

Esercizio-03-DLX sequenziale tipo esame

Si supponga di voler estendere il Set di Istruzioni del DLX con la seguente istruzione:

LOOP Rx, IMMEDIATE

dove Rx è un registro generale ed IMMEDIATE è un immediato con segno a 16 bit.

L'istruzione deve decrementare il registro Rx e quindi trasferire il controllo all'istruzione specificata mediante IMMEDIATE solo se il valore di Rx è diverso da 0. Una possibile descrizione delle operazioni svolte dalla nuova istruzione è data da:

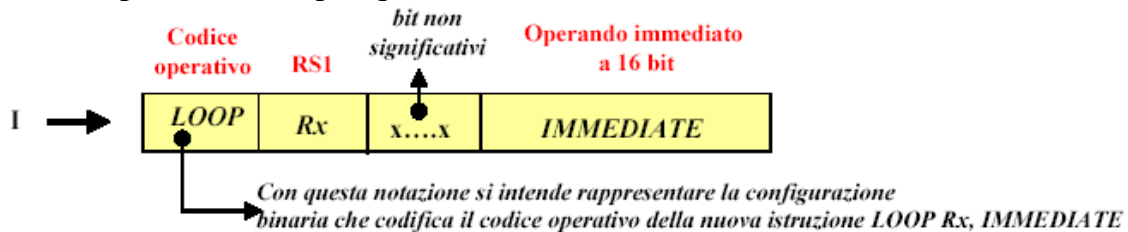
$Rx \leftarrow Rx - 1;$

if ($Rx \neq 0$) then $PC \leftarrow PC + IMMEDIATE;$

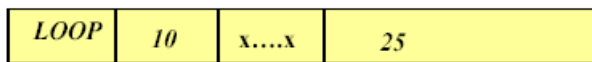
- Quale fra i 3 formati delle istruzioni DLX risulta il più idoneo alla codifica di questa nuova istruzione? Come potrebbe essere codificata l'istruzione LOOP R10, 25?
- Con riferimento al datapath del DLX sequenziale (cioè non-pipelined) visto a lezione, si sviluppi il diagramma degli stati che controlla l'esecuzione della nuova istruzione, inserendo anche gli stati necessari alle fasi di fetch e decodifica.
- A partire dal diagramma degli stati ottenuto e ipotizzando che ogni accesso alla memoria richieda 4 clock, si calcoli il CPI (clock-per-istruzione) della nuova istruzione.
- Si mostri come potrebbe essere utilizzata la nuova istruzione per realizzare in assembler DLX un loop che somma gli elementi corrispondenti di 2 vettori, V1 e V2, e memorizza il risultato in un terzo vettore, V3. Si assuma che, per tutti e 3 i vettori, l'indirizzo del primo elemento sia inferiore a 64K, che la dimensione dei 3 vettori sia 1K e che gli elementi siano delle word (32 bit).

Exe -03- su DLX sequenziale | 1

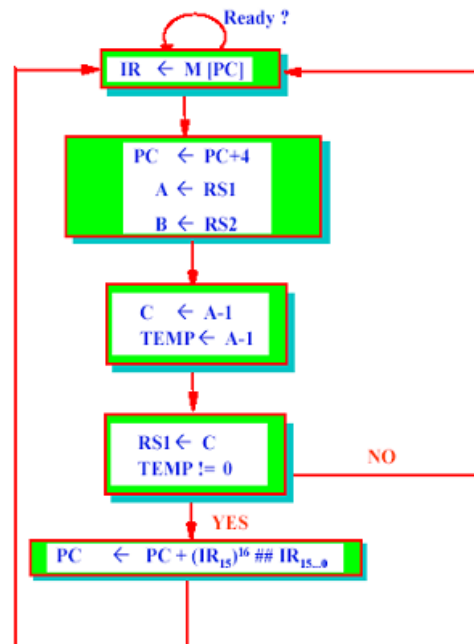
a) Il formato più idoneo è il formato I, che consente di usare uno dei campi registro per la codifica di "Rx" ed il campo immediato per quella di "IMMEDIATE":



Impiegando questo formato l'istruzione LOOP R10, 25 può essere codificata come segue:



b) Diagramma degli stati:



c) E' necessario distinguere fra 2 casi: l'istruzione causa un trasferimento di controllo ("branch taken") e l'istruzione non causa un trasferimento di controllo ("branch not-taken").

Nel primo caso CPI= 8 (5 stati + 3 clock aggiuntivi dovuti all'accesso in memoria), nel secondo caso CPI= 7 (4 stati +

3 clock aggiuntivi dovuti all'accesso in memoria).

Esercizio-03-DLX sequenziale tipo esame

Si supponga di voler estendere il Set di Istruzioni del DLX con la seguente istruzione:

LOOP Rx, IMMEDIATE

dove Rx è un registro generale ed IMMEDIATE è un immediato con segno a 16 bit.

L'istruzione deve decrementare il registro Rx e quindi trasferire il controllo all'istruzione specificata mediante IMMEDIATE solo se il valore di Rx è diverso da 0. Una possibile descrizione delle operazioni svolte dalla nuova istruzione è data da:

$Rx \leftarrow Rx - 1;$

if ($Rx \neq 0$) then $PC \leftarrow PC + IMMEDIATE;$

- Quale fra i 3 formati delle istruzioni DLX risulta il più idoneo alla codifica di questa nuova istruzione? Come potrebbe essere codificata l'istruzione LOOP R10, 25?
- Con riferimento al datapath del DLX sequenziale (cioè non-pipelined) visto a lezione, si sviluppi il diagramma degli stati che controlla l'esecuzione della nuova istruzione, inserendo anche gli stati necessari alle fasi di fetch e decodifica.
- A partire dal diagramma degli stati ottenuto e ipotizzando che ogni accesso alla memoria richieda 4 clock, si calcoli il CPI (clock-per-istruzione) della nuova istruzione.
- Si mostri come potrebbe essere utilizzata la nuova istruzione per realizzare in assembler DLX un loop che somma gli elementi corrispondenti di 2 vettori, V1 e V2, e memorizza il risultato in un terzo vettore, V3. Si assuma che, per tutti e 3 i vettori, l'indirizzo del primo elemento sia inferiore a 64K, che la dimensione dei 3 vettori sia 1K e che gli elementi siano delle word (32 bit).

Exe -03- su DLX sequenziale 2

d) Assembler DLX che impiega la nuova istruzione LOOP Rx, IMMEDIATE

ADDI R1, R0, 1024 ; *inizializzazione del contatore delle iterazioni*

ADD R2, R0, R0 ; *inizializzazione del registro impiegato come indice per l'accesso ai vettori*

ciclo: LW R3, V1(R2) ; *trasferimento di un dato dal vettore V1 in R3*

LW R4, V2(R2) ; *trasferimento di un dato dal vettore V2 in R4*

ADD R4, R3, R4 ; *somma dei dati prelevati dai due vettori*

SW V3(R2), R4 ; *memorizzazione del risultato nel vettore R3*

ADDI R2, R2, 4 ; *incremento dell'indice*

**LOOP R1, ciclo ; *decremento del contatore delle iterazioni e salto alla label "ciclo" se
; il contatore delle iterazioni è diverso da 0***

Si osservi che la nuova istruzione che questo esercizio propone di aggiungere all'ISA del DLX è di tipo "CISC". Difatti, mediante tale istruzione il programmatore può eseguire due operazioni elementari: decremento del contatore delle iterazioni e "branch" dipendente dal valore del contatore.

Nell'ISA del DLX, che è di tipo RISC, questa istruzione non esiste. Se avessimo voluto scrivere il codice precedente usando solo le istruzioni realmente presenti nell'ISA DLX, al posto dell'ultima istruzione avremmo dovuto inserire la seguente coppia di istruzioni:

SUB R1, R1, 1 ; *decremento del contatore delle iterazioni*

BNEZ R1, ciclo ; *salto alla label "ciclo" se il contatore delle iterazioni è diverso da 0*