

SISD, SIMD, MISD Y MIMD.

Michael Flynn ha clasificado las arquitecturas de computadoras mediante una variedad de características que incluyen el número de procesadores, número de programas que pueden ser ejecutados por dichos procesadores y la estructura de memoria. La clasificación de Flynn es un buen método para definir la taxonomía de las arquitecturas de computadoras aunque existen algunas otras taxonomías propuestas; tales como las de Dasgupta, Hockney, Skillicorn y Bell. Treleaven, Brownbridge y Hopking, sugieren que las computadoras convencionales pueden ser analizadas bajo dos puntos de vista:

- El mecanismo de control, el cual define el orden de ejecución
- El mecanismo de datos, que define la forma en que los operandos son utilizados

La clasificación de Flynn incluye las siguientes categorías:

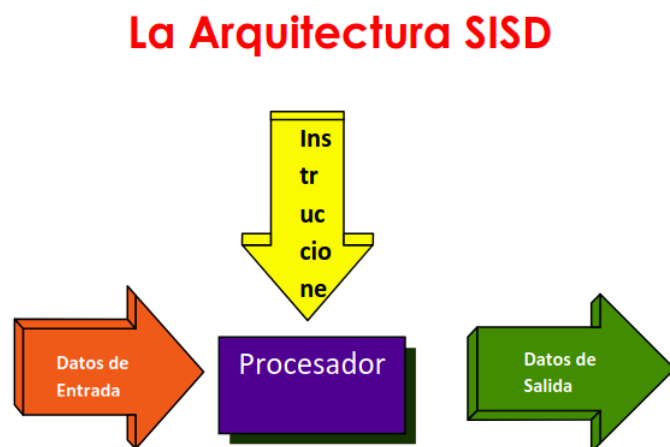
- SISD - Convencional.
- SIMD - datos paralelos, vector computing.
- MISD - arrays sistólicos.
- MIMD - Muy general, múltiples enfoques.

1. SISD (SINGLE INSTRUCTION STREAM, SINGLE DATA STREAM)

Las computadoras SISD tienen un CPU que ejecuta una instrucción a la vez (single, instruction stream) e instancia un ítem de datos a la vez (single data stream)

Todas las computadoras SISD utilizan un registro simple llamado el contador del programa, el cual lleva el conteo de la ejecución serial de las instrucciones.

Como cada instrucción es fetch-eada desde la memoria, el contador del programa es actualizado para direccionar a la siguiente instrucción ha ser fetch-eada y ejecutada.



Definición: Este modelo representa las arquitecturas más tradicionales de computadoras, donde una sola instrucción opera sobre un solo conjunto de datos. Es una arquitectura secuencial en la que solo un flujo de instrucciones y un flujo de datos están activos en un momento dado.

Características:

- Un solo procesador ejecuta una instrucción a la vez.
- Procesa un conjunto de datos a la vez.
- Es el modelo más común en los procesadores tradicionales, como los usados en las PC y las estaciones de trabajo.

Ejemplos de sistemas:

Procesadores secuenciales clásicos como el Intel x86 o ARM en su modo más básico.

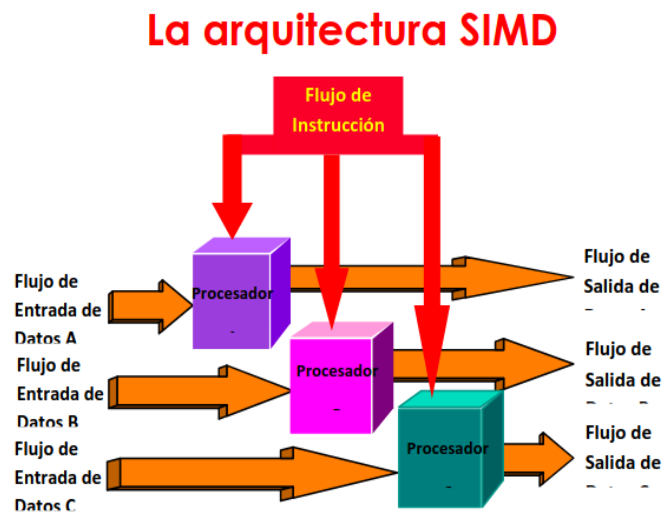
Lenguajes aplicables:

C, C ++, Java, Python y cualquier lenguaje de programación estándar que se ejecute en procesadores secuenciales es adecuado para esta arquitectura.

2. SIMD ((SINGLE INSTRUCTION STREAM, MULTIPLE DATA STREAM)

Las máquinas SIMD tienen una unidad de control que ejecutan un flujo de instrucción simple, pero tienen más de un elemento procesando.

La unidad de control genera las señales de control para todos los elementos que se están procesando, la cual ejecuta la misma operación en diferentes ítems de datos (esto es, multiple data stream).



Definición: En una arquitectura SIMD, una única instrucción se ejecuta de forma simultánea en múltiples conjuntos de datos. Esto permite que el procesador realice operaciones en paralelo, siendo eficiente para aplicaciones que implican grandes volúmenes de datos, como en la computación gráfica o los cálculos científicos.

Características:

- Una única instrucción se aplica a múltiples datos en paralelo.
- Generalmente se utiliza en aplicaciones de procesamiento de vectores o matrices, como las operaciones gráficas o de simulaciones físicas.
- Se encuentra en procesadores con unidades vectoriales o procesadores gráficos (GPUs).

Ejemplos de sistemas:

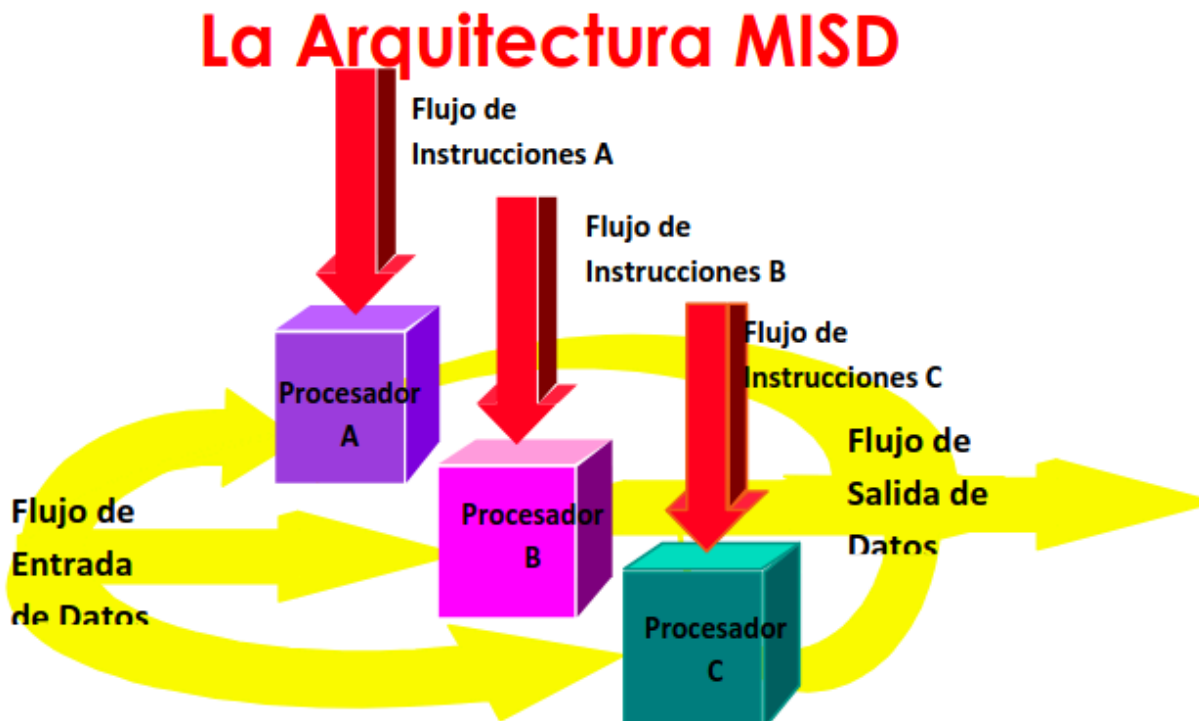
Procesadores con extensiones MMX, SSE, AVX de Intel o NEON en procesadores ARM. Unidades de procesamiento gráfico (GPU) de Nvidia y AMD.

Lenguajes aplicables:

CUDA, OpenCL: Utilizados para programación en GPU.

Fortran, C con extensiones SIMD: Usados en cálculos científicos y gráficos.

3. MISD (MULTIPLE INSTRUCTION STREAM, SINGLE DATA STREAM)



Las máquinas de esta categoría pueden ejecutar varios programas distintos con el mismo ítem de datos. Esto implica que algunas instrucciones son operadas con una sola pieza de datos. La arquitectura puede ser ilustrada en dos categorías:

- a) Una clase de máquinas que requieren de distintas unidades de procesamiento que pueden recibir distintas instrucciones para ser ejecutadas con los mismos datos. Sin embargo, este tipo de arquitecturas es más un ejercicio intelectual que una configuración práctica.
- b) Una clase de máquinas tales que el flujo de datos circula sobre una serie de elementos de procesamiento. Las arquitecturas pipeline tales como los arrays sistólicos entran dentro de este grupo de máquinas.

Las arquitecturas pipeline realizan un vector de procesamiento sobre una serie de etapas, cada una de las cuales lleva a cabo una función particular y produce un resultado inmediato. La razón por la que estas arquitecturas son agrupadas dentro de las máquinas MISD es que los elementos de un vector pueden tener el mismo grupo de datos, y todas las etapas del pipeline representan múltiples instrucciones que estén siendo aplicadas al vector

Definición: En una arquitectura MISD, múltiples instrucciones se aplican a un solo conjunto de datos. Este modelo es raro y poco utilizado en la práctica, ya que es difícil encontrar aplicaciones donde múltiples instrucciones tengan que operar sobre los mismos datos al mismo tiempo.

Características:

- Múltiples instrucciones procesan un único flujo de datos.
- Usado principalmente en sistemas de tolerancia a fallos y en algunos tipos de procesamiento en tiempo real.

Ejemplos de sistemas:

Sistemas de procesamiento redundante utilizados en naves espaciales, donde varias unidades de procesamiento ejecutan distintas operaciones para mejorar la confiabilidad.

Arquitecturas de procesadores como pipelined fault-tolerant systems.

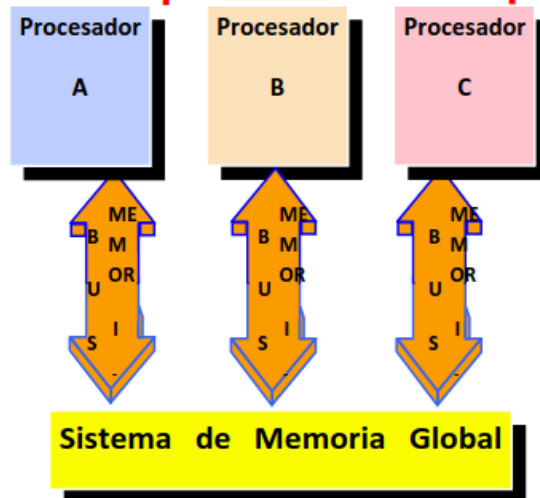
Lenguajes aplicables:

No hay un conjunto claro de lenguajes específicos para MISD, ya que esta arquitectura es muy rara. En algunos sistemas embebidos, el uso de Verilog o VHDL puede ser aplicable para diseñar circuitos personalizados.

4. MIMD (MULTIPLE INSTRUCTION STREAM, MULTIPLE DATA STREAM)

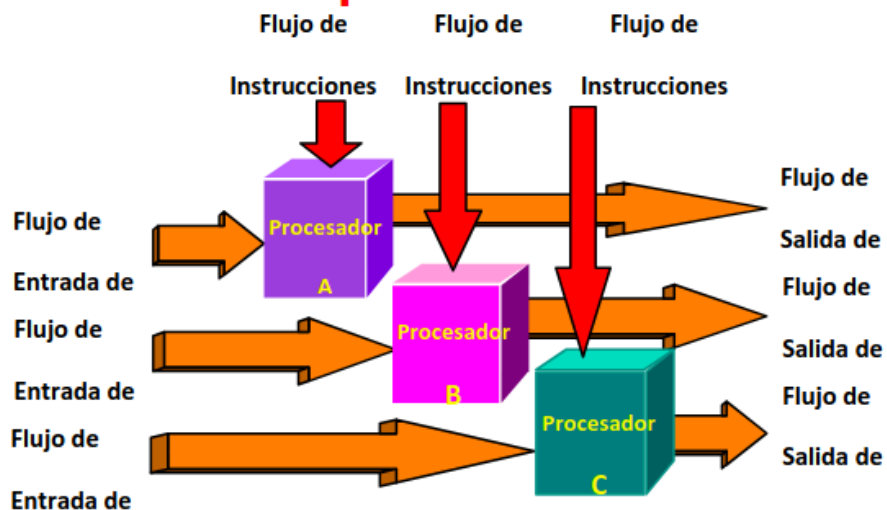
Las máquinas MIMD son llamadas multiprocesadores. Estas tienen más de un procesador y cada uno puede ejecutar un programa diferente (multiple instruction stream) con múltiples flujos de datos. En muchos sistemas MIMD, cada procesador tiene acceso a una memoria global, la cual puede reducir el tiempo de comunicación de los procesadores.

Memoria Compartida en máquinas MIMD



Además cada procesador posee una memoria privada. Muchas de las arquitecturas MIMD son utilizadas para el paralelismo de grano-medio y grano-largo.

La Arquitectura MIMD



En las arquitecturas paralelas MIMD actuales, el número de procesadores es más pequeño que en los sistemas SIMD. Las computadoras MIMD son las más complejas, pero ofrecen grandes promesas para obtener eficiencia acompañada del procesamiento concurrente. Algunas computadoras MIMD comerciales son, la BBN Butterfly, la serie Alliant FX, la serie iPSC de Intel, y la Ultra Computadora de la universidad de New Cork.

Definición: En una arquitectura MIMD, múltiples procesadores ejecutan diferentes instrucciones en múltiples conjuntos de datos. Es uno de los modelos más comunes en los sistemas de multiprocesadores modernos y se usa ampliamente en aplicaciones de computación paralela.

Características:

- Múltiples procesadores ejecutan diferentes flujos de instrucciones de manera simultánea.
- Cada procesador opera sobre diferentes conjuntos de datos.
- Es común en sistemas multinúcleo, clústeres y supercomputadoras.
- Admite operaciones altamente paralelas, lo que lo hace ideal para tareas de procesamiento intensivo como simulaciones, inteligencia artificial y análisis de big data.

Ejemplos de sistemas:

Sistemas multiprocesadores, como Intel Xeon o procesadores AMD Ryzen Threadripper.

Supercomputadoras que utilizan arquitecturas de clústeres o sistemas distribuidos.

Lenguajes aplicables:

MPI (Message Passing Interface): Para paralelismo en clústeres.

OpenMP, CUDA, OpenCL: Para programación en múltiples núcleos o GPUs.

Lenguajes concurrentes como Go, Erlang, y también extensiones en lenguajes como C, C++, que soportan concurrencia y paralelismo.

BIBLIOGRAFÍA

https://www.cs.buap.mx/~rossainz/PCyP_Maestria/1_Apuntes/7_ProgParalela2.pdf