

Una arquitectura para procesamiento paralelo

M. en I. Miguel Lindig Bos
Director del CINTEC-IPN

El procesamiento paralelo posee una innegable importancia en el campo de la computación técnica y científica. El paralelismo limitado (esto es, máquinas de un número reducido de procesadores), constituye la base de prácticamente todas las estaciones de trabajo, así como de muchas máquinas "grandes" ("mainframes") y supercomputadoras. El paralelismo masivo se ha aplicado con eficacia a muchos problemas computacionales específicos. Sin embargo, es válido afirmar que las máquinas paralelas han tenido poco éxito como máquinas de "propósito general", en el sentido de proporcionar una razonable eficiencia computacional para cualquier tipo de problema paralelizable.

En un número previo de esta revista se publicaron una serie de artículos relacionados con procesamiento paralelo. En particular, el artículo "*Procesamiento en Paralelo*" [1] distingue diferentes niveles de paralelismo en función de la **granularidad** del algoritmo. De manera informal, podemos definir este concepto como la relación entre tiempos de ejecución, y el tiempo requerido para el intercambio de información entre procesadores, demandado por el algoritmo. La granularidad es tanto más fina mientras la cantidad y frecuen-

cia de intercambio de información crecen con relación al tiempo de ejecución. Consecuentemente, una granularidad fina impone mayores demandas a la estructura de intercomunicación de la máquina, que una granularidad más gruesa.

Para los propósitos de este trabajo, aceptemos por "procesamiento paralelo" la operación simultánea de una cantidad arbitraria de procesadores con base en información compartida. Esto implica que todos los procesadores involucrados en un proceso deben tener acceso simultáneo a la misma memoria, una condición que no puede ser satisfecha en la práctica. Máquinas paralelas se diseñan con base en procesadores que poseen memoria propia, y mecanismos de intercambio de información entre procesadores. De manera simplista, el problema de diseño de una arquitectura paralela de propósito general consiste, entonces, en determinar aquella geometría de intercomunicación que se adapte al mayor número de problemas computacionales posible, en forma independiente a la granularidad del algoritmo utilizado, y dentro de un marco de restricciones de tipo físico y/o económico preestablecido.

Una máquina de paralelismo limitado que posee gran importancia práctica es el llamado multiprocesador de bus común. Esta arquitectura, tal vez la más frecuente-

mente utilizada en las estaciones de trabajo, satisface el principio de generalidad en el sentido de que no impone restricciones a la geometría de intercomunicación requerida por un algoritmo en particular. Esta característica es de fundamental importancia. Estructuras de interconexión fija, tales como procesadores de arreglo e hipercubos, así como estructuras basadas en redes de conmutación, constituyen soluciones óptimas para ciertas clases de problemas, siendo menos eficientes para otros tipos de algoritmos. Pero, por otra parte, en virtud de establecer esta intercomunicación por medio de un solo bus, la arquitectura es altamente sensible a la granularidad del algoritmo. Esto es, la arquitectura posee un "cuello de botella" intrínseco que limita tanto la velocidad de ejecución, como el número máximo de procesadores con que puede contar la máquina. El presente trabajo pretende dar una respuesta preliminar a las siguientes preguntas:

- 1.- ¿Con una cantidad de líneas físicas de intercomunicación similar al del multiprocesador de bus común, es posible construir una máquina de mayor rendimiento computacional?
- 2.- ¿Dentro de las limitaciones tecnológicas del momento, es posible rebasar el número máximo de procesadores utilizados en arquitecturas de bus común?

Para ello, se presenta un breve resumen del modelo abstracto de la máquina paralela de acceso aleatorio, y se establece la concordancia entre este modelo y los microprocesadores modernos. A continuación se discute el multiprocesador de bus común y se propone una arquitectura alternativa, basada en el multiprocesador de memoria compartida. Esta arquitectura es caracterizada por un modelo matemático basado en el acceso aleatorio, simultáneo de todos los procesadores, a memoria. Finalmente, se discute la factibilidad de construcción de una máquina basada en esta arquitectura y se presentan estimaciones de rendimiento computacional.

El modelo PRAM

La máquina paralela de acceso aleatorio ("Parallel Random Access Machine", PRAM), es un modelo idealizado de multiproceso (figura 1)[2]. En su forma original [3], el modelo consta de una colección infinita de procesadores, cada uno de los cuales tiene acceso a una memoria global, utilizada para comunicación entre procesadores. Cada procesador cuenta con una memoria local en forma de un conjunto ilimitado de registros. El conjunto de instrucciones incluye las transferencias a memoria (tanto local como global) LOAD, STORE y operaciones tales como suma y resta. Registros globales no pueden ser usados como base para direccionamiento indirecto. Tanto la memoria local, como la global, tienen capacidad para almacenar enteros de longitud arbitraria.

El modelo descrito en la referencia [3] incluye dispositivos de entrada/salida, así como las instrucciones correspondientes. Todos los procesadores ejecutan un mismo

programa, que forma parte de los procesadores. Esto es, existen tantos modelos PRAM como programas. El modelo PRAM respeta el principio de **costo unitario**, en el sentido de que el tiempo de ejecución de cada una de las instrucciones definidas para el modelo es el mismo. Lo anterior incluye accesos a memoria y operaciones de entrada/salida. Se han propuesto una serie de modificaciones al modelo original. En su forma más aceptada, el modelo PRAM está basado en:

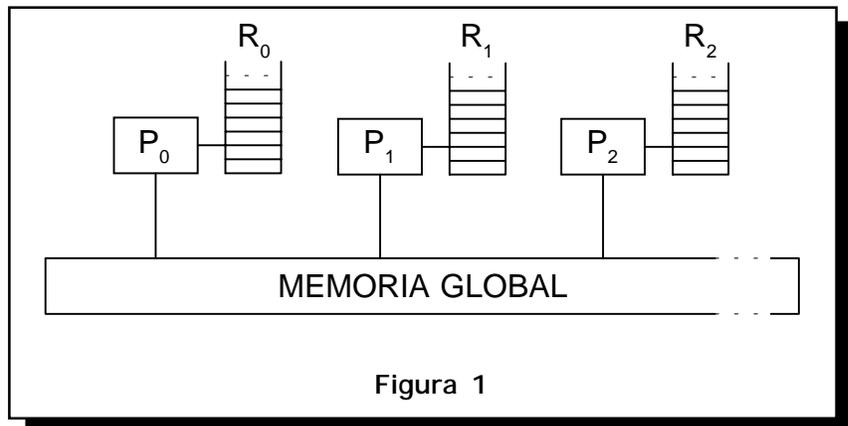
- 1.- Un número de identificación, **PIN** (processor identification number), único para cada procesador. Cada procesador puede leer su PIN en un registro de lectura dedicado.
- 2.- Todos los procesadores ejecutan la misma instrucción. Si esta instrucción es función de un PIN en particular, aquellos procesadores que no poseen este PIN ejecutan lazos de sincronización.

modelo en función a que la lectura y/o escritura concurrente (por más de un procesador) a la misma localidad de memoria global es permitida, o no.

El microprocesador como elemento de procesamiento

Definamos como procesador el conjunto formado por microprocesador, controlador de memoria *cache* y dispositivos de memoria *cache* (figura 2).

Como elemento de procesamiento de una PRAM, este procesador ejecuta un superconjunto del juego de instrucciones definido como completo. Por otra parte, la memoria *cache* puede, en principio, desempeñar las funciones de la memoria local, si se omite el requisito de tamaño ilimitado. La condición relativa al tamaño arbitrario de los datos no es satisfecha. Sin embargo, dispositivos modernos aceptan formatos de hasta 64 bits. En consecuencia, para la gran mayoría de los problemas compu-



Desde el punto de vista del conjunto de instrucciones, en la actualidad se acepta un modelo que incluye la multiplicación, división y desplazamientos de datos sobre un número de bits arbitrario. Por otra parte, existen subdivisiones de este

tacionales, esta limitación no es relevante.

En materia de tiempos de ejecución, el principio de costo unitario puede considerarse aplicable (particularmente en procesadores tipo

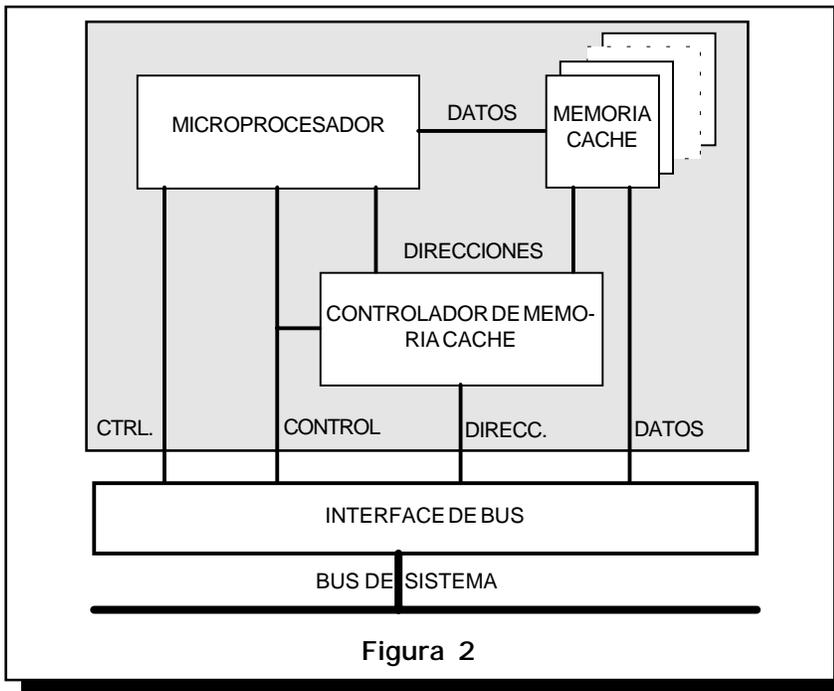


Figura 2

RISC), con la posible excepción de las instrucciones de desplazamiento, multiplicación y división. El principio de costo unitario no es aplicable a la memoria global. No solamente es más lenta que la memoria caché por razones de costo, sino por la mayor separación física entre este subsistema y los procesadores. Esta separación requiere dis-

positivos de transmisión/recepción, cuyos retardos de propagación son significativos en términos de tiempos de ciclo de los procesadores. De este hecho se desprende una consecuencia importante. La necesidad de intercomunicación de datos entre procesadores involucra procesos que son más lentos que los accesos a memoria caché. En

consecuencia, un procesador en una arquitectura paralela tiene un rendimiento individual menor, comparado con su rendimiento en un esquema uniprocador, independientemente de la arquitectura de que se trate.

El multiprocesador de bus común

Esta arquitectura es, tal vez, la más sencilla desde el punto de vista conceptual (figura 3). La comunicación entre procesadores y memoria global se efectúa por medio de un solo bus (de aquí el nombre de la arquitectura). El número de accesos a memoria global se reduce a un mínimo, permitiendo la comunicación directa entre procesadores. Esta comunicación se establece mediante un segundo bus, el llamado **bus de indagación** ("snoop bus") [4].

En rigor, el multiprocesador de bus común posee características tanto del modelo PRAM, como del modelo de interconexión fija. Sin

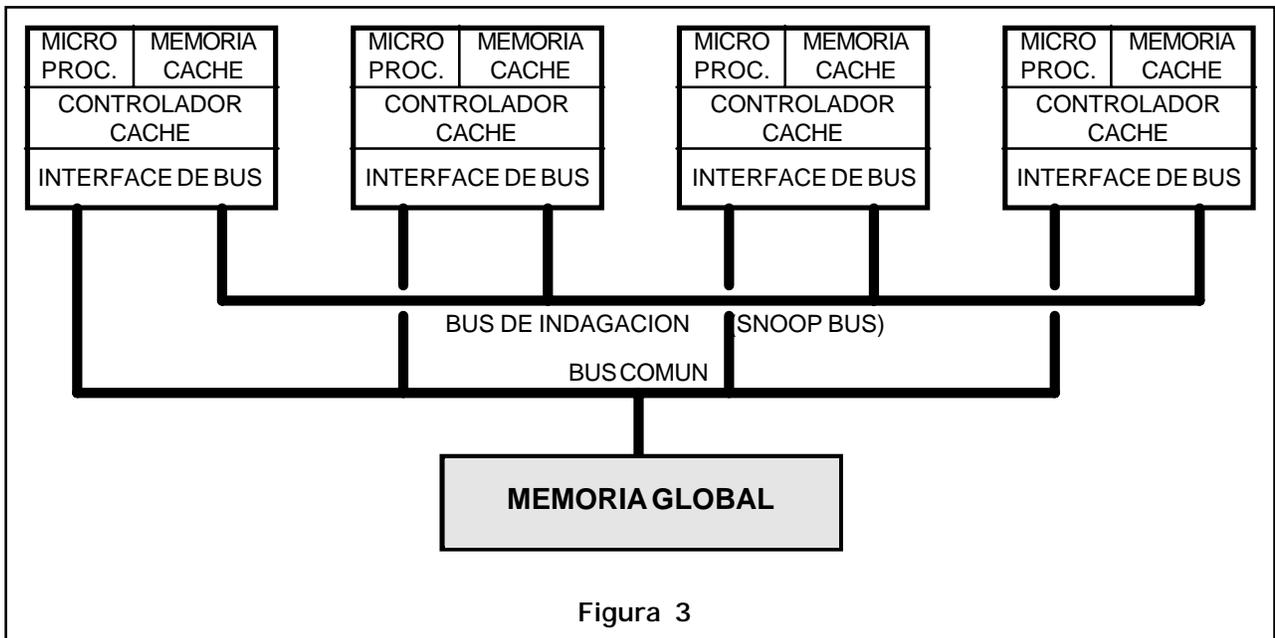


Figura 3

embargo, es posible visualizar esta arquitectura como una PRAM, en virtud de que la función del bus de indagación consiste en simular una memoria global más rápida que, la de hecho, existente. En efecto, la comunicación directa entre procesadores puede emular la función de la escritura de actualización, sin afectar a la memoria global.

La comunicación directa entre procesadores puede efectuarse, en principio, con un costo unitario, si se garantiza la operación sincrónica de los procesadores. Sin embargo, el uso de un bus común obliga a la implantación de un protocolo, para evitar la utilización simultánea del bus por más de una pareja de procesadores. Cualquiera que sea este tiempo, es evidente que la cantidad total de tiempo requerida crece linealmente con el número de intercomunicaciones que deben establecerse en un momento dado. Se puede probar que, para una granularidad dada, la máquina se "satura" después de un cierto número de procesadores, esto es, el rendimiento computacional ya no crece al aumentar el número de procesadores.

La diferencia fundamental entre el modelo PRAM y el multiprocesador de bus común consiste en el mecanismo de intercomunicación. Para la PRAM el medio de comunicación es la memoria global. La intercomunicación entre dos procesadores requiere dos accesos a memoria, uno de escritura por un procesador dado, seguido de uno de lectura por otro procesador. Para el multiprocesador de bus común, el medio de comunicación es el bus de indagación. Un procesador dado escribe a su memoria local (memoria caché), y la lectura por parte de otro procesador es efectuada a esta misma memoria local, en forma "transparente" a la

operación del procesador que efectuó la escritura. Este proceso es, desde luego, más rápido que la comunicación por medio de la memoria global. Nótese que lo anterior es aplicable a cualquier esquema de interconexión fijo.

El multiprocesador de memoria compartida

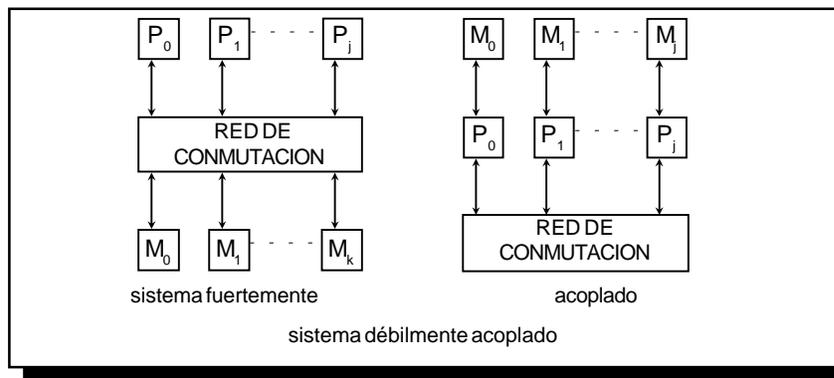
El término **multiprocesador** abarca todas las arquitecturas posibles, integradas por varios procesadores y una, o más, unidades de memoria. Desde el punto de vista de comunicación de datos entre procesadores, se distingue entre dos modelos, el de una memoria global, compartida, y el de memoria distribuida.

En el modelo de memoria distribuida, la memoria global es dividida en tantos módulos como procesadores existan en la estructura. Cada procesador posee un módulo, y el intercambio de información se produce a través de los procesadores.

ye un solo módulo. En general, esta memoria puede ser dividida en M módulos. Cada uno de N procesadores es unido mediante un bus de datos a una red de conmutación. Lo mismo se cumple para los M módulos de memoria, esto es, la red se encuentra entre procesador y memoria (modelo fuertemente acoplado) [5]. Nótese que N puede ser diferente a M .

Con este esquema, la cantidad de accesos simultáneos posibles a memoria aumenta proporcionalmente al número de módulos de memoria. Sin embargo, existe la posibilidad de que se generen colisiones, esto es, el intento de dos o más procesadores por acceder al mismo módulo. Tal vez por esta razón, el modelo de memoria compartida ha recibido poca atención en la literatura reciente. Aquí se analiza este modelo, por lo siguiente:

- 1.- El modelo de memoria compartida es el más cercano al modelo abstracto. Todos los



De acuerdo con [5], se trata de un multiprocesador débilmente acoplado (figura 4).

En el modelo de memoria compartida, los procesadores tienen acceso a toda la memoria global. En el caso del multiprocesador de bus común, esta memoria constitu-

modelos de memoria distribuida pueden ser mapeados en este modelo, por lo que es, intrínsecamente, el de mayor generalidad.

- 2.- El modelo de memoria compartida no considera la existencia de memorias locales. Esto no corresponde a la definición

de procesador para una PRAM. Bajo ciertas consideraciones, la memoria caché puede operar como memoria local, por lo que conviene re-examinar el modelo.

3.- Un multiprocesador de memoria compartida puede constituir un grupo ("cluster") de procesadores, en una arquitectura híbrida de memoria distribuida [6]. Por lo tanto, sus características definen las características globales de la arquitectura.

La discusión subsecuente se limita al caso de un número de módulos de memoria igual al de procesadores. También, consideramos una memoria de (NL) localidades, dividida en N módulos iguales, donde cada localidad almacena el número de bits correspondiente al formato de datos.

En el Centro de Investigación Tecnológica en Computación (CINTEC) del Instituto Politécnico Nacional se está evaluando la factibilidad de construcción de una máquina con el siguiente cambio en la estructura descrita. La red de conmutación es substituida por un con-

junto de buses, tal que cada procesador está conectado a un bus propio. Estos buses son comunes a los módulos de memoria, esto es, cada módulo se conecta al conjunto de buses, por medio de dispositivos de tercer estado (figura 5).

Este esquema permite la interconexión de cualquier procesador a todos los módulos de memoria, así como establecer tantas comunicaciones simultáneas, como procesadores existentes en el esquema. Llamaremos a este modelo **multi-procesador de memoria compartida de buses múltiples (MCBM)**.

Con relación al multiprocesador de bus común, parece evidente que este modelo posee un rendimiento relativo mayor. Pero, en virtud de que el modelo no excluye la posibilidad de colisiones, también es evidente que no puede ser tan eficiente como el modelo teórico de la PRAM. Por otra parte, accesos simultáneos deben ser atendidos secuencialmente, esto es, el esquema requiere arbitraje.

La probabilidad de que ocurra una colisión en un acceso dado depende de la naturaleza del algo-

ritmo, esto es, del tipo de simetría inherente a la gráfica del problema. Un caso límite puede constituir el acceso aleatorio, esto es:

- 1.- La probabilidad asociada a la dirección A_k es la misma para toda k.
- 2.- La ocurrencia de la dirección A_j es independiente de A_k .

Supóngase que el tiempo de acceso sea la unidad. El acceso simultáneo de dos procesadores a un mismo módulo debe efectuarse secuencialmente, esto es, requiere 2 unidades, y así sucesivamente. Para N procesadores, la cantidad total de intercambios posibles de información es N^N . Durante un conjunto de accesos dado, se pueden producir colisiones a más de un módulo de memoria. Sea n_j el número de colisiones de j procesadores, donde j corresponde al conjunto de más elementos. Entonces, el tiempo medio de acceso, τ_m , está dado por:

$$\tau_m = (1n_1 + 2n_2 + \dots + Nn_N) / N^N$$

..... 1

Se puede probar que el número de accesos que involucran n colisio-

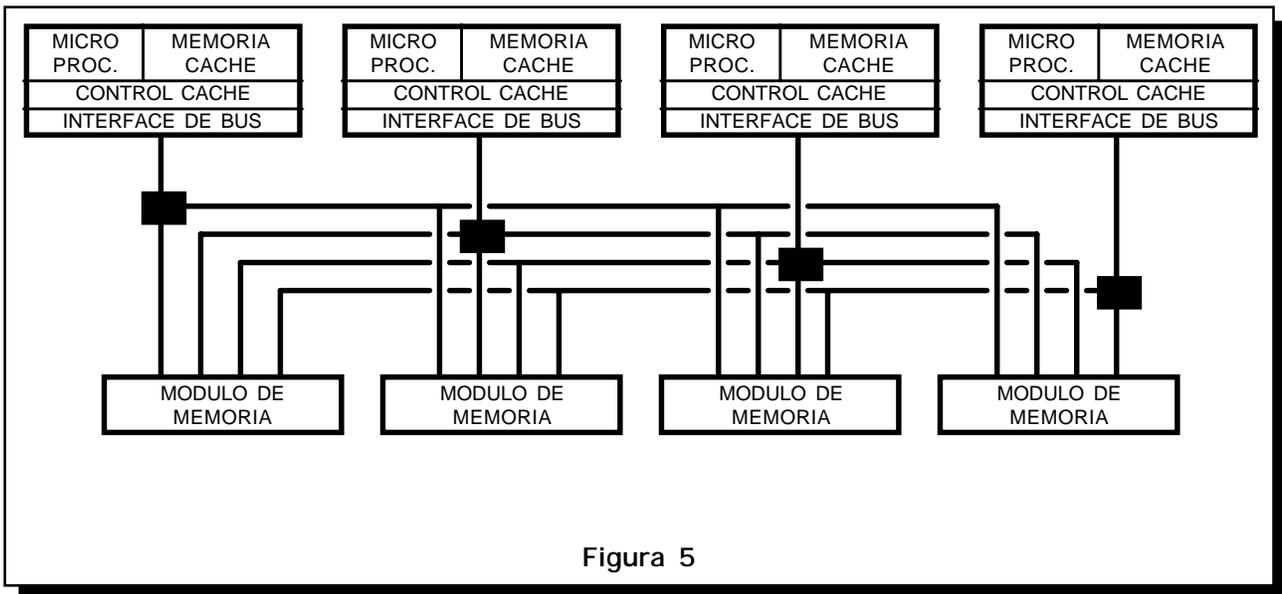


Figura 5

nes es numéricamente igual a la cantidad de términos cuyo exponente máximo es n, de la expansión de la siguiente expresión:

$$P = (p_1 + p_2 + \dots + p_N)^N \quad \dots\dots 2$$

donde p_m^n significa que n procesadores solicitan el acceso al módulo de memoria m.

Ejemplo. Sea N = 3. Entonces:

$$\begin{aligned} P &= [p_1 + p_2 + p_3]^3 \\ &= p_1^3 + p_2^3 + p_3^3 \\ &\quad + 3[p_1^2(p_2 + p_3) + p_2^2(p_1 + p_3) + p_3^2(p_1 + p_2)] \\ &\quad + 6 p_1 p_2 p_3 \end{aligned}$$

Aquí, P posee 3 términos de orden 3, 18 de orden 2 y 6 de orden 1. Esto es, en 3³ accesos se producen 3 que requieren 3 tiempos de ejecución, 18 que requieren 2 y 6 que requieren 1 tiempo de acceso cada uno. El tiempo medio de acceso es, de acuerdo con (1):

$$\tau_m = [6 + 2(18) + 3(3)]/27 = 1.889$$

La **tabla 1** reproduce el tiempo medio de acceso para hasta 16 procesadores. Para un multiprocesador de bus común, el tiempo medio de acceso es numéricamente igual al número de procesadores de la estructura. Sin embargo, el tiempo requerido para intercambiar información entre procesadores es menor que el tiempo de acceso a memoria (el costo unitario es menor). Por lo tanto, se puede concluir, con base en la **tabla 1**,

N	$\tau_m(N)$	N	$\tau_m(N)$
1	1.000	9	2.676
2	1.500	10	2.749
3	1.889	11	2.815
4	2.125	12	2.876
5	2.286	13	2.932
6	2.408	14	2.985
7	2.509	15	3.033
8	2.597	16	3.078

Tabla 1

biar información entre procesadores es menor que el tiempo de acceso a memoria (el costo unitario es menor). Por lo tanto, se puede concluir, con base en la **tabla 1**,

que existe un valor inferior para el número de procesadores N, tal que, para cantidades mayores, el rendimiento relativo del multiprocesador tipo MCBM es superior al del bus común.

Los valores dados en la **tabla 1** se ajustan, aproximadamente, a la siguiente expresión :

$$\tau_m = (1 + \log_2 N)^{0.7} \quad \dots\dots 3$$

Esto es, el tiempo medio de acceso crece en forma menor al logaritmo del número de procesadores.

Conviene hacer una observación final con relación a este modelo. El número de buses de interconexión crece linealmente con N. Esto impone un límite práctico a la escalabilidad del modelo, en tanto que todos los buses conectan a todos los módulos de memoria.

Consideraciones sobre la realización física

Se hizo un estudio preliminar de factibilidad de construcción del multiprocesador tipo MCBM, basado en el procesador PENTIUM de INTEL [7,8], con un tiempo de ciclo de 16 nanosegundos (una frecuencia de operación de 66 MHz), y una memoria caché secundaria de 256 Kbytes, operando sin estados de espera. El bus de datos de este procesador es de 64 bits, y el tamaño de línea de la memoria caché, de 256. El diseño del procesador permite la operación de subsistemas externos a la frecuencia del procesador, o submúltiplos de esta frecuencia. También soporta la operación síncrona, como asíncrona, de dispositivos externos. Se consideró que la memoria opera a la mitad de la frecuencia del procesador.

El número máximo de procesadores que puede ser utilizado en un multiprocesador tipo MCBM depende, esencialmente, del número de líneas físicas, esto es, de buses de interconexión, que pueden ser proporcionadas por una estructura de interconexión dada ("backplane"). Considerando reglas de diseño de 25 milésimas de pulgada, cada capa de circuito impreso soporta 40 líneas por pulgada, o 256 en poco más de 6 pulgadas. En un circuito multicapa, en el que se separan las líneas de comunicación por planos de tierra, es factible proporcionar 1024 líneas con tecnología convencional. Esto equivale a 32 buses de 32 bits, o 16 de 64 bits. Para el diseño de una máquina, la línea de comunicación debe ser vista como un recurso computacional, que posee un costo asociado. Para el multiprocesador de bus común, y para el procesador considerado, el total de líneas de intercomunicación es del orden de 200, incluyendo señales de control y de direccionamiento del bus de indagación. A este valor habría que sumar las líneas utilizadas para arbitraje de acceso, que depende del número de procesadores. Esto es, el multiprocesador de bus común requiere por el orden de 250 líneas físicas.

Para los módulos de memoria, se consideró una tecnología basada en dispositivos dinámicos de 60 nanosegundos de tiempo de acceso. En el modo de operación más sencillo, las memorias dinámicas almacenan secuencialmente la dirección de fila y la de columna para, posteriormente, proporcionar o aceptar el dato seleccionado. Esto es, los módulos de memoria pueden ser diseñados con base en un bus multiplexado, por medio del cual se transmite, primero, la dirección y comando, y posteriormente el dato.

Se consideró que los módulos de memoria soportan accesos en modo página para transferencias secuenciales de más de una palabra. Adicionalmente, se consideró el caso de módulos provistos con memorias tipo FIFO ("first in - first out") para el último dato, o línea, leída/escrita.

Los resultados de este estudio, expresados en el número de ciclos de procesador requeridos para diferentes tipos de transferencias de datos, se reproducen en las **tablas 2 y 3**. La **tabla 2** corresponde al intercambio aleatorio de líneas de 4 palabras de 64 bits (a) y de datos de 64 bits (b) para 4, 8 y 16 procesadores. Las cantidades sombreadas corresponden a 256 líneas físicas de intercomunicación, esto es, a un número de líneas comparable a las requeridas por el multiprocesador de bus común. Los valores restantes corresponden a configuraciones de 512, y 1024, líneas físicas.

La tabla 3 reproduce los resultados correspondientes al intercambio sin colisiones para los mismos casos descritos arriba.

Conclusiones

El multiprocesador tipo MCBM es la aproximación más rápida posible al modelo ideal de una PRAM. Bajo la hipótesis de costo unitario, el tiempo medio de intercambio aleatorio de información sin colisiones es 2, y de $1 + \tau_m$, para el caso de intercambios aleatorios con colisiones.

Desde el punto de vista de factibilidad de construcción, es posible construir máquinas de hasta 32 procesadores y buses de 32 bits, con base en tecnología convencional. Lo anterior equivale a más de 3,000 millones de instrucciones por segundo para el procesador consi-

derado (PENTIUM), y por el orden de 4,800 millones para el procesador i860 [9], ambos de INTEL. La complejidad de diseño, así como el costo, es básicamente comparable al de un multiprocesador de bus común. Si bien el esquema de buses múltiples requiere dispositivos de tercer estado para activar un bus en particular, no exige, en cambio, buses de muy alta velocidad, esenciales para el modelo de bus común. Por otra parte, el protocolo de escritura de actualización da origen a controladores de memoria, por lo menos un orden de magnitud más complejos, que los correspondientes a escritura transparente [10].

Una de las características más importantes del modelo MCBM, es la baja degradación que sufre el rendimiento de un procesador, comparado con su rendimiento individual. El único retardo asociado a la arquitectura es el tiempo de tránsito

Tipo de dato:	-a-		
Cant. de procesadores:	4	8	16
Multiproc. de bus común:	112	224	448
MCBM, 32 bits, sin FIFO:	98	119	142
MCBM, 32 bits, con FIFO:	85	104	123
MCBM, 64 bits, sin FIFO:	64	78	92
MCBM, 64 bits, con FIFO:	51	62	74

-b-		
4	8	16
48	96	192
43	52	62
30	36	43
38	47	55
26	31	37

Tabla 2

Tipo de dato:	-a-		
Cant. de procesadores:	4	8	16
Multiproc. de bus común:	112	224	448
MCBM, 32 bits, sin FIFO:	46		
MCBM, 32 bits, con FIFO:	40		
MCBM, 64 bits, sin FIFO:	30		
MCBM, 64 bits, con FIFO:	24		

-b-		
4	8	16
48	96	192
22		
16		
18		
12		

Tabla 3

por los dispositivos de tercer estado. Este retardo es comparable al asociado a dispositivos alimentadores de bus (buffer), que forman parte de cualquier diseño tipo uniprocador. En consecuencia, el rendimiento global depende, básicamente, de la cantidad de estados de espera introducida por los módulos de memoria. Dependiendo del rendimiento deseado, el tiempo de acceso puede reducirse a dos ciclos de memoria, con técnicas de diseño bien establecidas.

El modelo aquí propuesto es universal, en tanto que admite cualquier gráfica de problema, y no requiere algoritmos de tránsito. La granularidad de los algoritmos que pueden ser eficientemente ejecutados es mucho más fina a la de otros multiprocesadores. Con ello, se abren nuevas perspectivas para la investigación de algoritmos. Por otra parte, no existe limitación al formato de datos, en el sentido de que la fragmentación del mapa de memoria puede ser programada, y hasta dinámicamente reconfigurada, durante la ejecución del programa.

Una de las ventajas del protocolo de escritura transparente, es la implícita coherencia de memoria.

Esto simplifica la programación y sincronización de procesos considerablemente. Por otra parte, evita el tener que concentrar los resultados de un programa en memoria global al término de su ejecución ("flushing"). Finalmente, permite que el multiprocador constituya un elemento de procesamiento en una arquitectura de paralelismo masivo. En este sentido, el modelo propuesto amplía los límites del paralelismo físicamente realizable en, por lo menos, un orden de magnitud.

Bibliografía

- [1] Lindig, B. Miguel . *"Procesamiento en paralelo"*. Polibits, Año V, Número 11 Oct-Dic. 1993
- [2] Codenotti, B., Leoncini, M. *"Introduction to Parallel Processing"*. Addison-Wesley International Computer Science Series, 1993.
- [3] Fortune, S., Wyllie, J. *"Parallelism in Random Access Machines"*. Proc. of 10th Annual ACM Symp. on Theory of Computing, 1978, pp. 114-118.
- [4] Cache Tutorial. INTEL Corp., 1991.
- [5] Blasi, M. *"Computer Architecture"*. Addison-Wesley International Computer Science Series, 1990.
- [6] Lenoski, D. et al. *"The Stanford Dash Multiprocessor"*. IEEE Computer, vol.25, no.3, pp.63-79, marzo de 1992.
- [7] Pentium Processor User's Manual (3 tomos). INTEL Corp., 1993.
- [8] Reilly, J. *"Designing with the Pentium™ Processor, 82496 Cache Controller and SRAM CPU-Cache Chip Set"*. Application Note AP-481, INTEL Corp., julio de 1993.
- [9] i750, i860, i960 Processors and Related Products. INTEL Corp., 1994
- [10] Taufik T. *"Cache and Memory Design Considerations for the Intel486™ DX2 Microprocessor, Microprocessors"*. Vol.1, pp. 2-1137-1173, INTEL Corp., 1993.