

Cognome e nome dello studente:

Matricola:

A.A. 2007-2008 – Appello del 24 Luglio 2008

[13] Data la CPU con pipeline nella pagina seguente:

- a) Definire da quali fasi è costituito il ciclo di esecuzione di un'istruzione e quando inizia l'esecuzione vera e propria [1].
- b) Che cosa caratterizza una CPU con pipeline e qual è la funzione di ogni stadio? Quali sono le informazioni di input perché ciascuno stadio funzioni correttamente e quali informazioni produce in output? Dove si trovano le informazioni di input e dove vengono scritte le informazioni di output di ogni stadio? [1]
- c) Dato il seguente segmento di codice:

```
ori $s0, $t1, 64
lw $t2, 0($t0)
sub $t2, $s2, $t2
addi $t3, $t3, 64
and $t5, $t5, $t6
```

Scrivere cosa viene scritto in **tutti** i registri di pipe-line (l'ingresso) quando l'istruzione and \$t5, \$t5, \$t6 si trova nella fase di fetch [5]. Non tenere conto degli hazard.

- d) Definire cos'è un hazard e cos'è uno stallo. Identificare se nel frammento di codice di cui sopra ci sono hazard. Eventualmente modificare la CPU in modo tale che riesca a gestire al meglio questi hazard [4]. Come vengono gestiti eccezioni ed interrupt dalla CPU [2]?

[1] Descrivere l'architettura globale di una moderna scheda grafica. Che tipo di parallelismo viene implementato? Cosa è CUDA?

[3] Cosa si intende per multiple issue statico e dinamico? Compararli e commentare.

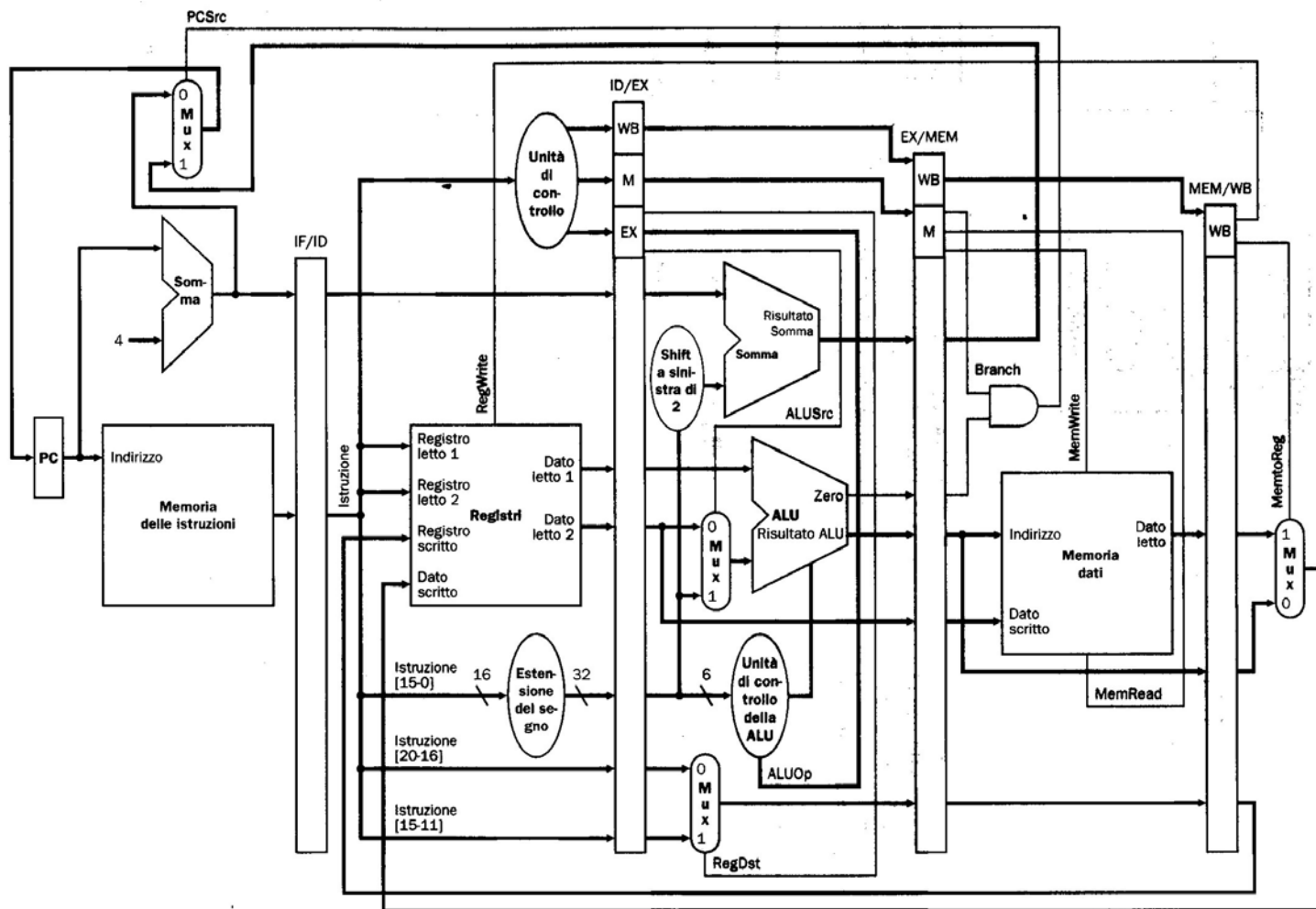
[5] Costruire un circuito firmware per la moltiplicazione di due parole di 8 bit. Disegnare un circuito che controlli se si verifica un overflow (risultato su un numero di bit maggiori di 8).

[3] Costruire un circuito logico in grado di effettuare l'AND e l'OR di quattro ingressi utilizzando porte a due bit.

[5] A cosa serve la memoria? Descrivere un disco rigido. Disegnare una cache a piacere associativa a 2 vie, il suo circuito di lettura e scrittura completo.

[3] Cos'è il refresh della memoria? Si applica alle memorie SRAM? E alle memorie DRAM? Motivare le risposte. Cosa si intende per write through e write back? Cosa si intende per coerenza (di una memoria)? Cosa si intende per memoria associativa?

[1] Definire almeno due metriche di valutazione delle prestazioni. Enunciare la legge di Amhdal.

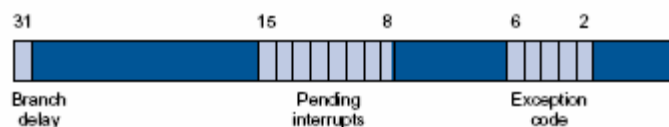


0	zero constant 0	16	s0 callee saves ... (caller can clobber)
1	at reserved for assembler	23	s7
2	v0 expression evaluation &	24	t8 temporary (cont'd)
3	v1 function results	25	t9
4	a0 arguments	26	k0 reserved for OS kernel
5	a1	27	k1
6	a2	28	gp Pointer to global area
7	a3	29	sp Stack pointer
8	t0 temporary: caller saves ... (callee can clobber)	30	fp frame pointer (s8)
15	t7	31	ra Return Address (HW)

Coprocessore 0

Nome del registro	Numero del registro in coprocessore 0	Utilizzo
Bad/Addr	8	Registro contenente l'indirizzo di memoria a cui si è fatto riferimento
Count	9	Timer
Compare	11	Valore da comparare con un timer. Genera un interrupt.
Status	12	Maschera delle interruzioni e bit di abilitazione. Stato dei diversi livelli di priorità (6 HW e 2 SW).
Cause	13	Tipo dell'interruzione e bit delle interruzioni pendenti
EPC	14	Registro contenente l'indirizzo dell'istruzione che ha causato l'interruzione.

Registro causa:



Registro stato:



Codici operativi

10	16	op(31:26)		rs (25:21)	(16:16)	funct (4:0)		rt (20:16)	10	funct(5:0)		10	funct(5:0)
0	00				0	0		0	0	sl		0	add.f
1	01				1	1		1	1	srl		1	sub.f
2	02	l			2	2		2	2	sra		2	mul.f
3	03	jal			3	3		3	3	sra		3	div.f
4	04	beq			4	4		4	4	sriv		4	abs.f
5	05	bne			5	5		5	5	srav		5	mov.f
6	06	blez			6	6		6	6	srav		6	neg.f
7	07	bgtz			7	7		7	7	jr		7	
8	08	addi			8	8		8	8	jalr		8	
9	09	addiu			9	9		9	9			9	
10	0a	sll			10	10		10	10	syscall		10	
11	0b	slliu			11	11		11	11	break		11	
12	0c	andi			12	12		12	12			12	
13	0d	ori			13	13		13	13			13	
14	0e	xori			14	14		14	14			14	
15	0f	lui			15	15		15	15			15	
16	10	$z = 0$			16	16		16	16	mthi		16	
17	11	$z = 1$			17	17		17	17	mthi		17	
18	12	$z = 2$			18	18		18	18	mto		18	
19	13	$z = 3$			19	19		19	19	mto		19	
20	14				20	20		20	20			20	
21	15				21	21		21	21			21	
22	16				22	22		22	22			22	
23	17				23	23		23	23			23	
24	18				24	24		24	24	mult		24	
25	19				25	25		25	25	multu		25	
26	1a				26	26		26	26	div		26	
27	1b				27	27		27	27	divu		27	
28	1c				28	28		28	28			28	
29	1d				29	29		29	29			29	
30	1e				30	30		30	30			30	
31	1f				31	31		31	31			31	
32	20			0	0	0		0	0	bltz		32	cvt.s.f
33	21	lb		1	1	1		1	1	bgez		33	cvt.d.f
34	22	lh		2	2	2		2	2			34	
35	23	lwl		3	3	3		3	3			35	
36	24	lw		4	4	4		4	4			36	cvt.w.f
37	25	lhu		5	5	5		5	5			37	
38	26	lhu		6	6	6		6	6			38	
39	27	lwr		7	7	7		7	7			39	
40	28			8	8	8		8	8			40	
41	29	sb		9	9	9		9	9			41	
42	2a	sh		10	10	10		10	10	sll		42	
43	2b	swl		11	11	11		11	11	slliu		43	
44	2c	sw		12	12	12		12	12			44	
45	2d			13	13	13		13	13			45	
46	2e			14	14	14		14	14			46	
47	2f	swr		15	15	15		15	15			47	
48	30			16	16	16		16	16	bitzal		48	c.f.f
49	31	lwc0		17	17	17		17	17	bgezal		49	c.un.f
50	32	lwc1		18	18	18		18	18			50	c.eq.f
51	33	lwc2		19	19	19		19	19			51	c.ueq.f
52	34	lwc3		20	20	20		20	20			52	c.olt.f
53	35			21	21	21		21	21			53	c.ult.f
54	36			22	22	22		22	22			54	c.ole.f
55	37			23	23	23		23	23			55	c.ule.f
56	38			24	24	24		24	24			56	c.st.f
57	39	swc0		25	25	25		25	25			57	c.ngle.f
58	3a	swc1		26	26	26		26	26			58	c.seq.f
59	3b	swc2		27	27	27		27	27			59	c.ngl.f
60	3c	swc3		28	28	28		28	28			60	c.lt.f
61	3d			29	29	29		29	29			61	c.nge.f
62	3e			30	30	30		30	30			62	c.le.f
63	3f			31	31	31		31	31			63	c.ngt.f

FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18, and 19. These operations are determined by other fields, identified by pointers. The last field (funct) uses “f” to mean “s” if rs = 16 and op = 17 or “d” if rs = 17 and op = 17. The second field (rs) uses “z” to mean “0”, “1”, “2”, or “3” if op = 16, 17, 18, or 19, respectively. If rs = 16, the operation is specified elsewhere: if z = 0, the operations are specified in the fourth field (bits 4 to 0); if z = 1, then the operations are in the last field with f = s. If rs = 17 and z = 1, then the operations are in the last field with f = d. (page A-54)