

# E.T.S.I.- U.A.M.- ETC I - Examen 18-6-02

**Problema 1:** La memoria de la figura tiene las entradas conectadas a niveles lógicos fijos, salvo A5 y A4 que están conectadas a 2 variables lógicas X1 y X0 respectivamente. Se desea programar esta memoria de manera que los bits de salida (datos) sean cero, con excepción de D4, D3 y D2, que deben realizar las siguientes funciones lógicas:

## TIPO - A

$$D4 = X1 \text{ AND } X0$$

$$D3 = X1 \oplus X0$$

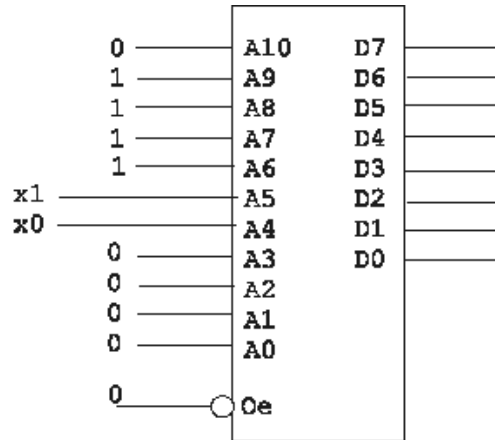
$$D2 = X1 \text{ OR } X0$$

## TIPO - B

$$D4 = X1 \text{ OR } X0$$

$$D3 = X1 \oplus X0$$

$$D2 = X1 \text{ AND } X0$$



Se pide escribir el fichero de programación correspondiente en formato HEX en la tabla adjunta. Utilice únicamente 1 línea o *record* para cada una de las direcciones no consecutivas que se necesitan, considerando que el grabador completa con ceros el resto de las posiciones no especificadas. Escriba un carácter en cada cuadro de la tabla, dejando en blanco los que no utilice. No olvide utilizar una fila para indicar el fin de fichero.

			Tipo - A				Tipo - B			
X1	X2	Dirección (11 bits)	D4	D3	D2	Valor Dato (8 bits)	D4	D3	D2	Valor Dato (8 bits)
0	0	3C0 <sub>16</sub>	0	0	0	00 <sub>16</sub>	0	0	0	00 <sub>16</sub>
0	1	3D0 <sub>16</sub>	0	1	1	0C <sub>16</sub>	1	1	0	18 <sub>16</sub>
1	0	3E0 <sub>16</sub>	0	1	1	0C <sub>16</sub>	1	1	0	18 <sub>16</sub>
1	1	3F0 <sub>16</sub>	1	0	1	14 <sub>16</sub>	1	0	1	14 <sub>16</sub>

## SOLUCION TIPO - A

:	0	1	0	3	C	0	0	0	0	0	3	C
:	0	1	0	3	D	0	0	0	0	C	2	0
:	0	1	0	3	E	0	0	0	0	C	1	0
:	0	1	0	3	F	0	0	0	1	4	F	8
:	0	0	0	0	0	0	0	1	F	F		

## SOLUCION TIPO - B

:	0	1	0	3	C	0	0	0	0	0	3	C
:	0	1	0	3	D	0	0	0	1	8	1	4
:	0	1	0	3	E	0	0	0	1	8	0	4
:	0	1	0	3	F	0	0	0	1	4	F	8
:	0	0	0	0	0	0	0	1	F	F		

Nota: El formato HEX ya fue tratado en una Práctica de Laboratorio que decía:

"Un archivo HEX típico tiene el siguiente aspecto:

:10008000AF5F67F0602703E0322CFA92007780C3FD

:00000001FF

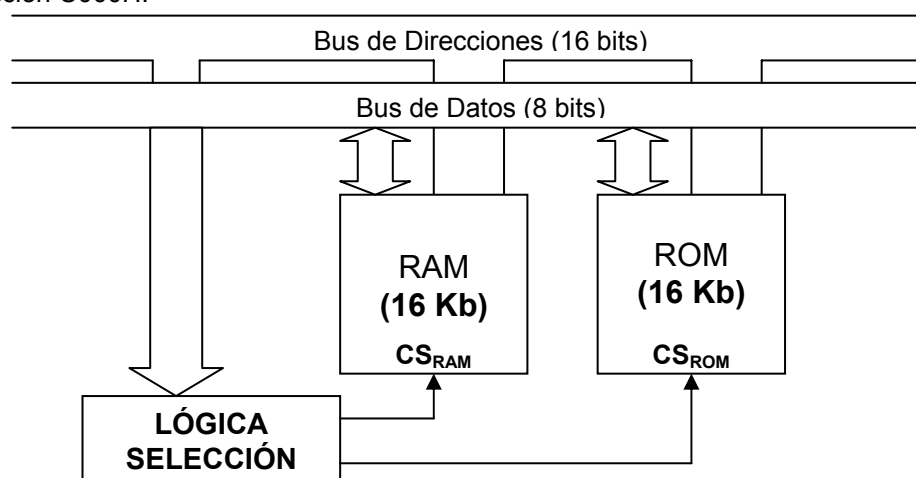
Los datos se dividen en campos (records), cada uno en una línea. Tomando como ejemplo la primera línea, el formato es:

El primer carácter (:) indica el comienzo de un campo. Los dos caracteres siguientes indican el número de bytes de datos del campo. A continuación, los cuatro caracteres siguientes indican la posición de la EPROM donde se cargarán los datos del campo. Los dos caracteres siguientes indican el tipo de campo, 00 si se trata de datos. A continuación vienen los datos; el primer byte se grabará en la posición de carga que se especificó anteriormente, el segundo en la siguiente posición, y así sucesivamente... Por último, los dos últimos caracteres son el checksum. Este checksum tiene que calcularse de tal manera que la suma de todos los bytes del campo más el propio checksum sea 0x00 (00 en hexadecimal). Sólo se considera el byte menos significativo de la suma. El último campo (la última línea) siempre tiene la misma forma, e indica el fin de archivo (tipo de campo 01)"

**Problema 2.-** Se desea diseñar y construir un sistema digital basado en un microprocesador (CPU) que dispone de un bus de direcciones de 16 bits y un bus de datos de 8 bits (1 byte). El sistema dispone de 2 módulos o bancos de memoria uno para los datos (RAM) y otro para el código (ROM). Ambas memorias son de 16 Kb de tamaño, es decir, 16384 posiciones de 8 bits (byte). Se pide diseñar el circuito de la **lógica de selección** de las memorias para este sistema sabiendo que:

**TIPO - A:** la memoria RAM debe estar situada a partir de la dirección 8000H y la memoria ROM a partir de la dirección 0000H.

**TIPO - B:** la memoria RAM debe estar situada a partir de la dirección 4000H y la memoria ROM a partir de la dirección C000H.



**Diagrama de bloques simplificado de la conexión entre las memorias, la lógica de selección y los buses del sistema**

Rellenar la siguiente tabla con las **direcciones finales** (en hexadecimal y en binario), para cada una de las memorias del sistema y deducir las ecuaciones lógicas para cada uno de sus *chip select* (CS), expresadas como minterms. Recuerde que las ecuaciones lógicas son funciones de algunos bits del bus de direcciones del sistema.

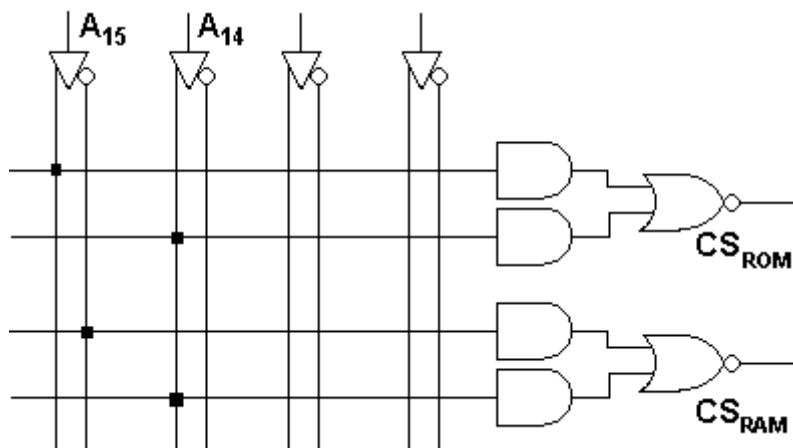
#### SOLUCION TIPO - A

DIRECCIONES	A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub>	A <sub>11</sub> A <sub>10</sub> A <sub>9</sub> A <sub>8</sub>	A <sub>7</sub> A <sub>6</sub> A <sub>5</sub> A <sub>4</sub>	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	MEMORIA
0000 <sub>16</sub>	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	ROM
3FFF <sub>16</sub>	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1	
XXXX	XXXXX	XXXXX	XXXXX	XXXXX	...
8000 <sub>16</sub>	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	RAM
BFFF <sub>16</sub>	1 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1	

$$CS_{ROM} = \overline{A_{15}} \cdot \overline{A_{14}} = \overline{(A_{15} + A_{14})}$$

$$CS_{RAM} = A_{15} \cdot \overline{A_{14}} = \overline{(\overline{A_{15}} + A_{14})}$$

Una vez obtenidas las ecuaciones para cada uno de los CS de las memorias, implemente el circuito de la lógica de selección del sistema que deseamos construir utilizando la siguiente PAL.



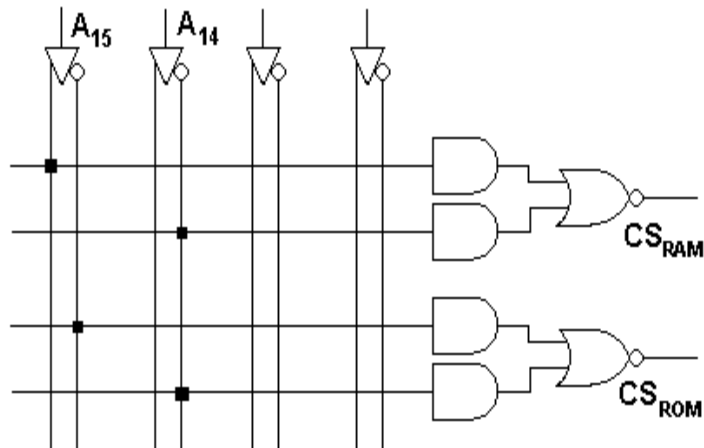
### SOLUCION TIPO - B

DIRECCIONES	A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub>	A <sub>11</sub> A <sub>10</sub> A <sub>9</sub> A <sub>8</sub>	A <sub>7</sub> A <sub>6</sub> A <sub>5</sub> A <sub>4</sub>	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	MEMORIA
4000 <sub>16</sub>	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	RAM
7FFF <sub>16</sub>	0 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	
XXXX	XXXXX	XXXXX	XXXXX	XXXXX	...
C000 <sub>16</sub>	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	ROM
FFFF <sub>16</sub>	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	

$$CS_{RAM} = \neg A_{15} \cdot A_{14} = \neg (A_{15} + \neg A_{14})$$

$$CS_{ROM} = A_{15} \cdot A_{14} = \neg (\neg A_{15} + \neg A_{14})$$

Una vez obtenidas las ecuaciones para cada uno de los CS de las memorias, implemente el circuito de la lógica de selección del sistema que deseamos construir utilizando la siguiente PAL.



**Problema 3:** Un circuito secuencial tiene una entrada X de datos serie sincronizados con una señal de reloj CLK. La salida Z del circuito debe ser 1 cuando se detecte una secuencia que incluya:

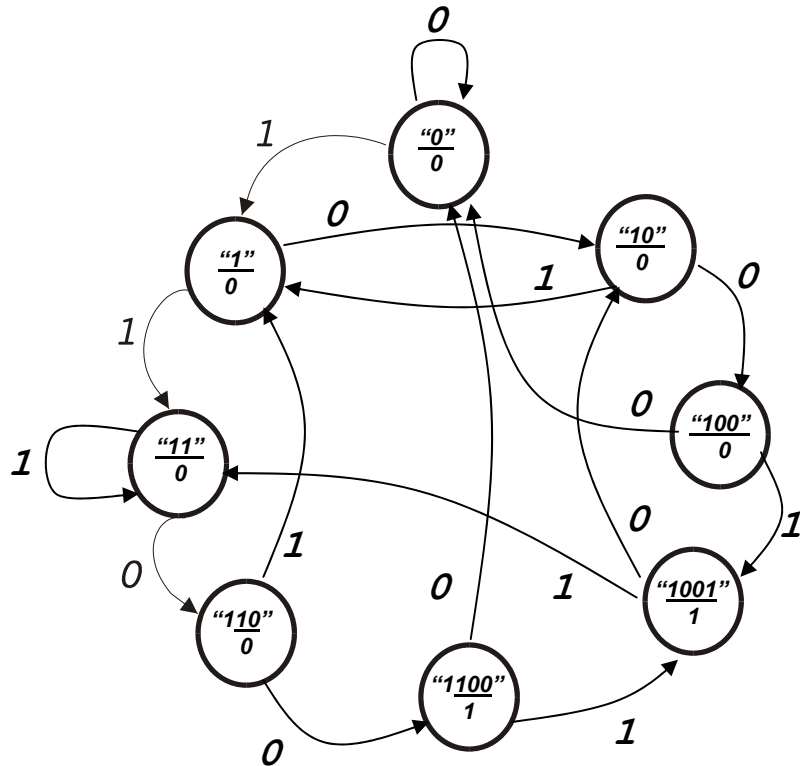
**TIPO – A:** 1100 o 1001 con solapamiento.

**TIPO – B:** 1100 o 1011 con solapamiento.

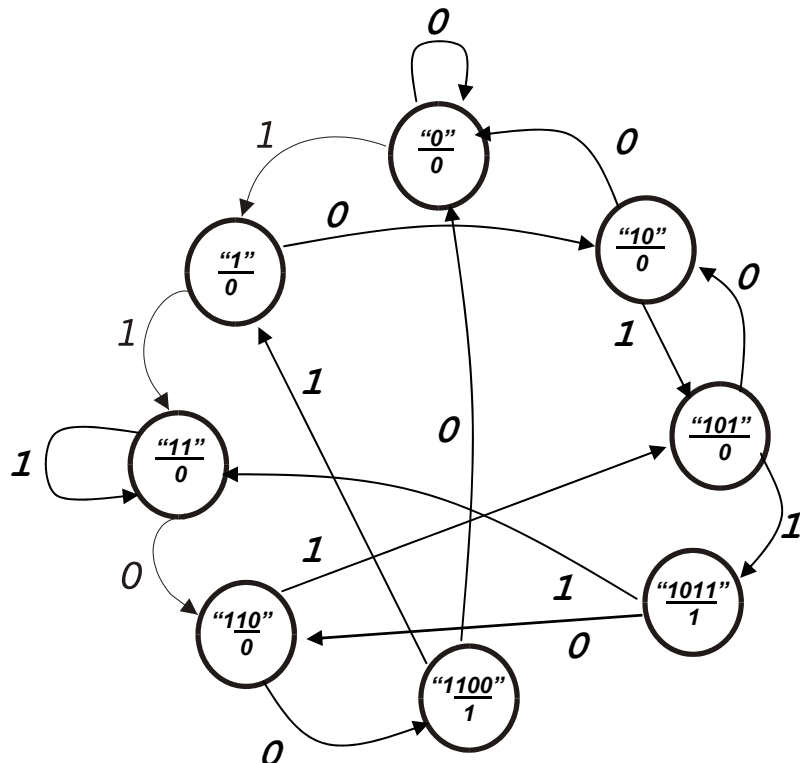
Considerando un *reset* asíncrono inicial que conduce al estado S0, dibuje el diagrama de estados Moore del sistema a partir de la solución parcial de la figura de abajo. Deje en blanco todos lo que no necesite. Se valorará positivamente que la solución esté completa y utilice el mínimo número de estados.

Entre las comillas se indica la secuencia válida de bits que caracteriza a cada estado.

### SOLUCION TIPO – A.-



### SOLUCION TIPO – B



**Problema 4:** A partir de la siguiente tabla de estados de una máquina Mealy, halle el valor de la función de salida Z. Utilice para simplificar el mapa K adjunto. La máquina tiene tres entradas  $A_2 A_1 A_0$  y una única salida.

**TIPO A**

	$Q^{n+1}_1 Q^{n+1}_0 / Z$					
$Q^n_1 Q^n_0$	$A_2 A_1 A_0 = 000$	$A_2 A_1 A_0 = 001$	$A_2 A_1 A_0 = 010$	$A_2 A_1 A_0 = 011$	$A_2 A_1 A_0 = 101$	$A_2 A_1 A_0 = 111$
00	00/0	00/1	01/0	00/0	00/0	00/1
01	01/0	01/1	00/0	01/0	00/0	01/1
10	00/0	10/1	01/0	00/1	00/0	01/0
11	11/1	11/0	11/0	11/1	11/0	11/0

**SOLUCION TIPO - A**

$Q_1 Q_0$	$A_1 A_0$			
	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	0	0	0	0

$A_2 = 0$

$Q_1 Q_0$	$A_1 A_0$			
	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	0	0	1	1

$A_2 = 1$

$$Z = \underline{A_2 A_0 / Q_1} + \underline{A_1 A_0 / Q_1} + \underline{A_2 A_0 / Q_0} + \underline{A_2 / A_0 Q_1} + \underline{A_1 / A_0 Q_1 Q_0}$$

**TIPO B**

	$Q^{n+1}_1 Q^{n+1}_0 / Z$					
$Q^n_1 Q^n_0$	$A_2 A_1 A_0 = 000$	$A_2 A_1 A_0 = 001$	$A_2 A_1 A_0 = 010$	$A_2 A_1 A_0 = 100$	$A_2 A_1 A_0 = 101$	$A_2 A_1 A_0 = 111$
00	00/1	00/0	01/1	00/1	00/1	00/0
01	01/1	01/0	00/1	01/1	00/1	01/0
10	00/1	10/0	01/1	00/0	00/1	01/1
11	11/0	11/1	11/1	11/0	11/1	11/1

**SOLUCION TIPO - B**

$Q_1 Q_0$	$A_1 A_0$			
	00	01	11	10
00	1	1	0	1
01	0	0	1	0
11	0	0	1	0
10	1	1	1	1

$A_2 = 0$

$Q_1 Q_0$	$A_1 A_0$			
	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	1	0	0

$A_2 = 1$

$$Z = \underline{A_0 / Q_1} + \underline{A_0 Q_1 Q_0} + \underline{A_2 / A_1 / Q_1} + \underline{A_2 A_0 Q_1} + \underline{A_2 A_1 / A_0} + \underline{A_2 / A_0 / Q_0}$$

**Problema 5.-** Dado el siguiente programa escrito para el microprocesador visto en la clase de teoría y programado en prácticas, se pide indicar el contenido del registro acumulador y de las posiciones de memoria señaladas al final del programa. Indicar la solución en hexadecimal. En la tabla adjunta se indica el contenido inicial de los registros internos y de algunas posiciones de memoria que intervienen en el programa. Justificar brevemente la respuesta.

**TIPO - A**

Reg. /Mem.	PC	SP	Mem(40 <sub>16</sub> )	Mem(41 <sub>16</sub> )	Mem(50 <sub>16</sub> )	Mem(51 <sub>16</sub> )	Mem(FF <sub>16</sub> )
Contenido	00 <sub>16</sub>	00 <sub>16</sub>	050 <sub>16</sub>	000 <sub>16</sub>	0A0 <sub>16</sub>	FF6 <sub>16</sub>	000 <sub>16</sub>

DIRECCION	CODIGO	Función que se realiza
00	CRA	;Pone a cero el acumulador.
01	ADI 0x40	;Suma el acumulador con dato en memoria. Direcc. indirecto.
02	STA 0x41	;Guarda en memoria el contenido del acumulador.
03	ITA	;Incrementa el valor del acumulador.
04	CAL 0x0A	;Llamada a una subrutina en la posición 0A <sub>16</sub> .
05	ISZ 0x51	;Salta una instrucción si el contador asociado llega a cero.
06	JMP 0x03	;Salta a una posición de programa.
07	HLT	; Fin del programa.
.....	.....	.....
0A	ISZ 0x50	;Salta una instrucción si el contador asociado llega a cero.
0B	RET	;Retorna de la subrutina.

SOLUCION						
Reg. /Mem.	Acumulador	Mem(40 <sub>16</sub> )	Mem(41 <sub>16</sub> )	Mem(50 <sub>16</sub> )	Mem(51 <sub>16</sub> )	Mem(FF <sub>16</sub> )
Contenido	0AA	050	0A0	0AA	000	005

El ejercicio de **TIPO- B** es igual cambiando los valores de las direcciones implicadas en el código.

**Problema 6.-** Un circuito secuencial recibe tres números A, B y C en serie de longitud indeterminada, sincronizados con una señal de reloj. Los bits entran al circuito comenzando por el LSB. Se pide diseñar una máquina de estados finita por Mealy (FSM) cuya salida Z sea el valor de la suma en serie.

**Nota:** La suma completa se obtendrá dos ciclos de reloj más tarde de la llegada del MSB. Por ejemplo:

