



ESCUELA DE INGENIERÍA EN COMPUTADORES

FUNDAMENTOS DE ARQUITECTURA DE COMPUTADORES

CE-1107

Bitácora: Lógica Combinatoria: Calculadora tomógrafo

Sergio Salazar Núñez, 2022155923

Daniel Duarte Cordero, 2022012866

Profesor: Luis Alberto Chavarría Zamora

16 de Agosto de 2023

1. Introducción

El objetivo de esta bitácora es proporcionar una visión clara y estructurada del proceso seguido, destacando los avances realizados, los problemas encontrados y las soluciones implementadas. Cada día de trabajo está debidamente registrado, incluyendo tanto los aspectos teóricos como las decisiones prácticas tomadas para asegurar el correcto funcionamiento del sistema. Además, se detallan las herramientas empleadas para el diseño automatizado, como software de simulación y simplificación de funciones lógicas, y se validan los procedimientos utilizados a lo largo del desarrollo. Esta bitácora también refleja la evolución del proyecto desde sus primeras etapas de planificación hasta la finalización de la implementación, resaltando los hitos clave alcanzados.

Este registro no solo facilita el seguimiento del progreso, sino que también constituye una herramienta importante para evaluar el proceso de aprendizaje y toma de decisiones a lo largo del proyecto, contribuyendo al desarrollo de competencias en ingeniería y diseño digital. Además, sirve como evidencia de la metodología empleada, permitiendo futuras consultas y mejoras en el diseño propuesto.

2. Desarrollo

2.1. Tablas Decodificadores 16/8/2024

Para el decodificador que traduce el número binario ingresado por los switches (simulando dedos), se ha diseñado una tabla de verdad que especifica la salida en función de las combinaciones de entradas binarias A, B, C y D. Estas entradas corresponden a los switches y permiten identificar qué dedos están "levantados". Las salidas Z_1 y Z_0 representan el número resultante en formato binario, el cual será utilizado como entrada para la suma.

A continuación, se presenta la tabla de verdad correspondiente al decodificador de dedos:

Cuadro 1: Tabla de verdad decodificador dedos

A	B	C	D	Z_1	Z_0
1	0	0	0	0	0
1	1	0	0	0	1
1	1	1	0	1	0
1	1	1	1	1	1

El comportamiento de este decodificador se limita a un patrón específico de dedos levantados, por lo que algunas combinaciones de entradas no son válidas y no se contemplan en el diseño.

Una vez obtenida la representación binaria de los dedos mediante el decodificador anterior, se procede a sumar este valor con el acumulado proporcionado por dos switches adicionales, representados por las entradas A, B, C y D del decodificador de suma. Este decodificador también presenta un comportamiento circular, es decir, al exceder el valor máximo permitido, el resultado rebotaz comienza desde el valor mínimo.

Cuadro 2: Tabla de verdad decodificador suma

A	B	C	D	F_0	F_1
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1

En esta tabla, se observa que los valores excedentes se reinician, asegurando la naturaleza circular del sistema. Esto permite que el diseño sea más eficiente para valores que no superen un umbral predeterminado.

2.2. Mapas de Karnaugh 19/8/2024

Con las tablas obtenidas previamente es posible aplicar los mapas de Karnaugh.

Los mapas de Karnaugh permiten simplificar las funciones booleanas que definen las salidas del decodificador. A continuación, se presentan los mapas de Karnaugh correspondientes para las funciones Z_1 , Z_0 , F_0 y F_1 .

Cuadro 3: Karnaugh para el Z_1 del deco de Dedos

$\frac{AB}{CD}$	00	01	11	10
00	x	x	0	0
01	x	x	x	x
11	x	x	1	x
10	x	x	1	x

Cuadro 4: Karnaugh para el Z_1 del deco de Dedos

$\frac{AB}{CD}$	00	01	11	10
00	x	x	1	0
01	x	x	x	x
11	x	x	1	x
10	x	x	0	x

Para estos mapas de Karnaugh se marca con x los casos que no se contemplan, ya que el la cantidad de dedos siempre tiene el mismo patrón. Por ejemplo, para que se lea un dedo será necesario hacerlo con el índice extendido, siendo este tomado como 1000. En caso de que se quiera extender el dedo corazón para ser leído como un 1 esto no será posible, ya que el número generado sería 0100, siendo esta una combinación que no se contempla.

Cuadro 5: Karnaugh para el f_0 del deco para la suma

$\frac{AB}{CD}$	00	01	11	10
00	0	1	0	1
01	1	1	0	0
11	0	0	1	1
10	1	0	1	0

Cuadro 6: Karnaugh para el f_1 del deco para la suma

$\frac{AB}{CD}$	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

2.3. Simplificación Álgebra Booleana y Simulaciones: 28/8/2024

A partir de los mapas de Karnaugh de la sección anterior se procede a sacar los mintérminos para luego simplificar con álgebra booleana.

$$Z_1 = C \quad (1)$$

$$Z_0 = B\overline{C} + D \quad (2)$$

$$f_0 = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{C}D + ACD + ABC + \overline{A}\overline{B}C\overline{D} + \overline{A}BC\overline{D}$$

$$f_0 = \overline{A}\overline{C}(B + D) + AC(B + D)\overline{B}\overline{D}(\overline{A}\overline{C} + \overline{A}C) \text{Asociatividad}$$

$$f_0 = (B + D)(\overline{A}\overline{C} + AC) + \overline{B}\overline{D}(A \oplus C) \text{Disyuncion - exclusiva, Asociatividad} \quad (3)$$

$$f_1 = BD + \overline{B}\overline{D} \quad (4)$$

Estas simplificaciones permiten un diseño más eficiente, reduciendo el número de compuertas lógicas necesarias para implementar los decodificadores.

Seguidamente se procedió a realizar las respectivas simulaciones, precisamente para el segundo decodificador. No se realizaron simulaciones para el resto del circuito lógico, ya que consideramos que únicamente este decodificador podía representar cierto grado de complejidad. Para ello se utilizó la herramienta de Software llamada Digital Logic Sim la cual corresponde a un software utilizado para diseñar y simular circuitos de lógica digital.

La primera versión de la simulación fue la siguiente:

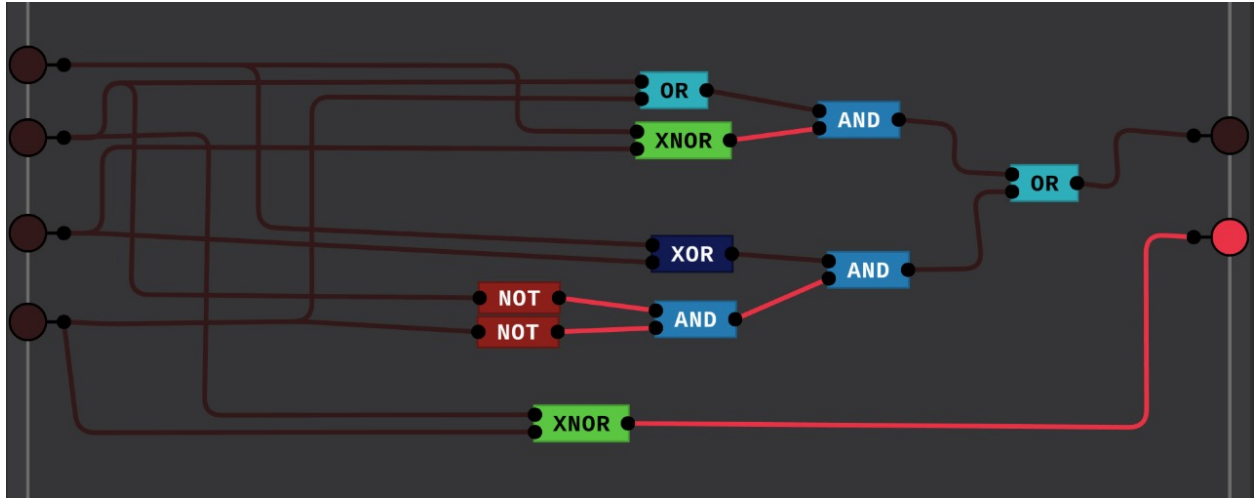


Figura 2.1: Simulación 1 Decodificador Suma

Si bien con esta simulación se reflejaba el comportamiento esperado para la suma circular, se buscó simplificar aún más el circuito con el fin de optimizar el rendimiento y reducir la cantidad de compuertas lógicas requeridas. Teniendo de esta manera la siguiente versión, siendo esta la final

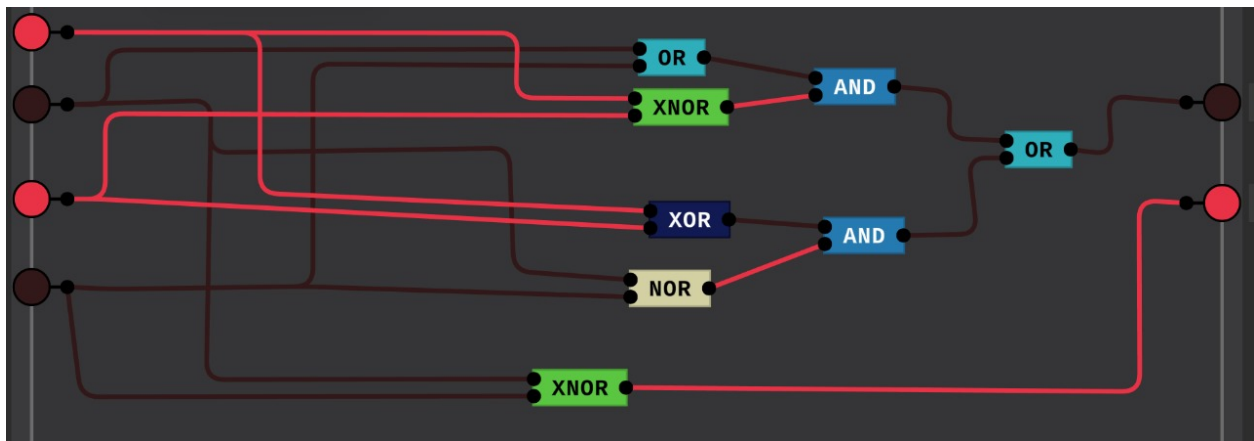


Figura 2.2: Simulación 2 Decodificador Suma

Si bien no se presenta una diferencia significativa entre 2.1 y 2.2. Se logra reducir dos de las compuertas. Teniendo que dos compuertas NOT, ambas conectadas a un AND pueden ser reducidas a un compuerta NOR. Así mismo, se presenta una diferencia en estas imágenes respecto a las ecuaciones 4 y 3. Partiendo de suponer que se tienen dos entradas A y b, al tener $\overline{A}B + AB$ se simplifica a un XNOR, o sea $A \oplus B$