A High Throughput FPGA Implementation of A Bit-Level Matrix-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-matrix multiplication algorithm. The paper describes the mathematical model for the algorithm (based on Baugh-Wooley algorithm), the associated design and implementation of the algorithm on a Xilinx FPGA board, and discusses the efficiency of the implementation requiring $O(N^2)$ and $O(2nN)$ as area and time complexities respectively, where N is the matrix size and n is the word length.**

I. INTRODUCTION

In today's rapidly changing world, designers and researchers continuously demand significant increases in computer performance. 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 are frequently occurring 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 computer 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 massive parallelism.

To achieve this, the algorithms have been modified and adapted to operate in modular and regular fashion. In this approach, a serial-parallel matrix-matrix multiplier based on the Baugh-Wooley algorithm has been selected because of the algorithm has the following features [1], [2]:

- Simplicity, regularity and modularity of structure;
- Flexible implementation for any desired matrix product size and
- Massive bit-level processing concurrency to achieve ultrahigh throughput rate.

Field Programmable Gate Arrays (FPGAs) offer the possibility that re-configurable arrays can be constructed to efficiently compute certain problems. Generally, FPGA technology is excellent for building systolic array-style processors [10]. The systolic architecture 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 Field-Programmable Gate Array;
- Flexible architecture that balances speed and density; and
- Built-in clock- management circuitry

The composition of the rest of the paper is as follows. The algorithm for a serial-parallel matrix-matrix multiplier is given in section II. Section III is concerned with the systolic architecture of matrix/matrix multiplier. Section IV gives the design characterisation. Concluding remarks are given in section V.

II. 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

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

Proc. 43rd IEEE Midwest Symp. on Circuits and Systems, Lansing MI, Aug 8-11, 2000
0-7803-6475-9/00/$10.00 ©IEEE 2000

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 2's complement number representation, then

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

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 the multiplier used. As mentioned in the introduction a Baugh-Wooley multiplier algorithm [1], [9] was chosen and using this algorithm, equation (4) becomes:

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

which can be re-expressed as follows

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

From equation (6) it can be seen that the multiplication of A_{ik} and B_{kj} expressed in two's complement representation can be written in a form which involves only positive bit products. Therefore, the product of two matrices can be computed by a systolic architecture.

III. DESIGN IMPLEMENTATION OF MATRIX PRODUCT

A. Proposed Serial-Parallel Baugh-Wooley multiplier

The multiplication algorithm for a word length n=4 is illustrated by the multiplication table shown in Fig. 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.

The proposed two's complement serial-parallel multiplier comprises a logic unit and an adder unit shown in Fig. 2 (for n=4). The logic unit consists of three AND gates, one NAND gate, four XOR gates, and OR gate. The output of the multiplier is obtained from the adder unit using the Carry-Save and Add-Shift (CSAS) technique. S1 and S2 are two control signals. In the first three and last four clock cycles S1=0 while in the fourth clock cycle S1=1. S2=1 in the eighth clock cycle and S2=0 in the previous seven clock cycles.
The extra "1" (required by the Baugh-Wooley algorithm) for the fifth column is provided to the right most adder through an OR gate using the delayed signal S1. The extra "1" for the eighth column is provided to the left most adder with help of S2 through a multiplexer.

				b_{kj}^{3} a_{ik}^{3}	b_{kj}^{2} a_{ik}^{2}	b_{kj}^{1} a_{ik}^{1}	b_{kj}^{0} 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

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

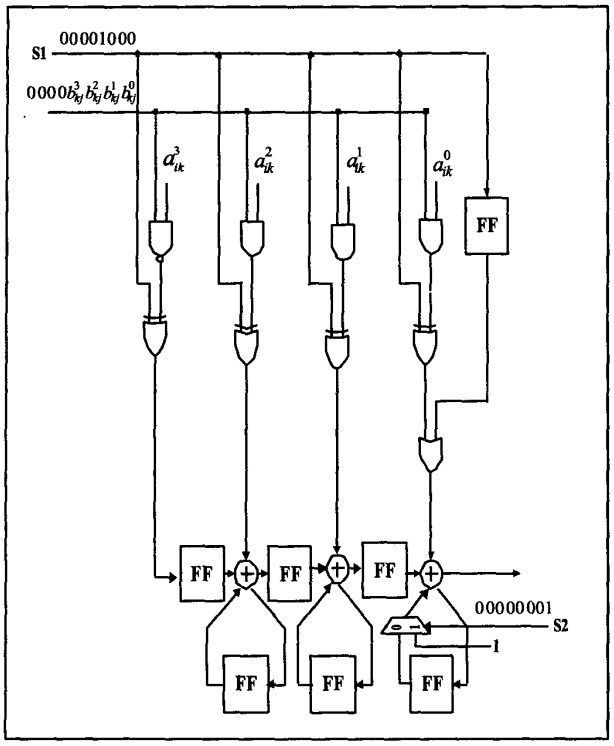

Fig .2. Serial - Parallel Baugh Wooley Multiplier -BWM -(n=4).

B. Proposed Bit- Level Systolic arrays Architecture

Equation (6) can be mapped into the proposed architecture. Fig. 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 Fig. 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 LSB 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 bits of each PE are added to complete the result bits using LSBF method.

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 the 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 $2nN$ clock cycles and the area complexity of the structure is N^2 number of PEs. The throughput rate, time complexity and 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.

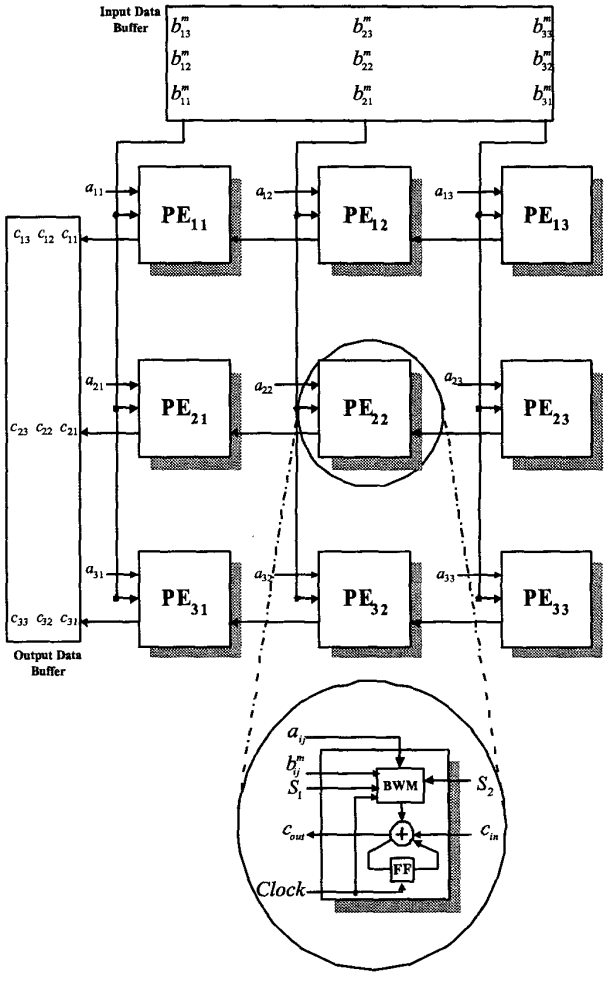

Fig .3. Systolic Architecture for implementation of matrix product (N=3)

IV. 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 is modular and can be implemented for larger matrices and word lengths. The design has $O(N^2)$ and $O(2nN)$ area and time complexities, respectively. Fig. 4 illustrates the performance obtained for the proposed architecture for the case of N=2,3,4,5,8 and n=4. In comparison with other approaches [4], [5], the design shows significant improvements requiring a single, global clock and reduced numbers of hardware slices for the logic operations of PE.

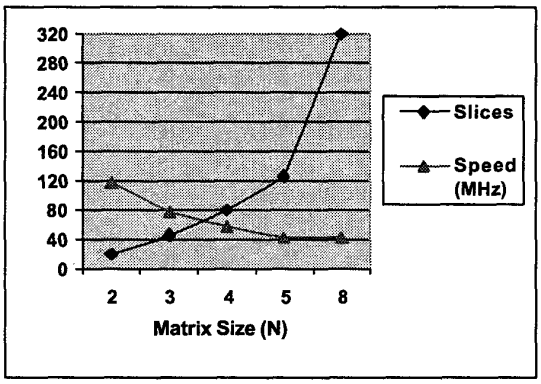

Fig 4. Design parameters for the design implementation (n=4). Matrix size influence

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 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*.

V. CONCLUSION

The paper has presented a novel matrix-matrix multiplier for using in signal/image processing applications. 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 and time complexities and less latency compared with existing structures. Apart from this, 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™-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 Compute*rs, Vol.C-22, pp. 1045-1047, Dec. 1973.

[10] I. M. Bland and G.M. Mergson, "The Systolic Array Genetic Algorithm, An Example of systolic Arrays as a Reconfigurable Design Methodology." *IEEE Symposium on FPGAs for Custom Computing Machines*, April 15-17, 1998, pp.260-261.

[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