

A.2. Actividades de Simulación asociadas al tema 2: Lógica Combinacional (I): Funciones Aritmético-Lógicas

Los temas 2, 3 y 4 estudian los circuitos lógicos combinacionales (segunda unidad temática). Estos temas no precisan comentarios puesto que no intervienen conocimientos ajenos a los expuestos en el texto y únicamente requieren como formación previa las tablas de verdad, la minimización de funciones lógicas, el álgebra de Boole y las codificaciones binarias, todos ellos sencillos. A partir de aquí y hasta llegar a temporizadores y memorias, los circuitos se caracterizan sólo por sus tablas de verdad. Primero se estudia la síntesis con puertas (integración en baja escala, S.S.I.) y después la solución M.S.I. (integración en media escala).

Recordemos que hay esencialmente tres tipos de funciones en lógica combinacional:

- Circuitos cambiadores de código
- Funciones aritmético-lógicas (ALU)
- Funciones de ruta de datos

En este tema estudiamos los circuitos cambiadores de código y las operaciones aritmético lógicas. El primer apartado surge como consecuencia de la necesidad de representar los números positivos y negativos en un rango de valores limitado. Así en la primera actividad de este tema implementaremos el convertidor de código de S-M a C-1 sintetizado con puertas y a continuación analizaremos el funcionamiento de dos convertidores de código integrados que convierten de BCD a Binario y a la inversa.

A continuación pasaremos a la simulación de las funciones aritmético-lógicas que son las encargadas de realizar operaciones aritméticas o lógicas sobre los datos, transformándolos y produciendo un resultado local que, en general, será usado en otra parte del sistema digital y cuya transferencia será realizada de forma controlada por los circuitos de ruta de dato (apartado siguiente).

A.2.1. Circuitos Convertidores de Código

Simulación de los siguientes circuitos convertidores:

A.2.1.1. Convertidor de código de S-M (Signo y Magnitud) a C-1 (complemento a 1) usando puertas lógicas. El diseño de este circuito se encuentra la pag. 268-269 y figura 5.4 del texto base.

A.2.1.2. Convertidor de código de BCD a BINARIO integrado (SN74184).

A.2.1.3. Convertidor de código de BINARIO a BCD integrado (SN74185a)

NOTA: Para entender el funcionamiento de las dos simulaciones de los convertidores BCD a Binario y Binario a BCD se recomienda estudiar las hojas de características de los circuitos SN74184 y SN74185A centrando la atención en el bit menos significativo (se encuentra en el Curso Virtual de la Asignatura).

A.2.2. Sumadores y Restadores

Diseño y simulación de los siguientes circuitos:

A.2.2.1: Semisumador con puertas XOR (SN7486) y AND (SN7408) (Pag. 270, fig. 5.5).

A.2.2.2: Sumador completo para dos entradas de un bit y arrastre, a partir del semisumador del apartado anterior (Pag. 271, fig. 5.6).

A.2.2.3: Semirestador con puertas (Pag. 274, fig. 5.8).

A.2.2.4: Restador completo para dos entradas de un bit y arrastre, a partir del semirestador del apartado anterior (Pag. 274, fig. 5.9)

A.2.3. Comparadores

Diseño y simulación de los siguientes circuitos:

A.2.3.1 Comparador de un bit (Pag. 283, fig. 5.16).

A.2.3.2 Comparador de dos palabras de 2 bits (circuito simplificado del de la fig. 5.17 del texto).

A.2.4. Detector de paridad

Diseño y simulación de los siguientes circuitos:

A.2.4.1 Detector de paridad de 4 bits (Pag. 290, fig. 5.21).

A.2.4.2 Detector de paridad de dos palabras de 4 bits utilizando el detector anterior como módulo de diseño.

A.2.5. ALUs (SN74181)

Como podemos observar la ALU opera con palabras de 4 bits y esto hace que las tablas de verdad sean muy extensas y laboriosas de construir por lo que aconsejamos simplificar los cálculos. Para ello se recomienda que los bits más significativos se pongan a "0" y sólo se usen relojes para los bits menos significativos de las palabras de entrada. Por ejemplo, se pueden usar relojes para generar las señales correspondientes a los bits A0 y A1 de la palabra A y para los bits B0 y B1, de la palabra B mientras que los bits más significativos de ambas palabras (A2, A3 y B2, B3) se pueden poner a cero. Sin embargo, es importante observar todas las salidas de la ALU cuando se realizan operaciones aritméticas porque en estas operaciones hay acarreo y aunque se pongan los bits más significativos de los datos de entrada a "0" nos encontramos que, en la mayoría de los casos, las salidas más significativas no están a cero como podemos pensar inicialmente y de forma errónea.

A continuación se proponen que se simulen tres funciones con ALU. Hemos elegido una función lógica y dos aritméticas una con acarreo y otra sin acarreo, sin embargo el alumno puede simular cualquiera de las 48 funciones aritmético-lógicas posibles (Pag. 292, fig. 5.23).

A.2.5.1 Programación de la ALU para que realice la función lógica: $A \oplus B$

A.2.5.2 Programación de la ALU para que realice la función aritmética (sin acarreo):

$A \text{ PLUS } \overline{AB}$

A.2.5.3 Programación de la ALU para que realice la función aritmética con acarreo:

$A \text{ MINUS } B$