

ARQUITECTURAS SIMD, MIMD y MISD

1st Ednan Josué Merino Calderón 2nd Angelo Patricio Sánchez Sarabia 3rd Justin Joshua Villarroel Barreno
Universidad de las Fuerzas Armadas Sangolquí, Ecuador *Universidad de las Fuerzas Armadas Sangolquí, Ecuador* *Universidad de las Fuerzas Armadas Sangolquí, Ecuador*
ejmerino@espe.edu.ec apsanchez9@espe.edu.ec jjvillarroel1@espe.edu.ec

I. RESUMEN

Las arquitecturas SIMD son fundamentales en la informática paralela, permitiendo el procesamiento simultáneo de datos mediante múltiples procesadores que ejecutan la misma operación. Esto se logra gracias a una unidad de control central que coordina la ejecución sincronizada de instrucciones en diferentes conjuntos de datos. Su aplicación se extiende a campos diversos como el procesamiento de imágenes, la simulación física y el análisis de secuencias de ADN, ofreciendo una eficiencia y velocidad significativas en el procesamiento de grandes volúmenes de datos. Estas arquitecturas son ampliamente utilizadas en aplicaciones exigentes, como supercomputadoras y sistemas de alto rendimiento, así como en dispositivos como cámaras digitales. Permiten ejecutar instrucciones idénticas en múltiples procesadores, cada uno con su conjunto único de datos. Esto se logra mediante una unidad de control central que envía instrucciones a varios procesadores, los cuales interactúan simultáneamente con cada CPU. Las aplicaciones prácticas de las arquitecturas SIMD incluyen la vectorización de algoritmos, el procesamiento de imágenes, la simulación de sistemas físicos, el análisis de secuencias de ADN y el filtrado de datos. Estas aplicaciones se benefician de la capacidad de procesamiento paralelo de SIMD, lo que permite una mayor eficiencia y velocidad en el procesamiento de datos.

II. INTRODUCCIÓN

En el vasto panorama de la informática, la búsqueda constante de mejorar el rendimiento y la eficiencia en el procesamiento de datos ha llevado al desarrollo de diversas arquitecturas de hardware. Entre estas, las arquitecturas SIMD (Single Instruction, Multiple Data) se destacan como una herramienta fundamental para la aceleración de cálculos en una amplia gama de aplicaciones computacionales. Las arquitecturas SIMD están diseñadas para ejecutar una misma instrucción en paralelo sobre múltiples conjuntos de datos, aprovechando así el potencial del paralelismo a nivel de instrucción. Esta capacidad de procesamiento masivo y simultáneo ha encontrado aplicaciones en áreas tan diversas como el procesamiento de imágenes, la simulación física, el análisis de datos y la inteligencia artificial, entre otras. A lo largo de las décadas, las arquitecturas SIMD han evolucionado significativamente, desde sus primeras implementaciones hasta las complejas unidades de procesamiento SIMD que se encuentran en los procesadores modernos. Esta evolución ha sido

impulsada tanto por avances en la tecnología de semiconductores como por la demanda creciente de potencia de cálculo en aplicaciones cada vez más complejas.

III. OBJETIVOS

- Entender, mediante una investigación, las diferentes arquitecturas de la Computación Paralela y extender las conceptualizaciones hacia las aplicaciones prácticas de estas arquitecturas. Entender, mediante una investigación, las diferentes arquitecturas de la Computación.
- Realizar un análisis exhaustivo de los diferentes tipos de arquitecturas presentes en la Computación Paralela. Enfocar su investigación y análisis en aplicaciones prácticas de la arquitectura SIMD.

IV. MARCO TEÓRICO

A. Definición

Las arquitecturas SIMD son esenciales en el mundo de las computadoras paralelas, debido a su habilidad para manejar grandes vectores y matrices de datos en tiempos muy cortos. El secreto detrás de este tipo de arquitectura es que cuentan con un varios procesadores ejecutando la misma operación sobre un conjunto de datos [1]. Es una técnica empleada para conseguir paralelismo a nivel de datos [2]. Los repertorios SIMD implican el uso de instrucciones que ejecutan una operación idéntica en múltiples datos simultáneamente. En esta estructura, una unidad de control central envía instrucciones a varias unidades de procesamiento. Aunque todas las unidades reciben la misma instrucción, trabajan con conjuntos de datos diferentes. Esto significa que la ejecución de la instrucción ocurre de forma coordinada entre todas las unidades de procesamiento [2].

Las arquitecturas SIMD ejecutan la misma instrucción sobre múltiples datos simultáneamente:

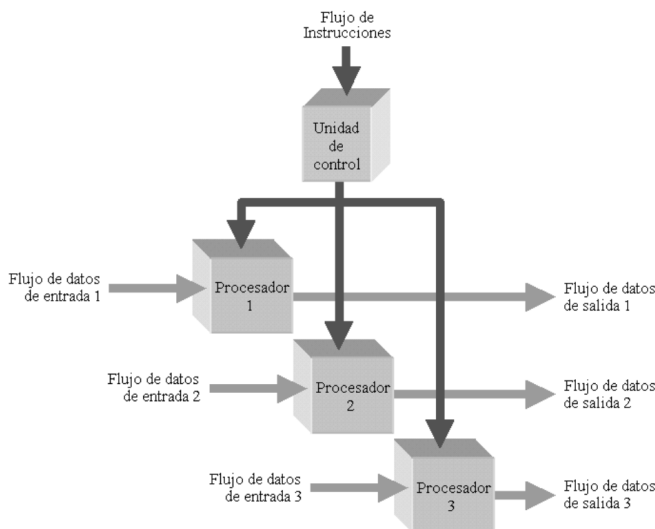


Fig. 1: Diagrama de bloques de la arquitectura SIMD

B. Funcionamiento

Al momento de ejecutar una instrucción asignada al CPU, este empieza a aplicar dicha instrucción con cada elemento que ingrese la registro SIMD de manera inmediata. Esta ejecución paralela permite realizar dicha instrucción en varios elementos en el mismo ciclo, logrando así el paralelismo, lo que brinda mayor versatilidad y fluidez de procesos [8]. Se han desarrollado dos arquitecturas diferentes, procesadores de matriz y procesadores vectoriales los cuales fueron desarrollados para abordar estas estructuras. Un procesador de matriz tiene muchos elementos de procesador que operan en paralelo muchos elementos de datos [9]. Un procesador vectorial tiene un solo elemento procesador que opera en secuencia en muchos elementos de datos [9].

C. Arquitecturas

Las arquitecturas SIMD y MIMD son mayormente utilizadas para el procesamiento de tareas exigentes, en especial en supercomputadoras, computadoras de alto desempeño o en procesos que requieran una gran cantidad de procesamiento [3]. Uno de los dispositivos donde se puede encontrar procesadores embebidos con capacidad de SIMD serían las cámaras digitales [5]. Las computadoras que utilizan la arquitectura SIMD (Instrucción Única, Datos Múltiples) poseen múltiples procesadores que realizan instrucciones idénticas. Aun que, cada procesador proporciona las instrucciones con su colección única de datos [7]. Las computadoras SIMD emplean el mismo algoritmo a varios grupos de datos. La arquitectura SIMD cuenta con diversos componentes de procesamiento, los cuales están bajo la responsabilidad de una única unidad de control. Mientras procesa numerosos datos, cada uno recibe la misma instrucción de la unidad de control. Varios módulos incluidos en el subsistema compartido ayudan en la interacción simultánea con cada CPU [7]. Los procesadores operan al mismo tiempo: en cada paso, los procesadores ejecutan la misma instrucción sobre diferentes datos. La instrucción

puede ser simple (sumar o comparar dos números) o compleja (la mezcla de dos listas de números). De igual forma, los datos pueden ser simples (un número) o complejos (un conjunto de números). Los procesadores que están inactivos durante la ejecución de una instrucción o que completan la ejecución de la instrucción antes que los demás están desocupados hasta que se da la próxima instrucción. [10]

D. Aplicaciones Prácticas

- **Vectorización:** La arquitectura SIMD nos permite la vectorización de algoritmos, lo que podemos intuir que una misma operación se puede aplicar a un conjunto de datos simultáneamente, mejorando significativamente el rendimiento del computador.
- **Algoritmo de Kolte:** Al momento de implementar el filtro mediano utilizando el algoritmo vectorial planteado por Kolte, el cual puede llegar a ser beneficiado por la arquitectura SIMD, ayudando a procesar múltiples datos a la vez [4].
- **Procesamiento de imágenes:** Dado que los algoritmos que son utilizados en el procesamiento digital de las imágenes pueden llegar a ser optimizados si usamos la arquitectura SIMD, debido a que su proceso consiste en la iteración de una instrucción en una gran cantidad de datos.
- **Simulación de Sistemas Físicos (sistemas moleculares):** Debido al uso continuo de cálculos repetitivos en una gran cantidad de datos, la arquitectura SIMD es una muy buena opción al momento de optimizar y acelerar el cálculo de estas operaciones.
- **Análisis de secuencias de ADN:** Al momento de alinear las secuencias de ADN, se puede utilizar la arquitectura SIMD para acelerar el proceso, donde podemos usar pocas instrucciones para analizar y procesar varios datos, esto permitiendo una alineación eficiente y reduciendo el tiempo de proceso [6].
- **Filtrado de datos:** En una empresa es necesario que se realicen los filtrados o clasificación de datos de manera eficiente, pero sobre todo de manera rápida, por lo que la arquitectura SIMD se adapta de mejor manera para esta tarea, brindando la clasificación de los datos con una instrucción de manera paralela.

E. Arquitectura MIMD

La clase MIMD de arquitectura paralela es la forma más familiar y posiblemente la más básica de procesador paralelo: consta de múltiples elementos de procesador interconectados. A diferencia del procesador SIMD, cada elemento del procesador se ejecuta de forma completamente independiente (aunque normalmente es el mismo programa). Aunque no es necesario que todos los elementos del procesador sean idénticos, la mayoría de las configuraciones MIMD son homogéneas y todos los elementos del procesador son idénticos [12].

F. Problemas MIMD

Cuando las comunicaciones entre elementos del procesador se realizan a través de un espacio de direcciones

de memoria compartida (ya sea global o distribuida entre elementos del procesador, denominada memoria compartida distribuida para distinguirla de la memoria distribuida) [12].

Interferencia de accesos a memoria. En general, la memoria no puede ser accedida simultáneamente por dos o más procesadores, y por lo tanto los requerimientos se deben secuencializar. De esta manera, el acceso a memoria se realiza más lentamente cuanto mayor sea la cantidad de conflictos entre los procesadores por accesos a la memoria principal [12]. Interferencia en la red de comunicación, o carga de la red de comunicación. La red que conecta a los procesadores con la memoria compartida es otro de los recursos a los cuales se debe acceder y no tiene capacidad ilimitada [12]. Tiempo de latencia. A medida que la cantidad de procesadores crece, la red de interconexión con la memoria se hace más compleja, y por lo tanto el tiempo de latencia de las redes de comunicación es mayor [12].

G. Aplicaciones MIMD

Las arquitecturas MIMD pueden utilizarse de diversas maneras, dependiendo de la aplicación. Por ejemplo, en una aplicación de procesamiento de vídeo, cada procesador podría ser responsable de procesar un cuadro diferente del vídeo. En una aplicación de minería de datos, cada procesador podría ser responsable de buscar patrones en un conjunto de datos diferente [11]

Las arquitecturas MIMD se utilizan a menudo en aplicaciones de computación de alto rendimiento, en las que el objetivo es alcanzar el mayor nivel de rendimiento posible. Las arquitecturas MIMD también se utilizan en muchas aplicaciones comerciales, como el procesamiento de vídeo y la minería de datos [11]

Arquitecturas MIMD:

- **Diseño Asistido:** Las máquinas MIMD se utilizan en aplicaciones de diseño asistido, como el diseño de circuitos integrados, modelado 3D y simulación de sistemas complejos.
- **Simulación y Modelado:** Las arquitecturas MIMD son ideales para aplicaciones de simulación y modelado, como simulaciones de física, dinámica de fluidos, y análisis de sistemas complejos.
- **Interruptores y Enrutadores:** Los dispositivos de red, como interruptores y enrutadores, a menudo utilizan arquitecturas MIMD para procesar grandes volúmenes de datos de manera eficiente.
- **Procesamiento de Vídeo:** Las aplicaciones de procesamiento de vídeo, como la compresión y descompresión de video, también se benefician de las arquitecturas MIMD.
- **Minería de Datos:** Las arquitecturas MIMD se utilizan en aplicaciones comerciales, como la minería de datos y el análisis de grandes conjuntos de datos.

H. Arquitectura MISD

Multiple instrucción, Single Data, es un tipo de arquitectura computacional donde muchas unidades funcionales realizan diferentes operaciones en los mismos datos. Las arquitecturas segmentadas pertenecen a este tipo, aunque en un extremo se podría llegar a decir que los datos son diferentes después de ser procesados por cada etapa en el pipeline, con lo cual no entraría en esta categoría.

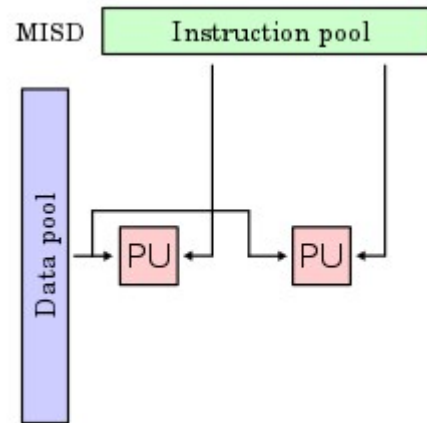


Fig. 2: Arquitectura MISD

I. Aplicaciones MISD

Pueden encontrarse en varios campos donde se requiera el procesamiento de datos de manera eficiente y simultánea. [13]

- **Procesamiento de imágenes:** En aplicaciones donde se necesite procesar grandes cantidades de datos visuales, como el procesamiento de imágenes médicas, gráficos por computadora o análisis de imágenes satelitales
- **Simulaciones y modelado:** la simulación de fenómenos naturales, el modelado de sistemas complejos o la simulación de procesos industriales
- **Criptografía y seguridad:** En aplicaciones donde se requiera un alto grado de seguridad y encriptación, como en la protección de datos sensibles
- **Procesamiento de señales:** En campos como el procesamiento de audio, el procesamiento de señales biomédicas o la comunicación inalámbrica, donde se procesan grandes volúmenes de datos en tiempo real [14]

V. RESULTADOS DE APRENDIZAJE

- **Análisis de ingeniería:** La capacidad de identificar, formular y resolver problemas de ingeniería en su especialidad; elegir y aplicar de forma adecuada métodos analíticos, de cálculo y experimentales ya establecidos; reconocer la importancia de las restricciones sociales, de salud y de seguridad, ambientales, económicas e industriales.
- **Comunicación y trabajo en equipo:** Capacidad para comunicar eficazmente información, ideas,

problemas y soluciones en el ámbito de ingeniería y con la sociedad en general.

VI. CONCLUSIONES

- En el contexto de SIMD, es crucial considerar la adaptabilidad y flexibilidad en el diseño de sistemas computacionales. Cada arquitectura, desde la memoria compartida hasta los clústeres distribuidos, tiene ventajas y desventajas que deben sopesarse según las necesidades de las aplicaciones.
- La relevancia de la paralelización en el desempeño informático es destacada. Se ha comprobado que la habilidad para repartir tareas entre varios procesadores aumenta de manera notable la eficacia y la velocidad de ejecución de programas computacionales complejos.
- Se resaltan los desafíos presentes en la programación y la optimización de código para sistemas paralelos. Es crucial diseñar algoritmos que aprovechen eficientemente el paralelismo ofrecido por la arquitectura SIMD, así como optimizar la comunicación entre elementos de datos para maximizar el rendimiento.
- Esta área sigue siendo objeto de investigación activa, con el propósito de mejorar tanto el rendimiento como la escalabilidad de las aplicaciones que hacen uso de SIMD.
- Aunque SIMD ofrece un alto rendimiento en aplicaciones con patrones de datos regulares y repetitivos, puede enfrentar desafíos al tratar con conjuntos de datos irregulares o dependencias de datos complejas. Estos casos pueden requerir estrategias adicionales de optimización y programación para aprovechar al máximo la arquitectura SIMD.
- La arquitectura MISD ofrece eficiencia mediante la ejecución simultánea de múltiples instrucciones sobre un solo conjunto de datos, lo que la hace valiosa en aplicaciones que requieren procesamiento paralelo.
- Al aprovechar al máximo los recursos de hardware disponibles, la arquitectura MISD puede mejorar la velocidad y la eficiencia del procesamiento de datos en comparación con otras arquitecturas.
- Aunque presenta ventajas significativas, su aplicabilidad puede ser limitada en contextos donde el paralelismo no es factible o eficiente, lo que sugiere áreas de mejora y desarrollo continuo.
- La arquitectura MIMD (Instrucción Múltiple, Datos Múltiples) es un paradigma de diseño de computadoras que se utiliza para realizar una variedad de cálculos, incluyendo simulaciones científicas, procesamiento de video, minería de datos y más. Mejorando la productividad de los procesos, a comparación de la arquitectura SIMD, la arquitectura MIMD tiene una mayor capacidad de abordar problemas complejos y variados, siendo más adecuados para su aplicación.

REFERENCES

- [1] UNAM. (2021). Capítulo 1: Introducción. En *Arquitectura de Computadoras*.
- [2] colaboradores de Wikipedia. (2020, 30 junio). SIMD. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/SIMD>
- [3] Morgado Valle, A. (1994). Introducción a la computación paralela. *Polibits*, 12, 3-7.
- [4] Inca, E. L. (2004). Estimación de densidad por kernel y sus aplicaciones utilizando programación paralela. University of Puerto Rico, Mayaguez (Puerto Rico).
- [5] Sánchez Loayza, R. M. Diseño e implementación del filtro mediano de dos dimensiones para arquitecturas SIMD.
- [6] Alachiotis, N., Berger, S.A. & Stamatakis, A. Coupling SIMD and SMT architectures to boost performance of a phylogeny-aware alignment kernel. *BMC Bioinformatics* 13, 196 (2012). <https://doi.org/10.1186/1471-2105-13-196>
- [7] Vive. (2024, enero 25). La computación paralela: características, tipos y usos. UNIR. <https://www.unir.net/ingenieria/revista/computacion-paralela/>
- [8] SIMD: Single instruction, multiple data. (2023, mayo 17). Hardly Working. <https://hardlyworking.dev/posts/simd/>
- [9] Lo, W. Y., Lun, D. P. K., Siu, W. C., Wang, W., & Song, J. (2011). Improved SIMD architecture for high performance video processors. *IEEE transactions on circuits and systems for video technology*, 21(12), 1769-1783.
- [10] Arquitectura paralela. (s/f). Numerentur.org. Recuperado el 14 de mayo de 2024, de <https://numerentur.org/arquitecturas-de-multiples-procesadores/>
- [11] Zara. (2023, julio 31). Instrucción múltiple, datos múltiples (MIMD) - Definición y explicación. TechEdu. <https://techlib.net/techedu/instruccion-multiple-datos-multiples-mimd/>
- [12] Tinetti, F. G., & De Giusti, A. E. (1998). *Procesamiento paralelo*.
- [13] Quinn, Michael J. *Programación paralela en C con MPI y OpenMP*. Boston: McGraw Hill, 2004.
- [14] Ibaroudene, Djaffer. "Procesamiento paralelo, EG6370G: Capítulo 1, Motivación e Historia". Universidad de St Mary, San Antonio, TX. Primavera de 2008.