



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLOGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Morelia

Arquitectura de Computadoras

Unidad 4: Procesamiento paralelo.

Presentado por:

**Alberto Montoya Arriaga
Michael Aaron Villalón Nieves**

Ingeniería en Sistemas Computacionales

Docente:

Servando González Hernández

Morelia, Michoacán, México. 12 de diciembre de 2025.

Índice

1. Introducción.	3
2. Taxonomía de Flynn.	4
2.1. SISD (Single Instruction, Single Data)	5
2.2. SIMD (Single Instruction, Multiple Data)	5
2.3. MISD (Multiple Instruction, Single Data)	6
2.4. MIMD (Multiple Instruction, Multiple Data)	7
2.5. Resumen	8
3. Multiprocesamiento.	8
3.1. Simétrico (SMP)	9
3.2. Heterogéneo	10
3.3. Cluster	10
4. Tipos de paralelismo.	11
4.1. A nivel de bit	12
4.2. A nivel de instrucción	12
4.3. De datos	13
4.4. De tareas	14
5. Cuestionario.	14
6. Conclusión.	17
7. Bibliografía.	18

1. Introducción.

El procesamiento paralelo es un método del campo de la computación que permite que dos o más procesadores de una computadora se utilicen para trabajar en partes separadas de una tarea. De esta manera, es posible reducir el tiempo dedicado a resolver el problema.

El concepto de computación paralela comenzó a desarrollarse a finales de la década de 1950 por investigadores de IBM. Ellos creían que una sola computadora no satisfaría la creciente demanda de potencia de procesamiento. Una posible solución sería tener dos procesadores (o núcleos) trabajando simultáneamente.

El primer chip comercial con múltiples núcleos fue el IBM Power4, lanzado en 2001. El procesador, basado en la arquitectura PowerPC, era un dual-core con una frecuencia de 1,1 a 1,3 GHz. La CPU, que fue la primera en tener dos núcleos en un solo chip de silicio, se fabricaba en una litografía de 180 nanómetros.

Hay varias formas diferentes de computación paralela: Paralelismo a nivel de bit, paralelismo a nivel de instrucción, paralelismo de datos y paralelismo de tareas. El paralelismo se ha empleado durante muchos años, sobre todo en la computación de altas prestaciones, pero el interés en ella ha crecido últimamente debido a las limitaciones físicas que impiden el aumento de la frecuencia. Como el consumo de energía —y por consiguiente la generación de calor— de las computadoras constituye una preocupación en los últimos años, la computación en paralelo se ha convertido en el paradigma dominante en la arquitectura de computadores, principalmente en forma de procesadores multinúcleo.

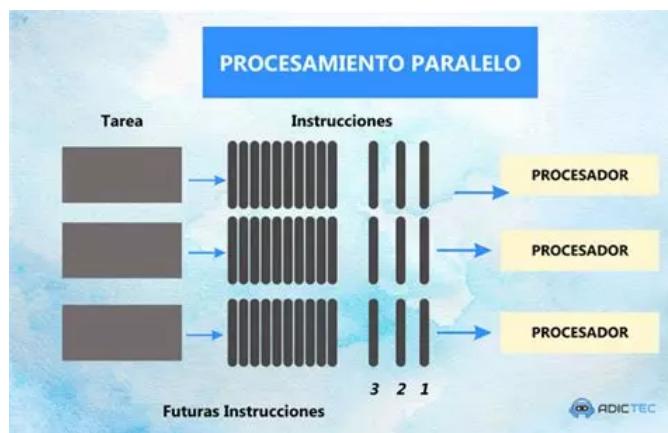


Imagen 1: Imagen de ilustracion.

2. Taxonomía de Flynn.

La taxonomía de Flynn es un sistema de clasificación de arquitecturas que se basa en la idea de cuántos flujos de instrucciones y cuántos flujos de datos puede manipular una computadora simultáneamente. Fue creada por el científico de la computación Michael J. Flynn en 1966 y se utiliza hasta el día de hoy en el concepto de computación paralela.

El flujo de instrucciones es una secuencia de instrucciones ejecutadas por el procesador. Una instrucción es una “orden” dada al chip para realizar una operación específica, como una adición o una resta. Cuando tenemos varias órdenes consecutivas, tenemos un flujo de instrucciones.

Por otro lado, el flujo de datos es el conjunto de datos sobre los cuales se ejecutan las instrucciones. Volviendo al ejemplo matemático anterior, los números que se sumarían o restarían serían el flujo de datos en un procesador.

La taxonomía de Flynn divide los sistemas en cuatro categorías:

- **SISD** (Single Instruction, Single Data): es el modelo de computadora más simple, en el cual una sola instrucción opera en un solo flujo de datos. La mayoría de los primeros chips funcionaban de esta manera.
- **MISD** (Multiple Instruction, Single Data): varias instrucciones se ejecutan sobre los mismos datos. Es un modelo teórico e inusual en el mundo real.
- **SIMD** (Single Instruction, Multiple Data): una sola instrucción se aplica a varios flujos de datos simultáneamente. Puede ser útil en procesadores vectoriales, como los utilizados en NPUs para inteligencia artificial.
- **MIMD** (Multiple Instruction, Multiple Data): varias instrucciones operan en varios flujos de datos. Es el modelo más utilizado en los procesadores modernos con múltiples núcleos y se puede encontrar en todas las categorías de dispositivos electrónicos, desde pequeños teléfonos inteligentes hasta grandes servidores.

2.1. SISD (*Single Instruction, Single Data*)

Flujo único de instrucciones y flujo único de datos. Una computadora secuencial que no aprovecha el paralelismo ni en las instrucciones ni en los flujos de datos. La unidad de control único (CU) obtiene un único flujo de instrucciones (IS) de la memoria. Luego, la CU genera señales de control apropiadas para dirigir un solo elemento de procesamiento (PE) para operar en un solo flujo de datos (DS), es decir, una operación a la vez.

Ejemplos de arquitecturas SISD son las máquinas tradicionales de un solo procesador, como las computadoras personales (PC) más antiguas (en 2010, muchas PC tenían varios núcleos) y las computadoras centrales.

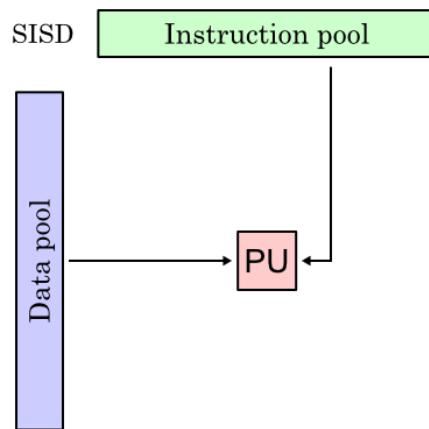


Imagen 2: Diseño de la categoría SISD.

2.2. SIMD (*Single Instruction, Multiple Data*)

Flujo de instrucción simple y flujo de datos múltiple. Esto significa que una única instrucción es aplicada sobre diferentes datos al mismo tiempo. En las máquinas de este tipo, varias unidades de procesado diferentes son invocadas por una única unidad de control. Al igual que las MISD, las SIMD soportan procesamiento vectorial (matricial) asignando cada elemento del vector a una unidad funcional diferente para procesamiento concurrente.

El artículo de Flynn de 1972 subdividió SIMD en tres categorías adicionales:

- **Procesador de rayos:** Estos reciben la instrucción (samo) pero cada unidad de procesamiento paralelo tiene su propio archivo de memoria y registro separado y distinto.

- **Procesador de tuberías:** Estos reciben la instrucción (samo) pero luego leen los datos de un recurso central, cada uno procesa fragmentos de esos datos, luego escribe los resultados al mismo recurso central. En la Figura 5 del papel de Flynn 1972 que el recurso es la memoria principal: para las CPU modernas que el recurso es ahora más típicamente el archivo de registro.
- **Procesador asociado:** Estos reciben la instrucción (samo) pero en cada unidad de procesamiento paralelo una independiente se toma la decisión sobre la base de datos local a la unidad, en cuanto a si realizar la ejecución o si saltarla. En terminología moderna esto se conoce como "predicado"(masked) SIMD.

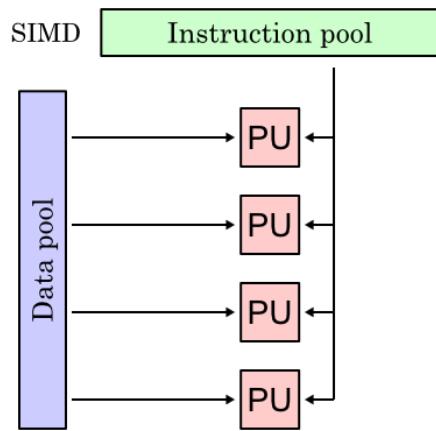


Imagen 3: Diseño de la categoría SIMD.

2.3. MISD (*Multiple Instruction, Single Data*)

Flujo múltiple de instrucciones y único flujo de datos. Esto significa que varias instrucciones actúan sobre el mismo y único trozo de datos. Este tipo de máquinas se pueden interpretar de dos maneras. Una es considerar la clase de máquinas que requerirían que unidades de procesamiento diferentes recibieran instrucciones distintas operando sobre los mismos datos. Esta clase de arquitectura ha sido clasificada por numerosos arquitectos de computadores como impracticable o imposible, y en estos momentos no existen ejemplos que funcionen siguiendo este modelo.

Las arquitecturas segmentadas, o encauzadas, realizan el procesamiento vectorial a través de una serie de etapas, cada una ejecutando una función particular produciendo un resultado intermedio. La razón por la cual dichas arquitecturas son clasificadas como MISD es que los elementos de un vector pueden ser

considerados como pertenecientes al mismo dato, y todas las etapas del cauce representan múltiples instrucciones que son aplicadas sobre ese vector.

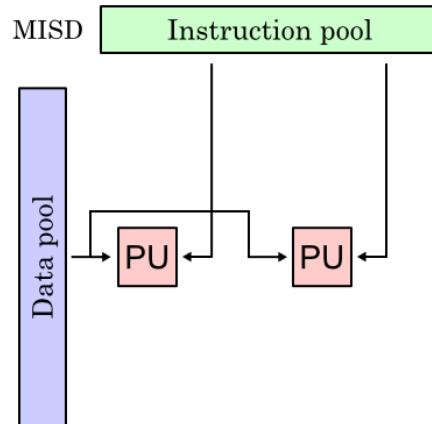


Imagen 4: Diseño de la categoría MISD.

2.4. MIMD (*Multiple Instruction, Multiple Data*)

Flujo de instrucciones múltiple y flujo de datos múltiple. Son máquinas que poseen varias unidades procesadoras en las cuales se pueden realizar múltiples instrucciones sobre datos diferentes de forma simultánea. Las MIMD son las más complejas, pero son también las que potencialmente ofrecen una mayor eficiencia en la ejecución concurrente o paralela. Aquí la concurrencia implica que no sólo hay varios procesadores operando simultáneamente, sino que además hay varios programas (procesos) ejecutándose también al mismo tiempo.

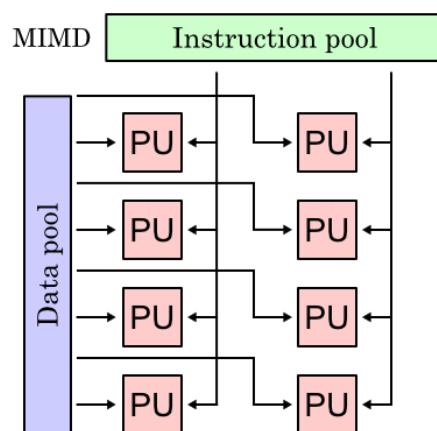


Imagen 5: Diseño de la categoría MIMD.

2.5. Resumen

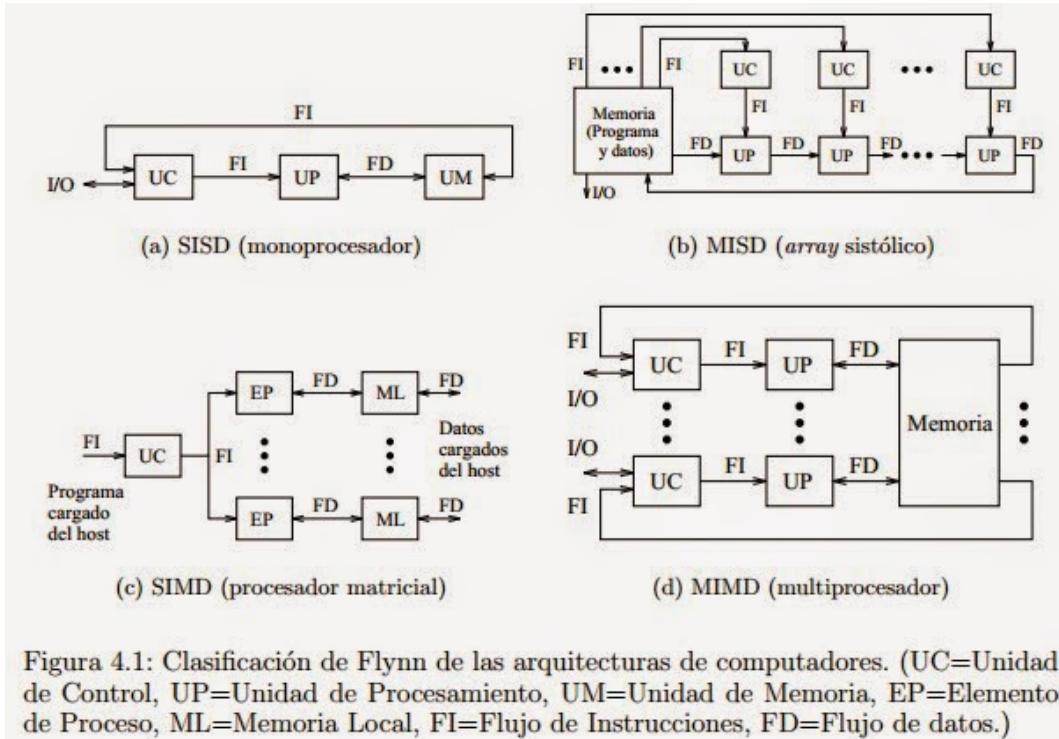


Imagen 6: Resumen de los diseños de cada taxonomía

3. Multiprocesamiento.

Multiprocesamiento o multiproceso es el uso de dos o más procesadores (CPU) en una computadora para la ejecución de uno o varios procesos (programas corriendo). Algunas personas, en el idioma español hacen sinónimo este término con el de multitareas (del inglés multitasking) el cual consiste en la ejecución de uno o más procesos concurrentes en un sistema. Así como la multitarea permite a múltiples procesos compartir una única CPU, múltiples CPU pueden ser utilizados para ejecutar múltiples procesos o múltiples hilos (threads) dentro de un único proceso.

El multiprocesamiento se ha empleado desde los años 60 en los entornos de cómputo de alto rendimiento; a pesar de esto, durante muchos años no muchos tomaban esta área de especialización, una computadora que contara con más de un procesador era cara y debido a esto muchos decidían no hacer uso de más de un procesador. Hasta que en 2005 y después de cumplirse 40 años del modelo conocido como Ley de Moore, se empezaron a exceder los 3 GHz de velocidad de esta manera creando problemas de calentamiento motivando el uso de múltiples procesadores.

3.1. Simétrico (SMP)

El multiprocesamiento simétrico o multiprocesamiento de memoria compartida (SMP) involucra una arquitectura de hardware y software de computadora multiprocesador donde dos o más procesadores idénticos están conectados a una sola memoria principal compartida, tienen acceso total a todos los dispositivos de entrada y salida, y están controlados por una sola instancia del sistema operativo que trata a todos los procesadores por igual, sin reservar ninguno para propósitos especiales. La mayoría de los sistemas multiprocesador actuales utilizan una arquitectura SMP. En el caso de los procesadores multinúcleo, la arquitectura SMP se aplica a los núcleos y los trata como procesadores independientes.

Los sistemas SMP permiten que cualquier procesador trabaje en cualquier tarea sin importar su localización en memoria; con un propicio soporte del sistema operativo, estos sistemas pueden mover fácilmente tareas entre los procesadores para garantizar eficientemente el trabajo.

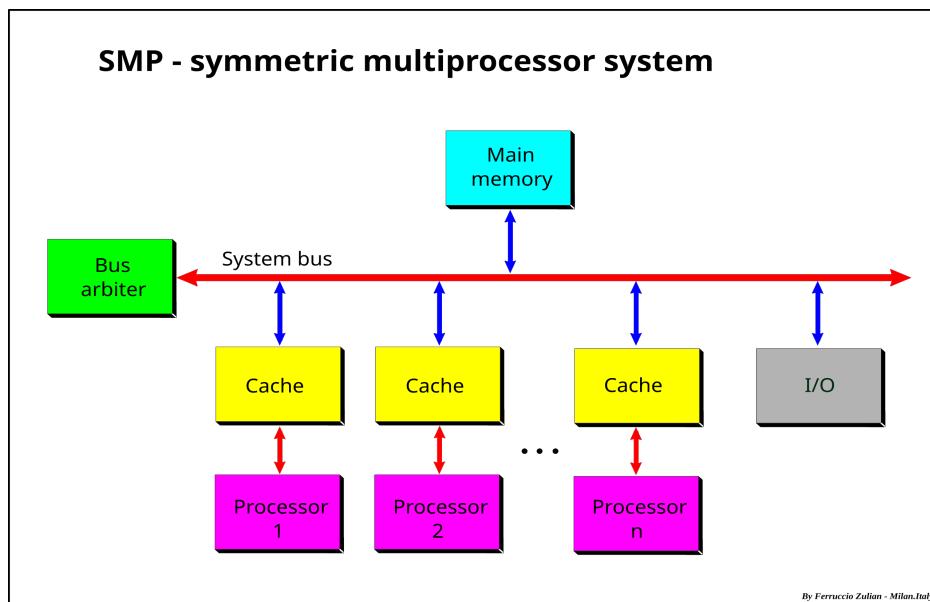


Imagen 7: Diseño del tipo SMP

3.2. Heterogéneo

Se refiere a un enfoque de computación que utiliza diferentes tipos de procesadores o arquitecturas para llevar a cabo tareas específicas. Este método permite aprovechar las fortalezas de cada tipo de procesador, como CPUs, GPUs y FPGAs, optimizando así el rendimiento y la eficiencia energética. En el contexto de sistemas distribuidos y procesamiento de datos, el procesamiento heterogéneo se vuelve crucial, ya que permite la ejecución de cargas de trabajo diversas en entornos que requieren escalabilidad y flexibilidad.

3.3. Cluster

Un clúster es un tipo de arquitectura paralela distribuida que consiste de un conjunto de computadores independientes interconectados operando de forma conjunta como único recurso computacional sin embargo, cada computador puede utilizarse de forma independiente o separada.

En esta arquitectura, el computador paralelo es esencialmente una colección de procesadores secuenciales, cada uno con su propia memoria local, que pueden trabajar conjuntamente.

- Cada nodo tiene rápido acceso a su propia memoria y acceso a la memoria de otros nodos mediante una red de comunicaciones, habitualmente una red de comunicaciones de alta velocidad.
- Los datos son intercambiados entre los nodos como mensajes a través de la red.
- Una red de ordenadores, especialmente si disponen de una interconexión de alta velocidad, puede ser vista como un multicomputador de memoria distribuida y como tal ser utilizada para resolver problemas mediante computación paralela.

La construcción de los ordenadores del clúster es más fácil y económica debido a su flexibilidad: pueden tener toda la misma configuración de hardware y sistema operativo diferente rendimiento pero con arquitectura y sistemas operativos similares o tener diferente hardware y sistema operativo lo que hace más fácil y económica su construcción. Para que un clúster funcione como tal no basta solo con conectar entre si los ordenadores, sino que es necesario proveer un sistema de manejo del clúster, el cual se encargue de interactuar con el usuario y los procesos que ocurren en él para optimizar el funcionamiento.

4. Tipos de paralelismo.

Una de las características clave del multiprocesamiento es la capacidad de realizar múltiples tareas al mismo tiempo. Esto se logra al distribuir las tareas entre los diferentes procesadores, lo que permite una mayor eficiencia en el manejo de la carga de trabajo.

El paralelismo puede estudiarse a varios niveles:

- **Trabajo:** Dos programas distintos pueden ejecutarse en paralelo.
- **Tarea:** En este nivel se consideran varias tareas independientes entre si formando parte de un programa determinado. Es posible la interacción de las tareas.
- **Proceso:** Varios procesos componen una tarea. Son bloques con funcionalidad bien definida.
- **Variable:** El paralelismo puede darse a nivel de variables ya que varias instrucciones pueden ser ejecutadas en paralelo siendo el punto de conflicto las variables en común.
- **Bit:** Todos los computadores usan paralelismo a nivel de bit.

Algunos ámbitos en los que se usa la computación paralela.

- Simulación de modelos complejos
- Diseño y automatización de proyectos de ingeniería
- Exploración petrolera y minera
- Medicina
- Área militar
- Cine: efectos visuales, animación 3D
- Realidad Virtual
- Comercio electrónico
- Mega bases de datos (google, youtube, rapidshare)

4.1. A nivel de bit

Se habla de paralelismo al nivel de bit, cuando se aumenta el tamaño de la palabra del procesador (tamaño de la cadena de bits a procesar). Este aumento reduce el número de instrucciones que tiene que ejecutar el procesador en variables cuyos tamaños sean mayores a la longitud de la cadena.

Ejemplo: En un procesador de 8-bits sumar dos números de 16bits tomaría dos instrucciones. En un procesador de 16-bits esa operación requiere solo una instrucción.

Desde el advenimiento de la integración a gran escala (VLSI) como tecnología de fabricación de chips de computadora en la década de 1970 hasta alrededor de 1986, la aceleración en la arquitectura de computadores se lograba en gran medida duplicando el tamaño de la palabra en la computadora, la cantidad de información que el procesador puede manejar por ciclo.

El aumento del tamaño de la palabra reduce el número de instrucciones que el procesador debe ejecutar para realizar una operación en variables cuyos tamaños son mayores que la longitud de la palabra.

Históricamente, los microprocesadores de 4 bits fueron sustituidos por unos de 8 bits, luego de 16 bits y 32 bits, esta tendencia general llegó a su fin con la introducción de procesadores de 64 bits, lo que ha sido un estándar en la computación de propósito general durante la última década.

4.2. A nivel de instrucción

Un programa de ordenador es, en esencia, una secuencia de instrucciones ejecutadas por un procesador. Estas instrucciones pueden reordenarse y combinarse en grupos que luego son ejecutadas en paralelo sin cambiar el resultado del programa. Esto se conoce como paralelismo a nivel de instrucción. Los avances en el paralelismo a nivel de instrucción dominaron la arquitectura de computadores desde mediados de 1980 hasta mediados de la década de 1990.

Los procesadores modernos tienen "pipeline" de instrucciones de varias etapas. Cada etapa en el pipeline corresponde a una acción diferente que el procesador realiza en la instrucción correspondiente a la etapa; un procesador con un pipelinede N etapas puede tener hasta n instrucciones diferentes en diferentes

etapas de finalización.

Además del paralelismo a nivel de instrucción del pipelining, algunos procesadores pueden ejecutar más de una instrucción a la vez. Estos son conocidos como procesadores superescalares. Las instrucciones pueden agruparse juntas sólo si no hay dependencia de datos entre ellas.

4.3. *De datos*

El paralelismo de datos es el paralelismo inherente en programas con ciclos, que se centra en la distribución de los datos entre los diferentes nodos computacionales que deben tratarse en paralelo. La paralelización de ciclos conduce a menudo a secuencias similares de operaciones —no necesariamente idénticas— o funciones que se realizan en los elementos de una gran estructura de datos. Muchas de las aplicaciones científicas y de ingeniería muestran paralelismo de datos.

Una dependencia de terminación de ciclo es la dependencia de una iteración de un ciclo en la salida de una o más iteraciones anteriores. Las dependencias de terminación de ciclo evitan la paralelización de ciclos.

Extensiones vectoriales como SSE, AVX o NEON permiten aplicar una misma instrucción sobre varios datos simultáneos. Es ideal para operaciones uniformes sobre arreglos: procesamiento de imágenes, señales o vectores numéricos. Requiere que los datos estén alineados y que las operaciones sean homogéneas para evitar divergencias.

- **Beneficio:** gran aumento del throughput (cantidad de trabajo completado por unidad de tiempo) sin aumentar el número de hilos.
- **Condiciones:** datos contiguos y alineados, bucles vectorizables y pocas dependencias.
- **Limitación:** código con muchas ramas o acceso irregular a memoria pierde eficiencia vectorial.

4.4. De tareas

Divide el problema en tareas independientes que pueden ejecutarse en paralelo. Cada tarea puede tener diferente trabajo y duración. Se implementa con hilos, pools de tareas o frameworks específicos. Funciona bien cuando el problema se puede fragmentar en unidades que comparten poco estado y pueden balancearse entre núcleos.

El desafío principal es el balanceo de carga: tareas muy desiguales dejan núcleos ociosos. Estrategias como dividir en unidades más pequeñas (work stealing) ayudan a usar todo el hardware disponible.

5. Cuestionario.

1. **¿Qué caracteriza a la arquitectura MIMD?** *Se caracteriza por tener múltiples flujos de instrucciones y múltiples flujos de datos, permitiendo que varios procesadores ejecuten diferentes instrucciones sobre distintos datos de forma simultánea.*
2. **Las arquitecturas MIMD permiten la ejecución concurrente de varios programas al mismo tiempo.** *Verdadero*
3. **¿Cuál de las siguientes arquitecturas es la más compleja pero también la más eficiente para ejecución paralela?**
 - a) *SISD*
 - b) *SIMD*
 - c) *MISMD*
 - d) *MIMD*

Respuesta correcta: d) MIMD

4. **¿Qué es el multiprocesamiento?** *Es el uso de dos o más procesadores (CPU) en una computadora para ejecutar uno o varios procesos de manera simultánea.*
5. **Multiprocesamiento y multitarea significan exactamente lo mismo.** *Falso*

6. ¿Cuál fue una de las principales razones para el auge del multiprocesamiento después de 2005? *Las limitaciones de velocidad por calentamiento al superar los 3 GHz, lo que motivó el uso de múltiples procesadores en lugar de aumentar la frecuencia.*
7. ¿Qué ley se menciona en relación con el crecimiento del hardware y sus límites?
- Ley de Amdahl*
 - Ley de Shannon*
 - Ley de Moore*
 - Ley de Newton*
- Respuesta correcta: c) Ley de Moore*
8. ¿Qué es un sistema SMP? *Es una arquitectura de multiprocesamiento simétrico donde varios procesadores idénticos comparten una sola memoria principal y están controlados por un único sistema operativo.*
9. En un sistema SMP, algunos procesadores están reservados para tareas especiales. *Falso*
10. ¿Qué componente comparten todos los procesadores en una arquitectura SMP?
- Caché privada*
 - Memoria principal*
 - Sistema Operativo distinto*
 - GPU dedicada*
- Respuesta correcta: b) Memoria principal*
11. ¿Qué significa multiprocesamiento heterogéneo? *Es un enfoque que utiliza diferentes tipos de procesadores o arquitecturas (CPU, GPU, FPGA) para ejecutar tareas específicas aprovechando sus fortalezas.*
12. El procesamiento heterogéneo ayuda a mejorar la eficiencia energética y el rendimiento. *Verdadero*
13. ¿Qué es un clúster de computadoras? *Es una arquitectura paralela distribuida formada por computadoras independientes interconectadas que trabajan juntas como un único recurso computacional.*

14. ¿Cómo intercambian datos los nodos de un clúster?

- a) *Mensajes a través de la red*
- b) *Registros internos*
- c) *Memoria compartida*
- d) *Caché global*

Respuesta correcta: a) Mensajes a través de la red

15. Un clúster solo funciona si todos los nodos tienen exactamente el mismo hardware y sistema operativo. *Falso*

16. ¿Qué es el paralelismo a nivel de bit? *Es el paralelismo que se logra aumentando el tamaño de la palabra del procesador; permitiendo procesar más bits por instrucción.*

17. ¿Qué tamaño de palabra es el estándar actual en la computación de propósito general?

- a) 16 bits
- b) 32 bits
- c) 64 bits
- d) 128 bits

Respuesta correcta: c) 64 bits

18. El paralelismo a nivel de bit reduce el número de instrucciones necesarias para operar con datos grandes. *Verdadero*

19. ¿Qué es el paralelismo a nivel de instrucción? *Es la ejecución simultánea de varias instrucciones reorganizadas de manera que no alteren el resultado del programa.*

20. ¿Cómo se llama la técnica que divide la ejecución de una instrucción en varias etapas?

- a) *Multitarea*
- b) *Pipeline*
- c) *Clustering*
- d) *Vectorización*

Respuesta correcta: b) Pipeline

21. Los procesadores superescalares pueden ejecutar más de una instrucción al mismo tiempo. *Verdadero*

22. **¿Qué es el paralelismo de datos?** *Es la ejecución paralela de operaciones similares sobre distintos elementos de una estructura de datos, común en ciclos y aplicaciones científicas.*
23. **¿Qué extensiones permiten aplicar una instrucción a varios datos simultáneamente?**
- a) *HTTP, FTP*
 - b) *BIOS, UEFI*
 - c) *RAID, LVM*
 - d) *SSE, AVX, NEO*

Respuesta correcta: d) SSE, AVX, NEO

24. **El código con muchas ramas pierde eficiencia en el paralelismo vectorial.** *Verdadero*
25. **¿Cuál es el principal desafío del paralelismo de tareas?** *El balanceo de carga, ya que tareas desiguales pueden dejar algunos núcleos ociosos.*

6. Conclusión.

La computación paralela y el multiprocesamiento representan una evolución clave en la arquitectura de los sistemas de cómputo modernos. A través de taxonomías como SISD, SIMD, MISD y MIMD, así como de arquitecturas como SMP, sistemas heterogéneos y clústeres, es posible aprovechar mejor los recursos de hardware para ejecutar múltiples instrucciones y procesar grandes volúmenes de datos de manera simultánea. Los distintos tipos de paralelismo (a nivel de bit, instrucción, datos y tareas) permiten optimizar el rendimiento, la eficiencia y la escalabilidad de los sistemas, siendo fundamentales en áreas científicas, industriales y comerciales actuales.

7. Bibliografía.

1. Adictech. (2024, April 9). ¿Qué es la Computación Paralela?: Beneficios y Aplicaciones. Adictec - Adicción Por La Tecnología. <https://adictec.com/que-es-computacion-paralela/>
2. de, B. (2025). Curso - 4.1 Aspectos Básicos de la Computación Paralela. Google.com. <https://sites.google.com/itmexicali.edu.mx/arquitectura-de-computadoras/4-procesamiento-paralelo/4-1-aspectos-b%C3%A1sicos-de-la-computaci%C3%B3n-paralela>
3. Taxonomía de Flynn AcademiaLab. (2024). Academia-Lab.com. <https://academia-lab.com/enciclopedia/taxonomia-de-flynn/>
4. de, T. (2025). Curso - 4.2 Tipos de Computación Paralela. Google.com. <https://sites.google.com/itmexicali.edu.mx/arquitectura-de-computadoras/4-procesamiento-paralelo/4-2-tipos-de-computaci%C3%B3n-paralela?authuser=0>
5. de, C. (2005, February 13). uso de dos o más procesadores (CPU) en una computadora para la ejecución de uno o varios procesos. Wikipedia.org; Wikimedia Foundation, Inc. <https://es.wikipedia.org/wiki/Multiprocesamiento>
6. Multiprocesamiento simétrico AcademiaLab. (2024). Academia-Lab.com. <https://academia-lab.com/enciclopedia/multiprocesamiento-simetrico/>
7. Procesamiento Heterogéneo. (2025). Glosarix. <https://glosarix.com/glossary/procesamiento-heterogeneo/>
8. de, S. (2025). Curso - 4.4 Sistemas de Memoria Distribuida. Multicomputadores: Clusters. Google.com. <https://sites.google.com/itmexicali.edu.mx/arquitectura-de-computadoras/4-procesamiento-paralelo/4-4-sistemas-de-memoria-distribuida-multicomputadores-clusters?authuser=0>
9. Todo sobre el multiprocesamiento: definición y tipos más comunes. (2024). Venceya.com. <https://venceya.com/multiprocesamiento-en-que-consiste-tipos/>
10. Introducción a las Arquitecturas Paralelas. (n.d.). <https://users.exa.unicen.edu.ar/catedras/arqui2/arqui2/filminas/Introduccion%20a%20las%20arquitecturas%20Paralelas.pdf>

11. Programación Paralela. (2017). Github.io. https://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela_teoria/index.html#thirteen
12. Moisset, D. (2025). Arquitecturas de Paralelismo. <https://www.tutorialesprogramacionya.com/computacionparalelaconcurrente/introduccion/tema4.html>