

FPGA-based Accelerator Platform for Big Data Matrix Processing

Ching-Che Chung, *Member, IEEE*, Chun-Kai Liu, and Dai-Hua Lee

Department of Computer Science and Information Engineering

National Chung Cheng University

No. 168, University Rd., Min-Hsiung, Chia-Yi, Taiwan

Email: wildwolf@cs.ccu.edu.tw

Abstract—Big data analytics requires to analyze data at the rate that matches the speed of data production. Therefore, some software frameworks such as Hadoop with high scalability and fault tolerance had been proposed to enable massive data storage and processing over large clusters of computing servers. However, the performance of data analytics can be further improved by deploying hardware accelerators to the computing servers. In this paper, an FPGA-based hardware accelerator platform for big data matrix processing is presented. The proposed accelerator platform is composed of many FPGA evaluation boards (EVBs). The computing server communicates with FPGA EVBs with Gigabit Ethernet. In addition, the FPGA can be reprogrammed for different data processing operations with high flexibility. The experimental results for one hundred 512x512 floating point matrix multiplications show that the proposed hardware accelerator platform with four FPGA EVBs at 125MHz clock rate can achieve the 4x speedup as compared with the computing server with an Intel I7-4770 CPU at 3.4GHz.

Keywords—Big data analytics, hardware accelerator, cloud computing, Hadoop.

I. INTRODUCTION

Nowadays, the growing popularity of web systems, mobile devices, surveillance videos, and wireless sensors generate large amounts of data from different sources. In fact, all of the industries need to confront the issues of big data analytics. For example, financial institutions can calculate the risk by analysis of data [3]; information technology industries can find the hidden value or solve problem by analysis of logs [3]. Big data is the term for collection of complex data sets that it makes difficult to manage, analyze and process using the traditional database system [1]. Big data includes activity logs, business transaction, images, and surveillance videos that can reach massive proportions over time [2]. In some statistics, those data generated exceed 2.5 quintillion bytes everyday [1]. In 2011, the volume of data reaches the Petabyte to Exabyte magnitude [7]. The velocity of data generation has gone beyond our imagination.

The properties of big data make it is not easy to handle. For instance, the properties include variety, volume, velocity and value, the “4Vs” is widely applied to the definition of big data [7]. The variety means the data produced is not of one flavor, they have structured, semi-structured and unstructured data, so traditional database systems are hard to handle them. The volume means the volume of big data is quite larger than traditional data. The velocity means big data must be analyzed at a rate that matches the speed of data production. Finally, by

analyzing big data, some useful values can be found, for instance, business trends and commercial benefits.

To deal with massive data processing, many software frameworks are developed, such as Hadoop and GridGain. Hadoop is an open-source software framework that enables massive data storage and distributed processing over large clusters of computing servers. It is mainly composed of two modules: Hadoop distributed file system (HDFS) and MapReduce. In HDFS, a file is split into one or more blocks, and each block has several replications to prevent missing data. The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster node. The master JobTracker schedules jobs for the slaves, and monitors and re-executing the failed tasks. The MapReduce framework enables the automatic paralleling and distribution of large-scale computation applications on large clusters of computing servers. Therefore, it becomes easier to implement big data analysis applications.

Besides the development of software frameworks, the computing servers also require new system capabilities. In [6], the IBM Power8 processor doubles L1 to L3 data cache size per core for big data analytics. In addition, the execution functional units are also increased to enhance per-core throughput. Since server workloads will continue to evolve, the IBM Power8 processor introduces the coherent accelerator processor interface (CAPI) to support the general purpose cores for a heterogeneous computing solution with off-chip hardware accelerators. These accelerators can be plugged into PCIe slots and implemented in FPGA or ASIC chips. In [5], the similar hybrid CPU/FPGA architecture is discussed. Since it is hard to calculate large amount of data by only CPUs, the FPGA can help to enhance the throughput of data processing.

In this paper, an FPGA-based hardware accelerator platform for big data matrix processing is presented. The proposed accelerator platform is composed of many VC707 FPGA evaluation boards (EVBs) [8]. The computing server communicates with FPGA EVBs with Gigabit Ethernet. In big data analysis, the data processing often includes many matrix operations. The matrix multiplication has lots of floating point multiplications and floating point additions. As the matrix size is increased, the total execution time with only CPUs is not acceptable. Therefore, in the proposed hardware accelerator platform, the workloads are shared to many VC707 EVBs. The experimental results show that for one hundred 512x512 floating point matrix multiplications, the proposed hardware accelerator platform with four FPGA EVBs at 125MHz clock

This work was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST-103-2221-E-194-063-MY3.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} & b_{18} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & b_{27} & b_{28} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} & b_{38} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} & b_{47} & b_{48} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} & b_{57} & b_{58} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} & b_{67} & b_{68} \\ b_{71} & b_{72} & b_{73} & b_{74} & b_{75} & b_{76} & b_{77} & b_{78} \\ b_{81} & b_{82} & b_{83} & b_{84} & b_{85} & b_{86} & b_{87} & b_{88} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} & c_{17} & c_{18} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} & c_{27} & c_{28} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} & c_{36} & c_{37} & c_{38} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} & c_{47} & c_{48} \\ c_{51} & c_{52} & c_{53} & c_{54} & c_{55} & c_{56} & c_{57} & c_{58} \\ c_{61} & c_{62} & c_{63} & c_{64} & c_{65} & c_{66} & c_{67} & c_{68} \\ c_{71} & c_{72} & c_{73} & c_{74} & c_{75} & c_{76} & c_{77} & c_{78} \\ c_{81} & c_{82} & c_{83} & c_{84} & c_{85} & c_{86} & c_{87} & c_{88} \end{bmatrix} \quad (1)$$

rate can achieve 4x speedup as compared with the computing server with an Intel I7-4770 CPU at 3.4GHz.

The rest of this paper is organized as follows: Section II describes the proposed hardware accelerator architecture with VC707 EVBs. The proposed algorithm for large matrix multiplications is presented in Section III. Section IV shows the experimental results. Finally, the conclusion is given in Section V.

II. PROPOSED ACCELERATOR ARCHITECTURE

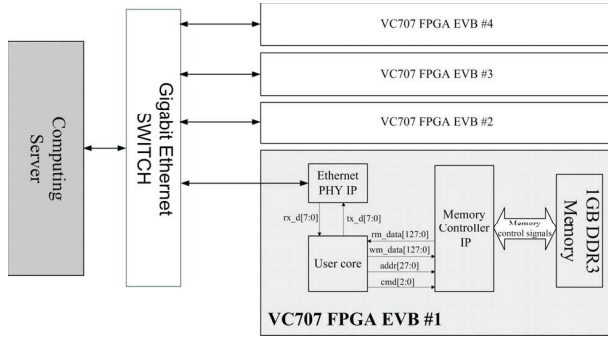


Fig. 1. The proposed FPGA-based hardware accelerator platform with VC707 evaluation boards.

The proposed FPGA-based hardware accelerator platform with VC707 EVBs is shown in Fig. 1. The computing server wraps the data into packets and sends them to VC707 EVBs through a Gigabit Ethernet switch. Then, the workloads of the computing server can be shared in FPGA EVBs. In a VC707 FPGA EVB, Ethernet physical layer IP is used to collect the packets sent from the computing server, and the computation results of the FPGA can be also sent back to the computing server through the Ethernet physical layer IP. In the proposed design, 32-bit single precision floating point operations are supported. Therefore, four 32-bit single precision floating point numbers can be combined into 128-bit data during read and write operations.

The DDR3 memory interface controller IP helps the user core to communicate with the on-board 1GB DDR3 memory. The commands (cmd) and addresses (addr) can be sent to the memory interface controller IP simultaneously. For a write request, the written data (wm_data) should be prepared before sending write command to the memory interface controller IP. After a write request is finished, the ready signal provided by

the memory interface controller IP indicates the completion of the write request to the DDR3 memory. For a read request, after several cycles, the read data (rm_data) are output by the memory interface controller IP.

The proposed matrix operation is designed in the user core module. It manages the data flow of all matrix processor in the matrix operation module. A large size matrix multiplication operation is split into many small size matrix multiplications in different matrix processors in parallel. Then, the computation results are combined, and the final answer are sent back to the computing server. Obviously, the execution time can be reduced by the proposed hardware accelerator platform.

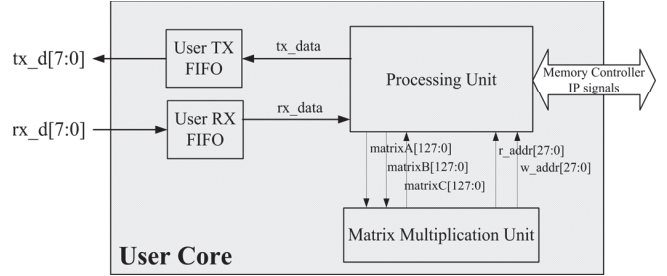


Fig. 2. Modules in user core.

The user core is composed of a TX FIFO, a RX FIFO, a processing unit, and a matrix multiplication unit, as shown in Fig. 2. The RX FIFO receives data (rx_d) from the Ethernet physical layer IP and combines them into rx_data. The TX FIFO sends the byte data (tx_d) split from the packet (tx_data) to the Ethernet physical layer IP. The processing unit controls the state machines and manages the data flow. The matrix operation module computes matrix multiplications with many small matrix processors in parallel that helps the computing server to quickly complete the matrix multiplication.

The behavior of the user core is described as follows. First, the processing unit receives data from RX FIFO, and these data are stored in the DDR3 memory. Then, the large size matrix multiplication is split into many small matrix multiplications. Subsequently, these small matrices A and B are sent to the matrix multiplication unit. After the temporal results of matrix C are obtained, these temporal results are combined in the processing unit to obtain the final answer and are stored in the DDR3 memory. Finally, the matrix multiplication results are sent back to the computing server through the Ethernet physical layer IP.

$$A = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \\ X_{31} & X_{32} \\ X_{41} & X_{42} \\ X_{51} & X_{52} \\ X_{61} & X_{62} \\ X_{71} & X_{72} \\ X_{81} & X_{82} \end{bmatrix} \quad B = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} & Y_{17} & Y_{18} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} & Y_{27} & Y_{28} \end{bmatrix} \quad C = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \\ R_{31} & R_{32} \\ R_{41} & R_{42} \\ R_{51} & R_{52} \\ R_{61} & R_{62} \\ R_{71} & R_{72} \\ R_{81} & R_{82} \end{bmatrix} \quad (2)$$

$$c_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21} + a_{13} \cdot b_{31} + a_{14} \cdot b_{41} + a_{15} \cdot b_{51} + a_{16} \cdot b_{61} + a_{17} \cdot b_{71} + a_{18} \cdot b_{81} \quad (3)$$

$$R_{11} = [c_{11} \ c_{12} \ c_{13} \ c_{14}]$$

$$= [X_{11} \cdot Y_{11} \ X_{11} \cdot Y_{12} \ X_{11} \cdot Y_{13} \ X_{11} \cdot Y_{14}] + [X_{12} \cdot Y_{21} \ X_{12} \cdot Y_{22} \ X_{12} \cdot Y_{23} \ X_{12} \cdot Y_{24}] \quad (4)$$

III. MATRIX MULTIPLICATION UNIT

For two floating point matrices A and B that both sizes are 8×8, the matrix multiplication result C is also 8×8, as expressed in Eq.1. If we define X_{ij} , Y_{ij} , and R_{ij} , as expressed in Eq. 2, the computation results for c_{11} , which is expressed in Eq. 3, can be further rewritten as Eq. 4, where the size of X_{ij} is 1×4, Y_{ij} is 4×1, and R_{ij} is 1×4. Thus, the first term in the Eq. 4 can be computed with four matrix processors in parallel. Then, the temporal results are stored and waiting for the second term in the Eq. 4 is computed. Finally, the R_{ij} can be computed with only four small matrix processors after several iterations.

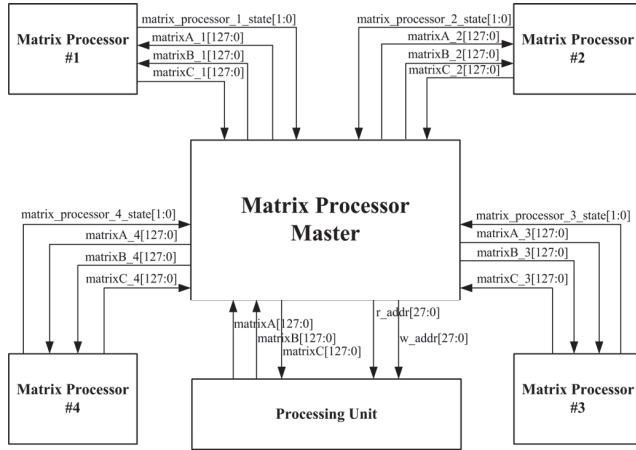


Fig. 3. The matrix multiplication unit.

Fig. 3 shows the architecture of the matrix multiplication unit. It is composed of four matrix processors and one matrix processor master. The matrix processor master reads matrix elements from the processing unit and sends the small matrices A and B to four matrix processors for computing results in parallel. When matrix processors are idle, the matrix processor master will dispatch job for them. Fig. 3 only shows the operation for 8×8 matrix multiplication. For trade-off between the usage percentage of the FPGA hardware resources and the execution time of the large matrix multiplication. The maximum number of matrix processor in the proposed matrix multiplication unit is 16.

IV. EXPERIMENTAL RESULTS

In the proposed FPGA-based hardware accelerator platform, the data transmission time between the computing server and the hardware accelerator platform depends on the I/O speed of the Ethernet physical layer IP. When data are sent from the computing server to the DDR3 memory of the hardware accelerator platform, the transmission data rate is tested and shown in Table I. Oppositely, the transmission data rate from the DDR3 memory of the hardware accelerator platform to the computing server is also tested, and shown in Table I.

TABLE I. NETWORK TRANSMISSION DATA RATE

Direction	Packet length	Number of Packet	Time	Transmission data rate
Server to VC707	1500bytes	50,000	5.375s	111.63Mbps
VC707 to Server	1500bytes	50,000	11.679s	51.37Mbps

Table II shows the hardware resource utilization of the proposed hardware accelerator for big data matrix multiplications. The number of matrix processor in the proposed matrix multiplication unit is 16 even for large size matrix multiplication. As shown in Table II, the number of matrix processor in the proposed matrix multiplication can be increased if shorter execution time is required.

TABLE II. FPGA RESOURCE UTILIZATION

Slice Logic Utilization	Used	Available	Utilization
Number of Slice registers	194,312	607,200	32%
Number of Slice LUTs	151,370	303,600	49%
Number of DSP48E1	1,281	2,800	45%

Table III shows the timing profile analysis for the computing server calculating thirty 512×512 floating point matrix multiplications. It takes a lot of time for the computing server to read data from the main memory. In addition, for large matrix multiplication, the amount of data is often larger

than the capacity of data caches of the CPU of the computing server, and therefore cache miss will often occur in the CPU of the computing server.

TABLE III. TIMING PROFILE ANALYSIS FOR THE COMPUTING SERVER

I7-4770 (3.4GHz)				
Total execution time	24.944 s	Memory read time	18.315 s	73.42 %
		Memory write time	0.033 s	0.13 %
		Computing time	6.592 s	26.46 %

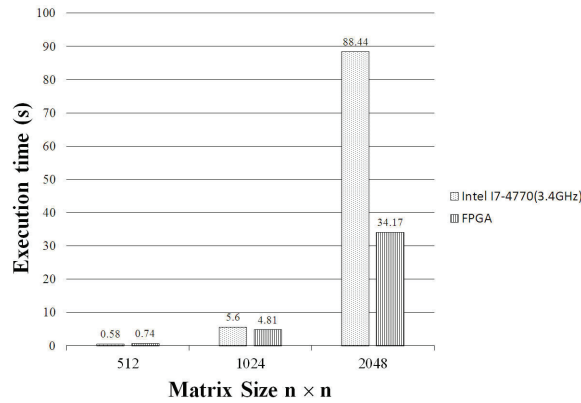


Fig. 4. Compare Intel I7-4770 (3.40GHz) with VC707 (125MHz) with different matrix size.

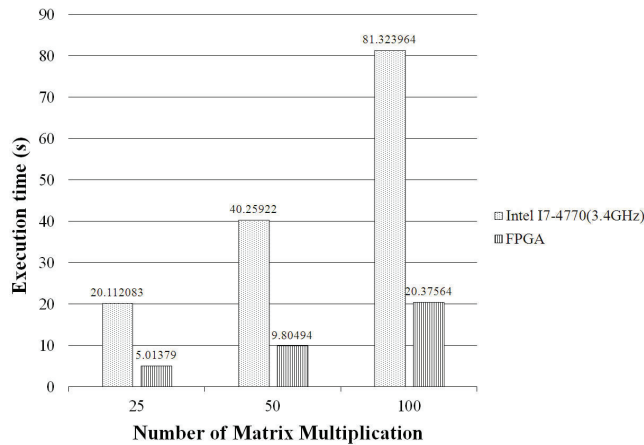


Fig. 5. Compare Intel I7-4770 (3.4GHz) with VC707 (125MHz) with different number of matrix multiplications.

Fig. 4 shows the execution time of one floating point matrix multiplication for the computing server with an Intel I7-4770 and the proposed hardware accelerator with one VC707 EVB. As shown in Fig. 4, if the matrix size becomes larger, the speedup of the proposed hardware accelerator can be increased. In addition, in small size matrix multiplication, the data transmission in the Ethernet will be the bottleneck in the proposed hardware accelerator.

Fig. 5 shows the execution time for 25, 50, and 100 times 512x512 floating point matrix multiplications. The workloads

are shared with four VC707 FPGA EVBs. The proposed hardware accelerator platform with four FPGA EVBs at 125MHz clock rate can achieve 4x speedup as compared with the computing server with an Intel I7-4770 CPU at 3.4GHz.

V. CONCLUSION

In this paper, an FPGA-based hardware accelerator platform for big data matrix processing is presented. The proposed accelerator platform can use many VC707 FPGA EVBs to speed up the big data matrix processing. Since server workloads will continue to evolve, the proposed FPGA-based hardware accelerator platform provides an easy way to support the CPUs for a heterogeneous computing solution with off-chip hardware accelerators. The experimental results for one hundred 512x512 floating point matrix multiplications show that the proposed hardware accelerator platform with four FPGA EVBs at 125MHz clock rate can achieve the 4x speedup as compared with the computing server with an Intel I7-4770 CPU at 3.4GHz.

ACKNOWLEDGMENT

The authors would like to thank their colleagues in the Silicon Sensor and System (S3) Laboratory of National Chung Cheng University for many fruitful discussions. The EDA tools supported by National Chip Implementation Center (CIC) are acknowledged as well.

REFERENCES

- [1] Udaigiri Chandrasekhar, Amarendra Reddy and Rohan Rath, "A comparative study of enterprise and open source big data analytical tools," in Proceedings of IEEE Conference on Information and Communication Technologies (ICT), Apr. 2013, pp. 372-377.
- [2] Jinson Zhang and Mao Lin Huang, "5Ws model for bigdata analysis and visualization," in Proceedings of IEEE Conference on Computational Science and Engineering (CSE), Dec. 2013, pp. 1021-1028.
- [3] Avita Katal, Mohammad Wazid, and R. H. Goudar, "Big data: issues, challenges, tools and good practices," in Proceedings of Sixth International Conference on Contemporary Computing (IC3), Aug. 2013, pp. 404-409.
- [4] Eser Kandogan, Mary Roth, Cheryl Kieliszewski, Fatma Özcan, Bob Schloss and Marc-Thomas Schmidt, "Data for all: a systems approach to accelerate the path from data to insight," in Proceedings of IEEE International Congress on Big Data (BigData Congress), Jun. 2013, pp. 427-428.
- [5] David Andrews, Douglas Niehaus and Peter Ashenden, "Programming models for hybrid CPU/FPGA chips," *Computer*, vol. 37, no. 1, pp. 118-120, Jan. 2004.
- [6] Joshua Friedrich, et al., "The POWER8™ processor: designed for big data, analytics, and cloud environments," in Proceedings of IEEE International Conference on IC Design and Technology (ICIDT), May 2014.
- [7] Han Hu, Yonggang Wen, Tat-Seng Chua and Xuelong Li, "Toward scalable systems for big data analytics: a technology tutorial," *IEEE Access*, vol. 2, pp. 652-687, Jul. 2014.
- [8] VC707 evaluation board for the Virtex-7 FPGA user guide, Xilinx Inc., Available: http://www.xilinx.com/support/documentation/boards_and_kits/vc707/ug885_VC707_Eval_Bd.pdf, Sep. 2014.