International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST

Area Efficient Vedic Multiplier Based on Homogenous Hybrid Adder for RISC V Applications

Sharada
M. Tech Scholar, Department of E&CE (VLSI & EMBEDDED SYSTEM), Sharanbasva University, Kalaburgi, India

Article Info
Volume 9, Issue 4
Page Number : 303-311

Publication Issue
July-August 2022

Article History
Accepted : 05 July 2022
Published : 20 July 2022

ABSTRACT

16-bit RISC processor with Vedic multiplier architecture is used in this project. In addition to multiplier which is implemented using vedic mathematics we are also proposing an adder which is hybrid adder for building higher bit adders in an area efficient which is implemented in addition as well as for compression in vedic mathematic to obtain the output. The multiplier unit is developed utilizing Vedic Sutras, which is the primary accomplishment of this study. The primary premise of Vedic mathematics is to minimize the computational complexity by reducing the usual calculation of conventional mathematics to a very simple calculation. The suggested RISC processor is extremely primitive, and it can only execute 14 instructions. The accomplishment of this study is that in the case of MAC and ALU, power savings and minimized latency are realized as compared to traditional ALU and MAC. Following that, the Vedic MAC and ALU are combined with other processing blocks to create a 16-bit Vedic processor. As a result, the major features of the developed RISC processor are an increase in operating speed, a decrease in power consumption, and a reduction in area consumption.

Keywords : Reduced Instruction Set Computer; Von Neumann architecture; Verilog HDL, Vedic Mathematics, Urdhva-Tiryagbhyam Sutra.

I. INTRODUCTION

Bharati Krishna, a researcher explained that Vedic mathematics is divided into 16 sutras and 13 sub-sutras, which makes it easier to solve mathematical equations. The prehistoric Vedic Mathematics method was enlightened by Swami Bharati Krishna Tirthaji Maharaj, Goverdhan Peath's Shankaracharya. Frequency domain filtering (FIR, IIR) and frequency transformations such as DFT, FFT, and DCT are among the tasks that digital signal processing must undertake. Multiplication is a crucial hardware component for these tasks. As a result, the multiplier's presentation is a critical factor in determining the framework's overall exhibition. This is because the multiplier is the framework's slowest and most time-consuming component. As a result, increasing the multiplier speed and area is a significant challenge for

the framework designers. Binary multipliers are used in digital circuit plan which makes them quick, dependable and easy to carry out in any activity.

The Urdhva Tiryakbhyam sutra's reasoning is quite similar to that of a traditional array multiplier. The Nikhilam Sutra is divided into nine sections, the first of which begins at number nine and ends at number ten. It finds the complement of the huge integer from its nearest base in order to conduct the multiplication operation. Ekanyunena Purvena denotes anything that isn't quite the same as the one before it, or something that isn't quite the same as what came before it. This sutra may be used to multiply integers with a multiplier of 9 or a cluster of 9 multipliers. In digital signal processing (DSP) situations where multiplication involves repeated addition operations, the squaring job comes in handy. To get the square of a number, the Ekadhikena Purvena Sutra is summarized.

Parameters such as latency, area, and power are traded off in many adder designs which have been proposed. Next by replacing multiplier and accumulator (adder) with existing designs, a number of MAC unit design may be created. There are comparisons in terms of latency, area, and power between different MAC unit models. Based on Vedic Mathematics, this paper shows many multiplier architectures that are both quick and low power.

The four Vedas include Vedic mathematics. It is part of the Sthapatya-Veda (Civil Engineering) upaveda (supplement) of the Atharva Veda. Vedic Maths explains arithmetic, geometry, factorization, trigonometry, quadratic equations, and calculus. An extensive investigation in the Atharva Veda produced 16 sutras (formulae) and 16 Upasutras (sub formulae). Vedic mathematics deals with a variety of basic and advanced number problems. The term vedic comes from the word Veda, which meaning all learning's storage place[7]. Vedic mathematics deals with a number of different aspects of mathematics, the most important of which are the 16 Sutras: (Anurupye) Shunyamanyat, Chalana-Kalanabyham, Ekadhikina Purvena, Ekanyunena Purvena, Gunakasamuchyah, Gunitasamuchyah, Nikhilam Navatashcaramam, Paraavartya Yojayet, Puranapuranabyham, Sankalana-vyavakalanabhyam, Shesanyankena Charamena, Shunyam Saamyasamuccaye, Sopaantyadvayamantyam, Urdhva-tiryagbhyam, Vyashtisamanstih, Yaavadunam

These sutras can be used to answer issues in any discipline of mathematics. Vedic Mathematics lends itself to FPGA implementation because of its regularity. By constructing the data route operators with Vedic Mathematics, a greater throughput may be achieved.

A recent tendency has been to create RISC processors that are efficient for certain applications and fulfil the application's basic needs. The key criterion for such CPU design is performance. The fundamental distinction between CISC and RISC architecture is that RISC processors are optimized with a high number of registers and instruction pipelining, which allows for a low number of clock cycles per instruction. The LOAD/STORE architecture is also a key component of RISC.

II. EARLIER WORK

As the name suggests the Multiplier and Accumulator (MAC) unit have multiplier and an accumulator in it. For An N-bit MAC unit, it is having N-bit multiplier and a (2N+-1)-bit accumulator. A three-step multiplier is common. First step of the process is to generate partial products. Next step is to produce a partial product matrix. For unsigned multiplication, AND gates are used. Further step is reduction of partial products. The PPM may be reduced to two rows by utilizing either the Dadda tree or Wallace tree approaches. The final addition comes in the third stage. The summing of the last two rows is done using final adder. A (2N-1)-bit adder is required for last step addition. In a traditional MAC unit, additions in multiplications and additions in accumulations must be performed twice. It is important to keep in mind

that carry propagation takes time. As a result, the PPR process's input is sent back to the multiplier output. Because the accumulation is handled by the final adder, only one carry propagation is required.

In multiplication the major problem is the high time consumption. For overcoming this problem various methods such as high-radix, scalable, and signed-digit multipliers have been proposed. When compared to previously developed design the proposed MAC unit is capable for performing ordinary arithmetic but not modular arithmetic.

The multiplier is one of the most important arithmetic blocks, and it is frequently utilized in a variety of applications, particularly signal processing applications. The multipliers may be built in two different architectures: sequential and parallel. Sequential designs are low-power, however they have a very long delay. Parallel architectures, on the other hand, (such as Wallace tree and Dadda) are quick but use a lot of power. Parallel multipliers are utilized in high-performance applications where their excessive power consumption might result in hotspots on the chip. The adjustment of these parameters for multipliers becomes vital since power consumption and speed are critical elements in the design of digital circuits. Frequently, one parameter is optimized while the other value is constrained. In particular, given the restricted power budget of portable devices, obtaining the necessary performance (speed) is a difficult issue. Furthermore, achieving the system's intended performance may be hampered by a predetermined degree

It's a logic circuit that combines two types of logic. One AND gate and one EX-OR gate may be connected to design it. The A and B input terminals make up a half-adder circuit. Both of these algorithms add two input digits (one-bit values) and provide a carry and a total as output. As a result, the output terminals are split into two.

The total of both one-bit values is obtained from the EX-OR gate's output. The output of the AND gate is nothing but carry. But carry propagation is not achieved due to lack of a logic gate that can process it. The Half Adder circuit gets its name from this.

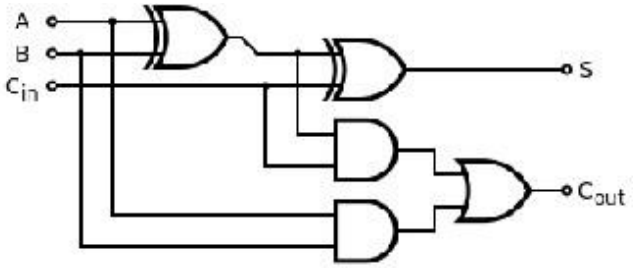
The output equation for both gates may be expressed as a logical operation performed by the logic gates. The sum equation is written as an EX-OR operation, whereas the carry equation is written as an AND operation.

Two AND gates, two EX-OR gates, and one OR gate make up a complete adder. The complete adder multiplies three binary digits together. The carry, which we get from the previous addition as C-IN, is one of the three, and the other two inputs are A and B. C-OUT is the input carry and is SUM is represented as S.

A full adder is an extension of half adder. However, there are more logic gates on it. As a result, it multiplies the preceding carry to get the final result. The Full Adder gets its name from this.

Using one OR gate and two Half Adders, a Full Adder may likewise be created. The OR gate creates a carry following the addition. The second Half Adder produces the total of these numbers as output.

The sum of all binary digits is the equation for the output that the EX-OR gate can produce. The carry obtained by addition is the output from the AND gate in this case. This is a logical operation



.

Fig 1. Full adder

Multipliers are utilized in a variety of applications including digital signal processing. Many academics have focused on design considerations for greater performance as a result of developments in current technology. High speed, precision, low power consumption, layout consistency, and a small footprint are only a few of the design goals. Multiplexers, adders, and MAC are only a few of the computational components of a DSP processor. In

comparison to prior versions, the operation and execution performance of these blocks has improved. The speed at which multipliers operate is determined by two factors: technology of semiconductor used and design of multiplier. The multiplier's speed depend on the adder's operating speed since adders are the basic component in it. Multipliers come in a variety of shapes and sizes, with the 44 array multiplier being one of the most sophisticated. There are three key phases in the multiplication procedure: Partial product generation, Partial product reduction and Final addition.

The most common multiplication process is "add and shift "algorithm which is shown below:

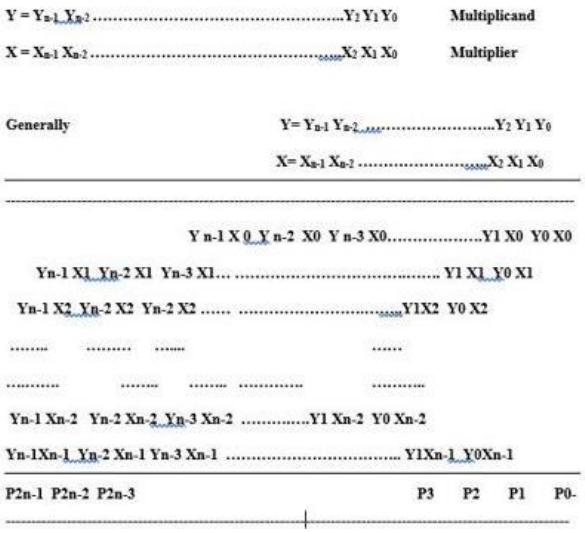


Fig 2. Array multiplier

Where, Multiplicand = N-bits, Multiplier = M-bits and Partial products = N*M.

The add shift algorithm underpins the multiplier circuit. The array multiplier's key benefit is its straightforward design and consistent form. An array multiplier's downside is that it has a long delay and uses a lot of energy. In order to execute continuous and complicated operations in digital signal processing, the MAC unit performs multiplication and accumulation procedures several times. To govern its operation, the MAC unit also has a clock and a reset. The Array Multiplier (AM), which is made up of partial products formed using AND Logic, is one of the finest traditional multipliers. The Half Adder (HA) and Full Adder (FA), depending on the amount of input bits, are used to add all partial products. The multiplication mechanism is comparable to that of an array multiplier in a ripple carry array multiplier with row bypassing approach. However, depending on the carry value obtained in the adder stage, certain product steps are bypassed from one state to the next.

As array multiplier is degrading the performance of the RISC processor we are going for the utilization of vedic multipliers. Vedic multiplication is the fastest way of implementation and these are derived at very ancient times which were given in vedas.

The hardware architectures of 2×2, 4×4 bit Vedic multiplier unit based on Urdhva-Tiryakbhyam. In Vedic multiplier, a concurrent execution approach is applied on the fractional product creation and additions. Hence, it is well adapted to parallel processing. This reduces delay and this is primary motivation behind this work.

Vedic Multiplier for 2×2 bit Module

The technique is illustrated in Figure 1 for two two-bit values A and B, where A = a1a0 and B = b1b0. Initially, the method considers the multiplication of LSBs to arrive at the final product's LSB (vertical). The LSB of the multiplicand is then multiplied by the multiplier's next higher digit, followed by the addition of the result and the product of the multiplier's LSB and the multiplicand's next higher bit (crosswise). The summation provides the second bit of the final result, and the carry is added to the part product formed by multiplying the most significant bits, which comes before the sum and carry terms.

$$S_0 = a_0 b_0 \quad (1)$$

$$c_1 s_1 = a_1 b_0 + a_0 b_1 \quad (2)$$

$$c_2 s_2 = c_1 + a_1 b_1 \quad (3)$$

The 2×2 Vedic multiplier module is implemented using four input AND gates along with two half-adders.

B. Vedic Multiplier for 4×4 bit Module

Incorporating four such modules of 22 multipliers completes the 444 bit Vedic multiplication unit. For

the 44 multiplier, the processing is shown in the form of a block diagram in Fig.

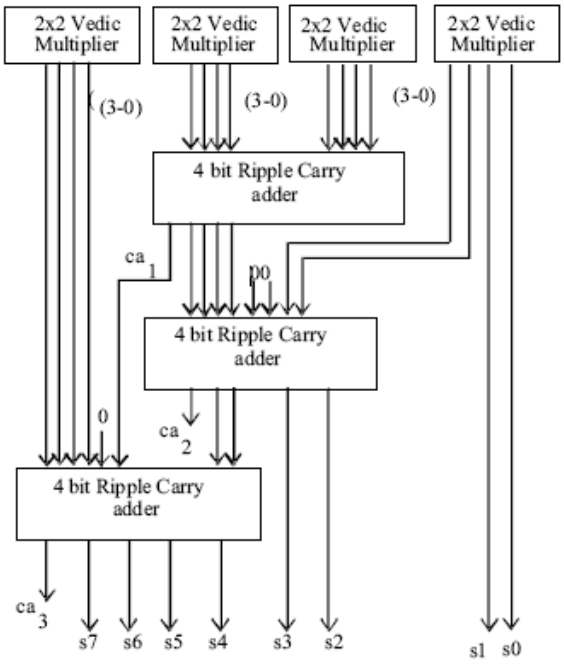


Fig. Schematic diagram of 4x4 bit Vedic mathematic based multiplier

C. Vedic Multiplier for 8×8 bit Module

Four 4x4-multiplier modules are used to create the 8x8 Vedic multiplier modules. In the diagram below, the Vedic technique is used to process the 8x multiplier. With two 8-bit numbers, a=a7a6a5a4a3a2a1a0 and b=b7b6b5b4b3b2b1b0, the result is S15-S0. Additionally, the integers a and b are separated into two equal four-bit pieces, aH-aL and bH-bL. P = a b = (aH-aL) (bH-bL) = aH bH + (aH bL + aL bH) + aL bL is now known as the 16 bit final product term. Following then, the operation of multiplication is carried out by feeding four bits concurrently to the four-bit multiplier unit, and then adding the output to get two eight-bit values as the final result.

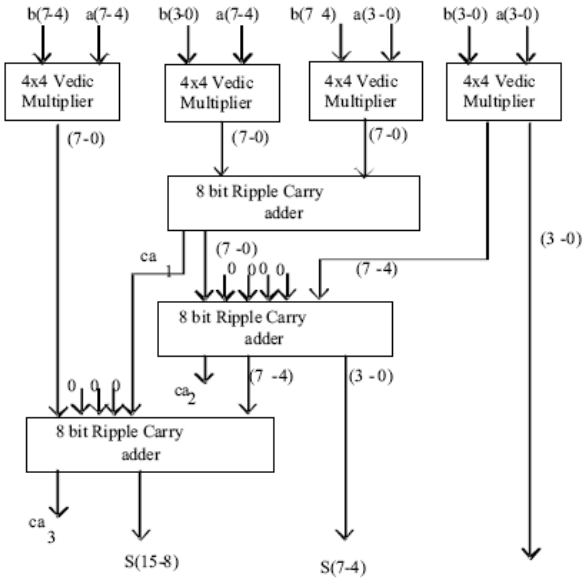


Fig .Schematic diagram of 8x8 bit Vedic mathematic based multiplier.

III. PROPOSED WORK

The processor's job is to efficiently execute all of the instructions in the machine language. The ALU stands for Arithmetic and Logical Unit in a combinational circuit. The purpose of this unit is to do a variety of calculations utilizing a variety of instruction sets. ALU inputs in a processor are made up of an instruction (machine word) and certain operands. The opcode instructs the ALU on which and what operation to do, and the operands are subsequently employed in the operation.

Register bank is a modest collection of data storage facilities. The ALU saves the outcome of the operation in an accumulator, which is then stored in a storage register, and it verifies the bits to see if the operation was successful. If the command is not properly executed, a status register, also known as a Z-Flag, will be shown. Its job is to run programs and keep data in memory running smoothly. A processor contains a set of instructions, which are nothing more than instructions for a computer to accomplish a task. Decoding the op-code, detecting the instruction, determining which operands are in memory, fetching the operands from memory, and finally issuing a

command to a processor to execute the instruction are all tasks that IR does. This is accomplished through the use of a control unit that creates timing signals that control the different processing parts involved in the instruction's execution.

The address in the program counter is applied to memory, and the current instruction is put in the Memory location following the increment in the PC to the next address. The MAR is full of Binary words that point to the word's position in RAM. The instruction is kept at this area.

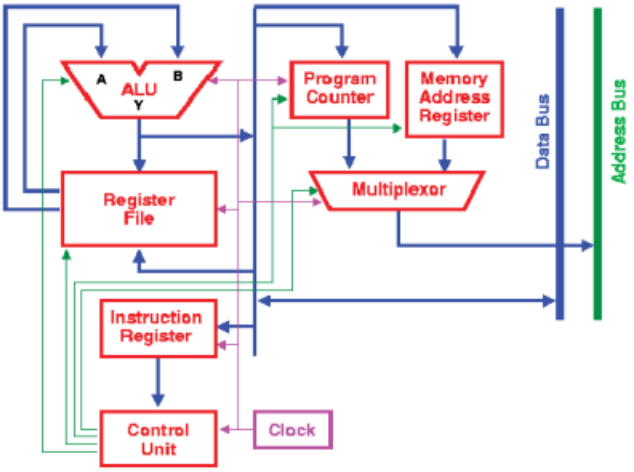


Fig: block diagram of processor

The sutra Urdhva Tiryakbhyam (Vertical and Crosswise) (Algorithm). It's a universal multiplication formula that works in every situation. All partial product and sum are generated in one step using the procedure. For nxn bit numbers, the procedure has been generalized. On comparing proposed multiplier with other multipliers, it is having an advantage of increased gate latency and area. The line's digits are multiplied, and the result is added to the preceding carry. When there are many lines in a single step, the results are summed together. The least significant digit of the number so produced serves as one of the result digits, while the remainder serves as the carry for the following step. The carry is set at zero in the beginning.

MODULE OF N×N VEDIC MULTIPLIER

An N×N bit Vedic multiplier module is designed similar to previously mentioned different Vedic multipliers. In the figure below there is a block diagram of N×N bit Vedic multiplier

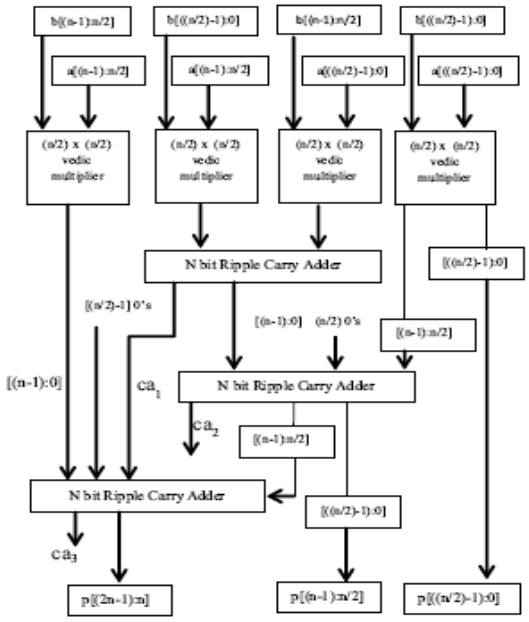


Fig: Schematic block diagram of N×N bit Vedic multiplier

In comparison to a conventional multiplier, the UT Technique based Multiplier is a more efficient hardware design for multiplying two integers; it is mostly used for high-speed multiplication. Figures 4 and 5 showthe d the block diagrams of 44 and 88 bit UT multipliers. Garbage Bits are the last two output bits, P8, P9 (for 4x Multiplier) and P16, P17 (for 8x Multiplier).

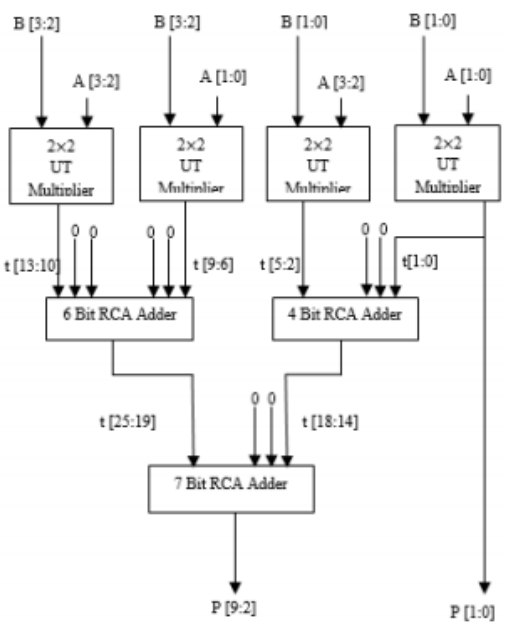


Fig: Block Diagram of Proposed 4×4 UT Multiplier

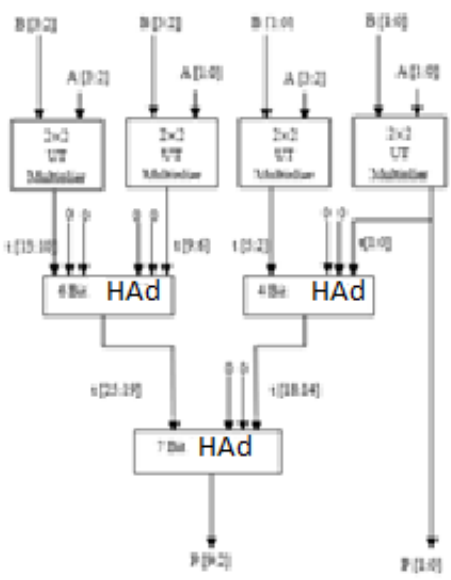


Fig: proposed Hybrid Adder based UT vedic multiplier

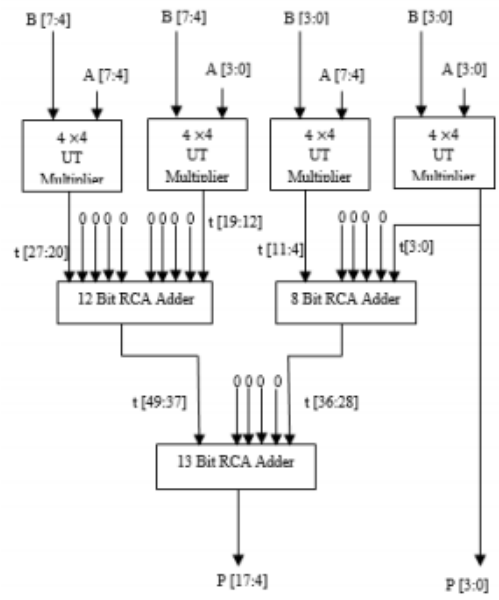


Fig: Block Diagram of Proposed 8×8 UT Multiplier

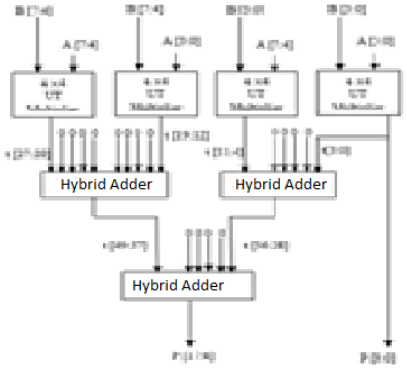


Fig: proposed hybrid adder based vedic multiplier

The next instruction to be performed is indicated by the program counter. After the processor gets the instruction from the memory address designated by the program counter, the instruction is loaded into the instruction register in the entire instruction cycle. Because this circuit creates the timing and control signals for the activities done by the CPU, the Control Unit is a crucial aspect of any type of computer or system. The ALU is in charge of signal transmission between the CPU, memory, and different buses, therefore communication is between it and the main memory.

The input selector of the multiplexer block is used. It has the ability to control certain lines through wires. It's a circuit that accepts many inputs and outputs a single signal.

HYBRID ADDER

The term hybrid adder refers to an adder that is built by incorporating one or more logic. Figure 6 shows a hybrid adder's block diagram. In modules 1, 2, and 3, we may use the same or various types of adders to give sum and carry as output. Using different adders or the same adder can produce multiple logic values.

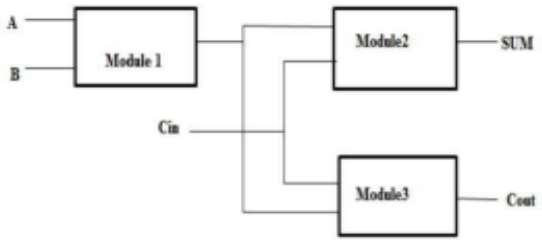


Fig: generalized block diagram of hybrid adder

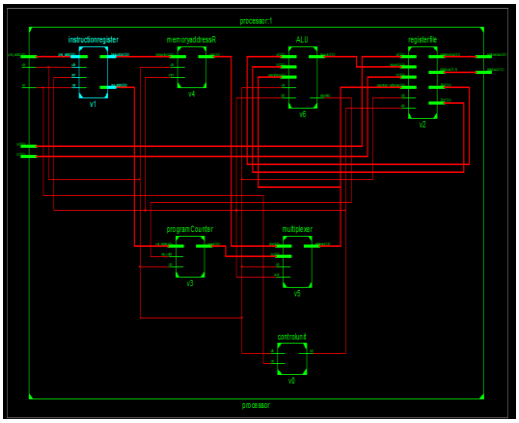
In order to build a hybrid adder, there are two forms of architecture. Homogeneous is a word that may be used to describe anything that is similar Homogeneous architecture is created by combining two or more adders of the same kind. Heterogeneous is a word that describes a group of people that have different Heterogeneous Architecture is formed by the fusion of two or more types of adders. The concept of combining designs to create a hybrid structure results in products that are both high-performing and low-cost. The engineers that created

the designs Can create a hybrid adder by considering the limitations and benefits of separate adders.

Homogenous hybrid adders are higher-bit adders made from a smaller number of n-bit adders, allowing for greater space savings.

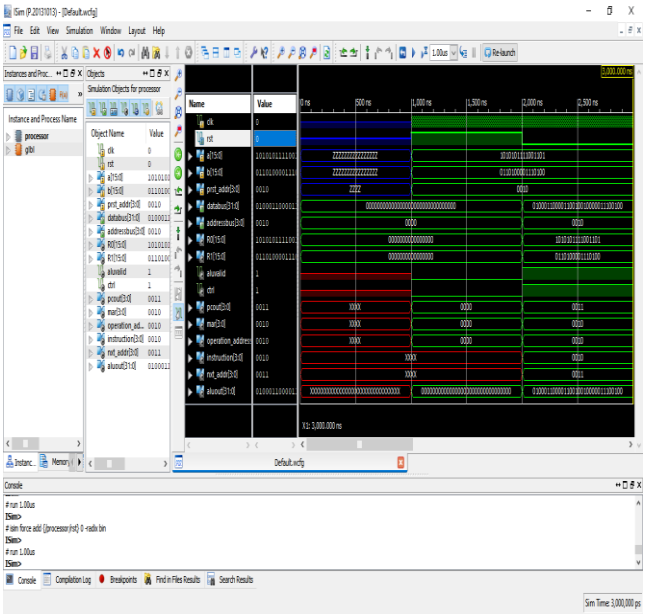
IV. EXPERIMENTAL RESULTS

RTL schematic:



Technology Schematic:

Simulation results:



Area:

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

Selected Device : 7a100tcsg324-3


Slice Logic Utilization:
 Number of Slice Registers:             204  out of  126800     0%
 Number of Slice LUTs:                 1289  out of  63400      2%
    Number used as Logic:              1289  out of  63400      2%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   1416
   Number with an unused Flip Flop:    1212  out of   1416     85%
   Number with an unused LUT:           127  out of   1416     8%
   Number of fully used LUT-FF pairs:    77  out of   1416     5%
   Number of unique control sets:         7

IO Utilization:
 Number of IOs:                          74
 Number of bonded IOBs:                  74  out of    210    35%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     32     3%
```

Delay:

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 25.673ns (frequency: 38.952MHz)
  Total number of paths / destination ports: 3543793745926270200 / 280
-------------------------------------------------------------------------
Delay:               25.673ns (Levels of Logic = 112)
  Source:            v2/R1_0 (FF)
  Destination:       v6/aluout_0 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: v2/R1_0 to v6/aluout_0
```

Evaluation table:

	Area (LUT's)	Delay (ns)
proposed	1289	25.673
extension	1267	25.672

V. CONCLUSION

In this paper, RISC processor based on vedic sutras in ALU has been implemented. Vedic Multipliers are used for multiplication to improve the speed and

reduce the area and power budget of the Multipliers and in addition to it adders are getting replaced with hybrid adders which will improve the area efficiency slightly and hardware complexity will be optimized compared conventional adders. In this work, Vedic processor is designed using Vedic MAC and Vedic ALU along with other conventional blocks in processor. The instruction set architecture of 14 instructions using register addressing modes is implemented using instruction register and 1-bit Z flag register which checks the status of the arithmetic group instructions. The comparison of simulation result of Vedic ALU and MAC design is done with the existing ALU and MAC results by using hybrid adders not only for multipliers addition operation in ALU can also be improvised. The 16-bit Vedic processor reduces the delay and saves power compared to conventional processor. Hence the improvement in speed of operation, reduction in power utilization and less area utilization are the key features of designed RISC processor.

VI. REFERENCES

[1]. S. Lad and V. S. Bendre, Design and Comparison of Multiplier using Vedic Sutras, 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-5

[2]. Balpande Vishwas V, Abhishek B. Pande, Meeta J. Walke, Bhavna D. Choudhari and Kiran R. Bagade. Design and Implementation of 16 Bit Processor on FPGA. (2015).

[3]. Seung Pyo Jung, Jingzhe Xu, Donghoon Lee, Ju Sung Park, Kang-joo Kim and Koon-shik Cho, Design & verification of 16 bit RISC processor, 2008 International SoC Design Conference, Busan, 2008, pp. III-13-III-14, doi: 10.1109/SOCDC.2008.4815726.

[4]. Mr. Nishant G. Deshpande, Prof. Rashmi Mahajan, Ancient Indian Vedic Mathematics based Multiplier Design for High Speed and Low Power Processor, IJAREEIE, Pune, 2014

[5]. P. S. Mane, I. Gupta and M. K. Vasantha, Implementation of RISC Processor on FPGA, 2006 IEEE International Conference on Industrial Technology, Mumbai, 2006, pp. 2096-2100, doi: 10.1109/ICIT.2006.372448.

[6]. Ram, G. & Lakshmanna, Y. & Rani, D. & Kandula, Bala. (2016). Area efficient modified vedic multiplier. 1-5. 10.1109/ICCPCT.2016.7530294.

[7]. Maroju SaiKumar, P.Samundiswary, Design and Performance Analysis of Various Multipliers using Verilog HDL, CiiT International Journal of Programmable Device Circuits and Systems, vol.5, no.9, pp.391-398, Sep 2013.

[8]. Jikku Jeemon, Low power pipelined 8-bit RISC processor design and implementation on FPGA, ICCICCT 2015.

[9]. Supraj Gaonkar and Anitha M, Design of 16-bit RISC Processor, IJERT Vol. 2 Issue 7, July 2013.

Cite this article as :

Sharada, "Area Efficient Vedic Multiplier Based on Homogenous Hybrid Adder for RISC V Applications", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 9 Issue 4, pp. 303-311, July-August 2022.
Journal URL : https://ijsrst.com/IJSRST229448