Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication

Mahendra Vucha
Research Scholar, MANIT, Bhopal and
Asst. Prof, Dept. EC, GGITM
Bhopal, India.

Arvind Rajawat
Associate Professor, Dept. EC, MANIT,
Bhopal, India.

ABSTRACT
The evolution of computer and Internet has brought demand for powerful and high speed data processing, but in such complex environment fewer methods can provide perfect solution. To handle above addressed issue, parallel computing is proposed as a solution to the contradiction. This paper provides solution for the addressed issues of demand for high speed data processing. This paper demonstrates an effective design for the Matrix Multiplication using Systolic Architecture on Reconfigurable Systems (RS) like Field Programmable Gate Arrays (FPGAs). Here, the systolic architecture increases the computing speed by combining the concept of parallel processing and pipelining into a single concept. Here, the RTL code is written for matrix multiplication with systolic architecture and matrix multiplication without systolic architecture in Verilog HDL, compiled and simulated by using Modelsim XE III 6.4b, Synthesized by using Xilinx ISE 9.2i and targeted to the device xc3s500e-5-ft256 and then finally the designs are compared to each other to evaluate the performance of proposed architecture. The proposed Matrix Multiplication with systolic architecture is enhances the speed of matrix multiplication by twice of conventional method.

Keywords
Systolic Array Architecture, Processing Element, Data Processing Unit, Reconfigurable Systems.

1. INTRODUCTION
In computer architecture, a systolic architecture is a pipelined network arrangement of Processing Elements (PEs) called cells. It is a specialized form of parallel computing, where cells compute the data which is coming as input and store them independently. A systolic architecture is an array composed of matrix-like rows of cells. Here, the Processing Elements is similar to central processing units (CPUs) (except for the usual lack of a program counter, instruction register, control unit etc. since operation is transport-triggered, i.e., sensitive to arrival of a data object across it). Each cell shares the information with its neighbors immediately after processing. The systolic array is often rectangular where data flows across the array between neighbor Data Processing Units (DPUs), often with different data flowing in different directions. Systolic architecture is arrays of DPUs which are connected to a small number of nearest neighbor DPUs in a mesh-like topology. DPUs perform a sequence of operations on data that flows between them. In this research, DPU performs the Multiplication and Accumulation (MAC) and the systolic array concept is used for multiply the matrices to enhance its computation speed.

2. LITERATURE REVIEW
The various Systolic architecture represented in [2, 3, 4, 5, 7] are shown bellow.

2.1. AB1 architecture
AB1 architecture is 1-D systolic array shown in Figure 1 has size of a block used for Block matching algorithm [3]. Consecutive computation of all $(2p + 1)^2$ candidate blocks per displacement vector may provide N $(2p + 1)^2$ time instances as can be seen of the input data indexes in Fig.1, where p represents the maximum displacement assumed and N is order of matrix. Computation of consecutive candidate blocks implies the replacement of one input data column by another. A regular data flow at the end of each candidate block line within a search area requires a continuous exchange of columns of input data, such that N-1 dummy time instances with invalid data at the output of AB1 occur.

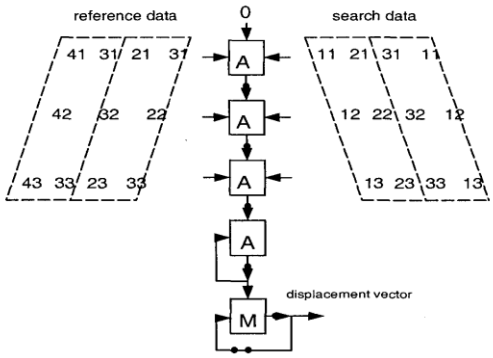


Figure 1 AB1 architecture

2.2. AS2 architecture
An alternative procedure in [2] is the decomposition of the algorithm into two subparts where the first is defined over a three-dimensional index space spawn by the indexes *i, k,* and *m.* The best matching candidate block is searched along a line of candidate blocks indexed by *m* within the search range. The second part of the algorithm is defined over a one-dimensional index space along the index n**.** previously determined minima of all search area lines are compared and the smallest denotes the displacement vector component shown in Figure. 2. Consecutive computation of search area lines with a regular exchange of input data requires C_{AS2} time instances.

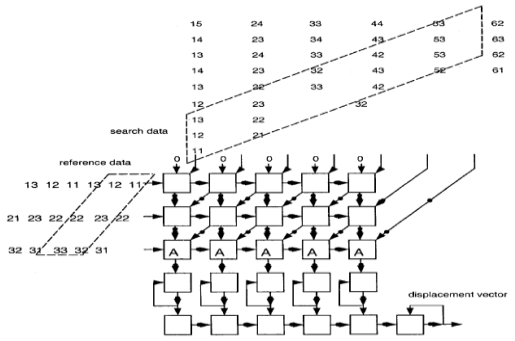


Figure 2. AS2 architecture

2.3. AB2 architecture

AB2 architecture [3] is, 2-D version of AB1, shown in Figure 3. Reference block data x(i, k) is loaded and remains fixed in the AD nodes. The input data flow of y(i + rn, k + n) permits sequential computation of consecutive search area lines. The computation of displacement vector takes C_{AB2} time instances.

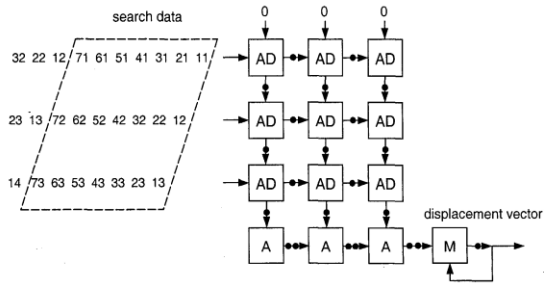


Figure 3. AB2 architecture

2.4. AS1 architecture

Architecture AS1 shown in Figure.4 is very simple.

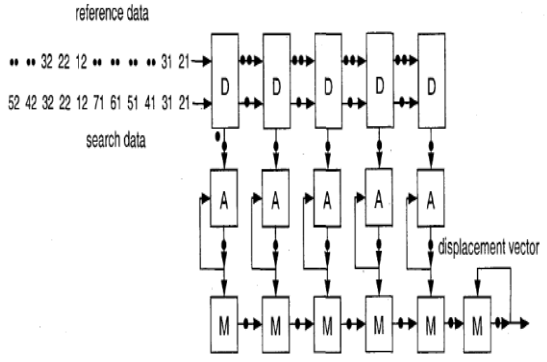


Figure 4. AS1 architecture

This systolic array in [8] used for Full Search Block Matching Algorithm, requires only sequential data input. Dummy data (denoted by dots) are inserted into reference area data stream. In result time instances are required to compute a motion vector. The processing speed can be improved if data sequences of adjacent search areas are mixed. This implies the storage of intermediate results in multiple registers or in a memory instead of the accumulators A.

3. PROPOSED ARCHITECTURE

The Parallel Matrix Multiplication [7] has many different identifications, but all with the similar implementation. That is, they immediately multiplex a pair of matrix elements in special. Parallel Matrix Multiplication on Systolic Array (PMMSA) uses this approach. In [5], PMMSA is characterized by processing data input in pipeline and comprised of regularly arrayed PE. Where neighbor PEs are connected with each other by shortest line and therefore mass data has no need to be stored before processing. Decrease of distance between the PEs in an array greatly reduces the internal communication delay and improves the utility of processing units. It also removes time consumption for controlling the establishment of data stream. In, this research, the PE is replaced with Multiplication and Accumulation (MAC) to enhance the speed and reduce the complexity of Systolic Architecture.

The algorithm for the matrix multiplication of order N×N is shown bellow.

1. For I = 1 to N → Start of for loop 1
2. For J = 1 to N → Start of for loop 2
3. For K = 1 to N → Start of for loop 3
4. C[I,J] = C[I,J] + A[J,K] * B[K,J]

   → Computation of Matrix Multiplication and it will be implemented by using systolic array
5. End → End of for loop1
6. End → End of for loop2
7. End → End of for loop3

The above algorithm can be implemented in two methods

1. Conventional method (with out Pipeline and Parallel Processing)
2. Systolic Architecture (Pipeline and Parallel Processing)

4. IMPLEMENTATION SCHEME

In this paper, we aim to compute the equation (1) with a two dimensional systolic array.

$$C_{m\times n} = A_{m\times k} \times B_{k\times n} \text{ ----- (1)}$$

Where A, B and C are the matrices with order $m \times k$, $k \times n$ and $m \times n$ respectively. Each PE of systolic array computes the multiplication of elements and accumulates to the corresponding element and then elements will be passed to neighbor PE in the systolic array. First elements $a_{i,j}$ in row i of matrix A are injected first into PE as pipeline with the sequence of $a_{i,k}$ and the input time to the element of $a_{i+1,j}$ is one time unit later than $a_{i,j}$. Similarly, elements $b_{i,j}$ in column j of matrix B are injected first into PE as pipeline with the sequence of $b_{k,j}$ and the input time to the element of the sequence of $b_{k,j+1}$ is one

time unit later than $b_{k,j}$. The architecture of PE in this approach is shown in figure 5 which performs the Multiplication and Accumulation on data.

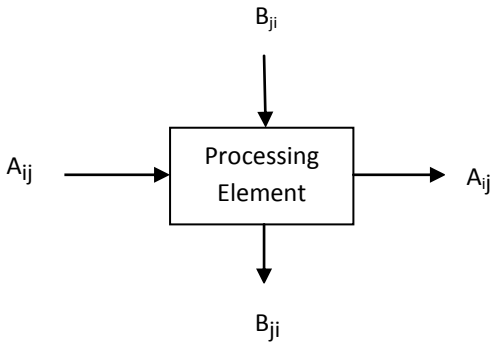


Figure 5. PE of Systolic Architecture

4.1. Systolic Array Architecture for Matrix Multiplication

A systolic architecture is an arrangement of processors i.e. PEs in an array (AB2 Architecture in [3]) where data flows synchronously across the array between neighbors, usually with different data flowing in different directions. PE at each step takes input data from one or more neighbors (e.g. Left and Top), processes it and, in the next step, outputs results in the opposite direction (Right and Bottom). The Proposed two dimensional systolic Architecture is given in the Figure 6.

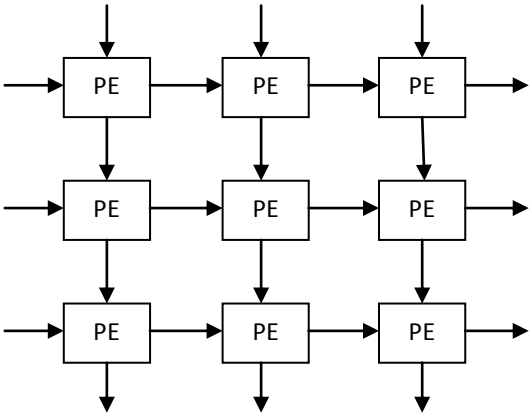
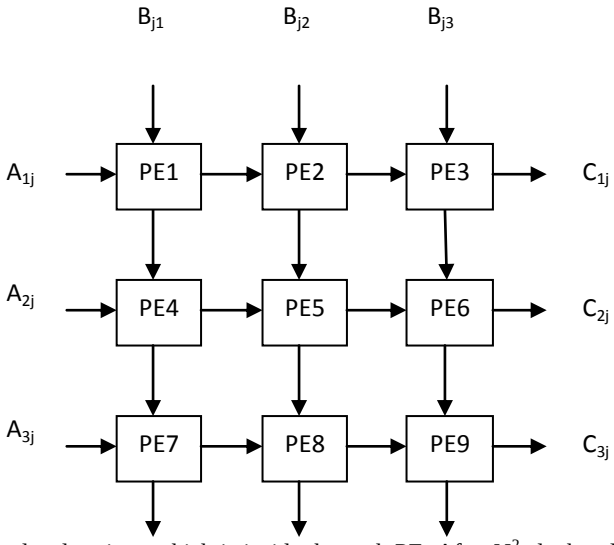


Figure 6. Two-dimensional Systolic Array

The array architecture given above takes input data in parallel into first PEs in the array and processes the Multiplication and Accumulation on them and then outputs result to the next level PEs of array. Systolic arrays do not lost their speed due to their connection like any other parallelism. Where, each cell (PE) is an independent Processor (CPU) and has its own registers and Arithmetic and Logic Units (ALUs) i.e. Multiplication and Accumulation unit. The cells share the information with their neighbors, after performing the necessary operations on the data.

Systolic Array Architecture (SAA) for Matrix Multiplication is shown in the Figure 7. Where each cell takes inputs from left and top, multiplies them and accumulates in the



local register which is inside the each PE. After N^2 clock pulses the result would be stored in each PE. The proposed systolic array architecture needs N^2 magnitude Multipliers, 2N magnitude Accumulators and 4N registers are needed to compute matrix multiplication where N is order of matrix.

Figure 7. Systolic Architecture for Matrix Multiplication

5. RESULTS & DISCUSSION

The implementation of Matrix Multiplication is done in both methods i.e. Conventional and Systolic Architecture, as described above, on FPGA. The RTL code is written in Verilog HDL, verification of logic and simulation is done by ModelSim XE 6.4b. The simulation results have given that, the Systolic architecture implementation requires less number of clock cycles then Conventional method and is shown in Figure 8. The simulation results in Figure 8, exposes the parallel processing and pipelining by the systolic array architecture and also the input and output matrices $A_{3\times3}$, $B_{3\times3}$ and $C_{3\times3}$ respectively where the matrix elements are of 4 bit each. After simulation, the design is passed for synthesis onto the platform XILINX ISE 9.2i to convert RTL logic into gate level netlist and also the schematic diagram is captured. The schematic diagrams are shown in Figure 9 and Figure 10. The Figure 9 represents the top level hierarchy of design and the Figure 10 shows internal hierarchy of top level schematic.

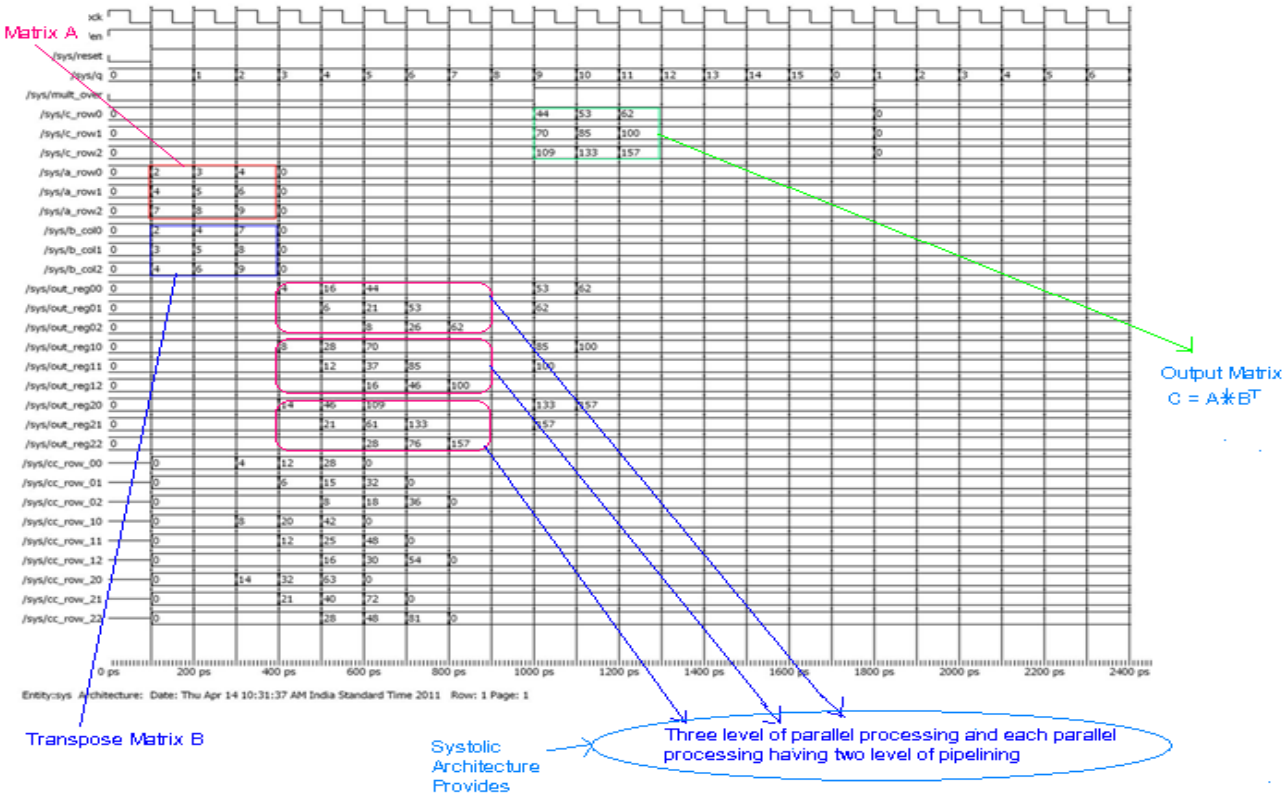


Figure 8. Simulation wave form of Systolic Array Architecture for Matrix Multiplication

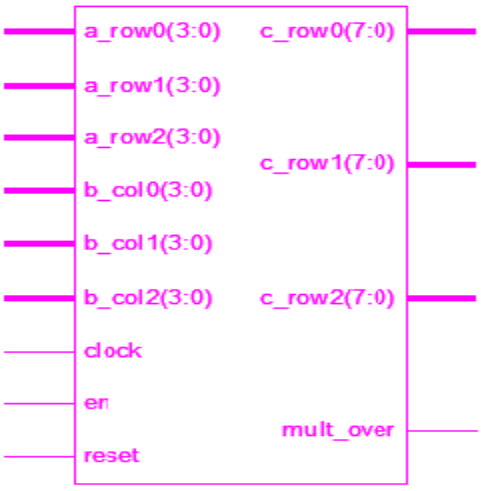


Figure 9. Schematic of Top level hierarchy

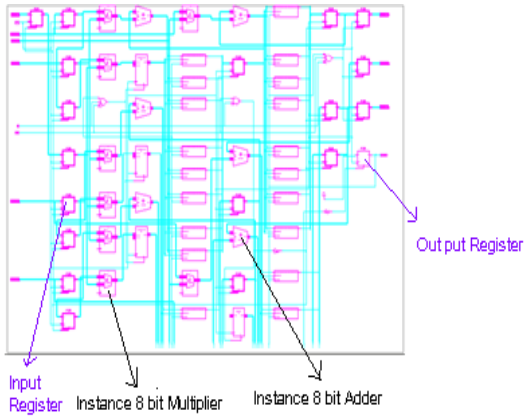


Figure 10. Schematic of low level hierarchy

The both designs Conventional method and Systolic Architecture for Matrix Multiplication are targeted to the device xc3s500e-5-ft256 and the synthesis report of the designs provides the gate level netlist with critical path delay between input and output. The critical path delay represents the core speed of the design. The brief summary of synthesis report is exposed in Table 1. From the Table 1, it is noticed that the core speed of Systolic Array Architecture for matrix multiplication is 210.2MHz which is more than two times of conventional method 101.7MHz.

Table 1. Performance evaluation of Systolic Array architecture for Matrix Multiplication

S.No	Name of Component	Number of components used	
		Conventional Method	Systolic Array Architecture
1	Critical path delay	9.831ns	4.757ns
2	4x4-bit registered multiplier	27	9
3	8-bit adder	18	6
4	4-bit up counter	0	1
5	8-bit up accumulator	0	3
6	8-bit register	72	34

6. CONCLUSION

The Systolic Array Architecture is designed for Matrix Multiplication and it is targeted to the Field Programmable Gate Array device xc3s500e-5-ft256. The parallel processing and pipelining is introduced into the proposed systolic architecture to enhance the speed and reduce the complexity of the Matrix Multiplier. The proposed design is simulated, synthesized, implemented on FPGA device xc3s500e-5-ft256 and it has given the core speed 210.2MHz.

7. REFERENCES

[1] H. T. Kung "*Why systolic architectures?*," IEEE Computer, vol. 15, pp. 37, Jan. 1982.

[2] Sung Burn Pan, Seung Soo Chae and Rae-Hong Park, VLSI Architecture for Block Matching Algorithms using Systolic Array, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 1, February 1996.

[3] Kuan-i Lee, Algorith and VLSI architecture design for H.264/AVC Inter Frame Coding, A PhD Thesis at National Cheng Kung University, Tainan, Taiwan, in 2007.

[4] Doru Florin Chiper, M. N. S. Swamy, M. Ohmair Ahmad, and Thanos Stouraitis, A Systolic Array Architecture for the Discrete Cosine Transform, IEEE Transactions on Signal Processig, Vol. 50, no. 9, September, 2002.

[5] Ganapathi Hegde, Cyril Prasanna Raj P and P.R.Vaya, Implementation of Systolic Array Architecture for Full Search Block Matching Algorithm on FPGA, European Journal of Scientific Research, Vol.33 No.4 (2009), pp.606-616.

[6] Chien-Min Ou, Chian-Feng Le and Wen-Jyi Hwang, An Efficient VLSI Architecture for H.264 Variable Block Size Motion Estimation, IEEE Transactions on Consumer Electronics, Vol. 51, No. 4, NOVEMBER 2005.

[7] Feifei Dong, Sihan Zhang and Cheng Chen, Improved Design and Analyse of Parallel Matrix Multiplication on Systolic Array Matrix, IEEE, 2009.

[8] Ziad Al-Qadi and and Musbah Aqel, erformance Analysis of Parallel Matrix Multiplication Algorithms Used in Image Processing, World Applied Sciences Journal 6 (1): 45-52, 2009.

[9] Mohammad Mahdi Azadfar, Implementation of A Optimized Systolic Array Architecture for FSBMA using FPGA for Real-time Applications, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008.

8. AUTHORS PROFILE

Mahendra Vucha received his B. Tech in Electronics & Communication Engineering from JNTU, Hyderabad in 2007 and M. Tech degree in VLSI and Embedded System Design from MANIT, Bhopal in 2009. He is currently working for his Ph. D degree at MANIT and also working as Asst. Prof in Gyan Ganga Institute of Tech & Mgmt, Dept. of Electronics and Communication Engineering, Bhopal (M.P), India. His areas of interest are Hardware Software Co-Design, Analog Circuit design, Digital System Design and Embedded System Design.

Arvind Rajawat received his B. Tech in Electronics & Communication Engineering from Govt. Engineering College in 1989, M. Tech degree in Computer Science Engineering from SGSITS, Indore in 1991 and Ph. D degree from MANIT, Bhopal. He is currently working as Associate professor in Dept. of Electronics and Communication Engineering, MANIT, Bhopal (M.P), India. His areas of interest are Hardware Software Co-Design, Embedded System Design and Digital System Design.