Sidebar: Scratchpad Based Communication Between CPUs and Accelerators

Aditya Saini 2018125
Nishant Chaubey 2018164
Pranshu Agrawal 2018170
Rahul Singh 2018178
Yashwin Agrawal 2018205

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

The Problem

- Hardware accelerators for neural networks have shown great promise for both performance and power. They are at their most efficient when optimized for a fixed functionality.

- The problem we are trying to solve is this inflexibility, which limits the longevity of the hardware itself as the underlying neural network algorithms and structures undergo improvements and changes over the years.

Authors Solution

- The authors propose and evaluate a flexible design paradigm for accelerators with a close coordination with host processors.

- They try to achieve this by introducing an architecture enabled by a low latency shared buffer, they call Sidebar.

- In this architecture, the relatively static matrix operations are implemented in specialized accelerators while fast-evolving functions, such as activations, are computed on the host processor.

- Sidebar memory is shared between the accelerator and host, and it exists outside of program address space and holds intermediate data only.

Goals and Expectations

Goal:

Implement the Sidebar Architecture.

Expectations:

The benefit of a Sidebar based accelerator design is that it achieves near identical performance and energy to equivalent fixed function accelerators while still providing all the flexibility of computing activations on the host processor.

Possible Extensions

- Can be expanded to accelerator - accelerator communication without intervention of the host processor.
- Sidebar can enable the use of multiple accelerators performing small generic tasks, rather than a few large accelerators performing specific tasks, which can reduce overall latency.
- An interrupt based mechanism can also be tested instead of the polling based for less memory overhead
- The user needs to handle memory management for Sidebar explicitly, a compiler can be made that does that automatically.

Project Breakdown

1. Understanding the working of Neural Networks
2. Understanding the working of Activation Functions
3. Understanding the Architectures:
    a. Monolithic
    b. Flexible DMA
    c. SideBar
4. Host System Integration (How the accelerator and the CPU will interact)
5. Coherence Integrations
6. Consistency Interactions
7. Designing and Implementing the Sidebar
8. Implementing Monolithic
9. Implementing Flexible DMA
10. Accelerators
11. Testing different Architectures
12. Evaluating the Results
13. Exploring Extensions and New Avenues

Timeline (Weeks 1–4)

WEEK 1

UNDERSTANDING THE WORKING OF DIFFERENT COMPONENTS

- Understanding the working of Neural Networks

- Understanding the working of Activation Functions

- Understanding the Architectures:
    - Monolithic
    - Flexible DMA
    - SideBar
- Learning to use Gem5 Aladdin

WEEK 2

UNDERSTANDING THE WORKING OF DIFFERENT COMPONENTS

- Understanding the working of Neural Networks

- Understanding the working of Activation Functions

- Understanding the Architectures:
    - Monolithic
    - Flexible DMA
    - SideBar
- Learning to use Gem5 Aladdin

WEEK 3

UNDERSTANDING THE IMPLEMENTATION AND INTRICACIES

- Host System Integration (How the accelerator and the CPU will interact)

- Coherence Integrations

- Consistency Interactions

WEEK 4

UNDERSTANDING THE IMPLEMENTATION AND INTRICACIES

- Host System Integration (How the accelerator and the CPU will interact)

- Coherence Integrations

- Consistency Interactions

Timeline (Weeks 5–9)

WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9
DESIGNING AND IMPLEMENTING THE SIDEBAR	**DESIGNING AND IMPLEMENTING THE COMPLETE ARCHITECTURE**	**DESIGNING AND IMPLEMENTING THE COMPLETE ARCHITECTURE**	**FINE TUNING THE SYSTEM**	**FURTHER EXTENSIONS**
- Host System Integration (How the accelerator and the CPU will interact) - Implementing accelerators - Designing and Implementing the Sidebar	- Implementing the monolithic - Implementing flexible DMA - Designing and Implementing the Sidebar	- Implementing the monolithic - Implementing flexible DMA - Designing and Implementing the Sidebar	- Testing Different Configurations and Architectures - Evaluating the results	- Exploring Extensions and New Avenues

Individual Work (Rahul & Yashwin)

Rahul Singh

- *Understanding the working of Neural Networks*
- *Understanding the working of Activation Functions*
- *Understanding the Architectures:*
  - *Monolithic*
  - *Flexible DMA*
  - *SideBar*
- *Consistency Interactions*
- *Implementing Monolithic*
- *Accelerators*
- *Testing different Architectures*

Yashwin Agrawal

- *Understanding the working of Neural Networks*
- *Understanding the working of Activation Functions*
- *Understanding the Architectures:*
  - *Monolithic*
  - *Flexible DMA*
  - *SideBar*
- *Consistency Interactions*
- *Implementing Monolithic*
- *Accelerators*
- *Testing different Architectures*

Individual Work (Pranshu & Nishant)

Pranshu Agrawal

- *Understanding the working of Neural Networks*
- *Understanding the working of Activation Functions*
- *Understanding the Architectures:*
  - *Monolithic*
  - *Flexible DMA*
  - *SideBar*
- *Designing Sidebar*
- *Implementing Sidebar*
- *Accelerators*

Nishant Chaubey

- *Understanding the working of Neural Networks*
- *Understanding the working of Activation Functions*
- *Understanding the Architectures:*
  - *Monolithic*
  - *Flexible DMA*
  - *SideBar*
- *Coherence Integrations*
- *Implementing Flexible DMA*
- *Evaluating the results*

Individual Work (Aditya)

Aditya Saini

- *Understanding the working of Neural Networks*

- *Understanding the working of Activation Functions*

- *Understanding the Architectures:*
  - *Monolithic*
  - *Flexible DMA*
  - *SideBar*

- *Coherence Integration*

- *Designing Sidebar*

- *Implementing Sidebar*

- *Evaluating the Results*