

--- Operaciones aritméticas

ADD: ADDition

Saca de la pila un número entero y le suma el acumulador A. El resultado queda en A.
Código: 10
Banderas: ZNCO se alteran de acuerdo a la suma.

SUB: SUBstraction

Saca de la pila un número entero y le resta el acumulador A. El resultado queda en A.
Código: 11
Banderas: ZNCO se alteran de acuerdo a la resta

MUL: MULTiplication

Saca de la pila un número entero y lo multiplica por acumulador A. El resultado queda en A.
Código: 12
Banderas: ZN se alteran de acuerdo a la multiplicación

DIV: DIVision

Saca de la pila un número entero y lo divide por el acumulador. El cociente queda en A.
Código: 13
Banderas: ZN se alteran de acuerdo al cociente.
0 queda en 1 si el divisor es cero.
queda en 0 si el divisor no es cero.

MOD: MODulus

Saca de la pila un número entero y lo divide por el acumulador. El módulo o residuo queda en A.
Código: 14
Banderas: ZN se alteran de acuerdo al residuo
0 queda en 1 si el divisor es cero.
queda en 0 si el divisor no es cero.

CMP: CoMPare

Saca de la pila un número entero y le resta el acumulador A. El resultado se pierde.
Código: 15
Banderas: ZNCO se alteran de acuerdo a la resta

SHL: SHift Left

Saca de la pila un entero y lo corre a la izquierda tantas veces como diga el acumulador A.
El resultado queda en el acumulador A.
Código: 16
Banderas: ZN se alteran de acuerdo al resultado

SHR: SHift Right

Saca de la pila un entero y lo corre a la izquierda tantas veces como diga el acumulador A.
El resultado queda en el acumulador A.
Código: 17
Banderas: ZN se alteran de acuerdo al resultado

NEG: NEGative

Calcula el inverso aditivo del acumulador A.

El resultado queda en el acumulador A.
Código: 18
Banderas: ZN se alteran de acuerdo al resultado.

INC: INCrement
Suma 1 al acumulador A.
El resultado queda en el acumulador A.
Código: 19
Banderas: COZN se alteran de acuerdo al resultado.

DEC: DECrement
Resta 1 al acumulador A.
El resultado queda en el acumulador A.
Código: 1a
Banderas: COZN se alteran de acuerdo al resultado.

XCH: eXCHange
Intercambia el valor en el acumulador A por el que se encuentre en el tope de la pila.
Código: 1b
Banderas: no se alteran

--- Instrucciones booleanas.

AND: AND
Saca un número de la pila y hace AND con el acumulador A. El resultado queda en A.
Código: 20
Banderas: ZN se alteran de acuerdo al resultado.

OR: OR
Saca un número de la pila y hace OR con el acumulador A. El resultado queda en A.
Código: 21
Banderas: ZN se alteran de acuerdo al resultado.

XOR: Excluive OR
Saca un número de la pila y hace OR excluyente con el acumulador A. El resultado queda en A.
Código: 22
Banderas: ZN se alteran de acuerdo al resultado.

NOT: NOT
Calcula NOT bit por bit del acumualdor A
El resultado queda en el acumulador A.
Código: 23
Banderas: ZN se alteran de acuerdo al resultado.

--- Operador lógico.

TST: Test
En el acumulador A deja:
1, Si el acumulador A es distinto de cero
0, Si el acumulador A es igual a cero.
Código: 24
Banderas: ZN se alteran de acuerdo al resultado.

--- Instrucciones de salto.

Las instrucciones siguientes hacen un salto sólo si se cumple la una cierta condición, de lo contrario, no efectúan ninguna acción.

JEQ: Jump if Equal:
Código: 30
Condición: Z==1
Destino: PC + Offset

JNE: Jump if Not Equal:
Código: 31
Condición: Z==0
Destino: PC + Offset

JLE: Jump if Less or Equal:
Código: 32
Condición: Z==1 || N==0 && O==1 || N==1 && O==0
Destino: PC + Offset

JGT: Jump if Less or Equal:
Código: 33
Condición: negación de la de JLE
Destino: PC + Offset

JGE: Jump if Greater of Equal
Código: 34
Condición: Z==1 || ! (N==0 && O==1 || N==1 && O==0)
Destino: PC + Offset

JLT: Jump if Less Than
Código: 35
Condición: negación de la de JGE
Destino: PC + Offset

JMP: Jump
Código: 38
Condición: Ninguna.
Destino: PC + Offset

JMP [B]
Código: 39
Condición: Ninguna.
Destino: [B]

--- Subrutinas.

JSR: Jump Subroutine
Guarda el valor del PC en la pila y carga el entero que sigue en el PC.
Código: 3a

JSR [b]
Guarda el valor del PC en la pila y carga el entero contenido en B en PC
Código: 3b

RTN: Return
Desempila un entero y guarda su valor en PC.

Código: 3c

--- Almacenamiento y recuperación de información.

CLR: CLear Accumulator

Carga un cero en el acumulador

Código: 40 para A

Código: 41 para B

LDA: LoAD Accumulator

Código 42 para LDA #int; A= int

Código 43 para LDA [addr]; A= Mem[addr]

Código 44 para LDA [B]; A= Mem[B]

Código 45 para LDA [BP+offset]; A= Mem[BP+offset]

STA: STore Accumulator

Código 4b para STA [addr]; Mem[addr]= A

Código 4c para STA [B]; Mem[B]= A

Código 4d para STA [BP+offset]; Mem[BP+offset]= A

LEA: Load Effective Address

Código 80 para LEA addr B= addr

Código 81 para LEA BP+addr B= BP+addr

Código 82 para LEA [BP+addr] B= Mem[BP+addr]

--- Manipulación de la pila

PSH: PuSH

Mete información a la pila

Código 50 para PSH A; Mem[SP]= A; SP+= 2;

Código 51 para PSH B; Mem[SP]= B; SP+= 2;

Código 52 para PSH BP; Mem[SP]= BP; SP+= 2;

Código 53 para PSH int; SP+= int;

POP: POP

Saca información de la pila

Código 60 para POP A; SP-=2; A= Mem[SP];

Código 61 para POP B; SP-=2; B= Mem[SP];

Código 62 para POP BP; SP-=2; BP=Mem[SP];

Código 63 para POP int; SP-= int;

--- Manipulación de registros

LBP: Load Base Pointer

Carga información en el registro BP

Código 70 para LBP B; BP= B ;

Código 71 para LBP SP; BP= SP;

Código 72 para LBP #val; BP= val

LSP: Load Stack Pointer

Carga información en el registro SP

Código 78 para LSP B; SP= B;

Código 79 para LSP BP; SP= BP

Código 7a para LSP #val; SP= val

--- Otras instrucciones

NOP: No OPeration

Esta instrucción no hace acciones

Código: 00

No afecta banderas.

ADB ADD accumulator A to B.

Suma el valor del acumulador A al del acumulador B
El resultado queda en B.

Código: 88

IRQ:

Código 3D

Servicios:

IRQ 0 detiene a la maquina virtual

IRQ 1 imprime el entero en el acumulador

IRQ 2 imprime una cadena cuya dirección esta en b

IRQ 3 lee un entero de la entrada estandar

IRQ 5 imprime el caracter que hay en a (parte baja)

IRQ 7 procesa una tabla de switch:

en la pila se han empilado

el valor de la expresión del switch

la direccion donde empieza la tabla

el número de casos que hay

la última entrada de la tabla debe contener

un cero y la dirección "default" o la

dirección de salida si no existe "default"

vm_hlt=0

pr_int=1

pr_str=2

rd_int=3

pr_chr=5

switch=7