COMP 3315 Lab 10: Single Cycle SMIPS Controller

Create Controller and ALU Controller:

a) Use ALU Controller table to build “ALU control” with 2-bit Aluop input and 3-bit Function field input and 3 bit alucontrol output.

b) Use Controller table to build “Control” with 4-bit OP code input and 10 bit control signal output.

Both controllers are combinational circuits. No need for a Synchronous signal in any of controllers.

You should build both Controllers as subcircuits and add them to your main SMIPS datapath. Upload .circ file for your labwork. (If you import different .circ files also include them. If you do it in 1 .circ file it will be better.)

To test the datapath you can try the program below:

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

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

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

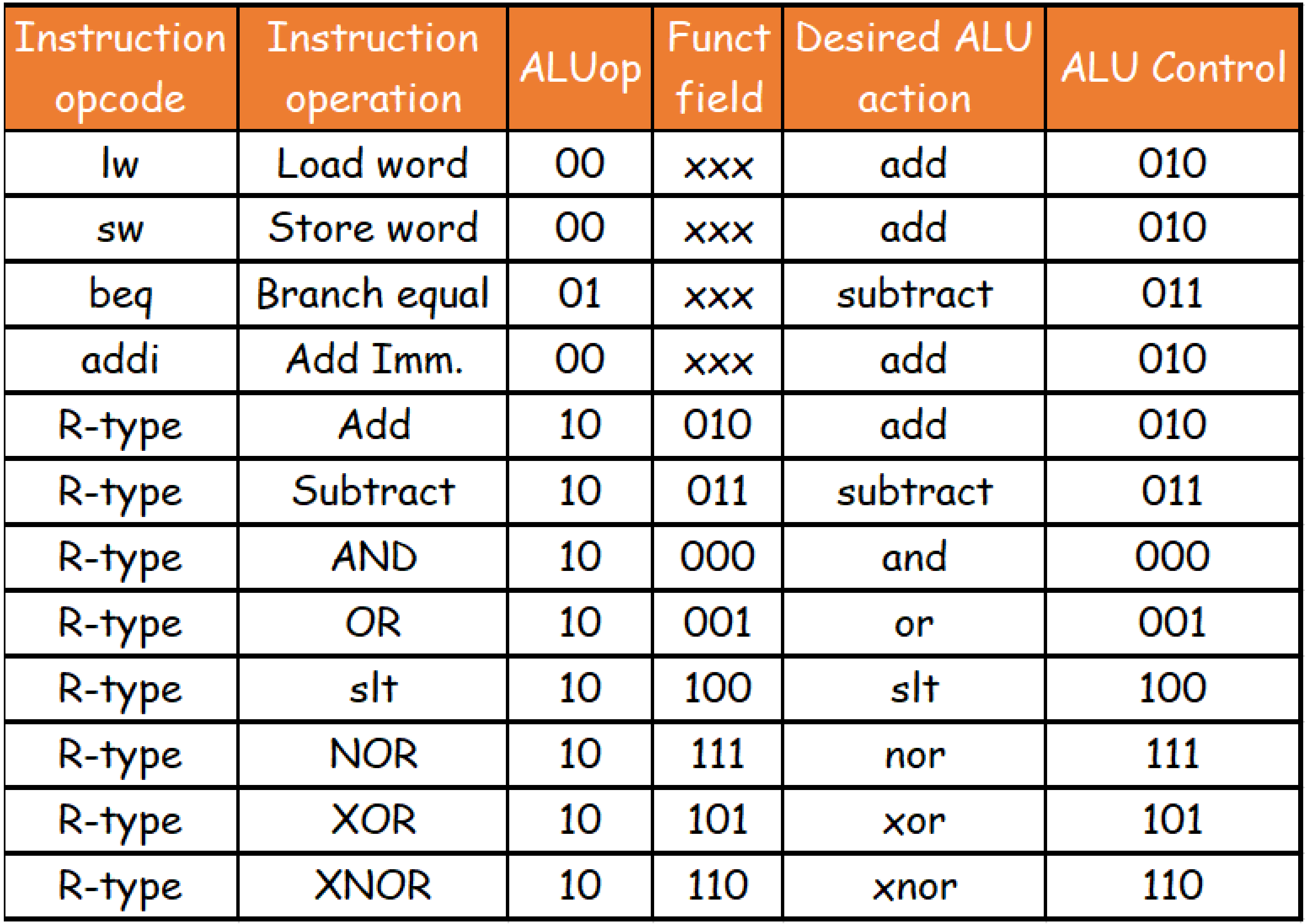
You should see 0x00001111 stored in register 3 of register file if everything works.

sw $3, 4($2) # RAM location 10 should have 0x00001111

lw $4, 4($2) # Register 4 should gave 0x00001111

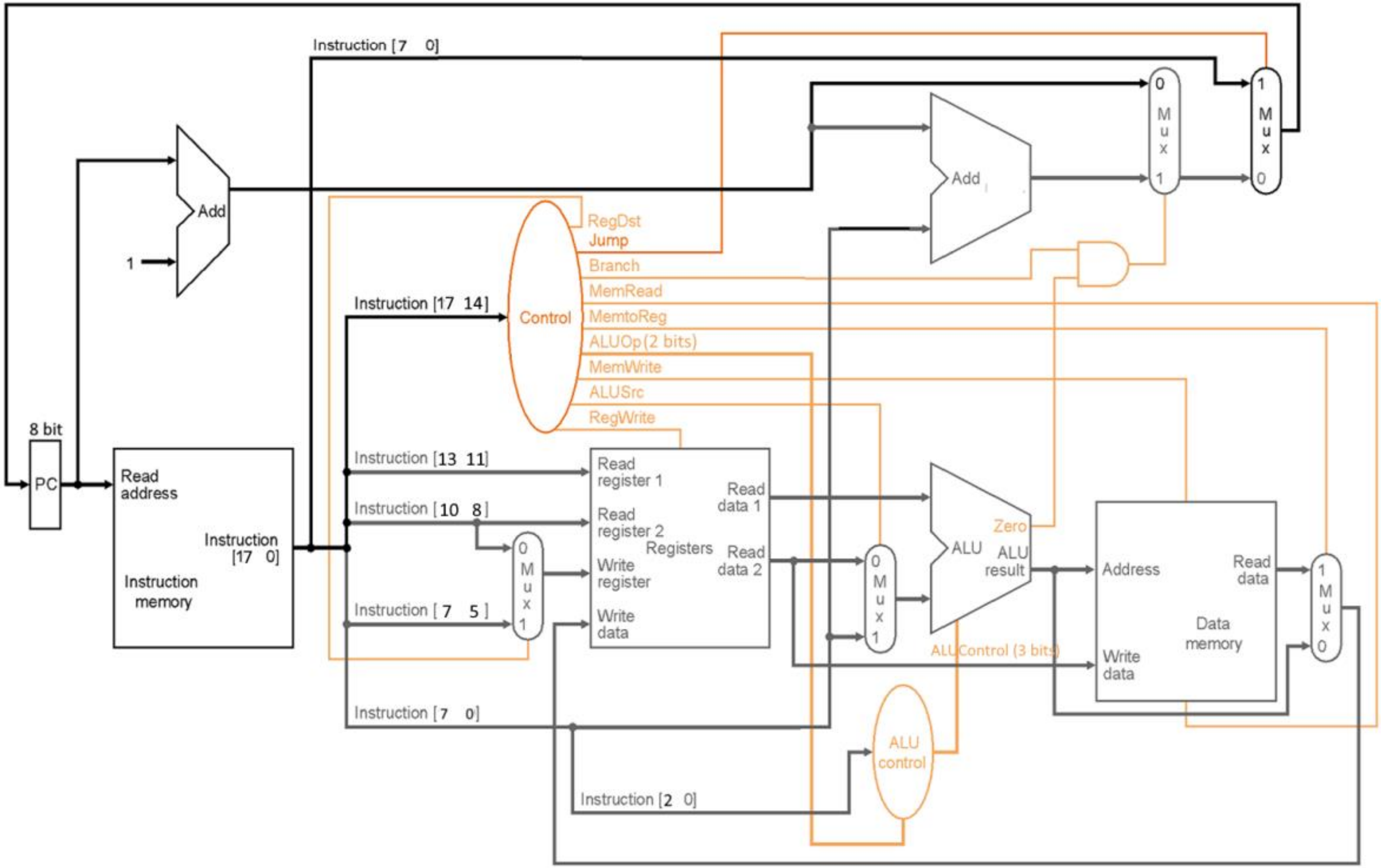
Input or output	Op Code				Outputs									
Signal name	Op3	Op2	Op1	Op0	RegDst	ALUSrc	Jump	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALIOp0
R-format	0	0	0	0	1	0	0	0	1	0	0	0	1	0
lw	0	0	1	1	0	1	0	1	1	1	0	0	0	0
sw	1	0	1	1	x	1	0	x	0	0	1	0	0	0
beq	0	1	0	0	x	0	0	x	0	0	0	1	0	1
addi	1	0	0	0	0	1	0	0	1	0	0	0	0	0
j	0	0	1	0	x	x	1	x	0	0	0	x	x	x

Controller Table



Instruction opcode	Instruction operation	ALUop	Funct field	Desired ALU action	ALU Control
lw	Load word	00	xxx	add	010
sw	Store word	00	xxx	add	010
beq	Branch equal	01	xxx	subtract	011
addi	Add Imm.	00	xxx	add	010
R-type	Add	10	010	add	010
R-type	Subtract	10	011	subtract	011
R-type	AND	10	000	and	000
R-type	OR	10	001	or	001
R-type	slt	10	100	slt	100
R-type	NOR	10	111	nor	111
R-type	XOR	10	101	xor	101
R-type	XNOR	10	110	xnor	110

Alu Control Table



MIPS Datapath

Specifications for SMIPS Datapath

Instruction Types:

- Memory-reference: lw, sw
- Arithmetic-logical: add, sub, and, or, nor, xor,xnor, slt, 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]).