Arquitectura FPGA para un Procesador Matricial

Miguel Morales-Sandoval, Moisés Pérez-Gutiérrez,
Claudia Feregrino-Uribe, Miguel Arias-Estrada

Coordinación de Ciencias Computacionales
Instituto Nacional de Astrofísica Óptica y Electrónica, INAOE.
e-mail : {mmorales, mperez}@ccc.inaoep.mx ,{cferegrino,ariasm}@inaoep.mx
Luis Enrique Erro #1 Apdo Postal 51 y 216, CP 72000, Tonantzintla, Puebla, México.

Resumen : La multiplicación de matrices es una operación muy común en ciencias e ingeniería, sin embargo, su costo computacional es elevado; por lo que es necesario disponer de alternativas para la ejecución de esta tarea de manera eficiente disminuyendo el tiempo de procesamiento que es muy importante en diversas aplicaciones. Este trabajo presenta una arquitectura para la multiplicación paralela de matrices en un dispositivo FPGA. El diseño que se presenta utiliza una matriz de elementos procesadores que realizan la multiplicación de matrices.

Palabras Clave: Procesador Matricial, FPGA, Handel-C.

1. Introducción.

Las matrices se utilizan en el cálculo numérico en la resolución de sistemas de ecuaciones lineales, de las ecuaciones diferenciales y de las derivadas parciales. Además de su utilidad para el estudio de sistemas de ecuaciones lineales, las matrices aparecen de forma natural en geometría, estadística, economía, informática, física, entre otras[1][4]. El problema tiene una estructura simple y un conjunto de propiedades bien definido por lo cual es utilizado como benchmark para computadoras paralelas[2].

Dentro del ámbito computacional, existen varios enfoques para atacar el problema que se tiene para multiplicar matrices de dimensiones grandes, donde principalmente, se busca reducir el tiempo de cómputo empleado. Los algoritmos propuestos en la literatura, buscan aprovechar el paralelismo inherente en el problema: se puede realizar la multiplicación en un tiempo menor si se aprovecha la localidad temporal de los coeficientes de ambas matrices[3].

El problema de la multiplicación de matrices tiene un orden de complejidad de $O(n^3)$, sin embargo, puede reducirse aplicando otras técnicas como el algoritmo de Strassen logrando una complejidad de $O(n^{2.7})$[1][4].

El presente trabajo propone una alternativa para la multiplicación de matrices mediante una arquitectura Hardware/Software. La arquitectura Hardware se implanta

en un FPGA *Xilinx Virtex 2000E* equivalente a 2 millones de compuertas lógicas. El FPGA se encuentra incluido en la tarjeta RC1000 y se programa en el lenguaje Handel C. El complemento en software se encarga de la interfaz entre el usuario y el dispositivo FPGA.

2. Arquitectura del procesador matricial.

La arquitectura hardware se compone por una matriz de elementos procesadores(EPs), en la que cada uno se encarga de calcular el elemento C_{ij} de la matriz resultante, (ver Fig. 1). Cada EP realiza la multiplicación de un solo renglón de la matriz A con una sola columna de la matriz B.

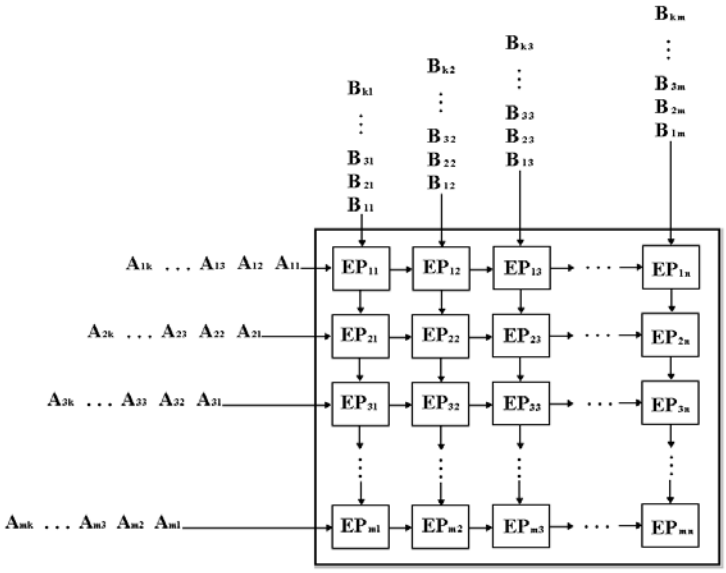


Fig.1 Malla de EP para realizar la multiplicación de matrices.

Para multiplicar matrices de dimensiones *mxk* y *kxn* se requieren de *mxn* elementos de procesamiento interconectados en forma de malla y ordenados en filas de *n* en *n* EP.

2.1 Elemento procesador

El EP es básicamente un multiplicador acumulador(Fig 2). Recibe 2 datos de 8 bits, los multiplica, suma el producto al resultado previo (inicialmente, ese resultado previo es cero) y envía los datos de entrada a los EPs vecinos. Cada elemento procesador realiza solo *k* multiplicaciones (tanto el vector renglón como el vector columna son de tamaño *k*). Después de estas *k* multiplicaciones, en el registro *Acc* de cada EP almacena el valor del coeficiente resultante de la matriz C_{ij}.

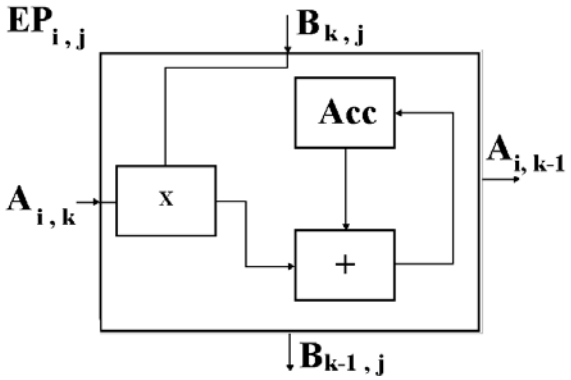
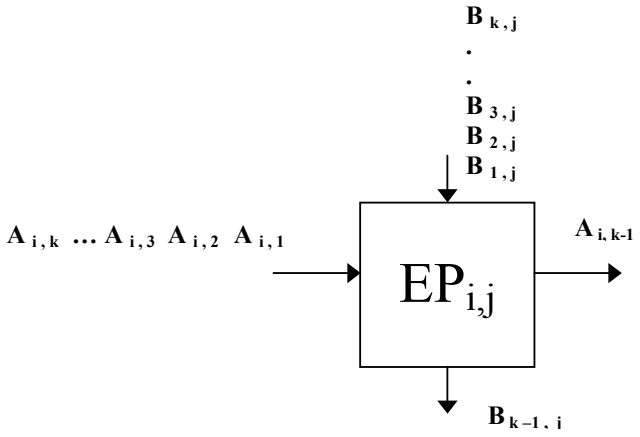


Fig. 2 Diagrama general del elemento procesador.

2.2 Asignación de coeficientes a la matriz de EPs.

Los coeficientes de la matriz A entran a la malla de EPs a través de la primera columna (EP_{11}, EP_{21}, EP_{31}, ... , EP_{m1}), los coeficientes de la matriz B entran a la malla por el primer renglón de la malla de EPs (EP_{11}, EP_{12}, EP_{13}, ..., EP_{1n}).



$$Acc_{i\,i} = A_{i1} * B_{1i} + A_{i2} * B_{2i} + ... + A_{ik} * B_{ki}$$

Fig. 3 Flujo de coeficientes en un elemento procesador.

La asignación de renglones a la primera columna de EPs se realiza con un retraso de un coeficiente entre renglones. Lo mismo ocurre con la asignación de columnas en la primera fila de EPs. Cada vez que llega un coeficiente *k* nuevo al EP_{ij}, se realiza la multiplicación y suma al acumulador propagando los datos a los siguiente EPs (Fig. 3).

3. Implantación en el FPGA de la RC1000

La arquitectura hardware se desarrolló bajo el lenguaje Handel-C. Los coeficientes de las matrices se almacenan en los bancos de memoria 0 y 1 de la tarjeta RC1000. Con esto, se realiza el acceso a las dos matrices en paralelo, acelerando la operación. Se tienen dos módulos que realizan la lectura de los coeficientes para realizar las asignaciones previamente descritas. Los elementos procesadores se activan a la llegada de nuevos coeficientes en sus entradas multiplicando y sumando al valor previo. Los resultados obtenidos se almacenan en el banco de memoria 2 de la tarjeta RC1000.

3.1 Descripción de la malla de elementos procesadores.

Para interconectar la matriz de elementos procesadores, se utiliza una matriz de canales que conectan los procesadores para la transmisión de los coeficientes (Fig 4). En Handel-C, un canal permite la comunicación entre procesos, es decir, mediante canales se pueden comunicar datos entre procesos independientes que se ejecutan en paralelo. Un canal asegura la recepción adecuada de los datos ya que tras escribir a un canal ninguna otra escritura podrá efectuarse hasta que el dato previo sea leído por otro proceso.

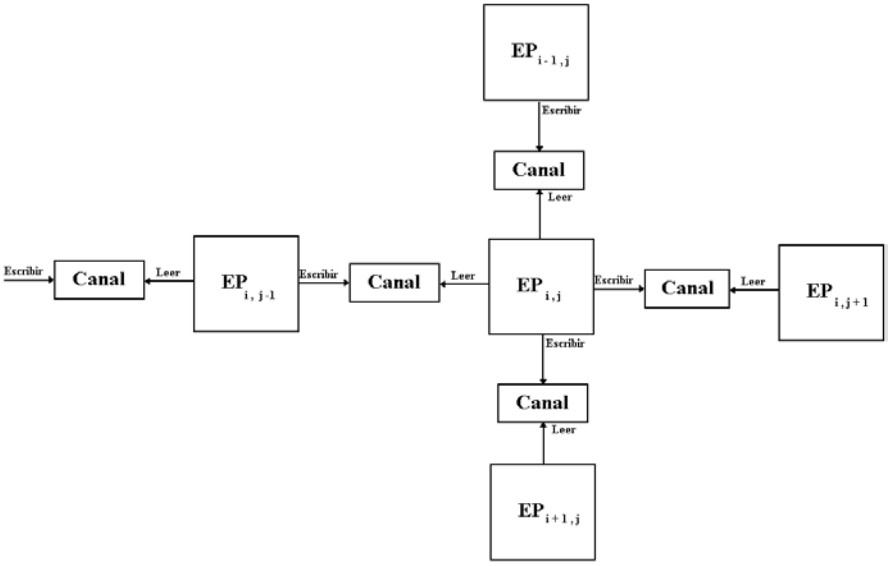


Fig. 4 Comunicación de coeficientes entre EPs mediante canales.

Cada EP se modela como una función (proceso) que se ejecuta independientemente y que tiene asociados dos canales: un canal por el que envía el coeficiente de A que recibe en algún instante su EP vecino y otro canal para transmitir a su EP correspondiente el coeficiente de B.

La interconexión se efectúa de tal forma que para multiplicar matrices de tamaño *mxk* y *kxn* se requiere una malla de *mxn* EPs. Así, se tienen canales que permiten la propagación de los coeficientes de la Matriz B (canales para columnas) y canales para propagar los coeficientes de la Matriz A (canales para renglones). Se tienen dos casos especiales, los EP del último renglón en la malla no propagan el coeficiente de la Matriz B que leen al igual que los EPs de la última columna en la malla no propagan el coeficiente de la Matriz A que leen.

Cada EP realiza la multiplicación y suma – acumulación de *k* coeficientes. Después de ella, el proceso termina dejando en el registro *Acc* el valor del coeficiente C_{ij} de la matriz resultante. Como paso final se escriben cada uno de los registros *Acc* a un banco de memoria para que el programa Host pueda obtener los resultados de la tarjeta.

4. Pruebas y Resultados

La arquitectura propuesta fue sintetizada para matrices cuadradas de hasta 7x7 elementos, utilizando un 25% del FPGA operando a una frecuencia de 62Mhz. Se realizó un programa Host en Visual C++ que se encarga de capturar los coeficientes de las matrices a multiplicar y transferir estos valores a los bancos de memoria de la tarjeta RC1000, configurar el FPGA de la RC1000 mediante el archivo *.bit* generado en la síntesis, generar una serie de señales de control para el procesamiento en el FPGA y obtener los resultados. La figura 5, muestra la interfaz definida entre la tarjeta RC1000 y el programa Host.

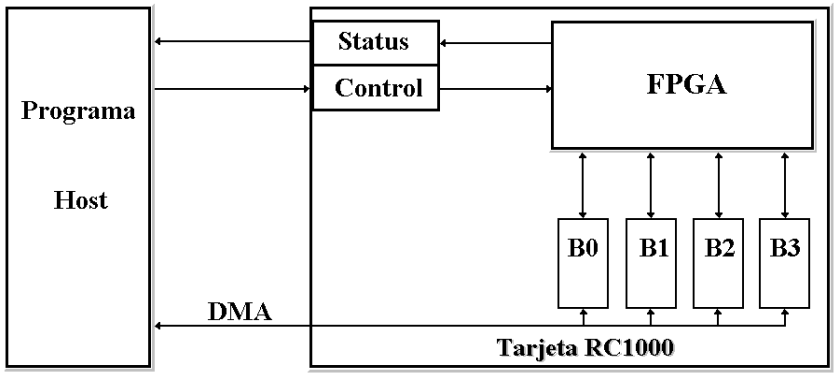


Fig. 5 Interfaz de la arquitectura propuesta.

La arquitectura se validó verificando los resultados obtenidos con resultados esperados calculados con la herramienta *MatLab*. A continuación, en la figura 6, se muestra un ejemplo de ejecución del programa Host que configura el FPGA y procesa dos matrices de dimensión 7x7 con lo que se verifica el funcionamiento del procesador matricial.

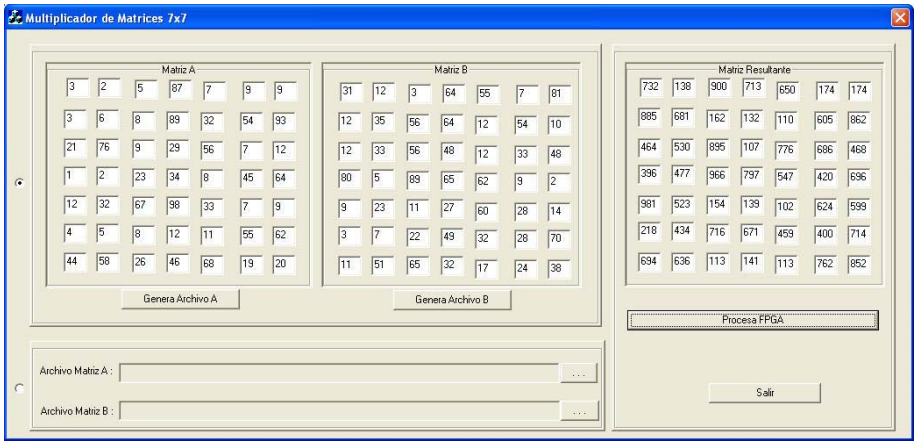


Fig. 6. Multiplicación de matrices 7x7

5. Conclusiones

Se diseñó e implantó una arquitectura hardware para la multiplicación paralela de matrices en un FPGA utilizando técnicas de paralelismo mediante una malla de EPs que, similar a la técnica de *pipeline*, propaga los coeficientes de las matrices eliminando los accesos repetidos a la memoria, reduciendo el tiempo de cómputo empleado.

La descripción del hardware se realizó en Handel C de tal forma que teóricamente puede realizarse la implantación en un FPGA para realizar la multiplicación de matrices de dimensiones *mxk* y *kxn* arbitrarias. La principal restricción radica en los recursos del FPGA.

Referencias

[1] Cormen Thomas H., Leiserson Charles E, Rivest Ronald L, " Introduction to Algorithms" The MIT Press- Mc Graw-Hill, USA 1996, pp 730-735.

[2] P. Bjorstad, F. Manne, T. Sorevik, M. Vajtersic, " Efficient Matrix Multiplication on SIMD Computers ", University of Bergen, Department of Informatics, Norway.

[3] John Gunnels, et al. "A Flexible class of parallel matrix multiplication algorithms", The University of Texas, Department of Computer Science.

[4] Keqin Li, Yi Pan, Si Qing, " Fast and Processor Efficient Parallel Matrix Multiplication Algorithms on a Linear Array With a Reconfigurable Pipelined Bus System ", IEEE Transactions on parallel and distributed systems, Vol. 9, no. 8, August 1998, pp 705-720.