

Cognome e nome dello studente:

Matricola:

A.A. 2007-2008 – Appello del 25 Settembre 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 la CPU "capisce" di che istruzione si tratta. In quale fase termina l'esecuzione di un'istruzione? [2].
- b) Discutere le differenze tra la CPU singolo ciclo, multi-ciclo e con pipe-line. [2]
- c) Dato il seguente segmento di codice:

```
ori $s0, $t1, 64
lw $t3, 16($t0)
sub $t2, $s2, $t3
addi $t3, $t3, 64
and $s1, $s5, $s6
```

Scrivere qual è il contenuto di **tutti** i registri di pipe-line (lo stato) quando l'istruzione and \$s1, \$s5, \$s6 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].

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

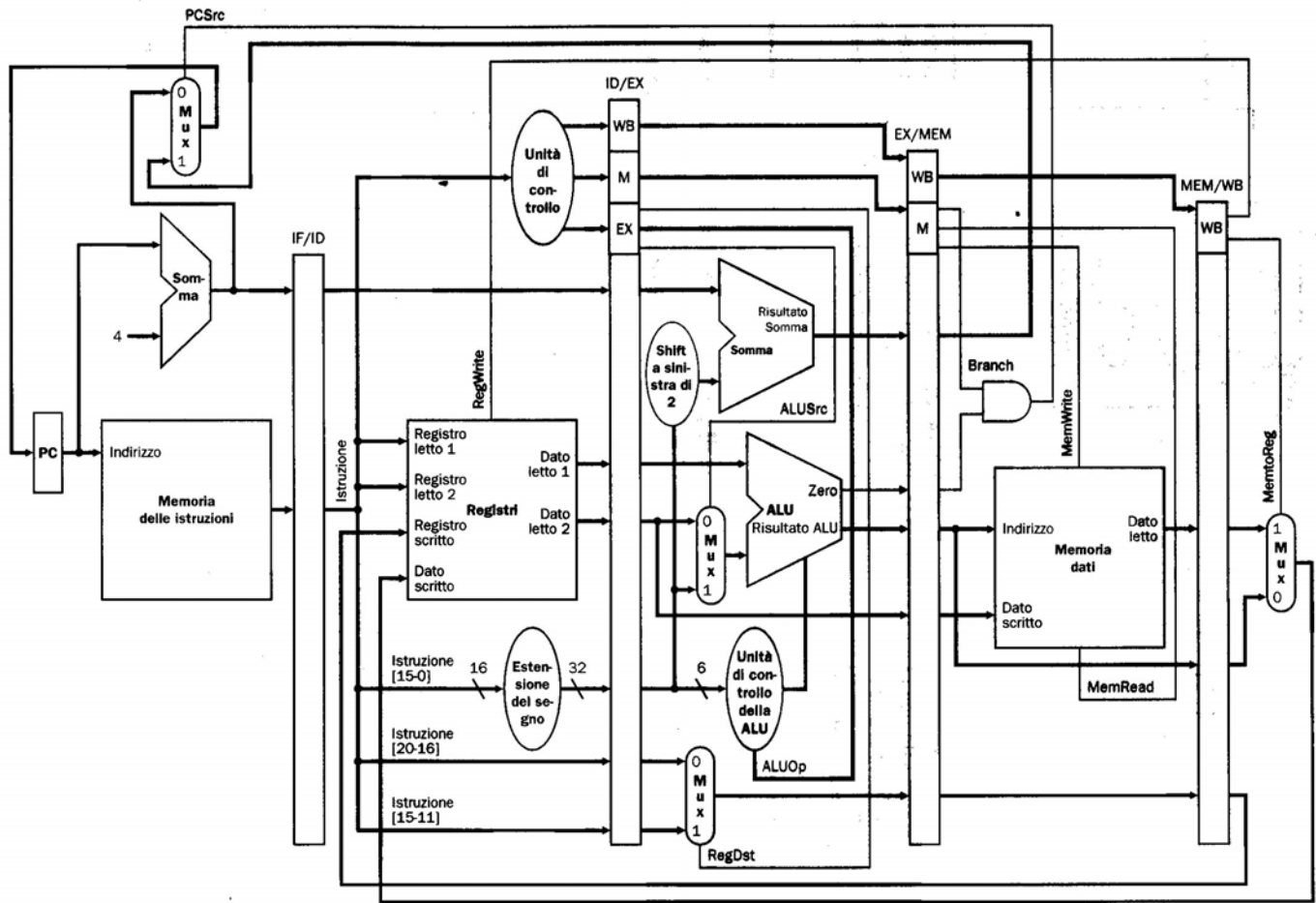
[5] Costruire un circuito firmware per la moltiplicazione di due parole di 8 bit. Disegnare un circuito che controlli se si verifica un overflow (risultato su un numero di bit maggiori di 8).

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

[7] Progettare una macchina a stati finiti di Moore che implementa un contatore da -2 a +2. Il contatore riceve 3 ingressi binari: "INCR", che quando va a "1" incrementa il contatore di 1, "DECR" che lo decrementa di 1, e "RESET" che lo azzerà. Quando il contatore raggiunge il numero +2, i successivi incrementi vengono ignorati, e lo stesso per i decrementi quando vale -2. Si supponga, per semplicità, che solo un ingresso alla volta possa andare a "1" per cui le configurazioni di ingresso possibile sono 000, 001, 010, 100. Il contatore presenta 2 uscite binarie: una va a "1" quando il contatore è arrivato a +2, l'altra quando è a -2. Determinare STG, STT, STT codificata e determinare le funzioni logiche di uscita e stato prossimo.

[5] Sintetizzare la funzione logical nella prima e seconda forma canonica associata alla seguente espressione logica. Dimostrare che le due forme canoniche sono equivalenti trasformando la prima forma canonica nella seconda:

$$Y = AC + A(!B) + B (!C)$$

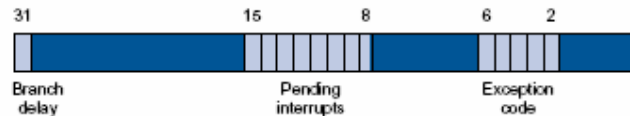


0	zero constant 0	16	s0 callee saves
1	at reserved for assembler	... (caller can clobber)	
2	v0 expression evaluation &	23	s7
3	v1 function results	24	t8 temporary (cont'd)
4	a0 arguments	25	t9
5	a1	26	k0 reserved for OS kernel
6	a2	27	k1
7	a3	28	gp Pointer to global area
8	t0 temporary: caller saves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



Codici operativi

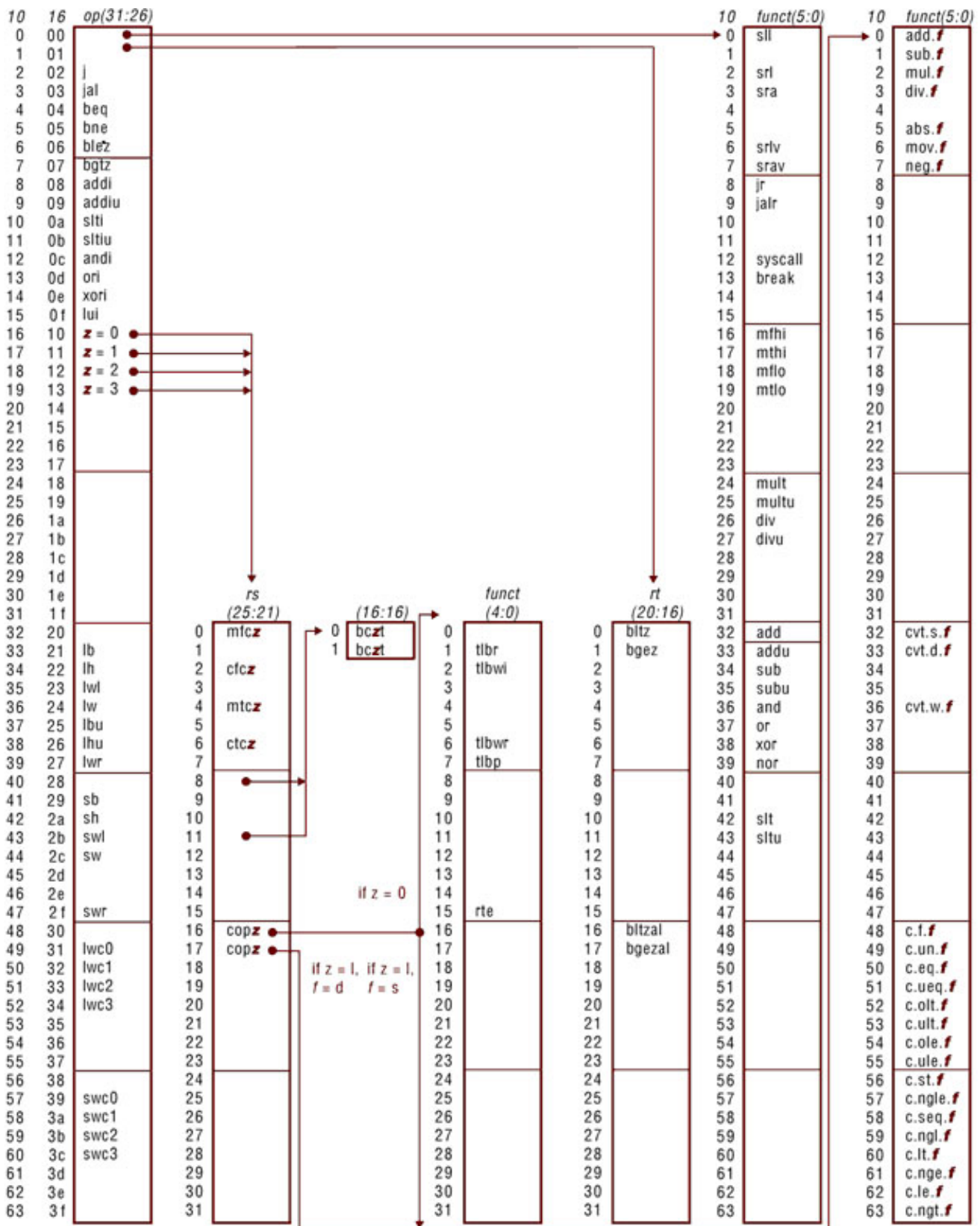


FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18, 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)