

Design and Implementation Modified Booth algorithm and systolic multiplier using FPGA

Himani

Student of M.Tech, Department of computer science
GNDU, Amritsar

Harmanbir Singh Sidhu

Student of M.Tech, Department of computer science
GNDU, Amritsar

ABSTRACT

From few years parallel computing is used in every field so that desirable results can be obtained in less time. Parallel computing enhances the speedup, performance, efficiency, cost etc. This paper discuss the effective design for binary multiplication using modified booth's algorithm and systolic multiplier. The systolic architecture increases the computing speed by combining the concept of parallel processing and pipelining into a single concept. Systolic array is an arrangement of processors in an array where data flows synchronously across the array between neighbors, usually with different data flowing in different directions. The design is simulated using modelsim by mentor graphics tool and Xilinx is used for the implementation of the code in FPGA

Keywords: Systolic array, FPGA, Processing element, binary multiplication.

1. INTRODUCTION

Systolic systems consists of an array of PE (Processing Elements) processors are called cells; each cell is connected to a small number of nearest neighbors in mesh like topology. Each cell performs a sequence of operations on data that flows between them. Generally the operations will be the same in each cell; each cell performs an operation or small number of operations on a data item and then passes it to its neighbor. Each cell at each step takes in data from one or more neighbors (e.g. North and West), processes it

and, in the next step, outputs results in the opposite direction (South and East). Systolic arrays compute in "lock-step" with each cell (processor) undertaking alternate compute/communicate phases. [1]

The systolic array can be defined as —Imagine n simple processors arranged in a row or an array and connected in such a manner that each processor may exchange information with only its neighbors to the right and left. The processors at either end of the row are used for input and output. Such a machine constitutes the simplest example of a systolic array and —Systolic Arrays are regular arrays of simple finite state machines, where each finite state machine in the array is identical .A systolic algorithm relies on data from different directions arriving at cells in the array at regular intervals and being combined. By pipelining, processing may proceed concurrently with input and output, and consequently overall execution time is minimized. Pipelining plus multiprocessing at each stage of a pipeline should lead to the best-possible performance. [2]

Features of Systolic arrays

A Systolic array is a computing network possessing the following features: [3]

1. **Synchrony** means that the data is rhythmically computed (Timed by a global clock) and passed through the network.
2. **Modularity** means that the array (Finite/Infinite) consists of modular processing units.
3. **Regularity** means that the modular processing units are interconnected with homogeneously.
4. **Spatial Locality** means that the cell has a local communication interconnection.
5. **Temporal Locality** means that the cells transmit the signals from one cell to other which require at least one unit time delay.
6. **Pipe inability** means that the array can achieve a high speed.

Systolic architecture can be used for special purpose processing architecture because of

1. Simple and Regular Design.
2. Concurrency and Communication.
3. Balancing Computation with I/O.

The advantages of systolic array are [2]

1. It has multiple cells networked together to form an array.
2. Faster speed due to register to register transfer of data.
3. Data is not destroyed until it has been completely used.
4. All cells run off of a central clock.
5. Host Data Entry – All cells are I/O capable.
6. Easily extend the architecture to many more processors.
7. Capable of supporting SIMD organizations for vector operations and MIMD for non homogeneous parallelism.
8. Allow extremely high throughput with multidimensional arrays.

Disadvantages of systolic arrays [3]

The main disadvantages of systolic arrays are:

1. Global synchronization limits due to signal delays.

2. High bandwidth requirements both for periphery (RAM) and between PEs.

3. Poor run-time fault tolerance due to lack of inter connection protocol

The paper is organized as follows. Section 1 says about the introduction. Section 2 deals with description of systolic array architecture. Literature review discussed in section 3. Section 4 discusses the design methodology and section 5 Implementation of the methodology. Conclusion of the topic is discussed in section 6.

2. DESCRIPTION OF SYSTOLIC ARRAY ARCHITECTURE

A systolic array is composed of matrix-like rows of processing elements called cells. Processing elements (PE)s are similar to central processing units (CPU)s, except for the usual lack of a program counter, since operation is transport-triggered, means by the arrival of a data object. Each cell shares the information with its neighbors immediately after processing. The systolic array is often rectangular where data flows across the array between neighbors PEs, often with different data flowing in different directions. The data streams entering and leaving the ports of the array are generated by auto-sequencing memory units, ASMs. Each ASM includes a data counter. In embedded systems a data stream may also be input from and/or output to an external source

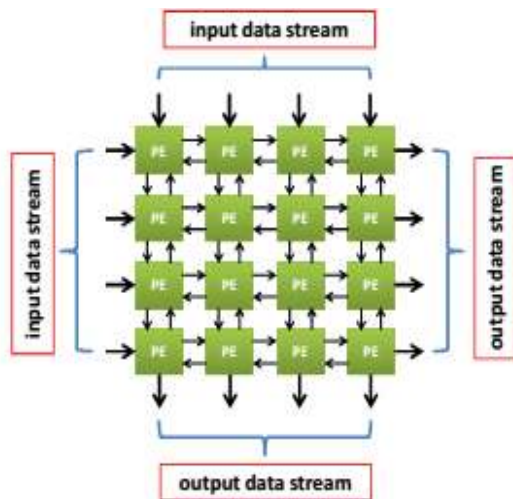


Figure 1. Architecture of systolic array

Systolic arrays are 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. Because the traditional systolic array synthesis methods have been practiced by algebraic algorithms, only uniform arrays with only linear pipes can be obtained, so that the architectures are the same in all DPUs. The consequence is that only applications with regular data dependencies can be implemented on classical systolic arrays. Like SIMD machines, clocked systolic arrays compute in "lock-step" with each processor undertaking alternate compute and communicate phases. But systolic arrays with asynchronous handshake between DPUs are called wavefront arrays. One well known systolic array is Carnegie Mellon University's iwrap processor, which has been manufactured by Intel. An iWarp system has a linear array processor connected by data buses going in both directions. A linear systolic array in which the processors are arranged in pairs where one multiplies its input by x and passes the result to the right and the next adds aj and passes the result to the right. [3]

3. LITERATURE REVIEW

3.1 Matrix multiplication on linear bidirectional systolic array [4]

This paper addresses the problem of rectangular matrix multiplication on bidirectional linear systolic arrays (SAs). Four different systolic algorithms for computing matrix product were proposed and corresponding BLSA, denoted as SA1, SA2, SA3 and SA4 that implement them were designed. It also compared the arrays SA1, SA2, SA3 and SA4 in term of efficiency and concludes that the efficiency depends on the relation between loop boundaries N_1 , N_2 and N_3 .

In order to compare synthesized arrays, it uses following performance measures:

Number of processing elements, Ω

Total execution time to compute matrix product, T_{tot}

Efficiency, E

	SA1	SA2	SA3	SA4
Ω	N_3	N_3	N_2	N_1
$T_{tot} \approx$	$N_2(N_1 + 2N_3 - 2)$	$N_1(N_2 + 2N_3 - 2)$	$N_3(N_1 + 2N_2 - 2)$	$N_3(N_2 + 2N_1 - 2)$
$E \approx$	$\frac{N_1}{N_1 + 2N_3 - 2}$	$\frac{N_2}{N_2 + 2N_3 - 2}$	$\frac{N_1}{N_1 + 2N_2 - 2}$	$\frac{N_2}{N_2 + 2N_1 - 2}$

Table 1. Performance measures of the synthesized arrays

3.2 Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication. [5]

This paper demonstrates an effective design for the Matrix Multiplication using Systolic Architecture on Reconfigurable Systems (RS) like Field Programmable Gate Arrays (FPGAs) device xc3s500e-5-ft256. The parallel processing and pipelining is introduced into the given systolic architecture to enhance the speed and reduce the complexity of the Matrix Multiplier. The given design is simulated, synthesized, implemented on FPGA device xc3s500e-5-ft256 and it has given the core speed 210.2MHz. It also compared the conventional design

and given systolic array design on FPGA device xc3s500e-5-ft256. From the Table 2, 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.

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

Table 2. Performance evaluation of Conventional method and Systolic Array architecture for Matrix Multiplication

3.3 Implementation of Binary Multiplication using Booth and Systolic Algorithm on FPGA using VHDL. [6]

In this paper, an attempt is made to implement the prototype of binary multiplier using Booth algorithm (for signed number) and the systolic array multiplication algorithm (for unsigned number). This is implemented using Xilinx I SE6 software, simulated using Modelsim XE 5.5a Simulator by Mentor graphics. The synthesis is done on Field Programmable Gate Array (FPGA) Spartan2S15 kit Using Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL).

3.4 Design & Implementation of Systolic Array Architecture. [7]

The paper describes the implementation of 2-D systolic array matrix multiplier architecture in RTL using one dimensional array to target the design on a

appropriate FPGA/PROM/CPLD devices. It also discusses the digital realization of a binary multiplier. The system development started with top-down planning approach and the blocks were designed using bottom-up implementation. The programs were written, simulated and synthesized using Mentor Graphics tools, Modelsim and Leonardo Spectrum.

3.5 Design and Implementation of an Efficient Modified Booth Multiplier using VHDL. [8]

This paper presents an efficient design of Modified Booth Multiplier and then also implements it. The Modified Booth Recoding method is widely used to generate the partial products for implementation of large parallel multipliers, which adopts the parallel encoding scheme. In this paper the software design of the Modified Booth Multiplier is explained with the help of flow chart. The simulation is done using Xilinx ISE Design Suite 14.2 tool and Modelsim tool and the results obtained are shown both for 4 bit and 8 bit multiplication. The implementation of this multiplier is done using VHDL on Spartan 3E kit and the hardware results are also shown.

4. DESIGN METHODOLOGY

Binary Multiplication with modified booth's algorithm and systolic multiplier

According to Modified Booth's algorithm, the two input binary numbers the one with minimum number of bit changes is considered as multiplier and the other as a multiplicand in order to reduce the time taken for calculating the multiplication product. Modified Booth Multiplier consist of three basic components namely Booth encoder (BE), Booth Selector (BS) and adder tree summation. The basic operation of Booth Encoder is to decode the multiplier signal and output will be used by booth selector to generate the partial product. Adder tree summation accumulates the entire partial product to produce the result. This method is explained in flow chart as below:-

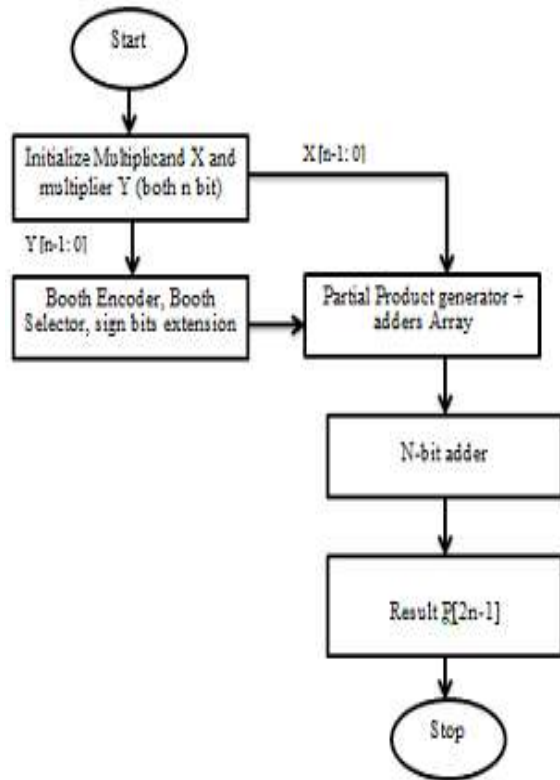


Figure 2. Flowchart of Modified Booth's algorithm

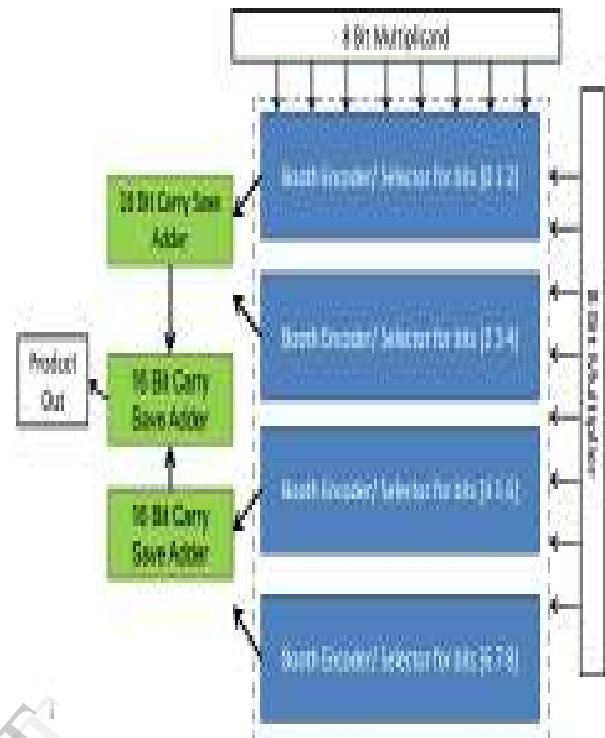


Figure 3. Block Diagram for Booth's Multiplier

Modified Booth Algorithm: (for unsigned numbers)

1. Pad the LSB with one zero.
2. Pad the MSB with 2 zeros if n is even and 1 zero if n is odd.
3. Divide the multiplier into overlapping groups of 3-bits.
4. Determine partial product scale factor from modified booth 2 encoding table.
5. Compute the Multiplicand Multiples
6. Sum Partial Products

In systolic multiplication, to carry out the multiplication and get the final product following steps should be followed

1. The multiplicand and multiplier are arranged in the form of array as shown in the Fig. (4).
2. Each bit of multiplicand is multiplied with each bit of multiplier to get the partial products.
3. The partial products of the same column are added along with carry generated.
4. So the resulted output by adding partial products and the carry is the final product of the two binary numbers.

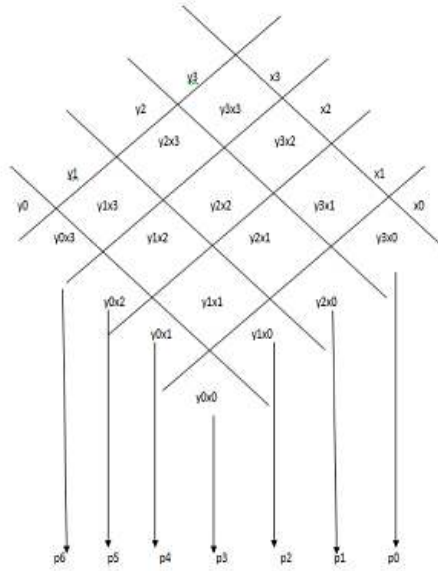


Figure 4. Block Diagram For systolic multiplier

5. IMPLEMENTATION

The simulation of the program is done using online tool to find results which is shown in Fig (5). Another method for simulation of the program is done by using Modelsim tool by mentor graphics and Xilinx is used for the implementation of the code in FPGA which is shown in Fig (6).

Here are some screenshots at various stages of the implementation procedure:



Figure 5. Modified booth algorithm using online tool

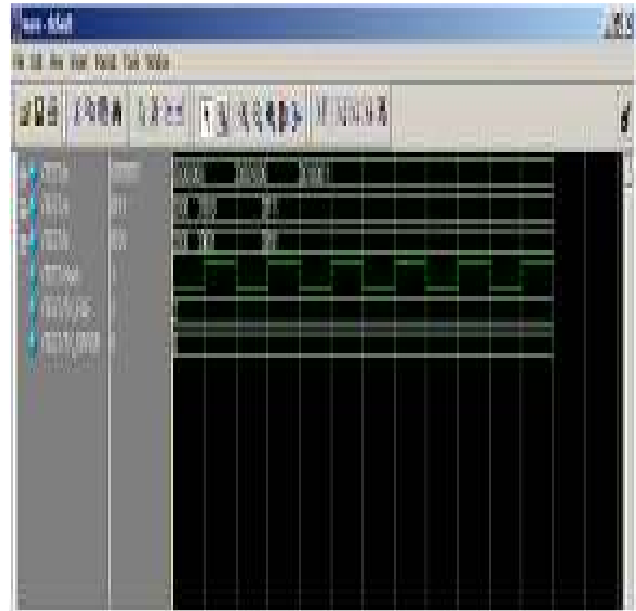


Figure 6. Output in waveform using Modelsim

6. CONCLUSION AND FUTURE SCOPE

There are many binary multipliers used at present, they can be replaced by FPGA binary multiplier which consumes less area, takes less time to multiply and consumes less power, which are the main criterions of the present system. In this paper, binary multiplication is done with the help of modified booth algorithm and systolic multiplier. The modelsim simulator is used to implement the multiplier. In future, synthesis can be done by FPGA Spartan 2 S15 kit using Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL).

7. REFERENCES

[1] Jonathan Break, "Systolic Arrays & Their Applications", http://www.cs.ucf.edu/courses/cot4810/fall04/presentations/Systolic_Arrays.ppt

[2]<http://web.cecs.pdx.edu/~mperkows/temp/May13/070-Systolic-Processors.pdf>

[3] Pratibhadevi Tapashetti, A.S Umesh, Ashalatha Kulshrestha, "Design and Simulation of Energy Efficient Full Adder for Systolic Array" IJSCE ISSN: 2231-2307, Volume-1, Issue-6, January 2012.

[4] E. I. Milovanović, B. M. Randjelović, I.Z. Milovanović, M. K. Stojčević, "Matrix Multiplication on Linear Bidirectional Systolic Array" ser. A: a ppl. Math. Inform. And mech. vol. 2, 1 11-20, 2010.

[5] Mahendra Vucha, Arvind Rajawat, "Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication", IJCA (0975 – 8887) Volume 26– No.3, July 2011.

[6] Jayashree Taralabench, Kavana Hegde, Soumya Hegde, Siddalingesh S. Navalgund, "Implementation of Binary Multiplication using Booth and Systolic Algorithm on FPGA using VHDL", International Conference & Workshop on Recent Trends in Technology, (TCET) 2012

[7] Sweta singh, N B singh, "Design & Implementation of systolic array architecture", (IJEER) ISSN 2250-155X Vol. 3, Issue 4, 117-128 Oct 2013.

[8] Kavita, Jasbir Kaur, "Design and Implementation of an Efficient Modified Booth Multiplier using VHDL", Vol.3 (3), July, 2013.