International Journal of Electrical, Electronics and Computer Engineering **6**(1): 62-70(2017)

ISSN No. (Online): 2277-2626

An Approach Towards the High Efficient and low Propagation Delay in Digital Processor

Ashish Tiwari* and Sonu Lal**

**Research Scholar, Department of Electronics & Communication Engineering,*
IES College of Technology, Bhopal, (Madhya Pradesh), INDIA
***Associate Professor, Department of Electronics & Communication Engineering,*
IES College of Technology, Bhopal, (Madhya Pradesh), INDIA

(Corresponding author: Ashish Tiwari, jackofrk@gmail.com)

(Received 27 November, 2016 Accepted 08 January, 2017)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Presently due to the low speed of adders as they involved partially in the multiplication operation the speed of multiplier is limited. In reference to the digital processor, the speed mainly depends on the multipliers used for processing different operations in the processor. In this work, we have presented an approach to improve the speed and performance efficiency of the processor with low propagation delay. We have designed and realized 4x4 and 8x8 Vedic multiplier by KSA based on X-OR gate followed by comparative analysis of different design constraints. We understood the concept of the adder and its categorization including prefix adder in detail with designing and implementation procedure. We have learned and analyzed the structure of multipliers and adders with their mathematical equation including characteristics. We have presented the designing approach based on Vedic sutra (Urdhva Tiryagbhyam) for high efficient multipliers in a digital processor. We synthesized and investigated the presented work in Xilinx 14.2 and ModelSim 6.1e tool used for modeling and simulation analysis.

Keywords*:* Prefix Adder, Kogge-Stone Adder, Carry Select Adder, Multiplier, Vedic Mathematics sutra (Urdhva Tiryagbhyam)

I. INTRODUCTION

The area of the advance technology processor of the digital devices must be efficient and less area. In digital processor, binary information will be there and multiplication is the most significant operation after addition. With newest encroachment in technology, digital processor acts with critical functions in the engineering domain. In digital processor architecture adder performs significant operation such as address calculation, ALU functions [1] and table directory. Different algorithms proposed for addition in existing work including half adder, full adder and parallel adder to prefix adder. Apart from addition multiplication is another operation responsible for digital processing. For fast accurate response digital processor depends on the performance of multipliers and adders [2]. In VLSI circuits and design the main area of research work deals with the improvement in the performance of multipliers and adders. Over the years different circuitry for adders and multipliers proposed to increase the response of digital processor with the less area as number of gates reduced. With the evolution of ICs (Integrated Circuits) the computing engineering mainly in customer electronics has been practicing strange burst in progress and augmentation.

In [3] the requirement of low power consumption and less propagation delay discussed with the new concepts of the digital processor with certain limitations. With the usage of high-speed multipliers, the performance of digital processors has enhanced. For better operations and efficiency of multipliers, many researchers worked on developing new algorithms.

II. PREVIOUS METHODOLOGY

In [4] present scenario of the digital system the fundamental arithmetic operation blocks such as adders or multipliers have increased the speed of the digital processor in reference to the performance of digital circuit designing and applications. Thus digital processors depend on the high-speed circuitry of adders and multipliers. As we know ALU (Arithmetic Logic Unit) is one of the main units of the digital processor as it performs binary addition and multiplications which help in the processing of binary data in any digital system or device. With the latest design and developments in the digital technology, researchers have done great work including IC's having small size and fewer power consumptions as they are considered as other factors involve in the performance analysis of digital processors.

With reference to the design and performance evaluation of digital processor, it is required to select proper adder and a multiplier circuit for future applications in the field of the digital world. In this section of the chapter, we have described the general concept of binary adders including half and full adders followed by further classification in detail. Finally, we have presented multipliers based on the concept of Vedic algorithm. In the initial level we described the basic concept of binary addition using adders including Half adder and Full adder, then we have talked about the addition in the parallel form using parallel adders including prefix adders and further covered the multiplication operation in Vedic sutras as presented in [5][6].

III. PROPOSED ALGORITHM

In any digital processor, the stage of addition and multiplication circuitry significantly influenced the performance in terms of speed as discussed in [13]. The speed based performance of the digital processing circuits mainly depends on the design methods of adders used whereas adders are responsible for the addition of two or more bits. According to the [32], adders and multipliers are the circuits required in the operation of any digital processing unit including signal processor or image processing. For fast response, we have referred the concepts of Vedic maths as 16 sutras presented which computes arithmetic operations in a different way. We have described and enlisted all the Vedic sutras in the previous chapter. So in this thesis work, we kept in our mind and focused on mitigating the area and delay in case of the digital processor. For digital circuit devices and applications propagation delay is a very important factor as if the delay is the low speed of the circuit will be high or vice-versa. Hence, the delay should be minimizing for high-speed performance of the processor.

We have considered N-bit addition for the description of delay and its importance in Fig. 1 . If we consider the addition of one high bit value for example A_0 A_1 A_2A_3 is high bit value information and another B_0 $B_1B_2B_3$ then there will be carried bit value as $C_0C_1C_2C_3$ is obtained. Every bitwise addition has produced a carry bit value which propagates to the next bit thus delay occurred. Principally we have discussed the general concept of half adder for 2-bit addition and a full adder for 3-bit addition. Also, there are other adders including serial and parallel which offers a fast response in the designing of the processor. In the present world of ULSI circuits and designs, we must think about the fast processing module with high-performance ability and compatibility integrated with the advanced technology as per the demands.

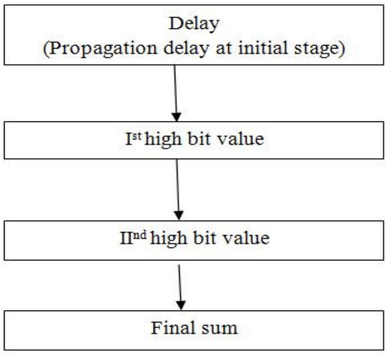


Fig. 1. Delay for bit binary addition circuit.

Since the digital processors are configured with combinational circuits including multipliers with adders having challenges of speed and delay in the operations. For this, many researchers proposed some parallel binary adders to design and develop fast and high-speed multipliers with adders such as ripple carry adder, look ahead carry adders and much more. But in these adders based multipliers number of gates increases which further increase the delay and reduces the speed of response as they used conventional method for multiplication. In our work we have considered all these challenges and worked with an approach towards high efficient and low propagation delay digital processor in which to overcome all the discussed challenges we have proposed to implement prefix adder using the concept of Vedic sutra to design and develop a digital processor having low delay, reduced number of gate, fewer number of slices and LUT's to enhance the performance of the digital processor.

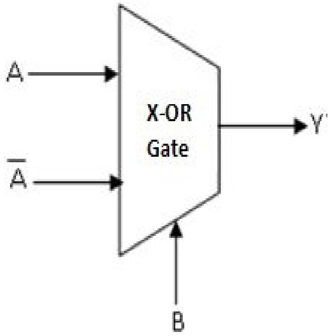


Fig. 2. XOR Gate by using Multiplexer.

The main motive of this thesis work is to design and implement high efficient digital processor with a low delay which can be integrated with many digital applications. In our work we have proposed Kogge Stone Adder as a prefix adder defined in [20] and named by the researchers Kogge P and Stone H in 1973 by using an Ex-OR gate (performs multiplexer operations) to mitigate the delay.

Ex-OR Gate by using Multiplexer

As proposed in [25] Ex-OR gate stands for exclusive operation logic gate expressed by plus symbol with encirclement. In designing digital circuits the complexity can be mitigated by using a proper number of gates design as a low number of gates consumes low area and power with less delay for efficient processing. In general Ex-OR gate can be designed and implemented with minimum 5 gates including 2 AND, 2 NOT and 1 OR gate but in our work we have implemented Ex-OR gate operations using Multiplexer as it has the ability to behave as a universal element in digital circuit designs. To get the 2-inputs based Ex-OR operation performed by the Multiplexer we have given one input value along with its inverse value in the 2-input multiplexer and another input value given as select line for the multiplexer. Hence the output value of multiplexer would be equivalence to the output value of Ex-OR gate, this designing concept is known as Differential Cascode Voltage Switch logic circuit in which we get true and complementary output values as per the requirement of the design applications. Here ripple carry adder produces the sum of two binary numbers as the functional diagram shown in figure- 3 the cascaded connections of full adders in which output carry connected as the input carry for the next adder. This layout allows fast response time and quite simple to design.

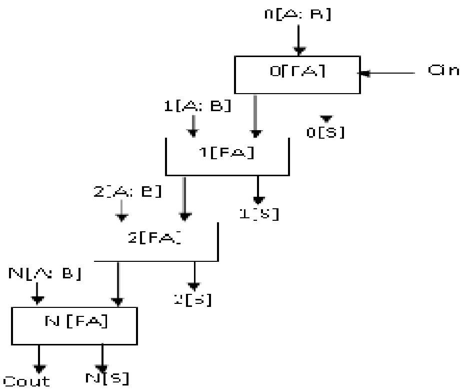


Fig. 3. Layout of Ripple Carry Adder functions.

In the case of ripple carry adder is the slowest one amid of all the available adders it is due to the fact that each full adder has to wait for the generation of carry value from preceding full adder as carry input value.

$$S_0= X_0 \oplus Y_0 \oplus C_{in} \tag{3.1}$$

$$C_0= (X_0 \oplus Y_0) +(Y_0 \oplus C_{in})+(C_{in} \oplus X_0) \tag{3.2}$$

$$S_1=X_1 \oplus Y_1 \oplus C_0 \tag{3.3}$$

$$C_1=(X_1 \oplus Y_1)+(Y_1 \oplus C_0) + (C_0 \oplus X_1) \tag{3.4}$$

$$S_2=X_2 \oplus Y_2 \oplus C_1 \tag{3.5}$$

$$C_2= (X_2 \oplus Y_2) + (Y_2 \oplus C_1) + (C_1 \oplus X_2) \tag{3.6}$$

$$S_n=X_n \oplus Y_n \oplus C_{n+1} \tag{3.8}$$

$$C_n=(X_n \oplus Y_n) +(X_n \oplus C_{n+1}) + (C_{n+1} \oplus X_n) \tag{3.9}$$

When consumption of area and delay required to be low ripple carry adder finds its applications as it as compare to other adders as the only limitation with ripple carry adder is the complexity in design as a number of gates used are more and it does not offer efficient speed with the area.

KOGGE-STONE ADDER (KSA)

Kogge Stone Adder as a prefix adder defined in [20] and named by the researchers Kogge P and Stone H in 1973, the concept was developed as KSA is the kind of prefix adder with some special and fast features. It comprises of three parts including preprocessing part, carry generator part and post processing part.

$$P=X_i \oplus Y_i \tag{3.10}$$

$$G = X_i \ Y_i \tag{3.11}$$

$$C_i = G_i \tag{3.12}$$

$$S_i = P_i \oplus C_{i+1} \tag{3.13}$$

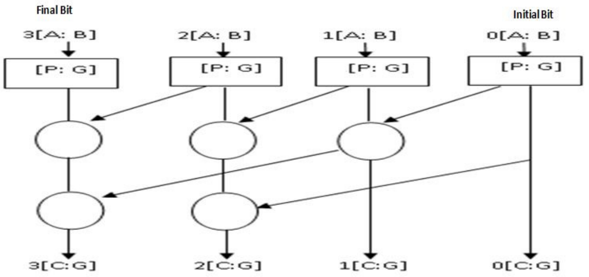


Fig. 4. A basic diagram of Kogge Stone Adder.

In prefix Kogge Stone Adder, Ex-OR gate can be substituted by Multiplexer which offers numerous configuration and confers the true value and complement value at any instant of time.

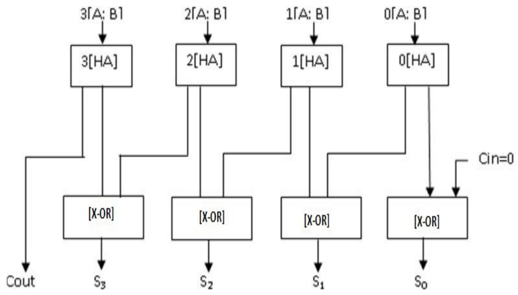


Fig. 5. Kogge-Stone Adder using X-OR gate.

VEDIC MULTIPLICATION

As discussed in the previous chapter we have used Urdhva Tiryakbhayam (UT) sutra of Vedic maths for multiplication concept which is appropriate for any number system such as decimal, binary or hexadecimal. To understand the concept well we have taken example based learning with two decimal numbers as shown in the figure below.

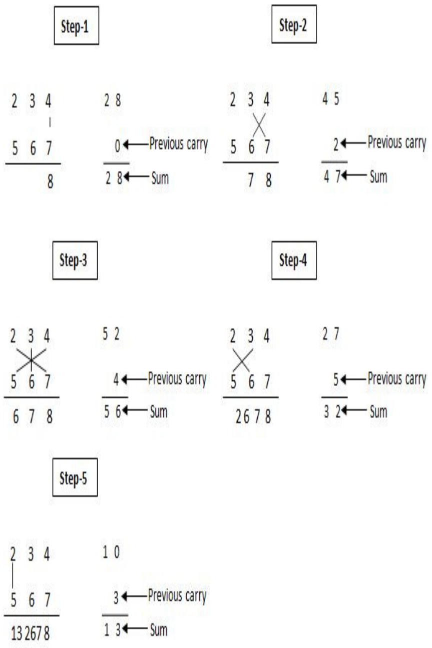


Fig. 6. Multiplication of two decimal numbers by UT Vedic sutra.

$S_0 = A_0 + B_0$ 3.14

$C_1S_1 = A_1B_0 + A_0B_1$ 3.15

$C_2S_2 = C_1 + A_1B_1$ 3.16

Above examples in the given figure 6 shows that how UT Vedic algorithm can be done by following all the steps. We have used this concept in our research work to improve the performance of the digital processor. The ability of UT Vedic sutra to compatible with digital devices and processor make it special and fast calculations due to the partial multiplication and their addition in parallel form simultaneously. The words Urdhav is taken from the Sanskrit language means vertical or perpendicular and Tiryagbhyam means diagonally in English so this sutra approach for the computation diagonally and perpendicularly for any two given numbers. As presented in [7] this approach for the multiplication based on the principle of partial products and sum occurred simultaneously thus increases the computation responses and performance.

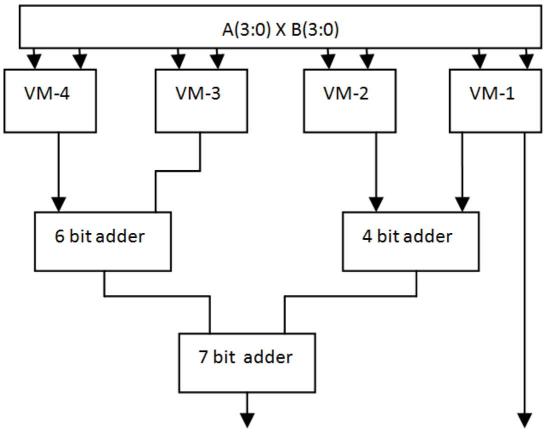


Fig. 7. Vedic Multiplier for 4bits.

In this 4-bit Vedic multiplier we have taken 2 binary numbers at a time as per the diagram and furthermore number of adders required for the computation, we have observed these problems and resolved them with the proposed method with respect to a number of gates and area consumed in our implementation processes. As a novel theory, this Vedic sutra provides the throughput, also the generation of partial product terms and their sum as displayed in figure 1. One of the unique features of this sutra is the reduction in the resources required for processors to operate at high frequency since this concept is very different from conventional type, also it improves the overall performance and efficiency of the digital processor. The final outcome of the above-considered equation is $C_2S_2S_1S_0$. Similarly, for the calculation of 4x4, 8x8, 16x16 and so on in other cases achieved with this process.

IV. DESIGN AND IMPLEMENTATION

In the recent time, technology developments from micro scale to nanoscale in circuit designing [1] motivated us to work towards the efficient digital processor. We have proposed and implemented a fast multiplier using Kogge-Stone Adder incorporated with UT Vedic sutra. Due to the usage of Vedic sutra the execution is simple and fast with two steps mainly generation of partial product terms and their addition involves in the multiplication operation as discussed in [5]. In this chapter, we have described the design and implementation of Vedic multiplier using Kogge -Stone adder to fulfill the objectives of this thesis work including an approach towards high speed and efficient digital processor. Vedic maths can be applied in engineering research work for the development of new algorithms and systems as part of their possible applications for its potential in the field of Technology and Science.

Customized Adder

For the design and implementation of 4x4 bit Kogge-Stone Adder using the X-OR gate, as Vedic sutra is used between 4-bit multiplicand state as A [3:0], B [3:0] and produced the outcome of multiplication in P[7:0] bit.

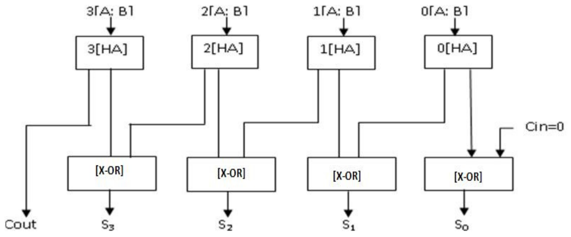


Fig. 8. 4 bit Kogge-Stone Adder using X-OR gate.

The four 2-bit fault tolerant reversible Vedic multiplier is required to implement a 4-bit multiplier.

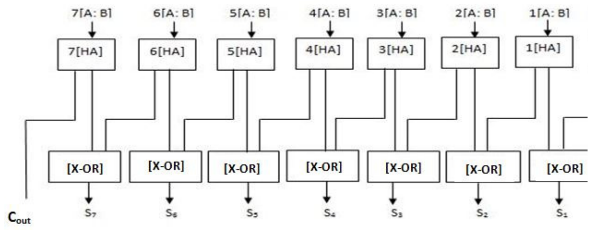


Fig. 9. 8 bit Kogge-Stone Adder using X-OR gate.

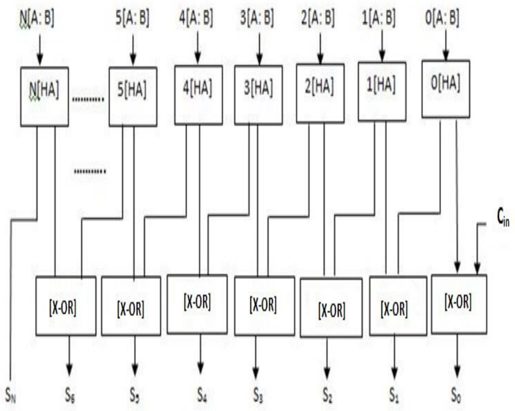


Fig. 10. N- bits Kogge-Stone Adder using X-OR gate.

Similarly, we have shown the functional blocks of 8 bit Kogge-Stone Adder using the X-OR gate and n bits Kogge-Stone Adder using the X-OR gate in over figures.

Customized Multiplier for High-Speed Digital Processor

In this section, we have shown the proposed Vedic multiplier based on UT Vedic sutra for multiplication operation in a digital processor. In reference to the figure 5.4, we have used small blocks to realize the complex diagram as in 4x4 Vedic multiplier realized by small blocks of 2x2 multipliers based on Verilog Hardware Description Language (VHDL).

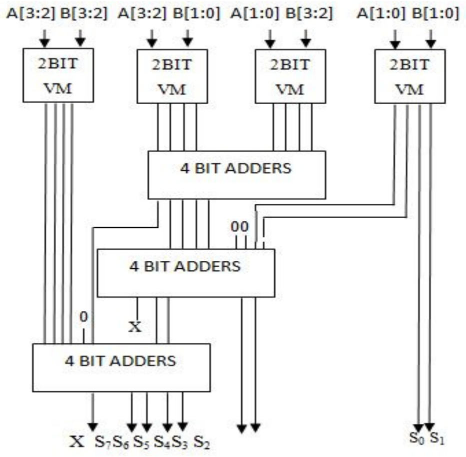


Fig. 11. 4x4 Vedic multiplier using Kogge-Stone Adder with XOR gate.

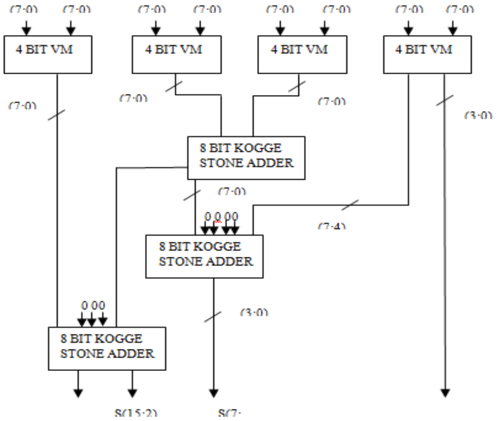


Fig. 12. 8X8 Vedic Multiplier by using Kogge Stone Adder.

UT Vedic sutra based 4x4, 8x8 bit multiplication as in Vedic multipliers architectures demonstrated in this section. The multiplication approach of UT Vedic sutra is quite different as perpendicular and diagonally which produces partial product terms and sum, thus it improves the parallel processing by reducing the delay. Using such structure in the design improves the efficiency and performance of the digital processor with less delay and area required for IC.

For the creation of the new project, we have shown here the snapshot in figure 12 where we mark the name of the project. This snapshot of the window describes the details such as the location of the file and also we can add the description of the project in the given area. There is a drop down area to select the source levels. We have used Xilinx ISE as a simulation tool for the development of the proposed work.

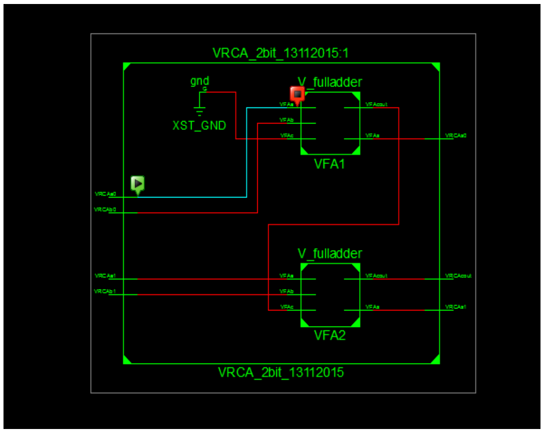
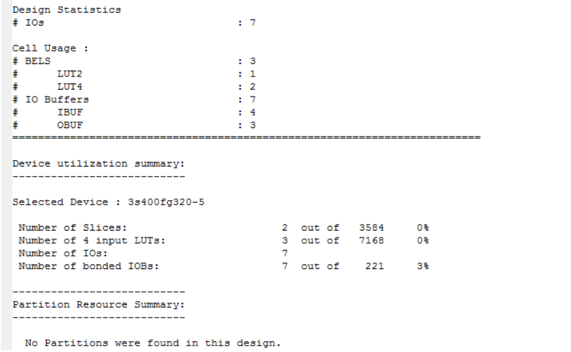


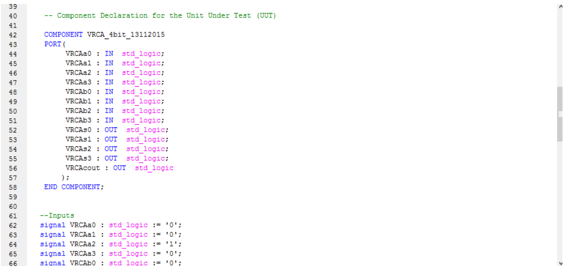
Fig. 13. RTL view of circuit with input and output.

In figure - 13 full adder circuitry is shown by the square box and all the full adders are connected in cascaded form. For '0' as input value, we have connected it with ground point and using the only block we can easily calculate the number of gates. As discussed earlier three bits can be added in full adder at a time so that carry of the first full adder is given as input value for the next full adder and so on in the designed circuit. Due to this a combinational path delay produced during the forwarding of each carry bit as an input value.



```
Design Statistics
# IOs                              : 7

Cell Usage :
# BELS                             : 3
#      LUT2                        : 1
#      LUT4                        : 2
# IO Buffers                       : 7
#      IBUF                        : 4
#      OBUF                        : 3
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 3s400fg320-5

 Number of Slices:                        2  out of   3584     0%
 Number of 4 input LUTs:                  3  out of   7168     0%
 Number of IOs:                           7
 Number of bonded IOBs:                   7  out of    221     3%

---------------------------
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.
```

As per the, we have selected devices based on the requirement of the proposed system model. This snapshot of the window shows the designed constraints with values as per their availability. The destined constraints include a number of slices, input/ output buffers, look up table and path delay. For the high speed and efficiency, the value of these constraints should be low as possible. The higher value of these constraints increases the overall cost of the system model with less mobility. The look-up table associated with the number of slices as the array of the same logic circuit as large look-up table takes greatest processing time but for high-speed operation this should be low. The path delay defined as the time taken by the device to give the response as output for the particular input so this should be low. Since speed and path delay are inverse to each other. Finally, the number of input or output buffers is less as it reduces the complexity of the logic circuit by which fault and error can be easily detected.



```

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT VRCA_4bit_13112015
    PORT(
         VRCAa0 : IN  std_logic;
         VRCAa1 : IN  std_logic;
         VRCAa2 : IN  std_logic;
         VRCAa3 : IN  std_logic;
         VRCAb0 : IN  std_logic;
         VRCAb1 : IN  std_logic;
         VRCAb2 : IN  std_logic;
         VRCAb3 : IN  std_logic;
         VRCAs0 : OUT  std_logic;
         VRCAs1 : OUT  std_logic;
         VRCAs2 : OUT  std_logic;
         VRCAs3 : OUT  std_logic;
         VRCAcout : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal VRCAa0 : std_logic := '0';
   signal VRCAa1 : std_logic := '0';
   signal VRCAa2 : std_logic := '1';
   signal VRCAa3 : std_logic := '0';
   signal VRCAb0 : std_logic := '0';
```

The idea of circuit and simulation testing for the verification purpose of the particular outcome is shown in figure. whereby changing the value of clock port in the program can change the input signal port. For the performance analysis we have enlisted here all the designed constraints with their definitions, functions, and specifications in detail:

- Look Up Table (LUT)
  It reduces the complexity of calculations and improves the processing duration. This designed constraints considered in many typical applications including signal and image processing and modeling.
- Slices
  It defines the connections also parallel connections of devices considered as an array. Normally a look-up table consists of a number of slices. If number of slices used the area consumption will be more.
- Propagation Delay
  For the propagation in the digital circuit, it varies from level '0' to '1' and vice-versa in zero time. But in practical it takes some finite value time period to switch the levels. So the time period it takes to change its output level is said to be switching time period. Propagation delay is defined as the time period taken by the output to give on response occurrence of the input pulse. It is the most significant constraints for logic circuit of any digital system to operate. If the value of propagation delay is low the logic circuit is more efficient and high in speed and vice-versa.
- Input/ Output Buffers
  For the simple logic circuit and design, this should be less in value.
- CPU Time-
  It defines the time taken by the CPU for processing the operations so for the fast response it should be low as possible.

For the result and performance analysis, we have to discuss the values of these design constraints as per the simulation performed in the proposed work. In our simulation work we have to analyze and compared the constraints with their value for both ripples carry adder and kogge-stone adder in tabular form as shown here below. We have performed simulation using Xilinx 14.2i Spartan 3E series. As per the data values in the table, we have done the comparative analysis between 4-bit RCA and 4-bit KSA, observed that number of slices less with less LUT's in the case of KSA which results in less delay and CPU time with an equal number of IOB's in the same memory device.

Table: Comparison between 4 bit RCA Adder and 4 bit KSA Modified Adder

Adder	**Slices**	**LUTs**	**IOBs**	**Delay**	**Memory**	**CPU Time**
4 bit RCA	4	7	13	7.848 ns	230016 KB	10.00 sec
4 bit KSA	3	5	13	5.776 ns	230016 KB	9.22 sec
8 bit RCA	9	15	25	12.142 ns	230016 KB	10.09 sec
8 bit KSA	5	9	25	5.776 ns	230016 KB	9.33 sec
16 bit RCA	18	31	49	20.629 ns	230056 KB	11 sec
16 bit KSA	10	17	49	5.776 ns	229824 KB	9.66 sec
32 bit RCA	36	63	97	37.604 ns	232064 KB	10.98 sec
32 bit KSA	19	33	97	5.776 ns	231040 KB	10.34 sec

We have done comparative analysis as in the case of KSA slices and LUT's are less in numbers and we have obtained a slight difference in the delay also which improves the performance in the speed of the digital processor

Adder	**Slices**	**LUTs**	**IOBs**	**Delay**	**Memory**	**CPU Time**
4 bit RCA	23	40	16	12.421 ns	232064 KB	11.14 sec
4 bit KSA	19	33	16	11.737 ns	231872 KB	11.34 sec

V. CONCLUSION

In the digital circuit systems and campaign, digital processor based on low power VLSI plays a vital role for the high and efficient performance. The designing points for assumptions in the case of a better recital in VLSI are speed, area and cost so in our work we have analyzed and focused on these factors. The present scenario of the low power devices advances with latest designing, cost effectiveness cooling and packaging, reliability, and portability.

In the thesis work we have performed detailed literature survey and studied prefix adders including CLA, CSA. We have developed an approach towards high efficient and low delay digital processor for this modification of Vedic multiplier using KSA based on X-OR gate designed and implemented which reduces the number of slices, area, and delay. Mainly, this research has discovered and completed following objectives. We thoroughly covered all the adders in the literature survey with classifications and their types including designing procedure. Undergone a review analysis of UT Vedic maths sutra with its multiplication concept and used it in the designing of the multiplier. Design and implementation of proposed KSA using X-OR gate for 4-bits, 8-bits, 16-bits and 32 bits and analyzed the designed constraints with values. Evaluated the performance of KSA using X-OR gate for binary addition to mitigating the size and enhancement of efficiency and speed in digital processing. Performed comparative analysis of the KSA using an X-OR gate with respect to the RCA at different design constraints values.

REFERENCES

[1]. M. Morris Mano, *Digital Design second edition*, Prentice Hall, 1991.

[2]. Carver Mead and Lynn Conway, *Introduction to VLSI design*, Addison-Wesley Company, 1980.

[3]. Stefan Sjoholm and Lennart Lind, *VHDL for designers*, Prentice Hall, 1997.

[4]. Kaushik Roy, Sharat C. Prasad, Low-Power CMOS VLSI Circuit Design, John Wiley & Sons, Inc, 2000.

[5]. A. P. Chandrakasan, S. Sheng, And R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-state Circuits,* vol. **27**, no. 04, pp. 480-484, April 1999.

[6]. Akhilesh Tyagi,1990.A Reduced Area Scheme for Carry-Select Adders, I*EEE International Conference on Computer design,* pp.255-258.

[7]. Youngjoon Kim and Lee-Sup Kim, 2001.A low power carry select adder with reduced area, IEEE International Symposium on Circuits and Systems, vol.**4**, pp.218-221.

[8]. Belle W.Y. Wei and Clark D.Thompson,1990.Area-Time Optimal Adder Design, *IEEE transactions on Computers,* vol. **39**, pp. 666- 675.

[9]. Madhu Thakur and Javed Ashraf, 2012.Design of Braun Multiplier with Kogge-Stone Adder and Its Implementation on FPGA *International Journal of Scientific and Engineering Research*, Vol. **3**, No. 10, pp. 03-06, ISSN 2229-5518.

[10]. Kogge P and Stone H, 1973.A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Relations, *IEEE Transactions on Computers*, Vol. **C-22**, No. 8, pp. 786-793.

[11]. Sumeer Goel, Mohammed A. Elgamel, Magdy A. Bayoumi, Yasser Hanafy, 2006.Design Methodologies for High- Performance Noise-Tolerant XOR-XNOR Circuits, *IEEE Transactions on Circuits and Systems*- I, Vol. **53**, No. 4, pp. 867-878.

[12]. Somayeh Babazadeh and Majid Haghparast, 2012.Design of a Nanometric Fault Tolerant Reversible Multiplier Circuit, *Journal of Basic and Applied Scientific Research.*

[13]. Mohammad Shamim Imtiaz, Md Abdul Aziz Suzon, Mahmudur Rahman,2012. Design of Energy efficient Full adder using hybrid CMOS logic style. *International Journal of Advances in Engineering and Technology*, Jan 2012.

[14]. M.J. Schulte, P.I. Balzola, A. Akkas, and R.W. Brocato, "Integer Multiplication with Overflow Detection or Saturation", *IEEE Trans. Computers*, vol. **49**, no. 7, July 2000.

[15]. Ila Gupta, Neha Arora, Prof. B.P. Singh, 2012.Analysis of Several 2:1 Multiplexer Circuits at 90nm and 45nm Technologies, *International Journal of Scientific and Research Publications*, Volume **2**, Issue 2, February 2012.

[16]. C.S. Wallace, "A suggestion for a fast multiplier", *IEEE Trans. Electron. Comp.,* vol. **EC-13**, pp 14-17, Feb. 1964.

[17]. R. K. J. Raghunath, *et al.* "A compact carry-save multiplier architecture and its applications," *Proc. IEEE 40th Midwest Symposium. Circuits and Systems*, vol. **2**, pp. 794-797, Aug. 1997.

[18]. J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, A Design Perspective, Prentice Hall, Upper Saddle River, NJ, 2003.

[19]. A. R. Omondi, Computer Arithmetic Systems. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[20]. A. Chandrakasan, W. J. Bowhill, and F. Fox, Design of High-Performance Microprocessor Circuits, IEEE Press, Piscataway, NJ, 2001.

[21]. I. Koren, Computer Arithmetic Algorithms, A.K Peters, Ltd., Natick, MA, 2002.

[22]. A. Weinberger and J. Smith, "A logic for high-speed addition," in National Bureau of Standards, 1958, pp. 3-12.

[23]. J. Sklansky, "Conditional-sum addition logic", *IRE Transaction on Electronic Computers,* vol.**EC-9**, pp. 226-231, 1960.

[24]. S. Knowles, "A Family of Adders", Proc. 14th Symposium Computer Arithmetic, 30-34, Apr. 1999.

[25]. O.J. Bedrij, "Carry-select adder", *IRE Transaction on Electronic Computers,* June 1962.

[26]. R.E. Ladner and M.J. Fisher, "Parallel Prefix Computation," *J. ACM,* vol. **27**, no. 4, pp. 831-838, Oct. 1980.

[27]. P.M. Kogge and H.S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Trans. Computers*, vol. **22**, no. 8, 786-792, Aug. 1973.

[28]. T. Han and D. Carlson, "Fast Area-Efficient VLSI Adders," *Proc. Symposium Computer Arithmetic*, pp. 49-56, May 1987.

[29]. R.P. Brent and H.T. Kung, "A Regular Layout for Parallel Adders," *IEEE Trans. Computers,* vol. **31**, no. 3, pp. 260-264, Mar. 1982.

[30]. Reto Zimmermann, "Non-Heuristic Optimization and Synthesis of Parallel Prefix Adders", International Workshop on Logic and Architecture Synthesis (IWLAS'96), Grenoble, December 1996.

[31]. C. S. Wallace. "A Suggestion for a Fast Multiplier", *IEEE Transactions on Electronic Computers,* **EC-13**: 14–17, February 1964.

[32]. J. Eyre and J. Bier, "The evolution of DSP processors", IEEE Signal Processing Magazine, vol. **17**, no. 2, pp. 43-51, Mar. 2000.

[33]. G. W. Bewick, Fast Multiplication: Algorithms and Implementation, Ph.D. thesis, Stanford University, Stanford, CA, Feb. 1994.

[34]. A. Weinberger, "4:2 Carry-Save Adder Module", *IBM Technical Disclosure Bulletin,* vol. **23**: 3811–3814, Jan.1981.

[35]. O.L. MacSorley, "High-speed arithmetic on binary computers," *IRE Transaction on Electronic Computers*, vol. **49**, pp. 67-91, 1961.