Day 1: Intro

Zach Neveu

January 6, 2020

Organization

- Based on PYNQ framework: Python code in Jupyter, FPGA code in Python subset
- Blackboard has all kinds of fun extras, discussion boards etc.
- Using Vivado 2018.3 - version important, will error out if different
- 3 quizzes and a late midterm (quasi final)
- Second half of semester largely project based
- No final exam, slot will be used for project presentations

Project

- Application w/ hw and sw on Pynq.
- Compare HW speedup
- Try different interface types/memory allocations
- Topics can vary, ML, CV, cryptography, signal processing etc.

Topics

- What is an FPGA?
- Programming
- FPGA overlays
- Architecture
- Computer Arithmetic
- All of this subject to change a bit over the course

What's on an FPGA?

- Fabric - general purpose logic gates
- Micro blaze - soft-core processor
- Zynq: true hard-core processor, AXI interface to a bunch of programmable logic (PL)

- Zynq processing system (PS): ARM A9 - 10-12 years old, but still widely used
- Using the Pynq Z2 for this class

Pynq

- Pynq written in Jupyter, uses IPython kernel
- Write hardware overlays and send to fabric. Call from Python.
- Overlays based on Pynq IP
- Overlays provide a good interface for software people to use custom hardware IP

What's on an FPGA p2

- configurable logic blocks (CLBs)
- input/output block (IOB)
- programmable switch matrix (PSM)
- programmable interconnect point (PIP) - memory can control the interconnect - what connects to what?
- Everything based on memory
- Logic based on lookup tables (LUTs)
- Many multiplexors (MUXs) to decide what goes down wire
- LUT: 2^k RAM cells for k inputs, with a $2^k \rightarrow 1$ MUX to choose which output to select
- MUXs implemented as tree of $2 \rightarrow 1$ MUXs
- Memory values (truth table) chosen at compile time, MUX path chosen by inputs
- 6 input LUTs are what's on the ZYNQ
- Each CLB has 2 LUTs and 2 flip flops, muxed so you can use any combination of these