

Organización y Arquitectura de Computadoras 2020-2

Práctica 2: Circuitos Combinacionales

Berenice Calvario
Toprak Memik

October 27, 2020

1 Objetivos

Genereales:

- El alumno aprenderá a diseñar y simular circuitos combinacionales.

Particulares: Al finalizar la práctica el alumno tendrá la capacidad de:

- Llevar a cabo el proceso de diseño de funciones de conmutación para resolver problemas lógicos simples.
- Utilizar las funciones básicas de *Logisim* para diseñar circuitos combinacionales.
- Realizar simulaciones de circuitos combinacionales con *Logisim*.

2 Requisitos

- **Conocimientos previos:**

- Función de conmutación.
- Minimización de funciones de conmutación por medio de manipulación algebraica y mapa de Karnaugh.
- Los componentes básicos del diseño de circuitos combinacionales: transistores y compuertas **AND**, **OR** y **NOT**.

Se puede consultar los temas en [Mano] y [Patterson].

- **Tiempo de realización sugerido:** 5 horas.
- **Número de colaboradores:** 2 personas.

- **Software a utilizar:**

- *Java Runtime Environment* versión 5 o superior.
- El paquete *Logisim* [**Logisim**].

3 Planteamiento

Para encontrar el funcionamiento de las computadoras modernas a un nivel general y antes de ser capaces de desarrollar nuevas arquitecturas, es necesario comprender el funcionamiento en el nivel de abstracción más bajo: los circuitos electrónicos. En la actualidad los circuitos electrónicos de una computadora se encuentran constituidos principalmente por transistores, por lo que es necesario conocer sus capacidades y limitaciones. El primer paso, entonces, es comenzar a diseñar circuitos combinatoriales con los componentes esenciales, además, antes de dar por finalizado el diseño, es de gran utilidad realizar pruebas por medio de simulaciones para detectar posibles errores, evitando realizarla gastos económicos y de tiempo manufacturando componentes defectuosos.

4 Desarrollo

Para diseñar soluciones con circuitos a problemas que involucran lógica combinatorial, debes seguir una serie de pasos que te ayudarán a obtener el circuito combinatorial con la solución del problema, evitando errores y reduciendo el número de compuertas lógicas en el circuito.

Primero debes analizar el problema encontrando cuáles son las variables *booleanas* de entrada, las necesarias para codificar los datos y cuáles serán las funciones de conmutación de salida, una vez más, considerando cuántas son necesarias para codificar la salida.

Realiza una tabla de verdad en la que a cada posible estado de las variables de entrada, asignes un estado a las funciones de conmutación. A partir de esta tabla, obtén la regla de correspondencia, ya sea reduciendo los minterminos o maxtérminos directamente con álgebra booleana o con ayuda de mapas de Karnaugh.

Finalmente realiza una simulación del circuito combinatorial con *Logisim* [**Burch**]. En el menú ayuda *ayuda* del programa podrás encontrar el manual de usuario con un tutorial para comenzar a usar el simulador, también se encuentra disponible en línea [**LogisimGuide**]. Se debe consultar las siguientes secciones, incluye el manejo básico de programa y las funciones necesarias para el desarrollo de la práctica:

- *Beginner's tutorial.*
- *Libraries and attributes.*
- *Subcircuits.*
- *Wire bundles.*

5 Entrada

Dependiendo del ejercicio, se deberá elegir la codificación apropiada al dato de entrada.

6 Salida

Dependiendo del ejercicio, se deberá elegir la codificación apropiada al dato de salida.

7 Variables libres

No hay variables libres para el desarrollo de esta práctica.

8 Procedimiento

Deberás entregar un solo archivo de *Logisim* con las soluciones de los ejercicios, un subcircuito por cada uno y un documento con las respuestas a las preguntas planteadas y el diseño de la solución de cada ejercicio: tabla de verdad, la regla de correspondencia y los pasos de álgebra booleana a los mapas de Karnaugh utilizados para reducirla.

Recuerda etiquetar las entradas y salidas de cada uno de los subcircuitos.

9 Ejercicios

1. Construye las 3 compuertas básicas: Not, Or y And. Para corroborar la funcionalidad de sus compuertas, deberán construir circuitos para simular las siguientes funciones. Solo puedes hacer uso de fuentes de alimentación power y ground, transistores tipo PNP y NPN y pines de entrada y salida.

COMPUERTA NOT	
Entrada	Salida
0	1
1	0



Figure 1: Compuerta NOT

COMPUERTA OR		
A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1



Figure 2: Compuerta OR

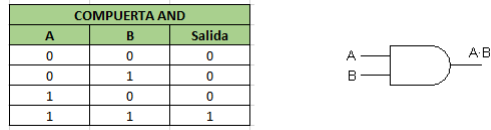


Figure 3: Compuerta AND

(a) $\neg A \vee B$

x_1	x_0	F
0	0	1
0	1	1
1	0	0
1	1	1

Table 1: Tabla de verdad del ejercicio 1a

$$F(x_1, x_0) = (\neg x_1 \neg x_0) + (x_1 \neg x_0) + (x_1 x_0)$$

$(x_1)(x_0)$	0	1
0	1	1
1	0	1

Figure 4: Mapa de Karnaugh de F del ejercicio 1a

$$F(x_1, x_0) = \neg x_1 + x_0$$

(b) $(A \wedge \neg B) \vee (\neg A \wedge B)$

x_1	x_0	F
0	0	0
0	1	1
1	0	1
1	1	0

Table 2: Tabla de verdad del ejercicio 1b

$$F(x_1, x_0) = (\neg x_1 x_0) + (x_1 \neg x_0)$$

No se puede minimizar más la función.

2. Construye un circuito que resuelva las situaciones que se pide. Debes hacer uso de las compuertas que construiste en el ejercicio anterior.

(a) Indicar si un número n es primo, con $n \in \{0, \dots, 15\}$

Número	x_3	x_2	x_1	x_0	F_1
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Table 3: Tabla de verdad ejercicio 2a

$$F_1(x_2, x_1, x_0) = (\neg x_3 \neg x_2 x_1 \neg x_0) + (\neg x_3 \neg x_2 x_1 x_0) + (\neg x_3 x_2 \neg x_1 x_0) + (\neg x_3 x_2 x_1 x_0) + (x_3 \neg x_2 x_1 x_0) + (x_3 x_2 \neg x_1 x_0)$$

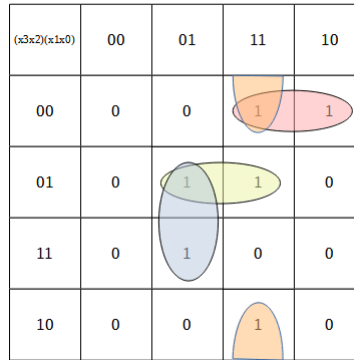


Figure 5: Mapa de Karnaugh de F_1 del ejercicio 2a

$$F_1(x_2, x_1, x_0) = (x_2 \neg x_1 x_0) + (\neg x_2 x_1 x_0) + (\neg x_3 x_2 x_0) + (\neg x_3 \neg x_2 x_1)$$

(b) Calcular el inverso de un número en complemento a 2 con 3 bits

x_2	x_1	x_0	F_1	F_2	F_3
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

Table 4: Tabla de verdad ejercicio 2b

$$F_1(x_2, x_1, x_0) = (\neg x_2 \neg x_1 x_0) + (\neg x_2 x_1 \neg x_0) + (\neg x_2 x_1 x_0) + (x_2 \neg x_1 \neg x_0)$$

$$F_2(x_2, x_1, x_0) = (\neg x_2 \neg x_1 x_0) + (\neg x_2 x_1 \neg x_0) + (x_2 \neg x_1 x_0) + (x_2 x_1 \neg x_0)$$

$$F_3(x_2, x_1, x_0) = (\neg x_2 \neg x_1 x_0) + (\neg x_2 x_1 x_0) + (x_2 \neg x_1 x_0) + (x_2 x_1 x_0)$$

X2(x1x0)	00	01	11	10
0	0	1	1	1
1	1	0	0	0

Figure 6: Mapa de Karnaugh de F1 del ejercicio 2b

$$F_1(x_2, x_1, x_0) = (x_2 \neg x_1 \neg x_0) + (\neg x_2 x_0) + (\neg x_2 x_1)$$

X2(x1x0)	00	01	11	10
0	0	1	0	1
1	0	1	0	1

Figure 7: Mapa de Karnaugh de F2 del ejercicio 2b

$$F_2(x_2, x_1, x_0) = (\neg x_1 x_0) + (x_1 \neg x_0)$$

X2(x1x0)	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Figure 8: Mapa de Karnaugh de F3 del ejercicio 2b

$$F_3(x_2, x_1, x_0) = x_0$$

- (c) En una planta de manejo de residuos tóxicos cuentan con la mejor tecnología para tratar desechos peligrosos y mantener a sus trabajadores seguros. Una parte fundamental de su sistema de protección consta de tres filtros, éstos tienen un sensor que indica si están en funcionamiento. Si uno de los filtros falla es posible trabajar con normalidad pero es necesario notificar al servicio técnico para que lo reparen a la brevedad. En caso de que fallen dos, la plataforma puede seguir trabajando pero los empleados se deben retirar más temprano. Finalmente, si es que fallan los 3, será necesario activar el protocolo de alerta y evacuar inmediatamente la planta. Se necesita un mecanismo que indique al personal de seguridad de la planta que acciones tomar según el estado de los filtros.

Primero asignemos una función para cada protocolo:

- **F1** será para el protocolo 1 el cual es que si uno de los filtros falla es posible trabajar con normalidad pero es necesario notificar al servicio técnico para que lo reparen a la brevedad.
- **F2** será para el protocolo 2 el cual es que si fallan dos filtros entonces la plataforma puede seguir trabajando pero los empleados se deben retirar más temprano.
- **F3** será para el protocolo 3 el cual es que si fallan los tres filtros entonces será necesario activar el protocolo de alerta y evacuar inmediatamente la planta.

Asignación de filtros:

- x_0 será para el filtro de seguridad 1.
- x_1 será para el filtro de seguridad 2.
- x_2 será para el filtro de seguridad 3.

Si falla algún filtro entonces x_n pasará a estado 1 lo cual indica que está lanzando una alerta.

x_2	x_1	x_0	F_1	F_2	F_3
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	1

Table 5: Tabla de verdad ejercicio 2c

$$F_1(x_2, x_1, x_0) = (\neg x_2 \neg x_1 x_0) + (\neg x_2 x_1 \neg x_0) + (x_2 \neg x_1 \neg x_0)$$

$$F_2(x_2, x_1, x_0) = (\neg x_2 x_1 x_0) + (x_2 \neg x_1 x_0) + (x_2 x_1 \neg x_0)$$

$$F_3(x_2, x_1, x_0) = x_2 x_1 x_0$$

Las funciones no se pueden minimizar más.

10 Preguntas

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

Para diseñar soluciones con circuitos a problemas que involucren lógica combinacional, debes seguir una serie de pasos que te ayudarán a obtener el circuito combinacional con la solución del problema, evitando errores y reduciendo el número de compuertas lógicas en el circuito.

Primero debes analizar el problema encontrando cuáles son las variables *booleanas* de entrada, las necesarias para codificar los datos y cuáles serán las funciones de conmutación de salida, una vez más, considerando cuántas son necesarias para codificar la salida.

Realiza una tabla de verdad en la que a cada posible estado de las variables de entrada, asignes un estado a las funciones de conmutación. A partir de esta tabla, obtén la regla de correspondencia, ya sea reduciendo los minterminos o maxtérminos directamente con álgebra booleana o con ayuda de mapas de Karnaugh.

Finalmente realiza una simulación del circuito combinacional con *Logisim*.

2. Si una función de conmutación se evalúa a más ceros que unos ¿es conveniente usar minterminos o maxtérminos?

Es mejor usar los miniterminos.

¿En el caso que se evalúe a más unos que ceros?

Es mejor usar maxiterminos.

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

Hacerlo a mano es propenso a cometer errores y en caso de que se desee usar álgebra booleana puede ser que no se llegue a la expresión más pequeña. Si se decide usar mapas de Karnaugh no se puede usar para más de seis variables y solo es práctico hasta 4 variables.