ECE1373S

2021

Digital Design for Systems-on-Chip

Paul Chow

pc@eecg.toronto.edu

Admin

- Lectures
  - Thursday, 3-5pm
  - https://zoom.us/j/96724230710?pwd=dDYzQTRRWG00MC96cFFyVlBSVU5FUT09
    - Meeting ID: 967 2423 0710
    - Passcode: 13732021
  - Will record and post on Quercus
  - Please fill out the Lecture attendance poll – reopened
- Web
  - Will use Piazza as primary discussion forum
  - Quercus for everything else
- Course material
  - Online textbook: Parallel Programming for FPGAs
    - http://kastner.ucsd.edu/hlsbook/
  - Handouts on Quercus

TAs

- Naif Tarafdar naif.tarafdar@mail.utoronto.ca
- Camilo Vega camilo.vega@mail.utoronto.ca
- Assignments mainly
- Will answer questions posted on Piazza
  - I can't answer detailed questions. I'm just a manager ☺

Please use Piazza

- Much easier to use than Quercus as a discussion forum
- Post all questions and issues so everyone can benefit from answers
- Look for group members – Post @5

Other sources of help

- The web
  - Mysterious error message? Type it in as a search
  - Online forums
- We'll post good links we know about
- Post what you find so that we can all benefit

Requirements

- Prerequisites
  - Digital design experience with an HDL is required
    - Strong undergrad background should be enough
    - Digital design and computer architecture
    - http://hdlbits.01xz.net – should be comfortable doing at least 75% of them
  - Will not be teaching hardware design with HDL
    - You should be comfortable with hardware design and techniques for achieving performance

Goals

- Hardware design is learned by experience
  - Build something real, large and significant
  - Do a Systems Design
  - Learn how long it really takes – aiming for late-April – longest was 1 year ☺ when we built chips
- Learn and experience the latest tools
  - FPGA system tools getting quite sophisticated
- Work in a team for projects – minimum group of three preferred

Design Approach

- Use high-level synthesis (HLS) to design some significant function
- Why need for HDL experience?
  - Using HLS still requires strong understanding of HDL design, hence requirement for knowing HDL design
  - HLS builds cores, so still need HDL skills to do integration into a system

General Expectations

- Large time commitment—learning is in the experience
  - You will be learning the tools on your own in a very short time – some lectures about HLS, but not the buttons, assignments
  - Tools can be a major component of your time
    - learning and dealing with them not doing what you want
  - Consider other course loads
- Projects done in groups—learn to manage group dynamics

Lecture Topics

- HLS intro tutorials from Xilinx training
  - Quick overview in lectures
  - Will make available to you
- Design methodologies
  - Managing large designs
  - Thinking about large designs
- FPGAs, technology spectrum
- Accelerator architectures
- Timing, clock distribution, clock domains
- High-speed interfaces

Assignments

- Two assignments + project
  - Assignments help ramp you up on HLS
  - HLS enables significant projects even in shorter time
  - Also need to learn basic RTL flow, so you are learning many tools in a short time
  - Assignment 1 – HLS features – Use Vivado HLS
  - Assignment 2 – Build a single convolution layer and run on a real FPGA in my lab -- Need Vivado

Grading

- 35% 2 Assignments
  - Help with getting up to speed with the tools
  - Vivado HLS – the HLS tool
  - Vivado – IP integrator, implementation
  - Gain familiarity with hardware environment
- 15% – presentations and class participation
- 50%
  - project demo, report
  - design process, practices, testing procedure used in project

Timeline

Date	Task
January 21, 2021	Assignment 1 hand out
February 11, 2021	Proposal Presentations start
February 11, 2021	Assignment 1 due Assignment 2 hand out
March 14, 2021	Assignment 2 due Projects become focus
April-May	Project completed

Course Completion

- Final grade submission
  - Anyone want to graduate in June?
  - Usually let projects continue into May or so
    - Significant projects, students want to finish, some publish
  - If you need to complete in April then let me know

Tools

- Xilinx Vivado 2018.3 for assignments
  - Download at xilinx.com – Windows and Linux 17GB in one shot! Use web installer instead
- Will provide license server—access by ssh tunnel
- Start working on home installation now
  - Instructions on Piazza @19
  - Post issues and success to Piazza
- How many need another option?

PROJECT

Project

- Groups – 3 or more preferred
- Build a fully functional design running on an FPGA board
- Must include at least one processor
  - ARM, MicroBlaze, or other
- Must have significant HLS component you design

People Challenges

- Make sure that the groups you form can meet together regularly
- Remote design is the norm today!

Possible Projects

- Your own ideas
  - From research or your interests
  - Accelerate something
- Will suggest some ideas next few weeks
  - Theme – FPGAs in the cloud

Platforms

Bear in mind the need to work remotely

- Your own FPGA board
- Avnet Zynq mini-ITX with 7Z100
  - Zedboard.org
- AlphaData cards in our cluster used for assignments
  - By remote access

HIGH-LEVEL SYNTHESIS (AT A HIGH LEVEL)

Design is Changing

- Changed the design approach five years ago
- Course Evolution: custom VLSI, ASIC design and layout, “Front-end” HDL design using FPGAs
- Time for High-level synthesis
  - Will use Xilinx Vivado tools with HLS
  - Algorithms and dataflow rather than pure hardware
  - My grad student’s are using it successfully
  - Using it successfully in my company, other industry
  - Time to move to this new technology to address TTM and complexity

First, RTL

- Register-transfer level
- Description used in HDLs – Verilog, VHDL
- Explicitly create registers to store data, control to move data, operations to process the data
- Clocks and timing is explicitly visible
- Higher-level of abstraction than gates, but lower than algorithms (software)

RTL Describes Circuits

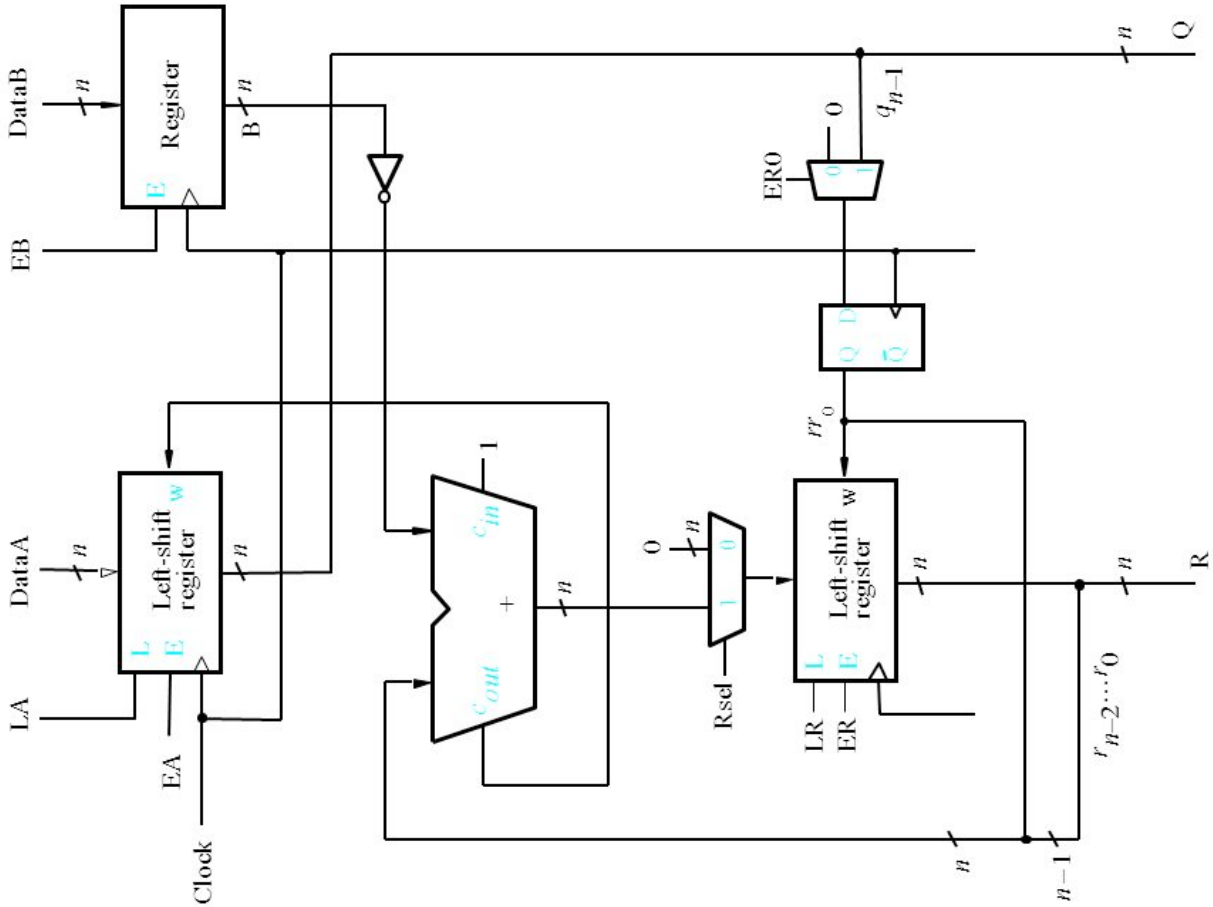


Figure 10.27 Datapath circuit for the enhanced divider

HLS – Briefly

- High-level synthesis
- Higher-level of abstraction
  - Describe algorithms and dataflow
  - Closer to software
  - Write in C, C++, OpenCL
  - Tool creates the hardware
  - Prof. Zhu's course is about the algorithms to do this
- Holy grail of hardware design
  - "software programmers" can build hardware…

HLS is an Algorithmic Description

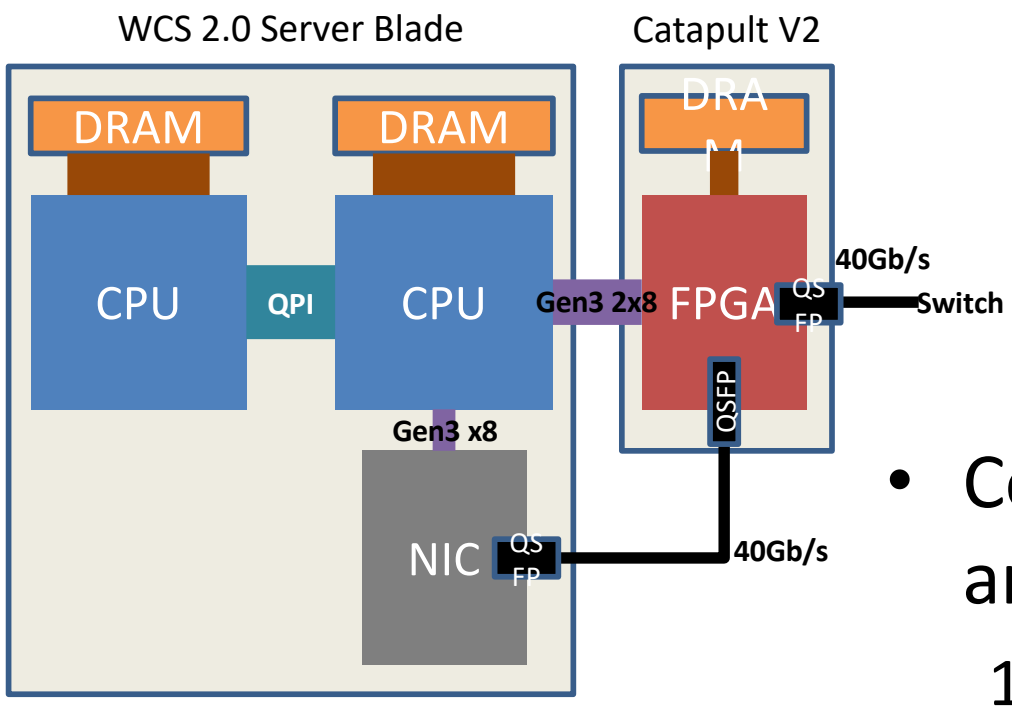
```
1   #include "myMult.h"
2
3   void myMult(const double a[12], const double b[20], double c[15])
4   {
5     int i0;
6     int i1;
7     int i2;
8     for (i0 = 0; i0 < 3; i0++) {
9       for (i1 = 0; i1 < 5; i1++) {
10        c[i0 + 3 * i1] = 0.0;
11        for (i2 = 0; i2 < 4; i2++) {
12          c[i0 + 3 * i1] += a[i0 + 3 * i2] * b[i2 + (i1 << 2)];
13        }
14      }
15    }
16  }
```

Current HLS Landscape

- ASIC tools – Calypto (Mentor) Catapult C
- FPGAs
  - Xilinx Vivado HLS
  - Xilinx SDx: SDAccel OpenCL, SDSoC
    - Vitis is latest umbrella environment
  - Intel SDK for OpenCL
  - Intel HLS Compiler
- HLS vs OpenCL

FPGAS IN THE CLOUD

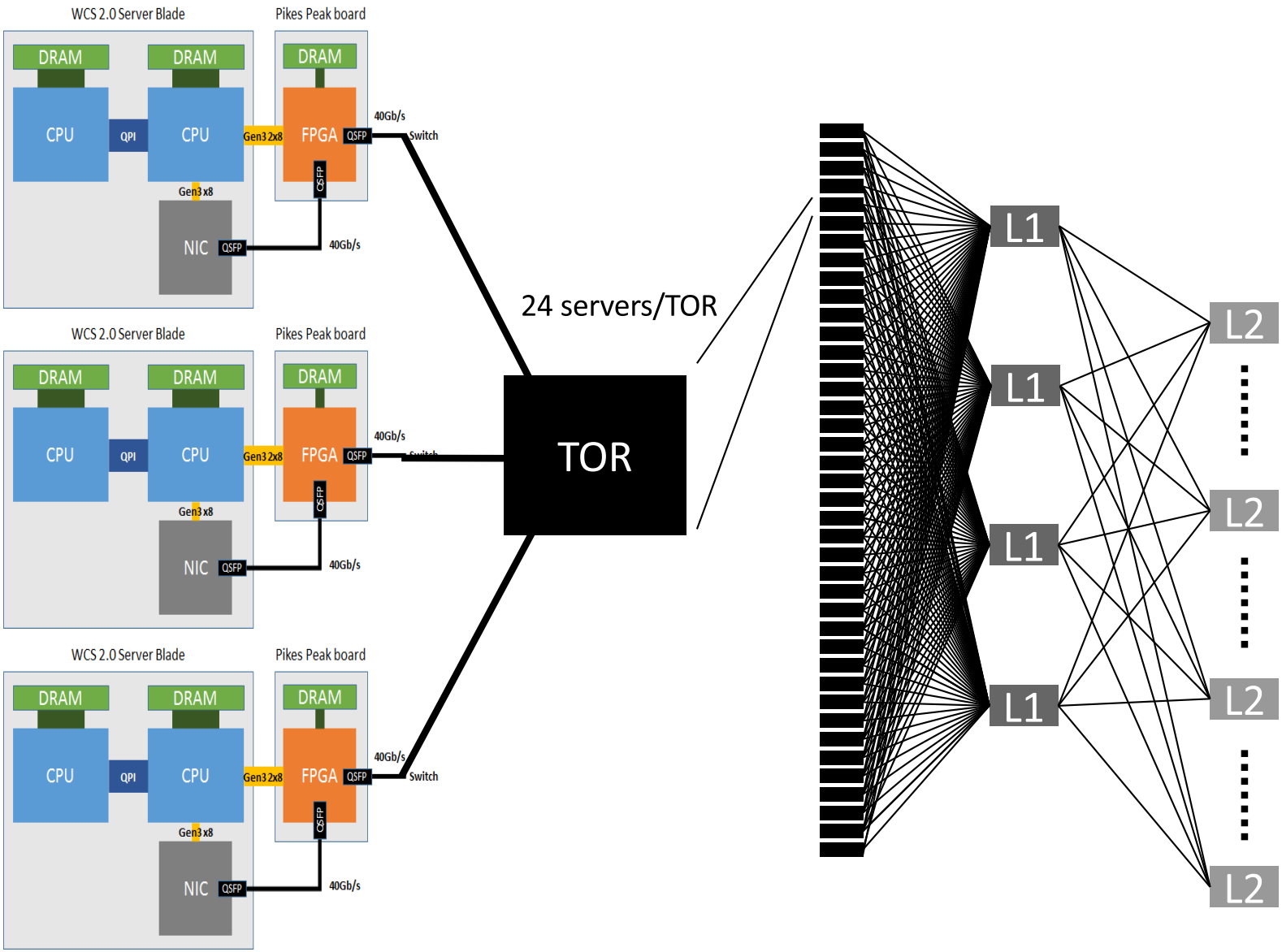
Converged Bing/Azure Architecture



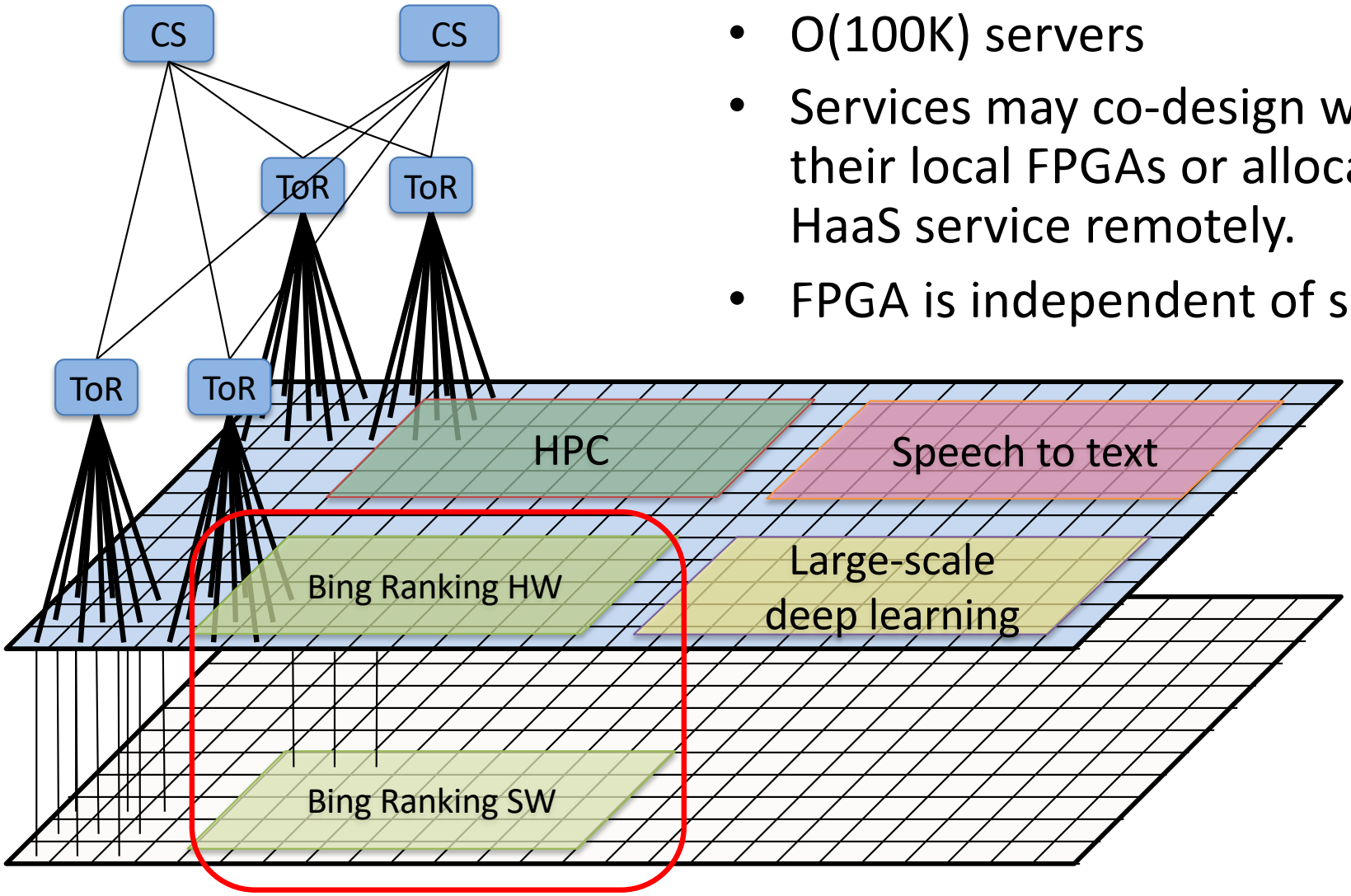
Catapult v2 Mezzanine card

- Completely flexible architecture
  1. local compute accelerator
  2. remote compute accelerator
  3. Network/storage accelerator

Network Connectivity (IP)



Hardware Acceleration as a Service Across Data Center (or even across Internet)



- O(100K) servers
- Services may co-design with their local FPGAs or allocate a HaaS service remotely.
- FPGA is independent of server

Our Cluster Platform

- Have all the components
- 100 GbE and 10 GbE ToR switches
- 8 Alpha Data 8V3
  - Will use in this course
  - 100 GbE interfaces
  - VU095 – 768 DSP blocks, 537,600 LUTs

More about our Cluster

- 8 Alpha Data 8K5
  - 10 GbE interfaces
  - KU115 – 5,520 DSP blocks, 663,360 LUTs
- 19 Fidus Sidewinder
  - 100 GbE interfaces
  - ZU19EG MPSoC – 1,968 DSP blocks, 523,000 LUTs
  - ARM complex, no host server
- Other misc.

Cluster Access

- Through containers
- Containers will have complete environment
  - Assignment 1 – just tools and tests
    - Can use if you do not want to set up your own tools
  - Assignment 2 – tools, tests, FPGA access

To Do

- Think about groups
- Think about projects

Questions?