Milestone 0: Charter

CAMLWare

Hardware in OCaml, bit by bit

Team Members

- Joseph Antonakakis, jma353
- Natasha Armbrust, nka8
- Celine Brass, cjb327
- Reuben Rappaport, rbr76

Plan for Team Meetings

We plan to meet every Saturday midday and Monday afternoon. At these meetings we will engage in scrum-esque standups, develop "Git Issues" for individuals to tackle, and plan any functionality we aim to add in the ~week time period. On Github, we will follow a "gitflow workflow," where we will have everyone working on feature branches. Pull requests will be peer-reviewed before merging in our master, production branch.

Project Proposal

Summary:

We plan to create a new functional hardware description language for digital logic with an OCaml interpreter that simulates the behavior of the circuit and an interactive GUI application that lets the user control the simulation.

Vision:

The two most popular hardware description languages (programming languages used to describe the behavior of electrical circuitry) are Verilog and VHDL. Unfortunately, both of these are imperative languages, and the imperative paradigm lends itself poorly to describing hardware. Verilog in particular runs into serious issues. The syntax is such that the vast majority of possible Verilog programs are impossible to translate into real world circuitry! To successfully design in Verilog one must first picture the circuit one wants and then attempt to construct code that will generate it - obviously this results in a design process that's exactly backwards.

Although analog circuitry is a good deal more complicated, all digital circuitry is at its core just a state machine. That is, it consists of registers which hold the state and combinational logic connecting them which serve as a transition function every time the clock ticks. This setup is far better suited to being described by a functional language than an imperative one.

For this project we plan to define the formal syntax, and semantics of a new functional hardware description language for digital logic. Then, to make it a usable digital design tool we'll write an OCaml interpreter for the language that simulates the behavior of the described circuit. Finally, we'll build a GUI application to let users interact with the simulation by visualizing their circuit, controlling the values of inputs, and stepping and running the clock.

Featureset:

For this project we plan to:

- Formally define and document a new functional hardware description language
- Write an OCaml interpreter for our new language that simulates the behavior of the defined circuit
- Build a GUI based application that allows the user to interact with the simulation
    - The GUI will allow the user to view a visualization of their circuit
    - The GUI will allow the user to step and run the clock
    - The GUI will allow the user to control the values of inputs to the circuit