

# Esame di Architetture – Canale AL – Prof. Sterbini – 17/6/13 – Compito A

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

---

## **Parte 1 (per chi non ha superato l'esonero)**

**Esercizio 1A.** Si ha il dubbio che in una partita di CPU a ciclo di clock singolo (vedi sul retro) la Control Unit sia rotta, producendo il segnale di controllo **MemWrite** attivo **se e solo se** è attivo il segnale di controllo **RegWrite**.

a) Si indichino *qui sotto* quali delle istruzioni base (**lw, sw, add, sub, and, or, xor, slt, beq, j**) funzioneranno male e perché.

b) si scriva *qui sotto* un breve programma assembly MIPS (senza pseudoistruzioni) che termina valorizzando il registro \$s0 con il valore 1 se il processore è guasto, altrimenti con 0. Si assume che RegDst sia asserito solo per le istruzioni di tipo R e che MemToReg sia asserito solo per l'istruzione lw.

---

**Esercizio 2A.** Considerate l'architettura MIPS a ciclo singolo in figura (diagramma sul retro).

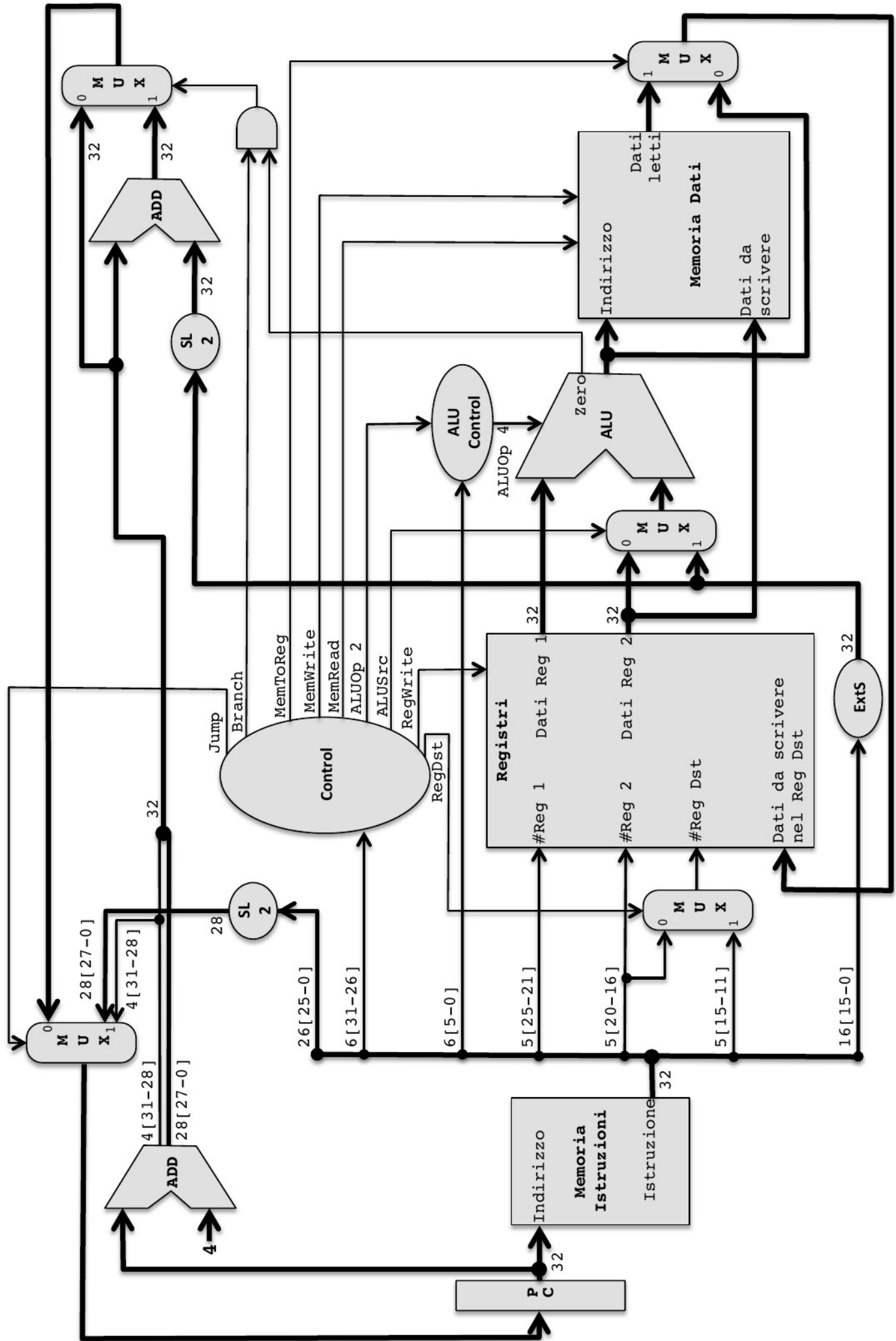
Si vuole aggiungere l'istruzione di tipo R **jalr rd, rs** che salta all'indirizzo contenuto nel registro **rs** e salva nel registro **rd** l'indirizzo della prossima istruzione. La codifica binaria ha i campi  $rt=0$  e  $func=010011$ .

1) modificate il diagramma mostrando gli eventuali altri componenti necessari a realizzare l'istruzione

2) indicate sul diagramma i segnali di controllo necessari a realizzare l'istruzione

3) supponendo che l'accesso alle memorie impieghi 133ns, l'accesso ai registri 66ns, le operazioni dell'ALU 200ns, e ignorando gli altri ritardi di propagazione dei segnali, indicate sul diagramma la durata totale del ciclo di clock per permettere l'esecuzione anche della nuova istruzione.

Implementazione ad un ciclo di clock di MIPS (solamente le istruzioni: add, sub, and, or, xor, slt, lw, sw, beq, j)



# Esame di Architetture – Canale AL – Prof. Sterbini – 17/6/13 – Compito A

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

## Parte 2 (per tutti)

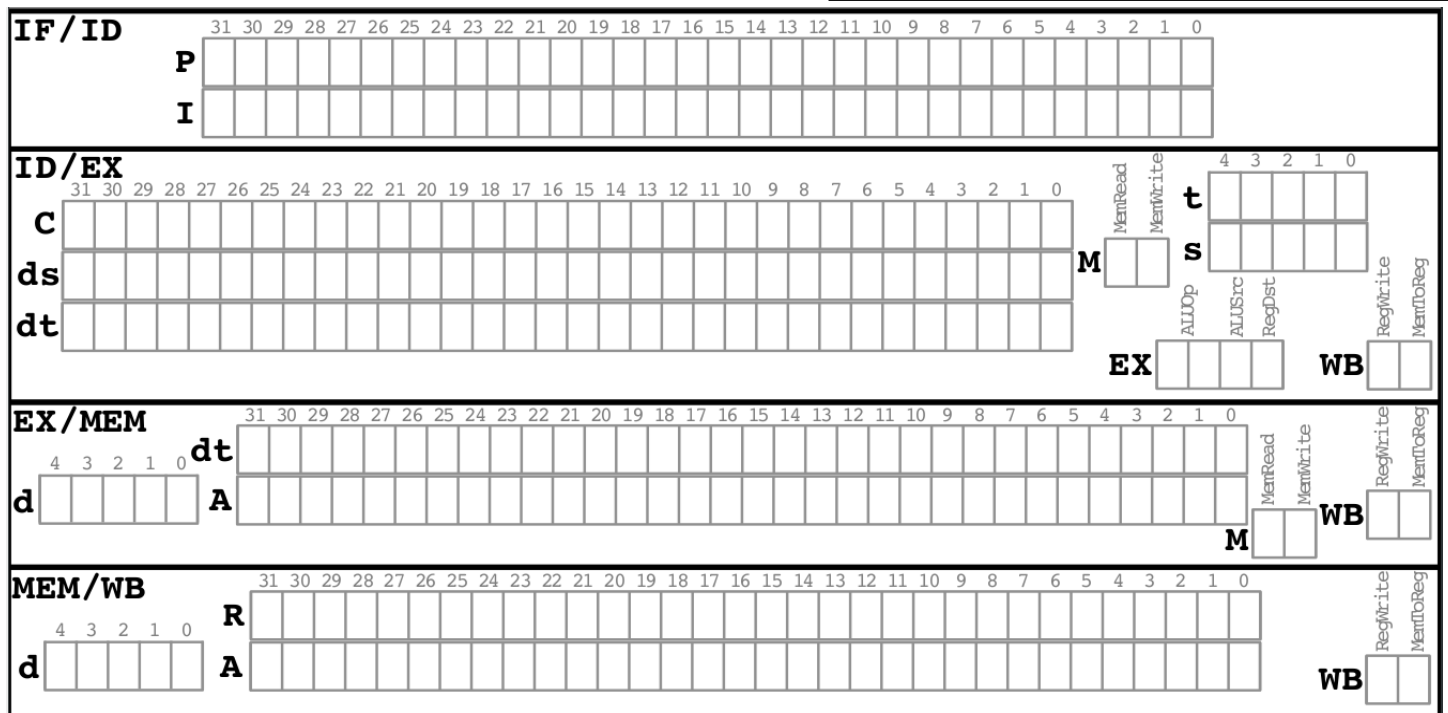
**Esercizio 3A.** Si consideri l'architettura MIPS con pipeline mostrata in figura (sul retro) ed il frammento di programma qui a destra che riempie ogni riga di una matrice 10x10 dei valori da 0 a 9.

Si indichino:

- 1) tra quali istruzioni sono presenti data hazard,
- 2) tra quali istruzioni sono presenti control hazard,
- 3) quanti cicli di clock sono necessari a eseguire il programma
- 4) quanti ne sarebbero necessari se il forwarding non esistesse
- 5) inserite nelle tabelle qui sotto quali sono le istruzioni contenute nei registri della pipeline alla fine del 9° ciclo di clock ed i relativi segnali di controllo (ove possibile invece dei valori binari usate gli mnemonici delle istruzioni e dei registri) (si assuma che tutte le istruzioni finiscono dopo la fase WB)

```

.data
0x00001000 matrice: .space 400
DIM:      .word 10 #lato matrice
.text
0x00004000 add $s0, $0, $0 #indice r
          add $s2, $0, $0 #indice c
          la  $s3, matrice
          lw  $s1, DIM($0)
loopRiga: beq $s0, $s1, end
loopCol:  beq $s2, $s1, nextCol
          mul $t1, $s0, $s1
          add $t1, $t1, $s2
          muli $t1, $t1, 4
          add $t1, $s3, $t1
          sw  $s2, 0($t1)
          addi $s2, $s2, 1
          j  loopCol
nextCol:  li  $s1, 0
          addi $s0, $s0, 1
          j  loopRiga
end:     lw  $v0, 1
    
```



**Esercizio 4A.** Si consideri un sistema dotato di due livelli di cache L1 ed L2 e di memoria virtuale con TLB.

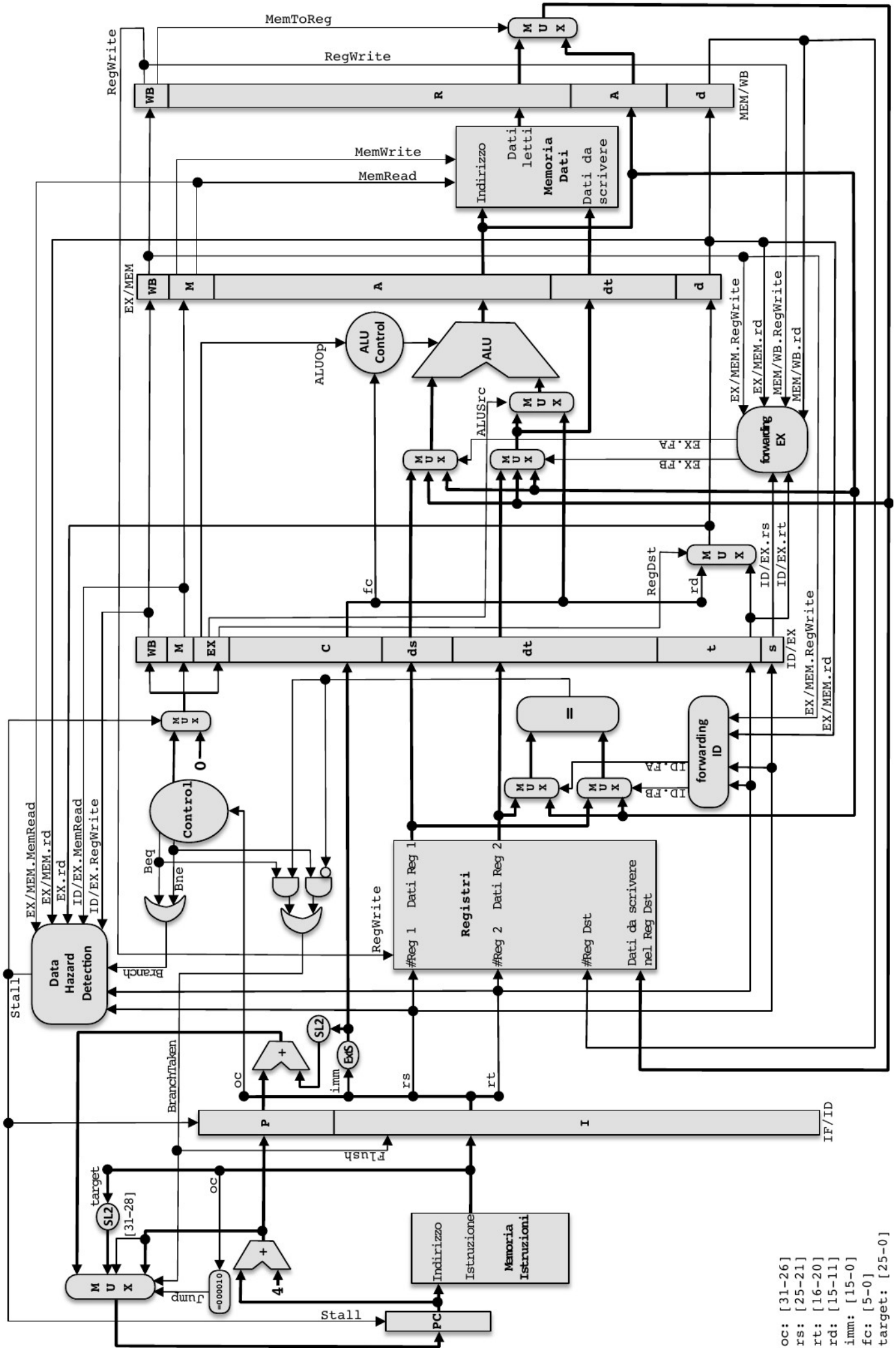
- L1 è una cache 2-way set-associative con 16 insiemi e blocchi grandi 8 word e strategia di rimpiazzo LRU.
- L2 è una cache direct-mapped con 64 linee e blocchi grandi 32 word.
- La memoria virtuale ha pagine da 1Kbyte ed un TLB fully associative da 8 linee con rimpiazzo LRU.

1) Supponendo che gli indirizzi virtuali siano da 32 bit (indirizzamento al byte) e che le cache L1 ed L2 operino su indirizzi virtuali (invece che su indirizzi fisici), e che all'inizio nessuna delle pagine dei dati siano in memoria, indicate quali dei seguenti accessi in memoria sono hit o miss in ciascuna delle due cache e in TLB e quali generano page fault:

**10, 28, 200, 540, 1000, 2020, 220, 980, 1950, 2200, 3050, 1010, 513, 510**

2) calcolate le dimensioni in bit delle tre cache L1, L2 e TLB.

Implementazione pipeline di MIPS (solamente le istruzioni: add, addi, sub, and, andi, or, ori, xor, xori, nor, slt, slti, lw, sw, beq, bne, j).



- oc: [31-26]
- rs: [25-21]
- rt: [16-20]
- rd: [15-11]
- imm: [15-0]
- fc: [5-0]
- target: [25-0]

# Esame di Architetture – Canale AL – Prof. Sterbini – 17/6/13 – Compito B

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

---

## **Parte 1 (per chi non ha superato l'esonero)**

**Esercizio 1B.** Si ha il dubbio che in una partita di CPU MIPS come quella in figura la Control Unit sia rotta, producendo il segnale di controllo **Branch** attivo **se e solo se** è attivo il segnale di controllo **RegDst**.

a) Si indichino qui sotto quali delle istruzioni base (**lw, sw, add, sub, and, or, xor, slt, beq, j**) funzioneranno male e perché.

b) si scriva qui sotto un breve programma assembly MIPS (senza pseudoistruzioni) che termina valorizzando il registro \$s0 con il valore 1 se il processore è guasto, altrimenti con 0. Si assume che RegDst sia asserito solo per le istruzioni di tipo R e che MemtoReg sia asserito solo per l'istruzione lw.

---

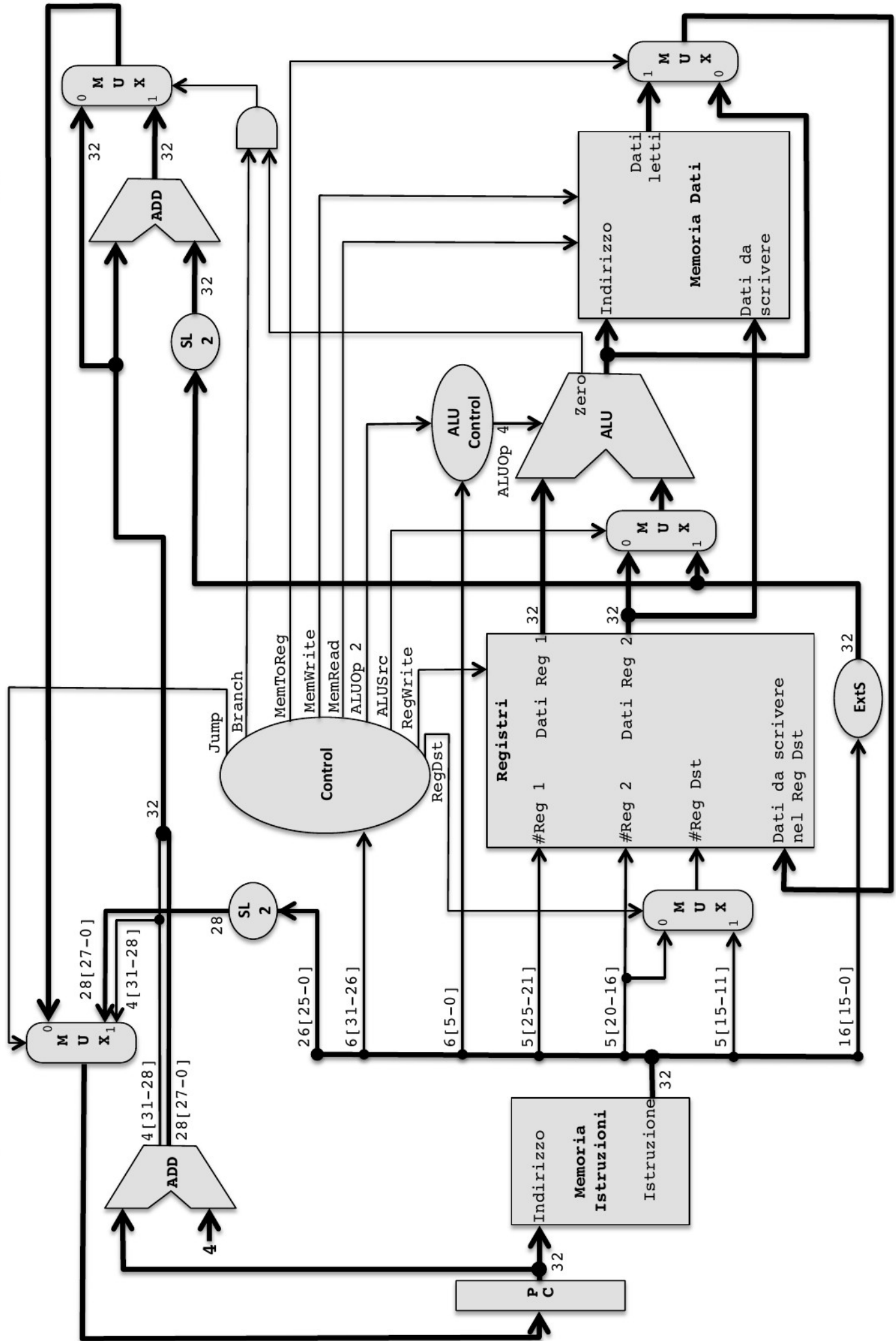
**Esercizio 2B.** Considerate l'architettura MIPS a ciclo singolo in figura (sul retro). Si vuole aggiungere l'istruzione di tipo I **beqal rs, rt, label** (branch if equal and link) cioè il salto condizionato che: solo quando esegue il salto memorizza nel registro **rt** l'indirizzo dell'istruzione successiva a questa.

1) modificate il diagramma mostrando gli eventuali altri componenti necessari a realizzare l'istruzione

2) indicate sul diagramma i segnali di controllo necessari a realizzare l'istruzione

3) supponendo che l'accesso alle memorie impieghi 100ns, l'accesso ai registri 50ns, le operazioni dell'ALU 150ns, e ignorando gli altri ritardi di propagazione dei segnali, indicate sul diagramma qual è la durata del ciclo di clock per permettere l'esecuzione anche della nuova istruzione.

Implementazione ad un ciclo di clock di MIPS (solamente le istruzioni: add, sub, and, or, xor, slt, lw, sw, beq, j)



# Esame di Architetture – Canale AL – Prof. Sterbini – 17/6/13 – Compito B

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

## Parte 2 (per tutti)

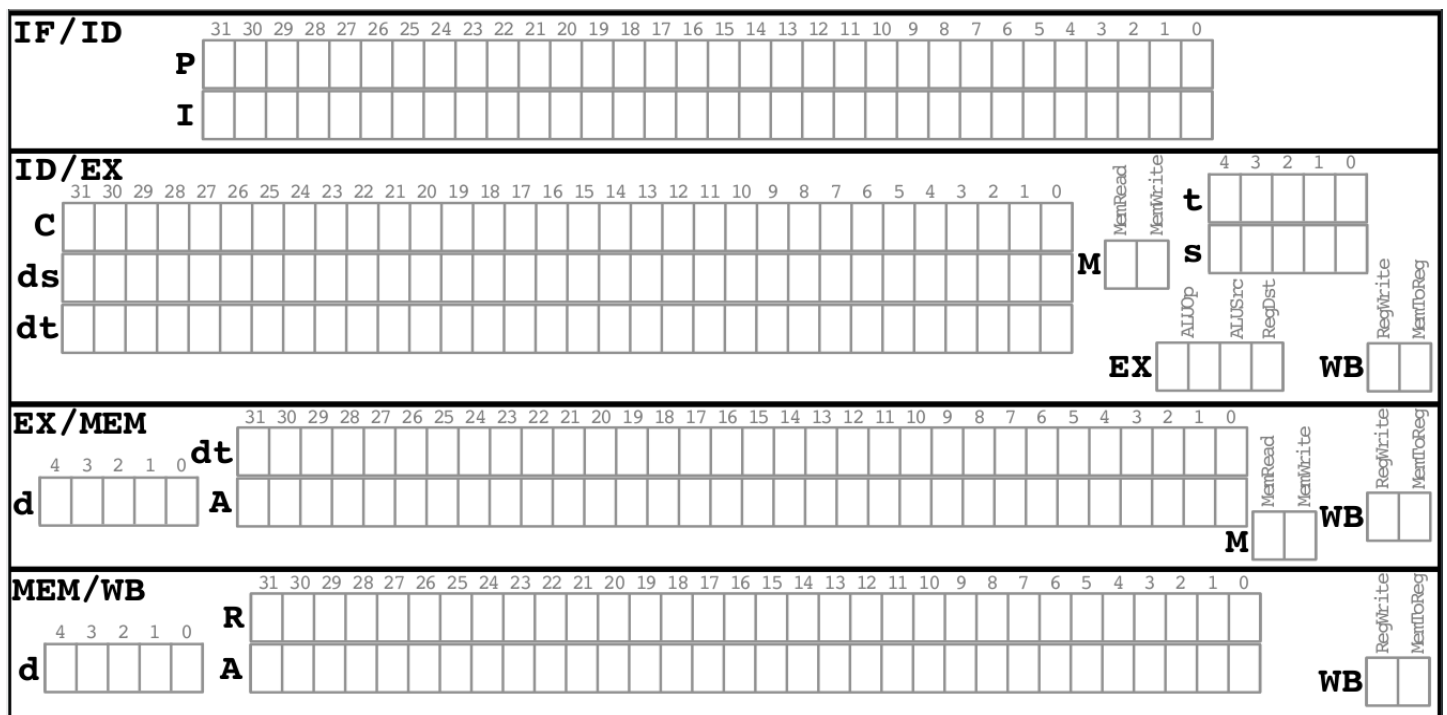
**Esercizio 3B.** Si consideri l'architettura MIPS con pipeline mostrata in figura (sul retro) ed il frammento di programma qui a destra che calcola la somma della diagonale di una matrice 10x10.

Si indichino:

- 1) tra quali istruzioni sono presenti data hazard,
- 2) tra quali istruzioni sono presenti control hazard,
- 3) quanti cicli di clock sono necessari a eseguire il programma
- 4) quanti ne sarebbero necessari se il forwarding non esistesse
- 5) inserite nelle tabelle qui sotto quali sono le istruzioni contenute nei registri della pipeline alla fine del 8° ciclo di clock ed i relativi segnali di controllo (ove possibile invece dei valori binari usate gli mnemonici delle istruzioni e dei registri) (si assuma che tutte le istruzioni finiscono dopo la fase WB e che i valori presenti sulla diagonale siano i numeri da 1 a 10)

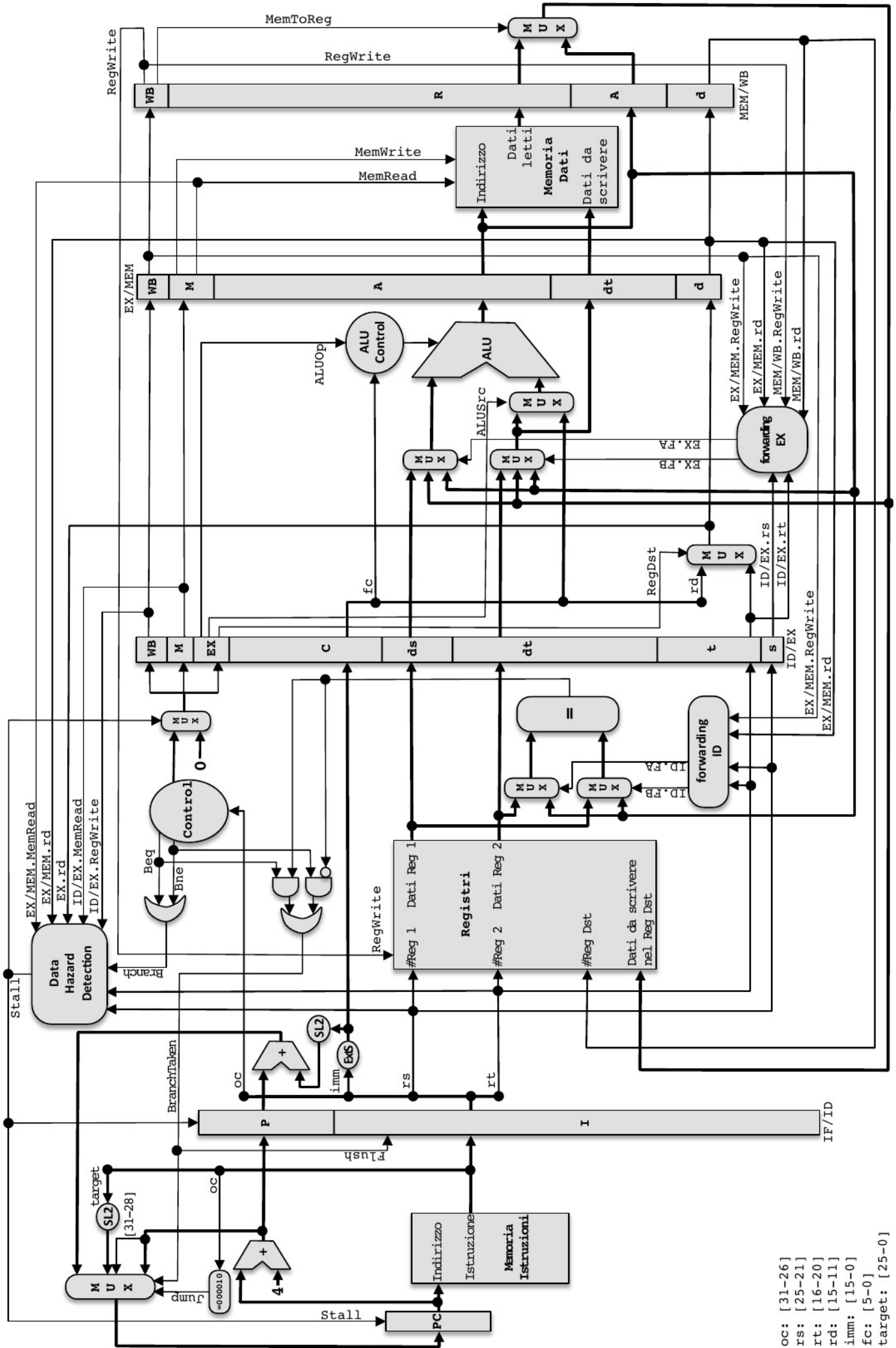
```

.data
0x00001000 matrice: .word . . . .
DIM:      .word 10
.text
0x00004000 add  $s2, $0, $0 #somma
          la   $s3, matrice
          lw   $s0, DIM($zero) # i
          lw   $s1, DIM($zero) # N
loop:    mul  $t1, $s0, $s1
          add $t1, $t1, $s0
          muli $t1, $t1, 4
          add $t1, $s2, $t1
          lw  $t1, 0($t1)
          add $s2, $s2, $t1
          addi $s0, $s0, -1
          beqz $s0, end
          j loop
end:     lw  $v0, 1
    
```



- Esercizio 4B.** Si consideri un sistema dotato di due livelli di cache L1 ed L2 e di memoria virtuale con TLB.
- L1 è una cache direct-mapped con 8 linee e blocchi grandi 4 word.
  - L2 è una cache 4-way set-associative con 16 insiemi e blocchi grandi 32 word e strategia di rimpiazzo LRU.
  - La memoria virtuale ha pagine da 2Kbyte ed un TLB fully associative da 4 linee con rimpiazzo LRU.
- 1) Supponendo che gli indirizzi virtuali siano da 32 bit (indirizzamento al byte) e che le cache L1 ed L2 operino su indirizzi virtuali (invece che su indirizzi fisici), e che all'inizio nessuna delle pagine dei dati siano in memoria, indicate quali dei seguenti accessi in memoria sono hit o miss in ciascuna delle due cache e in TLB e quali generano page fault:  
**10, 28, 200, 540, 1000, 2020, 220, 980, 1950, 2200, 3050, 1010, 513, 510**
  - 2) calcolate le dimensioni in bit delle tre cache L1, L2 e TLB.

Implementazione pipeline di MIPS (solamente le istruzioni: add, addi, sub, and, andi, or, ori, xor, xori, nor, slt, slti, lw, sw, beq, bne, j).



- oc: [31-26]
- rs: [25-21]
- rt: [16-20]
- rd: [15-11]
- imm: [15-0]
- fc: [5-0]
- target: [25-0]



## Esame di Architetture – Canale AL – Prof. Sterbini – 17/6/13 – Compito A

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

---

### Parte 3 (assembler)

#### Esercizio 5A.

Si realizzi il programma assembler che:

- legge un valore intero  $N < 1000$
- legge una sequenza di  $N$  interi, uno per riga
- li ordina in ordine crescente eliminando le ripetizioni
- li stampa in ordine, uno per riga, senza ripetizioni

---

## Esame di Architetture – Canale AL – Prof. Sterbini – 17/6/13 – Compito B

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

---

### Parte 3 (assembler)

#### Esercizio 5B.

Si realizzi il programma assembler ricorsivo che:

- legge due valori interi  $X$  e  $Y$  non negativi

- stampa il risultato della la funzione ricorsiva  $f(x, y) = \begin{cases} 1 & \text{se } x \text{ è pari} \\ 2 & \text{se } y \text{ è multiplo di 4} \\ f(x-1, y) + f(x, y-1) & \text{altrimenti} \end{cases}$

#### Esempi:

se  $X=2$  e  $Y=3$  viene stampato  $1 = f(2,3)$

se  $X=3$  e  $Y=3$  viene stampato  $5 = f(2,3) + f(3,2) = f(2,3) + f(2,2) + f(3,1) = f(2,3) + f(2,2) + f(2,1) + f(3,0) = 1 + 1 + 1 + 2$

se  $X=3$  e  $Y=5$  viene stampato  $3 = f(2,5) + f(3,4) = 1 + 2$