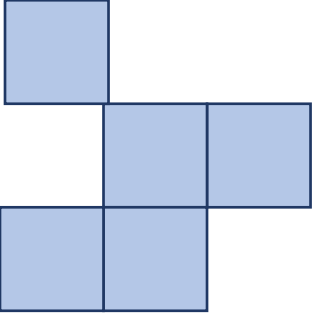
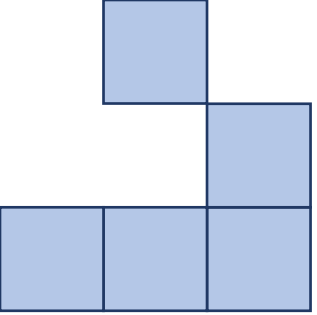
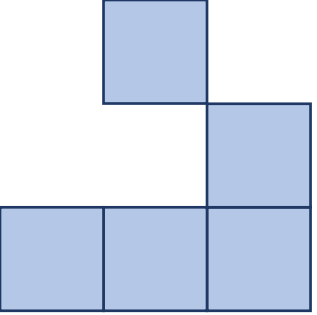
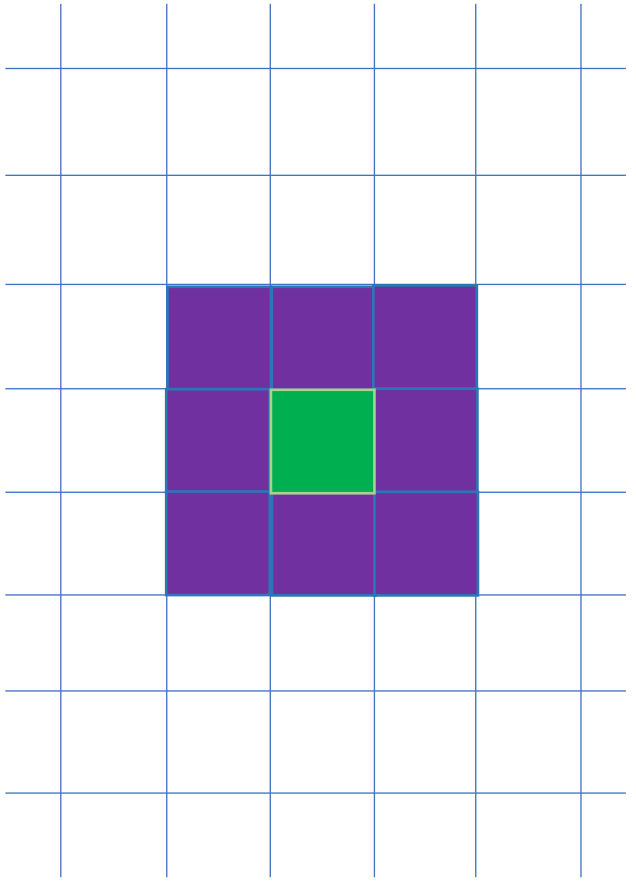


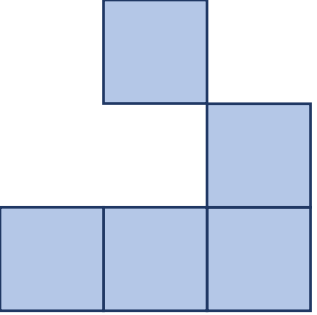
Conway's Game of Life on FPGA



What is Conway's Game of Life?

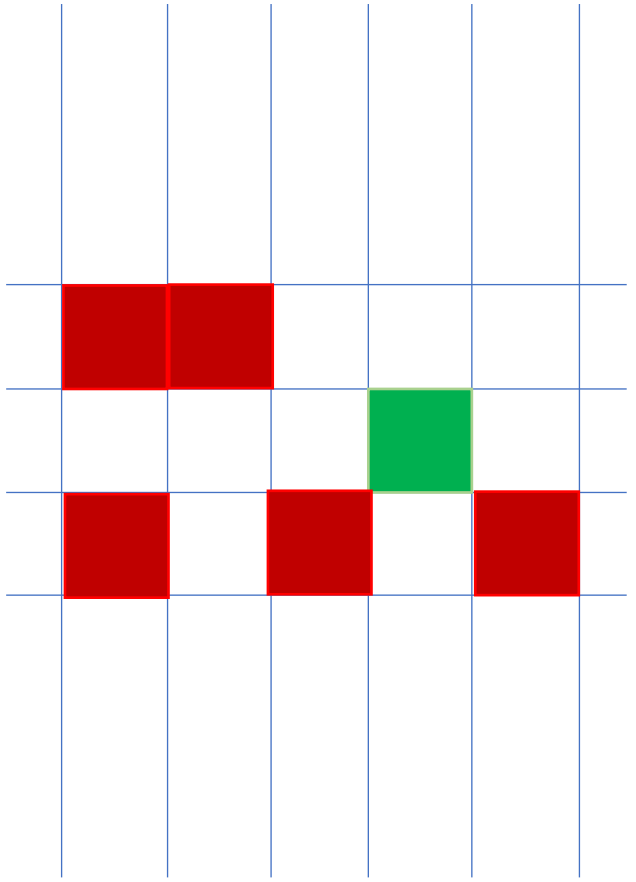


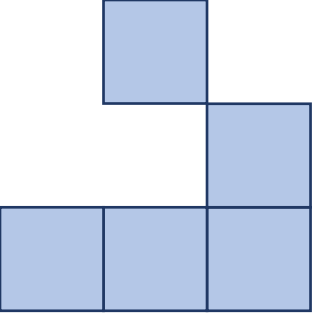




Underpopulation

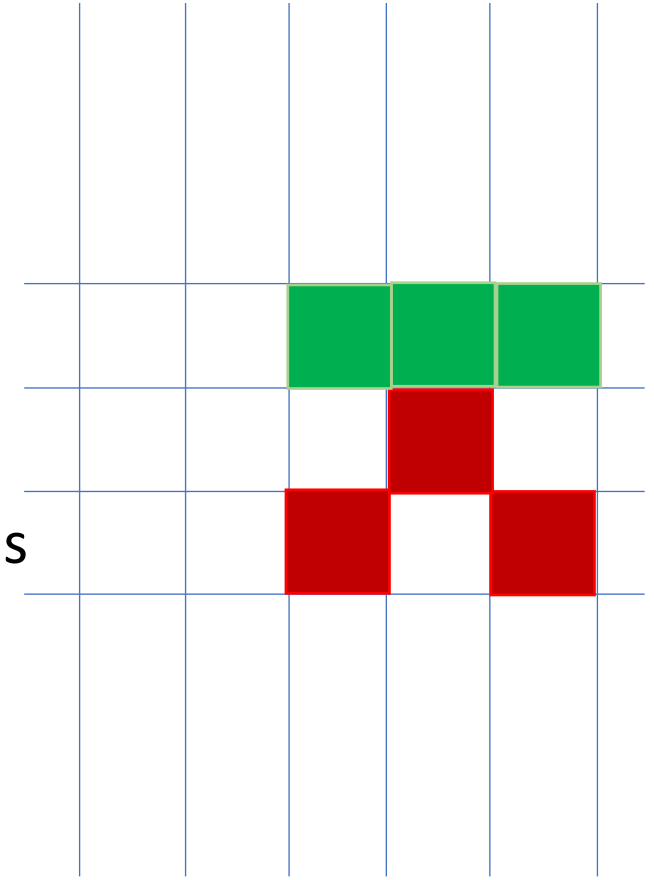
Any live cell with fewer than two live neighbors dies





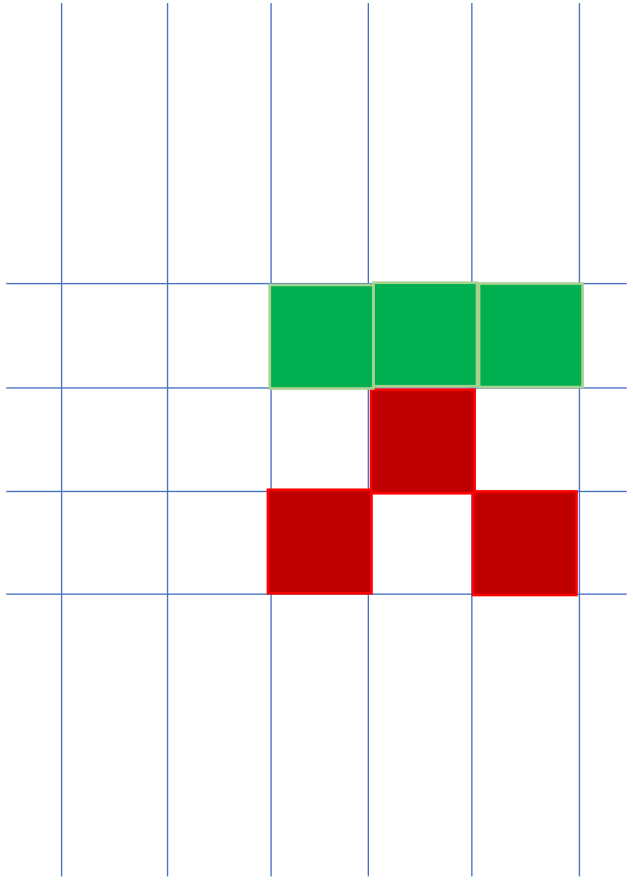
Overcrowding

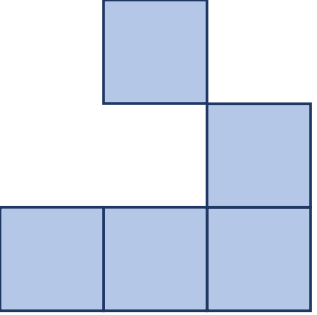
Any live cell with more than three live neighbors dies



Survival

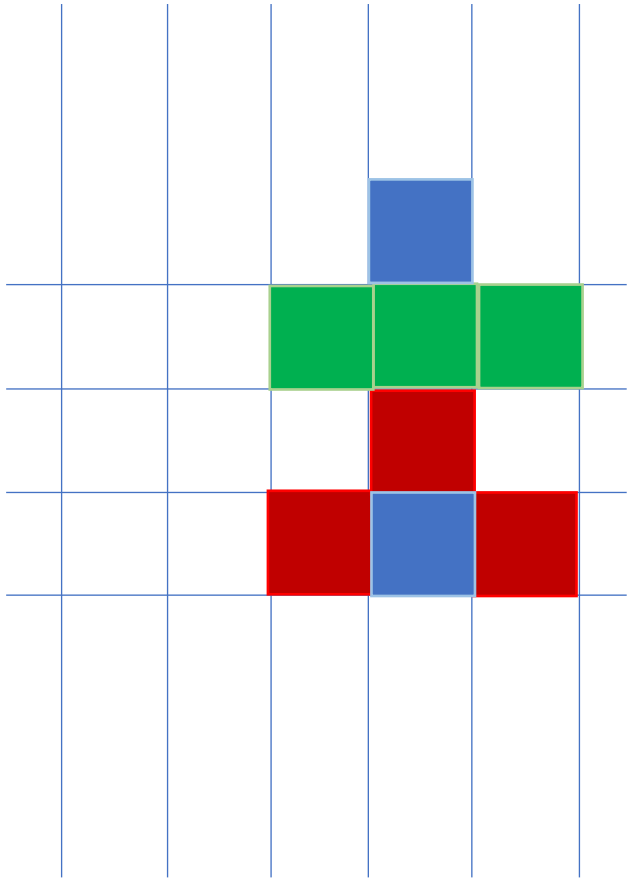
Any live cell with two or three live neighbors lives on





Birth

Any dead cell with three live neighbors will be born



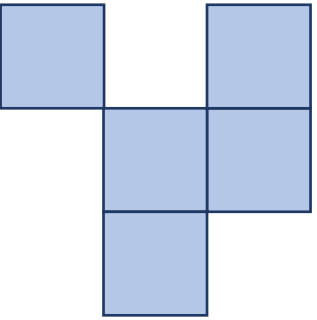
Object examples:

Stable:

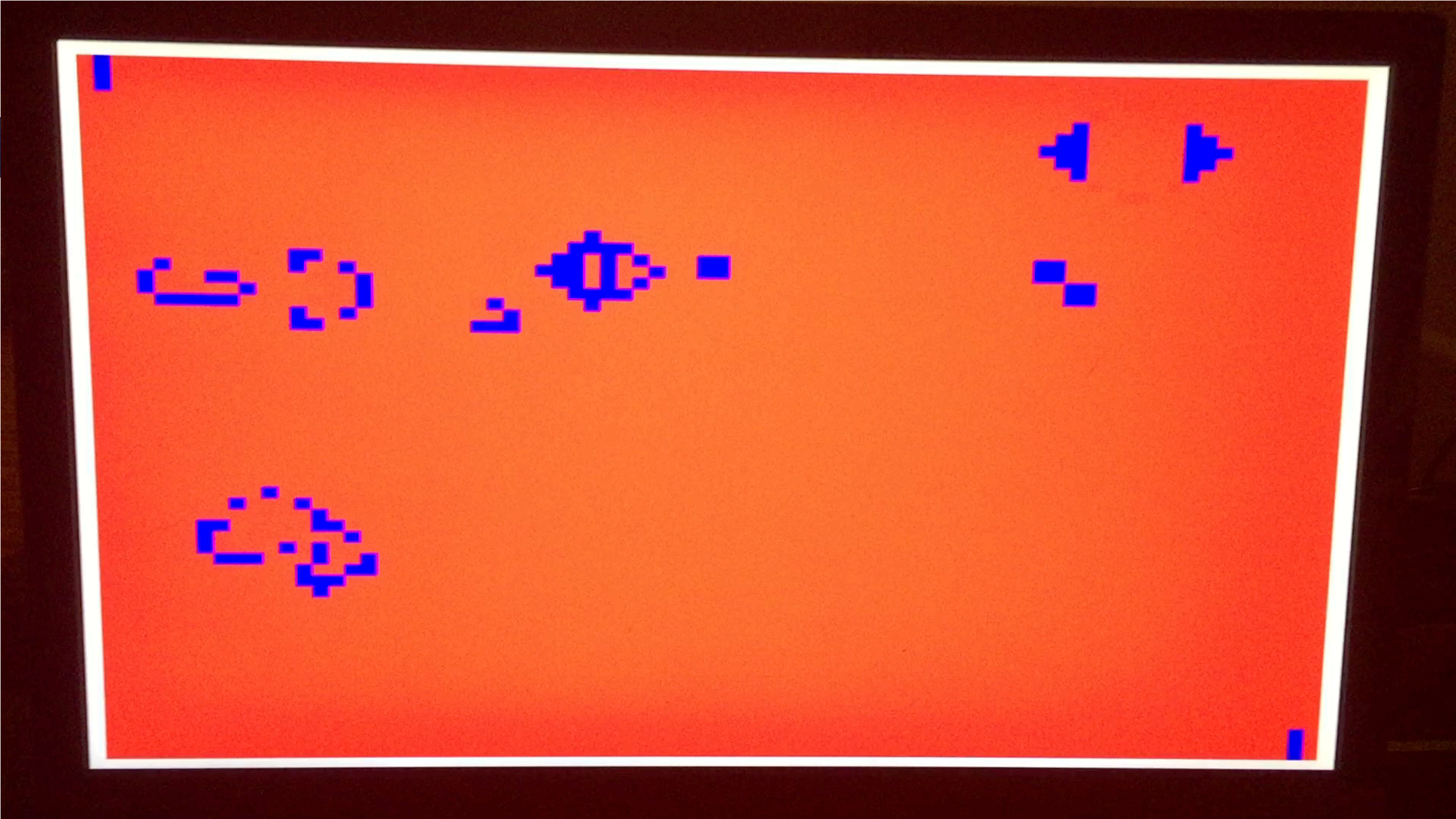
Oscillating:

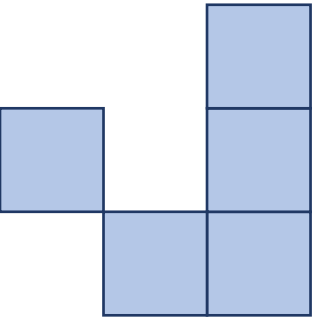
Glider:

Special cases (f-Pentomino):

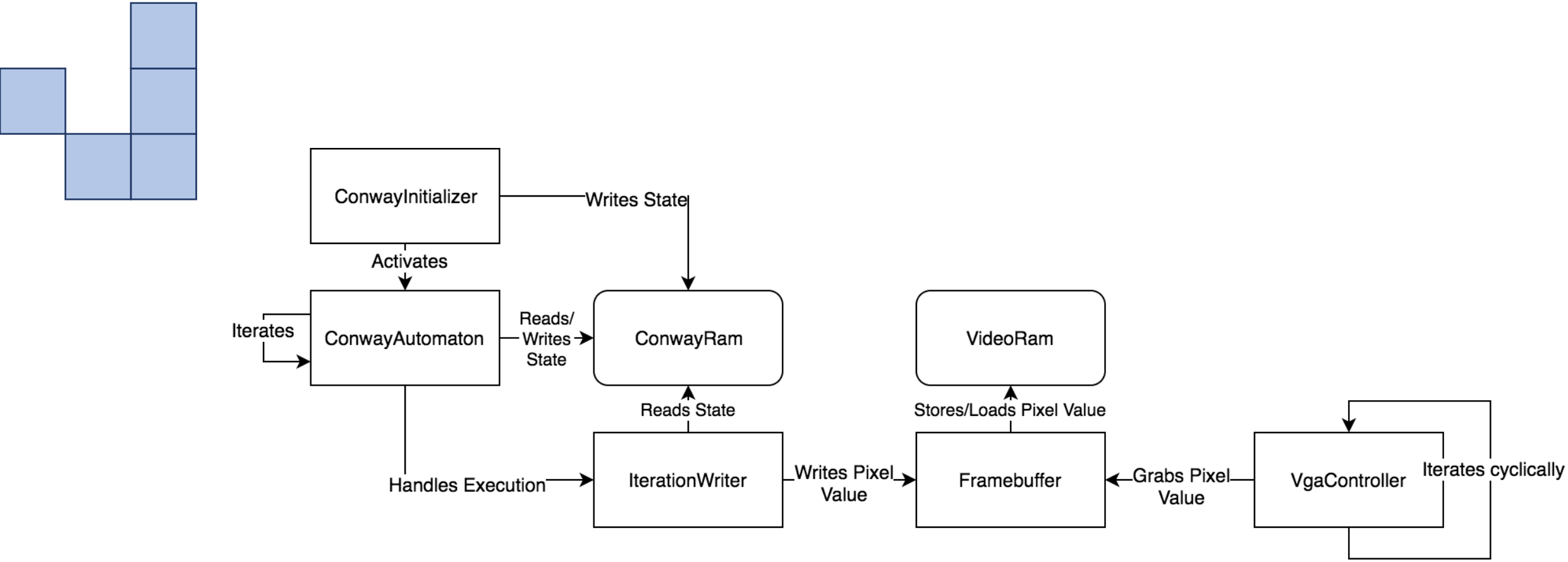


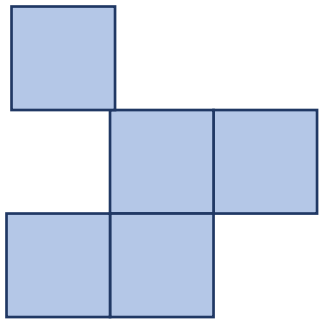
Live Demo



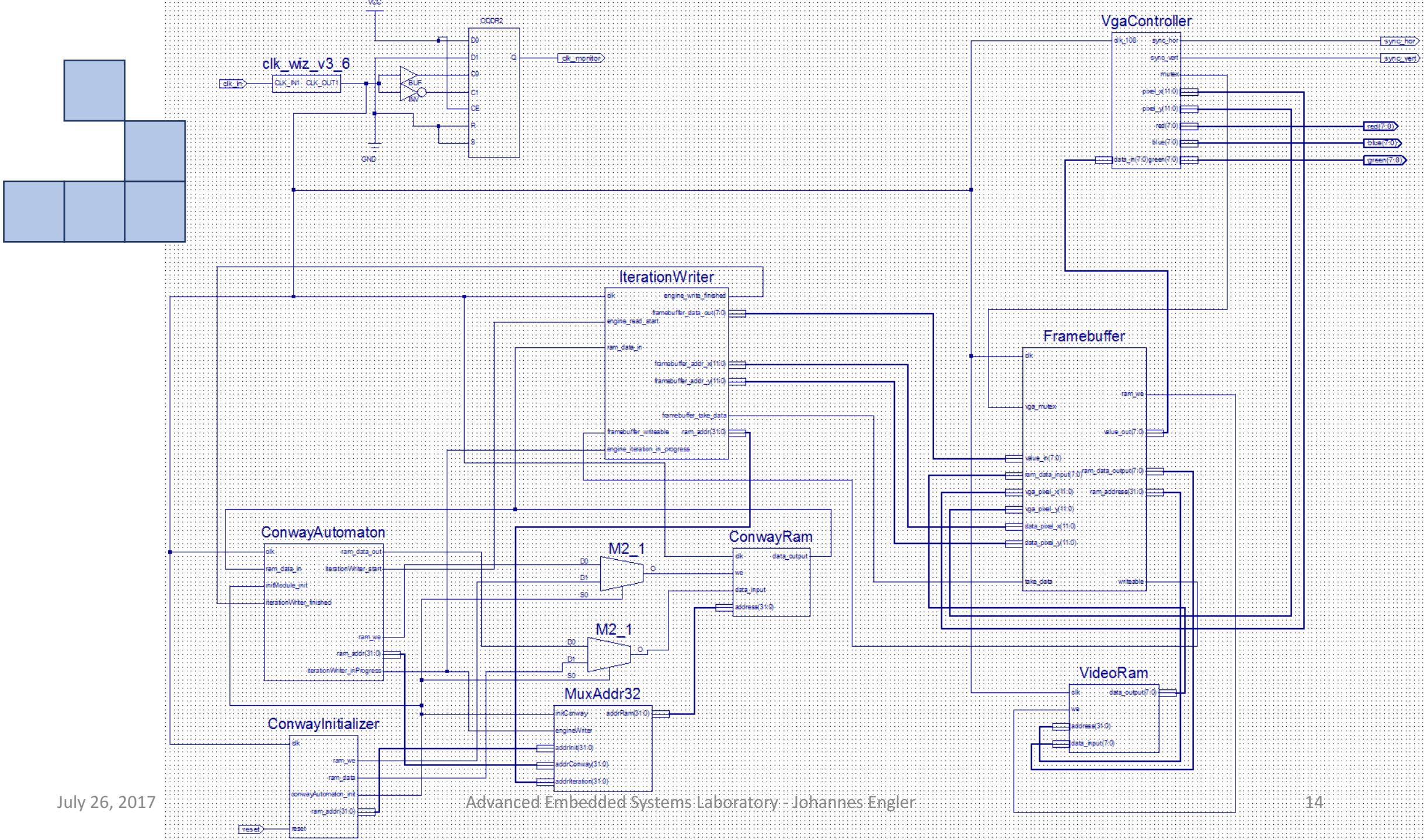


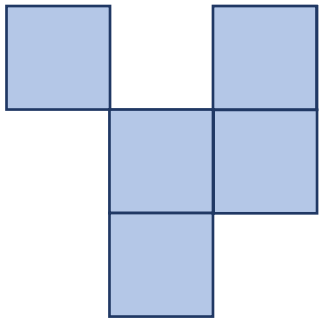
Process


ConwayInitializer
Writes State
Activates
Iterates
ConwayAutomaton
Reads/
Writes
State
ConwayRam
VideoRam
Reads State
Stores/Loads Pixel Value
Handles Execution
IterationWriter
Writes Pixel
Value
Framebuffer
Grabs Pixel
Value
VgaController
Iterates cyclically



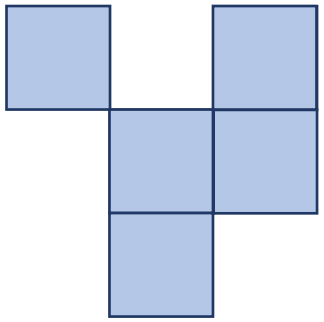
Implementation


clk_wiz_v3_6
clk_in
CLK_IN1
CLK_OUT1
BUF
INV
GND
ODDR2
D0
D1
C0
C1
CE
R
S
Q
clk_monitor
VgaController
clk_108
sync_hor
sync_vert
mutex
pixel_x(11:0)
pixel_y(11:0)
red(7:0)
blue(7:0)
data_in(7:0)
green(7:0)
IterationWriter
clk
engine_write_finished
framebuffer_data_out(7:0)
engine_read_start
ram_data_in
framebuffer_addr_x(11:0)
framebuffer_addr_y(11:0)
framebuffer_take_data
framebuffer_writeable
ram_addr(31:0)
engine_iteration_in_progress
Framebuffer
ram_we
vga_mutex
value_out(7:0)
value_in(7:0)
ram_data_input(7:0)
ram_data_output(7:0)
vga_pixel_x(11:0)
ram_address(31:0)
vga_pixel_y(11:0)
data_pixel_x(11:0)
data_pixel_y(11:0)
take_data
writeable
ConwayAutomaton
ram_data_out
iterationWriter_start
initModule_init
iterationWriter_finished
ram_we
ram_addr(31:0)
iterationWriter_inProgress
M2_1
D0
D1
S0
O
ConwayRam
data_output
we
data_input
address(31:0)
M2_1
MuxAddr32
initConway
addrRam(31:0)
engineWriter
addrInit(31:0)
addrConway(31:0)
addrIteration(31:0)
VideoRam
data_output(7:0)
ConwayInitializer
ram_data
conwayAutomaton_init
reset

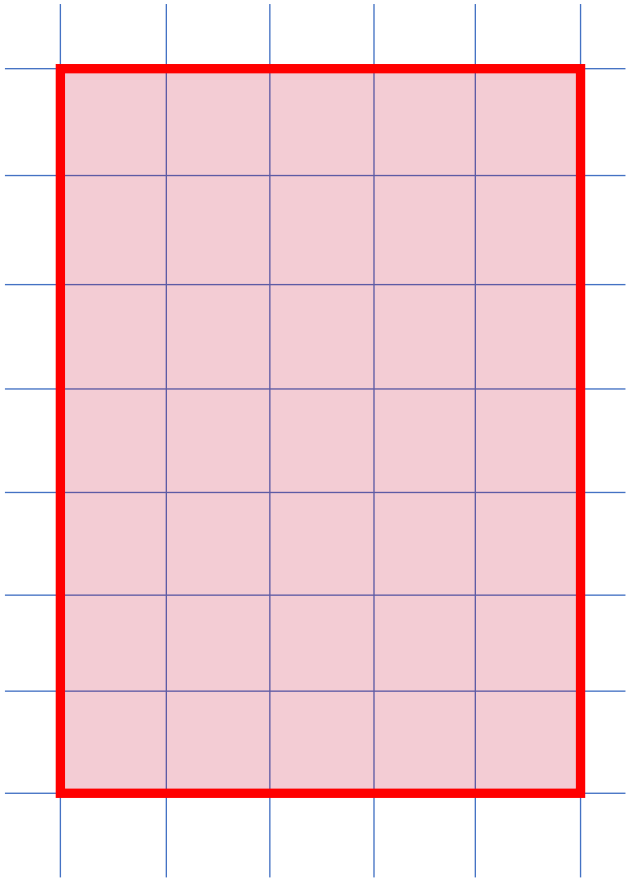


Design stupidity?

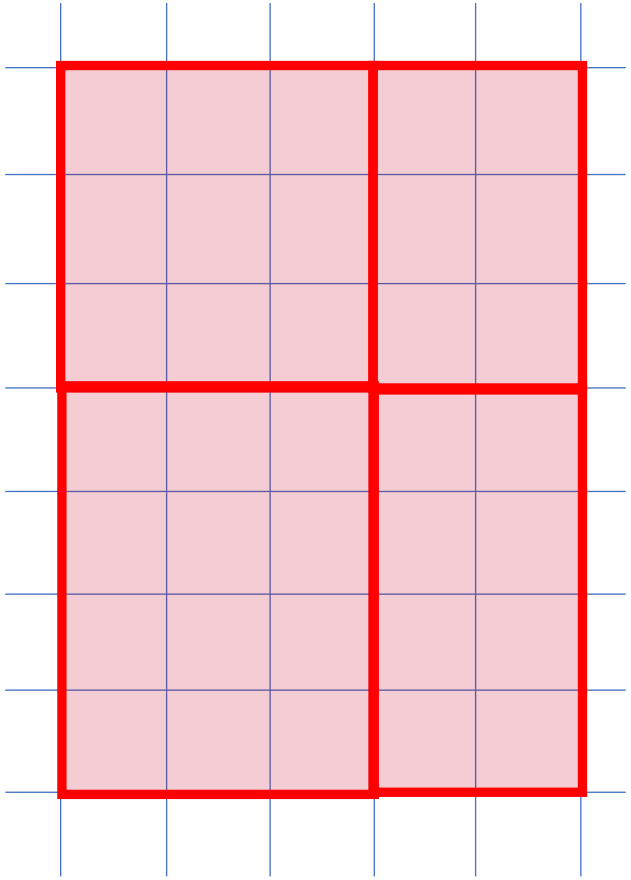
How the automaton could be even faster than it is right now?

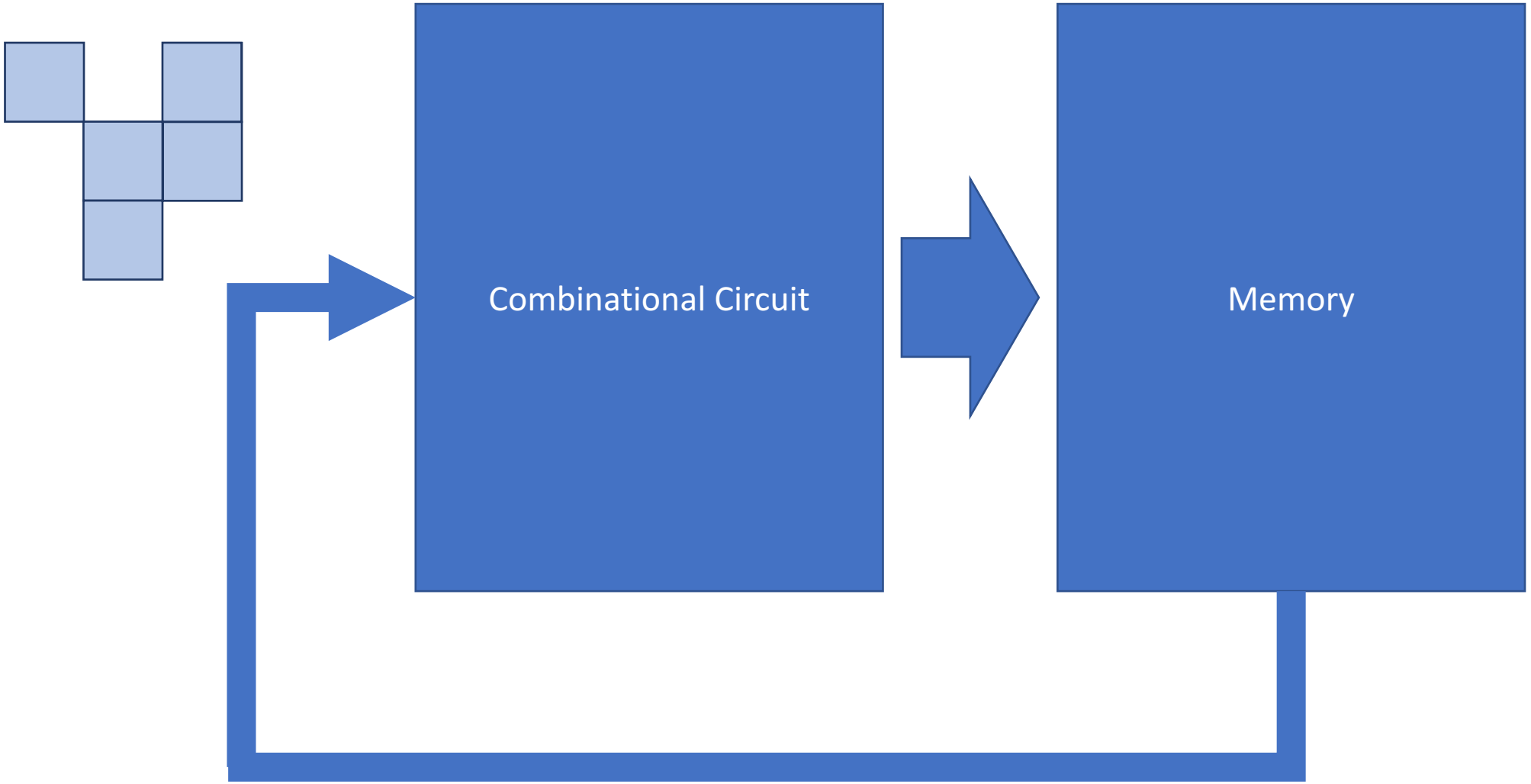


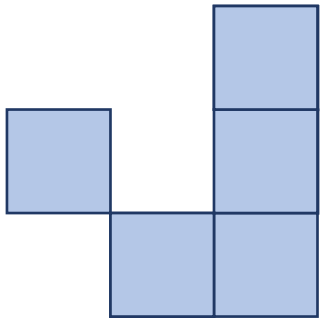
Current



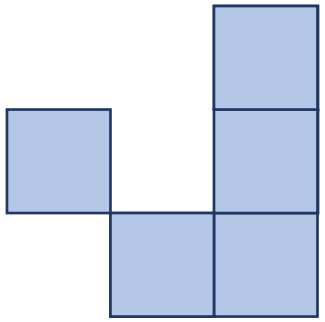
Possible

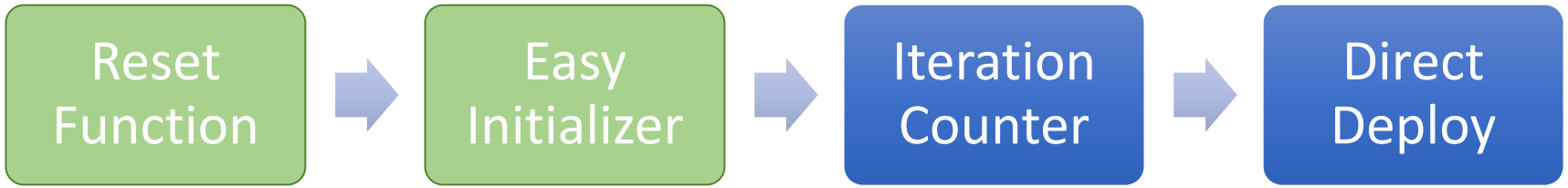


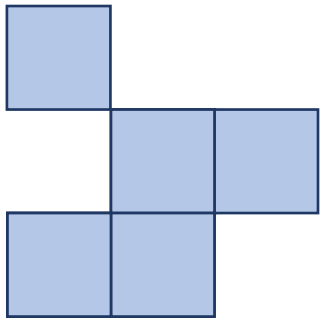

Combinational Circuit
Memory



Possible Extensions




Reset Function
Easy Initializer
Iteration Counter
Direct Deploy



Thank you!

Feel free to checkout the project on GitHub:

https://github.com/johannes5117/ConwayVHDL-Spartan6

Images:

- Images on slide 8 are from Wikipedia:
  - https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens
- The other images were self-painted