

**REDUCCION DE LA DEGRADACION Y EL CONFLICTO
EN LAS REDES DE INTERCONEXION
PARA SISTEMAS MULTIPROCESADORES**

J.M. LLABERIA (*) E.HERRADA (**)
J.LABARTA(*) M. VALERO (*)

(*) Facultad de Informática. U.P.C. Barcelona.
(**) E.T.S.I. Telecomunicación. U.P.C. Barcelona.

Resumen. Uno de los parámetros causante de una disminución potencial de la eficiencia de un sistema multiprocesador es el tiempo de respuesta del subsistema de memoria. En este trabajo se presentan diversas técnicas que mejoran el rendimiento del mecanismo que conecta procesadores a módulos de memoria (la red de interconexión). El objetivo es conseguir reducir el conflicto (en los accesos a memoria) y la degradación (por el uso de los enlaces de la red). Se consideran aquí redes de tipo crossbar, de buses y barajadas. La base de estas técnicas reside en proponer métodos que optimicen la selección de los enlaces (arbitraje inteligente), en usar multiplexado temporal de tales enlaces, y en la utilización de memorias rápidas.

INTRODUCCION

Dos de los parámetros que disminuyen la eficacia de un sistema multiprocesador son el tiempo de respuesta del subsistema de memoria y la necesidad de realizar las primitivas de comunicación y sincronización entre los procesadores. En este trabajo nos centraremos en el primer punto.

Se denomina red de interconexión al mecanismo que conecta los procesadores con los módulos de memoria. Así pues, podemos considerar a la red como un conjunto de enlaces que permiten, dinámicamente, establecer conexiones procesador- memoria.

En un sistema multiprocesador es posible que los procesadores deban esperar ser servidos debido a las siguientes dos causas : a) el módulo de memoria está siendo accedido por otro procesador (conflicto en el acceso a memoria, y b) a que la red no disponga en ese instante de un enlace para realizar la conexión (degradación de la red).

En la literatura técnica se han propuesto y evaluado distintos tipos de redes. Cada una de ellas obedece a compromisos entre coste y número medio de módulos de memoria accedidos por unidad de tiempo (ancho de banda). Por ejemplo, la red crossbar no produce degradación aunque es la más costosa. En el polo opuesto esta la red unibus que produce máxima degradación per tiene un coste bastante menor. Como redes intermedias, que producen degradación a coste aceptable, están la red multibus y las redes barajadas (shuffle-exchange).

La red crossbar nos da, al no producir

degradación, una medida del conflicto en un sistema multiprocesador bajo ciertas hipótesis. El número medio de módulos de memoria ocupados es $0.6*n$, siendo n el número de procesadores y el de módulos de memoria. En este artículo se presentan dos mecanismos para reducir el conflicto : a) utilización de memorias cuyo ciclo de acceso es menor que el de procesador y b) añadir buffers en los módulos de memoria para almacenar las peticiones que no se pueden servir ya que el módulo de memoria está ocupado. Evidentemente una forma de reducir la interferencia es aumentar el número de módulos de memoria.

La evaluación de la red multibus permite obtener una medida (algo enmascarada por el efecto correlacionado del conflicto) de la degradación de la red. Aceptando un 5% de degradación el número de enlaces se puede reducir a $1/2 \min(n,m)+1$. En este artículo se presenta el mecanismo de multiplexación de enlaces (asignar los enlaces únicamente en los instantes de transferencia de información) como un medio de reducir la degradación de una red con menor número de enlaces que el crossbar.

Al analizar el mecanismo de selección de peticiones en una red barajada podemos observar que, debido al arbitraje distribuido y local, el número medio de peticiones servidas es bastante menor al posible en el caso de que dicho arbitraje fuera global.

En la sección 1 se exponen las hipótesis generales utilizadas para modelar los sistemas. En la sección 2 se presentan mecanismos de arbitraje en redes barajadas. En la sección 3 se presenta la multiplexación de enlaces como un medio para reducir la degradación. En la

2.1.- Un algoritmo casi óptimo

Consiste en calcular el número de colisiones que cada petición produciría con otras en la red. En consecuencia, eliminamos secuencialmente aquellas peticiones de acceso que presenten un mayor índice de colisiones, hasta llegar a conseguir un subconjunto que no produzca bloqueos. Es un proceso secuencial, porque cada paso que produzca alguna eliminación de peticiones, obliga a una nueva evaluación de los índices de colisión.

Puede verse en la tabla II los resultados obtenidos al simular este algoritmo, comparando con los del crossbar y con los de esta misma red Omega con arbitraje totalmente aleatorio y sin aplicación del algoritmo de eliminación de entrada de peticiones.

2.2.- Eliminación aleatoria

De cada grupo de procesadores que solicitan acceso a un mismo módulo de memoria, se selecciona (al azar) uno solo. El resto no entra a la red. Se realizaría este filtrado mediante la implementación de m árbitros del tipo n -usuarios 1 -servidor en la primera etapa de la red, (LAN82). Los resultados obtenidos al simular este método nos enseñan que debe modificarse, pues se producen pérdidas de eficacia debido a que pueden quedar muchos módulos de memoria inactivos en cada ciclo, a causa de haberse bloqueado en la red aquellas peticiones (solo hay una para cada módulo) que les iban dirigidas.

Por consiguiente, se modifica el algoritmo haciendo que una parte (f) de las peticiones totales (1) dirigidas a cada módulo de memoria, entran en la red. Hemos estudiado los diversos casos, con $f=2,3,4,\dots,s$. Se observa que los mejores resultados se consiguen con $f=s$. Véase la figura 2.

2.3.- Marcado de las peticiones

Si dividimos las solicitudes de acceso en dos clases ("fuertes" y "débiles"), tenemos que añadir 1 bit a cada información que atraviesa la red, pero esperamos conseguir mejoras en el ancho de banda resultante.

Ahora el arbitraje necesario exige : cuando surge una colisión en un switch elemental, se examinan los 2 bits de marcaje; si no coinciden, se elige para continuar a la petición "fuerte", y si coinciden se elige cualquiera de ellas.

Se han simulado tres variaciones de este método :

a) de cada grupo de peticiones que van dirigidas a un mismo módulo de memoria, se marca aleatoriamente una de ellas como fuerte y el resto como débiles.

b) Se marcan f de ellas ($f>1$) como fuertes.

c) Igual que en a) pero con nuevo marcado al paso por cada etapa de la red.

Pueden examinarse en la figura 2 los resultados obtenidos.

2.4.- Marcado inteligente de las peticiones

Se acompaña al método 2.3 con una información adicional, que exprese cuantos procesadores solicitan acceso, en cada ciclo, a un mismo módulo de memoria. Mediante la simulación de esta estrategia se ha obtenido la conclusión de que el número práctico de hilos adicionales es bastante inferior al de N , que sería el indicado, en principio. Así para un sistema 64×64 ($N=6$) basta con 3 líneas que señalen si hay 0,1,2,..., 5 ó 6 procesadores con petición "débil"; solo uno se marca como "fuerte".

Se ha encontrado como algoritmo óptimo de arbitraje, para esta situación, el siguiente : cuando coincidan dos peticiones fuertes en un switch, se selecciona la que va dirigida a la memoria menos solicitada; si coinciden dos peticiones débiles, la selección es a la inversa; si el marcado de ambas es diferente, se elige la fuerte.

También se han estudiado dos variaciones a este método :

a) el algoritmo de marcado solo se realiza una vez, a la entrada de la red.

b) se renueva el marcado en cada etapa.

Los resultados se presentan en la fig. 2.

3.- MULTIPLEXACION DE LOS ENLACES

En una red multiplexada los enlaces se ocupan únicamente durante el intervalo de tiempo necesario para que las peticiones de los procesadores lleguen a los módulos de memoria, o para que los módulos de memoria devuelvan los resultados a los procesadores. La multiplexación obliga a introducir hardware adicional, pero permite un mayor ancho de banda, menor número de enlaces y reducción de la interferencia al añadir más módulos de memoria dado un coste de la red.

En esta sección se evalúa la red unibus trabajando en modo multiplexado. La evaluación se realiza utilizando las hipótesis generales de la sección 2 más las siguientes hipótesis acerca del multiplexado.

a) El ciclo básico del sistema es constante e igual al tiempo de transferencia por el bus (t).

b) El ciclo de todos los módulos de memoria es el mismo e igual a $r \cdot t$, donde r es un valor entero.

c) El ciclo de procesador es $(r+2)t$.

sección 4 se analiza la utilización de memorias rápidas para reducir la interferencia. En la sección 5 se presenta la técnica del "buffering" de peticiones como un medio para reducir la interferencia. En la sección 6 se presentan las conclusiones del trabajo.

1.- HIPOTESIS DE FUNCIONAMIENTO

Para evaluar a los distintos sistemas se utilizan las siguientes hipótesis generales.

a) Los sistemas se componen de n procesadores y m módulos de memoria.

b) Las peticiones de los procesadores son independientes y están uniformemente distribuidas entre los distintos módulos de memoria.

c) Después de recibir un servicio de memoria, un procesador realiza inmediatamente una nueva petición, con probabilidad p ; los procesadores realizan sus peticiones únicamente en el inicio de los ciclos de procesador.

d) Las peticiones de los procesadores no servidas en un ciclo se mantienen hasta que son servidas.

2.- ARBITRAJE EN LAS REDES BARAJADAS

Aunque originariamente las redes de interconexión barajadas (shuffle-exchange) se usaron en máquinas SIMD, para alinear los datos, más recientemente se han propuesto para multiprocesadores. Su uso se ha evaluado, especialmente las redes delta y omega, suponiendo que los bloqueos se resuelven mediante arbitraje aleatorio entre las peticiones de acceso, realizándose con árbitros locales a cada conmutador de la red.

Proponer una red de interconexión barajada, con la correspondiente cantidad de degradación que introduce, exige evaluar el compromiso entre el menor coste frente al crossbar equivalente y la menor eficiencia obtenida. Así, por ejemplo en sistemas con $n=m=64$ módulos, se reduce en un 80% el número de conexiones si se usa una red barajada con switches 2×2 , pero evidentemente se obtiene un descenso en la eficiencia de la red, que podemos estimar a través del ancho de banda conseguido.

Podemos ver en la Fig. 1 una red barajada de tipo delta, en el caso $n=m=8$ (3 etapas)

En el funcionamiento de una red multietapa general, debe tenerse en cuenta que el bloqueo obedece a dos causas distinguibles:

a) a causa de que pueden colisionar peticiones dirigidas a módulos de memoria diferentes, en su paso a través de enlaces comunes de la red, tenemos un bloqueo inherente a ésta.

b) debido a que en cada switch se selecciona aleatoriamente cual de las dos entradas (supuestas ambas presentes) se va a conectar a una salida que está solicitada por ambas peticiones. Tenemos así un bloqueo adicional.

Vamos a ocuparnos aquí de presentar varios métodos de arbitraje entre peticiones que permitan mejorar el ancho de banda de la red. Con objeto de poder comparar resultados, para establecer criterios de ganancia, presentamos en la tabla I unos valores de ancho de banda, obtenidos según diversos modelos presentados en (PAT81), (THA81) para evaluar redes barajadas, y para el caso de red crossbar, (H0077).

En (PAT81) se introduce la siguiente relajación para obtener un modelo sencillo.

Las peticiones no aceptadas (bloqueadas) se ignoran en el siguiente ciclo, de forma que no influyen sobre las generadas al comienzo del siguiente ciclo.

Esta última hipótesis (suposición de pérdida de la memoria de las peticiones) no responde, evidentemente, al funcionamiento real. No obstante, simplifica el análisis y ha sido ampliamente utilizada, comprobándose con los resultados obtenidos que introduce unos errores bastante pequeños. Puede examinarse, p. ej. la referencia (VAL82).

En el estudio que se presenta, la simulación no utiliza la aproximación anterior, y por tanto se modela de forma más realista el sistema. Es decir, las peticiones no servidas en el ciclo i -ésimo se mantienen hasta el ciclo en que sean servidas. La red analizada será una de tipo Omega, pero los resultados obtenidos son válidos para cualquier red de interconexión con N etapas ($2^n = n = m$) de $n/2$ conmutadores elementales 2×2 .

Los algoritmos para un mejor arbitraje que aquí vamos a considerar, tiene un común el objetivo de disminuir lo que hemos denominado bloqueo inherente a la red. Para ello, reduciremos la degradación mediante la técnica de impedir que algunas de las peticiones procedentes de los procesadores lleguen a entrar a la red. Buscamos impedir que varias solicitudes dirigidas al mismo módulo de memoria aumenten el tráfico por la red sin que exista posibilidad de éxito, pues solo una de ellas (como máximo) podrá acceder. Por otra parte, añadiremos cierta información adicional a las peticiones, con objeto de facilitar un arbitraje más inteligente en los switches de la red, con lo cual tratamos de reducir el bloqueo adicional antes mencionado. En todo caso nos limitaremos a proponer métodos de arbitraje que sean razonablemente sencillos, para que su implementación exija poca circuitería añadida y poca penalización temporal adicional.

caso en que $K=3$ y $R=0$ apenas es mejor que tener $K=2$ $R=1$.

Si la política de arbitraje se hace puramente aleatoria (y no FIFO), los resultados obtenidos implican una pérdida en el ancho de banda del orden de un 8-10%.

Se ha construido un modelo analítico aproximado, usando algunas hipótesis simplificadoras que pueda consultarse en (LAB82), y que conduce a resultados numéricos con un margen de error inferior al 7% respecto a los de simulación. Se trata de un modelo que, en esencia, constituye una generalización de la fórmula de Strecker para el caso de $K>1$. Es interesante señalar que la fórmula obtenida resulta ser una estimación pesimista para los casos en que $n < km$, y optimista en los contrarios.

En la fig.4 puede verse el ancho de banda en función de m para $K=2$ y $n=8$. Se aprecia la ganancia respecto al caso $K=1$ (el efecto de las memorias rápidas), así como la exactitud conseguida con el modelo.

5.- BUFFERS EN LOS MODULOS DE MEMORIA

En esta sección se estudia la introducción de buffers en los módulos de memoria como un medio para reducir la interferencia.

En la figura 5 se representa un sistema con buffers y con una red unibus multiplexado. Con esta organización, cada vez que un módulo de memoria acaba un servicio, pone el resultado en el buffer de salida, y está ya disponible para servir una nueva petición, tomándola (si existe) de su buffer de entrada. La eficiencia del sistema se incrementa debido al hecho de que un módulo de memoria puede ahora estar ocupado sirviendo peticiones distintas en ciclos contiguos de bus.

En la evaluación del sistema se han utilizado las hipótesis generales, las del unibus multiplexado y se han añadido las siguientes :

a) el tiempo de acceso a los buffer se considera despreciable o incluido en el tiempo de acceso a memoria.

b) los buffers se gestionan utilizando una política FIFO.

c) El bus se asigna de forma prioritaria a las peticiones de los procesadores.

En la figura 6 se presentan valores de EBW para el sistema propuesto. A efecto de comparación se representan también los valores de EBW para una red crossbar con un ciclo básico igual a $r+2$ unidades de tiempo.

Puede observarse que los valores de EBW del crossbar sin buffer pueden mejorarse con un unibus multiplexado, con buffers, debido a que la interferencia de memoria se disminuye cuando se utilizan buffers en los módulos de

memoria.

Sin embargo, a medida que r aumenta, el unibus multiplexado con buffers tiende a los valores correspondientes del crossbar, debido a que disminuye el efecto de los buffers cuando la utilización del bus disminuye. En otras palabras cuando r tiende a ∞ , suponiendo un ciclo de procesador constante, el tiempo de servicio de los módulos de memoria aumenta y consecuentemente el tiempo de ocupación del bus disminuye, produciendo una mayor interferencia y una menor degradación de la red.

Los valores de EBW calculados muestra que el efecto de buffering es proporcionalmente mayor cuando la diferencia ($n-m$) aumenta, debido a la gran interferencia que tal organización implica.

El unibus multiplexado con buffers en los módulos de memoria opera en saturación hasta valores de r cercanos a $\min(n,m)$.

Con $r \sim \min(n,m)+2$ se obtienen valores de EBW mejores que con un crossbar. Por ejemplo, con $r=18$ ($n=m=16$) y la red propuesta se obtiene una eficiencia semejante a la del crossbar 16×16 .

6.- CONCLUSIONES

En este trabajo se han presentado mecanismos para reducir la degradación introducida por la red y el conflicto debido al número finito de módulos de memoria.

Para reducir la degradación se ha presentado la técnica de multiplexación de enlaces. Permite aprovechar al máximo el recurso red con un incremento en la complejidad del hardware de arbitraje. Los resultados obtenidos de la evaluación, realizada con un único bus, permiten presentar esta técnica como una alternativa a la multiplexación de circuitos. Además, al tener en cuenta el coste tenemos dos parámetros de diseño (r y m) para obtener un ancho de banda determinado. Uno de los parámetros (r) influye en la degradación y el otro (m) en el conflicto.

En el caso de redes barajadas se han presentado distintas formas de arbitraje que permiten aumentar a un coste razonable el ancho de banda en un 20%.

En el campo de sistemas multimicro-procesadores se ha presentado la utilización de memorias rápidas como un medio para reducir la interferencia. Utilizando memorias cuyo ciclo de acceso es la mitad del de los procesadores se obtienen incrementos en ancho de banda del orden del 40% al 60%.

Por último se ha presentado la técnica de "buffering" de peticiones. Los resultados indican que se pueden obtener con un solo bus anchos de banda superiores al crossbar (sin multiplexación ni buffering). El coste de esta red permite pensar también en el aumento del

e) En un ciclo básico, el bus puede ser solicitado por procesadores y módulos de memoria. Se han evaluado las siguientes dos políticas de asignación del bus.

e') prioridad a las peticiones de los procesadores.

e") prioridad a las módulos de memoria.

f) Al ir a asignar el bus se tienen en cuenta únicamente las peticiones a módulos de memoria desocupados. La elección entre varias peticiones se realiza de forma aleatoria.

De acuerdo con las anteriores hipótesis de funcionamiento, el ancho de banda efectivo del sistema, EBW, se calcula por la siguiente expresión

$$EBW = P_b(r+2)/2$$

donde P_b es la utilización del bus, y EBW es el número esperado de peticiones a módulos de memoria servidos por ciclo de procesador.

El valor máximo alcanzable por EBW es $(r + 2)/2$, que es superior al alcanzable (1) si el bus no se multiplexa.

En la figura 3 se presentan resultados obtenidos por simulación utilizando las hipótesis anteriores. Estos resultados muestran que los EBW obtenidos con la política e') de asignación del bus son mejores que los obtenidos utilizando la política e"). Para propósitos de comparación en la figura se han representado valores de EBW de una red crossbar sin multiplexación de los enlaces y con un ciclo de operación básico igual a $(r+2)t$. Como se puede observar, la red unibus multiplexada presenta un buen rendimiento cuando r aumenta. Para valores bajos de r se obtiene la cota del ancho de banda y a medida que se aumente r el EBW se aproxima al del crossbar.

En el diseño de una red unibus multiplexada, se pueden utilizar dos parámetros (r, m) para obtener un rendimiento comparable a la red crossbar. Por ejemplo, el EBW de una red crossbar 8×8 se puede obtener con $r=8$ y $m=14$ utilizando una red con un bus multiplexado; si se acepta un 5% de degradación es suficiente $m=10$. En la red multibus son necesarios 4 buses para la misma degradación.

4.- MEMORIAS RAPIDAS EN UN SISTEMA CON RED CROSSBAR

Actualmente es razonable diseñar un sistema multiprocesador en el que los módulos de memoria permitan varios accesos en cada ciclo, si los elementos de proceso son microprocesadores estándar.

Si la red de interconexión es de tipo crossbar, no produce degradación, y con el uso de memorias de más de un acceso en cada ciclo

de procesador conseguimos, por su parte, reducir el conflicto en los accesos. Es una forma alternativa, que no obliga a aumentar el número m de módulos de memoria en el sistema, de aumentar la eficiencia del sistema, que podemos medir a través del ancho de banda.

Es posible emplear, hoy día, memorias con tecnología MOS (a precios razonables) que permitan 2 ó 3 accesos durante el tiempo de ciclo de un microprocesador normal de 16 bits. En todo caso, si el coste global del sistema multimicroprocesador debe mantenerse poco elevado, esta propuesta de memorias rápidas puede reservarse para el primer nivel de la jerarquía de memorias del sistema total. Una primera referencia a esta solución arquitectónica puede encontrarse en (VAL80), mientras que en (LI80) y en (VAL81) se presentan algunas evaluaciones de rendimiento.

Con objeto de poder construir modelos de predicción del sistema con memorias rápidas y red crossbar, que es el objetivo de esta sección, conviene definir algunos parámetros:

t_m es la unidad básica de tiempo, igual al ciclo de memoria. Se supone constante, e idéntica en lectura y en escritura.

t_p es el ciclo del procesador. Suponemos $t_p = (R+K) \cdot t_m$; K es el número de ciclos durante t_p los cuales, al menos, la petición de acceso está activa. R se refiere al número de ciclos básicos que la petición emplea en llegar al árbitro (para traducción de direcciones). Supongamos K y R enteros.

Un procesador puede obtener el acceso que solicitó, antes de que acabe su ciclo t_p ; en tal caso deberá esperar para poder lanzar una nueva petición. En el caso de que aun no haya logrado el servicio, mantiene la petición durante más intervalos de ciclo de memoria.

En cuanto a la política de arbitraje para este sistema, se seguirá una organización FIFO (en lugar de aleatoria ó de prioridad fija). De este modo se evita el posible inconveniente de conceder el acceso a un módulo de memoria a aquellos procesadores que hace K ciclos (ó menos) que lanzaron su petición, y que retardarían la nueva generación de peticiones. En caso de que haya varios procesadores con igual número máximo de ciclos de espera, se arbitrará con prioridad fija entre ellos.

Pueden verse, en las tablas III y IV, los resultados obtenidos al simular este sistema, con $R=0$ y valores de K iguales a 2 y a 3. La comparación de estas simulaciones (para $t_p = t_m$) con los resultados presentados en (BHA75), donde se estudia y evalúa la red crossbar a través de un modelo exacto, ha dado unos errores inferiores al 0,1%.

Examinando los resultados numéricos obtenidos, apreciamos la gran mejora al pasar de $K=1$ a $K=2$, mientras que posteriores aumentos de K apenas favorecen ya el ancho de banda. Esto es cierto a no ser que hagamos que $m \ll n$. Por otro lado observamos que tener el

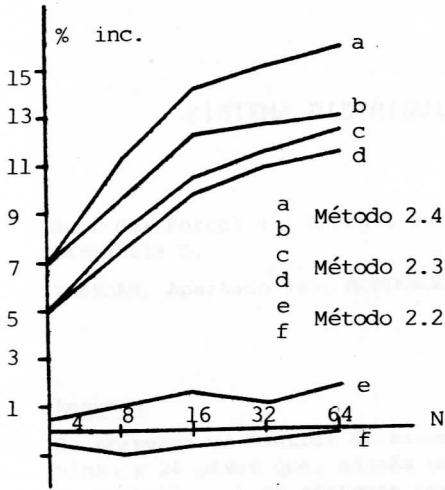


Fig. 2 Resultados

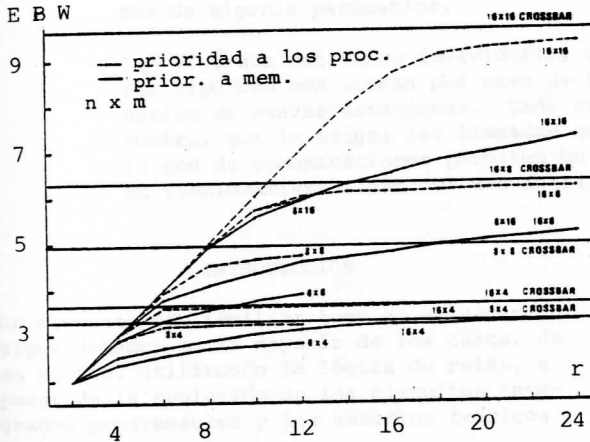


Fig. 3 Unibus multiplexado

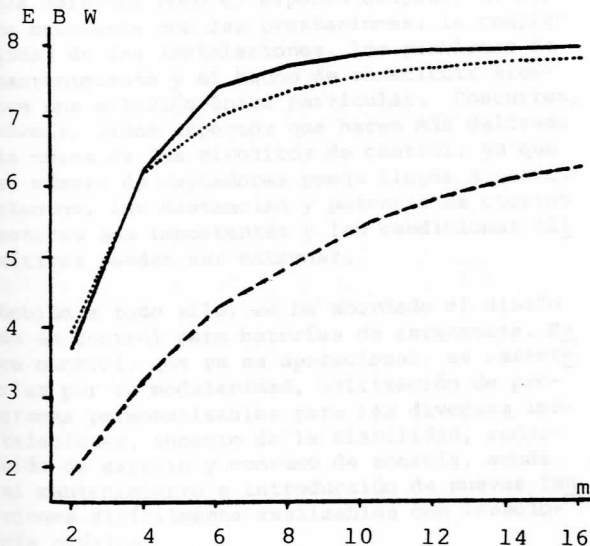


Fig. 4 EBW vs. m para 8 procesadores.
 — simulación espacio K=2, R=0
 modelo aproximado K=2, R=0
 - - - - simulación K=1, R=0

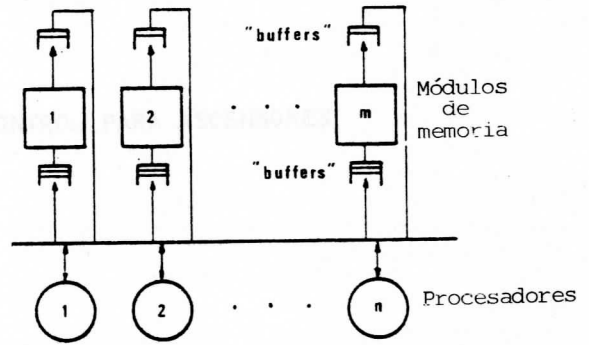


Fig. 5 Sistema con buffers

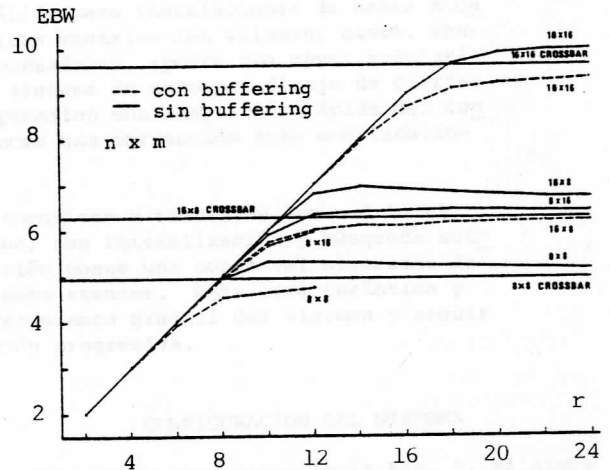


Fig. 6 Módulos de memoria con buffers y multiplexación del bus.

AGRADECIMIENTOS

Este trabajo está subvencionado por la Comisión Asesora de Investigación Científica y Técnica (ref. 1739/82).

número de módulos de memoria para reducir aun más el conflicto.

REFERENCIAS

(BHA 75) BHANDARKAR, D.P.
 "Analysis of memory interference in multi-processors". IEEE. Trans on Comp. Vol. C-24, n 9, Sept 1975, pp. 897-908.
 (HOO 77) HOOGENDOORN, C.
 "A general model for memory interference in multiprocessors". IEEE, Trans. on Comp. Vol. C-30, Oct. 1977
 (LAB82) J.LABARTA et al.
 "Performance evaluation of the crossbar interconnection networks with fast memories". Proc. of the Int. Symp. MIMI'82. Paris.
 (LAN 82) LANG, T. AND VALERO, M.
 "M-Users, B-servers arbiter for multiple-bus multiprocessor". Microprocessing and Microprogramming. The Euromicro Journal, Aug. 1982, pp 11-18.
 (LI 80) LI, H.F.
 "Bandwidth of fast memory in multiprocessing". Proc. IEEE, Vol. 68, n 5, May 1980.
 (PAT 81) PATEL, J.H.
 "Performance of processor-memory interconnection for multiprocessors". IEEE, Trans. on Comp. Vol. C-30, Oct. 1981.
 (THA 81) THANAWASTIEN, S. AND NELSON, V.P.
 "Interference analysis of shuffle/exchange networks". IEEE Trans. on Comp. Aug. 1981.
 (VAL 80) VALERO, M.
 "Contribución al estudio de redes locales de microcomputadores". Tesis Doctoral. Barcelona, March 1980.
 (VAL 81) VALERO, M. ALEGRE, I. AND SANVICENTE, E.
 "Mathematical models to evaluate the memory interference in multimicrocomputer systems". MIMI'81. México, pp 21-26.
 (VAL 82) VALERO, M. et al.
 "Markov and approximate models for multiple-bus and multiple-bus with partial busses interconnections networks". MIMI'82, Cambridge (USA). July 1982.

SUMMARY

One of the parameters responsible for the decreasing of a multiprocessor system efficiency is the memory subsystem response time.

In this paper, several techniques that lead to a better performance of the interconnection network between processors and memories, are presented. The goal is to achieve lower values of the conflict (in the memory accesses) and of the degradation (in the use of the networks links). Interconnection networks of the types: crossbar, single and multiple busses, and shuffle-exchange, are considered.

These techniques are based on optimization methods of links selection (intelligent arbitration), time-multiplexing of the links, and the utilization of fast memory modules.

$p = 1$ (sin procesado interno)

n = m	PAT	THA	*	+	CRO
4	2.43	2.46	2.37	2.44	2.62
8	4.12	4.18	3.91	4.13	4.95
16	7.18	7.27	6.71	7.18	9.59
32	12.77	12.86	11.81	12.82	18.96
64	22.98	23.11	21.13	23.04	37.70

TABLA I

* : hipótesis "memoryless" (simulación)
 + : sin " " (")
 CRO : crossbar
 PAT : modelo de Patel (PAT81)
 THA : modelo de Thanawastien (THA81)

n = m	CRO	Ω	casi óptimo
4	2.62	2.37	2.53
8	4.95	3.91	4.43
16	9.59	6.71	8.02
32	18.96	11.81	14.74
64	37.70	21.13	29.50

TABLA II. Anchos de banda por simulación

Número de procesadores ²	Nº de mod. de mem.				
	4	8	12	16	16
4	3.997	4.000	4.000	4.000	4.000
8	5.550	7.895	7.997	8.000	8.000
12	5.709	10.052	11.936	11.994	11.999
16	5.779	10.694	15.351	15.927	15.989

TABLA III. Resultados simulación, K=2 y R=0

Número de procesadores ²	Nº de mod. de mem.				
	4	8	12	16	16
4	3.411	3.949	3.995	3.998	4.000
8	3.721	6.251	7.739	7.922	7.969
12	3.813	6.921	10.362	11.498	11.795
16	3.860	7.233	11.905	14.398	15.340

TABLA IV. Resultados simulación, K=3 y R=0

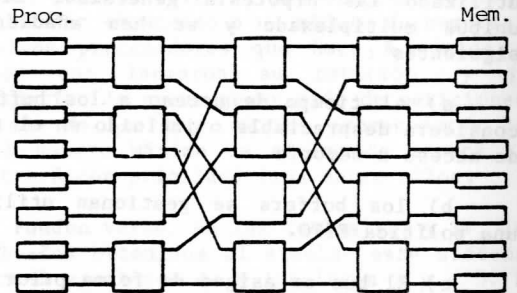


Fig. 1 Red Delta.