

Rel. Problemas 3

1. Considere la ALU y registros de la Figura 1. La interpretación de los pines de control se resume en la Tabla. Cada palabra de control se especifica con el formato $C_4 C_3 C_2 C_1 C_0$

C_3	C_2	F
0	0	$R+S$
0	1	$R-S$
1	0	$R \text{ and } S$
1	1	$R \text{ xor } S$

$C_3 C_0$	Entrada R	Entrada S
00	A	A
01	A	B
10	B	A
11	B	B

C_4	Acción
0	$B \leftarrow R$
1	$A \leftarrow R$

a) ¿Cómo se puede poner el registro A a cero? Sugiera, al menos, dos maneras de hacerlo. No dispone de entrada directa para poner a 0 el registro.

$$A \leftarrow A - A : 01101100$$

$$A \leftarrow B - B : 11011111$$

$$A \leftarrow A \text{ xor } A : 1111100$$

b) Sugiera una secuencia de control que intercambie los contenidos de los registros A y B

$$A \leftarrow A + B : 0100101$$

$$B \leftarrow B + A : 1100110$$

$$A \leftarrow A \text{ and } B : 0110101$$

$$B \leftarrow B \text{ and } A : 1110110$$

13. Dado el repertorio de instrucciones en ensamblador siguientes:

b) Realice una codificación del repertorio de instrucciones, teniendo en cuenta que hay 256 posiciones de memoria, cada una de un byte, y que sea lo más sencillo posible para el desarrollo de los siguientes apartados, aunque se desperdicie memoria.

$d \rightarrow \text{Direto (d)} : 1$
 $\text{Dirección} \rightarrow \text{Indirecto ([d])} : 0$

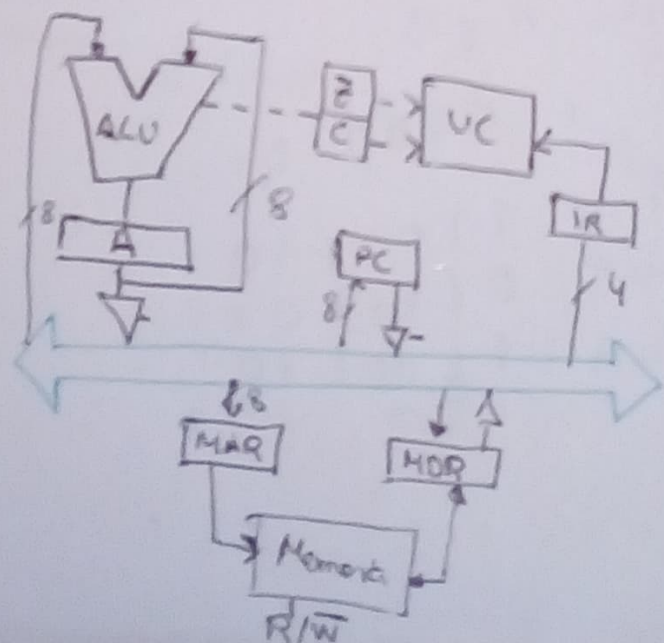
$C \rightarrow Z \text{ (si cero)} : 0$
 $\text{Condición} \rightarrow C \text{ (si acarreo)} : 1$

$f \rightarrow \text{Entrada - izquierda} : 0$
 $\text{Entrada de ALU} \rightarrow A - \text{Ent. Izq.} : 1$

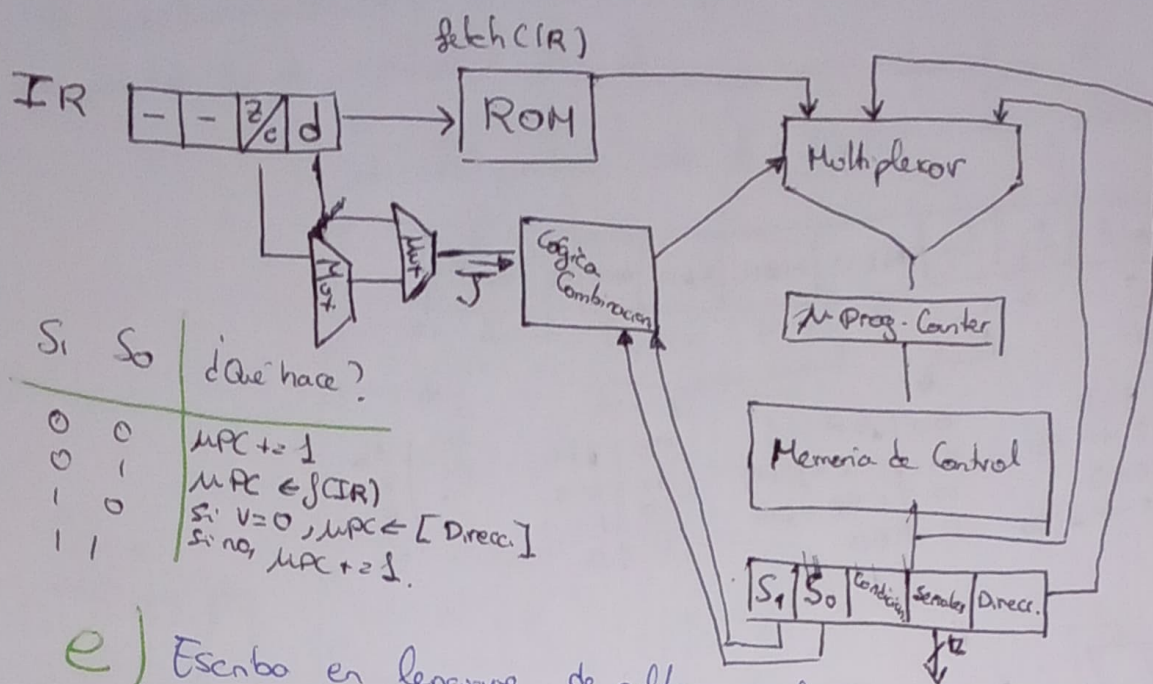
Visto estos generalistas, podremos codificar las instrucciones así:

	<u>Condición</u>	<u>Dirección d</u>
LOAD	00	fd
SUB	00	fd
STORE	01	0d
Jz	10	0d
Jc	10	1d
JND	11	0d

c) Disme el ensayo de datos del ordenador que contenga este repertorio de instrucciones



d) Diseña la unidad de control



e) Escribo en lenguaje de alto nivel el contenido de la memoria de control.

Fetch: MAR ← PC

MBR ← M[MAR]

PC ← PC + 1

IR ← MBR

MAR ← PC

MBR ← M[MAR], PC ← PC + 1

MAR ← MBR

if (CIR[0] = 0) goto Contenido
goto f(CIR)

JCondicionales: if (J=0) goto Fetch;

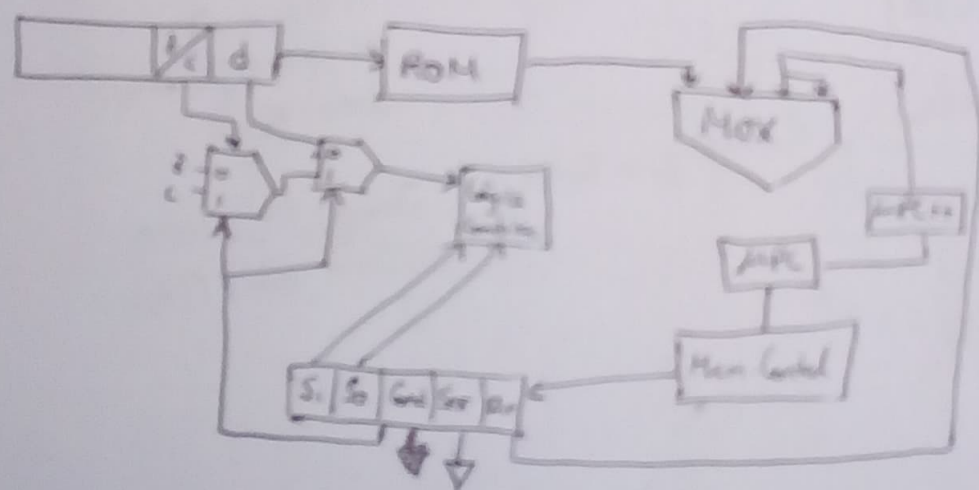
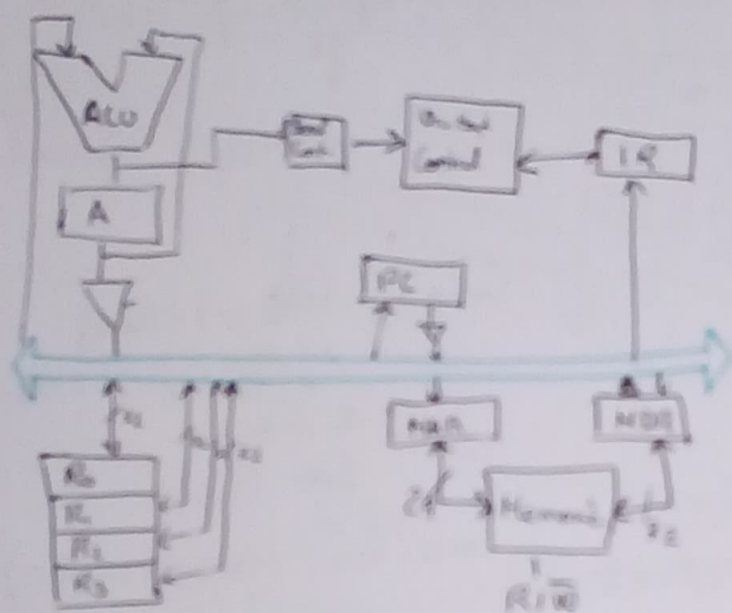
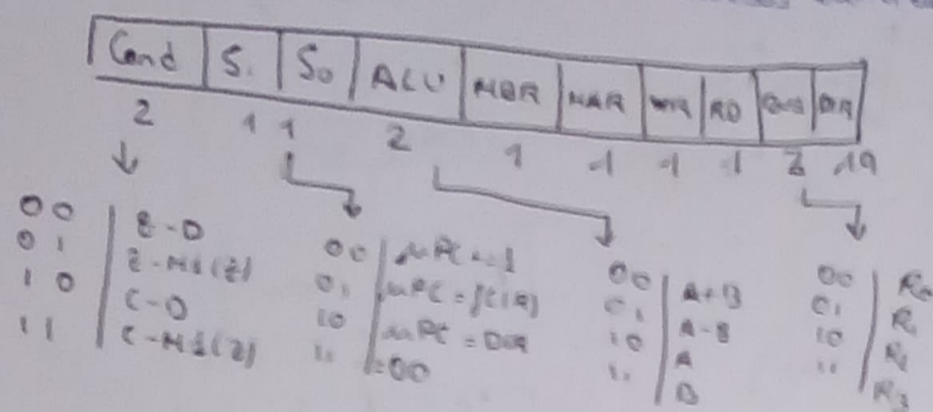
Load/Sub: MBR ← M[MAR]
if (CIR[1] = 1) A ← A - MBR
A ← MBR
goto Fetch;

Contenido: MBR ← M[MAR]
MAR ← MBR
goto f(CIR)

Store: MBR ← A
M[MBR] ← MBR
goto Fetch

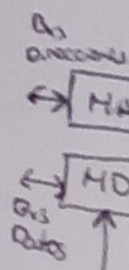
Jump: PC ← MBR
goto f Fetch.

15. Se quiere diseñar un computador microprogramado de 32 bits de ancho de palabra, con 24 bits de direccionamiento. Diseñar el formato de instrucciones, así como la ruta de datos y la UC del computador, estableciendo la conexión de los diversos elementos.



17.

a) Dibuja el camino de datos a partir de la descripción previa

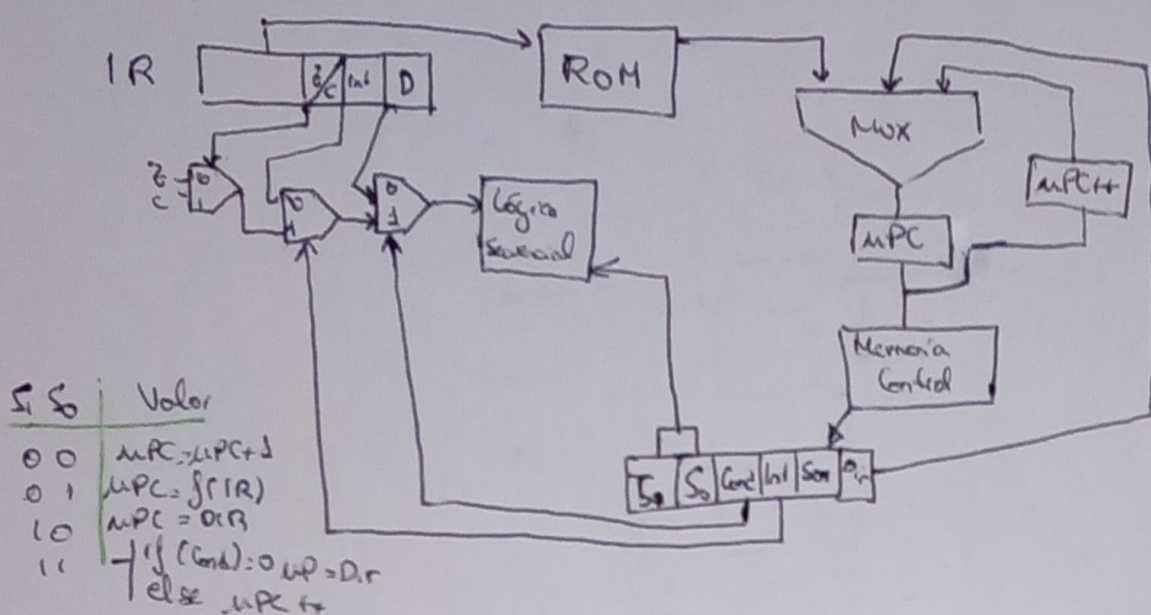


Cond		Lex
01	01	01
00	x	00
01	1	01
10	0	00
11	1	10

Logia General

$$\begin{aligned} \text{Max}_0 &= \text{Condo}_0 \cdot \text{Condo}_1 + \text{Condo}_0 \cdot \text{Flag} \\ \text{Max}_1 &= \text{Condo}_0 \cdot \text{Condo}_1 \end{aligned}$$

19. a) Diseña la UC microprogramada.



b) Escribe los pasos del microprograma correspondientes a

Captación $MAR := PC;$
 $MBR := M[MBAR];$
 $PC := PC + 1;$
 $IR := MBR; \text{ goto } f(IR);$

Interrupción $\text{if } (INT = 0)$
 $MBR := PC; \text{ goto Interrupción};$
Desactivación de Int $\text{if } (I = 1)$
 $MAR := IR; \text{ goto Captación};$

20. Escriba un microprograma que realice la fase de captación de instrucciones y la fase de ejecución de cada una de esas instrucciones.

Fetch $MAR := PC;$
 $MBR := M[MBAR];$
 $PC := PC + 1;$
 $IR := MBR;$
 $\text{goto } f(IR);$

Addr $R_{dst} := R_{dst} + R_{src};$
 $\text{goto Fetch};$

Addi $MAR := IR;$
 $MBR := M[MBAR];$
 $R_{dst} := R_{dst} + MBR;$
 $\text{goto Fetch};$

Addx $MAR := IR;$
 $MBR := M[MBAR];$
 $R_{dst} := R_{dst} + MBR;$
 $\text{goto Fetch};$