta$	(2)
0	1	0	0	$w2<\theta$	(3)
0	1	1	1	$w2+w3\geq\theta$	(4)
1	0	0	0	$w1<\theta$	(5)
1	0	1	1	$w1+w3\geq\theta$	(6)
1	1	0	1	$w1+w2\geq\theta$	(7)
1	1	1	1	$w1+w2+w3\geq\theta$	(8)

According to this solution set, we can draw the threshold logic gate in Fig.7.

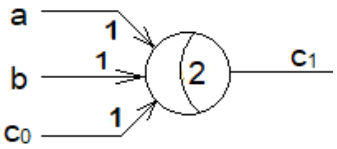


Fig. 7 TLG diagram of C_1

Now for making the threshold logic gate of Full adder we will take the sum equation $S=\bar{a}bc_0+ a\bar{b}c_0+ ab\bar{c}_0+ abc_0$, and carry equation $c_1=ab+bc_0+c_0a$ and construct the truth table of sum S, assuming the four variables a, b, c_0 and c_1. The truth table is given in Table-5.

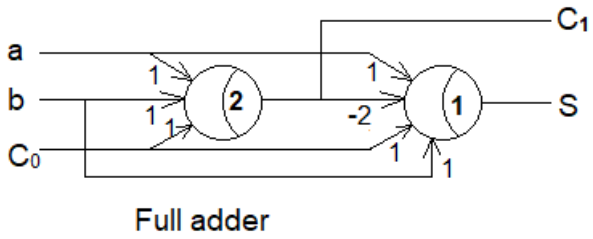
From the inequality (1) in Table-5, we have the value of θ to be positive. Since sum S is a positive logic, so the weights of variables values would be positive. Therefore, we can take them as lowest positive integers i.e., w1=w2=w3=1. Under these assumptions and comparing the equation in inequalities (2), (3) and (5) from the Table-5, we should take the value of θ as minimum positive integer 1, i.e., θ=1. After that, if we put the value of $w_4=-2$, the all the inequalities are satisfied. So, one solution set is $\{w_1, w_2, w_3, w_4; \theta\}=\{1,1,1,-2;1\}$. therefore the threshold equation of Sum (S) is given in equation (15).

$S=sgn\{a+b+c_0-2c_1-1\}$…….(15)

According to the solution set, the threshold logic gate is depicted in Fig.8.

Table-5

Inputs			out put	Sum	Threshold Condition	Eqn. No.
a	b	c_0	c_1	S	θ	
0	0	0	0	0	$0< \theta$	(1)
0	0	1	0	1	$w3\geq \theta$	(2)
0	1	0	0	1	$w2\geq \theta$	(3)
0	1	1	1	0	$w2+w3+w4 < \theta$	(4)
1	0	0	0	1	$w1\geq \theta$	(5)
1	0	1	1	0	$w1+w3+w4 < \theta$	(6)
1	1	0	1	0	$w1+w2+w4 < \theta$	(7)
1	1	1	1	1	$w1+w2+w3+w4 \geq \theta$	(8)



Full adder

Fig. 8 Full Adder Circuit

For preventing back propagation effect and correct operation, one buffer circuit must be appended in series with every Threshold Logic Gate. To do so, first the carry of the Full adder is taken into our justification. Complement of threshold logic of carry C_1 is thought of in equation (16).

$\bar{C}_1$ =sgn {w1.a+ w2.b+ w3.c_0-θ}…….(16)

Inputs			output	Threshold Condition
a	b	c_0	$\bar{c}_1$	
0	0	0	1	$0\geq \theta$
0	0	1	1	$w3\geq \theta$
0	1	0	1	$w2\geq \theta$
0	1	1	0	$w2+w3< \theta$
1	0	0	1	$w1\geq \theta$
1	0	1	0	$w1+w3< \theta$
1	1	0	0	$w1+w2< \theta$
1	1	1	0	$w1+w2+w3< \theta$

Table-6

From the Table-6, it would be clear that if we take the value of θ equal to -1.5 and w1= -1, w2= -1 and w3= -1 then all the conditions in last column of the Table-6 are satisfied. Hence the equation turns to the following equation (17).

$\bar{C}_1$ =sgn $\{-a-b-c_0-(-1.5)\}$……. (17)

As per the equation (17), somebody can draw the threshold logic gate with a buffer to get C_1 shown in Fig.9.

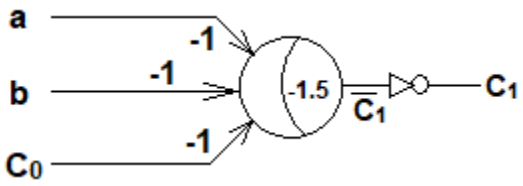


Fig. 9 Carry of a full Adder using buffer

Now we shall find out the complement of sum $S = sgn\ \{a + b + c_0 - 2c_1 - 1\}$.

Consider $\bar{S}$=sgn $\{w1.a+w2.b+w3.c_0+w4.c_1-(\theta)\}$

Table-7
Truth table of $\overline{sum}$ of a full adder

Inputs				$\overline{sum}$	Threshold Condition
a	b	c_0	c_1	$\bar{S}$	θ
0	0	0	0	1	$0 \geq \theta$
0	0	1	0	0	$w3 < \theta$
0	1	0	0	0	$w2 < \theta$
0	1	1	1	1	$w2+w3+w4 \geq \theta$
1	0	0	0	0	$w1 < \theta$
1	0	1	1	1	$w1+w3+w4 \geq \theta$
1	1	0	1	1	$w1+w2+w4 \geq \theta$
1	1	1	1	0	$w1+w2+w3+w4 < \theta$

$$\bar{S} = sgn\{w1.a + w2.b + w3.c_0 + w4.c_1 - (\theta)\} \quad \cdots\cdots (18)$$

After solving the conditional equations in 6th column of Table-7, we get a solution set as $\{w1, w2, w3, w4;\ \theta\} = \{-1, -1, -1, +2;\ -0.5\}$ for the equation (18). Hence, the complement of Sum equation will be

$$\bar{S} = sgn\ \{-a - b - c_0 + 2c1 - (-0.5)\} \quad \cdots\cdots (19)$$

As per the equation (19), we can implement the threshold logic gate with a buffer to get sum S.

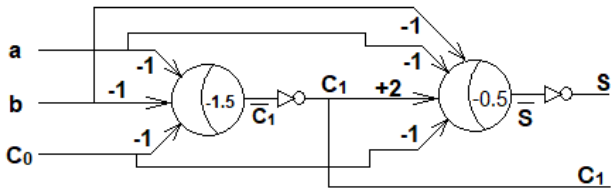


Fig. 10 Full Adder using TLG and Buffer

The full adder based on threshold gate and buffer depicted in Fig. 10 is more stable and correct in comparison with Fig. 8. Because as soon as we would like to draw a circuit on the basis of TLG we must append a buffer in series to that TLG for having correct answer.

When we want to simulate the full adder we choose the parameters [3, 9, 10, 11] as below.

The threshold logic gate input logic "0"=0V, logic "1" = 16mV, C=1aF, $C_1^P = C_2^P = C_3^P = C_1^n$ =0.5aF, $C_b = 18.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, . $V_b = 12.8mV$.

The simulation set of a Full adder and the simulation results of the Full adder (using SIMON) is depicted in Fig. 11(a) and 11(b),(c), (d), (e) and (f) respectively.

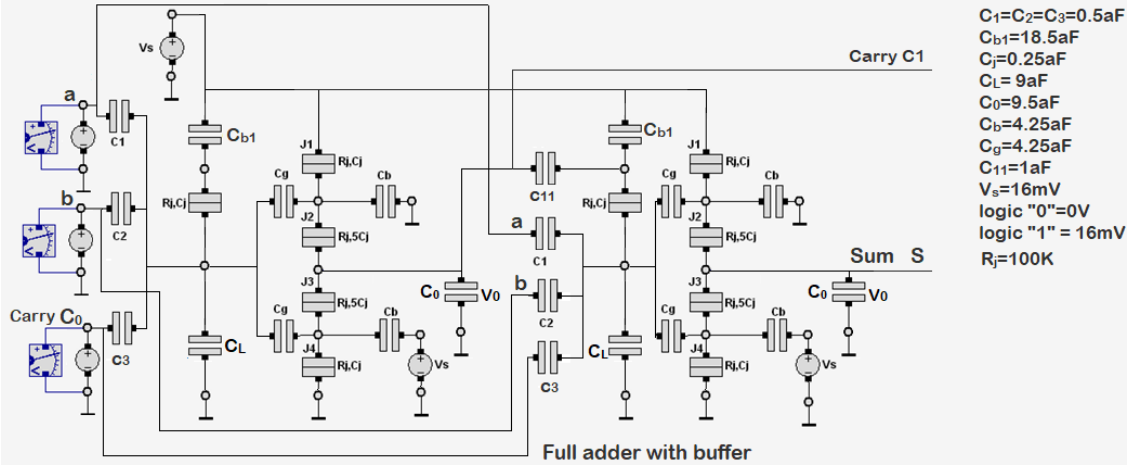


Fig. 11(a) Full adder with buffer(s)

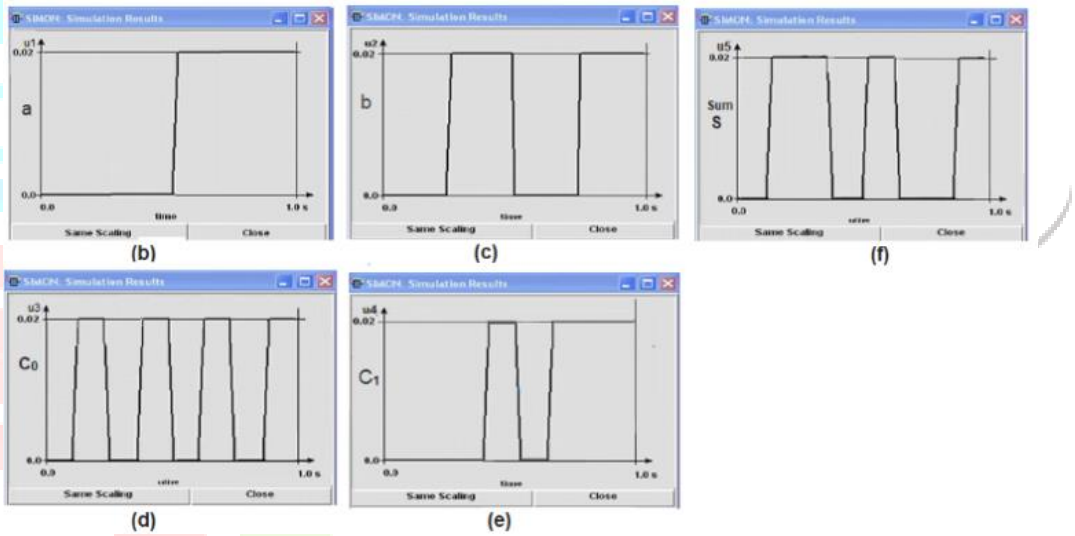


Fig. 11(b) Simulation result of variable a, (c) of b,
(d) of c_0, (e) of c_1 and (f) of sum S of Full adder

7. Description of BCD addition

Now the arithmetic addition of two decimal digits in BCD is considered, together with a possible carry from a previous stage. For a four bit BCD adder, the input digit does not exceed 9, so the output sum cannot be higher than (9 + 9 + 1) = 19, the 1 in the sum is being an input carry (C_0). Assuming, two BCD digits to a 4-bit binary adder are set in. The adder forms the sum in *binary* and it creates a result that may range from 0 to 19. These binary numbers are written in the Table-8 and are labeled by the symbols *K*, Z_8, Z_4, Z_2, and Z_1. Where *K* becomes the carry, and the subscripts written under each letter Z are representing the weights 8, 4, 2, and 1 which can be assigned to the four bits in the BCD code. The left-hand side column in Table-8 lists the binary sums as they appear in the outputs of a 4-bit *binary* adder (upper part of Fig.12). The output-sum of two *4-bit numbers* are represented in BCD and they appear in the form listed in the middle column of the table. While investigating the contents of the table, it is transparent that when the binary sum is equal to or less than 1001 (9), the corresponding BCD number is identical to the binary, and there will be no conversion is necessary. But as the binary sum exceeds 1001, we get a non-valid BCD

representation. If we add binary 6 (0110) to the binary sum then the addition converts it to the correct BCD representation and produces an output carry as required also.

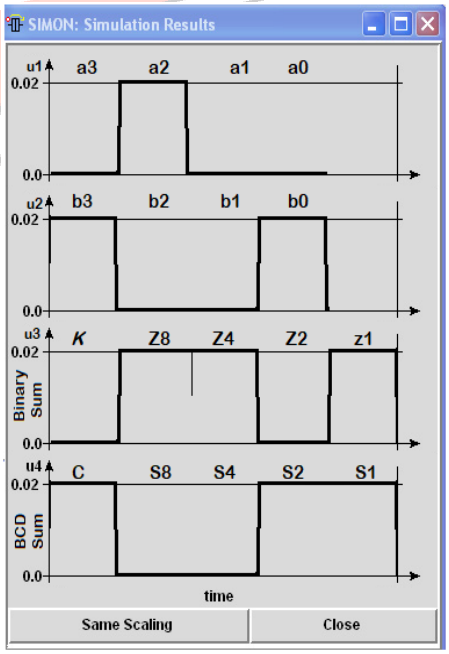
The logic circuit to be implemented detects the necessary correction are derived from the table entries. It is clear that a correction must requires when the binary sum will have an output carry $K = 1$. The other six combinations starting from 1010 to 1111 that require a correction have a 1 in position Z_8. To differentiate them from binary numbers 1000 and 1001, which also possess a 1 in position Z_8, we specify again that either Z_4 or Z_2 must have a 1. Therefore the condition for a correction and an output carry can be represented by the Boolean function as

$$C = K + Z_8Z_4 + Z_8Z_2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (12)$$

Table-8

Binary Sum					BCD SUM					Decimal Value
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

If C = 1, we must add 0110 to the binary sum and provide an output carry (=1) for the next stage. The *BCD adder* is a circuit which adds two BCD digits and generates a sum digit that must be in BCD. The BCD adder must have the correction logic in its internal construction. For adding 0110 to the binary sum for the purpose of creating the BCD numbers, we use another 4-bit binary adder, as shown in Fig. 12 (lower part). The two decimal digits, and one input carry (C_0), are first added in the top 4-bit binary adder for the purpose of producing the binary sum. Till the output carry (shown in Fig.8 the output carry terminal) becomes zero, nothing will be added to the binary sum. But when it is equal to one, a binary number 0110 is added to the binary sum in the bottom 4-bit binary adder. The output carry producing from the bottom binary 4-bit adder is avoided, as it provides information which has already been available at the output-carry terminal.



In the present work, a decimal parallel adder which adds *4* decimal digits needs *4* BCD adder stages. The output carry from one stage (starting from right hand side) will be connected to the input carry of the next higher order stage. Truth table of derivation of BCD Adder is presented in Table-8. In accordance with the discussion and the Truth table presented in Table-8, we will be able to construct a binary-cum-BCD adder [16]. The block diagram for this binary-cum BCD adder is shown in Fig. 12 and its corresponding threshold logic diagram is depicted in Fig. 13. The simulation result is provided in Fig. 13(a) also.

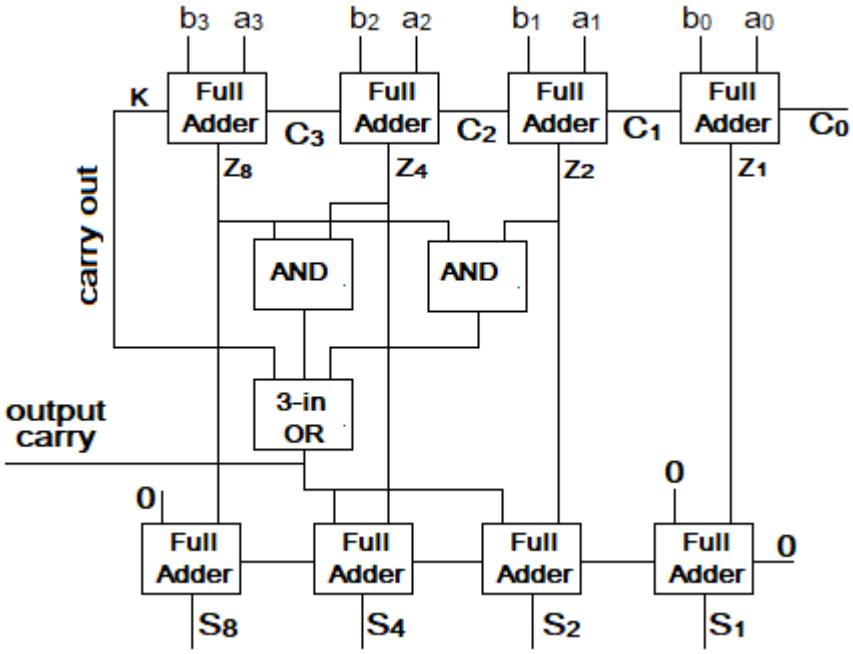


Fig. 12 Block Diagram of a BCD Adder

Fig. 13(a) one simulation result of BCD adder

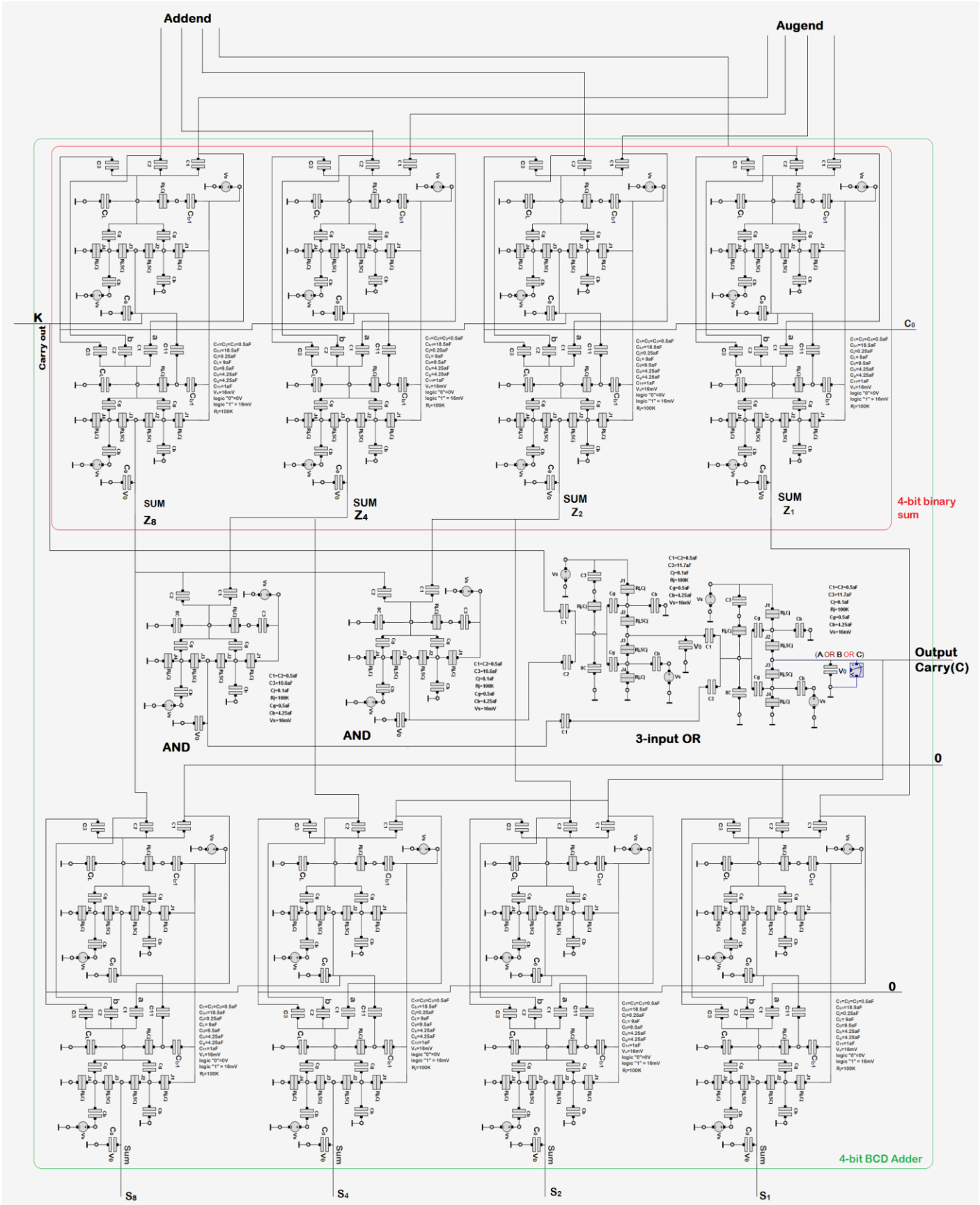


Fig. 13. 4-bit binary-cum-BCD Adder

8. Discussion

For this work, we will require 2-input AND, 3-input OR and full adder on the basis of LTG gates. We are interested in processing speed which is slow or fast. For calculating the executing /processing delay of any logic gates or circuits, we must involve the critical voltage "V_c" given in equations (6) and (7), and the tunnel junction capacitance "C_j". However, considering the atmosphere temperature at $T = 0K$, the switching/processing delay of a logic gate can be calculated by means of the approach [8, 29].

Delay = $-(e|\ln(P_{error})|R_t) / (|V_j| - V_c)$ ..... (13)

where V_j is the junction voltage and V_c is the critical threshold voltage and R_t being the junction internal resistance.

The switching will happen as soon as the critical voltage V_c has the value lower than the junction voltage V_j, i.e., $V_c < |V_j|$. This happens when V_{in1} is logic 1, resulting V_j =11.8mV for the case of a 2-input AND gate in Fig-3(b), the critical voltage of the tunnel junction voltage V_c is 11.58mV [2, 9, 10]. It is considered that the probability of error change P_{error} =10^{-12} and tunnel resistsnce $R_t = 10^5\Omega$. After calculation we obtain a gate delay equal to $0.062|\ln(P_{error})|$ns = 1.71ns. In this way, we can calculate the circuit delays listed in Table-9. Just when an electron passes through the tunnel junction, the amount of total energy in the circuit alters after the tunneling events. The difference between the energy levels before and after the tunneling event is found as [9]

$$\Delta E = E_{before\ tunnel} - E_{after\ tunnel} = -e(V_c - |V_j|) \quad \cdots\cdots (14)$$

and it is the switching energy being consumed at the time of a tunnel event occurring in the tunneling circuit. We have drawn curves as regards the switching delay as a function of the switching error probability in Fig. 14(a) and the switching delay as a function of the unit capacitance C depicted in Fig. 14(b).

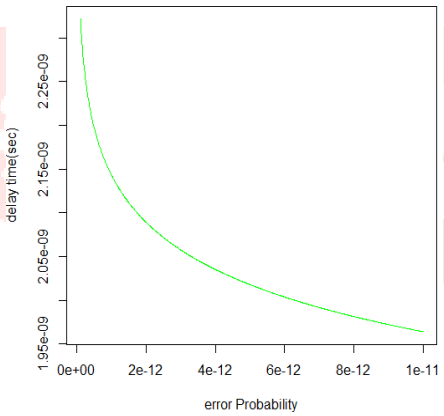


Fig. 14(a) Delay vs. Error Probability

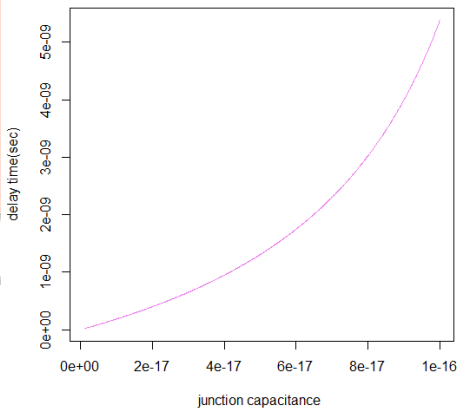


Fig. 14(b) Delay Vs. capacitance

We have computed the element numbers for each type of gates or circuits, their switching delays, and switching energy consumptions for their corresponding LTGs (using the same methodology as adopted for the Boolean gates). All the parameters found out are shown in tabular form in Table-9. Next, we have listed the delays for some gates, Full adder and BCD adder in Table-10.

The switching delays for different logic gates and circuits are different. For instance, for 2-input OR, the switching delay is $0.062|\ln(P_{error})|$ ns, for a 3-input AND gate it is $0.104|\ln(P_{error})|$ ns, for full adder it is $0.134|\ln(P_{error})|$ ns[1,2,3,8] ,and for BCD adder it is $0.476|\ln(P_{error})|$ ns. Given that the value of P_{error} = 10^{-12}, so the time after which the 1st output of the bit binary-cum BCD adder will fan out is $0.476|\ln(P_{error})|$ ns =13.14 ns. i.e., after every 13.14 ns, the next output bit will be taken from the adder circuit. Hence, the clock time/duration of the clock signal should be more than or equal to 13.14 ns provided synchronization is essential. In this situation, the speed of the binary-cum BCD adder will be $1/13.14\ ns$ = 76.1MHz.

Table-9

Gate/Device	elements	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
3-input AND	28 elements	0.104|ln(P_{error}) | ns	21.6 meV
3-input NAND	28 elements	0.072|ln(P_{error}) | ns	21.4 meV
2-input XOR	20 elements	0.102|ln(P_{error})| ns	21.2 meV
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
4-input AND	42 elements	0.104|ln(P_{error}) | ns	32.4 meV
RS Flip-flop	24 elements	0.082|ln(P_{error}) | ns	21.2 meV
T Flip-flop	23 elements	0.082|ln(P_{error}) | ns	21.1 meV
Carry C1	14 elements	0.062|ln(P_{error})| ns	10.8 meV
Full adder	29 elements	0.134|ln(P_{error})| ns	54.0 meV
Binary-cum BCD adder	296 elements	0.476|ln(P_{error})| ns	151.2 meV

We have a feeling of interest in comparing the circuit delays of CMOS, SET-based and LTG-based. The switching delay for a CMOS logic gate like AND, NAND, NOR, XOR is 12ns [14, 15], whereas the time required for tunneling through a single electron transistor (SET) [1- 4] will be approximately 4ns [2, 3, 4, 5, 14,15].

Table-10
Switching delays of SET and LTG

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 AND	4	1.71ns
3-input AND	8	2.87ns
3-input NAND	8	1.98ns
2-input XOR	4	2.81ns
3-input OR	8	2.87ns
4-input OR	12	2.87ns
4-input AND	12	2.87ns
RS Flip-flop	8	2.26ns
T Flip-flop	8	2,26ns
Carry C1	4	1.71 ns
Full adder	8	3.70 ns
Binary-cum BCD adder	28	13.14 ns

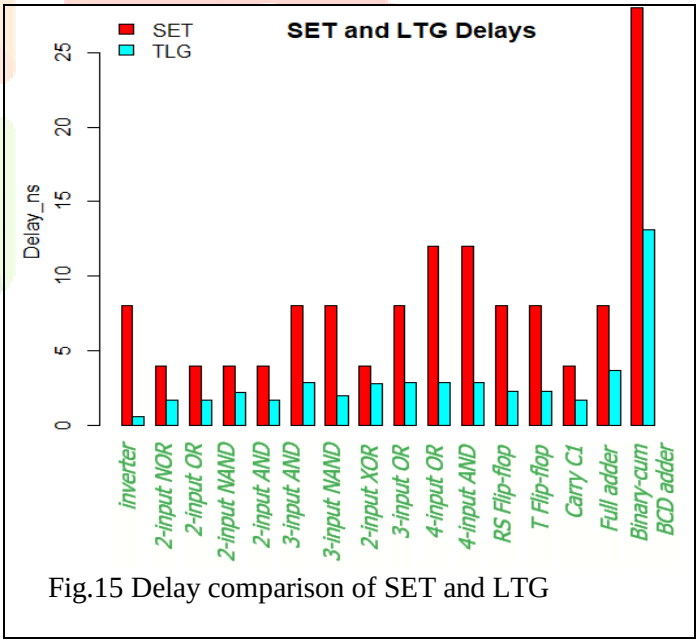


Fig.15 Delay comparison of SET and LTG

Assuming, the error probability is 10^{-12} then the delay for the 3-input OR gate will be 2.87ns and similarly the delays for the other gates can be calculated and they are all listed in Table-10. It is observed to us that the LTG based circuit is faster than the SET based circuit when P_{error}=10^{-12}.

9. Conclusion

In this present work, we discussed in regard to how an electron tunnels through and an inverter. A generic Linear Threshold logic Gate implementation has been elaborately discussed in respect of its construction and from which we have been able to derive logic gates like 2-input AND, 3-input OR, Full Adder etc. All the gates along with Full Adder have been implemented and are verified by simulation using SIMON. The number of elements (like capacitors, tunnel junction) requiring for logic gates, and other circuits, their switching /processing delays, power consumed by them are listed in tabular form and their respective curves are also given in the adjacent figures. By means of threshold logic equations some TLG gates have been depicted in appropriate places. For the case of single electron tunneling technology,

we have seen that the threshold logic gates based circuits are at least 2-times faster than SET based logic gates. The binary-cum BCD Adder circuit we have implemented is verified by simulation and the result which we got from is correct. Naturally, the temperature should be kept at 0K in real operation during the execution time.

10. 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 (IJERT), Vol. 10 Issue 05, May-2021 pp.299-310, ISSN: 2278-0181

[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 (IJERT), Vol. 10 Issue 04, April-2021 pp. 86-98, ISSN: 2278-0181

[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 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": Semiconductor Physics, Quantum Electronics & Opt-Electronics. 2003. Vol 6. No.1, pp 91-96

[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." 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 TRANSACTIONS 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," 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.

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

[12] 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.

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

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

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

[16] M. Morris Mano , "Digital logic and Computer Design ",PHI India

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(SRF) in Faculty of Engineering and Technology (*FET*) from 2002 to 2005 at the Department of Electronics and Telecommunication. Dr. Biswas has published 26 papers in national, international journals and conferences. The area of his scientific interests are computer arithmetic, parallel architectures, nanotechnology, and computer-aided design. He is engaged in research activities and teaching last 18 years.