

Apellidos:.....**SOLUCIÓN**.....

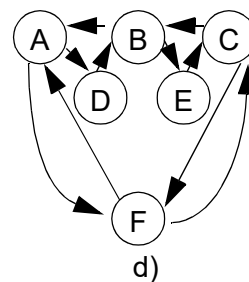
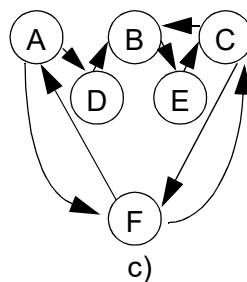
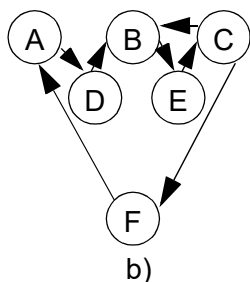
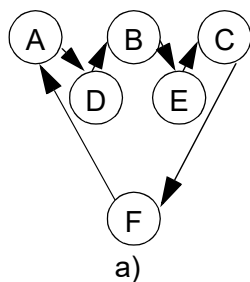
Nombre:.....

T1 T2 T3 T4

--	--	--	--

TEORÍA (Cada pregunta vale 1 punto)

- 1.- Defina las características temporales de los circuitos digitales (tanto combinacionales como secuenciales).
- 2.- Discuta razonadamente la selección de las características del contador (módulo, señales de entrada/salida, operaciones...) de menor coste con el que se puedan implementar los siguientes diagramas de estado, así como la asignación de estados de cuenta propuestos para cada estado de la FSM y la operación a realizar:



- 3.- Defina de forma detallada las operaciones y características de una memoria RAM, así como otras RAM. Ponga ejemplos de cómo realizar ampliaciones de la anchura de palabra y aumento del número de palabras de una memoria RAM.

SOLUCIÓN**CRITERIO DE CORRECCIÓN**

- 4.- Describa de forma detallada:
 - a) Las ventajas e inconvenientes de las dos arquitecturas de referencia para computadores.
 - b) Las dos alternativas de modelos de conjunto de instrucciones.

SOLUCIÓN**CRITERIO DE CORRECCIÓN**

Apellidos:.....**SOLUCIÓN**.....

P1 P2

Nombre:.....

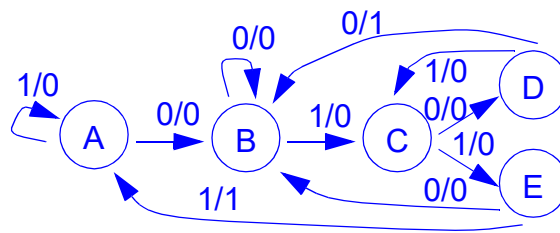
P1	P2

PROBLEMAS (Cada pregunta vale 3 puntos)

- 1.- Diseñe un detector de secuencias que active su salida z cuando por su entrada x los últimos bits sean 0100 y 0111. Determine las características e implemente un circuito óptimo usando un contador ascendente módulo 8 y puertas.

SOLUCIÓN

a) Diagrama de estados:



A: Secuencia vacía {}.

B: Recibido primer bit de la secuencia {0}.

C: Recibidos 2 bits de la secuencia {0, 1}.

D: Recibidos 3 bits de la primera secuencia {0, 1, 0}.

E: Recibidos 3 bits de la segunda secuencia {0, 1, 1}.

b) Características del contador:

- Son 5 estados, luego se puede hacer con un CONT módulo 8.
- El estado que recibe más flechas debería ser el estado 0. En este caso, el B.
- Los estados A y B tienen autotransiciones. Se necesita entrada de inhabilitación de cuenta o carga en paralelo (para el B también puesta a cero).
- Del estado D se vuelve al C. Se requiere decrementar o carga en paralelo.
- Del estado C se pasa al D y al E. Sólo se puede hacer con una carga en paralelo.
- Del estado E se va al B (reset) y al A. Si se asigna a A el estado siguiente al E, se hace con incremento en lugar de carga en paralelo.

La carga en paralelo es necesaria, por lo que el resto de operaciones podrían eliminarse. Por ejemplo, la inhabilitación sólo se requiere una vez (estado A). Sería necesario estudiar ambas opciones para ver si es mejor incluir esa operación o ver cómo de complejas resultan las ecuaciones de LD# y los datos asociados. En ese caso, se prescinde de INH.

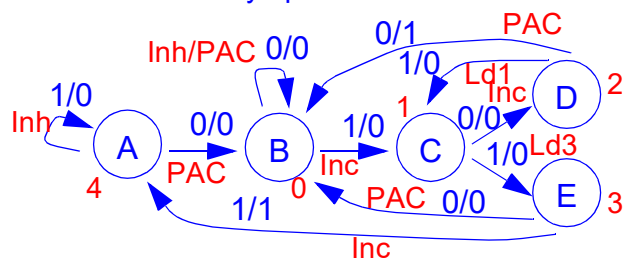
El decremento tiene una situación análoga (estado D), así que se prescinde de la operación.

Las operaciones de puesta a cero y de incremento son susceptibles de la misma discusión, pero al tratarse de más casos y al introducir más inespecificaciones en la tabla de verdad, optamos por dejarlas y darles prioridad frente a la carga en paralelo.

La tabla de funcionamiento del contador quedaría:

CL# LD#	CONT←	q _{2:0} ←
0- 10 11	0 d _{2:0} CONT+1	[CONT]

c) Asignación de estados y operaciones:



d) Tabla de verdad del CC:

S	q _{2:0} X	z	CL#	LD#	d _{2:0}
B	0000	0	0	-	---
	0001	0	1	1	---
C	0010	0	1	1	---
	0011	0	1	0	011
D	0100	1	0	-	---
	0101	0	1	0	001
E	0110	0	0	-	---
	0111	1	1	1	---
A	1000	0	0	-	---
	1001	0	1	0	100
-	1010	-	-	-	---
	1011	-	-	-	---
-	1100	-	-	-	---
	1101	-	-	-	---
-	1110	-	-	-	---
	1111	-	-	-	---

e) Mapas de Karnaugh y ecuaciones:

$q_2:1$ q_0x	00	01	11	10
00	0	1	-	0
01	0	0	-	0
11	0	1	-	-
10	0	0	-	-

z

$$z = q_1 \bar{q}_0 \bar{x} + q_1 q_0 x = q_1 (\bar{q}_0 \bar{x} + q_0 x) \\ = q_1 (\bar{q}_0 \oplus x)$$

$q_2:1$ q_0x	00	01	11	10
00	0	0	-	0
01	1	1	-	1
11	1	1	-	-
10	1	0	-	-

CL#

$$CL\# = x + \bar{q}_1 q_0$$

$q_2:1$ q_0x	00	01	11	10
00	-	-	-	-
01	1	0	-	0
11	0	1	-	-
10	1	-	-	-

LD#

$$LD\# = \bar{q}_2 \bar{q}_1 \bar{q}_0 + q_1 q_0 + q_0 \bar{x}$$

$q_2:1$ q_0x	00	01	11	10
00	-	-	-	-
01	-	0	-	1
11	0	-	-	-
10	-	-	-	-

d₂

$$d_2 = q_2$$

$q_2:1$ q_0x	00	01	11	10
00	-	-	-	-
01	-	0	-	0
11	1	-	-	-
10	-	-	-	-

d₁

$$d_1 = q_0$$

$q_2:1$ q_0x	00	01	11	10
00	-	-	-	-
01	-	1	-	0
11	1	-	-	-
10	-	-	-	-

d₀

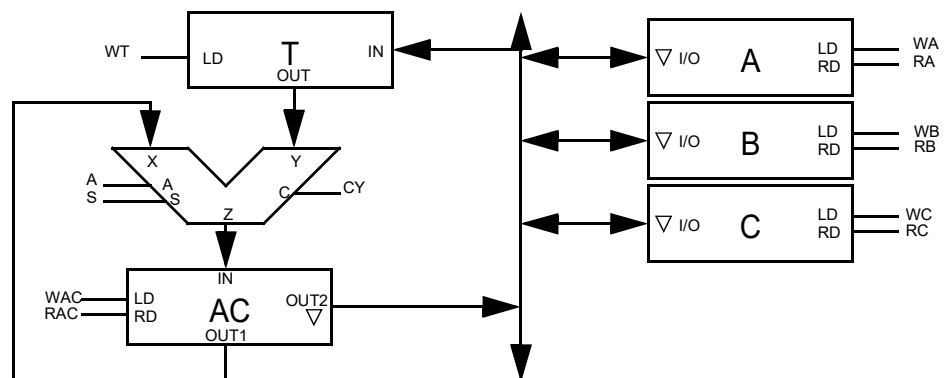
$$d_2 = \bar{q}_2$$

Sólo queda dibujar el circuito.

CRITERIO DE CORRECCIÓN

- a) 30% Diagrama de estados
- b) 20% Elección del contador
- c) 10% Asignación de estados y operaciones
- d) 10% Tabla de verdad del CC
- e) 30% Mapas K, ecuaciones y circuito

2.- Para la unidad de datos de la figura:

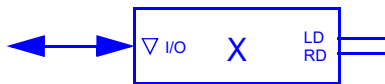


- a) [20%] Describa a nivel RT los componentes de la UD.
- b) [50%] Suponiendo números sin signo, diseñe la carta ASM (unidad de datos y unidad de control) de un circuito que permita realizar la siguiente macrooperación:
 $A \leftarrow \text{mayor}(A, B); B \leftarrow \text{menor}(A, B); C \leftarrow |A-B|$
- c) [30%] Realice la implementación de la UC usando la técnica de un biestable por estado.

SOLUCIÓN

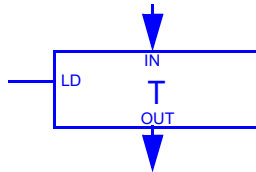
- a) Descripción RT:

Registro X (A, B y C):



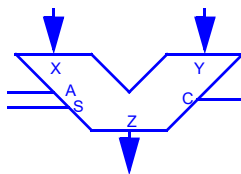
LD RD	$X \leftarrow$	I/O=
00	X	HiZ
01	X	[X]
1-	I/O	I/O

Registro T:



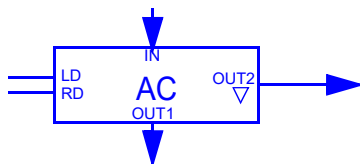
LD	$T \leftarrow$	OUT=
0	T	[T]
1	IN	[T]

ALU:



AS	Z=	C=
1-	X+Y	1 sii desbordamiento en suma
01	X-Y	1 sii desbordamiento en resta
00	0	0

Registro AC:



LD RD	$AC \leftarrow$	OUT1=	OUT2=
00	AC		HiZ
01	AC		[AC]
10	IN	[AC]	HiZ
11	IN		[AC]

- b) Sólo existe una macrooperación que hace 3 cálculos/transferencias. Por tanto, no hay señal de entrada para seleccionar la macrooperación. Para poder hacer los cálculos hay que poner $AC=0$, que se puede realizar cargando el valor actual de AC en T y haciendo una resta de ambos registros. Después el proceso consistirá en calcular $A-B$ y estudiar del carry de la ALU. Si $C=0$, $A \geq B$ y no hay que cambiar los valores de los registros A y B. Además, como el resultado es positivo, se guarda directamente en C. En otro caso, hay que intercambiar A y B y cambiar el signo al resultado antes de guardarlo en C.

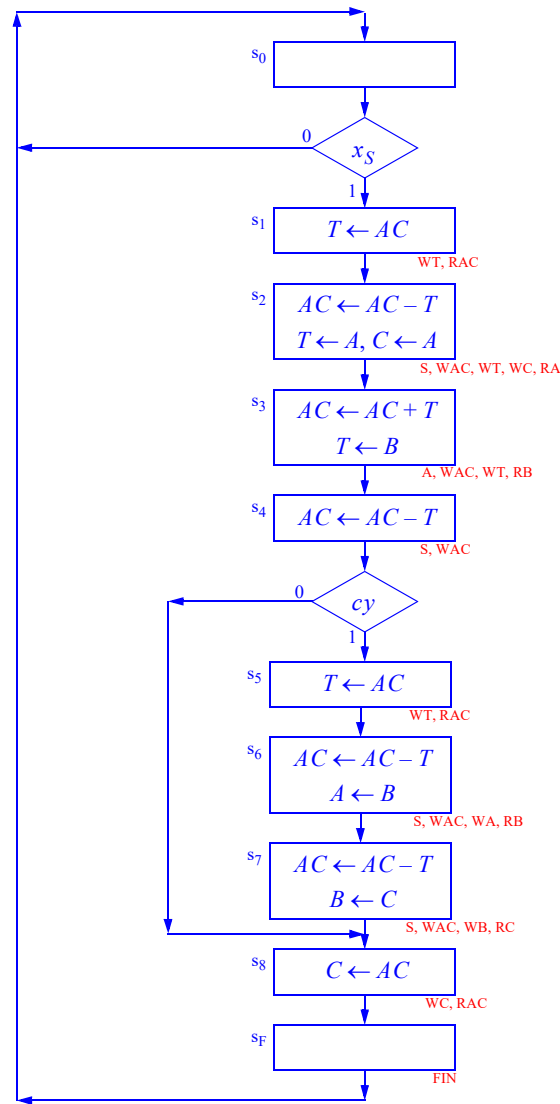
```

T ← AC                ;T=X
AC ← AC-T, T ← A      ;AC=0, T=A
AC ← AC+T, T ← B      ;AC=A, T=B
AC ← AC-T             ;AC=A-B
cy: T ← AC            ;T=A-B
    AC ← AC-T, C ← A  ;AC=0, C=A
    AC ← AC-T, A ← B  ;AC=B-A, A=B (mayor)
    B ← C             ;B=C (menor)
C ← AC                ;C=|A-B|

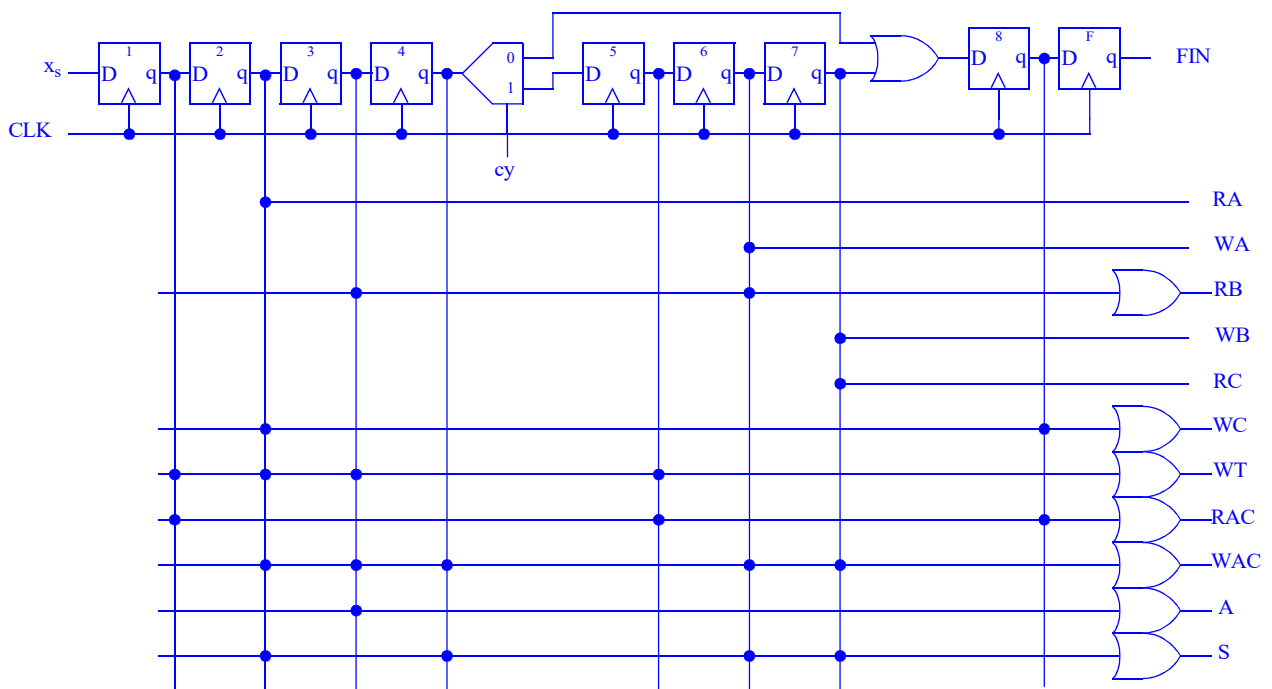
```

La carta ASM se ha optimizado para reducir un estado: aprovechando que se lee A en la segunda microoperación para llevarlo a T, se transfiere también a C (innecesario si $A > B$) para

ir adelantando trabajo en caso de que $A < B$. Con ello, la carta ASM sería (en rojo la versión de la unidad de control)



c) Implementación de la unidad de control con la técnica de un biestable por estado:



Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

T1 T2 T3 T4

--	--	--	--

CRITERIO DE CORRECCIÓN

- a)** 20% Descripción RT
- b)** 50% Cartas ASM
- c)** 30% Implementación UC