Configware in the Computer Science Curriculum

Christopher Vickery
Queens College of CUNY
CE, CS, SE, IT, and IS

- Computing Curricula 2005: The Overview Report
  - Joint project of ACM, IEEE-CS, AIS
- Five Computing Disciplines Today:
  - Computer Engineering
  - Computer Science
  - Software Engineering
  - Information Technology
  - Information Systems
From CC2005 Final Report
Figure 2.1
From CC2005 Final Report
Figures 2.3 and 2.4
Traditional Co-Design

- Software (CS)

- Hardware (CE)
Software-Driven Design

System Design and Simulation

Application Software

Device Fabric
Goal: Introduce CS Students to Software-Driven System Design

- Build on existing software skills
- Develop capabilities working with:
  - Clocking
  - Real parallelism
  - Data types
  - I/O control
Laboratory Vehicle Choices

- Schematic capture and simulation
- FPGA-based prototyping boards
  - Large range of capabilities and costs
- FPGA vendor toolchains
  - Tradeoffs between power and complexity
- System Implementation Languages
  - Availability evolving
Computer Science at Queens College

CS240 Assembly language and logic design
  - CircuitMaker (Software simulation only.)

CS343 Computer Architecture
  - Altera UP[23] boards
  - Quartus BDF/Verilog

CS345 Hardware Laboratory
  - Celoxica RC200E boards
  - DK Integrated Development Environment
Hardware Laboratory

RC200E Features
- LEDs, Buttons, Seven-Segment Displays, Touchscreen, RAM, Audio, Video, Ethernet, …
- Cost of a laptop

DK Software Environment
- Handel-C (CSP, Occam heritage)
- Platform Abstraction Layer, with Simulation
- Waveform Analyzer
- Generates EDIF for vendor toolchain processing
DK Layers

- Platform Abstraction Layer
  - Library of Generic Devices (LED, Video ...)

- Platform Support Layer
  - Provides interface to PAL for specific boards

- Pin I/O
Handel-C

- Macros
  - GCC *cpp*
  - macro proc
  - macro expr

- Statement-level clocking

- *par* blocks
  - *Loop unrolling*
  - *Runtime parallelism*

- CSP for thread synchronization (? !)

- Weird syntax for I/O
Student Assignments

- Moving average pipeline
- Sequence:
  - Keyboard to Seven-Segment Displays
  - Draw seven-segment displays on screen
  - Build framebuffer
- Servomotor controller
- UART
Student Projects

What works?
- Implement textbook CPU
- Touchscreen video games
- Voice/Video over Ethernet

What doesn’t work?
- Algorithms tied to dynamic data streams (Ogg Vorbis)
Conclusions

- CS Students *can* do hardware design.
- Not all are interested in it.
- Those who are find it highly stimulating and rewarding.
- Still learning how to do it.