**Aritmética Binaria**

La aritmética binaria es todo un conjunto de operaciones matemáticas lógicas, que se realizan o se expresan en base 2. Este sistema es utilizado en casi todos los artefactos tecnológicos digitales modernos porque se adapta y se acopla perfectamente a la tecnología electrónica, además se utiliza únicamente dos dígitos que son el 0 y 1. Esta elección no es arbitraria en el sistema binario, si no es una consecuencia directa de la naturaleza en base de la naturaleza física de los circuitos, los cuales solo pueden estar en dos estados: apagado y encendido, así también como alto o bajo, 1 o 0.[1]

Representar cualquier numero en el sistema de datos binario mediante varias combinaciones de bits. En el numero decimal por ejemplo el 5 se representa como un 101 en binario. Siendo esto que la verdadera potencia del sistema binario se da a conocer al realizar operaciones aritméticas, tales como la suma, resta, multiplicación y división, ejercicios los cuales son la base de todo procesamiento digital ya sea en datos de información vital para el usuario.[2]

Para la suma binaria, como regla básica se basa de manera simple con estas reglas 0+0=0, 0+1=1, 1+0=1 y 1 + 1 = 10, dando esto como resultado de 0 y se lleva 1 al siguiente bit. Este es fundamental para lo que permiten construir sumadores, que son circuitos fundamentales en unidades de control aritmético lógicas (ALU) de procesadores. La resta binaria sigue una lógica similar pero simplificada al diseño de circuitos restadores. Complementando así una estructura ordenada de pasos.[3]

A diferencia del sistema decimal que utiliza dígitos del 0 al 9, en cambio en el sistema binario o arimética binaria se usan apenas dos valores que son el 0 y 1, cada una de sus posiciones en una secuencia binaria se pueden representar en una potencia de 2, siendo un poco similar al sistema decimal solo que en este se representa en cada posición con una potencia de 10.[4]

En la multiplicación binaria se basa de varias sumas repetidas simultáneamente, al multiplicar algún numero por 1 significa mantenerlo o conservarlo para luego usarlo con el siguiente, y multiplicarlo 0 lo anula, son muy eficientes para implementarlas en hardware.[4]

Por otro lado, en la división binaria se pueden realizar utilizando técnicas similares a la división larga, pero en decimal, aunque tendríamos que adaptarlas al sistema binario.

**Función Booleana**

La función Booleana es una expresión matemática lógica, desarrollada por George Boole en el sigo XIX con la idea para poder representar relaciones de forma lógica usando variables binarias. En electrónica digital, las funciones Booleanas se volvieron fundamentales con esto ahora se usan para modelar el comportamiento de circuitos lógicos y creación de todo tipo de sistemas computacionales.[5]

Siendo una expresión booleana que toma valores binarios tanto de entrada 0 y 1 como también poder producir un valor binario de salida. Estas son funciones implementadas y diseñadas dando vida a los circuitos digitales, con esto poder implementar decisiones lógicas, dentro de condicionales de un sistema al que se vaya a ser usado.[6]

Una función Booleana puede representarse de varias formas como una tabla de verdad, una expresión algebraica o un circuito lógico. Las **operaciones lógicas** básicas de la función o lógica Booleana son AND conjunción, OR disyunción y NOT negación. Mediante de estas tres operaciones se pueden dar otras características mas complejas o funciones mas complejas como NAND, NOR, XOR y XNOR.[7]

Dicho que los circuitos en la vida real es necesario que se minimicen el numero de compuertas, esto para poderse optimizar el circuito, se utilizan técnicas de simplificación. El mapa de Karnaugh es una de las más comunes, una herramienta que permite agrupar combinaciones visualmente y con esto poder simplificar claro aplicando las leyes de la algebra Booleana.[8]

Se pueden construir todo tipo de dispositivos digitales con la ayuda de las funciones booleanas, desde una alarma que se activa mediante ciertos parámetros condicionados por procesadores enteros y sistemas conectados entre sí.[9]

Los decodificadores activan una única salida según la combinación de entradas, los codificadores hacen lo opuesto, multiplexores dan múltiples señales de entrada y salidas para poder interpretarlas y las memorias o registros las cuales se usan para determinar cuando guarda o liberan datos. Así como en la programación en la lógica booleana también es fundamental las instrucciones condicionales if, while, etc. Estas se basan en operaciones Booleanas que evalúan si se cumple aquella condición o no.[10]

[1] H. Winther, “SPIN REPRESENTATIONS AND BINARY NUMBERS,” *Archivum Mathematicum*, vol. 60, no. 4, pp. 231–241, 2024, doi: 10.5817/AM2024-4-231.

[2] T. Ndjountche, “Digital Electronics,” in *Digital Electronics*, Robert Baptist, Ed., Great Britain United States: ISTE Ltd and John Wiley & Sons, Inc., 2016, ch. 1. doi: 10.1002/9781119318620.fmatter.

[3] X. D. Li *et al.*, “Classification feature selection and dimensionality reduction based on logical binary sine-cosine function arithmetic optimization algorithm,” *Egyptian Informatics Journal*, vol. 26, Jun. 2024, doi: 10.1016/j.eij.2024.100472.

[4] M. P. Véstias and H. C. Neto, “Decimal multiplication in fpga with a novel decimal adder/subtractor,” Jul. 01, 2021, *MDPI AG, Lisboa, Portugal*. doi: 10.3390/a14070198.

[5] R. Malik, V. Paranjape, and M. Kulkarni, “Circuit Optimization using Arithmetic Table Lookups,” *Proceedings of the ACM on Programming Languages*, vol. 9, Jun. 2025, doi: 10.1145/3729258.

[6] E. Kubaczka *et al.*, “Energy Aware Technology Mapping of Genetic Logic Circuits,” *ACS Synth Biol*, vol. 13, no. 10, pp. 3295–3311, Oct. 2024, doi: 10.1021/acssynbio.4c00395.

[7] B. Steinbach and C. Posthoff, “Derivative Operations for Lattices of Boolean Functions,” Trinidad y Tobago, Sep. 2013. doi: 10.13140/2.1.2398.6568.

[8] T. Higuchi *et al.*, “Real-world applications of analog and digital evolvable hardware,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 3, pp. 220–234, Sep. 1999, doi: 10.1109/4235.788492.

[9] B. Sampathkumar, R. Das, B. Martin, F. Enescu, and P. Kalla, “An Algebraic Approach to Partial Synthesis of Arithmetic Circuits,” in *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, Institute of Electrical and Electronics Engineers Inc., Mar. 2025, pp. 1097–1103. doi: 10.1145/3658617.3697724.

[10] T. Higuchi *et al.*, “Real-world applications of analog and digital evolvable hardware,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 3, pp. 220–234, Sep. 1999, doi: 10.1109/4235.788492.