International Journal For Technological Research In Engineering
Volume 2, Issue 7, March-2015 *ISSN (Online): 2347 - 4718*

IMPROVED MATRIX MULTIPLICATION USING MUX FOR HIGH SPEED DSP APPLICATIONS

Raghavendra M[1], Prof. Seetha Rama Raju Sanapala[2]
Department of ECE, REVA ITM
Bangalore, India

Abstract: Power consumption has become a critical concern in today's VLSI system design. The growing market for fast floating-point co-processors, graphics processors and digital signal processing chips has created a demand for high speed, area efficient multipliers. Multiplication is most commonly used operation in mathematics. Integer multiplication is used commonly in the real world, binary multiplication used for the integer multiplication. An implementation of an improved matrix multiplier for high speed digital signal processing applications is sbased on matrix element transformation. The proposed methodology ensures substantial reduction in propagation delay, area, power compared with the conventional algorithm, systolic array and pseudo number theoretic transformation (PNTT)-based implementation, which are the most commonly used techniques, for matrix multiplication.

I. INTRODUCTION

Jianwen and Chuen [7] proposed partiallyre-configurability feature which was exploited for the first time to compute matrix multiplication. Partially reconfigurable devices offer the possibility of changing the design implementation without stopping the whole execution process. The design was evaluated in terms of area and latency. Mahendra Vucha and Arvind Rajawat [8] presented an effective design for the Matrix Multiplication using systolic Architecture on Reconfigurable systems like FPGAs. Here, the systolic architecture increases the computing speed by combining the concept of parallel processing and pipelining into single concept. Syed M. Qasim et al, [9] presented a preliminary design and FPGA implementation of dense matrix-vector multiplication for use in an image processing application. The architecture was designed to multiply large matrix and a vector. Nivedita A. Pandeet al., [9] proposed a design methodology for high-speed multiplications, where two integers of n-bit size each are multiplied to produce a 2n-bit product. We provide a novel approach to the design of fast algorithms for matrix multiplication. The operation of matrix multiplication is reformulated as a convolution, which is implemented using pseudo-number-theoretic transforms. Writing the convolution as multiplication of polynomials evaluated off the unit circle reduces the number of multiplications without producing any error, since the (integer) elements of the product matrix are known to be bounded. The new algorithms are somewhat analogous to the arbitrary precision approximation (APA) algorithms, but have the following advantages: (1) They do not suffer from round off error, (3) Reasons for their existence is clear and (2) a simple design produce is specified for them. The new algorithms are also non-commutative, so that they may be applied recursively to block matrix multiplication. This work provides a link between matrix multiplication and fast convolution algorithms and so opens another line of enquiry for the fast matrix multiplication problem [2]. Sparse matrix–vector multiplication (SpM * V) has been characterized as one of the most significant computational scientific kernels. The key algorithm characteristic of the SpM * V kernel, that achieving high performance, is its very low flop byte ratio. In this, they present a compressed storage format, called compressed sparse extended (csx) that is able to detect and encode simultaneously multiple commonly encountered substructures inside a sparse matrix. Relying on aggressive compression techniques of the sparse matrix's indexing structure; CSX is able to considerably reduce the memory footprint of a sparse matrix, alleviating the pressure to the memory subsystem. In a diverse set of sparse matrices, CSX was able to provide a more than 40 percent average performance improvement over the standard CSR format in SMP architectures and surpassed 20 percent improvement in NUMA systems, significantly outperforming other CSR alternatives. Additionally, it was able to adapt successfully to the nonzero element structure of the considered matrices, exhibiting very stable performance [3]. Engineering changes (ECs) are raised throughout the lifecycle of engineering products. A single change to one component produces knock-on effects on others necessitating additional changes. This change propagation significantly affects the development cost and time and determines the product's success. Predicting and managing such ECs is, thus, essential to companies. Some prediction tools model change propagation by algorithms, whereof a subgroup is numerical. Current numerical change propagation algorithms either do not account for the exclusion of cyclic propagation paths or are based on exhaustive searching methods. This paper presents a new matrix-calculation-based algorithm which can be applied directly to a numerical product model to analyze change propagation and support change prediction. The algorithm applies matrix multiplications on mutations of a given design structure matrix accounting for the exclusion of self-dependences and cyclic propagation paths and delivers the same results as the exhaustive search-based Trail Counting algorithm. Despite its factorial time complexity, the algorithm proves advantageous because of its straightforward matrix-based calculations which avoid exhaustive searching. Thereby, the algorithm can be implemented in established numerical programs such as

Microsoft Excel which promise a wider application of the tools within and across companies along with better familiarity, usability, practicality, security, and robustness [4]. Matrix multiplication is a core operation in digital signal processing operations with a variety of applications such as image processing, sonar processing, computer graphics and robotics. This paper presents the design and implementation of a high performance, fully parallel matrix multiplication core. The core is parameterized and scalable in terms of the matrices' dimensions (row and column number) and the input data word length [6]. Matrix multiplication is a computationally-intensive and fundamental matrix operation in many algorithms used in scientific computations. It serves as the basic building block for image processing, signal, graphics and robotic applications. To improve the performance of these applications, a high performance matrix multiplier is required. Traditionally, matrix multiplication operation is either realized as software running on fast processors or on dedicated hardware. Software based matrix multiplication is slow and can become a bottleneck in the overall system operation. However, Field Programmable Gate Array (FPGA)-based design of matrix multiplier provides a significant speed-up in flexibility and computation time. This paper presents an FPGA-based hardware realization of matrix multiplication based on a parallel architecture. The proposed parallel architecture employs advanced design techniques and exploits architectural features of FPGA [8].

II. DESIGN METHODOLOGY

A. Linear Algebraic

Matrix multiplication is an important kernel in linear algebraic operations and the performance of serial implementations is highly dependent on the number of multiplications. Many algorithms in numerical and non-numerical problems are solved efficiently using matrix multiplication. Matrix multiplication is essential not only in graph theory but also in applied fields, such as digital signal processing and computer graphics. In fact, DSP chips are found in all cell phones and digital cameras. The reason being, only through matrix operations, DSP chips can digitize sounds or images and ultimately these will be stored or transmitted electronically. Fast matrix multiplication is still an open problem, but implementation of existing algorithms has not yielded optimal results. The central role of matrix multiplication as a building block in numerical code has generated a significant amount of research into techniques for improving the performance of these basic operations. From the available literature survey [1] to [9], it is found that Strassen's algorithm for matrix multiplication gains its lower arithmetic complexity at the expense of reduced locality of references. Accordingly this makes it challenging to implement the algorithm efficiently on a modern machine with a hierarchical memory system. It should be noted that, this algorithm internally uses (i) a nonstandard array layout known as Morton order that is based on a quad-tree decomposition of the matrix and (ii) a dynamic section of the recursion truncation point to minimize padding without affecting the performance of the algorithm. The results of the present innovative work are of high practical and industrial interest.

B. Standard / Conventional Matrix Multiplication

In this section, a standard implementation of matrix multiplication is applied to the mathematical method of calculation i.e., multiplying row by column. This method is used as base-line. Here we consider the problem of computing the product of matrix multiplication. Accordingly, the product C=A X B of two 2 X 2 matrices is given by

$$A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad B=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad AB=\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \quad \ldots\ldots(1)$$

c11 = a11*b11 + a12*b21
c12 = a11*b12 + a12*b22
c21 = a21*b11 + a22*b21
c22 = a21*b12 + a22*b22

In general,

$$C_{ij} = \sum_{k=1}^{n} A_{ik} * B_{kj}$$

$$(\; 0<i<=n \,, 0<j <= n \;)$$

C. Recursive Matrix Multiplication

An alternative method for standard matrix multiplication is by applying Divide and Conquer technique and compute recursively i.e., partition the matrix and calculate the inner products. The smallest sub-problem can be defined as a 2 X 2 matrix.

D. Systolic Array Architecture

A systolic architecture is an array of Processing Elements, each called as a cell. Each cell is connected to a small number of nearest neighbours in a mesh like topology. Each cell performs a sequence of operations on data that flows between them. PE at each step takes input data from one or more neighbours (e.g. Left and Top), processes it and, in the next step, outputs results in the opposite direction (Right and Bottom) [9].The Proposed two dimensional systolic Architecture for 3 by 3 matrixes is given in below fig.

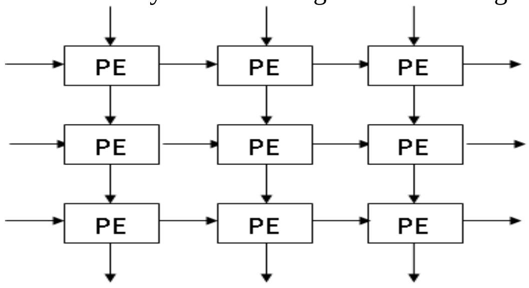

Figure 1. Two-dimensional Systolic Array

E. Implementation Of Modified Booth Recoded Wallace Tree Multiplier For Fast Arithmetic Circuits

The multiplier is composed of three blocks: the Modified Booth Encoder and multiplicand selector block for formation of partial products and Wallace tree section, which adds all of the partial products simultaneously to eventually produce two numbers, the third block is the adder section which adds the two numbers obtained from the Wallace tree section as fast as possible. 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. A Wallace tree multiplier is an improved version of tree based multiplier architecture and uses carry save addition algorithm to reduce the latency. This paper presents efficient design of multiplier that combines the features of Modified Booth algorithm and Wallace tree, and aimed at reduction of area and improvement in speed [11].

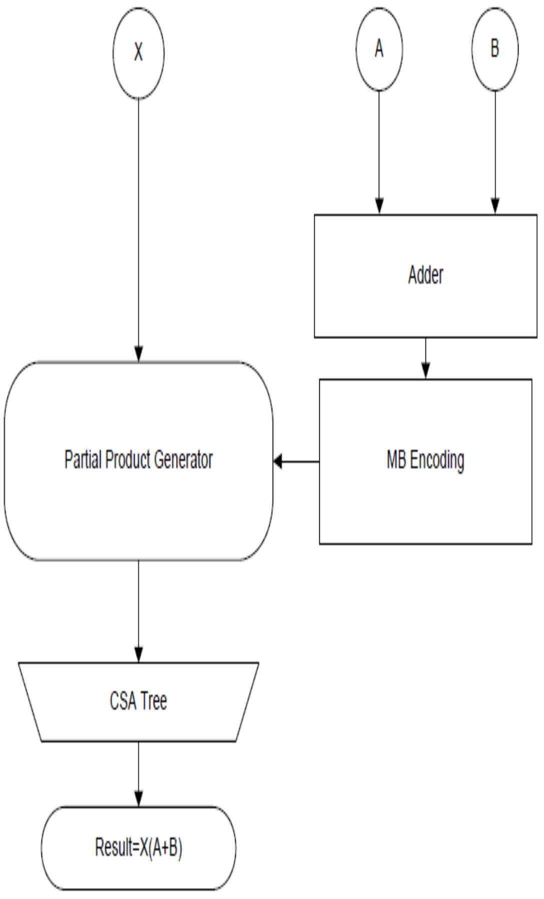

Fig. 2 Architecture of Add Multiply unit

III. PROPOSED METHODOLOGY

The Main Objective of the Project is to design an efficient matrix multiplication which involves in implementation of several techniques and comparing its performance and choosing the efficient and emerging new concept for matrix using multiplexer. The Implementation of Multiplication involves generation of partial products which can be easily designed by using multiplexer which consumes less power and area.

The design description is given as follows consider A = 1011 and B = 0101 Using Multiplexer the partial products can be designed as

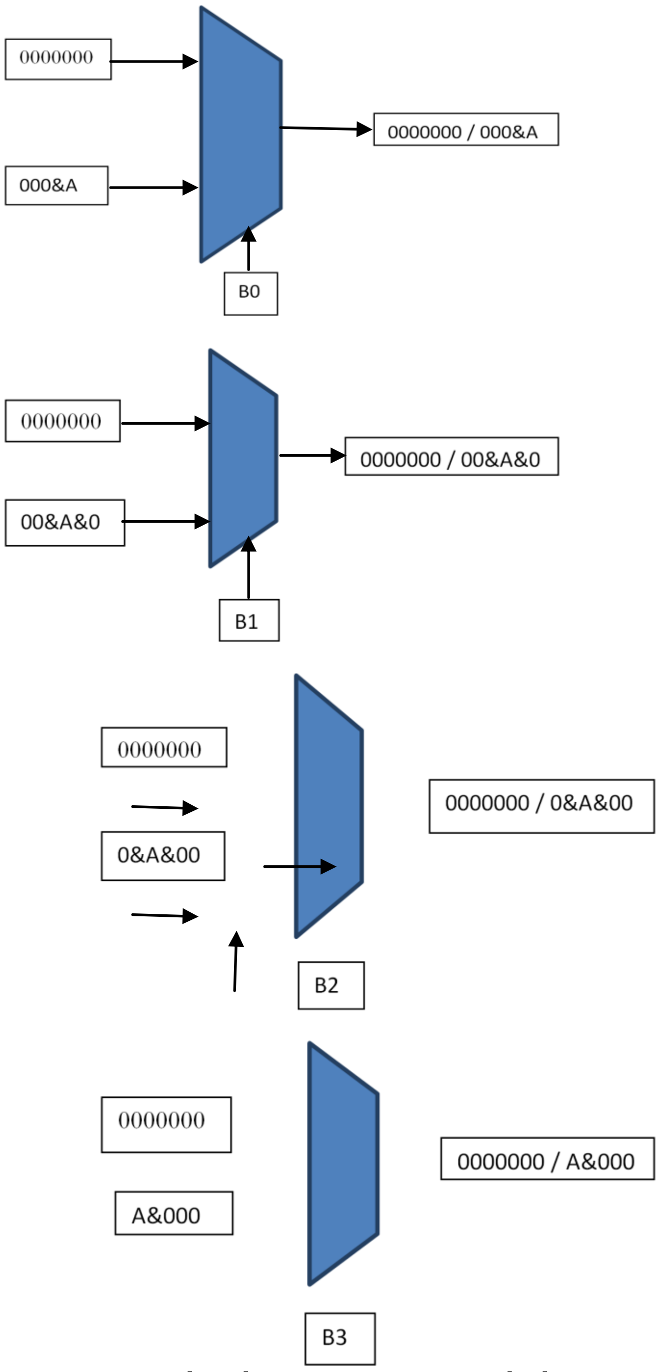

Fig.3 Partial product generation using multiplexer

After the generation of partial products the all partial products can be added by using Wallace tree multiplication. Here the systolic array multiplication equations considering a 2 x 2 metric is given as

c11 = a11*b11 + a12*b21
c12 = a11*b12 + a12*b22
c21 = a21*b11 + a22*b21
c22 = a21*b12 + a22*b22

The desired a11 b11 a12 b12 a21 b21 a22 b22 bits can be multiplied by the above method which reduces time in generation of partial product and increases the speed of operation.

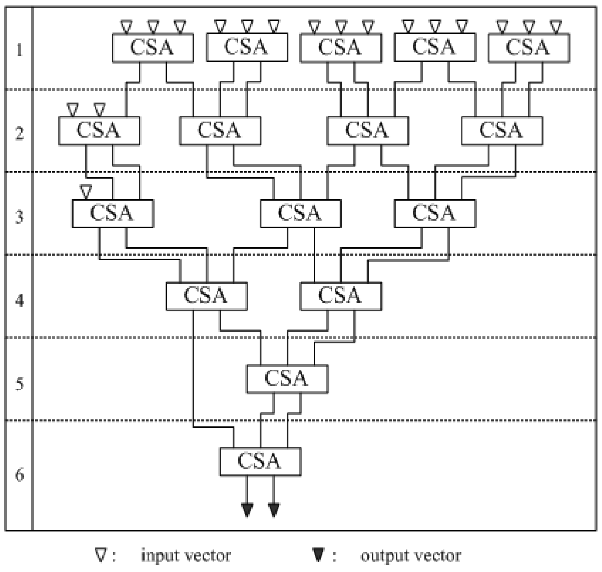

Fig.4 Wallace Tree based Addition

The Generated partial products using multiplexer can be added by using Wallace tree based to reduce the adding delay.

IV. CONCLUSION

The Implementation of Efficient matrix for high speed, less area and low power can be implemented by using the concept of multiplexer logic which consumes less area and suitable to handle more number of bit inputs.

REFERENCES

[1] Shu-Qing Li, Chi Hou Chan, Leung Tsan "Parallel Implementation of the Sparse-Matrix/Canonical Grid Method for the Analysis of Two-Dimensional Random Rough Surfaces (Three-Dimensional Scattering Problem) on a Beowulf System" IEEE Transactions on Geo science And Remote Sensing, Vol. 38, No. 4, July 2000.

[2] Nan Zhang "A Novel Parallel Scan for Multi core Processors and Its Application in Sparse Matrix-Vector Multiplication" IEEE Transactions On Parallel And Distributed Systems, Vol. 23, No. 3, March 2012.

[3] Bahram Hamraz, Nicholas HM Caldwell, and P. John Clarkson "A Matrix-Calculation-Based Algorithm for Numerical Change Propagation Analysis" IEEE Transactions on Engineering Management, Vol. 60, No. 1, February 2013.

[4] Vasileios Karakasis, Theodoros Gkountouvas, Kornilios Kourtis, Georgios Goumas, Nectarios Koziris "An Extended Compression Format for the Optimization of Sparse Matrix-Vector Multiplication" IEEE Transactions On Parallel And Distributed Systems- 2013.

[5] J. Jang, S. Choi, and V. Prasanna, “Energy-Efficient Matrix Multiplication on FPGAs”, International Conference on Field Programmable Logic and Applications, pp. 534-544, 2002.

[6] S. Belkacemi, K. Benkrid, D. Crookes, and A. Benkrid, “Design and Implementation of a High Performance Matrix Multiplier Core for Xilinx Virtex FPGA”,IEEE International Workshop on Computer Architectures for Machine Perception, pp. 156-159, 2003.

[7] L. Jianwen and J. C. Chuen, “Partially Reconfigurable Matrix Multiplication for Area and Time Efficiency on FPGAs”, Euro micro Symposium on Digital System Design, pp. 244-248, 2004.

[8] MahendraVucha and ArvindRajawat, “Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication”, International Journal of Computer Applications, Vol. 26, No.3,pp. 18-22, July 2011.

[9] Syed M. Qasim, Ahmed A. Telba and Abdulhameed Y. AlMazroo, “FPGA Design and Implementation of Matrix Multiplier Architectures for Image and Signal Processing Applications”, International Journal of Computer Science and Network Security, Vol.10, No.2, pp. 168-176, February 2010.

[10] C.Vinoth1, V. S. Kanchana Bhaaskaran2, B. Brindha, S. Sakthikumaran, V.Kavinilavu, B.Bhaskar, M. Kanagasabapathy and B. Sharath,” A Novel low power and high speed Wallace tree multiplier for risc processor”, C 978-1-4244-8679-3/11/$26.00 ©2011 IEEE.

[11] P. S. Tulasiram*, D. Vaithiyanathan, R. Seshasayanan, “ Implementation of Modified Booth Recoded Wallace Tree Multiplier for fast Arithmetic Circuits’. Vol.4, October 2014.