ECE 353
Lab 3 Overview

Prof. Daniel Holcomb
Brief Intro of Instructor

- Joined UMass Jan 2015
- Previously
  - BS @ UMass (Atmospheric research balloon payload)
  - MS @ UMass (IC Fingerprinting)
  - PhD @ UC Berkeley (QoS verification of on-chip networks)
  - Research Fellow @ Michigan (Embedded security)

- Research Interests
  - Embedded Systems: 597MB
  - VLSI Circuit Design: 558/559
  - Hardware Security
  - Formal Verification
Class Information for Labs 3 and 4

- http://ece353.ecs.umass.edu
- My office hours
  - Tu/Th 10AM-11AM in KEB 309H
  - Or send email for appointment
- Grading
  - Lab 3 and Lab 4 equal weight (each 17.5% of final grade)
  - Final exam (30% of grade) – 50% of questions from Lab 3 and 4
- Lab Kits:
  - Keith Shimeld
  - Kits contain *everything* needed for lab
- Duda Hall
  - Suggested workspace but not required
  - You have access 24 hrs/day
  - TAs will hold office hours here
  - No food or beverages in Lab!
Lab Groups

- Groups of 3, posted on Moodle
- Groups randomly assigned (using code shown below)
- Practice good teamwork, learn from each other
- Report will describe contribution of each member

```
use List::Util 'shuffle';
srand(0); #random seed = 0

open FIN, "< roster_fa15.txt"; #class roster, alphabetical by last name
my @Students = <FIN>;
close FIN;

my @Groups = ("" ) x 38;

@Students = shuffle(@Students); # shuffle according to random seed

my $idx_group = 0;
while (@Students) {
  $Groups[$idx_group] .= pop(@Students);
  $idx_group = ($idx_group + 1) % @Groups;
}

open FOUT, "> groups_fa15.txt";
foreach $i (0..$#Groups) {
  print FOUT "group $i:
$Groups[$i]\n";
}

close FOUT;
exit(0);
```
Grading for Lab 3 & 4

- Demo
  - Lab assessments (on course webpage)
    - Completed by students before checkoff
    - Complete by TA at checkoff
  - Lab checkoffs with TAs
    - Schedule will be posted online, sign up for slots
    - Recorded on video, stored with report

- Report
  - Submitted on Quark by 11:55pm on due date
  - Verilog code with comments
  - More details in assignment
  - Be concise
  - No late submissions, submit whatever you have by the deadline
Timeline of Labs 3 and 4

- 10/27: Lab 3 Introduction, get lab kits from Keith
- 10/29: Review of Design with Verilog for Lab 3
- 11/12: Lab 4 assigned
- 11/16: Lab 3 Due, will post signups for checkoff
- 11/17: Lab 4 assigned (lab intro lecture)
- 12/7: Lab 4 Due, will post signups for checkoff
- 12/18: Final Exam
Lab 3 Resources

- My office hours and by appointment
- TA office hours in Duda hall
- Video demonstrations on website
- Resources on website (and elsewhere) for Verilog
- Resources on website for using Quartus and programming CPLD
Technical Description of Lab 3
Lab 3 Overview

- MIDI OX program transforms keyboard into an electronic music keyboard.
- MIDI signal is generated by the PC, goes to CPLD
  - Complex Programmable Logic Device
- The CPLD will be clocked by a 4MHz crystal oscillator, from which you may have to derive local clock for sampling.
- Output of CPLD drives 7 LEDs to display the note number.
In this lab, you will...

- Design and implement a serial MIDI receiver
- Essentially a serial port that will read a MIDI packet from PC (soundcard)
- Interpret packet content
- Display the note number in binary on seven LEDs.

- Musical notes played using PC keyboard & MIDI OX (“the world's greatest all-purpose MIDI Utility” is free at www.midiox.com). The notes played on the computer’s keyboard will cause MIDI serial data to be sent serially out the MIDI OUT connector.
In this lab, you will... (contd.)

- Program Hardware in Verilog
- Hardware is an Altera Complex Programmable Logic Device (CPLD) MAX 7000S (part number EPM7064SLC44-10)
- Using ALTERA Quartus II software tools for synthesis
- Debug - functional simulation (waveforms)
- Wire up circuit on breadboard
- Debug of board - logic analyzer
Opto-isolator Programming Header

USB Programmer (from Quartus SW)

Configuration Bitstream

MIDI Signal/Cable (from MIDI-OX SW)

MIDI signal (referenced to breadboard GND)

CPLD
Implements FSM

MIDI Signal/Cable (referenced to breadboard GND)
MIDI

- Musical Instrument Digital Interface
- Developed in 1980s
- Common hardware interface and protocol
- Allows electronic musical devices to communicate with each other
- MIDI messages are transmitted asynchronously
  - Like UART – Universal Asynchronous Receiver Transmitter
  - UART is more flexible with many more parameters
MIDI Specification

- Groups of bytes, typically three
- Each byte with START and STOP bit
- Status byte – code for Note On, Off, other ctrl, ChID – MSB is on (1)
- Data bytes – MSB is off (0)
  - 2\textsuperscript{nd} B: Note On or Off message with \textit{note number}
    - 128 different notes, 10 octaves
  - 3\textsuperscript{rd} B: Note On or Off with \textit{velocity} (how hard is instr. pressed)
Decoding of a MIDI Message

- In your work you will be decoding MIDI.
- 31,250 bits/s fixed baud rate, bit time (BT) 32us.
- With START & STOP a MIDI msg is 10BT, 320us.
  - You see this with waveform.
  - Consecutive frames separated by undefined time.

![Diagram showing the decoding process of a MIDI message with start, stop, and frame markers.]
Decoding of MIDI (contd.)

- Board has a 4MHz clock that you need to divide (or count)
- Before a frame, signal line is high
- The receive must wait for 0 and detect neg edge
- Now start sampling 8 bits (payload) in the middle
  - See below at 1.5BT, …8.5BT
  - Stop at 9.5 BT, that is the STOP bit
- Repeat for each byte
Decoding of MIDI (contd.)

- Sampled value at 9.5BT is not logic 1?
  - A MIDI receiver sets a flag “Framing Error”
  - You can implement if you want (not required)

- Sampling at least once per bit but commonly more times and vote
  - A voter has a number of inputs and generates an output based on e.g., majority or plurality of inputs
  - TMR (Triple Modulo Redundancy) voter would vote 1 if 2 out of 3 inputs are 1
    - A 5 input majority voter would need 3 out 5
ECE353: Computer Systems Lab 1

Moritz, Kundu, Holcomb

Opto-isolator Programming Header

MIDI signal (referenced to breadboard GND)

Configuration Bitstream

USB Programmer (from Quartus SW)

CPLD Implements FSM

Altera CPLD EPM7064SLC44-10

I/O Pin

Configuration Bitstream

Programming Header

LEDs

CPLD

MIDI Signal/Cable (from MIDI-OX SW)

MIDI Signal/Cable (from MIDI-OX SW)

MIDI signal (referenced to breadboard GND)
Isolation and Grounding

- Why use isolation? How do we normally send signals?
- MIDI cable from computer to breadboard
  - Do they have the same ground? How close are the ground potentials?
  - Ethernet is isolated, USB is not necessarily. Why?
- Somewhat-related interesting research: (will not be on test)
  - Don’t assume all grounds are equal
  - Laptop chassis potential (ground) varies with computation relative to earth ground
  - Can steal RSA keys by touching laptop and measuring chassis potential

https://www.tau.ac.il/~tromer/handsoff/
ECE353: Computer Systems Lab 1

Moritz, Kundu, Holcomb

Opto-isolator Programming Header

MIDI signal (referenced to breadboard GND)

CPLD Implements FSM

Configuration Bitstream

USB Programmer (from Quartus SW)

MIDI Signal/Cable (from MIDI-OX SW)

LEDs

MIDI signal (referenced to breadboard GND)
LEDs

- Outputs of CPLD drive LEDs
- Why use a resistor (array)?
- If no resistor:
  - LED is bright
  - Likely hit current limit of CPLD outputs
  - Observe voltage drop on CPLD output

![Output Drive Characteristics of 5.0-V MAX 7000 Devices](image)

![Forward Current Vs Forward Voltage](image)

![Luminous Intensity Vs. Forward Current](image)
ECE353: Computer Systems Lab 1
Moritz, Kundu, Holcomb

Opto-isolator Programming Header
MIDI signal (referenced to breadboard GND)

Configuration Bitstream
USB Programmer (from Quartus SW)

MIDI Signal/Cable (from MIDI-OX SW)

CPLD Implements FSM
Altera CPLD EPM7064SLC44-10

LED bar
MIDI signal (referenced to breadboard GND)
Programming through JTAG

- JTAG - Joint Test Action Group: IEEE 1149.1 standard entitled: **Standard Test Access Port and Boundary-Scan Architecture**
  - test access ports used for testing printed circuit boards (and chips) using boundary scan.
  - Used also for programming embedded devices.
    - Most FPGAs, PLDs are programmed via a JTAG port.

- JTAG ports commonly available in ICs
  - Boundary scan, scan chains, mbist, logic bist connected
  - Chips chained together with JTAG signals and connected to main JTAG interface on PCB
Design in Verilog

- A quick overview provided next lecture
- Focus is on methodology for proper design and coding
- Please consult course website and demos for using programmer device