ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2386-7876
Vol.01, Issue.03
March-2026
Pages: 417-425

SQUARING CIRCUIT USING 14NM FINFET TECHNOLOGY WITH VEDIC MATHEMATICS APPROACH

[1] **Pechetti Girish**, [2] **Dikkupati Sushma Roselin**, [3] **Gudala Ratnaraju**, [4] **Kolla Sriram Kiran**, [5] **Gubbala Aseesu**

[1]Assistant Professor, Dept of E.C.E, BVCITS, Batlapalem, Amalapuram, AP
[2,3,4,5]B. Tech, Dept of E.C.E, BVCITS, Batlapalem, Amalapuram, AP

ABSTRACT: A novel method for squaring binary numbers using Vedic mathematics is proposed in this paper. The implementation of the binary squaring circuit uses the improved Vedic multiplier architecture. The circuit is designed in DSCH using the 14nm technology. Dwandwayoga has a duplex property that is used to find a square of an N-bit number. Also, Yavadunam is a special case for finding the square of a number that is closer to the base. This work targets a comparative analysis of these two square circuits having a Vedic approach with a conventional array multiplier. TThe circuit synthesis is performed using the Micro-wind, DSCH. The schematic based simulation is presented for 4-bit in DSCH. The design can easily be expanded for large bit sized inputs. The device utilization and delay comparison are presented using existing methods. The binary squaring circuit presented in this paper show better parametrised performance than the previously reported squaring circuits.

Keywords: FinFET; Square circuit; Vedic mathematics; Vedic multiplier; Dwandwayoga, Yavadunam, urdhva tiryagbhyam

INTRODUCTION: Among the forceful investigation in the field of low power, high speed digital applications due to the growing demand of systems like phones, laptop, palmtop computers, cellular phones, wireless modems and portable multimedia applications etc has directed the VLSI technology to scale down to nano-regimes, allowing additional functionality to be incorporated on a single chip. The designer‟s novel purpose in the field of multifaceted digital circuit design is minimization of power consumption. These investigations are responsible for special design techniques for digital circuits distant from conventional CMOS design style. A large body of investigate has been performed to expand and advance conventional Complementary Metal Oxide Semiconductor (CMOS) techniques for the fabrication of ULTRA low power integrated circuits (ICs). The purpose of this study is to expand a faster, lower power, and reduced area substitute to standard CMOS logic circuits. Mod-VEDIC technique is one such new technique for minimization ofpower consumption in the digital circuit design field[1,2,3]. The increasing market of portable, battery-powered electronic systems preferred microelectronic circuits design with ultra low power dissipation. Since the size, complexity and integration of the chips keep on to increase, thecomplexity in providing enough cooling might either include important cost or limit the operability of the computing systems which formulate utilization of those integrated circuits. At the same time as the technology node scales down to 65nm regime, not to a great extent increase in dynamic power dissipation is noticed, though the static or leakage power is similar as or exceeds the dynamic power beyond 65nm technology node. Therefore the methods of power dissipation reduction are not limited to dynamic power. In this paper we discuss circuit and logic level design approaches to power minimization such that dynamic, leakage and short circuit power dissipation. Power optimization for low power applications can be achieved at different levels such that System level, Algorithm level , Architecture level, Circuit/Logic level and technology level. Logic level power optimization method have been considered here which have a huge potential for power saving. As a result optimization at circuit / logic level is moreover very imperative for miniaturization of IC[4]s. For the period of the last two decades investigation has resulted in the progress of numerous logic design techniques. An alternative approach to CMOS logic design is pass-transistor logic (PTL) design having advantages over standard CMOS design, these are: high speed which results from the small node capacitances; low power dissipation which results from the reduced number of transistors used in fabrication; and lower interconnection effects which results from small area. But there were also some basic problems in PTL realizations; i) the threshold drop across the single channel pass transistors which results in reduced current drive and therefore slower function at reduced supply voltages [1]. This threshold voltage drop is significant for low power design because it is advantageous to function at the lowest possible voltage level. ii) In view of the fact that the input voltage for a high logic level at the regenerative inverters is not VDD, the PMOS device in the inverter is not fully turned off, and for this reason direct-path static power dissipation can be important. After that many investigations have performed to resolve these problems which results in techniques like: Transmission Gate CMOS (TG), Complementary Pass-transistor Logic (CPL), and Double Pass-transistor Logic

(DPL). The multiplier finds its usage in practically all kinds of processing systems ranging from application specific processors dealing with an infinitely large bit width or a small scale general processor dealing with a comparably smaller subset of data. It also happens to be one of the most time-consuming digital processes and offers a good scope for improvement in terms of area, delay as well as power efficiency.

LITERATURE SURVEY: There are different types and designs of full adder which is discussed in various papers at state of the art level and process and circuit level. Twelve state of the art full adder cells are: conventional CMOS, CPL, TFA, TG CMOS,C2MOS, Hybrid, Bridge, FA24T, N-Cell, DPL and Mod2f. R. Shalem, E. John, and L.K. John, proposed a conventional CMOS full adder consisting of 28 transistors [1]. Later, the number of transistor count is reduced to have less area and power consumption. A. Sharma, R Singh and R. Mehra, Member, IEEE, have improved performance with Transmission Gate Full adder using CMOS nano technology where 24 transistors are used [2]. The Complementary Pass transistor Logic (CPL) full Adder contains the 18 transistors. The power consumption of this structure is 2.5μw [3]. A Transmission Function Full Adder (TFA) based on the transmission function theory has 16 transistors. The power consumption of this structure is 12μw. N-CELL contains the 14 transistors and utilizes the low power XOR/XNOR circuit. The power consumption of this structure is 1.62μw. Mod2f Full Adder contains the 14 transistors, generates full swing XOR and XNOR signals by utilizing a pass transistor based DCVS circuit. The power consumption of this structure is 2.23μw [3]. Saradindu Panda, N. Mohan Kumar, C.K. Sarkar, optimized the full adder circuit to 18 Transistor using Dual Threshold Node Design with Submicron Channel Length [4]. T. Vigneswaran, B. Mukundhan, and P.Subbarami Reddy, designed 14 transistor high speed CMOS full adder and significantly improved threshold problem to50% [5]. Gate Diffusion Input Technique is a new method of reducing power dissipation, propagation delay with less area. T. Esther Rani, M. Asha Rani, Dr.RameshwarRao, designed an area optimized low power arithmetic and logic unit in which Arithmetic Logic Unit is implemented using logic gates, pass transistor logic, as well as VEDIC technique [6].Manish Kumar, Md. Anwar Hussain, and L.L.K. Singh explained a Low Power High Speed ALU in 45nm UsingVEDIC Technique and Its Performance Comparison [7].

EXISTING MTHOD:

SQUARE CIRCUIT USING URDHVA TIRYAGBHYAM and GDI:

URDHVA TIRYAGBHYAM :

Urdhva – Triyakbhyam is the general formula applicable to all cases of multiplication and also in the division of a large number by another large number. It means vertically and crosswise. We discuss multiplication of two, 4 digit numbers with this method [8-9].

VEDIC MULTIPLIER ARCHITECTURE : The hardware architecture of 2X2, 4x4 and 8x8 bit Vedic multiplier module are displayed in the below sections. Here, "Urdhva-Tiryagbhyam" (Vertically and Crosswise) sutra is used to propose such architecture for the multiplication of two binary numbers. The beauty of Vedic multiplier is that here partial product generation and additions are done concurrently. Hence, it is well adapted to parallel processing. The feature makes it more attractive for binary multiplications. This in turn reduces delay, which is the primary motivation behind this work

VEDIC MULTIPLIER FOR 2X2 BIT MODULE: The method is explained below for two, 2 bit numbers A and B where A = a1a0 and B = b1b0 as shown in Fig. 2. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit Of the finel product. The 2X2 Vedic multiplier module is implemented using four input AND gates & two half-adders which is displayed in its block diagram in Fig. 3. It is found that the hardware architecture of 2x2 bit Vedic multiplier is same as the hardware architecture of 2x2 bit conventional Array Multiplier [2]. Hence it is concluded that multiplication of 2 bit binary numbers by Vedic method does not made significant effect in improvement of the multiplier‟s efficiency. Very precisely we can state that the total delay is only 2-half adder delays, after final bit products are generated, which is very similar to Array multiplier. So we switch over to the implementation of 4x4 bit Vedic multiplier which uses the 2x2 bit multiplier as a basic building block. The same method can be extended for input bits 4 & 8. But for higher no. of bits in input, little modification is required.

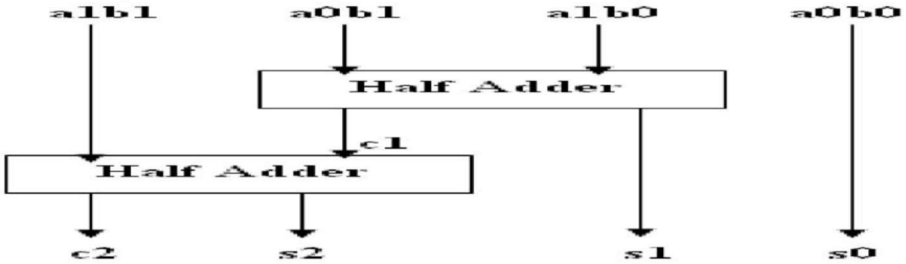

Fig.1 Block Diagram of 2x2 bit Vedic Multiplier

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules as discussed in Fig. 3. Let"s analyze 4x4 multiplications, say A= A3 A2 A1 A0 and B= B3 B2 B1 B0. The output line for the multiplication result is – S7S6S5S4 S3 S2 S1 S0 .Let"s divide A and B into two parts, say A3A2 & A1 A0 for A and B3 B2 & B1B0 for B. Using the fundamental of Vedic multiplication, taking two bit at a timeand using 2 bit multiplier block, we can have the following structure for multiplication as shown in Fig. 4.

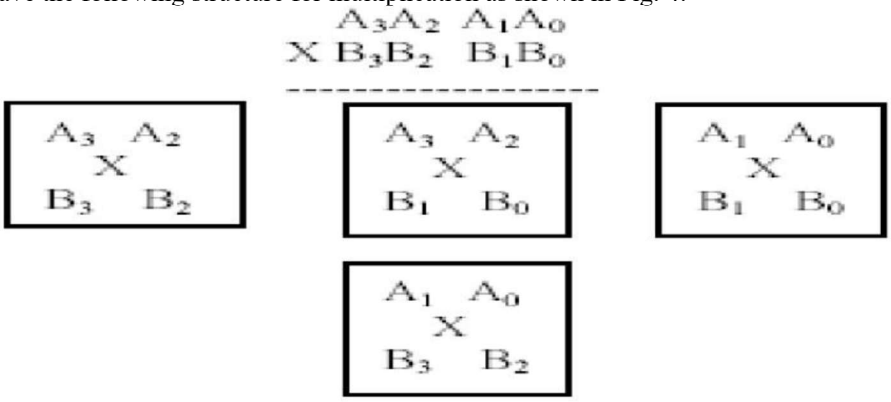

Fig. 2 Sample Presentation for 4x4 bit Vedic Multiplication

Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A1A0 and B1B0. The last block is 2x2 bit multiplier with inputs A3 A2 and B3 B2. The middle one shows two 2x2 bit multiplier with inputs A3 A2 & B1B0 and A1A0 & B3 B2. So the final result of multiplication, which is of 8 bit, S7 S6S5S4 S3 S2 S1 S0. To understand the concept, the Block diagram of 4x4 bit Vedic multiplier is shown in Fig. 5. To get final product (S7 S6 S5 S4 S3 S2 S1 S0), four 2x2 bit Vedic multiplier (Fig. 3) and three 4-bit Ripple-Carry (RC) Adders are required. The proposed Vedic multiplier can be used to reduce delay. Early literature speaks about Vedic multipliers based on array multiplier structures. On the other hand, we proposed a new architecture, which is efficient in terms of speed. The arrangements of RC Adders shown in Fig. 5, helps us to reduce delay. Interestingly, 8x8 Vedic multiplier modules are implemented easily by using four 4x4 multiplier modules

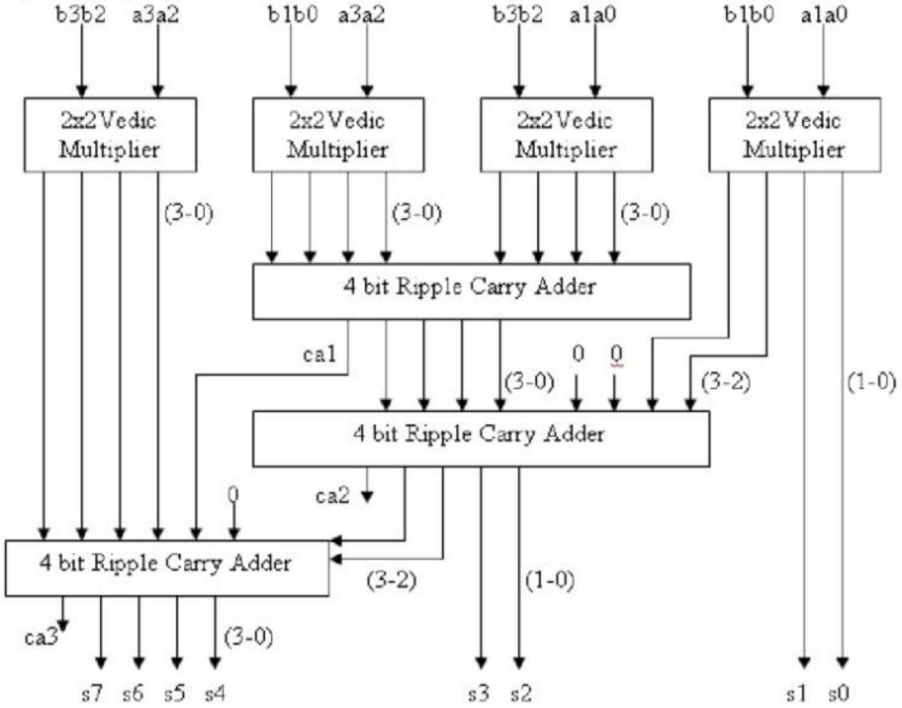

Fig. 3 Block Diagram of 4x4 bit Vedic Multiplier

GATE DIFFUSION TECHNIQUE:

The GDI method is based on the use of a simple cell as shown in Fig. 1. At a first glance the basic cell resembles the standard CMOS inverter, but there are some important differences: GDI cell contains three inputs – G (the common gate input of the nMOS and pMOS transistors), P (input to the outer diffusion node of the pMOS transistor) and N

(input to the outer diffusion node of the nMOS transistor). The Out node (the common diffusion of both transistors) may be used as input or output port, depending on the circuit structure.

The GDI cell is similar to a CMOS inverter structure. In a CMOS inverter the source of the PMOS is connected to VDD and the source of NMOS is grounded. But in a GDI cell this might not necessarily occur. There are some important differences between the two. The three inputs in GDI are namely-

1) G- common inputs to the gate of NMOS and PMOS
2) N- input to the source/drain of NMOS
3) P- input to the source/drain of PMOS

Bulks of both NMOS and PMOS are connected to N or P (respectively), that is it can be arbitrarily biased unlike in CMOS inverter. Moreover, the most important difference between CMOS and GDI is that in GDI N, P and G terminals could be given a supply 'VDD' or can be grounded or can be supplied with input signal depending upon the circuit to be designed and hence effectively minimizing the number of transistors used in case of most logic circuits (eg. AND, OR, XOR, MUX, etc). As the allotment of supply and ground to PMOS and NMOS is not fixed in case of GDI, therefore, problem of low voltage swing arises in case of GDI which is a drawback and hence finds difficulty in case of implementation of analog circuits.

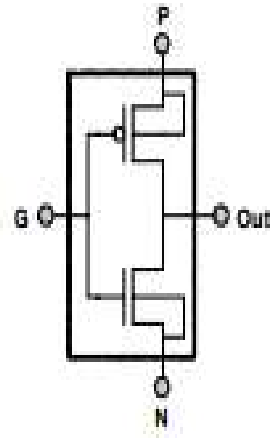

Fig 4: Basic GDI gate

Multiple-input gates can be implemented by combining several GDI cells. The buffering constrains, due to possible VT drop are described in detail in [8], as well as the technological compatibility with CMOS (and with SOI). Morgenshtein has proposed basic GDI cell shown in Fig.1 [8]. This is a new approach for designing low powerdigital combinational circuit.GDI technique is basically two transistor implementation of complex logic functions whichprovides in-cell swing restoration under certain operating condition.

PROPOSEED METHOD:

SQUARE CIRCUIT USING Dwandwayoga SUTRA WITH FIN-FET

The Duplex Combination or 'Dwandwa Yoga' is used for squaring the numbers. It is based on the Duplex property. According to this property, to find the square for the even numbers, the result is taken as twice the product of the outermost pair and then the product of the next outermost pair and so on, until no pairs are left. To find the square for the odd numbers, the same procedure is followed except that the one bit left in the middle and this enters as its square along with the product elements.

$$D(a) = a^2$$

$$D(ab) = 2ab$$

$$D(abc)=2ac+b^2$$

$$D(abcd) = 2ad+2bc$$

$$D(abcde) = 2ae+2bd+c^2$$

$$D(abcdef) = 2af+2be+2cd$$

As we seen above, D of any number is the sum of the square of the middle number and twice the product of the other pairs. When compared to array squaring algorithm, our squaring algorithm gives low delay. The performance of proposed squaring algorithm is increased compared to the conventional multiplier. Less number of LUT's is used in the proposed algorithm when compared to the conventional multiplier. The size of the hardware circuit can be

decreased as a result of decrease in the number of LUT's due to the increase in number of bits. In the same way the delay of the squaring algorithm is decreased and so it can be used in many applications where high speed is required. Algorithm for the 4×4 bit Square using Dwandwa Yoga.

1. D (A0) = A0*A0 =A.
2. D(A1A0) = 2*A1*A0 = B.
3. D(A2A1A0) = 2*A2*A0 +A1*A1 = C.
4. D(A3A2A1A0) = 2*A3A*A0+2*A2*A1 = D.
5. D(A3A2A1) =2*A3*A1 + A2*A2 =E.
6. D(A3A2) =2*A3*A2 = F.
7. D(A3)=A3*A3=G.

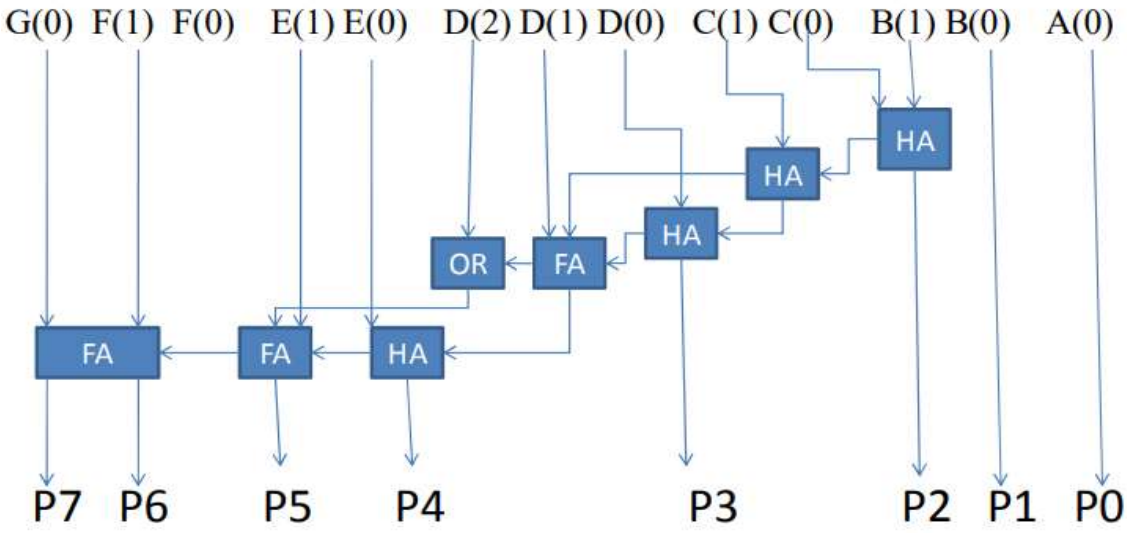

Figure . Blocks In Square Architecture

SQUARE CIRCUIT USING YAVADUNAM SUTRA WITH FIN-FET

YAVADUNAM SUTRA: Yavadunam sutra is used to find the square of a number that is closer to the base. Yavadunam Tavadunikartya Varganacha Yojayet is up sutra which means, "Find the deficiency that is added or subtracted from the number and later square the deficiency". This sutra works efficiently for the numbers that are closer to the base of power of 10. Hence it can be considered a special case to calculate the square of a number.

Proof: Let A be base and B be deficiency and (a + b) is the number

$(a+b)^2 = a^2 + 2ab + b^2$

$= a(a + 2b) + b^2$

$= a(a+b+b) + b^2$

= Base(number + deficiency) + (deficit) ^{2}

Let's see how this Yavadunam sutra works stepwise. (1) Calculate the deficiency close to the base. Here, Deficiency is the difference between number and base. (2) Find the square of deficiency and keep it to the right side of the final answer (3) Add deficiency to number (if deficiency (def) is negative then subtract) (4)

Result = (Num + def + Carryforward) and (def)2 Let's see an example to calculate a square of 102 which is close to base 100.

So deficiency = 102-100 = 02.

Deficiency square = (02) 2 = 04

Result = 100(102 + 02) + 04 = 10404 We will see another example of negative deficiency. Ex.(995) 2here deficiency is 995 − 1000 = − 5
Deficiency square = (−5) 2 = 025
Subtract deficiency from number = 995 − 5 = 990
Result = 990025 For Binary numbers, we can calculate the square of binary numbers using the same methodology. Binary numbers have base 2. The algorithm is used to find the square of the 4-bit binary number as follows. [1]
Let Input = A3 A2 A1 A0

[2] Let Output = $B_7\ B_6\ B_5\ B_4\ B_3\ B_2\ B_1\ B_0$
[3] Deficiency = $A_2\ A_1\ A_0$
[4] $(\text{Deficiency})^2 = (A_2\ A_1\ A_0)^2 = (P_5\ P_4\ P_3\ P_2\ P_1\ P_0)$

where $B_2\ B_1\ B_0 = P_2\ P_1\ P_0$

[5] Carry = P5 P4 P3
[6] $A_3\ A_2\ A_1\ A_0 + A_2\ A_1\ A_0 = X_5\ X_4\ X_3\ X_2\ X_1\ X_0$
[7] $B_7\ B_6\ B_5\ B_4\ B_3\ B_2\ B_1\ B_0 = X_5\ X_4\ X_3\ X_2\ X_1\ X_0 + P_5\ P_4\ P_3$

$$\begin{aligned}
\text{Ex. } A_3A_2A_1A_0 &= 1110 \\
\text{Deficiency} &= A_2A_1A_0 = 110 \\
(\text{Deficiency})^2 &= (A_2A_1A_0)^2 \\
&= (P_5P_4P_3P_2P_1P_0) \\
&= 100100 \\
B_2B_1B_0 &= P_2P_1P_0 \\
&= 100\ \text{Carry} = 100 \\
A_3A_2A_1A_0 + A_2A_1A_0 &= X_5X_4X_3X_2X_1X_0 = 10100 \\
B_7B_6B_5B_4B_3B_2B_1B_0 &= X_5X_4X_3X_2X_1X_0 \\
&\quad + P_5P_4P_3 \\
&= 11000100
\end{aligned}$$

FinFET

Basic Structure of FinFET

The basic structure of FinFET consist of 3D shaped Fin , gate , drain and source and substrate

- **Fin :** Name of FinFET is derived from this 3 D vertical shaped fin like structure that act as channel in FinFET . Fin is made of semiconductor material or silicon.
- **Gate :** Role of gate in FinFET is similar to common MOSFET.Fin of the FinFET surrounds the gate of FinFET and gate form 3 D structure , their can be more than one gate in FinFET . Gate is made up of metal . Gate is used to control the flow of electric current in channel .
- **Drain and Source :** Drain and source of FinFET plays similar role as in MOSFET . Current enter from source and drain and gate is used to control the current . Carriers in channel enter through source and exist from drain . As in MOSFET , In FinFET also drain is in high potential and source is in low potential.
- **Substrate :** Substrate of FinFET act a base for whole structure and it helps to isolate device in chips.

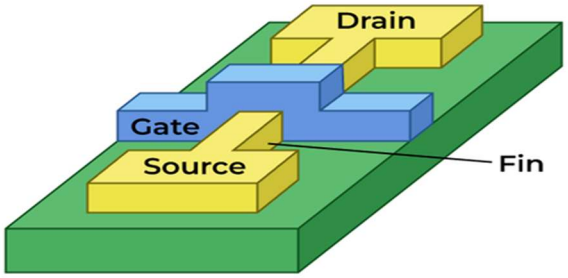

Fig: Basic Structure of FinFET

Computing FinFET Transistor Width

Width of transistor is denoted by w , w plays a crucial role in determining the current flowing through transistor , short - channel effect . width is also used to determine aspect ratio(w/l) which decide the design rule of transistor . For a double gate finFET w is equals to twice of FinFET effective height.

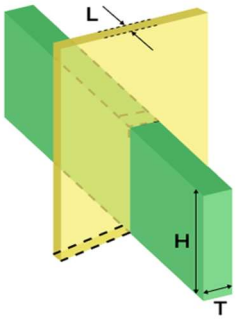

Fig: FinFET Dimensions

Here,
L= length of gate
T= thickness of Fin
H=height of fin
w= 2(H) for digate FinFET
w=(2(H)+T) for trigate FinFET

RESULT:

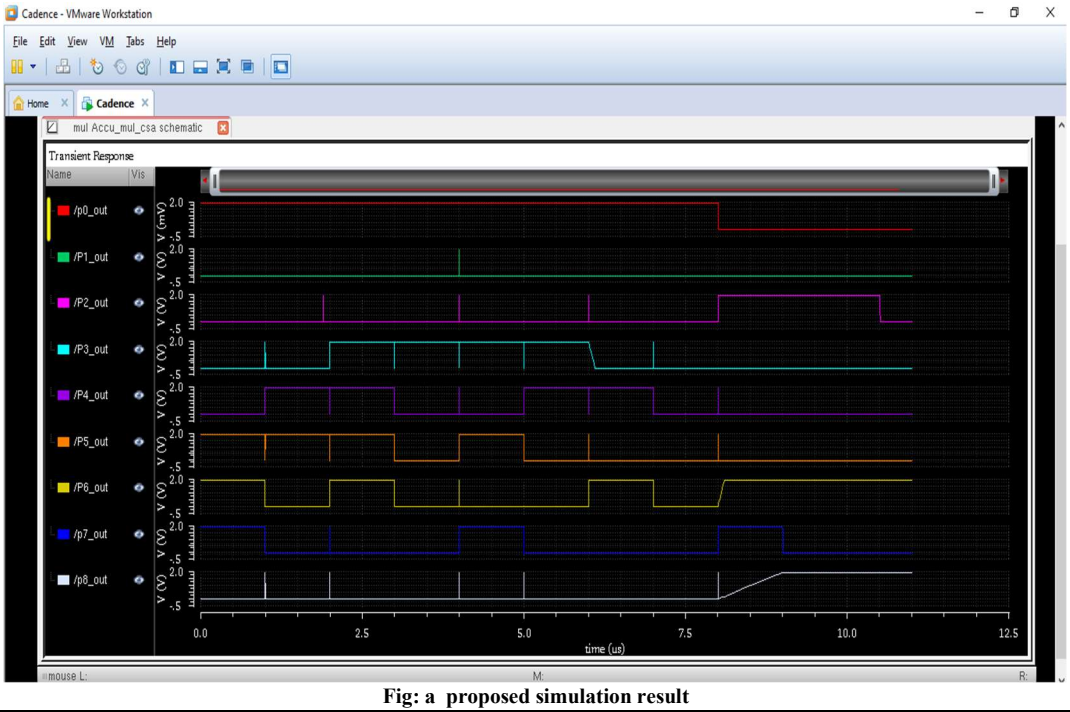

Fig: a proposed simulation result

CONCLUSION: All Simulations are performed through microwind based on CMOS technology, and results show power characteristics of VEDIC technique of low power digital circuit design. VEDIC approach allows realization of a broad variety of multifaceted logic functions by means of less transistors.

Innovative square units are proposed in this work which is based on Vedic mathematics using 14 nm FinFET technology. The proposed design gives primitive results. Proposed designs also excel in previous conventional designs in terms of area and speed. Also, design complexity is reduced. Hence, the proposed square architectures are improved designs and can be used in digital arithmetic applications efficiently. **FUTURE SCOPE:** Overall, the future scope of a squarer unit will be applied to dual channel multiplier project is likely to be shaped by the continued growth and development of various industries that rely on signal processing, control systems, and other applications that require the manipulation of analog signals.

REFERENCES :

[1] Jagadguru Swami, Sri Bharati Krisna, Tirthaji Maharaja, "Vedic Mathematics or Sixteen Simple Mathematical Formulae From the Veda, Delhi (1965)", Motilal Banarsidas, Varanasi, India,

[2] M. Morris Mano, "Computer System Architecture", 3rd edition, Prientice-Hall, New Jersey, USA, 1993, pp. 346-348.

[3] H. Thapliyal and H.R Arbania. "A Time-Area-Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", Proceedings of the 2004 International Conference on VLSI (VLSI‟04), Las Vegas, Nevada, June 2004, pp. 434-439.

[4] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, Vol.8, No.2, 2004, UICEE Published in Australia.

[5] Thapliyal H. and Srinivas M.B, "High Speed Efficient NxN Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, 2004, Vol.2.

[6] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced– Bit Multipliction Algorithm for Digital Arithmetics" International Journal of Computational and Mathematical Sciences

[7] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim and Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematician", International SoC Design Conference, pp. 65- 68, 2008.

[8] Parth Mehta and Dhanashri Gawali, "Conventional versus Vedic mathematics method for Hardware implementation of a multiplier", International conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 640-642, 2009.

[9] Ramalatha, M.Dayalan, K D Dharani, P Priya, and S Deoborah, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", International Conference on Advances In Computationa Tools for Engineering Applications (ACTEA) IEEE, pp. 600-603, July15-17, 2009.

[10] Sumita Vaidya and Deepak Dandekar, "Delay-Power Performance comparison of Multipliers in VLSI Circuit Design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, pp 47-56, July 2010.

[11] S.S.Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A "Implementation of Vedic Multiplier For Digital Signal " International conference on VLSI communication & instrumentation

[12] Pushpalata Verma, K. K. Mehta" Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-5, June 2012

[13] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power," IEEE Trans. VLSI Syst., vol. 2, pp. 426–435, Dec. 1994.

[14] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," IEEE J. Solid-State Circuits, vol. 32, pp. 1079–1090, June 1997.

[15] J. P. Uyemura, Circuit Design for CMOS VLSI. Norwell, MA: Kluwer Academic, 1992, pp. 88–129.

[16] R. J. Baker, H. W. Li, and D. E. Boyce, "CMOS circuit design, layout, and simulation," IEEE Press Series on Microelectronic Systems, pp. 205–242.

[17] SIA Roadmap [Online]. Available: http://www.sematech. org/public/roadmap/doc

[18] E. Shannon and W. Weaver, The Mathematical Theory of Information. Urbana-Champaign: University of Illinois Press, 1969.

[19] A. Morgenshtein, A. Fish, and I. A. Wagner, Gate-diffusion input (VEDIC)—A technique for low power design of digital circuits: Analysis and characterization, submitted for publication.

[20] , "Gate-diffusion input (VEDIC)—A novel power efficient method for digital circuits: A design methodology," presented at the 14th Int. ASIC/SOC Conf., Washington, DC, Sept. 2001.

[21] A. Chandrakasan, W. J. Bowhill, and F. Fox, Design of High Performance Microprocessor Circuits, 2000, ch. 5, pp. 80–97.

[22] I. Sutherland, B. Sproull, and D. Harris, Logical Effort: Designing Fast CMOS Circuits. San Mateo, CA: Morgan Kaufmann, p. 7.

[23] V. Stojanovic and V. G. Oklobdzija, "Comparative analysis of masterslave latches and flip-flops for high-performance and low-power systems," IEEE J. Solid-State Circuits, vol. 34, pp. 536–547, Apr. 1999.