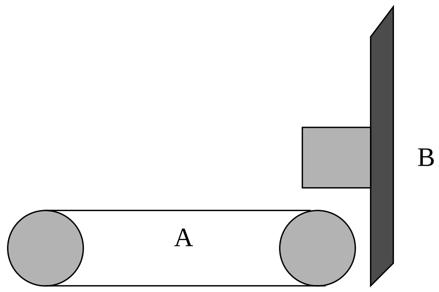
TEMA IV.- DISEÑO DE CIRCUITOS COMBINACIONALES.

Como ya hemos comentado, el problema del diseño o síntesis consiste en determinar un circuito que cumpla con unas determinadas especificaciones, tanto de comportamiento como de funcionalidad. Este problema se puede dividir en dos partes:

- Obtención de la función de conmutación que cumpla la funcionalidad deseada.
- Obtención de los circuitos que implementa dicha función con el comportamiento deseado.

La obtención de la función de conmutación a partir de unas especificaciones no tiene una metodología establecida sino que depende de la pericia del diseñador y de la precisión de las especificaciones dadas.

Así, por ejemplo, vamos a considerar el diseño del sistema mostrado en la figura 4.1. Se trata de un sistema que opere sobre el estado de operación de una línea de montaje que dispone de dos sensores: peso en la cinta (A), y fin de la cinta (B). El sistema debe ser tal que los motores deben actuar siempre y cuando haya algo en la cinta. La función de conmutación que cumpla estas especificaciones puede ser la mostrada en la misma figura, es decir, exista un peso sobre la cinta (A=1) y no haya llegado al final (B=0).



A	B	Y
0	0	0
0	1	0
1	0	1
1	1	0

Figura 4.1.- Ejemplo de un problema de diseño.

El problema de la obtención de un circuito que implementa la función de conmutación dada sí tiene unas metodologías bien establecidas. La realización de esta parte depende de los elementos con los que se implemente el circuito. Podemos distinguir los siguientes tipos de diseño:

- Diseño *semi-custom*.- Los bloques que disponemos serán puertas lógicas existentes en alguna librería disponible.

- Diseño MSI.- Los bloques disponibles serán grandes bloques de una complejidad media, como pueden ser bloques sumadores.
- Diseño programable.- Los bloques disponibles serán dispositivos programables en los que se programará la función deseada.
- Diseño *full-custom*.- No se dispondrá de ningún bloque, sino que todos los elementos necesarios, simples o complejos, deberán ser generados por el diseñador.

Cada uno de estos tipos será objeto de estudio en diferentes temas. Concretamente, en el presente tema estudiaremos el diseño *semi-custom*.

1. Introducción.

En el diseño *semi-custom*, podemos distinguir entre implementaciones en dos niveles o multiniveles, donde los niveles son el número máximo de puertas que están conectados en cascada (en los niveles no se suelen contar los inversores de entrada). La implementación en dos niveles es del tipo de suma de productos o de producto de sumas; en cambio, la implementación multinivel corresponde a una fórmula compleja en las que las operaciones AND y OR aparecen mezcladas.

La implementación en dos niveles tiene la ventaja de ser rápida, solamente el retraso de dos puertas; en cambio, como contraposición, las puertas deberán tener un mayor número de entradas, siendo más complejas y lentas. Mientras, la implementación multinivel tiene la ventaja de usar puertas más pequeñas y rápidas (por lo general), pero el circuito global será más lento al tener un retraso de más de dos puertas. En la figura 4.2 mostramos dos implementaciones de la misma función: en dos niveles y en multinivel.

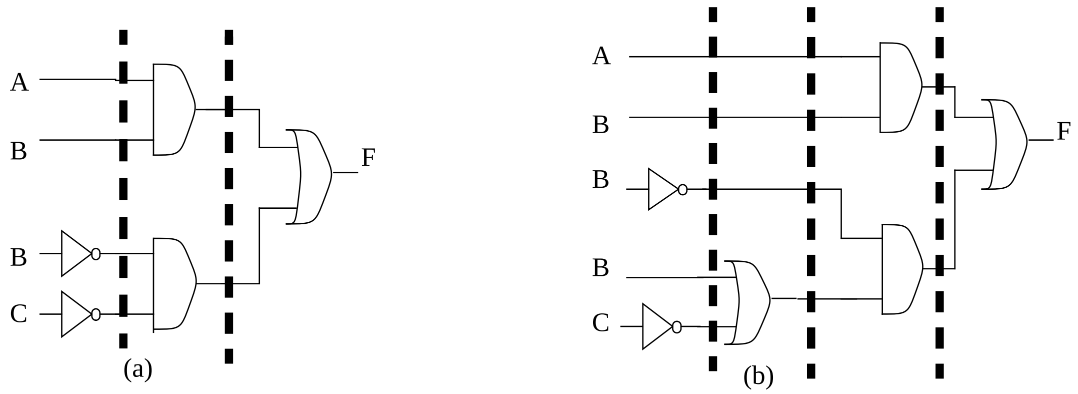


Figura 4.2.- Diferencia entre una implementación (a) en dos niveles y (b) multinivel.

La meta del diseñador consiste en una realización mínima de un circuito que implemente una determinada funcionalidad. Los criterios de minimalidad que deben seguir son los que vimos en el tema II, con el orden siguiente:

- Menor número de términos (que equivalen a puertas)
- Menor número de variables, incluso en los diferentes términos (que equivalen al número de entradas)

- Menor valor asociado.

La implementación mínima multinivel no tiene un método bien establecido y los que existen son muy complejos. Por lo tanto, lo que se suele hacer es:

- Dividir el diseño en varias partes, implementadas en dos niveles cada una, y unirlas posteriormente.
- Obtener una implementación en dos niveles y una vez que tengamos la fórmula, operar mediante el álgebra de Boole para unir los términos posibles.

Luego, la implementación en dos niveles suele ser la base para la implementación multinivel. Esta es la razón por la que nos vamos a centrar en este tipo de implementación, de dos niveles.

A la hora de plantearnos una fórmula mínima, debemos diferenciar dos casos de funciones: funciones con una salida y funciones con más de una salida. Esta diferenciación es debido a que una solución por separado de cada salida puede que no sea mínima si contamos con todas las salidas simultáneamente. En la figura 4.3 vemos dos funciones cuya implementación conjunta muestra un coste menor que su implementación por separado:

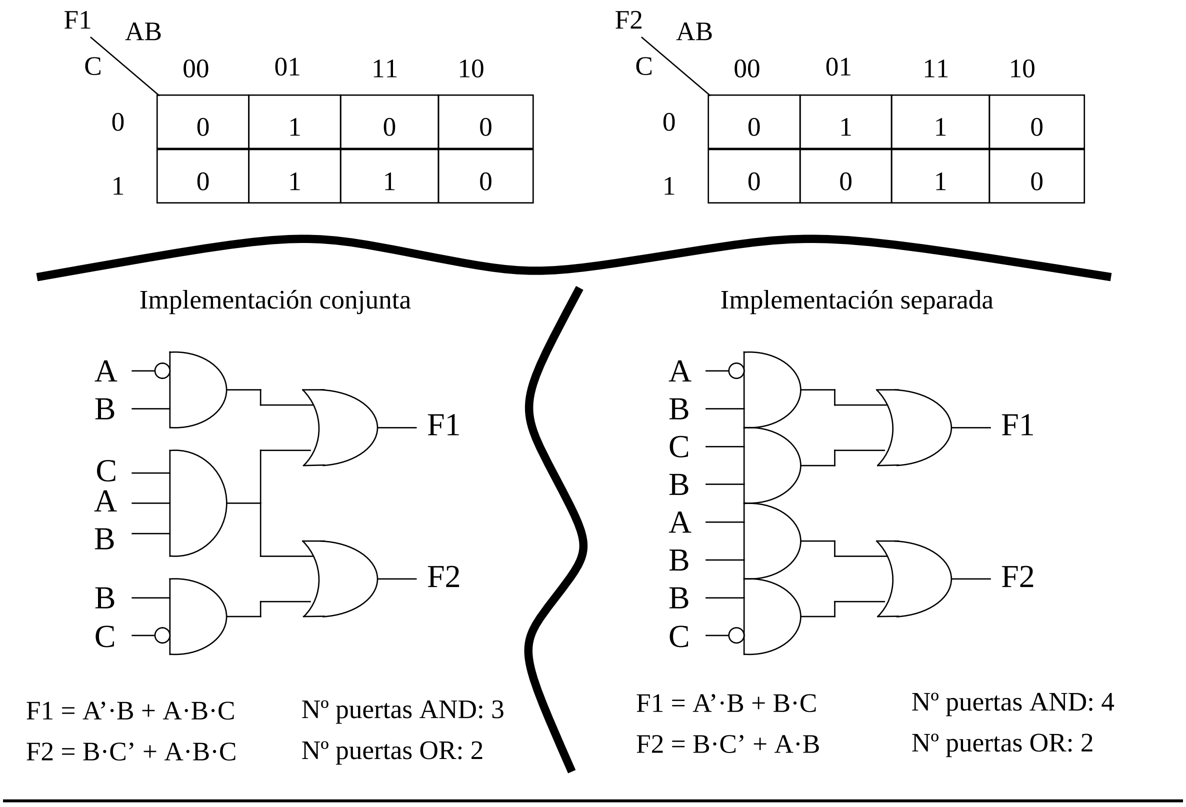


Figura 4.3.- Diferencia entre la implementación monosalida y multisalida.

Una vez que hemos visto la necesidad de realizar esta distinción, existen dos métodos para implementar funciones en dos niveles:

- Método del mapa.- suele ser usado para implementar funciones independientes, y por lo tanto funciones de una sola salida, basándose en los mapas de Karnaugh.
- Método de McCluskey.- suele ser empleado para implementar funciones multisalida, aunque también se puede usar para funciones con una sola salida.

2. Método del Mapa.

Este método se basa en el mapa de Karnaugh y el 4º postulado del álgebra de Boole. Si recordamos los mapas de Karnaugh, consiste en una representación tabular en dos dimensiones, tal que cada celda adyacente (geométricamente hablando) también es adyacente desde el punto de vista de la distancia de Hamming. Debido a la característica de que todas las celdas adyacentes tienen una distancia igual a la unidad (sólo cambia una variable), se pueden demostrar las siguientes igualdades:

Un grupo de 1's adyacentes en un número igual a una potencia de dos, es realizado por el término producto de las variables que no cambian de valor, tomándolas como sin complementar si valen '1' y complementadas si valen '0'.

Supongamos que tenemos un grupo de dos 1's adyacentes en los que la variable que cambia sea x_n. Los posibles mintérminos de dichos 1's podrían ser $x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_m$ y $x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_m$. Una posible fórmula de la función que tendría dichos 1's sería la suma de los mintérminos, por lo tanto:

$$F = x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_m = x_1 \cdot \ldots \cdot x_{n-1} \cdot (x_n + x_n') x_{n+1} \cdot \ldots \cdot x_m = x_1 \cdot \ldots \cdot x_{n-1} \cdot x_{n+1} \cdot \ldots \cdot x_m$$

es decir, sería el producto de todas las variables (complementadas o sin complementar) excepto la variable que ha realizado el cambio.

Supongamos ahora que el grupo es de cuatro 1's adyacentes en los que las variables que cambian son x_n y x_p. Los posibles mintérminos podrían ser $x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p' \cdot \ldots \cdot x_m$, $x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p' \cdot \ldots \cdot x_m$, $x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m$ y $x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m$. Si sumamos los mintérminos y operamos, obtenemos:

$$F = x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p' \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p' \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m =$$
$$= x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot (x_p' + x_p) \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot (x_p' + x_p) \cdot \ldots \cdot x_m =$$
$$= x_1 \cdot \ldots \cdot (x_n' + x_n) \cdot \ldots \cdot x_{p-1} \cdot x_{p+1} \cdot \ldots \cdot x_m = x_1 \cdot \ldots \cdot x_{n-1} \cdot x_{n+1} \cdot \ldots \cdot x_{p-1} \cdot x_{p+1} \cdot \ldots \cdot x_m$$

Luego, la afirmación anterior es verificada para los grupos de potencia de dos. Pero supongamos que tenemos tres 1's adyacentes. El número de variables que cambian son de nuevo de dos. Supongamos que los mintérminos son: $x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p' \cdot \ldots \cdot x_m$, $x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m$ y $x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m$, variando de nuevo x_n y x_p. Si realizamos las sumas y operamos según el álgebra de Boole obtenemos:

$$F = x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p' \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m =$$
$$= x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot (x_p' + x_p) \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m =$$
$$= x_1 \cdot \ldots \cdot x_n' \cdot \ldots \cdot x_{p-1} \cdot x_{p+1} \cdot \ldots \cdot x_m + x_1 \cdot \ldots \cdot x_n \cdot \ldots \cdot x_p \cdot \ldots \cdot x_m$$

Por lo tanto, los grupos deben tener un número de 1's igual a una potencia de dos, es decir, 1, 2, 4, 8, 16, 32, 64,...

Por una aplicación directa del principio de dualidad, obtenemos unas definiciones equivalentes para el caso de los 0's y de los términos sumas.

Un grupo de 0's adyacentes en un número igual a una potencia de dos, es realizado por el término suma de las variables que no cambian de valor, tomándolas sin complementar si valen '0' y complementadas si valen '1'.

Supongamos que tenemos un grupo de dos 0’s adyacentes en los que la variable que cambia sea x_n. Los posibles maxtérminos de dichos 0’s podrían ser $x_1+...+x_n+...+x_m$ y $x_1+...+x_n'+...+x_m$. Una posible fórmula de la función que tendría dichos 0’s sería el producto de los maxtérminos, luego:

$$F = (x_1+...+x_n+...+x_m)(x_1+...+x_n'+...+x_m) = x_1+...+x_{n-1}+x_{n+1}+...+x_m + x_n \cdot x_n' =$$
$$= x_1+...+x_{n-1}+x_{n+1}+...+x_m$$

es decir, sería la suma de todas las variables (complementadas o sin complementar) excepto la variable que ha realizado el cambio.

Para el resto de la demostración se opera siguiendo los mismo pasos que en el caso de los grupos de 1’s.

Veamos un ejemplo al respecto. En la figura 4.4 se muestra el mapa de Karnaugh de una función cuya fórmula de mintérminos es $F(A,B,C,D) = \sum m(1, 5, 7, 10, 11, 14, 15)$. Podemos apreciar que en dicha función podemos formar cuatro grupos de 1’s adyacentes, etiquetados I1, I2, I3 e I4.

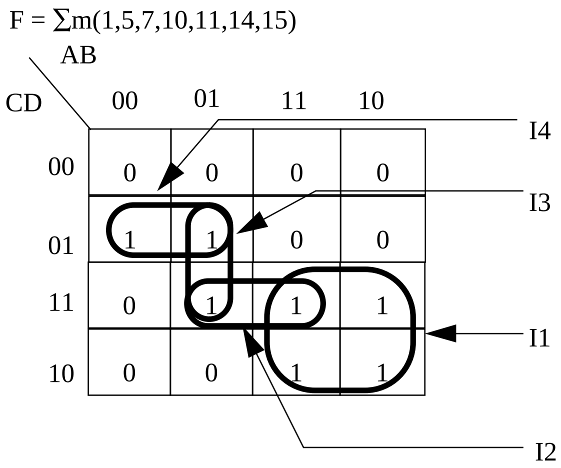


Figura 4.4.- Ejemplo de grupos de 1’s adyacentes en un mapa de karnaugh.

Dichos grupos se corresponderán con los siguientes términos producto (puesto que hemos agrupados los 1’s):

- $I1 = m(10) + m(11) + m(14) + m(15) = A \cdot B' \cdot C \cdot D' + A \cdot B' \cdot C \cdot D + A \cdot B \cdot C \cdot D' + A \cdot B \cdot C \cdot D$ $= A \cdot B' \cdot C \cdot (D+D') + A \cdot B \cdot C \cdot (D+D') = A \cdot B' \cdot C + A \cdot B \cdot C = A \cdot (B'+B) \cdot C = A \cdot C$
- $I2 = m(7) + m(15) = A' \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D = (A'+A) \cdot B \cdot C \cdot D = B \cdot C \cdot D$
- $I3 = m(5) + m(7) = A' \cdot B \cdot C' \cdot D + A' \cdot B \cdot C \cdot D = A' \cdot B \cdot (C+C') \cdot D = A' \cdot B \cdot D$
- $I4 = m(1) + m(5) = A' \cdot B' \cdot C' \cdot D + A' \cdot B \cdot C' \cdot D = A' \cdot (B'+B) \cdot C' \cdot D = A' \cdot C' \cdot D$

Por lo tanto, tenemos términos con dos entradas (el grupo I1) y con tres entradas (los grupos restantes). Como podemos comprobar, los términos con un número menor de entradas serán aquellos que más 1’s engloben, ya que habrá más señales de entrada que cambien (con lo que no influirán en dicho término). Un posible grupo formado por los mintérminos m(10) y m(11) no será considerado ya que se encuentra completamente dentro del grupo I1.

En relación con estos grupos de 1’s (0’s), tenemos las siguientes definiciones:

*Un **implicante prima** o **implicante** es un grupo de 1’s (0’s) con un número igual a una potencia de dos que no está contenido totalmente en otro grupo de las mismas características.*

Si observamos en el ejemplo anterior, podemos ver que todos los implicantes son primas.

*Cuando algún ‘1’ (‘0’) es cubierto por un solo implicante, se dice que dicho implicante es **esencial**.*

Si observamos en el ejemplo anterior, podemos identificar que los implicantes I1 e I4 son esenciales, mientras que los I2 e I3 no son esenciales.

Una vez que se han definido los implicantes, podemos demostrar que

Teorema 4.1.- La fórmula mínima de la función será aquella compuesta por la unión de los implicantes esenciales y puede que de algunos no esenciales, de tal forma que cubramos toda la tabla.

Demostración.- En la fórmula deberán estar incluidos todos los 1’s (0’s) de la función. Luego, los implicantes que estarán presentes en la fórmula deben cubrir todos los 1’s (0’s). Por definición de implicante esencial, es el único implicante que cubre a algún 1 (0); por lo tanto, para que dicho 1 (0) esté involucrado en la fórmula, dicho implicante debe aparecer en ella. No obstante, puede que exista algún 1 (0) cubierto por implicantes no esenciales; luego puede que alguno de estos implicantes no esenciales deban aparecer en la fórmula para cubrir algún 1 englobado sólo por implicantes no esenciales.

Esta unión será a través de sumas, cuando estamos considerando los 1’s y los implicantes se corresponderán con términos productos; o través de productos, cuando estemos considerando los 0’s y los implicantes se corresponderán con términos sumas.

Si consideramos la función del ejemplo anterior, y consideramos la fórmula como suma de productos (consideraremos los 1’s), todas las fórmulas de dicha función serán:

$$f = I1 + I3 + I4 = a\cdot c + a'\cdot b\cdot d + a'\cdot c'\cdot d,$$

$$f = I1 + I2 + I4 = a\cdot c + b\cdot c\cdot d + a'\cdot c'\cdot d,$$

$$f = I1 + I2 + I3 + I4 = a\cdot c + b\cdot c\cdot d + a'\cdot b\cdot d + a'\cdot c'\cdot d$$

Las dos primeras soluciones tienen el mismo coste, siendo el mínimo, mientras que en la última puede eliminarse un implicante, I2 o I3, lo cual nos llevará a una implementación no mínima. Luego, según lo visto, podemos encontrarnos con fórmulas en las que podemos eliminar algún término y con otras en las que no se puede eliminar ningún término. Según las fórmulas podemos encontrar las siguientes definiciones:

*Una fórmula se denomina **irredundante** cuando no se puede eliminar ningún implicante sin que cambia la función lógica. En caso contrario se denominará **redundante**.*

Las dos primeras fórmulas serán irredundantes ya que si eliminamos algún implicante, la función lógica que representa cambiará. En cambio, la tercera fórmula es redundante porque se puede eliminar el implicante I2 o I3 sin que cambie la función lógica.

Una situacíón que no nos puede llevar a error consiste en que podemos obtener una fórmula irredundante no mínima. Por ejemplo las siguientes fórmulas

$$F1 = a'\cdot b + a'\cdot b'\cdot d + a\cdot b\cdot d$$

$$F2 = a'\cdot b + a'\cdot d + b\cdot d$$

son equivalentes y también son irredundantes. No obstante, queda claro que la fórmula F1 tiene un coste mayor debido a la utilización de puertas on un mayor número de entradas.

Los mapas de Karnaugh mantienen su peculiaridad (diferenciarse en una variable sólo para las celdas adyacentes) para las funciones con cuatro variables o menos. Si aumentamos este número, por ejemplo a cinco variables (figura 4.5), podemos ver que existen celdas con una distancia igual a 1 (y por lo tanto, se pueden unir en un mismo implicante) que no son adyacentes (desde un punto de vista geométrico). En el mapa de cinco variables mostrado existen columnas que se deben unir en un mismo implicante que no son adyacentes geométricamente. La columna 010 tendrá una distancia de Hamming de 1 con respecto a la columna 000, pero no son adyacentes, por lo que el implicante a'·c' no sería fácilmente distinguible por el método del mapa. Para resolver este problema, existen dos soluciones.

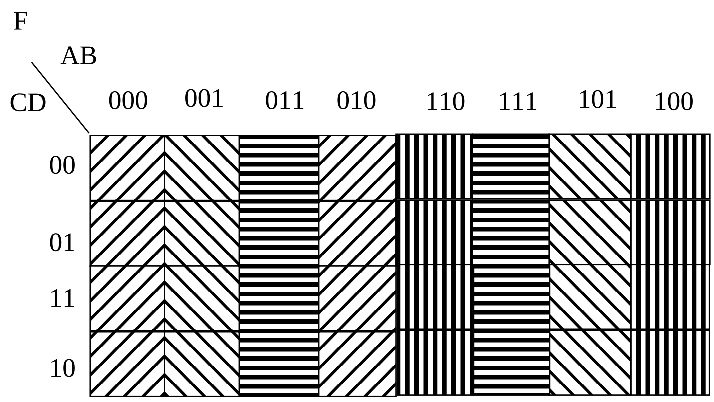


Figura 4.5.- Ejemplo de mapa de karnaugh de cinco variables.

La primera solución consiste en la aplicación de los teoremas de expansión: f(x5, x4, x3, x2, x1) = x5·f(1, x4, x3, x2, x1) + x5'·f(0, x4, x3, x2, x1), obteniendo los implicantes correspondientes a las funciones f(1, x4, x3, x2, x1) y f(0, x4, x3, x2, x1) que ya son de cuatro variables. Una vez que tengamos la fórmula escrita de esta manera, se reduce (si es posible) con el cuarto postulado. El resultado de esta forma depende ampliamente de la pericia del diseñador a la hora de la reducción. En el caso de funciones de un número mayor de variables, se aplica sucesivamente los teoremas de expansión hasta reducir la función a varias de cuatro variables.

Por ejemplo, consideremos la función F(x5, x4, x3, x2, x1)= $\sum$ m(0, 2, 8, 10, 30, 31). En la figura 4.6 mostramos los mapas de Karnaguh de dicha función, tanto en el mapa de cinco variables como aplicando el teorema de expansión sobre la variable x4

El teorema de expansión nos dice que F = x4·F(x4=1) + x4'·F(x4=0), es decir

$$F = x4 \cdot (x5\cdot x3\cdot x2 + x5'\cdot x3'\cdot x1') + x4 \cdot (x5'\cdot x3'\cdot x1')$$

Y pasamos a reducir la fórmula utilizando los diferentes postulados del álgebra de Boole, por lo que la función queda:

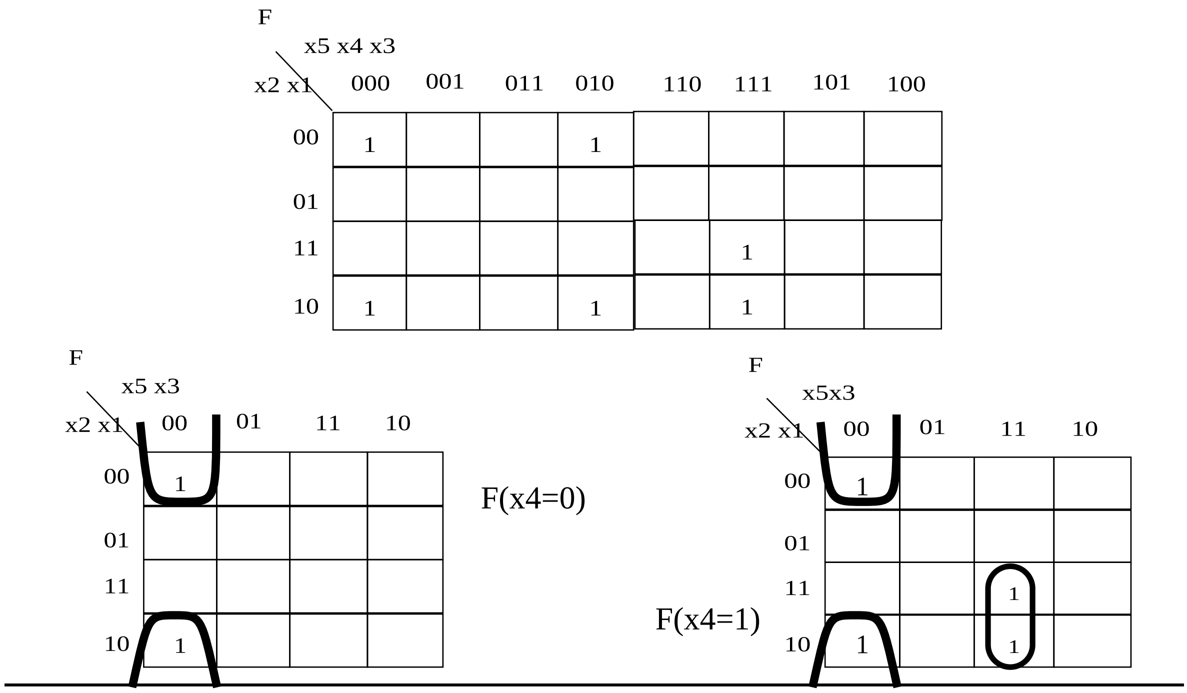


Figura 4.6.- Mapas de Karnaugh de la función
F(x5, x4, x3, x2, x1)= $\sum$ m(0, 2, 8, 10, 30, 31)

F = x5·x4·x3·x2 + x5'·x3'·x1'

La segunda forma consiste en la obtención de las implicantes primas por métodos tabulares. Uno de estos métodos es el denominado método por carácter binario. Consiste en poner la combinación de todos los mintérminos que se encuentran en la función. Los mintérminos se agrupan por el número de 1's. Se unen aquellos que sólo se diferencian en una variable (por lo que sólo hay que mirar en los grupos adyacentes), eliminando dicha variable. Este procedimiento se repite mientras se pueda. Los implicantes primas serán aquellos que no se han podido repetir, de tal forma que el producto estará formado únicamente por las variables que queden de la misma forma que en el método del mapa (complementada si su valor es 0, o sin complementar si su valor es 1, en el caso de considerar los 1's).

Veamos el ejemplo para la función f = Σ m(0, 2, 8, 10, 30, 31), cuya ejecución se muestra en la figura 4.7. Al agrupar los mintérminos por el número de 1's, nos han quedado 5 grupos diferentes (A={0}, B={2,8}, C={10}, D={30}, E={31}). Estos grupos son agrupados con sus adyacentes según el número de 1's, tal que nos quedan tres grupos: A1={(0,2), (0,8)}, B1={(2,10), (8,10)} y C1={(30,31)}. Tenemos que notar que los grupos C y D no se pueden agrupar ya que no son adyacentes (el grupo C tiene dos 1's y el D tiene 4 1's). En los agrupamientos, la variable que ha sufrido el cambio se elimina, por lo que colocamos en su posición un guión. Para el nuevo agrupamiento, los mintérminos a agrupar solamente deben tener un valor diferente (incluyendo ceros, unos y guiones), por lo tanto, también debemos tener en cuenta que no todos los elementos de grupos adyacentes pueden ser agrupados (como (0,2) y (2,10)). Si realizamos un nuevo agrupamiento nos quedan dos grupos, A2={(0,2,8,10)} y B2={(30,31)}. Como podemos observar, ya no se puede realizar ningún agrupamiento más. Por lo tanto, los implicantes obtenidos para esta función han sido dos: x5'·x3'·x1' y x5·x4·x3·x2.En cambio, observando simplemente el mapa de Karnaugh de la figura 4.6, el primer implicante no se apreciaría.

	x5	x4	x3	x2	x1
0	0	0	0	0	0
2	0	0	0	1	0
8	0	1	0	0	0
10	0	1	0	1	0
30	1	1	1	1	0
31	1	1	1	1	1

	x5	x4	x3	x2	x1
0,2	0	0	0	--	0
0,8	0	--	0	0	0
2,10	0	--	0	1	0
8,10	0	1	0	--	0
30,31	1	1	1	1	--

	x5	x4	x3	x2	x1
0,2,8,10	0	--	0	--	0
30,31	1	1	1	1	--

Figura 4.7.- Obtención de los implicantes de una función de cinco variables mediante el método tabular

Hasta ahora sólo hemos considerado funciones completas. Para el caso de las funciones incompletas, las inespecificaciones se tratan para aumentar el tamaño de los implicantes. Si las especificaciones no pueden aumentarlos, no se tienen en cuenta.

Veamos el ejemplo de la figura 4.8. Las inespecificaciones en los mintérminos 3 y 6 nos han servido para pasar de tres implicantes de tres variables a dos implicantes de dos variables. En cambio, la inespecificación 12 no nos sirve para aumentar ningún implicante, por lo que no se considera. En la nueva función, se puede observar que los implicantes I1 y I3 son esenciales, mientras que el I2 no. En este caso, la función mínima sólo es la suma de los dos implicantes esenciales, es decir, f = a·c + a'·d.

$F = \sum m(1,5,7,10,11,14,15) + \phi\ (3, 6, 12)$

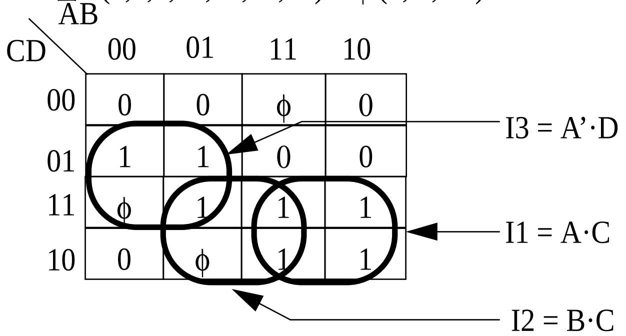
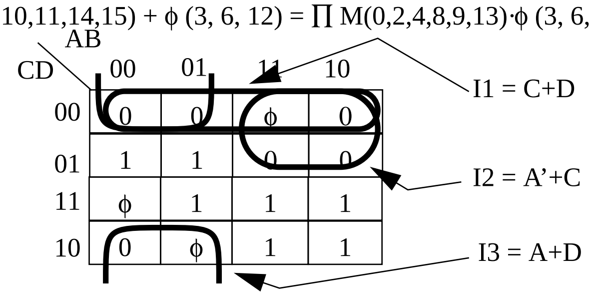


Figura 4.8.- Ejemplo de diseño de una función incompleta utilizando el método del mapa.

Si en lugar de querer una implementación de suma de productos, queremos una implementación de producto de sumas, se opera de la misma forma pero con los ceros y los términos sumas. Como recordaremos, la única diferencia para pasar del término suma a la operación OR con respecto al paso del término producto a la operación AND, consiste en que ahora la variable aparecerá complementada si tiene el valor 1 y sin complementar si tiene el valor 0.

En el caso del ejemplo anterior, en la figura 4.9 mostramos la implementación en producto de sumas:

$F = \sum m(1,5,7,10,11,14,15) + \phi\,(3, 6, 12) = \prod M(0,2,4,8,9,13)\cdot\phi\,(3, 6, 12)$



<u>*F = (A+D)(A'+C)*</u>

Figura 4.9.- Ejemplo de diseño de una función incompleta en producto de sumas.

3. Método de McCluskey

Como ya hemos apuntado, cuando estamos considerando una función multisalida, la posibilidad de compartir puertas para varias salidas puede conducirnos a una implementación más óptima. Luego deberemos encontrar un método que le dé un mayor peso a los términos comunes a la hora de su elección para desarrollar la fórmula lógica.

El método del mapa no es apropiado para la implementación de funciones multisalida al no considerar la posibilidad de términos comunes en las diferentes salidas de la función. Esta consideración sí se lleva a cabo en el método de McCluskey. Por lo tanto, al considerar funciones multisalida, la definición de implicante cambia de la siguiente forma.

> *Los implicantes de una función multisalida serán aquellos implicantes de las salidas por separado como de todas las posibles combinaciones (productos en el caso de suma de productos o sumas en el caso de producto de sumas) de las diferentes salidas.*

Si tenemos una función con tres salidas, F, G y H, y queremos hallar su suma de productos, los implicantes de la función serán aquellos correspondientes a F·G·H, F·G, F·H, G·H, F, G y H, de tal forma que siguiendo este orden no se añadan los implicantes repetidos. De igual forma, si la implementación fuese en producto de sumas, los implicantes de la función serían los correspondientes a F+G+H, F+G, F+H, G+H, F, G, H.

A partir de ahora, y salvo que se diga lo contrario, supondremos que la fórmula que deseamos obtener deberá estar en suma de productos. En el caso de producto de sumas, el desarrollo es paralelo considerando el principio de dualidad.

Una vez que tengamos todos los implicantes, creamos la denominada tabla de McCluskey. En dicha tabla, cada fila corresponde a un implicante y cada columna corresponde a un 1 de cada función. Dicha tabla se divide por salida (según las columnas) e implicantes de cada salida (según las filas). Se marcan los 1's obtenidos por cada implicante en cada salida. Un ejemplo de dicha tabla se muestra en la figura 4.10. En ella podemos ver implicantes de una sola salida (el implicante 1 y 2 para F1 y el n para F2) e implicantes para dos salidas (el implicante 3).

	F1					F2					
	mi	mj	mk	. . .	mn	ma	mb	mi	mn	. . .	ml
Implicante 1	X							X			
Implicante 2		X	X								
Implicante 3		X			X				X		
⋮											
Implicante n						X	X				X

Figura 4.10.- Ejemplo de una tabla de McCluskey.

La reducción de esta tabla nos llevará al número mínimo de implicantes que serán necesarios para cubrir todos los mintérminos de las salidas, y por tanto a la fórmula mínima. La minimización se reduce a la aplicación sucesiva de los criterios de esencialidad y de dominancia, y eventualmente de equivalencia, hasta que estén cubiertos todos los 1’s de todas las salidas.

El criterio de esencialidad nos indica cuáles de los implicantes son esenciales, y por lo tanto, deben aparecer en la fórmula mínima.

*Un **fila esencial** es aquella cuyo implicante es esencial para alguna de las salidas.*

Este implicante debe aparecer en la fórmula de la salida correspondiente. Una vez cogido dicho implicante, se tacha la fila donde estaba y todas las columnas marcadas en dicha fila ya que dichos 1's han sido obtenidos. Este proceso se repite para todas las filas esenciales. En la tabla, la esencialidad se observa cuando existe alguna columna (algún 1) que únicamente tiene una cruz (es cubierto por un solo implicante); luego el implicante de dicha cruz es esencial.

Una vez que ya no existan más filas esenciales, pasamos a aplicar los criterios de dominancia. Este criterio nos indica los implicantes que son redundantes, y por lo tanto, no deberán aparecer en la fórmula mínima.

Una fila o implicante A se dice que domina a otro B, representándose A ⊃ B, si todas las marcas de B están contenidas en A, que a su vez tiene más.

En este caso se puede eliminar el implicante B tachando su fila.

Se dice que A y B son equivalentes si todas las marcas del implicante A están en el implicante B que no tiene ninguna más.

En este caso se puede eliminar cualquiera tachando la fila correspondiente, no obstante por criterios de minimalidad se elimina aquel implicante que tiene mayor coste (es decir, el que tiene más variables de entrada).

Supongamos que deseamos diseñar un circuito que tenga dos salidas definidas por f1= $\sum m(1,3,7)$ y f2 = $\sum m(2,6,7)$. Los implicantes de esta función se muestran en la figura 4.11.

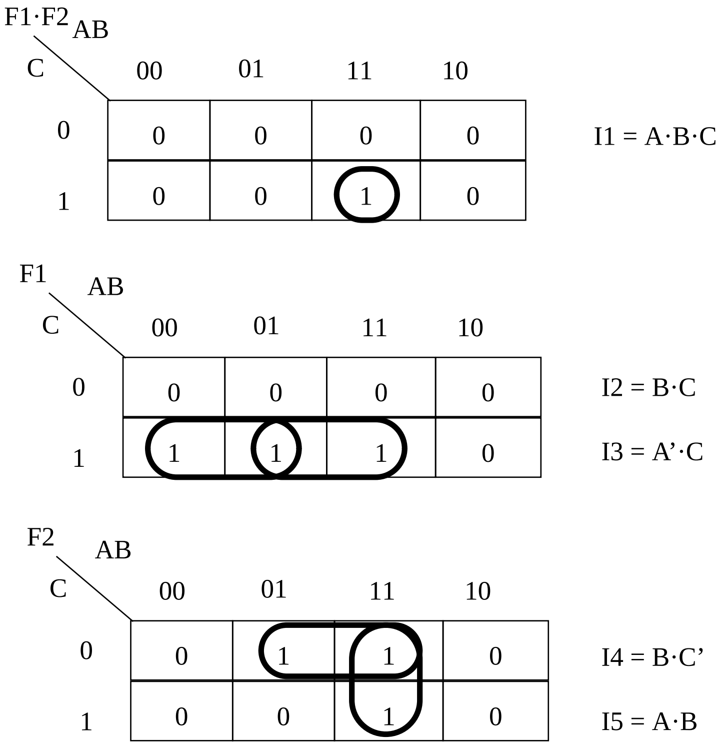


Figura 4.11.- Implicantes de una función multisalida.

En la figura 4.12 se muestra el proceso completo de minimización. En la figura 4.12 (a) mostramos la tabla de McCluskey original, en la que se pueden ver los implicantes comunes a las dos salidas (I1), y los de cada una de ellas (I2 e I3 para F1, I4 e I5 para F2). Así mismo, en las columnas se encuentran los mintérminos de cada salida por separado (como el mintérmino 7 es común a ambas, se coloca dos veces, una en cada salida).

En la figura 4.12 (b), se ha aplicado el criterio de esencialidad. Se comprueba que existen dos implicantes esenciales; I3 para F1 debido al mintérmino 1, e I4 para F2 debido al mintérmino 2. Luego ambos implicantes aparecerán en las fórmulas correspondientes. Al aparecer ya en las fórmulas, los mintérminos cubiertos por dichos implicantes y para esas salidas no son necesarios que se cubran de nuevo. Luego, son eliminadas dichas filas (o la parte correspondiente a la/s salida/s para los que son esenciales) y las columnas cubiertas por dichos implicantes. Por lo tanto, obtenemos una nueva tabla en la que aparecerán los implicantes I1, I2 e I5, y las columnas 7 de ambas salidas.

Seguidamente se aplica los criterios de dominancia (figura 4.12c). Dichos criterios hay que aplicarlos sobre la fila completa, y no sobre cada salida por separado. De esta forma garantizamos que se premia a los implicantes comunes a más de una salida, ya que tendrán más cruces. En este caso, podemos observar que el implicante I1 domina a los otros dos. Por lo tanto, estos últimos serán eliminados.

Después de eliminar los implicantes dominados, volvemos a aplicar los criterios de esencialidad sobre la nueva tabla (figura 4.12d). En este caso observamos que el implicante I1 es

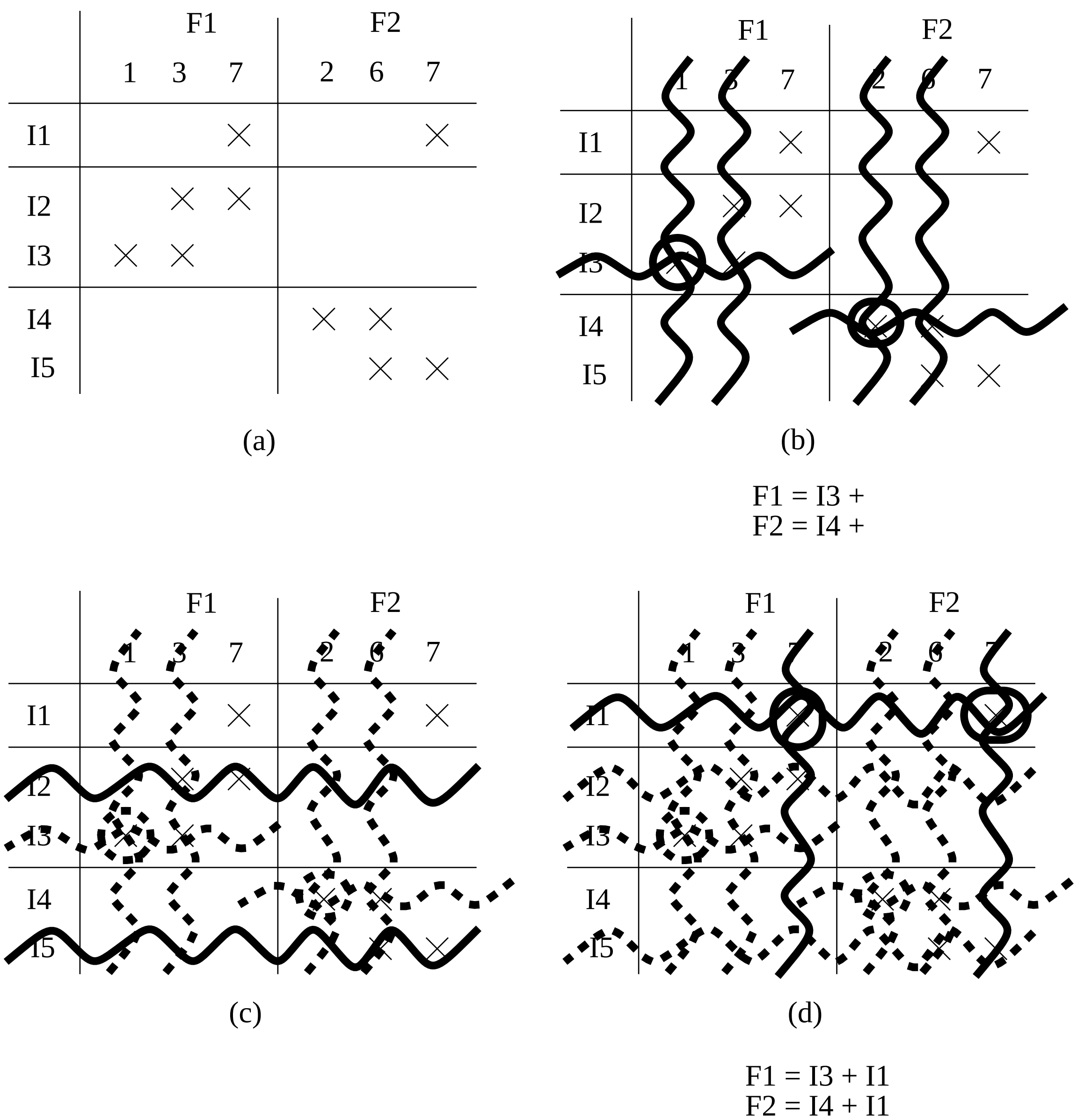


Figura 4.12.- Desarrollo de la minimización de la tabla de McCluskey. (a) Tabla original; (b) después del cirterio de esencialidad; (c) después del criterio de dominancia; y (d) después de la segunda aplicación del criterio de esencialidad.

esencial para ambas salidas, luego deberá aparecer en ambas fórmulas. Una vez hecho esto, se elimina la fila de los implicantes esenciales (o las porciones de ellas), así como las columnas donde tengan cruces.

Este procedimiento finalizará cuando no exista ninguna columna en la tabla. Con ello garantizamos que todos los mintérminos de todas las funciones estarán cubiertos.

Se puede dar el caso de que una vez que hayamos utilizado los criterios de implicantes esenciales y dominantes, no se puedan seguir aplicando ya que no existe ningún implicante dominante sobre otro. Entonces llegamos a lo que se conoce como **tabla de implicantes cíclica**. A partir de este momento no existe ningún método que nos dé una solución mínima. Lo único que se puede hacer es elegir una implicante de forma arbitraria como esencial y seguir con el procedimiento anterior. Esto se repite para todos los implicantes que queden y la fórmula mínima se escoge como la de menor coste de todas las obtenidas.

En el caso de que tengamos funciones incompletas, las inespecificaciones solamente se utilizarían a la hora de obtener las implicantes primas, de tal forma que las implicantes tengan el menor coste posible. Una vez obtenidas dichas implicantes, solamente se trata la función completa asociada.

Al igual que en el método del mapa, si en vez de querer obtener una fórmula como suma de productos, queremos obtenerla como producto de sumas se opera de igual forma pero obteniendo los implicantes a partir de los ceros y en lugar del producto de las funciones se coge la suma, como podemos ver en la figura 4.13

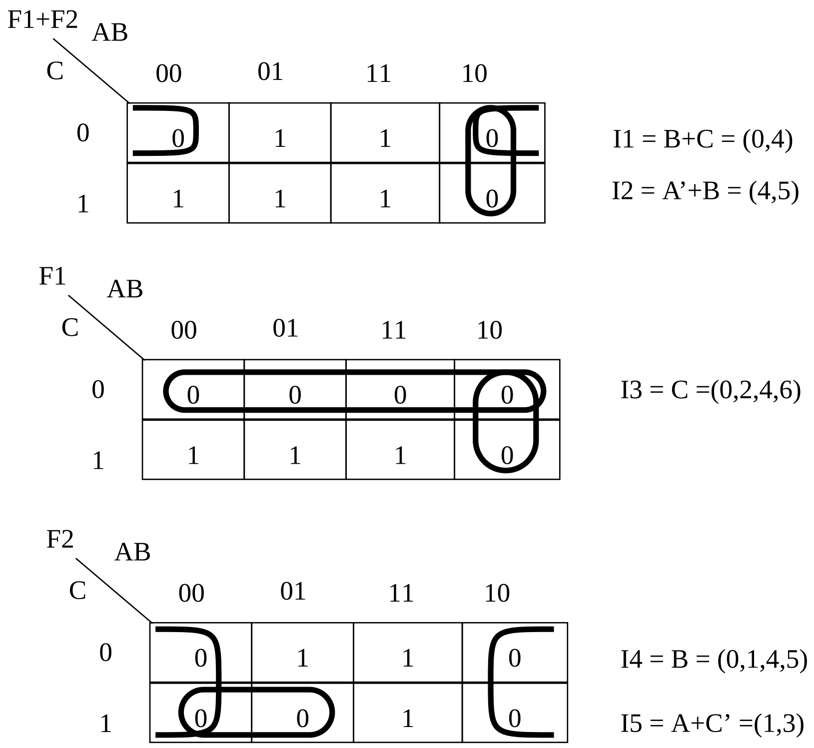


Figura 4.13.- Implicantes de una función multisalida para su implementación como producto de sumas.

El proceso de minimización es mostrado en la figura 4.14. Empezamos con los criterios de esencialidad, observando que los implicantes I3 e I2 son esenciales para F1, y el implicante I5 es esencial para F2. En este caso, el implicante esencial I2 también es implicante para F2 (aunque no esencial para dicha salida). Por lo tanto, se ha cogido para la salida F1, pero no se ha eliminado de la tabla (solamente se ha eliminado la porción de F1); no obstante se ha etiquetado dicho implicante para recordarnos que su nueva elección para otra salida no conlleva coste alguno (si fuese necesario acudir a este parámetro para arbitrar un posible criterio de equivalencia) .

El resto del proceso de minimización es equivalente al visto anteriormente. Se aplican alternativamente los criterios de esencialidad y de dominancia hasta acabar con todas las columnas o llegar a una tabla cíclica.

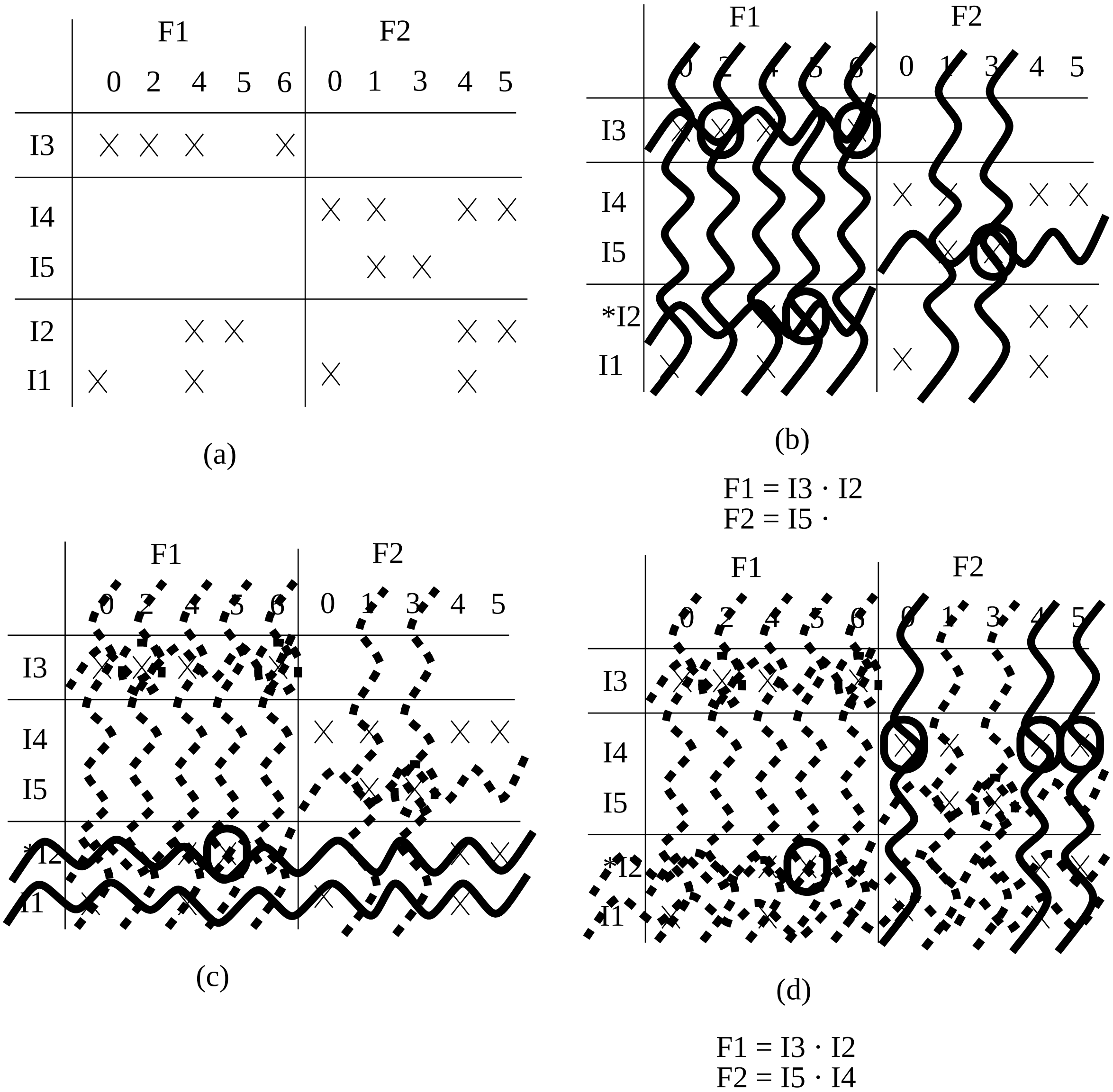


Figura 4.14.- Desarrollo de la minimización de la tabla de McCluskey. (a) Tabla original; (b) después del cirterio de esencialidad; (c) después del criterio de dominancia; y (d) después de la segunda aplicación del criterio de esencialidad.

4. Diseño libre de azares.

Los métodos que hemos visto hasta ahora nos proporcionan una implementación mínima sin tener en cuenta el comportamiento transitorio, es decir, sólo se centra en el comportamiento estático. Para poder obtener un diseño libre de azares, solamente tenemos que hacer algunas modificaciones sobre los métodos anteriores.

Para ello haremos uso de la premisa de que un par de 1’s (0’s) adyacentes no englobados por el mismo implicante, constituirá un riesgo de azar. Luego para evitar todos los riesgos de azares, debemos garantizar que no existan ninguna pareja de 1’s (0’s) adyacentes con la propiedad anterior. Con esta condición conseguimos eliminar todos los azares estáticos, y por lo tanto todos los azares dinámicos (ya que están basados en ellos).

Obviamente, al añadir una nueva funcionalidad al diseño, la no existencia de azares, los diseños obtenidos tendrán un coste mayor o igual que no fuesen considerada esta característica.

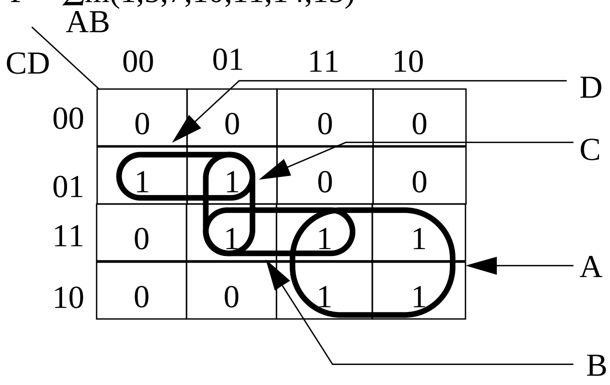
4.1. Método del mapa.

En este caso, tenemos que garantizar que no quede ninguna pareja de 1's adyacentes sin que estén cubiertos por algún implicante común. Por lo tanto, en la fórmula lógica deberán aparecer los implicantes esenciales y no esenciales necesarios para garantizar que no exista ninguna pareja de 1’s (0’s) adyacentes que no esté cubierta por un mismo implicante.

Para añadir esta nueva peculiaridad más a la fórmula de conmutación, servirá toda la formulación desarrollada en el apartado anterior, simplemente sustituyendo los 1’s (0’s) por las parejas de 1’s (0’s) adayacentes.

En el ejemplo correspondiente, la implementación libre de azares se muestra en la figura 4.15. En este caso todos los implicantes son esenciales, por lo que deberán aparecer en la fórmula lógica.

$F = \sum m(1,5,7,10,11,14,15)$



$F = A + B + C + D = a \cdot c + b \cdot c \cdot d + a' \cdot b \cdot d + a' \cdot c' \cdot d$

Figura 4.15.- Implementación libre de azares utilizando el método del mapa.

4.2. Método de McCluskey

Al igual que sucedía con la formulación basada en el mapa de Karnaugh, la formulación desarrollada en el método de McCluskey es la misma sustituyendo los 1’s (0’s) por las parejas de 1’s (0’s) adyacentes. En este caso, solamente se vería alterada la creación de la tabla de McCluskey en el sentido en que donde antes se colocaban los 1’s (0’s) de cada salida, es decir, en las columnas, ahora se colocan las parejas de 1’s adyacentes o los 1’s que no tengan ninguno adyacente. Seguidamente el implicante deberá cubrir a toda la pareja de cada columna.

En la figura 4.16 mostramos la tabla de McCluskey libre de azares del ejemplo anterior. En este caso el implicante I1 no cubre ninguna columna, ya que dicho implicante sólo cubre el mintérmino 7, pero no la pareja 3-7 ni la 6-7.

	F1		F2	
	1-3	3-7	2-6	6-7
I1				
I2		×		
I3	×			
I4			×	
I5				×

Figura 4.16.- Tabla de McCluskey libre de azares.

El proceso de minimización será idéntico al mostrado en el apartado anterior. Es decir, la aplicación alternativa de los criterios de esencialidad y de dominancia. En este caso, los implicantes I2 e I3, e I4 e I5 son esenciales para las salidas F1 y F2 respectivamente. Por lo tanto, la implementación mínima libre de azares quedaría de la siguiente forma:

$$f1 = I2 + I3 = a' \cdot c + b \cdot c$$

$$f2 = I4 + I5 = b \cdot c' + a \cdot b$$

5. Implementación del circuito digital

Hasta ahora, hemos desarrollado los métodos para obtener una fórmula de conmutación mínima que realiza una determinada función de conmutación. El siguiente paso, y último, consiste en pasar de la fórmula al circuito digital.

Este paso es simple y directo. Se procede de tal forma que se sustituye el operador que aparece en la fórmula por su puerta equivalente. Luego el operador producto es sustituido por una puerta AND, mientras que el operador suma es sustituido por una puerta OR. De igual forma, cuando aparezca, también se sustituirá el operador negación por un inversor. Así, para el caso de la función f = a·c + a'·d, la implementación sería la mostrada en la figura 4.17:

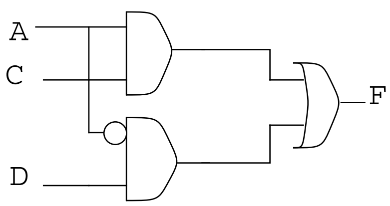


Figura 4.17.- Implementación en suma de productos de la función f = a·c + a'·d

6. Conversión entre implementaciones de dos niveles.

Las implementaciones de dos niveles son, como ya hemos dicho, suma de productos o producto de sumas. La implementación para suma de productos (producto de sumas) se suelen hacer con un primer nivel de puertas AND (OR) seguidas de una sola puerta OR (AND) con tantas entradas como puertas AND (OR) existan.

Para convertir una implementación de suma de productos en producto de suma se procede a la doble inversión, como ya se ha comentado. No obstante, la implementación obtenida no tiene porqué ser mínima, sino que puede ser necesario una minimización utilizando los teoremas del álgebra de Boole. Veamos el ejemplo de la fórmula:

$$F = a'\cdot b\cdot c + a\cdot b' + a\cdot b\cdot c'$$

$$F' = (a+b'+c')\cdot(a'+b)\cdot(a'+b'+c) = (a+b'+c')\cdot(a'+b\cdot c) = a\cdot b\cdot c + a'\cdot b' + a'\cdot c'$$

$$F = (F')' = (a'+b'+c')\cdot(a+b)\cdot(a+c)$$

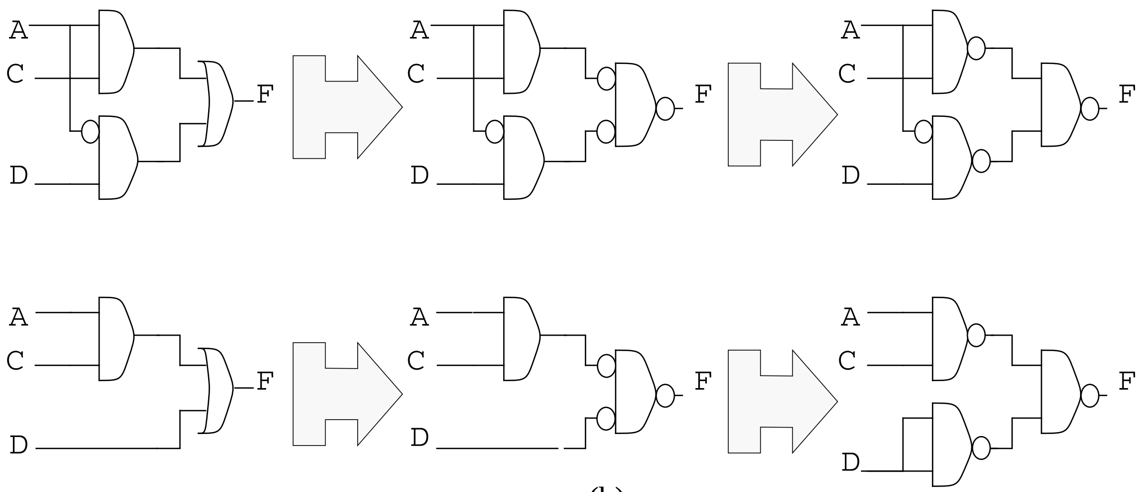
Es decir, negamos la fórmula que deseamos convertir dejándola de la misma forma (suma de producto o producto de sumas) que la fórmula original. Una vez realizada la primera negación, se vuelve a negar, obteniendo directamente la expresión deseada.

Hasta ahora sólo hemos contado con el conjunto completo de las puertas AND, OR e inversores. No obstante, existen más conjuntos completos como el de las puertas NAND y el de las puertas NOR. Por lo tanto, a continuación nos planteamos la conversión de suma de productos y producto de sumas a implementaciones NAND-NAND y NOR-NOR.

En el caso de suma de productos, vemos que si cambiamos la puerta OR por su implementación con puertas NAND es decir, (a' and b')', obtenemos la implementación mostrada en la figura 4.18. De donde observamos que la conversión de suma de productos a NAND-NAND es inmediata y sin necesidad de realizar casi ningún cambio. El único caso en que sería necesario algún cambio se da cuando existe alguna entrada que va directamente a una entrada de la puerta OR del segundo nivel. En este caso, como podemos observar en la figura 4.18b es necesario la adición de un inversor (o de algún elemento con su funcionalidad).

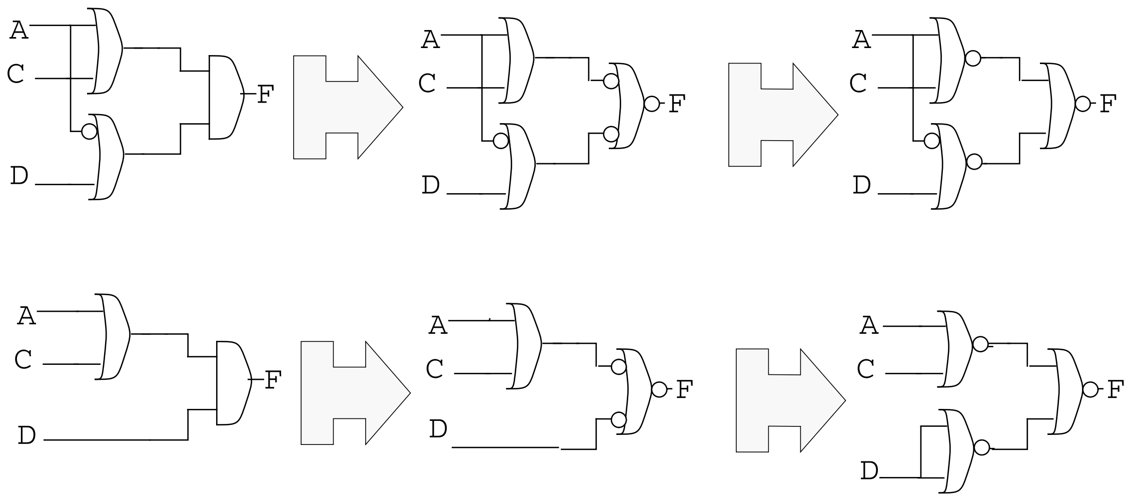
En el caso del producto de suma, se puede ver de igual forma que su conversión a NOR-NOR es igualmente inmediata, como sucede con el caso anterior. Un ejemplo similar al anterior se muestra en la figura 4.19.

Por último para pasar de una implementación NAND-NAND a una NOR-NOR (y viceversa), solamente se necesita pasar de la implementación de suma de productos a producto de sumas (y viceversa).



(b)

Figura 4.18.- Conversión de la implementación AND-OR a la implementación NAND-NAND.



(b)

Figura 4.19.- Conversión de la implementación OR-AND a la implementación NOR-NOR.