COMP 3315 Lab 9: Single Cycle SMIPS Connections

Read specifications below carefully.

Build:

- **a)** Create a PC component as 8 bit Register.
- **b)** Create Multiplexers with respect to given data path diagram.
- **c)** Create RAM with given specifications.
- **d)** Create Inputs for control signals (RegDst,Jump,ALUSrc,MemtoReg,RegWrite,MemRead,MemWrite,Branch)
- **e)** Use splitters to divide Instruction bit partitions.
- **f)** Connect the cables according to diagram.

Test:

Provide instruction and all control signals for each of these instructions of the following MIPS program.

addi $1, $0, 9 # reg1 = 0x00001001

addi $2, $0, 6 # reg 2 = 0x00000110

or $3, $1, $2 # reg 3 = 0x00001111

You should see 0x00001111 stored in register 3 of register file if everything works.

After:

sw $3, 3($1) # RAM location 12 should have 0x00001111

lw $4, 3($1) # Register 4 should gave 0x00001111

You are expected to implement your Datapath design and Upload .circ file for your labwork. (If you import different .circ files include them)

Specifications for SMIPS Datapath

Instruction Types:

- Memory-reference: lw, sw
- Arithmetic-logical: AND, OR, Add, Sub, Slt, XOR, XNOR and NOR, addi
- Control flow: beq, j

Memory Specifications:

- Instruction Memory: 256 x 18 (8-bit word addressing / 18-bit words)
    - The least significant 8 bits of Instruction is used
- Data Memory: 256 x 8 (8-bit word addressing / 8-bit words)
    - 8 bits of ALU result is used

Memory-reference instruction specifications:

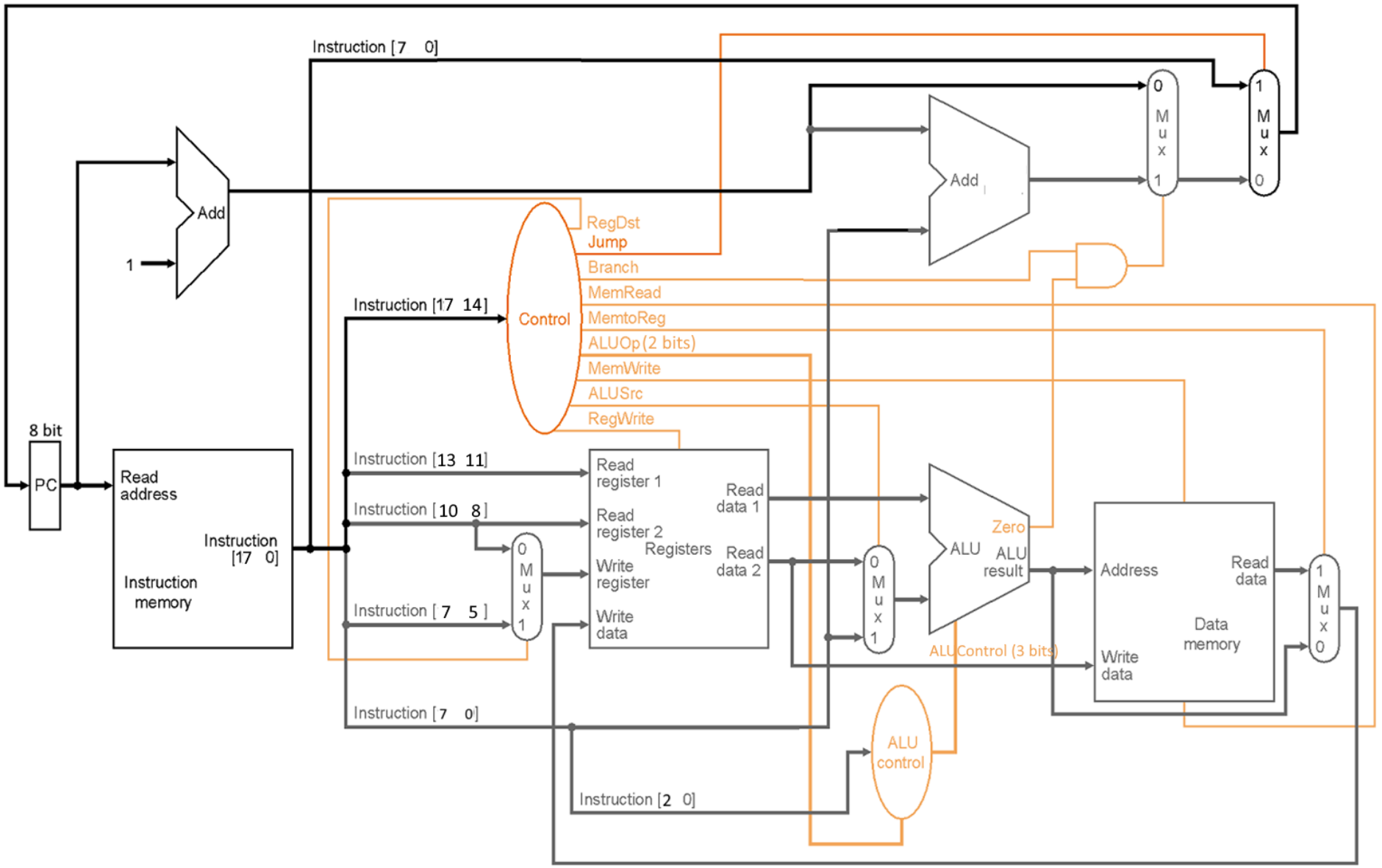
- For sw and lw, an immediate value in the interval $[-2^7, 2^7-1]$ is used as offset.
    - Memory address is the least significant 8 bits of Reg[Ins[14..12]] + Ins[7..0]

Control flow specifications:

- For beq, an immediate value in the interval $[-2^7, 2^7-1]$ is used to compute the branch address.
    - Branch address is PC + 1 + Ins[7..0]
- For j, the least significant 8 bits of the instruction is used directly as the jump address.
    - Jump address is Ins[7..0]

Immediate Instruction specifications:

- addi instruction is included.
    - Reg[Ins[11..9]] = Reg[Ins[14..12]] + Ins[7..0]


Instruction [7 0]
Add
1
8 bit
PC
Read address
Instruction [17 0]
Instruction memory
Instruction [17 14]
Control
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp (2 bits)
MemWrite
ALUSrc
RegWrite
Instruction [13 11]
Instruction [10 8]
Instruction [7 5]
Instruction [7 0]
Instruction [2 0]
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
0 M u x 1
Add
0 M u x 1
1 M u x 0
ALU
Zero
ALU result
ALU Control (3 bits)
ALU control
Address
Read data
Data memory
Write data