Introduction

I hope to determine if using Cartesian Genetic Programming (CGP) along with Dynamic Search Space Reduction (DSSR) to evolve bitstreams for iCE40 FPGAs yields higher overall fitness and robustness than traditional evolutionary algorithms, and potentially human designed circuits. This research will be conducted under the guidance of Dr. Eduardo Izquierdo. I wish to research this topic because it builds on research I have been performing with Dr. Jason Yoder in his Evolvable Hardware research group, it interests me, and I want to contribute to it in a novel way.

Background

Evolvable Hardware (EHW) applies evolutionary algorithms to evolve iCE40 FPGA bitstreams to perform specific tasks such as outputting a specific number of pulses in a second or discriminating between high and low frequencies. In EHW experiments, the process typically follows these main steps:

1. A seed bitstream is slightly mutated to create a population of candidate bitstreams.
2. Each candidate bitstream is uploaded to an FPGA and evaluated against a target task to determine its fitness.
3. Using these fitness scores, individuals are selected for the next generation.
4. The selected individuals are mutated slightly, and the cycle repeats.

A significant drawback of EHW's approach to bitstream evolution is the massive search space present in iCE40 FPGA bitstreams. Each bitstream has about $10^{40,000}$ possible configurations. Even with this evolutionary approach, we need to drastically reduce the number of individual bits EHW's evolutionary algorithm is allowed to mutate. Consequently, this limits the complexity of evolved circuits and risks overfitting to specific FPGAs.

Moreover, EHW randomly flips individual bits in its bitstreams, which often leads to illegal configurations on the FPGAs, further hampering EHW's performance. It is not uncommon for a highly successful bitstream to have a fitness score of 0 for the next generation because an illegal configuration had evolved.

Proposed Solution

I propose to use CGP and DSSR to address these issues. With CGP, I can represent each bitstream as a graph and ensure only legal bitstream configurations can be mutated. Additionally, DSSR allows me to prune unnecessary structures dynamically without significantly limiting the complexity and robustness of evolved configurations.

Previous Research

This strategy was touched on by Julian Miller, et. al in *Principles in the Evolutionary Design of Digital Circuits—Part I*. They represented their circuits as a list of nodes, each with a basic function, such as AND, XOR, IF, etc. These nodes can be connected to others to create a circuit. Their research concluded that CGP is effective at creating novel circuits that can sometimes outperform human-designed ones. In fact, some of their evolved 3-bit binary multipliers were 20% more efficient than the best human designed ones of the time.

My research differs from Miller's in that it will focus on the FPGA bitstreams from the beginning, not abstract logic gates that can be put onto FPGAs. Furthermore, I will build on his research by applying DSSR, which will further decrease circuit size.

Methodology

I will separate my research into two main sections: experimentation and benchmarking. They are defined as follows:

In my experiment design step, I will design a graph representation of the FPGA circuits that can be easily translated back to bitstreams. I will also create an algorithm to identify unimportant sections of the graphs for removal.

After creating my experiment, in the benchmarking step, I will run initial rounds of evolution on simulated FPGAs, allowing for rapid evolution. I choose to do this because uploading and evaluating bitstreams on physical iCE40 FPGAs is highly time-consuming and will severely reduce the amount of work I am able to perform. I will have three goals for my evolved bitstreams to accomplish (separately): a tone discriminator that can discriminate between high and low frequency signals, a pulse-count circuit that outputs a specified number of pulses in 1 second, and an 8-bit multiplier circuit. The first two tasks are relatively simple and have been demonstrated in Evolvable Hardware. This allows me to compare my evolved circuits against previously evolved EHW circuits. The multiplier circuit is an example of a much more complicated design that requires digital logic to be formed. I specifically chose 8 bits because this level of complexity is likely high enough to prevent the bitstream from memorizing values instead of doing the multiplication itself.

For the tone discriminator and pulse count circuits, I will compare my circuits to those generated by EHW on the grounds of circuit complexity (number LUTs used and number of connections between the LUTs) and overall fitness. For the multiplier circuit, I will compare my circuits to those designed by humans. I will judge my circuits on the grounds of complexity (same as above) and correctness (what percentage of inputs yield a correct output).

Stages of Development

This is a significant project, so I believe it is beneficial to lay out a roadmap of milestones I plan to achieve throughout the year.

During the first milestone, I plan to learn in depth about CGP and DSSR techniques. I plan to perform independent research on the theory behind these techniques and to familiarize myself with ongoing and past research in the field. By the end of this milestone, I will be able to understand most CGP based experiments and should be comfortable designing my own CGP experiments. At this point, I will have a rough idea of the graph representation my bitstreams will take.

The second milestone will involve experiment design. I will finalize how the bitstreams will be represented in my graphs, and I will develop methods to identify unimportant vertices to my graphs. By the end of this stage, I will have a mathematical model for my graphs and functions to identify weak vertices, but I will not have any code written.

The third milestone involves writing code. I will work to translate my mathematical model into a computer model. This model will be able to represent bitstreams as graphs and convert the graphs to binary bitstream files that can be uploaded to FPGAs.

The fourth milestone implements evolution. I will use open-source tools to simulate my bitstreams and assign them fitnesses, and I will implement DSSR. Next, I will implement evolution. By the end of this stage, I will have a fully functioning experiment platform.

The fifth milestone involves benchmarking my circuits. I will run many experiments to evolve many bitstreams and score them as stated above. By the end of this stage, I will have the data needed to write my thesis.

The final milestone involves writing my thesis. I will spend the rest of the year writing and preparing my thesis for submission.

Success Criteria

I will deem my experiment successful if I am able to evolve bitstreams that are capable of at least 99% accuracy. For the pulse count circuit, this means the circuits must, on average, produce a number of pulses that is within 1% of the goal number. For the tone discriminator circuits, this means correctly identifying the tone 99% of the time. For the multiplier circuits, this means producing the correct product 99% of the time.

Furthermore, if my circuits meet or exceed circuits designed by people or EHW in the criteria discussed above, I will deem my experiment a further success.