

## LABORATORIO DE ARQUITECTURA DE COMPUTADORES.

### I. T. I. SISTEMAS / GESTIÓN

#### GUÍA DEL ALUMNO

<i>Práctica 4: La Unidad de Control</i>
---

#### **Objetivos**

- Comprender qué es y cómo funciona la Unidad de Control.
- Diseñar una U.C. cableada mediante el método de la Tabla de Estados.

#### **Método de trabajo**

Lee cuidadosamente las indicaciones de la guía, realizando los pasos que se indican. La guía tiene como propósito que **TÚ** te asegures de que comprendes lo que va sucediendo mientras realizas lo que te indica que hagas.

Para todas las sesiones, necesitaremos textos de referencia del lenguaje para realizar programas o para comprender lo que ocurre durante las simulaciones. Por lo tanto se recomienda traer libros, apuntes y cualquier bibliografía de VHDL que sirva para hacer consultas. En el apartado "bibliografía" se proporcionan algunas referencias.

También traer un **disquete** o una **memoria USB** para llevar copia de los ficheros que se generen y se deseen conservar ya que, al tratarse de una máquina compartida, no hay garantías de que más tarde encontremos los trabajos que dejamos hoy. Pero dado que V-System no trabaja bien con los disquetes siempre habrá **que copiar de/al disco duro y trabajar en él.**

#### **Bibliografía**

- Estructura y diseño de computadores. 3-apéndices. David A. Patterson • John L. Hennessy.
- Buscar en la WEB: Google encuentra referencias y pueden encontrarse libros en pdf

## **Introducción**

La **Unidad de Control** genera un conjunto de señales de control, activas durante cada fase de ejecución de las distintas instrucciones, cada una de las cuales necesita un ciclo de reloj. La U.C. es una máquina de estados finitos, en donde cada estado representa una microoperación determinada, es decir, un conjunto de señales activadas.

Existen básicamente dos métodos de diseño: las unidades de control cableadas y las microprogramadas.

### **Unidades de Control cableadas**

El controlador o unidad de control es una máquina secuencial del tipo Moore, donde las salidas son completamente dependientes del estado del controlador y no de las entradas. La única entrada X no tiene influencia directa sobre las salidas. El número de estados dependerá del número de operaciones elementales que se realicen.

Otras características del control cableado son:

- Implementación en hardware, por lo tanto muy rápido.
- No es flexible, una modificación posterior implica cambiar el circuito entero.
- Con este control, la Unidad de Control se diseña para una determinada ruta de datos.

La implementación de estos circuitos puede hacerse de diferentes formas: método de la tabla de estados, método de las células de retardo y método del contador-secuenciador. En esta práctica se realizará mediante el método de la tabla de estados.

### **Método de la tabla de estados.**

Las características de este método son:

- Basada en una máquina de estados finitos. Una máquina de estados finitos consta de:

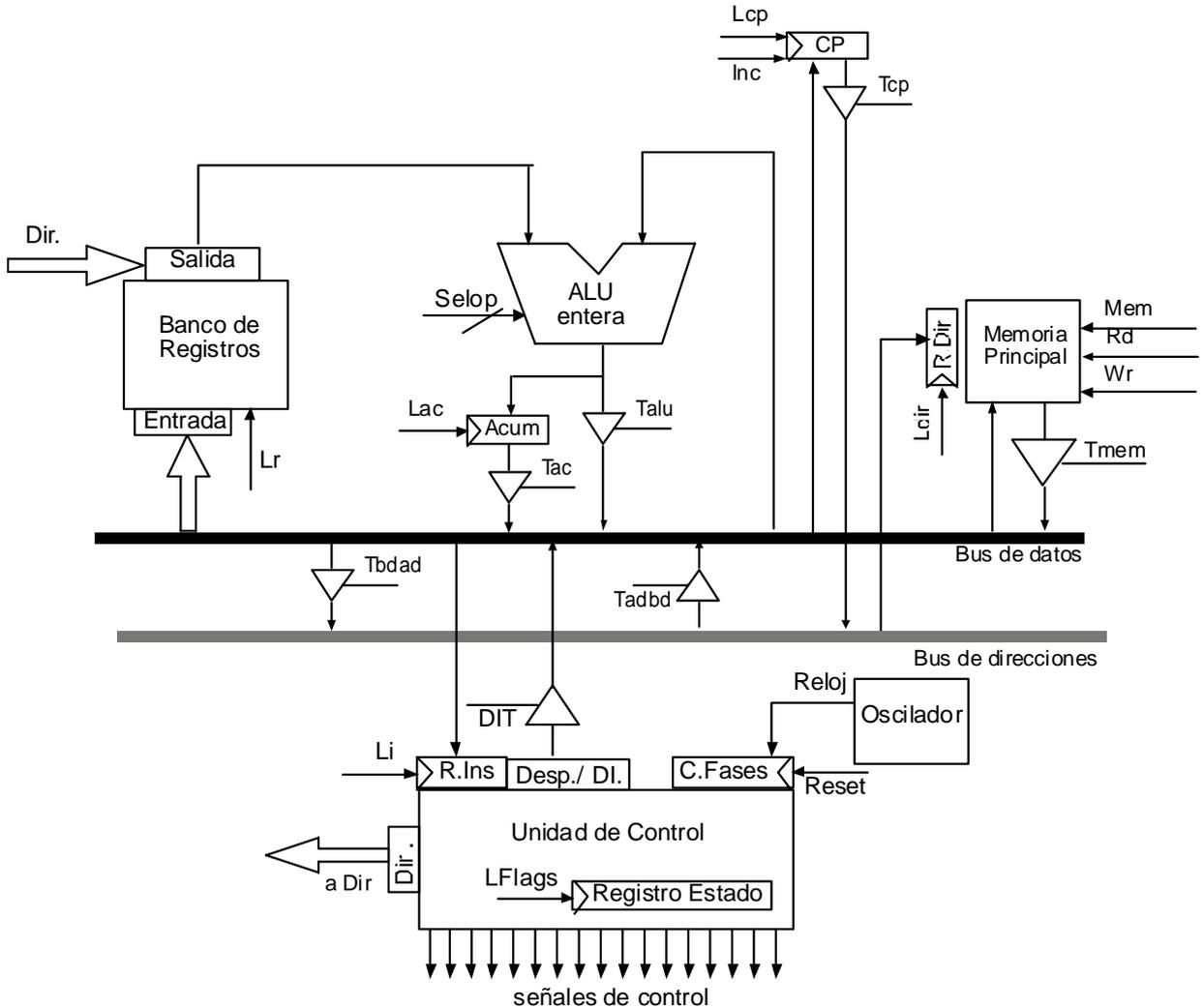
Memoria interna que contiene el estado. Cada estado corresponde a un ciclo de reloj y contiene las operaciones a realizar en ese ciclo.

Dos funciones combinacionales:

- La función de estado siguiente. Es una función combinacional que a partir de las entradas y el estado actual determina el estado siguiente, es decir, dice a qué estado pasa, dependiendo del estado en el que se encuentra.
- La función de salida. Produce el conjunto de señales de control a partir de sus entradas y el estado actual.

### Actividades

Realizar el diseño cableado de la unidad de control del circuito siguiente:



### Pasos:

1.- Máquina de estados indicando las operaciones elementales que se realizan en cada uno de ellos.

Instrucción	Código de operación		
	O3	O2	O1
LD	0	0	0
ST	0	0	1
LDI	0	1	0
ADD R1, R2, R3	0	1	1
SUB R1, R2, R3	1	0	0
JZ Dirección	1	0	1
JC Dirección	1	1	0

JMP Dirección	1	1	1
---------------	---	---	---

### Instrucciones de carga-almacenamiento: LD y ST

CO	R	R	Desplazamiento
(3)	(4)	(4)	(21)

### Instrucción de inicialización de registro: LDI

CO	R1	Inmediato
(3)	(4)	(25)

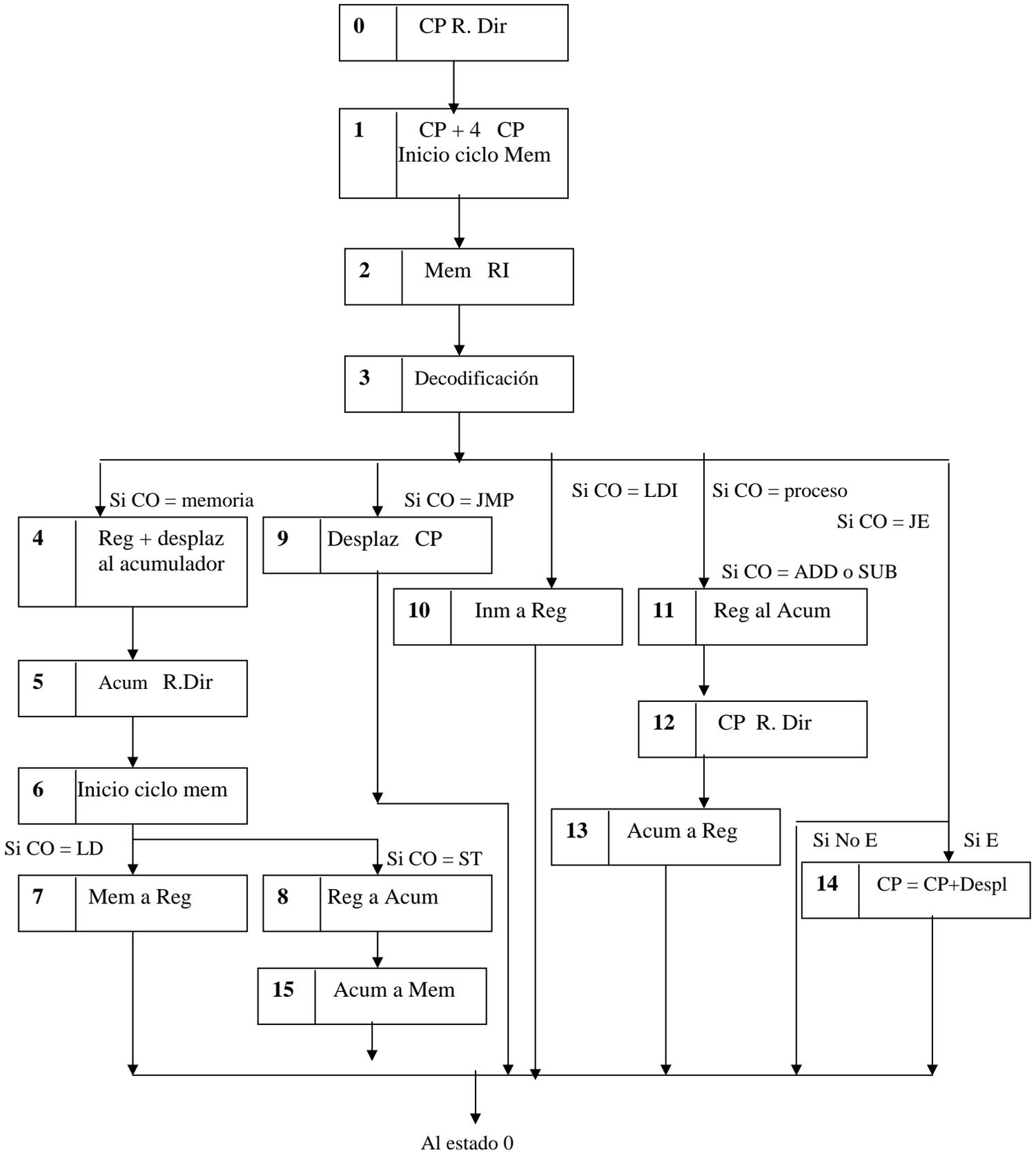
### Instrucciones aritmético-lógicas: ADD y SUB

CO	R1	R2	R3	No usado
(3)	(4)	(4)	(4)	(17)

### Instrucciones de salto condicional /incondicional: JZ, JC Y JMP

CO	Desplazamiento / Dirección directa de memoria
(3)	(29)

GRAFO DE ESTADOS



## 2.- Función de estado siguiente

Estado	Estado nuevo				Condición	Instrucción
	S3	S2	S1	S0		
1	0	0	0	1	Estado Anterior 0	XXX
2	0	0	1	0	Estado Anterior 1	XXX
3	0	0	1	1	Estado Anterior 2	XXX
4	0	1	0	0	Estado Anterior 3	CO = memoria
5	0	1	0	1	Estado Anterior 4	
6	0	1	1	0	Estado Anterior 5	
7	0	1	1	1	Estado Anterior 6	CO = LD
8	1	0	0	0	Estado Anterior 6	CO = ST
9	1	0	0	1	Estado Anterior 3	CO = JMP
10	1	0	1	0	Estado Anterior 3	CO = LDI
11	1	0	1	1	Estado Anterior 3	CO = ADD o SUB
12	1	1	0	0	Estado Anterior 11	
13	1	1	0	1	Estado Anterior 12	
14	1	1	1	0	Estado Anterior 3	CO = JE, si E
15	1	1	1	1	Estado Anterior 8	
0	0	0	0	0	Estado Anterior 7	
0	0	0	0	0	Estado anterior 15	
0	0	0	0	0	Estado anterior 9	
0	0	0	0	0	Estado anterior 10	
0	0	0	0	0	Estado anterior 13	
0	0	0	0	0	Estado anterior 3	CO = JE, si no E
0	0	0	0	0	Estado anterior 14	

Las señales generadas por los estados son:

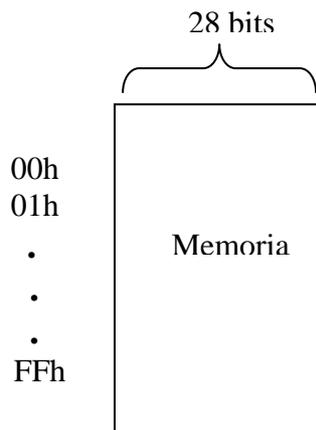
Estado	Op. Elmtales.	Señales
0	CP a reg dir	TCP, LDIR
1	CP = CP + 4	INC, LCP
	Inicio ciclo mem	MEM, RD
2	Memoria a RI	RD, TMEM, LI
3	Decodificación	NADA solamente se miran los flag
4	Reg + desp al acumulador	DIR, DIT, SELOP, LAC
5	Acum a Reg Dir	TAC, TBDAD, LDIR
6	Inicio ciclo mem	MEM
7	Memoria a Reg	RD, TMEM, DIR, LR, RESET
8	Reg a Acumulador	DIR, DIT, SELOP, LAC, WR
9	Destino a CP	DIT, LCP, RESET
10	Inmediato a registro	DIR, DIT, LR
11	Reg al acumulador	DIR, DIT, SELOP, LAC
12	Reg OP Acum al Acum	DIR, DIT, SELOP, TAC, LAC, LFLAGS
13	Acum a Reg	TAC, DIR, LR, RESET
14	CP + Desp al CP	DIT, INC, LCP, RESET
15	Acumulador a Mem	TAC, WR, RESET

La implementación de la U.C. tendrá las siguientes características:

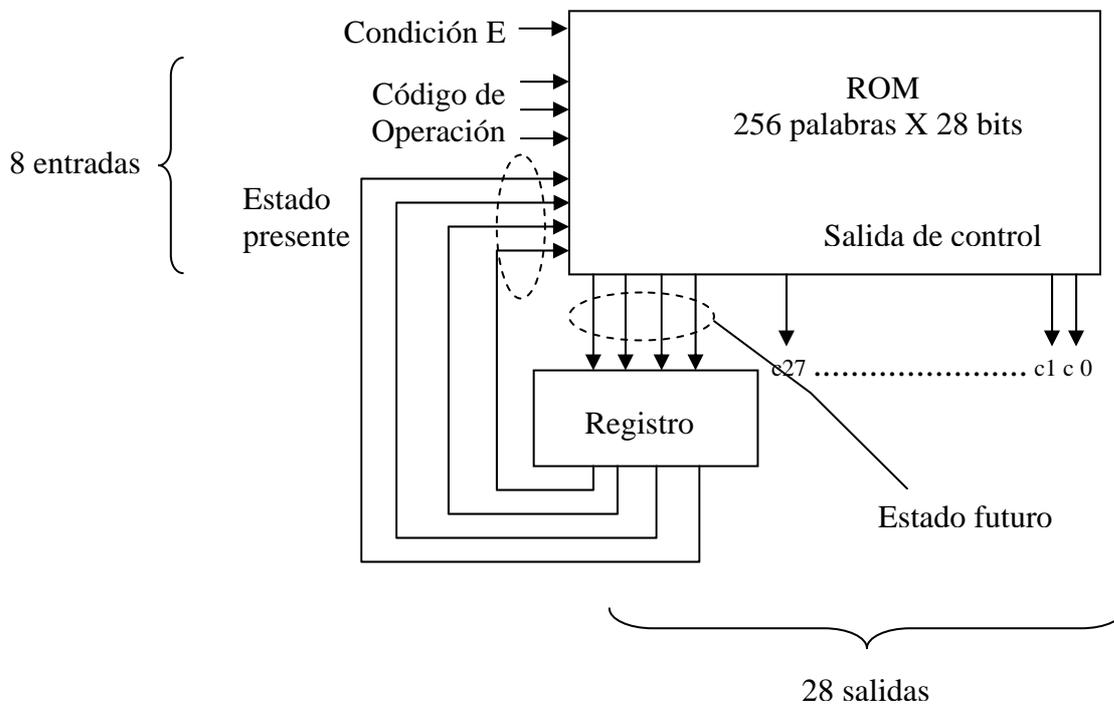
Función salida	Estados															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Señales control</b>																
LCP	0	1	0		0	0	0	0	0	1	0	0	0	0	1	0
TCP	1	0	0		0	0	0	0	0	0	0	0	0	0	0	0
DIR					xxx			xxx	xxx		xxx	xxx	xxx	xxx		
TBDAD	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
LR	0	0	0		0	0	0	1	0	0	1	0	0	1	0	0
SELOP					xxxx				xxxx			xxxx	xxxx			
LAC	0	0	0		1	0	0	0	1	0	0	1	1	0	0	0
TAC	0	0	0		0	1	0	0	0	0	0	0	1	1	0	1
DIT	0	0	0		1	0	0	0	1	1	1	1	1	0	1	0
LI	0	1	1		0	0	0	0	0	0	0	0	0	0	0	0
RESET	0	0	0		0	0	0	1	0	1	0	0	0	1	1	1
LFLAGS	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0
MEM	0	1	0		0	0	1	0	0	0	0	0	0	0	0	0
RD	0	1	1		0	0	0	1	0	0	0	0	0	0	0	0
WR	0	0	0		0	0	0	0	1	0	0	0	0	0	0	1
LDIR	1	0	0		0	1	0	0	0	0	0	0	0	0	0	0
TMEM	0	0	1		0	0	0	1	0	0	0	0	0	0	0	0
INC	0	1	0		0	0	0	0	0	0	0	0	0	0	1	0
TBDAD	0	0	0		0	1	0	0	0	0	0	0	0	0	0	0

- U.C. construida como una máquina de estados finitos.
- Circuito combinacional que puede implementarse mediante una ROM.
- U.C. con 8 entradas (1 bit para la condición E, 3 bits para el código de operación y 4 para el estado actual), lo que implica que se puede representar a la entrada un total de  $2^8 = 256$  posibles combinaciones.
- U.C. con 28 señales de salida (24 bits para las salidas de control y 4 bits para el estado siguiente).

Luego el bloque de memoria ROM para la U.C. sería el mostrado a continuación:



Y el esquema por bloques:



## Ejercicios

- 1) Diseñar la Unidad de Control que active las señales de control para el circuito anterior, utilizando una memoria ROM que cumpla los requisitos especificados y los elementos que se consideren necesarios. Deberá indicarse el contenido de la memoria detallada y razonadamente.

Un ejemplo de implementación de la memoria, sería la siguiente tabla que muestra las entradas  $a_i$  de 8 bits y las salidas  $c_i$  de 28 bits.

Estado actual				Bit E	Código de operación			Líneas de control							Estado siguiente				
a7	a6	a5	a4	a3	a2	a1	a0	c27	c26	c25	c24	.....	c6	c5	c4	c3	c2	c1	c0
0	0	0	0	X	X	X	X	0	0	0	1	.....	0	1	0	0	0	0	1
0	0	0	1	X	X	X	X	0	1	0	0	.....	0	0	1	0	0	1	0
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....

- 2) Realizar el test bench para comprobar su correcto funcionamiento.

**Nota:** junto con el enunciado de la práctica hay un enlace a nuevas fuentes de la memoria ROM y de los paquetes `arq_pack_ext.vhd` y `del_file_pack.vhd`.