ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

Design And Implementation Of 32-Bit MIPS Risc Processor with Flexible 5-Stage Pipelining and Dynamic Thermal Control in Xilinx Vivado

[1] K.G. VENKATA KRISHNA, [2] DEVANURI REVATHI, [3] BUSIREDDY MEGHANATH REDDY, 4 TANDU JAYA PHANEENDRA PAVAN

[1] Assistant Professor (C), [2,3,4] UG Student

[1,2,3,4] Department of Electronics and Communication Engineering, Krishna University College of Engineering and Technology Krishna University, Rudravaram, Machilipatnam.

[1] Ph.:9966963399 Email Id: kgvk.aca.ece@kru.ac.in

[2] Ph: 8688276283, Email id: revathidevanuri2003@gmail.com

[3] Ph: 9701099359, Email id: meghanathreddybusireddy@gmail.com

[4] Ph: 7842878450, Email id: tandupavan43@gmail.com

Abstract— Reduced The main aim of this design is designing a 5-stage flexible pipelined 32-bit RISC-V processor using system Verilog including it Dynamic thermal management technique. This Design is be based on MIPS instruction set architecture (ISA) in which stages of pipeline include Instruction fetch, Instruction Decode, Execute, Memory access, Write back. The RISC [Reduced Instruction set Computed] compared to CISC [Complex instruction set computer] executes instruction in one clock and instructions are always uniformly lengthen and fixed instruction format-based opcode. Hence RISC is preferable more used for less complex tasks and low power designs. Based on the literature the proposed design brings significant improvements in simulation speeds due to flexible pipeline technique and low power consumption is achieved. The Primary desire of this paper is to simulate and synthesize the design by perform basic ALU operations and provide desired outputs.

Keywords—RISC, ALU, RISC, CISC, MIPS, ALU, ISA

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

INTRODUCTION

RISC [Reduced Instruction Set Computers] are extensively used in all type of computational tasks especially in the area of basic scientific computing such as DSP and DIP etc. There are a variety of processor types on the market, including some of the processors that come with Hardware Description Language (HDL) like VHDL (Very High-speed Integrated Circuit HDL) and Verilog-HDL is used to write a specific type of processor. RISC (Reduced Instruction Set Computer) is an efficient computer architecture which is used for high- speed, low-power applications[1] . The processor uses the reduced instruction set (RISC) to allow pipelined execution of instruction, increasing performance and throughput performance. We included Dynamic thermal management unit using dynamic frequency scaling for thermal stability of processor. The processor can work at a high clock frequency and thus yields higher speed. More recently Verilog is used as an input for synthesis programs which will generate a gatelevel description for the circuit. The RISC-V architecture comprises of load and store operation. This means it has instruction called LOAD for accessing data from the memory and STORE instruction for write the data back to memory[14].The architecture is called Hardwired architecture and it will increase the overall performance of RISC processor. Our design is based on synchronous technique which is fast and reliable technique in the present industry processor.

People began looking at other options as the controller architecture in CISC became more difficult and the performance was also below expectations. It had been discovered that speed is eliminated when a CPU communicates with memory. The only way to improve CPI was to keep the instruction set as basic as possible. Simple in terms of appearance rather than functionality. Because of this, relatively few instructions—probably only load and store—in a typical RISC architecture require the CPU to access data from memory. In the end, pipelining increased performance by a new dimension just by adding a few extra registers, which lowers CPI and raises throughput. As a result, the command can be successfully carried out in a signal.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

LITERATURE REVIEW

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 computational world, fast processing units are a requirement for many of real time applications. These units contain ALU and MAC as the fundamental blocks which are essential for efficient and fast execution. Multipliers are fundamentally utilized in Digital signal processors as its main component. To maintain the accuracy and increase the speed of execution multiplier, adder, registers have to be modified for which the ALU & MAC can have better performance. Design of faster multipliers is emphasized for its implementation in processors due to the increasing constraints on delay. To enhance the speed of multiplication, there is very high importance of designing faster multipliers. Among several multipliers, Vedic multipliers are preferred for their speed of operation, area usage and low power consumption over the other existing multipliers. Out of this, algorithms which are based on Vedic mathematics are intensive designing fast and low power multipliers. Vedic mathematics has sixteen sutras, out of which UrdhvaTiryakbhyam, EkanyunenaPurvena, EkadhikenaPurvena are discussed here along with simulation results. The focus of this paper is to achieve best results for area, speed, and power parameters for each of the sutras.

This paper presents highly efficient Vedic multiplier unit using various Sutras of Vedic Mathematics. Design, synthesis and simulation of 16-bit Vedic multiplier unit is done using Vivado 17.1 and Verilog. And thus, the performance parameters for multiplier like Delay, Power and Area for each sutra is analyzed. It is observed that 70.26% of reduction in area is achieved using EkadhikenaPurvena sutra with 90.41% increase in speed as compared to UrdhwaTiryakbhyam. Nikhilam sutra gives optimum area and delay, where it is the most effective in terms of power. Significant performance values are given for each of the sutra whic

can be used for selection of multiplier in image and signal processing applications along with digital filters.

Summary: from this implementation area has been reduced with an increase in speed of ekadhikenapurvena sutra and nikilam sutra gets better area and delay

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 includes the designing of 16- Bit RISC processor and modeling of its components using Verilog HDL. The processor is based on Harvard architecture. The instruction set adopted here is

extremely simple that gives an insight into the kind of hardware which should be able to execute the set of instructions properly. Along with sequential and combinational building blocks of a processor such as adders and registers more complex blocks such as ALU and memories have been designed and simulated. The modeling of ALU which has been done in this project is fully structural starting from half adders. At the end the semi-custom layout has been developed for ALU. Complex blocks such as memories have been modeled using behavioral approach, whereas simple blocks such as adders have been done through structural approach.

Thus, we have designed and simulated a 16-bit RISC processor using Verilog HDL and verified its working by simulation. The processor can be extended to a 32 or even a 64-bit processor in the future by simple changes in the code and the datapath can be altered to include various new blocks which is not possible on a traditional processor unit. The control unit (CU) is a component of the processor. It tells the computer's memory, arithmetic and logic unit and input and output devices how to respond to the instructions that have been sent to the processor. It directs the operation of the other units by providing timing and control signals. Most computer resources are managed by the CU. Hardwired control units are implemented through use of combinational logic units, featuring a finite number of gates that can generate specific results based on the instructions that were used to invoke those responses. Hardwired control units are generally faster than the micro programmed designs.

Summary: In this implementation compared to conventional risc processor this can achieve high performance as the datapath is being changed.

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

The procedure of design and verification for a 16-bit RISC processor is introduced in this paper. The proposed processor has Harvard architecture and consists of 24-bit address, 5-stage pipeline instruction execution, and internal debug logic. ADPCM vocoder and SOLA algorithm are successfully carried out on the processor made with FPGA. The portable multimedia player (PMP) and the personal digital assistant (PDA) are not special item of people. So the low power processor and small size processor are made as SOC level ASIC (Application Specific Integrated Circuits). The popular processors on SOC level ASIC are the 8051 processor and the ARM 7 processor. The ARM 7 is used on SOC level ASIC which is millions gates size. And the 8051 processor is used on simple system which needs the small size But the bit size of simple system is changing 8 into16. So the 8051 processor calculation time is increasing. To replace the 8051 processor the new 16bit RISC

processor is proposed in this paper. The proposed processor is designed to embed on SOC at ASIC. The core execution test is done. But for on-chip debugging GDB server program is needed for application debugging. The figure 5 shows the connection of the GDB server program, the SW debugger and the new core. And for fast programming compiler is also needed. Next target of the proposed processor is to develop a compiler and GDB server program for high level user.

EXISTING METHOD

The function of the processor is to execute each and every instruction set efficiently as per the machine language. ALU is the combinational circuit which means Arithmetic and Logical Unit. This unit is designed to perform various numbers using various instruction sets. In Processor, ALU inputs consist of instruction (machine word) which is operation code (opcode) and some operands. So, the opcode tells the ALU which and what operation is to be performed then these operands are used in the operation.

There is a small set of data holding place that is known as Register bank. The ALU stores the result of operation in accumulator which later on is placed in a storage register and it checks the bits and indicates whether the operation was performed successfully. If not successfully executed then some type of status will be shown i.e. even known as Z-Flag or status register. Its function is to execute programs and operate efficiently for the data stored in memory. A processor has a set of instructions which is nothing but a command to perform a task in a computer. The control unit holds the instruction to be executed. In CPU, the registers such as address register, data register and an instruction register is present. The performance of the CPU is to fetch, decode and execute the operations on memory according to the registers. The task of IR includes the decoding the op- code, determining the instruction, determining which operands are in memory, retrieving the operands in memory then assigning a command to a processor to execute the instruction. This is done with the help of control unit which generates the timing signals that controls the various processing elements which involves in execution of the instruction.

PROPOSED METHOD

The Our System mainly focused on flexible pipelining and dynamic thermal control which in turn provides low power consumption and faster performance speed of the processor. Main features include 32-Bit MIPS (Microprocessor without Interlocked Pipeline Stages) Architecture based Processor, Synchronous Pipeline Implementation. Supports 48 Instructions. RISC architecture used

follows single-cycle instruction execution. Low power and high-performance pipelining technique is being used.[1] It majorly Contains three addressing modes: Immediate mode: In this addressing mode the operand is explicitly specified. Register-Register mode this mode defines the transfer of data between two registers Memory Mode: This mode represents the data I/O between external program and ram memory.[5] Pipeline technique: Pipeline can defined as processing a set of data elements in series order having connections where one output of one processed element is input to next new element. This technique comes under Harvard Architecture which is known for its speed and performance.[4]

1. Instruction fetch (IF): The processor reads the next Instruction to be executed.

2. Instruction decodes (ID): Processor works out what this Instruction is.

3. Instruction execution (EX): The processor executes Instruction on operand values.

4. Memory Access (MEM): The Processor will cess the input and output memory.

5. Write Back (WB): Processor writes the result of the operation in the register/memory.

MIPS: [Microprocessor without Interlocked Pipeline Stages] architecture:

Five-stage of execution pipeline: fetch, decode, execute, memory-access, write-result. Regular instruction set which means all instructions are 32-bit.Three-operand arithmetical and logical instructions.32 general-purpose registers of 32- bits each are present. No status register or instruction sideeffects. No complex instructions (like stack management instruction, string operations, etc.)[6]

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

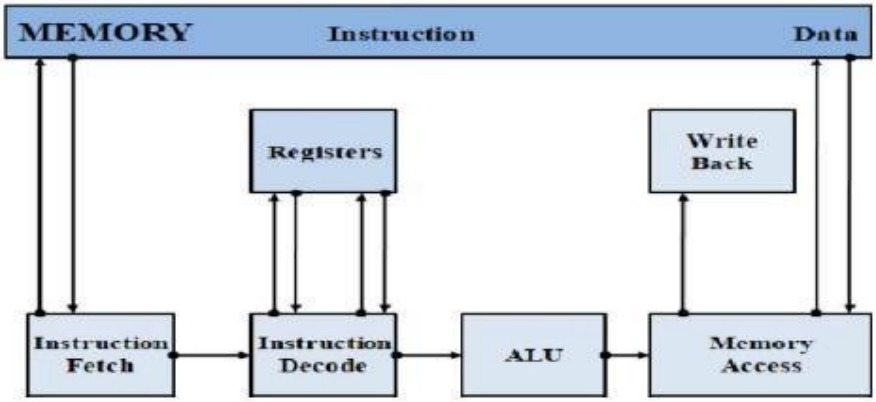

Fig.1: Block diagram of processor

Flexible Pipelining: The pipeline is made such a way that it can flexibly to switch between 4 stage and 5 stage with based on instruction type. Thus by reducing the data dependency the processing time is reduced. This feature was included to enhance the overall speed of the processor based with less simulation time

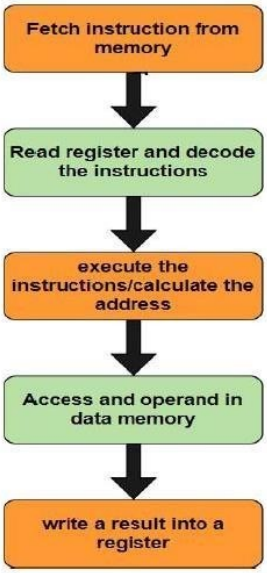

Fig.2: Flowchart of Instruction flow

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

RESULTS

1.1 RESULTS

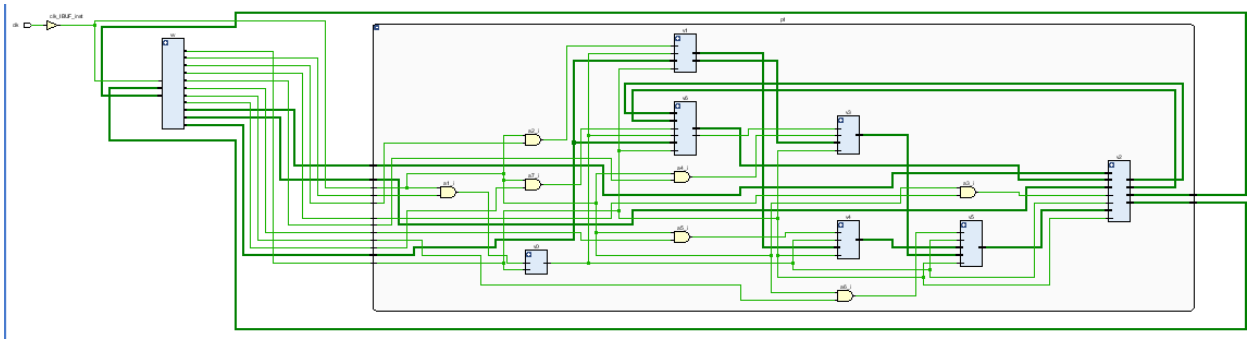

Fig.3: RTL

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT as Memory (9600)	LUT Flip Flop Pairs (20800)	DSPs (90)	Bonded IOB (106)	BUF
top	1298	1446	15	525	1274	24	605	1	1	
dbg_hub (dbg_hub)	473	727	0	225	449	24	303	0	0	
p1 (processor)	557	160	15	188	557	0	81	1	0	
vv (vio_0)	268	559	0	131	268	0	211	0	0	

Fig.4: Area

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay
Path 157	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[21]/D	4.018	0.248
Path 158	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[24]/D	4.010	0.248
Path 159	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/BS...canid_reg[5]/D	4.010	0.248
Path 160	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[22]/D	3.927	0.248
Path 161	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/BS..._state_reg[1]/D	3.843	0.248
Path 162	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[12]/D	3.843	0.248
Path 163	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[27]/D	3.841	0.248
Path 164	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[18]/D	3.835	0.248
Path 165	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[17]/D	3.816	0.248
Path 166	∞	2	2	36	dbg_hub/inst/B...can_inst/SHIFT	dbg_hub/inst/B...anid_reg[23]/D	3.759	0.248

Fig.4: Delay

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

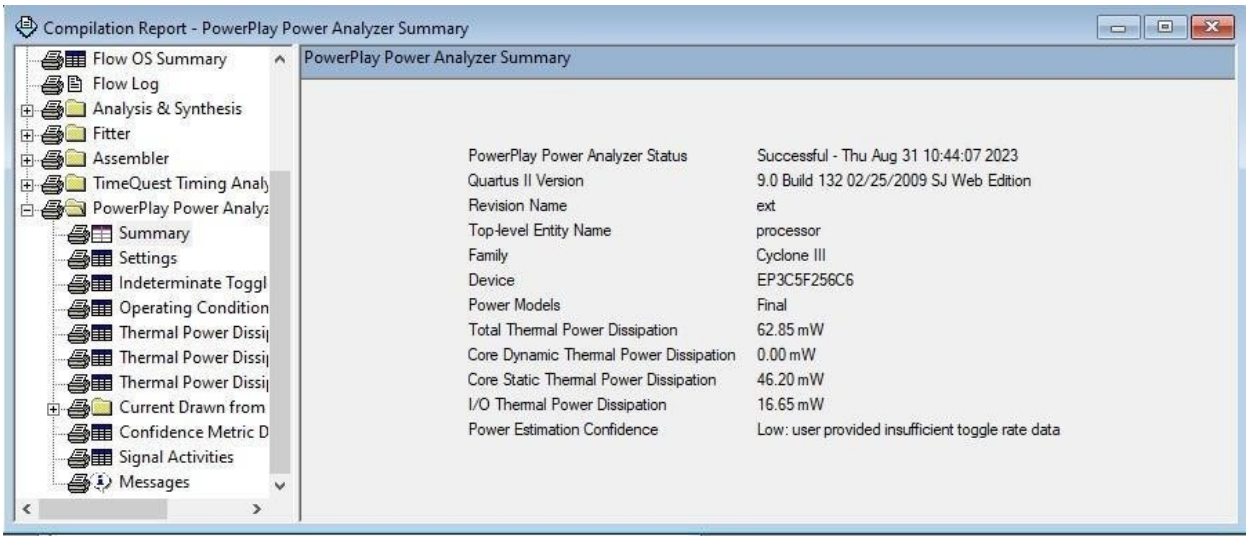

Fig.5: Power

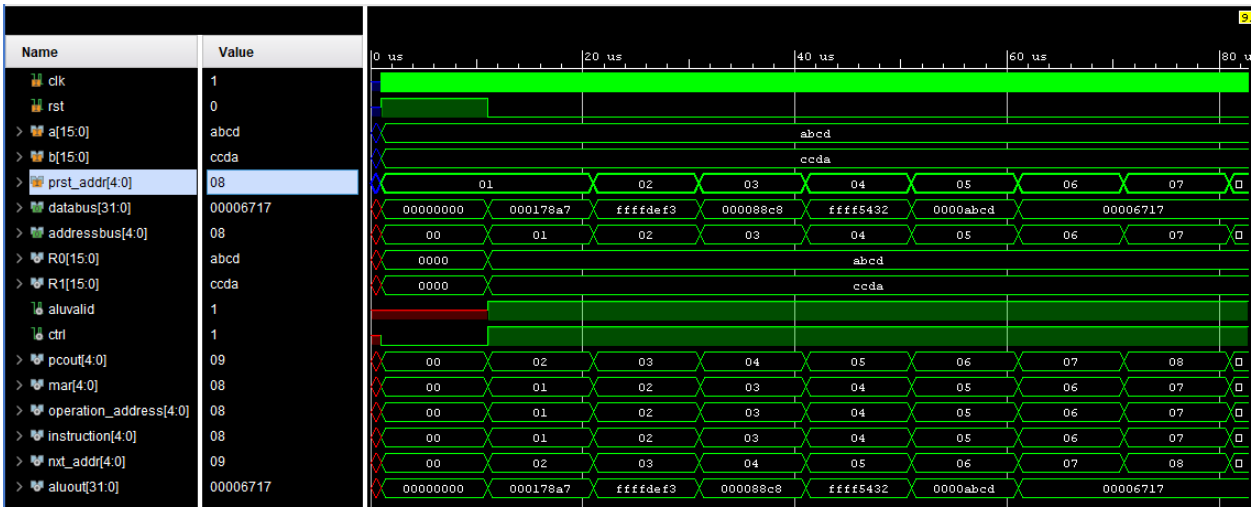

Fig.5: Simulated Output

	Area (LUT's)	Delay (ns)	Power(mW)
existing	814	3.210	119.89
proposed	1298	4.018	62.85

Table.1: Comparison Table

CONCLUSION

Design, simulation, functionality testing, and evaluation of the 16-bit processor was done in the proposed work. Also by applying clock gating at a fine-grained level allows for power optimization, the performance of 16-bit RISC processor will be increased. Modules are grouped in accordance with the instructions to perform instruction level clock gating. A group of flip flops' clock activation functions are extensively examined for both on and off states. In the comparison table power and area evaluated before and after the use of the clock gating approach. The developed CPU has a lower absolute power dissipation than the 16-bit processor created

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.

[12] Xilinx13.4, "Synthesis and Simulation Design Guide", UG626 (v13.4) January 19, 2012.

[13] Xilinx 13.1, "RTL and Technology Schematic Viewers Tutorial", UG685 (v13.1), March 1, 2011.

[14] Xilinx, "7 Series FPGAs Configurable Logic Block", UG 474 (v 1.5), August 6, 2013.

[15] Xilinx 12.4, "ISim User Guide", UG660 (v 12.4), December 14, 2010.

[16] JikkuJeemon, "Low power pipelined 8-bit RISC processor design and implementation on FPGA", ICCICCT 2015.

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

[18] D. J. Smith, "HDL Chip Design", International Edition, Doone Publications, 2000.

[19] J.F. Wakerly, "Digital Design: Principles and Practices", Third Edition, Prentice-Hall, 2000.

[20] A. S. Tanenbaum, "Structured Computer Organization", Fourth Edition, Prentice-Hall, 2000.

[21] Yatin Trivedi and others, "Verilog HDL", IC, 2000.

[22] Mauriss M Mano, "Digital Design", Third Edition, Perason Edition, 2000.

[23] Shraddha M. Bhagat and Sheetal U. Bhandari, "Design and Analysis of 16-bit RISC Processor", Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

K.G.VENKATA KRISHNA
Assistant Professor (C)
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam.
Ph.: 9966963399
EId: kgvk.aca.ece@kru.ac.in

DEVANURI REVATHI
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam
Ph:8688276283
Eid : revathidevanuri2003@gmail.com

BUSIREDDY MEGHANATH REDDY
B.Tech Student
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam.
Ph:9701099359
Eid : meghanathreddybusireddy@gmail.com

TANDU JAYA PHANEENDRA PAVAN
B.Tech Student
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam.
Ph:7842878450
Eid: tandupavan43@gmail.com