
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 RISC PROCESSOR

[1] K.G. VENKATA KRISHNA, [2] VADDADI DEVI SIVA KUMAR, [3] POTHUKUCHI SREE SHANMUKHA SAI PRIYA, [4] BADDE VENKATA SWAMY

[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:7207459662 Email id: shivakumarbalu15@gmail.com
[3] Ph:8185801940, Email id: priyapothukuchi1926@gmail.com
[4] Ph:6304720218, Email id: baddevenkataswamy4@gmail.com

Abstract— Reduced Instruction Set Computer (RISC) Processors have recently become quite an important aspect of keeping up with all the advances in technology. This research presents a comprehensive study on the design and simulation of an improved RISC processor architecture on Field- Programmable Gate Arrays (FPGAs). The objective is to enhance the performance, efficiency, and versatility of RISC processors by incorporating novel design techniques and optimizations. The proposed design mainly focuses on the improved design of the Arithmetic and Logic Unit (ALU). The effectiveness of the proposed design is evaluated through extensive simulations. The results demonstrate significant improvements in performance in time and total power. The findings of this study indicate that the improved RISC processor design offers a promising approach to address the increasing demands of modern computing systems.

Keywords— Reduced Instruction Set Computer (RISC), Arithmetic and Logic Unit (ALU), One hot encoding, Verilog, Processor, Field Programmable Gate Array (FPGA).

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

INTRODUCTION

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 single clock cycle.

It's a widespread misconception that instructions are simply deleted to produce a smaller set of instructions when the term "Reduced Instruction Set Computer" is used. The size of RISC instruction sets has actually increased over time, and currently many of them contain a greater number of instructions than many CISC CPUs. The word "Reduced" in that sentence refers to the fact that compared to the "complex instructions" of CISC CPUs, which may require multiple data memory cycles to execute a single instruction, any given instruction performs less work, accomplishing at most one data memory cycle.Because RISC design uses streamlined machine instructions for commonly utilized functions, research has demonstrated that it significantly increases computer performance.

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.

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.

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 into

16. 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

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 architecture of the 16 bit RISC processor. There are 20 fundamental instructions in the processor, including data transport, branching, control, and logical and arithmetic instructions. The processor is made up of the 16-bit register sets R0 through R7, PC, IR, RegY, and Add_reg R. Logic and arithmetic operations are designed to be carried out via the 16 bit ALU. Depending on the instruction being followed, the control unit generates the control signals. Idle, fetch, decode, and execute are the basic states of the control unit's state machine. There are two flag bits in the processor: 0 and Carry.

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

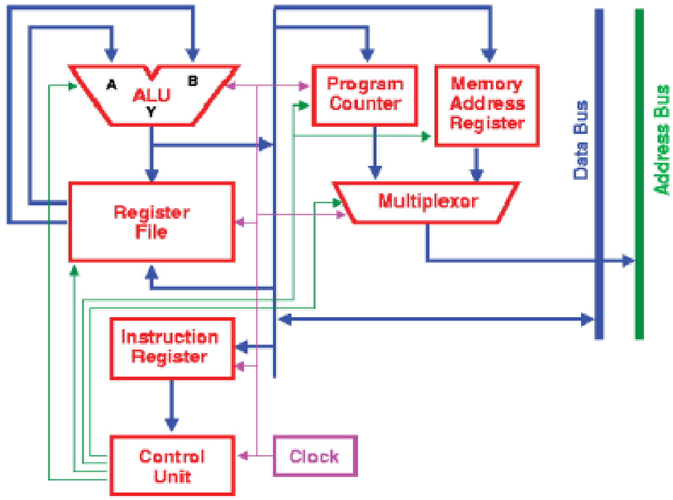


Fig.1: block diagram of processor

Figure 1 depicts the processor's architectural layout. There are two buses, Bus_1 and Bus_2, which are each driven by a separate Mux. This multiplexer has an 8:1 ratio. A 3 bit choose line is used to select any of the registers R0 to R6 and PC, which are input to Mux-1. The Bus_1 is being driven by the output of the Mux_1.ALU, Bus_2, and Memory all get the output of Bus_1 as their input. The output of Mux_2 is driving the Bus_2 and is a 4 to 1 mux that employs a 2 bit select line to choose the ALU output, Bus_1, and memory word. Using the appropriate load _x signal from the control unit, the data from the Bus_2 is loaded into any one of the 16 bit registers. Figure 3 displays the processor's instruction format. The three-bit address identifies the source and destination registers. There are a total of 32 potential instructions because the opcode is only 5 bits long.

Given that there are 4 bits left over for potential future use, there is room to increase the number of instructions and registers. When a memory location serves as the source or destination, the instruction's second word will provide the memory location's address. The current instruction to be executed's address is stored in the program counter. Bus-1 and Bus-2 are used to transport the contents of the program counter to the address register. The program counter is increased as well as the contents of the memory location pointed by the address register are sent to the instruction register via Bus_2. To carry out the operation, the control unit generates the control signals after decoding the instruction. Once the processor has been created, all of the instructions should be tested for functionality. By using the clock gating technique at a fine-grained level, the processor is optimized for power dissipation. Clocks are turned off in a circuit using the clock gating technique in order to reduce power consumption by reducing the amount of time that logic modules

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

spend idle. The units for which the clock is to be gated have been identified in the processor architecture, and each module's condition for gating is assessed separately. The clock is masked with an AND gate in the case of a register file because only the source and destination registers are used during execution and the remaining registers are idle.

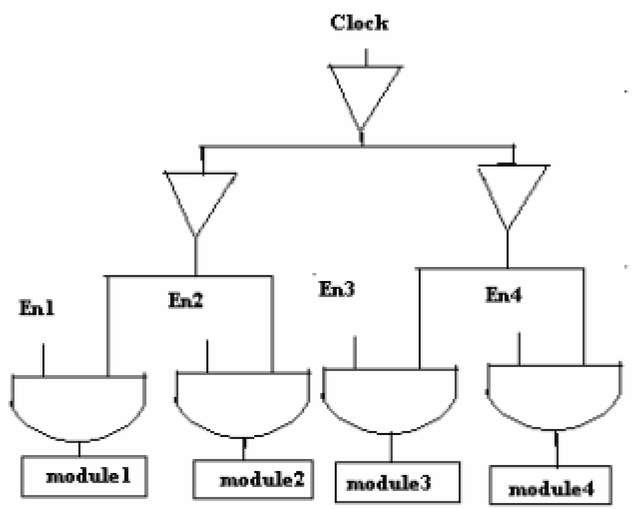


Fig.2: Clock Gating at fine grained level

The masking AND gate with an enable signal is used to connect the various modules to the clock in Figure 2. Carefully examining the processor's functionality and timing diagram yields the information needed to determine the prerequisites for activating the enable signal for individual modules.

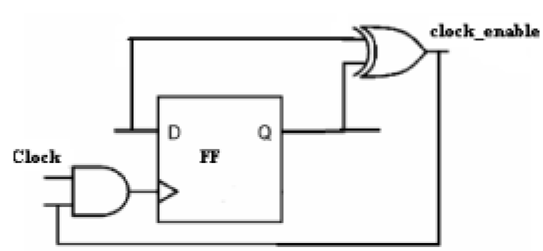


Fig.3: Clock enable signal for flip-flop

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

RESULTS

1.1 RESULTS

1.1.1 RTL Schematic:

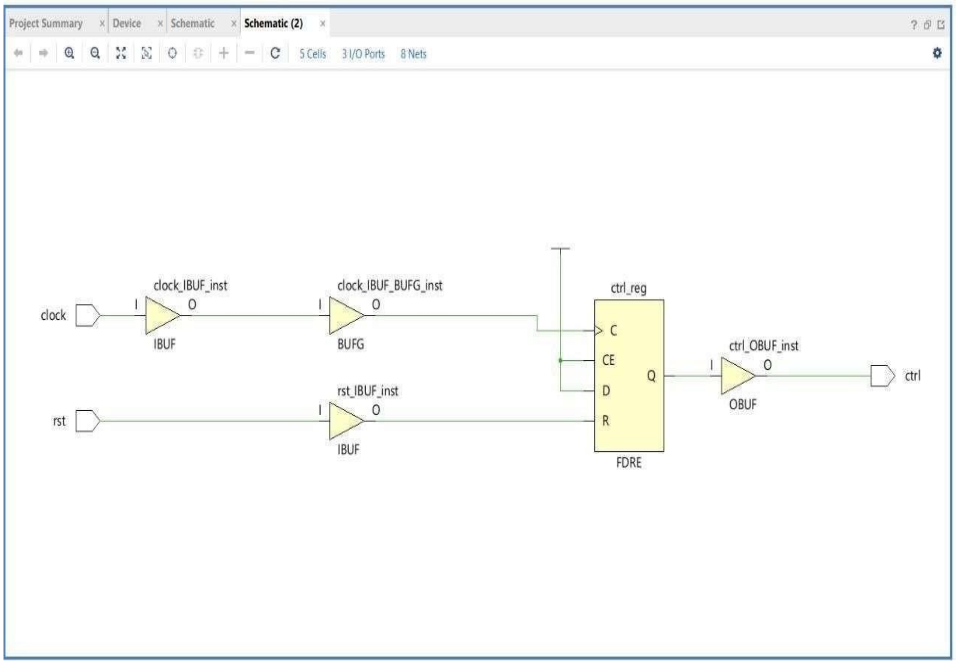
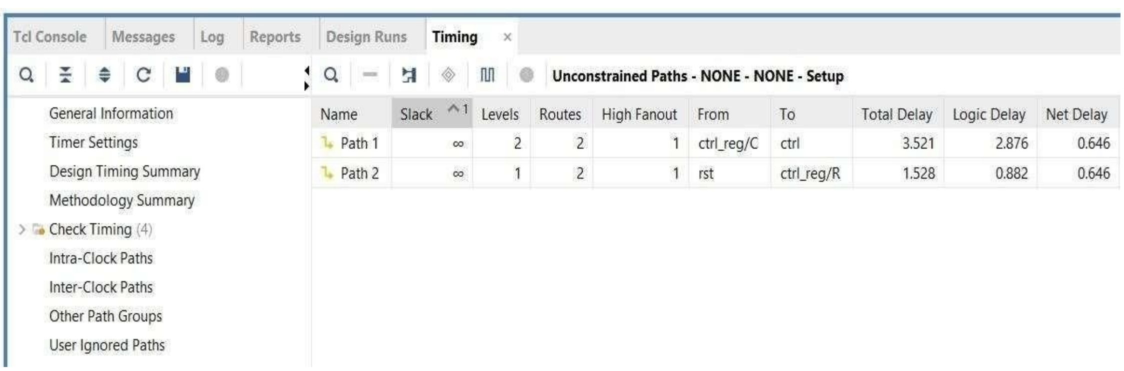


Fig.6.1 RTL schematic

6.2 Delay:



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	2	2	1	ctrl_reg/C	ctrl	3.521	2.876	0.646
Path 2	∞	1	2	1	rst	ctrl_reg/R	1.528	0.882	0.646

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

6.3 Power

Total On-Chip Power:	0.144 W
Design Power Budget:	Not Specified
Process:	typical
Power Budget Margin:	N/A
Junction Temperature:	25.3°C
Thermal Margin:	59.7°C (31.6 W)
Ambient Temperature:	25.0 °C
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Fig.6.3.1 Time

6.4 Simulated Output:

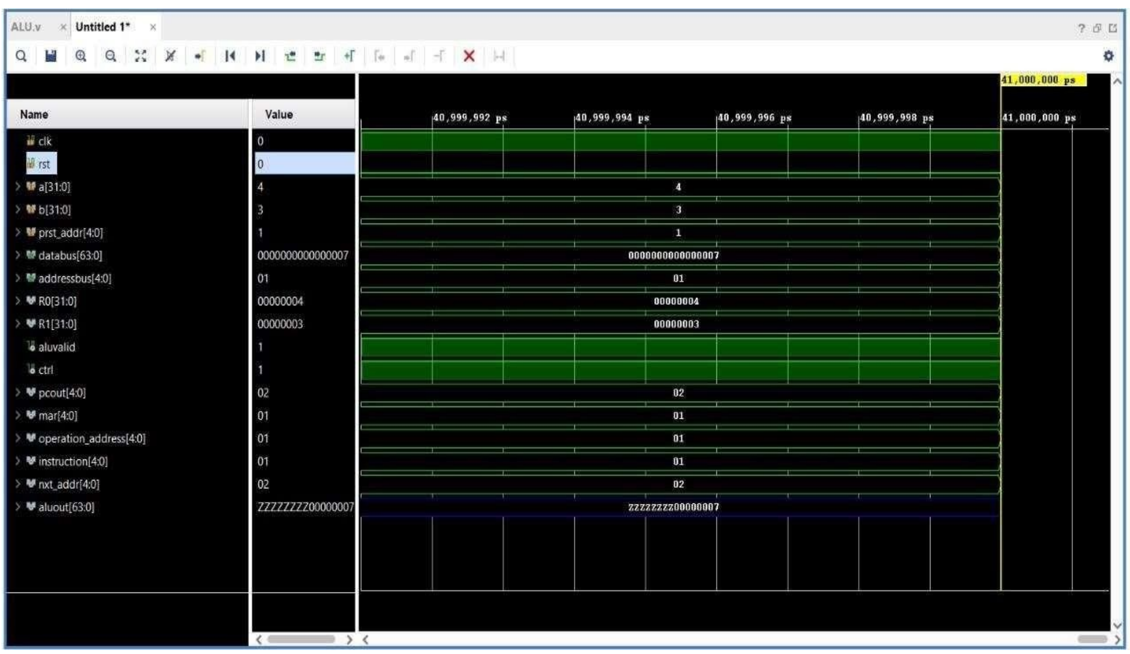


Fig.6.4.1 Simulated Output

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.

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

VADDADI DEVI SIVA KUMAR
UG Student
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam.
Ph:7207459662
Eid : shivakumarbalu15@gmail.com

POTHUKUCHI SREE SHANMUKHA SAI PRIYA
UG Student
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam.
Ph:8185801940
Eid : priyapothukuchi1926@gmail.com

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

BADDE VENKATA SWAMY
UG Student
Department of Electronics and Communication Engineering
Krishna University College of Engineering and Technology
Krishna University
Rudravaram, Machilipatnam.
Ph:6304720218
Eid: baddevenkataswamy4@gmail.com