

**Examen E1 (temas 2 y 3)**

- Duración del examen: 1:15 horas.
- La solución de cada ejercicio se tiene que escribir en el espacio reservado para ello en el propio enunciado.
- No podéis utilizar calculadora, móvil, apuntes, etc.
- La solución del examen se publicará en Atenea mañana y las notas antes del 12 de marzo a la noche.

**Pregunta 1) (1.5 puntos)**

Cada fila de la tabla tiene 3 columnas con: el vector X de 8 bits, X expresado en hexadecimal y el valor en decimal,  $X_u$ , que representa X interpretado como un número natural codificado en binario. Completa todas las casillas vacías.

X	X (hexa)	$X_u$
10011010	9A	154
10000111	87	135
10101001	A9	169

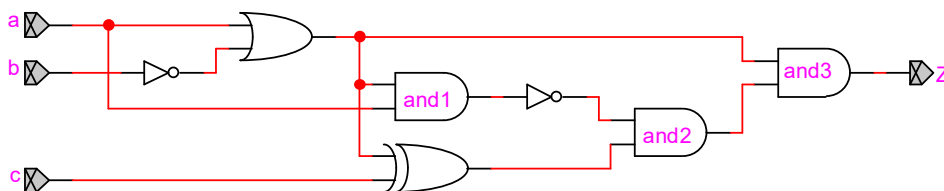
Criterio de corrección: +0.25 puntos por cada casilla correcta.

**Pregunta 2) (2 puntos)**

Dado el esquema del siguiente circuito completad la tabla de verdad de la salida Z y escribid la expresión lógica en suma de minterms.

Tabla de verdad de Z:

a	b	c	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



Expresión en suma de minterms de Z:  **$\neg a \cdot \neg b \cdot \neg c$**

Criterio de corrección: La tabla de verdad vale 1.5 puntos: -0.5 puntos por cada bit incorrecto. La expresión +0.5 puntos si es correcta y la tabla también estaba correcta.

**Pregunta 3) (1.5 puntos)**

Dado el esquema del circuito de la pregunta anterior, escribid el camino crítico (todos si hay varios) y el tiempo de propagación del circuito. Los tiempos de propagación de las puertas son:  $T_{p(\text{Not})} = 10$ ,  $T_{p(\text{And-2})} = 20$ ,  $T_{p(\text{Or-2})} = 20$  y sabemos que el tiempo de propagación de la puerta XOR-2 es el mismo que tendría si internamente estuviese implementada como suma de minterms.

Caminos Críticos =

**$B \rightarrow \text{Not} \rightarrow \text{Or-2} \rightarrow \text{Xor-2} \rightarrow \text{and2} \rightarrow \text{and3} \rightarrow Z$**

Criterio de corrección: 1 punto si el camino crítico es correcto. 0 puntos si hay algún camino NO crítico.

$T_p =$

**120 ut**

Criterio de corrección: Tiempo de ciclo 0.5 puntos (sólo si el camino crítico es correcto)

#### Pregunta 4) (1.5 puntos)

Dibujad el mapa de Karnaugh con las agrupaciones de unos adecuadas para obtener la expresión mínima en suma de productos de la función w de un circuito al que le correspondería la siguiente tabla de verdad.

a	b	c	d	w
0	0	0	0	1
0	0	0	1	x
0	0	1	0	1
0	0	1	1	x
0	1	0	0	1
0	1	0	1	x
0	1	1	0	1
0	1	1	1	1
1	0	0	0	x
1	0	0	1	0
1	0	1	0	1
1	0	1	1	x
1	1	0	0	0
1	1	0	1	x
1	1	1	0	1
1	1	1	1	0

a) Dibuja el Mapa de Karnaugh donde se vea **claramente** los grupos que has escogido

		c d			
		00	01	11	10
a b	00	1	x	x	1
	01	1	x	1	1
	11	0	x	0	1
	10	x	0	x	1

b) Indica la expresión mínima de w en suma de productos

w =  **$\neg a + c \cdot \neg d$**

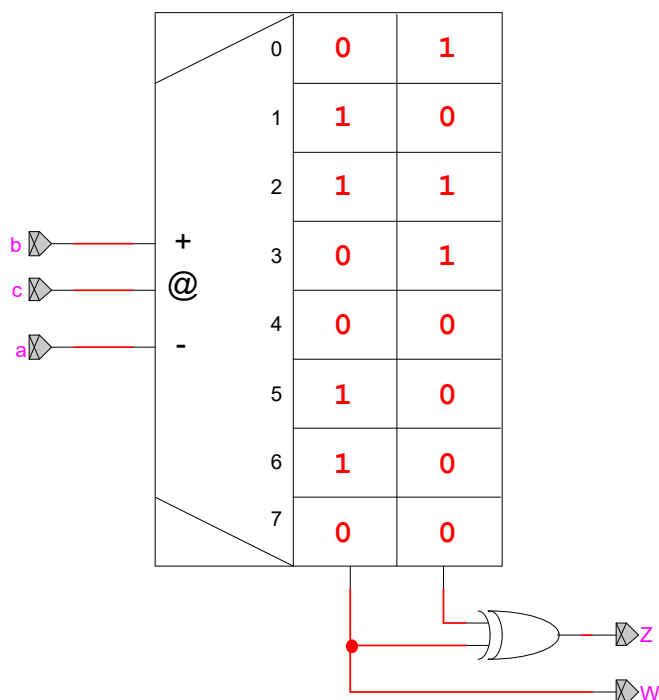
*Criterio de corrección: 1,5 puntos si grupos de unos óptimos y la expresión correcta. 1 punto si grupos de unos óptimos pero la expresión incorrecta. 0.5 puntos si grupos de unos correctos pero no óptimos. 0 Puntos si el Mapa esta mal montado (casillas con valores incorrectos)*

#### Pregunta 5) (1.5 puntos)

Queremos implementar un circuito combinacional cuya función/comportamiento esta descrito en la siguiente tabla de verdad. Para implementarlo ya tenemos el circuito diseñado. Rellena el contenido de la ROM del circuito para que haga la función descrita en la tabla de verdad.

Tabla de verdad

a	b	c	Z	W
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0



*Criterio de corrección: Cada columna vale 0.75 puntos. -0.25 puntos por cada bit mal en la columna.*

**Pregunta 6) (2 puntos)**

Queremos implementar un circuito combinacional para gestionar el sistema de control de un motor de un toldo (parasol) automático. El sistema de control del toldo recibe dos señales que le indican que debe hacer. Cuando la señal **extender** vale 1, el toldo se abre completamente. Cuando la señal **recoger** vale 1 el toldo se repliega completamente. Estas dos señales nunca pueden valer 1 simultáneamente. Mientras el toldo está haciendo una acción (abrirse o repliegarse) el sistema del control del motor ignora cualquier cambio en las señales **extender** y **recoger** hasta que ha finalizado la acción por completo.

El circuito combinacional que vamos a diseñar recibe la información de dos sensores. El primer sensor nos indica si hay mucho viento o no. La señal **viento** vale 1 cuando hace mucho viento. El segundo sensor nos indica si hace mucho sol o no. La señal **sol** vale 1 cuando hace mucho sol.

El funcionamiento del sistema automático de control es que si hace mucho sol el toldo automáticamente se debe extender y si deja de hacer mucho sol se debe replegar. Si hace mucho viento, haga sol o no, el toldo debe estar replegado. Además, nuestro sistema también puede funcionar manualmente. Tiene una señal **abrir** que cuando vale 1 indica que el toldo debe extenderse y permanecer abierto. También tiene una señal **cerrar** que cuando vale 1 indica que el toldo debe repliegarse y mantenerse cerrado. Se nos garantiza que estas señales (**abrir** / **cerrar**) nunca se activarán simultáneamente. Estas señales son prioritarias respecto a los sensores. Es decir, si se activa la señal **abrir**, da igual si hace mucho viento que el toldo debe extenderse y si se activa la señal **cerrar** el toldo debe repliegarse.

Rellenad la tabla de verdad que implementa el circuito. Poned X (valor no importa) siempre que sea posible.

viento	sol	abrir	cerrar	extender	recoger
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0*	x*
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0*	x*
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	0*	x*
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0*	x*

En las filas de la tabla de verdad correspondientes a la combinación de valores de entrada imposibles (**abrir**=1 y **cerrar**=1 simultáneamente) el valor de las salidas **extender** y **recoger** pueden ser **0X** o **X0**. No pueden valer **XX** ya que si las dos señales de salida valen X, ambas podrían escogerse como valor 1 y esto estaría en contradicción con la frase que dice que estas dos señales nunca pueden valer 1 simultáneamente.

*Criterio de corrección: Cada columna vale 1 punto. -0.3 puntos por cada bit mal en la columna.*