MFA-Based RISC-V Evaluation and Strategy Exploration of MRAM

M. Prabhakar
Assistant Professor,
Department of Electronics and Communication Engineering
KSRM College of Engineering
Kadapa, Andhra Pradesh
prabhakar@ksrmce.ac.in

G. Sowjanya
Student, Department of ECE
KSRM College Of Engineering
Kadapa, Andhra Pradesh
Sowjanyaganjikunta12@gmail.com

P. Chandra Hasan Reddy
Student, Department of ECE
KSRM College Of Engineering
Kadapa, Andhra Pradesh
219y1a04d5@ksrmce.ac.in

G. Jagan
Student, Department of ECE
KSRM College Of Engineering
Kadapa, Andhra Pradesh
229y5a0406 @ksrmce.ac.in

T. Anil Kumar
Student, Department of ECE
KSRM College Of Engineering
Kadapa, Andhra Pradesh
229y5a0420@ksrmce.ac.in

T. Mahendra
Student, Department of ECE
KSRM College Of Engineering
Kadapa, Andhra Pradesh
219y1a04g0@ksrmce.ac.in

S. Venkata Nikhil varma
Student, Department of ECE
KSRM College Of Engineering
Kadapa, Andhra Pradesh
219y1a04e1@ksrmce.ac.in

Abstract: This project presents a 16-bit RISC processor designed with a Multiplier-Free Algorithm (MFA) to replace the conventional Full Adder (FA), thereby reducing power consumption, delay, and area utilization. The Reduced Instruction Set Computer (RISC) architecture offers enhanced performance, higher operational speed, and a simplified instruction set. The key achievement in this work is the implementation of MFA-based multipliers in the Arithmetic and Logic Unit (ALU) and Multiplier and Accumulator (MAC), replacing traditional full-adder-based designs.MFA optimizes computational efficiency by eliminating complex multiplication operations, significantly improving processing speed and reducing overall power consumption. Additionally, the processor incorporates Magnetoresistive Random Access Memory (MRAM) for enhanced energy efficiency and non-volatile storage. The designed RISC processor consists of functional blocks such as a Control Unit, Data Path, Register Bank, Program Counter, and Memory Unit.The proposed MFA-based RISC processor is capable of executing 14 instructions efficiently. It exhibits power savings and reduced delay in In comparison with conventional MAC and ALU architectures, the incorporation of the Multi-Function ALU (MFA) within the Vedic ALU and MAC units significantly improves computational speed while effectively reducing power consumption and area utilization. This approach achieves approximately a 2.5% decrease in LUT (Look-Up Table) usage. The complete 16-bit RISC processor architecture, based on MFA, is implemented using Verilog HDL and synthesized using Xilinx Vivado 2018.3. Additionally, MRAM (Magneto resistive RAM) is designed using Tanner EDA tools and seamlessly integrated into the Vivado environment for evaluation as part of the RISC processor system.

Keywords:- Reduced Instruction Set Computer; VonNeumann architecture; Verilog HDl, Vedic Mathematics, Urdhva-Tiryagbhyam Sutra

I . INTRODUCTION

Vedic mathematics is easily examined by research scientist Bharati Krishna, with a total of 16 scriptures and 13 aids explain Vedic mathematics, increasing the simplicity of the solution to mathematical equations for calculations.

Swami Bharati Krishna Tirthaji Maharaj, Shankaracharya by Goverdhan Peath illuminated the prehistoric system of Vedic mathematics. It's simple and not just regular, but also logical. This normality in Vedic mathematics makes it so popular that it is very committed in India and the rest of the world. Digital signal processing requires activities such as frequency domain filtering (FIR, IIR) and frequency conversions such as DFT, FFT, DCT.

In these tasks, multiplication is the basic hardware part. Thus, multiplier presentations are an important component in deciding to publish the entire framework. This is why multipliers are the slowest and most boring components in the framework. In this sense, improving multiplier speed and domain is an amazing test of frame architects. This test can be effectively overwhelmed by using old Vedic mathematical techniques.

Binary multipliers are used in digital circuit planning. This means it's quick, reliable and effective to perform all your tasks. Depending on the segment procedure, you can access different types of multipliers. Vedic Multiplier is based on Vedic Mathematics, and contains 16 different Sutras, divided into various arithmetic operations. The strength of Vedic

mathematics is not only useful with his efforts and consistency. The high importance is attributed to the certainty mentioned above. The logic behind the urdhva tiryakbhyam sutra is particularly similar to traditional array multipliers.

All Nikhilam Sutras are 9 and 10. A number of compliments are determined for the following base to perform the multiplication process: Ekanyunena Purvena suggests people who don't immediately turn what you're doing into one or one. This SUTRA is the iterative addition process of digital signal processing applications (DSPs) in which multiplication activities are repeated in digital signal processing applications. The Ekadhikena Purvena Sutra is summarized to find the squares of numbers.

Typically, there are three steps in a multiplier. The first step is the PPG process (partial product generation). For example, you can use the target to generate a partial product matrix (PPM) for unsigned multiplication. The second step is the PPR process (partial product reduction). By using the Dadda Tree or Wallace Tree approach, you can reduce the PPM to two rows. The third step is the final addition. Use Add -On (final Add-ON) to perform the sum of the last two lines. For N-bit multipliers, a (2N-1) bit adder is required for the final addition.

Various Addier architectures have been proposed for compromises between delay, area and power. Additionally, different Mac unit models can be developed by replacing both multipliers and accumulators (Addierers) with different architectures. Comparisons of delay, area and performance between different Mac unit models have been reported.

Despite advances in semiconductor technology and the development of energy-efficient design technologies, the total energy consumption of computer systems is growing rapidly at an astonishing speed to process more information. Particularly when computer systems are ubiquitous, they are increasingly used to interact with the physical world and process large amounts of data from a variety of sources. Furthermore, we expect you to be the natural human interface known to you. The result is a large number of applications known as RMS applications for recognition, mining and synthesis (RMS), taking into account the important parts of the overall computational resources of calculations, from mobile and mobile Internet (IoT) to large, large data centers. Dramatically improving the energy efficiency of these emerging workloads is important to meet the growth of information they need to process.

The most important task of mathematical manipulation is multiplication. It can be done on many types such as MAC, CIAF (Intensive Arithmetic Functions of Computation), and is currently applied to DSP applications. Multiplication is the most important task, so the speed of the system is determined relative to the multiplier used for operation. This work shows a variety of high-speed, low multiplier architectures based on Vedic mathematics. Different types of multipliers are available depending on the component placement.

A particular multiplier architecture is chosen based on the application. ch. Harish Kumar explains a variety of multiplication techniques. Comparative checks of types of 16-bit Vedic multipliers such as Urdhwa Tiryakbhyam, Nikhilam, and Ekadhikena Purvena are also performed with traditional 16-bit multipliers. By investigating various parameters, the authors found that Nikhilam Sutra had lower latency compared to Urdhwa Tiryakbhyam. However, the Ekadhikena Purvena Sutra is the fastest of all multipliers. Multipliers with traditional methods have the largest delay compared to all VEDIC multipliers.

The authors present time analysis and utilization of the devices of the 16, 32, and 64-bit urdhva tiryakbyham and Nikiram nabatashkaram dashata sutra, indicating that the Nikiram nabatashkaram dashata sutra is superior in speed and has less space than the urdhva tiryakbyham sutra. Furthermore, it was observed that the Urdhva Tiryakbhyam Sutra works faster with fewer bits, while the Nikhilam Sutra works faster with larger inputs. The authors investigated and analyzed performance and delay properties and found that the Urdhava Tiryakbhyam multiplier is the optimal multiplier compared to the array and Nikhilam multiplier compared to the delay and performance calculations.

II . RELATED WORKS

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

In this computer world, quick processing units are a prerequisite for many real applications. These units include ALU and MAC as essential basic blocks for efficient and rapid execution. Multipliers are commonly used as the main component of digital signal processors. To maintain accuracy and speed up the execution multiplier Addierer, ALU & MAC changes the registers to improve performance. Fastest multiplier designs are highlighted because of increased limits on slowing down implementations in processors. Designing faster multipliers is extremely important to improve the speed of multiplication. For some multipliers, Vedic multipliers are preferred to accommodate operating speed, area usage, and low power consumption compared to other existing multipliers. This means that algorithms based on Vedic mathematics must design intense, fast, and low capacity supervision. Vedic Mathematics has 16 Sutras, of which urdhva Tiryakbhyam, Ekanyunena Purvena and Ekadhikena Purvena are discussed along with the simulation results. The focus of this paper is to achieve optimal results for each scripture area, speed and performance parameters.

This paper shows a very efficient Vedic multiplier unit with a variety of Sutras Vedic mathematics. The design, synthesis and simulation of 16-bit Vedic multiplier units is performed on Vivado 17.1 and Verilog. Therefore, performance parameters for multipliers such as delay, electricity, and

surface are analyzed for each scripture. It has been observed that the speed increases by 90.41% compared to Urdhwa Tiryakbhyam at 70.26% of the region's decline using Ekadhikena Purvena Sutra. Nikhilam Sutra offers the most effective optimal area and optimal latency in terms of performance. Important performance values are given for each SUTRA. It can be used to select image and signal processing applications and digital filters.

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).

This project involves designing a 16-bit RISC processor and modeling components using Verilog HDL. The processor is based on the Harvard architecture. The instruction set applied here is very simple and provides insight into the types of hardware that should be able to properly execute the instructions. More complex blocks such as aluminum and memory have been designed and simulated, along with building blocks in combination with sequential processors such as adders and registers. The aluminum modelling carried out in this project begins entirely structurally from half. Finally, a half-customer layout of aluminum was developed. Complex blocks such as memory were modeled using behavioral approaches, whereas simple blocks such as adders were performed using structural approaches. In the future, processors can be extended to 32-bit or 64-bit processors through simple changes to code, and data APATH can be changed to include various new blocks that are not possible in traditional processor units. A control unit (CU) is a component of a processor. There is memory for computers, arithmetic and logic units, and output devices, to respond to instructions sent to the processor. It directs the operation of other units by providing time and control signals. Most computer resources are managed by the CU. Hardwired control units are implemented using a combinational logic unit that contains a limited number of goals that can produce a specific result based on the instructions used to access these answers. The retained control unit is generally faster than a microprogrammed design.

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

This article introduces the process of designing and reviewing a 16-bit RISC processor. The proposed processor has a Harvard architecture, consisting of 24-bit addresses, five-stage pipeline statements, and internal debug logic. The ADPCM bocoll and the Sola algorithm run successfully on the processor generated by the FPGA. Portable Multimedia Players (PMPs) and Personal Digital Assistants (PDAs) are not special elements for people. Therefore, processors with low power processes and small sizes are manufactured as SOC-level ASICs (application-specific integrated circuits). Popular processors for SOC-level ASICs are the 8051 processor and the ARM 7 processor. The ARM 7 is used in SOC-level ASICs with gate sizes of millions. The 8051 processor is used in simple systems that require small sizes. However, the bit size of a simple system changes to 8 at 16. Therefore, the 8051 processor's computation time increases. To replace the 8051 processor, this article proposes a new 16-bit-RISC processor. The proposed processor is designed to be embedded in the SOC of an ASIC. Core execution tests are performed. However, on-chip Debrogen debugging is required for application obligations. Figure 5 shows the connections for the GDB server program, SW-Debugger, and the new core. It is also required for the Quick Programming Compiler. The next goal of the proposed processor is to develop compilers and GDB server programs for high users.

III. EXISTING METHOD:

Existing methods implement MAC units with binary multipliers, often providing additional to large power consumption and large path delays (including final additions to multiplication and additions to the battery). The MAC unit consists of two separate blocks: a multiplier and an accumulator (i.e., accumulation adder). The N-bit MAC unit contains an N-bit multiplier and one (2N+±-1) bit accumulator (Addierer). Many previous work paid attention to multiplier optimization or adder optimization. Typically, there are three steps in a multiplier. The first step is the PPG process (partial product generation). For example, you can use the target to generate a partial product matrix (PPM) for unsigned multiplication. The second step is the PPR process (partial product reduction). By using the Dadda Tree or Wallace Tree approach, you can reduce the PPM to two rows. The third step is the final addition. Use Add -On (final Add-ON) to perform the sum of the last two lines. For N-bit multipliers, a (2N-1) bit adder is required for the final addition.

Various Addier architectures have been proposed for compromises between delay, area and power. Additionally, different Mac unit models can be developed by replacing both multipliers and accumulators (Addierers) with different architectures. Comparisons of delay, area and performance between different Mac unit models have been reported. A traditional MAC unit requires two support times to be performed. Adding multiplication and adding to accumulation. Note that diffusion is time consuming. Therefore, the multiplier output is due to the input of the PPR process. As accumulation is handled by the final ADD-ON, there is only one thing to be needed.

Various Addier architectures have been proposed for compromises between delay, area and power. Additionally, different Mac unit models can be developed by replacing both multipliers and accumulators (Addierers) with different architectures. Comparisons of delay, area and performance between different Mac unit models have been reported. A traditional MAC unit requires two support times to be performed. Adding multiplication and adding to accumulation. Note that diffusion is time consuming. Therefore, the

multiplier output is due to the input of the PPR process. As accumulation is handled by the final ADD-ON, there is only one thing to be needed.

It is worth noting that spending time is also a challenging problem with modular proliferation. This suggests some highradix, scalable, and signed digit multipliers for modular multiplication. In contrast to these previous studies, the proposed MAC unit for standard arithmetic (instead of modular arithmetic) has been designed.

Multiplier accumulator units (MACs) support a large number of DSP applications (digital signal processing). It also provides the microcontroller with the ability to handle signal processing for a variety of applications, such as servo/audio controls. MAC has implemented a three-stage plumbing arithmetic architecture in which MAC, the processor's execution unit, optimizes the 16 RAW 16 Duck multiplier. This design supports both 16-bit and 32-bit operators. It also supports signed/unsigned integers and signed fixed point fraction input operators. The MAC unit supports three main functions:

Signed integers and unsigned integer multiplication. Multiplies operational operations for supported, unsigned, and signed fracture operators.

Under different arithmetic blocks, multipliers are one of the main blocks that are often used in a variety of applications, especially in signal processing applications. Multipliers have two common architectures, continuous parallel. The sequential architecture is low, but the latency is very high. Parallel architectures, such as Wallace Tree and Dadda, are quick, but consume more power. Parallel multipliers are used in high performance applications where large power commitments can generate hotspot locations on stamps. Optimizing these parameters of multipliers is very important, as power consumption and speed are important parameters in digital circuit design. Very often, parameter optimizations are performed taking into account the limitations of other parameters. Achieving the desired performance (speed) is a challenge, especially when a limited power budget takes into account portable systems. Furthermore, certain levels of reliability can be another obstacle to achieving systematic performance.

Multiplication is the fundamental operation of signal processing. Basic multiplication methods have been added, and algorithm switching is available. The multiplier has a wide area, has a long latency, and consumes considerable strength. Therefore, designing multipliers at high speed is an important task, and low power consumption is an important task. Multiplier designs with low power supply play a key role in designing VLSI systems with low power consumption. System performance is always dependent on multiplier circuits.

Therefore, optimizing the speed and area of multipliers is one of the main design problems. However, space and speed are usually inconsistent con artists, allowing for faster speeds in large areas. According to another number (multiples), different times (multiples) are even more diverse times, according to another number (multiples) to achieve the result (product). Multipliers play a key role and a variety of applications in today's digital signaling processes. Therefore, designing multipliers at high speed is an important task, and low power consumption is an important task. Multiplier styles should provide high speed and low power consumption. Multiplication involves three main steps to partially generate product generation of partial product reductions in the final addition.

The multiplier design, or H. binary multiplier, is then found an estimate of the two N-bit countries, before implementing it on the Spartan 6 FPGA board in Nexys 3. After implementation, the 32-bit multiplier is compared to the traditional multiplier based on a summary report.

The half and the perfect adder are both combined - logical circuits, with both different aspects of the input processing. Each combination circuit does not have a memory element. These include only logic gates. There are major differences between half-addie and full ads. Half Addierer adds the current input only as a bit number and does not focus on the previous input. On the other hand, the full addition can easily carry current inputs and outputs from previous additions.

It is a combination logic circuit. It can be designed by connecting the gate to the original gate. The semi-target circuit consists of A and B, with two input terminal names. Both add two input points (1 bit number) and generate the output in the form of carriers and sums. So there are two output connections.

The output obtained from the ex-or gate is the sum of two 1 bits. and the output obtained from the gate is called a stretcher. However, you cannot transfer the carry you receive with another additional supplement. That's because there's no logical gate to handle it. Therefore, we have discussed the Halvader circuit.

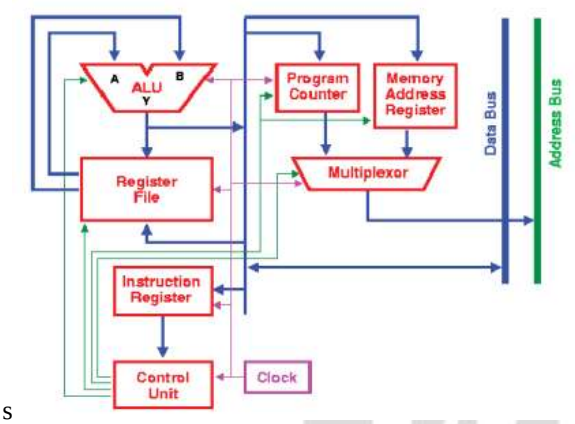
IV . PROPOSED METHOD:

The function of the processor is to efficiently execute each command rate according to the machine language. ALU is a combination circuit, meaning arithmetic units and logical units. This device is designed so that different numbers are created at different instruction rates. In a processor, the aluminum input consists of a procedure (machine language), an operation code (OPCODE), and several operands. The OP code tells aluminum which operations to perform, and these operands are used in the operations.

There is a small statement called bank registration that stops data. Aluminum stores the results of the operation in the accumulator. The accumulator will later be placed in the

memory register to check the bits and indicate whether the operation was performed successfully. If not executed properly, a type of status, namely H. itself, is known as a Z-Flag or status register. Its function is to run programs and work efficiently on data stored in memory. The processor has many instructions that are merely commands to perform tasks on a computer. The control unit continues to execute the instructions. The CPU uses registers such as address registers, data registers, and instruction registers. CPU performance is to collect, decode and execute processes in memory according to registers. The IR task involves deciphering the surgical code. This is determined and determined by the indication of operands that are in memory, and calls operands in memory and refers to the processor to perform the indication. This is done with the help of a control unit that generates time signals that control the various processing elements, including the execution of instructions.

MAR is also known as an address buffer. The address of the program switch is applied to memory, so after PC increments, the current command is saved in the location. The MAR is fully loaded with binary words that indicate the position of the words in the RAM. This place keeps instructions inside it.



s

Fig: block diagram of processor

urdhva tiryakbhyam (vertical and crossing) sutra (algorithm). This is a common multiplication formula that applies equally to multiplication. The algorithm generates all the subproducts and sums in one step. The algorithm is generalized to NXN - BITZAHL. This multiplier has the advantage that it increases very slowly compared to other multipliers as the number of bits, gate delays and area increases. The normal structure improves multiplier processing performance by increasing the width of the input data bus. The numbers on either end of the line are multiplied and the results are added along with the previous carry. If there are more rows in one step, all results will be added to the previous carry. The fewer numbers received this way act as one of the resulting numbers, while the rest serves as a stretcher for the next step. First of all, the stretcher is zero. In Vedic Multiplier, a concurrent implementation approach applies to break protected product locations and additions. Therefore, it is suitable for parallel processing. This reduces delays, which is the main motivation for this work.

Vedic Multiplier for 2×2 bit Module

This scheme is informed for 2, 2-bit numbers A and B. This allows a = a1a0 and b = b1b0 to be shown below. This process first takes into account multiplication of the LSB. This leads to the ultimate product (vertical) LSB. The LSB is then multiplied by a subsequent high multiplier drawing of multiplication, followed by the addition of the LSB result and product by the multiplier, and the next high bit multi-licand (crossway) increase. The total provides the second bit of the final product, and the carrier is added to the sub-product. Sub-products are purchased by multiplying the total and the most important bits leading to wear intervention.

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

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

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

Based on the above equation, the final result is C2S2S1S0. Other cases can be calculated in the same way. The 2 U 2 Vedic multiplier module is implemented with four inputs and gates with two half guards. The 4, 8 & n -bit multiplier is also designed with little change.

B. vedic multiplier for 4-4-bit modules

The 4-bit vedic-multiplication unit is further implemented by including four similar modules of 2 u 2 Mardier. The diagram shows the processing in the form of a block diagram.

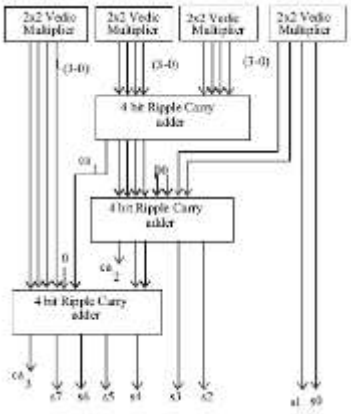


Fig. Schematic diagram of 4x4 bit Vedic mathematic based multiplier

C. Vedic Multiplier for 8×8 bit Module

The eight vedic multiplier modules are realized with four 4-4 multiplier modules. The processing of the multiplier of 8-8 based on the Vedic methodology is shown in the diagram. This is made of 8 bits, called A = a7a6a5a3a2a1a0, respectively, b = b7b6b5b4bb2b1b0, and the output is received at S15 -S0 with 16 bits long. Furthermore, the numbers A and B are divided into two identical parts, 4-bit AH-AL and BH-BL. Now the final production term for 16 bits is understood as follows: p= aãb=(ah-al)â(bh-bl)= ahâbh +(ah bl +alââÂbrh) + al ro. The multiplication process then runs in 4 bits and is also fed into the 4 bit multiplier unit, adding more output to get the final result of two 8 bits.

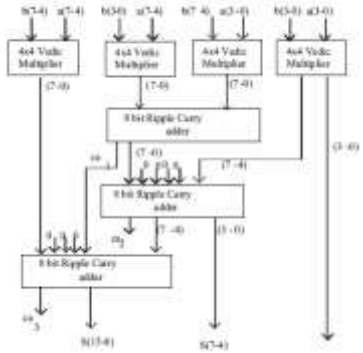


Fig .Schematic diagram of 8x8 bit Vedic mathematic based multiplier.

MODULE OF N×N VEDIC MULTIPLIER

The N×N bit Vedic multiplier module is executed similar to previously discussed various Vedic multipliers. The block diagram of N×N bit Vedic multiplier is shown in Fig. below.

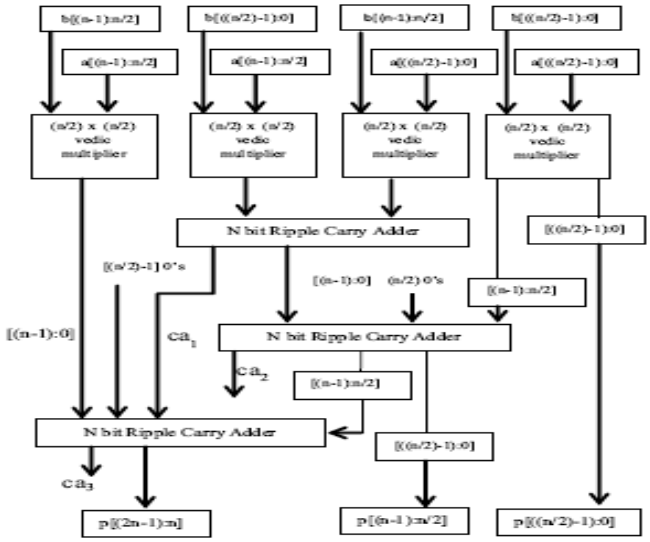


Fig: Schematic block diagram of N×N bit Vedic multiplier

UT Technique based Multiplier is efficient hardware architecture for multiplying two integers, compared to a normal multiplier; it is mostly used for high speed multiplication. The block diagram of 4×4 bit, 8×8 bit UT Multipliers are respectively given below in Figures below 5. Last two output bit P8, P9 (for 4×4 Multiplier) and P16, P17 (for 8×8 Multiplier) are Garbage Bits.

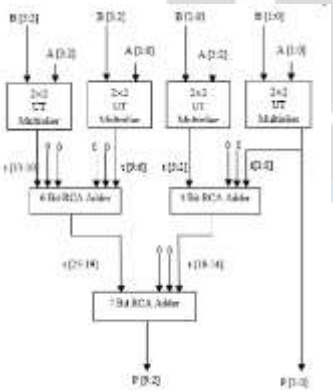


Fig: Block Diagram of Proposed 4×4 UT Multiplier

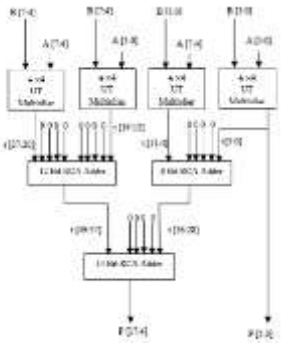


Fig: Block Diagram of Proposed 8×8 UT Multiplier

Program Counter points to the next instruction to be executed. In the complete instruction cycle, the instruction is loaded into the instruction register after the processor fetches it from the memory location which is pointed by the program counter. Control Unit is an essential part of any kind of computer or systems because this circuits generates the timing and control signals for the operations which is performed by the CPU. Here, the communication is between the ALU and the main memory as it controls the transmission of signals between the processor, memory and various buses.

The multiplexer block works as input selector. It can control wires which act as select lines. It is a circuit which takes multiple inputs and gives the single output.

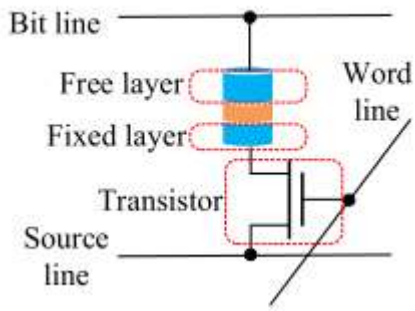


Fig : MRAM

Magnetoresistive Random Access Memory (MRAM) is a non-volatile storage technology that uses magnetic states to store data instead of charge. It is based on a magnet tunnel junction (MTJ) to determine binary states based on resistance levels. MRAM offers high speeds, low power consumption and excellent endurance suitable for embedded systems and high performance computing. Variations such as STT-MRAM and VC-MRAM improve efficiency and scalability. With potential applications in IoT, aerospace and data storage, MRAM is a promising alternative to traditional DRAM and flash memory.

MFA [MODIFIED FULL ADDER]:

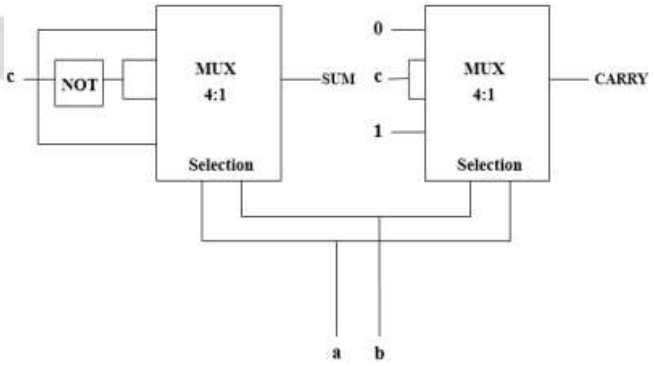


Fig: Modified Full Adder

Combinatorial circuits with up to 2N data inputs, n-selected lines and individual output lines are multiplexers. One of these data inputs is connected to the output based on the value of the selection line. There is an "n-selection line", so there are two combinations of zero and one. Therefore, each combination only selects data entry. Multiplex is also known as Mux. The 4x1 Multiplexer has four data inputs I3, I2, I1, I0, two selection lines S1 and S0, and an output Y.

The block diagram of the 4x1 multiplexer is shown in the following diagram.

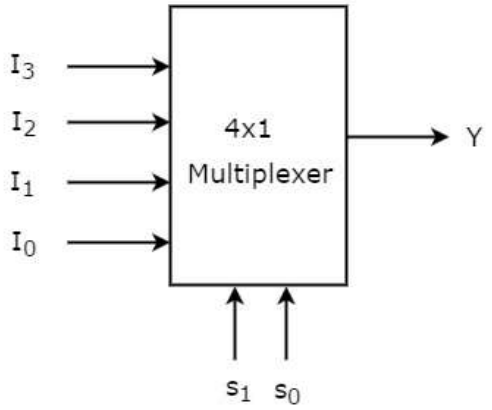


Fig: 4:1 multiplexer

This modified full adoll consists of two 4:1 multiplexers. A, B are the two selection lines of multiplexer C, the only inputs of multiplexer, specifying the output, 0, C, 1, and the inputs of multiplexer as outputs. After examining traditional, fully ADD-based designs, the Multiplexer-based design is modeled by identifying operations based on the truth table and input and output values when logic 0 and logic 1 are input as inputs to a consistent input line. Using this multiplexer-based adder design, it can be observed that efficient implementation is achieved, as the number of logic designs is reduced as the maximum.

RESULTS:

RTL schematic:

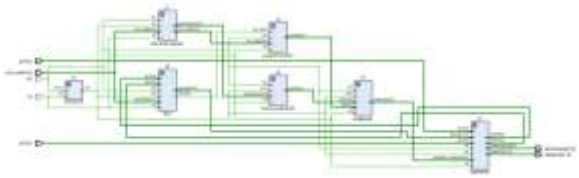


Fig 2: RTL Schematic for Risc Processor

Simulation:

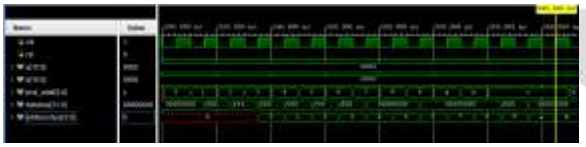


Fig 3: RTL Simulation for Risc Processor

Area:

Name	Slice LUTs (134600)
processor	975

Fig 4: Area for Risc Processor

Delay:

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay
Path 1	∞	85	19	69	v2/R1_reg[0]/C	v6/aluout_reg[0]/D	26.550

Fig 5: Delay for Risc Processor

Power:

Fig 6: Power for Risc Processor

Comparison Table:

	Area(lut's)	Power(W)	Delay (nS)
Proposed	999	17.745	26.555
Extension	975	17.058	26.550

CONCLUSION:

This paper presents the implementation of a 16-bit RISC processor optimized by replacing the Full Adder (FA) with a Mux based Full Adder (MFA) in the Arithmetic and Logic Unit (ALU) and Multiplier and Accumulator (MAC). The MFA-based ALU and MAC significantly reduce chip area, power consumption, and computational delay, making the processor more efficient than conventional designs.The processor architecture integrates a Vedic ALU and Vedic MAC alongside conventional RISC components. The instruction set architecture (ISA) consists of 14 instructions, utilizing register addressing modes and featuring an instruction register with a 1-bit Z flag to track arithmetic operations. To further optimize performance, Magnetoresistive Random-Access Memory (MRAM) is used for result storage, offering non-volatile, high-speed, and energy-efficient memory access.Simulation results comparing the MFA-based ALU and MAC with conventional designs demonstrate superior efficiency in terms of lower delay, reduced power consumption, and minimized area usage. These advancements make the proposed MFA-based 16-bit RISC processor a highly efficient alternative to traditional architectures, well-suited for low-power and high-performance applications.

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] F. Adamec and T. Fryza, "Design — Time configurable processor basic structure," 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, Vienna, 2010, pp. 119-120, doi: 10.1109/DDECS.2010.5491804.

[5] A. Bisoyi, M. Baral and M. K. Senapati, "Comparison of a 32-bit Vedic multiplier with a conventional binary multiplier," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 2014, pp. 1757- 1760, doi: 10.1109/ICACCCT.2014.7019410.

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

[7] Priyanka jain, Dr. G. S. Virdi, : Multiplier-Accumulator (MAC) Unit: International Journal of Digital Application & Contemporary Research ,Volume 5, Issue 3, October 2016

[8] 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.

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

[10] Yogesh M. Motey, Tejaswini G. Panse, “Traditional and Truncation Schemes for Different Multipliers”, International Journal of Electronics and Computer Science Engineering, vol.2, no.2, pp.627-633, May 2013.

[11] 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.