8-bit CPU Design from Scratch

Initial Design Document

Pranav Verma

May 17, 2025

Contents

1 Introduction

This document outlines the initial design approach for constructing an 8-bit CPU from fundamental components. The project aims to build a fully functional computer system using discrete logic ICs, providing deep insights into computer architecture principles through hands-on experience.

The CPU design follows a modular approach with several key components:

- Adjustable clock module
- General-purpose registers
- Arithmetic Logic Unit (ALU)
- Random Access Memory (RAM)
- Program Counter
- Output display system
- Control logic unit

Each module will be constructed and tested independently before final integration, allowing for systematic troubleshooting and verification of functionality.

2 Components List

Below is a comprehensive list of components required for this project, with prices converted to Indian Rupees (INR). The components are readily available from Indian electronics suppliers or can be ordered online.

** Note: The 74LS107 is functionally equivalent to the 74LS76 but has a different pinout. Please refer to the datasheets for proper connections.*

3 Module Designs

3.1 Clock Module

The clock module serves as the central synchronization mechanism for the entire CPU. It generates precisely timed pulses that coordinate all operations within the system.

3.1.1 Design Features

- Variable frequency operation from sub-1Hz to several hundred Hz
- Manual clock mode for step-by-step debugging
- Visual LED indicator for clock state
- Built around the versatile 555 timer IC

Table 1: Component List with Prices (INR)

Component	**Unit Price**	**Total Price**	**Source**
1 × 1kΩ resistor	5 INR	5 INR	Local store
9 × 10kΩ resistor	5 INR	45 INR	Local store
1 × 100kΩ resistor	5 INR	5 INR	Local store
24 × 470Ω resistor	5 INR	120 INR	Local store
1 × 1MΩ resistor	5 INR	5 INR	Local store
1 × 1MΩ potentiometer	80 INR	80 INR	Local store
6 × 0.01µF capacitor	8 INR	48 INR	Local store
16 × 0.1µF capacitor	10 INR	160 INR	Local store
1 × 1µF capacitor	10 INR	10 INR	Local store
4 × 555 timer IC	20 INR	80 INR	Local store
2 × 74LS00 (Quad NAND)	50 INR	100 INR	Local store
1 × 74LS02 (Quad NOR)	40 INR	40 INR	Local store
5 × 74LS04 (Hex inverter)	40 INR	200 INR	Local store
3 × 74LS08 (Quad AND)	45 INR	135 INR	Local store
1 × 74LS32 (Quad OR)	35 INR	35 INR	Local store
1 × 74LS107 (Dual JK FF)*	120 INR	120 INR	Online
2 × 74LS86 (Quad XOR)	40 INR	80 INR	Local store
1 × 74LS138 (Decoder)	45 INR	45 INR	Local store
1 × 74LS139 (Decoder)	45 INR	45 INR	Local store
4 × 74LS157 (Multiplexer)	45 INR	180 INR	Local store
2 × 74LS161 (Counter)	50 INR	100 INR	Local store
8 × 74LS173 (Register)	85 INR	680 INR	Local store
2 × 74189 (64-bit RAM)	350 INR	700 INR	Online
6 × 74LS245 (Bus transceiver)	50 INR	300 INR	Local store
1 × 74LS273 (D flip-flop)	50 INR	50 INR	Local store
2 × 74LS283 (Binary adder)	90 INR	180 INR	Local store
3 × 28C16 EEPROM	250 INR	750 INR	Online
3 × Toggle switch	60 INR	180 INR	Local store
3 × Tact switch	25 INR	75 INR	Local store
1 × 8-position DIP switch	55 INR	55 INR	Local store
1 × 4-position DIP switch	60 INR	60 INR	Local store
44 × Red LED	8 INR	352 INR	Local store
8 × Yellow LED	6 INR	48 INR	Local store
12 × Green LED	7 INR	84 INR	Local store
21 × Blue LED	40 INR	840 INR	Local store
4 × 7-segment display	75 INR	300 INR	Local store

3.1.2 Implementation Approach

The 555 timer will be configured in astable multivibrator mode for automatic clock generation. The addition of a 1MΩ potentiometer allows for precise frequency adjustment. A switch will toggle between automatic and manual modes, where a momentary push button can advance the clock one cycle at a time.

This module is critical for system debugging as it allows the observation of CPU operations at human-readable speeds or single-step execution to verify proper functioning of each component.

3.2 Register System

Registers form the CPU's immediate working memory, storing data actively being processed. Our design includes three 8-bit registers:

3.2.1 A and B Registers

These are general-purpose registers for storing operands during arithmetic and logical operations. They will be constructed using 74LS173 4-bit D-type registers, with two ICs per 8-bit register.

3.2.2 Instruction Register (IR)

The IR holds the current instruction being executed by the CPU. It functions similarly to the general-purpose registers but serves a specialized role in the instruction execution cycle.

Each register will include:

- 8 LEDs for visual display of current value
- Clock synchronization
- Bus interface for data transfer
- Load and output enable control lines

3.3 Arithmetic Logic Unit (ALU)

The ALU performs mathematical and logical operations on data. While commercial CPUs include numerous operations, our simplified ALU focuses on:

- 8-bit binary addition (A + B)
- 8-bit binary subtraction (A - B)
- Status flags for zero and carry conditions

The ALU implementation uses two 74LS283 4-bit binary adders chained together, with additional logic for subtraction via two's complement. An operation selector will determine whether addition or subtraction is performed.

3.4 Random Access Memory (RAM)

The RAM module provides temporary storage for both program instructions and data. Our design features:

- 16 bytes of memory (4-bit addressing)
- Read and write capability
- Visual display of memory contents
- Address selection via the program counter or manual override

While 16 bytes is very limited, it's sufficient to demonstrate key CPU concepts through simple programs. The RAM will be implemented using 74189 64-bit RAM ICs.

3.5 Program Counter

The program counter tracks which instruction is currently being executed. Key features include:

- 4-bit counter for addressing 16 memory locations
- Automatic increment with each instruction cycle
- Jump capability for program flow control
- Reset function

The module will be built using 74LS161 4-bit synchronous binary counters with additional logic for jumps and reset operations.

3.6 Output System

The output system provides visual feedback of CPU operation through:

- 8-bit binary LED display (primary output)
- 7-segment decimal display for human-readable output
- Output register for stable display independent of internal operations

This module uses a 74LS273 octal D flip-flop as the output register, with additional decoding logic to convert binary values to 7-segment display format.

3.7 Control Logic

The control logic is the CPU's command center, coordinating all modules based on the current instruction. This complex module:

- Decodes instructions from the instruction register
- Generates appropriate control signals for all modules
- Manages the instruction cycle (fetch, decode, execute)
- Implements the CPU's instruction set

The control logic will be implemented using a combination of discrete logic and programmable EEPROMs to store microcode sequences. This approach simplifies the hardware while maintaining flexibility in the instruction set design.

4 System Integration

The modular design approach allows for independent testing of each component before final integration. The central element that connects all modules is an 8-bit data bus, implemented using:

- Common data pathways between all modules
- Bus transceivers (74LS245) for controlled access
- Control signals to manage bus access and prevent conflicts

The integration process will follow this sequence:

1. Connect clock to all synchronous components
2. Establish bus connections between registers and ALU
3. Add RAM and program counter to the bus system
4. Integrate output display
5. Implement control logic and connect control lines
6. Verify complete system operation

5 Instruction Set

The CPU will support a minimal but functional instruction set including:

- Data movement (load, store)
- Arithmetic (add, subtract)
- Conditional and unconditional jumps
- Input/output operations

Each instruction will be encoded in 8 bits, with the upper 4 bits defining the operation and the lower 4 bits typically used for memory addressing or immediate values.

6 Testing and Validation

Each module will be thoroughly tested before integration:

- Clock module: Verify stable operation at various frequencies and in manual mode
- Registers: Test data storage and retrieval functionality
- ALU: Validate correct computation of sums and differences
- RAM: Verify read/write operations across all addresses
- Program counter: Test increment, reset, and jump operations
- Output system: Confirm correct display of binary and decimal values

System integration testing will involve running simple programs that exercise all aspects of the CPU, gradually increasing in complexity.

7 Future Enhancements

Potential extensions to the basic design include:

- Expanded memory capacity (8-bit addressing for 256 bytes)
- Additional ALU operations (logical AND, OR, XOR, shift, etc.)
- Stack implementation for subroutine support
- Hardware interrupts
- Interface to external devices

8 Conclusion

This 8-bit CPU project provides an excellent platform for understanding fundamental computer architecture principles through hands-on building and experimentation. Despite its simplicity, the design incorporates all essential elements of a functional CPU: clock, registers, ALU, memory, and control logic.

The modular approach facilitates incremental building and testing, making the project manageable while providing clear visibility into how each component contributes to the overall system. Upon completion, this breadboard computer will be capable of executing simple programs, demonstrating the fundamental principles that underlie all computing systems.

9 Bibliography

- https://eater.net/8bit