ECE/CS 350 Final Project

Fall 2017

Overview

Breakdown

- Team Github Assignment: https://classroom.github.com/g/Xdm9LnkM
- Proposal
- Implementation
- Demo/Presentation
- Report
- Self and Team Evaluations

Important Deadlines

- November 14 at 11:59 PM (Duke time)—project proposal due on gradescope
- December 4 or 5, various times—project demos, project demo visual aid due
- December 14 at noon (Duke time)—project code/files due in Github repo
- December 14 at noon (Duke time)—project write-ups due as .tex files in Github repo
- December 14 at noon (Duke time)—project demo visual aid turned into gradescope
- December 14 at noon (Duke time)—self/team evaluations due

Introduction

Welcome to the last month of *Digital Systems*. In groups of two or three, you'll be working on a final project to show off your ECE/CS skills. Your project should be a non-trivial I/O application that makes substantial use of your processor. You can modify and create any hardware that you like and you will be writing software (in the form of our ISA) to make use of that hardware; otherwise, you have total freedom to do what you want.

Example Projects

While we don't want to limit you, it may be helpful to have some ideas for inspiration:

- Faceboard Messenger: a chat application
- Flappyboard: flappybird, but on an FPGA
- A piano (including custom-built keys)
- Dance Dance Revolution with custom controllers
- Remote-controlled Robot

Feel free to come up with any kind of project, as long as it makes a substantial use of your processor and has non-trivial input/output.

Grading Overview

- Proposal: this is a “show stopper.” Without it, you won’t get a grade.
- Implementation (~70%): “create-your-own” rubric
- Demo/Presentation (~15%): graded based on the quality of your presentation skills
- Report (~15%): grade based on the quality of your writing skills
- Self/Team Evaluations: also a “show stopper.” Without it, you won’t get a grade.
- Adjustments: the TAs have the discretion to adjust your final grade significantly

Project Specification

Tasks

Overview

The bulk of this project is completing a series of tasks. Based on difficulty and novelty, your tasks will be worth 0, 10, 20, or 30 points. Depending on your group size, **you are required to complete** a certain number of tasks:

- Group of 2: 70 points worth of tasks (40 for a “complete” project)
- Group of 3: 90 points worth of tasks (60 for a “complete” project)

Additionally, your tasks are subject to the constraint:

- 2/3 of your points must come from tasks that involve substantial use of your processor
- 1/3 of your points must come from tasks that involve significant input
- 1/3 of your points must come from tasks that involve significant output

Tasks can overlap in their use of the processor, input, and output.

Based on your demo and your report, you will receive a completion score and a quality score for each task. If you complete fewer than the required number of points, then we will add “ghost tasks,” each with a score of a 0 to bring you to the required number. However, if you do not meet the minimum threshold for a “complete” project, you cannot receive a passing grade for this course.

Your overall grade will be the total points earned for your tasks divided by the total number of points of tasks you complete. But, if you implement more than your required number of tasks, **we will select the best possible subset of tasks** to give you the highest score. It it thus to your advantage to propose and attempt ambitious tasks. e.g. If you complete eight 10-point tasks in a group of two, we will select the best seven to compute your final score.

At the end of your project, all of your code should be in your team Github repo:
https://classroom.github.com/g/Xdm9LnkM

IMPORTANT: task weights are subject to change at the discretion of the head TA. The weight of the tasks will be finalized by the actual tasks you perform. Therefore, a task may be “upgraded” or “downgraded.” You can ask a TA for an “expectation” of points and if you communicate your ideas clearly and in a timely fashion, your task value should not be a surprise.

Logistics

You will complete the project over the rest of the semester. Each group will be assigned a mentor TA. This TA is your main “point person” to talk to. **Additionally, you should meet with your TA *at least weekly*.** They will help advocate for task values and will form your grading panel.

If you have major changes to your project, then you should notify your TA immediately. That way, we can make sure that you still have a reasonable set of tasks and that you will complete them on time.

Logistically, you have access to the lab 24/7. Feel free to make use of any resources you need. If you require additional items to complete your project, please ask a TA.

Proposal

Overview

You should submit your proposal to gradescope. In your proposal, you should include:

- Your group
- Your overall idea
- How you plan on using inputs and outputs
- Your tasks
- A rough timeline

You should propose a series of tasks that you want to complete. You will not be required to do everything you suggest, so it is definitely preferable to suggest many different tasks. This way, you will have more freedom when you actually implement your project.

Be sure to describe:

- What the task is
- How difficult you think it will be
- How many points you think it should be worth (10, 20, or 30)
- What kind of I/O you perform (if any)
- How you use your processor

Proposal Example

For example, suppose you want to do the game Pong. Your proposal might include tasks, like:

Task 1, two-player pong

- This is the core of the game, allowing two players to move a paddle around a screen and hit a ball back-and-forth
- This is a very difficult task
- We propose it should be worth 20 points
- Input: FPGA buttons to control paddles
- Output: VGA screen to display game
- Processor use: extensively used to take in input and send output; will run main program for pong

Task 2, power-ups

- Players will be allowed to get power-ups, like getting a bigger paddle or a speeding up the game
- This is a non-trivial task, as it requires us to keep a track of what power-ups players have gotten and how long they are active for
- We propose it should be worth 10 points
- Input: no extra input
- Output: VGA screen, we have to draw power-ups on the game
- Processor use: we need the processor to keep a track of how long a power-up is active for

Task 3, amazing aesthetics

- This pong game will look amazing and include colors, explosions, and allow users to customize a game “skin”
- Pretty difficult task because xyz
- We suggest it is worth 20 points
- Input: switches to select which “skin” players want to use
- Output: VGA . . .
- Processor use: nothing significant

Task 4, four-player pong

- Have the option of adding two players to the game, for a total of four-player pong
- Pretty difficult task for such and such reason
- We think it should be 10 points
- Input: a switch
- Output: VGA . . .

- Processor use: extensive

Task 5, high score leaderboard

- Save high scores and display them from the menu on LCD display
- Pretty simple
- We want 10 points
- Input: none
- Output: LCD display
- Processor use: nothing significant

Task 6, multi-board FPGA

- Lets you connect FPGA's together to play multiplayer over the wire
- Very difficult, very novel
- We ask for 20 points
- Input: GPIO
- Output: GPIO
- Processor use: significant

Task 7, custom controllers

- Custom breadboard controllers to play the game
- Somewhat difficult and cool
- We request 10 points
- Input: controllers
- Output: LEDs on controllers
- Processor use: nothing significant

You'll notice that this group proposed a total of 100 points. In the end, they can implement whatever they want to. If they are a group of two, they have to at least complete 70 points. If they are a group of three, then then have to complete at least 90 points.

Scoring Example 1

Suppose the TAs gave all of the tasks above the value that was asked and the tasks scores came out to be:

Task 1: 18/20
Task 2: 10/10
Task 3: 15/20
Task 4: 10/10
Task 5: 10/10
Task 6: 1/10
Task 7: 4/10

A group of two would receive a score based on Tasks 1, 2, 3, 4, and 5, for a total score of 63/70.

Scoring Example 2

Suppose a group of two had the following scores:

Task 1: 30/30
Task 2: 10/10
Task 3: 20/20
Task 4: 5/20
Task 5: 5/10

The sum of all the points earned here is equal to 70. However, you are graded based on a set of tasks that add to 70 (if possible). Tasks 1, 2, 3, and 5 add to a possible max of 70, so the grade would be 65/70.

Scoring Example 3

Suppose we have the following scores in a group of three.

Task 1: 20/20
Task 2: 20/20
Task 3: 20/20
Task 4: 15/20
Task 5: 15/20

This group’s score would be a 90/100, because there is no way to calculate a score out of 90.

Demo

Your demo will be comprised of four things:

- A ten-minute presentation
- A deliverable
- A fifteen-minute Q&A session
- A five-minute recorded “show-off” session

The demo will be ~30 minutes. You will have 10 minutes of uninterrupted time to present your project to us. You must have some kind of deliverable to go along with your presentation—whether it be a powerpoint, prezi, video of your project, etc.—to facilitate your presentation. The deliverable should be such that any novice would be able to understand your entire project from this deliverable. You will submit this to gradescope, along with your final report.

Then, for up to 15 minutes, we will ask questions about your project. Finally, for five minutes, you will show off your project. We will record this demo, and post it on a private YouTube channel for you.

Your presentation will be graded based on your ability to articulate technical information in a concise and cohesive manner. Every group member must substantially participate in the presentation. Additionally, you will also be graded on how well you can answer our questions. Finally, you will also be graded on your deliverable.

We will be sending out scheduling information via email for demos as these dates approach.

Report

You will also submit a report as a group (i.e. one report per project). We expect this to be a demonstration of your technical writing skills.

Report Guidelines:

- In LaTeX (no exceptions) on Github (https://classroom.github.com/g/Xdm9LnkM)
- Kind of like a lab report but longer
- Expecting about 5 to 10 pages, single-spaced, 12-point font
- Describe design including what the project is
- Describe input and output
- Specifications
- Changes you had to make to your processor
- Challenges you faced and how you overcame them
- Any circuit diagrams, how you constructed them, and the rationale behind them (only if you have circuits obviously)
- Describe how you tested your project
- Describe your assembly programs or code
- Flowchart of assembly code (draw.io is a good tool for this)
- Pictures of your project

Self/Team Evaluations

You will be responsible for submitting self and team evaluations on Sakai. It should include:

- Reflections on what you and each team member contributed to the project
- What further improvements you would have added if you had more time
- How well you think you completed your tasks
- How much you felt that you learned through the process
- Any other reflections on the value of the project and general feedback

If need be, grades will be adjusted according to self and team evaluations. As a part of this process, we may ask to speak with you.