RIKEN Center for Computational Science

大規模並列数値計算技術研究チーム
Large-scale Parallel Numerical Computing Technology Research Team

Design of an FPGA-based Matrix Multiplier with Task Parallelism

Yiyu Tan and Toshiyuki Imamura
RIKEN Center for Computational Science
Email: tan.yiyu@riken.jp; imamura.toshiyuki@riken.jp

(i) Introduction

- Matrix multiplication is a fundamental operation of linear algebra, and has been widely applied in high performance computing (HPC) to solve science and engineering problems. It requires computer systems have huge computing capacity, data throughput, and memory bandwidth as problem size is increased. How to improve the performance of matrix multiplication at hardware and software levels is still an open problem.
- Current HPC systems face power problems, and how to reduce power consumption while increasing performance is becoming one of the core concerns.
- FPGAs have much higher energy efficiency over other computation devices, and the design flow of FPGAs becomes much easier owing to the development of high-level synthesis in recent years.
- This poster presents an FPGA-based matrix multiplier with task parallelism, in which the systolic array architecture, data reuse and optimization techniques are applied to improve system performance and energy efficiency.

(ii) System Design and Implementation

The system is designed using OpenCL, and consists of different modules with each being single work-item. The main features are

- The computation kernel is a **10 x 16 systolic array (Fig. 1) [1]**, and **the data vectorization is 8 (Fig. 2)**. Except the kernels loading matrices A and B, the other kernels are **autorun** to reduce overhead.
- **High-speed and high-bandwidth channels** are applied to connect PEs and kernels.
- Data of matrix A are shifted from **left to right** while data of matrix B are moved from **top to bottom** in the systolic array to reuse data (Fig. 1).
- Table 1 shows the hardware resource utilization. Compared with the Intel's OpenCL example with data parallelism, the proposed matrix multiplier utilizes more DSP blocks and achieves much higher clock frequency.

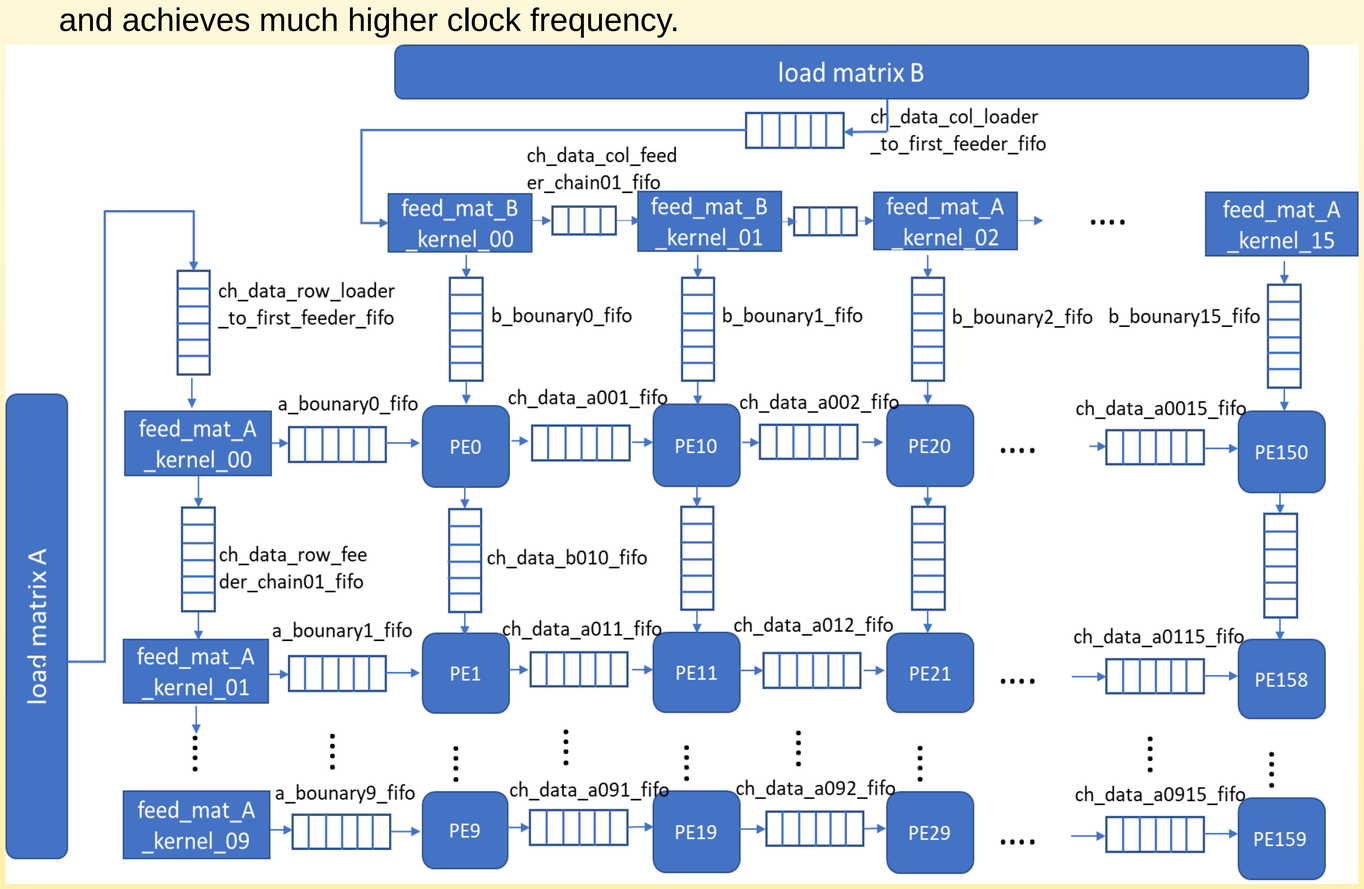


Fig. 1 System diagram

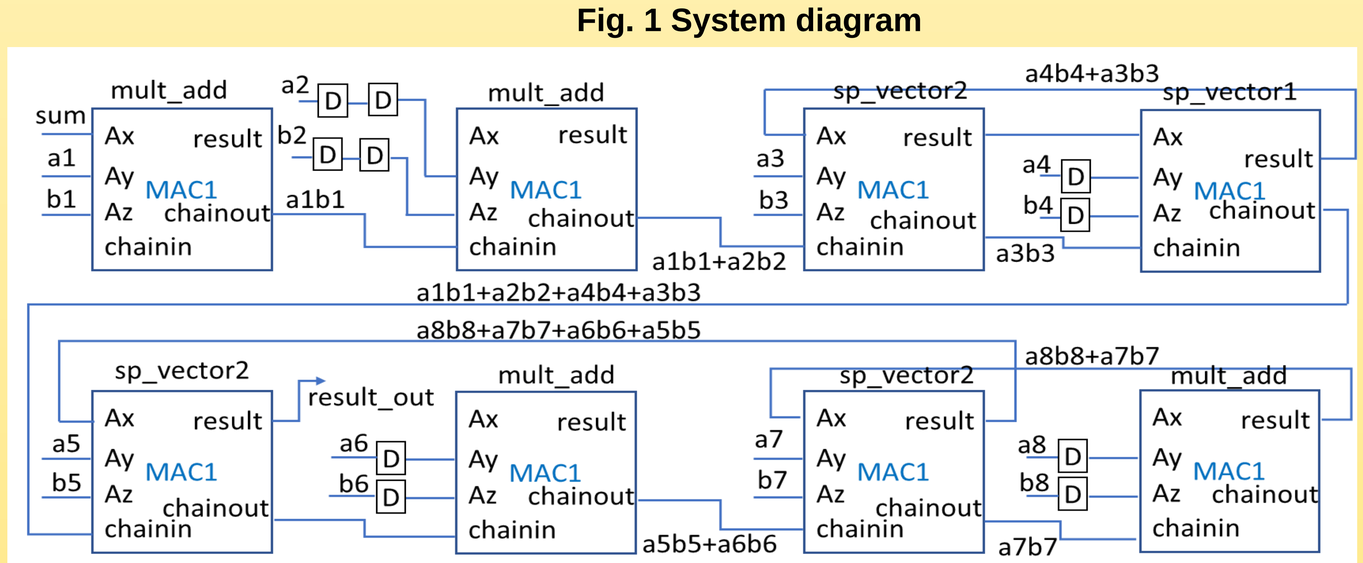


Fig. 2: PE diagram

Table 1 Hardware resource utilization

FPGA	Hardware resource			
	Logic utilization	DSP blocks	RAM blocks	Clock frequency
the proposed	237128(56%)	1280(84%)	2529(93%)	305 MHz
Intel OpenCL example	92002(22%)	520 (34%)	1774(65%)	235 MHz

[1] Andrew Ling, "Creating High Performance Applications with Intel's FPGA SDK for OpenCL", http://www.iwocl.org/wp-content/uploads/iwocl2017-andrew-ling-fpga-sdk.pdf

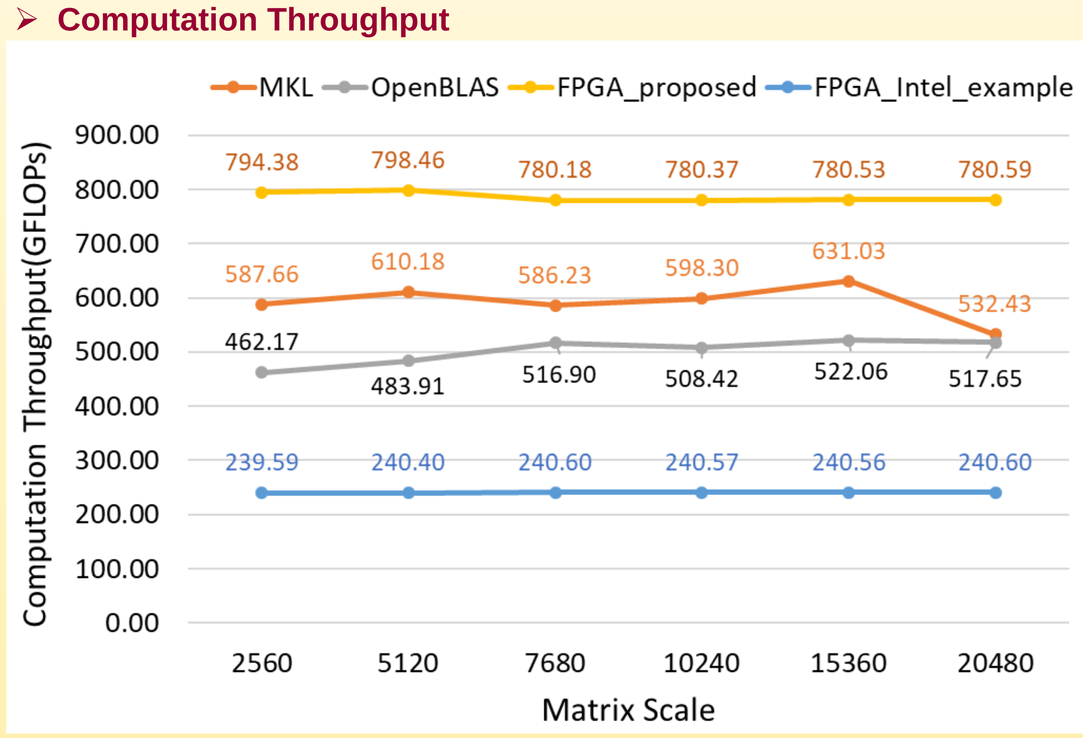
(iii) Performance Evaluation

- The proposed matrix multiplier is implemented using the FPGA board DE5a-NET. Its performance is evaluated and compared with the Intel's OpenCL example with data parallelism on FPGA, software simulations based on the Intel MKL and OpenBLAS libraries.
- The proposed matrix multiplier averagely achieves about **785 GFLOPs** in computation throughput and **81 GFLOPs/W** in energy efficiency in the case of data being single-precision, which are about **3.2 times, 1.3 times, 1.6 times, and 3.4 times, 12.7 times, 14.6 times in average** over the Intel's OpenCL example with data parallelism on FPGA, software simulations based on the Intel MKL and OpenBLAS, respectively, **even if the fabrication technology of FPGA is significantly lagged that of CPU**.

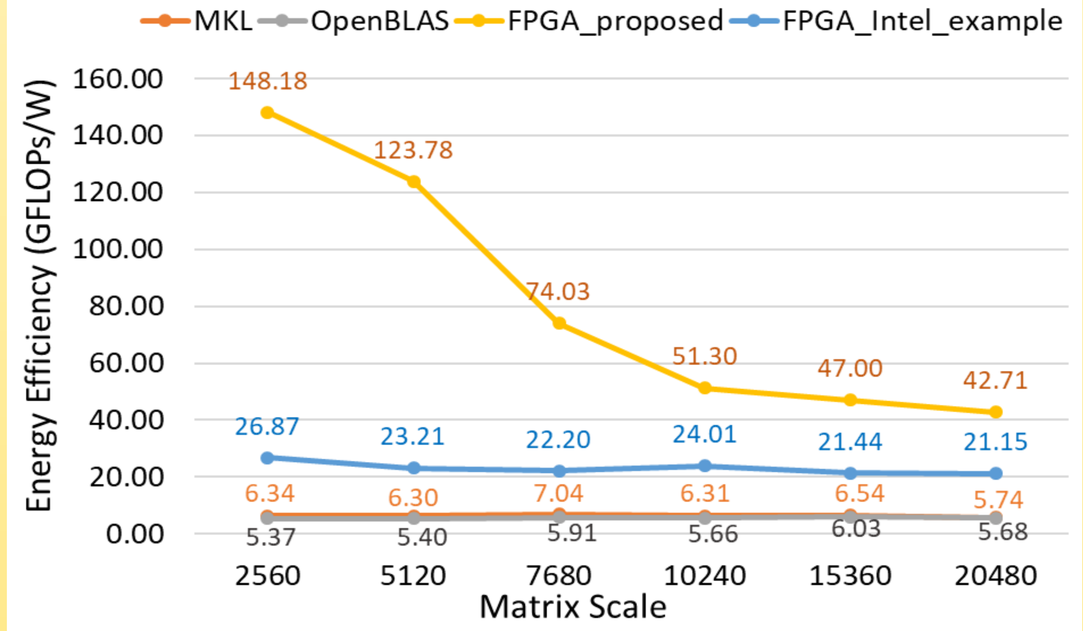
Technology Specification of Evaluation Environment

	FPGA	CPU
Model	Arria 10 GX (10AX115N2F45E1SG)	Intel i7-6800K
Cores	1518 DSP blocks	6 cores
Clock frequency	About 300 MHz	3.4 GHz
On-chip memory	6.25 MB block RAMs	L1 cache: 192 KB L2 cache: 1.5 MB L3 cache: 15 MB
External RAMs	8 GB	32 GB
Fabrication	**20 nm**	**14 nm**
Library		MKL 2018; OpenBLAS_0.2.20
Compiler	Intel SDK for FPGA 16.1	gcc 4.9.4; icc 18.0.0
Program. lang.	OpenCL	C

Computation Throughput



Energy Efficiency



(iv) Conclusions

A FPGA-based matrix multiplier with task parallelism is designed and implemented using the FPGA board DE5a-NET in this research. Compare with the matrix multiplier with data parallelism on FPGA and software simulations based on the highly optimized Intel MKL and OpenBLAS libraries, it achieves much better performance. In future work, the proposed system will be applied to solve other numerical problems.

Acknowledgements

Thanks for Intel's donation of the FPGA board DE5a-NET through University Program, and this work was partially supported by the Grant-in-Aid from the Foundation for Computational Science (FOCUS).