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

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)

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

List of attended classes

- 01DNRV – System level low power techniques for IoT (15/7/22, 20 h)
- 01DCPRV – Principles of digital image processing and technologies (22/7/22, 27 h)
- 01QTEUI – Data mining concepts and algorithms (3/2/22, 20 h)
- 01RBRV – Optimization methods for engineering problems (7/6/22, 30 h)
- 01RISRV – Public speaking (25/11/21, 5 h)
- 01SMRVM – Entrepreneurial Finance (12/12/21, 5 h)
- 01SWPRV – Time management (24/11/21, 2 h)
- 01WQRV – Responsible research and innovation, the impact on social challenges (12/12/21, 5 h)
- 01YSRV – Research integrity (12/12/21, 5 h)
- 01UBRV – 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)
- 02LHRVR – Communication (24/11/21, 5 h)
- 02RHORV – The new Internet Society: entering the black-box of digital innovations (12/12/21, 6 h)
- 06IXTRV – Project management (11/11/21, 5 h)
- Machine Learning (Coursera), Andrew Ng (23/12/21, 20 h)

Novel contributions

Tab. 1 – Memory optimization approaches.

<table>
<thead>
<tr>
<th></th>
<th>HLS cache</th>
<th>LCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance speedup</td>
<td>10⁻¹⁰</td>
<td>10⁻¹⁰</td>
</tr>
<tr>
<td>Energy saving</td>
<td>10⁻¹⁰</td>
<td>10⁻¹⁰</td>
</tr>
<tr>
<td>Resources</td>
<td>Buffers + logic</td>
<td>Buffers</td>
</tr>
<tr>
<td>Integration</td>
<td>Parameters DSE</td>
<td>Algorithm rewriting</td>
</tr>
<tr>
<td>Applicability</td>
<td>Run-time spatial and temporal locality</td>
<td>Design-time spatial and temporal locality</td>
</tr>
</tbody>
</table>

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)
- 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.wall), by running it at T.wall/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