A Review on Design and Implement EfficientSystolic Array Architecture Multiplier

1st Aryan Ajit Tiwary
*Electronics and Telecommunication Pune Institute of Computer Technology*Pune, India

2nd Arya Raghuveer Mane *Electronics and Telecommunication Pune Institute of Computer Technology*Pune, India

3rd Akshay Dharmesh Lende
Electronics and Telecommunication Pune Institute of Computer Technology
Pune, India

4th Mr.N.G Nirmal
Electronics and Telecommunication Pune Institute of Computer Technology
Pune, India

***Abstract*—In this paper, the designing of Systolic array architecture multiplier is provided. The evolution of computers and the Internet has brought demand for powerful and high-speed data processing, but in such a complex environment fewer method can provide perfect solutions. To handle the above addressed issue, parallel computing is proposed as a solution. This paper demonstrates an effective design for the Matrix Multiplication using Systolic Architecture on Reconfigurable Systems (RS) like Field Programmable Gate Arrays (FPGAs). The relevant topics and literature regarding the elements in a systolic array architecture system have been studied and reviewed. A system with less path delay and more data processing speed, is being designed in Xilinx.**

***Index Terms*—Reconfigurable Systems, Systolic array architecture, Parallel computing, Matrix multiplication, Path delay, Xilinx**

I. INTRODUCTION

The demand for high-speed and powerful data processing has become increasingly important in modern technologies due to the growing amounts of data that need to be processed. Traditional data processing techniques involved serial processing, which was a time-consuming and inefficient method that introduced various delays in the execution process.

To address this issue, new techniques have been developed, such as pipelining and parallel processing, which have led to faster execution times. Pipelining involves breaking down tasks into smaller components and processing them concurrently to reduce the overall processing time. On the other hand, parallel processing involves using multiple processors to perform tasks simultaneously, thereby increasing the speed and efficiency of data processing.

Researchers have also explored other techniques to achieve high-speed data processing, such as optimizing hardware complexity parameters and working on system architectures. By reducing the length of the data path and improving system architecture, data processing efficiency and speed can be improved.

Multipliers play a crucial role in digital signal processing and other applications, such as data encryption and compression. Researchers are continuously working on developing multipliers that offer high speed, low power consumption, regularity of layout, and compact VLSI implementation. These improvements make multipliers suitable for a range of high-speed, low-power, and compact applications.

In modern technology, the demand for high-speed and powerful data processing continues to grow as more data is generated, and the need for processing this data in real-time increases. Parallel computing technology is becoming increas- ingly popular due to its ability to overcome the complexities of traditional serial processing. This technology uses pipelining concepts to perform tasks concurrently and reduce processing time.

Overall, advancements in technology have led to various techniques that can be utilized to achieve high-speed data processing. These techniques include pipelining, parallel processing, and optimized hardware complexity parameters, which have improved the efficiency and speed of data processing. With ongoing research and development, the future of data processing looks promising, and we can expect even more powerful and efficient techniques to emerge in the coming years.

II. SYSTOLIC ARRAY ARCHITECTURE

The concept of a systolic architecture, which is a pipelined network arrangement of Processing Elements (PEs) known as cells in computer architecture. This architecture is a subset of parallel computing in which cells independently compute and store the data that is fed to them. The systolic architecture is a cell array composed of matrix-like rows, with Processing Elements analogous to central processing units (CPUs) except for the absence of a programme counter, instruction register, control unit, and so on. The systolic architecture performs parallel matrix multiplication using a PMMSA approach, which immediately multiplexes a pair of matrix elements. This approach is characterized by processing data input in a pipeline and is comprised of regularly arrayed PE, where neighbor PEs are connected with each other by the shortest

line, and therefore mass data has no need to be stored before processing. This minimizes the distance between PEs in an array, greatly reduces internal communication delay, and improves processing unit utility. To enhance the speed and reduce the complexity of the systolic architecture computation of matrix multiplication, the PE is replaced with Multiplication and Accumulation (MAC), which will be implemented using systolic array. The proposed systolic array architecture multiplier multiplies n x n matrices with a smaller number of clock cycles. Multiple data streams are typically sent and received by the systolic array, and multiple data counters are required to generate these data streams, allowing for data parallelism. A systolic array is a network of processors that compute and pass data through the system in a rhythmic manner. Every PE is composed of a full adder with three inputs and one output. The two inputs are X and Y respectively, and the third is an input as Carry-IN. The output carry is Carry- OUT, and the normal output is SUM. The working principle of the PE, which is to add two binary numbers using a full adder. The full adder has three inputs, X, Y, and Carry-IN, and it outputs two values, Carry-OUT and SUM. The sum output is calculated by taking the exclusive OR of X, Y, and Carry-IN. The Carry-OUT output is calculated by taking the AND of X and Y, the AND of X and Carry-IN, and the AND of Y and Carry-IN, and then taking the OR of these three results. The systolic architecture can be implemented in two methods: the conventional method (without pipeline and parallel processing) and the systolic architecture (with pipeline and parallel processing). The PE is the most basic element for implementing the systolic array architecture, and initially, a single PE is designed that consists of an AND gate and a full adder. For an 8/16-bit system, eight full adders are needed, which will be arranged like the systolic array architecture multiplier given in the article. In conclusion, the systolic architecture, which is a pipelined network arrangement of Processing Elements used for parallel computing. The article describes the concept of parallel matrix multiplication using a PMMSA approach, which is characterized by processing data input in a pipeline and is comprised of regularly arrayed PE.

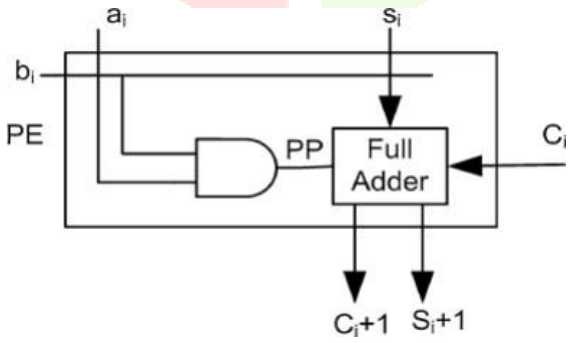

Fig. 1. Basic Structure of Processing Element

The article then explains how the PE works and how it can be used to implement the systolic array architecture. The proposed Systolic Array architecture multiplier multiplies n x n matrices with a smaller number of clock cycles. Systolic arrays typically send and receive numerous streams of data, as well as multiple data counters are needed to produce these data streams; thus, data parallelism is supported.

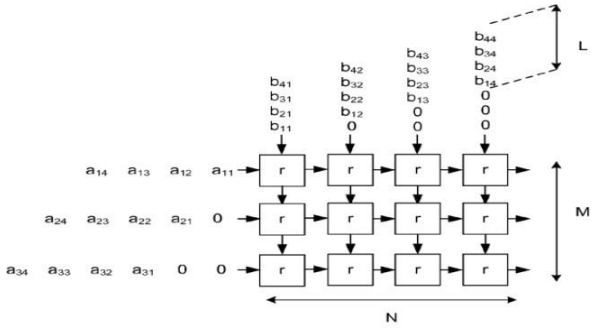

FIG. 9

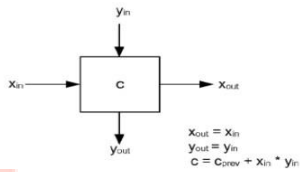

Fig. 2. Systolic Array Architecture Multiplier

Every PE is made up of full adder. the working principle of PE is described below Full adder has three inputs and gives out one output. The inputs are X and Y , the third input is Carry-IN. The output carry is Carry-OUT, and the output is SUM.

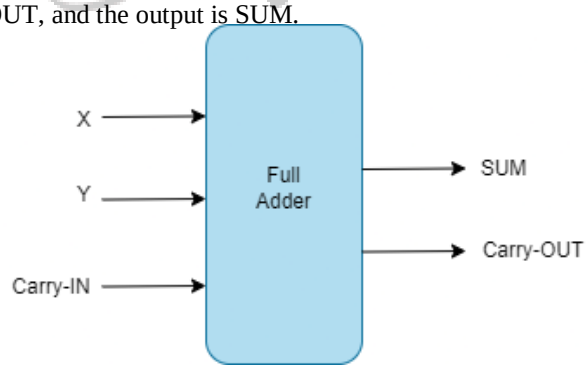

Fig. 3. Full Adder

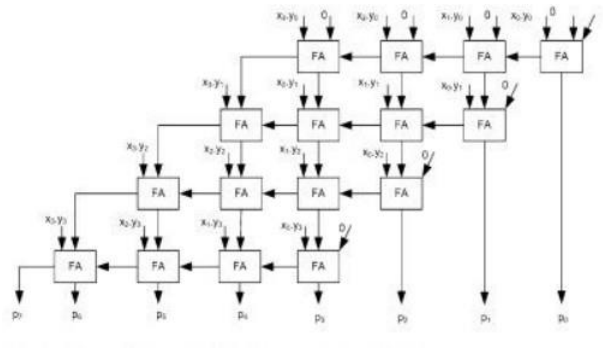

Fig. 4. Full Adder

III. LITERATURE SURVEY

Subathradevi [2018] have been studied a detailed review of design for the systolic array multiplier that minimises the delay product utilising different input bit sizes. Systolic architecture converts high-level computation into hardware structures. Systolic systems are simple to implement since these are regular and simple to calculate. Systems using this architecture are more efficient, provides better performance, and also includes specialised functions that can address a wide range of issues.In order to save energy, this paper developed a VLSI design, based on systolic multipliers that can be used on network links. The proposed study focuses on the creation of a unique architecture with cells used for the decomposition of systolic multiplier. There are two distinct designs that have been presented. The Architecture-I has been built using registers and decomposition cells to minimize latency in the event of a path delay. The second architecture i.e., Architecture-II consists of the tristate buffers and the full adder-based multiplexer along with the processing elements. Their effects lead to a product with a lower power delay. The experiment is conducted in a systolic array binary multiplier with a 1-level pipelining and the PEs will be organized in a row as well as a column-based structure symmetry and the results are appropriately listed in Table[citation]. Based on this observation, it has been deduced that the architecture provided a reduced delay compared to that of the architecture[citation] by using the register in the other feed-forward cut-sets of the middle cut-set. In conclusion, Subathradevi et.al[2018] has provided us the design of a systolic array multiplier that can handle different input bit sizes while minimizing the delay product is a crucial task in many real-time applications. In this proposed design, we focused on utilizing a variable word-size processing element (PE) array, combined with a pipelined architecture, to achieve high efficiency and flexibility. By utilizing a combination of fixed and variable word-size PEs, our design can handle different input data sizes while minimizing the overall delay product. The pipelined architecture enables parallel processing of multiple operands, further reducing the delay and increasing efficiency. Overall, our proposed design provides an effective solution for fast and efficient multiplication in various applications, including digital signal processing, scientific computing, and many others. Future research can focus on further optimizing the design parameters to achieve even higher performance while reducing power consumption.

Mahendra Vucha [2011] have been studied a detailed review of the design of a systolic array multiplier that can handle different input bit sizes while minimizing the delay product is a crucial task in many real-time applications. In this proposed design, they focused on utilizing a variable word-size processing element (PE) array, combined with a pipelined architecture, to achieve high efficiency and flexibility. By utilizing a combination of fixed and variable word-size PEs, our design can handle different input data sizes while minimizing the overall delay product. The pipelined architecture enables parallel processing of multiple operands, further reducing the delay and increasing efficiency. Overall, our proposed design provides an effective solution for fast and efficient multiplication in various applications, including digital signal processing, scientific computing, and many others. Future research can focus on further optimizing the design parameters to achieve even higher performance while reducing power consumption. The article describes several architectures used for block matching algorithms, including AB1, AS2, AB2 and AS1.These architectures use systolic arrays to process input data and compute the motion vector. The article proposes a new architecture that uses a two-dimensional systolic array to perform matrix multiplication of order N x N. The architecture uses Multiplication and Accumulation (MAC) to enhance speed and reduce complexity. Each processing element (PE) of the systolic array computes the multiplication of elements and accumulates the corresponding element. The architecture implements the algorithm in a pipeline and parallel processing to improve speed. The proposed architecture is shown below.

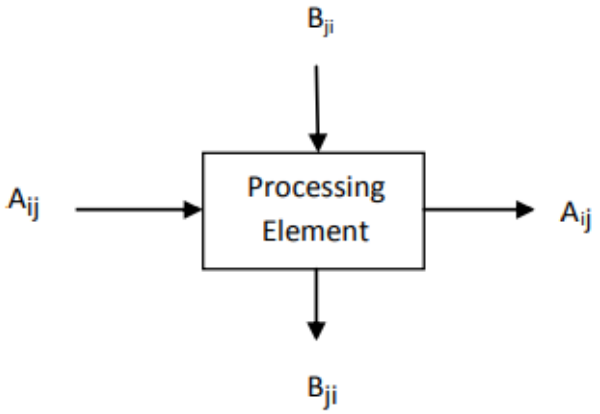

Fig. 5. PE of Systolic Architecture

In the conclusion, Vucha et.al [2018] we got Systolic Array Architecture that 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.

R Hughey[1988] have studied the detailed review of B SYS systolic array. They got to know that the B-SYS systolic array uses a linear interconnection scheme and a simple arithmetic logic unit to efficiently implement combinatorial algorithms. It uses an outside memory for program sequencing, resulting in a simpler architecture with the advantages of instruction broadcasting within a systolic framework. Pro- gramming involves mapping the data flow of the algorithm onto the systolic array and managing potential issues such as pipeline hazards. A sequence comparison algorithm has been programmed and simulated, but its performance is limited by the need for bit-serial communication at chip boundaries. In conclusion, Hughey et.al[1988] the B-SYS architecture demonstrates the potential for fully systolic programmable processor arrays that do not require local program memory or global instruction broadcasting. The use of the processing phase concept allows for the avoidance of hazards introduced by the systolic instruction stream. The basic cell of this architecture is simple and flexible, making it possible to build highly parallel, programmable systolic arrays. Further research on B-SYS will focus on exploring the architecture's limitations, examining techniques for mapping algorithms onto the array, and implementing a B-SYS prototype in CMOS technology.

Stojanovic et.al[1988] presents an approach for the imple- mentation of matrix-vector multiplication on a unidirectional linear systolic array (ULSA) and a bidirectional linear systolic array (BLSA) for efficient processing. The authors propose a systematic procedure for designing an optimal ULSA with the minimal execution time for a given problem size. They also propose a transformation method to partition a dense matrix into band matrices for adequate matching to the size of BLSA, with a new index transformation to avoid the insertion of zero elements between successive iterations. The obtained processing time is approximately two times shorter than previous methods, and the transformation is simpler. The paper provides a model for matrix-vector multiplication on a fixed-size ULSA and discusses the application of the proposed methods to improve processing efficiency. In the conclusion Stojanovic et.al[2007] accommodate the matrix size with the ULSA, the matrix A is partitioned into quasidiagonal blocks. During computation of the resulting vector cr, the elements of block matrices and vector cr are reordered to decrease computation time. The resulting processing time is about two times shorter than a previous method and equivalent to another method. The article proposes a global structure for the memory interface subsystem that facilitates data transfer to and from the ULSA. The article also estimates the performance of the fixed size ULSA and compares it to a bidirectional systolic array (BLSA).

Mishra et.al[2011] proposes a low power and fast archi- tecture for Viterbi decoding of convolutional codes, which integrates the benefits of systolic array architecture and register exchange strategy. The design and implementation of the systolic array based memoryless Viterbi decoder, as well as a modified register exchange decoding scheme, are pre- sented. The architecture achieves low power consumption and high throughput and is suitable for FPGA implementation. Performance analysis is based on maximum frequency of operation, FPGA resource utilization, and power consumption. The proposed architecture uses linear systolic structures and arithmetic pipelined processors, and reconfigurable composite branch metrics. In conclusion, Mishra et.al[2011] concluded that the decoder uses a modified register exchange strategy to achieve a memoryless design. The integration of a pointer-based register exchange method with the systolic architecture results in a faster and higher throughput system with a lower critical path compared to the trace-back strategy. Design reuse and time-multiplexing approaches are used to establish a fully reconfigurable system, contributing to area saving at the cost of latency. The architecture is modeled using Verilog and exported to Xilinx Virtex II pro xc2vp307fg676 using system generator tool. Results show that the proposed architecture outperforms its trace-back versions in terms of performance.

IV. CONCLUSION

Systolic arrays are a type of parallel computing architecture that can be used to perform a wide range of computational tasks, including digital signal processing and linear algebra operations such as matrix multiplication. The systolic array architecture is particularly well-suited for these applications because it is highly parallel and can process large amounts of data simultaneously. The design and implementation of efficient systolic array architecture for multiplier operations requires careful consideration of several design parameters. One of the most important design considerations is the number of processing elements in the systolic array. Increasing the number of processing elements can improve the throughput of the system but can also increase the power consumption and complexity of the design. Another important consideration is the data flow within the systolic array. The data flow deter- mines how data is propagated through the array and can greatly impact the performance of the system. For example, a diagonal data flow can reduce the number of data transfers and increase the system efficiency. Interconnect topology is also an impor- tant design parameter to consider. The topology determines how processing elements are connected to each other and can impact the system's scalability and flexibility. To further optimize systolic array architectures for multiplier operations, techniques such as pipelining, loop unrolling, and parallel processing can be used. Pipelining can reduce the latency of the system by dividing the computation into stages. Parallel processing can also be used to increase the throughput of the system by processing multiple data streams simultaneously. In conclusion, the design and implementation of efficient systolic array architectures for multiplier operations require a thor-

ough understanding of the application requirements, system constraints, and design trade-offs. By carefully considering these factors, designers can create systolic array architectures that provide high throughput, low power consumption, and scalability for a wide range of applications.

REFERENCES

[1] S. Subathradevi, C. Vennila "Systolic array multiplier for augmenting data center networks communication link"
[2] Mahendra Vucha, Arvind Rajawat "Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication" International Journel of Computer Applications (0975 – 8887)
[3] R. Hughey, Daniel P. Lopresti "Architecture of a programmable systolicarray".
[4] A. K. Mishra, P.P. Jiju "Low power, dynamically reconfigurable mem- oryless systolic array based architecture for Viterbi decoder".
[5] N. M. Stojanovic, I. Z. Milovanovic, M. K. Stojcev, E. I. Milovanovic "Matrix-vector multiplication on a fixed size unidirectional SystolicArray"