Course Name:	Computer Architecture	Course Code:	EE204
Program:	BS(Computer Science)	Semester:	Fall 2020
		Total Marks:	25
Due Date:	21-12-2020	Weight	~2
Evaluation Type:	Assignment 6	Page(s):	2

Name:_____ Student Id:_____
Section :_____

Question 1 [8+3+7+7]

Consider the following program fragment executing on a MIPS pipeline *with multiple execution units.* Execution stage takes a variable number of cycles, depending on the functional unit used. Other than execution, all other stages have one cycle latency.

Functional unit	Number of EX cycles
Integer ALU	2
Floating Point Add	3
Floating Point Multiply	4

a) For the correct execution of the following program, inserts bubbles (stall cycles) if required into the pipeline to avoid structural, data and control hazards. Instructions can be executed and written back out of order if the circumstances allow. **Data forwarding is implemented**.

```
Loop: L.D F1, 0(R1)
      L.D F2, 8(R1)
      ADD.D F4, F1, F3
      MUL.D F1, F1, F2
      S.D F4, 16(R1)
      S.D F1, 24(R1)
      ADDI R1, R1, 8
      Bne R1, R2, Loop
```

Note: R registers are for integer and F registers are for floating. Integer ALU is used for integer operations and Floating point units are used if the operands are floating.

Show timing in the table below.

Cycle / Inst	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
(1)																														
(2)																														
(3)																														
(4)																														
(5)																														
(6)																														
(7)																														
(8)																														

b) What would be the average CPI and the execution time of this program on a processor with 4MHz frequency?

c) Reschedule the instruction to remove as many stalls as possible. Use loop unrolling if required to remove all stalls.

d) Consider the given program (used in part a) to be executed on 2-issue superscalar processor with following specification.
   - There are three execution units with latency mentioned in part a.
   - All instructions consumes one cycle in each stage except Execution stage where number of cycles depends on the execution unit being used.
   - Full Forwarding is implemented. Branch operation is performed in decode stage.
   - Instructions can execute and write back out of order if circumstances allow.
   - In case of false dependencies, write back should be in order. No need to wait in decode stage.

Cycle No.	IF		ID		EX			MEM		WB	
					IADD	FADD	FMUL				
1											
2											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											
.											