

CALCOLATORI ELETTRONICI B – 14 aprile 2010

NOME:

COGNOME:

MATR:

Scrivere chiaramente in caratteri maiuscoli a stampa

1. 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]

2. 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 (in assenza della cache secondaria): 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:	25 %
sw:	15 %
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 10% delle istruzioni lw siano seguite da istruzioni Tipo-R che ne utilizzano il risultato;
- il 5% delle istruzioni lw siano seguite da istruzioni sw o lw che ne utilizzano il risultato solo per il calcolo dell'indirizzo;
- il 10% delle istruzioni lw siano seguite da istruzioni sw che ne utilizzano il risultato solo per immagazzinarlo in memoria.
- il 15% delle istruzioni lw siano seguite da istruzioni sw che ne utilizzano il risultato sia per il calcolo dell'indirizzo sia per immagazzinarlo in memoria.
- il 3% delle istruzioni lw sono seguite da istruzioni beq che ne utilizzano il risultato (si assuma che il confronto tra gli operandi sia effettuato al terzo stadio della pipeline).

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]

3. Con riferimento all'esercizio precedente, si considerino i seguenti tempi per le operazioni atomiche che coinvolgono le unità funzionali principali:

Lettura da memoria:	2 ns
Scrittura in memoria:	1 ns
Lettura Register File:	1 ns
Scrittura in Register File:	2 ns
Operazione ALU:	1 ns

Disponendo di unità di propagazione verso lo stadio E, si calcolino le prestazioni in termini di tempo di esecuzione medio per istruzione.

Si assuma ora che il tempo di decodifica non sia trascurabile: qual è il tempo massimo oltre il quale si ha un peggioramento delle prestazioni? Motivare la risposta. [3]

4. Si consideri, mostrato alla pagina seguente, il datapath corrispondente all'implementazione con tecnica pipeline a 5 stadi relativamente alle istruzioni MIPS lw, sw, beq ed alle istruzioni di tipo-R.

Si vuole implementare la nuova istruzione

add (r1), r2, r3 // M[r1] ← r2+r3

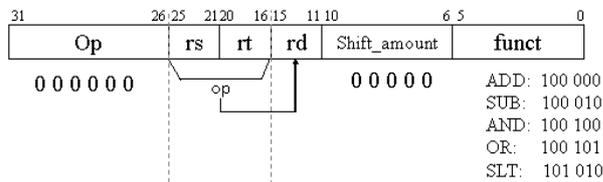
che carica nella parola di memoria di indirizzo *r1* la somma tra il valore di *r2* e quello di *r3*.

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

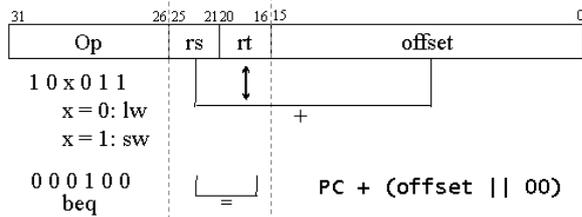
- riportare il formato della nuova istruzione macchina, specificando i campi destinati a r1, r2, r3
- riportare, nella corrispondente figura, le modifiche necessarie al datapath
- specificare come viene implementato il controllo della pipeline, riportando nel datapath l'unità di controllo e precisando (solo per la nuova istruzione) i segnali di controllo settati.

NB: è possibile modificare il Register File in modo da poter leggere 3 registri. [5]

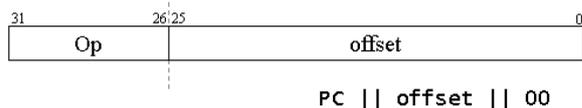
Promemoria formati delle istruzioni:



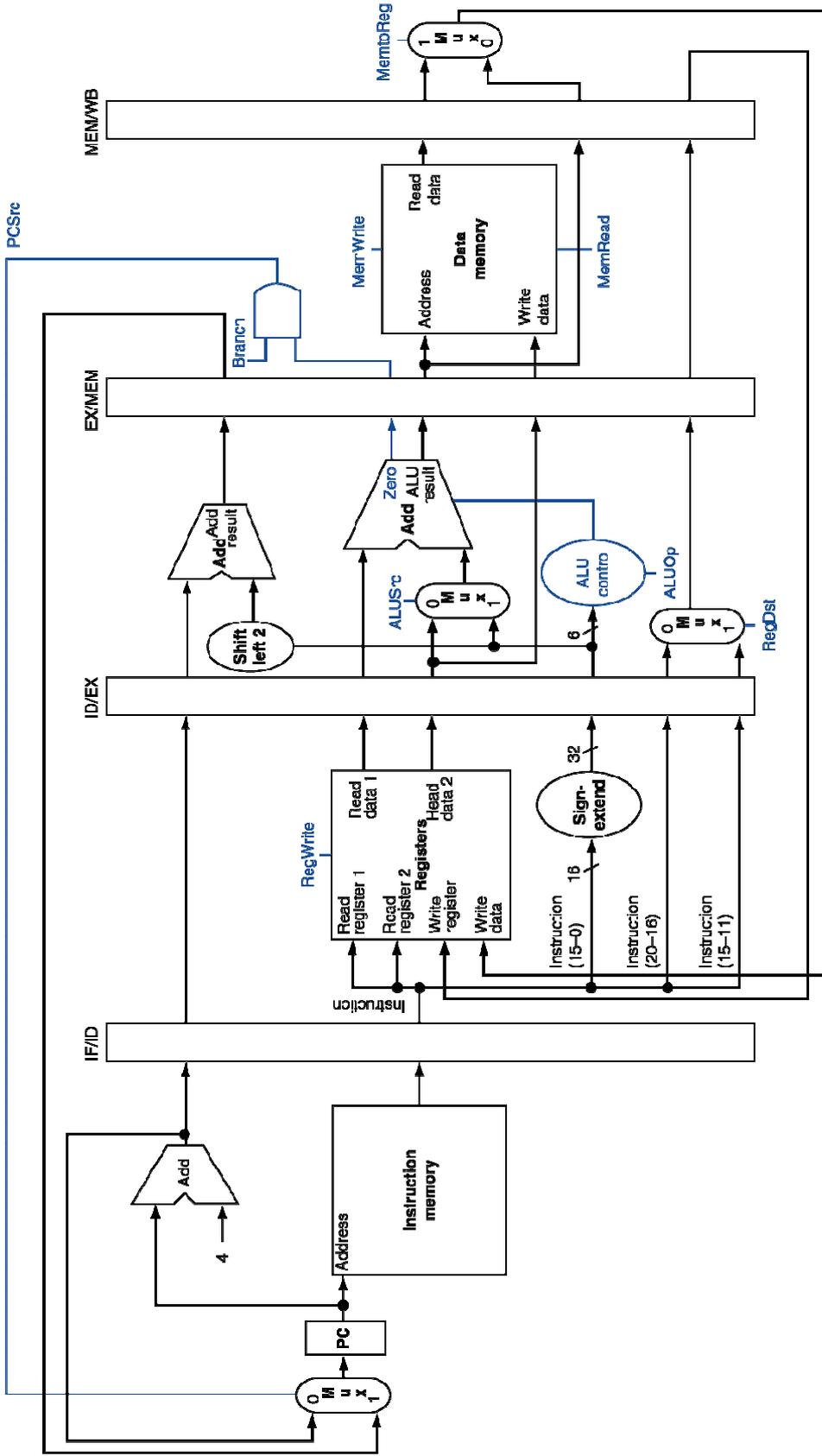
Aritmetiche:
Tipo-R



lw, sw, beq:
Tipo-I



J: Tipo-J



5. Con riferimento all'esercizio precedente, si supponga di voler implementare l'istruzione

add r1, offset (r2)

che somma al valore di r1 il valore di M[offset+r2] e pone il risultato in r1.

Considerando (come usualmente) le memorie, la ALU ed il register file "unità funzionali critiche", descrivere schematicamente come si può implementare tale istruzione. [3]

6. Si descriva la tecnica di predizione dinamica dei salti che utilizza BTB (Branch Table Buffer), illustrandone le funzioni, i vantaggi e sinteticamente la realizzazione. Si supponga che la destinazione del salto venga calcolata nel secondo stadio della pipeline, mentre la condizione di salto sia determinata nel quarto stadio. Assumendo che l'indirizzo di destinazione predetto sia corretto, qual è la penalità se il BTB predice di saltare ma la predizione è errata? E nel caso in cui anche l'indirizzo di destinazione predetto sia errato? E quale sarebbe utilizzando un BPB (Branch Prediction Buffer)? Giustificare precisamente le risposte. [4]

7. Illustrare dettagliatamente (aiutandosi con uno schema) la tecnica di arbitraggio dei bus mediante daisy chain, spiegando anche da che cosa dipende la priorità tra i dispositivi. [3]