

Interactive Fractal Viewer

CSEE W4840 Project Proposal

Luis E. P.	lep2141
Nathan Hwang	nyh2105
Stephen Pratt	sdp2128
Richard Nwaobasi	rcn2105

Description/Goals

We wish to create a device that will generate and display quadratic polynomial Julia Set fractals. Members of a quadratic polynomial Julia Set are points on the complex plane which stay bounded in a sequence of iterated function evaluations where the function f is of the form:

$$f(z) = z^2 + c$$

Where z is the complex number in question and c is some function parameter.

Typically, images of Julia sets are colorized according to how long it takes for each point on the plane to become unbounded (pass some distance threshold). We would like to perform this same colorization.

Note that there are a few parameters that define the device's behavior. These include:

- The constant c that uniquely describes the Julia Set generated. (complex number)
- The horizontal range of the display window. (range in the reals)
- The vertical range of the display window. (range in the reals)
- The colorization function k . (function mapping positive integers to an RGB color value)

We are certain that we would like the user to be able to select the constant c from a number of pre-selected constants using the builtin switches on the Altera DE2 board. Window settings might be handled similarly, by way of the push buttons on the board, using a peripheral such as a keyboard, or not at all. Our ambition is to cycle through colorization functions by way of audio-in processing, but we recognize that this feature will be both time consuming and auxiliary.

Mission-Critical Modules

- Iterated function module (IFM) takes complex number as input, applies function iteratively, and then produces an output describing (a) if the function stayed bounded or (b) for how many iterations it stayed bounded if it became unbounded

- Generator module with two jobs:
 - Keep track of which pixels still need to be generated. Associate those pixels with a complex number determined by our data window and display resolution
 - Ensure that each pixel gets sent off to an IFM for processing. Two caveats here:
 - Multiple IFMs can and should work in parallel
 - The processing time for each pixel varies
- Intermediate display buffer (ID Buffer) maps each pixel passed along by the Generator module to the output of the IFM for that pixel.
- Colorization module takes each value from the ID Buffer, applies the colorization function k to the value, and stores the resulting RGB value $k(n)$ at a point in a frame buffer corresponding to the point in the ID buffer from whence the value came.
- Frame buffer stores RGB values for each pixel.
- VGA out module reads from the frame buffer and produces a VGA output signal.

Parametrization Modules

Time permitting their implementation, these modules can modulate the output in interesting ways.

- The Seed Module maps switch configurations to a Julia Set constant c . The module must communicate this value to each IFM, and ask the generator module for a remapping.
- Display selection module determines the x range and y range of the display window. The module must communicate these ranges to the generator module and ask for a remapping.
- K-selector module cycles through colorization functions and reports the function to the colorization module.
- Spectral analysis module takes audio input and uses it to effect the performance of the k-selector module.

Implementation Ideas of Mission-Critical Modules

- IFM: floating point multiplication, addition, and comparison loop
- Generator Module: State, integer addition, interface with IFM
- ID Buffer: Encoder to turn Pixels to Addresses, RAM
- Colorization Module: Fancy decoder
- Frame Buffer: RAM
- VGA Out: VGA module a la Lab 3

Milestones

Milestone 0 (ASAP):

Be able to add and multiply floating point numbers.

Milestone 1 (Mar 27):

Have a static Julia set filled into the ID buffer.

Milestone 2 (Apr 10):

Display the colorized Julia set through VGA.

Milestone 3 (Apr 24):

Implement parameter changing, with subsequent updates to the displayed Julia set.