



PRÁCTICA 5

CIRCUITOS COMBINACIONALES: CIRCUITOS ARITMÉTICOS. SUMADORES

OBJETIVOS:

Una vez finalizada la práctica deberemos ser capaces de:

- ⦿ Describir el funcionamiento, dibujar el diagrama lógico e implementar un semisumador y un sumador completo.
- ⦿ Utilizar sumadores completos para implementar sumadores binarios en paralelo.
- ⦿ Comprender los problemas de la propagación del acarreo en los sumadores binarios con *acarreo propagado*.
- ⦿ Comprender el funcionamiento de un circuito sumador/restador.
- ⦿ Diseñar circuitos sumadores/restadores que operen en complemento a 2.

REFERENCIAS:

- ⦿ T.L. Floyd, *Fundamentos de Sistemas Digitales*, 9ª Edición, Capítulo 6, “Funciones de la Lógica Combinacional”, secciones 6-1 a 6-3.
- ⦿ Transparencias Tema 4 “Sistemas Digitales”. Fundamentos de los Computadores.
- ⦿ J.Mª Angulo & J. García, “*Sistemas Digitales y Tecnología de Computadores*”, Capítulo 6, “Elementos Aritméticos”.

ELEMENTOS NECESARIOS:

- ⦿ Programa de Simulación LogiSim 2.7.1 con:
 - ✗ Diferentes tipos de puertas lógicas.
 - ✗ Circuitos sumadores completos.
 - ✗ Indicadores (ver) y displays de 7 segmentos (*Hex Digit Display*).



INTRODUCCIÓN TEÓRICA

En esta práctica introduciremos la utilización de uno de los elementos más importantes no solamente en los ordenadores, sino en la mayoría de tipos de sistemas digitales en los que se procesan datos numéricos.

El circuito básico es el *semisumador*. Un semisumador admite dos dígitos binarios en sus entradas y produce dos dígitos binarios en sus salidas: un bit de suma y uno de acarreo.

Puesto que una suma binaria necesitará en la mayoría de las ocasiones tener presente un posible acarreo procedente de una etapa anterior, se hace necesario un circuito que lo contemple: el *sumador completo*. El sumador completo tendrá, por tanto, tres entradas y generará una salida de suma y una de acarreo. Puesto que la suma es asociativa, siempre podremos construir un sumador completo utilizando semisumadores.

Un sumador paralelo de 4 bits básico se puede implementar mediante cuatro sumadores completos. La salida de acarreo de cada sumador se conecta a la entrada de acarreo del siguiente sumador de orden superior. Estos acarreos se denominan acarreos internos y, por tanto, se irán propagando de uno a otro, lo que ralentiza su funcionamiento. A este tipo de sumadores se les denomina de *acarreo propagado*.

REALIZACIÓN PRÁCTICA

1. Se desea diseñar un circuito combinacional capaz de realizar la resta de dos números de dos bits. La salida debe mostrar en todo momento el valor absoluto del resultado.
 - a) Realiza la tabla de verdad de dicho circuito.
 - b) Utilizando tablas de Karnaugh simplifica la tabla anterior y obtén las expresiones de las salidas.
 - c) Implementa el circuito utilizando puertas básicas.
 - d) Comprueba su funcionamiento introduciendo diferentes combinaciones. Visualiza las salidas conectando en ellas un display de 7 segmentos con entrada hexadecimal (*Hex Digit Display*). Utiliza un separador para introducir los bits de forma individual y conecta a 0 las entradas del display que no utilices.
2. Utilizando sumadores:
 - a) Construye un sumador paralelo de 4 bits con entrada y salida de acarreo
 - b) Utilizando dos de estos sumadores construye un sumador de 8 bits y comprueba su funcionamiento.
3. El circuito restador de la figura 4.1 está formado por una serie de sumadores binarios que, apoyándose en los inversores, se encargan de realizar resta de dos números (X , Y) expresados en complemento a 2.

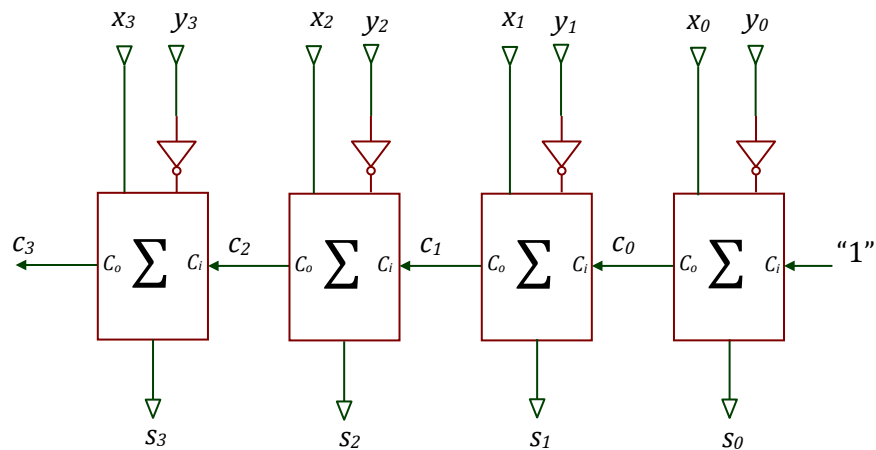


Figura 4.1 – Circuito Restador en Complemento a 2

- Implementa este circuito y comprueba su correcto funcionamiento poniendo ejemplos de casos de valores positivos y negativos tanto para X como para Y .
- Indica el rango del resultado.
- Sustituye los inversores por puertas XOR y convierte el circuito en un sumador/restador de 4 bits. Vuelve a poner diferentes ejemplos de operaciones de suma y resta con valores positivos y negativos tanto de X como de Y .
- Desde el punto de vista aritmético ¿Qué condiciones se tienen que dar para que se produzca overflow?