TYRION Design Documentation

E. Jubb (ecj2122) and Ruchir Khaitan (rk2660)

A Singular Value Decomposition (SVD) hardware accelerator used to significantly speed up the execution time of Latent Semantic Analysis (LSA).

Detailed Milestones

Milestone 1: (April 2)

SVD Thread

Tasks

- Understand multiple SVD algorithms
- Review multiple existing SVD implementations
- Port one to SystemC

Final Deliverable:
Working SystemC implementation of SVD

Milestone 2: (April 14)

SVD Thread

Tasks

- Optimized SystemC implementation
- Finalize algorithm choice
- Benchmark implementation (preliminary)

Final Deliverable:
Release candidate SystemC Implementation

LSA Thread

Tasks

- Read up on LSA algorithms
- Chose one algorithm and have it running with C implementation of SVD of our choice

Final Deliverable:
Verified LSA / identified another use case

Milestone 3: (April 28)

SVD Thread

Tasks

- Commit to a final version of SVD accelerator with only minor changes to be made

Final Deliverable:
SVD running on FPGA

LSA Thread

Tasks

- Finalize LSA application

Final Deliverable:
Use case integrated with SystemC impl

Final: (May 14)

Tasks

- Complete and finalize SVD and LSA

Final Deliverable:
SVD and LSA working together
Benchmarking
(Live) Demo

Milestone Explanations

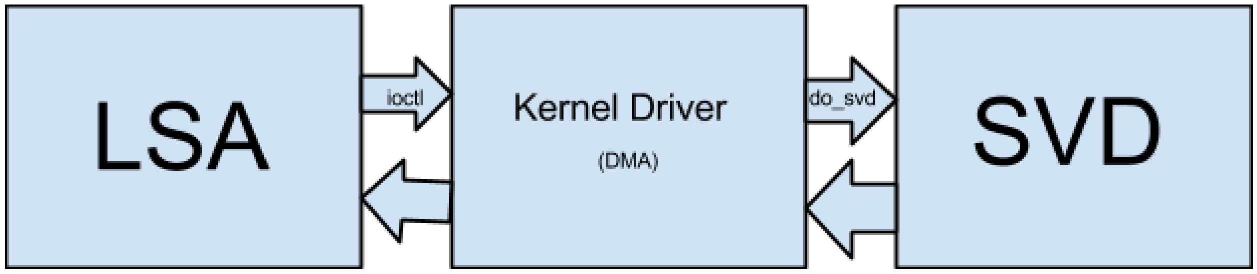
We will be developing in two parallel threads. One of these threads follows the development of the hardware accelerator for the SVD algorithm. The other follows the development of the software implementation of LSA.

In order to allow these threads to progress, we will first use a software implementation of SVD, which will allow LSA development to proceed. That software implementation will also function as a golden model for the hardware implementation.

We will slowly integrate the hardware with the software in a modular way. This will allow us to debug in a sensible manner.

Development of the SVD accelerator will be conducted using SystemC. This will allow for faster development and optimization through the development process. This SystemC will be compiled to Verilog using the Cadence CtoS compiler. At this point, we will synthesize the Verilog in the usual way onto the SoCKit's FPGA.

Block Diagram


LSA
ioctl
Kernel Driver
(DMA)
do_svd
SVD