Mylonakis Emmanouil
Electrical & Computer Engineering School
Technical University of Crete

Development of a CAD tool and Hardware Design in Order to Execute Cellular Automata on a Reconfigurable Platform by non-FPGA-Conversant Users

Table of Contents

1. Introduction
2. Theoretical Background
3. Related Work
4. The Baseline Hardware Architecture
5. The Re-programmable Framework
6. The CAD Tool To Drive the FPGA-based Accelerator
7. The Graphical User Interface CAD Tool to Describe the CA Model
8. System Verification, Examples of Use, Evaluation and Results
9. Conclusions and Future Work

What are Cellular Automata (CA) ?

CA are mathematical structures, characterized by the properties of:

- ✓ Discreteness
- ✓ Determinism
- ✓ Turing-Completeness
- ✓ Locality
- ✓ Uniformity

Applications to almost every scientific field:

- ✓ Mathematics
- ✓ Physics
- ✓ Biology
- ✓ Bio-medicine
- ✓ Chemistry
- ✓ Urban Planning
- ✓ ...

Thesis Contribution

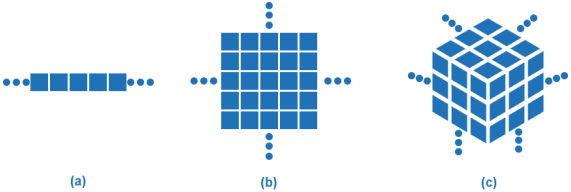
Next Step of Kyparissas' work

- + A Parallel FPGA-based Architecture
- + Customizable Framework
- + General-Purpose CA simulator
- + Large Neighborhood Windows
- + Highly Optimized @120 FPS
- - Manual Initialization
- - Manual Reconfiguration via VHDL
- - Time-intensive placement and routing

The present Thesis contribution:

- + Re-programmable Framework
- + CAD tool
- + Automated Initialization
- + Cellular Automata Description Language (CDL)
- + Graphic User Interface (GUI) environment

Cellular Automata: Space



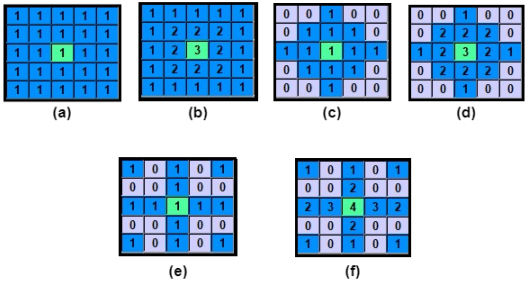
(a)-1D, (b)-2D, (c)-3D grid depiction.

- ✓ Theoretically infinite lattice
- ✓ May be multi-dimensional

Notwithstanding ...

- ✓ Finite size in practise
- ✓ Focus on 2-D grid

Cellular Automata: Neighborhood Types



(a), (b) Non-weighted and Weighted Moore,
(c), (d) Non-weighted and Weighted von Neumann,
(e), (f) Non-weighted and Weighted Custom.

Let,

$N^{M}_{(x_0,y_0)} \rightarrow$ Moore set

$N^{V}_{(x_0,y_0)} \rightarrow$ von Neumann set

Then,

- $N^{M}_{(x_0,y_0)} = \{(x,y) : |x - x_0| \leq r, |y - y_0| \leq r\}$
- $N^{V}_{(x_0,y_0)} = \{(x,y) : |x - x_0| + |y - y_0| \leq r\}$

where, $(x,y) \in Z$, (x_0, y_0) central cell, r radius

Cellular Automata: Calculation Procedure & Classes

1. $\forall$ cells, repeat steps (2-5) indefinitely
2. Determine the neighborhood
3. Multiply with weights
4. Compute total sum
5. Apply Transition Rule

Total Sum

$$S = \sum_{x=-r}^{r} \sum_{y=-r}^{r} w(x,y) * c_t(x,y)$$

Totalistic Class

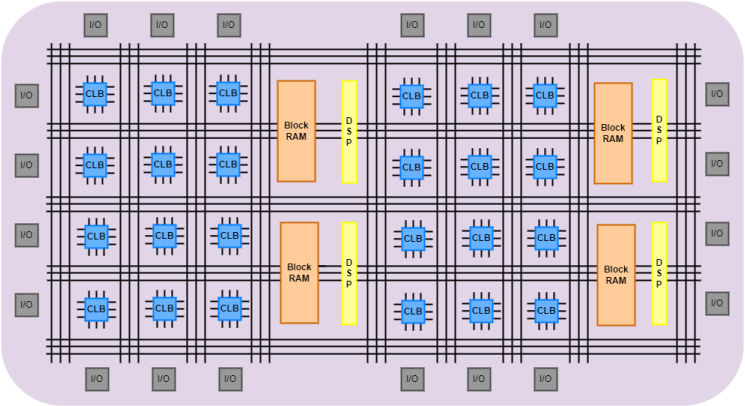
$$c_{t+1}(x,y) = \begin{cases} s_0 & , \text{ if } a_0 \leq S < b_0 \\ s_1 & , \text{ if } a_1 \leq S < b_1 \\ \vdots & \\ s_n & , \text{ if } a_n \leq S \leq b_n \end{cases}$$

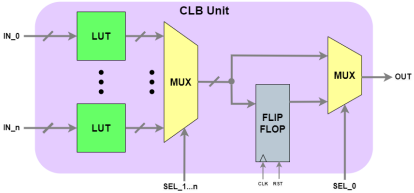
Outer-Totalistic Class

$$c_{t+1}(x,y) = \begin{cases} s_0 & , \text{ if } a_0 \leq S < b_0 \ \& \ c_t(x_0,y_0) \in [t_0,t_1) \\ s_1 & , \text{ if } a_1 \leq S < b_1 \ \& \ c_t(x_0,y_0) \in [t_1,t_2) \\ \vdots & \\ s_n & , \text{ if } a_n \leq S \leq b_n \ \& \ c_t(x_0,y_0) \in [t_n,t_{n+1}] \end{cases}$$

FPGA Technology

Basic Hardware Architecture of FPGA



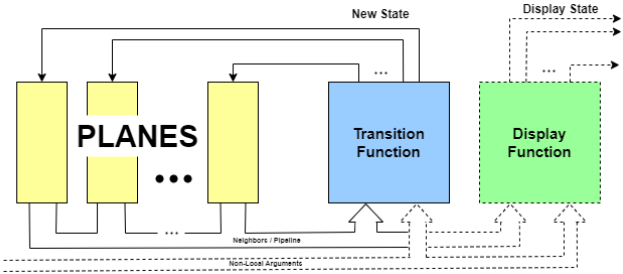


Performance comes from:

- ✓ Parallelism
- ✓ Pipelining
- ✓ Internal Bandwidth

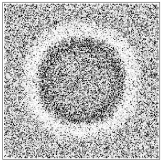
Tomasso Toffoli's and Normam Margolus' CAM Architecture (1984~2000)

CAM's hardware architecture



CAM's programming environment alongside a sample

```
new-experiment          512 by 512 space

                        0 0 == north  1 1 == south
                        2 2 == east   3 3 == west

define-rule hpp-rule    north south =  east west =  and
                        if east <-> north  west <-> south then

end-rule

define-step hpp-step    lut-data    hpp-rule
                        site-src    lut
                        lut-src     site
                        kick        north field -1 y
                                    south field  1 y
                                    east  field  1 x
                                    west  field -1 x
                        run         new-table

end-step
```



First version:

- Based on TTL, non-FPGA
- Pipelined Architecture
- 256×256 Lattice
- 1-bit depth per Plane, 8 Planes top
- SRAM LUT, 4×1024 bits

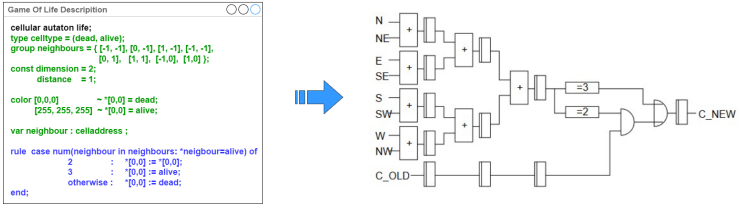
Later Versions of CAM:

- Arbitrary Large Grid/Dimensions/Cell States
- Arbitrary Large Neighborhood Windows
- Multiple Grid Types

Configuring CAM box:

- Jumpers and pins on backplane
- Plugging card (Planes, LUTs, etc)
- Low-Level Language

CEPRA: **Ce**llular **Pr**ocessing **A**rchitecture (1994~2000)



CEPRA's-1X environment alongside the hardware.

CEPRA-8L version:

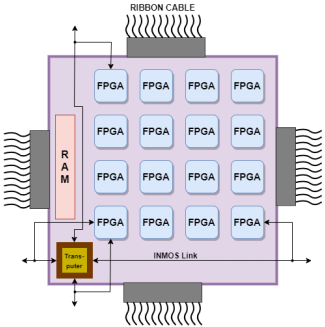
- Cluster of 8 FPGAs
- Simultaneous update of all neighbors
- 3x3 Neighborhood
- $512 \times 512 \times 8$ Lattice
- 22 FPS in real-time

CEPRA-1X version:

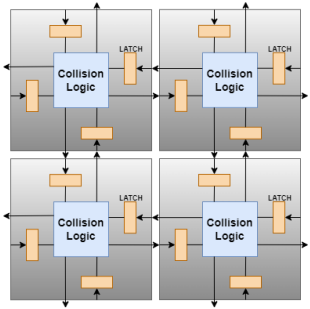
+ 2D or 3D Grid
+ $1024 \times 1024 \times 16$ Lattice
+ 50x Speedup over a CPU
+ CDL

SPACE: **S**calable **P**arallel **A**rchitecture for **C**oncurency **E**xperiments (1996)

One SPACE board



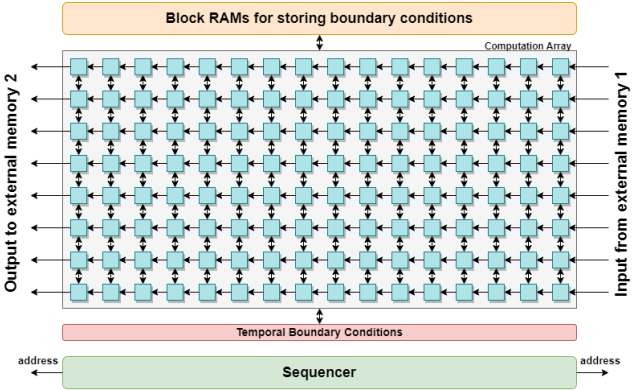
FPGA's Architecture



- 16 FPGAs per board
- Targeted to HPP models
- 9×30 Lattice per board
- Linear Scalability via Ribbon Cables
- One cell per PE
- Horizontal/Vertical Cell Shifts
- 10x Speedup over CAM-8
- Driven under a C program
  - ☞ Manipulate Hardware
  - ☞ Derive Netlists
  - ☞ Optimized place and route

Kobori, Maruyama and Hoshino (2001)

Overview of hardware architecture. 8×16 ($k \times n$) PEs.



- Fully-Pipelined, FPGA-based
- Time step: $+n$
- Distributed RAM for boundary conditions
- 2048×1024 FHP Lattice
- 400 generations per second/pseudo-real
- $155\times$ Speedup over Pentium-III @700MHz
- Simplified extension of C code

Phepls' and Islam's Cellular A Automata Compiler (ACAC) (2023)

Approach:

- Optimized Framework
- For Clusters of CPUs/GPUs
- CUDA/HIP/OpenMP + MPI
- Parallelization Strategies
  - ☞ Shared and Distributed Memory
  - ☞ Packing up cell states
  - ☞ Double Buffering

Specifications:

- 29×29 neighborhoods
- 8-bit cell states
- 2D or 3D grids
- Up to 20480×11520

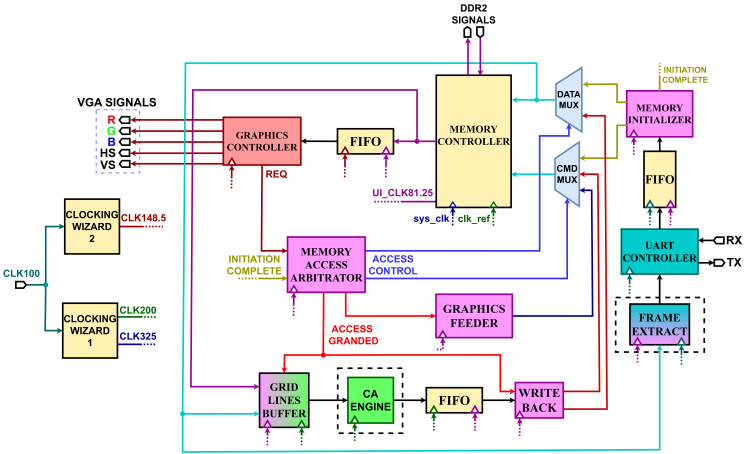
Tested on:

- Stampede2 cluster
  - ☞ 68 Intel Xeon Phi 7250 @1.4GHz
- Maverick2 cluster
  - ☞ 68 Intel Xeon E5-2629 @2.1GHz
  - ☞ 4 NVIDIA 1080-TI
- Penguin On-Demand cluster
  - ☞ 48 AMD EPYC-7001
  - ☞ 8 AMD Radeon M150

Comparison:

- − $16\times/12\times$ Speedup on AP/GH
- + Medium-sized FPGA
- + Limited models
- + Performance drops $\propto$ radius

Top Level View of Kyparissas' Architecture



- 1920x1080 FHD grid
- 29×29 Neighborhood windows
- 4-bit or 8-bit cell states.
- Real-time Display @60FPS
- Grid Types
  - a Rectangular
  - b Cylindrical
  - c Toroidal

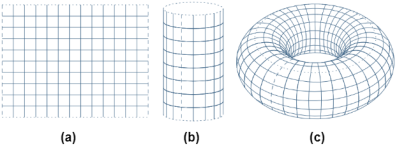
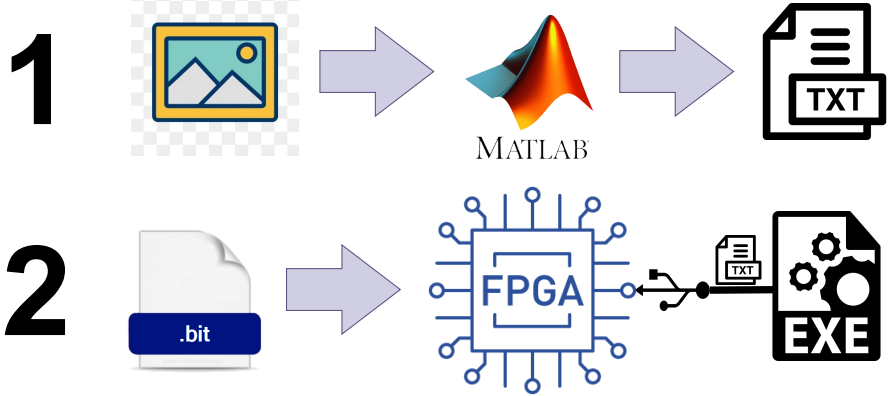


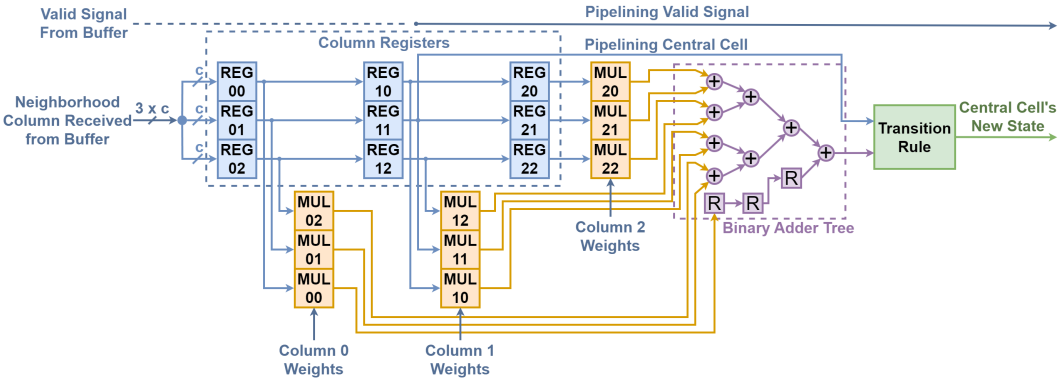
Figure Designed by Kyparissas

Initialization of Kyparissas' Machine

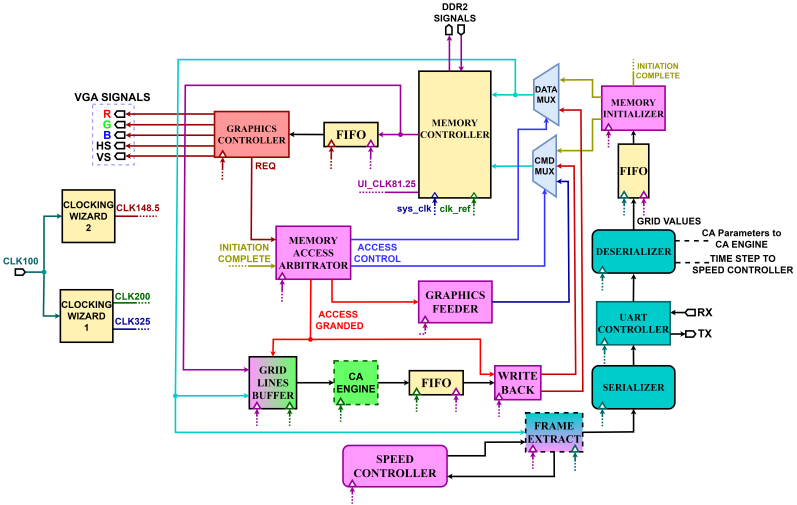
Manual Steps



CA Engine


Valid Signal
From Buffer
Pipelining Valid Signal
Column Registers
Pipelining Central Cell
Neighborhood
Column Received
from Buffer
3 x c
c
c
c
REG 00
REG 01
REG 02
REG 10
REG 11
REG 12
REG 20
REG 21
REG 22
MUL 20
MUL 21
MUL 22
MUL 02
MUL 01
MUL 00
MUL 12
MUL 11
MUL 10
R
R
R
Transition Rule
Central Cell's New State
Column 2 Weights
Binary Adder Tree
Column 0 Weights
Column 1 Weights

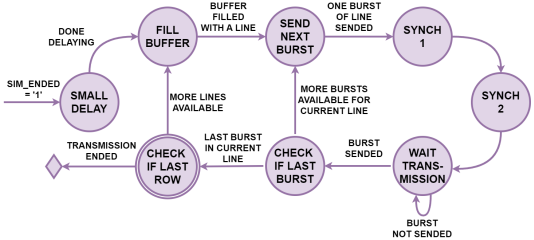
Top Level View of The Extended Architecture



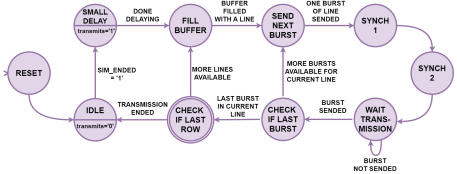
Frame Extraction and Speed Control

FRAME EXTRACT FSM

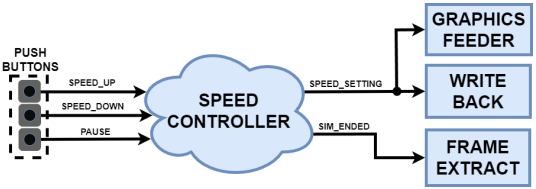
BEFORE

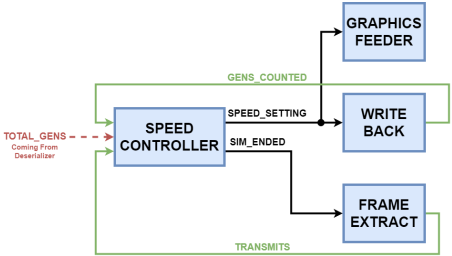


AFTER

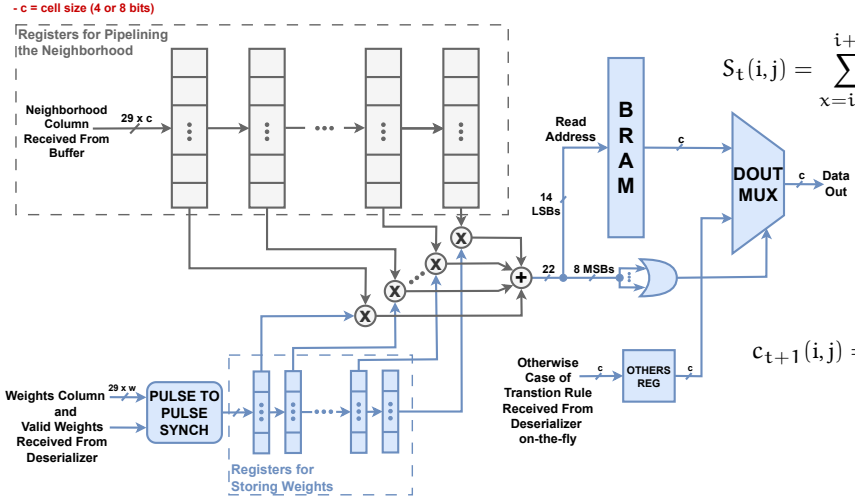


CIRCUIT





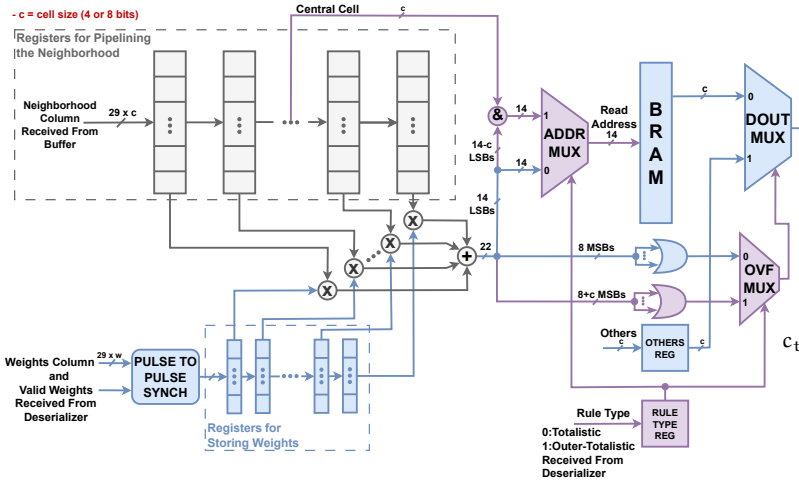
CA Engine: Totalistic Rules



$$S_t(i,j) = \sum_{x=i-r}^{i+r} \sum_{y=j-r}^{j+r} w(x-i, y-j) \times c_t(x,y)$$

$$c_{t+1}(i,j) = \begin{cases} 0, & \text{if } S_t(i,j) \in [0,19] \\ 1, & \text{if } S_t(i,j) \in (19,23] \\ 0, & \text{if } S_t(i,j) \in (23,58] \\ 1, & \text{if } S_t(i,j) \in (59,100] \\ 0, & \text{otherwise} \end{cases}$$

CA Engine: Outer-Totalistic Rules



$$S = \sum_{x=i-r}^{i+r} \sum_{y=j-r}^{j+r} w(x-i, y-j) \times c_t(x, y)$$

$$c_{t+1} = \begin{cases} 1, & \text{if } 2 \leq S \leq 3 \ \& \ c_t(x_0, y_0) = 1, \\ 1, & \text{if } S = 3 \ \& \ c_t(x_0, y_0) = 0, \\ 0, & \text{otherwise} \end{cases}$$

Protocol Buffers (or Protobuf): Background

What *Protobuf* is

- by Google
- Platform-Neutral
- Serialize Structured Data
- TLV scheme

Simple Message Example

```
message Example {
  repeated uint32 msg = 1;
}
```

Fields of Example:

- ☞ Label
- ☞ Data Type
- ☞ Name
- ☞ Field Number

Serialization Procedure:

Let msg = [723, 234, 5]

– Tag: $(\texttt{field_number} \ll 3)|\texttt{wire_type}$, wire_type = 2

$$T = ((1)_{10} \ll 3)|(2)_{10} = ((1)_2 \ll 3)|(10)_2 = (1000)_2|(010)_2 \implies$$
$$T = (00001010)_2 = (0a)_{16}$$

– Length: L= $(5)_{10} \implies$ L= $(05)_{16}$

– Value:

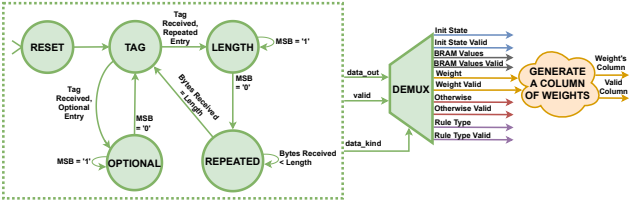
1. $(723)_{10} = (1011010011)_2$
2. 101 1010011
3. 00000101 01010011
4. 01010011 00000101
5. 11010011 00000101
6. D3 05

Final Value : V = [D305 EA01 05]

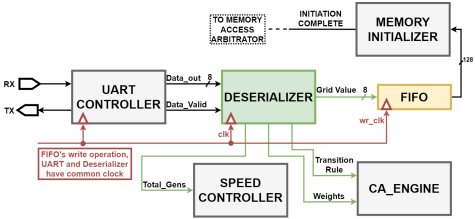
Final Message : msg = [0A05 D305 EA01 05]

Protocol Buffers (or Protobuf): Deserializer

Deseriliazer's FSM

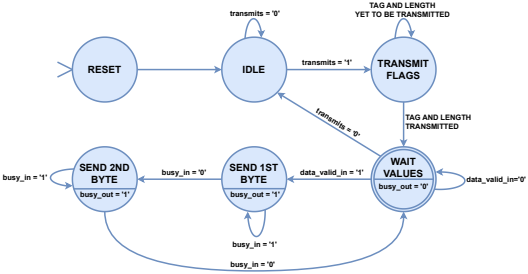


Connectivity within the circuit

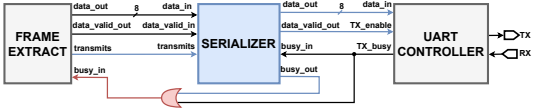


Protocol Buffers (or Protobuf): Serializer

Serializer's FSM



Connectivity within the circuit



Assembling the complete picture

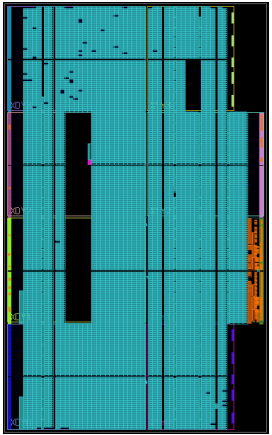
- 2 versions of *CA Engine*
  - 4-bit rules
  - 8-bit rules
- 3 different grid types
  - Rectangular
  - Cylindrical
  - Toroidal
- 6 bit files overall

Nexys 4 DDR Artix-7 XC7A100T Utilization

Components	Availability	Consumed	Percentage
Slices	15850	15563	98.20%
LUT	63400	42581	67.16%
FF	126800	69441	54.76%
LUTRAM	19000	1890	9.94%
BRAM (Tiles*)	135	125	92.59%
DSP	240	1	0.04%
IO	210	69	32.85%

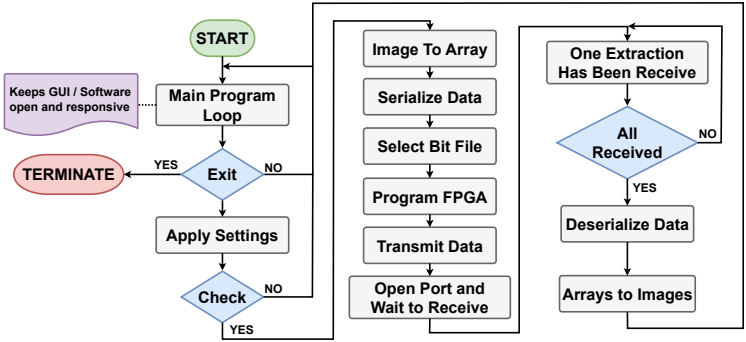
* 1 BRAM Tile is equal to 18Kbits

Implemented Circuit. Case of 8-bit Toroidal



Overview of The Tool

Flowchart of the back-end functionality



- Code consists of:
  - Python
    - ☞ Most functionalities
  - TCL commands
    - ☞ Integrated in Vivado
    - ☞ To Program FPGA

Cellular Automata Description Language (CDL)

The Greenberg-Hastings Model described in CDL

$$c_{t+1}(i,j) = \begin{cases} 1, & \text{if } c_t(i,j) = 0 \text{ and the number of excited neighbors} > T \\ c_t(i,j) + 1, & \text{if } c_t(i,j) > 0 \\ c_t(i,j), & \text{otherwise} \end{cases}$$

```
Definitions:
States: 0-quiescent | 1-excited | others-refraction;
T:4;
Function:
next_state = 1, if current_state = 0 and total_excited > T;
next_state = current_state + 1, if current_state > 0;
next_state = current_state, otherwise;
_END_
```

- "Definitions" are optional
- "Function" is mandatory
  - ☞ At least one transition rule
  - ☞ "otherwise" can't be omitted

General Form of CDL

```
Definitions:
value_1-name_1 | value_2-name_2 | ... | others-name_n
C_1 :N_1
C_2 :N_2
.
.
.
C_n :N_n
Function:
next_state = <expression1>, if <condition1>;
next_state = <expression2>, if <condition2>;
.
.
.
next_state = <expression>, otherwise;
```

General Form of supported CDL

```
Definitions:Empty
Function:
next_state=V_0, if sum in [X,Y] and current_state=S_0;
.
.
.
next_state=V_n, otherwise;
```

Interpreting CDL: Case 1

Totalistic Rules

Dummy Input

```
next_state=1, if sum in [0,10];
next_state=2, if sum in (15,20];
next_state=3, if sum in (23,78);
next_state=0, otherwise;
```

Output

```
LUT=1:[0,10]|2:(15,20]|3:(23,78)|0:others
```

Address	0	1		10	11		16		21		24		77	78		16383
Value	1	1	•••	1	0	•••	2	•••	0	•••	3	•••	3	0	•••	0

Interpreting CDL: Case 2

Outer-Totalistic Rules

Dummy Input

```
next_state=1, if sum in [0,10] and current_state=1;
next_state=2, if current_state=1;
next_state=3, if sum in [0,10];
next_state=0, otherwise;
```

Output

```
LUT=1:C1:[0,10]|2:C1|3:[0,10]|0:others
```

– Generalized Condition:

$c(t+1) = s$, when sum $\in [n \times L + a, n \times L + b)$

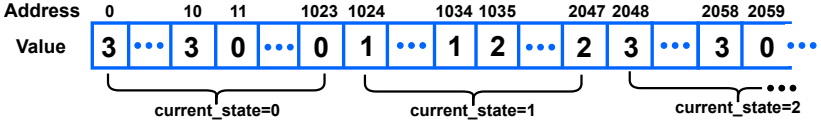
where,

s = next state

n = current state or (n+1)-th interval

a/b = lower/upper bounds

L = length of sub-intervals: 1024/64 for 4/8-bit



Data Serialization/Deserialization

Structure of Data (*".proto"*):

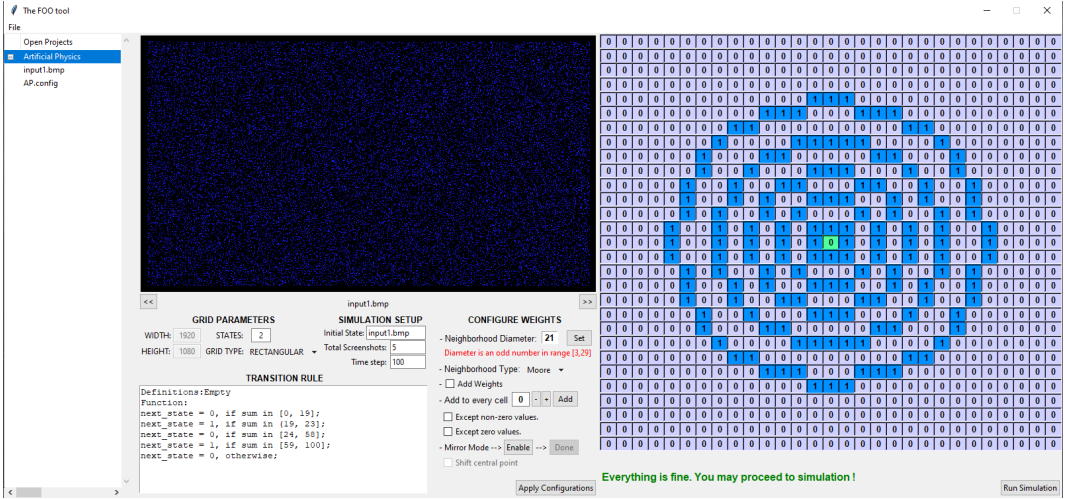
```
message DataStruct {
   optional uint32 time_step = 1;
   repeated uint32 weights = 2;
   repeated uint32 bram_values = 3;
   optional uint32 others = 4;
   optional uint32 rule_type = 5;
   repeated uint32 initial_state = 6;
}
```

1. Create Schema
2. Compile *".proto"*
3. File Descriptors
4. Integrated Functions
5. Binary File

Image to ASCII and vice versa

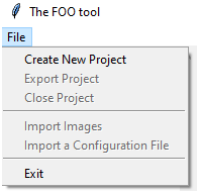
- ✓ PIL Library
- ✓ BMP files only
- ✗ JPEG or PNG are compressed
- ✓ Preserve color palette
- ✓ Zero-padding
- ✗ Correctness of input image.

GUI: Using TKinter

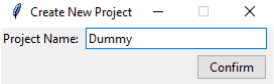


User's Options

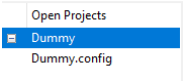
"File" Option



Create New Project

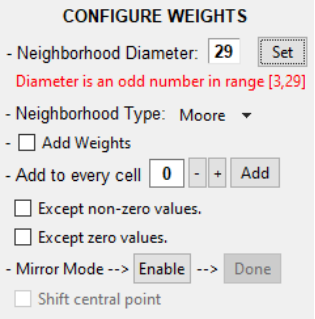


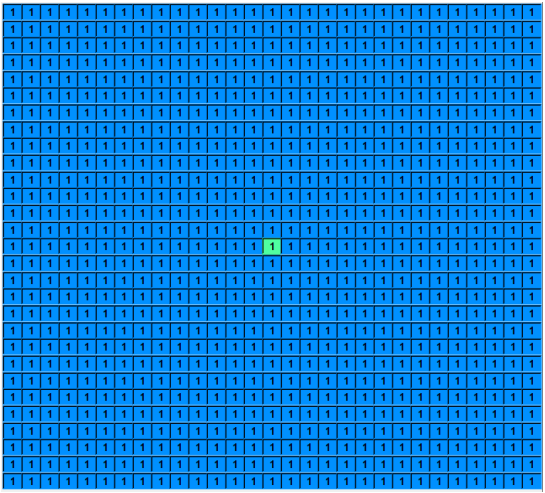
Updated Tree View



ERROR CHECKS

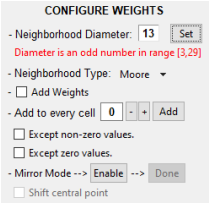
Configuring Neighborhood

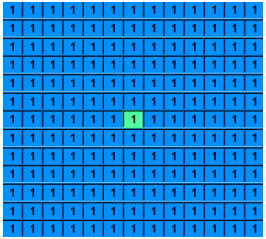




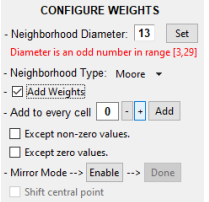
Configuring Neighborhood: Adding Weights

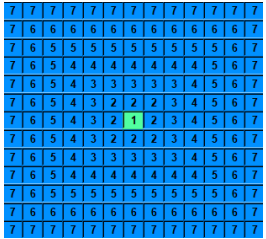
(a)





(b)



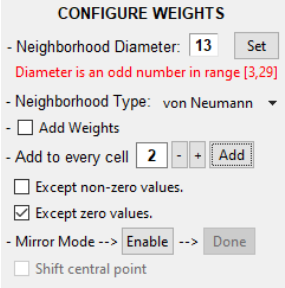


Configuring Neighborhood: Increasing/Decreasing Weights

Non weighted von Neumann

0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0

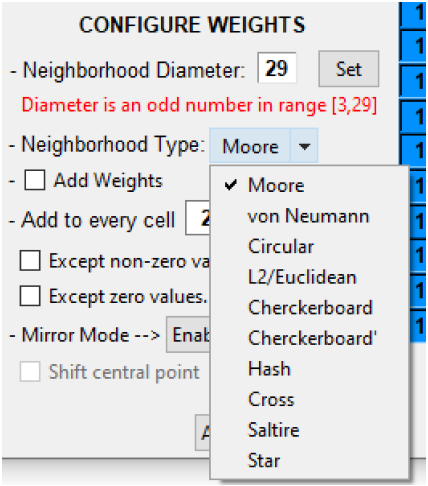
Adding 2 and excluding 0s



von Neuman preserved

0	0	0	0	0	0	3	0	0	0	0	0	0
0	0	0	0	0	3	3	3	0	0	0	0	0
0	0	0	0	3	3	3	3	3	0	0	0	0
0	0	0	3	3	3	3	3	3	3	0	0	0
0	0	3	3	3	3	3	3	3	3	3	0	0
0	3	3	3	3	3	3	3	3	3	3	3	0
3	3	3	3	3	3	3	3	3	3	3	3	3
0	3	3	3	3	3	3	3	3	3	3	3	0
0	0	3	3	3	3	3	3	3	3	3	0	0
0	0	0	3	3	3	3	3	3	3	0	0	0
0	0	0	0	3	3	3	3	3	0	0	0	0
0	0	0	0	0	3	3	3	0	0	0	0	0
0	0	0	0	0	0	3	0	0	0	0	0	0

Configuring Neighborhood: Neighborhood Types



Configuring Neighborhood: Neighborhood Types (1 ~ 2)

6	6	6	6	6	6	6	6	6	6	6
6	5	5	5	5	5	5	5	5	5	6
6	5	4	4	4	4	4	4	4	5	6
6	5	4	3	3	3	3	3	4	5	6
6	5	4	3	2	2	2	3	4	5	6
6	5	4	3	2	1	2	3	4	5	6
6	5	4	3	2	2	2	3	4	5	6
6	5	4	3	3	3	3	3	4	5	6
6	5	4	4	4	4	4	4	4	5	6
6	5	5	5	5	5	5	5	5	5	6
6	6	6	6	6	6	6	6	6	6	6

(a)

0	0	0	0	0	6	0	0	0	0	0
0	0	0	0	5	5	5	0	0	0	0
0	0	0	4	4	4	4	4	0	0	0
0	0	4	3	3	3	3	3	4	0	0
0	5	4	3	2	2	2	3	4	5	0
6	5	4	3	2	1	2	3	4	5	6
0	5	4	3	2	2	2	3	4	5	0
0	0	4	3	3	3	3	3	4	0	0
0	0	0	4	4	4	4	4	0	0	0
0	0	0	0	5	5	5	0	0	0	0
0	0	0	0	0	6	0	0	0	0	0

(b)

(a) Moore: $|x - x_0| \leq r \wedge |y - y_0| \leq r$

(b) von Neumann: $|x - x_0|^2 + |y - y_0|^2 \leq r$

Configuring Neighborhood: Neighborhood Types (3 ~ 4)

0	0	0	0	6	6	6	0	0	0	0
0	0	5	5	5	5	5	5	5	0	0
0	5	4	4	4	4	4	4	4	5	0
0	5	4	3	3	3	3	3	4	5	0
6	5	4	3	2	2	2	3	4	5	6
6	5	4	3	2	1	2	3	4	5	6
6	5	4	3	2	2	2	3	4	5	6
0	5	4	3	3	3	3	3	4	5	0
0	5	4	4	4	4	4	4	4	5	0
0	0	5	5	5	5	5	5	5	0	0
0	0	0	0	6	6	6	0	0	0	0

(a)

0	0	0	0	0	6	0	0	0	0	0
0	0	5	5	5	5	5	5	5	0	0
0	5	4	4	4	4	4	4	4	5	0
0	5	4	3	3	3	3	3	4	5	0
0	5	4	3	2	2	2	3	4	5	0
6	5	4	3	2	1	2	3	4	5	6
0	5	4	3	2	2	2	3	4	5	0
0	5	4	3	3	3	3	3	4	5	0
0	5	4	4	4	4	4	4	4	5	0
0	0	5	5	5	5	5	5	5	0	0
0	0	0	0	0	6	0	0	0	0	0

(b)

(a) Circular: $|x - x_0|^2 + |y - y_0|^2 \leq (r + 1)^2$

(b) L2/Euclidean: $|x - x_0|^2 + |y - y_0|^2 \leq r^2$

Configuring Neighborhood: Neighborhood Types (5 ~ 6)

0	6	0	6	0	6	0	6	0	6	0
6	0	5	0	5	0	5	0	5	0	6
0	5	0	4	0	4	0	4	0	5	0
6	0	4	0	3	0	3	0	4	0	6
0	5	0	3	0	2	0	3	0	5	0
6	0	4	0	2	0	2	0	4	0	6
0	5	0	3	0	2	0	3	0	5	0
6	0	4	0	3	0	3	0	4	0	6
0	5	0	4	0	4	0	4	0	5	0
6	0	5	0	5	0	5	0	5	0	6
0	6	0	6	0	6	0	6	0	6	0

(a)

6	0	6	0	6	0	6	0	6	0	6
0	5	0	5	0	5	0	5	0	5	0
6	0	4	0	4	0	4	0	4	0	6
0	5	0	3	0	3	0	3	0	5	0
6	0	4	0	2	0	2	0	4	0	6
0	5	0	3	0	1	0	3	0	5	0
6	0	4	0	2	0	2	0	4	0	6
0	5	0	3	0	3	0	3	0	5	0
6	0	4	0	4	0	4	0	4	0	6
0	5	0	5	0	5	0	5	0	5	0
6	0	6	0	6	0	6	0	6	0	6

(b)

(a) Checkerboard: Moore Set $\wedge\, |x| + |y| = 2 \times k + 1, k \in Z$

(b) Checkerboard': Moore Set $\wedge\, |x| + |y| = 2 \times k, k \in Z$

Configuring Neighborhood: Neighborhood Types (7 ~ 8)

0	0	0	0	6	0	6	0	0	0	0
0	0	0	0	5	0	5	0	0	0	0
0	0	0	0	4	0	4	0	0	0	0
0	0	0	0	3	0	3	0	0	0	0
6	5	4	3	2	2	2	3	4	5	6
0	0	0	0	2	0	2	0	0	0	0
6	5	4	3	2	2	2	3	4	5	6
0	0	0	0	3	0	3	0	0	0	0
0	0	0	0	4	0	4	0	0	0	0
0	0	0	0	5	0	5	0	0	0	0
0	0	0	0	6	0	6	0	0	0	0

(a)

0	0	0	0	0	6	0	0	0	0	0
0	0	0	0	0	5	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0
0	0	0	0	0	3	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0
6	5	4	3	2	1	2	3	4	5	6
0	0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	3	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0
0	0	0	0	0	5	0	0	0	0	0
0	0	0	0	0	6	0	0	0	0	0

(b)

(a) Hash: $x = x_0 - 1 \lor x = x_0 + 1 \lor y = y_0 - 1 \lor y = y_0 + 1$

(b) Cross: $x = x_0 \lor y = y_0$

Configuring Neighborhood: Neighborhood Types (9 ~ 10)

6	0	0	0	0	0	0	0	0	0	6
0	5	0	0	0	0	0	0	0	5	0
0	0	4	0	0	0	0	0	4	0	0
0	0	0	3	0	0	0	3	0	0	0
0	0	0	0	2	0	2	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	2	0	2	0	0	0	0
0	0	0	3	0	0	0	3	0	0	0
0	0	4	0	0	0	0	0	4	0	0
0	5	0	0	0	0	0	0	0	5	0
6	0	0	0	0	0	0	0	0	0	6

(a)

6	0	0	0	0	6	0	0	0	0	6
0	5	0	0	0	5	0	0	0	5	0
0	0	4	0	0	4	0	0	4	0	0
0	0	0	3	0	3	0	3	0	0	0
0	0	0	0	2	2	2	0	0	0	0
6	5	4	3	2	1	2	3	4	5	6
0	0	0	0	2	2	2	0	0	0	0
0	0	0	3	0	3	0	3	0	0	0
0	0	4	0	0	4	0	0	4	0	0
0	5	0	0	0	5	0	0	0	5	0
6	0	0	0	0	6	0	0	0	0	6

(b)

(a) Saltire: $y = x + 2 \times y_0 \ \vee \ y = -x$

(b) Star: $y = x + 2 \times y_0 \ \vee \ y = -x \ \vee \ x = x_0 \ \vee \ y = y_0$

Configuring Neighborhood: Mirror Mode

Mirror Mode Enabled

0	0	0	0	0	0	7	0	0	0	0	0	0
0	0	0	0	0	6	6	0	0	0	0	0	0
0	0	0	0	5	5	5	0	0	0	0	0	0
0	0	0	4	4	4	4	0	0	0	0	0	0
0	0	5	4	3	3	3	0	0	0	0	0	0
0	6	5	4	3	2	2	0	0	0	0	0	0
7	6	5	4	3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Done

0	0	0	0	0	0	7	0	0	0	0	0	0
0	0	0	0	0	6	6	6	0	0	0	0	0
0	0	0	0	5	5	5	5	5	0	0	0	0
0	0	0	4	4	4	4	4	4	4	0	0	0
0	0	5	4	3	3	3	3	3	4	5	0	0
0	6	5	4	3	2	2	2	3	4	5	6	0
7	6	5	4	3	2	1	2	3	4	5	6	7
0	6	5	4	3	2	2	2	3	4	5	6	0
0	0	5	4	3	3	3	3	3	4	5	0	0
0	0	0	4	4	4	4	4	4	4	0	0	0
0	0	0	0	5	5	5	5	5	0	0	0	0
0	0	0	0	0	6	6	6	0	0	0	0	0
0	0	0	0	0	0	7	0	0	0	0	0	0

Configuring Neighborhood: Enhanced Mirror Mode

Mirror Mode Enabled

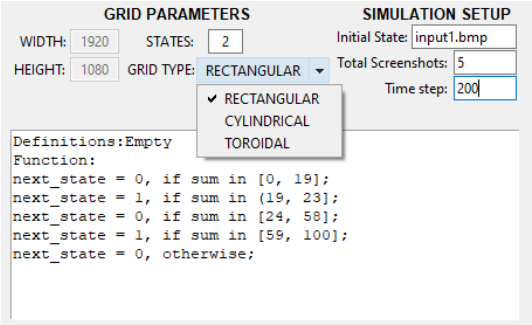
0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	3	3	3	0	0	0	0	0	0	0	0
0	3	2	2	2	3	0	0	0	0	0	0	0
4	3	2	1	2	3	4	0	0	0	0	0	0
0	3	2	2	2	3	0	0	0	0	0	0	0
0	0	3	3	3	0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Done

0	0	0	4	0	0	0	0	0	4	0	0	0
0	0	3	3	3	0	0	0	3	3	3	0	0
0	3	2	2	2	3	0	3	2	2	2	3	0
4	3	2	1	2	3	4	3	2	1	2	3	4
0	3	2	2	2	3	0	3	2	2	2	3	0
0	0	3	3	3	0	0	0	3	3	3	0	0
0	0	0	4	0	0	0	0	0	4	0	0	0
0	0	3	3	3	0	0	0	3	3	3	0	0
0	3	2	2	2	3	0	3	2	2	2	3	0
4	3	2	1	2	3	4	3	2	1	2	3	4
0	3	2	2	2	3	0	3	2	2	2	3	0
0	0	3	3	3	0	0	0	3	3	3	0	0
0	0	0	4	0	0	0	0	0	4	0	0	0

Rest Configurations + ".config" file

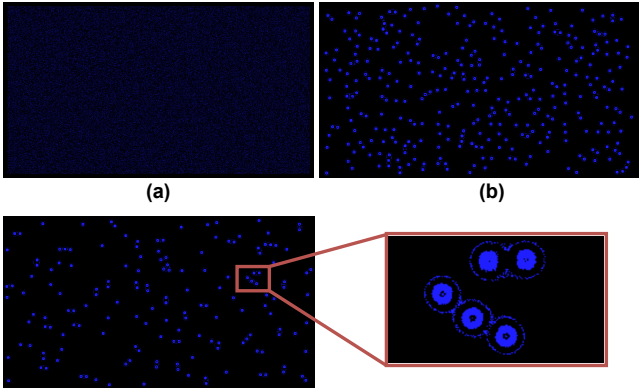
CA Parameters & Transition Rule



An Example of a configuration file

```
*** PLEASE DO NOT EDIT. This file is auto-generated by the tool. ***
States:2
Grid_Type:RECTANGULAR
# Transition Rule
Definitions:Empty
Function:
next_state = 0, if sum in [0, 19];
next_state = 1, if sum in (19, 23];
next_state = 0, if sum in [24, 58];
next_state = 1, if sum in [59, 100];
next_state = 0, otherwise;
_END_
NEIGHBORHOOD:3
0 2 0
2 1 2
0 2 0
```

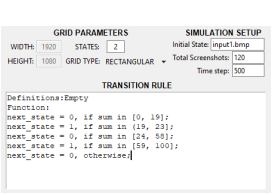
Artificial Physics



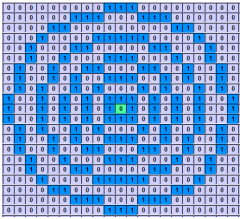
(a) (b) (c)

(a) Initial State, **(b)** 500 gens and **(c)** 60,000 gens

$$c_{t+1}(i,j) = \begin{cases} 0, & \text{if } S_t(i,j) \in [0,19] \\ 1, & \text{if } S_t(i,j) \in (19,23] \\ 0, & \text{if } S_t(i,j) \in (23,58] \\ 1, & \text{if } S_t(i,j) \in (59,100] \\ 0, & \text{otherwise} \end{cases}$$



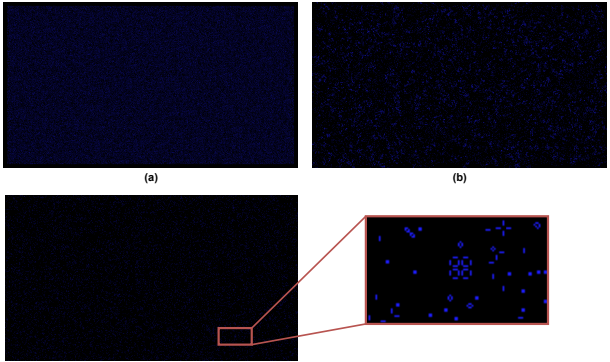
(a)



(b)

(a) Parameters, Transition Rule
(b) 21×21 neighborhood within the GUI.

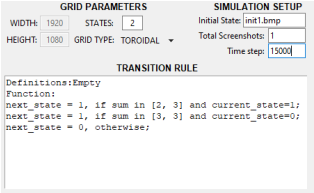
Game Of Life



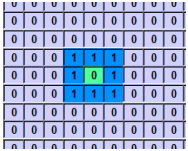
(a) (b) (c)

(a) Initial State, **(b)** 500 gens and **(c)** 15,000 gens

$$c_{t+1}(x,y) = \begin{cases} 1, & \text{if } 2 \leq S \leq 3 \ \& \ c_t(x_0,y_0) = 1, \\ 1, & \text{if } S = 3 \ \& \ c_t(x_0,y_0) = 0, \\ 0, & \text{otherwise} \end{cases}$$



(a)



(b)

(a) Parameters, Transition Rule
(b) 3×3 neighborhood within the GUI.

Hodgepodge Machine: Experiments

– Definition:

$$c_{t+1}(i,j) = \begin{cases} \frac{K_t(i,j)+I_t(i,j)}{k} & , \text{ if } c_t(i,j) = 0 \\ 0 & , \text{ if } c_t(i,j) = q \\ \frac{\text{sum of all neighbors}}{I_t(i,j)} + g & , \text{ otherwise} \end{cases}$$

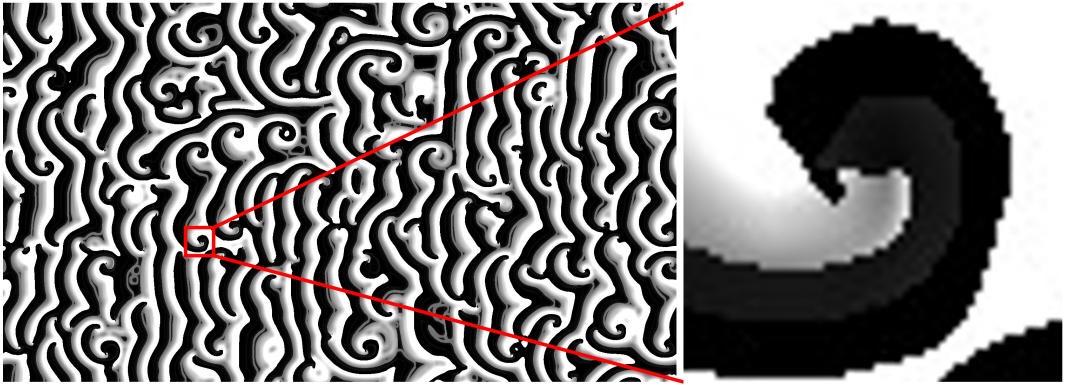
- $q+1$ states $\longrightarrow [0, q]$
  - ☞ 0: "healthy"
  - ☞ q: "ill"
  - ☞ $[1, q-1]$: degree of "infection"
- $K_t(i,j)$: "ill" neighbors
- $I_t(i,j)$: "infected" neighbors
- k: "weighting" parameter of healthy cells
- g: growth rate of infection

– Our case:

- $q = 255$, $q+1 = 256$ states
- $k = 5$
- $g = 105$
- 1920x1080 Toroidal Grid
- Initial State
  - ☞ Bernoulli's Variable: $p \rightarrow 0$s, $(1-p) \rightarrow 1$s
  - ☞ Uniform distribution in $[1, 255]$
- Non-weighted Moore Neighborhoods, 15×15 up to 29×29

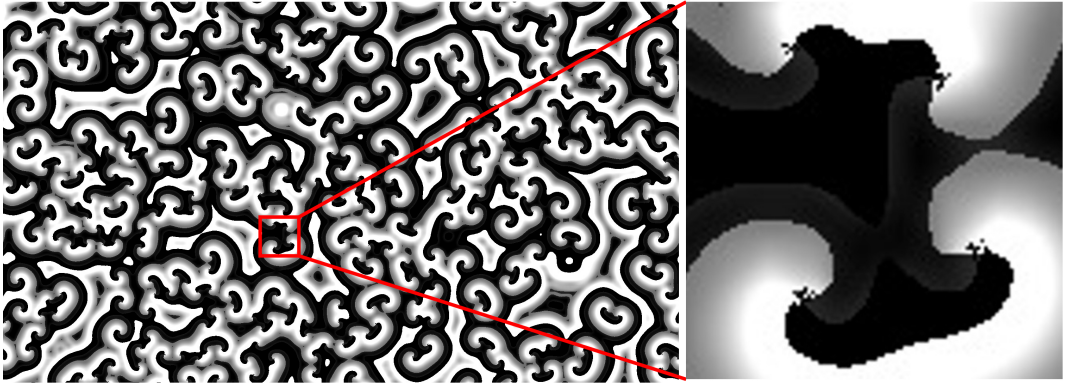
➡ Interested in: *Chimera States*

Experiment 1



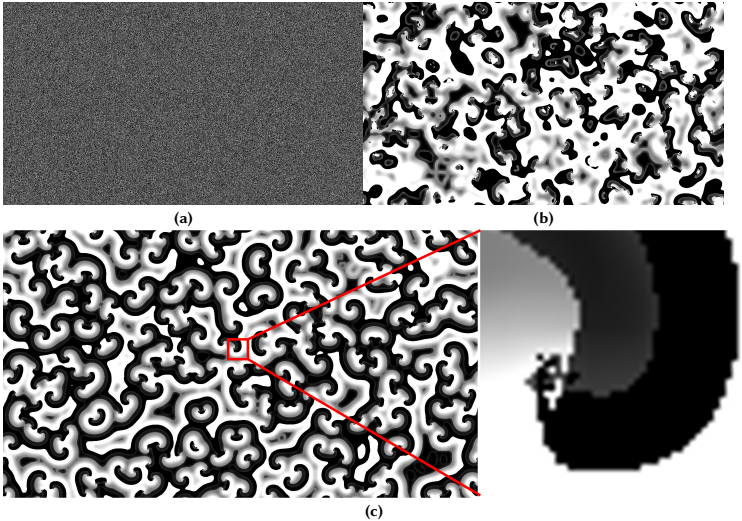
15×15 Neighborhood, $p = 47\%$, System Converged

Experiment 2



17×17 Neighborhood, $p = 50\%$, System Converged

Experiment 3



- 29×29 Neighborhood

(a) Initial State: $p = 57\%$

(b) 40 Generations

(c) 200 Generations

Results

Neighborhood Size	Range of p	Total Length of range
19×19	$[49, 53]$	5
21×21	$[49, 53]$	5
23×23	$[50, 54]$	5
25×25	$[52, 58]$	7
27×27	$[49, 59]$	11
29×29	$[51, 62]$	12

- Chimera states existence
- The larger radius, the wider the window of p
- Ideal value at the center of window
- Center shifts to higher values

Conclusions

- ✓ Re-programmable Framework
- ✓ CAD Tool
- ✓ Automated Initialization and vice versa
- ✓ Protocol Buffers
- ✓ CDL
- ✓ GUI

Extra Models

– The Greenberg-Hastings

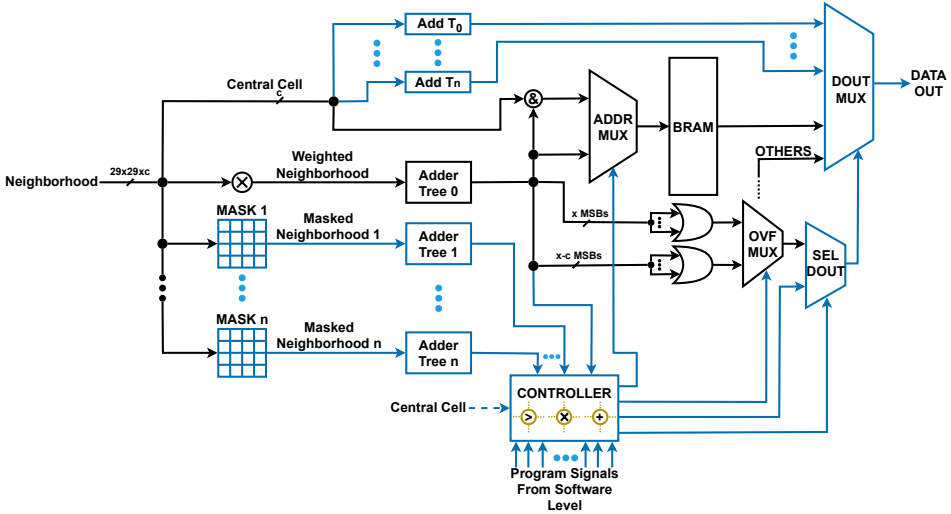
$$c_{t+1}(i,j) = \begin{cases} 1, & \text{if } c_t(i,j) = 0 \ \& \text{ the total excited neighbors} > T, \\ & \text{where T is a threshold value} \\ c_t(i,j) + 1, & \text{if } c_t(i,j) > 0 \\ c_t(i,j), & \text{otherwise} \end{cases}$$

– Anistropic Rules

$$c_{t+1}(i,j) = \begin{cases} c_t(i,j) - 1, & \text{if } S_t(i,j) = 0 > \text{threshold} \\ c_t(i,j) + 1, & \text{if } S_t(i,j) = 0 < \text{threshold} \\ c_t(i,j), & \text{otherwise} \end{cases}$$

+ The Hodgepodge Machine

A Potential Hardware Approach



Last But Not Least

- ✓ Amazon Cloud EC2 F1 Instance
  - ☞ "State-of-the-art" UltraScale+ FPGA
  - ☞ PCIe fabric
  - ✘ UART
  - ✘ Graphics Controller
- ✓ Video streaming in real-time
- ✓ Control Simulation Speed
- ✓ Image Drawing/Generator Tool

⋮

"The Nature is a book written in the language of mathematics."
~ Galileo Galilei

Thank You!

Any Questions?
Share Screen?