



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Generación de funciones lógicas mediante multiplexores

Apellidos, nombre	Martí Campoy, Antonio (amarti@disca.upv.es)
Departamento	Informática de Sistemas y Computadores
Centro	Universidad Politécnica de Valencia



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



1 Resumen de las ideas clave

En este artículo se va a presentar la utilización de multiplexores para la generación de funciones lógicas. Son muchas las formas de diseñar una función lógica, y una de las más sencillas es la utilización del bloque combinacional conocido como multiplexor. Para poder adquirir los conocimientos y habilidades presentadas en este artículo debes contar con unos conocimientos previos, listados en la tabla 1. Pero tranquilo, durante el texto se incluirán breves descripciones de estas ideas previas.

Tabla 1 Conocimientos previos

Conocimientos previos
1. Qué es una función lógica y su aridad
2. Tipos y tablas de verdad de puertas lógicas
3. Formas de representar una función lógica: tabla de verdad, formas canónicas y expresiones algebraicas
4. Funcionamiento de un multiplexor

2 Objetivos

Una vez hayas leído este artículo docente y reproducido los ejemplos presentados, deberás ser capaz de **implementar** una función lógica mediante el uso de multiplexores.

Además, la implementación de la función lógica podrá tomar como punto de partida diferentes representaciones de la misma, como la tabla de verdad o una forma canónica, por lo que serás capaz de **traducir** desde una representación a otra.

Por último, y atendiendo a criterios de simplificación de circuitos, serás capaz de **simplificar** el diseño reduciendo el tamaño del multiplexor.

3 Introducción

En primer lugar, una breve descripción de los conceptos previos más importantes para poder alcanzar los objetivos propuestos en este artículo. Estas descripciones pueden ampliarse consultando la bibliografía propuesta al final del documento.

- Función lógica: expresión formal del comportamiento de un circuito digital. La aridad de una función lógica es el número de variables de entrada.
- Tabla de verdad: representación única en forma de tabla de una función lógica.



- Formas canónicas: representación única como suma de productos o como producto de sumas de una función lógica.
- Expresión algebraica: combinación de variables y operadores lógicos para expresar una función lógica.
- Puerta lógica: circuito digital que implementa una función lógica básica.
- Circuito o función combinacional: circuito en el que las salidas en un instante de tiempo dependen exclusivamente de las entradas en ese mismo instante de tiempo.
- Multiplexor: circuito combinacional, con m entradas de selección, $n=2^m$ entradas de datos, y una única salida.
- La función realizada por el multiplexor corresponde con la de un selector o conmutador, conectando la salida con una de las entradas.
- En concreto, el valor de la salida toma el valor presente en la entrada i -ésima cuando en las entradas de selección se asigna el valor i .
- La Figura 1 muestra el símbolo lógico para un multiplexor de m entradas de selección y $n=2^m$ entradas de datos.

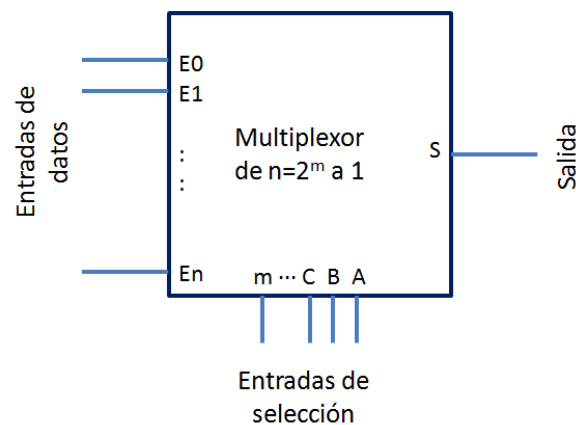


Figura 1 Símbolo lógico de un multiplexor de n entradas de datos

Para ver si el funcionamiento del multiplexor está claro, vamos a hacernos un par de preguntas. Tenemos un multiplexor de 2 entradas de selección, llamadas B y A, siendo A la de menor peso. Las entradas de datos, cuatro, se llaman E0, E1, E2 y E3, siendo E0 la de menor peso. La Tabla 2 muestra en cada fila una combinación de valores para las entradas. ¿Cuál es el valor de la salida para cada una de las 6 combinaciones? Por favor, piensa las respuestas antes de ver las soluciones!

Tabla 2 Ejercicios de multiplexores

Combinación	Entradas de selección		Entradas de datos				Salida
	B	A	E3	E2	E1	E0	
0	0	1	1	1	0	1	¿?
1	0	1	1	0	1	0	¿?
2	1	1	1	0	0	1	¿?
3	1	1	0	1	1	1	¿?
4	1	0	1	0	1	1	¿?
5	1	0	0	1	0	0	¿?



4 Generación de funciones

La generación de funciones utilizando multiplexores es muy sencilla. En primer lugar veremos como generar una función utilizando un multiplexor con el mismo número de entradas de selección que variables de entrada (aridad) tiene la función. En segundo lugar, veremos una optimización que permite utilizar un multiplexor con una entrada menos que variables de entrada tiene la función.

4.1 Multiplexores con la misma aridad que la función

La forma más sencilla de implementar una función con un multiplexor es utilizar uno que tenga el mismo número de entradas de selección que variables de entrada tenga la función. Así, para una función con tres variables de entrada (aridad 3) utilizaríamos un multiplexor con tres entradas de selección, es decir, un multiplexor de 8 a 1.

De este modo, en cada una de las entradas de datos colocaremos el valor correspondiente de la salida de la función. Finalmente, conectaremos las variables de entrada de la función a las entradas de selección del multiplexor.

Pero lo veremos mejor con un ejemplo. La Tabla 3 muestra la tabla de verdad de la función F, de aridad 3. La Ecuación 1 muestra las formas canónicas conjuntiva y disyuntiva para la misma función F.

Tabla 3 Tabla de verdad de la función F

Valoración	C B A	F
0	0 0 0	0
1	0 0 1	1
2	0 1 0	0
3	0 1 1	0
4	1 0 0	1
5	1 0 1	1
6	1 1 0	1
7	1 1 1	0

Ecuación 1 Formas Canónicas Conjuntiva y Disyuntiva para la función F

$$F = \prod_{C,B,A} (0,2,3,7)$$
$$F = \sum_{C,B,A} (1,4,5,6)$$

Podemos ver que cuando las entradas de la función valen 1, 4, 5 o 6 la salida de la función es 1. Por tanto, en las entradas del multiplexor 1, 4, 5, y 6 pondremos un uno. De este modo, cuando seleccionemos una de estas entradas, tendremos en la salida un 1.

Igualmente, podemos ver que cuando las entradas de la función valen 0, 2, 3, o 7 la salida de la función es 0. Por tanto, en las entradas del multiplexor 0, 2, 3 y 7 pondremos un cero. De este modo, cuando seleccionemos una de estas entradas, tendremos en la salida un 0.

La Figura 2 muestra la implementación de la función F usando un multiplexor de 8 a 1. Es muy importante que te fijes en el orden en que se conectan las variables a las entradas de selección.

Algunas funciones tienen entradas indiferentes, es decir, que la salida no está definida para algunas entradas. Estas entradas se identifican en la tabla de verdad con el símbolo "X", y en las formas canónicas como un grupo de minitérminos o maxitérminos diferenciados. La Tabla 4 muestra la tabla de verdad para la función



H, con entradas indiferentes. La Ecuación 2 muestra las formas canónicas conjuntiva y disyuntiva para la misma función H.

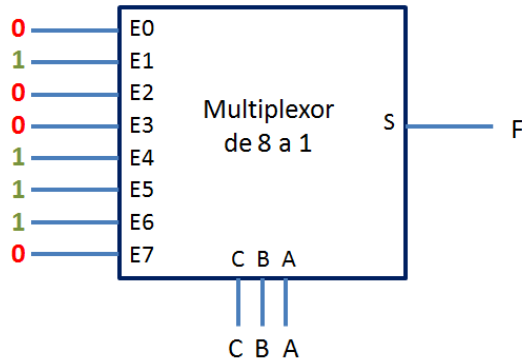


Figura 2 Implementación de la función F usando un multiplexor

Tabla 4 Tabla de verdad de la función H

Valoración	C B A	H
0	0 0 0	1
1	0 0 1	X
2	0 1 0	1
3	0 1 1	X
4	1 0 0	X
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0

Ecuación 2 Formas Canónicas Conjuntiva y Disyuntiva para la función H

$$H = \prod_{C,B,A} (5,7) \cdot \prod_{\emptyset} (1,3,4)$$

$$H = \sum_{C,B,A} (0,2,6) + \sum_{\emptyset} (1,3,4)$$

Cuando la función incluye entradas indiferentes, es obligatorio identificarlas correctamente en la tabla de verdad y en las formas canónicas. Pero a la hora de implementar la función con un circuito, el símbolo "X" no puede utilizarse, por lo que hay que asignarle un valor de 0 o de 1. Y precisamente, como son entradas *indiferentes*, es indiferente asignar valor 0 a todas las X, o valor 1 a todas las X, o valor 0 y 1 de forma aleatoria. Lo que no debe hacerse nunca es dejar una entrada al aire, o utilizar valores distintos de 0 y 1. La Figura 3 muestra una de las posibles implementaciones de la función H usando un multiplexor de 8 a 1.

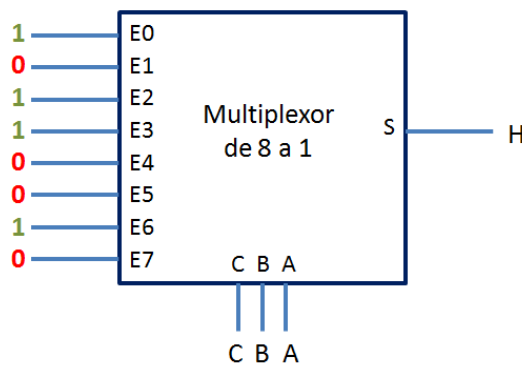


Figura 3 Implementación de la función H con un multiplexor de 8 a 1



La figura Figura 4 muestra los pasos a seguir para implementar una función con un multiplexor, partiendo de la tabla de verdad o de las formas canónicas.

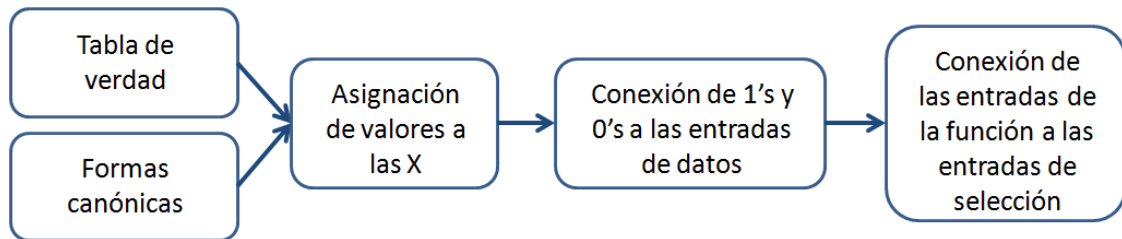


Figura 4 Pasos para la implementación de una función usando un multiplexor

4.2 Multiplexores de menor aridad que la función

En este apartado veremos como implementar una función de aridad n utilizando un multiplexor con $n-1$ entradas de selección.

Al tener una entrada de selección menos, el multiplexor dispone de, exactamente, la mitad de entradas de datos que valoraciones tiene la función. En principio solo podríamos implementar media función, lo que no sería de gran utilidad. La opción de utilizar dos multiplexores tampoco sería muy útil.

¿Cómo solucionar este problema? Fácil. Expresar la salida de la función lógica como función de la variable de entrada de menor peso.

Vuelve a leer el párrafo anterior. Estoy de acuerdo contigo. No queda demasiado claro, así que vamos a verlo con una función muy sencilla, de aridad dos. La Tabla 5 muestra la tabla de verdad de la función E . Sus variables de entrada son B y A , siendo A la de menor peso.

Tabla 5 Tabla de verdad de la función E

Valoración	B A	E
0	0 0	1
1	0 1	1
2	1 0	0
3	1 1	1

Si miramos con atención la tabla de verdad veremos que cuando la entrada B vale 0, la salida vale 1 independientemente del valor de la entrada A . Y cuando la entrada B vale 1, la salida de la función toma el valor de la entrada A . De este modo podemos mover la entrada A a la columna de la salida, y obtener una tabla reducida, que se muestra en la Tabla 6. Fíjate que ahora la función se expresa como $E(A)$, es decir, se indica que depende de la entrada A .

Tabla 6 Tabla de verdad reducida de la función $E(A)$

Valoración	B	$E(A)$
0	0	1
1	1	A



Una vez obtenida la tabla reducida, ya podemos utilizar un multiplexor con una entrada de selección menos. En este caso la aridad de la función es 2, por lo que utilizaremos un multiplexor de una entrada de selección, o lo que es lo mismo, un multiplexor de 2 a 1. La Figura 5 muestra la implementación de la función E usando un multiplexor con una única entrada de selección. Es importante que te fijes que a la entrada de selección se ha conectado la entrada de mayor peso de la función E, y no la de menor peso.

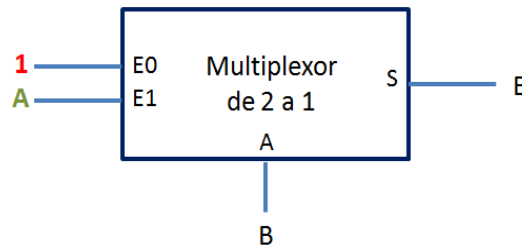


Figura 5 Implementación de la función E con un multiplexor de una entrada de selección

Si la función cuenta con entradas indiferentes, estas pueden utilizarse para simplificar la implementación. La Tabla 7 muestra la tabla de verdad de la función F, de aridad 3, y la tabla reducida para implementarla utilizando un multiplexor de dos entradas de selección. Fíjate en la agrupación de la salida de la función para las cuatro combinaciones de las variables C y B.

Tabla 7 Tabla de verdad y tabla reducida de la función F

Valoración	C B A	F
0	000	1
1	001	X
2	010	0
3	011	X
4	100	X
5	101	0
6	110	1
7	111	0

C B	F(A)
00	1, \bar{A}
01	0, A
10	0, \bar{A}
11	\bar{A}

Es importante que te fijes que al ser una función con entradas indiferentes, existen varias posibilidades de reducir la función. En la tabla reducida deben aparecer todas las opciones, ya que de lo contrario estamos ocultando información sobre la función lógica. Sin embargo, a la hora de implementar el circuito utilizando un multiplexor, es necesario decidir cual de las posibles entradas escogeremos. Esta decisión se realizará en base a criterios de optimización y simplificación de circuitos. Por ejemplo, si podemos escoger entre poner un 1 o la entrada negada, elegiremos el valor fijo 1, ya que nos ahorramos la puerta NOT necesaria para negar la entrada.

La Figura 6 muestra una posible implementación de la función F. Es importante que te fijes en como se conectan las entradas de la función a las entradas de selección del multiplexor.

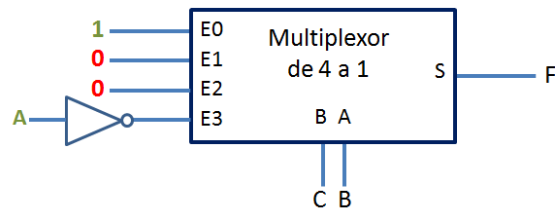


Figura 6 Implementación de la función F utilizando un multiplexor con una entrada de selección menos

La Figura 7 muestra los pasos para implementar una función utilizando un multiplexor con una entrada de selección menos que la aridad de la función.

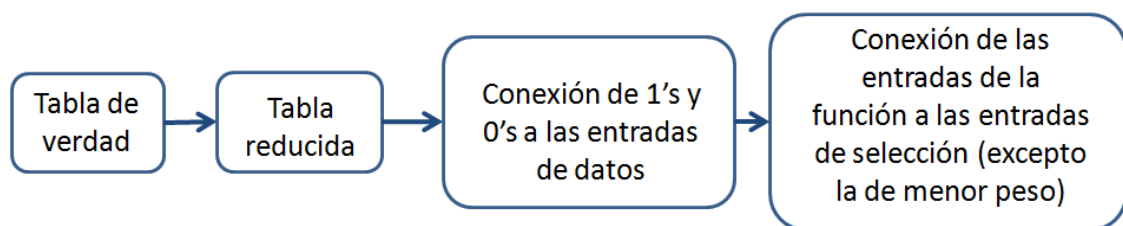


Figura 7 Pasos para la utilización de un multiplexor con una entrada de selección menos

4.3 Resumen

La implementación de una función lógica mediante el uso de multiplexores puede realizarse de forma directa a partir de la tabla de verdad, o de las formas canónicas, utilizando un multiplexor con el mismo número de entradas de selección que variables de entrada tenga la función. Es decir, coincidiendo la aridad de la función y del multiplexor.

También es posible implementar una función lógica de n variables de entrada utilizando un multiplexor con $n-1$ entradas de selección. En este caso, es necesario partir de la tabla de verdad y crear una tabla reducida, donde se exprese la salida del circuito como función de la entrada de menor peso.

En este último caso, la existencia de entradas indiferentes, conocidas como X s, permite realizar simplificaciones y optimizaciones.

4.4 Ejercicios

A continuación unos sencillos ejercicios. Para la función lógica J expresada por su forma canónica disyuntiva, responde a las siguientes preguntas:

$$J = \sum_{Z,Y,X,W} (0,1,3,7,8,13) + \sum_{\emptyset} (2,4,5,9,12,15)$$

- Si utilizamos un multiplexor de 4 entradas de selección, ¿Cuáles serán los valores de las entradas? (Para las entradas indiferentes, decidimos tomar valor 0)
- Si utilizamos un multiplexor de 4 entradas de selección, llamadas D , C , B y A , siendo A la de menor peso, ¿Cuál debe ser la conexión de las entradas de la función a las entradas de selección del multiplexor?



- c. Si utilizamos un multiplexor de 3 entradas de selección, ¿Cuáles serán los valores de las entradas? Indica, para cada entrada, todos los posibles valores, tal como se mostraría en la tabla reducida
- d. Si utilizamos un multiplexor de 3 entradas de selección, llamadas C, B y A, siendo A la de menor peso, ¿Cuál debe ser la conexión de las entradas de la función a las entradas de selección del multiplexor?
- e. Intenta responder las preguntas antes de ver las soluciones, por favorⁱⁱ.

5 Conclusiones

En este artículo has podido conocer una forma rápida y sencilla de implementar una función lógica mediante el uso de multiplexores. Para una función de aridad n , es posible utilizar un multiplexor de n entradas de selección, o un multiplexor de $n-1$ entradas de selección. En el primer caso, los valores que deben conectarse a las entradas de datos son directamente los mostrados en la tabla de verdad o en las formas canónicas. En el segundo caso, es necesario partir de la tabla de verdad y realizar agrupaciones, para expresar la salida en función de la variable de entrada de menor peso.

6 Bibliografía

6.1 Libros:

[1] [John F. Wakerly](#) "Digital design : principles and practices", Upper Saddle River : Pearson Prentice Hall. 2006

[2] Antonio Lloris Ruiz; Alberto Prieto Espinosa; Luis Parrilla Roure "Sistemas digitales", Aravaca, Madrid : McGraw-Hill/Interamericana de España. 2003

6.2 Referencias de fuentes electrónicas:

[3] [Marfí Campoy, Antonio](#) "Circuitos combinacionales: multiplexores" Universitat Politècnica de València. Escola Tècnica Superior d'Enginyeria Informàtica. 2011. <http://politube.upv.es/play.php?vid=46987>

ⁱ Las salidas, para las combinaciones 0 a 5 son: 0, 1, 1, 0, 0, 1.

ⁱⁱ Respuestas a los ejercicios: a) Los valores de las entradas de E0 a E15 son: 1,1,0,0,0,0,1,1,1,0,0,0,1,0,1; b) La conexión de las entradas de selección es: $W \rightarrow A$, $X \rightarrow B$, $Y \rightarrow C$, $Z \rightarrow D$; c) Los posibles valores de las entradas de E0 a E7 son: E0 = 1; E1 = (1, W); E2 = (0,1,W, Not(W)); E3 = W; E4 = (1, Not(W)); E5 = 0; E6 = (1,W); E7 = (0,W); d) La conexión de las entradas de selección es: $X \rightarrow A$, $Y \rightarrow B$, $Z \rightarrow C$.