

Cognome e nome dello studente:

Matricola:

A.A. 2008-2009 – Appello del 18 Giugno 2009

Es. 1. [11] Consideriamo le seguenti coppie di istruzioni di linguaggio assembler MIPS:

- 1) add \$0, \$1, \$2
bne \$1, \$2, etichetta
- 2) lw \$2, 40(\$3)
and \$1, \$2, \$3

Su supponga che queste istruzioni facciano parte di un lungo programma.

- a) [1] Per ognuna di queste coppie definire quali hazard si verificano e perché. Determinare in quale stadio della pipeline l'hazard viene riconosciuto. Cosa si intende per hazard? Quali tipi di hazard conoscete?
- b) [4] Modificare la pipeline riportata sotto in modo che ciascuna delle due coppie sia eseguita correttamente.
- c) [3] Mostrare il contenuto dei registri di pipeline (la loro uscita), il cui contenuto può essere definito, al passo in cui l'hazard viene riconosciuto. Definire un indirizzo a piacere delle coppie di istruzioni, compatibile con la suddivisione logica della memoria MIPS.
- d) [2] Una delle conseguenze principali delle architetture a programma memorizzato è che le parole di memoria acquistano un significato nel momento in cui vengono lette o scritte: possono essere istruzioni, numeri interi, in virgola mobile, carattere... Uno dei problemi generati da ciò è la diversa codifica delle parole. Cosa si intende per big endian e little endian? Mostrare un esempio a piacere di ciò.
- e) [1] Cosa si intende per pipeline superscalare? E per superpipeline? Che tipo di pipeline è quella mostrata in figura?

Es. 2. [7] In una memoria cache a mappatura diretta, gli indirizzi su 32 bit possono essere suddivisi come segue:

- Tag: 31-12
- Indice 11-5
- Offset 4-0

[3] Determinare il numero di linee, la dimensione totale e la dimensione della linea della cache in byte, supponendo parole di 32 bit. Quali di queste quantità cambierebbe se la cache fosse in realtà a 4 vie?

[1] Quanti blocchi di RAM possono essere mappati in questa cache, supponendo una RAM di 1 Gbyte? Quale sarebbe il numero massimo di blocchi di RAM che possono essere mappati?

[1] Cosa si intende per hit e miss?

[2] Cosa si intende per writeback e write-through? Quali sono i vantaggi e svantaggi di ciascuna delle due modalità?

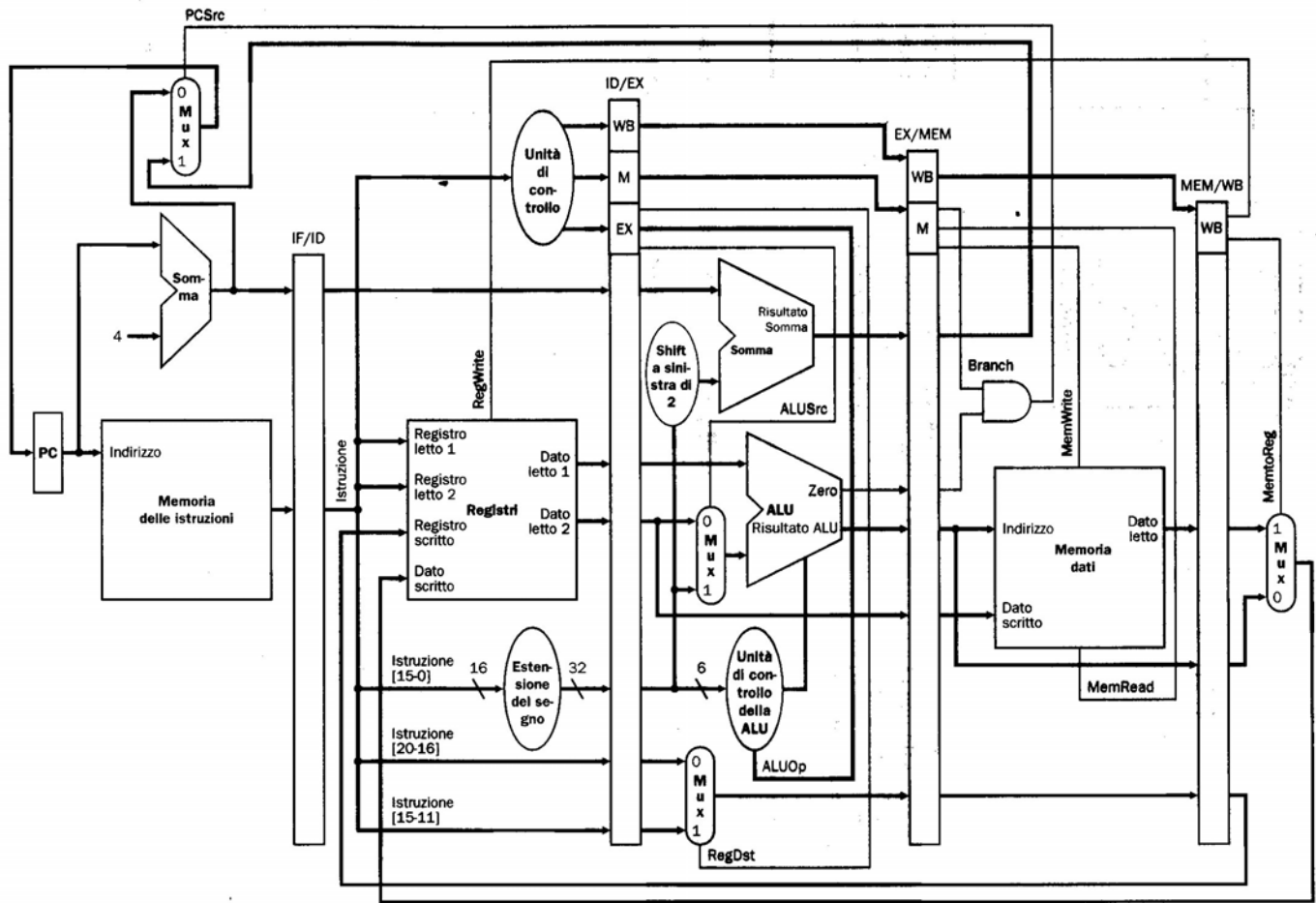
Es. 3. [6] Costruire un sommatore ad anticipatore di riporto per sottrazioni e somme su 4 bit. Calcolare la complessità ed il cammino critico. 1. A cosa servono le forme canoniche? Hanno una diretta implementazione circuitale? Data un circuito digitale, è unica la forma canonica che gli corrisponde? Cos'è un mintermine? Cos'è un maxtermine?

Es. 4. [3] Definire cos'è un'eccezione ed un interrupt? Esiste una posizione preferenziale all'interno del ciclo di esecuzione per servire interrupt ed eccezioni? Cosa vuol dire servire un interrupt per un'architettura (descrivere a grandi linee)?

Es. 5. [1] Perché l'elettronica digitale funziona? A cosa è dovuto il tempo di commutazione? Si può ridurre aumentando la frequenza del clock? Perché?

Es. 6. [4] Descrivere le principali strutture di bus. Come viene gestito l'accesso a bus. Descrivere almeno un meccanismo di controllo degli accessi. Definire una procedura di trasferimento dati su bus sincrono o asincrono. Qual è il vantaggio dell'utilizzo di un bus sincrono nel trasferimento dati da e verso la memoria?

Es. 7. [3] Descrivere, mediante diagramma di flusso, un algoritmo di moltiplicazione "firmware" su 4 bit e disegnarne la corrispondente struttura circuitale. Dimensionare tutti i componenti.



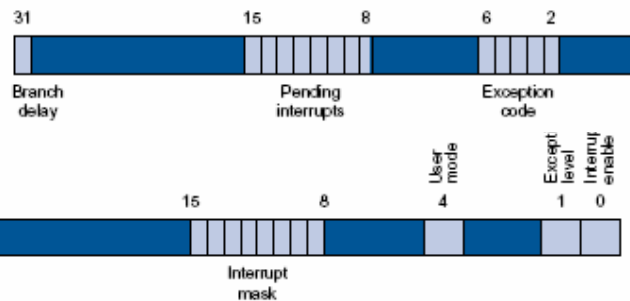
Register File

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

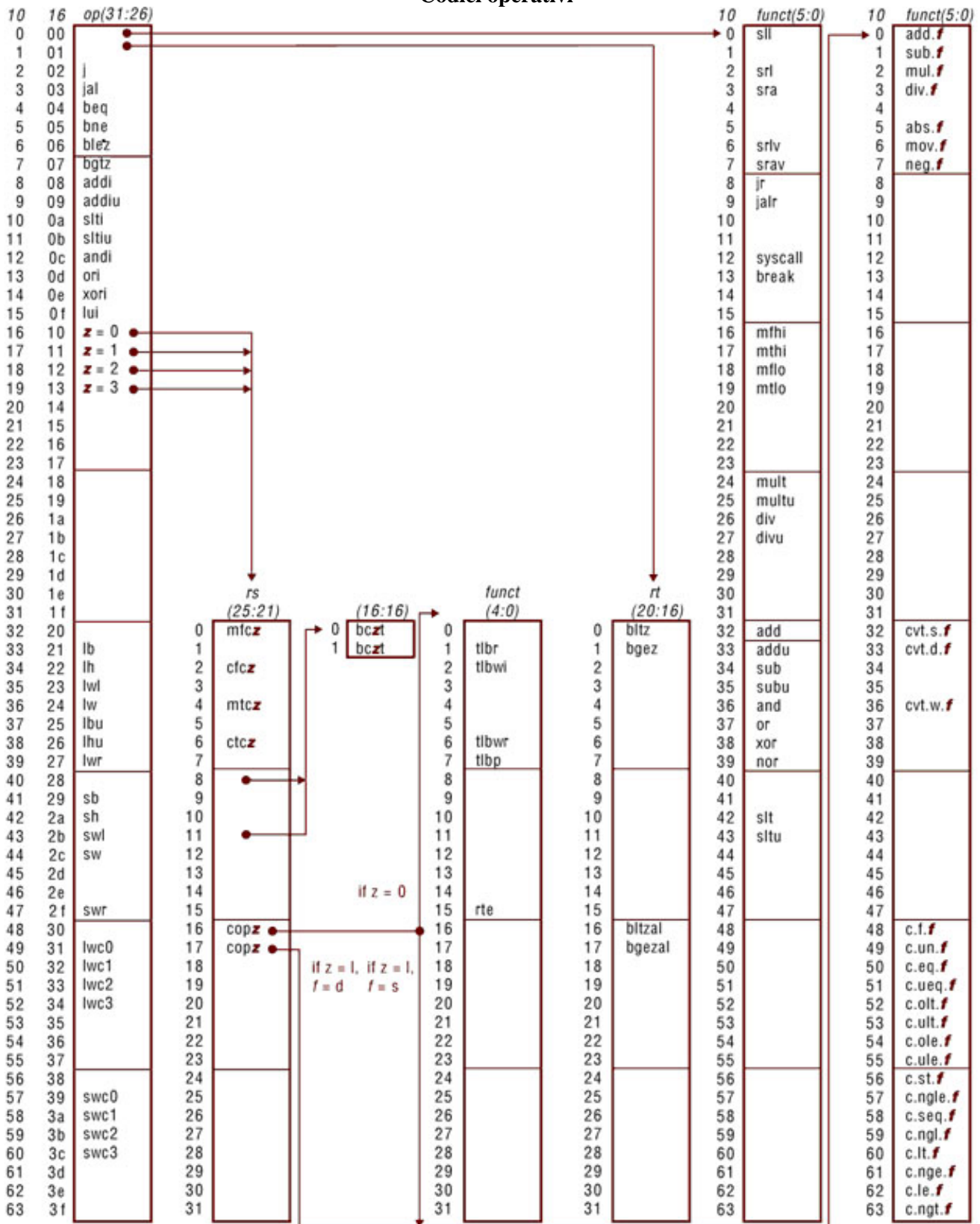


FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18,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