Problem Set 5

Justin Ely

605.411 Foundations of Computer Architecture

04 October, 2016

---

1)

An r-type instruction goes through the states: 0, 1, 6, 7, before returning back to 0. From the values given in the slides, the control values and next step codes in sequence are shown below.

Step	Control	Next
0	1001010000001000	0001
1	0000000000011000	0110
6	0000000001000100	0111
7	0000000000000011	0000

In hex, this sequence would be: 0x94081, 0x00186, 0x00447, 0x00030.

2)

- r-type
- load word

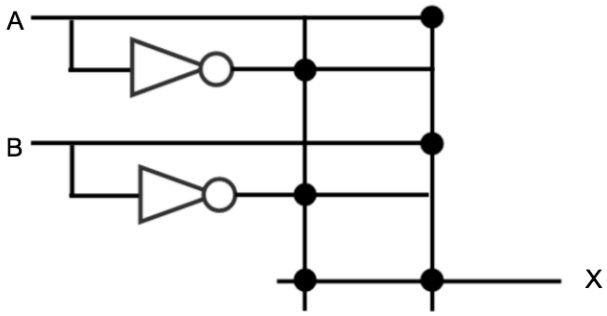
3)

A	B	X	Min-term	Max-term
0	0	1	A‘B‘	
1	0	0		A‘ + B
0	1	0		A + B‘
1	1	1	AB	

4a)

000100	00000	00100	0000000000001010

Figure 1: PLA for problem 3.



The op-code says this is a BEQ (I-type) instruction. BEQ goes through states 0, 1, 8 on the state diagram.

4b)

0, 1, 2, 6, 8

4c)

- Step 0: ALUSrcA = 0, ALUSrcB = 01, ALUOp = 00
- Step 1: ALUSrcA = 0, ALUSrcB = 11, ALUOp = 00
- Step 2: ALUSrcA = 1, ALUSrcB = 10, ALUOp = 00
- Step 6: ALUSrcA = 1, ALUSrcB = 00, ALUOp = 10
- Step 8: ALUSrcA = 1, ALUSrcB = 00, ALUOp = 01

4d)

- Step 0: ALU1 = PC, ALU2 = 4
- Step 1: ALU1 = PC, ALU2 = shift-by-two unit output
- Step 2: ALU1 = Register A, ALU2 = sign-extension unit output
- Step 6: ALU1 = Register A, ALU2 = Register B
- Step 8: ALU1 = Register A, ALU2 = Register B

4e)

The branch is comparing register \$0 to register \$4. Since \$0 = \$4 = 0, the branch is taken. The new PC value will be PC + 4 + (I-value < < 2) = 0000 0000 0100 0000 0000 0000 0010 1100.

5)

X2	X1	X0	a	b	c	d	minterm	maxterm
0	0	0			y		X2‘X1‘X0‘	X2 + X1 + X0
1	0	0		y		y	X2X1‘X0‘	X2‘ + X1 + X0
0	1	0		y	y		X2‘X1X0‘	X2 + X1‘ + X0
1	1	0	y			y	X2X1X0‘	X2‘ + X1‘ + X0
0	0	1		y	y		X2‘X1‘X0	X2 + X1 + X0‘
1	0	1	y			y	X2X1‘X0	X2‘ + X1 + X0‘
0	1	1	y		y		X2‘X1X0	X2 + X1‘ + X0‘
1	1	1				y	X2X1X0	X2‘ + X1‘ + X0‘

5a)

X = (X2X1X0’) + (X2X1‘X0) + (X2‘X1X0)

5b)

X = (X2X1‘X0‘) + (X2‘X1X0‘) + (X2‘X1‘X0)

5c)

X = (X2‘X1‘X0‘) + (X2‘X1X0‘) + (X2‘X1‘X0) + (X2‘X1X0)

5d)

X = (X2X1‘X0‘) + (X2X1X0‘) + (X2X1‘X0) + (X2X1X0)

6)

See figure below.

7)

From the state diagram given in the lectures:

- beq: 3
- j: 3
- or: 4
- add: 4
- lw: 5
- sw: 4
- bne: 3

8)

- lw $2,40($4): Uses the memory system when it is retrieving the specified word, the ALU to calculate the address, and the register file to see in which register to write the loaded word.
- add $3,$6,$7: Uses the ALU do perform the arithmetic, and the register file to read operands and write result.
- slt $5,$6,$7: Uses the ALU to check if one register is less than another, and the register file to read operands and set the result.
- sw $6,44($4): Uses the register file and ALU to retrieve and calculate the address in memory to put the register value.
- and $9,$11,$10: Uses the ALU to perform the AND operation, and the register file to read operands and write result.
- or $21,$27,$8: USes the ALU to perform the OR operation, and the register file to read operands and write result.

Figure 2: PLAs for problem 6. Top to bottom: 5a, 5b, 5c, 5d