

Estudiante: Henry Yonathan Condori Casa

- 1 Describa de manera teórica los siguientes conceptos: SISD, SIMD, MISD y MIMD. Indique además que lenguajes aplican a estos.**

SISD

En computación, **SISD** (del inglés *Single Instruction, Single Data*, en español: "una instrucción, un dato") es un término que se refiere a una arquitectura computacional en la que un único procesador ejecuta un solo 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.

C, Java, Python (programación secuencial).

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 estas 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. Ejemplos de estos repertorios son 3DNow! de AMD, y SSE de Intel, aunque existen ejemplos más antiguos como el microprocesador Zilog Z80.

C con extensiones vectoriales, CUDA (para GPUs), OpenCL (para procesadores paralelos).

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 de 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.

Algunos argumentan que un array sistólico es un ejemplo de una estructura MISD

No tiene lenguajes de aplicación clara, ya que es más teórico.

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.

MPI (Message Passing Interface), OpenMP (para sistemas multiprocesador), Python con multiprocessing o asyncio.