International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET218547

Threshold Logic Technology based E-cube Routing on a 4-dimensional hypercube

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

Article Info
Volume 8, Issue 6
Page Number : 07-24

Publication Issue :
November-December-2021

Article History
Accepted : 01 Nov 2021
Published: 06 Nov 2021

ABSTRACT

Hypercube network connection is formed by connecting different N number of nodes that are expressed as a power of 2. If each node has an address of m bits then the total number of nodes in the Hypercube network is $N=2^m$. In calculating the predefined routing path for the case of this E-cube network, we apply deterministic algorithm which gives a deadlock free concept. For determining predefined routing path, node addresses involved in the path are calculated by using the exclusive operation, firstly, on the node addresses of source and destination, next, on the derived nodes according to the algorithm. In the present work, the Exclusive-OR operation is performed with the help of electron-tunneling based XOR gate which is made up of Multiple input threshold logic gate. This multiple input threshold logic gate technology is really different from the existing one. By using an emerging technology we are capable of making an electronic circuit with high speed, low cost, high concentration density, light in weight, reduced gate numbers and low power consumption. This technology is relies on the condition of linear threshold logic and electron-tunneling event. When we are interested in implementing a circuit, a multi-inputs but one-output based logic-gate will be taken account of consideration. In this work, we have designed an E-cube Routing on a 4-dimensional hypercube to find out the node addresses for predefining the deadlock free routing path from source to destination. To develop this "E-cube Routing on a 4-dimensional hypercube", we must require a specific logic called Exclusive-OR gate and for this, some small components like 2-input OR gate, 2-input AND gates of different input conditions are essential. After arranging this XOR gate in a pattern discussed in section 2, a desired circuit is implemented. All the circuit we are intended to construct are given in due places with their threshold logic and simulation set, the simulation results are provided as well. Different truth tables, derivation of threshold logic expressions are given for clear understanding.

We have taken our consideration of whether the present work circuits are faster or slower than the circuits of CMOS based- and Single electron transistor (SET) based-circuits. The power consumed at the time of tunneling event for a circuit is measured and sensed that it exists in the range between 10meV to 250meV which is very small amount. All the combinational circuits we have presented in this work are of 'generic multiple input threshold logic gate'-based.

Keywords : Routing-path, Hypercube, Electron-tunneling, Threshold logic, high-speed, low-cost

I. INTRODUCTION

In network, packet routing may be conducted either by adaptively or deterministically, Depending upon the network conditions the adaptive routing is preferred. In this case an alternate path is possible. On the other hand, in the case of deterministic routing, the routing path from source to destination is strictly determined by the source point node address and goal/destination point address. That is, the routing path is measured in advance irrespective of network condition. Dimension order routing is the concept emerging from the deterministic routing algorithm.

Dimension order routing needs to select the successive channels or sequential node addresses to follow a specific order depending on the dimensions of multidimensional network. When we are considering networking in two dimensional mesh, the scheme is said to be $X-$ **Y routing**. If we think of a $\mathbf{n-}$ **cube or Hypercube** network, then the scheme is defined to be $\mathbf{E-}$ **cube routing** [1, 22].

By using a **threshold logic gate** (TLG), we are able to implement a Boolean function in an absolutely different way far from the traditional one. For TLG gate, every input variable has its corresponding coupling weight, and the gate will have an intrinsic or threshold value. As soon as the sum of all the coupling weights of all inputs variables concerned is equal to or greater than this threshold value then the output of the related gate is high(1), otherwise, the output is low (0) [4-6]. For constructing a threshold logic gate a new thought of generic Multiple input threshold logic gate is introduced. By this Multiple input threshold logic gate we are able to implement any circuit whether it is combinational or sequential or more complex [2-3].

1. E-cube routing over a n-dimensional hypercube for 2^n nodes

We think of an n-cube having 2^n nodes (peaks). Every node (p) in n-cube is coded in binary form like $p = (p_{n-1}p_{n-2}p_{n-3} \dots p_3p_0)$. The source node from which we are interested in starting routing is $s = (s_{n-1}s_{n-2}s_{n-3}\dots s_1s_0)$. And the destination node to which the routing ends is $d = (d_{n-1}d_{n-2}d_{n-3} \dots d_3d_0)$. Now we need to find the destination length from source(s) to destination (d) with the help of a small number of nodes or steps required.

We indicate the n-dimensions in the present situation as $i = 0, 1, 2, 3, \dots, n$; where the i^{th} dimension marks to the $(i-1)^{th}$ bit in a node to be represented by $q = (q_{n-1}q_{n-2}q_{n-3} \dots q_3q_0)$. Suppose q is any node along the route to be gone through. The route from source(s)

node to the destination (d) node can be determined by the following algorithm.

Algorithm:

(i) Measure the direction bit $k_i = s_{i-1} \oplus d_{i-1}$ for all i in n-dimensions i.e. $i = 1, 2, 3, \ldots, n$; Commence with dimension $i = 1$ and $q = s$ and follow the following.

(ii) Start routing from the node q to the next node as $q \oplus 2^{i-1}$ if $k_i = 1$. Skip the step whenever $k_i = 0$.

(iii) Move to the next dimension $i + 1$, that is $i \leftarrow i + 1$. If $i \leq n$ go to the step (ii), else done.

With the help of an example the algorithm cited above can be explained. Consider an E-cube routing over a 4-dimensional hypercube given in Fig. 1.

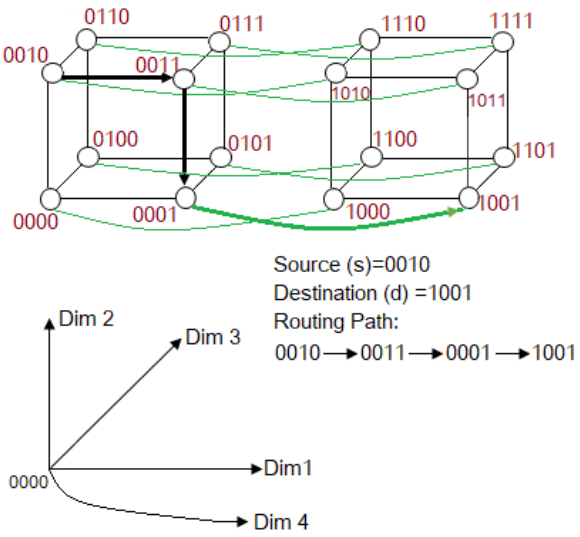


Fig. 1 E-cube routing over a 4-dimensional hypercube for 2^4nodes

Here, we have chosen the dimension $n = 4$, so the number of nodes is equal to 2^4=16, the nodes are represented by $0000, 0001, 0010, \ldots, 1111$. For the case of routing, we assume that the source and destination nodes are s=0010 and d=1001 respectively. So, $k = k_4k_3k_2k_1 = (s \oplus d)$=1011. Now we are to route from source (s) node 0010 to $s \oplus (000k_1)$=0010⊕0001=0011, since $k_1 = 0 \oplus 1 = 1$. Next route from q= 0011 to the to $q \oplus (00k_2 0)$=0001, since $k_2 = 1 \oplus 0 = 1$. Skip the dimension $i = 3$ as $k_3 = 0 \oplus 0 = 0$. Route from $q = 0001$ to the to $q \oplus (k_4000)$=1001, since k_4=0 ⊕ 1 = 1. Hence the routing path will be as:

0010→0011→ 0001 → 1001 and this path is shown in the Fig. 1 by thick lines with arrows.

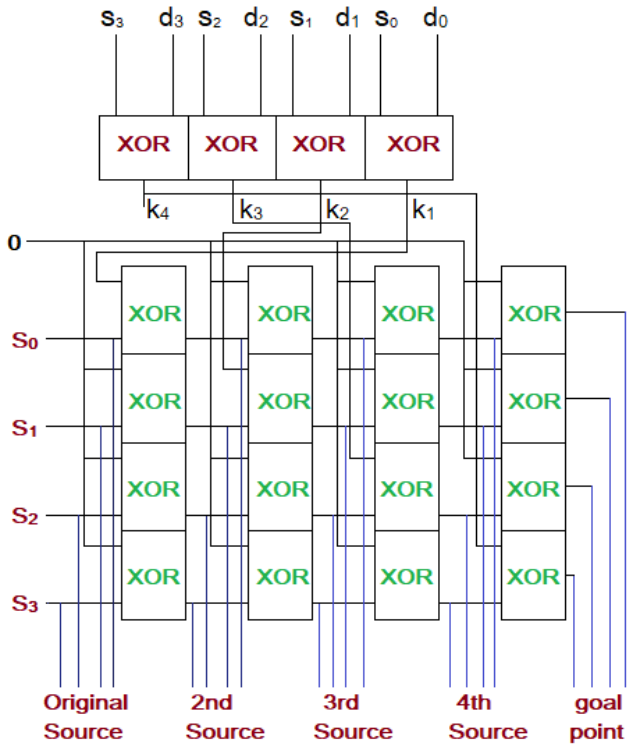


Fig.2 Block diagram of finding out the routing path

According to the algorithm and with the diagram drawn in the Fig. 2, we find it easy to get the routing path from source to destination. Some paths consisting of the nodes or vertices of the cube are listed in the Table-1 below. In the above diagram, whenever we provide the source and destination code values to input terminatals, the output results we got from the output terminals are listed in the Table-1. Depending upon the number of nodes involved in a path we have calculated the path length. Equation of the path length is given by the following relation:

$$Path\ length\ =\ number\ of\ nodes\ involved - 1.$$

For instance, consider the serial number 7 in Table-1. There are 5 different nodes (0000, 0001, 0011, 0111, 1111) in a routing path, so the $path\ length = 5 - 1 = 4$. Similarly, for serial number 9, there are 3 different nodes (1110,1010,0010) involved in the path, therefore the $path\ length = 3 - 1 = 2$.

Table-1. Minimum routing path(s)

Sl. No.	Source	Source 2	Source 3	Source 4	Goal point	Path length
1	0000	0001	0001	0001	1001	2
2	0010	0010	0010	0110	1110	2
3	0001	0001	0001	0101	1101	2
4	0011	0011	0011	0111	1111	2
5	0100	0101	0101	0001	1001	3
6	0101	0101	0111	0011	1011	3
7	0000	0001	0011	0111	1111	4
8	1001	1001	1001	1101	1101	1
9	1110	1110	1110	1010	0010	2
10	0101	0100	0110	0010	0010	3
11	1100	1100	1100	1100	0100	1
12	0000	0000	0000	0100	1100	2
13	1101	1100	1110	1010	0010	4
14	1111	1110	1100	1000	0000	4
15	1100	1101	1111	1011	0011	4
16	0011	0011	0011	0011	1011	1

II. TERMINPLOGIES

A. Threshold Logic Gate

A **threshold logic gate** (TLG) has a capable of implementing a Boolean function in a distinct way away from the semiconductor based logic gates as follows. Every input variable has its corresponding determined weight, and the gate must have an intrinsic value. When the sum of products of the weights with their ON inputs is equal to or greater than the threshold value then the output is '1'. Otherwise, the output is low (0). This operating principle can be expressed as below.

$$f = \begin{cases} 1, & \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0, & otherwise \end{cases}$$

where x_i represents input, w_i represents the corresponding weight of each input, and θ is the threshold value.

Threshold logic function (TLF):

The TLF indicating a Boolean function can be implemented by using a single TLG. The. A TLF can be realized by a vector $[w_1, w_2, \ldots, w_n : \theta]$. TLF becomes linearly separable [1-7, 14]. A TLF can be a more complex one, for instance, g = $x_1 x_2 + x_2 x_3 x_4 + x_1 x_3 + x_2 x_3 x_5$ that corresponds to compact vector as g = [4,3,3,1,1; 7] and this five variables based logical expression is represented by a single node TLG as follows.

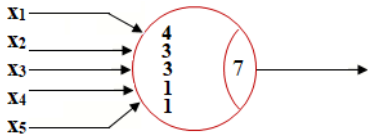


Fig.3 TLG of f = [4,3,3,1,1; 7]

The typical advantage of a threshold logic is that number of gates can be decreased to a minimum number of nodes (gates) used in the circuit, reducing area or element numbers.

Threshold Logic Properties

A TLF, of course, will be a unite function, but the reverse is not always true, that is, all the unite functions are not always TLF [3]. Hence, if a function consists of binate variables, the function cannot be

TLF. The function f = x1x2+x3x4 is an example of an unate function but it is not of a TLF. Assume that the logic function contains complemented variables, these variables can be manipulated in the similar way as the functions with all non-complemented variables, to identify whether it is a TLF. For instance, if f(x1, x2,…, xn) is a threshold function defined by [w1,w2,…,wn; θ] then its complement $\bar{f}$(x1,x2,…, xn) is also a threshold function defined by [-w1,-w2,…,-wn ;1-θ].

Majority Decision Functions (MDF)

This operating principle can also be explained in another way as Majority Decision Functions. A "majority decision function" of n variables for a logical element having n-inputs, $x_1, x_2, \dots, x_n$ and one output. The output value of the function is expressed as

$$\begin{cases} 1, & for \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0, & for \sum_{i=1}^{n} w_i x_i \leq \theta - 1 \end{cases}$$

Where, wi represents the coupling weight of each input x_i, xi represents input, and θ is the threshold value. For instance, the structure (2, 1, 1; 2) can be represented by a MDF as $x_1 + x_2x_3$; so, this function is a MDF. Whereas, the function $f(x_1x_2x_3x_4) = x_1x_2 + x_3x_4$ is not a MDF because of that it cannot be thought out by any single majority decision element.

2. Generic Multiple input threshold logic gate

A multiple threshold logic gate [1-7, 12-14] which may also be called a generic Multiple input threshold logic gate is shown in Fig. 4.

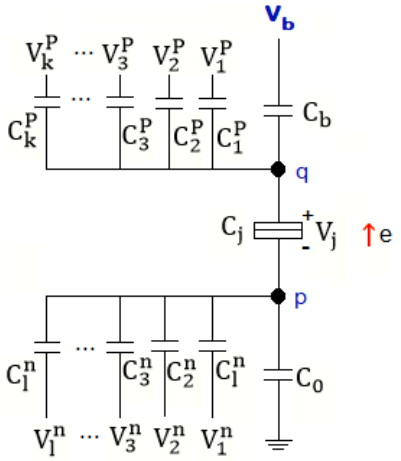


Fig. 4 TLG with multi-input

It consists of a tunnel junction having internal capacitance C_j and resistance R_j, two multi-input-signals (i) V_k^Ps and (ii) V_l^ns connected at two points 'q' and 'p' respectively. Each input voltage V_k^P (upper side) is connected to the point "q" through the capacitance C_k^P ; and each input *voltage* V_l^n (lower side), is joined at the point "p" through the *capacitance* C_l^n . The bias *voltage* V_b , for the assistance of the tunneling, is connected to the point "q" through the true capacitor C_b. Tunnel Junction capacitor C_j is linked between "p" and "q". Point "p" is grounded through C0. C_j and C0 play the important role in the circuit. For a LTG, the operation of it can be expressed by a function called *signun function*. *The signun function* of h(x) expressed by two equations (1) and (2).

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

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

where x_k represents n-Boolean inputs, and w_k are their corresponding integer weights.

The LTG compares the threshold value θ with the weighted sum of the *inputs* $\sum_{k=1}^{n}(w_k \times x_k)$. When $\sum_{k=1}^{n}(w_k \times x_k)$ is more than or equal to the threshold value θ of the circuit, then output of the

LTG shows high (logical "1"), otherwise low (logical "0").

3. 2-input AND gate (linear threshold logic)

By using the Generic Multiple input threshold logic gate, a 2-input AND gate with positive logic will be obtained easily. Consider the threshold logic expression of 2-input AND gate to be

$$AND(A,B) = sgn\{w_A A + w_B B - \theta\}\ldots\ldots. (3)$$

In order to obtain the values of the coupling weights w_A and w_B, we are to first assume $A = 1\ and\ B = 1$. Then the equation (3) turns to

$$AND(A,B) = sgn\{w_A + w_B - \theta\}\ldots\ldots. (4).$$

To obtain a 2-input threshold logic AND gate, we can draw the truth Table-2 of a 2-input AND gate and compare the coupling weights w_A and w_B of two variables A and B respectively in equation (4) with the threshold value θ [1-5,7,8].

Table-2

A	B	F(A,B)	θ	inequality
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)

As 2-input AND gate is a positive logic, we can assume the weights of A and B are 1 each. Then from the four inequalities in the Table-2, if we take w_B=1, w_A=1 and θ=2, then the all four inequalities in 4th column in Table-2 are satisfied. Hence the linear threshold logic equation for the cited AND gate becomes

$$AND(A,B) = sgn\{A + B - 2\}\cdots\cdots\cdots\cdots (5)$$

Based upon the equation (5), we can place two input capacitances $C_1^p\ and\ C_2^p$ with two positive logic inputs in the Generic Multiple input threshold logic gate given in section 4. For this purpose the parameter values we have measured are as follows: logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_2^p = \frac{1}{2}C$ =0.5aF, $C_b = 9.1aF$, $C_j = 0.25aF$, $C_L = 9aF, C_0 = 10aF$, R_j=$10^5\ \Omega$, . V_b =V_s=16mV. The corresponding 2-input threshold logic gate is given in Fig. 5(a) with its simulated results in Fig. 5(b).

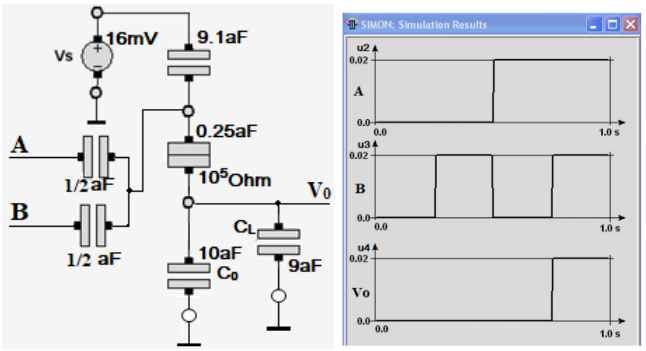


Fig. 5(a) 2-input AND Gate (b) Simulation result

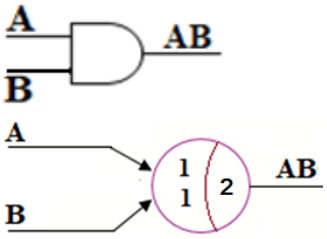


Fig.5(c) symbol of AB 5(d) TLG based node of AB

4. 2-input AND gate with one complement input

We are to explore the construction of the Fig. 5(a) and find out the threshold logic symbol for the same. If we think of the similarity of logic equations AND(A,B)=AB and AND(A,$\bar{B}$)=A$\bar{B}$, a threshold logic equation of AND(A,$\bar{B}$)=A$\bar{B}$ can be written as

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

But we know, $\bar{B}$+B=1 or $\bar{B}$=−B+1,

Putting this value into equation (6), we obtain equation (7)

F(A,B)= AND(A,$\bar{B}$)= $sgn\{A - B - 1\}$ ····· (7)

Based upon the equation (7), we can place two input capacitances $C_1^p\ and\ C_1^n$ with one positive logic input

and one negative input in the Generic Multiple input threshold logic gate. For this purpose the parameter values we have measured are as follows: logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_1^n = \frac{1}{2}C$ =0.5aF, $C_b = 9.5aF$, $C_j = 0.25aF$, $C_L = 9aF, C_0 = 9.5aF$, $R_j = 10^5\ \Omega$, . $V_b = V_s = 16$mV. The corresponding 2-input threshold logic gate is given in Fig. 6(b) with its simulated results in Fig. 6(d).

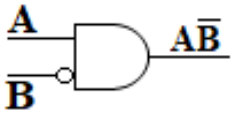
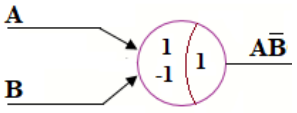
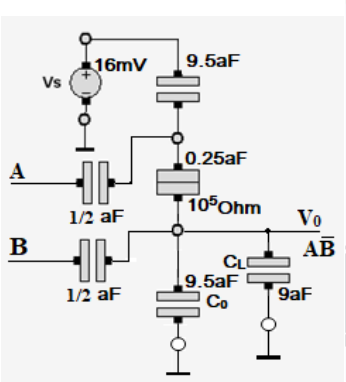


Fig. 6(a) symbol of $A\bar{B}$



6(b) TLG based node of $A\bar{B}$



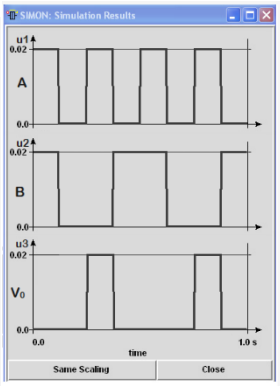


Fig. 6(c) 2-input TLG of $A\bar{B}$ Fig. 6(d) simulated result

5. Threshold logic equation for OR gate

Expression of an OR gate of two variables A and B is F(A,B)=A+B. If we want to construct the threshold logic gate of an OR gate, we first draw the truth table of an OR gate in Table-3 and compare the weights of variables w_A and w_B of two variables A and B respectively with the intrinsic or threshold value θ.

Table-3

A	B	F(A,B)	θ
0	0	0	$0<\theta$
0	1	1	$w_B \geq \theta$
1	0	1	$w_A \geq \theta$
1	1	1	$w_B + w_A \geq \theta$

For positive logic, we assume the weights of A and B to be 1 each. Then we write the three equations taken from 4th column of Table-3.

$0<\theta$ ..... (8)

$w_B > \theta$ ..... (9)

$w_A > \theta$ ..... (10)

$w_B + w_A > \theta$ ..... (11)

As $0<\theta$, θ must be positive, If we assume w_B=1, w_A=1 and θ=0.5, then the three equations in (9), (10) and (11) are satisfied. Therefore, the Threshold logic equation OR(A,B)=$sgn\{ w_A A+w_B B- \theta\}$ for an OR gate is given in equation (12) and its corresponding threshold logic gate is drawn in Fig. 7(a)

$OR(A,B) = sgn\{A + B - 0.5\}$ ..... (12)

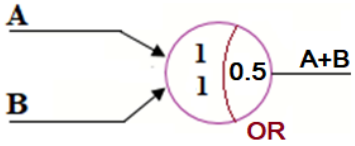
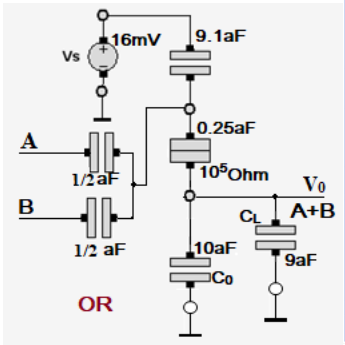


Fig. 7(a) Threshold logic OR gate

For implementing the OR gate we will use the parameters $C_1^p = C_2^p$=0.5aF, $C_b = 9.1aF, C_L = 9aF, C_0 = 10aF$, $R_j = 10^5\ \Omega, V_s = 16mV$ and accordingly after running the simulator, the output we get is given in Fig. 7(c).



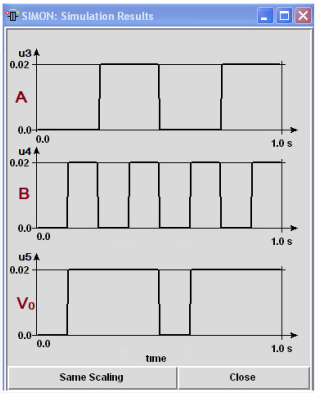


Fig. 7(b) OR gate Fig. 7(c) simulation result of OR gate

6. Inverter

An inverter that is capable of inverting its own input is shown in Fig. 8(a) [1-4, 8-10]. It is made up of 9 elements: 4 Tunnel junctions, 5 true capacitors; one

supply voltage V_s=16mV, one input terminal and one output terminal. The inverter is constructed with two single electron transistors (SETs) which are connected in series. Upper two Tunnel junctions (J_1, J_2) and two true capacitors (C_g, C_b) belong to SET1. Similarly, The lower two Tunnel junctions (J_3, J_4) and two true capacitors (C_g, C_b) belong to SET2. Two inputs terminals (both V_{in}) are directly coupled to the small islands (small circles) of the SET1 as well as SET2 [1, 2, 8-10] through their respective capacitances C_g and C_g respectively. The island sizes of each SET are of size close to 10 nm diameter of gold and their respective capacitor values will be less than 10aF. The output terminal V_0 is directly connected to the common channel in between SET_1 and SET_2. The output terminal must be grounded through a load capacitance C_L to suppress charging effects.

When an inverter is used we should choose the parameter values as: V_{g1}=0, V_{g2}=0.1$\times\frac{q_e}{C}$, $C_L = 9C$, $C_j = \frac{1}{10}C$, $5C_j = \frac{1}{2}C$, $C_g = \frac{1}{2}C$, $C_b = \frac{17}{4}C$ R_j=50KΩ . For the case of simulation, we take the value of C =1aF. The simulation set and its simulation result are provided in Fig. 8(a) and 8(b) respectively.

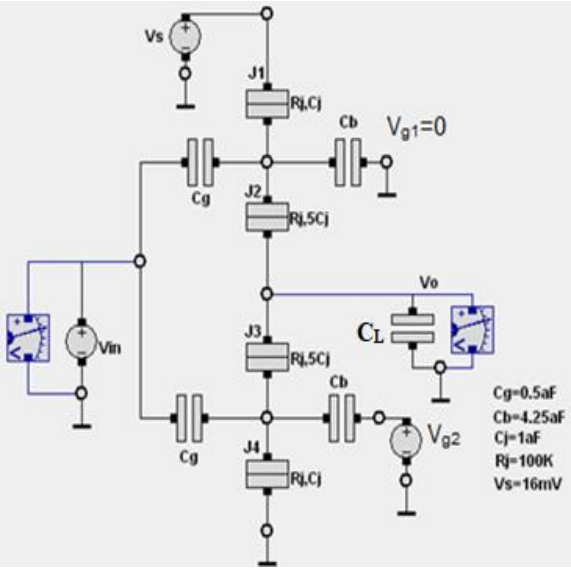


Fig. 8(a) Inverter based on parameters

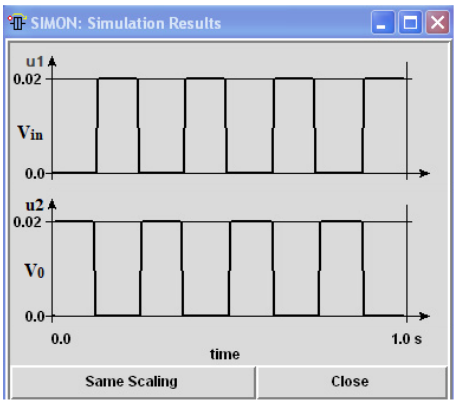


Fig.8 (b) Result after simulation of the inverter

We maintain the voltages $V_{g1} = 0$ *and* V_{g2} =16mV all time. The operation of the inverter can be marked as: - the output V_0 value will be 1 (logic high) provided that the input voltage V_{in} is low and that will be 0 (logic low) whenever the input voltage V_{in} is high. SET_1 (upper part in fig. 8(a)) will be in conduction mode and the SET_2 is in Coulomb blockade [1-5, 12] if the input signal V_{in} is connected to low (0). This causes the output voltage V_0 to connect to V_b , as a result the output voltage becomes high. When the input signal is high (logic 1), then the charges which is accumulated on each of the islands (for two SETs) are transferred by a fraction of an electron charge. By turns, SET_1 turns up in Coulomb blockade mode and the SET_2 turns up in conducting mode. In these circumstances, we call that the output V_0 changes the condition from high to low (logic 0).

For high and low inputs, we use the logic inputs "0" =0 Volts and logic "1"=0.1$\times\frac{q_e}{C}$.

For simulation purpose, we assume 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.

7. Exclusive OR (XOR)

An Exclusive-OR gate expression of two variables A and is expressed as $XOR(A,B) = \bar{A}B + A\bar{B}$ which is not linearly separable, because if any variable X and its complement $\bar{X}$ exist in a Boolean expression, then

this expression is not linearly separable. Hence $XOR(A,B)$ is not linearly separable, as a result it can't be represented by a Linear Threshold Gate (LTG).

When we want to express $XOR(A,B) = \bar{A}B + A\bar{B}$ with a threshold logic gate, first we express P = $(A\bar{B})$ with the help of threshold gate-based equation as given in equation (6). We can write as-

P = sgn {A+$\bar{B}$ − 2}..... (13)

We know $\bar{B}$ + B =1 or $\bar{B}$= − B + 1, so equation (13) will be

P = sgn { $\bar{B}$ −B− (1) } ..... (14)

Now the Boolean expression $XOR(A,B) = \bar{A}B + A\bar{B}$ can be written as

Y= $XOR(A,B)$ = P + $\bar{A}B$ ..... (15)

For finding out the threshold gate logic of equation (15), we draw the truth Table-4. There are four inequalities in 5th column in this table. Note that in equation 7, there are two variables A and B, though it seems that P is a variable in equation (15), it would not be correct. Only two variables A and B are the fact and P is depending on A and B. We can write the TLG expression as

$Y = sgn\{ W_1A + W_2B + W_3P - (\theta)\}$..... (16)

Table-4

A	B	P	Y	θ
0	0	0	0	0<θ
0	1	0	1	$W_2 \geq \theta$
1	0	1	1	$W_1 + W_3 \geq \theta$
1	1	0	0	$W_1 + W_2 < \theta$

After solving the inequalities in the 5th column of Table-4, we obtain a solution set { W_1 , W_2 , W_3 ; θ} = {-1, 1, 2 : 1}. Hence the Threshold equation for the Y is

Y = sgn { $-A + B + 2P - (1)$}..... (17)

By using the TLG expression in equation (17) we will be able to draw the threshold logic circuit of depth 2 as given in Fig.9a.

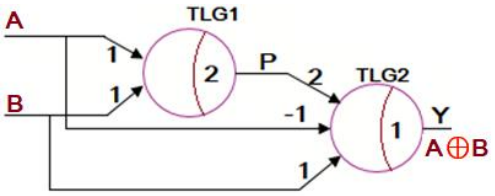


Fig. 9(a) TLG of $XOR(A,B) = \bar{A}B + A\bar{B}$

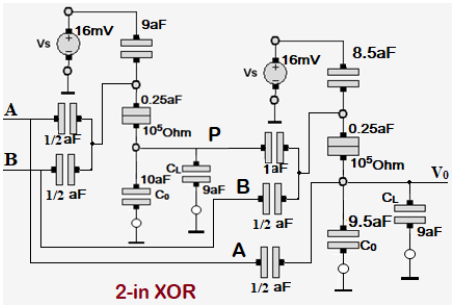


Fig. 9(b) Set of A⊕B

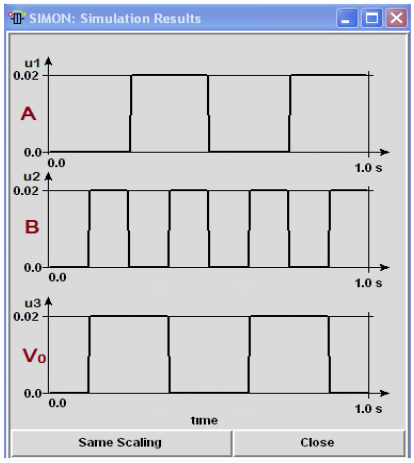


Fig. 9(c) Simulation result of A⊕B

To check the linearly separable condition of equation (15) Y= $XOR(A,B)$ = P + $\bar{A}B$, we think of the three variables A, B and P. In this expression, no variable is existing both the forms of variable and its complement simultaneously, so the expression is linearly separable. A truth table of P + $\bar{A}B$ is drawn in Table-5. When the logical value of the expression is 1 we represent it by a small red circle in Fig. 10 and that of the expression is 0 we represent it by a small colorless circle. It is transparent from the Fig. 10, there are two kinds of circles-one for Red and the other for colorless circles. These two of circles can be segregated by a plane. So the expression P + $\bar{A}B$ is

linearly separable and we are able to draw a linearly threshold logic gate with depth two and it is shown in Fig. 9(a). Truth table of $(P + \bar{A}B)$ is given in Table-5.

Table-5

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

8. E-cube Routing on a 4-dimensional hypercube

In section 2, we have discussed about the design of an E-cube Routing on a 4-dimensional hypercube and block diagram of it is provided there. Now our intention is to draw same thing by using the TLGs. The prominent component required for it is an Exclusive-OR gate. By arranging the total number of twenty LTG- based XOR in a pattern shown in the Fig. 11(a), we have been able to construct an "E-cube routing on a 4-dimensional hypercube".

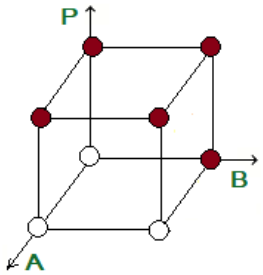


Fig. 10 Solution space Separation

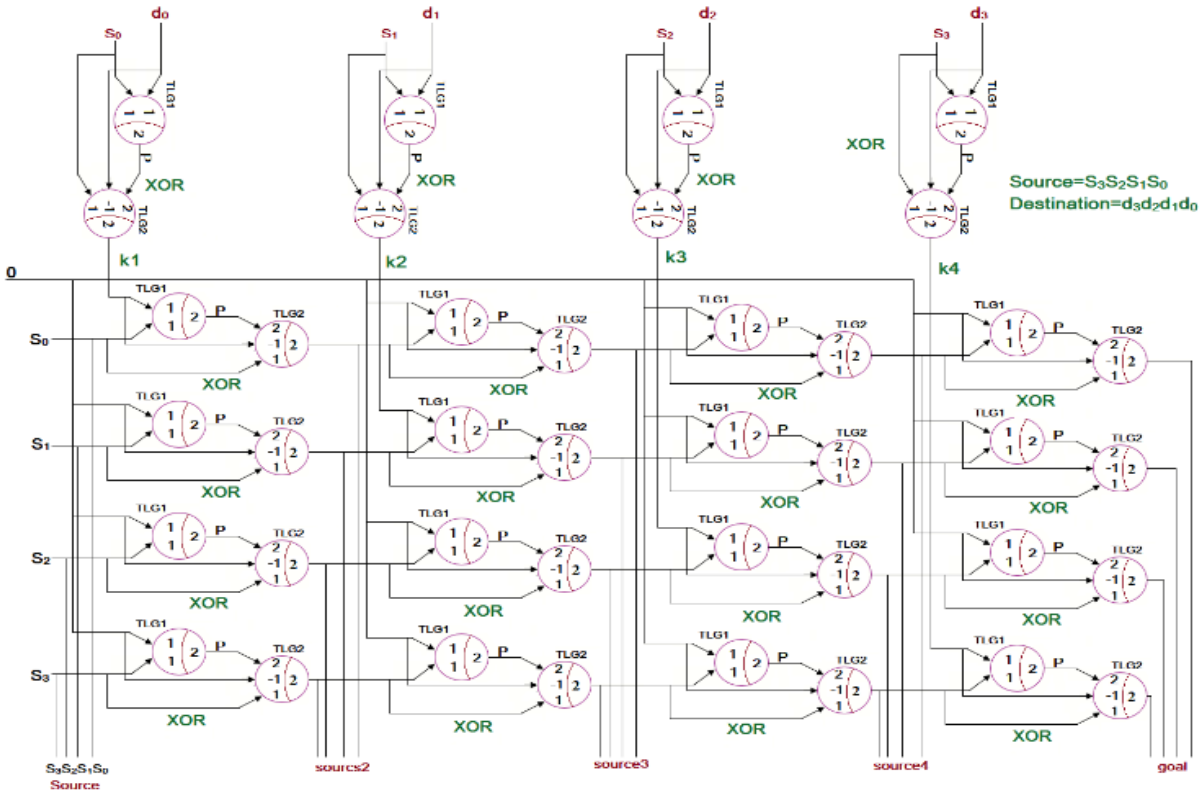


Fig. 11(a) Threshold Logic Gate Based routing path measurement of an E-cube routing on a 4-dimensional hypercube

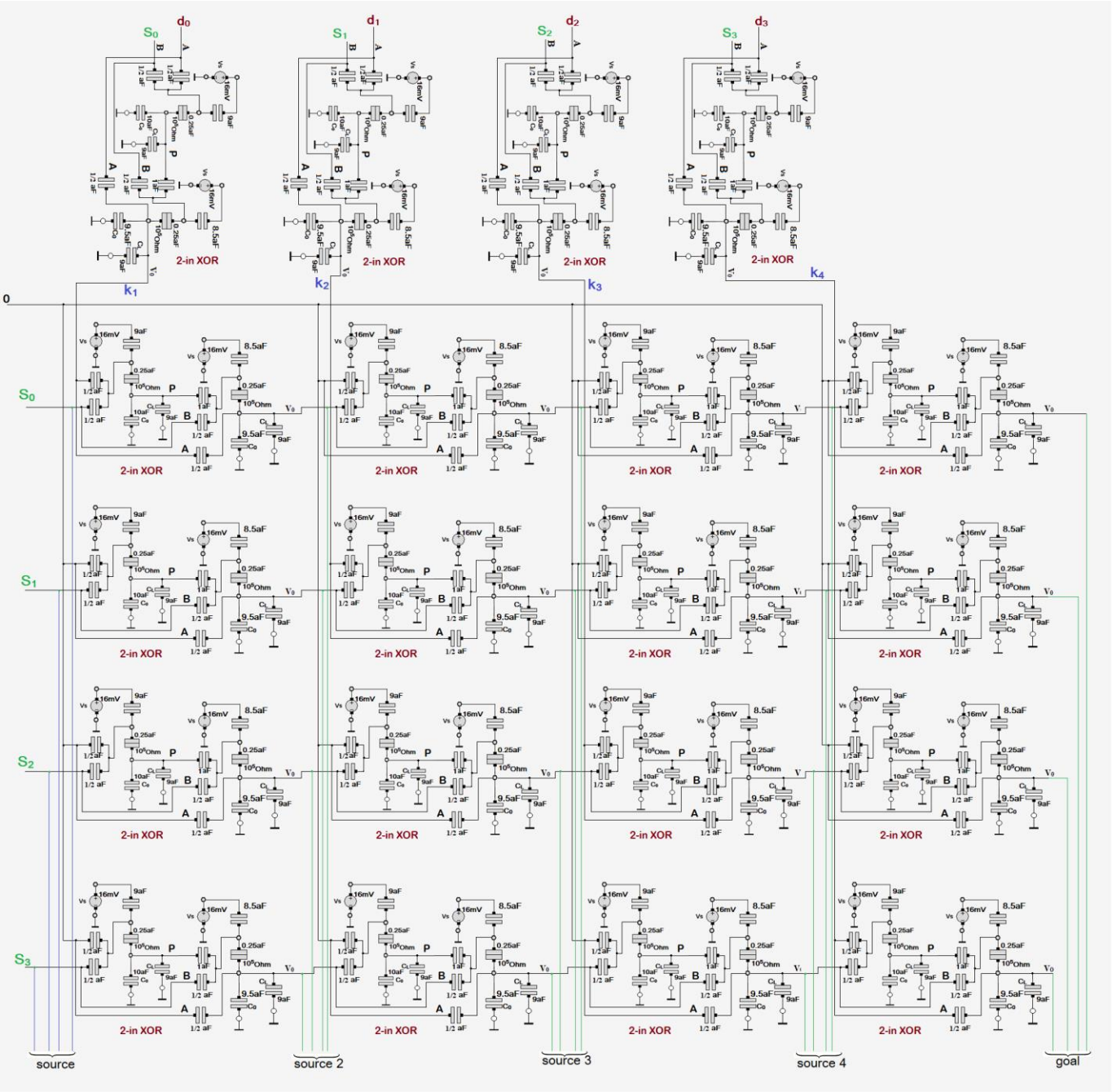


Fig. 11 (b) Simulation set for determining the source to goal routing path nodes

SIMULATION SET OF E-cube routing on a 4-dimensional hypercube

For the purpose of implementing Simulation set for determining the source to goal routing path nodes, the parameter values of all component required will be measured. We have measured them and they are given below:

For P=A$\bar{B}$: logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_1^n = = \frac{1}{2}C$ =0.5aF, $C_b = 9.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, . V_b =V_s=16mV.

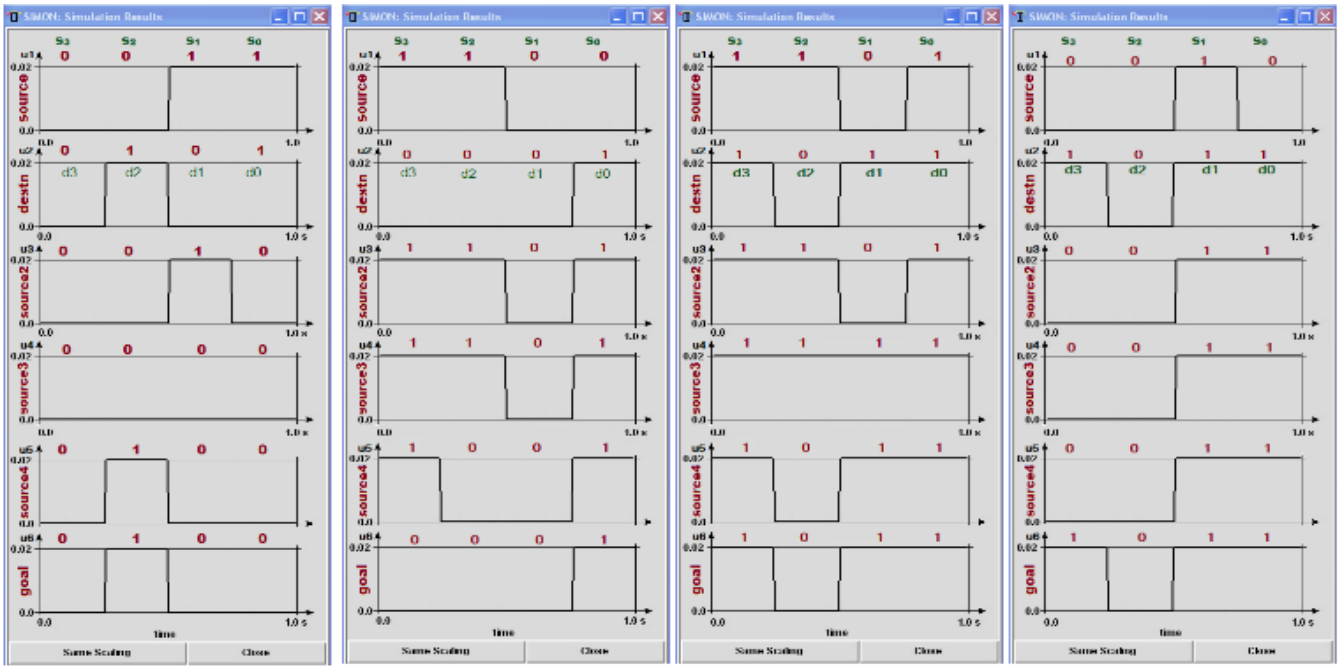


Fig. 11(c) Some simulation results of Fig. 11(b)

For Y= XOR(A, B) = P + $\bar{A}$B: 0"=0V, logic "1" = 16mV, C=1aF, $C_1^P = 0.5C_2^P = C_1^n = \frac{1}{2}C$=0.5aF, $C_b = 8.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, . V_b =V_s=16mV.

After placing these values in due positions, we have implemented the E-cube routing circuit which is given in Fig. 11(b).

OPERATION: At the time of execution, source and destination address or code values in binary are given as input in the input terminals, where source (S) =$S_3S_2S_1S_0$ and destination (d) = $d_3d_2d_1d_0$. First, we obtain a set of data $k_4k_3k_2k_1$. These data values i.e., k_1, k_2, k_3 and k_4 are used sequentially or one after another starting from k_1. The next operations are performed as given in short in Table-6.

For example, if we take source = $s_3s_2s_1s_0$=1100 and destination $d_3d_2d_1d_0$ =0010 then the executed values we
obtained from the circuit given in Fig. 11(b) are: $k_4k_3k_2k_1$=1110, 2nd source= 1100, 3rd source= 1110, 4th source =1010, 5th source = 0010 and goal = 0010. The goal address is the same as the destination.

Table-6

source =$s_3s_2s_1s_0$ destn = $d_3d_2d_1d_0$	2nd source	3rd source
$K_1 \leftarrow s0 \oplus d0$ $K_2 \leftarrow s1 \oplus d1$ $K_3 \leftarrow s2 \oplus d2$ $K_4 \leftarrow s3 \oplus d3$	$s0 \leftarrow s0 \oplus k1$ $s1 \leftarrow s1$ $s2 \leftarrow s2$ $s3 \leftarrow s3$	$s0 \leftarrow s0$ $s1 \leftarrow s1 \oplus k2$ $s2 \leftarrow s2$ $s3 \leftarrow s3$
4th source	**5th source**	**Goal**
$s0 \leftarrow s0$ $s1 \leftarrow s1$ $s2 \leftarrow s2 \oplus k3$ $s3 \leftarrow s3$	$s0 \leftarrow s0$ $s1 \leftarrow s1$ $s2 \leftarrow s2$ $s3 \leftarrow s3 \oplus k4$	Goal=5th source

9. Delay, Switching energy and Fastness

When a complex circuit contructed, we are to keep in mind that some inevitable components which are combined together in a proper way to obtained the circuit. In our present work, we are concentrated in an "E-cube routing on a 4-dimensional hypercube" wherein we need some inevitable components like 2-input AND, 2-input OR and 2-input XOR gate. For the measuring of the efficiency of any electronic circuit weather it may be conventional or Nano- or Pico-, some factors are to be considered properly like- (i) time required for an execution, (ii) number of components, (ii) speed, (iv) power consumption, (v) costs of materials, (vi) fan-outs, (vii) controlling

charge, (viii) atmospheric temperature etc. Ditterent parameter values of different TLG based gates are listed in Table-7.

Table-7

Sl. No.	Expressions used & Name of Expression	parameters
1	AND($A\bar{B}$)=A $\bar{B}$ 2-input AND gate	Logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_1^n = = \frac{1}{2}C$ =0.5aF, $C_b = 9.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, . V_b =V_s=16mV
2	AND(AB)= A.B 2-input AND gate	Logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_2^p = = \frac{1}{2}C$ =0.5aF, $C_b = 9aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 10aF$, R_j=10^5 Ω, . V_b =V_s=16mV
3	OR(AB)=A.+B 2-input OR gate	Logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_2^p = = \frac{1}{2}C$ =0.5aF, $C_b = 9.1aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 10aF$, R_j=10^5 Ω, . V_b =V_s=16mV
4	Y= XOR(A, B) = P + $\bar{A}$B	**For P:** logic input "0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = C_1^n = = \frac{1}{2}C$ =0.5aF, $C_b = 9.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, . V_b =V_s=16mV **For Y:** 0"=0V, logic "1" = 16mV, C=1aF, $C_1^p = 0.5C_2^p = C_1^n = \frac{1}{2}C$=0.5aF, $C_b = 8.5aF$, $C_j = 0.25aF$, $C_L = 9aF$, $C_0 = 9.5aF$, R_j=10^5 Ω, . V_b =V_s=16mV
5	$Y \leftarrow \bar{Y}$ Inverter	For the inverter, the parameter values chosen are: V_{g1}=0, V_{g2}=0.1×$\frac{q_e}{C}$, $C_L = 9C$, $C_j = \frac{1}{10}C$, $5C_j = \frac{1}{2}C$, $C_g = \frac{1}{2}C$, $C_b = \frac{17}{4}C$, R_j =50KΩ. For the sake of simulating, the value of C =1aF is fixed.

For calculating the processing/switching delays in the threshold logic circuit, we must think of the critical voltage V_c and the tunnel junction capacitance C_j in the multi-input threshold logic gate concerned. The processing delay for that circuit is find out by using the logarithmic expression of probable error given by the approach as written below [3-7, 14-15].

Delay = $-(e|\ln(P_{error})|R_t) / (|V_j| - V_c)$ ..... (18)

, where, R_t = the internal resistance of the tunnel junction, V_c = the critical voltage and V_j is being the tunnel junction voltage.

Execution is done when an electron passes through the tunnel junction barrier. When the tunnel junction voltage V_j being a fraction of the bias voltage is being greater than or equal to the internal critical voltage V_c of the tunnel junction i.e., when $|V_j| \geq V_c$, then a tunneling event occurs.

If this conditional expression $|V_j| \geq V_c$ is satisfied by a 2-input AND gate, junction voltage of the 2-input AND gate becomes V_j =10.80, the internal critical voltage V_c of the tunnel junction is found to be 11.58mV. Given that the parameter values of tunnel resistance $R_t = 10^5\Omega$ and the probability of error $P_{error} = 10^{-12}$. Putting them in equation (18) we can calculate the gate delay = 0.062|ln ($\boldsymbol{P_{error}}$)| = 1.71ns. In a similar manner we can calculate the delays of distinct circuits and they are listed in Table-8 and 9.

While tunneling, an electron passes over the junction barrier, by changing the total energy present in the circuit. The energy changes before and after a tunneling event is measured by the equation (19) given below.

$$\Delta E = E_{before\ tunnel} - E_{after\ tunnel}$$
$$= -e(V_c - |V_j|) \quad \text{..... (19)}$$

This is considered as the switching/tunneling energy $-e(V_c - |V_j|)$ that is consumed for a tunnel event in the tunneling circuit. We listed the energy consumptions for distinct and separate TLG circuits in Table-8.

Table-8

Gate/Device	n-elements	Delay (ns)	Switching Energy meV
2-in AND	6	$0.062\|\ln(P_{error})\|$	10.80
2-in NOR	6	$0.060\|\ln(P_{error})\|$	10.70
2-in OR	6	$0.062\|\ln(P_{error})\|$	10.80
2-in NAND	6	$0.062\|\ln(P_{error})\|$	10.80
inverter	9	$0.022\|\ln(P_{error})\|$	10.40
3-in AND	7	$0.072\|\ln(P_{error})\|$	11.58
3-in NAND	7	$0.072\|\ln(P_{error})\|$	11.58
2-in XOR	13	$0.080\|\ln(P_{error})\|$	21.20
3-in OR	7	$0.073\|\ln(P_{error})\|$	11.58
3-in NOR	7	$0.072\|\ln(P_{error})\|$	11.57
4-in OR	8	$0.076\|\ln(P_{error})\|$	14.22
8-in OR	12	$0.077\|\ln(P_{error})\|$	21.4
9-in OR	13	$0.079\|\ln(P_{error})\|$	23.18
JK flip-flop	19	$0.197\|\ln(P_{error})\|$	33.88
E-cube routing on a 4-dim hypercube	260	$0.400\|\ln(P_{error})\|$	424.00

Two distinct curves regarding Delay vs. Error Probability in TLGs and the switching delay vs. the unit capacitance C (=1aF) have been traced in Fig. 12(a) and Fog. 12(b) respectively.

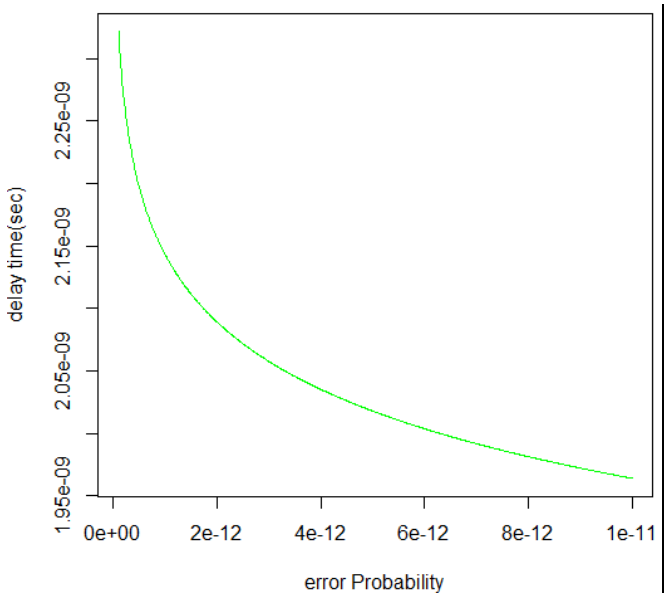


Fig. 12(a) Delay vs. Error Probability in TLGs

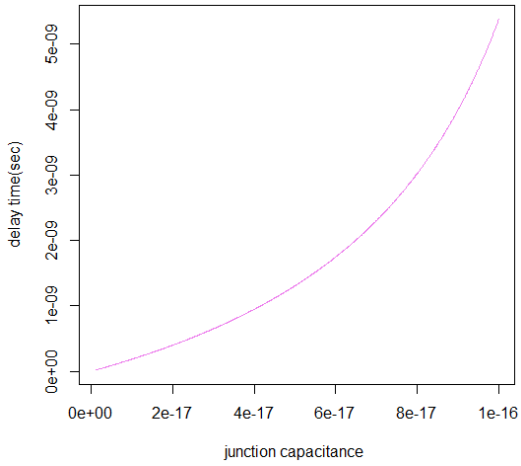


Fig. 12(b) Delay Vs. capacitance in circuits of TLGs.

The number of prime elements like capacitors, resistors are needed for constructing a small elements like logic gate or circuit are counted. We have kept track of the parameter values essential from this work in relation to small element numbers, delays or late, and switching power. All of them are tabulated in Table-8. Power consumption (meV) for each circuit vs. element numbers is shown in Fig. 13 below. The processing delays for single electron Transistor (SET) and TLG based circuits in reference to the various elements are depicted by using the bar diagram side by side in Fig. 14.

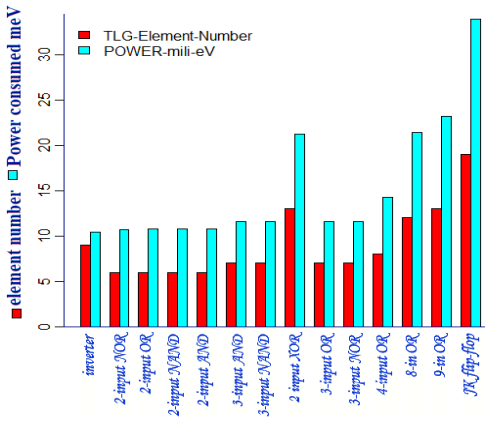


Fig. 13 Comparison of small Elements and switching energy

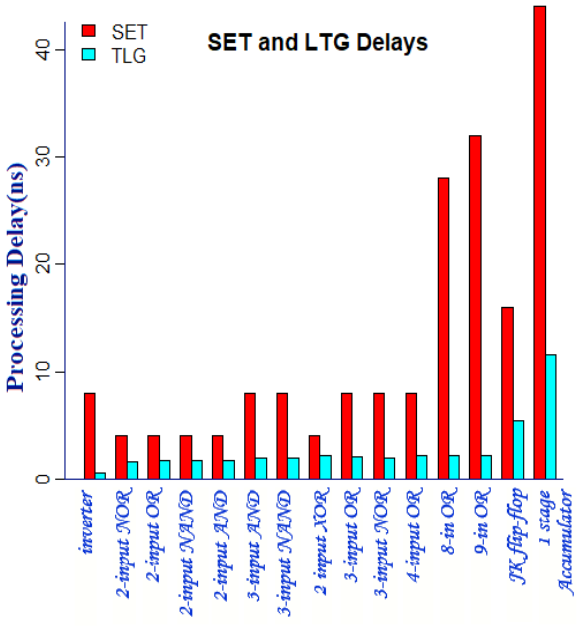


Fig. 14 Delays of SET/TLG Vs. gates

The processing delays for distinct TLG based circuits must not be similar, for instance, in a 2-input XOR gate the latency is 2.21 for the 4-input OR gate the processing delay is 2.15ns, and in an E-cube routing on a 4-dimensional hypercube the delay is 11.05ns.

It is given that the value of P_{error} is $\frac{1}{10^{12}}$, the time after which the first output in the present situation from 2nd source (s3s2s1s0) will fan out is $2.21 \times 2 = 4.42ns$. Similarly for 3rd source (s3s2s1s0) fan-out time=$2.21 \times 3 = 6.63ns$, for 4th source (s3s2s1s0) fan-out time=$2.21 \times 4 = 8.84ns$ and for 5th source (s3s2s1s0) fan-out time=$2.21 \times 3 = 11.05ns$ Hence, we are to maintain the data that are providing at the inputs should be at least after 11.05ns. In this condition, frequency of fan-out for the $E-cube\ routing\ on\ a\ 4-dimensional\ hypercube$ circuit will be $1/11.05\text{ns} = 90.49$ MHz.

We are intended in bringing to light the circuit delays in connection with CMOS, SET-based and LTG-based. We have detected the processing delays as 12ns [5-11, 17-18] each for the case of a CMOS logic gate like AND, NAND, NOR and XOR. The tunneling time delay in the case of a single electron transistor (SET) [13-15] is approximately 4ns [4-10, 17-18]. The speed-up of LTG-based delays in connection to the corresponding delays of SET-based delays are listed in Table-9 and the three curves of (i) SET-based delay, (ii) LTG-based delay and (iii) Speed-up times (ratio) versus the different logic gates / circuits are depicted in Fig.15.

10. Switching delays of SET and LTG

Given that the probability of error is $\frac{1}{10^{12}}$, the processing delays for the different circuits are provided in Table-9. From this table, we can easily realize that the LTG based circuit is at least 2 times faster than the SET based circuit. In view of CMOS, the LTG based circuit is 8 times faster than (it).

Table 9

Gate/Device	SET-based delay ns	LTG-based delay ns	Speed-up (times)
2-input NOR	4	1.65	2.42
2-input OR	4	1.71	2.33
2-input NAND	4	1.71	2.33
2-input AND	4	1.71	2.33
inverter	8	0.60	13.33
3-input AND	8	1.98	4.04
3-input NAND	8	1.98	4.04
2-input XOR	4	2.21	1.81
3-input OR	8	2.01	3.98
3-input NOR	8	1.99	4.02
4-input OR	8	2.15	3.72
8-in OR	28	2.12	13.2
9-in OR	32	2.18	14.67
JK flip-flop	16	5.44	2.94
E-cube routing on a 4-dim hypercube	44	11.05	3.98

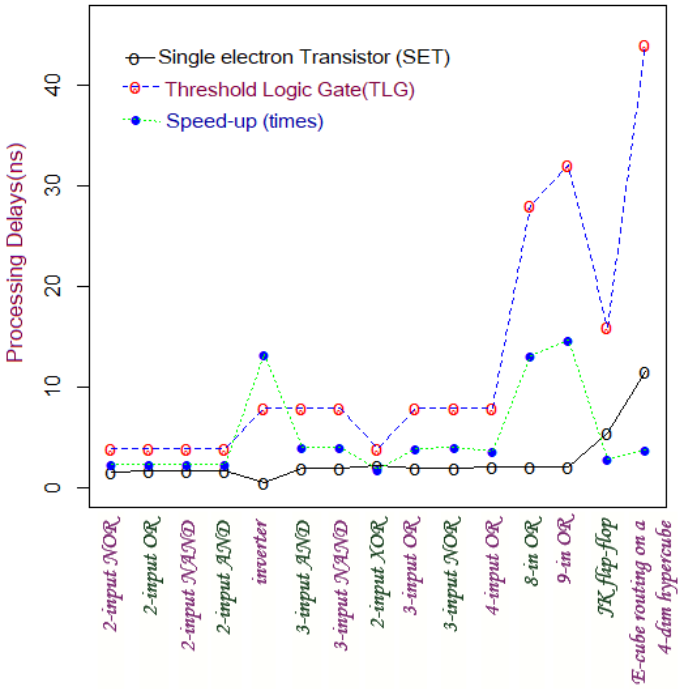


Fig. 15 Delays of SET- and TLG- based and their ratio

11. CONCLUSION

In a network, packets need to be travelled without deadlock. When we want to avoid deadlocking, a proper algorithm to be followed for packet routing so as to move through the busses without halt. If the deterministic routing algorithm is used, then the routing path from source to destination is strictly determined by the source point node address and goal/destination point address without deadlocking. To embody the deterministic routing algorithm, we can make a circuit called "E-cube Routing on a 4-dimensional hypercube" based on an emerging nanotechnology— on the principle of electron-tunneling. On the basis of the electron-tunneling, a multiple input threshold logic gate is introduced. Relying upon this multiple input threshold logic gate, we have been able to create threshold logic gates, combinational and sequential logic circuits. As the size of tunnel junction is very small (diameter 10nm), the concentration density of the TLG based circuits will be be high. The power consumed for the E-cube Routing on a 4-dimensional hypercube is 424.00meV $=424.00\times10^{-3}\times1.602\times10^{-19}=679.24\times10^{-24}$ Joule which is, really, very low amount. From the point of view of the result of the processing delay, it is clear that the TLG-based circuit is at least 2-times faster than the SET based circuit and 8-times faster in comparison with the CMOS based circuit. In our observation, we have decided that after every 11.05ns one desired output is fanned out from output terminal of the "E-cube Routing on a 4-dimensional hypercube". Almost all the circuits drawn are verified by the simulator-SIMON and realized that the simulated results are squared with the theoretical aspects. The reality is that atmospheric temperature for the case of an electron tunneling based linear threshold logic technology is to maintain at 0K which is really very difficult.

12. REFERENCES

[1] H. Sullivan and T.R. Bashkow,'A Large Scale Homogeneous, Fully Distributed Parallel Machine," Proc. 4th Symp. Computer Arch., vol.5 pp105-125. Mar, 1977

[2] Chin-Heng Liu, Chia-Chun Lin *et al* , "Threshold Function Identification by Redundancy Removal and Comprehensive Weight Assignments" IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 38, NO. 12, DECEMBER 2019

[3] Anup Kumar Biswas, "Integrated-Circuit Random Access Memory based on an emerging Technology—electron tunneling through Tunnel Junction" , International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Vol. 8

(4) pp.409-424 July-August-2021, Print ISSN: 2395-1990 | Online ISSN : 2394-4099

[4] Anup Kumar Biswas, "An Accumulator using Electron Tunneling Through Tunnel Junction" International Journal of Scientific Research in Science, Engineering and Technology Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com) Volume 8, Issue 5 Page Number : 163-183, September-October-2021

[5] A.K. Biswas, "Measuring of an unknown voltage by using single electron transistor based voltmeter" , Semiconductor Physics, Quantum Electronics & Optoelectronics(SPQEO), 24 (3), P. 277-287 (2021) August 2021; ISSN 1605-6582 (On-line)

[6] Anup Kumar Biswas, "A High-Speed Bidirectional Register with Parallel Loading using single electron Threshold Logic Technology" ,International Journal of Scientific Research in Science, Engineering and Technology (**IJSRSET)**, Vol. 8 (4) : pp394-408; July-August-2021, Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)

[7] Anup Kumar Biswas, "Integrated-Circuit Random Access Memory based on an emerging Technology—electron tunneling through Tunnel Junction" , International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET**)**, Vol. 8 (4) pp.409-424 July-August-2021, Print ISSN: 2395-1990 | Online ISSN : 2394-4099

[8] Anup Kumar Biswas, "A High-Speed Bidirectional Register with Parallel Loading using single electron Threshold Logic Technology", International Journal of Scientific Research in Science, Engineering and Technology Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com),July-August-2021, 8 (4) : pp394-408

[9] 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

[10] 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,

[11] 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

[12] 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,

[13] 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,

[14] 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

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

[16] Casper Lageweg, Student Member, IEEE, Sorin Cot¸ofanˇa, 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

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

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

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

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

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

[22] Kai Hwang & Naresh Jotwani, "Advanced Computer Architecture Parallelism, Scalability, Programmability" Second Edition, Mc Graw Hill.

BIOGRAPHY

ANUP KUMAR BISWAS, is an Assistant Professor in

Computer Science and Engineering, Kalyani Govt. Engineering College,West Bengal, India. 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

Cite this article as :

Anup Kumar Biswas, "Threshold Logic Technology based E-cube Routing on a 4-dimensional hypercube", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 6, pp. 07-24, November-December 2021. Available at doi: https://doi.org/10.32628/IJSRSET218547

Journal URL: https://ijsrset.com/IJSRSET218547