**Comparative Analysis of CNFET based Signed bit Adders and Multipliers**

**//authors**

***Abstract*— In this paper, the implementation of the NAND based Full Adder, 16-bit Carry Save Adder, 8-bit Ripple Carry Adder, and 4\*4-bit Vedic Multipliers are performed using CNTFETs on the 16 nm scale. Due to Moore's law, further optimization is not possible in CMOS technology, so Carbon Nanotube Technology (CNT) serves as an excellent alternative. Comparing all the CNTFET based circuits with CMOS based circuits implemented in this paper, CNTFET based circuits has an optimisation of 13% on an average in delay and 98.83% on an average in power. The results are promising for building large scale circuits in future.**

***Keywords*— A****DDER** · **MULTIPLIER** · **CNFET** · **CMOS** · **CNTFET INTERCONNECT**

I. INTRODUCTION

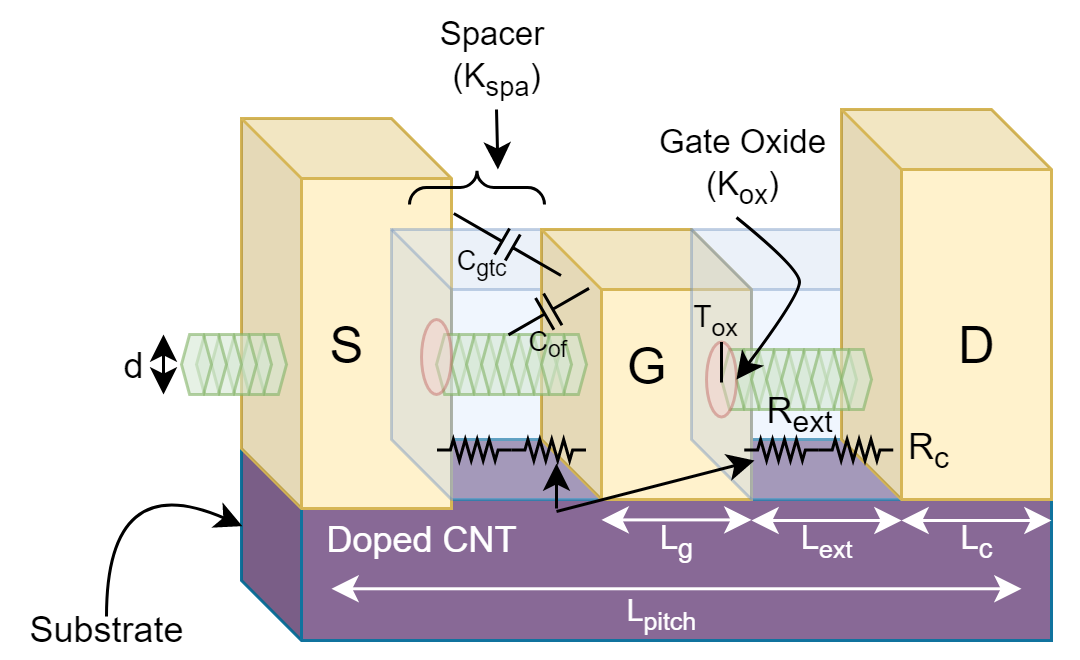
An adder is a digital circuit which performs the addition of numbers. These are used in used in the arithmetic logic units or ALU in computers and processors. In processors, they are used to calculate addresses, increment and decrement operators, table indices and similar operations [1]. MAC (Multiply Accumulate) is used to carry out specific arithmetic operations like Multiply and Accumulation. In today’s high-speed technological world, high speed and low power MAC are used to perform fundamental operations and are used widely in microcontrollers and microprocessors etc. [2]. So, designing optimised MAC is part of trending ongoing research. Generally, the standard architecture of MAC units consists of a product calculator or multiplier, summer or adder and a data accumulation register. The multiplier unit here performs functions like computation of partial products, trimming and cutback of these generated partial products, next stage adding carry of precious stage and further addition of the propagated carry. It uses the PIPO (Parallel Input Parallel Output) shift register. After data accumulation, the generated results are stored in these registers [3].

For complex operations and designing efficient algorithms, Adders and MAC form the fundamental units of any digital circuit involving large scale and heavy processing of data. Since there is always constant room for improvement, optimisation plays a crucial role in overall enhancement of any circuit. For any circuit to be optimised, we need to optimise every fundamental component in the circuit. Further optimisation is not possible in the conventional CMOS circuits due to Moore’s Law [4]. So, there is a lot of ongoing research on alternatives to CMOS in which we can perform further optimization, miniaturisation and obtain results better than CMOS in terms of performance, area, delay and power. One such alternative is CNT (Carbon Nanotube Technology). CNT is potentially the best alternative for CMOS and has many useful properties as well. Until now, half adders, full adders, ternary half adders and ternary full adders have been implemented using CNFETs and have proved to be far better than those implemented using CMOS Circuits [5, 6, 7, 8, 9]. The current major setback of CMOS is that while reducing the channel length, threshold voltage is also reduced which also leads to increase in leakage power. This setback is not applied to CNTFET. The major reason why CNTFET has a promising future is because we can increase the threshold voltage at 10 nm and beyond the channel length. This also ensure no increase in leakage power [10].

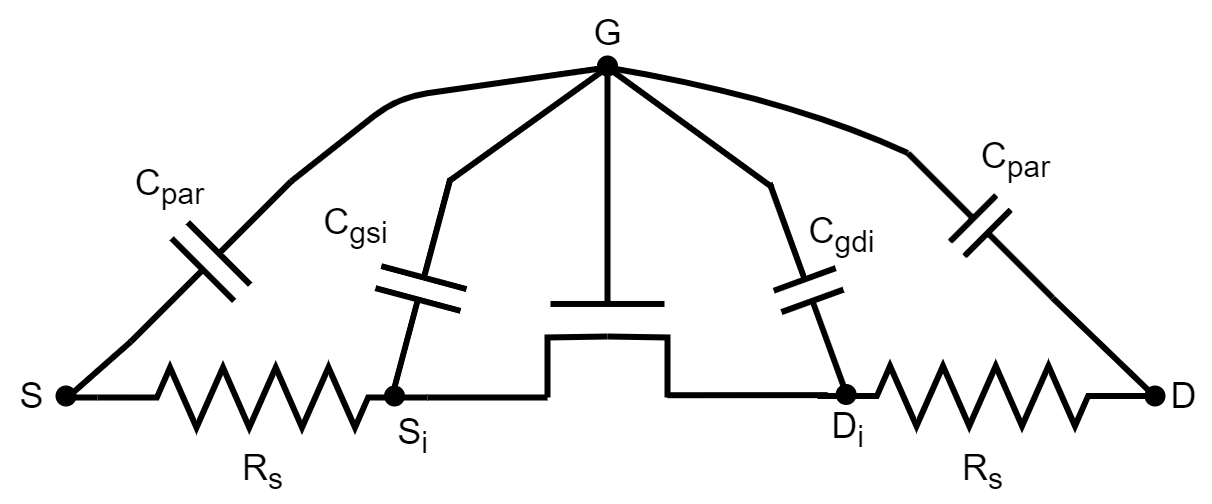
To build CNFET based circuits, Carbon Nanotube Interconnects are very useful. Earlier copper interconnects were used but it became a major hurdle to on-chip communication due to high resistivity and electromigration. Hence CNFET Interconnects are used which are proven to have reduced delay and optimised overall performance of circuits. Carbon nanotube interconnects refer to the use of carbon nanotubes in the interconnects between the elements of an integrated circuit. We have implemented NAND Based Full Adder, 16-bit Carry Save Adder, 8-bit Ripple Carry Adder and 8\*8-bit Vedic Multiplier using CNT Interconnect. In this paper, we have focused on comparing different bit-based adders and Multiplier units made with CMOS to CNTFETs and proved that performance, area, delay and power are well optimised when CNTFETs are used.[11]

II. CNTFET WORKING AND STIMULATION SETUP

For stimulation purpose, we have used VS-CNFET model [12]. A representative CNFET structure modelled in the VS-CNFET is shown in Fig [1a]. It consists of a cylindrical gate-all-around structure with heavily doped source/drain extensions. The body terminal is assumed to have no effects since the CNTs sit on a thick insulator (e.g., SiO2). Hence, the VS-CNFET is a three-terminal transistor model. Vds(i) and Vgs(i) denote the voltages across the gate to the (internal) drain and the (internal) source, respectively, as shown in the corresponding transistor schematic in Fig. 1b. Due to the symmetry of CNT’s conduction band and valence band, P-type CNFETs and N-type CNFETs are completely symmetric (characteristics of I-V and C-V are the same for p-CNFETs and n-CNFETs given the same |Vgs| and |Vds|).



*Fig [1a] CNTFET device structure*



*Fig [1b] CNTFET device in 2D*

The Stanford Virtual-Source Carbon Nanotube Field-Effect Transistor model (VS-CNFET) is a semi-empirical model that describes the current-voltage (I-V) and capacitance-voltage (C-V) characteristics in a short-channel MOSFET with CNT as the channel material. The model captures dimensional scaling properties and includes parasitic resistance (CNT-metal contact and doped extensions), parasitic capacitance (mate-to-metal coupling capacitance and metal-CNT fringe capacitance), and tunnelling leakage currents (direct source-to-drain tunnelling and gate-to-drain junction band-to-band tunnelling).

|  |  |
| --- | --- |
| INPUT | VALUE |
| Type | 1 |
| s | 10 nm |
| w | 1 um |
| Lg | 11.7 nm |
| Lc | 12.9 nm |
| Lext | 3.2 nm |
| d | 1.2 nm |
| tox | 3 nm |
| kox | 23 |
| kcnt | 1 |
| ksub | 3.9 |
| kspa | 7.5 |
| Hg | 20 nm |
| Esfd | 0.258 |
| Vfb | 0.15 |
| Geomod | 1 |
| Rcmod | 1 |
| Rs0 | 3.3 e3 |
| SDTmod | 1 |
| BTBTmod | 1 |
| temp | 25 |

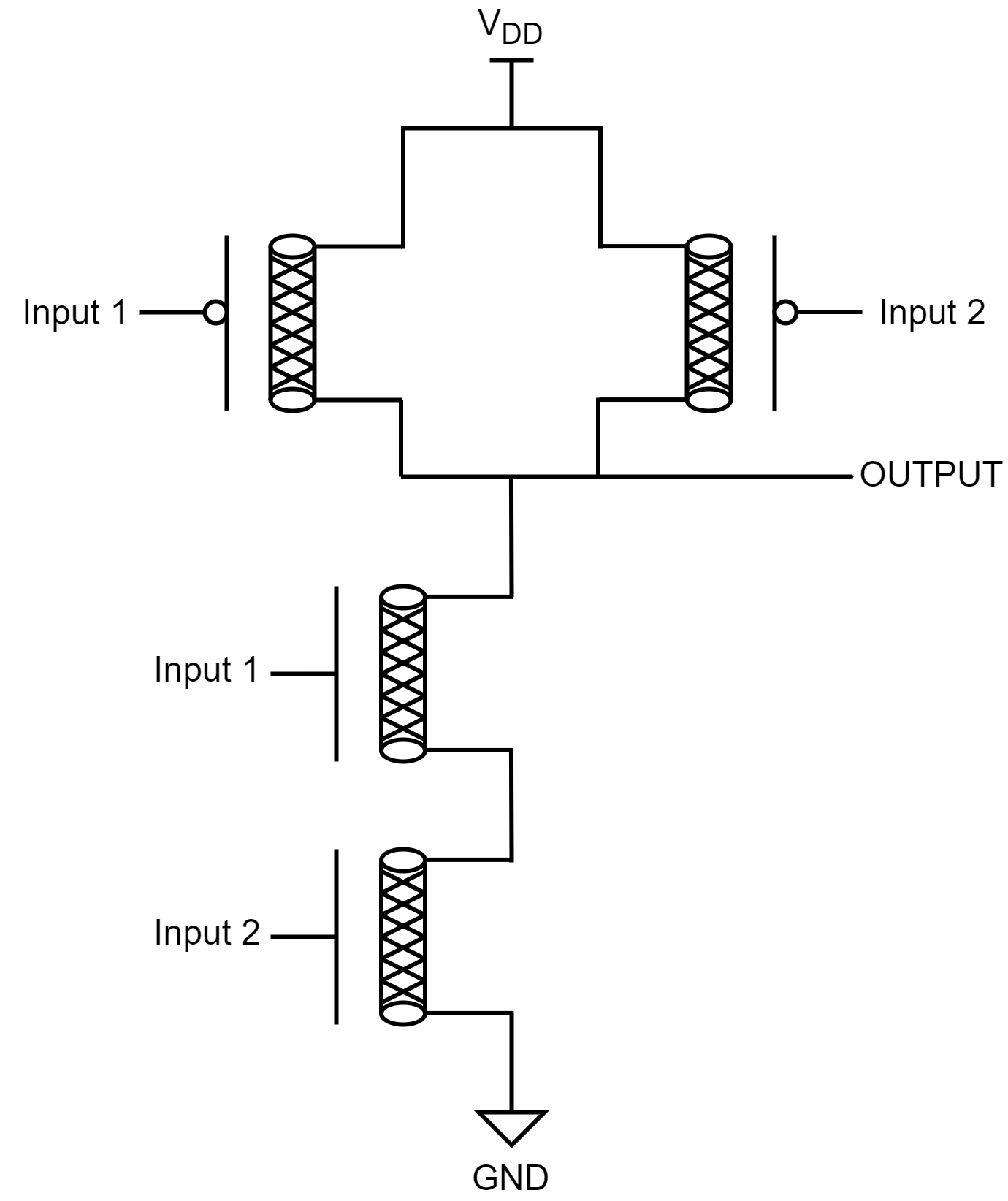
**Table [1]** Parameters of VS-CNFET Model

III. BASIC BUILDING BLOCKS

For CNTFET based circuits, it is essential to build the building blocks of the circuits. i.e., CNTFET based NAND and NOR. NAND and NOR are universal gates and thus can be used to build any other gates. Since these are basic fundamental blocks, we first implemented them and using this, we further implemented other CNTFET based circuits. We can easily fabricate NAND and NOR logic gate circuits and we can make any logical Boolean expression using these gates.

1. NAND Gate

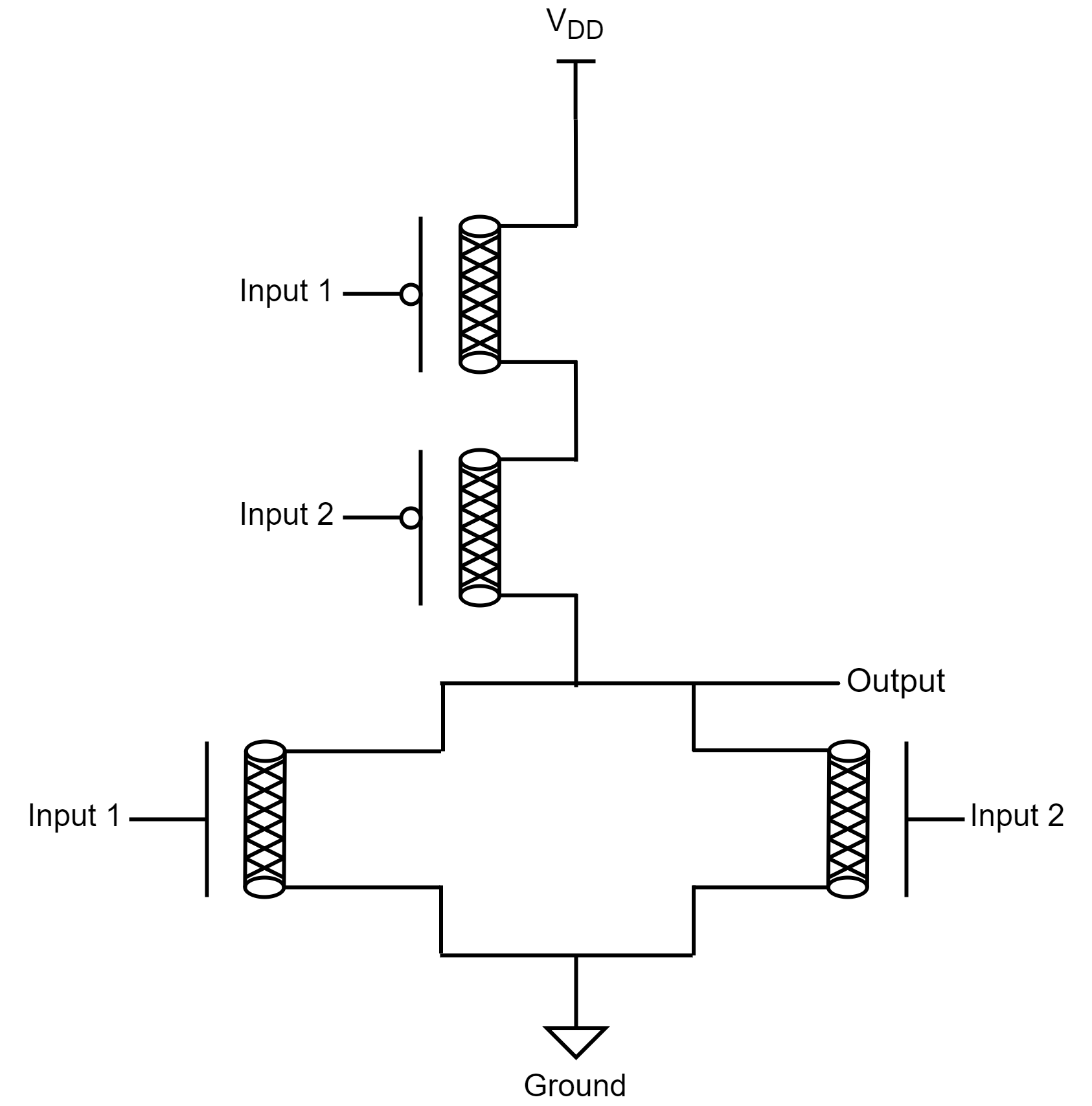
NAND is a logic gate which produces an output which is false only if all its inputs are true. Below given figure shows the circuit of CNTFET based NAND.



*Fig [2] CNTFET based NAND*

1. NOR gate

The output of the gate is A HIGH output if both the inputs to the gate are LOW and if one or both input is HIGH, a LOW output results.

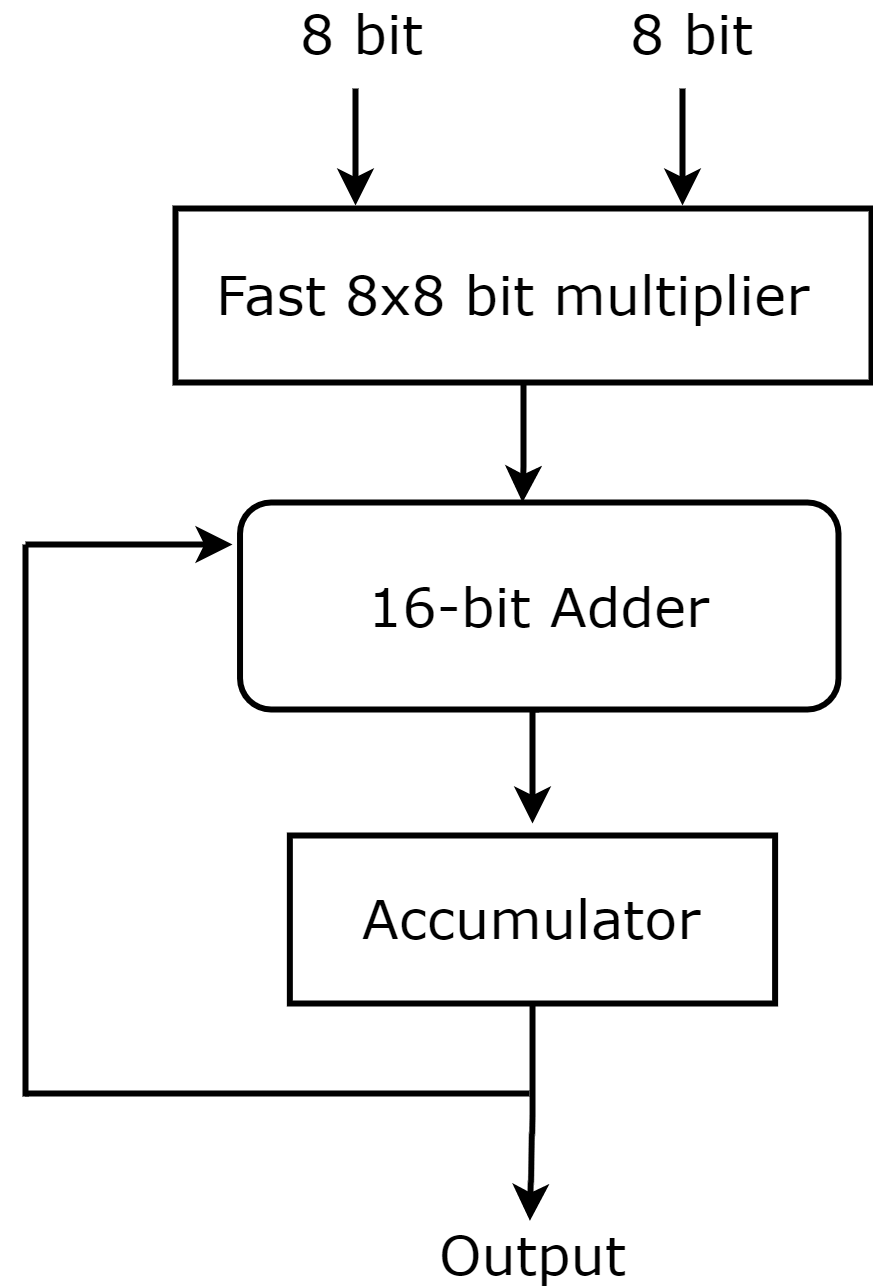


*Fig [3] CNTFET based NOR*

IV. PROPOSED WORK

*A. 8-bit MAC Unit*

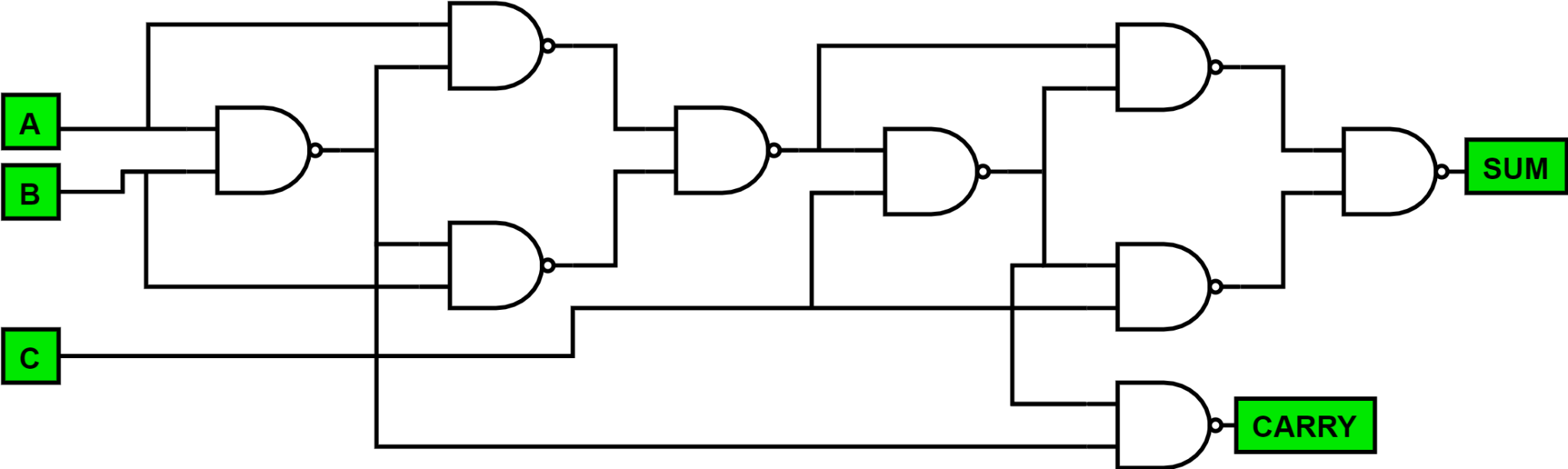
MAC unit stands for multiply and accumulate unit. It performs multiplication and addition operations. The first stage is the multiplier stage. In this stage, MAC performs multiplication of given numbers and forwards into the second stage. The second stage is the accumulator stage in which MAC performs the addition of that product to an accumulator. MAC has a wide range of applications in digital signal processing (DSP) like filtering, convolution, averaging and power estimation.[13] Other than that, two main bottlenecks play an important role in a MAC, partial product reduction and accumulator. These two factors control the speed of a MAC unit.[14] The circuit of the 8-bit MAC unit is shown below:



*Fig [4] 8-bit MAC unit*

*B. NAND based Full Adder*

Full adder is the combinational circuit that executes the arithmetic addition of three input bits and brings out two output variables ‘sum’ and ‘carry’. A NAND gate is the universal gate and we can make any logic design using the NAND gate [15]. The NAND based full adder circuit is shown below:

****

*Fig [5] NAND Based Full Adder*

We can determine logical expressions for ‘sum’ and ‘carry’ of the above circuit using the truth table of the full adder. Considering the truth table of the full adder,

Sum = A’ B’ Cin + A’ B Cin’ + A B’ Cin’ + A B Cin

= (A ⊕ B) ⊕ Cin

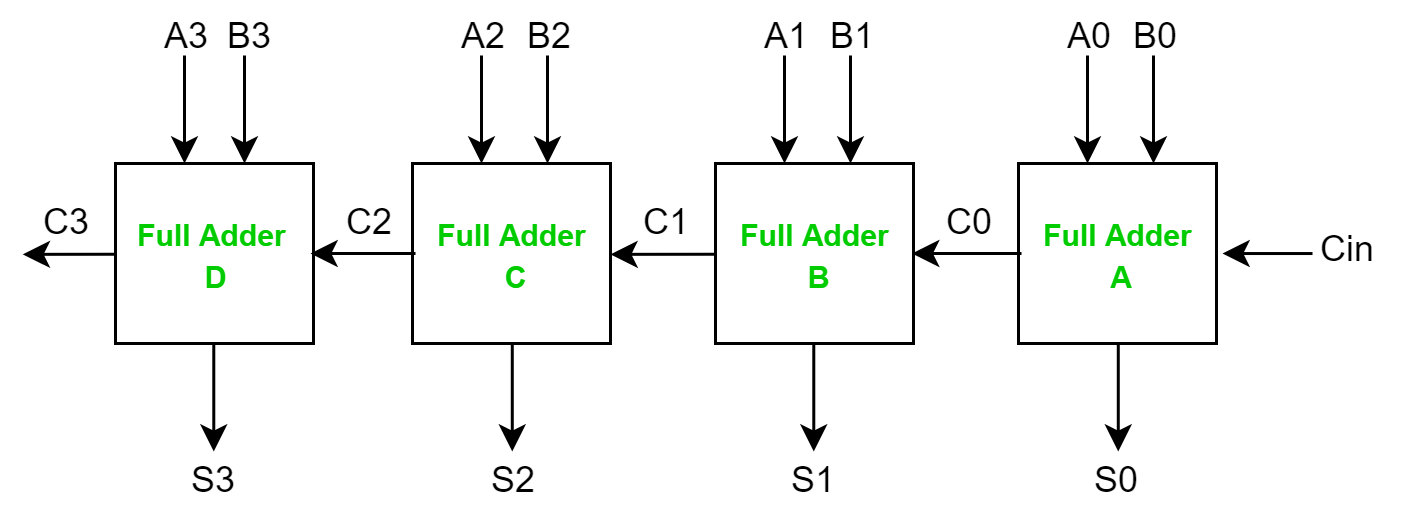
Cout = A’ B Cin + A B’ Cin + A B Cin’ + A B Cin

= AB + BCin + CinA

= AB + Cin (A ⊕ B)

*C. Ripple Carry Adder*

To add N-bit binary numbers, it is possible to cascade N numbers of full adder circuits in parallel, in which the input of each adder is the output of the previous adder. It is called Ripple Carry Adder because each generated carry bit of every stage ripples into the next stages. So, when the carry input bit is available, it triggers the full adder for the operation. In this Ripple Carry Adder, there is the time elapsed between the carry input signal and carry output signal which is called carry propagation delay [16]. The circuit implementation of the 4-bit ripple carry adder is shown below:

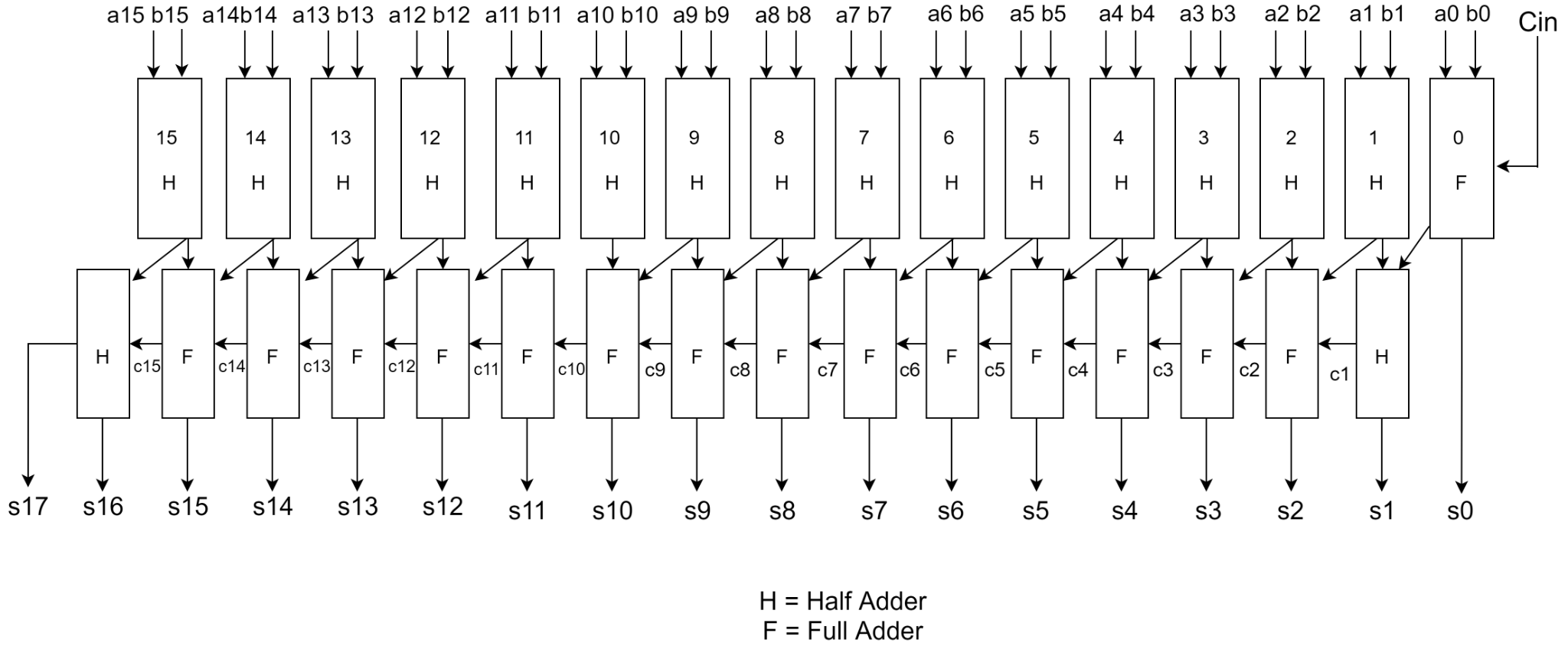


*Fig [6] Circuit of 4-bit Ripple Carry Adder*

Considering the above circuit, the sum S0 will be valid after the propagation delay of adder A. Similarly, the sum S3 will be valid after the joint propagation delay of adders 1 to 4. So, in the worst case, the carry has to propagate a pathway from the LSB to the MSB. If we make wider and wider adders, this carry propagation delay becomes a huge problem. Due to this reason, this adder becomes extremely slow.

*D. Carry Save Adder*

Carry Save Adder is a digital adder that is used to calculate the sum of n-bit binary numbers. The difference between this adder and other digital adders is that this adder outputs two or more numbers and we can achieve the final sum by adding these numbers together. It is used in binary multipliers as binary multipliers include the summation of two or more binary numbers [17, 18]. The circuit of 16-bit Carry Save Adder is shown below:



*Fig [7] CNTFET Circuit of 16-bit Carry Save Adder*

The implementation of 16-bit Carry Save Adder includes 16 full adders. Each full adder calculates the single sum and carry-bit of three input numbers called partial sum and shift carry, respectively. In the above circuit, let’s have a, b and c as input numbers and generate partial sum and shift carry:

Partial sum (ps) = a ⊕ b ⊕ c

Shift carry (sc) = ab + bc + ac

Now, the final summation stage comes into play. In this stage, we first do one unit left-shift to the carry sequence, then put 0 in the partial sum at the MSB. Finally, by using a ripple carry adder, we can calculate the final summation.

*E. Vedic Multiplier*

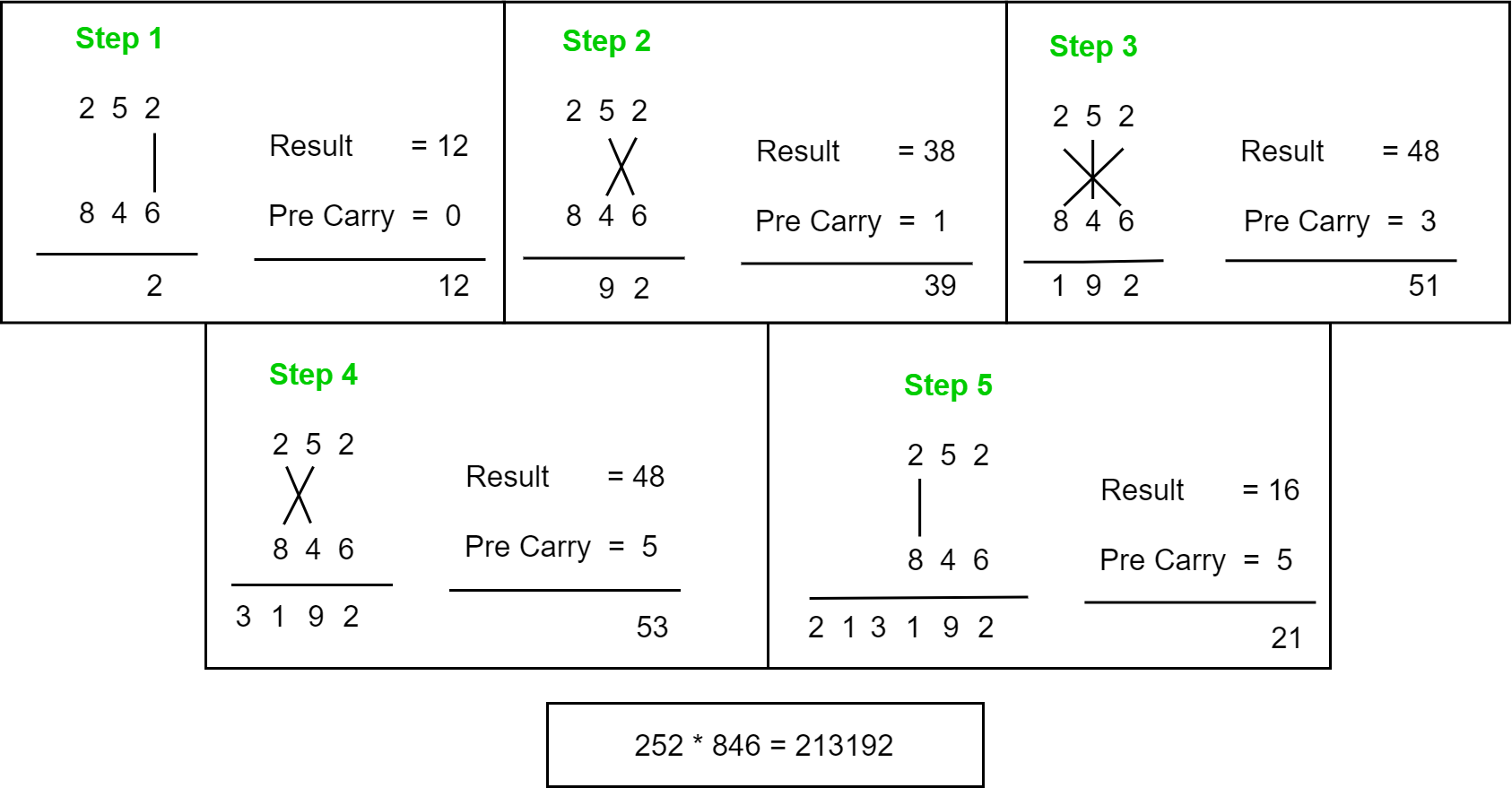
Urdhva tiryagbhyam is one of the most traditional techniques of Vedic maths. In English, we call it ‘vertically and crosswise’. With the help of this technique, we can multiply any two numbers within a single line and we can get the answer. Let’s consider we have two 2-digit numbers (p1x+q1) and (p2x+q2) and we want to multiply them using this sutra and the expected answer will be p1p2x2  + (p1q2+q1p2) x + q1q2 [19].

Step1: In this step, we’ll calculate the coefficient of x2 and it’ll be the vertical multiplication of the coefficients of x which are p1 and p2.

Step2: This step will calculate the coefficient of x and it’ll be obtained by performing crosswise multiplication i.e., p1 multiply q2 in the summation of p2 multiply q1.

Step3: The last step will calculate the constant value and it’ll be achieved by performing vertical multiplication of constant values i.e., q1 multiply q2.

To better understand this scheme, let’s calculate the multiplication of two decimal numbers (252 x 846) by using this sutra and the method shown in the figure below. In the initial stage, the pre-carry bit will be zero. Now, in each stage, the LSB will be the output bit and the MSB will be the propagate bit which will be the pre-carry bit.

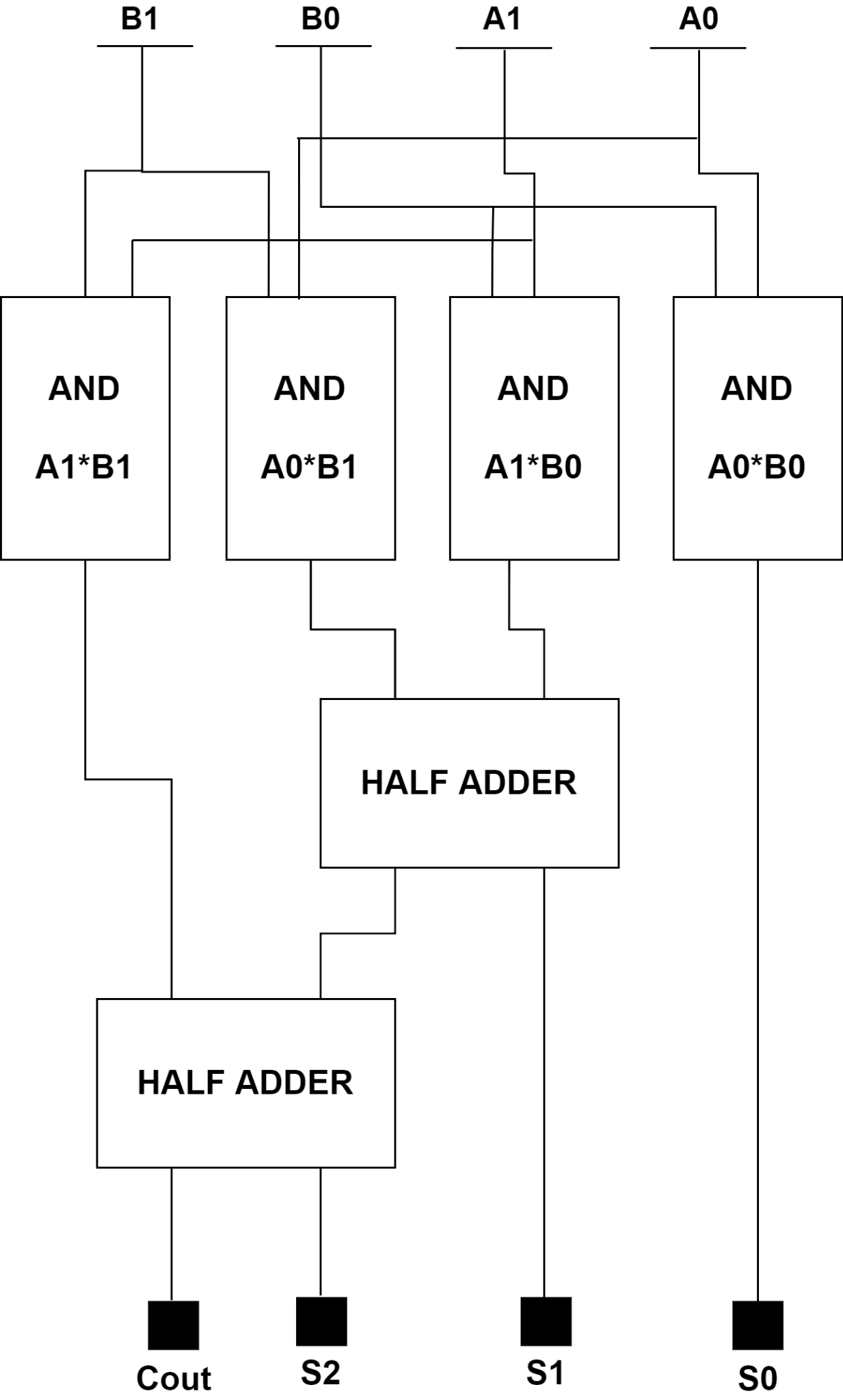


*Fig [8] Steps for Vedic Multiplication*

To implement an 8\*8-bit Vedic multiplier, we had first implemented a 2\*2-bit Vedic Multiplier. Using 2\*2-bit Vedic Multiplier, we implemented 4\*4-bit Vedic Multiplier. And finally using 4\*4-bit Vedic Multiplier, we implemented 8\*8-bit Vedic Multiplier. We scaled the multiplier step by step.

*i) 2x2 bit Vedic multiplier*

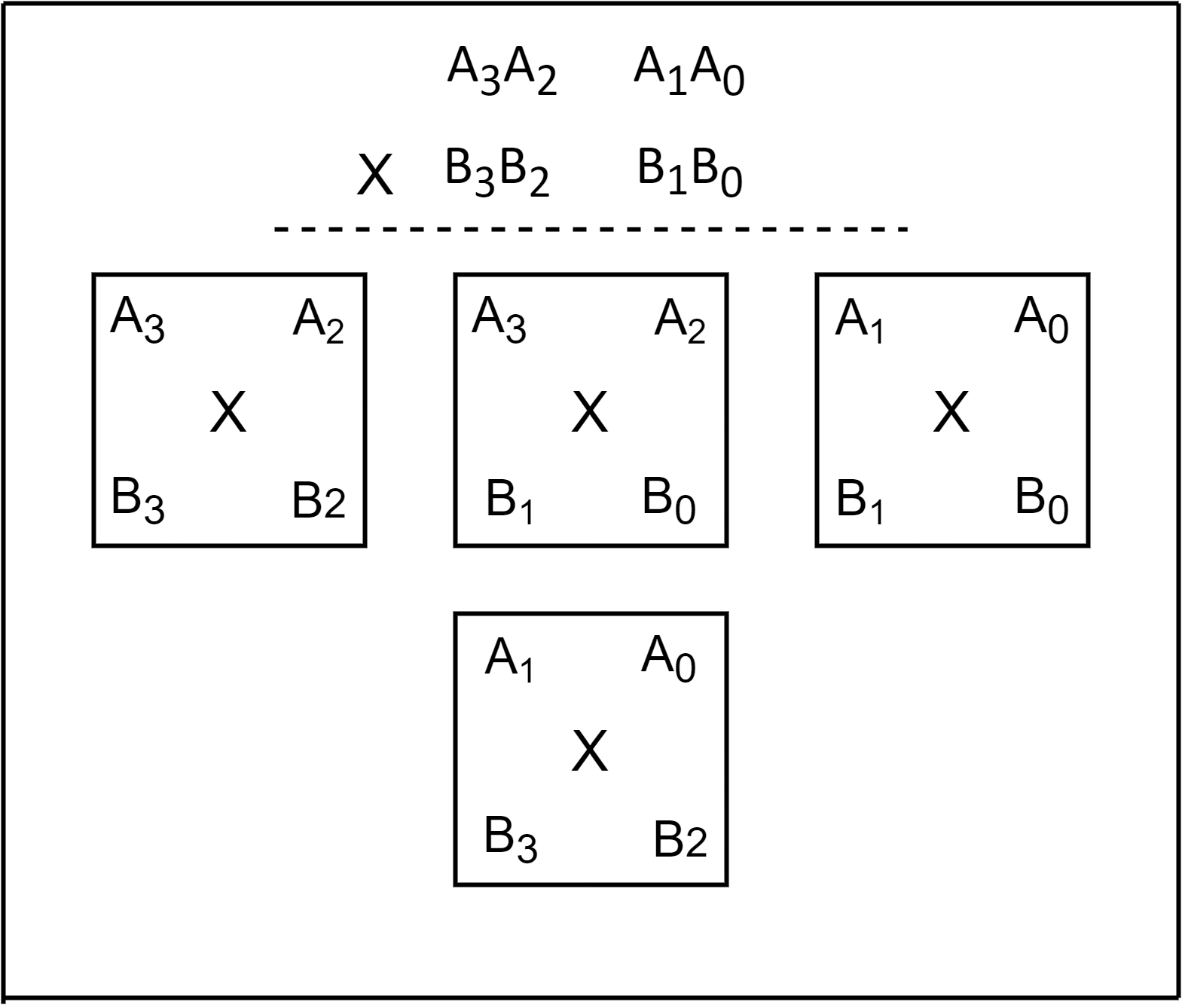
As we mentioned earlier in this paper, we can multiply any two numbers within a single line with the ‘vertically and crosswise’ technique. We have implemented a 2x2 bit Vedic multiplier with the help of four input AND two half adders. The block diagram and implementation circuit diagram is shown below. Here, our architecture of 2x2 bit Vedic multipliers is matching the array multipliers. So, in terms of delay, this multiplier doesn’t make a significant improvement. But as we go for higher bits, Vedic multipliers can be brought into play.



*Fig [9] Diagram of 2x2 bit Vedic Multiplier*

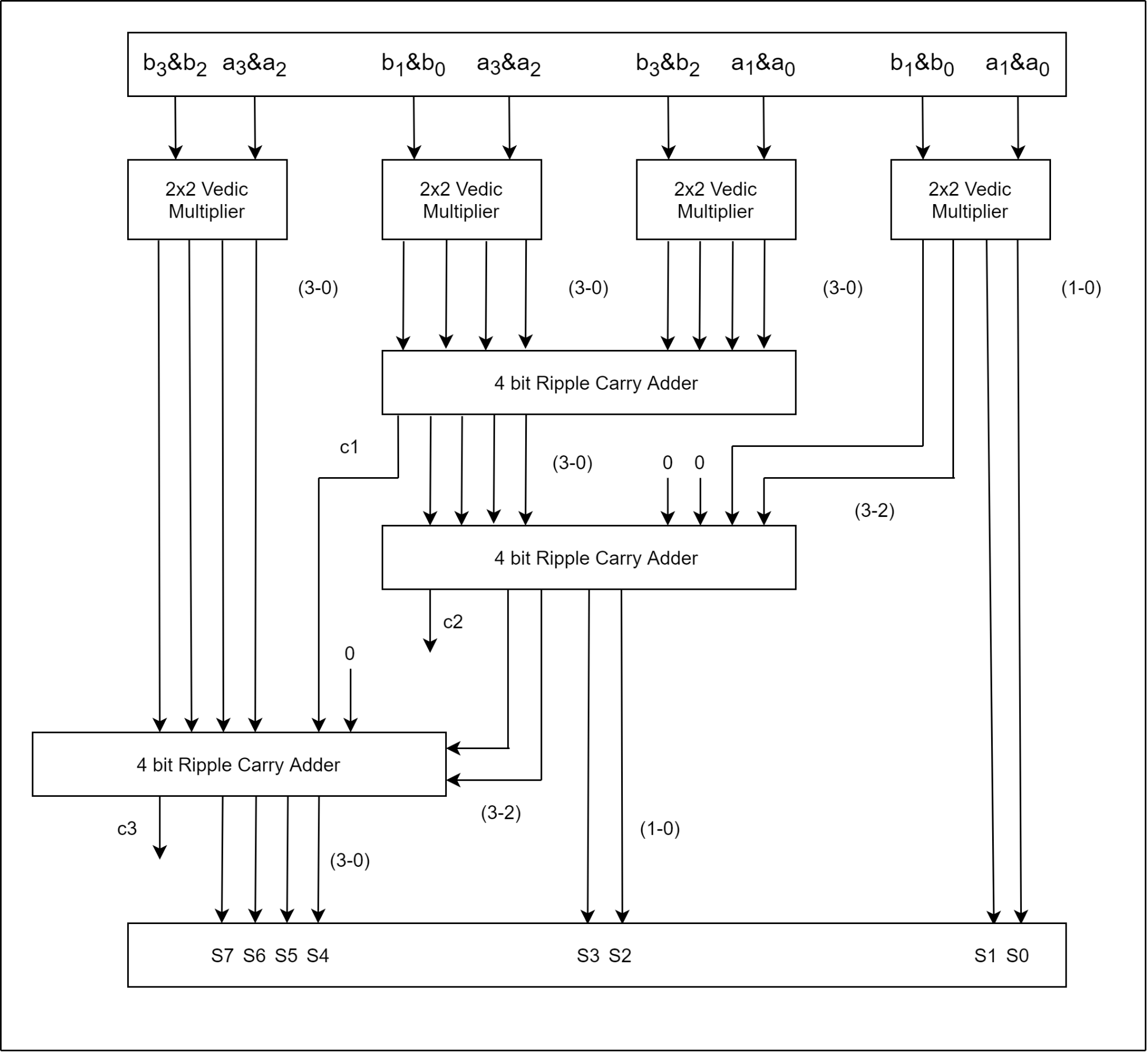
*ii) 4x4 bit Vedic multiplier*

We can make a 4x4 bit Vedic multiplier using 2x2 bit Vedic multipliers [14]. Let’s understand this architecture. Let’s say we want to multiply two 4-bit numbers a = a3a2a1a0 and b = b3b2b1b0. Now, our final product will be 8-bit, say s = s7s6s5s4s3s2s1s0. We can divide these 4-bit numbers into 2x2-bit numbers and ap ply the crosswise multiplication using 2-bit Vedic multipliers. The logic diagram is given below [20]:



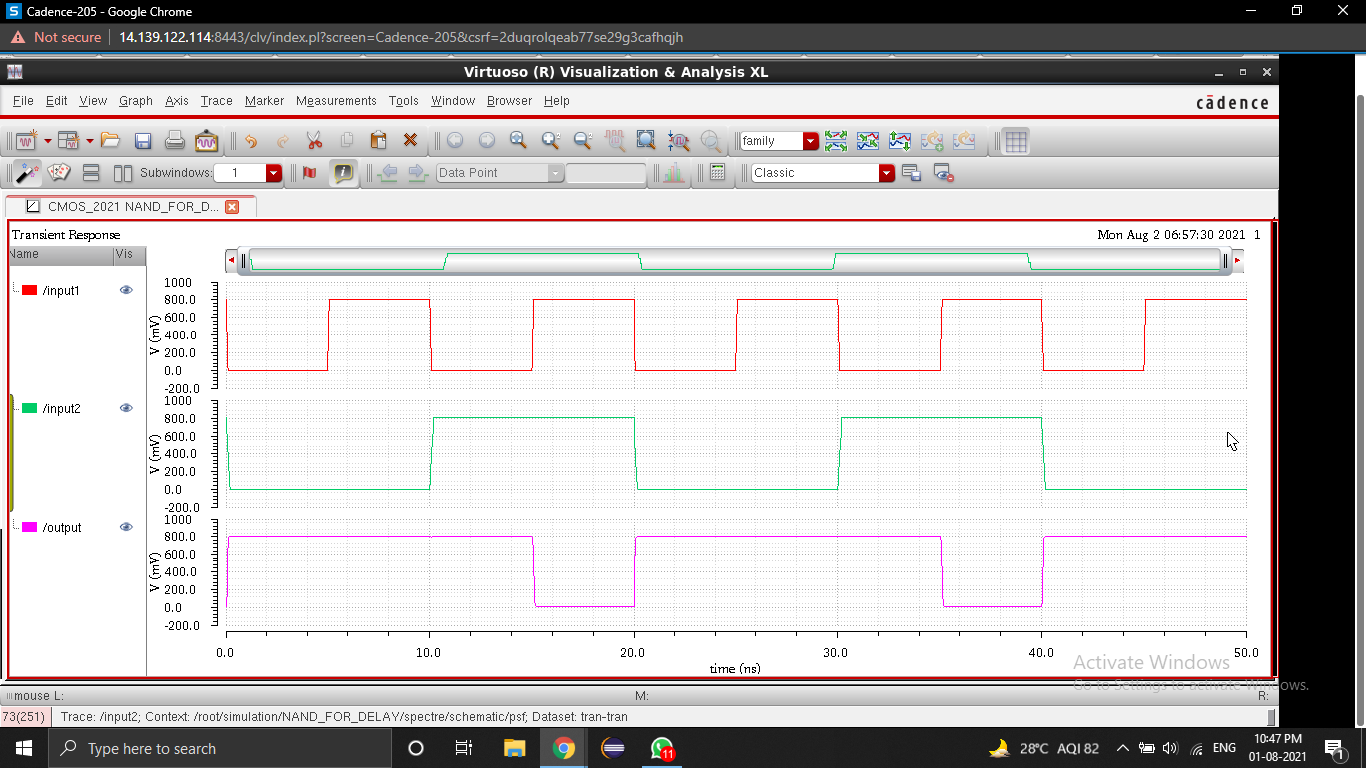
*Fig [10] Logic of 4x4-bit Vedic Multiplier*

From the above circuit, we can say that we’ll need four 2x2 Vedic multipliers to implement a 4x4 Vedic multiplier. Also, we have to perform a summation of generated partial products which will be achieved by three 4-bit ripples carry adder s which we discussed earlier. To understand this concept, the circuit diagram of the 4x4 Vedic multiplier is shown below:

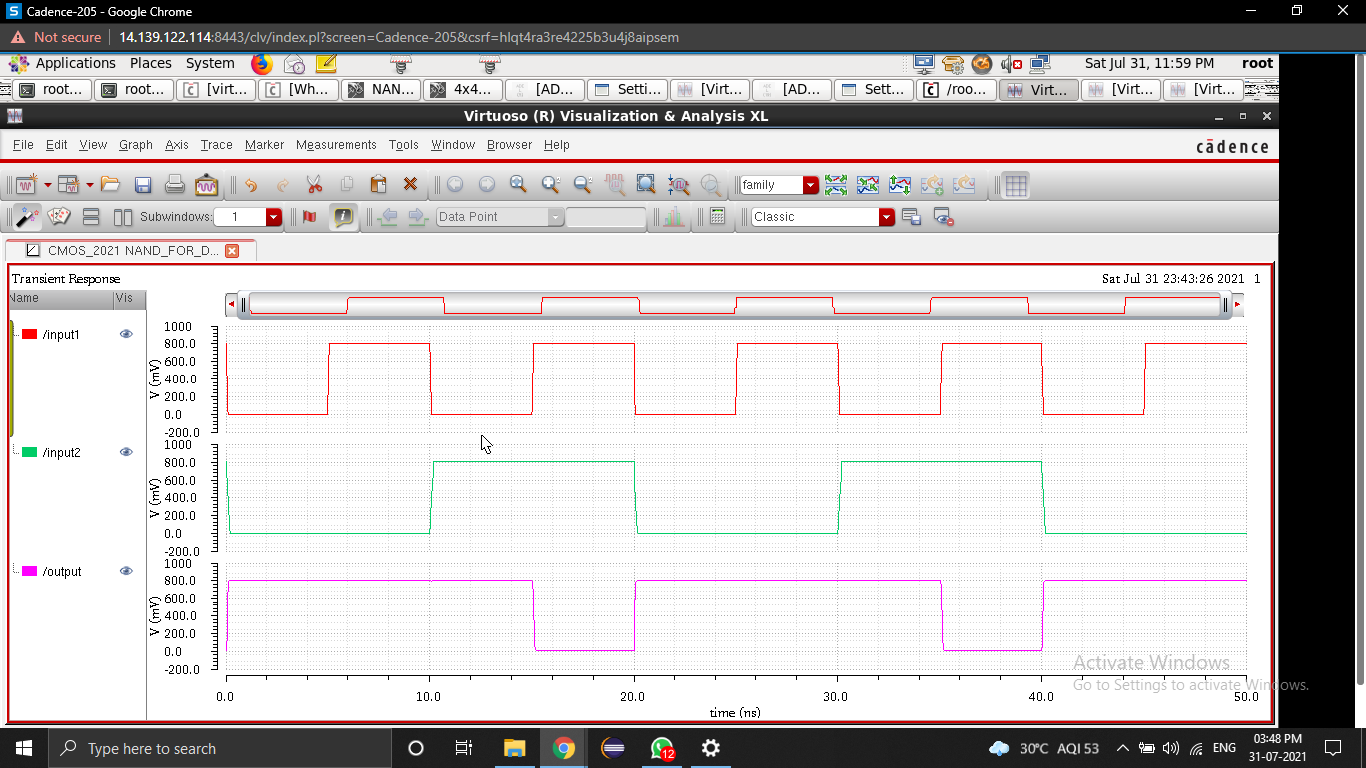


*Fig [11] Diagram of 4x4- bit Vedic Multiplier*

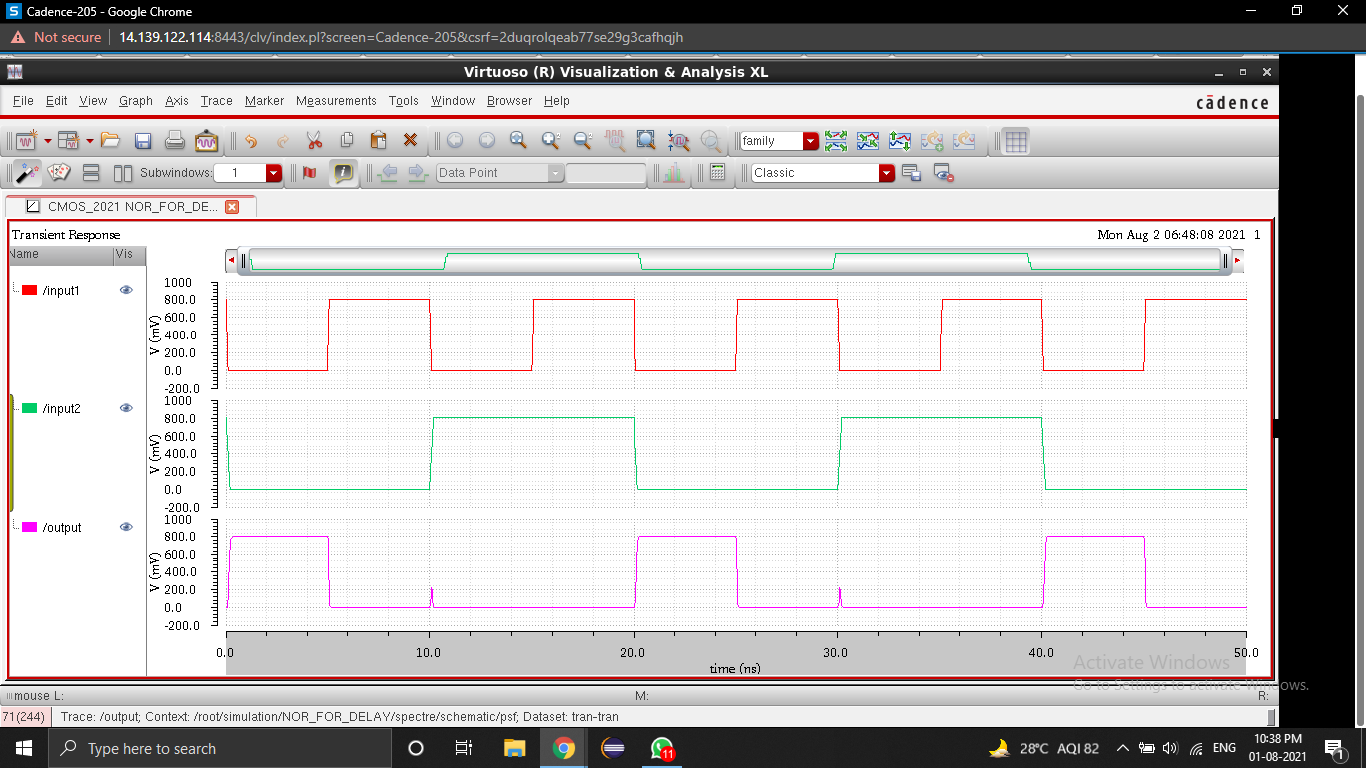
V. SIMULATION RESULTS



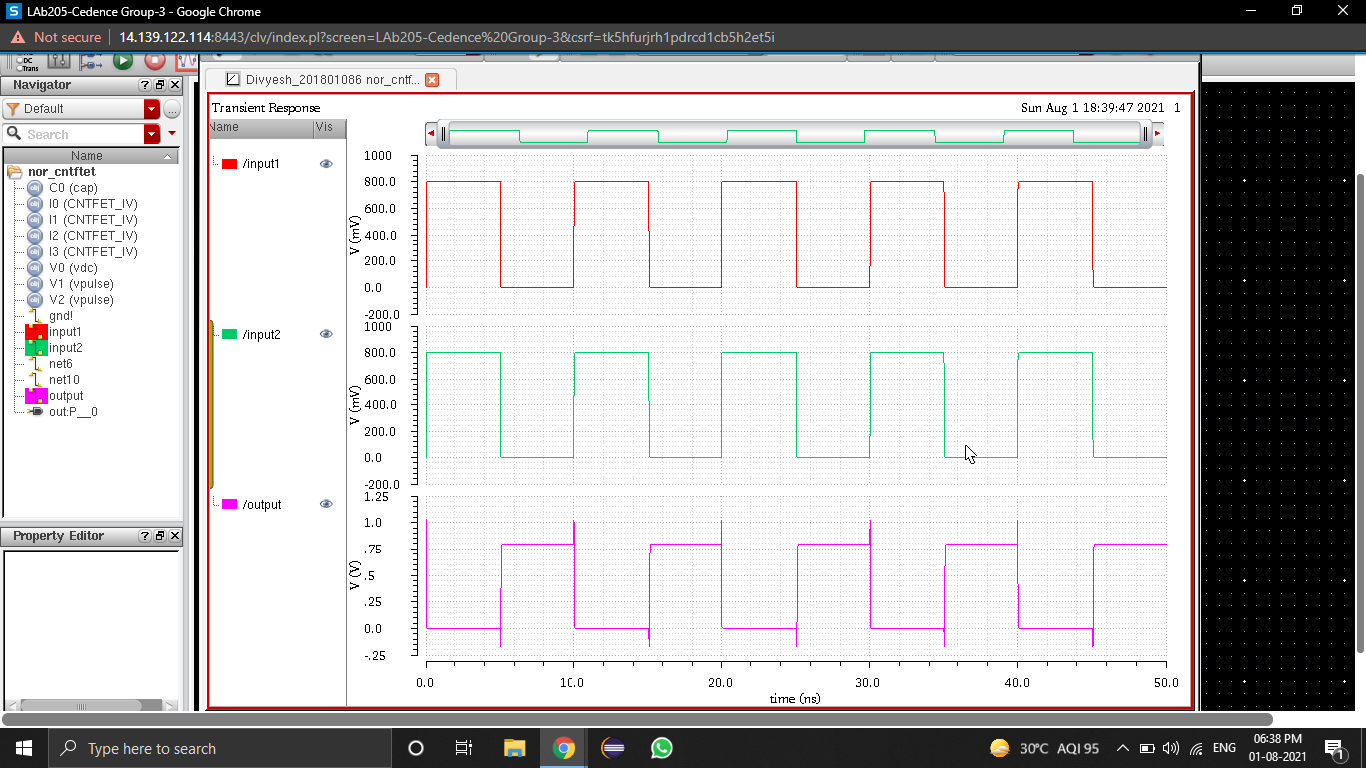
*Fig [12] Stimulation result of CMOS based NAND*



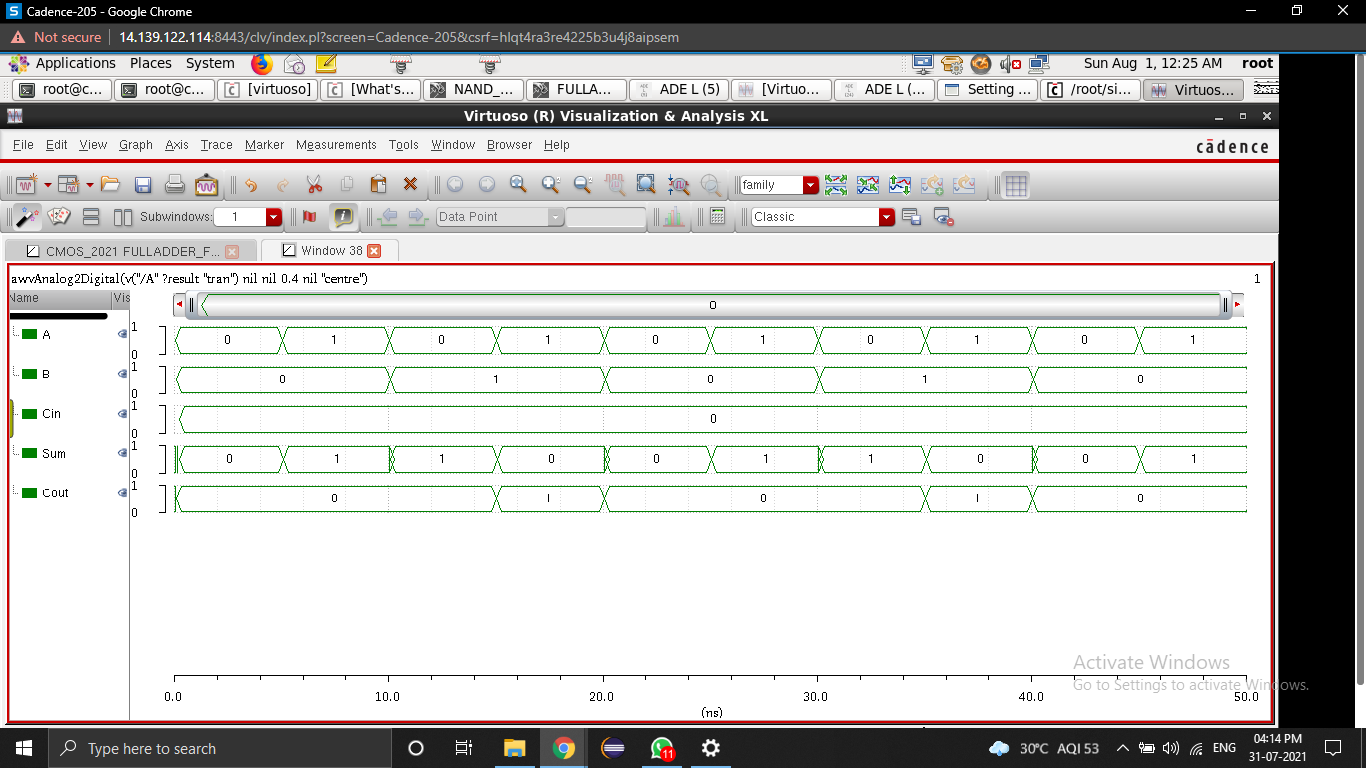
*Fig [13] Stimulation result of CNTFET based NAND*



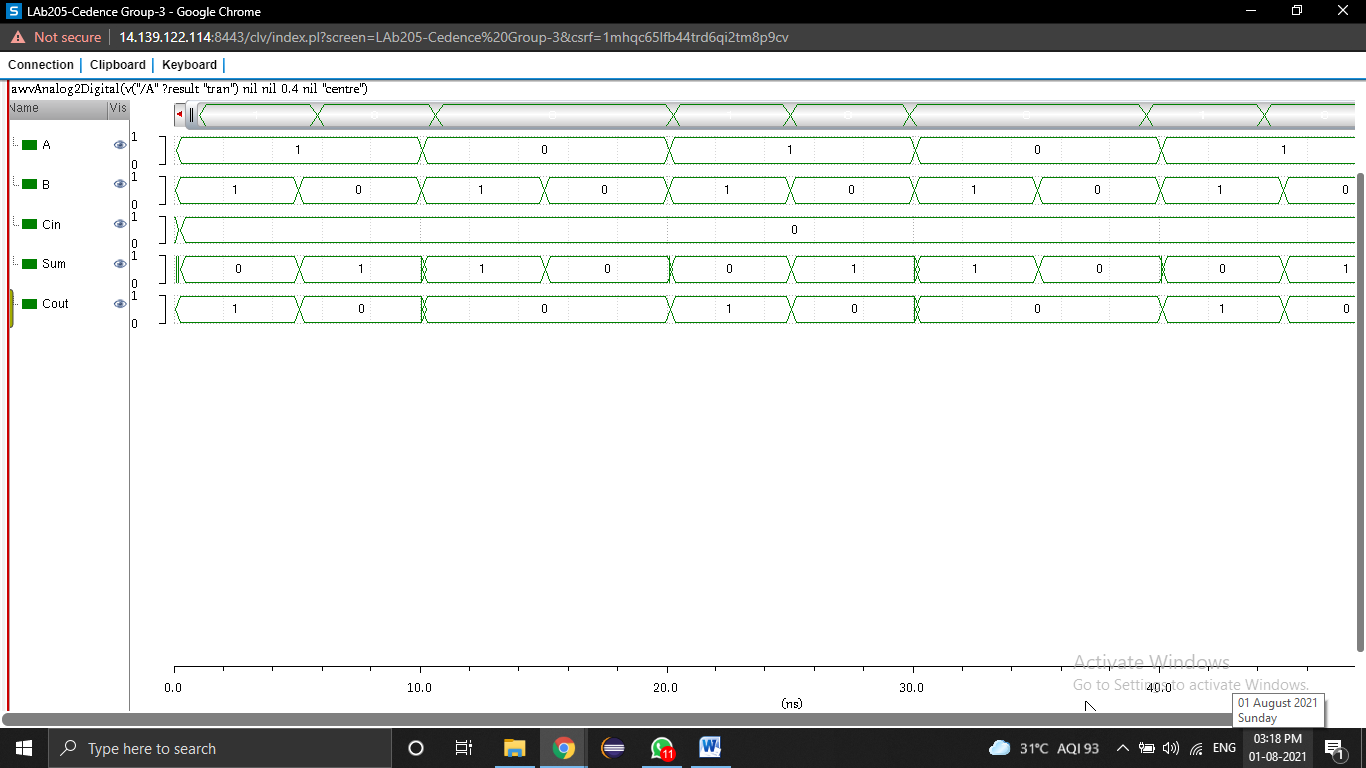
*Fig [14] Simulation result of CMOS based NOR*



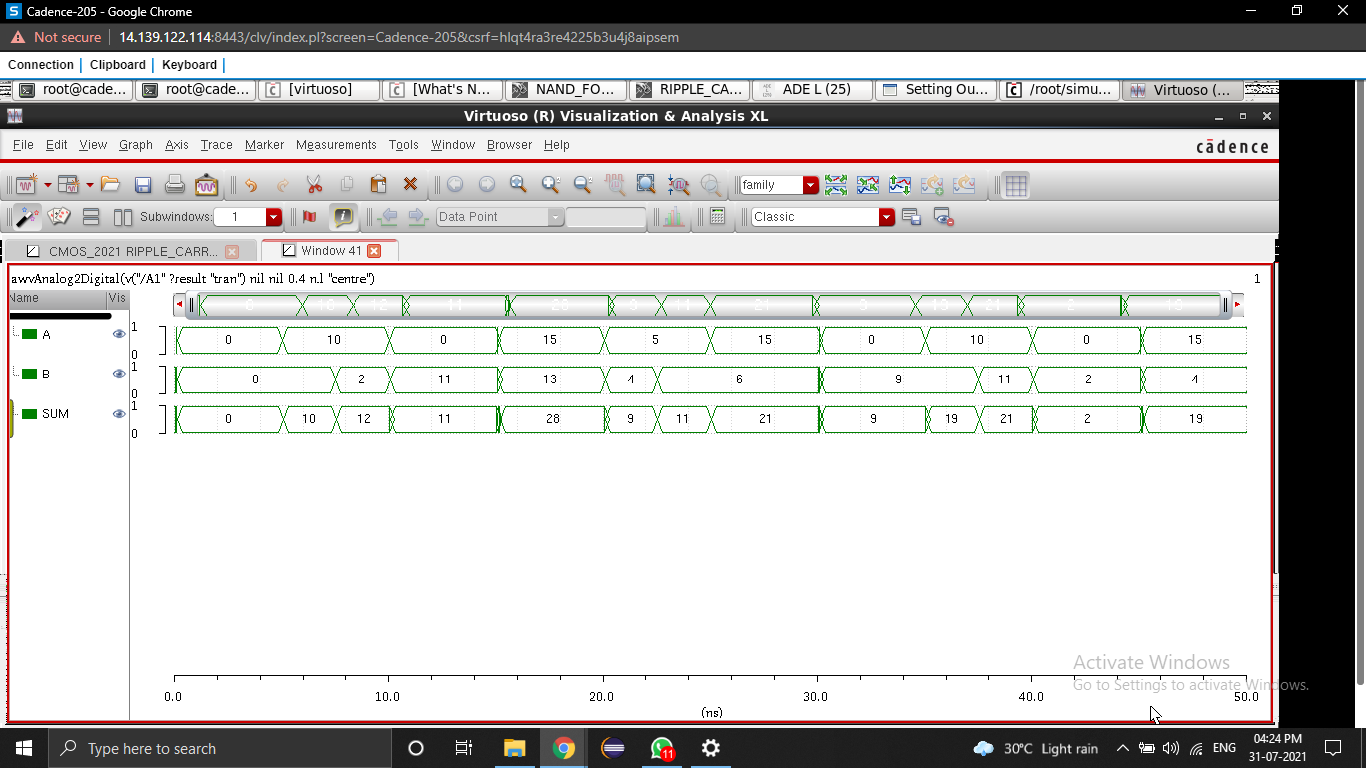
*Fig [15] Stimulation result of CNTFET based NOR*



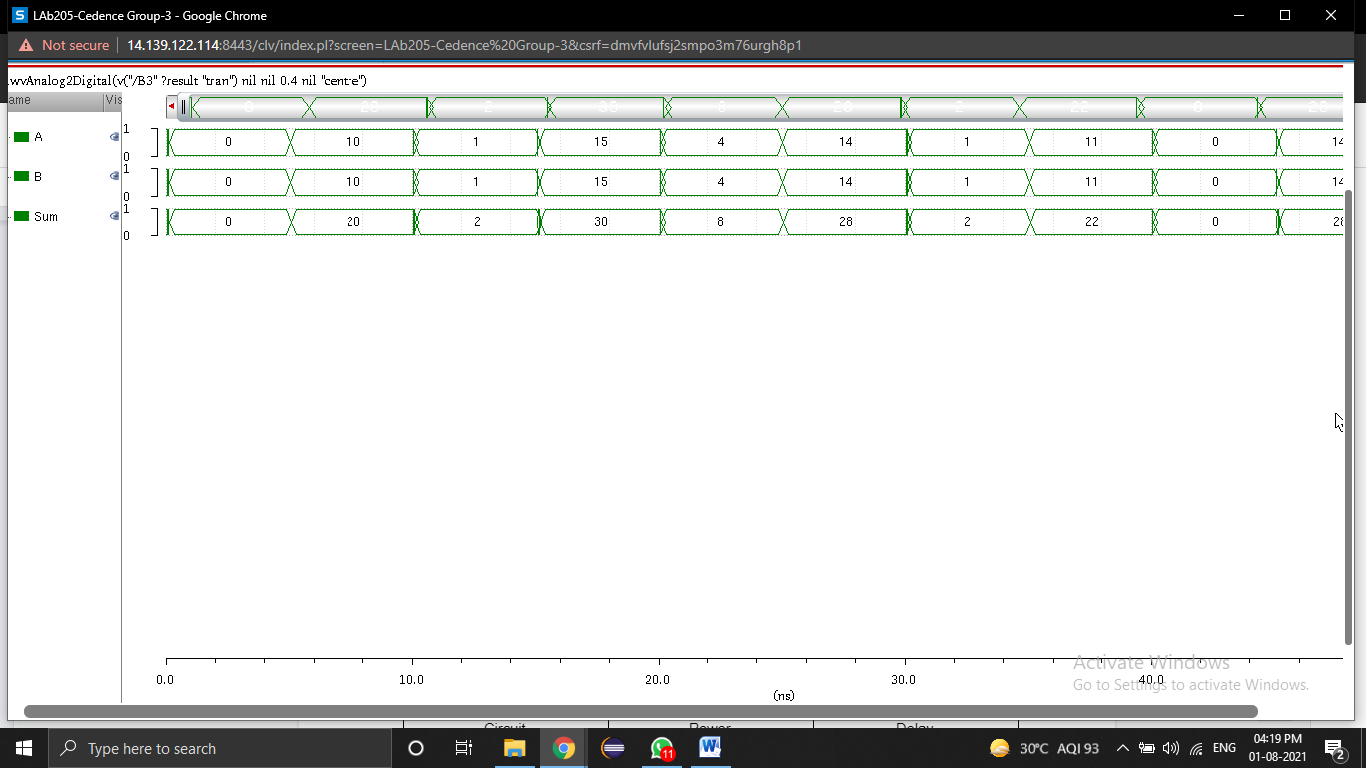
*Fig [16] Simulation results of CMOS NAND Based Full Adder*



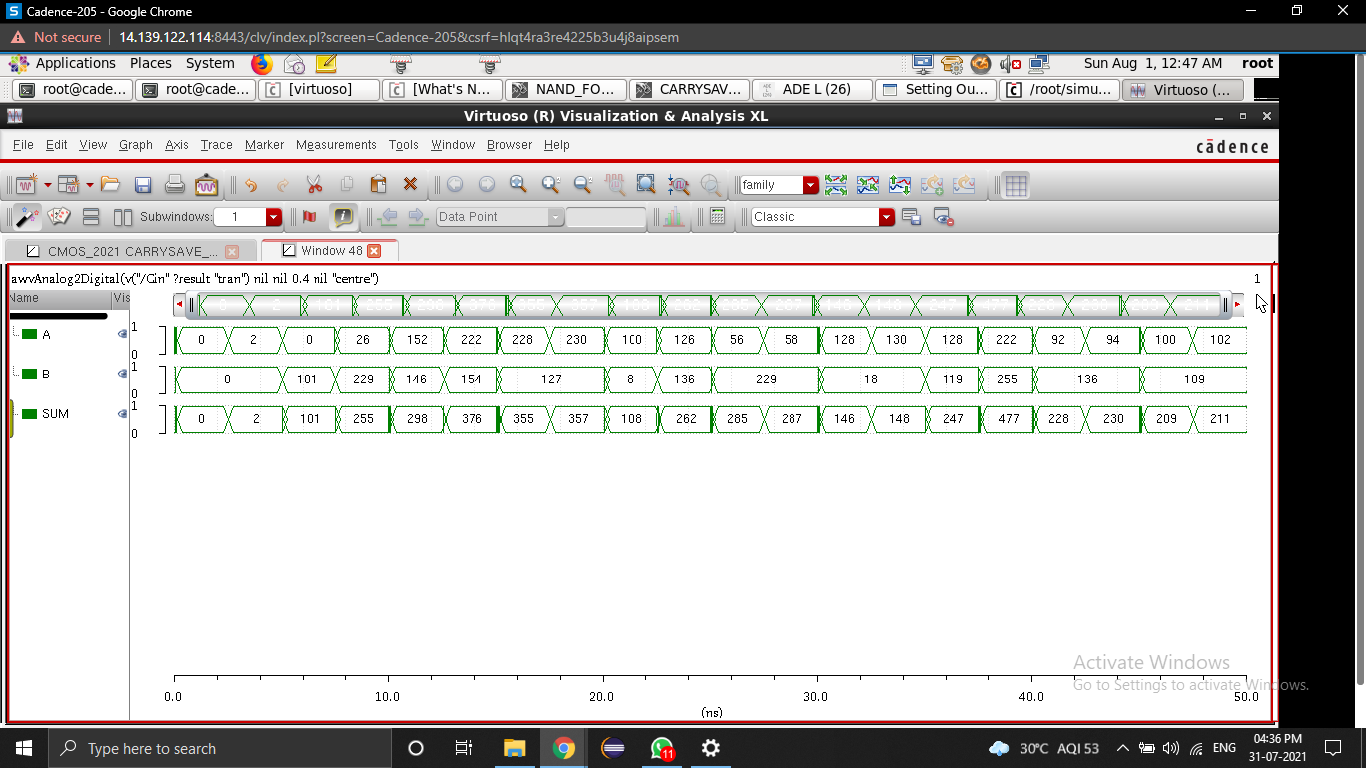
*Fig [17] Simulation results of CNTFET NAND Based Full Adder*



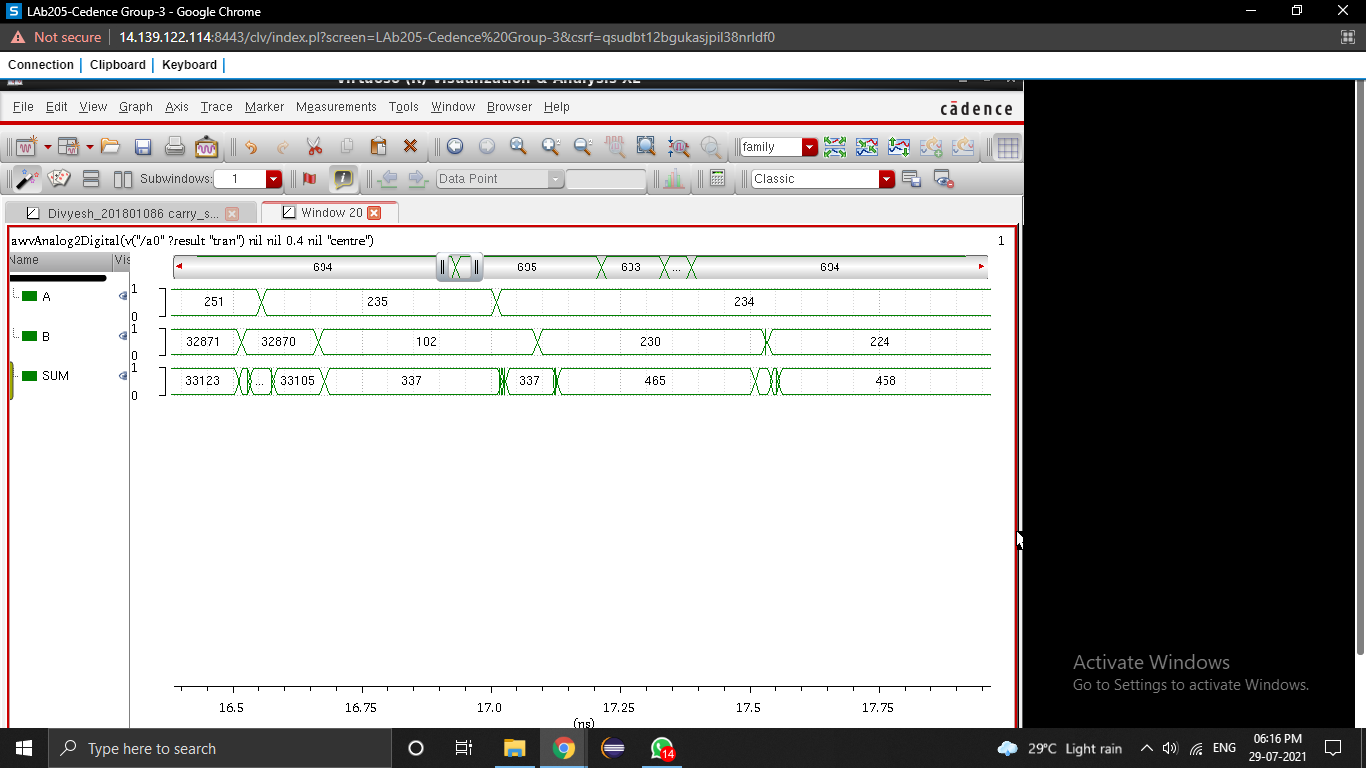
*Fig [18] Simulation results of CMOS based 4-bit Ripple Carry Adder*



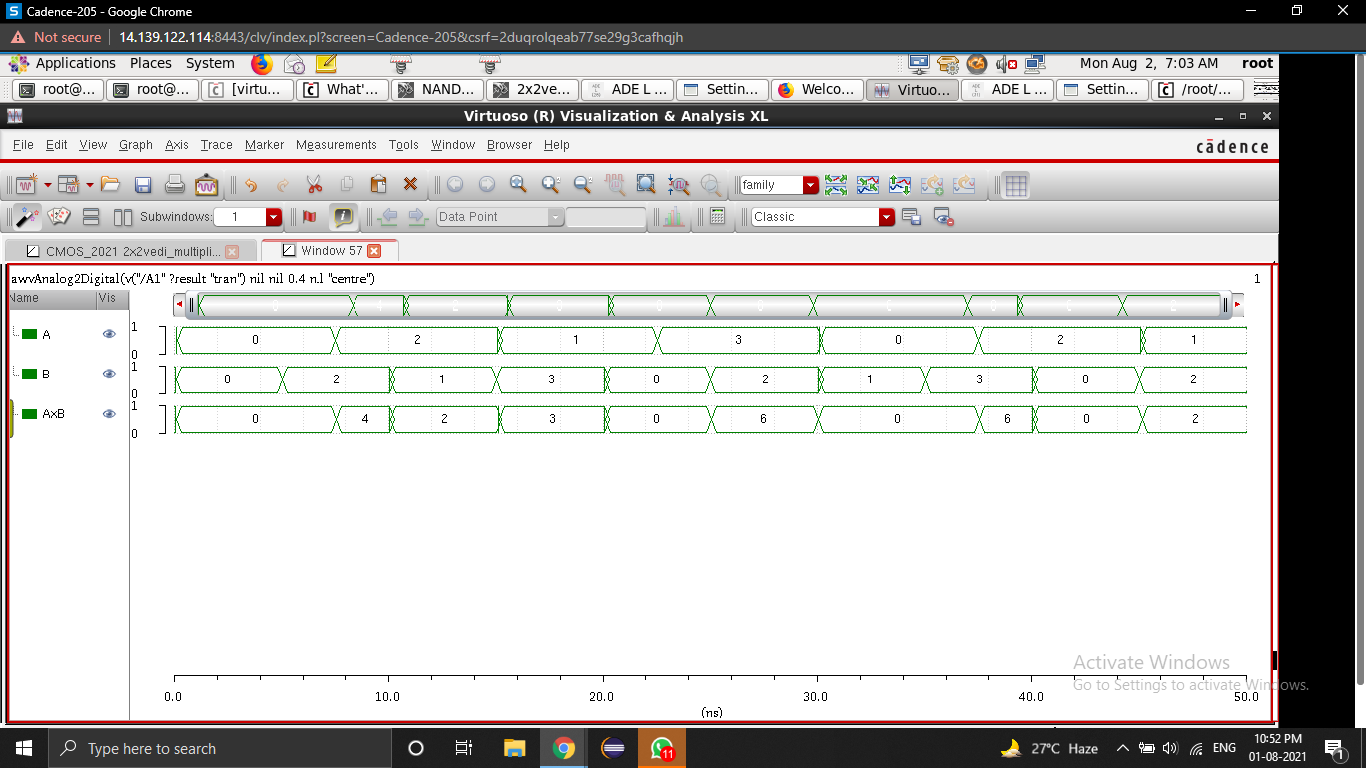
*Fig [19] Simulation results of CNTFET based 4-bit Ripple Carry Adder*



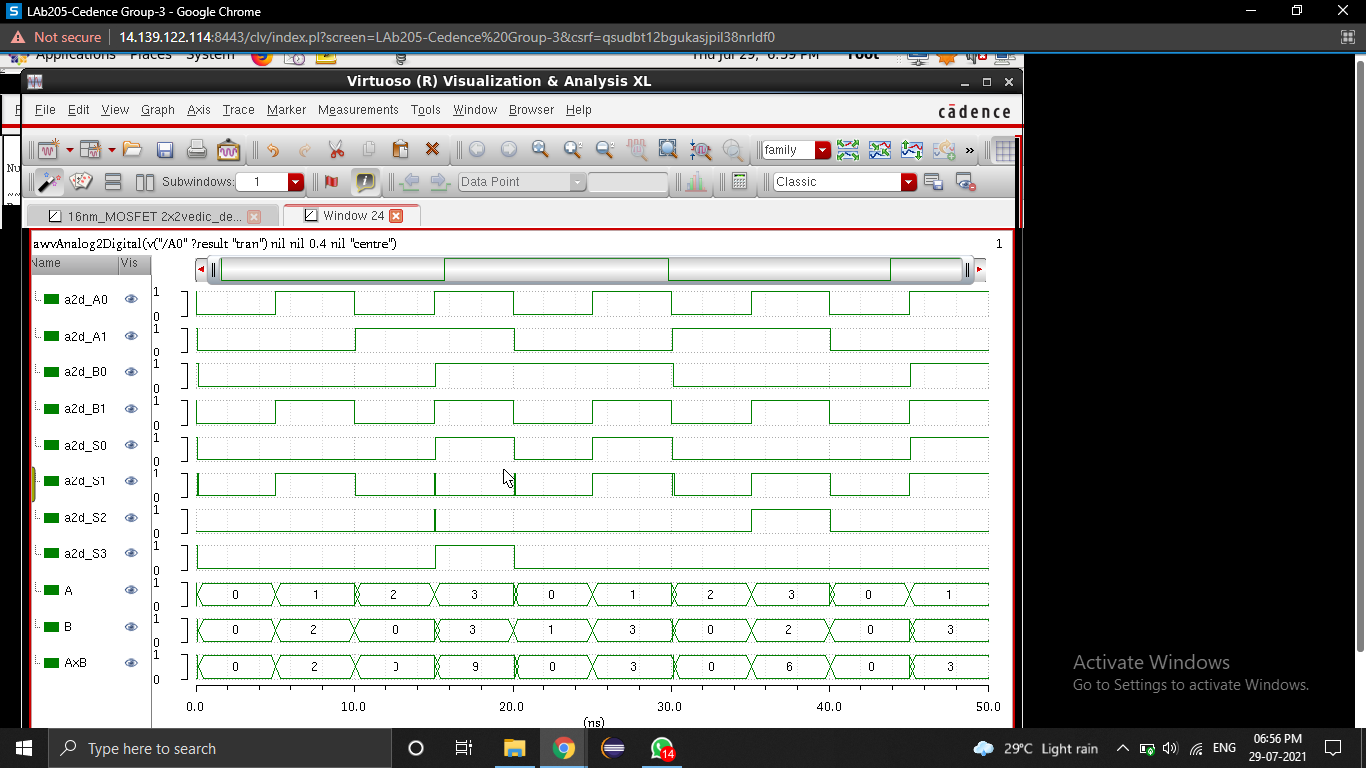
*Fig [20] Simulation results of CMOS based Carry Save Adder*



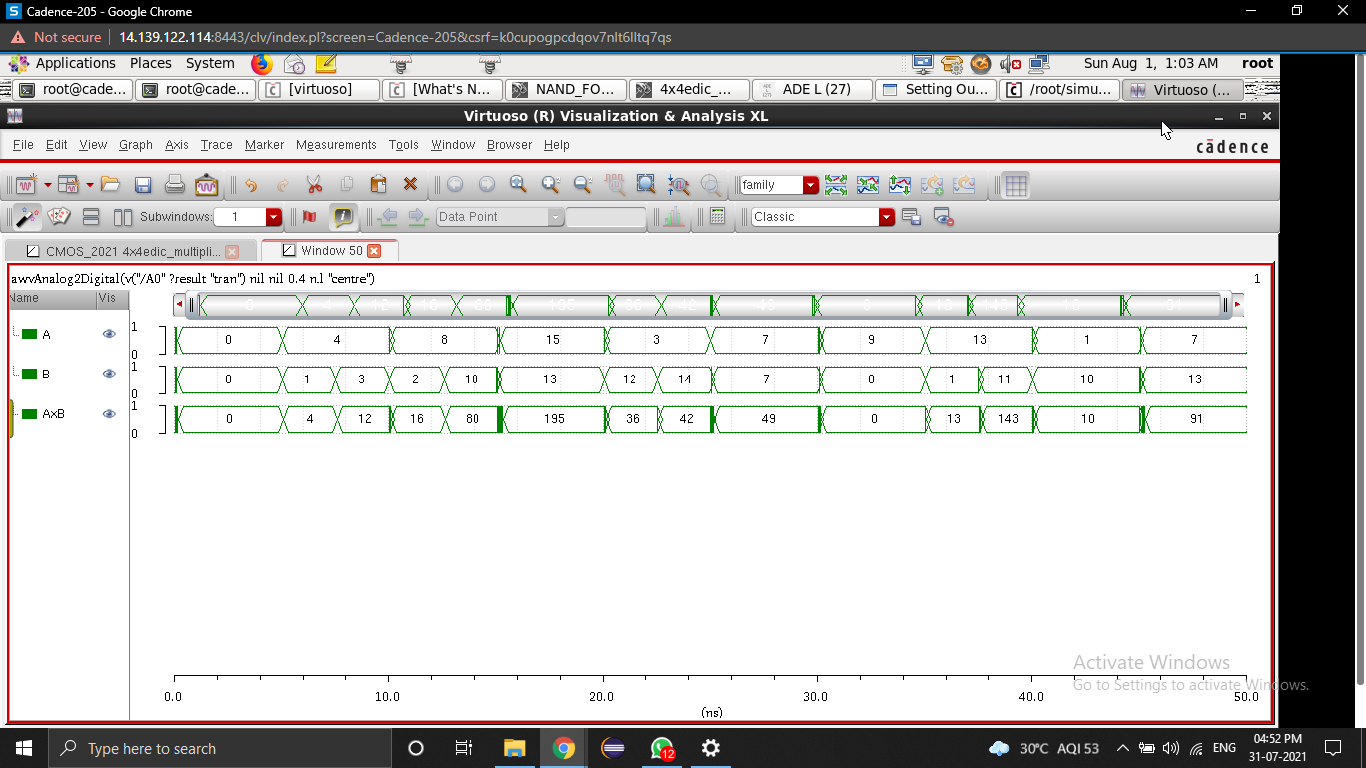
*Fig [21] Simulation results of CNTFET based Carry Save Adder*



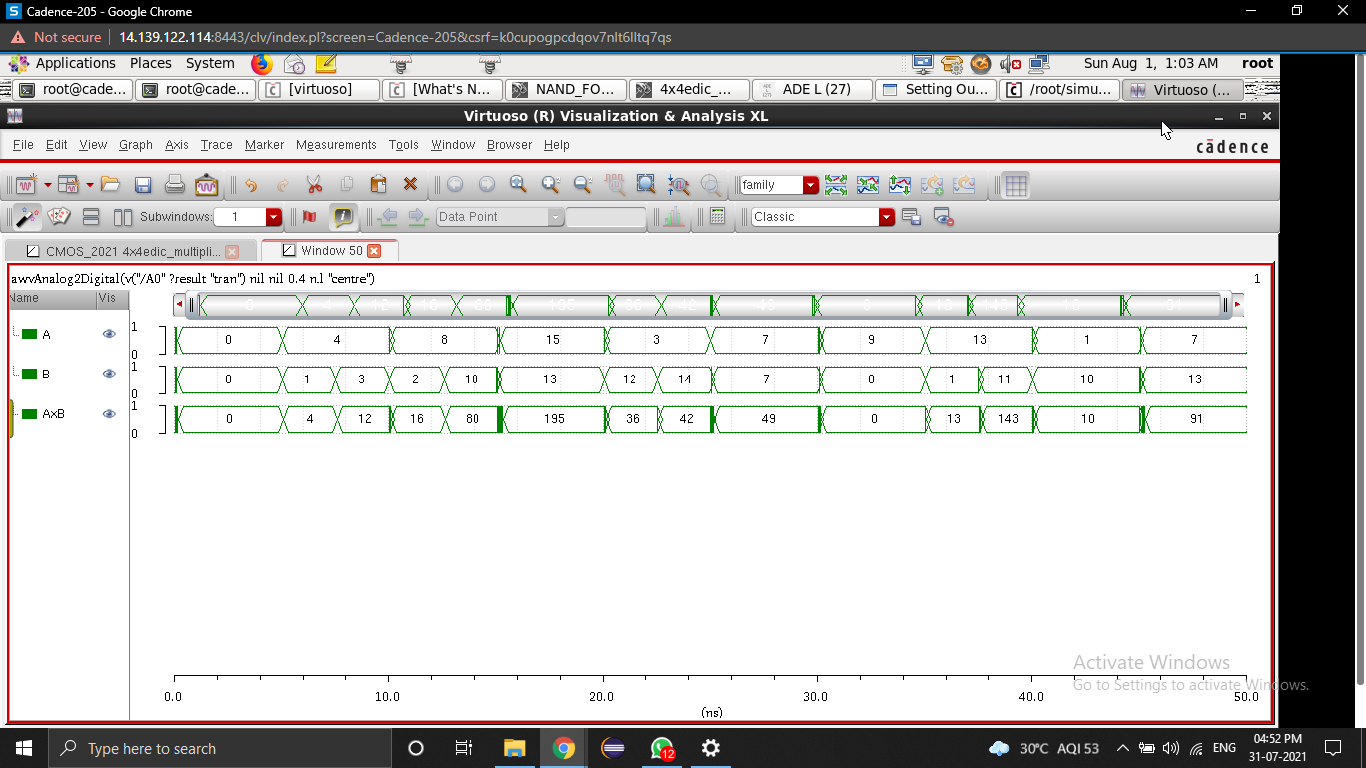
*Fig [22] Simulation results of CMOS based 2x2 bit Vedic Multiplier*



*Fig [23] Simulation results of CNTFET based 2x2 bit Vedic Multiplier*



*Fig [24] Simulation results of CMOS based 4x4 bit Vedic Multiplier*



*Fig [25] Simulation results of CNTFET based 4x4 bit Vedic Multiplier*

VI. DISCUSSIONS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CIRCUIT NAME | CMOS | | CNTFET | |
| **DELAY** | **POWER** | **DELAY** | **POWER** |
| NAND | 22.47 ps | 1.26 nW | 21.70 ps | 50.45 pW |
| NOR | 24.15 ps | 2.79 nW | 22.51 ps | 109.6 pW |
| FULL ADDER | 32.14 ps | 137.5 nW | 31.42 ps | 80.22 pW |
| RIPPLE CARRY ADDER | 43.40 ps | 389.8 nW | 25.55 ps | 23.38 pW |
| CARRY SAVE ADDER | 55.99 ps | 92.92 nW | 55.06 ps | 9.99 pW |
| 2x2 VEDIC MULTIPLIER | 70.2 ps | 373.4 nW | 59.2 ps | 165 pW |
| 4x4 VEDIC MULTIPLIER | 81.5 ps | 305.2 nW | 60.2 ps | 410.6pW |

**Table [2]** Comparative analysis between CMOS based and CNTFET based circuits on basis of power and delay

VII. ANALYSIS AND CONCLUSION

|  |  |  |
| --- | --- | --- |
| CIRCUIT NAME | Increased Efficiency in % | |
| **DELAY** | **POWER** |
| NAND | 3.43 | 95.96 |
| NOR | 6.79 | 96.07 |
| FULL ADDER | 2.24 | 99.94 |
| RIPPLE CARRY ADDER | 41.1 | 99.99 |
| CARRY SAVE ADDER | 1.66 | 99.98 |
| 2x2 VEDIC MULTIPLIER | 15.67 | 99.96 |
| 4x4 VEDIC MULTIPLIER | 26.13 | 99.87 |

**Table [3]** Percentage analysis of CMOS based and CNTFET based circuits on basis of power and delay

As we can see from above table, CNTFET based circuits are much better than CMOS based circuit on basis of area, delay and power. CNTFET based circuits has an optimisation of 13% on an average in delay and 98.83% on an average in power. CNTFET based NAND showed improvement in delay by 3.4% and improvement in power by 95.99%. CNTFET based NOR exhibited enhancement in delay by 6.7% and improvement in power by 96.07%. Considering the fact that important factors are highly optimised and these are universal gates, we can successfully state that when CNTFET based NAND and NOR are used in any circuit, the circuit will be have great reduced delay and power consumption and will perform more efficiently as compared to any CMOS based circuits.

In addition, CNTFET based Full Adder was further optimised by 2.2% of reduction in delay and 99.94% of reduction in power. CNTFET based Ripple Carry Adder was optimised to minimise 41.12% in delay and 99.99% in power. CNTFET based Carry Save Adder was also enhanced to decrease 1.66% in delay and 99.98% in power. The adders on an overall basis very optimised a lot. From the results we can also say that CNTFET based Ripple Carry Adder is the best adder among the three on the basis on delay and CNTFET based Carry Save Adder is the best adder on the basis of power.

In case of CNTFET based 2x2 Vedic Multiplier, there was reduction in delay by 15.66% and reduction in power by 99.95%. Also, CNTFET based 4x4 Vedic Multiplier, delay was improved by 26.13% and power was improved by 99.86%. In terms of power, CNTFET based 2x2 Vedic Multiplier is better than CNTFET based 4x4 Vedic Multiplier. In terms of delay, CNTFET based 4x4 Vedic Multiplier is better than CNTFET based 2x2 Vedic Multiplier.

Overall, CNTFET based circuits are better considering all important factors like power, delay and area as compared to CMOS based circuits. Till now, ternary half adder and full adder have been implemented. Based on latest founding, in a research paper, the author used CMOS based 4x4 Vedic Multiplier and delay calculated was 0.93 ns and the power calculated was 13.05 uW [21]. These values are higher compared to this paper. In another paper, the author used CNTFET based 4x4 Vedic Multiplier. The computed power was 317nW [22], which is more than the power calculated in this paper. Based on ongoing research and the results, we can thus say that CNTFET circuits are better than CMOS based circuits and thus have promising future.

REFERENCES

[1] Oklobdzija, V. “High-Speed VLSI Arithmetic Units: Adders and Multipliers.” (1999).

[2] K.V. Karthikeyan, R. Babu, N. Mathan, B. Karthick, “Performance analysis of an efficient MAC unit using CNTFET technology”

[3] H. M. Rakesh and G. S. Sunitha, "Design and Implementation of Novel 32-Bit MAC Unit for DSP Applications," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154177.

[4] Geunho Cho, Yong-Bin Kim, F. Lombardi and MinSu Choi, "Performance evaluation o CNFET-based logic gates," 2009 IEEE Instrumentation and Measurement Technology Conference, 2009, pp. 909-912, doi: 10.1109/IMTC.2009.5168580.

[5] K.V. Karthikeyan, R. Babu, N. Mathan, B. Karthick, “Performance analysis of an efficient MAC unit using CNTFET technology”

[6] R.A. Jaber, A.M. El-Hajj, A. Kassem, L.A. Nimri, A.M. Haidar, CNFET-

based designs of Ternary Half-Adder using a novel “decoder-less” ternary multiplexer based on unary operators, Microelectronics Journal (2020)”

[7] “Design of a high performance CNTFET-based full adder cell applicable in: Carry ripple, carry select and carry skip adders” by Hani Taheri Tari, Arezoo Dabaghi Zarandi, Mohammad Reza Reshadinezhad.

[8] “High performance, variation-tolerant CNFET ternary full adder a process, voltage, and temperature variation-resilient design” by Samane Firouzi, Sepehr Tabrizchi, Fazel Sharifi, Abdel-Hameed Badawyc.

[9] R. A. Jaber, A. Kassem, A. M. El-Hajj, L. A. El-Nimri and A. M. Haidar, "High-Performance and Energy-Efficient CNFET-Based Designs for Ternary Logic Circuits," in IEEE Access, vol. 7, pp. 93871-93886, 2019, doi: 10.1109/ACCESS.2019.2928251.

[10] S. K. Sinha and S. Chaudhury, "Advantage of CNTFET characteristics over MOSFET to reduce leakage power," 2014 2nd International Conference on Devices, Circuits and Systems (ICDCS), 2014, pp. 1-5, doi: 10.1109/ICDCSyst.2014.6926211.

[11]https://en.wikipedia.org/wiki/Carbon\_nanotubes\_in\_interconnects#:~:text=In%20nanotechnology%2C%20carbon%20nanotube%20interconnects,elements%20of%20an%20integrated%20circuit.&text=Metallic%20carbon%20nanotubes%20have%20been,and%20to%20replace%20copper%20interconnects.

[12] https://nano.stanford.edu/stanford

[13] Sugana, R. and Rathinasabapthy, V., 2018. A SURVEY ON DESIGN AND IMPLEMENTATIONS ON LOW POWER MULTIPLY ACCUMULATE UNIT (MAC) FOR FUTURE WIRELESS SENSOR NETWORK(WSN).

[14] Nagaraj, S., Reddy, D. S., & Mastani, D. A. (2020). ANALYSIS OF MAC UNIT FOR CONVENTIONAL CMOS AND LOW POWER CPL & DPL LOGICS. JOURNAL OF CRITICAL REVIEWS.

[15] Kumar, A.P. & Aditya, B.L.V.S.S. & Sony, G. & Prasanna, Ch & Satish, A. (2019). Estimation of power and delay in CMOS circuits using LCT. Indonesian Journal of Electrical Engineering and Computer Science. 14. 990-998. 10.11591/ijeecs.v14.i2.pp990-998.

[16] https://www.gatevidyalay.com/ripple-carry-adder/

[17] Preet, Raminder et al. “Performance Analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry-Look- Ahead Adder.” (2009).

[18] R. A. Javali, R. J. Nayak, A. M. Mhetar and M. C. Lakkannavar, "Design of high speed carry save adder using carry lookahead adder," International Conference on Circuits, Communication, Control and Computing, 2014, pp. 33-36, doi: 10.1109/CIMCA.2014.7057751.

[19] Sinha, Professor G. (2014). Implementation of High Speed Vedic Multiplier.

[20] Verma, Pushpalata. (2012). Design of 4x4 bit Vedic Multiplier using EDA Tool. International Journal of Computer Applications. 48. 32-35. 10.5120/7466-0564.

[21] Yadav, Rishu & Kumar, Manish. (2020). Implementation of 4×4 Fast Vedic Multiplier using GDI Method. 527-529. 10.1109/ICE348803.2020.9122890.

[22] Senthilkumar, V. M., Ravindrakumar, S., Nithya, D., & Kousik, N. V. (2019). A vedic mathematics based processor core for discrete wavelet transform using FinFET and CNTFET technology for biomedical signal processing. *Microprocessors and Microsystems*, *71*, 102875.