International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Manipulation of Adaptive Matrix Using VHDL-High Performance Field Programmable Gate Arrays

Mr.Rounak R. Gupta, Prof. Atul S. Joshi

Department of Electronics and Telecommunication Engineering,
Sipna College of Engineering and Technology,
S.G.B.Amravati University
Amravati,444603, India

Department of Electronics and Telecommunication Engineering,
Sipna College of Engineering and Technology,
S.G.B.Amravati University
Amravati,444603, India

ABSTRACT

This research consist of a novel field programmable gate array (FPGA) based reconfigurable coprocessor board being used to evaluate hardware architecture for speedup of matrix computation using high speed field programmable array. This paper have been developed using a mathematics of array (MOA) and are optimal in the sense they reduce normal complexity of calculation to a series of primitive additions and offsets based on different matrix operation. The method simple yet powerful, produces right and fast result which are

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

correct, and are independent of dimensionality. Software implementation has been used to provide speedups on the order of 100% to the solution of matrix on a coprocessor. We can extend this method to application design for the coprocessor.

Keywords: Coprocessor, FPGA, Matrix Computation, MOA, VLSI.

1. INTRODUCTION:

Many numerical problems in a real life applications such as engineering, scientific computing and economics use huge matrices with non zero elements, referred to as matrices. As there is no reason to store and operate on a huge number of zeros, it is often necessary to modify the existing algorithm to take advantage of the structure of the matrix. Matrix can be easily compressed, yielding significant savings in memory usage. During the working of the PC processor have to deal with the number of operations like addition, subtraction, multiplication and many more. Above are normal operations but operations like matrix calculation, summation etc. puts more load on the processor which also affects the processor speed.

Also there is a coprocessor attached to main processor where this operation are performed and analyzed. So that individual operation can be performed and system performance can be validated. Matrix manipulation is a computation intensive operation and plays a very important role in many scientific and engineering applications. For high performance applications, this operation must be realized in hardware. This paper presents a arithmetic logic unit (ALU) using field programmable gate array (FPGA). This proposed architecture employs advanced design techniques and exploits architectural features of FPGA.

2. Review of Prior Feature Technique:

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Mencer implemented matrix multiplication on the Xilinx XC4000E FPGA device. Their design employs bit-serial MACs using Booth encoding. They focused on tradeoffs between area and maximum running frequency with parameterized circuit generators. For the specific example of 4 4 matrix multiplication, 954 CLBs are used to achieve a maximum running frequency of 33MHz.

Amira improved the design in using the Xilinx XCV1000E FPGA device. Their design uses modified Booth-encoder multiplication along with Wallace tree addition. The emphasis was once again on maximizing the running frequency. For the specific example of 4 4 matrix multiplication,296 CLBs are used to achieve a maximum running frequency of 60MHz. Area/speed or, equivalently, the number of CLBs divided by the maximum running frequency was used as a performance metric.

Kumar and Tsai achieved the theoretical lower bound for latency for matrix multiplication with a linear systolic design. They provide tradeoffs between the number of registers and the latency. Their work focused on reducing the leading coefficient for the time complexity.

The limitations with all this methods is that, the matrix operations is that it took large time and memory to deal with this complex type of operations. This paper have been developed using a Mathematics of Arrays (MOA) and are optimal in the sense that they reduce normal complexity of calculation operations to a series of primitive additions and offsets based on different matrix operations. The method is simple yet powerful, produces right and fast result which are correct, and are independent of dimensionality.

3. System Building Blocks:

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

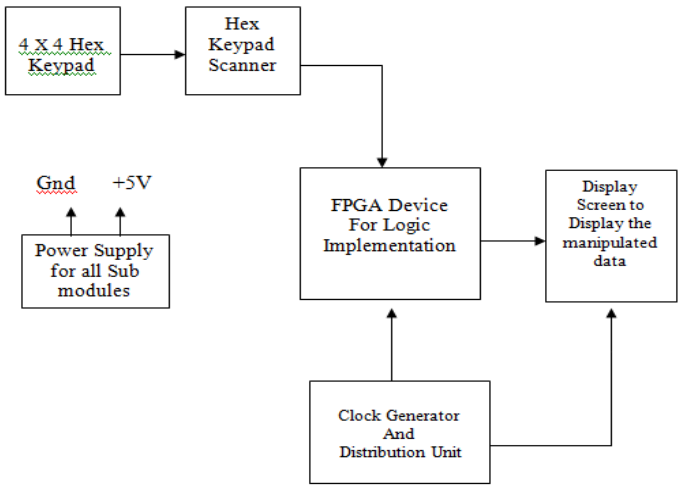


Fig.1 Basic Building Blocks

Matrix manipulation can be best implemented around the high computing programmable gate array. Fig.1 shows the projected building blocks for matrix manipulation around PGA. System can be realized using different platforms like parallel processing, advanced computing and so on, out of this possibilities best suitable means is the VLSI platform. Because of starling features of this new trend that makes it more suitable for the said application.

VLSI platform supports range of devices that can be used but out of the supported range XC2S200 becomes best means because of the exceptional features like 16-bit LUT RAM, In-System Programming (ISP), Boundary scan and Read back ability Fully 3.3V PCI compliant to 64 bits at 66 MHz and Card Bus compliant, Low-power segmented routing architecture, Dedicated carry logic for high-speed arithmetic, etc.

FPGA:

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Field Programmable Gate Array is basically collection of Configurable Logic Block (CLB). CLBs can be designed using PLA architecture. PLA is Programmable Logic Array. PLA when designed with multiplexers and flip-flops it gives choice between combinational logic and sequential logic to be implemented. Such architecture is decoder architecture and final logic is implemented using Look Up Table (LUT). LUT is purely combinational circuit that makes FPGA volatile device and hence to keep the logic intact additional Platform Flash memory is required. On each power up the logic is booted up to the FPGA from the Platform Flash.

4. Proposed Flow Diagram:

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

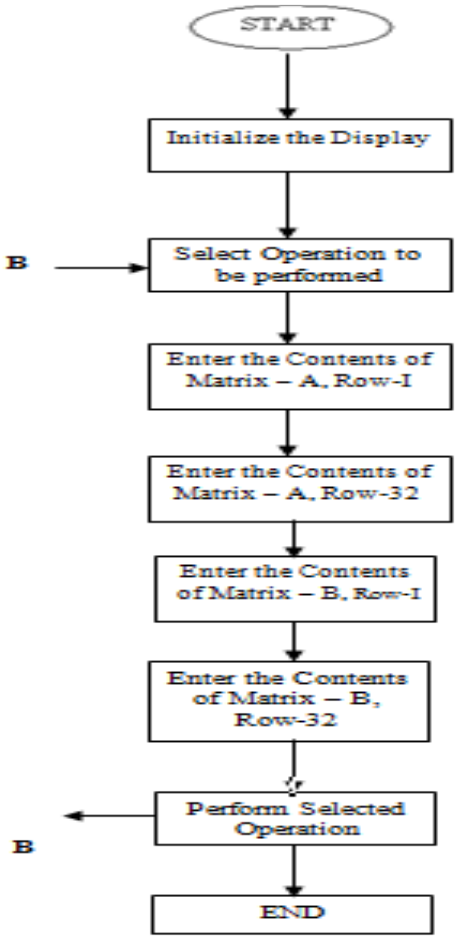


Fig.2 Flow Chart

Let A and *B* be the two input matrices, *C*= the output of two Matrices A and B. The whole matrix coprocessor architecture is depicted in Fig.1. The new coprocessor runs as follows. Matrix *A* is entirely transmitted from the data memory of the host processor to the coprocessor and it is stored into the register file. At each step of the computation, an

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

element of matrix B is transferred from the data memory into the coprocessor register file, following a column order. Through a Variable-Module-Counter the appropriate elements in the proper row *of matrix A* are selected, which execute their multiply-accumulate operations. After some steps, it will give the final result matrix *C i*s available. This column is stored into the result buff*er* and then sequentially transferred to the processor through the bus interface. The coprocessor architecture presented here performs matrix multiplications, and an input/output band cycle. In fact, just one element can be transferred from/to the host processor to/from the coprocessor at a time. The proposed circuit is able to perform a matrix multiplication on *n* input matrices. The same architecture can be interfaced using two separate communication channels that can be used in parallel to have an input band and a separate output band.

5. Observation:

```
when 0 =>
        data <= "00111000";  -- "38"

        count <= 0;
        rs <= '0';
        lcd_state <= 1;

when 1 =>
        en <= '0';

        count <= 0;
        lcd_state <= 2;
```

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

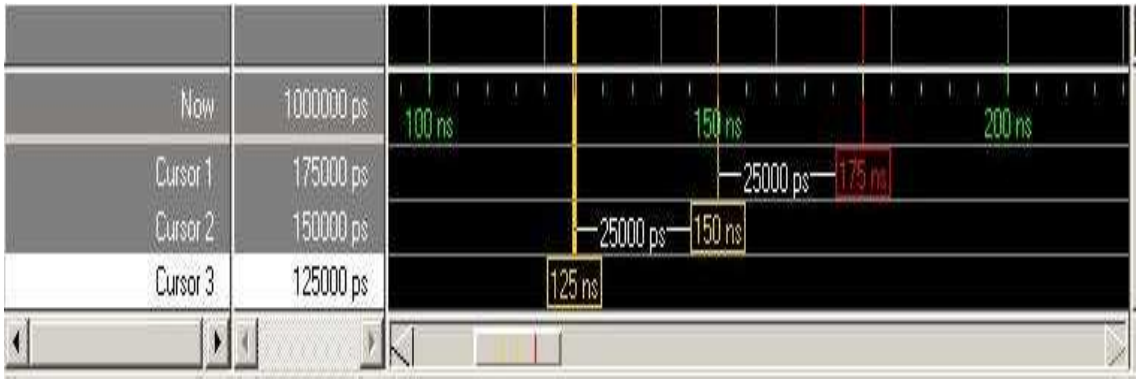
```
when 2 =>
        data <= "00001100";  -- "0C"

        en <= '1';
        count <= 0;
        lcd_state <= 3;

when 3 =>
        en <= '0';

        count <= 0;
        lcd_state <=4;

when 4 =>
        data <= "00000110";  --  "06"

        en <= '1';
        count <= 0;
        lcd_state <= 5;
```

International Journal of Enterprise Computing and Business Systems

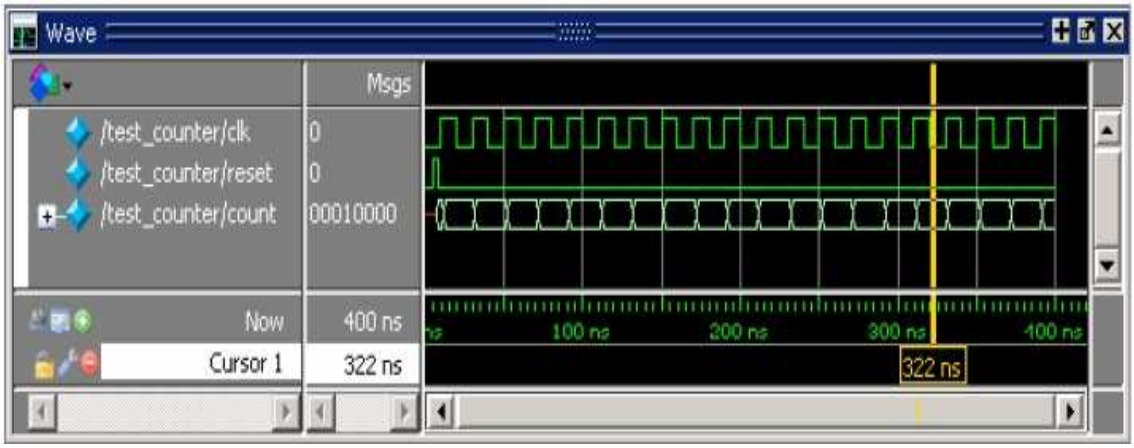
ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012



Waveform 1



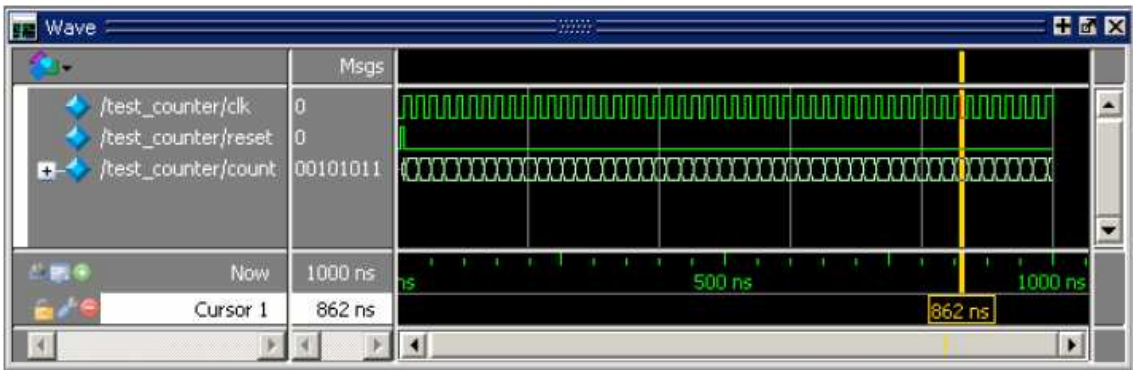
Waveform 2

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012



Waveform 3

The architecture presented can be easily scaled and dimensioned for different requirements, and it can be integrated with different processors with few interface changes.

TABLE: COMPARISON RESULT

Matrix Dimension N	Coprocessor Solution Execution Time(μs)	Software Solution Without Coprocessor Execution Time(μs)	Speed-up
4	4.22	34.3	8.10
8	5.68	53.12	9.13
16	7.38	78.24	10.20
32	14.01	109.14	7.87

new architecture exhibits a running frequency of 143 *MHz* and it occupies much less area (in terms *of* CLBs) than [3], [4], and *[5]*.It operates with the minimum number of clock cycles.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Finally, the circuit described in [3] supports a restricted *I/0* band from/to the storage block. Table shows that the proposed matrix multiplier offers the best area-time trade-off. Thus, It appears as the optimum design solution for the integration with the target coprocessors. The complete solution of matrix processor-coprocessor system here presented can *be* easily extended with further matrix coprocessor function implementations.

6. Acknowledgment:

Thanks to department of Electronics and Telecommunication of SIPNA college of engineering and technology, Amravati.

7. Conclusion:

Operations on matrices are important part in computational science. In this paper we proposed a new matrix operation and explained the expected benefits for the hardware execution unit. We suggest that at expenses of an extra bit per value representation when compared to existing techniques that the design and speed-up of matrix operations could be facilitated.

8. References:

[1] M. deLorimier and A. DeHon. Floating-point sparse matrixvector multiply for fpga. In Proceedings of the International Symposium on Field-Programmable Gate Arrays, February 2005.

[2] A. Amirq, F. Bensaali, *"An* FPGA parameterisable system for matrix product Implementation", *Proc. of The IEEE* Workshop on *Signal Processing Systems* Design *and Implementation, 2002.* pp. *75-79,* California ,USA.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

[3] J. Jang, S. Choi, V. K. Prasanna, "Area and Time Efficient Implementations of Matrix Multiplication on FPGAs", *Field-Programmable Technology (FPTZUOZ),* IEEE International Conference 2002, Hong Kong.

[4] 0. *Mencer,* M. Morf and M. Flynn, "PAM-Elox: *High* Performance FPGA Design for Adaptive Computing", *IEEE Symposium on FPGAs for Custom.*

[5] A. Amira, A. Bouridane and P. Milligan, "Accelerating Matrix Product on Reconfigurable Hardware for Signal Processing", *Field-Programmable logic and Application (FPL),* 2001, pp. 101-111.

[6] J. Dongarra. Sparse matrix storage formats. In Templates for the Solution of Algebraic Eigen value Problems: A Practical Guide, SIAM, 2000.

[7] Y. Dou, S. Vassiliadis, G. Kuzmanov, and G. N. Gaydadjiev. 64-bit floating-point fpga matrix multiplication.In Proceedings of the International Symposium on Field-Programmable Gate Arrays, February 2005.

[8] P. Stathis, S. D. Cotofana, and S. Vassiliadis. Sparse matrix vector multiplication evaluation using the bbcs scheme. In Proc. of 8th Panhellenic Conference on Informatics, pages 40–49, November 2001.

[9] S. Vassiliadis, S. D. Cotofana, and P. Stathis. Vector isa extension for sprase matrix multiplication. In Proceedings of EuroPar'99 Parallel Processing, pages 708–715, September 1999.

[10] S. Vassiliadis, S. D. Cotofana, and P. Stathis. Bbcs based sparse matrix-vector multiplication: initial evaluation. In Proc. 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, pages 1–6, August 2000.

[11] S. Vassiliadis, S. D. Cotofana, and P. Stathis. Block based compression storage expected performance. In Proc. 14th Int. Conf. on High Performance Computing Systems and Applications (HPCS 2000), June 2000.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

[12] Y.Saad. Krylov subspace methods on supercomputers. In SIAM Journal on Scientific and Statistical Computing, 1989.

[13] J. D. Z. Bai, J. Dongarra, A. Ruhe, and H. van der Vorst. Templates for the solution of algebraic eigenvalue problems: A practical guide. In Society for Industrial and Applied Mathematics, 2000.

[14] L. Zhuo and V. K. Prasanna. Sparse matrix-vector multiplication on fpgas. In FPGA '05: Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays, pages 63–74, New York, NY, USA, 2005. ACM Press.

9. Author Information:

Mr. Rounak R. Gupta is currently working as a lecturer in Electronics and Telecommunication Engineering, Sipna College of Engineering, Amravati (India), since 2011. He is also pursuing his M.E. in Digital Electronics from the same institute. His areas of interest are Digital Image Processing, VLSI Design and Digital Signal Processing.

Prof. Atul S. Joshi is currently working as Professor in Electronics and Telecommunication Engineering Department, Sipna College of Engineering, Amravati (India) since inception. He is also perusing his Ph.D. in the faculty of Electronics Engineering From SGB Amravati

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

University, Amravati (India). His areas of interest are Communication networks, Electronics Circuits and Communication Engineering.