8-bit CPU Processor Simulation

NAME:

JAGAADHEP UK [CH.EN.U4CSE22120]

JAYANTHRAGAVAN M [CH.EN.U4CSE22121]

KAVINKUMAR VS [CH.EN.U4CSE22125]

MANISHANKAR SG [CH.EN.U4CSE22130]

Introduction

This project presents the design and simulation of a simple 8-bit CPU, providing an educational tool to better understand fundamental computer architecture. The processor supports a minimalist instruction set, enabling it to execute essential arithmetic and logical operations. By simulating processor behavior, it demonstrates the core concepts of hardware-software interactions and the execution of low-level operations. This project is especially valuable for those studying digital systems, CPU design, and assembly programming.

Key Features

The 8-bit CPU comprises several key components, each playing a pivotal role in processing data and executing instructions. These components include:

- **A Register:** A general-purpose register.
- **B Register:** Another general-purpose register.
- **Memory Address Register (MAR):** Stores the address of the memory location to be accessed.
- **Memory Buffer Register (MBR):** Temporarily holds data being transferred between memory and other components.
- **Instruction Register (IR):** Holds the current instruction being executed.
- **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical operations.
- **Program Counter (PC):** Keeps track of the address of the next instruction to be executed.

Instruction Set Architecture (ISA)

The CPU supports the following instruction set:
- LOADA (0000): Loads data into register A.
- LOADB (0001): Loads data into register B.
- ADD (0010): Adds the contents of registers A and B.
- STORE (0011): Stores the result back into memory.

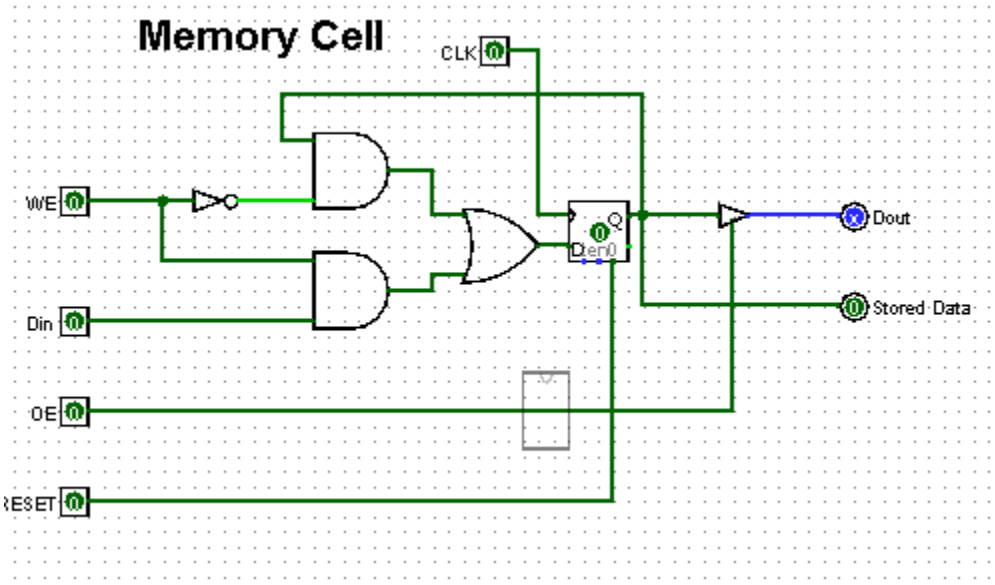
Each instruction is represented by a 4-bit opcode, and in most cases, a 4-bit operand specifying a memory address or data location.

Processor Architecture

Memory Cells

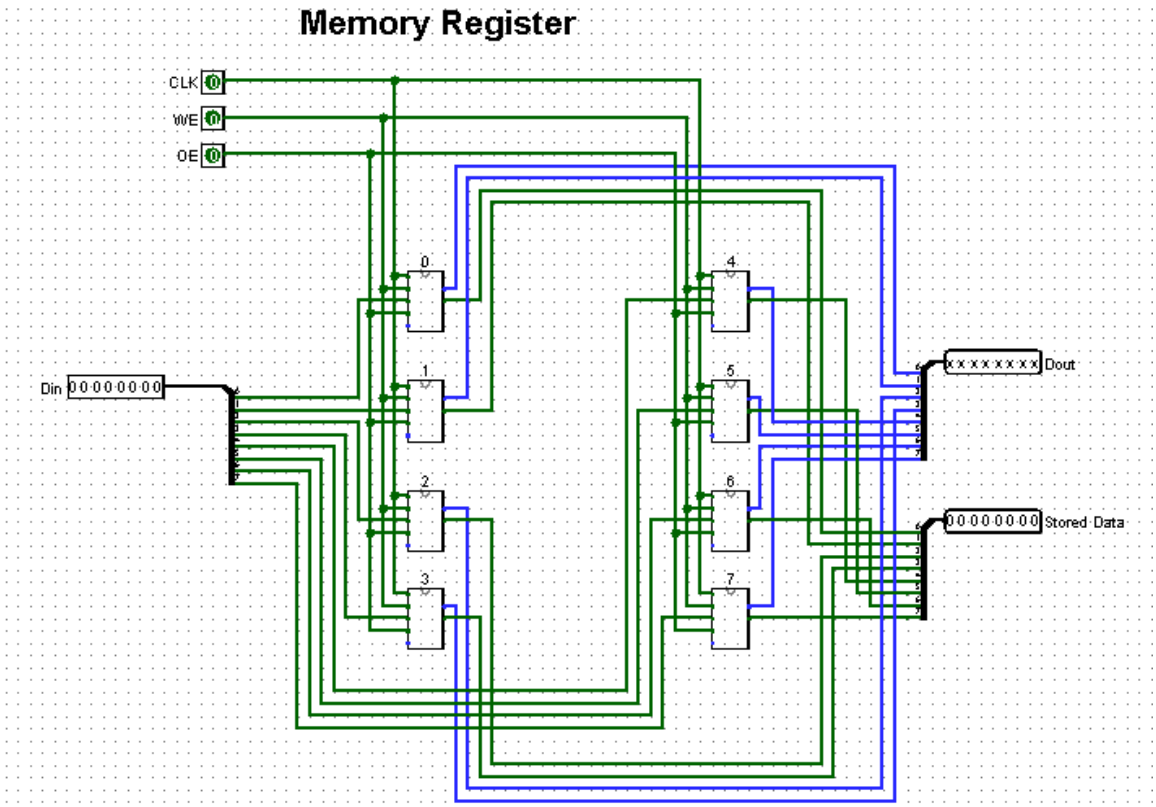
Definition: A memory cell is the fundamental building block of a computer's memory, capable of storing a single bit of information.

Working Mechanism: Data can be written into or read from the memory cell. A write-enable (WE) signal controls data storage, while output-enable (OE) allows data reading. Clock signals (CLK) synchronize operations, and reset signals clear the data.



Memory Registers

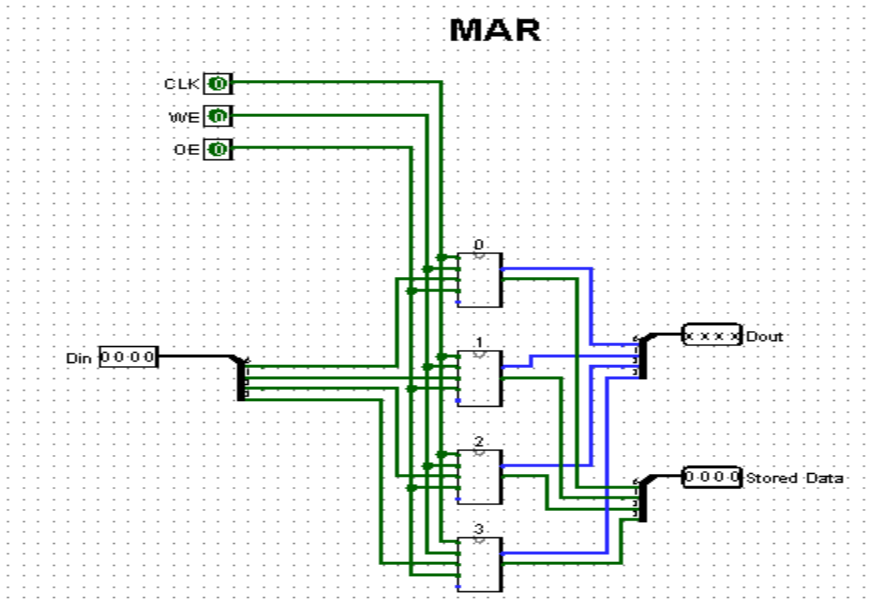
Definition: Memory registers consist of multiple memory cells, used to store larger units of data, such as bytes. Working Mechanism: These registers allow data to be temporarily stored and managed for processing, with control signals (WE, OE) managing input/output and CLK ensuring synchronization.



Memory Address Register (MAR)

Purpose: MAR holds the address of the memory location that is to be accessed for read/write operations.

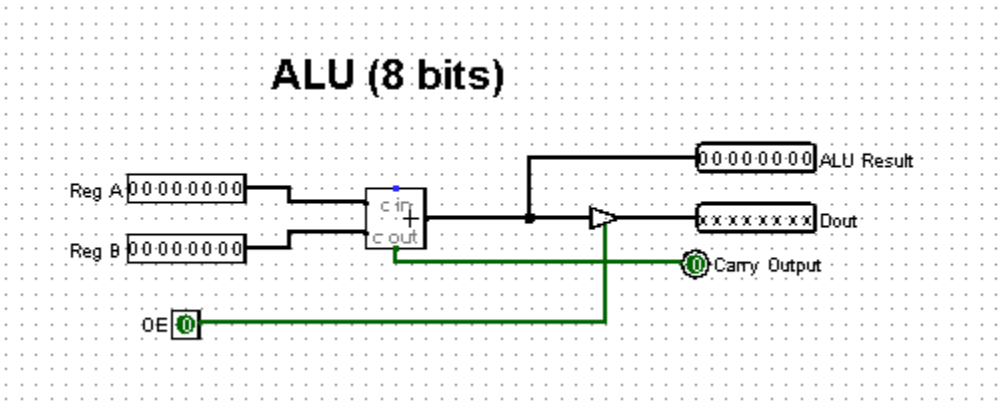
Working Mechanism: Similar to other registers, MAR uses WE and OE to control data flow and CLK to synchronize operations.



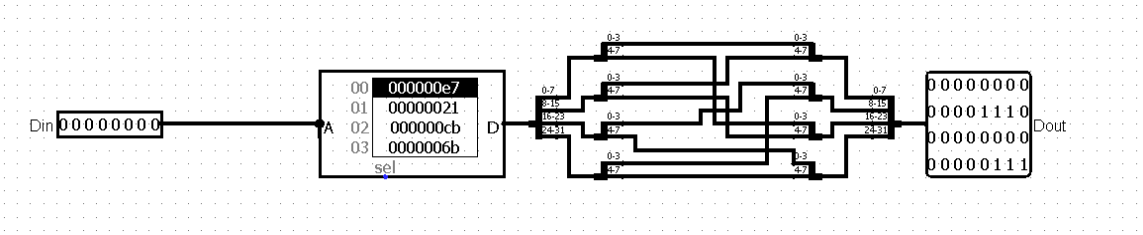
Arithmetic Logic Unit (ALU)

Purpose: The ALU is responsible for carrying out arithmetic and logical operations such as addition, subtraction, and comparisons.

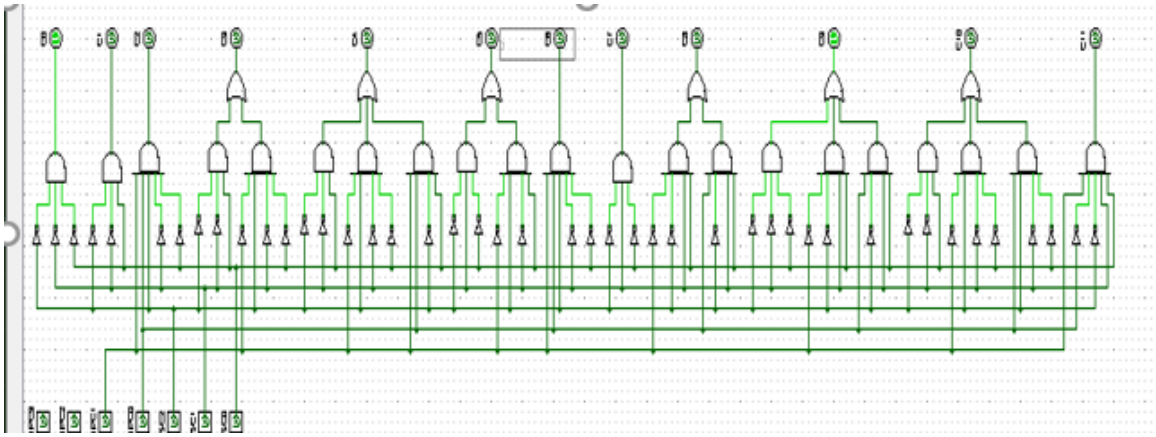
Working Mechanism: It processes two 8-bit inputs from the A and B registers, performs the required operation, and outputs the result. Carry-in (C_in) and carry-out (C_out) signals handle multi-bit operations.



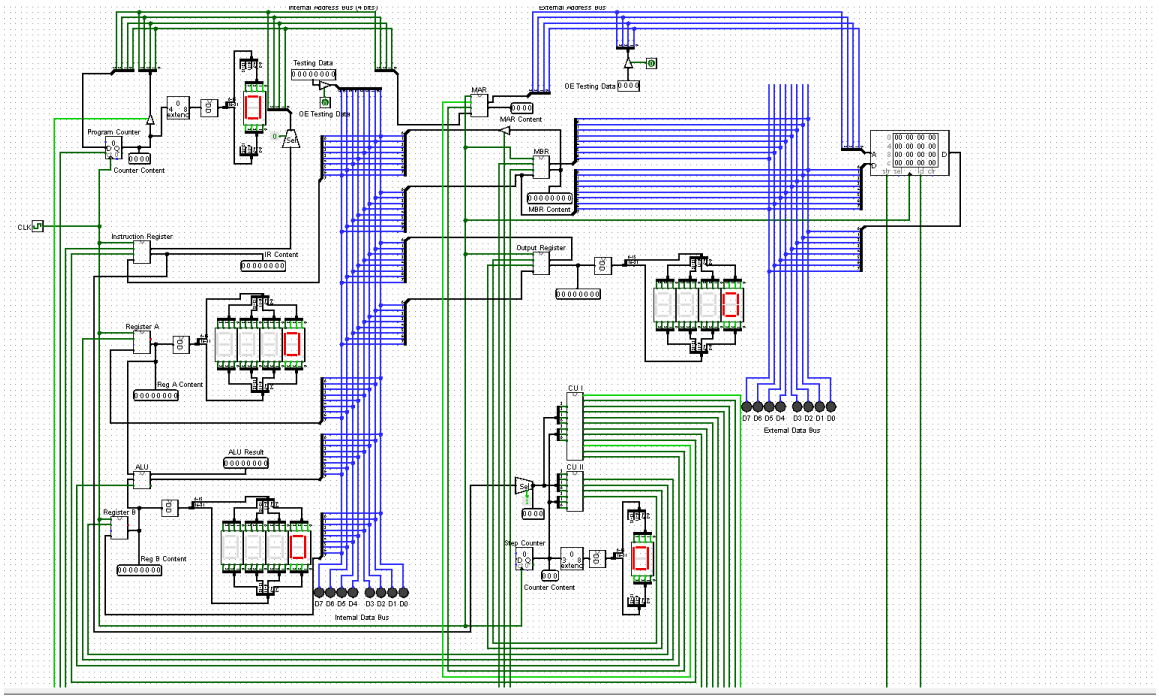
Decimal Decoder:



Control Unit:



Main Diagram:



Conclusion

This project provides a clear and practical approach to understanding basic CPU architecture through the design of an 8-bit processor. It highlights the essential components like the ALU, registers, and memory units that form the foundation of computer systems. By simulating processor operations, users can gain insights into how instructions are executed at a low level, helping bridge the gap between theoretical knowledge and practical application in digital systems and CPU design. The simplicity of the processor makes it an excellent educational tool for students and learners in computer engineering.