

Assembly

Na msp340 nas instruções existem o operador uma fonte e um destino, sempre nessa ordem.

Ex: mov.w R5,R6

Lista de instruções

Mov.w R5,R6 ----- move R5 para R6

Sub.w R5,R6 ----- subtrai R6 de R5

Add.w R5,R6 ----- adiciona R5 a R6

Dec.w R5 ----- R5-1/R5--

Decd.w R5 -----R5-2

Inc.w R5 -----R5+1/R5++

Incd.w R5 -----R5+2

Rla.w R5 -----Desloca um bit para a esquerda(left), deslocar n vezes para a esquerda corresponde a multiplicar por 2^n

Rra.w R5 -----Desloca de um bit para a direita (right), deslocar n vezes para a esquerda corresponde a dividir por

Bis.w R4,R5 ----- Or entre R4 e R5 e ele guarda em R5

And.w R4,R5 -----And entre R4 e R5 e ele guarda em R5

Bic.w R4,R5 -----Operação And com R5 e o **INVERSO** de R4, guarda em R5

Xor.w R4,R5-----Xor entre R4,R5 , guarda em R5

Inv.w R5 ----- Inverte todos os bits de R5

Bis.b R4,R5 ----- mesmo com o .w só que com bytes(8bits)

And.b R4,R5 ----- mesmo com o .w só que com bytes(8bits)

Bic.b R4,R5 ----- mesmo com o .w só que com bytes(8bits)

Xor.b R4,R5----- mesmo com o .w só que com bytes(8bits)

Inv.b R5 ----- mesmo com o .w só que com bytes(8bits)

Condicionais:

1. Cmp R4,R5 ----- Compara R5 com R4 e seta ou não os bits N,Z,C e V
 - 1.1. N= 1 se $R5 < R4$. N=0 caso contrário
 - 1.2. Z=1 se $R5 = R4$. Z=0 caso contrário
 - 1.3. C=1 se houver carry
 - 1.4. V=1 se houver overflow
2. Tst R11 ----- Pode ser usado pra testar se R11 é menor ou igual a zero.
 - 2.1. N=1 se $R11 < 0$. N=0 caso contrário
 - 2.2. Z=1 se $R11 = 0$. Z=0 caso contrário
 - 2.3. C=1
 - 2.4. V=0

3. Cmp R5,R4
Jeq LABEL_1

Se R4 for **igual** a R5 o “jump” do jeq é executado.

4. Cmp R5,R4
Jne LABEL_2

Se R4 for **diferente** de R5 o “jump” do jne é executado.

5. Jmp LABEL_3
Executa o “jump” automaticamente sem condições