International Journal of Engineering Innovation & Research
Volume 3, Issue 2, ISSN: 2277 – 5668

A High Speed Matrix Multiplication Base Algorithm Implementation for Improved Computational Time and Throughput

Iffat Ashfaque, Shazia
Email: Iffath_2k@yahoo.com

***Abstract* – Matrix multiplication is used in applications such as signal and image processing, graphics and robotics etc. In this work we design and discuss the results of implementing matrix-vector multiplication which is computationally very exhaustive. It requires several multiply and add units. Processing system take numerous clock cycles to perform all the necessary multiply add operations. The overall performance is restricted by the number of multiplications and additions that could be done in parallel. Matrix multiplication architecture is based on serial-parallel sign multiplier and adder component. Our proposed design provides better speed, better throughput structural design for matrix multiplier as compared to antecedent reportable implementation of matrix operation with cut back gate count mistreatment VHDL.**

***Keywords* – HDL Synthesis, RTL, FIFO, Structural Style of Modelling, BlockRAM.**

I. INTRODUCTION

A matrix multiplication system is designed and implemented in this work. The basic operation to be perform is AX=Y, where A is an arbitrary size of matrix, X and Y are vectors with size of matrix A's column number. The figure below shows the 3X3 Matrix multiplication process.

$$\begin{bmatrix} A00 & A01 & A02 \\ A10 & A11 & A12 \\ A20 & A21 & A22 \end{bmatrix} X \begin{bmatrix} B00 & B01 & B02 \\ B10 & B11 & B12 \\ B20 & B21 & B22 \end{bmatrix} = \begin{bmatrix} Y00 & Y01 & Y02 \\ Y10 & Y11 & Y12 \\ Y20 & Y21 & Y22 \end{bmatrix}$$

The latency is defined as the time between reading the first elements from the input matrices, A and B, and writing the first element C to the result matrix. The total computation time is the time required for analysis between reading the first elements from the input matrices, A and B, and writing the final result matrix element C to memory. The structural design for the matrix multiplier unit consist of Memory blocks require to store the image pixel data, Control unit require to operate the submodules, Matrix arrangement of multiplier units to design the matrix of suitable size, FIFO (First In First out) which accumulate the multiplier outputs for cyclic addition, Adders and counters. The control logic will implement as a finite state machine with states and transition logic. The Start signal transitions the state machine out of the idle state and into the initialize state whereby it commands the multiplicand and multiplier to be loaded into registers. Once loaded, the state machine goes through a series of test and shift, or test, add and shift operations depending on the status of the LSB bit. Upon reaching the highest count for the multiplication cycle, the state machine goes back to the idle state and outputs a Stop signal.

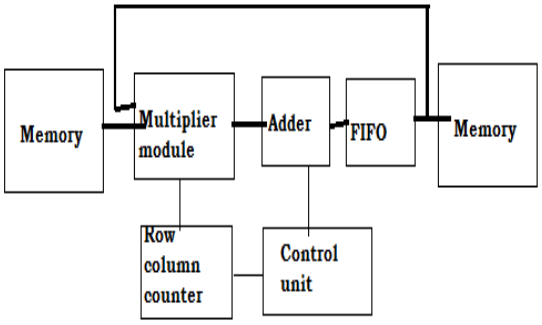

Fig.1. Block diagram of matrix-element multiplication

II. PROPOSE ALGORITHM

The method for matrix multiplication consumes lot of time, as it engage in reading the data from memory, multiplying data byte to data byte and again storing the result back in memory for serial addition, and this process has to be repetitive for each element of the matrix. The propose algorithm works as:

1. Generate the matrix pixel data.
2. Store the data in memory.
3. Split the design into functional modules, the entire design to be implemented to be broken into smaller modules.
4. Design Multiply the row ,column and matrix counters module for matix arrangements.
5. Accumulate the multiplier outputs and added results is store in distributed memory as per equations :
   Y00 = A00*B00 +A01*B10 + A02*B20
   Y01 = A00*B01 +A01*B11 + A02*B21
   Y02 = A00*B02 +A01*B12 + A02*B22
   Y10 = A10*B00 +A11*B10 + A12*B20
   Y11 = A10*B01 +A11*B11 + A12*B21
   Y12 = A10*B02 +A11*B12 + A12*B22
   Y20 = A20*B00 +A21*B10 + A22*B20
   Y21 = A20*B01 +A21*B11 + A22*B21
   Y22 = A20*B02 +A21*B12 + A22*B22
6. Adding the designed sub modules which are frequently use in design so that length of VHDL coding will shrink.
7. Using structural style of modelling be integrated all the modules to complete the circuit.

International Journal of Engineering Innovation & Research
Volume 3, Issue 2, ISSN: 2277 – 5668

Row and column counter is in used to count the number of input data that is read from the memory which is given to matrix arrangement logic. To read/write a data from the memory an address location is send. Hence the data will become available at the each clock cycle. As the data are retrieve from the memory start from the memory location of zero, for every nine clock-cycles the retrieve data is sent to the multiplier.

III. Adder Multiplier Module

The multiplier will have 8-bit inputs and 8-bit multiplicand as well as a Start signal. The multiplier will then calculate the result using the shift and add method and provide the 16-bit result along with a Stop signal. The design shall be coded in VHDL and simulated for proper functionality and timing implementations. Older multiplier architectures use a shifter and accumulator to sum each partial product, often one partial product per cycle, trading off speed for die area. Modern multiplier architectures use the Baugh–Wooley algorithm, Wallace trees, or Dadda multipliers to add the partial products together in a single cycle. Matrix multiplication structural design is based on serial-parallel sign multiplier and adder module, which shrink the number of partial products that must be summed.

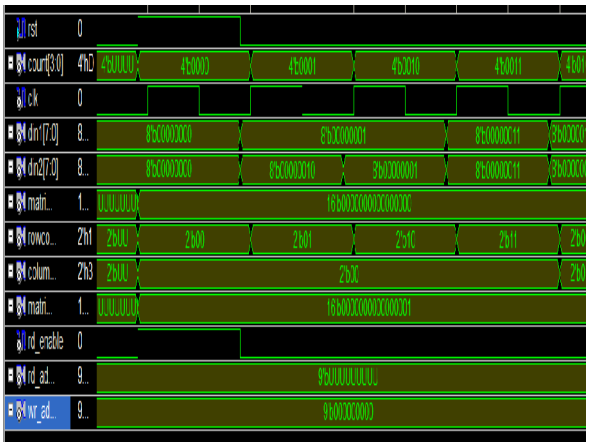

Fig.2. Timing simulation for matrix multiplication unit

Row and column counter is in used to count the number of input data that is read from the memory which is given to matrix arrangement logic. To read/write a data from the memory an address location is sent. Hence the data will become accessible at the each clock cycle. As the data are retrieve from the Block RAM start from the memory location of zero, for every nine clock-cycles the retrieved data is sent to the multiplier. The image pixel data can be written in memory when write enable signal is at low level depends on the address given at addr(5 :0) signal. The row count register counts the matrix byte data to set nXm matrix. As shown in above figure the matrix pixel data can be write in memory, at that time.

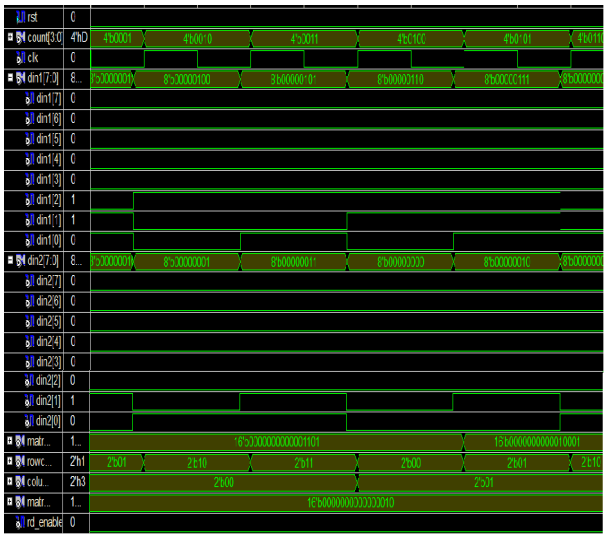

Fig.2. Timing simulation for matrix multiplication with row, column, and matrix counter

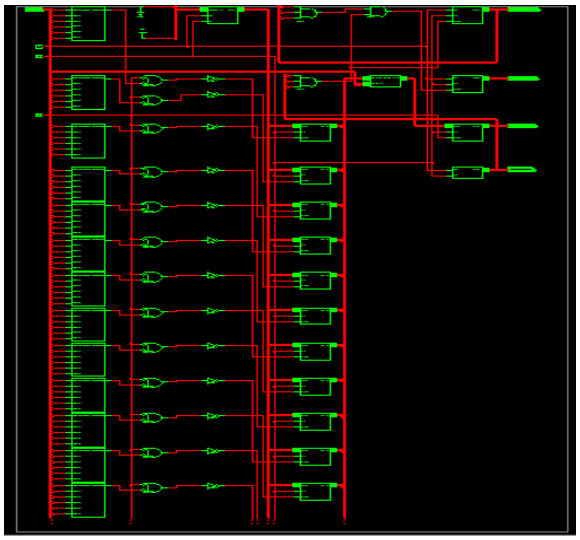

Fig.4. RTL view for Matrix formation unit

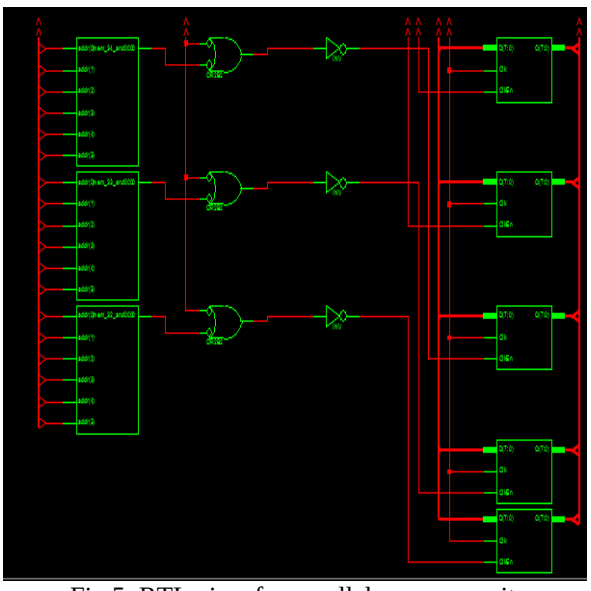

Fig.5. RTL view for parallel memory unit

HDL Synthesis Report Adder/Multiplier unit in processesing Element. The adder multiplier unit RTL view shows the 213 flipflops ,48 8 bit multiplier elements modules, 32 adder modules and three eight bit 64X1 multiplexer cells.

IV. CONCLUSION

Matrix multiplication architecture is based on serial-parallel sign multiplier and adder module. Our proposed design provides improved speed, improved throughput structural design for matrix multiplier as compared to earlier reported implementation of matrix multiplication with reduce gate count using VHDL. Algorithm proposed removes the intercommunication between parallel processing elements (PEs), and allows each PE to operate in isolation. Also, this algorithm allows the implementation using matrices of random dimension. HDL Synthesis Report Adder/Multiplier unit in processesing Element. The adder multiplier unit RTL view shows the 213 flipflops ,48 8 bit multiplier elements modules, 32 adder modules and three eight bit 64X1 multiplexer cells.

REFERENCES

[1] Syed M. Qasim, Shuja A. Abbasi,and Bandar A. Almashary "Throughput Latency Implementation ofMatrix Multiplication using Field ProgrammableGateArray" IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 26, No. 6, Nov2012.

[2] Ju-Wook Jang,Seonil B. Choi, Member, and Viktor K. Prasanna "Energy- and Time-Efficient Matrix Multiplication on FPGAs" IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 13, No. 11, November 2005 pp no. 1305.

[3] Tai-Chi Lee, Mark White, and Michael Gubody "Matrix Multiplication on FPGA-Based Platform" Proceedings of the World Congress on Engineering and Computer Science 2013 Vol I WCECS 2013, 23-25 October, 2013, San Francisco, USA.

[4] Ms. Shamshad Shirgeri,Ms. Pallavi Umesh Naik ,Mrs. Rohini, Prof. Krishnananda Shet "Reconfiguration of Memory for High Speed Matrix Multiplication" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, No 5, May 2013.