
SIMULACIÓN DE UNA RED DE INTERCONEXIÓN DINÁMICA OMEGA PARA ESQUEMAS DE PROCESAMIENTO PARALELO

Israel Rivera Zárate

irivera@ipn.mx

Miguel Hernández Bolaños

mbolanos@ipn.mx

Patricia Pérez Romero

promerop@ipn.mx

INSTITUTO POLITECNICO NACIONAL-CIDETEC

I. Introducción

Los sistemas de cómputo han presentado una constante evolución de su rendimiento desde la propuesta original de Von Newman en 1945 con una sola unidad de procesamiento y una sola unidad de almacenamiento interconectadas por bus común. Ahora sabemos que esta disposición primaria es denominada máquina SISD (flujo único de instrucciones y flujo único de datos) según la topología definida por Michael Flynn en la década de los setenta[1]; en atención a las fuentes que proveen de instrucciones y datos al sistema. Posteriormente como parte de la búsqueda de un mayor poder de cómputo surge el procesamiento paralelo en el que intervienen un número mayor de unidades de procesamiento y almacenamiento y de buses de interconexión. Según Flynn estas nuevas arquitecturas serían denominadas como máquinas MIMD (múltiple flujo de instrucciones y múltiple flujo de datos) o bien máquinas SIMD (flujo único de instrucciones y flujo múltiple de datos) donde se observan arreglos de procesadores que son capaces de ejecutar una misma instrucción o bien instrucciones diferentes sobre diferentes conjuntos de datos que han sido almacenados ya sea en una memoria local a cada procesador o en un sistema de almacenamiento distribuido entre los diferentes elementos de procesamiento o nodos.

En este punto como es posible apreciar se destaca la importancia de la red de interconexión ya que juega un papel principal en el rendimiento del sistema de cómputo cuanto mayor sea el número de elementos de procesamiento y de almacenamiento que enlaza debido a que establece un compromiso de desempeño en la tasa de comunicación[2]. De acuerdo con [3] las redes de interconexión más comunes en la clasificación de Flynn de máquinas MIMD a las que pertenecen los sistemas multiprocesadores y multicomputadoras se denominan redes estáticas y dinámicas.

Redes estáticas y dinámicas

De acuerdo con [3] y [4] una red estática es la constituida por la interconexión de elementos o nodos cuya configuración o topología queda estable y sin cambio alguno. Esta clasificación se aplica principalmente en la arquitectura de sistema de multicomputadoras; donde predomina el paso de mensajes entre procesadores motivo por el cual se le conoce adicionalmente como red de acoplamiento débil. Ejemplos de topología de redes estáticas aparece ilustrado en la figura 1.

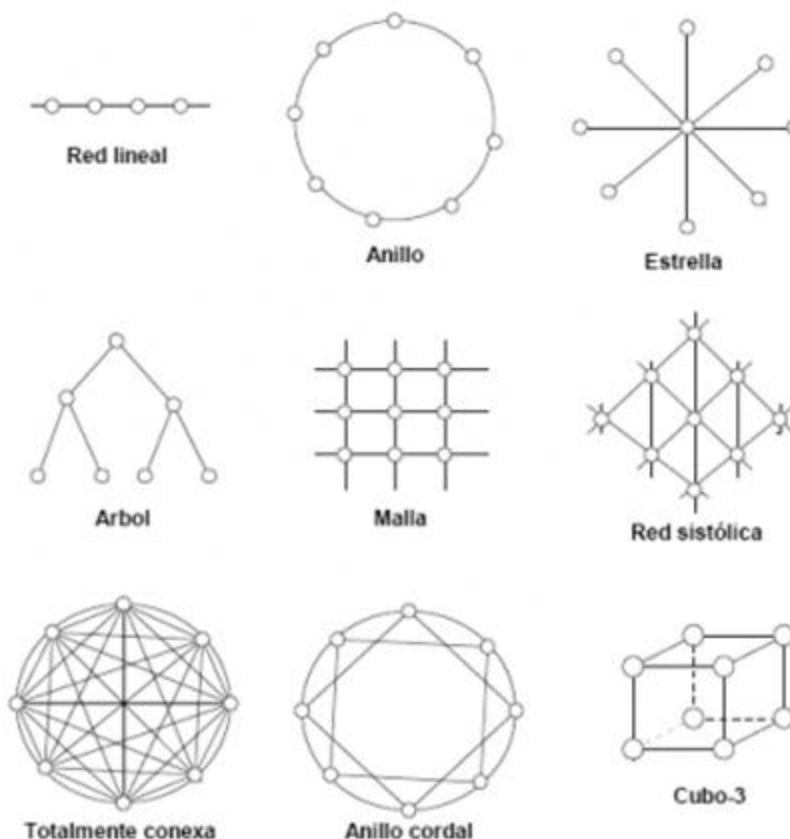


Figura 1. Redes estáticas.

Por otra parte, de acuerdo con [5] una red dinámica es una red cuya disposición o configuración puede cambiar durante la ejecución de un programa paralelo o bien entre dos o más ejecuciones de programas. En el caso particular la red está formada por elementos de conmutación llamados típicamente interruptores o conmutadores. Este tipo de redes dinámicas se utilizan sobre todo en arquitecturas de sistemas multiprocesadores. Ejemplo de topología de redes dinámicas se presenta en la figura 2. Los conmutadores aparecen ilustrados como recuadros que cabe mencionar permiten el enrutamiento adecuado de los datos mediante la codificación de palabras de control.

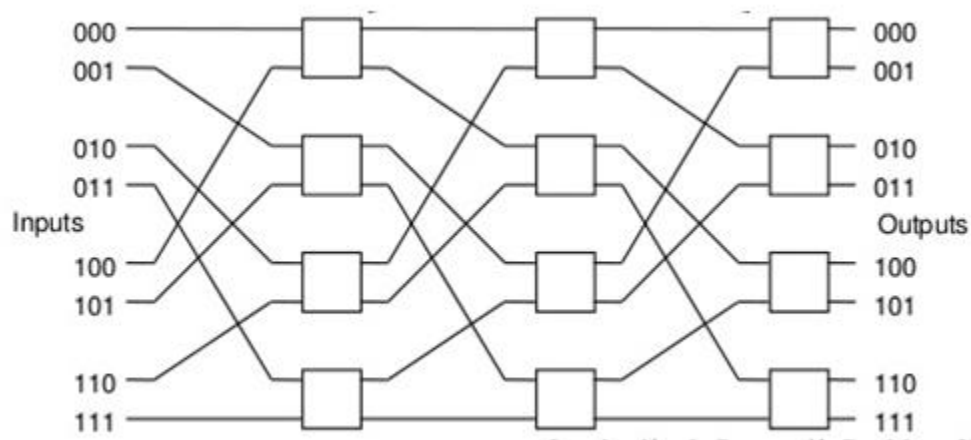


Figura 2. Red dinámica multietapa.

La red Omega

La red Omega es un caso de red dinámica ampliamente usado en la práctica y se basa en que cada conmutador está configurado para realizar la función de *Exchange* (Cambio) que consiste en invertir el bit menos significativo de la palabra de control [7] y[8]. A su vez y de manera inicial los conmutadores están conectados entre ellos en modo *Shuffle*, lo que produce el corrimiento o desplazamiento de un bit a la izquierda. Por su parte, el algoritmo de enrutamiento es el siguiente: en la etapa i , se deben comparar los bits i -ésimos (desde los más significativos hacia los menos significativos) del elemento origen y del elemento destino en donde ocurre que si son iguales, no se deben cruzar; o bien en caso contrario, se deben cruzar (*Exchange*).

II. Desarrollo

Para simular el funcionamiento y el camino que se sigue para comunicar un procesador con una memoria, el esquema es poder comunicar 8 procesadores con 8 unidades de memoria, mediante la red dinámica de interconexión multietapa OMEGA.

Las redes de interconexión dinámicas son convenientes en los casos en que se desee una red de propósito general ya que son fácilmente reconfigurables. También por eso, este tipo de redes facilitan mucho la escalabilidad. En general, las redes dinámicas necesitan de elementos de conexión específicos como puede ser árbitros de bus, conmutadores, etc.

Esquema propuesto Red multietapa OMEGA

La red tiene N entradas y define $\log_2 N$ etapas de $N/2$ elementos conmutadores de 2×2 cada uno. En total, la red tendrá $(N/2)\log_2 N$ conmutadores. Por su puesto, cada conmutador se gobierna de forma independiente a los demás. El patrón de interconexión entre las etapas es un *perfect shuffle*

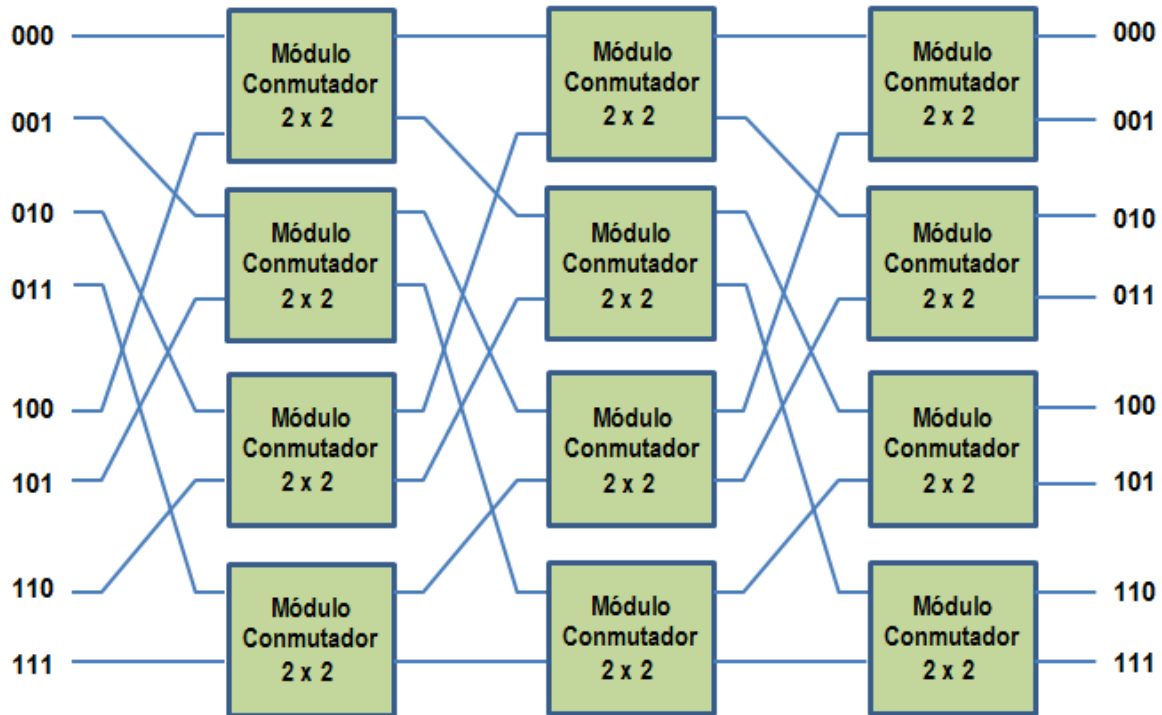


Figura 3. Red Omega

De acuerdo a la investigación, existe una manera para determinar el número de módulos conmutadores esta es $N/2$ es decir en nuestro caso se tendrán:

$$N = \text{ENTRADAS} = 8$$

$$\text{Conmutadores} = 8/2 = 4$$

$$\text{Etapas} = \ln(N)/\ln(2) = 3$$

Solo existen las siguientes conmutaciones dentro de un conmutador:

0 a 0 (Directa)

0 a 1 (Cruzada)

1 a 0 (Cruzada)

1 a 1 (Directa) Ver figura 4.

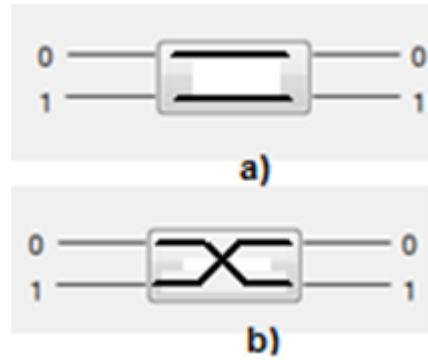


Figura 4. Tipos de conmutador a) directo y b) cruzado.

Es decir tendremos 3 etapas cada una con 4 conmutadores, utilizaremos la representación binaria en 3 bits para referirnos al número de procesador y el número de la memoria que se deseen comunicar, esto será de manera transparente al usuario, es decir el usuario introducirá el número en decimal e internamente nosotros haremos la conversión. A continuación se detallan los pasos y la forma para obtener la simulación:

Supongamos que deseamos comunicar el procesador 4 (100 en binario) con la memoria 2 (010 en binario). Ver figura 5.



Figura 5. Configuración número de procesadores vs número de memorias

1. Ambos números en binarios los concatenamos, quedando una cifra de 6 dígitos o 6 bits como se conoce en el área de cómputo:100010.

2. Para determinar el conmutador que tendrá que activarse y el tipo de conmutación que se realizara en la etapa 1 (sabemos que serán 3) se comienza en la posición 0 de la cadena concatenada y se obtienen 4 bits, quedando 1000, de esta subcadena los bits de en medio es decir el 1y2 determinan el conmutador es decir 00 en decimal es el switch 0 y el bit 0 (que es un 1) determina el origen del switch, el bit 3 (que es un 0) determina el destino. *Ver figura 6.*

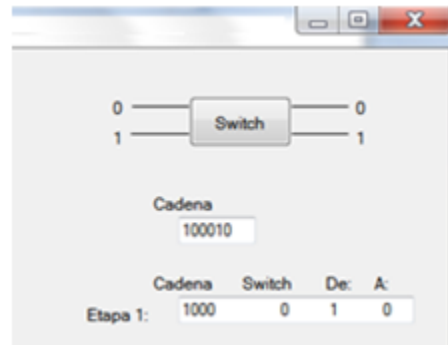


Figura 6. Configuración subcadena etapa1

3. Para determinar el conmutador que tendrá que activarse y el tipo de conmutación que se realizara en la etapa 2 (sabemos que serán 3) se comienza en la posición 1 de la cadena concatenada y se obtienen 4 bits, quedando 0001, de esta subcadena los bits de en medio es decir el 1y2 determinan el conmutador es decir 00 en decimal es el switch 0 y el bit 0 (que es un 0) determina el origen del switch, el bit 3 (que es un 1) determina el destino. *Ver figura 7.*

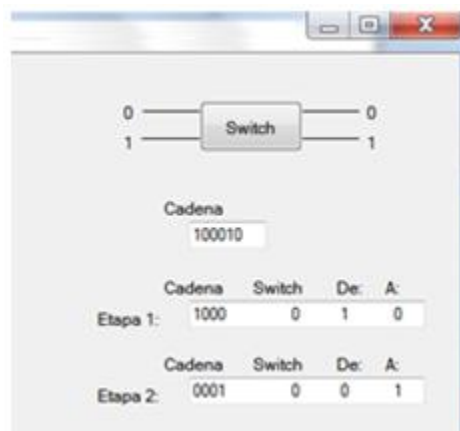


Figura 6. Configuración subcadena etapa 2.

- Para determinar el conmutador que tendrá que activarse y el tipo de conmutación que se realizara en la etapa 3 (sabemos que serán 3) se comienza en la posición 1 de la cadena concatenada y se obtienen 4 bits, quedando 0010, de esta subcadena los bits de en medio es decir el 1y2 determinan el conmutador es decir 01 en decimal es el switch 1 y el bit 0 (que es un 0) determina el origen del switch, el bit 3 (que es un 0) determina el destino. *Ver figura 8.*

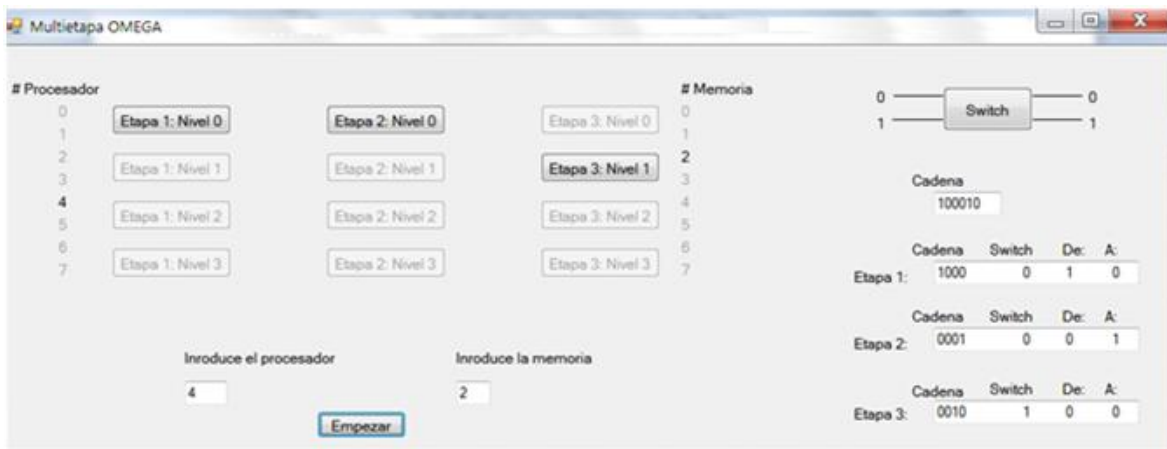


Figura 8. Pantalla de salida del simulador

- Si lo notamos conforme avanzamos en la etapa recorreremos en una posición hacia adelante, siempre obteniendo 4 dígitos, esto es debido a que solo necesitamos 2 bits que nos determinan los conmutadores que se pueden encender (00,01,10,11) y un bit para saber el origen y el destino.

III. Código en Visual C# 2010

[Clic aquí para ver el código](#)

IV. Conclusiones

Si se quisiera tener un algoritmo general se obtendría de la siguiente manera:

1. El número de procesadores debe ser igual al de las memorias.
2. Se debe calcular el número de las etapas y el número de conmutadores mediante las formulas ya mencionadas en la investigación y solución.
3. Para determinar el número de bits que se extraen de la cadena concatenada en cada etapa se determina mediante la fórmula: $(\text{numero de etapas}-1)+2$.
4. Recordando que en cada etapa se va incrementando el índice inicial en 1, es decir en la etapa 1 se empieza en el índice 0, en la etapa 2 el índice es 1 y así sucesivamente.

V. Bibliografía

1. Flynn, M.J. Some computer organizations and their effectiveness. *IEEE Trans. on Computers*, Vol. C-21, pp. 948-960, 1972.
2. Germain-Renaud, C. y Sansonnet, J.P. *Ordenadores Masivamente Paralelos*. Paraninfo, Madrid, 1993.
3. Reed, D.A. and Fujimoto, R.M. *Multicomputer Networks: Message-Based Parallel Processing*, MIT Press, Cambridge, Mass., 1987.
4. Siegel, H.J. *Interconnection Networks for Large-scale Parallel Processing*. McGraw-Hill, New York, 1990.
5. Athas, W.C. and Seitz, C.L. Multicomputers: Message-passing concurrent computers. *IEEE Computer*, Vol. 21, No. 8, pp. 9-24, August 1988.

6. Bokhari, S.H. On the mapping problem. *IEEE Trans. on Computers*, Vol. C-30, No. 3, pp. 207-214, March 1981. Cambridge 1991.

7. Bauch, A.; Braam, R. and Maehle, E. DAMP: A dynamic reconfigurate multiprocessor system with a distributed switching network. *2nd European Distributed Memory Computing Conference*, Munich, April 1991.

8. García, J.M. and Duato, J. Dynamic reconfiguration of multicomputer network: Limitations and Tradeoffs, in P. Milligan and A. Nuñez (Eds.), *Euromicro Workshop on Parallel and Distributed Processing*, IEEE Computer Society Press, pp. 317-323, 1993.