Architettura degli elaboratori – MZ – 16-4-18 – Compito A

Cognome e Nome: _____ **Matricola:** _____

Esercizio 1A. Considerate l'architettura MIPS a ciclo singolo in Figura (sul retro). Qual è il valore dei segnali che partono dalla unità di controllo e dei diversi campi della istruzione se l'istruzione in esecuzione è:
`lw $11, 4241` ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	**Branch =**	**Memtoreg =**	**RegWrite =**	**MemWrite =**
Jump =	**MemRead =**	**ALUOp =**	**ALUSrc =**	**Func =**
rs =	**rt =**	**rd =**	**Imm. =**	**Shamt =**

Esercizio 2A. 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 **MemToReg** attivo **se e solo se** è attivo il segnale **MemWrite**.
NOTA: Si assume che tutti i valori don't care valgano 0.
a) Indicate *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j, di tipo R immediate**) funzioneranno male e in che modo cambia il loro comportamento.

b) scrivete *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 3A. Si vuole aggiungere alla CPU della Figura l'istruzione *jump vectorized* (`j_vect`), di tipo I, e sintassi assembly `j_vect` $rs, ***indice*** (salto vettorizzato) che ha in $rs l'indirizzo iniziale di un vettore di word che contiene gli indirizzi di destinazione del salto e che salta all'indirizzo indicato dall'elemento del vettore che ha indice **indice**.

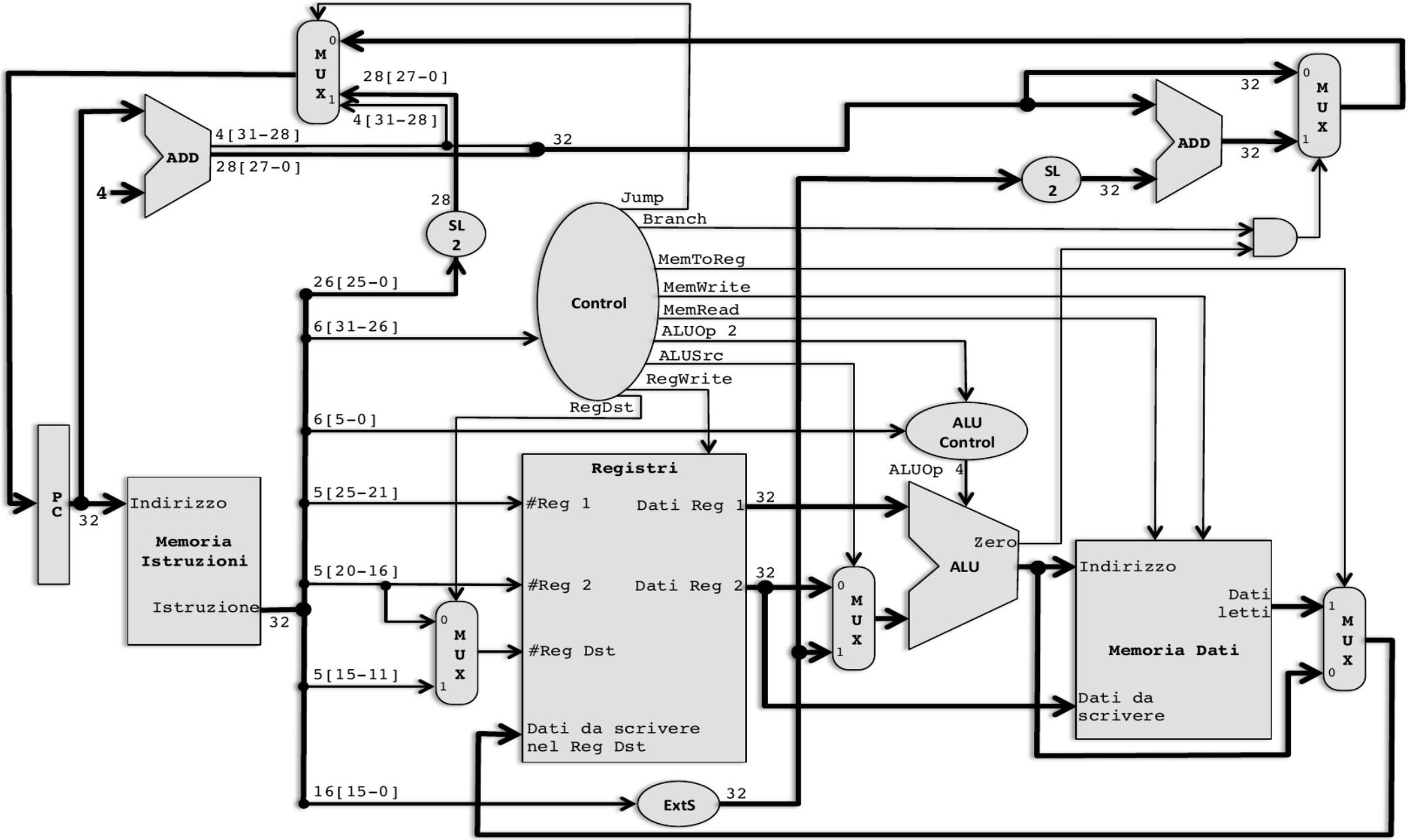
a) disegnate *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU, unità funzionali e sommatori che ritenete necessari.

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

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di 33ns, l'accesso alla memoria impiega 100ns, la ALU e gli adder impiegano 200ns e ignorando gli altri ritardi di propagazione dei segnali, calcolate e scrivete *sullo schema di Figura* il tempo di esecuzione della istruzione.

Esercizio 4A. Si scriva *su foglio protocollo nome, cognome e matricola e:*
a) la funzione che riceve come argomenti l'indirizzo iniziale di una matrice di word, le sue dimensioni M ed N ed un valore 0<=k<min(M,N)/2, che calcola il massimo dei valori contenuti nella cornice k-esima (partendo da quella esterna che corrisponde a k=0) della matrice.
b) il main che chiama la funzione e stampa il risultato usando dei valori di esempio allocati staticamente.

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 – MZ – 16-4-18 – Compito B

Cognome e Nome: _____ **Matricola:** _____

Esercizio 1B. Considerate l'architettura MIPS a ciclo singolo in Figura (sul retro). Qual è il valore dei segnali che partono dalla unità di controllo e dei diversi campi della istruzione se l'istruzione in esecuzione è:
`sll $25, $26, 9` ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	**Branch =**	**Memtoreg =**	**RegWrite =**	**MemWrite =**
Jump =	**MemRead =**	**ALUOp =**	**ALUSrc =**	**Func =**
rs =	**rt =**	**rd =**	**Imm. =**	**Shamt =**

Esercizio 2B. Si ha il dubbio che in una partita di 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 **MemToReg**.
NOTA: Si assume che tutti i valori don't care valgano 0.
a) Indicate *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j, di tipo R immediate**) funzioneranno male e in che modo cambia il loro comportamento.

b) scrivete *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 3B. Si vuole aggiungere alla CPU della Figura l'istruzione branch if equal to zero and link (**`beqz_al`**), di tipo **I** e sintassi assembly **`beqz_al`** $rs, *label* che salta alla destinazione indicata da label se il registro $rs vale 0, ed inoltre salva in $ra (registro 31) l'indirizzo della istruzione seguente (solo se il salto viene eseguito).

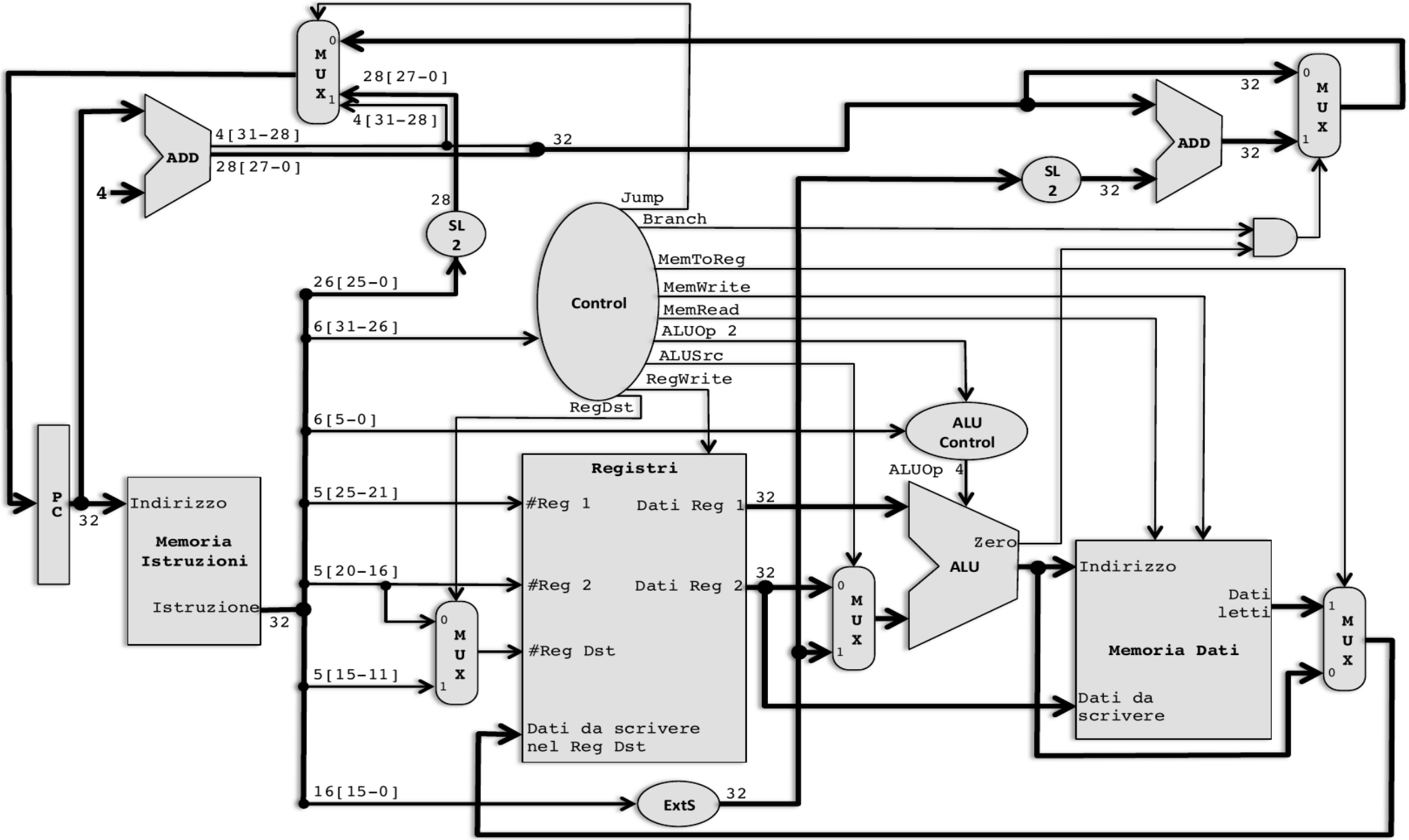
a) disegnate *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU, unità funzionali e sommatori che ritenete necessari.

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

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di 25ns, l'accesso alla memoria impiega 100ns, la ALU e gli adder impiegano 150ns e ignorando gli altri ritardi di propagazione dei segnali, calcolate e scrivete *sullo schema di Figura* il tempo di esecuzione della istruzione.

Esercizio 4B. Si scriva *su foglio protocollo nome, cognome e matricola e:*
a) la funzione assembly che, ricevendo come argomenti l'indirizzo di un **testo** terminato da zero, e di una **chiave** lunga 26 caratteri, modifica in memoria il testo sostituendo solo le lettere minuscole dell'alfabeto con le corrispondenti lettere della chiave (primo carattere della chiave al posto della 'a', secondo al posto della 'b' ….)
b) la procedura **main** che legge con le opportune syscall il testo e la chiave, chiama la funzione e stampa il risultato. **Esempio:** testo = "PER bacca!" chiave="panino…" risultato "PER apnnp!"

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 – MZ – 16-4-18 – Compito C

Cognome e Nome: _____ Matricola: _____

Esercizio 1C. Considerate l'architettura MIPS a ciclo singolo in Figura (sul retro). Qual è il valore dei segnali che partono dalla unità di controllo e dei diversi campi della istruzione se l'istruzione in esecuzione è:
`sw $21, ($23)` ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	**Branch =**	**Memtoreg =**	**RegWrite =**	**MemWrite =**
Jump =	**MemRead =**	**ALUOp =**	**ALUSrc =**	**Func =**
rs =	**rt =**	**rd =**	**Imm. =**	**Shamt =**

Esercizio 2C. 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 **MemToReg** attivo **se e solo se** NON è attivo il segnale di controllo **AluSrc**.
NOTA: Si assume che tutti i valori don't care valgano 0 e che l'uscita della mem=0 se MemRead=0.
a) Indicate *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j, di tipo R immediate**) funzioneranno male e in che modo cambia il loro comportamento.

b) scrivete *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 3C. Si vuole aggiungere alla CPU della Figura l'istruzione ***branch if not zero vectorized*** (`beqz_vect`), di tipo I, e sintassi assembly `beqz_vect` $rt, $rs, index che esegue il salto alla destinazione contenuta nell'elemento di indice **index** del vettore che inizia all'indirizzo $rs se il registro $rt contiene zero.

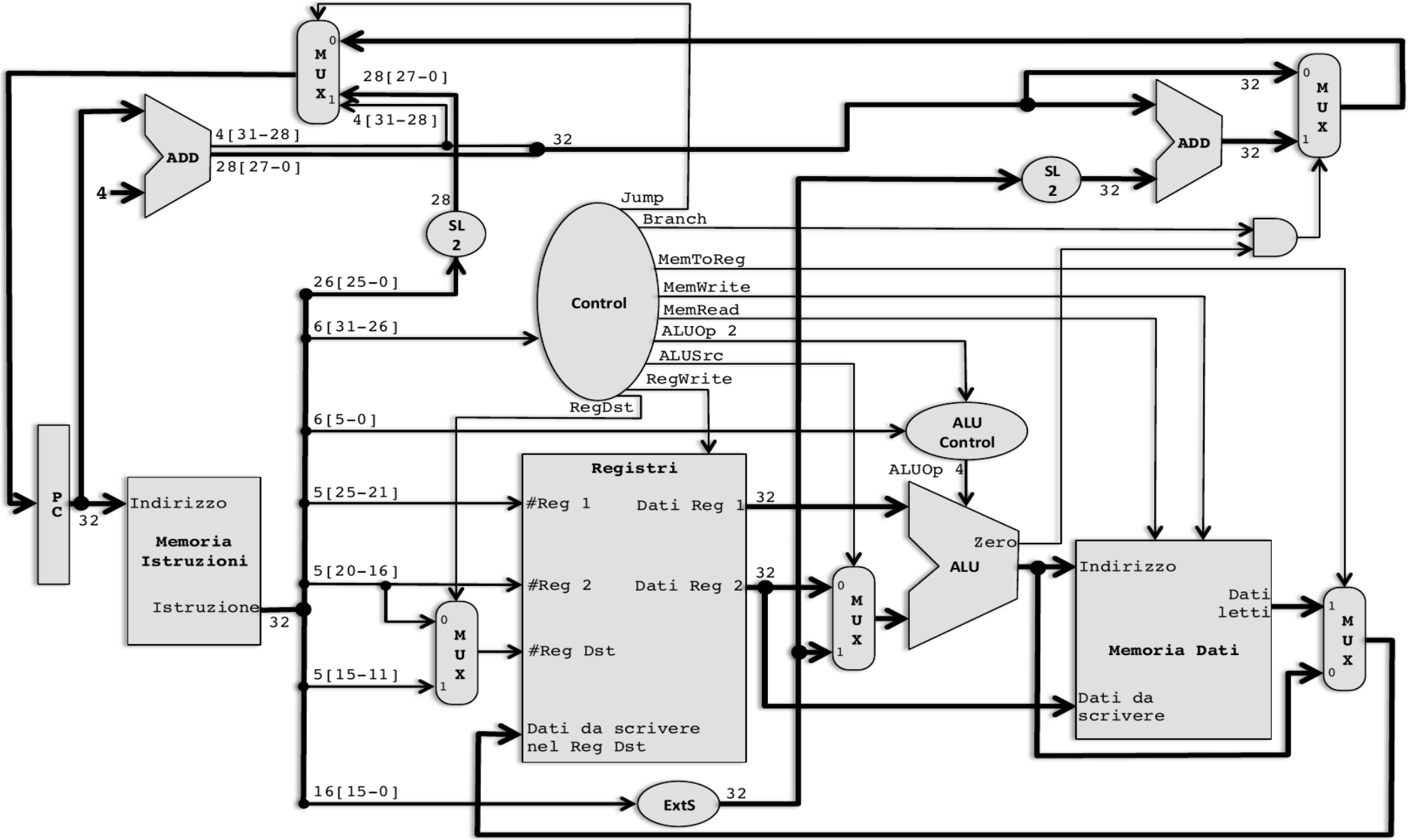
a) disegnate *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU, unità funzionali e sommatori che ritenete necessari.

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

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di 25ns, l'accesso alla memoria impiega 100ns, la ALU e gli adder impiegano 300ns e ignorando gli altri ritardi di propagazione dei segnali, calcolate e scrivete *sullo schema di Figura* il tempo di esecuzione della istruzione.

Esercizio 4C. Si scriva *su foglio protocollo nome, cognome e matricola e:*
a) la funzione assembly che riceve come argomenti l'indirizzo di una matrice di **half-word** e le sue dimensioni **N e M** e che elimina dalla matrice la riga e la colonna che contengono l'elemento massimo spostando opportunamente gli altri elementi per ottenere una matrice **N-1 x M-1**.
b) il **main** che chiama la funzione e stampa il risultato, allocando dei dati di esempio staticamente in memoria.

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



Architettura degli elaboratori – MZ – 16-4-18 – Compito D

Cognome e Nome: _____ **Matricola:** _____

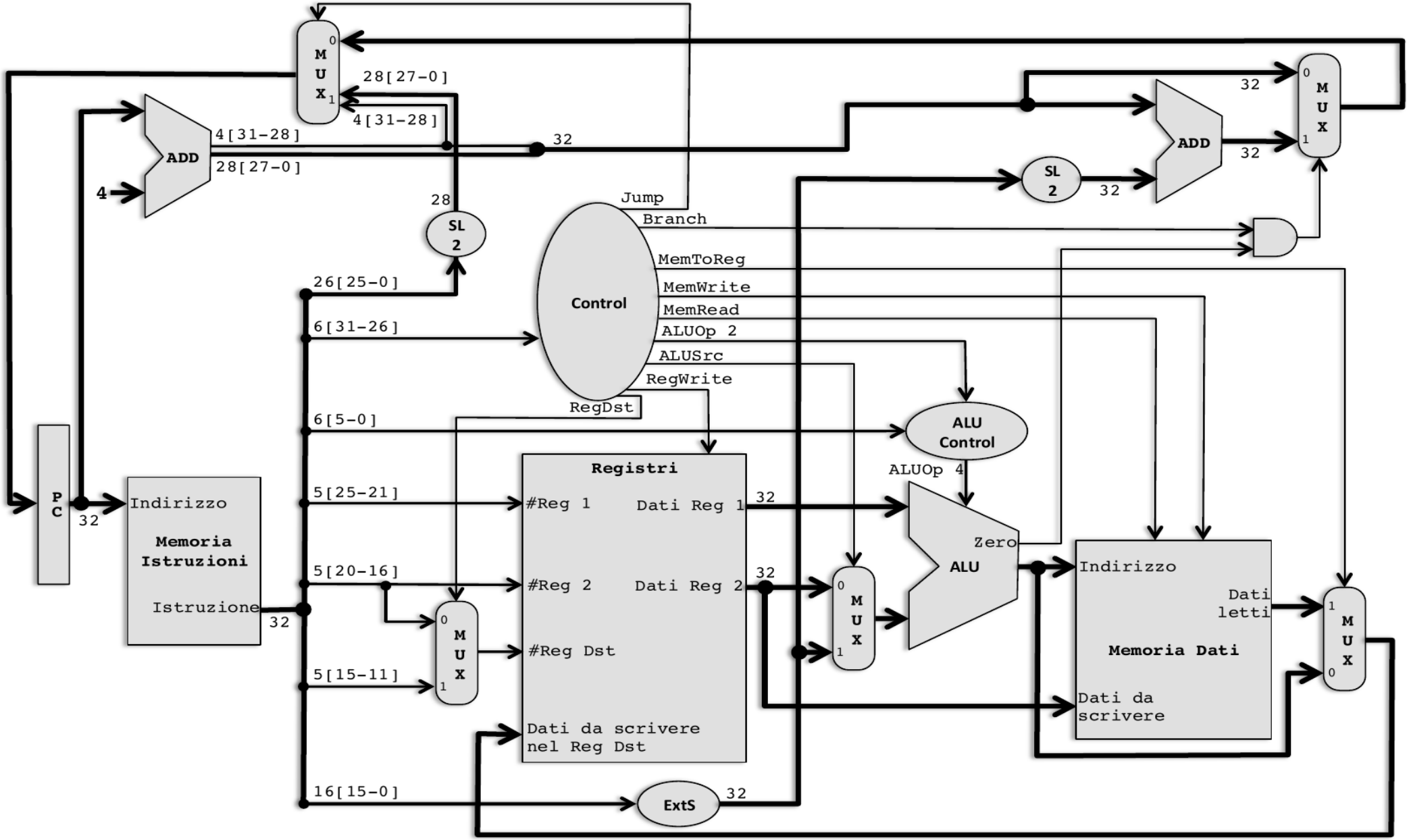
Esercizio 1D. Considerate l'architettura MIPS a ciclo singolo in Figura (sul retro). Qual è il valore dei segnali che partono dalla unità di controllo e dei diversi campi della istruzione se l'istruzione in esecuzione è:
`sub $14, $15, $16` ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	**Branch =**	**Memtoreg =**	**RegWrite =**	**MemWrite =**
Jump =	**MemRead =**	**ALUOp =**	**ALUSrc =**	**Func =**
rs =	**rt =**	**rd =**	**Imm. =**	**Shamt =**

Esercizio 2D. Si ha il dubbio che in una partita di CPU MIPS come quella della Figura la Control Unit sia rotta, producendo il segnale di controllo **Jump** attivo **se e solo se** è attivo il segnale di controllo **Branch**.
NOTA: Si assume che tutti i valori don't care valgano 0.
a) Indicate *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j, di tipo R immediate**) funzioneranno male e in che modo cambia il loro comportamento.

b) scrivete *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 3D. Si vuole aggiungere alla CPU della Figura l'istruzione *add_power_of_2* (`add_p2`), di tipo **R** e sintassi `add_p2` $rd, $rs, *potenza* che mette in $rd la somma di $rs con la potenza di due 2^*potenza*.
a) disegnate *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU, unità funzionali e sommatori che ritenete necessari.
b) indicate *sullo schema* i valori di tutti i segnali di controllo della CU necessari a eseguire l'istruzione.
c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di 33ns, l'accesso alla memoria impiega 133ns, la ALU e gli adder impiegano 150ns e ignorando gli altri ritardi di propagazione dei segnali, calcolate e scrivete *sullo schema di Figura* il tempo di esecuzione della istruzione.
Suggerimento: quale circuito studiato a Progettazione di sistemi digitali potrebbe produrre una potenza di 2?

Esercizio 4D. Si scriva *su foglio protocollo nome, cognome e matricola e:*
a) la funzione assembly che riceve come argomenti: l'indirizzo di una matrice di word quadrata ed il suo lato N **dispari**, ed una stringa contenente solo caratteri maiuscoli 'U'(up) 'D'(down) 'L'(left) o 'R'(right) che indicano una sequenza di spostamenti a partire dalla casella centrale e che calcola la somma di tutte le caselle visitate almeno una volta compresa quella centrale.
b) il main che chiama la funzione e stampa il risultato, allocando dei dati di esempio staticamente in memoria.

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