Problem Set 4

Justin Ely

605.411 Foundations of Computer Architecture

27 September, 2016

---

1)

The BEQ instruction is performed using **c: subtraction**.

2)

The ALU control bits 0110 signify **b: subtract**.

3)

The 4-bit ALU control pattern for add is **0010**.

4)

If the zero flag is 1, then the branch will be taken. Thus, the PC will be set to 0x08000000, incremented by 4, plus 0xAA shifted two bits to the left.

PC	0000	1000	0000	0000	0000	0000	0000	0000
+4	0000	0000	0000	0000	0000	0000	0000	0100
+(0xAA<<2)	0000	0000	0000	0000	0000	0010	1010	1000
result	0000	1000	0000	0000	0000	0010	1010	1100

Thus, the PC will be set to **0x80002AC**

5)

From the slides in the lecture:

	Memory?	Sing-extend?
add	No	No
addi	No	Yes
not	NA	NA
beq	No	Yes
lw	Yes	Yes
sw	Yes	Yes

5a)

Data memory is used in lw and sw, which account for $25\% + 10\% = \mathbf{35}\%$ of cycles in a single-cycle datapath.

5b)

The sign extend circuit is needed in addi, beq, lw, and sw, which account for $20\% + 25\% + 25\% + 10\% = \mathbf{80}\%$ of cycles in a single-cycle datapath.

6a)

RegDst	ALUSrc	MemtoReg	RegWrt	MemRead	MemWrt	Branch	ALUOp1	ALUp0
1	0	0	1	0	0	0	1	0

6b)

- Main control
- ALU
- ALUControl
- Register block
- PC count incrementer

6c)

- Sign extend
- Branch ALU

6d)

- Memory

7)

The SW instruction access the register file, the ALU, memory, and the register file again. This totals $2 + 6 + 12 + 2 = \mathbf{22ns}$.

8)

101011	10011	00010	0000000000010100
43	19	2	20
sw	$s3	$v0	20

8a)

The control unit's inputs are the ALUOp and the function code. The ALUOp for the SW instruction is 00, and the function code is ignored.

8b)

The input values to the ALU are the contents of the two registers, the offset (immediate value), and the ALU control code for SW: 0010.