A maximum-selector design in Codd-ICRA cellular automaton

By D. V. Takács

I. The Codd-ICRA cellular automaton

Introduction

The idea of the cellular automaton is created by John von Neumann [4], in 1948.

The cellular automaton is a set of identical, synchronized, finite, deterministic automata — called cells — connected by a homogeneous pattern. The cells directly connected with a certain cell, called neighbours of the latter one, and the state of a cell in a certain time step depends only on its own state and its neighbours' state in the last time step. The function, which governs this dependence, is called transition function.

In 1968 E. F. Codd published in his book *Cellular Automata* [1] a transition function in explicite form. This and his construction's techniques were our starting points in the Hungarian ICRA team since 1971.

We modified in some measure the Codd's transition function and techniques, keeping the main principles of the Codd-type cellular automaton. This new version is named Codd-ICRA cellular automaton.

The work of the ICRA team was motivated by practical purposes. Accordingly with the present technologies, it's not hopeless anymore the industrial producing such microelectronical circuits, which are the implementation of the Codd-ICRA cells. Therefore, one gets the possibility for constructing a new type of computer, constituted by cells only, in which the following advantages would be joined

— absolutely parallel working,
— hierarchiless,
— production homogeneity,
— flexible connection,
— flexible transformation,
— high speed,
— simple extensibility,
— absolute invertability of the software with the hardware,
— the manufacturing use of the selfreproducing ability.

Now let us introduce a bit formally the basic concepts of cellular automata. After having these, we show some Codd-ICRA techniques. Finally, we present a possible maximal number selector designed as a Codd-ICRA automaton.

Cellular automata and Codd-ICRA automaton

A *cellular automaton* consists of a collection of identical cells occupying a certain set of "places" I, and at each time step they change their states belonging to a finite set of possible states S, and this change takes place as a consequence of their mutual influences.

A cellular automaton, by definition, is a five-tuple $\mathfrak{A}=(I, L, v, S, \perp)$ where

$I\subseteq \mathbf{Z}^k$ is a subset of the points having integer coordinates in a k-dimensional Euklidian space. We shall associate a cell to each point i of I, and we shall identify it with i.

$L\subseteq \mathbf{Z}^k\ominus I$ is the set of dummy cells.

$v: I\longrightarrow (I\cup L)^n$ is the *neighbourhood function*, where n is the number of all neighbours of a single cell.

(I, L, v) is the socalled *cellular space*, which is homogeneous in the following sense; for every $i\in I$, $v(i)=\langle i+a_1, \dots, i+a_n\rangle$, where $a_1, \dots, a_n$ are fixed vectors. The cells, having dummy cell neighbour, namely the elements of the set P

$$P = \{i\in I|v(i)\notin I^n\},$$

we call *boundary cells.*

S is the finite set of the possible *states.*

$\perp: S\times S^n\longrightarrow S$ is called *local transition function.* If $i\in I$ and $s\in S$, then the cell, located at i, being in state s, with the neighbourhood $v(i)_1$ in state $s_1', \dots, v(i)_n$ in state s_n', will have the next state $\perp(s, \langle s_1', \dots, s_n'\rangle)$ denoted as

$$s\perp\langle s_1', \dots, s_n'\rangle.$$

If $s_0\in S$, and $s_0\perp\langle\underbrace{s_0, \dots, s_0}_{n\text{-times}}\rangle=s_0$, s_0 is called *quiescent state.*

$q: I\longrightarrow S$ is the *global state,* or *configuration* of the cellular automaton, which associates a state to each cell. The state of the cell $i\in I$, will be denoted by $q(i)$ or q_i, and the set of the global states by S^I.

$\lambda: L\longrightarrow S$. The mapping λ is the *input* of the cellular automata, which comes from the state of the dummy cells.

$\pi: P\longrightarrow S$ is the *output* of of the cellular automaton, which is a restriction of $q(P\subseteq I)$.

S^L: is the set of the inputs.

S^P: is the set of the outputs.

The generalization of the local transition function is the global *transition function,* which is the following mapping:

$\perp\!\!\perp: S^I\times S^L\longrightarrow S^I$ such that $q\perp\!\!\perp\lambda=q'$ iff

$$(\forall i\in I), q'(i) = q(i)\perp\!\!\perp\langle r(v(i)_1), \dots, r(v(i)_n)\rangle, \text{ where}$$

$$r(j) = \begin{cases} q(j), & \text{if } j\in I, \\ \lambda(j), & \text{if } j\in L. \end{cases}$$

The *global quiescent state* is q_0 for which

$$(\exists \lambda_0) q_0 \perp\!\!\perp \lambda_0 = q_0.$$

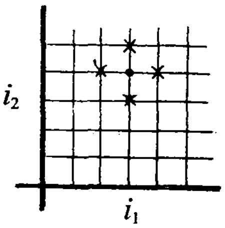


Fig. 1

The *Codd-ICRA automaton* $\mathfrak{A}=(I, L, v, S, \perp)$, is a special cellular automaton ([5], [6])[1], where $I \subset \mathbf{Z}^2$ ($\mathbf{Z}^2$ is the plane of integers), $S=\{0, 1, 2, 3, 4, 5, 6, 7\}$. For every $i=(i_1, i_2) \in I$, v is defined by $v(i_1, i_2)=\langle (i_1+1, i_2), (i_1, i_2-1), (i_1-1, i_2), (i_1, i_2+1)\rangle$ (see fig. 1).

Let ν be the mapping

ν: $S^4 \longrightarrow S^4$ such that $\nu(\langle s_1, s_2, s_3, s_4\rangle)$ is the smallest word under lexicographic ordering, what one can get from $\langle s_1, s_2, s_3, s_4\rangle$ by cyclic permutation: min lex $\{\langle s_1, s_2, s_3, s_4\rangle, \langle s_2, s_3, s_4, s_1\rangle, \langle s_3, s_4, s_1, s_2\rangle, \langle s_4, s_1, s_2, s_3\rangle\}$.

$\perp$ is defined for all $s \in S$ and $\langle s_1', s_2', s_3', s_4'\rangle \in S^4$ by

$$s \perp \langle s_1', s_2', s_3', s_4'\rangle =$$

$$=\begin{cases} s \quad \text{if} \quad \langle s\rangle \nu(s_1', s_2', s_3', s_4') \text{ does not appear in the Codd-ICRA transition function table,} \\ s \perp \nu(s_1', s_2', s_3', s_4'), \quad \text{if} \quad \langle s\rangle \nu(s_1', s_2', s_3', s_4') \text{ appears in the Codd-ICRA transition function table.} \end{cases}$$

How to use the transition function table? E.g. we are looking for the transition

$$0 \perp \langle 7212\rangle$$

we have to look instead of this the term

$$0 \perp \langle 1272\rangle$$

which appeares in the list and equal to 1, one can find it in the short form

$$0\ 1272\ 1$$

which means, that the next state of the central cell will be 1. Obviously, the $0 \perp \langle 4131\rangle$, instead of which we have to look $0 \perp \langle 1314\rangle$, and it doesn't occur, it means, the central cell in this environment has stable state. By regular notations the first example gives

$$0 \perp \langle 7212\rangle = 1,$$

the second one

$$0 \perp \langle 4131\rangle = 0.*$$

Codd-ICRA techniques

Basic operations. Let us make ourself a bit familiar with some Codd-ICRA techniques ([1], [3]).

The cells themselves are denoted by squares, their states with numbers written into them respectively.

* See [1] [7].

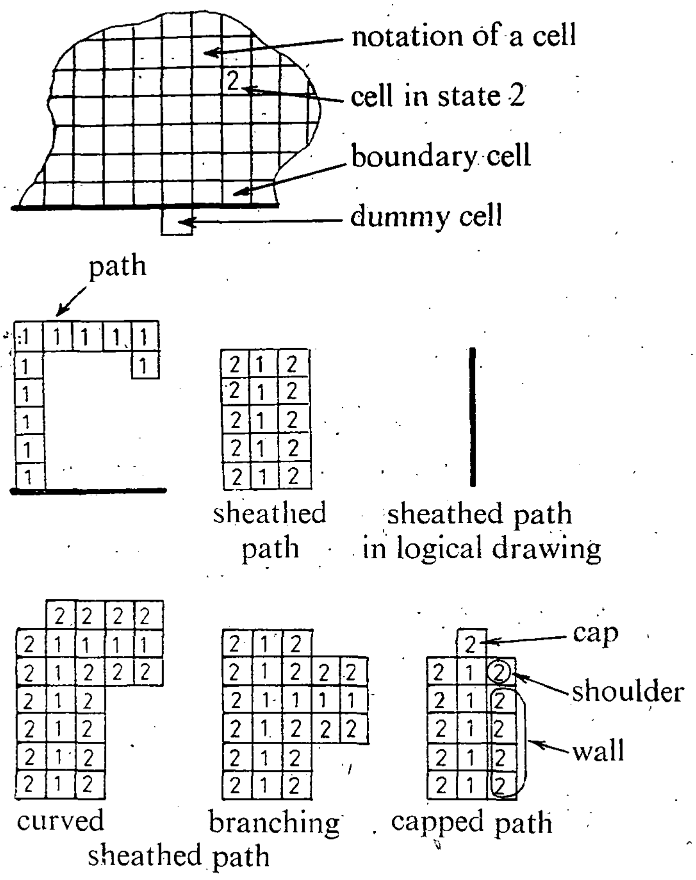


Fig. 2

We use very often phase figures from which one can see the situations in time steps $t=1, \ldots$. For instance we have the following configuration at time step one. The blank place means that the corresponding cell is in 0 state. We have to look in the transition table all the instances which appear on the figure. In the upper left corner of fig. 3 the cell is in state 2 and its neighbourhood is $2 \perp \langle 0021 \rangle$, which is equal to 2 (200212 doesn't appear in the table). And so on, we find three terms only, which show state changing, namely

012621,

602120,

112626,

so we have the following phase figures.

Along the sheathed paths the states' pair is propagating if this is in form OS, where $S=4, 5, 6, 7$. We call this OS couple of the cells signal. The signal propagation is basic behavior of cellular automaton. Since the signal is propagating cell by cell, so the time required for the propagation is strictly proportional to the distance which is made by it expressed with the number of the cells. In a fixed automaton this factor is constant. Rather often, with inaccurate speaking we say that the distance, which is made is equal to the necessary time.

2	2	2	2	2
1	0	6	1	1
2	2	2	2	2

Fig. 3

	0	
0	2	2
	1	

Fig. 4

2	2	2	2	2
1	0	6	1	1
2	2	2	2	2

$t=1$

2	2	2	2	2
1	1	0	6	1
2	2	2	2	2

$t=2$

Fig. 5

2	2	2	2	2	2	2	2	2
1	0	7	1	1	1	6	0	1
2	2	2	2	2	2	2	2	2

$t=1$

2	2	2	2	2	2	2	2	2
1	1	0	7	1	6	0	1	1
2	2	2	2	2	2	2	2	2

$t=2$

2	2	2	2	2	2	2	2	2
1	1	1	0	1	0	1	1	1
2	2	2	2	2	2	2	2	2

$t=3$

2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2

$t=4$

Fig. 6

Along a path collision of signals can happen. Fig. 6 shows such a case. The effect of several collisions depends on the parity of the cells' number is state 1, between two signals, and depends on the signal's content ($S=4$ or 5 or 6 or 7). The result is always the annihilation of the signals, except with the odd parity, homogenous case, where

$$04\times04=05,$$
$$05\times05=06,$$
$$06\times06=06,$$
$$07\times07=04.$$

Fig. 6 gives an example for odd parity, heterogenous collision, fig. 7 an odd, homogenous one.

In the next part we will show, that the original Codd's operations are valid for the Codd-ICRA constructions too. We connect three dummy cells, to three neighbouring boundary cells of the empty cellular automaton. (Empty: it contains cells in state 0 only.) The two extremal cells have state 2 constantly, and we introduce into the middle one the sequences of the Codd-ICRA programmes.

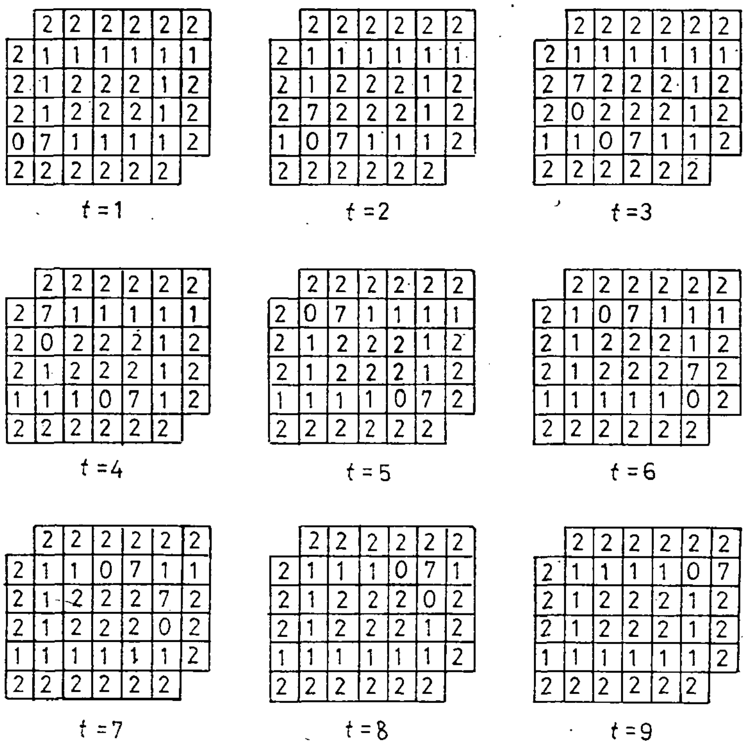


Fig. 7

The *Codd-ICRA programme* is a word constituted by cells' states: 0, 1, 4, 5 and 7, as elements, where each 1 is followed by 1, 4, 5, 6 or 7; the 4, 5, 6 or 7 are followed by one 0, and after a 0 there is at least one 1. With other words, the set of the programmes is identical to an arbitrary walking on the following graph on fig. 8. E.g. the word ⟨1, 1, 1, 5, 0, 1, 1, 6, 0, 1, 6, 0, 1⟩ is a programme. Firstly, we show a *pathway's embryo producing* programme, in order to be able to enter into the empty automaton.

It's the following: ⟨6, 0, 1, 7, 0, 1, 6, 0, 1⟩.

The effect of the programme is shown by fig. 9.

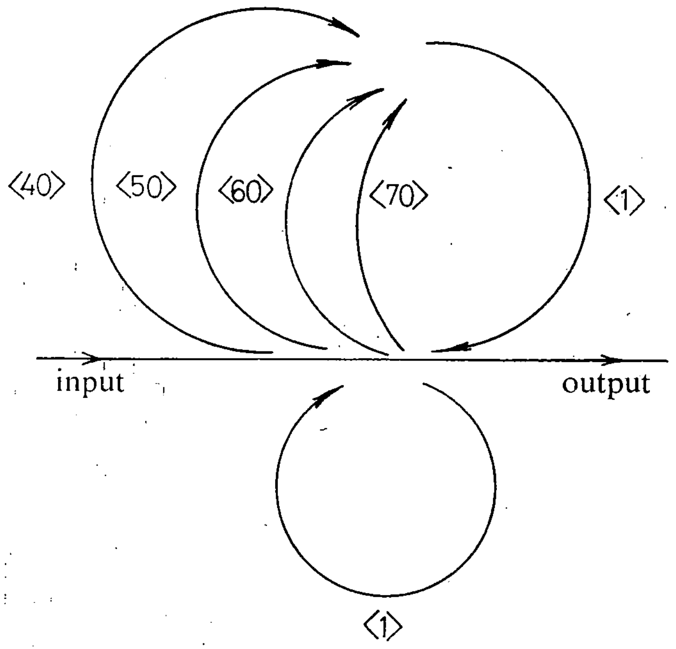


Fig. 8

The three cells under the double line are dummy cells

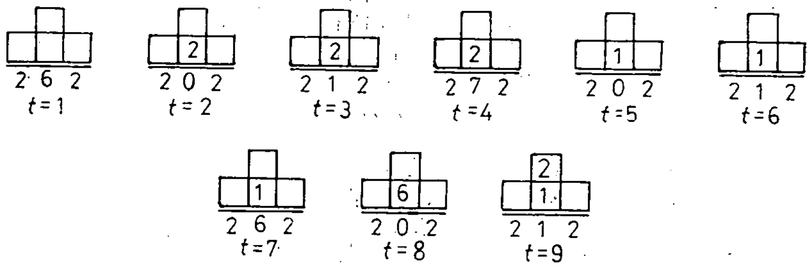


Fig. 9

The effect of the pathway embryo producing programme

The main functions of the pathway which are the Codd's operations.

Extend ⟨7, 0, 1, 6, 0, 1⟩.

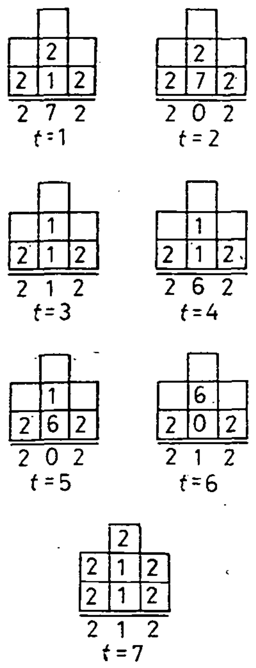


Fig. 10

Extend left ⟨4, 0, 1, 4, 0, 1, 5, 0, 1, 6, 0, 1⟩.

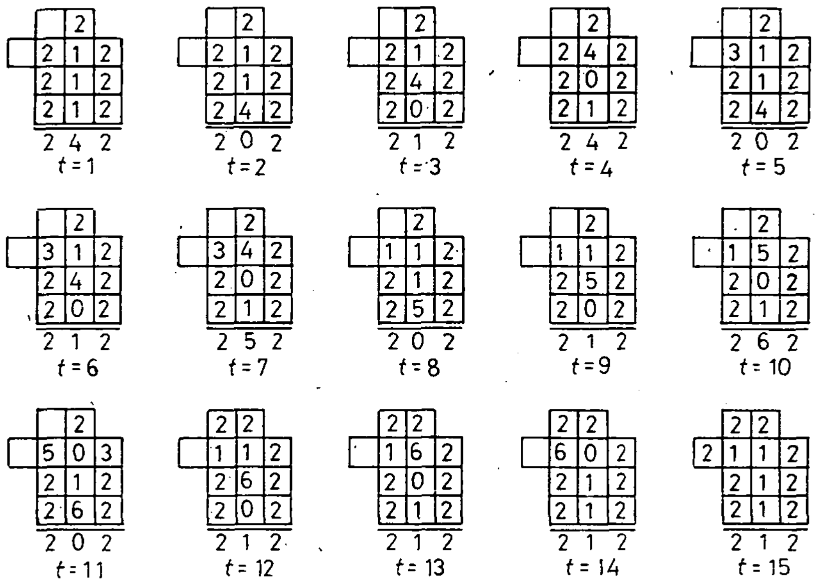


Fig. 11

Extend right ⟨5, 0, 1, 5, 0, 1, 4, 0, 1, 6, 0, 1⟩.

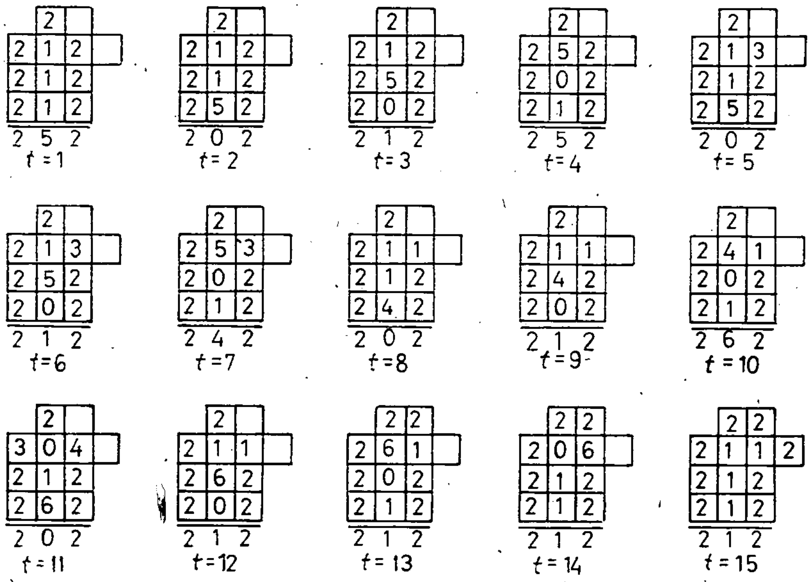


Fig. 12

Retract ⟨4, 0, 1, 5, 0, 1, 6, 0, 1, 6, 0, 1⟩.

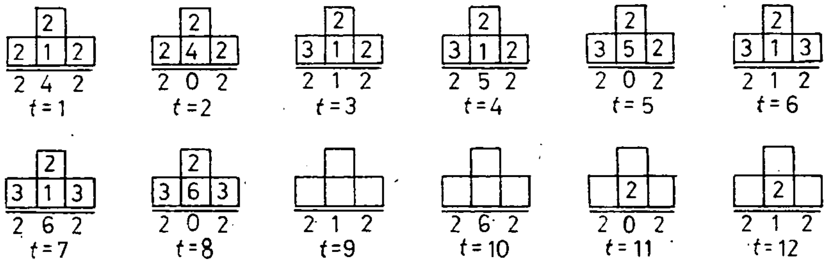


Fig. 13

Retract left ⟨5, 0, 1, 6, 0, 1, 6, 0, 1, 6, 0, 1⟩.

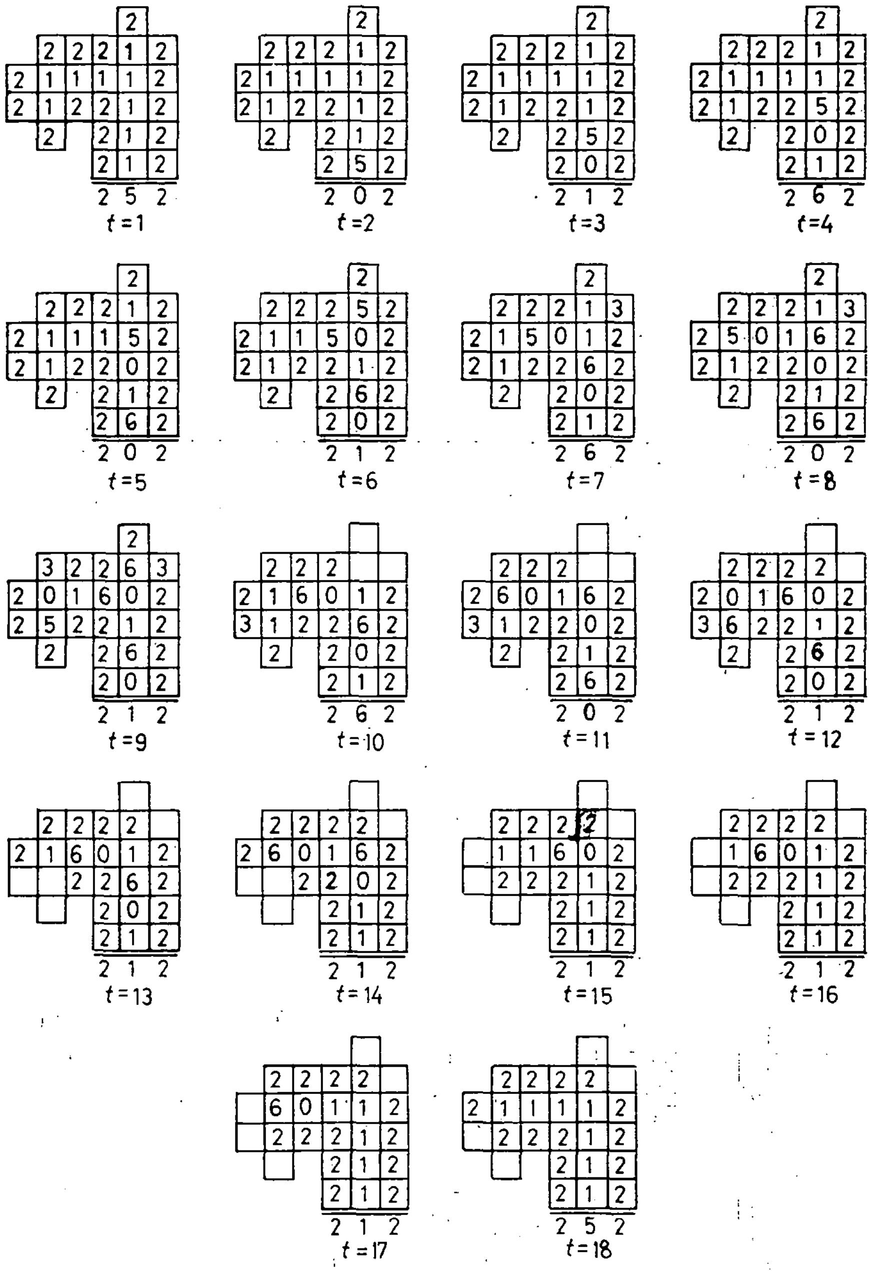


Fig. 14

Retract right ⟨4, 0, 1, 6, 0, 1, 6, 0, 1, 6, 0, 1⟩.

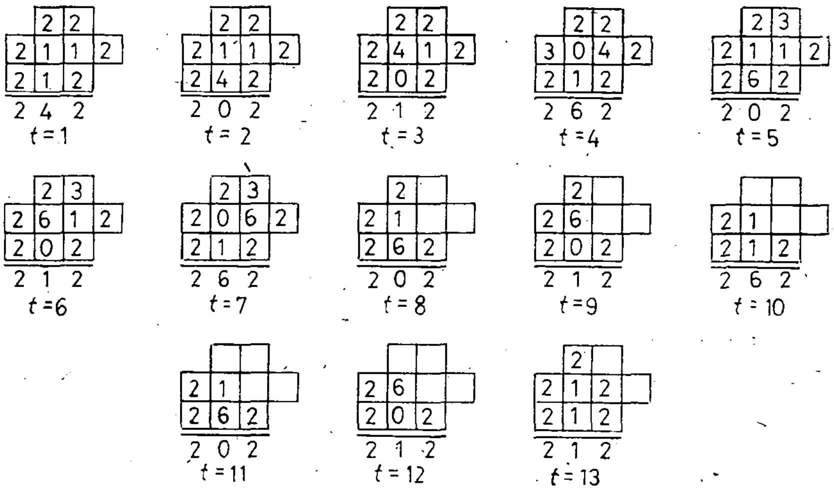


Fig. 15

Sheathing ⟨6, 0, 1⟩.

Sheathed, capped path made by the sheathing signal

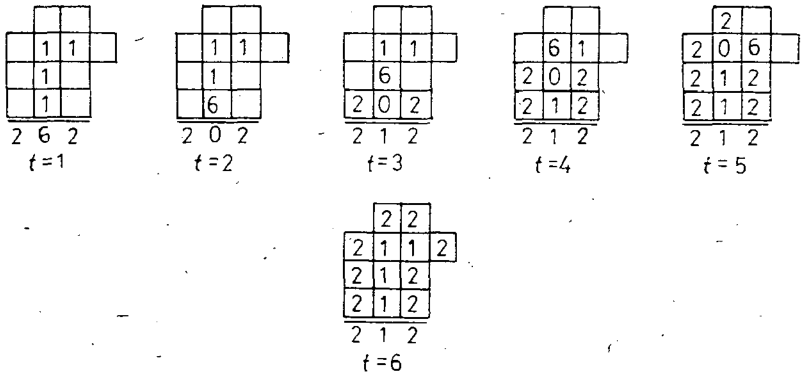


Fig. 16

Activation ⟨7, 0, 1⟩.

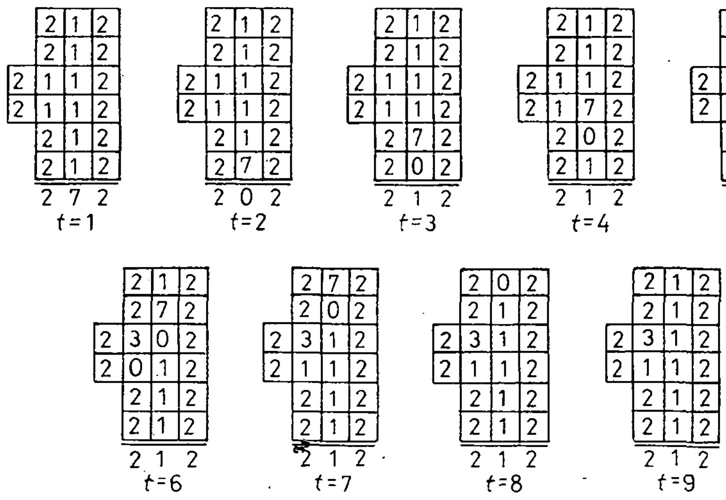


Fig. 17

The activating signal changes the gates' and transformer' (see later) suitable cells being in state 0 or 1 to state 3. These tools become ready for working by this way.

Mark ⟨7, 0, 1, 6, 0, 1, 4, 0, 1, 5, 0, 1, 7, 0, 1, 6, 0, 1⟩.

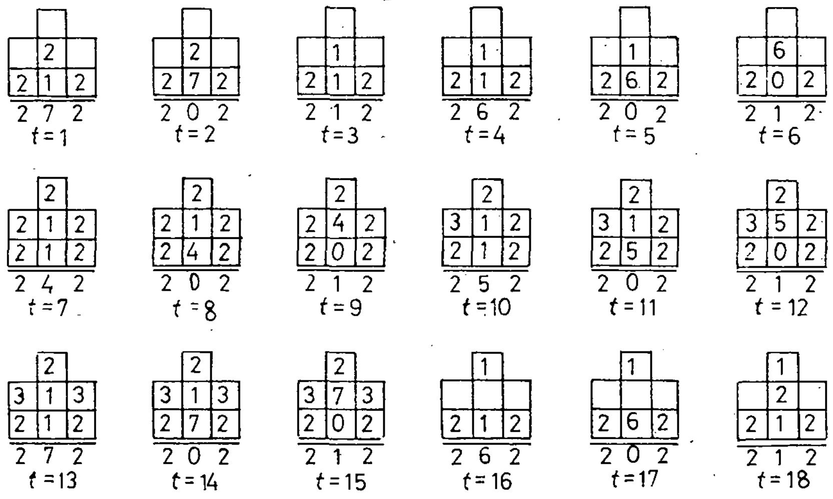


Fig. 18

Erase ⟨6, 0, 1, 7, 0, 1, 4, 0, 1, 5, 0, 1, 6, 0, 1, 6, 0, 1⟩.

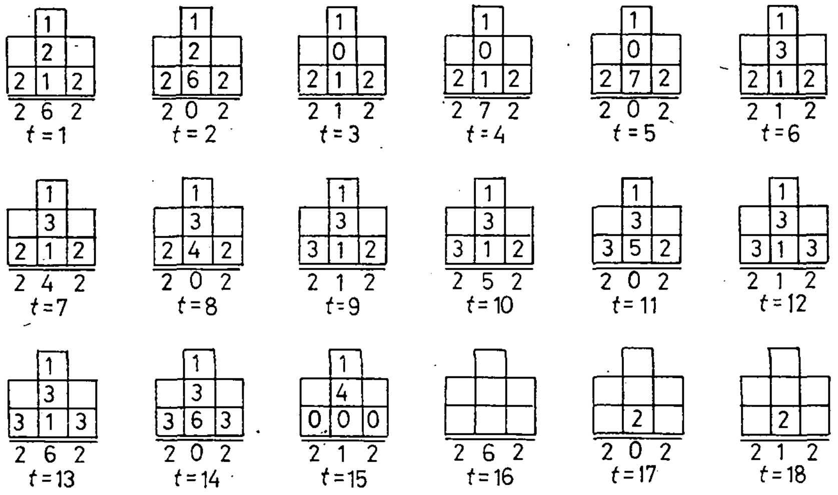


Fig. 19

Sense ⟨7, 0, 1, 7, 0, 1⟩

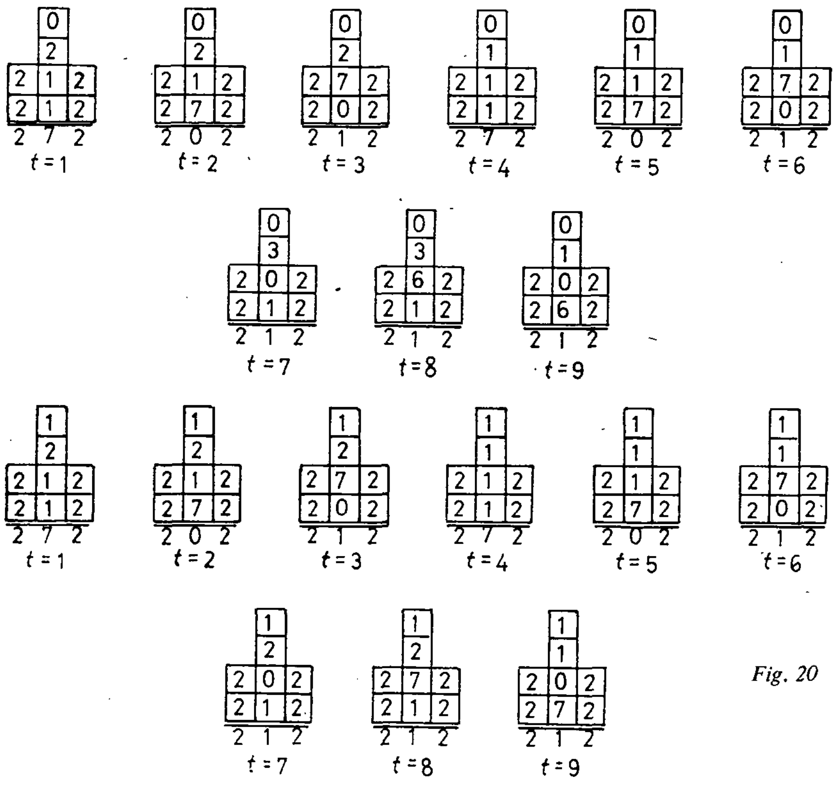


Fig. 20

Cap after sense ⟨4, 0, 1, 6, 0, 1⟩.

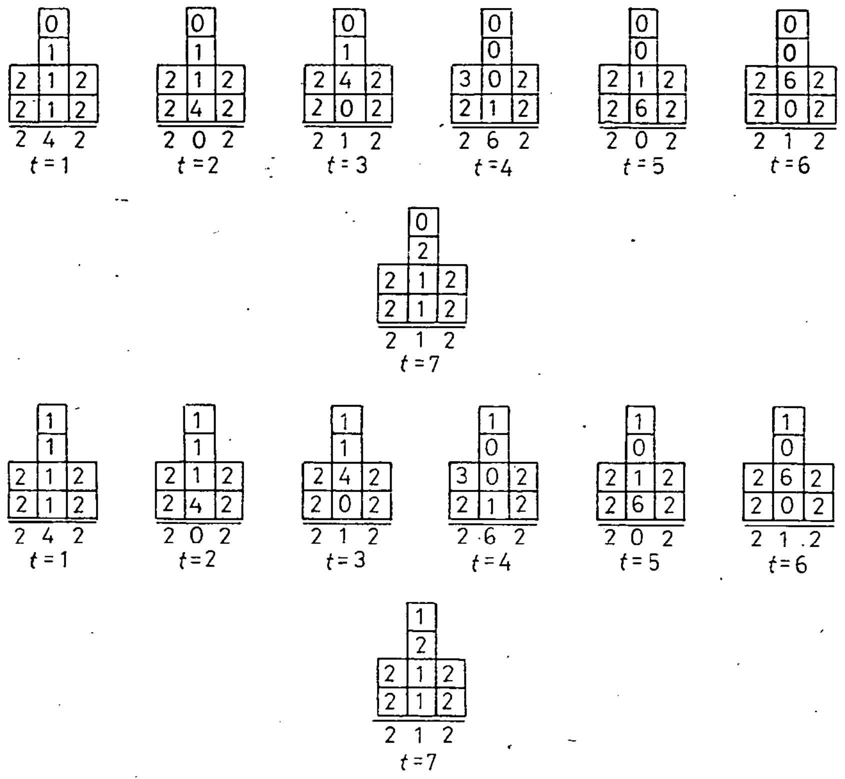
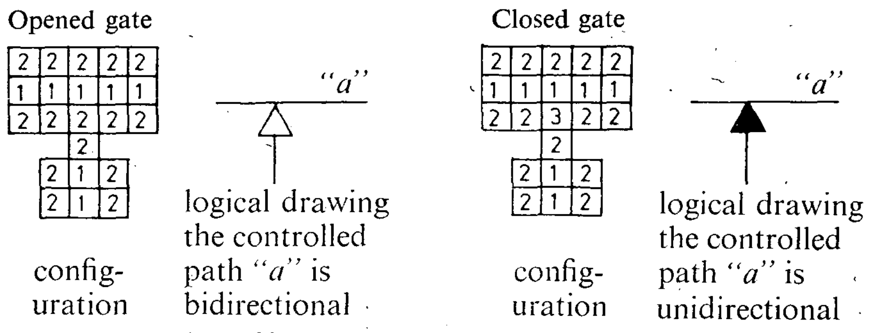


Fig. 21

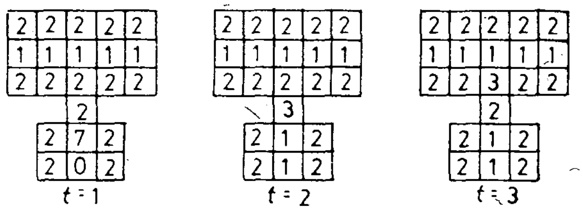
We have seen, that it's possible to develop directly sheathed paths in the empty cellular space by means of programmes, introduced into the boundary cells from the dummy cells. Why we develop in this case, hovewer, unsheathed paths, making the sheating afterwards? One has to do this, because branching and looping sheathed paths are not programmable directly. So, one has to prepare firstly a simple formed sheathed path, and with this, used it as a writing arm, is possible to develop the required network by the repeated application the mark and retract operation. Finally the stacking of the network is followed by the sheathing and activation.

Some components. Configurations constituted from a few cells which are able to make some prescribed effect on signals propagating along pathways, called components. These effects: to close in one direction the way, the signals' transformation, protection of the paths in the case crossover, and creating signal sources.

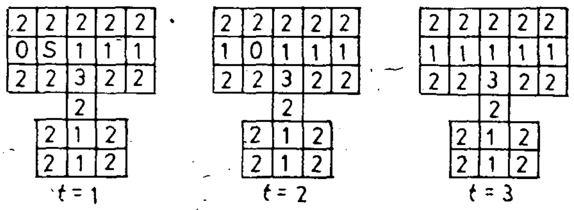
One of the most important properties in our cell space is gating. We speak about gating if the signal can propagate in one direction along its path but in the other direction the signals are annihilated at a point. The component doing this annihilation is called gate.


Opened gate
"a"
logical drawing the controlled path "a" is bidirectional
config-uration
Closed gate
"a"
logical drawing the controlled path "a" is unidirectional
config-uration

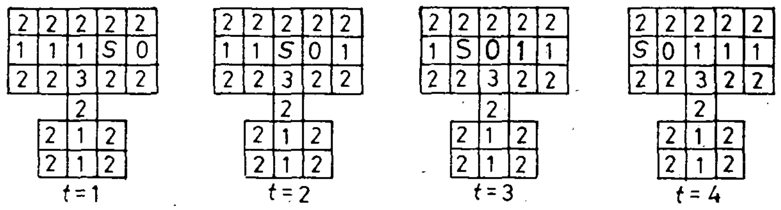
Gate activation by signal 70



The annihilation of signals SO coming from leftside; S = 4,5,6,7



The propagation of signals SO coming from rightside



Gate desactivation by signal 70 or 60

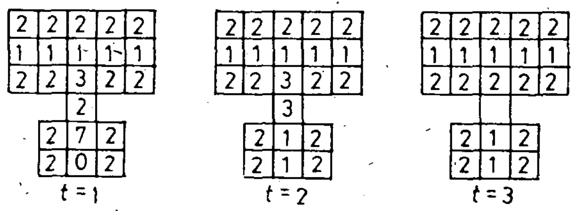


Fig. 22

Codd type remote controlled gate

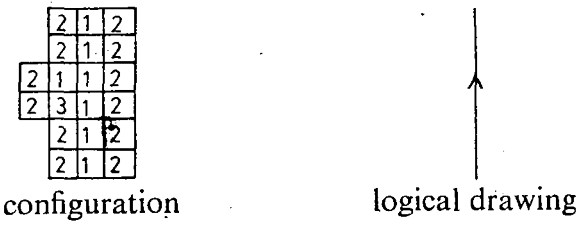
Codd-type remote controlled gate. On fig. 22 one can see the activation of this gate, as well as the phenomenon, which in closed state annihilates the signal coming from leftside, but permits from rightside. The opening or dezactivation of the gate can happen by signals 60 or 70. This gate remembers. Its state depends not only on the control signal but on its last own state too. Thus, one can call it bistable gate.

Local gate or valve. On fig. 23 one can see how the local gate is functioning. This is built into the sheathing wall during the construction of the controlled pathway, and works at any time.

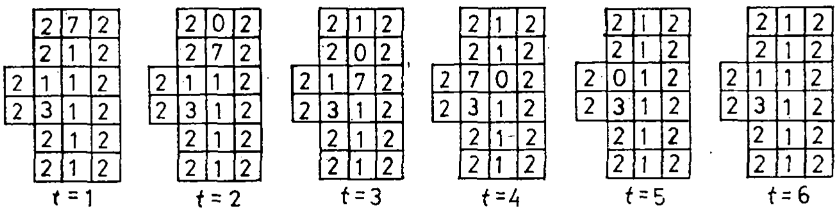
The location of the local gate can be along straight or branched way. The latter case is important in the control of sophisticated networks [2].

Gates combinations. By means of gates' combinations, one can build a lot of tools for pathway controls. Such tools are shown by fig. 24. Let's see only one example

Local gate for signals 60 and 70 along straight path



The annihilation of the signals 70 and 60 coming from leftside



The propagation of the signals 70 and 60 coming from rightside

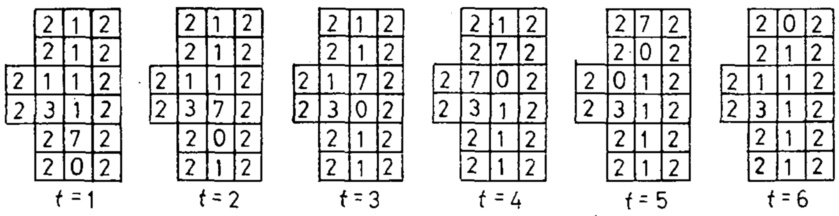
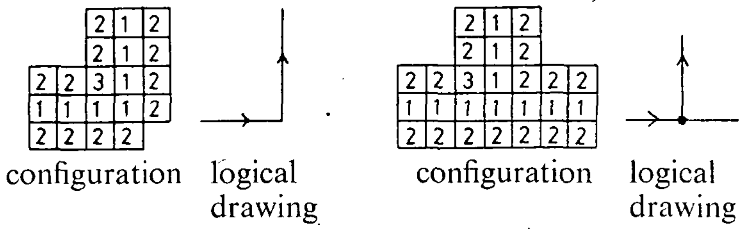
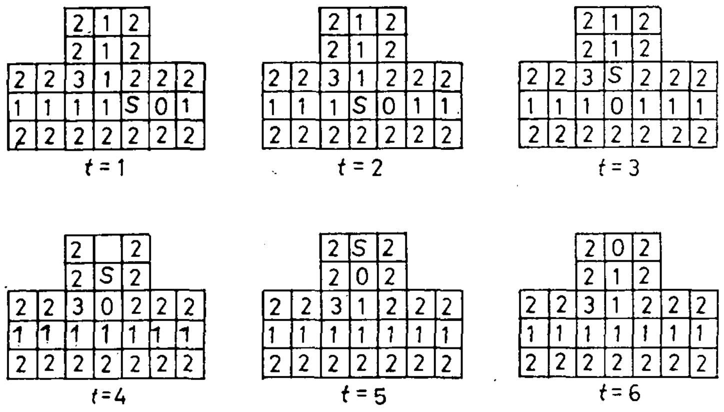


Fig. 23a

Rectifier gate in corner and junction



The annihilation of signals SO coming from leftside; S = 4,5,6,7



The propagation of signals coming from rightside

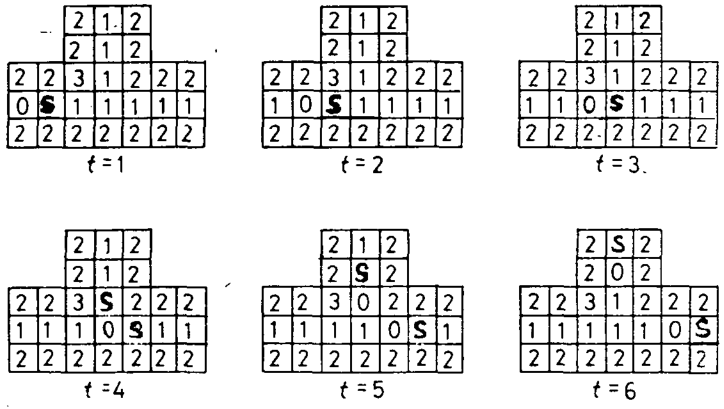


Fig. 23/b
Local gates or valves

in details, the socalled monostable gate. This is a Codd-type remote controlled gate which has in its control path a duplicating loop. The first sample of the duplicated signal closes the normally opened gate for 4 tacts only because the second sample of the signal opens it again [3] (see fig. 25).

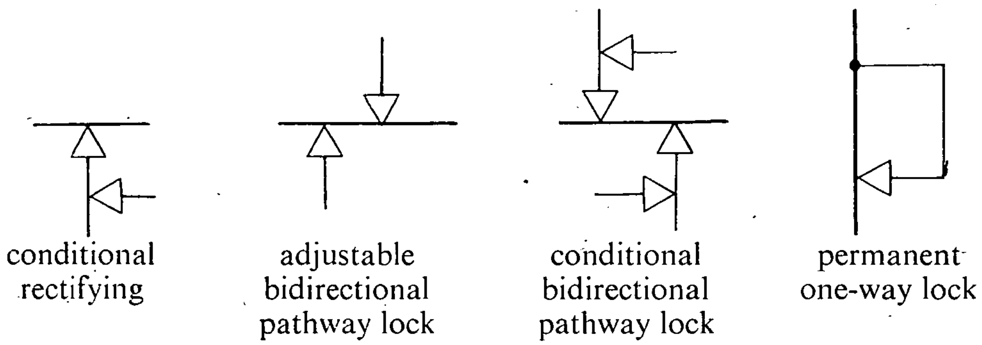

conditional rectifying
adjustable bidirectional pathway lock
conditional bidirectional pathway lock
permanent one-way lock

Fig. 24
Some gate combinations

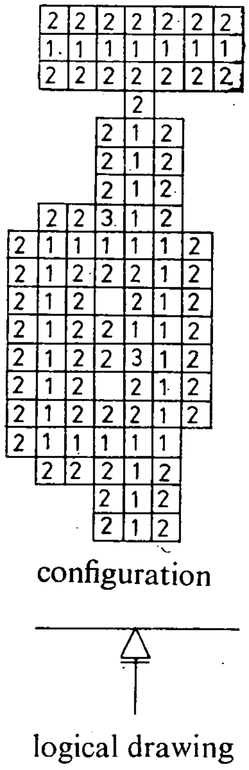
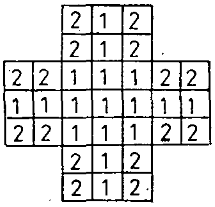


Fig. 25'
Monostable gate

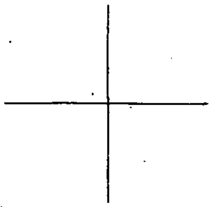
Cross-over. It is a hard problem how to cross the paths in the plane (in two dimensions). Without any protection obviously false signals propagate from one way into the other at the cross point. Since this is one of the favourite subject in the cellular automata litterature the publication of new crossover solutions, usually designed with a let of roundabout ways. E.g. Codd nimself, in his book made a solution, concerning two one-ways, which requires about 1500 cells. Therefore we suppose the result by Dettai [2] is significant giving a crossover which requires 9 cells only for two bidirectional ways, but only for 60 and 70 signals (see fig. 26). The minimal following distance between two signals is 6 [see 2].

6—7 transformer. On fig. 27 one can see a transformer which is usable in one direction only for 60 and 70 signals. For both inputs 60 and 70 the output is 70 [see 3].

Crossover for signals 60 and 70

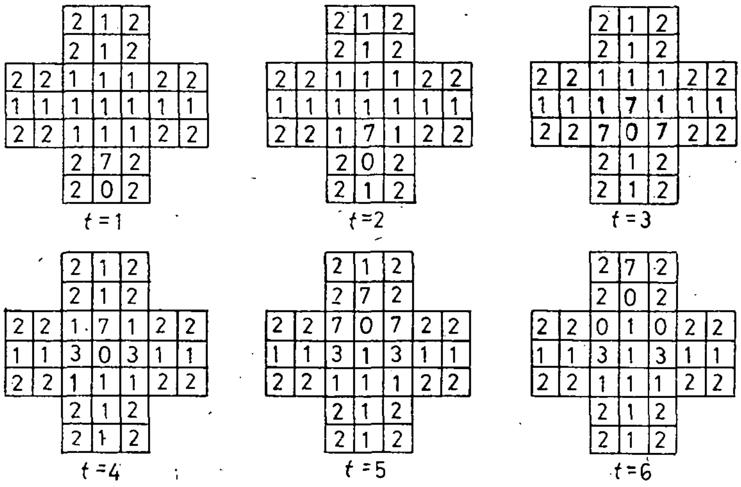


configuration



logical drawing

Functioning



The functioning of the crossover. The minimal following distance for the signals is 6.

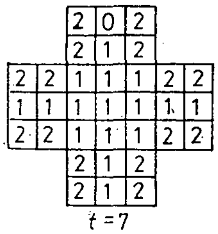
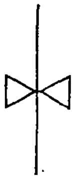


Fig. 26
Bidirectional pathway crossing

The 60, 70 transformer

	2	1	2	
	2	1	2	
2	3	1	3	2
2	1	1	1	2
	2	1	2	
	2	1	2	

configuration



logical drawing

The transformation of signal 60 to signal 70

	2	0	2	
	2	6	2	
2	3	1	3	2
2	1	1	1	2
	2	1	2	
	2	1	2	

$t=1$

	2	1	2	
	2	0	2	
2	3	7	3	2
2	1	1	1	2
	2	1	2	
	2	1	2	

$t=2$

	2	1	2	
	2	1	2	
2	3	0	3	2
2	1	7	1	2
	2	1	2	
	2	1	2	

$t=3$

	2	1	2	
	2	1	2	
2	3	1	3	2
2	7	0	7	2
	2	7	2	
	2	1	2	

$t=4$

	2	1	2	
	2	1	2	
2	3	1	3	2
2	0	1	0	2
	2	0	2	
	2	7	2	

$t=5$

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

$t=6$

The propagation of signal 70

	2	0	2	
	2	7	2	
2	3	1	3	2
2	1	1	1	2
	2	1	2	
	2	1	2	

$t=1$

	2	1	2	
	2	0	2	
2	3	7	3	2
2	1	1	1	2
	2	1	2	
	2	1	2	

$t=2$

	2	1	2	
	2	1	2	
2	3	0	3	2
2	1	7	1	2
	2	1	2	
	2	1	2	

$t=3$

	2	1	2	
	2	1	2	
2	3	1	3	2
2	7	0	7	2
	2	7	2	
	2	1	2	

$t=4$

	2	1	2	
	2	1	2	
2	3	1	3	2
2	0	1	0	2
	2	0	2	
	2	7	2	

$t=5$

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

$t=6$

The transformer does not works correctly from output

	2	1	2	
	2	1	2	
2	3	1	3	2
2	1	1	1	2
	2	6	2	
	2	0	2	

$t=1$

	2	1	2	
	2	1	2	
2	3	1	3	2
2	1	6	1	2
	2	0	2	
	2	1	2	

$t=2$

	2	1	2	
	2	1	2	
2	3	7	3	2
2	6	0	6	2
	2	1	2	
	2	1	2	

$t=3$

	2	1	2	
	2	7	2	
2	3	0	3	2
2	0	0	0	2
	2	1	2	
	2	1	2	

$t=4$

Fig. 27

Gene. The configuration called gene is constituted by four cells in state 1 located in square. It is usable only connected to pathway. The gene has three functions: rectifying, signal transformation and signal source. We already have shown now the gene rectifies as a local gate. As a transformer the gene runs as a minimal sized loop. Fig. 28 shows its two possible situations. The multiplying table is identic with the arbitrary sized loop.

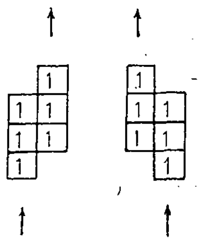


Fig. 28

The signal transformer gene wich is a minimal sized loop

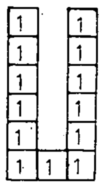


Fig. 29

Sharp turned pathway

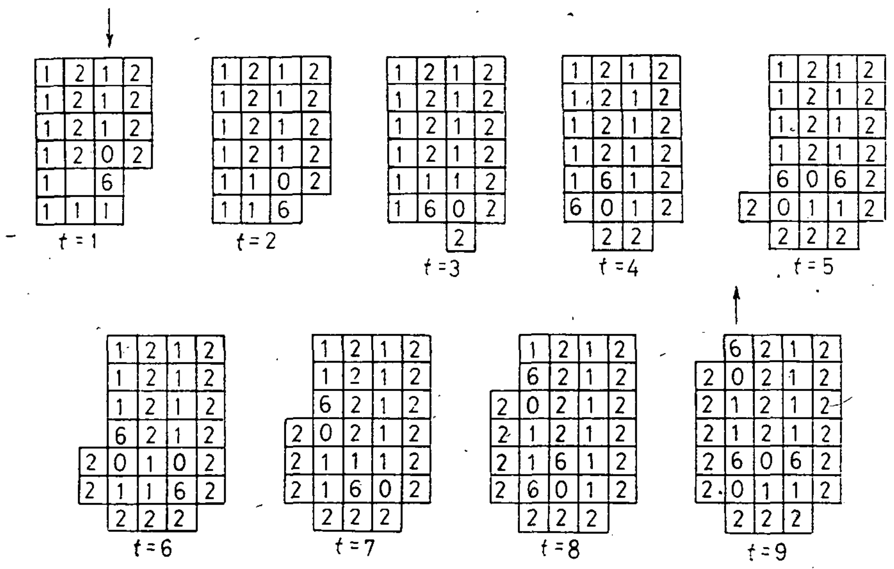
That is also a possibility, to use the gene as a signal source. Let us make a sharp turned way as in fig. 29.

A single 60 signal makes the sheating, but is is enough to create simoultaneously a high intensity signal source giving the sequence

$$\langle 6, 0, 1, 1, 6, 0, 1, 1, \dots\rangle$$

as in fig. 30.

When constructing genes one has to be careful for the activation which must be done from the opposite direction where the later input will be.



The gene in the sharp turned pathway made by the sheathing signal 60 giving the sequence $\langle 6,0,1,1,6,0,1,1, \dots\rangle$

Fig. 30

Signal source

II. The maximum-selector*

The problem

Given a word of length l

$$x = \langle x^1, \dots, x^l \rangle$$

whose elements $x^1, \dots, x^l$ are binary words of length k from B^k, where $B = \{0, 1\}$. The binary numbers given by these words are denoted by

$$n^1, n^2, \dots, n^l,$$

i.e.,

$$n^j = \sum_{i=1}^{k} 2^{k-i} x_i^j, \quad j = 1, \dots, l, \quad \text{where} \quad x^j = x_1^j, \dots, x_k^j \quad \text{and} \quad x_i^j = \{0, 1\}.$$

The problem is to find the binary word n representing the maximal one among $n^1, \dots, n^l$.

The solution

The algorithm. Let us consider an alphabet $C = \{0, 1, S\}$ which is an extension of the binary Boolean alphabet with a "stop signal" $S \cdot C = B \cup \{S\}$. Define the Boolean sum function σ as the following mapping $\sigma\colon S^l \longrightarrow C$ such that for a word $u = \langle u^1, \dots, u^l \rangle$,

$$\sigma(u) = \begin{cases} S & \text{if} \quad u^1 = \dots = u^l = S, \\ 1 & \text{if} \quad \exists j \, u^j = 1, \\ 0 - \text{otherwise.} \end{cases}$$

Let us define also the mapping $\tau\colon C^2 \longrightarrow C$ such that for any $(y, y') \in C^2$,

$$\tau(y, y') = \begin{cases} 0 & \text{if} \quad y = y' = 0, \\ 1 & \text{if} \quad y = y' = 1, \\ S - \text{otherwise.} \end{cases}$$

Define also the mapping $C^l \longrightarrow C^l$ such that for the any $u \in C^l$,

$$\theta(u) = \langle \tau(u^1, \sigma(u)), \dots, \tau(u^l, \sigma(u)) \rangle.$$

The function θ serves as a "stop function". If the relating word contains 1 then θ keeps the element of the word having the value 1, giving stop signal otherwise. If the relating word does not contain 1, leaves the elements unchanged.

Let μ be the mapping $\mu\colon B \times C \longrightarrow C$ for which

$$\mu(b, c) = \begin{cases} S & \text{if} \quad c = S, \\ b & \text{if} \quad c \neq S \end{cases}$$

for any $b \in B$ and $c \in C$.

* See [8].

Finally, if $x=\langle x^1, ..., x^l\rangle$ is given, define the words $\psi_1(x), ..., \psi_k(x)$ of length l by

$$\psi_i(x) = \begin{cases} x_1 & \text{if} \quad i = 1, \\ \langle \mu(x_i^1, \theta\psi_{i-1}(x)^1), ..., \mu(x_i^l, \theta\psi_{i-1}(x)^l)\rangle & \text{if} \quad i = 2, 3, ..., k. \end{cases}$$

By means of μ, we get a stop signal keeping function, which keeps in the stop signal at each the place in $\psi_i(x)$, where it already appeared for $i'<i$. The algorithm is running in the following way.

First step: Calculate the values

$$\sigma\psi_1 = \sigma(x_1)$$

and

$$\theta\psi_1(x) = \theta(x_1) = \langle \tau(x_1^1, \sigma(x_1)), ..., \tau(x_1^l, \sigma(x_1))\rangle$$

and

$$\psi_2(x) = \langle \mu(x_2^1, \theta(x_1)^1), ..., \mu(x_2^l, \theta(x_1)^l)\rangle.$$

Second step: Calculate the values

$$\sigma\psi_2(x)$$

and

$$\theta\psi_2(x) = \langle \tau(\psi_2(x)^1, \sigma\psi_2(x)), ..., \tau(\psi_2(x)^l, \sigma\psi_2(x))\rangle$$

and

$$\psi_3(x) = \langle \mu(x_3^1, \theta\psi_2(x)^1), ..., \mu(x_3^l, \theta\psi_2(x)^l)\rangle.$$

i-th *step:* Calculate the values

$$\sigma\psi_1(x)$$

and

$$\theta\psi_i(x) = \langle \tau(\psi_i(x)^1, \sigma\psi_i(x)), ..., \tau(\psi_i(x)^l, \sigma\psi_i(x))\rangle$$

and

$$\psi_{i+1}(x) = \langle \mu(x_{i+1}^1, \theta\psi_i(x)^1), ..., \mu(x_{i+1}^l, \theta\psi_i(x)^l)\rangle.$$

After the k-th step the result in the case

$$(\exists j')(\forall i)(\tau^{j'} \neq S)$$

is

$$n = \langle \tau_1^{j'}, ..., \tau_k^{j'}\rangle.$$

Example: $x=\langle\langle 1, 1, 0, 1, 0, 1\rangle, \langle 1, 1, 0, 0, 1, 1\rangle, \langle 1, 1, 0, 1, 0, 0\rangle\rangle$, $(l=3, k=6)$.

First step:

$$x_1 = \langle 1, 1, 1\rangle, \quad \psi_1(x) = x_1, \quad \sigma\psi_1(x) = 1,$$

$$\theta(x_1) = \langle \tau(1, 1), \tau(1, 1), \tau(1, 1)\rangle = \langle 1, 1, 1\rangle;$$

$$x_2 = \langle 1, 1, 1,\rangle, \quad \psi_2(x) = \langle \mu(1, 1), \mu(1, 1), \mu(1, 1\rangle = \langle 1, 1, 1\rangle.$$

Second step:

$$\sigma\psi_2(x) = 1,$$

$$\theta(\psi_2(x)) = \langle \tau(1, 1), \tau(1, 1), \tau(1, 1)\rangle = \langle 1, 1, 1\rangle,$$

$$x_3 = \langle 0, 0, 0\rangle, \quad \psi_3(x) = \langle \mu(0, 1), \mu(0, 1)\, \mu(0, 1)\rangle = \langle 0, 0, 0\rangle.$$

Third step:

$$\sigma\psi_3(x) = 0,$$

$$\theta\psi_3(x) = \langle\tau(0,0), \tau(0,0), \tau(0,0)\rangle = \langle 0,0,0\rangle,$$

$$x_4 = \langle 1,0,1\rangle, \quad \psi_4(x) = \langle\mu(1,0), \mu(0,0), \mu(1,0)\rangle = \langle 1,0,1\rangle.$$

Fourth step:

$$\sigma\psi_4(x) = 1,$$

$$\theta\psi_4(x) = \langle\tau(1,1), \tau(0,1), \tau(1,1)\rangle = \langle 1,S,1\rangle,$$

$$x_5 = \langle 0,1,0\rangle, \quad \psi_5(x) = \langle\mu(0,1), \mu(1,S), \mu(0,1)\rangle = \langle 0,S,0\rangle.$$

Fifth step:

$$\sigma\psi_5(x) = 0,$$

$$\theta\psi_5(x) = \langle\tau(0,0), \tau(S,0), \tau(0,0)\rangle = \langle 0,S,0\rangle,$$

$$x_6 = \langle 1,1,0\rangle, \quad \psi_6(x) = \langle\mu(1,0), \mu(1,S), \mu(0,0)\rangle = \langle 1,S,0\rangle.$$

Sixth step:

$$\sigma\psi_6(x) = 1,$$

$$\theta\psi_6(x) = \langle\tau(1,1), \tau(S,1), \tau(0,1)\rangle = \langle 1,S,S\rangle.$$

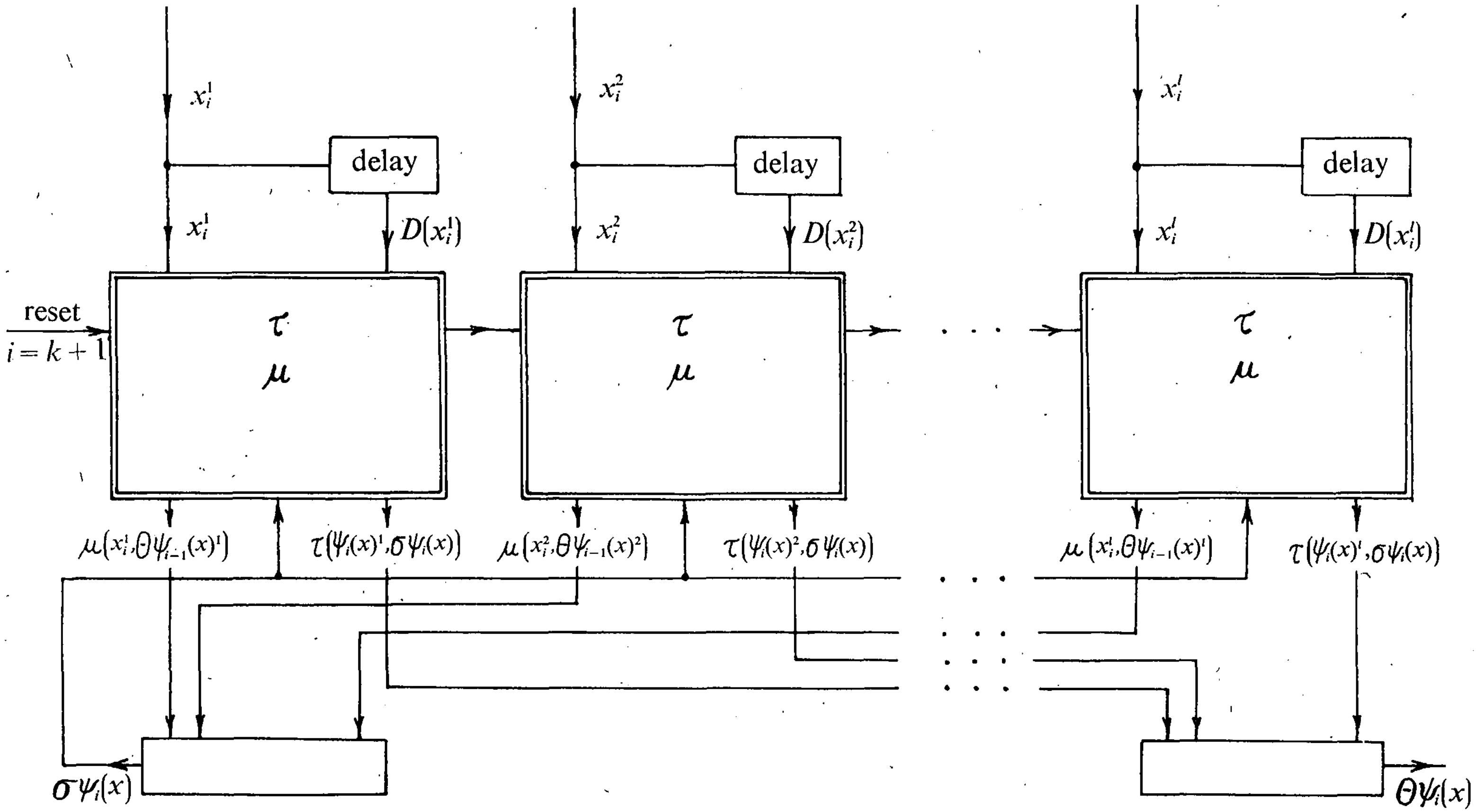
Since $j'=1$ thus

$$n = x^1 = \langle 1,1,0,1,0,1\rangle.$$

The solution principle in Codd-ICRA automaton. The selection of the maximal number happens in Codd-ICRA automaton. The solution principle is the following: the numbers are introduced into the cellular space simoultaneously; they are propagating along parallel pathways. After the inputs each number is duplicated and one of their sample arrives into a gate system as information. The second samples — after delay — go further in their channels as data. In these channels they annihilate or move on, according to the situation of the already opened or closed gate system. At all those places where the figures with highest local value are 1 the numbers go further, while at places where they are 0 the numbers are annihilated. In the second step, where the figures with the second highest local value are 1, among the remained numbers, those numbers go further, and the numbers are annihilated if they are 0, and so an, until the k-th figures, which have the lawest local value (the coefficients of 2^0). Exception is if at the actual step among the remained numbers 0-s are everywhere. In this case, all the remained numbers go further. The outputs of the channels are joined, so at the final, single output one gets the maximal number (even if the same one appeared at several channels).

The maximal number selector is called MAXEL.

The principal plan. The principal plan shows the parallel pathways, the duplication, the delay and the selector units, forming the functions τ and μ. At last we signed the feedback for the Boolean sum, and the output giving the maximal number (see fig. 31).



x_i^1
delay
x_i^1
$D(x_i^1)$
x_i^2
delay
x_i^2
$D(x_i^2)$
x_i^l
delay
x_i^l
$D(x_i^l)$
reset
$i = k + 1$
τ
μ
τ
μ
τ
μ
$\mu(x_i^1, \Theta\psi_{i-1}(x)^1)$
$\tau(\psi_i(x)^1, \sigma\psi_i(x))$
$\mu(x_i^2, \Theta\psi_{i-1}(x)^2)$
$\tau(\psi_i(x)^2, \sigma\psi_i(x))$
$\mu(x_i^l, \Theta\psi_{i-1}(x)^l)$
$\tau(\psi_i(x)^l, \sigma\psi_i(x))$
$\sigma\psi_i(x)$
$\Theta\psi_i(x)$

Fig. 31
l-MAXEL
Principal plan

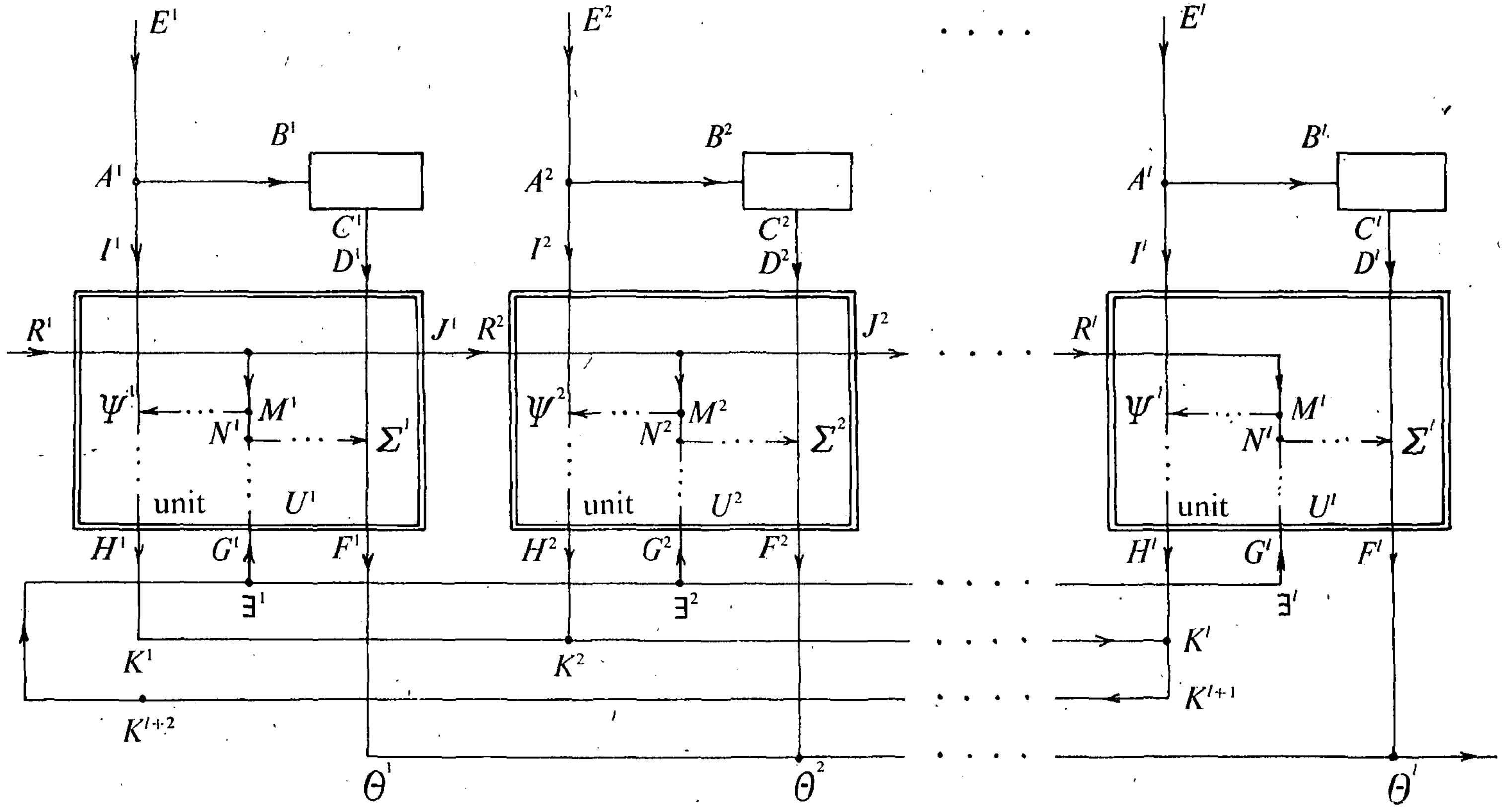


Fig. 32
l-MAXEL
System scheme

System scheme. The structure of the system scheme is identical with the principal plan, apart from the mapping of the input and output points. The selecting unit here is also almost a black box only, just showing the pathway control role of the gates located in the unit (see fig. 32).

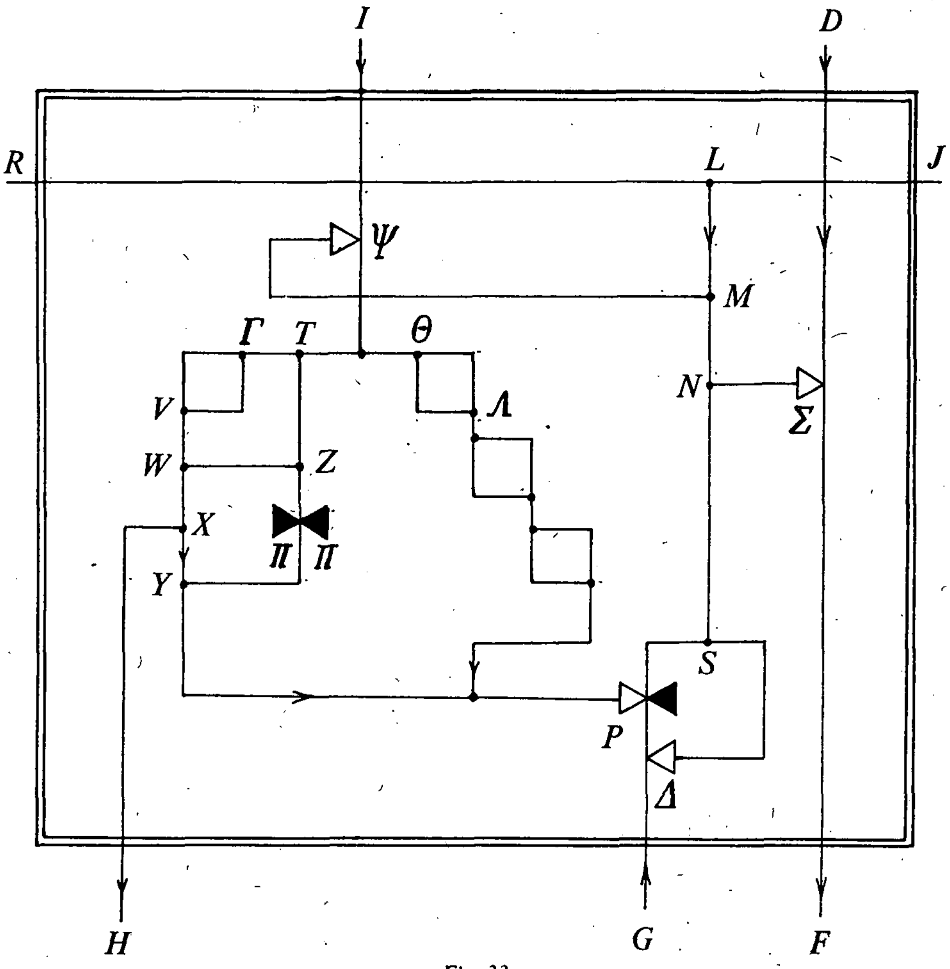


Fig. 33
The U^j selecting unit of *l*-MAXEL
Logical drawing

Logical drawing. Figure 33 shows the complete logical structure of a selecting unit.

The several cases of the functioning are the following.

α) $\sigma(x_i) = 1$

α′) $x_1^j = 1$ (the signal content is 60).

Inputs The name of the cell	Mode of application	Signal content
I	x_i^j information	$\left\{\begin{matrix}60\\11\\70\end{matrix}\right.$
D	$D(x_i^j)$ data	$\left\{\begin{matrix}60\\11\\70\end{matrix}\right.$
G	$\sigma(x_i)$	$\left\{\begin{matrix}70\\11\end{matrix}\right.$
R	reset; active once at each $(k+1)$-th time step	$\left\{\begin{matrix}60\\11\end{matrix}\right.$
Outputs The name of the cell	Mode of application	Signal content
H	x_i^j information	$\left\{\begin{matrix}60\\11\end{matrix}\right.$
F	$D(x_i^j)$ data	$\left\{\begin{matrix}60\\11\\70\end{matrix}\right.$
J	reset; active once at each $(k+1)$-th time step	$\left\{\begin{matrix}60\\11\end{matrix}\right.$

In this case four events happen.

I. The regeneration of gate P

$$x_i^j\colon \quad \text{goes via} \quad I \to \Psi \to \theta \to \Lambda \to P.$$

The loops $\theta\Lambda$ and the two next assure, that in each case one gets signal 60 at cell P, thus if it was in state 2 then it remains there, and if it was in state 3, it changes to state 2.

II. Information

$$x_i^j\colon \quad \text{goes via} \quad \begin{tikzcd} I \arrow[r] & \Psi \arrow[r] & T \arrow[r] \arrow[dr] & \Gamma \arrow[r] & V \arrow[dr] & & & \\ & & & & Z \arrow[r] & W \arrow[r] & X \arrow[r] & H. \end{tikzcd}$$

By this way the signal can not arrive at cell P since the transformer Π transforms both 60 and 70 to 70, and this joins at cell Y with 60, where it annihilates.

III. The sum $\sigma(x_i)$ enters into the unit at cell G and annihilates at gate P.

IV. Data

$$D(x_i^j)\colon \quad \text{goes via} \quad D \to \Sigma \to F$$

α″) $\quad x_i^j = 0$ (the signal content is 70).

In this case three events happen.

I. The regeneration of gate P

$$x_i^j\text{:}\quad \text{goes via}\quad I \to \Psi \to \theta \to \Lambda \to P.$$

The signal 60 does not change the state of P cell, if it is 2, but it will be changed to 2, if it is 3.

II. Information

$$x_i^j\text{:}\quad \text{goes via}\quad I \to \Psi \to T \to Z \to \Pi \to Y \to P,$$

and changes by the signal 70, the state of P cell from 2 to 3. The signal does not arrive at point X, thus nor at output H, since 40 arising from the end of the loop ΓV joins with 70 coming from TZ at point W, and because of $70\times 40=11$ annihilates here.

III. The sum $\sigma(x_i)$ enters into the unit at cell G and goes via

$$\begin{tikzcd}
 & & & \Delta & & \\
G \arrow[r] & P \arrow[r] & S \arrow[ur] \arrow[dr] & & \Sigma & \\
 & & & N \arrow[ur] \arrow[r] & M \arrow[r] & \Psi.
\end{tikzcd}$$

This $\sigma(x_i)$ places to state 3 in the gate cells Δ, Σ and Ψ respectively, which have had previous state 2. Afterwards, the inputs of the selecting unit are already closed for the signals

$$x_{i+1}^j,\quad D(x_i^j)\quad \text{and}\quad \sigma(x_{i+1}),$$

β) $$\sigma(x_i) = 0,$$

$$x_i^j = 0\quad \text{(the signal context is 70).}$$

In this case three events happen.

I. The regeneration of gate P

$$x_i^j\text{:}\quad \text{goes via}\quad I \to \Psi \to \theta \to \Lambda \to P.$$

The signal 60 doesn't change the state of P cell, if it's 2, but it will be changed to 2, if it is 3.

II. Information

$$x_i^j\text{:}\quad \text{goes via}\quad I \to \Psi \to T \to Z \to \Pi \to Y \to P,$$

and changes the state of P cell to 2 from 3. At points X and H no signal arrives because it was annaihilated at W.

III. Data

$$D(x_i^j)\text{:}\quad \text{goes via}\quad D \to \Sigma \to F.$$

The 0—1 *configuration of the selector unit's.* Figure 34 shows the 0—1 configuration of the unit described in the previous point.

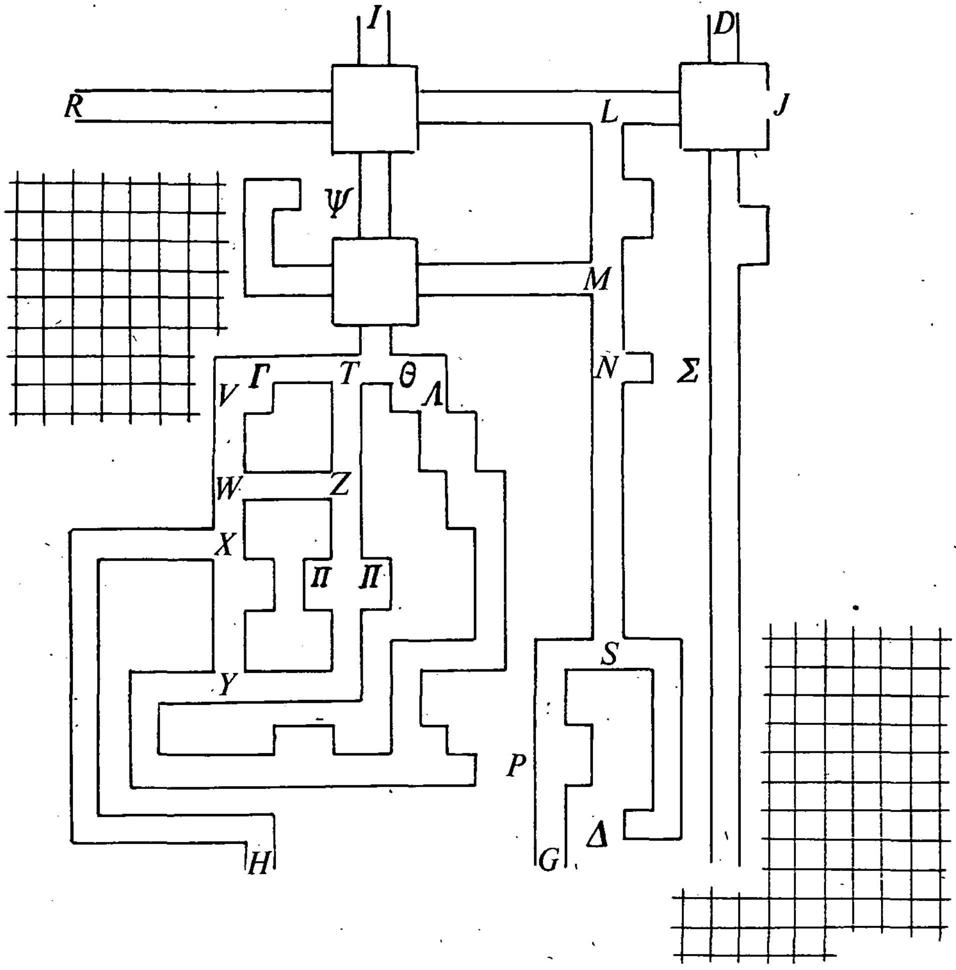


Fig. 34
The U^j selecting unit of *l*-MAXEL
0-1 configuration

The logical drawing of l-MAXEL's. $l=3$ is the smallest value of l for which one can show the slight assimmetries of the extremal selector units. Apart from these extremities, the construction of the MAXEL is modular (building bricks system).

The 0—1 *configuration of l-MAXEL's.* Figure 36 shows for $l=3$ and arbitrary k the 0—1 configuration. On the upper part of the figure one can see the input synchronization.

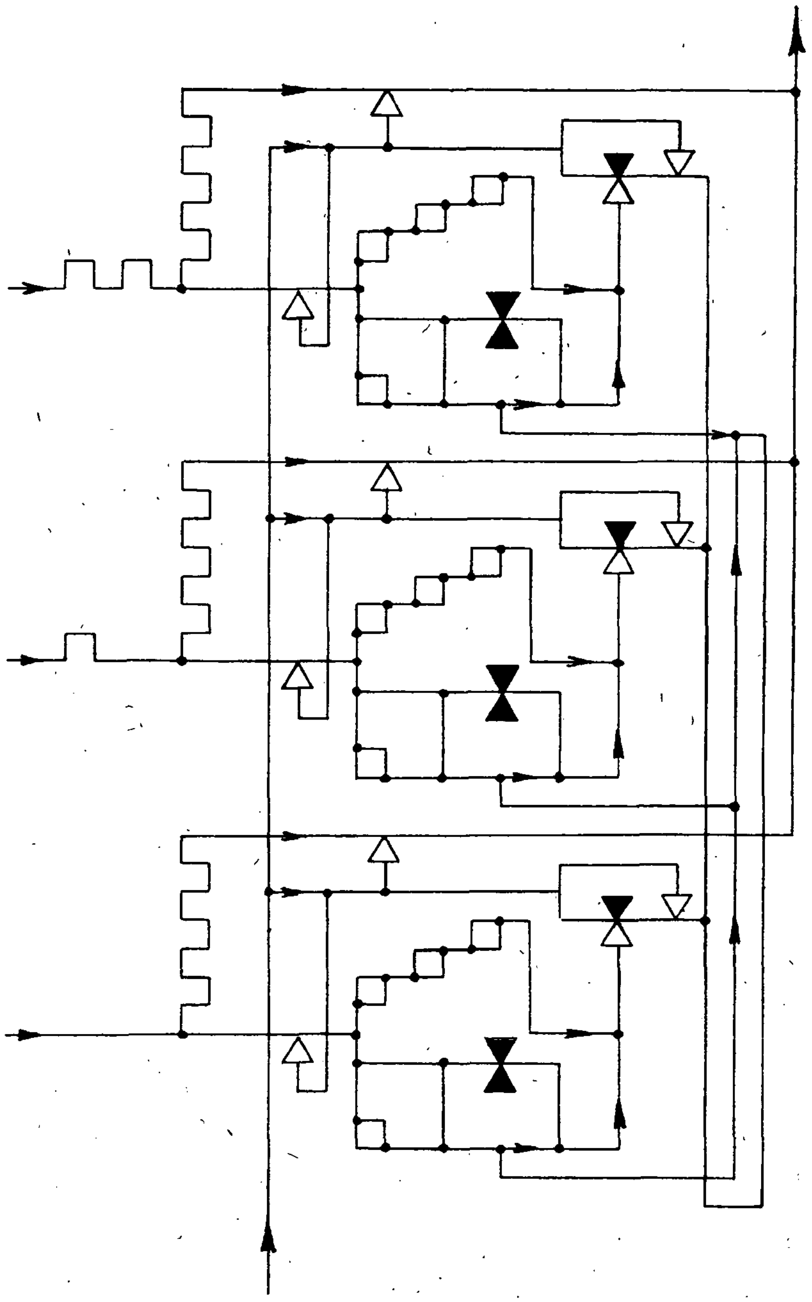


Fig. 35

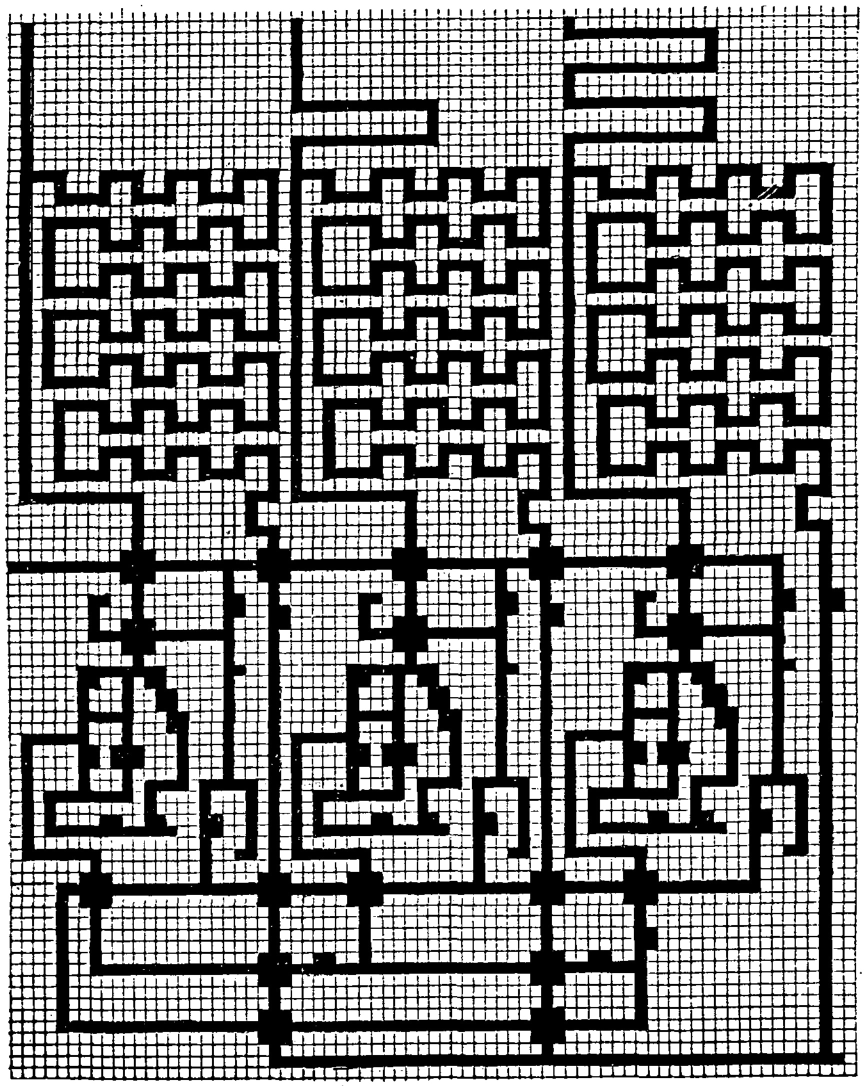


Fig. 36

l-MAXEL for $l = 3$
0-1 configuration

Coding

Inputs.

The Boolean 0 is coded by 70

The Boolean 1 is coded by 60

Outputs from the point of junctions $0^1, \ldots, 0^l$.

The Boolean 1 is coded by 60

$$\text{The Boolean 0 is coded by} \begin{cases} 70 \\ 40 \\ 11 \end{cases}$$

If $N_j(x) = \{j \mid x_i^j = n\}$,

$\# N_j(x) = 1 \pmod 3$ the Boolean 0 is coded by 70,

$\# N_j(x) = 2 \pmod 3$ the Boolean 0 is coded by 40,

$\# N_j(x) = 3 \pmod 3$ the Boolean 0 is coded by 11.

Working times. The distance betwen cells A and B expressed by the number of cells between them is denoted $\overline{AB}$, when the shortest possible walking on is chosen from A to B. The retard between points I^j, I^{j+1} and D^j, D^{j+1} consists always 24 tacts. The retard T_{ID} between points I^j, D^j is the following

$$T_{ID} = \overline{I^1 H^1 K^1 K^2 \ldots K^l K^{l+1} K^{l+2} \exists^1 G^1 \Sigma^1} - \overline{D^1 \Sigma^1} - 1.$$

$\overline{I^1 H^1 K^1} = 55.$

$\overline{K^1 K^2 \ldots K^l} = (l-1)24.$

$\overline{K^l K^{l+1}} = 4.$

$\overline{K^{l+1} K^{l+2}} = (l-1)24.$

$\overline{K^{l+2} \exists^1} = 29.$

$\overline{\exists^1 G^1 \Sigma^1} = 24.$

$\overline{D^1 \Sigma^1} = 12.$

$$T_{ID} = 48(l-1)+99.$$

The periodicity p, in other words the distance between signals x_i^j and x_{i+1}^j, is

$$p = \overline{I^1 H^1 K^1 K^2 \ldots K^l K^{l+1} K^{l+2} \exists^1 G^1 \Psi^1} - \overline{I^1 \Psi^1} - 1.$$

$\overline{\exists^1 G^1 \Psi^1} = 42.$

$\overline{I^1 \Psi^1} = 6.$

$$p = 48(l-1)+123.$$

The $T(n)$ working time for selecting the m maximal number is the running time for the first figure — n_1 — plus the time of the remained figures.

$$T(n) = T(n_1)+(k-1)p.$$

$$T(n_1) = T_{ID}+\overline{D^1F^1\Theta^1}+\overline{\Theta^1\Theta^2\ldots\Theta^l}.$$

$$\overline{D^1F^1\Theta^1} = 47.$$

$$\overline{\Theta^1\Theta^2\ldots\Theta^l} = (l-1)24.$$

$$T(n_1) = 72(l-1)+146.$$

$$T(n) = 24\,l(1+2k)+75\,k+1.$$

For example in the case

$$k=4,\quad l=3:\quad T_{ID}=195,\quad p=219,\quad T(n)=733;$$

$$k=10,\quad l=10:\quad T_{ID}=531,\quad p=555,\quad T(n)=5791.$$

One has to calculate the necessary time for the reactivization, by this way. The effect of putting in the gate Ψ last figure of the first number — x_k^1 — is

$$kp+\overline{I^1\Psi^1}.$$

By substracting from this the propagating time of the reactivating signal, one gets its starting time tact t_{reac}: minimal following distance,

$$t_{\text{reac}} = kp+\overline{I^1\Psi^1}-\overline{R^1L^1M^1\Psi^1}+\text{ minimal following ditance}$$

minimal following distance = 6,

$$\overline{I^1\Psi^1} = 7,$$

$$\overline{R^1L^1M^1\Psi^1} = 44.$$

Since

$$t_{\text{reac}} = kp-31$$

the necessary time interval between two selecting procedure T_{reac} is

$$T_{\text{reac}} = kp+\overline{I^1\Psi^1}+\text{ minimal following distance }-\overline{I^1\Psi^1}-kp,$$

$$T_{\text{reac}} = 6.$$

Size. Let S^l be the size of an l-MAXEL expressed in terms of the number of its constituting cells. Then

$$S^l = [l(\overline{R^lJ^l})+1](\overline{E^lI^l}+\overline{I^lH^l}+\overline{F^l\Theta^l}),$$

$$\overline{R^lJ^l} = 24,$$

$$\overline{E^lI^l} = 32+(l-3)6,\quad l\geqq 3,$$

$$\overline{I^lH^l} = 30,$$

$$\overline{F^l\Theta^l} = 17,$$

$$S^l = (24\,l+1)[79+(l-3)6],\quad l\geqq 3.$$

For example in the case

$$l = 3, \quad k = 4: \quad S^3 = 73\times79,$$
$$l = 10, \quad k = 10: \quad S^{10} = 241\times121.$$

Conclusion concerning the design

The advantages of the solution. 1. The data according to x_i^j coefficients are propagating along parallel pathways in the l-MAXEL, which permits a high simoultaneity within the frames of the task, making the profit from the two-dimensionality of Codd-ICRA cellular space.

2. The size of the l-MAXEL depends on the number l of numbers only, it is completely independent of the length k of the numbers.

3. The dependence of the working time the l-MAXEL on both l and k is very simple — linear — function.

4. If $l' > l$, l-MAXEL is easily extend ableby the insertion of new selector units.

The conditions of the solution. I. Inputs

1. For the inputs x_i^j one has to assure a constant speed which is independent of i and j.

2. If the inputs are not synchronized, one has to apply 24 tacts retard between two neighbouring j-th and $(j+1)$-th inputs, or build in input synchronization blocks.

3. The p periodicity of the signals as a function of l is

$$p = 48(l-1)+123.$$

II. The output periodicity is p, too.

III. One has to give in advance the values of k and l.

IV. Apart from the l inputs the l-MAXEL needs an $(l+1)$-th input for the reactivization after a whole selecting procedure.
The required reactivization time is

$$t_{\text{reac}} = kp-31.$$

Summary

After becoming acquainted with the basic techniques, and concepts of cellular automaton and cellular space we present a machine, which is able to select the maximal one among the numbers simoultaneously introduced into it.

NATIONAL CENTRE FOR EDUCATIONAL TECHNOLOGY
VESZPRÉM, HUNGARY

[1] Sometimes differently from the usual way of speaking, we say cellular *space* for the five-tuple $\mathfrak{A}=(I, v, L, S, \perp)$ and in these cases by a cellular *automaton* we mean the ordered pair $(\mathfrak{A}, q_0)$, where q_0 is a quiescent state.

[2] This published solution is a corrected version by means of the simulation experiences on PDP 10.

TABLE *Codd-ICRA*

s	s'	$s \perp s'$	s	s'	$s \perp s'$	s	s'	$s \perp s'$	s	s'	$s \perp s'$	s	s'	$s \perp s'$	s	s'	$s \perp s'$	s	s'	$s \perp s'$
0	0006	2	0	1214	1	0	1662	1	1	0262	6	1	1242	4	1	2255	5	2	0306	3
0	0007	3	0	1215	1	0	1666	1	1	0263	6	1	1243	4	1	2263	6	2	0307	3
0	0015	3	0	1216	1	0	1717	1	1	0272	7	1	1244	5	1	2266	6	2	0711	1
0	0016	2	0	1217	1	0	1722	1	1	0273	7	1	1252	5	1	2273	7	2	1117	1
0	0025	2	0	1222	7	0	1727	1	1	0342	4	1	1253	5	1	2277	3	2	1232	3
0	0026	2	0	1223	1	0	1772	1	1	0343	4	1	1255	6	1	2324	4	2	2324	3
0	0042	2	0	1224	1	0	1777	1	1	0352	5	1	1262	6	1	2325	5	2	2325	3
0	0051	3	0	1225	1	0	2226	2	1	0353	5	1	1263	6	1	2326	6	2	2326	3
0	0061	2	0	1226	1	0	2266	2	1	0362	6	1	1266	6	1	2327	7	2	2327	3
0	0062	2	0	1227	1	1	0004	0	1	0363	6	1	1272	7	1	2334	4	3	0002	2
0	0066	2	0	1232	6	1	0006	6	1	0372	7	1	1273	7	1	2335	5	3	0006	1
0	0106	2	0	1235	1	1	0007	3	1	0373	7	1	1277	4	1	2336	6	3	0025	1
0	0107	3	0	1242	1	1	0014	0	1	0411	0	1	1343	7	1	2337	7	3	0026	0
0	0116	2	0	1244	1	1	0016	6	1	0606	6	1	1353	7	1	2343	4	3	0027	0
0	0126	2	0	1252	1	1	0017	2	1	0611	6	1	1363	7	1	2353	5	3	0042	1
0	0161	2	0	1253	1	1	0024	4	1	0616	6	1	1373	7	1	2363	6	3	0062	0
0	0162	2	0	1255	1	1	0026	6	1	0621	6	1	1422	4	1	2373	7	3	0072	0
0	0166	2	0	1262	1	1	0036	6	1	0622	6	1	1424	5	1	2424	4	3	0102	2
0	0206	2	0	1266	1	1	0041	0	1	0626	6	1	1442	5	1	2433	4	3	0103	0
0	0207	3	0	1272	1	1	0052	5	1	0661	6	1	1522	5	1	2525	5	3	0106	4
0	0213	1	0	1273	1	1	0061	6	1	0722	7	1	1525	6	1	2533	5	3	0107	7
0	0226	2	0	1277	1	1	0062	6	1	1114	0	1	1552	6	1	2626	6	3	0111	1
0	0227	1	0	1313	1	1	0063	6	1	1115	6	1	1616	6	1	2633	6	3	0162	0
0	0261	2	0	1322	1	1	0066	6	1	1116	6	1	1622	6	1	2727	7	3	0172	0
0	0262	2	0	1324	1	1	0071	7	1	1117	7	1	1626	6	1	2733	7	3	0261	0
0	0272	1	0	1342	1	1	0104	0	1	1124	4	1	1662	6	1	3334	4	3	0271	0
0	0363	1	0	1352	1	1	0105	5	1	1125	5	1	1666	3	1	3335	5	3	1111	1
0	0611	2	0	1362	1	1	0106	6	1	1126	6	1	1717	7	1	3336	6	3	1232	2
0	0621	2	0	1363	1	1	0107	2	1	1127	7	1	1722	7	1	3337	7	3	2324	2
0	0622	2	0	1372	1	1	0114	0	1	1142	4	1	1727	4	2	0006	0	3	2325	2
0	0626	2	0	1373	1	1	0116	6	1	1152	5	1	1772	4	2	0007	1	3	2326	2
0	0661	2	0	1422	1	1	0126	6	1	1162	6	1	1777	3	2	0017	1	3	2327	2
0	0722	1	0	1424	1	1	0141	0	1	1166	3	1	2224	4	2	0025	3	4	s'	1 ([a])
0	1116	2	0	1432	1	1	0161	6	1	1172	7	1	2225	5	2	0042	3			0 ([b])
0	1124	1	0	1442	1	1	0162	6	1	1177	3	1	2226	6	2	0071	1	5	s'	1 ([a])
0	1125	1	0	1522	1	1	0166	6	1	1214	4	1	2227	7	2	0106	0			0 ([b])
0	1126	1	0	1523	1	1	0226	6	1	1215	5	1	2234	4	2	0107	1	6	s'	1 ([c])
0	1127	1	0	1525	1	1	0227	7	1	1216	6	1	2235	5	2	0117	1			0 ([d])
0	1142	1	0	1532	1	1	0242	4	1	1217	7	1	2236	6	2	0142	3	7	s'	1 ([c])
0	1152	1	0	1552	1	1	0243	4	1	1224	4	1	2237	7	2	0171	1			0 ([d])
0	1162	1	0	1616	1	1	0252	5	1	1225	5	1	2243	4	2	0206	0			
0	1166	2	0	1622	1	1	0253	5	1	1226	6	1	2244	4	2	0207	3			
0	1212	1	0	1626	1	1	0261	6	1	1227	7	1	2253	5	2	0251	3			

s : the own state of the cell
$s' = \langle s'_1, s'_2, s'_3, s'_4 \rangle$: the neighbourhood states
$s \perp s'$: the next state of the cell

(*a*) : if s' does not contain 1
(*b*) : if s' contains 1
(*c*) : if s' does not contain any even number
(*d*) : if s' contains an even number

References

[1] CODD, E. F., *Cellular automata*, ACM monograph series, Academic Press, 1968.
[2] DETTAI, I., *Effectivity problems and suggestions concerning Codd-ICRA cellular space*, KGM ISZSZI Technical Report, 1974.
[3] FAY, G., *A basic course on cellular automata*, University Lecture Notes, Budapest University of Eötvös Loránd (in Hungarian) 1975.
[4] NEUMANN VON, J., *Theory of self-reproducing automata*, University of Illinois Press, Urbana Edited and completed by A. W. Burks, 1966.
[5] RIGUET, J., Automates cellulaires à bord et automates Codd-ICRA, I. *Comptes rendus de l'academis les sciences de Paris*, t 282, Serie A., 1976, pp. 167—170.
[6] RIGUET, J., Automates cellulaires à bord et automates Codd-ICRA, II. *Comptes rendus de l'academie les sciences de Paris*, t 282 Serie A, 1976, pp. 239—242.
[7] SZŐKE, M. (Mrs), *Destruction in Codd-ICRA*, Graduate thesis in mathematics, Budapest University of Eötvös Loránd (in Hungarian), 1975.
[8] TAKACS, D. V. (Mrs), *Un automate cellulaire Codd-ICRA pour la recherche du maximum de l nombres*, These présentée a Université René Descartes de Paris Sorbonne, Doctorat d'Université, 1975.

(Received Oct. 13, 1976)