National University of Sciences and Technology

School of Electrical Engineering and Computer Science (SEECS)

ASSIGNMENT NO.1

MICROPROCESSOR SYSTEMS

SUBMITTED BY:

ABDULLAH AYAZ

M.ISMAEL BUTT

SYED M. MUSLIM

SALMAN ALI

WALEED AMMIR

SUBMITTED TO:

PROFESSOR IMRAN ABEEL

4-BIT MICROPROCESSOR

For this assignment we were given the task of making a 4-bit microprocessor that should be able to comprehensively execute a few instructions of our own choice and provide correct answers. For the implementation of instructions, we were required to make all the components of the microprocessor needed to execute the instructions on software of our choice. So, we chose **DIGITAL LOGIC SIM** for making all these components.

COMPONENTS:

- Program Counter
- Instruction Decoder
- ROM
- CLU
- 4-bit ALU
- 4- byte memory
- Enabling Buses
- Clock and clock selector
- Seven Segment display
- Accumulator Register

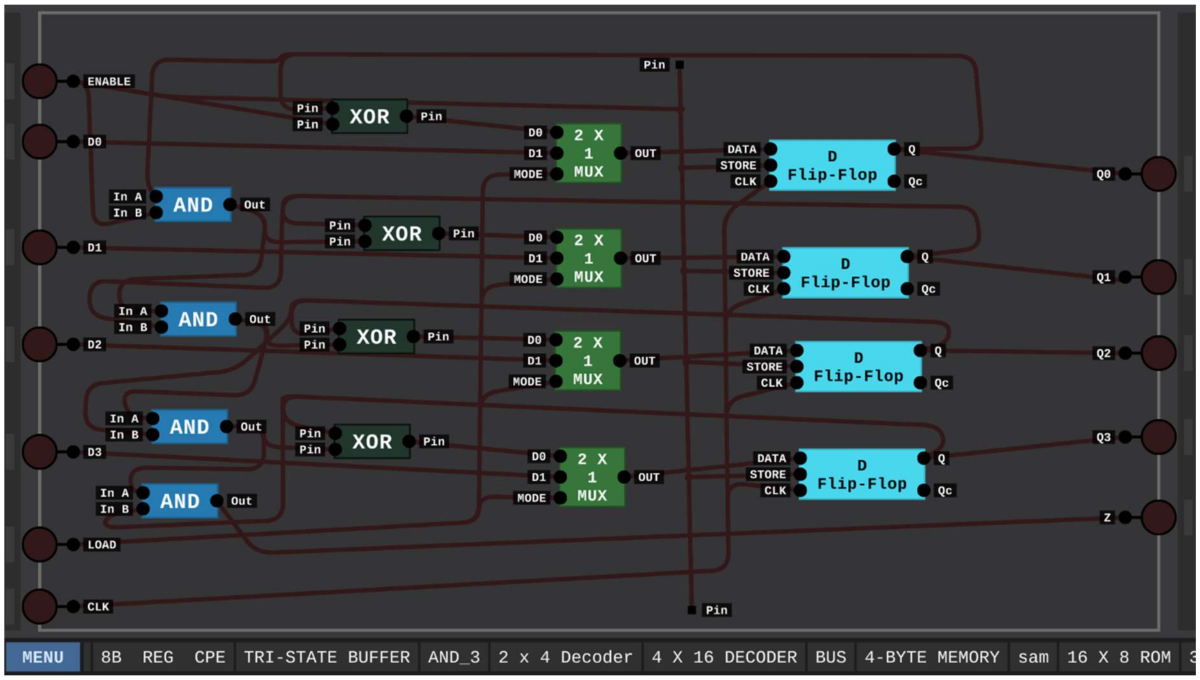
ROM:

The ROM is a **permanent storage memory** that cannot be changed by the user.

- Data stored in ROM is not lost.
- It comprises of registers to hold the data.

PROGRAM COUNTER:

- The program counter holds the address of the next instruction to be executed by the microprocessor.
- It requires cascaded AND gates, XOR gates and a few D-flipflops to hold the address.



INSTRUCTION DECODER:

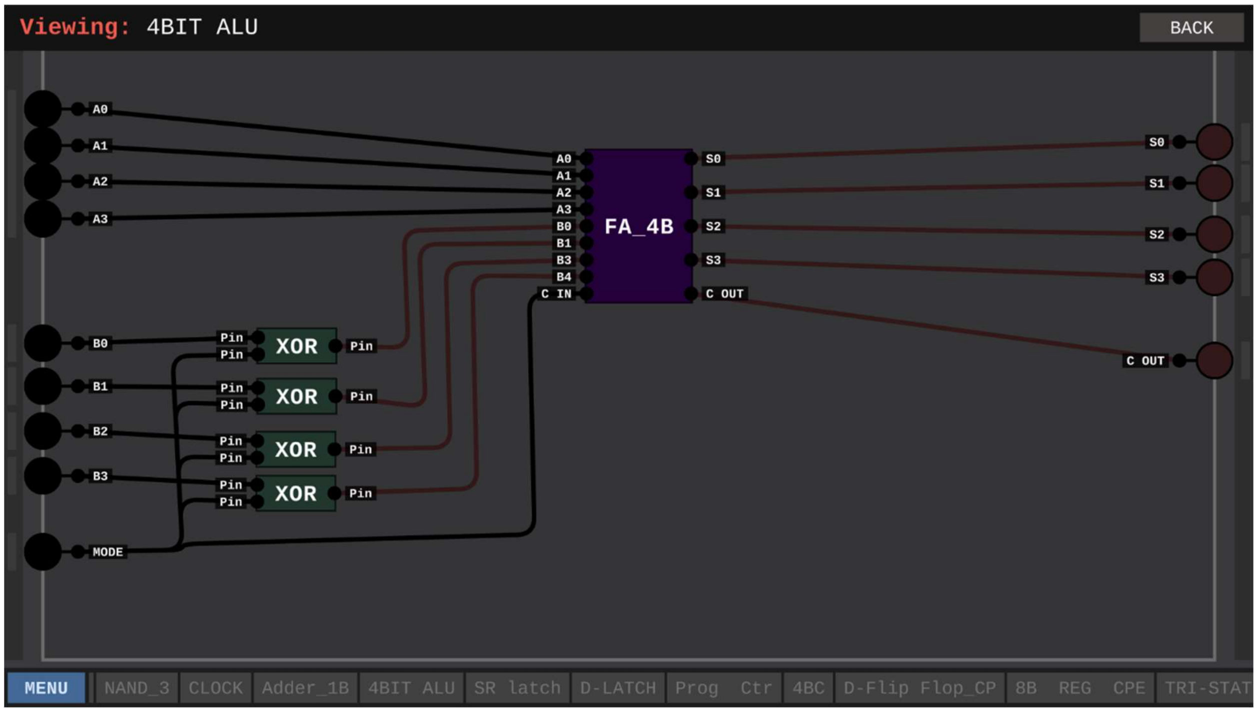
- Once the address comes into the program counter, it sends it to the instruction decoder.
- The instruction decoder decodes the address and identifies the opcode of the instruction and the operand on which the operation is to be performed. Then it sends it to the ALU.

4-BIT ALU:

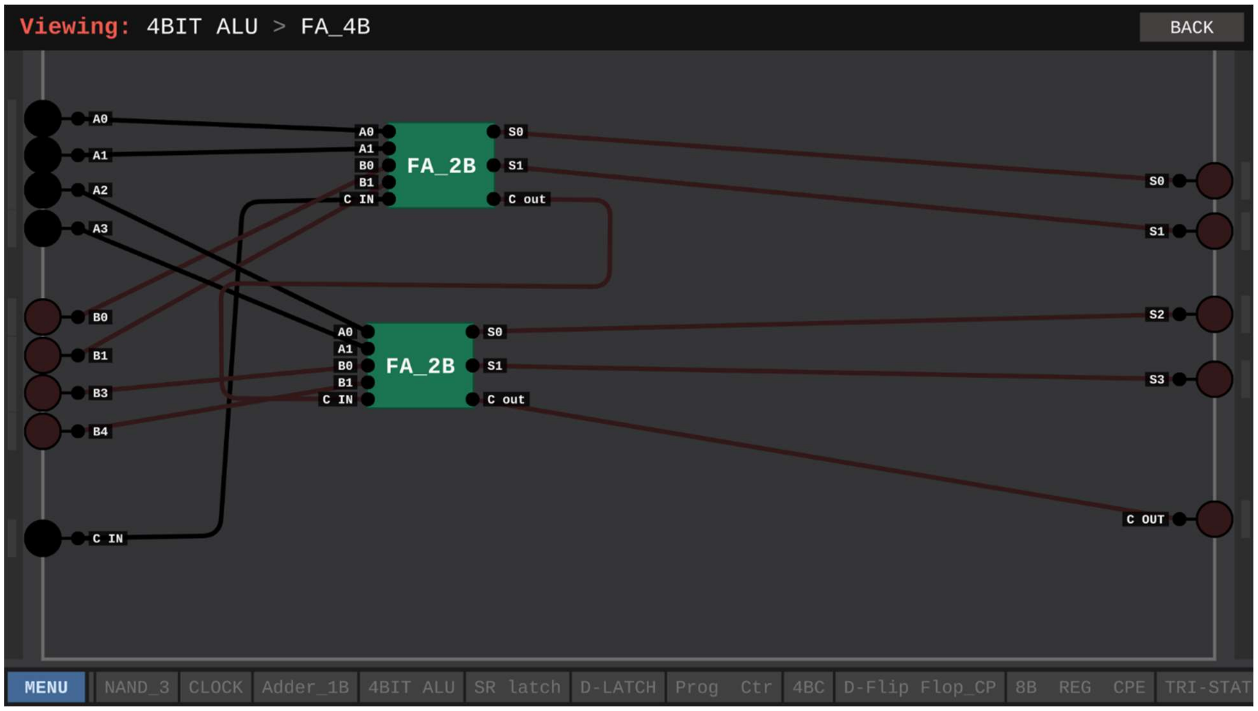
Our microprocessor requires a 4-bit ALU **(Arithmetic Logic Unit)** to execute the logical instructions.

- The ALU comprises of a 4-bit full adder and several XOR gates to select the mode so that we can perform both addition and subtraction using the same logic.
- This part of our microprocessor applies the knowledge of DLD **(Digital Logic Design)**. The ALU has four Full Adders and each of them has two Half-Adders to perform the required arithmetic.

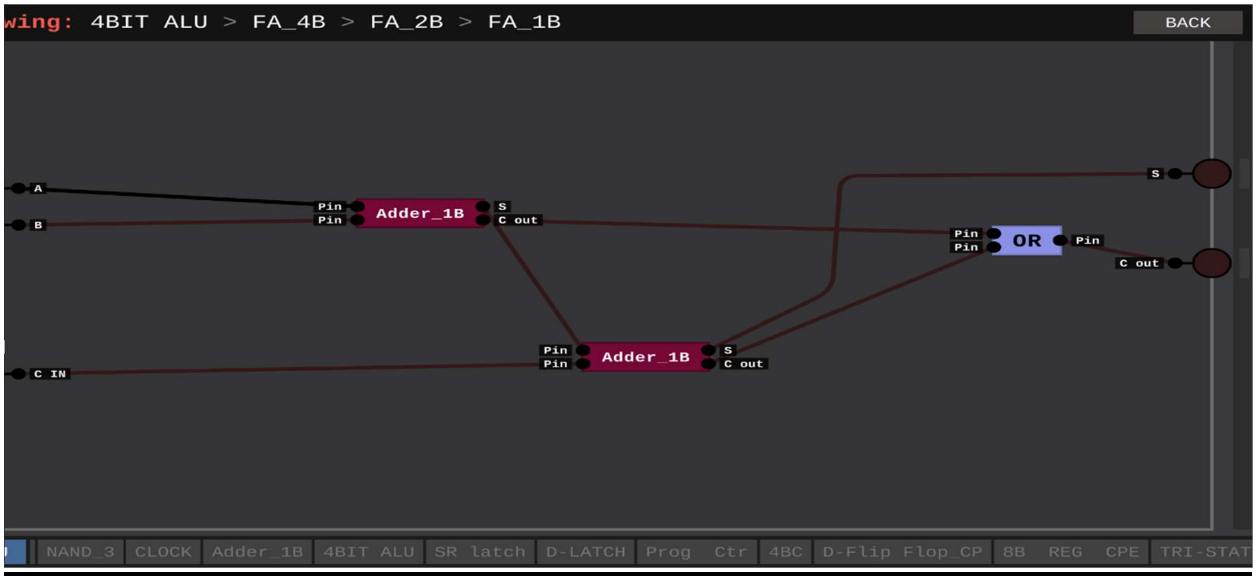
ALU:



FULL ADDERS:

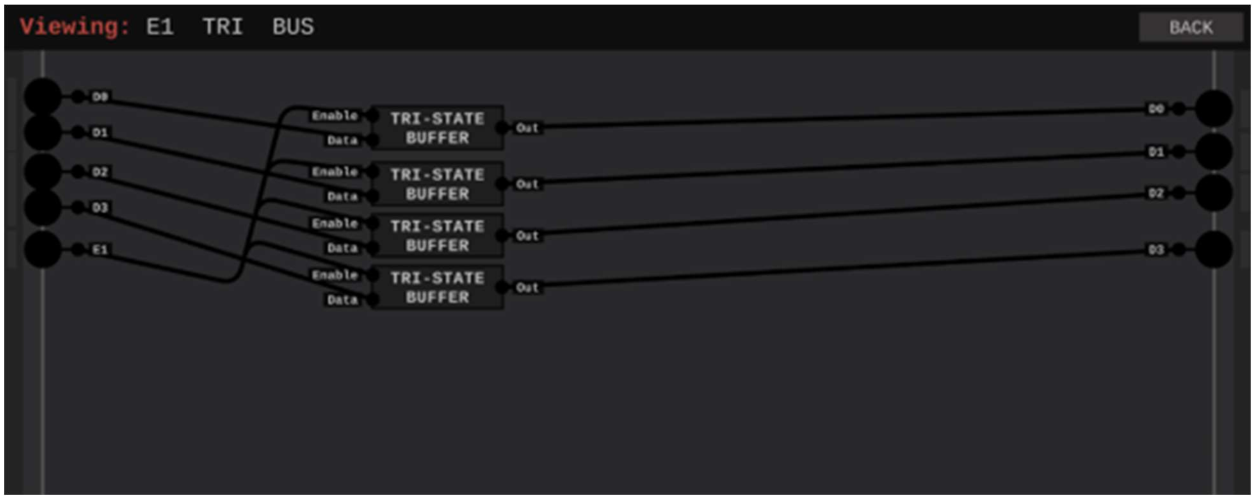


HALF-ADDERS:



ENABLING BUSES:

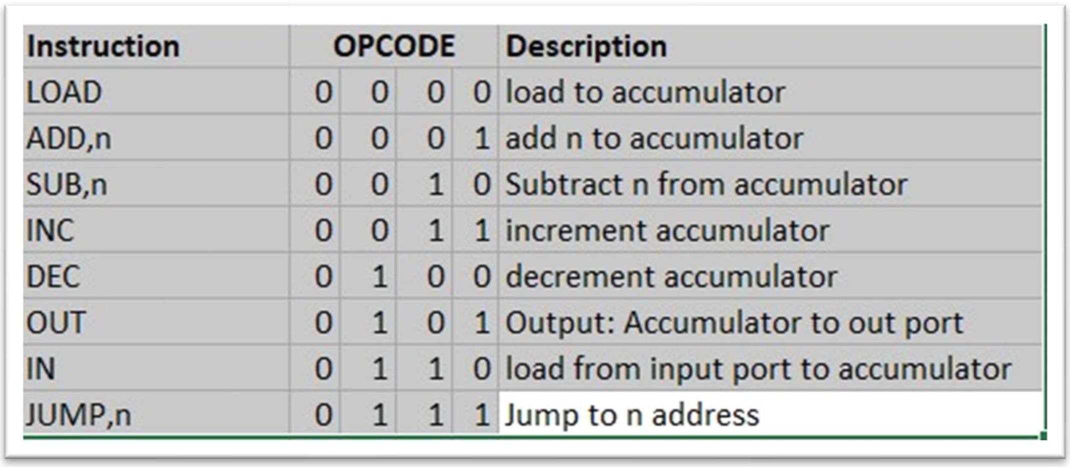
- These hold a prime importance in data transmission.
- The Enabling buses open and close path for the data to flow Obviously, under the influence of CLU.
- Each enabling bus comprises of 4-tri state buffers as the data to be transmitted is 4-bit.
- Our microprocessor has almost 9 enabling paths, each of this looks like the following:



INSTRUCTIONS:

Using our microprocessor, we are implementing the following instructions:

- LOAD.
- ADD, n.
- SUB, n.
- INC
- DEC
- OUT
- IN
- JUMP, n
- Following are the opcodes and description of the instructions:



Instruction	OPCODE				Description
LOAD	0	0	0	0	load to accumulator
ADD,n	0	0	0	1	add n to accumulator
SUB,n	0	0	1	0	Subtract n from accumulator
INC	0	0	1	1	increment accumulator
DEC	0	1	0	0	decrement accumulator
OUT	0	1	0	1	Output: Accumulator to out port
IN	0	1	1	0	load from input port to accumulator
JUMP,n	0	1	1	1	Jump to n address

LOAD INSTRUCTION:

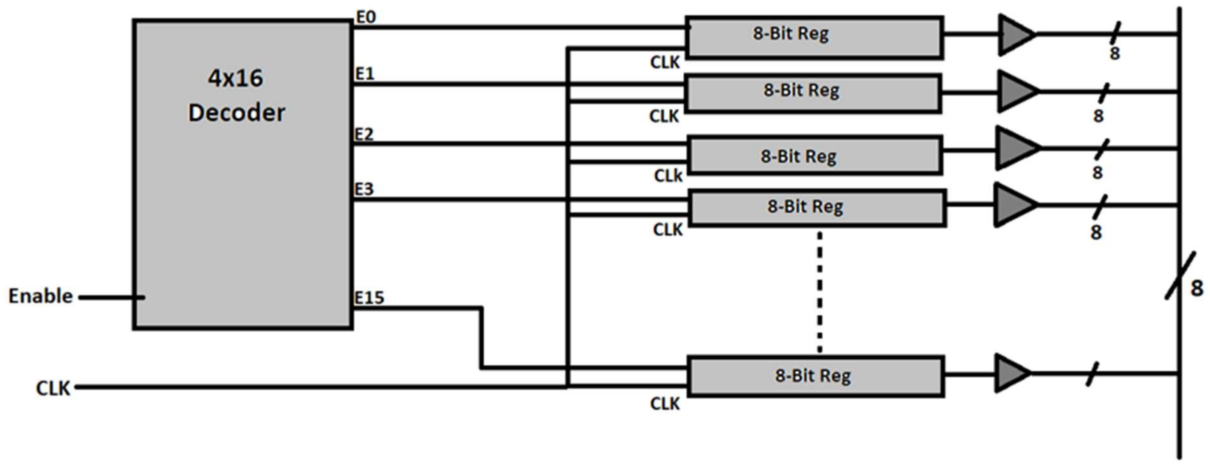
- The load instruction allows the data to be transferred to and from the accumulator.
- It requires an E3-tri bus, an AND gate, NOT gate and 4 D-flipflops to make a register.

IN INSTRUCTION:

- The IN instruction allows us to transfer data from the in-port to the accumulator register.
- With each passing clock, each enabling path opens and data is transferred to the destination.

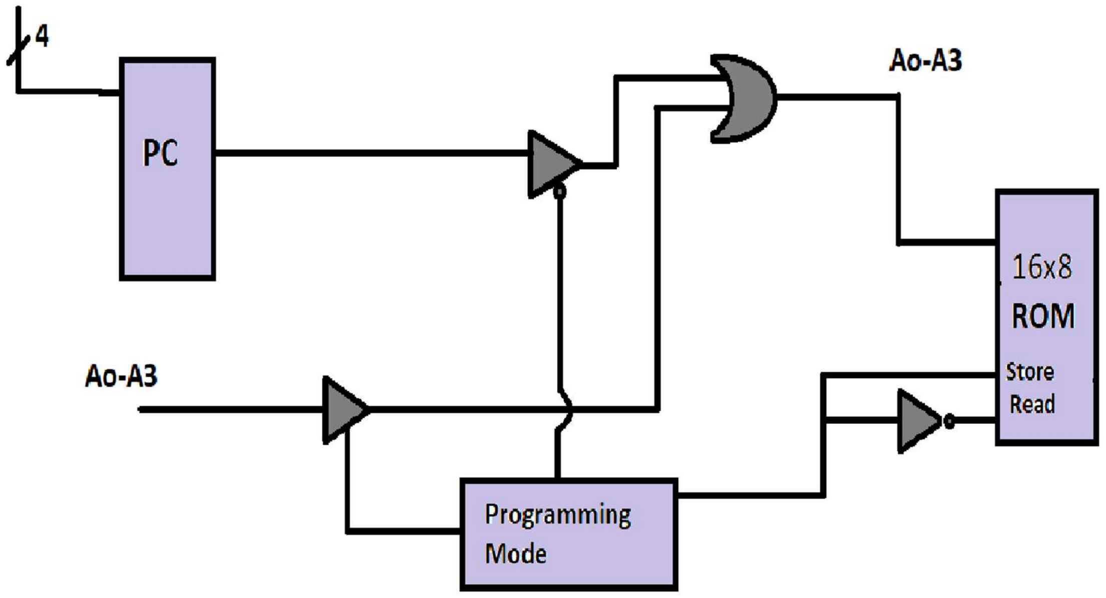
ROM 16x4 Architecture:

ROM Is made by the following structure as shown below:



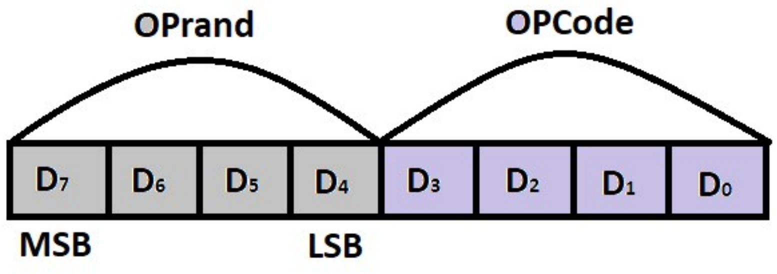
MODE Selector Architecture:

The Structure of the mode selector of the microcontroller is as follows:



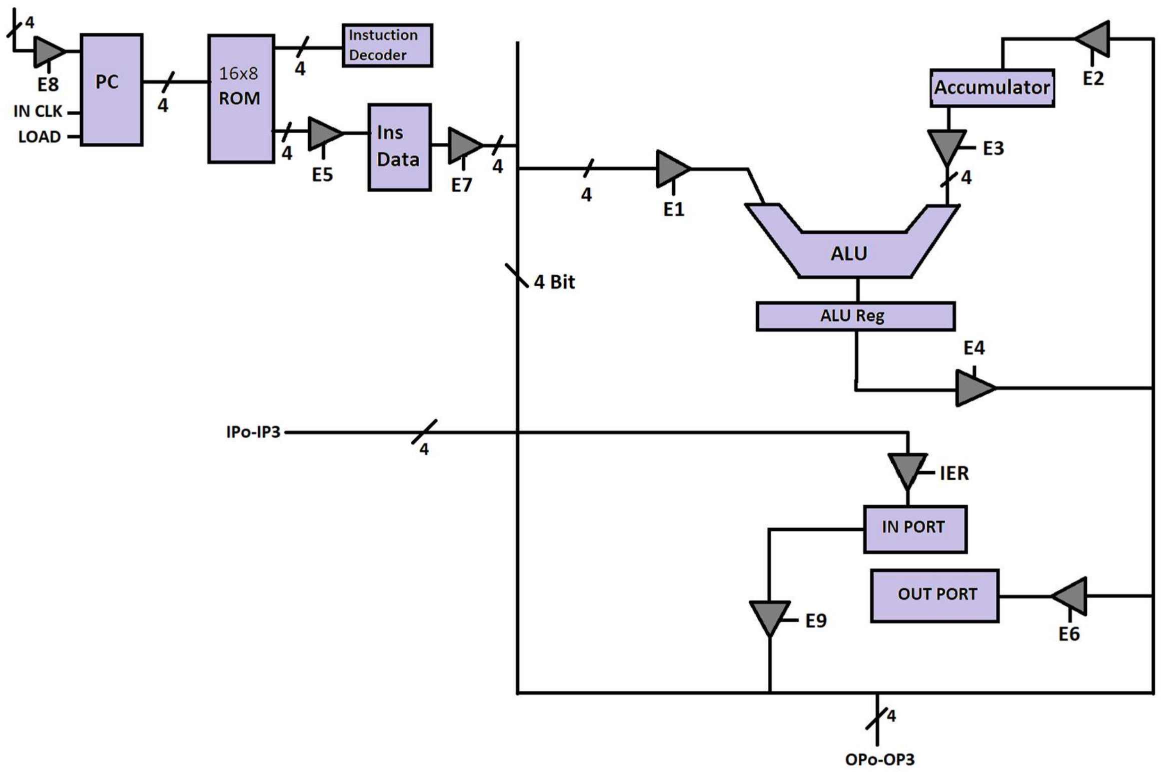
BYTE Structure:

The byte structure of the data is selected in the following fashion as the least four bits are used for the Opcode and Starting four bits are used for DATA.

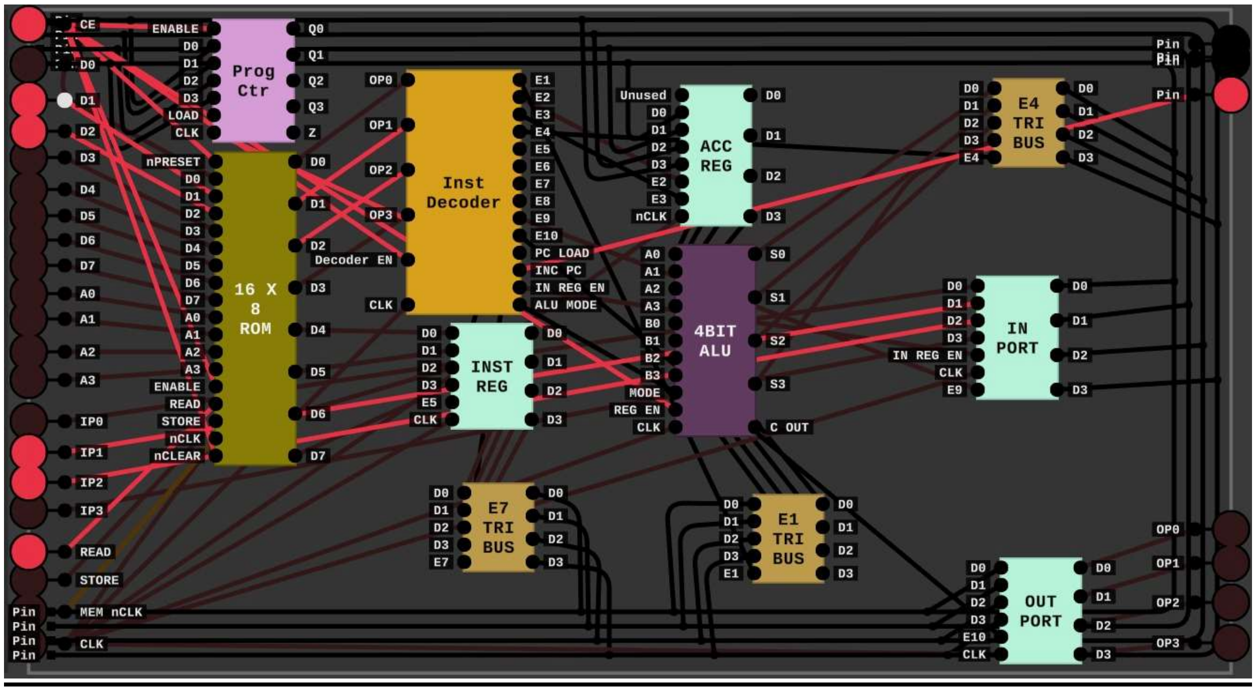


Micro controller Architecture:

The complete Architecture of the micro-controller is shown below:

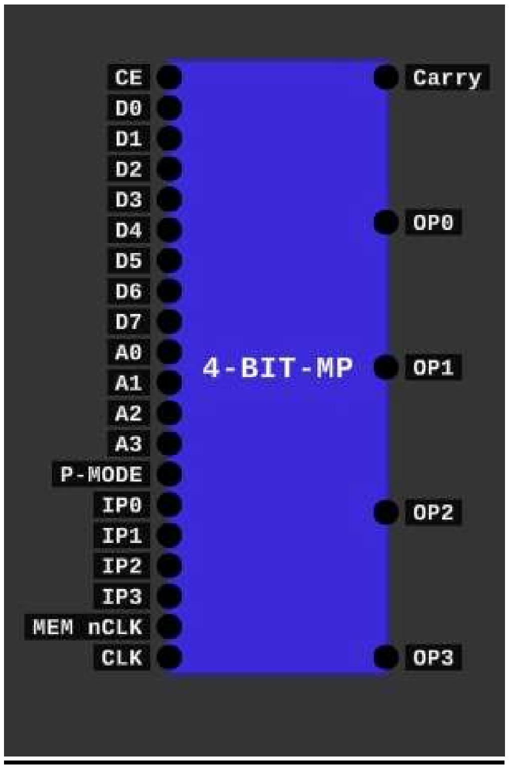


Microcontroller Structure in digital logic sim:



4-bit Microcontroller Chip:

After completion the chip of microcontroller as final is shown below:



Conclusion:

As a result, we have successfully implemented the structure of a simple 4-bit microcontroller. As in the **Digital logic sim** software the initial components available are only **AND** and a **NOT** Gate so we started with scratch and then make all the other useful logic gate such as (OR, NAND, XOR) in order to make other components like (ADDER, SR LATCH, D Flip-flops, DECODER) so by making these components we then be able to use them multiple times for implementation of complex parts of the circuit as ALU, Registers, ROM, RAM, CLU, Instruction Decoder and etc.

Challenges:

The most challenging part to implement is **CLU** (control logic unit) as it takes a lot of time to understand its logic and second is the use of data buses as in the **Digital logic sim** the buses are not Bi-directional, so we change the structure of busses to use them for bidirectional purposes.

Some useful links:

https://www.robots.ox.ac.uk/~dwm/Courses/2CO_2014/2CO-N3.pdf

Build a Superscalar CPU - YouTube