National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Advanced Computer Architecture	Course Code:	EE502
Program:	MS (Computer Science)	Semester:	Fall 2020
Duration:	90 Minutes	Total Marks:	30
Paper Date:	24-11-2020	Weight	15
Exam Type:	Midterm II	Page(s):	4

Student : Name:_____ Roll No._____

Instruction/Notes:

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 your assumption and solve accordingly.

Question 1 [3+3+4]

a) What is the effect of pipelining on latency of a single instruction and overall program throughput? Explain for each term, either it increases or decreases, provide reasoning for your answer!

b) Why we need loop unrolling? Suppose a loop has 20 iterations and we want to unroll it by degree 3. In this scenario how many iterations of new loop will be required and how the count of 20 will be completed? Explain your answer with details.

c) Why Tomasulo is better than scoreboarding? How it provides the extra functionality? What is the purpose of reorder buffers in Tomasulo approach? Explain with details!

Question 2 [4+2+2+2]

a. While writing in memory, contents can be brought to cache or not depending on the policy being used. Suppose we use **Allocate on write** policy, if following addresses are accessed in the given sequence then for each address tell either it would be hit or miss. Assume cache is initially empty and block size is 16 bytes.

Address	Hit/ miss?
Load 0x1200	
Store 0x1000	
Load 0x1208	
Load 0x1012	
Load 0x1000	
Store 0x1216	

b. If the cache is 4 way set associative with 512 sets and 16 bytes block size, then what is the size of the cache?

c. If the total address bits are 16 then divide it into tag, index and offset bits?

d. What is the total size of the memory in bytes?

Question 3 [10]

Code	Instructions

FOR: ld R1,100(R6)
ld R2,200(R6)
add R5,R2,R1
ld R4,0(R5)
addi R6,R6,4
and R4,R5,R4
st R4,0(R5)
beq R6,R7, FOR
	○ ALU instructions take 1 cycle in each stage except Mem. No cycle is consumed in memory.
	○ Beq takes one cycle in each stage and calculation are performed in decode stage.
	○ Memory type instructions take 2 cycles in memory and one in all other stages.
	○ In case of false dependencies, write back should be in order.
	○ Instructions are fetched in order however decode, execution, memory and write back can be out of order if no dependency.
	○ Forwarding is implemented so values can be forwarded as soon as they are produced.

Show the execution of the above code in the following table having two slots in each stage (2-way superscalar). A slot can be filled in fetch and decode as soon as it is empty.

Cycle No.	IF		ID		EX		MEM		WB	
1										
2										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										
.										