Kris Gaj

Research and teaching interests:

• FPGA design
• cryptography & computer arithmetic
• software/hardware codesign
• high-level synthesis

Contact:

Engineering Bldg., room 3225
kgaj@gmu.edu

Office hours:  Monday,  3:00-4:00 PM,
              Wednesday, 3:00-4:00 PM,
              Thursday,  6:00-7:00 PM,
and by appointment
Farnoud Farahmand

MS CpE student,
working on a Master’s thesis devoted to software/hardware co-design and high-level synthesis of cryptographic algorithms,
member of the Cryptographic Engineering Research Group (CERG)
https://cryptography.gmu.edu
Farnoud Farahmand

Research and teaching interests:
• design of hardware accelerators
• software/hardware interfaces (AXI, PCIe)
• high-level synthesis
• cryptography

Contact:
Engineering Bldg., room 3224
kgaj@gmu.edu

Office hours: Wednesday, 6:00-8:00 PM, and by appointment
Getting Help Outside of Office Hours

- 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 ECE 545 in Fall 2015 = 2 hours**
- You can submit your questions anonymously
- You can ask private questions visible only to the instructors
A few words about You

- 3 MS CpE students
- 2 MS EE students
- 2 PhD ECE students
- 1 NDG EE student
MICROPROCESSOR AND EMBEDDED SYSTEMS

1. ECE 510 Real-Time Concepts
   – P. Pachowicz, project, design of real-time systems

2. ECE 511 Microprocessors
   – J.P. Kaps, K. Lilly, project, system based on MSP430 microcontroller

3. ECE 611 Advanced Microprocessors
   – A. Sasan, H. Homayoun, project, computer architecture simulation tools

4. ECE 612 Real-Time Embedded System
   – C. Sabzevari, project, programming distributed real-time systems

5. ECE 641 Computer System Architecture
   – H. Homayoun, project, computer architecture simulation tools

6. ECE 699 Software/Hardware Codesign
   – K. Gaj, project, SoC design with VHDL and C

7. ECE 699 Heterogeneous Architectures and Green Computing
   – H. Homayoun, project, computer architecture simulation tools
DIGITAL SYSTEMS DESIGN

1. ECE 545 Digital System Design with VHDL  
   – K. Gaj, project, FPGA design with VHDL

2. ECE 586 Digital Integrated Circuits  
   – D. Ioannou, homework, small projects

3. ECE 645 Computer Arithmetic  
   – K. Gaj, project, FPGA design with VHDL or Verilog

4. ECE 681 VLSI Design for ASICs  
   – H. Homayoun, A. Sasan, project/lab, ASIC design with Synopsys tools

5. ECE 682 VLSI Test Concepts  
   – T. Storey, homework

6. ECE 699 Software/Hardware Codesign  
   – K. Gaj, project, SoC design with VHDL and C

7. ECE 699 Digital Signal Processing Hardware Architectures  
   – A. Cohen, project, FPGA design with VHDL and Matlab/Simulink
Prerequisites

• ECE 511 Microprocessors
• ECE 545 Digital System Design with VHDL

Useful Knowledge

• Basics of computer organization
• High level programming language (preferably C)
• RTL design with VHDL
• FPGA devices and tools
Course web page

ECE web page → Courses → Software/Hardware Codesign

Google “Kris Gaj” → ECE 699 Software/Hardware Codesign

http://ece.gmu.edu/coursewebpages/ECE/ECE699_SW_HW/S16
Grading Scheme

Concepts
- Midterm Exam - 20%
- Final Exam - 30%

Practice
- Class Exercises - 5%
- Homework - 10%
- Project - 35%
Bonus Points for Class Activity

- Based on answers provided during the lecture and 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>36</td>
<td>4.5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12. Charlie</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>
Literature (1)

Required Textbooks:

L.H. Crockett, R.A. Elliot, M.A. Enderwitz, R.W. Stewart, and D. Northcote, University of Strathclyde, Glasgow, UK

• *The Zynq Book*: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC
• *The Zynq Book Tutorials*

PDF copies available for free at http://www.zynqbook.com
Supplementary Textbooks:


C & VHDL Resources:


Other Resources

• Video Tutorials
• Tutorials
• Reference Manuals
• User Guides
• Journals
• On-line C Resources
• On-line VHDL Resources
Exams

Midterm Exam   – 2 hrs 40 minutes, in class

Final Exam     – 2 hrs 45 minutes, in class comprehensive

Tentative days of the exams:

Midterm Exam:  Thursday, March 24, 7:20-10:00 PM
Final Exam:    Thursday, May 5,   7:30-10:15 PM
Class Exercises, Homework, & Project

• Based on the **Digilent ZYBO Zynq-7000 Development Board** (distributed for free at the beginning of the semester, and collected at the end of the semester)
  The use of more powerful **ZedBoard** allowed, but no such boards available for loan.

• Involve **Xilinx Vivado Design Suite**
  (preferably to be installed on your laptops and desktops at home)

• Can be **done individually or in a group of two students**
  (group assignments will involve a larger number of tasks and/or more time-consuming tasks)
Class & Homework Exercises

- **Deliverables**, typically due on Wednesday @ 6:00 PM, to be submitted **on Blackboard**

- The corresponding **demo** on Wednesday, 6:00-8:00 PM, or after the class

- No deliverables **or** no demo = **one-week late submission**, penalized by 33% of the maximum score

- No submissions accepted more than one week after the deadline

- **Honor code** strictly enforced
Project

- Can be done individually or in a group of two students
- Semester-long
- Area & topic of your choice
- Can be used to
  - fulfill new scholarly paper requirements
  - start or advance your Master’s Thesis research
  - advance your Ph.D. Thesis research
  - develop a paper to be submitted to a conference or a journal
  - develop an open-source project to be placed in public domain
  - develop new lab exercises
Project – Common Requirements

- Specification
- Literature Analysis
- Progress Reports
- Deliverables
- Demo
- Written Report
- Short Oral Presentation
Project – Common Project Tasks

• Locating **C implementation** and porting it to the ARM of Zynq-7000

• **Profiling**

• Software/hardware **partitioning**

• Development of a **hardware accelerator** using existing IP cores and **RTL** design

• Establishing and optimizing **communication** between the microprocessor and the hardware accelerator

• Experimental **testing**

• Repeating the **hardware accelerator** design using **HLS**

• **Timing** measurements
Project – Possible Extensions/Modifications

- Using domain-specific **languages and tools**, such as Matlab/Simulink
- Using domain-specific **software libraries and hardware IPs**, such as OpenCV - an open source computer vision and machine learning software library
- Running your application under **Linux**
Project – Topics Proposed by the Instructor

Efficient software/hardware implementations of:

- Authenticated ciphers
  (possible extension of your ECE 545 project)
- Hash functions
- Post-Quantum Cryptosystems (PQC)
Post-Quantum Cryptography (PQC)

• PQC refers to cryptographic algorithms (usually public-key algorithms) that are thought to be secure against an attack by a quantum computer.

• Security of traditional public-key algorithms relies on
  - Integer factorization problem,
  - Discrete logarithm problem or
  - Elliptic curve discrete logarithm problem.

• All of these problems can be easily solved on a sufficiently large quantum computer running Shor's algorithm, if such a computer is ever built.
## Short History of Quantum Computers (1)

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1985</td>
<td>David Deutsch came up with the idea of quantum logic gates.</td>
</tr>
<tr>
<td>1994</td>
<td>Peter Shor designed a quantum algorithm for factoring integers [1].</td>
</tr>
<tr>
<td>1996</td>
<td>Lov Grover formulated a quantum algorithm capable of reducing the time necessary to break a secret-key cipher from $2^n$ to $2^{\frac{n}{2}}$ operations.</td>
</tr>
<tr>
<td>1998</td>
<td>First quantum computer built using 2 qubits.</td>
</tr>
<tr>
<td>2000</td>
<td>A 7-qubit quantum computer developed by Los Alamos National Laboratory.</td>
</tr>
<tr>
<td>2001</td>
<td>IBM demonstrated Shor’s Algorithm by factoring 15 using a Nuclear Magnetic Resonance quantum computer with 7 qubits.</td>
</tr>
<tr>
<td>2005</td>
<td>The first qubyte was created by the Institute of Quantum Optics and Quantum Information of the University of Innsbruck based on ion traps.</td>
</tr>
</tbody>
</table>
# Short History of Quantum Computers (2)

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>Scientists in Massachusetts established methods for controlling a 12-qubit system.</td>
</tr>
<tr>
<td>2007</td>
<td>A Canadian startup, D-Wave, successfully demonstrated a 16-qubit quantum computer which was able to solve a Sudoku puzzle.</td>
</tr>
<tr>
<td>2011</td>
<td>D-Wave Systems claimed developing a 128-qubit processor chipset.</td>
</tr>
<tr>
<td>2011</td>
<td>Proof that a quantum computer can be made with a Von Neumann architecture (separation of RAM).</td>
</tr>
<tr>
<td>2011</td>
<td>Physicists at the University of Science and Technology of China in Hefei, factored 143 into prime factors 11 and 13 using just 4 qubits.</td>
</tr>
<tr>
<td>2013</td>
<td>Google announced launching a “Quantum Artificial Intelligence Lab,” holding a 512-qubit quantum computer developed by D-Wave Systems.</td>
</tr>
<tr>
<td>2013</td>
<td>An international team of researchers led by Mike Thewalt of Simon Fraser University in Canada were able to maintain the superposition states of qubits for an entire 39 minutes, thus breaking all previous records by a wide margin [2].</td>
</tr>
</tbody>
</table>
# PQC Families of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Hash-based</th>
<th>Code-based</th>
<th>Lattice-based</th>
<th>Multivariate</th>
</tr>
</thead>
</table>
## Major PQC Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Hash-based</th>
<th>Code-based</th>
<th>Lattice-based</th>
<th>Multivariate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encryption</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Niederreiter (1986) [10]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Identification schemes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Identity based encryption</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Very Few Hardware & Embedded Software Implementations

<table>
<thead>
<tr>
<th></th>
<th>Hash-based</th>
<th>Code-based</th>
<th>Lattice-based</th>
<th>Multivariate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller implementations</td>
<td>Merkle [65, 66]</td>
<td>QC-MDPC McEliece [58, 61, 67]</td>
<td>BLISS [68], Ring-LWE [69, 70]</td>
<td></td>
</tr>
</tbody>
</table>
Scope of Research: Platforms & Design Strategies

Platform Options
- Micro-controllers
- Micro-processors
- FPGAs
- ASICs
- Discrete FPGA-Processor Comb.

Design Strategies
- Register-Transfer Level (RTL)
- High-Level Synthesis (HLS)

HW/ SW Codesign

Software

Hardware

SOCs

Preliminary work by several students:
Brian Loop
Ahmed Ferozpuri
Malik Umar Sharif
Rabia Shahid
Project – Topics Proposed by the Students

Efficient software/hardware implementations of:

- Computer Vision and Machine Learning Algorithms
- Software Defined Radio / Communications
- Digital Signal Processing
- Bioengineering / Medical Applications
- Big Data
- Control & Robotics