ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

XXXVII Cycle

Improving Quality of Results for HLS designs

Brignone Giovanni

Supervisor: Prof. Lavagno Luciano

Research context and motivation

High-level synthesis
- C/C++ to register-transfer level
- Natural trend toward higher abstraction levels
  - Complex systems
  - Verification
  - Time-to-market
  - Design space exploration
  - ...

Open issues
- Sub-optimal Quality of Results (w.r.t. manually optimized RTL)
- Significant manual optimization effort

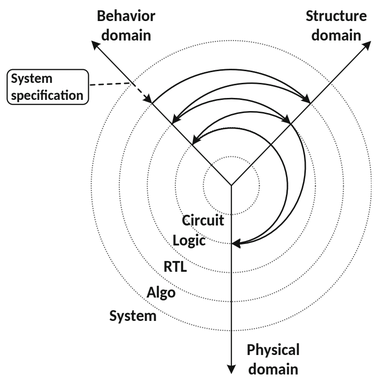

Fig. 1 – HLS Y-diagram.

Addressed research questions/problems

Memory management

Motivation
- FPGA memory hierarchy
  - Large slow off-chip DRAM
  - Small fast on-chip BRAMs/registers

State-of-the-art solution
- Manual scratchpad-like management
- Load-compute-store architecture for memory-bound algorithms
  - Efficient
  - Not automated
  - Not always applicable (irregular or data-dependent access patterns)

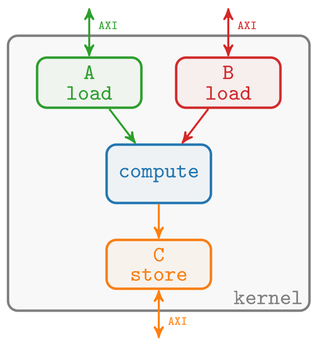

Fig. 2 – Load-compute-store architecture.

Adopted methodologies

Cache for HLS

Architecture
- Level 2: set of concurrent processes
  - Protocol for cyclic dataflow network
  - High throughput: one load/store per clock cycle when hit
- Level 1: inlined
  - Low latency
- Multi-port: single L2, multiple L1
  - Unrolling
  - Very high throughput: N loads per clock cycle when hit

Implementation
- C++ class with array-like interface and fully-configurable parameters

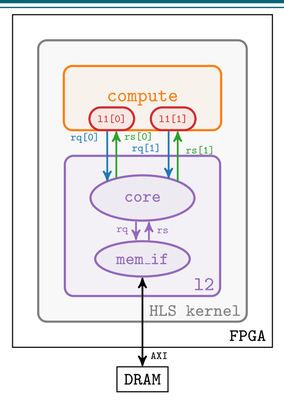

Fig. 3 – Cache internal architecture.

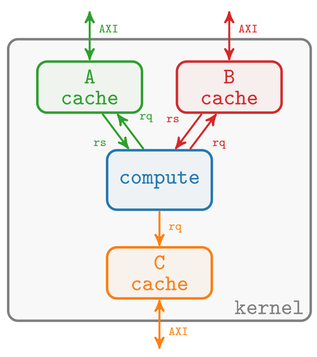

Fig. 4 – LCS-like cache architecture.

List of attended classes

- 01DNHRV – System level low power techniques for IoT (15/7/22, 20 h)
- 01DUCRV – Principles of digital image processing and technologies (22/7/22, 27 h)
- 01QTEIU – Data mining concepts and algorithms (3/2/22, 20 h)
- 01RGBRV – Optimization methods for engineering problems (7/6/22, 30 h)
- 01RISRV – Public speaking (25/11/21, 5 h)
- 01SHMRV – Entrepreneurial Finance (12/12/21, 5 h)
- 01SWPRV – Time management (24/11/21, 2 h)
- 01SWQRV – Responsible research and innovation, the impact on social challenges (12/12/21, 5 h)
- 01SYBRV – Research integrity (12/12/21, 5 h)
- 01UJBRV – Adversarial training of neural networks (6/6/22, 15 h)
- 01UNVRV – Navigating the hiring process: CV, tests, interview (2/12/21, 2 h)
- 01UNXRV – Thinking out of the box (17/11/21, 1 h)
- 01UNYRV – Personal branding (18/11/21, 1 h)
- 02LWHRV – Communication (24/11/21, 5 h)
- 02RHORV – The new Internet Society: entering the black-box of digital innovations (12/12/21, 6 h)
- 08IXTRV – Project management (11/11/21, 5 h)
- Machine Learning (Coursera), Andrew Ng (23/12/21, 20 h)

Novel contributions

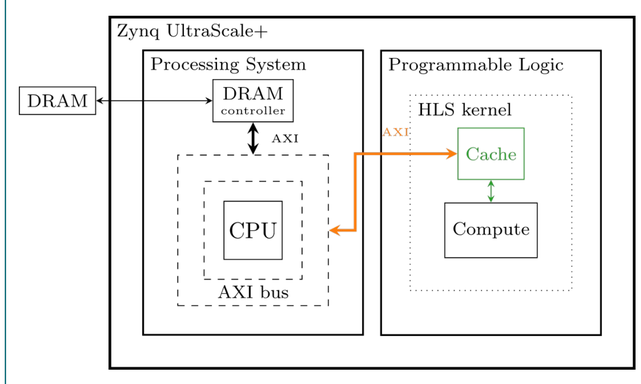

Fig. 5 – Hardware setup integrating the HLS cache.

Tab. 1 – Memory optimization approaches.

	HLS cache	LCS
Performance speedup	10^1-10^2	10^2-10^3
Energy saving	10^1-10^2	10^2-10^3
Resources	Buffers + logic	Buffers
Integration	Parameters DSE	Algorithm rewriting
Applicability	Run-time spatial and temporal locality	Design-time spatial and temporal locality

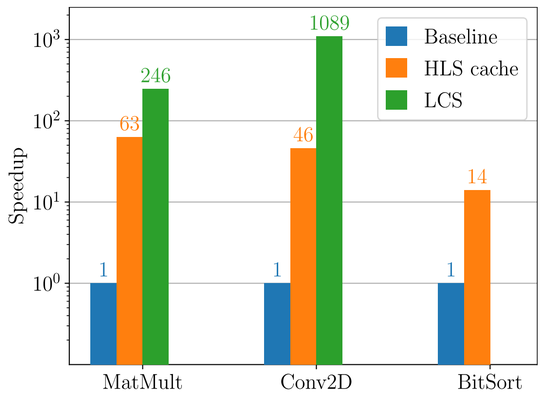

Fig. 6 – Performance of memory optimizations.

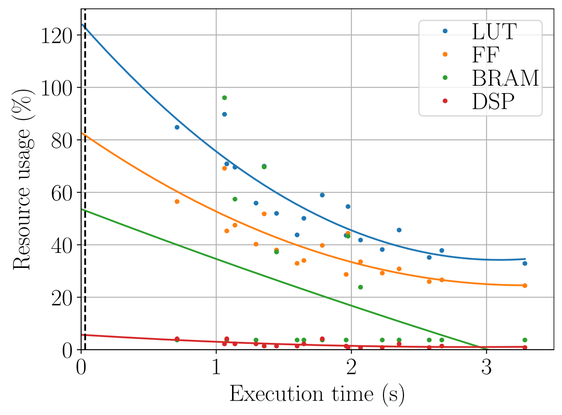

Fig. 7 – Resource usage for 2D convolution with HLS cache.

Future work

Multiple clock domains for HLS

Motivation
- Low maximum clock frequency (w.r.t. RTL)
  - Single global clock (worst-case)
  - Lack of reliable timing predictions during HLS synthesis
- Dataflow designs inherently compatible with multiple clock domains
  - Separate tasks with FIFOs at the boundaries

Implementation
- One clock per dataflow task
- Clock domain crossing through dual-clock FIFOs
- Data-driven control (blocking read/write on FIFOs)

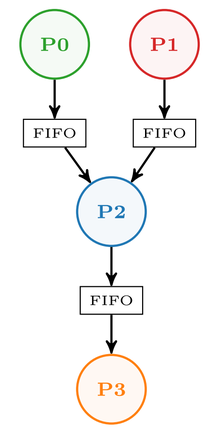

Fig. 8 – Example of a dataflow network.

Applications
- Task frequency maximizing
  - Clock frequency no longer limited by the worst case of slowest task
  - Each task running at the highest possible clock frequency
  - Example – HLS cache: AXI adapter limited at 333 MHz
    - mem_if clock @ 333 MHz
    - core and compute clock @ maximum frequency
- Throughput matching
  - Adjust frequency to match producer rate with consumer rate
  - Example – Producer produces 1 token, and consumer consumes 2 tokens per clock cycle
    - Improve performance by running producer at double frequency
    - Save area/power by running consumer at half frequency
- Multi-pumping
  - Reuse a resource N times within a system clock cycle (T_{sys}), by running it at T_{sys}/N
  - Especially effective if combined with throughput matching

Submitted and published works

- (Submitted) Brignone G., Jamal M. U., Lazarescu M. T., Lavagno L., "*Array-specific dataflow caches for high-level synthesis of memory-intensive algorithms on FPGAs*", IEEE Access

POLITECNICO DI TORINO

PhD program in
Electrical, Electronics and Communications Engineering