

Apellidos:.....**SOLUCIÓN**.....

T.1	T.2	T.3	T.4

Nombre:..... Aula:..... Puesto:.....

**TEORÍA (Cada pregunta vale 1 punto. Entregue TEORÍA y PROBLEMAS por separado)**

1.- Estudie la compatibilidad de las familias ACT y FAST.

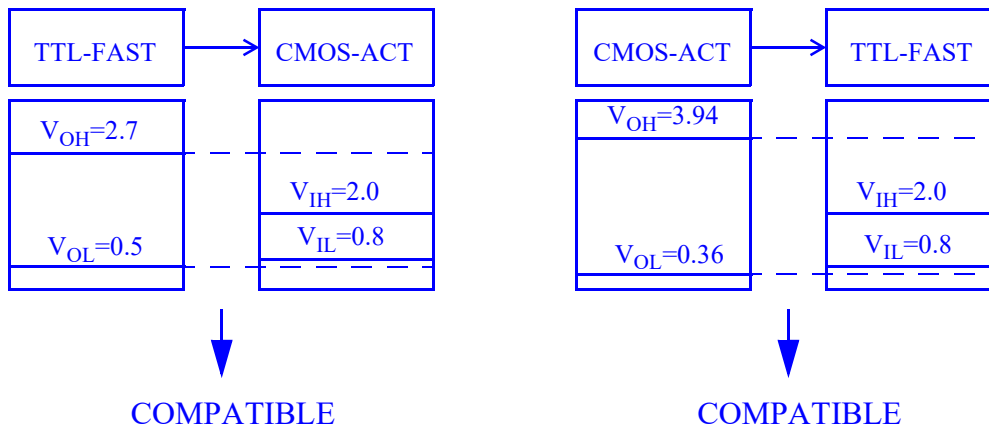
- a) Detalle las condiciones que se deben cumplir para que las familias sean compatibles.
- b) Compruebe si se cumplen en estas dos familias.

	$V_{IHmin}(V)$	$V_{ILmax}(V)$	$V_{OHmin}(V)$	$V_{OLmax}(V)$	$I_{IHmax}(\mu A)$	$I_{ILmax}(\mu A)$	$I_{OHmax}(mA)$	$I_{OLmax}(mA)$
ACT	2	0'8	3,94	0'36	0'1	-0'1	-24	24
FAST	2	0'8	2'7	0'5	20	-600	-1	20

**SOLUCIÓN**

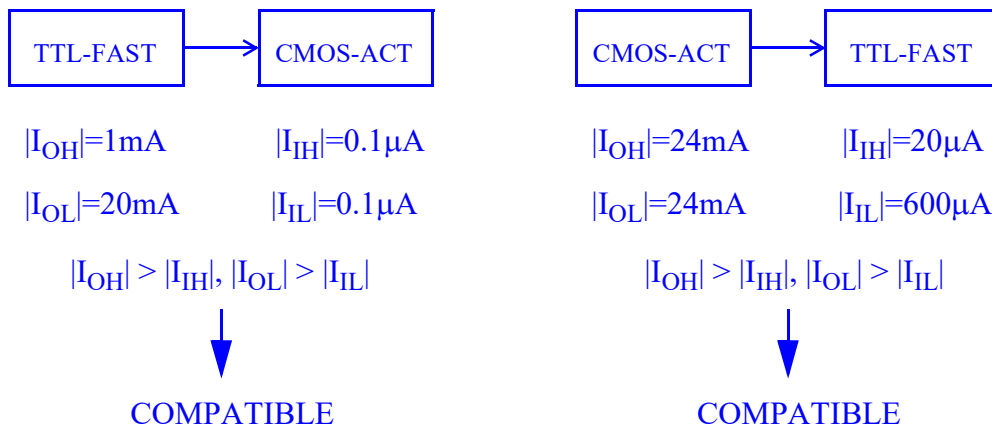
Las condiciones de compatibilidad son  $V_{OH} > V_{IH}$ ,  $V_{OL} < V_{IL}$ ,  $|I_{OH}| > |I_{IH}|$ ,  $|I_{OL}| > |I_{IL}|$ . Y esto para las salidas de una familia con respecto a las entradas de la otra familia y viceversa.

Compatibilidad en Tensión:



En ambos casos  $V_{OH} > V_{IH}$  y  $V_{OL} < V_{IL}$ . Esto significa que los ceros y unos lógicos generados por las puertas de una familia son interpretados correctamente por las de la otra. Son, por tanto, compatibles en tensión.

Compatibilidad en intensidad:



En ambos casos  $|I_{OH}| > |I_{IH}|$  y  $|I_{OL}| > |I_{IL}|$ . Esto significa que las puertas de una familia son capaces de suministrar la intensidad requerida por las entradas de la otra familia.

Apellidos:.....**SOLUCIÓN**.....

P1	P2

Nombre:..... Aula:..... Puesto:.....

Son, por tanto, compatibles también en intensidad. Son familias compatibles.

**CRITERIO DE CORRECCIÓN**

- Cada condición (son 8), 1'25 puntos
- Dar buen resultado, pero no considerar que las intensidades son en valor absoluto, -1

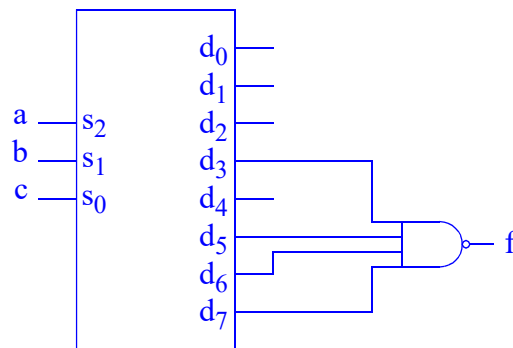
2.- Describa cómo realizar una función combinacional empleando un decodificador con salidas activas en baja y una puerta NAND. Puede apoyarse en un ejemplo.

**SOLUCIÓN**

El proceso de diseño con DEC y puertas requiere que la función se exprese en forma canónica. Dado que se pide hacerlo con puertas NAND, la forma adecuada es un producto de máx términos. Un decodificador con salidas activas en baja pone en sus salidas los máx términos de la función. Sin embargo, como la puerta a usar es una NAND y no una AND, hay que tomar la función complementada.

Se debe elegir un DEC con tantas entradas como variables tenga la función. Es importante que la variables se coloquen en el orden adecuado de peso en las entradas del DEC:

	ab	00	01	11	10
c	0	0 <sup>0</sup>	0 <sup>2</sup>	1 <sup>6</sup>	0 <sup>4</sup>
	1	0 <sup>1</sup>	1 <sup>3</sup>	1 <sup>7</sup>	1 <sup>5</sup>
		f			



$$f(a, b, c) = \prod(0, 1, 2, 4) = M_0M_1M_2M_4$$

$$\overline{f(a, b, c)} = \prod(3, 5, 6, 7) = M_3M_5M_6M_7$$

**CRITERIO DE CORRECCIÓN**

- Poner sólo ejemplo: -6
- Hacer análisis en lugar de procedimiento de síntesis: -7
- Hacer con más de una NOR: -2

3.- Explique las similitudes y diferencias entre una ROM, una PROM, una EPROM y una EEPROM.

**CRITERIO DE CORRECCIÓN**

•

4.- Para un procesador, explique los tipos de direccionamiento de memoria que conozca.

**SOLUCIÓN**

Hay que explicar cómo funcionan los siguientes modos de direccionamiento:

- Directo de registro, inmediato, directo de memoria, indirecto de registro, indirecto de registro con post-incremento, indirecto de registro con pre-decremento, indirecto de registro con offset y relativo a PCo SP.

Apellidos:.....**SOLUCIÓN**.....

Nombre:..... Aula:..... Puesto:.....

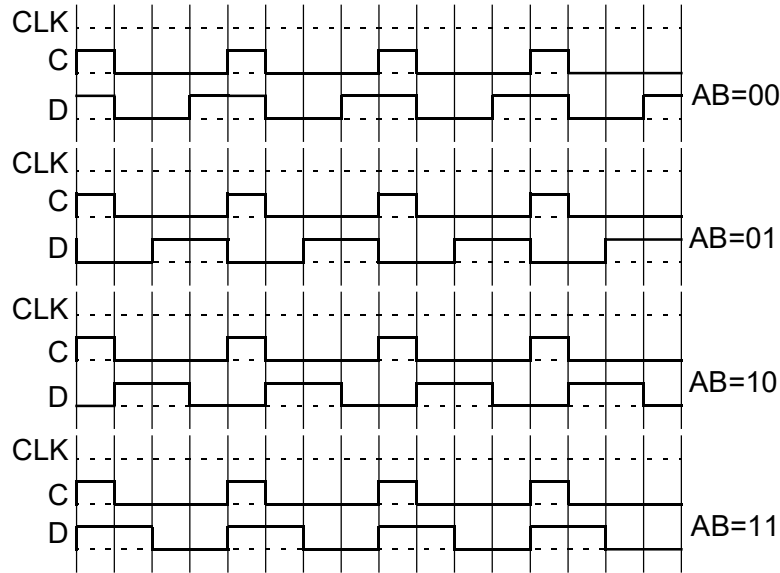
Se puede conseguir la máxima puntuación explicando únicamente 5 de ellos.

**CRITERIO CORRECCIÓN**

2 puntos por cada modo. Si sólo pone el nombre un punto.

**PROBLEMAS (Cada pregunta vale 3 puntos. Entregue TEORÍA y PROBLEMAS por separado)**

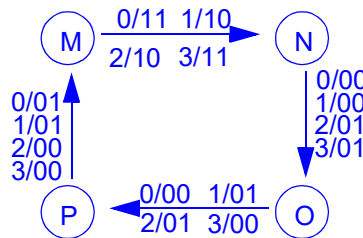
1.- Un circuito secuencial genera las señales C y D en función de las entradas A y B. En el cronograma pueden observarse las formas de ondas generadas si las entradas permanecieran constantes:



- a) Obtenga el diagrama de estados del autómata de Mealy del circuito.
- b) Implementelo usando un contador módulo 8 con el menor coste en operaciones especiales, una ROM 16x1 y subsistemas combinacionales no programables.

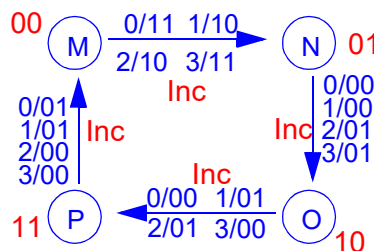
**SOLUCIÓN**

a) En el cronograma se observa que se trata de un generador de secuencias de 2 bits. La secuencia varía en función de las entradas AB, aunque la forma de onda de C siempre es la misma. Dichas secuencias son de 4 bits (o cuatro ciclos), ya que se repite a partir del cuarto ciclo. Eso no lleva al siguiente diagrama de estados:

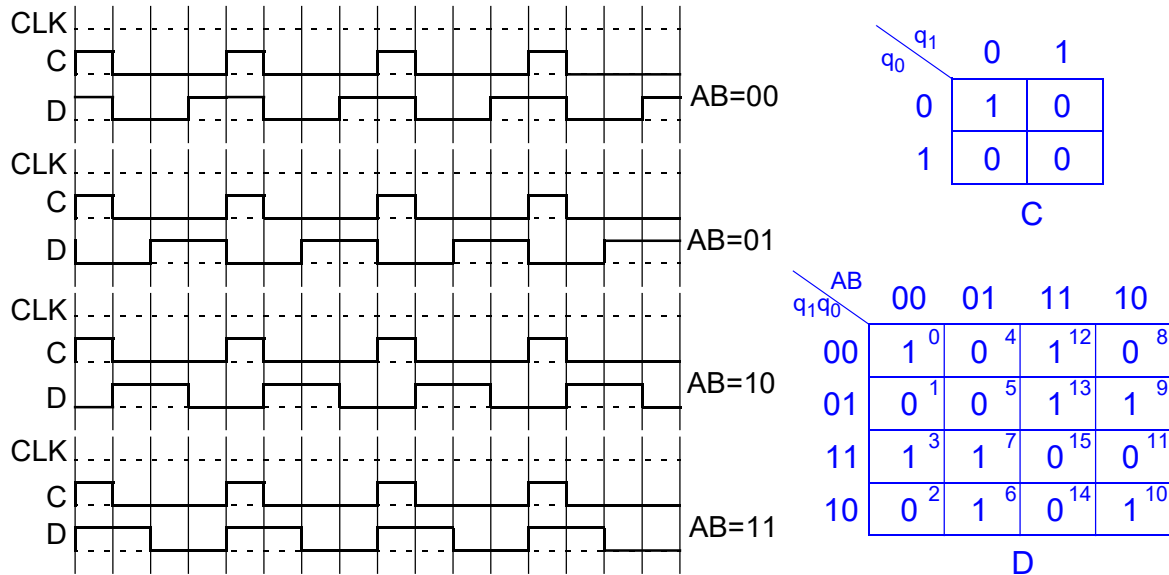


b) Implementación:

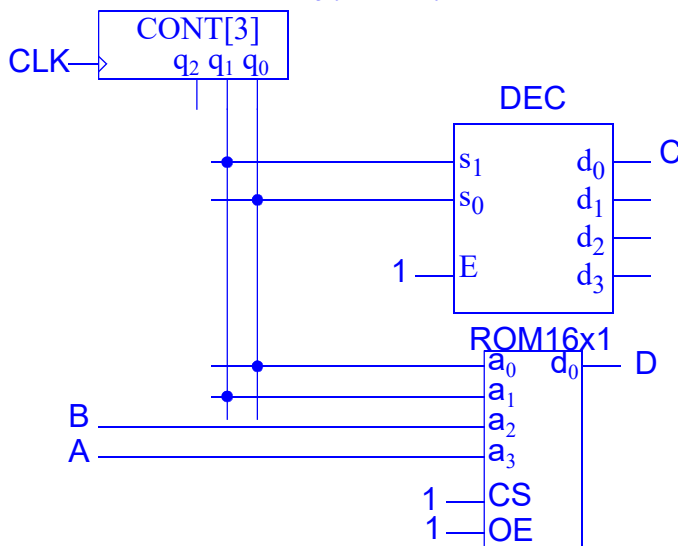
La implementación con contadores es directa. Se necesita un contador módulo 4 sencillo, con capacidad de incremento y nada más:



El contador no necesita ninguna entrada adicional y se prescinde de la salida  $q_2$ , ya que sólo se necesita un contador módulo 4.



La salida C se puede generar fácilmente con un MUX4:1 con las salidas  $q_{1:0}$  conectadas a las entradas de selección y los canales tienen los valores 1,0,0,0. También se podría usar un DEC2:4 conectados a  $q_{1:0}$  y tomar la salida  $d_0$  como salida C. Para generar D se usará la ROM de 16 bits, haciendo  $a_{3:0}=ABq_{1:0}$ .



Contenido de la ROM

$a_{3:0}$	$d_0$
0000	1
0001	0
0010	0
0011	1
0100	0
0101	0
0110	1
0111	1
1000	1
1001	1
1010	0
1011	0
1100	0
1101	1
1110	1
1111	0

**CRITERIO DE CORRECCIÓN**

- a) FSM: 30%
- b) Implementación: 70%

2.- Realice la subrutina *strchr* que busca en una cadena de caracteres S (cuya dirección de inicio está en R0) un carácter C (que está en R1). Devuelve en R2 el número de ocurrencias y en R0 la dirección de la primera (o NULL si no se encontró). NOTA: Una cadena de caracteres es un vector de caracteres codificados en ASCII, que en este caso está almacenado en una memoria, y que termina en el código NUL (valor 0).

Apellidos:.....**SOLUCIÓN**.....

Nombre:..... Aula:..... Puesto:.....

NULL es una dirección de valor 0.

- Explique con palabras el algoritmo que usará para resolver el problema.
- Escriba la subrutina en ensamblador del CS3.

### SOLUCIÓN

- Sea S la cadena apuntada por R0 y C el carácter a buscar. El proceso consiste en recorrer S buscando C o el final de cadena. Si se encuentra NUL, se sale con R0=R2=0. En caso contrario, se guarda en R0 la dirección de la ocurrencia y se pasa a un segundo bucle que cuenta las apariciones de C en S hasta que se encuentra el final de cadena.

b)

```

;-----
; strchr                                     v1.1
; Busca en una cadena de caracteres S el carácter C
; Entrada: R0: Cadena S
;          R1: Carácter C
; Salida:  R0: Posición de la primera ocurrencia o NULL si no se encontró
;          R2: Número de ocurrencias.
;-----
strchr:  ld      r2, (r0)  ;Leer un carácter de la cadena
         addi   r0, 1     ;Avanzar puntero
         cpi   r2, 0     ;Es el final?
         breq  strchrNE ;...sí. Carácter no encontrado
         cp    r2, r1    ;...no. Es el carácter buscado?
         breq  strchrE  ;...sí. Anotar y seguir
         jmp   strchr    ;...no. Siguiente carácter
strchrE: mov    r4, r0    ;Conservar dirección de ocurrencia en R0
         subi  r0, 1     ;Corregir el puntero

         ; Repetir el proceso de búsqueda para contar las ocurrencias
         ldi   r2, 1     ;R2 contador de ocurrencias. Ya hay 1
strchrBuc: ld   r3, (r4)  ;Leer un carácter de la cadena
         addi  r4, 1     ;Avanzar puntero
         cpi  r3, 0     ;Es el final de cadena?
         breq  strchrFin ;...sí. Salir
         cp   r3, r1    ;...no. Es el carácter buscado?
         breq  strchrEnc ;...sí. Contabilizar
         jmp   strchrBuc ;Siguiente carácter
strchrEnc: addi r2, 1     ;Contar currencia
         jmp   strchrBuc ;Siguiente carácter
strbscNE: ldi   r0, 0     ;R0=0 y R2=0 para salir con error
         ldi   r2, 0
strchrFin: ret

```