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

1. [6] Data la CPU di Figura 1, specificare il contenuto di **tutti** i bus, quando è in esecuzione il seguente segmento di codice [5]:

```
0x00000400 or $s5, $t2, $t1
0x00000404 sw $s1, 8($s0)
0x00000408 add $t4, $s5, $s1
0x0000040C addi $t1, $s1, 100
0x00000410 lw $s2, 32($t4)
```

quando l'istruzione di `or` si trova in fase di WB. Specificare sullo schema (con colore o con tratto grosso) quali linee, all'interno dei diversi stadi, trasportino dati utili all'esecuzione dell'istruzione [2]. Ci sono hazard nel codice precedente? Motivare la risposta [1].

2. [5] Cosa sono gli interrupt e le eccezioni? Come vengono gestiti dalle architetture Intel e dalle architeture MIPS/ARM? Specificare gli elementi della CPU MIPS che sono dedicati alla gestione delle eccezioni e supptano la gestione delle eccezioni e cosa contengono. Modificare la CPU sopra per potere gestire un'eccezione di "Overflow" e un'eccezione di "Memory Miss". Cosa si intende per mascheramento degli interrupt? Viene praticato nei MIPS? Come vengono gestite le eccezioni e gli interrupt dai sistemi operativi sul MIPS? Scrivere uno scheletro di possibile codice.

3. [5] Modificare la pipeline in Figura 2 perché diventi una pipeline superscalare. Spiegare la ragione e lo scopo di tutte le modifiche. Che differenza c'è tra pipeline super-scalare e pipeline dotata di VLIW? Quali sono i vantaggi di un approccio rispetto all'altro.

4. [1] Come si implementa l'esecuzione vettoriale in una pipeline multi-scalare.? Mostrarlo modificando un cammino di esecuzione

5. [1] Cos'è il branch delay slot? Quando è conveniente implementarlo? Quali sono gli inconvenienti?

6. [1] Spiegare chiaramente cosa si intende per stallo e illustrare almeno una tecnica per ottenerlo.

7. [4] Disegnare una memoria cache (parte dati + TAG + bit di validità – non è necessario disegnare le porte di lettura e scrittura) per un'architettura MIPS a 64 bit, a 8 vie di 128 KByte per banco, e linee di 16 parole (per ciascun banco). Definire cosa rappresenta il campo TAG e dimensionarlo opportunamente. Dove posso trovare il dato letto dall'istruzione lw $t1, 1024($0)? Da quanti bit è costituita questa memoria complessivamente? Cosa succede quando si verifica una miss? Come si può limitare la frequenza di miss? Da quale degli otto banchi viene scaricato il dato quando occorre caricare una nuova linea? Perché?

8. [2] Cosa sono i codici di rilevamento e correzione degli errori? Come funziona il codice di Hamming? Fare un esempio per un dato su 8 bit.

9. [2] Cosa rappresenta il "roof model"? Cosa rappresenta l'intensità aritmetica? Si riferisce ad una CPU o ad un particolare programma? Un programma che elabora matrici sparse sarà un programma con intensità aritmetica alta o bassa? Perché? Quali sono i passi per ottimizzare le prestazioni del codice suggeriti dal roof-model? Cos'è un kernel benchmark? Cos'è lo SPEC?

10. [5] Le memorie sono costituite da una gerarchia costruita secondo criteri ben precisi. Rispondere a queste domande motivando la risposta:
    a) Cosa si intende per gerarchia delle memorie?
    b) Cosa si intende per coerenza e consistenza di una memoria?
    c) A quale tipo di memoria si applicano?
    d) Quali sono i meccanismi messi in atto per garantire la coerenza e la consistenza della memoria nelle architetture mono-processore e nelle architetture multi-processore?

e) Quali sono i vantaggi e svantaggi di ciascun meccanismo?
f) Cosa si intende per hit e miss e come vengono gestiti? Chi li gestisce?
g) Perché le miss sono critiche?
h) Che differenza c'è tra una miss e un page fault? Cos'è un page fault?
i) Cos'è la memoria virtuale?
j) Cos'è la Tabella delle pagine? Dove si trova?
k) Cos'è il "Translation Lookaside buffer"? Dove si trova?
l) A cosa servono la memoria virtuale, il TLB e la tabella delle pagine?
m) Che relazione c'è tra la memoria virtuale e la memoria fisica?
n) Chi utilizza la memoria virtuale? Chi utilizza la memoria fisica?
o) Cosa succede quando la CPU chiede una parola alla memoria?

11. [3] Come vengono gestiti gli I/O da un'architettura? Come funziona il "daisy chain"? Cosa si intende per arbitraggio?

Registri del register file

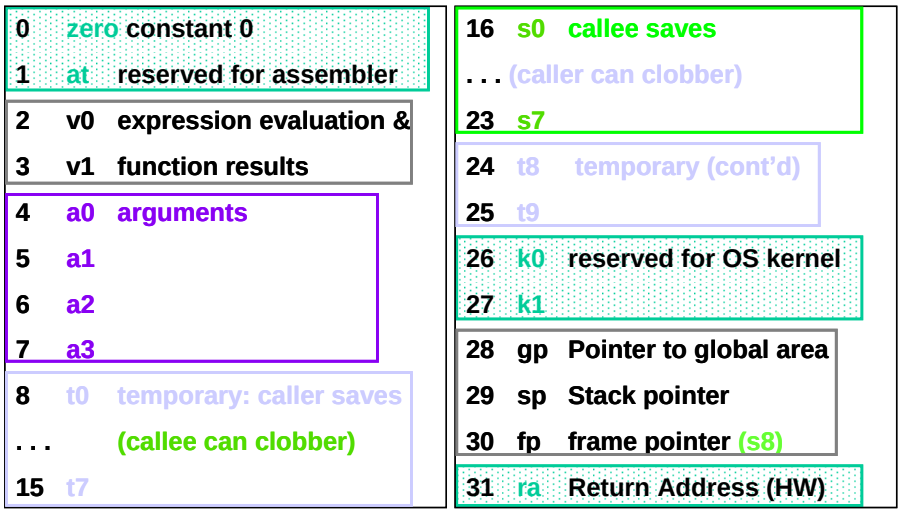


Figure 1

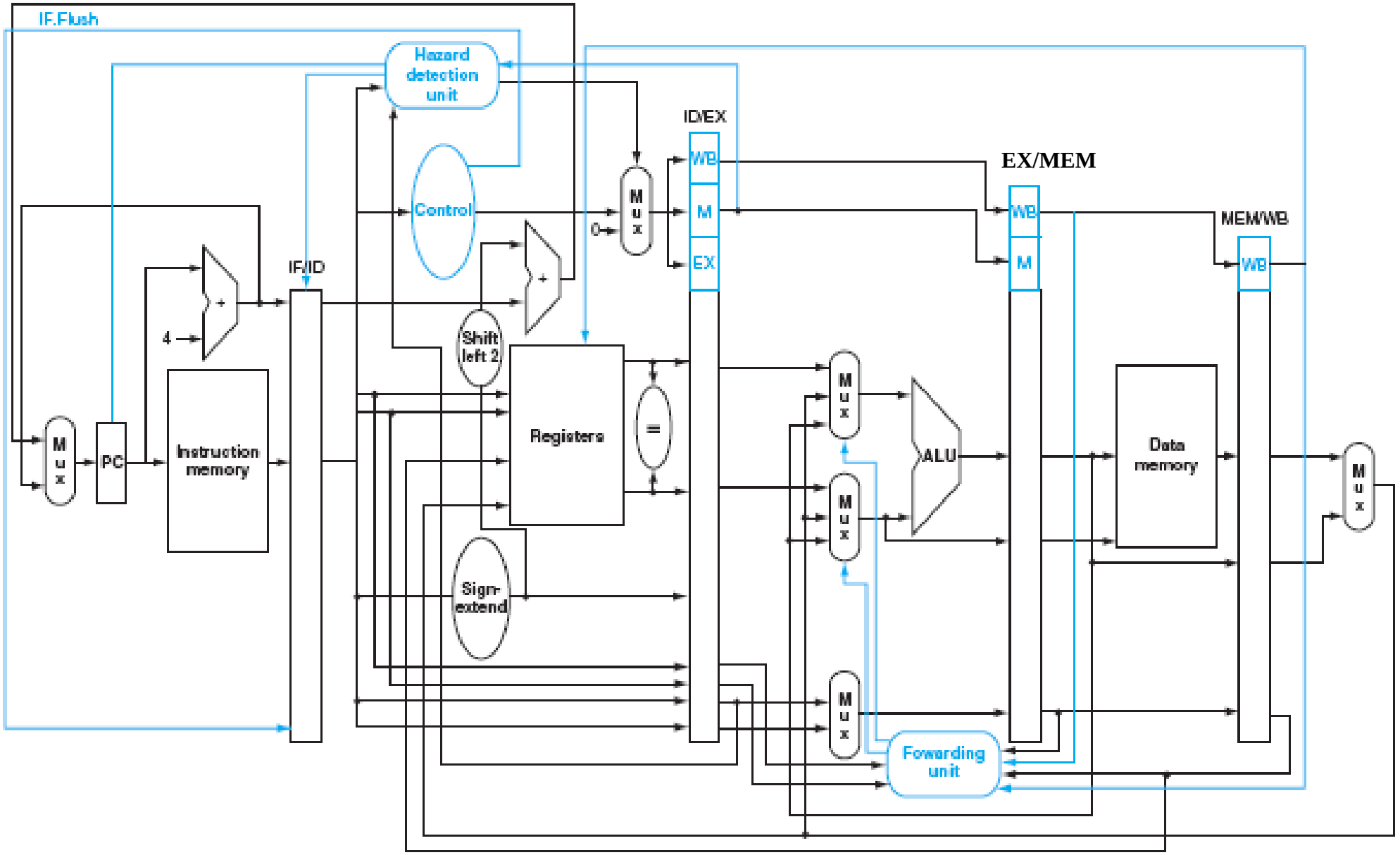
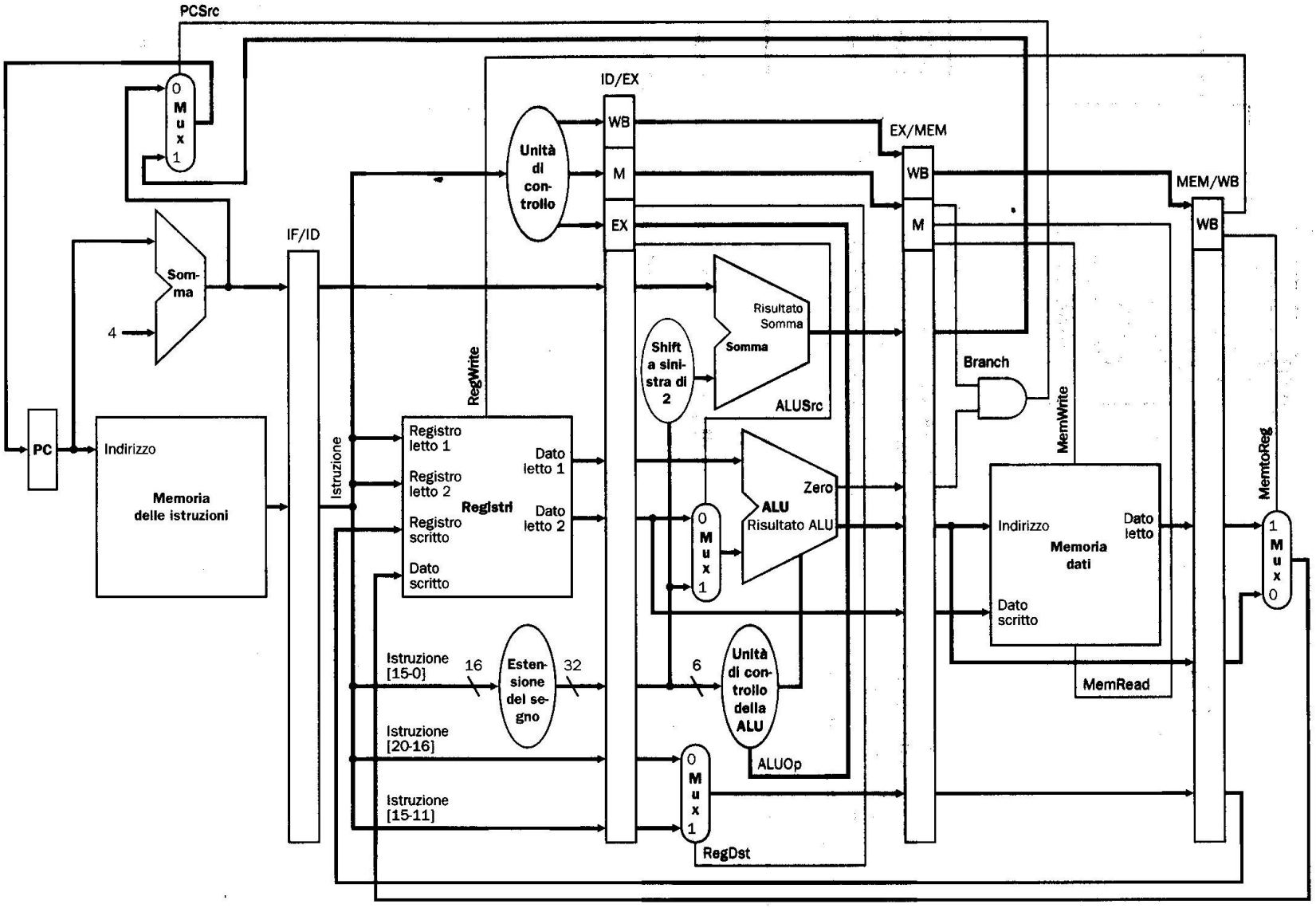

IF.Flush
Hazard detection unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
M u x
0
IF/ID
+
4
Shift left 2
Registers
=
PC
Instruction memory
Sign-extend
ALU
Data memory
Fowarding unit

Figure 2


PCSrc
0
M
u
x
1
ID/EX
WB
M
EX
Unità di controllo
EX/MEM
WB
M
MEM/WB
WB
Som-
ma
4
IF/ID
Risultato Somma
Somma
Shift a sinistra di 2
ALUSrc
RegWrite
Branch
MemWrite
MemtoReg
PC
Indirizzo
Memoria delle istruzioni
Istruzione
Registro letto 1
Registro letto 2
Registro scritto
Dato scritto
Registri
Dato letto 1
Dato letto 2
0
M
u
x
1
Zero
ALU
Risultato ALU
Indirizzo
Memoria dati
Dato letto
Dato scritto
MemRead
1
M
u
x
0
Istruzione [15-0]
16
Estensione del segno
32
6
Unità di controllo della ALU
ALUOp
Istruzione [20-16]
Istruzione [15-11]
0
M
u
x
1
RegDst