# ENGR 3410: Miniproject 3

due October 27, 2025

In this miniproject, you will use the OSS CAD suite to design a digital circuit that plays a colorful variant of Conway's Game of Life on an $8 \times 8$ grid with cyclic boundary conditions and displays the succession of board states on the supplied $8 \times 8$ WS2812B RGB LED matrix.

This miniproject is an *individual* one. You can discuss design approaches and help each other with learning SystemVerilog and how to use the OSS CAD suite, but each of you must complete all aspects of this assignment.

**Background.** The Game of Life is a *cellular automaton* (CA) devised by the British mathematician John Conway in 1970 as an exploration of how surprising patterns and complexity can arise from a set of deceptively simple rules. The game is played on an infinite two-dimensional rectangular grid of *cells*, which can be in one of two states: *living* or *dead*. From a given initial state, the *seed*, in each successive generation, each cell in the grid interacts with its eight nearest neighbors according to the following rules:

1. Any living cell with fewer than two living neighbors dies.
2. Any living cell with two or three living neighbors continues living.
3. Any living cell with more than three living neighbors dies.
4. Any dead cell with exactly three living neighbors becomes a living cell.

From these simple rules, a surprising amount of complexity arises, including stable patterns of cells, oscillating patterns of cells, patterns of cells that propagate across the grid, and even patterns of cells that can give rise to the propagating patterns. Conway's Game of Life has been shown to be *Turing complete*, which means that it is a universal computational structure that is capable of computing anything that is computable!

In this miniproject, you will design a digital circuit that plays a colorful variant of Conway's Game of Life on an $8 \times 8$ grid with cyclic boundary conditions (i.e., the boundaries wrap around to the other side of the grid) to better approximate an infinite grid with no boundaries. **Your cirucit must play at least one Game of Life on the grid and display the succession of grid states on the provided $8 \times 8$ RGB LED matrix in a visually interesting way.** Here are some Game of Life variants for you to consider:

1. Play a single Game of Life, but rather than show the living cells in a single unchanging color, make the color of living cells cycle around the hue color wheel.
2. Play three independent Games of Life on the grid displaying the state of the three games on the red, green, and blue LEDs, respectively. When a given cell is alive simultaneously in more than one game, other colors will be displayed.

3. Play twenty-four independent Games of Life on the grid, one for each of the 24 bits in the three 8-bit RGB values in the RGB LED color space. This is similar to the last variant, but on steroids.

4. Play a single Game of Life and think of the RGB color of a living cell as its genetic information (a dead cell would be black). When a new cell is born, its color components are derived from the three parent cells using some kind of genetic recombination rule.

5. Play three Games of Life on the same grid and think of the red, green, and blue cells as three different species that interact with each other according to some kind of predator/prey relationships, maybe along the lines of the game Rock, Paper, Scissors.

6. Play three independent Games of Life on the same grid, maintaining an 8-bit color value associated with each channel, which is updated by adding a relatively large value if the cell is living or subtracting a small value if the cell is dead, limiting at 0 and 255. When cells die in a given color plane, they fade to black gradually rather than disappearing abruptly. See `https://www.algoritmarte.com/conways-game-of-life-variations` for more information on this variant.

You are also free to come up with a variant of your own, if you would like.

**Requirements.** Your design must meet the following requirements:

1. Your circuit must play at least one Game of Life on an $8 \times 8$ rectangular grid with cyclic boundary conditions. The state of your grid in each successive generation must be displayed in a visually interesting way on the provided $8 \times 8$ WS2812B RGB LED matrix at a reasonable frame rate (i.e., one that humans can see).

2. The initial state of the grid in your circuit should be specified using one or more $memh files.

3. Your circuit must be specified in one or more SystemVerilog source files.

4. You must provide a SystemVerilog test bench and simulation results using Icarus Verilog (iverilog) demonstrating that your circuit correclty implements the succession rules of Conway's Game of Life.

**Deliverables.** By the start of class on October 27, you must submit the following items to the course Canvas site:

1. A PDF file containing a brief report explaining your Game of Life variant, the design of your circuit, and its operation. In your report, you should include simulation results demonstrating that your circuit correctly implements the succession rules of Conway's Game of Life.

2. A demo video showing your circuit playing your Game of Life variant on your iceBlinkPico board and the provided $8 \times 8$ RGB LED matrix.

3. Copies of all of the source files specifying your circuit as well as your test bench. You may provide the URL of a Github repo or a shared folder containing your source files.