**Álgebra de Circuitos Digitales y Función Boole.**

**Función Boole.**

Una función booleana es una relación matemática que convierte variables booleanas en una salida booleana, estás salidas solo pueden tener dos estados falso o verdadero, lo cuales se representan como 0 y 1 respectivamente, y son utilizados para simular el comportamiento lógico de circuitos digitales. [1]

Las funciones booleanas también se representan mediante expresiones algebraicas, ya sea suma de productos o producto de sumas, tablas de verdad, circuitos con compuertas lógicas o diagramas. Cada una de estas maneras de representar a las funciones booleanas tienes sus ventajas dependiendo del contexto que sea requerido. Estas funciones también se pueden utilizar como operaciones lógicas suma, resta, multiplicación y comparación. [2]

**Funciones y Operadores Booleanos.**

En el álgebra booleana encontramos los operadores booleanos, que son los operadores fundamentales para aplicar en variables binarias. Los más conocidos son:

* AND (∧): devuelve 1 si ambas entradas son 1.
* OR (∨): devuelve 1 si al menos una de las entradas son 1.
* NOT (¬): invierte el valor de la entrada.

De estos operadores se derivan otros operadores como XOR, NAND, NOR.

Las leyes del álgebra booleana, ya sea conmutativa, distributiva, De Morgan o nula, permiten simplificar expresiones booleanas sin cambiar su resultado booleano, verdadero o falso. Esto es indispensable para optimizar circuitos booleanos.[3]

**Álgebra de circuitos digitales.**

El álgebra de circuitos digitales es importante para el uso de expresiones booleanas y sirve para diseñar y optimizar circuitos lógicos. Gracias a esta se pueden mostrar requerimientos funcionales en diseños con puertas lógicas básicas (AND, OR, NOT) o compuertas compuestas (NAND, XOR, NOR). [4]

Un circuito digital puede ser combinacional, este depende de las entradas actuales, o secuencial, este no solo depende de las entradas actuales, sino que también de estados anteriores. Para los circuitos combinacionales, el álgebra booleana sirve para definir la lógica, lo cual nos permite construir circuitos mas eficientes, mejorar el control del flujo de datos y optimizar la cantidad de puertas lógicas utilizadas. [5]

**Análisis de álgebra booleana aplicada a circuitos lógicos y digitales.**

Este análisis nos permite diseñar sistemas con una alta precisión. Gracias a las tablas de verdad podemos escribir una función booleana que describe el comportamiento requerido en el circuito, luego de esto se utilizan técnicas para la simplificación o reducción de la función booleana para poder adaptarla al hardware usando la menor cantidad de componentes posibles. [6]

Los mapas de Karnaugh también son fundamentales para simplificar funciones booleanas, ya que nos ayuda a organizar lo valores de la tabla de verdad de manera visual en una cuadricula, donde cada una de sus celdas representa una combinación de sus entradas, luego se procede a agrupar la mayor cantidad de unos posibles con el fin de reducir la cantidad de operaciones lógicas necesarias en una función. Con lo cual de obtienes circuitos más pequeños, rápidos y eficientes. A diferencia del método algebraico, los mapas de Karnaugh nos ayudan a facilitar la simplificación porque al ser este un medio visual, la tarea se vuelve mas intuitiva. [7]

**Microcontroladores.**

**Características básicas de los microcontroladores.**

Los microcontroladores son circuitos integrados que tienen en un solo chip la CPU, memoria y periféricos. A diferencia de los microprocesadores, los microcontroladores están diseñados para controlar tareas específicas, lo que lo los hace ideales para aplicaciones autónomas. [8]

Un microcontrolador tiene 3 tipos de memorias: memoria de programa (Flash), memoria de datos (SRAM) y memoria de configuración (EEPROM). También tienes sus periféricos como convertidores analógico-digital (ADC), puertos GPIO, módulos de temporizador, contadores y mecanismos de comunicación serial. Gracias a estas características el microcontrolador puede interactuar eficientemente con el entorno físico y procesar datos en tiempo real. [9]

La arquitectura de los microcontroladores también viene optimizada para reducir el consumo energético y con el mismo poderío funcional. Esto gracias a que tienen modos de bajo consumo, interrupciones configurables y escalado dinámico de frecuencias. [10]

**Velocidad de procesamiento y ciclo de instrucciones.**

La velocidad de procesamiento de los microcontroladores depende directamente de su frecuencia de reloj y de la eficiencia del ciclo de instrucciones. Cada una de las instrucciones tiene un ciclo básico que incluye la fase de búsqueda (Fetch), decodificación (decode) y ejecución (execute). Todo esto depende del número de ciclos de reloj que requiere y de la arquitectura interna del microcontrolador. [11]

Otro aspecto clave es el uso de interrupciones, estas permiten que el microcontrolador responda a los eventos externos sin necesidad de consultar constantemente el estado de los dispositivos conectados. El análisis del ciclo de instrucción también es fundamental en las aplicaciones donde el tiempo real es un factor importante, ya que el cálculo preciso de cuantos ciclos consume cada una de las operaciones, con esto se puede prever el rendimiento del sistema y ajustar temporizadores de control en consecuencia. [12]

Bibliografias

[1] S. Kurgalin and S. Borzunov, “Boolean Algebra,” 2020, pp. 217–249. doi: 10.1007/978-3-030-42221-9\_6.

[2] K. Erciyes, “Boolean Algebras and Combinational Circuits,” 2021, pp. 173–195. doi: 10.1007/978-3-030-61115-6\_9.

[3] R. Drechsler and S. Huhn, Eds., *Advanced Boolean Techniques*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-28916-3.

[4] S. Huhn and R. Drechsler, *Design for Testability, Debug and Reliability*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-69209-4.

[5] D. M. Miller and G. W. Dueck, “Translation Techniques for Reversible Circuit Synthesis with Positive and Negative Controls,” in *Recent Findings in Boolean Techniques*, Cham: Springer International Publishing, 2021, pp. 143–165. doi: 10.1007/978-3-030-68071-8\_7.

[6] A. Mahzoon, D. Große, and R. Drechsler, *Formal Verification of Structurally Complex Multipliers*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-24571-8.

[7] M. Garrido, “Simplifying Karnaugh Maps by Making Groups of Non-power-of-two Elements,” *Circuits Syst Signal Process*, vol. 41, no. 10, pp. 5895–5902, Oct. 2022, doi: 10.1007/s00034-022-02040-4.

[8] C. Ünsalan, H. D. Gürhan, and M. E. Yücel, *Embedded System Design with ARM Cortex-M Microcontrollers*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-88439-0.

[9] C. Ünsalan, B. Höke, and E. Atmaca, *Embedded Machine Learning with Microcontrollers*. Cham: Springer International Publishing, 2025. doi: 10.1007/978-3-031-69421-9.

[10] E. H. Currie, “Microcontroller Subsystems,” in *Mixed-Signal Embedded Systems Design*, Cham: Springer International Publishing, 2021, pp. 35–97. doi: 10.1007/978-3-030-70312-7\_2.

[11] B. Goossens, *Guide to Computer Processor Architecture*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-18023-1.

[12] P. Bulić, *Understanding Computer Organization*. Cham: Springer International Publishing, 2024. doi: 10.1007/978-3-031-58075-8.