

**Grado en Ingeniería de Tecnologías de Telecomunicación.
Escuela Técnica Superior de Ingeniería Industrial y de Telecomunicación.
Electrónica Digital I.**

Práctica nº 8. Diseño lógico con módulos combinacionales.

En esta práctica se realizará el montaje de circuitos combinacionales implementados mediante módulos combinacionales y puertas lógicas.

El primer circuito consiste en el diseño de un circuito sumador/restador, mientras que el segundo diseño corresponde a la implementación de una función lógica usando un circuito multiplexor.

Todo el trabajo de la práctica debe realizarse en el directorio (o carpeta) Pr8 (o similar), que debe crearse en el directorio de trabajo de cada alumno, donde deben guardarse los ficheros utilizados o generados durante la práctica, usando subdirectorios para cada apartado.

8.1. Diseño de un circuito sumador/restador.

En este apartado se va a realizar el diseño, montaje y verificación de un circuito sumador/restador para números con signo de 4 bits en complemento-2. Los operandos son dos números X e Y de 4 bits en complemento-2 y la salida Z es otro número de 4 bits en complemento-2. Además se necesita una entrada de control S/R que determina si se realiza la suma o la resta. Debido a que la salida y las entradas son de 4 bits puede producirse desbordamiento en las operaciones, lo que debe tenerse en cuenta a la hora de evaluar los resultados producidos. Recordar que en 4 bits el rango en complemento 2 es $[-8, +7]$, por lo que operaciones del tipo $(5 + 7)$ o $(-3 - 6)$, por ejemplo, dan resultados que no pueden almacenarse en una salida de 4 bits. El circuito de detección de desbordamiento se diseñará usando un multiplexor de 8 entradas en implementación de tipo 0: entradas de selección del multiplexor a las entradas del problema, y entradas de datos del problema a 1 (Vcc) o 0 (GND).

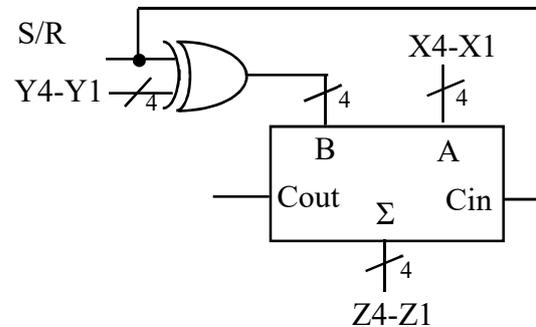
El diseño del circuito sumador/restador se explica en las clases teóricas. Se desarrolla en base a un sumador, teniendo en cuenta que la operación de resta en complemento-2 se puede definir como una suma $X \text{ MINUS } Y = X \text{ PLUS } (Y)_{2,c}$; y que $(Y)_{2,c} = \text{NOT}(Y) \text{ PLUS } 1$. Por ello se puede hacer el diseño en base a la señal de control S/R:

- $S/R = 0 \Rightarrow$ Suma $\Rightarrow Z = X \text{ PLUS } Y = X \text{ PLUS } Y \text{ PLUS } 0$.
- $S/R = 1 \Rightarrow$ Resta $\Rightarrow Z = X \text{ MINUS } Y = X \text{ PLUS } \text{NOT}(Y) \text{ PLUS } 1$.

Según esto, la entrada X pasa siempre directamente a una entrada del sumador ($A = X$); la entrada Y si S/R es 0 pasa directa y si S/R es 1 pasa complementada, luego la entrada B del sumador es (para cada bit):

$$B = \overline{S/R} \cdot Y + S/R \cdot \overline{Y} = S/R \oplus Y$$

Para sumar 0 o sumar 1 se utiliza la entrada Cin del sumador: si S/R es 0 se necesita que el acarreo de entrada del sumador Cin sea 0, y si S/R es 1 se necesita que Cin sea 1, luego Cin se debe conectar a S/R. El circuito queda:



El desbordamiento Ov de la operación es función de los bits de signo de los operandos del sumador A y B y del resultado Z, de forma que hay desbordamiento solo si los signos de A y B son iguales, pero el signo de Z es diferente. Luego $Ov = F(A(4), B(4), Z(4))$. Para implementar Ov con un multiplexor de 8 entradas habilitado, conectar A(4) a S2, B(4) a S1 y Z(4) a S0, y las entradas de datos del multiplexor I7-I0 al valor lógico del *minterm* correspondiente: 0 (GND) o 1 (Vcc)

El diseño y prueba del circuito se hará mediante estos pasos:

- Diseñar el circuito usando un sumador 74LS83 y puertas EXOR 74LS86 para la operación aritmética y un multiplexor de 8 entradas 74LS151 para el circuito de desbordamiento. El circuito de desbordamiento debe simularse, pero no debe montarse, solo se montará en la regleta el circuito sumador/restador. Buscar las hojas de características para obtener las conexiones del circuito.

- Simular la operación del circuito con Circuit Maker usando un *switch* (*hotkey s*) para la entrada S/R y un generador de secuencia (*hotkey G*) para las entradas X e Y. Los circuitos 74LS83 y 74LS151 pueden encontrarse en *Digital by Number*, en las subclases 74xx y 74lxx) Cargar el *Data Sequencer* con al menos 10 operandos distintos en X e Y, y comprobar el funcionamiento para la suma y la resta, usando displays hexadecimales (*hotkey h*), para las entradas X, Y, y la salida Z, y un display lógico (*hotkey 9*) para la salida Ov. Situar *Step Size* a 10 *Ticks* y simular paso a paso pulsando en el icono . Una metodología de prueba similar se realizó en el apartado 4 de la práctica 2 (simulación de un circuito aritmético).

Hay que tener en cuenta que los números en complemento-2 se aplicarán y observarán normalmente en formato hexadecimal donde 0-7 corresponden a esos mismos números positivos y el resto de valores hexadecimales corresponden a números negativos:

8 => -8; 9 => -7; A => -6; B => -5; C => -4; D => -3; E => -2; F => -1.

- Montar el circuito sobre una regleta. Se recomienda tener preparado un esquema con las conexiones antes de la sesión de prácticas, usando los *datasheets* de los componentes y definiendo sobre ellos donde van a estar cada entrada y cada salida. Los circuitos deberían colocarse relativamente cerca en la placa, y se deberían usar cables cortos para hacer las conexiones. No hay que olvidarse de realizar las conexiones de los circuitos a alimentación y tierra.

- Verificar el funcionamiento del circuito con el analizador lógico LA5240 para las operaciones de suma y de resta. No se debe montar el circuito de desbordamiento. Aplicar dos veces la secuencia de prueba con el generador a los operandos de entrada A y B, una con la entrada S/R conectada a GND (suma) y la otra con la entrada S/R conectada a la tensión de alimentación (resta), comprobando que las operaciones se realizan correctamente, y teniendo en cuenta que puede haber desbordamiento (en desbordamiento el resultado no es correcto). Leer las respuestas del circuito con el analizador lógico.

La conexión del analizador lógico al circuito y su programación son similares a la realizadas durante el apartado 2 de la práctica 1 (verificación de un sumador), con la salvedad de que no se tiene entrada de acarreo Cin y la salida Z es solo de 4 bits (no se utiliza el acarreo de salida). Se puede, por tanto, realizar casi las mismas conexiones entre el circuito:

- Conectar cables a los 8 canales del *pod* 3. Fijar una pinza en sus otros extremos y conectar de la siguiente manera: canales 0-3 (de 0 a 3) del *pod* 3 a Y1-Y4 (0 a Y1, 1 a Y2, 2 a Y3 y 3 a Y4), canales 4-7 (de 4 a 7) del *pod* 3 a (4 a X1, 5 a X2, 6 a X3 y 7 a X4). Conectar un cable a uno de los canales marcados como GND del *pod* 3, conectar el otro extremo del cable a un conector de una misma pinza, y conectar la pinza a GND del circuito.

- Conectar cables a los 4 primeros canales del *pod* 1. Conectar los canales 0-3 (de 0 a 3) del *pod* 1 a las salidas S1 (Z1), S2 (Z2), S3 (Z3) y S4 (Z4) del sumador, fijando una pinza en el otro extremo del cable. Conectar un cable a uno de los canales marcados como GND del *pod* 1, conectar su otro extremo a una pinza y conectar la pinza a GND del circuito (o conectar el cable a otros conectores libres de los canales GND del *pod* 3).

- Conectar un cable entre el canal 1 del *pod* 4 con el canal 5 del *pod* 1. Este canal se utilizará para disparar (o sincronizar) las direcciones de los datos cargados en el generador de patrones con las direcciones donde se leerán las respuestas con el analizador lógico. Conectar un cable a uno de los canales marcados como GND del *pod* 4, conectar su otro extremo a una pinza y conectar la pinza a GND del circuito (o conectar el cable a otros conectores libres de los canales GND de los *Pods* 1 o 3).

Encender el analizador, arrancar el programa de control y cargar la programación hecha durante el apartado 2 de la práctica 1 (verificación de un sumador): ficheros SumI.ini de configuración (utilizar el comando *Load Settings* del menú **File**) y SumD.Dso de datos (utilizar el comando *Load Data* del menú **File**) y realizar esta pequeña modificación:

- Ejecutar el comando *Group Edit* del menú **View**. Sobre la ventana que aparece seleccionar con los botones  el grupo cuyo campo *Name* es Suma: cambiar el nombre a Z y fijar su

campo *Base* a valor HEX y *Number* a valor 4 bits en lugar de 5 bits. Seleccionar los grupos A y B, y fijar también su campo *Base* a valor HEX y cambiar sus nombres a X e Y, respectivamente.

- Asegurarse de que la marca roja de disparo está sobre el pulso cargado en el canal 1 del *pod* 4. Si no lo está, hay que moverla allí con el ratón.
- Guardar la nueva configuración en el fichero SR_Pr8.ini (utilizar el comando *Save Settings* del menú **File**) en el directorio de trabajo de cada grupo.

Conectar la entrada S/R a GND para realizar la suma, pulsar con el ratón en el icono **GO**, y realizar una captura, con lo que se deben medir valores en los canales conectados a las salidas. Comprobar que todas las salidas toman valores 0 y 1, si no repasar las conexiones. Examinar las respuestas como lista de estados. En el menú en línea **Timing** desactivar el campo *Timing 1 <- data1* y activar el campo *Timing 2 <- data1*. En el menú **View** activar el campo *State of logic analyzer*. Situar con el cursor vertical a la izquierda de las medidas las direcciones visibles donde están las medidas (a partir de la posición 90). Comprobar que los valores bajo las columnas A y B son los que deberían ser para cada dirección y comprobar que las medidas tomadas indican que el circuito suma (recordar que las salidas aparecen desplazadas en la dirección anterior o posterior). Si las sumas no son correctas repasar las conexiones. Una vez que los resultados sean correctos, guardar las medidas tomadas en el fichero Suma_Pr8.Dso en el directorio de trabajo de cada grupo, con la orden *Save Data As...* del submenú *Save Data* del menú **File**.

Conectar la entrada S/R a Vcc para realizar la resta, pulsar con el ratón en el icono **GO**, y realizar una captura, repitiendo el proceso realizado para la suma. Una vez que los resultados sean correctos, guardar las medidas tomadas en el fichero Resta_Pr8.Dso en el directorio de trabajo de cada grupo, con la orden *Save Data As...* del submenú *Save Data* del menú **File**.

8.2. Implementación de una función lógica con un circuito multiplexor.

En este ejercicio se van a implementar una función lógica con ayuda de un circuito multiplexor y una puerta NOT. La función lógica a implementar es

$$F(A,B,C,D) = \sum(1,2,5,7,9,10,14) + \sum\phi(0,15)$$

considerando todas las entradas y la salida en polaridad positiva (.H).

- Diseñar el circuito usando un multiplexor de 8 entradas 74LS151 y una puerta EXOR 74LS86 que opere como inversor. Buscar las hojas de características para obtener las conexiones del circuito.
- Simular la operación del circuito con Circuit Maker usando una *Hex Key (hotkey H)* para aplicar la tabla de verdad al circuito y un display lógico (*hotkey 9*) para la salida, comprobando que el circuito funciona correctamente.

- Montar el circuito sobre una regleta. Se recomienda tener preparado un esquema con las conexiones antes de la sesión de prácticas, usando los *datasheets* de los componentes y definiendo sobre ellos donde van a estar cada entrada y cada salida. Los circuitos deberían colocarse relativamente cerca en la placa, y se deberían usar cables cortos para hacer las conexiones. No hay que olvidarse de realizar las conexiones de los circuitos a alimentación y tierra.

- Verificar el funcionamiento del circuito con el analizador lógico LA5240 mediante la prueba exhaustiva de toda la tabla de verdad. La programación del analizador lógico es similar a la realizada durante la práctica 3, en el que se probó un circuito combinacional de 4 entradas y una salida, con la salvedad de los nombres y grupos utilizados para las entradas. Se puede, por tanto, realizar las mismas conexiones entre el circuito y el LA5240:

- Utilizar 4 canales del generador de patrones (*pod* 3, canales 16-19) para programar los estímulos a aplicar al circuito. Por ejemplo, conectar el canal 19 a A, 18 a B, 17 a C y 16 a D.
- Utilizar un canal del analizador lógico (*pod* 1, canal 0) conectado a la salida del circuito, y otros cuatro canales más para leer también las entradas del circuito en los canales 4-1 del *pod* 1 (4 a A, 3 a B, 2 a C, y 1 a D), conectándolos a la misma pinza que la del generador de patrones (cada pinza tiene dos conectores).

Cargar la programación del analizador lógico hecha en la práctica 3: ficheros *FuncionI.ini* de configuración (utilizar el comando *Load Settings* del menú **File**) y *FuncionD.Dso* de datos (utilizar el comando *Load Data* del menú **File**), cambiando los grupos:

- Crear solo dos grupos (comando *Edit Group* del menú **View**): uno para las entradas (en el *Group 0*) de nombre ABCD en decimal (DEC) y de 4 bits con los canales 4 (Bit 3), 3 (Bit 2), 2 (Bit 1) y 1 (Bit 0) para las entradas A, B, C y D, respectivamente, y otro para la salida (en el *Group 1*) de nombre Z en binario (BIN) de 1 bit con el canal 0 (Bit 0).

Pulsar con el ratón en el icono , y realizar una captura. Comprobar que todos los canales 0-4 toman valores 0 y 1, si no reparar las conexiones. Examinar las respuestas tanto como forma de ondas como lista de estados (en el menú en línea **Timing** desactivar el campo *Timing 1 <- data1* y activar el campo *Timing 2 <- data1*; en el menú **View** activar el campo *State of logic analyzer*). Comprobar que la función lógica es correcta. Si no lo es, reparar las conexiones. Una vez que los resultados sean correctos, guardar en el directorio de trabajo de cada grupo la nueva configuración en el fichero *SR_Pr8.ini* (utilizar el comando *Save Settings* del menú **File**) y guardar las medidas tomadas en el fichero *FuncionI_Pr8.Dso* (utilizar el comando *Save Data As...* del submenú *Save Data* del menú **File**).

Para finalizar:

- Abandonar el programa con la orden *Exit* del menú **File**. Apagar el analizador lógico.
- Soltar la alimentación, desconectar los cables conectados al circuito y a los *pods* con cuidado para no romperlos, y desmontar los dispositivos de la regleta.