

Entrega de Lógica Digital

March 31, 2015

Lógica proposicional

El funcionamiento de la computadora se basa en el procesamiento y almacenamiento de datos binarios. Se necesitan elementos (dispositivos, aparatos) que puedan almacenar y mantener un valor binario (uno de dos estados) y otros elementos que puedan operar sobre datos (cadenas) binarias bajo el control de señales y así realizar las funciones elementales de la computadora. Estos dispositivos pueden construirse a partir de la lógica digital, cuyo fundamento matemático es la lógica proposicional.

¿Que son las proposiciones? Son enunciados en lenguaje natural que pueden ser verdaderas o falsas. Las proposiciones pueden ser simples, como por ejemplo $A = \text{"hoy es viernes"}$ o $B = \text{"hoy está lloviendo"}$, o pueden ser más complejas, componiendo proposiciones simples, por ejemplo: *"hoy es viernes y hoy está lloviendo"*. Estas proposiciones compuestas pueden ser difíciles de analizar sin un mecanismo formal o estructurado. Para facilitar esto, utilizaremos una herramienta estructurada que es la tabla de verdad.

Las tablas de verdad son una herramienta formal para analizar las expresiones mediante la enumeración de sus posibles casos, es decir: las diferentes posibilidades en cuanto a la asignación de valores de verdad (verdadero o falso) a las variables, que son las expresiones elementales dentro de la expresión compleja. En el ejemplo anterior, las variables son A y B, y en cada caso se relaciona una combinación de valores en esas variables con el valor final de la expresión.

Relación con la lógica digital

En 1938, un tal Shannon del MIT, sugirió utilizar la lógica digital (álgebra de boole) para diseñar circuitos de conmutación y esta idea fue empleada posteriormente en el diseño de circuitos electrónicos. De esta manera, el fundamento matemático es útil para:

- describir el funcionamiento del dispositivo digital
- simplificar la expresión para implementar circuitos más simples

A partir de esto, es posible pensar la lógica proposicional como una herramienta para resolver problemas y la lógica digital como una herramienta que permite automatizar esas soluciones.

Entonces es necesario construir dispositivos que implementen las funciones lógicas y luego puedan componerse. Las compuertas lógicas elementales son:

- AND (conjunción, y, $\bullet$, $\wedge$)
- OR (disyunción, o, $+$, $\vee$)
- NOT (negación, no, $\neg$, $\bar{p}$)

Las compuertas se conectan entre sí formando circuitos digitales, que traducen un conjunto de entradas en una salida, y la salida se actualiza de inmediato cuando cambian las entradas. De esta manera, el circuito permite “decidir” o “resolver” problemas.

Ejemplo de construcción de un circuito

Se necesita un circuito (de nombre 2VF) que compare dos entradas (que son las variables de verdad) y que indique si ambas son iguales: ambas verdaderas o ambas falsas (ver Figura 1).

El primer paso hacia la construcción del circuito es la creación de la tabla de verdad, para esto se deben enumerar todas las posibles combinaciones de valores de verdad entre $e1$ y $e2$, y definir en cual de ellos la salida debe ser 1. En este caso es cuando ambas variables tienen el mismo valor de verdad.

e_1	e_2	s
0	0	1
0	1	0
1	0	0
1	1	1

Table 1: Tabla de verdad para el circuito 2VF

A partir de la tabla es posible deducir su fórmula de verdad usando la técnica **SOP** (*Sum of Products*), donde se describe la expresión a partir de los casos que hacen verdadera. Esto es, elegir los casos donde la salida es **1** y conectarlos mediante una conjunción.

e_1	e_2	s
0	0	1 $\bar{e}_1 \bullet \bar{e}_2$
0	1	0
1	0	0
1	1	1 $e_1 \bullet e_2$

$$s = (\bar{e}_1 \bullet \bar{e}_2) + (e_1 \bullet e_2)$$

Table 2: SOP del circuito 2VF

Dado que la expresión SOP del circuito está construida con compuertas elementales, el ensamblado es una tarea literal (ver Figura 2)

Este método de derivación de circuitos es robusto por construcción, dado que la tabla de verdad es un método exhaustivo para enumerar los posibles casos.

1 Ejercicios

Para cada uno de los circuitos, se pide:

1. Dibujar la caja negra del circuito
2. Construir la tabla de verdad
3. Construir la SOP
4. Dibujar el circuito

1.1 Max3

Realizar un circuito **MAX3**, de 3 entradas que calcule la función mayoría: Si dos o más entradas valen 1, se obtiene un uno. En caso contrario se obtiene un 0.

1.2 Mux

Se necesita construir un circuito **MUX2** que permita seleccionar una de sus dos entradas para proyectarla en la salida, en función de una línea de control.

Este tipo de operaciones se denomina **multiplexación**. Si la línea de control vale 1, se proyecta la entrada e0, y en caso contrario se proyecta e1. Es decir: si $c=0$ entonces $s=e0$, si $c=1$ entonces $s=e1$

Importante: La línea de control es otra entrada mas!

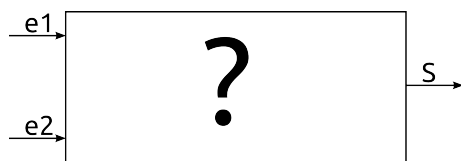


Figure 1: Esquema del circuito 2VF

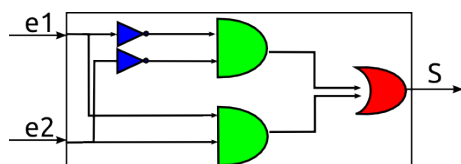


Figure 2: Circuito 2VF