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

A.A. 2008-2009 – Terza prova in itinere – 19 Maggio 2009

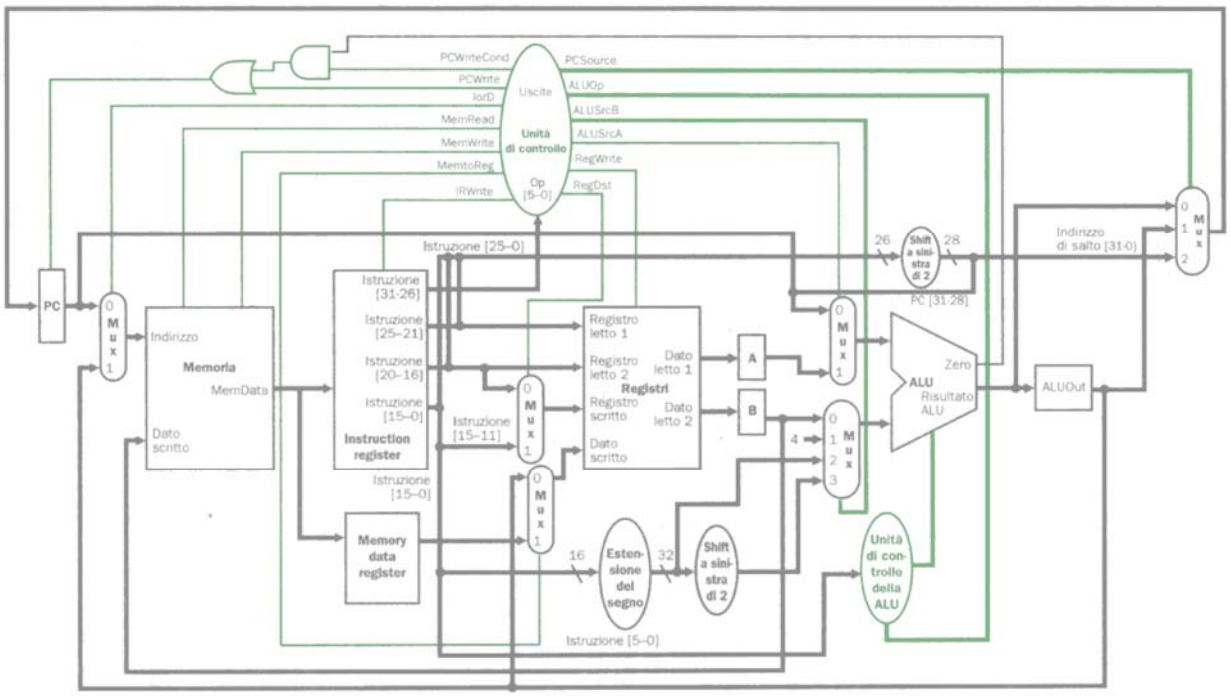
Nome e Cognome dello studente:

[5] Cosa si intende per superpipeline? Cosa si intende per pipeline superscalari? Cosa si intende per reorder buffer? A cosa servono le reservation stations? Cosa si intende per flush di una pipeline? Cosa si intende per esecuzione speculativa? Cos'è un branch prediction buffer? Cos'è un correlator predictor? Descrivere alcune proprietà della pipeline del Pentium IV. Quali problemi si riscontrano nella pipeline del Pentium IV? Cosa si intende per architettura RISC e architettura CISC?

[5]. Specificare lo stato (uscita) dei registri A e B, del registro ALUOut, del Registro Istruzioni, del MDR e del PC al termine del terzo stadio di esecuzione (stadio di EX, prima della commutazione del clock) di ciascuna delle seguenti istruzioni:

Loop: 0x8000 lw $s0, 64($s1)
0x8004 sw $t0, 20($t1)
0x8008 add $t4, $t5, $t5
0x800C beq $t0, $t1, Loop
0x8010 addi $t0, $t1, 16

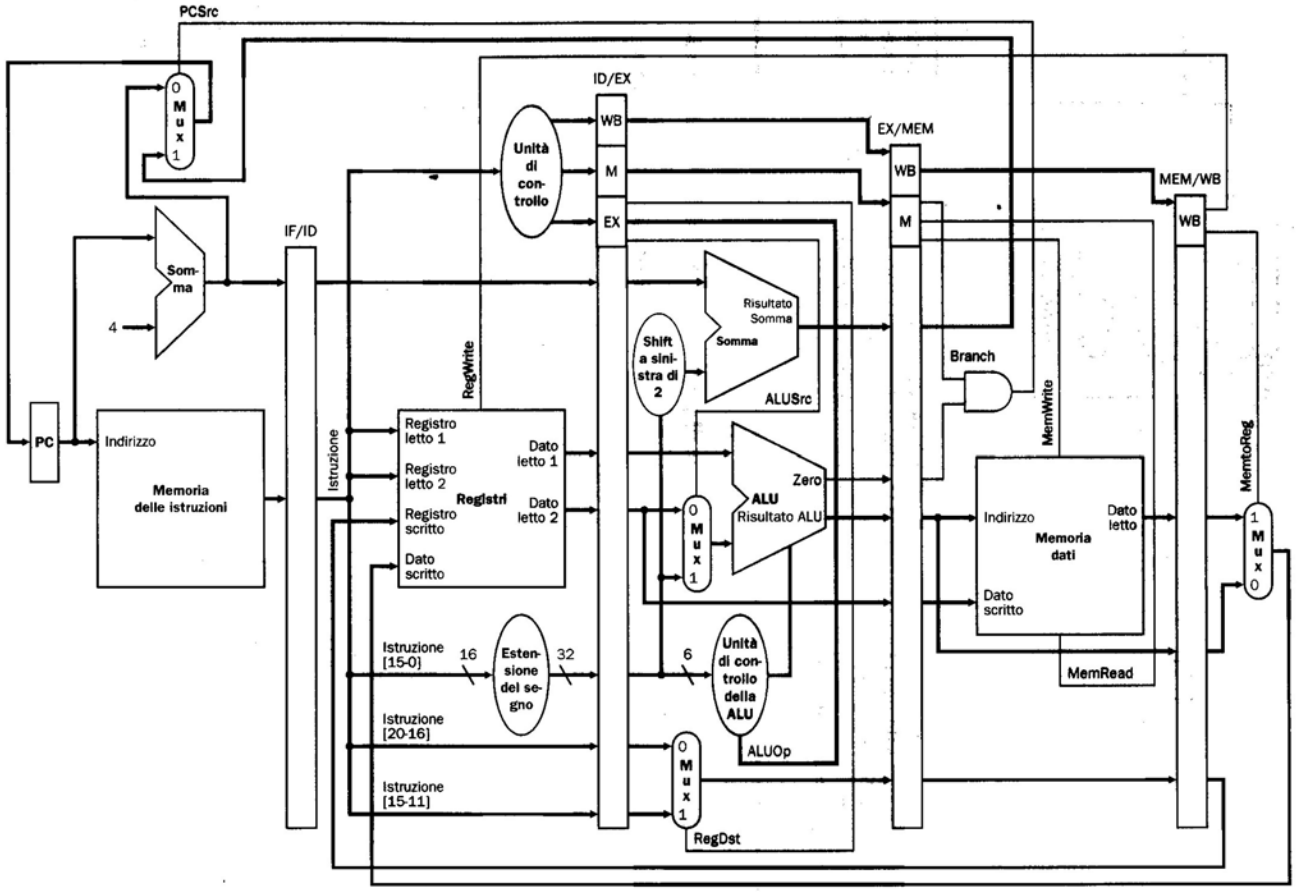
Per quali segnali di controllo è possibile utilizzare il segnale di clock?



[3] Modificare la CPU in modo tale da potere essere in grado di fornire il supporto alla gestione delle eccezioni.

[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?

[6] 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, $t0, $t0
            0x4004: lw $s1, 20($t1)
            0x4008: add $s4, $s5, $s5
            0x400C: or $s3, $s4, $s5
            0x4010: beq $t0, $t2, Loop
            …
            …
```

quando l'istruzione `beq` ha terminato la fase di fetch (subito prima della commutazione del clock).

[7] Modificare la CPU disegnata sopra in modo da potere gestire gli hazard presenti nel frammento di codice di cui sopra. Dimensionare correttamente tutti i registri presenti nella CPU.

[7] Scrivere un programma assembler di risposta ad un interrupt che fa quanto segue:
- se è un interrupt di livello 3, visualizza il messaggio "interrupt " + indirizzo dell'istruzione in cui l'interrupt si è verificato.
- se è un'eccezione, visualizza il messaggio "eccezione" + il numero dell'eccezione.

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

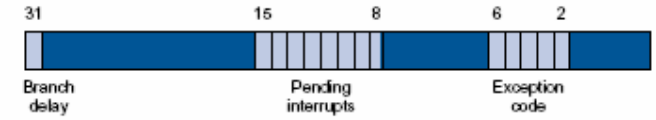
0	zero	constant 0
1	at	reserved for assembler
2	v0	expression evaluation &
3	v1	function results
4	a0	arguments
5	a1	
6	a2	
7	a3	
8	t0	temporary: caller saves
. . .		(callee can clobber)
15	t7	

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)

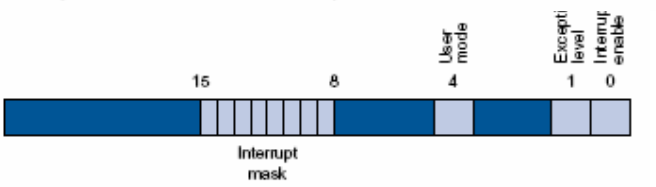
Coprocessore 0

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

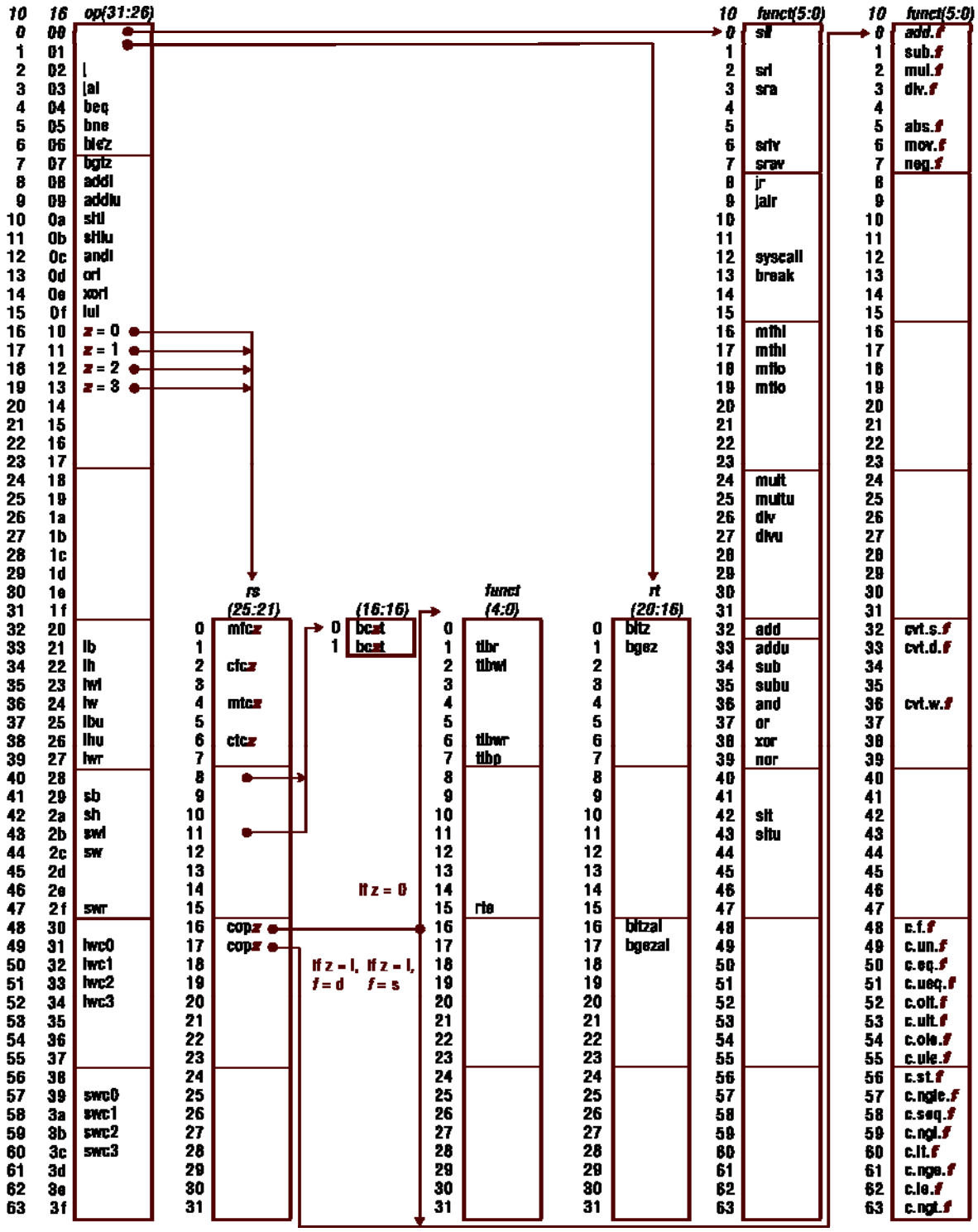


FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18,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