Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 2,2025 ISSN:2582-5887 www.uijes.com

DESIGN AND IMPLEMENTATION OF A CLOCKED SEQUENTIAL AND COMBINATIONAL BINARY MULTIPLIER

B.MADHU[1] Dr P.VAMSI KRISHNA[2]
[1] E.C.E Student, Dept of E.C.E, RK College Of Engineering, Kethanakonda, A.P., India
[2] Assistant Professor, Dept of E.C.E, RK College Of Engineering, Kethanakonda, A.P., India

ABSTRACT
For many years VLSI chip designers have been using Metal-oxide semiconductor field effect transistor. Designers have used MOSFET circuits in the designs because MOSFET use little power and they are cheaper to develop or fabricate. To satisfy the demands of making the chip smaller, the designers just simply shrink the dimensions to fit the circuit layouts. The shrinkage is also known as "Scaling". Unfortunately, the continuous scaling of the circuit design will eventually cause problems in terms of area, timing and power. Signal processing is one of the most power hungry applications. Adders and Multipliers are the main building blocks for signal processing applications. Saving area, power and timing in adders and multipliers would reduce the area, power and timing significantly at the chip level .In this present investigation of Clocked Sequential and Combinational binary multiplier structures are selected from existing multiplier types. The selected multipliers are Booth multiplier, Wallace tree multiplier and Dadda multiplier. Benchmarking for these 3 multipliers is done in terms of area, power and timing for 32 bit technology. The HDL code for the Clocked Sequential and Combinational binary multiplier is written in Verilog HDL and simulated in Vivado. To calculate the power and delay of this Sequential multiplier we have used Design Compiler Devices and Applications.

Keywords: Sequential Multipliers, Binary Multiplers, Wallance Tree Multiplers, Clocked Signals.

INTRODUCTION
In the process of evolution of technology, many Information Technology (IT) applications are requiring low-power integrated circuits (ICs), since handling huge amount of data to process. Due to integrated applications the size also increases. The energy efficiency is measured as the product of power consumption and computational time. To provide an energy efficient solution, the designer has to reduce the power consumption and the computing time as well. By reducing logic blocks, the static power can be reduced. However, reduction in power consumption has been achieved by different methods that demonstrates a trade-off relationship with circuit performance [1]. Low power consumption has become the

decisive design goal in wide range of electronic systems and sub-systems. Inbuilt-powered sensing modules stand first in this row for the designers [2].

At present, the technology is advancing very rapidly in very short duration of time. The circuits being design have some billions of components with low area, high speed and low power consumption. Hence area, speed and power plays crucial role in the design of any circuit [1], [2]. In order to satisfy the current trend demand a circuit must be designed with low area and less delay constraints. Arithmetic units are major blocks in any processing units which perform various arithmetic operations. Multiplication operation is important among all arithmetic operations. Several multiplication algorithms are studied in literature survey of multiplier designs like Binary multiplier, array multiplier, Booth's multiplier, Dadda multiplier, Wallace tree multiplier [4]. Wallace tree multiplier is advantageous in different types of multipliers[5].

The operation of Wallace tree multiplier is same in the first stage of multiplication which is generating partial products. In the second stage, Wallace tree multiplier adds first three rows partial products. Then the generated sum and carry are added with the next row of partial products. This addition process continues until the generation of final products. For this row-wise addition process half and full adders are employed. Thus, adders are playing a very important role in generation of final product terms. The speed of addition is going to affect the operation of the multiplication.

In order to improve the performance of multiplication operation, the adder structure used in design of Wallace tree multiplier has a major role. In this paper, a new structure of Wallace tree multiplier is proposed in which PPAs are used to add final row of partial products with the previous stage generated sum and carry out terms to generate final product terms. The PPAs are designs which are originally derived from carry look ahead adder concept of generating and propagating of carry bits. In PPAs, a carry generation tree is present which generates carry for all preceding stages which improves the speed of operation. The carry generation tree mainly consists of two components-black cell and grey cell. The black cell and grey cell are interconnected to from carry tree network. Carry generation tree block is also called as parallel carry generation block as it generates carry bits for all stages at a time parallel[6].

LITERATURE SURVEY

Wallace Tree Multiplier Designs: A Performance Comparison Review by Himanshu Bansal, K. G. Sharma, Tripti Sharma

Multiplication process is often used in digital signal processing systems, microprocessors designs, communication systems, and other application specific integrated circuits. Multipliers are complex units and play an important role in deciding the overall area, speed and power consumption of digital designs. This paper presents a comparison review of various Wallace tree multiplier designs in terms of parameters like latency, complexity and power consumption.

Design of 64-bit low power parallel prefix VLSI adder for high speed arithmetic circuits by Nehru, K., A. Shanmugam, and S. Vadivel.

The addition of two binary numbers is the basic and most often used arithmetic operation on microprocessors, digital signal processors and data processing application specific integrated circuits. Parallel prefix adder is a general technique for speeding up binary addition. This method implements logic functions which determine whether groups of bits will generate or propagate a carry. The proposed 64-bit adder is designed using four different types prefix cell operators, even-dot cells, odd-dot cells, even-semi-dot cells and odd-semi-dot cells; it offers robust adder solutions typically used for low power and high-performance design application needs. The comparison can be made with various input ranges of Parallel Prefix adders in terms power, number of transistor, number of nodes. Tanner EDA tool was used for simulating the parallel prefix adder designs in the 250nm technologies.

A comprehensive review on the VLSI design performance of different Parallel Prefix Adders by Rakesh.S,K.S.Vijula Grace,

Adders are an important part of digital systems. In VLSI digital circuits, such adders should satisfy certain design constraints like low power and high speed. In this paper we present a review of the performance of some conventional adders and parallel adders. Parallel Prefix Adders (PPA) are considered to be one of the fastest adders that had been designed and developed. Parallel Prefix Adders were established as the most efficient circuits for binary addition. These adders which are also called Carry Tree Adders were found to have better performance in VLSI designs. This paper investigates the performance of four different Parallel Prefix Adders namely Kogge Stone Adder (KSA), Brent Kung Adder (BKA), Han Carlson Adder (HCA) and Hybrid Han Carlson Adder (HHCA). In this paper the key contribution is the information about the structure of the Parallel Prefix Adders and their performance parameters. This paper can serve as a reference to the beginners in the digital electronics and VLSI area to gain more knowledge on the Carry Tree Adders.

Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter by Kesava, R. Bala Sai, B. Lingeswara Rao, K. Bala Sindhuri, and N. Udaya Kumar

Multipliers are major blocks in the most of the digital and high performance systems such as Microprocessors, Signal processing Circuits, FIR filters etc. In the present scenario, Fast multipliers with less power consumption are leading with their performance. Wallace tree multiplier with carry select adder (CSLA) is one of the fastest multiplier but utilizes more area. To improve the performance of this multiplier, CSLA is replaced by binary excess-1 counter(BEC) which not only reduces the area at gate level but also reduces power consumption. Area and power calculations for the Wallace tree multiplier using CSLA with BEC are giving good results compared to regular Wallace tree multiplier.

MULTIPLIER OPERATIONS

Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer to itself a specified number of times. A number (multiplicand) is added to

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com

itself a number of times as specified by another number (multiplier) to form a result (product). In elementary school, students learn to multiply by placing the multiplicand on top of the multiplier. The multiplicand is then multiplied by each digit of the multiplier beginning with the rightmost, Least Significant Digit (LSD). Intermediate results (partial products) are placed one atop the other, offset by one digit to align digits of the same weight. The final product is determined by summation of all the partial-products. Although most people think of multiplication only in base 10, this technique applies equally to any base, including binary. Figure 1.1 shows the data flow for the basic multiplication technique just described. Each black dot represents a single digit.

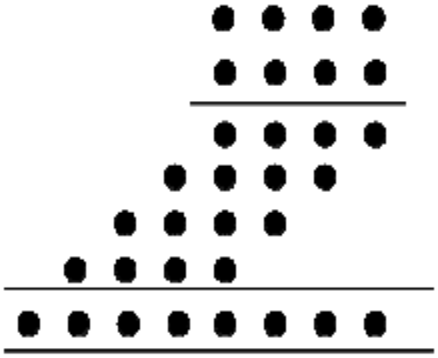


Figure 2.1: basic Multiplication

Here, we assume that MSB represent the sign of the digit. The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers. Based upon the above procedure, we can deduce an algorithm for any kind of multiplication. We can check at the initial stage also that whether the product will be positive or negative or after getting the whole result, MSB of the results tells the sign of the product.

MULTIPLICATION PROCESS

The simplest multiplication operation is to directly calculate the product of two numbers by hand. This procedure can be divided into three steps: partial product generation, partial product reduction and the final addition. To further specify the operation process, let us calculate the product of 2 two's complement numbers, for example, 1101_2 (-3_{10}) and 0101_2 (5_{10}), when computing the product by hand, which can be described according to figure 1.2.

The bold italic digits are the sign extension bits of the partial products. The first operand is called the multiplicand and the second the multiplier. The intermediate products are called partial products and the final result is called the product. However, the multiplication process, when this method is directly mapped to hardware, is shown in figure. As can been seen in the figures, the multiplication operation in hardware consists of PP generation, PP reduction and final addition steps. The two rows before the product are called sum and carry bits.

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com

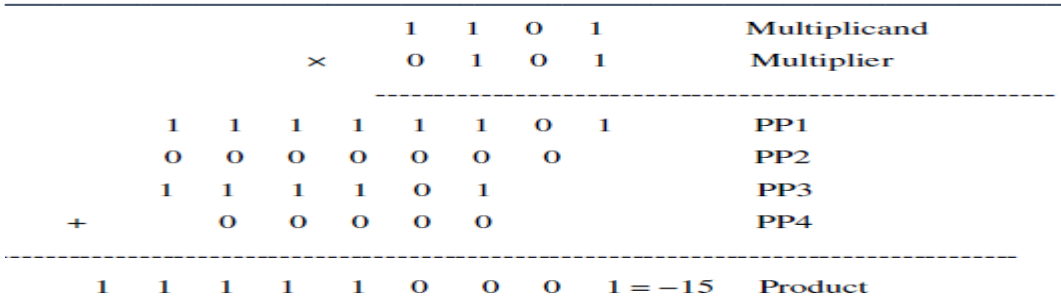


Fig 2.2 Multiplication process

The operation of this method is to take one of the multiplier bits at a time from right to left, multiplying the multiplicand by the single bit of the multiplier and shifting the intermediate product one position to the left of the earlier intermediate products. All the bits of the partial products in each column are added to obtain two bits: sum and carry. Finally, the sum and carry bits in each column have to be summed. Similarly, for the multiplication of an *n*-bit multiplicand and an *m*-bit multiplier, a product with *n* + *m* bits long and *m* partial products can be generated.

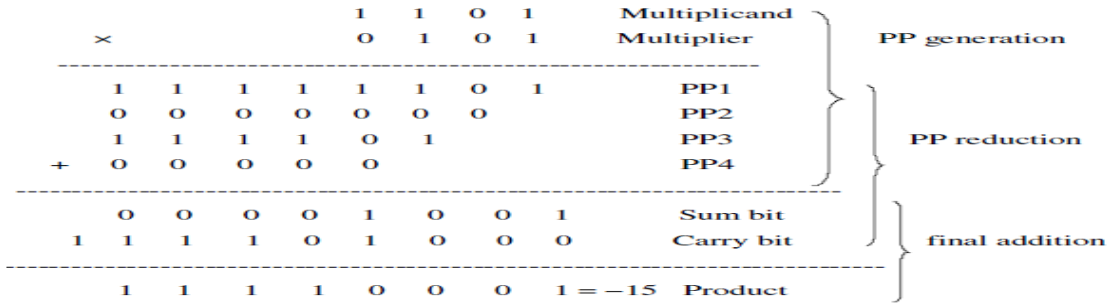


Fig 2.3: Multiplication showing sum and carry

PROPOSED SYSTEM

The idea behind the suggested implementation of binary multiplier is using the very basic definition of multiplication. The multiplication can be basically defined as a repetitive addition in which one of the two multiplied numbers (multiplicand) will be added to itself a number of times equal to the value of the second number (multiplier). Using the definition as it requires a number of additions equal to the value of the multiplier and that is a time consuming operation.

The hardware shows the internal structure for 8-bit of the proposed sequential multiplier which consists of the following units:

1- Two 8-bit registers to hold the data input values for the multiplicand and multiplier (i.e. X & Y). The two registers have the control signals; ‘Load XY’ to store data in to the registers, and ‘Clear’ to clear the content of the registers. One more control signal is used with the multiplier register named ‘Dec’ to count down based on the content of the MSB (7-bit) of register.

2- 16-bit binary adder to do the repetitive addition operations.

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com

3- 16-bit register R [15:0] to hold the accumulated value of repetitive additions. This register has the control signals; 'load R' to store accumulated data in register, 'clear' to clear the content of register, and 'ShiftL' to shift the content of register to the left.
4- Control signals including: 'Zero' signal is active when the value of the seven MSBs of Y equal to zero. 'Odd' signal is active when the value of LSB of Y equal to 1. A signal named 'Done' is used to indicate the end of multiplication process.Based on the simulation results, our proposed design shows better performance than the conventional sequential multiplier in terms of delay time and power consumption. Results of simulation show that the proposed design has a slightly larger number of logic elements, but the delay time, and power consumption are significantly reduced. Since the standard design of adder (i.e. Ripple Carry Adder RCA) is used in implementing the accumulator unit of the proposed multiplier, the area of the proposed multiplier is slightly bigger than the area of conventional multiplier.

The Digital multiplication process majorly has two phases. In phase 1, the input numbers are applied to AND gate to produce partial products. These partial products are added in step by step process by using half and full adders in phase 2 to obtain final product output. In detailed multiplication process of Digital multiplier is explained through Fig.1 for input size of 4-bits. The phase 1 comprises of generation of partial products through multiplying every bit of given input numbers with each other. Four rows of partial products are generated as the size of input is 4- bits. The phase 2 comprises of many sub phases of addition of the partial products obtained in phase. The addition operation is carried out using half and full adders.

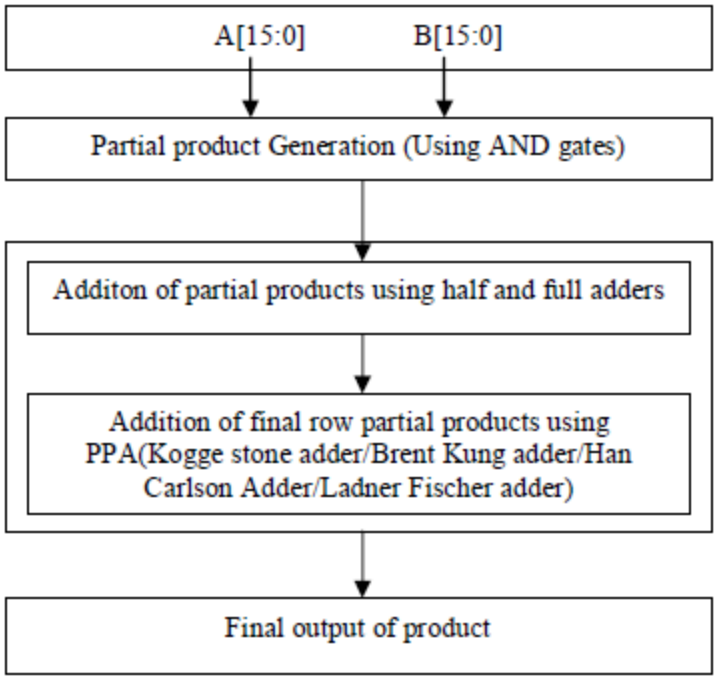


Fig.1. Block diagram of proposed Digital multiplier using PPAs

Initially in phase 2, the addition operation is performed on first three rows of partial products generated in phase 1 which generates result of two rows having sum terms in first row and

carry terms in second row. Then, the last row of partial products of phase 1 result is added with the sum and carry row which again result two rows comprising of one sum row and one carry row. To acquire final product, the sum and carry row are added. In this paper, Digital multiplier structure is modified by using parallel prefix adders to add partial products in final phase addition process to obtain final product. The reason behind to use PPA in place of full adders is to improve the speed of operation. In PPA, the carry input for the next bits is generated at a time with the help of parallel prefix carry tree which consists of black cells and grey cells. There are many types of PPAs are present whose basic design idea is originated from carry look ahead adder. The PPA consists of three main blocks as shown in fig 2.

The multiplier operation is same which is performed in two phases as explained in section 2. In phase 1, partial products are generated with the help of AND gates. In Phase 2 partial products generated in phase 1 are added in step by step approach using half and full adders. In proposed design the final phase of addition of partial products in phase 2 is performed using PPA.

In the proposed multiplier, these last two rows are added with Kogge stone adder (KSA) [7] which is also known as a fast adder. Kogge Stone Adder (KSA) adds the bits in parallel prefix form as shown in Fig.2.

The Parallel Prefix expansion is done in three stages, which is appeared in fig1. The essential produce and engender sign are utilized to create the convey contribution for every adder. Two distinct administrators dark and dim are utilized here.

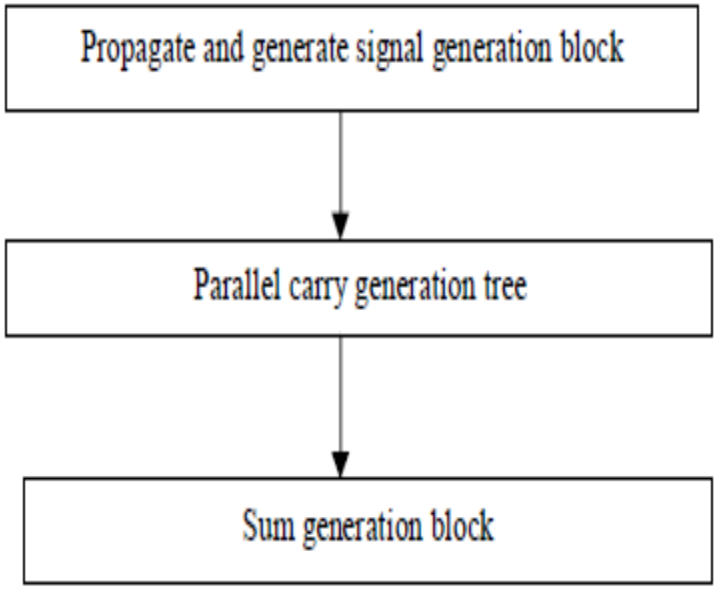


Fig 2. Addition procedure using Parallel Prefix tree structures

In every bit (i) of the two operand block, the two input signals (ai and bi) are added to the corresponding carry-in signal (carryi) to produce sum output (sumi) The equation to produce the sum output is:

$$Sumi = ai \wedge bi \wedge carryi \quad (1)$$

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com

Computation of the carry-in signals at every bit is the most critical and time – consuming operation. In the carry- look ahead scheme of adders (CLA), the focus is to design the carry-in signals for an individual bit additions. This is achieved by generating two signals, the generate (gi) and propagate (pi) using the equations:

$$Gi = ai \wedge bi \quad (2)$$

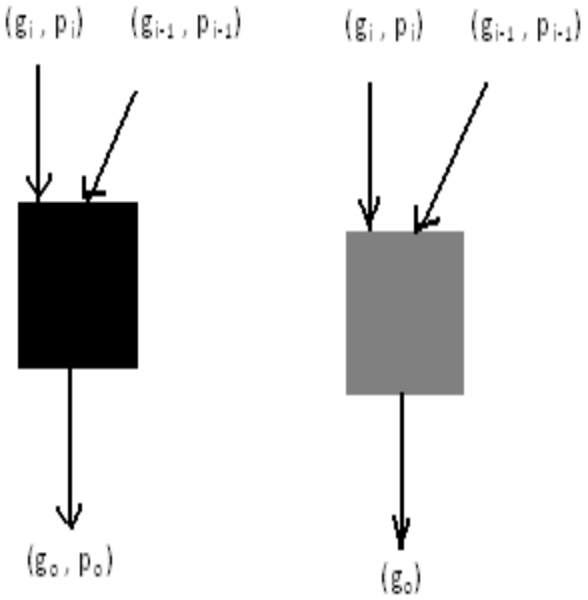
$$Pi = ai \wedge bi \quad (3)$$

The carry in signal for any adder block is calculated by using the formula

$$Ci+1 = gi \; V \; (pi) \quad (4)$$

Where ci must be expanded to calculate ci+1 at any level of addition

Parallel Prefix adders compute carry-in at each level of addition by combining generate and propagate signals in a different manner. Two operators namely *black* and *gray* are used in parallel prefix trees are shown in fig 2(a), fig 2(b) respectively.



(a) black operator (b) gray operator

Fig 3 Operators used in Parallel Prefix trees

The black operator receives two sets of generate and propagate signals (gi , pi),(gi-1 ,pi-1), computes one set of generate and propagate signals (go , po) by the following equations:

$$Go = gi \; V \; (pi \wedge gi\text{-}1) \quad (5)$$

$$Po = pi \wedge pi\text{-}1 \quad (6)$$

The gray operator receives two sets of generate and propagate signals (gi, pi),(gi-1 ,pi-1), computes only one generate signal with the same equation as in equation (5).

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com

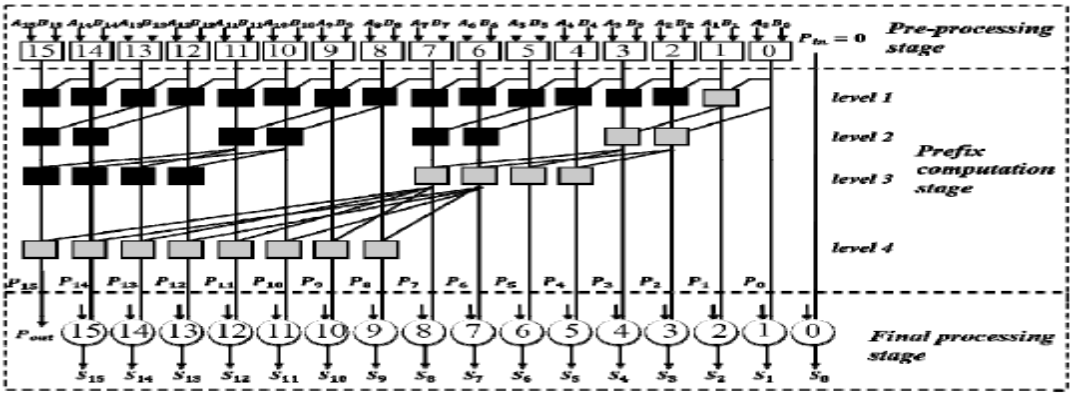


Figure 4 Sequential Binary Multiplication

The construction of the first level of the prefix tree of this adder is similar to the construction of Kogge-Stone adder. The main structural difference begins from the second level of the prefix tree. At the second level of the prefix tree, the groups of two schematic nodes are formed, at the 3rd level – groups compose four schematic nodes and at the 4th level – groups including 8 schematic nodes, etc. This adder first computes giand hi signals for the first stage. Then at the first level of prefix tree, Gi:kand Pi:ksignals of 2-bit are computed at the same time, and then, it computes Gi:kand Pi:ksignals for pairs of columns, then for blocks of 4, then for blocks of 8, then 16, and so on until the final Gi:ksignal for every column is known. Finally, at the last stage this adder computes the sums together with the generated signals obtained from the previous prefix computation stage. The number of levels of the prefix tree corresponds to (log2 n) and the number of schematic nodes

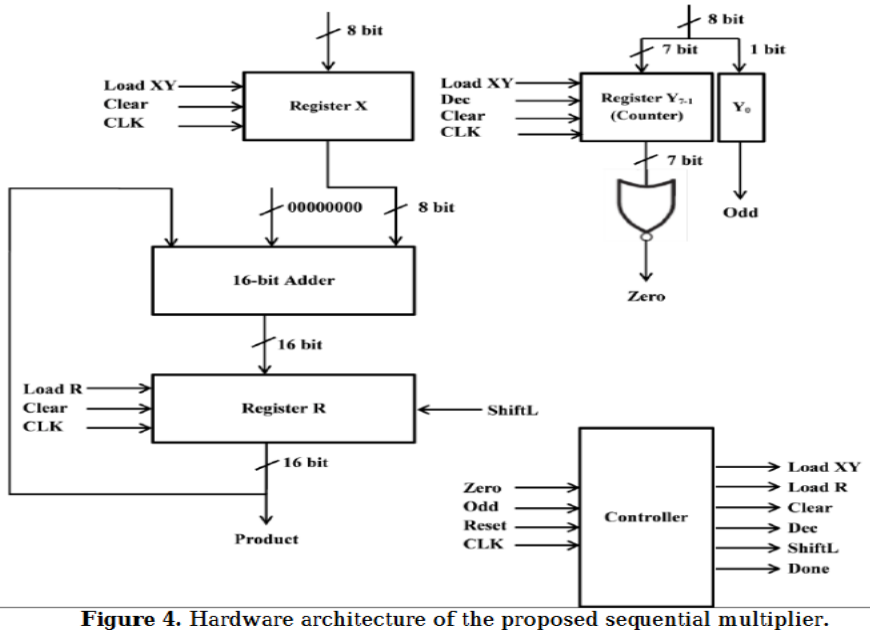


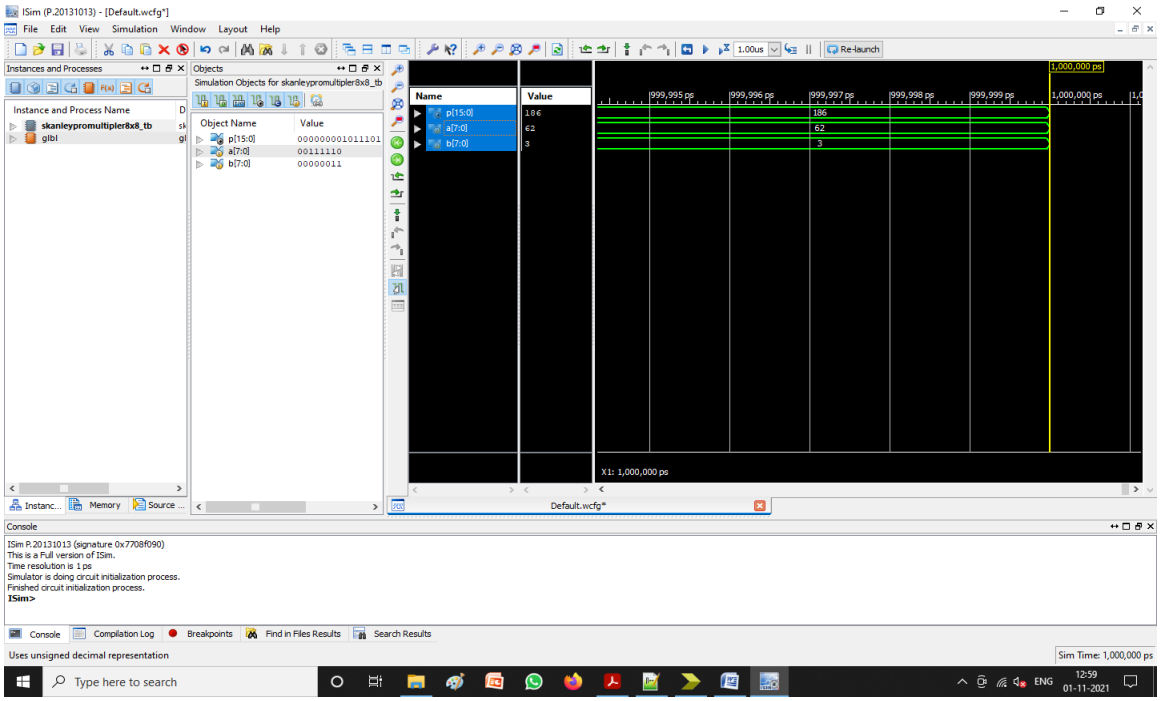
Figure 4. Hardware architecture of the proposed sequential multiplier.

Multipliers are one of the basic units used in implementing different simple and complicated digital circuits. Digital multipliers are the heart of many devices and applications used in our daily life. Since multipliers consume large area and power of implementation, optimization in their design can play a key role in optimizing the speed and area of digital circuits such as

Digital Signal Processing DSP digital communication systems and any other circuit use multiplier in its structure [1, 2]. Multipliers occupied the core of different operations such as convolution, cross correlation, and filter implementation which are mainly used in DSP processes and applications [3, 4]. Different algorithms have been used in implementing the circuit of multiplication. Some of the algorithms simulate the process of doing the multiplication by hand and other uses special algorithms to implement the process of multiplication. The hardware implementation of digital multiplier can be classified in to combinational multiplier design and sequential multiplier design. Combinational multipliers are the direct and basic version of multiplication. Most of combinational multipliers such as Array multipliers, Wallace Tree Multiplier, and Booth multipliers mimic the basic definition of multiplication in which a number of add and shift operations to find the final product. Sequential multipliers are basically using a single circuit of addition to accumulate the product of multiplication. Sequential multipliers are smaller in area than combinational multipliers.

SIMULATION RESULTS

Simulation Result of Multiplier:

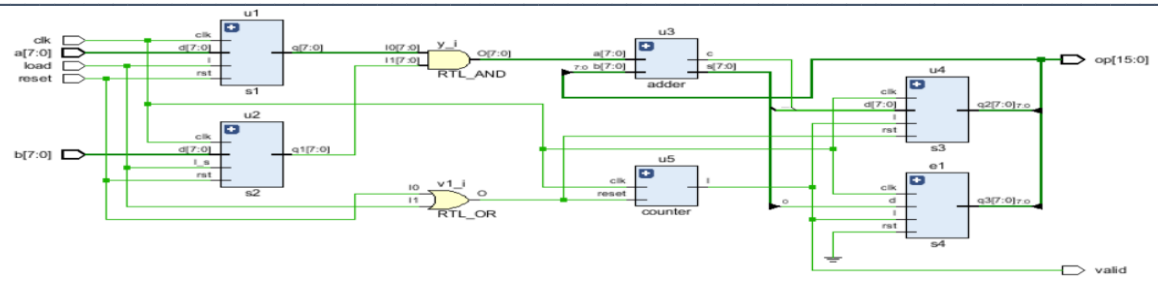


Simulation Result of Multiplier

Here we can give the inputs as A=62, B=3, then the final output is 186.

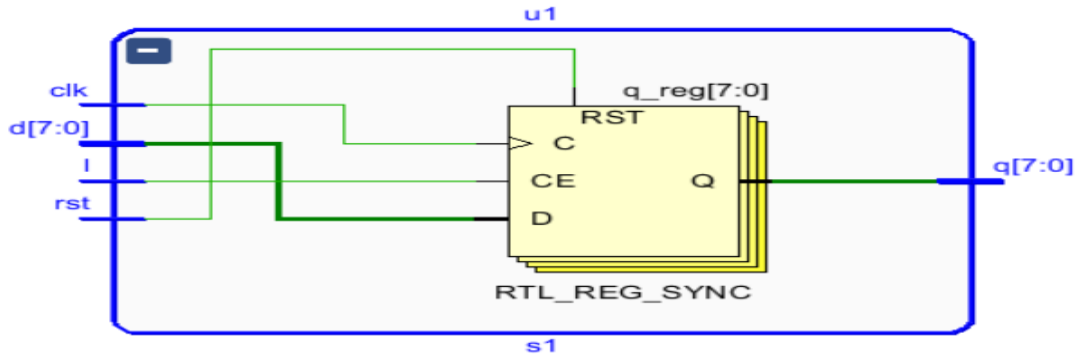
RTL SCHEMATIC OF MULTIPIER:

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com



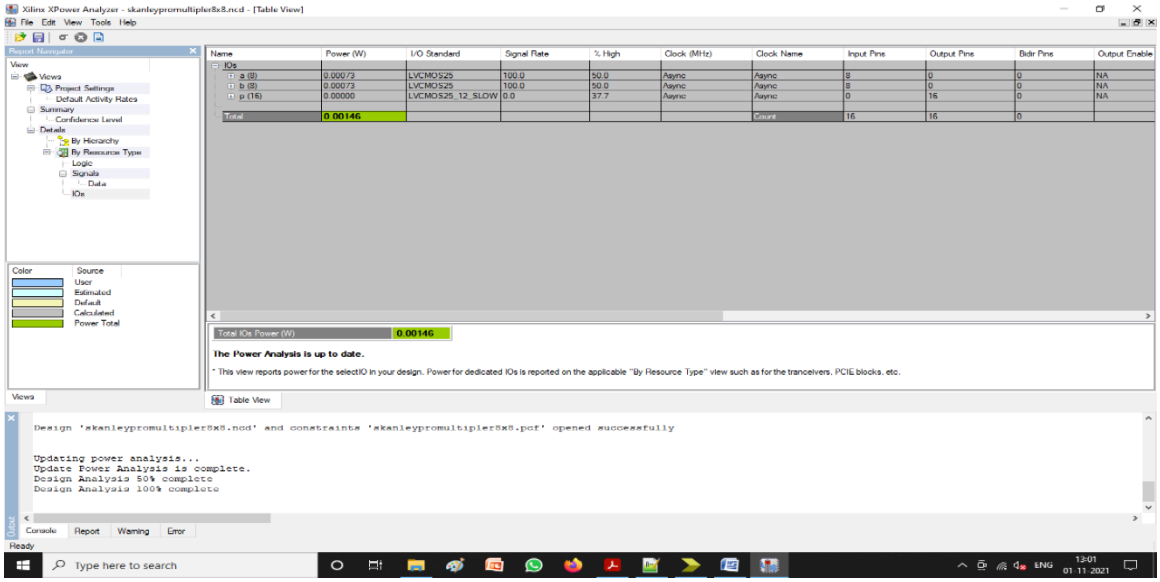
RTL Schematic of Multiplier

INTERNAL DIAGRAM OF RTL SCHEMATIC:



Internal diagram of RTL schematic

Estimation of power:

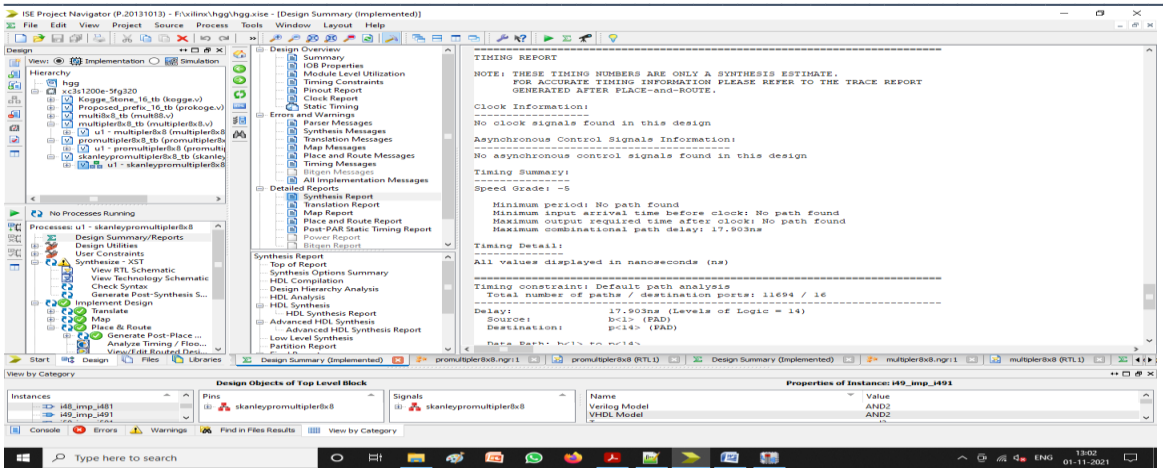


Estimation of power

Estimates power using approximate multiplier and it is 0.00190W power.

Estimation of Delay:

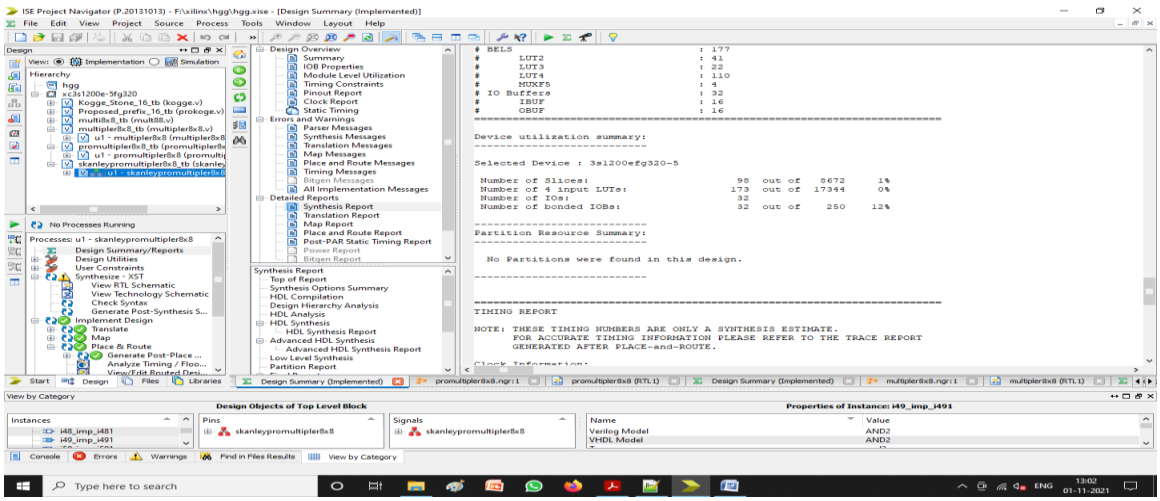
Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 1,2025 ISSN:2582-5887 www.uijes.com



Estimation of Delay

Estimates delay of approximate multiplier and it is 17.903ns delay.

Estimation of Area:



Estimation of Area

CONCLUSION

A new approach is proposed for implementing a fast sequential multiplier. The new approach comes up with a new implementation for sequential multiplier that eliminates all shift operations required by conventional sequential multiplier to only one shift operation with the final accumulated result. Conventional and proposed designs of sequential multiplier are simulated in software tool using Verilog implementations. Results of simulation demonstrate that the proposed design of the sequential multiplier is faster than the conventional design. The proposed sequential multiplier is found to have a reduction in delay time of conventional design by 17.15%. As a future work reducing the total logic elements of the implementation could be achieved by applying more enhancements to the design of adder that occupied most of the area of the proposed multiplier.

REFERENCES

[1] N.Weste and D. Harris, CMOS VLSI Design. Reading, MA: Addison Wesley, 2004.

[2] N. Weste and K. Eshragian, Principles of CMOS VLSI Designs: A System Perspective, 2nd ed., Addison-Wesley, 1985-1993.
[3] Milos D. Ercegovac and Thomas Lang, "Digital arthimetic," Morgan Kaufmann, Elsevier INC, 2004
[4] J. M. Rabaey, Digtal Integrated Circuits—A Design Perspective. Upper Saddle River, NJ: Prentice-Hall, 2001.
[5] Himanshu Bansal, K. G. Sharma, Tripti Sharma," Digital Multiplier Designs: A Performance Comparison Review",Innovative Systems Design and Engineering, Vol.5, No.5, 2014.
[6] Nehru, K., A. Shanmugam, and S. Vadivel. "Design of 64-bit low power parallel prefix VLSI adder for high speed arithmetic circuits." In *2012 International Conference on Computing, Communication and Applications*, pp. 1-4. IEEE, 2012.
[7] Rakesh.S,K.S.Vijula Grace, "A comprehensive review on the VLSI design performance of different Parallel Prefix Adders" ScienceDirect, Materials Today: Proceedings 11 (2019) 1001–1009
[8] Kesava, R. Bala Sai, B. Lingeswara Rao, K. Bala Sindhuri, and N. Udaya Kumar. "Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter." In *2016 Conference on Advances in Signal Processing (CASP)*, pp. 248-253. IEEE, 2016.