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.

○ 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	5
daddui r2,r0,100	r2 <= 100	1
loop: l.d f1,v1(r1)	f1← v1[i]	1
l.d f2,v2(r1)	f2← v2[i]	1
div.d f4,f1,f2	f4 ← v1[i]/v2[i]	11
s.d f4,v4(r1)	v4[i] ← f3	1
l.d f3,v3(r1)	f3← v3[i]	1
mul.d f5,f3,f4	f5 ← v3[i]*v4[i]	7
s.d f5,v5(r1)	v5[i] ← f5	1
daddi r2,r2,-1	r2 ← r2 - 1	1
add.d f6,f4,f5	f6 ← v4[i]+v5[i]	2
s.d f6,v6(r1)	v6[i] ← f6	1
daddui r1,r1,8	r1 ← r1 + 8	1
bnez r2,loop		1
halt		1
total		**3006**

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).

○ 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	2m	3	4	5
1	l.d f2,v2(r1)	1	3m	4	5	6
1	div.d f4,f1,f2	2	6d		16	17
1	s.d f4,v4(r1)	2	4m			17
1	l.d f3,v3(r1)	3	5m	6	7	18
1	mul.d f5,f3,f4	3	17x		23	24
1	s.d f5,v5(r1)	4	6m			24
1	daddi r2,r2,-1	4	5i		6	25
1	add.d f6,f4,f5	5	24a		26	27
1	s.d f6,v6(r1)	5	7m			27
1	daddui r1,r1,8	6	7i		8	28
1	bnez r2,loop	7	8j			28
2	l.d f1,v1(r1)	8	9m	10	11	29
2	l.d f2,v2(r1)	8	10m	11	12	29
2	div.d f4,f1,f2	9	16d		26	30
2	s.d f4,v4(r1)	9	11m			30
2	l.d f3,v3(r1)	10	12m	13	14	31
2	mul.d f5,f3,f4	10	27x		33	34
2	s.d f5,v5(r1)	11	13m			34
2	daddi r2,r2,-1	11	12i		13	35
2	add.d f6,f4,f5	12	34a		36	37
2	s.d f6,v6(r1)	12	14m			37
2	daddui r1,r1,8	13	14i		15	38
2	bnez r2,loop	14	15j			38