

# TYRION

A Hardware Accelerator for SVD

Chae Jubb Ruchir Khaitan

# A Singularly Valuable Decomposition

- Do you remember linear algebra? Neither do we
- SVD allows you to decompose a matrix **A** into its singular values and left and right singular vectors

# What is it good for?

- Can make a low rank approximation  $\mathbf{A}'$  using only the first  $k$  singular values
- Has uses in machine learning, natural language processing, image compression, seismic tomography analysis, etc

# Image Compression

- Original (square) image requires  $n^2$  storage space
- Using only  $k$  singular values requires  $(2n + k)k$  space
- Relatively small  $k$  provides good approximation

# Example



$k = 64$



$k = 128$



$k = 512$

# 2-Sided Jacobi Algorithm

- Basic idea: we want a diagonal matrix, so we want all of the off-diagonal elements to be zero
- Multiply matrix  $A$  with  $2 \times 2$  rotation matrix to make off-diagonal element at index  $i, j$  go away
- Keep doing that, and collect the rotation matrices into the left & right singular vectors

# Algorithm Pros and Cons:

- Easily parallelizable since each “elimination” depends only on that row and column
- Converges in quadratic time
- An implementation existed online
- Rotation matrices require trig functions
- Trig functions mean we can't use integer data types
- Requires conversion to fixed point
- Online implementation wasn't super great

# SystemC

- System level modeling provides a higher level of abstraction (think in terms of threads and logical transactions not digital circuits)
- Generates correct and fast Verilog
- Novel toolchain



# Architecture

- Defined a high level wrapper over the hardware (between the driver and actual hardware)
- Send data to/from hardware with buffered FIFOs (one 32 bit chunk at a time)
- Communication with device done with 4-way handshake

# Interface

- You put a matrix in (either one 32 bit integer at a time or memory mapped) and you get 3 matrices out.
- Doubles are convert to 64 bit fixed point numbers (40 bit fractional part)

# Testing

- Fully randomized testbench
- SystemC provides full simulation environment
- Cargo CAD tools

**DEMO!**



# Challenges

- Setting up toolchain
- Dealing with communication between different Avalon protocols
- Bus woes

# Lessons Learned

- **Ruchir:** Hardware is hard whereas software is fun. Also, having good partners makes everything much better. Also, 620 CEPSR is a great room.
- **Chae:** Mixing toolchains is hard. Mixing IPs is hard. Mixing semantics is hard. Writing code is easy.