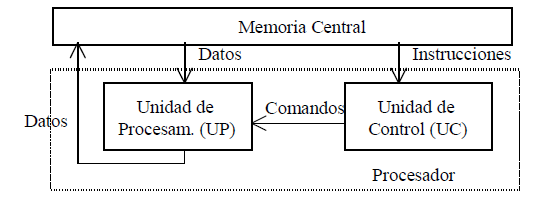
**1 SISD (Single Instruction Stream, Single Data Stream)**

Esta representa la clásica máquina de Von-Neumann, en la cual un único programa es ejecutado usando solamente un conjunto de datos específicos a él. Está compuesto de una memoria central donde se guardan los datos y los programas, y de un procesador (unidad de control y unidad de procesamiento) En este caso, ni = nd =1. En esta

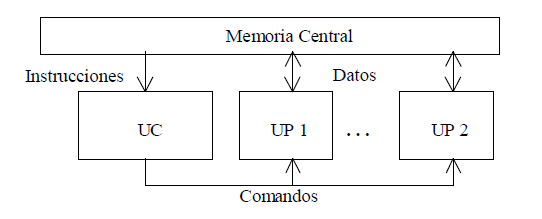
plataforma sólo se puede dar un tipo de paralelismo virtual a través del paradigma de multitareas, en el cual el tiempo del procesador es compartido entre diferentes programas.

Así, más que paralelismo lo que soporta esta plataforma es un tipo de concurrencia. Los PC son un ejemplo de máquinas que siguen este tipo de procesamiento. 

**SIMD (Single Instruction Stream, Multiple Data Stream)**

Arreglo de elementos de procesamiento, todos los cuales ejecutan la misma instrucción al mismo tiempo. En este caso, ni =1 y nd > 1. El enfoque de paralelismo usado aquí se denomina paralelismo de datos. Los arreglos de procesadores son típicos ejemplos de esta clase de arquitectura. En estas arquitecturas, un controlador recibe y decodifica secuencias de instrucciones a ejecutar, para después enviarlas a múltiples procesadores esclavos. El arreglo de procesadores procesa los datos que llegan a los diferentes procesadores, usando la instrucción enviada por el controlador. Los procesadores están conectados a través de una red. Los datos a tratar pueden estar en un espacio de memoria que es común a todos los procesadores o en un espacio de memoria propio a cada unidad de procesamiento Todos los procesadores trabajan con una perfecta sincronización. SIMD hace un uso eficiente de la memoria, y facilita un manejo eficiente del grado de paralelismo. La gran desventaja es el tipo de procesamiento (no es un tipo de procesamiento que aparece frecuentemente), ya que el código debe tener una dependencia

de datos que le permita descomponerse.



**MISD (Multiple Instruction Stream, Single Data Stream)**

Estas son computadoras con elementos de procesamiento, cada uno ejecutando una tarea diferente, de tal forma que todos los datos a procesar deben ser pasados a través de cada elemento de procesamiento para su procesamiento (ver figura 1.4). En este caso, ni > 1 y nd = 1. Implementaciones de esta arquitectura no existen realmente, excepto ciertas realizaciones a nivel de mecanismos de procesamiento tipo encauzamiento (pipelining en ingles) (internamente en los procesadores RISC, etc.) o sistemas de tolerancia a fallas. Muchos autores consideran que esta clase no corresponde a un modo de funcionamiento realista. Otros autores consideran que representan el modo de procesamiento por encauzamiento.

La idea es descomponer las unidades de procesamiento en fases, en donde cada una se encarga de una parte de las operaciones a realizar. De esta manera, parte de los datos pueden ser procesados en la fase 1 mientras otros son procesados en la 2, otros en la tres, y así sucesivamente. El flujo de información es continuo y la velocidad de procesamiento crece con las etapas. Este tipo de clasificación de Flynn puede ser incluida dentro de la clasificación SIMD si se asemeja el efecto de estas máquinas, al tener múltiples cadenas de datos (los flujos) sobre las etapas de procesamiento, aunque también puede ser visto como un modo MIMD, en la medida de que hay varias unidades y flujos de datos y cada etapa está procesando datos diferentes.

**MIMD (Multiple Instruction Stream, Multiple Data Stream)**

Es el modelo más general de paralelismo, y debido a su flexibilidad, una gran variedad de tipos de paralelismo pueden ser explotados. Las ideas básicas son que múltiples tareas heterogéneas puedan ser ejecutadas al mismo tiempo, y que cada procesador opere independientemente con ocasionales sincronizaciones con otros. Está compuesto por un conjunto de elementos de procesamiento donde cada uno realiza una tarea, independiente

o no, con respecto a los otros procesadores. La conectividad entre los elementos no se especifica y usualmente se explota un *paralelismo funcional*. La forma de programación usualmente utilizada es del tipo concurrente, en la cual múltiples tareas, quizás diferentes entre ellas, se pueden ejecutar simultáneamente. En este caso, ni > 1 y nd > 1. Muchos sistemas de multiprocesadores y sistemas de múltiples computadores están en esta categoría. Un computador MIMD es fuertemente acoplado si existe mucha interacción entre los procesadores, de lo contrario es débilmente acoplado. Las MIMD se pueden distinguir entre sistemas con múltiples espacios de direcciones

y sistemas con espacios de direcciones compartidos. En el primer caso, los computadores se comunican explícitamente usando pase de mensajes entre ellos según una arquitectura NUMA (non-uniform memory access). En el segundo caso, al usarse una memoria centralizada, se trabaja con modelos UMA. En el primer caso se habla de MIMD a memoria distribuida y en el otro de MIMD a memoria compartida. Por el problema de

gestión del acceso concurrente a la información compartida, se origina un cierto no

determinismo en las aplicaciones sobre estas plataformas. 