TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE MORELIA

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

MATERIA: **Arquitectura de Computadoras**

NOMBRE: Temario de Exposición
NÚMERO: **Unidad 4**

Nombre del Maestro: **Servando González Hernández**

Nombre de los Integrantes del Equipo: **Andres Arroyo Cervantes**
Fidel Villegas Hernández

Horario y Gpo. De Laboratorio: **Lunes 17:00 - 19:00, AA**

Fecha de Realización: **27/11/2023**
Fecha de Entrega: **26/11/2023**

Contenido

1. TAXONOMÍA DE FLYNN

La taxonomía de Flynn es una clasificación para las computadoras con arquitectura paralela, propuesta por el profesor emérito de la Universidad de Stanford Michael J. Flynn, la cual clasifica a las mismas atendiendo a la cantidad de instrucciones y flujo de datos concurrentes en un instante de procesamiento. Esta taxonomía enuncia 4 clasificaciones en función de las siguientes características:

- Número de procesadores
- Número de programas
- Estructura de memoria

Las cuales son:

1.1. SISD (Single Instruction Single Data)

Esta clasificación se refiere a las computadoras tradicionales y secuenciales en las cuales una instrución a la vez se ejecuta sobre un único dato cada ciclo de reloj. Los datos en cuestión se almacenan en una única memoria en la cual se usan técnicas como la segmentación para evitar errores de fragmentación interna. Un ejemplo sencillo de estas computadoras son los antiguos mainframe basados en la arquitectura de Von-Neumann.

- Un solo flujo de instrucciones es ejecutado por un solo procesador en un solo conjunto de datos.

- Este es el modelo de computación más básico y se asemeja a una máquina convencional de un solo procesador.

1.2. SIMD (Single Instruction Multiple Data)

Esta arquitectura representa la ejecución de una misma instrucción sobre un conjunto de datos. La misma es comúnmente vista en ciclos de programación que ejecutan una misma instrucción una y otra vez sobre datos de un arreglo o conjunto de datos. En la arquitectura SIMD estos datos son procesados por múltiples CPU que ejecutan la misma instrucción sobre una parte del conjunto o arreglo, cada uno, hasta llegar a procesar la totalidad de los mismos.

- Un solo flujo de instrucciones se ejecuta en paralelo en múltiples conjuntos de datos.

- Se utiliza para aprovechar la paralelización en problemas que involucran el mismo tipo de operaciones en grandes conjuntos de datos, como en aplicaciones gráficas o científicas.

1.3. MISD (Multiple Instruction Single Data)

Arquitectura que se refiere a múltiples instrucciones ejecutándose sobre un único dato. Comúnmente se considera esta arquitectura poco práctica ya que en tiempo de ejecución la efectividad del paralelismo requiere un múltiple flujo de datos y, además, el acceso concurrente a un mismo dato en memoria puede ocasionar que un CPU tenga que esperar a que el recurso(dato) esté disponible para poder acceder a él.

- Múltiples flujos de instrucciones operan en paralelo sobre un solo conjunto de datos.

- Este modelo es menos común en la práctica y generalmente se aplica a situaciones especializadas, como la redundancia para la tolerancia a fallas.

1.4. MIMD (Multiple Instruction Multiple Data)

Esta arquitectura representa a un conjunto de instrucciones que se ejecutan sobre un conjunto múltiple de datos. La misma es muy usada hoy en día para explotar el paralelismo ya sea con memoria distribuida y memoria compartida o híbridos como los clústers de computadoras. Muchos multiprocesadores modernos (como los de la tecnología Core i de Intel) entran en esta clasificación.[2]

- Múltiples flujos de instrucciones se ejecutan en paralelo en múltiples conjuntos de datos.

- Este modelo es común en sistemas multiprocesador y multicomputadora, donde cada procesador puede ejecutar su propio conjunto de instrucciones en datos independientes.

La taxonomía de Flynn proporciona un marco conceptual para entender las arquitecturas de computadoras paralelas y sus capacidades de procesamiento. La mayoría de las computadoras modernas caen en la categoría de SISD o MIMD, ya que son sistemas de un solo procesador o multiprocesador.

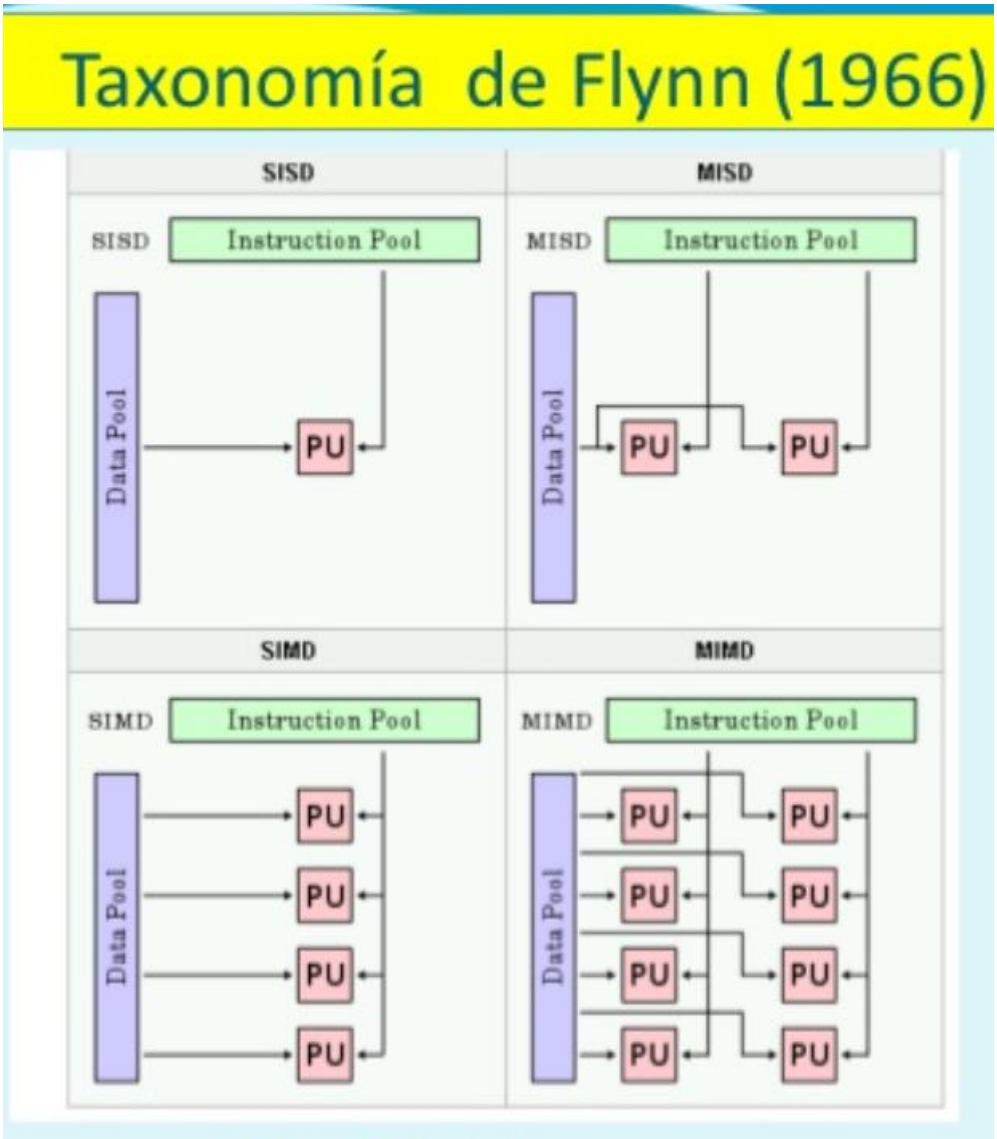


Ilustración 1: Taxonomía de Flynn

Concepto:	La taxonomía de Flynn es una clasificación para las computadoras con arquitectura paralela, basada en el número de instrucciones concurrentes en un instante de procesamiento.

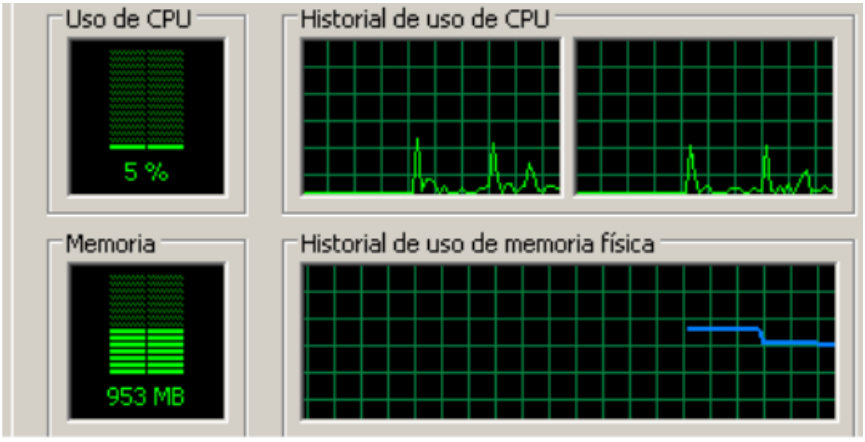


Ilustración 2 Un ejemplo aplicado en Sistemas de Computo

Podemos decir que en los sistemas de cómputo trabajamos con procesos individuales compartidos, lo que quiere decir que la aplicación de código va dirigida tanto a los procesos ejecutados por el CPU, tanto a la memoria física o incluso a la GPU, en computadoras modernas se puede apreciar la arquitectura paralela con una clasificación múltiple autónoma, de esta manera se logra un mejor rendimiento.

2. MULTIPROCESAMIENTO

El multiprocesamiento se refiere a la capacidad de una computadora o sistema informático para utilizar múltiples procesadores o núcleos de procesamiento para llevar a cabo tareas y procesar datos de manera simultánea. Esto contrasta con los sistemas de un solo procesador, donde solo hay un procesador central encargado de ejecutar todas las instrucciones.

El multiprocesamiento puede mejorar significativamente el rendimiento de los sistemas informáticos al permitir la ejecución simultánea de múltiples tareas. Se utiliza en una variedad de entornos, desde estaciones de trabajo y servidores hasta supercomputadoras. Sin embargo, la eficiencia del multiprocesamiento a menudo depende de la naturaleza de las aplicaciones y del grado en que se pueden dividir las tareas para ejecutarse en paralelo.

La programación para sistemas multiprocesador puede ser más compleja que para sistemas de un solo procesador, ya que los desarrolladores deben tener en cuenta la sincronización y la gestión adecuada de los recursos compartidos. Herramientas y tecnologías como OpenMP, MPI (Message Passing Interface) y bibliotecas específicas de multiprocesamiento pueden facilitar el desarrollo de software para entornos multiprocesador.

2.1. Simétrico (SMP)

Procesamiento Simétrico (SMP - Symmetric Multiprocessing):

El multiprocesamiento simétrico se entiende como los sistemas que incluyen más de un procesador. Son capaces de ejecutar diversos procesos de forma simultánea y, además, comparten una misma memoria para el cumplimiento de sus funciones.

Esto quiere decir que esta opción destaca como una de las formas de implementación de los sistemas de multiprocesador, cuya popularidad de uso ha crecido gracias a sus propiedades de aumento de escalabilidad y rendimiento frente a aquellos sistemas que solo usan un único procesador.

Cabe destacar, por tanto, que gran parte de los sistemas multiprocesador en la actualidad emplean la arquitectura de multiprocesamiento simétrico. De esta forma, tenemos el caso de los procesadores con diversos núcleos, donde el SMP se aplica a la totalidad de estos núcleos para gestionarlos como tipos de procesadores separados.

- En un sistema SMP, todos los procesadores comparten igualmente el acceso a la memoria principal y otros recursos del sistema.

- Cada procesador ejecuta su propio sistema operativo y tiene la capacidad de ejecutar cualquier tarea.

- La programación en sistemas SMP a menudo implica dividir tareas entre los procesadores, lo que permite realizar operaciones en paralelo.

2.2. Heterogéneo

El multiprocesamiento heterogéneo es un tipo de arquitectura de computadora que utiliza dos o más procesadores de diferentes capacidades. Esto puede ser útil para aplicaciones que requieren un procesamiento intensivo de datos o una gran cantidad de memoria.

Cómo funciona:

En un multiprocesamiento heterogéneo, los procesadores pueden ser de diferentes tipos, como procesadores de propósito general, procesadores de gráficos o procesadores especializados para aprendizaje automático.

Las tareas se asignan a los procesadores de forma dinámica, en función de las necesidades de la tarea. Por ejemplo, una tarea que requiere un gran volumen de memoria podría asignarse a un procesador con una gran cantidad de memoria.

Ventajas:

- Flexibilidad: Los multiprocesamientos heterogéneos pueden adaptarse a una amplia gama de aplicaciones.
- Rendimiento: Los multiprocesamientos heterogéneos pueden ofrecer un rendimiento superior a los SMP en aplicaciones que requieren un procesamiento intensivo de datos o una gran cantidad de memoria.

Aplicaciones:

- Superordenadores: Los superordenadores suelen utilizar multiprocesamiento heterogéneo para proporcionar un rendimiento extremo.
- Servidores: Los servidores de aplicaciones exigentes, como los servidores de aprendizaje automático, suelen utilizar multiprocesamiento heterogéneo.

2.3. Clúster

El multiprocesamiento en clúster es un tipo de arquitectura de computadora que utiliza múltiples computadoras conectadas en red para trabajar juntas como una sola computadora. Esto puede ser útil para aplicaciones que requieren una gran cantidad de potencia de procesamiento, como las aplicaciones de aprendizaje automático o las simulaciones.

Cómo funciona:

En un clúster, cada computadora tiene su propio procesador, memoria y recursos. Las computadoras están conectadas entre sí a través de una red de alta velocidad.

Las tareas se asignan a las computadoras del clúster de forma dinámica, en función de la disponibilidad y la carga de trabajo. Los sistemas operativos modernos utilizan algoritmos de planificación para asignar las tareas de forma eficiente.

Ventajas:

- Escalabilidad: Los clústeres se pueden escalar fácilmente agregando más computadoras.
- Resistencia: Los clústeres pueden resistir fallos de componentes.
- Rendimiento: Los clústeres pueden ofrecer un rendimiento superior a los SMP y los multiprocesamientos heterogéneos en aplicaciones que requieren una gran cantidad de potencia de procesamiento.

Aplicaciones:

- Superordenadores: Los superordenadores suelen utilizar multiprocesamiento en clúster para proporcionar un rendimiento extremo.
- Simulación: Los clústeres se utilizan para simular fenómenos complejos, como el clima o el comportamiento de los mercados financieros.
- Aprendizaje automático: Los clústeres se utilizan para entrenar modelos de aprendizaje automático complejos.

3. TIPOS DE PARALELISMO

El paralelismo en arquitectura de computadoras se refiere a la capacidad de realizar múltiples operaciones o tareas simultáneamente. Esta característica es esencial para mejorar el rendimiento de los sistemas, ya que permite el aprovechamiento máximo de los recursos disponibles.

Esto se logra mediante el uso de múltiples recursos de procesamiento, como procesadores, memoria y buses.

A continuación, se explorarán más a fondo los distintos tipos de paralelismo, cada uno operando en diferentes niveles dentro de la arquitectura de un sistema computacional.

3.1. A nivel de bit

El paralelismo a nivel de bit implica procesar varios bits simultáneamente. Esto puede manifestarse en operaciones de bajo nivel donde se manipulan los datos a nivel de bits. En arquitecturas especializadas, se implementan unidades de procesamiento que pueden ejecutar operaciones bit a bit en paralelo, acelerando así tareas específicas. Esto se puede lograr mediante el uso de unidades de procesamiento vectorial (GPU) o procesadores de arreglo de datos (FPGA).

Las GPU están diseñadas específicamente para el procesamiento vectorial, que es un tipo de procesamiento que permite que se ejecuten instrucciones en paralelo sobre múltiples elementos de datos. Las GPU se utilizan a menudo en aplicaciones que requieren un procesamiento intensivo de datos, como el procesamiento de imágenes o el aprendizaje automático.

Los FPGA son dispositivos programables que se pueden utilizar para crear procesadores personalizados. Los FPGA se pueden utilizar para implementar paralelismo a nivel de bit mediante la creación de unidades de procesamiento vectorial personalizadas.

Ejemplo de Aplicación: En la criptografía, algoritmos como DES (Data Encryption Standard) pueden beneficiarse del paralelismo a nivel de bit para agilizar el proceso de cifrado y descifrado.

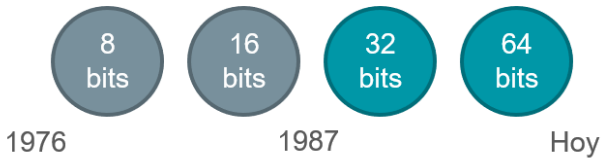


Ilustración 3 Este método está "estancado" desde el establecimiento de las arquitecturas de 32 y 64 bits.

Ejemplo 2: 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.

3.2. A nivel de instrucción

El paralelismo a nivel de instrucción implica ejecutar varias instrucciones simultáneamente. En arquitecturas superscalares, múltiples unidades de ejecución pueden procesar diferentes instrucciones al mismo tiempo, aprovechando la instrucción de flujo de datos y aumentando la eficiencia de la ejecución.

Las instrucciones superescalares permiten que el procesador ejecute más de una instrucción a la vez. Esto se logra mediante el uso de múltiples unidades de ejecución que pueden ejecutar instrucciones diferentes al mismo tiempo.

Las instrucciones especulativas permiten que el procesador ejecute instrucciones que pueden no ser necesarias. Esto se logra mediante el uso de predicción de ramas para predecir el resultado de una rama condicional. Si la predicción es correcta, el procesador puede ejecutar las instrucciones especulativas sin necesidad de volver a ejecutarlas si la predicción es incorrecta.

Ejemplo de Aplicación: Los procesadores modernos utilizan técnicas de ejecución fuera de orden para optimizar la ejecución de instrucciones y mejorar el rendimiento general.

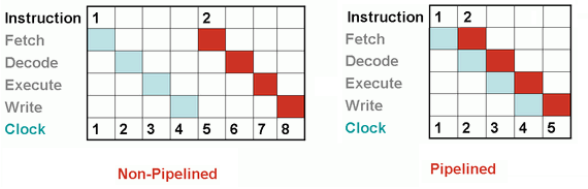


Ilustración 4 En el gráfico anterior se observa el procesamiento de dos instrucciones sin pipeline, tomando un tiempo de 8 ciclos, y con pipeline reduciendo este tiempo a solo 5 ciclos.

Ejemplo 2: Un pipeline de 5 etapas: fetch (buscar la instrucción), decode (decodificarla), execute (ejecutarla), write (escribir en memoria el resultado de la operación).

3.3. De datos

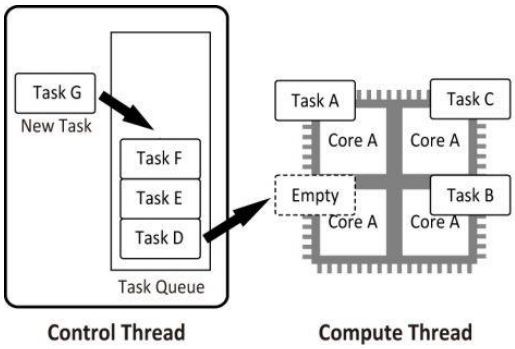
El paralelismo a nivel de datos se centra en realizar operaciones simultáneas en conjuntos de datos. Este enfoque puede implementarse a través de instrucciones SIMD, donde una única instrucción opera en varios datos, o mediante arquitecturas vectoriales que permiten el procesamiento eficiente de conjuntos de datos.

Esto se puede lograr mediante el uso de unidades de procesamiento de datos (FPU) o unidades de procesamiento de gráficos (GPU).

Las FPU están diseñadas específicamente para el cálculo de números de punto flotante. Las FPU se utilizan a menudo en aplicaciones que requieren cálculos de números de punto flotante, como el procesamiento de imágenes o el aprendizaje automático.

Las GPU están diseñadas específicamente para el procesamiento de gráficos. Las GPU se utilizan a menudo en aplicaciones que requieren procesamiento de gráficos, como los juegos o la visualización de imágenes.

Ejemplo de Aplicación: En el procesamiento de gráficos, las operaciones realizadas en píxeles de una imagen pueden ejecutarse simultáneamente, mejorando la velocidad de renderizado.



3.4. De tareas

El paralelismo a nivel de tareas implica dividir una tarea compleja en sub-tareas independientes que pueden ejecutarse simultáneamente. Esto se logra mediante sistemas multiprocesador o mediante técnicas de programación paralela que distribuyen las tareas entre varios núcleos o procesadores.

Esto se puede lograr mediante el uso de múltiples procesadores o múltiples computadoras.

El paralelismo de tareas se utiliza a menudo en aplicaciones que requieren un procesamiento intensivo de datos o una gran cantidad de potencia de procesamiento, como las aplicaciones de aprendizaje automático o las simulaciones.

Ejemplo de Aplicación: En la simulación de procesos complejos, como modelos climáticos, el paralelismo a nivel de tareas puede acelerar significativamente el tiempo de cálculo al distribuir las tareas en múltiples nodos de procesamiento.

Referencias

1. DUNCAN, Ralph. A survey of parallel computer architectures. Computer, 1990, vol. 23, no 2, p. 5-16.
2. JOHNSON, Eric E. Completing an MIMD multiprocessor taxonomy. ACM SIGARCH Computer Architecture News, 1988, vol. 16, no 3, p. 44-47.
3. Aprendiendo bases teóricas del computador: Taxonomía de Flynn — Steemit. (s. f.). Steemit. https://steemit.com/stem-espanol/@greylml/aprendiendo-bases-teoricas-del-computador-taxonomia-de-flynn
4. Team, K. (2023, 28 febrero). ¿Qué es el multiprocesamiento simétrico? KeepCoding Bootcamps. https://keepcoding.io/blog/que-es-multiprocesamiento-simetrico/
5. Multiprocesamiento simétrico. (SMP) Fuente: Tanenbaum, A. S., & Woodhull, A. S. (2016). Sistemas operativos: diseño y aplicación. Pearson Educación. Páginas: 302-305

6. Multiprocesamiento heterogéneo. Fuente: Hennessy, J. L., & Patterson, D. A. (2019). Arquitectura de computadoras: una perspectiva moderna (7a ed.). Pearson Educación. Páginas: 724-727
7. Multiprocesamiento en clúster. Fuente: Stallings, W. (2018). Sistemas operativos: conceptos y aplicaciones (9a ed.). Pearson Educación. Páginas: 582-585
8. Hennessy, J. L., & Patterson, D. A. (2019). Arquitectura de computadoras: una perspectiva moderna (7a ed.). Pearson Educación.
9. Stallings, W. (2018). Sistemas operativos: conceptos y aplicaciones (9a ed.). Pearson Educación.
10. Tanenbaum, A. S., & Woodhull, A. S. (2016). Sistemas operativos: diseño y aplicación. Pearson Educación.