Acceleration by Separate-Process Cache for Memory-Intensive Algorithms on FPGA via High-Level Synthesis

Master's Degree Thesis

Brignone Giovanni

Supervisor: Prof. Lavagno Luciano

October 21, 2021

Politecnico di Torino

Introduction

High-Level Synthesis

Designer:

- Functionality
- Architecture
- Constraints

HLS:

- Low-level implementation

⇨

- High productivity
- Quick design space exploration
- Limited low-level control

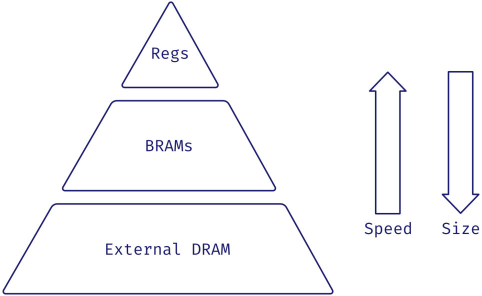
Memory management

Scratchpad:

- Manual memory selection
- HLS state of the art

Cache:

- Automatic full hierarchy exploit
- Thesis work objective



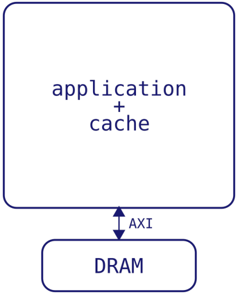
Previous work

Architecture:

- Cache inlined in application

User friendliness:

- *Integrable*: `operator[]` overload
- *Configurable*: template parameters
- *Observable*: hit ratio reports



Development

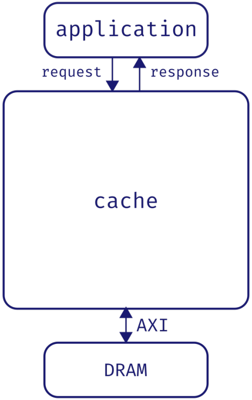
Basic architecture

Objective:

- No application cluttering

Proposed solution:

- Application:
  1. Send *request*
  2. Receive *response*
- Cache: separate process
  - SW simulation: `std::thread`
  - HW synthesis: dataflow



Basic architecture

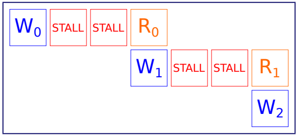
Cache process:

- Optimal pipeline (II=1)

Interface:

- Scheduler unaware of latency between *request* and *response*
- Workaround: forced clock cycles

Original schedule

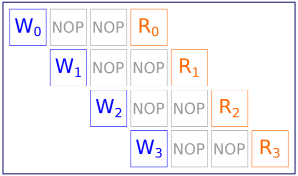


W_0 STALL STALL R_0

W_1 STALL STALL R_1

W_2

Schedule with workaround



W_0 NOP NOP R_0

W_1 NOP NOP R_1

W_2 NOP NOP R_2

W_3 NOP NOP R_3

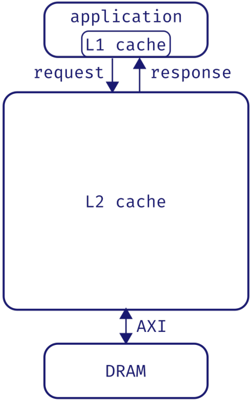
Multi-levels architecture

Objective:

- Improve performance with sub-optimal interface schedule

Proposed solution:

- L1 cache inlined in application



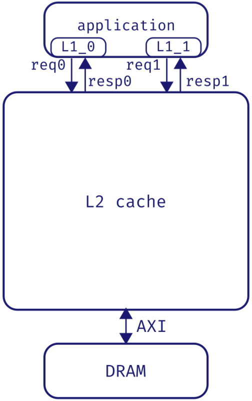
Multi-ports architecture

Objective:

- Improve performance with loop unrolling

Proposed solution:

- Multiple ports
- Single L2 cache, multiple L1 caches



Results

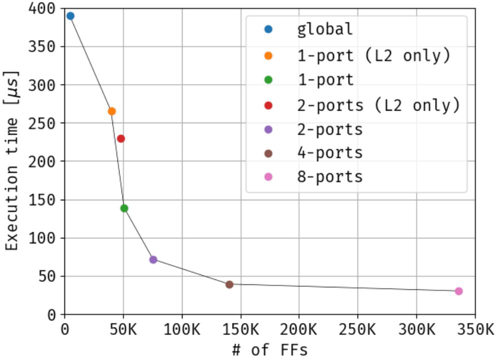
Matrix multiplication

Algorithm:

$$C = A \times B, \quad A, B, C \in \mathbb{R}^{32\times32}$$

Caches configuration:

- A: 1 line of 32 elements
- B: 32 lines of 32 elements (direct mapped)
- C: 1 line of 32 elements



Summary

Summary

Achieved results:

- Multi-process modeling for HLS
- Extended design space

Future work:

- RTL implementation for optimizing interface

References

Ma, L., Lavagno, L., Lazarescu, M., & Arif, A. (2017). Acceleration by inline cache for memory-intensive algorithms on fpga via high-level synthesis. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2017.2750923

Pursley, D. (2016). *What's the real benefit of high-level synthesis?* Retrieved October 16, 2021, from https://semiengineering.com/whats-the-real-benefit-of-high-level-synthesis

Xilinx Inc. (2021). *Vitis high-level synthesis user guide.* https://www.xilinx.com/support/documentation/sw_manuals/xilinx2021_1/ug1399-vitis-hls.pdf