

# Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 13-4-2015 – Compito A

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

**Esercizio 1. (4/30)** Nell'architettura MIPS a ciclo singolo (fig. sul retro), quali sono i segnali di controllo ed i campi della istruzione se l'istruzione è **la \$12, -42 (\$11)** ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =
Jump =	MemRead =	ALUOp =	ALUSrc =	
rs =	rt =	rd =	immediate =	

**Esercizio 2. (6/30)** Si ha il dubbio che in alcune CPU MIPS come quella in figura la Control Unit sia rotta, producendo il segnale di controllo **RegWrite** attivo **se e solo se** è attivo il segnale **ALUSrc**. Si assume che RegDst sia asserito solo per le istruzioni di tipo R e che MemtoReg sia asserito solo per l'istruzione lw.

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 che termina valorizzando il registro \$s0 con il valore 1 se il processore è guasto, altrimenti con 0.

## **Esercizio 3. (10/30)**

1) Scrivete la procedura assembly che riceve come argomenti l'indirizzo di una stringa terminata da NULL, di massima lunghezza 100 e quello di una seconda area di memoria di 120 caratteri in cui mettere il risultato. La funzione deve costruire in memoria la seconda stringa terminata da NULL, contenente il testo iniziale seguito dalle parole "è pari" oppure "è dispari" a seconda che il testo contenga un numero pari o dispari di vocali (maiuscole o minuscole, NON accentate).

2) Scrivete (sempre qui) un programma main di esempio che chiama la procedura, allocando tutti i dati staticamente in memoria e che stampa il risultato.

## **Esercizio 4. (10/30)**

Si vuole aggiungere alla CPU in Figura (sul retro) l'istruzione **jump added (jadd)**, di tipo R e sintassi assembly

**jadd \$rs, \$rt** che salta alla istruzione che si trova all'indirizzo **(\$rs+\$rt)\*4**.

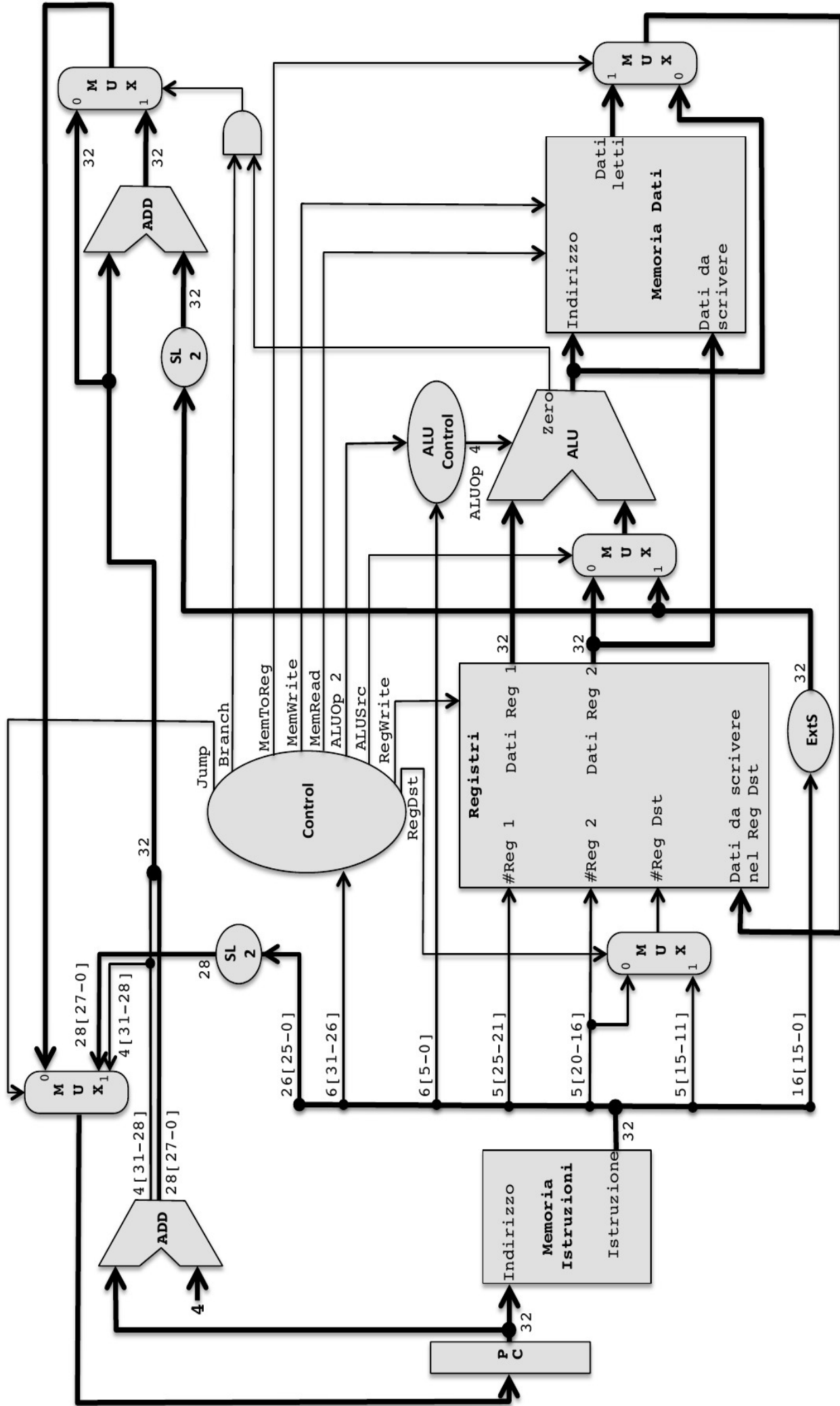
a) si disegnino *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU e sommatore (ecc) che ritenete necessari.

**Nota:** se necessario potete fare modifiche alle uscite/entrate dei componenti o dell'ALU.

b) si indichino *sullo schema della Figura* i valori dei segnali di controllo necessari all'istruzione.

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di **66ps**, l'accesso alla memoria impiega **133ps**, la ALU e gli adder impiegano **100ps** e ignorando gli altri ritardi di propagazione dei segnali, calcolate *sullo schema della Figura* il tempo di esecuzione minimo della nuova istruzione e indicate se è necessario aumentare il periodo di clock della CPU per poter svolgere questa nuova istruzione.

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



# Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 13-4-2015 – Compito B

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

**Esercizio 1. (4/30)** Nell'architettura MIPS a ciclo singolo (fig. sul retro), quali sono i segnali di controllo ed i campi della istruzione se l'istruzione è **subi \$20, \$21, 42**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =
Jump =	MemRead =	ALUOp =	ALUSrc =	
rs =	rt =	rd =	immediate =	

**Esercizio 2. (6/30)** Si ha il dubbio che in alcune CPU MIPS come quella della Figura la Control Unit sia rotta, producendo il segnale di controllo **Branch** attivo **se e solo se** è attivo il segnale di controllo **MemWrite**. Assumete che RegDst sia asserito solo per le istruzioni di tipo R e MemtoReg solo per l'istruzione lw.

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.

---

### Esercizio 3. (10/30)

1) Scrivete la funzione assembly che riceve come argomento l'indirizzo di una matrice di word quadrata M e **il suo lato** N, e che conta quante coppie di elementi in posizioni simmetriche rispetto alla diagonale hanno somma divisibile per 3. Ovvero conta quante volte accade che  $M[X,Y]+M[Y,X] \% 3 == 0$ , per tutti i valori X ed Y diversi (non guardate la diagonale). Attenti a non contare le coppie due volte.

2) Scrivete un esempio di main che usa la funzione e stampa il risultato, allocando staticamente tutti i dati necessari.

---

### Esercizio 4. (10/30)

Si vuole aggiungere alla CPU della Figura (sul retro) l'istruzione push (**push**), di tipo **R** e sintassi assembly **push \$rs** che aggiunge sullo stack il contenuto del registro **\$rs** ed aggiorna lo stack pointer.

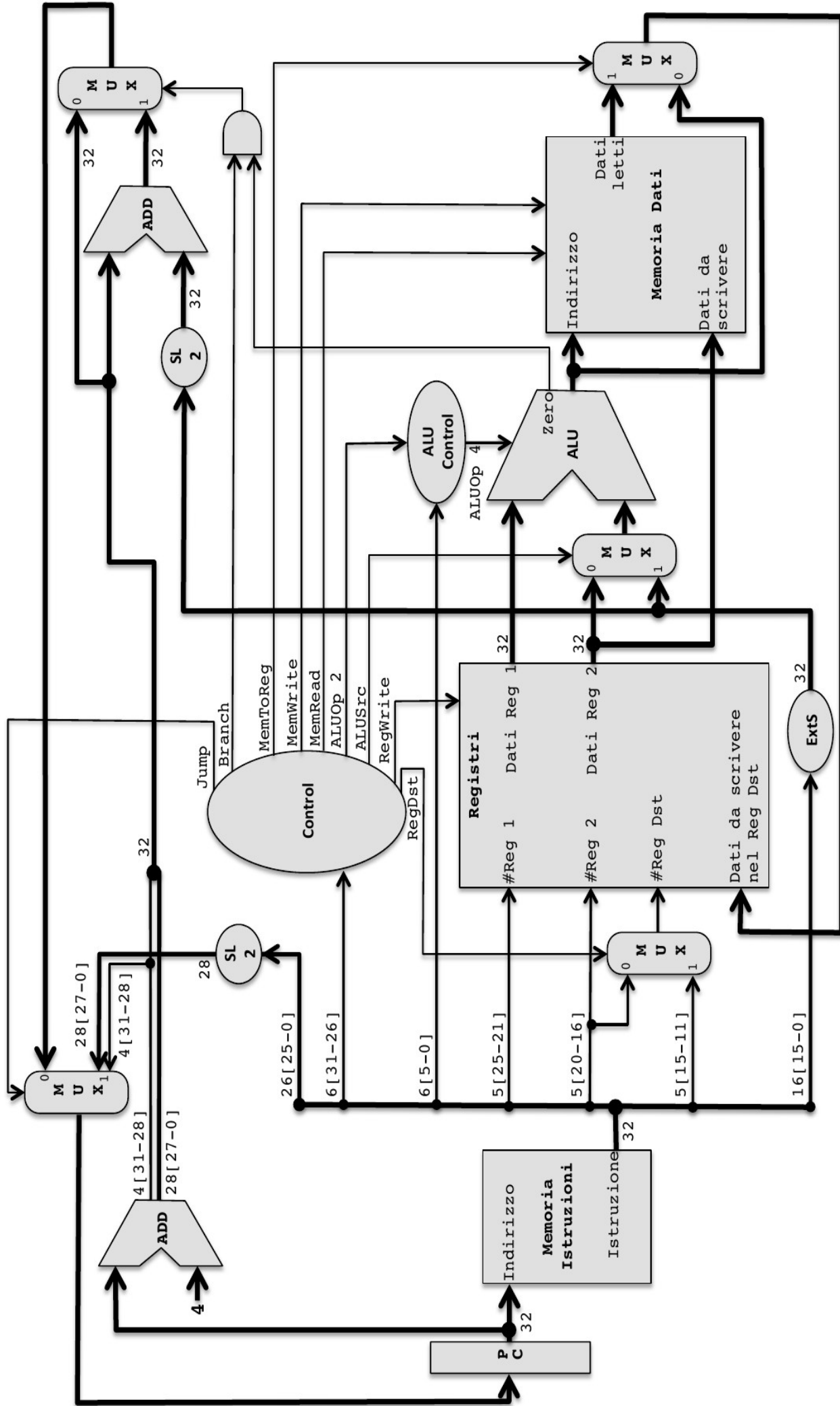
a) si disegnano *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU e sommatore (ecc) che ritenete necessari.

**Nota:** se necessario potete fare modifiche alle uscite/entrate dei componenti o dell'ALU.

b) indicate *sullo schema della Figura* i valori dei segnali di controllo necessari a eseguire l'istruzione.

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di **25ps**, l'accesso alla memoria impiega **50ps**, la ALU e i sommatore impiegano **150ps** e ignorando gli altri ritardi, calcolate *sullo schema della Figura* il tempo di esecuzione minimo della nuova istruzione e indicate se è necessario aumentare il periodo di clock della CPU per poter svolgere questa nuova istruzione.

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



# Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 13-4-2015 – Compito C

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

**Esercizio 1. (4/30)** Nell'architettura MIPS a ciclo singolo (fig. sul retro), quali sono i segnali di controllo ed i campi della istruzione se l'istruzione è **sw \$10, -42**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =
Jump =	MemRead =	ALUOp =	ALUSrc =	
rs =	rt =	rd =	immediate =	

**Esercizio 2. (6/30)** Si ha il dubbio che in alcune CPU MIPS come quella della Figura la Control Unit sia rotta, producendo il segnale di controllo **RegWrite attivo se e solo se NON è attivo il segnale ALUSrc**. Assumete che RegDst sia asserito solo per le istruzioni di tipo R e MemtoReg solo per l'istruzione lw.

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 che termina valorizzando il registro \$s0 con il valore 1 se il processore è guasto, altrimenti con 0.

---

### Esercizio 3. (10/30)

1) Scrivete la funzione assembly che riceve come argomento l'indirizzo di un testo unicode (16 bit per carattere) terminato da 0x0000, che conta quante parole separate da spazio (0x0020) ci sono nel testo.

2) Scrivete un esempio di main che usa la funzione e stampa il risultato, allocando staticamente tutti i dati necessari.

---

### Esercizio 4. (10/30)

Si vuole aggiungere alla CPU della Figura (sul retro) l'istruzione *branch if greater than* (**bgt**), di tipo I e sintassi assembly **bgt \$rs,\$rt,label** che salta all'indirizzo indicato da **label** solo se **\$rs > \$rt**.

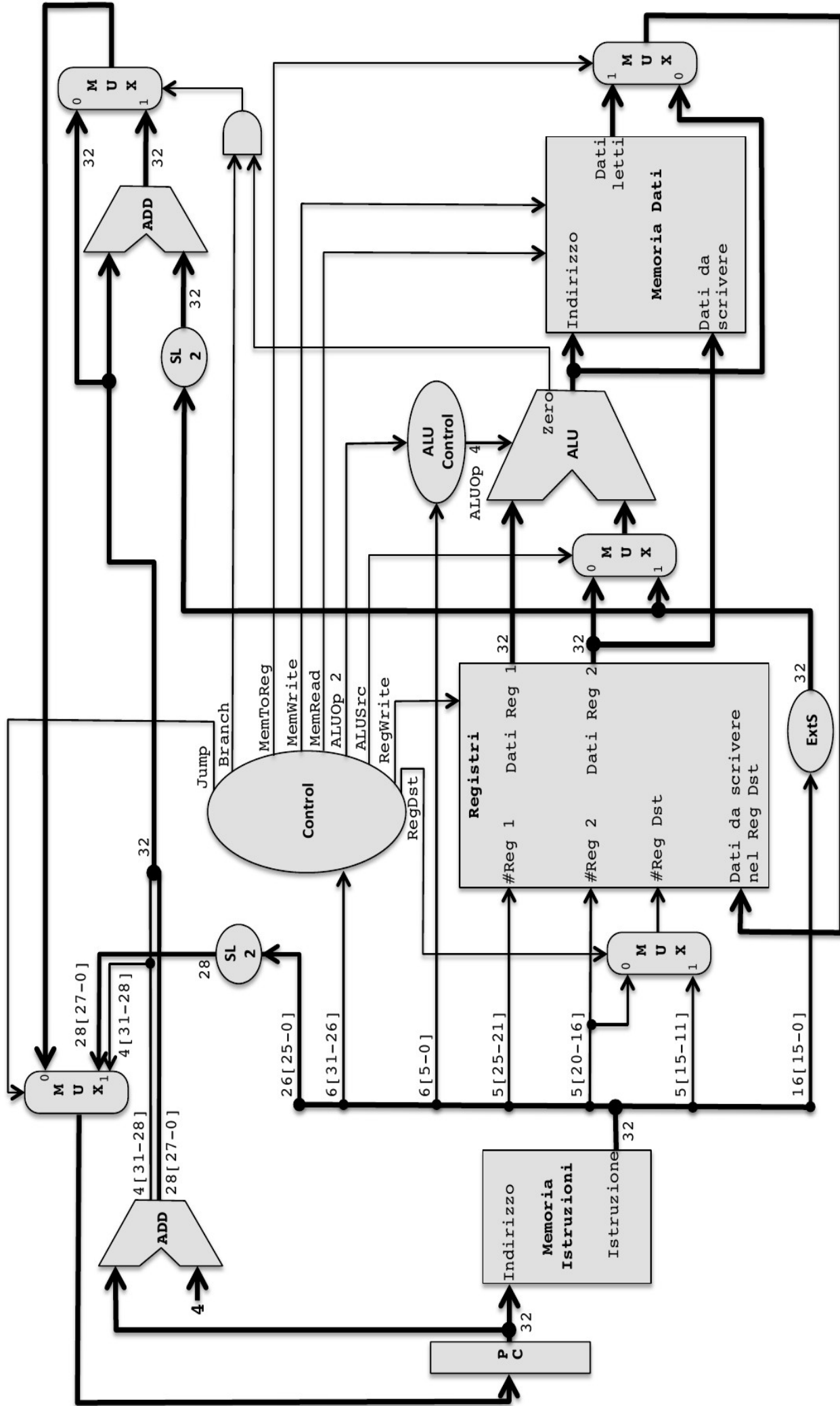
a) si disegnano *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU e sommatore (ecc) che ritenete necessari.

**Nota:** se necessario potete fare modifiche alle uscite/entrate dei componenti o dell'ALU.

b) indicate *sullo schema della Figura* i valori dei segnali di controllo necessari a eseguire l'istruzione.

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di **33ps**, l'accesso alla memoria impiega **66ps**, la ALU e i sommatore impiegano **100ps** e ignorando gli altri ritardi, calcolate *sullo schema della Figura* il tempo di esecuzione minimo della nuova istruzione e indicate se è necessario aumentare il periodo di clock della CPU per poter svolgere questa nuova istruzione.

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



# Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 13-4-2015 – Compito D

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

**Esercizio 1. (4/30)** Nell'architettura MIPS a ciclo singolo (fig. sul retro), quali sono i segnali di controllo ed i campi della istruzione se l'istruzione è **beqz \$20, 42**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =
Jump =	MemRead =	ALUOp =	ALUSrc =	
rs =	rt =	rd =	immediate =	

**Esercizio 2. (6/30)** Si ha il dubbio che in alcune CPU MIPS come quella della Figura la Control Unit sia rotta, producendo il segnale di controllo **Branch attivo se e solo se è attivo il segnale di controllo ALUSrc**. Assumete che RegDst sia asserito solo per le istruzioni di tipo R e MemtoReg solo per l'istruzione lw.

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.

### Esercizio 3. (10/30)

1) Scrivete la funzione assembly che riceve come argomenti l'indirizzo di una matrice di half-word quadrata M e **il suo lato** N, e che conta quante coppie di elementi in posizioni simmetriche rispetto al centro della matrice hanno prodotto positivo. Ovvero conta quante volte accade che  $M[X,Y]*M[X',Y'] > 0$ , con  $X'=N-1-X$  e  $Y'=N-1-Y$ . Attenti a non contare le coppie due volte.

2) Scrivete un esempio di main che usa la funzione e stampa il risultato, allocando staticamente tutti i dati necessari.

### Esercizio 4. (10/30)

Si vuole aggiungere alla CPU della Figura (sul retro) l'istruzione pop (**pop**), di tipo **R** e sintassi assembly **pop \$rd** che toglie dallo stack una word e la mette in **\$rd** ed aggiorna lo stack pointer.

a) si disegnino *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU e sommatore (ecc) che ritenete necessari.

**Nota:** se necessario potete fare modifiche alle uscite/entrate dei componenti o dell'ALU.

b) indicate *sullo schema della Figura* i valori dei segnali di controllo necessari a eseguire l'istruzione.

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di **25ps**, l'accesso alla memoria impiega **75ps**, la ALU e i sommatore impiegano **100ps** e ignorando gli altri ritardi, calcolate *sullo schema della Figura* il tempo di esecuzione minimo della nuova istruzione e indicate se è necessario aumentare il periodo di clock della CPU per poter svolgere questa nuova istruzione.

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

