ELEX 7660 Review

"I didn't know how much I didn't know"

for
B.Eng. Curriculum Committee

Ed. Casas

June 14, 2017

Background

- Course based on proposal in 2014 to improve the digital logic design part of the BEng curriculum

Survey of Digital Logic Curricula

- surveyed the digital logic design component of the EE curriculum at 11 Canadian and 8 US universities
- methodology:
  - looked at each institution’s web site for course requirements for a general EE degree
  - identified required/elective digital logic design courses
- Results:

Survey of Digital Logic Curricula (*ctd*)

- 9 of 11 programs offer more than one course in digital logic design
- 5 of the 9 programs that offer two courses require both courses, in the other 4 programs the second course is an elective
- all programs use HDLs and FPGAs to teach logic design
- the weighting of digital logic design at BCIT (2 courses, 10 credits for ELEX 1115/2115) is comparable to other EE programs
- but BCIT's courses are at lower level than other Canadian Engineering programs (e.g.: do not use HDLs for RTL design)

University	Required	Elective	HDL	Course Numbers
Toronto	1	1	Verilog	ECE241H1, ECE532H1
UBC	2	0	VHDL	EECE 259, EECE 353
McGill	2	0	HDL	ECSE 221, ECSE 323
Waterloo	1	1	HDL	ECE 124, ECE 327
Alberta	1	1	VHDL	ECE280, ECE410
Calgary	2	0		353, 453
McMaster	1	1	HDL	2DI4, 3DQ5
Ryerson	1	0	HDL	COE328
Victoria	1	0	Verilog	CENG241
SFU	3	0	VHDL	ENSC 150/250/350
Manitoba	1	2		ECE 2220, ECE 3760/3770
BCIT	2	0	VHDL	ELEX 1115/2115

Similar Courses

- The following digital design courses have a similar project component
- Students were told to look through these to get an idea of possible project ideas and scope:
  - 6.111 - Introductory Digital Systems Laboratory (and 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016)
  - Cornell - ECE5760 Advanced Microcontroller Design an d SoC
  - U of T - ECE532S - Digital Systems Design
  - Columbia - CSEE 4840 - Embedded System Design

Learning Outcomes

- Upon successful completion of this course, the student will be able to:
  - Convert an informal description of a design problem into precise interface and behavioural specifications *(this is implicitly done in the project – should the project be incorporated into the learning outcomes somehow?)*
  - Derive timing constraints from interface specifications. *(change to... write timing constraints for an interface)*
  - Iteratively decompose modules into simpler modules *(done in most projects)*
  - Define an RTL (datapath plus controller) solution for a module.
  - Write synthesizable HDL for an RTL module description.
  - Select appropriate interface methods and standards. (*low-level details of FIFO interfaces only).*
  - Integrate third-party IP into a design.
  - ~~Apply processing pipelines to meet timing requirements.~~ *(remove: no time to cover detailed design alternatives)*
  - ~~Write a behavioural description of a module in C or an HDL.~~ *(merge into objective below)*
  - Select and apply appropriate design verification techniques including importing test vectors from other software and writing behavioural test benches.
  - *Should add: convert between schematic and HDL descriptions of a circuit*

Course Management

- See course information sheet

Choice of HDL

- 2 traditional HDLs: Verilog and VHDL
  - VHDL is verbose due to lack of implicit type casting, inability to read output ports, case insensitivity, ...
  - Verilog type system (wire/reg) is very limited and confusing for beginners
- System Verilog combines the best of both languages:
  - nice type system (`logic' data type can be used for most designs)
  - implicit type casting, familiar C operators and (mostly) C-like syntax
- highly recommend using System Verilog for teaching HDL design
- maybe... someday... will do logic design in C (HLS)

Software

- used Altera (now Intel) Quartus Prime 16.1 for synthesis, ModelSim (Altera Edition) for simulation
- most students installed the software on their own machines
- pushing the limits of what can be done with free software (the free version of Modelsim has no support for useful verification features such as assertions and coverage analysis)
- obtaining eduational licenses through CMC would be an option

Hardware

- Terasic/Altera DE0-Nano board (~CAD100)
- various peripherals; typically introduce one per lab:
  - multiplexed LED display
  - matrix keypad
  - speaker
  - analog joystick (driving on-board ADC)
  - bit-mapped color graphics display
- first lab was used to build a wiring harness
- Each project group received one board and lab parts kit to use for the project
- emphasized logic level compatibility and back-EMF issues: no board fatalities in labs or projects (!)
- May be worth switching to an SoC board (for networking), possibly to Xilinx for HLS software

Project

- The distinguish feature of this course; modelled on similar courses elsewhere
- students told that project complexity should have been about same as 5 lab's worth (of code, interfaces, etc)
- half of the lab sessions (5/6) spent on set labs, half on the project
- most students came to project lab sessions; often questions were too difficult to be resolved in the lab
- students resolved most issues on their own (surprise!)
- everyone got something going in the end
- open-ended nature (choose your own project, order your own hardware, $50 additional parts cost limit):
  - was appreciated by many but a challenge for some
  - a "reality check" for many ("I didn't know how much I didn't know")
  - should help students prepare for capstone
- Might be worthwhile for other design-oriented courses (e.g. microcontroller)
- my impresssion: students spend too much time in lectures and 'set' labs and too little time learning on their own

Student Projects

In approximate order of accomplishment:

1. Digital Music Player
2. Nintendo (NES) Emulator
3. *Simon* Game
4. Autonomous Robot
5. Line-Following Vehicle
6. IR-based Distance/Direction Scanner
7. Composite Video generator
8. Digital Audio Synthesizer
9. Adaptive Street Lighting
10. Home Security System
11. Garage Door Controller
12. Audio Visualizer (spectrum analyzer)
13. Persistence of Vision Display
14. Coin Counter/Dispenser
15. Game Controller (kepad to serial port)

Project Parts

- It went surprisingly smoothly
- students filled out a spreadsheet with the item description, vendor and URL
- the spreadsheets were submitted on D2L, combined and then split up by vendor
- did one set of orders shortly after the proposal due date
- $50 limit; no shipping added if from one of 5 suggested suppliers and by due date
- some groups put much effort into mechanical design and enclosures (e.g. 3D printing); others did the bare minimum (cardboard and popsicle sticks)

Project Report Publishing

- Students were required to fill out BCIT library form LIB 73.
- This is a license that allows publication of the report under Creative Commons "CC-BY-NC-ND" terms. Reports will be published on the BCIT library's public repository web site.
- May motivate students to do a good job
- Useful reference for future students (e.g. other course sites)

Student Survey Results - Project

- 21 responses out of 30 students (all responses are scanned and available on-line)
- In the student survey I asked if the project was worthwhile and how the running of the project could be improved. Some answers:
- "Project was a time sink ... it put a huge strain on all of us."
- "Project was fun but turned out to be way too complicated and time consuming. Perhaps needs a bit more direction."
- "Keep the project open ended."
- "Try to restrain the scope of projects to ensure students don't try to do too much."
- "Project was a lot of fun, free reign to chose project scale is awesome"
- "Project was worthwhile", "definitely worthwhile"
- "Yes, most fun I've had at school. Learned lots"
- "Project was definitely worthwhile. I enjoyed it but I wish the program allowed more free time so we could focus on our project."
- "Project was worthwhile. ... more time for projects"
- "Project was best part of course!"
- "Project worthwhile, allowed to apply learning, time for project was reasonable, overall term course load too heavy"
- "The project was worthwhile: it helped me understand a lot of things. The project could be improved by having more restrictive guidelines."
- "Project was not worthwhile, I would recommend scrapping the project and build your labs by adding onto each lab week by week like adding layers."
- "The project was worthwhile. Perhaps time should be spent on clocks, ADCs and interfaces. In the labs we were just given these items working. Different projects were a good idea."

Content Changes

- should explicitly cover design of [interacting] state machines (not as an implicit part of RTL design)
- use one RTL coding style consistently in most of course; show other styles late in the course
- timing analysis needs work (difficult but necessary topic)
- simplify the partial lab solutions; these were so optimized that students found them difficult to understand or modify for use in their projects
- hand out labs earlier (some students had less than a week to prepare)
- make videos of project demos

Course-specific Suggestions

- Could use more time: 3 lecture hours/week instead of 2, 3-hour labs instead of 2
- For project work:
  - larger lab than SW1-3575 would be useful (somewhat crowded for full sets of 16 students)
  - place for students to lay out and leave their projects (most students worked at home)
  - after-hours access (relied on Telecom lab proctor hours)

Other Topics

- Copying textbooks under Access Copyright
- D2L features used:
  - publishing content
  - publishing encrypted marks
  - dropboxes for submitting assignments, labs, project documentation