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/IJSRSET218610

Finding out square root of an integer number using Single Electron Transistor

Dr. Anup Kumar Biswas
Assistant Professor, Department of Computer Science and Engineering
Kalyani Govt. Engineering College, Kalyani, Nadia-741235, West Bengal, India
akbcse456@gmail.com

Article Info
Volume 8, Issue 6
Page Number : 96-110

Publication Issue :
November-December-2021

Article History
Accepted : 20 Nov 2021
Published: 30 Nov 2021

ABSTRACT

The single-electron transistor (SET) attracts the researchers, scientists or technologists to design and construct large scale circuits for the sake of the consumption of ultra-low power and its small size. All the incidences in a SET-based circuit happen when only a single electron tunnels through the transistors under the proper applied bias voltage and a small gate voltage or multiple gate voltages. The oscillatory conduction as the function of the variable-multiple /single gate voltage is exhibited by SET. This uncommon characteristic provides the ability of executing the functions of AND, OR, XOR, Inverter and some combinational circuits like multiplexer, subtractor etc. For implementing a square root circuit, SET would be a best candidate to fulfil the requirements. The processing speed of SET based devices will be nearly close to electronic speed. Noise during processing gets ultra-low when the circuits is built with SETs. The square root circuit is presented here for sixteen bit input numbers. The input bit numbers can be increased with the increasing of the depth of the pattern very easily. And this will provide us the greater accuracy about the squared root value. Power consumption in the single electron circuit is low irrespective of bipolar junction transistor (BJT) or Complementary Metal Oxide Semiconductor (CMOS) circuits. Reducing the numbers of nodes, the power consumption is reduced.
Keywords : electron-tunneling, Coulomb blockade, single-electron transistor (SET), logic gates, square root

I. INTRODUCTION

Coulomb blockade in a tunnel junction

Between two conducting materials, a thin insulator is sliced to make a tunnel junction. These conducting materials considered as two electrodes required for connecting the bias voltage for tunneling phenomenon. If anybody selects the electrodes as superconducting substances, the current carrying through the junction will be by virtue of the two types of elementary charges, electron and proton, defined as Cooper pairs.

Instead of superconducting electrodes, one's intention to use normal conducting things whose characteristics differ from semiconductor and superconductor is to provide the current by only a single elementary charge—electron.

For the case of Classical electrodynamics, through the insulating wafer no current can pass. On the other hand, for the case of Quantum mechanics, there is a nonfading probability of an electron-transport from one electrode to another through the insulator. For the Quantum mechanics case, if we apply bias voltage to the tunnel electrodes then the current following through it will be in proportionate to the voltage provided, we are neglecting all other additional effects like shunt effect. So the tunnel junction acts as a resistor whose resistance varies exponentially with the thickness (nanoscale) of the wafer-barrier.

The tunnel junction discussed above has a resistance along with a capacitance. In this case the capacitance is made up of two conducting plates and the dielectric insulator between them. The value of the capacitance is finite (auto scale).

When the current passing through the tunnel junction, it is the fact that there is a sequence of events of tunneling of exactly one electron at a time i.e., not co-tunneling in which a pair of electrons passes (tunnels) concurrently.

For the discreteness of electrical charge, the current passing through a tunnel junction is a series of events and only a single electron *tunnels* through the tunnel-barrier at a time when tunneling. The capacitor of the tunnel junction is charged with one elementary charge (i.e. 1.602×10^{19} coulomb) by the tunneling electron, causing a voltage buildup V=$\frac{e}{C}$, where e is the elementary charge and C the capacitance of the junction. If the value of C is very small, the voltage built up can be large enough to prevent another electron from tunneling. We are able to suppress the electric current with a low bias voltage, as a result the resistance of the device will remain no longer constant. The increment of the differential resistance around zero bias is defined as the Coulomb blockade.

II. Inverter

The inverter [7, 9, 10, 12, 20] depicted in Fig.1 is constructed with two Single electron transistors (SETs) in series connection. Input voltage is directly coupled to both of the islands of the SET1 and SET2 through the capacitances C_1 and C_2 respectively. The island of each SET is a very small size of gold having total capacitance of C_{tot}=1pF. The output terminal V_0 is connected to the common channel of the two SETs and to the ground via a capacitor C_L to put down charging effects. The two extra gates V_{g1} and V_{g2} through two capacitors C_{g1} and C_{g2} respectively are intentionally connected to the islands of SET1 and SET2 to tune the induced charges on the islands of SET1 and SET2.

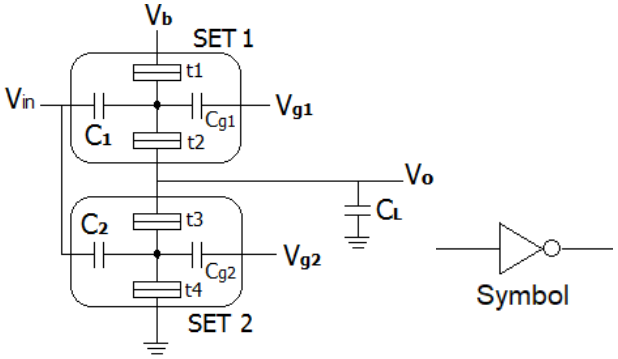


Fig.1 SET based inverter

For the case of inverter, the parameters values chosen are: 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 $C_{g2} = \frac{17}{4}C$. For the simulation cases the value of C, we consider, is=1aF.

The operation of the inverter will be like this: - the output V_0 value will be high when the input voltage V_{in} is low and V_0 value will be low when the input voltage is high. To obtain this target, we put the voltage V_{g1}=V_{g2} which will be equal to $\frac{e}{C_{tot}}$=16mV along with the tuning gate voltages (at

present V_{in}) of SET1 and SET2, which are adjusted so that as soon as the V_{in} is low the SET1 is in conduction mode and the SET2 is in Coulomb blockade.

This effectively results the output to bias voltage V_b and causes the output voltage to high, i.e, this effectively connects the output to the supply voltage and makes the output high.

Coulomb blockade interferes the steady flow of current as when the high voltage is applied to the input it causes to shift the induced charge on each of the islands of the SETs by a part of an electron charge and keeps the SET1 in the state of Coulomb blockade and the SET2 in the conducting mode. As a consequence, the output result shifts from high to low.

In this work I assume the Boolean logic inputs corresponding to the voltages like: logic "0" =0 Volts and logic "1"=0.1× $\frac{q_e}{C}$

For our simulation and other purposes, we must assume C=1aF then 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 ≅ 16 mV

III. Single Electron NAND and AND gates

By using four SETs, a two input NAND gate [1, 18] can be constructed as depicted in Fig-2(a). The **Single Electron Transistor** circuits are configured as CMOS circuits, the pull up transistor called p-channel (P-SET) is tied to V_d and the pull down transistor called n-channel transistor (N-SET) tied to ground. The p-channel bears doubles the area factor of the n-channel SET which is the equivalent of setting twice the gate width. For the case of **Single Electron NAND,** the upper two SETs called P-channel Single Electron Transistors (P-SETs) are connected in parallel and the lower two SETs called N-channel Single Electron Transistors (N-SETs) are connected in series. When each of the input voltages V_{in1} and V_{in2} are at high levels, the lower two SETs should be in tunneling mode and a single electron must pass through them and virtually the output V_{out} connected to ground which leads V_{out} to low. As opposed to, if any one or both of the inputs V_{in1} and V_{in2} are low, the output is pulled to V_d by the P-SETs which (either one or both) are in tunneling mode and the resultant output becomes high as the N-SETs are in Coulomb blockade.

When the output of the NAND gate is connected to the input of inverter shown in Fig-1, we get the Single Electron AND gate that is drawn in Fig-2(b)

For the case of AND and NAND Gates, the parameters values chosen are: $C_L = 9aF$, $Drain = 0.5aF$, $Source = 0.1aF$, and $C_1 = C_2 = 0.5aF$.

The same parameters are chosen for the case of OR, NOR, XOR, 2:1 Multiplexer, and subtraction circuit.

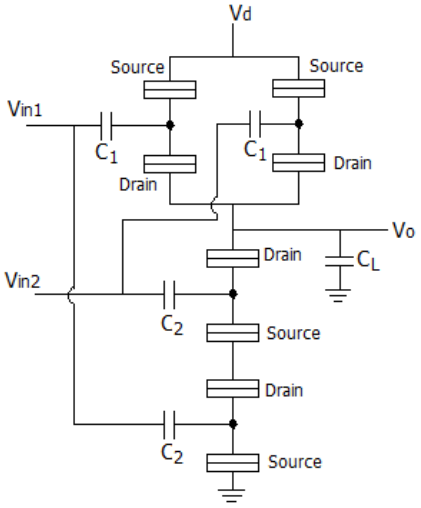


Fig.2 (a) NAND Gate

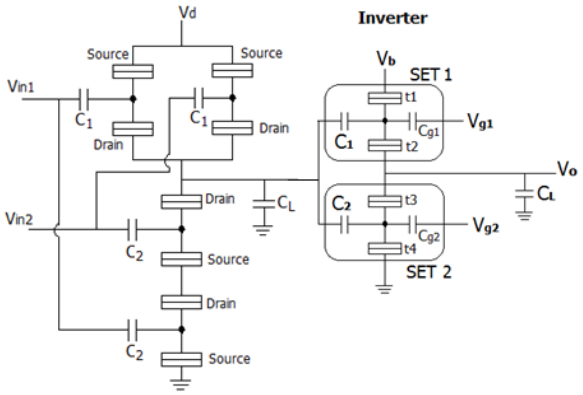


Fig.2 (b) AND Gate

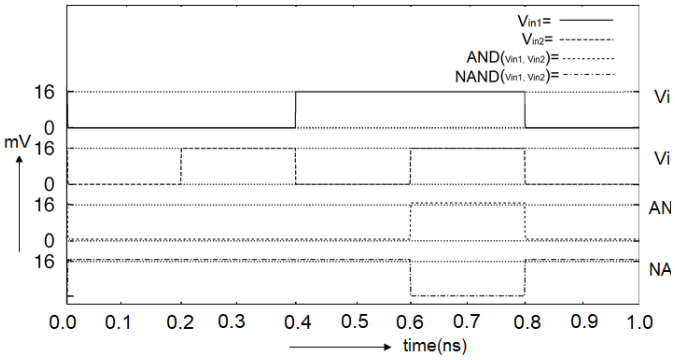


Fig.2(c) simulation result of AND and NAND Gate

IV. Single Electron NOR and OR gates

For the case of NOR gate [1, 13, 18] four SETs are taken of which upper two P-SETs are connected in series and the lower two N-SETs in parallel shown in Fig-3(a), When both input voltages V_{in1} and V_{in2} are low, the upper two SETs are in tunneling mode towards V_d , the lower two N-SETs are in Coulomb blockade and the output V_{out} connected to bias voltage V_d , so the output V_{out} becomes high.

When any one or both of the inputs V_{in1} and V_{in2} are low , the upper P-SETs are non-tunneling and the lower N-SETs (any or both) are in tunneling and the output is virtually grounded. This indicates the output voltage V_{out} low. If we connect the output of NOR gate to the input of the inverter, the combined circuit is treated as **Single Electron OR gate** shown in Fig-3(b).

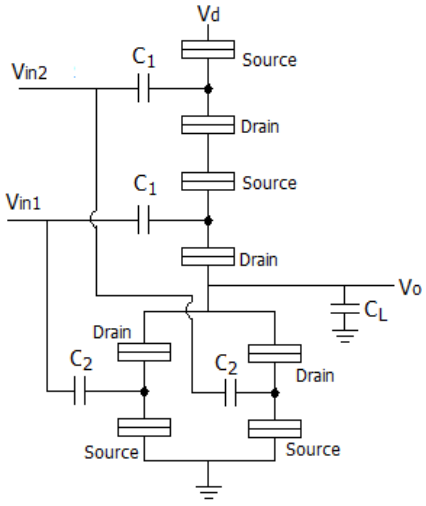


Fig.3 (a) NOR gate

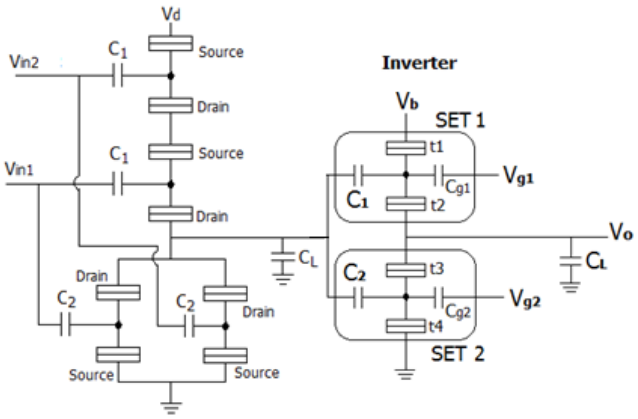


Fig.3 (b) OR Gate

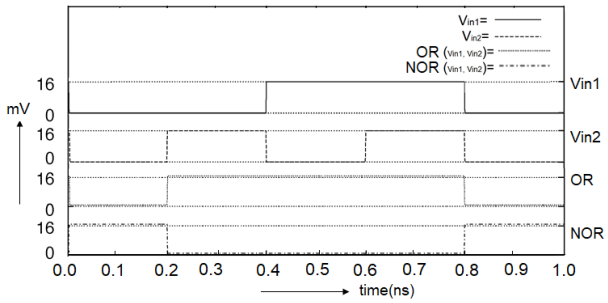


Fig. 3(c) simulated result of OR and NOR gate

V. Single Electron XOR gate

Single Electron transistor exhibits an oscillatory conductance [Fig. 4] when we apply gate voltage as input voltage. This enables the SET the capability of functioning as an exclusive-OR gate.

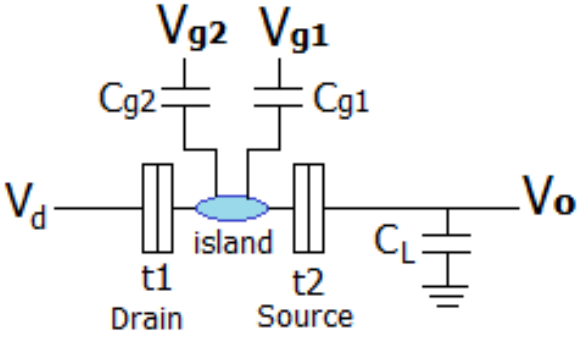


Fig.4 XOR Gate

There are two gates capacitors C_{g1} and C_{g2} having almost the same value for their symmetric constructions in fabrication. A discrete load capacitor C_L is connected to the drain and the output voltage is taken as the voltage V_o of the load capacitor in response to the combined input gate voltages V_{g1} and V_{g2} in the range 0 to 16mV.

When both the input gate voltages V_{g1} and V_{g2} are 0V or 16mV the output shows low and when one of them is high i.e., 16mV, the output is high. These phenomena indicate the operation of an XOR gate [2, 3, 4, 6, 13, 20].

When anybody wishes to use V_{g1} as electrode and V_{g2} as control gate, then the device can transmit current signal for a particular voltage at V_{g1}.

By using conventional logic circuits we need 16 transistors to make an XOR gate [1, 5, 16, 17, 19] on the other hand the function of it can be implemented with a single SET. The two gate capacitances Cg1 and Cg2 are almost equal on account of the symmetric structure of the gate. The range of V_{g1} and V_{g2} are kept between 0 and 16mV.

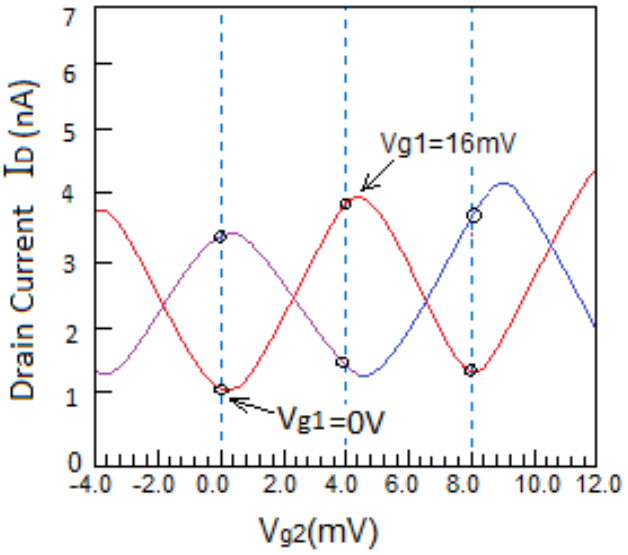


Fig.4(a)

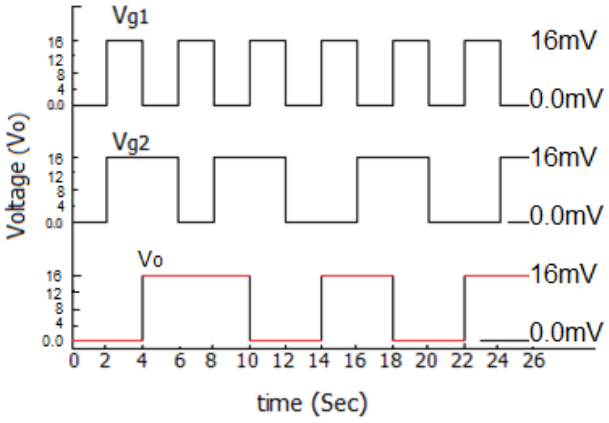


Fig. 4(b)

VI. Multiplexer

A Multiplexer is combinational circuit that connects to one input out of many inputs and send that to the output. It is a combinational circuit whose present output don't rely on the previous output. The circuit needs n control lines for 2^n inputs. A circuit for 2:1 Multiplexer [21, 22] drawn in Fig. 5(a) containing two inputs V_{in1} and V_{in2}, output V_o and a single control line input 'Sel' that have two operational stages. The outputs of the first two stages (upper and lower in Fig. 5(a)) which are really two AND gates give the outputs $\text{Vin1}.\overline{\text{Sel}}$ and Vin2.Sel respectively. The last stage is a two input OR gate which gives the output ($\text{Vin1}.\overline{\text{Sel}}$ +Vin2.Sel). If Sel=0, the output is V_0=V_{in1} and if Sel =1, the output is V_0=V_{in2}. So the circuit given in Fig.5(a) acts as a 2:1 Multiplexer.

When we trigger the selection line (Sel) from low to high or from high to low, the delay time becomes approximately 1 picosecond. Same time delay is for the case of inputs. The time it takes for stable in high stage is approximately 15-20 picoseconds.

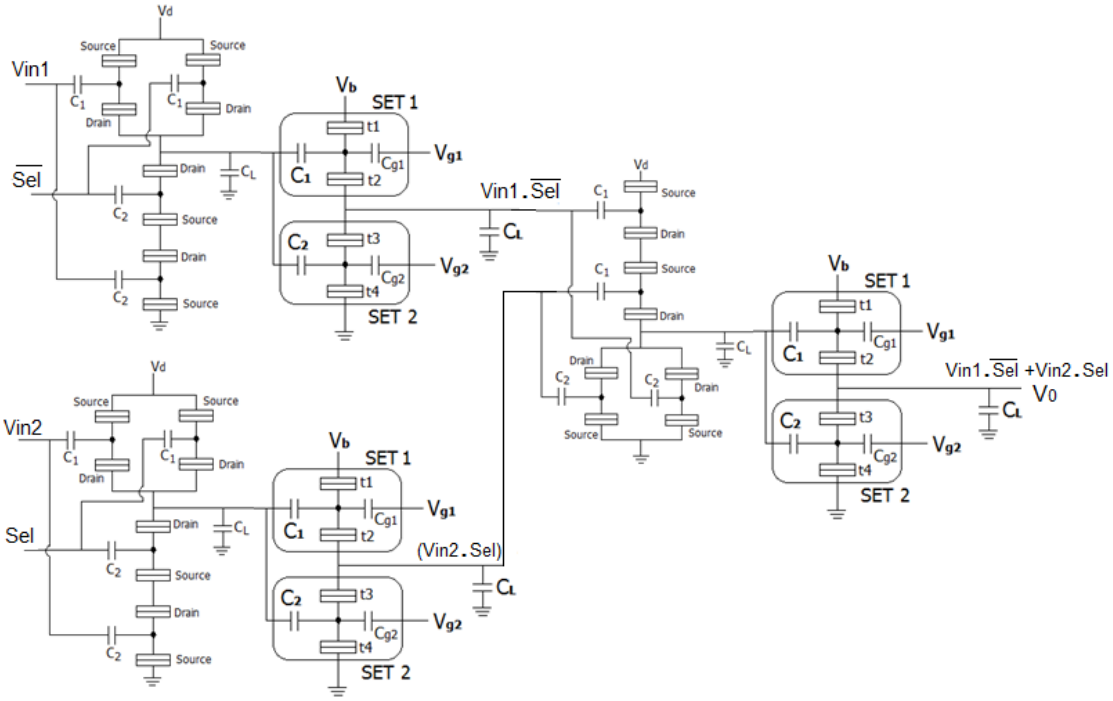
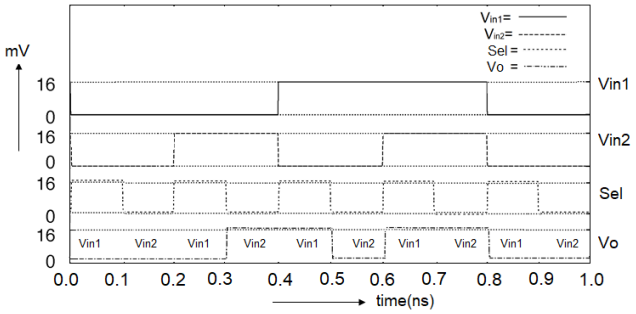


Fig. 5(a) 2:1 Multiplexer



Fig, 5(b) simulated result of 2:1 Multiplexer

VII. Subtraction Circuit

For the implementation of a square root circuit we need to construct a subtraction circuit. The input-output relationship of this circuit and the related equations that can be exposed by using Karnaugh map is shown in the Table-1. Based on the single electron transistor gates like AND gate, XOR gate, OR-gate and inverter gate a subtractor (SUB=DIFF) with its borrow (B_{out}) has been drawn in Fig.6. We can write the equation related to the subtractor as:

Diff=X'Y'Z + XY'Z' + X'YZ' + XYZ = X⊕Y⊕Z = X⊕Y⊕ B_{in} ……(i)

B_{out} =X'Y'Z+ X'YZ' + X'YZ + XYZ ……(ii)

Table-1

X	Y	B_{in}	DIFF	B_{out}	Diff=X'Y'Z + XY'Z' + X'YZ' + XYZ= X⊕Y⊕Z = X⊕Y⊕ B_{in}
0	0	0	0	0	
0	0	1	1	1	B_{out} =X'Y'Z+ X'YZ' + X'YZ + XYZ
0	1	0	1	1	=X'Z(Y'+Y)+ Y(X'Z' + XZ)
0	1	1	0	1	= X'Z+Y(X⊕Z)'
1	0	0	1	0	= X'Y+Z(X⊕Y)'
1	0	1	0	0	
1	1	0	0	0	
1	1	1	1	1	

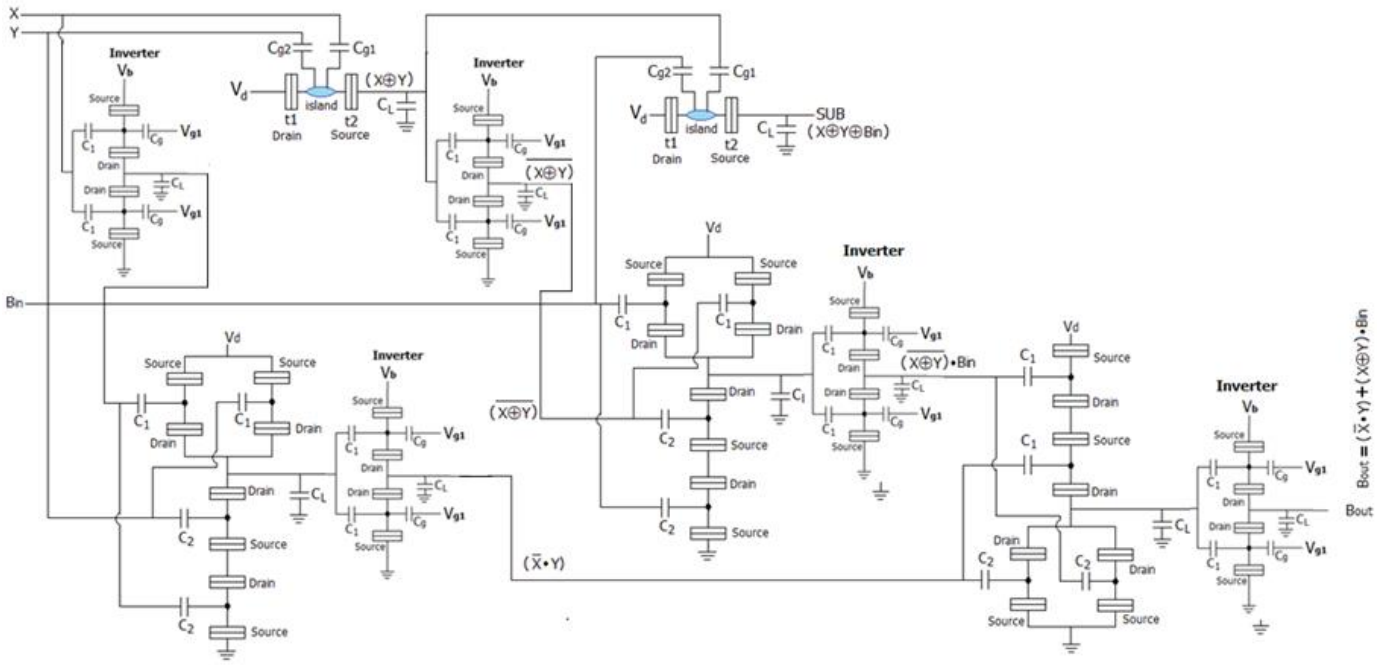


Fig. 6 A Subtraction Circuit based on SET

VIII. Restoring Algorithm for calculating the square-root

Algorithm for Square-root

(i) Start

(ii) Initialize n-bit radicand which is to be squared root

(iii) Segment the radicand into groups of two bits starting from least significant bit

(iv) Starting from left, if the radicand size is odd, the first group is considered as one bit otherwise the first group is of two bits.

(v) Take '1' squared and subtract it from first group. if the subtracted value/remainder is positive the first square root is '1', if negative square root is '0'.

(vi) Shift 2nd group of two bits from the remaining of the radicand to the right of the subtracted value, subtract it by the guess square with append '01'. If the subtracted value is positive the 2nd square root is '1', if negative 2nd square root is '0'.

(vii) When negative, restore
Its meaning is: add this negative value with previous "the guess squared with append 01" to get the remainder.
And follow the steps until last group of two bits is shifted from.

Following restoring algorithm, we have tried to find out the square root of the number $1111110110_2 = 1014_{10}$. After manipulation of 16 steps, the result given at the topmost row in the Table-2 is 11111.110101111_2. The decimal value of this binary number is being=31.841796875_{10}.

Table-2

1	1 1 1 1 1 . 1 1 0 1 0 1 1 1 1 $\sqrt[2]{1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0.0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}$ 0 1	0 1
2	1 0 1 1 +ve [when 1st +ve or –ve then sqrt bit starts] → 1 0 1	1 1 0 1
3	1 1 0 1 1 +ve 1 1 0 1	1 1 1 0 1
4	1 1 1 0 0 1 +ve 1 1 1 0 1	1 1 1 1 0 1
5	1 1 1 0 0 1 0 +ve 1 1 1 1 0 1 1	1 111101
6	1 1 0 1 0 1 0 0 +ve 1 1 1 1 1 0 1	1 5th from right 1111101
7	1 0 1 0 1 1 1 0 0 +ve 1 1 1 1 1 1 0 1	. 1 11111101
8	0 1 0 1 1 1 1 1 0 0 +ve 1 1 1 1 1 1 1 0 1	1 111111101
9	1 0 1 1 1 1 1 0 0 -ve 1 1 1 1 1 1 1 0 1 restore	0
10	1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1	-- 1111111001
11	1 1 1 1 1 0 1 1 1 0 0 +ve 1 1 1 1 1 1 1 1 0 1 0 1	1 11111110101
12	-ve 1 1 1 1 1 1 1 1 0 1 0 1 restore	0
13	1 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 1 0 0 1	-- 111111101001
14	1 1 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1	1 +ve 1111111010101
15	1 1 1 1 0 0 1 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1	1 +ve 11111110101101
16	1 1 1 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1	1 +ve 111111101011101
	1 1 0 0 1 1 0 0 1 0 1 1 1 1 1	1 +ve

The original number of $\sqrt[2]{1014}$ = 31.84336665618132

From the Table-2, we observe the different square root values from the **top-most** row as

11111.110101_2 = 31.828125_{10} [from top most row of Table 2] six digits after dot

11111.1101011_2 = 31.8359375_{10} [from Table 2] seven digits after dot

$11111.11010111_2 = 31.83984375_{10}$

[from Table 2] eight digits after dot

$11111.110101111_2 = 31.841796875_{10}$ [from Table 2] nine digits after dot

The error for the last step is calculated with respect to 31.84336665618132 is

$$\frac{\sqrt[2]{1014} - 31.841796875}{\sqrt[2]{1014}} = \frac{0.0015697811813153}{31.84336665618132} = 4.929696028263887 \times 10^{-5}$$

$= 0.0049296960282639\% \approx 0.0049\%$ (**error**)

So the error is 0.0049% which is acceptable. The more the steps we proceed, the less the error we obtain.

IX. Implementation of Square root circuit

The building block (cell) of the Square-root circuit is also called as subtract-multiplex (SM). The inputs of the SM are X, Y, B_{in} and Sel, and an outputs are Bo (borrow) and V_o (result). When S=0, then difference will be equal to X-Y-Bin (X minus Y minus Bin) else it will be equal to X.

Based on the Single Electron transistor, the fundamental gates like AND, OR, XOR and Inverter are already drawn. Another two circuits like Multiplexer and Subtractor are shown as well. Depending on the algorithm to draw the square-root circuit, we require two fundamental cells shown in Fig.8(b) which is drawn in Fig.7 and in Fig.8 (a) that is the same as Fig.8 (b) but only a inverter is added to

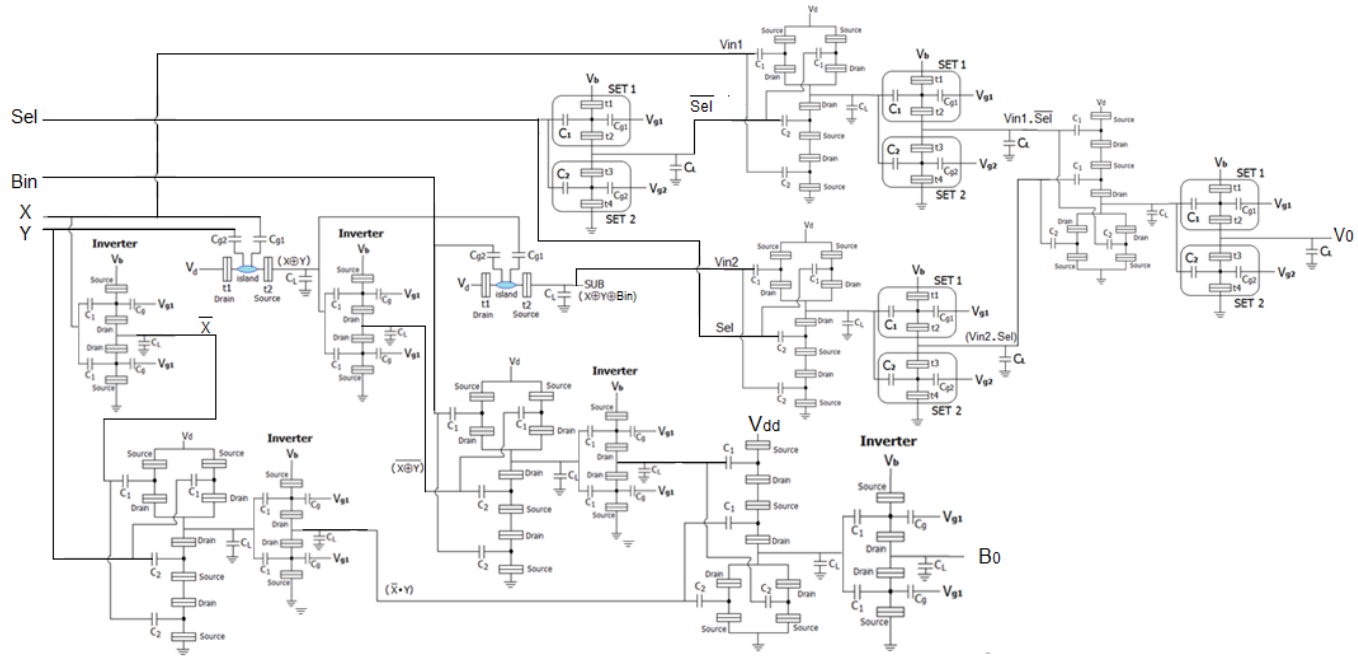


Fig.7 Building block (cell) of the Square root circuit

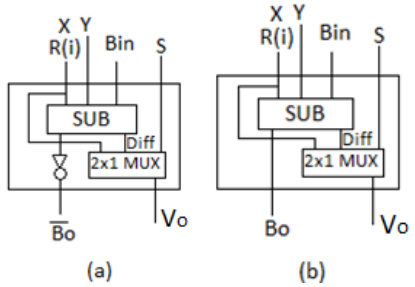


Fig. 8(a) "Symbol sqrt ckt" (a) with inverted borrow $\overline{Bo}$" and 8(b) "with borrow Bo"

create complement $\overline{Bo}$ instead of Bo. The leftmost column cells of Fig.9 are of 8(a) and the remaining cells are of 8(b). The square root circuit drawn in Fig 9 has eight steps to calculate eight bits square root of 16 bits radicand R(0), R(1), R(2), R(3), R(4),…..,R(15). The square root bits are SR(7), SR(6), SR(5), SR(4),….,SR(0). When we increase

the steps along with the increase of two cells each time, we get the more accurate result(s).

If our total pipeline stages are 16, we require 32 bits input (radicand). Some input-output instances are exposed in the Table-3 for the diagram given in Fig. 9. Here are 16 radicand bits and the output are of 8 bits.

For the Fig.9 we have taken only integer input. This is why there are some error in the calculated root values as the output bit length equals to 8. If the pipeline stages are increased to 16, 32 bits inputs will be required.

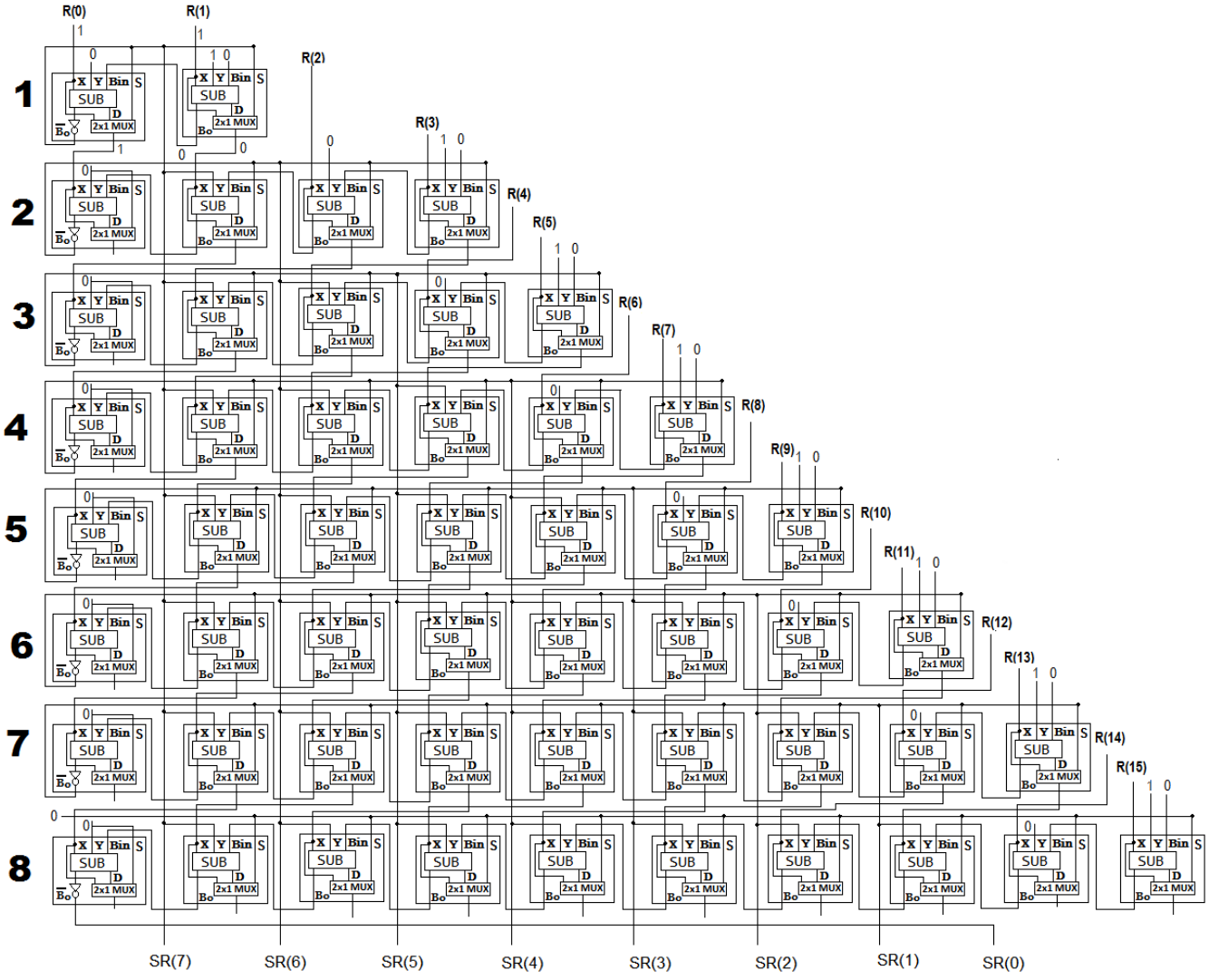


Fig.9 SET-based Square root calculation circuit

Now we shall fix the dot (point) in the mid-point of input then the dot is also to be fixed in the middle of the output bits for getting the fractional output squared values. Noted that 16 stages pipelines are not shown in the figure for the cause of limited space. But results of 16 bit for 32 bit inputs are delivered in Table-4.

Table 3

Sl.No.	Result from C- Program	Input and calculated root (Pipeline stage=8)
1	Radicand r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16 is =1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 INPUT=43711 Square root of r1,...,r16 is =1 1 0 1 0 0 0 1 Calculated root = 209	Input =1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 INPUT=43711 Square root is =1 1 0 1 0 0 0 1 Calculated root = 209
2	Radicand r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16 is =0 1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 INPUT=30950 Square root of r1,...,r16 is =1 0 1 0 1 1 1 1 Calculated root = 175	Radicand =0 1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 INPUT=30950 Square root =1 0 1 0 1 1 1 1 Calculated root = 175
3	Radicand r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16 is =0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 INPUT=245 Square root of r1,...,r16 is =0 0 0 0 1 1 1 1 Calculated root = 15	Radicand =0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 INPUT=245 Square root =0 0 0 0 1 1 1 1 Calculated root = 15
4	Radicand r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16 is =1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 INPUT=62259 Square root of r1,...,r16 is =1 1 1 1 1 0 0 1 Calculated root = 249	Radicand =1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 INPUT=62259 Square root is =1 1 1 1 1 0 0 1 Calculated root = 249

Table 4

Sl. No	Result from C- Program	Input and calculated root (Pipeline stage=16)
1	Radicand =1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=43711.000000 Square root of r1,...,r32 is =1 1 0 1 0 0 0 1 . 0 0 0 1 0 0 1 0 Calculated root = 209.070313	Radicand =1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=43711.000000 Square root is =1 1 0 1 0 0 0 1 . 0 0 0 1 0 0 1 0 Calculated root = 209.070313
2	Radicand =0 1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=30950.000000 Square root of r1,...,r32 is =1 0 1 0 1 1 1 1 . 1 1 1 0 1 1 0 1 Calculated root = 175.925781	Radicand =0 1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=30950.000000 Square root is =1 0 1 0 1 1 1 1 . 1 1 1 0 1 1 0 1 Calculated root = 175.925781
3	Radicand =0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=245.000000 Square root of r1,...,r32 is =0 0 0 0 1 1 1 1 . 1 0 1 0 0 1 1 1 Calculated root = 15.652344	Radicand =0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=245.000000 Square root is =0 0 0 0 1 1 1 1 . 1 0 1 0 0 1 1 1 Calculated root = 15.652344
4	Radicand =1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=62259.000000 Square root of r1,...,r32 is =1 1 1 1 1 0 0 1 . 1 0 0 0 0 1 0 0 Calculated root = 249.515625	Radicand =1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 . 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=62259.000000 Square root is =1 1 1 1 1 0 0 1 . 1 0 0 0 0 1 0 0 Calculated root = 249.515625

When we are interested in comparing the calculated root in Table-4 with the value from calculator, there will be an error and this error shown in Table-5 is in tolerable level.

In Table-6, any binary value of word-length 32 is taken keeping the dot (point) in the middle position (between 16th and 17th bit) and result is shown.

Table 5

Sl. No.	Input value	Square root taken from calculator(X)	Calculated sqr root (Y) from ckt	Difference=X-Y	Error $\frac{X-Y}{X} \times 100\%$
1	43711	209.071758016237095016287079638 54	209.070313	0.001445016237095016287079638 54225	0.00069 %
2	30950	175.926120857591809745296172784 88	175.925781	3.39857591809745296172784881893 41e-4	0.00019 %
3	245	15.652475842498527874864215681 119	15.652344	1.31842498527874864215681118933 65e-4	0.00084 %
4	62259	249.517534453999383394412995188 689	249.515625	0.00190945399383394412995188688 8775	0.00076 %

Sl.No.	Table 6
1	Radicand =1 0 1 0 1 1 1 1 1 0 0 0 1 1 1 0 . 0 0 1 1 1 0 0 1 1 0 0 1 1 0 1 1 INPUT=44942.225021 Square root of r1,...,r32 is =1 1 0 1 0 0 1 1 . 1 1 1 1 1 1 1 0 Calculated root = 211.992188
2	Radicand =0 1 0 0 0 1 1 1 0 1 1 1 0 1 1 1 . 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 INPUT=18295.799988 Square root of r1,...,r32 is =1 0 0 0 0 1 1 1 . 0 1 0 0 0 0 1 1 Calculated root = 135.261719
3	Radicand =0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 . 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 INPUT=10.000076 Square root of r1,...,r32 is =0 0 0 0 0 0 1 1 . 0 0 1 0 1 0 0 1 Calculated root = 3.160156
4	Radicand =0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 . 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 INPUT=0.750000 Square root of r1,...,r32 is =0 0 0 0 0 0 0 0 . 1 1 0 1 1 1 0 1 Calculated root = 0.863281

X. Time delay

For a CMOS/TTL logic gate the time delay for a gate is 12ns [7, 15], on the contrary the time required for tunneling through a SET is approximately 4ns [2] [3]. The XOR gate using conventional logic circuits needs 16 transistors, whereas this function can be implemented with just one SET [1,2,3,8].

All the gates and devices drawn in Fig.1 through 7 are combinational circuits. Considering that they are constructed as different pipeline stages, we are easily

able to calculate the time required to find out the output of a circuit. In table-7, the delays involved up to the Fig.7 are set in.

Table-7		
Sl. No.	Circuit name	SET Circuit Delay(ns)
1	Inverter	4×2=8
2	NAND/AND	4×3=12
3	NOR/OR	4×3=12
4	XOR	4×1=4
5	SUTRACTOR	4×10=32
6	MUX	4×10=40
7	Symbol8(a)/ Symbol8(b	4×12=48
8	Squared Root Circuit	(Stage number× 48) For 8 stage pipelines 8× 48=384ns For 16 stage pipelines 16× 48=768ns

From the Table-7, we can find the Square root without fractional parts of a 16 bit input number after 265ns. Hence the frequency of the output is $\frac{1}{384ns}$ = $\frac{10^9}{384}$ = 2604167Hz

For more accurate result we need 16 pipeline stages and 32 bit input number. In this case, the time delay is 512ns and the frequency of the output is $\frac{1}{768ns} = \frac{10^9}{512}$ = 1302083Hz

XI. Switching Power

For switching energies required for the AND, OR, NOR, NAND and INVERTER are 10.8meV, 10.8meV, 10.7meV, 10.7meV and 10.4meV respectively [8]. Depending on these, the switching energies necessary for the different gates, combinational circuits and square root circuits in this work are given in the Table-8. It is really clear that the energy required will be of order O(10^{-17}). So very negligible energy is required in comparison with our conventional TTL-, JFET- or CMOS-based gates O(10^{-6}) and combinational circuits.

Table-8		
Gate Name	**Required elements**	**Energy (Joule)**
AND	fourteen (13)	$1.73016e^{-18}$
OR	fourteen (13)	$1.73016e^{-18}$
NOR	fourteen (13+9)	$2.92796e^{-18}$
NAND	fourteen (13+9)	$2.92796e^{-18}$
Inverter	Nine(9)	$1.19750e^{-18}$
XOR	Four(4)	$0.53235e^{-18}$
Subtraction	Inverter(5)+OR(1)+AND(2)	$14.50672e^{-18}$
MUX 2:1	Sixty six (66)	$8.78389e^{-18}$
Building block(cell)	MUX(1)+Inv(6)+NAND(2)+NOR(1)	$21.69354e^{-18}$
Building block(cell)with an inverter	Cell(1)+Inverter(1)	$22.49203e^{-18}$
Square root circuit	Cell(43)+Cell with Inv(8)	$7.849977e^{-17}$

XII. CONCLUSION

In this work, we have elaborately discussed concerning a square root circuit, different radicands as inputs and their relevant outputs as square root values which are reliable. The more the stages the less the error. We have studied the time delays both for CMOS/TTL logic and SET based logic and found that SET related circuit at least 3 times faster than first one. Observed that when the number of stages are increased for the square root circuit, more accurate results are produced and which satisfy our expectations. Switching power in the case of SET based devices are the order of O(10^{-17}) which is very low. Single electron-based devices have turned out to

be valuable tools in the research of science and engineering fields. It has drawn one's attention as the SET-devices consume very low power and require small number of nodes so it provides higher order integrity of population (i.e. large-scale integration) and gives the output rate more than O(10^6).

XIII. References

1. 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
2. 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
3. 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
4. Alexander N. Korotkov, "Single-electron logic and memory devices" INT. ELECTRONICS, 1999, Vol. 86, No. 5, 511- 547
5. Casper Lageweg, Student Member, IEEE, Sorin Cot¸ofanˇa, 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
6. Yasuo Takahashi, Akira Fujiwara, Yukinori Ono, and Katsumi Murase "Silicon Single-Electron Devices and Their Applications" Proceedings of The International Symposium on Multiple-Valued Logic January 2000 1:411- DOI: 10.1109/ISMVL.2000.848651
7. K. K. Likharev, "Physics and possible applications of single-electron devices," FED Journal 6 p. 5 (1995).
8. D.V. Averin and K. K. Likharev, "Single-electronics: A correlated transfer of single electrons and Cooper pairs in systems of small tunnel junctions," in Mesoscopic Phenomena in Solids, eds. B.L. Altshuler, P. A. Lee, and R. A. Webb, Elsevier (1991).
9. Millman's Electronic Devices & Ciruits 4th Edition (English, Paperback, Millman Jacob)
10. H. Grabert and M. H. Devoret, Eds., Single Charge Tunneling, Plenum Press, London (1991)
11. Advanced Luminescent Materials and Quantum Confinement: Proceedings of the international Symposium Proceedings volume 99-22, the electrochemical society Inc. USA (1999)
12. Casper Lageweg, *Student Member, IEEE*, Sorin Cot¸ofanˇa, *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
13. Alexander N. Korotkov, Single-electron logic and memory devices INT. ELECTRONICS, 1999, Vol. 86, No. 5, 511- 547
14. K. K. Likharev, "Physics and possible applications of single-electron devices," FED Journal 6 p. 5 (1995).
15. J. Millman and C. C. Halkias; Integrated Electronics- Analog and Digital Circuits and Systems_ McGraw Hill Education; 2 edition
16. A. Venkataratnam and A.K. Goel "CMOS Architectures for NOR & NAND Logic Gates Using Single Electron Transistors "NSTI-Nanotech 2005, www.nsti.org, ISBN 0-9767985-2-2 Vol. 3, 2005
17. P. Vishnu Chaitanya, Pankaj B. Agarwal "Two Input Multiplexer Based on Single-Electronics" Pages 41-44 17th International Workshop on

the Physics of Semiconductor Devices 2013 Springer

18. Dr.C.Arunabala, Digital mode with Single-Electron Transistor (DSET)GSJ: VOLUME 6, ISSUE 7, July 2018 1086 GSJ
19. Yasuo Takahashi, et al. “Multigate single-electron transistors and their application to an exclusive-OR gate” Article in Applied Physics Letters · January 2000
20. C. P. Heij, P. Hadley, J. E. Mooij, “A single-electron inverter” APPLIED PHYSICS LETTERS VOLUME 78, NUMBER 8 19 FEBRUARY 2001
21. BananiTalukdar1 , Dr.P.C.Pradhan2 and Amit Agarwal3” DESIGN OF DIFFERENT DIGITAL CIRCUITS USING SINGLE ELECTRON DEVICES” Advances in Materials Science and Engineering: An International Journal (MSEJ), Vol. 3, No. 1, March 2016
22. P. Vishnu Chaitanys et al, “ Two Input Multiplexer based on Single-Electronics” 17th International Workshop on the Physics of Semiconductor Devices 2013, Editors: **Jain**, Vinod Kumar, **Verma**, Abhishek (Eds.)

BIOGRAPHY

Dr. Anup Kumar Biswas is an Assistant Professor in Commuter Science and Engineering in Kalyani Govt. Engineering College (India). He has engaged in teaching and research activities since the last 16 years. His Specialization field is Single Electron Device. Biswas has published several papers in various national, international conferences and journals.

Cite this article as :

Dr. Anup Kumar Biswas, "Finding out square root of an integer number using Single Electron Transistor ", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 6, pp. 96-110, November-December 2021. Available at doi : https://doi.org/10.32628/IJSRSET218610
Journal URL : https://ijsrset.com/IJSRSET218610