

# Introducción a la computación paralela

*M. en C. Antulio Morgado Valle  
Jefe del Departamento de Electrónica  
del CINTEC-IPN.*

**E**n este artículo se desea presentar al lector una revisión de conceptos básicos usados en el área de computadores en paralelo, así como algunos de los problemas en este campo que actualmente son de interés para la comunidad científica. Para ello, al final del artículo se presenta una revisión bibliográfica al respecto.

---

## Introducción

---

La investigación en computadoras de alto rendimiento incluye varios campos que van desde la arquitectura de la computadora por sí misma hasta los algoritmos y aplicaciones. [1.2.3]. En el campo del diseño de computadoras, en esta década se ha dado más énfasis al desarrollo de arquitecturas de computadoras en paralelo, debido a que éstas tienen el enorme potencial de incrementar sustancialmente la capacidad de cómputo. Esto se debe al hecho de que un número considerable de procesadores trabajando en paralelo reducen en gran medida el tiempo de cómputo, siempre y cuando la computadora tenga la arquitectura adecuada y que los "algoritmos" de los programas sean susceptibles a una división en procesos paralelos.

Actualmente no existen acuerdos en cuanto a la topología ideal para las computadoras con arquitectura en paralelo y es posible que no se llegue a un consenso a corto plazo, ya que las aplicaciones y algoritmos son muy diferentes y diversos. Además de la limitante tecnológica que impide realizar físicamente muchas de las topologías propuestas.

Sin embargo, el diseño de procesadores de alto desempeño suple las limitantes anteriores, y en un futuro próximo veremos cuajados los esfuerzos en esta área que se traducirán en computadoras paralelas cuyos costos sean accesibles a la mayoría de los usuarios que requieran hacer uso de la supercomputación.

---

## Antecedentes

---

M. J. Flynn [4] ha clasificado las arquitecturas para computadoras paralelas en 4 tipos, dependiendo del flujo de datos y/o del flujo de instrucciones:

Nombre	Significado
SISD	Single Instruction Stream Single Data Stream.
SIMD	Un Solo flujo de Instrucciones Un Solo Flujo de Datos. Single Instruction Stream Multiple Data Stream.
MISD	Un Solo flujo de Instrucciones Múltiple Flujo de Datos. Multiple Instruction Stream Single Data Stream.
MIMD	Múltiple flujo de Instrucciones Un Solo Flujo de Datos. Multiple Instruction Stream Multiple Data Stream.

\* En la arquitectura SISD una sola instrucción es procesada o actúa sobre cada dato a la vez; en esta arquitectura están basadas todas las máquinas monoprocesador y computadoras personales que existen en la actualidad (Modelo de Von Newman).

Las instrucciones se ejecutan secuencialmente pero pueden estar solapadas ("pipelined"). Un computador con esta arquitectura contiene varias unidades funcionales, bajo la supervisión de una unidad de control principal.

\* En la arquitectura SIMD la misma instrucción es ejecutada por todos los procesadores a la vez; así cada procesador ejecuta la misma operación sobre diferentes datos (Máquinas de procesamiento matricial).

\* En la arquitectura MISD existen N procesadores, cada uno ejecutando diferentes instrucciones sobre el mismo flujo de datos y sus derivados. Los resultados de un procesador pasan a ser la entrada

del siguiente procesador en el macrocauce. Este es un modelo teórico, ya que no existe ningún desarrollo práctico de esta arquitectura.

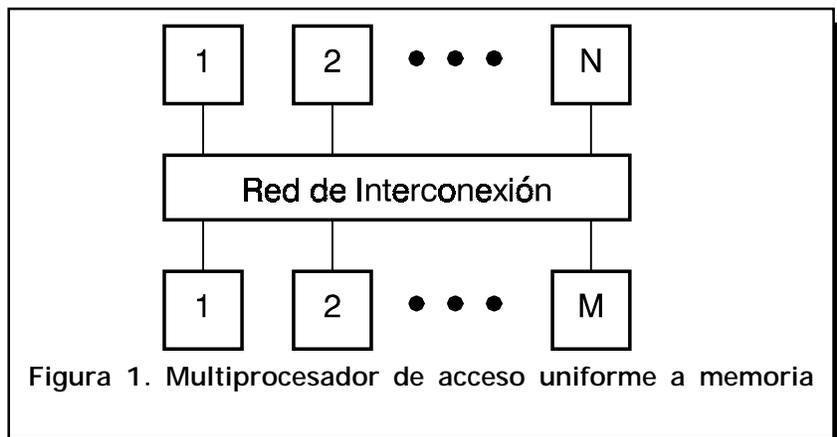
\* La arquitectura MIMD permite ejecutar diferentes instrucciones sobre diferentes conjuntos de datos a la vez. Esta arquitectura es la más flexible de todas, ya que permite el procesamiento de datos que pueden o no estar relacionados entre sí. Cabe aclarar que en esta arquitectura se han basado la mayoría de las computadoras paralelas comerciales que existen en la actualidad.

Una máquina multiprocesadora es aquella en donde una serie de procesadores se interconectan a través de una red con un bloque de memoria global, dividida en pequeñas secciones, lo cual hace aparecer M bloques de memoria contra N procesadores. La comunicación en estos procesadores se efectúa a través de "pase de variables" en "memoria compartida".

Dependiendo de la distribución física de la memoria, los sistemas multiprocesadores se dividen en tres tipos:

\* UMA "Uniform Memory Access" (Acceso Uniforme a Memoria). Este nombre se debe a que los retardos al acceder a cualquier localidad de memoria son similares. Ver **figura 1**.

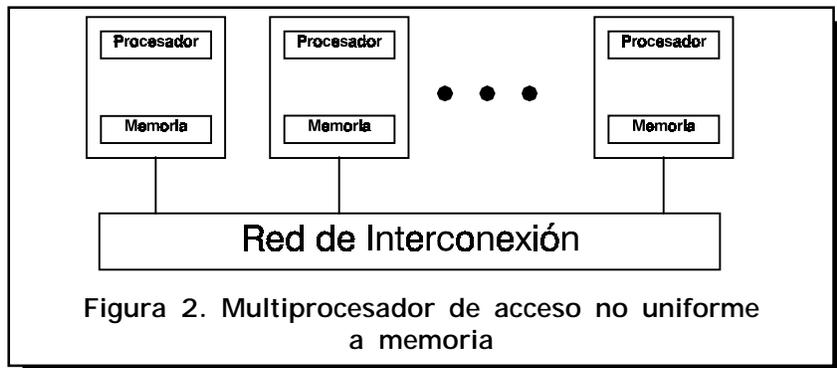
\* NUMA "Non Uniform Memory Access" (Acceso No Uniforme a Memoria). En este modelo el bloque de memoria se coloca en forma local al procesador por lo que cuando un procesador accesa a una memoria remota (que se encuentra asignada a otro procesador) el tiempo de acceso es mayor al tiempo de acceso a su memoria local. Ver **figura 2**.



**Figura 1. Multiprocesador de acceso uniforme a memoria**

\* COMA "Cache Only Memory Access" (Acceso Solo a Memoria Caché). En este modelo la memoria compartida se sustituye por memoria caché, comportándose como el modelo NUMA con una mayor velocidad de acceso. Se emplean directorios distribuidos para localizar direcciones en cachés remotas. Ver **figura 3**.

Un sistema multicomputador consiste de un número de nodos de proceso interconectados a través de una red, donde cada nodo de proceso contiene un procesador, un módulo de memoria local y una interface de interconexión a la red.



**Figura 2. Multiprocesador de acceso no uniforme a memoria**

Todas las comunicaciones entre procesadores se realizan por medio de "envío de mensajes".

El rendimiento de un sistema en paralelo depende en alto grado de la velocidad de comunicación que permita la red de intercomunica-

ción. Por esto la red de interconexión, en su conjunto, constituye uno de los factores más importantes para construir un sistema en paralelo.

Normalmente una red de interconexiones se evalúa tomando en cuenta varios elementos. Entre otros se mencionan la forma de interconexión de los nodos, la configuración misma de éstos y el algoritmo de control de rutas.

La arquitectura de sistemas multiprocesadores cuenta básicamente con tres elementos principales que son: el procesador, la memoria

y una red de interconexión, quedando clasificados los sistemas UMA, NUMA y COMA dentro de estos sistemas.

La arquitectura de sistemas multicomputadores adiciona un elemento más, el ruteador, **figura 4**.

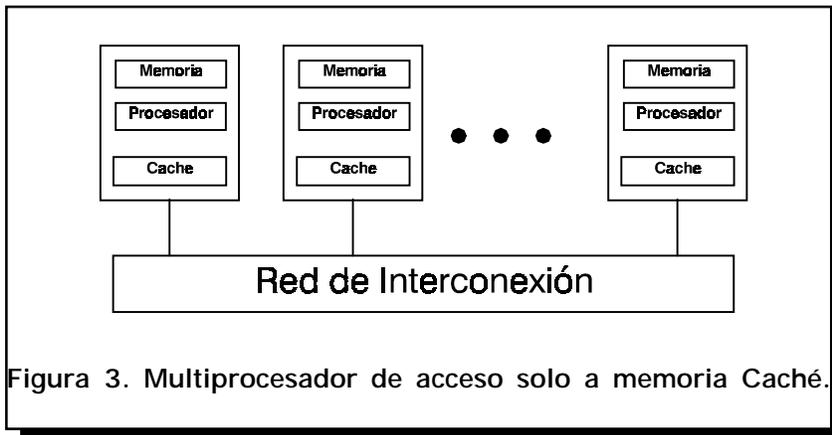


Figura 3. Multiprocesador de acceso solo a memoria Caché.

Dentro de este esquema se tienen dos bloques funcionales: La unidad de proceso ("Process Unit", PU) y la red de interconexión. La PU está constituida por un procesador, memoria local y la unidad de control de rutas ("Control Data Routing", CDR) que incorpora el ruteador y la interface a la red de interconexión. El procesador sigue siendo la unidad clásica compuesta por una unidad aritmética y lógica ("Arithmetic Logic Unit", ALU) y registros. El bloque de memoria consta de un manejador de memoria ("Memory Management Unit", MMU) y unidades de memoria. La unidad de control de rutas CDR consta de un área de memoria ("buffer"), una unidad de control de flujo de información y la interface a la red. También pueden incluirse otras funciones (Dispositivos Periféricos, E/S, etc). Las E/S pueden adicionarse como un subconjunto en una o todas las unidades de proceso.

En esta arquitectura cada procesador tiene asignada una parte de la memoria global. Un procesador no tiene acceso directo a la memoria local de otro procesador o nodo; debido a esto, toda la coordinación y sincronización entre múltiples procesadores en un sistema multiprocesador se efectúa por medio de un protocolo de "pase de mensajes".

El CDR [15], presente sola-

mente en sistemas paralelos, tiene la función de controlar los mensajes que se transmiten entre los procesadores, desde su origen hasta su destino. Un mensaje puede pasar por varios nodos si es que no existe una conexión directa entre el pro-

Normalmente existen dos tipos de CDR con respecto a la topología:

El primer tipo se relaciona con sistemas de topología estática en donde, para cada sistema el CDR ejecuta un solo algoritmo conforme al tipo de topología definida. Ejemplo de esto son los sistemas "TORUS" [5], los sistemas "IPSC/2" [6] y el cubo "COSMIC" [7].

El segundo bloque de CDR se relaciona con sistemas con topología dinámica. Para cada uno de estos sistemas el CDR es capaz de ejecutar varios algoritmos o un algoritmo que se adapta a diferentes topologías. Ejemplo de esto son el

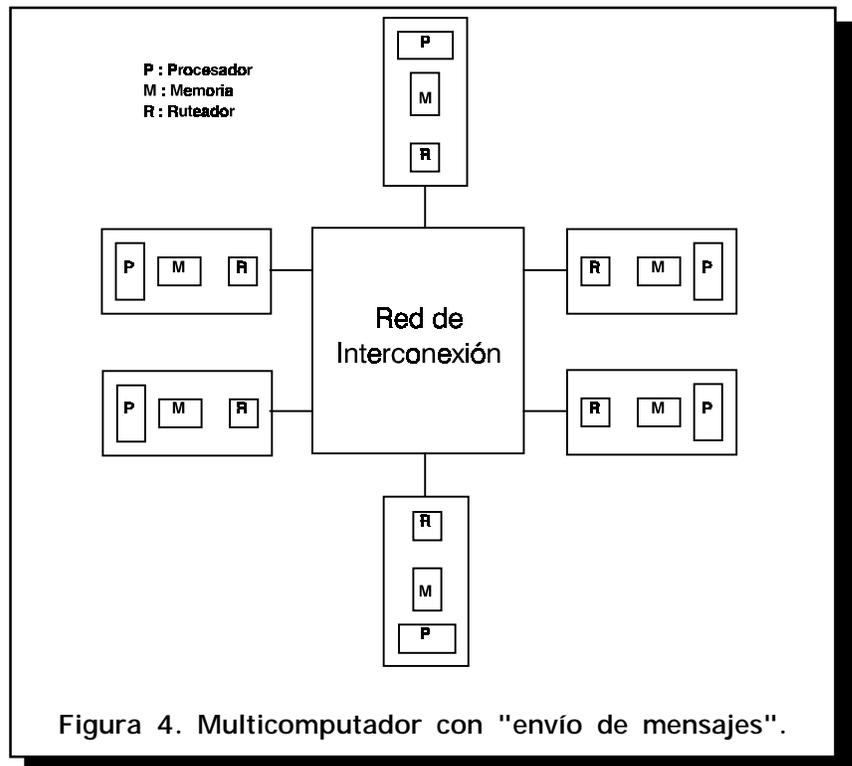


Figura 4. Multicomputador con "envío de mensajes".

cesador que originó el mensaje y el procesador destino. Dependiendo de la topología con que se conecten los procesadores, el C.D.R buscará la o las rutas más idóneas para enviar el mensaje que se esté transmitiendo.

"INTEL/CMUS'S iWARP" [8] o el "INMOS" transputer [9].

Para ambos tipos de CDR existen a su vez dos tipos de protocolos de ejecución. El primero está basado en la tecnología "SOURCE

ROUTING", en donde la decisión de la ruta a tomar pertenece directamente al nodo. En este caso el paquete de información a procesar tiene un encabezado ("Header") que almacena toda la información necesaria y en base a ésta, junto con la topología del sistema y con la condición del nodo procesador, se toma la decisión de la ruta a seguir para llegar al nodo destino.

En el segundo tipo de protocolo la información de los nodos se almacena en una "TABLA" que reside en la memoria global del sistema. Las condiciones particulares de cada nodo en el momento de la ejecución son tomadas en cuenta junto con las condiciones generales de la "TABLA", para determinar el nodo destino.

Los dos tipos de CDR mencionados son comerciales. Ambos tienen ventajas y desventajas, ya sea a nivel protocolo o topológico.

Las desventajas más importantes entre ellos son: los "ruteadores" utilizados en topologías fijas son de uso específico, poco flexibles y se vuelve costoso y de difícil manejo a nivel fabricante, ya que es necesario desarrollar un CDR para cada topología. Los "ruteadores" utilizados en topologías dinámicas son de uso general y difíciles de implementar lo cual eleva más sus costos, presentando poca flexibilidad en ciertas topologías. Aunque en realidad con un solo tipo de CDR es suficiente, siempre y cuando sea lo suficientemente flexible. Entonces el problema se transfiere al control central y a la memoria principal para su manejo a nivel de protocolo.

Como podemos notar existen serios problemas para ambos tipos de "ruteadores". A nivel protocolario, en el "SOURCE ROUTING" es

necesario aumentar la lógica del CDR para permitir que cada paquete de información lleve consigo todo lo necesario. En el segundo, el problema principal reside en el tamaño de la "TABLA", siendo esta proporcional a la cantidad de nodos en el sistema, pudiendo esto ser también una limitante en la cantidad de nodos posibles.

Como se observa, en la actualidad el problema de implementación de cualquier topología paralela radica principalmente en la intercomunicación entre nodos procesadores, y es aquí en donde se esperan aportaciones que ayuden a resolver dichas limitantes.

Una de ellas sería el implementar algoritmos eficientes y compactos para establecer los protocolos de comunicación entre nodos. Otro sería implementar estos algoritmos como parte del CDR y por último encontrar una red de conmutación de alto desempeño que se adapte bien a la mayoría de las topologías propuestas, todo esto bajo las limitantes que impone el uso de tecnología planar en VLSI para la realización de dispositivos semiconductores.

---

### Bibliografía

---

- [1] G.S. Almasi y A.Gottlieb. *"Highly parallel computer"*. The Benjamin Cummins. 1989.
- [2] R.W. Hockney y C.R. Yesshope. *"Parallel computer"*. Adam Hilger LTD. 1983.
- [3] Al Kernek, *"Massively parallel systems - The revolution has begun"*. Super Computing Magazine, pp. 27-28, Otoño 1988.
- [4] M. J. Flynn. *"Some computer organizations and their effectiveness"*. IEEE Transactions on Computers, Vol 21, N° 9, pp. 948-960. September 1972,
- [5] S. Park S. Vassiliadis y J. G. Delgado-Frias. *"Flexible oblivious router architecture"*. IBM Technical Report Trol. C749, IBM Endicott, Nueva York 13760, Sept. 1993.
- [6] W. J. Dally and C. L. Seitz. *"The TORUS routing chip"*. Distributed Computing, 1: 187-196, 1986.
- [7] S. Borkar. *"iWRAP: An integrated solution to High-speed parallel computing"*. In proceeding of Supercomputing Conference, p.330-338 1988.
- [8] C.L. Seitz. *"Cosmic cube"*. Communication of ACM, 28(1): 22-33 1988.
- [9] N. Santoro y R. Khitib. *"Routing without routing tables"*. Technical report SCS-TR-6, School of Computer Science.
- [10] I. Van Leeuwen y R.B. Tan. *"Routing with compact routing tables"*. Technical report RUU-CS-83-16, Department of Computer Science, University of Utrecht, 1983.
- [11] R. Payne and J. G. Delgado-Frias. *"MPU: A Matching-Processing Unit"*. IEEE Int. Conf. on Computing Design, Cambridge, Mass. Oct 1991.
- [12] J. Park, S. Vassiliadis and J. G. Delgado-Frias. *"Packet flow controller whit self*

*compacting buffer*". IBM 1993.

- [13] Harold D. Johnson, José Delgado-Frias, S. Vassiliadis and Douglas M. Green. "A Petri net technique for accessing performance of loosely couple multiproce-

*sor machine architecture*". Int. Journal on microcomputers applications, Vol 11, N° 1 pp 6-40, 1992.

- [14] José Delgado-Frias, S. Vassiliadis, G Pecha-Nek, Harold D. Johnson and Douglas M. Green. "A processing

*unit for flexible multiprocessor machine organizations*". Conf. on Computer Design, October 1992.

- [15] K. Hwang. "Advanced Computer architecture: Parallelism, Scalability, Programmability". Mc Graw-Hill, 1981.