

# 4 July 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 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++) {
;          v5[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 zeroes"
```

```
.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 f6,f3,f4
      div.d f7,f1,f2
      mul.d f5,f6,f7
      s.d f5,v5(r1)
      daddui r1,r1,8
      addi r2,r2,-1
      bneq r2,loop
      halt
```

comments	Clock cycles
r1 ← pointer	5
r2 ≤ 100	1
f1 ≤ v1[i]	1
f2 ≤ v2[i]	1
f3 ≤ v3[i]	1
f4 ≤ v4[i]	1
f6 ≤ v3[i]/v4[i]	9
f7 ≤ v1[i]/v2[i]	8
f5 ≤ f6*f7	8
	1
r1 ≤ r1 + 8	1
r2 ≤ r2 - 1	1
	2
	1
total	3506

5) out of loop = 6

$$\Rightarrow 6 + 38 \cdot 100 = \underline{\hspace{2cm}3806}$$

Exam 4 July 2011

daddi	F D E M W	5
daddi	F D E M W	1
l.d	F D E M W	1
l.d	F D E M W	1
l.d	F D E M W	1
l.d	F D E M W	1
l.d	F D E M W	1
div	F D S d d d d d d d d d d N W	9
div	F S D S S S S S S S d d d d d d d d M W	8
mul	F S S S S S S S D S S S S S S m m m m m m m m m m M W	8
s.d	F S S S S S S S D S S S S S S S S E M W	1
daddi	F S S S S S S S D E M W	1
daddi	F D E M W	1
bnez	F S D E M W	2
halt	F D E M W	1

5 ) out of loop  $\Rightarrow$  6

$$\begin{array}{r}
 & & 5 \\
 & & | \\
 & & 4 \\
 & & | \\
 & & 29 \\
 & & | \\
 9 & ) & 25 \\
 8 & ) & \\
 8 & ) & \\
 & & 3 \\
 & & | \\
 & & 6 \\
 2 & ) & 3 \\
 1 & ) &
 \end{array}$$

$$\Rightarrow (35 \cdot 100) + 6 = \underline{\underline{3506}} \quad \checkmark$$

	A 2	M 8	D 8		
11 . . . . .	6	1	. . . . .	10 . . . . .	. . . . . 20 . . . . . 30 . . . . . 35
deddui	F D E M W				Σ
deddui	F D E M W				1
ld f1	F D E M W				1
ld f2	F D E M W				1
ld f3	F D E M W				1
ld f4	F D E M W				1
div f6 = f3, f4	F D (S d d d d d d d) M W				9
div f7 = f1, f2	F D (S S S S S S S d d d d d d) M W				8
mul f5 = f6, f7	F (D)(S S S S S S) (S S S S S S S m m m m m m m m m m) M W				8
sd f5	(F (D) S S S S S S) (D) (S S S S S S S E) M W				1
deddui r1	(F S S S S S S (F S S S S S S S S S S S S S S S S) E M W				1
deddi r2	(F S S S S S S S S S D E M W				1
bmez r2	F (O S) E M W				2
halt	(F S) O E M W				1

3506 ✓

	1 . . . . 6   1 . . . . . . . . . . . .	15 . . . . . . . . . . . . . . . . . .	30 . . . . . . . . . . . . . . . . . .	35
daddi	F D E M W			S
daddi	F D E M W			1
ld f1	F D E M W			1
ld f2	F D E M W			1
ld f3	F D E M W			1
ld f4	F D E M W			1
div f6 = f3, f4	F D (S J d d d d d d d) M W			9
div f7 = f1, f2	F (D S S S S S S S) d d d d d d d m w			8
mul f5 = f6, f7	(F S S S S S S S) D (S S S S S S S m m m m m m m m m) M W			8
sd f5	F (D S S S S S S) (S S S S S S E) M W			1
daddi r1	(F S S S S S S) (D S S S S S S E) M W			1
deddi r2	(F S S S S S S) D E M W			1
bne r2	F (D S) E M W			2
halt	F X X X X X			1

3506

# 4 July 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: 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	2 m	3	4	5
1	l.d f2,v2(r1)	1	3 m	4	5	6
1	l.d f3,v3(r1)	2	4 m	5	6	7
1	l.d f4,v4(r1)	2	5 m	6	7	8
1	div.d f6,f3,f4	3	8 d		16	17
1	div.d f7,f1,f2	3	<del>15d</del> → ①		24	25
1	mul.d f5,f6,f7	4	25 x		33	34
1	s.d f5,v5(r1)	4	6 m			34
1	daddui r1,r1,8	5	6 i		7	35
1	daddi r2,r2,-1	5	7 i		8	35
1	bnez r2,loop	6	9 d			36
2	l.d f1,v1(r1)	7	8 m	9	10	36
2	l.d f2,v2(r1)	8	9 m	10	11	37
2	l.d f3,v3(r1)	8	10 m	11	12	37
2	l.d f4,v4(r1)	9	11 m	12	13	38
2	div.d f6,f3,f4	9	24 d		32	38
2	div.d f7,f1,f2	10	32 d		40	41
2	mul.d f5,f6,f7	10	41 x		39	41
2	s.d f5,v5(r1)	11	12 m			42
2	daddui r1,r1,8	11	12 i		13	42
2	daddi r2,r2,-1	12	13 i		14	43
2	bnez r2,loop	13	15 i			43

① ERRORE! Sto dato per scattato la seconda div  
partire DOPPO le precedente: invece parte PRIMA  
al clock 14. Correzione pag successiva.

# 4 July 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: 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	I.d f1,v1(r1)	1	2 m	3	4	5
1	I.d f2,v2(r1)	1	3 m	4	5	6
1	I.d f3,v3(r1)	2	4 m	5	6	7
1	I.d f4,v4(r1)	2	5 m	6	7	8
1	div.d f6,f3,f4	3	8d 14d	16 22	17 23	
1	div.d f7,f1,f2	3	1* 6d	14	15 23	
1	mul.d f5,f6,f7	4	23 x	31	32	
1	s.d f5,v5(r1)	4	6 m			32
1	daddui r1,r1,8	5	6 i		7	33
1	daddi r2,r2,-1	5	7 i	3*	8	33
1	bnez r2,loop	6	9 j			34
2	I.d f1,v1(r1)	7	8 m	9	10	34
2	I.d f2,v2(r1)	8	9 m	10	11	35
2	I.d f3,v3(r1)	8	10 m	12	13	35
2	I.d f4,v4(r1)	9	11 m	14	15	36
2	div.d f6,f3,f4	9	16d 20d	24 28	36	
2	div.d f7,f1,f2	10	2* 12d	20	27	
2	mul.d f5,f6,f7	10	29 x	37	38	
2	s.d f5,v5(r1)	11	12 m			38
2	daddui r1,r1,8	11	12 i		13	39
2	daddi r2,r2,-1	12	13 i		14	39
2	bnez r2,loop	13	15 j			40

1\* Si tratta prima della precedente DIV, quindi  
bisogna poi aggiornare le precedenti per "allinearle".

2\* Anche qui, vale lo stesso discorso fatto qui sopra.

3\* Anche questo è uno degli errori più comuni  
che si riscontrano nei temi d'esame.

# 4 July 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: 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	2 m	3	4	5
1	l.d f2,v2(r1)	1	3 m	4	5	6
1	l.d f3,v3(r1)	2	4 m	5	6	7
1	l.d f4,v4(r1)	2	5 m	6	7	8
1	div.d f6,f3,f4	3	8d 14d	16 22		17 23
1	div.d f7,f1,f2	3	6d !	14		23
1	mul.d f5,f6,f7	4	23x	31		32
1	s.d f5,v5(r1)	4	6m			32
1	daddui r1,r1,8	5	6i		7	33
1	daddi r2,r2,-1	5	7i		8	33
1	bnez r2,loop	6	9i			34
2	l.d f1,v1(r1)	7	8m	9	10	34
2	l.d f2,v2(r1)	7	9m	10	11	35
2	l.d f3,v3(r1)	8	10m	11	12	35
2	l.d f4,v4(r1)	8	11m	12	13	36
2	div.d f6,f3,f4	9	22d		30	36
2	div.d f7,f1,f2	9	30d		38	39
2	mul.d f5,f6,f7	10	33d		47	48
2	s.d f5,v5(r1)	10	12m			48
2	daddui r1,r1,8	11	12i		13	49
2	daddi r2,r2,-1	11	13i		14	49
2	bnez r2,loop	12	15i			50