Corso Architettura degli elaboratori e delle reti – Turno 1 – Prof. Borghese

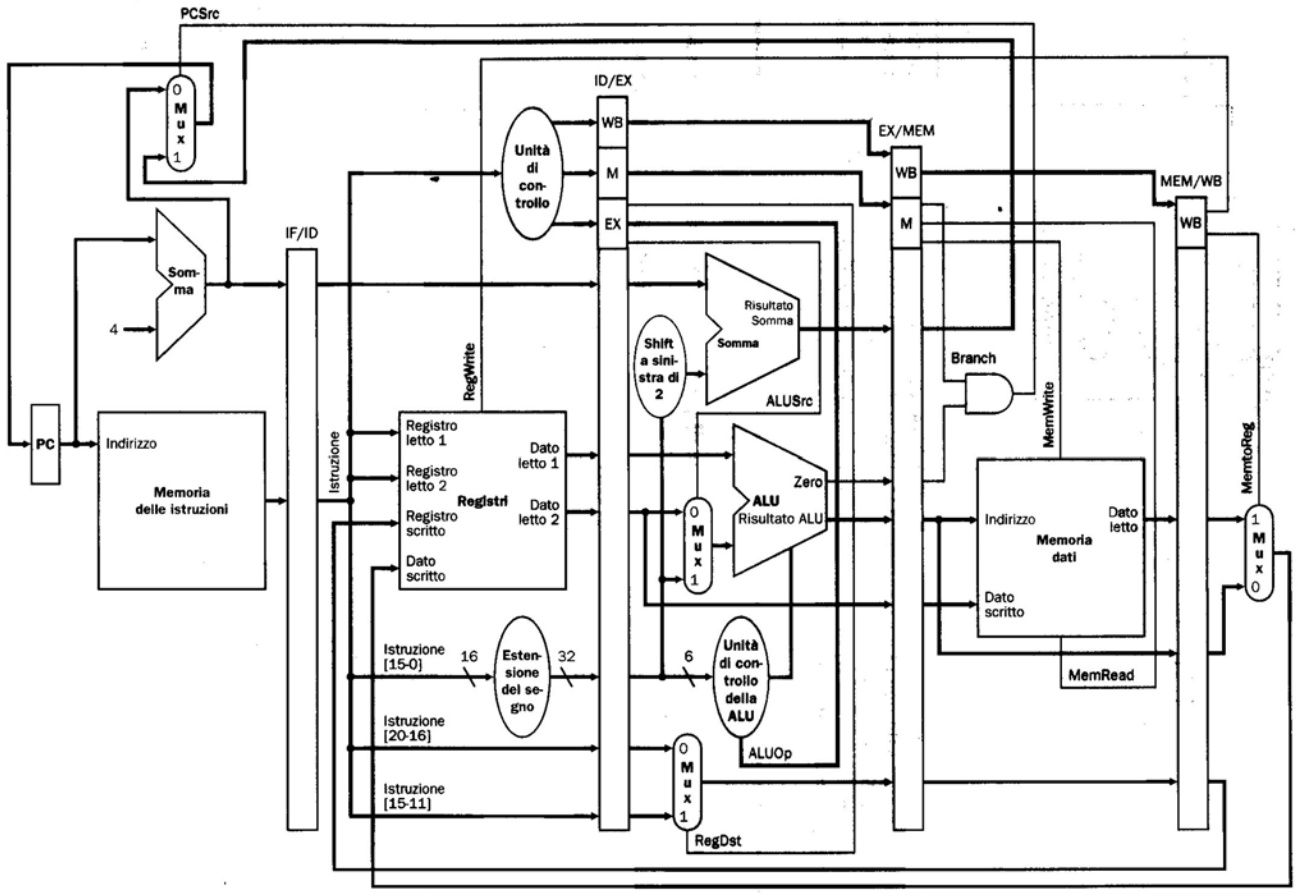
Cognome e nome dello studente: ***Matricola:***

A.A. 2009-2010 – Seconda prova in itinere – Lunedì 10 Maggio 2010

1. [7] Cos'è un branch prediction buffer? Cos'è un correlator predictor? Quale problema risolve? Con quale grado di affidabilità? Progettare la macchina a stati finiti che controlla i salti condizionati, supponendo che un salto debba essere eseguito per due volte consecutive prima di cambiare la predizione da preso a non preso e, analogamente che un salto debba non essere eseguito per due volte consecutive prima di cambiare la predizione da non preso a preso. Definire il cammino critico della macchina.

2. [2] Cosa si intende per hazard? Cos'è uno stallo di una pipeline? E' corretto affermare che una pipeline aumenta la velocità di esecuzione di un'istruzione e perché? In quali condizioni evitereste di utilizzare una pipeline?

3. [7] Data la CPU disegnata sotto:



Scrivere il contenuto di tutti i registri (parte master di PC + parte master dei registri di pipeline) durante l'esecuzione di questo frammento di codice [6]:

```
            …
            …
Loop:   0x4000: and $t1, $s0, $s0
        0x4004: lw $t1, 20($s1)
        0x4008: add $s4, $t5, $t5
        0x400C: or $t3, $t4, $t5
        0x4010: beq $s0, $s2, Loop
        0x4014: sub $s1, $s2, $s3
```

```
0x4018: ori $t6, $t6, 128
…
…
```

quando l'istruzione `beq` ha terminato la fase di fetch (subito prima della commutazione del clock). Evidenziare eventuali hazard che siverificano nell'esecuzione del codice.

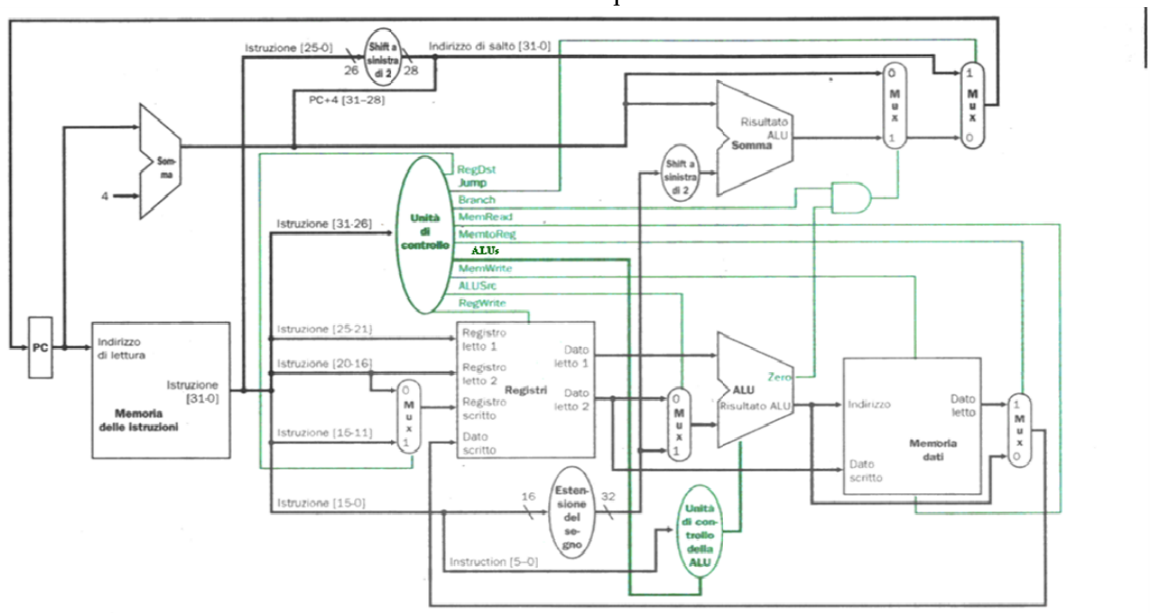
4. [8] Modificare la CPU disegnata sopra in modo da potere gestire gli hazard presenti nella seguente coppia di istruzioni:

lw $t0,24($t1)
add $s0, $t0, $t1
lw $s0,24($t1)

Si supponga gia' di avere a disposizione un circuito di propagazione o feed-forwarding e di doversi occupare solamente degli hazard causati dalle lw. Disegnare i circuiti aggiunti specificando le porte logiche e le loro connessioni.

5. [5] Cosa si intende per gestione vettorializzata e gestione tramite registro di stato delle eccezioni? Come deve essere modificata la CPU riportata sopra in questi due casi? Quali informazioni contengono i registri stato e causa? Come viene gestita dalla CPU questa informazione?

6. [3] Data la CPU a singolo ciclo riportata sotto, specificare i segnali di controllo attivi e quelli indifferenti per l'esecuzione di un'istruzione di addi. Cosa succede quando l'istruzione di addi è in fase di memoria?



7. [4] Riorganizzare il codice seguente perchè possa essere eseguito su una pipeline MIPS senza dovere inserire stalli (nop):

400: add $s0, $s1, $s2
404: add $s3, $s3, $s3
408: add $s4, $s4, $s4
40C: add $t0, $t1, $t2,
410: beq $t0, $t1, 1
414: or $s1, $s2, $s3
418: or $s5, $s5, $s5
41C: andi $s6, $s6, 0
420: j 80000:

Si consideri la pipeline contenenti tutti i circuiti di controllo degli hazard e di possibile soluzione (propagazione).

Number	Name	Cause of exception
0	Int	Interrupt (hardware)
4	AdEL	address error exception (load or instruction fetch)
5	AdES	address error exception (store)
6	IBE	bus error on instruction fetch
7	DBE	bus error on data load or store
8	Sys	syscall exception
9	Bp	breakpoint exception
10	RI	reserved instruction exception
11	CpU	coprocessor unimplemented
12	Ov	arithmetic overflow exception
13	Tr	trap
15	FPE	floating point

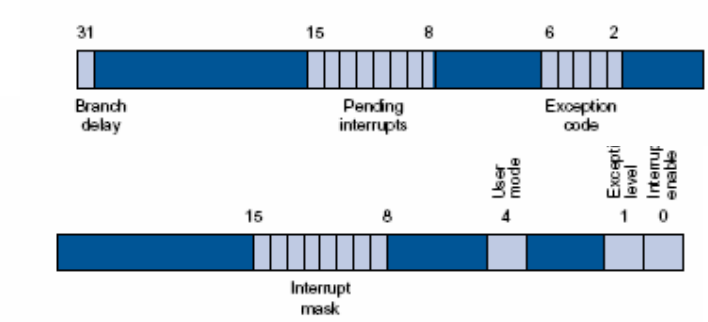
0	zero	constant 0
1	at	reserved for assembler
2	v0	expression evaluation &
3	v1	function results
4	a0	arguments
5	a1	
6	a2	
7	a3	
8	t0	temporary: caller saves
. . .		(callee can clobber)
15	t7	

16	s0	callee saves
. . .		(caller can clobber)
23	s7	
24	t8	temporary (cont'd)
25	t9	
26	k0	reserved for OS kernel
27	k1	
28	gp	Pointer to global area
29	sp	Stack pointer
30	fp	frame pointer (s8)
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

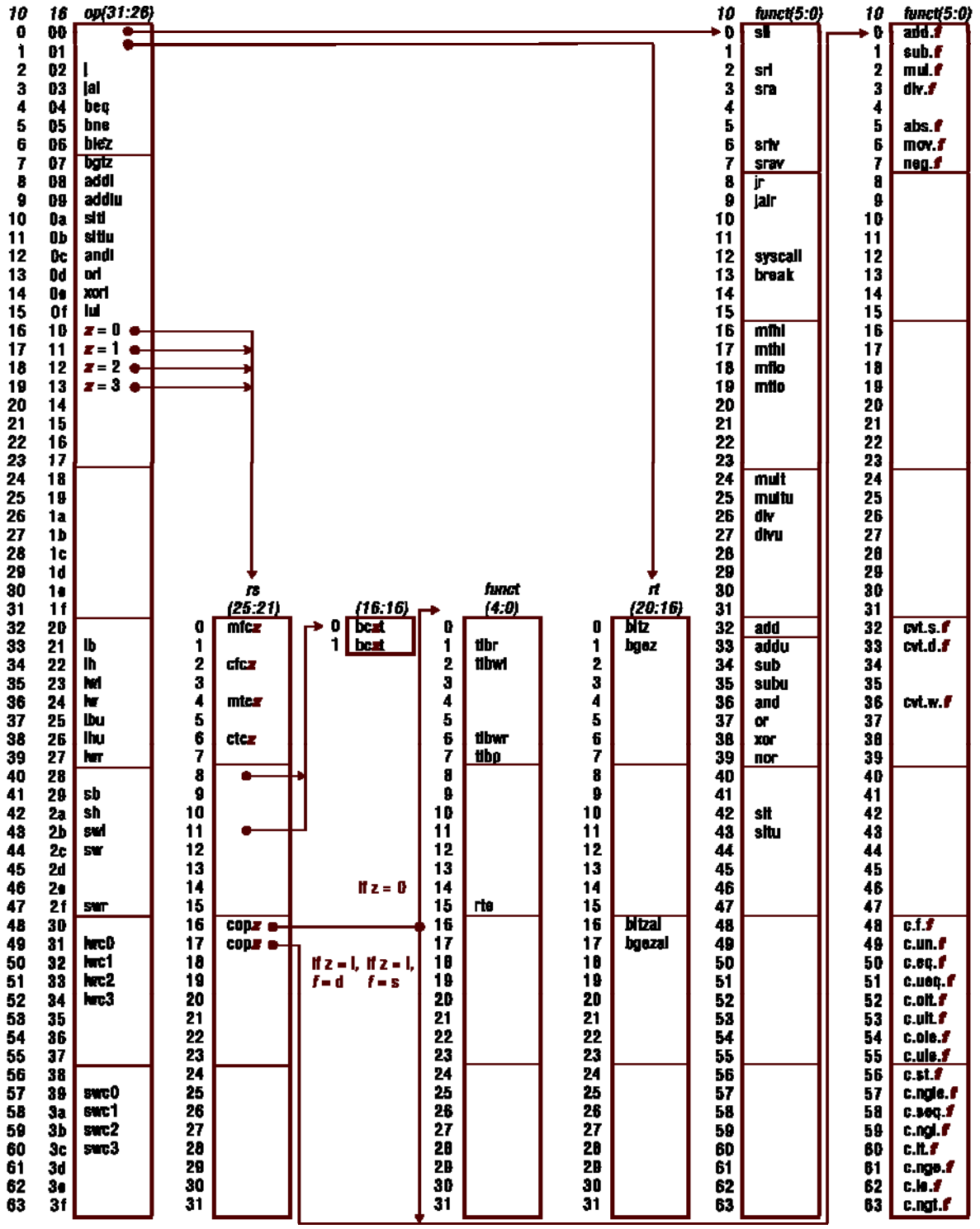


FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18,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