PROYECTO ARQUITECTURA DE COMPUTADORAS

ETAPA 1

25/04/2024

Integrante 1: [Giacomodonato Giulia - 142049]
Integrante 2: [Kreczmer Tomás - 140009]
Logisim-Evolution: [3.8.0]

Índice

Introducción 2
Etapa 1 3
Descripción del circuito 3
Funciones 3
Estados 6
Comparadores 8
Cálculo del estado siguiente 9

Introducción

Este proyecto de Arquitectura de Computadoras se centra en la implementación de un circuito controlador de estados basado en el diagrama proporcionado. La implementación se realizó utilizando la herramienta Logisim-Evolution.

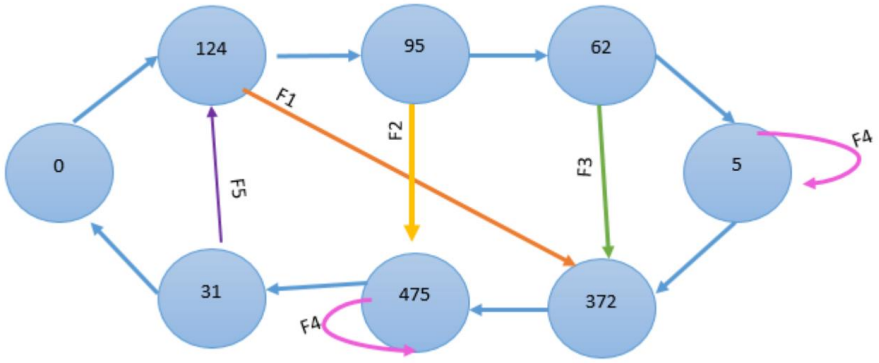


Figura 1: Grafo de estados

La idea general que utilizamos para resolver el problema fue utilizar un contador de 9 bits, separarlos en la entrada y en la salida, determinar en qué estado está y a cuál estado debe saltar para luego conformar el número siguiente bit por bit según corresponda.

Durante todo el proyecto, intentamos mantener la organización y claridad en el diseño para garantizar el correcto funcionamiento del circuito y facilitar su comprensión.

Etapa 1

Descripción del circuito

Funciones

Minimizamos las funciones mediante el método de Karnaugh y agrupamos los mintérminos adyacentes que forman implicantes primos esenciales y no esenciales. Dos términos productos están en casillero adyacentes si difieren en exactamente un literal, que aparece complementado en uno y sin complementar en el otro (código Gray).

Tanto como para F1, F2 y F3 utilizamos como líneas de selección a las entradas A y B para los multiplexores.

- F1(A, B, C, D, E) = Σ (3, 11, 12, 15, 16, 20, 21, 23, 24, 28), Σ_{op} (0)

AB\CDE	000	001	011	010	110	111	101	100
00	*		1					
01			1			1		1
11	1							1
10	1					1	1	1

Las funciones residuo que se generan son las siguientes:
00: C'DE
01: C'DE + CDE + CD'E'
11: D'E'
10: CE + D'E'

- F2(A, B, C, D, E) = Σ (0, 4, 10, 11, 15, 16, 19, 20, 23, 27, 31), Σ_{op} (2, 5, 13)

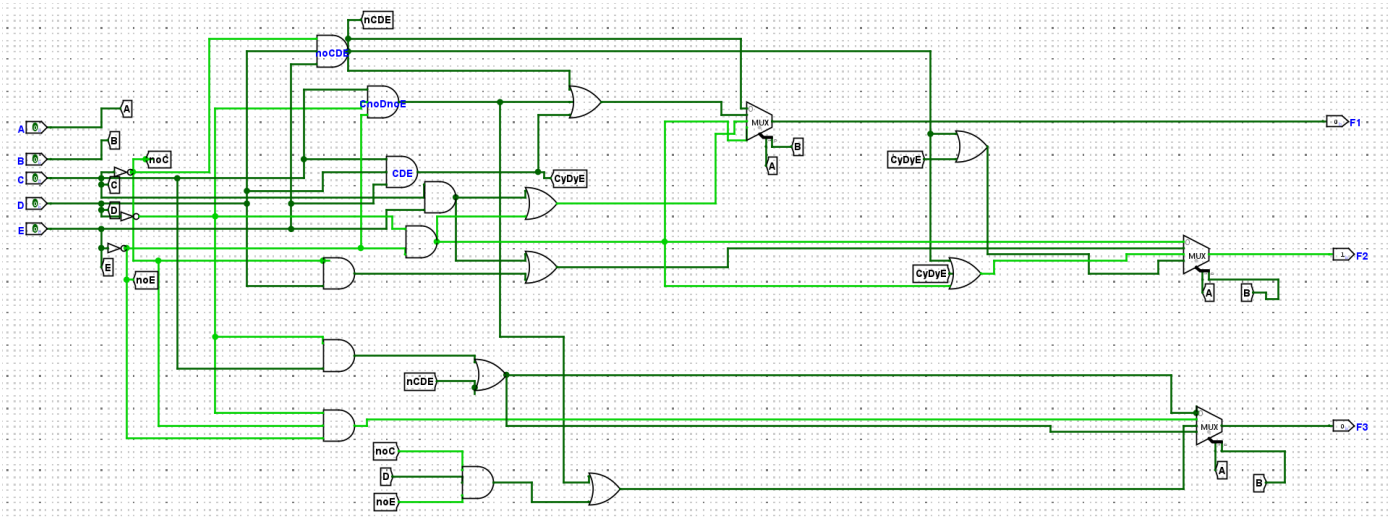
AB\CDE	000	001	011	010	110	111	101	100
00	1			*			*	1
01			1	1		1	*	
11			1			1		
10	1		1			1		1

Las funciones residuo que se generan son las siguientes:
00: D'E'
01: C'D + CE
11: C'DE + CDE
10: D'E' + C'DE + CDE

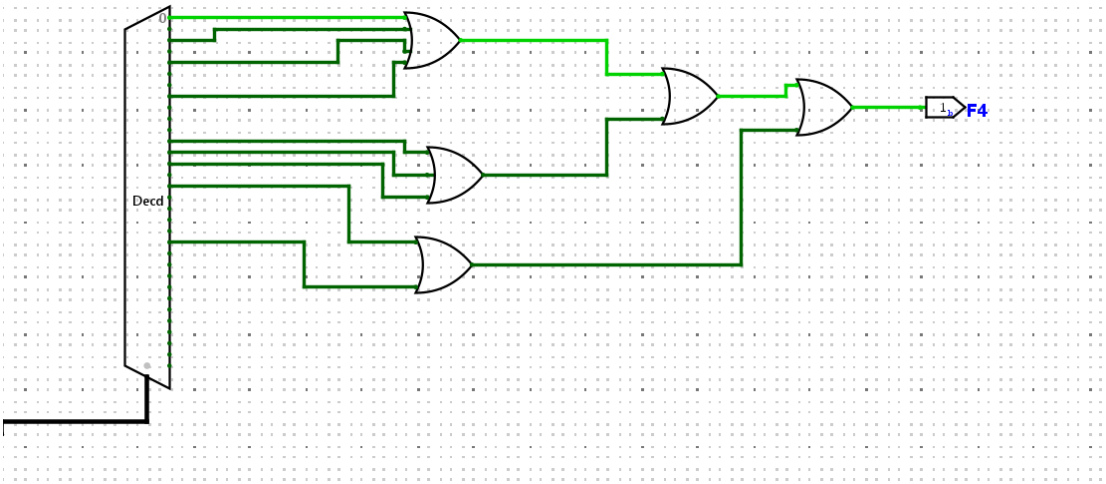
- F3(A, B, C, D, E) = Σ (3, 4, 5, 8, 18, 20, 27, 28, 29), Σ_{op} (31)

AB\CDE	000	001	011	010	110	111	101	100
00			1				1	1
01	1							
11			1			*	1	1
10				1				1

Las funciones residuo que se generan son las siguientes:
00: CD' + C'DE
01: C'D'E'
11: CD' + C'DE
10: C'DE' + CD'E'



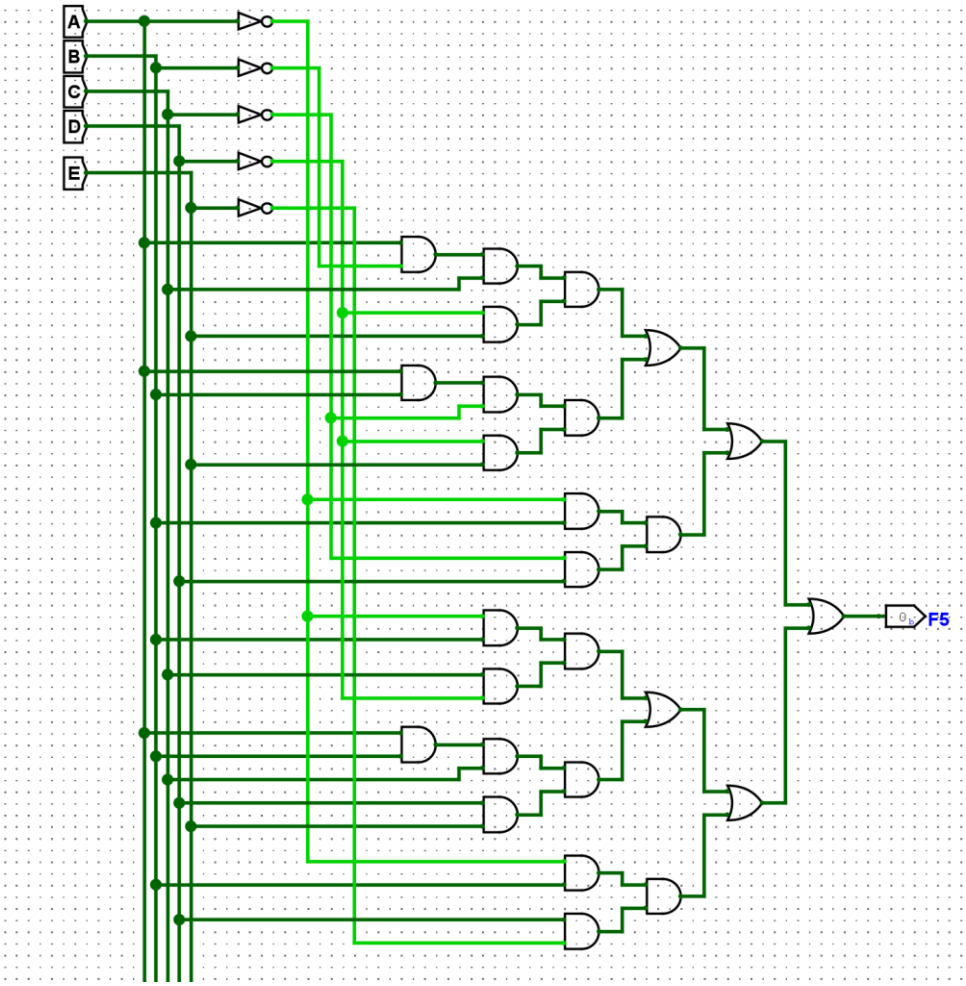
- F4(A, B, C, D, E) = Σ (0,2, 4, 7, 11, 12, 13, 15, 20), Σ_{op} (14)
Para computar F4 ingresamos A, B, C, D, y E al decodificador, elegimos las salidas que representan los números en los que F4 debe devolver 1 según su descripción, y las unimos con compuertas OR.



- F5(A, B, C, D, E) = Σ (10, 11, 12, 13, 14, 21, 23, 25, 31)
  Para F5 no tenemos líneas de selección ya que debemos utilizar compuertas.

AB\CDE	000	001	011	010	110	111	101	100
00								
01			1	1	1		1	1
11		1				1		
10							1	

Luego, obtenemos F5 = AB'CD'E + ABC'D'E + A'BC'D + A'BCD' + ABCDE + A'BDE' y escribimos la función mediante compuertas.



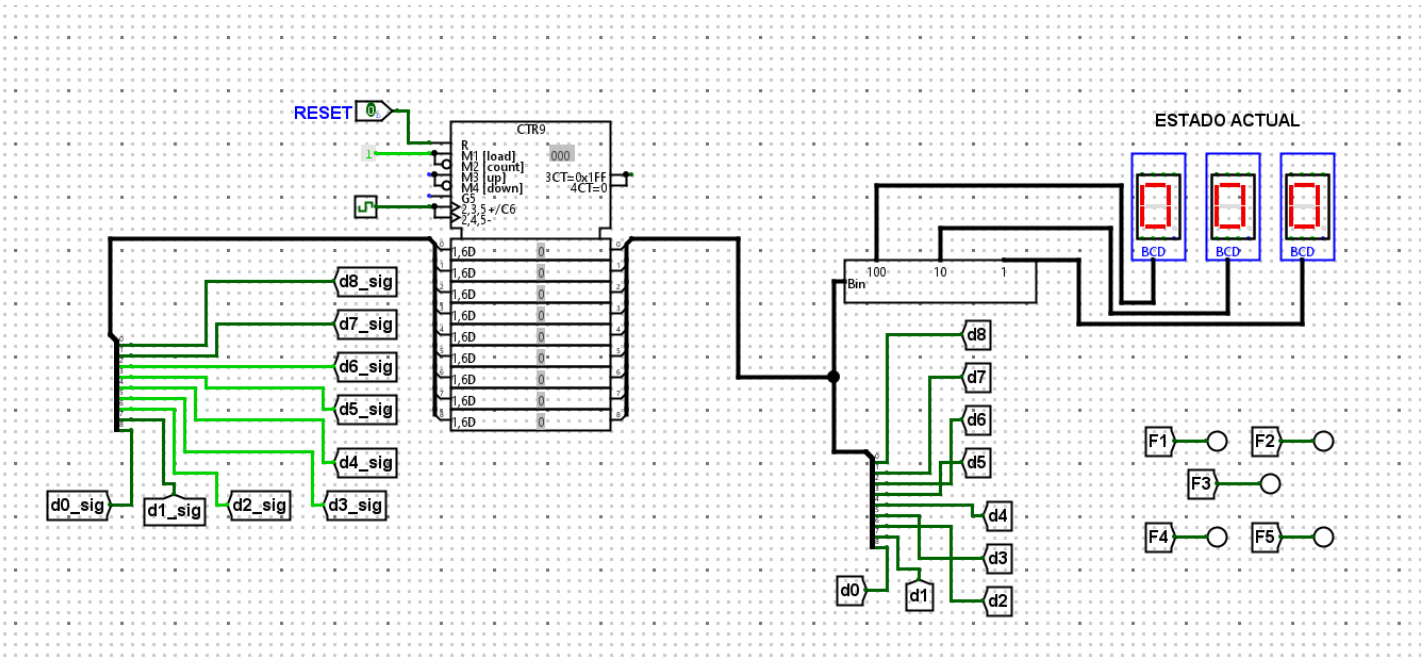
Estados

Los cambios de estados del circuito quedan definidos mediante la siguiente tabla realizada a partir del diagrama de estados proporcionado, que contiene el estado actual, los valores de las diferentes funciones y el estado siguiente. El contador trabajará con números de 9 bits.

Estado actual d0d1d2d3d4d5d6d7d8	**F1**	**F2**	**F3**	**F4**	**F5**	**Estado siguiente (d0d1dd2d3d4d5d6d7d8)+**
000000000 (0)	*	*	*	*	*	001111100 (124)
001111100 (124)	0	*	*	*	*	001011111 (95)
001111100 (124)	1	*	*	*	*	101110100 (372)
001011111 (95)	*	0	*	*	*	000111110 (62)
001011111 (95)	*	1	*	*	*	111011011 (475)
000111110 (62)	*	*	0	*	*	000000101 (5)
000111110 (62)	*	*	1	*	*	101110100 (372)
000000101 (5)	*	*	*	0	*	101110100 (372)
000000101 (5)	*	*	*	1	*	000000101 (5)
101110100 (372)	*	*	*	*	*	111011011 (475)
111011011 (475)	*	*	*	0	*	000011111 (31)
111011011 (475)	*	*	*	1	*	111011011 (475)
000011111 (31)	*	*	*	*	0	000000000 (0)
000011111 (31)	*	*	*	*	1	001111100 (124)

El contador recibe como entradas los 9 bits del número que representa el estado, el reloj, y siempre está en modo load para poder realizar los saltos necesarios. Las salidas son los 9 bits del estado actual, que se muestran en el display de 7 segmentos primero habiendo convertido el número de binario a decimal utilizando el conversor que provee Logisim.

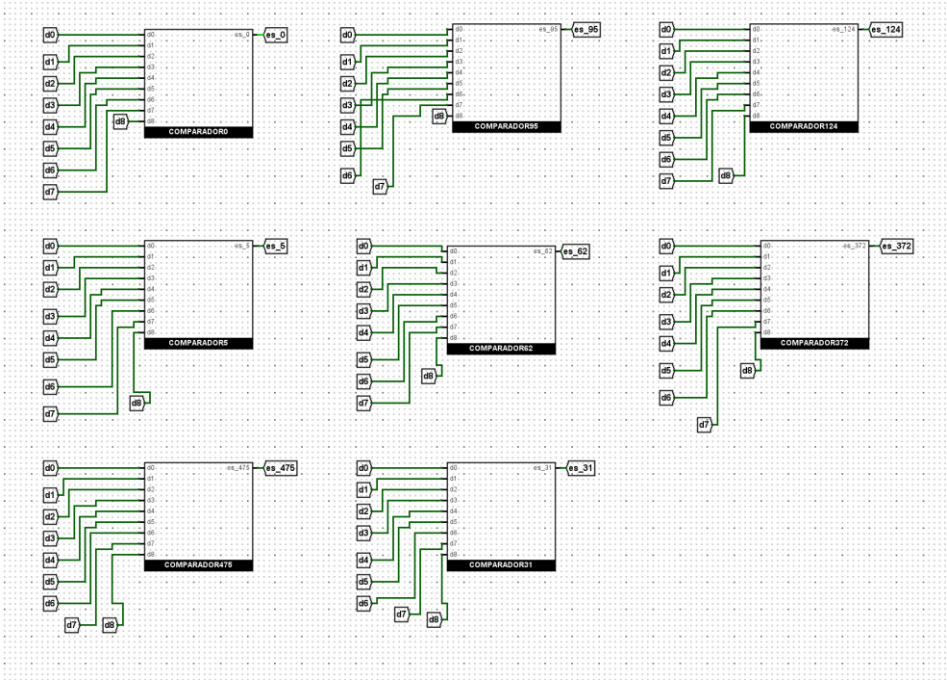
En todos los circuitos, para mantener la claridad y evitar errores, utilizamos etiquetas y túneles en todas las entradas y salidas.



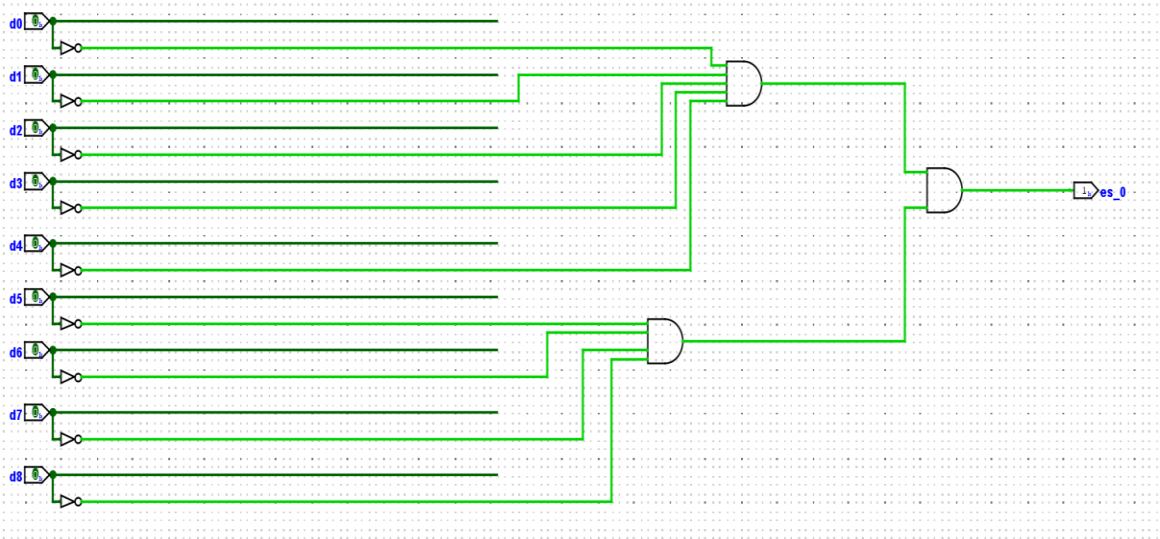
Comparadores

Para saber cuál es el estado actual y a qué estado debemos saltar, implementamos 8 comparadores que representan los diferentes 8 estados del circuito. Cada uno retorna 0 o 1 dependiendo de la salida del contador.

Por ejemplo, si el estado actual es el 0 el circuito COMPARADOR0 devuelve 1 y todos los demás devuelven 0.

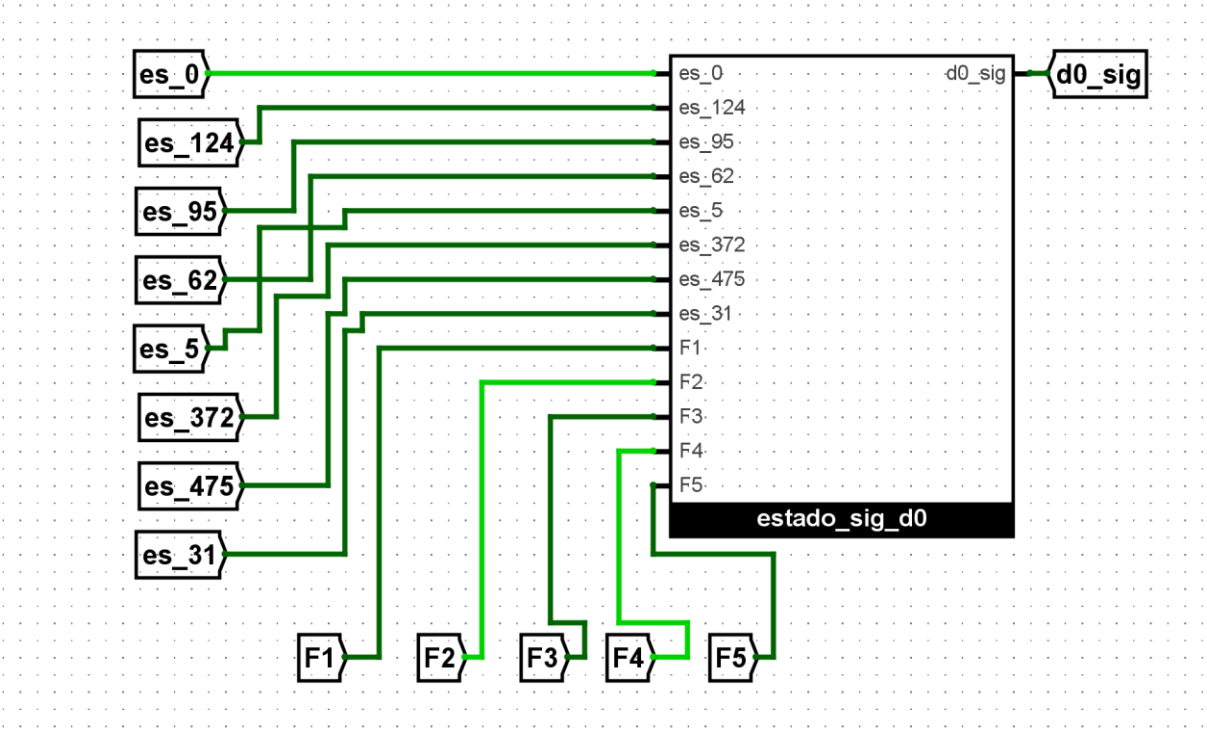


Dentro de cada uno, analizamos bit por bit si es el número que queremos y realizamos el cómputo mediante compuertas.



Cálculo del estado siguiente

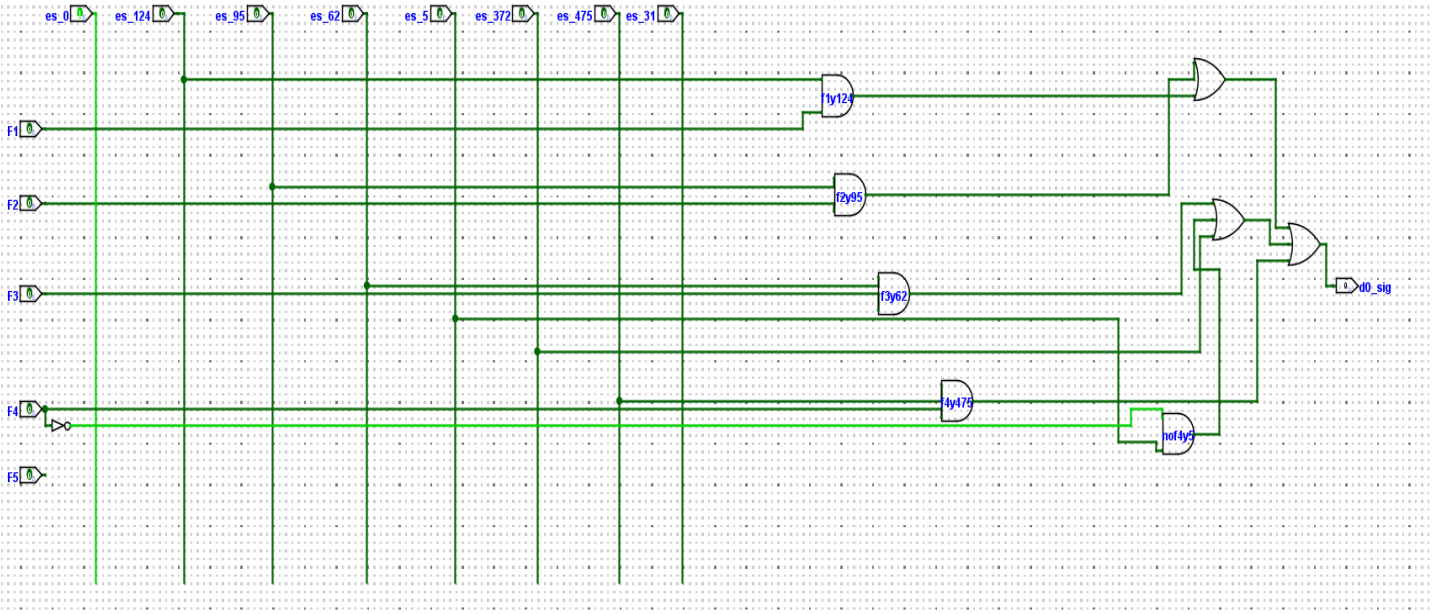
Una vez calculado el estado actual y para computar el estado siguiente, implementamos 9 circuitos donde cada uno calcula 1 bit para conformar el número al que se debe saltar.



Cada uno recibe como entrada las salidas de los 8 comparadores, los estados de las 5 funciones y devuelve 1 bit. Para realizarlo, utilizamos la tabla anteriormente mencionada, y para cada bit creamos una tabla reducida que define en qué momentos ese bit es 1. Por ejemplo, para el bit d0:

Estado actual d0d1d2d3d4d5d6d7d8	F1	F2	F3	F4	F5	(do)+
001111100 (124)	1	*	*	*	*	1
001011111 (95)	*	1	*	*	*	1
000111110 (62)	*	*	1	*	*	1
00000101 (5)	*	*	*	0	*	1
101110100 (372)	*	*	*	*	*	1
111011011 (475)	*	*	*	1	*	1

Luego, calculamos el resultado mediante compuertas.



Cada una de estas salidas son ingresadas al contador para que cuando ocurra el siguiente pulso, pueda saltar al estado esperado.