ECE 448
FPGA and ASIC Design with VHDL

Spring 2018
Kris Gaj

Research and teaching interests:

• reconfigurable computing
• cryptography
• computer arithmetic
• network security

Contact:
The Engineering Building, room 3225
kgaj@gmu.edu
Course Web Page

Google “Kris Gaj” →

ECE 448 FPGA and ASIC Design with VHDL
Farnoud Farahmand
Ph.D. Student of Dr. Gaj
Member of the Cryptographic Engineering Research Group (CERG)
http://cryptography.gmu.edu
since Fall 2014.

Internship as a Hardware Engineer in the Crypto Group of Google in Summer 2017.
Undergraduate Computer Engineering Courses

ECE 331 → ECE 332
ECE 445
ECE 448 → ECE 492
ECE 447 → ECE 493

Color code:
BS EE
BS CpE
Computer Engineering Course Progression

This is not a suggested schedule. It only illustrates dependencies and shows courses in earliest possible semester.

Must be taken in sequence

Co–Requisite
Should be taken concurrently but not earlier

Co–Requisite +
Suggested to be taken in sequence

Semester
Courses between dashed lines can be taken concurrently

*) Math 203 can be taken concurrently with ECE 220
Electrical Engineering Course Progression

This is not a suggested schedule. It only illustrates dependencies and shows courses in earliest possible semester.

*) Math 203 can be taken concurrently with ECE 220

---

Prerequisite
Must be taken in sequence

Co-Requisite
Should be taken concurrently but not earlier

Co-Requisite +
Suggested to be taken in sequence

Semester
Courses between dashed lines can be taken concurrently
Digital system design technologies coverage in the CpE & EE programs at GMU

Microprocessors
- ECE 445 Computer Organization
- ECE 447 Single Chip Microcomputers
- ECE 511 Microprocessors
- ECE 611 Advanced Microprocessors
- ECE 612 Real-Time Embedded Systems
- ECE 615 Software/Hardware Codesign

FPGAs
- ECE 448 FPGA and ASIC Design with VHDL
- ECE 545 Digital System Design with VHDL
- ECE 645 Computer Arithmetic

ASICs
- ECE 431 Digital Circuit Design
- ECE 586 Digital Integrated Circuits
- ECE 681 VLSI Design for ASICs
Course Hours

Lecture:
Monday, Wednesday
1:30-2:45 PM, Blue Ridge Hall, room 129

Lab Sessions:
Tuesday
9:00-11:40 AM

Wednesday
7:20-10:00 PM

Friday
8:30-11:20 AM

The Nguyen Engineering Bldg., room 3208
General Section Assignment Rules

• You should do your best to attend all lab meetings of the section you are registered for

• If you have missed a meeting of your section please attend a meetings of another section, but give preference in access to the lab computers to the students attending their own lab section

• All lab assignment demos should be normally done exclusively during the class time of your section

• Any requests for exceptions to these rules (due to illness, accident, etc.) should be well documented and presented to the TA & primary instructor for approval
Office Hours

You are welcome to attend all office hour sessions!
You can direct your questions regarding lab assignments to the TA and the instructor.

Do your best to avoid “chasing” the TA outside of his office hours!
He has other important and often urgent work to do!

Farnoud Farahmand
• Monday, 12:00-1:00pm, ENGR 3204
• Tuesday, 5:30-7:00pm, ENGR 3204
• Thursday, 12:00-1:30pm, ENGR 3204

Kris Gaj
• Monday, 3:00-4:00pm, ENGR 3225
• Wednesday, 3:00-4:00pm, ENGR 3225
Getting Help Outside of Office Hours

piazza

• System for asking questions 24/7
• Answers can be given by students and instructors
• Student answers endorsed (or corrected) by instructors
• Average response time in Spring 2016 = 48 minutes
• You can submit your questions anonymously
• You can ask private questions visible only to the instructors
Lab Access Rules and Behavior Code

Please refer to

ECE Labs website

and in particular to

Access rules & behavior code
# Course Web Page

http://ece.gmu.edu/coursewebpages/ECE/ECE448/S18

<table>
<thead>
<tr>
<th>Organization</th>
<th>Lecture</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor</td>
<td>Textbooks</td>
<td>Rules</td>
</tr>
<tr>
<td>Teaching Assistant</td>
<td>Lecture Slides</td>
<td>Lab Assignments</td>
</tr>
<tr>
<td>Lecture and Lab Time</td>
<td>Homework</td>
<td>Lab Slides &amp; Examples</td>
</tr>
<tr>
<td>Office Hours</td>
<td>Past Quizzes</td>
<td>Software</td>
</tr>
<tr>
<td>Grading</td>
<td>Past Midterm Exams</td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Past Final Exams</td>
<td>Useful References</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Past Lab Exams</td>
</tr>
</tbody>
</table>
Grading criteria

First part of the semester (before the Spring break)

Lab experiments - Part I
16%

Lab exercises – Part I
1.6%

Quizzes & homework: 3%
Midterm exam for the lecture: 10%
Midterm exam for the lab: 15%

Second part of the semester (after the Spring break)

Lab experiments - Part II
24%+2%

Lab exercises – Part I
2.4%

Quizzes & homework: 3%
Final exam 25%
## Tentative Grading Scheme for the Labs

| Lab 1: Developing VHDL Testbenches | – 4 points |
| Lab 2: Combinational & Sequential Logic | – 4 points |
| Lab 3: State Machines. Basic I/O Devices. | – 8 points |
| **Total:** | **16 points** |
| Lab 4: VGA Display | – 8 points |
| Lab 5: Computer Vision / HLS | – 8 points |
| Lab 6: Software/Hardware Codesign | – 8 points |
| Lab 7: Logic Analyzer (in class) | – 2 bonus points |
| **Total:** | **24+2 points** |
Penalties and Bonus Points

Penalties:

one-week delay: 1/3 of points

i.e., you can earn max. 4 out of 6 points

No submissions or demos will be accepted more than one week after the assignment is due!

Bonus points:

Majority of labs will have opportunities for earning bonus points by doing additional tasks
# Flexibility in the Second Part of the Semester

## Schedule A:

| Lab 4: VGA display | (2 weeks) | – 8 points |
| Lab 5: Computer Vision / HLS | (2 weeks) | – 8 points |
| Lab 6: SW/HW Codesign | (2 weeks) | – 8 points |
| Lab 7: Logic Analyzer | (in class) | – 2 bonus points |

**Total:** 24+2 points

## Schedule B:

| Lab 4: VGA display | (3 weeks) | – 8 points |
| Lab 5: Computer Vision / HLS *or* Lab 6: SW/HW Codesign | (3 weeks) | – 8 points |
| Lab 7: Logic Analyzer | (in class) | – 2 bonus points |

**Total:** 16+2 points
Flexibility in the Second Part of the Semester

Schedule A+:

• Intended for students who do very well in the first part of the semester (preferably $\geq 90\%$ of points for Labs 1-3)
• An open-ended project proposed by students, the TAs, or the instructor
• Can be done individually or in groups of two students

Schedule:

- Detailed Specification (1 week)
- Milestone 1 (2 weeks)
- Milestone 2 (2 weeks)
- Final Report & Deliverable (1 week)

Total: 25 points
Bonus Points for Class & Piazza Activity

• Based on class exercises during lecture and lab sessions, as well as your answers to other student’s questions on Piazza
• “Small” points earned each week posted on BlackBoard
• Up to 5 “big” bonus points
• Scaled based on the performance of the best student

For example:

<table>
<thead>
<tr>
<th></th>
<th>Small points</th>
<th>Big points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Alice</td>
<td>40</td>
<td>5</td>
</tr>
<tr>
<td>2. Bob</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>26. Zach</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>
LinkedIn Accounts

- LinkedIn Accounts with photos encouraged
- Facilitate associating name with the face during class exercises
- Endorsements and recommendation letters available at the end of the class (per your request)
Exams

• Midterm Exam for the Lecture – 10 points
  Wednesday, February 28

• Midterm Exam for the Lab (hands-on) – 15 points
  Tuesday, March 6
  Wednesday, March 7
  Friday, March 9

• Final Exam – 25 points
  Wednesday, May 9, 1:30-4:15pm
  (first day of the exam session)
Required Textbook

Supplementary Textbook – Basics Refresher

Supplementary Textbook – Advanced

ECE 448, FPGA and ASIC Design with VHDL

Topics

**VHDL:**
- writing testbenches
- writing synthesizable RTL level code in VHDL

**FPGAs:**
- architecture of FPGA devices
- embedded resources (memories, DSP units)
- tools for the computer-aided design with FPGAs
- current FPGA families & future trends
**Applications:**
- basics of computer arithmetic
- applications from communications, computer vision, cryptography, etc.

**Platforms & Interfaces:**
- FPGA boards
- I/O modules (VGA controller, serial communication modules)

**New Trends:**
- microprocessors embedded in FPGAs (MicroBlaze, ARM)
- using high-level programming languages to design hardware

**High-level ASIC Design:**
- standard cell implementation approach
- logic synthesis tools
- differences between FPGA & standard-cell ASIC design flow
Tasks of the course

Advanced course on digital system design with VHDL
- writing VHDL code for synthesis
- design using division into the datapath & controller
- testbenches

Comprehensive introduction to FPGA & front-end ASIC technology
- hardware: Xilinx FPGAs, Library of standard ASIC cells
- software: VHDL simulators, Synthesis tools, Implementation tools

Testing equipment
- oscilloscopes
- logic analyzer
Levels of design description

Algorithmic level

Register Transfer Level

Logic (gate) level

Circuit (transistor) level

Physical (layout) level

Levels supported by HDL

Level of description most suitable for synthesis
Register Transfer Level (RTL)
Design Description
What is an FPGA?

- Block RAMs
- Configurable Logic Blocks
- I/O Blocks
- Block RAMs
Modern FPGA

(#Logic resources, #Multipliers/DSP units, #RAM_blocks)

Graphics based on The Design Warrior’s Guide to FPGAs
Devices, Tools, and Flows. ISBN 0750676043

Copyright © 2004 Mentor Graphics Corp. (www.mentor.com)
General structure of an FPGA
Two competing implementation approaches

**ASIC**

- Designed all the way from behavioral description to **physical layout**
- Designs must be sent for expensive and time consuming **fabrication** in semiconductor foundry

**FPGA**

- No physical layout design; design ends with a **bitstream** used to configure a device
- Bought **off the shelf** and reconfigured by designers themselves
FPGAs vs. ASICs

**ASICs**
- High performance
- Low power
- Low cost (but only in high volumes)

**FPGAs**
- Off-the-shelf
- Low development costs
- Short time to the market
- Reconfigurability

Low power (in both ASICs and FPGAs)
Design and implement a simple unit permitting to speed up encryption with RC5-similar cipher with fixed key set on 8031 microcontroller. Unlike in the experiment 5, this time your unit has to be able to perform an encryption algorithm by itself, executing 32 rounds.

Library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity RC5_core is
port(
  clock, reset, encr_decr: in std_logic;
  data_input: in std_logic_vector(31 downto 0);
  data_output: out std_logic_vector(31 downto 0);
  out_full: in std_logic;
  key_input: in std_logic_vector(31 downto 0);
  key_read: out std_logic;
); end AES_core;

Specification (Lab Assignments)

On-paper hardware design
(Block diagram & ASM chart)

VHDL description (Your Source Files)

Functional simulation

Synthesis

Post-synthesis simulation
FPGA Design process (2)

Implementation

Configuration

Timing simulation

On chip testing
FPGA Tools
Xilinx ISE
Integrated Software Environment
(not used this semester)

• Older development environment
• Frozen at version 14.7 in 2012 (although new releases of v14.7 still posted periodically)
• Replaced by Xilinx Vivado for the new FPGA families, starting at Series-7 FPGAs (Artix-7, Kintex-7, Virtex-7)
• Used in all previous offerings of ECE 448
# Support for Xilinx Families

<table>
<thead>
<tr>
<th>Technology</th>
<th>Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 nm</td>
<td>Spartan-3, Virtex-4</td>
</tr>
<tr>
<td>65 nm</td>
<td>Virtex-5</td>
</tr>
<tr>
<td>45 nm</td>
<td>Spartan-6</td>
</tr>
<tr>
<td>40 nm</td>
<td>Virtex-6</td>
</tr>
<tr>
<td>28 nm</td>
<td>Artix-7, Kintex-7,</td>
</tr>
<tr>
<td></td>
<td>Virtex-7</td>
</tr>
<tr>
<td>20 nm</td>
<td>Kintex-7 UltraSCALE,</td>
</tr>
<tr>
<td></td>
<td>Virtex-7 UltraSCALE</td>
</tr>
<tr>
<td>16 nm</td>
<td>Kintex-7 UltraSCALE+,</td>
</tr>
<tr>
<td></td>
<td>Virtex-7 UltraSCALE+</td>
</tr>
</tbody>
</table>

**Future Families**
Vivado Design Suite
(used for the first time this year)

• 4 years of development and 1 year of beta testing
• first version released in Summer 2012
we will use version 2017.2
• scalable data model, supporting designs with
up to 100 million ASIC gate equivalents (GEs)
• Support for VHDL-2008
• Support for High-Level Synthesis
Productivity Gains

- Vivado Simulator 3x faster than ISim
- Synthesis 3x faster than Xilinx XST (part of ISE)
- Substantial improvement in runtime and maximum design size compared to Xilinx ISE
- Much better visibility into key design metrics, such as timing, power, resource utilization, and routing congestion much earlier during the design process
- Estimates becomes progressively more accurate
Additional Simulation Tool

ModelSim®

ModelSim-Intel FPGA Starter Edition

ModelSim:
• Industry standard for simulation
• Significantly faster than Vivado Simulator
• Windows, Linux OS
• Mixed-language support: VHDL, Verilog, System Verilog
• Recommended for advanced users

Features of the Starter Edition:
• Free, no license required
• 10,000 executable line limit
Why ECE 448 is a challenging course?

• need to refresh and strengthen your VHDL skills

• need to learn new tools

• need to perform practical experiments

• time needed to complete experiments
Difficulties
(based on a student survey)

• finding time to do the labs  – 15

• learning VHDL  – 2

• getting used to software  – 1
Self-evaluation (based on a student survey)

- 16 – as well as expected
- 8 – worse than expected
- 3 – better than expected
Why is this course worth taking?

- **VHDL** for synthesis: one of the most sought-after skills
- Knowledge of state-of-the-art tools used in the industry
- Knowledge of the modern FPGA & ASIC technologies
- Knowledge of state-of-the-art testing equipment
- Design portfolio that can be used during job interviews
- Unique knowledge and practical skills that make you competitive on the job market