CSE 306

Computer Architecture Sessional

Quiz Examination

Time: 65 minutes (**including uploading time**) Full Marks: 45

1. **MCQ Questions:** Write down the correct answer with the full answer. For example, if the correct answer of MCQ no. xi is a then write down in the following format *9x1=9*
XI. a

I. After an unsigned operation $A - B$, if you are shown only the value of Carry flag and you find it set (Carry Flag=1), then which of the following can you guarantee?

a. $A > B$

b. $A < B$

c. $A \ngtr B$

d. $A \nless B$

II. Consider a 7-bit ALU, what is the function to determine overflow (V) flag? Note that C_1 is the first input carry bit.

a. $C_8 \oplus C_9$

b. $C_8 + C_9$

c. $C_8 C_9 + C_8' C_9'$

d. $C_7 C_8' + C_7' C_8$

III. If $A = 0100$, $B = 0100$ and $C_{in} = 0$, then which of the following is FALSE after the add operation using a 4-bit ALU?

a. $Z = 0$

b. $C = 0$

c. $V = 0$

d. $S = 1$

IV. The formula for calculating the bias for an arbitrarily sized floating-point number where there are k bits for the exponent is,

a. $2^k - 1$

b. $2^{(k-1)} - 1$

c. 2^k

d. $2^{(k-1)}$

V. For IEEE 754 standard, single precision 32-bit and double precision 64-bit numbers have x and y bits respectively for the exponent. Here the value of x and y are,

a. 8 and 11

b. 7 and 13

c. 7 and 15

d. 8 and 16

VI. What is NaN in IEEE standards?

a. Not a finite number

b. Not a node

c. Not a number

d. Not a normalized number

VII. Which of the following MIPS instruction does NOT need any arithmetic operation?

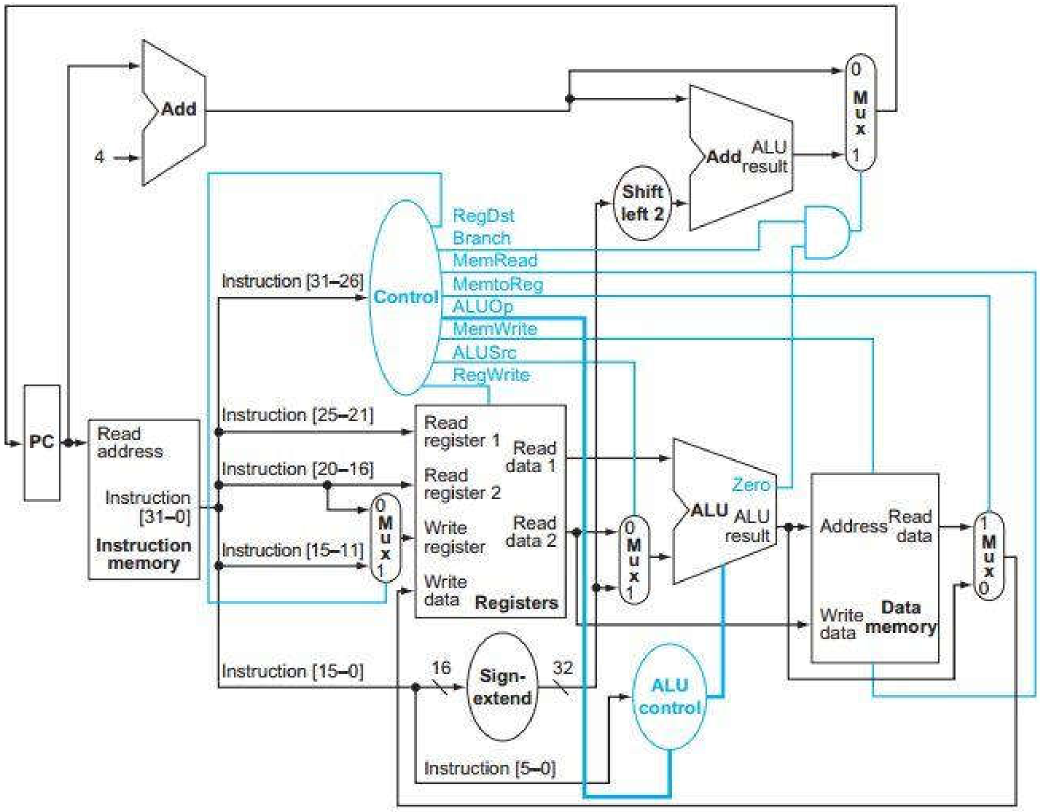
a. addi

b. beq

c. j

d. none of the above

VIII. See the figure below.



Now, consider the follwing statements with respect to the above figure.

Statement 1: For instruction *beq* (branch on equal) the control signal ***MemtoReg*** can be 1.

Statement 2: The *addi* instruction can be handled with the datapath.

Statement 3: The only implementation difference between *jump* and *branch* instruction is that the latter one is conditional.

Now which one of the following is correct about the above mentioned three statements?

a. True: Statement 2 and Statement 3
   False: Statement 1

b. True: Statement 1 and Statement 3
   False: Statement 2

c. True: Statement 1 and Statement 2
   False: Statement 3

d. True: Statement 1
   False: Statement 2 and Statement 3

IX. How many instructions are in different stages in the same clock cycle in a 5-stage pipelined execution?

a. Up to 4

b. Up to 5

c. Up to 6

d. One

2. Suppose A ($A_3A_2A_1A_0$), B ($B_3B_2B_1B_0$), $\boldsymbol{S_2}, \boldsymbol{S_1}$and $\boldsymbol{S_0}(C_{in})$ are the inputs to your circuit while X_i, Y_i *and* Z_i are the inputs to the i^{th} parallel adder. Now, derive the input equations (X_i, Y_i *and* Z_i) for the 4-bit parallel adders to be used in the ALU which satisfies the following functional design specification. Your design should use as less number of gates as possible. ***12***

S_2	S_1	$S_0(C_{in})$	**Required Operations**
0	0	0	Transfer A
0	0	1	2's complement of A
0	1	0	Add A with B
0	1	1	Add 2's complement of A with B
1	0	x	1's complement of A
1	1	x	A XOR B

3. For the IEEE 754 standard, answer the following questions: ***3+3=6***

   a. What are the advantages of using biased exponent representation?

   b. What is the representation of Zero and the representation of Infinity?

4. Consider the following two formats to represent a 16-bit floating-point number. For each format, determine the largest and smallest positive floating-point numbers. ***3+3=6***

   a. Format 1:

Sign 1 bit	Exponent 4 bit	Mantissa 11 bit

   b. Format 2:

Sign 1 bit	Exponent 5 bit	Mantissa 10 bit

5. In this question, you need to consider the pipelined execution of the R-type MIPS instructions. The execution of each instruction is divided into five stages: stage 1 (Instruction Fetch), stage 2 (Instruction Decode), stage 3 (Execution), stage 4 (Data Memory), and stage 5 (Write Back). ***3+3=6***

Also consider that the pipeline register between stage 1 and stage 2 is denoted as Register **A**, the pipeline register between stage 2 and stage 3 is denoted as Register **B**, the pipeline register between stage 3 and stage 4 is denoted as Register **C**, and the pipeline register between stage 4 and stage 5 is denoted as Register **D**.

stage 1 |**A**| stage 2 |**B**| stage 3 |**C**| stage 4 |**D**| stage 5

Now consider the following instructions.

//Initially $t1= 10, $t2 = 5, $t3 = 0, $t4 =10, $t5 =0

Inst1: add $t3, $t1, $t2
Inst2: or $t4, $t1, $t2
Inst3: and $t2, $t1, $t2
Inst4: add $t5, $t3, $t1

When ***Inst1*** is in Stage 5 (Write Back), ***Inst4*** is in Stage 2 (Instruction Decode) in the same clock cycle CC. Ideally, $t3 is written on the 1st half cycle of CC for ***Inst1*** and $t3 is read in the 2nd half cycle of CC for ***Inst4***. So ***Inst4*** reads and propagates the updated value $t3 = 15.

Suppose, you have made some mistake in your implementation which have caused some race conditions in your implementation. As a result, $t3 has NOT been updated to the expected value 15, and hence ***Inst4*** reads $t3 = 0 (old value of $t3) in its 2nd stage and propagates $t3 = 0 to the successive stages. Let us name this circumstance as *WB hazard*.

Now you want to handle *WB hazard* using data forwarding techniques, i.e., forwarding the updated value $t3 = 15 to Stage 2 of ***Inst4***. To verify the correctness of your strategy, answer the following questions.

   a. Show the pipelined execution for the above 4 instructions. Also, show the forwarding path between ***Inst1*** and ***Inst4*** to resolve the *WB* hazard.
   b. Write down the corresponding forwarding condition.

6. In the following MIPS architecture (Figure for Question 6), mark the Datapath and control signal for the instruction from the following table based on the last three digits of your student ID. You can find an assigned alphabet (in red font) beside each of the wire of the Datapath. In your script you must write the letters whose associated lines are active for completing your assigned instruction. Additionally, in your script, write the control signals that need to be enabled. For each mux (m1, m2, m3, m4), write down the mux control input (0/1). *6*

(Last 3 digits of student ID) mod 3	Instruction
0	*lw* (load word)
1	*sw* (store word)
2	*beq* (branch on equal)

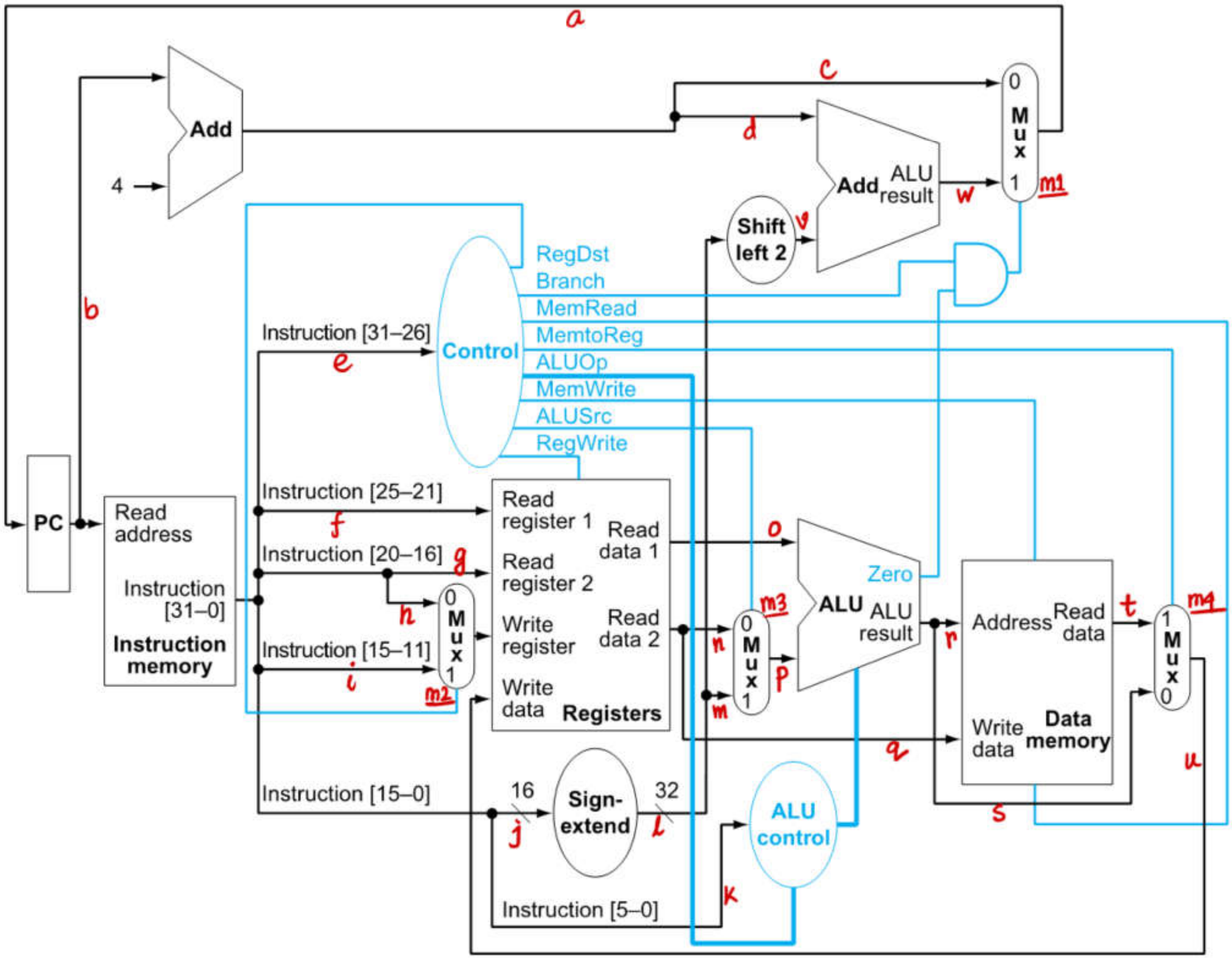


Figure for Question 6