A High Throughput FPGA Implementation of A Bit-Level Matrix Product

A. Amira *, A. Bouridane, P. Milligan and **P. Sage**
School of Computer Science
The Queen's University of Belfast
Belfast BT7 1NN, UK
*E-mail: A.abbes@qub.ac.uk

Abstract - This paper presents a novel architecture for a matrix product algorithm. The paper describes the mathematical model for the algorithm (based on the Baugh-Wooley algorithm), the associated design and implementation of the algorithm on a Xilinx FPGA board, and discusses the efficiency of the implementation. The architecture developed requires $O(N^2)$ and $O(2nN)$ and $O(N)$ and $O(2nN)$ as area and time complexities respectively for matrix-matrix product and matrix-vector product, respectively. (where N is the matrix size and n is the word length).

1. INTRODUCTION

Areas such as signal processing and imaging require enormous computing power. A close examination of the algorithms used in these, and related, applications reveal that many of the fundamental actions involve matrix or vector operations. Many of these operations are matrix multiplications, which occur frequently operations in a wide variety of real world algorithms, e.g. the Discrete Cosine Transform (DCT), the Discrete Fourier Transform (DFT), and Singular Values Decomposition (SVD), used in digital image/signal processing including compression and beam-forming applications [1], [3], [6], [8].

Unfortunately matrix multiplication algorithms are of $O(N^3)$ complexity and hence are computationally intensive for large size problems. Consequently, techniques that reduce this complexity are required. In real applications, data to be processed is often available as a stream of input values. In such cases, the throughput rate (which is the time between two consecutive input data) is more important than the latency (which is the time from the input of a set of data to its computed result). Therefore, there is a need to incorporate some level of pipelining in the overall system. This paper discusses the development of a generalized bit level systolic architecture suitable for matrix multiplication. These architectures are regular and require simple processing elements (comprising only gated full adders compared with word-level structures). Furthermore, the computation time of the bit-level Processing Elements (PEs) is small when compared with word-level PEs. Consequently a high throughput computation can be achieved by using bit level systolic arrays [2],[11]. It is the aim of the work reported in this paper to develop efficient architectures that are ideally qualified for computationally intensive applications with inherent parallelism.

To achieve this, the algorithms have been modified and adapted to operate in a modular and regular fashion. In this approach, a serial-parallel matrix multiplier

0-7803-6488-0/00/$10.00 © 2000 IEEE

based on the Baugh-Wooley algorithm has been selected because the algorithm has the following features [1], [2]:

- simplicity, regularity and modularity of structure; and
- flexible implementation for any desired decomposition length.

Field Programmable Gate Arrays (FPGAs) are excellent for building systolic array-style processors [7]. The systolic structures proposed in this paper has been designed and targeted to the Xilinx XCV1000E of the VirtexE family, which has the following important features [7]:

- fast, and high-density and flexible architecture that balances speed and density.
- built-in clock- management circuitry.

The composition of the rest of the paper is as follows. The algorithm for a serial-parallel matrix product is given in section 2. Section 3 is concerned with the systolic architecture for a matrix product. Section 4 gives the design characterisation. Concluding remarks are given in section 5.

2. MATHEMATICAL BACKGROUND

Consider two N x N matrices $A=[A_{ij}]$ and $B=[B_{ij}]$. The product $C=[C_{ij}]$ of A by B is given by

$$C = AB \tag{1}$$

Such that

$$C_{ij} = \sum_{k=0}^{N-1} A_{ik} B_{kj} \tag{2}$$

If the elements of the two matrices are represented using the 2's complement number representation, then:

$$\begin{aligned} A_{ik} &= -a_{ik}^{n-1} 2^{n-1} + \sum_{l=0}^{n-2} a_{ik}^{l} 2^{l} \\ \text{and } B_{kj} &= -b_{kj}^{n-1} 2^{n-1} + \sum_{m=0}^{n-2} b_{kj}^{m} 2^{m} \end{aligned} \tag{3}$$

where a_{ik}^{l} and b_{kj}^{m} are the lth bit of A_{ik} and mth bit of B_{kj}, respectively, (which are zero or one). a_{ik}^{n-1} and b_{kj}^{n-1} are the sign bits, where n is the word length .

By substituting (3) into (2), the matrix product C_{ij} can be computed as follows:

$$C_{ij} = \sum_{k=0}^{N-1} \left[-a_{ik}^{n-1} 2^{n-1} + \sum_{l=0}^{n-2} a_{ik}^{l} 2^{l} \right] \left[-b_{kj}^{n-1} 2^{n-1} + \sum_{m=0}^{n-2} b_{kj}^{m} 2^{m} \right] \tag{4}$$

From equation (4), it can be seen that the computation of the matrix product depends on the type of multiplier used. As mentioned in the introduction a Baugh-Wooley multiplier algorithm [1], [9] has been chosen and using this algorithm equation (4) becomes:

0-7803-6488-0/00/$10.00 © 2000 IEEE

$$C_{ij} = \sum_{k=0}^{N-1} \left[\begin{array}{l} \sum_{l=0}^{n-2} \sum_{m=0}^{n-2} 2^{l+m} a_{ik}^{l} b_{kj}^{m} + 2^{2n-2} a_{ik}^{n-1} b_{kj}^{n-1} + \\ \left(\sum_{l=0}^{n-2} -2^{l} a_{ik}^{l} b_{kj}^{n-1} + \sum_{m=0}^{n-2} -2^{m} b_{kj}^{m} a_{ik}^{n-1} \right) 2^{n-1} \end{array} \right] \tag{5}$$

Which can be re-expressed as follows

$$\begin{array}{l} C_{ij} = \sum_{k=0}^{N-1} \left[\sum_{l=0}^{n-2} \sum_{m=0}^{n-2} 2^{l+m} a_{ik}^{l} b_{kj}^{m} + 2^{2n-2} a_{ik}^{n-1} b_{kj}^{n-1} + \right. \\ \left. \left(\sum_{l=0}^{n-2} 2^{l} \overline{a_{ik}^{l} b_{kj}^{n-1}} + \sum_{m=0}^{n-2} 2^{m} \overline{b_{kj}^{m} a_{ik}^{n-1}} \right) 2^{n-1} + 2^{n} - 2^{2n-1} \right] \end{array} \tag{6}$$

In addition, this approach is applicable to matrix-vector multiplication. Consider a N x N matrix $A = [A_{ij}]$ and a vector $B = [B_j]$. The product $C = [C_i]$ of A by B is given by

$$C_i = \sum_{k=0}^{N-1} A_{ik} B_k \tag{7}$$

Following the same steps for matrix-matrix multiplication, the coefficients C_i can be computed as follows:

$$\begin{array}{l} C_{i} = \sum_{k=0}^{N-1} \left[\sum_{l=0}^{n-2} \sum_{m=0}^{n-2} 2^{l+m} a_{ik}^{l} b_{k}^{m} + 2^{2n-2} a_{ik}^{n-1} b_{k}^{n-1} + \right. \\ \left. \left(\sum_{l=0}^{n-2} 2^{l} \overline{a_{ik}^{l} b_{k}^{n-1}} + \sum_{m=0}^{n-2} 2^{m} \overline{b_{k}^{m} a_{ik}^{n-1}} \right) 2^{n-1} + 2^{n} - 2^{2n-1} \right] \end{array} \tag{8}$$

From (6) and (8) it can be seen that the multiplication of A_{ik} with B_{kj} or B_k expressed in two's complement number representation can be written in a form which involves only positive bit products. Therefore, the matrix product can easily be mapped into a systolic architecture.

3. DESIGN IMPLEMENTATION OF MATRIX PRODUCT

3.1. Proposed serial-parallel Baugh-Wooley multiplier

The multiplication algorithm for a word length n=4 is illustrated by the multiplication table shown in Figure 1. The partial–product terms are formed by ANDing each multiplicand bit with each multiplier bit. According to Baugh-Wooley algorithm, the product terms containing the sign information are complemented to obtain the partial product. The result is computed by the addition of '1' to the fifth and eighth columns along with all partial product terms, which is provided through an OR gate and a multiplexer with the help of the delayed control signal S1 and control signal S2 respectively.

0-7803-6488-0/00/$10.00 © 2000 IEEE

The two's complement serial-parallel multiplier is shown in Figure 2.

				b_{kj}^3	b_{kj}^2	b_{kj}^1	b_{kj}^0
				a_{ik}^3	a_{ik}^2	a_{ik}^1	a_{ik}^0
			1	$\overline{a_{ik}^3 b_{kj}^0}$	$a_{ik}^2 b_{kj}^0$	$a_{ik}^1 b_{kj}^0$	$a_{ik}^0 b_{kj}^0$
			$\overline{a_{ik}^3 b_{kj}^1}$	$a_{ik}^2 b_{kj}^1$	$a_{ik}^1 b_{kj}^1$	$a_{ik}^0 b_{kj}^1$	
		$\overline{a_{ik}^3 b_{kj}^2}$	$a_{ik}^2 b_{kj}^2$	$a_{ik}^1 b_{kj}^2$	$a_{ik}^0 b_{kj}^2$		
1	$a_{ik}^3 b_{kj}^3$	$\overline{a_{ik}^2 b_{kj}^3}$	$\overline{a_{ik}^1 b_{kj}^3}$	$\overline{a_{ik}^0 b_{kj}^3}$			
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Figure 1. Tabular form of bit level Baugh-Wooley multiplication (n=4)

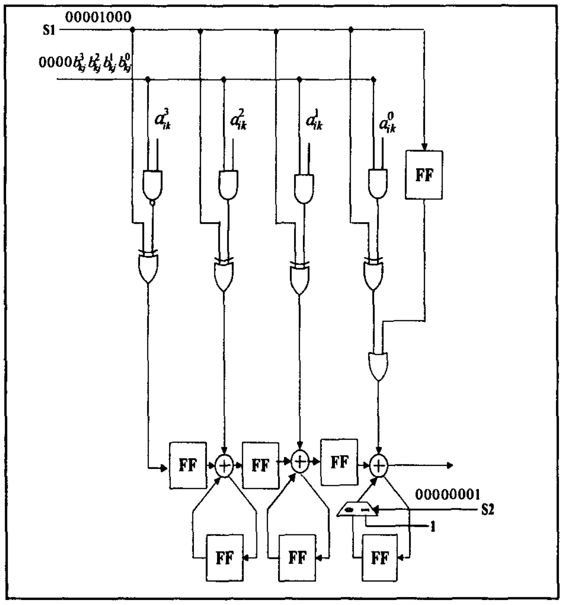

Figure 2. Serial - Parallel Baugh Wooley Multiplier -BWM -(n=4).

3.2 Proposed bit- level systolic arrays architecture

Equation (6) can be mapped into the proposed architecture. Figure 3 shows the architecture obtained for N=3 and n=4. It consists of nine identical processing elements (PEs). Each PE comprises a serial-parallel Baugh-Wooley Multiplier (BWM) as shown in Figure 2, a Flip Flop (FF) for saving the carry bit and a full adder that adds the result of the partial product and the result generated from the previous PE. The matrix elements b_{ij} are fed from the north in a parallel/serial fashion bit by bit Least Significant Bit First (LSBF) while the matrix elements a_{ij} are fed in a parallel fashion and remain fixed in their corresponding PE cell during the entire computation of the operation. Each bit of the final product of the PE is fed to the full adder of the preceding PE so that the corresponding output bit of each PE is added to complete the result bit using LSBF method.

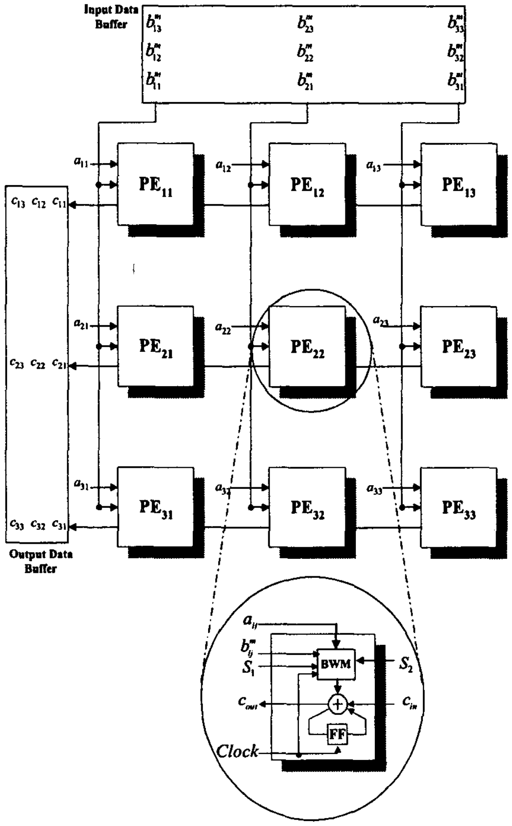

Figure .3. Systolic Architecture for implementation of matrix product (N=3)

0-7803-6488-0/00/$10.00 © 2000 IEEE

During the first eight cycles the three inner products $[C_{i1}]$ (i=1, 2, 3) are computed to produce the result coefficients using LSBF fashion. Then, during the second eight cycles the three inner products $[C_{i2}]$ are computed. Finally, the three inner products $[C_{i3}]$ are available in the output buffer in the end of the group of third eight cycles.

It is worth mentioning that the array produces three coefficients of the matrix *C* every eight clock cycles based on the multiple accumulate technique and therefore the entire computation can be carried out in *2nN* clock cycles and the area complexity of the structure is N^2 number of PEs. The throughput rate, the time complexity and the area of the structure are *N/2nT, 2nN* and N^2, respectively. (where T is the clock cycle fixed by the total gate delay of the BMW multiplier).
In the special case of matrix-vector multiplication, the matrix elements a_{ij} are fed from the north in a parallel/serial fashion bit by bit (LSBF) method while the vector elements b_j are fed in a parallel fashion and remain fixed in their corresponding *PE* cell during the entire computation of the operation (see Figure 4). Each bit of the final product of the *PE* is fed to the full adder of the preceding *PE* so that the corresponding output bit of each *PE* are added to give the bit of desired output LSBF. Following the same procedure for matrix-matrix multiplication with the same *PE* structure, each result coefficient is obtained after eight cycles as shown in Figure. 4

Features	Proposed Structure	Structure of [2]	Structures of [12] and [13]
Computation time	$(2nN)T$	$(2n)T$	$[N_{10}(4+n)+2n_1(1+n)+2n(N_1+N_2)+4N_{20}+N_2(2+n)-2n-21]T$
Area Complexity	$O(N)$	$O(N^2)$	$O(4N+n\ (N_1+N_2+N_1^2))$

Table 1. Comparison of proposed structure with the existing structures ([2],[12],[13]) for computation of matrix-vector product

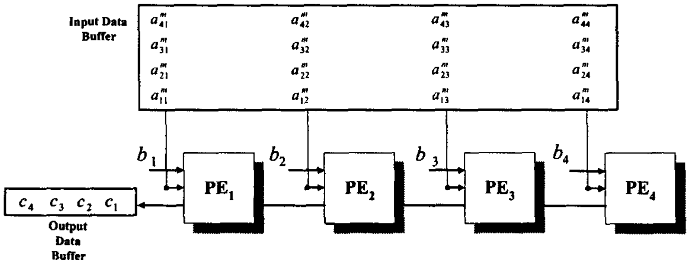

Figure 4. Proposed architecture for matrix-vector multiplication (N=4)

It is worth mentioning that the array produces one coefficient of the vector *C* every eight clock cycles based on the multiple accumulate technique and therefore the entire computation can be carried out *2nN* clock cycles and the area complexity of the structure is N number of *PEs*.

4. RESULTS AND ANALYSIS

The systolic architecture described above has been implemented using a Xilinx Virtex XCV1000E FPGA series board with Target Package: fg680. The design was carried out using Relative LoCations (RLOC) attributes in the View logic schematics to obtain efficient placement. The most relevant feature of the CLB in the Virtex-E FPGA is the dedicated carry logic to implement fast, efficient arithmetic functions. Dedicated carry logic provides fast arithmetic carry capability for high-speed arithmetic functions. The full adder structure suitable for the Virtex-E FPGA implementation is shown in Figure 5. The design is modular and can be implemented for larger matrices and word lengths. The circuit developed for matrix-matrix product has $O(N^2)$ and $O(2nN)$ area and time complexities, respectively. Figure 6 illustrates the performance obtained for the proposed architecture for the case of N=2,3,4,5,8 and n=4. In comparison of the architecture presented with similar structures [4], [5], the design has shown significant improvements when implemented on a single FPGA structure, requiring a single global clock and reduced numbers of hardware slices for the logic operations of PE.

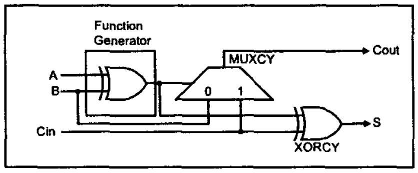

Figure 5. Full adder based Virtex-E FPGA
Virtex Carry Logic

In addition, each *PE* requires only five slices (four slices for the multiplier and one slice for the full adder and the flip-flop). Therefore, it is possible to predict the number of slices required to implement a matrix-matrix product given the size *N* of the matrix. If *PEs* is the number of slices required by each *PE*, the matrix product design requires N^2 *PEs*.

The circuit developed for matrix-vector multiplication has $O(N)$ and $O(2nN)$ area and time complexities, respectively. Figure 7 illustrates the influence of the matrix size and the performance obtained for the proposed architecture for the case of *N*=2,4,8,16,32 and *n*=4.

Using the same *PE* structure used in matrix-matrix product, it is possible to predict the number of slices required to perform a matrix-vector product given the size *N* of the matrix. If *PEs* is the number of slices required by each *PE*, the matrix product design requires N^2 *PEs*.

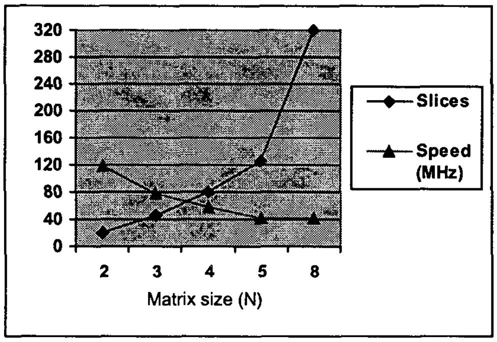

Figure 6. Design parameters for matrix-matrix product (n=4)

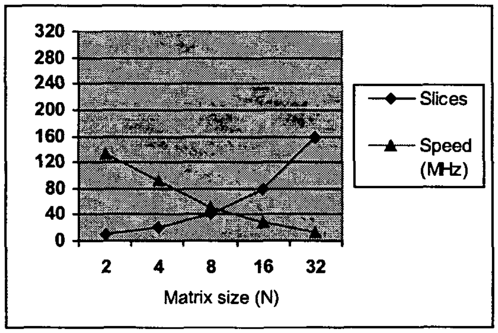

Figure 7. Design parameters for matrix-vector multiplication (n=4)

0-7803-6488-0/00/$10.00 © 2000 IEEE

5. CONCLUSION

The paper has presented a novel matrix multiplier, including the special case of matrix-vector multiplication. The underlying mathematical model has been presented and verified by the design and implementation of the algorithm on a Virtex series Xilinx FPGA. The proposed architecture requires less area, time complexities and less latency when compared with existing structures. In addition, it has a high throughput. Due to the importance of the different transforms in image and signal processing, and their dependence on matrix multiplication, the novel architecture developed can have a significant and valuable impact.

REFERENCES

[1] Keshab K. Parhi, "VLSI Digital Signal Processing Systems Design and Implementation." Wiley, 1999.

[2] SS. Nayak., PK. Meher, "High throughput VLSI implementation of discrete orthogonal transforms using bit-level vector-matrix multiplier." *IEEE Trans.on Circ.& Syst. II, Analog and Digital Sig. Proc.*, 1999, Vol.46, No.5, pp.655-658.

[3] S.Y. Kung, "VLSI Array Processors." Prentice Hall, 1988.

[4] P.L. Mills "The Design of Bit-Parallel Systolic Algorithms for Matrix -Vector and Matrix-Matrix Multiplication." *Proceedings of the 1985 ACM Computer Science Conference.*, 1985 March 12-14

[5] W.B. Ligon III, S. McMillan and al., "A re-evaluation of the practicality of floating-point operation on FPGAs." *IEEE Symposium on FPGAs for Custom Computing Machines,* April 15-17, 1998, pp.206-215.

[6] J.V. McCanny and J.C. White, "VLSI Technology and Design." Academic Press, 1987.

[7] Virtex^{TM}-E 1.8V Field programmable Gate Arrays, May 23, 2000 - Advance product specification -Xilinx documentation, URL: www.xilinx.com.

[8] A. Beaumont-Smith, M. Liebelt, C.C. Lim, K. To, "A digital signal multi-processor for matrix application." *14th Australian Microelectronics Conference (MICRO'97), Melbourne*, October, 1997.

[9] C. Baugh and B. Wooley, "A two's complement parallel array multiplication algorithm." *IEEE Trans. on Comput*ers, Vol.C-22, pp. 1045-1047, Dec. 1973.

[11] J. V . McCanny, J. G. McWhirter, and S. Y. Kung, "The use of data dependence graphs in the design of bit-level systolic arrays." *IEEE Trans. Acoust., Speech., Signal Processing*, vol. 18, pp. 787-793, May 1990

[12] R. M Owens and J. Ja'Ja', "A VLSI chip for the Winograd/prime-factor algorithm to compute the discrete Fourier transform." *IEEE Trans. Acoust., Speech., Signal Processing*, vol. ASSP-34, pp. 979-989, Aug 1986.

[13] C. Chakrabarti and J. Ja'Ja', "Systolic architectures for the computation of the discrete Hartley and the discrete cosine transforms based on prime-factor decomposition." *IEEE Trans. on Comput*ers, Vol.39, pp. 1359-1368, Nov. 1990.

0-7803-6488-0/00/$10.00 © 2000 IEEE