The Implementation of Logic Gates in Conway's Game of Life and its Implication

Seldi Kurnia Trihardja / 13516042[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516042@std.stei.itb.ac.id
xeldkt@gmail.com

***Abstract*— This paper will cover mainly the prototypical cellular automata and zero-player game, Game of Life by John Conway and his team, and the formation of logic gates and its implication inside the game. Conway's Game of Life is a set of simple rules on a grid that changes the grid based on the rules and its initial pattern, and is proven to have the power of a turing machine, and thus Turing Complete. One of the contributing factor to this are the logic gates that are constructed in the Game.**

***Keywords*— Conway's game of life, Logic gates, Turing machine**

I. Introduction

Conway's Game of Life ,shortened to Life for the remainder of the paper, is originally a pen and paper mathematical game with very simple rules. This game originated from the brains of John Conway and the group of graduates that he assembled at Cambridge. Originally Life began as an effort to simplify Von Neumann theoretical machine that can copy itself. This Neumann two dimensional automated-machine, where a square could be in one of 29 states, in which the state depends on the surrounding four squares, could produce 20,155,149 or 29^5 different configuration for each square. For Conway, this is "excessively complicated" and so he envisioned a simple and elegant universal machine. First he came across Fractran, a one dimensional universal machine that uses a row of fractions to do various things, one of the most popular is the one that produces the prime powers of 2. In spite of his new invention, Fractran has obvious defects such as the length of computation that it requires to do its job, it took millions of operation to produce not comparable results. And so Conway's search to invent an even simpler and less cumbersome solution continues.

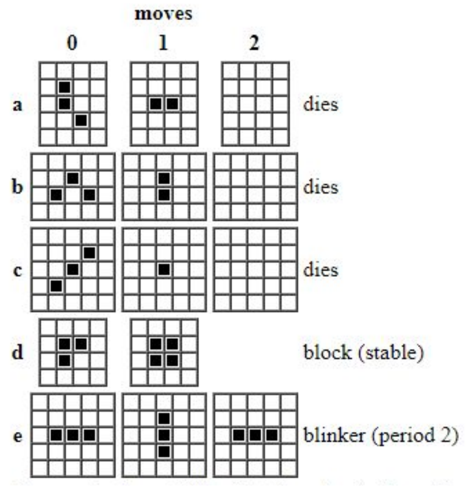


Figure 1 . Example of Conway's Game of Life behaviour for 2 generations (https://web.archive.org/web/20090603015231/http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm)

After a set of inventions by him and his team, they arrived at a prototype of Life. And after many experimentation they arrived at the current rules of Life. Which gave birth to an interesting ruleset that shows interesting patterns out of seemingly random initial position. After that Martin Gardener popularized Life through his column on the Scientific American on 1970. From then on, Conway's invention steadily grow in popularity and people started coming together to find patterns and "structures" in Life. Its thought-provoking parallel to life with its idea of Survival, Death, Reproduction, Underpopulation, and Overpopulation also helps its popularity in fields other than mathematics such as philosophy due to the results Life shows of "Life". Life surprisingly can produce a variety of "lifeforms" and exciting starting confgurations such as "The Garden of Eden" and the "Puffers". Even though its original creator has already left the field, the sustained interest from the mathematical community and advances in computational power steadily gave birth to new innovation in

Life. One of the groundbreaking pattern in Life are The Gosper Glider Gun. The Gosper Glider Gun is a infinitely looping pattern that produces another pattern that moves, and so it is a pattern that infinitely grows provided infinite time and space. The properties of this Gun causes a chain reaction of innovation in Life, leading to proving that Life is Turing Complete, and the production of Turing machines inside Life

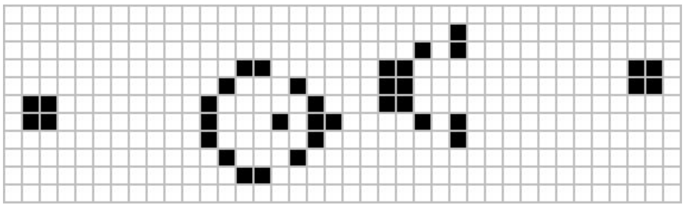


Figure 2 . Gosper Glider Gun initial state (https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

All that comes from the implementation of glider guns to make logic gates and many more structure to simulate a computer. As such this paper will explain the production of logic gates in Life and the many implications that it brings.

II. Theory of Logic Gates

Logic Gates, or Logic Circuits are a form of representation for Boolean Functions. Or simply, it produces a boolean value according to the gates and the input. Since computers basicly uses boolean values (0 or 1), these logic gates are used in Integrated Circuits in the form of transistors and other electrical devices to represent the bits.

There are three basic logical gates, AND gates, OR gates, and NOT gates.

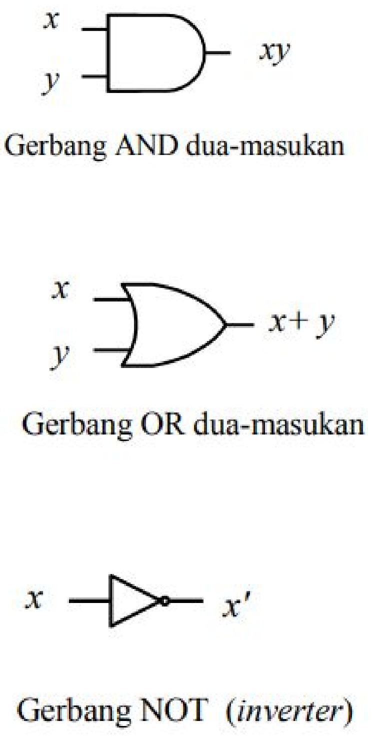


Figure 3 . AND OR NOT Gates (Slide Kuliah)

From those three basic gates that are shown on Fig. 3, we can derive more gates by combining the basic gates. Those logic gates are : NAND, NOR, XOR, and XNOR.

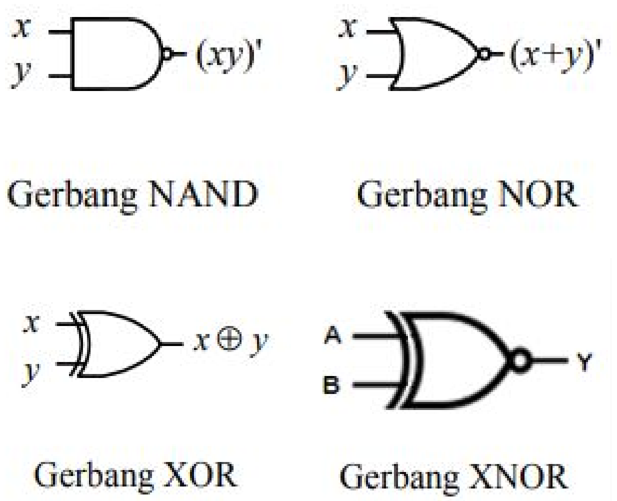


Figure 4 . Derived Logic Gates (Slide Kuliah)

All of the above logic gates will output a certain boolean signal according to the input as shown on the table below, where X and Y are inputs as seen on Fig.3 and Fig.4.

INPUTS	X	Y	OUTPUT
AND	0	0	0
	0	1	0
	1	0	0
	1	1	1
OR	0	0	0
	0	1	1
	1	0	1
	1	1	1
NOT	0		1
	1		0
NAND	0	0	1
	0	1	1
	1	0	1
	1	1	0
NOR	0	0	1
	0	1	0
	1	0	0
	1	1	0
XOR	0	0	0
	0	1	1
	1	0	1
	1	1	0
XNOR	0	0	1
	0	1	0
	1	0	0
	1	1	1

Table I . Logic Gates Truth Table

Logic gates and the theory behind it are the building blocks of modern computers. With logic gates, we can create multiplexers, registers, ALU(Arithmetic Logic Unit), computer memory, or even microprocessors that can contain hundreds of millions of gates. With logic gates, all of algorithms and mathematics that can be represented in boolen logic can be represented in the physical world.

III. Brief Explanation of Turing Machines and Turing Completeness

A Turing Machine in a nutshell is a machine that can read something from a tape of theoretical infinite length and according to the data that it read it can do actions such as writing a new data into the tape or "move" to another part of the tape. The Turing Machine is important because it is the most powerful automata because it can write to an infinite tape, which gives it infinite memory. A Turing Machine that can simulate a Turing machine with an input is called an Universal Turing Machine. A Universal Turing Machine has been proven to be able to compute anything that can be computed algorithmically. To put it into perspective, anything you can do on your computer can theoretically be done in a Turing Machine. This trait is often reffered to as the machine to be "Turing Complete". Even though there are no infinites in the physical world, the machine that is Turing Complete would be so if it is given infinite resources to work with, and so it can do task such as simulating calculators, memory, and computers. Another important aspect of a Universal Turing Machine is that it could loop infinitely. The study and the details of the Turing Machine and the condition of Turing completeness is beyond the scope of this paper so it will not be included here.

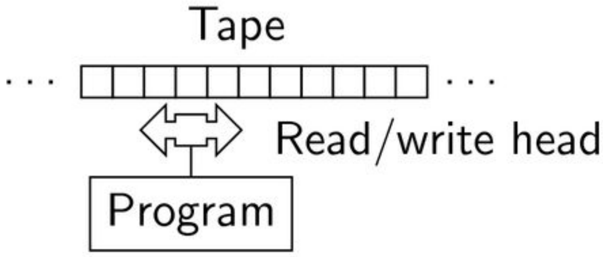


Figure 5 . Turing Machine concept

III. Conway's Game of Life

A. Rules of the Game

The game is played on an infinitely large grid consisting of single squares that are called cells. These cells can be in one of two states which is "alive" or "dead" that are usually represented by the cells lighting up with a color different than the color of the grid. Each cell has eight neighbours around it. Neighbours are cells that are horizontally vertically, or diagonally adjacent to another cell. The state of each cell are determined by the states of all 8 of its neighbours using these simple rules :

1. Any live cell with fewer than two live neighbours will die, as if caused by underpopulation .
2. Any live cell with two or three live neighbours live on to the next generation
3. Any live cell with more than three live neighbours will die, as if caused by overpopulation
4. Any dead cell with exactly three neighbours becomes a live cell, as if by reproduction.

The Game operates in iterations called ticks or also often called generations. The Game then will apply the four rules to the grid simultaneously to all the cells every iteration.

B. Patterns

As said in the introduction, through 4 simple and elegant rules that are devised by Conway, interesting pattern emerges from Life. As more and more patterns show up, Conway started to classify the patterns. With the founding of new type of patterns, classes also increases to accommodate the new patterns of Life

First, there are patterns that is classified as "Still Life". In correspondence to the name, this class of patterns does not change at all except with outside interaction that most probably will break the stillness.

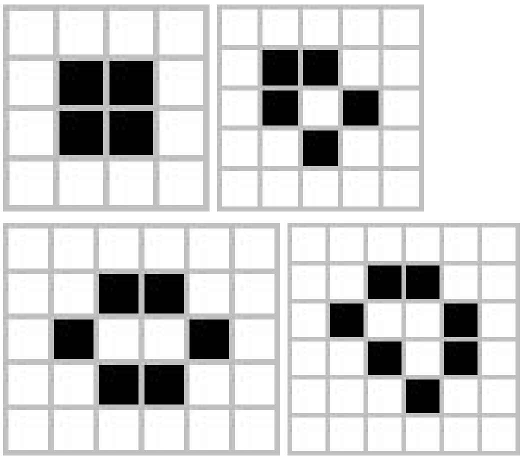


Figure 6 . Still Life (From top left to bottom right, Block, Boat, Beehive, Loaf)

Next, there are oscillators. These patterns are more complex and change over a specific number of ticks and they repeat their pattern infinitely. The basic oscillators have periods of two or three ticks, but complex oscillators have been discovered with periods of twenty or more ticks.

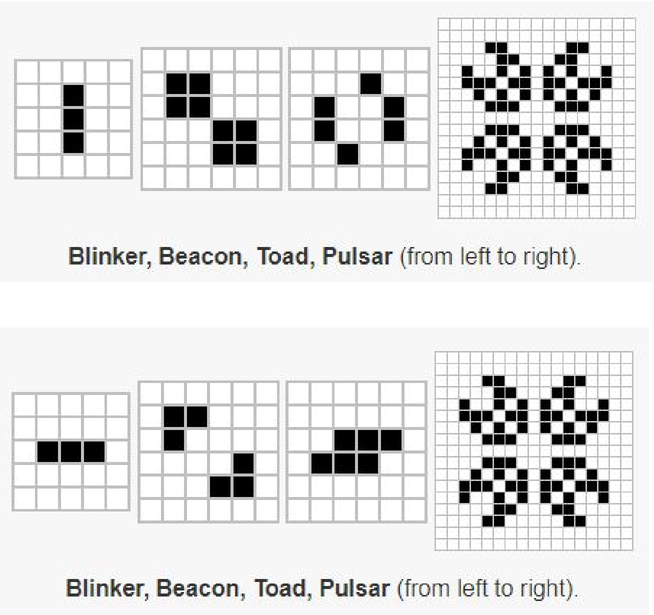
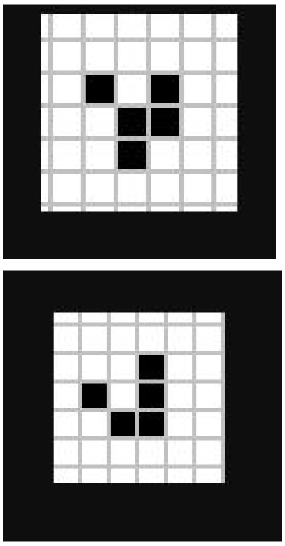

Blinker, Beacon, Toad, Pulsar (from left to right).
Blinker, Beacon, Toad, Pulsar (from left to right).

Figure 7 . Oscillator in 2 generations

The two pictures are the two phases of three basic oscillators that are the Blinker, Beacon, and Toad. These three oscillators will go back and forth according to its pattern for an indefinite amount of time. The Pulsar on the right have a different period of oscillation compared to the three to the left of it and so in this instance the repeating transformation is not captured well.

Moving on, is the Spaceship class of Life's patterns. A Spaceship is a pattern that changes and moves along the grid and then changes back to its original pattern after a set number of finite generations. The spaceship type is one of the more useful type of patterns in Life because the movement of the pattern can then be compared to a transmission of some kind, like an electrical signal in a perfect circuit because the speed of the spaceship is always constant.



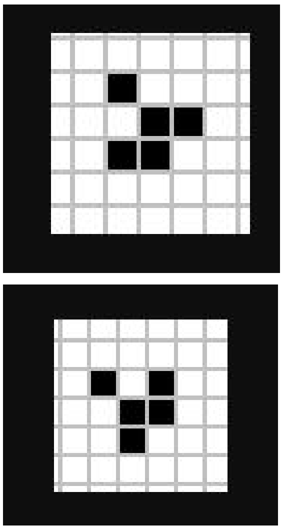


Figure 8 . The Glider, a simple spaceship

The next pattern that we will gothrough is the Gun. The Gun is one of the most important aspects of any complicated Life creation. What the Gun does is it produces a glider from a pair of oscillating patterns colliding. The simplest gun in Life was found by Bill Gosper in 1970 as a response to Conway himself challenging anyone that can make a configuration that grows in live cells for 50 dollars. The gun itself is named Gosper Glider Gun, named based on its founder. The gun is extraordinary because given infinite time and space, the amount of live cells will grow to be infinite too. Using the gun Life could create all logic gates and thus making it theoretically Turing Complete.

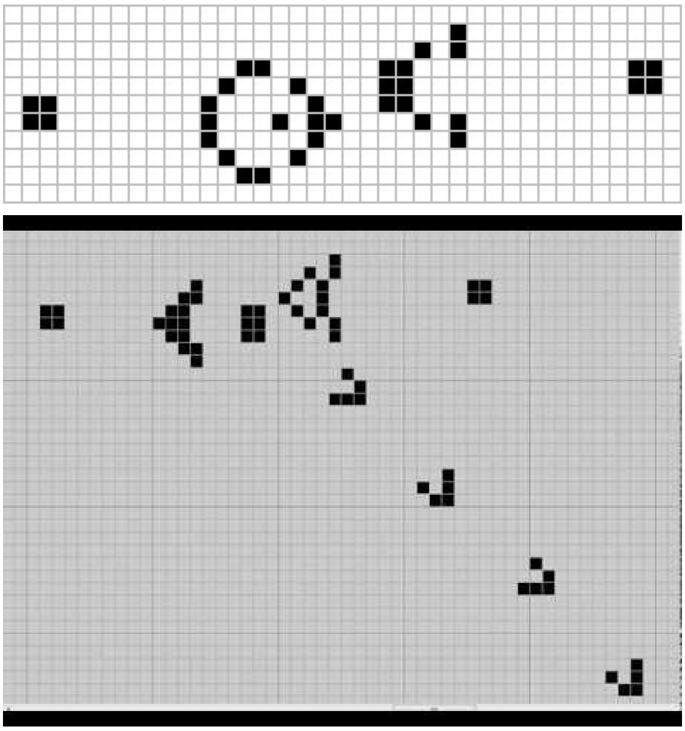


Figure 9 . Gosper Glider Gun

Other than that, there are plenty more examples of the many variants of patterns for each class. There are also other classes beside the fundamental ones that are already mentioned. Those classes are Puffers, Rakes, and Breeders. The Puffers are moving patterns that leave behind stable or oscillating debris. The Rakes are moving patterns that emits spaceships at regular intervals as they move, while the Breeders are complex moving patterns that leave behind a trail of Guns.

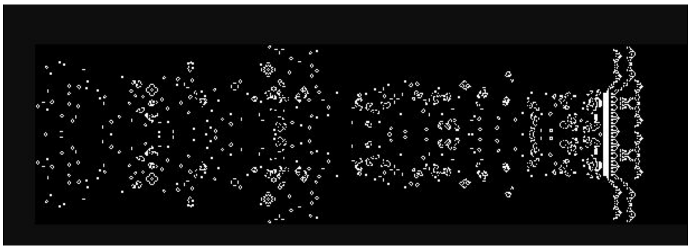


Figure 10 . A Puffer Pattern

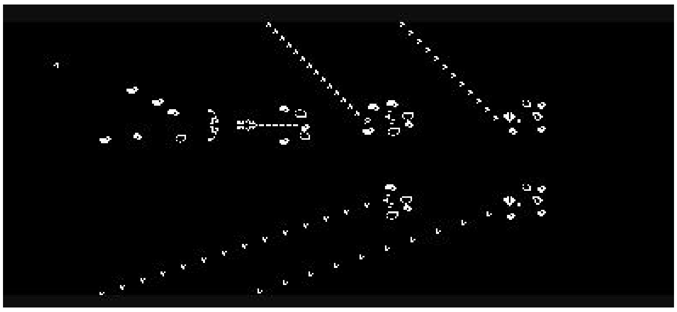


Figure 11 . A Rake Pattern

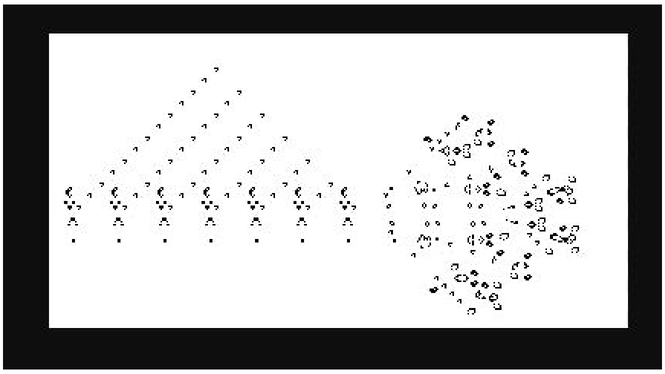
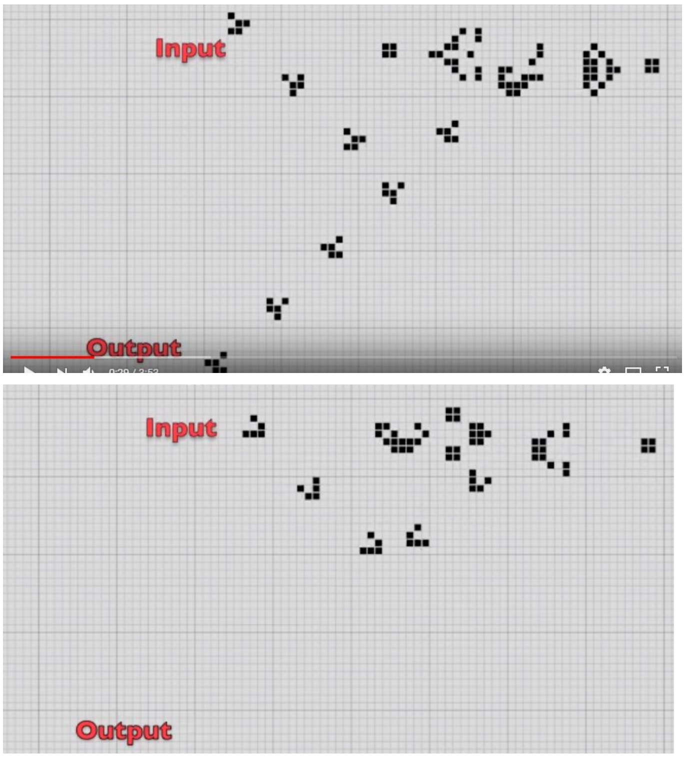


Figure 12 . A Breeder

C. Logic Gates

Using the information that we receive from previous sections, we can construct the basic logic gates using Gosper Glider Gun, treating the presence of a glider stream to be a form of boolean I/O



.Figure 13 . An NOT Gate in Life

Above, in Fig.13 is the representation of a NOT Gate in Life, The stream from both paths will terminate each other if the input is True but will continue and output True if the input is False, hence its a NOT Gate

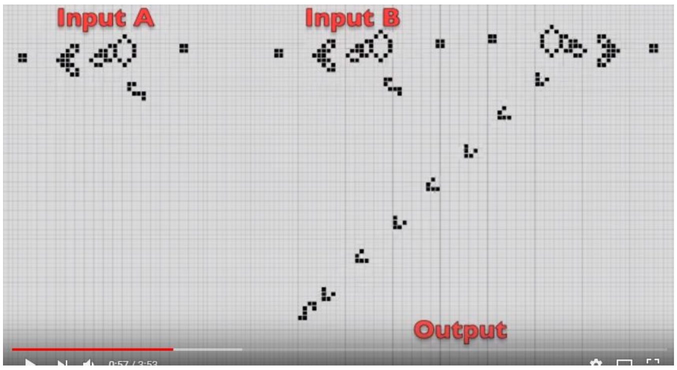


Figure 14 . An AND Gate in Life

In the representation of an AND Gate on Fig.14, we can see that when the input is both false it will still be false because no input gets to the output. Meanwhile if only one of the input is turned on, the stream will get terminated by the stream of gliders that goes to the bottom left of the figure, and so it is an AND Gate because both need to be true to get an output to be true

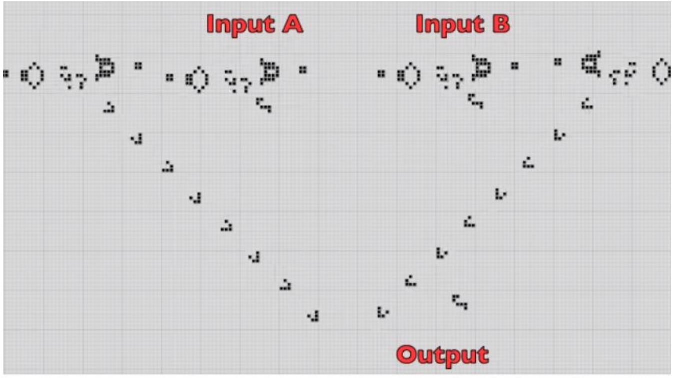


Figure 15 . An OR Gate in Life

Last but not least is the OR Gate in Fig.15. In it the inputs are set to go to the left once turned on. From the configuration we can see that to get an output of gliders, all we need to do is terminate the stream that is coming from the top right by using any one of our inputs, and so we succesfully made an OR Gate.

From the three basic logic gates, we can combine them to get the other logic gates by using the laws of Boolean Algebra and converting them to the form of basic logic gates.

IV. Implications

Even though the implementation of logic gates in Conway's Game of Life seems to be ordinary, its impact on the mathematical and computer science world are big. As far as we know from 1970, Turing Complete machines are the zenith of theoretical computational models and it is not fully explored yet at that time. And so, Life became one of many innovations that shape the way for this field of study of cellular automata. Through the production of logic gates in Life, proof of its Turing Completeness became easy to find, but nobody has actually proven it with tangible methods before. Paul Rendell, 30 years after the invention of Life designed an actual Turing Machine in Life, further cementing Lifes place among the most powerful computational yet simple machines ever.

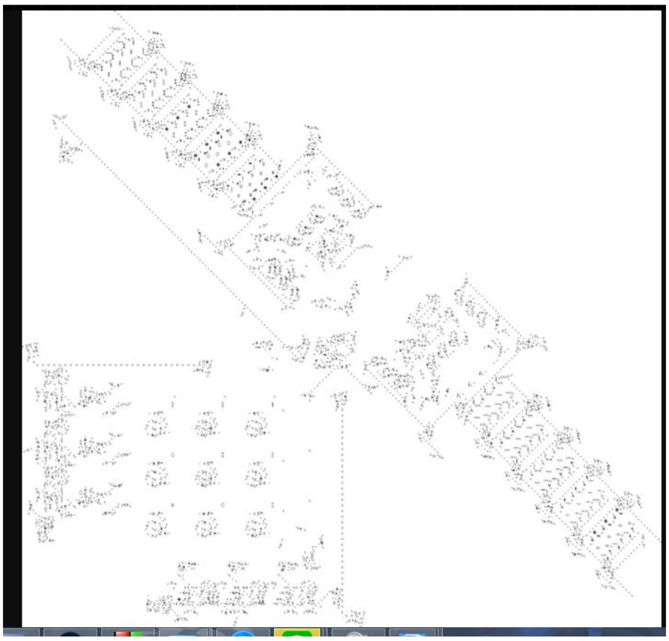


Figure 16 . Turing Machine in Life (http://rendell-attic.org/gol/tm.htm)

There are also people who made a Universal Computer using many complex parts that are previously founded like Minsky Register Machine sliding block memory and many more.

Life can even run itself by using a pattern that is called an OTCA Metapixel, a pattern that represents a pixel on a computer screen. With it, provided infinite computational power, Life can be run recursively forever.

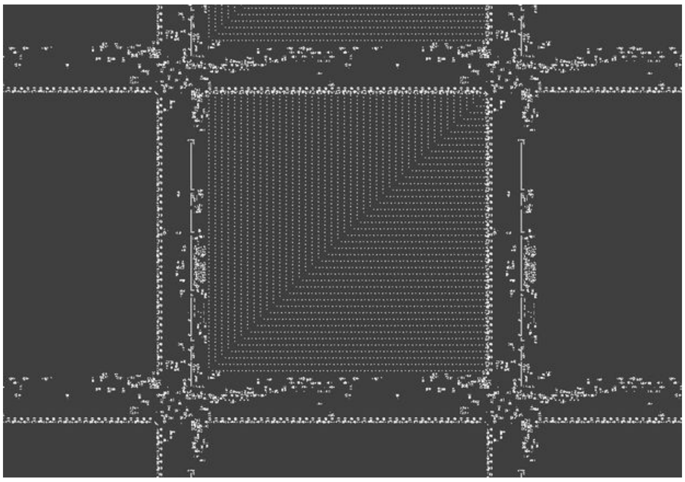


Figure 17 . An OTCA Metapixel.

VII. Acknowledgment

The author firstly would like to thank God for helping me give the power and will to finish this writing. I would like to also thank my parents for all the support they give so i can finish this paper

References

[1] https://web.archive.org/web/20090603015231/http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm (diakses pada tanggal 3 Desember 2017)

[2] http://web.stanford.edu/~cdebs/GameOfLife/ (diakses pada tanggal 3 Desember 2017)

[3] https://jeremykun.com/2011/06/30/turing-machines-and-conways-dreams/ (diakses pada tanggal 3 Desember 2017)

[4] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2016-2017/Aljabar-Boolean-2016.pdf (diakses pada tanggal 3 Desember 2017)

[5] https://jeremykun.com/2011/07/04/turing-machines-a-primer/ (diakses pada tanggal 4 Desember 2017)

[6] https://www.youtube.com/watch?v=vGWGeund3eA (diakses pada tanggal 4 Desember 2017)

[7] http://rendell-attic.org/gol/tm.htm (diakses pada tanggal 4 Desember 2017)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2017

Seldi Kurnia Trihardja 13516042