

4) SET R1, 0x03

SET R2, 0x00

SET R3, RA: ~~ALMACENAMOS~~ 'X'

STR [0x00] RA: ~~GUARDAMOS X,~~
STR USA
SOLO REGISTROS

STI:

HABILITAMOS INTERRUPTS.

LOOP:

CMP R1, R2

JZ FIN

JMP LOOP

LOOP HASTA QUE

R1 == 0 (SIN INTERRUPTS
NO PASA)

FIN:

CLI

DES HABILITAMOS
INTERRUPTS

HALT:

JMP HALT

FIN

RA:

DEC R1

IRET

POSICION DE MEMORIA
'X'

O CUANDO SIMULAMOS UNA INTERRUPT,

SALTAMOS A R_{A_i} Y DISMINUIMOS R_1
O TRAS 3 INTERRUPTOS, $Z=0$ Y EL
LOOP TERMINA

O SI SE INTERRUPE ANTES DE
STI, NO PASARIA NADA YA QUE
 $I=0$

O SI SE INTERRUPE VARIAS VECES
ANTES DE CMP, PODRIA NO CORRERSE
NINGUNA VEZ EL SAME LOOP

O SI SE INTERRUPE ENTRE CMP
Y STI CUANDO $R_1=0$, EL VALOR
DEL REGISTRO SE IRIA A <0 Y
EL PROGRAMA HARIA COSAS INESPERADAS

1) a)

FETCH: UC, PC, MEMORIA

DECODE: DECODE

EXECUTE: UC

b) DETECTARSE: I.C

ATENDERSE: U.C

c) ROM EN LA UC

- d)
- BUSCAR EL OPCODE EN LA CARTILLA
 - ADELANTAR Y CERRAR A LA DERECHA DEL OPCODE
 - COMENZAR A LEER LA ROM DE LA UC EN LA POSICION $OPCODE + 0000$

e) LA MAQUINA DECIDE SI EJECUTAR O NO UN SALTO MEDIANTE LA COMBINACION DE LAS CONDICIONES:

- SE ACTIVA LA M.I. DE SALTO
- SE ACTIVA SU FLAG ASOCIADA

SI SE REALIZA EL SALTO, EL MPC AUMENTA 2 Y SE CARGA UN PC, SINO, EL MPC AUMENTA 1 Y SE SIGUE DESDE DONDE ESTABA

2) b)

0: EL MPC SOLO SE PUEDE
CARGAR O INCREMENTAR EN
1 O 2 (SALTOS)

1: EL MPC SOLO SE PUEDE
INCREMENTAR EN 1 O 7 (ENTRADA)
(COMO ZLIC. TODAVIA NO EXISTE, SOLO
SE PODRIA HACER INC 1)

3) b)

0: IGUAL

1: IGUAL (AHORA SE PUEDEN
USAR LAS INTERRUPT / INC 7)

