

# Estructura de Computadores

## Tema 5. La unidad de control

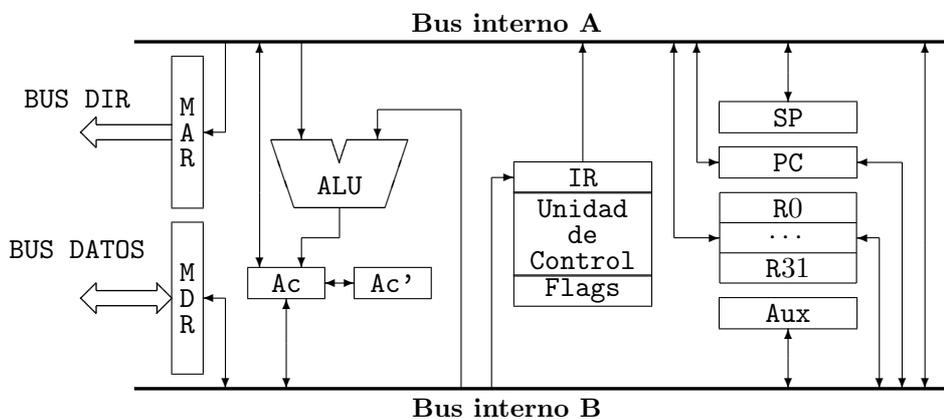
- Estructura interna de la CPU.
- Estrategias de diseño de unidades de control.
- Unidad de control cableada.
- Máquina de Wilkes.
- Microprogramación.
- Horizontalidad y verticalidad de la palabra de control.
- Nanoprogramación

## 1. Ejercicios Resueltos

### 1.1.

Dada la CPU con la estructura interna que se muestra en la figura,

1. Determinar la lista de señales de control (microórdenes) que emitiría la Unidad de Control para las instrucciones de carga en el registro acumulador (Ac) del dato obtenido con los siguientes modos de direccionamiento:
  - a) Inmediato (LDA constante)
  - b) Directo (LDA [dirección])
  - c) Indirecto (LDA [[dirección]])
  - d) Indexado (LDA [Rxx])
2. Determinar los formatos de instrucción y la secuencia de microórdenes que emitiría la Unidad de Control para las siguientes instrucciones:
  - a) Salto incondicional absoluto (BR dirección)
  - b) Salto a subrutina (CALL dirección) y retorno de subrutina (RET). Suponer que se gestiona la pila con el registro SP (puntero de pila), que mantiene la dirección de la cima de la pila.



Los registros *Ac* y *Aux* no son accesibles al programador. Los registros *PC* y *SP* tiene la capacidad de autoincremento y autodecremento, sin emplear la *ALU*.

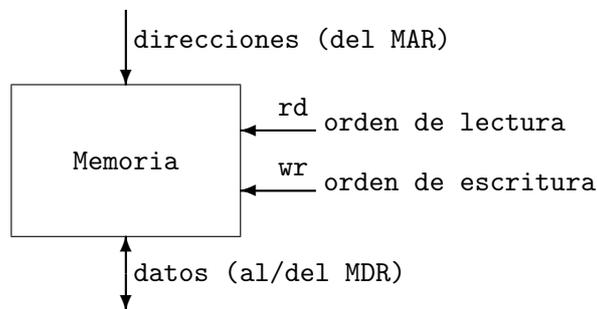
**Solución**

(1) Tras la extracción de la instrucción, que no se considera, las micro-órdenes para realizar cada una de las instrucciones propuestas son:

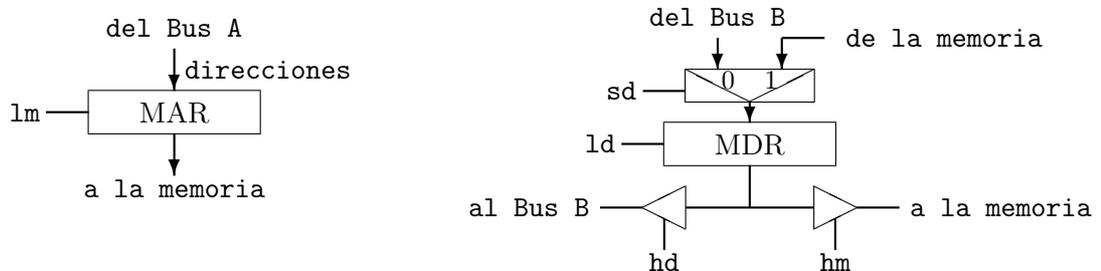
- a)  $Ac \leftarrow IR\langle \text{dirección} \rangle$  ; a través del bus A  
ya que el dato inmediato (constante) forma parte de la instrucción ya almacenada en *IR* en el ciclo de *fetch*.
- b)  $MAR \leftarrow IR\langle \text{dirección} \rangle$   
 $MDR \leftarrow M[MAR]$   
 $Ac \leftarrow MDR$  ; a través del bus B.
- c)  $MAR \leftarrow IR\langle \text{dirección} \rangle$   
 $MDR \leftarrow M[MAR]$   
 $MAR \leftarrow MDR$   
 $MDR \leftarrow M[MAR]$   
 $Ac \leftarrow MDR$
- d)  $MAR \leftarrow R[IR\langle \text{dirección} \rangle]$   
 $MDR \leftarrow M[MAR]$   
 $Ac \leftarrow MDR$

Las partes de la máquina afectadas por las anteriores micro-instrucciones se verán con más detalle, destacando las señales de control en cada parte.

- la memoria



- los registros *MAR* y *MDR*

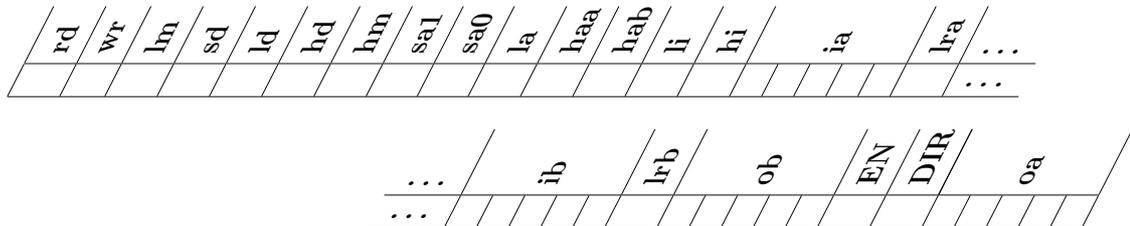


- el registro acumulador



EN	DIR	operación
1	×	buses desconectados
0	0	conecta B con A, transmite en ese sentido
0	1	transmite de A hacia B

La lista de señales de control empezaría así:



Y las micro-órdenes anteriores serían:  
para

$$Ac \leftarrow IR\langle \text{dirección} \rangle$$

rd	wr	lm	sd	ld	hd	hm	sa	la	haa	hab	li	hi	ia				...	
0	0	0	0	0	0	0	0	0	1	0	0	1	x	x	x	x	x	...
...																		
lra		ib				lrb		ob				EN	DIR	oa				
0		x	x	x	x	x	0	x	x	x	x	x	1	x	x	x	x	x

con  $sa = 00$  que hace que la entrada al acumulador sea desde el Bus A, y activando las señales  $la$  y  $hi$  y quedando todas las demás en reposo o *don't care*.

Igualmente, para la micro-operación

$$MAR \leftarrow IR\langle \text{dirección} \rangle$$

se activan las señales  $\{ lm, hi \}$ .

Para las restantes micro-operaciones:

$$MDR \leftarrow M[MAR] \quad : \{ sd=1, ld, rd \}$$

$$Ac \leftarrow MDR \quad : \{ hd, sa=11, la \}$$

$$MAR \leftarrow MDR \quad : \{ lm, hd, EN=0, DIR=0 \}$$

$$MAR \leftarrow R[IR\langle \text{dirección} \rangle] \quad : \{ lm, oa=\text{reg}(\text{dirección}), hra \}$$

en cada caso, las señales que no se nombran estarán inactivas o serán *don't cares*.

## (2) Formatos de instrucción.

Tanto BR como CALL tendrán un formato semejante

CÓDIGO DE LA OPERACIÓN	DIRECCIÓN
------------------------	-----------

en cambio RET no tiene operandos ni campo de dirección, sólo *opcode*.

Las secuencias de micro-órdenes que ejecutan a estas instrucciones son:

BR: $PC \leftarrow IR\langle \text{dirección} \rangle$
--

CALL: $SP \leftarrow SP + 1 \parallel MDR \leftarrow PC$ $MAR \leftarrow SP$ $M[MAR] \leftarrow MDR$ $PC \leftarrow IR\langle \text{dirección} \rangle$
--

RET: $MAR \leftarrow SP$ $MDR \leftarrow M[MAR]$ $PC \leftarrow MDR \parallel SP \leftarrow SP - 1$
---

las tres primeras micro-operaciones empujan en la pila a la dirección de vuelta.

---

### 1.2.

Completar el siguiente cronograma, que corresponde a un ciclo de lectura de un bus asíncrono.

<i>Líneas de estado</i>
<i>Líneas de dirección</i>
$\overline{\text{Lectura}}$
<i>Líneas de datos</i>
$\overline{\text{Reconocimiento}}$

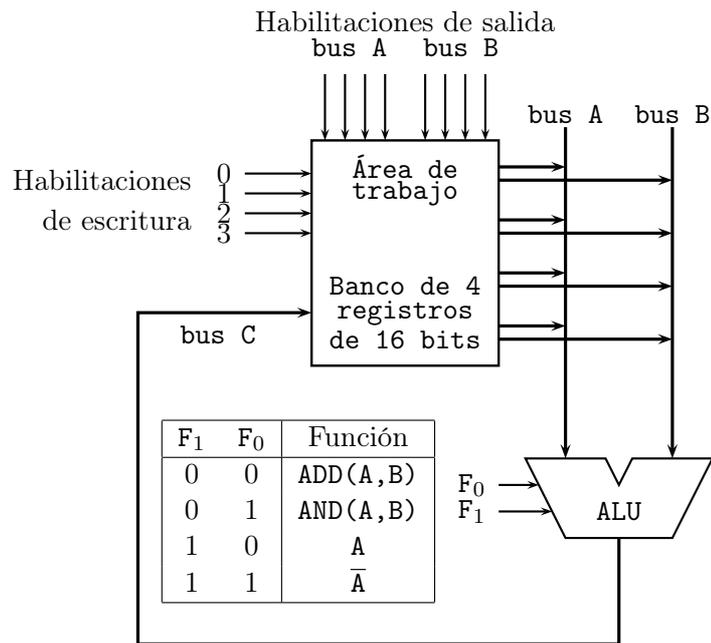
### **Solución**

<i>Líneas de estado</i>
<i>Líneas de dirección</i>
$\overline{\text{Lectura}}$
<i>Líneas de datos</i>
$\overline{\text{Reconocimiento}}$

---

### 1.3.

Con referencia a la figura adjunta, los registros 0, 1 y 2 son registros de uso general. El registro 3 contiene el valor fijo +1, y no puede ser modificado.



a) Escribir una secuencia de control que calcule la diferencia entre los contenidos de los registros 0 y 1, en representación de complemento a 2, y deje el resultado en el registro 0. Simbólicamente se puede expresar lo pedido como  $r0 \leftarrow r0 - r1$ . No modificar registro alguno excepto  $r0$  y  $r1$  (si hiciera falta). Completar la tabla siguiente con ceros o unos (usando ceros cuando el valor a asignar sea irrelevante). Suponer que si no se selecciona registro alguno para los buses A o B, el bus adopta el valor 0.

Habilitaciones de escritura				Habilitaciones del bus A				Habilitaciones del bus B						Tiempo
0	1	2	3	0	1	2	3	0	1	2	3	F <sub>1</sub>	F <sub>0</sub>	
														0
														1
														2

b) Escribir una secuencia de control que calcule la XOR de los contenidos de los registros 0 y 1, dejando el resultado en el registro 0. Simbólicamente, se puede expresar lo pedido como  $r0 \leftarrow XOR(r0, r1)$ . Utilizar el mismo estilo de solución que para la parte a.

**Solución**

(a) La ALU no tiene la operación de resta, de modo que se hará:

$$r0 \leftarrow r0 - r1 = r0 + (-r1)$$

y habrá que cambiar el signo de  $r1$ . En C2 el cambio de signo se consigue invirtiendo bit a bit todo el número y sumando 1 a continuación, de modo que será:

$$-r1 = (\overline{r1}) + 1$$

que puede realizarse así:

$$r1 \leftarrow \overline{r1}$$

$$r1 \leftarrow r1 + 1 = r1 + r3$$

Así pues, la operación de resta es:

$$\begin{aligned}
 r1 &\leftarrow \overline{r1} && ; F1F0 = 11, busA = 0100, busB = 0000, busC = 0100 \\
 r1 &\leftarrow r1 + 1 = r1 + r3 && ; F1F0 = 00, busA = 0100, busB = 0001, busC = 0100 \\
 r0 &\leftarrow r0 + r1 && ; F1F0 = 00, busA = 1000, busB = 0100, busC = 1000
 \end{aligned}$$

Habilitaciones de escritura				Habilitaciones del bus A				Habilitaciones del bus B				F <sub>1</sub>	F <sub>0</sub>	Tiempo
0	1	2	3	0	1	2	3	0	1	2	3			
0	1	0	0	0	1	0	0	0	0	0	0	1	1	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0	1	0	0	0	0	2

(b) Como antes:

$$XOR(r0, r1) = AND(r0, \overline{r1}) + AND(\overline{r0}, r1)$$

En este caso, primero sustituimos la operación *OR* entre los productos por la suma aritmética, que aparte del acarreo es equivalente, y segundo, debemos emplear un registro adicional, el *r2*, para mantener un resultado parcial:

$r2 \leftarrow \overline{r1}$	0	0	1	0	0	1	0	0	0	0	0	0	1	1
$r2 \leftarrow r0 \wedge r2$	0	0	1	0	1	0	0	0	0	0	1	0	0	1
$r0 \leftarrow \overline{r0}$	1	0	0	0	1	0	0	0	0	0	0	0	1	1
$r0 \leftarrow r0 \wedge r1$	1	0	0	0	1	0	0	0	0	1	0	0	0	1
$r0 \leftarrow r0 + r2$	1	0	0	0	1	0	0	0	0	0	1	0	0	0

#### 1.4.

¿Cuál es la función general de la unidad de control de un procesador?

#### Solución

Es la unidad que controla el funcionamiento paso a paso de un computador. Normalmente realiza estas tareas:

- Interpretación de las instrucciones.
- Secuenciado de las instrucciones.

#### 1.5.

Bosqueje un proceso de tres pasos que conduzca a la unidad de control de un procesador.

#### Solución

1. Definir los elementos básicos del procesador.
2. Describir las microoperaciones que ejecuta el procesador.
3. Determinar las funciones que debe realizar la unidad de control para hacer que se ejecuten las microoperaciones.

## 1.6.

¿Qué tareas básicas realiza una unidad de control?

### **Solución**

La Unidad de Control de un procesador realiza dos tareas: (1) hace que el procesador ejecute las microoperaciones en la secuencia correcta, determinada por el programa que se está ejecutando, y (2) genera las señales de control que provocan la ejecución de cada microoperación.

---

## 1.7.

Explicar las funciones de una unidad de control en un ordenador digital.

### **Solución**

La unidad de control lee o toma una instrucción de la memoria y la decodifica. En el caso de las instrucciones lógicas o aritméticas, la unidad de control genera las señales necesarias para que la ALU ejecute las operaciones deseadas. En el caso de instrucciones de entrada/salida, la unidad de control genera señales para que los dispositivos de E/S reciban o envíen datos desde o hacia los dispositivos externos.

---