



Design of Lightweight 16-Bit Vedic RISC Processor for Small Scale Embedded Systems

^A Dharmendra Singh Thakur, ^B Priyanka Tripathi, ^C Shailesh Khaparkar

M. Tech Scholar, Gyan Ganga Institute of engineering and Sciences, Jabalpur, MP

Asst. Prof., Gyan Ganga Institute of engineering and Sciences, Jabalpur, MP

Asst. Prof., Gyan Ganga Institute of engineering and Sciences, Jabalpur, MP

ABSTRACT

In this paper, Vedic mathematics based 16-bit RISC processor is designed which can be used in small scale embedded systems or Internet of Things (IoT) nodes. A new instruction set has been designed. Also, to improve processor performance Vedic Arithmetic and Logical Unit (ALU) is used with a new modified tree addition. Very high-speed integrated circuits Hardware Description Language (VHDL) programming is used for Register Transfer Logic (RTL) entry and simulation of the work is done with help of Xilinx ISE tool. Vertex Series FPGA is used as target device for implementation of this work.

Keywords: Vedic Multiplication, Tree addition, ALU, FPGA, RISC

1. INTRODUCTION

In recent years small battery-operated systems applications have increased in their computational complexity. In digital systems the Processor is the key-player to perform all sort of computation hence our concern is to develop high speed RISC processor. Processor is the key element for Microprocessors, Microcontrollers and embedded systems. Figure below explains the component for Processor. In the given figure we explain the implementation of Vedic Processor on FPGA board.

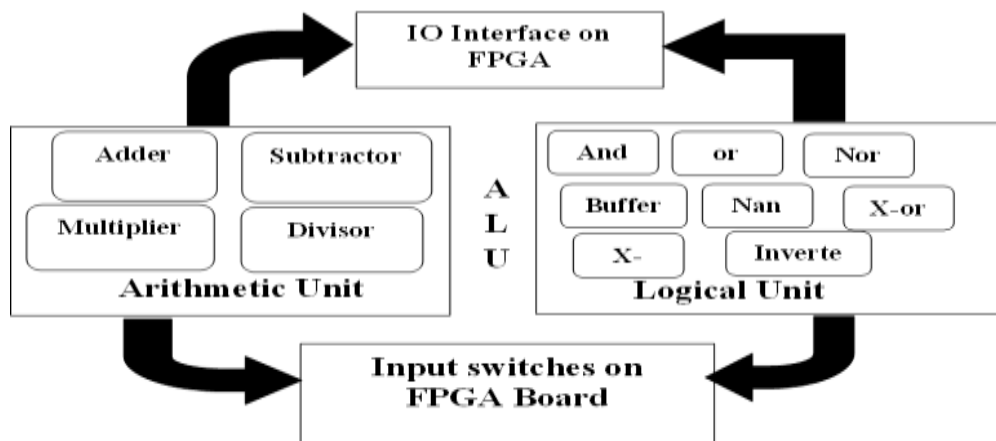


Fig 1 FPGA interface with proposed Processor module

2. DESIGN METHODOLOGY

The RISC processor we introduced is a 16-bit pipelined RISC processor using Harvard architecture, which is a memory construction that program instructions and data stored individually. Fig.2 shows RISC 5-stage pipeline architecture. The fetch unit contains an instruction counter to calculate the address of instruction so as to fetch the instruction from the instruction cache. In decode unit, instruction can be transcribed four parts: 5-bits operation code, 4-bits address of the operand A, 4-bits address of the operand B, and 5-bits destination address to confirm the output of register files.

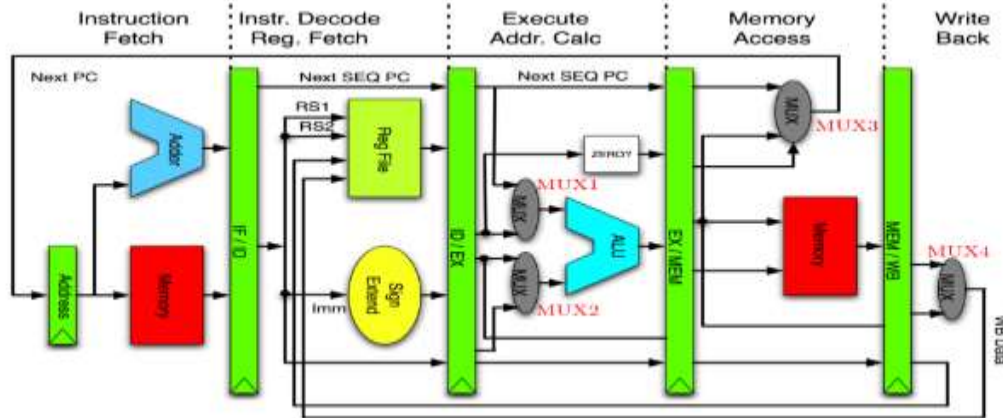


Fig. 2 RISC pipeline architecture

The front-end logic execution unit can fetch the corresponding data from register files depending on the addresses of the operand A and B, then send them to the next unit. Arithmetic execution unit receives the operand A and B. According to the operation code, it's determined that which operation will be implemented. Meanwhile, read-write enable signal confirms which operation will be executed in registers. Arithmetic execution unit is the most important unit in the whole processor and has two parts, ALU and the extended MAC. ALU involves essential arithmetic operation, logic operation and comparing unit. Multiply-accumulation includes the 16x16-bit Vedic algorithm, Wallace tree and CLA [7]. In register access unit, the destination address will be decoded through the code translator, cooperating with read-write enable signal determines which register will be chose. Register files can read the data in the DCACHE, also can store the results from the front unit.

Xilinx ISE is used for implementation proposed RISC processor with, the result shows that the pipeline can correctly implement all functions of proposed RISC processor.

3. ARCHITECTURE

The instruction set width is set to 18-bit including operation code and register address and store address. Instructions' operation codes are varied between 13-17 bits determining the mode of operation. The two operand addresses are determined by 9-12 bits and 5-8 bits of instructions and store address changes between 0-4 bits. The MAC instruction is configured according to the existing instructions. Fig. 3 shows the overall instruction format of the RISC processor.

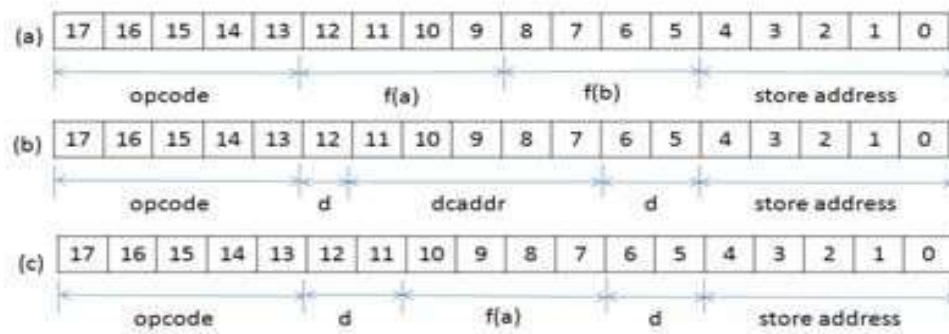


Fig. 3. Overall instruction format (a) the instruction architecture of Register operation model (b) the instruction architecture of LOAD model (c) the instruction architecture of STORE mode

3.1 ALU Architecture:

In this 16-bit RISC processor architecture, multiply- accumulate operation is implemented whereby the multiplier and adder functions are configured independently of the processor's ALU [8]. The multiplier is made of the 16x16-bit modified Vedic algorithm, Wallace tree and CLA. It can perform 16x16-bit signed number multiplying operation. In order to improve the speed of the multiplier, a improved architecture of Vedic encoder is proposed, and 8-2 compressor and CLA is optimized. The modified Vedic multiplier that involves a plurality of selectors to produce partial product then use Wallace Tree to compress them to improve the speed of the multiplier. According to the column, Wallace Tree will divide all the partial products into groups, each column corresponds to a set of adders and is executed together at the same time. Carry signal produced by the front column is sent to the after one to create new partial product. Recycling the process until all partial products are compressed to only two lines [7], then with CLA add the two rows of partial product together to get the final result.

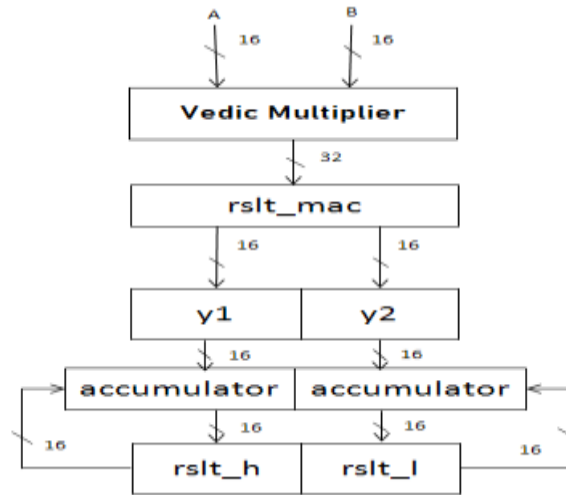


Fig. 4 The overall MAC architecture

Fig. 4 shows the 8-2 compressor architecture and Wallace Tree architecture. The 16-bit multiplication result is stored in register pair y1 and y2 where register y1 contained the higher 16-bit data and y2 contained the lower 16-bit data. The accumulate operation is done by adding y2 with previous value of rslt_l, adding y1 with previous value of rslt_h. Then the data in registers y1 and y2 can be relayed back to rslt_h and rslt_l. In the FPGA implementation, all RISC processor modules are instantiated and synthesized using the Altera Cyclone Ć platform. The outputs of register rslt_h and rslt_l has been observed during the MAC instruction execution cycle.

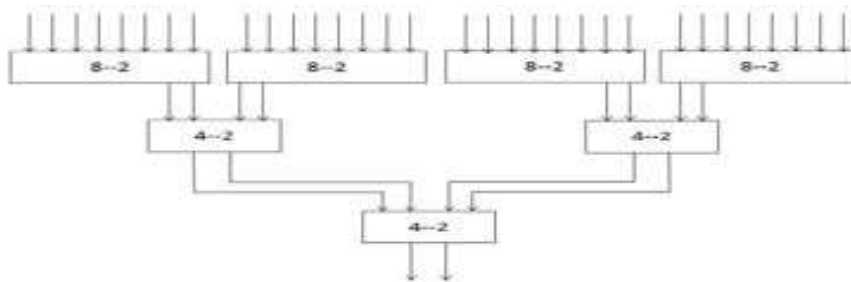


Fig.5 Wallace Tree architecture

4. IMPLEMENTATION

This work has designed a 4-bit Vedic ALU using dataflow modelling style, with help this proposed ALU a 16-bit LAU is developed using structured modelling and finally 16-RISC processor has been developed using 16-bit ALU.

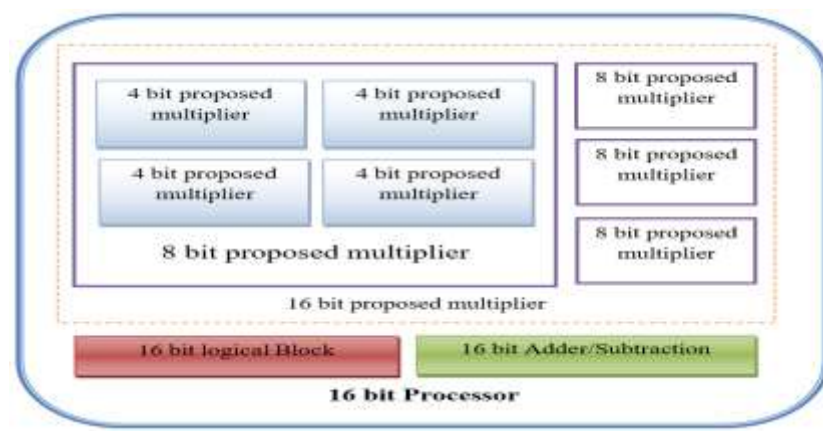


Fig 6: Proposed Architecture of 16-bit RISC processor.

Figure 6 shown is the design of our proposed 16-bit Processor.in the proposed ALU we use 4-bit Vedic multipliers for the design of 8-bit multiplier, and use four 8-bit Vedic multipliers to perform computation on 16-bit input numbers. Figure 7 shows below is the design of our proposed 4-bit Processor. In

the program entity “vedicmau” is the 4-bit Processor. here In1 & In2 are the 4-bit numbers, for multiplication we use vertical and crosswise approach. IN1 and IN2 are two different inputs.

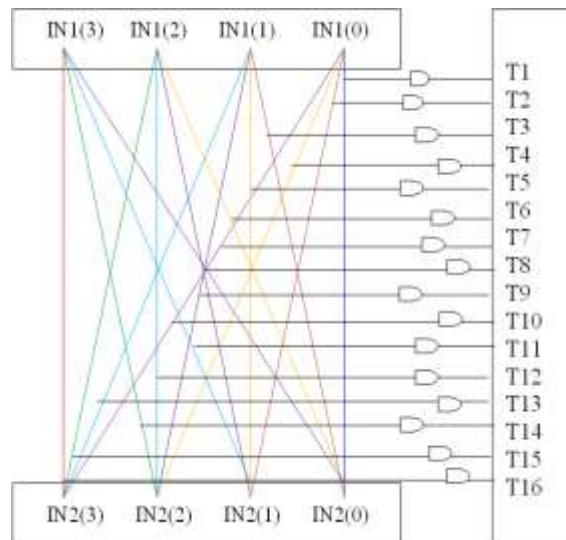


Figure 7 The Vedic multiplication architecture

The partial products of multiplication are outputs T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15 and T16 are the outputs after logical AND operation. The value obtains after logical AND cross wise are t1, t2, t3.....t16. all we required to add these partial products.

		T16	T14	T11	T7	T4	T2	T1
			T15	T12	T8	T5	T3	
				T13	T9	T6		
					T10			
S7	S6	S5	S4	S3	S2	S1	S0	

Figure 8: Vedic Intermediate data

Full adder	Half Adder		ANS				
T16	T14	T11	T7	T4	T2	T1	
	T15	T12	T8	T5	T3		
		T13	T9	T6			
T16	T14	S3	S2	S1	T2	T1	
	T15	C2	C1		T3		
	C3		T10				
T16	S5	S3	S4	S1	T2	T1	
C5		C2	C1		T3		
		C4					
T16	S5	S6	S4	S1	T2	T1	
C5	C6	C11	C1	C9	T3		
C13	C12		C10				
C14	S14	S13	S12	S11	S10	S9	S7

Figure 9: the proposed addition structure

Figure 9 above shows the proposed addition approach in this addition approach first the LSB put as it is then we add the 3 bits data only, after that we add remaining data. Using proposed addition tree, we used 9 full adder (FA) and 4 half adders (HA) however the Wallace tree [6] addition used 7 FA and 8 HA, we reduced 5 HA and increase 2 FA. As we know 1 FA can be developed with 2 HA, so we can say 2 FA will cost 4 HA and in this work we increase 2 FA (equivalent to 4 HA) and reduce 5 HA, hence still we saved 1 HA, so in one 4x4 bit multiplication with proposed addition structure we saved 1 HA, and for 8x8 multiplication we need four 4x4 multiplier hence we saved 4 HA in one 8x8 multiplication and also we know that in one 16x16 multiplication we need four 8x8 multiplier hence we saved 16 HA in 16.x16 multiplication.

5. RESULT AND DISCUSSION

Table 1 below shows the comparative results in results parameters of number of FPGA slices and logical delay (ns).

Table 1: Comparative results of proposed Vedic 16x16 with others.

Platform used: Vertex6-XC4VLX25			
Work By	Logical Delay	Slice	LUT
Proposed Vedic 16-bit RISC Processor	7.261 ns	384	689
Ankita Yadav et al [1]	10.80 ns	--	1293
Shraddha M. Bhagat et al [2]	--	421	831
Chiranjeevi G.N et al [3]	19.868 ns		

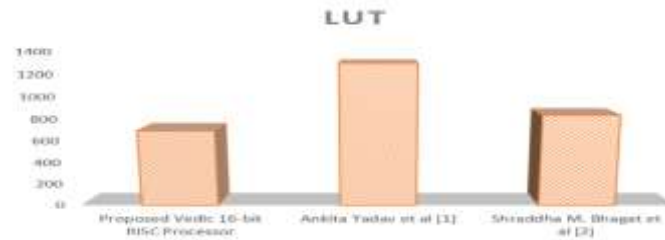


Figure 10: Look up Table uses comparison

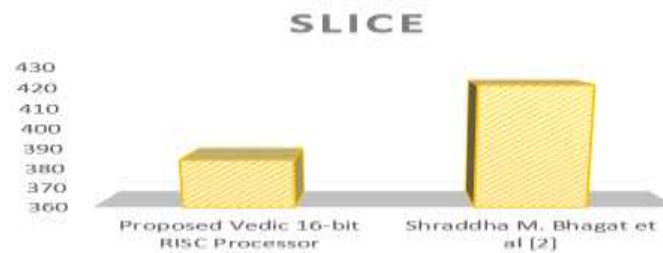


Figure 11: Number of Slices uses comparison



Figure 12: Logical delays comparison for speed

The results are been generated after RTL entry in Xilinx EDA tool. Verification is done on Xilinx ISE and all results are been verified correctly. Above as can observe that proposed results are batter in terms of area (means a smaller number of Slice) and speed (means less logical delay) in 16-bit RISC Processor and multiplication as compare to base papers. above as can observe that proposed results are batter in aspect of speed (means logical delay) in 16-bits as compare to base papers 1 and 2. Proposed work have design 4-bit Vedic multiplier (original research work) and use it to design 16-bit Processor. RTL view in figure 13 describes the exact behavior of digital circuit on the chip as well as interconnection between I/P & O/P. The use of structure to control the synthesis process is the main ingredient for success when writing RTL descriptions. Structure is defined as the way in which parts are arranged or put together to form a whole, and it is created through the use of modules and cell instantiation. Figure 13 below shows the RTL implementation of proposed RISC processor Design.

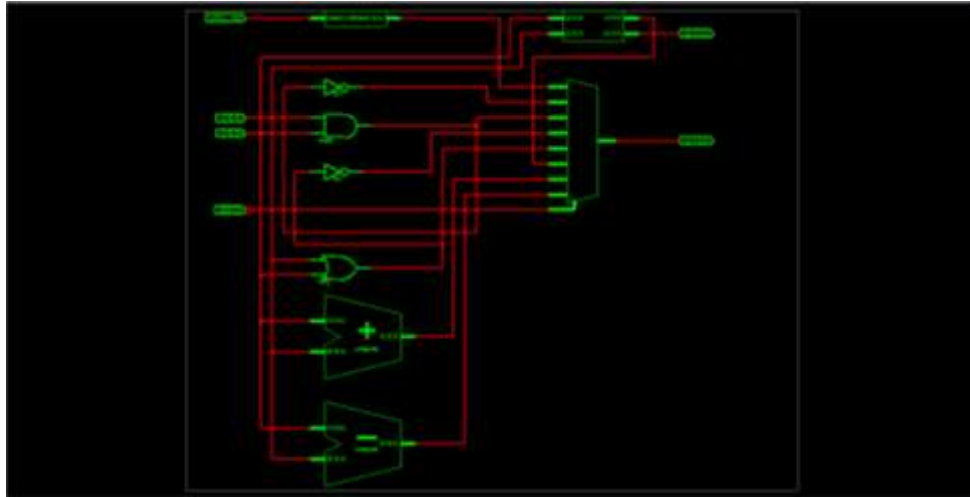


Figure 13 The RTL view for proposed Processor

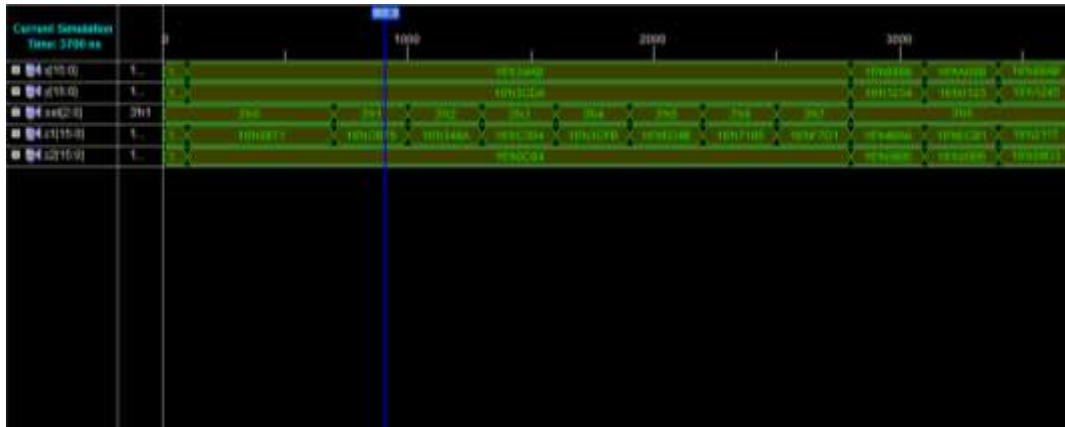


Figure 14 simulation results observed for proposed Processor

all the simulation performed on Xilinx ISE simulator.in the fig-14 shown the result for 16-bit ALU. when the select input is (101) then only multiplication operation is performed .and at that time we get 32-bit output in z1, z2. Table 2 shows the results observed for 16x16 multiplications and it is been observed for many other values dissertations work shows three results observed and found corrected

Table 2 simulation results

OPCODE	IN1	IN2	OUT1	OUT2	Test
000 (Logical x-or)	34AB	3CDA	0871	--	Ok
001 (Logical nand)	34AB	3CDA	CB75	--	Ok
010 (Logical and)	34AB	3CDA	348A	--	Ok
011 (Logical nor)	34AB	3CDA	C304	--	Ok
100 (Logical or)	34AB	3CDA	3CBF	--	Ok
101(multiplication)	34AB	3CDA	ED9E	0C84	Ok
110(addition)	34AB	3CDA	7185	--	Ok
111 (subtraction)	34AB	3CDA	F7D1	--	Ok
101(multiplication)	8888	1234	4BA0	09B5	Ok
101(multiplication)	A00B	0123	EC81	00B5	Ok
101(multiplication)	89AB	1245	2117	09D3	Ok

6. CONCLUSION

This paper extends the new MAC function based on the essential logic and arithmetic operation of RISC processor, designed a simple 5-level pipelined RISC processor that is different from the classical one. The extensional MAC is made of the modified Vedic multiplier that involves the 16x16-bit Vedic

algorithm, Wallace tree and CLA and accumulator. RISC processor is described by using the top-down design method and Verilog HDL. Through the implementation of RISC processor with MODELSIM 6.5e, the simulation result shows that correct output has been observed and the MAC architecture has been verified and synthesized on FPGA platform successfully. Additionally, the successful operation of MAC provides a good basis for more complex application development in digital signal processing through RISC processor.

REFERENCES

- [1] A. Yadav and V. Bendre, "Design and Verification of 16 bit RISC Processor Using Vedic Mathematics," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 759-764, doi: 10.1109/ESCI50559.2021.9396965.
- [2] S. M. Bhagat and S. U. Bhandari, "Design and Analysis of 16-bit RISC Processor," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697859.
- [3] G. N. Chiranjeevi and S. Kulkarni, "Pipeline Architecture for N=K 2L Bit Modular ALU: Case Study between Current Generation Computing and Vedic Computing," 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-4, doi: 10.1109/I2CT51068.2021.9417917.
- [4] S. N. S. Vishnu, A. Gandluru and R. S. R., "32-Bit RISC Processor Using Vedic Multiplier," 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824747.
- [5] J. Kuppli, M. Abhiram and N. A. Manga, "Design of Vedic Mathematics based 16 bit MAC unit for Power and Delay Optimization," 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), 2021, pp. 1-4, doi: 10.1109/ICNTE51185.2021.9487570.
- [6] S. Dhanasekar, P. M. Bruntha, L. J. Ahmed, G. Valarmathi, V. Govindaraj and C. Priya, "An Area Efficient FFT Processor using Modified Compressor adder based Vedic Multiplier," 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), 2022, pp. 62-66, doi: 10.1109/ICDCS54290.2022.9780676.
- [7] A. A. Hayum, S. Chinnapparaj, G. Sujatha, G. T. Selvi, M. Naved and P. Mohanraj, "Review of Vedic Multiplier Using Various Full Adders," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 644-647, doi: 10.1109/ICCMC51019.2021.9418339.
- [8] K. Arunkumar, P. Mangayarkarasi, B. Jackson and A. A. Juliette, "Design of High Speed, Low Power 16x16 Vedic Multiplier With Adiabatic Logic," 2022 8th International Conference on Smart Structures and Systems (ICSSS), 2022, pp. 1-9, doi: 10.1109/ICSSS54381.2022.9782274.
- [9] N. H. Sastry, J. B. S. Bharadwaj and G. S. Jeevith, "Design and Implementation of 8-Bit Vedic Multiplier in 18nm FinFET Technology," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021, pp. 251-256, doi: 10.1109/RTEICT52294.2021.9573681.
- [10] H. Jing-yu, L. Li-li, Z. Yan-chao, Y. Wen-tao and Y. Jian-hong, "Multiply-accumulator using modified booth encoders designed for application in 16-bit RISC processor," 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013, pp. 416-419, doi: 10.1109/IMSNA.2013.6743304.
- [11] S. Islam, D. Chattopadhyay, M. K. Das, V. Neelima and R. Sarkar, "Design of High-Speed-Pipelined Execution Unit of 32-bit RISC Processor," 2006 Annual IEEE India Conference, 2006, pp. 1-5, doi: 10.1109/INDCON.2006.302780.
- [12] Y. d. Ykuntam, K. Pavani and K. Saladi, "Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225404.
- [13] S. N. S. Vishnu, A. Gandluru and R. S. R., "16-bit RISC Processor Using Vedic Multiplier," 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824747.