

Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 19-4-2013 – Compito A

Cognome e Nome: _____ Matricola: _____

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

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

Esercizio 2. 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 **MemWrite** attivo **se e solo se** è attivo il segnale di controllo **Branch**.

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. Si vuole aggiungere alla CPU della Figura l'istruzione load address (**la**), di tipo I, e sintassi assembly **la \$dest, offset(\$base)** che carica nel registro \$dest l'indirizzo calcolato sommando offset (esteso a 32 bit) al contenuto del registro \$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 che ritenete necessari.

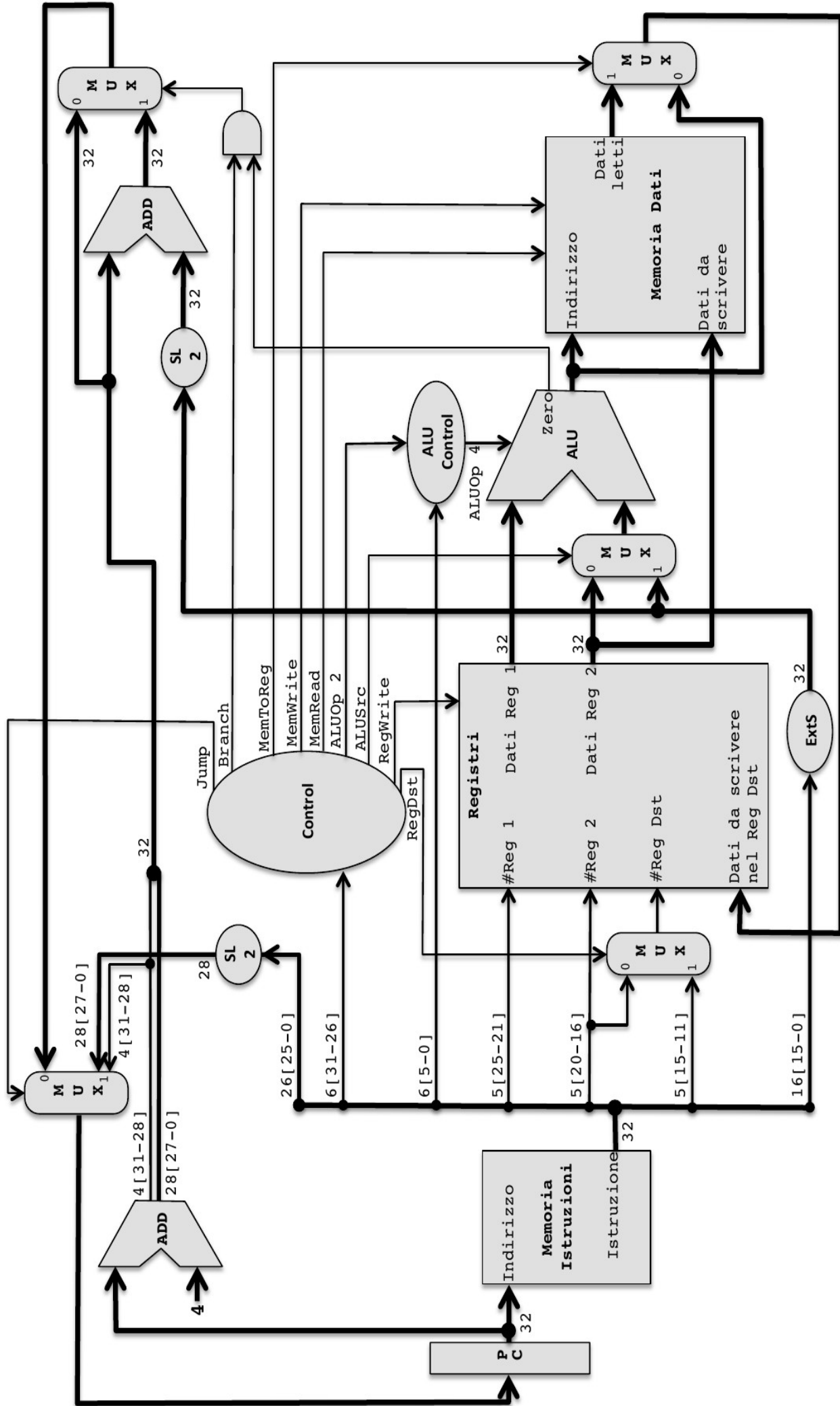
b) si indichino *sullo schema della Figura* i valori dei segnali di controllo necessari all'istruzione **la**.

c) tenendo conto che il tempo di accesso ai registri (sia in lettura che scrittura) è di 33ps, l'accesso alla memoria impiega 100ps, la ALU e gli adder impiegano 133ps e ignorando gli altri ritardi di propagazione dei segnali, calcolate e scrivete *sullo schema di Figura* il tempo di esecuzione minimo della istruzione **la**.

Esercizio 4. Si scriva *sui fogli protocollo* la procedura assembly che riceve come argomenti: l'indirizzo di un vettore di **half-word**, la sua dimensione **N** ed un valore intero positivo **K**, e calcola e stampa nell'ordine i primi **K** valori minimi del vettore. Scrivete poi il main che chiama la procedura, considerando che i dati sono tutti allocati staticamente in memoria.

Esempio: se il vettore contiene **1, 55, 19, 3, 92, 18, 62** **N=7** **K=3**
la funzione deve stampare i valori **1 3 18**

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 – 19-4-2013 – Compito B

Cognome e Nome: _____ Matricola: _____

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

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

Esercizio 2. 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**.

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. Si vuole aggiungere alla CPU della Figura l'istruzione increment immediate (**inci**), di tipo **I** e sintassi assembly **inci**.\$dest, valore immediato che incrementa il registro \$dest del valore immediato esteso a 32 bit.

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 che ritenete necessari.

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

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 i sommatore impiegano **200ps** e ignorando gli altri ritardi, calcolate e scrivete *sullo schema della Figura* il tempo di esecuzione minimo della istruzione **inci**.

Esercizio 4. Si scriva *sui fogli protocollo* la funzione assembly che, ricevendo come argomento l'indirizzo di un testo terminato da zero, calcola se il testo è palindromo (cioè se è identico a sé stesso se letto in senso contrario) **IGNORANDO GLI SPAZI** e torna 1 o 0 nei due casi. Definite quindi il main che usa la funzione realizzata per testare una stringa allocata staticamente in memoria e stampa la stringa "PALINDROMO" oppure "NON PALINDROMO" a seconda del risultato del test.

Esempio: "amo Roma" è palindromo (ignorando lo spazio tra le due parole)

NOTA: verrà valutata con un punteggio maggiore l'implementazione ricorsiva del test.

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

