International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Efficient High Speed Computing Low Power Multiplier Architecture using Vedic Mathematics For Digital Signal Processing Applications

Dr. S. Sridhar
Electronics and Communication Engineering
Lendi Institute of Engineering and Technology
Vizianagaram, India

Sai Viswanath Singamsetty
Electronics and Communication Engineering
Lendi Institute of Engineering and Technology
Vizianagaram, India

Monisha Tanniru
Electronics and Communication Engineering
Lendi Institute of Engineering and Technology
Vizianagaram, India

Rasagna Sidhantam
Electronics and Communication Engineering
Lendi Institute of Engineering and Technology
Vizianagaram, India

***Abstract*— DSP(Digital signal processing) applications typically invole complicated mathematical operations to be performed repetitively on data samples with less delay and power consumption. Multiplication based operations like multiply and accumulate(MAC) and inner products are few computationally Aggressive Arithmetic Functions(CAAF) frequently contained in today's VLSI systems and DSP applications like FFT, Finite Impulse Response filters etc. Design of a high speed 4/8 bit Vedic multiplier based on the techniques of ancient Vedic Mathematics is proposed in this manuscript. Vedic method of multiplication strikes a difference in the conventional method of multiplication. Verilog HDL is employed for designing the structural modules, while Xilinx ISE tool is used for validating the designs.**

Keywords*— *Vedicmultiplier, fast binary multiplication, HDL simulation, synthesis.

I. INTRODUCTION

Multipliers are the primary arithmetic operational units that determine the execution time in many digital signal/image processing, networking and communication based applications. For the past many decades multiplier designs are revolving around parameters of consideration like high speed, low power consumption, regularity of layout and hence less area or even combination of them (with trade-offs among them) thus paving to implement compact digital systems. Owing to these parameters, technological advancements are urging researchers to innovate novel multiplication algorithms based on various serial, parallel and hybrid multiplication techniques. Although there exist several types of multiplication algorithms, we have chosen to implement a digital multiplier suitable for digital signal and image processing applications using Vedic mathematics based on the ensured guarantee of faster and accurate calculations by Vedic mathematics.

In twentieth century Vedic mathematics name is discovered which is given to the ancient Indian system of mathematics. The system of Vedic Mathematics was entirely based upon sixteen Sutras or word-formulae and thirteen sub-sutras [8] [9]. Vedic mathematics reduces the complex calculations that exist in conventional mathematics. Out of all the sixteen sutras available in Vedic mathematics, only two of them are made use of in multiplication operations. They are Urdhva Triyakbhyam sutra and Nikhilam Sutra. Urdhva Triyakbhyam sutra deals with vertical and cross wise where as Nikhilam deals with all from 9 and last from 10.Techniques of Vedic mathematics for digital multiplication are quite different from conventional multiplication techniques. S Akhter et.al [1] implemented fast N×N multiplier in Vedic mathematics using VHDL, as it gives effective utilization of structural method of modelling [1]. B. D. Kumar et.al [2] highlighted the efficiency of Vedic multiplication where the design complexity is drastically reduced modularity greatly increased for inputs having large no of bits. H. Goyal et.al [3] implemented fast multiplier based on Vedic mathematics using modified square root carry select adder, this method reduces the carry propagation delay. Since adders are main components used in multipliers, using fast adders definitely enhances the overall performance of Vedic multipliers. Various Adder topologies like Ripple Carry Adder (RCA), Carry Select Adder (CSA), Square Root Carry Select Adder (SQRT-CSA) etc., are used to compare area, delay and power [4]. Since squaring operation plays a vital role in high speed applications (where the speed is a crucial performance characteristic) like animation, Digital signal processing, and image processing, etc. Yavadunam sutra and bit reduction techniques are employed for squaring binary numbers in High speed VLSI architecture [5].

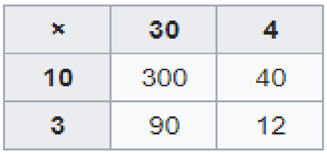
Multipliers employed in binary cubing architectures are designed using Anurupyena sutra of Vedic mathematics. Combinational path delay is reduced and it achieves better area in terms of slices. Comparison is made between conventional and Vedic methods [6]. These are faster than conventional square and cube because of its regular and and parallel structure of Vedic square and cube. Because of reduced hardware complexity and delay the proposed Vedic multipliers square and cube can be implemented in Arithmetic and Logic Unit designs replacing the traditional square and cube circuits [7]. Most of the computing time is occupied by squaring operation and it becomes to increase the speed which are to be squared.

Remaining sections of the manuscript are organised as follows. Section II deals with Different multiplication algorithms and section III deals with the proposed Vedic multiplication architecture. Experimental evaluations and analysis are depicted in section IV and section V concludes the manuscript.

II. DIFFERENT MULTIPLICATION ALGORITHMS

Multipliers play a prominent role in various digital signal/image processing, wireless communications, Networking, Embedded Systems and various applicable areas. The basic arithmetic operations - addition and multiplication are fundamentally present in majority of the electronic circuits. Statistics clarify the facts that more than 70% instructions in microprocessors (and/or microcontrollers) and DSP algorithms include multipliers. Usage of multiplier circuits always compel designers to trade for added complexity and increased silicon area requirements with enhanced speed and power consumption etc. Of course there are numerous number of multipliers having unique pros and cons for themselves which include the Grid Method, Binary Multiplier, Shift and Add multiplier, Dadda Multiplier, Array multiplier and Sequential multiplier etc. Few of the multiplication algorithms are discussed in this section.

A. Grid Multiplication Algorithm Grid multiplication algorithm represents an introductory method of multiple-digit multiplication; it remained a standard part of mathematics curriculum in England since late 1990s. Process includes partitioning the both factors to be multiplied into their hundreds, tens and units parts. Further partial products are calculated explicitly in a simple multiplication-only stage and these contributions are totaled to generate the final product in separate addition stage. For example, calculation of 34 × 13 is computed using the grid as shown in fig 1.



×	30	4
10	300	40
3	90	12

Fig 1: Example for Grid Multiplication

Product is obtained through forming row-by-row totals (300 + 40) + (90 + 12) = 340 + 102 = 442.

B. Binary Multiplication Algorithm Binary Multiplication algorithm generates product of two binary numbers. Process includes calculating the partial products, shifting and adding them together. Long multiplicand is multiplied by 0 or 1 which is much easier than decimal multiplication as product by 0 or 1 is 0 or same number respectively Architecture for Binary Multiplier as shown in the fig 2 .

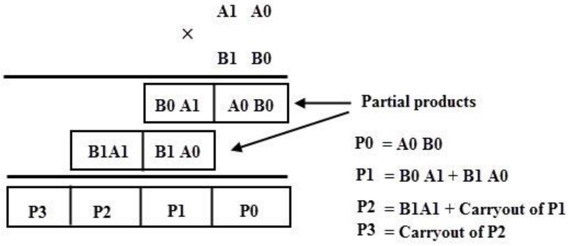


Fig 2: Example for Binary Multiplication

C. Shift and Add Multiplication Algorithm Depending on the LSB bit value of multiplier, the multiplicand value is added and accumulated. For every clock cycle the multiplier bits are shifted one bit to the right and its value is tested. If the multiplier bit is ‘0’, then only a shift operation is performed. Similarly if the multiplier bit is ‘1’, then the multiplicand bits are added to the accumulator and resultant partial product is shifted to the right by one bit. After testing all multiplier bits, the product is retained in the accumulator. The accumulator is 2N (M+N) in size. This circuit carry several advantages in asynchronous circuits. Architecture for shift and add multiplication is shown in the figure.3 below.

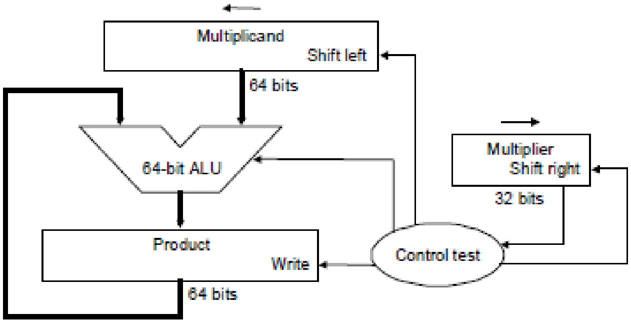


Fig 3: Example for Shift and Add Multiplication

D. DADDA Multiplication Algorithm Dadda multiplication almost resembles the Wallace multiplication algorithm. Unlike the Wallace multipliers that perform many reductions on each layer, Dadda multipliers do very few (minimum) reductions. Hence Dadda multipliers represent a less expensive reduction phase, while the numbers are a few bit longer requiring slightly larger adder circuits. This implies that only fewer columns are compressed in the initial stages of the column compression tree, and more number of columns are compressed in later levels of the multiplication algorithm. Dot-diagram representation of 8x8 Dadda multiplier is shown in Fig 4.Below.

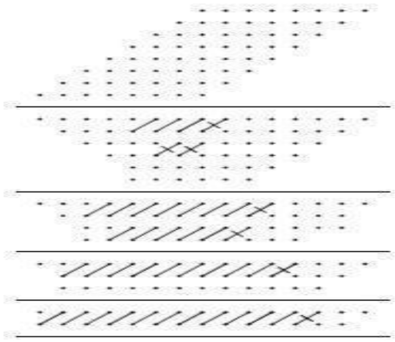


Fig 4: Example for Dadda Multiplication

A plain diagonal line connecting two dots in figure above represent the outputs of the full adder while, the outputs of half adder are represented by crossed diagonal lines connecting two dots.

E. Array Multiplication Algorithm Array multiplication algorithm represents an efficient layout of combinational multiplier design. Here the multiplication process is purely based on add and shift algorithm, each of the partial products are generated through ascertaining product of multiplicand with one multiplier bit every time. Likewise N-1 (N denotes number of multiplier bits) number of adders are required to complete the process. Hence array multiplier is considered to consume more number of logic gates and hence occupy more area. Therefore Array multiplier is economically less feasible though it is faster in operation [10] Architecture for Array multiplication as shown in fig 5.

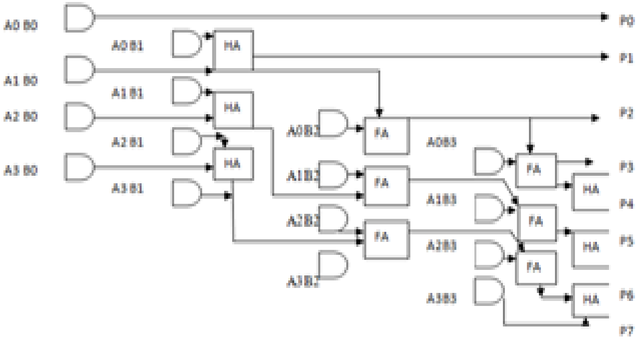


Fig 5: Example for Array Multiplication

F. Sequential Multiplication Algorithm Sequential multipliers are mostly preferred because of their low area requirements. A sequential multiplier multiplies two binary numbers (multiplicand X - n bits and multiplier Y - m bits) using single n bit adder, firstly the circuit processes a single partial product at once and it repeats the it to m times. Figure 6 below shows the partial product generation and addition in a sequential multiplier.

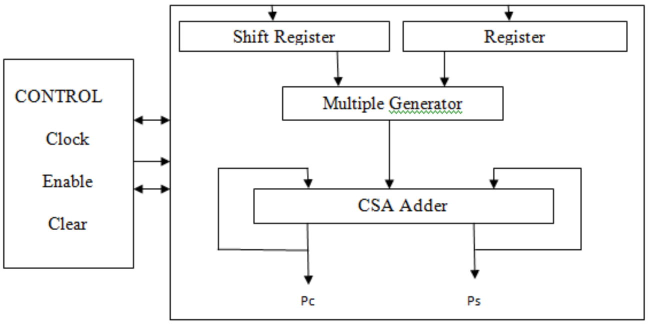


Fig 6: Example for Sequential Multiplication

Here the multiplication process is divided into some sequential steps such that in each step, the generated partial products are added to an accumulated partial sum. Further the partial sum is shifted to align the accumulated sum with partial product in next steps. In every step sequential multiplier includes three operations to generate partial products, adding partial products to the together partial sum later shifting the partial sum.Therefore, every step in sequential multiplier includes three different operations of generating partial products, adding partial products to the accumulated partial sum and shifting the partial sum.

III. PROPOSED VEDIC MULTIPLICATION ALGORITHM

The word “Vedic” is procured from the word “Veda”. The word Veda represents the store-house of all knowledge. Vedic mathematics is a part of four Vedas (books of wisdom), it is a part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. Vedic mathematics can give a proper knowledge of several concepts in mathematics like arithmetic, geometry, trigonometry, factorization, calculus, quadratic equations. Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884- 1960) makes all part of the work together and gave a proper mathematical explaination which is useful for various applications. Swamiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda[7] [8]. These 16 Strutas are dealing with various groups in mathematics like geometry calculus etc. These Sutras are enlisted alphabetically (below) along with their brief meanings [9].

- Chalana-Kalanabyham – Differences and Similarities.
- Ekadhikina Purvena – Increment one more than the previous one.
- Ekanyunena Purvena – Decrement one less than the previous one.
- Gunakasamuchyah – Factors of sum is equal to the sum of factors.
- Gunitasamuchyah – Product of sum is equal to the sum of product.
- Nikhilam Navatashcaramam Dashatah– All from 9 and last from 10.
- Paraavartya Yojayet – Transpose and adjust.

- Puranapuranabyham – By the completion or non-completion.
- Sankalana- vyavakalanabhyam – By addition and by subtraction.
- Shesanyankena Charamena – Remainders by the last digit.
- Shunyam Saamyasamuccaye – When the sum is the same that sum is zero.
- Sopaantyadvayamantyam – Ultimate and twice the penultimate.
- Urdhva-tiryagbhyam – Vertically and crosswise.
- Vyashtisamanstih – Part and Whole.
- Yaavadunam – Whatever the extent of its Deficiency

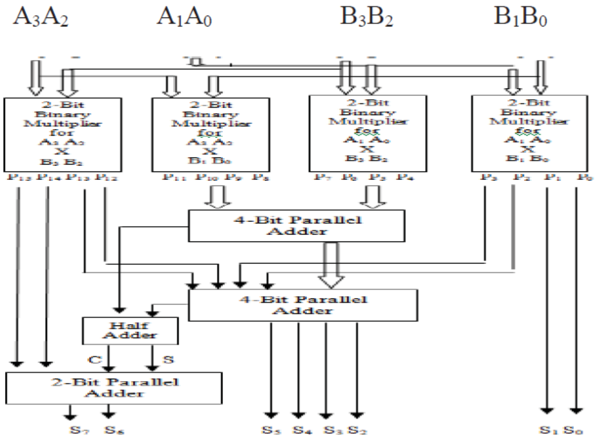

A3A2
A1A0
B3B2
B1B0
2-Bit Binary Multiplier for A3 A2 X B3 B2
2-Bit Binary Multiplier for A3 A2 X B1 B0
2-Bit Binary Multiplier for A1 A0 X B3 B2
2-Bit Binary Multiplier for A1 A0 X B1 B0
P15 P14 P13 P12
P11 P10 P9 P8
P7 P6 P5 P4
P3 P2 P1 P0
4-Bit Parallel Adder
4-Bit Parallel Adder
Half Adder
C
S
2-Bit Parallel Adder
S7 S6
S5 S4 S3 S2
S1 S0

Fig 7: Architecture of Vedic Multiplication

Architectural design of proposed Vedic multiplier is shown in Figure 7. above. For 4-bit multiplication Input bits are divided in two equal parts, cross and vertical product computations are performed. Input data is $A=A_3A_2A_1A_0$ and $B=B_3B_2B_1B_0$, while the output bits represented by $S=S_1S_2S_3S_4S_5S_6S_7S_8$. Proposed architecture is comprised of designing different structural modules like:

A) Four 2-bit binary multipliers
B) Two 4-bit parallel adders
C) One 2-bit parallel adder.
D) One Half Adder

Detailed operations relevant to structural details of above modules are elaborated below in a sequential manner.

A) TWO BIT BINARY MULTIPLIER:

Design of 2-bit binary multiplier is meant for multiplying the two 2-bit binary numbers and provides the result. Methodology of multiplication is based on calculating the partial products, shifting them and adding them together. The logic diagram of 2-bit binary multiplier (as depicted below in Figure 8) contains two **Half Adders** and **AND Gate** units.

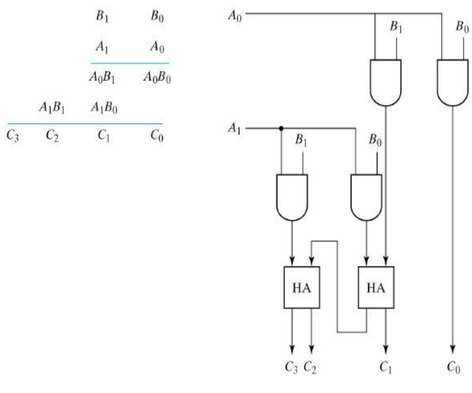


Fig 8: Logic Diagram of 2 –Bit Binary Multiplier

Four such 2-bit binary multiplier units are employed for the overall top level design unit.

B) 4-BIT PARALLEL ADDER:

A full adder can take three inputs and performs addition operation with an input carry. But a Parallel Adder is a digital circuit, which is a cascade of several full adders. Such a n-bit adder formed by cascading n full adders (FA_1 to FA_n) is used to add two n-bit binary numbers. A ripple-carry adder is a simple form of parallel adder, where the carry-out of each full adder is connected to the carry-in of the next full adder. Hence the total delay time of the adder is the time it would take for a carry to ripple through all bit-pair full adders. Block diagram of a 4-bit parallel adder is shown in Figure 9 below.

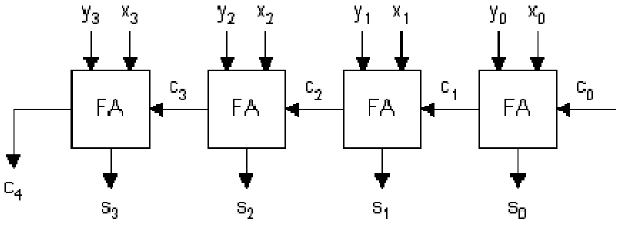


Fig 9: Block Diagram of 4-Bit Parallel Adder

C) 2-BIT PARALLEL ADDER:

A 2-Bit parallel adder is designed using an **EX-OR (Exclusive OR) gate** and AND gate. Here two full adders are connected in a parallel manner. The Block Diagram of 2-bit parallel adder is shown in fig 10.

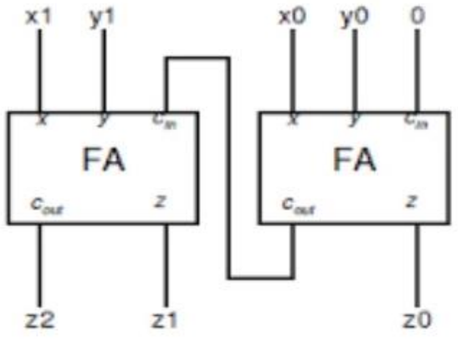


Fig 10: Block Diagram of 2-Bit Parallel Adder

The logic diagram of 2-Bit Parallel Adder is shown in Figure 11 below.

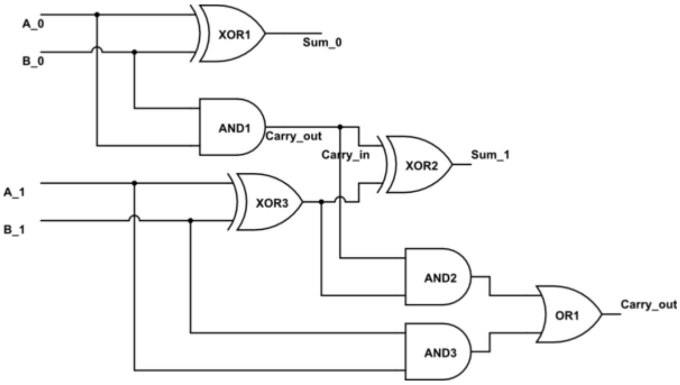


Fig 11: 2-Bit Parallel Adder Logic Diagram

D) HALF ADDER:

Half adder is constructed using an EX-OR gate and an AND gate, it is meant for adding 2 bits and generate a sum bit (S) and carry bit (C) as the outputs. Though half adder is a simple circuit it has a major disadvantage of adding only two input bits (A and B) and has nothing to do with the carry (if there is any in the input). That means the binary addition process is not complete and hence it is called a half adder. Logic diagram of an Half Adder circuit is shown in Figure 12. Below.

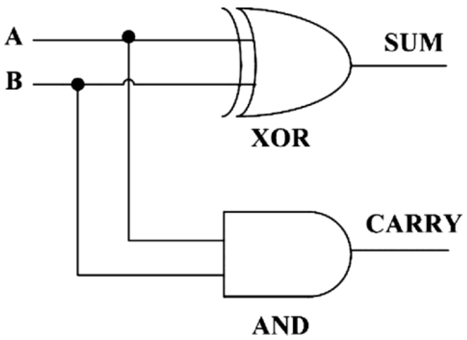
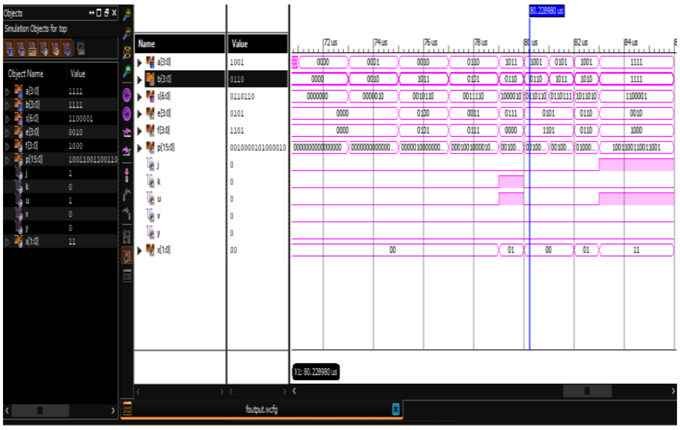


Fig 12: Half Adder Circuit

IV. EXPERIMENTAL ANALYSIS

Structural modules of proposed multiplier are designed using Verilog (HDL) programming language. The structural design modules are validated using Xilinx ISE tool. Experimental evaluations pertaining to the simulated waveforms and results obtained for various other blocks are critically evaluated below.



Object Name	Value
a[3:0]	1111
b[3:0]	1111
s[6:0]	1100001
e[3:0]	0010
f[3:0]	1000
p[15:0]	10011001100110
j	1
k	0
u	1
v	0
y	0
x[1:0]	11

Name	Value
a[3:0]	1001
b[3:0]	0110
s[6:0]	0110110
e[3:0]	0101
f[3:0]	1101
p[15:0]	0010000101000010
j	0
k	0
u	0
v	0
y	0
x[1:0]	00

Fig 13: Simulation Waveforms of 4-Bit Multiplier

Experimental analysis performed for 2-bit, 4-bit, 8-bit and 16-bit multiplications. Simulations relevant for 4-bit Vedic multiplication operations corresponding to different 4-bit input values (A and B) are displayed in Figure 13.Similarly waveforms relevant to 2-bit operations are shown in Figure14. below.

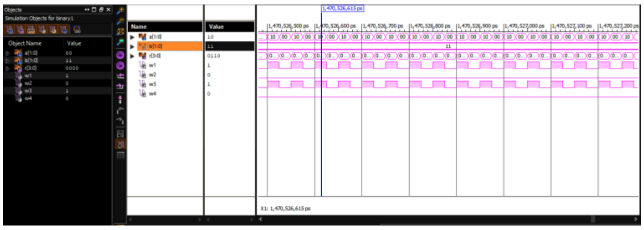


Fig 14: Simulation Waveforms of 2-Bit Multiplier

The waveforms relevant to 4-bit parallel adder with inputs A[3:0] and B[3:0] generating 8-bit sum output and single bit carry output is displayed in Figure15.below.

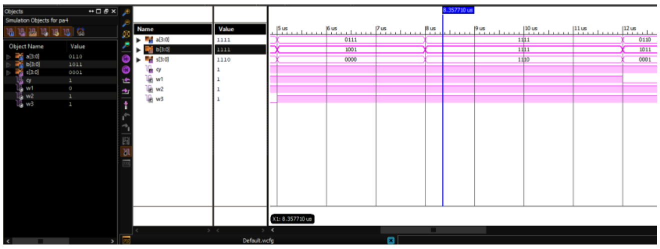
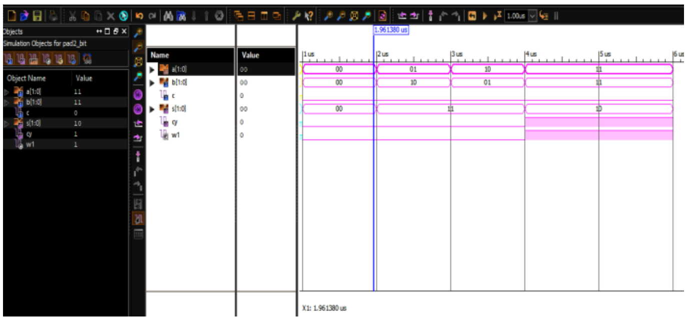


Fig 15: Simulation Waveforms of 4-bit Parallel Adder

Waveforms relevant to 2-bit parallel adder with inputs A[1:0] and B[1:0] generating 4-bit sum output and single bit carry output is displayed in Figure16.below.



Objects
Simulation Objects for pad2_bit

Object Name	Value
a[1:0]	11
b[1:0]	11
c	0
s[1:0]	10
cy	1
w1	1

Name	Value
a[1:0]	00
b[1:0]	00
c	0
s[1:0]	00
cy	0
w1	0

Fig 16: Simulation Waveforms of 2-bit Parallel Adder

V. CONCLUSION

With regards to operation of various multiplier circuits parameters like operating speed, power consumption and silicon area under usage are of primary concern, that are in turn dependent on the typical adder circuits deployed. Proposed Vedic mathematics binary multiplier circuits are explored with binary parallel adder circuit components. Here it is observed that parallel adders are more efficient in terms boperation. In contrast Carry Increment Adder circuits are meant for future deployment as they require only little `additional logic and reduced delay at the cost of circuit area.

REFERENCES

[1] S. Akhter, "VHDL implementation of fast N×N multiplier basedon Vedic mathematics," in *Proc. 18th European Conference on Circuit Theory and Design*, 2007, pp. 472-475.

[2] B. D. Kumar and M. Bharathi, "A high speed and efficient design for binary number squaring using Dwandwa yoga," *International Journal of Advanced Research in Computer Engineering &Technology*, vol. 1, no. 4, pp. 476-479, Jun. 2012.

[3] H. Goyal and S. Akhter, "VHDL implementation of fastmultiplier based on Vedic mathematic using modified square rootcarry select adder," *International Journal of Computer Applications*, vol. 127, no. 2, pp. 24-27, Oct. 2015.

[4] S. Akhter, V. Saini, and J. Saini, "Analysis of Vedic multiplierusing various adder topologies," in *Proc. IEEE 2017 InternationalConference on Signal Processing and Integrated Networks*, 2017,pp. 173-176.

[5] A. Deepa and C. N. Marimuthu, "High speed VLSI architecturefor squaring binary numbers using Yavadunam sutra and bitreduction technique," *International Journal of Applied EngineeringResearch*, vol. 13, no. 6, pp. 4471-4474, 2018.

[6] P. Vidyashankari and B. Lokesha, "Design and implementation of square and cube architectures using Vedic sutras on FPGA," *International Journal of Emerging Technology in Computer Science& Electronics*, vol. 14, no. 2, pp. 377-381, Apr. 2015.

[7] K. Pichhode, M. D. Patil, D. Shah, and B. Chaurasiya Rohit,"FPGA implementation of efficient Vedic multiplier," in *Proc. IEEE2015 International Conference on Information Processing*, 2015,pp. 565-570.

[8] G.Ganesh Kumar, V.Charishma, "Design of High Speed Vedic Multiplier paper. using Vedic Mathematics Techniques", International Journal of Scientific and Research Publications, Volume 2, Issue 3, March 2012.

[9] Krishnaveni, D.Umarani "VLSI IMPLEMENTATION OF VEDIC MULTIP-LIER WITH REDUCED DELAY", International Journal of Advanced Technology & Engineering Research (IJATER) National Conference on Emerging Trends in Technology (NCET-Tech).

[10] Sumit R. Vaidya, D. R. Dandekar, "Performance Comparison of Multipliers for Power-Speed Trade-off in VLSI Design," RECENT ADVANCES in NETWORKING, VLSI and SIGNAL PROCESSING, ISSN: 1790-5117, ISBN: 978-960-474-162-5.