

Preguntas Más Frecuentes Tema 3

Contenido

- P.3.1: ¿Los terminales de los circuitos están definidos claramente como entradas y salidas o se pueden usar de forma indiferente?..... 2
- P.3.2: ¿Las entradas de las señales de control y de inhibición/facilitación se pueden usar como entradas de datos? 2
- P.3.3: ¿Siempre que se diseñan circuito combinacionales tenemos que minimizar las expresiones resultantes? 3
- P.3.4: ¿Me pueden explicar el análisis realizado en el ejercicio de E.6.3 (pag. 186) del libro de problemas? 3
- P.3.5: En el apartado C) del ejercicio E.6.4 se realiza el diseño de un circuito decodificador de Exceso-3 a Gray. ¿Por qué motivo se calculan de forma independiente las expresiones de los bits de la palabra G (G_3 G_2 G_1 G_0) y cómo se consiguen para su implementación mediante MUX. 4
- P.3.6: ¿Qué postulados y teoremas se usan para obtener la expresión de B en el apartado D) del ejercicio E.6.4 (pag. 193) en el que se realiza el diseño de un circuito decodificador de Decimal a BCD.?..... 5
- P.3.7: ¿No entiendo cómo se obtiene el mapa de Karnaugh de la Fig. 6.4.5 (pag. 190) del apartado C) del ejercicio E.6.4, en el que se realiza el diseño de un circuito decodificador de BCD Exceso a 3 a Gray? 6
- P.3.8: ¿Por qué para obtener la expresión de G_2 en el apartado C (decodificador de Exceso-3 a Gray) del ejercicio E.6.4 no se pueden agrupar los 8 términos mínimos $m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}$ de la parte inferior de la tabla a) de la figura 6.4.6? 6
- P.3.9: ¿Hay que estudiar los Amplificadores (Buffers-Drivers) y los Transmisores-Receptores de bus?.... 7
- P.3.10: ¿Si los Amplificadores (Buffers-Drivers) no son inversores y dejan la señal tal cual, para qué sirven? 7
- P.3.11: ¿Hay distintos tipos de puertas de tres estados o sólo son Amplificadores? 8
- P.3.12: ¿Cómo funcionan los Amplificadores (Buffers-Drivers) y qué diferencia hay entre los distintos tipos? 8
- P.3.13: ¿Qué función realizan los buffers drivers de la figura 6.19? 8
- P.3.14: ¿Los Buffers Drivers que aparecen en la fig. 6.20 de la pag. 332 del texto invierten la señal de entrada?.9
- P.3.15: Hay distintos tipos de Amplificadores o “Buffers Drivers” ¿Cuáles son estos tipos y cómo funcionan?..... 9

P.3.1: ¿Los terminales de los circuitos están definidos claramente como entradas y salidas o se pueden usar de forma indiferente?

R.3.1: Los circuitos poseen dos tipos de terminales claramente diferenciados y dependientes de su propia estructura: las entradas mediante las cuales se introducen las distintas señales ya sean de datos o de control y las salidas en las que se obtienen las respuestas del circuito de acuerdo con la función para la que ha sido diseñado. Por tanto, las entradas siempre actúan como tal y nunca pueden actuar como salidas y a la inversa.

Las entradas son los puntos por donde se estimulan los circuitos, lo cual implica que en las entradas siempre tengamos que conectar generadores de señales variables (en nuestro caso, generadores de trenes de pulsos) o señales de continua (0V, 5V, etc...) o señales procedentes de otros circuitos (las salidas de otros circuitos).

En las salidas se obtienen las respuestas del circuito a dichas señales de entrada que, de nuevo, pueden ser trenes de pulsos o valores de continua.

En los circuitos reales es fundamental no confundir las salidas con las entradas, ya que poner en una salida una tensión o un generador de señal supone romper el circuito correspondiente. Sin embargo, el mismo error en una simulación sólo implica que el circuito no funciona.



P.3.2: ¿Las entradas de las señales de control y de inhibición/facilitación se pueden usar como entradas de datos?

R.3.2: Sí, al fin y al cabo son entradas del circuito mediante las que se introducen tensiones y corrientes. El significado se lo pone el usuario en función del uso que hace de dichas entradas. Los ejemplos más representativos de este cambio de uso se pueden ver en los multiplexos y demultiplexos. En estos circuitos se distingue entre señales de control y datos de entrada cuando se usan en aplicaciones de ruta de datos cuya función es presentar en la salida un determinado dato en función de las señales de control. Sin embargo, cuando se usan para el diseño de funciones lógicas, las entradas de control actúan como entradas de datos. Por ejemplo, en el circuito de la *figura 6.6* del texto las entradas de control *A* y *B* de los MUX se usan como entradas de los datos *U*, *V*, *X* e *Y*.

Lo mismo ocurre en el circuito de la *figura 6.15*. En este circuito se usa el integrado SN74138 que consta de 2 DEMUX de 3 a 8, es decir, tiene tres entradas de datos y 8 salidas, pero además tiene una entrada de control o de facilitación/inhibición, *EN*, que permite que el circuito funcione como tal o se inhiba. Sin embargo, esta entrada de *facilitación/inhibición*, *EN*, no se usa como tal sino que lo hace como el bit más significativo de los datos de entrada que son palabras de 4 bits. En la tabla de verdad de la *figura 6.13*, se observa que el bit *D* toma el valor "0" para los 8 primeros términos mínimos y toma el valor "1" para los 8 últimos, lo que nos lleva a considerar que la señal de facilitación del circuito, *EN*, puede realizar la función de este bit (*D*) de la palabra de entrada y así sólo tenemos que usar 2 DEMUX. Por tanto, en este caso, se ha considerado como un dato ya que, al fin y al cabo, es una entrada y lo importante es su

funcionamiento, no su nombre (ver explicación al inicio del último párrafo de la página 326).

Como comentario general, los circuitos tienen unas entradas y unas salidas que al usarlas podemos darles el significado que nos convenga, pero teniendo siempre en cuenta su funcionamiento, que no lo podemos cambiar ya que depende de la propia estructura del circuito y es el que se especifica en su tabla de verdad.



P.3.3: ¿Siempre que se diseñan circuito combinacionales tenemos que minimizar las expresiones resultantes?

R.3.3: No, depende del tipo de dispositivo que se va a usar para implementar el circuito resultante. Un circuito combinacional se puede implementar con puertas, con multiplexos o con PLDs.

Cuando se implementa un circuito con puertas, se tienen que minimizar las expresiones resultantes para usar el menor número de puertas posibles. Por tanto, hay que intentar que los agrupamientos que se realizan en el diagrama de Karnaugh abarquen el mayor número de elementos posibles con el fin de eliminar el mayor número de variables y, por tanto, usar el menor número de puertas y con el menor número de entradas.

Si se implementa con multiplexos, hay que minimizar por filas o por columnas, dependiendo de las variables que se tomen como señales de control de los multiplexos y cómo se construya el diagrama de Karnaugh, porque el término mínimo que encabeza cada una de las filas o columnas da lugar a las variables de control y el resultado de minimizar los contenidos de las celdas de esa fila o columna son las señales de entrada al Multiplexo.

Por último, si se va a implementar el circuito con PLDs no sólo no tenemos que realizar ninguna minimización, sino que hay que expandir las funciones en el caso de que estén minimizadas, ya que en este tipo de circuito se usan directamente los términos mínimos de las variables de entrada.



P.3.4: ¿Me pueden explicar el análisis realizado en el ejercicio de E.6.3 (pag. 186) del libro de problemas?

R.3.4: Analizar un circuito es obtener las expresiones de sus señales de salida en función de las señales de entrada y de control, para ello debe conocer las funciones que realizan los distintos componentes del circuito, en este caso, multiplexos.

La función del multiplexo es transferir a su/s salida/s las señales de entrada en función de los valores que toman las señales de control.

En este caso tenemos un circuito con dos etapas de multiplexos de 4 a 1. Así, estos multiplexos poseen 4 entradas que se controlan con dos señales de control (u y v para el control de los dos MUX de la etapa de entrada y, x e y para el control del multiplexo de la etapa de salida) y cada uno posee una salida.

La forma de analizar el circuito es ir obteniendo las expresiones de las funciones a la salida de cada elemento del circuito en función de sus entradas llegando, finalmente, a obtener las expresiones de las variables de salida en función de las variables de entrada.

El análisis lo podemos hacer desde las entradas hacia las salidas o a la inversa. En este ejercicio está hecho desde las entradas hacia las salidas.

Veamos la salida del MUX0:

Recordemos que las D_i se corresponde con los términos mínimos de las señales de control. Es decir,

$$D_0 = \bar{u}\bar{v}, \quad D_1 = \bar{u}v, \quad D_2 = u\bar{v}, \quad D_3 = uv$$

Por tanto, la salida de MUX0 es:

$$f_3 = 0D_0 + 0D_1 + zD_2 + \bar{z}D_0 = zu\bar{v} + \bar{z}uv = u(z\bar{v} + \bar{z}v) = u(z \oplus v)$$

Análogamente, la salida de MUX1 es:

$$f' = 1D_0 + 1D_1 + 0D_2 + 0D_3 = \bar{u}\bar{v} + \bar{u}v = \bar{u}(\bar{v} + v) = \bar{u}$$

Por último, realizando la misma operación con el MUX de salida obtenemos:

$$f_4 = 0D_0 + f_3D_1 + 1D_2 + f'D_3 = f_3\bar{x}y + x\bar{y} = f'xy$$

Sustituyendo f_3 y f' por sus expresiones y operando se obtiene el resultado del texto.

$$f_4 = \bar{x}y u(z \oplus v) + x\bar{y} \bar{u}$$



P.3.5: En el apartado C) del ejercicio E.6.4 se realiza el diseño de un circuito decodificador de Exceso-3 a Gray. ¿Por qué motivo se calculan de forma independiente las expresiones de los bits de la palabra G (G_3 G_2 G_1 G_0) y cómo se consiguen para su implementación mediante MUX.

R.3.5: Partimos de las dos últimas columnas de la tabla de verdad de la figura 6.4.1. Se calculan por separado G_3 , G_2 , G_1 y G_0 porque es la forma general de obtener cada una de las variables de salida (en este caso del código Gray) en función de las palabras de entrada (en este caso en Exceso a 3). Es decir, vemos a qué términos mínimos de las variables de entrada corresponden los "1" de las distintas variables de salida que deseamos obtener, en este caso, G_3 , G_2 , G_1 , G_0 . Esta es la forma general de diseño de un circuito combinacional a partir de la tabla de verdad.

Como queremos implementar el decodificador con MUX, para todas las G_i elegimos como variables de control de los MUX las mismas variables del exceso a 3. En este caso x_3 y x_2 . Por tanto, como variables de entrada usamos las variables restantes, x_1 y x_0 .

Si observamos la tabla de verdad, vemos que hay una serie de términos mínimos que no se van a presentar porque no existen en Exceso a 3, por lo que los consideramos como términos indiferente (los nombramos con x porque pueden tomar cualquiera de los dos

valores, 0 ó 1). Esto nos facilita la minimización ya que los podemos usar para hacer mayores agrupamientos y obtener la expresión “más” mínima. Hasta aquí, es la forma normal de resolver cualquier problema de síntesis de circuitos combinacionales.

A partir del punto en el que tenemos las expresiones lógicas de las G_i , o bien podemos implementar el circuito directamente, o bien primero implementamos cada una de las G y después unimos los 4 circuitos correspondientes a cada una de las variables. Hemos hecho esta última forma por razones pedagógicas, ya que si ponemos directamente el circuito final, *Fig. 6.4.9*, es más difícil de entender y de hacer, salvo que tengamos cierta práctica.



P.3.6: ¿Qué postulados y teoremas se usan para obtener la expresión de **B** en el apartado **D)** del ejercicio **E.6.4** (pag. 193) en el que se realiza el diseño de un circuito decodificador de Decimal a BCD.?

R.3.6: Partimos de: $B = d_4 + d_5 + d_6 + d_7$

Aplicando el *postulado de la complementariedad* ($A + \bar{A} = 1$) y el de la *existencia del elemento neutro para el producto* ($A \cdot 1 = A$) resulta para d_7 :

$$B = d_4 + d_5 + d_6 + d_7 = d_4 + d_5 + d_6 + d_7(d_4 + d_5 + \overline{d_4 + d_5})$$

Si lo volvemos a aplicar a d_6 :

$$B = d_4 + d_5 + d_6 + d_7 = d_4 + d_5 + d_6(d_4 + d_5 + \overline{d_4 + d_5}) + d_7(d_4 + d_5 + \overline{d_4 + d_5})$$

Aplicando *De Morgan*: $\overline{d_4 + d_5} = \bar{d}_4 \bar{d}_5$ y sacando factor común ($d_6 + d_7$), obtenemos:

$$B = d_4 + d_5 + (d_6 + d_7)(d_4 + d_5) + (d_6 + d_7)(\bar{d}_4 \bar{d}_5)$$

Si sacamos factor común ($d_4 + d_5$) resulta:

$$B = (d_4 + d_5)(1 + d_6 + d_7) + \bar{d}_4 \bar{d}_5 (d_6 + d_7) = (d_4 + d_5) + \bar{d}_4 \bar{d}_5 (d_6 + d_7)$$

Aplicando *De Morgan*: $d_4 + d_5 = \overline{\bar{d}_4 \bar{d}_5}$, por tanto

$$B = \overline{\bar{d}_4 \bar{d}_5} + \bar{d}_4 \bar{d}_5 (d_6 + d_7)$$

Como $\overline{\bar{d}_4 \bar{d}_5} + \bar{d}_4 \bar{d}_5 = d_4 \bar{d}_5 + \bar{d}_4 d_5 + d_4 d_5 = 1$

Aplicando el *postulado de la complementariedad*:

$$\overline{\bar{d}_4 \bar{d}_5} = 1 - \bar{d}_4 \bar{d}_5 = \bar{d}_4 d_5 + d_4 \bar{d}_5 + d_4 d_5$$

Resultando:

$$B = \bar{d}_4 d_5 + d_4 \bar{d}_5 + d_4 d_5 + \bar{d}_4 \bar{d}_5 (d_6 + d_7)$$



P.3.7: ¿No entiendo cómo se obtiene el mapa de Karnaugh de la Fig. 6.4.5 (pag. 190) del apartado C) del ejercicio E.6.4, en el que se realiza el diseño de un circuito decodificador de BCD Exceso a 3 a Gray?

R.3.7: Para diseñar un decodificador de BCD a Gray nos basamos en las dos últimas columnas de la tabla de la figura 6.4.1. Si nos fijamos en la columna de G_3 , vemos que sólo tiene dos unos en los dos últimos términos mínimos. Pero, si observamos la columna de BCD Exceso a 3, vemos que faltan los términos mínimos: 0000, 0001, 0010, 1101, 1110 y 1111. Esto quiere decir que estos términos mínimos no se van a presentar nunca en la entrada del circuito, por lo tanto, los podemos considerar como indiferentes. Es decir, estos son términos mínimos que lo mismo pueden tomar el valor "0" que "1", puesto que no hay ninguna posibilidad de que se presenten en la entrada del decodificador, por lo que en el diagrama de Karnaugh ponemos x (ó d, o cualquier otra letra) y los usamos para minimizar junto con los "1". Así, resulta el diagrama de Karnaugh de la figura 6.4.5 y de ella obtenemos la expresión de G_3 .

La única limitación que tiene el uso de estos términos indiferentes es que no se pueden usar para formar grupos que estén formados sólo y exclusivamente por este tipo de términos, en un grupo siempre tiene que haber algún "1".



P.3.8: ¿Por qué para obtener la expresión de G_2 en el apartado C (decodificador de Exceso-3 a Gray) del ejercicio E.6.4 no se pueden agrupar los 8 términos mínimos $m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}$ de la parte inferior de la tabla a) de la figura 6.4.6?

R.3.8: Porque en este ejercicio no se está diseñando con puertas, sino con Multiplexos. Obsérvese que en este diseño cada fila del diagrama de Karnaugh se corresponde con una de las entradas del multiplexo, pues se han considerado como señales de control x_3 y x_2 , cuyos términos mínimos son los que encabezan las filas de dicho diagrama de Karnaugh.

Si nos centramos en la primera fila del diagrama de Karnaugh de la figura 6.4.6 vemos que no hay ningún elemento que tome el valor 1. Por tanto, la entrada del canal de entrada E0 del MUX que es el que se controla a $\bar{x}_3 \bar{x}_2$ es "0". Así mismo, en la segunda fila hay un "1" en el elemento correspondiente a la columna encabezada por 11, es decir, encabezada por el término mínimo $x_1 x_0$, Por lo tanto, la entrada del canal E1 del MUX que es el canal controlado por el término mínimo $\bar{x}_3 x_2$ debe ser $x_1 x_0$. Como se puede ver en el circuito que aparece en el apartado c) de la figura 6.4.6 esta función se implementa con una puerta AND.

En la tercera fila del diagrama, la encabezada por el término mínimo $x_3 x_2$, vemos que aparece un "1" en el elemento de la columna encabezada por $\bar{x}_1 \bar{x}_0$ y el resto de los elementos de esa fila son indiferentes por lo tanto los podemos agrupar dando lugar a un "1" y lo que supone que en la entrada del canal E3, controlado por $x_3 x_2$, debemos poner "1".

Finalmente, todos los elementos de la cuarta fila del diagrama, la encabezada por el término mínimo $x_3 \bar{x}_2$, son unos, lo que da lugar a un "1" en la entrada del canal E2, controlado por $x_3 \bar{x}_2$

Como hemos podido ver, para diseñar con Multiplexos la minimización se realiza por filas, ya que los términos mínimos que encabezan las filas del diagrama de Karnaugh coinciden con los términos mínimos que controlan los canales de entrada del Multiplexo.

De igual forma podíamos haber elegido que las señales de control del Multiplexo fueran los términos mínimos de las columnas con lo que, todo el razonamiento anterior es válido sin más que cambiar filas por las columnas. Sin embargo, lo que se suele hacer es estudiar antes el diagrama y elegir la configuración que se ve que se puede minimiza más. Es decir, si estudiamos la tabla de la figura 6.4.6 vemos que es conveniente elegir como señales de control del Multiplexo las filas de la tabla porque las dos últimas filas las podemos minimizar resultando un "1", mientras que si elegimos las columnas como señales de control del multiplexo, las señales de entrada de este son más complejas y hay que usar más puertas.



P.3.9: ¿Hay que estudiar los Amplificadores (Buffers-Drivers) y los Transmisores-Receptores de bus?

R.3.9: Si, hay que estudiarlos y entran en el examen.

Los Amplificadores o "Buffers Drivers" están descritos en el *apartado 3.5.2 del texto base (pag. 194 y 195)*, De su descripción sólo hace falta quedarse con el concepto, ya que su estructura interna no es objeto de estudio en esta asignatura. Para entender el concepto y usarlo no hace falta más que entender el primer párrafo y la *figura 3.20*.



P.3.10: ¿Si los Amplificadores (Buffers-Drivers) no son inversores y dejan la señal tal cual, para qué sirven?

R.3.10: Sirven, principalmente, para aumentar los niveles de corriente y para reconfigurar las señales.

Actúan como amplificadores digitales que adaptan impedancias y proporcionan los niveles de corriente adecuados para que no se confunda los estados. Por ejemplo, puede ocurrir que una señal se haya ido atenuando conforme se ha ido transmitiendo a lo largo de un circuito de muchas etapas de forma que, en un determinado punto el circuito trabaje con niveles de corriente más bajos de los necesarios lo que implica que se pueda confundir el "1" con el "0" y a la inversa, dependiendo del tipo de circuito. En este caso, la solución pasa por intercalar entre algunas de las etapas del circuito un amplificador de este tipo para reconstruir la señal y adaptar los niveles sin que introduzca ninguna función adicional.

Además, al ser dispositivos de tres estados, sirven para aislar la fuente de señal del circuito al que se le aplica dicha señal en función del valor de la señal de control.

Se utilizan, por ejemplo, en los circuitos de direccionamiento de memorias y en circuitos cuya interconexión se hace a través de una organización en buses.



P.3.11: ¿Hay distintos tipos de puertas de tres estados o sólo son Amplificadores?

R.3.11: Si, hay puertas de tres estados de todos los tipos, Inversores, no inversores, NAND, NOR, etc. y la señal de control puede actuar negada (circulito en la entrada de la señal de control, como el de la puerta NAND de la *figura 3.20*) y no negada (sin el circulito, como los de los no inversores de la *figura 6.19*).

Son puertas que, además de las entradas de los datos, poseen una entrada adicional para una señal de control haciendo que el funcionamiento de estas puertas dependa de dicha señal de control de forma que, si la señal de control es tal que la puerta está habilitada, en su salida aparece la función correspondiente al tipo de puerta, mientras que si está inhibida en su salida se presenta alta impedancia, lo que equivale a un circuito abierto.



P.3.12: ¿Cómo funcionan los Amplificadores (Buffers-Drivers) y qué diferencia hay entre los distintos tipos?

R.3.12: La descripción de su funcionamiento de forma resumida está en la página 333 *del texto*, en el párrafo que comienza por "Recordemos que los amplificadores de 3 estados ("3-state drivers") cuya síntesis ...".

De todas formas su funcionamiento es el siguiente: cuando la puerta está facilitada (la señal de control es "0" ó "1" dependiendo de cómo esté construida lo cual se conoce porque la señal de control tiene un circulito de negación o no lo tiene) su salida es la correspondiente a la función que debe realizar dicha puerta sobre las entradas y que depende del tipo de puerta (NOR, NAND, Inversor, No Inversor, etc.). Es decir funciona como tal.

Cuando la puerta no está facilitada (la señal de control es "1" o "0", contrario al caso anterior) la salida de la puerta presenta alta impedancia (tercer estado). Esto es, su salida es equivalente a un circuito abierto, no es ni "1" ni "0" (que son los datos), es como si no existiera.



P.3.13: ¿Qué función realizan los buffers drivers de la figura 6.19?

R.3.13: No realizan ninguna función sobre el dato de entrada, simplemente lo dejen pasar o no en función de la señal de control, \overline{IG} . Así, si la entrada de control \overline{IG} se pone "1" los 4 buffers presentan en su salida alta impedancia y si se pone a "0" cada buffer presenta en su salida la señal que tiene en su entrada.



P.3.14: ¿Los Buffers Drivers que aparecen en la fig. 6.20 de la pag. 332 del texto invierten la señal de entrada?.

R.3.14: Si, son inversores de tres estados. Es decir, son circuitos que en su salida presentan la señal complementada de la que hay en cada momento en su entrada, siempre que la señal de control esté en alta. Si la señal de control está en baja, en su salida presenta alta impedancia (el tercer estado).

En la descripción de la tabla de verdad de la figura 6.20 aparece la frase “Datos de B pasan a A”, pero como los Buffers Drivers son inversores realmente son los datos negados de B los que pasan a A. De igual forma, en la siguiente fila también son los datos negados de A los que pasan a B.



P.3.15: Hay distintos tipos de Amplificadores o “Buffers Drivers”¿Cuáles son estos tipos y cómo funcionan?

R.3.15: En efecto, hay unos circuitos que en la salida presentan la misma señal de entrada y otros que en su salida la presentan complementada, es decir actúan también como inversores. Si combinamos los dos tipos de funciones que pueden realizar y las dos formas de controlar su funcionamiento, obtenemos 4 tipos diferentes. A continuación presentamos los 4 circuitos y sus correspondientes tablas de verdad.

