Università degli Studi di Milano - Corso Architettura degli elaboratori e delle reti II – Prof. Borghese – Appello del 22.07.2025

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

1. [7] Modificare la CPU di Figura 1 perchè possa essere esguito correttamente questo segmento di codice [3]:

```
0x800 xor $t3, $t2, $t1
0x804 lw $t1, 64($0)
0x808 sw $t1, 64($0)
0x80C add $t4, $t1, $t1
0x810 and $s1, $t2, $t3
```

Su questa CPU specificare il contenuto di **tutti** i bus, quando è in esecuzione il segmento di codice di cui sopra [3] con l'ìstruzione di `xor` 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 in quello stadio [1].

2. [3] Cosa si intende per gerarchia delle memorie? Spiegare chiaramente cosa si intenda per **coerenza** e **consistenza** di una memoria. Fare degli esempi. A quali memorie si applicano? Spiegare come funzionano i seguenti protocolli che mirano a garantire la coerenza:
a) Write-back
b) Write-through
c) Write invalidate
Cos'è il lock? A cosa serve?

3. [4] Come viene gestito l'input/output dall'ISA delle architetture MIPS e dalle architetture Intel? Identificare i componenti principali di un'interfaccia di una periferica. Descrivere a grandi linee come avviene una transazione su un bus sincrono e asincrono ed evidenziare le differenze. Cosa è un bus? Quale è la sua funzione? Descrivere il protocollo e la struttura di un collegamento daisy chain e fare un esempio di funzionamento Cosa si intende per arbitraggio centralizzato e decentralizzato? Cosa sono i bridge? Come sono organizzati i dischi magnetici? Come viene calcolata la latenza in lettura di un disco? Come si può mascherare? Come funziona una unità a nastro magnetico e quale è la sua funzione? Da cosa dipende il tempo di lettura?

4. [2] Cosa si intende per weak scaling e strong scaling? A cosa si applica? Cosa sono i benchmark? Perchè sono stati introdotti? Da chi vengono usati e perchè?

5. [1] Cos'è il blocking? A cosa serve? Come funziona?

6. [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 cosa contengono. Modificare la CPU di Figura 1 per potere gestire un'eccezione di "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.

7. [1] Spiegare chiaramente cosa si intende per stallo e illustrare almeno una situazione in cui si verifica e perché.

8. [5] Disegnare una memoria cache (parte dati + TAG + bit di validità) e la sua porta di lettura per un'architettura MIPS a 64 bit, a 4 vie di 32 KByte per banco, e linee di 8 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? Definire quali sono i tipi di miss della cache e quali le possibili soluzioni per ridurne l'impatto. Come si può limitare la frequenza di miss? Spiegare come funziona la tecnica di sostituzione LRU esatta e LRU approssimata per questa memoria.

9. [4] Cos'è la memoria virtuale? Cos'è la Tabella delle pagine? Dove si trova? Cos'è il "Translation Lookaside buffer"? Dove si trova? A cosa servono la memoria virtuale, il TLB e la tabella delle pagine? Che relazione c'è tra la memoria virtuale e la memoria fisica? Chi utilizza la memoria virtuale? Chi utilizza la memoria fisica? Cosa succede quando la CPU chiede una parola alla memoria? Cosa à un page fault? Quando

si verifica?

10. [4] Cosa è un'architettura multiple-issue? Cosa si intende per architettura multiple-issue statica e dinamica? Descrivere le caratteristiche, le problematiche, pro e contro. Descrivere il funzionamento della micro-architettura del Cortex A53 ARM, riportata in Figura 2.

Registri del register file

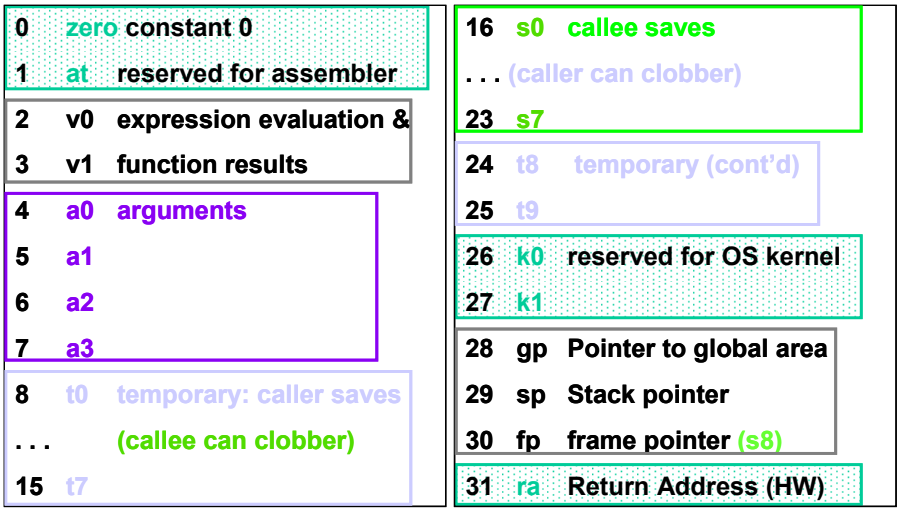


Figure 1

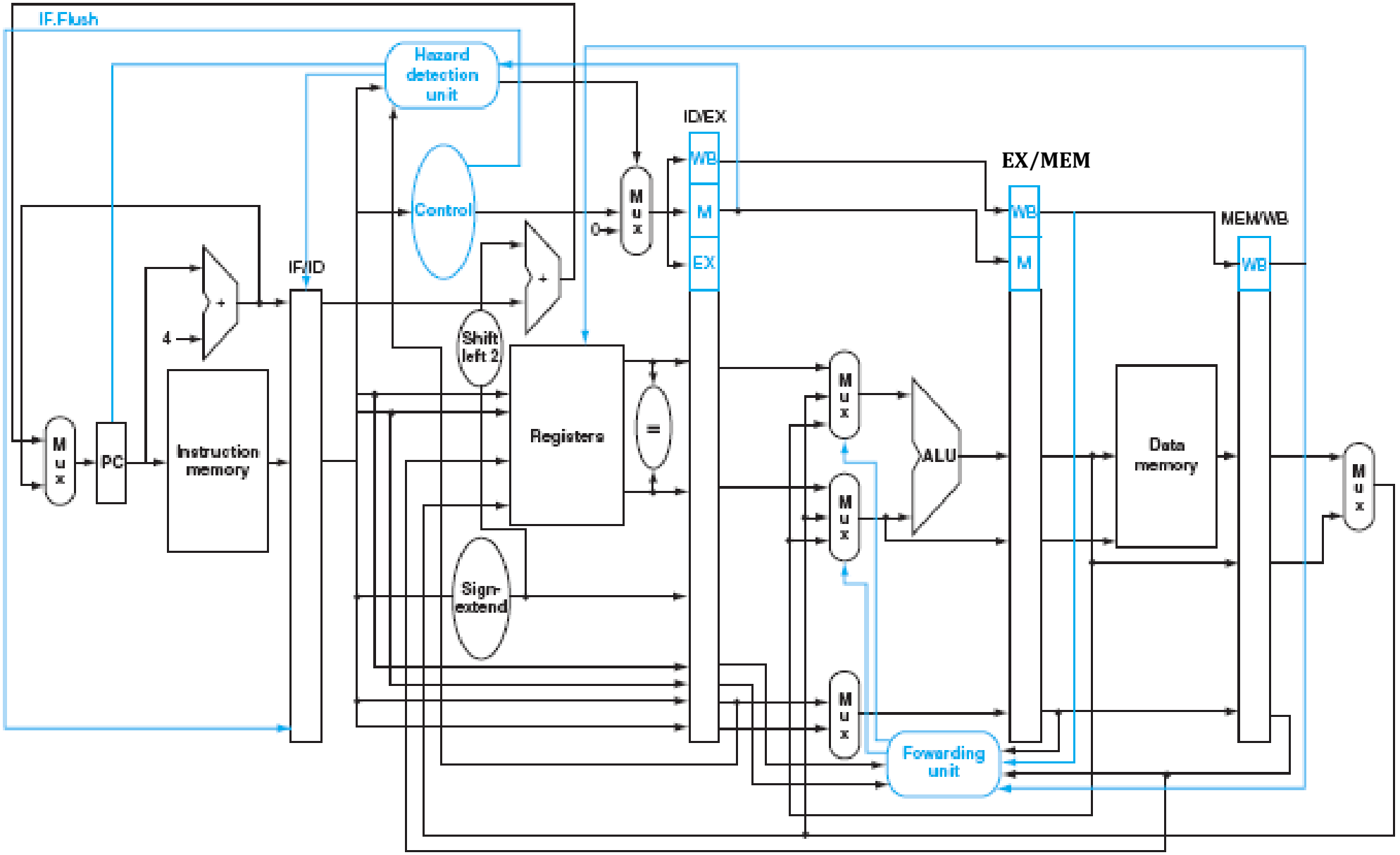

IF.Flush
Hazard detection unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
M u x
0
IF/ID
4
+
Shift left 2
Registers
=
PC
Instruction memory
Sign-extend
ALU
Data memory
Fowarding unit

Figure 1

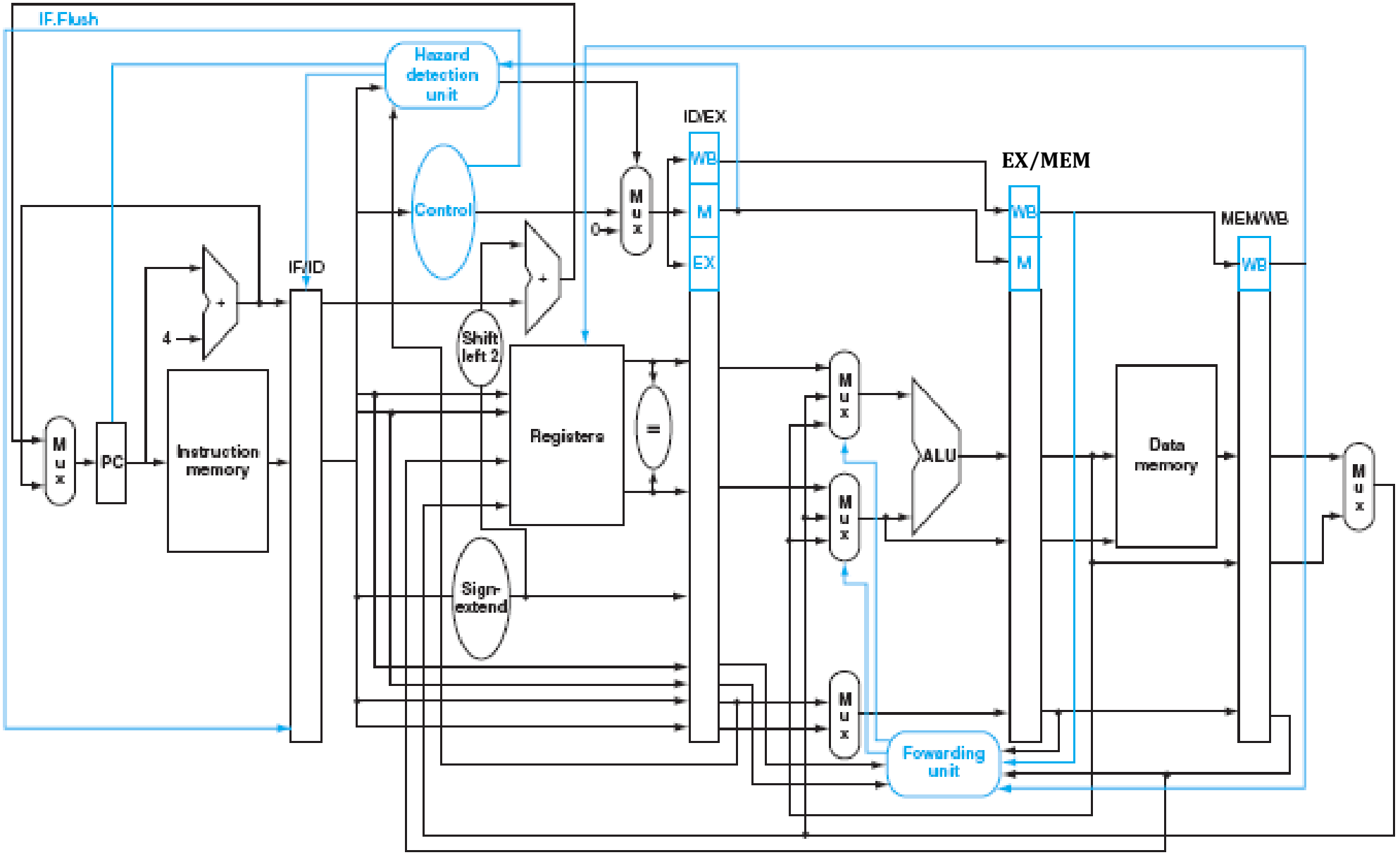
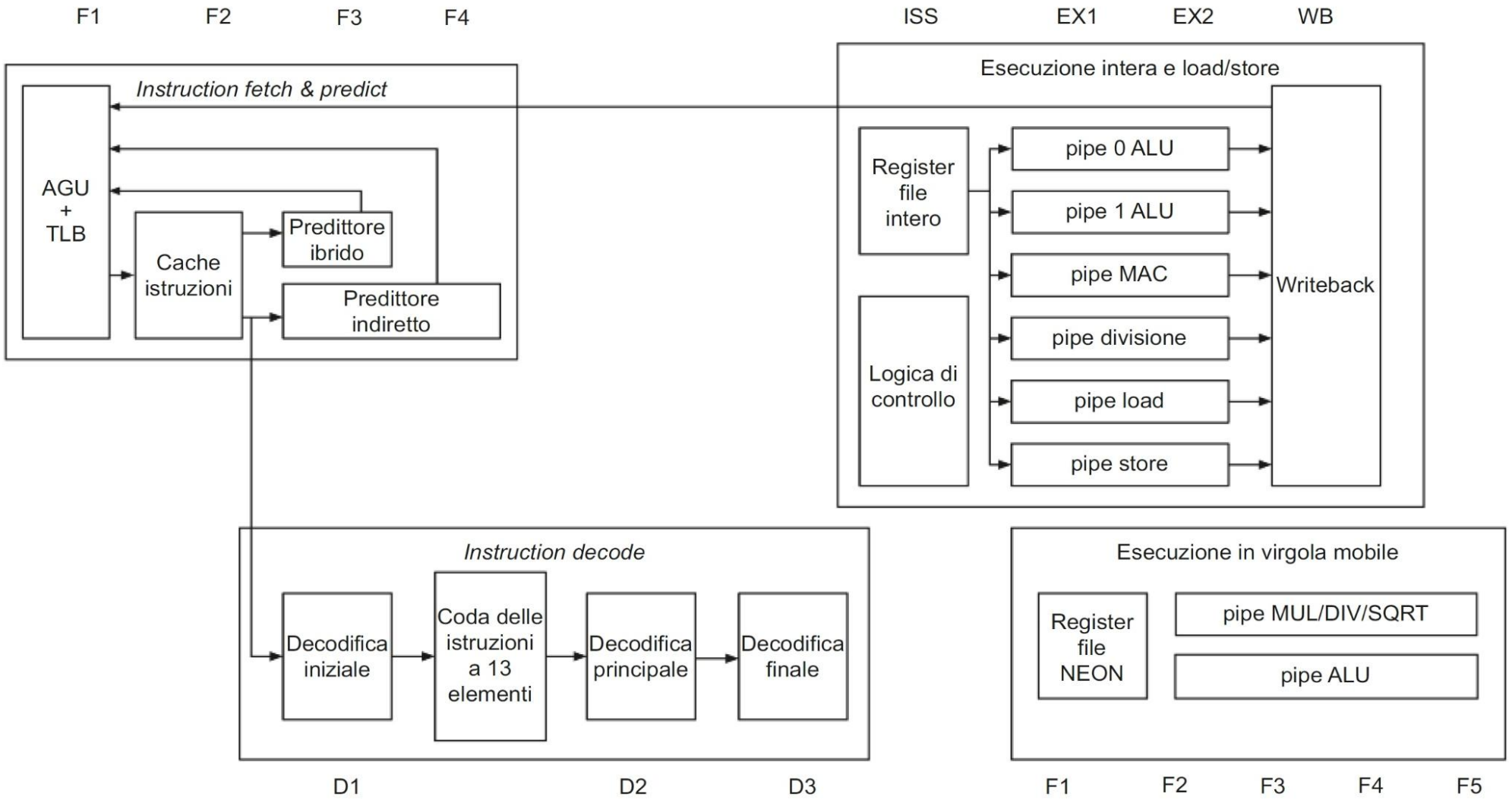

IF.Flush
Hazard detection unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
M u x
0
IF/ID
4
+
Shift left 2
Registers
=
PC
Instruction memory
Sign-extend
ALU
Data memory
Fowarding unit

Figura 2


F1
F2
F3
F4
ISS
EX1
EX2
WB
Instruction fetch & predict
AGU + TLB
Cache istruzioni
Predittore ibrido
Predittore indiretto
Esecuzione intera e load/store
Register file intero
Logica di controllo
pipe 0 ALU
pipe 1 ALU
pipe MAC
pipe divisione
pipe load
pipe store
Writeback
Instruction decode
Decodifica iniziale
Coda delle istruzioni a 13 elementi
Decodifica principale
Decodifica finale
Esecuzione in virgola mobile
Register file NEON
pipe MUL/DIV/SQRT
pipe ALU
D1
D2
D3
F1
F2
F3
F4
F5