Neural Networks as Function Primitives

Software/Hardware Support with X-FILES/DANA

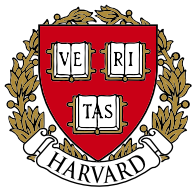
Schuyler Eldridge[1] Tommy Unger[2] Marcia Sahaya Louis[1]
Amos Waterland[3] Margo Seltzer[3]
Jonathan Appavoo[2] Ajay Joshi[1]

[1]Boston University Department of Electrical and Computer Engineering

[2]Boston University Department of Computer Science

[3]Harvard University School of Engineering and Applied Sciences

Boston Area Architecture Workshop '16



Neural Networks as Function Primitives

Motivation

Neural Networks as Function Primitives

Motivation

- Neural networks and machine learning are everywhere (*again*)
  - Broad use in high tech and big data, e.g., Google's Tensorflow [1]
  - Enable automatic parallelization, e.g., ASC [3]
  - Provide a means for approximate computing, e.g., NPU [2]

[1] Google TensorFlow, `https://github.com/tensorflow/tensorflow`

[2] H. Esmaeilzadeh, A. Sampson *et al.*, "Neural acceleration for general-purpose approximate programs," in *Proc. MICRO*, 2012.

[3] A. Waterland, E. Angelino, R. P. Adams, J. Appavoo, and M. Seltzer, "Asc: Automatically scalable computation," in *Proc. ASPLOS*, 2014.

Neural Networks as Function Primitives

Motivation

- Neural networks and machine learning are everywhere (*again*)
  - Broad use in high tech and big data, e.g., Google's Tensorflow [1]
  - Enable automatic parallelization, e.g., ASC [3]
  - Provide a means for approximate computing, e.g., NPU [2]
- Our vision
  - Neural networks are a new *functional primitive* useful at various scales of computation [4]

[1] Google TensorFlow, `https://github.com/tensorflow/tensorflow`

[2] H. Esmaeilzadeh, A. Sampson *et al.*, "Neural acceleration for general-purpose approximate programs," in *Proc. MICRO*, 2012.

[3] A. Waterland, E. Angelino, R. P. Adams, J. Appavoo, and M. Seltzer, "Asc: Automatically scalable computation," in *Proc. ASPLOS*, 2014.

[4] S. Eldridge, A. Waterland *et al.*, "Towards general-purpose neural network computing," in *Proc. PACT*, 2015.

Neural Networks as Function Primitives

Motivation

- Neural networks and machine learning are everywhere (*again*)
  - Broad use in high tech and big data, e.g., Google's Tensorflow [1]
  - Enable automatic parallelization, e.g., ASC [3]
  - Provide a means for approximate computing, e.g., NPU [2]
- Our vision
  - Neural networks are a new *functional primitive* useful at various scales of computation [4]
- With that in mind, we've developed software and hardware for the use of accelerator-backed neural network computation

[1] Google TensorFlow, `https://github.com/tensorflow/tensorflow`

[2] H. Esmaeilzadeh, A. Sampson *et al.*, "Neural acceleration for general-purpose approximate programs," in *Proc. MICRO*, 2012.

[3] A. Waterland, E. Angelino, R. P. Adams, J. Appavoo, and M. Seltzer, "Asc: Automatically scalable computation," in *Proc. ASPLOS*, 2014.

[4] S. Eldridge, A. Waterland *et al.*, "Towards general-purpose neural network computing," in *Proc. PACT*, 2015.

Our Contributions Towards this Vision

Our Contributions Towards this Vision

X-FILES: Software/Hardware Extensions

Extensions **f**or the **I**ntegration of Machine **L**earning in **E**veryday **S**ystems

- A defined user and supervisor interface for neural networks
- This includes supervisor architectural state (hardware)

Our Contributions Towards this Vision

X-FILES: Software/Hardware Extensions

Extensions **f**or the **I**ntegration of Machine **L**earning in **E**veryday **S**ystems

- A defined user and supervisor interface for neural networks
- This includes supervisor architectural state (hardware)

DANA: An Example Multi-Transaction Accelerator

Dynamically **A**llocated **N**eural Network **A**ccelerator

- An accelerator aligning with our multi transaction vision

Our Contributions Towards this Vision

X-FILES: Software/Hardware Extensions

Extensions **f**or the **I**ntegration of Machine **L**earning in **E**veryday **S**ystems

- A defined user and supervisor interface for neural networks
- This includes supervisor architectural state (hardware)

DANA: An Example Multi-Transaction Accelerator

Dynamically **A**llocated **N**eural Network **A**ccelerator

- An accelerator aligning with our multi transaction vision

Neural Network Transactions

A transaction encapsulates a request by a process to compute the output of a specific neural network for a provided input

Or: A Drop in Accelerator for a RISC-V Microprocessor

What does that mean?

Or: A Drop in Accelerator for a RISC-V Microprocessor

What does that mean?

1. Grab a Rocket Chip RISC-V Microprocessor [1]

[1] Rocket Chip git repository, UC Berkeley, Online: `github.com/ucb-bar/rocket-chip`

Or: A Drop in Accelerator for a RISC-V Microprocessor

What does that mean?

1. Grab a Rocket Chip RISC-V Microprocessor [1]
2. Build a RISC-V toolchain

[1] Rocket Chip git repository, UC Berkeley, Online: `github.com/ucb-bar/rocket-chip`

Or: A Drop in Accelerator for a RISC-V Microprocessor

What does that mean?

1. Grab a Rocket Chip RISC-V Microprocessor [1]
2. Build a RISC-V toolchain
3. Grab a copy of our X-FILES/DANA accelerator [2]
   - Implemented in Chisel [3]

[1] Rocket Chip git repository, UC Berkeley, Online: `github.com/ucb-bar/rocket-chip`

[2] X-FILES/DANA git repository, Boston University, Online (*soon*!): `github.com/bu-icsg/xfiles-dana`

Or: A Drop in Accelerator for a RISC-V Microprocessor

What does that mean?

1. Grab a Rocket Chip RISC-V Microprocessor [1]
2. Build a RISC-V toolchain
3. Grab a copy of our X-FILES/DANA accelerator [2]
   - Implemented in Chisel [3]
4. Build an FPGA configuration for Rocket + X-FILES/DANA

[1] Rocket Chip git repository, UC Berkeley, Online: `github.com/ucb-bar/rocket-chip`

[2] X-FILES/DANA git repository, Boston University, Online (*soon*!): `github.com/bu-icsg/xfiles-dana`

[3] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis *et al.*, "Chisel: Constructing hardware in a scala embedded language," in *Proc. DAC*, 2012, pp. 1216–1225.

Or: A Drop in Accelerator for a RISC-V Microprocessor

What does that mean?

1. Grab a Rocket Chip RISC-V Microprocessor [1]
2. Build a RISC-V toolchain
3. Grab a copy of our X-FILES/DANA accelerator [2]
   - Implemented in Chisel [3]
4. Build an FPGA configuration for Rocket + X-FILES/DANA
5. User processes can safely throw *transactions* at X-FILES hardware
   - With support for feedforward and learning computation

[1] Rocket Chip git repository, UC Berkeley, Online: `github.com/ucb-bar/rocket-chip`

[2] X-FILES/DANA git repository, Boston University, Online (*soon*!): `github.com/bu-icsg/xfiles-dana`

[3] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis *et al.*, "Chisel: Constructing hardware in a scala embedded language," in *Proc. DAC*, 2012, pp. 1216–1225.

X-FILES Software Components

X-FILES Software Components

Supervisor API

- Establishes sets of processes that can access neural network hardware
- Defines the neural networks that processes are allowed to access
- *More details on the poster!*

X-FILES Software Components

Supervisor API

- Establishes sets of processes that can access neural network hardware
- Defines the neural networks that processes are allowed to access
- *More details on the poster!*

User API

- Works at the level of *transactions*
  - A complete request for access to neural network resources, communication of inputs, processing, and communication of outputs
- Initiating a new transaction
- Writing data
- Reading data

X-FILES/DANA Hardware

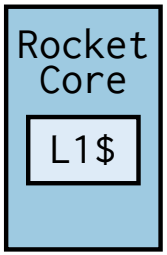


Figure: X-FILES/DANA hardware architecture

Components

X-FILES/DANA Hardware

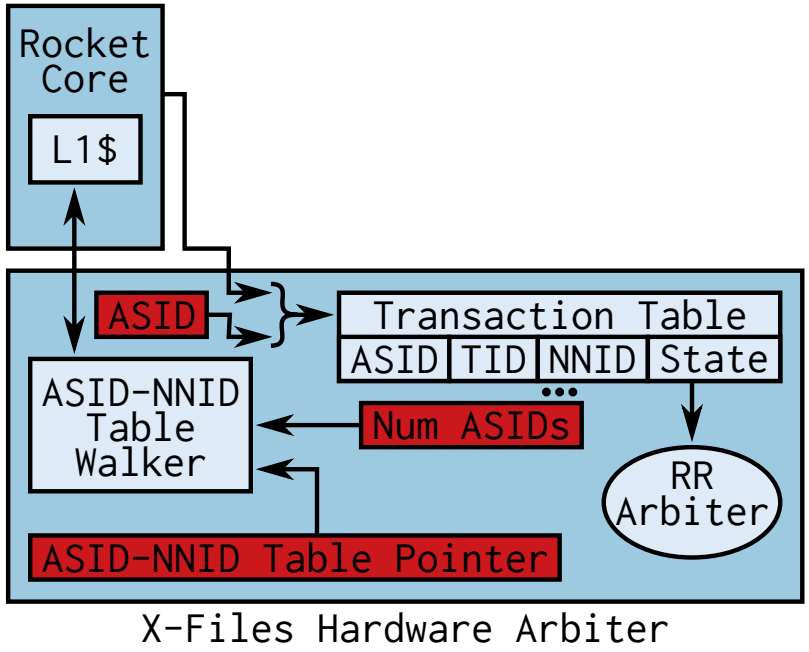


Figure: X-FILES/DANA hardware architecture

Components

- X-FILES Hardware Arbiter maintaining transaction state

X-FILES/DANA Hardware

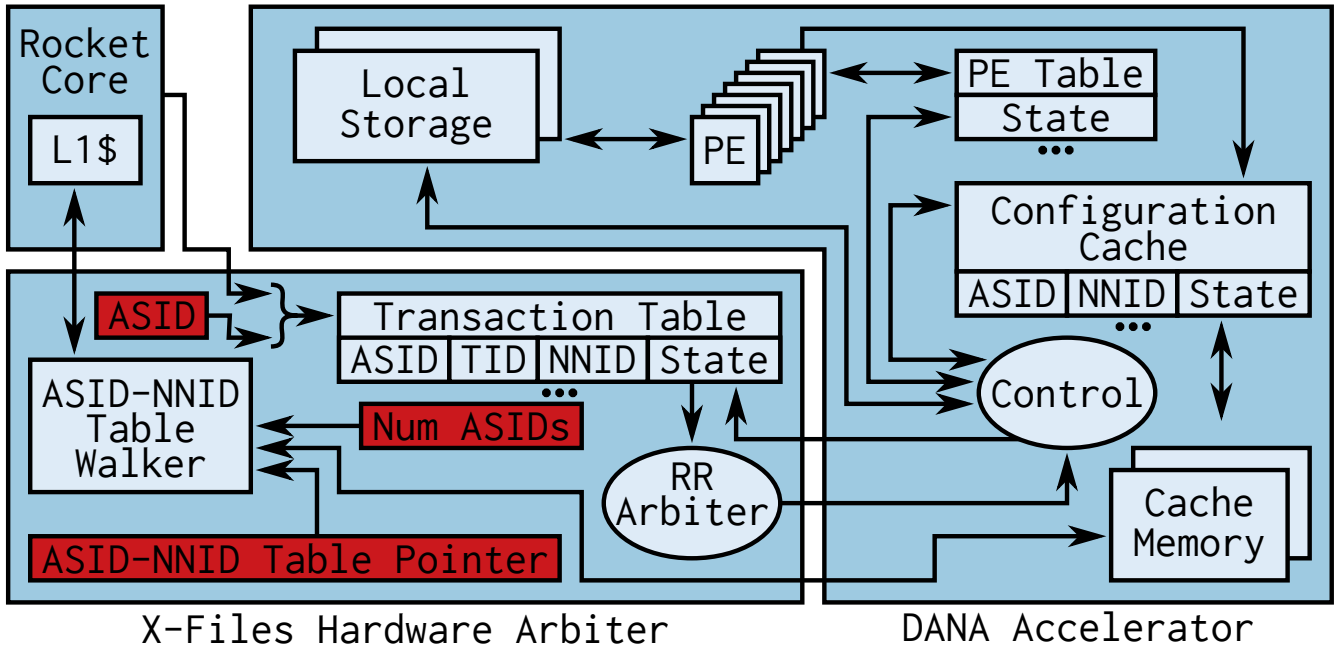


Figure: X-FILES/DANA hardware architecture

Components

- X-FILES Hardware Arbiter maintaining transaction state
- DANA to move transactions towards completion
  - With support for feedforward or learning computation

Open Source Plans

Open Source Plans

Remaining Items

- Linux kernel integration
- Support for asynchronous data transfer

Open Source Plans

Remaining Items

- Linux kernel integration
- Support for asynchronous data transfer

Open Source Availability

- Should be ready by the end of February
- On GitHub: `github.com/bu-icsg/xfiles-dana`

Acknowledgments

This work was supported by the following:

- A NASA Space Technology Research Fellowship
- An NSF Graduate Research Fellowship
- NSF CAREER awards
- A Google Faculty Research Award