Yongming Shen

(631) 949-5996
yoshen@waymo.com
https://compas.cs.stonybrook.edu/~yoshen/

Education

Stony Brook University	**Stony Brook, New York, USA**
Ph.D. in Computer Science	May 2021
Thesis: The Argus FPGA-based CNN Accelerator Generator	
South China University of Technology	**Guangzhou, Guangdong, China**
M.E. in Computer Systems Organization	June 2011
B.E. in Computer Science and Technology	July 2008

Work Experience

Waymo	**New York, New York, USA**
ML Architect	July 2021 – Present
Stony Brook University	**Stony Brook, New York, USA**
Research / Teaching Assistant	August 2013 – May 2021
NVIDIA	**Santa Clara, California, USA**
Deep Learning Architect Intern	May 2017 – August 2017
Cavium	**San Jose, California, USA**
Software Engineering Intern	June 2015 – August 2015

Honors and Awards

Catacosinos Fellowships for Excellence in Computer Science (2021)
RoboCup China Open 2007, Simulation 3D League, Second Prize

Publications

2021 **The Argus FPGA-Based CNN Accelerator Generator**
Yongming Shen. Ph.D. Dissertation. Published by ProQuest Dissertations Publishing.

2019 **Argus: An End-to-End Framework for Accelerating CNNs on FPGAs**
Yongming Shen, Tianchu Ji, Michael Ferdman, and Peter Milder.
In IEEE Micro (Volume: 39, Issue: 5).

2018 **Medusa: A Scalable Memory Interconnect for Many-port DNN Accelerators and Wide DRAM Controller Interfaces**
Yongming Shen, Tianchu Ji, Michael Ferdman, and Peter Milder.
In the 28th international conference on field programmable logic and applications (FPL).

2017 **Maximizing CNN Accelerator Efficiency Through Resource Partitioning**
Yongming Shen, Michael Ferdman, and Peter Milder.
In the 44th international symposium on computer architecture (ISCA).

2017 **Escher: a CNN Accelerator with Flexible Buffering to Minimize Off-chip Transfer**
Yongming Shen, Michael Ferdman, and Peter Milder.
In the 25th IEEE international symposium on field-programmable custom computing machines (FCCM).

2016	**Overcoming Resource Underutilization in Spatial CNN Accelerators** Yongming Shen, Michael Ferdman, and Peter Milder. In the 26th international conference on field programmable logic and applications (FPL).
2016	**Demystifying Cloud Benchmarking** Tapti Palit, Yongming Shen, and Michael Ferdman. In the 2016 IEEE international symposium on performance analysis of systems and software (ISPASS).
2015	**Architectural Support for Dynamic Linking** Varun Agrawal, Abhiroop Dabral, Tapti Palit, Yongming Shen, and Michael Ferdman. In the 20th international conference on architectural support for programming languages and operating systems (ASPLOS).
2014	**Temporal Stream Branch Predictor** Yongming Shen and Michael Ferdman. In JWAC-4: championship branch prediction workshop (in conjunction with ISCA'14).

Projects

2015 – 2021 **Argus – end-to-end CNN acceleration on FPGAs (Ph.D. Thesis)**

Led the development of Argus, an end-to-end framework for accelerating the inference of convolutional neural networks (CNNs) on FPGAs. Argus takes a CNN exported from a machine learning framework (PyTorch, TensorFlow, etc.) plus the parameters of an FPGA board as input, and as output, automatically generates the RTL of a highly optimized accelerator for running the target CNN on the target FPGA board. As project lead, I was responsible for most of the design and implementation of Argus. The innovative ideas that I put into the design of Argus have led to first author publications in top tier computer architecture and FPGA conferences (ISCA, FCCM, and FPL). In particular, the heterogenous multi-processor approach used by Argus's accelerator design optimization algorithm significantly reduced the dynamic underutilization of arithmetic units and resulted in up to 3.8 times throughput improvement over competing designs. The Escher processor template used by Argus has a flexible data-buffering design which can balance feature map and weight data transfer on a layer-by-layer basis, reducing the peak off-chip bandwidth demand of an accelerator by up to 2.4 times. Additionally, Argus features Medusa, a novel memory interconnect which uses a barrel shifter to efficiently route off-chip data to on-chip buffers. Medusa achieved 4.7 times hardware cost reduction while improving performance by 1.8 times when compared to competing designs. In addition to generating accelerator RTLs, as an end-to-end solution, Argus also includes a Linux driver and a ZeroMQ-based micro-service for serving inference requests. Overall, the design and implementation of Argus requires in depth knowledge of CNNs, FPGAs, and deep learning accelerator architectures, as well as fluency in several programming languages and hardware description languages (C, C++, Scala, Python, Verilog, and Bluespec).

2014 **Musk – a CPU that implements a subset of the x86-64 ISA**

From scratch, built a 5+ stage pipelined, super-scalar CPU with set-associative caches, with synthesizable System Verilog.

2014 **JOS Hypervisor**

Extended JOS (an Exokernel operating system) to become a virtual machine hypervisor.

2014 **CBP 2014**

Built a temporal-streaming branch predictor for the Championship Branch Prediction 2014 competition.

2013 **MzOS – a UNIX-like Operating System**
From scratch, developed an operating system that supports preemptive scheduling, demand paging and DMA-based IO. Also, a file system and a set of user utilities were implemented.

2007 **RoboCup China Open 2007**
Developed a control program that enabled virtual humanoid robots to automatically balance and walk. Our team won the second prize in the Simulation 3D league.

Teaching Experience

Stony Brook University	**Stony Brook, New York, USA**
AMS 595 Fundamentals of Computing	Fall 2017
(TA) CSE 219 Computer Science III	Spring 2014
(TA) CSE 310 Computer Networks	Fall 2013
South China University of Technology	**Guangzhou, Guangdong, China**
(TA) Discrete Mathematics	Fall 2008

Professional Service

External Reviewer: The 54th IEEE/ACM International Symposium on Microarchitecture (2021); IEEE Transactions on Computers (2020); IEEE TVLSI (2020, 2019); IEEE JETCAS (2019, 2018); Transactions on Computer-Aided Design of Integrated Circuits and Systems (2019, 2018); IEEE Signal Processing Letters (2018); Journal of Systems Architecture (2018); Integration, the VLSI Journal (2018, 2017).

Technical Skills

Software Development: C, C++, Java, Scala, Python, Javascript, Assembly.
Hardware Development: FPGA, Verilog, Bluespec, Chisel.