

LABORATORIO 1 - DISEÑO DE CIRCUITOS ELECTRONICOS

ANDRÉS FELIPE GIRALDO YUSTI¹ y JUAN CAMILO VELEZ²

¹Universidad de Antioquia, Facultad de Ingeniería, andres.giraldo@udea.edu.co

²Universidad de Antioquia, Facultad de Ingeniería, camilo.velez4@udea.edu.co

JOHN BYRON BUITRAGO PANIAGUA

1. Introducción

En esta práctica se abordan los fundamentos del diseño de sistemas digitales mediante la creación y simulación de circuitos en el entorno Logisim. A lo largo de la actividad, se aplican conceptos clave como los estados finitos, mapas de Karnaugh, señales de reloj y sincronización, lo que permite analizar y comprender el comportamiento lógico y temporal de los circuitos digitales.

Para ello, se emplean diversos componentes electrónicos fundamentales, como comparadores, flip-flops, memorias RAM y generadores de reloj, integrándolos en arquitecturas funcionales que responden a objetivos específicos.

Esta experiencia práctica brinda una base sólida para el análisis, diseño e implementación de sistemas digitales, reforzando la comprensión teórica a través de la experimentación en un entorno virtual.

2. Panteamiento del problema y objetivos

2.1. Problema

En el contexto de la computación digital, los algoritmos de reducción permiten optimizar operaciones sobre grandes volúmenes de datos. La implementación de estos algoritmos en hardware digital requiere un diseño jerárquico y modular que aproveche eficientemente los recursos computacionales disponibles. El desafío consiste en desarrollar un sistema digital capaz de realizar operaciones de reducción sobre un arreglo de datos, bajo restricciones de diseño lógico y utilizando únicamente compuertas básicas y componentes secuenciales definidos.

2.2. Objetivos

Diseñar e implementar un sistema digital en Logisim Evolution que realice la operación de reducción sobre un arreglo de 32 datos de 8 bits, siguiendo un enfoque jerárquico, modular y cumpliendo con las restricciones lógicas impuestas.

Restricciones:

Diseñar el bloque básico de operación a partir de compuertas lógicas AND, OR y NOT.

Implementar una máquina de estados finitos que controle las fases del proceso de reducción.

Visualizar el estado del sistema mediante displays de siete segmentos y señalización de fases.

Verificar el funcionamiento del sistema mediante simulación exhaustiva en Logisim Evolution.

3. Fundamentos teóricos

En el ámbito del diseño de sistemas digitales, los algoritmos de reducción son herramientas esenciales para optimizar operaciones que involucran grandes volúmenes de datos, como sumatorias, búsquedas de mínimos o máximos, entre otras. Estas técnicas permiten reducir la complejidad computacional de $O(n)$ a $O(\log(n))$ mediante una estructura jerárquica de fases en las que los datos son procesados en paralelo y reutilizando componentes computacionales.

Para implementar estos sistemas en hardware digital, se emplean conceptos clave como:

Máquinas de Estados Finitos (MEF): estructuras de control secuencial que permiten controlar el flujo de ejecución del sistema por fases.

Flip-Flops: componentes básicos de almacenamiento utilizados para construir registros y controlar estados. Su tipo (D, T, JK, etc.) depende del diseño asignado.

Memorias RAM: permiten el almacenamiento temporal de los datos de entrada y los resultados intermedios durante la reducción.

Sumadores jerárquicos: diseñados a partir de operadores de 1 bit utilizando únicamente compuertas lógicas básicas (AND, OR, NOT), se ensamblan en estructuras de mayor tamaño para realizar operaciones de reducción.

Mapas de Karnaugh: utilizados para simplificar funciones lógicas.

4. Metodología

El desarrollo del sistema de reducción se realizó siguiendo un enfoque estructurado, apoyado en la herramienta de simulación digital Logisim Evolution. La metodología aplicada se puede resumir en las siguientes etapas:

Diseño del operador básico: se implementó un comparador de 2 bits empleando únicamente compuertas lógicas básicas (AND, OR, NOT). Este bloque se validó individualmente antes de su integración.

Construcción jerárquica de comparador de 8 bits: a partir del sumador de 2 bit, se ensamblaron comparadores de mayor tamaño siguiendo una arquitectura jerárquica y modular.

Diseño de la memoria y control de fases: se implementó una memoria RAM capaz de almacenar los 32 datos de 8 bits, y se diseñó una Máquina de Estados Finita (MEF) para controlar el avance por cada fase de reducción. Esta MEF fue construida utilizando el tipo de flip-flop tipo TK.

Visualización del sistema: se desarrolló un módulo combinacional para decodificar los resultados y la fase actual en displays de siete segmentos.

Integración y simulación: todos los bloques fueron conectados e integrados en un diseño superior. Se realizaron simulaciones exhaustivas para verificar el correcto funcionamiento del sistema, tanto en condiciones normales como ante eventos de error como overflow, los cuales provocan la detención del sistema y su notificación al usuario.

5. Diseño del sistema

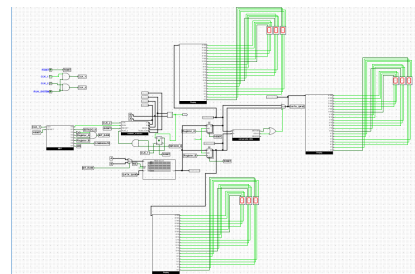


Figura 1. Circuito completo

La Figura 1 muestra el diseño completo del sistema digital implementado para llevar a cabo la operación de reducción sobre un arreglo de 32 datos de 8 bits. El sistema ha sido construido de forma jerárquica.

Este diseño se compone de varios bloques funcionales fundamentales: una memoria RAM para el almacenamiento de los datos y resultados intermedios, una Máquina de Estados Finita (MEF) encargada

de controlar el avance por cada fase del proceso de reducción, una unidad de comparadores jerárquicos diseñados a partir de operadores de 1 bit construidos únicamente con compuertas AND, OR y NOT, y un módulo de visualización que permite observar tanto el contenido actual de la RAM como la fase en ejecución a través de displays de siete segmentos.

El control del sistema está regido por señales de reloj, RESET y RUN. Durante cada fase, los datos son leídos desde posiciones específicas de la memoria, procesados en paralelo por el comparador, y los resultados se almacenan nuevamente en las posiciones correspondientes. Este proceso se repite indefinidamente.

5.1. Memoria RAM

Se usó la memoria RAM que nos brinda el programa de diseño Logisim, esta fue usada de manera asíncrona, para que no necesitara de la señal del reloj para leer y escribir.

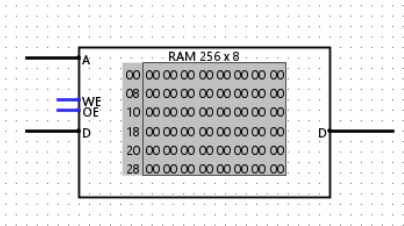


Figura 2. Memoria RAM

- Esta cuenta con cuatro entradas y una salida que son:
- A: Dato de 5 bits que define la posición de almacenamiento
 - D: Dato de 8 bits que se guardará en la ram (Esta es entrada es para escribir)
 - WE: Esta habilita la escritura en la RAM
 - OE: Esta habilita la lectura en la RAM

5.2. Maquina de estado finito (MEF)

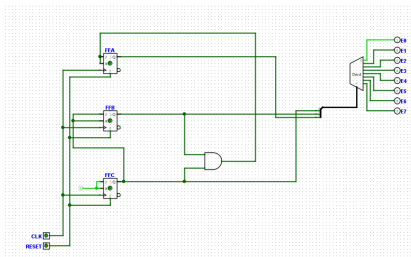


Figura 3. MEF

La Máquina de Estados Finita diseñada para este sistema recorre una secuencia de estados numerados del 0 al 7, donde cada estado representa una fase específica del proceso de reducción. Esta MEF se encarga de sincronizar la operación del sistema, controlando cuándo deben leerse los datos, cuándo se debe de hacer la comparación y en qué momento deben almacenarse los resultados en la memoria.

El valor actual del estado es enviado a un decodificador, el cual traduce el número del estado activo en una salida codificada que permite activar únicamente la fase correspondiente.

Q(t)	Q(t+1)	J_aK_a	J_bK_b	J_cK_c
000	001	0x	0x	1x
001	010	0x	1x	x1
010	011	0x	x0	1x
011	100	1x	x1	x1
100	101	x0	0x	1x
101	110	x0	1x	x1
110	111	x0	x0	1x
111	000	x1	x1	x1

Figura 4. Tabla de estados de MEF

KARNOFF

J _A				
abc	00	01	11	10
0	0	0	1	0
1	x	x	x	x

$$J_A = BC$$

K _A				
abc	00	01	11	10
0	x	x	x	x
1	0	0	1	0

$$K_A = BC$$

J _B				
abc	00	01	11	10
0	0	1	x	x
1	0	1	x	x

$$J_B = C$$

K _B				
abc	00	01	11	10
0	x	x	1	0
1	x	x	1	0

$$K_B = C$$

J _C				
abc	00	01	11	10
0	1	x	x	1
1	1	x	x	1

$$J_C = 1$$

K _C				
abc	00	01	11	10
0	x	1	1	x
1	x	1	1	x

$$K_C = 1$$

Figura 5. Tablas de Karnoff

5.3. Comparador de 8 bits

Para la implementación del comparador de 8 bits, se siguió un enfoque jerárquico. Inicialmente, se diseñó un comparador básico de 2 bits, el cual permitió establecer la lógica fundamental de comparación teniendo en cuenta el bit de mayor significancia. A partir de este bloque, se construyó un comparador de 4 bits combinando dos comparadores de 2 bits de forma estructural. Finalmente, se ensambló el comparador de 8 bits utilizando el mismo principio modular.

Adicionalmente, se implementó un circuito externo de control jerárquico que permite gestionar la comparación de los bits más significativos en caso de igualdad parcial, asegurando que el resultado final refleje correctamente la relación entre los dos valores evaluados. Esta estructura modular facilita la escalabilidad del diseño y garantiza un comportamiento predecible en cada nivel de comparación.

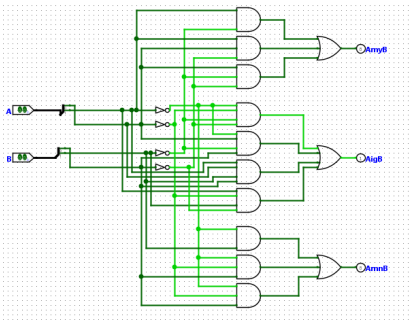


Figura 6. Comparador de 2 Bits

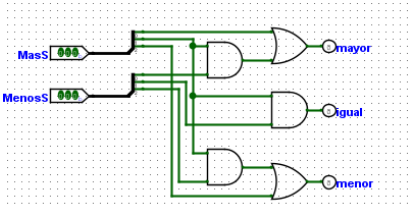


Figura 7. Circuito de control Jerárquico

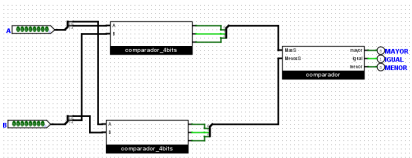


Figura 8. Conexión de los comparadores para comparación jerárquica

5.4. Contador de 8 bits

Se diseñó un contador de 8 bits secuencial utilizando una tabla de estados y mapas de Karnaugh para obtener las expresiones mínimas de cada uno de los bits de salida. El contador incrementa su valor con cada pulso de reloj y fue implementado con flip-flops tipo JK, cuyos circuitos se derivaron a partir del análisis lógico de los cambios de estado. Este diseño garantiza una transición estable y eficiente entre los estados binarios, y forma parte del control de fases del sistema de reducción, exactamente para recorrer las posiciones de la RAM.

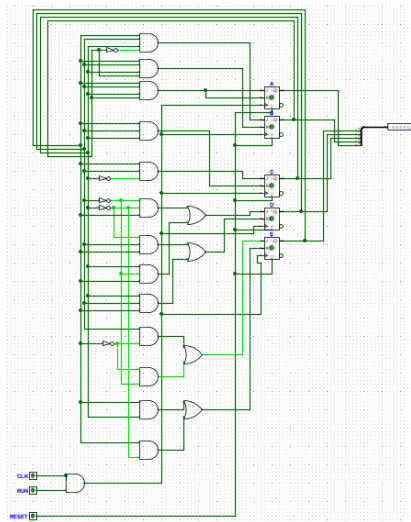


Figura 9. Contador de 5 bits

Q(t)	Q(t+1)	JK_a	JK_b	JK_c	JK_d	JK_e
ABCDE	ABCDE					
00000	00001	0x	0x	0x	0x	1x
00001	00010	0x	0x	0x	1x	x1
00010	00011	0x	0x	0x	x0	1x
00011	00100	0x	0x	1x	x1	x1
00100	00101	0x	0x	x0	0x	1x
00101	00110	0x	0x	x0	1x	x1
00110	00111	0x	0x	x0	x0	1x
00111	01000	0x	1x	x1	x1	x1
01000	01001	0x	x0	0x	0x	1x
01001	01010	0x	x0	0x	1x	x1
01010	01011	0x	x0	0x	x0	1x
01011	01100	0x	x0	1x	x1	x1
01100	01101	0x	x0	x0	0x	1x
01101	01110	0x	x0	x0	1x	x1
01110	01111	0x	x0	x0	x0	1x
01111	10000	1x	x1	x1	x1	x1
10000	10001	x0	0x	0x	0x	1x
10001	10010	x0	0x	0x	1x	x1
10010	10011	x0	0x	0x	x0	1x
10011	10100	x0	0x	1x	x1	x1
10100	10101	x0	0x	x0	0x	1x
10101	10110	x0	0x	x0	1x	x1
10110	10111	x0	0x	x0	x0	1x
10111	11000	x0	1x	x1	x1	x1
11000	11001	x0	x0	0x	0x	1x
11001	11010	x0	x0	0x	1x	x1
11010	11011	x0	x0	0x	x0	1x
11011	11100	x0	x0	1x	x1	x1
11100	11101	x0	x0	x0	0x	1x
11101	11110	x0	x0	x0	1x	x1
11110	11111	x0	x0	x0	x0	1x
11111	00000	x1	x1	x1	x1	x1

Figura 10. Tabla de estados

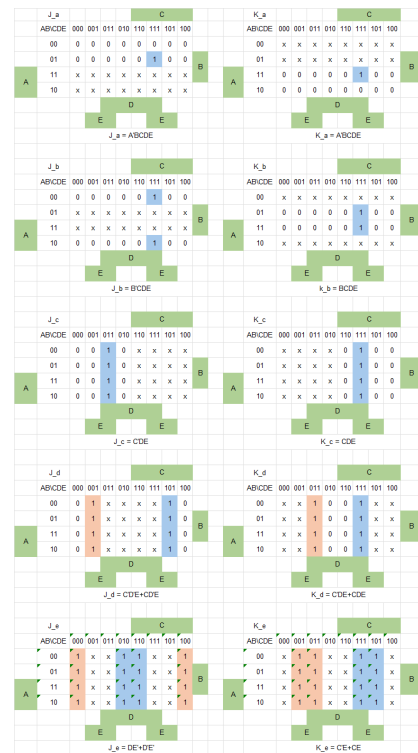


Figura 11. Mapas de Karnaugh

5.5. Contador con máscara y control de fases

El contador con máscara es un circuito compuesto por tres módulos principales: un contador de 5 bits, un contador de fase y una unidad de suma de fase. Este bloque cumple la función de controlar el recorrido por las diferentes fases del algoritmo de reducción, así como definir las posiciones del arreglo que deben ser comparadas en cada etapa.

El contador de fase es una máquina de estados que sigue la secuencia: 01111, 00111, 00011, 00001, 00000, volviendo luego a 01111. Cada estado tiene bits en alto que actúan como una máscara, permitiendo activar solo ciertas posiciones. Por ejemplo, en la primera fase (01111), se habilitan las primeras 16 posiciones del arreglo, y se ignora la última, ya que el bit menos significativo está en cero. Esta máscara se utiliza junto con compuertas AND para seleccionar los datos relevantes en cada fase.

Por otro lado, el módulo de suma de fase genera la secuencia: 10000, 01000, 00100, 00010, 00001, repitiéndose cíclicamente. Esta salida indica el desplazamiento que debe sumarse a cada posición base para acceder al segundo operando en las comparaciones. Por ejemplo, en la fase 1 se suma 16 (10000), en la fase 2 se suma 8 (01000), luego 4, 2 y finalmente 1. Esto permite realizar comparaciones como: 0 y 16, 1 y 17, ..., luego 0 y 8, 1 y 9, y así sucesivamente, reduciendo progresivamente el arreglo hasta llegar al resultado final.

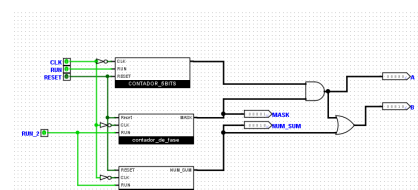


Figura 12. Contador con máscara

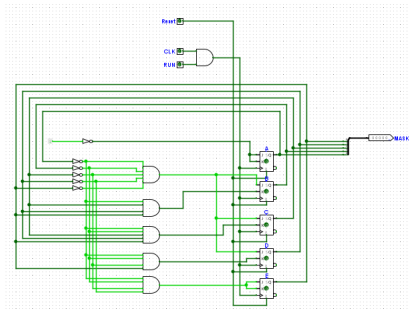


Figura 13. Contador de fase

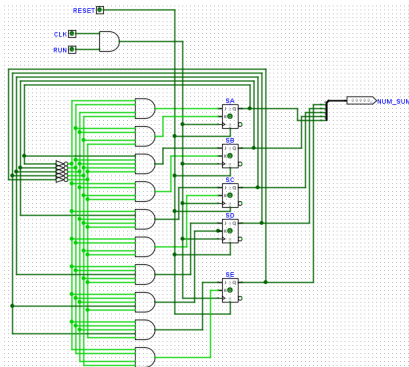


Figura 14. Contador de suma de fase

Q(t)	Q(t+1)	JK _a	JK _b	JK _c	JK _d	JK _e
ABCDE	ABCDE					
01111	00111	0x	x1	x0	x0	x0
00111	00011	0x	0x	x1	x0	x0
00011	00001	0x	0x	0x	x1	x0
00001	00000	0x	0x	0x	0x	x1
00000	01111	0x	1x	1x	1x	1x

Figura 15. Tabla de estado, contador de fase

Q(t)	Q(t+1)	JK _a	JK _b	JK _c	JK _d	JK _e
ABCDE	ABCDE					
00000	10000	1x	0x	0x	0x	0x
10000	01000	x1	1x	0x	0x	0x
01000	00100	0x	x1	1x	0x	0x
00100	00010	0x	0x	x1	1x	0x
00010	00001	0x	0x	0x	x1	1x
00001	10000	1x	0x	0x	0x	x1

Figura 16. Tabla de estado, contador de suma de fase

J _a	C	K _a	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	0 0 0 0 0 0 0 0	00	0 0 0 0 0 0 0 0
01	0 0 0 0 0 0 0 0	01	0 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	0 0 0 0 0 0 0 0
J _a =0		K _a =0	
J _b	C	K _b	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	1 0 0 0 0 0 0 0	00	x x x 0 0 x 0 0
01	0 0 0 0 0 0 x 0	01	0 0 0 0 0 0 1 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	0 0 0 0 0 0 0 0
J _b =A'B'C'D'E'		K _b =A'C'D'E	
J _c	C	K _c	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	1 0 0 0 0 0 x 0	00	x x x 0 0 1 0 0
01	0 0 0 0 0 0 0 0	01	0 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	0 0 0 0 0 0 0 0
J _c =A'B'C'D'E'		K _c =A'B'C'D'E	
J _d	C	K _d	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	1 0 x 0 0 x 0 0 0	00	x 1 0 0 0 0 0 0 0
01	0 0 0 0 0 0 x 0	01	0 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	0 0 0 0 0 0 0 0
J _d =A'B'C'D'E'		K _d =A'B'C'E	
J _e	C	K _e	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	1 x 0 0 0 0 0 0 0	00	x 1 0 0 0 0 0 0 0
01	0 0 0 0 0 0 0 0	01	0 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	0 0 0 0 0 0 0 0
J _e =A'B'C'D'		K _e =A'B'C'D'	

Figura 17. Tabla de Karnoff, contador de fase

J _a	C	K _a	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	0 1 0 0 0 0 0 0	00	x x 0 0 0 0 0 x
01	0 0 0 0 0 0 0 0	01	0 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	x 0 0 0 0 0 0 0	10	1 0 0 0 0 0 0 0
J _a =A'B'C'D'E		K _a =A'B'C'D'E	
J _b	C	K _b	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	0 0 0 0 0 0 0 0	00	0 x 0 0 0 0 0 0
01	x 0 0 0 0 0 0 0	01	1 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	1 0 0 0 0 0 0 0	10	x 0 0 0 0 0 0 0
J _b =A'B'C'D'E'		K _b =A'B'C'D'E	
J _c	C	K _c	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	0 0 0 0 0 0 0 0	00	0 x 0 0 0 0 0 0
01	0 0 0 0 0 0 0 0	01	x 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	x 0 0 0 0 0 0 0
J _c =A'B'C'D'E'		K _c =A'B'C'D'E'	
J _d	C	K _d	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	0 0 0 0 0 0 0 0	00	0 x 0 0 0 0 0 0
01	0 0 0 0 0 0 0 0	01	1 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	x 0 0 0 0 0 0 0
J _d =A'B'C'D'E'		K _d =A'B'C'D'E	
J _e	C	K _e	C
AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100	AB ₁ CD ₁ E ₁	000 001 011 010 110 111 101 100
00	0 0 0 0 0 0 0 0	00	0 x 0 0 0 0 0 0
01	0 0 0 0 0 0 0 0	01	x 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0	11	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0	10	x 0 0 0 0 0 0 0
J _e =A'B'C'D'		K _e =A'B'C'D'	

Figura 18. Tabla de Karnoff, contador de suma de fase

5.6. Display

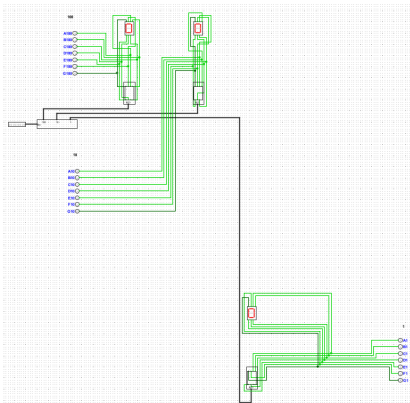


Figura 19. Display 7 segmentos

6. Integración del sistema

El sistema completo está diseñado de forma modular, donde cada componente cumple una función específica dentro del algoritmo de

reducción. La coordinación de estos módulos está a cargo de una Máquina de Estados Finitos (MEF), la cual gobierna el flujo general del sistema mediante señales de control.

Todos los componentes que dependen de una señal de reloj (CLK) poseen una entrada adicional llamada RUN. Esta entrada permite habilitar o detener su funcionamiento según el estado actual de la MEF. De este modo, la MEF determina en qué momento debe actuar cada bloque, asegurando un funcionamiento sincronizado y secuencial.

Secuencia de la MEF La MEF está compuesta por 8 estados, los cuales se ejecutan en el siguiente orden:

0. Incrementa el contador de 5 bits, que indica la posición actual del arreglo.

1. Actualiza el estado de la máscara y la suma de fase, si la posición B es igual a la suma de fase.

2. Lee el primer valor a comparar desde la RAM.

3. Guarda el primer valor en un registro temporal.

4. Lee el segundo valor a comparar desde la posición desplazada.

5. Guarda el segundo valor en un segundo registro.

6. Compara ambos valores utilizando el comparador jerárquico.

7. Guarda el valor menor (según el resultado de la comparación) en la posición correspondiente de la RAM.

Visualización del sistema Para facilitar la verificación del funcionamiento del sistema, se incorporaron varios displays de siete segmentos:

Dos displays están conectados a las salidas de los registros, permitiendo observar los valores almacenados en cada fase.

Otro display está conectado a la salida del multiplexor que selecciona el resultado del comparador, permitiendo validar visualmente el resultado de la operación.

Además, todos los componentes están interconectados de forma que permitan ejecutar correctamente cada fase del algoritmo de reducción sin intervención externa, y el sistema puede ser reiniciado globalmente mediante una señal de reset asíncrono.

7. Resultados y análisis

Durante las simulaciones realizadas en Logisim Evolution, se verificó el funcionamiento correcto del sistema de reducción diseñado para encontrar el valor menor entre pares de elementos de un arreglo de 32 datos de 8 bits.

En cada fase de la máquina de estados, el sistema fue capaz de:

Leer correctamente los datos desde la RAM.

Almacenar temporalmente los valores a comparar en registros.

Comparar los datos usando un comparador jerárquico de 8 bits construido a partir de bloques de 1 bit.

Seleccionar y almacenar en memoria el menor de los dos valores comparados.

Gracias al control preciso ejercido por la máquina de estados finita, cada fase se ejecutó secuencialmente de manera controlada, sin

generar errores ni resultados incorrectos. Se verificaron visualmente los valores intermedios usando displays conectados a los registros y al comparador, confirmando así el funcionamiento esperado en cada etapa.

El sistema demostró ser modular, facilitando su comprensión y validación por partes. El uso de señales de control como RUN, gobernadas por la MEF, permitió detener o habilitar componentes en momentos específicos, evitando conflictos de datos.

Finalmente, se observó que el sistema es capaz de reducir el arreglo original hasta obtener el valor mínimo global en fases sucesivas, optimizando el uso de recursos mediante reutilización de componentes.

8. Resultados y análisis

El diseño e implementación del sistema de reducción permitió aplicar conceptos clave de arquitectura digital como máquinas de estados, diseño jerárquico y control secuencial. Se logró comparar correctamente pares de datos para obtener el valor menor, utilizando únicamente los componentes vistos en clase.

La modularidad del sistema facilitó tanto su construcción como su depuración. Además, el uso de una MEF permitió controlar el flujo de operaciones de manera precisa, garantizando resultados correctos en cada fase del proceso de reducción.