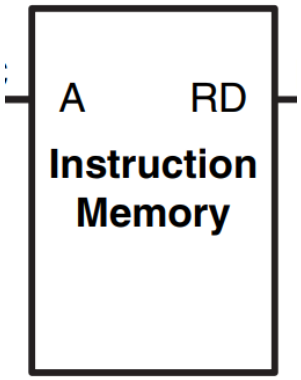
Phase Two Requirement:

- **32-Bit Program Counter**
- **32-Bit MIPS Register File**
- **Byte Addressable Instruction Memory**

Byte Addressable Instruction Memory:

Implement a byte addressable instruction memory using Verilog (address locations are 8 bit)



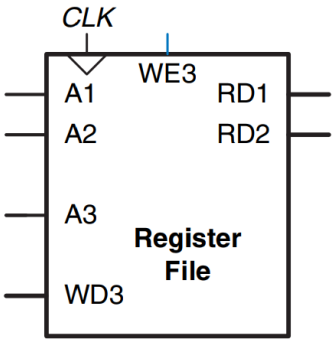
A: is a 32-Bit input determining which locations to read (the input is generally the PC)

RD: is a 32-Bit output (the instruction that is to be executed)

function of instruction memory: Holds the program instructions that are to be executed in 4 locations each (8-bit in each location, 32 bit total for each instruction)

32-Bit MIPS Register File

Implement 32-Bit MIPS Register file (Zero Register…etc)



A1: First source register (5-Bits)

A2: second Source Register (5-Bits)

A3: Destination Register (5-Bits)

WD3: the value which is to be saved (the result of the ALU in our implementation)

RD1: 32-Bit output of the value in register referred to by A1

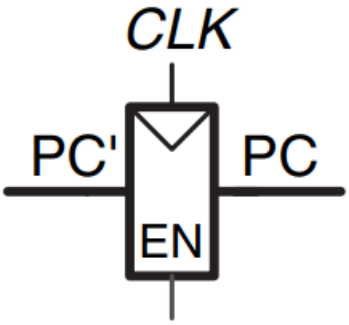
RD2: 32-Bit output of the value in register referred to by A2

WE3: Write Enable

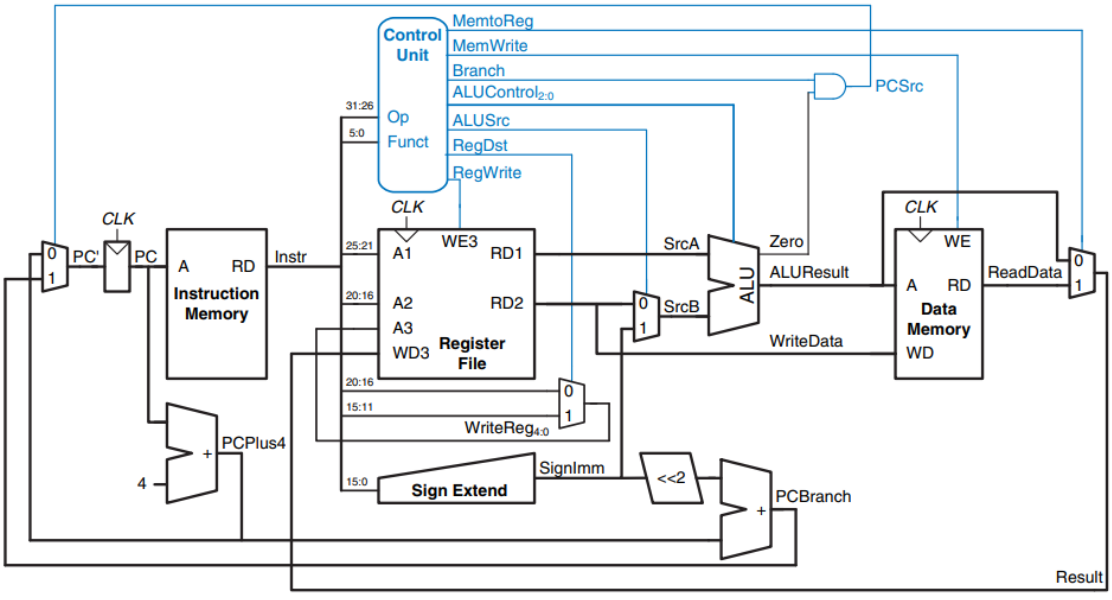
CLK: Clock Signal

32-Bit Program Counter:

Implement a 32-Bit Counter which holds the address of The instruction which is to be executed



PC Function: Increments by 4 with every clock cycle (Its initial value should be the address of the first instruction to be executed)



Full MIPS single Cycle processor Implementation

Testing: To test your circuit, Choose a simple assembly code of R-Type or I-Type Instructions and save their binary values in the instruction memory and run a behavioral simulation and see the final value in the Destination register.

Required:

-PDF of team members names and IDs, Screen shot of the simulation, Screen shots of each module and the Testbench module.

- A zipped folder of the whole project

Deadline: Saturday 25/12/2021