F4PGA tools in the classroom

Faculty: Mike Wirthlin, Jeff Goeders, and Brent Nelson
Students: Zach Driskill, Weston Smith, and Jared Robinson
Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah

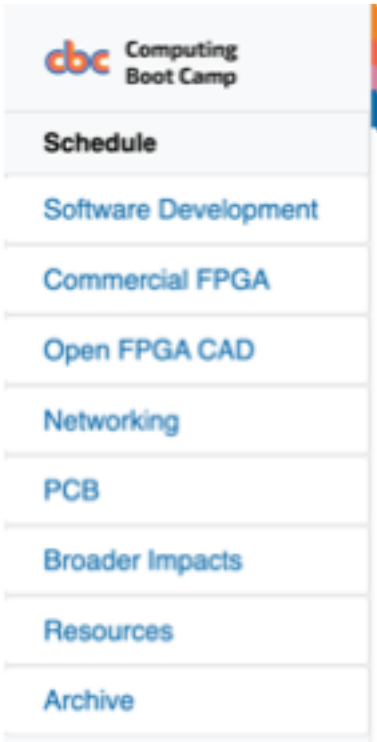
This work was supported by an educational gift from Google

BYU Configurable Computing Lab

- Investigate the use of FPGAs for computing, reliability, and security
  - Active user of multiple commercial FPGAs and supporting tools
  - Custom tools developed to support research efforts
- OpenSource Software tools
  - JHDL — Java Hardware Description Language
  - RapidSmith/2 — FPGA XDL physical design tool
  - SpyDrNet — Netlist representation and manipulation tool (used for TMR)
  - tincr — TCL CAD framework for Vivado
  - BFAT — Bitstream Fault Analysis Tool
- Actively support a variety of open source projects

BYU Computing Bootcamp

https://byu-cpe.github.io/ComputingBootCamp/



Online resources for training undergraduate students in applied topics

- Software Development Skills
- FPGA/Digital Design Skills
- OpenSource Software Development

Applied topics that develop skill beyond typical university course

Annual summer training activity for undergraduate research assistants

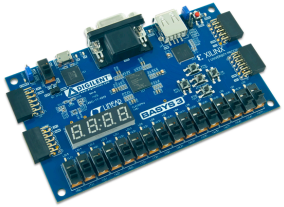
F4PGA Classroom Goals

1. Evaluate the feasibility and suitability of using OpenSource (F4PGA) hardware design tools in a university classroom environment
2. Encourage adoption of opensource tools by students
3. Identify bugs/issues with the tools based on student designs and tool use
4. Work with open source community to resolve bugs/issues found in the tools

Experiment will involve two BYU undergraduate courses: ECEN 220 and ECEN 323

BYU Undergraduate Digital Laboratory

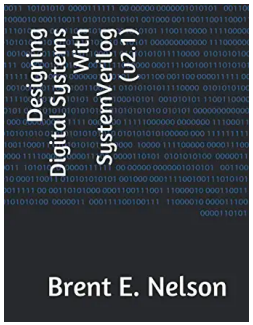
- 60 dedicated student workstations
  - Linux operating system
  - Xilinx Vivado design tools
- Instructional resources for lab teaching
- Student network drive space for storing student work (10 GB)
- Basys3 FPGA development boards
  - Purchased by students
  - Used in multiple classes



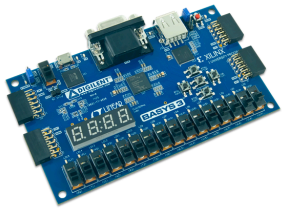
ECEN 220: Fundamentals of Digital Systems

- Binary representations & arithmetic
- Binary logic gates and logic minimization
- Timing of gates
- Sequential logic circuits
- Logic timing
- FSM design
- HDL design using SystemVerilog

- ~200 students annually
  https://ecen220wiki.groups.et.byu.net/



Text: Brent Nelson self-published book

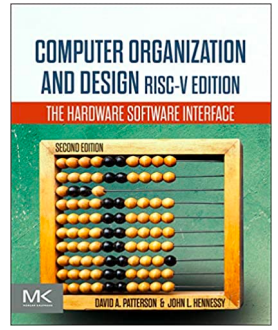


12 Laboratory exercises using student-purchased Digilent Basys3 FPGA board

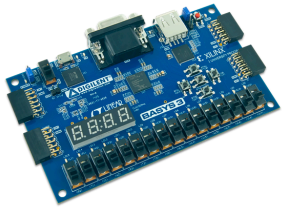
ECEN 323: Computer Organization

- Processor Instructions & Instruction Sets
- Computer Arithmetic
- Processor organization
  - Datapath, control
  - Pipelining
  - Hazards, forwarding, stalling
- Memory Hierarchy
- Parallel Processors

- ~85 students annually
  http://ecen323wiki.groups.et.byu.net



Text: Patterson/Hennessy RISC-V Book



Students build RISC-V processor (HDL), write assembly language program over course of 12 laboratory assignments (FPGA download)

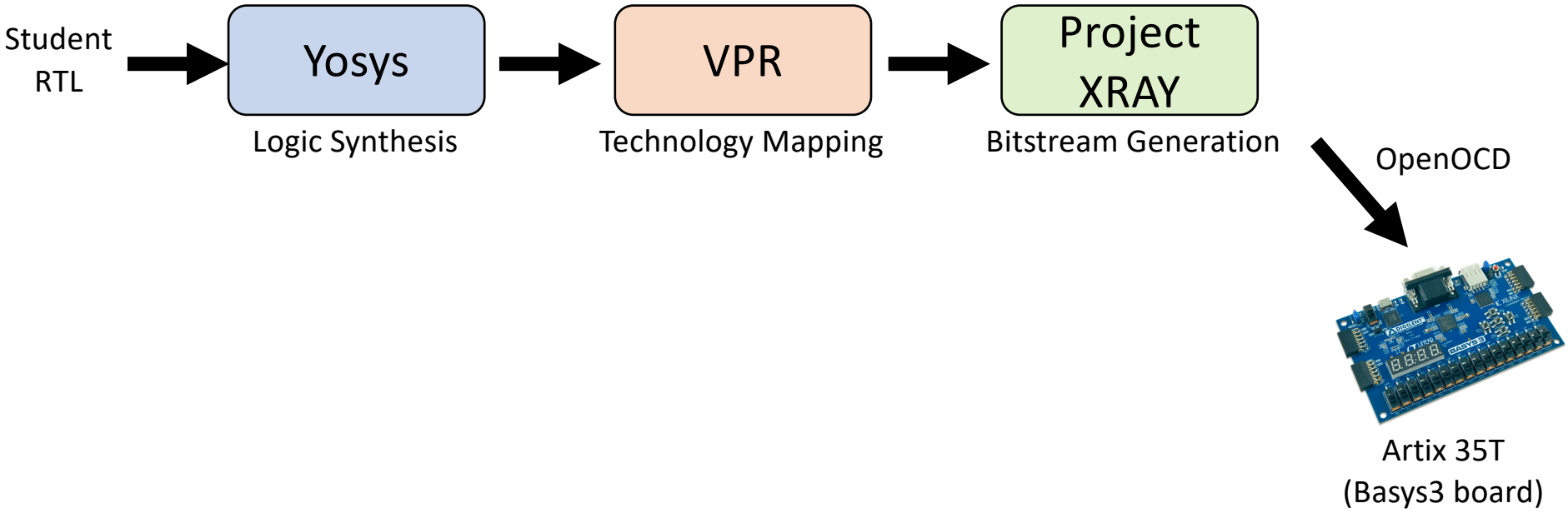
Classroom OpenSource Experiment Structure

1. Student volunteers recruited from ECEN 220 and ECEN 323 classes
   - Participation on volunteer basis
   - Financial incentive provided to students completing each lab
2. Students debug and complete labs using commercial tools
   - Students use commercial tools and extensive support network for debug
   - Must complete lab assignment successfully before proceeding to opensource part
3. Students implement working lab in opensource tools
   - HDL->Bitstream using F4PGA opensource toolflow
   - Bitstream downloaded to the board and design operation verified
   - Code submitted to GitHub classroom repository
   - Summary Readme.md file submitted to summarize any problems
4. Student TAs assist volunteers and review each laboratory submission

F4PGA Design Flow

F4PGA

https://f4pga.org/



OpenSource Classroom Support+

Student Teaching Assistants

- Support students using F4PGA
- Reviewing student submissions

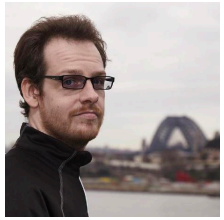
Jared Robinson
(jcrob2)

Zach Driskill
(zach227)

Weston Smith
(westonMS)

antmicro

Bi-Weekly meetings to discuss and resolve Issues identified by students



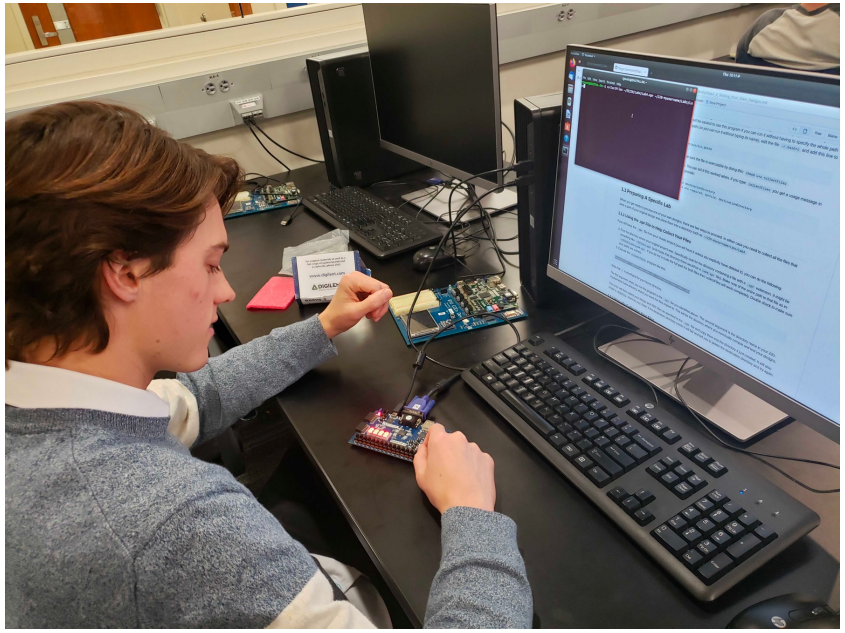
Google

Tim Ansell
(mithro)

Monthly meetings to discuss plans and progress

Winter 2022 Summary

- ECEN 220
  - 35 participating students
  - 103 laboratory submissions total
- ECEN 323
  - 28 participating students
  - 76 laboratory submissions total
- All student designs that produced a bitstream operated correctly on FPGA



Summer 2022 Summary

- TAs processed student issues
  - Replicate student issues
  - Combine related issues
  - Submit issues to F4PGA repo
- Resolve Issues
  - Identify source of issue
  - Work with community to fix issue
- New F4PGA installation process created for Fall

Fall 2022 Summary

- ECEN 220
  - 17 participating students
  - 70 laboratory submissions
- ECEN 323 (not taught)

- Several new issues identified
  - TAs currently reviewing issues

Winter 2022 Issues

- 22 unique issues identified during semester
  - Most of the issues were identified by multiple students
  - Issues reproduced by student TAs
  - Details of the issue were identified and posted to the appropriate GitHub repository
  - All “bugs” have been resolved, additional features not yet addressed
  - Many issues related to Yosys front end Verilog parser (future semester used Surelog)
- Issue summary
  - typedef enum default size
  - Support built in gate types
  - localparam usage
  - Array initialization in declaration
  - Use of unsupported vendor primitives
  - Inferring vendor memory primitive modes
  - Memory initialization in declaration

Fall 2022 Issues

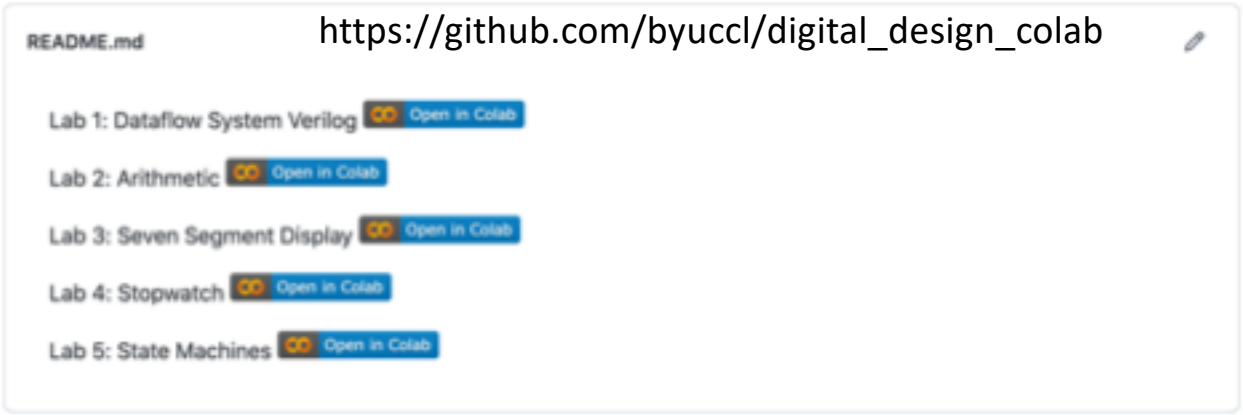
- Issues (student TAs currently evaluating/categorizing)
  - Slow execution time over network drive (large files generated)
  - Cryptic errors when environment variables not properly set
  - Built-in gates cannot take multi-bit inputs
  - Unable to handle shift with ternary operator
  - Typed parameters
  - Parameterization of top-level modules
  - $readmemh()
- Overall, use of Surelog reduced problems significantly

OpenSource Tools in the Classroom: Summary

- **Pros:**
  - Great environment for finding bugs/corner cases (student code)
  - Students enjoy participating in opensource activities
  - Students can view the source code of the tools they use
  - Tools run faster (for our labs)
- **Cons:**
  - Less mature than commercial tools
  - Error messages can be difficult for students to understand
  - Frequent updates and difficulty of managing versions/applying updates

Google Colabs for Digital Design

- Goal: Provide online teaching and training in Digital Design
  - Instructional materials teaching fundamental principles
  - Online learning widgets (simulators, etc.)
  - Design implementation (generate bitstream using F4PGA)



Colab Examples

Just think of the word AND. If input A AND input B are 1, then the output is also 1. Else, the output is 0.

Show code

Figure 1.1: AND Truth Table

A	B	F=AB	Click to Check
0	0	0	
0	1	0	
1	0	0	
1	1	0	

Source code area:

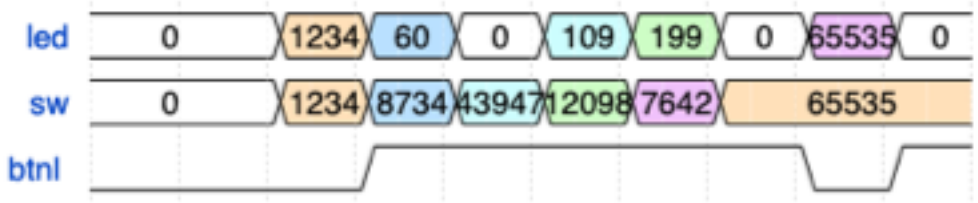
```
module function1 (
 output logic [15:0] led,
 input logic  btnd,
 input logic [15:0] sw
);

// You can type code here
// This is the answer: assign led = btnd ? sw << 3 : sw;
```

Simulation code area:

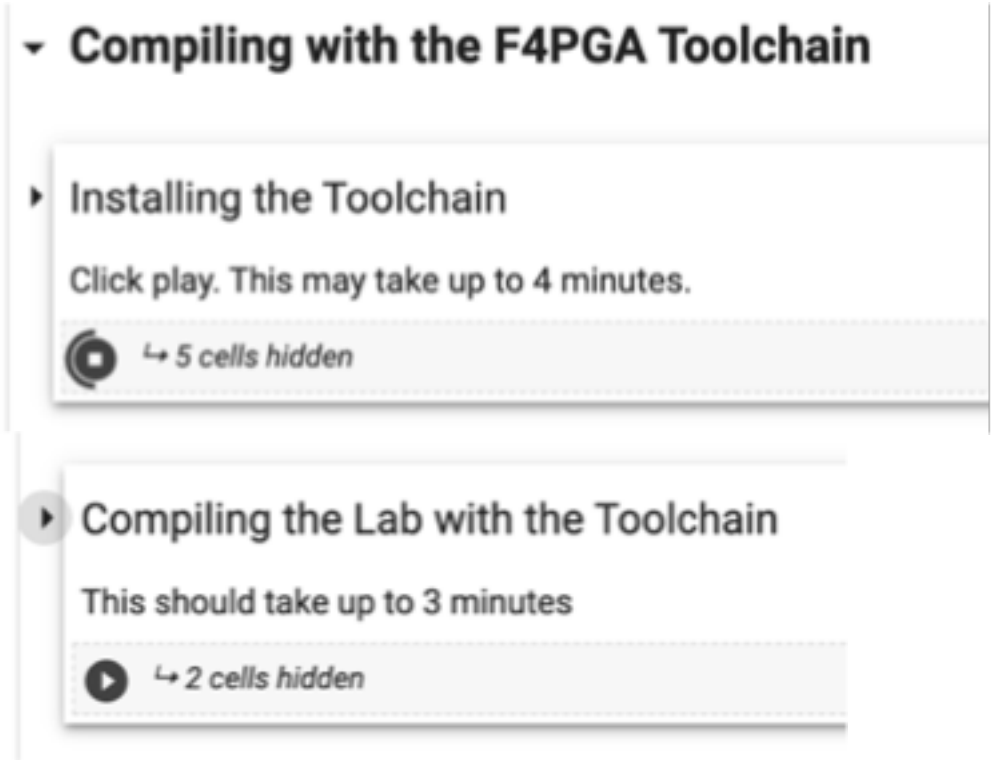
```
comb
btnd sw
 0 0
 0 11
 1 0
 1 1
 1 8010
 1 3
```

Bitwise 1

What is the value of 4'b1011&4'b1001? 0 Check



Colab Examples – F4PGA



Digital Design Education using an Open-Source, Cloud-based FPGA Toolchain

***Abstract*—Digital design tools typically have a high barrier to entry: setup is challenging for beginners, and installation often requires a high-performance workstation. In this work we present an educational framework for digital design using cloud-based resources and open-source design tools. We leverage Google Colab notebooks to allow users to run FPGA simulation and design tools remotely, using only a simple web browser. The tools are capable of running simulations with visualizations, and can generate FPGA bitstreams for commercial devices. In addition, we have developed several notebook-based labs to teach digital design topics, including arithmetic, combinational and sequential circuits, and FSMs.**

***Index Terms*—digital design, education, open-source hardware design**

Paper submission for the Design Automation Conference (DAC 2023)

Future Plans

- Analyze problems from Fall 2022 and submit appropriate issues
  - Combine similar problems together and submit simple example
  - Work with partners to identify cause and solution
- Train student TAs to understand tool flow and source code
  - Provide more information when submitting issues
  - Participate more closely in resolution of bugs
- Investigate use of F4PGA as the *primary* design tool
  - Perform experiment involving students using F4PGA *before* commercial ltools
  - How well can students debug circuits using the F4PGA flow?
    - Much higher bar: debug messages are essential
- Continue developing Colab Digital Design resources

Questions?