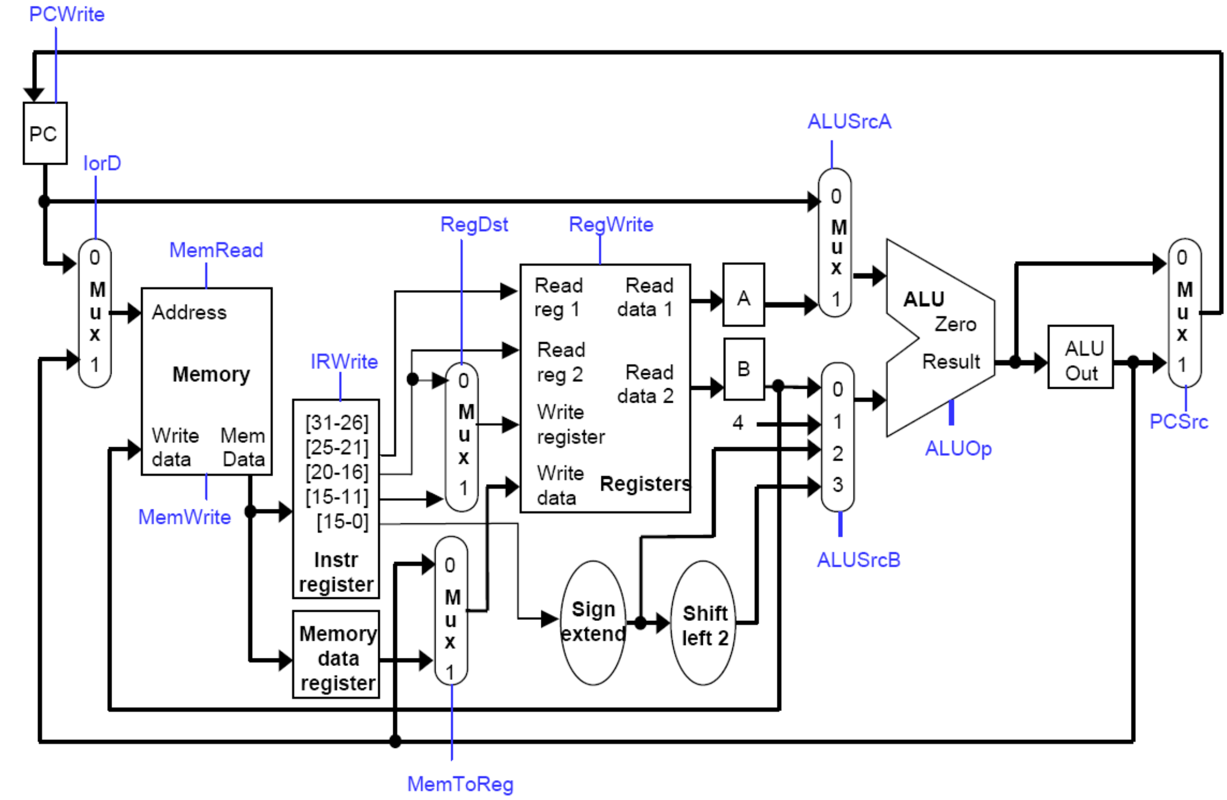
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
II SEMESTER 2013-2014
IS F242 COMPUTER ORGANIZATION
CS C342 ADVANCED COMPUTER ORGANIZATION
COMPREHENSIVE EXAM PART A (CLOSE BOOK)

TIME: 120 Min. **5/5/2014** **MM: 55**

IDNO: **NAME:**

Q1. It is possible to include ***jalr*** in a multicycle processor by modifying the datapath and control unit as discussed in class and shown below. Mark on the diagram shown below the changes in connections and/or components of the datapath for implementation of ***jalr***. You can remove existing components and include new ones if required. Your implementation of ***jalr*** should use minimum stages for execution.

Draw the finite state machine diagram for the ***jalr*** instruction. Remember to account for any control signals that you added or modified in the previous part of the question. **[8]**



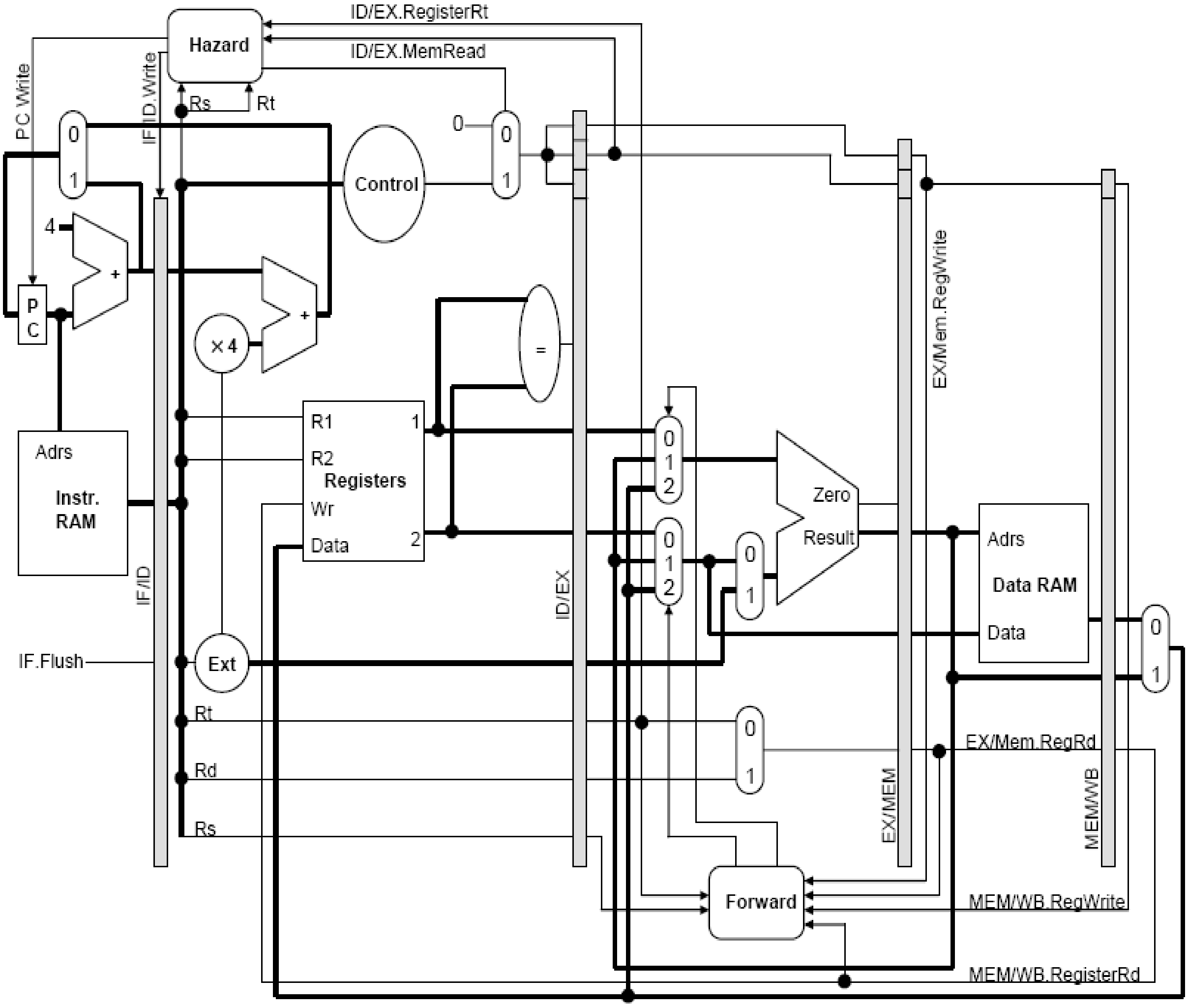
Q2. Consider the pipelined datapath shown in Figure below.

(a) Describe the difficulty in executing the following instructions in this datapath. How many stall cycles must be inserted to correctly execute this code?

Lw $a4, 5($a7)
Sub $a3, $a3, $a1
Beq $a3, $a4, Label

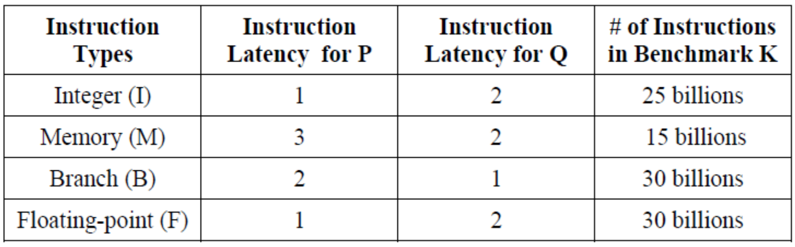
(b) Show how the number of stalls could be minimized with forwarding in the figure below. You may add new components in the data path and strike off current components if required. Give the governing equations for the forwarding/hazard detection unit.

You can assume the register file and ALU have the same latency, while muxes have negligible delays. [8]



EQUATIONS:

Q3. Given two processors **P** and **Q** running at 2.8 GHz and 3.0 GHz respectively. These two machines were designed based on the same ISA. The instructions are classified into 4 types: integer, memory, branch and floating-point. Due to the constraint of different frequency objectives, these 2 processors have different instruction latencies as shown in the table. Also note, both processors are single-issue, non-pipelined. You are asked to evaluate the performance by running a benchmark program **K**. The instructions breakdown for K is shown in the table.



Instruction Types	**Instruction Latency for P**	**Instruction Latency for Q**	**# of Instructions in Benchmark K**
Integer (I)	1	2	25 billions
Memory (M)	3	2	15 billions
Branch (B)	2	1	30 billions
Floating-point (F)	1	2	30 billions

(a) Which machine actually delivers better overall performance (measured in time)? Derive your answer.

(b) For **machine P**, architects invent a new instruction called I-M that can **combine** some integer with memory instruction into one instruction. The new instruction has a **2-cycle latency**. For example, if one can find 10 billion such instructions to combine, the I and M instructions for P will be reduced to 15 and 5 billions respectively while with additional 10 billion I-M instructions. However, due to the addition of this new instruction, the frequency is slowed down to 2.7GHz. In order to receive speedup on P for using this new class of instructions, how many I and M instructions need to be found and combined? [4]

Q4. (a) The figure below shows the pipelined datapath with forwarding, but no hazard detection unit. Both of the code fragments below have dependencies, but only one of them will execute correctly with the forwarding datapath. Which one, and why? Draw the pipeline diagram for the problematic case showing the stalls required for correct execution.

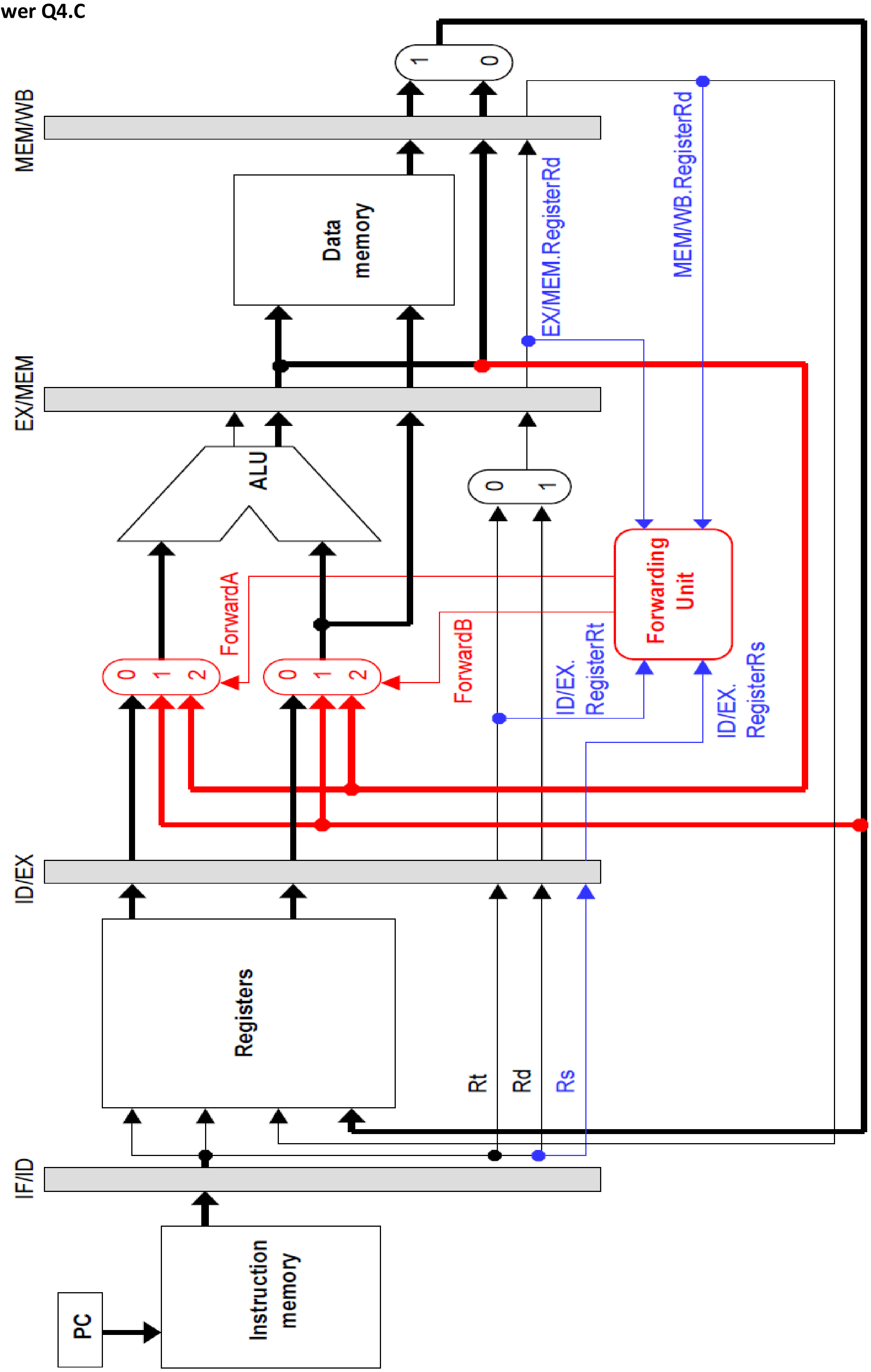
```
add $t0, $a0, $a1          lw $t0, 0($a0)
add $v0, $t0, $t0         add $v0, $t0, $t0
```

(b) Now consider the following code fragment. Draw the pipeline diagram for this code showing the stalls (if any) required for correct execution.

```
Add $t0, $t1, $t2
sw $t0, 4($t3)
```

(c) It is possible to modify the datapath so the code in Part (b) executes with *minimum* stalls. Explain how this could be done, and show your changes on the diagram. Also write the equations governing the generation of any new control signal you may use. [8]

Answer Q4.C


MEM/WB
EX/MEM
ID/EX
IF/ID
PC
Instruction memory
Registers
Rt
Rd
Rs
ALU
Data memory
0
1
2
0
1
2
ForwardA
ForwardB
Forwarding Unit
EX/MEM.RegisterRd
MEM/WB.RegisterRd
ID/EX. RegisterRt
ID/EX. RegisterRs
0
1
1
0

Q9. What is meant by Register Pressure? [1]

Q10. How is CDB different from a normal data bus? [1]

Q12. The branch outcome of one branch instruction was (T, T, NT, NT, NT). What are the accuracies of the following branch predictors on this branch instruction? [5]

(a) (1,1) correlating predictor with all entries initialized to Taken

(b) Always-not-taken predictor

(c) Two-bit predictor assuming that the predictor starts in the deep state of predict not taken

(d) Two-bit predictor when the predictor executes this pattern repeatedly forever

Q13. (a)Use register renaming to rewrite the following code segment to eliminate WAW and WAR dependencies. Assume only two more registers R4 and R5 are available. [5]

```
lw R2, 0(R1)
add R3, R2, R2
sw R3, 0(R1)
lw R2, 4(R1)                    ➔
add R3, R2, R2
sw R3, 4(R1)
```

(b) Assuming that infinite registers are available; can all name dependences be solved through register renaming? Explain in brief.