Fast Matrix Multiplication IP for Face Recognition Applications on Xilinx SOC FPGA (Zybo Board)

Sai Agnihotri, Chang Choo

Department of Electrical Engineering, San Jose State University, San Jose, California 95192

Introduction

Face recognition has become extremely critical in many applications. The code profiling study shows that matrix multiplication is the most computation-intensive function of the algorithm and accounts for 80% of the total time. Therefore, the objective of this project was to develop a FPGA based fast matrix multiplication unit which can be used as a hardware accelerator in face recognition systems. The matrices of 32-bit-fixed point unsigned integers were subdivided to form the blocks, which were multiplied in parallel to utilize the resources available on FPGA. The design was modeled in Verilog-HDL and simulated using Xilinx Vivado 2014.3 tool. The synthesis was done by targeting the Zynq-7000 FPGA on Zybo board. For realization of this unit 3952 LUTs, 48 DSP48 slices and 128KB BRAM was required. The design was successfully tested on 100 MHz frequency. For real-time face recognition, the designed unit will take 4.5ms for one QVGA frame

Methodology

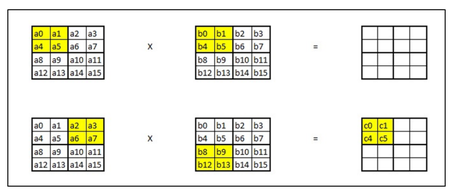
Block Based Matrix Multiplication

As the name suggests block based matrix multiplication is the multiplication of large matrices that are divided into smaller ones and the result of the smaller matrices are added to form the final result. A matrix can be visualized as containing elements which can be broken down to form smaller matrices. For example a 16x16 matrices can be broken down four 8x8 matrices, a 8x8 matrix can be broken down to four 4x4 matrices and a 4x4 matrix can be broken down to four 2x2 matrices.

Shown below is the example of 4x4 matrix multiplication using 2x2 matrix as the base block:

$$P = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 12 & 13 & 14 & 15 \end{bmatrix} \qquad P = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix} \quad C = \begin{bmatrix} 9 & 10 \\ 12 & 13 \end{bmatrix} \quad D = \begin{bmatrix} 11 & 12 \\ 14 & 15 \end{bmatrix}$$



In this project the base block for matrix multiplication is considered as 4x4 which is implemented on the hardware.

Design Approach

The matrix multiplication is implemented in software as well as hardware to do the comparative analysis of the time required in both the cases.

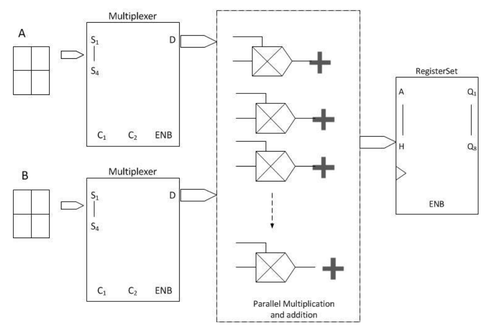
Software Implementation

Pseudo code for block matrix multiplication is as shown:

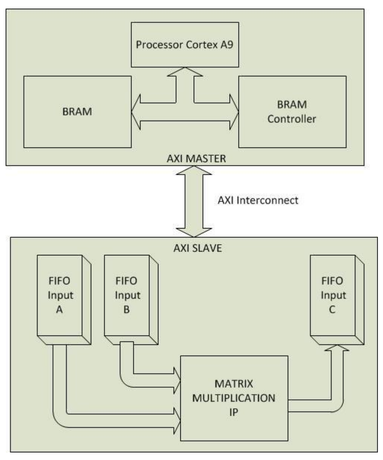
```
for (i=0; i<(MAT_ROW/BLK_SIZE); i=i+1)
for (j=0; j<(MAT_COL/BLK_SIZE); j=j+1) {
    for (Li=0; Li<BLK_SIZE; Li=Li+1)
        for (Lj=0; Lj<BLK_SIZE; Lj=Lj+1)
            Cout[i*BLK_SIZE+Li][ j*BLK_SIZE+Lj] = 0;
        for (p=0; p<(MAT_ROW/BLK_SIZE); p=p+1)
        for (k=0; k<BLK_SIZE; k=k+1) {
            for (Li=0; Li<BLK_SIZE; Li=Li+1)
                for (Lj=0; Lj<BLK_SIZE; Lj=Lj+1)
                    Cout[(i*BLK_SIZE)+Li][(j*BLK_SIZE)+Lj] = Cout[(i*BLK_SIZE)+Li][(j*BLK_SIZE)+Lj] + ((A[(i*BLK_SIZE)+Li][(p*BLK_SIZE)+k])
                }
    }
```

Hardware Implementation

This design is 4- stage pipelined and makes use of 16 multipliers and 20 adders in parallel.

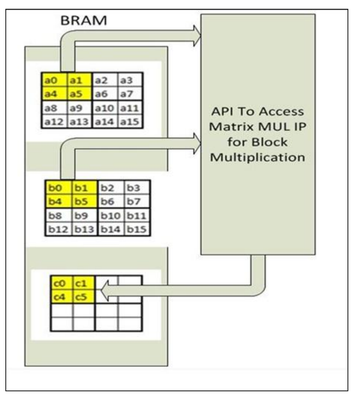


Design supports AXI-4 Lite protocol in slave mode and synchronization is done using FIFO.



Matrix multiplication System working

After the whole system is build we have combined the hardware and the software to get the actual implementation. API is called from the software, to perform block matrix multiplication using hardware.

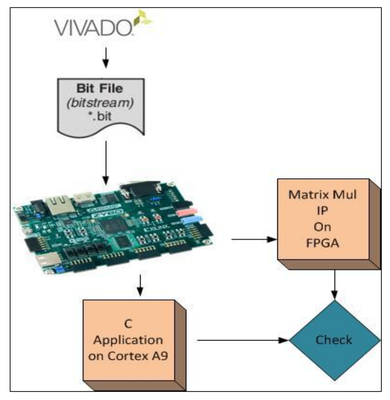


Results

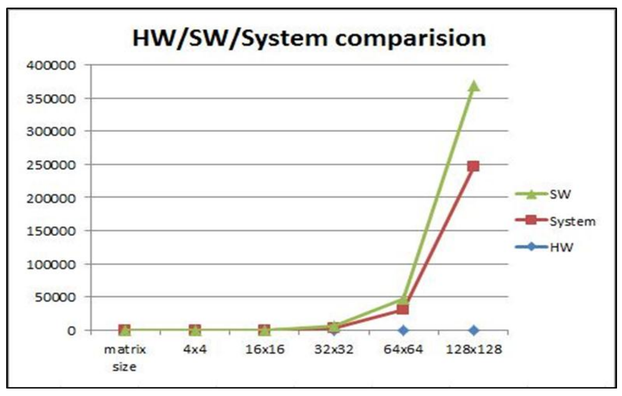
Design is synthesized by targeting Zynq 7000 SOC and post implementation device utilization table is as shown below:

Resource	Utilization	Available	Utilization%
FF	6118	35200	17.38
LUT	3952	17600	22.45
Memory LUT	154	6000	2.57
BRAM	40	60	66.67
DSP48	48	80	60.00
BUFG	1	32	3.12

System test on Zybo board : The IP is integrated with the processor in ZynQ SOC and verified with Software application.



The comparative analysis shows that the efficiency of matrix multiplication is improved when a HW IP is used. The graph shows the comparison:



Conclusions

- Fast Matrix Multiplication IP for face recognition application is successfully implemented on Xilinx ZynQ-7000 SOC FPGA using Zybo board.
- Compatible with AXI-4 lite protocol and customizable user option for truncation and rounding for different precisions.
- 4.5ms per QVGA frame efficiency achieved

Key References

[1] Sotiropoulos, I; Papaefstathiou, I, "A fast parallel matrix multiplication reconfigurable unit utilized in face recognitions systems," Field Programmable Logic and Applications, 2009.

[2] Shimin Wang, Jihua Ye, “Research and Implementation of Embedded Face Recognition System Based on ARM9”, Mechanic Automation and Control Engineering (MACE), 2010 International Conference on 26-28 June 2010, WuhanH.

[3] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg, “Face Recognition by Elastic Bunch Graph Matching”, Image Processing, 1997. Proceedings, International Conference on (Volume:1) 26-29 Oct 1997, Santa Barbara, CA.

[4] Chin-Shu Chang, Kuo-Kuang Chen, Teh-Lu Liao, Po-Yun Hsu, “Human Face Recognition System Using Modified PCA Algorithm and ARM Platform”, Computer Communication Control and Automation (3CA), 2010 International Symposium on (Volume:2)5-7 May 2010, Tainan

.

Acknowledgments

I would like to thank my project advisor, Dr Chang Choo for his support, guidance and undivided attention towards making this project a success. I would also like to take this opportunity to thank San Jose State University, Electrical Engineering Department and the Digital Signal Processing/FPGA lab. I would like to thank my family, friends, and relatives for the relentless motivation and support throughout this project