

Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 11-4-16 – 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 `sll $12, $11, 7` ? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =	
Jump =	MemRead =	ALUOp =	ALUSrc =		
rs =	rt =	rd =	imm. =	shamt =	func =

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 **ALUSrc** attivo se e solo se è attivo il segnale **Branch**.

NOTA: assumete che tutti i don't care=0.

a) Si indichino *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j**) funzioneranno male, e qual'è il nuovo comportamento.

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 **rovesciaEScambia** che riceve come argomenti l'indirizzo di una stringa terminata da zero ('\0') di massima lunghezza 100 e quello di una seconda area di memoria di 100 caratteri in cui mettere il risultato. La funzione deve costruire la seconda stringa terminata da zero, contenente il testo iniziale rovesciato e con i caratteri minuscoli trasformati in maiuscolo e viceversa.

Esempio: "speriamo CHE me LA cavo" → "OVAC al EM ehc OMAIREPS"

2) Scrivete un esempio di **main** che legge la stringa da input con la opportuna syscall, usa la funzione sulla stringa letta, e stampa la stringa risultante.

Esercizio 4. (10/30)

Si vuole aggiungere alla CPU l'istruzione *test and set* (**tset**), di tipo R e sintassi assembly

`tset $rs, $rt, $rd` che setta $\$rt = \rs ed intanto mette il vecchio valore di $\$rt$ in $\$rd$.

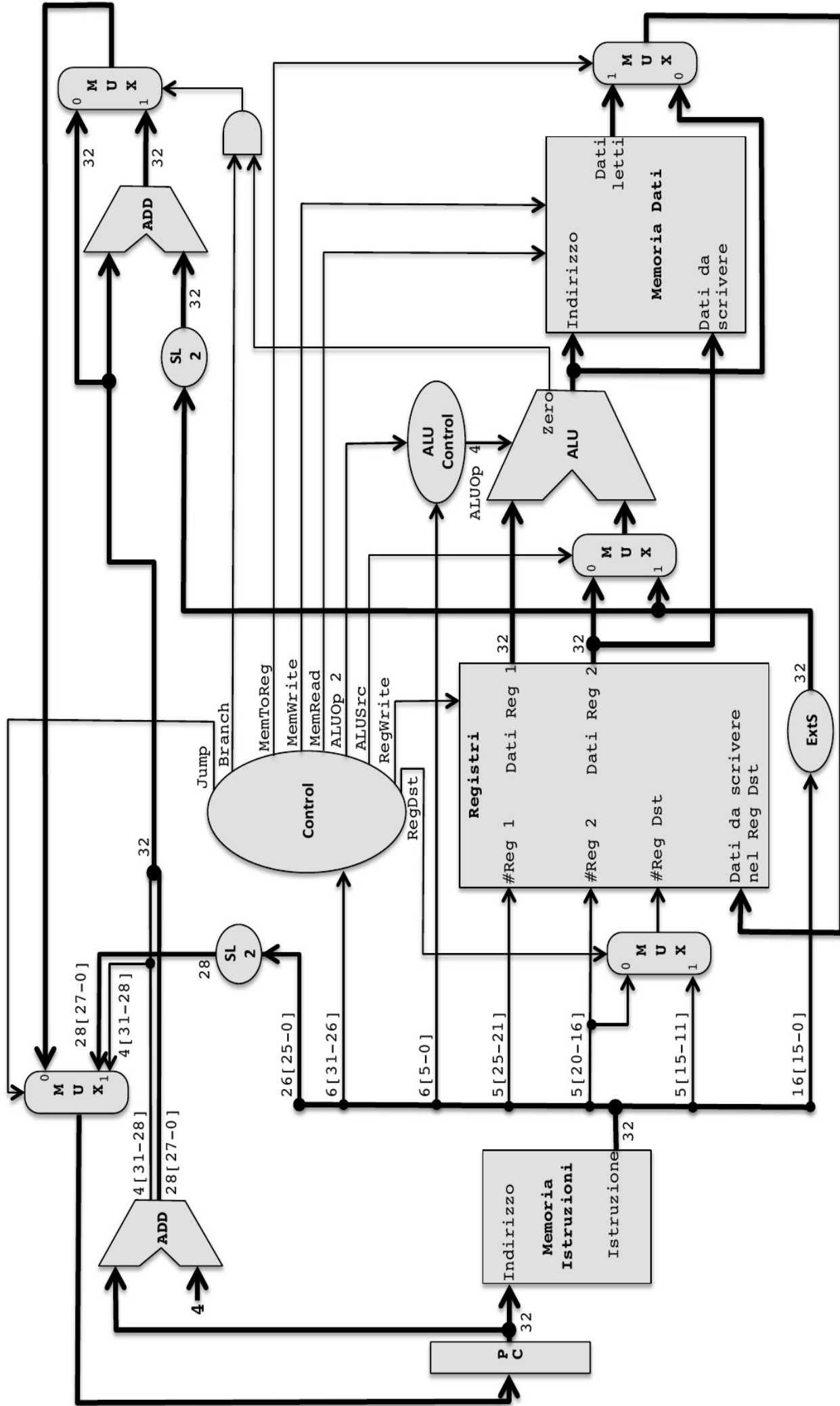
a) si disegnano *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU e sommatore (ecc) che ritenete necessari.

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

b) 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 – 11-4-16 – 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 **lw \$20, 333**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =	
Jump =	MemRead =	ALUOp =	ALUSrc =		
rs =	rt =	rd =	imm. =	shamt =	func =

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 **Jump** attivo **se e solo se** è attivo il segnale di controllo **MemToReg**.

NOTA: assumete che tutti i don't care=0.

a) Si indichino *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j**) funzioneranno male, e qual'è il nuovo comportamento.

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 **antisimmetricaP** che riceve come argomento l'indirizzo di una matrice di **halfword** quadrata **M** il suo lato $N < 100$, e verifica se **M** è antisimmetrica (ovvero se $M[x][y] == -M[y][x]$ per ogni coppia di indici x, y).

2) Definite staticamente un esempio di matrice con dati non nulli, e scrivete il **main** che usa la funzione sulla matrice, e stampa "M è antisimmetrica" oppure "M NON è antisimmetrica" a seconda del risultato.

Esercizio 4. (10/30)

Si vuole aggiungere alla CPU l'istruzione **swap**, (*scambia*) di tipo **R** e sintassi assembly

swap \$rs, \$rt che scambia tra di loro i valori dei due registri **\$rs** ed **\$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 **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.

Prova intermedia di Architettura degli elaboratori canale MZ – Sterbini – 11-4-16 – 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 **srav \$10, \$11, \$12**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =	
Jump =	MemRead =	ALUOp =	ALUSrc =		
rs =	rt =	rd =	imm. =	shamt =	func =

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 **ALUSrc attivo se e solo se è attivo il segnale MemWrite**.

NOTA: assumete che tutti i don't care=0.

a) Si indichino *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j**) funzioneranno male, e qual'è il nuovo comportamento.

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 terminato da zero lungo al massimo 200 caratteri, e che conta quante sono le parole che hanno solo la lettera iniziale maiuscola nel testo. Una parola inizia per maiuscola ed ha almeno 2 caratteri e può essere attaccata ad un'altra parola.

Esempio: "TreTigri controTrentatrèTrentini entrarono a TN" → 4 (sottolineate nel testo)

2) Scrivete il **main** che legge il testo da input con la opportuna syscall, usa la funzione e stampa il risultato.

Esercizio 4. (10/30)

Si vuole aggiungere alla CPU l'istruzione *compare and set immediate (casi)*, di tipo **I** e sintassi assembly **casi \$rs, \$rt, newvalue** che setta il registro **\$rt=newvalue** se e solo se **\$rs==\$rt**.

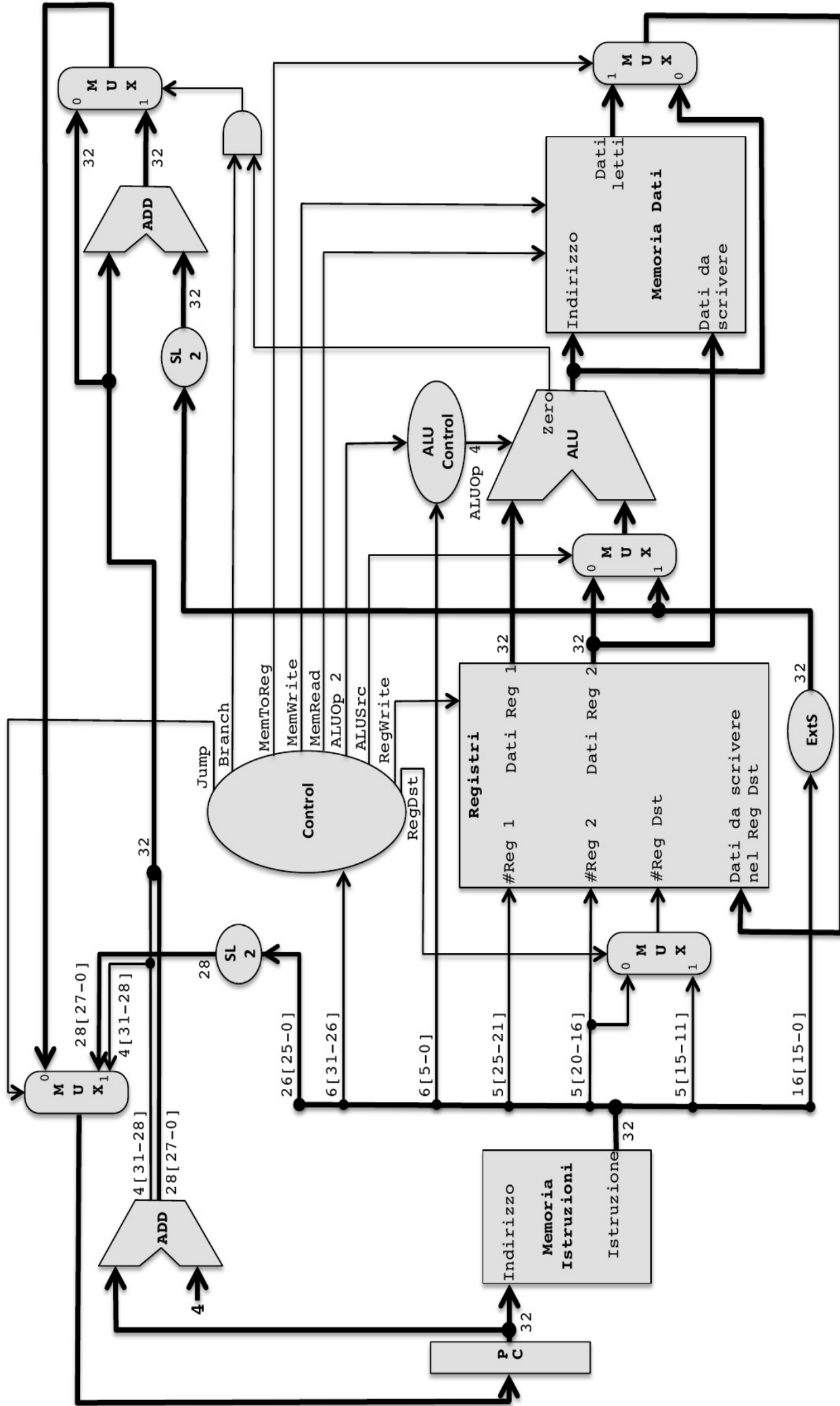
a) si disegnano *sullo schema della Figura* le modifiche necessarie a realizzare la funzione, aggiungendo tutti gli eventuali MUX, segnali di controllo, bus, ALU e sommatore (ecc) che ritenete necessari.

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 – 11-4-16 – 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 **bnez \$23, 444**? (*scrivete qui sotto, NON sulla figura*)

RegDst =	Branch =	MemtoReg =	RegWrite =	MemWrite =	
Jump =	MemRead =	ALUOp =	ALUSrc =		
rs =	rt =	rd =	imm. =	shamt =	func =

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 **Jump attivo se e solo se è attivo il segnale di controllo MemWrite**.

NOTA: assumete che tutti i don't care=0.

a) Si indichino *qui sotto* quali delle istruzioni base (**lw, sw, di tipo R, beq, j**) funzioneranno male, e qual'è il nuovo comportamento.

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 argomenti l'indirizzo di una matrice di word **M** e le sue dimensioni $2 < X < 20$ e $2 < Y < 20$ e che calcola la differenza **C1-C2** tra **C1=massimo degli elementi sul bordo esterno** ed **C2=minimo degli elementi sulla seconda cornice** (quella di lati X-2 e Y-2).

2) Definite staticamente un esempio con dati non nulli e il **main** che usa la funzione e stampa il risultato.

Esercizio 4. (10/30)

Si vuole aggiungere alla CPU l'istruzione **b_case**, (*branch if case*) di tipo **I** e sintassi assembly **b_case \$rs, \$rt, valore** che salta all'indirizzo contenuto in \$rt se \$rs==valore.

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)

