National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Advance Computer Architecture	Course Code:	EE502
Program:	MS (Computer Science)	Semester:	Fall2018
Duration:	90 Minutes	Total Marks:	40
Paper Date:	31-10-2018	Weight	20
Exam Type:	Midterm	Page(s):	6

Student : Name:_____ **Roll No.**_____

Instruction:
1. Attempt all question in the provided space. You can use rough sheets but it should not be attached.
2. If you think some information is missing, write assumption and solve accordingly.

Question 1 [5]

1. Which of the following statements regarding pipeline hazards is TRUE? (Multiple options can be selected)
   A. All data hazards can be avoided by forwarding
   B. Branching causes both data and control hazards
   C. Data hazards may cause error in execution result
   D. Structural hazards occur whenever an instruction depends on another for operand
   E. None of the above

2. Following MIPS code fragment is executed on the 5-stage pipelined datapath with forwarding.
   add R3, R1, R2
   sub R1, R1, R2
   andi R2, R3, 0xFF
   In the EX stage, the operand R3 for the **andi** instruction gets forwarded from

   A. Pipeline register IF/ID
   B. Pipeline register ID/EX
   C. Pipeline register EX/MEM
   D. Pipeline register MEM/WB
   E. No forwarding is necessary

3. CPU time can be reduced by reducing: (Multiple options can be selected)
   A. Instruction count
   B. Clocks per instruction
   C. Clock cycle time
   D. Clock frequency
   E. None of the above

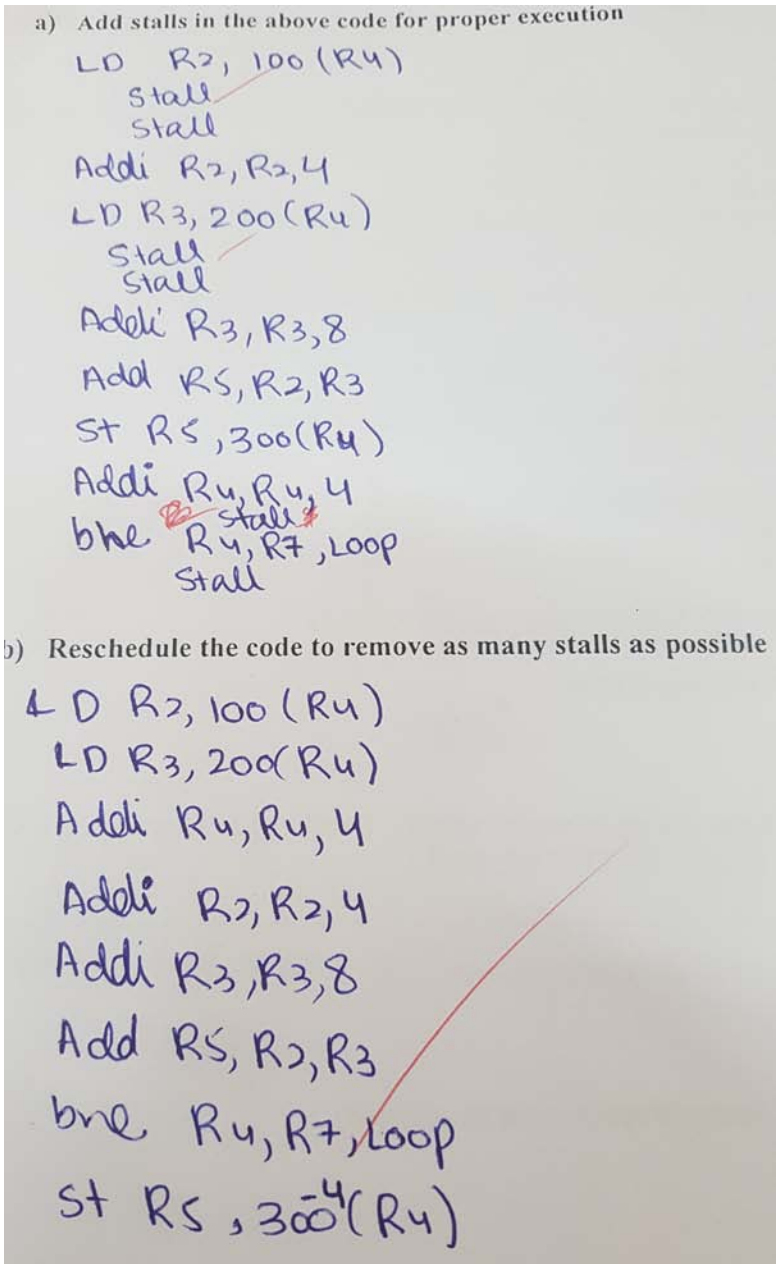
4. How many stall cycles will be introduced during execution of the following code fragment on a 5-stage pipelined datapath without forwarding?
   ADD R3, R3, R1
   SUB R3, R1,R2
   SW R3, 2 (R1)
   A. 0
   B. 1
   C. 2
   D. 3
   E. 4

5. Structural hazards can be avoided by (Multiple options can be selected)
   A. adding stalls
   B. forwarding
   C. adding more hardware
   D. register renaming
   E. loop unrolling

Question 2[15]

Loop: LD R2, 100 (R4)
Addi R2, R2, 4
LD R3, 200 (R4)
Addi R3, R3, 8
Add R5, R2, R3
ST R5, 300 (R4)
Addi R4, R4, 4
bne R4, R7, Loop

Consider the following specification for our pipelined processor:
Five stages MIPS pipeline where all arithmetic type instructions consume 1-cycle in each stage (5-cycle latency) but Instructions with data memory operation (Load, Store) consumes 2 cycles in the memory stage (6-cycle latency).
Forwarding has been implemented



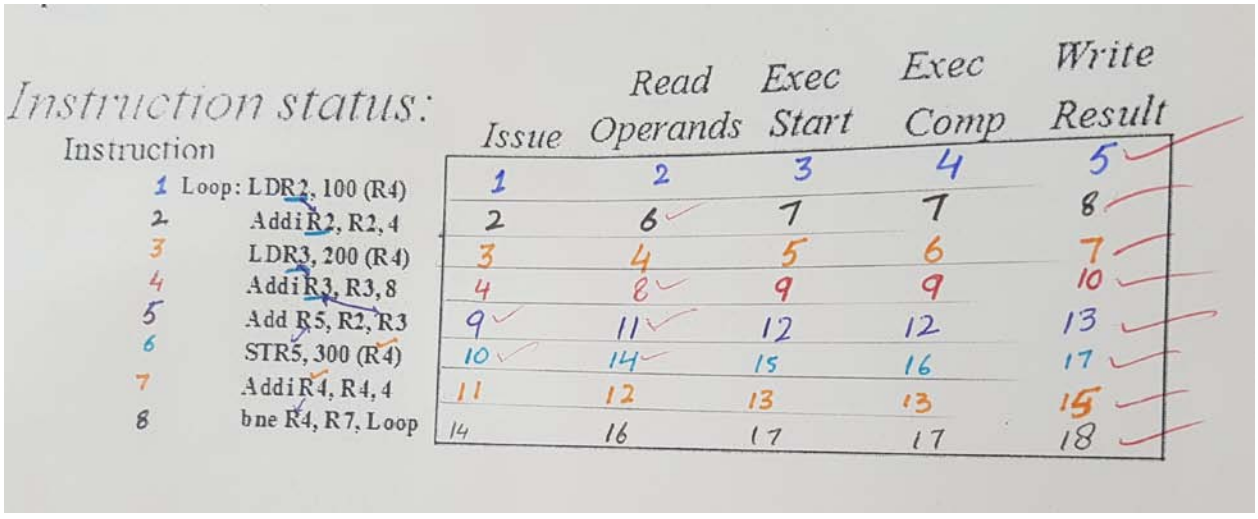
Question 3[15]

Loop: LD R2, 100 (R4)
Addi R2, R2, 4
LD R3, 200 (R4)
Addi R3, R3, 8
Add R5, R2, R3
ST R5, 300 (R4)
Addi R4, R4, 4
bne R4, R7, Loop

The above program is to be executed on a CPU with dynamic scheduling based on the SCOREBOARD technique with:

- 2 LOAD/STORE Functional Units (Load1, Load2) with latency 2 cycles
- 2 ALU/BR Functional Units (Add1, Add2) with latency 1 cycle
- Check structural hazards in ISSUE phase
- Check RAW hazards in READ OPERANDS phase
- Check WAR and WAW in WRITE BACK phase
- No forwarding
- Static branch prediction for backward branches: branch always taken

a) Show one iteration of this loop using SCOREBOARDING technique
b) If the processor clock cycle is 2 ns, calculate CPI and execution time for 1 iteration of this code.



Instruction status:

Instruction	Issue	Read Operands	Exec Start	Exec Comp	Write Result
1 Loop: LDR2, 100 (R4)	1	2	3	4	5
2 Addi R2, R2, 4	2	6	7	7	8
3 LDR3, 200 (R4)	3	4	5	6	7
4 Addi R3, R3, 8	4	8	9	9	10
5 Add R5, R2, R3	9	11	12	12	13
6 STR5, 300 (R4)	10	14	15	16	17
7 Addi R4, R4, 4	11	12	13	13	15
8 bne R4, R7, Loop	14	16	17	17	18

al unit status

ne Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
Load1	~~No~~ Yes(1) No(5) Yes No	Load store	R2 1	R4 R5	R4	~~Add1~~		Yes ~~No~~ Yes(13)	Yes
Load2	~~No~~ ~~Yes(9)~~ No(7)	Load	R3	R4				Yes	
Add1	~~No~~ Yes(2)	Addi	R2	R2		~~Load1~~		~~No~~ Yes(5)	
	No(8) Yes	Add	R5	R2	R3		Add2	Yes	~~No~~ Yes(10)
	No(13) Yes(14) No	Branch		R4	R7	~~Add2~~	[illegible]	~~No~~ Yes(15)	Yes
Add2	~~No~~ Yes(4)	Addi	R3	R3		~~Load2~~		~~No~~ Yes(7)	
	No(10) No	Addi	R4	R4				Yes	

Register result status:

R1	R2	R3	R4	R5	R6	R7	...
	Load1(1) ~~Add1(2)~~	~~Load2(3)~~	~~Add2~~ ~~Add2(4)~~	~~Add1~~			

Answer for part b:

No. of instructions = 8

No. of cycles = 18

CPU Clock cycle = 2 ns

→ $CPI_{(1\ iter)} = \frac{\text{No. of cycles}}{\text{No. of instructions}} = \frac{18}{8} = 2.25$ cycles/inst

→ Execution Time (1 iter) = IC × CPI × CPU clock cycle time

= 8 × 2.25 × 2ns

Exec Time (1 iter) = ~~36 ns~~

Question 4 [5]

Suppose the execution time of a program is 200 milliseconds. 120 milliseconds of this time is spent on multiplication, 40 milliseconds on addition and 40 milliseconds on load store instruction. How much do we have to improve the speed of multiplication if we want the program to run 2 times faster?

$$\text{Exec Time}_{New} = \text{Ex Time}_{unaffected} + \frac{\text{Ex Time}_{affected}}{Improvement}$$

$$= \left[\text{Exec time}_{Add} + \text{Exec time}_{LD-ST}\right] + \frac{\text{Exec time}_{MUL}}{Improvement}$$

$$\frac{200\ msec}{2} = 80\ msec + \frac{120\ msec}{N}$$

$$100\ msec = 80\ msec + \frac{120\ msec}{N}$$

$$N = \frac{120\ msec}{20\ msec}$$

$$N = 6\ times$$

$\Rightarrow$ If we want our program to run 2× faster, speed of multiplication should be improved by 6 times.