Hands-on with FPGA’s: Module 1

Venkat Rangan

15th June 2022

Topics

- Welcome: Jackline
- Introduction: Venkat
- Motivation for learning about HW design
- Course overview
- Resources
- Open discussion

Introduction

Venkat Rangan: Introduction

tinyvision.ai

Enabling embedded vision.

QUALCOMM®

What is Digital HW?

- Compute using only 2 digits (0, 1)

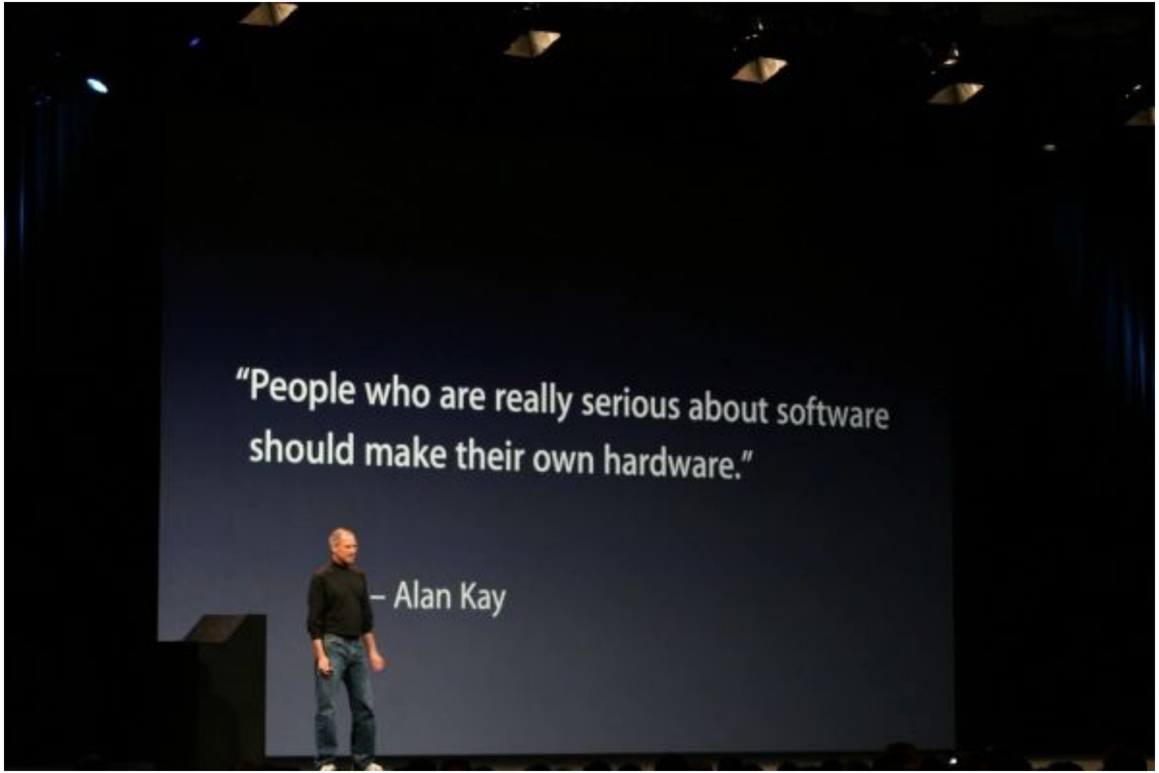
Frameworks	Web programming etc.
High Level languages	C++, Java etc.
Compiler, Operating systems	GCC, Unix, Windows
ISA	ARM, x86, RISCv
Machine level	Assembly, microcode
Micro-architecture	Register Transfer Level (RTL)
Circuits, devices	Transistors, chip layout
Physics	Chip fabrication etc.

Why learn about Digital HW?

- Basis of most modern day devices
- Fundamental knowledge required to build a real world product
- Significant advantage to knowing about hardware when dealing with embedded systems
- Jobs are high paying, can be remote and less competition than software

So why doesnt everyone learn about this?

- Wide range of topics to be learnt from physics/devices, protocols, computer science, architecture, languages…
- Very steep learning curve
- Tools and processes tend to be arcane compared to SW
- Lots of the valuable knowledge can be gained only through experience
- Lack of open source tools until recently so high bar to hands-on learning



“People who are really serious about software should make their own hardware.”

– Alan Kay

Steve Jobs, 2007

You can build your own hardware too!

Course Overview

- Hand on course
  - Use physical or simulated hardware
  - Hardware requirements are general purpose
  - Learn by doing
- Meta-course:
  - We will reuse plenty of material found on the web
  - Flexible format: will change topics to adapt/accommodate class needs
  - Will require 4-6 hours of work per week or more
    - Lots of material to cover so self learning is essential
- Targeted at engineering students but anyone is welcome to attend!
- Prerequisites
  - Basic understanding of computers, access to a Windows/Linux/Mac/RaspberryPi
  - Admin rights on machine to install programs
  - Desire to learn and apply knowledge, not just attend a course

Course overview

- Self-driven
  - No tests
  - There will be challenges at each module, your responsibility to go through them
  - 1 hour per week for guided discussions
  - Encourage working together in teams
  - Can you propose a project that 4-5 people can work on together?
    - Will provide practical experience working in a team which is how the world works
  - Goal is to introduce you to the topics, not get in depth

- Weekly remote discussion hour
  - Brief introduction to next weeks topic
  - Motivation to learn the topic: practical examples of where the topic will be used
  - Intuition behind the topic
  - Open floor for answering questions/doubts from the previous week

Possible Outcomes

- You will be able to create simple IP blocks
    - LED/Servo drivers, UART, I2C, SPI
- You will be able to run a Verilog simulation and debug it. Follow up with mapping it into an FPGA and running it on a board.
- You will know where to start finding relevant material about a wide range of digital HW topics and get introduced to the open source HW community.
- You might create your own project and get a team to work on it!

Resources

Github: https://github.com/tinyvision-ai-inc/Hands-on-FPGA-class

- Check out a list of resources

Discord: https://discord.gg/cq4tp6Zp

- Code of conduct on Discord
    - Important to learn to work with the open source community & teams
- Please use the fpga-class-discussions channel for class discussions

https://pastecode.io/ for sharing snippets of code

Google it!

Module 1: Getting started

Open Discussion