

---

## IMPLEMENTACIÓN DE UNA ARQUITECTURA MIMD PARA CÓMPUTO CIENTÍFICO UTILIZANDO BUS ISA Y LÓGICA PROGRAMABLE

*Israel Rivera Zárate*

*irivera@ipn.mx*

*Miguel Hernández Bolaños,*

*mbolanos@ipn.mx*

*Patricia Pérez Romero*

*promerop@ipn.mx*

*INSTITUTO POLITECNICO NACIONAL-CIDETEC*

### I. Introducción

La computación paralela surge como una respuesta ante la creciente necesidad de aumento en el rendimiento de cómputo, entendido como una aceleración en la ejecución de instrucciones ya sea por aumento en la frecuencia del ciclo de reloj o bien por la ejecución simultánea o concurrente de instrucciones en función del tiempo. Para satisfacer dichas demandas se han llevado a cabo diferentes formas de computación paralela, desde el paralelismo a nivel de bit, paralelismo a nivel de instrucción, paralelismo de datos y paralelismo de tareas [1].

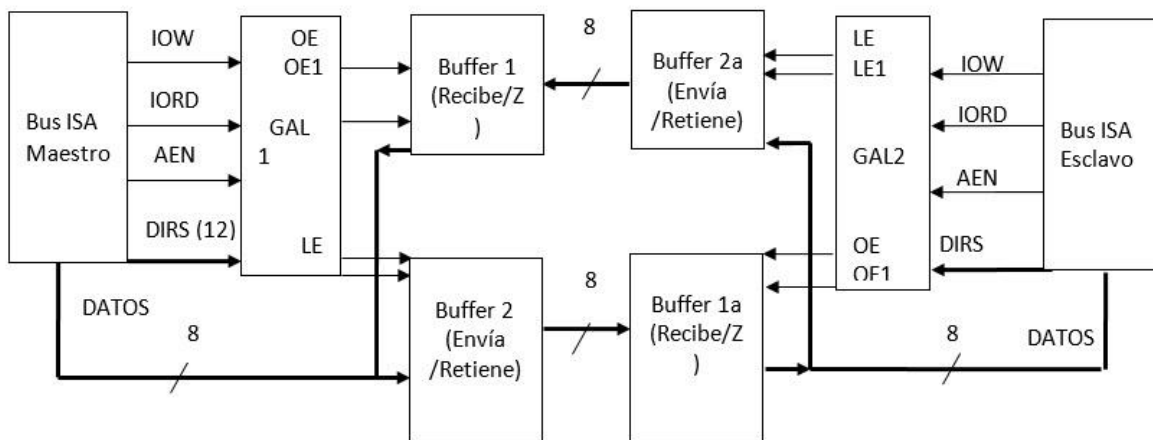
De acuerdo con [1] y [2] Michael Flynn en los años setenta propuso una taxonomía para la clasificación de las computadoras. El método de Flynn se basa en la fuente de instrucciones a lo que llama flujo de instrucciones y de la secuencia de datos que la computadora utiliza para procesar información a lo que llama flujo de datos. Puede haber secuencias de instrucciones sencillas o múltiples y secuencias de datos sencillas o múltiples. Dicha clasificación da lugar a cuatro tipos de computadoras (Ver figura 1).

- SISD : Flujo único de instrucciones y flujo único de datos
- SIMD : Flujo único de instrucciones y flujo múltiple de datos
- MIMD : Flujo múltiple de instrucciones y flujo múltiple de datos
- MISD : Flujo múltiple de instrucciones y flujo único de datos

Los modelos **SIMD** y **MIMD** son los únicos modelos aplicables a las computadoras paralelas. También hay que mencionar que el modelo MISD es teórico.



Se implementará un esquema de procesamiento paralelo tipo MIMD débilmente acoplado, esto es; dos nodos computacionales (2 equipos de cómputo) con memoria distribuida interconectados por un bus común. En este caso se empleará el bus de expansión ISA (Industry Standard Architecture) donde adicionalmente mediante lógica programable se podrá llevar a cabo la transferencia de información como sucede en el protocolo de paso de mensajes que distingue a esta arquitectura de procesamiento paralelo. Para lograr la comunicación de forma bidireccional usando el bus ISA se propone utilizar 2 GAL 22V10D que serán programadas como decodificadores para la dirección y para la activación de comunicación, de tal forma que permita la comunicación entre dos equipos. A continuación se muestra el diagrama de bloques donde se observa de manera general la configuración del circuito. (Figura 2.)



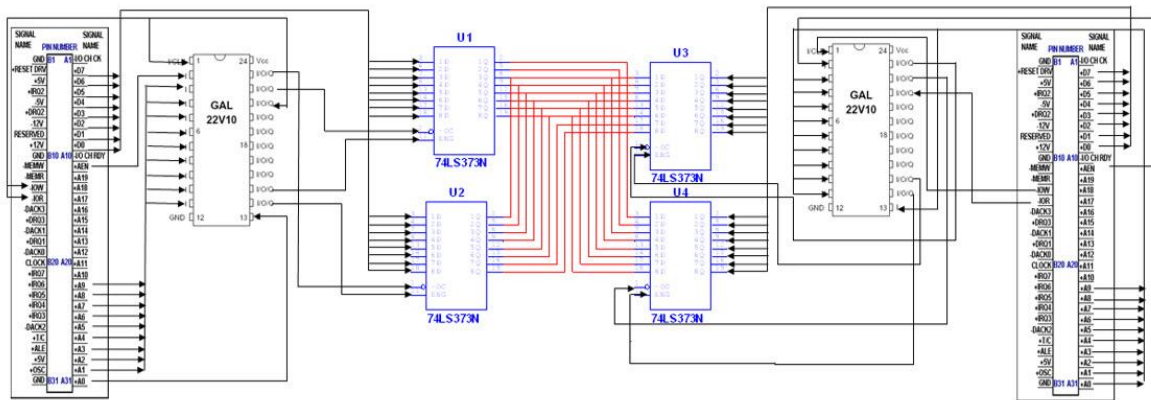
**Figura 2.** Diagrama del sistema basado en bus ISA

Como se puede observar en el modelo esquemático las dos GAL tienen que ser programadas de la misma forma, lo que permitirá tener las dos operaciones tanto de lectura como de escritura, todo esto en la dirección 300h (hexadecimal). Esto requiere que se estén activadas las señales lógicas OE (Output Enable) y LE (Latch Enable) en las 2 GAL lo cual permitirá llenar los buffers para que se logre retener el dato y poderlo procesar.

Los buffers 1 y 1ª están configurados de manera que puedan funcionar como transmisores de datos mientras que los buffers 2 y 2a están configurados de manera de retención de datos, para lograr la retención de datos, nos auxiliamos de unos integrados con serie 74F373.

En la figura 3. Se observa el modelo específico de conexiones, donde nos muestra también el modelo esquemático de la GAL que fue empleado en las dos tarjetas ISA.

El modelo fue desarrollado en base a lo que se proponía en el modelo general, pero con las excepciones que se crearon dos salidas de LACTH una para cada uno simplemente para saber cuál es la que tenía la función de escritura y cuál era la de función de lectura y los OUTPUT ENABLE se crearon una para cada uno de igual manera para saber cuál era de lectura y cual de escritura, mientras que el bus se creó bidireccional para que este procese y envíe los datos.



**Figura 3.** Modelo Específico de Conexiones

**Protocolo de comunicación propuesto.**

Para poder establecer una comunicación entre ambas computadoras a través del Bus ISA se implementó un protocolo dentro de los 8 bits de datos con los que se cuenta. El hardware empleado con las tarjetas ISA de ambas computadoras crean dos Bus de datos, uno para envío y otro para recepción, donde cada tarjeta controla su Bus de envío.

El protocolo consiste en asignar 4 bits del envío para datos (los más significativos) y los otros 4 para control (los menos significativos) de acuerdo a como se aprecia en la tabla 1.

**Tabla 1.** Formato de la trama del protocolo de comunicación

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Dato 3	Dato 2	Dato 1	Dato 0	Inicio de Envío	Fin de envío	Confirmación de Recepción	Envío De Información

Como el envío de información se realiza a través de archivos, cada envío se maneja como una cadena de bytes y cada Byte se envía en 2 tiempos. (Ver tabla 2.) Es decir, cada Byte se envía como se muestra en esta tabla.

**Tabla 2.** Envío de 1 byte en 2 tiempos

	Dato a enviar							
1er Byte	D1 b7	D1 b6	D1 b5	D1 b4	D1 b3	D1 b2	D1 b1	D1 b0
2o Byte	D2 b7	D2 b6	D2 b5	D2 b4	D2 b3	D2 b2	D2 b1	D2 b0

**Tabla 3.** Envío de 2 bytes en 2 tiempos

	Envío							
1	D1 b7	D1 b6	D1 b5	D1 b4	1	0	0	1
2	D1 b3	D1 b2	D1 b1	D1 b0	0	0	0	1
3	D2 b7	D2 b6	D2 b5	D2 b4	0	0	0	1
4	D2 b3	D2 b2	D2 b1	D2 b0	0	1	0	1

En este caso se está mandando una cadena de 2 Bytes en 4 envíos indicando en los bits 2 y 3 el inicio y fin de la cadena, y en el bit 0 que se trata de un elemento de la cadena. (Ver tabla 3.)

Finalmente el algoritmo para el proceso de comunicación entre ambas computadoras, se lleva a cabo a través de los siguientes pasos. Suponiendo que la Pc A envía información a la Pc B:

1. La **Pc A** genera el dato a enviar, indicando si se trata del primer Byte o el último.
2. La **Pc A** coloca el envío en el **Bus A**.
3. La **Pc B** lee el dato del **Bus A**.
4. La **Pc B** coloca la confirmación de recepción en el **Bus B** (00000010).
5. La **Pc A** lee la confirmación del **Bus B** y limpia el **BUS A** (00000000).
6. La **Pc B** lee que se haya limpiado el **Bus A** y limpia el **Bus B**.
7. Si la **Pc A** tiene otro dato por enviar, vuelve al paso 1, de lo contrario continua.

### III. Prueba de funcionamiento

#### Multiplicación de matrices

La multiplicación de las matrices cuadradas A y B se realizó de forma paralela, de la siguiente manera:

En la máquina maestra se generan pseudoaleatoriamente ambas matrices, A y B, y en ésta se calcula la mitad de la matriz C (producto), las partes necesarias para el cálculo y las obtenidas se muestran sombreadas. (Ver figura 4.)

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}$$
  

$$C = \begin{bmatrix} a_{00}b_{00} + a_{01}b_{10} + a_{02}b_{20} + a_{03}b_{30} & a_{00}b_{01} + a_{01}b_{11} + a_{02}b_{21} + a_{03}b_{31} & a_{00}b_{02} + a_{01}b_{12} + a_{02}b_{22} + a_{03}b_{32} & a_{00}b_{03} + a_{01}b_{13} + a_{02}b_{23} + a_{03}b_{33} \\ a_{10}b_{00} + a_{11}b_{10} + a_{12}b_{20} + a_{13}b_{30} & a_{10}b_{01} + a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{10}b_{02} + a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{10}b_{03} + a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{20}b_{00} + a_{21}b_{10} + a_{22}b_{20} + a_{23}b_{30} & a_{20}b_{01} + a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{20}b_{02} + a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{20}b_{03} + a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{30}b_{00} + a_{31}b_{10} + a_{32}b_{20} + a_{33}b_{30} & a_{30}b_{01} + a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{30}b_{02} + a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{30}b_{03} + a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{bmatrix}$$

**Figura 4.** Esquema de partición de elementos para el producto de matrices

La máquina maestra le pasa a la máquina esclava la matriz A completa y la mitad de la matriz B para que ésta calcule la mitad de la matriz C, las partes utilizadas y las obtenidas se muestran sombreadas. (Ver figura 5.)

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \\ b_{20} & b_{21} \\ b_{30} & b_{31} \end{bmatrix}$$

$$C = \begin{bmatrix} a_{00}b_{00} + a_{01}b_{10} + a_{02}b_{20} + a_{03}b_{30} & a_{00}b_{01} + a_{01}b_{11} + a_{02}b_{21} + a_{03}b_{31} & a_{00}b_{02} + a_{01}b_{12} + a_{02}b_{22} + a_{03}b_{32} & a_{00}b_{03} + a_{01}b_{13} + a_{02}b_{23} + a_{03}b_{33} \\ a_{10}b_{00} + a_{11}b_{10} + a_{12}b_{20} + a_{13}b_{30} & a_{10}b_{01} + a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{10}b_{02} + a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{10}b_{03} + a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{20}b_{00} + a_{21}b_{10} + a_{22}b_{20} + a_{23}b_{30} & a_{20}b_{01} + a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{20}b_{02} + a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{20}b_{03} + a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{30}b_{00} + a_{31}b_{10} + a_{32}b_{20} + a_{33}b_{30} & a_{30}b_{01} + a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{30}b_{02} + a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{30}b_{03} + a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{bmatrix}$$

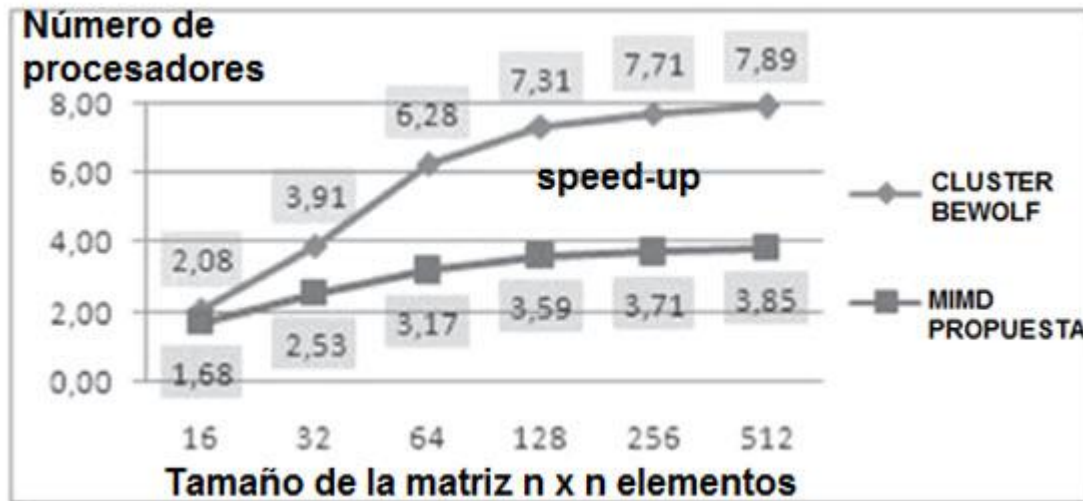
**Figura 5.** Segunda parte del esquema de partición de elementos para el producto de matrices

Finalmente la computadora esclava manda la mitad de C a la computadora maestra para unir las y presentar el resultado (Ver figura 6.)

$$C = \begin{bmatrix} a_{00}b_{00} + a_{01}b_{10} + a_{02}b_{20} + a_{03}b_{30} & a_{00}b_{01} + a_{01}b_{11} + a_{02}b_{21} + a_{03}b_{31} & a_{00}b_{02} + a_{01}b_{12} + a_{02}b_{22} + a_{03}b_{32} & a_{00}b_{03} + a_{01}b_{13} + a_{02}b_{23} + a_{03}b_{33} \\ a_{10}b_{00} + a_{11}b_{10} + a_{12}b_{20} + a_{13}b_{30} & a_{10}b_{01} + a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{10}b_{02} + a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{10}b_{03} + a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{20}b_{00} + a_{21}b_{10} + a_{22}b_{20} + a_{23}b_{30} & a_{20}b_{01} + a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{20}b_{02} + a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{20}b_{03} + a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{30}b_{00} + a_{31}b_{10} + a_{32}b_{20} + a_{33}b_{30} & a_{30}b_{01} + a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{30}b_{02} + a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{30}b_{03} + a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{bmatrix}$$

**Figura 6.** Matriz producto

**Tabla 3.** Speedup comparativa entre cluster beowolf versus el esquema MIMD propuesto proyectando hasta 8 procesadores



#### IV. Códigos fuente del protocolo y multiplicación de matrices

[Clic aquí para ver el código](#)

#### V. Resultados y Conclusiones

Por simplicidad, todo el tiempo se trabajó con matrices cuadradas de tamaño  $n \times n$ . Se consideró que el número de procesadores de la máquina paralelas es 2. El esquema de procesamiento utilizado correspondió al débilmente acoplado; por lo que se empleó protocolo de paso de mensajes para la comunicación. Las matrices a multiplicar serán A y B, ambas tratadas como matrices densas (con pocos 0's), el resultado lo almacenaremos en la matriz C.

Por otra parte se realizaron pruebas de funcionamiento variando el tamaño de la matriz vs la ejecución secuencial y paralela obteniéndose ganancias en velocidad (speedup). Se debe considerar que el tiempo paralelo del algoritmo  $T_p$  se determina [6], para sistemas paralelos con memoria distribuida, como  $T_p = T_a + T_c + T_{sol}$  donde  $T_a$  es el tiempo aritmético,  $T_c$  el tiempo de comunicación y  $T_{sol}$  es el tiempo de solapamiento. Debido a la complejidad del cálculo de  $T_{sol}$  se hace la aproximación  $T_p \approx T_a + T_c$ . El speedup ( $S_p$ ) del algoritmo, que indica la ganancia de velocidad del algoritmo comparado con el secuencial, está dado en [3.]

Para tener una idea del desempeño del sistema se decidió tomar como referencia los resultados presentados en [4]. Donde se observa la comparación de ejecución paralela de cálculo de matrices de diferentes tamaños de un cluster beowolf. Cluster formado por 9 PC con

procesador Intel (R) Core (TM) 2 Duo CPU E4500 2.2.0 GHz y memoria principal de 1(2x512) GB de RAM con sistema operativo Linux Debian Lenny 5.0 y una red 100Mbps.

Por lo tanto los resultados obtenidos se presentan en la tabla 3. Respecto del esquema MIMD desarrollado. Implementado en 2 PC con procesador Intel (R) Core (TM) 2 Duo CPU E4500 2.2.0 GHz y memoria principal de 1(2x512) GB de RAM con sistema operativo Windows 7 y vía interconexión de Bus ISA con tasa de transferencia de 16MB/s. Se puede observar que existe una tenencia de speedup y eficiencia óptimos a medida que el tamaño de las matrices aumenta, así mismo con el aumento en el número de procesadores. Bajo estas condiciones se puede evidenciar que comparando el cluster bewolf en el caso de 2 procesadores alcanza el valor de de 2.0 versus el tiempo del sistema MIMD se obtuvo un speedup promedio de 1.68. Lo que demuestra que la arquitectura propuesta satisface en un 80% la cota teórica esperada de 2.0. Por lo que se considera satisfactoria la arquitectura MIMD que se propone dirigida a aplicaciones de cómputo científico donde el cálculo matricial es ampliamente utilizado.

Link del video del proyecto final: <http://www.youtube.com/watch?v=NIKZyNHM90Q>

## Referencias

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. V. Kumar, Ananth Grama, Anshul Gupta and George Karypis, Introduction to parallel computing. Design and Analysis of Algorithms, The Benjamin/Cummings Pub. Company, RedwoodCity, California 1994.
4. Dannier Trinchet y Asnay Guirola. Algoritmo paralelo para el cálculo de matrices de probabilidades de transición: aplicación a la modelación de yacimientos lateríticos mediante cadenas de Markov. 2011

