

# 25 January 2016 – 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.

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

```
        .data
v1:     .double "100 values"
v2:     .double "100 values"
v3:     .double "100 values"
v4:     .double "100 values"
v5:     .double "100 zeros"
v6:     .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)
        l.d f4,v4(r1)
        div.d f5,f1,f2
        s.d f5,v5(r1)
        mul.d f5,f1,f2
        div.d f6,f3,f4
        add.d f1,f5,f6
        s.d f1,v6(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 ≤ v4[i]	
f5 ← v1[i]/v2[i]	
v5[i] ← f5	
f5 ← v1[i]*v2[i]	
f6 ← v3[i]/v4[i]	
f1 ← f5 + f6	
v6[i] ← f1	
r1 ← r1 + 8	
r2 ← r2 - 1	
Total	

## 25 January 2016 – 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	l.d f4,v4(r1)					
1	div.d f5,f1,f2					
1	s.d f5,v5(r1)					
1	mul.d f5,f1,f2					
1	div.d f6,f3,f4					
1	add.d f1,f5,f6					
1	s.d f1,v6(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	l.d f4,v4(r1)					
2	div.d f5,f1,f2					
2	s.d f5,v5(r1)					
2	mul.d f5,f1,f2					
2	div.d f6,f3,f4					
2	add.d f1,f5,f6					
2	s.d f1,v6(r1)					
2	daddui r1,r1,8					
2	daddi r2,r2,-1					
2	bnez r2,loop					