

4. ARQUITECTURAS PARALELAS Y DISTRIBUIDAS

La arquitectura de computadoras es el estudio de la organización e interconexión de los componentes que integran un sistema computacional. La computadora puede ser constituida a partir de la construcción de bloques tales como memorias, unidades aritméticas, elementos de procesamiento y buses. Con estos elementos se pueden conformar un número indeterminado de diferentes tipos de máquinas computacionales, desde las más pequeñas o básicas hasta las llamadas supercomputadoras. El comportamiento funcional de los componentes de las diferentes computadoras son similares unos con otros. Por ejemplo, el sistema de memoria realiza funciones de almacenamiento, la unidad de procesamiento central lleva a cabo operaciones, y las interfaces de entrada y salida transfieren datos desde un procesador a los dispositivos apropiados.

Las mayores diferencias sobre los distintos tipos de computadoras estriban en la forma en que sus módulos son conectados entre sí, las características de rendimiento de dichos módulos, y el método mediante el cual un sistema computacional es controlado por sus operaciones. Los dos elementos principales de un sistema computacional convencional son el procesador y la memoria.

Un procesador manipula datos almacenados en la memoria mediante instrucciones. Las instrucciones son colocadas en los módulos de memoria y el flujo siempre va de la memoria al procesador. El movimiento de datos en un sistema es bidireccional; pues estos pueden ser leídos desde, o escritos hacia los módulos de memoria. La figura 9 representa la interconexión memoria-procesador conocida como el modelo de computación Von Neumann.

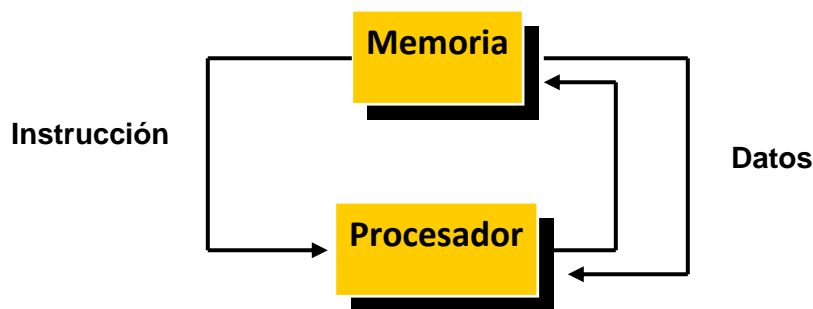


Fig. 9. Interconexión Memoria-Procesador

Una extensión natural del modelo Von Neumann es la llamada Red de Computadoras. En este esquema, cada nodo en la red es una computadora en sí misma, la cual puede ser considerablemente compleja y operar completamente de forma autónoma

respecto a las otras computadoras que integran la red. Además, una red de computadoras puede ser geográficamente distribuida.

En adición a esta extensión natural del modelo Von Neumann, es posible tomar un enfoque más fundamental y diseñar nuevos modelos computacionales exclusivamente para el procesamiento paralelo. El número de instrucciones leídas y datos manipulados simultáneamente por el procesador forman la base para la clasificación de arquitecturas que a continuación de detalla.

4.1. Clasificación de las arquitecturas de computadoras

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 [Roosta99], 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:

1. SISD - Convencional.
2. SIMD - datos paralelos, vector computing.
3. MISD - arrays sistólicos.
4. MIMD – Muy general, múltiples enfoques.

1. SISD (Single Instruction Stream, Single Data Stream)

Las computadoras SISD tienen un CPU que ejecutan una instrucción a la vez (single, instruction stream) e instancia un item de datos a la vez (single data stream). La figura 10 muestra la estructura general de la arquitectura SISD¹.

¹ Las figuras 10 a 15 de esta sección han sido tomadas de <http://www.dgs.monash.edu.au/~rajkumar> y traducidas al español.

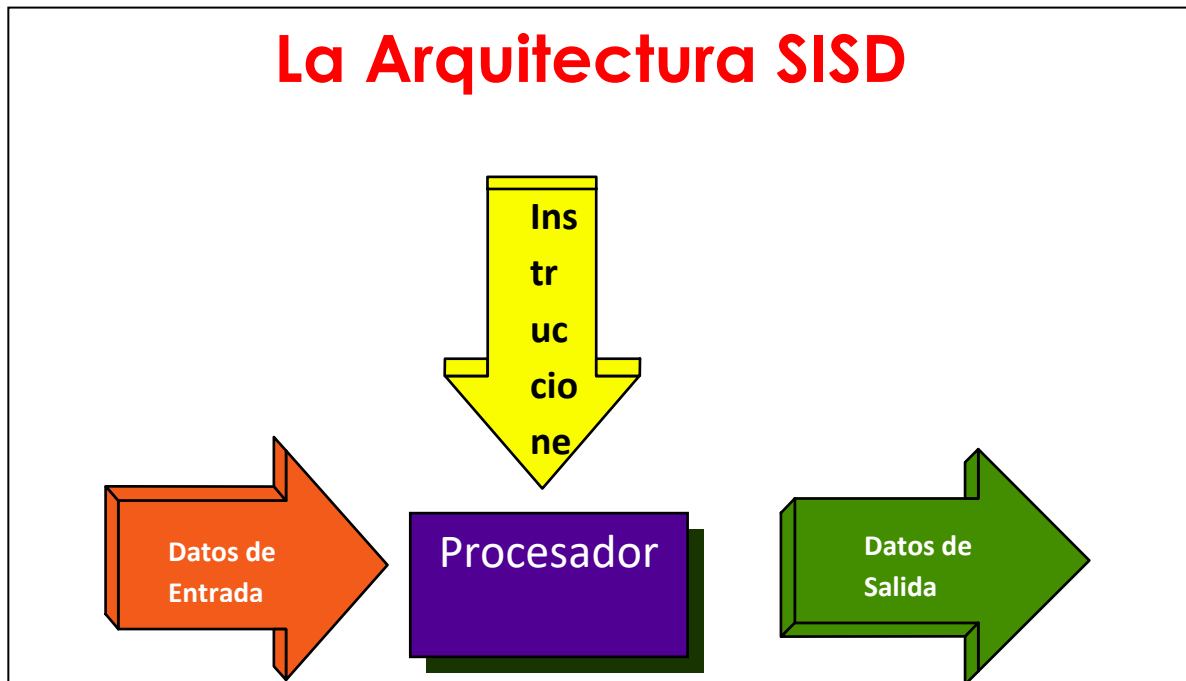


Fig. 10. Modelo de una Arquitectura SISD

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; lo que resulta ser una orden serial de ejecución.

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). En otras palabras, muchos elementos de procesamiento separados son invocados mediante una simple unidad de control. Estas computadoras son utilizadas frecuentemente para problemas que tienen un alto grado de paralelismo de grano pequeño (small-grain). Algunas computadoras SIMD popularmente comerciales son la ILLIAC IV, la DAP o la CM-2 [Roosta99]. Las computadoras SIMD pueden soportar vectores de procesamiento, los cuales pueden ser acompañados por vectores de elementos para procesamiento individual de elementos dentro de cálculos concurrentes. La figura 11 representa una vista general de una arquitectura SIMD.

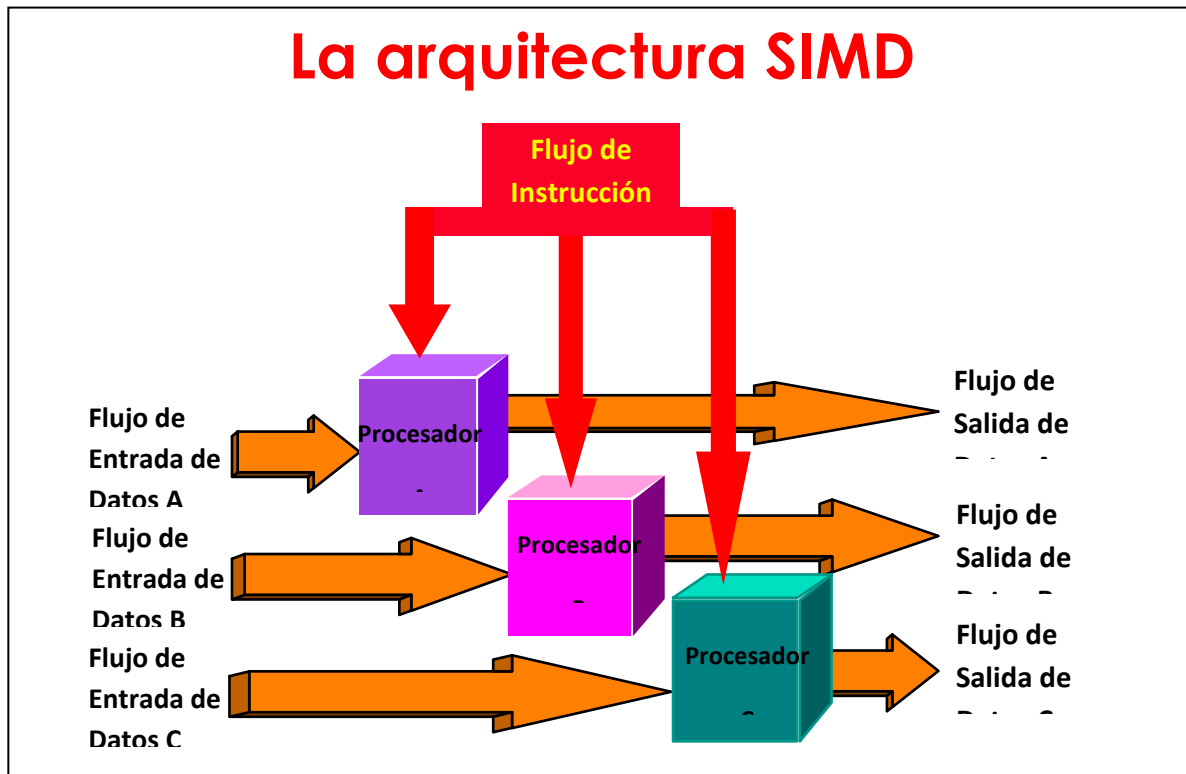


Fig. 11. Modelo de una Arquitectura SIMD

3. MISD (Multiple Instruction Stream, Single Data Stream)

Las máquinas de esta categoría pueden ejecutar varios programas distintos con el mismo item 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 representen múltiples instrucciones que estén siendo aplicadas al vector. La figura 12 representa la estructura general de una arquitectura MISD.

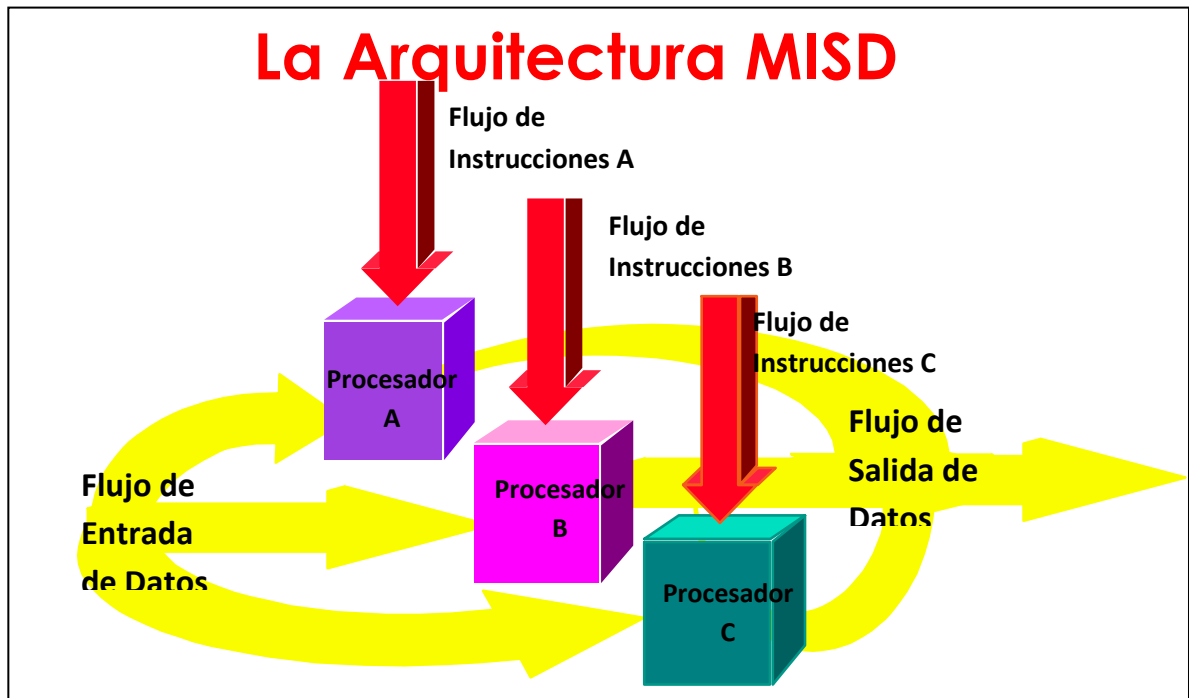


Fig. 12. Modelo de una Arquitectura MISD

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, tal como se ilustra en la figura 13.

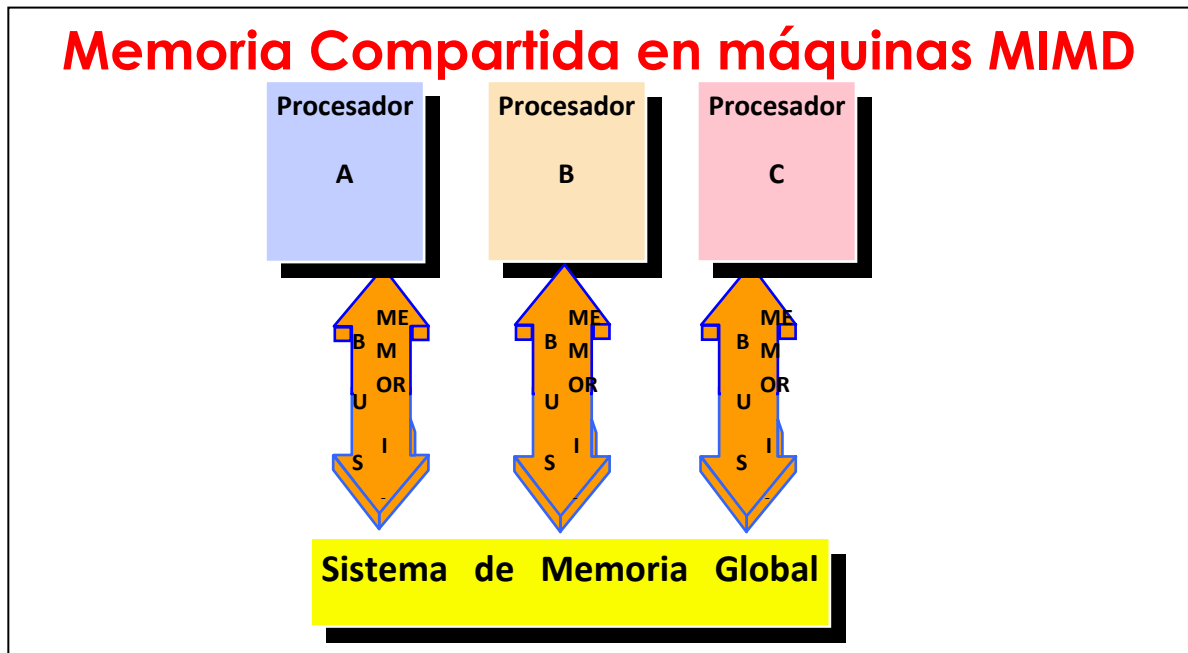


Fig. 13. Modelo de memoria compartida en la arquitectura MIMD

Además cada procesador posee una memoria privada (ver figura 14). Muchas de las arquitecturas MIMD son utilizadas para paralelismo de grano-medio y grano-largo.

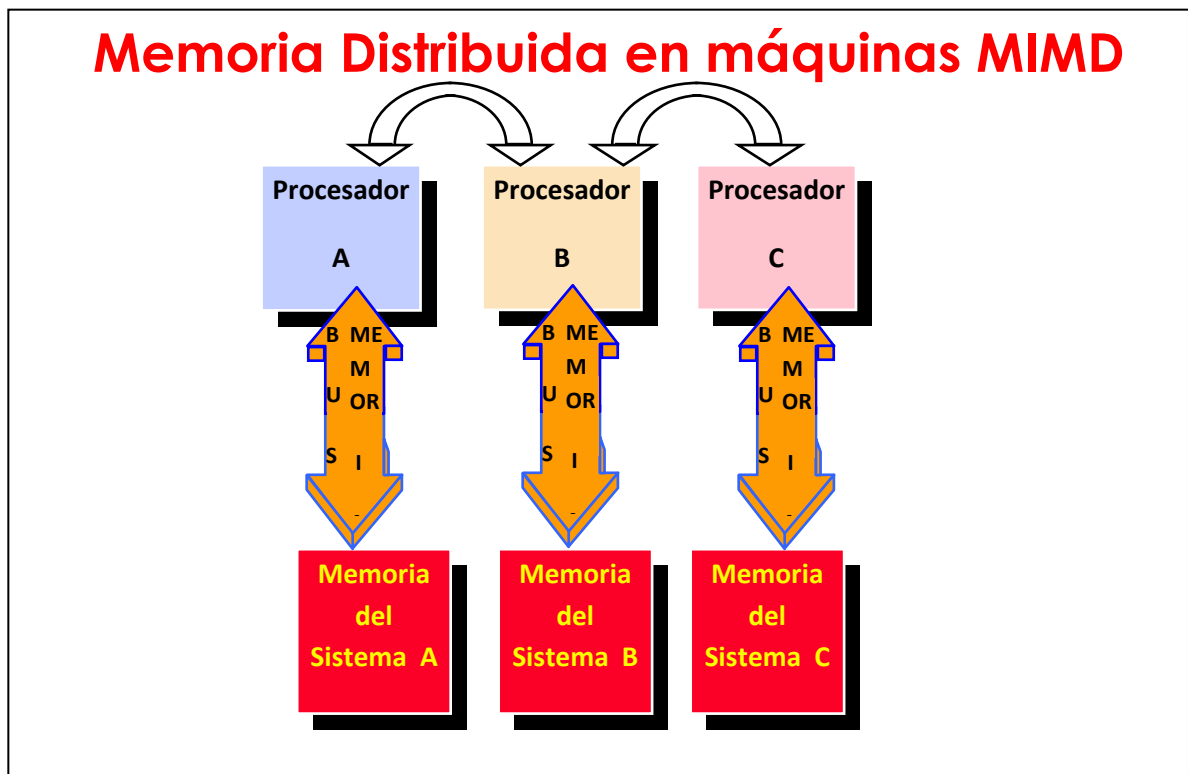


Fig. 14. Modelo de memoria distribuida en la arquitectura MIMD

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 Ultracomputadora de la universidad de New Cork [Roosta99]. La figura 15 representa la estructura general de una arquitectura MIMD.

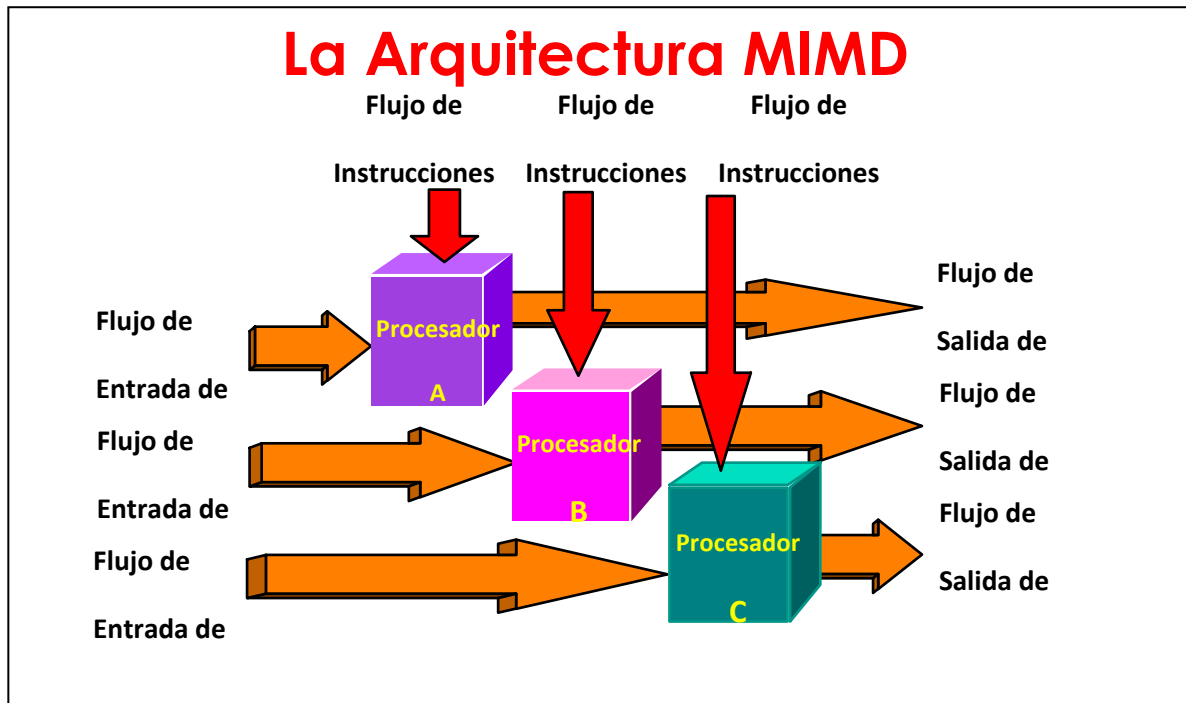


Fig. 15. Modelo de una arquitectura MIMD

4.2. Arquitecturas Paralelas

En la sección anterior, la clasificación de Flynn divide a las arquitecturas paralelas en dos familias importantes: la familia de las computadoras SIMD y la familia de las computadoras MIMD. La figura 16 muestra una taxonomía que representa algunas de las características de las arquitecturas paralelas. En esta sección se hablará entonces de las arquitecturas paralelas SIMD y MIMD.

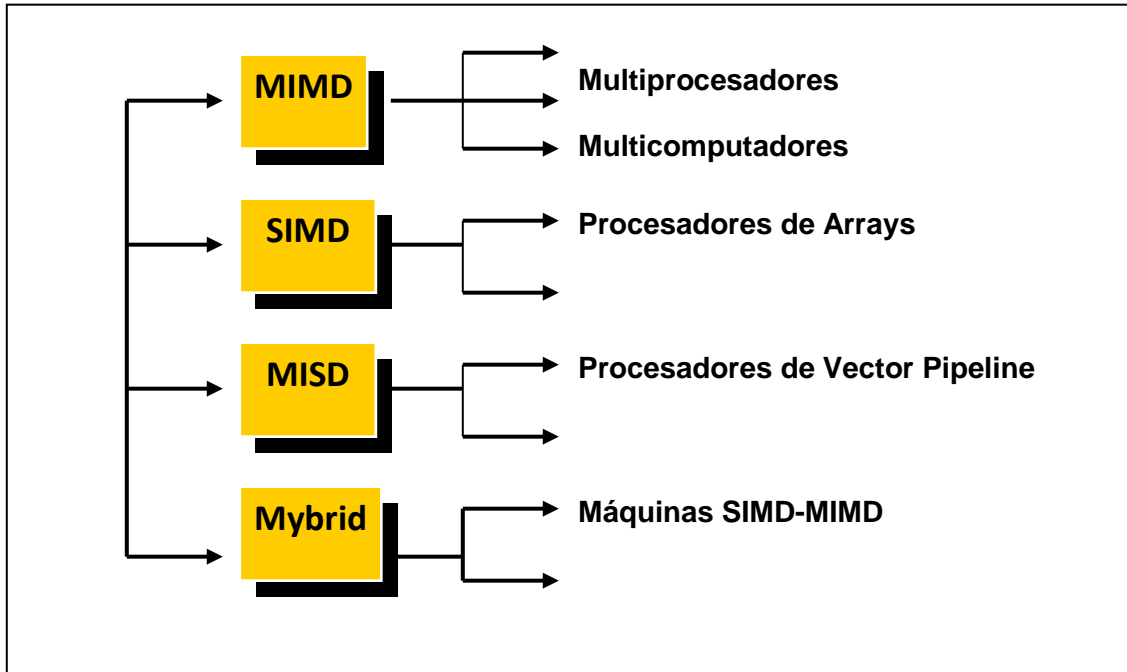


Fig. 16. Taxonomía de las arquitecturas de procesamiento paralelo

Arquitecturas SIMD

En una máquina SIMD, varios elementos procesados se supervisan por una unidad de control. Todas las unidades de procesamiento reciben la misma instrucción desde la unidad de control, pero operan con diferentes conjuntos de datos, los cuales provienen de distintos flujos de datos (ver figura 1.3). Las características principales de este tipo de máquinas paralelas son:

- Distribuyen el procesamiento sobre un larga cantidad de hardware
- Operan concurrentemente con muchos elementos de datos diferentes
- Realizan el mismo cálculo en todos los elementos de datos

Así, cada unidad de procesamiento ejecuta la misma instrucción al mismo tiempo, y los procesadores operan de manera síncrona. El potencial de speedup o aceleración de las máquinas SIMD es proporcional a la cantidad de hardware disponible. El paralelismo hace que las máquinas SIMD desarrollen altas velocidades.

El procesamiento de Arrays fue la primera forma de procesamiento paralelo estudiada e implementada. De aquí se derivan dos tipos diferentes de máquinas:

- Aquellas que realizan operaciones por bits, tales como las máquinas MPP y CM-1
- Aquellas que realizan operaciones por palabras, tales como la ILLIAC IV.

Algunas características generales de las computadoras SIMD comparándolas con las computadoras MIMD son las siguientes:

- Menos Hardware que las MIMD, debido a que utilizan sólo una unidad global de control.
- Menos memoria que las MIMD, ya que sólo se necesita una copia de las instrucciones colocada en la memoria del sistema.
- Flujo de instrucciones simples y sincronización implícita del procesamiento de elementos, lo que hace que una aplicación SIMD sea entendible, fácil de programar y fácil de trazar.
- Instrucciones de control de flujo y operaciones escalares que son comunes a todos los elementos procesados que pueden ser ejecutados en la unidad de control, mientras los procesadores están ejecutando otras instrucciones.
- Necesidad de mecanismos de sincronización sobre los procesadores, después de cada ciclo de ejecución de una instrucción. En contraste con las primitivas de sincronización explícitas que se requieren en la arquitectura MIMD.
- Menos costo ya que sólo se necesita un decodificador de instrucción simple en la unidad de control, en contra de un decodificador en cada elemento que está siendo procesado en una arquitectura MIMD.

Ejemplos de computadoras SIMD incluyen la ILLIAC IV, MPP, DAP, CM-2, MasPar MP-1 y MasPar MP-2 [Roosta99].

Arquitecturas MISD

Un procesador Pipeline es un procesador MISD que trabaja acorde al principio del funcionamiento de un Pipe. La arquitectura pipeline es la forma fundamental de ejecución paralela de un proceso y es una idea poderosa que puede probar de manera significativa el rendimiento de una computadora SIMD. El principio Pipeline implica la segmentación o partición de un proceso computacional. Un proceso puede ser dividido en varios segmentos o etapas (stages). El procesamiento serial es concerniente con la ejecución de todos los stages de un proceso antes de iniciar la ejecución del primer stage del siguiente proceso. Sin embargo en el procesamiento serial, un proceso finaliza completamente antes de iniciar el siguiente proceso. Un procesador puede aumentar su velocidad de ejecución utilizando un pipeline. En un pipeline, mientras un stage está siendo ejecutado, otro stage está siendo cargado y la entrada de un stage se corresponde con la salida del stage previo. La figura 17 muestra los principios básicos de un pipeline para un proceso que se constituye de cuatro stages, en una ejecución serial y paralela. Un resultado del pipeline es que el

procesador realiza diferentes cálculos concurrentemente, pero en cualquier instante de tiempo cada cálculo es llevado a cabo en un stage de ejecución distinto.

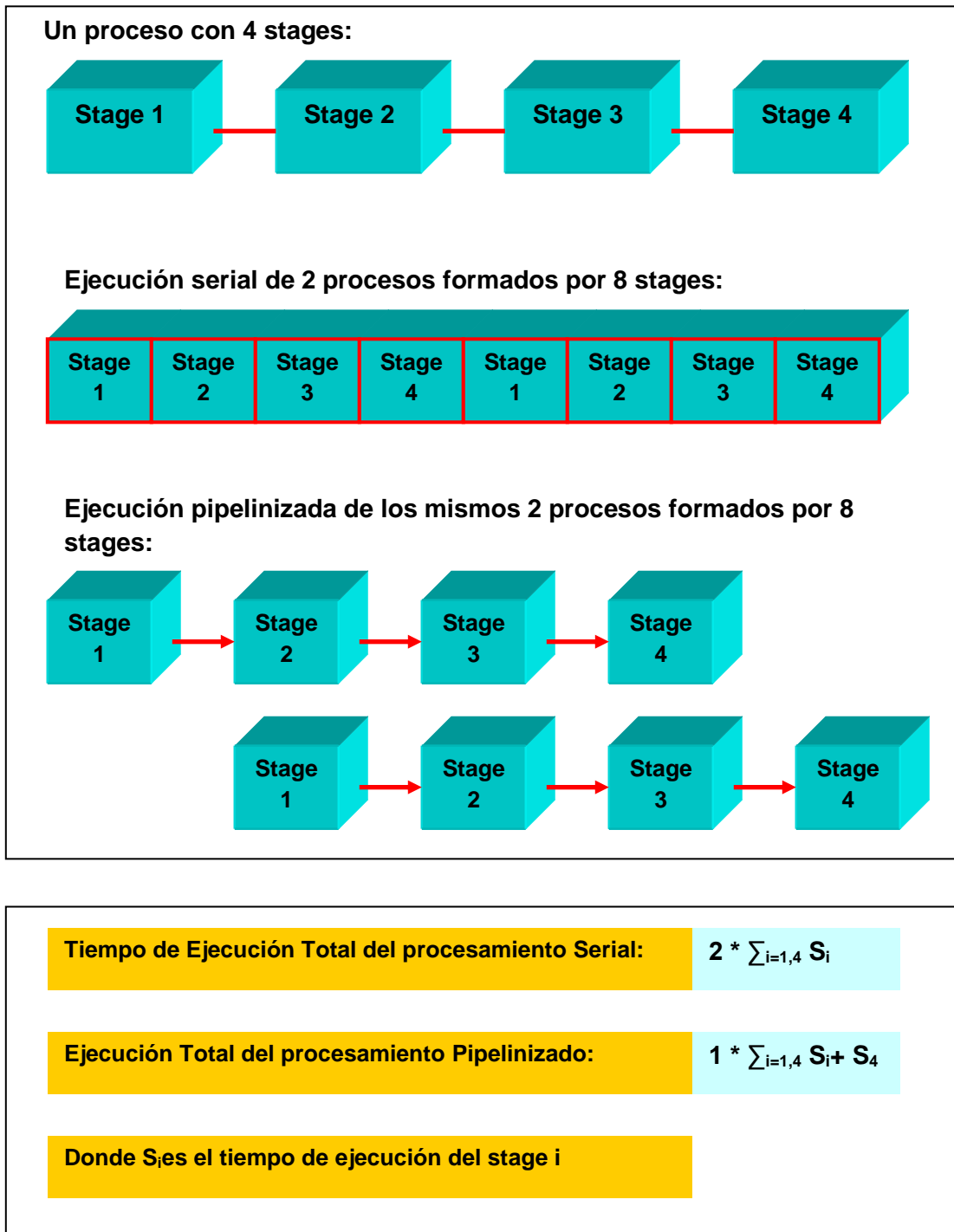


Fig. 17. Ejecución serial y pipelinizada de un proceso formado por 4 stages

Los principios del pipeline pueden ser usados en dos niveles diferentes: unidades aritméticas pipelines y unidades de control pipelines; lo cual nos da dos enfoques de diseño distintos. En general, los principios del pipeline, como la capacidad de ejecutar operaciones de forma simultánea, pueden ser explotados dentro de una arquitectura de computadora en tres niveles:

1. A nivel de instrucción: Es utilizado en el diseño de unidades de procesamiento de instrucciones. Una instrucción pasa sobre un segmento durante cada ciclo, de manera que después las entradas de instrucciones dentro del pipeline, son emitidas en cualquier ciclo.
2. A nivel de Subsistema: Las unidades aritméticas pipeline son un ejemplo de este nivel. Las operaciones pipelineizadas ADD, MUL, DIV y SORT se encuentran en muchas arquitecturas de computadoras.
3. A nivel de Sistema: El segmento pipeline no necesita estar a nivel de hardware, sino que el pipeline puede formar una estructura de software. Esto incluye redes de computadoras especializadas.

Consecuentemente, el pipeline es en principio un concepto para procesamiento de vectores y puede ser hallado en muchas computadoras diseñadas para tales propósitos. Arquitecturas como la CDC STAR 100, Texas Instruments ASC y Cray I [Roosta99], son ejemplos fieles.

Arquitecturas MIMD

Un sistema MIMD es un sistema multiprocesador o una multicomputadora en donde cada procesador individual tiene su unidad de control y ejecuta su propio programa. Las computadoras MIMD tienen las siguientes características:

- Distribuyen el procesamiento sobre un número independiente de procesadores.
- Comparten fuentes, incluyendo el sistema de memoria principal, sobre los procesadores.
- Cada procesador opera independientemente y concurrentemente.
- Cada procesador corre su propio programa.

Esto indica que los sistemas MIMD ejecutan operaciones en paralelo de manera asíncrona; los nodos activos cooperan pero operan independientemente.

Las arquitecturas MIMD difieren con la interconexión de redes, procesadores, técnicas de direccionamiento de memoria, sincronización y estructuras de control. La interconexión de redes hace que los procesadores se comuniquen e interactúen unos con otros. Ejemplos de computadoras MIMD incluyen la Cosmic Cube, nCUBE2, iPSC, Symmetry, FX-8, FX-2800, TC-2000, CM-5, KSR-1 y la Paragon XP/s [Roosta99].

Las computadoras MIMD se pueden categorizar en sistemas fuertemente acoplados y sistemas débilmente acoplados; dependiendo de cómo los procesadores accedan a la memoria.

Los procesadores en un sistema multiprocesador fuertemente acoplado generalmente comparten un sistema de memoria global; estos sistemas son conocidos como sistemas de memoria compartida. Aquellos sistemas MIMD débilmente acoplados pueden compartir un sistema de memoria, pero cada procesador tiene su propia memoria local. A estos sistemas se les conoce como sistemas de paso de mensajes. Las computadoras fuertemente acopladas y débilmente acopladas corresponden a los sistemas MIMD de Memoria Global (GM-MIMD) y MIMD de Memoria Local (LM-MIMD) respectivamente.

Las computadoras MIMD de paso de mensajes se refieren a multicomputadoras en donde cada procesador tiene su propia memoria, llamada memoria local o privada, y es accesible sólo por su propio procesador. Las arquitecturas MIMD de paso de mensajes están referidas tanto a arquitecturas de memoria distribuida como a arquitecturas de memoria privada.

Un sistema MIMD de memoria compartida es llamado Sistema de Acceso Uniforme a Memoria (UMA, Uniform Memory Access), ya que el tiempo de acceso a memoria es el mismo para todos los procesadores que la comparten. En este modelo, los procesadores pueden comunicarse sin restricciones y de una manera simple compartiendo datos utilizando un mismo espacio de direccionamiento. Los datos que se comparten pueden protegerse mediante el uso de métodos de ocultamiento de datos que los modernos lenguajes de programación ofrecen. Debido a esta capacidad de soportar una gran variedad de modelos de programación eficientes, un sistema multiprocesador de memoria compartida es siempre la primera elección de los usuarios de programación paralela.

Esto entra en contraste con los sistemas de paso de mensajes que son más fáciles de diseñar pero más difíciles de implementar o programar. En general, los sistemas MIMD fuertemente acoplados proporcionan mayor rapidez en el intercambio de datos entre los procesadores que los sistemas MIMD débilmente acoplados.

Ejemplos de computadoras GM-MIMD son la serie CDC 6600 y la Cray XM-P. Ejemplos de computadoras LM-MIMD son la Carnegie-Mellon Cm* y la Tandom/16 [Roosta99].