

California State University,  
Monterey Bay

Week 6 – Homework  
Homework 6

*Clarence Mitchell*

*CST337*

*Computer Architecture*

*Summer 2016*

*Instructor: Professor Eckhardt*

## EXERCISE

Take the schematic from the single cycle CPU and add a new instruction “MAX”.

It should do following:

MAX \$d, \$s, \$t d = max (s,t);

So \$d gets either the value of register \$s or \$t depending on which is bigger. Implement new logic to the schematic to achieve that.

You can implement new lines, muxer and logic gates.

### 1. Steps to creating

#### 1. Choose OP Code

##### 1.1. Determine the type of instruction

Since the new instruction only involves register, it is an **R-Type** instruction.

##### 1.2. Choose ALU operation (ALUOp) needed

In order to compare the register values, a SUBTRACTION (10) will be made:

**\$s - \$t :**

Zero or greater result means \$s is equal to or greater than \$t

Negative result means \$s is less than \$t.

##### 1.3. Choose ALU control input needed

As per 1.2 control input will be 110

Results table:

Instruction opcode	ALUOp	Instruction operation	Function field	Desired ALU action	ALU control input
R-type	10	subtract	100010	subtract	0110

##### 1.4. Choose actual opcode 01100 as it is not used as an opcode and is a reminder of the function code needed. So the format of my new opcode is:

01100	rs	rt	rd	shamt	100010
31 – 26	25 – 21	20 – 16	15 – 11	10 – 6	5 - 0

## 2. Modification to Datapath for New Instruction

1. In order for the new instruction to work, the following modifications to the datapath need to be made:
  - 1.1. Lines are added just after the register file from read data 1 and read data 2 lines that go into a multiplexer (**MUX**) with read data 1 as 0 and read data 2 as 1
  - 1.2. A line is added to take bit 31 from the ALU results that goes into the multiplexer (**MUX**). This selects which data value from read data lines is taken.
  - 1.3. The line leading from the rightmost multiplexer that is connected to the **MemtoReg** flag is modified to have a multiplexer before going back to the **Write Data**. The 0 line from this multiplexer is the original line and the 1 is the data line from the multiplexer in step 1.1..
  - 1.4. A AND gate is added that where the output lines goes into the multiplexer in step 1.3
  - 1.5. A line is added to the line for the instruction bits 31 – 26 that leads back to the AND gate in step 1.4. The AND gate is coded to only create a 1 when the opcode is the MAX opcode.

## 3. Datapath flow for Instruction

1. The control line setting and datapath flow for the new instructions are as follows:

RegDst	ALUSrc	Memto - Reg	Reg- Write	Mem- Read	Mem- Write	Branch	ALUOp	ALUOp0
1	0	0	1	0	0	0	1	0

- 1.1. The instruction is loaded.
- 1.2. Bits 31 – 26 go to the control unit and to the AND gate
- 1.3. Bits 25 – 21 go to Read register 1
- 1.4. Bits 20 – 16 go to Read register 2
- 1.5. Bits 15 – 11 go to the Write Register (as RegDst is asserted – set active)
- 1.6. The register data for register 1 is outputted to Read data 1 and the data for register 2 is outputted to Read data 2.
- 1.7. The read data 1 and 2 go to the ALU and to the MUX.
- 1.8. The ALU performs the subtraction operation.
- 1.9. Bit 31 from the results is sent to multiplexer that controls which Read Data value is sent down the line, and the entire bit data is sent towards the Write Data for the register file (as

the MemtoReg flag is deasserted – not active)

- 1.10. the AND gate controlled by the opcode is set asserted (active) and data from the one of the Read Data lines (from the multiplexer) is sent to the Write Data

