

CALCOLATORI ELETTRONICI B – 13 luglio 2005

NOME:	COGNOME:	MATR:
Scrivere chiaramente in caratteri maiuscoli a stampa		

1. Si considerino, mostrati nelle figure riportate di seguito, il datapath ed il diagramma a stati finiti che specifica l'unità di controllo secondo la tecnica a multiciclo relativamente alle istruzioni MIPS lw, sw, beq, j ed alle istruzioni di tipo-R. Si vuole implementare una nuova istruzione del tipo:

$$\text{addmem } rt, \text{ offset}(rs) \quad // \text{ } rt \leftarrow rt + M[rs + \text{offset}]$$

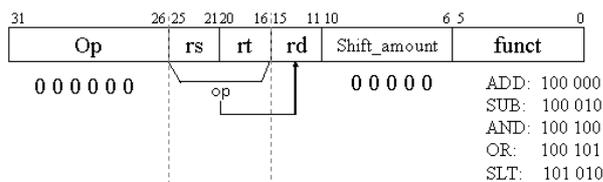
che somma al registro rt il contenuto della locazione di memoria indirizzata da rs+offset, dove offset è un numero a 16 bit con segno specificato nell'istruzione macchina (cfr. le istruzioni lw e sw).

Ricordando i tre formati di codifica delle istruzioni (riportati di seguito) si chiede di:

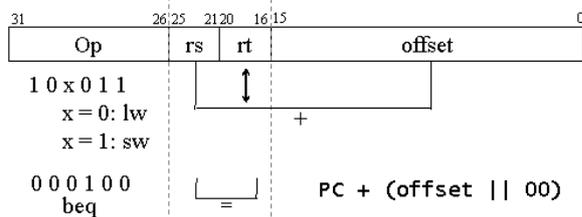
- riportare il formato della nuova istruzione macchina
- riportare, nella corrispondente figura, le modifiche necessarie al datapath
- estendere il diagramma degli stati per implementare la nuova istruzione

[6]

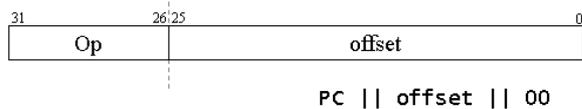
Promemoria formati delle istruzioni:



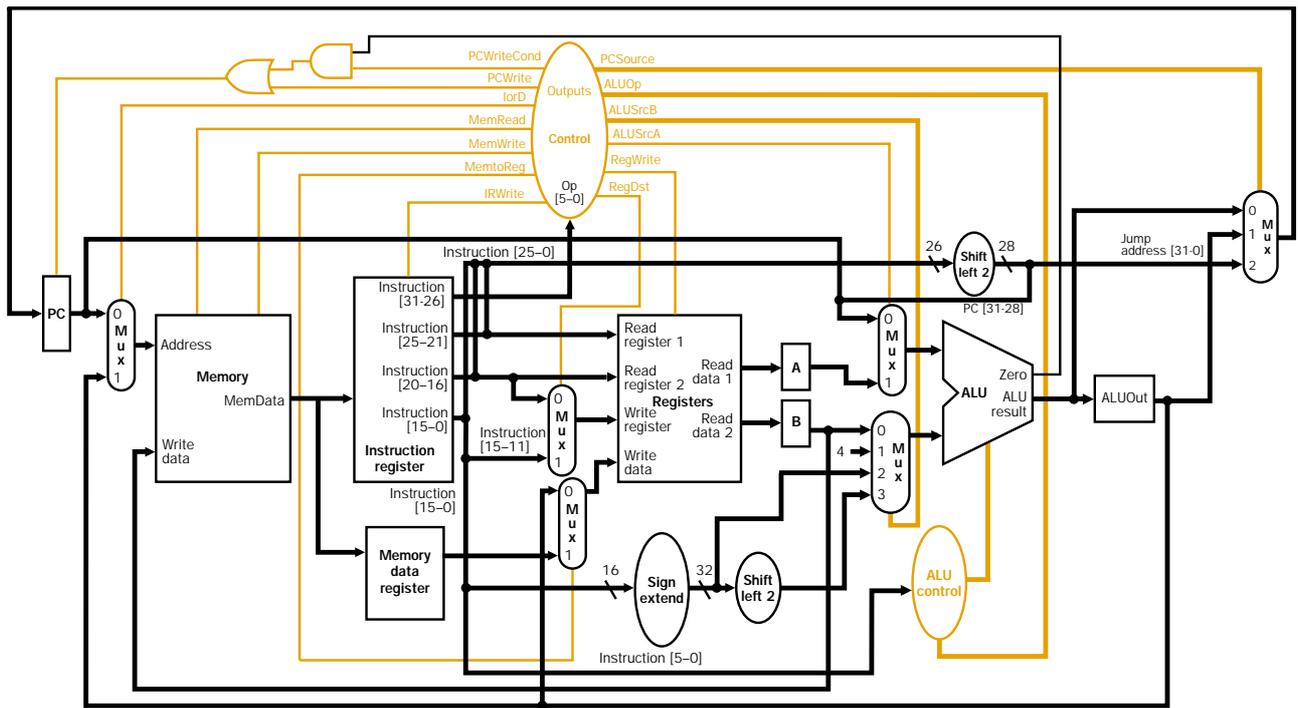
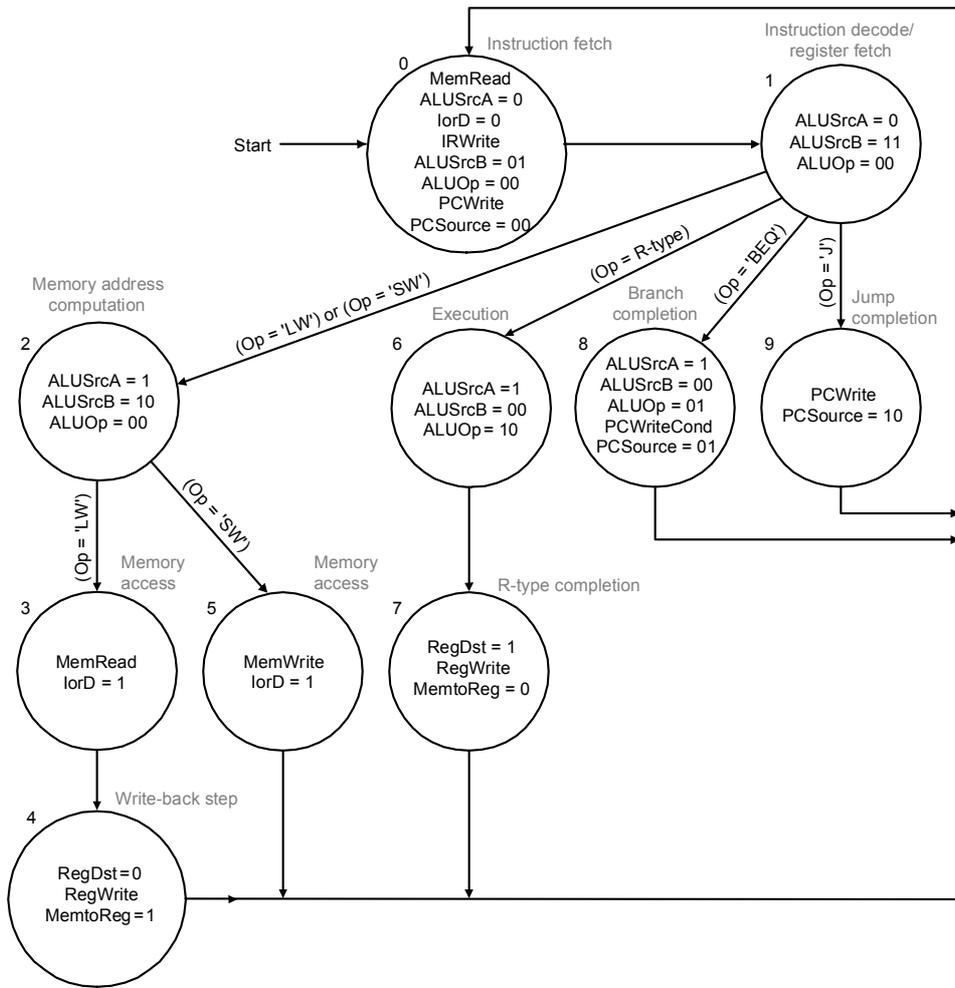
Aritmetiche:
Tipo-R



lw, sw, beq:
Tipo-I



J: Tipo-J



2. Si consideri l'esercizio risolto al punto precedente; si assuma un carico di lavoro che preveda la seguente distribuzione delle istruzioni:

lw:	20 %
sw:	20 %
Tipo-R:	30 %
beq:	10 %
j:	10 %
addmem:	10 %

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

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

Si confrontino le prestazioni (in termini di rapporto tra tempi di esecuzione) di un'ipotetica implementazione a singolo ciclo rispetto all'implementazione multiciclo individuata al punto precedente. Si riportino i passi significativi dell'analisi. [4]

NB: nel caso non si sia riusciti a risolvere l'esercizio precedente, si risolva questo punto facendo opportune ipotesi arbitrarie sulla nuova istruzione addmem.

3. Si consideri il seguente frammento di codice in un ipotetico linguaggio assembler (le istruzioni non commentate sono istruzioni MIPS):

```
lw $s0, ($s1)      \\ $s0 ← M[$s1]
add $t0, $s0, $s1
sub $t0, $t0, $s2
sub $s2, $t0, $s1
```

Si consideri l'implementazione con pipeline a 4 stadi (F: Fetch, D: Decode, E: Execute, W: Write-Back). Si chiede di:

- individuare tutte le dipendenze tra i dati
- tracciare il diagramma temporale delle istruzioni nell'ipotesi sia disponibile un'unità di propagazione verso lo stadio E (indicando esplicitamente le propagazioni e, per ognuna di esse, quale dato è propagato)

[4]

4. Si consideri il seguente frammento di codice MIPS:

```
lw $s0, 20($t1)
beq $s0, $s1, Ndb
add $t2, $t2, $t2
Ndb: add $t2, $t2, $t3
sub $s2, $s2, $s3
```

Si consideri un'implementazione tramite pipeline a 6 stadi in cui la decisione e l'esecuzione del salto beq avviene nel secondo stadio della pipeline. Per la gestione delle criticità sui salti, viene adottata la tecnica del salto ritardato.

Quanti slot di ritardo sono presenti? Perché?

Si indichi, motivando brevemente la soluzione proposta, come potrebbe essere riordinato il codice per gestire la criticità sul salto beq. [6]

5. Con riferimento alla memoria virtuale, si illustri sinteticamente (4-5 righe al massimo) la funzione del TLB. Si consideri il caso in cui il numero di pagina virtuale non sia contenuto nel TLB: è possibile dire che ciò provoca in ogni caso un'eccezione per mancanza di pagina (page-fault)? Perché?

Nel caso in cui la pagina fisica associata alla pagina virtuale sia presente in memoria, come viene calcolato l'indirizzo fisico a partire da quello virtuale? [4]

6. Con riferimento a due ipotetici segnali REQUEST (inviato dal master allo slave) e ACK (inviato dallo slave al master) si illustri sinteticamente un'ipotetica transazione in un bus asincrono utilizzando il protocollo di handshaking, evidenziando in modo preciso il significato dei segnali e le relazioni di causa-effetto tra le loro transizioni. [4]

7. Illustrare brevemente la differenza tra l'approccio per l'I/O "mappato in memoria" (memory mapped) e l'approccio che utilizza istruzioni speciali di I/O. [2]

