Capítulo 2

Procesadores matriciales

Según la clasificación de Flynn uno de los cuatro tipos de sistemas es el SIMD (*Single Instruction stream Multiple Data stream*) Este tipo de sistemas explotan el paralelismo inherente en los datos más que en las instrucciones. El computador de tipo SIMD clásico es el computador matricial.

La bibliografía para este tema se encuentra en [HB87] y [Hwa93].

2.1 Organización básica

La configuración básica de un procesador matricial se muestra en la figura 2.1. Como vemos se trata de N elementos de proceso (EP) sincronizados y bajo el control de una única unidad de control (UC). Cada elemento de proceso está formado básicamente por una unidad aritmético lógica, asociada a unos registros de trabajo, y una memoria local para el almacenamiento de datos distribuidos. La unidad de control, que muchas veces es un procesador escalar, tiene su propia memoria para almacenar el programa y datos. Las instrucciones escalares y de control como saltos, etc. se ejecutan directamente en la unidad de control. Las instrucciones vectoriales son transmitidas a los EPs para su ejecución. De esta manera se alcanza un alto grado de paralelismo gracias a la multiplicidad de los elementos procesadores.

Este esquema que se acaba de comentar y que corresponde al de la figura 2.1, es el modelo de computador matricial con *memoria distribuida*. Otra posibilidad consiste en tener la *memoria compartida* intercalando la red de interconexión entre los elementos de proceso y las memorias. Las diferencias con el modelo anterior son que las memorias ligadas a los EPs son sustituidas por módulos en paralelo que son compartidos por todos los EPs mediante la red de interconexión. La otra diferencia es que la red de interconexión del modelo de la figura se intercambia por la red de interconexión o *alineamiento* entre los elementos de proceso y las memorias.

Un modelo operacional para los computadores matriciales viene especificado por la siguiente quíntupla:

$$M = \langle N, C, I, M, R \rangle \tag{2.1}$$

donde:

1. N es el número de elementos de proceso en la máquina. Por ejemplo, la Illiac IV tiene 64 EPs, mientras que la Connection Machine CM-2 tiene 65.536 EPs.

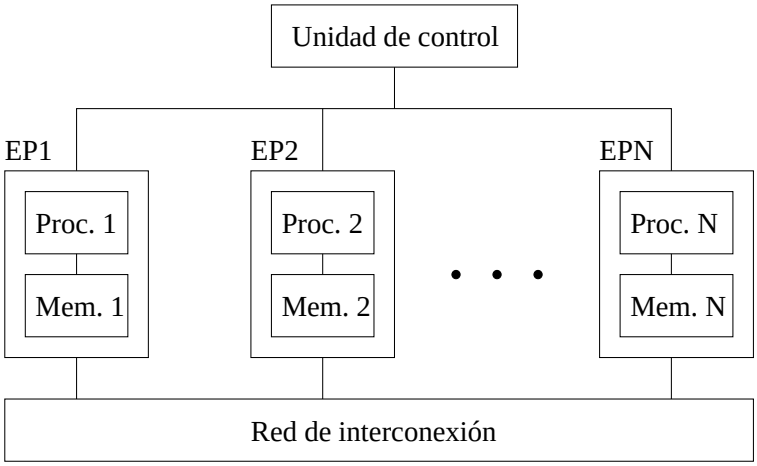

Figura 2.1: Computador matricial básico.

2. C es el conjunto de instrucciones ejecutadas directamente por la unidad de control incluyendo las instrucciones escalares y las de control del flujo de ejecución.
3. I es el conjunto de instrucciones que la unidad de control envía a todos los EPs para su ejecución en paralelo. Estas instrucciones son aritméticas, lógicas, rutado de datos, enmascaramiento, y otras operaciones locales que son ejecutadas por los EPs sobre la memoria local.
4. M es el conjunto de posibilidades de enmascaramiento donde cada máscara se encarga de dividir el conjunto de EPs en subconjuntos de EPs habilitados o deshabilitados.
5. R es el conjunto de funciones de rutado que especifican varios patrones para ser establecidos en la red de interconexión para intercomunicación entre EPs.

2.2 Estructura interna de un elemento de proceso

Aunque las características de un EP en un procesador matricial pueden variar de unas máquinas a otras, se van a dar aquí unas nociones generales de los elementos básicos que forman los EPs en un computador matricial. La figura 2.2 muestra un procesador ejemplo para ilustrar las explicaciones.

Vamos a suponer que cada elemento de proceso EP_i tiene su memoria local MEP_i. Internamente al EP habrá un conjunto de registros e indicadores formado por A_i, B_i,C_i y S_i, una unidad aritmético-lógica, un registro índice local I_i, un registro de direcciones D_i, y un registro de encaminamiento de datos R_i. El R_i de cada EP_i está conectado al R_j de otros EPs vecinos mediante la red de interconexión. Cuando se producen transferencias de datos entre los los EPs, son los contenidos de R_i los que se transfieren. Representamos los N EPs como EP_i para $i = 0, 1, \ldots, N-1$, donde el índice i es la dirección del EP_i. Definimos una constante m que será el número de bits necesarios para codificar el número N. El registro D_i se utiliza entonces para contener los m bits de la dirección del EP_i. Esta estructura que se cuenta aquí está basada en el diseño del Illiac-IV.

Algunos procesadores matriciales pueden usar dos registros de encaminamiento, uno para la entrada y otro para la salida. Se considerará en nuestro caso un único registro

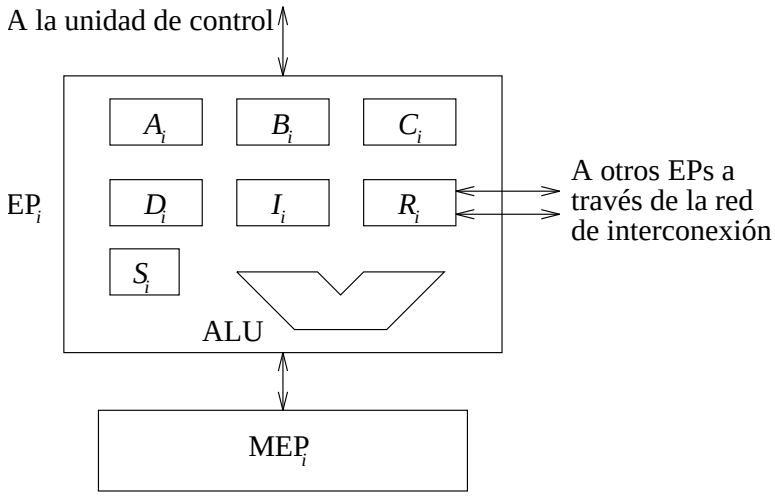

Figura 2.2: Componentes de un elemento de proceso (EP) en un computador matricial.

(R_i) en el cual las entradas y salidas están totalmente aisladas gracias al uso de biestables maestro-esclavo. Cada EP$_i$ está o bien en modo activo o bien en modo inactivo durante cada ciclo de instrucción. Si un EP$_i$ está activo, ejecuta la instrucción que le ha enviado la unidad de control (UC). Si está inactivo no ejecuta la instrucción que reciba. Los esquemas de enmascaramiento se utilizan para modificar los indicadores de estado S_i, supondremos que si $S_i = 1$ entonces el EP$_i$ está activo, y sino, está inactivo.

El conjunto de indicadores S_i para $i = 0, 1, \ldots, N-1$, forma un registro de estado S para todos los EPs. En la unidad de control hay un registro que se llama M que sirve de máscara para establecer el estado de los EPs, por lo tanto M tiene N bits. Obsérvese que las configuraciones de bits de los registros M y S son intercambiables bajo el control de la UC cuando se va a establecer un enmascaramiento.

Desde el punto de vista hardware, la longitud física de un vector está determinada por el número de EPs. La UC realiza la partición de un vector largo en bucles vectoriales, el establecimiento de una dirección base global y el incremento de desplazamiento respecto de esa dirección base. La distribución de los elementos vectoriales sobre las diferentes MEP es crucial para la utilización eficaz de una colección de EPs. Idealmente se deberían poder obtener N elementos simultáneamente procedentes de las diferentes MEP. En el peor caso, todos los elementos del vector estarían alojados en una sola MEP. En tal situación, tendrían que ser accedidos de forma secuencial, uno tras otro.

Un vector lineal unidimensional de n elementos puede ser almacenado en todas las MEP si $n \leq N$. Los vectores largos ($n > N$) pueden ser almacenados distribuyendo los n elementos cíclicamente entre los N EPs. El cálculo de matrices bidimensionales puede causar problemas, ya que pueden ser necesarios cálculos intermedios entre filas y columnas. La matriz debería almacenarse de modo que fuera posible el acceso en paralelo a una fila, una columna, o una diagonal en un ciclo de memoria.

En un procesador matricial, los operandos vectoriales pueden ser especificados por los registros a utilizar o por las direcciones de memoria a referenciar. Para instrucciones de referencia a memoria, cada EP$_i$ accede a la MEP$_i$ local, con el desplazamiento indicado por su propio registro índice I_i. El registro I_i modifica la dirección de memoria global difundida desde la UC. Así, diferentes posiciones pueden ser accedidas en diferentes MEP$_i$ simultáneamente con la misma dirección global especificada por la UC.

La utilización del registro índice local (I_i) es evidente cuando se quiere acceder a los elementos de la diagonal de una matriz. Si la matriz se encuentra colocada de forma consecutiva a partir de la dirección 100, habrá que cargar cada registro índice con el desplazamiento correspondiente a cada elemento de la diagonal. Una vez inicializados los índices, se pasará como operando la dirección 100 que es la del comienzo de la matriz, y cada procesador leerá un elemento distinto indicado por el índice, en este caso la diagonal de la matriz.

Aparte del propio paralelismo obtenido por el cálculo en paralelo sobre los elementos de un vector, el propio encaminamiento de los datos reduce el tiempo de ejecución de determinadas tareas. Por ejemplo, en el tratamiento de imágenes, donde muchas operaciones son cálculos entre píxels vecinos, una arquitectura matricial paraleliza con facilidad los cálculos. También en operaciones donde unos datos en un vector dependen de resultados anteriores sobre ese vector puede beneficiarse de la flexibilidad de interconexión entre los EPs.

Los procesadores matriciales son computadores de propósito específico destinados a limitadas aplicaciones científicas donde pueden alcanzar un rendimiento elevado. Sin embargo, los procesadores matriciales tienen algunos problemas de vectorización y programación difíciles de resolver. Lo cierto es que los computadores matriciales no son populares entre los fabricantes de supercomputadores comerciales.

2.3 Instrucciones matriciales

2.4 Programación

2.4.1 Multiplicación SIMD de matrices

2.5 Procesadores asociativos

2.5.1 Memorias asociativas

2.5.2 Ejemplos de procesadores asociativos