

ASSEMBLY LC-2

Istruzioni Operative (Operazioni logiche)

AND DR,SR1,SR2 – Scrive in DR il prodotto logico bit a bit dei valori contenuti in SR1 e SR2
AND DR,SR1,Imm5 – Scrive in DR il prodotto logico bit a bit di SR1 e 0000 0000 000 imm5
ADD DR,SR1,SR2 – Scrive in DR la somma logica di SR1 e SR2
ADD DR,SR1,Imm5 – Scrive in DR la somma logica di SR1 e 0000 0000 000 imm5
NOT BR,DR – Scrive in DR la negazione logica di BR

Istruzioni di trasferimento (Save e load da celle a cpu e viceversa)

LEA DR,label – Carica in DR l'indirizzo rappresentato dalla label, rimanendo nella stessa “pagina”
LD DR,label – Carica in DR il contenuto della cella che si trova all'indirizzo rappresentato dalla label
LDI DR,label – Carica in DR il contenuto della cella che si trova all'indirizzo contenuto nella label
LDR DR,BR,index6 – Carica in DR il contenuto della cella all'indirizzo BR+0000 0000 00 index6
ST SR,label – Salva nell'indirizzo rappresentato dalla label il contenuto di SR
STI SR, label – Salva nell'indirizzo rappresentato dalla label il contenuto della cella il quale indirizzo è scritto in SR (simile a LDI, serve per scrivere in tutte le celle anche fuori dalla pagina)
STR SR,BR,index6 – Salva nella cella BR + 0000 0000 00 index6 il contenuto di SR

Istruzioni di controllo (Salti a istruzioni o sottoprogrammi)

BR [NZP] label – Esegue un salto all'indirizzo della cella rappresentata da label se si verifica la condizione rappresentata dallo scrivere N e/o Z e/o P (si possono scrivere da una a tutte e 3 le lettere). Se l'ultimo valore scritto in un registro è negativo si verifica N, P se positivo, Z se nullo.
JSR label – Scrive il PC in R7 ed esegue un salto incondizionato all'indirizzo rappresentato da label
JSRR BR,index6 – Scrive il PC in R7 ed esegue il salto all'indirizzo BR + 0000 0000 00 index6 (per andare ad ogni cella anche non nella pagina corrente come invece fa JSR)
RET – La prossima istruzione eseguita sarà quella contenuta nella cella con indirizzo scritto in R7
RTI – Ritorno da un interrupt
TRAP vect8 – Scrive PC in R7 e la prossima istruzione eseguita sarà quella contenuta in vect8
TRAP x21 – Scrive su console il carattere ASCII rappresentato da R0
TRAP x23 – Scrive in R0 il valore del carattere ASCII preso in input da console
TRAP x25 – Arresta il programma

Pseudo istruzioni

.orig – inizio del programma, non vuole label davanti
.end – fine del programma, non vuole label davanti
(label) .blkw N – riserva un array di N parole che potranno esser usate dal programma (variabili)
(label) .fill X – scrive nella cella la costante X (non ci si potrà quindi scrivere dopo)

Registri della CPU

Da R0 a R7, GPR cioè i registri per le normali operazioni
CC, Condition Code cioè il segno dell'ultimo numero scritto in un registro (N, Z o P)
PC, Program Counter cioè l'indirizzo dell'ultima operazione eseguita (per saper quale eseguire dopo)

Simbologia e Note

Le cifre possono essere scritte in bit, in esadecimale preceduti da una x, o in decimale preceduti da #
Ogni riga può essere preceduta da una label, un'etichetta a cui si ci potrà riferire con istruzioni del programma al posto dell'indirizzo delle celle (ovviamente non potrà esser duplicata)
DR, SR e BR = Registro di destinazione, “sorgente” e base
Imm5 = valore da -16 a +15 oppure 5bit da usare nell'AND e nell'ADD
index6 = valore da -32 a +31 oppure 6bit da sommare ad un BR
vect8 = 8bit per andare ad un sottoprogramma nelle prime 256celle