❮Back to Week 1 ✕Lessons Prev Next

Instructor: Tim Scherr

I. Overview

This course will give you the foundation for FPGA design in Embedded Systems. You will learn what an FPGA is and how this technology was developed, how to select the best FPGA architecture for a given application, how to use state of the art software tools for FPGA development, and solve critical digital design problems using FPGAs. If you are thinking of a career in Electronics Design or looking at a career change, this is a great course to enhance your career opportunities.

This course consists of 4 modules, approximately 1 per week for 4 weeks. Each module will include an hour or two of video lectures, reading assignments, discussion prompts, frequently an application assignment, and an end of module assessment. Some modules include application assignments that will be peer graded, while module assessments will be auto-graded.

Hardware Requirements

You must have access to computer resources to run the development tools, a PC running either Windows 7, 8, or 10 or a recent Linux OS which must be RHEL 6.5 or CentOS Linux 6.5 or later. Either Linux OS could be run as a virtual machine under Windows 8 or 10. Whatever the OS, the computer must have at least 8 GB of RAM. Most new laptops will have this, older ones may be upgraded. A target FPGA development board, while helpful, is NOT required for this course.

Examples:

As we best learn by doing, there will be several examples that we will do together during the class. These examples will not require the purchase of a development kit although the example target is relatively inexpensive ($30) and the first one on the list below. The other boards are also useful for further investigation and may be more available:

1. Altera BeMicroMax10. Description and purchasing here: https://www.arrow.com/en/products/bemicromax10/arrow-development-tools

2. Macnica Odyssey Evaluation Kit. Description: https://www.m-pression.com/solutions/boards/odyssey-fpga, purchasing http://store.macnica-na.com/product-p/odyssey-max10-kit.htm.

3. Terasic DE10-lite. Description http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=english&No=1021, purchasing http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1021&PartNo=8.

The DE10-lite will be used as target board for the next 3 courses in the specialization.

II. Rationale

Engineers design and build systems and products for the betterment of humanity. Graduating engineers should appreciate the engineering process, be able to contribute to the development of engineering products, and do so while working in engineering organizations. Graduate Engineers should be able to conceive-design-implement-operate (CDIO) complex value-added engineering systems in a modern team-based environment.

Programmable Logic has become more and more common as a core technology used to build electronic systems. By integrating soft-core or hardcore processors, these devices have become complete systems on a chip, steadily displacing general purpose processors and ASICs. In particular, high performance systems almost always are now implemented with FPGAs. However, there are a bewildering array of Programmable Logic Devices (PLDs) available from semiconductor device suppliers, and knowing which device is best for meeting a set of requirements is a challenge. Learning to make proper choices that will determine the difference between success and failure of a product is something we will examine in detail.

III. Description and Content

The objective of this course is to acquire an understanding of programmable systems on a chip for the purpose of creating prototypes or products for a variety of applications. We will explore complexities, capabilities and trends of Field Programmable Gate Arrays (FPGA) and Complex Programmable Logic Devices (CPLD). We will learn specifics around embedded IP and processor cores, including tradeoffs between either implementing or acquiring IP. Worked examples will involve the latest FPGA development tools to help develop a broad perspective of the capabilities of various FPGA solutions. Topics include:

1. FPGA development tools flow: specify, synthesize, simulate, program and debug
2. Look-up tables for Logic Design

3. Timing Analysis and Improvement
4. Verification and Simulation
5. IP development and incorporating 3rd-party IP
6. IO Standard internals (CML, LVDS, PECL, LVCMOS, SSTL, HSTL) and pin assignments
7. Configurable embedded processors

IV. Objectives and Expected Outcomes

1. Acquire an understanding of programmable systems on a chip for the purpose of creating prototypes or products for a variety of applications.
2. Explore complexities, capabilities and trends of Field Programmable Gate Arrays (FPGA) and Complex Programmable Logic Devices (CPLD).
3. Understand and practice all aspects of FPGA development, including conception, design, implementation, and debugging.
4. Learn specifics around embedded IP and processor cores, including tradeoffs between implementing versus acquiring IP.
5. Complete a couple of example designs using FPGA development tools.

At the end of this course, students will be able to:

- Enter and compile an FPGA design using current FPGA development tools.
- Recall a list of common PLD architectures and the applications for which they are best suited.
- Appreciate complexities, capabilities and trends of Field Programmable Gate Arrays (FPGA) and Complex Programmable Logic Devices (CPLD).
- Describe the difference between an FPGA, a CPLD, an ASSP, and an ASIC.
- Recite the historical development of programmable logic devices.
- Design logic circuits using Look-up Tables (LUTs).
- Employ state of the art computer aided design tools to design and implement programmable logic solutions.
- Recall the steps in a standard FPGA design flow.
- Describe several design entry methods, including schematic, VHDL, and Verilog.
- Draw the basic FPGA Logic Cell and describe how it is used.
- Use timing closure methods to assure FPGA design reliability.

- Describe the pros and cons of FLASH-based, SRAM-based, and Anti-Fuse based FPGAs.
- Evaluate the tradeoff between implementing versus acquiring Intellectual Property (IP) cores.
- Create a programmable logic design using schematic entry.
- Implement pin assignments in an FPGA design.
- Describe the process of configuring and programming an FPGA.
- Create a NIOS II softcore processor.
- Understand and demonstrate FPGA design verification techniques, including simulation using ModelSim.

V. Requirements and Format

Prerequisites:

Knowledge of assembly and C Programming, Digital Logic Design, and basic computer architecture. Students should have a first course in each of these subjects. The corresponding CU-Boulder courses are ECEN 2120/2350, ECEN 3100/3350, and ECEN 1030/1310/CSCI 1300. To be specific, you are expected to be able to perform tasks similar to designing sequential circuits using Karnaugh maps or Boolean equations.

Participation:

Students are expected to participate in discussion prompts.

Readings:

Course materials include textbooks, papers, lecture slides, project guides, and other online materials.

Textbooks [optional]

- Rapid Prototyping of Digital Systems: SOPC Edition, by Hamblen, Hall and Furman; ISBN 9780387726700
- Design Recipes for FPGAs Using Verilog and VHDL, 2nd Edition, by Peter Wilson; ISBN 978-0-08-097129-2

Course Google Drive Repository

- Course project files posted here for paid learners to access

Other online materials

- Altera MAX10 FPGA
- DE10-lite development board
- BeMicroMAX10 development board
- Others, TBD

Application Assignments:

Expect weekly assignments giving you the opportunity to apply what you have learned from the videos or the reading. Unless otherwise stated, assignments should be submitted in either pdf or word documents, and by the deadline to be eligible for grading.

Assessments:

There will be quiz at the end of every module to assess what you have learned. For course credit, you must get 70% of the questions correct. These assessments are to be done using individual effort alone.

VI. Evaluation and Grading Procedures

The course grade will be based on application assignments and quizzes. The grade proportions are as follows:

- Application Assignments 60%
- Quizzes 40% (10% each)

Grading will be based on total points accumulated from each of these areas.

VII. Resources:

Vendors

https://www.altera.com – Altera FPGA supplier site

https://www.altera.com/events/northamerica/intel-soc-fpga-developer-forum/overview.html - SoC Developers Forum

https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/tt/tt_my_first_fpga.pdf - First FPGA design tutorial

http://www.microsemi.com/products/fpga-soc/fpga-and-soc Microsemi FPGA supplier site

http://www.latticesemi.com/en/Products.aspx - Lattice Semiconductor FPGA supplier site

http://www.xilinx.com – Xilinx FPGA supplier Site

http://www.cypress.com/products/programmable-system-chip-psoc - Cypress PSoC supplier site

http://www.quicklogic.com – Quicklogic programmable logic supplier site

http://www.achronix.com/?url=sourcetech411 – Achronix FPGA supplier site

http://www.atmel.com/products/programmable-logic/field-programmable-gate-array/default.aspx - Atmel FPGA supplier site

http://www.e2v-us.com – e2V site, military FPGAs

Media

Magazines: EDN . Electronic Design . EETimes . Circuit Cellar Ink . Popular Science . Dr. Dobbs Journal . IEEE

Magazines: Embedded Systems Design . Embedded Control Europe . Embedded Computing Design . Military Embedded Systems

http://www.eetimes.com/programmable-logic-designline.asp - Programmable Logic Designline

http://www.eetimes.com/soc-designline.asp - SoC Designline

http://www.eetimes.com/document.asp?doc_id=1274593 – how to design an FPGA from scratch

http://www.eetimes.com/author.asp?section_id=216&doc_id=1326502 – FPGAs for MCU designers

Design Tools

http://www.mathworks.com/solutions/fpga-design/ - FPGA design with Matlab

http://www.mathworks.com/products/?s_tid=gn_ps – MATLAB/Simulink for FPGA Design

http://www.altium.com/files/training/module5fpgadesign.pdf - Altium Tool for FPGA design

http://www.synopsys.com/tools/implementation/fpgaimplementation/Pages/default.aspx - Synposis Tools for FPGA Design (Synplicity)

http://www.mentor.com/products/fpga/ - Mentor tools for FPGA Design

https://www.cadence.com/rl/resources/white_papers/fpga_wp.pdf - Cadence FPGA design guidance article

https://www.aldec.com/en/solutions/fpga_design - Aldec Tools for FPGA design

http://opencores.org/projects - Open Source IP blocks for FPGAs

Tutorials

http://www.design-reuse.com/articles/18067/fpga-system-designs-methodology.html - FPGA design guide

https://embeddedmicro.com/tutorials/mojo/ - FPGA tutorial

www.fpga4fun.com/ and http://www.fpga4fun.com/FPGAsoftware1.html - FPGA HDL tools and code

http://www.fpgadeveloper.com – mostly a consultant site.

http://www.arrow.com/bemicro/ - Arrow FPGA kits

https://www.arrow.com/en/design-center - Arrow Design portal

http://www.siliconexpert.com - component search tool

https://www.sparkfun.com/news/1203 - FPGA Design Tutorial

https://electronics.stackexchange.com/questions/41528/fpga-programming-where-to-begin - FPGA FAQ

Embedded

http://www.embedded.com – Essential site for all things embedded http://www.embedded.com/design/prototyping-and-development/4006429/FPGA-programming-step-by-step - Programming FPGAs

http://www.embedded.com/education-training/courses - free classes to learn more

http://www.embedded.com/magazines - 20 years of industry articles on the best of embedded design

http://www.embedded.com/education-training/tech-papers - embedded papers

http://www.ganssle.com – Jack Ganssle, embedded guru "Perfecting the Art of Building Embedded Systems"

http://www.ganssle.com/bkreviews.htm - best books on embedded

http://www.ganssle.com/tools.htm#texteditor – best tools for embedded

http://www.barrgroup.com/Embedded-Systems/Books - more embedded books

http://www.barrgroup.com/Embedded-Systems/How-To - articles on embedded

http://www.koopman.us/embsys/books/ - even more best books

Software Tools

http://mbed.org/ - site for online tools to use with ARM processors

http://freerangefactory.org – open embedded projects site

http://www.freertos.org/ - site for FreeRTOS

http://opensource.org – initiative for open source software

https://www.96boards.org – open source hardware and software together

https://www.linaro.org – ARM ecosystem organizer

http://sourceforge.net – repository of open source software

http://www.fsf.org – Free software foundation

http://www.gnu.org - central site for free software, which is not the same as open see http://www.gnu.org/philosophy/open-source-misses-the-point.en.html

https://gcc.gnu.org – the GNU compiler

http://www.gnu.org/software/gdb/ - the GNU Debugger

https://launchpad.net/gcc-arm-embedded - GCC for ARM embedded

https://www.iar.com - IAR dev tool site

http://www.keil.com - Keil dev tool site

http://www.lauterbach.com/frames.html?home.html - Lauterbach dev tool site

http://www.isystem.com/products/software/winidea - WinIdea IDE site

https://www.eclipse.org - Eclipse IDE site

http://bpmmicro.com/programmers/ BPMicro programmer site

http://www.macraigor.com - MacGreigor programmer site

http://micrium.com - Micrium OS site

http://www.qnx.com - QNX OS site

http://www.windriver.com/products/vxworks/ - VxWorks OS Site

http://www.linux.com – Main Linux OS Site

https://www.yoctoproject.org - Yocto embedded linux package

http://www.angstrom-distribution.org – Angstrom embedded linux distribution

http://elinux.org/Main_Page - Embedded Linux site

http://www.timesys.com – commercial embedded Linux

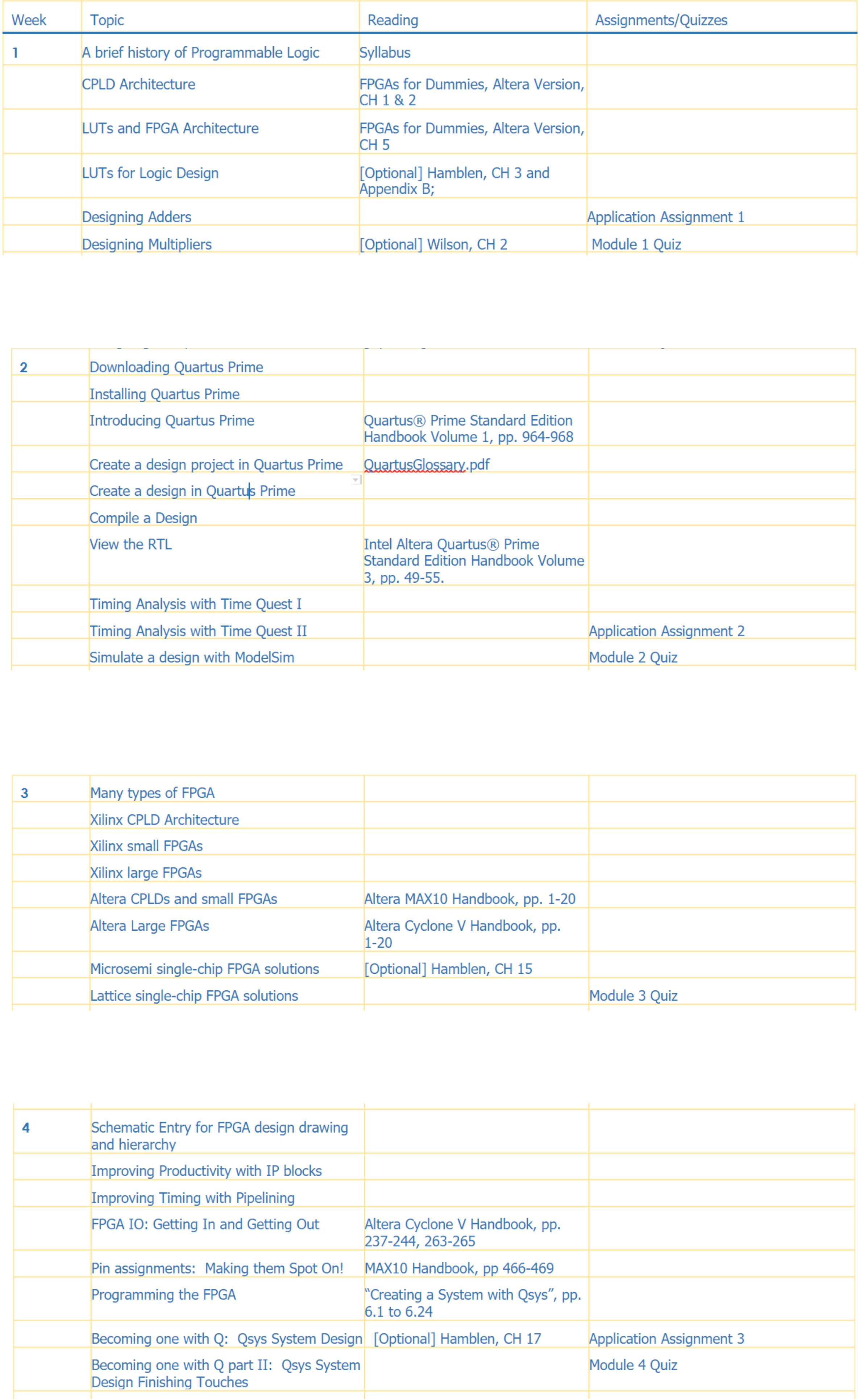
http://www.uclinux.org – Linux for processors without an MMU

http://www.openembedded.org/wiki/Main_Page - build framework for embedded linux

Essential C (Tutorial/Refresher), Programming Information

The C Book (free on-line book) . PDF Version

VIII. Course Outline:



Week	Topic	Reading	Assignments/Quizzes
1	A brief history of Programmable Logic	Syllabus	
	CPLD Architecture	FPGAs for Dummies, Altera Version, CH 1 & 2	
	LUTs and FPGA Architecture	FPGAs for Dummies, Altera Version, CH 5	
	LUTs for Logic Design	[Optional] Hamblen, CH 3 and Appendix B;	
	Designing Adders		Application Assignment 1
	Designing Multipliers	[Optional] Wilson, CH 2	Module 1 Quiz
2	Downloading Quartus Prime		
	Installing Quartus Prime		
	Introducing Quartus Prime	Quartus® Prime Standard Edition Handbook Volume 1, pp. 964-968	
	Create a design project in Quartus Prime	QuartusGlossary.pdf	
	Create a design in Quartus Prime		
	Compile a Design		
	View the RTL	Intel Altera Quartus® Prime Standard Edition Handbook Volume 3, pp. 49-55.	
	Timing Analysis with Time Quest I		
	Timing Analysis with Time Quest II		Application Assignment 2
	Simulate a design with ModelSim		Module 2 Quiz
3	Many types of FPGA		
	Xilinx CPLD Architecture		
	Xilinx small FPGAs		
	Xilinx large FPGAs		
	Altera CPLDs and small FPGAs	Altera MAX10 Handbook, pp. 1-20	
	Altera Large FPGAs	Altera Cyclone V Handbook, pp. 1-20	
	Microsemi single-chip FPGA solutions	[Optional] Hamblen, CH 15	
	Lattice single-chip FPGA solutions		Module 3 Quiz
4	Schematic Entry for FPGA design drawing and hierarchy		
	Improving Productivity with IP blocks		
	Improving Timing with Pipelining		
	FPGA IO: Getting In and Getting Out	Altera Cyclone V Handbook, pp. 237-244, 263-265	
	Pin assignments: Making them Spot On!	MAX10 Handbook, pp 466-469	
	Programming the FPGA	"Creating a System with Qsys", pp. 6.1 to 6.24	
	Becoming one with Q: Qsys System Design	[Optional] Hamblen, CH 17	Application Assignment 3
	Becoming one with Q part II: Qsys System Design Finishing Touches		Module 4 Quiz

Mark as completed

👍 👎 ⚑