

CALCOLATORI ELETTRONICI B – 13 dicembre 2007

NOME:

COGNOME:

MATR:

Scrivere chiaramente in caratteri maiuscoli a stampa

1. Si considerino, mostrati nelle figure alla pagina seguente, 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 *Tipo-R*.

Si vuole implementare una nuova istruzione *call* che salta ad un indirizzo specificato mediante indirizzamento relativo al program counter PC (lo stesso indirizzamento usato da *beq*: costante a 16 bit che specifica lo scostamento in istruzioni rispetto a PC+4) e memorizza l'indirizzo dell'istruzione seguente in un registro. Ad esempio, l'esecuzione dell'istruzione

call \$s0, 100

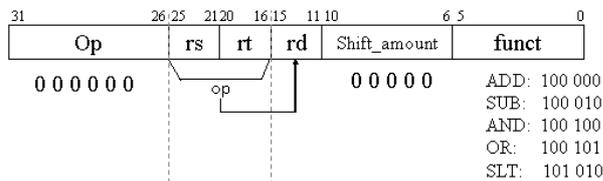
scrive in *\$s0* l'indirizzo dell'istruzione successiva a *call* e salta all'indirizzo ($PC + 4 + 400$).

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

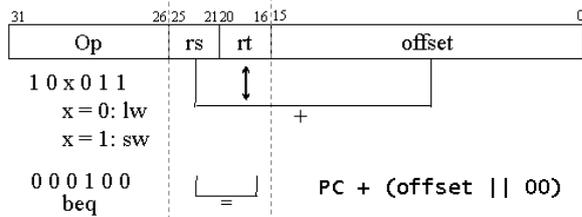
- specificare quale dei registri (*rs*, *rt* o *rd*) funge da operando dell'istruzione *call* (si cerchi per quanto possibile di minimizzare le modifiche al datapath)
- riportare il formato della nuova istruzione macchina
- riportare, nella corrispondente figura, le modifiche necessarie al datapath
- estendere il diagramma degli stati per implementare la nuova istruzione

[6]

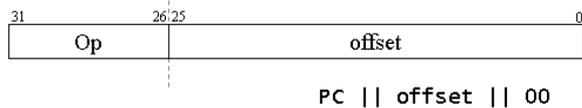
Promemoria formati delle istruzioni:



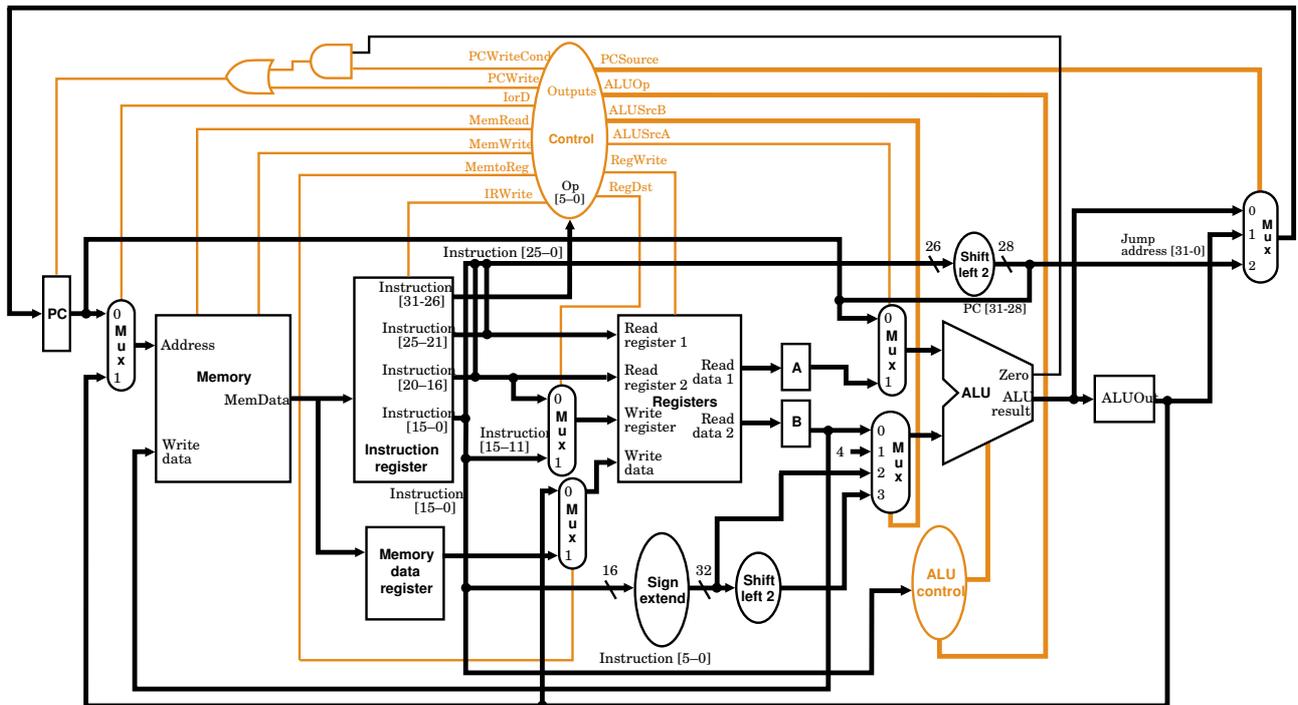
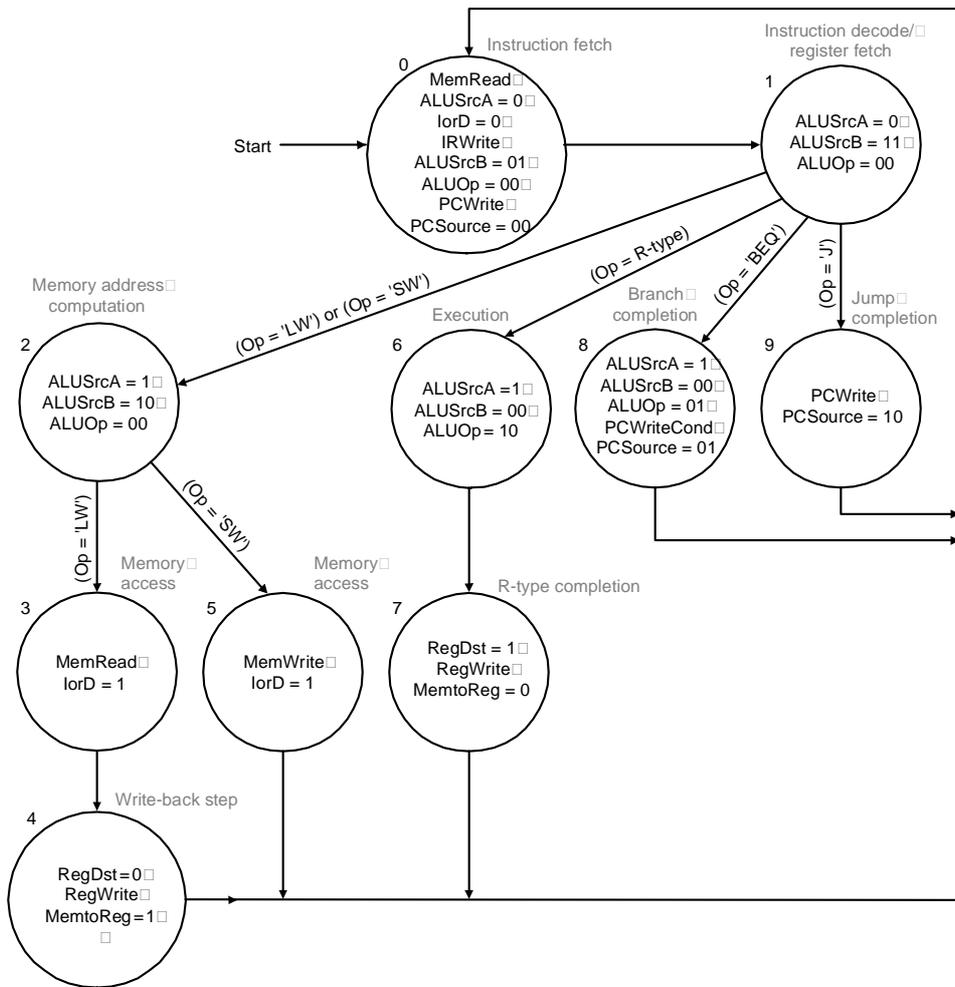
Aritmetiche:
Tipo-R



lw, sw, beq:
Tipo-I



J: Tipo-J



2. Si consideri il seguente frammento di codice MIPS:

```
add    $t0, $s1, $s1
sub    $t0, $t0, $s1
lw     $s2, 20($t0)
sw     $s2, 20($s1)
add    $s1, $s1, $s1
```

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]

3. Si consideri un processore MIPS, implementato tramite pipeline a 5 stadi, che disponga di una cache primaria e di una cache secondaria (distinte per i dati e le istruzioni). La cache primaria presenta le seguenti caratteristiche:

- numero di cicli di clock richiesti in caso di successo (hit): 1
- percentuale di successo (hit rate): 80% per le istruzioni, 70% per i dati
- penalità di fallimento: 10 cicli di clock

Per la cache secondaria valgono invece le seguenti condizioni:

- percentuale di successo (hit rate): 90% per le istruzioni, 80% per i dati
- cicli di clock per trasferire un blocco dalla cache secondaria alla cache primaria: 3

Si assuma un carico di lavoro che prevede la seguente distribuzione delle istruzioni MIPS:

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

Si supponga inoltre che:

- il 30% delle istruzioni Tipo-R siano seguite da istruzioni che ne utilizzano il risultato;
- il 20% delle istruzioni lw siano seguite da istruzioni Tipo-R che ne utilizzano il risultato;
- il 10% delle istruzioni lw siano seguite da istruzioni sw che ne utilizzano il risultato per il calcolo dell'indirizzo;
- il 15 % delle istruzioni lw siano seguite da istruzioni sw che ne utilizzano il risultato per immagazzinarlo in memoria.

[NB: per semplicità, si trascuri il caso in cui una sw utilizza il risultato della lw sia per il calcolo dell'indirizzo sia per immagazzinarlo in memoria]

Tenendo conto dei miss di cache e delle criticità sui dati, si calcoli il CPI (numero medio di cicli di clock per istruzione) ottenuto nei due casi seguenti:

- si dispone di un'unità di propagazione solo verso lo stadio E
- si dispone di un'unità di propagazione verso lo stadio E ed una verso lo stadio M. [6]

4. Si illustri sinteticamente la tecnica del salto ritardato per la gestione delle criticità sul controllo nell'implementazione con pipeline.
Se un'istruzione di salto aggiorna il program counter nel terzo stadio della pipeline, quanti slot di ritardo occorrono? Perché? [2]

5. Si consideri un processore con pipeline a 4 stadi in cui le istruzioni di salto condizionato hanno sintassi e significato riportati di seguito:

bz \$r1, (\$r2) // se \$r1=0 allora PC←PC+\$r2

Si ipotizzi che il salto sia implementato in modo da assicurare ragionevolmente una certa efficienza. In questo caso, quali vantaggi può comportare 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)? Si giustifichi la risposta. [2]

6. Si consideri un sistema di memoria virtuale con le seguenti caratteristiche:

- indirizzi virtuali di 40 bit
- dimensione delle pagine: 16 KB
- indirizzi fisici di 36 bit
- bit di controllo utilizzati per ciascuna riga della tabella delle pagine: 4

Si calcoli la dimensione della tabella delle pagine per ciascun processo, assumendo che le pagine virtuali siano tutte utilizzate e che la tabella non memorizzi gli indirizzi su disco. [3]

7. E' dato un bus sincrono che collega un processore P ad una memoria M e che consente il trasferimento di parole di memoria a blocchi. Il bus è dotato di linee distinte per dati e indirizzi. Inoltre, le linee di controllo coinvolte in un'operazione di trasferimento di un blocco di parole dalla memoria al processore (lettura) sono le seguenti:

$\overline{\text{READ}}$: utilizzato dal processore P per segnalare una richiesta di lettura di un blocco dalla memoria. Esso rimane attivo (a livello basso) fino a quando il processore desidera leggere ulteriori parole; viene disattivato dal processore nel corso della lettura dell'ultima parola per segnalare la fine del trasferimento.

$\overline{\text{WAIT}}$: utilizzato dalla memoria per segnalare, prima dell'inizio del trasferimento, la necessità di uno o più cicli di clock di attesa.

La pagina seguente riporta l'evoluzione temporale di un'operazione di lettura di un blocco costituito da due parole di memoria.

Si chiede di:

- Specificare la macchina a stati finiti che controlla l'esecuzione, nel processore P, del protocollo di lettura di un blocco di due parole, assumendo che la memoria possa richiedere l'attesa di un numero arbitrario di cicli.
- Specificare la macchina a stati finiti che controlla l'esecuzione, nella memoria M, del protocollo di lettura da parte del processore di un blocco di un numero generico di parole, assumendo che la memoria M necessiti di richiedere l'attesa di un ciclo di clock.

Le specifiche proposte devono essere coerenti con il diagramma che segue.

[6]

