

Versione sia word che pdf

APPUNTI PER **ESERCIZI PIPELINE**

CODICE ASSEMBLY **R o \$ indicano entrambi registro**

- nei processori della famiglia MIPS \$0 contiene sempre zero
- processori MIPS hanno due cache separate (istruzioni/dati) per evitare conflitti tra IF e MEM.
- nel processore MIPS l'unica dipendenza che si rileva è RAW
- eventuale dipendenza Read after Write non sussiste in caso di coincidenza tra le fasi WB (istruzione scrittura) e fase ID (istruzione lettura) in quanto scrittura avviene nella prima metà del ciclo di clock, lettura nella seconda (in esercizio comunque indicarla con freccia verticale)

POTREBBERO ESSERCI DELLE IMPRECISIONI/ERRORI correggete pure

ADD R2, R1, R3 addizione $R2 = R1 + R3$

risultato pronto alla fase EX, scritto in WB

ADDI \$7, \$3, 8

SUB #1, R2 sottrazione $R2 = R2 - 1$

SUBI \$2, \$3, 4

la I in più sta per il formato immediato dell'istruzione (e non formato R) quindi uno dei valori da sommare o sottrarre è una costante

MUL moltiplicazione

MOVE #10, R1 assegna valore 10 al registro R1 $R1 = 10$

LW R1, 0(R2) load word carica nel registro uno ciò che è contenuto in memoria nell'indirizzo calcolato a partire dal registro 2 più spiazzamento.

$R1 = \text{mem}[0 + [R2]]$

risultato pronto alla fase MEM (non dalla ALU ma da LMD), scritto in WB

SW R1, 0(R2) store word (carica in memoria il contenuto del registro) calcola l'indirizzo a partire dal R2 a cui somma lo spiazzamento e lì scrive in memoria il contenuto del registro uno $\text{mem}[0 + [R2]] = [R1]$

risultato pronto alla fase MEM, scritto in MEM

BEQ \$1, \$2, 16 salto condizionato if($\$1 == \2) $PC = PC + 16$

condizione pronta alla fase EX e anche si calcola l'indirizzo dell'eventuale salto, verifica condizione e assegnazione del target in fase MEM

istruzione mips j salto incondizionato (non so la sua eventuale scrittura assembly) j 45054 $[PC] = 45054$ (in base 10)

?? BEZ TAR branch if zero

?? BENZ R4 _____ condizione: contenuto del registro diverso da zero