

Cognome e nome dello studente:

Matricola:

A.A. 2008-2009 – Appello del 24 Febbraio 2010

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

- Definire quando inizia l'esecuzione vera e propria di un'istruzione e da quali fasi è costituita [1].
- 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]
- Dato il seguente segmento di codice:

```
addi $s0, $t1, 32
```

```
lw $t2, 0($s6)
```

```
sub $t2, $s2, $t2
```

```
add $t3, $t3, $t3
```

```
or $t5, $t5, $t6
```

Scrivere il contenuto di **tutti** i registri di pipe-line (l'uscita) quando l'istruzione `or $t5, $t5, $t6` si trova nella fase di fetch [5].

- 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].

[11] Memoria: domande:

Cosa si intende per write through e write back? Cosa si intende per coerenza (di una memoria)? Cosa si intende per memoria associativa? [1]

Disegnare la struttura di DRAM sincrona di 1Kbyte. Dimensionare gli elementi. Cosa sono i segnali RAS e CAS? Perché sono stati introdotti? Come avviene la lettura e la scrittura della memoria? Come si fa ad aumentare il throughput di una memoria DRAM? [3].

Cos'è il refresh della memoria? Si applica alle memorie SRAM? E alle memorie DRAM? Motivare le risposte [2].

Disegnare una cache a mappatura diretta con linee di 256 byte e parole di 2 byte. Definire il ruolo dei vari bit di indirizzamento tenendo conto che l'indirizzo è assegnato su 64 bit. [5]

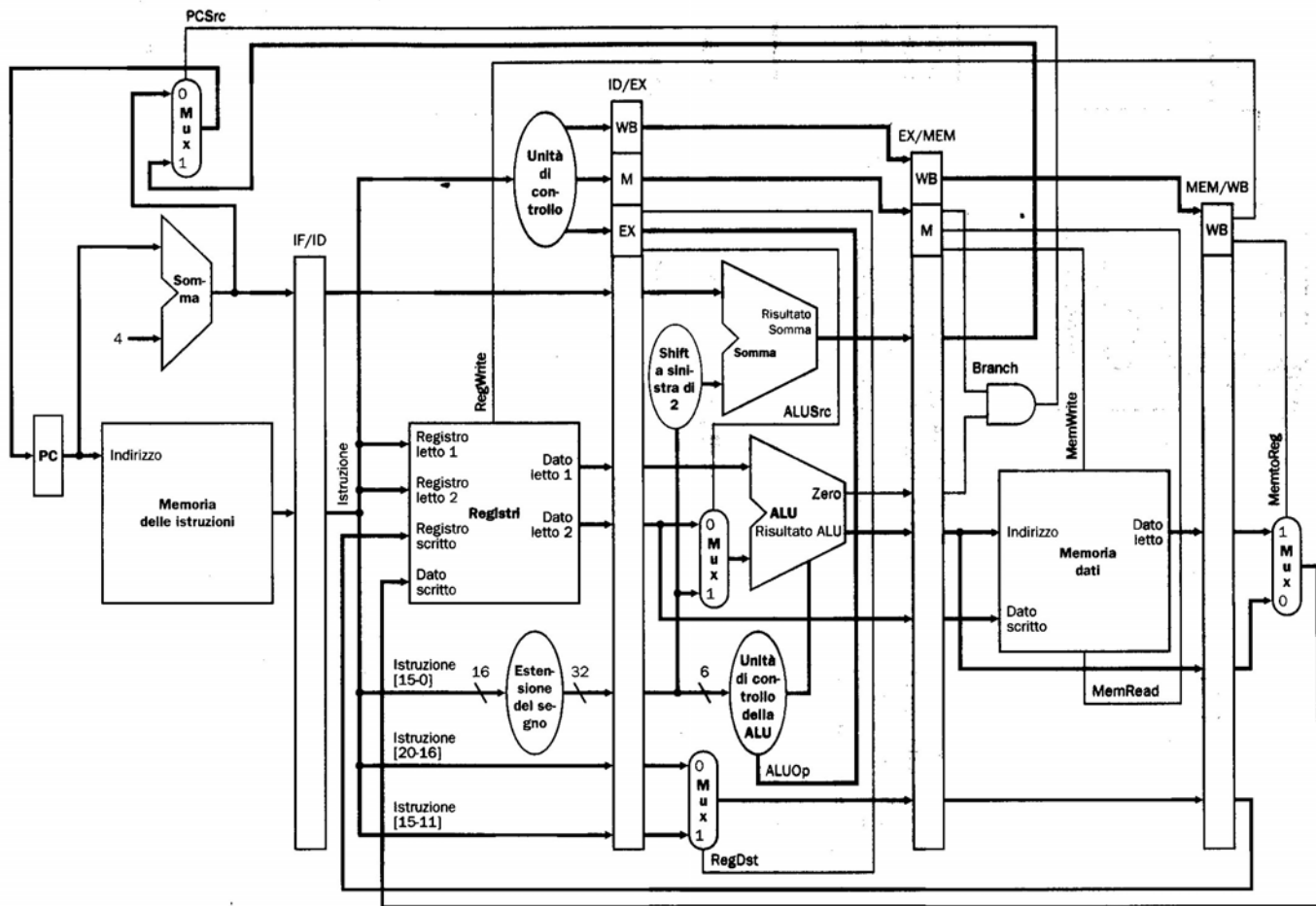
[4] Scrivere nelle due forme canoniche la seguente funzione logica: $Y = (B!C + A)D + !D$

[5] Definizione di interrupt ed eccezione e modalità di gestione da parte di una CPU Mips ed Intel. Scrivere un programma assembly che risponda ad una eccezione. Cosa deve fare? Quali registri deve considerare?

[2] Definire alcune metriche di valutazione delle prestazioni di un'architettura.

[3] Definire l'architettura a bus di un Intel di nuova generazione. Definire la procedura di trasferimento su bus asincrono. Cosa si intende per arbitraggio di un bus?

Codici operativi: `addi 8`, `lw 36`. Campi Funct: `add 32`, `sub 34`, `or 37`.

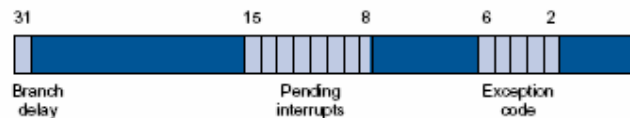


0	zero	constant 0	16	s0	callee saves
1	at	reserved for assembler	... (caller can clobber)		
2	v0	expression evaluation &	23	s7	
3	v1	function results	24	t8	temporary (cont'd)
4	a0	arguments	25	t9	
5	a1		26	k0	reserved for OS kernel
6	a2		27	k1	
7	a3		28	gp	Pointer to global area
8	t0	temporary: caller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

