Clase 4

Práctico de Diseño Lógico

Clase 4 – temas

- Mapas de Karnaugh (mapas K)
- Circuitos AND – OR
- Circuitos NAND – NAND
- Eliminación de Azares

Repaso mapas K

- Es una forma gráfica de representar la tabla de la verdad de una función lógica.
- Cada combinación de entrada selecciona un único cuadrado y cada cuadrado corresponde a uno de los renglones.
- Todo se basa en la propiedad: a.b + a.!b = a
  - (Aplica lo mismo para su propiedad dual (a + b)(a + !b) = a)
- Codifico en Gray. Así, al encerrar cuadros adyacentes elimino la variable que cambia de valor.

Ejemplo mapa-k de 3 variables f(a,b,c)

a	b	c	f
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

a / bc	00	01	11	10
0				
1				

a / bc	00	01	11	10
0	0	1	1	0
1				

a / bc	00	01	11	10
0	0	1	1	0
1	0	0	1	1

Repaso algoritmo mapas K

- Caso suma de minitérminos: a.b + a.!b = a
- Tengo que cubrir todos los 1s con la menor cantidad de grupos (cantidad de compuertas AND) y con el mayor tamaño posible (menor tamaño de la compuerta).
- 1ero agrupo 1s que están solos.
- Si quedan 1s sin cubrir, luego agrupo 1s que solo se pueden agrupar de a dos de una sola forma.
- Si quedan 1s sin cubrir, luego agrupo 1s que solo se pueden agrupar de a cuatro y de una sola forma…

Ejemplo mapa-k de 3 variables f(a,b,c)

a	b	c	f
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

a / bc	00	01	11	10
0				
1				

a / bc	00	01	11	10
0	0	1	1	0
1				

a / bc	00	01	11	10
0	0	1	1	0
1	0	0	1	1

a / bc	00	01	11	10
0	0	1	1	0
1	0	0	1	1

f(abc) = !a.c + a.b

Ejemplo mapa-k de 3 variables f(a,b,c)

a	b	c	f
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

a / bc	00	01	11	10
0				
1				

a / bc	00	01	11	10
0	0	1	1	0
1				

a / bc	00	01	11	10
0	0	1	1	0
1	0	0	1	1

a / bc	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$\bar{a}.\bar{b}.c + \bar{a}.b.c = \bar{a}.c$

f(abc) = !a.c + a.b

Ejercicio minimización

Recordar:
xy + x!y = x

- Minimizar la siguiente función utilizando Algebra de Boole:
- F(a,b,c)= !a!b!c + !ab!c + !abc + ab!c + a!b!c
- Solución:
- 1. F(a,b,c)= !a!b!c + !ab!c + **!ab!c** + !abc + ab!c + a!b!c [[x + x= x]]
- 1. F(a,b,c)= !a!b!c + !ab!c + !ab!c + !abc + ab!c + a!b!c
- 2. F(a,b,c)= !a!c + !ab + a!c   [[xy + x!y = x]]
- 3. F(a,b,c)= !ab + !c [[xy + x!y = x]]

Ahora lo hacemos usando Mapas K

- F(a,b,c)= !a!b!c + !ab!c + !abc + ab!c + a!b!c

a / bc	00	01	11	10
0	1	0	1	1
1	1	0	0	1

a / bc	00	01	11	10
0				
1				

F(a,b,c) = !a.b + !c

Ejercicio 4.1.c

Ejercicio 1. (Kohavi 4.1) Con la ayuda de un mapa de Karnaugh de cuatro variables, halle las expresiones mínimas para las siguientes funciones:

a) $f_1(w,x,y,z) = \sum (0, 1, 2, 3, 4, 6, 8, 9, 10, 11)$

b) $f_2(w,x,y,z) = \sum (0, 1, 5, 7, 8, 10, 14, 15)$

c) $f_3(w,x,y,z) = \sum (0, 2, 4, 5, 6, 8, 10, 12)$

wx / yz	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

wx / yz	00	01	11	10
00	1	0	0	1
01	1	1	0	1
11	1	0	0	0
10	1	0	0	1

- No hay 1s solos
- Agrupo primero el 1 de la posición 5, ya que solo lo puedo agrupar en un único paquete con el 1 de la posición 4.
- Los otros 1s los puedo agrupar en grupos de 4. Empiezo por los que tienen una sola opción de agrupamiento: la posición 6, la posición 12 y la posición 10
- f3(w,x,y,z) = !w.x.!y + !w.!z + !y.!z +!x.!z

Ejercicio 4.2.b

b) Determine la expresión mínima (como suma de productos) para:

$$f(w,x,y,z) = \sum (0, 2, 4, 9, 12, 15) + \sum_{\phi} (1, 5, 7, 10)$$

NOTA: el símbolo $\sum_{\phi}$ indica que los mintérminos indicados son *don't care* o indetermindos, lo cual quiere decir que pueden tomar cualquier valor lógico.

wx / yz	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

wx / yz	00	01	11	10
00	1	X	0	1
01	1	X	X	0
11	1	0	1	0
10	0	1	0	X

- Orden: posición 12, 15, 9.
- Agrupo juntos los 1s de las esquinas de la fila 00 (posiciones 0 y 2)
- f(w,x,y,z) = x.!y.!z + x.y.z + !x.!y.z + !w.!x.!z
- Si voy en desorden, podría agregar grupo (innecesario) con pos 0,1,4 y 5.

Ejercicio 4.2.b

b) Determine la expresión mínima (como suma de productos) para:

$$f(w,x,y,z) = \sum(0, 2, 4, 9, 12, 15) + \sum_{\phi}(1, 5, 7, 10)$$

NOTA: el símbolo $\sum_{\phi}$ indica que los mintérminos indicados son *don't care* o indetermindos, lo cual quiere decir que pueden tomar cualquier valor lógico.

wx / yz	00	01	11	10
00	1	X	0	1
01	1	X	X	0
11	1	0	1	0
10	0	1	0	X

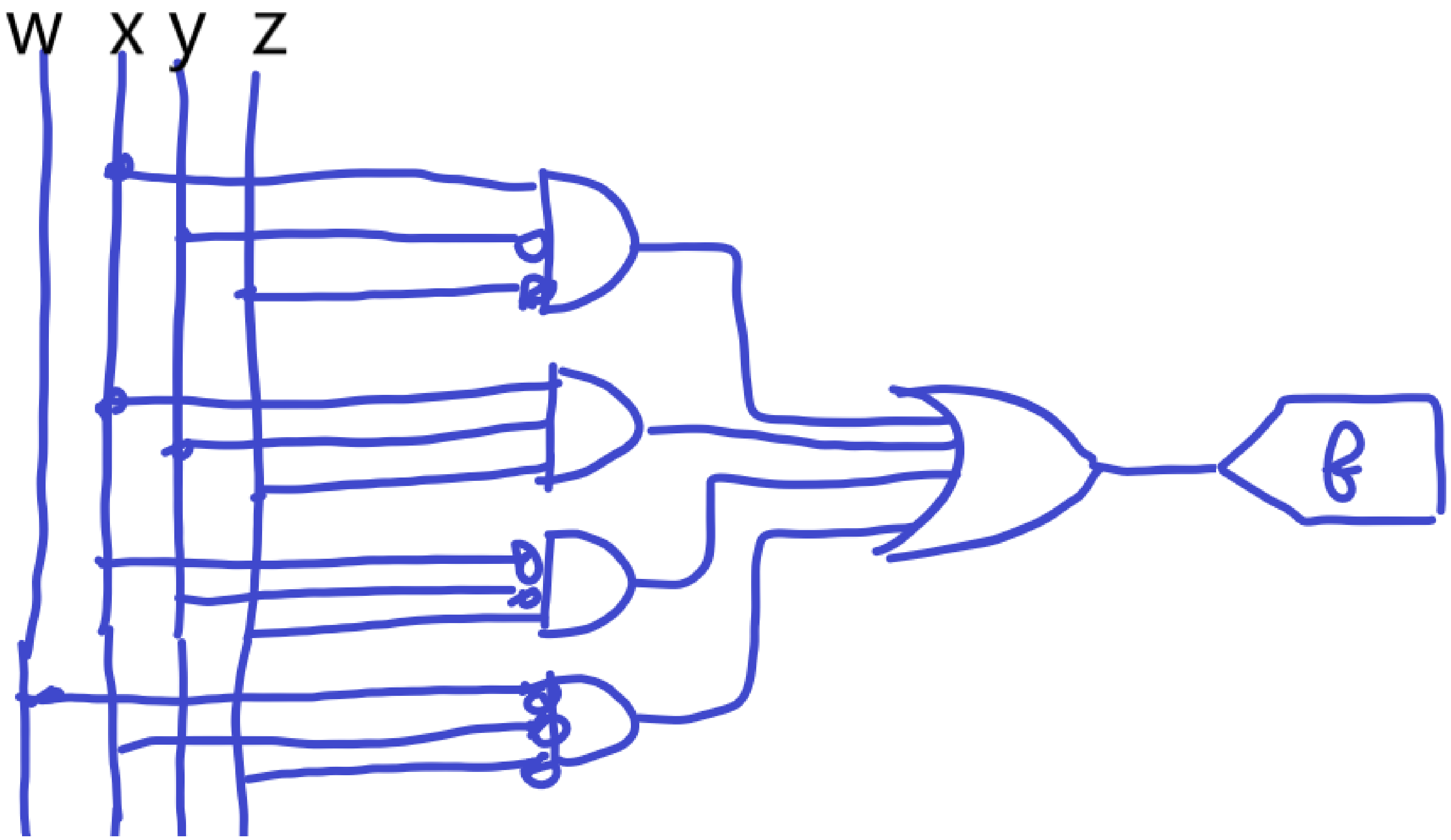
Estos X al ser agrupados, pasan a tomar el valor 1

Estos X al NO ser agrupados, pasan a tomar el valor 0

- f(w,x,y,z) = x.!y.!z + x.y.z + !x.!y.z + !w.!x.!z
- La función f(w, x, y, z) pasa a tener una salida determinada para todas sus combinaciones posibles de entradas. Por ejemplo:
- f(0,0,0,1) = 0 + 0 + !0.!0.1 + 0 = 1
- f(0,1,0,1) = 0 + 0 + 0 + 0 = 0

Circuito con compuertas And-Or

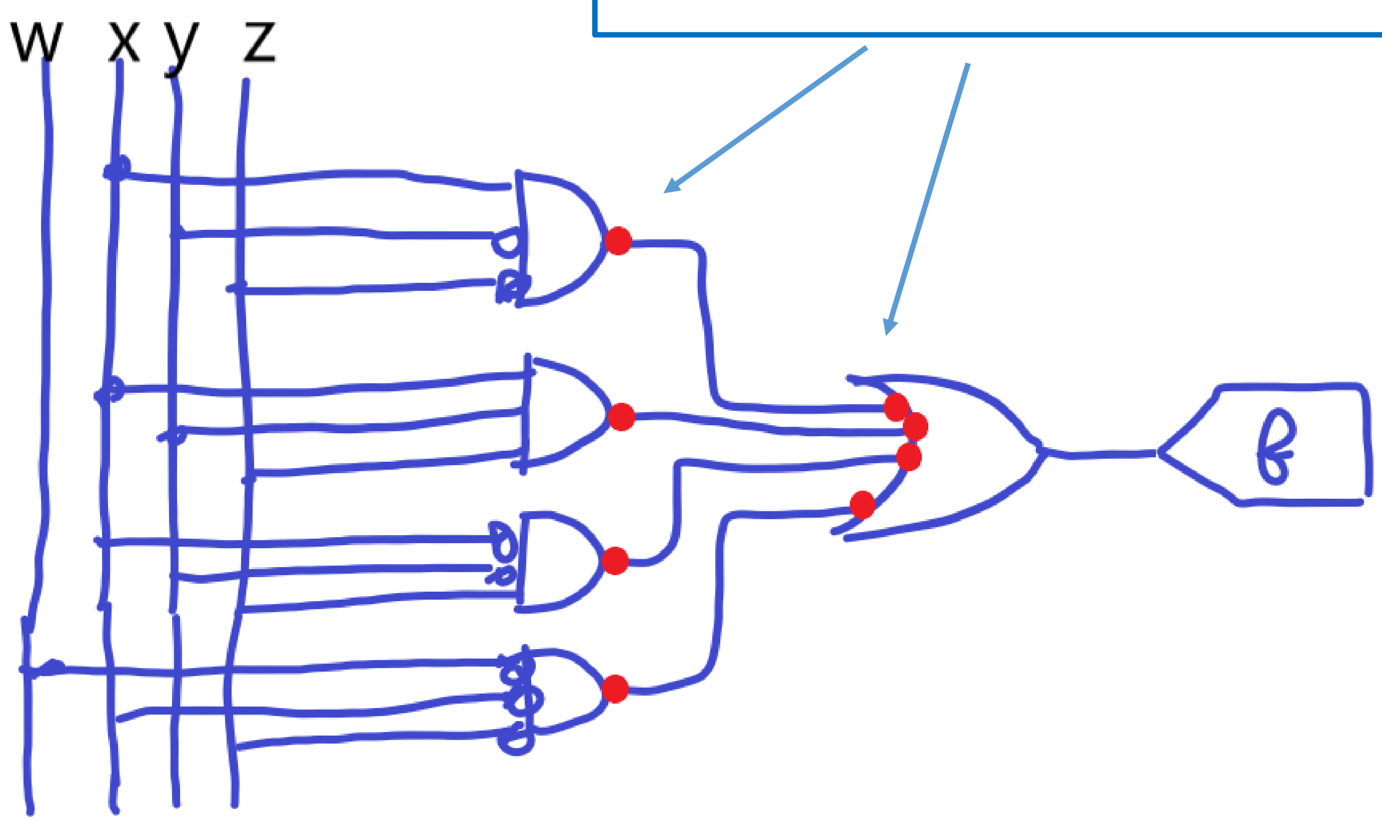
- f(w,x,y,z) = x.!y.!z + x.y.z + !x.!y.z + !w.!x.!z



Circuito con compuertas Nand-Nand

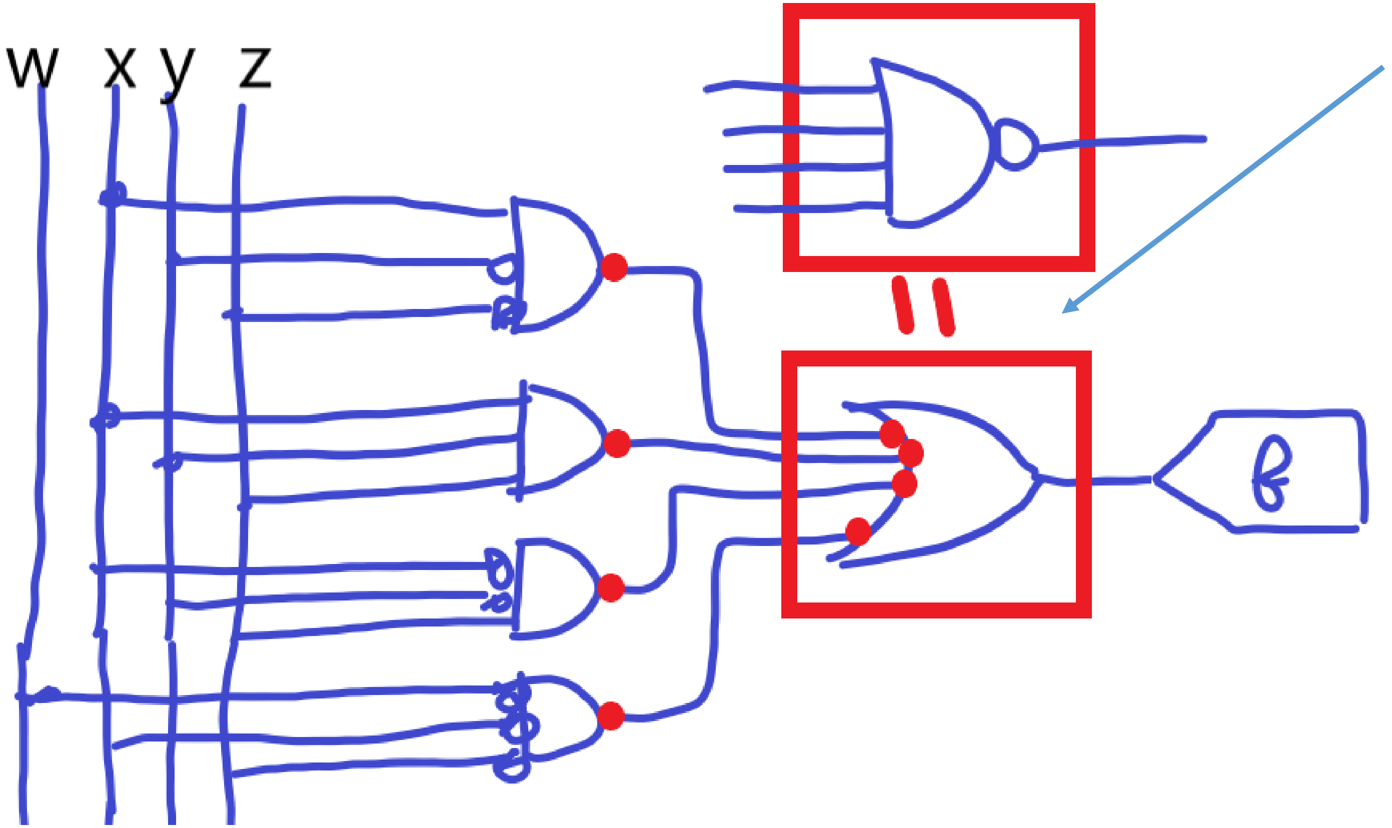
- f(w,x,y,z) = x.!y.!z + x.y.z + !x.!y.z + !w.!x.!z
- Se basa en De Morgan: !(ab)= !a + !b

1) Negamos dos veces en el cable de la and a la or (!(!x) = x)



Circuito con compuertas Nand-Nand

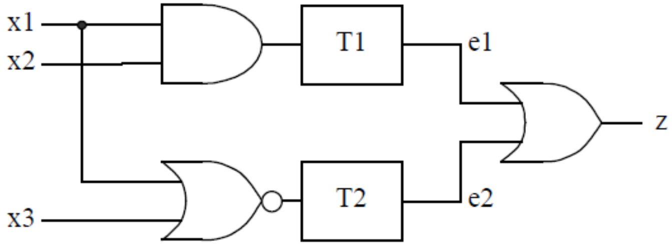
- f(w,x,y,z) = x.!y.!z + x.y.z + !x.!y.z + !w.!x.!z
- Se basa en De Morgan: !(ab)= !a + !b

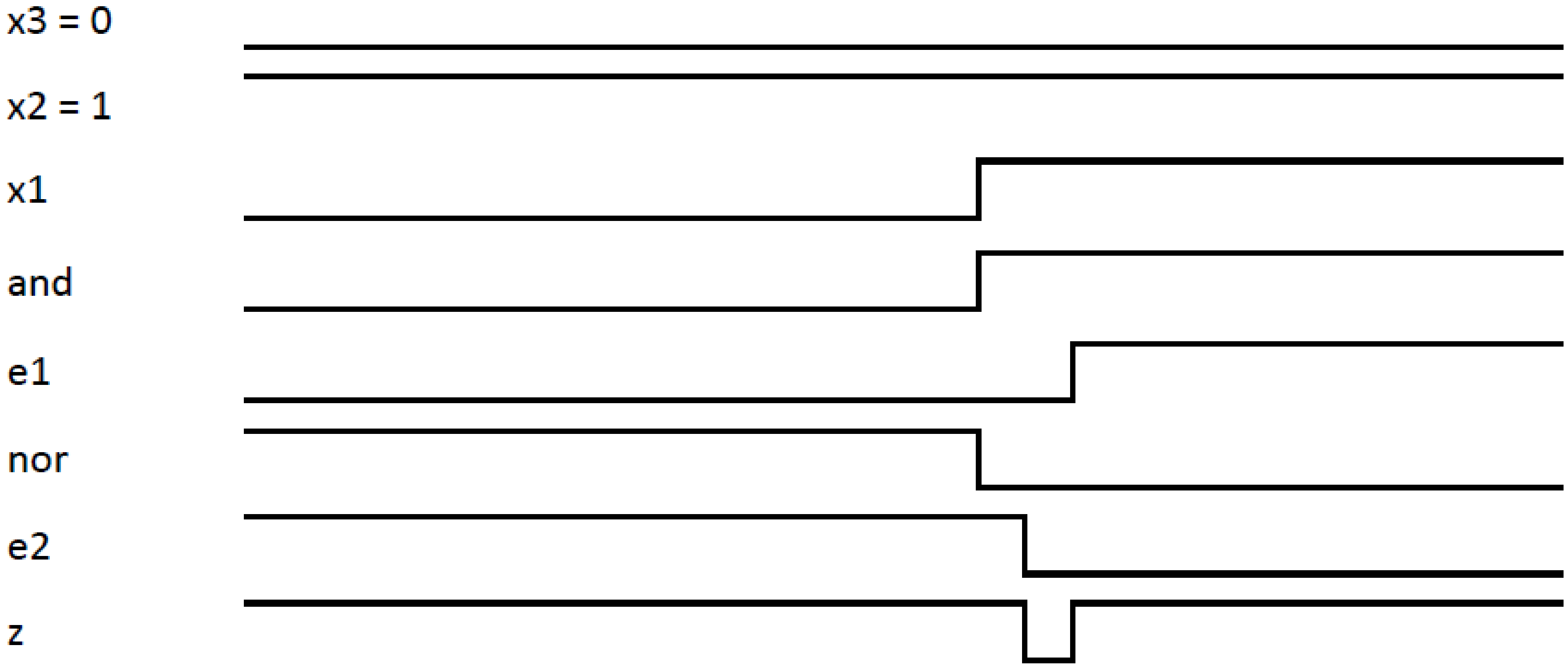


2) Reemplazamos la compuerta OR con sus entradas negadas por la compuerta NAND (De Morgan)

Azares

Ejercicio 12. (Booth 3.10) En el circuito lógico de la figura, z=1 será el valor de régimen de la salida, tanto para [x1 x2 x3] igual a [0 1 0] como para [1 1 0]. De todas maneras, cuando la entrada es [0 1 0], e1=0 y e2=1, mientras que para [1 1 0] e1=1 y e2=0. Entonces, cuando las entradas cambian de [0 1 0] a [1 1 0] o viceversa, existe la posibilidad de que z vaya momentáneamente a 0. Esta salida espuria que se produce durante tal cambio en las entradas se llama azar estático. Utilizar un diagrama de tiempos para mostrar que un azar estático existirá siempre que T1 sea distinto de T2. Use un mapa K de tres variables para explicar por qué existe ese azar. ¿Cómo puede agregarse una compuerta AND extra para eliminar este azar?



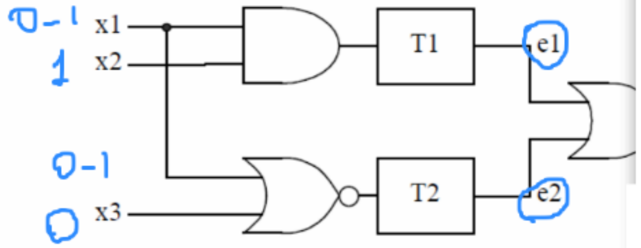


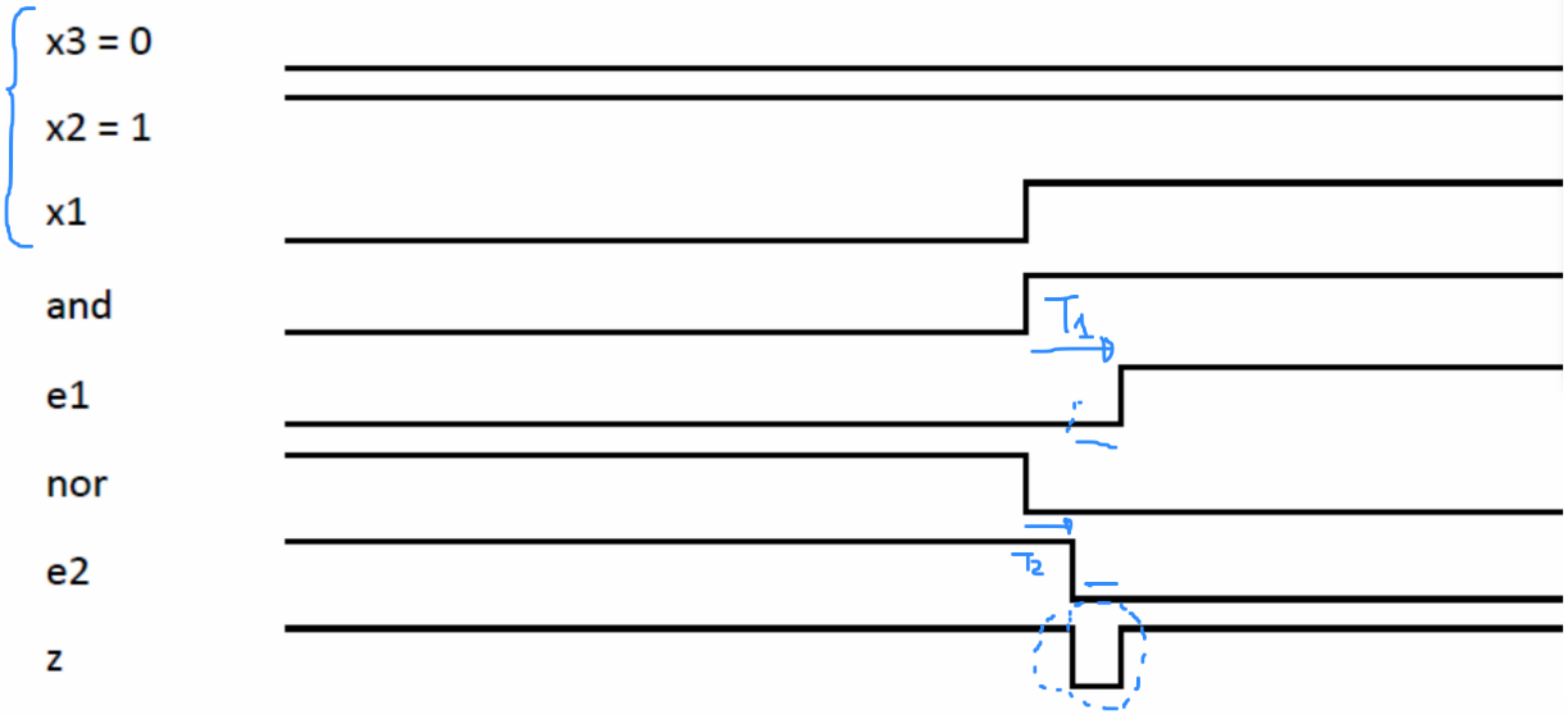
Azares

Ejercicio 12. (Booth 3.10) En el circuito lógico de la figura,
de la salida, tanto para [x1 x2 x3] igual a [0 1 0] como para
...as que
...] a [1
...vaya momentáneamente a 0. Esta sa
...en las entradas se llama azar estáti
tiempos para mostrar que un azar estático existirá siempre
Use un mapa K de tres variables para explicar por qué exi
agregarse una compuerta AND extra para eliminar este aza

$$z = x_1 \cdot x_2 + \overline{x_1 + x_3}$$

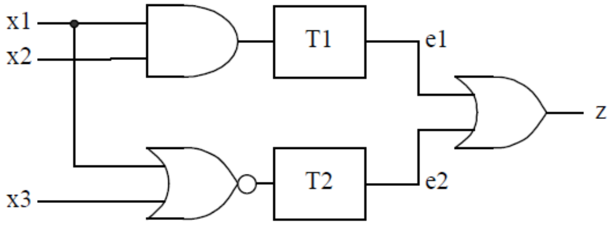
$$z = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_3}$$





Azares

Ejercicio 12. (Booth 3.10) En el circuito lógico de la figura, z=1 será el valor de régimen de la salida, tanto para [x1 x2 x3] igual a [0 1 0] como para [1 1 0]. De todas maneras, cuando la entrada es [0 1 0], e1=0 y e2=1, mientras que para [1 1 0] e1=1 y e2=0. Entonces, cuando las entradas cambian de [0 1 0] a [1 1 0] o viceversa, existe la posibilidad de que z vaya momentáneamente a 0. Esta salida espuria que se produce durante tal cambio en las entradas se llama azar estático. Utilizar un diagrama de tiempos para mostrar que un azar estático existirá siempre que T1 sea distinto de T2. Use un mapa K de tres variables para explicar por qué existe ese azar. ¿Cómo puede agregarse una compuerta AND extra para eliminar este azar?



- El problema sucede cuando, siendo x2 = 1 y x3 = 0 (columna 10), x1 cambia de 0 a 1 o de 1 a 0.
- Se arregla agrupando los 1s adyacentes de la columna 10.
- Esto nos da el término: x2.!x3
- Es decir, hay que agregar a la entrada de la compuerta OR del circuito una compuerta AND con entradas x2 y !x3.

x1 / x2x3	00	01	11	10
0	1			1
1			1	1

Circuito sin azares