

Convolution Co-processor for ZYNQ7000 processing system

Joey De Smet Sam Decorte

Faculty of Engineering Technology, KU Leuven - Bruges Campus
Sporwegstraat 12, 8200 Bruges, Belgium
{joey.desmet, sam.decorte}@student.kuleuven.be

Abstract

Keywords— Co-Processor, SIMD

I. INTRODUCTION

II. IMPLEMENTATION

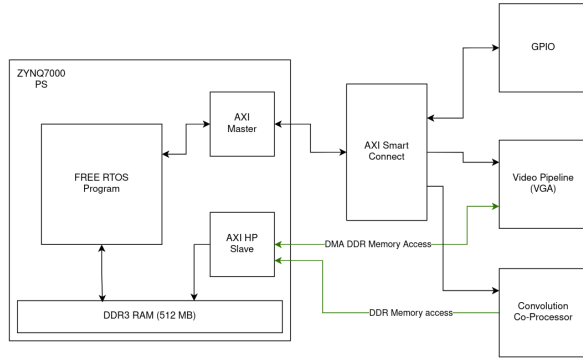


Fig. 1. Overview interconnect architecture

III. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the proposed convolution co-processor. Metrics include processing throughput, latency, resource utilization, and energy efficiency. Comparisons are made with a reference CPU-only implementation on the ZYNQ7000 processing system.

A. Experimental Setup

The experiments were performed on a Digilent ZedBoard development board with the following specifications:

- Processing System: Dual-core ARM Cortex-A9, 667 MHz
- FPGA: XC7Z020 (Artix-7), 53k LUTs, 106k FFs, 4.9 Mb BRAM
- Clock frequency of co-processor: 100 MHz
- Test images: resolution 640×480 , 32-bit RGBA
- Convolution kernel: 3×3

B. Latency and Throughput

The latency T_{latency} of the co-processor is measured as the time between issuing a convolution request and receiving the processed data:

$$T_{\text{latency}} = T_{\text{transfer}} + T_{\text{compute}} + T_{\text{response}} \quad (1)$$

Throughput $R_{\text{throughput}}$ is calculated as:

$$R_{\text{throughput}} = \frac{\text{Number of pixels processed}}{T_{\text{latency}}} \quad (2)$$

TABLE I. Latency and throughput for processing new versus in-memory images

In memory	Latency [ms]	Throughput [MPix/s]
No	—	—
Yes	—	—

C. Resource Utilization

The FPGA resource usage of the convolution co-processor is summarized in Table II:

TABLE II. FPGA Resource Utilization

Resource	Used	Available
LUTs	—	—
Flip-Flops	—	—
BRAM [Kb]	—	—
DSP Slices	—	—

D. Comparison with CPU Implementation

For reference, a CPU-only implementation, as a FreeRTOS task with highest priority, was run on the ARM Cortex-A9 core. Table III summarizes the speed-up achieved:

TABLE III. Speed-Up of FPGA Co-Processor vs CPU

CPU Latency [ms]	FPGA Speed-Up
—	—

E. Energy Efficiency

Energy consumption was measured for the convolution co-processor using onboard power monitoring or external measurement tools. The energy efficiency η is defined as the number of pixels processed per joule of energy consumed:

$$\eta = \frac{\text{Number of pixels processed}}{E_{\text{total}}} \quad [\text{MPixels/J}] \quad (3)$$

where E_{total} is the total energy consumed during the convolution operation.

TABLE IV. Energy efficiency of the co-processor for CPU and FPGA

Platform	Energy [mJ]	Efficiency [MPix/J]
FPGA	—	—
CPU	—	—

IV. CONCLUSION

V. FUTURE WORK

- Splitting the data into the different buffers to allow for more parallelism, is now managed by the processor. A hardware implementation could make it possible for data to be streamed in bigger burst which would decrease the delay for data transfer.
- Currently only 3×3 kernels are supported some minor changes could be done to expand this to a $n \times n$ kernel.

ACKNOWLEDGMENT

The authors used generative AI tools to assist with language refinement, LaTeX table template generation and grammar correction during the preparation of this paper.

REFERENCES

- [1] Digilent, *ZedBoard User's Guide*, 2014. Available: https://files.digilent.com/resources/programmable-logic/zedboard/ZedBoard_HW_UG_v2_2.pdf
- [2] ARM, *AXI specification*, 2025. Available: <https://developer.arm.com/documentation/dhi0022/latest/>
- [3] AMD, *Zynq 7000 SoC Technical Reference Manual*, 2023. Available: <https://docs.amd.com/r/en-US/ug585-zynq-7000-SoC-TRM/Register-ICCIDR-Details>