Università degli Studi di Milano - Corso Architettura II – Prof. Borghese – Appello del 24.02.2016

Cognome e nome: ***Matricola:***

1. [7] Data la CPU N. 1, specificare il contenuto di **tutte** le linee (dati e controllo). Indicare quali linee trasportano segnali utili, quando è in esecuzione il seguente segmento di codice [4]:

```
0x00000400 addi $s5, $t2, 24
0x00000404 lw $t0, 8($t0)
0x00000408 and $t0, $s5, $t1
0x0000040C addi $t1, $s1, 4
0x00000410 sw $s2, 32($t0)
0x00000414 sub $s2, $s0, $s2
```

quando la `addi` è in fase di WB. Modificare la CPU in modo che gestisca correttamente la propagazione anche quando un'istruzione di `addi` si trova in fase di calcolo e spiegare la modifica.

2. [2] Cos'è un hazard? Quali tipi di hazard vengono identificati? Cos'è uno stallo? Stallo e bolla sono la stessa cosa? Si verificano hazard nell'esecuzione del codice precedente? Motivare la risposta.

3. [5] Cosa sono gli interrupt e le eccezioni? Come vengono gestite dai sistemi operativi Intel e RISC? Specificare gli elementi della CPU che sono dedicati alla gestione delle eccezioni e supportano il sistema operativo nel MIPS. Modificare la CPU sopra per potere gestire un'eccezione di "Istruzione non valida". Cosa si intende per mascheramento degli interrupt? Viene praticato nei MIPS? Scrivere lo scheletro di un programma assembler di risposta alle eccezioni per il MIPS.

4. [8] Descrivere come funzionano le seguenti tecniche e dire se sono tecniche principalmente software o hardware e perchè. In alcuni casi la risposta corretta può essere entrambi gli approcci. Identificare quali sono i punti forti ed i punti deboli.

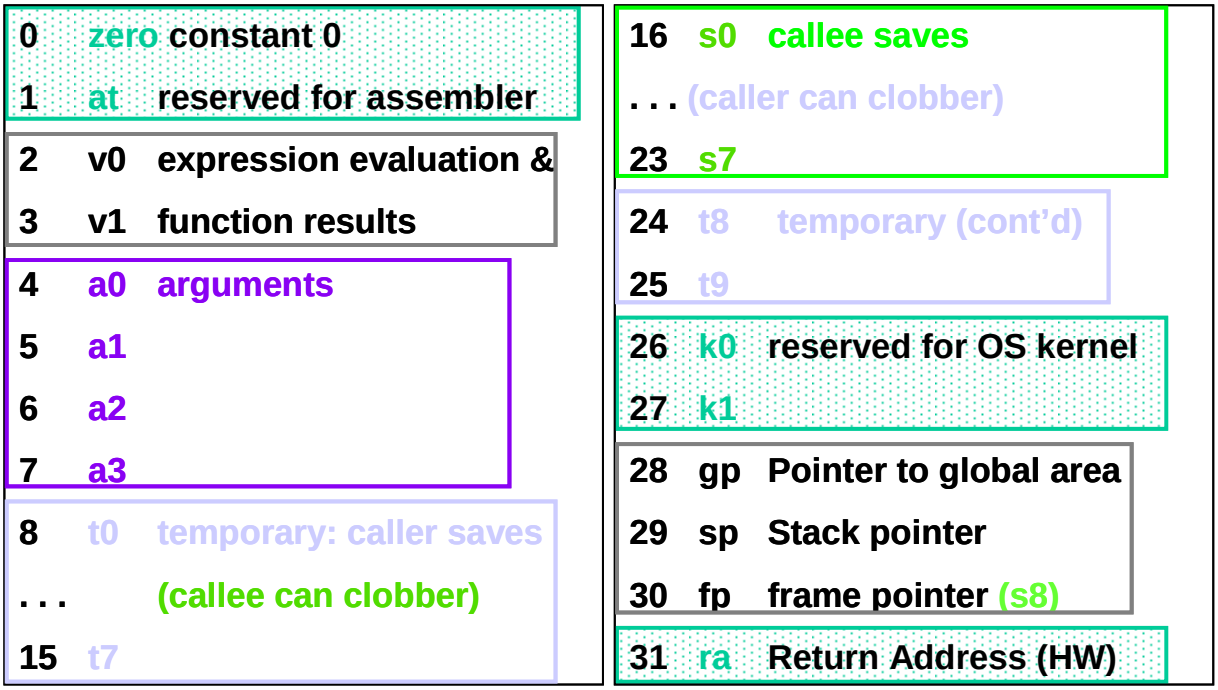
a) Superpipeline
b) Predizione dei salti
c) Branch predicion buffer
d) Speculazione
e) Parallelizzazione dell'esecuzione
f) Parallelizzazione a livello di parola
g) Parallelismo implicito ed esplicito
h) Pipeline superscalari
i) Pipeline dotate di VLIW
j) Esecuzione fuori ordine
k) Reservation station
l) Buffer di riordino
m) Ridenominazione dei registri
n) Branch delay slot
o) Write back
p) Write through
q) Bus snooping
r) Protocolli di coerenza della memoria con un esempio
s) Consistenza della memoria
t) Cluster
u) Multi-core
v) Issue
w) Hazard
x) Bolla
y) Stallo

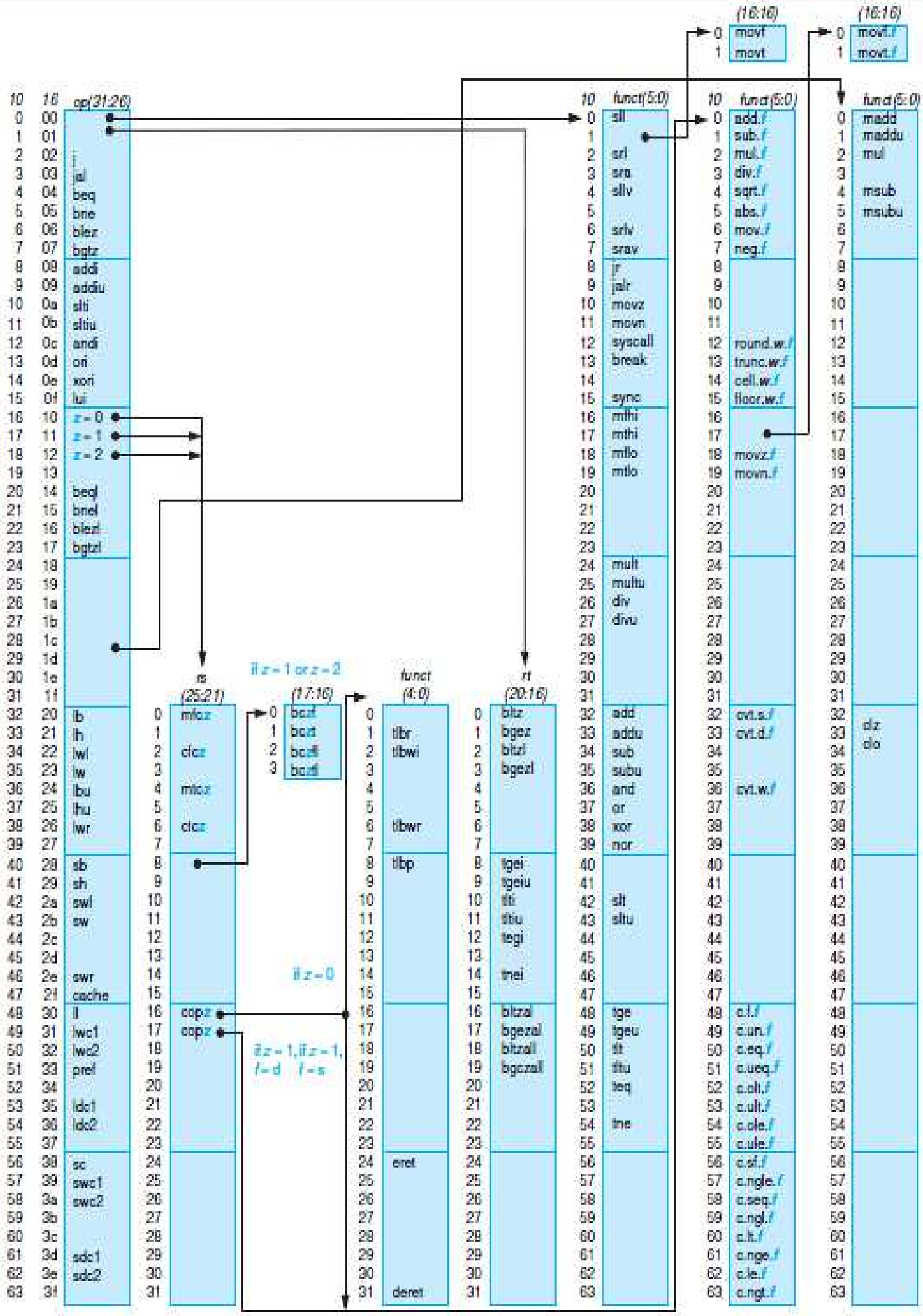
5. [3] Trasformare il codice dell'esercizio 1 in un codice che può essere eseguito in modo efficiente da una pipeline multiple issue statica a due vie, nella quale 1 via è riservata alle operazioni di memoria e 1 via alle operazioni sui dati. Quanto si guadagna in tempo di esecuzione complessivo? Qual'è lo speed-up su questo codice? Può essere generalizzato?

6. [6] Progettare e disegnare una memoria cache a due vie di 2 Kbyte per banco. Si supponga che ciascun banco abbia un'ampiezza di 8 parole. Si supponga un'architettura MIPS a 64 bit. Dimensionare correttamente tutti i campi. Identificare dove viene cercato il dato associato alla seguente istruzione di lettura: lw $t0, 2120($zero) (lettura della parola che inizia all'indirizzo 2120 della memoria principale).

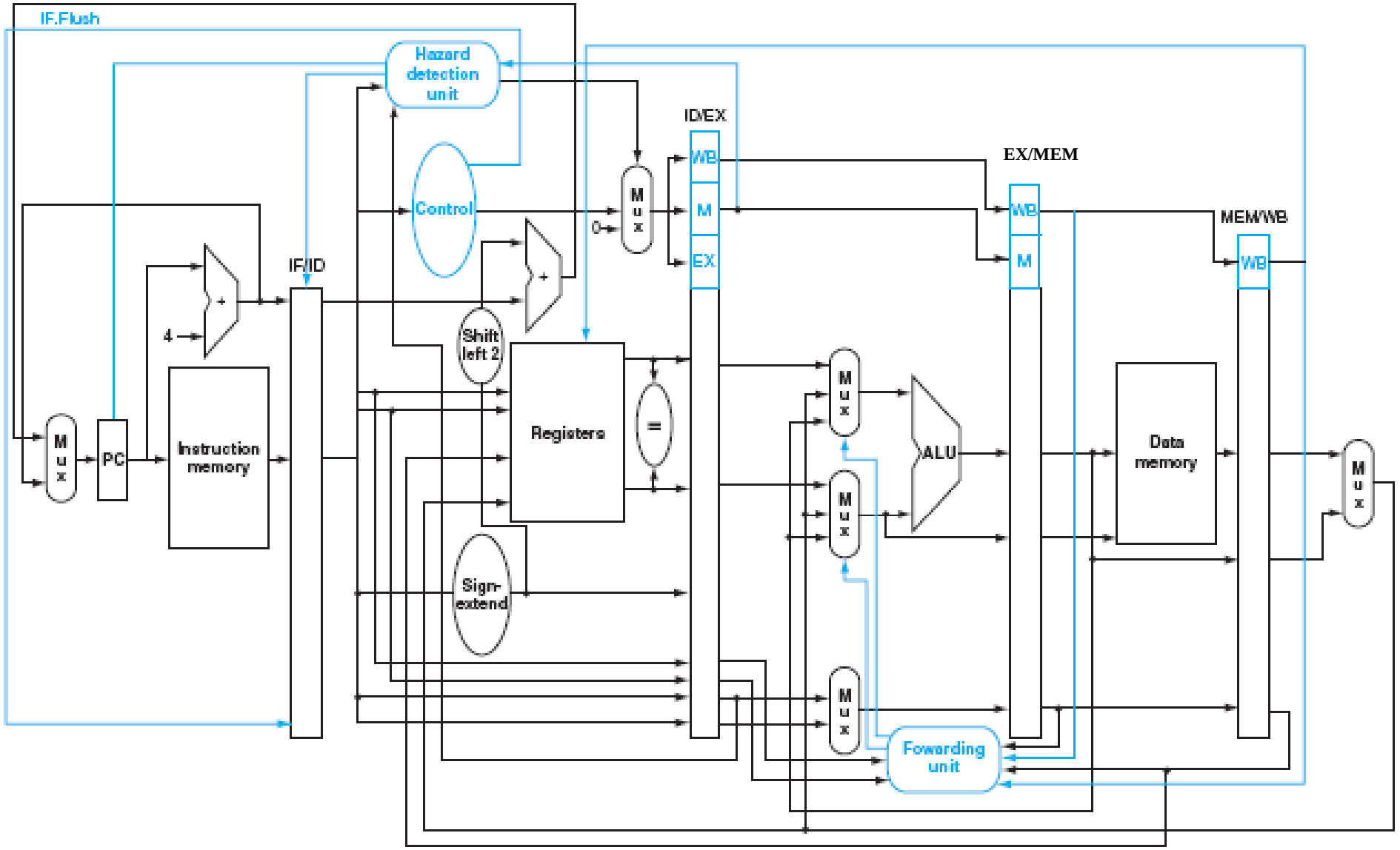
7. [4] Cosa sono i benchmark? Cos'è lo SPEC? Quale ruolo ha? Cosè il "roof model"? Cosa rappresenta l'intensità aritmetica? Si ricerisce ad una CPU o ad un particolare programma? Data una CPU Core i7 Intel con 8 core, a 64 bit (dati float su 64 bit) con 16 cammini di calcolo per ciascun core, in ciascuno dei quali vengono elaborati dati a 128 bit. Detta CPU ha un clock di 4GHz. A questa CPU è associato un sistema di memoria che è in grado di sostenere l'elaborazione con un flusso dati dalla memoria alla CPU pari a 16Gbyte/s. Determinare la massima velocità di elaborazione della CPU per 6 diversi programmi benchmark che hanno intensità aritmetica rispettivamente di: 1/8, 1/4, 1/2, 1, 4, 8, 32 e determinare se le prestazioni sono limitate dalla memoria o dal calcolo.

Registri del register file




op(31:26)
rs (25:21)
rt (20:16)
funct (4:0)
funct(5:0)
(16:16)
if z = 1 or z = 2
(17:16)
if z = 0
if z = 1, if z = 1, f = d f = s

CPU N. 1


IF.Flush
Hazard detection unit
ID/EX
EX/MEM
MEM/WB
Control
WB
M
EX
IF/ID
Shift left 2
Registers
Instruction memory
PC
ALU
Data memory
Sign-extend
Fowarding unit
Mux
4
0