FUNDAÇÃO ESCOLA TÉCNICA LIBERATO SALZANO VIEIRA DA CUNHA

A SOLUTION OF THE GENERAL MODEL FOR A DIGITAL SYSTEM

Filipi Damasceno Vianna

Novo Hamburgo, April , 1998

"To all injusticed inventors. Them that, with their ideas, revolutionized their time, contributed to a faster progress of the mankind... But, unfortunately, because a lot of reasons (reasons?) — maybe the most bruising reason could be the foreign ignorance — they do not are more than anonymous beings in the universal history."

B. Hamilton Almeida
in *"O outro lado das telecomunicações"*.

ACKNOWLEDGEMENTS

To Teacher Wagner Waneck Martins, the pionner of the Non-Von Newmann machines.
To Santos Dumant, the Brazilian Inventor for a invention of the machines *more weighty than air*.
To Priest Landell de Moura, precursor of Telecommunications in Brazil.

IDENTIFICATION DATA

School	Fundação Escola Técnica Liberato Salzano Vieira da Cunha
Course	Electronic
Habilitation	Technical
Research	THE SOLUTION OF THE GENERAL MODEL FOR A DIGITAL SYSTEM
Category	Computer Science - SYST
Scientist supervising	Teacher José Inácio Coelho da Silva
Researchers	Filipi Damasceno Vianna

SUMMARY

1. INTRODUCTION

The intent of this research is presents the Logical [1]Esção (n~m~p) in its 3rd year of investigation. The Logical Esção (n~m~p) goes towards the 5th generations computers, vanishing the called Von Neumann's Bottleneck from the current computers. Enable, also, a [2]deterministic perception of the variable inputs and functioning in a real time. From this datas, the theoretical aspects of this new computation was investigated and tried to use it in a lot of systems to confirm its functioning and evaluate the advantages against the conventional controllers.
The significance of this work is contribution for a search to eliminate the Von Neumann bottleneck from the present computational architecture. It deals by putting in practical the "General Model for a Digital System", (PHISTER, 1958), that has a entire [3]random functioning and processing in real time. In abridgment, the right control system.
The Von Neumann bottleneck is a present computer characteristic where the information is obliged to pass by a centralized stage which is gonna deal, arrange and decide the destiny that the program information flow must to take. Happens that the advanced technology, like computers architecture and computing language project, use very elevated math abstractions. To become this knowledge accessible to market and mainly towards technical terms, specially to Technical High Schools, is the fulfillment of this work.

---

[1] **Esção** It pronunciation is **ES'ÄW** ['] shows main stress – [ä] is a nasal vowel

[2] Deterministic Dependent of the starting state of the inputs

[3] Random 1, depends of uncertain factors, favorable or not to a determined event. 2- eventual, fortuitous, douthful.

2. WHERE THIS INVESTIGATION COMMING FROM

This project started from a reading indication made three years ago (june, 1995) done by the Liberato Foundation teacher José Inácio Coelho towards the students Daniel Fink (now in the Electrical Engineer College) and Filipi D. Vianna (4th year), both in the Technical Electronic Program at that time. The reading refers to the work of teacher Wagner W. Martins, entitled as a "**ESÇÃO** (*n*~*m*~*p*) — a non Von Neumann Computer". During the reading, gave us a huge curiosity to investigate the subject, considering extremely promising inside our area. Thus, we decided to start a deep investigation about the subject developing a search project to prove the theories shown in the book and to give further contribution to his development.

The most attractive part of a **ESÇÃO** is to realize as a partner of the main technology that was conceived by humanity. How important is informatic nowadays, the responsibility of this work then becomes very important.
On august 14, 1995, a letter was sent to the teacher W. W. Martins, from EPUSP. The letter had the proposal to inform the existing of a new research line in the subject presented, and asks for new datas search after the book publication. Our letter was promptly answered back by Mr. Martins, who sent a list of all the material published before and after of his book "**ESÇÃO** (*n*~*m*~*p*) - Um Computador Não-Von Neumann". In the enclosed letter, we received wishes of good profit in the subject reading and Fairs participation.

3. THE PROBLEM

This work is justified by the studies, analyzes to controllers improvement and implementation of a logic circuit based in the Phister's theory of the General Model for a Digital System (see topic **Erro! Argumento de opção desconhecido.** at page 15), called **EsÇÃo** (*n~m~p)*.

Primarily the objective was to understand the logic explained in the book "**EsÇÃo** (*n~m~p)* — a non Von-Neumann computer", by teacher Wagner W. Martins. Fortify the research concepts already done and to it , collect all the necessary material, wherever it be.

The main problem of this work was to know:

*Is the logic **EsÇÃo** really brings advantages towards a microprocessed system?*

and

*Is the actual high integration of devices technology already prepared to implement a **General Model for a Digital System**?*

3.1 The Sub Problems

- The logic **EsÇÃo** brings advantages in the changing of a system with a microprocessor?
- What is the limitation?
- If exist, what is the best acting area for this Boolean Technology?
- What the aplication of this propouse in a world dominated by the Von Neumann bottleneck?

- How this technology can concours with the alternatives of actual Non-Von Neumann machines?
- Are we prepared to (popular) use a Non-Von Neumann machine?

3.2 The Hypothesis

- It is probable an **EsÇÃo** ($n{\sim}m{\sim}p$) Machine, can replace a microprocessor.
- It is probable the **EsÇÃo** ($n{\sim}m{\sim}p$) can replace all the microprocessed systems.
- It is probable the **Non-Von** Generation Machine brings advantages in terms of cost.
- It is probable the Logic **EsÇÃo** offers equivalent control to the microprocessed systems with a short benefit cost
- It is probable the **EsÇÃo** Machine can be versatile, simple and with easy implementation.

4. COMPUTER SYSTEMS ORGANIZATION

How the mankind information flow has growed up last years, was necessary a lot of improvements in computacional architecture. Always to provide gains in speed, data handle and storage. Some of this as follows.

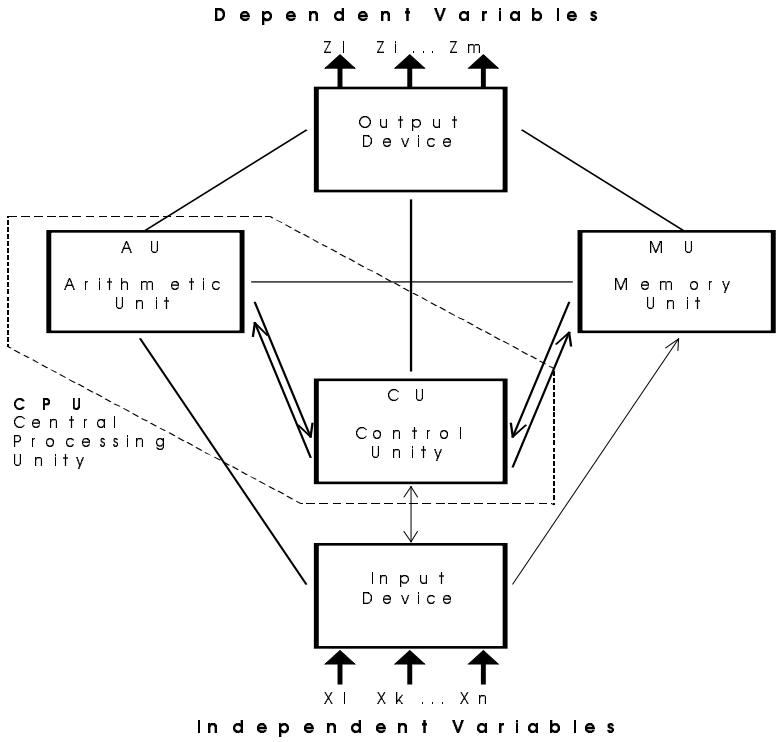
		Number of Data Streams	
		Single	Multiple
Number of Data Streams	Single	**SISD** (von Neumann)	**SIMD** (Vector, Array)
	Multiple	**MISD** (pipeline)	**MIMD** (Multiple micros)

4.1 A Von Neumann Computer - SISD

This concept use a Single stream of Instruction controlling a Single stream of Data, where a development factor exists that is normal in basically all the machines already created after ENIAC (1946) up to the most recent Personal Computers. It deals about a computational architecture developed in 1945 by a mathematician called John Von [4]Neumann. Knowing that would be impossible to create a "General Model for a Digital System", Von Neumann divided the complex computer problem in single five parts:

a) Input devices
b) Output devices
c) Memory Unit (MU)
d) Arithmetic Unit (AU)
e) Control Unit (CU)

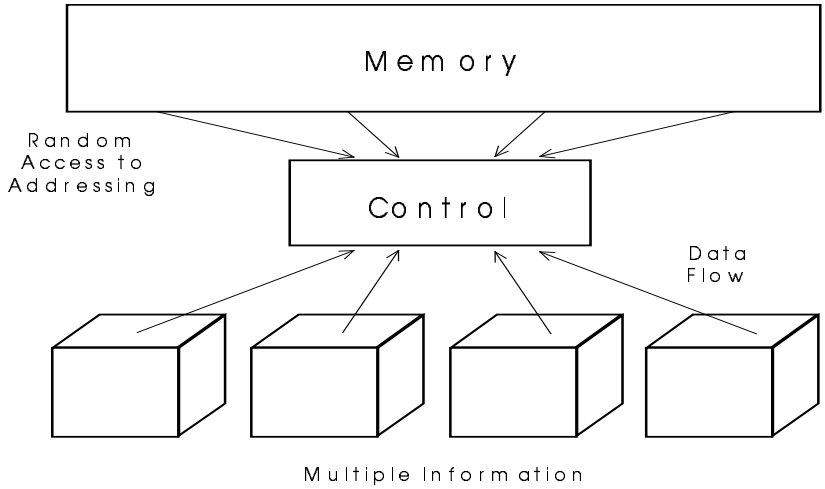
[4] Von Neumann Hungarian-American Mathematician (Budapest, DEC 28th, 1903 - Washington, FEB 3th, 1957)



The only difference existing between the pioneer machines and the present ones is the coupling of Arithmetic Unit with the Control Unit which resulted in already known **UCP** (Central Processing Unit). As a result of this coupling has the appearance of the microprocessor through the Very Large Scale Integration (VLSI).
The CPU works as a manager that is gonna share the time for every work division in a computer, due to processing. Thus, in a determined instant,

1. The CPU collects a input data
2. Checks out the memory to know the next step to be done
3. Goes to the arithmetic unit to calculate the received data
4. Finally sends the processed data to the output

Each task takes a determined time break, due to necessity of a **centralized control** and by communication based in addressing inside the large memory, which limits the machine to it is own inner clock.
Despite of using several ways to increase the computer speed and perception like multiple processing, none of them can get rid off them selves from the “Von Neumann’s bottleneck”. The Von Neumann bottleneck can be observed in the sketch:



So, the bottleneck force a serial processing, even when parts of the program could be processed in parallel, or, simultaneously. Nowadays, several attempts have been done towards the elimination of this bottleneck, had as a hurdle to the development of the present informatic.

4.2 A Non Von Neumann Machines

The great informations in the present world deals about the necessity of having better ways to save and manage each time faster and faster these informations. The researchers have been working to contour the Von Neumann architecture problems.

4.2.1 Parallel Processing

A lot of developers believes that the better , faster and cheapest way to a Von Neumann architecture is the parallel processing. The advantages have been known a long time ago. Also Von Neumann got impressed in the late forties with the discovered of parallel processing in animal brains. If the charges in hardware were not so expensive that time, he could have created a machine similar to a brain.
The Parallel System Processing type MIMD (Multiple Instruction -Multiple Data) englobe two kinds of machines:

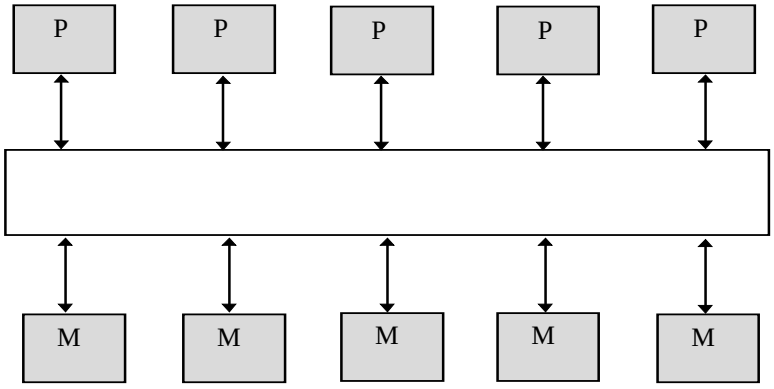
- Shared Memory Systems - multiprocessors
- Distributed Memory Systems - multicomputers

4.2.2 Shared Memory Systems - Multiprocessors

All processors has access to a global memory and it comunication and sincronization are controlled by datas stored in it memorys. It is possible to say the access time to any memory space of a generic processor is the same. Usually conections among buses and memorys are extremally hard (*Tightly Coupled Systems*).

Despite the shared memory offers flexibility and easy programming, there is the problem of complexity and interconection cost together with difficult of low modularity and scalability, forcing a limited number of processors units.

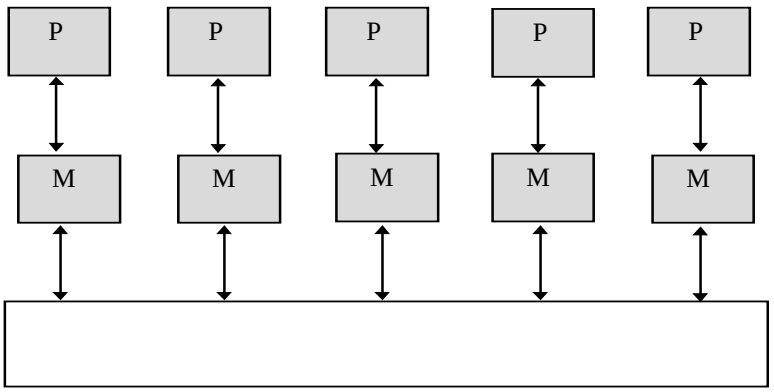
> *"... shared memory offers incresead flexibility and programming ease at the cost of additional hardware complexity and thus lower peak performance for a given machine cost. Software implementation is easier, and hardware implementation is harder."*
> *(ALMASI-GOTLIEB, The Pros and Cons of Shared Memory, page 29)*



4.2.3 Distributed Memory Systems - Multicomputers

Each processor has it own memory, local access, and it is interconect with other processors by a conection system. The processor-memory cells has also some ROM, to know what topology, rooting and other protocols to conect with another cells.

This configuration can be compared with actual network computers, showing also the same bus bottleneck problem. The speed will be affected if the necessary data is in other processor memory. However, the amount of processors can, now, reach tenths to thousand of units.



4.2.4 Josephson's Computes

Using the system of super conductor we get a high gain in speed. But all devices based in Josephson terminals needs to be kept in low temperature, in order of tenths of degrees Kelvin.

4.2.5 ULSI — Ultra Large Scale Integration

The firms sought, normally, process to increase the density of integration. This because it is [5]CISC, asking for a high quantity of transistors. It has occurred between them, innovations to couple UCP memory sections cache, events prevention, pipelines (to execute instructions forward), join together others processors. Even the instruction reduction CISC is tried to increase the speed — [6]RISC processors.

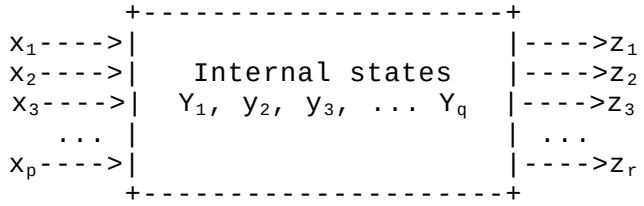
---

[5] CISC — Complex Instruction Set Computer— CPU Project who instructions set has several numbers of instruction, complex, who allows an easy program elaboration, but reduce the execution speed

[6] RISC — Reduced Instruction Set Computer — CPU Project who instructions set has a low instruction number and of fast execution, who brings a complex program writing, increasing the speed

5. THE GENERAL MODEL FOR A DIGITAL SYSTEM

Historically, the computacional architecture of the "Von Neumann machines" is closed in a difficulty to solve the "General Model for a Digital System", by PHISTER in his book "Logical Desing of Digital Computers", 1958. His theory say that any model can be represented like a black box, including digital systems or sequential switching circuits.



This black box consists in a certain number of inputs (x_i), a certain number of outputs (z_i) and a certain number of internal states (y_i).

> *"At any bit-time, the state of each of the output lines* z_i *is a function of the state of all the input lines at that time and of all the internal states at that time."*
> *(PHISTER, pp 144)*

This general relationship is indicated in equation bellow:

$$z_i^n = f_i(x_1, x_2, \cdots, x_p; y_1, y_2, \cdots, y_q)^n \quad i = 1, 2, 3, \cdots, r$$

The internal state of the machine depends on the state of the memory devices inside the box. The relationship is indicated in equation below, where the future internal state ($n+1$) is a function of the input lines and the anterior internal states:

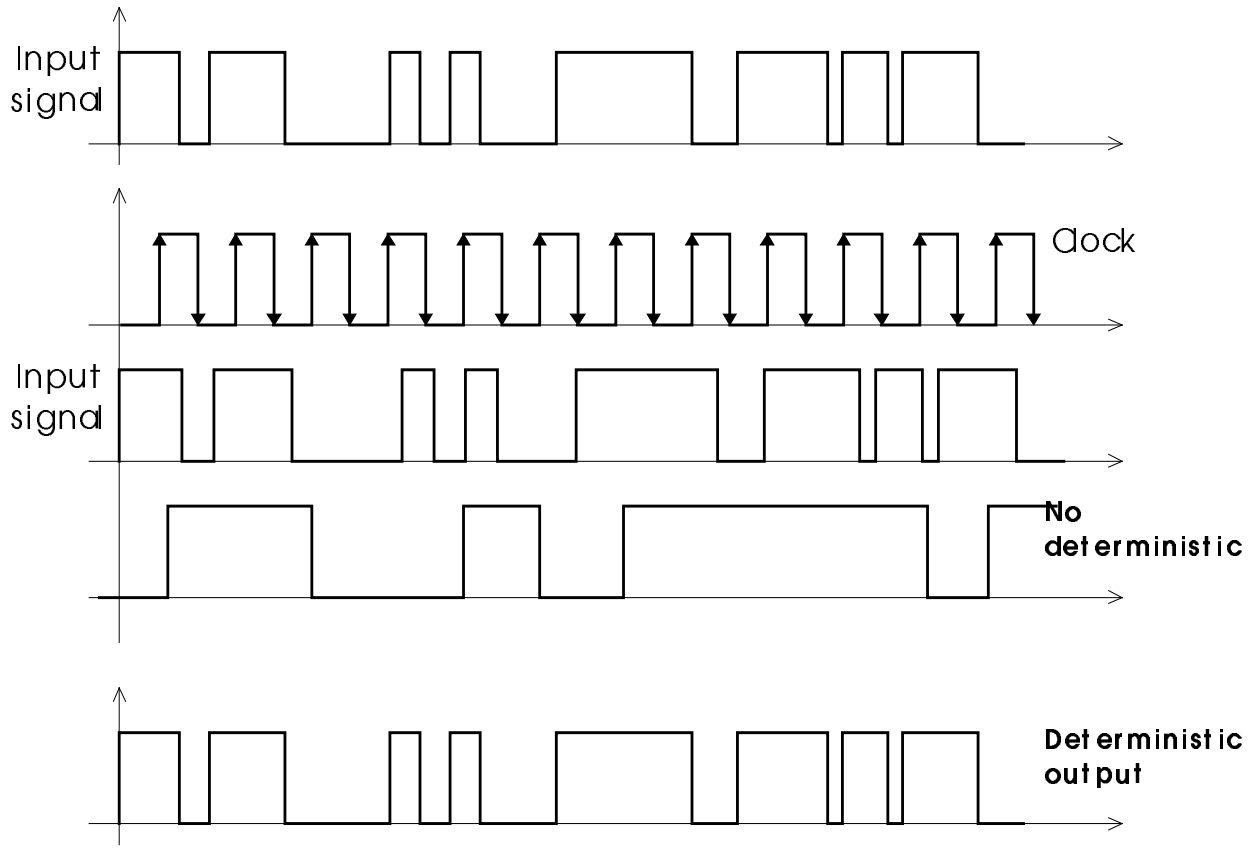
$$y_i^{n+1} = g_i(x_1, x_2, \cdots, x_p; y_1, y_2, \cdots, y_q)^n \quad i = 1, 2, 3, \cdots, q$$

"This will be recognized as a set of difference equations determining the state of every memory element inside the box at one bit-time as a function of the state of all of the input lines and all of the memory units at the previous bit-time."
(PHISTER, pp 144)

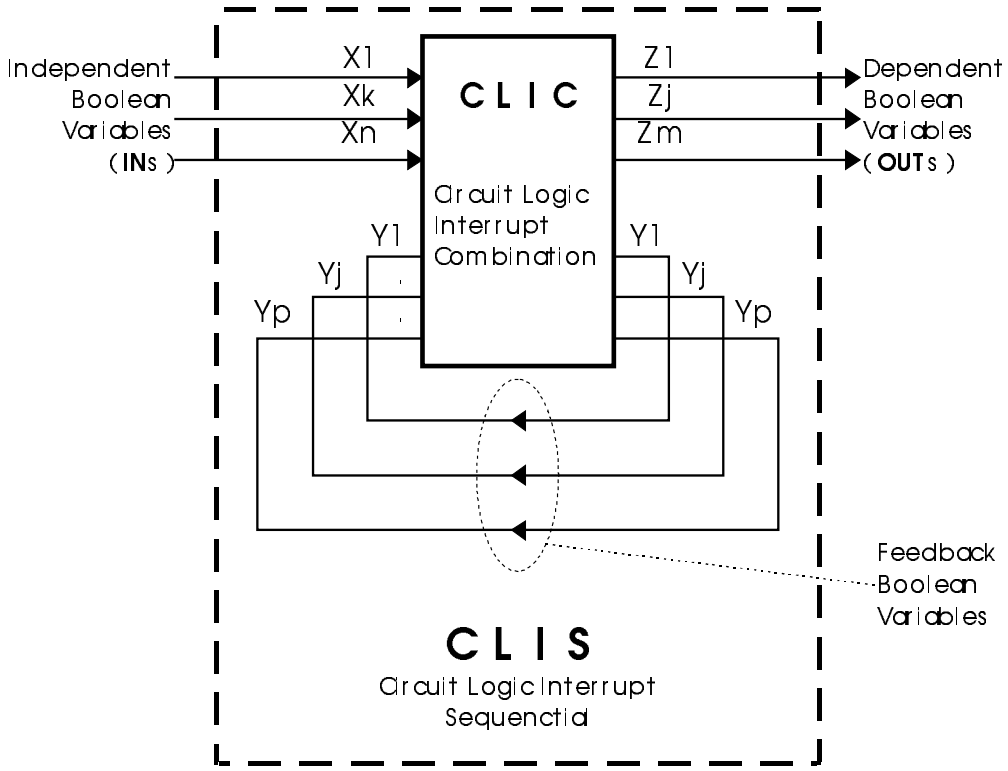
According to his theory, the problem of the practical application of this model is that it is too general. It conception produce a number extremaly large of permutation and combinations of internal states created by equations describes upon. So, there were not devices with enough capacity to load a very large data amount. And, probaly some of them will not be used by the system.
Because it, this model was broken down into at least five parts, as showed in the chapter "**Erro! Argumento de opção desconhecido.**" (page 11).

5.1 Sequential Machines

Sequential machines are circuits that execute a sequency of tasks defined by the user. As saw in the last chapters, there are different implementation ways for sequencial machines, always trying increments and artifices to minimize the execution time of each task.
As fast as it stay, this machines will continue tie in one characteristic: the eternal battle against the time. It occurs because they are assembled by devices like Flip-Flops tieds in a temporal step (clock) following orders comming from a combinacional circuit and its input sensors.
Because the fact this conventional controllers are tie in a set of clock pulses that sequence its tasks, the machine has a **non-deterministic perception** of its inputs sensors. Looking at the graph below, the events that occurs in a shorter time than the clock time are not percepted by machine.
Therefore, there is a **probalistic behavior** because the sample detection of the values received in its sensors.



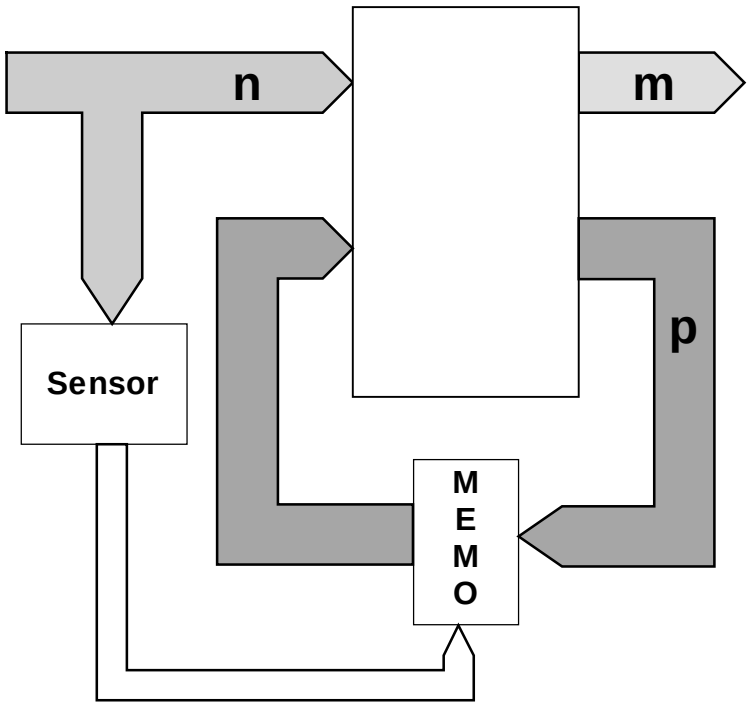
By the way, how could be concepted a machine with **deterministic perception**?
The answer is using the actual VLSI technology. Keeping off the flip-flops and his clock, is it possible implement a logical circuit sequential switch (CLIS – Circuito Lógico Interruptor Síncrono) with feedback booleans variables that allow the stable state corresponding the memories "**p**" (Y_1, Y_2, Y_3, …, Y_n)



This sketch shows in a very wide way the functioning of a **EsÇÃo** machine. It deals about a **CLIS** Sequencial machine) established by only a **CLIC** (memory devices). It means that this system does not need the control unit, only inputs and outputs devices and some feedback variables.
But where is the Arithmetic Unit? It was eliminated by the called **DDD**(know by heart department). It shows a memory device characteristic which is the capacity of storing tables. Therefore, instead of exist a responsible to execute the calculations, like there is something a child learning the table of figures. This procedure brings, above all the fast access to the results, the precision can be desired.
This fact matches with almost the Sequential Machines in the market (like the microcomputers, microprocessors, programmable controllers, and so on) that has a synchronized functioning with it internal clock, almost always at high frequency, introducing a discreet time showing the variation of the inputs. Events that can not be realized occurs between one and another exhibition, having errors, or hazards in the combined circuits. In resume, the machine can lost itself. Because it there is the necessity of high frequency clock in the present computers, what is totally eliminated in an **EsÇÃo** machine.
As you can see, analyzing the computers structures already available in the market (from the Von Neumann generation) with an **EsÇÃo**, I noted that we changed the microprocessor, and consequently keeps off the famous bottleneck, including a perception that becomes capable of representing any digital system in a REAL TIME, independent of capacity and physical size.
What really tells about a machine from the Non Von Neumann generation, like an **EsÇÃo**, is the practice development of a Pure Asynchronous Sequential Machine, or machines that always works in a real time with a entire aleatoric starting.

6. ESÇÃO MACHINES

Following the theories developed in the thesis published in the book "**ESÇÃO** (*n*~*m*~*p*) — A Non Von Neumann Computer", from the teacher W. W. Martins, the **ESÇÃO** is a sequential machine that use a lot of devices from the family **ROM** (read only memory), who can works in asynchronous mode, without clock, keeping out the use of the traditional flip-flop devices, or even synchronism using one of the inputs as a temporal step, which is gonna be called of XØ, who is input undependable variable.
It has then two kinds of **ESÇÃO** machines, being the first one of asynchronous functioning, with 100% inner capacity occupied to memorize the programs and conversions in a real time. We got 2^n synchronism combinations possible in the inputs, being in the number of undependable variables, or monitored sensors for example.
The second kind would be an **ESÇÃO** functioning with a temporal step in low frequency. It means we are gonna use a independent variable as a clock input and then we are gonna have 50% of inner memory capacity utilization to save the programs. So we have 2^n-1 possible combination to a variation of the other independent variables. It is good to remember that the functioning continues in a real time and with intelligence of searching among the programs the suitable outputs to the aleatoric occurrences of the inputs.



- *n* number of <u>independents</u> Boolean variables (input)

- *m* number of dependable Boolean variables (output)
- *p* number of internal Boolean variables (feedback or memories)

So then, a **ESÇÃO** (n-m-p) is nothing more than a computacional machine, that using the microelectronic devices (Integration in a Very Large Scale) like RAM, ROM, EPROM, etc., creates a control able to eliminate completely a microprocessor, having economical advantages in "real time".

It is good to remember that comparing both structures, it got a great technology advantage, which is the whole elimination of the microprocessor, who takes together the historical "virus" of the Von Neumann Bottleneck. Another characteristic of a **ESÇÃO** machine is the non-probability behaviour. It means that it do not have a segment period of time to verify the input events, but we have a kind of entire monitoring of the independent variables, which give us a great fidelity towards the perception information received.

6.1 Understanding Esção (n-m-p)

To really understand the functioning of a Esção is important to know two [7]concepts.

- **EIE** Stable Internal State
- **EII** Instable Internal State

To explain this topics, let is go back to Phister Theory, in the title *"**Logical Design of Digital Computers**"*,1958(see chapter 5), who tries to solve the "***General Model for a Digital System***".

With the intention to give a particular nomenclature to Esção logic, the name of the variables used by Phister to explain the "General Model for a Digital System" (p, r, y) was changed by (n,m,p) as follows below:

- **n** number of independents inputs
- **m** number of dependents outputs
- **p** number of internal states

The last number [**p**] deals towards the system of the memory, who can be computed by the base-two, giving the **EIE** number. These will define the space to store of sequential programs that the machine will execute.

$$2^p = \mathrm{EIE}$$

Any similar from a Phister Model towards the logic shown here is not mere coincidence. This work, among several things, suggests a solution to the ***General Model to a Digital System*** using the VLSI technology.

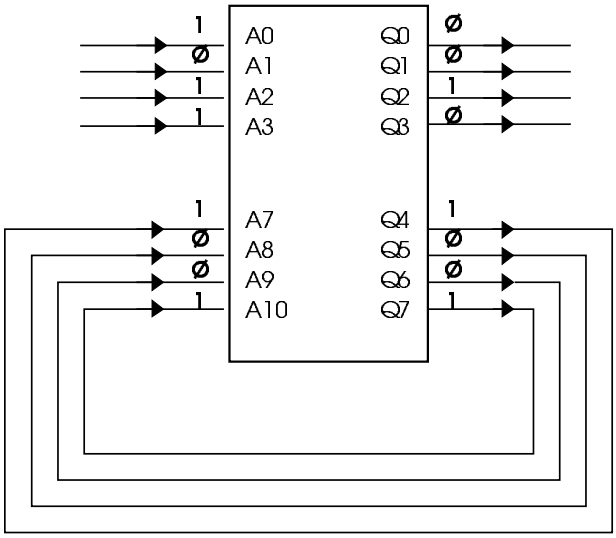
The amount of **EIE** will define the space to store the sequential programs that the machine will execute. So, for a **ESÇÃO** machine, as a **EIE** is simply a memory space, or else, a BYTE. As each byte uses one address, and to access must be gived the right address in the inputs of memory device.

Remembering about input variables, it always will be in the EPROM's address bus, sharing place with the feedback variables. It means that each input change shows a new address. And from there starts a process of **ESÇÃO** machine!

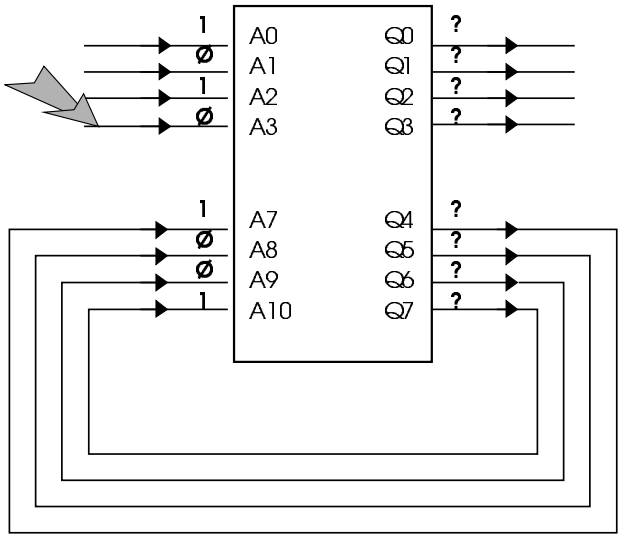
---

[7] **EIE** & **EII** Stable Internal State and Instable Internal State — will be maintained the original abbreviature in this work

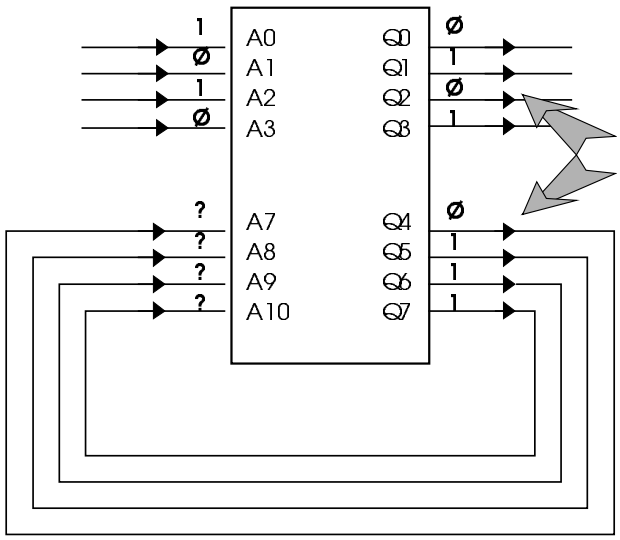
1. The EPROM found itself in a Stable Internal State (**EIE**), a constant input, and the feedback is identical as in the input as in the output. This state can be called of standby, because it is waiting a new order.



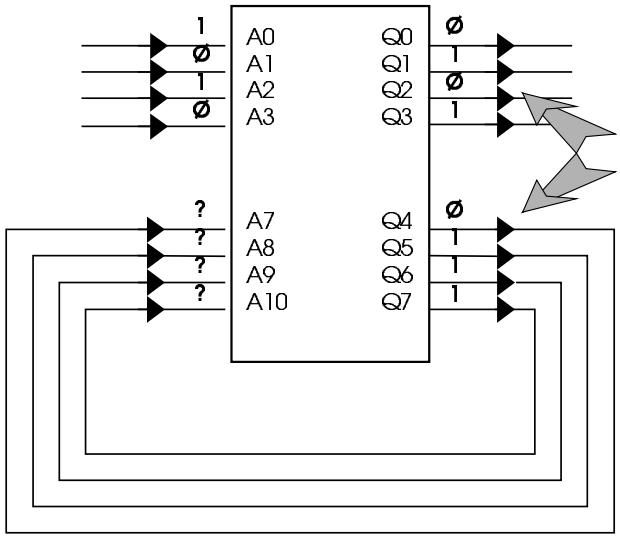
1. Happens a variation in the inputs.



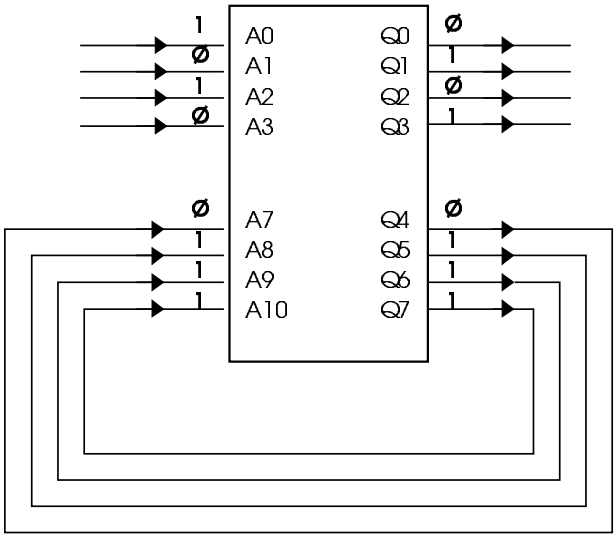
1. The EPROM gets it as a new address and immediately sends a new content in the output.



1. Now there is a Instable Internal State (**EII**), where through the feedback lines, a part from new data goes back to the adress bus, pointing out a new address. It is good to remember that this process is extremely fast, having the speed from the changing electrons in silicon.



1. The new adress pointed out by feedback will be again Stable Internal State [**EIE**]. It has the exact information that was requested with the change in the input bit. This answer stays still in the EPROM output, who stays again in standby, waiting for the new order.



1. Note that the new address pointed out by **EII** must have, in the bits reserved to feedback, the same data of the anterior adress. So, the **EIE** store the answers that are needed and that the machine should give according to input variables. What determines the content of **EIE** is the programmer. In another words the [8]**DDD** is made uniquely **EIE**.

2. On the order side, the **EII** are not determined by the programmer, but by the software that is gonna generate the FIRMWARE. It is function is to direct the correct addressing to EPROM. Another function of a **EII** is to make a suggestion of output to be availed by another circuit, that will accept or not(see A/D Converter). So we have several **EII** to a **EIE**, despite of usually the proportion is one by one.

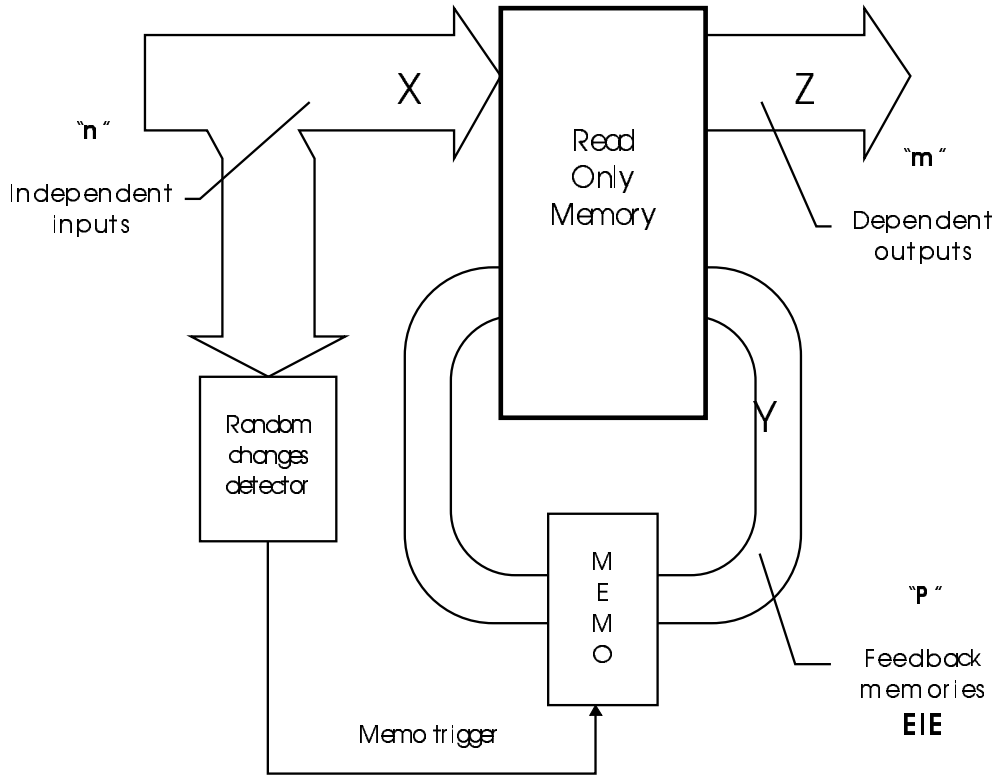
6.2 The ESÇÃO (*n~m~p*) Machine Conception

To project a sequential machine being synchronous or asynchronous, using the logic **ESÇÃO**, I got a follow methodology. The steps below were used as a foundation to the accomplishments of the prototypes for the practical presentation of this research.

[8] **DDD** Departamento Da Decoreba — Learn by Heart Department

1. Accomplish a list of internal stable state **EIE**
2. Elect the independent variable (input VAB), the dependent variables (output VAB) and the internal variables (feedback VAB), drawing in a operational block, the logic circuit sequential switch, whose project have been intended.
3. Drawing of the correspondent GRAPH with or without the temporal step (variable XØ), depending on the application.
4. The XØ introduction as temporal step, will , allow that the **EIE** list, gave lately change to 'project matrix' where will be introduced the unstable state concepts and stable state.
5. It's possible optionally, to transform the "project matrix" in a kind of truth-table equivalent
6. By the numerical definition, that will be seen later, from the Project Matrix or the equivalent truth table, we'll get a expression of FABs (Booleans Arithmetic Functions) that will correspond to the direct programming of the own technologic device VLSI, who can be any unit of the family RAM / ROM / EPROM / EEPROM and so on.

Now let's go for a bigger zoom in the **EsÇÃo** diagram, as shown in the picture below.



Here is the ROM device (Reading Only Memory). The MEMOS are the feedback memories (Y_1,...,Y_P) that is gonna determine the internal state tables **EIE** of the controllers. Here is the general condition to an EPROM:

$$n^{o}\ EIE = 2^{p}$$

The Aleatoric Changes Detector is the memory's feedback trigger, for every variation in the input variables: X1,...,Xn

In general conditions to an EPROM 2716 (2k Bytes) we have following:

$$n + p \leq 11$$

Address bus is made by 11 bits that takes the 2^{11}memory addresses.

$$n + p \leq 8 \cdot N$$

The number of unused exits plus the feedback variable are always less or equal to 8, which is the induced limitation made by the memory of the Data Bus. It is possible to increase the numbers adding more memory devices in the circuit. Then it has N • 8 outputs, where N is the amount of 2716, or another memory device.

6.3 About Project Matrix

The project matrix deals with an important step to get the firmware correspondent to a recording combined device. The translation to an algorithm, has the main element to get a compiler to these machines. I. e., the compiler will be a project matrix automated by a software.

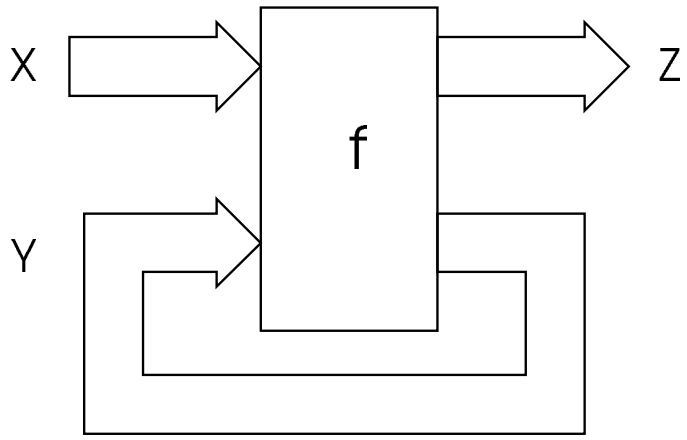
Once settled the answers (or EIE), goes to the determination of the numbers n, m, p. And after accomplish the correspondent graph, adding or not the temporal step.

From the graph loops is that determines the **EII** to be added in the project matrix, together with the **EIE**. In the first line, all variations of an input variable (X and XØ) as in the first column showing the feedback variables.

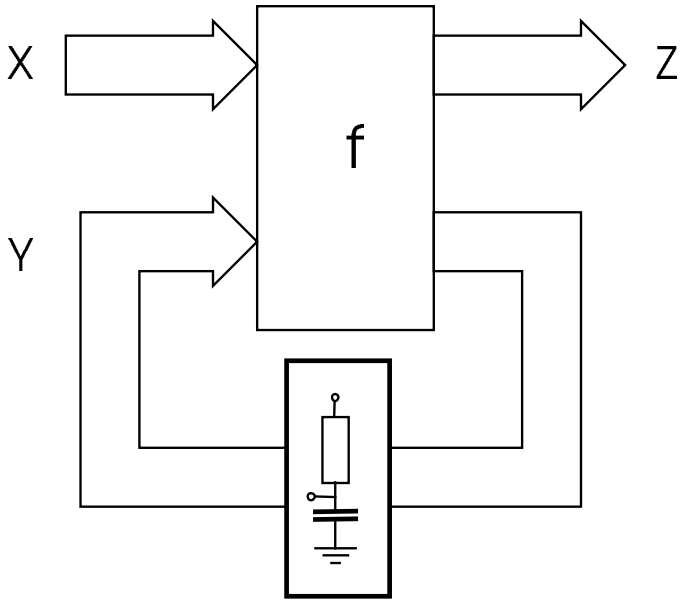
The blocks gives a continuos order with 4 bits (16 possible states). From these, 8 will be **EII**. Aleatoric, depends on the hooks that bond each Instable State to it is stable. It is possible to note that there is no jump limits in this board. The specific variables inside the block shows the change from **Ø** to **1**.
Finally, the datas must be organized in a truth table to be got the definitive firmware.

6.4 Pratical Search

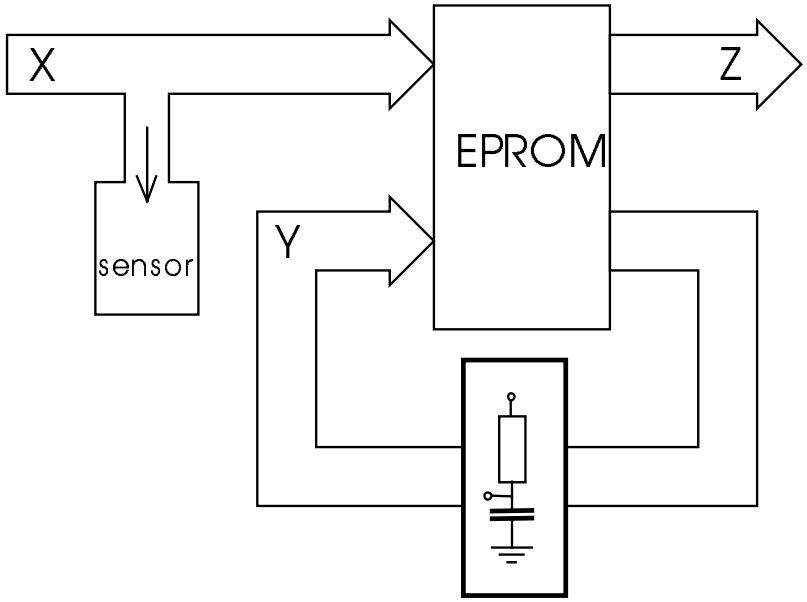
One of the most important data of this research was to know what already have been done in practical terms.
The basic principle of a **EsÇÃo** is to get the logical combined function that, when feedback has the sequential behaviour desired.



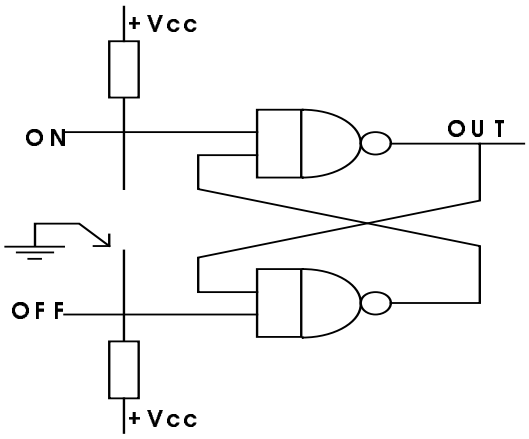
Among the memory devices that can be used, the EPROM seem to be the most versatile because is non volatile and erasable by ultra violet light.
To filter the occurred transients between the state variations In the feedback , a RC circuit was placed to transmit the data without noise to the output as shown in the picture below.



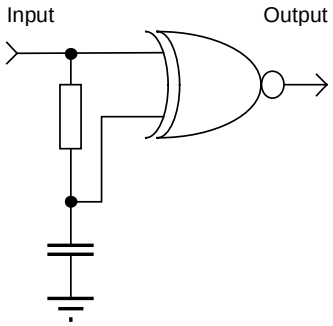
This configuration has a big problem, because is practically impossible say that all the feedback bits would change at the same time, due to component tolerances. The way to solve is to use a Gray Code in the output of the system who commands the bit to change one of them per time.
The use of Gray Code impair the generation of state tables in the moment to project the machine. To this a register circuit was created, ignited by the changing level of anyone of inputs .



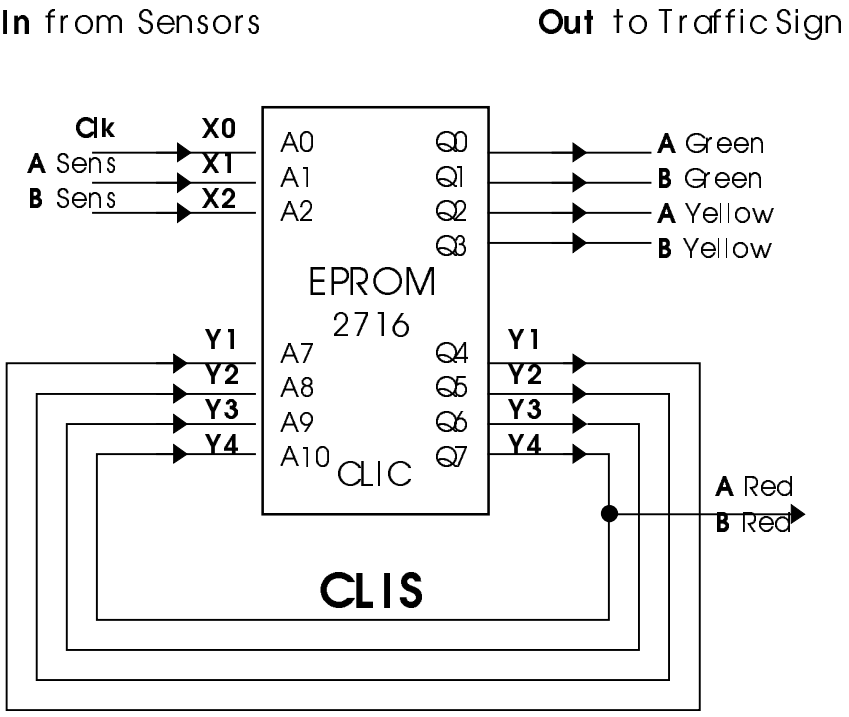
Some data of the first projects are important to the confection of new machines, like the use of “antibouncing” circuit to extinguish any state transition noise in the inputs. It is basic configuration is two logic NAND gates.



The state changing sensor is made of a logic door NOR EXCLUSIVE with a integrator circuit in the input. This circuit will produce a pulse in each transition state of the input, with a determined duration made by the integrator.



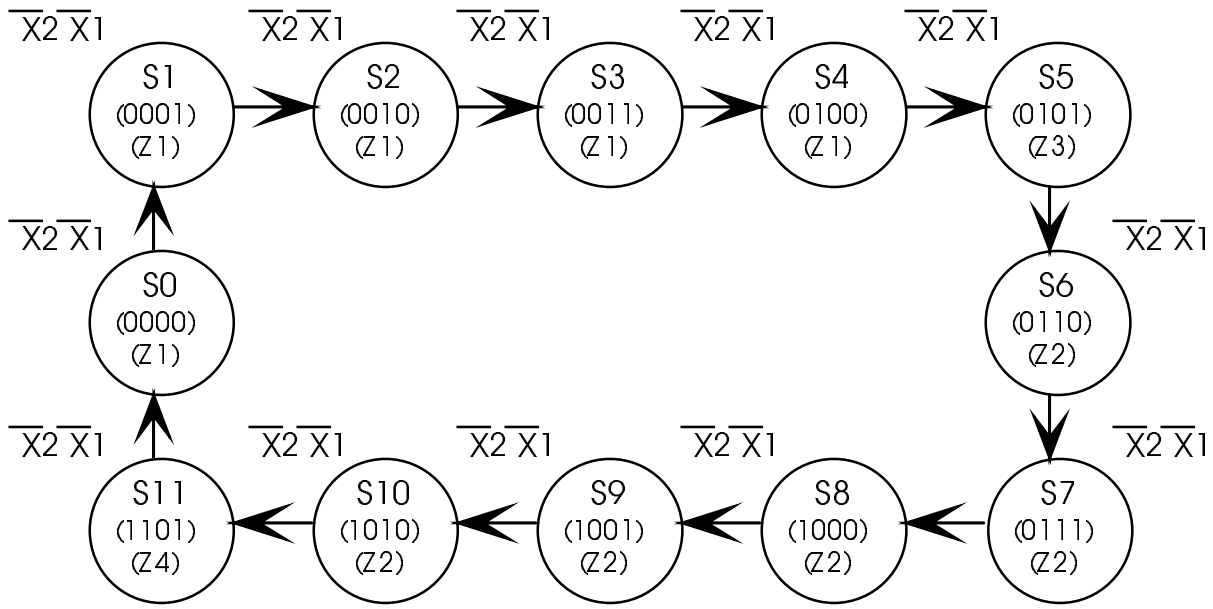
Was used a gate called Not Or Exclusive, because the pulse will give a monostable that has the right duration to the EPROM memory transition.
With this datas at hand, was assembled the first implementation of a **ESÇÃO** machine made by me. So I used a published article in a magazine called "Ciência & Cultura" from the author W. W. Martins, who tells about the Eprom as a logical device, universal and programmable. The article tells about a assembling of a traffic light controller using only a EPROM 2716, which basic plan can be observed below.



This traffic light would control the traffic flow in a street , but not in a crossing, because has the capacity to manage only two traffic lights. In functioning, as we have 4 feedback variables, we will have 2^4 **EIE**. So each one of this states will determine an output in a traffic light. Was accomplished then, a correspondent output at each state of a traffic light in a table.

	Memos (EIE)				**Power Outputs**						
Discrimination	**Y4**	**Y3**	**Y2**	**Y1**	**Z1**	**Z2**	**Z3**	**Z4**	**Z5**	**Y4**	**Sk**
1st street A Green	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	S 0
2nd street A Green	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	S_1
3rd street A Green	Ø	Ø	1	Ø	1	Ø	Ø	Ø	Ø	Ø	S_2
4th street A Green	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	S_3
5th street A Green	Ø	1	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	S_4
Street A Yellow	Ø	1	Ø	1	Ø	Ø	1	Ø	Ø	Ø	S_5
	Ø	1	1	Ø	-	-	-	-	-	-	-
	Ø	1	1	1	-	-	-	-	-	-	-
1st street B Green	1	Ø	Ø	Ø	Ø	1	Ø	Ø	1	Ø	S_6
2nd street B Green	1	Ø	Ø	1	Ø	1	Ø	Ø	1	Ø	S_7
3rd street B Green	1	Ø	1	Ø	Ø	1	Ø	Ø	1	Ø	S_8
4th street B Green	1	Ø	1	1	Ø	1	Ø	Ø	1	Ø	S_9
5th street B Green	1	1	Ø	Ø	Ø	1	Ø	Ø	1	Ø	S_{10}
Street B Yellow	1	1	Ø	1	Ø	Ø	Ø	1	1	Ø	S_{11}
	1	1	1	Ø	-	-	-	-	1	Ø	-
	1	1	1	1	-	-	-	-	1	Ø	-

After was defined the **EIE** for each program, or for each one of the independent inputs. To avoid increasing the subject too much I will talk about one of them only, to later get all the necessary conclusions in the next chapter.
The first program is a normal sequence operation of a traffic light to the X_1 and X_2 inputs in a low level logic, there is a GRAPH with the sequence, or path, that EPROM will follow during the variations of $X_Ø$, who is the system clock.
This is the graph:



With this I also can tell the mentioned "Project Matrix", that has the change than EPROM execute inside the state table. This table resume all the informations to get the FIRMWARE for a direct EPROM recording. Inside contains:

- Stable Internal States (**EIE**)
- Instable Internal States (**EII**)
- The clock pulse variation
- The traffic light variation output

If I set it to a truth-table, having the EPROM's address, I will get the as expected FIRMWARE.
Having been done all the necessary tables to the programming, I went to the traffic light hardware development. This circuit uses the concepts mentioned above to solve the problems regarding change of states. We use a IC 74174 (Hex flip-flop's D with clock and common reset to all elements). Will be turned on by a monostable that will give a pulse to the FF Clock, with a enough time to the EPROM's output conversions.
In the $X_{Ø}$ input It has a **IC 555** working as stable in frequency from 1Hz and the other it has X_1 and X_2 anti-bouncing circuits already spoken. To exit by being an experimental assembly, LED's was installed through buffers, because the maximum output current from an eprom is only 10 μA. When then use for this the **IC 74LS373**.

7. THE GREEN WAY PROTOTYPE

The sample was selected by being a present idea in mind of some people, but still not find a better way to implementation. So, the logic **ESÇÃO**, served like a glove to this project, as a excellent implementation to our research.
To implant the idea together with the **ESÇÃO** logic initially was developed the physical of the project, related to the theoretic Cinemathics and the right adaptations to a real world. To this I got the help from the Liberato Foundation teacher Mr. R. Rodrigues, who gave us a tremendous help for the principle development.
It deals about an intelligent traffic controller system, that inform to the drivers the speed that they must keep to reach the next traffic light in the green signal. Respecting the speed limit indicated by the system, time will be saved, because stops in the crossings would be avoided, also saving fuel and brakes. In general, we'll gonna find a harmonious traffic (like a maneuver in martial band) without congestions and less accidents, because the driver is obliged to not exceed the speed limit. Also, the pedestrians are going to have more safety and be correct about the time available to cross the street. Everybody are gonna find a *Green Path*. Little by little the drivers are going to start to understand the uncountable advantages of a disciplinary traffic, thus the system has something didactic and educational.
During the night, will not be necessary the use of a blinking yellow, because with the *green way*, there is no reason to stop in the red. The driver will know which speed he must keep to find the traffic light in the green, diminishing the attempts of assault. So, the safety in the cities will also increase. By the way, the yellow lost the function because the need to pay "attention" starts several meters before the traffic light.
Being a system based in **real time**, it is, without a own synchronism, always will be adjusted with the environment. There is not necessity to make alterations in the traffic lights that operates nowadays. However nothing impair a **ESÇÃO** control of being created in a traffic light. The cells that were placed would not be damaged.
Thus, lets set three datas subdivisions of the **GREEN WAY**: physics, software-firmware and hardware.

7.1 Physical Description (Cinemathics)

Trying to get the speed that the **GREEN WAY** have to show, I accomplished together with a physic teacher some algebraic equation deductions, that will give the referred speed.
If I think that the traffic light time that the car have to pass between the green was phases, or else, rotating vectors, we could make an analyze.

$$\delta = \frac{2.\pi.t}{T} => t = \frac{\delta.T}{2.\pi}$$

If the car pass in 1, suggest.

$$v = \frac{d}{t_{verde}}$$

If the car pass in between 1 (exclusive) and 2 (exclusive), suggest.

$$v = \frac{d}{\Delta t + \delta}$$

where,

δ is the necessary addition, so that all the cars arrive in a 2s time interval.

$$v = \frac{d}{t_{verde} \rightarrow \delta} = \frac{d}{\frac{\varphi_{verde}}{2.\pi} - \frac{\varphi_{total}}{2.\pi} + \delta} = \frac{d}{\frac{T}{2.\pi}.(\varphi_{verde} - \varphi) + 2}$$

$$v = \frac{d}{\frac{T}{2.\pi}.(\varphi_{verde}) + 2}$$

If car pass between 2 (inclusive) and 2 (exclusive), suggest.

$$v = \frac{d}{\Delta t + \delta} = \frac{d}{(t_{verde} - t) + \delta} = \frac{d}{(T - t) + \delta} = \frac{d}{(2\pi - \frac{\varphi T}{2\pi} t) + 2}$$

$$v = \frac{d}{2\pi - \frac{\varphi T}{2\pi} + 2}$$

Remake: being the maximum speed 80km/h (40nm)

a) the shorter possible period is:

$$\frac{d}{\frac{\varphi_{verde} \cdot t_{mín}}{2\pi}} = 22,2 m/s \Rightarrow d = 2,2 - \frac{\varphi_{verde} \cdot t_{mín}}{2\pi} \Rightarrow t_{mín} = \frac{2\pi \cdot d}{22,2 \cdot \varphi_{verde}}$$

b) the largest possible period is:

$$\frac{d}{\frac{\varphi_{verde} \cdot t_{máx}}{2\pi}} = 11,1 m/s \Rightarrow t_{má} = \frac{2\pi \cdot d}{11,1 \cdot \varphi_{verde}}$$

Then I can define a distance interval, where the green way works with a good confiability. The break (interval) is explained in a equation below, explaining a system with the times.

- Yellow time 31s
- Green time 43s
- Red time 57s

$$\varphi = \frac{2\pi \cdot t}{T} \Rightarrow \varphi_{verde} = \frac{2\pi \cdot 74}{131} = 3,55$$

Having a settled period, I can deduct the distance.

$$T = \frac{2\pi \cdot d}{v \cdot \varphi} \Rightarrow \frac{2\pi}{v \cdot \varphi} \cdot \frac{d}{1} \Rightarrow$$

$$\frac{1}{d} = \frac{2\pi}{v \cdot \varphi} \cdot \frac{1}{T} \Rightarrow$$

$$\boxed{d = \frac{v \cdot \varphi \cdot t}{2\pi}}$$

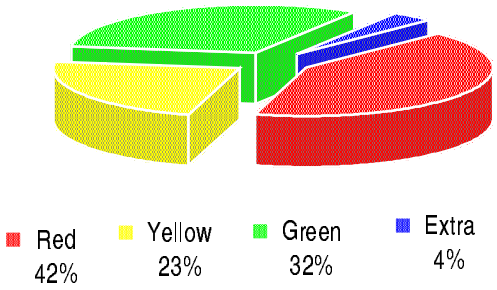
80km/h

$$d_{máx} = \frac{T.\varphi.v_{máx}}{2\pi} \Rightarrow$$
$$= \frac{131 \cdot 3{,}55 \cdot 22{,}2}{2\pi} \Rightarrow$$
$$= \frac{10322m}{2\pi} = 1643m$$

40km/h

$$d_{mín} = \frac{T.\varphi.v_{mín}}{2\pi} \Rightarrow$$
$$= \frac{131 \cdot 3{,}55 \cdot 11{,}1}{2\pi} \Rightarrow$$
$$= 821{,}6m$$

In practical terms, the system cannot obey this equation. This is due to the fact of following them, I will get a critical area where the signal will change only when the car will be passing below the traffic light. Which as a fear, the drivers are used to lower the speed when getting close to a traffic light in the red band. So I will make happen that the drivers will see the green light before reaching the semaphore. This, I will make happen that the drivers will see the green light before reaching the sign. Thus, I will make the system to show a speed that leads the car to pass under the traffic light after a time that the traffic light is in the green band.
On of the most difficult time, was to find the exact moment for the drivers to reach the traffic light without having a agglomeration of cars without allowing the speed to become a cycle that would impair the system.



7.2 Software / Firmware

Trying to ease the state table generation to an EPROM, a program in a PASCAL program was made that, having a time basis of Red, Green & Yellow, the distance from semaphore to the cell and local speed limit, give the speeds that will be recorded in the EPROM. Pascal language was used because it obliges the structure programmatic and have strong variable typing, simplifying the paperworks and understanding the program towards the required changes.
Executing the program, maximum speed will be asked for in the place where the **GREEN WAY** is installed and the Red, Yellow & Green bands. After the program suggests a distance band where the **GREEN WAY** can operate with confiability, and asks how many cells (**GREEN WAY**) will be used and the distance from the traffic light. The program will only accept if the distance be greater than the latest one, avoiding a sum error. After that, the program gives, for every **EŞÇÃO** machine, a table having time, starting in Green (zero instant) and the respective speeds as a decimal or hexadecimal.
For the speed calculations, the program used the equations got from physics, and the found speeds are kept in a vector, whose index is equal to the traffic light period. So that this vector can be filed in disc as a text, and the text can be accessed in the program that record the EPROMs.
Below we have a table created in this algorithm to:

1000m distance **120 s total period**

80s green period 35s yellow period 35s red period

	0	1	2	3	4	5	6	7	8	9
00	40	40	40	45	45	45	50	50	50	55
01	55	55	60	60	60	65	65	65	70	70
02	70	75	75	75	80	80	80	80	80	80

03	40	40	40	40	40	40	40	40	40	40
04	45	45	45	45	45	45	45	45	45	45
05	50	50	50	50	50	50	50	50	50	50
06	55	55	55	55	55	55	55	55	55	55
07	60	60	60	60	60	60	60	60	60	60
08	65	65	65	65	65	65	65	65	65	65
09	70	70	70	70	70	70	70	70	70	70
10	75	75	75	75	75	75	75	75	75	75
11	80	80	80	80	80	80	80	80	80	80

Program in basic used to get the FIRMWARE from a n° 2 EPROM.

```
1Ø   CLS                              16Ø  FOR X=2 TO STEP 4
2Ø   D=Ø                              17Ø  P=X+2
3Ø   INPUT  ”Qual  o  período:“       18Ø  A(X,2)=A(P,2)
;Q                                    19Ø  NEXT X
4Ø   DIM A(512,2)                     2ØØ  FOR X=3 TO Q STEP 4
5Ø   Z=1                              21Ø  D=X+2
6Ø   FOR X=Ø TO Q                     22Ø  IF D>Q THEN GOTO 18Ø
7Ø   A(Z,1)=X                         23Ø  A(X,2)=A(D,2)
8Ø   Z=Z+1                            24Ø  NEXT X
9Ø   NEXT X                           25Ø  FOR X=1 TO Q
1ØØ  FOR X=1 TO Q STEP 4              26Ø  PRINT A(X,1)
11Ø  A(X,2)=FIX(A(X,1)/2)             27Ø  PRINT A(X,2)
12Ø  NEXT X                           28Ø  PRINT “--”
13Ø  FOR X=4 TO Q STEP 4              29Ø  NEXT X
14Ø  A(X,2)=FIX (A(X,1)/2)            3ØØ  END
15Ø  NEXT X
```

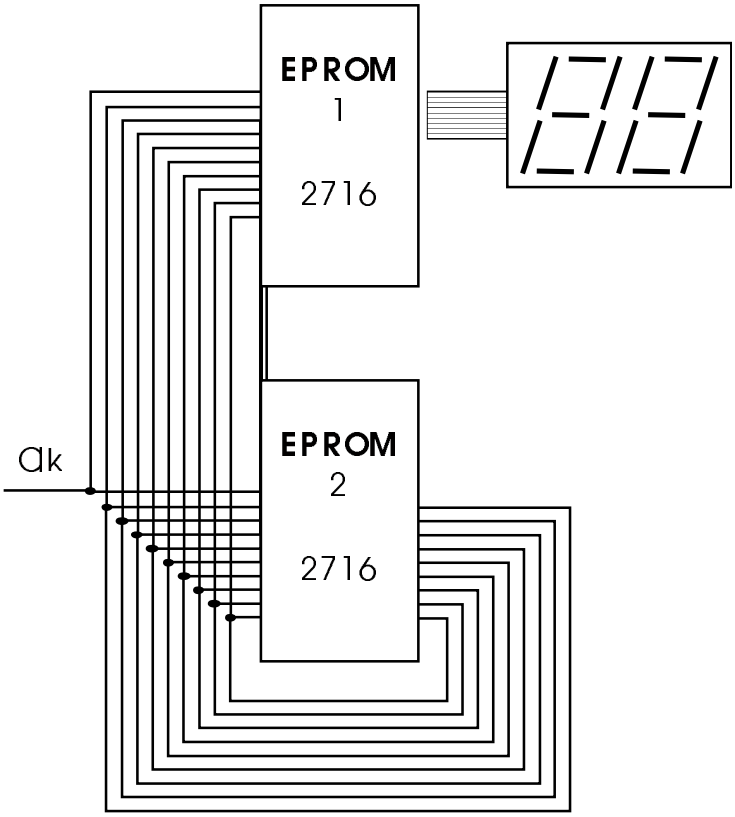
This program has the intention to generate the recording table from a n° 2 EPROM of a **GREEN WAY** project. As shown above, the language used was a BASIC in a microcomputer MSX EXPERT version 1.1.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
00	00	01	02	01	02	03	04	03	04	05	06	05	06	07	08	07
01	08	09	0A	09	0A	0B	0C	0B	0C	0D	0E	0D	0E	0F	10	0F
02	10	11	12	11	12	13	14	13	14	15	16	15	16	17	18	17
03	18	19	1A	19	1A	1B	1C	1B	1C	1D	1E	1D	1E	1F	10	1F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
04	20	21	22	21	22	23	24	23	24	25	26	25	26	27	28	27
05	28	29	2A	29	2A	2B	2C	2B	2C	2D	2E	2D	2E	2F	20	2F
06	30	31	32	31	32	33	34	33	34	35	36	35	16	17	18	17
07	38	39	3A	39	3A	3B	3C	3B	3C	3D	3E	3D	1E	1F	10	1F
08	40	41	42	41	42	43	44	43	44	45	46	45	26	27	28	27
09	48	49	4A	49	4A	4B	4C	4B	4C	4D	4E	4D	2E	2F	20	2F
0A	50	51	52	51	52	53	54	53	54	55	56	55	06	07	08	07
0B	58	59	5A	59	5A	5B	5C	5B	5C	5D	5E	5D	0E	0F	10	0F
0C	60	61	62	61	62	63	64	63	64	65	66	65	16	17	18	17
0D	68	69	6A	69	6A	6B	6C	6B	6C	6D	6E	6D	1E	1F	10	1F
0E	70	71	72	71	72	73	74	73	74	75	76	75	26	27	28	27
0F	78	79	7A	79	7A	7B	7C	7B	7C	7D	7E	7D	2E	2F	20	2F
10	80	81	82	81	82	83	84	83	84	85	86	85	06	07	08	07
11	88	89	8A	89	8A	8B	8C	8B	8C	8D	8E	8D	0E	0F	10	0F
12	90	91	92	91	92	93	94	93	94	95	96	95	16	17	18	17
13	98	99	0A	99	9A	9B	9C	9B	9C	9D	9E	9D	1E	1F	10	1F
14	A0	A1	A2	A1	A2	A3	A4	A3	A4	A5	A6	A5	A6	A7	A8	A7
15	A8	A9	AA	A9	AA	AB	AC	AB	AC	AD	AE	AD	AE	AF	A0	AF
16	B0	B1	B2	B1	B2	B3	B4	B3	B4	B5	B6	B5	B6	B7	B8	B7
17	B8	B9	BA	B9	BA	BB	BC	BB	BC	BD	BE	BD	BE	BF	B0	BF
18	C0	C1	C2	C1	C2	C3	C4	C3	C4	C5	C6	C5	C6	C7	C8	C7
19	C8	C9	CA	C9	CA	CB	CC	CB	CC	CD	CE	CD	CE	CF	C0	CF
1A	D0	D1	D2	D1	D2	D3	D4	D3	D4	D5	D6	D5	D6	D7	D8	D7
1B	D8	D9	DA	D9	DA	DB	DC	DB	DC	DD	DE	DD	DE	DF	D0	DF
1C	E0	E1	E2	E1	E2	E3	E4	E3	E4	E5	E6	E5	E6	E7	E8	E7
1D	E8	E9	EA	E9	EA	EB	EC	EB	EC	ED	EE	ED	EE	EF	E0	EF
1E	F0	F1	F2	EF	F2	F3	F4	F3	F4	F5	F6	F5	F6	F7	F8	F7

7.3 Hardware

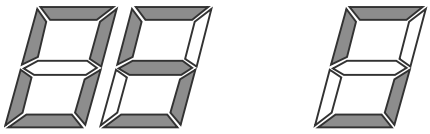
The hardware elaboration took me a week work till I get a right configuration to the project. Was a week of snaps in my brain, the result in a **first** prototype of a machine en a **EsÇão** generation, created in the Liberato Foundation!!.
Initially I knew that I will need two EPROMs to get a greater output number to the displays. Imagine a configuration that is shown below, with four feedback, a low frequency clock input and eight outputs to two BCD decodes to seven segments.



I realized the this configuration would be unsuitable and hard due to the following factors.

- The four feedback does not give only 16 **EIE**, when I need at least 60, which is the traffic light period
- the used of BCD-7 segments decodifiers will give increase cost, because I can use the own EPROM as a codifies

You could thin that if only 8 outputs we cannot connect two displays. Yes, you are right, but observe this part about car speedometer. In a car equipped with a speedometer.
We have a standard scale of 20 in 20 kilometers (Brazil). Starting in 0 or 20 towards the maximum speed. Between each speed there is a dash that should mean the intermediate speed. For example between the 40 and 60 there is a dash or line to means the 50km/h indication. So the speed precision that the driver can set is in 10 increments. So, I accomplished one that has the speed indication of 5 in 5 increments, to be used in the **GREEN WAY** systems, which is gonna be enough for drivers to keep themselves in a standard pattern. For 80 and 40, I will have 9 speeds, that will be shown 40, 45, 50, 55, 60, 65, 70, 75, 80. This example works for the Brazilian's expressways regulation, as the BR-116.
My prototype ideals to this area, expressways, so, going back to the speedometers, we had some segment that did not need to be controlled by an Eprom, because always stays turned on. The less significant algarism uses only the zero and 5 indication, allowing that ***a***, ***c, d*** and ***f*** segments to be always ON. The same happens to the most important, with ***c*** and ***f*** always ON. Observe that the next figure shows the ***zero*** and ***five*** digits, of the common segment



So, I need only eight **ESÇÃO** outputs to control the displays without the use of decodifiers. Related the feedback, the number of **EIE** is determined by the traffic light period, and not by the output states. So, as I have 1Hz clock, the number of **EIE** will be high. In my work, the maximum number that can be extracted is 2^8=256 **EIE**, being possible to help a traffic light of 256s. I know that the most traffic light does not exceed this value.
I went away then for a search of the suitable peripheral components, for the prototype. In TTL family, I needed the following devices.

a) 8 D flip/flop with clear and common clock

b) 1 monostable

c) 1 stable

d) 2 gate XOR

e) 8 current drivers

f) 1 D flip flop

To the item, I decided to use the IC 74LS273, because it is characteristics met the need ones. It has 20 pins having 8 D Flip Flop with a active master reset in low level and common clock input to all elements. The input current level is 20μA and output we can get up to 1mA without problems.
The b and c items, refer to a component: the **IC 555**. One for the monostable and other to astable of 1Hz. The last item (f letter) is used to guarantee a active cycle of 50% in the astable, because the **IC 555** does not guarantee level equals to one level. So, our low frequency oscillator, works in a 2Hz.
The 2 XOR gates (item d) holds the changing level dectetor cell, who manage the input variations. Was utilize the **IC 7486** to the job by being the simplest in the market.
In the end, the current driver was the most difficult found. Searching in Data Book TTL, I found the **IC 74F621**. I could not use it because I did not find in the market and also present's the disadvantage of having a open collector, where as would increase the cost with pull up resistors. Searching more, we found **IC 74LS241**, that gives little input currents (10 μA) and output current (24 mA), for me to active our display. In the case of a common display, just change **IC 74121** by a **IC 74244**, whose only difference is in the opposite of the output.

7.4 Developed Analysis

7.4.1 Driver Behaviour

During the ideas development, I put in practical a discussion related to the drivers behaviour in front of a traffic sign. Was an informal research between my colleagues and teachers, but that contribute a lot in the **GREEN WAY** concept organizations.
The most interesting one happened when I told a driver the existing of a system that would inform the right speed to be keep to reach the semaphore in the green light. He asked when the system would be implanted, immediately accepting the idea.
I also made a research about the human reaction towards a traffic light signal. When seeing a open traffic light (green) the drivers tend to speed up to avoid a possible stop in the red light. By the other side, realizing that the semaphore is closed (red) the drivers tend to slow down, begging to change to a green light.
The curious is that for a experiment driver this reaction happens automatically. In the case of seeing a yellow, the thing changes a bit. Then I get two kinds of drivers, the reckless and the cautions one. The reckless will speed up to cross in the last second, the cautions deals it as it was a red signal.
This datas gave a new heading to the **GREEN WAY**, increasing the use of it. I included some new tolerances to be given to the drivers:

a) Avoid the occurrence of state changes in the area seen by traffic light/driver

b) Avoid that given speeds takes the drivers to cross in a yellow signal.

The first item say that a greater time should be given after red to green, because would be disastrous, despite of legal, to someone cross the t. Light in the time of it's changing state. I also had to consider the vehicles with from 90° angle.

7.4.2 Cost Statistics

The **GREEN WAY** prototype were determined in the *PNP Electronic Components Co*, located in Novo Hamburgo. It deals only with the hardware part, without no worrying to the cost of the instruments use, like EPROMs recorders, computers.

Amount	Discrimination	Price	Total
2	EPROM's 2716	3,95	7,90
1	CI 74LS241	1,50	1,50

Amount	Discrimination	Price	Total
1	CI 74LS74	1,40	1,40
1	CI 74273	1,35	1,35
1	CI 7486	0,95	0,95
2	CI 555	0,80	1,60
1	Voltage Regulator 7805	0,95	0,95
1	Dual Common Anode Display	5,50	5,50
13	Resistor 1/8 W	0,02	0,20
5	Ceramic Capacitor	0,08	0,40
1	Eletrolitical Capacitor	0,80	0,80
1	Printed Circuit Board	10,00	10,00
20m	Wire Wrap	0,55	11,00
	TOTAL		**43,55**

In the case of building a machine with the same application using a micro processed system (Von Neumann Bottleneck), I would have to get the following components.

Amount	Discrimination	Price	Total
1	Microprocessor Z80	3,30	3,30
1	EPROM's 2716	3,95	3,95
2	RAM 6116	5,00	10,00
1	PIA 8255	6,60	6,60
2	CI 7400	1,00	2,00
2	CI 7404	1,00	2,00
1	CI 74373	1,45	1,45
1	CI 74244	1,45	1,45
1	CI 74138	1,15	1,15
1	CI 7486	0,95	0,95
1	CI 555	0,80	0,80
1	Voltage Regulator 7805	0,95	0,95
2	Dual Anode Display Common	5,50	11,00
22	Resistor 1/8 W	0,02	0,33
15	Ceramic Capacitor	0,08	1,20
3	Eletrolitical Capacitor	0,80	2,40
1	Printed Circuit Board	30,00	30,00

Amount	Discrimination	Price	Total
20m	Wire Wrap	0,55	11,00
	TOTAL		**90,53**

8. THE A/D ESÇÃO CONVERTER PROTOTYPE

The **ESÇÃO** architecture allows events in a real time. A application that could ask for high speed is the analogic signal conversion to the equivalent binary. Being in binary, can better be processed and evaluated in quantity and quality. A important detail is that **ESÇÃO** is strongly based in memory. So, allows that advanced algorithm, such as binary search be implanted.
The main difference of the **ESÇÃO ANALOGIC/DIGITAL CONVERTER** and the traditional system is the conversion method. The usually uses only hardware implementation, with electronic circuits. The **ESÇÃO** System implements an algorithm solution. The software will predict all steps of the hardware A/D converters. The most important to the comprehension of this converter is to know the algorithm.

8.1 The Binary Search

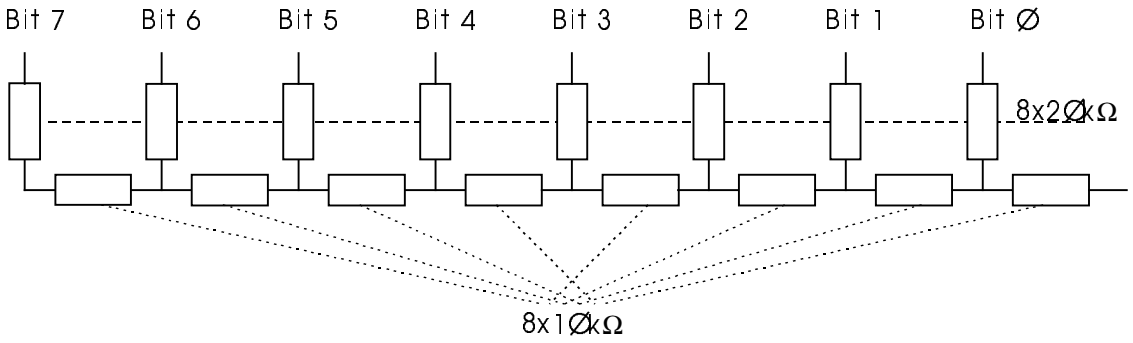
The binary search is a sort algorithm faster and better in efficiency. It's principles is to share the half problems, and each half in the half, until the point where the problem is solved. The implementation of a binary research, to a proposal application, would be impossible by the electronic circuit and very hard of a micro processor, in the right speed.
The **ESÇÃO** system allow the use of binary search to discover the digital correspondent of a analogic value. All the values are treated and record in the **ESÇÃO** firmware. If compared to a common dedicated system, is much more dynamic, despite of being cheaper than a micro processed system.
To generate the firmware all steps of the usual converter are simulate. The result of each answer is saved in a file. The method of Binary Search, or Quick-Sort, for discover the values, uses the pivot, that is the medium value. For example, an application to converts in interval of Ø-5V, primarily will be generate the pivot in 2.5V. The next value depends of input, and this compute depends if is higher or lower than reference.

$$Suggest = pivot \pm \left(\frac{pivot}{2} \right)$$

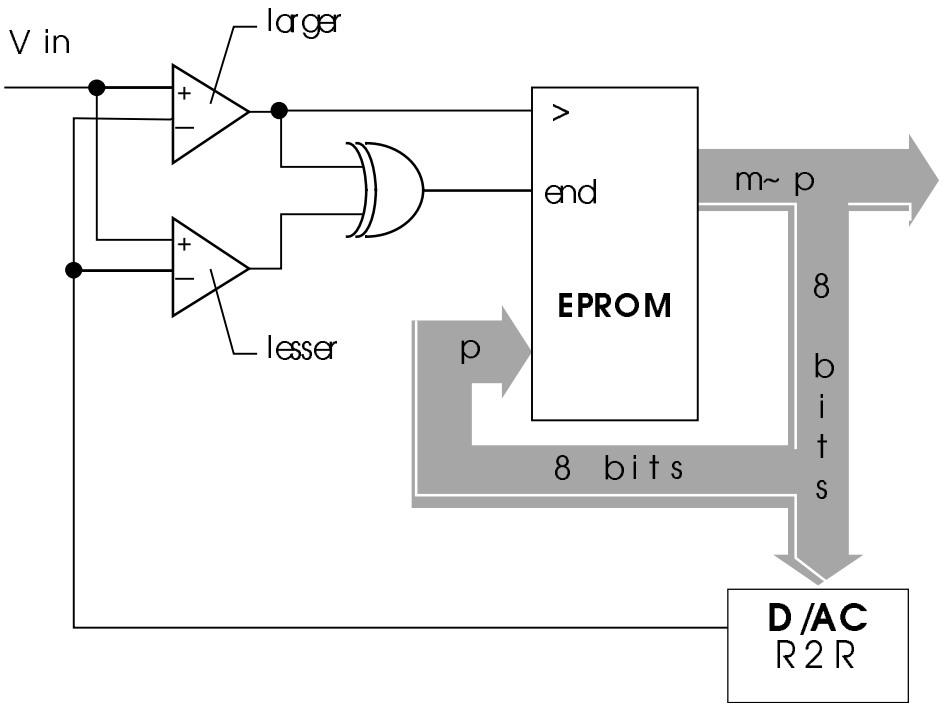
Still is needs more one information: if the input is equal to pivot. This generate one bit flag in the binary search, which indicate the finish of conversion.

8.2 Hardware

The most complex job is executed for the software, then the hardware is very simple. The bits of the search path (higher, lower and ready) are obtain for two voltage comparators, which compares the analogic value to the search against the value present in the EPROM data. The dependent variables [**Z**] are the suggested values to compare with the measured sample, after have been past by D/A resistors. The dependent variables are the same feedback variables, suggesting the **EII**.
To accomplish the binary hardware, a bit of EPROM address bus must be reserved to indicate whether the analogic value is bigger or shorter that the converted value. Just one bit is necessary to separate the halves.



In every suggestion, just one bit change. It characterize that would not be necessary. The use of MEMO, keeping off the possibility of Hazards. The suggested bit will be accepted or not by **ESÇÃO**, indicating if it is bigger or shorter than the sample.



8.3 Software

For the firmware confection, which is the EPROM program, was elaborated a algorithm that simulates what the converter would do. Realizing the simulation graphically, better can evaluate the A/D **ESÇÃO** converter functioning, really seeing a binary tree.
The following table shows the variables for a **4V** valve test.

R	>	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0		Suggest.
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	e	0
Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	i	2,5
Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	e	
Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	i	3,75
Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	e	
Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	i	4,375
Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	e	
Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	1	1	Ø	1	Ø	Ø	Ø	Ø	i	4,0625
Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	1	1	Ø	1	Ø	Ø	Ø	Ø	e	

R	>	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0		Suggest.
Ø	Ø	1	1	Ø	1	Ø	Ø	Ø	Ø	1	1	Ø	Ø	1	Ø	Ø	Ø	i	3,90625
Ø	1	1	1	Ø	1	Ø	Ø	Ø	Ø	1	1	Ø	Ø	1	Ø	Ø	Ø	e	
Ø	1	1	1	Ø	Ø	1	Ø	Ø	Ø	1	1	Ø	Ø	1	1	Ø	Ø	i	3,9843
Ø	1	1	1	Ø	Ø	1	Ø	Ø	Ø	1	1	Ø	Ø	1	1	Ø	Ø	e	
Ø	1	1	1	Ø	Ø	1	1	Ø	Ø	1	1	Ø	Ø	1	1	1	Ø	i	4,0234375
Ø	Ø	1	1	Ø	Ø	1	1	Ø	Ø	1	1	Ø	Ø	1	1	1	Ø	e	
Ø	Ø	1	1	Ø	Ø	1	1	1	Ø	1	1	Ø	Ø	1	1	Ø	1	i	4,00390625
Ø	Ø	1	1	Ø	Ø	1	1	1	Ø	1	1	Ø	Ø	1	1	Ø	Ø	e	
Ø	1	1	1	Ø	Ø	1	1	Ø	Ø	1	1	Ø	Ø	1	1	Ø	1		Finish

Occurs that any conversion of a analogic value, is the worst hypotheses, of eight iterations and given that the speed reached by **EsçÃo** was of 2.8 Mhz there is watchers of a complete conversion of 8 bits of 350 ns. As it is much faster than a typical commercial it's probably to be necessary a latch as a intermediate converter memory to compensate with the microprocessors.

9. CONCLUSIONS

The **EsÇÃo** logic brings advantages in a change of a microprocessed system due to the following reasons.

a) It has a low cost due to the comparing already shown in relation to the cost of **GREEN WAY** system. Where the main component — EPROM — get several functions eight a accessible cost.

b) Presents a versatile for implementations in adverse situations (temperature, pressure) seen that the components are not so touchy as complex microprocessed system

c) The processing is much faster than a microprocessor due to the elimination of the called Von Neumann Bottleneck, bringing a asynchronous functioning with the pre define answers in it's Know By Heart Department (**DDD**)

d) By being totally asynchronous, the **EsÇÃo**, that has a deterministic behavior of it's inputs, catch every and any variation independent the speed, leaving off the old clock of the present computers (how fast the be always leaves gaps in the perception input)

A **EsÇÃo** machine can, through a small circuit controller, solve very complex problems, whose for sure would be used a microprocessed controller. This characteristic come from the fact of having the possibility of recording several different programs in a unique memory device (usually a EPROM). Then, this Non Von Neumann Generation machine does not depend of last generation circuits to accomplish any kind of task, seen that we can associate general devices depending of the program complexity to be solved.

To implement this logic, the user must initially break the paradigms of microprocessors and think in accordance with a **EsÇÃo**. Despite of solving very similar problems, the programming of a microcontroller is a lot different from a Non Von Machine. In the first one, establish a lot of attitudes to be followed ([9]algorithm) to get the final product.

And the second is necessary to forecast all the desired results and organize them in the memory address in accordance with the sensors distributed in the inputs. The algorithm now is used before, in a program that generates the firmware. A

---

[9] Algorithm Rules to execute a explicit task or to solve a specific problem

analogy that can be use is the efficacy of interpreted programs against the compiled. A **ESÇÃO** is equal to a sequential circuit combinatory that was compiled.
The limitation of a **ESÇÃO** towards on microprocessed system is the fact of not having a inner clock, which impair the use of temporization to a eventual application. This problem can be solved with the use of a external timer, like the **GREEN WAY**, where we use a monoestable a stable low frequency with IC 555.
The development level of a **ESÇÃO** logic is too low compared to Von Neumann machines. This kind of computers architecture represents the other path to be followed in the world of informatic. We can do a analogy, in a automobiles area, where the man decided to create a combustion engine due to the fact of having plenty of oil. Today that we feel this is vanishing, we search alternatives as the electrical cars and so on. But this show several difficulties and problems. Same to the Non Von Neumann machines. Show the 'other side of the coin', but across to upgrade more the artificials thoughts of this thoughtful machines.
The area of computer without microprocessor is a tendency to be fowled inside of the **5th generations** computers development. Occurs that with the biochips development (informations and data stored in atoms), we can have a innate memory in a short space. So, that's why the **ESÇÃO** ($n \sim m \sim p$) start here to give it is **100% Brazilian** contribution to this new technologies.
This research deal only with a drop in a ocean. Still there are a lot to be researched and implemented inside this promising area. The main step to be followed will software elaboration capable of allowing a easier communication between the user and the **ESÇÃO COMPUTER**. Only to produce compiler, where the variables needs to be informed (***n***, ***m***, ***p***) to table parts of firmware.

9.1 Future Applications

As the quantity of proposal ideas increased very fast, will be required to create another work groups, each one responsible to a application. It is a good way to continue the research. The intend is to share the knowledge with another students in a technical area of a **ESÇÃO** research.

9.1.1 Stepper motor controls - mechanic arm

To the control of a step motor only needs position tables, by being simple sequences, can be used as a inner variable of a esção. The dependable variables that last, allow the activation of other programs (speed, reverting, half-step…). Positions sensors, strategically collocated, card activate these programs. A unique, simple, esção control system can to automate one of typical use of micro controllers.

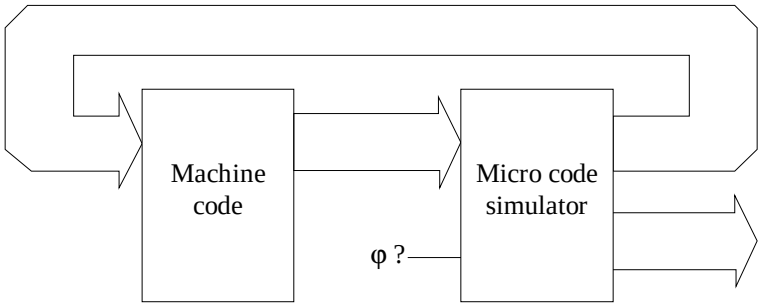
9.1.2 Ram **Esção**

Circuits integration in silicon, have reached around millions of devices per square centimeters, also for decreasing a lot the distance and the transistors, high speed are reached (better than 75ns, actually). But the memory still stays short, slow or to expensive. The EPROM technology practically stays the same.
The computers like the IBM-PC, uses a thing to increase the speed. Accomplish a copy from bios firmware in ram. In cache memories, i the motherboard. Some time, the minors stays in a internal cache of the microprocessor chip. Or, it's possible for certain moments, the firmware goes from EPROM to a encapsulated memory in the micro processor. So a great way to increase the performance of a system based in a esção architecture is substitute the EPROM by a ram, in the moment of initialize the system, staying in the ram during the whole system functioning.
New hypothesis are created, specifically to this job.

- Several programs can be used in ROM and copy in RAM only when needed to be executed?
- Using the **Esção** RAM, has the possibility of the own system recording values in the chip?
- It will provide treatment of memorized variables?
- ...maybe auto programming — 5th generation?
- Could be of individual address of bit or only byte?

9.1.3 Discrete Microprocessor

- If initiating the microcode table of a microprocessor, would be possible to create a instruction set to a **Esção** can be made a discrete microprocessor?



- The application could be only didactic or allows a develop of a Logical Programmable Controller, with only a standard hardware configuration execute several tasks with only the change of it is programming.
- It would be a recreation of the microprocessor
- It would be bit [10]slice?

9.1.4 Neural Network & Fuzzy

- The results form the learning of a neural net could be record in a **ESÇÃO** firmware.
- Could be a **ESÇÃO** system have topology and behaviour of the neural network?
- To use to advantage a configuration in RAM, would the neural **ESÇÃO** learn?
- Could be recorder in **ESÇÃO** firmware, algorithms of fuzzy logic.

9.1.5 Scrambler /Encoder

Trying to guarantee protection of informations at this time, we can use the information as steps of a **ESÇÃO** program.

- A **ESÇÃO** would deal with error detection codes, like hamming, for example?
- The own message would work as codification key?
- Analysis and redundancy generation are possible to a **ESÇÃO** ?

9.1.6 Esção Compiler

The main point of this work will be when there is operating a compiler, preferentially in a graphical interface. It only needs to be drawn the main elements (input number, outputs, feedback, memory in cascade, event sensors, memo latches) for the compiler start to generate the firmware tables. Informing the input and output events (**EIE**) to be completed the programming tables.

- Is it possible the compiler be bonded to a EPROM recorder?
- Could be created a EPROM's **ESÇÃO** recorder?
- The compiler necessarily needs to be created in OOP object oriented language?
- How to make the SPICE Language understing the Esção System?

9.2 Suggest Improvement - An IC Esção

What is the ideal device to a Esção System? Knows the the EPROM nowadays are:

- Slow

---

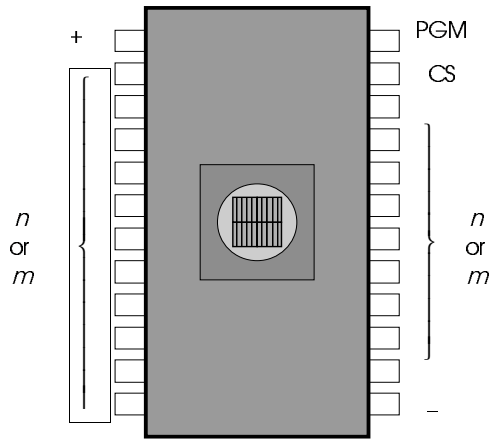
[10] Bit Slice A CPU construct with a wide size of word by union of the words blocks

- Eigth outpus limit
- Short memory by low cost
- Complex hardware to feedback variables link

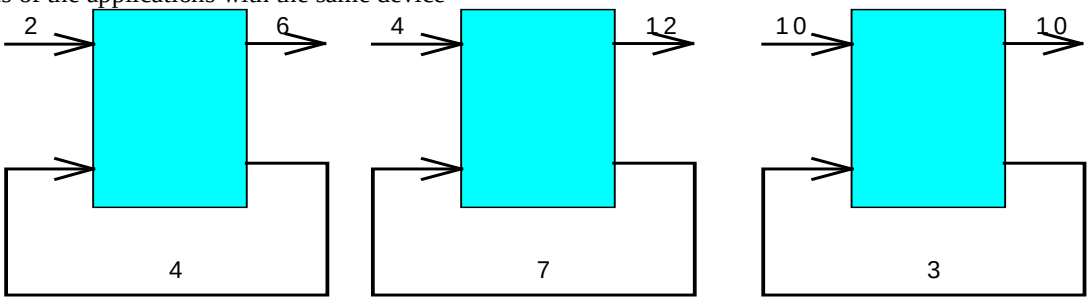
However, the perfect device to implementation of a Esção technology

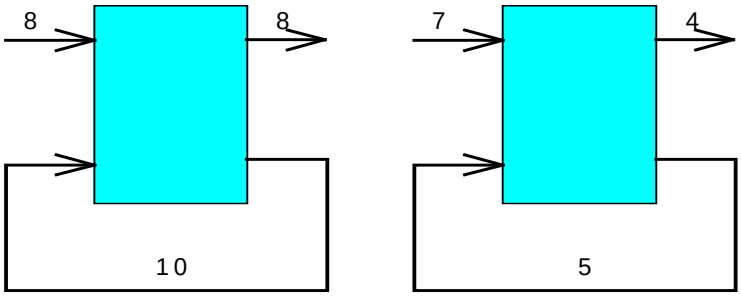
- Reprogramable (mix E²PROM-PAL)
- Progamable number of EIE
- Exclusively **n~m** leads in package
- Electrically erasable — could change the data in Esção (autoprogrammation?!?)

Suggestion of 24 pins package



Examples of the applications with the same device





10. CONCLUSION

"The dwarf see longer than the giant
when he climbs on the giant's shoulder"

The giant are the big nations and corporations with resource and advanced technology to improve more and more the research about semi conduction devices.
The dwarf from the 3rd world, needs to use a lot of creativity when against technological paradigms.
We see as the Science responsibility the rescue of a further research technology and already developed concepts but that could not be implanted, maybe because of being to advanced to the time.
However, the **ESÇÃO** architecture brings a paradigm. The programming way is exclusively in firmware. Similar to ENIAC project, from J Presper Eckert Jr., and John Mauchly, 1946, where were used a system of externally mounted switches and plugs to program it. So, having many improvements and the advanced of technology, why return? Is this a tribute to ENIAC 50th anniversary?
No, the science can not ignore the knowledge because there is a tendency about it or it is alternatives. Despite of appearing old, comparing towards a stored - Program Technology, now the firmware programming can be explored due to memory switches of large size and so compacts, cheap and of easy programming.
Furthermore, this research **recicle** an idea that was impossible to implement half century ago. So, the Esção Generation adds more a brick. It is possible this concepts will wait until Technology be prepared to use the General Model for a Digital System. But there is no problem: we are always in evolution.

11. BIBLIOGRAPHY REFERENCES

[1] AKONTEH, Benny, *Introdução à Organização e Arquitetura de Computadores Digitais*, ATI Press - Akonteh Instituto de Tecnologia. Brasília. 1983. 177 pág.

[2] ALMASI, George, GOTTLIEB, Allan, *Highly Parallel Computing,* 2nd edition, Benjamimn/Cummings Publishing Company Inc. pp 689.

[3] ALMEIDA, B. Hamilton, *O Outro Lado das Telecomunicações* — A saga do Pe LANDEL, Editora Sulina, Porto Alegre. 1983. 151 pág

[4] BENNATON, Jocelyn, *O que é Cibernética*. Editora Brasiliense. São Paulo. 1984. 88 pág.

[5] BONATTI, Ivanil e MADUREIRA, Marcos. *Introdução à Análise de Circuitos Lógicos.* Editora Unicamp. Campinas SP. 1990. 148 pág.

[6] BRADEN Jr, Willian, *Manual do Micro Computador Z80*, Editora Campus LTDA. Rio de Janeiro. 1985. 364 pág.

[7] CAMILO, Daniel et alli. Circuito Lógicos - *Teoria e Laboratório. 1ª Edição*. Editora Livraria Ciência e Tecnologia. São Paulo. SP. 1984. 413 pág.

[8] DIANESE, Antônio, *Computação e Simulação Analógica e Híbrida*, Ed Guanabara Dois, Rio de Janeiro. 1984. 251 pág.

[9] *Entenda o Computador - Microchips e Microcomputadores*, Editora Nova Cultural. São Paulo. 1988. 64 pág.

[10] GONICK, Larry, *Introdução Ilustrada À Computação* (com muito humor), Ed Habra. São Paulo. 1984. 242 pág.

[11] HARTNELL, Tim, *Simulaciones – Replica la Realidad en tu Ordenador*, Ediciones Anaya Multimedia S. A. Madrid. España. 1985. 247 pág.

[12] HIBBERB, Robert, *Circuitos Integrados - Curso Básico para Ingenieros y Técnicos*, Texas Instrument Incorporated, Ed. Marcombo S. A. Barcelona, España, 1974. 205 pág.

[13] INTEL EPROM's catalog, 1980

[14] MAMMANA, Carlos, ZUFFO, João, *Dispositivos e Circuitos Eletrônicos*, Editora McGraw Hill do Brasil LTDA. São Paulo, 1971. 271 pág.

[15] MANDA, F. F. *Circuitos Integrados: Tecnologia e Aplicações*. LTC - Livros Técnicos e Científicos. Rio de Janeiro, 1982. 221 pág.

[16] MARTINS, Wagner Waneck. *A Linguagem Numérica*, Revista Ciência e Cultura. S.B.P.C. São Paulo. 598 a 605, nº38(4), abril de 1986.

[17] MARTINS, Wagner Waneck. *A Matemática Booleana e os Pensamentos Humano e Artificial*, Revista Ciência e Cultura.;S.B.P.C. São Paulo. Biblioteca;do Departamento de Engenharia de Eletricidade, 427 a 431, abril de 1983.

[18] MARTINS, Wagner Waneck. *EPROM, Um Dispositivo Lógico Universal e Programável*, Revista Ciência e Cultura da S.B.P.C.. São Paulo. Biblioteca do Departamento de Engenharia de Eletricidade, 839 a 858, setembro de 1989.

[19] MARTINS, Wagner Waneck. ***ESÇÃO*** *(n~m~p) - Um Computador Não-Von Neumann*. 1ª Edição. Editora Cartograf. Campinas. São Paulo. 1985. 267 pág.

[20] MARTINS, Wagner Waneck. *Lógica Lingüística - 1ª parte*, Revista Ciência e Cultura. S.B.P.C. São Paulo. Biblioteca do Departamento de Engenharia de Eletricidade, 1331 a 1349, nº36(8), agosto de 1984.

[21] MARTINS, Wagner Waneck. *Lógica Lingüística: Dedução de Textos - 2ª parte*, Revista Ciência e Cultura. S.B.P.C. São Paulo. 584 a 600, nº37(4), abril de 1985.

[22] MARTINS, Wagner Waneck. *Software para Circuitos Lógicos a Eprom*. Revista Ferrovia. São Paulo. Associação dos Engenheiros da E.F.S.J., ano 47, No 88, janeiro/fevereiro, 1983.

[23] MELLO, Hilota, *Circuitos Integrados*, Editora Edgar Blücher LTDA. São Paulo. 1976. 153 pág.

[24] MORAIS, Robert, MILHE, John, *Projetos com Circuitos Integrados TTL*, Texas Instrument Incorporated, Ed. Guanabara. Rio de Janeiro, 1978, 325 pág.

[25] PHISTER, Montgomery, *Logical Design of Digital Computers*, John Wiley & Sons, Inc. New York, 1958, 408 pag.

[26] POUNTAIN, Dick, *A Solução Multiprocessadora*. Byte Brasil Vol 2 N°11, 65 a 71. novembro de 1983

[27] PRATA, Stephen, *Divertindo-se com a Vida Artificial*, Berkeley Brasil Editora. São Paulo. 1993. 182 Pág.

[28] RONY, Peter R. Et alli, *The 8080A Bugbook Microcomputer Interfacing and Programming by Radio Shack*. Howard W. Sams & Co., Inc. Indiannapolis, Indiana, USA. 1977. 416 pág.

[29] STARK, Peter, *Micro-programed Comtrolers – bridging the gap between TTL and microprocessors*, 73 Magazine, Janeiro. 1982. Pág 68-84

[30] VLADIMIRESCU, A et alli, *SPICE User´s Guide Version 2G*, Dept. Of Eletrical Engineering and Computer Sciences, Univ. Of California, USA, August, 1981 55 pág

[31] ZACKS, Rotinay, *From Chips to Systems*, Sybex Inc. Berkeley. CA. USA. 1981. 552 Pág.

[32] ZILOG, *Z80 Product Specifications*, Doc Num 03-0027-02. março. 1978.10 pág

[33] ZILOG, *Z8000 Technical Overview*, Doc Num 03-8074-01. agosto. 1979.23 pág

[34] ZUFFO, Marcelo K., LOPES, Roseli D. *Multicomputadores: o futuro do processamento*. Byte Brasil Vol 2 N°11, 74 a 85. novembro de 1983

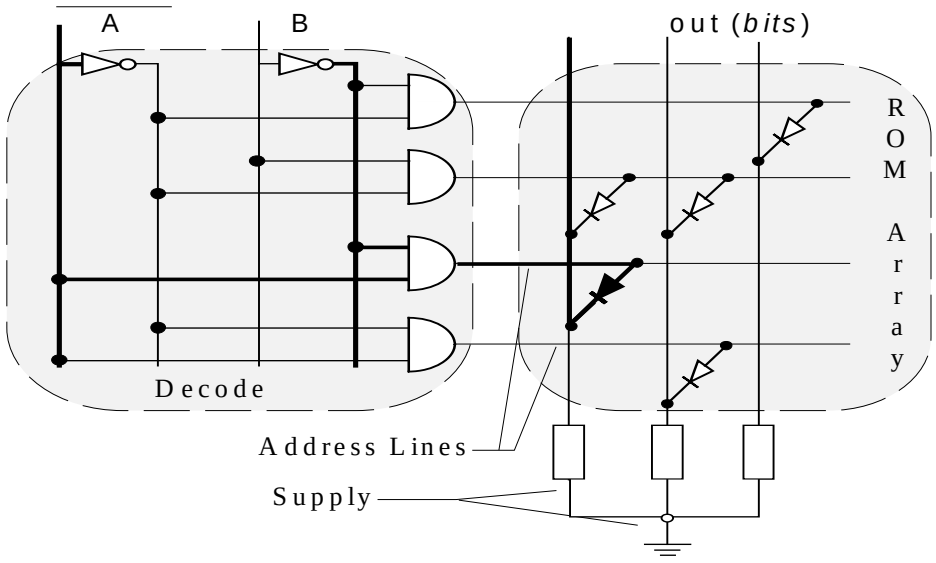
12. APPENDIX

12.1 The EPROM functioning

As it is assembled essentially by memory, for a **EsçÃo** is necessary to find a device that be non volatile, so the program in firmware will not be missed. However it should allow changes and re-record at least the prototypes. The only component existing that is non volatile and allows re-recording is the integrated circuit [11]EPROM. So, it becomes necessary a deep knowledge about the functioning and electrical characteristics.

12.2 Specimen

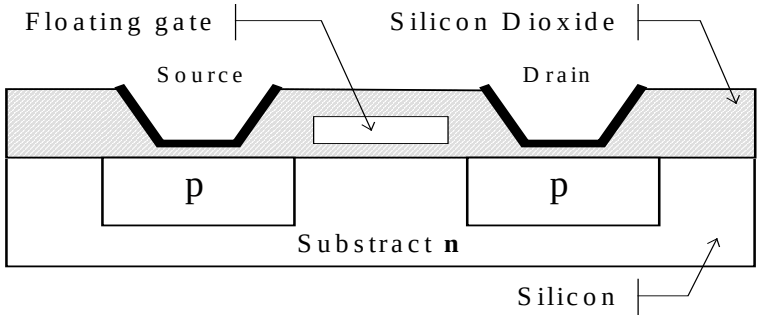
During the functioning, the **EsçÃo** will work as a read only memory. The ROM as Matrixes that allow electric current in a cell to a adress chosen. Only one bit arrangement outputs will be energized when the column will be addressed. In the figure example, considering that the adress AB is **1Ø**, only first column will be activated, resulting **1ØØ** in data out.



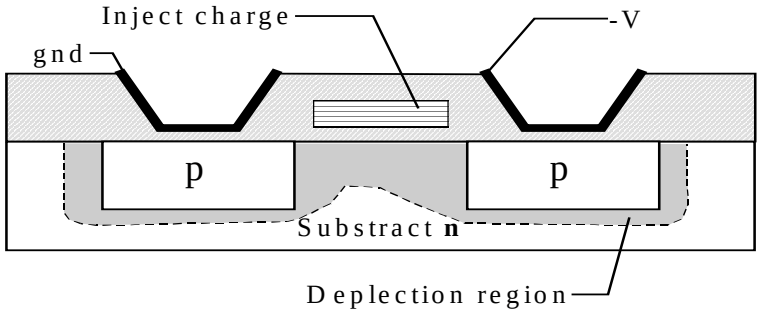
[11] EPROM Erasable Programmable Read-Only Memory

12.2.1 EPROM Construction

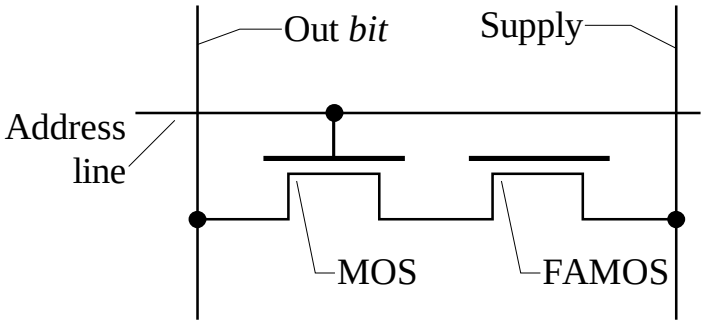
Each bit of a EPROM stays saved in a kind of transistor [12]MOS. By being MOS; guarantees that a large warehousing. but there is a difference. As MAZDA (1982) and the Data Sheet INTEL (1980) a EPROM has the floating door (famos floating avalanche MOS) composed of Si wrapped up in SiO_2.



In the recording moment, high tension is applicated, typically 25V, between the source and the drain. The Pn junction suffers avalanche causing a injection of electrons in the floating gate. By being surround by a low conductivity oxide, the load stays retain for a long time.



Each memory bit is stored in a couple of transistors, one MOS and another FAMOS. The MOS works as a qualifier, and stays in a arrangement in a matrix shape. When addressed, the transistor FAMOS completes the circuit with the MOS transistor, taking the bit to output, **1** or **Ø**, according to the value memorized in the floating sate.



[12] MOS Metal Oxide Semiconductor – production method and design for some integrated circuits family, using patterns of metal conductors an oxides depositing above a semiconductor

13. APPENDIX II

13.1 A Neural GREEN WAY— The Final Frontier

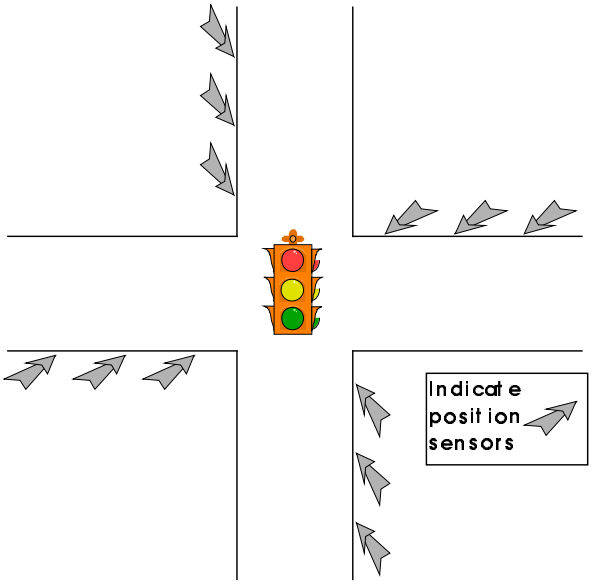
The Green Way model proposed is pretty simple and it is accomplishment does not bring changes in the traffic control method used nowadays. Only to drivers a change in behavior is required. But we got to be very optimist to consider that all the problems will be solved. Also because we are being based in the subject less adaptable that exist: *the mind and the human behavior*.
A good project would be not bring a new thing to the people. So, to this, all the project of traffic light should appear being the same, but after wards had been totally changed. Changed, of course, to serve the citizens.

13.2 New hypothesis

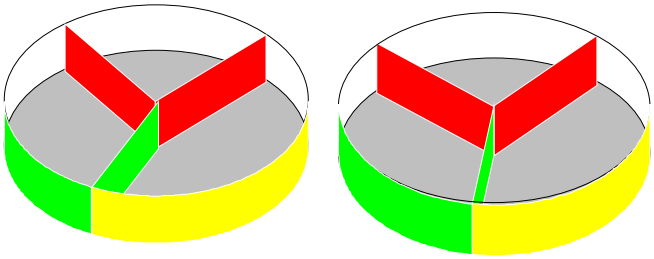
To accomplish the studies of a control system that would be completely transparent to the people, and then confortable to our minds, was necessary to create a new paradigm. Take out the responsibility of people action control as happens to normal systems. The **GREEN WAY** tries to educate the drivers, but it is only applicable to expressways.

If there were a system that could observe all the traffic flow and dynamically adapts the traffic light time, the lines would be decreased and traffic jams avoided?

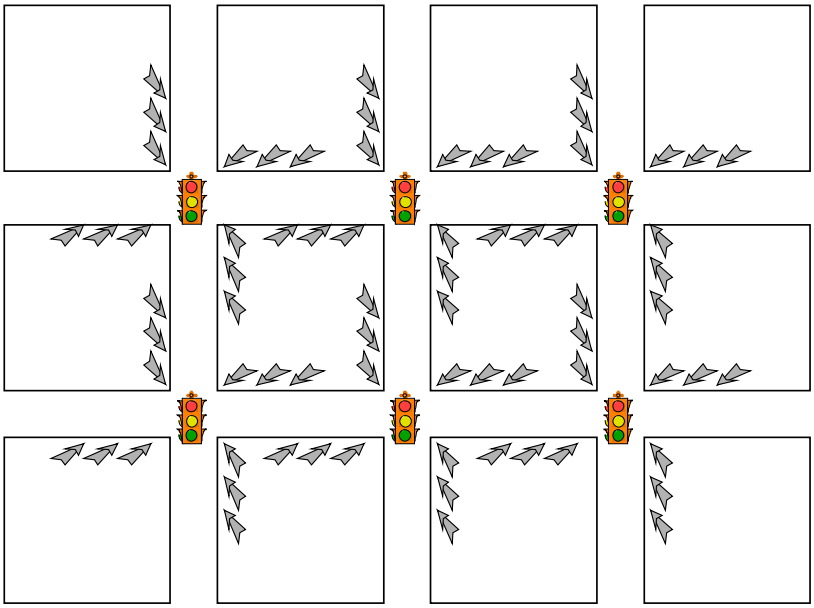
Reviewing, the **EsÇÃO** allows a complex application being decentralized and with intelligence in each one of the cells. Let is imagine that every corner a **EsÇÃO** receives informations of how many cars are coming in each street. It could be done by optics sensors , ultra-sound, magnetic or of hall effect.



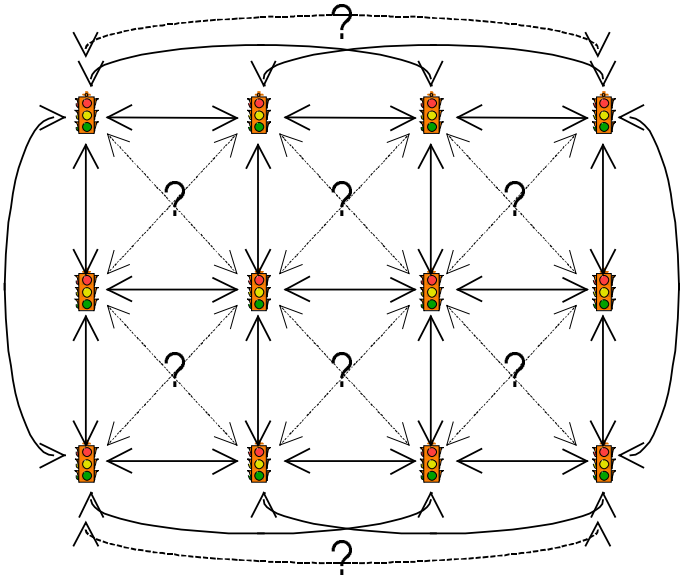
The **EsÇÃo** program would be consider the normal time of traffic lights, and would give priority to the most crowded street. In the example, there are 12(twelve) inputs (dependable variables), despite of temporal variable (X0) to the permission control time of each street . The sensors analyze if are there any cars stopped, waiting for the change from red to green. The programs evaluate how many are there and could change the times vector.



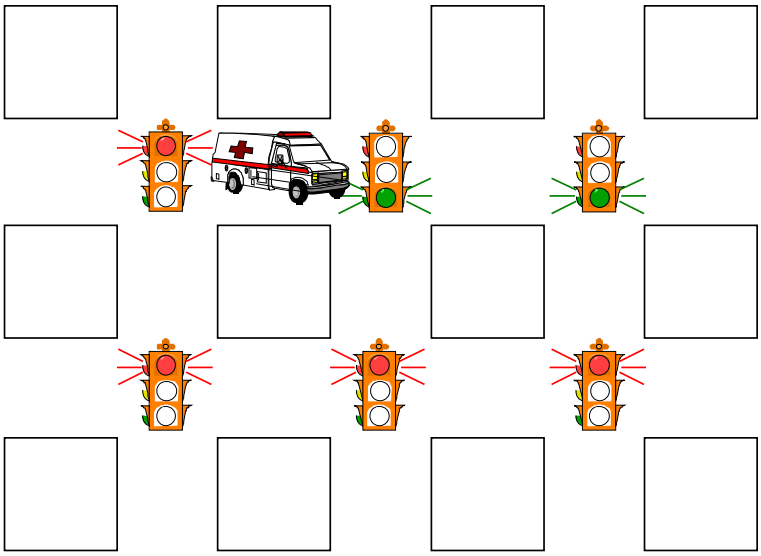
As all corners would have similar cells, they would communicate themselves by the vehicles flow! They would adapt the times according to traffic intensity.



This configuration would have resemblance with a neural network. Especially with the HOPFIELD topology, because the informations change is multidimensional. Probably a limit exist of how many "layers" a cell gets the knowledge, i.e., till the point where a cell action influences the evaluation of others.



Also could be considered, asking for ped xings and could be installed horn sensors (police, firemen and ambulances) that would change the times, stopping completely the others traffic lights giving the priority to the public service vehicle.



The creation of a unique prototype would be of low return to a accurate evaluation. The camp implementation, with several prototypes in a real condition, could be critic and would jeopardize the people safety. Actually, it was the initial idea of Green Way. But a preliminary study showed some difficulties, already spoken. A simulation program must be crated, as it can be studied in HARTNELL (1986), preferentially used objects (OOP) and with details learned in PRATA (1994), with down-top modeling, local rules – global behaviour, emergency behaviour.