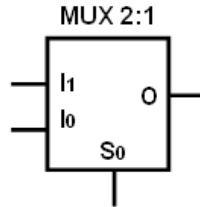


SOLUCIONES

P1.-

1.1. Utilizando **exclusivamente** multiplexores 2:1 como el mostrado en la figura, implementar las siguientes funciones:

- a. $\neg A$ (A negada)
- b. $A \cdot B$
- c. $A + B$
- d. $A \oplus B$
- e. $A \oplus B$

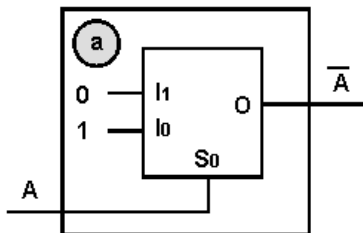
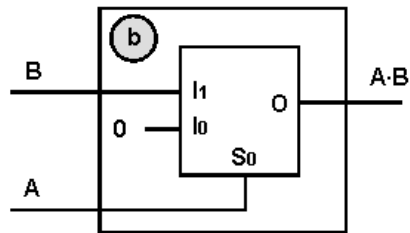
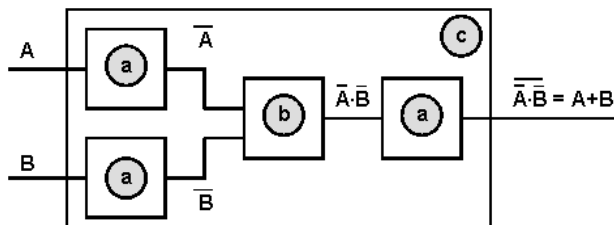
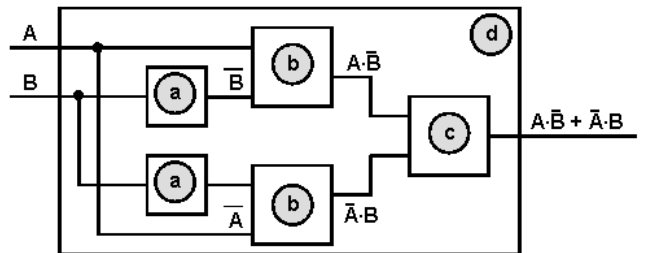
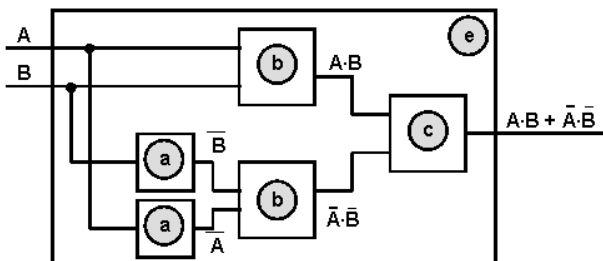


Indicar siempre el valor de cada una de las entradas de todos los multiplexores (tener precaución con el peso de las entradas).

1.2. Implementar un circuito sumador de 2 números A_1A_0 y B_1B_0 , de dos bits cada uno, cuya suma se represente en tres bits $S_2S_1S_0$. Utilizar exclusivamente 3 multiplexores 4:1 (uno para cada una de las salidas) y el mínimo número de los circuitos del punto anterior que sean necesarios, como "cajas negras". NOTA: No se dispone de las entradas negadas.

SOLUCION

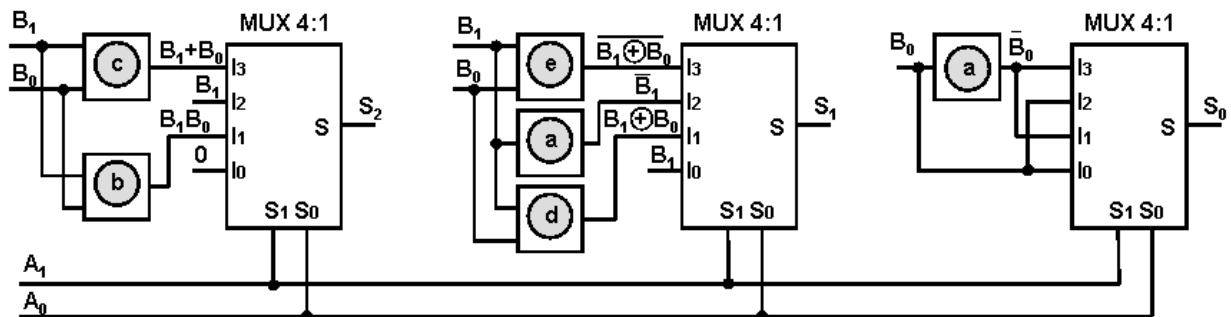
1.a. A negada

1.b. $A \cdot B$ 1.c. $A + B = \neg(\neg(A + B)) = \neg(\neg A \cdot \neg B)$ 1.d. $A \oplus B = A/B + \neg A/B$ 1.e. $A \oplus B = A/B + \neg A/B$ 

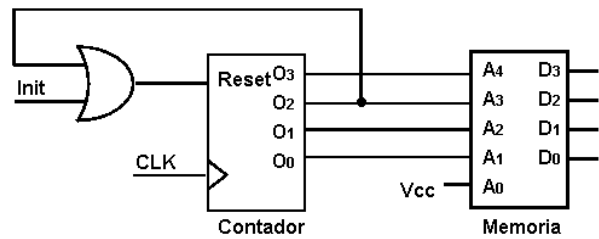
1.2 La tabla de verdad del sumador pedido es:

A_1	A_0	B_1	B_0	S_2	S_1	S_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Si se utilizan los bits A_1A_0 como los bits de control de los multiplexores y los bits B_1B_0 como entradas de los mismos, la solución queda como sigue:



P2.- El circuito de la figura está compuesto por un contador ascendente de 4 bits conectado a una memoria, cuyo contenido se adjunta en la tabla inferior. El contador dispone de una entrada asíncrona de Reset, activa a nivel alto, que al ser activada lo pone a cero tras 4 ms.



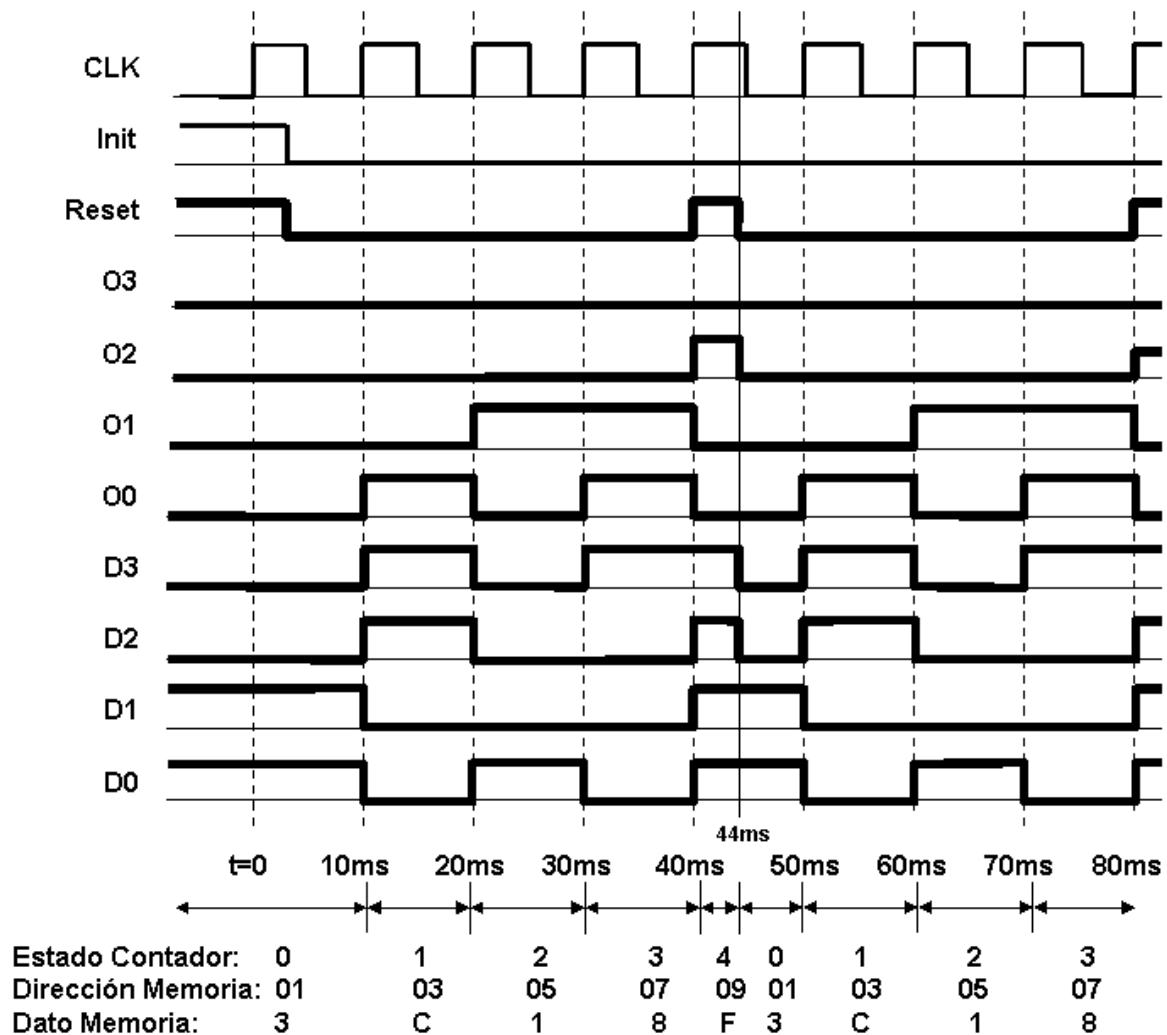
Direcc.	Dato	Direcc.	Dato	Direcc.	Dato	Direcc.	Dato
00	B	08	0	10	B	18	A
01	3	09	F	11	E	19	0
02	B	0A	F	12	B	1A	3
03	C	0B	5	13	E	1B	B
04	C	0C	6	14	C	1C	C
05	1	0D	B	15	A	1D	D
06	F	0E	A	16	F	1E	E
07	8	0F	E	17	E	1F	F

Contenido de la memoria, direcciones y datos están en hexadecimal.

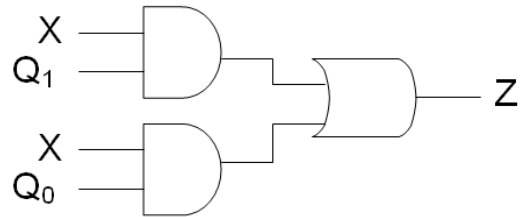
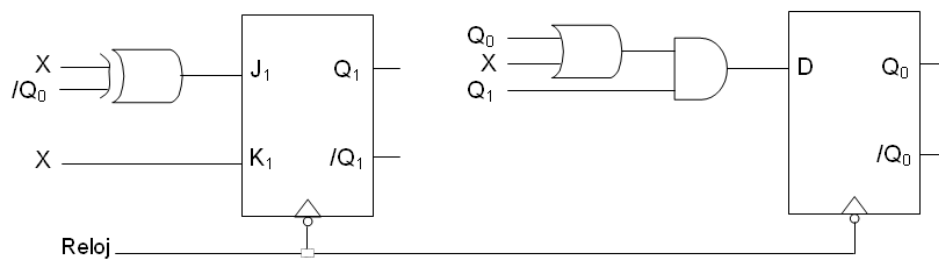
Se pide, para la situación de las entradas mostradas en el cronograma siguiente, completarlo hasta el instante "80 ms" del reloj para las salidas mostradas. Suponer que no existe ningún retardo, excepto el mencionado de la señal de Reset.

SOLUCION

El cronograma pedido es el siguiente:



P3.- Dado el siguiente circuito secuencial:



Se pide determinar las ecuaciones del estado siguiente y de la salida Z

SOLUCION:

$$Q_0^{n+1} = Q_1 (Q_0 + X)$$

$$Q_1^{n+1} = / Q_1 (Q_0 \oplus X) + Q_1 / X$$

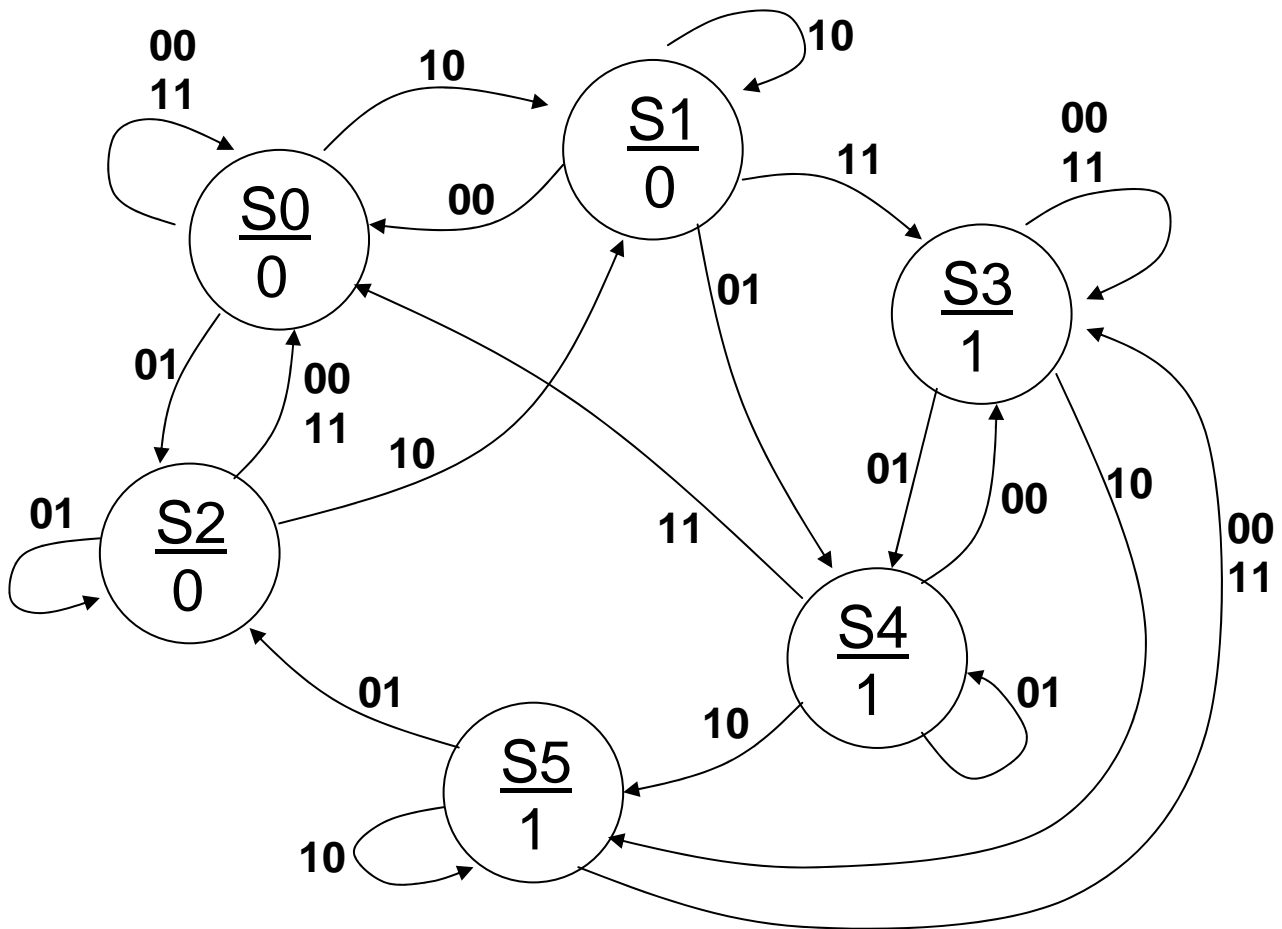
$$Z = (Q_1 + Q_0) X$$

P4.- Un circuito secuencial tiene dos entradas (X_2, X_1) y una salida (Z). La salida permanece a nivel constante (sin variar) hasta que se recibe alguna de las siguientes secuencias de entrada:

- La secuencia de entrada $X_2X_1 = 01 \rightarrow 11$ hace que la salida pase a 0
- La secuencia de entrada $X_2X_1 = 10 \rightarrow 11$ hace que la salida pase a 1
- La secuencia de entrada $X_2X_1 = 10 \rightarrow 01$ hace que la salida cambie de valor

(La notación $X_2X_1 = 01 \rightarrow 11$ significa que $X_2 = 0, X_1 = 1$ y al ciclo siguiente $X_2 = 1, X_1 = 1$)

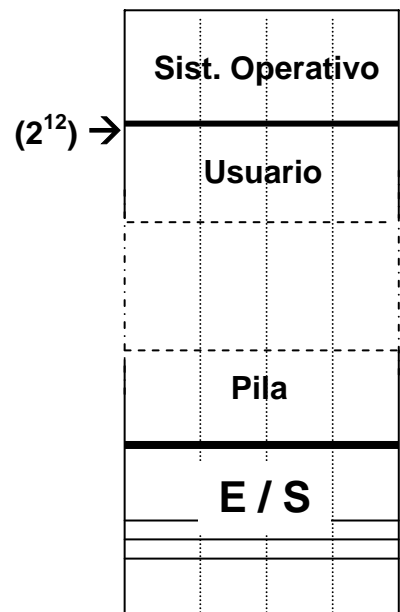
Se pide construir un diagrama de estados de Moore para el circuito, utilizando los estados necesarios del diagrama adjunto, añadiendo más si fuera necesario o dejando en blanco los que no se necesiten.



P5.- El mapa de memoria de ARC (ver figura) utiliza direcciones de 32 bits). Se sabe que esta implementado en tres zonas, utilizando tres pastillas (chips) diferentes. En la figura se indica la primera dirección, en decimal, de la segunda zona. Se pide, justificando brevemente las respuestas:

a) Completar la tabla adjunta, indicando las correspondientes direcciones en hexadecimal.

b) Encontrar las ecuaciones lógicas que sirven para seleccionar las tres memorias definidas en la tabla.



SOLUCION

Tamaño RAM S.O.: 4 kbytes

Tamaño I/O: 128 Mbytes = 131.072 kbytes

Tamaño RAM USUARIO: TOTAL – RAM SO – I/O = 4 Gbytes – 4 kbytes – 128 Mbytes

Pastilla	Selector	Tamaño	Dirección inicial	Dirección final
RAM S.O.	CS_1	4 kbytes	00000000_{16}	$0000FFF_{16}$
RAM USUARIO	CS_2	4.063.228 kbytes	00001000_{16}	$F7FFFFFF_{16}$
I/O	CS_3	128 Mbytes	$F8000000_{16}$	$FFFFFFFF_{16}$

$$CS_1 = /A_{31}/A_{30}/A_{29}/A_{28}/A_{27}/A_{26}/A_{25}/A_{24}/A_{23}/A_{22}/A_{21}/A_{20}/A_{19}/A_{18}/A_{17}/A_{16}/A_{15}/A_{14}/A_{13}/A_{12}$$

$$CS_3 = A_{31}A_{30}A_{29}A_{28}A_{27}$$

$$CS_2 = / (CS_1 + CS_3) = /CS_1 \cdot /CS_3$$

P6.- Se añade al conjunto de instrucciones de ARC el desplazamiento lógico a la izquierda (**sll**) equivalente a la estudiada **srl**, pero el desplazamiento es en el sentido contrario. Se pide completar el programa dado para ARC, que calcula el producto de dos números $P = M * N$, donde N se puede escribir como $N = 2^X - 2^Y$.

NOTA: Es necesario realizar un breve comentario de cada una de las instrucciones que se escriban.

SOLUCION		
.begin		
.org 1024		
ld	[M], %r1	!Lee M en %r1
ld	[X], %r2	!Lee X en %r2
ld	[Y], %r3	!Lee Y en %r3
orncc	%r1,%r0,%r4	!calcula C1 de M en %r4
addcc	%r4,1,%r4	!calcula C2 de M en %r4
sll	%r1,%r2,%r1	! $M * 2^X$
sll	%r4,%r3,%r4	! $- M * 2^Y$
addcc	%r1,%r4,%r1	! $M * (2^X - 2^Y)$
st	%r1,%r0, [P]	!escribe el resultado
fin: ba fin		
! Memoria de datos		
M:		
X:		
Y:		
P:		
.end		