Preguntas Más Frecuentes del Tema 1

P.1.1: ¿Por qué la función XOR es la función Anticoincidencia y la XNOR la función Coincidencia?

R.1.1: La función XOR es la función Anticoincidencia porque toma el valor "1" cuando los valores de las variables sobre las que opera no coinciden. Es decir, si opera sobre dos variables, A y B, la función anticoincidencia es:

$$f_1 = A \oplus B = \overline{A}\,B + A\,\overline{B}$$

Obsérvese cómo la función $\boldsymbol{f_1}$ toma el valor "1" cuando el valor del bit A es el complementario del bit B. Es decir, cuando: A =0 y B=1 ó A=1 y B=0.

Análogamente la función XNOR es la función **Coincidencia** porque toma el valor "1" cuando ambos bits coinciden, por tanto es:

$$f_2 = \overline{A \oplus B} = A\,B + \overline{A}\,\overline{B}$$

Es decir, la función toma el valor "1" cuando los dos bits toman el valor "1" ó los dos toman el valor "0".

Ambas funciones son complementarias: $f_1+f_2=1$.

✧ ✧ ✧

P.1.2: ¿Cuál es la expresión de la función Anticoincidencia o XOR de tres variables en función de los términos mínimos?

R.1.2: La función XOR de tres variables es:

$$A \oplus B \oplus C = (A \oplus B) \oplus C = \left(\overline{A}\,B + A\,\overline{B}\right) \oplus C = \left(\overline{\overline{A}\,B + A\,\overline{B}}\right)C + \left(\overline{A}\,B + A\,\overline{B}\right)\overline{C} =$$
$$= \left(A\,B + \overline{A}\,\overline{B}\right)C + \left(\overline{A}\,B + A\,\overline{B}\right)\overline{C} = A\,B\,C + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C}$$

✧ ✧ ✧

P.1.3: ¿Cómo puedo obtener una puerta NOR de 3 entradas con puertas NOR de 2 entradas?

R.1.3: Veámoslo primero a través de las expresiones lógicas:

Si aplicamos los teoremas de De Morgan a la expresión NOR de tres variables, $F_1 = \overline{A + B + C}$, la podemos expresar como:

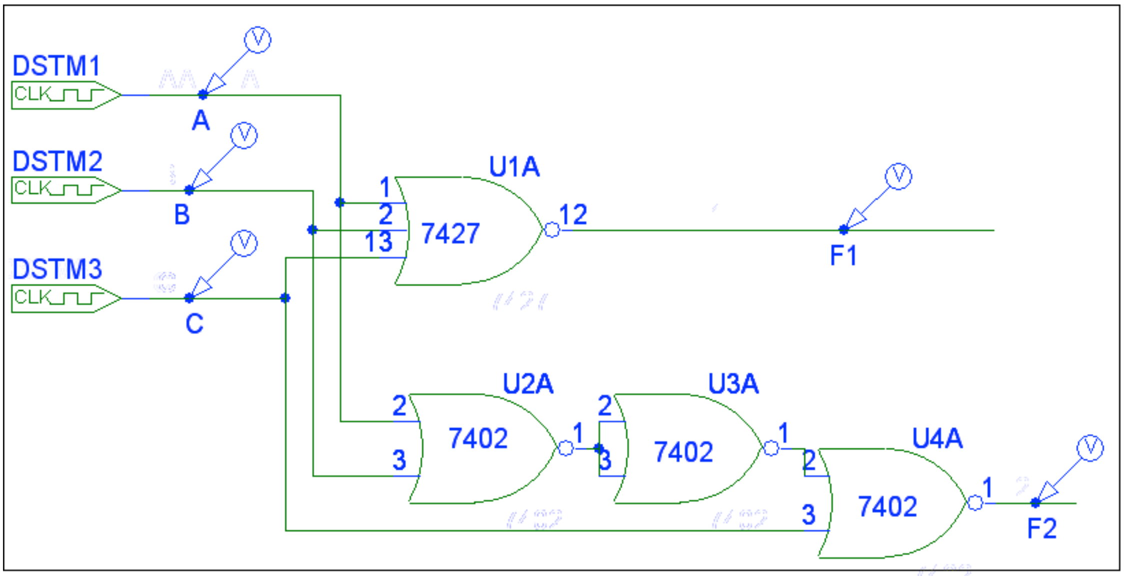
$$F_2 = \overline{(A+B)+(C)} = \overline{A+B}\ \overline{C}$$

Si ahora negamos dos veces esta función, lo que equivale a dejarla como está, y aplicamos de nuevo De Morgan, obtenemos:

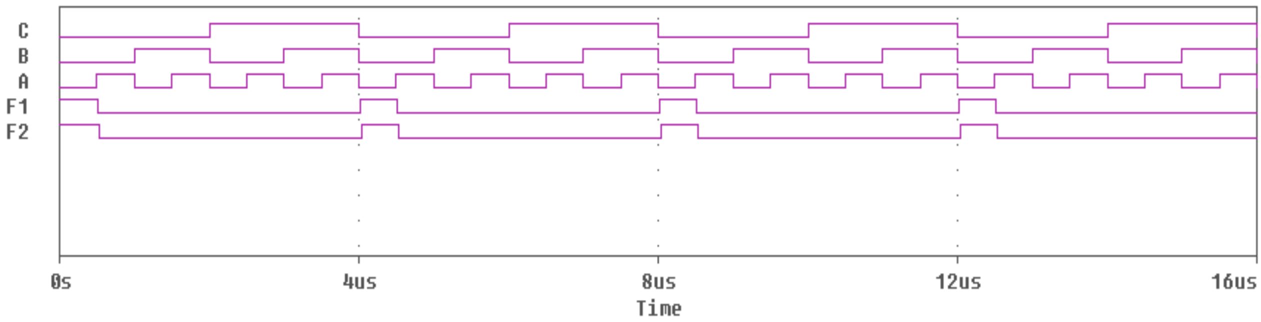
$$F_2 = \overline{A+B}\ \overline{C} = \overline{\overline{\overline{A+B}\ \overline{C}}} = \overline{\overline{\overline{A+B}} + \overline{\overline{C}}} = \overline{\overline{\overline{A+B}} + C}$$

Luego, $F_1 = \overline{A+B+C} = F_2 = \overline{\overline{\overline{A+B}}+C}$

Comprobémoslo simulando ambas expresiones.



En efecto, vemos cómo los cronogramas de ambas funciones, F1 y F2, coinciden.



✦✦✦

P.1.4: ¿Cómo puedo obtener una puerta NOR de 4 entradas con puertas NOR de 2 entradas? ¿Y con puertas NOR de 3 y 2 entradas?

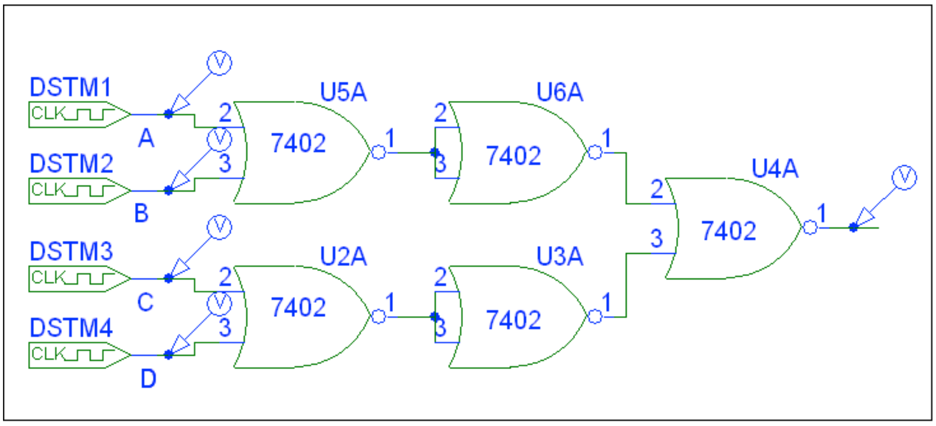
R.1.4: Veámoslo a través de las expresiones lógicas:

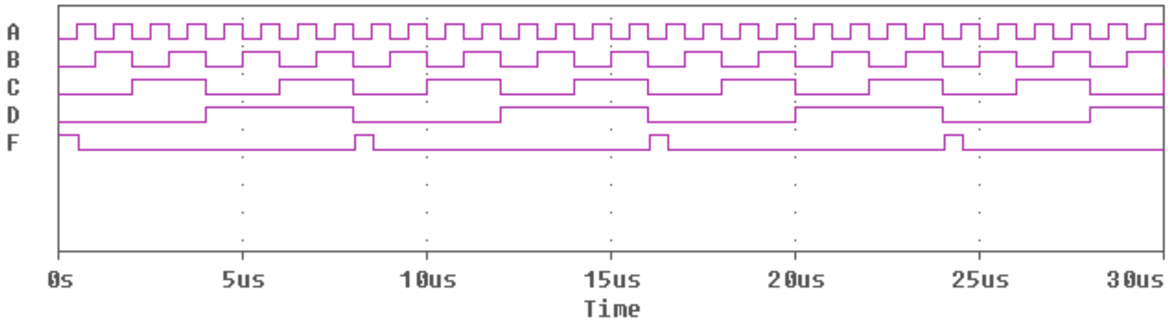
Aplicando los teoremas de De Morgan obtenemos: $F = \overline{A+B+C+D} = \overline{A+B}\cdot\overline{C+D}$

Negando dos veces esta función, y aplicando de nuevo De Morga resulta:

$$F = \overline{\overline{\overline{A+B}\cdot\overline{C+D}}} = \overline{\overline{\overline{A+B}}+\overline{\overline{C+D}}}$$

Si implementamos esta expresión en el simulador, obtenemos:

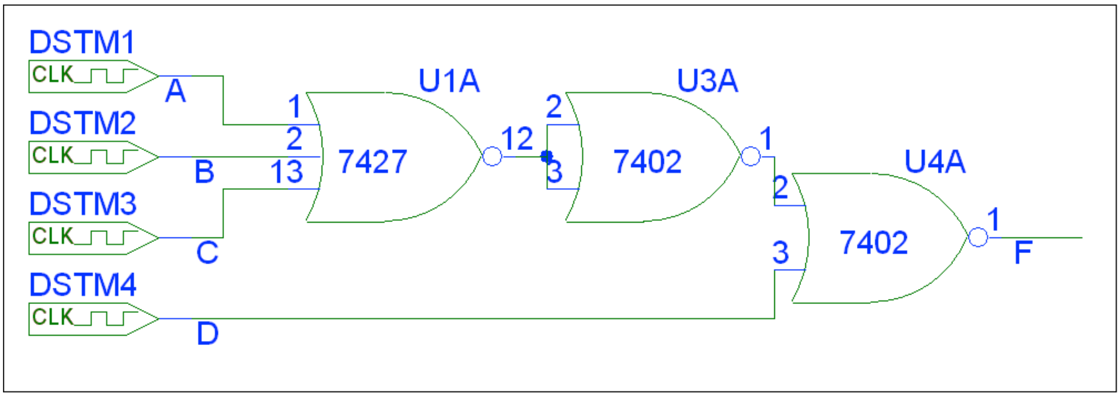


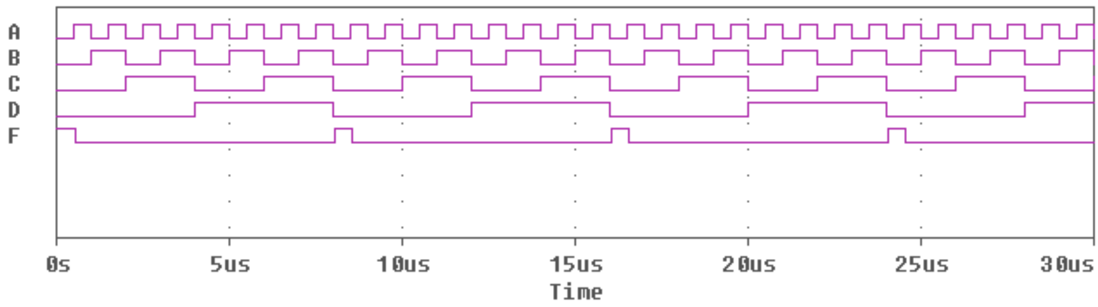


Como podemos comprobar, este circuito realiza la función NOR de las cuatro variables de entrada, ya que se produce un pulso en la salida, F, sólo cuando las 4 variables de entrada toman el valor "0", mientras que en todas las configuraciones de entrada en las que alguna de las variables toma el valor "1", la salida toma el valor "0".

Análogamente, para obtener puertas NOR de 4 variables mediante NOR de 2 y 3 entradas tenemos:

$$F = \overline{A+B+C+D} = \overline{A+B+C}\;\overline{D} = \overline{\overline{\overline{A+B+C}\;\overline{D}}} = \overline{\overline{\overline{A+B+C}} + \overline{\overline{D}}} = \overline{\overline{\overline{A+B+C}} + D}$$





Si comparamos este cronograma con el anterior vemos que las respuestas de ambos circuitos o funciones F en ambos cronogramas son las mismas ante las mismas configuraciones de entrada.

✦✦✦

P.1.5: ¿Cómo se construye la tabla de verdad de una función y cómo se obtiene la expresión lógica de la función que representa?

R.1.5: La tabla de verdad de una función es una forma de representar dicha función en la que se especifican de forma completa, en extenso, el valor que toma la función para cada una de las posibles configuraciones de valores de las variables que participan en ella.

Generalmente, usamos la tabla de verdad para obtener la expresión lógica de la función que dicha tabla representa.

Veámoslo mediante un ejemplo.

Supongamos que queremos obtener la expresión lógica de la función OR que definimos, en lenguaje natural, como aquella función que toma el valor "1" cuando al menos una de sus variables toma el valor "1" y, lógicamente, toma el valor "0" cuando todas las variables toman el valor "0".

Cuando tenemos que sintetizar alguna función lógica, lo primero que hacemos es construir la tabla de verdad correspondiente, empezando por poner todos los términos mínimos a que dan lugar las variables y expresados mediante sus valores, el "0" para las variables negadas y el "1" para las variables sin negar.

En este ejemplo la función que vamos a sintetizar es de 2 variables por lo que hay 2^2=4 términos mínimos. Así, si las variables son, por ejemplo, *A* y *B*, los términos mínimos son:

$\overline{A}\,\overline{B}$ que se corresponde con *A*=0, *B*=0.

$\overline{A}\,B$ que se corresponde con *A*=0, *B*=1.

$A\,\overline{B}$ que se corresponde con *A*=1, *B*=0.

$A\,B$ que se corresponde con *A*=1, *B*=1.

Así, la tabla de verdad es:

A	B	F
0	0	
0	1	
1	0	
1	1	

El siguiente paso es poner los valores que toma la función, F=A OR B, ante cada uno de los términos mínimos. Luego en este caso ponemos un "1" en todos términos mínimos menos en el primero que es el único que no tiene ningún "1". Por tanto la tabla de verdad de a función OR es:

A	B	F
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

A continuación tenemos que obtener la expresión lógica de la función OR, expresada como suma de los términos mínimos que participan en ella. Para ello, vemos en la Tabla de Verdad a qué términos mínimos corresponden los "1" de *F* y sumamos todos esos términos mínimos. Así, obtenemos:

$$F = \overline{A}\,B + A\,\overline{B} + A\,B$$

A partir de aquí, minimizamos y obtenemos:

$$F = A\left(\overline{B} + B\right) + B\left(\overline{A} + A\right) = A + B$$

Que, lógicamente, es la función OR o suma lógica de A y B: *F = A OR B = A+B.*

✦✦✦

P.1.6: No acabo de entender la representación de las funciones lógicas en la forma Normal Disyuntiva y en la forma Normal Conjuntiva. ¿Me lo podrían explicar?

R.1.6: Como se explica en el texto base (pag. 33), toda función lógica se puede representar de distintas formas sin que cambie dicha función.

Aquí vamos a centrarnos en la representación en su Forma Normal Disyuntiva o suma de términos mínimos (minterms) y en su Forma Normal Conjuntiva o producto de términos máximos (maxterms) (pag 33 a 43 del texto). Es muy importante que quede claro que, en realidad, son dos formas distintas y duales de representar la misma función.

Partimos de la base de que ya sabemos construir la tabla de verdad de una función lógica. Por tanto sólo nos vamos a centrar en cómo representamos una función lógica mediante minterms y mediante maxterms y comprobaremos que ambas funciones coinciden, sólo que representadas de diferente forma.

Vamos a verlo a través de un ejemplo y para ello vamos a usar la función anticoincidencia de dos variables.

La función anticoincidencia de dos variables, como se explica en otra de las Preguntas Más Frecuentes, se caracteriza por tomar el valor "1" siempre que las dos variables son distintas y tomar el valor "0" cuando las dos variables coinciden. Así, si llamamos *F* a la función y *A* y *B* a las variables, tenemos que se debe verificar que: *F(A,B) = 1*, si *A* es distinto de *B* y *F(A,B) = 0*, si *A* es igual a *B*.

Vamos a partir de la tabla de verdad a la que vamos a añadirle dos columnas encabezadas con *minterms* y *maxterms* en cuyas filas vamos a poner las representaciones mediante minterms y maxterms de las configuraciones de entrada correspondientes. Así, la tabla de verdad para esta función es:

A	***B***	***Minterms***	***Maxterms***	***F=Anticoincidencia***
0	0	$m0 = \overline{A}\,\overline{B}$	$M0 = A + B$	**0**
0	1	$m1 = \overline{A}\,B$	$M1 = A + \overline{B}$	**1**
1	0	$m2 = A\,\overline{B}$	$M2 = \overline{A} + B$	**1**
1	1	$m3 = A\,B$	$M3 = \overline{A} + \overline{B}$	**0**

Veamos los procedimientos para obtener las distintas formas de representar *F*.

a) Representación en la ***Forma Normal Disyuntiva*** (ver pag. 33) en la que se representa mediante suma de productos (suma de términos mínimos). Veamos dos formas de hallar esta representación:

a.1: Elegimos las filas en las que ***F = 1*** y ***sumamos los términos mínimos*** (productos de las variables) de las filas correspondientes.

A	***B***	***Minterms***	***Maxterms***	***F=Anticoincidencia***
0	0	$m0 = \overline{A}\,\overline{B}$	$M0 = A + B$	**0**
0	1	$m1 = \overline{A}\,B$	$M1 = A + \overline{B}$	**1**
1	0	$m2 = A\,\overline{B}$	$M2 = \overline{A} + B$	**1**
1	1	$m3 = A\,B$	$M3 = \overline{A} + \overline{B}$	**0**

Así, la expresión de F representada mediante minterms es:

$$F = A \oplus B = m1 + m2 = \overline{A}\,B + A\,\overline{B}$$

a.2: Elegimos las filas en las que ***F = 0*** y ***sumamos los términos mínimos*** correspondientes, pero como hemos tomado los términos mínimos que hacen que ***F*** sea igual a "0", la función que obtenemos es $\overline{F}$.

A	***B***	***Minterms***	***Maxterms***	***F=Anticoincidencia***
0	0	$m0 = \overline{A}\,\overline{B}$	$M0 = A + B$	**0**
0	1	$m1 = \overline{A}\,B$	$M1 = A + \overline{B}$	**1**
1	0	$m2 = A\,\overline{B}$	$M2 = \overline{A} + B$	**1**
1	1	$m3 = A\,B$	$M3 = \overline{A} + \overline{B}$	**0**

Así, en este caso tenemos: $\overline{F} = m0 + m3 = \overline{A}\,\overline{B} + A\,B$

Por tanto, $F = \overline{m0 + m3} = \overline{\overline{A}\,\overline{B} + A\,B}$

Ahora tendremos que demostrar las dos formas de calcular F (a.1 y a.2) nos llevan realmente a la misma expresión: .

Para ello vamos a partir de esta última expresión de F (calculada en a.2) y, aplicando el álgebra de Boole, deberemos llegar a la expresión de F que hemos obtenido en el apartado a.1. En efecto,

$$F = \overline{m0 + m3} = \overline{\overline{A}\,\overline{B} + A\,B} = \overline{\overline{A}\,\overline{B}}\;\overline{A\,B} = (A+B)(\overline{A}+\overline{B}) = \overbrace{\cancel{A\,\overline{A}}}^{0\ (\text{porque } 1\cdot 0 = 0)} + A\,\overline{B} + \overline{A}\,B + \overbrace{\cancel{B\,\overline{B}}}^{0} = A\,\overline{B} + \overline{A}\,B = A \oplus B$$

Aplicando De MORGAN **Operando**

Luego ambas funciones coinciden.

Otra ***forma más inmediata*** de demostrar que ambas funciones coinciden es aplicando directamente el postulado de la complementariedad que nos dice que, $X + \overline{X} = 1$. Por tanto:

$$F = \overline{m0 + m3} = 1 - (m0 + m3) = m1 + m2 = \overline{A}\,B + A\,\overline{B},$$

puesto que $m0 + m1 + m2 + m3 = 1$ y, por tanto, $1 - (m0 + m3) = m1 + m2$

Luego, $F = \overline{m0 + m3} = m1 + m2$

b) Representación en la **Forma Normal Conjuntiva** (ver pag. 36) en la que se representa la función mediante producto de sumas (producto de términos máximos). Veamos dos formas de hallar esta representación:

b.1: Elegimos las filas en las que ***F=0*** y multiplicamos los términos máximos (sumas de las variables) de las filas correspondientes.

A	*B*	*Minterms*	*Maxterms*	*F=Anticoincidencia*
0	0	$m0=\bar{A}\,\bar{B}$	$M0=A+B$	**0**
0	1	$m1=\bar{A}\,B$	$M1=A+\bar{B}$	**1**
1	0	$m2=A\,\bar{B}$	$M2=\bar{A}+B$	**1**
1	1	$m3=A\,B$	$M3=\bar{A}+\bar{B}$	**0**

Así, $F=M0\cdot M3=(A+B)(\bar{A}+\bar{B})$. Esta es la expresión de ***F*** representada mediante ***maxterms***.

Veamos que esta función también coincide con la función calculada anteriormente. En efecto,

$$F=M0\cdot M3=(A+B)(\bar{A}+\bar{B})\underset{\text{Operando}}{=}\overset{0}{A\bar{A}}+A\bar{B}+\bar{A}B+\overset{0}{B\bar{B}}=A\bar{B}+\bar{A}B=A\oplus B$$

b.2. Al igual que hicimos en la representación mediante minterms podemos obtener la misma función representación mediante maxterms usando los "1" en vez de usar los "0". Las filas en las que ***F=1*** y multiplicamos los términos máximos correspondientes, pero como hemos tomado los términos máximos que hacen que *F* sea igual a "1", la función que obtenemos es $\bar{F}$.

A	*B*	*Minterms*	*Maxterms*	*F=Anticoincidencia*
0	0	$m0=\bar{A}\,\bar{B}$	$M0=A+B$	**0**
0	1	$m1=\bar{A}\,B$	$M1=A+\bar{B}$	**1**
1	0	$m2=A\,\bar{B}$	$M2=\bar{A}+B$	**1**
1	1	$m3=A\,B$	$M3=\bar{A}+\bar{B}$	**0**

Así, en este caso obteneos: $\bar{F}=M1\cdot M2=(A+\bar{B})\,(\bar{A}+B)$.

Por tanto, $F = \overline{M1 \cdot M2} = \left(A + \overline{B}\right)\left(\overline{A} + B\right)$

Comprobemos ahora que ambas expresiones coinciden. En efecto:

$$F = \overline{M1 \cdot M2} = \overline{\left(A+\overline{B}\right)\left(\overline{A}+B\right)} = \overline{\left(A+\overline{B}\right)} + \overline{\left(\overline{A}+B\right)} = \overline{A}\,B + A\,\overline{B} = A \oplus B$$

Aplicando De MORGAN

Como podemos observar, hemos obtenido la representación de la función anticoincidencia mediante ***minterms*** y mediante ***maxterms*** y además ***seleccionando los ceros*** o ***los unos*** de ***F***. Es decir tenemos 4 formas (las cuatro combinaciones posibles) de obtener la expresión lógica de cualquier función y todas coinciden y, por tanto, queda claro que todas estas formas representan a la misma función.

Observemos cómo la representación de una función en su ***Forma Normal Disyuntiva*** es la dual de la representación en su ***Forma Normal Conjuntiva*** ya que podemos pasar de una representación a otra haciendo los siguientes cambios:

Suma (Σ) por ***Producto*** (Π) ó (***OR*** por *AND*) ó (**+** por **·**)

Producto (Π) por ***Suma*** (Σ) ó (***AND*** por ***OR***) ó (**·** por **+**)

"**1**" por "**0**"

"**0**" por "**1**"

$\boldsymbol{F}$ por $\overline{\boldsymbol{F}}$

$\overline{\boldsymbol{F}}$ por $\boldsymbol{F}$

variable por $\overline{\textbf{\textit{variable}}}$ (ejemplo A por $\overline{A}$)

$\overline{\textbf{\textit{variable}}}$ por ***variable***

En otra de las Preguntas Más Frecuentes se explica cómo aplicar los diagramas de Venn a las expresiones mediante la Forma Normal Disyuntiva y la Forma Normal Conjuntiva.

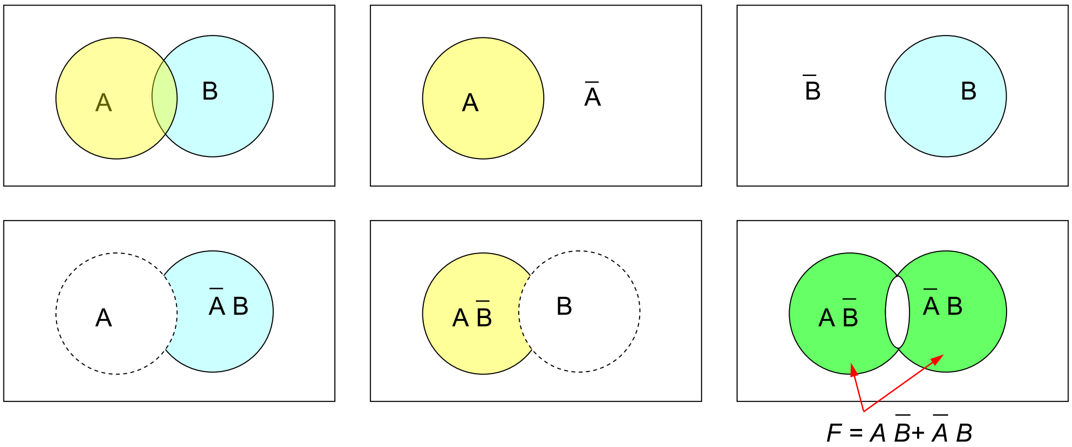
✧ ✧ ✧

P.1.7: Me gustaría conocer un ejemplo de uso de los diagramas de Venn, en concreto su relación con la Forma Normal Disyuntiva y la Forma Normal Conjuntiva

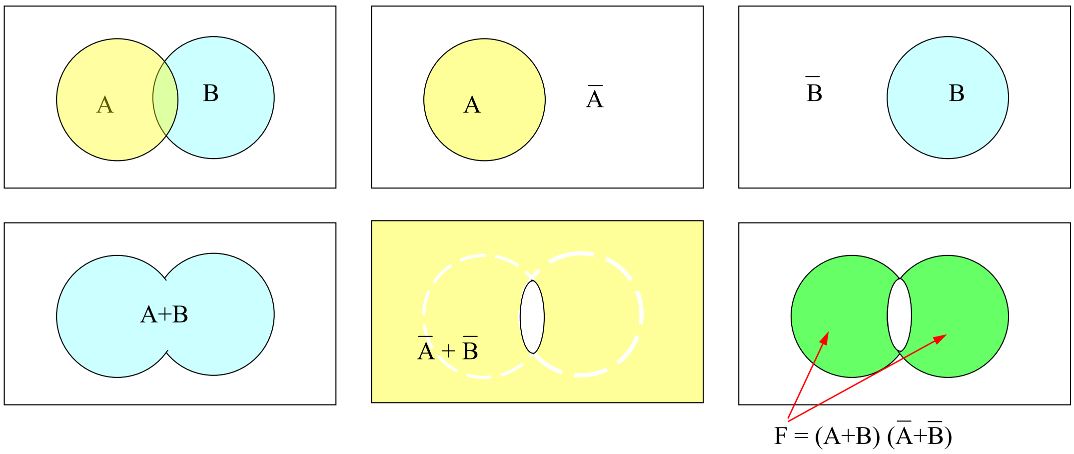
R.1.7: Vamos a aplicar el concepto de los diagramas de Venn a las diferentes expresiones de la función anticoincidencia de dos variables, que ya se han calculado en otra de las Preguntas Más Frecuentes, en la que se ha explicado la Forma Normal Conjuntiva y la Forma Normal Disyuntiva.

Vamos a aplicarlo a las expresiones obtenidas en los apartados ***a.1*** y ***b.1*** de esa Pregunta Más Frecuente.

Para la función del apartado ***a.1:*** $F = A \oplus B = m1 + m2 = \overline{A}\,B + A\,\overline{B}$, resulta:



Para la función del apartado ***b.1:*** $F = M0 \cdot M3 = (A + B)(\overline{A} + \overline{B})$ y resulta:



Como vemos las áreas matizadas en verde coinciden. La primera está obtenida como ***unión de intersecciones*** (suma de productos) y la segunda como ***intersección de uniones*** (producto de sumas).

✦ ✦ ✦

P.1.8: ¿Cómo puedo representar la función $F(A,B) = \overline{\overline{A} + A\,\overline{B}}$ en las dos formas canónicas (Normal Conjuntiva y Normal Disyuntiva), o sea, mediante suma de minterms y producto de maxterms?

R.1.8: Para usar los términos mínimos y máximos es muy importante entender la tabla de la figura 1.18, ya que en ella está resumido todo el conocimiento necesario.

Hay distintas formas de hacerlo. En la Pregunta anterior (P.1.8) hemos explicado la forma de hacerlo mediante el uso de las tablas de verdad. Aquí lo vamos a hacer operando directamente sobre las variables y sobre los términos mínimos y los máximos.

a) **Representación por Minterms**.

Si aplicamos repetidamente los Teoremas de DeMorga y el postulado de la complementariedad resulta:

$$F(A,B) = \overline{\overline{A} + A\,\overline{B}} = \overline{\overline{A}}\;\overline{A\,\overline{B}} = A\left(\overline{A} + B\right) = A\,\overline{A} + A\,B = AB \quad \text{ya que} \quad A\,\overline{A} = 0$$

Por tanto la representación de la función mediante términos mínimo es:

$$F(A,B) = \overline{\overline{A} + A\,\overline{B}} = AB = m3$$

b) **Representación por Maxterms**.

Hay varias formas de hacerlo.

b.1) La más inmediata es recordando de la tabla de la fig. 1.18 que:

$$M_i = \overline{m_i}\ , \quad \text{o bien} \quad m_i = \overline{M_i}$$

Por tanto podemos poner que $F(A,B) = \overline{\overline{A} + A\,\overline{B}} = AB = m_3 = \overline{M_3} = M_0\,M_1\,M_2$

b.2) Otra forma de llegar a la solución correcta, sin necesidad de recordar nada, es operando y aplicando los postulados y teoremas del Álgebra de Boole sobre la expresión inicial. Este procedimiento, por lo general, suele ser el más seguro, aunque más largo y laborioso. Así,

$$F(A,B) = \overline{\overline{A} + A\,\overline{B}} = \overline{\overline{A}\left(B + \overline{B}\right) + A\,\overline{B}} = \overline{\overline{A}\,B + \overline{A}\,\overline{B} + A\,\overline{B}} = \overline{\overline{A}\,B}\;\;\overline{\overline{A}\,\overline{B}}\;\;\overline{A\,\overline{B}} = \left(A + \overline{B}\right)\left(A + B\right)\left(\overline{A} + B\right) = M_1\,M_0\,M_2$$

b.3) Operando sobre el resultado de la representación por minterms. Así, si negamos dos veces dicha expresión y operamos, resulta:

$$F(A,B) = \overline{\overline{A} + A\,\overline{B}} = AB = m_3 = \overline{\overline{m_3}} = \overline{1 - m_3} = \overline{m_0 + m_1 + m_2} = \overline{m_0}\;\overline{m_1}\;\overline{m_2} = M_0\,M_1\,M_2$$

Luego, la solución correcta es:

$$F(A,B) = \overline{\overline{A} + A\,\overline{B}} = m_3 = M_0\,M_1\,M_2$$

P.1.9: ¿Cómo podemos pasar de la función $F = M_1\,M_2\,M_5\,M_9\,M_{12}\,M_{13}$ que está representada mediante el producto de términos máximos a representarla mediante la suma de los términos mínimos?

R.1.9: Como hemos visto reiteradamente hay varias formas de pasar de una representación a otra.

a) La más rápida e inmediata es negamos dos veces, complementamos, aplicamos De Morgan y que $M_i = \overline{m_i}$. Así,

$$F = \overline{\overline{F}} = \overline{\overline{M_1\ M_2\ M_5\ M_9\ M_{12}\ M_{13}}} = \overline{M_0\ M_3\ M_4\ M_6\ M_7\ M_8\ M_{10}\ M_{11}\ M_{14}\ M_{15}} =$$
$$= \overline{M_0} + \overline{M_3} + \overline{M_4} + \overline{M_6} + \overline{M_7} + \overline{M_8} + \overline{M_{10}} + \overline{M_{11}} + \overline{M_{14}} + \overline{M_{15}} =$$
$$= m_0 + m_3 + m_4 + m_6 + m_7 + m_8 + m_{10} + m_{11} + m_{14} + m_{15}$$

Por tanto:

$$F = M_1\ M_2\ M_5\ M_9\ M_{12}\ M_{13} = m_0 + m_3 + m_4 + m_6 + m_7 + m_8 + m_{10} + m_{11} + m_{14} + m_{15}$$

✧ ✧ ✧

P.1.10: ¿Me podrían explicar cómo a partir de 2 variables podemos obtener 16 funciones lógicas distintas y cuáles son estas?

R.1.10: Vamos a ver la representación de todas las posibles funciones lógicas de dos variables expresadas mediante términos mínimos o suma de productos.

En general, partimos de un determinado número de variables, cada una de las cuales puede tomar el valor “0” ó “1”. En lógica positiva, que es la que usamos a lo largo de todo el texto, si la variable toma el valor “0” la representamos por la variable negada y si toma el valor “1” la representamos por la variable sin negar. Los términos mínimos son productos de todas las variables, negadas y sin negar, sin repetirse ninguna.

Si tenemos n variables, el número total de configuraciones distintas o de términos mínimos que podemos obtener son: 2^n.

En este caso vamos a considerar que sólo tenemos 2 variables, A y B, por lo que el número de términos mínimos o de configuraciones posibles son 2^2=4: Estos términos mínimos son:

$$\overline{A}\,\overline{B},\ \overline{A}\,B,\ A\,\overline{B} \text{ y } A\,B.$$

El número de funciones lógicas que podemos representar en su ***Forma Normal Disyuntiva*** (suma de términos mínimos) con estos 4 términos mínimos es: 4^2=16. Estas funciones son todas las funciones que se pueden formar sumando uno, o dos, o tres, o los cuatros o ninguno de los términos mínimos. Veamos cuales son estas funciones, sin minimizar:

Función lógica en la que no interviene ***ningún término mínimo***:

- $F_0 = 0$

Funciones lógicas en las que sólo interviene ***un término mínimo***:

- $F_1 = A\,B$
- $F_2 = A\overline{B}$
- $F_3 = \overline{A}\,B$

- $F_4 = \overline{A}\,\overline{B}$

Funciones lógicas en las que sólo intervienen dos términos mínimos:

- $F_5 = A\,B + A\,\overline{B}$
- $F_6 = A\,B + \overline{A}\,B$
- $F_7 = A\,B + \overline{A}\,\overline{B}$
- $F_8 = A\,\overline{B} + \overline{A}\,B$
- $F_9 = A\,\overline{B} + \overline{A}\,\overline{B}$
- $F_{10} = \overline{A}\,B + \overline{A}\,\overline{B}$

Funciones lógicas en las que interviene tres términos mínimos:

- $F_{11} = A\,B + A\,\overline{B} + \overline{A}\,B$
- $F_{12} = A\,B + A\,\overline{B} + \overline{A}\,\overline{B}$
- $F_{13} = A\,B + \overline{A}\,B + \overline{A}\,\overline{B}$
- $F_{14} = A\,\overline{B} + \overline{A}\,B + \overline{A}\,\overline{B}$

Función lógica en las que interviene los cuatro términos mínimos:

- $F_{15} = A\,B + A\,\overline{B} + \overline{A}\,B + \overline{A}\,\overline{B}$

Estas son las mismas funciones que aparecen en la tabla de la parte inferior de la figura 1.16 del texto, aunque allí, para hacerlo de forma ordenada respecto a los valores que va tomando la palabra a, están en el orden correspondiente al equivalente en decimal de la palabra de programación del circuito (a_0, a_1, a_2, a_3).

✦✦✦

P.1.11: ¿Cómo se puede simplificar la expresión $F = \overline{x_1}\,\overline{x_2} + x_1\,\overline{x_2} + x_1\,x_2$ **?**

R.1.11: Veamos las distintas formas de minimizar esta expresión lógica.

1) Aplicando directamente los postulados y teoremas del Algebra de Boole:

Si tenemos en cuenta el ***Teorema de Idempotencia*** podemos duplicar cualesquiera de los sumandos o términos mínimos que participan en una determinada función sin que esta cambie, ya que este teorema nos dice que $a + a = a$. Es decir, la función OR de una variable consigo misma es ella misma.

Si aplicamos este teorema a la expresión de *F* duplicando $x_1\,\overline{x_2}$ resulta

$$F = \overline{x_1}\,\overline{x_2} + x_1\,\overline{x_2} + x_1\,x_2 + x_1\,\overline{x_2}$$

Si ahora sacamos factor común $\overline{x_2}$ del primer y segundo sumando y x_1 del tercero y cuarto obtenemos:

$$F = \overline{x_2}\left(\overline{x_1} + x_1\right) + x_1\left(x_2 + \overline{x_2}\right)$$

Si ahora tenemos en cuenta el postulado de la complementariedad, $a + \overline{a} = 1$, resulta:

$$F = x_1 + \overline{x_2}$$

2) Usando el postulado de la complementariedad

Si observamos la expresión de ***F*** vemos que está formada por la suma de tres de los cuatro términos mínimos posibles de dos variables.

Como la suma de todos los términos mínimos vale 1. Es decir, $\overline{x_1}\;\overline{x_2} + x_1\;\overline{x_2} + x_1\;x_2 + \overline{x_1}\;x_2 = 1$, podemos poner:

$$F = \overline{x_1}\;\overline{x_2} + x_1\;\overline{x_2} + x_1\;x_2 = 1 - \overline{x_1}\;x_2,$$

Pero, $1 - \overline{x_1}\;x_2 = \overline{\overline{x_1}\;x_2}$

Si aplicamos el teorema de De Morgan resulta:

$$F = \overline{\overline{x_1}\;x_2} = x_1 + \overline{x_2}$$

3) Mediante el diagrama de Karnaugh:

x_2 \ x_1	$\overline{x_2}$ 0	x_2 1
$\overline{x_1}$ 0	**1**	**0**
x_1 1	**1**	**1**

x_1

$\overline{x_2}$

$$F = x_1 + \overline{x_2}$$

Como podemos comprobar, cuando simplificamos una función mediante los diagramas de Karnaugh, estamos aplicando de forma indirecta el ***Teorema de Idempotencia*** ya que, al hacer los agrupamientos tenemos la posibilidad de que un término mínimo participe en más de un agrupamiento. Obsérvese cómo el término mínimo $x_1\;\overline{x_2}$ participa en los dos agrupamientos.

✦ ✦ ✦

P.1.12: ¿Por qué se diseñan circuitos con sólo puertas NAND o sólo puertas NOR?

R.1.12: Un parámetro muy importante que hay que optimizar a la hora de implementar físicamente un circuito electrónico es el área que ocupa.

En Electrónica siempre se ha intentado miniaturizar los circuitos introduciendo el mayor número de dispositivos o circuitos integrados en el menor espacio posible. Pues bien, si observamos las hojas de características de estos dispositivos veremos que contienen varias puertas. Por ejemplo, en el caso de AND, OR, NAND y NOR de dos entradas, todos los dispositivos contienen 4 puertas (ver hojas de características correspondientes que se encuentran en el tema 1) y en el caso del inversor contiene 6 inversores. Por tanto, si diseñamos con distintos tipos de puertas, tenemos que usar varios tipos de dispositivos y es muy probable que no usemos todas las puertas que contienen dichos dispositivos. Esto nos lleva a un derroche de área.

Por el contrario, si diseñamos con sólo puertas NAND o NOR utilizaremos menos dispositivos, aprovechando mejor sus puertas y, por consiguiente, el área que ocupa el circuito es menor. Por ejemplo, si consideramos el circuito de la figura 1.24 del texto vemos que nos hacen falta tres dispositivos y nos sobran 3 inversores, 2 puertas OR y 3 puertas AND, mientras que si diseñamos con sólo puertas NOR (figura 1.25) sólo usaremos 2 dispositivos (de uno utilizaremos las 4 puertas NOR y del otro sólo 2) ahorrándonos, por tanto, el área de un dispositivo.

✧ ✧ ✧

P.1.13: ¿Cual es procedimiento para pasar de una función lógica expresada con distintos tipos de operadores lógicos (AND, OR, NOT, NAND, etc) a su expresión con sólo la función NAND ó sólo la función NOR?

R.1.13: Los pasos a seguir para expresar una función con sólo operadores NAND son:

1º. Obtener la expresión mínima como suma de productos.

2º. Complementar la expresión resultante dos veces, lo que es equivalente a dejarla tal cual.

3º. Aplicar reiterativamente los teoremas de De Morgan hasta obtener la función expresada sólo con variables negadas y con productos negados.

Veámoslo a través de un ejemplo:

Sea la función lógica: $f = XY\bar{Z} + XY + \bar{X}Z$

1º. Minimizamos: $f = XY\bar{Z} + XY + \bar{X}Z = XY(\bar{Z} + 1) + \bar{X}Z = XY + \bar{X}Z$

puesto que $(\bar{Z} + 1) = 1$

2º. Negamos dos veces: $f = XY + \bar{X}Z = \bar{\bar{f}} = \overline{\overline{XY + \bar{X}Z}}$

3º. Aplicamos De Morgan reiteradamente: $f = \overline{\overline{XY + \bar{X}Z}} = \overline{\overline{XY}\;\overline{\bar{X}Z}}$ (aunque en este caso sólo ha hecho falta aplicar De Morgan una vez para obtener la expresión con sólo operadores NAND).

Recordemos que la expresión de una variable negada con sólo NAND es la función NAND de la variable consigo misma. Es decir, $\bar{X} = \overline{X\,X}$

- Para expresar una función con ***sólo operadores NOR*** el procedimiento es análogo, sólo que detenemos la aplicación reiterada de los teoremas de De Morgan cuando tengamos la expresión expresada sólo con variables negadas y con sumas negadas.

Veamos ahora la forma de representar la misma función con sólo puertas NOR.

Los 3 pasos anteriores son comunes. Por tanto, partimos de la expresión:

$$f = \overline{\overline{XY}\;\;\overline{\overline{X}Z}}$$

Si seguimos aplicando De Morgan reiteradamente obtenemos:

$$f = \overline{\overline{XY}\;\;\overline{\overline{X}Z}} = \overline{\left(\overline{X} + \overline{Y}\right)\left(X + \overline{Z}\right)} = \overline{\left(\overline{X} + \overline{Y}\right)} + \overline{\left(X + \overline{Z}\right)}$$

Como podemos observar, la función $\boldsymbol{f}$ está expresada como la función OR (no NOR) de dos funciones NOR. Por lo tanto, como tenemos que expresarla en función sólo de NOR, tendremos que ver la forma de pasar de la OR a la NOR. Esto se consigue directamente negando la expresión dos veces. Así, en este caso conseguimos la función NOR de la NOR de dos funciones NOR. Es decir:

$$f = \overline{\left(\overline{X} + \overline{Y}\right)} + \overline{\left(X + \overline{Z}\right)} = \overline{\overline{\overline{\left(\overline{X} + \overline{Y}\right)} + \overline{\left(X + \overline{Z}\right)}}}$$

P.1.14: En el simulador sólo encuentro la puerta OR de 2 entradas SN7432 ¿cómo puedo conseguir puertas OR de más entradas?

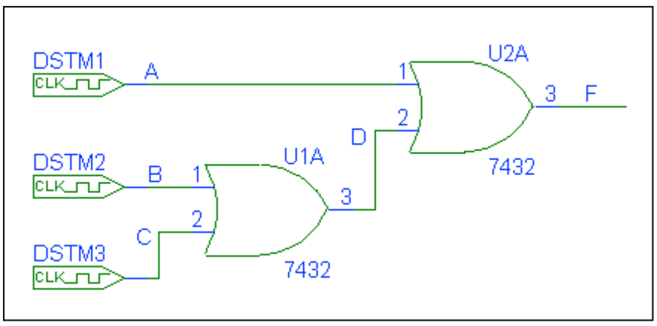
R.1.14: Veamos primero la solución operando sobre las expresiones lógicas:

Partimos, por ejemplo, de la función OR de 3 entradas: $F = A + B + C$

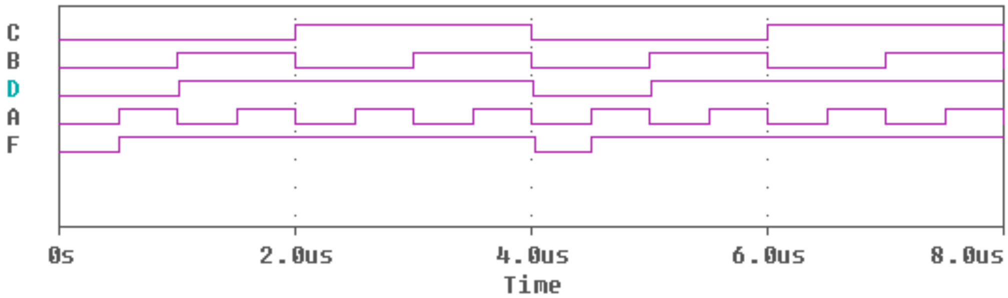
Esta función la podemos representar como:

$$F = A + (B + C) = A + D, \text{ siendo } D = B + C$$

Por tanto, el circuito resultante es:



El cronograma resultante de la simulación es:



A partir de este cronograma obtenemos la siguiente tabla de verdad práctica:

C B	D	A	F
0 0	0	0	0
0 0	0	1	1
0 1	1	0	1
0 1	1	1	1
1 0	1	0	1
1 0	1	1	1
1 1	1	0	1
1 1	1	1	1

Si seleccionamos los términos mínimos de las variables A y D que hacen que F sea igual a “1” y operamos, obtenemos:

$$F = \overline{D}\,A + D\,\overline{A} + D\,A = A + D$$

Análogamente, si seleccionamos los términos mínimos de las variables C y B que hacen que D sea igual a “1” y operamos, obtenemos:

$$D = \overline{C}\,B + C\,\overline{B} + C\,B = C + B$$

Si sustituimos en F la expresión obtenida para D resulta:

$$F = A + D = A + (C + D) = A + C + D$$

De igual forma operaríamos para el caso de una puerta OR de más entradas.

P.1.15: ¿Cómo podemos demostrar los teoremas de De Morgan?

R.1.15: Los teoremas de DeMorgan nos dicen que:

El negado de la suma lógica de n variables es igual al producto lógico de las n variables negadas. Análogamente, *el negado del producto lógico de n variables es igual a la suma lógica de las n variables negadas*.

Para el caso de dos variables la expresiones lógicas de estos teoremas son:

$$\overline{A+B} = \overline{A}\,\overline{B}$$
$$\overline{A\,B} = \overline{A} + \overline{B}$$

La demostración del primer teorema se encuentra en la pag. 31, fig. 1.14 del texto.

El segundo teorema lo podemos demostrar, por ejemplo, a través de la tabla de verdad. Así,

A	B	$A\,B$	$\overline{A\,B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Como podemos comprobar las columnas 4ª y 7ª coinciden. Luego, $\overline{A\,B} = \overline{A} + \overline{B}$

P.1.16: En el ejercicio de la Pag.31 del texto no concuerdan el valor de la frecuencia de la señal Y (fY=2MHz) y el periodo que aparece en el cronograma de la figura 1.15 (TY=2us). ¿Es una errata?

R.1.16: Si, es una errata.

En la línea (-2) donde dice: "… y para Y una frecuencia de 2 MHZ…"

debe decir: "… y para Y una frecuencia de 0.5 MHZ…"

P.1.17: ¿No veo muy claro cómo funciona el circuito de la Función Universal para dos variables de la figura 1.16 (pag.35) del texto?

R.1.17: Con 2 variables (x_1 y x_2) tenemos 2^2=4 términos mínimos $\left(\overline{x_1}\,\overline{x_2},\ \overline{x_1}\,x_2,\ x_1\,\overline{x_2},\ x_1\,x_2\right)$ y con estos 4 términos mínimos podemos obtener 2^4=16 funciones lógicas.

La función Universal es una función en la que cada término mínimo es controlado por un bit de la señal de control, "*a*" (palabra de 4 bits: $a_3\,a_2\,a_1\,a_0$), el cual puede tomar el valor "0" ó "1".

La figura 1.16 muestra la implementación de dicha función universal con puertas y mediante los términos mínimos. Su funcionamiento es el siguiente: Si un bit de la palabra "*a*" toma el valor "1" el término mínimo que es controlado por ese bit participa en la función de salida de la función universal y si toma el valor "0", no participa. Así, la función que realiza el circuito sobre x_1 y x_2 , y que será la que aparezca en la salida, *f*, va a ser una u otra dependiendo del valor de la palabra "*a*". Por tanto, puede ocurrir que no participe ningún término mínimo, que participe sólo uno de los cuatro, que participen dos, tres o los cuatro términos mínimos, lo que supone 16 casos posibles correspondientes a los 16 valores que puede tomar una palabra de 4 bits, ya que puede tomar cualquier valor comprendido entre la palabra 0000 y 1111.

Por ejemplo, si la palabra "a" toma el valor 0000 la función resultante es:

$$f = 0\,\overline{x_1}\,\overline{x_2} + 0\,\overline{x_1}\,x_2 + 0\,x_1\,\overline{x_2} + 0\,x_1\,x_2 = 0$$

Pero, si la palabra es 0001 la función que realiza es:

$$f = 0\,\overline{x_1}\,\overline{x_2} + 0\,\overline{x_1}\,x_2 + 0\,x_1\,\overline{x_2} + 1\,x_1\,x_2 = x_1\,x_2$$

Es decir, realiza la función AND entre las variables de entrada x_1 y x_2.

Análogamente, si la palabra de control, "*a*", toma el valor 1001, la función que realiza es:

$$f = 1\,\overline{x_1}\,\overline{x_2} + 0\,\overline{x_1}\,x_2 + 0\,x_1\,\overline{x_2} + 1\,x_1\,x_2 = \overline{x_1}\,\overline{x_2} + x_1\,x_2 = \overline{x_1 \oplus x_2}$$

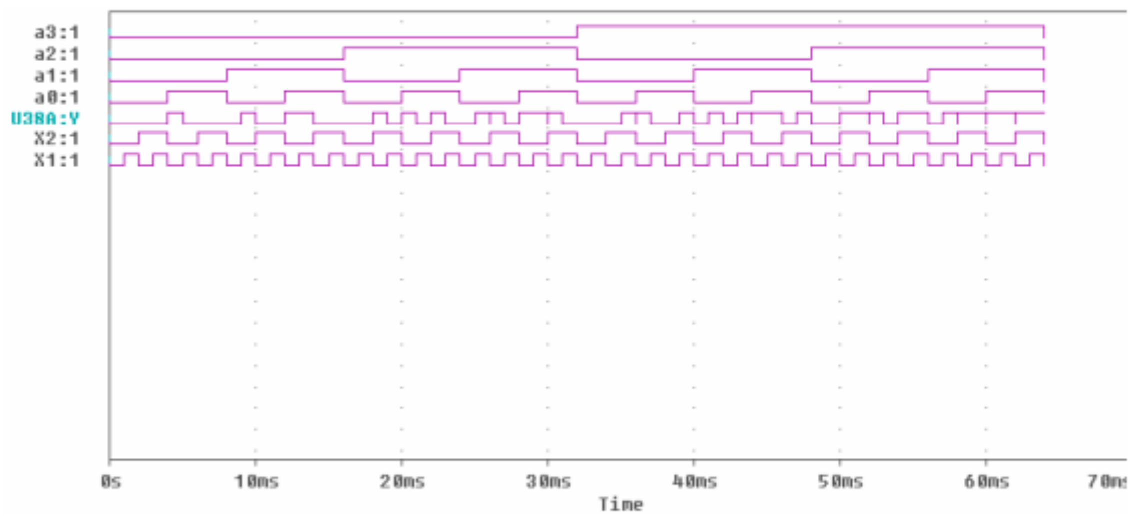
De igual modo podemos calcular cada una de las 16 funciones posibles, hasta llegar a la palabra "*a*" = 1111 cuya respuesta es:

$$f = 1\,\overline{x_1}\,\overline{x_2} + 1\,\overline{x_1}\,x_2 + 1\,x_1\,\overline{x_2} + 1\,x_1\,x_2 = 1$$

✦✦✦

P.1.18: Una vez que he simulado la función Universal mediante términos mínimos y obtengo el cronograma, ¿cómo puedo verificar que el circuito funciona correctamente?

R.1.18: Hemos simulado la función Universal con términos mínimos y hemos obtenido el cronograma que se muestra a continuación.



El siguiente paso es comprobar que el circuito implementado realiza las operaciones para las que ha sido diseñado. Para ello debemos comprobar que ante cada una de las palabras de programación $a(a_3\ a_2\ a_1\ a_0)$, y para las 4 configuraciones de las variables de entrada (x_1 y x_2), realiza la función que le corresponde.

Para ello, construimos la tabla de verdad, a partir del cronograma obtenido, viendo los valores que toma la salida, y, ante las distintas configuraciones de las entradas (datos, x_j, y control, a_i,)

Así, si nos fijamos en los primeros cuatro valores de la salida, y, que se corresponden con los valores de los coeficientes de control $a_3=a_2=a_1=a_0=0$, tenemos:

$a_3\ a_2\ a_1\ a_0$	$x_2\ x_1$	y
0000	00	0
"	01	0
"	10	0
"	11	0

Luego: $y=f_0=0$

Veamos ahora los valores de las siguientes cuatro salidas correspondientes a:

$a_3=a_2=a_1=0$ y $a_0=1$.

$a_3\ a_2\ a_1\ a_0$	$x_2\ x_1$	y
0001	**00**	**1**
"	01	0
"	10	0
"	11	0

Por tanto, $y = f_1 = \overline{x}_2\overline{x}_1$

Análogamente, para: $a_3=a_2=a_0=0$ y $a_1=1$, resulta:

$a_3\ a_2\ a_1\ a_0$	$x_2\ x_1$	y
0010	00	0
"	**01**	**1**
"	10	0
"	11	0

Por tanto, $y = f_2 = \overline{x}_2 x_1$

Así seguiremos hasta completar las 16 funciones lógicas posible que se pueden obtener con las dos variables de entrada.

Observe cómo la última, para $a_3=a_2=a_1=a_0=1$, la salida es: $y = f_{15} = 1$.

✦✦✦

P.1.19: ¿Me podrían explicar el fundamento de los diagramas de Karnaugh?

R.1.19: La distribución de los términos mínimos en los mapas o diagramas de Karnaugh no es arbitraria. En esta asignatura todo tiene su explicación "lógica", todo tiene que encajar y se explica de forma razonada. Por lo tanto, ni tiene sentido ni es recomendable aprenderse las cosas de memoria.

Recuerde que estos mapas se basan en el Teorema de la Adyacencia en el que, cuando tenemos dos términos mínimos que sólo se diferencian en el valor que toma un bit, podemos sacar factor común el producto de todos los demás y queda multiplicado por la suma del bit que es diferente y de su negado. Como esta suma es la unidad, el resultado final es el producto de las demás variables, las que se han sacado factor común.

Con esta premisa, si se observa las cabeceras de las filas y las columnas del diagrama de Karnaugh se ve que entre la primera fila que está encabezada con $\bar{A}\bar{B}$ y la segunda, encabezada con $\bar{A}B$, sólo hay un bit diferente, el B, y entre ambas palabras se puede sacar factor común $\bar{A}$ obteniéndose $\bar{A}\left(\bar{B}+B\right)=\bar{A}$. Lo mismo ocurre entre la segunda y la tercera columna encabezadas con $\bar{A}B$ y AB, sólo se diferencian en el valor del bit A resultando la expresión simplificada B. Lo mismo ocurre con el resto de las filas. Pero es más, si comparamos la primera fila y la última nos encontramos que también se diferencia en un único bit ya que tenemos $\bar{A}\bar{B}$ y $A\bar{B}$ y, por tanto, dan lugar a $\bar{B}$.

Si analizamos, de igual forma, las cabeceras del resto de las columnas vemos que ocurre lo mismo, las palabras están colocadas de forma que son adyacentes y entre dos contiguas sólo se diferencian en el valor que toma un único bit, cumpliéndose también que la última columna, encabezada con $C\bar{D}$, es adyacente con la primera, encabezada por $\bar{C}\bar{D}$.

Si la primera fila es adyacente con la última y la primera columna con la última podemos imaginarnos que las unimos en el espacio y obtenemos una esfera, la de la figura 1.39 del texto.

AB \ CD		$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$	00	0	1	3	2
$\bar{A}B$	01	4	5	7	6
AB	11	12	13	15	14

$A\overline{B}$	**10**	8	9	11	10

Con cuatro bits podemos obtener $2^4 = 16$ términos mínimos, desde el 0000 (0 en decimal) al 1111 (15 en decimal). Así, cada casilla de la tabla la numeramos con el decimal equivalente de la palabra que le corresponde. Es decir, la primera casilla (primera fila, primera columna) corresponde a la palabra, $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$, o sea 0000 (0 en decimal) que es el número pequeñito que ponemos en la esquina inferior derecha de la casilla. Análogamente, la casilla correspondiente a la primera fila y segunda columna la numeramos con un 1, que es el decimal correspondiente a la palabra $\overline{A}\,\overline{B}\,\overline{C}\,D$ (0001=1 en decimal), la casilla siguiente (primera fila y tercera columna) corresponde a $\overline{A}\,\overline{B}\,C\,D$ y por tanto la numeramos con 3 en decimal, y así sucesivamente.

Observe ahora cómo cada una de las casillas se diferencia de sus 4 vecinas (la de encima, la de abajo, la de su derecha y la de su izquierda) en un solo bit. Así, por ejemplo, las cuatro vecinas de la casilla numerada con 13 (correspondiente a $AB\,\overline{C}\,D$) son: 5 ($\overline{A}\,B\,\overline{C}\,D$, que se diferencia del 13 en la variable A), 9 ($A\,\overline{B}\,\overline{C}\,D$, que se diferencia del 13 en la variable B), 12 ($AB\,\overline{C}\,\overline{D}$, que se diferencia del 13 en la variable D), y 15 ($AB\,C\,D$,que se diferencia del 13 en la variable C), que son las cuatro palabras posibles cada una de las cuales se diferencian de ella en un sólo bit.

Con esto, espero que quede clara la estructura de la tabla, el porqué no se enumeran las casillas de forma consecutivas y que ahora resulte fácil entender su funcionamiento, cuando agrupamos las casillas contiguas formando agrupamientos cuadrados o rectangulares para eliminar variables y minimizar las expresiones, ya que en la propia estructura de la tabla está implícitamente aplicado el teorema de la adyacencia.

✧ ✧ ✧

P.1.20: ¿Podrían explicarme cómo se resuelve el caso c) del Ejercicio de la pag. 41 del texto?

R.1.20: Veamos distintas formas de obtener la solución al Caso c) del ejercicio de la pag. 41. Partimos de $f_m^c = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 + x_1\overline{x}_2$

1. Si usamos el Teorema de Idempotencia (x+x=x) podemos duplicar, por ejemplo, el primer sumando y la función sigue siendo la misma. Por tanto,

$$f_m^c = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 + x_1\overline{x}_2 = \left(\overline{x}_1\overline{x}_2 + \overline{x}_1\overline{x}_2\right) + \overline{x}_1 x_2 + x_1\overline{x}_2$$

Ahora podemos sacar factor $\overline{x}_1$ del primer y tercer sumando y $\overline{x}_2$ del segundo y cuarto sumando. Llegando a la expresión:

$$f_m^c = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 + \overline{x}_1 x_2 + x_1\overline{x}_2$$

$$f_m^c = \overline{x}_1\left(\overline{x}_2 + x_2\right) + \overline{x}_2\left(\overline{x}_1 + x_1\right)$$

Si aplicamos ahora el Postulado de la Complementariedad: $\overline{x}_2 + x_2 = 1$ y $\overline{x}_1 + x_1 = 1$

Resulta: $f_m^c = \overline{x}_1 \cdot 1 + \overline{x}_2 \cdot 1 = \overline{x}_1 + \overline{x}_2$

2. Otra forma de obtenerla es representando la función mediante los términos mínimos. Es decir, podemos poner $f_m^c = m_0 + m_1 + m_2$. Como podemos observar, sólo falta entre los sumandos de la función f_m^c el término mínimo m_3. Por tanto, podemos poner que $\overline{f_m^c} = m_3$ (el complementario de $f_m^c = m_0 + m_1 + m_2$ ya que $f_m^c + \overline{f_m^c} = m_0 + m_1 + m_2 + m_3 = 1$).

Si ahora complementamos y aplicamos De Morgan, obtenemos:

$$f_m^c = \overline{m_3} = \overline{x_1 x_2} = \overline{x}_1 + \overline{x}_2$$

Como podemos ver hemos obtenido la misma expresión.

3. Análogamente, otra forma de resolverlo es desarrollando la expresión por maxterms y así obtenemos:

$$f_M^c = M_3 = \overline{x}_1 + \overline{x}_2$$

De nuevo, llegamos a la misma expresión.

4. Por último, si minimizamos por Karnaugh tendremos

x_2 / x_1	0	1
0	1	1
1	1	0

$\overline{x}_1$ $\overline{x}_2$

Luego, $f_m^c = \overline{x}_1 + \overline{x}_2$

✦✦✦

P.1.21: ¿Qué son los términos indiferentes y cómo se usan?

R.1.21: Los términos indiferentes son aquellos que no intervienen en la función por lo que consideramos que pueden tomar valor "0" ó "1" y, consecuentemente, no nos preocupa el valor que toman. Como nos da igual si es "0" ó "1" lo solemos nombrar, por ejemplo, con una "d" ó una "x". Este es el motivo por el que al minimizar por Karnaugh los usamos para que intervengan en los grupos de términos mínimos con el fin de obtener funciones mínimas, bien con los "1" ó bien con los "0" dependiendo de la forma en la que nos interese hacer el diseño. Sin embargo, nunca podemos hacer un grupo con sólo términos indiferentes.

P.1.22: ¿Sería posible ver un ejemplo concreto en el que se usaran los términos indiferentes para minimizar?

R.1.22: Supongamos que queremos diseñar un sistema muy sencillo para controlar la sirena de la alarma de un apartamento. El sistema de la alarma cuenta con 4 sensores: uno en la puerta de entrada, otro en el dormitorio, otro en el salón y el cuarto en la puerta de la terraza del salón.

Queremos que la sirena de la alarma funcione de la siguiente forma:

1. La sirena de la alarma no debe sonar si no se activa ningún sensor, o si sólo se activa un único sensor, o si se activan muy próximos en el tiempo (podemos considerarlo a la vez) el del dormitorio y el del salón sin haberse activado ninguno de los de las puertas, o si sólo se activan el del dormitorio y el de la puerta de la terraza.
2. La sirena de la alarma debe sonar siempre que se active "a la vez" el sensor de una de las dos puertas y el del salón, o bien el de la puerta de entrada y el del dormitorio.
3. Consideramos que nunca se van a presentar el resto de los casos. Es decir, que nunca se van a activar "a la vez" los sensores de las dos puertas, con independencia de que se dispare o no alguno de los otros dos sensores.

Lo primero que tenemos que hacer es definir las variables de entrada y de salida y asignarle un significado a los dos valores que toma cada variable. Así, las variables de entrada son las señales de los sensores del sistema de alarma, a las que llamaremos: P a la del sensor de la puerta, T a la del sensor de la terraza, D a la del dormitorio y S a la del salón. Como variable de salida tenemos la sirena de la alarma, a la que vamos a llamar SA.

Cada una de estas variables, tanto las de entrada como la de salida, pueden tomar dos valores, "0" ó "1". Para las variables de entrada, que son las señales de los sensores, consideraremos que el "0" corresponde a que el sensor no se ha activado y el "1" a que se ha activado porque ha detectado el movimiento de una persona. Para la variable de salida del sistema, SA (sirena de la alarma), consideraremos que cuando se activa toma el valor 1 y suena la alarma y cuando no se activa toma el valor 0 y no suena. Todo esto es por convenio, podríamos haber considerado lo contrario, pero debemos especificarlo claramente.

En resumen, tenemos:

Variables de entrada: P, T, D, S (1 = activada, 0 = no activada)

Variable de salida: SA (1 = Sirena sonando, 0 = Sirena en silencio)

Como tenemos 4 variables de entrada, tendremos $2^4=16$ configuraciones de entrada. Sólo necesitamos una variable de salida, que tomará el valor “0”, “1” ó “d” (indiferente). En realidad esta “d” tomará un valor, pero a nosotros lo mismos nos da porque, por lo general, lo que ocurre es que son configuraciones que no se van a presentar.

Pasemos ahora a construir la tabla de verdad del funcionamiento del sistema. Para ello, transcribimos las condiciones especificadas en los tres apartados de la descripción del sistema. Así hemos matizado en amarillo las condiciones que producen “0” en la salida VA, en verde las que producen un “1” y en azul las que son indiferentes.

P	***T***	***D***	***S***	***SA***
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d

Para obtener la función mínima de SA usaremos los diagramas de Karnaugh. Así,

DS / ***PT***	**00**	**01**	**11**	**10**
00	**0**	**0**	**0**	**0**
01	**0**	**1**	**1**	**0**
11	**d**	**d**	**d**	**d**
10	**0**	**1**	**1**	**1**

TS *PS* *PD*

Agrupando las celdas enmarcadas en rojo, azul y verde, obtenemos la expresión mínima de la función lógica que disparará a la alarma, SA. Así,

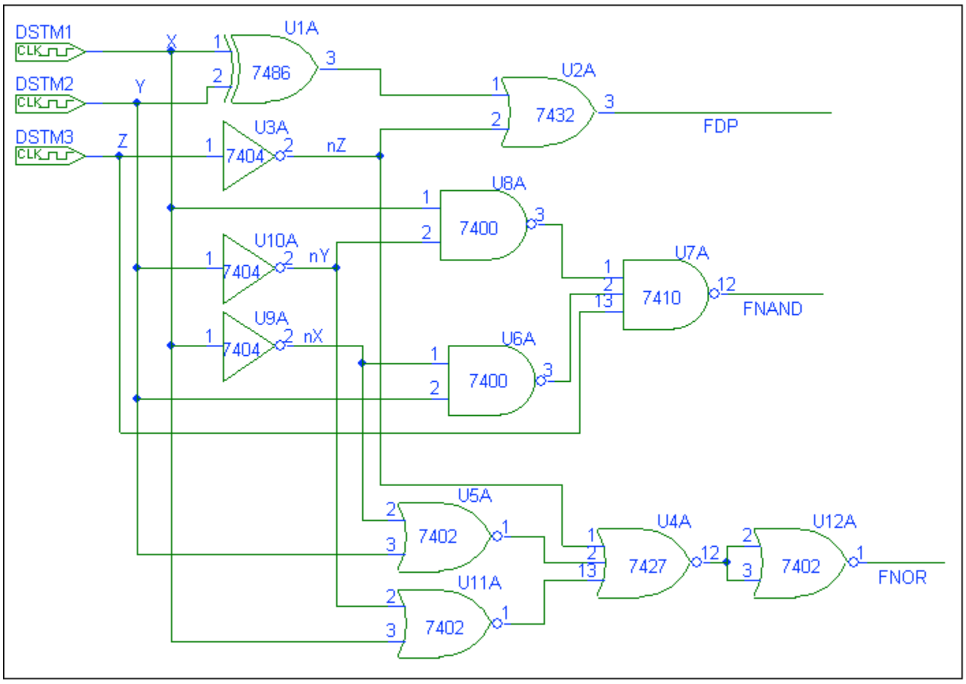
SA = TS+PS+PD

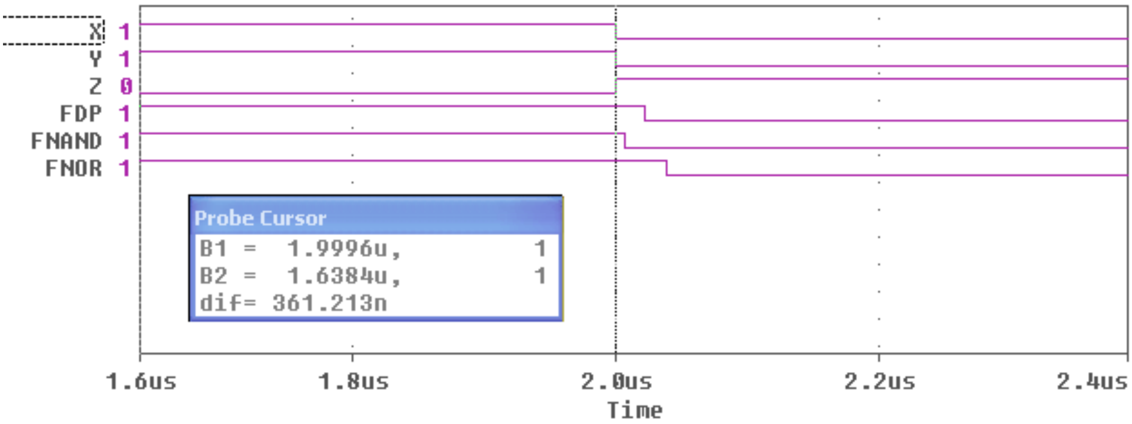
Que traducido al lenguaje natural equivale a decir: “la alarma sonará siempre que se active el sensor del salón y el de alguna de las dos puertas, o el de la puerta de entrada y el del dormitorio”.

P.1.23: He implementado la función de la ecuación 1.63 del texto, una vez minimizada, con distintos tipos de puertas, con sólo puertas NAND y con sólo puertas NOR, y he simulado las tres en el PSpice, Las respuestas que obtengo en el cronograma de estos circuitos son cualitativamente iguales, pero veo que no conmutan a la vez, sino que hay un desfase entre las conmutaciones de las salidas de los tres circuitos. ¿A qué se deben estas pequeñas diferencias?

R.1.23: Se deben a que las puertas introducen un retardo y el retardo total existente en un circuito combinacional depende del tipo de puertas (cada tipo introduce un retardo distinto) y del número de niveles de puertas que tenga dicho circuito.

Si implementamos y simulamos las tres representaciones de la función minimizada de la ecuación 1.63 resulta:





Al ampliar el cronograma vemos que la configuración que introduce menos retardo es la realizada con puertas NAND (6.893ns). En esta representación tenemos tres niveles de puertas.

El siguiente retardo lo introduce la configuración realizada con distintos tipos de puertas (22.059ns). En este caso sólo tenemos dos niveles de puertas, pero hemos usado una puerta XOR que introduce mayor retardo (del orden de 15ns) que el resto de las puertas.

Finalmente, el mayor retardo lo introduce la configuración realizada con sólo puertas NOR (38,603ns).

De todas formas, en el simulador hay posibilidad de controlar el retardo que introducen las puertas. Para ello, en la ventana del "Analysis Setup" se selecciona la opción "Digital Setup" y en el cuadro "Timing Mode" se puede elegir la opción "Minimum". Observe que, por defecto, aparece el valor "Typical".