

CS 371
Assignment #A
Course Project

Due The Last Day/09, in class

The goal of the project is to put together a simple CPU using Multisim and run the programs from Assignments 8 and 9. Use Assignment 7 as the basis for the data path and the in-class labs as the basis for the control path. Your CPU should implement the following instructions. All instructions are 8 bits and have the following format:

OpCode₁ OpCode₀ Target B-Source Value₃ Value₂ Value₁ Value₀

Instructions

ADD <Register target> <Register or value source>
 result: <Register target> = <Register target> + <Register or value source>

CLR <Register target>
 result: <Register target> = 0

CMP <Register> <Register or value source>
 result: flags set true iff <Register> == <Register or value source>

JE <target address>
 result: if equal flag is set true then IP = <target address>
 IP is the instruction pointer and holds the address of the next instruction.

MOV <Register target> <Register or value source>
 result: <Register target> = <Register or value source>

Instruction Break Down

- 1) The opcode bits (*i.e.*, function select OpCode₁ OpCode₀) have the following meaning

OpCode	Name	Description
00	CMP	compare Target and B-Source (Target is not updated)
01	JE	jump equal (not used for this assignment)
10	MOV	move B-Source into Target, <i>i.e.</i> , Target = B-Source
11	ADD	add B-Source and Target, <i>i.e.</i> , Target = Target + B-Source

- 2) Target determines the target register as follows

Target	target register
0	R0
1	R1

Target also determines the register output on the register file's A port.

- 3) B-Source selects the source of the ALU's B input, which can be either the value output on the B port of the register file (which is always register 1) or the value component of the instruction.

B-Source	ALU's B input receives
0	Value
1	Register R1 (extra credit: use Value ₀ as a register select)

- 4) Finally, Value is a 4 bit constant value.

What to hand it

- (1) Top level schematic with your name in the **Title Block**.
- (2) Email a zip file **named** <your name>-running-ROM1.zip that contains your .ms8 files configured to run the program from Assignment 8.
- (3) Email a zip file **named** <your name>-running-ROM2.zip that contains your .ms8 files configured to run the program from Assignment 9.
- (4) Be prepared to do an in-class demo of the program from Assignment 9 (preferred) or Assignment 8.
- (5) Extra credit - implement some additional instructions.

Notes

- (1) For a cost of 10 points, you can implement ROM I only.
- (2) Use lots of LABELED 7-segment displays to help show off your circuit (this also makes the grader happier). Include two that show the current values held in R0 and R1.
- (3) Include a clock sub-circuit that can switch between automatic (using a frequency generator) and manual (using a switch) clocking.
- (4) You can optionally use Value₀ for the register file's B select input.