15th International Conference and School "Cellular Automata for Research and Industry"

ACRI2022

Tutorial Part II B: Selected Applications

Simulation of Belousov-Zabotinsky based logic circuits

12 Sept. 2022
University of Geneva, Switzerland

UWE Bristol | University of the West of England

Dr. Tsompanas Michail-Antisthenis
Unconventional Computing Laboratory,
Computer Science and Creative Technologies,
University of the West of England,
Bristol, United Kingdom

Presentation Outline

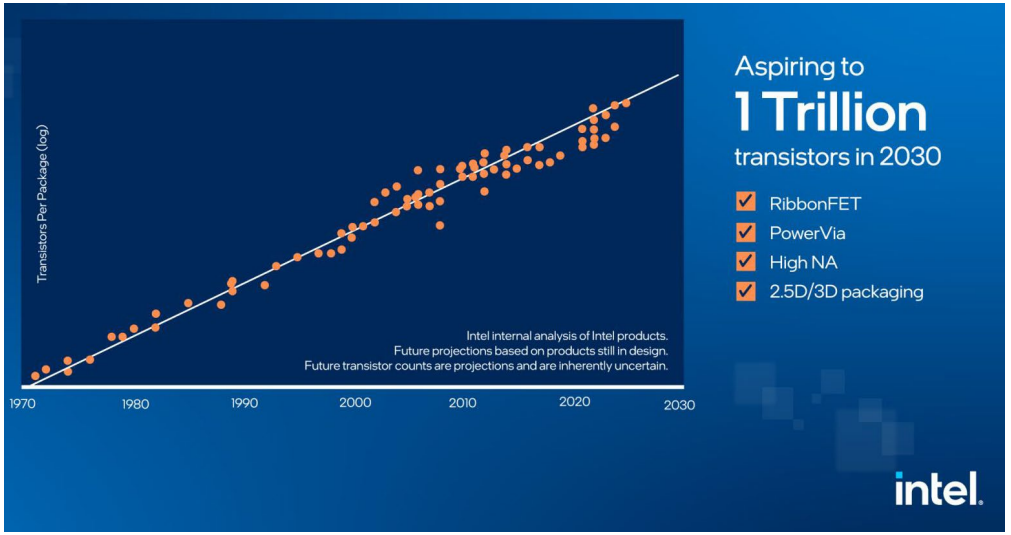
1. Motivation of the application
2. Previous works
3. CA methodology
4. First results
5. More experiments

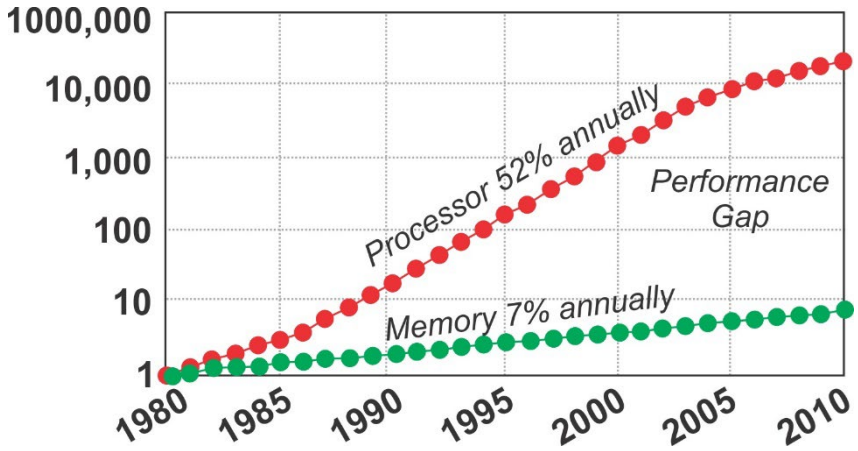
Are today's computers efficient?

- No.
- Yes.
- It depends.

Are today's computers efficient?

- ***No.***
  - Moore's Law
  - Memory wall
  - e.t.c.

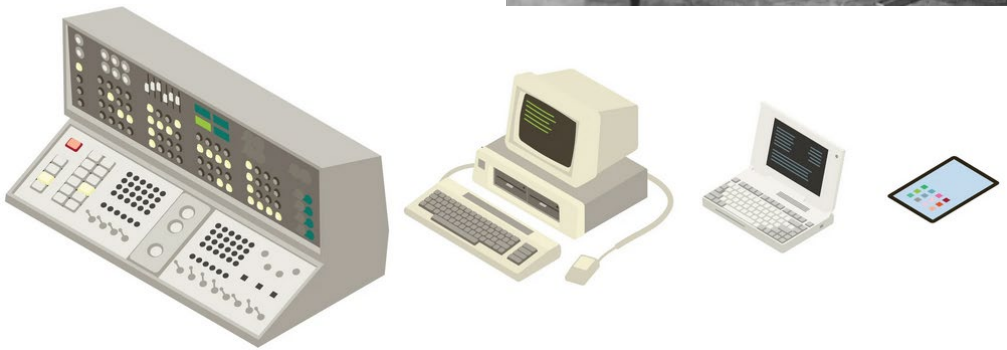




https://www.intel.com/content/www/us/en/newsroom/opinion/moore-law-now-and-in-the-future.html#gs.b5avkm
https://link.springer.com/referenceworkentry/10.1007/978-0-387-09766-4_234

Are today's computers efficient?

- ***Yes.***
  - When compared with previous generation ones.

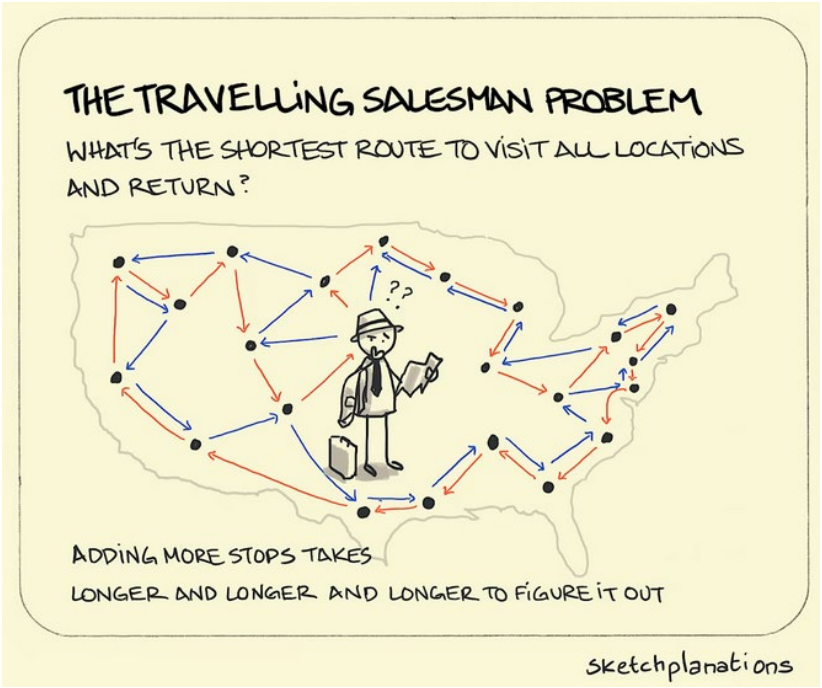


https://www.g2.com/articles/history-of-computers

Are today's computers efficient?

- *It depends.*
  - On what application/algorithm they execute.

Powers of 2	2^6	2^5	2^4	2^3	2^2	2^1	2^0
a = 27	0	0	1	1	0	1	1
a << 1 = 54	0	1	1	0	1	1	0



https://iq.opengenus.org/bitwise-division/
https://sketchplanations.com/the-travelling-salesman-problem
https://singularity2030.ch/wp-content/uploads/2019/03/Hauptbild-Essay-Nr.72-Stanford-University.jpg

Are there more efficient computing machines than silicon computers?

- ***It depends.***
  - On what application/algorithm they execute.
- ***Yes***
  - Graphene
  - Quantum computers
  - Memristors (a portmanteau of memory resistor)

Some types of unconventional computers

- Biological
- Chemical
- Physical
  - Other material than silicon

Introduction to liquid computing systems

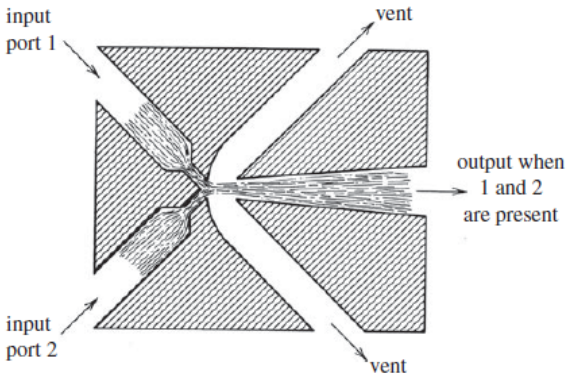
"A substrate does not have to be solid to compute. It is possible to make a computer purely from a liquid."

- Prof. Andrew Adamatzky

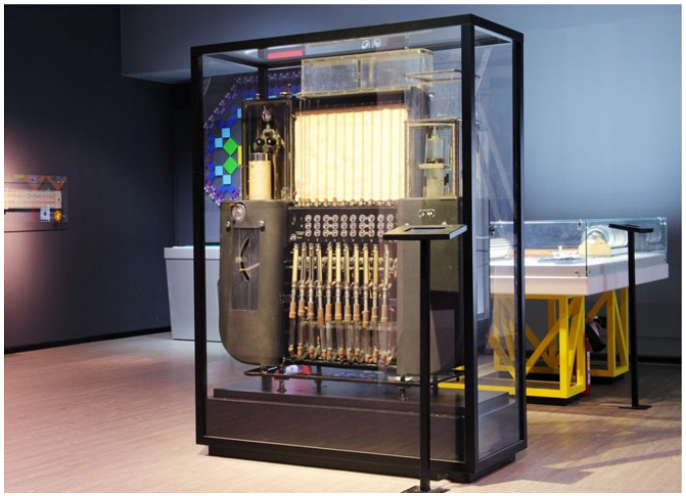
There is a plethora of experimental laboratory prototypes that prove this quote with different approaches. The most prominent categories can be defined as:

- a liquid carries signals,
- actuates mechanical computing devices and
- hosts chemical reactions (encoding signals).

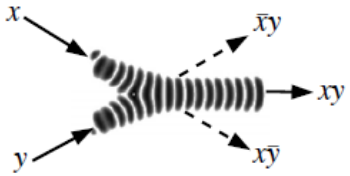
Fluidic (jet) gates



Lukyanov's hydraulic integrator

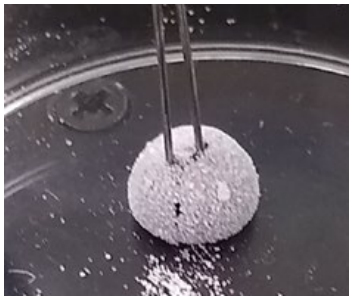


Belousov–Zhabotinsky reaction



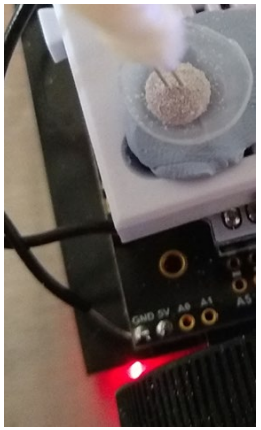
[1] Adamatzky, A. (2019). A brief history of liquid computers. *Philosophical Transactions of the Royal Society B*, *374* **(1774), 20180372.**

Previous works on BZ computing

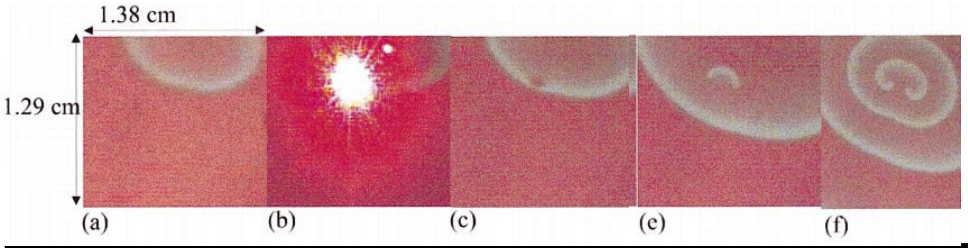


Belousov-Zhabotinsky (BZ) is a class of chemical reactions that due to their potential as nonlinear chemical oscillators, have inspired scientists to use them as chemical computers.

A thin-layer of BZ is regarded as capable of massively parallel operations, with each minuscule portion of its surface in a position to replicate a simplistic processor.



The system can be manipulated by mixing the medium to reset the system, apply light to change its properties or probe it with silver wires to initiate waves that represent the input data.

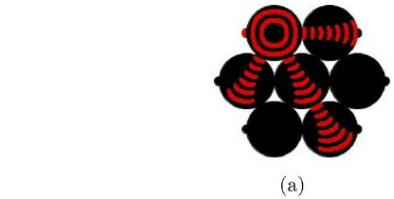


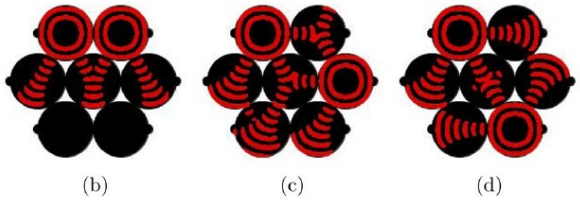
[1] Gentili, P. L., & Micheau, J. C. (2020). Light and chemical oscillations: Review and perspectives. Journal of Photochemistry and Photobiology C: Photochemistry Reviews, 43, 100321.
[2] Tsompanas, M. A., Fullarton, C., & Adamatzky, A. (2019). Belousov-Zhabotinsky liquid marbles in robot control. Sensors and Actuators B: Chemical, 295, 194-203.
[3] Tóth, R., Gáspár, V., Belmonte, A., O'Connell, M. C., Taylor, A., & Scott, S. K. (2000). Wave initiation in the ferroin-catalysed Belousov–Zhabotinsky reaction with visible light. Physical Chemistry Chemical Physics, 2(3), 413-416.
[4] Yokoi, H., Adamatzky, A., de Lacy Costello, B., & Melhuish, C. (2004). Excitable chemical medium controller for a robotic hand: closed-loop experiments. International Journal of Bifurcation and Chaos, 14(09), 3347-3354.

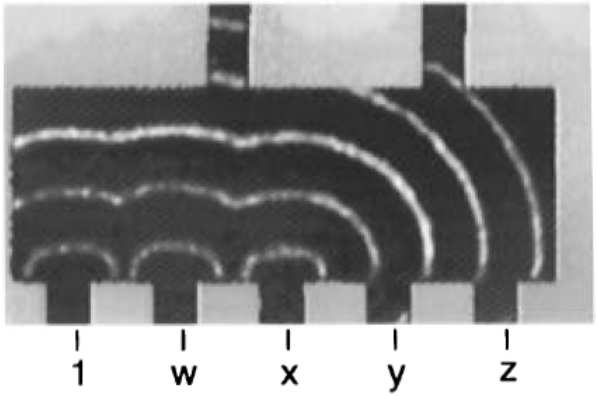
Previous works on BZ computing

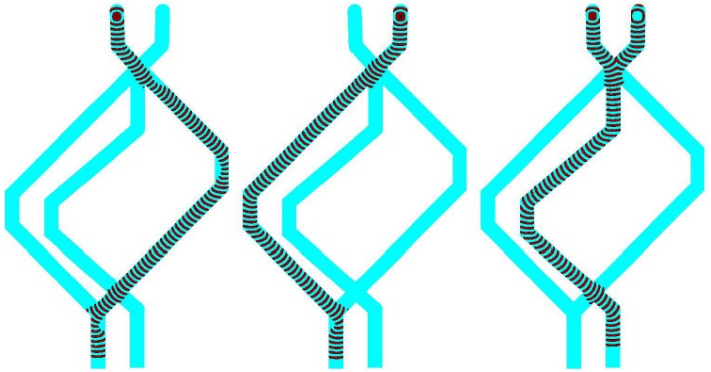
This chemical computational substrate has been utilized in several types of problems.

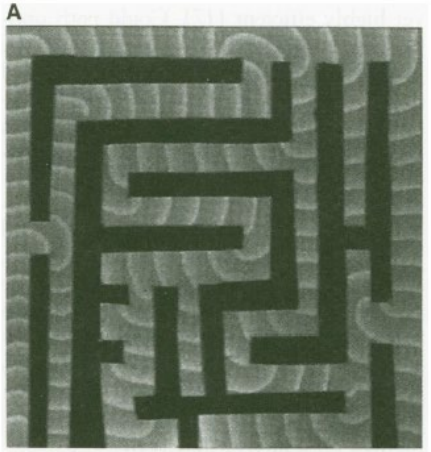
Namely, implementing basic Boolean logic gates [1], binary adders [2], solving labyrinths [3], analyzing transport infrastructure, implementing dataset classifier [4] and performing pattern recognition [5].

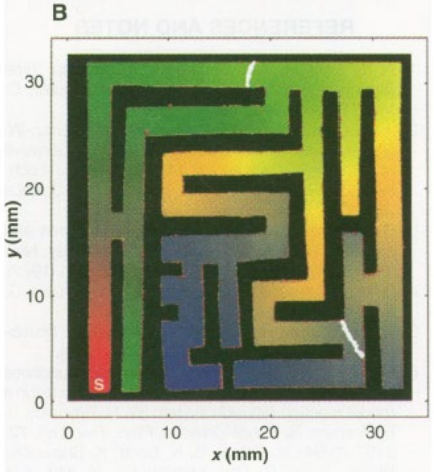










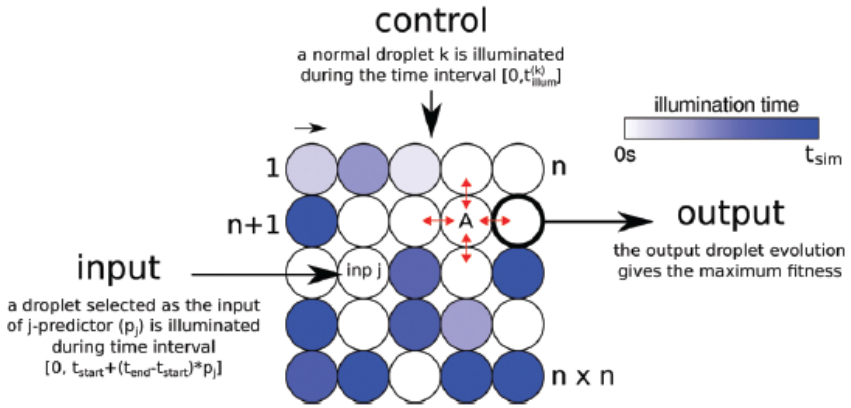


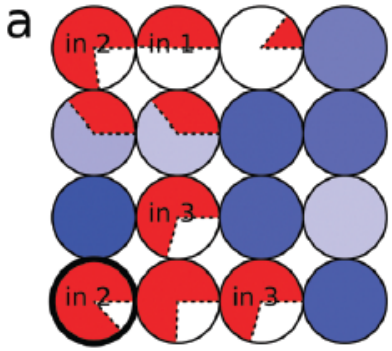
[1] Steinbock, O., Kettunen, P., & Showalter, K. (1996). Chemical wave logic gates. The Journal of Physical Chemistry, 100(49), 18970-18975.
[2] Adamatzky, A., Holley, J., Bull, L., & Costello, B. D. L. (2011). On computing in fine-grained compartmentalised Belousov–Zhabotinsky medium. Chaos, Solitons & Fractals, 44(10), 779-790.
[3] Steinbock, O., Tóth, Á., & Showalter, K. (1995). Navigating complex labyrinths: optimal paths from chemical waves. Science, 267(5199), 868-871.
[4] Gizynski, K., & Gorecki, J. (2017). Cancer classification with a network of chemical oscillators. Physical Chemistry Chemical Physics, 19(42), 28808-28819.
[5] Parrilla-Gutierrez, J. M., Sharma, A., Tsuda, S., Cooper, G. J., Aragon-Camarasa, G., Donkers, K., & Cronin, L. (2020). A programmable chemical computer with memory and pattern recognition. Nature communications, 11(1), 1-8.

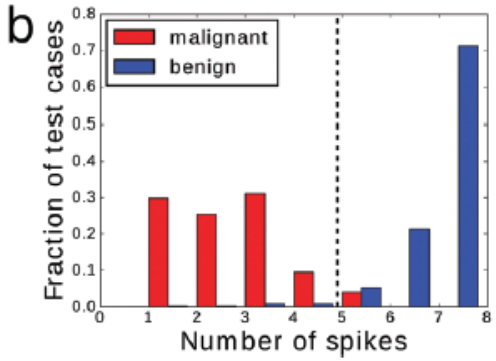
Previous works on BZ computing

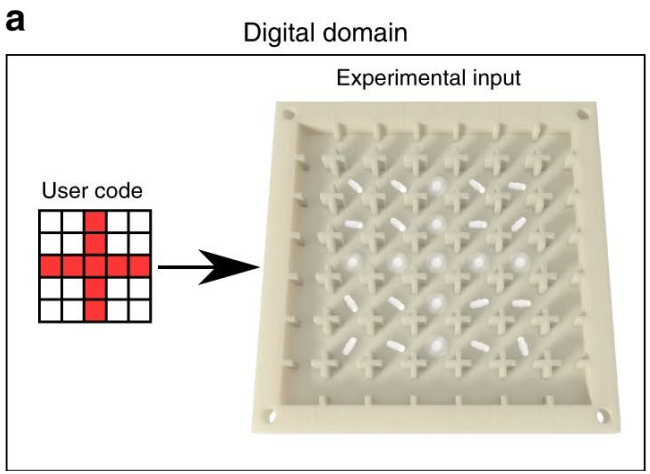
This chemical computational substrate has been utilized in several types of problems.

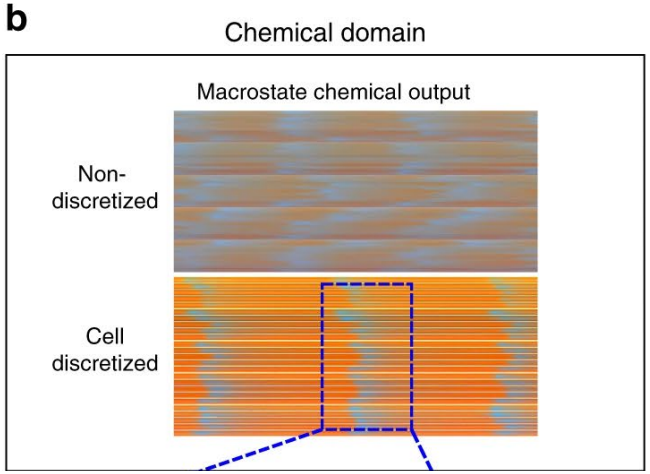
Namely, implementing basic Boolean logic gates [1], binary adders [2], solving labyrinths [3], analyzing transport infrastructure, implementing dataset classifier [4] and performing pattern recognition [5].

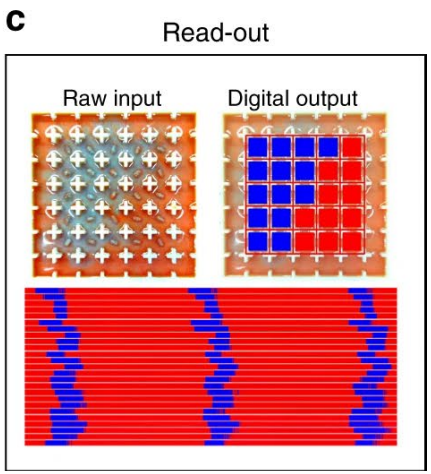












[1] Steinbock, O., Kettunen, P., & Showalter, K. (1996). Chemical wave logic gates. The Journal of Physical Chemistry, 100(49), 18970-18975.
[2] Adamatzky, A., Holley, J., Bull, L., & Costello, B. D. L. (2011). On computing in fine-grained compartmentalised Belousov–Zhabotinsky medium. Chaos, Solitons & Fractals, 44(10), 779-790.
[3] Steinbock, O., Tóth, Á., & Showalter, K. (1995). Navigating complex labyrinths: optimal paths from chemical waves. Science, 267(5199), 868-871.
[4] Gizynski, K., & Gorecki, J. (2017). Cancer classification with a network of chemical oscillators. Physical Chemistry Chemical Physics, 19(42), 28808-28819.
[5] Parrilla-Gutierrez, J. M., Sharma, A., Tsuda, S., Cooper, G. J., Aragon-Camarasa, G., Donkers, K., & Cronin, L. (2020). A programmable chemical computer with memory and pattern recognition. Nature communications, 11(1), 1-8.

What is the problem with designing chemical computers?

- Cost (Equipment / Reactants)
- Specialized knowledge
- Can be harmful
- Time constraints

UWE Bristol | University of the West of England

The alternative approach is….

- Simulation
  - Oregonator (PDEs)

$$\frac{du}{dt} = \frac{1}{\epsilon}\left(u - u^2 - (fv + \varphi)\frac{u - q}{u + q}\right) + D_u \nabla^2 u$$

$$\frac{dv}{dt} = u - v$$

  - **CAs**
    - 1st generation
    - 2nd generation
    - 3rd generation

UWE Bristol | University of the West of England

1st Gen CA simulating BZ (Excitable Media)

Weimar, J. R., Tyson, J. J., & Watson, L. T. (1992). Third generation cellular automaton for modeling excitable media. *Physica D: Nonlinear Phenomena*, *55*(3-4), 328-339.

- " […] the earliest model (1946) of excitable media was a cellular automaton introduced by Wiener and Rosenblueth [1]. The cellular automaton approach was pursued later by Moe et al. [2] Greenberg and Hastings [3] and others. These **'first generation models'** featured nearest neighbor connections and only a few possible states per cell. Although they exhibited wave propagation, they failed in several crucial aspects [4]:
  - The curvature effect […]
  - Dispersion relation […]
  - Spatial isotropy […] "

[1] Weiner, N., & Rosenblunth, A. (1946). The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements specifically in cardiac muscle Arch Inst Cardiol Mex. Jul;16(3):205-65.
[2] Moe, G. K., Rheinboldt, W. C., & Abildskov, J. A. (1964). A computer model of atrial fibrillation. *American heart journal*, *67*(2), 200-220
[3] Greenberg, J. M., & Hastings, S. P. (1978). Spatial patterns for discrete models of diffusion in excitable media. *SIAM Journal on Applied Mathematics*, *34*(3), 515-523.
[4] Winfree, A. T. (1987). *When Time Breaks Down. The 3-dimensional Dynamics of Electrochemical Waves and Cardiac Arrhytmias*. Princeton.

2nd Gen CA simulating BZ reactions

Weimar, J. R., Tyson, J. J., & Watson, L. T. (1992). Third generation cellular automaton for modeling excitable media. *Physica D: Nonlinear Phenomena*, *55*(3-4), 328-339.

- " These three shortcomings are addressed in 'second generation' models by Gerhardt, Schuster and Tyson [1], Markus and Hess [2] and Efimov and Fast [3].

To model curvature effects, Gerhardt, Schuster and Tyson introduce bigger neighborhoods: squares of size $(2r+1)^2$, where $r = 1, \ldots, 6$. In their automaton the number of excited cells within this square neighborhood is counted and a resting cell becomes excited if this count exceeds a certain threshold. […] Unfortunately, their automaton still exhibits some directional anisotropy. […] "

[1] Gerhardt, M., Schuster, H., & Tyson, J. J. (1990). A cellular automaton model of excitable media including curvature and dispersion. *Science*, *247*(4950), 1563-1566.
[2] Markus, M., & Hess, B. (1990). Isotropic cellular automaton for modelling excitable media. *Nature*, *347*(6288), 56-58.
[3] Fast, V. G., & Efimov, I. R. (1991). Stability of vortex rotation in an excitable cellular medium. *Physica D: Nonlinear Phenomena*, *49*(1-2), 75-81.

3rd Gen CA simulating BZ reactions

- Then, the 'third generation automaton' that combines the strengths and avoids the weaknesses of the earlier versions was proposed. The CA rules were closely related to the underlying PDEs. [1]

$$\text{Mask} = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 2 & 4 & 5 & 6 & 6 & 6 & 5 & 4 & 2 & 1 & 0 & 0 \\
0 & 1 & 2 & 4 & 6 & 8 & 9 & 10 & 9 & 8 & 6 & 4 & 2 & 1 & 0 \\
1 & 2 & 4 & 6 & 9 & 11 & 13 & 13 & 13 & 11 & 9 & 6 & 4 & 2 & 1 \\
1 & 3 & 5 & 8 & 11 & 14 & 16 & 17 & 16 & 14 & 11 & 8 & 5 & 3 & 1 \\
1 & 3 & 6 & 9 & 13 & 16 & 19 & 20 & 19 & 16 & 13 & 9 & 6 & 3 & 1 \\
1 & 3 & 6 & 10 & 13 & 17 & 20 & 21 & 20 & 17 & 13 & 10 & 6 & 3 & 1 \\
1 & 3 & 6 & 9 & 13 & 16 & 19 & 20 & 19 & 16 & 13 & 9 & 6 & 3 & 1 \\
1 & 3 & 5 & 8 & 11 & 14 & 16 & 17 & 16 & 14 & 11 & 8 & 5 & 3 & 1 \\
1 & 2 & 4 & 6 & 9 & 11 & 13 & 13 & 13 & 11 & 9 & 6 & 4 & 2 & 1 \\
0 & 1 & 2 & 4 & 6 & 8 & 9 & 10 & 9 & 8 & 6 & 4 & 2 & 1 & 0 \\
0 & 0 & 1 & 2 & 4 & 5 & 6 & 6 & 6 & 5 & 4 & 2 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Figure 1: *Effective weights in the mask diamond(2)∗square(3).*

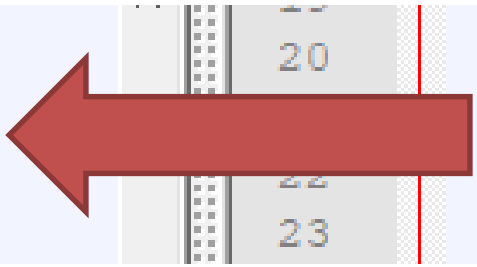
[1] Weimar, J. R., Tyson, J. J., & Watson, L. T. (1992). Third generation cellular automaton for modeling excitable media. *Physica D: Nonlinear Phenomena*, *55*(3-4), 328-339.

Cellular Automata simulating BZ reactions

- We can start with the *forest_fire.html* code.

- First we need to define the states of each cell.

```
const NEUTRAL = 0;
const EXCITED = 1;
const REFREACTORY = 2;
const WALL = -1;
```




```
const BURNING = -1;
const EMPTY = 0;
const TREE = 1;
```

Cellular Automata simulating BZ reactions

- We can update the initialization function to randomly seed the area with "excited" cells.

```
function init(x,y) {
if( Math.random() <= 0.9 ) {
        return NEUTRAL;
    } else {
        return EXCITED;
    }
}
```

- You can change the probability to get different initial states of the area.

Cellular Automata simulating BZ reactions

- The local rule needs to be updated as in the slide to reflect the behaviour of the chemical solution.

```
function Belousov(  ) {

return function( x, y, c ) {

    if( c[x][y] === WALL ) {
    return WALL;
    }

    if( c[x][y] === NEUTRAL ) {
    if( c[x-1][y] === EXCITED ||
        c[x+1][y] === EXCITED ||
        c[x][y-1] === EXCITED ||
        c[x][y+1] === EXCITED
      ){
        return EXCITED;
    } else {
    return NEUTRAL;
    }
    }

    if( c[x][y] === EXCITED ) {
    return REFREACTORY;
    }

    if( c[x][y] === REFREACTORY ) {
    return REFREACTORY;
    }

}
}
```

Cellular Automata simulating BZ reactions

- Then we can change the colouring function as we wish.

```
function color(state) {
    if( state === NEUTRAL )
        return "white";
if( state === EXCITED )
    return "red";
if( state === REFREACTORY )
    return "gray";
}
```

- And update the arguments in ***start()*** function.

```
start( context, cells, Belousov(), color, iter, delay );
```

» So, the result should look like: Example 1

Cellular Automata simulating BZ reactions

- The result is not that great, is it?
- So, lets update the rule and neighbourhood:

```
if( c[x][y] === NEUTRAL ) {
if( c[x-1][y] + c[x+1][y] + c[x][y-1] + c[x][y+1]+
    c[x-1][y+1] + c[x-1][y-1] + c[x+1][y-1] + c[x+1][y+1] > 2 ){
    return EXCITED;
} else {
return NEUTRAL;
}
}
```

» So, the result should look like: Example 2

UWE Bristol University of the West of England

Cellular Automata simulating BZ reactions

- If we increase the neighbourhood will we get better behaviour?
- So, lets try a neighbourhood with bigger radius:

```
if( c[x][y] === NEUTRAL ) {
if( c[x-1][y] + c[x+1][y] + c[x][y-1] + c[x][y+1] +
    c[x-2][y] + c[x+2][y] + c[x][y-2] + c[x][y+2]+
    c[x-1][y+1] + c[x-1][y-1] + c[x+1][y-1] + c[x+1][y+1] +
    c[x-2][y+1] + c[x-2][y-1] + c[x-2][y+2] + c[x-2][y+2]+
    c[x+2][y+1] + c[x+2][y-1] + c[x+2][y+2] + c[x+2][y+2]+
    c[x-1][y+2] + c[x-1][y-2] + c[x+1][y-2] + c[x+1][y+2] > 2 ){
    return EXCITED;
} else {
return NEUTRAL;
}
}
```

- **We need to update the library as well**

» So, the result should look like:

Example 3

UWE Bristol University of the West of England

Cellular Automata simulating BZ reactions

- **We need to update the library as well**

- https://github.com/Antisthenis/ACRI2022_Tutorial_B/blob/main/ca.js

» So, the result should look like: Example 3

UWE Bristol University of the West of England

Cellular Automata simulating BZ reactions

- If there is something wrong, check your rule-neighbourhood.

```
if( c[x][y] === NEUTRAL ) {
if( c[x-1][y] + c[x+1][y] + c[x][y-1] + c[x][y+1] +
    c[x-2][y] + c[x+2][y] + c[x][y-2] + c[x][y+2]+
    c[x-1][y+1] + c[x-1][y-1] + c[x+1][y-1] + c[x+1][y+1] +
    c[x-2][y+1] + c[x-2][y-1] + c[x-2][y+2] + c[x-2][y+2]+
    c[x+2][y+1] + c[x+2][y-1] + c[x+2][y+2] + c[x+2][y+2]+
    c[x-1][y+2] + c[x-1][y-2] + c[x+1][y-2] + c[x+1][y+2] > 2 ){
    return EXCITED;
} else {
return NEUTRAL;
}
}
```

Cellular Automata simulating BZ reactions

- Here is the correct one.

```
if( c[x][y] === NEUTRAL ) {
if( c[x-1][y] + c[x+1][y] + c[x][y-1] + c[x][y+1] +
    c[x-2][y] + c[x+2][y] + c[x][y-2] + c[x][y+2]+
    c[x-1][y+1] + c[x-1][y-1] + c[x+1][y-1] + c[x+1][y+1] +
    c[x-2][y+1] + c[x-2][y-1] + c[x-2][y-2] + c[x-2][y+2]+
    c[x+2][y+1] + c[x+2][y-1] + c[x+2][y-2] + c[x+2][y+2]+
    c[x-1][y+2] + c[x-1][y-2] + c[x+1][y-2] + c[x+1][y+2] > 2 ){
    return EXCITED;
} else {
return NEUTRAL;
}
}
```

» So, the result should look like:

Example 4

Cellular Automata simulating BZ reactions

- If we want we can initialize just one specific area.

```
function init(x,y) {
if( x > 15 && x < 21 && y > 15 && y < 21  ) {
        return EXCITED;
    } else {
        return NEUTRAL;
    }
}
```

» So, the result should look like:

Example 5

Cellular Automata simulating BZ reactions

- Now, we can build an architecture with walls to guide the waves and form a basic logic function.

- Let's start with the OR gate we saw earlier.
  - This is a synchronisation gate.

- Download and run BZ10.html
- (https://github.com/Antisthenis/ACRI2022_Tutorial_B)
- Then, in the web-browser click the ***Load*** button and load the file gate01.bmp

Example 6

Cellular Automata simulating BZ reactions

- We can change the gate's input by commenting and uncommenting the following as required.

```
if (( x > 16 && x < 18 && y > 57 && y < 68  ) ||  // Always HIGH input "1"
    ( x > 16 && x < 18 && y > 98 && y < 109  ) ||  // Input "x"
    ( x > 16 && x < 18 && y > 180 && y < 191  ) || // Input "y"
    ( x > 16 && x < 18 && y > 135 && y < 146  )){  // Input "z"
      return EXCITED;
    } else {
      return NEUTRAL;
    }
```

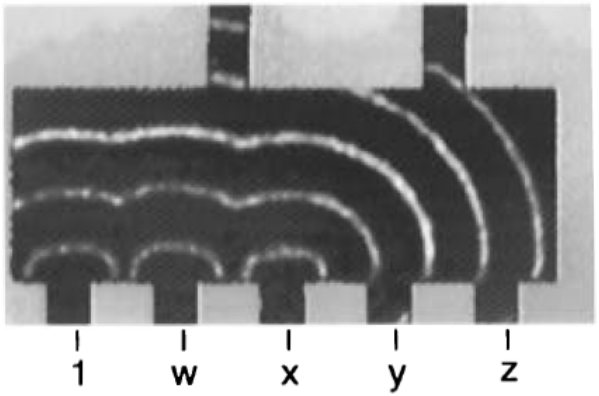
- Try out all inputs' combinations and verify the truth table of the OR gate.

Example 6

Cellular Automata simulating BZ reactions

- The we can use a different architecture of walls and channels that implement a more complex logic function.

- We will implement the AND[OR(w, x),OR(y, z)] gate we saw earlier.
  - This is again a synchronisation gate.



- Download and run BZ11.html
- Then, in the web-browser click the ***Load*** button and load the file gate02.bmp

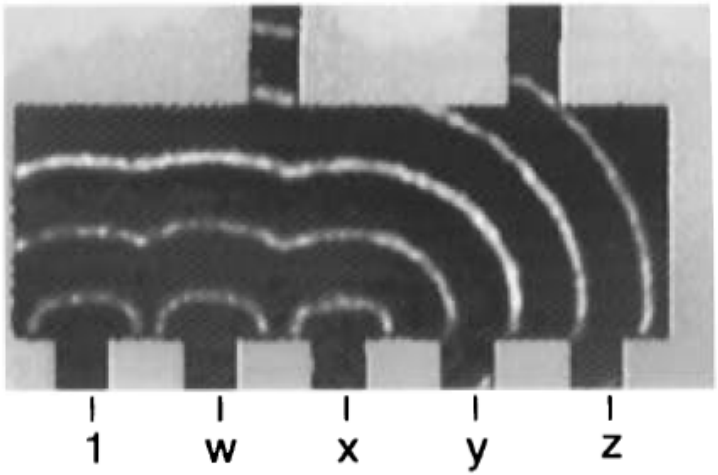
Example 7

Cellular Automata simulating BZ reactions

- We can change the gate's input by commenting and uncommenting the following as required.

```
if (( x > 60 && x < 80 && y > 330 && y < 332  )  ||  // Always HIGH input "1"
    ( x > 160 && x < 180 && y > 330 && y < 332  ) || // Input "w"
    ( x > 260 && x < 280 && y > 330 && y < 332  ) || // Input "x"
    ( x > 360 && x < 380 && y > 330 && y < 332  ) || // Input "y"
    ( x > 460 && x < 480 && y > 330 && y < 332  ) ){ // Input "z"
```

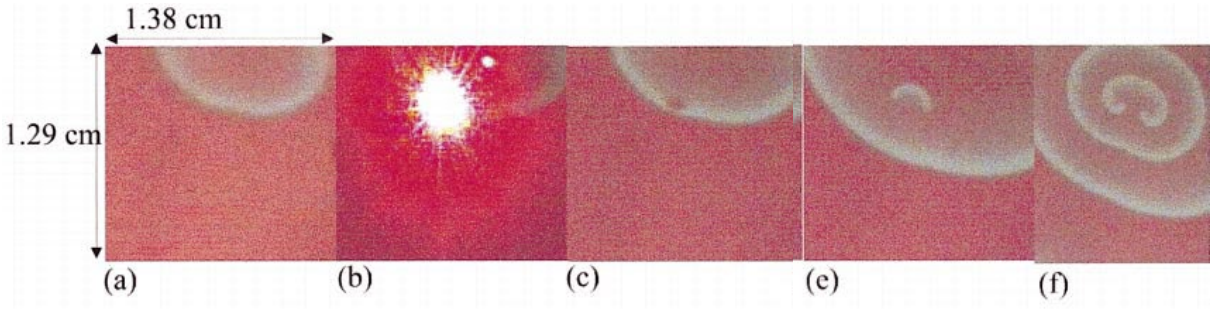
- Try out all inputs' combinations and verify the truth table of the gate.



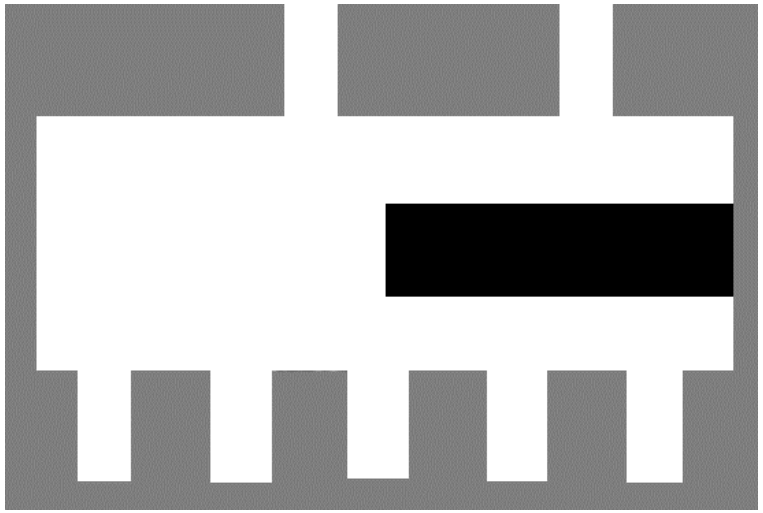
Example 7

Cellular Automata simulating BZ reactions

- As we saw previously light can alter the way the reactions propagate.



- Light masks can be used to extract a more complicated behaviour.



Cellular Automata simulating BZ reactions

- So, CAs' ability to implement inhomogeneties is utilized here:

```
if( c[x][y] === NEUTRAL ) {
var tempAmount = 1;
var tempValue = imageData[getColorIndicesForCoord(x, y, canvaslWidth)];
if ( tempValue < 20 ) { // That is Black color
    tempAmount = 10;
}
if( c[x-1][y] + c[x+1][y] + c[x][y-1] + c[x][y+1] +
    c[x-2][y] + c[x+2][y] + c[x][y-2] + c[x][y+2]+
    c[x-1][y+1] + c[x-1][y-1] + c[x+1][y-1] + c[x+1][y+1] +
    c[x-2][y+1] + c[x-2][y-1] + c[x-2][y-2] + c[x-2][y+2]+
    c[x+2][y+1] + c[x+2][y-1] + c[x+2][y-2] + c[x+2][y+2]+
    c[x-1][y+2] + c[x-1][y-2] + c[x+1][y-2] + c[x+1][y+2] > tempAmount ){
    return EXCITED;
} else {
return NEUTRAL;
}
```

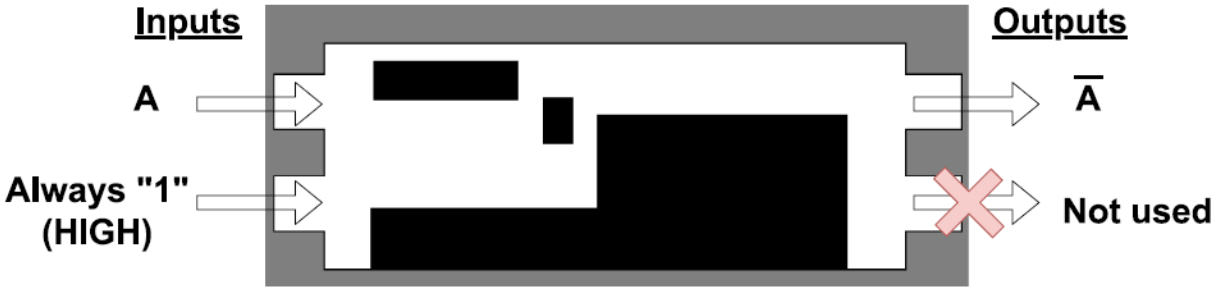
» Load the file gate02light.bmp:

Example 8

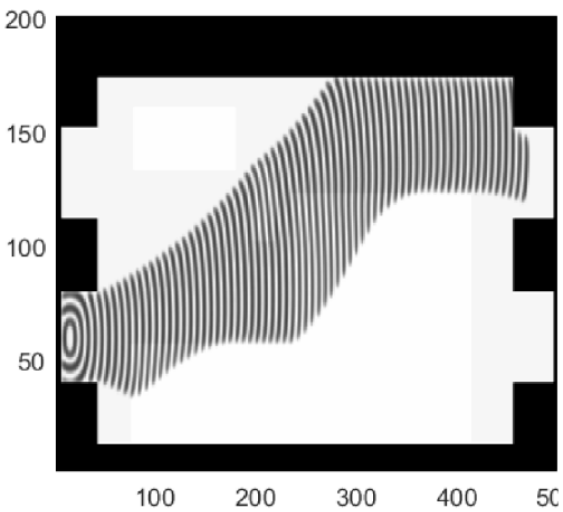
Results for *NOT* gate with Oregonator (PDEs)

The expected truth table is:

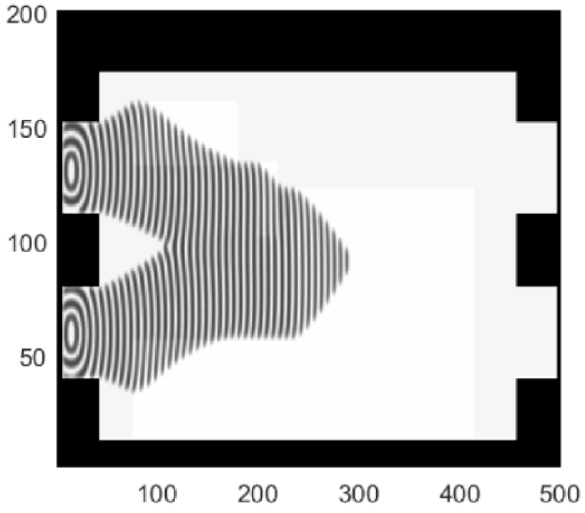
Input	Output
0 (not exist.)	1 (exist.)
1 (exist.)	0 (not exist.)



Input A = 0



Input A = 1



- Can we replicate this with the CA tool at hand?

» What do we need to change?

Example 9

3rd Gen CA simulating BZ reactions

- We can try to decrease the level of abstraction used to design the CA model rule. Thus, we need to check if using the 'third generation automaton' can provide a suitable simulation tool.

- Note that for this type of logic gates subexcitable media is required, so the threshold values for both rules need to be appropriately set.

- **Also, note that this is an open question and not tested yet. It might not work with the 3rd generation GA and we need to use the even better approximation of PDEs, that are worse in terms of efficiency.**

- **For more implementation details of the 3rd generation GA and the connection with PDE check the following work:**

UWE Bristol | University of the West of England

[1] Weimar, J. R., Tyson, J. J., & Watson, L. T. (1992). Third generation cellular automaton for modeling excitable media. *Physica D: Nonlinear Phenomena*, *55*(3-4), 328-339.

Thank you!

antisthenis.tsompanas@uwe.ac.uk

UWE Bristol | University of the West of England