

ALGORITMOS PARALELOS

DÁVILA GUILLÉN GRIMALDO JOSÉ

Universidad Nacional de San Agustín
Escuela profesional de Ciencia de la Computación

29 de marzo de 2017



Arequipa – Perú

Taxonomía de Flynn

En 1966 Michael Flynn propuso un mecanismo de clasificación de las computadoras. El método de Flynn se basa en el número de instrucciones y de la secuencia de datos que la computadora utiliza para procesar información. Puede haber secuencias de instrucciones sencillas o múltiples y secuencias de datos sencillas o múltiples. Esto da lugar a 4 tipos de computadoras, de las cuales solamente dos son aplicables a las computadoras paralelas

		Datos	
		Simple	Múltiple
Instrucciones	Simple	SISD	SIMD
	Múltiple	MISD	MIMD

Figura 1 Esquema de la Taxonomía de Flynn

Una instrucción, un dato (SISD)

En computación, SISD (del inglés Single Instruction, Single Data) es un término que se refiere a una arquitectura computacional en la que un único procesador ejecuta un sólo flujo de instrucciones, para operar sobre datos almacenados en una única memoria. Se corresponde con la arquitectura de Von Neumann.

Según Michael J. Flynn, SISD puede tener características del procesamiento concurrente. La carga de instrucciones y la ejecución segmentada de instrucciones son ejemplos comunes encontrados en las computadoras SISD más modernas.

Características:

- La CPU procesa únicamente una instrucción por cada ciclo de reloj
- Únicamente un dato es procesado en cada ciclo de reloj
- Es el modelo más antiguo de computadora y el más extendido

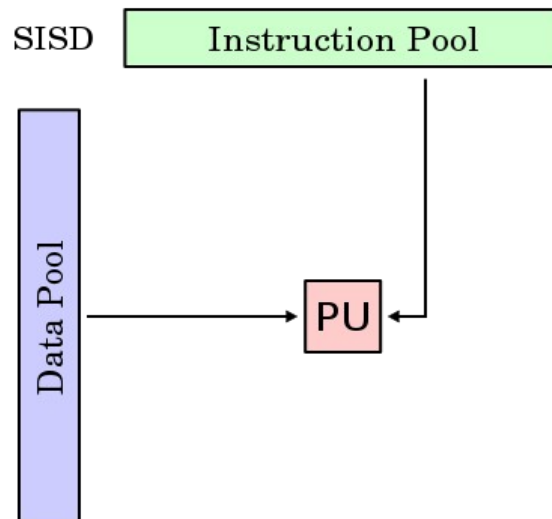


Figura 2 Esquema SISD

Múltiples instrucciones, un dato (MISD):

En computación, MISD (del inglés Multiple Instruction, Single Data, en español: "múltiples instrucciones, un dato") es un tipo de arquitectura computacional (particularmente de computación paralela) 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.

Las máquinas tolerantes a fallos ejecutan la misma instrucción redundantemente para detectar y corregir errores, utilizando task replication, son consideradas de este tipo. No existen muchos ejemplos de esta arquitectura dado que las técnicas más comunes de procesamiento de datos en paralelo suelen ser más apropiadas para MIMD y SIMD. Específicamente, facilitan el escalamiento y el uso de recursos computacionales mejor que MISD.

Características del modelo MISD:

- Cada unidad ejecuta una instrucción distinta
- Cada unidad procesa el mismo dato
- Aplicación muy limitada en la vida real

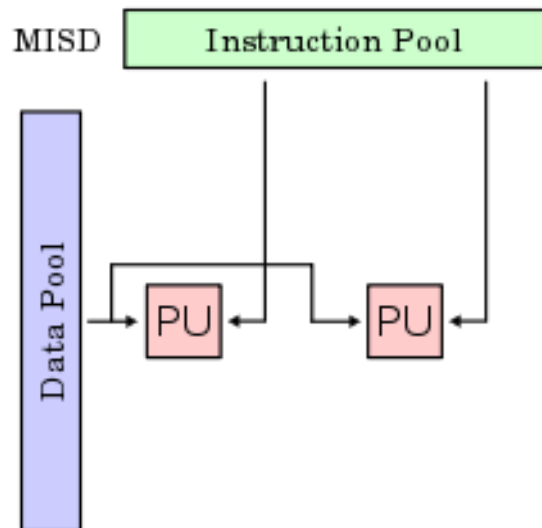


Figura 3 Esquema MISD

Una instrucción, múltiples datos (SIMD)

En computación, SIMD (del inglés Single Instruction, Multiple Data, en español: "una instrucción, múltiples datos") es una técnica empleada para conseguir paralelismo a nivel de datos.

Los repertorios SIMD consisten en instrucciones que aplican una misma operación sobre un conjunto más o menos grande de datos. Es una organización en donde una única unidad de control común despacha las instrucciones a diferentes unidades de procesamiento. Todas éstas reciben la misma instrucción, pero operan sobre diferentes conjuntos de datos. Es decir, la misma instrucción es ejecutada de manera sincronizada por todas las unidades de procesamiento.

Características del modelo SIMD:

- Todas las unidades ejecutan la misma instrucción
- Cada unidad procesa un dato distinto
- Todas las unidades operan simultáneamente

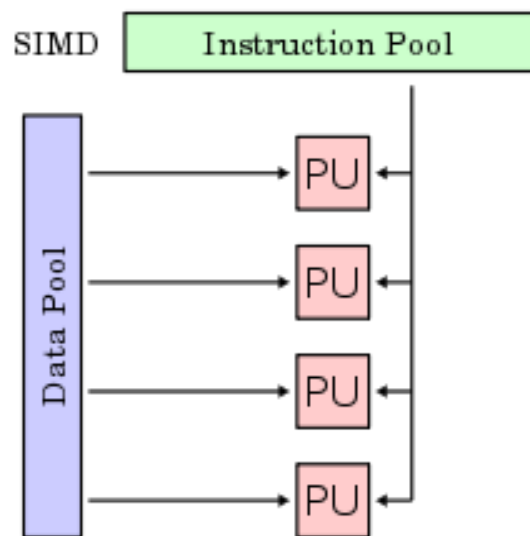


Figura 4 Esquema SIMD

Múltiples instrucciones, múltiples datos (MIMD)

En computación, MIMD (del inglés Multiple Instruction, Multiple Data, en español "múltiples instrucciones, múltiples datos") es una técnica empleada para lograr paralelismo. Las máquinas que usan MIMD tienen un número de procesadores que funcionan de manera asíncrona e independiente. En cualquier momento, cualquier procesador puede ejecutar diferentes instrucciones sobre distintos datos. La arquitectura MIMD pueden utilizarse en una amplia gama de aplicaciones como el diseño asistido, simulación, modelado y en interruptores. Las computadoras MIMD pueden categorizarse por tener memoria compartida o distribuida, clasificación que se basa en cómo el procesador MIMD accede a la memoria. La memoria compartida de las máquinas puede estar basada en buses, extensiones, o de tipo jerárquico. Las máquinas con memoria distribuida pueden tener esquemas de interconexión en hipercubo o malla

Características del modelo MIMD:

- Cada unidad ejecuta una instrucción distinta
- Cada unidad procesa un dato distinto
- Todas las unidades operan simultáneamente

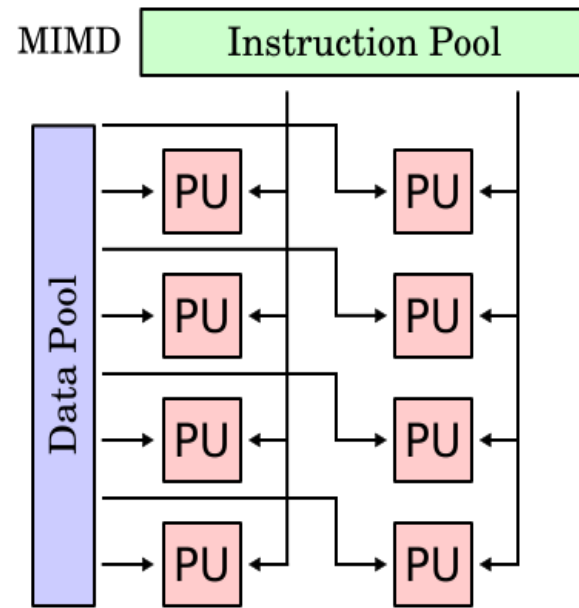


Figura 5 Esquema MIMD

Interconexión de Redes

La interconexión desempeña un papel decisivo en el desempeño de los sistemas de memoria compartida: incluso si los procesadores y la memoria tienen un rendimiento prácticamente ilimitado, una interconexión lenta degradará seriamente el rendimiento general de todos los programas paralelos, excepto el más sencillo.

Aunque algunas de las interconexiones tienen mucho en común, hay suficientes diferencias para que valga la pena tratar las interconexiones para la memoria compartida y la memoria distribuida por separado.

Interconexiones de memoria compartida

Actualmente, las dos interconexiones más utilizadas en sistemas de memoria compartida son los buses y crossbars. Recuerde que un bus es una colección de cables de comunicación paralelos junto con algún hardware que controla el acceso al bus.

Bus

La característica clave de un bus es que los cables de comunicación son compartidos por los dispositivos que están conectados a él. Los autobuses tienen la virtud de bajo costo y flexibilidad; Se pueden conectar múltiples dispositivos a un bus con poco coste adicional. Sin embargo, puesto que los hilos de comunicación son compartidos, a medida que aumenta el número de dispositivos conectados al bus, aumenta la probabilidad de que haya contención para el uso del bus y disminuya el rendimiento esperado del bus. Por lo tanto, si conectamos un gran número de procesadores a un bus, esperamos que los procesadores tengan que esperar frecuentemente el acceso a la memoria principal. Así, a medida que aumenta el tamaño de los sistemas de memoria compartida, los buses se están reemplazando rápidamente por interconexiones de switched.

Switch

Como su nombre indica, las interconexiones switched utilizan switches para controlar el enrutamiento de datos entre los dispositivos conectados.

Se ilustra un crossbar en la figura 6. Las líneas son enlaces de comunicación bidireccionales, los cuadrados son núcleos o módulos de memoria, y los círculos son conmutadores.

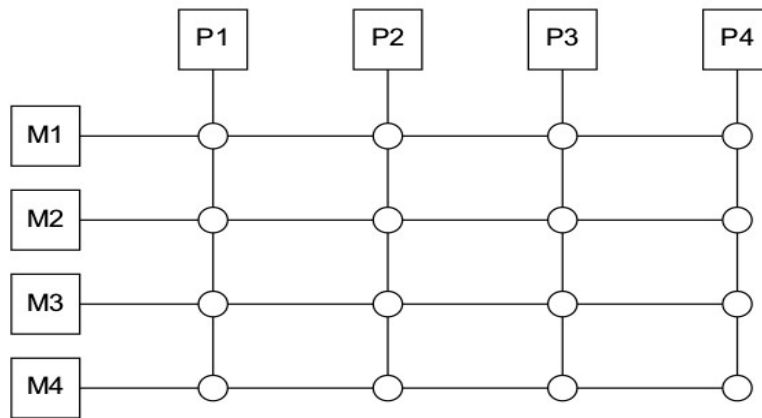


Figura 6 A crossbar switch conectando cuatro preprocesadores.

Los switches individuales pueden asumir una de la dos configuraciones mostradas en Figura 7. Con estos switches y al menos tantos módulos de memoria como procesadores, sólo habrá un conflicto entre dos núcleos que intentan acceder a la memoria si los dos núcleos intentan acceder simultáneamente al mismo módulo de memoria.

Por ejemplo, la Figura 8 muestra la configuración de los interruptores si P 1 escribe en M 4, P 2 lee de M 3, P 3 lee de M 1 y P 4 escribe en M 2.

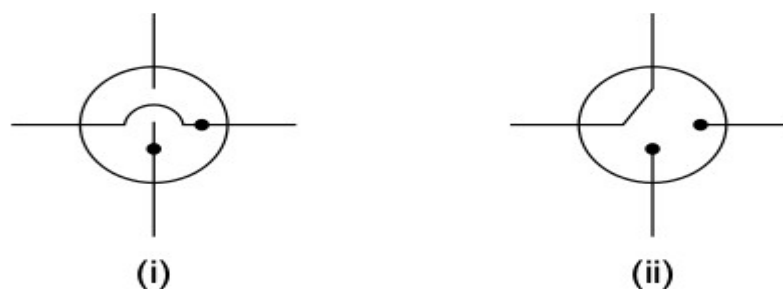


Figura 7 Configuración de los switches internos en un crossbar

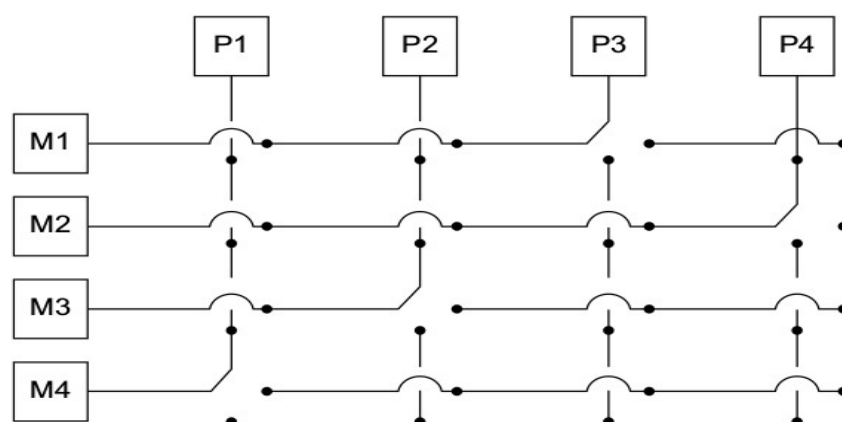


Figura 8 simultaneos accesos de memoria por los microprocesadores.

Crossbars permiten la comunicación simultánea entre diferentes dispositivos, por lo que son mucho más rápidos que los buses. Sin embargo, el costo de los switches y enlaces es relativamente alto. Un pequeño sistema basado en bus será mucho menos costoso que un sistema basado en crossbar del mismo tamaño.