iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79682

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Design and Implementation of an 8-Bit CPU Using Discrete BJTs and Logic ICs

Mrs. J. R. Khandare[1], Mr. Ajinkya Sanas[2], Mr. Varad Sagade[3], Mr. Omkar Tathe[4], Mr. Tanmay Agrawal[5]
[1]*Professor, Department of Electronics and Telecommunications, AISSMS Polytechnic, Pune, Maharashtra, India*
[2, 3, 4, 5]*Student, Department of Electronics and Telecommunications, AISSMS Polytechnic, Pune, Maharashtra, India*

Abstract: ***In today's world of embedded systems, microprocessors and microcontrollers are widely used in nearly every electronic application. While these devices deliver high performance and compact designs, their internal structure often remains unclear to students and beginners.***

This means that learners frequently do not understand how a processor executes instructions at the hardware level. To tackle this issue, this paper outlines the design and implementation of an 8-bit Central Processing Unit (CPU) created using discrete Bipolar Junction Transistors (BJTs) and standard Logic Gate ICs. The proposed CPU design includes key processing units such as the Arithmetic Logic Unit (ALU), Program Counter (PC), Accumulator Register, Control Unit, and Clock Generator, all built from basic digital components like logic gates, flip-flops, and transistor-based circuits. A 555 timer IC is used as the system clock, enabling the processor to run at a low frequency. This allows for clear observation of individual instruction cycles. The CPU operates on a standard fetch, decode, execute cycle, processing simple instructions such as LOAD, ADD, SUBTRACT, JUMP, and HALT. This project offers a complete view of data paths, control signals, and timing relationships, helping learners connect theoretical ideas with real-world implementation. The system is affordable, works at safe voltage levels, and acts as a useful educational tool for understanding computer architecture and the basics of digital electronics.

I. INTRODUCTION

The increasing number of embedded systems and computer systems has revolutionized modern technologies. These devices depend a lot on advanced processors. Nevertheless, today's processors are highly integrated and consist of millions of transistors in one chip. Despite being more effective, this level of integration leaves no chance for a student to see and learn about the internal workings of the CPU. In the educational system, students get familiar with different processors by coding them via platforms, such as Arduino, Raspberry Pi, and ARM controllers. However, despite offering a good environment for building applications, they do not show the internal structure of the CPU, nor its mechanisms of work, nor its control circuitry. Therefore, the basics of instruction decodig, arithmetic operations, registers usage, and other important processes become too abstract.

It is crucial to build a processor based on simple electronic components. Designing an 8-bit CPU on BJTs and ICs can be considered as an ideal educational example that illustrates how the modern processor came from the basic logical circuits. In particular, the aim of the task is to visualize the entire process of executing instructions.

II. LITERATURE REVIEW

Several educational approaches have been explored to understand computer architecture and processor design at the hardware level. Traditional microprocessors and microcontrollers provide high computational efficiency but conceal internal processor operations, making them unsuitable for in-depth architectural learning. As a result, researchers and educators have increasingly focused on building simplified CPUs using discrete components to enhance conceptual clarity.

Global Science Network demonstrated the construction of a 4-bit computer using individual transistors on breadboards, emphasizing the importance of understanding transistor-level switching and logic formation [1]. This work highlights how basic computing functions can be achieved using minimal hardware, reinforcing the educational value of discrete component-based processor designs. Dominic Boeuf, through the FULXOR platform, explored CPU construction with minimal transistor usage, providing a deep examination of how computer logic can be implemented efficiently while maintaining functional correctness [2]. This approach inspired the optimization of control logic and instruction decoding used in the proposed 8- bit CPU.

Big Chaks documented the complete journey of building a computer system from scratch, demonstrating practical challenges such as timing, signal integrity, and logical synchronization [3]. This work reinforced the importance of modular CPU design and systematic debugging strategies.

©IJRASET: All Rights are Reserved

Additionally, Vocademy provided a detailed explanation of transistor saturation and switching behavior, which is a critical concept in designing reliable digital circuits using BJTs [4]. Understanding saturation characteristics played a vital role in implementing stable control circuitry in the proposed system.

These works collectively influenced the architectural decisions and implementation strategies adopted in this project, validating the feasibility and educational effectiveness of a discrete-component-based CPU.

III. PROBLEM STATEMENT

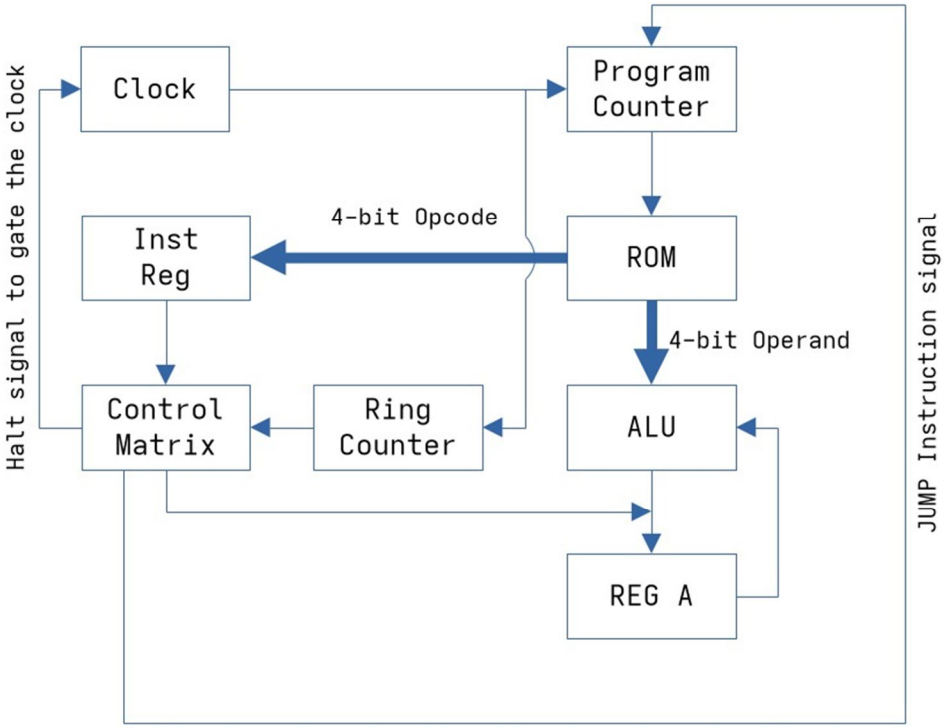
In modern electronic and embedded systems, microprocessors and microcontrollers are widely used due to their compact size, high speed, and high level of integration. However, these highly integrated devices conceal their internal architecture, making it difficult for students and beginners to understand how a Central Processing Unit (CPU) actually operates at the hardware level.

Most educational approaches rely heavily on software-based programming or simulation tools, which provide limited visibility into fundamental operations such as instruction fetching, decoding, control signal generation, and arithmetic execution. As a result, learners often lack a clear understanding of how basic digital components like transistors, logic gates, and flip-flops work together to perform computing tasks.

Additionally, existing learning platforms such as microcontroller kits and FPGA-based systems abstract away transistor-level behavior, preventing students from observing real- time electrical characteristics such as switching, saturation, and propagation delay.

Therefore, there is a need for a simple, low-cost, and hardware-transparent computing system that can demonstrate the internal working of a CPU using fundamental electronic components. The challenge is to design and implement an 8-bit CPU using discrete Bipolar Junction Transistors (BJTs) and logic ICs that allows step-by-step visualization of the fetch–decode–execute cycle, thereby bridging the gap between theoretical knowledge and practical hardware understanding.

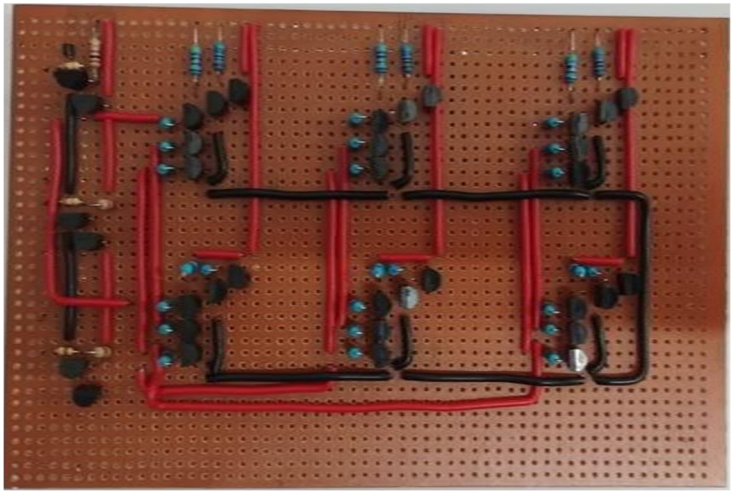
IV. BLOCK DIAGRAM



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

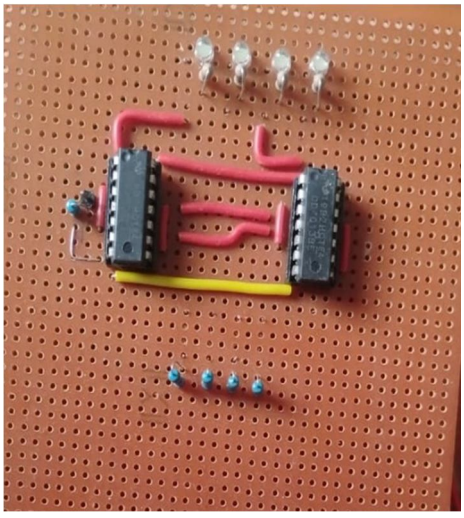
V. PARTS OF CPU

A. Instruction Decoder



The instruction decoder is implemented using BC547 bipolar junction transistors (BJTs). Each decoder unit consists of five transistors. Three of those transistors are configured to form a NAND gate. An additional transistor is used as an inverter (NOT gate) to convert the NAND output into an AND gate. The fifth transistor acts as a buffer stage to drive output.
The complete decoder has six such units for six different instructions. Input signals to the Instruction Decoder are taken from the ROM.
To generate complemented input signals, three pairs of NOT gates are used, each followed by buffer stage to get reliable output and to get better fan out.

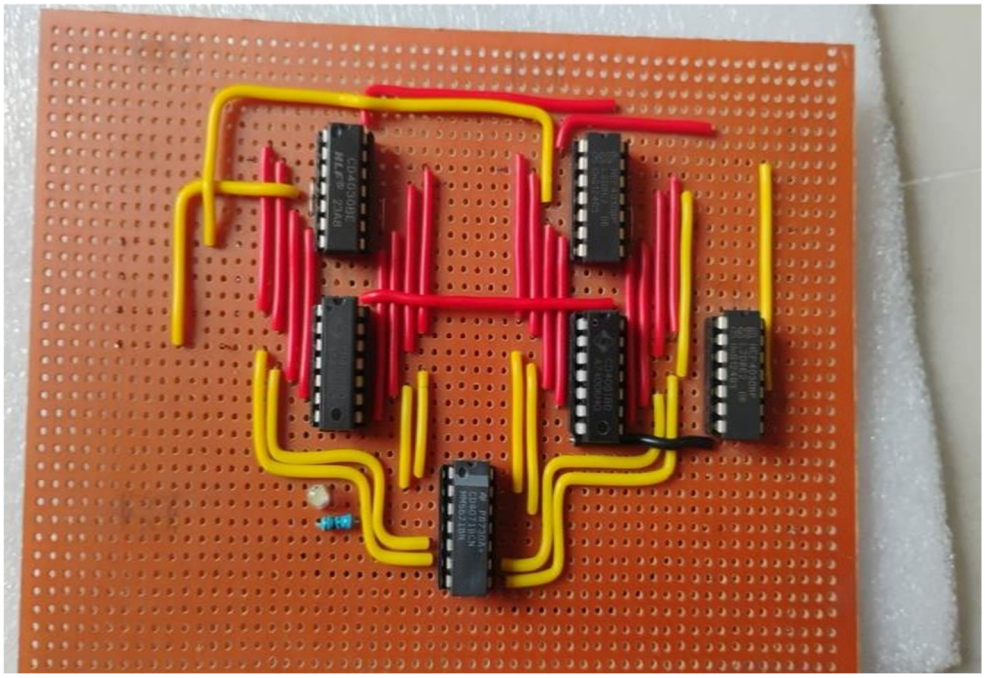
B. Program Counter



The Program Counter is made of two D FlipFlop ICs CD4013 working as a 4 Bit UP Asynchronous Counter. The Clock signal is given from the Clock Module. For Jump Instruction the PRESET and CLEAR pins of the flip flops area used. For JUMP instruction – during first cycle the flip flops are cleared and during the second cycle only the required bits are preset.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

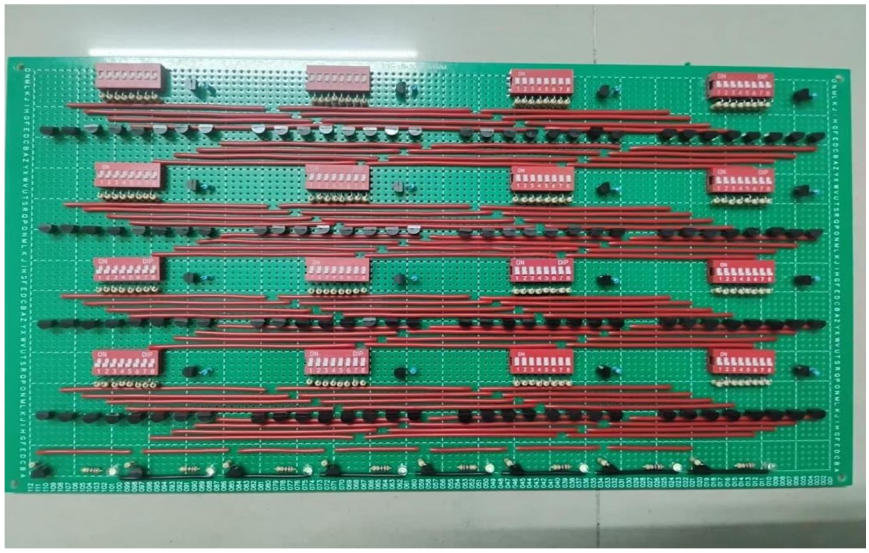
C. Arithmetic and Logic Unit (ALU)



The ALU is made using logic gate ICs like XOR, AND, and OR. The main circuit consists of a 4-bit Adder made by coupling 4 Full Adders made using the Logic Gates.

For subtraction, 2's complement method is used. For this, the B input is passed through XOR gates before going into the adder. A control signal from the Control Unit is used here. When the control signal is HIGH, the XOR gates invert the B input. The Control Signal is also used as 'carry-in' with the LSB to simulate the 'adding 1 to the LSB' of the 2's complement method. When the control signal is LOW, B is not changed and normal addition is performed.
So basically, the same circuit is used for both addition and subtraction by just changing the control signal. This reduces the number of components and keeps the design simple.

D. ROM



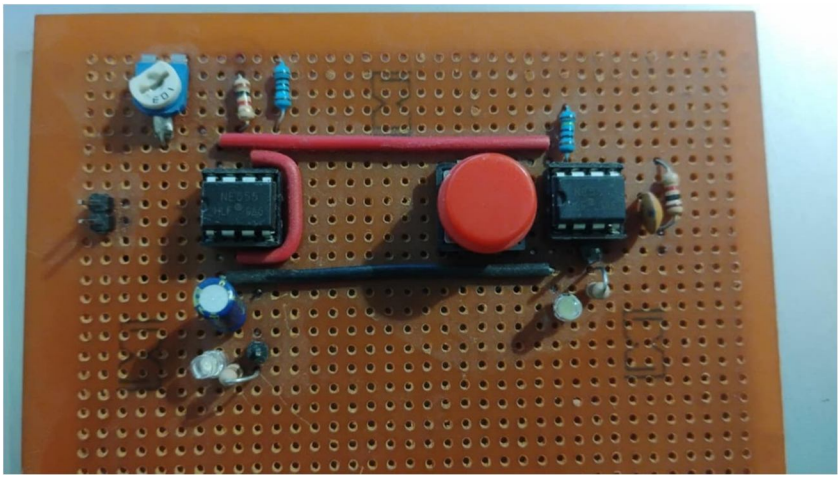
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

The ROM has a capacity of 16 bytes. Each byte is implemented using nine BC547 BJTs, consisting of eight data transistors and one enable transistor.

The collectors of the data transistors are connected to DIP switches. The position of each DIP switch determines whether the corresponding data transistor represents a logic '1' or '0'. The enable transistor controls the output of data transistors, its output is connected to the bases of all eight data transistors within that byte.

The emitters of corresponding data transistors from all bytes are connected together to form a common output bus, there are total of 8 such buses. To get proper signal driving, the output bus is connected to transistors working as buffers implemented using BJT emitter followers. These buffers provide stable outputs for the circuits that are to be used interfaced.

E. Clock



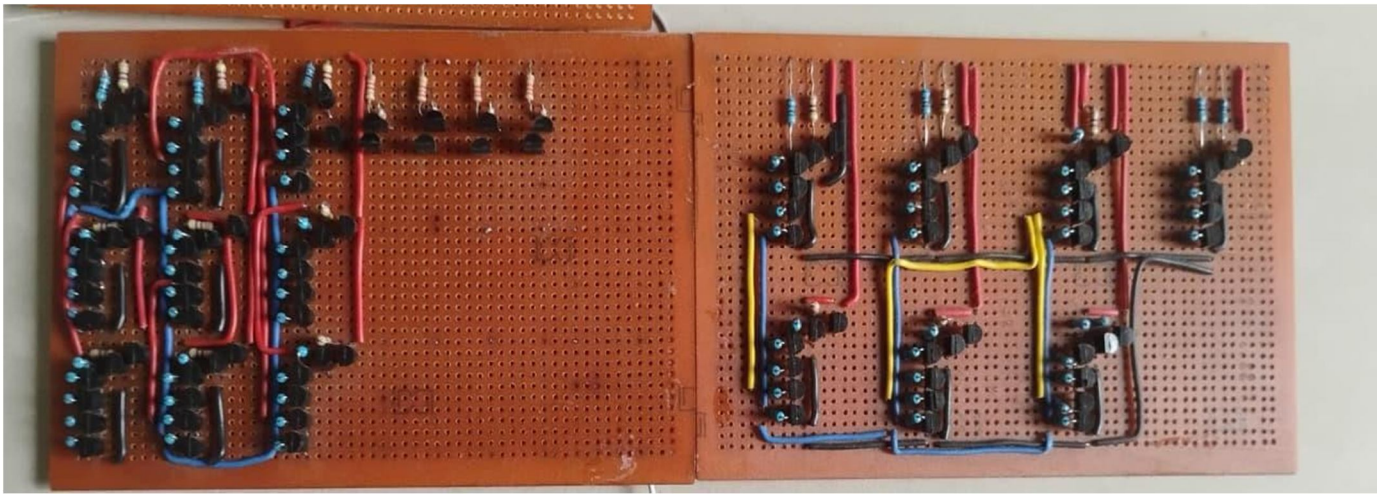
The clock module has two modes:

1. Manual
2. Automatic.

For Automatic mode a 555 Timer IC is used in Astable Mode working as a Voltage Controlled Oscillator to get clock signal of different frequencies, ranging from 1 kHz to 3.5 kHz.

For Manual mode a 555 Timer IC is used in Monostable Mode. The Timer IC is triggered using a simple Push Button. The 555 Timer IC is used to compensate debouncing of the Push Button and to prevent multiple output triggers.

F. ROM Decoder



©IJRASET: All Rights are Reserved

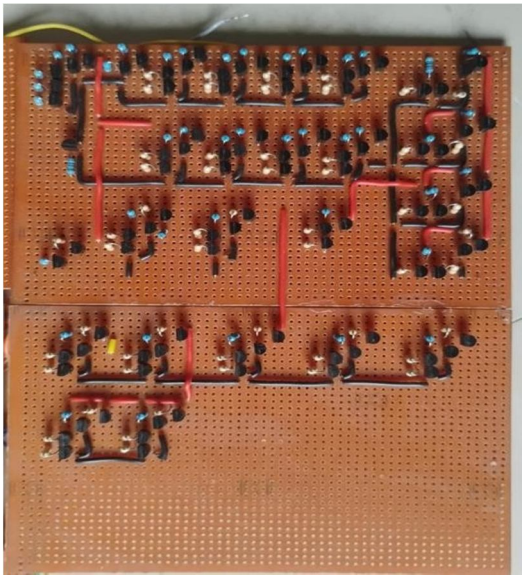
The ROM decoder is made using BC547 bipolar junction transistors (BJTs). Each decoder unit uses six transistors. Out of these, four transistors are used to form a NAND gate. One transistor is used as a NOT gate to convert the NAND output into an AND function, and the sixth transistor is used as a buffer to drive the output. The Active Components used are BC547 transistors and the passive components used are Resistors of values 47K Ω, 4.7K Ω and 470 Ω.

The inputs to each decoder unit are set depending on the address at which that unit should turn ON. For example, if a unit should activate at D=0, C=1, B=0, A=1 (0101), then the inputs to that unit are connected as A, NOT B, C, and NOT D. This ensures that the output of that unit becomes HIGH only for that specific input combination.

The decoder takes a 4-bit input from the Program Counter. Based on this input, only one decoder unit becomes active at a time.

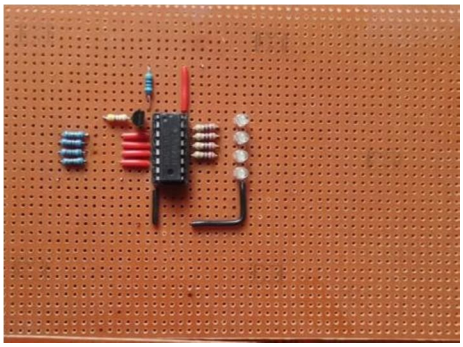
The ROM decoder is used to enable the specific ROM byte corresponding to the address provided by the Program Counter.

G. Control Unit



The Control Unit is implemented using BC547 transistors. Based on the inputs, this module generates output signals to control other circuits within the system. The Input is taken from the ROM and the Instruction Decoder. A simple one-bit Ring Counter is also implemented within the Control Unit to control the flow of execution.

H. Accumulator Register



The Accumulator Register is made using a 4 Bit Shift Register IC 74LS194. It stores and displays the output of operations performed by ALU. The output is displayed using four 3 mm LEDs. The Master Reset Pin (Pin 1) of the IC is controlled using ACLR (Accumulator Clear) signal from the Control Unit through a NOT gate made using BC547 as the Master Reset pin is Active LOW type. It also takes inputs directly from ROM using LDA (Load Accumulator) Instruction.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. OUTCOME

The successful design and implementation of the proposed 8-bit CPU using discrete Bipolar Junction Transistors (BJTs) and logic ICs resulted in a fully functional hardware- based computing system capable of executing basic instructions. The system demonstrated the complete fetch–decode–execute cycle, providing clear visibility into internal processor operations.

The developed CPU was able to perform fundamental arithmetic and logical operations such as addition, subtraction, data transfer, and conditional branching, validating the correctness of the ALU and control unit design. The use of a 555 timer-based clock generator enabled low-frequency operation, allowing real-time observation of instruction execution and signal transitions through LEDs.

The project successfully achieved the following outcomes:

1) Developed a working 8-bit CPU architecture using discrete electronic components
2) Demonstrated hardware-level instruction execution without using microcontrollers or processors
3) Provided clear visualization of data flow, control signals, and timing sequence
4) Enhanced understanding of digital electronics, computer organization, and transistor switching behavior
5) Created a low-cost and educationally effective model for academic demonstrations

The implementation confirms that complex computing operations can be achieved using simple components, thereby strengthening the conceptual foundation of processor design. This project serves as a practical bridge between theoretical knowledge and real-world hardware implementation

VII. CONCLUSION

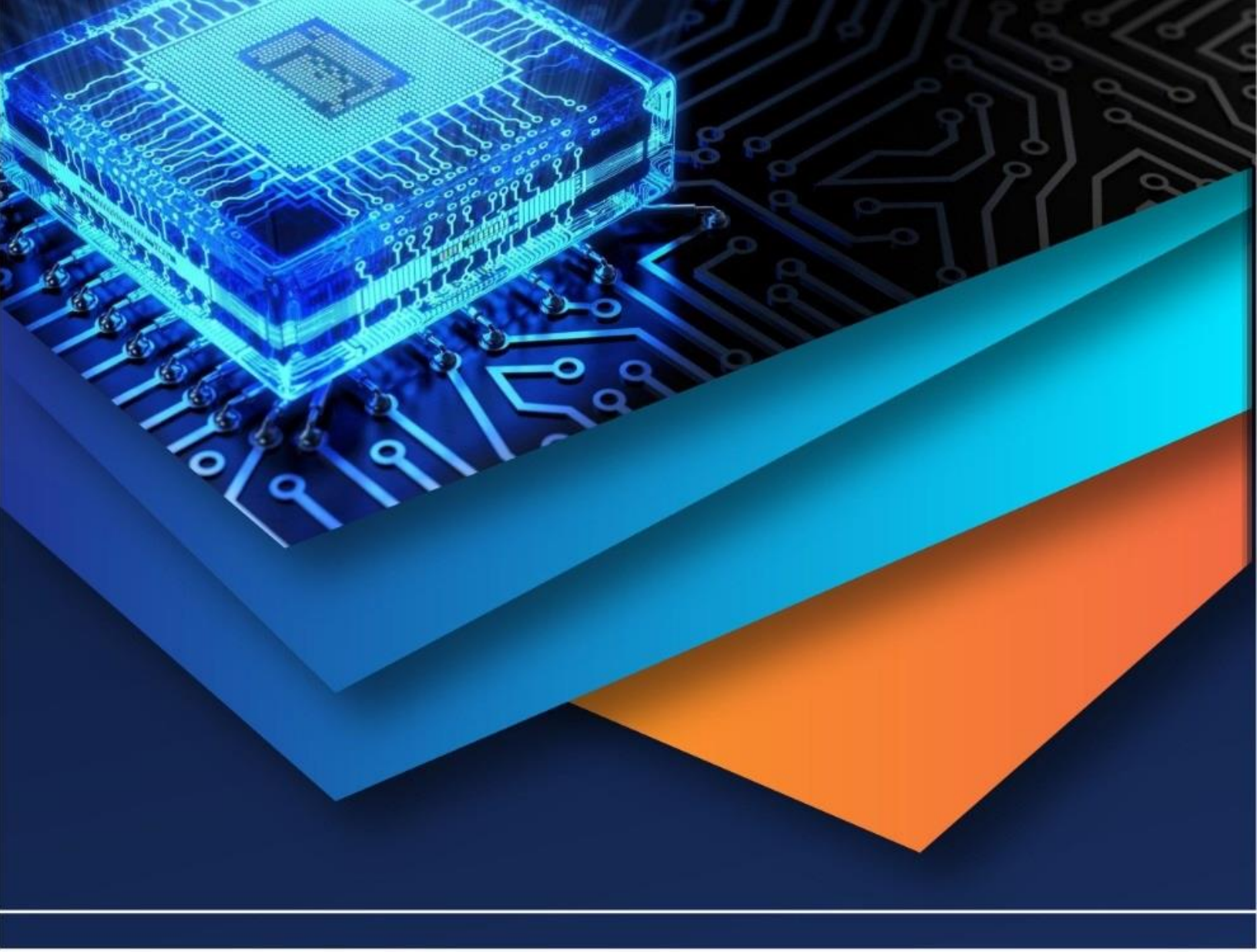
The proposed 8-bit CPU using discrete BJTs and logic ICs successfully demonstrates the fundamental working of a processor at the hardware level. The system effectively performs basic operations and follows the fetch–decode–execute cycle. It provides clear visualization of data flow and control signals, enhancing understanding of digital electronics and computer architecture. This project serves as a low-cost and effective educational model, bridging the gap between theory and practical implementation.

VIII. ACKNOWLEDGEMENT

I take this opportunity to express my sincere appreciation for the cooperation given by Prof. S. G. Giram, Principal of AISSMS's POLYTECHNIC, Pune and need a special mention for all the help extended by him, constant inspiration and encouragement to make my project a memorable experience. I am thankful to our H. O. D of ENTC Department, Prof. V. S. Gaikwad for her time to time support and valuable guidance. I am deeply indebted to my internal guide Mrs. J. R. Khandare, for completion of this project for which she guided and helped me going out of the way… I am thankful to all the teachers and professors of our department for sharing with me, valuable knowledge no their respective fields. I would also thank my fellow classmates and friends for their support and timely suggestions. I would also like to thanks library staff and laboratory staff for providing me cordial support and necessary facilities, which were of great help for preparing the project report. Thanks to all!!!

REFERENCES

[1] Global Science Network, "How to Build a 4-Bit Computer on Breadboards Using Individual Transistors," YouTube video. Available: https://youtu.be/_eo8l7HP-9U
[2] FULXOR Dominic Boeuf, "How I Built a CPU with Minimal Transistors: Fully Exploring Computer Logic," YouTube video. Available: https://youtu.be/xupra9UPzQ0
[3] Big Chaks, "I Built a Computer From Scratch!," YouTube video. Available: https://youtu.be/vgeDAoJcGaM
[4] Vocademy, "What is Saturation?," YouTube video. Available: https://youtu.be/rkVDwUaXQ4Y



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)