

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?

dedd.u	F D E M W	S
deddu	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 d d d d d d d d d N W	11
s.d	F D s s s s s s s s E M W	
l.d	F s s s s s s s s D E M W	1
mul.d	F D m m m m m m m M W	
s.d	F D s s s s s s s E M W	1
deddi	F s s s s s D E M W	2
odd.d	F n Q Q M W	1
s.d	F D S E M W	1
deddu	F S D E M W	1
bnez	F D S E M W	1
halt	F X X X X X	

Il dato, dopo una l.d e
 disponibile dopo
 l'unità di Memory ⑥ => 7
 delle l.d stessa

deddwi	F D E M W
deddwi	F D E M W
l.d	F D E M W
l.d	F D E M W
div	F D S d d d d d d d d d M W
s.d	F S D S S S S S S S S E M W
l.d	F S S S S S S S S D E N W
mul.d	F D S m m m m m m m M W
s.d	F S D S S S S S S E M W
deddi	F S S S S S D E M W
add.d	F D Q Q M W
s.d	F D S E N W
deddwi	F S D E M W
bnez	F D S E M W
halt	F X X X X X

errore oh!
 d'introsione:
 la PNEN può fare la
 DECODE SUBITO, senza messo
 STALLO

daddi	F D E N W	5
daddi	F D E H W	1
f.d	F D E N W	1
f.d	F D E M W	1
div	F D S d d d d d d d d d M W	11
s.d	F S D S S S S S S S S E M W	1
f.d	F S S S S S S S S D E M W	1
mul.d	F D S m m m m m m m M W	7
s.d	F S D S S S S S E M W	1
daddi	F S S S S S D E M W	1
add.d	F D Q Q M W	2
s.d	F D S E H W	1
daddi	F S D E H W	1
bnez	F D E M W	1
halt	F X X X X	1

5 } out of loop

6

1) 13)

13

22

01

7)
1)

1

$$\begin{array}{r} 1 \\ 2 \\ \hline 3 \end{array} \quad \left. \begin{array}{r} 4 \\ \hline 7 \end{array} \right)$$

1

1) 3)

3

|) |) |

-

$$30 \times 100 = 3000$$

→ 3006

1 6 / 1 10 20 25 30

doddiui FDEMw

$\in \cap \in M \in W$

deddwi FDEM w

F D E M W

$\ell_0 \quad \ell_1 \quad F \in M_w$

$$F \in M_w$$

Qd Pz

$$F = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9) \in W$$

div f4

$D(S \cup d \cup d) \leq M$

5d t^4
71 72

(G 3 3 3 3 3 3 3 3 3 3 3 3) E M W
(E 3 3 3 3 3 3 3 3 3 3 3 3) D F N M

$$Cd \quad f_3$$

F D(S m m m m m m) M W

meek f3
e-d D5

$$F(0.555555) \in N_w$$

daddi r2

(F S S S S S S) D ∈ M W

odd $f_6 = f_4, f_5$

F D (Q Q) M W

sd f6

F D(s ∈) M w

Joddui tı

F(D S) E M W

bmet
1/1

$(P \wedge S) \rightarrow D \in M \wedge$

work

1

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