# Absolute Value Detector

### OVERVIEW

1.25 GHz 6-bit absolute value detector. A digital absolute value detector was designed using logic optimization and energy minimization techniques. Cadence was used to simulate layout and signal delay.

**Responsibilities**

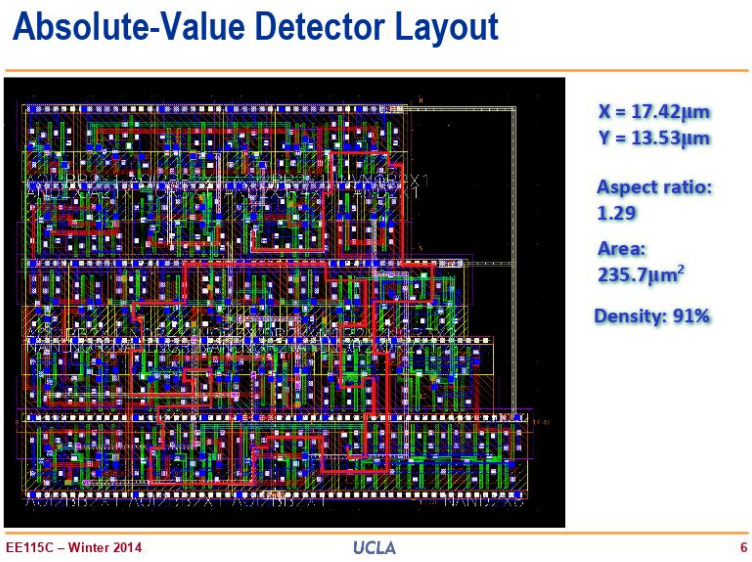
1. Reduced Logic for both the Adder/ Comparator
2. Staged design, making for easy/ smaller Layout
3. Reduced Carry gen unit for both Adder/ Comparator to streamline process
4. Routed the wires for our Circuit Design

### GOALS

1. Maintain an Aspect Ratio of at most 1.5
2. Minimize the number of gates, optimize logic, and minimize the critical length delay.

## Most Compact Design

Our circuit occupied the smallest area of the class at 235.7µm2




Complete Information including Design/ Logic Optimizations can be found at the following link: https://github.com/tomd1990/UCLA/tree/master/EE115C%20Final%20Project

# Simple Shell, a Bash like Interpreter clone

### OVERVIEW

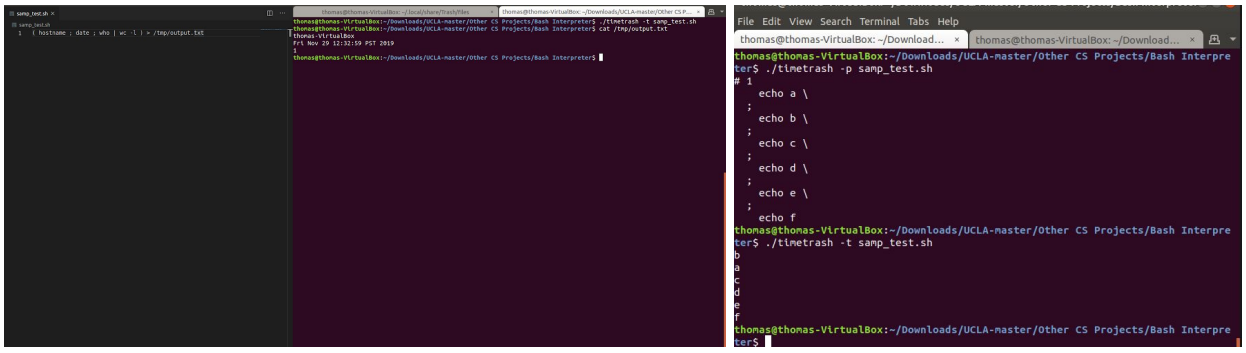
Using the C language, I created a bash like interpreter that could read .sh files and execute a subset of Bash syntax.  This included subshell commands (), pipes |, i/o redirection <>, sequences terminated by ; , comments #, and andors &&/||.  This program was able to execute compound commands, and use parallelism within the execution model if desired.

### Responsibilities

1. Understand communication between processes for i/o redirection and piping.
2. Understand the use of forking and execvp to execute system commands
3. Understanding the non parallel execution model and using stacks/ trees  to organize order of execution
4. Understanding parallelism, and using dependency lists to avoid deadlock

### GOALS

1. To demonstrate how the bash interpreter is able to function, and the tools used behind the scenes to make them work.




Complete code can be found at the following link:
https://github.com/tomd1990/UCLA/tree/master/Other%20CS%20Projects

# Imitris, A Tetris Clone

### OVERVIEW

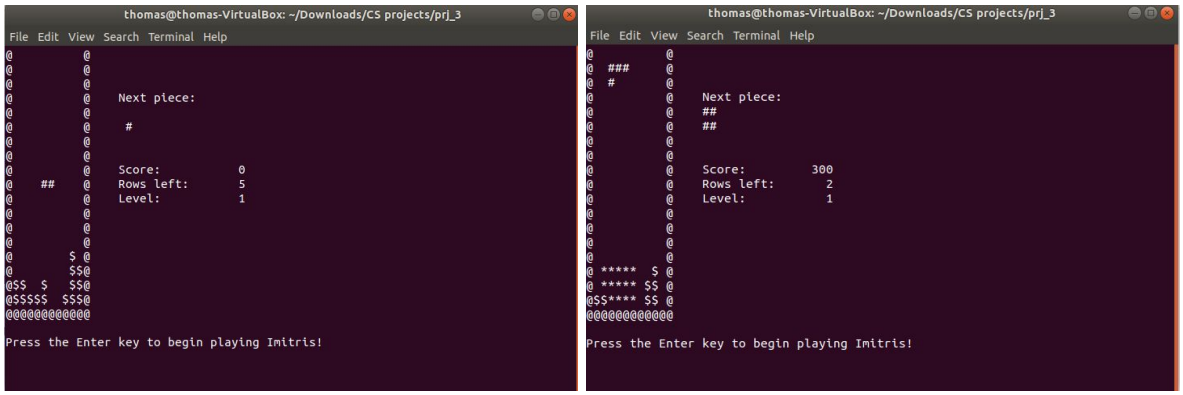
Using Object oriented programming I was able to create a Tetris clone, that featured a scoring system and three new pieces:  Foam bomb piece (which would recursively fill up empty space), Vapor piece which would destroy tiles in a radius, and inverted tile (where movement of the tile was inverted).

#### Responsibilities

5. I was responsible for creating the entire game and using object oriented programming paradigms to create this game
6. Demonstrate the use of polymorphism and inheritance for piece design, and using objects to represent both the Game and the tank/ container for the pieces that were generated.

### GOALS

2. To demonstrate Object Oriented programming, namely polymorphism, and inheritance
3. To demonstrate ability to execute a design from specification to application.




Complete code can be found at the following link:
https://github.com/tomd1990/UCLA/tree/master/CS32/Prj3_imitris