High Performance Matrix Multiplication Architecture Based on PPI-MO Techniques

Bhishm Jawarkar[#1], Puran Gour*[2], Braj Bhihari Soni[$3]

[1]M. Tech Scholar Department of Electronics & Communication NRI Institute of Technology Bhopal, INDIA
[2]Head of Department of Electronics & Communication NRI Institute of Technology Bhopal, INDIA
[3]Assistant professor Department of Electronics & Communication NRI Institute of Technology Bhopal, INDIA
[1]bhishm.jawarkar@gmail.com
[2]purangour@rediffmail.com
[3]brizsoni@gmail.com

***Abstract:*-Matrix multiplication is the kernel operation used in many transform, image and discrete signal processing application. We develop new algorithms and new techniques for matrix multiplication on configurable devices. In this paper, we have proposed three designs for matrix-matrix multiplication. These design reduced hardware complexity, throughput rate and different input/output data format to match different application needs. These techniques have been designed implementation on Virtex-4 FPGA. Wehave synthesized the proposed designs and the existing design using Synopsys tools. Interestingly, the proposed parallel-fixed-input and multiple-output (PPI-MO) structure consumes 40% less energy than other two proposed structures and 70% less energy than the existing structure.**

Keywords:-**Parallel-Parallel Input and Single Output (PPI-SO), Parallel-Parallel Input and Multi Output (PPI-MO), Synopsis Simulation.**

I. INTRODUCTION

With the growth in scale of integration circuits, more and more sophisticated digital signal processing circuits are being implemented in (field programmable gate array) FPGA based circuit. Indeed, FPGA have become an attractive fabric for the implementation of computationally intensive application such as digital signal processing, image, graphics card and network processing tasks used in wireless communication. These complex signal processing circuits not only demand large computational capacity but also have high energy and area requirements. Though area and speed of operation remain the major design concerns, power consumption is also emerging as a critical factor for present VLSI system designers [1]-[4]. The need for low power VLSI design has two major motivations. First, with increase in operating frequency and processing capacity per chip, large current have to be delivered and the heat generated due to large power consumption has to be dissipated by proper cooling techniques, which account for additional system cost. Secondly, the exploding market of portable electronic appliances demands for complex circuits to be powered by lightweight batteries with long times between re-charges (for instance [5].

Another major implication of excess power consumption is that it limits integrating more transistors on a single chip or on a multiple-chip module. Unless power consumption is dramatically reduced, the resulting heat will limit the feasible packing and performance of VLSI circuits and systems.From the environmental viewpoint, the smaller the power dissipation of electronic systems, the lower heat pumped into the surrounding, the lower the electricity consumed and hence, lowers the impact on global environment [6].

Matrix multiplication is commonly used in most signal processing algorithms. It is also a frequently used kernel operation in a wide variety of graphics, image processing as well as robotic applications. The matrix multiplication operation involves a large number of multiplication as well as accumulation. Multipliers have large area, longer latency and consume considerable power compared to adders. Registers, which are required to store the intermediate product values, are also major power intensive component [7]. These components pose a major challenge for designing VLSI structures for large-order matrix multipliers with optimized speed and chip-area. However, area, speed and power are usually conflicting hardware constraints such that improving upon one factor degrades the other two.

Silicon-area, speed, power consumption and design cost are the general parameters that are taken care while designing VLSI architecture, DSP system and high performance system. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. Earlier, power consumption was a secondary concern in comparison to area and speed. However, in recent years, power is being given more importance as area and speed due to phenomenal growth of portable and wireless handheld multimedia devices. The power consumption is the most critical design concern for these devices [5].

We have proposed designs for implementing the matrix multiplication operation in hardware keeping the goal of a power efficient architecture. These designs are verified using various hardware simulating tools.

The entire paper has been partitioned into four parts. In II,Theoretical Background for matrix multiplication has been discussed. In III, hardware complexity and performance comparison of the proposed architecture is discussed.In IV, simulation result has been discussed. In V, conclusions and future scope of the paper work has been presented.

II. THEORETICAL BACKGROUND

Power consumption in a VLSI circuit depends on various factors. Some of them are discussed below.

A. Factors Contributing for Power

The major power dissipation sources in CMOS circuits are given by the following equation –

$$P_{avg} = P_s + P_{sc} + P_l \qquad (1)$$

where P_{avg} is the average power dissipated by the circuit, P_s is the switching component of the power caused by charging/discharging of the circuit output load capacitance C_l, and P_{sc} and P_l reflect the power dissipated due to short- circuit and leakage currents respectively(I_{sc} and I_l). The above equation could be expanded to reveal the basic circuit parameters contributing to each of the power components as follows –

$$P_{avg} = k.C.V_{dd}^2.f + I_{sc}.V_{dd} + I_l.V_{dd} \qquad (2)$$

where V_{dd} is the power supply voltage, f is the clock frequency, C is the physical capacitance of the circuit, and k is the transition activity factor which gives the average number of times the circuit makes a power consuming transition in a single clock cycle. The parameters k and C are often lumped together in a single parameter termed as effective capacitance of a design.

B. SWITCHING ACTIVITY

The second major contributor to the dynamic power consumption along with, voltage and physical capacitance is switching activity. The data activity determines how often this switching occurs. There are two components to switching activity: f_{clk} which determines the average periodicity of data arrivals and *E(sw)* which determines how many transitions each arrival will generate. For circuits that do not experience glitching, *E(sw)* can be interpreted as the probability that a power consuming transition will occur during a single data period. Even for these circuits, calculation of *E(sw)* is difficult as it depends not only on the switching activities of the circuit inputs and the logic function computed by the circuit, but also on the spatial and temporal correlations among the circuit inputs.

The data activity *E(sw)* can be combined with the physical capacitance *C* to obtain switched capacitance, C_{sw} =*C.E(sw)*, which describes the average capacitance charged during each data period $1/f_{clk}$. It should be noted that it is the switched capacitance that determines the power consumed by a CMOS circuit.

C. SCHEMES FOR OPTIMIZING SPEED

1)Pipelined Mapping

It is a powerful transformation which can improve the performance of both general purpose and special purpose architectures. It involves the insertion of delay elements at specific points of a DFG of an algorithm\structure. The aim of pipelining is to increase the amount of concurrency in the design. The application of pipelining to synchronous hardware architectures can allow operation with a faster system clock. However, pipelining increases both system latency and the number of delay.

2)Parallel Processing

Parallel processing is the use of multiple processors to execute different parts of the same program simultaneously. It is similar to pipelining in that it exploits parallelism in a system; however, here this is achieved by duplicating hardware sections in order to perform a number of similar tasks concurrently. Parallelism gives better performance by reducing the processor complexity but increases the area occupancy of the system.

3)Parallel and Pipelined Mapping

The processor requirements for parallel mapping can be greatly optimized with the use of both parallel and pipelining mapping. This technique enables the advantage of both the mapping techniques.

D. HARDWARE AND TIME COMPLEXITY OF MATRIX MULTIPLICATION

Let us consider the matrix – matrix multiplication for two n×n matrices A and B given by-

$$C\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix} = A\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \times \mathrm{B}\begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix} \quad \ldots. (3)$$

The matrix multiplication can be represented as

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \times b_{kj} \quad \ldots\ldots\ldots\ldots (4)$$

for all i, *j*,a_{ik}, b_{kj}, and c_{ij}represent elements of the n×n matrices A, B and C.

So a Matrix-vector multiplication can be performed through M inner-product computation for M rows in A. Each Inner product computation (IPC) involves N multiply and add operations.

III. PROPOSED ARCHITECTURE

A. Proposed Parallel-Parallel Input and Multi Output(PPI - MO)

In this design, we opted for faster operating speed by increasing the number of multipliers and registers performing the matrix multiplication operation. From equation 2 we have derived for parallel computation of 3 × 3 matrix-matrix multiplication and the structure is shown in figure 1.

For an n×n matrix – matrix multiplication, the operation is performed using n^2 number of multipliers, n^2 number of registers and $n^2 - n$ number of adders. The registers are used to store the partial product results. Each of the n^2 number of multipliers has one input from matrix B and the other input is obtained from a particular element of matrix A. The dataflow for matrix B is in row major order and is fed simultaneously to the particular row of multipliers such that the i^{th} row of matrix B is simultaneously input to the i^{th} row of multipliers, where 1 < i < n . The elements of matrix are input to the multipliers such that, $(j,i)^{th}$ element of matrix A is input to

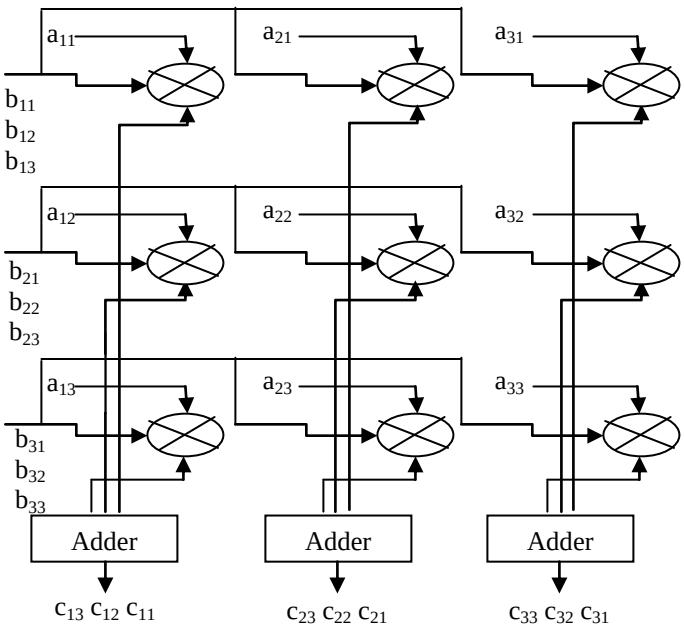

Figure 2: Proposed PPI – MO Design for n = 3

the $(i,j)^{th}$ multiplier, where1 <i,j< n. The resultant products from each column of multipliers are then added to give the elements of output matrix C. In one cycle, n elements of matrix C are calculated, so the entire matrix the elements of matrix C are obtained in column major order with n elements multiplication operation requires n cycles to complete.

Let us consider the example of a 3×3 matrix – matrix multiplication operation, for a better analysis of the design (as shown in figure 1). The hardware complexities involved for this design are 9 multipliers, 9 registers and 6 adders. Elements from the first row of matrix B (b_{11} b_{12} b_{13}) are input simultaneously to the first row of multipliers (M_{11} M_{12} M_{13}) in 3 cycles. Similarly, elements from other two rows of matrix B are input to the rest two rows of multipliers. A single element from matrix A is input to each of the multipliers such that, $(j,i)^{th}$ element of matrix A is input to the multiplierM_{ij}, where 1 < i,j< 3. The resultant partial products from each column of multipliers (M_{1k} M_{2k} M_{3k} where 1 < k 3) are added up in the adder to output the elements of matrix C. In each cycle, one column of elements from matrix C is obtained (C_{1k} C_{2k} C_{3k} where1 < k < 3) and so the entire matrix multiplication operation is completed in 3 cycles.

IV. SIMULATION RESULTS

For an×n matrix multiplication, bhishm et al. [11] design uses *n* multipliers and *n* registers. This design is optimized for reduced component use and has a penalty of increased operating times (n^2 cycles). The input is obtained through 2*n* ports and output is calculated out by a single port. The proposed architecture, designs have been optimized for faster operating speeds. Both designs require n^2 multipliers and n^2 registers to complete the matrix multiplication operation in *n* cycles. The major difference between PPI – MO and PPI – SO is that, PPI - MO requires $n^2 + n$ input ports whereas only *n* input ports are required for PPI – SO design.These designs were compared with prevalent matrix multiplication architecture proposed by Jang *et al*. [9], Qasim et al. [10] and Bhishm et al. [11] to show for the improvements obtained. A comparative theoretical analysis is given in Table 1.

So Bhishm et al. [11] architecture is best in all these architectures. Implementing the bhishm [11], proposed architecture PPI-MO design has been captured by VHDL and the functionality is verified by RTL and gate level simulation. To estimate the timing, area and power information for ASIC design, we have used Synopsys Design Compiler to synthesize the design into gate level. Comparison of Xilinx result is given in Table 2 respectively.

Table 1: Theoretical comparative hardware analysis

	Design by Jang et al. [9]	**Design by Xiaoxiao Jiang *et al.* [10]**	**Design by Bhishm et al. [11]**	**Proposed Architecture**
Matrix Size	3*3	4*4	4*4	4*4
Delay time	6.99nsec	16Ts	18.610nsec	6.959nsec
No. of CLB's	144	293	110	96
Number of 4 input LUT's	270	458	84	102
Maximum frequency	87MHz	734.754 MHz	353.482 MHz	60.835 MHz

Table 2: Xilinx Result

Architecture (4×4)	**Number of Slice**	**4 Input LUTs**	**Maximum Combination Path Delay**
Design by Bhishm et al. [11]	110	84	18.610nsec
Proposed Architecture	97	102	6.959nsec

V. CONCLUSION AND FUTURE SCOPE

Most of the digital signal processing (DSP) algorithms is formulated as matrix-matrix multiplication, matrix-vector multiplication and vector-vector (Inner-product and outer-product) form. Few such algorithms are digital filtering, sinusoidal transforms, wavelet transform etc. The size of matrix multiplication or inner-product computation is usually large for various practical applications. On the other hand, most of these algorithms are currently implemented in hardware to meet the temporal requirement of real-time application [9]. When large size matrix multiplication or inner product computation is implemented in hardware, the design is resource intensive. It consumes large amount of chip area and power. With such a vast application domain, new designs are required to cater to the constraints of chip area and power and high speed.

We have compared the proposed designs with the existing similar design and found that, the proposed designs offer higher throughput rate at relatively lower hardware cost.

Other possible improvements can also be obtained by applying different design strategies such as low-complexity multiplier less approach for optimizing the power/energy of the proposed designs.

VI. REFERENCE

[1] MassoudPedram, "Design Technologies for Low Power VLSI," Encyclopedia of Computer Science and Technology, pp. 1 – 32, 1995.

[2] Pramod Kumar Meher, "Hardware-Efficient Systemization of DA-Based Calculation of Finite Digital Convolution," IEEE Transaction on Circuits and Systems, vol. 53, no. 8, pp. 707 - 711, 2006

[3] L. Benini, G. De Micheli, E. Macii, "Designing Low-Power Circuits: Practical Recipes," IEEE Circuits and Systems Magazine, vol. 1, no. 1,pp. 6-25, 2001.

[4] M. Horowitz, T. Indermaur, R. Gonzalez, "Low Power Digital Design,"IEEE Symposium on Low Power Electronics, pp. 8-11, 1994.

[5] Pramod Kumar Meher, "New Approach to Look-Up-Table Design and Memory-Based Realization of FIR Digital Filter," IEEE Transaction on Circuits and Systems, vol. 57, no. 3, pp. 592 - 603, 2010

[6] S. Tugsinavisut, S. Jirayucharoensak and P. A. Beerelt, "An Asynchronous Pipeline Comparisonswith Applications to DCT Matrix-vector Multiplication," in Proceedings of the 2003 International Symposium on Circuits and Systems(ISCAS), vol. 5, pp. V-361 - V-364, 2003

[7] J. Lloyd, "Parallel Formulations of Matrix-Vector Multiplication for Matrices with Large Aspect Ratios, "in IEEE Proceedings of the Fourth Euro micro Workshop on Parallel and Distributed Processing, pp. 102-108, 1996

[8] K. K. Parhi, VLSI Digital Signal Processing Systems. John Wiley & Sons, Inc. 1999.

[9] Ju-Wook Jang, Seonil B. Choi, and Viktor K. Prasanna," Energy- and Time-Efficient Matrix Multiplication on FPGAs", IEEE Transaction on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 11, pp. 1305 – 1319, 2005.

[10] Xiaoxiao Jiang,Jun Tao,"Implementation of Effective Matrix Multiplication on FPGA",IEEE IC-BNMT2011.

[11] BhishmJawarkar, PuranGour, BrajBihariSoni, "Parallel Processing Technique for Time Efficient Matrix Multiplication", Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 4, Issue 9(Version 4), September 2014, pp.83-86.

