ECE580B4 FPGA Signal Processing and Software-Defined Radio

Spring 2019

Lectures: 10:00 AM -- 10:50 AM, Mon/Wed/Fri, Weber 202

Instructor: Jesse Wilson

Email: jesse.wilson@colostate.edu

Skype Messaging: jessew@colostate.edu

Phone: 970-491-3706

Office Hours: TBD

This is the first offering of this experimental course. This syllabus is subject to change. Your feedback is always welcome, and will be valuable for shaping this class for future students.

COURSE OBJECTIVES:

The student successfully completing this course will be able to:

- Distinguish FPGA vs microprocessor-based DSP implementation, explain tradeoffs between them, and select an architecture by weighing power consumption, bandwidth, and development effort considerations.
- Identify and explain consequences of analog/digital conversion of radio-frequency signals, such as aliasing and effective bit gain from oversampling.
- Distinguish and select between and fixed-point and floating-point arithmetic, depending on design objectives, and utilize IEEE math libraries to implement basic operations in either format.
- Construct and test in Verilog or VHDL the basic signal processing components: Numerically-controlled oscillator (NCO), mixers, finite- and infinite-impulse response filters (FIR, IIR), adaptive filters, and cascaded integrator-comb downsampling and upsampling (CIC) filters.
- Explain the Fast-Fourier Transform, identify how its implementations on an FPGA differs from on a DSP microprocessor, and construct it in Verilog or VHDL.
- Configure and make use of commercially-available intellectual property (IP) blocks to speed up DSP implementation.
- Configure and integrate on-chip soft CPU to reduce development time of functionality better suited to microprocessors (e.g. configuration, control, serial communications)
- Design and build all the digital elements of a basic communications system, including amplitude modulation of an NCO-generated carrier, transmission through a simulated channel, and demodulation with a digital downconverter (DDC).

- Engage in continued learning beyond this course, to be able to design and implement FPGA-based DSP systems with more sophisticated components (e.g. wavelets, spread-spectrum coding, real-time video processing)

PREREQUISITES: ECE311, ECE312, ECE451

REQUIRED BACKGROUND KNOWLEDGE:

- VHDL or Verilog experience
- Binary arithmetic and computer number systems
- Linear systems concepts, e.g. convolution, Fourier series, ...
- Digital filters, z transform, etc.

REQUIRED MATERIALS:

- *Digital Signal Processing with Field Programmable Gate Arrays,* 4th ed. by Uwe Meyer-Baese. Springer, 2014. (Accessible online through CSU library website, but hardcopy strongly recommended)
- MATLAB software.
- Xilinx Vivado Design Suite or Altera Quartus. (ENS installation pending)
- A Verilog textbook (search CSU library website; there are a few options with online e-book access).

Canvas: canvas.colostate.edu will have the syllabus, links, homework, course grades and other postings. It is your responsibility to check the calendar under the Index tab each week for new postings.

COURSE TOPICS: The planned topics for this course are:

Weeks 1-2	Overview, FPGA and Verilog review, binary arithmetic
Week 3	FIR and IIR filter theory (Chandra)
Weeks 4-5	FPGA architectures for digital filters
Week 6	Adaptive filters; **Midterm 1**
Weeks7-8	Multirate signal processing
Weeks 9-10	Fourier Transform on FPGAs
Weeks 11-12	Communications Systems; **Midterm 2**
Week 13	On-chip microprocessors
Week 14	FPGA-host data transfer
Week 15	Current topics (DDC for radar?)
Week 16	Finals week

GRADING:

Quizzes (online and in-class): 10%

Homework and Verilog coding assignments: 40%

Midterm Exams: 25%

Final Project: 25%

A+	100%	to 96.67%
A	< 96.67%	to 93.33%
A-	< 93.33%	to 90%
B+	< 90%	to 86.67%
B	< 86.67%	to 83.33%
B-	< 83.33%	to 80%
C+	< 80%	to 76.67%
C	< 76.67%	to 70%
D	< 70%	to 60%
F	< 60%	to 0%

HOMEWORK:

All submitted homework and code must be your own individual work. Since a large portion of the work will be writing code, students are expected to adhere to the Academic Integrity Policies found on the Computer Science Department website: http://www.cs.colostate.edu/cstop/csacademics/student_info.php

Each coding assignment must be submitted with a written report, describing the background/motivation for the problem, your approach (including relevant highlights from your code), obstacles and pitfalls encountered, and results.

ACADEMIC INTEGRITY: Students are expected to adhere to the Academic Integrity Policy of Colorado State University, outlined in the CSU General Catalog. Students are also expected to follow the Student Conduct Code which can be found at www.conflictresolution.colostate.edu. Academic dishonesty is not accepted in this course, and any form of cheating (including plagiarism) will be reported. Penalties may include a lowered course grade, loss of course credit, and expulsion from the university.