Optimizing 1D and 2D Convolution Accelerators for Area, Energy, and Flexibility

Massachusetts Institute of Technology

David Paz[1], Andy Wright[2]

Advisor – Prof. Arvind[2]

[1]Department of Electrical and Computer Engineering, University of California, San Diego, San Diego, CA

[2]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA

UC San Diego JACOBS SCHOOL OF ENGINEERING Electrical and Computer Engineering

Introduction

What is a One-Dimensional Convolution?

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

What is a Two-Dimensional Convolution?

$$y[m,n] = x[m,n] * k[m,n] = \sum_{j=-\infty}^{+\infty}\sum_{i=-\infty}^{+\infty} k[i,j]x[m-i,n-j]$$

Applications of Discrete Time Convolutions:

- **Signal Processing**
- **Sound**
- **Image Processing**
- **Pattern and Edge Detection**

What is an Accelerator?

- Highly optimized hardware accelerators are capable of increasing performance on very specific computational tasks by large factors over sequential instructions. These accelerators are often optimized for high performance and instructions per clock cycle.

One-Dimensional and Two-Dimensional Applications:

- Ideal for IoT and small devices to provide significant performance gains over sequential computations and flexibility over application-specific accelerators such as CNN accelerators.

Implementation

Bluespec System Verilog (BSV) has been used to develop the accelerator by implementing pipelined Full Binary/3D Tree structures.

- Bluespec Verilog Design
- Three-stage pipelined RISC-V processor
- Complete accelerator-memory interaction
- Internal Caching (2D Accelerator)
- Implemented on Kintex-7 KC705 Board

RISC-V

bluespec

Results

Input Matrix Dimensions	Sequential Clock Cycle Count	Accelerator Clock Cycle Count	Speed-up
1x100	10,486	204	51x
1X1000	107,686	2,007	53x
1X10000	1,079,686	20,010	53x

Table 1. One Dimensional Convolution: $1x4\ Kernel$

Input Matrix Dimensions	Sequential Clock Cycle Count	Accelerator Clock Cycle Count	Speed-up
15x15	70,747	657	107x
30x30	249,802	2,248	111x
100x100	2,531,242	22,661	111x

Table 2. Two Dimensional Convolution: $3x3\ Kernel$

One-Dimensional Accelerator

$$x[n] = [X_0 \quad X_1 \quad X_2 \quad X_3 \quad \dots \quad X_n]$$

$$k[n] = [K_0 \quad K_1 \quad K_2 \quad K_3]$$

$$Y_0 = K_3 * X_0 + K_2 * X_1 + K_1 * X_2 + K_0 * X_3$$

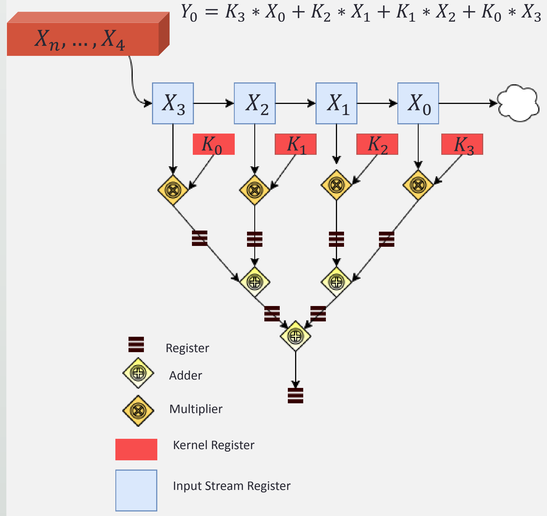
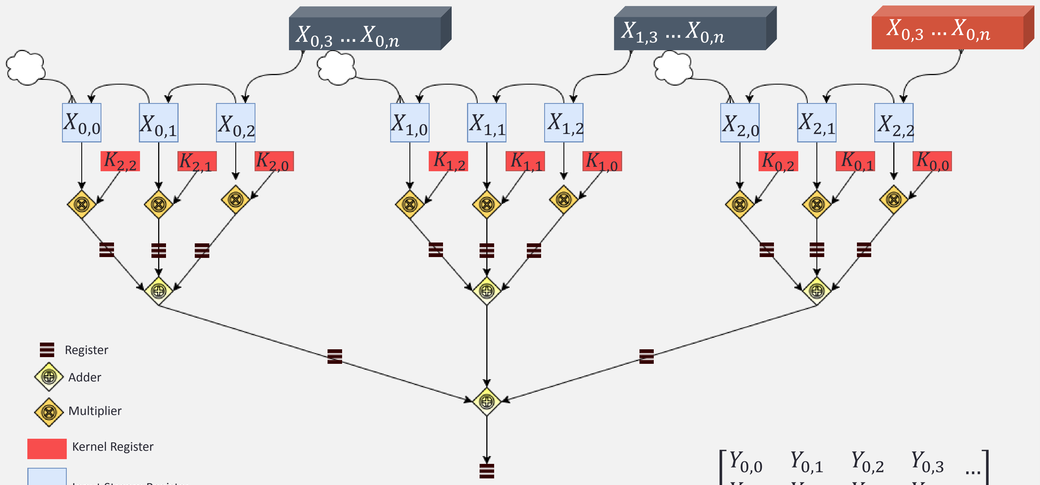


Figure 1. One Dimensional - Full Binary Tree structure

Two-Dimensional Accelerator

$$k[a,b] = \begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} \\ K_{1,0} & K_{1,1} & K_{1,2} \\ K_{2,0} & K_{2,1} & K_{2,2} \end{bmatrix} \qquad x[m,n] = \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} & X_{0,3} & \dots \\ X_{1,0} & X_{1,1} & X_{1,2} & X_{1,3} & \dots \\ X_{2,0} & X_{2,1} & X_{2,2} & X_{2,3} & \dots \end{bmatrix}$$

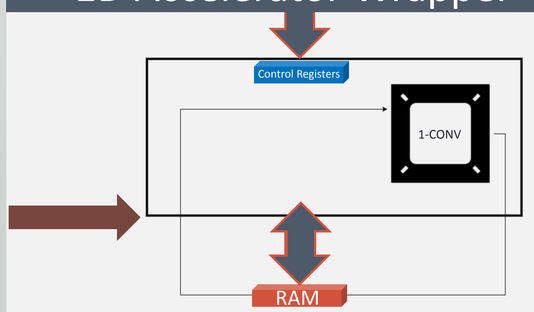
$$Y_{1,1} = K_{0,0} * X_{2,2} + K_{0,1} * X_{2,1} + K_{0,2} * X_{2,0} + K_{1,0} * X_{1,2} + K_{1,1} * X_{1,1} + K_{1,2} * X_{1,0} + K_{2,0} * X_{0,2} + K_{2,1} * X_{0,1} + K_{2,2} * X_{0,0}$$



$$Y_{2,2} \qquad y[m+1,n+1] = \begin{bmatrix} Y_{0,0} & Y_{0,1} & Y_{0,2} & Y_{0,3} & \dots \\ Y_{1,0} & Y_{1,1} & Y_{1,2} & Y_{1,3} & \dots \\ Y_{2,0} & Y_{2,1} & Y_{2,2} & Y_{2,3} & \dots \\ \vdots & & \vdots & \vdots & \end{bmatrix}$$

Figure 2. Two Dimensional – 3D Tree structure

1D Accelerator Wrapper



2D Accelerator Wrapper

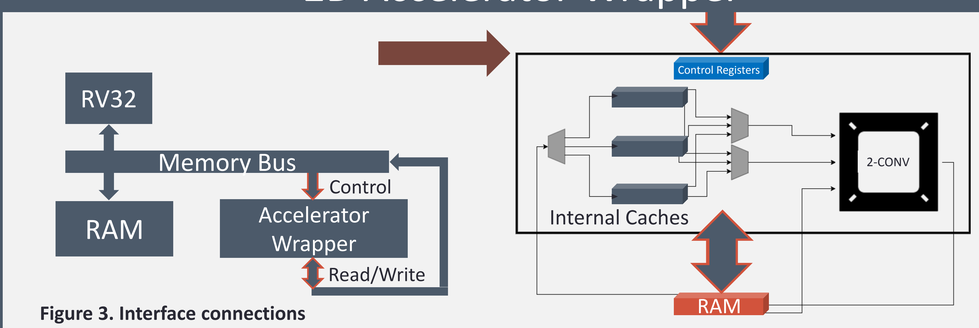


Figure 3. Interface connections

Conclusions

- Significant performance gains over sequential instruction executions by large factors.
- High-level interface offers users high customization options tailored specifically for their own applications
- Variable input and kernel matrices (more ideas for future work)

Acknowledgements

- **Mentors:**
  Arvind, PhD, Andy Wright (MIT, EECS);
  I would like to acknowledge the full support for this work provided by the MIT Summer Research Program (MSRP) and Bluespec for providing the software and tools.

MIT EECS ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MSRP MIT Summer Research Program