

# CALCOLATORI ELETTRONICI B – 6 settembre 2007

**NOME:**

**COGNOME:**

**MATR:**

Scrivere chiaramente in caratteri maiuscoli a stampa

1. Si considerino, mostrati nelle figure riportate di seguito, il datapath e il diagramma a stati finiti che specifica l'unità di controllo secondo la tecnica multiciclo relativamente alle istruzioni MIPS lw, sw, beq, j ed alle istruzioni di tipo-R (il cui formato è riportato di seguito).

Si supponga che le operazioni atomiche che coinvolgono le unità funzionali principali richiedano:

Unità di memoria (lettura):	1 ns
Unità di memoria (scrittura):	2 ns
Register File (lettura e scrittura):	1 ns
Operazione ALU:	2 ns

e si assuma un carico di lavoro che preveda la seguente distribuzione delle istruzioni:

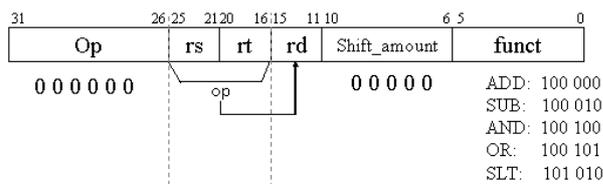
lw:	10 %
sw:	10 %
Tipo-R:	60 %
beq:	10 %
j:	10 %

Si chiede, riportando i passi significativi dell'analisi, di:

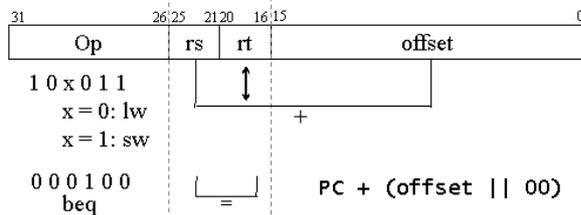
- Calcolare le prestazioni (in termini di tempo medio di esecuzione) della nota implementazione corrispondente al datapath e al diagramma degli stati riportati di seguito.
- Indicare una modifica (riportando le corrispondenti variazioni nel datapath e nel diagramma degli stati) in grado di migliorare le prestazioni.
- Calcolare il miglioramento delle prestazioni ottenuto.

[5]

## Promemoria formati delle istruzioni:



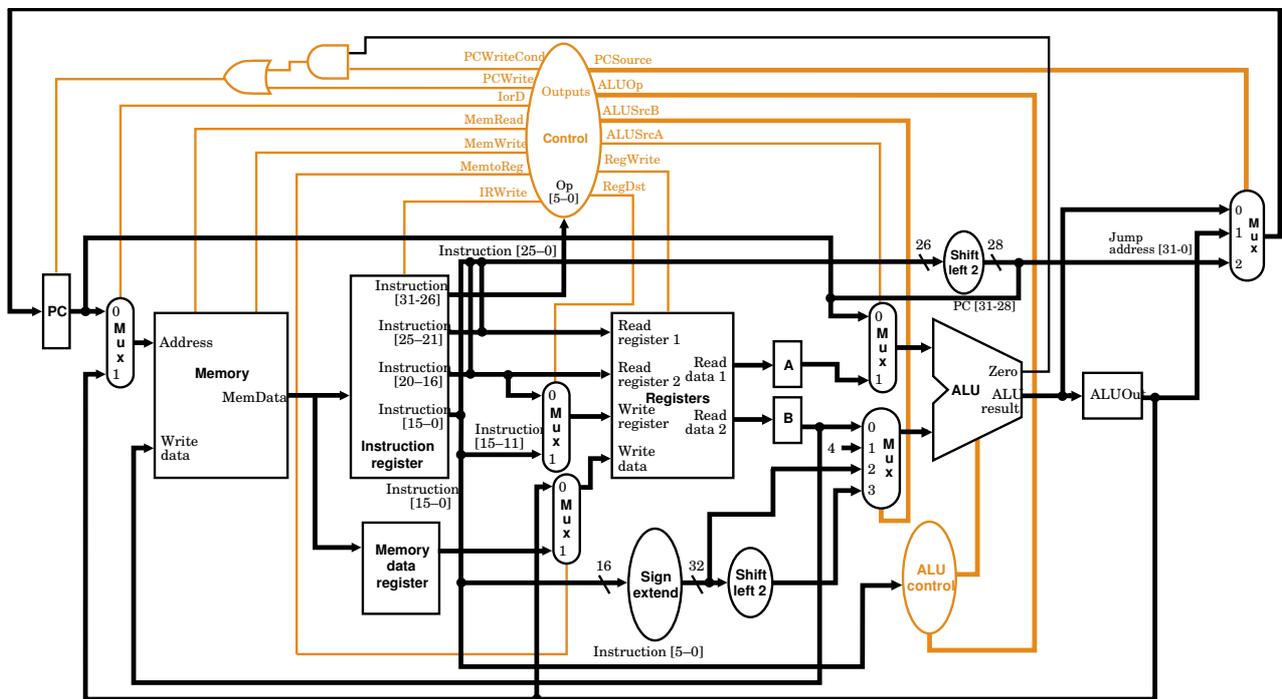
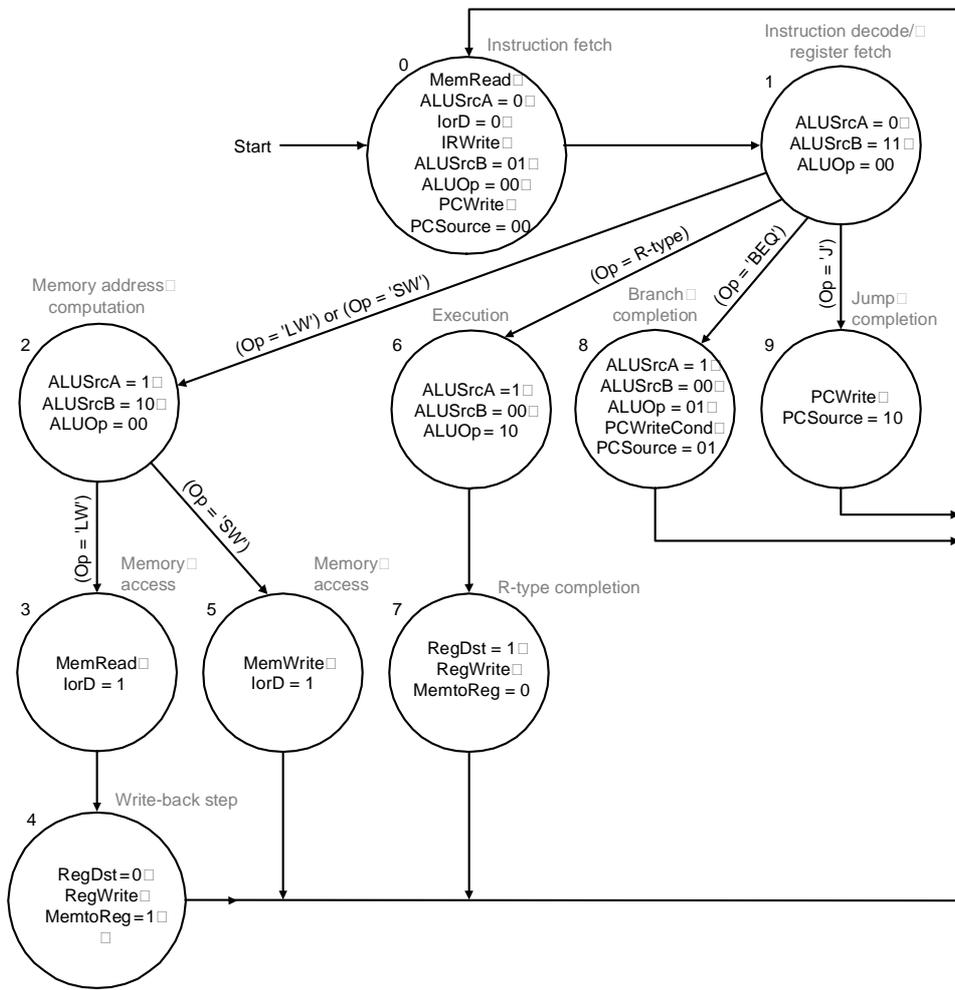
Aritmetiche:  
Tipo-R



lw, sw, beq:  
Tipo-I



J: Tipo-J





2. Nell'ambito di un processore basato su pipeline, si consideri l'implementazione di un'istruzione del tipo:

*addmem rt, offset(rs)*       $\ll rt \leftarrow rt + M[rs+offset]$

Si chiede di delineare una possibile implementazione di tale istruzione mediante uno schema che riporti:

- gli stadi della pipeline
- le unità funzionali coinvolte nell'istruzione *addmem*
- i collegamenti tra registri interstadio e unità funzionali che garantiscono il flusso dei dati (valori dei registri e offset) che interessano l'istruzione *addmem* attraverso gli stadi della pipeline.

[4]

3. Si illustri il motivo principale per cui, in una implementazione con pipeline, vengono tipicamente utilizzate due cache separate per le istruzioni e i dati.

Si consideri una implementazione con pipeline a 5 stadi (F, D, E, M, W) che disponga di una sola memoria cache per i dati e per le istruzioni. Si assuma un carico di lavoro che prevede la seguente distribuzione delle istruzioni MIPS:

lw:	20 %
sw:	20 %
Tipo-R:	40 %
salto:	20 %

Mostrando i passaggi fondamentali, si calcoli il CPI trascurando i miss di cache, le criticità sui dati e sui salti. [3]

4. Si consideri il seguente frammento di codice MIPS:

```
add  $s0, $s1, $s2
lw   $s0, 20($s1)
sub  $s0, $s0, $s0
lw   $s1, 20($s0)
sw   $s1, 20($s0)
```

Si consideri l'implementazione con pipeline a 5 stadi (F: Fetch, D: Decode, E: Execute, M: Mem, W: Write-Back). Si chiede di:

a) individuare in modo preciso tutte le dipendenze tra i dati

b) tracciare il diagramma temporale delle istruzioni (indicando esplicitamente le eventuali propagazioni e, per ognuna di esse, quale dato è propagato) in ognuna delle seguenti ipotesi:

- non è disponibile alcuna unità di propagazione

- è disponibile un'unità di propagazione verso lo stadio E

- è disponibile un'unità di propagazione verso lo stadio E ed una verso lo stadio M.

Nei diagrammi, si chiede di indicare il numero di cicli di penalità.

[6]

5. Nell'ambito delle tecniche di controllo con pipeline, si illustri brevemente (max 5-6 righe) la differenza tra tecniche di predizione statica e dinamica dei salti.
- Si consideri l'utilizzo di un BPB (Branch Prediction Buffer – Tabella di predizione delle diramazioni che a differenza del BTB non contiene l'indirizzo di destinazione predetto). Con riferimento agli stadi della pipeline in cui, rispettivamente, viene calcolata la condizione di salto e determinato l'indirizzo di destinazione, quale condizione deve essere necessariamente soddisfatta affinché l'utilizzo del BPB possa portare qualche vantaggio? Perché? [4]

6. E' dato un bus asincrono che collega un processore P e diversi dispositivi slave (quale ad esempio la memoria). Il bus è costituito da  $n$  linee dati,  $m$  linee indirizzi e dalle linee di controllo seguenti:

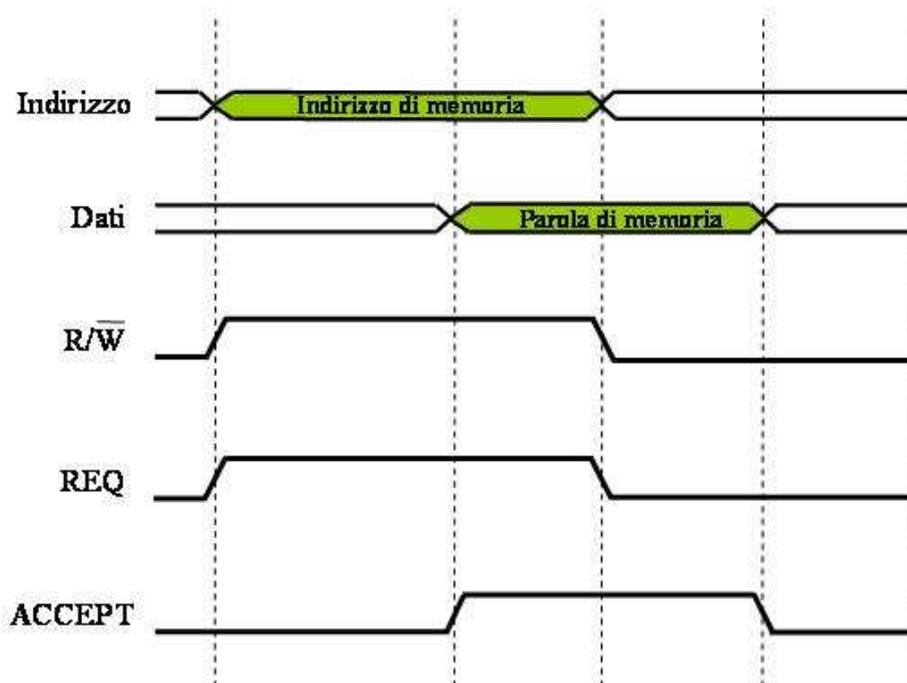
$R/\overline{W}$ : utilizzato dal processore P per segnalare una richiesta di lettura (se alto) o di scrittura (se basso).

REQ: utilizzato dal processore P per segnalare una richiesta di trasferimento.

ACCEPT: utilizzato dal dispositivo indirizzato per segnalare il completamento del trasferimento richiesto.

I segnali di controllo REQ e ACCEPT sono attivi a livello alto.

La figura seguente riporta l'evoluzione temporale di un'operazione di trasferimento di una parola dal dispositivo slave al processore P (lettura).



Si chiede di:

- Mostrare in un diagramma temporale come può avvenire un'operazione di scrittura (dal processore P al dispositivo slave), illustrando le relazioni tra i segnali.
- Specificare la macchina a stati finiti che controlla l'esecuzione, nel processore P (master), del protocollo di handshaking in scrittura di cui al punto a).
- Specificare la macchina a stati finiti che controlla l'esecuzione, nel dispositivo (slave), del protocollo di handshaking in scrittura di cui al punto a).







