

Esercizio 4. (10/30)

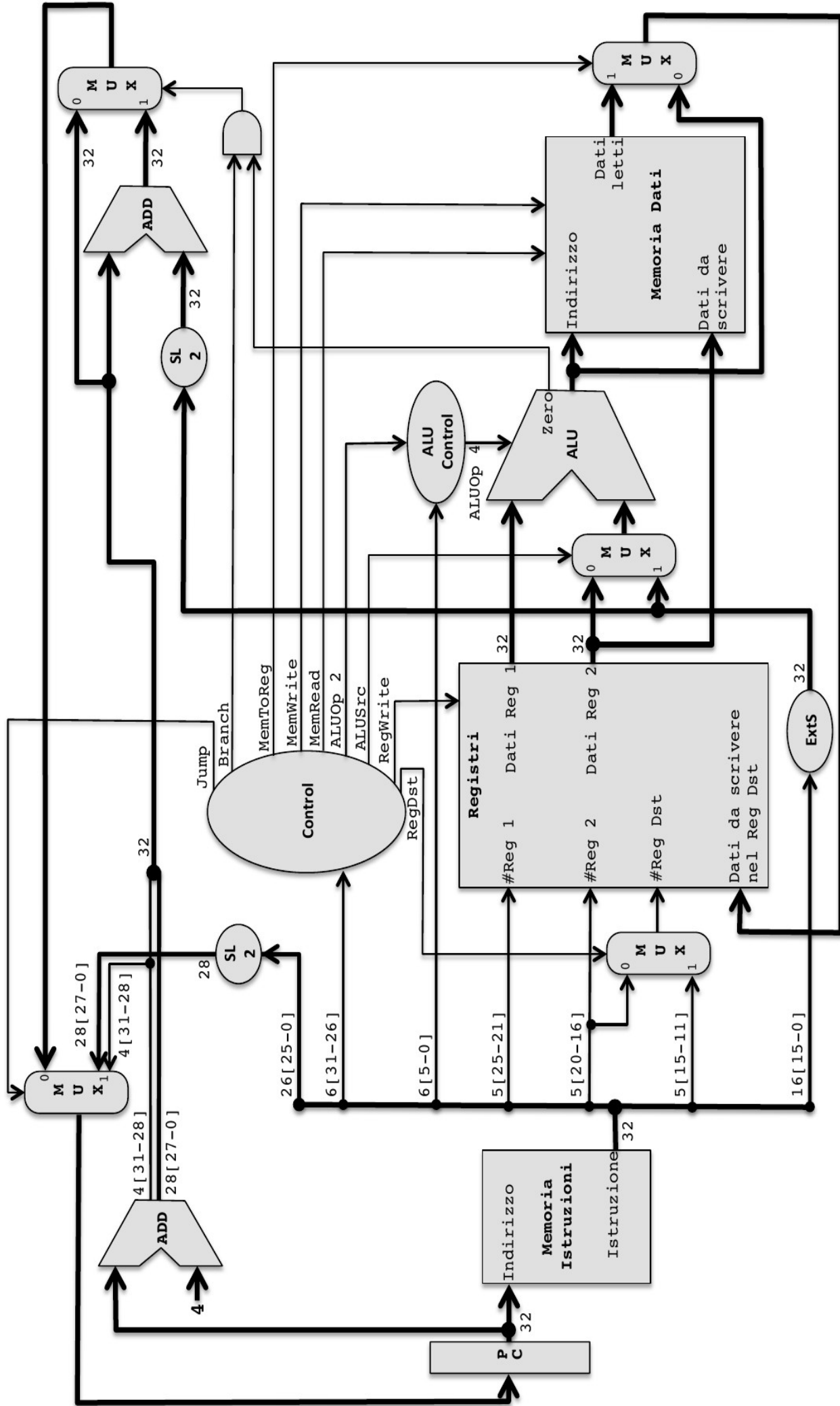
Si vuole aggiungere alla CPU in Figura (sul retro) l'istruzione jump register and link (**jral**), di tipo I e sintassi assembly **jral** \$dest, offset(\$base) che salva nel registro \$dest l'indirizzo della istruzione successiva ed allo stesso tempo salta all'indirizzo contenuto in memoria nella posizione *offset+(\$base)*.

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.

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

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 qui sotto il tempo di esecuzione minimo della istruzione **jral** 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)



Esercizio 3. (10/30)

Si vuole aggiungere alla CPU della Figura (sul retro) l'istruzione vectorized jump (**vj**), di tipo **I** e sintassi assembly **vj** \$indice, *vettore* che salta all'indirizzo contenuto nell'elemento \$indice-esimo del *vettore*.

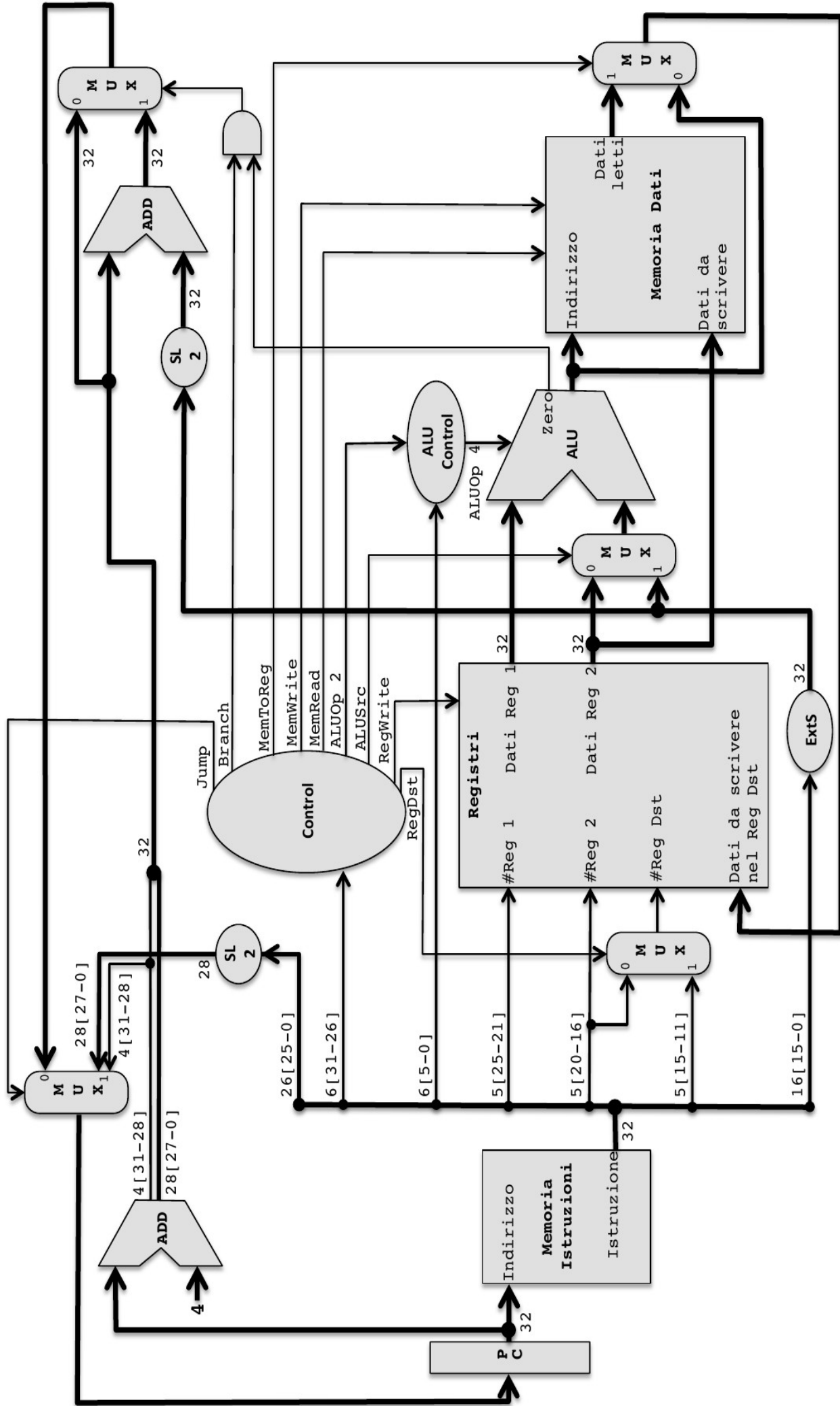
Esempio: se in memoria c'è **vettore: .word 15, 24, 313, 42** e nel registro **\$t0** c'è il valore **3**
allora **vj \$t0, vettore** salterà all'indirizzo **42**

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.

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

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 qui sotto il tempo di esecuzione minimo della istruzione **vj** 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 – 24-4-2014 – Compito A

Cognome e Nome: _____ Matricola: _____

Esercizio 1. (4/30) Nell'architettura MIPS a ciclo singolo in Figura (retro Es. 4). Qual è il valore dei segnali che partono dalla unità di controllo se l'istruzione in esecuzione è **sw** ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =
Jump =	MemRead =	ALUOp =	ALUSrc =	

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 **MemWrite** attivo **se e solo se NON** è attivo il segnale **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 3. (10/30)

1) Scrivete *qui sotto e sul retro del foglio* la procedura assembly che riceve come argomenti l'indirizzo di una matrice quadrata di **half-word** e la sua dimensione **N**, e che modifica la matrice scambiando tutti gli elementi a coordinate X,Y con quelli a coordinate Y,X solo se entrambi gli elementi della coppia sono pari.

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

Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 24-4-2014 – Compito B

Cognome e Nome: _____ Matricola: _____

Esercizio 1. (4/30) Nell'architettura MIPS a ciclo singolo in Figura (retro Es. 4). Qual è il valore dei segnali che partono dalla unità di controllo se l'istruzione in esecuzione è **la**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =
Jump =	MemRead =	ALUOp =	ALUSrc =	

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 NON** 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 3. (10/30)

1) Scrivete *qui sotto e sul retro del foglio* la funzione assembly che riceve come argomento l'indirizzo di un testo terminato da zero formato da parole separate da uno spazio, e che rovescia tutte le parole dispari trovate nel testo (la prima, la terza, la quinta ...). (se preferite potete suddividere il programma in più funzioni)

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

Esempio: “soPra la pancA la Capra camPa” diventa “arPos la Acnap la arpaC camPa”