Project Proposal: Cellular Automata on FPGA

James Olsen (jco2127)
Serena Shah-Simpson (ss4354)
Jeff Tao (jat2164)
Robert Viramontes (rsv2111)
Priscilla Wang (pyw2102)

Overview
Our FPGA project will focus on the development of one or more hardware-accelerated cellular automata. A cellular automaton consists of a grid of cells, each with a finite number of states. A cell's state can change each generation depending on the states of the other cells in its neighborhood, or the grid of cells closest to the given cell. There is a multitude of algorithms that dictate the rules for how a cell should change its state given the state of its neighborhood. For example, Conway's Game of Life, which is perhaps the most famous cellular automaton, decides if a cell is "alive" or "dead" based on the number of cells that are "alive" near it.

Motivation
Cellular automata have a variety of useful applications, from traffic modelling to constructing full Turing machines. However, calculating state transitions across a large grid can become computationally intensive which can limit the usefulness or size of the problems they can be applied to. Researchers have demonstrated significant acceleration by implementing cellular automata in FPGA[1]. We look to explore not only the possible applications of cellular automata on FPGA, but the acceleration possible with FPGA implementation.

Also, they look cool. Cellular automata can also act as a visual model for understanding similar discrete models in mathematics, physics, or theoretical biology.

Input/Output
Inputs
- Keyboard input
    - UI control for automata selection
    - UI control for selecting seed or random initializations
    - UI control for start/stop/step forward

Output
- Visualization of system

Algorithm(s)
We are considering implementing one or more of the following cellular automata:

- Conway's Game of Life[2]: The livelihood of the cell is determined by its neighbors and dies from under- or over-population and brought by to life by reproduction.
- WireWorld[3]: A cell can be either background, wire, electron head, or electron tail. In the next cycle, the cell "in front" of the electron head will turn into electron head, the electron head into the electron tail, and the electron tail into wire. Background never changes.

Hardware/Software Division

Hardware components:

- State transitions in cellular automata
- Graphical display of state via VGA

Software Components:

- User interface
    - Choose which cellular automaton to display
    - Create unique bodies to place in cellular automata initial configurations
    - Start cellular (certain) cellular automata with random state or seed
    - Provide an interface for editing custom initial configurations

Milestones

Milestone 1 (March 31):

- Graphical display interfacing
- Implement algorithms in software

Milestone 2 (April 12):

- Begin translating algorithms to SystemVerilog
- Write preliminary user interface

Milestone 3 (April 26):

- Finish translating algorithms to SystemVerilog
- Complete user interface

Final Presentation (May 12):

- Add additional fun features (time permitting: music, colors, etc)
- Documentation

Resources

1. https://www.ra.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_RA/papers/wmpp2004p.pdf
2. https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life#Rules
3. http://mathworld.wolfram.com/WireWorld.html