

TALLER 4

(1)

a) 256 CELDAS DE 8 BITS.
256 BYTES.

b) EL PC TIENE 8 BITS, PARA PODER ACCEDER A LAS 256 CELDAS DE MEMORIA
O CON 0000 0000 HASTA LA ULTIMA 255 CON 1111 1111.

c) EL INSTRUCTION REGISTER SE ENCUENTRA EN LA UNIDAD DE CONTROL Y ES DE 512 CELDAS DE 32 BITS. 2048 BYTES

d)

?? ACTUALMENTE NO

21 = 2^5 - 1 ESCUADRAS
+ EL OPCODE 00000 que
ACTUALMENTE NO SE USA

e) $\underbrace{12}_{\text{C}} \cdot (2^{\text{A}} - 1) + \underbrace{2}_{\text{B}} \cdot (2^{\text{B}} - 1)$
 $+ 4(2^{\text{C}} - 1) = 36846$

OPOS: MUCHAS INSTRUCCIONES

O CONTAS: UN DECODER MUY COMPLICADO

(2)

a) INC: INCREMENTA EL VALOR
DEL PC EN 1.

b) OPW: PERMITE ESCRIBIR EL
ESTADO ACTUAL DE LAS FLAGS

c)

o SI LA MICROI. ES UN SALTO,
SE ACTIVAN LAS SEÑALES LDAD_MICO
JC/SZ/SN MICRODE

• AHIENDE EL MICROPC PUEDDE VOLVER
SU VALOR MÁS 1 O MÁS 2, DEPENDIENDO DE SI LA FLG6 ASOCIADA
AL SALTO ESTÁ ENCONDIGADA.

• SI PEGAS ESTO AL MPC SUMARÁ
2 Y SI UNA MICROF. DE LOS
CONTENIDOS TUMPLANTES 1 Y CORRER
OTRA (SI TERMINA LA INSTRUCCIÓN)

d) LA SEÑAL PERMITIR VOLCAR
AL DATA EL VALOR/PARAMETRO
INMEDIATO DE LA INSTRUCCION
DECODIFICADA. ESTA SEÑAL ES CONTROLADA
POR LA U.C.

OJUEZ REGISTRO LEER Y ESCRIBIR CO
DECIEN LAS SEÑALES RB - SELECT INDEXIN
Y RB - SELECT INDEX OUT JUNTO A LOS
VALORES POST-DECODE X, Y Y SHIFT (ALU)

(3)

a)

$$\circ R_0 := FF$$

$$\circ R_1 := 11$$

$$\circ R_0 := R_0 + R_1 \oplus$$

$$\circ Si \ C=1$$

b) Loop



INFINITO SMP HALT

b) JMP SEGUIR

10100

SEGUIR: SET R₀, 0xFF

000|

SET R₁, 0x11

0000

SIGUIENTE: ADD R₀, R₁ ✓

0001

JC SIGUIENTE

HALT: JMP HALT

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	A
00 A0	02 02	F8 F8	FF FF	F9 F9	11 11	09 09	20 20	A8 A8	06 06	

FG PF FG 11 DS 20 ZS 06 AO 04
 SRT R₀, 0x FF 1111 | 7 000 | 1111 | 1111
 F 8 F F
 1111 | 007 | 0007 | 0007
 F 9 7 7
 //

ADD R₀, R₁ 0000 | 1 000 | 001 0 | 0000
 0 8 2 0

JC SIG 1010 | 1 000 | 0000 | 0002
 A 8 0 2

JMP HALT 1010 | 0 000 | 0000 | 0100
 A 0 0 4

c) 32 CICLOS HASTA LLAMAR A

APUNTAZ LA MEMORIA A 00

d) 5.

R_B-EN OUT ALU-EN A R_B-SELECT INDEX OUT
= 0

R_B-EN OUT A_W-EN B R_B-SELECT INDEX OUT = 1

ALU-OP = ADD ALU-OP W

R_B-EN IN ALU-EN OUT R_B-SELECT INDEX IN = 0

RESET-MICROOP

4.

SC-MICROOP LOAD MICROOP

RESET-MICROOP

DE-EN OUT FMM PC-LOAD

RESET-MICROOP

e) 32 Hz \leftrightarrow "32 CICLOS POR SEGUNDO"
CICLO = MICROINSTRUCCION

• LA EJECUCION ENTERA Tarda 1 SEGUNDO
= 1000 ms

- JMP SEQWIR : 2 CICLOS = 0,0625 s = 62,5 ms
- SET Rx, CTE: 2 ciclos = 62,5 ms
- ADD Rx, Ry: 5 ciclos = 156,25 ms
- JC M : 4 ciclos = 425 ms

4) • MICROORGANISMOS

- LA UC. NO CONOCERIA EL ESTADO DE LAS FLAGS DE LA MAQUINA Y LOS
- b) LA UC. PENSARIA QUE TODAS LAS FLAGS TIENEN 0 o 1

• REGISTERS

- queremos Fijo ENOUT, si en 0,

NO SE PODRÁN LLEVAR LOS REGISTROS
SI EN 9, SIEMPRE SE VOLCARÁ AL EL
CONTENIDO DEL ALGUNO REGISTRO AL DATA
PATH
(b) y c) SIEMPRE 0 o 1

- PC

- a) NO SE PODRÍA CAMBIAR ENTRE
EL MODO DE MODIFICACIÓN DEL PC
- b) NO SE PODRÍA CARGAR VALOR,
NO SE PODRÍA SACTAR
- c) SOLO SE PODRÍA HACER SALTO
EN EL PC Y. NO INC.

- ALU

- a) NO SE PODRÍA OPERAR LA ALU: NO
SE PUEDE DECIR QUE VALOR SE
ALMACENA EN EL REGISTRO INTERNO si:
EN LA FLAG DE CARRY.
- b) FLAG C = 0 SIEMPRE
 $ALU - 226 - OUT = 0000 \ 0000$ SIEMPRE
- c) FLAG C = 0 SIEMPRE
 $ALU - 226 - OUT = 1111 \ 1111$ SIEMPRE

• MEMORIA

- a) NO SE PUEDEN CARGAR TOSKONES
EN MEMORIA, SE TRAUDIA QUE ESTAN
EN LA PRIMERA Y HACIA INC.
- b) Y c) SOLO SE PODERÁ CARGAR
LA PRIMERA O ULTIMA CELDA DE LA
MEMORIA

• DECODE

- a) NO SE PODERÁ CARGAR
AL INMEDIATO EN AL REGISTER M
DEL DECODER. SOLO SE PODRÁ USAR
0000 PARA INSTRUCCIONES

- b) M VALE SIEMPRE 0000 0000
c) M VALE SIEMPRE 1111 1111

• CONTROL UNIT

- a) NO SE PODRÁN CARGAR
NUEVOS MICROPC, SOLO ARRANCAR DESDE
0 Y HACER INC

b) SET R₂ 0000000000000000 CARGAL

0000000000000000 BN RL MICROPC

c) ↑ TDEM CON 11111111

S) SET R₀, 0x0005 ← PLUGBR
m=s

SET R₁, 0x0000

MAIN :

ADD R₁, R₀

DEC R₀

JZ FIN

JMP MAIN

FIN :

JMP FIN

