## Architetture - canale PZ

**Scritto di teoria del 13 luglio 2009**

Nome: _____

Matricola: _____

**Esercizio 1.** Considerate l'architettura MIPS con pipeline in Figura 1 (è quella vista a lezione) e rispondete a queste due domande:

1. quanti cicli di clock sono necessari per completare l'esecuzione del seguente frammento di codice?
2. E' possibile riordinare il codice in modo da renderlo più veloce?

Tenete conto delle seguenti cose:

- l'esecuzione termina quando l'ultima istruzione *esce* dalla pipeline, per semplicità, considerate che *tutte* le istruzioni eseguano tutti e cinque gli stadi della pipeline;
- se nel blocco registri scrivete e leggete contemporaneamente (durante lo stesso ciclo di clock) lo stesso registro, il valore letto è quello che state scrivendo (quello nuovo).

Fate vedere la correttezza della vostra risposta disegnando lo schema della pipeline.

```
lw $s1, 4($s0)
sw $s1, 4($s2)
add $s2, $s2, $s1
lw $s0, 0($s5)
add $s5, $s0, $s2
lw $s7, 0($s5)
```

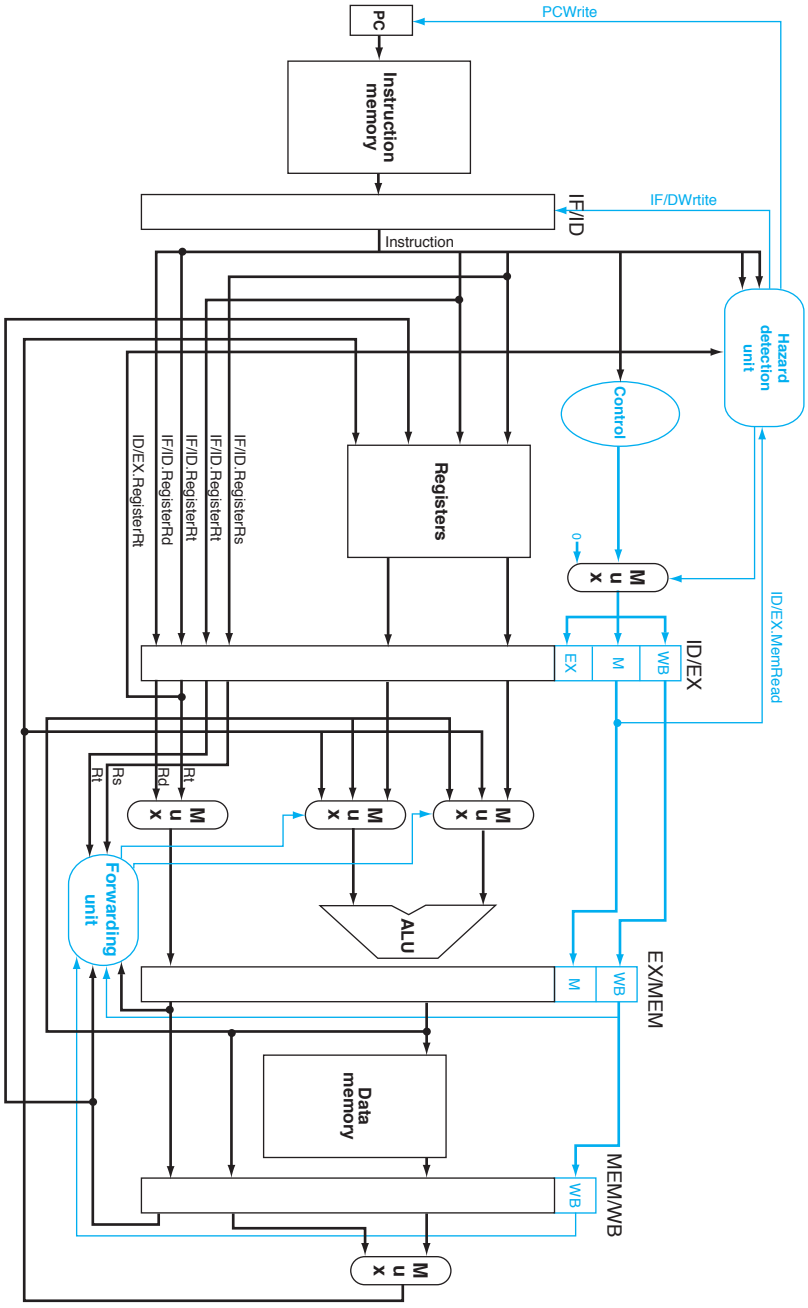


Figure 1: Architettura con pipeline.

**Esercizio 2.** Ecco una serie di accessi a word di 32 bit in memoria: 28, 16, 20, 48, 144, 88, 132, 20, 76, 248, 92, 96. Gli indirizzi sono a byte! Assumete di avere una cache *one-way associative* con 4 blocchi di 8 word, inizialmente vuota. Indicate quali accessi sono miss e quali hit.

**Esercizio 3.** Considerate il problema di fare la somma di due vettori $a$ e $b$ di lunghezza $N$. Quali di questi due programmi è più veloce e perchè? (Tenete conto della nostra architettura pipeline con cache) Quale struttura di cache è migliore per questo programma e perché?

```
Programma A:

s=0;
for (i=0; i<N; i++)
   s+=a[i];
for (i=0; i<N; i++)
   s+=b[i];

Programma B:

s=0;
for (i=0; i<N; i++) {
   s+=a[i];
   s+=b[i];
}
```