1.- Aritmética Binaria

¿Qué es la aritmética binaria?

La aritmética binaria es el sistema primario para la representación y manipulación de la información en los dispositivos digitales. Este sistema se diferencia del decimal, que emplea diez símbolos, por basarse solamente en dos: el cero (0) y el uno (1). Esta característica no es casualidad y permite a las computadoras realizar sus operaciones internas con las señales eléctricas que representan la presencia de voltaje (1) o su ausencia (0). Tal como explican Dubey et al. (2020), la eficiencia de las operaciones binaras se ve reflejada en la rapidez y confiabilidad de los procesos aritméticos realizados por arquitecturas digitales modernas [1].

Origen de la aritmética binaria

Gottfried Wilhelm Leibniz fue el primero en proponer el sistema binario en el siglo XVII. Su interpretación filosófica vinculaba el 1 y el 0 con conceptos metafísicos, como el ser y la nada. Sin embargo, fue George Boole quien en el siglo XIX introdujo el álgebra booleana, y posteriormente Claude Shannon en el siglo XX quien conectó dicha lógica con los circuitos eléctricos, dando origen al diseño de sistemas digitales. Como se documenta en la Universal Numbers Library de Omtzigt et al. (2020), este enfoque sentó las bases para representar todo tipo de datos mediante cadenas binarias [2].

Operaciones básicas y estructuras lógicas

En el sistema binario, las operaciones aritméticas se ejecutan siguiendo reglas específicas. La suma binaria, por ejemplo, se basa en combinaciones de 0 y 1, donde 1+1 da como resultado 10, lo que implica un acarreo. Las operaciones se implementan mediante compuertas lógicas como AND, OR y XOR. Según lo describen Qian et al. (2021), el diseño de aceleradores en hardware para operaciones binarias, como los utilizados en codificadores de video UHDTV, permite ejecutar dichas operaciones de forma paralela y eficiente [3].

Tabla: Suma binaria de un bit

A	В	A + B
0	0	0
0	1	1
1	0	1
1	1	10

Representación de números negativos

Una de las mejorías del sistema binario es la forma en la que es capaz de representar los valores negativos mediante el complemento a dos, que se lleva a cabo invirtiendo los valores de cada uno de los bits de un número y sumando uno. Esto permite a las ALUs llevar a cabo sumas y restas indistintamente de los signos. Como señalan Aljaedi et al. (2023), esta representación es fundamental para algoritmos criptográficos como la multiplicación de puntos sobre curvas elípticas, donde la eficiencia de la aritmética binaria mejora la seguridad sin sacrificar rendimiento [4].

Ejemplo:

Número	Representación en 8 bit
+5	00000101
-5	11111011 (complemento a 2)

Precisión binaria y sistemas avanzados

La representación de números reales en binario no es siempre exacta, lo cual introduce pequeños errores de redondeo. Por ello, el IEEE desarrolló el estándar 754-2019, que define formatos binarios de punto flotante, reglas de redondeo y tratamiento de excepciones. Este estándar permite garantizar interoperabilidad y precisión numérica en diferentes plataformas. Como indican Damsgaard et al. (2023), la identificación de oportunidades para la aproximación aritmética en hardware de borde permite reducir complejidad y consumo de energía sin perder exactitud significativa, lo que refuerza la utilidad de la aritmética binaria en entornos de recursos limitados [5].

Aritmética binaria en sistemas neuromórficos

Los sistemas neuromórficos pretenden imitar el cerebro humano de una manera binaria para simular el disparo que llevaría por parte de las neuronas. Para llevarlo a cabo se desarrolló una arquitectura de multiplicación en punto flotante usando una red de neuronas artificiales de tal manera que valida que la aritmética binaria es compatible con los paradigmas de computación emergente [1]. Esto refuerza la idea de que el sistema binario se adapta incluso más allá de los esquemas de computación clásica.

Nuevas representaciones: sistemas logarítmicos

Alam et al. (2021) van aún más allá al apuntar que determinados sistemas numéricos logarítmicos de baja precisión, como los que se basan en potencias de dos no estándar, también pueden ofrecer mayor eficiencia que la aritmética binaria convencional para las tareas que requieren bajo consumo y alta velocidad, como, por ejemplo, la inteligencia artificial en el borde (*Edge computing*) [6]. No obstante, estos no reemplazan a los anteriores; son simplemente alternativas que permiten seguir trabajando, a la espera de que el estado de la técnica permita su uso eficiente en dispositivos de recursos limitados.

Conclusión

La aritmética binaria es la base de los sistemas digitales. Su simplicidad, alta eficiencia y elasticidad para ser adaptada a diferentes formatos ha permitido su uso en un gran número de aplicaciones, en el ámbito de la criptografía, en la inteligencia artificial, etc. Las mejoras en la representación y en el cálculo, junto con su fusión con sistemas neuromórficos son indicios que probablemente podamos contar con ella en el futuro. Entonces, conociendo su funcionamiento, la historia y su uso, podemos valorar la importante aportación de la aritmética binaria a la revolución tecnológica.

2.- Aritmética Decimal

¿Qué es la aritmética decimal?

La aritmética decimal es el sistema de numeración más utilizado por los seres humanos. Se basa en diez símbolos (del 0 al 9) y sigue un principio posicional en el que la ubicación del dígito determina su valor. Aunque las computadoras operan internamente en binario, el sistema decimal sigue siendo fundamental para aplicaciones financieras, contables, científicas y de interacción con el usuario. Como señalan Mian et al. (2021), la necesidad de incorporar aritmética decimal en hardware ha llevado al desarrollo de estrategias de co-diseño entre software y hardware que mejoran la eficiencia en sistemas modernos [7].

Evaluación de hardware en ecosistemas RISC-V

Los microcontroladores y sistemas embebidos que implementan las arquitecturas RISC-V han comenzado la incorporación de unidades funcionales para el manejo de aritmética decimal, lo cual facilita la adaptación de operaciones financieras / contables a entornos de bajo consumo. De

acuerdo con Shintani et al. (2019), se han desarrollado métodos de evaluación ciclo-a-ciclo para analizar el rendimiento del código decimal implementado en hardware, logrando mayor precisión y menor latencia [8].

Diseño de sumadores BCD

Uno de los componentes fundamentales en la aritmética decimal que se lleva a cabo en hardware son los sumadores BCD (Binary-Coded Decimal), que permiten codificar cada uno de los dígitos decimal a partir de un grupo de cuatro bits binarios. Chu et al. (2021) proponen un diseño eficiente de sumadores BCD basados en puertas XOR de tres entradas y funciones de mayoría, lo que reduce significativamente el área y el consumo de energía [9].

Tabla de Código BCD de los dígitos 0 a 9

Decimal	Digito en BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Precisión decimal: exactitud vs. redondeo

En los sistemas de punto flotante, la representación en decimal puede llegar a ser exacta o inexacta. Esto depende de si los valores pueden representarse exactamente como se encuentran en el formato usado o si se debe realizar redondeo. Al respecto, coinciden Muhamed et al. (2023), quienes también explican que esta diferencia puede tener implicaciones importantes en cálculos de tipo financiero y en simulaciones de alta precisión, cuando el redondeo acumulativo puede dar lugar a errores significativos [10].

Representaciones de alta precisión

El impulso de representar valores numéricos elevados con gran fidelidad ha propiciado la creación de un número elevado de nuevos esquemas numéricos. Tal y como manifiestan los autores de la publicación de 2023 de *Advancements in number representation for high precision computing*, han sido investigadas algunas arquitecturas mixtas que utilizan la representación decimal y la representación binaria para coincidir el esquema decimal con el esquema binario y a la vez contemplar lo mejor de ambos sistemas [11]. Estas soluciones son especialmente útiles en simulaciones científicas, meteorología, y en cálculo financiero de alto rendimiento.

Conversión y parsing decimal

La transformación de cadenas decimales en representaciones numéricas eficaces presenta un enorme reto en sistemas de gran volumen de datos. Según Lemire (2021), puede analizar el conjunto de datos numéricos en cadenas de texto a velocidades alcanzables de gigabytes por segundo con los algoritmos optimizados correspondientes, lo que incide de forma notable en el rendimiento en entornos de big data y bases de datos [12].

Aritmética decimal en bibliotecas universales

Por último, Omtzigt et al. (2020) implementan operaciones para realizar operaciones decimales de alta precisión mediante la Universal Numbers Library en arquitecturas modernas que simulan diferentes formatos y permiten el uso de la reproducibilidad de los resultados; esta librería ha desempeñado un papel importante para el progreso de soluciones que requieran portabilidad entre diferentes plataformas [2].

Conclusión

La implementación de la aritmética decimal es fundamental en aplicaciones donde la precisión y la interpretabilidad son críticas. Aunque su implementación en hardware es más compleja incluso que la aritmética binaria, los trabajos más recientes donde coexisten co-diseño HW/SW, arquitecturas RISC-V y sumadores BCD prometen soluciones que la hacen más viable y eficiente. Los trabajos relacionados con el *parsing* numérico y las bibliotecas universales afianzan su papel como pieza clave en el equipaje de la computación moderna, especialmente en los casos en los que los errores no son aceptables.

Bibliografía

- [1] K. Dubey, U. Kothari, and S. Rao, "Floating-Point Multiplication Using Neuromorphic Computing," Aug. 2020.
- [2] E. T. L. Omtzigt, P. Gottschling, M. Seligman, and W. Zorn, "Universal Numbers Library: design and implementation of a high-performance reproducible number systems library," Dec. 2020.
- [3] S. Qian, Y. Hu, Y. Wu, W. Yang, X. Bai, and R. Shao, "A Pipelined Multi-Tiered Hardware Acceleration Approach Towards Content Addressable Binary Arithmetic," in *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, New York, NY, USA: ACM, Oct. 2021, pp. 1297–1302. doi: 10.1145/3501409.3501638.
- [4] A. Aljaedi, S. S. Jamal, M. Rashid, A. R. Alharbi, M. Alotaibi, and D. J. Alanazi, "Area-Efficient Realization of Binary Elliptic Curve Point Multiplication Processor for Cryptographic Applications," *Applied Sciences*, vol. 13, no. 12, p. 7018, Jun. 2023, doi: 10.3390/app13127018.
- [5] H. J. Damsgaard, A. Ometov, and J. Nurmi, "Approximation Opportunities in Edge Computing Hardware: A Systematic Literature Review," *ACM Comput Surv*, vol. 55, no. 12, pp. 1–49, Dec. 2023, doi: 10.1145/3572772.
- [6] S. A. Alam, J. Garland, and D. Gregg, "Low-precision Logarithmic Number Systems," *ACM Transactions on Architecture and Code Optimization*, vol. 18, no. 4, pp. 1–25, Dec. 2021, doi: 10.1145/3461699.
- [7] R. Mian, M. Shintani, and M. Inoue, "Hardware–Software Co-Design for Decimal Multiplication," *Computers*, vol. 10, no. 2, p. 17, Jan. 2021, doi: 10.3390/computers10020017.
- [8] R. Mian, M. Shintani, and M. Inoue, "Cycle-Accurate Evaluation of Software-Hardware Co-Design of Decimal Computation in RISC-V Ecosystem," in 2019 32nd IEEE International System-on-Chip Conference (SOCC), IEEE, Sep. 2019, pp. 412–417. doi: 10.1109/SOCC46988.2019.1570559752.
- [9] Z. Chu, Z. Li, Y. Xia, L. Wang, and W. Liu, "BCD Adder Designs Based on Three-Input XOR and Majority Gates," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 6, pp. 1942–1946, Jun. 2021, doi: 10.1109/TCSII.2020.3047393.

- [10] M. F. Mudawar, "Exact Versus Inexact Decimal Floating-Point Numbers and Arithmetic," *IEEE Access*, vol. 11, pp. 17891–17905, 2023, doi: 10.1109/ACCESS.2023.3244891.
- [11] H. Mora, M. T. Signes-Pont, F. A. P. López, J. Mora-Pascual, and J. M. G. Chamizo, "Advancements in number representation for high-precision computing," *J Supercomput*, vol. 80, no. 7, pp. 9742–9761, May 2024, doi: 10.1007/s11227-023-05814-y.
- [12] D. Lemire, "Number parsing at a gigabyte per second," *Softw Pract Exp*, vol. 51, no. 8, pp. 1700–1727, Aug. 2021, doi: 10.1002/spe.2984.