Università degli Studi di Milano - Corso Architettura degli elaboratori e delle reti – Turno 1 – Prof. Borghese

Cognome e nome dello studente: ***Matricola:***

A.A. 2009-2010 – Appello del 10 Giugno 2010

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

a) Definire quando inizia l'esecuzione vera e propria di un'istruzione e da quali fasi è costituita [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? Come viene modificata per diventare una CPU multiple-issue statica? Come viene modificata per diventare una CPU multiple-issue dinamica? Cosa si intende per multiple-issue? [5]

c) Dato il seguente segmento di codice:

```
add $s0, $t1, $t0
lw $t2, 24($s6)
addi $t4, $s2, -4
add $t3, $t3, $t3
or $t2, $t2, $t3
```

Scrivere il contenuto di **tutti** i registri di pipe-line (l'uscita) quando l'istruzione or $t2, $t2, $t3 si trova nella fase di fetch [5].

d) Definire cosa sia un hazard e cosa sia uno stallo [2].

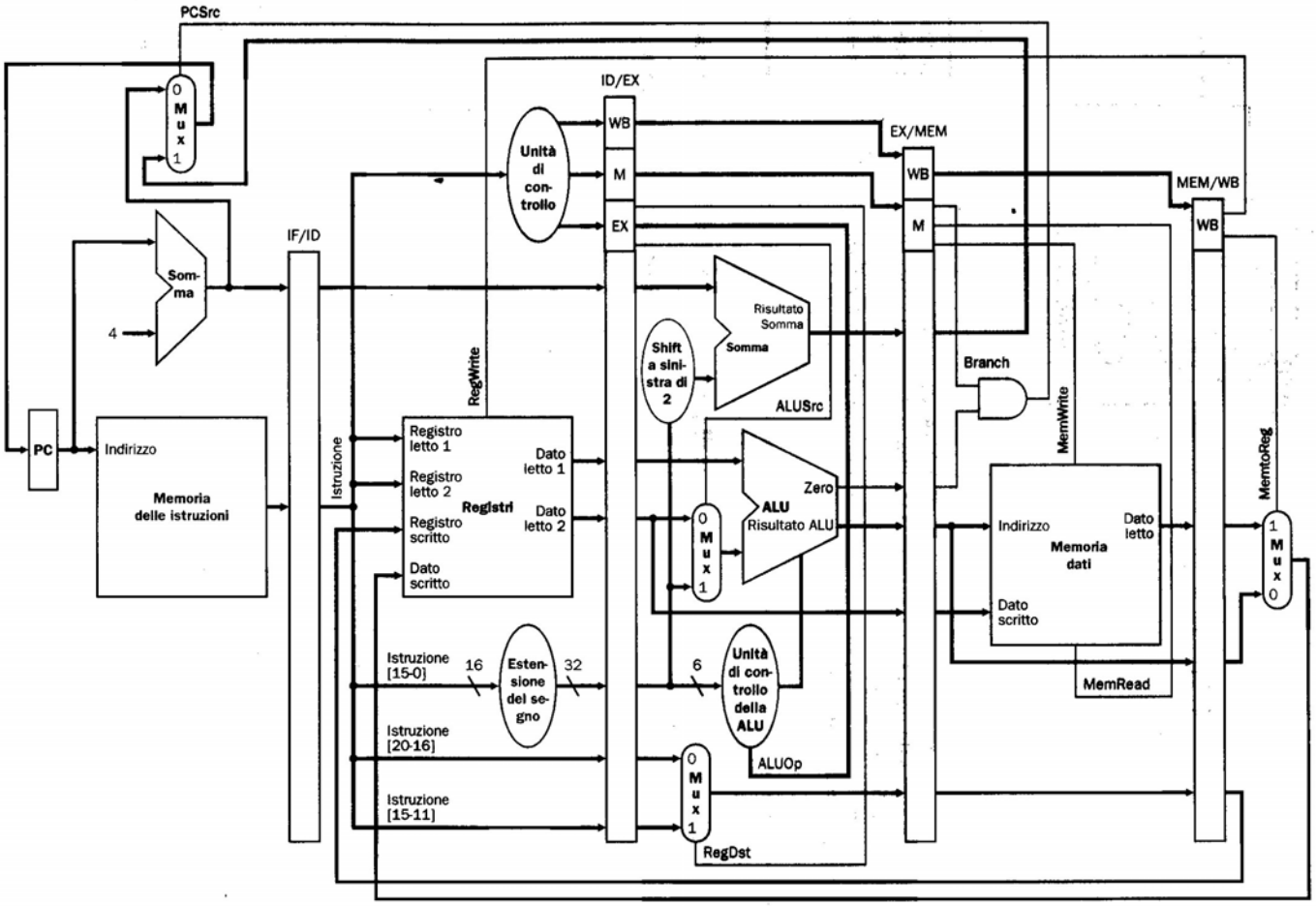
[6] Disegnare una cache a mappatura diretta di 8Kbyte con linee di 256 byte e parole di 2 byte. Definire il ruolo dei vari bit di indirizzamento tenendo conto che l'indirizzo e' assegnato su 32 bit. Disegnare la porta di lettura. Disegnare una cella di memoria SRAM e di memria DRAM ed illustrare .la differenza. Spiegare cosa si intende per modalità di trasferimento a burst, a chi si applica e perchè si utilizza e su quali proprietà delle architetture si basa [6].

[4] Scrivere nelle due forme canoniche la seguente funzione logica: Y = (B!C + A)D. Implementare la funzione utilizzando una PLA con 8 porte AND.

[5] Definizione di interrupt ed eccezione e modalità di gestione da parte di una CPU Mips ed Intel. Modificare la CPU riportata nel foglio successivo perchè sia in grado di gestirli.

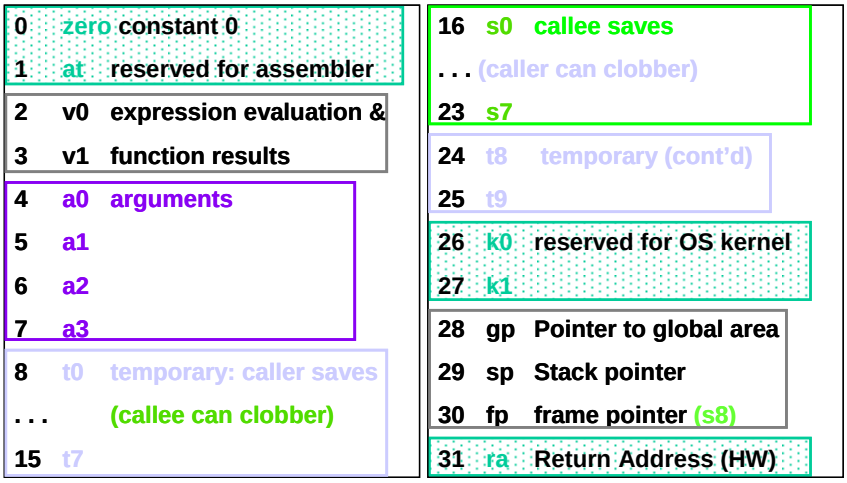
[1] Descrizione di un'architettura CUDA.

[5] Scrittura di un algoritmo di moltiplicazione e di un algoritmo di divisione binaria implementabile in firmware da un'architettura.



Codici operativi:
addi -> 8
lw -> 35
sw -> 43
j -> 2
jal -> 3
beq -> 4

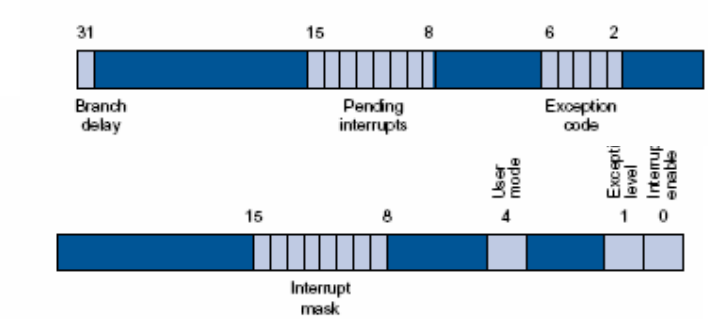
Campo funzione:
add -> 32
sub -> 34
or -> 37
and -> 36



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: