A Full Adder Implementation Along with AND and OR Gates Using SET Based Linear Threshold Gates

Dr. Anup Kumar Biswas

Department of Computer Science and Engineering, Kalyani Govt. Engineering College, Kalyani

DOI: https://doi.org/10.51584/IJRIAS.2026.11100103

Received: 24 October 2026; Accepted: 29 October 2026; Published: 16 November 2026

ABSTRACT

Modern electronic design increasingly demands cost-effective, ultra-low power consumption, high processing speeds, and maximum integration density—attributes crucial for today's engineering, business, and manufacturing sectors. This paper investigates single-electron tunneling (SET) devices from a logic design perspective, leveraging the tunnel junction's capability to precisely control individual electron transport. Specifically, we present the implementation of a high-performance Full Adder using SET-based threshold gates, alongside detailed implementations of fundamental AND and OR logic gates. To mitigate feedback effects common in passive SET networks, we introduce an augmented threshold gate architecture integrated with an active buffer. This buffered threshold gate configuration is utilized to construct the Full Adder circuit. The functional behavior of the Full Adder, AND gate, and OR gate is validated through rigorous circuit simulations. Finally, we analyze the critical trade-offs involving circuit delay, error probability and capacitance.

Keywords: Threshold, Tunneling, Coulomb blockade, processing delay, error probability

INTRODUCTION

Over the past few decades, the feature size of MOS-based circuits has scaled down dramatically. However, foundational research indicates that conventional transistors cannot be shrunk beyond certain physical limits dictated by their basic operating principles. To sustain this reduction in feature size, emerging successor technologies with superior scaling potential—such as Single-Electron Tunneling (SET) [1-7]—are being actively investigated. To date, most studies on SET-based logic circuits treat the tunnel junction as a simple switch, utilizing it to construct the SET equivalent of a standard MOS transistor. While this method allows existing CMOS layouts to be easily ported to SET technology, it introduces a significant drawback: the current flowing through an "open" transistor still relies on a large stream of individual electrons "dripping" through the tunnel junctions. This collective transport is inherently slower than moving a single electron through the same junction. Consequently, design strategies that merely mimic CMOS logic fail to leverage the full potential of SET technology.

Our research shifts this paradigm by designing SET logic circuits that exploit the junction's unique ability to control the transport of individual electrons. The core concept is to execute logic operations by actively guiding single electrons to or from the output node, modulated by the input signals. Specifically, this paper investigates the implementation of a Full Adder utilizing SET linear threshold gates [10-11] alongside standard AND and OR gates. The remainder of this paper is structured as follows. Section 3 introduces Multi-Input Threshold Logic Gates and the background theory of SET tunneling behavior. Sections 4 and 5 present the full adder threshold implementation and the active buffer, respectively. Sections 6 and 7 discuss Logic AND and OR gates. Section 8 analyzes delay, switching energy, and operating speed, and Section 9 provides concluding remarks.

LITERATURE REVIEW

Advancements in next-generation fields—such as advanced computing, aerospace communication, medicine,

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

smart agriculture, and multimedia—depend fundamentally on ultra-fast processing hardware. To accelerate these systems, conventional electronic components must transition toward ultra-fast architectures. Single-Electron Tunneling (SETun) circuits have emerged as a leading alternative to meet these stringent speed and performance requirements. Over recent decades, this domain has progressed from Binary Decision Diagrams (BDD) [8, 9, 19-20] to single-electron transistors (SETs) [2, 3, 7, 13-15, 21], and more recently to sophisticated threshold logic gates [1, 5-]. Experts in threshold logic and tunneling circuitry predict that SET-based hardware will deliver key microelectronic benefits, including microscopic footprints, lowered fabrication costs, minimal power usage, high integration density, reduced operational latency, and ultra-low noise levels. Capitalizing on these advantages, recent literature has successfully demonstrated the design and deployment of highly complex SET-based architectures.

CIRCUIT ARCHITECTURE AND MATHEMATICAL MODELING OF A MULTI-INPUT THRESHOLD LOGIC GATES (TLG)

A generic, multi-input single-output linear threshold gate (LTG) [3–8, 14–16] utilizes a single tunnel junction as its core architectural element in Fig. 1. Characterized by its capacitance C_j and resistance R_j, this junction is interfaced between nodes p and q. The circuit is driven by two distinct sets of multi-input combinational signals:

- **Upper-side inputs** (V_k^P)s**):** Coupled to node q through their respective input capacitances C_k^P.
- **Lower-side inputs** (V_l^n)s**:** Coupled to node p through their respective input capacitances C_k^n.

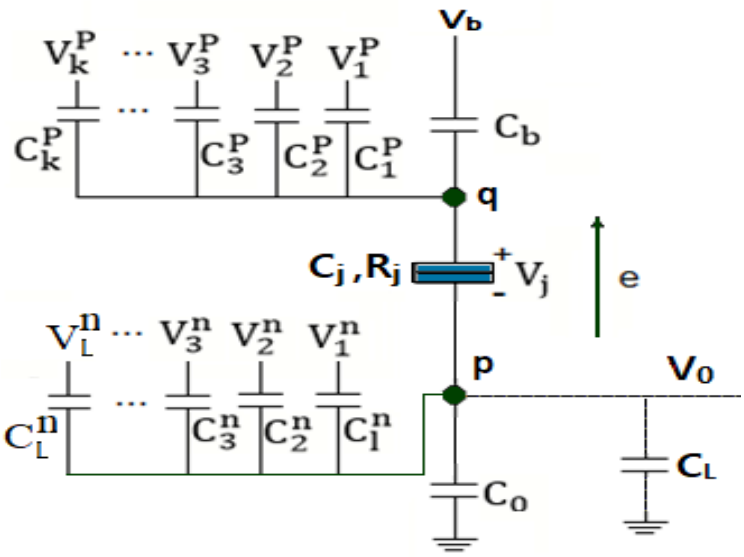


Fig. 1 multi-input one-output TLG

The functional operation of the multi-input single-output LTG is evaluated using a $signum$ function [1, 5,10-11]. The system behavior is governed by the following mathematical framework: The functional operation of the multi-input single-output (MISO) LTG is evaluated using a $signum$ function [4-7,10]. The system behavior is governed by the following mathematical framework:

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

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

Where $x_k$represents the n distinct inputs k=1, 2, 3…n, and w_k denotes their respective integer weights.

In practice, the gate functions as a digital comparator that evaluates a predefined threshold valueθ against the weighted sum of the inputs$\sum_{k=1}^{n} (w_k \times x_k)$.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

When this weighted sum equals or exceedsθ, i.e., $\sum_{k=1}^{n}(w_k \times x_k) \geq \theta$, the activation function h(x) becomes non-negative. This triggers a high logical state, transitioning the circuit to yield an output of "1" at the output terminal V_0. Conversely, if the weighted sum falls below θ the output remains at a logical "0".

The critical voltage $V_c$required to enable electron tunneling serves as the circuit's intrinsic threshold. If the voltage across the tunnel junction exceeds V_c, an electron tunnels through the junction in the direction of the arrow. This event drives the output high, representing a logical "1".

To configure the gate threshold to the desired valueθ, a biasing voltage V_b,is applied and weighted by a biasing capacitor C_b. The positive input signals, $V^p = \{V_1^p, V_2^p \ldots V_k^p\}$, are weighted by their corresponding capacitors, $C^p = \{C_1^p, C_2^p \ldots C_k^p\}$, and added to the voltage across the tunnel junction. Conversely, the negative input signals, $V^n = \{V_1^n, V_2^n \ldots V_l^n\}$, are weighted by their respective capacitors, $C^n = \{C_1^n, C_2^n \ldots C_l^n\}$, and subtracted from the junction voltage. Based on these configurations, the resulting threshold function for the circuit [10] is:

$$h(x) = C_\Sigma^n \sum_{k=1}^{r} C_{\mathrm{k}}^p V_\Sigma^p - C_\Sigma^n \sum_{k=1}^{8} C_{\mathrm{i}}^n V_{\mathrm{i}}^n - \theta \quad \cdots\cdots (3)$$

$$\theta = 0.5\mathrm{e} \times (C_\Sigma^n + C_\Sigma^p) - C_\Sigma^n \mathrm{C_b V_b} \quad \cdots\cdots (4)$$

Here, $C_\Sigma^p = \mathrm{C_b} + \sum_{k=1}^{r} C_{\mathrm{k}}^p$ and $C_\Sigma^p = \mathrm{C_0} + \sum_{i=1}^{s} C_{\mathrm{i}}^n$

Threshold gates with arbitrary positive and negative weighted inputs can be created using this approach, though practical circuits are restricted by the desired output voltage and the sizes of the input capacitors C^pandC^n. Furthermore, implementing these gates within a network structure triggers strong feedback effects that can cause erroneous network behavior. To resolve the feedback problems associated with interconnecting passive gates, an active buffer can be utilized. A simple active buffer design is introduced in the next section.

FULL ADDER THRESHOLD IMPKEMENTATION BASED ON CARRY LOOKAHEAD ADDER

This section presents a Full Adder (FA) architecture implemented using two Single-Electron Transistor (SET) threshold gates paired with active buffers [10-12]. The full adder computes the sum of two input bits, $\mathrm{A_i}$and $\mathrm{B_i}$, along with a carry-in bit $\mathrm{C_{i-1}}$. Itsubsequently generates a sum bit $\mathrm{S_i}$and a carry-out bit $\mathrm{C_i}$.By leveraging threshold logic gates, the full adder can be realized using only two gates structured across two logic levels [9-10]. In alignment with Equations (1) and (2), the operation is defined as follows:

To determine the threshold representation for both carry and sum, the carry operation is analyzed first. Using the carry logic expression from equation (5a), Table-1 presents the corresponding truth table along with the required threshold inequalities.

$$C_i(\mathrm{A_i B_i C_{i-1}}) = \mathrm{sgn}(w_A \mathrm{A_i} + w_B \mathrm{B_i} + w_C \mathrm{C_{i-1}} - \theta) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5a)$$

Table-1

$\mathrm{A_i}$	$\mathrm{B_i}$	$\mathrm{C_{i-1}}$	$C_i(\mathrm{A_i B_i C_{i-1}})$	θ	Eqn No.
0	0	0	0	$0 < \theta$	(i)
0	0	1	0	$w_C < \theta$	(ii)
0	1	0	0	$w_B < \theta$	(iii)
0	1	1	1	$w_B + w_C \geq \theta$	(iv)
1	0	0	0	$w_A < \theta$	(v)
1	0	1	1	$w_A + w_C \geq \theta$	(vi)
1	1	0	1	$w_A + w_B \geq \theta$	(vii)
1	1	1	1	$w_A + w_B + w_C \geq \theta$	(viii)

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

Based on Table 1, when the weights and threshold values are set as $\{w_A,w_B,w_C:\theta\}$ ={ 1, 1, 1 : 2}, all inequalities in the fifth column are satisfied. This yields the solution set {1, 1, 1: 2}, allowing the carry function to be expressed as:

$$C_i = sgn(\mathrm{A_i}+\mathrm{B_i}+\mathrm{C_{i-1}}-2) \quad \text{(5b)}$$

Similarly, to derive the threshold representation for the sum (\(S_{i}\)), we use the logic expression from equation (5a). The truth table for the carry output alongside its corresponding threshold inequalities is detailed in Table-1.

The threshold logic function for the sum is defined as:

$$S_i(\mathrm{A_iB_iC_{i-1}C_i}) = \mathrm{sgn}(w_A\mathrm{A_i}+w_B\mathrm{B_i}+ w_{C(i-1)}\mathrm{C_{i-1}} + w_{C(i)}\mathrm{C_{i-1}} - \theta) \quad \text{(6a)}$$

Given that the logic variables $\mathrm{A_i},\mathrm{B_i},\mathrm{C_{i-1}}$ and $\mathrm{C_i}$ take binary values of either 0 or 1, the 8 possible combinations of these inputs and their resulting threshold inequalities are mapped out in Table-2 as follows:

Table-2

$\mathrm{A_i}$	$\mathrm{B_i}$	$\mathrm{C_{i-1}}$	$C_i(\mathrm{A_iB_iC_{i-1}})$	S_i	θ	Eqn. No.
0	0	0	0	0	$0 < \theta$	(i)
0	0	1	0	1	$w_{C(i-1)} \geq \theta$	(ii)
0	1	0	0	1	$w_B \geq \theta$	(iii)
0	1	1	1	0	$w_B + w_{C(i-1)} < \theta$	(iv)
1	0	0	0	1	$w_A \geq \theta$	(v)
1	0	1	1	0	$w_B+w_{C(i-1)}+ w_{C(i)} < \theta$	(vi)
1	1	0	1	0	$w_A+w_B+ w_{C(i)} < \theta$	(vii)
1	1	1	1	1	$w_A+w_B+ w_{C(i-1)} + w_{C(i)} \geq \theta$	(viii)

By substituting the parameters $\{w_A,w_B,w_{C(i-1)}w_{C(I)}: \theta\}$ ={ 1, 1, 1,-2: 1}, all inequality constraints listed in the fifth column of Table 2 are simultaneously satisfied. This specific solution set yields the precise parameter configuration required to derive equation (6b). The resulting Full Adder circuit is displayed in Fig. 2.

$$\mathrm{S_i} = sgn(\mathrm{A_i}+\mathrm{B_i}+ \mathrm{C_{i-1}}- 2\mathrm{C_i} - 1) \quad \text{(6b)}$$

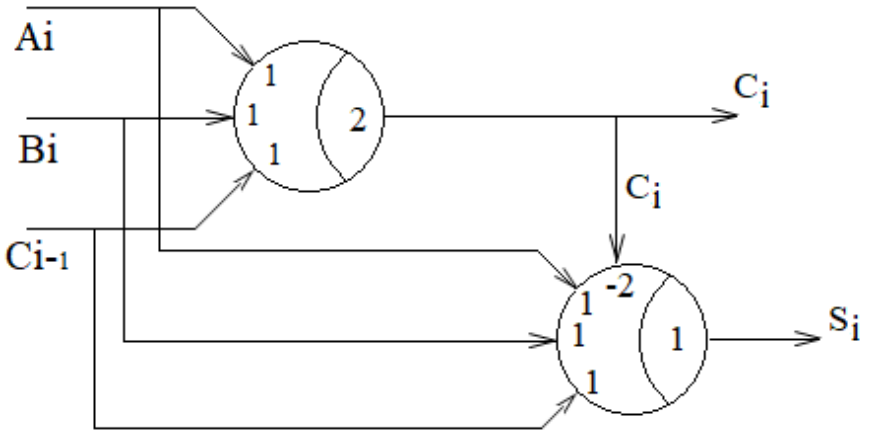


Fig. 2. Diagram of a Full Adder implemented in threshold gates.

In theoretical models, the threshold parameters θ are integers. Specifically, a threshold defined as $\theta = i$ (where $i \in \mathbb{Z}$) implies the operational range $i - 1 < \theta \leq i$. To maximize robustness against parameter variations, these thresholds are shifted to $\theta - \frac{1}{2}$.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

Implementing these threshold gates alongside an active buffer allows for partial sharing of the required computations. The design parameters and hardware choices are established as follows:

- **Logic Levels:** Logic '0' corresponds to 0V, and logic '1' corresponds to = 0.1 e/C Volt for both input and output signals.
- **Junction Capacitance:** The tunnel junction capacitor is chosen to be small, withC_j = 0.25C.
- **Input Weighting:** To implement the required input weights of +2 and +1 derived from Equations (5) and (6), small and identically sized capacitors C_p = C_n= 0.5C are selected for all inputs. Consequently, Equation (3) scales to C_Σ^p=2C_Σ^n.

The active buffer requires a high-level gate voltage (Fig. 5) of V_g'high' = 0.3e/C. For the threshold gate, a 'high' output voltage V_o(1e)—generated by the tunneling of a single electron through the junction—is defined by: (1e)=$\frac{eC_\Sigma^p}{C_\tau}$, where the total network capacitance C_τ is given by $C_\Sigma^p C_j$+$C_\Sigma^p C_\Sigma^n$+ $C_j C_\Sigma^n$To satisfy the buffer requirement of V_0(1e) =$\frac{0.30e}{C}$, the structural capacitances solve to $C_\Sigma^n = 2.96C$ and $C_\Sigma^p = 5.92$. Given the buffer's specific input capacitance of 0.66V, this configuration yields an effective junction value of $C_j = 2.3C$.

To determine the unit value for the thresholdθ , we evaluate the circuit parameters across Equations (3) and (4). An input value of 1 with a ' +1' weight yields $C_\Sigma^n C^p V^p$ = 0.148 under Equation (3). From Equation (4), a threshold of θ = 1.50 corresponds to $C_b V_b$= 1.425, whereas a threshold of θ = 0.50 corresponds to $C_b V_b$ = 1.475. The active buffer's bias voltageV_b (BUF) = 0.3e/C shifts the tunnel junction voltage V_j by in δV_j =− 0.045e/C. To compensate for this reduction, we must increase the threshold gate's bias voltage, V_b (TG). Because V_j (V_b(TG)) = $C_n C_b V_b$/C, increasing the $C_b V_b$product by 0.3 neutralizes the buffer's bias effect. We set V_b = 1.5e/C for both threshold gates. For the gate calculating the carry C_i, this configuration requires C_b = 1.15C. Finally, adding an extra capacitor C_g^p = 3.27 between node 'q' (Fig. 1) and ground ensures that C_Σ^p = 5.92C.

In summary, the threshold gates for the carry-out C_i and sum bit S_i feature the following circuit parameters:

Carry-out C_i Gate:C_b = 1.15C, C_p = 0.5C, C_g^p = 3.27, C_j = 0.25, C_l =2.3C, C_g = 1C, C_d= 1.5C, C_s = 0.5C, C_o = 10C, V_b (TG) = 1.5e/C, V_b (BUF)′ high′ = 0.3e/C, V_b(BUF)'reset′ = 033e/C.

Sum Bit S_i Gate:C_b = 1.18C, C_p = 3.24,C_n= 0.5C, C_l= 1.8C (all other parameters are equal to those for C_i).

Using these parameters, both gates were verified via simulation using SIMON[1, 4-7, 10], a dedicated single-electron device and circuit simulator. The simulation environment was configured with a capacitance C = 1aF (10^{-19} F), yielding a charging energy e/C = 160mV. Additional simulation parameters included a tunnel resistance R_t = $10^5\Omega$ and an operating temperature T = 0K. The simulation results for the carry gate, C_i and the sum gate S_i are presented in Figure 4 and Figure 3, respectively, confirming that both circuits compute the correct logical outputs. Finally, the carry and sum gates were integrated to form the complete Full Adder structure. In this combined configuration, the threshold gates must compensate not only for their own internal buffer bias voltages but also for the bias voltage introduced by the interlocking threshold gate. Consequently, the shift in the junction voltage δV_j for the sum gate S_i becomes −0.047e/C. To account for this, the circuit parameters were adjusted to C_b= 1.19C, C_g^p= 3.23. The final verification results for the integrated Full Adder and individual carry threshold simulations are reflected in Fig. 3 and Fig. 4.

To overcome feedback effects in passive SET networks, we first augmented the threshold gates with an active buffer. We then derived and presented the simulation results. Finally, we utilized these buffered threshold gates to construct a Full Adder circuit and verified its behavior through simulation.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

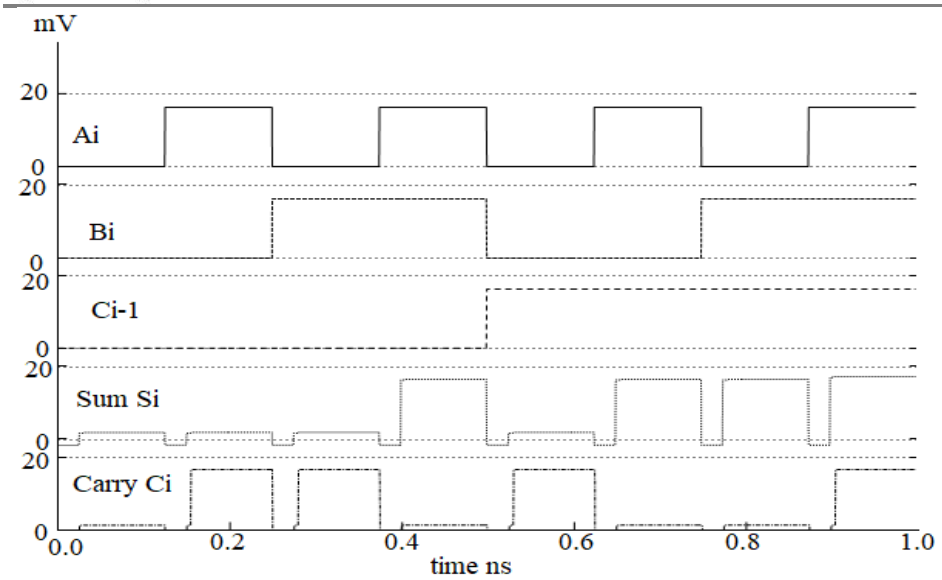


Fig. 3. Simulation curve for Full Adder implemented in threshold gates.

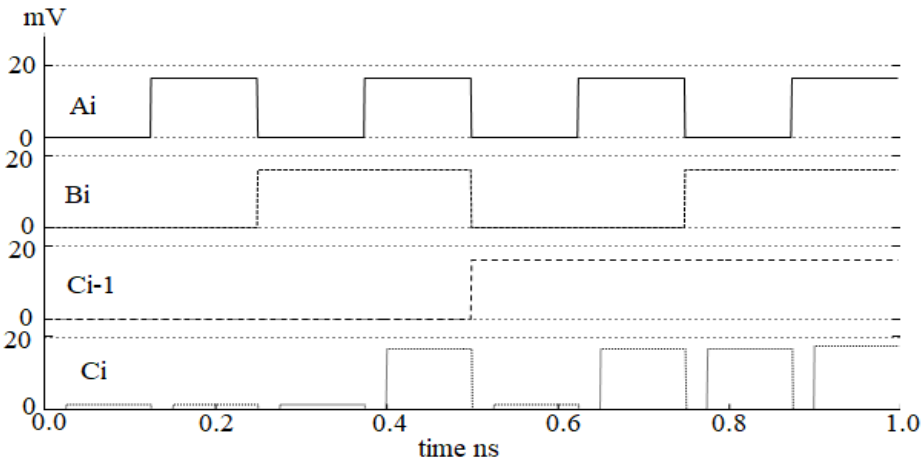


Fig.4 Simulation curve for Carry threshold

1. The Active SET Buffer

The circuit in Figure-5 operates as follows:

Biasing: Setting only V_b high maintains a net zero charge on nodes "a" and "b", keeping the output V_o at zero[10].

Charging: Toggling V_g to high drives V_o high by transporting electrons from node "a" through node "b" to the V_b source. The buffer's intrinsic memory locks the output in this high state, even if V_b and V_g return to zero.

Resetting: Applying a negative bias voltage (V_b reset) while V_g is zero replaces the electrons removed from node "a", resetting the circuit.

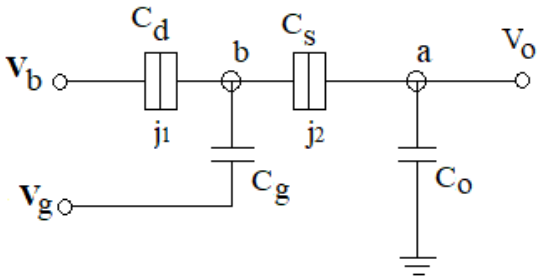


Fig. 5 An active SET buffer

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

2. 2-Input Threshold AND Gate

A 2-input threshold logic AND gate in eqn. (7a) can be designed by mapping the behavior of a standard AND gate truth table in Table-1. Governing system constraints are established by evaluating the weighted inputs, w_A and w_B, against the gate's threshold value, θ. Solving the four resulting inequalities regarding 4th column inTable-3yields the valid solution set w_B=1, w_A=1 and θ=2. Consequently, the threshold logic expression for the AND gate is defined in Equation (7b), with its schematic representation illustrated in Figure 6(a).

$F(A,B) = sgn(w_A A + w_B B - \theta)$ ……..(7a)

Table-3

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)

$AND(A,B) = sgn\{A + B - 2\}$ ..... (7b)

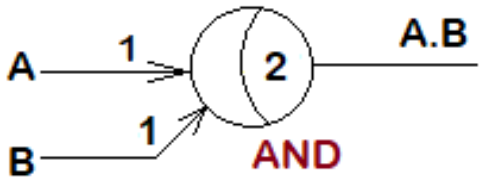


Fig. 6(a) Threshold logic AND gate

The implementation parameters for the proposed AND gate are set as follows:$C_1^n = C_2^n = 0.5\text{aF}, C_{b1} = C_{b2} = 4.25\text{aF}, C_{g1} = C_{g2} = 0.5aF, C_L = 9aF, C_0 = 8aF,$ and$R_j = 10^5\Omega$. The corresponding schematic configuration is illustrated in Fig. 6(b), while the resulting simulation trajectories are presented in Figs. 6(c) through (e).

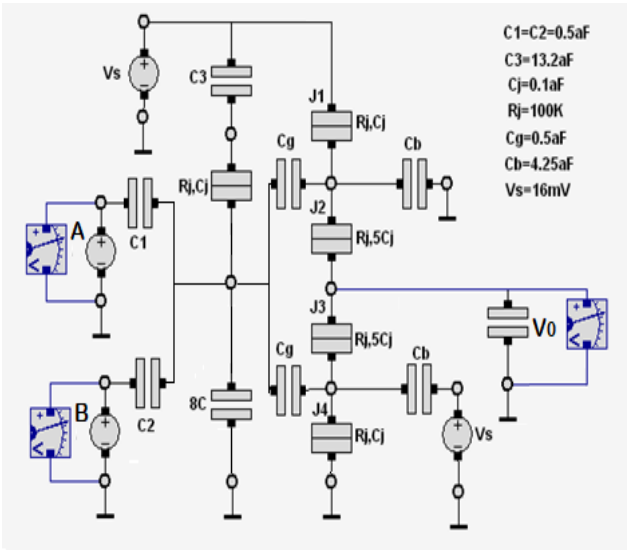
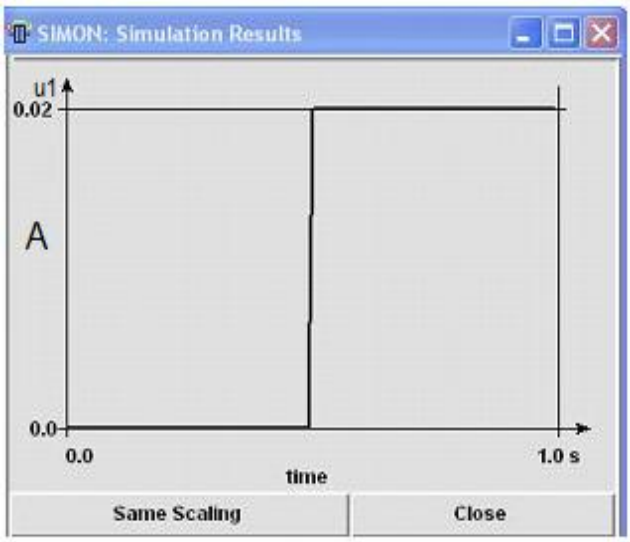
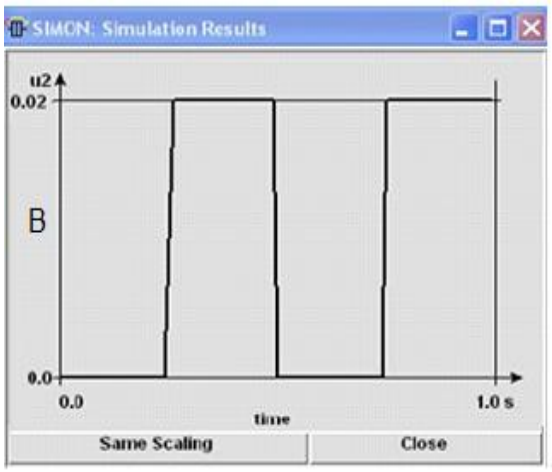


Fig.6 (b) AND Gate

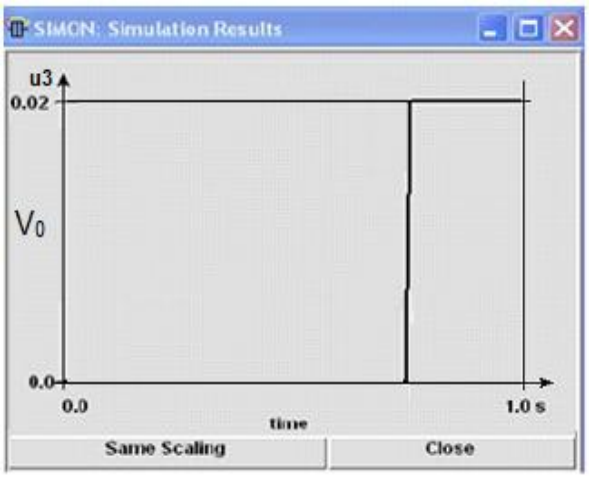


(c)

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026



(d)



(e)

Fig.6 (d), (e) simulation trajectories of 2-input AND gate

3. Threshold Logic Equation For 3-Input AND Gate

Assuming positive logic—where all weights $w_A, w_B, w_C$and the threshold must be positive—the threshold logic equation for a 3-input AND gate can be derived directly. By assigning the weights$w_A = 1, w_B = 1$ andw_C=1, and selecting a threshold of $\theta = 2.5$ (or any value in the range $2 < \theta \leq 3$, all conditional equations in the 5th column of Table-4 are satisfied. Consequently, the threshold logic equation is defined as:

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

$$= sgn\{A + B + C - 2.5\} \cdots\cdots (8)$$

Table-4 (Truth table of 3-in AND gate)

A	B	C	F(A,B,C)=ABC	θ
0	0	0	0	0<θ
0	0	1	0	$\theta > w_C$
0	1	0	0	$\theta > w_B$
0	1	1	0	$\theta > w_B + w_C$
1	0	0	0	$\theta > w_A$
1	0	1	0	$\theta > w_A + w_C$
1	1	0	0	$\theta > w_A + w_B$
1	1	1	1	$w_A + w_B + w_C \geq \theta$

Figures 7(a) and 7(b) illustrate the simulation setup and its corresponding input-output results.

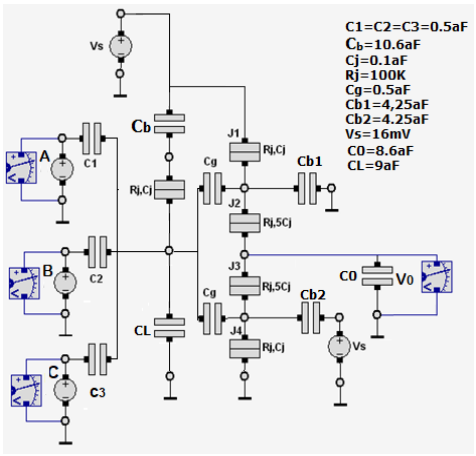


Fig. 7(a) set of 3-input AND gate

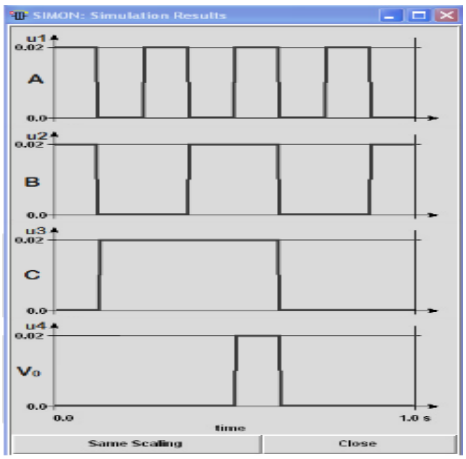


Fig. 7(b) result

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

4. Threshold Logic Equation For 2-Input OR Gate

The Boolean expression for a two-variable OR gate is **Y = A + B**, where the "+" symbol denotes the logical OR operation. To implement this function using threshold logic, we must first analyze its truth table. Table-5 outlines the standard truth table for the two-variable OR gate. By evaluating these input-output combinations, we can determine and compare the appropriate weights for the variables A and B relative to the gate's threshold or intrinsic valueθ. Analyzing these input-output pairs allows us to determine the variable weights w_A and w_B and compare them against the gate's threshold value θ.

Table-5

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	$w_B + w_A \geq \theta$

By assuming positive logic and setting both weights w_B=1, and w_A=1, we obtain the characteristic inequalities derived from the fourth column of Table 3.

$0<\theta$ ..... (9)

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

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

$w_B + w_A > \theta$ ..... (12)

Because condition (9) dictates a positive thresholdθ, choosing (θ= 0.5) satisfies inequalities (10), (11), and (12). This choice defines the threshold logic equation for the OR gate in equation (13), and Fig. 8(a) illustrates the resulting gate configuration.

$OR(A,B) = sgn\{A + B - 0.5\}$ ..... (13)

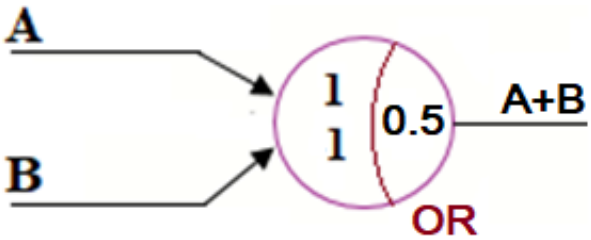


Fig. 8(a) Threshold logic OR gate

Figure 5(c) illustrates the finalized simulation response of the OR gate in Figure 5 (b), which was evaluated using the following specific circuit parameters:

Gate Capacitances: $C_1^p = C_2^p$=0.5aF

Coupling & Load Capacitances: $C_b = 9.1aF, C_L = 9aF, C_0 = 10aF$

Junction Resistance R_j=$10^5\Omega$

Supply Voltage $V_s = 16mV$

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

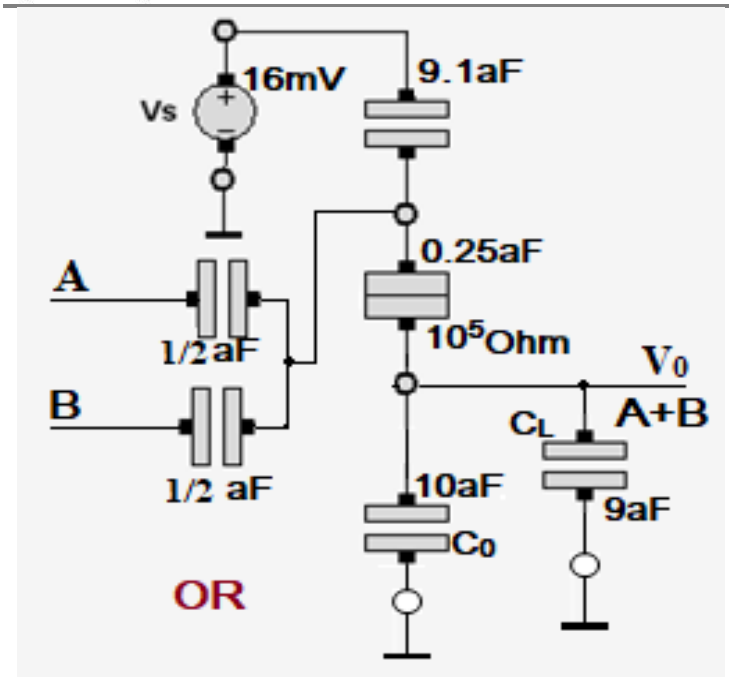


Fig. 8(b) OR gate

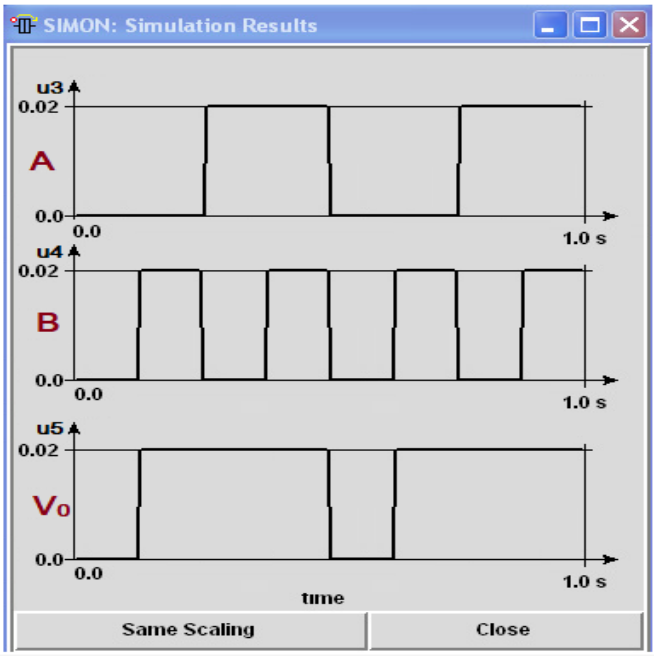


Fig. 8(c) simulation result of OR gate

DELAY, SWITCHING ENERGY AND OPERATING SPEED

Every electronic circuit exhibits a characteristic processing delay that varies based on its architecture. The proposed circuit in Fig. 8(a) relies on specific threshold gates—namely AND, OR, and XOR—each possessing a distinct propagation delay. To evaluate and compare the performance and efficiency of any circuit (whether conventional, Nano-scale, or pico-scale), several critical figures of merit must be analyzed:

- **Circuit Complexity:** Total component count and structural density.
- **Operating Speed:** Total execution time and clock freqency.
- **Energy Efficiency:** Power consumption and dynamic switching energy.
- **Signal Integrity:** Fan-out capabilities and load driving capacity.
- **Manufacturing Feasibility:** Material costs and fabrication constraints.
- **Charge Dynamics:** Electron transport control and tunneling behaviors.
- **Environmental Stability:** Thermal tolerance and operating temperature limits.

To calculate the processing or switching delay in a multi-input threshold logic gate, both the tunnel junction capacitance V_j and its internal critical voltage V_c must be considered. Based on the established logarithmic expression for error probability [3–7, 15], the switching delay is defined by the following equation:

$$\text{Delay} = -(e|\ln(P_{error})|R_t) / (|V_j| - V_c) \cdots\cdots (14)$$

In this framework, V_c represents the critical voltage, $V_j$denotes the tunnel junction voltage, and $R_t$signifies the internal resistance of the tunnel junction. A tunneling event—which drives the execution of the logic gate—is triggered when the magnitude of the tunnel junction voltage V_j, operating as a fraction of the bias voltage V_b, meets or exceeds the internal critical voltage V_c, satisfying the condition: $|V_j| \geq V_c$

For a 2-input AND gate operating under this threshold, the junction voltage is V_j =10.80mV and the critical voltage is V_c= 11.58 mV [2,7]. Applying Equation (17) with a target error probability $P_{error} = 10^{-12}$and a

tunnel resistance $R_t = 10^5\Omega$, the gate delay is calculated as $0.062|\ln(P_{error})| = 1.71$ns. The evaluated propagation delays for the somecircuits are compiled in Table 4.

When an electron tunnels through the junction barrier, it alters the electrostatic energy of the system. This energy shift,ΔE, is defined as:

$$\Delta E = E_{before\ tunnel} - E_{after\ tunnel}$$

$$= -e(V_c - |V_j|) \cdots\cdots\cdots\cdots\cdots\cdots (15)$$

This represents the active switching energy dissipated during a tunneling event. Individual energy consumption values for the TLG circuits are provided in Table-6.

Table-6

Gate/Device	SET-based delay ns	LTG-based delay ns	Speed-up (times)	Switching Energy (meV)
2-input NAND	4	$0.062\|\ln(P_{error})\| = 1.71$	2.33	10.80
2-input XOR	4	$0.080\|\ln(P_{error})\| = 2.21$	1.81	21.20
3-input OR	8	$0.073\|\ln(P_{error})\| = 2.01$	3.98	11.58
2-input NOR	4	$0.060\|\ln(P_{error})\| = 1.65$	2.42	10.70
2-input OR	4	$0.062\|\ln(P_{error})\| = 1.71$	2.33	10.80

To evaluate performance, the processing delays of CMOS, single-electron transistor (SET), and linear threshold gate (LTG)-based designs were evaluated and compared. For standard CMOS logic gates (such as NAND, NOR, AND, and XOR), the processing delay was found to be 12 ns [3-7]. By comparison, the electron tunneling delay in a SET is roughly 4 ns [1-3, 7–11]. The specific speed-up achieved by LTG-based designs relative to SET-based delays is outlined in Table 4. Additionally, the relationship between switching delay and switching error probability is illustrated in Figure 9(a), while the impact of the unit capacitance C on switching delay is presented in Figure 9(b).

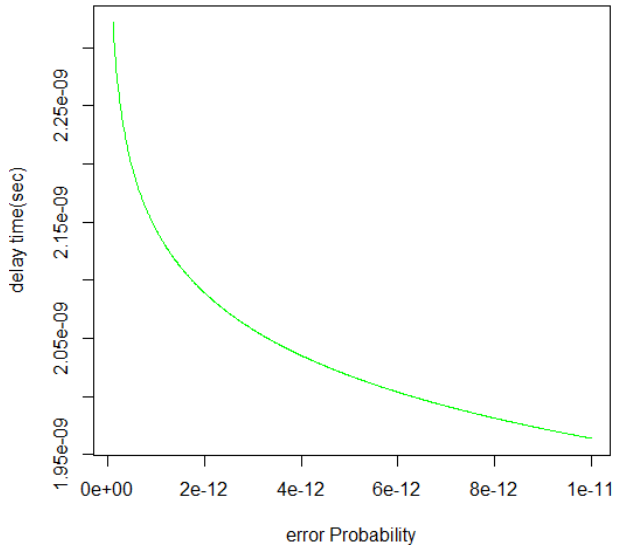


Fig. 9(a) Delay vs. Error Probability

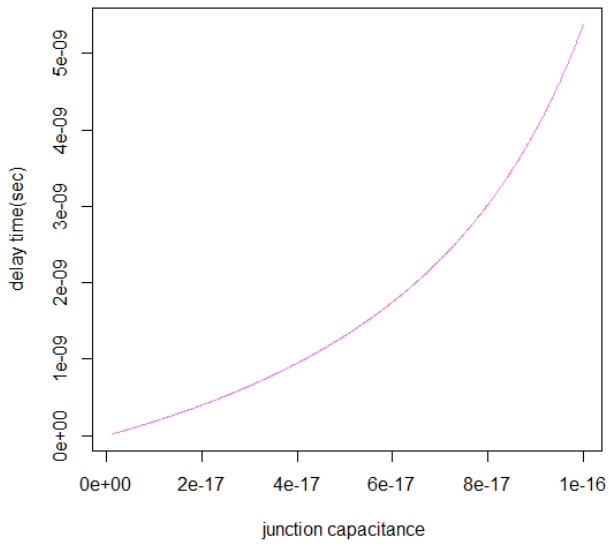


Fig. 9(b) Delay Vs. capacitance

CONCLUSION

This study presents the implementation of a Full Adder, 2-input and 3-input AND gates, and a 2-input OR gate utilizing an emerging nanotechnology based on Threshold Logic Gates (TLGs). By leveraging a Multiple-Input Single-Output (MISO) TLG architecture, complex combinational circuits like the Full Adder can be

efficiently constructed. Due to the ultra-small footprint of the underlying tunnel junctions (approximately 10 nm in diameter), these TLG-based circuits achieve exceptionally high integration and functional density. The paper discusses the construction of a generic Linear Threshold Logic Gate, demonstrating how it can be adapted to derive standard logic gates such as AND, NAND, OR, XOR, and Full Adders. To validate this approach, specific circuits—including the 2-input and 3-input AND gates, the 2-input OR gate, and the Full Adder—were implemented and verified through SIMON simulations. Key performance metrics, including the required element count, processing delays, and power consumption, are presented in tabular form and further illustrated through corresponding behavior curves. The results indicate that in single-electron tunneling (SET) technology, threshold logic gates operate at least two times faster than conventional SET-based logic gates[17-18]. The simulation data perfectly aligns with theoretical expectations, practical operation currently requires an atmospheric temperature maintained at 0K.

REFERENCES

1. Anup Kumar Biswas, “Single Electron Tunneling based Threshold Logic Unit”, 5th International Conference , CICBA 2023, Kalyani, India January 27-28, 2023, Springer
2. Anup Kumar Biswas, “Booth’s Multiplication based reversible gates and Single Electron Tunneling gates”, “ Journal for Basic sciences” Vol. 25, Issue 4, 2025; pp466-484
3. 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) August2021; ISSN 1605-6582
4. 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
5. 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,
6. 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
7. 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,
8. 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,
9. 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
10. C. Lageweg and S.Cotofana and S. Vassiliadis, “Achieving Fanout Capabilities in Single Electron Encoded Logic Networks,” in 6th InternationalConference on Solid-State and IC Technology(ICSICT), October 2001, pp. 1383–1386.
11. Alexander N. Korotkov, “Single-electron logic and memory devices” INT. ELECTRONICS, 1999, Vol. 86, No. 5, 511- 547
12. S.Cotofana and S.Vasiliadis, “Low Weight and Fan-In Neural Networks for Basic Arithmetic Operations,” in Congress on Scientific Computation, Modelling and AppliedMathematics, Volume 4: Artificial Intelligence and Computer Science, August 1997,pp. 227–232.
13. 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
14. 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
15. K. Likharev, “Single-electron devices and their applications,” Proc. IEEE, vol. 87, pp. 606–632, Apr. 1999.

16. J. R. Tucker,- "Complementary digital logic based on the Coulomb-blockade", Journal of Applied Physics., vol. 72, no. 9, pp. 4399–4413, November 1992.
17. Jacob. Millman and C. C. Halkias; "Integrated Electronics- Analog and Digital Circuits and Systems - second edition" McGraw Hill Education;
18. Millman's Electronic Devices & Circuits 4th Edition (English, Paperback, Jacob Millman)
19. Asahi N., Akazawa, M. and Amemiya, Y.. 1997. Single electron logic device based on the binary decision diagram, IEEE Trans on ED. 44, 7. pp. 1109-1116.
20. Asahi N. Akazawa, M. and Amemiya, Y. 1998. Single electron logic systems based on binary decision diagram. IEICE Trans. Electron. E-81-C. 1. pp. 49-56.
21. 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

BIOGRAPHY



ANUP KUMAR BISWAS, is an Associate 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 and Telecommunication Engineering. Dr. Biswas has published 30 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 21 years.