Examination example -- Computer Architectures

Name, Matricola ....

Question 1

Considering the MIPS64 architecture presented in the following:

- Integer ALU: 1 clock cycle
- Data memory: 1 clock cycle
- FP multiplier unit: pipelined 8 stages
- FP arithmetic unit: pipelined 2 stages
- FP divider unit: not pipelined unit that requires 8 clock cycles
- branch delay slot: 1 clock cycle, and the branch delay slot is not enable
- forwarding is enabled
- it is possible to complete instruction EXE stage in an out-of-order fashion.

○ and using the following code fragment, show the timing of the presented loop-based program and compute how many cycles does this program take to execute?

```
; ******************** MIPS64 **********************
; for (i = 0; i < 100; i++) {
;    v7[i] = (v1[i]/v2[i])*v3[i] + (v4[i]*v5[i]*v6[i]);
;}
```

```
                .data
V1:     .double "100 values"
V2:     .double "100 values"
V3:     .double "100 values"
  …
V7:     .double "100 zeroes"

                .text

main:   daddui r1,r0,0
        daddui r2,r0,100
loop:   l.d  f1,v1(r1)
        l.d  f2,v2(r1)
        div.d  f7,f1,f2
        l.d  f3,v3(r1)
        mul.d  f8,f7,f3
        l.d  f4,v4(r1)
        l.d  f5,v5(r1)
        mul.d  f9,f4,f5
        l.d  f6,v6(r1)
        mul.d  f10,f9,f6
        add.d  f11,f8,f10
        s.d  f11,v7(r1)
        daddi  r2,r2,-1
        daddui  r1,r1,8
        bnez  r2,loop

        halt
```

comments	Clock cycles
r1 <= puntatore	
r2 <= 100	
f1 <= v1[i]	
f2 <= v2[i]	
f7 <= v1[i]/v2[i]	
f3 <= v3[i]	
f8 <= (v1[i]/v2[i])*v3[i]	
f4 <= v4[i]	
f5 <= v5[i]	
f9 <= v4[i]*v5[i]	
f6 <= v6[i]	
f10 <= v4[i]*v5[i]*v6[i]	
f11 <= f8+f10	
r2 <= r2 - 1	
r1 <= r1 + 8	

total

Examination example -- Computer Architectures

Name, Matricola ....

Question 2

Using the static scheduling technique and enabling the *Branch Delay Slot*, re-schedule the presented code in order to eliminate the most data hazards. Compute the number of clock cycles necessary to execute the new program.

Examination example -- Computer Architectures

Name, Matricola ....

Question 3

Considering the same loop-based program, and assuming the following processor architecture for a superscalar MIPS64 processor implemented with multiple-issue and speculation:

- issue 2 instructions per clock cycle
- jump instructions require 1 issue
- handle 2 instructions commit per clock cycle
- timing facts for the following separate functional units:
  - i. 1 Memory address 1 clock cycle
  - ii. 1 Integer ALU 1 clock cycle
  - iii. 1 Jump unit 1 clock cycle
  - iv. 1 FP multiplier unit, which is pipelined: 8 stages
  - v. 1 FP divider unit, which is not pipelined: 8 clock cycles
  - vi. 1 FP Arithmetic unit, which is pipelined: 2 stages
- Branch prediction is always correct
- There are no cache misses
- There are 2 CDB (Common Data Bus).

○ Complete the table reported below showing the processor behavior for the 2 initial iterations.

○

# iteration		Issue	EXE	MEM	CDB x2	COMMIT x2
1	l.d f1,v1(r1)					
1	l.d f2,v2(r1)					
1	div.d f7,f1,f2					
1	l.d f3,v3(r1)					
1	mul.d f8,f7,f3					
1	l.d f4,v4(r1)					
1	l.d f5,v5(r1)					
1	mul.d f9,f4,f5					
1	l.d f6,v6(r1)					
1	mul.d f10,f9,f6					
1	add.d f11,f8,f10					
1	s.d f11,v7(r1)					
1	daddi r2,r2,-1					
1	daddui r1,r1,8					
1	bnez r2,loop					
2	l.d f1,v1(r1)					
2	l.d f2,v2(r1)					
2	div.d f7,f1,f2					
2	l.d f3,v3(r1)					
2	mul.d f8,f7,f3					
2	l.d f4,v4(r1)					
2	l.d f5,v5(r1)					
2	mul.d f9,f4,f5					
2	l.d f6,v6(r1)					
2	mul.d f10,f9,f6					
2	add.d f11,f8,f10					
2	s.d f11,v7(r1)					
2	daddi r2,r2,-1					
2	daddui r1,r1,8					
2	bnez r2,loop					