

# Vector Processing Support for FPGA-Oriented High Performance Applications

Hongyan Yang, Shuai Wang, Sotirios G. Ziavras and Jie Hu  
New Jersey Institute of Technology  
Department of Electrical and Computer Engineering  
University Heights, Newark, NJ 07102  
{hy34, sw63, ziavras, jhu}@njit.edu

## Abstract

*In this paper, we propose and implement a vector processing system that includes two identical vector microprocessors embedded in two FPGA chips. Each vector microprocessor supports floating-point calculations and efficient sparse matrix operations. Dense matrix-matrix multiplication and sparse matrix-vector multiplication with benchmark matrices from various application domains were run on the system to evaluate its performance. The resulting calculation times are compared with those of a commercial PC to show the effectiveness of our approach.*

## 1. Introductions

A vector processor is efficient in extracting data parallelism and can greatly speedup applications with array-intensive operations. Although it is not flexible for other kinds of parallelism, this inflexibility also simplifies its hardware implementation and results in low power consumption [3]. As shown in [5], the degree of vectorization can exceed 90% for most benchmarks in the EDN Embedded Microprocessor Benchmark Consortium (EEMBC) [1]. Thus, a vector microprocessor could be a very competitive choice for System-On-a-Chip (SOC) designs. In this paper, we propose a vector processing system that includes two vector microprocessors embedded in two FPGA chips. Sparse matrix-vector multiplication and floating-point matrix multiplication on FPGAs have been implemented in [4, 6], but they focus on application-specific matrices while our goal is to build a generic vector processing system.

## 2. architecture of the system

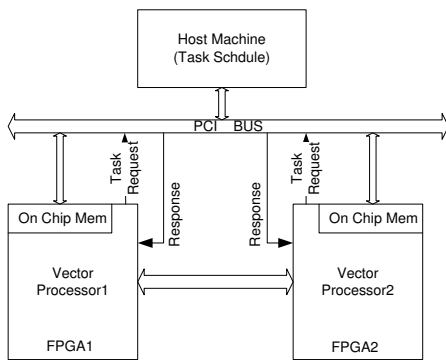
Our vector processing system includes two identical programmable vector microprocessors located on two FPGA

chips. This system is implemented on the Annapolis Wild-Star II board which has two Xilinx Virtex-II XC2V6000-5 chips. The Annapolis board is mounted on the host machine through a PCI socket [2]. Our two vector processors communicate with the host machine via the on-chip dual port memories. The host machine assigns work to them based on their requests and load balance. Each vector processor runs at 70MHz, and supports the IEEE 754 single-precision floating-point standard and efficient implementation for sparse matrices. The overview of the system is shown in Fig. 1.

Each of the vector microprocessors is composed of a vector core and a tightly coupled five-stage pipelined scalar unit. A vector register file organized in eight banks is located at the center of the vector unit. Each bank has three read ports and one write port. These ports are connected to the addition, multiplication and vector memory control units in a time-multiplexed way. The arithmetic units and data memory are also organized in eight banks to match the structure of the vector register file. Eight single-precision floating-point results from the addition/multiplication units can be generated every clock cycle or eight data can be loaded/stored at top performance. The system supports 16 scalar mode instructions and 8 vector instructions; the former support memory accesses, arithmetic operations and data transfers. The vector instructions support vector load/store, vector indexed load/store, vector multiplication/addition and vector-scalar multiplication/addition.

## 3 Performance results and conclusion

To test the performance of our vector processing system, dense matrix-matrix multiplications and sparse matrix-vector multiplications were run on it. Comparisons are made with the calculation time on a commercial Dell PC that contains a 1.2G Pentium-III processor, 512M bytes of memory and 512K bytes of L2 cache, and employs the



**Figure 1. Overview of the system**

Linux operating system.

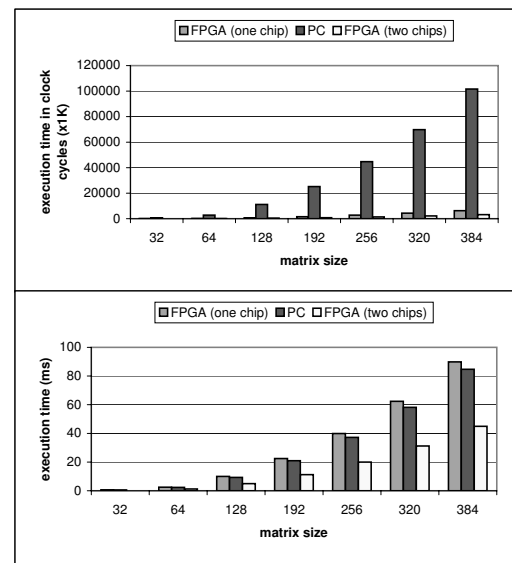
Fig. 2 shows the performance of matrix-matrix multiplication on one vector processor, the commercial PC and two vector processors. When the matrix-matrix multiplications are divided to run on two vector processors residing on two FPGAs, the calculations time is reduced approximately by half since equivalent numbers of calculations are assigned to the two vector processors. Our vector processing system can run almost two times fast than the commercial PC despite its much lower frequency (70Mhz of the vector processing system and 1.2G of the Dell PC).

Sparse matrices from various engineering application areas were run on our vector processing system for evaluation. For larger matrices having more non-zeros, the calculation is divided into the two vector processors by dividing the matrix into sub-blocks. Since the distribution of non-zeros is not balanced in these sub-blocks as in dense matrices, the calculation time on the two processors varies from 50% to 100% of the time on one processor.

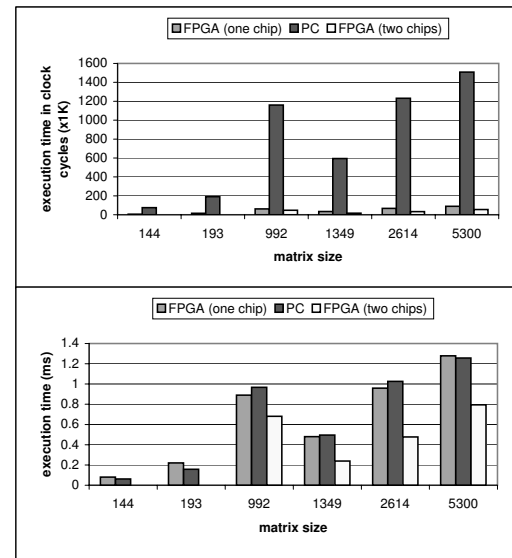
In this paper we presented the design and implementation of a vector processing system implemented on FPGAs. With abundant calculation units, tightly coupled memory and a well designed data storage scheme, our vector processing system can outperform a commercial PC on array-intensive problems despite its much lower frequency. Besides the examples listed in the paper, other applications having abundant data parallelism can also get similar benefits on our vector processing system.

## References

[1] The Embedded Microprocessor Benchmark Consortium.  
 [2] The WildstarII datasheet.  
 [3] K. Asanović. *Vector Microprocessors*. PhD thesis, University of California, Berkeley, 1998.  
 [4] Y. Dou, S. Vassiliadis, G. Kuzmanov, and G. Gaydadjiev. 64-bit floating-point FPGA matrix multiplication. In *Proc. of the ACM/SIGDA 13th International Symposium on FPGA*, pages 86–95, February 2005.



**Figure 2. Performance of dense matrix-matrix multiplication.**



**Figure 3. Performance of sparse matrix-vector multiplication.**

[5] C. Kozyrakis and D. Patterson. Vector vs. superscalar and VLIW architectures for embedded multimedia benchmarks. In *35th Annual IEEE/ACM International Symposium*, pages 283 – 293, 2002.  
 [6] L. Zhuo and V. K. Prasanna. Sparse matrix-vector multiplication on FPGAs. In *FPGA '05: Proc. of the 2005 ACM/SIGDA 13th international symposium on FPGA*, pages 63–74, New York, NY, USA, 2005. ACM Press.