31 January 2012 -- 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 8 stages
- FP arithmetic unit: pipelined 4 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] = v1[i]+v2[i]+(v1[i]*v3[i]);
;}

               .data
V1:     .double “100 values”
V2:     .double “100 values”
V3:     .double “100 values”
V4:     .double “100 zeros”
V5:     .double “100 zeros”

               .text

main:   daddui r1,r0,0
        daddui r2,r0,100
loop:   l.d  f1,v1(r1)
        l.d  f2,v2(r1)
        l.d  f3,v3(r1)
        mul.d f4,f1,f2
        s.d f4,v4(r1)
        add.d f5,f1,f2
        mul.d f2,f1,f3
        add.d f1,f5,f2
        s.d f1,v5(r1)
        daddui r1,r1,8
        daddi r2,r2,-1
        bnez r2,loop
        Halt
```

	comments	Clock cycles
	r1← pointer	
	r2 <= 100	
	f1 <= v1[i]	
	f2 <= v2[i]	
	f3 <= v3[i]	
	f4 ← v1[i]*v2[i]	
	v4[i] ← f4	
	f5 ← v1[i]+v2[i]	
	f2 ← v1[i]*v3[i]	
	f1 ← v1[i]+v2[i]+v1[i]*v3[i]	
	v5[i] ← f1	
	r1 ← r1 + 8	
	r2 ← r2 - 1	
Total		

31 January 2012 -- 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: 8 stages
  v. 1 FP divider unit, which is not pipelined: 10 clock cycles
  vi. 1 FP Arithmetic unit, which is pipelined: 4 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	l.d f3,v3(r1)					
1	mul.d f4,f1,f2					
1	s.d f4,v4(r1)					
1	add.d f5,f1,f2					
1	mul.d f2,f1,f3					
1	add.d f1,f5,f2					
1	s.d f1,v5(r1)					
1	daddui r1,r1,8					
1	daddi r2,r2,-1					
1	bnez r2,loop					
2	l.d f1,v1(r1)					
2	l.d f2,v2(r1)					
2	l.d f3,v3(r1)					
2	mul.d f4,f1,f2					
2	s.d f4,v4(r1)					
2	add.d f5,f1,f2					
2	mul.d f2,f1,f3					
2	add.d f1,f5,f2					
2	s.d f1,v5(r1)					
2	daddui r1,r1,8					
2	daddi r2,r2,-1					
2	bnez r2,loop					