9 September 2011 –- Computer Architectures –- part 2/2

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 6 stages
- FP arithmetic unit: pipelined 2 stages
- FP divider unit: not pipelined unit that requires 10 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.

o 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++) {
;          v4[i] = v1[i]/v2[i]
;          v5[i] = v3[i]*v4[i]
;          v6[i] = v4[i]+v5[i]
;}
```

	comments	Clock cycles
.data		
V1: .double "100 values"		
V2: .double "100 values"		
…		
V5: .double "100 zeroes"		
V6: .double "100 zeroes"		
.text		
main: daddui r1,r0,0	r1← pointer	
daddui r2,r0,100	r2 <= 100	
loop: l.d f1,v1(r1)	f1← v1[i]	
l.d f2,v2(r1)	f2← v2[i]	
div.d f4,f1,f2	f4 ← v1[i]/v2[i]	
s.d f4,v4(r1)	v4[i] ← f3	
l.d f3,v3(r1)	f3← v3[i]	
mul.d f5,f3,f4	f5 ← v3[i]*v4[i]	
s.d f5,v5(r1)	v5[i] ← f5	
daddi r2,r2,-1	r2 ← r2 - 1	
add.d f6,f4,f5	f6 ← v4[i]+v5[i]	
s.d f6,v6(r1)	v6[i] ← f6	
daddui r1,r1,8	r1 ← r1 + 8	
bnez r2,loop		
halt		
total		

9 September 2011 -- Computer Architectures –- part 2/2

Name, Matricola ....…..................……...

Question 2

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: 6 stages
    v. 1 FP divider unit, which is not pipelined: 10 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).

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

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