



TESIS

MODELADO, ANALISIS Y SIMULACION DE
REDES DE INTERCONEXION EN SISTEMAS
MULTIPROCESADORES

por

ANTONIO JESUS TORRALBA SILGADO

Ingeniero Industrial por la E.T.S. de I.I.

de la Universidad de Sevilla.

presentada en la

ESCUELA TECNICA SUPERIOR DE INGENIEROS INDUSTRIALES

de la

UNIVERSIDAD DE SEVILLA

para la obtención del

GRADO DE DOCTOR INGENIERO INDUSTRIAL

SEVILLA, OCTUBRE DE 1983

RESUMEN.

En este trabajo se estudian los diferentes tipos de redes de interconexión procesador-memoria en sistemas multiprocesadores con especial hincapié en las redes "cross-bar", multibuses y delta, por ser las más ampliamente conocidas y utilizadas.

Se hace una descripción de los principales métodos de análisis, tanto exacto como aproximado, discreto y continuo en el tiempo. Los métodos aproximados se comparan entre sí y con resultados exactos cuando es posible o con resultados de simulación en los casos más complejos.

Se propone un modelo original aplicable a los sistemas "cross-bar" y, sobre todo, a los sistemas multibus en tiempo continuo. Este modelo, de una gran simplicidad, da, no obstante, unos resultados muy próximos a los exactos.

A partir de los modelos anteriores se comparan los tres tipos de redes antes citadas en cuanto a sus prestaciones, complejidad y fiabilidad y tolerancia a faltas. Se realiza también un análisis del efecto que tienen sobre las prestaciones del sistema algunos parámetros no considerados en los modelos anteriores.

Por último, se presenta un simulador de sistemas multiprocesadores que, por estar basado en las Redes de Petri, supone una innovación respecto de los simuladores tradicionalmente empleados.

AGRADECIMIENTOS.

Deseo expresar mi agradecimiento a todos aquellos, familiares, profesores, alumnos y compañeros que de una forma u otra me han ayudado a la realización de este trabajo y muy especialmente:

- A D. José Luis Calvo, bajo cuya dirección ha sido realizada esta Tesis, por sus acertados consejos y su constante apoyo y estímulo.
- Al Colegio Oficial de Ingenieros Industriales de Andalucía Occidental por la confianza depositada al otorgarme una Beca.
- A todos mis compañeros de Dpto. y, en especial a D. Leopoldo G. Franquelo, por las sugerencias y ayuda prestadas.
- A mis padres y a Laura por su paciencia.

INDICE

CAPITULO 1.	1
INTRODUCCION GENERAL.	1
1.1 GENERALIDADES.	1
1.2 OBJETIVOS Y PLANTEAMIENTO DE LA TESIS.	3
CAPITULO 2.	6
METODOS DE MODELADO Y ANALISIS DE SISTEMAS MULTIPROCESADORES.	6
2.1 INTRODUCCION.	6
2.2 TECNICAS BASADAS EN SIMULACION.	7
2.3 ELABORACION DE PROTOTIPOS.	11
2.4 METODOS MARKOVIANOS Y BASADOS EN LA TEORIA DE COLAS.	13
CAPITULO 3.	23
SISTEMAS "CROSS-BAR".	23
3.1 INTRODUCCION.	23
3.2 ANALISIS DISCRETO DE LOS SISTEMAS "CROSS-BAR".	24
3.2.1 Modelo Discreto de un Sistema Completamente Interconectado.	26
3.2.2 Parámetros del Sistema.	28
3.2.3 Modelo Exacto.	29
3.2.4 Aproximación de STRECKER.	36
3.2.5 Aproximación Binomial.	38
3.2.6 Aproximación de Descomposición.	41
3.2.7 Comparación de las diferentes aproximaciones.	44

3.3 SISTEMAS CON PROBABILIDAD m DE ACCESO POR CICLO.	46
3.3.1 Modelo Exacto.	48
3.3.2 Aproximación de Patel.	51
3.3.3 Aproximación Binomial.	52
3.3.4 Aproximación de Hoogendoorn (Modelo GMI).	54
3.3.5 Aproximación de Bhandarkar.	57
3.3.6 Aproximación de Yen.	59
3.3.7 Aproximación de Bhandarkar Modificada.	61
3.3.8 Comparación de las diferentes aproximaciones.	66
3.5 ANALISIS EN TIEMPO CONTINUO.	67
CAPITULO 4.	70
INTERCONEXION MEDIANTE BUSES.	70
4.1 INTRODUCCION.	70
4.2 ANALISIS EN TIEMPO DISCRETO.	72
4.3 ANALISIS EN TIEMPO CONTINUO.	78
4.3.1 Modelo Aproximado de Kriz.	78
4.3.2 Modelo de Jacobson.	83
4.3.3 Modelos aproximados de Ajmone Marsan.	87
4.4 SISTEMAS CON BUSES PARCIALES.	91
4.5 SISTEMAS CON UN UNICO BUS.	94
4.6 OTROS SISTEMAS CON UN UNICO BUS.	103
CAPITULO 5.	108
REDES DE INTERCONEXION.	108
5.1 INTRODUCCION.	108
5.2 MODELO DE UNA RED DELTA.	117
5.3 REDES DELTA TOLERANTES A FALTA.	123
5.4 REDES DELTA CON BUFFERS ENTRE ETAPAS.	127

CAPITULO 6.	131
ESTUDIO COMPARATIVO DE LOS SISTEMAS DE INTERCONEXION PROCESADOR MEMORIA.	131
6.1 INTRODUCCION.	131
6.2 ANCHOS DE BANDA.	134
6.3 ORDEN DE COMPLEJIDAD DE LAS REDES.	138
6.4 ORDEN DE COMPLEJIDAD VLSI.	144
6.5 FIABILIDAD Y TOLERANCIA A FALTAS.	150
CAPITULO 7.	155
EXTENSION DE LOS MODELOS.	155
7.1 INTRODUCCION.	155
7.2 ANALISIS DISCRETO EN EL TIEMPO.	156
7.2.1 Modelos no uniformemente referenciados.	156
7.2.2 Sistemas con más de un ciclo por acceso. (El problema del "cache").	169
7.2.3 Modelos con diferentes políticas de arbitraje de las memorias.	172
7.3 ANALISIS CONTINUO EN EL TIEMPO.	175
7.3.1 Procesadores sesgados hacia una única memoria.	176
7.3.2 Procesadores sesgados a memoria favorita.	177
7.3.3 Los procesadores acceden con diferente velocidad.	179
CAPITULO 8	182
PAQUETE DE PROGRAMAS DE SIMULACION	182
8.1 INTRODUCCION.	182
8.2 LENGUAJE DE SIMULACION.	185
8.3 ESTRUCTURA DEL PAQUETE DE PROGRAMAS DE SIMULACION.	189

8.3.1 Programa de Expansión de Macros y Preprocesado de Expresiones (PRORED).	191
8.3.2 Programa de Generación de Tablas (TRADUC).	191
8.3.3 Programa de Transformación y Envío de Tablas (LINBUS).	191
8.3.4 Programa de Simulación.	192
8.3.5 Módulos Auxiliares.	192
8.4 TRATAMIENTO DE LAS REDES DE PETRI.	193
8.5 VALIDACION DEL SIMULADOR Y EJEMPLOS DE SU USO.	196
CONCLUSIONES.	200
BIBLIOGRAFIA	202
APENDICE I.	216

CAPITULO 1.

INTRODUCCION GENERAL.

1.1 GENERALIDADES.

Los sistemas basados en procesadores han sufrido un notable crecimiento desde los sistemas de trabajo en tanda ("batch-processing") de los años 60 hasta los sistemas en tiempo compartido ("time-sharing processing") de la década de los 70.

El coste de los componentes hardware, como resultado de la baja capacidad de integración disponible, supuso en esta época el principal factor limitativo del crecimiento de los sistemas.

La necesidad presente en la década de los 80 de disponer de sistemas con capacidad para procesar miles de millones de instrucciones por segundo (lo que sobrepasa en varios de órdenes de magnitud las velocidades alcanzables, incluso hoy día, con sistemas basados en único procesador) así como los cada día más exigentes niveles de fiabilidad requeridos obligó al diseño de nuevas arquitecturas que rompían el ya tradicional esquema de Von Newman.

Dentro de los Planes de Investigación de UNESA se acometieron, por parte del Dpto. de Electrónica y Automática de la ETSII de Sevilla, diferentes trabajos tendentes al Análisis de Redes Eléctricas y Automatización de Subestaciones y pequeñas Centrales

Hidráulicas.

Ambas aplicaciones son claros ejemplos de sistemas que requieren de una rápida capacidad de respuesta en tiempo real y una elevada fiabilidad. Las arquitecturas clásicas basadas en uno o pocos buses no dan respuesta a estas necesidades y ello fue el motivo que dio origen a esta Tesis.

Dos direcciones diferentes son posibles para la resolución del problema.

De una parte, los sistemas SIMD. Estos sistemas, aún cuando siguen conservando una única CPU, la rodean de dispositivos hardware que permiten realizar simultáneamente una misma instrucción sobre múltiples datos. Estos sistemas parecen ser bastante apropiados para la resolución de determinados problemas tales como complejos sistemas de ecuaciones lineales o diferenciales.

De otra parte, la utilización conjunta de cientos de procesadores actuando coordinadamente en paralelo cuyas velocidades (al menos en teoría) pueden sumarse, se ofrece como una alternativa más amplia (si bien más compleja) dando origen a los sistemas MIMD.

No obstante, esta evolución sólo ha sido posible debido al desarrollo de las modernas tecnologías de fabricación VLSI. Componentes más rápidos (Arsenio-Galio, dispositivos de unión de Josephson) y mayor densidad de integración han sido las claves para el desarrollo de estos nuevos sistemas.

Si bien los sistemas multiprocesadores parecen la respuesta más clara a los problemas de velocidad y fiabilidad, estos sistemas tienen aún hoy día problemas muy importantes que resolver.

De entre estos problemas el más inmediato es el de su interconexión y comunicación a través de memorias comunes y que constituye el objetivo de esta Tesis.

Desde la aparición del sistema cross-bar, manifestado en el Cmp de la Universidad de Carnegie-Mellon, muchas otras redes de interconexión han sido propuestas entre las cuales cabe destacar como las más interesantes las redes de Banyan y, de entre ellas, las redes Delta propuestas por Patel en PATE79.

1.2 OBJETIVOS Y PLANTEAMIENTO DE LA TESIS.

El objetivo principal de esta Tesis es el estudio de las diferentes Redes de Interconexión Procesador-Memoria en Sistemas Multiprocesadores.

De entre las diferentes arquitecturas propuestas se han seleccionado las tres que se han considerado más representativas. A saber:

- Redes Cross-Bar
- Redes Multibuses y
- Redes Delta.

En un primer paso y dada la generalidad del problema planteado se define un modelo para cada una de estas redes que las haga evaluables y estructuralmente comparables.

Estos modelos son abordados en los primeros capítulos de la Tesis desde los puntos de vista discreto y continuo en el tiempo.

El análisis en tiempo discreto es representativo de sistemas fuertemente interconectados y sincronizados en que es necesario considerar su comportamiento al nivel de instrucción.

El análisis en tiempo continuo es representativo de sistemas débilmente interconectados que intercambian mensajes a través de las memorias comunes.

Los modelos planteados son resueltos analíticamente en los casos más simples y, para los casos más complejos, se estudian y comparan las diferentes aproximaciones presentadas por diferentes autores, propóniéndose, en algunos casos, nuevas aproximaciones.

Allí donde ni siquiera los modelos aproximados son analíticamente tratables se ha hecho uso de resultados procedentes de simulación a cuyo fin se ha realizado un Paquete de Programas de Simulación basado en Redes de Petri y presentado en el Capítulo 8. La utilización de las Redes de Petri para la representación y simulación de sistemas multiprocesadores supone una interesante innovación.

En el Capítulo 6 se realiza un estudio comparativo de estas Redes en cuanto a sus prestaciones, complejidad, fiabilidad y tolerancia a faltas.

Por último, en el Capítulo 7 se estudia una aproximación a modelos más reales para los sistemas tipo "cross-bar". En este capítulo se analizará cómo inciden en las prestaciones de estos sistemas algunas desviaciones respecto de los modelos ideales analizados en los primeros capítulos.

CAPITULO 2

METODOS DE MODELADO Y ANALISIS DE SISTEMAS MULTIPROCESADORES.

2.1 INTRODUCCION.

En la figura 2.1 se muestra el espectro de las métodos utilizados para el análisis y modelado de sistemas multiprocesadores.

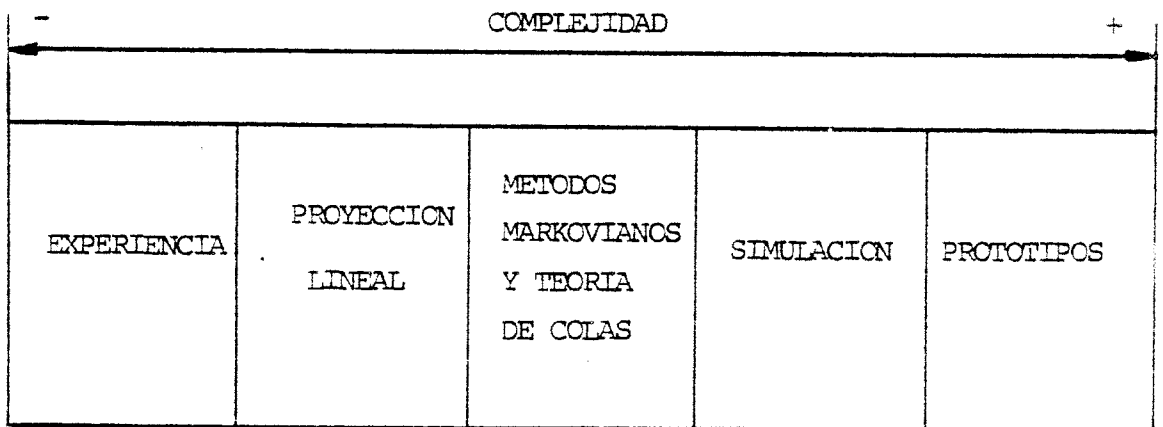


Fig. 2.1

Estas técnicas cubren desde las reglas obtenidas por la experiencia cuya aplicación es fácil y su costo muy reducido o nulo pero cuya efectividad es dudosa hasta la elaboración de prototipos que permiten obtener medidas muy fiables del comportamiento final del sistema pero con la penalización de un elevado coste y complejidad.

Referido a las reglas basadas en la experiencia (del tipo "un canal ocupado más del 30% de su tiempo está sobrecargado" o similares), poco puede decirse, si bien puede justificarse su uso en la resolución de problemas cotidianos y fueron las primeras herramientas usadas por los diseñadores de sistemas a falta de otros.

Moviéndonos en el espectro anterior la proyección lineal ha sido durante mucho tiempo una poderosa herramienta de diseño, si bien su efectividad se puede poner en duda sobre todo cuando intenta dar una extrapolación lineal a sistemas inherentemente no lineales. Estas técnicas han sido profusamente utilizadas en el pasado pero su aplicación actual es muy reducida.

El resto de las técnicas de la fig. 2.1 tienen un mayor uso y serán presentadas a continuación.

2.2 TECNICAS BASADAS EN SIMULACION.

La complejidad de los sistemas a analizar ha hecho que los métodos basados en simulación hayan sido ampliamente utilizados, tanto para estudiar el comportamiento de los sistemas como para validar métodos de análisis aproximado.

No obstante, respecto a los métodos basados en simulación es preciso realizar algunas consideraciones.

En primer lugar hemos de tener en cuenta que los resultados obtenidos mediante simulación no son resultados exactos sino en todo caso, resultados próximos a los resultados exactos.

Por otra parte, el analista que decide utilizar métodos basados en simulación debe resolver dos problemas previos a la simulación.

1.- Establecer qué nivel de aproximación a los resultados exactos se obtiene a partir de la simulación efectuada (es necesario recalcar que este nivel de aproximación se establece respecto al modelo utilizado para simular el sistema y nunca respecto del comportamiento del sistema modelado).

2.- Establecer en qué medida el modelo del sistema se comporta tal como lo hace el sistema modelado.

El nivel de aproximación a que hace referencia este segundo problema es muy difícil de cuantificar. En cualquier caso el modelo será generalmente una idealización simplificada del sistema y el analista deberá tener presente que con dicho modelo nunca podrá conocer cómo inciden las variables no consideradas en el comportamiento del sistema.

Por otra parte pretender tener en cuenta todas las variables que inciden sobre el sistema puede conducir a modelos que resultan ser más complejo que el propio sistema a simular, dando lugar a un proceso largo, tedioso y difícil de construir y validar.

Una vez asumido el modelo por el analista, éste se enfrenta al primero de los problemas antes planteados; es decir, cuánto se aproxima el resultado de simulación al resultado exacto del modelo.

En general, se utiliza como parámetro estimativo el denominado INTERVALO DE CONFIANZA, de forma que podemos establecer que el parámetro estimado en el proceso de simulación (su media o varianza) se encontrará dentro del intervalo obtenido con un determinado nivel de confianza (digamos como ejemplo el 95%).

La estimación de un intervalo de confianza es difícil para una simulación arbitraria. La aproximación más rigurosa ha sido establecida por Lavenberg [LAVE75] y es conocida por el método regenerativo.

El método regenerativo se basa en la propiedad que tienen determinados procesos (p.e. los procesos markovianos) en los cuales es posible identificar unos estados (llamados puntos de regeneración) los cuales cuando son alcanzados por el sistema puede decirse que el proceso probabilísticamente comienza de nuevo; es decir, la posible evolución del sistema cuando alcanza uno de tales puntos se hace independiente de la historia seguida por el sistema hasta alcanzar dicho punto.

Una vez determinado uno de estos puntos y en el supuesto de que dos pasos sucesivos del sistema por dicho punto se realice en un tiempo finito, se procede a la simulación del sistema midiendo los tiempos que tarda en atravesar dos veces consecutivas el pun-

to de regeneración.

Midiendo estos intervalos de tiempo y la evolución de la variable (o variables) que se pretende estimar a lo largo de cada uno de estos intervalos, Lavenberg establece un intervalo de confianza asintótico por aplicación del teorema central del límite.

De una forma más simple otros intervalos de confianza (esta vez aproximados) pueden ser obtenidos mediante técnicas más clásicas tales como asumir una distribución normal de la variable cuya media o varianza se pretende estimar.

Por último, el analista que ha hecho uso de un método de simulación, se pregunta cuántas veces ha de repetir el proceso (partiendo cada vez de nuevas secuencias aleatorias de entrada) de forma que su intervalo de confianza sea tan estrecho como el que inicialmente escogió.

Este último problema puede resolverse mediante la aplicación del método denominado de "las paradas secuenciales" y establecido por Lavenberg en [LAVE77]. Este método simplemente deja correr durante un tiempo el proceso de simulación tras lo cual lo para y se estima (por el método regenerativo antes mencionado o por cualquier otro método) el intervalo de confianza del nivel fijado. Si este intervalo no es lo suficientemente estrecho se continúa el proceso de simulación hasta que en una de las paradas se alcance el nivel requerido.

En la actualidad existen paquetes de programas de simulación que tratan modelos de multiprocesadores que facilitan en gran medida la tarea del analista.

Un paquete de programas de simulación ha sido también realizado en el curso de esta Tesis bajo un punto de vista bastante original usando la gran potencia intuitiva de representación de sistemas que poseen las redes de Petri. Este paquete de programas de simulación así como su utilización y resultados se presentan en el capítulo 8.

2.3 ELABORACION DE PROTOTIPOS.

En el extremo más alejado del espectro nos encontramos con métodos basados en la elaboración de prototipos. La base de estos métodos es muy simple y consta de tres pasos:

- 1.- Elegir un conjunto de estados representativos de la carga de funcionamiento que va a soportar el sistema
- 2.- Hacer trabajar a la máquina propuesta (o un prototipo de ella) bajo este nivel de carga y
- 3.- Recoger los datos sobre los parámetros que nos interesan.

Sin embargo, cualquiera de estos tres pasos es de un elevado nivel de complejidad. En primer lugar es muy difícil definir qué

condiciones debe cumplir un estado de trabajo para que sea considerado como representativo.

Por otra parte si se desea diseñar un nuevo sistema lo normal es que no dispongamos del mismo y, por tanto, difícilmente puede hacérsele trabajar en unas condiciones determinadas de carga.

A este problema existen dos posibles soluciones:

- 1.- La clásica que requiere la elaboración de un prototipo del sistema y
- 2.- La aplicación de sistemas reconfigurables de propósito general pero que actúan emulando al sistema propuesto.

La primera de las alternativas es extraordinariamente costosa ya que hoy en día un nuevo sistema incorpora generalmente una nueva tecnología, al menos en algunas de sus partes. Poner a punto esta tecnología y aplicarla para la elaboración del prototipo puede acarrear una gran parte del coste de desarrollo que tendría el nuevo producto.

La segunda de estas alternativas exige la existencia de estos sistemas reconfigurables (que no obstante, existen. Véase, por ejemplo, el sistema de pruebas de arquitecturas distribuidas presentado en McD082). Pero aun cuando estos sistemas estuvieran disponibles y fueran aplicables quedaría siempre la duda de si el sistema emulado tendría el mismo comportamiento que el sistema

que lo emula (al fin y al cabo, esta segunda alternativa es un híbrido entre los métodos de simulación y la elaboración de un prototipo).

Por último, someter el prototipo o el sistema que lo emula a la condición representativa de carga requiere el desarrollo de un sistema (o driver) que simule esta carga representativa y que puede ser tan complejo como el mismo proceso que se pretende controlar.

2.4 METODOS MARKOVIANOS Y BASADOS EN LA TEORIA DE COLAS.

Los métodos markovianos y basados en la Teoría de Colas han sido dejados para el final de forma intencionada. Estos son métodos de análisis matemático que ofrecen resultados muy interesantes a un nivel de costo y complejidad reducido por lo que son ampliamente utilizados en el diseño y evaluación de sistemas.

Al igual que ocurría en el caso de la simulación al tratar de determinar las prestaciones de un determinado sistema multi-procesador es necesario definir unas condiciones representativas de la carga que el sistema habrá de soportar.

Estas condiciones de carga pueden ser conocidas "a priori" cuando nos referimos a una aplicación concreta [FUNG79, MARK84], sin embargo, cuando lo que se pretende es evaluar estructuralmente un sistema y obtener una estimación de su comportamiento en una aplicación cualquiera (la cual será, en general, distinta

para cada usuario) es necesario establecer una carga genérica representativa frente a la cual poder comparar unos sistemas y otros.

Dada la naturaleza razonablemente aleatoria de las posibles incidencias exteriores sobre el sistema las cuales serán a fin de cuentas, responsables del nivel de carga que el sistema deberá afrontar, resulta lógico plantearse el estudio de las prestaciones que pueden extraerse de la propia estructura de la interconexión del sistema como un estudio de éste frente a unas cargas que tienen una fuerte componente aleatoria.

Cabría ahora preguntarse cuál es la naturaleza de las variables aleatorias que mejor representan un estado de carga genérico del sistema.

Desgraciadamente las herramientas matemáticas (de tipo Markoviano) de que disponemos para nuestro análisis no nos permiten elegir arbitrariamente la naturaleza de estas variables aleatorias.

Tal como se planteará más adelante estas herramientas matemáticas nos obligará en la mayor parte de los casos a considerar variables aleatorias exponencialmente distribuidas (geométricamente distribuidas en el caso discreto) si bien, el actual desarrollo de los elementos de la Teoría de Colas, la aparición de nuevas técnicas tales como el Análisis Operacional y el desarrollo de técnicas de resolución aproximada, aumentan día a día el campo de aplicación de estas técnicas.

La misma pobreza de estas herramientas (en realidad muy potentes si se tiene en cuenta que el problema planteado es de una extraordinaria generalidad y complejidad) tan sólo nos permitirá estudiar los sistemas en su estado estacionario de equilibrio, supuesto que tal estado estacionario exista tal como ocurrirá en los casos que estudiaremos.

Es necesario destacar el hecho de que tal estado estacionario no presupone que el sistema permanezca invariable en él una vez alcanzado. De hecho estos sistemas (salvo casos muy excepcionales) tendrán una evolución dinámica y el estado estacionario o de equilibrio viene definido a partir de las probabilidades estacionarias o de equilibrio de los estados de la cadena de Markov representativa del comportamiento del sistema.

El comportamiento transitorio del sistema es, incluso para casos muy simples, extraordinariamente complejo de analizar y su utilidad, por otra parte, es bastante escasa.

Por último, las técnicas antes mencionadas nos permitirán el cálculo de parámetros globales de tipo medio del sistema tales como número medio de procesadores esperando para recibir servicio o número medio de módulos de memoria ocupados bastante más que su valor en un instante determinado o su evolución con el tiempo.

No es nuestro propósito aquí el presentar un desarrollo exhaustivo sobre los métodos Markovianos y basados en la Teoría de Colas, disciplinas acerca de las cuales existen excelentes tratados [KLEI75, TRIV82], pero sí presentaremos algunos de sus resul-

tados más interesantes así como las técnicas de análisis aproximado que serán utilizadas en capítulos posteriores.

De una manera formal, un conjunto de variables aleatorias X_1, X_2, \dots, X_n constituye una cadena de Markov discreta en el tiempo si cumple la siguiente propiedad:

Para todo n y para todo $i_1 < i_2 < \dots < i_{n+1}$, se tiene:

$$P(X_n=i_n / X_1=i_1, \dots, X_{n-1}=i_{n-1}) = P(X_n=i_n / X_{n-1}=i_{n-1}) \quad (2.1)$$

es decir, el estado de la cadena en cada instante resume la historia anterior del sistema.

De igual forma puede definirse una cadena de Markov continua en el tiempo.

Es fácil demostrar, a partir de su definición, que el tiempo que pasa una cadena de Markov discreta en el tiempo en cada estado está geoméricamente distribuido (en el caso continuo, exponencialmente distribuido), lo que limita en gran medida el uso de estas técnicas.

Es usual el representar los sistemas Markovianos mediante cadenas en las que cada nodo representa un estado del sistema y cada uno de los arcos orientados que une un par de nodos representa la probabilidad de transición de uno a otro estado (velocidades de transición en el caso continuo).

En la figura 2.2 se muestra la cadena de Markov representativa de la clásica cola M/M/1.

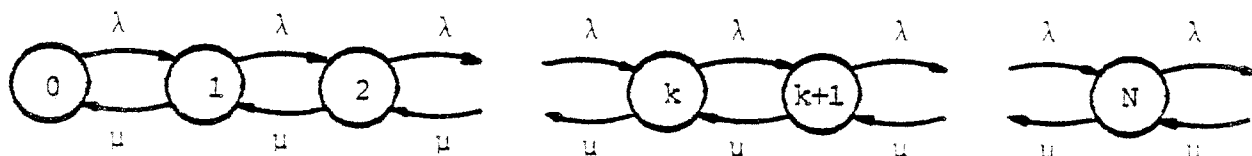


Fig. 2.2

En una cadena de Markov homogénea (las probabilidades de transición entre estados son independientes del tiempo), irreducible (cualquier estado es alcanzable desde cualquier otro en un número finito de pasos) y aperiódica (no existen estados que sólo son alcanzable por el sistema de forma periódica) se puede demostrar que existe el límite:

$$\Pi = \lim_{n \rightarrow \infty} \Pi^{(n)} \quad (2.2)$$

donde $\Pi^{(n)}$ es el vector cuyo elemento $\pi_k^{(n)}$ es la probabilidad de encontrar el sistema en el estado k en el instante n .

Al vector límite Π se le denomina vector de probabilidades estacionarias de los estados de la cadena y puede demostrarse que verifica el sistema de ecuaciones lineales:

$$\sum_i \pi_i = 1 \quad (2.3)$$

$$\pi_j = \sum_i \pi_i p_{ij} \quad (2.4)$$

donde p_{ij} es la probabilidad de transición del estado i al estado j .

Análogas consideraciones pueden realizarse para las cadenas continuas en el tiempo reemplazando las probabilidades de transición entre estados p_{ij} por las velocidades de transición entre estados q_{ij} definidas por:

$$q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(t, t + \Delta t)}{\Delta t} \quad i \neq j \quad (2.5)$$

$$q_{ii} = \lim_{\Delta t \rightarrow 0} \frac{p_{ii}(t, t + \Delta t) - 1}{\Delta t} \quad (2.6)$$

Es fácil ver que el sistema de ecuaciones lineales (2.3)-(2.4) puede ser establecido por simple inspección visual de la cadena representativa del sistema. Bien mediante la formulación de unas ecuaciones de balance entre nodos (estados) de la cadena (ecuaciones de balance local), bien mediante la formulación de estas ecuaciones de balance entre grupos de estados (ecuaciones de balance global).

Una clase especial de sistemas Markovianos denominados procesos de "nacimiento-muerte" ("birth-death" en literatura inglesa) constituyen la base de la Teoría de Colas.

En estos procesos el estado del sistema viene definido por el número de individuos que componen una población y sólo se admiten transiciones entre estados contiguos (desde el estado i sólo son alcanzables en un solo paso los estados $i-1$ e $i+1$).

De una forma más general, redes abiertas de Colas Markovianas han sido resueltas por Jackson. Una red abierta de colas está constituida por N nodos, cada uno de los cuales contiene uno o más servidores exponenciales.

Un cliente que ha obtenido servicio en un nodo puede abandonar el sistema o pasar a otro nodo con una determinada y conocida probabilidad. De igual forma, en cada nodo pueden recibirse clientes del exterior con una distribución de llegadas de Poisson.

Jackson demostró que un tal sistema se comporta como un sistema de N colas independientes, de forma que si k_1, k_2, \dots, k_n es el estado del sistema definido como el número de clientes (k_i) que hay en cada nodo (i), entonces, la probabilidad estacionaria del estado del sistema viene dada por:

$$P(k_1, k_2, \dots, k_n) = P_1(k_1) * P_2(k_2) * \dots * P_n(k_n) \quad (2.7)$$

es decir, las probabilidades estacionarias de los estados del sistema pueden considerarse como el producto de las probabilidades estacionarias de N colas $M/M/m_i$ cuyo comportamiento es bien conocido.

Gordon obtuvo un resultado similar para el caso de redes cerradas de colas Markovianas (ningún cliente puede entrar o salir del sistema) y, posteriormente, el propio Jackson en [JACK63] y Baskett, Chandy, Muntz y Palacios en [BASK75] extendieron este tipo de soluciones, denominadas genéricamente "soluciones en forma de producto" ("product-form solution") a redes más generales de colas.

Es de notar el gran interés que presentan las soluciones de este tipo que permiten abordar sistemas de gran complejidad como un conjunto de sistemas muy simples de solución conocida.

Desgraciadamente este tipo de soluciones no son aplicables sino a un número muy reducido de sistemas (de forma genérica, sistemas con disciplina de colas del tipo "processor-sharing", sistemas con un número de servidores mayores que el número de clientes o sistemas con servidores exponenciales).

Algunos sistemas específicos o muy simples todavía pueden ser abordados de forma exacta mediante métodos tales como el de las "etapas" de England, el método de la cadena de Markov "embebida" o el método de la variable "suplementaria" que permiten tratar sistemas con servidores (o distribuciones de llegadas "entre" clientes) no exponencialmente distribuidas.

Para el estudio de otro tipo de sistemas no hay más remedio que acudir a soluciones de tipo aproximado.

Uno de los métodos de análisis aproximado más interesantes se basa en el "teorema de descomposición" de Chandy, Woo y Herzog presentado en [CHAND75].

Este método (análogo al teorema de Norton en redes eléctricas) permite reemplazar una subred dentro del sistema por un único servidor equivalente cuyas características pueden determinarse por aplicación de este método.

El método presupone que la subred reemplazada se comporta de cara al sistema exterior tal como si se encontrara permanentemente en estado estacionario y se ha demostrado que conduce a resultados exactos en aquellos sistemas que admiten una solución en forma de producto, lo que permite simplificar la resolución de estos sistemas.

No obstante, la aplicación de este método a un sistema cualquiera conduce a soluciones aproximadas que pueden ser de gran interés.

Otra aproximación muy utilizada es la denominada de "difusión" [KOB74], aplicada sobre todo en sistemas con servidores no exponenciales y con un elevado tráfico, si bien la aplicación de este método a nuevos problemas requiere un complejo tratamiento matemático no siempre recompensado por los resultados obtenidos.

Otras soluciones aproximadas de tipo iterativo o una mezcla de las aproximaciones antes citadas han sido también utilizadas para el análisis de sistemas muy complejos y ejemplos de su aplicación serán vistos en los capítulos siguientes.

CAPITULO 3.

SISTEMAS "CROSS-BAR".

3.1 INTRODUCCION.

En este capítulo estudiaremos los sistemas interconectados a través de una red del tipo "cross-bar" analizando los efectos que sobre la efectividad del sistema originan los conflictos derivados del uso de recursos compartidos.

En el caso discreto supondremos que los procesadores y memorias trabajan sincronizados de forma que tanto la duración de los tiempos de procesamiento de los procesadores como los tiempos de ciclo de las memorias tan sólo pueden tomar valores discretos en la forma $t_c, 2t_c, \dots, kt_c, \dots$.

En el caso continuo asumiremos que estos tiempos tienen una cierta distribución continua en el tiempo.

Intentaremos modelar el sistema mediante cadenas de Markov exactas o aproximadas. Para cada una de estas cadenas intentaremos obtener las probabilidades estacionarias de cada uno de sus estados. Estas probabilidades serán obtenidas en forma exacta (en los casos en que sea posible) mediante la formulación de las ecuaciones locales o globales de balance de flujo y resolviendo el sistema de ecuaciones lineales a que da lugar estos balances

de flujo (Apartado 2.4).

En aquellos sistemas cuya complejidad desaconseje su resolución exacta estudiaremos las aproximaciones propuestas por diferentes autores o propondremos nuevas aproximaciones, comparándolas con resultados de simulación para establecer su validez.

Conviene resaltar el notable interés que presentan aquellas aproximaciones que conducen a expresiones explícitas de las magnitudes principales, por su gran simplicidad de cálculo frente a la siempre más engorrosa resolución del sistema de ecuaciones lineales a que da lugar la formulación de las ecuaciones de balance en una cadena de Markov. No obstante veremos que, en general, este tipo de soluciones explícitas suelen tener un error bastante elevado.

3.2 ANALISIS DISCRETO DE LOS SISTEMAS "CROSS-BAR".

Un sistema de N procesadores (computadores o microprocesadores) y M módulos de memorias interconectados en la forma representada en la figura 3.1 recibe el nombre de "Sistema Completamente Interconectado" o "Crossbar System" en la literatura inglesa, y será denominado de ahora en adelante en la forma abreviada $N \times M$.

Su más directo exponente lo constituye el sistema desarrollado en la Universidad de Carnegie-Mellon (Pittsburgh, Pennsylvania) y bautizado con el nombre de C.mmp [WULF72]. Este mul-

tiprocesador consta en la versión inicial y en su módulo primario de hasta 16 procesadores del tipo PDP-11 y hasta 16 módulos de memoria, cada una de las cuales tiene una capacidad de 130 Kbytes.

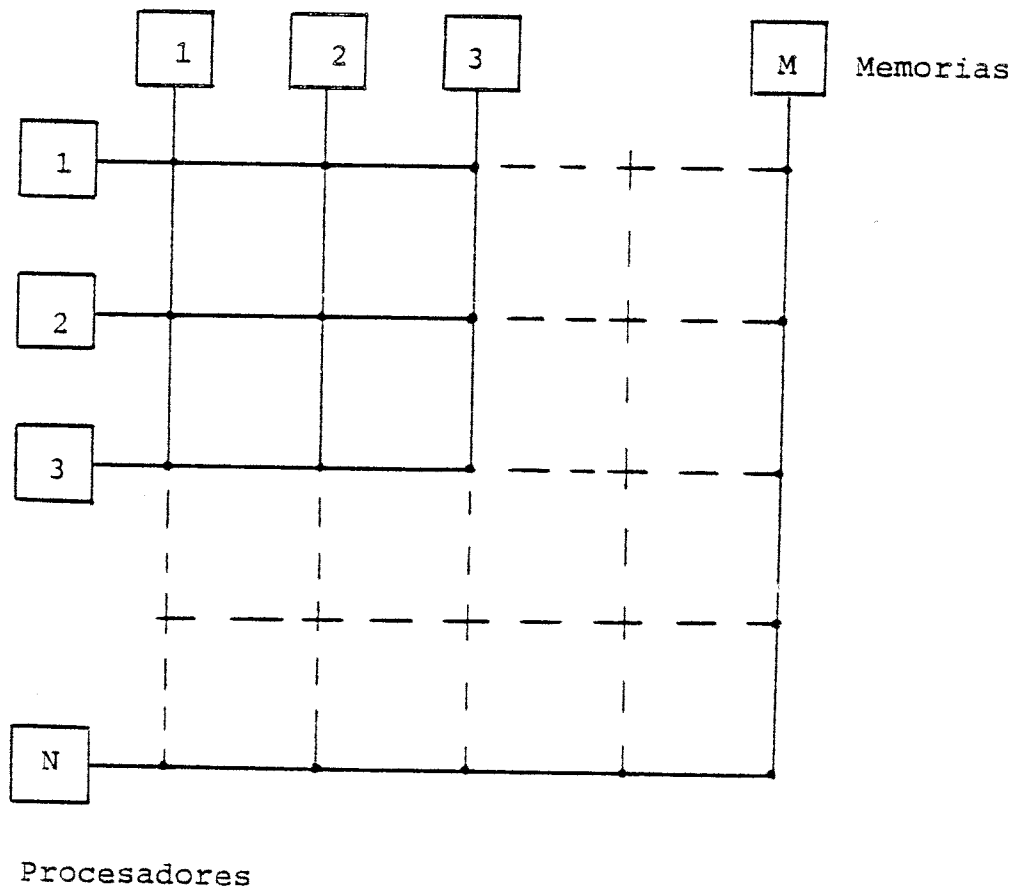


Fig. 3.1 Sistema Completamente Interconectado.

En este tipo de sistemas todos los procesadores pueden acceder a través una red de interconexión a cualquiera de los módulos de memoria. Para resolver el conflicto que puede originarse como consecuencia de que varios procesadores intenten acceder simultáneamente a un mismo módulo de memoria, se dispone delante un selector ("Crossbar Switch") que, atendiendo a una determinada política de arbitraje, tan sólo permite el acceso de uno de los

procesadores quedando los restantes en cola de espera hasta que el módulo le sea asignado.

3.2.1 Modelo Discreto de un Sistema Completamente Interconectado.

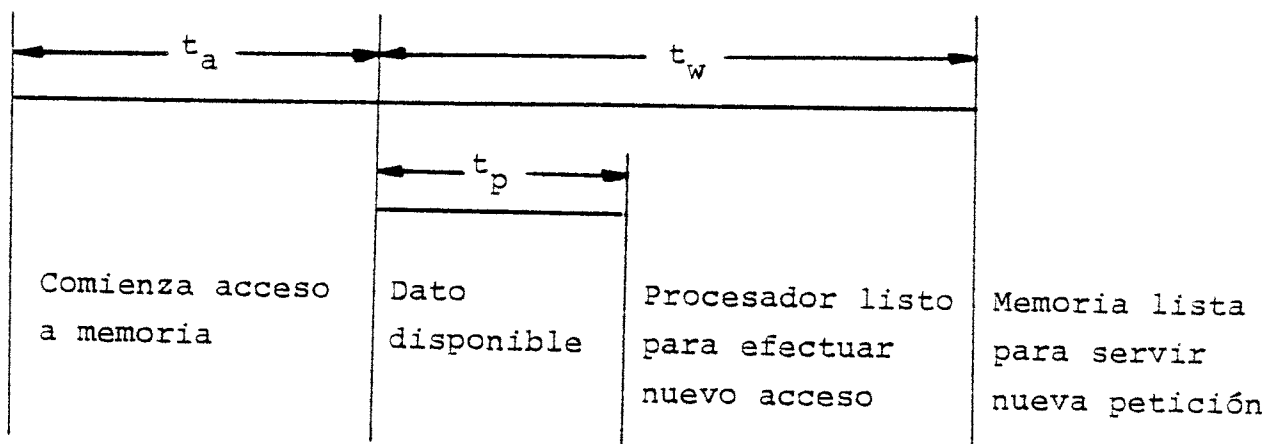
Para obtener un modelo de este tipo de sistemas hemos de tener en cuenta que, en el caso más general, los procesadores y las memorias no estarán necesariamente sincronizados entre sí, siendo el diagrama de tiempos de una instrucción bastante complejo. Este diagrama de tiempos será, por otra parte, distinto para cada tipo de instrucción (es claro que una operación aritmética con el acumulador tendrá un diagrama de tiempos bastante diferente al de una instrucción de lectura de un dato de una posición de memoria. De igual forma, una instrucción de lectura será, en general, muy diferente a una operación de escritura). Cada procesador tendrá asimismo un secuencia de accesos ("access pattern") que será función del tipo de programa que se encuentre ejecutando y que será, en general, distinto en cada procesador. De esta misma forma podríamos considerar otras circunstancias que hacen muy complicado establecer un modelo general para estos sistemas.

Por tanto, para poder establecer su análisis, será necesario introducir algunas simplificaciones las cuales presentaremos a continuación.

En primer lugar, despreciaremos el efecto que las operaciones de entrada-salida (E/S) tienen sobre el comportamiento del sistema. De hecho, Strecker demostró que el efecto de estas operaciones puede ser modelado como una fracción de los accesos to-

tales a memoria [BHAN75]. De forma más general aún la mayor parte de los autores consideran que el efecto debido a las operaciones de E/S es despreciable y por ello no lo consideran.

Adoptaremos el diagrama simplificado de tiempos de instrucción que se muestra en la figura 3.2, no haciendo distinción entre uno u otro tipo de instrucciones ejecutadas por los procesadores considerando que todas dan lugar al mismo diagrama de tiempos.



t_a = tiempo de acceso a memoria.

t_w = tiempo de recuperación de la memoria ("rewrite time") y

t_p = tiempo de procesamiento del procesador.

Fig. 3.2 Diagrama de Tiempos de Instrucción.

Todos los procesadores se consideran idénticos con igual tiempo de procesamiento t_p y todas las memorias se consideran idénticas con igual tiempo de ciclo t_c ($t_c = t_a + t_w$).

Los tiempos de propagación y arbitraje asociados con la red de interconexión no son considerados específicamente, si bien puede considerarse que forman parte del tiempo de acceso t_a .

Se considera asimismo que los tiempos de procesamiento de los procesadores (t_p) y los tiempos de recuperación de las memorias (t_w) son constantes en cada ciclo y, para mayor simplicidad, iguales. Asumiremos también que las memorias están perfectamente sincronizadas.

Si al comienzo de un ciclo hay k procesadores que requieren acceso a un mismo módulo de memoria j , uno cualquiera de ellos es seleccionado aleatoriamente con probabilidad $1/k$ para ser servido en este ciclo quedando los $k - 1$ restantes bloqueados en la cola de la memoria j .

Por último, admitiremos que la secuencia de accesos de un procesador a cada una de las memorias es una secuencia de pruebas de Bernouilli, de forma que, en cada ciclo, la probabilidad de que el procesador i acceda a la memoria j viene dada por $x_{ij} = 1/M$ para cualquier i y cualquier j .

3.2.2 Parámetros del Sistema.

El parámetro que utilizaremos para evaluar las prestaciones del sistema será el ancho de banda B de las memorias, definido como el número medio de memorias ocupadas por ciclo.

El número medio de procesadores esperando en cola N_q será igual a $N - B$.

3.2.3 Modelo Exacto.

Con las simplificaciones del apartado anterior el sistema puede ser modelado mediante una cadena discreta de Markov cuyo estado es caracterizado en cada ciclo por un vector de M dimensiones (k_1, k_2, \dots, k_M) donde $\sum_{i=1}^M k_i = N$ con $0 \leq k_i \leq N$, $i = 1, 2, \dots, M$ y donde cada elemento k_i de este vector representa el número de procesadores que se encuentran en cola de la memoria i en el ciclo considerado.

El número de estados de la cadena viene dado por todas las formas en que N procesadores indistinguibles pueden repartirse entre M memorias y que resulta ser:

$$\binom{N + M - 1}{M - 1} \quad (3.1)$$

Las probabilidades de transición entre estados de la cadena pueden evaluarse en la forma siguiente establecida por Bhandarkar [BHAN73] y [BHAN75]:

A cada estado $K (k_1, k_2, \dots, k_M)$ se le asocia un estado parcial (j_1, j_2, \dots, j_M) definido por:

$$j_i = \max(0, k_{i-1}) ; i = 1, 2, \dots, M \quad (3.2)$$

Un nuevo estado $L (l_1, l_2, \dots, l_M)$ será alcanzable en un único paso desde el estado $K (k_1, k_2, \dots, k_M)$ si y sólo si $l_i \geq j_i$ para $i = 1, 2, \dots, M$, en cuyo caso, la probabilidad de transición del estado K al estado L vendrá dada por:.

$$P_{KL} = \frac{x!}{d_1! d_2! \dots d_M!} \left(\frac{1}{M} \right)^x \quad (3.3)$$

donde $d_i = l_i - j_i$ para $i = 1, 2, \dots, M$ y $x = \sum_{i=1}^M d_i$

Esta fórmula se explica en la forma siguiente: si x es el número de elementos no nulos en el vector $K (k_1, k_2, \dots, k_M)$ entonces al final del presente ciclo un número de x procesadores habrán sido servidos y realizarán un nuevo acceso en el ciclo siguiente.

En la cola de la memoria i ésima quedarán $j_i = \max(k_{i-1}, 0)$ procesadores no servidos en el ciclo anterior. El nuevo estado L será alcanzable sólo si la longitud l_i de la cola de la memoria i ésima correspondiente a este estado es mayor o igual que j_i y, en este caso, la probabilidad de que los x nuevos accesos se distribuyan de forma que $d_1 = l_1 - j_1$ sean dirigidos a la memoria 1, $d_2 = l_2 - j_2$ a la memoria 2, ..., $d_M = l_M - j_M$ a la memoria M viene dada por la fórmula (3.3)

El que esta cadena tiene solución única es evidente sin necesidad de demostración ya que cumple la propiedad Markoviana (la elección del próximo estado sólo depende del estado actual), es

Un nuevo estado $L (l_1, l_2, \dots, l_M)$ será alcanzable en un único paso desde el estado $K (k_1, k_2, \dots, k_M)$ si y sólo si $l_i \geq j_i$ para $i = 1, 2, \dots, M$, en cuyo caso, la probabilidad de transición del estado K al estado L vendrá dada por:.

$$P_{KL} = \frac{x!}{d_1! d_2! \dots d_M!} \left(\frac{1}{M} \right)^x \quad (3.3)$$

donde $d_i = l_i - j_i$ para $i = 1, 2, \dots, M$ y $x = \sum_{i=1}^M d_i$

Esta fórmula se explica en la forma siguiente: si x es el número de elementos no nulos en el vector $K (k_1, k_2, \dots, k_M)$ entonces al final del presente ciclo un número de x procesadores habrán sido servidos y realizarán un nuevo acceso en el ciclo siguiente.

En la cola de la memoria i ésima quedarán $j_i = \max(k_{i-1}, 0)$ procesadores no servidos en el ciclo anterior. El nuevo estado L será alcanzable sólo si la longitud l_i de la cola de la memoria i ésima correspondiente a este estado es mayor o igual que j_i y, en este caso, la probabilidad de que los x nuevos accesos se distribuyan de forma que $d_1 = l_1 - j_1$ sean dirigidos a la memoria 1, $d_2 = l_2 - j_2$ a la memoria 2, ..., $d_M = l_M - j_M$ a la memoria M viene dada por la fórmula (3.3)

El que esta cadena tiene solución única es evidente sin necesidad de demostración ya que cumple la propiedad Markoviana (la elección del próximo estado sólo depende del estado actual), es

aperiódica (cualquier estado puede evolucionar hacia él mismo) y es irreducible (cualquier estado puede ser alcanzado desde cualquier otro en un número finito de pasos). En general, todas las cadenas que veremos tienen solución única y este mismo argumento será aplicable a todas ellas por lo que no será repetido.

La resolución exacta de este modelo fue primeramente abordada por Skinner y Asher [SKIN69].

Estos autores consideraron una definición del estado de la cadena más compleja. El estado de la cadena queda definido, en cada ciclo, indicando el módulo de la memoria que se encuentra referenciando cada procesador. El número de tales estados es extraordinariamente elevado incluso para valores muy reducidos de N y M por lo que los autores redujeron su estudio a los casos más simples ($N, M \leq 2$).

Considerando tal como se indicó en el apartado 3.2.1 los procesadores indistinguibles, y utilizando la definición de estado dada anteriormente, Bhandarkar [BHAN73 y BHAN75] fue capaz de resolver el sistema para valores más elevados de N y M ($N, M \leq 8$) haciendo uso de la evidente simetría que presenta el modelo respecto de procesadores y memorias.

Dado que cada uno de los procesadores y cada una de las memorias son exactamente iguales, el estado $X_1 (x_{11}, x_{12}, \dots, x_{1M})$ y cualquier estado X_2 cuyos elementos $(x_{21}, x_{22}, \dots, x_{2M})$ sean una permutación de los elementos de X_1 tienen exactamente la misma probabilidad y pueden ser, por tanto, englobados en un único ma-

croestado X . Denominaremos "estado representativo" del macroestado X al estado $X_1 (x_1, x_2, \dots, x_M) \in X$ tal que $x_1 \geq x_2 \geq \dots \geq x_M$.

Sea $S_X = (X_1, X_2, \dots, X_a)$ el conjunto de estados que constituyen el macroestado X y sea $S_Y = (Y_1, Y_2, \dots, Y_b)$ el conjunto de estados que constituyen el macroestado Y , entonces la probabilidad de alcanzar el macroestado Y desde el macroestado X vendrá dada por

$$P_{XY} = \sum_{Y_i \in S_Y} P(\text{trans. del estado } X_j \text{ al estado } Y_i)$$

y esto para cualquier $X_j \in S_X$.

N	numero estados cadena original	numero estados cadena simplificada
2	3	2
4	35	5
8	6435	22
10	92378	42
12	1352078	77
16	300540195	231

Tabla 3.1 Numero de estados para $N = M$.

Consideraremos a partir de ahora la cadena reducida cuyos estados son los macroestados definidos anteriormente. El número de estados de esta nueva cadena es muy reducido en comparación con el número de estados de la cadena original (tabla 3.1). El número de tales estados viene dado por la expresión:

$$\min(N, M) \sum_{i=1}^{\min(N, M)} p_i(N) \quad (3.4)$$

donde $p_i(N)$ es el número de particiones no ordenadas de N en i partes y se define en la forma recurrente siguiente:

$$p_i(N) = p_i(N-1) + p_{i-1}(N-1) + \dots + p_1(N-1) + p_0(N-1)$$

$$\begin{aligned} \text{Con } p_i(N) &= 0 \text{ si } N < i \text{ ó } i < 0 \\ p_0(N) &= 0 \text{ si } N > 0 \\ p_i(i) &= 1 \text{ si } i \geq 0 \end{aligned} \tag{3.5}$$

No es posible encontrar una expresión explícita para la suma anterior pero se conoce que su valor es asintótico a la expresión

$$\frac{1}{4\pi\sqrt{3}} e^{\sqrt{\frac{2N}{3}}\pi} \tag{3.6}$$

Bhandarkar reconoce que no fue capaz de derivar una fórmula directa para evaluar las probabilidades de transición entre cada uno de estos macroestados, proponiendo un algoritmo de cálculo basado en un árbol de enumeración que se construye como sigue:

Consideremos en un sistema 4×4 el macroestado cuyo estado representativo es $(2,2,0,0)$ y cuyo estado parcial correspondiente será $(1,1,0,0)$. Si el sistema se encontrara en este estado, al final del ciclo quedaría un procesador en las colas de las memorias M_1 y M_2 y 2 procesadores habrían sido servidos teniendo que ser asignados en el ciclo siguiente.

Añadamos un único procesador a cada una de las colas. De esta forma, obtenemos los estados intermedios A_1 , A_2 , B_1 y B_2 .

$$\begin{array}{ll} A_1 : (2,1,0,0) & A_2 : (1,2,0,0) \\ B_1 : (1,1,1,0) & B_2 : (1,1,0,1) \end{array}$$

Los estados intermedios A_1 y A_2 por un lado y B_1 y B_2 por otro son equivalentes dado que los elementos de uno son permutaciones de los elementos del otro y pueden ser agrupados en los macroestados intermedios A $(2,1,0,0)$ y B $(1,1,1,0)$ respectivamente. Así pues, desde el estado parcial $(1,1,0,0)$ y añadiendo un único procesador, hay dos caminos que nos permiten alcanzar cada uno de los dos macroestados intermedios A y B, lo que se representa en la forma siguiente:

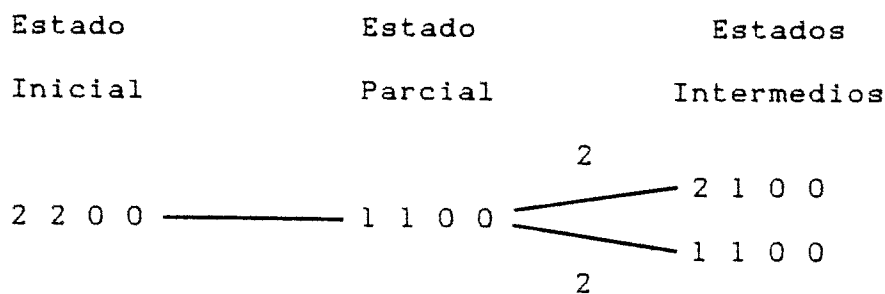


Fig. 3.3

El proceso se repite añadiendo de cada vez un único procesador hasta que todos han sido asignados. En nuestro caso sólo hay un procesador más por asignar. Operando en la misma forma para añadir el último procesador nos queda el siguiente árbol de enu-

enumeración para el estado $(2,2,0,0)$:

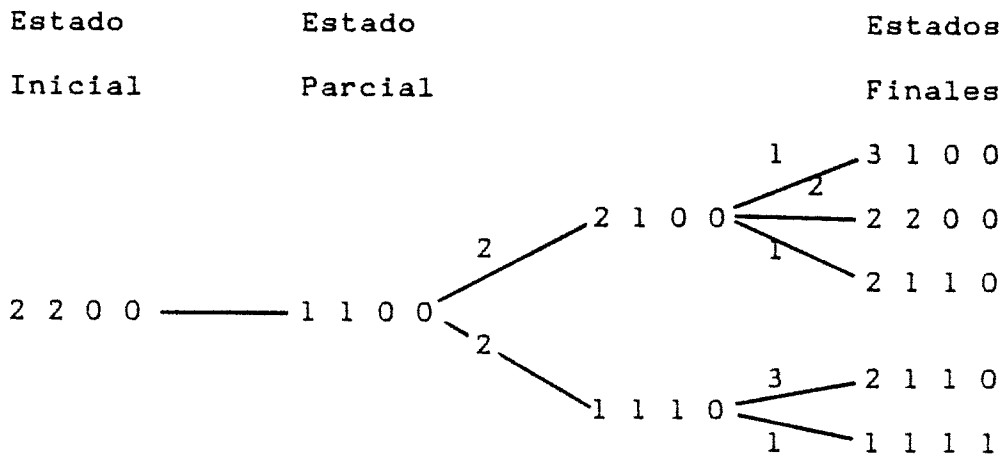


Fig. 3.4 Arbol de enumeración para el estado $(2,2,0,0)$.

La probabilidad de alcanzar un estado final a partir del estado inicial viene dada por el número de caminos diferentes en el árbol que conducen del estado inicial al estado final dividido por el número total de caminos del árbol.

Conocidas las probabilidades de transición entre los estados puede resolverse el sistema de ecuaciones lineales que nos permite obtener las probabilidades estacionarias de cada estado y, a partir de ellas, el Ancho de Banda del Sistema.

Es fácil ver que, a pesar de las simplificaciones introducidas, la resolución del modelo sigue siendo muy compleja ya que para valores elevados de N y M el número de estados crece según (3.6), lo que ha movido a diferentes autores a proponer modelos más simples que aproximen los resultados exactos.

Especialmente significativos han sido los modelos de Streck-ker, Baskett y Smith [BAS76] y Rau [RAU79]

3.2.4 Aproximación de STRECKER.

Strecker asume que los procesadores que no han sido servidos en el presente ciclo, son desbloqueados al final del ciclo, de forma que vuelven a realizar una nueva petición de acceso en el ciclo siguiente. De esta forma el estado del sistema en un ciclo cualquiera es independiente de los estados alcanzados en los ciclos anteriores.

Nótese que esta aproximación sobreestimaré el ancho de banda de las memorias, ya que un procesador bloqueado en la cola de una memoria puede efectuar una petición a otra memoria cualquiera en el siguiente ciclo. (En el modelo exacto un procesador bloqueado en la cola de la memoria i volverá a efectuar una petición de acceso a la memoria i en el ciclo siguiente y no a ninguna otra).

Con esta aproximación, la probabilidad estacionaria de que la longitud L_j de la cola de una memoria cualquiera M_j sea d , vendrá dada por:

$$P [L_j = d] = \binom{N}{d} (1/M)^d (1 - 1/M)^{N-d} \quad 0 \leq d \leq N \quad (3.7)$$

y la probabilidad de que una memoria esté ocupada será:

$$P [\text{mem. } j \text{ ocupada}] = 1 - P [L_j = 0] = 1 - (1 - 1/M)^N \quad (3.8)$$

de donde, el ancho de banda tendrá la expresión:

$$B = M [1 - (1 - 1/M)^N] \quad (3.9)$$

Ravi [RAVI72] llega a idéntico resultado (como se demuestra en Chang [CHANG77]), si bien, su fórmula es bastante más compleja de cálculo.

Esta solución aproximada por Strecker resulta ser bastante segura, sobre todo para valores de $M > N$ y para valores bajos de M y N lo que es lógico ya que, en estos casos, la probabilidad de conflicto disminuye y es menor el efecto que tiene la simplificación de Strecker.

Dado que los resultados exactos obtenidos por Bhandarkar (tabla 3.2) son prácticamente simétricos respecto de N y M , y la fórmula de Strecker aproxima mejor los sistemas con $M > N$, el propio Bhandarkar propone una modificación de la fórmula de Strecker para hacerla simétrica:

$$B = i [1 - (1 - 1/i)^j] \quad (3.10)$$

donde $i = \max (N,M)$ y $j = \min (N,M)$

la cual da mejores resultados que la fórmula original.

3.2.5 Aproximación Binomial.

Baskett y Smith [BASK76] estudian el comportamiento asintótico del modelo ($N \rightarrow \infty$, $M \rightarrow \infty$, $N/M \rightarrow L$) utilizando resultados conocidos de la teoría de colas.

La longitud media de la cola en un sistema M/G/1 (llegadas de Poisson, distribución arbitraria de los tiempos de servicio, 1 servidor) viene dada [KLEI75] por:

$$L = L_0 \frac{A''(1) + 2 A'(1) - 2A''(1)}{2 - 4 A'(1) + 2 A'^2(1)} \quad (3.11)$$

donde L_0 es la fracción de tiempo durante la cual la cola permanece vacía y $A(z)$ es la función generadora en z de la distribución de las llegadas

$$A(z) = \sum_{j=0}^{\infty} A_j z^j \quad (3.12)$$

siendo $A_j = \text{prob. (j elementos lleguen al terminar el intervalo)}$.

En el caso de N y M finitos y conocido el número b de servidores (memorias) ocupados, las llegadas a una cola son binomialmente distribuidas en la forma

$$P [d \text{ llegadas}] = \binom{b}{d} (1/M)^d (1 - 1/M)^{b-d} \quad 0 \leq d \leq b \quad (3.13)$$

Cuando $N \rightarrow \infty$, $M \rightarrow \infty$, el número de servidores ocupados será una fracción constante $\rho = 1 - L_0$ del número de memorias. La distribución binomial en el límite pasa a ser la distribución de Poisson y, por tanto:

$$P [d \text{ llegadas}] = \rho^d e^{-\rho} / d! \quad (3.14)$$

cuya función generadora es $A(z) = e^{\rho(z-1)}$.

Es decir, para $N \rightarrow \infty$, $M \rightarrow \infty$ nuestro sistema se convierte en M sistemas de colas $M/G/1$, por lo que aplicando (3.11) para cada una de ellas, la longitud media de la cola vendrá dada por

$$L = \rho + \frac{\rho^2}{2(1-\rho)} \quad (3.15)$$

Ahora bien, la longitud media de la cola es conocida ya que, en el límite, $N/M \rightarrow L$. De (3.15)

$$\rho = 1 + \frac{N}{M} - \sqrt{\left(\frac{N}{M}\right)^2 + 1} \quad (3.16)$$

y el ancho de banda

$$B = M \rho = M + N - M \sqrt{(N/M)^2 + 1} \quad (3.17)$$

Para el caso en que M y N sean finitos podemos considerar en primera aproximación que la función generadora de las llegadas a cada cola corresponde a la de una distribución binomial asumiendo que el número de memorias ocupadas en cada ciclo es constante e igual a n, entonces

$$P [k llegadas] = A_k = \binom{n}{k} (1/M)^k (1 - 1/M)^{n-k} \quad (3.18)$$

sustituyendo en (3.11) y despejando

$$\rho = 1 - L_0 = \frac{(2L + 2 - 1/M) - \sqrt{(1/M - 2 - 2L)^2 - 8L}}{2} \quad (3.19)$$

Asumiendo que la longitud media de la cola viene dada por $L = N/M$, el ancho de banda vendrá dado por:

$$B = M \rho = N + M - 0.5 - \sqrt{(N + M - 0.5)^2 - 2 N M} \quad (3.20)$$

Esta aproximación, denominada "aproximación binomial", da unos resultados excelentes para valores de N y M muy elevados lo que resulta lógico, ya que, como sabemos, el modelo es asintóticamente exacto.

3.2.6 Aproximación de Descomposición.

Esta aproximación fue originalmente propuesta por Baskett y Smith en BASK76, si bien estos autores la desecharon por considerarla de gran complejidad siendo posteriormente desarrollada por Rau en RAU79. Esta aproximación consiste en suponer que todos los procesadores (en número de K) de un sistema $N \times M$ que no se encuentran en la cola de una memoria dadas en un ciclo dado, se distribuyen entre las $M - 1$ memorias restantes tal como lo harían K procesadores en un sistema $K \times (M - 1)$.

Esta suposición se expresa analíticamente en la forma:

$$P_{MN} (k_1, k_2, \dots, k_{M-1}, k_M / k_M = N - K) = P_{M-1, K} (k_1, k_2, \dots, k_{M-1}) \quad (3.21)$$

siendo $P_{MN} (k_1, k_2, \dots, k_{M-1}, k_M)$ la probabilidad estacionaria del estado $(k_1, k_2, \dots, k_{M-1}, k_M)$ en un sistema $N \times M$.

Si ahora se define la probabilidad normalizada del estado $(k_1, k_2, \dots, k_{M-1}, k_M)$, como

$$\hat{P}_{M,N} (k_1, k_2, \dots, k_{M-1}, k_M) = \frac{P_{M,N} (k_1, k_2, \dots, k_{M-1}, k_M)}{P_{M,N} (N, 0, \dots, 0)} \quad (3.22)$$

aplicando (3.21) y (3.22)

$$\frac{P_{M,N}(k_1, k_2, \dots, k_{M-1}, N-K)}{P_{M,N}(K, 0, \dots, 0, N-K)} = \frac{P_{M-1,K}(k_1, k_2, \dots, k_{M-1})}{P_{M-1,K}(K, 0, \dots, 0)} = \hat{P}_{M-1,K}(k_1, k_2, \dots, k_{M-1}) \quad (3.23)$$

y, por tanto,

$$\hat{P}_{M,N}(k_1, k_2, \dots, k_{M-1}, N-K) = \hat{P}_{M-1,K}(k_1, k_2, \dots, k_{M-1}) \times \hat{P}_{M,N}(K, 0, \dots, 0, N-K) \quad (3.24)$$

Aplicando de forma recurrente este resultado

$$\begin{aligned} \hat{P}_{M,N}(k_1, k_2, \dots, k_{M-1}, k_M) &= \hat{P}_{M-1, N-k_M}(k_1, k_2, \dots, k_{M-1}) \times \\ \hat{P}_{M,N}(N-k_M, 0, \dots, 0, k_M) &= \hat{P}_{M-2, N-k_M-k_{M-1}}(k_1, k_2, \dots, k_{M-2}) \times \\ \hat{P}_{M-1, N-k_M} &(N-k_M-k_{M-1}, 0, \dots, 0, k_{M-1}) \times \hat{P}_{M,N}(N-k_M, 0, \dots, 0, k_M) \\ &= \dots = \prod_{i=2}^M \hat{P}_{i, S_i+k_i}(S_i, 0, \dots, 0, k_i) \end{aligned} \quad (3.25)$$

$$\text{donde } S_i = N - \sum_{j=i}^M k_j. \quad (3.26)$$

dado que aplicando (3.21)

$$\hat{p}_{i, S_i + K_i}(S_i, 0, \dots, 0, k_i) = \hat{p}_{2, S_i + k_i}(S_i, k_i)$$

se deduce en consecuencia que el estudio del sistema $N \times M$ se reduce al estudio de los sistemas $K \times 2$ ($K = 1, 2, \dots, N$). Ahora bien, un sistema de K procesadores y 2 memorias, es un sistema fácilmente resoluble respecto al cual es fácil demostrar que:

$$\hat{p}_{2, K}(K - k_1, k_1) = \begin{cases} 2 & \text{si } k_1 \neq 0, K \text{ y } K \neq 0 \\ 1 & \text{en caso contrario} \end{cases} \quad (3.27)$$

$$\text{y, por tanto, } \hat{p}_{M, N}(k_1, k_2, \dots, k_{M-1}, k_M) = 2^K - 1 \quad (3.28)$$

donde K es el número de elementos no nulos en el vector (k_1, k_2, \dots, k_M) .

El número de formas de seleccionar M memorias de forma que K estén ocupadas es $\binom{M}{K}$ y el número de formas de repartir N procesadores indistinguibles entre K módulos viene dado por $\binom{N-1}{K-1}$, por lo que el factor Π de normalización de las probabilidades estacionarias será

$$\Pi = 1 / \left(\sum_{k=1}^M 2^{k-1} \binom{M}{k} \binom{N-1}{k-1} \right) \quad (3.29)$$

y el ancho de banda

$$B = \sum_{k=1}^M k \Pi 2^{k-1} \binom{M}{k} \binom{N-1}{k-1} \quad (3.30)$$

Sustituyendo el valor de Π y definiendo $\binom{x}{y} = 0$ para $y > x$, se tiene

$$B = \frac{\sum_{i=0}^{j-1} 2^i \binom{M-1}{i} \binom{N-1}{i}}{\sum_{i=0}^{j-1} \frac{2^i}{i+1} \binom{M-1}{i} \binom{N-1}{i}} \quad (3.31)$$

$$j = \min(M, N)$$

Esta expresión aproxima extraordinariamente los resultados exactos sobre todo para valores elevados de N y M .

3.2.7 Comparación de las diferentes aproximaciones.

En la tabla 3.2 se muestran los resultados exactos obtenidos por Bhandarkar para sistemas $N \times N$ ($N, M \leq 8$) y en las tablas 3.3 y 3.4 se muestran las diferencias en tanto por ciento relativas al valor exacto de las aproximaciones de Strecker mejorada por Bhandarkar y de la aproximación de descomposición.

En estas tablas puede observarse la gran precisión obtenida con la aproximación de descomposición cuya mayor diferencia respecto de los resultados exactos no supera el 0.25%. Por el contrario, el modelo de Strecker mejorado por Bhandarkar presenta errores importantes a medida que aumentan M y N .

Para estos valores tan bajos de N y M la aproximación binomial tiene un error muy grande.

M	1	2	3	4	5	6	7	8
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.5000	1.6667	1.7500	1.8000	1.8333	1.8571	1.8750
3	1.0000	1.6667	2.0476	2.2692	2.4095	2.5054	2.5748	2.6272
4	1.0000	1.7500	2.2701	2.6210	2.8630	3.0365	3.1657	3.2652
5	1.0000	1.8000	2.4102	2.8633	3.1996	3.4530	3.6482	3.8019
6	1.0000	1.8333	2.5059	3.0370	3.4533	3.7809	4.0415	4.2518
7	1.0000	1.8571	2.5751	3.1663	3.6486	4.0418	4.3636	4.6292
8	1.0000	1.8750	2.6274	3.2657	3.8024	4.2521	4.6294	4.9471

Tabla 3.2. Resultados Exactos

M	1	2	3	4	5	6	7	8
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	-0.002	0.000	0.000	0.002	0.002	0.000
3	0.000	-0.002	3.102	1.908	1.266	0.893	0.662	0.511
4	0.000	0.000	1.868	4.326	3.109	2.305	1.765	1.389
5	0.000	0.000	1.236	3.098	5.063	3.931	3.102	2.494
6	0.000	0.002	0.873	2.288	3.922	5.547	4.516	3.712
7	0.000	0.002	0.650	1.746	3.090	4.508	5.889	4.952
8	0.000	0.000	0.503	1.373	2.481	3.705	4.947	6.146

Tabla 3.3. Aproximación de Strecker mejorada por Bhandarkar.

M	1	2	3	4	5	6	7	8
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	-0.002	0.000	0.000	0.002	0.002	0.000
3	0.000	-0.002	0.246	0.155	0.094	0.058	0.037	0.027
4	0.000	0.000	0.116	0.153	0.128	0.098	0.075	0.057
5	0.000	0.000	0.065	0.118	0.125	0.112	0.092	0.075
6	0.000	0.002	0.038	0.081	0.103	0.105	0.097	0.083
7	0.000	0.002	0.026	0.056	0.081	0.089	0.090	0.084
8	0.000	0.000	0.019	0.041	0.062	0.076	0.080	0.080

Tabla 3.4. Aproximación de Descomposición.

En la tabla 3.5 se comparan las diferentes aproximaciones para N y M elevados con resultados procedentes de simulación. En esta tabla puede observarse el buen comportamiento de la aproximación binomial a medida que N y M aumentan.

N	M	Simulacion	Aprox de Strecker	Aprox Binomial	Aprox de Descomp
12	8	5.813	6.018	5.578	5.818
16	8	6.313	6.453	6.112	6.317
20	8	6.632	6.732	6.459	6.635
24	8	6.850	6.926	6.702	6.854
8	16	6.315	6.453	6.112	6.317
12	16	8.253	8.625	8.000	8.247
16	16	9.632	10.303	9.373	9.630

Tabla 3.5.

3.3 SISTEMAS CON PROBABILIDAD m DE ACCESO POR CICLO.

Un paso hacia modelos más reales se obtiene suponiendo que los procesadores no realizan accesos en todos los ciclos.

En los modelos que veremos a continuación, cada procesador tiene una probabilidad m de efectuar un acceso a memoria en cada ciclo. Una vez que un procesador ha efectuado una petición de acceso, el comportamiento del modelo es exactamente igual al visto para el caso $m = 1$. Si la memoria referenciada está libre el pro-

cesador accede a ella durante un ciclo. Una vez que ha sido servido el procesador vuelve a tener una probabilidad m de efectuar un nuevo acceso. Si la memoria es referenciada por varios procesadores simultáneamente se concede el acceso a uno cualquiera de ellos con igual probabilidad permaneciendo el resto bloqueado en la cola de la memoria.

Con este modelo pueden ser representados sistemas en que los procesadores disponen de una zona de memoria privada a la cual acceden con probabilidad $(1 - m)$ por ciclo y cuyo acceso no introduce ningún tipo de conflicto.

De igual modo estos modelos permiten tener en cuenta la existencia de instrucciones que no requieren acceso a memoria común excepto, quizás, para leer el código de la propia instrucción (por ejemplo, operaciones con acumuladores).

Por último, estos modelos han sido también utilizados para representar sistemas "cross-bar" en los que se tiene $t_p > t_w$ si asumimos que, en cada ciclo, el tiempo de procesamiento t_p varía sólo en forma discreta ($t_p = t_w, t_w + t_c, \dots, t_w + nt_c, \dots$) y geométricamente distribuida:

$$P(t_p = t_w + it_c) = (1 - m)^i m \quad i = 0, 1, \dots \quad (3.32)$$

y siempre que hagamos coincidir el valor medio del tiempo de procesamiento del modelo

$$\sum_{i=0}^{\infty} (1 - m)^i m (t_w + it_c) = t_w + t_c (1 - m)/m \quad (3.33)$$

con el tiempo medio de procesamiento de nuestro sistema real.

3.3.1 Modelo Exacto.

El planteamiento analítico de este modelo será efectuado en forma similar al caso $m = 1$ (apartado 3.2.3) representando el sistema por una cadena discreta finita de Markov cuyo estado queda totalmente definido mediante un vector de M dimensiones (k_1, k_2, \dots, k_M) donde k_i representa el número de procesadores en cola de la memoria i con $0 \leq k_i \leq N$ para $i = 1, 2, \dots, M$ y donde ahora $\sum_{i=1}^M k_i \leq N$.

A cada estado $K (k_1, k_2, \dots, k_M)$ le corresponde un estado parcial (j_1, j_2, \dots, j_M) definido según (3.2).

Un nuevo estado $L (l_1, l_2, \dots, l_M)$ será alcanzable desde el estado K en un solo paso si y sólo si se cumple la condición

$$l_i \geq j_i ; i = 1, 2, \dots, M$$

Si x es el número de elementos k_i no nulos de (k_1, k_2, \dots, k_M) entonces al final del ciclo quedarán $N - \sum_{i=1}^M k_i + x$ procesadores con capacidad de realizar un acceso.

Para alcanzar el estado L desde el estado K será necesario que $\sum_{i=1}^M l_i - \sum_{i=1}^M k_i + x$ de ellos realicen una petición de acceso. La probabilidad de que esto ocurra viene dada por:

$$\binom{N - k_0 + x}{l_0 - k_0 + x} m^{l_0 - k_0 + x} (1 - m)^{N - l_0} \quad (3.34)$$

donde $l_0 = \sum_{i=1}^M l_i$ y $k_0 = \sum_{i=1}^M k_i$

y la probabilidad de que estos $l_0 - k_0 + x$ se distribuyan de forma que, exactamente, $d_i = l_i - j_i$ se dirijan a la memoria i -ésima ($i = 1, 2, \dots, M$) viene dada por

$$\left(\frac{1}{M}\right)^{l_0 - k_0 + x} \frac{(l_0 - k_0 + x)!}{d_1! d_2! \dots d_M!} \quad (3.35)$$

de donde la probabilidad de transición del estado K al estado L vendrá dada por:

$$\binom{N - k_0 + x}{l_0 - k_0 + x} m^{l_0 - k_0 + x} (1 - m)^{N - l_0} \left(\frac{1}{M}\right)^{l_0 - k_0 + x} \frac{(l_0 - k_0 + x)!}{d_1! d_2! \dots d_M!} \quad (3.36)$$

donde $d_i = l_i - j_i$ para $i = 1, 2, \dots, M$

El número de estados de esta cadena puede ser reducido en la misma forma que para el caso $m = 1$ agrupando en macroestados aquellos estados cuyos elementos son permutaciones de los elementos de otro estado. La matriz de transición entre estos macroestados puede ser evaluada con mayor comodidad utilizando el método

de enumeración de Bhandarkar que para el caso del sistema 3 x 3 daría lugar al árbol de la figura 3.5.

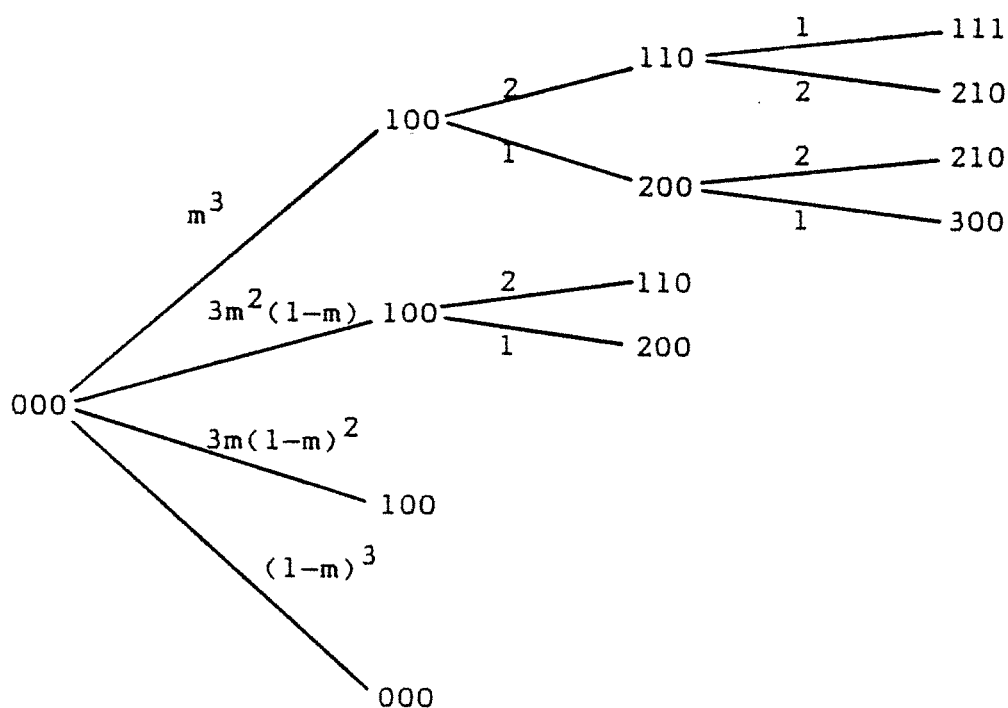


Fig. 3.5 Arbol de enumeración del Sistema 3 x 3.

El número total de estados será ahora $\sum_{j=0}^N \sum_{i=0}^{\min(j,M)} p_i(j)$, donde cada $p_i(j)$ se define en la forma recurrente dada en (3.5).

En la tabla 3.6 se aprecia que, incluso para valores bajos de N y M , el número de estados es muy elevado, por lo que el análisis exacto del modelo es intratable.

N	Numero Estados
2	4
4	12
6	30
8	67
10	139

Tabla 3.6.

Nuevamente diferentes autores proponen modelos simplificados, de los cuales destacaremos los más interesantes.

3.3.2 Aproximación de Patel.

La más simple de todas es la propuesta por Patel [PATE79] y [PATE81], quien realiza una aproximación similar a la de Strecker para el caso $m = 1$; es decir, aquellas peticiones no servidas en el presente ciclo son rechazadas de forma que en el próximo ciclo todos los N procesadores pueden realizar nuevas peticiones a los módulos de memoria común de forma independiente con probabilidad m .

De forma análoga a como se obtuvo la fórmula de Strecker en el caso $m = 1$, la expresión del ancho de banda de las memorias será ahora

$$B = M [1 - (1 - m/M)^N] \quad (3.37)$$

donde $[1 - (1 - m/M)^N]$ representa la probabilidad de que no haya procesadores en la cola de una memoria dada en el ciclo con-

siderado, supuesto que todos los N procesadores pueden requerir acceso a esa memoria con probabilidad m/M .

3.3.3 Aproximación Binomial.

Otra expresión explícita del ancho de banda fue derivada por Baskett y Smith [BASK76] como una ampliación de su aproximación binomial vista para el caso $m = 1$.

Para $m < 1$, por cada T ciclos durante los cuales el procesador no realiza ningún acceso a memoria común, habrá una media de $mT/(1 - m)$ accesos a memoria común, cada uno de los cuales requerirá de 1 ciclo para ser servido y de una media de w ciclos de espera antes de ser servido.

Nótese que el número medio de ciclos que espera cada petición de acceso antes de ser atendida es igual a la longitud media de la cola que ve un procesador al llegar al sistema (esto es cierto, ya que cada procesador requiere sólo un ciclo de servicio).

En un sistema infinito con llegadas de Poisson, el tamaño medio de la cola que ve un procesador al llegar al sistema es igual a la longitud media de la cola [KLEI75]. Sin embargo, en un sistema finito, esto no es necesariamente cierto ya que la distribución de llegadas de procesadores al sistema no es independiente del estado del sistema (es lógico pensar que llegarán menos procesadores cuanto mayor sea la cola). De hecho, la longitud media de la cola vendrá dada en función de w por la expresión:

$$L = \frac{\frac{m}{1-m} T (w+1) \quad N \quad w + 1 \quad N}{T + \frac{m}{1-m} T (w+1) \quad M \quad w + 1 + \frac{1-m}{m} \quad M} \quad (3.38)$$

No es conocida una fórmula general que nos proporcione el valor de w . Una primera aproximación puede establecerse si admitimos que $w = L (N - 1)/N$. Con esta aproximación y sustituyendo en (3.38)

$$L = \frac{- \left(\frac{1}{m} - \frac{N-1}{M} \right) + \sqrt{\left(\frac{1}{m} - \frac{N-1}{M} \right)^2 + 4 \frac{N-1}{M}}}{2 \frac{N-1}{N}} \quad (3.39)$$

haciendo ahora uso de la aproximación binomial (3.19), el ancho de banda tendrá la expresión:

$$B = M_p = 0.5 M [2 + 2L - 1/M - \sqrt{(2 + 2L - 1/M)^2 - 8L}] \quad (3.40)$$

donde L está dado en (3.39)

Eata expresión resulta ser más segura mientras más elevados son los valores de N y M por la misma razón que en el caso $m = 1$.

3.3.4 Aproximación de Hoogendoorn (Modelo GMI).

Una aproximación más refinada fue realizada por Hoogendoorn [HOOG77]. Este autor supone que si y_{ik} es la probabilidad de que el procesador i esté referenciando memoria k en un ciclo dado, entonces y_{ik} es independiente de y_{jh} ; $h, k = 1, 2, \dots, M$; $i, j = 1, 2, \dots, N$; $i \neq j$.

Esto se traduce en asumir que las peticiones no servidas en un ciclo son remitidas de nuevo en el ciclo siguiente como nuevas e independientes peticiones de acceso. Nótese que esta aproximación se reduce a la de Strecker para el caso $m = 1$, ya que, para este caso, todos los procesadores tienen una petición de acceso en cada ciclo. Por tanto, para $m = 1$, los resultados de Hoogendoorn y los resultados de Patel (3.37) coinciden con los resultados de Strecker (3.9).

Si bien el modelo de Hoogendoorn (denominado por su autor "Modelo GMI") es más general y permite tratar los sistemas no uniformemente referenciados (no todas las memorias son accedidas con igual probabilidad), estudiaremos ahora este modelo en el caso uniformemente referenciado.

Según este modelo, si p_{ik} es la probabilidad de encontrar el procesador i referenciando la memoria k con éxito, entonces

$$P_{ik} = y_{ik} \prod_{j \neq i} (1 - y_{jk}) + (y_{ik}/2) \sum_{j \neq i} y_{jk} \prod_{h \neq i, j} (1 - y_{hk}) + \dots + (y_{ik}/N) \prod_{j \neq i} y_{jk} \quad (3.41)$$

$$k = 1, 2, \dots, M ; i = 1, 2, \dots, N$$

Por otra parte, si X es la matriz de acceso estático, cuyos elementos x_{ik} representan la probabilidad de que un acceso a memoria del procesador i sea dirigido a memoria k , entonces, en el equilibrio

$$x_{ik} = p_{ik}/p_i, \text{ siendo } p_i = \sum_{k=1}^M p_{ik} \quad (3.42)$$

$$i = 1, 2, \dots, N ; k = 1, 2, \dots, M$$

Por definición de p_{ik} , por cada T ciclos el procesador i realizará tan sólo $T \sum_{k=1}^M p_{ik}$ accesos con éxito. Dado que, por término medio, cada acceso conlleva $(1 - m)/m$ ciclos sin realizar ninguna petición de acceso a memoria, podemos establecer que por cada T ciclos el procesador i gastará $T \sum_{k=1}^M y_{ik}$ ciclos en referenciar memoria ($T \sum_{k=1}^M p_{ik}$ de los cuales serán accesos con éxito) y el resto del tiempo, en cantidad de $(1 - m)/m T \sum_{k=1}^M p_{ik}$, no realizará ninguna petición.

Igualando la suma de estos tiempos a T obtenemos:

$$T \sum_{k=1}^M y_{ik} + (1-m)/m T \sum_{k=1}^M p_{ik} = T \quad (3.43)$$

$$i = 1, 2, \dots, N$$

Las expresiones (3.41), (3.42) y (3.43) proporcionan un sistema de $2N \times M + N$ ecuaciones, de las cuales tan sólo $2N \times M$ son linealmente independientes y mediante las cuales se puede evaluar

y_{ik} y p_{ik} . El propio Hoogendoorn [H00G77] propone un algoritmo iterativo para la resolución de este sistema.

Para el caso del modelo uniformemente referenciado

$$x_{ik} = 1/M ; i = 1, \dots, N ; k = 1, \dots, M \quad (3.44)$$

y considerando todos los procesadores iguales y todas las memorias iguales:

$$\left. \begin{array}{l} y_{ik} = y \\ p_{ik} = p \end{array} \right\} \quad i = 1, \dots, N \quad k=1, \dots, M \quad (3.45)$$

sustituyendo en el conjunto de ecuaciones (3.41) y (3.42)

$$p = (1 - (1 - y)^N)/N \quad (3.46)$$

de (3.43):

$$My + \frac{1-m}{m} \frac{M}{N} (1 - (1-y)^N) = 1$$

y como:

$$B = N \frac{m}{1-m} \quad U = N \frac{m}{1-m} (1 - My)$$

sustituyendo:

$$B = M \left[1 - \left(1 - \frac{1 - \frac{B}{N} \left(\frac{1}{m} - 1 \right)}{M} \right)^N \right] \quad (3.47)$$

fórmula recurrente para el cálculo del ancho de banda.

Esta fórmula fue anteriormente obtenida por Strecker tal como queda constancia en [WULF72].

3.3.5 Aproximación de Bhandarkar.

Bhandarkar [BHAN75] usa una aproximación parecida a la Hoo-gendoorn para reducir el número de estados de la cadena exacta de Markov que describe el estado del sistema (apartado 3.2.3) asumiendo que en cada ciclo las peticiones de acceso bloqueadas en las colas de las memorias se redistribuyen uniformemente entre todas las memorias, siendo tratadas, pues, como nuevas peticiones independientes. El estado de la cola puede ahora ser caracterizado exclusivamente por el número de peticiones de acceso en el sistema por lo que el número de estados de esta cadena aproximada es de tan sólo $N + 1$ para un sistema $N \times M$.

La probabilidad de que las n peticiones ocupen exactamente k memorias viene dada por

$$\text{CMS}(k, n, M) = \frac{k! S(n, k)}{M^n} \binom{M}{k} \quad (3.48)$$

donde $S(n, k)$ es el número de Stirling [FELL66] de segunda clase, definido por

$$k! S(n, k) = \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n \quad (3.49)$$

Si las n peticiones refieren exclusivamente k memorias, entonces $n - k$ peticiones quedarán bloqueadas en este ciclo y k peticiones serán atendidas. Al final del ciclo habrá $N - n + k$ procesadores con capacidad de hacer una petición a memoria común. La probabilidad de que j de los $N - n + k$ hagan una petición vendrá dada por

$$\binom{N-n+k}{j} m^j (1 - m)^{N-n+k-j} \quad (3.50)$$

y en el próximo ciclo habrá $n - k + j$ peticiones de acceso. Por tanto, la probabilidad de alcanzar el estado h desde el estado n será

$$\sum_{k=1}^{\min(n, M)} \text{CMS}(k, n, M) \binom{N-n+k}{j} m^j (1 - m)^{N-n+k} \quad (3.51)$$

con $n - k + j = h$ y $0 \leq j \leq N - n + k$

Dado que la simplificación adoptada es muy parecida a la de Hoogendoorn los resultados de Bhandarkar en este caso son prácticamente idénticos a los obtenidos por Hoogendoorn (las diferen-

cias entre ambos modelos se explican debido a que Bhandarkar distribuye los procesadores entre las memorias al final de cada ciclo mientras que Hoogendoorn lo hace al principio de cada ciclo).

3.3.6 Aproximación de Yen.

Yen, Patel Y Davidson [YEN82], basándose en la experiencia, propone una fórmula iterativa que de resultados más seguros que todos los modelos anteriores para todos los valores de M , N y m (excepto para $m = 1$).

Para presentar el modelo de Yen, consideremos el caso en que los procesadores no sufran conflictos en sus accesos a memoria. En este caso, por cada T ciclos habrá mT accesos y el ancho de banda de las memorias será $B = NmT/T = Nm$.

Si ahora consideramos la existencia de conflictos, el ancho de banda se reducirá en un factor $f < 1$ desconocido, de forma que $B = Nfm$.

Realizando un balance en el flujo estacionario de procesadores en el sistema, el número medio de procesadores no bloqueados (accediendo o no realizando petición) vendrá dado por $B/m = Nf$ (de los cuales, en valor medio, Nfm estarán accediendo a memoria común y $Nf(1 - m)$ no realizarán petición en este ciclo). El número medio de procesadores bloqueados por ciclo será $N - Nf = N(1 - f)$.

En esencia el problema de determinar f se reduce al problema de determinar cómo se distribuyen estas $N(1 - f)$ peticiones entre las M colas (de hecho éste ha sido nuestro problema desde el principio).

La hipótesis de Hoogendoorn de considerarlos como nuevas peticiones independientes en el ciclo siguiente sabemos que sobrestima el ancho de banda de las memorias.

En primera aproximación Yen considera que los procesadores bloqueados en las colas de las memorias se reparten entre las M colas tal como lo harían N procesadores que realizaran peticiones de acceso independientes con probabilidad $(1 - f)$ por ciclo, de forma que la probabilidad de que una cualquiera de las memorias sufra, al menos, un acceso en el próximo ciclo procedente de estos procesadores bloqueados en el ciclo anterior será $r' = 1 - (1 - (1 - f)/M)^N$. La aproximación de Yen consiste en sustituir los, en valor medio, $N(1 - f)$ procesadores bloqueados por M colas que se requieren a sí mismas en el próximo ciclo de formas independiente con probabilidad r'/M . De esta forma, la probabilidad de que una cualquiera de estas memorias no sufra ningún acceso en el ciclo siguiente procedente de los procesadores bloqueados en el presente ciclo será $(1 - r'/M)^M$.

Por otra parte, la probabilidad de que una cualquiera de estas colas no sufra un acceso procedente de los, en valor medio, Nf procesadores no bloqueados será $(1 - mf/M)^N$ y, por tanto, el ancho de banda será

$$B = Nfm = M \left[1 - \left(1 - \frac{mf}{M} \right)^N \left(1 - \frac{1 - \left(1 - \frac{1-f}{M} \right)^N}{M} \right)^M \right] \quad (3.52)$$

lo que proporciona una fórmula iterativa para el cálculo de f y, por tanto, del ancho de banda.

Si bien, la simplificación de Yen no tiene una clara justificación teórica, los resultados que se obtienen de su modelo son los más cercanos a los valores exactos que se conocen, excepción hecha de los resultados obtenidos de la Aproximación de Descomposición de Rau (apartado 3.2.4), pero cuya aplicación está restringida al caso $m = 1$.

3.3.7 Aproximación de Bhandarkar Modificada.

Un último modelo original será propuesto cuyo principal interés se mostrará en el capítulo siguiente y que denominaremos modelo de Bhandarkar modificado, dado que es un refinamiento de este último.

En este modelo, el estado del sistema queda caracterizado por el número de procesadores que se encuentren requiriendo acceso a memoria en un ciclo dado. Las probabilidades de transición entre estados se evalúan en la misma forma que en el modelo de aproximado de Bhandarkar, pero asumiendo que la probabilidad de encontrar k memorias ocupadas al final de un ciclo supuesto que hay n procesadores requiriendo acceso a memoria viene dada por:

$$P_{nk} = P(k \text{ mem. ocupadas/n proc. acceden}) = \alpha \text{ CMS}(k, n, M) \\ + (1 - \alpha) \sum_{j=k+1}^{\min(n, M)} \text{CMS}(j, n, M) \text{CMS}(k, j, M) \quad (3.53)$$

con $0 \leq \alpha \leq 1$ que sustituye a (3.48) en el modelo aproximado de Bhandarkar.

La probabilidad de transición del estado n al estado h vendrá dada por:

$$P(n \rightarrow h) = \sum_{k=0}^{\min(n, M)} P_{nk} \binom{N-n+k}{h-n+k} m^{N-n+k} (1-m)^{N-h} \quad (3.54)$$

cuando $h \geq \max(0, n - M)$ y 0 en caso contrario.

Experimentalmente se ha comprobado que para $\alpha = 0.75$ se obtienen los mejores resultados.

Se ha comprobado también que, salvo para casos donde el conflicto es muy elevado ($N \gg M$ y $m > 0.5$), este modelo tiene un comportamiento muy superior al modelo aproximado de Bhandarkar y tan sólo mejorado por el modelo de Yen antes comentado.

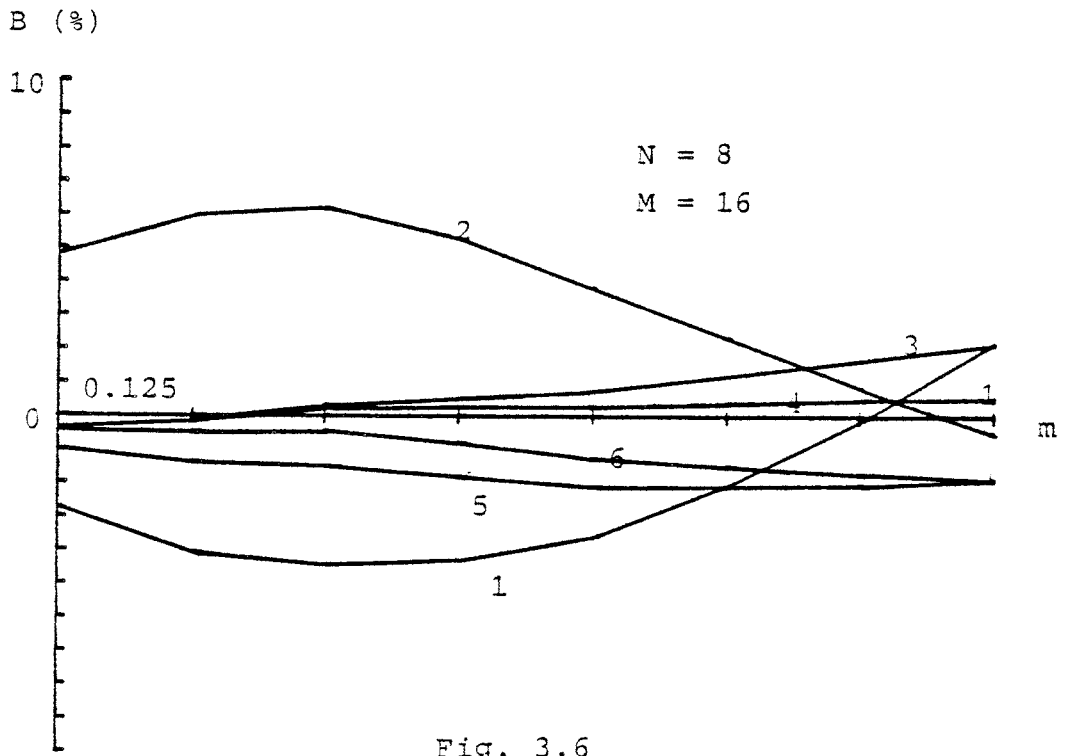


Fig. 3.6

- | | |
|---------------------------|-------------------------------|
| 1 = Modelo de Patel | 4 = Modelo de Yen |
| 2 = Aproximación Binomial | 5 = Modelo de Bhandarkar |
| 3 = Modelo de Hoogendoorn | 6 = Modelo de Bhandarkar Mod. |

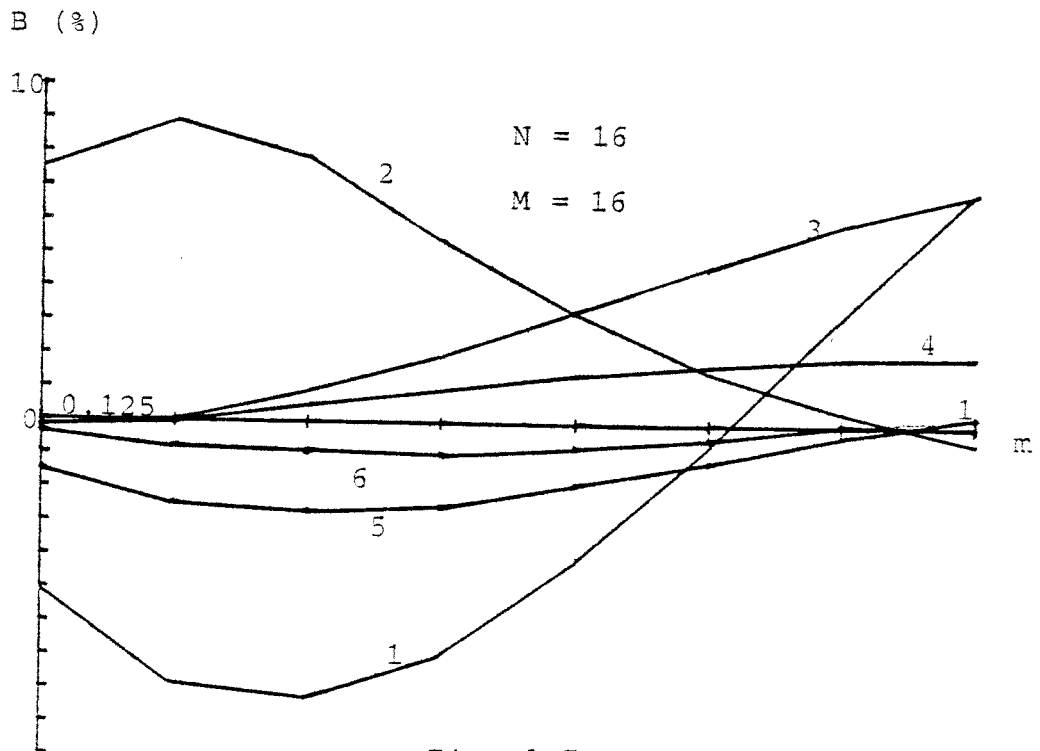
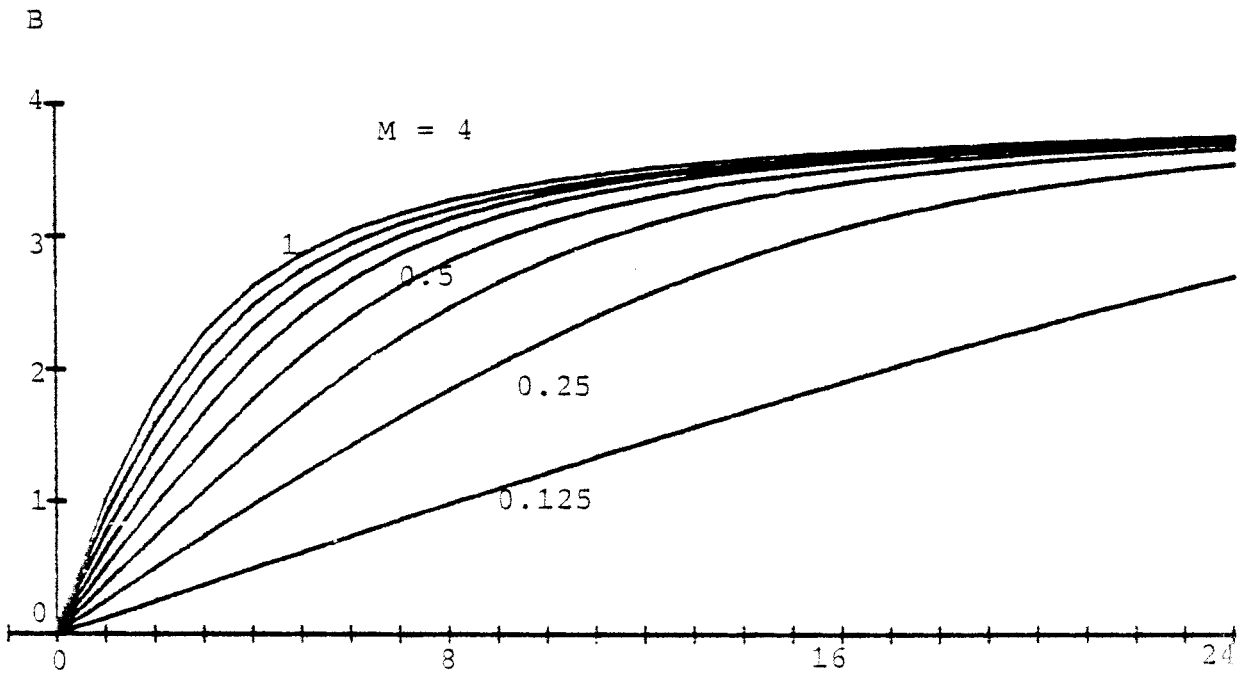
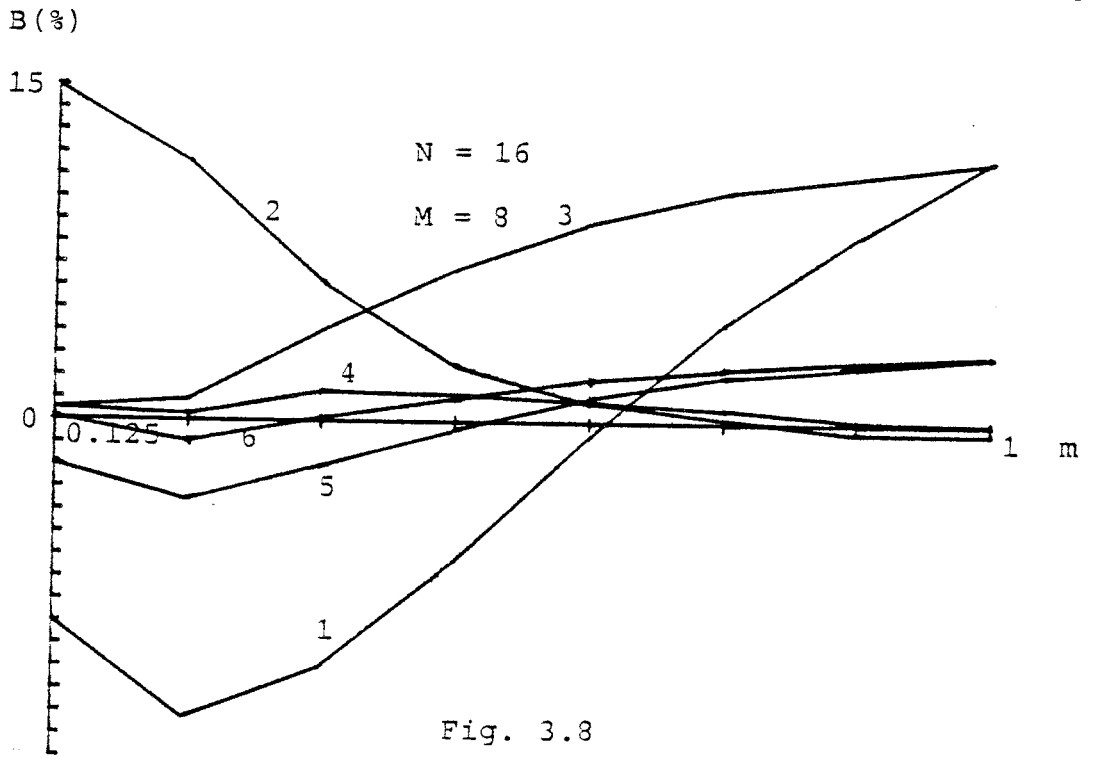


Fig. 3.7



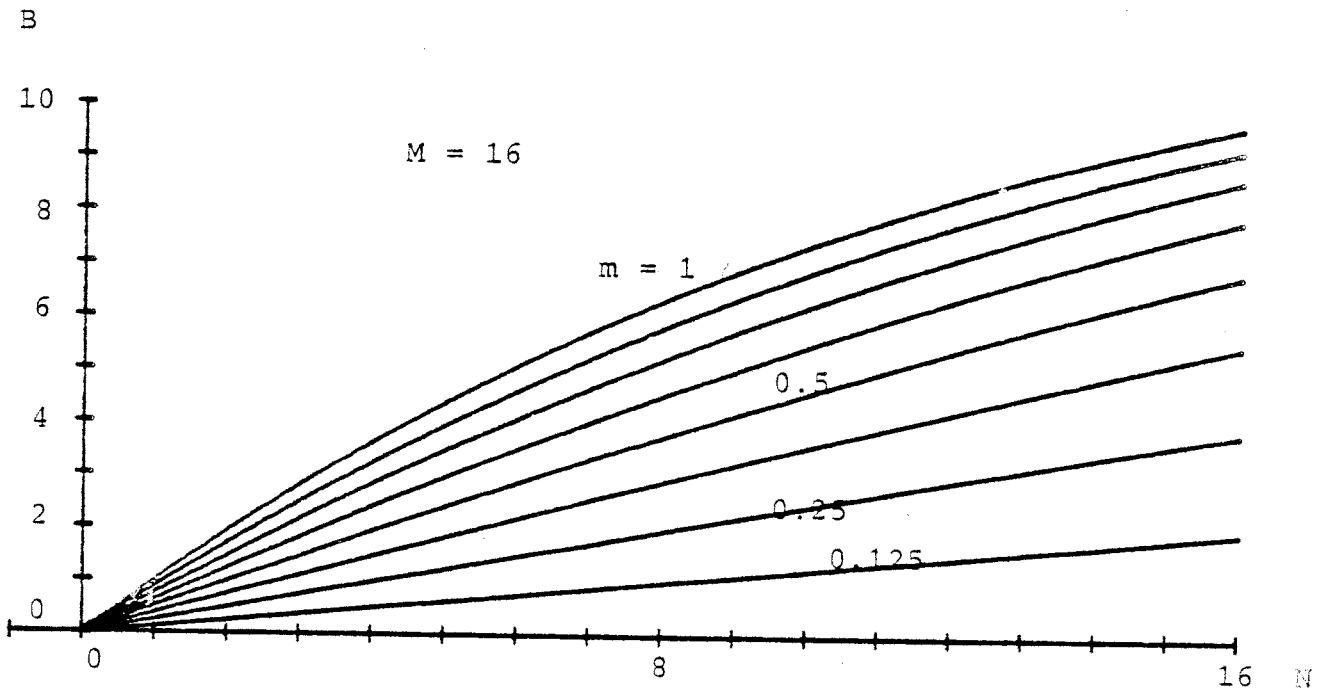


Fig. 3.11

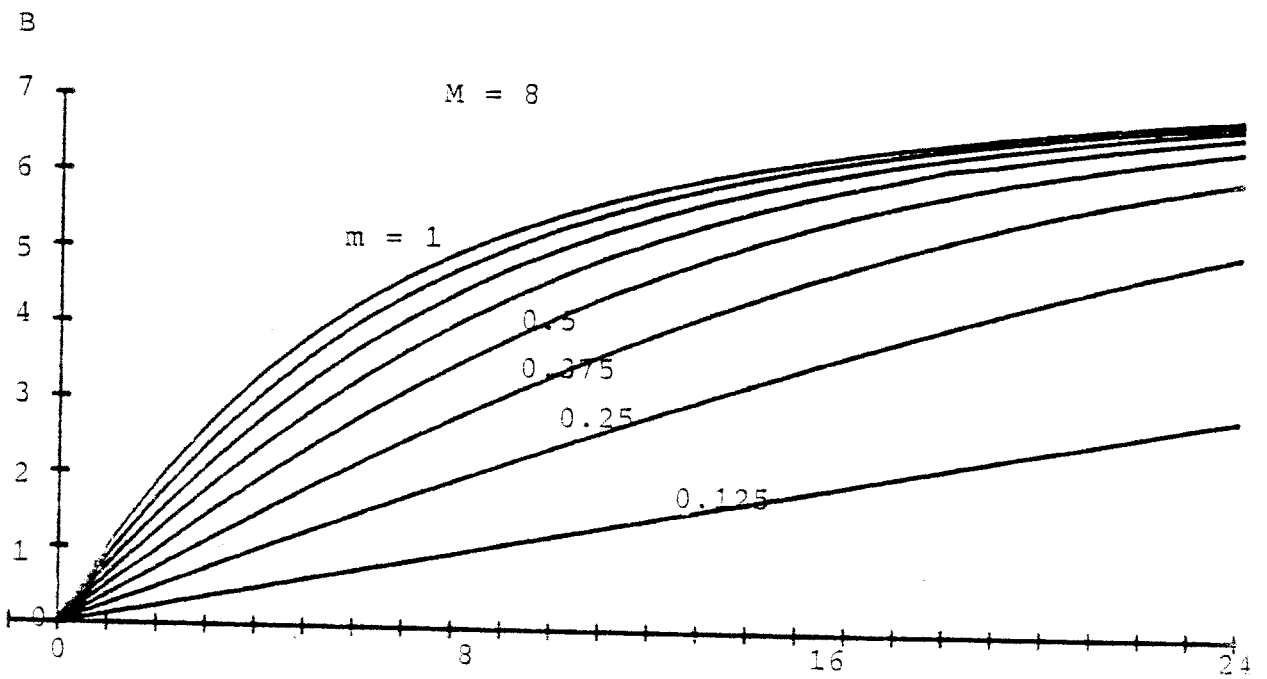


Fig. 3.10

Es de notar el hecho de que para $m = 1$ los resultados de este modelo original son los mismos que los obtenidos con el modelo aproximado de Bhandarkar.

3.3.8 Comparación de las diferentes aproximaciones.

En las figuras 3.6, 3.7 y 3.8 se muestra la diferencia en tantos por ciento entre los diferentes modelos aproximados y los valores obtenidos por simulación para diferentes valores de N y M .

Se puede observar que los resultados de Yen son extraordinariamente próximos a los obtenidos por simulación y que, excepto para el caso $M = 8$, $N = 16$ y $m > 0.5$, el modelo original propuesto (denominado Modificado de Bhandarkar) es el siguiente con mayor precisión.

Se observa asimismo que el modelo de Baskett y Smith va haciéndose más preciso a medida que aumentan los valores de N y M como era lógico esperar ya que el modelo es asintóticamente exacto. Por el contrario los modelos de Patel y Strecker-Hoogendoorn van perdiendo precisión a medida que N y M aumentan.

En las figuras 3.9, 3.10 y 3.11 se aprecia el efecto de incrementar N para $M = 4$, $M = 8$ y $M = 16$ (los resultados proceden todos de simulación).

En todas ellas puede apreciarse que el incremento del ancho de banda es, en un principio, lineal alcanzándose una saturación

para N elevados. Cuanto mayor es el valor de m tanto más rápidamente se alcanza el punto de saturación y, en general, para $m > 0.5$ el aumento deja de ser lineal para valores de N ligeramente inferiores a M , alcanzándose la región de saturación para valores de N entre dos y tres veces el número de memorias.

3.5 ANALISIS EN TIEMPO CONTINUO.

Para efectuar el análisis en tiempo continuo de un sistema "cross-bar" haremos análogamente al caso discreto una serie de simplificaciones que se indican a a continuación.

Asumiremos que los tiempos de servicio de las memorias son exponencialmente sistribuidos de media $1/\mu$ y que los tiempos durante los cuales los procesadores ejecutan en memoria privada son también exponencialmente distribuidos de media $1/\lambda$.

Adoptaremos una política de asignación de memorias del tipo FCFS, si bien cualquier política sin interrupciones que se considerara daría lugar a los mismos anchos de banda dado que los procesadores son indistinguibles.

Este modelo es muy representativo de un sistema en que los procesadores se encuentran normalmente ejecutando en memoria privada y tan sólo utilizan los módulos de memoria común para el intercambio de mensajes y commutación de las tareas.

El modelo así planteado admite una solución en "forma de producto" (Apartado 2.4) pudiendo ser resuelto, bien por aplicación directa de la fórmula de Gordon-Newell, bien por análisis paramétrico mediante el uso del teorema de descomposición de Chandy, Woo y Herzog.

Haciendo uso de este último método y definiendo el estado de la cadena como el número de procesadores que se encuentran accediendo o en cola de las memorias, las probabilidades estacionarias de los estados vienen dadas mediante la resolución del sistema de ecuaciones lineales:

$$P(k) = \frac{(N - k + 1)(M + k - 1)}{k M \mu} \lambda P(k - 1) \quad k = 1, 2, \dots, N \quad (3.55)$$

$$k = 1, 2, \dots, N$$

$$\sum_k P(k) = 1 \quad (3.56)$$

que admite una solución explícita de la forma:

$$P(k) = \frac{N! (M + k - 1)! e^k}{M! (N - k)! k! M^{k-1}} \quad (3.57)$$

$$1 + \sum_{k=1}^N \frac{N! (M + k - 1)! e^k}{M! (N - k)! k! M^{k-1}}$$

$$k = 0, 1, 2, \dots, N$$

El número medio de procesadores ejecutando en memoria privada viene dado por:

$$P = \sum_{k=0}^N (N - k) P(k) \quad (3.58)$$

y, por aplicación de la ley de Little:

$$B = P \rho \quad (3.59)$$

CAPITULO 4.

INTERCONEXION MEDIANTE BUSES.

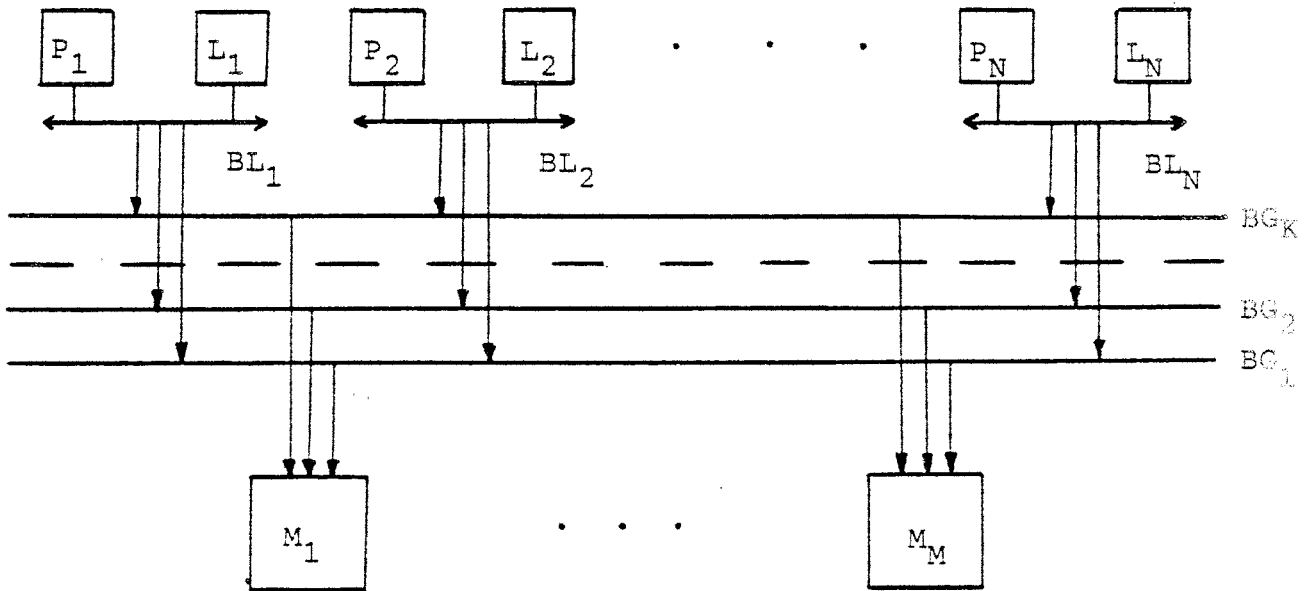
4.1 INTRODUCCION.

En el capítulo anterior hemos estudiado los sistemas completamente interconectados o sistemas "cross-bar". Hemos visto que si bien el ancho de banda de las memorias y la utilización de los procesadores en estos sistemas no siguen, al aumentar N y M , la línea recta que cabría esperar en un sistema ideal sin conflictos, no obstante, para valores de N y M próximos y para probabilidades de acceso por ciclo no muy elevadas ($m < 0.5$), la variación de sus prestaciones con N y M no se aparta mucho de la línea recta ideal.

A nadie se le escapa, sin embargo, la gran complejidad de estos sistemas que requieren un selector de tamaño $N \times N$ y, sobre todo, su baja tolerancia a faltas.

Como respuesta a los sistemas "cross-bar" se han venido utilizando los sistemas basados en buses cuyo grado de redundancia los hace muy atractivos y los cuales presentan, siempre que el número de buses sea reducido, un nivel de complejidad inferior al de los sistemas "cross-bar".

Nuestro propósito en este capítulo es el de analizar los sistemas basados en buses, dedicando una especial atención a los sistemas con un único bus debido a su gran sencillez y a la popularidad que estos sistemas están alcanzando.



P	Procesador	M	Memoria Común
L	Memoria Local	BG	Bus Global
BL	Bus Local		

Fig. 4.1 Sistemas con Buses.

En el análisis que sigue, dos formas diferentes de acceso a los buses pueden ser consideradas.

En la primera de ellas, los procesadores compiten en primer lugar por los buses y, una vez ganado el acceso a estos, aquellos que han conseguido un bus compiten por acceder a memoria.

En la segunda estrategia los procesadores compiten en primer lugar por las memorias y, posteriormente, sólo los que han ganado acceso a éstas compiten por el uso de un bus.

La primera estrategia, que denominaremos "estrategia 1", requerirá de 1 árbitro de N usuarios y K servidores para asignar los buses y de M árbitros de K usuarios y 1 servidor para asignar las memorias a las peticiones que han conseguido el uso de un bus.

La segunda estrategia, que denominaremos "estrategia 2", requerirá un árbitro de M usuarios y K servidores para seleccionar $\min(K, J)$ de entre los J módulos de memoria que tienen al menos una petición pendiente y M árbitros de N usuarios y 1 servidor para asignar a cada memoria sólo uno de los N posibles procesadores que requieren su acceso.

Si bien esta segunda estrategia requiere árbitros ligeramente más complejos, es evidente que sus prestaciones serán mejores que las de la primera y, por ello, es el caso más estudiado en la bibliografía.

4.2 ANALISIS EN TIEMPO DISCRETO.

Bajo las mismas hipótesis simplificativas que se consideraron en el sistema "cross-bar" (apartado 3.2.1) es posible establecer un modelo analítico exacto basado en cadenas discretas de Markov.

Admitiendo asimismo que los árbitros de las memorias y buses son neutrales y una estrategia de tipo 2, el estado de tal cadena vendría completamente definido por:

$(n_m, q_1, q_2, \dots, q_K, q_{K+1}, q_{K+2}, \dots, q_M)$ donde:

n_m = número de procesadores que no realizan una petición a memoria en el ciclo considerado.

q_1, q_2, \dots, q_K = número de procesadores en cola de las memorias actualmente accedidas y ordenados estos q_i en orden decreciente.

$q_{K+1}, q_{K+2}, \dots, q_M$ = número de procesadores en cola de las memorias no accedidas a causa de que los K buses se encuentran ya ocupados y ordenados estos q_{K+i} en orden decreciente.

La razón de que se ordenen en orden decreciente tiene por finalidad reducir el número de estados de la cadena. En efecto, dado que los procesadores, memorias y buses son indistinguibles, dos estados con el mismo valor de n_m y cuyos elementos q_1, q_2, \dots, q_K de uno sean una permutación de los del otro y cuyos elementos $q_{K+1}, q_{K+2}, \dots, q_M$ de uno sea una permutación de los del otro, tienen igual probabilidad estacionaria y pueden ser agrupados en un único macroestado del cual podemos escoger como estado representativo aquél cuyos q_i y q_{K+i} están ordenados en orden decreciente.

Nuevamente el número de estados de esta cadena exacta es extraordinariamente elevado incluso para valores reducidos de N, M y K .

No conocemos ninguna aproximación en el caso discreto para el análisis de estos sistemas, no obstante, el método original

introducido en el apartado 3.3.7 y denominado de Bhandarkar modificado es fácilmente aplicable a este caso.

En este modelo, se asume que los procesadores que se encuentran efectuando una petición de acceso se distribuyen entre las memorias de forma que la probabilidad de que haya k memorias referenciadas dado que hay n peticiones de acceso, viene dada por P_{nk} , donde P_{nk} se define en (3.53).

En el caso de un sistema "cross-bar" toda memoria que tiene, al menos, una petición de acceso en un ciclo dado se encontrará dando servicio en dicho ciclo.

En un sistema multibus con estrategia 2 esto será también cierto siempre que el número total J de memorias referenciadas en un ciclo no supere el número de buses K . Si J es mayor que K , tan sólo K memorias podrán estar dando servicio a un procesador, permaneciendo el resto a la espera de que un bus quede libre.

Denominemos P'_{nk} a la probabilidad de que haya k memorias dando servicio en un ciclo, dado que hay n peticiones de acceso.

Para un sistema "cross-bar":

$$P'_{nk} = P_{nk} \quad ; \quad n = 1, 2, \dots, N \quad ; \quad k = 1, 2, \dots, \min(n, M) \quad (4.1)$$

Para un sistema con K buses,

$$P'_{nk} = \begin{cases} P_{nk} & n = 1, 2, \dots, N \quad k = 1, 2, \dots, \min(n, K-1) \\ \sum_{k=K}^N P_{nk} & n = K, K+1, \dots, N \quad k = K \\ 0 & k > K \end{cases} \quad (4.2)$$

y la probabilidad de transición entre estados vendrá dada por (3.54) sustituyendo P_{nk} por P'_{nk} .

El comportamiento de este modelo es bastante bueno tal como se observa en las tablas 4.1 y 4.2 al comparar los resultados del modelo con los obtenidos mediante simulación por T.Lang y otros y mostrados en [LANG83].

N	4		8		12		16	
	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.
K								
1	1.00	0.97	1.00	1.01	1.00	1.01	1.00	1.00
2	1.97	1.88	2.00	2.02	2.00	2.01	2.00	2.00
3	2.55	2.43	3.00	3.02	3.00	3.02	3.00	3.01
4	2.62	2.53	3.93	3.94	4.00	4.02	4.00	4.01
5			4.62	4.62	4.99	5.00	5.00	5.01
6			4.90	4.92	5.93	5.90	6.00	6.01
7			4.94	4.97	6.68	6.63	6.98	6.97
8			4.95	4.98	7.12	7.09	7.92	7.85
9					7.27	7.27	8.72	8.61
10					7.28	7.30	9.27	9.17
11					7.30	7.31	9.53	9.49
12					7.30	7.31	9.61	9.60
13							9.63	9.62
14							9.63	9.63
15							9.63	9.63
16							9.63	9.63

Tabla 4.1. Comparación de los resultados del Modelo (MOD.) con resultados de Simulación (SIM.) para $m = 1$.

N	4		8		12		16	
	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.
K								
1	1.00	0.97	1.00	1.01	1.00	1.00	1.00	1.00
2	1.65	1.62	2.00	1.99	2.00	2.01	2.00	2.01
3	1.77	1.75	2.87	2.86	3.00	3.01	3.00	3.01
4	1.77	1.76	3.33	3.30	3.95	3.96	4.00	4.01
5			3.45	3.43	4.67	4.66	4.98	4.99
6			3.47	3.45	5.03	5.00	5.85	5.85
7			3.47	3.45	5.13	5.11	6.43	6.41
8			3.47	3.45	5.16	5.13	6.70	6.68
9					5.16	5.14	6.82	6.79
10					5.16	5.14	6.83	6.81
11					5.16	5.14	6.83	6.82
12					5.16	5.14	6.83	6.82
13							6.84	6.82
14							6.84	6.82
15							6.84	6.82
16							6.84	6.82

Tabla 4.2. Comparación de los resultados del Modelo (MOD.) con resultados de Simulación (SIM.) para $m = 0.5$.

En las figuras 4.2 y 4.3 se observa cómo varía el ancho de banda de las memorias en los sistemas 16×16 y 24×24 al variar el número de buses con diferentes valores de m ; las curvas han sido obtenidas a partir del modelo aproximado propuesto.

En la figura 4.4 se representa la variación del ancho de banda de un sistema 24×24 al variar m para $K = N$ (sistema cross-bar), $K = N/2$ y $K = N/4$, observándose que, para valores de m reducidos ($m < 0.5$), incluso con un número pequeño de buses ($K = N/2$) los anchos de banda obtenidos con los sistemas multibuses son comparables a los obtenidos con un sistema "cross-bar".

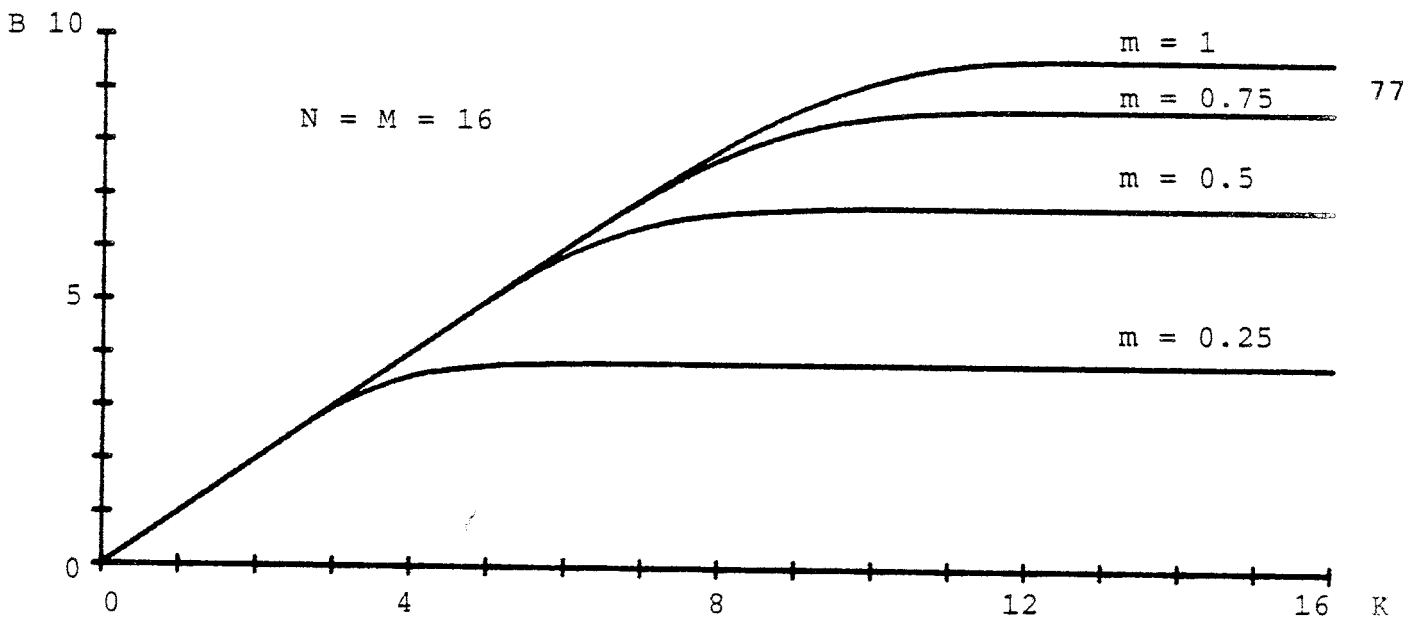


Fig. 4.2.

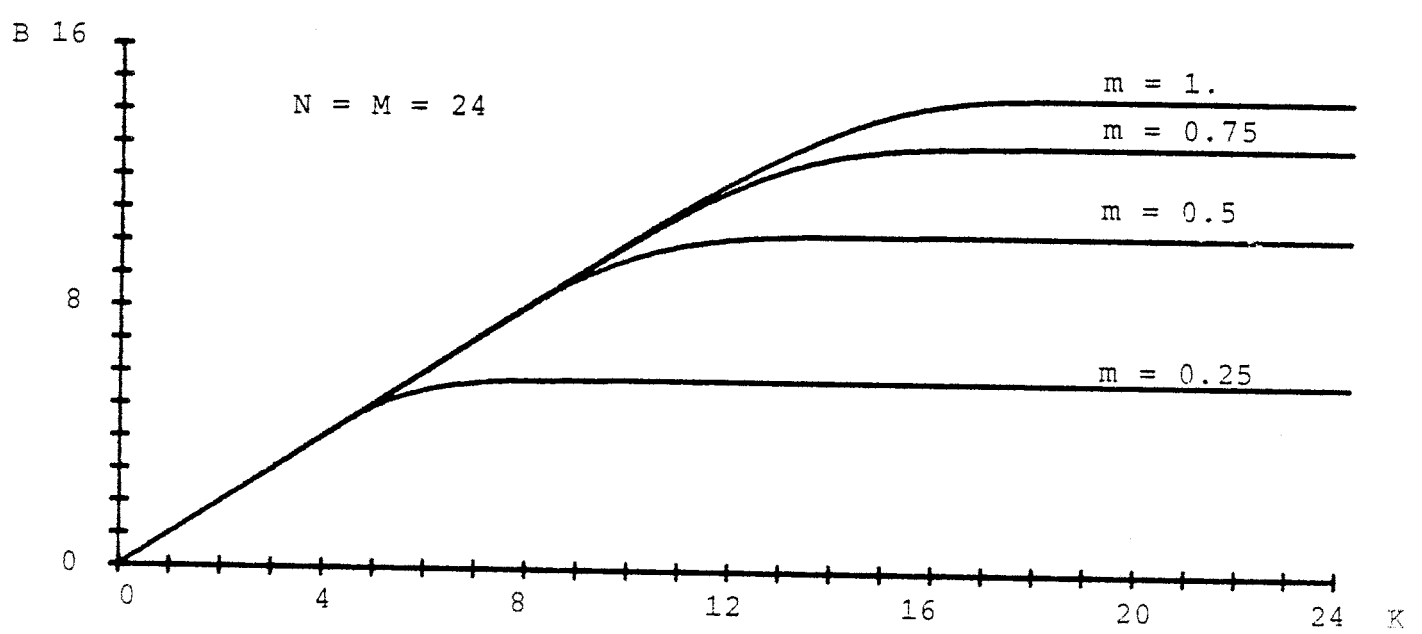


Fig. 4.3.

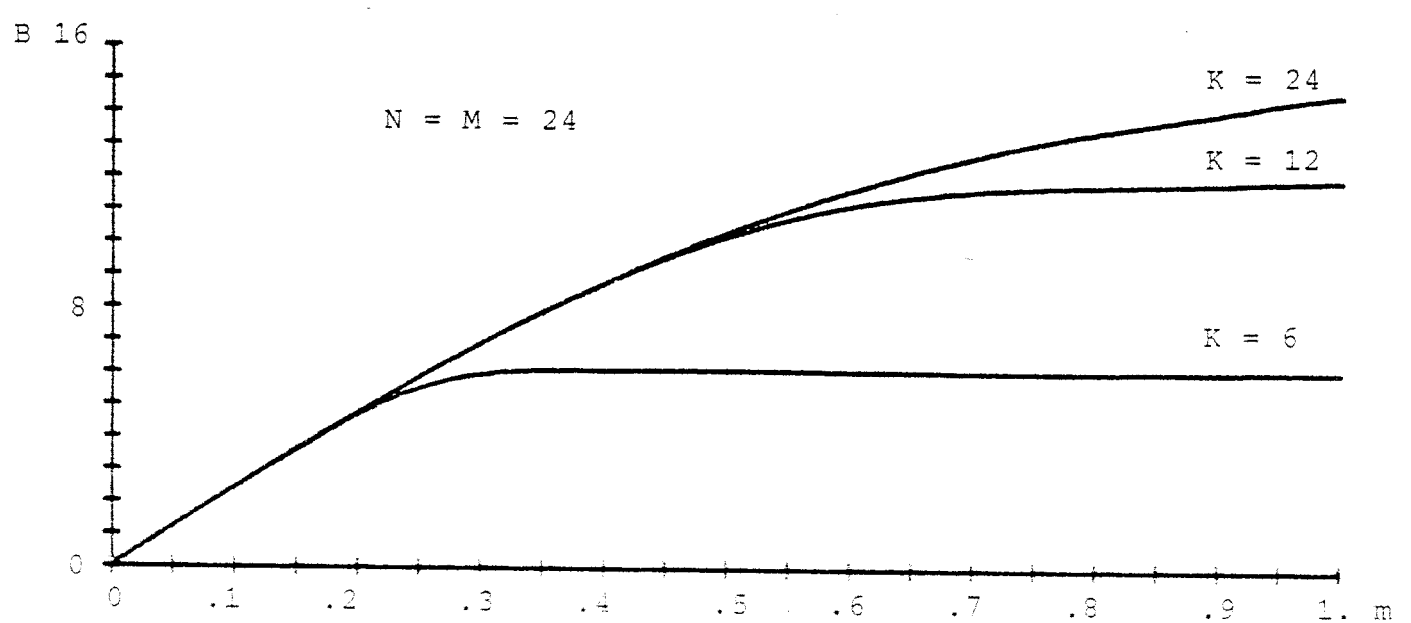


Fig. 4.4

4.3 ANALISIS EN TIEMPO CONTINUO.

Para el análisis en tiempo continuo de sistemas multibuses adoptaremos las mismas hipótesis simplificativas que las adoptadas para un sistema "cross-bar" (apartado 3.2.1).

A diferencia de los sistemas "cross-bar", los sistemas multibus en tiempo continuo no admiten una solución explícita en forma de producto (Apartado 2.4) ya que la utilización de recursos pasivos es una de las situaciones en que se violan las condiciones para que se de una solución de este tipo.

Por supuesto, una cadena de Markov con la misma definición de estado que en el caso discreto es posible construir para analizar de forma exacta el problema, pero por la misma causa que en el caso discreto, el número excesivo de estados a que da lugar esta cadena desaconseja estos métodos exactos.

Diferentes modelos aproximados han sido propuestos en la bibliografía, algunos de los cuales presentamos a continuación.

4.3.1 Modelo Aproximado de Kriz.

Kriz [KRIZ83] propone un modelo aproximado para un sistema multibus con una estrategia de tipo 1. El autor asume una política de tipo FIFO para el manejo de los recursos, si bien los resultados obtenidos son aplicables a cualquier política sin interrupciones tal como demuestra Regis en REGI73.

El modelo de colas de un sistema multibus con estrategia 1 se representa en la figura 4.5.

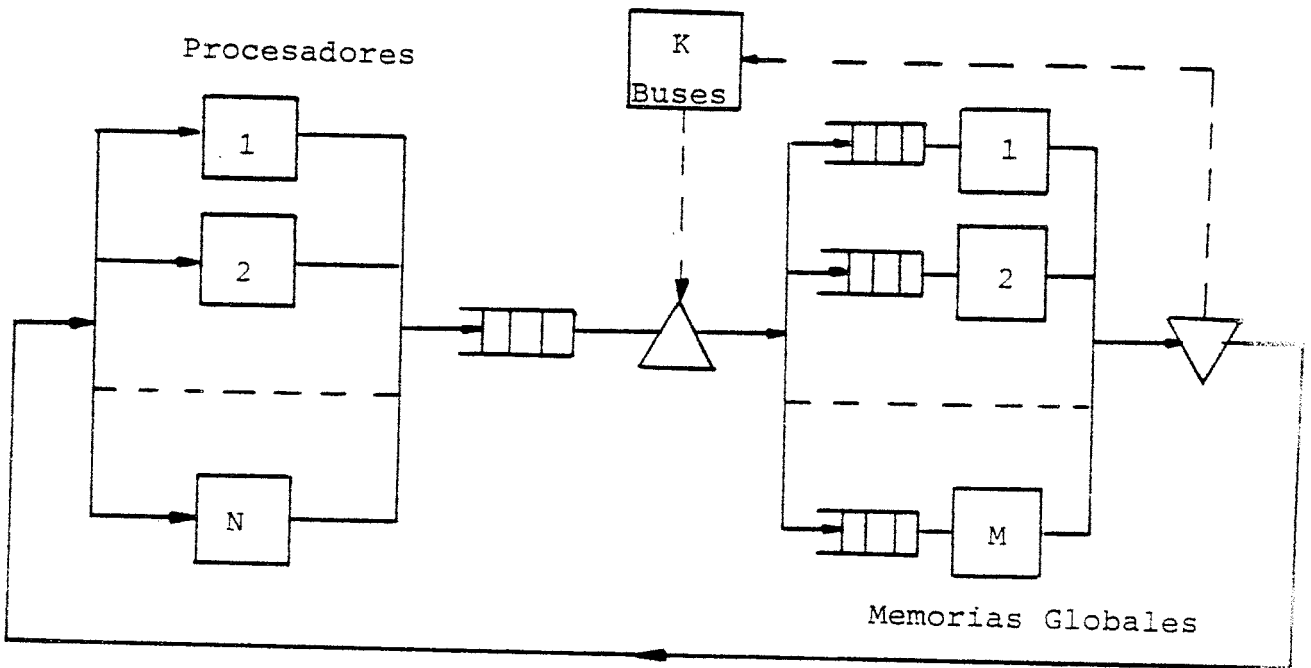


Fig. 4.5. Modelo de colas de un sistema multibus con estrategia 1

La utilización de un recurso pasivo hace que a este modelo no le pueda ser aplicado en forma exacta el teorema de descomposición de Chandy, Woo y Herzog [CHAND75]. La aproximación de Kriz se basa en aplicar este teorema al modelo.

Para este caso, el modelo interno definido en este teorema sería el que se representa en la figura 4.6.

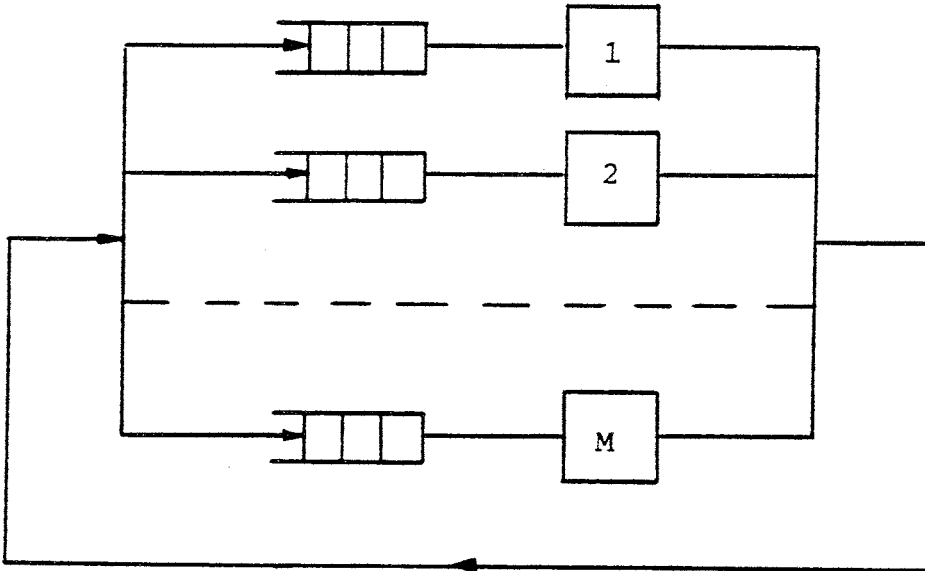


Fig. 4.6. Modelo del Sistema Interno.

Asumiendo que hay i procesadores en el sistema ($i = 1, 2, \dots, K$) el modelo del sistema interno puede ser resuelto por aplicación de la fórmula de Jackson y Gordon (Apartado 2.4), obteniéndose una velocidad de servicio del servidor equivalente igual a:

$$x(i) = i M \mu / (i + M - 1) \quad ; \quad i < K \quad (4.3)$$

Para el caso $i > K$ se extiende el resultado en la forma:

$$x(i) = K M \mu / (K + M - 1) \quad ; \quad i > K \quad (4.4)$$

Todo el sistema interno se reemplaza ahora por un único servidor con esta velocidad de servicio. El modelo del sistema externo quedará por tanto como el representado en la figura 4.7.

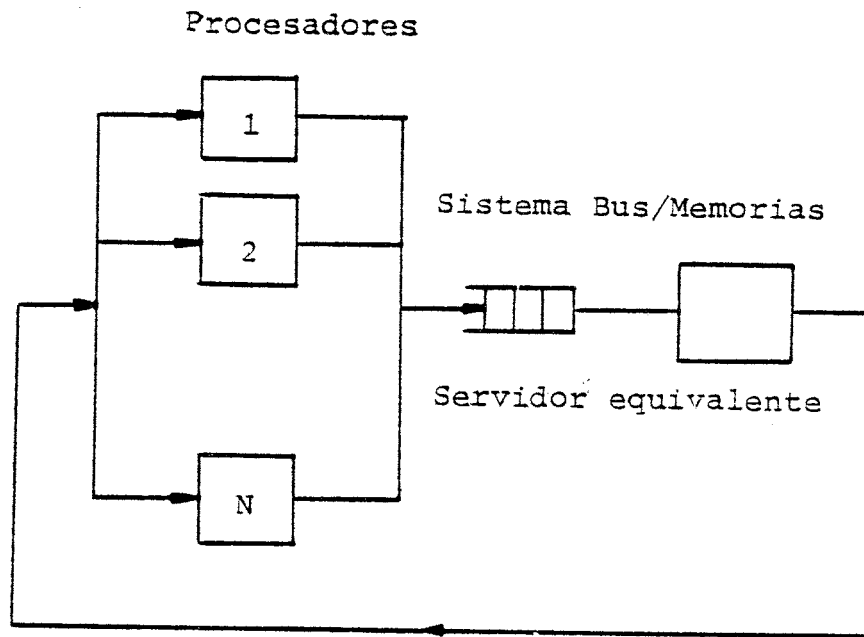


Fig. 4.7. Modelo del Sistema Externo.

Este modelo externo puede representarse ahora en una cadena de Markov cuyo estado queda completamente definido por el número de peticiones de acceso.

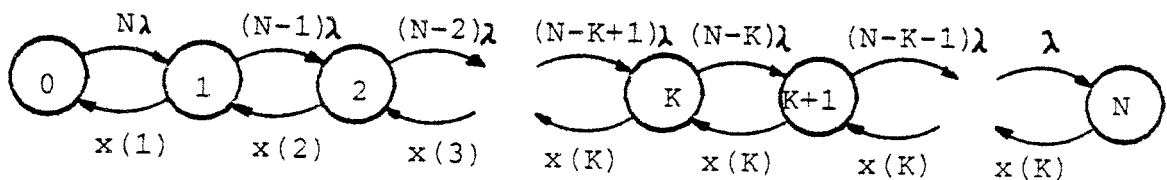


Fig. 4.8. Cadena de Markov correspondiente al sistema externo.

La determinación de las probabilidades estacionarias de cada uno de los estados es ahora inmediata.

Admitir los resultados del teorema de descomposición en este caso es tanto como admitir que el sistema interno actúa tal como si se encontrara permanentemente en estado estacionario.

Esta aproximación será exacta en los siguientes casos:

- a) $K \circ M = 1$.
- b) $K \geq \min(N, M)$. Este caso corresponde al de un sistema crossbar.
- c) Para valores asintóticos, es decir:
 - c-1) $N \rightarrow \infty$ o $\lambda \rightarrow \infty$ y $\mu \rightarrow 0$. En este caso tendremos un nivel de conflicto extraordinariamente alto y el sistema interno permanecerá casi todo el tiempo con K procesadores.
 - c-2) $\lambda \rightarrow 0$ y $\mu \rightarrow \infty$. Este sería el caso en que el nivel de conflicto es prácticamente nulo y, por tanto, el efecto de los buses es despreciable.

En el resto de los casos la aproximación presenta, no obstante, un error bastante apreciable. En la tabla 4.3 pueden observarse los errores cometidos por este modelo para sistemas 8 x 8 con 2 y 4 buses para diferentes valores de ρ al ser comparados con resultados procedentes de simulación.

El autor obtiene también mediante técnicas de análisis del valor medio cotas superior e inferior las cuales tienen un gran interés para conocer el comportamiento asintótico ($N \rightarrow \infty$) del modelo.

		0.1	0.2	0.5	1.0	2.0	10.0
K = 4	MOD	0.72	1.29	2.32	2.79	2.90	2.91
	SIM	0.42	0.85	1.71	2.41	2.73	2.84
K = 2	MOD	0.72	1.22	1.73	1.78	1.78	1.78
	SIM	0.41	0.82	1.53	1.70	1.71	1.76

Tabla 4.3. Comparación del Modelo con resultados de Simulación para un sistema 8 x 8 con 2 y 4 buses.

4.3.2 Modelo de Jacobson.

Jacobson y Lazowska [JAC082] consideran una estrategia de tipo 2 y proponen un modelo iterativo aproximado que denominan Método de los Sustitutos y que está basado en el teorema de descomposición.

El modelo de colas de un sistema multibus con estrategia 2 se muestra en la figura 4.9.

Aplicando el método de los sustitutos, este modelo se descompone en otros 2 representados en las figuras 4.10 y 4.11.

El modelo de la figura 4.10 tiene en cuenta el conflicto debido a las memorias. El conflicto debido a los buses es simplemente representado por un retraso exponencialmente distribuido o "delay server" (Sustituto 1).

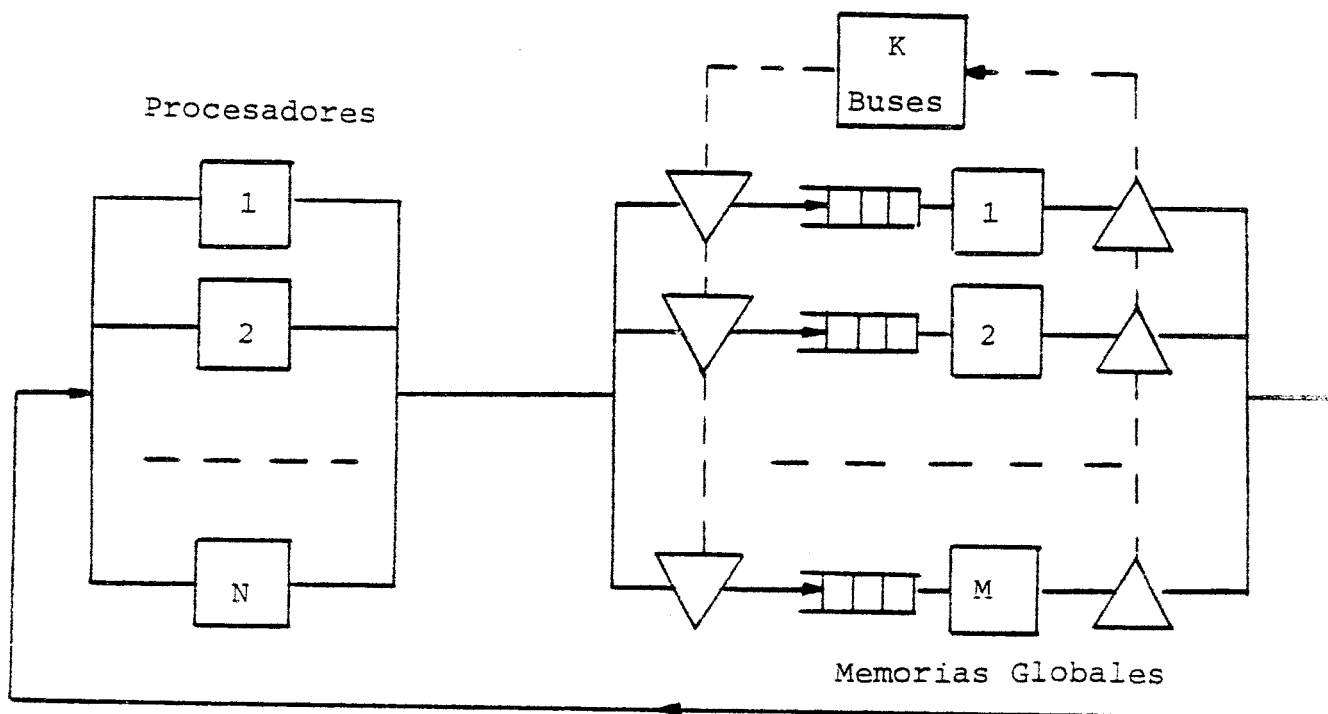


Fig. 4.9. Modelo de un sistema multibus con estrategia 2.

El modelo de la figura 4.11, por el contrario, tiene en cuenta el conflicto debido a los buses, representándose el conflicto debido a las memorias como un simple retraso (Sustituto 2).

El problema se resuelve mediante un algoritmo iterativo que procede como sigue:

1. Se resuelve el sistema 4.10 (al que puede ser aplicado la fórmula de Jackson y Gordon) haciendo el retraso R_2^i debido a los buses = 0 y se calcula el retraso debido a las memorias como el tiempo medio que pasa un procesador en la cola de una memoria antes de ser servido.

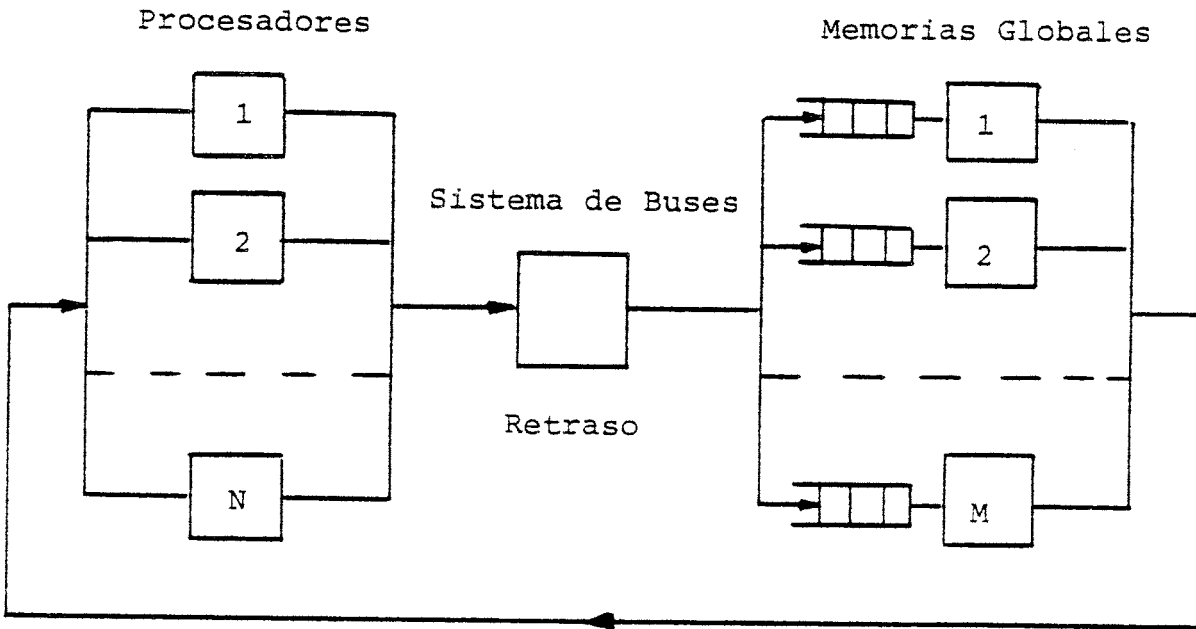


Fig. 4.10

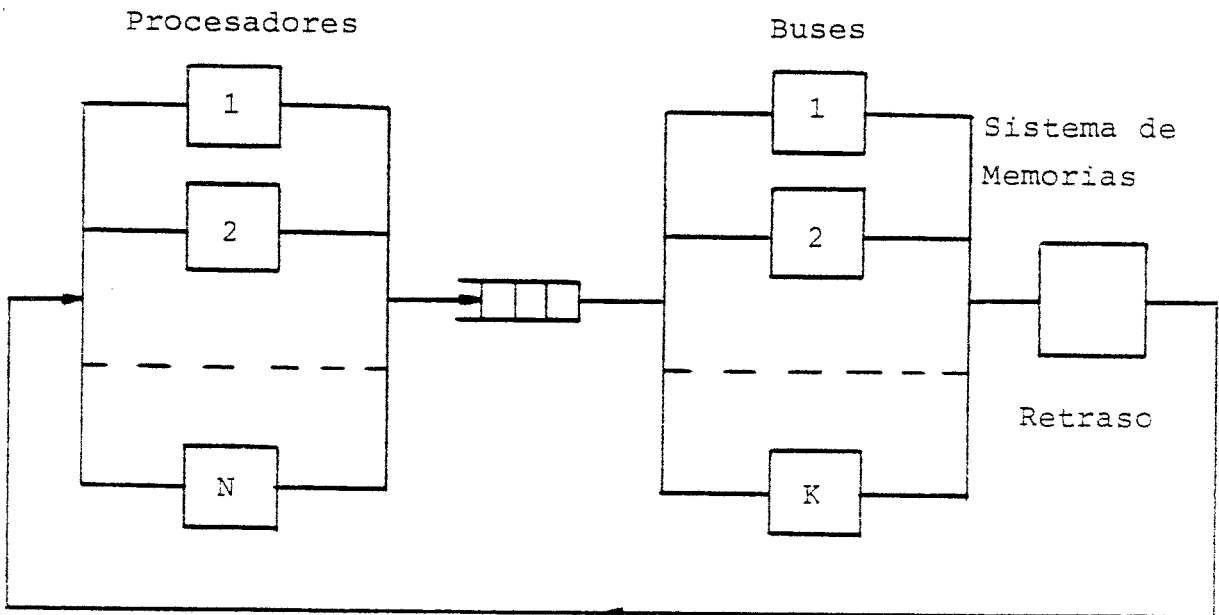


Fig. 4.11

2. Tomar como tiempo de servicio de los buses en el sistema 4.11 el tiempo de servicio de las memorias y hacer el retraso debido a los conflictos de memoria (Sustituto 2) igual a R_2^i calculado anteriormente.
3. Resolver el modelo 4.11 (que admite la fórmula de Jackson y Gordon) y calcular R_1^i o tiempo medio que pasa un procesador en la cola de buses antes de ser servido.
4. Resolver el sistema 4.10 haciendo el retraso en el sustituto 1 igual a R_1^i y calcular R_2^{i+1} .
5. Si $R_2^{i+1} = R_2^i$ terminar el proceso. En caso contrario, $R_2^i = R_2^{i+1}$ e ir al punto 2.

La convergencia del algoritmo está asegurada por su propia naturaleza.

Este método, en general, subestima el ancho de banda del sistema y el propio autor muestra por simulación que, para $N = 8$ y $M = 4$, los errores no sobrepasan el 2.5% (tabla 4.4) y afirma que, en general, los errores típicos son del orden de un 5%.

K	1	2	3	4
MODELO	2.0	3.6	4.1	4.2
SIMUL.	2.0	3.67	4.19	4.2

Tabla 4.4. Ancho de Banda de un sistema 8 x 4 con K buses $\rho = 0.5$
Comparación del modelo con resultados exactos o de simulación.

Este método de gran belleza tiene la ventaja de ser aplicable a cualquier problema de utilización de recursos pasivos, pero veremos en el punto siguiente que los resultados obtenidos por Ajmone Marsan y Gerla en AJM082a son mejores que los obtenidos por Jacobson y Lazowska.

4.3.3 Modelos aproximados de Ajmone Marsan.

Ajmone Marsan y Mario Gerla en AJM082a proponen 6 modelos diferentes basados en cadenas finitas aproximadas de Markov (obtenidas como una simplificación de la cadena exacta) que clasifican en base a la definición del estado de la cadena aproximada y a la forma de evaluar las velocidades de transición entre estados.

En los modelos numerados con 1 las velocidades de transición entre estados se evalúan asumiendo que los procesadores en cola de las memorias o buses pueden requerir acceso a cualquiera de las memorias cuando una memoria o bus queda libre, lo que, sin duda, sobreestima la utilización de los procesadores.

En los modelos numerados con 2 las velocidades de transición entre estados se evalúan conforme a un método que los autores denominan "de promediado".

Según este método, si los estados X_1, X_2, \dots, X_h de la cadena exacta se agrupan en el estado X de la cadena aproximada y los estados Y_1, Y_2, \dots, Y_n en el estado Y , entonces la velocidad de transición del estado X al estado Y viene dada por:

$$R (X \rightarrow Y) = 1/h \sum_{i=1}^h \sum_{j=1}^n R (X_i \rightarrow Y_j) \quad (4.5)$$

Este método de promediado sólo será exacto si las probabilidades estacionarias de los estados X_1, X_2, \dots, X_h son iguales. En caso contrario, la resolución será sólo aproximada.

En los modelos bautizados con la letra A el estado de la cadena se define como :

(n_m, n_q) , donde:

n_m = número de procesadores siendo servidos en memoria.

n_q = número de procesadores en cola por un bus o memoria.

En los modelos bautizados con la letra B a la definición de estado de los modelos A se agrega un "flag" f , de forma que:

$f = 0$ si no hay procesadores en espera por un bus

$f = 1$ si hay procesadores esperando un bus.

Ni que decir tiene que esta definición de estado es más completa, por lo que los resultados obtenidos con los modelos tipo B serán más exactos que los obtenidos con los modelos de tipo A, si bien el número de estados en los primeros será mayor que en los segundos y su resolución, por tanto, más compleja.

Por último, en los modelos denominados con la letra C, la definición de estado se establece simplemente con el número de procesadores en cola de un bus o memoria n_q .

Los autores estudian los modelos A1, A2, B2 y C2, proponiendo unas expresiones explícitas para el cálculo de las velocidades de transición entre estados para los modelos A1, A2 y C2, e incluso una expresión explícita de la utilización de los procesadores para el modelo C2 y que resulta ser:

$$NU = P = \sum_{i=1}^N \frac{e^{N-i} \frac{N!}{(i-1)!} \prod_{k=1}^{N-i} \beta_k^{-1}}{1 + \sum_{j=0}^{N-1} \left(e^{N-j} \frac{N!}{j!} \prod_{k=1}^{N-j} \beta_k^{-1} \right)} \quad (4.6)$$

con

$$\beta_K = \frac{\sum_{j=1}^{K-1} j p_j(i) + K \sum_{j=0}^{i-K} (p_K(j+K) p_{M-K}(i-2K-j+M))}{\sum_{j=1}^{K-1} p_j(i) + \sum_{j=0}^{i-K} (p_K(j+k) p_{M-K}(i-2K-j+M))} \quad (4.7)$$

$$i \geq 1$$

siendo $\rho = \lambda / \mu$ y estando $p_j(i)$ definido por la expresión recurrente (3.5).

La comparación de estos modelos con resultados procedentes de simulación muestra que cualquiera de estos cuatro modelos es una excelente aproximación del comportamiento de un sistema multibus.

Como era lógico esperar, el modelo B2 es el mejor de todos ellos sobre todo para elevados valores de conflicto, si bien la elevada complejidad de su resolución aconseja los modelos más simples (A1 o C2) para valores elevados de N y M.

	EXACTO	A1	A2	B2	C2
0.001	3.996	.00	.00	.00	.00
0.01	3.960	.00	.00	.00	.00
0.1	3.613	.00	.00	.00	-.18
0.3	2.948	.06	.11	.04	-.82
0.5	2.440	.25	.42	.15	-1.28
1.0	1.651	1.00	1.73	.58	-1.69
3.0	0.685	3.13	5.67	1.78	-1.91
5.0	0.428	3.94	7.30	2.23	-2.36
10.0	0.220	4.63	8.78	2.63	-2.36
100.0	0.023	5.26	10.25	3.01	-2.77

Tabla 4.5. Comparación de los modelos para un sistema 4 x 4 con 3 buses.

En la tabla 4.5 tomada de AJM082a se refleja el comportamiento de los 4 modelos listando los tantos por ciento de diferencia en la utilización de los procesadores respecto de resultados exactos para un sistema 4 x 4 con 3 buses y diferentes valores de ρ .

Los mismos autores estudian por simulación el efecto de considerar constantes los tiempos de servicio de las memorias observando el resultado ya conocido en la teoría de colas de que aumentan los anchos de banda y los tiempos de utilización de los procesadores, si bien para valores reducidos de ρ ($\rho < 1$.) el aumento es muy pequeño.

4.4 SISTEMAS CON BUSES PARCIALES.

Los sistemas basados en buses presentan frente a los sistemas "cross-bar" un nivel de redundancia muy interesante, pero cuando el número de buses es elevado (K del orden de $N/2$), la complejidad de un sistema multibus es comparable a la de un sistema "cross-bar".

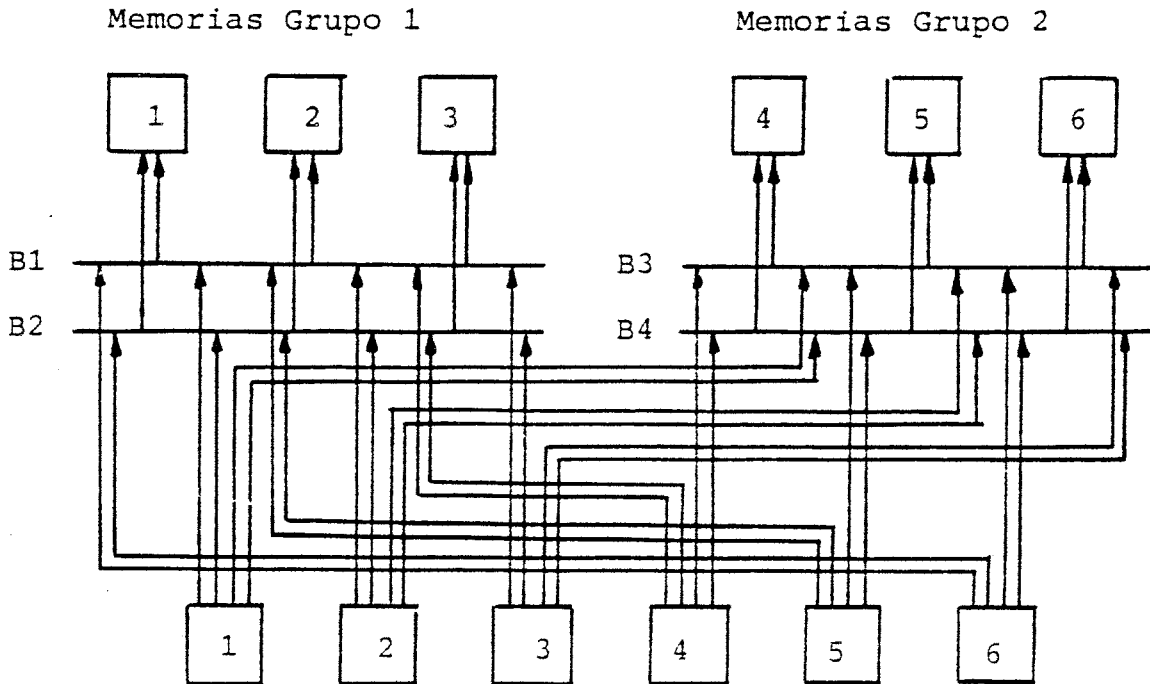


Fig. 4.12. Sistema 6 x 6 con dos grupos de 2 buses parciales.

T. Lang, M. Valero e I. Alegre [LANG83] proponen una arquitectura basada en varios grupos de buses parciales. En esta arquitectura (fig. 4.12), las memorias se dividen en g grupos de M/g memorias cada una entrelazadas por K/g buses; por contra, cada procesador se conecta a todos los K buses.

El análisis tanto en el caso discreto como continuo de estos sistemas es posible a partir de una cadena finita de Markov. Nuevamente el elevado de número de estados de esta cadena exacta

hace inviable una solución de este tipo.

No conocemos una aproximación analítica al estudio de estos sistemas.

Para el caso discreto, adoptando una definición de estado de la cadena aproximada como $(n_{q1}, n_{q2}, \dots, n_{qg})$, siendo n_{qi} el número de procesadores en cola para una memoria o bus en el grupo de buses parciales i , y asumiendo que, en cada grupo, los procesadores en cola se distribuyen entre las memorias conforme a (3.53), hemos comprobado que los resultados obtenidos aproximan bastante los resultados exactos.

En la tabla 4.6 puede observarse el comportamiento del modelo mediante la comparación con los anchos de banda obtenidos por simulación en LANG83 para $m = .5$ y $m = 1$.

N	4				8				12			
	m = 0.5		m = 1		m = 0.5		m = 1		m = 0.5		m = 1	
	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.
1+1	1.74	1.79	1.50	1.50	1.87	1.91	1.83	1.84	1.92	1.94	1.90	1.93
2+2	2.62	2.73	1.77	1.78	3.61	3.66	3.11	3.08	3.81	3.83	3.64	3.64
3+3					4.72	4.78	3.44	3.42	5.52	5.56	4.76	4.73
4+4					4.93	5.11	3.47	3.45	6.77	6.79	5.10	5.07
5+5									7.24	7.32	5.16	5.13
6+6									7.28	7.43	5.16	5.14

Tabla 4.6. Comparación del Modelo (MOD.) con resultados de Simulación (SIM.). $N = M$.

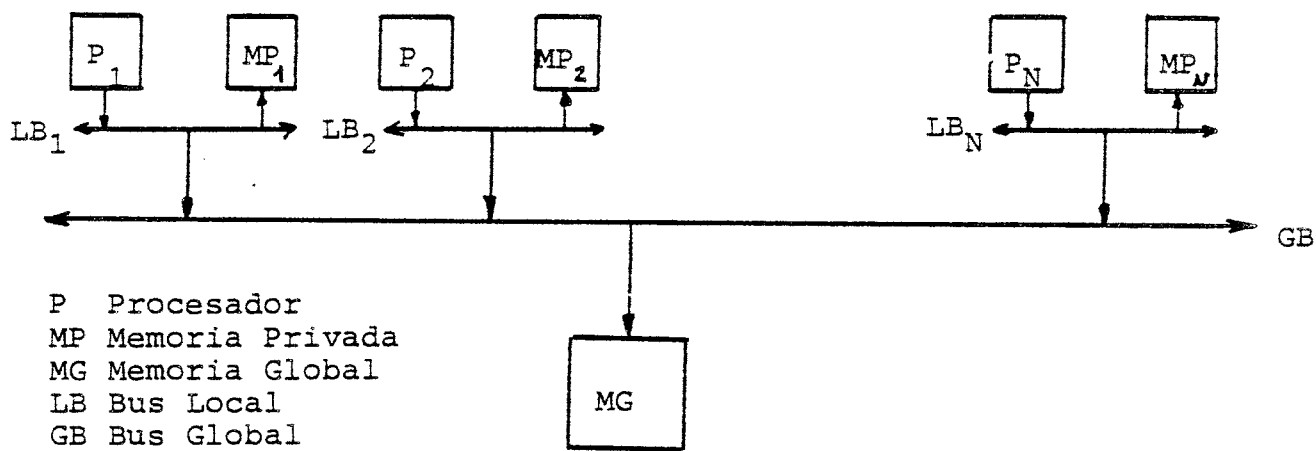


Fig. 4.13

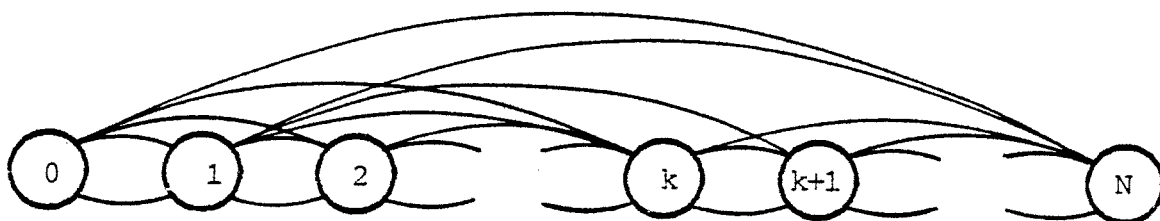


Fig. 4.14

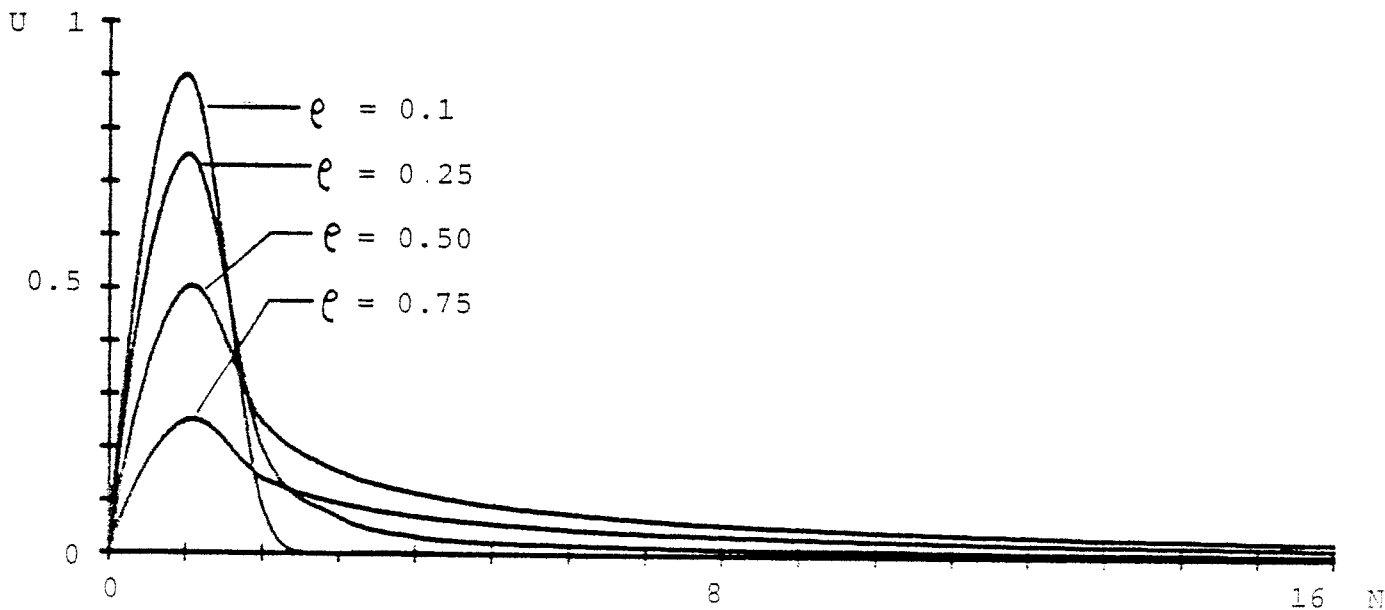


Fig. 4.15

4.5 SISTEMAS CON UN UNICO BUS.

El estudio discreto de un sistema con un único bus como el representado en la figura 4.13 da origen a una cadena discreta de Markov muy simple (fig. 4.14) en la que el estado queda simplemente representado por el número de procesadores en cola por el bus o memoria común.

Las velocidades de transición entre estados viene dada por:

$$P_{0j} = P_{1j} = \binom{N}{j} m^j (1 - m)^{N-j} \quad (4.8)$$

$$P_{ij} = \begin{cases} \binom{N-i+1}{j-i+1} m^{j-i+1} (1-m)^{N-j} & j = i-1, i, \dots, N \\ 0 & j < i-1 \end{cases} \quad (4.9)$$

donde m es la probabilidad de acceso por ciclo.

Esta cadena es fácilmente resoluble en forma exacta y la figura 4.15 muestra la utilización de los procesadores al variar su número para diferentes valores de m .

Una simple mirada a la figura pone de manifiesto la rápida saturación del bus incluso para bajos valores de m , lo que nos

indica que esta arquitectura, si bien de una gran simplicidad, no es en absoluto apropiada para sistemas fuertemente cargados (elevados N o m).

Desde el punto de visto continuo es posible efectuar un análisis más general.

Admitiendo que el tiempo que un procesador ejecuta en memoria privada se encuentra exponencialmente distribuido con media $1/\lambda$ para todos ellos y, admitiendo una distribución arbitraria de los tiempos de servicio requeridos por cada procesador (pero con igual distribución para cada uno de ellos), el sistema puede ser modelado por una cola clásica $M/G/1$ cuya solución exacta puede ser obtenida por aplicación del método de la variable suplementaria [JAIS68] o por el método de la cadena de Markov embebida [KLEIN75]. La probabilidad estacionaria de que haya k procesadores en cola viene dada por:

$$\pi_k \begin{cases} 0 & k > N \\ \sum_{r=k}^N (-1)^{r-k} \binom{r}{k} S_r^* & 0 \leq k \leq N \end{cases} \quad (4.10)$$

donde:

$$S_r^* = \begin{cases} 1 & r = 0 \\ \frac{N C_{r-1}}{r} \frac{\sum_{j=r-1}^{N-1} \binom{N-1}{j} \frac{1}{C_j}}{1 + N \frac{\lambda}{\mu} \sum_{j=0}^{N-1} \binom{N-1}{j} \frac{1}{C_j}} & 1 < r \leq N \end{cases} \quad (4.11)$$

con

$$C_j = \begin{cases} 1 & j = 0 \\ \prod_{i=1}^j \frac{\phi(i\lambda)}{1 - \phi(i\lambda)} & j \neq 0 \end{cases} \quad (4.12)$$

siendo $\phi(s)$ la transformada de Laplace de la distribución de los tiempos de servicio.

El ancho de banda del sistema vendrá dado por:

$$B = 1 - \pi_0 = 1 - \frac{1}{1 + N \frac{\lambda}{\mu} \sum_{i=0}^{N-1} \binom{N-1}{i} \frac{1}{C_i}} \quad (4.13)$$

y el número medio de procesadores ejecutando en memoria privada será $P = B \mu / \lambda$ por aplicación de la Ley de Little.

Nótese que, dado que todos los procesadores son indistinguibles, el arbitraje de las memorias es, en este caso, irrelevante.

Para el caso en que los tiempos de servicio son también exponencialmente distribuidos el modelo anterior se reduce a la cola M/M/1 también denominado "modelo del reparador de máquinas" de Palm [PALM58], en cuyo caso, el número medio de procesadores ejecutando en memoria privada vendrá dado por:

$$P = \frac{\sum_{k=0}^N e^{-\rho} \frac{N!}{(N-k)!} - 1}{e^{-\rho} \sum_{k=0}^N e^{-\rho} \frac{N!}{(N-k)!}} \quad (4.14)$$

con $\rho = \lambda / \mu$.

En la figura 4.16 se representa el número medio de procesadores activos frente al número de procesadores N para diferentes valores de ρ en los casos de tiempo de servicio exponencialmente distribuido (obtenido de la fórmula 4.14) y constante (obtenido de 4.10 con $\vartheta(s) = e^{-s/\mu}$). De la comparación de ambas curvas puede apreciarse que la distribución de los tiempos de servicio no tiene una gran importancia en el comportamiento del sistema, sobre todo cuando la carga éste es débil (ρ bajos).

Nuevamente puede apreciarse el efecto de saturación del bus cuando N aumenta.

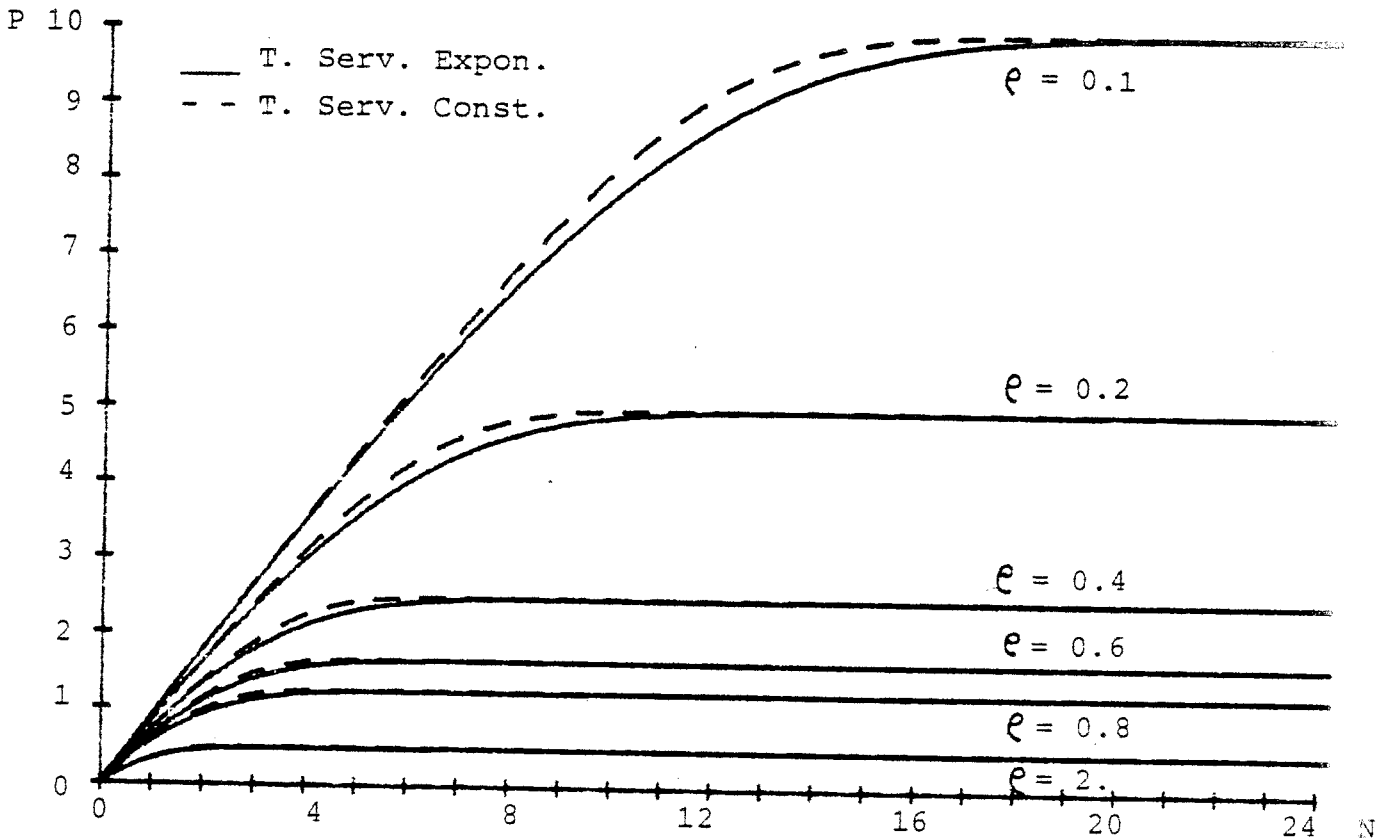


Fig. 4.16.

Ajmone Marsan [AJM081b] y Protopapas [PROT80] estudian el caso en que los procesadores presentan cargas diferentes al sistema. Ambos consideran tiempos de procesamiento en memoria privada y tiempos de servicio en las memorias exponencialmente distribuidos y disciplina FIFO.

El primero de ellos simplifica el problema asumiendo que los tiempos de servicio se distribuyen de igual forma para todos los procesadores y plantea una cadena finita de Markov para el cálculo exacto. Ahora no basta para definir el estado del sistema el número de procesadores en la cola del bus o memoria, sino que es necesario también conocer en qué orden han llegado los procesadores a la cola, ya que cada procesador tiene un comportamiento di-

ferente.

El número de estados de la cadena es extraordinariamente alto, incluso para valores de N reducidos:

$$\sum_{j=0}^N N! / (N-j)! \quad (4.15)$$

por lo que el autor se limita al estudio de los casos más simples ($N = 2$ y $N = 3$).

El segundo de los autores citados plantea la resolución basándose en un análisis de colas que le lleva a la conclusión de que el tiempo medio de espera del procesador i en la cola viene dado por:

$$W_{Qi} = \left(\frac{1}{\lambda_i} \right) \left(\frac{Z_N}{Z_N^{(i)}} - 1 \right) - \frac{1}{\mu_i} \quad (4.16)$$

siendo:

$Z_N = 1 + \sum_{k=1}^N k! \times$ (Suma de los productos de k factores de carga ρ_i extendido a todas las posibles combinaciones de los N factores).

y $Z_N^{(i)} = (Z_N) e_{i=0}$

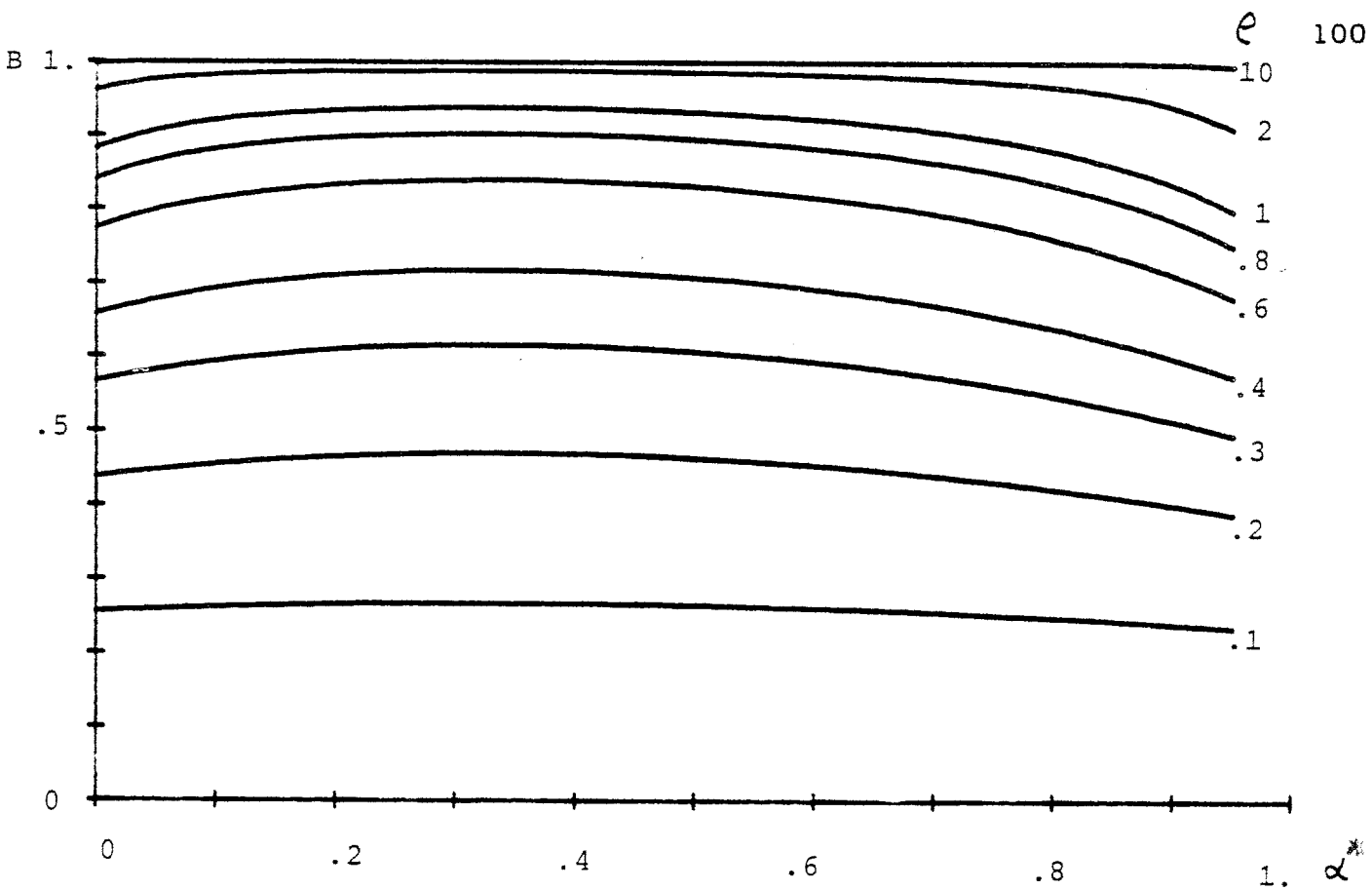


Fig. 4.17

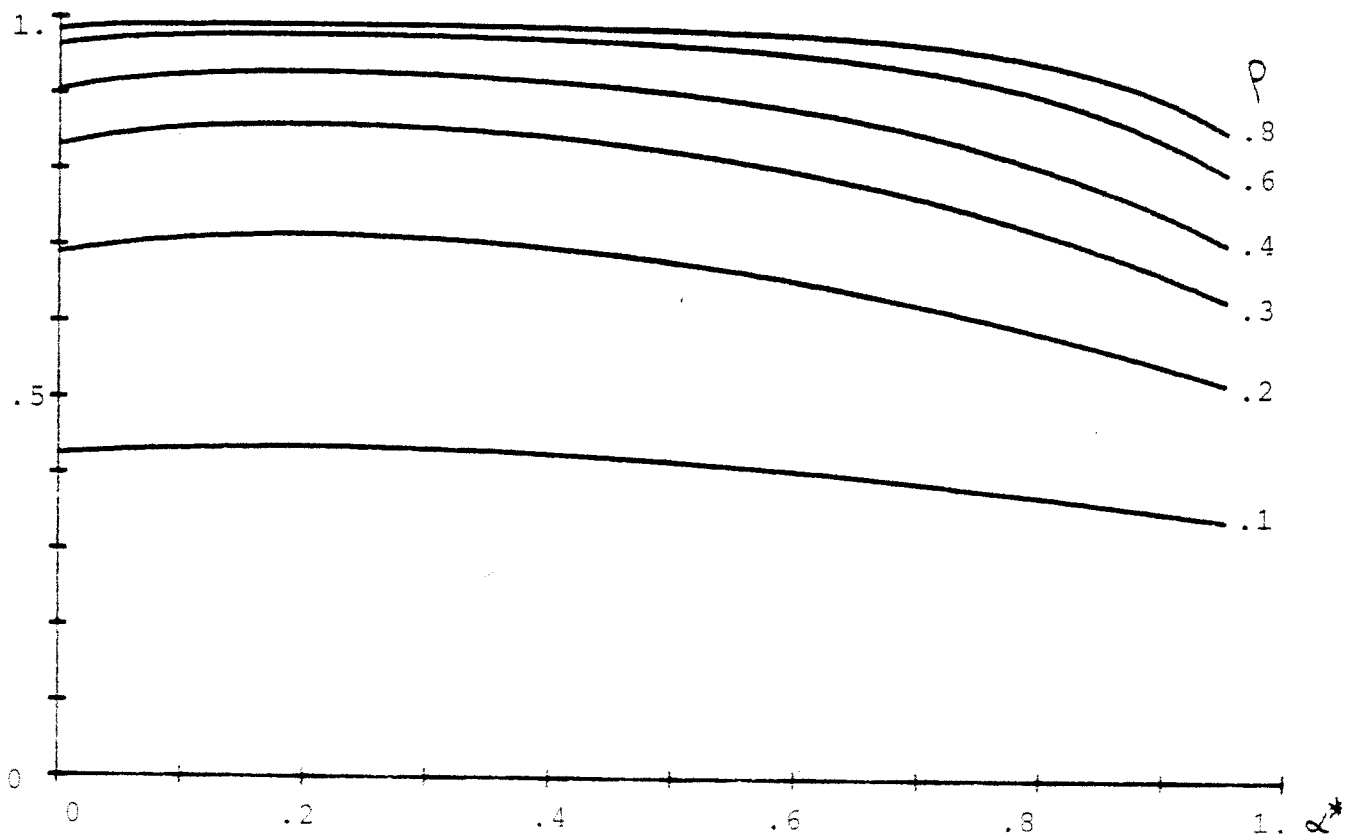


Fig. 4.18

En las figuras 4.17 y 4.18 se representan el número medio de procesadores activos para un sistema de 3 y 5 procesadores respectivamente, por aplicación del segundo de los métodos para:

$$\mu_i = \mu \quad ; \quad i = 1, 2, \dots, N \quad \text{y} \quad \lambda_i = \lambda \quad ; \quad i = 2, 3, \dots, N$$

$$\text{donde } \alpha^* = \rho_1 / \sum_{i=1}^N \rho_i$$

De estas figuras puede deducirse que:

los mayores anchos de banda se consiguen para sistemas con carga uniforme ($\alpha^* = 1/N$). De todas formas, se puede observar que el efecto de carga no uniforme sólo es significativo para valores elevados de ρ .

Por último otros autores ([MARK84, KINN78]) han estudiado los sistemas con un único bus para cargas determinísticas, es decir, con tiempos de ejecución en memoria privada y tiempos de servicio constantes.

Este es claramente un caso en el que el sistema al cabo de un número determinado de ciclos alcanzará un conjunto de estados que se repetirán cíclicamente.

Kinney y Arnold [KINN78] consideran el caso de una tarea que es subdividida en N tareas independientes las cuales ejecutan en paralelo en N procesadores unidos por un único bus.

Cada una de las tareas i es ejecutada cíclicamente y requiere en cada ciclo E_i unidades de ejecución en memoria privada seguidos de I_i unidades de uso del bus.

Si E es el número de unidades de ejecución en memoria privada de la tarea original e I el número de unidades de uso del bus requerido por la misma, en general, como consecuencia del reparto en N subtareas:

$$\sum E_i = (1 + \beta)E \text{ y } \sum I_i = (1 + \alpha)I \quad (4.17)$$

β y α representan respectivamente el aumento global en los tiempos de ejecución en memoria privada y de uso del bus debido al reparto de la tarea original en N subtareas independientes.

Los autores determinan una cota superior para el número medio P de procesadores ejecutando en memoria privada y establecen el número mínimo de procesadores requeridos para alcanzar esta cota superior en función de β , α y E/I .

Los autores observan asimismo que si predomina el aumento de tiempos de ejecución en memoria privada ($\beta \gg \alpha$), entonces la cota superior P se ve limitada por el valor de β y el número mínimo de procesadores necesarios para alcanzar esta cota superior es mayor que en el caso $\beta = 0$ y $\alpha = 0$.

Si, por el contrario, predomina el aumento de los tiempos de uso del bus ($\alpha \gg \beta$), entonces la cota superior de P es muy pequeña en comparación con la obtenida para el caso $\alpha = \beta = 0$ y el

número mínimo de procesadores necesarios para alcanzar esta cota es también muy pequeño, lo que es debido a la rápida congestión del bus.

Markenscoff [MARK84] estudia el caso de dos tareas secuenciales, la primera de las cuales puede ser subdividida en N subtareas que ejecutan en paralelo en N procesadores y las cuales envían los datos al procesador (o procesadores) que ejecutan la segunda tarea a través de un único bus global.

A fin de aumentar el tiempo de respuesta global del sistema se impone la restricción de que un procesador no puede enviar la información correspondiente al $(k+1)$ ésimo ciclo en tanto no hayan sido enviados todos los datos correspondientes al k ésimo ciclo.

El autor determina el conjunto de estados que constituyen el comportamiento periódico hacia el que tiende el sistema y establece una cota superior para el número de ciclos que tarda el sistema en alcanzar este comportamiento periódico en los dos casos posibles de bus saturado y bus no saturado.

4.6 OTROS SISTEMAS CON UN UNICO BUS.

Ajmone Marsan y otros estudian en AJM081b, AJM082b y AJM083 cuatro arquitecturas basadas en un único bus y que se representan en las figuras 4.19 a 4.22.

Los autores asumen que el tiempo de ejecución en memoria privada de los procesadores así como el tiempo de servicio requerido en las memorias están exponencial e idénticamente distribuidos con media $1/\lambda$ y $1/\mu$ respectivamente.

El análisis de estos sistemas es realizado admitiendo que la comunicación entre procesadores se realiza mediante mensajes depositados y recogidos en módulos de memoria común.

La arquitectura 1 es ya bien conocida y pueda ser modelada por una cola M/M/1 y resuelta mediante la fórmula (4.14).

En la arquitectura 3 la memoria local de cada procesador es sustituida por una memoria doble puerta la cual permite el acceso simultáneo de su procesador asociado a través del bus local y de otro procesador a través del bus global sin introducir conflicto.

Bajo esta suposición, el modelo 3 puede ser igualmente modelado por un sistema de colas M/M/1 y resuelto mediante la fórmula (4.14).

En la arquitectura 2, a cada procesador se le asocia una memoria local y un módulo de memoria común en el cual recoge los mensajes a él destinados. Con vistas a mejorar las prestaciones del modelo, si otro procesador en posesión del bus global pretende acceder al módulo de memoria común de un procesador que ejecuta en sus memoria privada, este último se bloquea debiendo esperar a que el primero termine su acceso.

Arquitectura 1

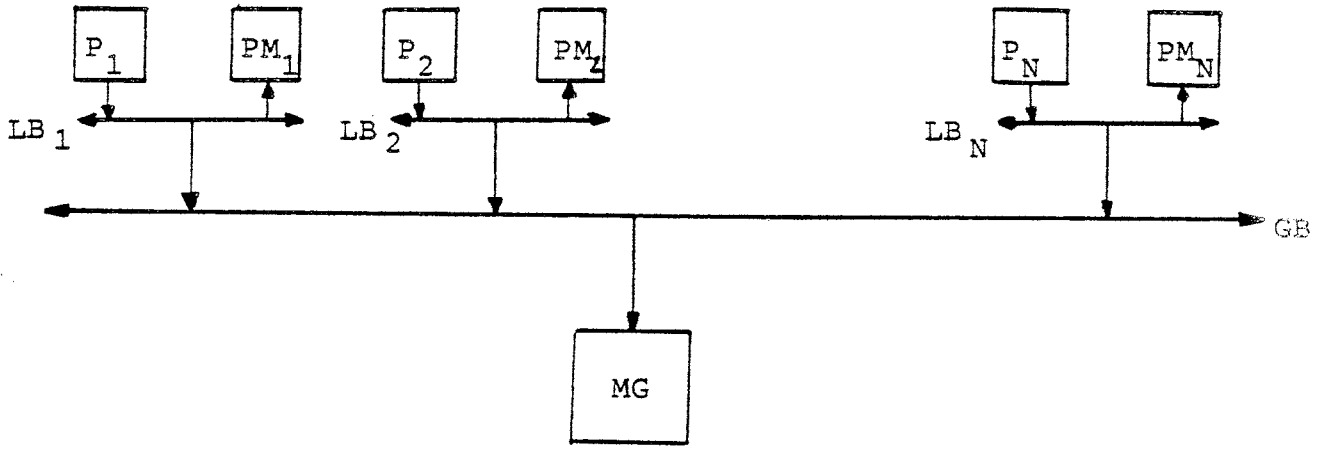


Fig. 4.19

Arquitectura 2

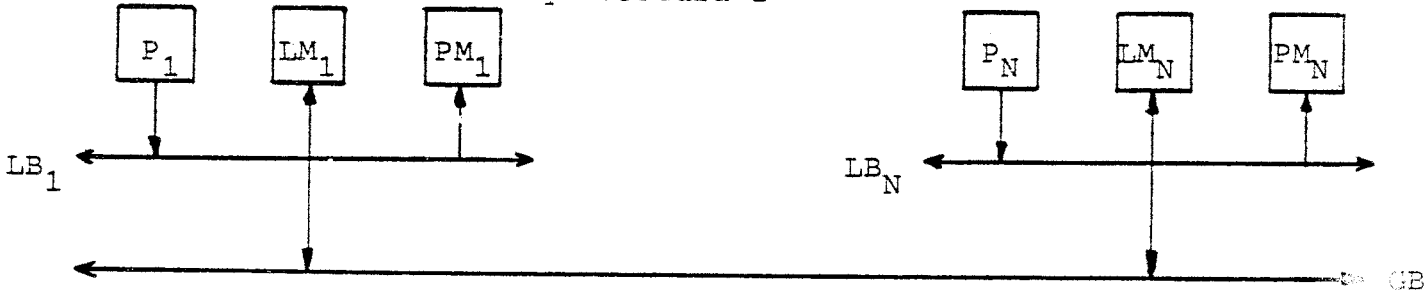


Fig. 4.20

Arquitectura 3

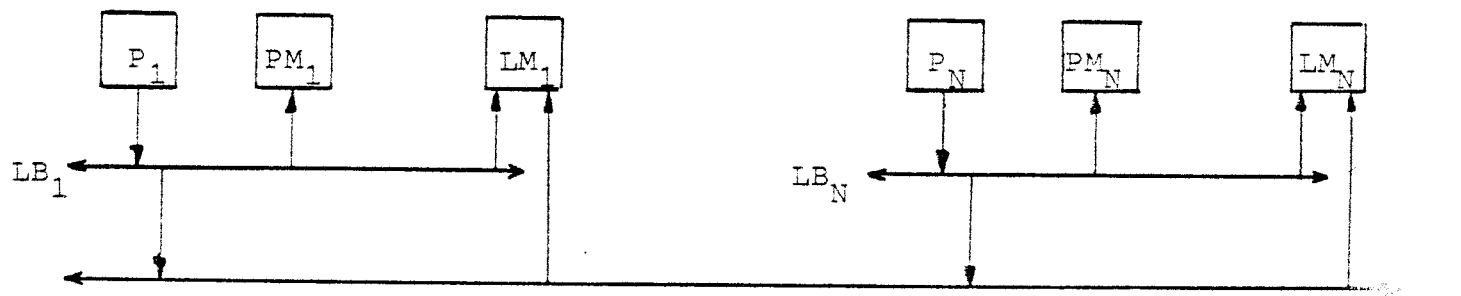


Fig. 4.21

Arquitectura 4

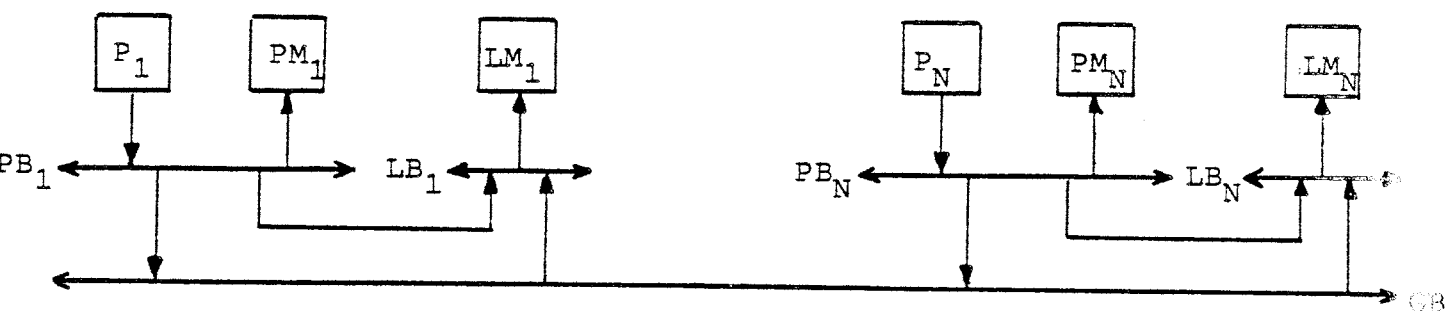


Fig. 4.22

P = Procesador

PB = Bus Privado

PM = Memoria Privada

LB = Bus Local

LM = Memoria Local

GB = Bus Global

MG = Memoria Global

Por último, la arquitectura 4 presenta una mejora de la arquitectura 2 en el sentido de que el acceso a través del bus global al módulo de memoria común de otro procesador no requiere el uso de su bus local. Aparte del conflicto originado en el acceso al bus global, también puede originarse conflicto en el acceso simultáneo de un procesador a través de su bus local a su módulo de memoria común asociado y de otro procesador al mismo módulo a través del bus global. Para aumentar las prestaciones del sistema este tipo de conflicto siempre es resuelto a favor del procesador que ocupa el bus global.

El análisis de estos dos últimos sistemas es realizado por los autores a partir de cadenas exactas finitas de Markov asumiendo un árbitro neutral del bus global.

Nótese que una comunicación mediante mensajes requiere dos accesos a memoria a común por mensaje:

1. El primer acceso para llevar el mensaje al módulo de memoria común, tarea que realiza el procesador que envía el mensaje y
2. el segundo acceso para recoger el mensaje de memoria común, tarea que lleva a cabo el procesador que recibe el mensaje.

Por tanto, si λ_p es la velocidad con que un procesador genera un mensaje, para los modelos 1 y 4 $\lambda = 2 \lambda_p$ y para los modelos 2 y 3 $\lambda = \lambda_p / (\lambda_p + \mu)$.

Los autores comparan los resultados obtenidas para cada una de estas cuatro arquitecturas diferentes observando que la arquitectura 1 es, con mucho, la peor de ellas, saturándose el bus rápidamente. Las arquitecturas 3 y 4 son, por el contrario, las más bondadosas, si bien la arquitectura 3 supone que el acceso simultáneo a memoria doble puerta de dos procesadores no introduce ningún retraso y esto en un sistema real es una hipótesis muy optimista.

CAPITULO 5

REDES DE INTERCONEXION.

5.1 INTRODUCCION.

En los últimos 20 años han sido propuestas un elevado número de redes de interconexión procesador-memoria en sistemas multi-procesadores. Estas redes han sido recogidas y clasificadas por Feng en FENG81.

Atendiendo a su topología, Feng las clasifica en Estáticas y Dinámicas.

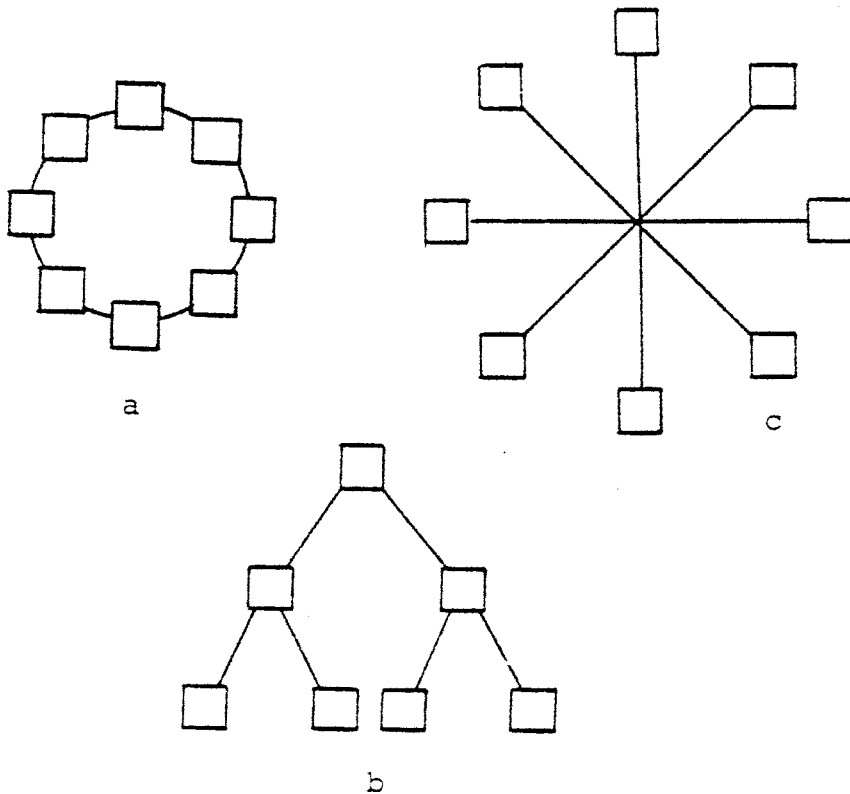


Fig. 5.1

Las redes estáticas se caracterizan porque sus líneas de interconexión son pasivas y no pueden ser reconfiguradas. Ejemplos en dos dimensiones de estas redes son las configuraciones en anillo (fig. 5.1a), árbol (fig. 5.1b) y estrella (fig. 5.1c).

Estas redes son normalmente usadas en sistemas geográficamente distribuidos para la interconexión de grandes redes de ordenadores [WITTS1] y no son habitualmente usadas en la interconexión procesador-memoria por lo que no las vamos a considerar.

Por el contrario, las redes dinámicas se caracterizan porque sus conexiones se realizan a través de selectores que pueden ser reconfigurados.

Dentro de estas redes dinámicas podemos distinguir asimismo varias categorías:

- Sistemas Crossbar.
- Sistemas Multibuses.
- Sistemas con una única etapa de selectores y
- Sistemas con varias etapas de selectores.

Los sistemas Cross-bar y Multibuses han sido ampliamente estudiados en los capítulos anteriores, por lo que ahora nos centraremos en los restantes tipos.

De una forma general, un selector "a x b" es un dispositivo con "a" entradas y "b" salidas que posee la capacidad de conectar cualquiera de sus entradas a cualquiera de sus salidas. Si las

salidas se numeran en la forma $0,1,2,\dots,b-1$, una entrada es conectada a la salida d si el dígito de control aportado por esa entrada vale d , donde d es un dígito en base b .

Si varias entradas intentan acceder a la misma salida, se originará un conflicto, el selector elige una sólo de ellas y rechaza las demás.

En la figura 5.2 se muestra un selector 2×2 representándose se todas las posibles configuraciones que puede adoptar.

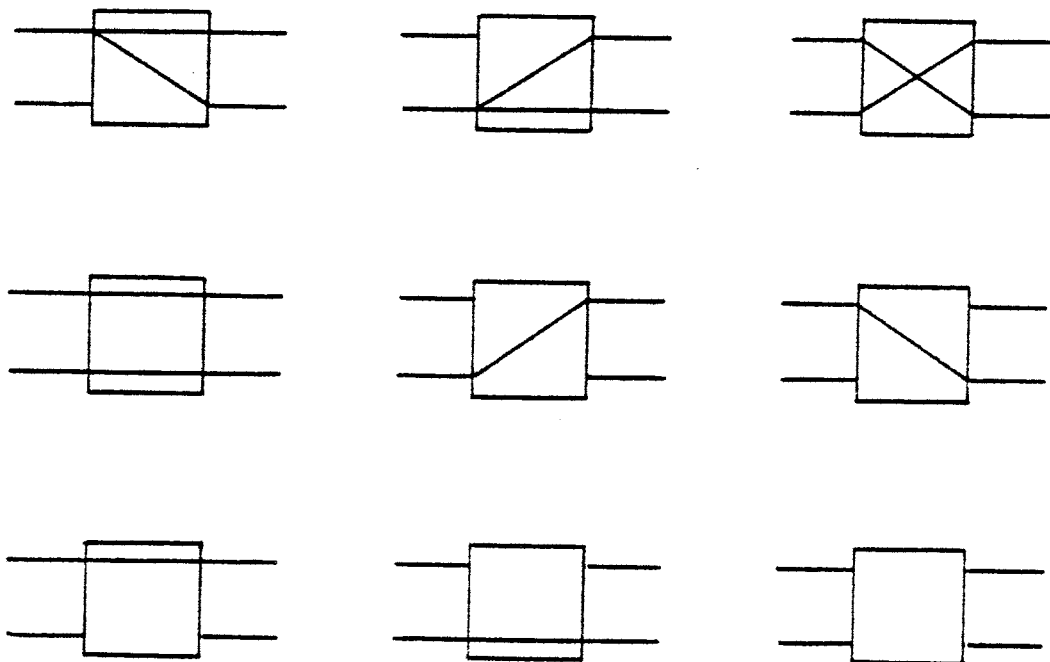


Fig. 5.2.

Aquellos sistemas en que los procesadores se unen con las memorias a través de una única etapa de selectores se denominan sistemas de una sola etapa. En la figura 5.3a se representa una red con una única etapa para un sistema 8×8 con selectores 2×2 .

2, denominada "Shuffle-Exchange" y definida por Stone en STON71, en donde las salidas del selector se han numerado en el mismo orden que sus entradas.

Es fácil ver que una tal red no permite una interconexión completa, es decir no es posible encontrar una configuración de sus selectores que permita la conexión de un procesador a cualquier memoria.

Para que esta conexión completa sea posible, estas redes se hacen recirculantes, es decir, la información atraviesa la red varias veces aportando de cada vez un dígito distinto de control lo que convierte, de hecho, a estas redes en redes multietapa (fig. 5.3b).

Un estudio detallado basado en simulación de estos sistemas recirculantes para diferentes políticas de arbitraje de conflictos puede encontrarse en CHEN81.

Las redes multietapa son redes de interconexión que constan de varias etapas de selectores. Las diferentes formas en que estas etapas pueden estar constituidas y las diferentes formas en que pueden interconectarse da origen a una amplia gama de redes diferentes que, a su vez, pueden ser clasificadas en:

- Redes Libres de Conflicto por su propia Naturaleza.
- Redes Libres de Conflicto por Reconfiguración.
- Redes con Conflicto.

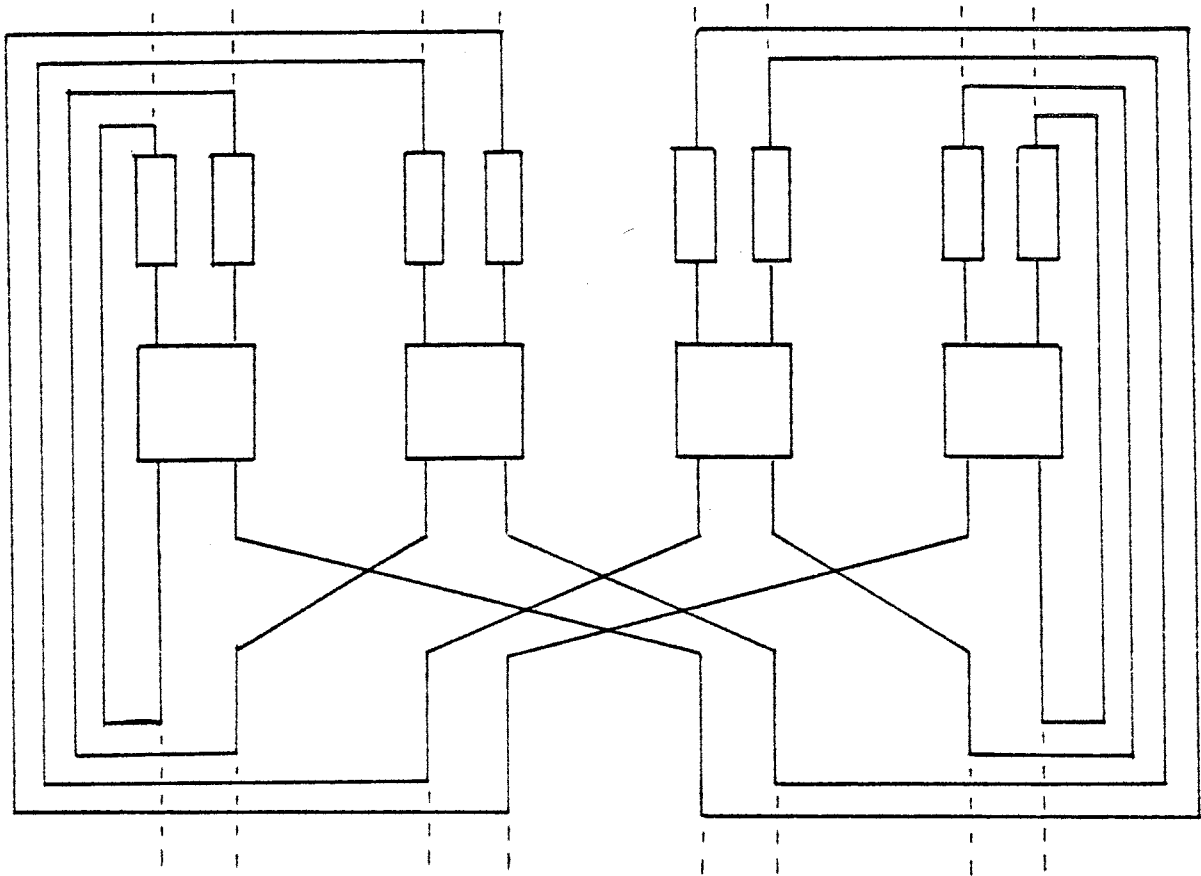


Fig. 5.3.b

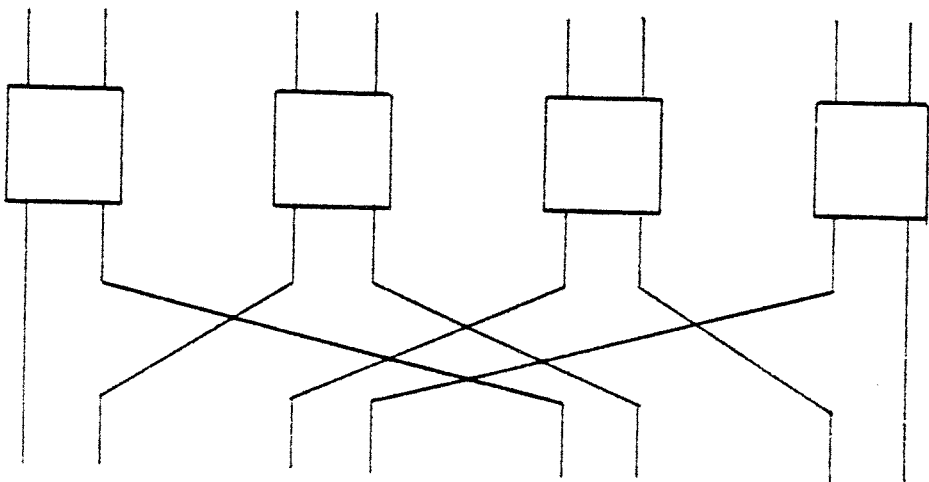


Fig. 5.3.a

Las Redes Libre de Conflicto por su Naturaleza son aquellas que permiten cualquier conexión simultánea entre cualquier procesador y cualquier memoria. Estas Redes pueden ser punto a punto o multipunto [THOM78], [GEC77]. En cualquier caso, estas redes tienen el grave inconveniente de su gran complejidad y elevado número de componentes. Desde el punto de vista de sus prestaciones, estas redes por sí mismas no introducen conflicto, por lo que el ancho de banda de la interconexión procesador-memoria se verá tan sólo disminuido por los conflictos producidos en el propio acceso a las memorias y será el mismo que el de un sistema "cross-bar" (despreciando el efecto del retraso introducido al atravesar múltiples etapas).

Las Redes Libre de Conflictos por Reconfiguración también permiten cualquier conexión simultánea procesador-memoria pero con la diferencia de que el establecimiento de una nueva interconexión puede originar una reconfiguración de toda la red. Entre estas redes se encuentran las denominadas Redes de Benes [LENF78], [NASS81]. En la figura 5.4 se representa una red de Benes para una interconexión 8 x 8. Ni que decir tiene que los algoritmos de control y arbitraje de este tipo de redes son muy complejos, si bien reducen el número de componentes frente a las redes Libres de Conflicto por su propia Naturaleza.

El ancho de banda de estas redes (despreciando igualmente el efecto del retraso debido al paso por múltiples etapas y a los complejos algoritmos de control y reconfiguración) será nuevamente el de un sistema "cross-bar".

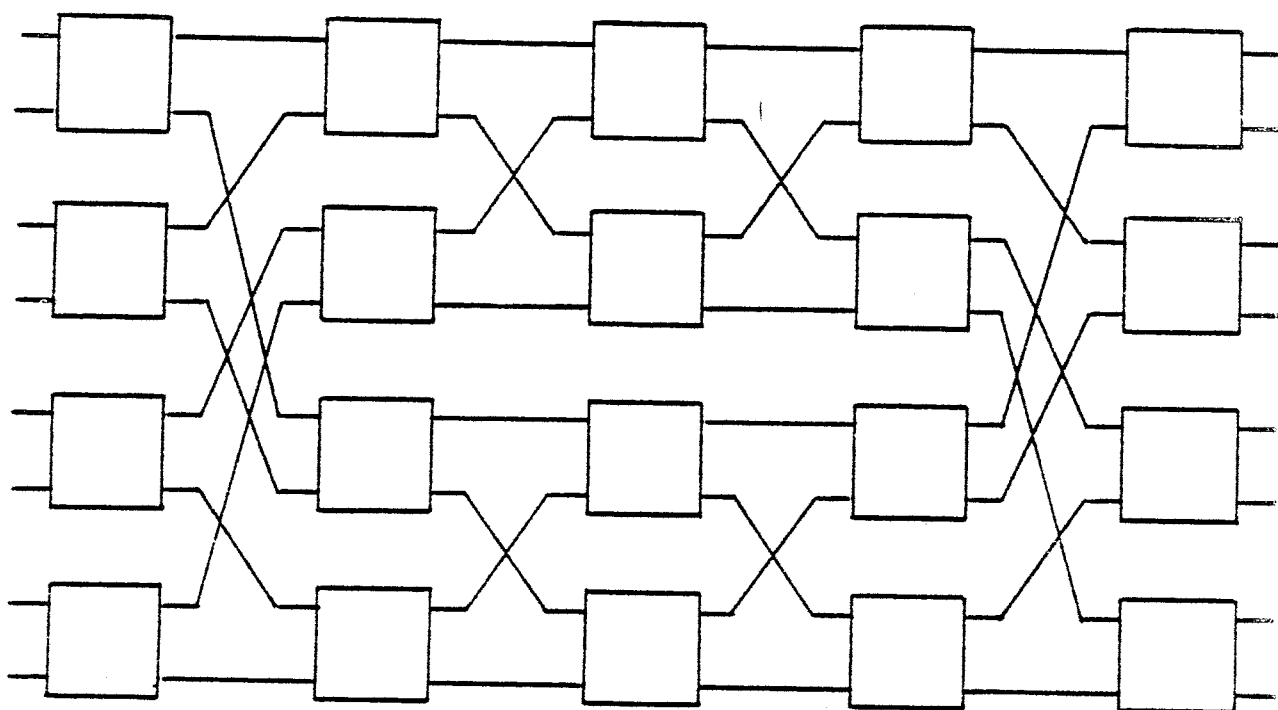


Fig. 5.4. Red de Benes 8 x 8.

Las Redes con Conflicto se caracterizan porque no es posible la conexión simultánea entre cualquier procesador y cualquier memoria. Cada procesador tiene un camino para alcanzar cada memoria pero parte de este camino puede atravesar el camino de conexión de otro procesador con otra memoria. Si ambas conexiones pretenden establecerse simultáneamente, se originará un conflicto y un mecanismo de arbitraje dejará establecer tan sólo una de ellas.

Las más populares entre éstas son las redes de Banyan que, de una manera informal, pueden definirse como redes multietapa en las que existe un único camino entre cada procesador y cada memoria. Una definición más exacta puede obtenerse en [GOKE73].

Multitud de este tipo de redes han sido propuestas: Baseline [WU80a], SW Banyan, Manipuladores de Datos [SIEG80], Omega

[LAWR75], n-Cubo [PEAS77] delta [PATE81], etc.

Entre ellas se diferencian por sus algoritmos de "routing" y por las diferentes permutaciones que pueden realizar.

En WU80a y WU80b se demuestra la equivalencia topológica entre estas redes, entendiéndose por dos redes topológicamente equivalentes aquellas en las que por una simple reenumeración de sus componentes se hacen iguales.

Dado que estas redes son topológicamente equivalentes nos fijaremos en una de ellas que, en cierto modo, las engloba. Esta es la red delta definida por Patel en PATE79 y PATE81 y que es un caso particular de la Red Shuffle Generalizada (GSN) definida por Bhuyan [BHUY83].

Es de notar aquí que el hecho de ser topológicamente equivalentes no hace necesariamente que sus prestaciones sean iguales, si bien los estudios realizados al efecto [KRUS83] parecen afirmar que ello es cierto.

Una red delta $a^n \times b^n$ es una red multietapa que interconecta $a^n = N$ procesadores con $b^n = M$ memorias a través de $N \log_b M$ selectores $a \times b$ distribuidos en n etapas y de forma que:

- 1.- Es posible encontrar un camino entre cada procesador y cada memoria.
- 2.- Este camino es único, de longitud constante e igual para cada procesador y cada memoria y

3.- Ninguna de las entradas y salidas de los selectores es desaprovechada.

Diferentes configuraciones de esta red delta pueden ser construidas. Patel en PATE81 demuestra que una red delta puede ser construida mediante una secuencia a-shuffle entre cada etapa (fig. 5.5).

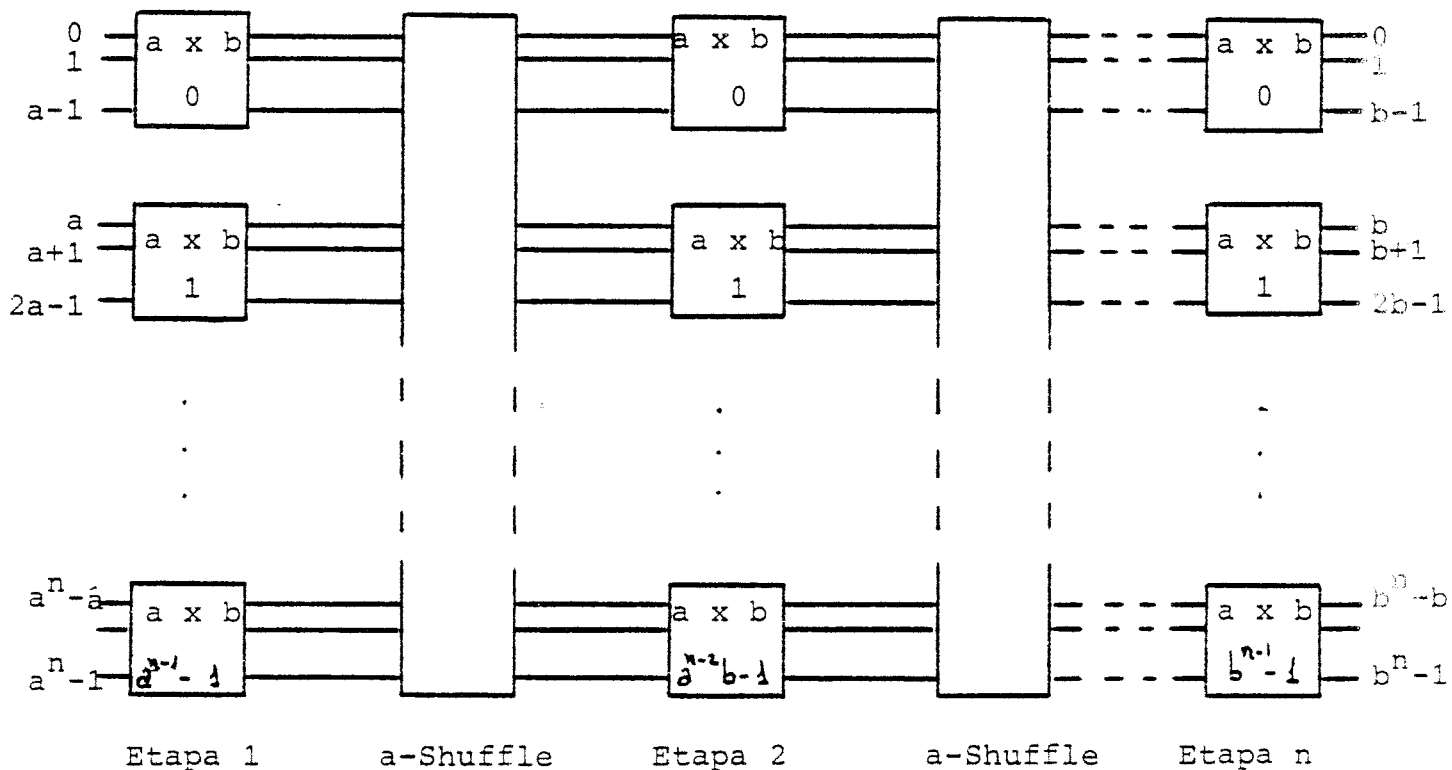


Fig. 5.5 Red delta $a^n \times b^n$ construida mediante una secuencia de a-shuffles.

La red delta de la figura 5.5 para el caso $a=b=2$ es la red omega definida por Lawrie en LAWR75, la cual es topológicamente equivalente a otras muchas redes tal como queda constancia en WU80a y WU80 b.

La red delta presenta, no obstante, algunas ventajas respecto a otras redes y es la simplicidad de su "routing", dado que los propios dígitos de las direcciones actúan directamente como los dígitos de control de los selectores, lo que la hace preferible a las demás.

5.2 MODELO DE UNA RED DELTA.

Si admitimos que las peticiones no atendidas en un ciclo son simplemente eliminadas, el modelo de una red delta $a^n \times b^n$ se reduce al modelo de un sistema crossbar tal como demostró Patel en PATE79 y PATE 81.

Cada etapa de una red delta está controlada por un dígito de las direcciones. Dado que las direcciones de cada una de las peticiones se suponen independientes y uniformemente distribuidas, también los serán sus dígitos. Entonces, bajo la hipótesis de Patel, desde el punto de vista de la etapa i , la etapa $i + 1$ se comporta como un sistema crossbar $a \times b$.

Sea m la probabilidad de acceso por ciclo de los procesadores. De cara a los procesadores, la etapa 1 se comporta como un sistema crossbar $a \times b$ y, por tanto, el número medio de peticiones que atraviesan la primera etapa por cada línea de salida será B_{ab}^m/b , donde B_{ab}^m es el ancho de banda de un sistema crossbar $a \times b$ con probabilidad m de acceso por ciclo.

De cara a la primera etapa, la segunda se comporta como un sistema crossbar a x b con una probabilidad de acceso por ciclo:

$$m_2 = B_{ab}^m / b \quad (5.1)$$

En general, de cara a la etapa $i-1$, la etapa i se comporta como un sistema crossbar a x b con una probabilidad de acceso por ciclo:

$$m_i = B_{ab}^{m_{i-1}} \quad (5.2)$$

Lo que nos proporciona una expresión iterativa para el cálculo de m_i con $m_1 = m$.

El ancho de banda de la red delta $N \times M$ ($N = a^n$, $M = b^n$) vendrá dado por:

$$B_{ab}^m \times b^{n-1} \quad (5.3)$$

$$\text{donde } B_{ab}^i = b (1 - (1 - m_i/b)^a) \quad (5.4)$$

es la fórmula aproximada de Patel para un sistema crossbar a x b .

Kruskal y Snir [KRUS83] presentan un valor asintótico para la expresión 5.2 en un sistema $a^n \times a^n$ para valores muy elevados de n y que demuestran ser igual a.

$$m_n = \frac{2a}{(a-1)n} \left[1 - \frac{(a+1)}{3(a-1)} \frac{\log_n n}{n} + O\left(\frac{1}{n}\right) \right] \quad (5.5)$$

Si bien podría parecer que la aplicación iterativa de esta aproximación a una red de n etapas iría acumulando un error muy elevado incluso para valores reducidos de n , resultados de simulación obtenidos por Patel [PATE81], Kruskal y Snir [KRUS83] y Bhuyan [BHUY85] demuestran que el error cometido es pequeño incluso para valores muy grandes de n , siendo de una gran exactitud la expresión asintótica (5.5) obtenida a partir de la expresión recurrente (5.4).

m	$2^2 \times 2^2$		$2^3 \times 2^3$		$2^4 \times 2^4$	
	SIM.	MOD.	SIM.	MOD.	SIM.	MOD.
0.25	0.963	0.882	1.878	1.668	3.381	3.162
0.50	1.708	1.559	3.101	2.814	5.646	5.132
0.75	2.137	2.066	3.656	3.599	6.400	6.388
1.00	2.362	2.438	3.912	4.132	6.720	7.197

Tabla 5.1. Comparación entre el ancho de Banda de redes delta $2^n \times 2^n$ obtenido por Simulación (SIMUL.) y obtenido por el modelo de Patel (MOD.) para diferentes valores de m y n .

En la figuras 5.6 se representa el ancho de banda de un sistema $N \times N$ tipo crossbar frente a uno con redes delta del tipo $2^n \times 2^n$ y $4^h \times 4^h$ ($n = \log_2 N$ y $h = \log_4 N$) para diferentes valores

de m . Los resultados han sido obtenidos usando el modelo de Patel (expresión 5.4) para el cálculo de B_{ab}^m y el modelo iterativo anterior para el cálculo del ancho de banda de las redes delta.

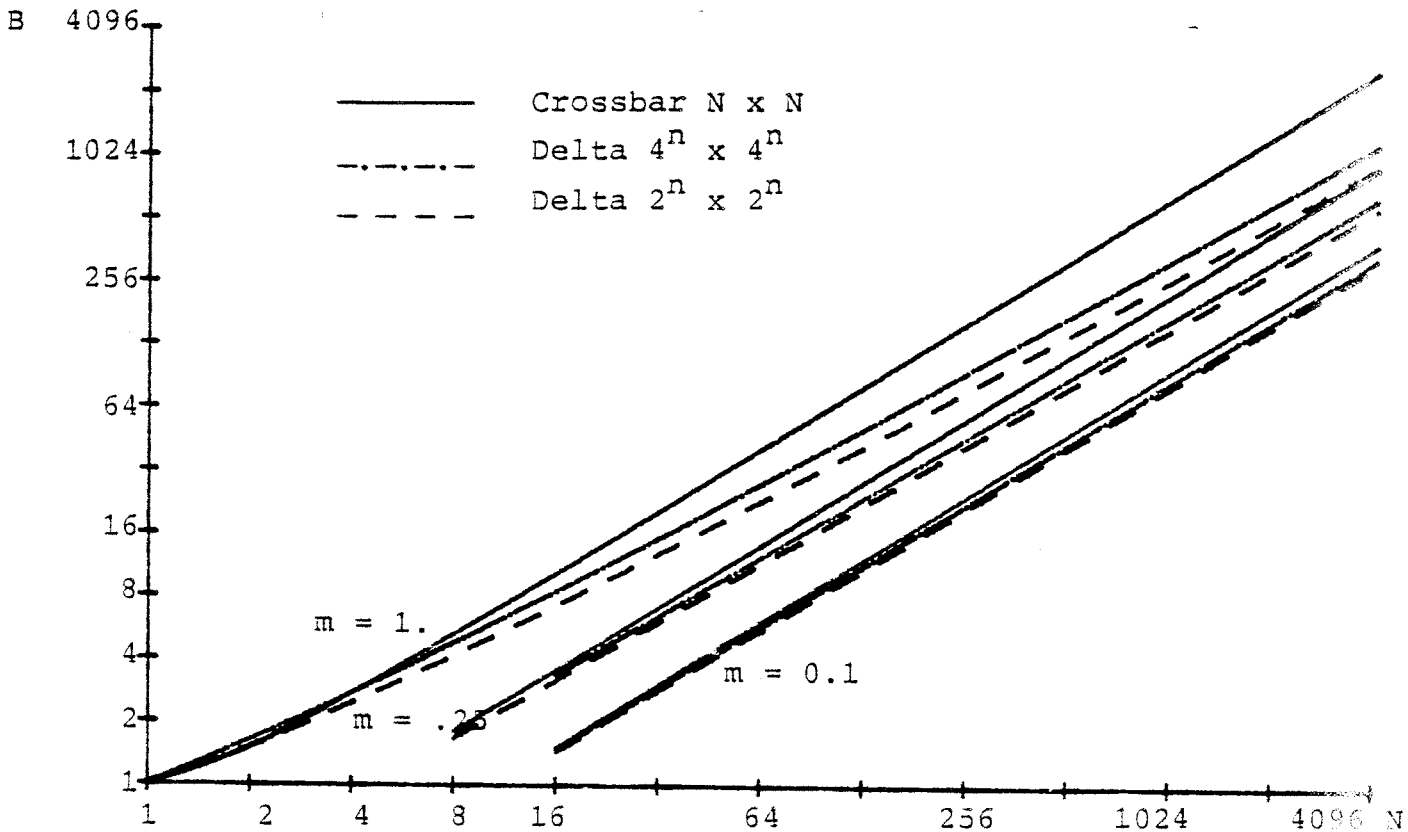


Fig. 5.6.

En esta figura, puede apreciarse que la reducción en el ancho de banda, respecto al de un sistema cross-bar, puede ser muy notable (en torno a un 50%) para valores muy elevados de N . El efecto es menos notable para valores bajos de la probabilidad de acceso por ciclo de los procesadores m .

Thanawastien y Nelson [THAN81] desarrollaron un modelo analítico basado en cadenas discretas de Markov para redes delta 2^n

$x \in 2^n$ con $m = 1$, bajo la misma suposición de que las peticiones no atendidas en un ciclo son simplemente ignoradas.

Cada uno de los ciclos es dividido en n subciclos en cada uno de los cuales se atraviesa una etapa. Sea X_j el número de peticiones que sobreviven la etapa j -ésima y llamemos $T_{j,j+1}$ a una matriz cuyos elementos t_{ik} se definen en la forma:

$$t_{ik} = P [x_{j+1} = k / X_j = i] \quad (5.6)$$

Es fácil ver que la matriz $T_{j,j+1}$ es estacionaria, es decir:

$$T_{j,j+1}(t_1) = T_{j,j+1}(t_2) \quad (5.7)$$

$$t_1, t_2 = t_j, t_j + t_c, t_j + 2 t_c, \dots$$

y además no depende de la etapa considerada:

$$T_{j,j+1} = T_{k,k+1} ; j, k = 1, 2, \dots, n-1 \quad (5.8)$$

Bajo la hipótesis de que las direcciones de destino son independientes y uniformemente distribuidas, los autores evalúan la matriz T [THAN81] en función de la probabilidad p de que no se haya conflictos en un selector 2×2 .

Por aplicación de la ecuación de Chapman-Kolmogorov [KLEI75], la matriz de probabilidades de transición de la etapa 1 a la etapa n -ésima será:

$$T^n, \text{ donde } T^n = T^{n-1} \times T \quad (5.9)$$

cuyos elementos t_{ij} representan la probabilidad de que j peticiones alcancen los módulos de memoria dado que todas las peticiones no atendidas son eliminadas.

El ancho de banda del sistema será:

$$B = \sum_{k=0}^N k t_{n_0 n_k}^n \quad (5.10)$$

donde n_0 es el número de nuevas peticiones de acceso por ciclo y que resulta ser $N = 2^n$ para el caso $m = 1$.

Para valores de m distintos de 1, los autores adoptan un modelo muy parecido al aproximado de Bhandarkar (apartado 3.3.5) para el sistema crossbar. El tiempo de procesamiento se supone múltiplo del tiempo de ciclo y geoméricamente distribuido.

El estado del sistema se caracteriza por el número de peticiones presentes en el sistema y la probabilidad de transición entre el estado i y el estado j viene dada por:

$$P_{ij} = \sum_{c=0}^j \binom{N-c}{j-c} m^{j-c} (1-m)^{N-j} T_{i,j-c}^{(n)} \quad (5.11)$$

donde c representa el número de procesadores cuyas peticiones no han sido servidas en el presente ciclo y $t_{i,j-c}^{(n)}$, represen-

ta el elemento i, j -ésimo de la matriz T^n definida en (5.9).

Este modelo es bastante más complejo que el modelo propuesto por Patel, su aplicación es más restringida y, sin embargo, no representa una apreciable mejora frente a los resultados de este último [THAN81].

5.3 REDES DELTA TOLERANTES A FALTA.

Las Redes Delta tal como han sido definidas son redes no tolerantes a faltas dado que hay un único camino posible entre cada procesador y cada memoria.

Diferentes formas de convertir las redes Delta en redes tolerantes a falta han sido propuestas en la literatura.

Aparte las soluciones estrictamente software basadas en una codificación de la información o soluciones sofisticadas como el triplicado y "voting", la solución más generalmente aceptada consiste en aumentar el número de etapas de la red, aumentando de esta forma el número de caminos posibles entre cada procesador y cada memoria ([FENG81], [AGRA82]).

Padmanabhan y Lawrie [PADM83] presentan una red delta modificada con múltiples caminos entre cada procesador y cada memoria con la particularidad de que estos caminos múltiples son disjuntos entre sí (no comparten más de un selector).

En estas redes tolerantes a faltas y, sobre todo, cuando el nivel de redundancia no es muy elevado, normalmente se asume que la red trabaja como si fuera una red delta no utilizándose los caminos alternativos en ausencia de falta.

En otras aproximaciones tales como las redes d-Dilatadas y redes d-Duplicadas se puede admitir, sin embargo, otro tipo de comportamiento por lo que serán tratadas aparte.

Una red d-dilatada es una red en que cada línea de salida y cada línea de entrada de un selector es sustituida por d líneas de entrada y d líneas de salida idénticas. En una red de este tipo un mismo selector puede recibir d peticiones por una misma entrada (una por línea) y emitir d peticiones por una misma salida (una por línea). Si $j \leq d$ peticiones pretenden alcanzar una misma salida, todas alcanzarán dicha salida a través de una de sus d líneas. Si $d > j$ peticiones pretenden alcanzar una misma salida, el selector escoge arbitrariamente d de ellas y las restantes $j - d$ serán rechazadas.

En la figura (5.7a) se representa una red $2^2 \times 2^2$ 2 - dilatada.

Una red d-duplicada está constituida simplemente por d redes en paralelo (fig. 5.7b).

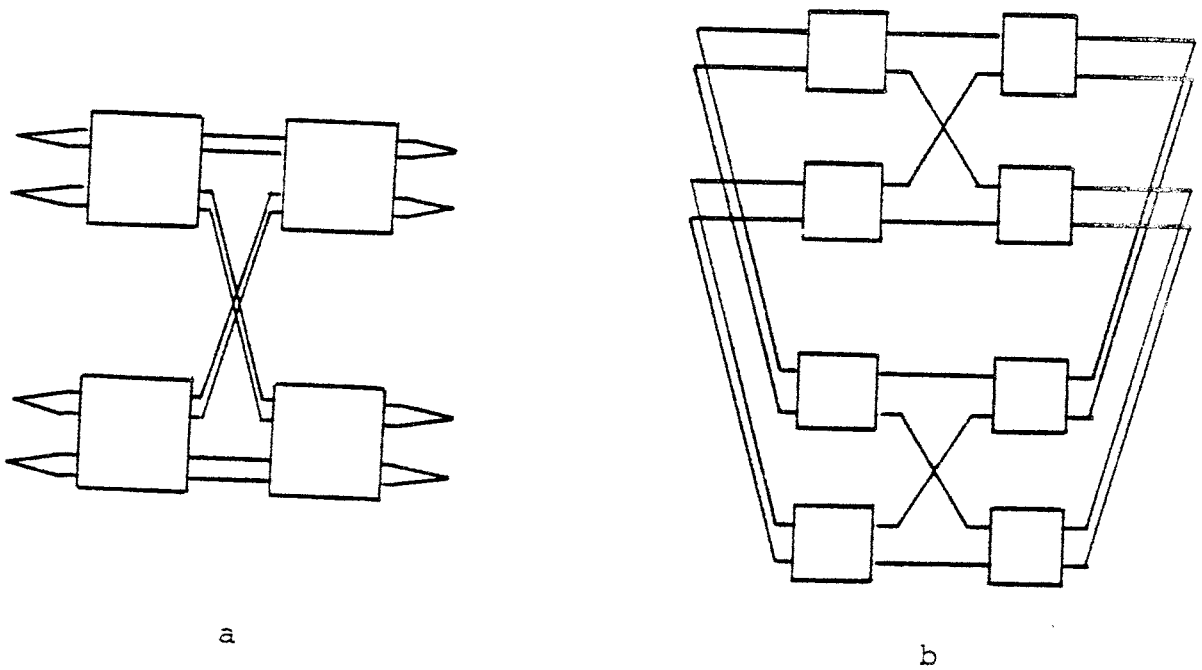


Fig. 5.7. Red $2^2 \times 2^2$ 2 - dilatada (a) y 2-duplicada (b).

Bajo la suposición ya repetida de que las peticiones no atendidas en un ciclo son simplemente ignoradas en el ciclo siguiente, Kruskal y Snir [KRUS83] establecen un modelo para las redes d -dilatadas y d -duplicadas.

Para una red d -dilatada $2^n \times 2^n$, llamemos $R(k, j)$ a la probabilidad de que j peticiones salgan por las d líneas idénticas de una salida de un selector de la etapa k . $R(k, j)$ se inicializa en la forma:

$$R(0, j) = \begin{cases} 1 - m & j = 0 \\ m & j = 1 \\ 0 & j \neq 0, 1 \end{cases} \quad (5.12)$$

La probabilidad de que i peticiones entren en un selector de la etapa $k + 1$ ésima será $\sum_{r+s=i} R(k,r) \times R(k,s)$. La probabilidad de que j de estas i peticiones se dirijan a una misma salida del selector vendrá dada por $\binom{i}{j} 2^{-i}$, de donde:

$$R(k+1, j) = \sum_{i=j}^{2d} \left(\sum_{r+s=i} R(k,r)R(k,s) \right) \binom{i}{j} 2^{-i} \quad (5.13)$$

$j < d$

$$R(k+1, d) = \sum_{i=d}^{2d} \left(\sum_{r+s=i} R(k,r)R(k,s) \right) \sum_{t=d}^i \binom{i}{t} 2^{-i} \quad (5.14)$$

$j = d$

La probabilidad de que, al menos, una de las d líneas de una salida del n ésimo selector contenga al menos una petición será:

$$1 - R(n,0) \quad (5.15)$$

y el ancho del banda del sistema vendrá dado por:

$$B = 2^n (1 - R(n,0)) \quad (5.16)$$

De forma análoga este razonamiento puede ser aplicado a redes $k^n \times k^n$, si bien las expresiones se hacen más complejas.

Para una red d -duplicada $k^n \times k^n$ asumiremos que cada procesador elige al azar una de las d posibles subredes delta para transmitir una petición. De esta forma, cada una de las d subre-

des puede ser tratada como una red delta independiente en la que los procesadores tienen una probabilidad m/d de emitir un mensaje en cada ciclo.

Aplicando a cada una de las subredes la expresión iterativa de Patel (5.2), (5.4) podemos determinar m_n que tendrá el mismo valor para todas ellas.

La probabilidad de que, al menos, una de las d subredes tenga una petición a memoria en la salida será:

$$1 - (1 - m_n/k)^k \quad (5.17)$$

y el ancho de banda total, teniendo en cuenta que en la última etapa se produce un conflicto adicional debido a las memorias, será:

$$B = k^n (1 - (1 - m_n/k)^{kd}) \quad (5.18)$$

donde m_i tiene la expresión iterativa (5.2) con $m_1 = m/d$.

5.4 REDES DELTA CON BUFFERS ENTRE ETAPAS.

Hasta ahora hemos supuesto que las peticiones no atendidas son ignoradas, bien porque los son realmente o porque se indica a los procesadores que deberán volver a realizarlas en el ciclo siguiente (en cuyo caso, más real, los modelos vistos anteriormente sólo son aproximaciones).

Una forma de aumentar el ancho de banda de las redes multietapa consiste en mantener las peticiones no atendidas bloqueadas en unos buffers dispuestos en la misma etapa en que se originó el conflicto. Estas son las que denominamos redes con buffers.

Estas peticiones vuelven a ser lanzadas en el ciclo siguiente, pero desde la etapa en que quedaron bloqueados, reduciéndose el nivel de conflicto en las primeras etapas y aumentando el ancho de banda.

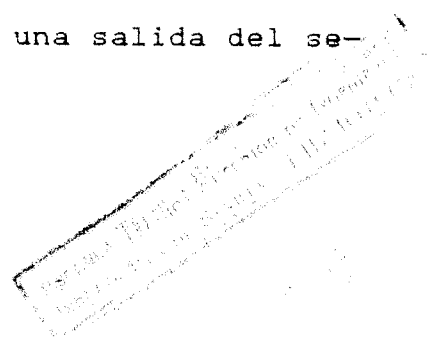
Varios autores [DIAS81a], [KRUS83] han tratado de estudiar este tipo de redes.

Días y Jump [DIAS81a] realizaron un estudio aproximado para redes delta $2^n \times 2^n$ asumiendo que se encuentran en su estado de "carga máxima", es decir, los buffers están permanentemente llenos.

Cuando una petición es aceptada por el selector de la etapa $i+1$ ésima y removido de su buffer de entrada, una nueva petición procedente de un selector de la etapa anterior llena el hueco en el buffer de inmediato.

El retraso introducido por cada buffer se modela como si consistiera de dos subintervalos:

- El tiempo necesario para seleccionar una salida del selector (t_{selec}) más



- El tiempo necesario para alcanzar la salida seleccionada (t_{pass}) y entrar en el buffer de la etapa siguiente.

En cualquier caso, $t_{selec} + t_{pass} = t_{ciclo} = \text{constante}$.

Los autores establecen un modelo iterativo de bastante complejidad para analizar los casos límites ($t_{selec} = 0$ y $t_{pass} = 0$), resolviendo los casos más simples y simulando los de mayor complejidad [DIAS81b].

Kruskal y Snir [KRUS83] estudian las redes $k^n \times k^n$ en el caso $t_{pass} = 0$ asumiendo que los buffers son de longitud infinita, proponiendo una fórmula para evaluar el tiempo medio que tarda una petición en atravesar la red:

$$T = \log_k N \left(t_r + t_c \frac{\left(1 - \frac{1}{k}\right)^m}{2(1-m)} \right) \quad (5.19)$$

donde $N = k^n$, t_r es el tiempo que tarda una petición en atravesar un selector supuesto que no hubiera conflicto, t_c es el tiempo de ciclo de un selector, es decir, el tiempo que transcurre entre dos llegadas sucesivas a una misma entrada ($t_r < t_c$) y m es la probabilidad de acceso por ciclo de los procesadores. Los autores muestran que para valores bajos de m ($m < 0.6$) esta fórmula es una buena aproximación, incluso en el caso en que los buffers tuvieran una longitud finita pero no pequeña.

En ambos modelos de redes con buffers sus autores muestran que éstas tienen unas prestaciones cercanas o incluso superiores a las de un sistema crossbar, si bien con modelos más reales (sistemas con una carga reducida o buffers de tamaño finito y reducido) los aumentos en el ancho de banda respecto de las redes sin buffers (obtenidos por simulación) no son tan notables [DIAS81b].

Sin embargo, este incremento en el ancho de banda del sistema no justifica el incremento de complejidad y coste que se deriva de la disposición de los buffers entre etapas para direcciones y datos y la lógica de arbitraje y control de estos buffers [PATE81].

CAPITULO 6

ESTUDIO COMPARATIVO DE LOS SISTEMAS DE INTERCONEXION PROCESADOR MEMORIA.

6.1 INTRODUCCION.

Diferentes sistemas de interconexión procesador-memoria han sido estudiados en los capítulos anteriores.

No es tarea fácil establecer una comparación entre estas redes. Una simple comparación entre los anchos de banda de sus memorias o la utilización de sus procesadores, si bien necesaria, no es en absoluto suficiente.

Muchos otros parámetros han de ser tenidos en consideración. Entre ellos:

- a) Fiabilidad de la red y de sus elementos.
- b) Tolerancia a Fallos y Mantenibilidad.
- c) Modularidad y Reconfigurabilidad.
- d) Retrasos en el paso de la información.
- e) Complejidad en los algoritmos de control.
- f) Integrabilidad VLSI.
- e) Costo de la red relativo al sistema global, etc.

Establecer una evaluación cuantitativa en base a un parámetro global que incluya todas estas cualidades de los sistemas no es realmente posible. En cada aplicación concreta una o varias de estas características de las redes será predominante y no podemos establecer un parámetro único que permita englobarlas.

Por ello este estudio se centrará en un reducido número de parámetros que, en cierta medida, representan algunas de las cualidades antes citadas. Estos parámetros son:

1. Prestaciones del sistema en forma de sus anchos de banda.
2. Orden de Complejidad de la Red. Entendiendo por orden de complejidad el número relativo de puertas necesarias para construir la red.

En gran medida, el orden de complejidad es un índice de la fiabilidad, mantenibilidad y coste de cada una de estas redes.

3. Fiabilidad y tolerancia a faltas.

La comparación se establecerá para el caso discreto con las hipótesis simplificadoras reflejadas en el apartado 3.2.1, aunque de igual forma podría haberse planteado el análisis en el caso continuo.

Utilizaremos como base de la comparación seis sistemas tipo:

- a) Sistemas Crossbar $N \times N$ (sistemas tipo 1)
- b) Sistemas Multibuses $N \times N$ con K buses, para $K = N/2$ (sistemas tipo 2) y con $K=N/4$ (sistemas tipo 3)
- c) Sistemas de N procesadores y dos grupos de $N/2$ memorias cada uno, unidos mediante dos grupos de $K = N/4$ buses parciales (sistemas tipo 4) y
- d) Sistemas Delta $2^n \times 2^n$ (tipo 5) y $4^n \times 4^n$ (tipo 6)

En el cálculo del ancho de banda utilizaremos preferentemente resultados procedentes de simulación y sólo en los casos más complejos utilizaremos resultados obtenidos con modelos aproximados.

Estos modelos aproximados serán el de Yen, Patel y Davidson para el sistema crossbar, el modelo de Patel para las redes delta y el modelo original denominado de Bhandarkar modificado para los sistemas multibuses y con varios grupos de buses parciales. En los capítulos anteriores se ha mostrado la exactitud de estos modelos y este análisis presentará, en consecuencia, la misma exactitud.

No deben esperarse resultados que conduzcan a conclusiones maximalistas. No obstante, creemos que este estudio puede servir de gran ayuda para estimar las prestaciones de los sistemas actuales y para orientar futuros diseños.

6.2 ANCHOS DE BANDA.

En las figuras 6.1 y 6.2 se observa cómo afecta el número de procesadores al ancho de banda de las memorias en cada uno de estos sistemas para valores de $m = 0.5$ y $m = 1$.

En la figura 6.3 se representa el ancho de banda de un sistema 16×16 para cada uno de los sistemas considerados al variar la probabilidad de acceso por ciclo m .

Las mismas figuras representarían la utilización de los procesadores mediante un simple cambio de escalas.

De la observación de estas figuras pueden deducirse interesantes conclusiones.

Para valores de m muy bajos ($m < 0.25$) el ancho de banda de un sistema crossbar es sensiblemente igual al de un sistema con un número reducido de buses ($K = N/4$), incluso para valores razonablemente elevados de m los sistemas con $K = N/2$ buses sufren una degradación pequeña respecto a un sistema crossbar.

Un sistema con dos grupos de $K = N/4$ buses parciales cada uno tiene un comportamiento bastante similar al de un sistema con $K = N/2$ buses globales siendo tan sólo notable la diferencia para valores de m elevados.

Las redes delta tienen un comportamiento bastante inferior al de un sistema cross-bar, siendo las redes con mayor tamaño de

los selectores las que dan origen a los mayores anchos de banda.

Al aumentar N ninguna de las redes consideradas tienen un comportamiento lineal en su ancho de banda, de hecho en las figuras puede observarse que este comportamiento es, sin embargo, bastante lineal en escala logarítmica.

Las redes más fuertemente afectadas por el aumento de N son las redes delta siendo menor el efecto a medida que aumenta el tamaño de los selectores.

Los sistemas multibuses presentan un buen comportamiento para valores bajos de m , pudiendo observarse la saturación de los buses cuando el sistema está sobrecargado (m elevados). Cuanto menor es el número de buses, tanto más rápidamente se satura el sistema.

En cualquier caso, para valores suficientemente elevados de N , el ancho de banda de los sistemas multibuses llega a superar al de las redes delta más simples (selectores 2×2 y 4×4). Este efecto es más notable para valores bajos de m .

Los resultados obtenidos son, por otra parte, fácilmente explicables. De todas las redes consideradas tan sólo los sistemas crossbar tienen la característica de ser "no bloqueantes", es decir, el conflicto se origina tan sólo en el acceso a las memorias, pero la red por sí misma no presenta ningún conflicto adicional.

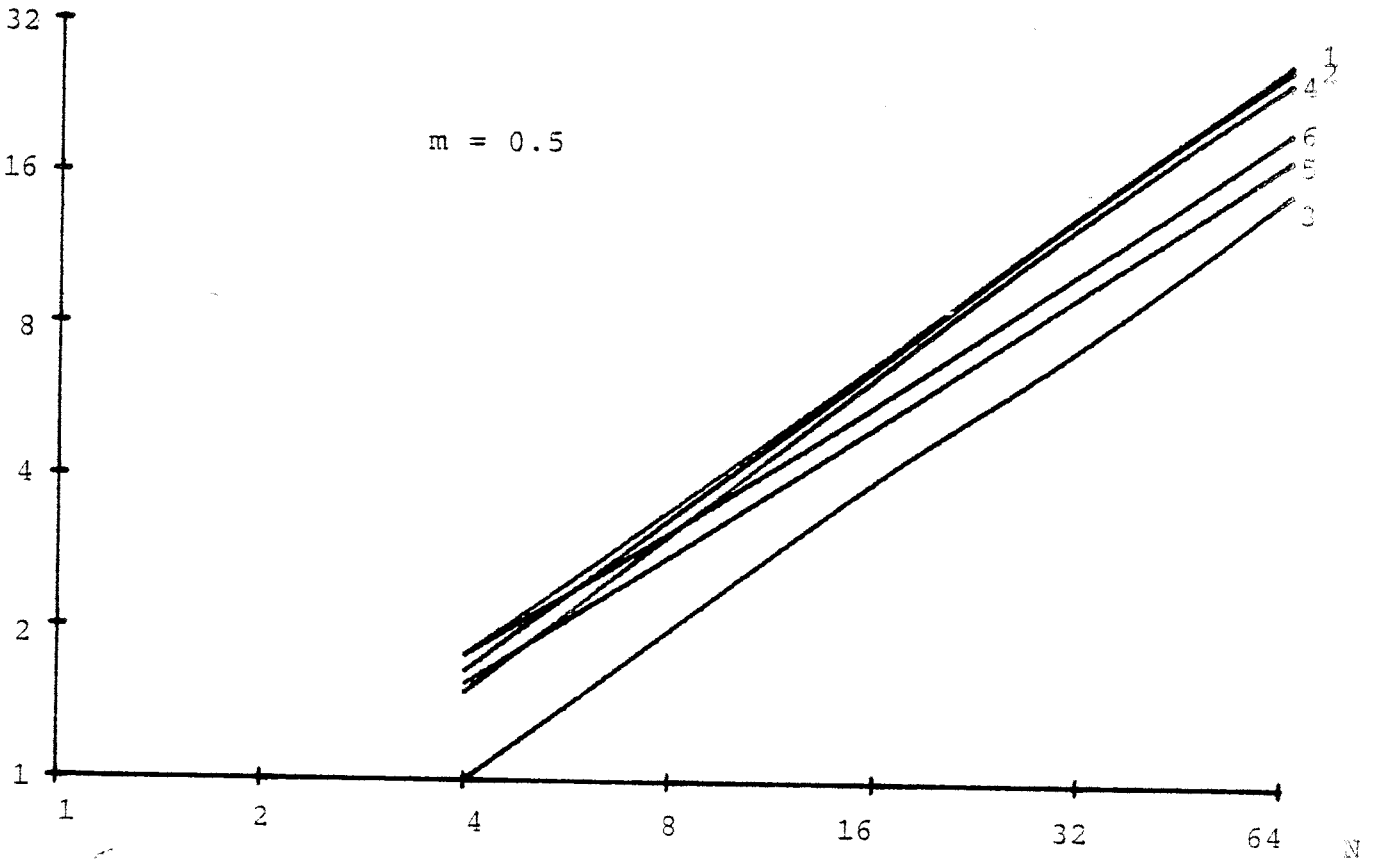


Fig. 6.1

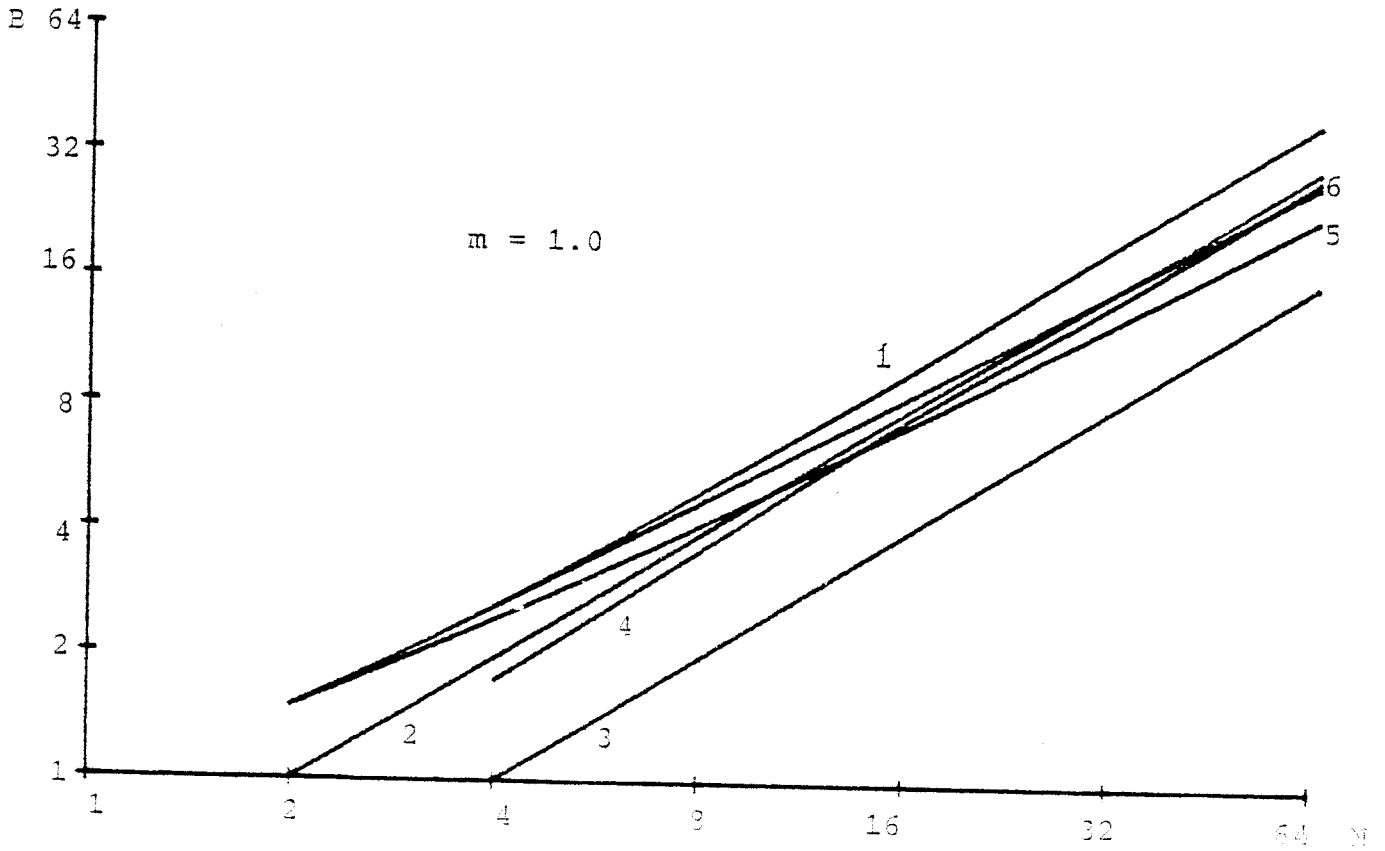


Fig. 6.2

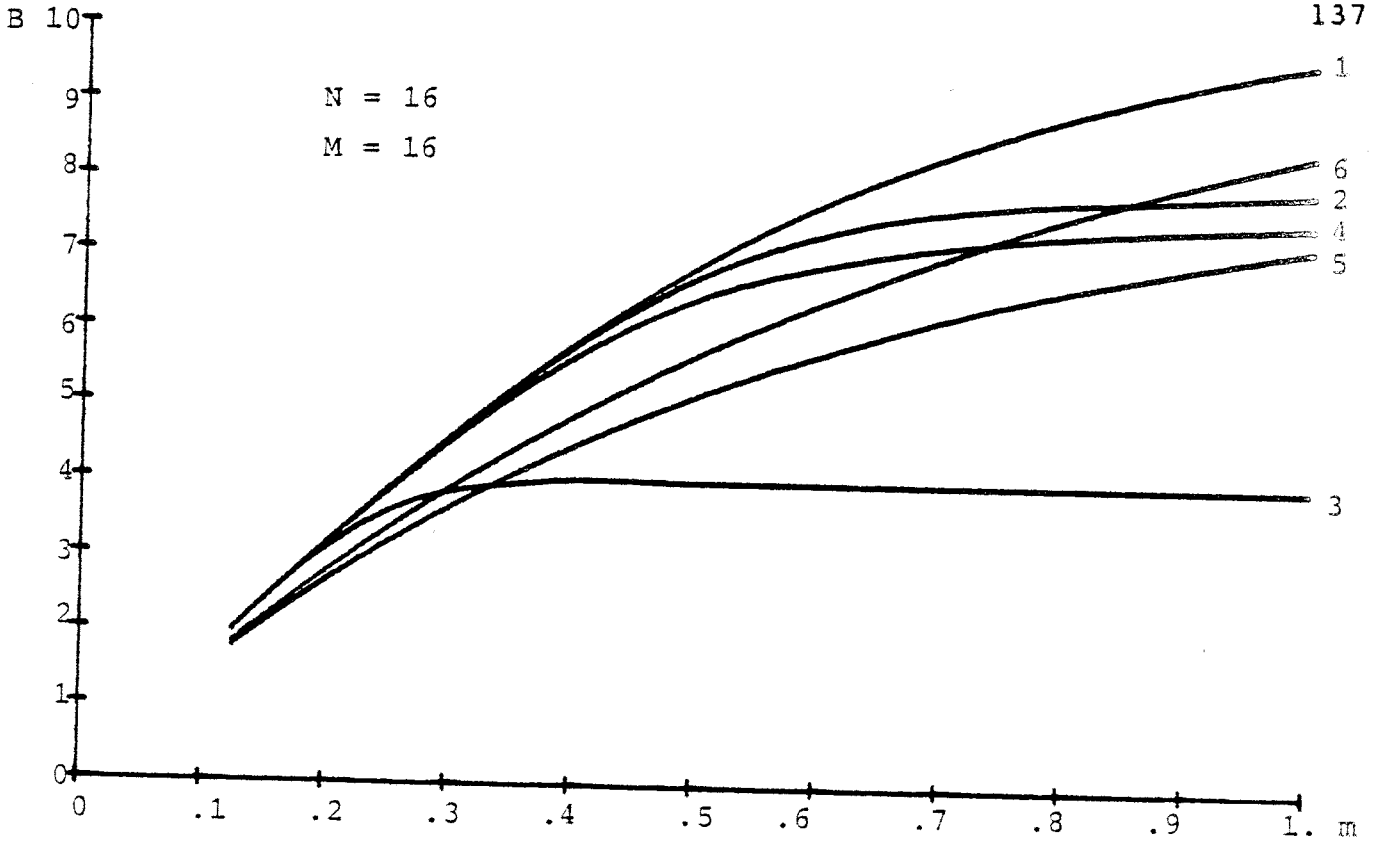


Fig. 6.3

Complej.

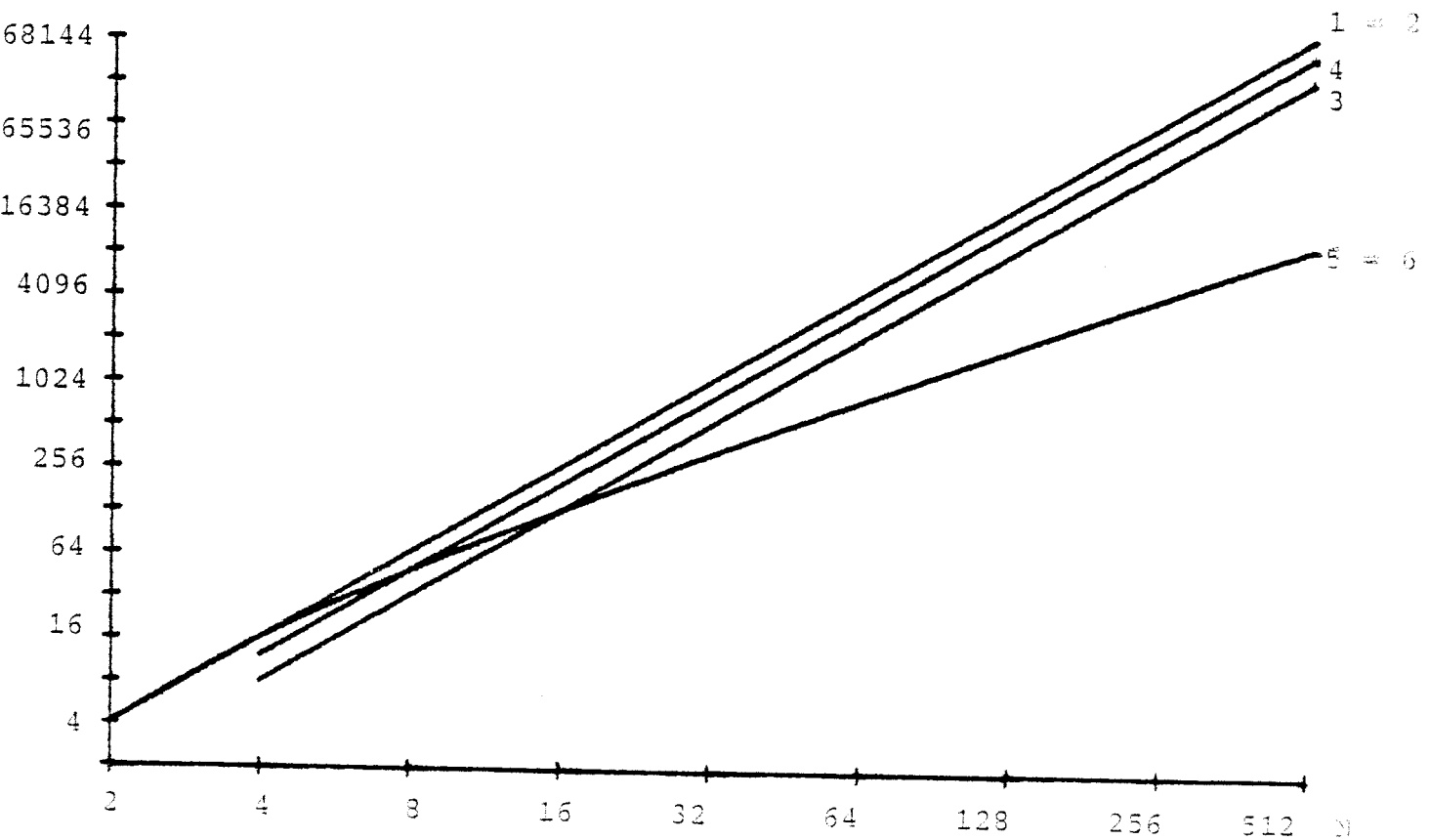


Fig. 6.4

Los sistemas multibuses son, por el contrario, "bloqueantes" cuando el número de buses es menor que el mínimo de procesadores y memorias. Pero el elevado aprovechamiento que se hace de los buses como recursos pasivos utilizables por cualquier par procesador-memoria que se encuentre con capacidad de comunicarse hace que el conflicto que ellos introducen sólo sea apreciable cuando la red está fuertemente sobrecargada (valores elevados de m) o el número de buses es muy reducido.

Por último las redes delta son redes en las que puede originarse un conflicto en el acceso a una de las salidas de los selectores en cada una de las etapas. Cuanto mayor sea el tamaño del selector, menor será el número de etapas, aproximándonos al sistema crossbar y reduciéndose el nivel de conflicto originado por la red.

6.3 ORDEN DE COMPLEJIDAD DE LAS REDES.

El orden de complejidad de una red de interconexión procesador-memoria puede definirse como el número de puertas elementales necesarias para la realización física de la red.

Tan sólo tendremos en cuenta aquí el orden de complejidad debido a la transmisión de datos sin tener en cuenta las puertas necesarias para el control y arbitraje de esta información cuyo número, relativo al anterior, suele ser despreciable.

En el caso de un diseño basado en componentes de baja escala de integración puede asumirse que la complejidad de un selector de tamaño $a \times a$ es a^2 / b^2 veces más complejo que el de un selector de tamaño $b \times b$.

Bajo esta suposición y tomando N^2 como el orden de complejidad de un selector $N \times N$, tenemos:

a) Sistema Crossbar.

El sistema crossbar tiene un único selector de tamaño $N \times N$ y su orden de complejidad será N^2 .

b) Sistemas Multibuses.

Un sistema multibus conecta N procesadores con K buses y K buses con N memorias. Asumiendo que estas conexiones se realizan mediante selectores $N \times K$, el orden de complejidad de un sistema multibus sería $2 N K$.

Para el caso de $K = N/2$ tendríamos N^2 .

Para el caso de $K = N/4$ tendríamos $N^2/2$.

c) Sistema con dos Grupos de Buses Parciales.

En este caso, todos los N procesadores se interconectan con todos los $2 K$ buses parciales, y cada grupo de K buses parciales se conecta a $N/2$ memorias.

Asumiendo que estas conexiones se realizan mediante selectores $N \times 2K$ y $N/2$ por K , respectivamente, el orden de complejidad de un sistema de este tipo vendría dado por $NK + 2 K N/2 = 3 NK$.

Para el caso $K = N/4$ tendríamos $3 N^2 / 4$.

d) Redes Delta.

Una red delta $a^n \times a^n$ consta de n ($n = \log_a N$) etapas, cada una de las cuales tiene a^{n-1} ($= N/a$) selectores de tamaño $a \times a$, por lo que su orden de complejidad vendría dado por $a N \log_a N$.

Para el caso $2^n \times 2^n$ tendríamos $2 N \log_2 N$.

Para el caso $4^n \times 4^n$ tendríamos $4 N \log_4 N$.

En la figura 6.4 se muestra cómo varía el orden de complejidad de cada una de estas redes a medida que aumenta el número de procesadores y memorias N .

Es interesante comprobar que la complejidad de un sistema multibus con $K = N/2$ buses es la misma que la del sistema crossbar. En el caso $K = N/4$ la complejidad es la mitad que la de un sistema crossbar.

Por tanto, un sistema multibus tan sólo representa una notable mejora en cuanto a orden de complejidad respecto de un sistema crossbar cuando el número de buses es muy reducido frente al número de procesadores y memorias.

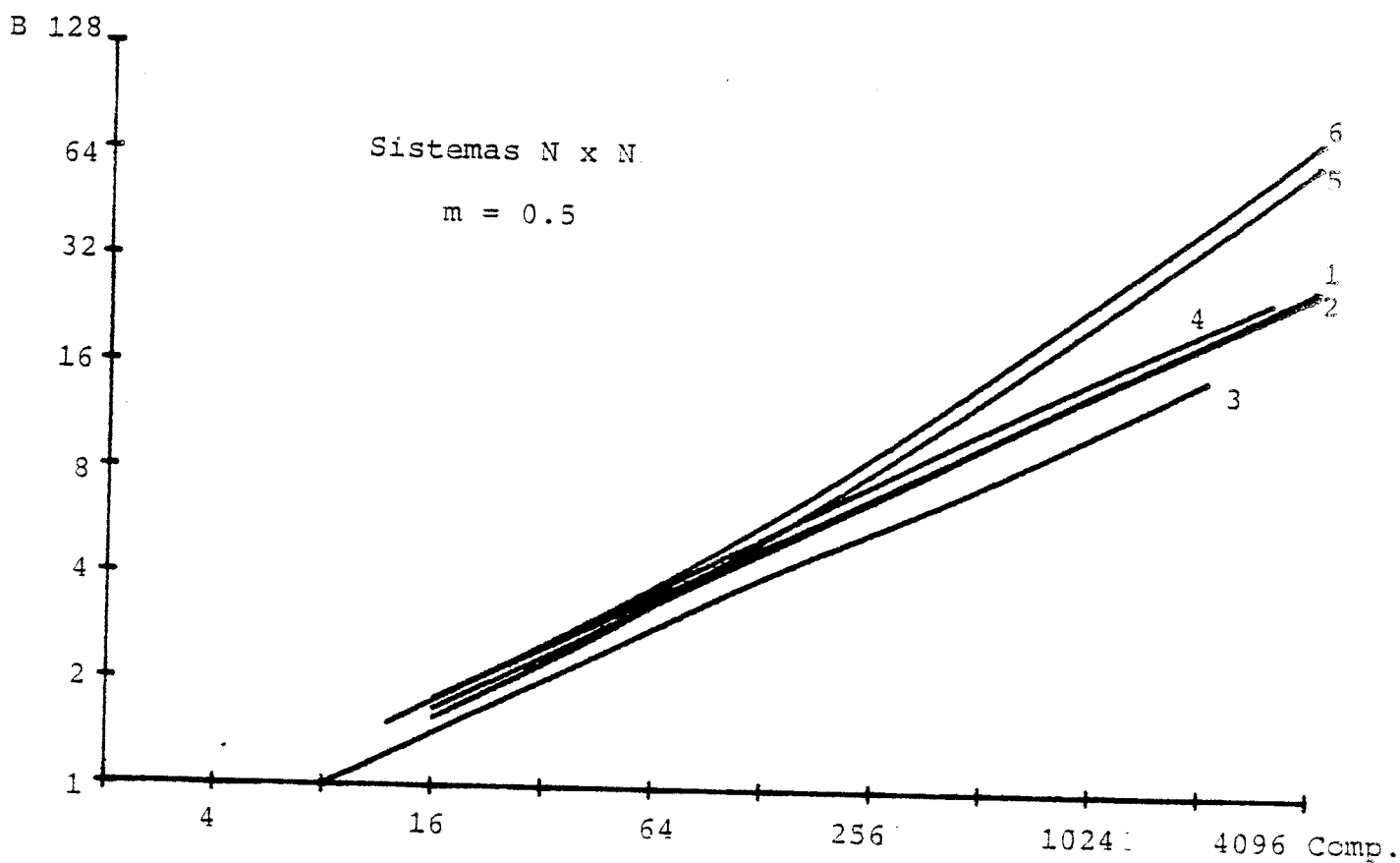


Fig. 6.5

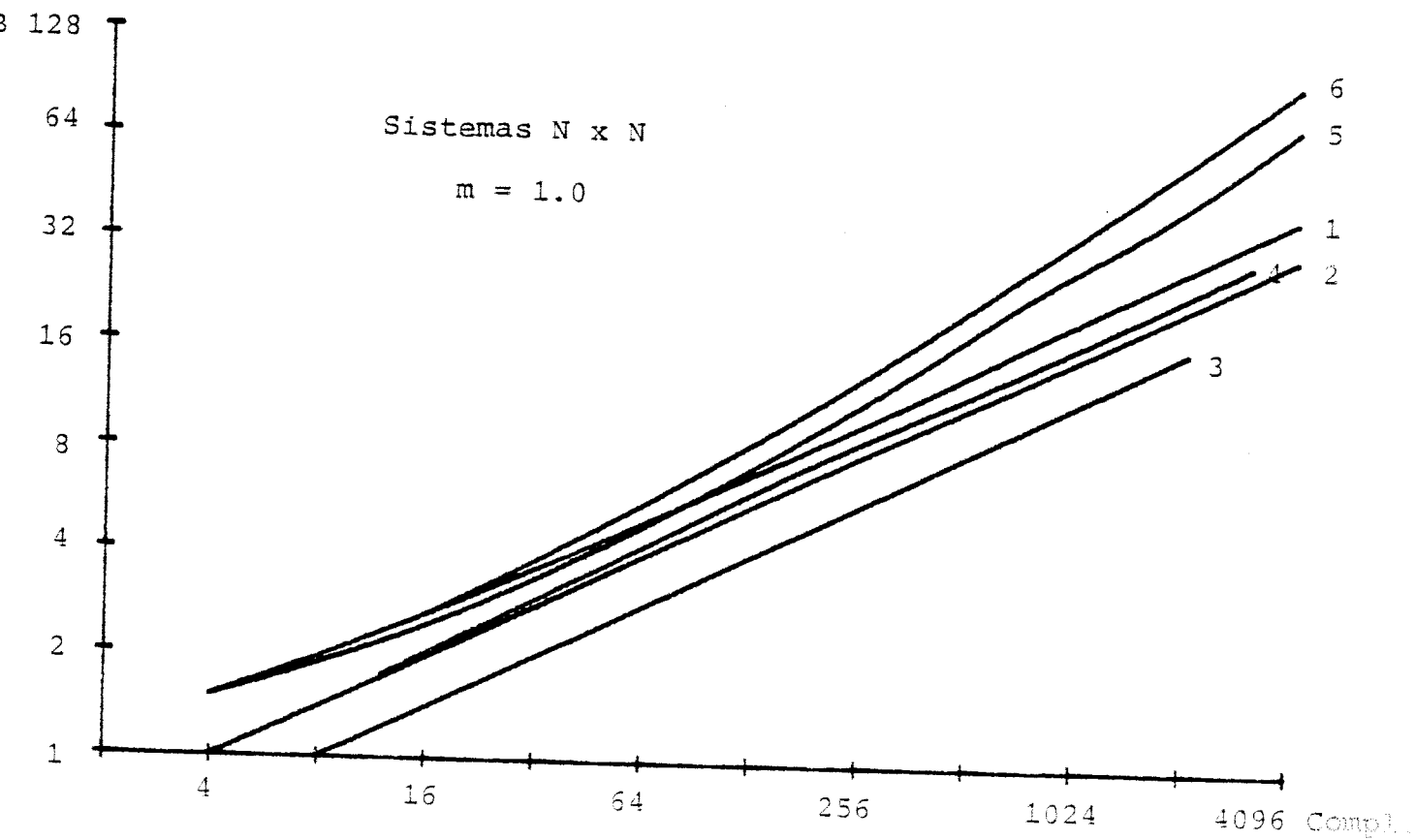


Fig. 6.6

Un sistema con dos grupos de $K = N/4$ buses parciales cada uno presenta sólo una pequeña reducción en el orden de complejidad frente a un sistema multibus con $K = N/2$ buses globales.

Por último, las redes delta son las únicas que presentan un orden de complejidad que no crece cuadráticamente con N , lo que en principio las hace interesantes para valores de N elevados.

Si bien las redes delta $2^n \times 2^n$ y $4^n \times 4^n$ presentan el mismo orden de complejidad, redes con un tamaño mayor de los selectores aproximan a los sistemas crossbar incrementándose el orden de complejidad.

En las figuras 6.5 y 6.6 se muestra el ancho de banda de cada uno de estos sistemas frente al orden de complejidad de las redes para $m = 0.5$ y $m = 1$ respectivamente.

Dos hechos notables pueden ser observados.

En primer lugar, las redes delta aparecen como las de mejor relación prestaciones/complejidad para sistemas con un número muy elevado de procesadores y memorias y que se encuentren fuertemente cargados ($m > 0.5$).

En segundo lugar, a medida que m disminuye los sistemas multibuses (y en particular los sistemas con buses parciales) se van haciendo más atractivos, de forma que para sistemas muy débilmente cargados (p.e. sistemas que sólo usan la red de interconexión para un intercambio de mensajes entre tareas), resultaría más

conveniente la utilización de sistemas con un reducido número de buses.

Por último cabe destacar que en un sistema multibus puede reducirse en gran medida el número de interconexiones sin perder efectividad.

T. Lang, M. Valero y A. Fiol en LANG83 proponen varios sistemas multibuses en que no todas las memorias están conectadas a todos los buses.

De hecho, estos autores demuestran que se puede conseguir una conectividad completa con un mínimo de $M - K + 1$ módulos de memoria conectados a cada bus.

En este caso extremo, el orden de complejidad de un sistema multibus $N \times N$ con K buses vendría dado por:

$$N \times K + (N - K + 1) \times K = K (2N - K + 1) \quad (6.1)$$

Para el caso $K = N/2$ la expresión anterior daría $3 N^2/4 + N/2$.

Para el caso $K = N/4$ sería $7 N^2/16 + N/4$.

De la misma forma podría considerarse el caso de dos grupos de buses parciales.

Si bien esta reducción de conexiones no tiene por qué afectar a las prestaciones del sistema (ancho de banda), esto se consigue mediante una notable complicación de los algoritmos de control y arbitraje, reduciéndose además el nivel de tolerancia a faltas del sistema.

6.4 ORDEN DE COMPLEJIDAD VLSI.

La suposición de que un selector $a \times a$ es a^2/b^2 veces más complejo que un selector $b \times b$ puede ser asumida para niveles bajos de integración, pero Franklin [FRAN81] demuestra que en una implementación VLSI de estas redes no se cumple esta regla.

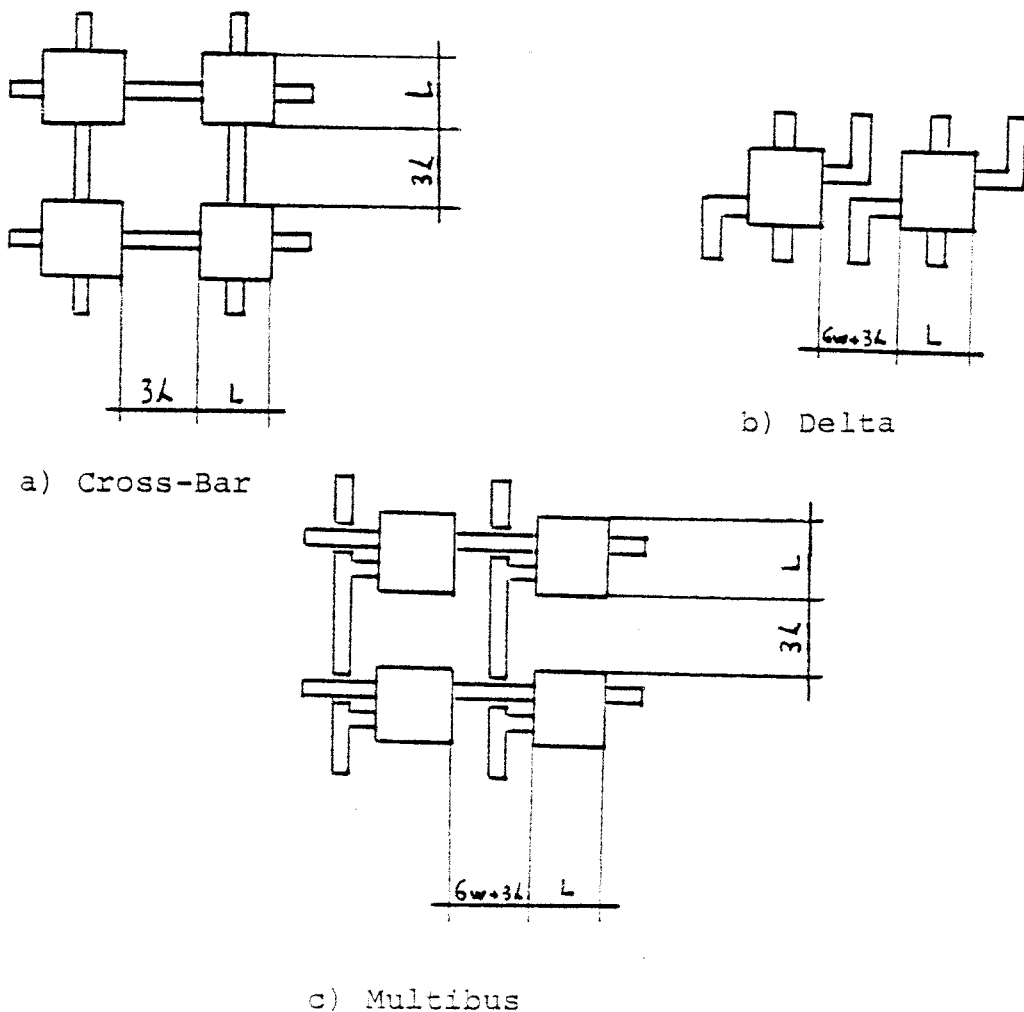


Fig. 6.7 Realización VLSI.

En una implementación VLSI el orden de complejidad de una red puede ser expresado en términos del área que ocupa.

Asumiendo que cada selector elemental ocupa un área cuadrada de lado L (figura 6.7), Franklin demuestra que una red crossbar del tipo Cmpmp ocupa un área de la forma:

$$A_{CB} = (NL + 3(N - 1))^2 \quad (6.2)$$

donde el área L del cuadrado viene dada por:

$$L = 6 \sqrt{k(\gamma + w^2)} \quad (6.3)$$

siendo w el número de líneas de datos y k y γ parámetros que dependen de la tecnología de fabricación. Se toma como unidad de medida la longitud característica de la tecnología λ asumiéndose que, tanto el ancho mínimo de una pista como la separación mínima entre pistas vienen dados por 3λ .

Para una red delta el análisis es más complejo debido al cruce de las líneas de interconexión entre selectores de dos etapas contiguas lo que origina un aumento en la longitud vertical a medida que avanzan las etapas.

Franklin demuestra que el área VLSI mínima para una red delta $2^n \times 2^n$ viene dado por:

$$A_{BA} = L_{BAH} \times L_{BAV} \quad (6.4)$$

donde L_{BAH} es el área mínima horizontal que es del orden de:

$$L_{BAH} = (N/2) L + 3 ((N/2) - 1) (2w + 1) \quad (6.5)$$

y L_{BAV} es el área mínima vertical dada por:

$$L_{BAV} = 3 + \sum_{i=2}^{\lceil \log_2(N/2) \rceil} 3n_i + \sum_{i=1}^{\lceil \log_2(N/2) \rceil} (L + 3wv (p_{i1} + p_{i2})/L) \quad (6.6)$$

siendo $n_i = 2(2^i - 1)w + 1$ el número de caminos horizontales en el nivel i ,

$$p_{i1} = \begin{cases} 3n_i & i > 1 \\ 3 & i = 1 \end{cases} \quad (6.7)$$

$$y p_{i2} = (2^{i-1} - 1)L + 2^{i-1} 3(2w + 1) + L/2 + p_{i1} + L/2 \quad (6.8)$$

son las longitudes verticales de los caminos en cada etapa de la red. v es una constante relacionada con los drivers que dirigen las líneas entre etapas y depende de la tecnología de fabricación empleada.

La relación A_{BA}/A_{CB} no sigue la ley $\log_2 N/2N$ ya que, si bien, el número de selectores elementales crece en la red delta proporcional a $N \log_2 N$, el área debida a las interconexiones crece proporcional a N^2 .

Es de notar asimismo el hecho de que el área ocupada por la red crossbar depende en pequeña medida de w (anchura mínima de la

pista), mientras que para las redes delta es determinante ya que fija el área de las interconexiones que es una parte muy importante del área total de estas redes cuando el número de sus etapas es elevado.

Para los sistemas multibuses, de la figura 6.7 se tiene:

$$A_{BK} = 2 (NL + 3(N - 1)(2w + 1))(KL + 3(K - 1)) \quad (6.9)$$

lo cual para los sistemas considerados se convierte en:

a) Sistema multibus $N \times N$ con $K = N/2$ buses globales:

$$A_{B2} = (NL + 3(N - 1))^2 + (NL + 3(N - 1))(6wN - 6w - 3) - 18 w(N - 1) \quad (6.10)$$

b) Sistema multibus $N \times N$ con $K = N/4$ buses globales:

$$A_{B4} = 1/2 (NL + 3(N - 1))^2 + 1/2 (NL + 3(N - 1))(6wN - 6w - 9) - 27 w(N - 1) \quad (6.11)$$

Para un sistema $N \times N$ con dos grupos de $K/2$ buses parciales se tiene, por último:

$$A_{BP} = (NL + 3(N - 1)(2w + 1)) (KL + 3(K - 1)) (NL + 3(N - 2)(2w + 1))(KL/2 + 3 (K/2 - 1)) \quad (6.12)$$

En las figuras 6.8, 6.9, 6.10 y 6.11 se muestra el área ocupada por cada una de estas redes relativa al área del sistema crossbar para valores típicos: $\beta = 20$, $k = 1.5$ y $v = 0.25$ para diferentes valores de w .

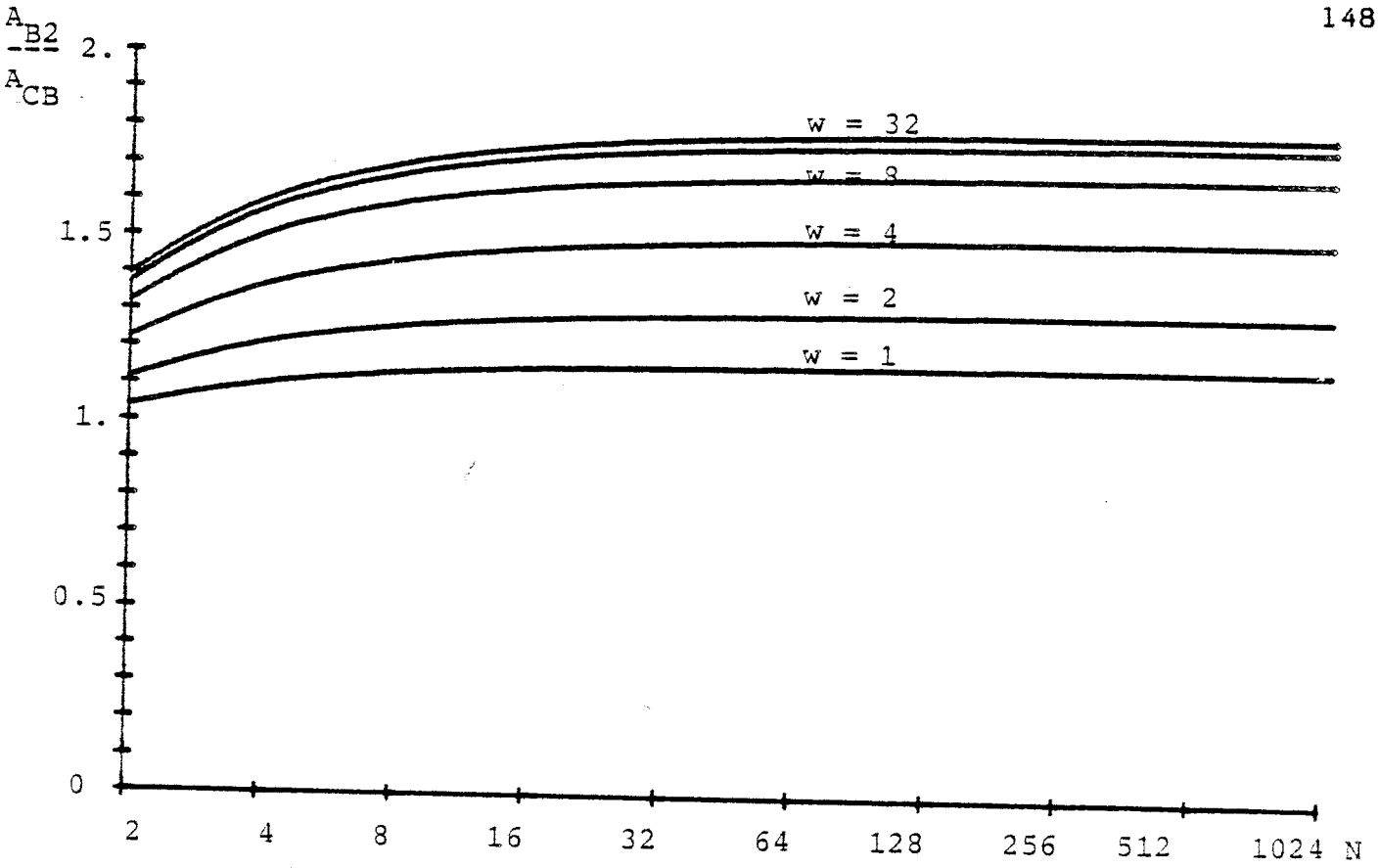


Fig. 6.8

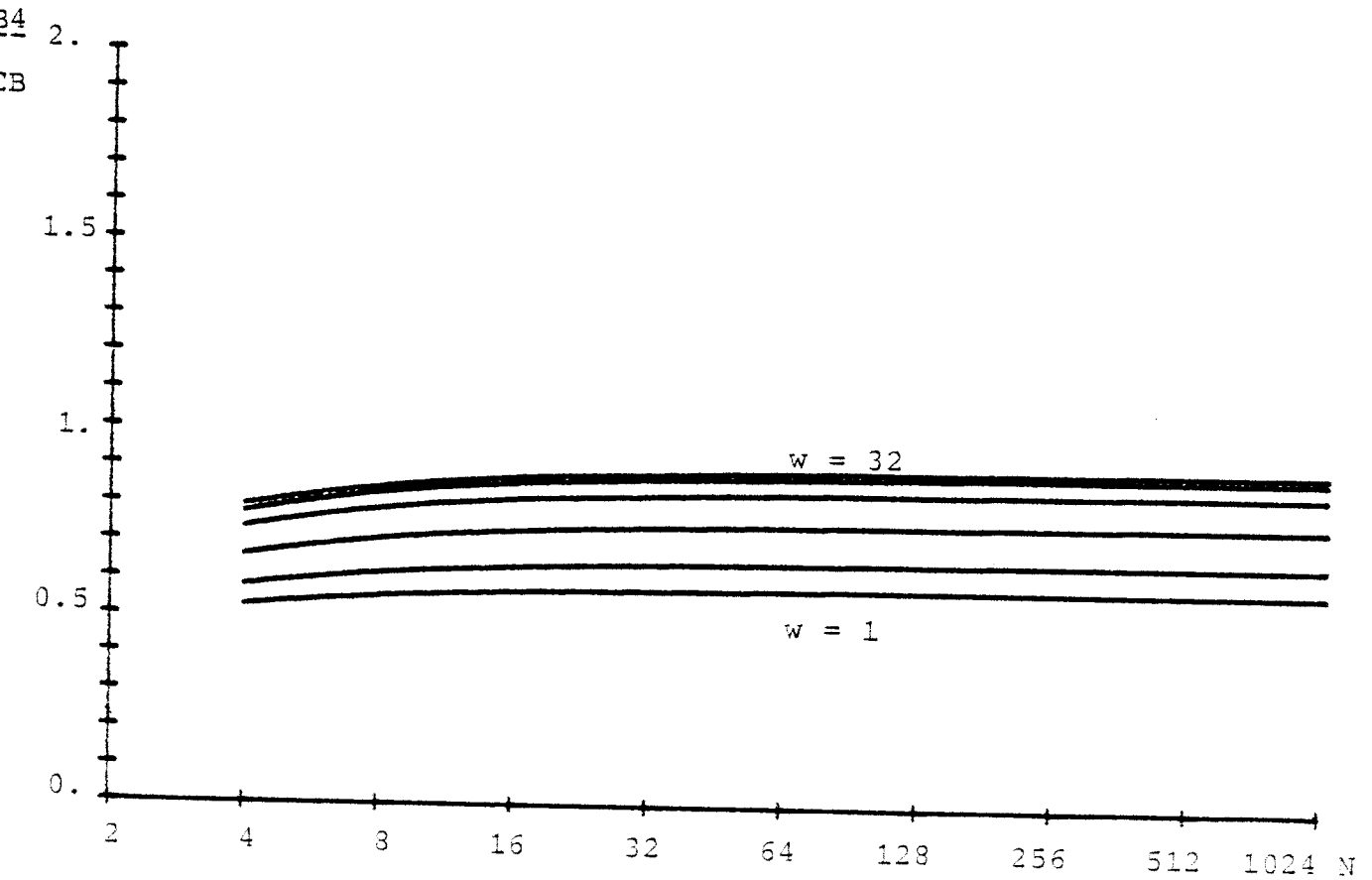


Fig. 6.9

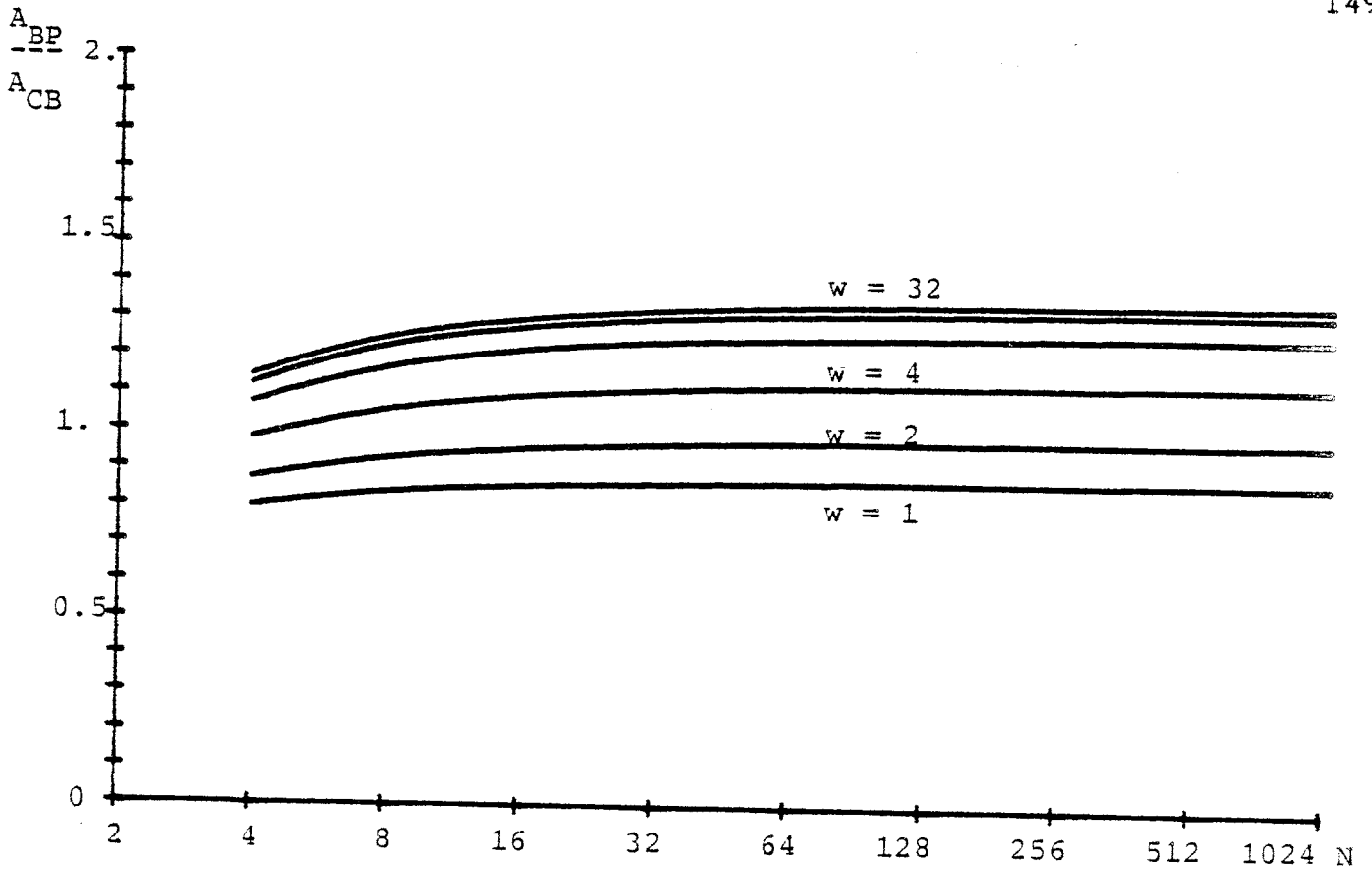


Fig. 6.10

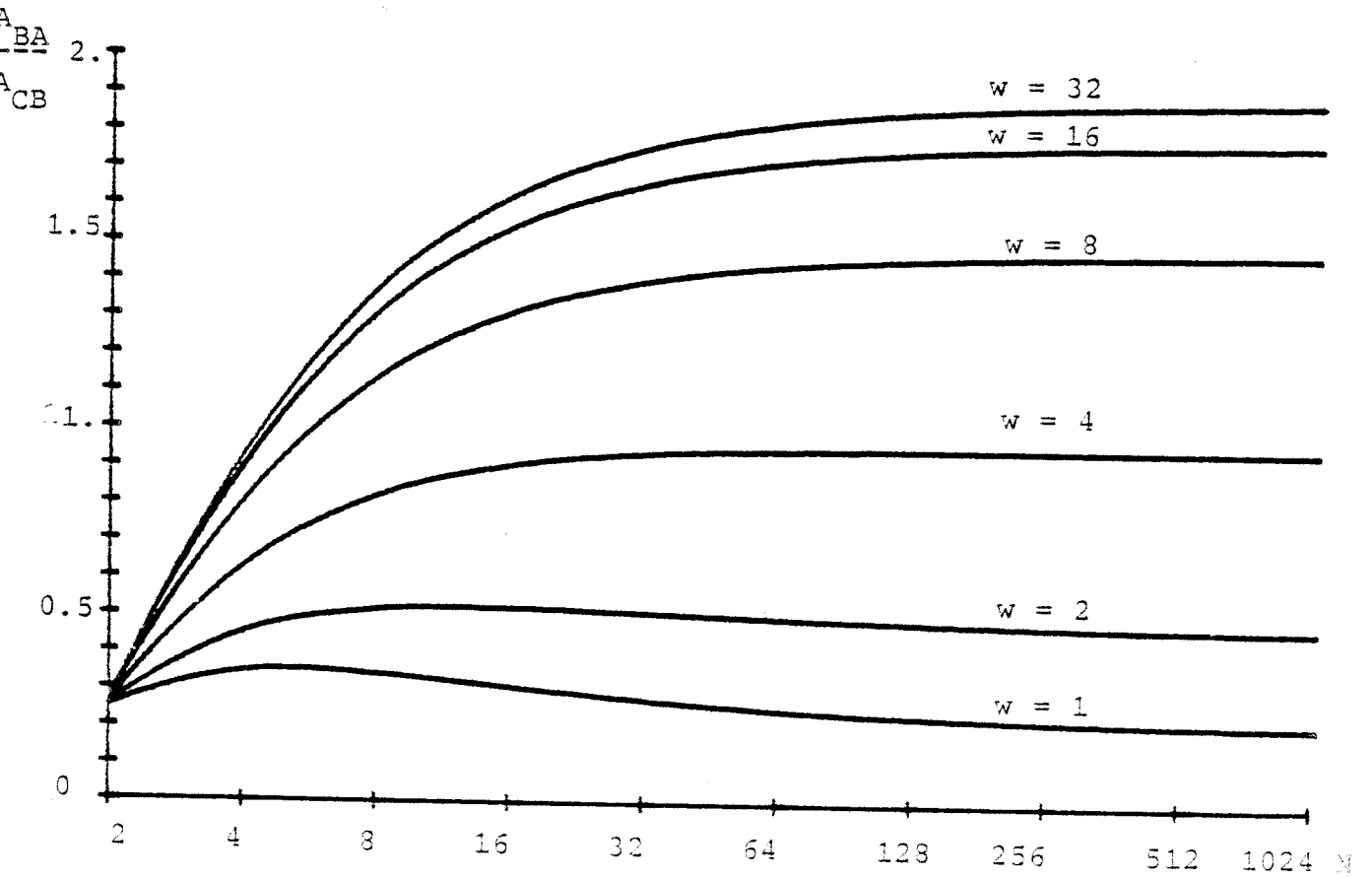


Fig. 6.11

En la figura 6.11 se observa que las redes delta ocupan un área inferior a la de una red crossbar sólo en el caso de que w sea muy pequeño y el número de etapas no sea muy elevado, lo que contradice el análisis discreto realizado en el apartado anterior.

En las figuras 6.8, 6.9 y 6.10 se puede observar que la relación entre el área ocupada por un sistema multibus y la ocupada por un sistema cross-bar se mantiene prácticamente constante con N y tan sólo aumenta ligeramente con el número de líneas de datos w .

6.5 FIABILIDAD Y TOLERANCIA A FALTAS.

Intentar obtener una expresión analítica del tiempo medio entre fallos (MTBF) para cada una de estas redes no es realmente posible.

Los modelos habitualmente empleados para obtener estos parámetros sólo son aplicables a sistemas muy simples o requieren de un conocimiento muy profundo [BORG75] no sólo de la estructura del sistema sino también de sus componentes, de la lógica de arbitraje y control empleada y de las técnicas de detección y corrección de fallos, conocimiento que tan sólo puede ser extraído de la experiencia de aplicaciones concretas de estas redes.

No obstante, una aproximación muy simple puede ser realizada bajo la suposición de que estos sistemas no son reparables (no

existen mecanismos de detección y corrección de fallos) y que todos los selectores elementales tienen la misma tasa de fallos λ .

En este caso, los sistemas crossbar y delta que son intolerantes a fallos tendrían una tasa de fallos que sería la suma de las tasas de fallos de cada uno de sus selectores elementales y la fiabilidad de estos sistemas vendría dada por:

$$R = e^{-O\lambda t} \quad (6.13)$$

donde O es el orden de complejidad de la red y que tendría, para el caso discreto, la expresión N^2 para los sistemas crossbar y a $N \log_a N$ para las redes delta a x a.

El caso de los sistemas multibus es bastante diferente, ya que el fallo de un bus o uno de los elementos de interconexión a un bus no provoca el fallo del sistema. En este caso y, admitiendo que el sistema tiene un funcionamiento admisible siempre que, al menos, $K/2 + 1$ buses de los K existentes no hayan fallado, se tiene una expresión de la fiabilidad en la forma:

$$R = \sum_{i=K/2}^K \frac{K!}{i! (K-i)!} e^{-O_B \lambda t} (1 - e^{-O_B \lambda t})^{K-i} \quad (6.14)$$

donde O_B es el orden de complejidad de un subsistema bus y que, para el caso discreto, vendría dado por N .

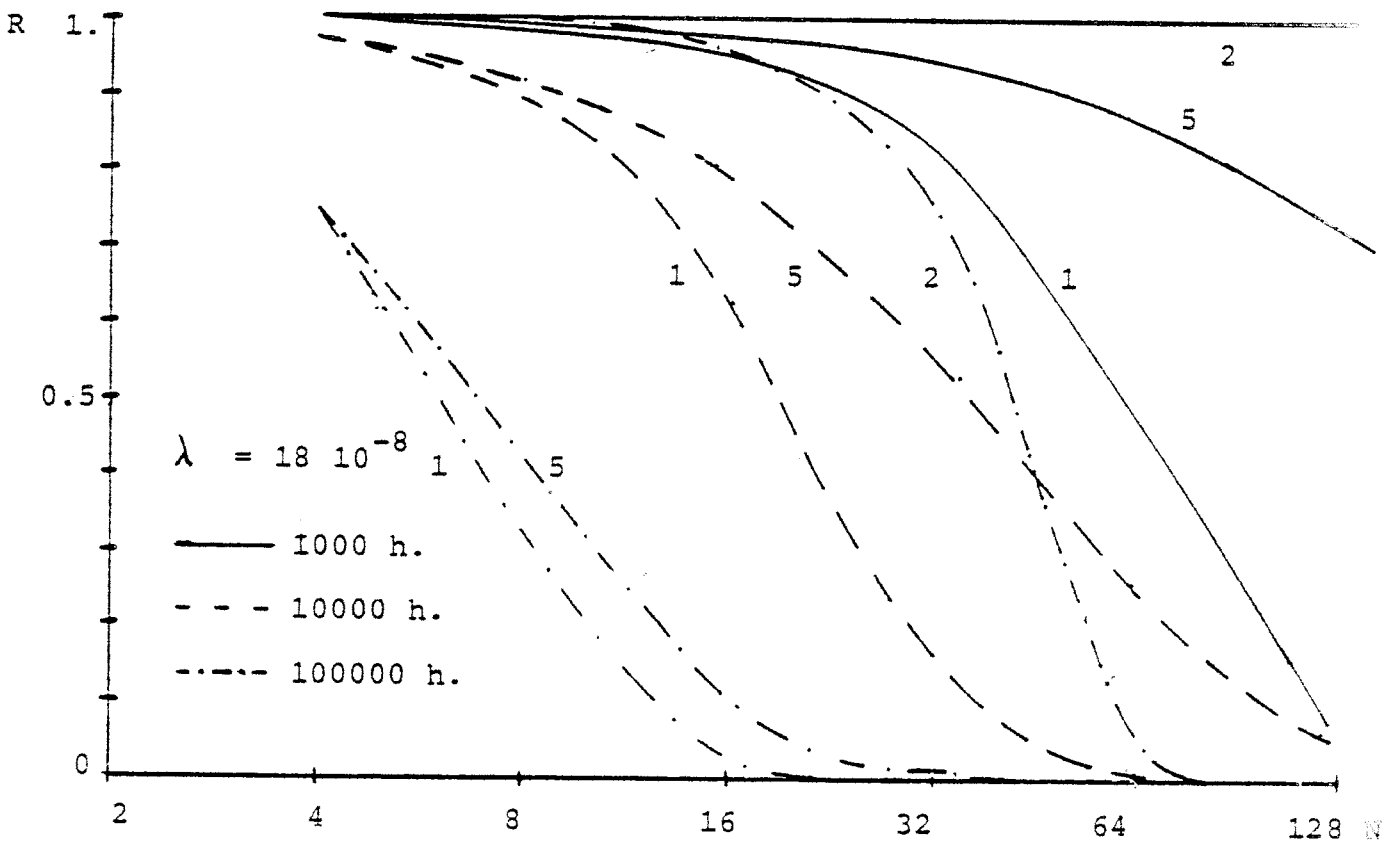


Fig. 6.12

En la figura 6.12 se muestra cómo varía la fiabilidad de cada uno de estos las redes cross-bar (1), multibus con $K = N/2$ (2) y delta $2^n \times 2^n$ (5) con el tamaño de la red para un valor de λ típico para los dispositivos de Media Escala de Integración, pudiendo observarse la gran fiabilidad de los sistemas multibuses como consecuencia de su gran tolerancia a faltas.

Las redes delta, en cuanto que son menos complejas que los sistemas cross-bar, son lógicamente más fiables.

Ni que decir tiene que el análisis anterior es extremadamente simple. No se ha tenido en cuenta la distinta naturaleza que puede tener cada uno de las faltas que pueden originarse en cada uno de los selectores. Cada una de estas distintas causas de fal-

ta puede originar un fallo distinto o no introducir fallo alguno e incluso cada tipo de falta puede tener una tasa de ocurrencia diferente. Tampoco se ha tenido en cuenta el posible fallo de las líneas de interconexión o de los drivers entre selector y pista.

Las redes cross-bar difícilmente pueden hacerse tolerantes a faltas. Tan sólo una duplicación de la red o la existencia de una línea de selectores auxiliares que conmutara ante un fallo podría hacer tolerante a faltas una red cross-bar, pero a costa de un enorme aumento en su ya gran complejidad.

Diferentes soluciones se han propuesto para aumentar el grado de tolerancia de las redes delta. La más simple de ellas consiste en aumentar el número de etapas lo que se traduce en proporcionar caminos alternativos para una misma interconexión procesador-memoria. Esta solución es bastante práctica y poco costosa.

Otras soluciones tales como las redes d-duplicadas y d-dilatadas presentadas en el capítulo 5 aumentan el grado de tolerancia pero también aumentan grandemente el orden de complejidad.

Padmanabhan y Lawrie proponen unas redes omega modificadas que unen N procesadores con N memorias mediante selectores $B \times B$ ($B = 2^b$, $N = 2^n$, pero n no divisible por b). Los autores demuestran que, para estas redes, aparecen $R = B^{\lfloor n/b \rfloor}$ caminos diferentes entre un mismo procesador y una misma memoria, siendo además estos caminos disjuntos; es decir, sólo comparten un selector en la primera y en la última etapa de la red. Estas redes son, por

tanto, intolerantes a la presencia de faltas en un selector de la primera o última etapa pero presentan un grado R de redundancia frente a la falta de un selector en una etapa intermedia.

Por último Sawrirajan y Reddy [AGRA82] proponen hacer las redes delta recirculantes, es decir, cada información atraviesa dos veces la red antes de alcanzar la memoria deseada. El efecto es similar al de aumentar el número de etapas, sólo que se ahorran selectores a costa de complicar los algoritmos de control y arbitraje y aumentar la probabilidad de conflicto y el retraso introducido por la red.

CAPITULO 7.

EXTENSION DE LOS MODELOS.

7.1 INTRODUCCION.

En los capítulos anteriores hemos estudiado las redes de interconexión procesador-memoria mediante modelos muy simples en los que se asume que todos los procesadores (y todas las memorias) son indistinguibles entre sí, que los procesadores dirigen sus accesos a todas las memorias con igual probabilidad, etc.

En los sistemas reales rara vez se dan estas condiciones: cada procesador ejecutará un programa diferente caracterizado por una secuencia particular de accesos a un número determinado de memorias.

Plantearse un modelo general que incluya todas las posibles desviaciones de los modelos idealizados estudiados previamente resulta totalmente inabordable, pero sí es posible estudiar el comportamiento de los sistemas frente a cada una de estas desviaciones por separado y éste es el objetivo de este capítulo.

Nuestro análisis se centrará exclusivamente y, por razones de simplicidad, en los sistemas "cross-bar" ya que son los más fácilmente modelables. Los resultados obtenidos sino cuantitativamente, al menos sí cualitativamente, son aplicables a los otros

tipos de redes.

El efecto de cada uno de estas no idealidades se estudiará, como siempre hemos pretendido hasta ahora, en los casos discreto y continuo en el tiempo. Dentro del análisis discreto en el tiempo pueden encontrarse algunos modelos en la bibliografía, los cuales serán presentados y comparados. En el análisis continuo en el tiempo se hará uso de la propiedad que tienen los modelos que analizaremos de admitir una solución en "forma de producto", por lo que serán resueltos mediante análisis paramétrico o por aplicación directa de las fórmulas propuestas por Baskett, Chandy, Muntz y Palacios en BASK75.

7.2 ANALISIS DISCRETO EN EL TIEMPO.

7.2.1 Modelos no uniformemente referenciados.

Los modelos uniformemente referenciado son adecuados para representar sistemas cuyas memorias están interpaginadas por los bits menos significativos de las direcciones. En este apartado estudiaremos, en el caso discreto, el efecto de entrelazar las memorias por los bits más significativos y el efecto de utilizar una matriz arbitraria de acceso estático.

Consideraremos en primer lugar este segundo caso. Para ello supondremos que la probabilidad de que un procesador i haga una petición a memoria j (que denominaremos por x_{ij}) tiene un valor constante pero arbitrario, distinto en general de $1/M$ (el caso

$x_{ik} = 1/M$ para todo i y todo k sería el caso uniformemente referenciado).

La aplicación de un modelo exacto requeriría en este caso conocer en cada estado no sólo cuántos procesadores hay en cada cola, sino además conocer cuáles son los procesadores que se encuentran en cada una de las colas. Skinner y Ascher [SKIN69] resolvieron este caso para un sistema muy simple ($N = M = 2$, $m = 1$). Para casos más complejos un modelo exacto no es tratable.

Bhandarkar propone un modelo aproximado bajo la suposición de que los procesadores en cola son servidos con disciplina FIFO, derivando las siguientes expresiones referenciadas por Hoogendoorn [HOOG77].

$$Y_{ij} = \prod_{k=i}^N (1 - x_{kj}) \quad (7.1)$$

$$T_{ij} = x_{ij} (y_{ij} + (1 + \prod_{k=i}^N x_{kj} (1 - y_{kj})) (1 - y_{ij}))$$

$$\text{con } q_{ij} = T_{ij}/T \text{ y } T = \sum_{j=1}^M T_{ij}$$

$$B = \sum_{j=1}^M (1 - \prod_{i=1}^N (1 - q_{ij})) \quad (7.2)$$

donde y_{ij} es la probabilidad de que el procesador i esté referenciado memoria j en un ciclo dado.

(Nótese que x_{ij} representa la probabilidad de que un acceso sea dirigido a memoria j por el procesador i y es, por tanto, un valor estático independiente de los conflictos mientras que y_{ij}

es la probabilidad estacionaria de encontrar el procesador i referenciado a memoria j y es, por el contrario, un valor dinámico).

Sí bien estas expresiones fueron derivadas por Bhandarkar para el caso en que los procesadores fueran servidos con disciplina FIFO, sus resultados son aplicables al caso general cuando todos los procesadores tienen exactamente las mismas probabilidades de acceso a las memorias ya que entonces [REGI73] los procesadores son indistinguibles.

Como ya se anticipó (Apartado 3.3.4) el modelo de Hoogenboom es igualmente aplicable al caso de memorias no uniformemente referenciadas. El sistema de ecuaciones (3.41), (3.42) y (3.43) es igualmente válido para este caso, sólo que ahora la matriz de acceso estático X puede tomar valores arbitrarios.

El modelo permite tratar incluso el caso en que cada procesador tenga un factor m de acceso a memoria por ciclo distinto, en cuyo caso la expresión (3.43) se transforma en:

$$\sum_{k=1}^M y_{ik} + (1-m_i)/m_i \sum_{k=1}^M p_{ik} = 1 \quad (7.3)$$

$$i = 1, 2, \dots, N.$$

Bhuyan [BHUY85] realiza una extensión de la aproximación de Patel (3.37) para tener en cuenta el caso en que cada procesador tenga una memoria favorita.

Generalizando aún más el modelo de Bhuyan para tener en cuenta matrices de acceso estático arbitrarias y valores de m distintos para cada procesador la expresión del ancho de banda quedaría:

$$B = \prod_{k=1}^M (1 - \prod_{i=1}^N (1 - x_{ik} m_i)) \quad (7.4)$$

donde, adoptando la suposición de Strecker de que las peticiones no atendidas en un ciclo son ignoradas en el ciclo siguiente, $(1 - x_{ik} m_i)$ representa la probabilidad de que el procesador i no referencie memoria k en un ciclo dado, y $\prod_{i=1}^N (1 - x_{ik} m_i)$ representa la probabilidad de que ningún procesador referencie memoria k en un ciclo dado.

Es de notar que el método de Bhuyan es de una gran simplicidad y aplicable cualquiera que sea la política de colas que se considere (dado que este método ignora las colas) pero los resultados obtenidos de su aplicación difieren en muy alta medida de los resultados exactos como se muestra en las figuras 7.4, 7.5 y 7.6 al compararlo con resultados de simulación.

En estas figuras se expresan las diferencias en tantos por ciento relativos entre los resultados obtenidos por aplicación de los métodos de Hoogendoorn y Bhuyan y los resultados obtenidos por simulación para el sistema 8×8 en dos casos distintos:

- a) Procesadores Sesgados hacia su Memoria Favorita con árbitro neutral (figuras 7.4 y 7.5).

$$x_{ii} = \alpha ; i = 1, 2, \dots, N \quad (7.5)$$

$$x_{ij} = (1 - \alpha)/(M - 1) ; i = 1, 2, \dots, N \text{ j diferente de } i$$

b) Procesadores sesgados hacia una misma Memoria. En este caso el árbitro es indiferente (figura 7.6).

$$x_{i1} = \alpha ; i = 1, 2, \dots, N \quad (7.6)$$

$$x_{ij} = (1 - \alpha)/(M - 1) ; i = 1, 2, \dots, N ; j = 1, 2, \dots, M$$

El modelo de Hoogendoorn es bastante seguro (sobre todo para valores de $\alpha > 0.5$), siendo aplicable (con modificaciones) a cualquier política de manejo de colas que asigne prioridades estáticas a los procesadores. Por ejemplo, para el caso en que el procesador 1 tenga máxima prioridad en el acceso, luego el 2 y así hasta el enésimo, del sistema (3.41), (3.42) y (3.43) tan sólo se modificarán las ecuaciones de (3.43) que quedarán en la forma:

$$P_{1k} = y_{1k}$$

$$P_{2k} = y_{2k} (1 - y_{yk}) \quad (7.7)$$

.

$$P_{Nk} = y_{Nk} \prod_{i=1}^{N-1} (1 - y_{iyk})$$

$$k = 1, 2, \dots, M$$

lo que se ha denominado en las figuras modelo de Hoogendoorn con prioridades.

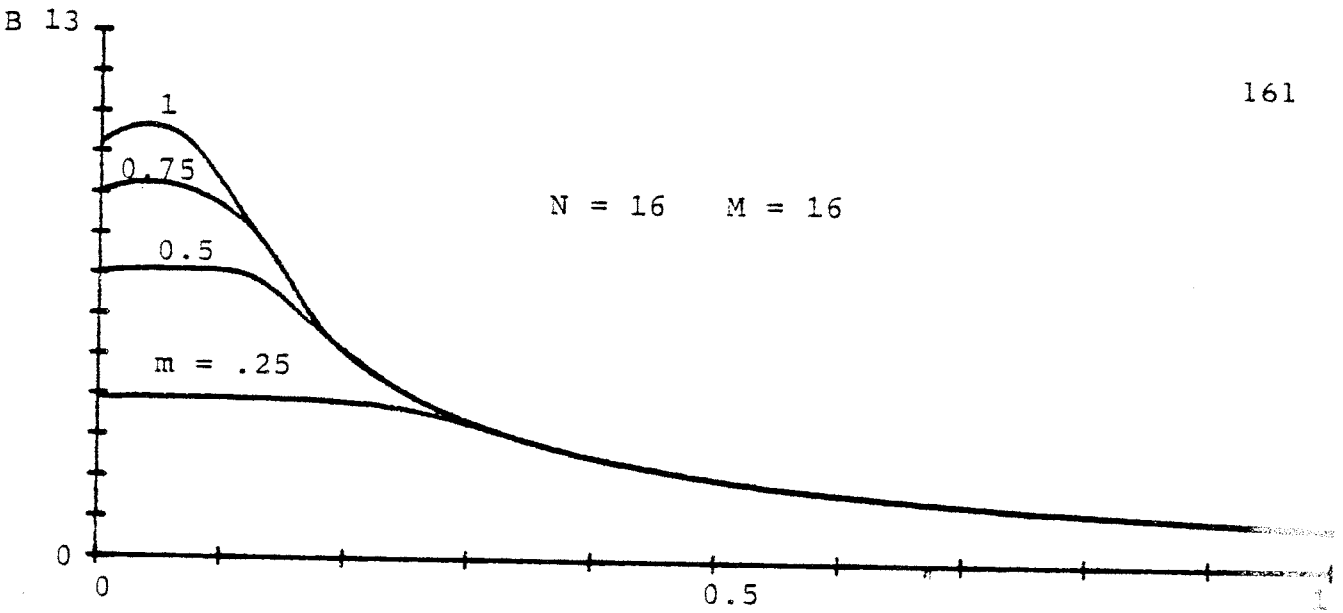


Fig. 7.1 Procesadores sesgados a una única Memoria.

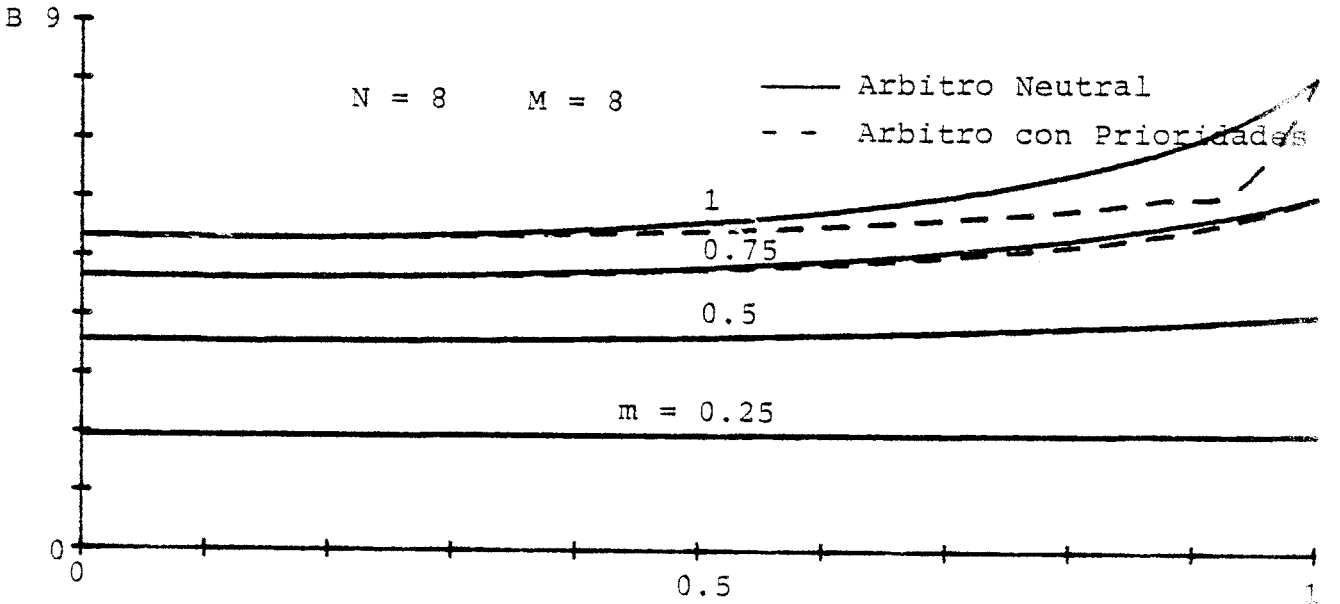


Fig. 7.2 Procesadores con Memoria Favorita.

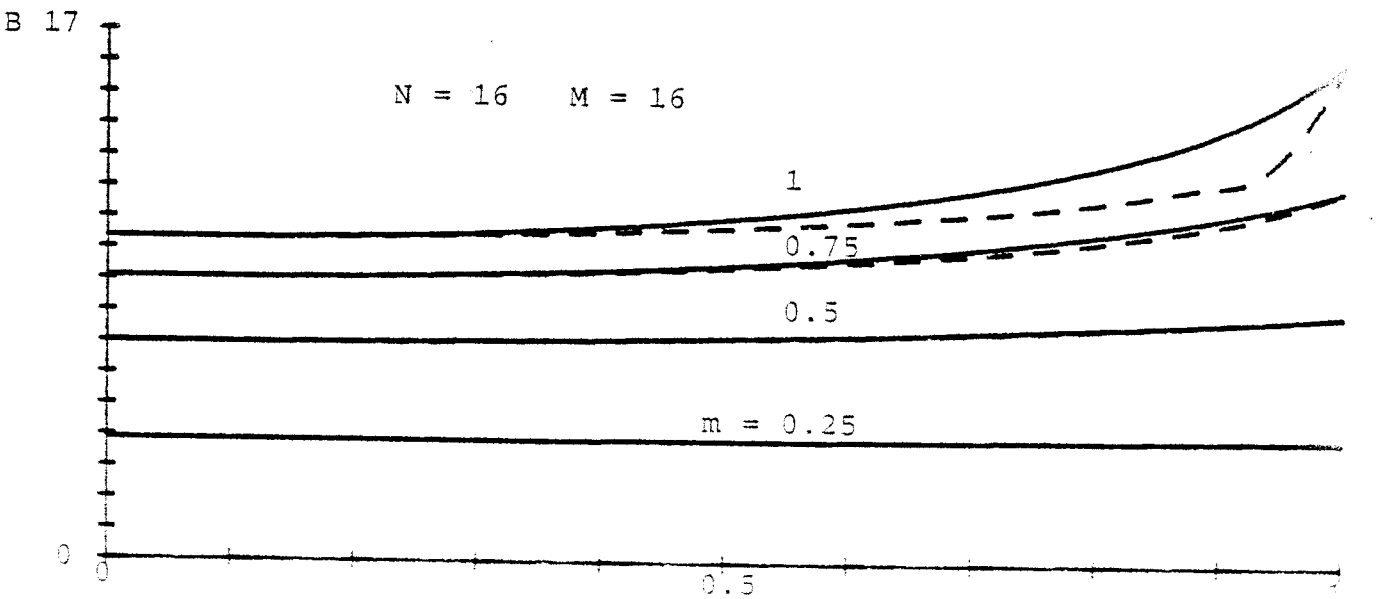


Fig. 7.3 Procesadores con Memoria Favorita.

B 10 %

1 = modelo de Hoogendoorn
2 = modelo de Hoogendoorn con prior.
3 = Modelo de Bhuyan

N = 8 M = 8 m = .25

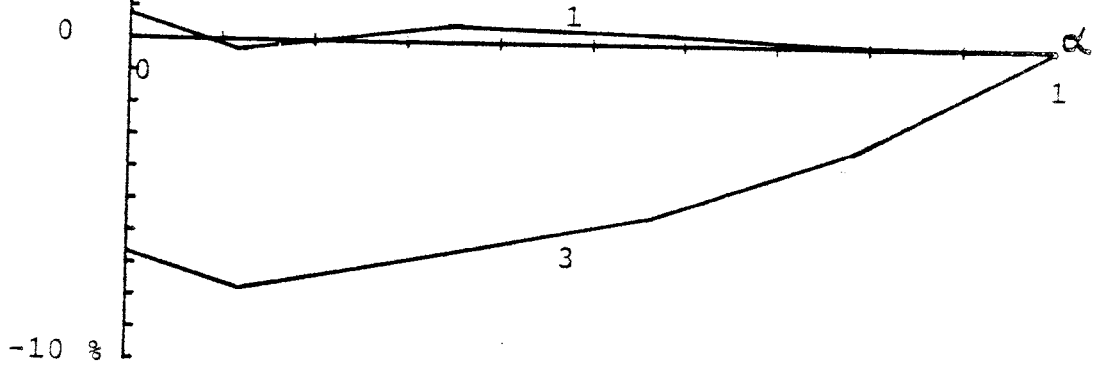


Fig. 7.4

B 10 %

N = 8 M = 8 m = 0.5

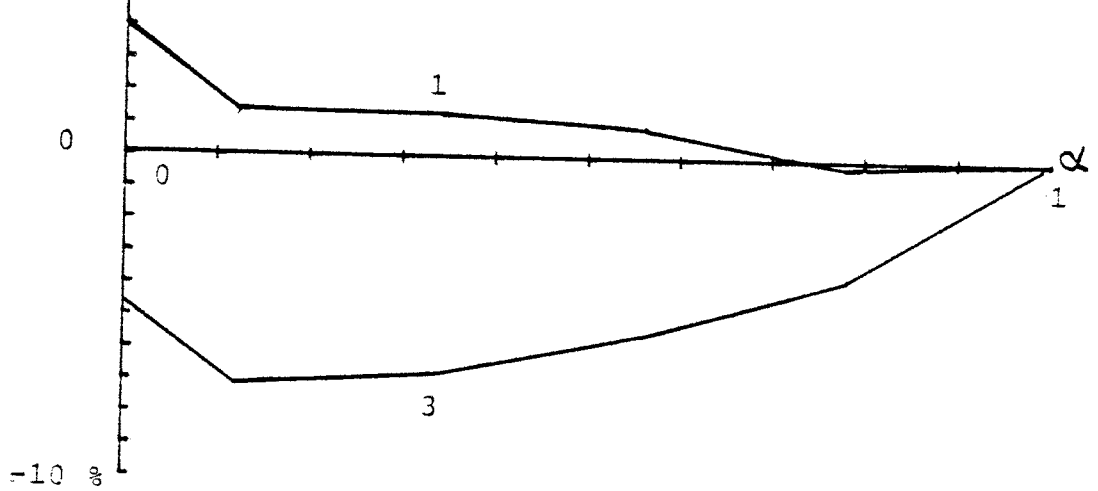


Fig. 7.5

B 10 %

N = 8 M = 8 m = .25

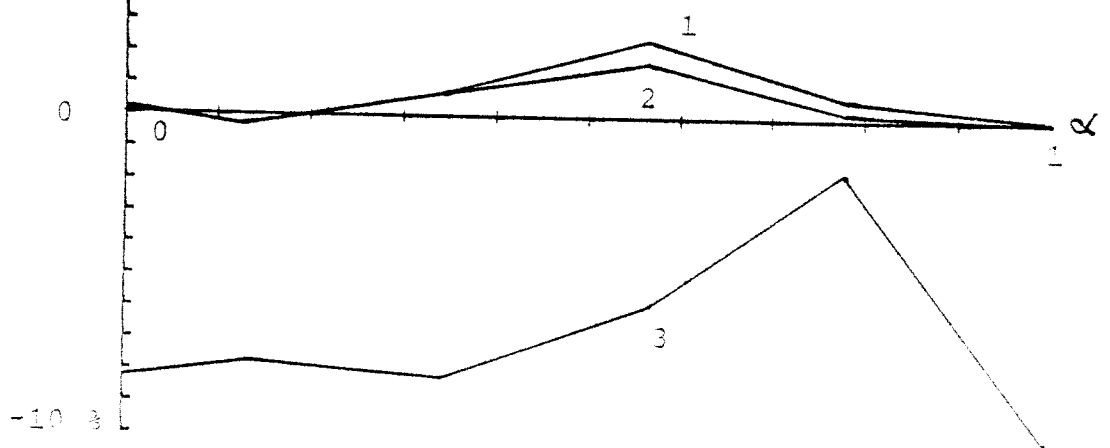


Fig. 7.6

En las figuras 7.1, 7.2 y 7.3 se muestra como varía el ancho de banda a medida que varía α en los tres casos siguientes:

- a) Procesadores sesgados hacia una única memoria (figura 7.1).
- b) Procesadores sesgados hacia su Memoria Favorita (figuras 7.2 y 7.3) con las siguientes políticas de arbitraje de las memorias:
 - b.1) Arbitro neutral
 - b.2) Arbitro con prioridades

Los resultados presentados han sido obtenidos a partir del modelo de Hoogendoorn en el caso b.1 y a partir del modelo de Hoogendoorn con prioridades en los casos a y b.2.

Nótese que el valor del ancho de banda en el caso a (Procesadores sesgados a una única memoria) es independiente de la política seguida para el arbitraje de las memorias por lo que tanto el modelo de Hoogendoorn como el de Hoogendoorn con prioridades son igualmente aplicables, siendo éste último, sin embargo, de menor complejidad de cálculo.

En las figuras 7.1, 7.2 y 7.3 se pueden observar varios resultados de interés. Para el caso a (procesadores sesgados a una única memoria) se observa como era lógico esperar, que el valor de $\alpha = 1/M$ proporciona los mayores anchos de banda ya que, en este caso, el modelo es uniformemente referenciado. Al aumentar el valor de α la memoria favorita a todos los procesadores se

convierte en el cuello de botella del sistema reduciéndose muy rápidamente el ancho de banda. Para $\alpha = 1$ el sistema pasa a tener, de hecho, una única memoria compartida, y para valores suficientes de m , el ancho de banda tiende asintóticamente a 1 al aumentar N .

En el caso b puede observarse que la política de arbitraje de las memorias no tiene un efecto significativo sobre las prestaciones del sistema (lo que fue igualmente apreciado por Hoogenboom quien realizó la simulación de otras políticas de arbitraje con análoga conclusión), salvo para sistemas muy cargados (m próximo a 1).

El valor máximo del ancho de banda se corresponde con $\alpha = 1$, ya que en este caso (y supuestas diferentes las memorias favoritas de cada procesador) no hay conflicto en la red y el ancho de banda de un sistema $N \times N$ será N .

En cuanto al caso de memorias interpaginadas por los bits más significativos de las direcciones, A. S. Sethi y N. Deo [SETH79] estudiaron el sistema $N \times M$ con $m = 1$.

El interpaginado por los bits de mayor orden se modela en la forma siguiente:

Sea $r_i(k) = j$ la memoria referenciada por el procesador i en su k ésimo acceso. Entonces la probabilidad de que en el $(k + 1)$ ésimo acceso el procesador i referencie la memoria h será:

$$P [r_i (k+1)=h / r_i(k)=j] \begin{cases} \alpha \text{ si } j = h \\ (1 - \alpha)/(M - 1) \text{ si } j \neq h \end{cases} \quad (7.8)$$

Nótese que, para $\alpha = 1/M$ el problema se reduce al caso uniformemente referenciado estudiado en el apartado 3.3.

Los autores resuelven de forma exacta, mediante el uso de cadenas discretas finitas de Markov, los casos más simples para $\alpha = 1$ (sistemas $N \times 2$, $2 \times M$ y $3 \times M$) encontrando que el ancho de banda para estos casos es de la forma

$$B = M N / (M + N - 1) \quad (7.9)$$

Se demuestra que igual resultado se obtiene para cualquier valor de α en el caso de considerar el sistema $N \times M$ en tiempo continuo con una distribución exponencial de los tiempos de servicio de las memorias. Este resultado se obtiene de la aplicación de las fórmulas de Jackson y Gordon y Newell (Apartado 2.4) para el caso de una matriz de ruta $R = [r_{ij}]$ dada por:

$$r_{ij} = \begin{cases} \alpha \text{ si } i=j \\ (1 - \alpha)/(M - 1) \text{ si } i \neq j \end{cases} \quad (7.10)$$

la demostración puede encontrarse en [SETH79] y [BHAN73].

La aplicación de la fórmula (7.9) en el caso discreto con tiempos de servicio de un ciclo resulta ser una buena aproximación

ción para cualquier sistema $N \times M$ siempre que $\alpha > 0.75$, probándose se por simulación que a menores valores de α se obtienen mayores anchos de banda, obteniéndose un máximo para $\alpha = 1/M$ como era lógico esperar.

Nótese que al caso estudiado por Sethi y Deo no le puede ser aplicado al modelo de Hoogendoorn en cuanto que éste parte de una matriz estática de accesos, X , que se mantiene en todos los ciclos.

Otros modelos no uniformemente referenciados han sido considerados en la bibliografía. Así, Siomalas y Bowen [SIOM83] consideran tres arquitecturas $N \times M$ para el caso de memoria favorita y $m = 1$, bajo la suposición (algo extraña) de que L peticiones sucesivas de acceso de un procesador a una misma memoria no favorita viene seguida necesariamente de un acceso a la memoria favorita. Los autores sólo resuelven los casos más simples y simulan los casos de mayor complejidad.

Esta hipótesis de los autores tiende a reducir el efecto de la interferencia en el acceso a memoria ya que evita el que un procesador se mantenga más de L ciclos requiriendo acceso a una memoria de la que es favorita otro procesador. Sin embargo, la realización física de esta restricción en un sistema real es difícilmente justificable. De los tres modelos propuestos por estos autores, el más interesante es el denominado "1 x 2" que puede ser modelado en la forma representada en la figura 7.7 en donde M_i es la memoria favorita del procesador P_i .

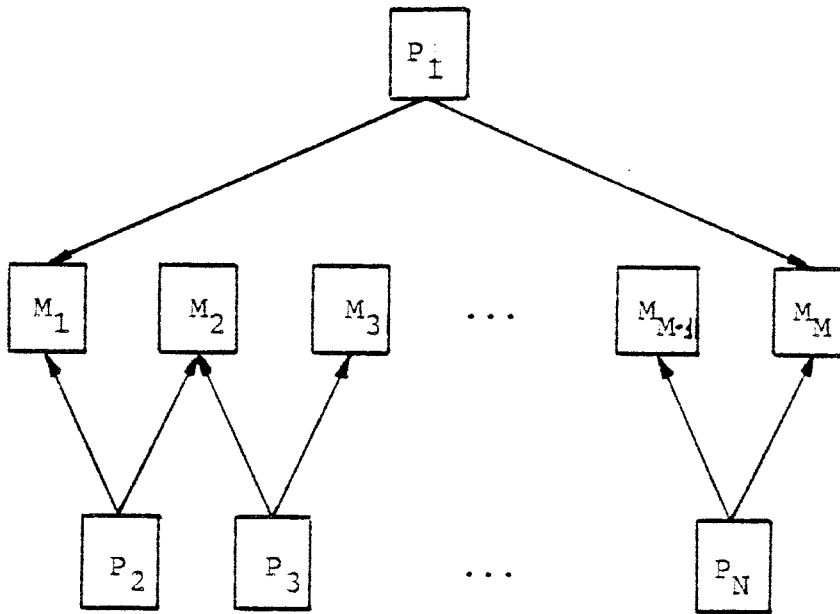


Fig. 7.7. Sistema "1 x 2".

Los autores demuestran que este sistema tiene unas prestaciones coste/servicio superiores a las de un sistema completamente interconectado para valores de N y L razonables y siempre que el tráfico por ambas redes sea el mismo. Sin embargo, el nivel de multiprocesamiento del sistema 1 x 2 es extremadamente bajo y, por tanto, el acceso a una información contenida en una memoria no contigua originará, en la mayor parte de los casos, un elevado tráfico añadido por la red de interconexión lo que hará bajar dramáticamente las prestaciones del sistema.

De este apartado pueden extraerse varias conclusiones interesantes.

En primer lugar aparece como conveniente interpaginar las memorias por los bits menos significativos lo que contribuye a hacer más arbitrarios los accesos y a reducir el nivel de conflicto. Por otra parte, también interesa conseguir, en la medida de lo posible que cada procesador tenga una memoria favorita dis-

tinta a la cual dirige la mayor parte de sus accesos.

Estas dos conclusiones pueden parecer contradictorias pero no los son necesariamente. Nótese que dirigir los accesos hacia memorias favoritas distintas implica reducir el nivel de multiprocesamiento obligando en algunos casos a la repetición de zonas de memoria común en distintos módulos (en el caso de que la zona replicada sea una zona de datos esta replicación puede dar origen a problemas de consistencia de base de datos en los que no vamos a entrar). Pero este resultado, de alguna forma, viene a recordarnos que todo aquello que no vaya a ser compartido no introduce ningún conflicto y es preferible que se almacene en una zona de memoria privada.

Por otra parte, aquellas zonas que vayan a ser compartidas entre un número elevado de procesadores (y que deberán estar, por tanto, en memoria común) originarán un nivel de conflicto más reducido si cada una de ellas se localiza en un número elevado de número de memorias y su acceso se hace aleatorio, efectos ambos que se consiguen interpaginando por los bits menos significativos de las direcciones.

Este efecto también fue observado por Sastry y Kain [SAST75] quienes recomiendan además separar la zona de datos y la zona de códigos de instrucción. Nótese que el efecto de interpaginado es más notable en el caso de las zonas de código común, dado que el acceso a estas zonas tiene una clara componente secuencial (tan sólo alterada por los saltos) y el interpaginado por los bits menos significativos tiene el efecto de aleatorizar estos accesos.

7.2.2 Sistemas con más de un ciclo por acceso. (El problema del "cache").

Patel [PATE82] hace una extensión de su modelo (apartado 3.3.2) para el caso en que el número de accesos por ciclo sea inferior a 1. Este modelo es representativo del problema del "cache" en que cada procesador dispone de una zona de memoria privada donde almacenar los datos sobre los que trabaja en cada momento. La probabilidad de que un procesador referencie una dirección no contenida en el "cache" en un ciclo dado, que denominaremos factor de pérdida del "cache" ("cache miss ratio"), es representado en nuestro caso por el parámetro m o probabilidad de acceso a memoria común por ciclo.

Cada vez que se produce una referencia a una dirección no contenida en el "cache", el procesador realiza una petición a memoria común y tras ganar su acceso reemplaza el contenido del "cache" actual por el correspondiente a la nueva dirección referenciada.

Varios autores, entre otros [KAPL73], [CENS78], [RAO78] [LEHM80], han estudiado diferentes políticas de reemplazamiento del "cache" tendentes a reducir el factor m de pérdida.

En realidad, el factor de pérdida del "cache" será algo inferior a m dado que no todos los ciclos de CPU conllevan accesos a memoria.

Dado que una vez reemplazado el "cache" el primer acceso a éste no producirá ninguna falta, el modelo de Patel asume que un procesador no podrá requerir acceso a memoria común hasta haber transcurrido, al menos, un ciclo tras su último acceso.

Por cada T ciclos en los que un procesador no ha realizado ninguna petición de acceso habrá una media de mT accesos cada uno de los cuales requiere, en valor medio, t_c ciclos (durante los cuales se reemplaza el "cache") y pasará una media de w ciclos (w en principio desconocidos) en espera antes de ganar el acceso.

La utilización de un procesador o número medio de ciclos que ejecuta en el "cache" vendrá dada por:

$$U = T / (T + mT(w + t_c)) \quad (7.11)$$

y el ancho de banda de las memorias

$$B = NmTt_c / (T + mT(w + t_c)) \quad (7.12)$$

$$\text{de donde } B = Nmt_c U \quad (7.13)$$

Para simplificar el modelo Patel considera que los, en valor medio, w ciclos durante los cuales un procesador requiere un acceso más los t_c ciclos que consume en el acceso se comportan tal como lo harían $w + t_c$ peticiones de acceso independientes. De esta forma, el sistema se comporta como un sistema completamente intrconectado con un tiempo de acceso de 1 ciclo y una probabilidad de acceso por ciclo dada por:

$$m' = mT(w + t_c)/(T + mT(w + t_c)) = 1 - U \quad (7.14)$$

cuyo ancho de banda puede ser aproximado en la forma:

$$B = M [1 - (1 - m'/M)^N] \quad (7.15)$$

e igualando a (7.13)

$$Nmt_c U = M [1 - (1 - (1 - U)/M)^N] \quad (7.16)$$

fórmula iterativa cuyos resultados son muy buenos para valores $mt_c > 0.5$.

El modelo puede ser extendido al caso en que los procesadores puedan requerir acceso en el ciclo siguiente al de finalización de un acceso, (7.13) se transforma en

$$B = (N m t_c U) / (m - 1) \quad (7.17)$$

y (7.14) en

$$m' = \frac{\frac{m T}{1 - m} (m + t_c)}{T + \frac{m T}{1 - m} (m + t_c)} \quad (7.18)$$

y por tanto

$$N \frac{mt_c}{1-m} U = M \left[1 - \left(1 - \frac{1-U}{M} \right)^N \right] \quad (7.19)$$

7.2.3 Modelos con diferentes políticas de arbitraje de las memorias.

En general los teoremas demostrados por Regis [REGI73] nos permiten asegurar que el ancho de banda es indiferente de la política de disciplina de colas siempre que la llegada de procesadores a las colas sean independientes entre sí como ocurre en todos los casos considerados hasta ahora.

No obstante, si consideramos ahora que los procesadores tienen diferentes prioridades en los accesos, aun cuando el ancho de banda de las memorias sea el mismo, la utilización de cada uno de los procesadores será diferente.

Pocos estudios se han hecho del comportamiento de estos sistemas para diferentes políticas de arbitraje de las memorias.

Yen aplicó su modelo (apartado 3.3.2) para el caso de que haya n grupos de procesadores con diferentes prioridades. Si cada grupo tiene N_i procesadores con probabilidades de acceso por ciclo m_i , el ancho de banda B de las memorias vendrá dado por

$B = \sum_{i=1}^n B_i$, siendo B_i el ancho de banda debido al grupo de procesadores i .

El grupo 1 de los procesadores de mayor prioridad se comporta, de hecho, tal cual si no existiera otro tipo de procesadores y su ancho de banda puede calcularse en la forma iterativa vista en (3.52)

$$B_1 = N_1 f_1 m_1 = M \left[1 - \left(1 - \frac{m_1 f_1}{M} \right)^{N_1} \left(1 - \frac{1 - \left(1 - \frac{1-f_1}{M} \right)^{N_1}}{M} \right)^M \right] \quad (7.20)$$

Para el segundo grupo, con prioridad inferior al primero pero superior al resto, se puede ignorar igualmente la existencia de los grupos de prioridad inferior.

Considerando exclusivamente la existencia de procesadores del grupo 1 y 2

$$B_1 + B_2 = N_1 f_1 m_1 + N_2 f_2 m_2 = M \left[1 - \left(1 - \frac{m_1 f_1}{M} \right)^{N_1} \left(1 - \frac{m_2 f_2}{M} \right)^{N_2} \left(1 - \frac{1 - \left(1 - \frac{1-f_1}{M} \right)^{N_1} \left(1 - \frac{1-f_2}{M} \right)^{N_2}}{M} \right)^M \right] \quad (7.21)$$

donde f_1 ha sido anteriormente determinado en (7.20) y f_2 puede ser calculado en forma iterativa.

Procediendo en forma análoga, para los procesadores de categoría k

$$\sum_{i=1}^k B_i = \sum_{i=1}^k N_i f_i m_i = M \left[1 - \prod_{i=1}^k \left(1 - \left(1 - \frac{m_i f_i}{M} \right)^{N_i} \right) \left(1 - \frac{1 - \prod_{i=1}^k \left(1 - \frac{1 - f_i}{M} \right)^{N_i}}{M} \right)^M \right] \quad (7.22)$$

donde f_1, f_2, \dots, f_{k-1} han sido calculados anteriormente y f_k puede ser calculado en forma iterativa.

En general, cualquiera de los modelos aproximados de la sección 3.3.2 puede ser utilizado en la misma forma a la descrita para el modelo de Yen.

Para el caso en que la probabilidad de acceso por ciclo m de todos los procesadores fuera la misma, el problema antes planteado sería mucho más fácil de resolver. Denominemos B_{NM} al ancho de banda de un sistema $N \times M$. Si el número de accesos por ciclo es 1 y todos los procesadores tienen la misma probabilidad de acceso por ciclo, entonces el ancho de banda debido al procesador de mayor prioridad vendrá dado por

$$B_{1 \times M} = m$$

y el i ésimo procesador con mayor prioridad dará origen a un aumento del ancho de banda

$$\Delta B_i = B_{iM} - \sum_{j=1}^{i-1} B_{jM} ; i = 2, 3, \dots, N \quad (7.23)$$

y por tanto la utilización del procesador i ésimo será

$$U_i = (1 - m) \Delta B_{iM} / m \quad (7.24)$$

donde B_{NM} puede obtenerse a partir de cualquiera de los modelos vistos en el apartado 3.2.

En forma análoga podría haberse considerado que cada nivel de prioridad tiene asignado más de un procesador.

7.3 ANALISIS CONTINUO EN EL TIEMPO.

Tal como se indicó en el apartado 7.1 los modelos que veremos en este apartado admiten todos una solución en forma de producto.

Antes de analizar estos modelos conviene destacar varios hechos interesantes.

En el modelo de un sistema cross-bar visto en el apartado 3.4 se tiene:

1. Cualquier distribución de tiempos (tanto de ejecución en memoria privada o de accesos a memoria común) no necesariamente exponencial pero que admita una transformada racional de Laplace y que tenga por valor medio $1/\lambda$ (o $1/\mu$ en el caso de los accesos a memoria) conduce a los mismos resultados globales (ancho de banda) que el modelo visto en el apartado 3.4.
2. Cualquier política de arbitraje sin interrupciones de las memorias e incluso las políticas de arbitraje con interrupciones tipo "processor-sharing" y "resume LCFS" también conducen al mismo resultado.

7.3.1 Procesadores sesgados hacia una única memoria.

En este modelo los procesadores ejecutan en memoria privada por un tiempo exponencialmente distribuido de media $1/\lambda$. La duración de los accesos es exponencialmente distribuida de media $1/\mu$. La disciplina de arbitraje de las memorias es FIFO. La matriz de "routing" entre procesadores y memorias viene dada por:

$$x_{i1} = \alpha \quad ; \quad i = 1, 2, \dots, N \quad (7.25)$$

$$x_{ij} = (1 - \alpha)/(M - 1) \quad ; \quad i = 1, 2, \dots, N \quad ; \quad j = 2, 3, \dots, M$$

Este caso admite una solución explícita en la forma:

$$P = \frac{\sum_{k_2=0}^N \binom{k_2+M-2}{M-2} \sum_{k_1=0}^{N-k_2} (N-k_1-k_2) \frac{A^{k_1} B^{k_2}}{(N-k_1-k_2)!}}{\sum_{k_2=0}^N \binom{k_2+M-2}{M-2} \sum_{k_1=0}^{N-k_2} \frac{A^{k_1} B^{k_2}}{(N-k_1-k_2)!}} \quad (7.26)$$

$$\text{con } A = \alpha p \quad \text{y} \quad B = \frac{(1-\alpha)p}{(M-1)}$$

En la figura 7.8 se representa cómo varía el número medio de procesadores activos frente a α para diferentes valores de p para un sistema 3 x 3.

Las conclusiones que pueden extraerse de esta figura son análogas a las vistas en el caso discreto. El máximo valor de P se obtiene para $\alpha = 1/M$ que corresponde al modelo uniformemente referenciado.

7.3.2 Procesadores sesgados a memoria favorita.

Este caso es idéntico al anterior sólo que ahora la matriz de "routing" viene dada por:

$$x_{ii} = \alpha \quad ; \quad i=1,2,\dots,N \quad (7.27)$$

$$x_{ij} = (1-\alpha)/(M-1) \quad ; \quad i \neq j.$$

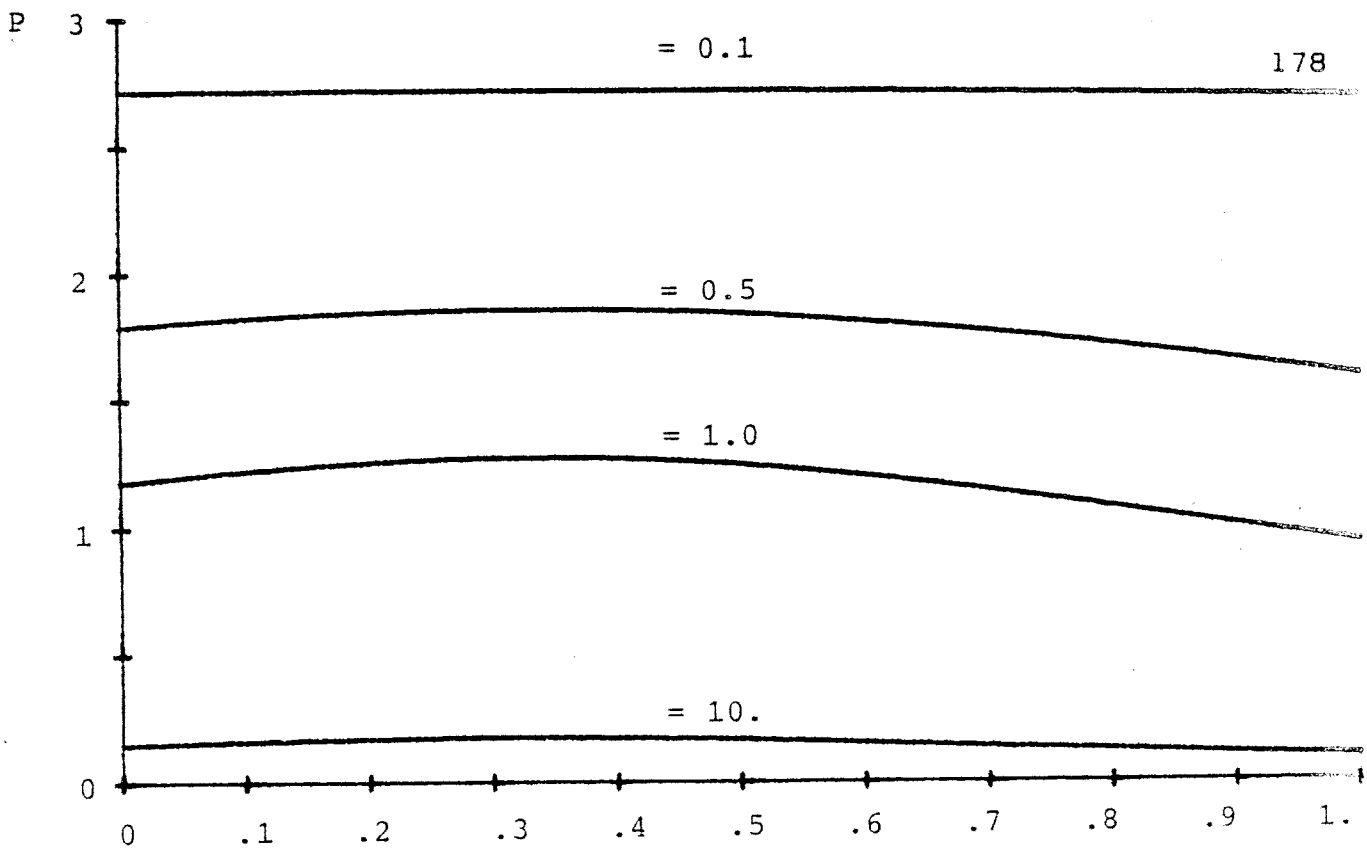


Fig. 7.8

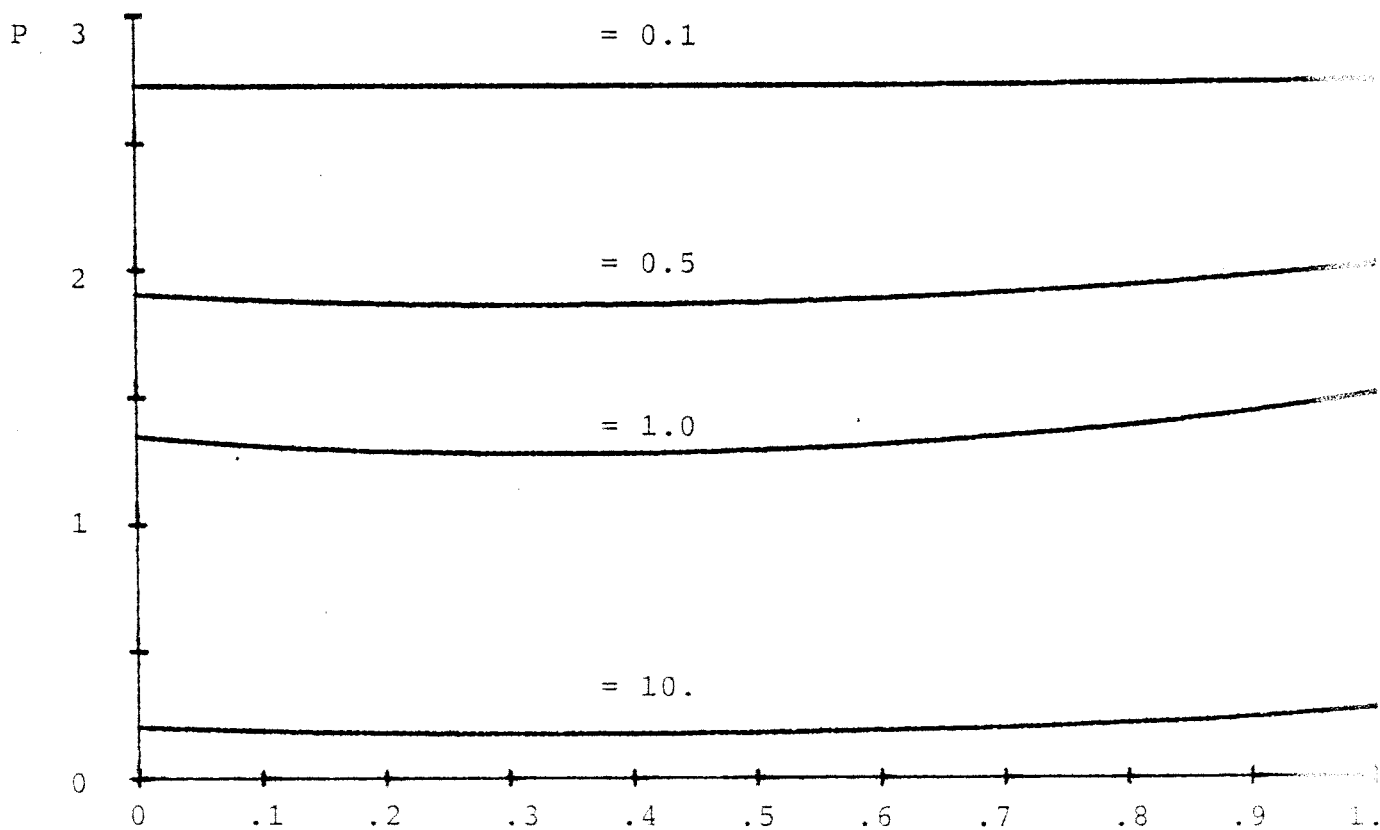


Fig. 7.9

En la figura 7.9 se observa cómo varía el número medio de procesadores activos P frente a α para un sistema 3×3 en el que P tiene la expresión:

$$P = \frac{3+6(A+2B)+3(A^2+5B^2+6AB)}{1+3(A+2B)+3(A^2+5B^2+6AB)+(A^3+14B^3+12A^2B+33AB^2)} \quad (7.28)$$

donde A y B están dados según (7.26).

Nuevamente la conclusión es análoga a la del caso discreto. Los valores más bajos de P (y del ancho de banda $B = P \rho$) se tienen para $\alpha = 1/M$ y los más elevados para $\alpha = 1$.

7.3.3 Los procesadores acceden con diferente velocidad.

En este caso, el procesador P_i accede a memoria privada por un tiempo exponencialmente distribuido de media $1/\lambda_i$. Cada acceso conlleva un tiempo exponencialmente distribuido de media $1/\mu$. La disciplina de arbitraje de las memorias es FIFO.

La matriz de "routing" viene dada por:

$$x_{ij} = 1/M \quad (7.29).$$

$i=1,2,\dots,N$; $j=1,2,\dots,M$.

Para un sistema 3×3 y tomando:

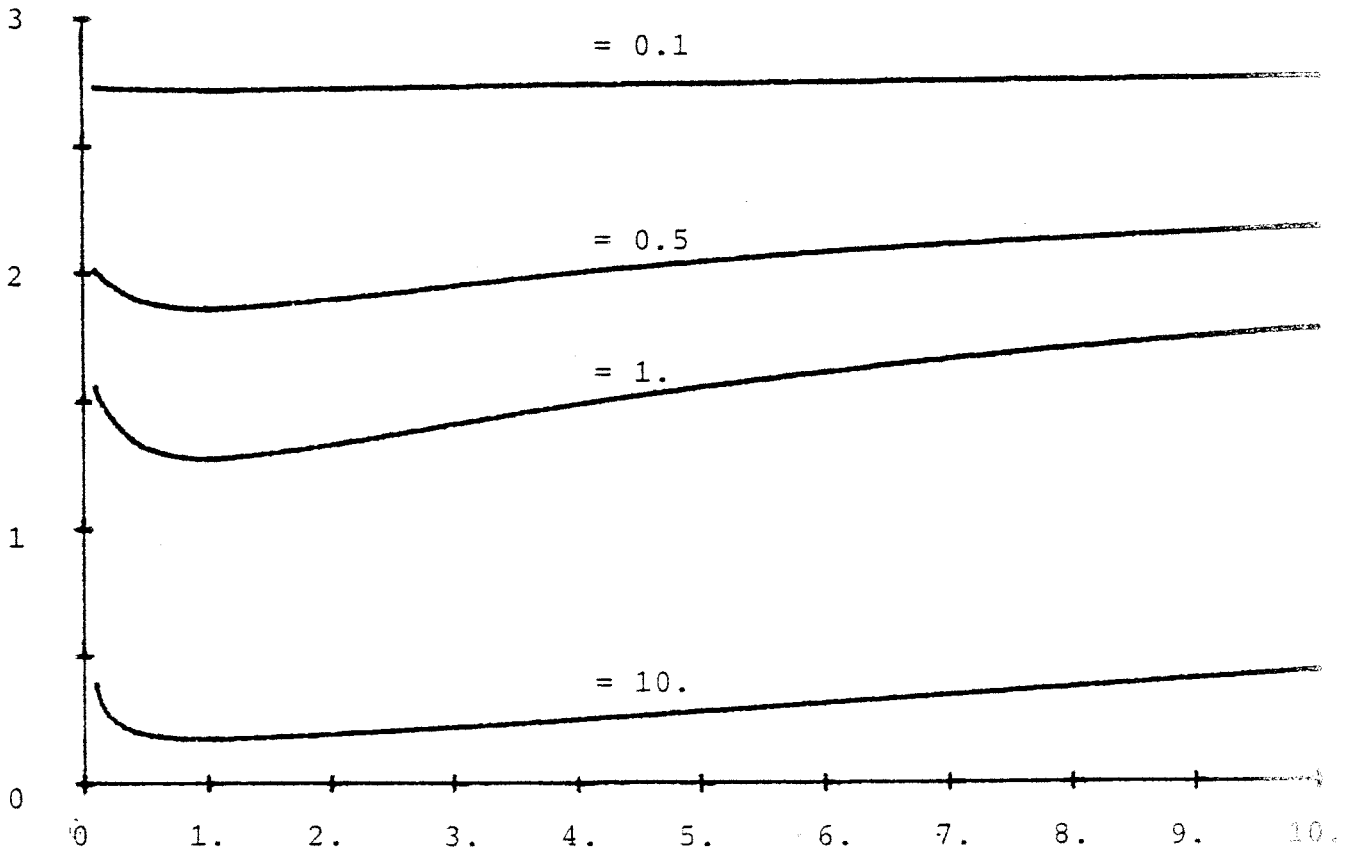


Fig. 7.10

$$p_1 = \frac{k_1}{\mu} \quad p = \frac{k_1 (1+2/\alpha)}{3} \quad k_2 = k_3 = \frac{k_1}{\alpha}$$

$$P = \frac{1.5 \alpha^2 + p_1 \alpha^2 + 2 p_1 \alpha + 4/3 p_1^2 \alpha + 2/3 p_1^2}{.5 \alpha^2 + .5 \alpha^2 p_1 + \alpha p_1 + 4/3 p_1^2 \alpha + 2/3 p_1^2 + 10/9 p_1^3} \quad (7.30)$$

En la figura 7.10 se muestra cómo varía P con α para diferentes valores de ρ .

Se puede observar que el mínimo valor de P se obtiene para $\alpha = 1$ que corresponde al caso en que todos los procesadores acceden con la misma velocidad, dado que sería el caso en que mayor con-

flicto se originaría en la red.

Se puede apreciar que el efecto es tanto más notable mientras menores son los valores de ρ .

Por último cabe destacar que en cualquiera de los casos estudiados las desviaciones obtenidas respecto de los modelos planteados en el capítulo 3 no son muy significativas por lo que los modelos de dicho capítulo representan una adecuada base para el estudio de las prestaciones estructurales de los sistemas "cross-bar".

CAPITULO 8

PAQUETE DE PROGRAMAS DE SIMULACION

8.1 INTRODUCCION.

Como quedó patente en los capítulos anteriores la gran complejidad de los sistemas de interconexión procesador-memoria hace que su tratamiento analítico exacto no sea posible.

Los modelos estudiados en estos capítulos han sido, en todo caso, aproximaciones idealizadas de los problemas reales e incluso estos modelos adquieren tal complejidad que motivan la proliferación de soluciones aproximados.

Dos problemas quedan por tanto abiertos:

1. Establecer la validez de estas soluciones simplificadas.
2. Establecer la validez de estos modelos idealizados cuantificando la influencia de parámetros no considerados en el modelo.

La imposibilidad material y tecnológica en la adquisición o desarrollo de prototipos y la carencia de sofisticados sistemas reconfigurables a los que se hace mención en el capítulo 2 nos

lleva irrevocablemente al desarrollo de simuladores.

Con este fin se ha implementado un paquete de programas de simulación que, por su gran flexibilidad y cómoda utilización permite atacar los problemas antes comentados.

La simulación de redes de interconexión procesador-memoria podría haberse encaminado en dos direcciones:

1. Desarrollo de Simuladores específicos para cada tipo de sistema de interconexión procesador-memoria.
2. Desarrollo de un sistema general de simulación aplicable a todos las arquitecturas.

La primera de las alternativas permite que estos paquetes de simulación sean más compactos en el sentido de una mayor simplicidad y mayor comodidad en la elaboración de los resultados finales.

No obstante, la complejidad intrínseca de los sistemas a simular hace que estos simuladores específicos ([SCHN78], [WIRS81]) no sean en absoluto simples y su reducida flexibilidad no permite evaluar nuevas arquitecturas o modificar las existentes.

En cuanto a la segunda alternativa se han venido utilizando simuladores basados en Redes de Colas los cuales se adaptan con relativa facilidad a la simulación de estos sistemas en tiempo continuo.

El Simulador que aquí se presenta aborda el problema desde un punto de vista original proponiendo una simulación en base a una herramienta ampliamente conocida como son las Redes de Petri.

Las Redes de Petri tienen como principal virtud su gran capacidad de síntesis y su elevada potencia para la representación de sistemas lógicos secuenciales [PETE81].

No entraremos aquí en la descripción de las Redes de Petri, las reglas de evolución de su marcado o las propiedades de las Redes Vivas y Sanas. Una completa información puede obtenerse en [PETE81] y [SILV84].

No obstante, es necesario indicar que las Redes de Petri fueron concebidas como un elemento de descripción de sistemas deterministas y son, por definición, deterministas.

Sin embargo, las Redes de Petri pueden ser tratadas en forma estocástica, para ello basta con utilizar como variables para el franqueo de las transiciones variables aleatorias.

Las Redes de Petri estocásticas son isomórficas [AJM083] con las Cadenas de Markov continuas en tiempo correspondiéndose el marcado de la red con el estado de la cadena. Es posible entonces construir una cadena de Markov equivalente a una red de Petri estocástica (con temporizaciones exponencialmente distribuidas) y obtener las probabilidades estacionarias de cada vector de marcado. Sin embargo, el poder de representación visual que tiene una Red de Petri no se corresponde con un tratamiento analítico sim-

ple resultando más interesante plantear de forma directa la cadena de Markov del sistema ya que la obtenida a partir de la Red de Petri isomórfica es, generalmente, más compleja.

Las Redes de Petri constituyen, por tanto, una adecuada herramienta para la simulación de estos sistemas pero no una herramienta para su tratamiento analítico.

Por razones de simplicidad en vez de usar transiciones franqueables por variables aleatorias hemos utilizado lugares temporizados en los que el valor de la temporización es una variable aleatoria distribuida según una tabla que constituye una entrada para el simulador. El simulador calcula el tiempo de la temporización accediendo a dicha tabla con una variable aleatoria uniformemente distribuida y multiplicando el resultado por el tiempo medio de la temporización definido por el programador.

Un lugar temporizado que recibe una marca no la valida hasta que su temporización ha transcurrido impidiendo por tanto el franqueo de sus transiciones de salida. El efecto es el mismo que considerar variables aleatorias en las transiciones.

8.2 LENGUAJE DE SIMULACION.

Diferentes lenguajes para la representación de Redes de Petri han sido considerados en la literatura. Un lenguaje de gran simplicidad y representatividad fue desarrollado en el Dpto. de Electrónica y Automática de la E.T.S.I.I. de Sevilla y descrito

en [CAMA83] y [CALV85] para un simulador básico de Redes de Petri que, si bien muy limitado, ha servido como base para el desarrollo de nuestro paquete de simulación.

En la figura 8.1 se presenta la descripción según este lenguaje de la Red correspondiente a un procesador de un sistema con una única memoria común en tiempo continuo junto con la representación gráfica de esta Red de Petri.

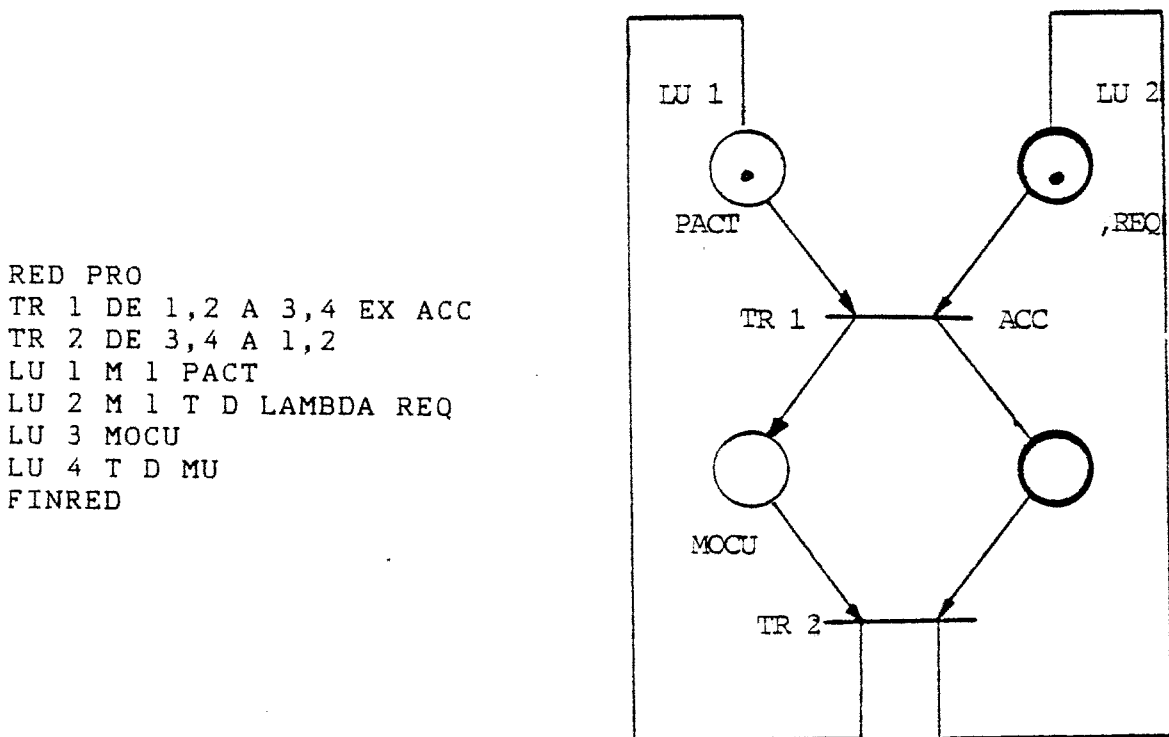


Fig. 8.1 Red de un Procesador en un sistema con una única memoria (bus) común.

Los lugares representados en trazo grueso son lugares temporizados y sólo se marcan una vez transcurrida la temporización correspondiente. La variable en letra griega que acompaña a estos lugares representan el tiempo medio de la temporización. El resto de variables que acompañan a un lugar representan las variables que se activan cuando el lugar está marcado. Las expresiones que

acompañan a las transiciones se corresponden con aquellas expresiones cuyo resultado debe ser un "1" lógico para que se franquee la transición.

En líneas muy generales, una red queda delimitada con las palabras RED y FINRED en la forma:

```

RED (nombre)
Cuerpo de la Red
FINRED

```

Una transición dentro de la red se define mediante la sentencia:

```

TR número DE lugares de entrada A lugares de salida (EX expresión lógica para el franqueo de la transición).

```

Un lugar se define con la expresión:

```

LU número (T D valor medio de la temporización) (M número de marcas iniciales) (lista de variables a activar cuando el lugar está marcado).

```

Las partes presentadas entre paréntesis son opcionales.

En nuestro ejemplo, PACT es una variable que indica que el procesador en cuestión se encuentra ejecutando en memoria privada, REQ indica que el procesador efectúa una petición a memoria común, ACC indica que el procesador ha ganado el acceso a la me-

moria y MOCU indica que el procesador está accediendo a memoria común.

Es de notar que de un lugar temporizado se indica el valor medio de su temporización, la distribución de esta temporización se ve recogida en una tabla auxiliar que sirve de entrada al sistema de simulación. En cada aplicación se admiten hasta dos tipos diferentes de distribución de las temporizaciones (p.e. exponencial y constante).

Aparte de las propias redes pueden definirse asimismo funciones combinacionales en la forma:

FS variable = expresión booleana.

El simulador, por otra parte, es gráfico y permite la visualización en una pantalla semigráfica del estado del proceso. A este fin, sobre un mímico fijo, se pueden definir un conjunto de puntos activos a través de los cuales el operador del sistema introduce órdenes desde teclado (recogidas en forma de variables lógicas de teclado) y sobre los cuales el simulador puede representar hasta cuatro caracteres distintos (definidos por el programador del sistema) representativos del estado del proceso (recogido en forma de variables de pantalla) [TORR84].

Otras variables lógicas de entrada y salida permiten al simulador actuar sobre contactos externos (variables de salida) o leer la posición de contactos externos (variables de entrada).

El simulador considera también la existencia de variables aleatorias uniformemente distribuidas (construidas a partir de un polinomio de orden 31 [PETE61]) y variables acumulativas no booleanas que se incrementan cada vez que son activadas y que permiten retener el número de ciclos que un procesador ha permanecido activo, el número de ciclos que un bus está ocupado, etc.

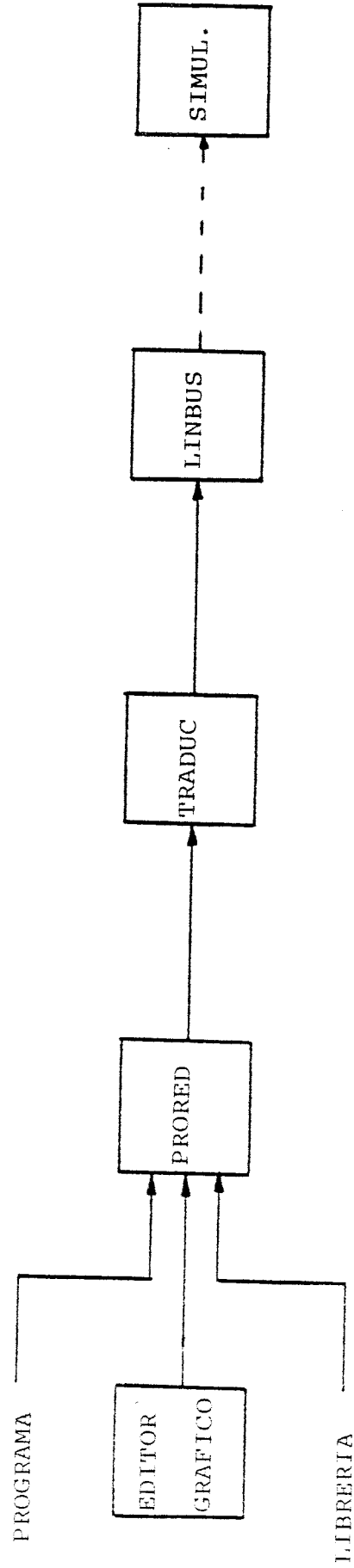
Por último, el lenguaje permite la utilización de MACROS de forma que para un sistema con 16 procesadores basta con la definición de la red de un único procesador englobada en un MACRO e invocar 16 veces esta MACRO variando tan sólo los parámetros de llamada.

En los ejemplos listados en el Apéndice I puede apreciarse la utilización del lenguaje y de los MACROS antes referidos.

8.3 ESTRUCTURA DEL PAQUETE DE PROGRAMAS DE SIMULACION.

El paquete de programas de Simulación consta de 4 módulos principales y varios módulos auxiliares que, a continuación, se describen brevemente.

En la figura 8.2 se representa un esquema de bloques del Sistema de Simulación.



PDP 11

M68000

Fig. 8.2 Estructura del Paquete de Programas de Simulación.

8.3.1 Programa de Expansión de Macros y Preprocesado de Expresiones (PRORED).

En este módulo se expanden los MACROS definidos en el programa y se realiza un primer procesado de expresiones consistente en eliminar paréntesis innecesarios, aplicar las Leyes de Morgan a los paréntesis negados y reordenar cada expresión a fin de poner en primer lugar los términos con menor número de variables.

Este programa está realizado en lenguaje FORTRAN IV y corre sobre un ordenador PDP 11.

8.3.2 Programa de Generación de Tablas (TRADUC).

Este módulo traduce el lenguaje específico de Redes antes comentado en tablas directamente ejecutables por el Simulador.

Estas tablas se conforman en base a una estructura de Listas comentada en el apartado 8.4 de este capítulo.

Realizado en lenguaje FORTRAN IV, corre sobre un ordenador PDP 11.

8.3.3 Programa de Transformación y Envío de Tablas (LINBUS).

Este módulo convierte las tablas generadas por el programa Traductor a formato "standard" (Formato "S") y envía las tablas a la memoria del Simulador.

Realizado en FORTRAN IV, corre sobre un ordenador PDP 11.

8.3.4 Programa de Simulación.

El simulador propiamente dicho, a partir de las tablas anteriormente elaboradas y transmitidas a su memoria, realiza el procesamiento de las redes actualizando el valor de sus salidas tanto externas como de pantalla y leyendo el valor de sus entradas tanto externas como de teclado.

Este programa está realizado, por razones de rapidez de tratamiento de las redes en lenguaje ensamblador del microprocesador de 16 bits 68000 corriendo sobre una tarjeta tipo MVME 101.

8.3.5 Módulos Auxiliares.

De entre los módulos auxiliares que constituyen el paquete de Simulación cabe considerar un editor gráfico que permite la elaboración gráfica del mímico que se presenta durante el proceso de simulación y una Librería de Redes predefinidas. En esta librería se contiene un conjunto de Redes ya definidas para el sistema de simulación. De esta forma, aquellos elementos que son usualmente utilizados tales como Procesadores, Memorias, Selectores, etc. pueden ser definidos por una única vez en la librería y remitir a ella cada vez que uno de estos elementos vaya a ser usado, sin necesidad de volver a definirlos en cada aplicación.

8.4 TRATAMIENTO DE LAS REDES DE PETRI.

El tratamiento de las Redes de Petri en el simulador se encuentra íntimamente ligado a la estructura de la representación interna de las Redes de Petri.

El Simulador que sirvió de base para el desarrollo del sistema que aquí se presenta partía de una representación interna matricial de las Redes de Petri basada en el vector de marcado $M_{n \times 1}(t)$, la matriz de lugares de entrada de las transiciones $E_{n \times m}$ y la matriz de lugares de salida de las transiciones $S_{n \times m}$, cuyos elementos se definen en la forma:

$$e_{ij} = 1 \text{ si } p_i \text{ es lugar de entrada de la transición } t_j \text{ y } 0 \text{ en caso contrario.}$$

$$s_{ij} = 1 \text{ si } p_i \text{ es lugar de salida de la transición } t_j \text{ y } 0 \text{ en caso contrario.}$$

La transición t_j estará válida si y sólo si $E^j \times \bar{M}(t) = 0$, en cuyo caso, $M(t+1) = M(t) \oplus (E + S)^j$ donde \oplus representa la operación OR-exclusiva, A^j representa la columna j -ésima de la matriz A y \bar{A} representa la matriz cuyo elemento \bar{a}_{ij} es el negado del elemento a_{ij} de la matriz A [SILV82].

En general, las estructuras matriciales se caracterizan por una elevada ocupación de memoria y una gran complejidad debido a las operaciones booleanas con matrices lo que les hace, a su vez, ser lentos.

En el simulador aquí desarrollado la evolución del mercado se realiza en forma síncrona, de manera que un lugar que recibe una marca en un ciclo dado no es realmente marcado hasta el comienzo del ciclo siguiente. La evolución síncrona del mercado evita el tratamiento incorrecto de la evolución de las Redes tal como queda indicado en SILV82.

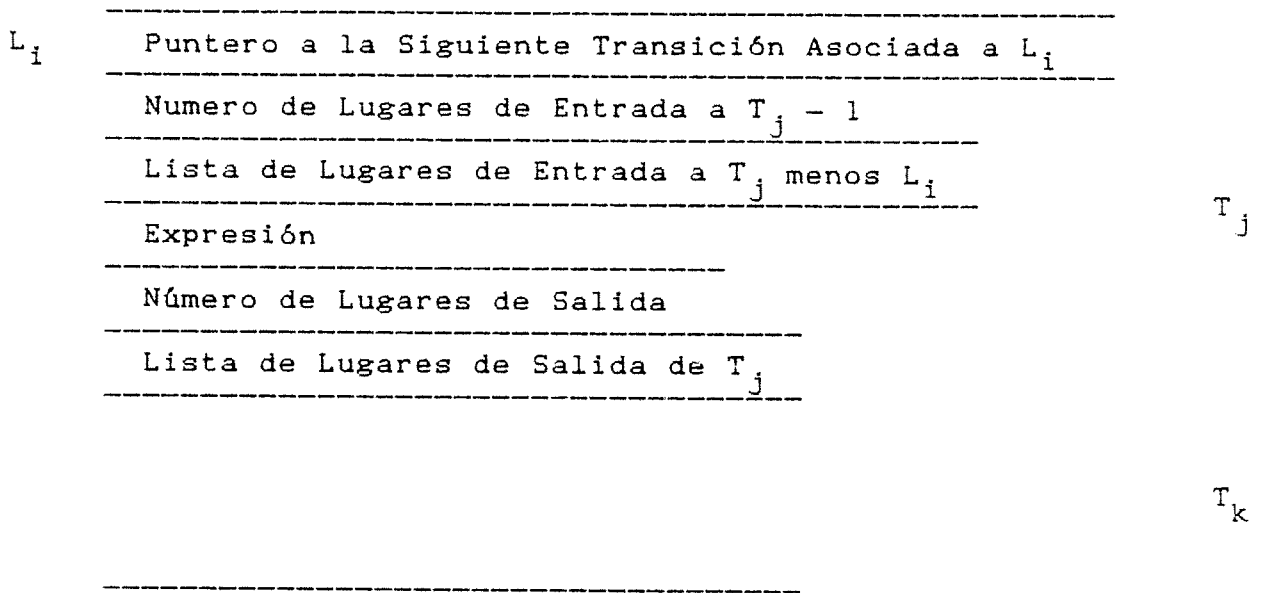


Fig. 8.3 Estructura de Listas para la Representación Interna en Memoria de Redes de Petri.

En este simulador se ha escogido una representación de las Redes en base a una estructura de Lista + Pila tal como queda reflejado en la figura 8.3.

A cada lugar se le asocia uno sólo de sus lugares de entrada que lo representa. Una transición sólo es considerada en un ciclo

cuando su lugar de entrada asociado se encuentra marcado lo que agiliza en gran medida el proceso de simulación.

Las estructuras de Listas representan un ahorro considerable de memoria sobre todo cuando el número de Lugares y Transiciones por Red es muy elevado [SILV82].

Los lugares que resultan marcados al cabo de un ciclo van siendo almacenados en una Pila de donde se extraen en el ciclo siguiente para analizar sus transiciones asociadas.

Por otra parte, es usual tratar las expresiones mediante su conversión a notación polaca y evaluarlas mediante un algoritmo de pilas. Sin embargo, mayor rapidez se consigue interpretando cada una de las expresiones durante el tiempo de procesado de las Redes previo a la simulación, mediante su conversión en árboles de decisiones binarias [SILV78], lo que obliga a eliminar paréntesis negados como se indicó en el apartado 8.3.1 y lo que aconseja, a su vez, a ordenar los términos de las expresiones en orden creciente del número de variables que contienen.

Por último y buscando siempre una gran rapidez en el proceso de simulación, el programa de tratamiento de las redes se ha realizado en lenguaje de bajo nivel (lenguaje ensamblador) sobre un microprocesador dedicado de gran rapidez (8 MHz).

8.5 VALIDACION DEL SIMULADOR Y EJEMPLOS DE SU USO.

Para validar el simulador ofrecemos dos ejemplos de su uso, contrastando los resultados obtenidos con resultados exactos conocidos.

Consideraremos dos casos:

1. Estimación del número medio de memorias ocupadas (Ancho de Banda B) de un sistema crossbar 8×8 en tiempo discreto con $m = 1$ (SISTEMA 1).
2. Estimación del número medio de procesadores ejecutando en memoria privada (Poder de Procesamiento P) en un sistema de 8 procesadores y una única memoria (o bus) común, para $\rho = 0.5$ y $\rho = 1$ (SISTEMA 2).

En ambos casos los procesadores (y las memorias para el sistema crossbar) se consideran indistinguibles.

En el Apéndice I se recoge el listado de ambas aplicaciones. Es de notar el hecho de que el arbitraje se realiza por prioridades sin interrupción en los dos casos considerados por razones de simplicidad, si bien, como ya sabemos, el resultado global es indiferente de la política de arbitraje escogida.

En la tabla 8.1 se muestran los tiempos de procesado y generación de tablas desglosados para cada una de las partes que componen el sistema simulación. Asimismo se muestran los tiempos de

simulación medidos para 26000 ciclos (25000 útiles) en ambos modelos.

	TIEMPOS PROCESAMIENTO (SEG)				TIEMPOS SIMULACION
	PRORED	TRADUC	LINBUS	TRANS.	25000 CICLOS (SEG.)
SISTEMA 1	66	103	97	193	376
SISTEMA 2	32	28	75	136	156

Tabla 8.1

Los tiempos de procesamiento se han medido sobre un PDP 11 que, en el momento de efectuar las pruebas, se encontraba ejecutando exclusivamente los programas respectivos.

En las tablas 8.2 y 7.3 se muestran los intervalos de confianza al 90% obtenidos para simulaciones de 26000 ciclos cada una suprimiendo de cada vez los 1000 primeros ciclos. Para el sistema 1 bastaron 4 simulaciones para alcanzar la máxima amplitud admisible del intervalo. Para el sistema 2 fueron necesarios en cada caso 8 simulaciones.

N	ESTIMADO	INT. 1	INT. 2	EXACTO
2	1.8753	0.0017	0.0019	1.8750
3	2.6262	0.0043	0.0050	2.6272
4	3.2659	0.0077	0.0089	3.2652
5	3.8000	0.0098	0.0114	3.8019
6	4.2479	0.0111	0.0130	4.2518
7	4.6259	0.0128	0.0149	4.6292
8	4.9449	0.0135	0.0158	4.9471

Tabla 8.2 Sistema 1.

RO	ESTIMADO	INT. 1	INT. 2	EXACTO
0.1	6.615	0.040	0.062	6.617
0.5	1.987	0.040	0.062	1.998

Tabla 8.3 Sistema 2.

Para el sistema crossbar 8 x 8 en tiempo discreto estos resultados se comparan con los exactos obtenidos por Bhandarkar para $m = 1$.

Para el sistema con una única memoria común y 8 procesadores en tiempo continuo se han considerado tiempos de ejecución en memoria privada y tiempos de servicio exponencialmente distribuidos y los intervalos de confianza se comparan con los resultados exactos obtenidos por aplicación de la fórmula (4.14).

Los intervalos de confianza se han obtenido en dos formas diferentes. En la primera (denominados Intervalos de Confianza 1) se ha admitido que la variable cuya media se pretende estimar tiene una distribución normal con una desviación típica desconocida [RIOS77] y son, por tanto, intervalos de confianza aproximados.

$$\left(\bar{x} - t_{\alpha} \frac{s}{\sqrt{n-1}}, \bar{x} + t_{\alpha} \frac{s}{\sqrt{n-1}} \right) \quad (8.1)$$

donde \bar{x} es la media estimada, $1 - \alpha$ es el nivel de confianza requerido, s es la desviación típica de las muestras y t_{α} es el valor de la distribución t de Student con $n - 1$ g. l.

En la segunda (denominados Intervalos de Confianza 2) se ha utilizado la conocida fórmula de Tchebysheff [RIOS77] aplicable a cualquier distribución que pudiera tener la variable cuya media se estima considerando como única aproximación que la desviación típica de la variable coincide con la de las muestras.

$$\left(\bar{x} - \frac{s}{\sqrt{\alpha n}}, \bar{x} + \frac{s}{\sqrt{\alpha n}} \right) \quad (8.2)$$

Los intervalos de confianza obtenidos con la segunda aproximación, si bien más amplios, son más exactos que los primeros.

Los resultados de simulación presentados en los Capítulos anteriores de la Tesis se corresponden con el centro de los intervalos de confianza cuyo diámetro (según la aprox. 2) se encuentra, en cualquier caso, dentro del 4% del centro del intervalo.

CONCLUSIONES.

Tomando como origen la necesidad de rapidez de procesamiento y fiabilidad que requieren los sistemas eléctricos de Potencia, en este trabajo se ha realizado un estudio profundo de los principales sistemas de interconexión procesador-memoria en sistemas multiprocesadores, centrándonos en los tres sistemas básicos:

1. Redes Cross-Bar.
2. Redes Multibus. (Como casos particulares, sistemas con un solo bus y Sistemas con buses parciales) y
3. Redes Delta

Se han planteado los modelos de cada uno de estos sistemas estableciendo soluciones analíticas exactas (cuando son abordables) y estudiando y comparando las diferentes aproximaciones propuestas para los casos más complejos.

Como principales aportaciones de esta Tesis caben señalarse:

1. Formulación de nuevas soluciones aproximadas para sistemas multibuses y sistemas con buses parciales en el caso discreto. Estas aproximaciones han probado ser muy seguras y de una gran simplicidad de cálculo.
2. Establecimiento de una comparación entre los diferentes sistemas en cuanto a sus prestaciones, orden de complejidad, fiabilidad y tolerancia a faltas.

3. Estudio del comportamiento del sistema "cross-bar" frente a cargas más reales.
4. Realización de un Sistema de Simulación basado en Redes de Petri adaptado para la simulación de sistemas multi-procesadores. Este Sistema de Simulación representa una aplicación original de las Redes de Petri y una innovación respecto a los simuladores tradicionalmente empleados.

Al ser este un campo en constante evolución una gran variedad de futuras líneas de investigación quedan abiertas, entre las que cabe citar:

1. El estudio y comparación de algoritmos de control, arbitraje y "routing" de la información a través de la red.
2. Formulación de modelos aún más reales que permitan un tratamiento analítico de problemas más cercanos a la realidad.
3. Proposición de nuevas arquitecturas tendentes a obtener mejores prestaciones con órdenes de complejidad bajos.
4. Estudio de Redes Inteligentes autoconfigurables que eviten la presencia de conflictos con el correspondiente estudio de los algoritmos de control que precisarían estas redes.

BIBLIOGRAFIA

- AGRA82 Agrawal D.P. "Testing and Fault Tolerance of Multistage Interconnection Networks". IEEE Computer, April 1982, pag. 41-53.
- AGRA83 Agrawala A.K. and Herzog U. "Performance Evaluation of Multiple Processor Systems". IEEE Trans. on Comp., Vol. C-32, No. 1, pag. 2-3.
- AJM081a Ajmone Marsan M., Conte C., Del Corso D., Gregoretti G. "A Study on Processor/Memory Interconnection in Multiprocessor Systems". Alta Frequenza, Vol. L, No. 3, May-June 1981, pag. 122-130.
- AJM081b Ajmone Marsan M., Gregoretti F. "Memory Interference Models for a Multiprocessor System with a Shared Bus and a Single External Common Memory". Euromicro J., No. 7, Feb. 1981, pag. 124-133.
- AJM082a Ajmone Marsan M. and Gerla M. "Markov Models for Multiple Bus Multiprocessor Systems". IEEE Trans. on Comp., Vol. C-31, No. 3, March 1982, pag. 239-248.
- AJM082b Ajmone Marsan M., Balbo G., Conte C. "Comparative Performance Analysis of Single Bus Multiprocessor Architectures". IEEE Trans. on Comp., vol C-31, No. 12, December 1982, pag. 1179-1191.
- AJM083 Ajmone Marsan M., Balbo G., Conte C., Gregoretti F. "Modeling Bus Contention and Memory Interference in a Multiprocessor System", IEEE Trans. on Comp., Vol. C-32, No. 1, Jan. 1983, pag. 60-71.

- ALLE75 Allen A.O. "Elements of Queueing Theory for System Design". IBM Sys J. No 2, 1975, pag. 161-187.
- ALLE80 Allen A.O. "Queueing Models of Computer Systems". IEEE Computer, Abril 1980, pag. 13-24.
- BARN81 Barnes G.H. and Lundstrom S.F. "Design and Validation of a Connection Network for Many-Processor Systems". IEEE Comp., Vol. 14, No. 12, Dec. 1981, pag. 31-41.
- BASK75 Baskett F., Chandy K.M., Muntz R.R. and Palacios F.G. "Open, Closed and Mixed Networks with Different Classes of Customers". JACM, Vol. 22, No. 2, April 1975, pag. 248-260.
- BASK76 Baskett F. and Smith A.J. "Interference in Multiprocessor Computer Systems with Interleaved Memory". Com. ACM, Vol. 19, No. 6, June 1976, pag. 327-334.
- BHAN73 Bhandarkar D.P. and Fuller S.H. "Markov Chain Models for Analyzing Memory Interference in Multiprocessor Systems". Proc. 1st. Annu. Symp. on Comp. Arch., December 1973, pag. 1-6.
- BHAN75 Bhandarkar D.P. "Analysis of Memory Interference in Multiprocessors". IEEE Trans. on Comp., Vol. C-24, No. 9, September 1975, pag. 897-908.
- BHAN77 Bhandarkar D.P. "Some Performance Issues in Multiprocessor System Design". IEEE Trans. on Comp., May 1977, pag. 506-511.
- BHUY83 Bhuyan L. and Agrawal D.P. "Design and Performance of Generalized Interconnection Networks". IEEE Trans. on Comp., Vol. C-32, No.12 Dec. 1983, pag. 1081-1090.
- BHUY85 Bhuyan L.N. "An Analysis of Processor-Memory Interconnection Networks". IEEE Trans. on Comp., Vol. C-34, No. 3, March 1985, pag. 279-283.

- BORG75 Borgerson B.R. and Freitag R.F. "A Realibility Modelo of Graceful Degradation and Standby-Sparing Systems". IEEE Trans. on Comp., Vol. C-24, May 1975, pag. 517-525.
- BRIG77 Briggs F.A. and Davidson E.S. "Organization of Semiconductor Memories for Parallel-Pipelined Processors". IEEE Trans. on Comp., Vol. C-26, No. 2, Feb. 1977, pag. 162-169.
- BRIG81 Briggs F.A, Dubois M. and Hwang K. "Throughput Analysis and Configuration Design of a Shared-Resource Multiprocessor System: PUMPS". Proc. of the 8th Annu. Symp. on Comput. Arch., 1981, pag. 67-79.
- BRIG83 Briggs F.A. and Dubois M. "Effectiveness of Private Caches in Multiprocessor Systems with Parallel-Pipelined Memories". IEEE Trans. on Comp., Vol. C-32, No. 1, Jan. 1983, pag. 48-59.
- BURN70 Burnett G.J. and Coffman E.G.Jr. "A Study of Interleaved Memory Systems". 1970 Spring Joint Comput. Conf. AFIPS. Proc. Vol 36, Montvale NJ: AFIPS Press, pag. 467-474.
- BURN73 Burnett G.J. and Coffman E.G.Jr. "A Combinatorial Problem Related to Interleaved Systems". JACM, Vol. 20, Jan. 1973, pag. 39-45.
- BURN75 Burnett G.J. and Coffman E.G.Jr. "Analysis of Interleaved Memory Systems Using Blockage Buffers". Comm. ACM, Vol. 19, Feb. 1975, pag. 91-95.
- BUZE73 Buzen J.P. "Computational Algorithms for Closed Queueing Networks with Exponential Servers". Comm. ACM, Vol. 16, No. 9, Sept. 1973, pag. 527-531.
- BUZE77 Buzen J.P. and Poitier D. "Accuracy of Exponential Assumptions in Closed Queueing Models". ACM Sigmetrics /CMG. Int. Conf. on Comput. Perfomance, Modeling, Measurement and Management. Washington, Nov. 1977, pag. 53-64.

- CAMA83 Camacho E.F., Franquelo L.G. and Lozano J. "A Language for Real Time Simulation of Processes with Booleans Inputs and Outputs". Preprints of IMACS Int. Symp. on Simulation in Engineering Sciences. Nantes, France, 1983.
- CALV85 Calvo J.L. et al. "Sistemas de Simulación para la Prueba de Automatismos de Subestaciones Eléctricas". Memoria. Enero 1985.
- CENS78 Censier L.M. and Fautrier P. "A New Solution to Coherent Problems in Multicache Systems". IEEE Trans. on Comp., Vol C-27, Dec. 1978, pag 1112-1118.
- CHAND75 Chandy K.M., Herzog U. and Woo L. "Parametric Analysis of Queueing Networks". IBM J. Res. Dev., Jan. 1975, pag. 36-42.
- CHAND78 Chandy K.M. and Sauer C.H. "Approximate Methods for Analyzing Queueing Network Models of Computing Systems". Computer Surveys, Vol. 10, No. 3, Sept. 1978, pag. 281-317.
- CHANG77 Chang D.Y., Kuck D.J. and Lawrie D.H. "On the Effective Bandwidth of Parallel Memories". IEEE Trans. on Comp., May 1977, pag. 480-490.
- CHEN81 Cheng P-Y., Lawrie D.H., Padua D.A. and Yew P-C. "Interconnection Networks using Shuffles". IEEE Comp. Vol. 14, No. 12, Dec. 1981, pag. 55-64.
- COFF71 Coffman E.G.Jr., Burnett G.J. and Snowdon R.A. "On the Performance of Interleaved Memories with Multiple Word Bandwidths". IEEE Trans. on Comp., Vol. C-20, Dec. 1971, pag. 1566-1569.
- COVO74 Covo A.A. "Analysis of Multiprocessor Control Organization with Partial Program Memory Replication". IEEE Trans. on Comp., Vol. C-23, Feb 1974, pag. 113-120.

- DIAS80 Dias D.M. and Jump J.R. "Packet Communication in Multistage Shuffle-Exchange Networks". Proc. 1980 Int'l Conf. on Parallel Processing. Aug. 1980, pag. 327-328.
- DIAS81a Dias D.M. and Jump J.R. "Analysis and Simulation of Buffered Delta Networks". IEEE Trans. on Comp. Vol. C-30, No. 4, Apr. 1981, pag. 273-282.
- DIAS81b Dias D.M. and Jump J.R. "Packet Switching Interconnection Networks for Modular Systems". IEEE Comp., Vol. 14, No. 12, Dec. 1981, pag. 43-53.
- DU85 Du H.-C. "On the Performance of Synchronous Multiprocessors". IEEE Trans. on Comp., Vol. C-34, No. 5, May 1985, pag. 462-466.
- EMER78 Emer J.S. and Davidson E.S. "Control Store Organization for Multiple Stream Pipelined Processors". Proc. of the 1978 Int. Conf. on Parallel Processing, pag. 43-48.
- FELL66 Feller W. "An Introduction to Probability Theory and its Applications". Vol II, Wiley, 1966.
- FENG81 Feng T.Y. "A Survey of Interconnection Networks". IEEE Computer, Dec.1981, pp. 12-27.
- FLYN72 Flynn M.J. "Some Computer Organizations and Their Effectiveness". IEEE Trans. on Comp., Vol. C-21, No. 9, Sept. 1972, pag. 948-960.
- FOST74 Foster D.V., Mcgearty P.F., Sauer C.H. and Waggoner C.N. "A Language for Analysis of Queueing Models". Proc. Fifth Annual Pittsburg Modeling and Simulation Conf., 1974, pag. 381-386.
- FRAN81 Franklin M.A. "VLSI Performance Comparison of Banyan and Crossbar Communications Networks". IEEE Trans. on Comp., Vol. C-30. No. 4, Apr. 1981, pag. 283-291.

- FROM83 Fromm H., Hercksen U., Herzog U., John K.-H., Klar R. and Kleinoder W. "Experiences with Performance Measurement and Modeling of a Processor Array". IEEE Trans. on Comp., Vol. C-32, No. 1, Jan. 1983, pag. 15-31.
- FUNG79 Fung K. T. and Torng H.C. "On the Analysis of Memory Conflicts and Bus Contentions in a Multiple-Microprocessor System". IEEE Trans. on Comp., Vol. C-27, No. 1, Jan. 1979, pag. 28-37.
- GOKE73 Goke L.R. and Lipovski. "Banyan Networks for Partitioning Multiprocessing Systems". Proc. First Ann. Comp. Arch. Conf., Dec.1973, pp.21-28.
- HOOG77 Hoogendoorn C.H. "A General Model for Memory Interference in Multiprocessors". IEEE Trans. on Comp., Vol. C-26, No. 10, October 1977, pag. 998-1005.
- HOR081 Horowitz E. and Zorat A. "The Binary Tree as an Interconnection Network: Applications to Multiprocessor Systems and VLSI". IEEE Trans. on Comp., Vol. C-30, No. 4, Apr. 1981, pag. 247-253.
- JACK63 Jackson, J.R. "Joshop-Like Queueing Systems". Management Science, 10, 1963, pag. 131-142.
- JAC082 Jacobson P.A. and Lazowska E.D. "Analyzing Queueing Networks with Simultaneous Resource Possession". Comm. ACM, Vol. 25, No. 2, Feb. 1982, pag. 142-151.
- JAIS68 Jaiswal N.K. "Priority Queues". Academic Press. New York, 1968.
- KAPL73 Kaplan K.R. and Winder R.O. "Cache Based Computer Systems". Computer, Mar. 1973, pag. 30-36.
- KINN78 Kinney L.L. and Arnold R.G. "Analysis of a Multiprocessor System with a Shared Bus". Proc. of the 5th. Annu. Symp. on Comp. Arch., Vol.6. No. 7, Apr. 1978, pag. 89-95.

- KLEI75 Kleinrock L. "Queueing Systems". Vol. I "Preliminaries". John Wiley & Sons. New York 1975.
- KLEI76 Kleinrock L. "Queueing Systems". Vol. II "Computer Applications".
- KOBA74 Kobayashi H. "Application of the Diffusion Approximation to Queueing Networks: Part I - Equilibrium Queue Distributions". JACM, Vol. 21, 1974, pag. 316-328.
- KRIZ83 Kriz J. "A Queueing Analysis of a Symmetric Multiprocessor with Shared Memories and Buses". IEE Proc., Vol. 130, Pt. E, No. 3, May 1983, pag. 83-89.
- KRUS83 Kruskal, C.P. and Snir M. "The Performance of Multistage Interconnection Networks for Multiprocessors". IEEE Trans. on Comp., Vol. C-32, No. 12, Dec. 1983, pag. 1091-1098.
- KURT74 Kurtzberg J.M. "On the Memory Conflict Problem in Multiprocessor Systems". IEEE Trans. on Comput., Vol. C-23, No. 3, March 1974, pag. 286-293.
- LANG76 Lang T. and Stone H.S. "A Shuffle-Exchange Network with Simplified Control". IEEE Trans. on Comp., Vol. C-25, No. 6, Jan. 1976, pag. 55-65.
- LANG82 Lang. T., Valero M., Alegre I. "Bandwidth of Crossbar and Multiple-Bus Connections for Multiprocessors". IEEE Trans. on Comp., Vol. C-31, No. 12, Dec. 1982, pag. 1277-1234.
- LANG83 Lang T., Valero M. and Fiol M. "Reduction of Connections for Multibus Organization", IEEE Trans. on Comp., Vol. C-32, No. 8, August 1983, pag. 707-716.
- LAVE75 Lavenberg S.S and Shutz D.R. "Introduction to Regenerative Simulation". IBM J. Res. Dev., 19, Sept. 1975, pag. 458-463.

- LAVE77 Lavenberg S.S. and Sauer C.H. "Sequential Stopping Rules for the Regenerative Method of Simulation". IBM J. Res. Dev., 21, Nov. 1977, pag. 545-558.
- LAWR75 Lawrie D.H. "Access and Alignment of Data in an Array Processor". IEEE Trans. on Comp., Vol. C-24, Dec. 1975, pag. 1145-1155.
- LEHM80 Lehmann A. "Performance Evaluation and Prediction of Storage Hierarchies". ACM Sigmetrics Performance'80, Vol. 9, May 1980, pag. 43-54.
- LENA80 Lenahan J.J. and Fung F.K. "Performance of Cooperative Loosely Coupled Microprocessor Architectures in an Interactive Data Base Task". IEEE Trans. on Comp., Vol. C-29, No. 2, Feb. 1980, pag. 161-180.
- LENF78 Lenfant J. "Parallel Permutations of Data: A Benes Network Control Algorithm for Frequently Used Permutations". IEEE Trans. on Comp., Vol. C-27, No. 7, July 1978, pag. 637-647.
- MARK84 Markenscoff P. "A Deterministic Model for Evaluating the Performance of a Multiple Processor System with a Shared Bus". IEEE Trans. on Comp., Vol. C-33, No. 3, March 1984, pag. 281-285.
- MAS079 Mason G.M., Gingher C. and Nakamura S. "A Sampler of Circuit Switching Networks". IEEE Computer, Vol. 12, No. 6, June 1979, pag. 32-48.
- McD082 McDonald W.C. and Smith R.W. "A Flexible Distributed Testbed for Real-Time Applications". IEEE Computer, Vol. 15, No. 10, 1982, pag. 25-39.
- MEHR80 Mehra S.K. and Majithia J.C. "A Comparative Study of Some Two-Processor Organizations". IEEE Trans. on Comp., Vol. C-29, No. 1, Jan. 1980, pag. 44-49.

- MUDG82 Mudge T.N. and Makrucki B.A. "Probabilistic Analysis of a Crossbar Switch". Proc. 9th. Annual Symp. on Comp. Arch., April 1982, pag. 311-320.
- NASS81 Nassimi D. and Sahni S. "A Self-Routing Benes Network and Parallel Permutation Algorithms". IEEE Trans. on Comp., Vol. C-30, No. 5, May 1981, pag. 332-340.
- NUTT77 Nutt G.J. "Memory and Bus Conflict in an Array Processor". IEEE Trans. on Comp., Vol. C-26, No. 6, June 1977, pag. 514-521.
- PADM83 Padmanabhan K. and Lawrie D.H. "A Class of Redundant Path Multistage Interconnection Networks". IEEE Trans. on Comp., Vol. C-32, No. 12, Dec. 1983, pag. 1099-1108.
- PALM58 Palm C. "The Assignment of Workers in Service Machine". J. Indust. Eng., Sept. 1958.
- PARK83 Parkinson D. and Liddell H.M. "The Measurement of Performance on a Highly Parallel System". IEEE Trans. on Comp., Vol. C-32, No. 1, Jan. 1983, pag. 32-37.
- PATE79 Patel J.H. "Processor-Memory Interconnections for Multiprocessors". Proc. 6th Annu. Symp. on Comp. Arch., April 1979, pag. 168-177.
- PATE81 Patel J.H. "Performance of Processor-Memory Interconnections for Multiprocessors". IEEE Trans. on Comp., Vol. C-30, No. 10, Oct. 1981, pag. 771-780.
- PATE82 Patel J.H. "Analysis of Multiprocessors with Private Cache Memories". IEEE Trans. on Comp., Vol. C-31, No. 4, April 1982, pag. 296-304.
- PEAS77 Pease M.C. "The Indirect Binary n-Cube Microprocessor Array". IEEE Trans. on Comp., Vol. C-26, No. 5, May 1977, pag. 548-573.

- PETE81 Peterson J.L. "Petri Nets Theory and the Modelling of Systems". Prentice Hall, Englewood Cliffs, N.J., 1981.
- PETE61 Peterson W. and Weldon E. "Error Correcting Codes". Cambridge MA. MIT Press, 1961.
- PEUT77 Peuto B.L. and Shustek L.J. "An Instruction Timing Model of CPU Performance". Proc. of 4th. Symp. on Comp. Arch., 1977, pag. 165-178.
- PROT80 Protopapas D.A., Smith E.J. "Modeling and Analysis of Single and Multiple-Bus Multi-Microcomputer Systems". Proc. Distributed Computing, COMCOM 80, 21st IEEE COM.SOC. Int. Conf., Washinton D.C., Sept. 1980, New York.
- RAO78 Rao G.S. "Performance Analysis of Cache Memories". JACM, Vol. 25, Jul. 1978, pag. 378-395.
- RAU79 Rau B.R. "Interleaved Memory Bandwith in a Model of a Multiprocessor Computer System". IEEE Trans. on Comp., Vol. C-28, No.9, September 1979, pag. 678-681.
- RAVI72 Ravi, C.V. "On the Bandwidth and Intereference in Interleaved Memory Sytems". IEEE Trans. on Comp., Vol. C-21, No. 8, Aug. 1972, pag. 899-901.
- REED83 Reed D.A. and Scwetman H.D. "Cost-Performance Bounds for Multimicrocomputer Networks". IEEE Trans. on Comp., Vol. C-32, No. 1, Jan. 1983, pag. 83-95.
- REGI73 Regis R.C. "Multiserver Queueing Models of Multiprocessing Systems". IEEE Trans. on Comp., Vol. C-22, No. 8, August 1973, pag. 736-744.
- REIS76 Reiser M. "Interactive Modeling of Computer Systems". IBM Syst. J., No. 4, 1976, pag. 309-327.

- REIS78 Reiser M. and Sauer C.H. "Queueing Network Models: Methods of Solution and their Program Implementations". Current trends in programming Methodology, Vol. III: Software Modeling and its Impact on Performance". K.M. Chandy and R.T. Yen (Eds.). Prentice-Hall Inc., Englewood Cliffs, NJ 1978, pag. 115-167.
- RIOS77 Ríos S. "Métodos Estadísticos". Ed. Castillo, Madrid 1977.
- SAST75 Sastry K.V., Kain R.Y. "On the Performance of Certain Multiprocessor Computer Organitations". IEEE Trans. on Comp., Vol. C-24, No. 11, Nov. 1975, pag. 1066-1074.
- SCHN78 Schneider G.M. "A Modeling Package for Simulation of Computer Networks". Simulation, Vol. 31, No. 6, Dec. 1978, pag. 181-192.
- SETH79 Sethi A.S., Deo N. "Interference in Multiprocessor Systems with Localized Memory Access Probabilities". IEEE Trans. on Comp., Vol. C-28, No. 2, February 1979, pag. 157-163.
- SIEG79 Siegel H.J. "Interconnection Networks for SIMD Machines". IEEE Computer, Vol.12, No. 6, June 1979, pag. 57-65.
- SIEG80 Siegel H.J. "The Theory Underlying the Partitioning of Permutation Networks". IEEE Trans. on Comp., Vol. C-29, No.9, Sept. 1980, pag. 791-801.
- SIEG81a Siegel H.J. and McMillen R.J. "Using the Augmented Data Manipulator Network in PASM". IEEE Computer, Vol. 14, No. 2, Feb. 1981, pag.23-33.
- SIEG81b Siegel H.J. "Interconnection Networks for Parallel and Distributed Processing: An Overview". IEEE Trans. on Comp., Vol. C-30, No. 4, Apr. 1981, pag. 245-246.

- SIEG81c Siegel H.J. and McMillen R.J. "The Multistage Cube: A Versatile Interconnection Network". IEEE Comp., Vol. 14, No. 12, Dec. 1981, pag. 65-76.
- SIOM83 Siomalas K.O., Bowen B.A. "Performance of Cross-Bar Multiprocessor Systems". IEEE Trans. on Comp., Vol. C-32, No. 7, Jul. 1983, pag. 689-695.
- SILV78 Silva M. "Contribution a la Synthese Programmé des Automatismes Logiques". These Docteur-Ingenieur. I.N.P. Grenoble, Juin, 1978.
- SILV82 Silva M. "Programable Logic Controllers and Petri Nets: A Comparative Study".
- SILV84 Silva M. "Las Redes de Petri en la Automática y la Industria". ED. AC, Madrid, 1984.
- SKIN69 Skinner C.E. and Asher J.R. "Effects of Storage Contention on System Performance". IBM Syst. J., Vol. 8, 1969, pag. 319-333.
- SMIT77 Smith A.J. "Multiprocessor Memory Organization and Memory Interference". Comm. ACM, Vol. 20, No. 10, Oct. 1977, pag. 754-761.
- STON71 Stone H.S. "Parallel Processing with the Perfect Shuffle". IEEE Trans. on Comp., Vol. C-20, Feb. 1971, pag. 153-161.
- STRE70 Strecker W.D. "Analysis of the Instruction Execution Rate in Certain Computer Structures". Ph. D. Dissertation, Carnegie-Mellon Univ., Pittsburg, Pa.
- THAN81 Thanawatien S. and Nelson V.P. "Interference Analysis of Shuffle-Exchange Networks". IEEE Trans. on Comp., Vol. C-30, No. 8, Aug. 1981, pag. 545-555.

- THOM78 Thompson C.D. "Generalized Connection Networks for Parallel Processor Intercommunication". IEEE Trans. on Comp., Vol. C-27, No. 12, Dec. 1978, pag. 1119-1125.
- TORR84 Torralba A., Franquelo L.G. y Serrano J.C. "Sistema de Creación de Gráficos y Definición de Modelos en Subestaciones Eléctricas". V Reunión del G.E.M., Sevilla 27-28, Sept. 1984, Servicio de Publicaciones de la Univ. de Sevilla, pag. 512-527.
- TRIV82 Trivedi K.S. "Probability & Statistics with Reliability, Queueing and Computer Science Applications". Prentice-Hall Inc., Englewood Cliffs, NJ 1982.
- VANT78 Vantilborgh H. "Exact Aggregation in Exponential Queueing Networks". JACM, Vol. 25, No. 4, Oct. 1978, pag. 620-629.
- WAH84 Wah B.W. "A Comparative Study of Distributed Resource Sharing on Multiprocessors". IEEE Trans. on Comp., Vol. C-33, No. 8, August 1984, pag. 700-711.
- WILL78 Willis P.J. "Derivation and Comparison of Multiprocessor Contention Measures". Computers Digital Techniques, Vol 1, No. 3, August 1978, pag. 94-98.
- WIRS81 Wirsching J.E. and T. Kishi. "CONET: A Connection Network Model". IEEE Trans. on Comp., Vol. C-30, No. 4, pag. 298-301.
- WITT81 Wittie L.D. "Communication Structures for Large Networks of Microcomputers". IEEE Trans. on Comp., Vol. C-30, No. 4, pag. 264-273.
- WU80a Wu C. and Feng T. "On a Class of Multistage Interconnection Networks". IEEE Trans. on Comp., Vol. C-29, No. 8, Aug. 1980, pag. 694-702.

- WU80b Wu C. and Feng T. "The Reverse-Exchange Interconnection Network". IEEE Trans. on Comp., Vol. C-29, No. 9, Sept. 1980, pag. 801-811.
- WU81a Wu C. and Feng T. "Universality on the Shuffle-Exchange Network". IEEE Trans. on Comp., Vol. C-30, No. 5, May 1981.
- WU81b Wu S.B. and Liu T. "A Cluster Structure as an Interconnection Network for Large Multimicrocomputer Systems". IEEE Trans. on Comp., Vol. C-30, No. 4, Apr. 1981, pag. 254-264.
- WU81c Wu C. "Interconnection Networks". IEEE Computer, Vol. 14, No. 12, Dec. 1981, pag. 8-9.
- WULF72 Wulf W. and Bell C.G. "C.mmp-A Multi-mini-processor". Proc. AFIPS 1972 Fall Joint Comput. Conf., Vol. 41, Part II, pag. 765-777.
- YEH83 Yeh P.C.C, Patel J.H. and Davidson E.S. "Shared-Cache for Multiple-Stream Computer Systems". IEEE Trans. on Comp., Vol. C-32, No. 1, pag. 38-47.
- YEN80 Yen W.C., Fu K.S. "Performance Analysis on Multiprocessor Memory Organizations". Proc. ACM Pacific'80 Conf. on Distributed Processing, Nov. 1980, pag. 142-153.
- YEN82 Yen D.W.L., Patel H. and Davidson E.S. "Memory Interference in Synchronous Multiprocessor Systems". IEEE Trans. on Comp., Vol. C-31, No.11, Nov. 1982, pag. 1116-1121.

```

0001 ;
0002 ; *****
0003 ; *** 8 PROCESADORES * 1 UNICO BUS GLOBAL ***
0004 ; *****
0005 ;
0006 ;     MACRO PROCESADOR
0007 ;
0008 .MACRO PROC P1ACT,R1,GB1,LAMBDA,MU,NUM,PEP
0009 RED PEP
0010 TR 1 DE 1 A 2,3
0011 TR 2 DE 2,3 A 4 EX 'GB1
0012 TR 3 DE 2,3 A 5,6 EX GB1
0013 TR 4 DE 4 A 5,6 EX GB1
0014 TR 5 DE 5,6 A 2,3
0015 LU 1 M 1
0016 LU 2 T D LAMBDA R1
0017 LU 3 P1ACT
0018 LU 4 R1
0019 LU 5 'R1,BUSOCU
0020 LU 6 T D MU 'OB
0021 FINRED
0022 .ENDM
0023 ;
0024 ;     PROGRAMA
0025 ;
0026 ;     DEFINICION DE VARIABLES
0027 ;     VARIABLES ACUMULATIVAS DE PROCESADORES ACTIVAS
0028 ;
0029 P1ACT=E+
0030 P2ACT=E+
0031 P3ACT=E+
0032 P4ACT=E+
0033 P5ACT=E+
0034 P6ACT=E+
0035 P7ACT=E+
0036 P8ACT=E+
0037 ;
0038 ;     VARIABLE ACUMULATIVA DE BUS OCUPADO
0039 ;
0040 BUSOCU=E+
0041 ;
0042 ;     VARIABLES DE PROCESADOR EN COLA
0043 ;
0044 R1=T+
0045 R2=T+
0046 R3=T+
0047 R4=T+
0048 R5=T+
0049 R6=T+
0050 R7=T+
0051 R8=T+
0052 ;
0053 ;     VARIABLES INTERNAS DE BUS CONCEDIDO
0054 ;
0055 GB1=T+
0056 GB2=T+

```



```
0057 GB3=T+
0058 GB4=T+
0059 GB5=T+
0060 GB6=T+
0061 GB7=T+
0062 GB8=T+
0063 ;
0064 ; VARIABLE DE BUS OCUPADO
0065 ;
0066 OB=T+
0067 ;
0068 ; PROCESADOR 1
0069 ;
0070 PROC P1ACT,R1,GB1,40,4,1,PP1
0071 ;
0072 ; PROCESADOR 2
0073 ;
0074 PROC P2ACT,R2,GB2,40,4,2,PP2
0075 ;
0076 ; PROCESADOR 3
0077 ;
0078 PROC P3ACT,R3,GB3,40,4,3,PP3
0079 ;
0080 ; PROCESADOR 4
0081 ;
0082 PROC P4ACT,R4,GB4,40,4,4,PP4
0083 ;
0084 ; PROCESADOR 5
0085 ;
0086 PROC P5ACT,R5,GB5,40,4,5,PP5
0087 ;
0088 ; PROCESADOR 6
0089 ;
0090 PROC P6ACT,R6,GB6,40,4,6,PP6
0091 ;
0092 ; PROCESADOR 7
0093 ;
0094 PROC P7ACT,R7,GB7,40,4,7,PP7
0095 ;
0096 ; PROCESADOR 8
0097 ;
0098 PROC P8ACT,R8,GB8,40,4,8,PP8
0099 ;
0100 ; ARBITRAJE
0101 ;
0102 FS GB1=R1*'OB
0103 FS GB2=R2*'R1*'OB
0104 FS GB3=R3*'R2*'R1*'OB
0105 FS GB4=R4*'R3*'R2*'R1*'OB
0106 FS GB5=R5*'R4*'R3*'R2*'R1*'OB
0107 FS GB6=R6*'R5*'R4*'R3*'R2*'R1*'OB
0108 FS GB7=R7*'R6*'R5*'R4*'R3*'R2*'R1*'OB
0109 FS GB8=R8*'R7*'R6*'R5*'R4*'R3*'R2*'R1*'OB
0110 FS OB=OB+GB1+GB2+GB3+GB4+GB5+GB6+GB7+GB8
0111 FIN
0112 END
```

```

0001 ;
0002 ; *****
0003 ; ** SISTEMA CROSS-BAR 8 X 8 **
0004 ; *****
0005 ;
0006 NUM E 1 40
0007 NUM E 2 0
0008 NUM E 3 0
0009 NUM E 4 0
0010 NUM S 1 10
0011 NUM S 2 0
0012 NUM S 3 0
0013 NUM S 4 0
0014 PAGINA=0
0015 CELDA=0
0016 X=0
0017 Y=0
0018 ;
0019 ; MACRO DEL PROCESADOR
0020 ;
0021 .MACRO PROC R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,A1
CT,REQ, /
0022 NOM,N1,N2,N3,N4,N5,N6,N7
0023 RED NOM
0024 &S1=S N1
0025 &S2=S N2
0026 &S3=S N3
0027 &S4=S N4
0028 &S5=S N5
0029 &S6=S N6
0030 &S7=S N7
0031 R1=T+
0032 R2=T+
0033 R3=T+
0034 R4=T+
0035 R5=T+
0036 R6=T+
0037 R7=T+
0038 R8=T+
0039 &RR=T+
0040 REQ=T+
0041 A1=T+
0042 PACT=E+ 1
0043 TR 1 DE 1 A 2 EX 'A1
0044 TR 2 DE 1 A 3 EX A1
0045 TR 3 DE 2 A 3 EX A1
0046 TR 4 DE 3 A 2 EX 'A1
0047 LU 1 M 1 'REQ
0048 LU 2 REQ
0049 LU 3 'REQ,ACC,PACT
0050 FS R1=REQ*R1+'REQ*&S1*(&S2+&S3*(&S4+&S5))
0051 FS R2=REQ*R2+'REQ*&S1*(&S2*(&S3*(&S4*(&S5+'&S3*&S4)
0052 FS R3=REQ*R3+'REQ*((&S1*(&S2*(&S3*(&S4+'&S1*&S2*&S3*&S4*&S5)
0053 FS R4=REQ*R4+'REQ*'&S1*&S2*&S3*(('&S4+'&S5)
0054 FS R5=REQ*R5+'REQ*'&S1*&S2*'&S3*((&S4+&S5)
0055 FS R6=REQ*R6+'REQ*'&S1*((&S2*'&S3*'&S4*'&S5+'&S2*&S3*&S4)
0056 FS R7=REQ*R7+'REQ*'&S1*(&S2*((&S3*'&S4+'&S3*&S4*&S5)

```

```
0057 FS R8=REQ*R8+'REQ*'&&S1*'&&S2*'&&S3*('&S4+'&S5)
0058 FINRED
0059 .ENDM
0060 ACC=E+ 1
0061 ;
0062 ; PROCESADORES
0063 ;
0064 PROC RA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,RA9,RA10,RA11,RA12,RA13,RA14,/
0065 RA15,RA16,RA, PRACT,RERA,NRA,1,2,3,4,5,6,7
0066 PROC RB1,RB2,RB3,RB4,RB5,RB6,RB7,RB8,RB9,RB10,RB11,RB12,RB13,RB14,/
0067 RB15,RB16,RB, PRBCT,RERB,NRB,33,34,35,36,37,38,39
0068 PROC RC1,RC2,RC3,RC4,RC5,RC6,RC7,RC8,RC9,RC10,RC11,RC12,RC13,RC14,/
0069 RC15,RC16,RC, PRCCT,RERC,NRC,65,66,67,68,69,70,71
0070 PROC RD1,RD2,RD3,RD4,RD5,RD6,RD7,RD8,RD9,RD10,RD11,RD12,RD13,RD14,/
0071 RD15,RD16,RD, PRDCT,RERD,NRD,97,98,99,100,101,102,103
0072 PROC RE1,RE2,RE3,RE4,RE5,RE6,RE7,RE8,RE9,RE10,RE11,RE12,RE13,RE14,/
0073 RE15,RE16,RE, PRECT,RERE,NRE,129,130,131,132,133,134,135
0074 PROC RF1,RF2,RF3,RF4,RF5,RF6,RF7,RF8,RF9,RF10,RF11,RF12,RF13,RF14,/
0075 RF15,RF16,RF, PRFCT,RERF,NRF,161,162,163,164,165,166,167
0076 PROC RG1,RG2,RG3,RG4,RG5,RG6,RG7,RG8,RG9,RG10,RG11,RG12,RG13,RG14,/
0077 RG15,RG16,RG, PRGCT,RERG,NRG,193,194,195,196,197,198,199
0078 PROC RH1,RH2,RH3,RH4,RH5,RH6,RH7,RH8,RH9,RH10,RH11,RH12,RH13,RH14,/
0079 RH15,RH16,RH, PRHCT,RERH,NRH,225,226,227,228,229,230,231
0080 ;
0081 ; ARBITRAJE
0082 ;
0083 M1=T+
0084 M2=T+
0085 M3=T+
0086 M4=T+
0087 M5=T+
0088 M6=T+
0089 M7=T+
0090 M8=T+
0091 FS RA=RA1+RA2+RA3+RA4+RA5+RA6+RA7+RA8
0092 FS RB=RB1*'RA1+RB2*'RA2+RB3*'RA3+RB4*'RA4+RB5*'RA5+RB6*'RA6+RB7*'RA7+RB8
0093 FS RC=RC1*'RA1*'RB1+RC2*'RA2*'RB2+RC3*'RA3*'RB3+RC4*'RA4*'RB4+RC5*'RA5
RB5
0094 FS RC=RC+RC6*'RA6*'RB6+RC7*'RA7*'RB7+RC8*'RA8*'RB8
0095 FS M1=RA1+RB1+RC1
0096 FS M2=RA2+RB2+RC2
0097 FS M3=RA3+RB3+RC3
0098 FS M4=RA4+RB4+RC4
0099 FS M5=RA5+RB5+RC5
0100 FS M6=RA6+RB6+RC6
0101 FS M7=RA7+RB7+RC7
0102 FS M8=RA8+RB8+RC8
0103 FS RD=RD1*'M1+RD2*'M2+RD3*'M3+RD4*'M4+RD5*'M5+RD6*'M6+RD7*'M7+RD8*'M8
0104 FS M1=M1+RD1
0105 FS M2=M2+RD2
0106 FS M3=M3+RD3
0107 FS M4=M4+RD4
0108 FS M5=M5+RD5
0109 FS M6=M6+RD6
0110 FS M7=M7+RD7
0111 FS M8=M8+RD8
0112 FS RE=RE1*'M1+RE2*'M2+RE3*'M3+RE4*'M4+RE5*'M5+RE6*'M6+RE7*'M7+RE8*'M8
```

0113 FS M1=M1+RE1
0114 FS M2=M2+RE2
0115 FS M3=M3+RE3
0116 FS M4=M4+RE4
0117 FS M5=M5+RE5
0118 FS M6=M6+RE6
0119 FS M7=M7+RE7
0120 FS M8=M8+RE8
0121 FS RF=RF1*'M1+RF2*'M2+RF3*'M3+RF4*'M4+RF5*'M5+RF6*'M6+RF7*'M7+RF8*'M8
0122 FS M1=M1+RF1
0123 FS M2=M2+RF2
0124 FS M3=M3+RF3
0125 FS M4=M4+RF4
0126 FS M5=M5+RF5
0127 FS M6=M6+RF6
0128 FS M7=M7+RF7
0129 FS M8=M8+RF8
0130 FS RG=RG1*'M1+RG2*'M2+RG3*'M3+RG4*'M4+RG5*'M5+RG6*'M6+RG7*'M7+RG8*'M8
0131 FS M1=M1+RG1
0132 FS M2=M2+RG2
0133 FS M3=M3+RG3
0134 FS M4=M4+RG4
0135 FS M5=M5+RG5
0136 FS M6=M6+RG6
0137 FS M7=M7+RG7
0138 FS M8=M8+RG8
0139 FS RH=RH1*'M1+RH2*'M2+RH3*'M3+RH4*'M4+RH5*'M5+RH6*'M6+RH7*'M7+RH8*'M8
0140 ;
0141 ; UNA SENAL DE PANTALLA
0142 ;
0143 TA,TC,PA,PC=1,1 32,A,32,A,PAN
0144 FS PC=TC
0145 FS PA=TA
0146 FIN
0147 END