

Práctica 3: Circuitos combinacionales

Organización y Arquitectura de Computadoras

Hernández Ferreiro Enrique Ehécatl
López Soto Ramses Antonio

24 de febrero de 2019

1 Introducción

A lo largo de los años nos hemos preguntado cómo es que las computadoras funcionan si no piensan, no razonan, sólo realizan las series de instrucciones que nosotros les indicamos. Esto se responde con la lógica proposicional.

Pero, ¿la lógica proposicional? En el año de 1854 un hombre llamado George Boole desarrolló el *álgebra booleana* que abrió camino para el desarrollo de la *lógica digital*, con la cual nuestras computadoras funcionan.

2 Desarrollo

Para el desarrollo de los circuitos se utilizó el software *Logisim*. La primera parte consistió en el desarrollo de la implicación lógica y la segunda en la construcción de un comparador de dos bits.

2.1 Implicación lógica

Para la construcción de la implicación lógica se ocuparon:

- 2 pines de entrada
- 1 pin de salida
- 3 transistores tipo PNP
- 3 transistores tipo NPN

2.2 Comparador

Para la construcción del comparador se usó lo siguiente:

2.2.1 Fórmulas

El circuito funciona y se comporta como se muestra en la siguiente tabla, donde $A = A_0A_1$ y $B = B_0B_1$:

A_0	A_1	B_0	B_1	$A = B$	$A > B$	$A < B$
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

Table 1: Tabla de verdad del circuito

Usando la tabla de valores de verdad para los valores del bit A y B, sabemos que los números serán: (00 = 0, 01 = 1, 10 = 2, 11 = 3) entonces podemos establecer una relación entre A y B, en las celdas donde se cumpla la relación tendremos un 1 y donde no un 0, entonces procederemos a formar una expresión booleana.

Las expresiones booleanas en cada uno de los casos son:

- $(A = B) = \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1}$
- $(A > B) = \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1}$
- $(A < B) = \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0B_1}$

2.2.2 Mapas de Karnaugh

Los mapas de karnaugh para minimizar las fórmulas son:

- $A = B$

		B_0B_1			
		00	01	11	10
A_0A_1	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

Por lo que $(A = B) = \overline{A_0A_1B_0B_1} + \overline{A_0A_1B_0}B_1 + A_0\overline{A_1B_0B_1} + A_0A_1B_0B_1$

- $A > B$

		B_0B_1			
		00	01	11	10
A_0A_1	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	10	1	1	0	0

Por lo que $(A > B) = A_1\overline{B_0B_1} + A_0A_1\overline{B_1}$

- $A < B$

		B_0B_1			
		00	01	11	10
A_0A_1	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

Por lo que $(A < B) = \overline{A_0}\overline{A_1}B_1 + \overline{A_0}B_0 + \overline{A_1}B_0B_1$

3 Conclusiones

La construcción de circuitos digitales es una tarea importante para quienes se dedican a la fabricación de computadoras. A pesar que nosotros nos enfocamos más en el cómo nunca es malo saber un poco de cómo funcionan los componentes de nuestras computadoras.

En resumen, la práctica nos ayudó a entender un poco mejor el comportamiento de algunos componentes de nuestras herramientas de trabajo.

3.1 Preguntas

1. ¿Cuál es el procedimiento a seguir para desarrollar un circuito que resuelva un problema que involucre lógica combinacional?

R. Hacer una tabla de verdad con todos los casos posibles, extraer aquellas combinaciones que den el resultado deseado, juntar las expresiones y reducir usando mapas de Karnaugh.

2. Si una función de conmutación se evalúa a más ceros que unos ¿es conveniente usar mintérminos o maxtérminos? ¿En el caso que se evalué a más unos que ceros?

R. Más ceros usamos mintérminos y para más unos usamos maxtérminos.

3. Analizando el trabajo realizado, ¿cuáles son los inconvenientes de desarrollar los circuitos de forma manual?

R. Que muchas veces abstraes un funcionamiento mentalmente pero resulta en fallas que no consideraste además de ser más complicado y se causan cortos más fácilmente.