CS 240 SI Worksheet
Valerie Zhao
Session #7
2/23/17

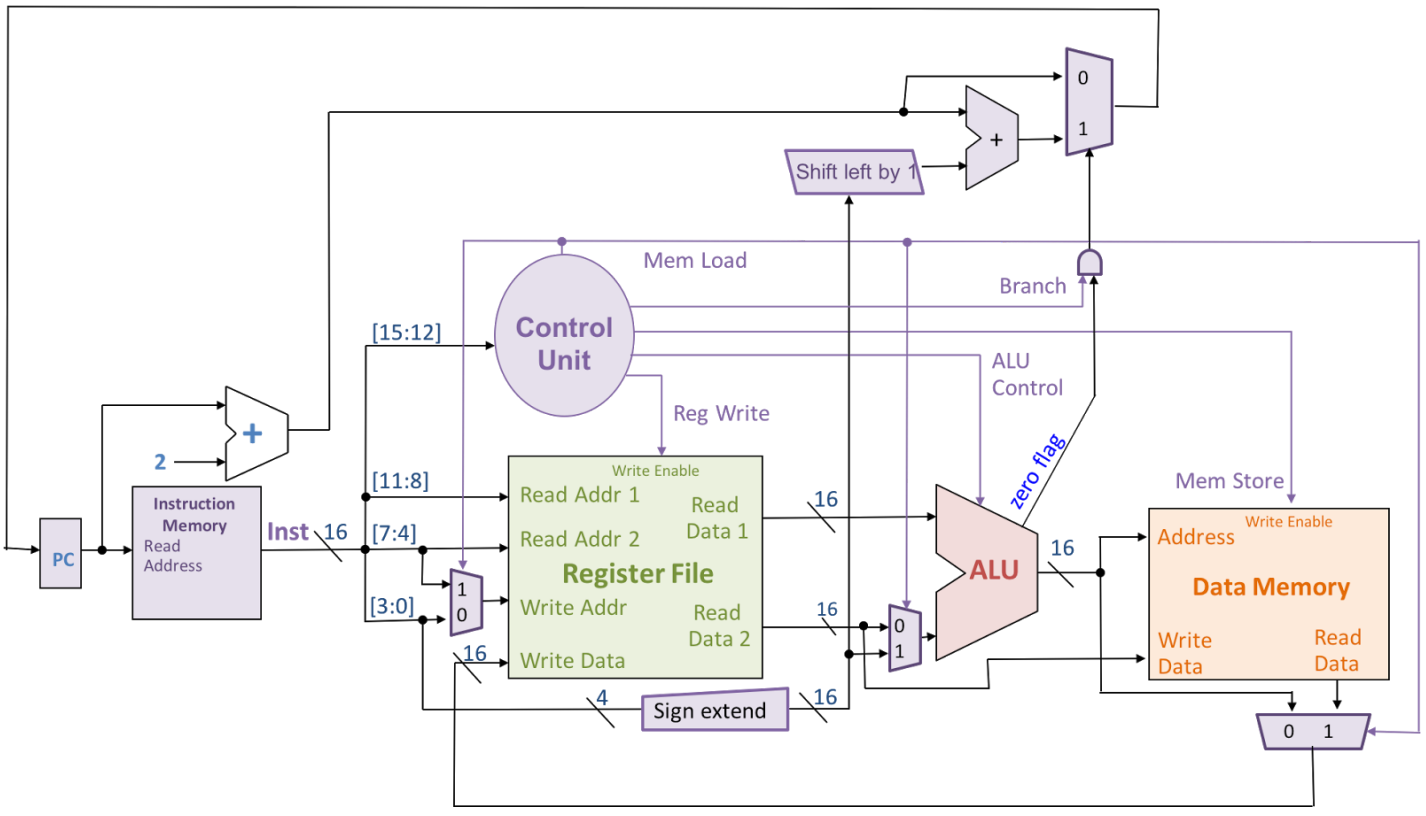
The Processor: Instruction Set Architecture + Microarchitecture

Complete the following table (only the blanks, not the ??):
(You get to simulate the program on the next page)

Address	Opcode	Operation	s	t	d/Offset
0x0 - 0x1	**0011**	SUB	5	5	5
0x2 - 0x3	**0000**	LW	6	7	4
0x4 - 0x5	**0111**	BEQ	6	0	3
0x6 - 0x7	**0010**	ADD	7	5	5
0x8 - 0x9	**0011**	SUB	6	1	6
0xA - 0xB	**0111**	BEQ	6	1	-4
0xC - 0xD	**0001**	SW	6	5	4
0xE - 0xF	HALT				

Register File	
Reg	**Contents**
R5	??
R6	2
R7	??

Data Memory		
Address	**Contents**	
0x4 - 0x5	??	??
0x6 - 0x7	3	0



Given the state of the register file and data memory, **simulate each step** of the program and record the **state of the program counter, register file, and data memory after each step**, as well as the **outputs of the control unit** that enabled the execution of the instruction.

Instruction in Memory (in Assembly Syntax)	**PC**	**Reg File Changes**	**Data Memory Changes**	**ALU Control**	**Mem Load**	**Mem Store**	**Reg Write**	**Branch**
SUB R5, R5, R5	0x2	R5 = 0	n/a	0110	0	0	1	0
LW R7, 4(R6)	0x4	R7 = 3	n/a	0010	1	0	1	0
BEQ R6, R0, 3	0x6	n/a	n/a	0110	0	0	0	1
ADD R7, R5, R5	0x8	R5 = 3	n/a	0010	0	0	1	0
SUB R6, R1, R6	0xA	R6 = 1	n/a	0110	0	0	1	0
BEQ R6, R1, -4	0x4	n/a	n/a	0110	0	0	0	1
BEQ R6, R0, 3	0x6	n/a	n/a	0110	0	0	0	1
ADD R7, R5, R5	0x8	R5 = 6	n/a	0010	0	0	1	0
SUB R6, R1, R6	0xA	R6 = 0	n/a	0110	0	0	1	0
BEQ R6, R1, -4	0xC	n/a	n/a	0110	0	0	0	1
*(BEQ R6, R0, 3)	(0x6)	(n/a)	(n/a)	(0110)	(0)	(0)	(0)	(1)
SW R5, 4(R6)	0xE	n/a	6 at 0x4	0010	1	1	0	0
HALT								

In the given program, the instruction marked with * (BEQ R6, R0, 3) doesn't execute (again). What changes to the instruction at 0xA - 0xB, **BEQ R6, R1, -4** (which is also the instruction that would execute before the * instruction), can you make so that the new program gives the same result as the original one?

(The run of the new program should be very similar to the original, only because of the given values in the register.)