CALCOLATORI ELETTRONICI B – 1 settembre 2009

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

1. Si consideri il seguente frammento di codice MIPS:

```
lw    $s0, 20($t1)
add   $s0, $t1, $t2
lw    $s0, 20($s0)
sw    $s0, 10($t1)
add   $s0, $s0, $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 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)
c) risolvere il punto precedente supponendo sia disponibile un'unità di propagazione verso lo stadio E ed una verso lo stadio M. [4]

2. Si consideri l'implementazione del MIPS con pipeline a 5 stadi in cui l'esecuzione dell'istruzione di salto condizionato *beq* viene anticipata al secondo stadio. L'hardware utilizzato richiede i seguenti tempi di esecuzione:

- prelievo istruzione e accesso alla memoria dati: 4 ns
- ogni altra operazione critica (ALU, decodifica, lettura e scrittura register file): 2 ns

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

lw:	30 %
sw:	10 %
formato-R:	40 %
beq:	15 %
j:	5 %

Si supponga che:

- il 25% delle istruzioni Tipo-R siano seguite da istruzioni che ne utilizzano il risultato nello stadio E, il 5% delle istruzioni Tipo-R siano seguite da istruzioni *beq* che ne utilizzano il risultato.
- il 15% delle istruzioni che seguono immediatamente *lw* utilizzino il risultato nello stadio E (ed eventualmente anche in M), il 10% utilizzino il risultato solo nello stadio M ed il 5% siano *beq* che ne utilizzano il risultato.
- il 2% delle istruzioni che seguono *lw* distanziate di due posizioni siano *beq* che ne utilizzano il risultato (*lw – istruzione – beq*)

Il processore utilizza una cache primaria distinta per i dati e le istruzioni, mentre non dispone di cache secondaria. La cache, che in caso di successo consente di accedere all'istruzione o al dato in un ciclo di clock, presenta le seguenti caratteristiche:
- percentuale di successo (hit rate): 80% per le istruzioni, 70% per i dati
- penalità di fallimento in scrittura: 5 cicli di clock
- penalità di fallimento in lettura: 10 cicli di clock

Trascurando le criticità sui salti (ovvero, considerando solo le criticità sui dati e i miss di cache), si calcoli il tempo medio di esecuzione per istruzione nei due casi seguenti:

- disponendo di un'unità di propagazione solo verso lo stadio E
- disponendo di un'unità di propagazione verso lo stadio E ed una verso lo stadio M.

Si giustifichino brevemente le risposte fornite. [5]

3. Con riferimento al seguente frammento di codice MIPS, indicare nel datapath in quale stadio si trovano le istruzioni durante il quinto ciclo di clock (in cui tutte le istruzioni sono in pipeline) e quali segnali di controllo sono attivi [si assuma siano presenti – non indicate – le unità di propagazione verso E ed M, inoltre che il salto condizionato non sia effettuato]:

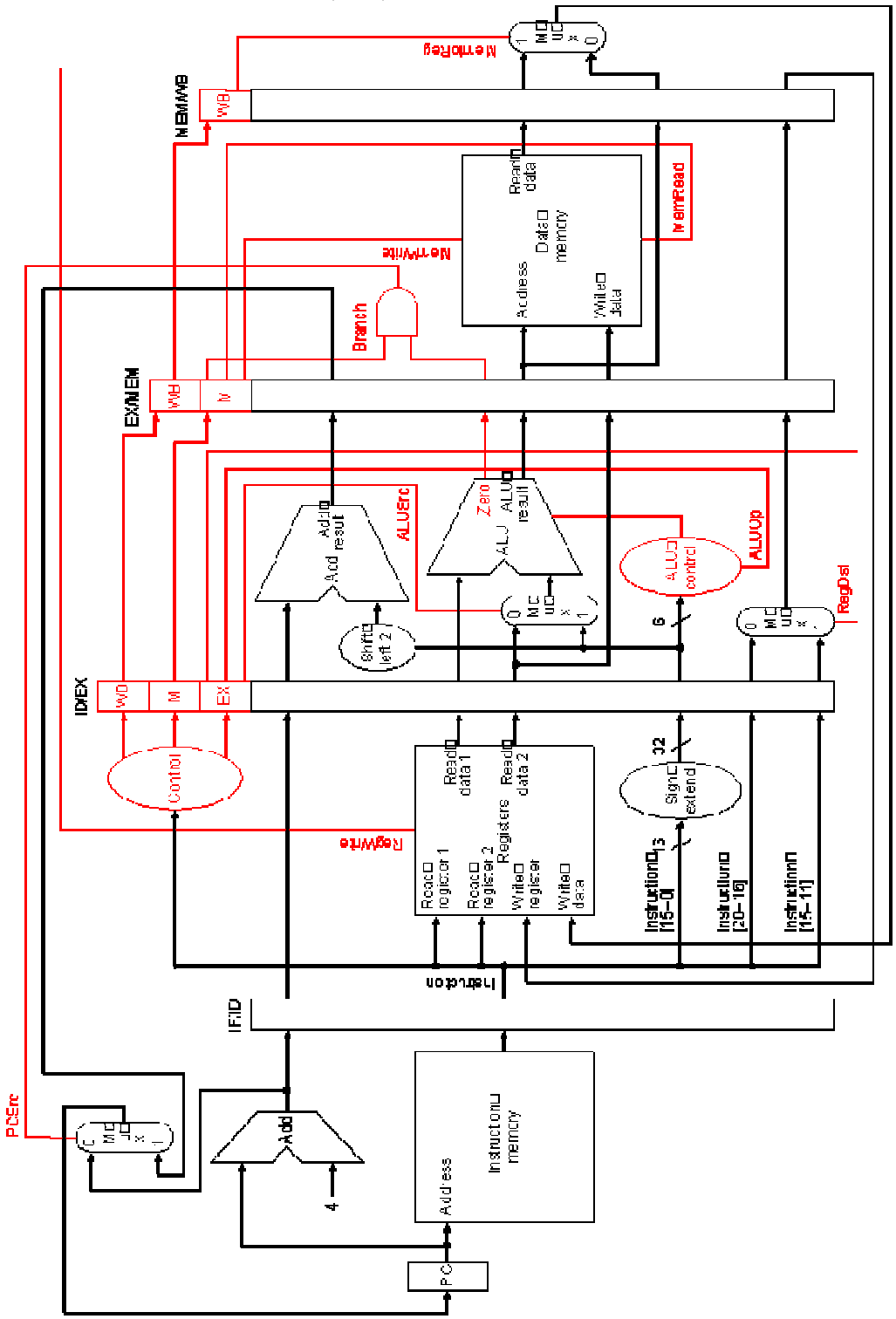
lw $t0, 20($s0)

sw $t0, 28($s0)

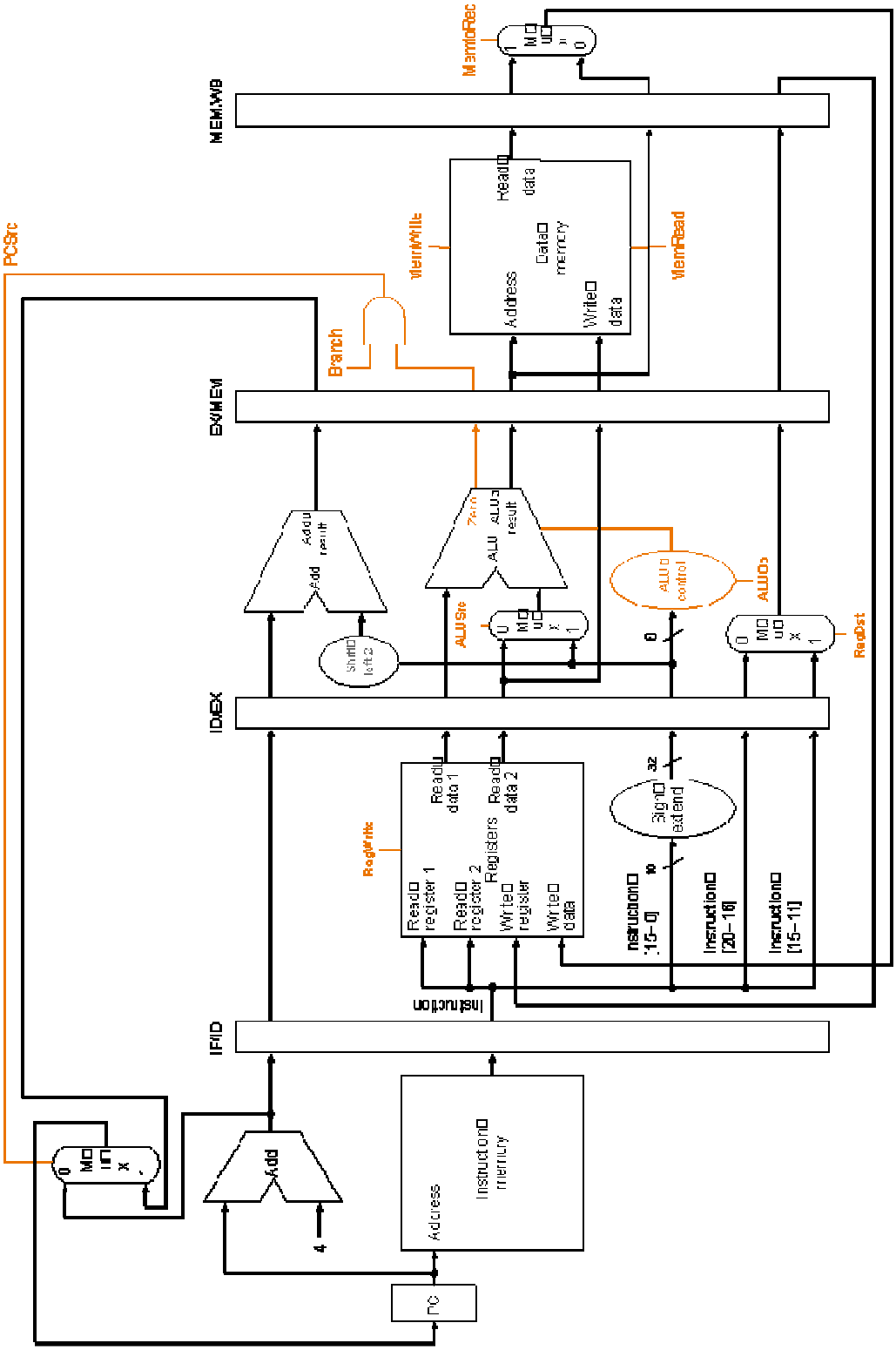
beq $t0, $t1, Dest

lw $t2, 20($t2)

add $t0, $t2, $t2 [4]



4. A partire dalla figura seguente, si disegnino schematicamente i circuiti che consentono di gestire un'eccezione per overflow aritmetico in istruzioni Tipo-R. Si richiede che, al verificarsi di un'eccezione, l'hardware scriva nel registro EPC l'indirizzo dell'istruzione responsabile dell'overflow, nel registro CAUSE un codice identificativo dell'overflow aritmetico e che il controllo sia trasferito ad una routine di servizio di indirizzo prefissato.[4]



5. Si consideri il seguente frammento di codice MIPS (i puntini indicano la presenza di un certo numero di istruzioni non specificate):

```
        lw    $t0, 20($t1)
        add   $t2, $t0, $t1
        beq   $t2, $t3, Dest
        add   $s4, $s4, $s2
        …
Dest:   …
```

Si consideri un'implementazione con pipeline a 5 stadi (Fetch – Decode – Execute – Mem – Write Back) dotata di una coda delle istruzioni e con le seguenti caratteristiche:
- la coda può contenere 4 istruzioni, che l'unità di prelievo è in grado di caricare contemporaneamente in un solo ciclo di clock;
- i salti condizionati vengono eseguiti dallo stadio O (apposita sotto-unità dell'unità di Fetch), parallelamente all'esecuzione delle istruzioni negli altri stadi;
- per i salti condizionati, si usa la semplice predizione di "salto non eseguito"
- l'hardware rende possibile la propagazione dei dati verso lo stadio O e verso lo stadio E.

Si chiede di:
- tracciare i diagrammi temporali nei due casi in cui la predizione per la beq sia corretta ed errata (assumere che le quattro istruzioni del frammento di codice siano caricate insieme nella coda delle istruzioni)
- determinare in entrambi i casi la penalità di salto. [5]

6. Si descriva la tecnica di predizione dinamica dei salti che utilizza BTB (Branch Table Buffer), illustrandone le funzioni e sinteticamente la realizzazione.
Qual è la principale differenza rispetto al BPB (Branch Prediction Buffer)? Se l'indirizzo di destinazione del salto viene calcolato in uno stadio successivo a quello in cui viene determinata la condizione di salto, è più conveniente usare un BPB o una tecnica di predizione statica? Motivare la risposta. [5]

7. Discutere il seguente diagramma rispetto al protocollo di handshaking per i bus asincroni. Il segnale $\overline{\text{REQ}}$ è un segnale di richiesta inviato dal master allo slave, mentre il segnale $\overline{\text{ACK}}$ è un segnale di conferma inviato dallo slave al master. Entrambi i segnali sono attivi a livello basso. [3]