A Solution of the General Model for a Digital System

Filipi Damasceno Vianna
Pontifical Catholic University of Rio Grande do Sul
E-mail: filipi@em.pucrs.br [*]

Daniel Fink
Pontifical Catholic University of Rio Grande do Sul
E-mail: dfink@crt.net.br

José Inácio Coelho
University of Rio dos Sinos Valey
E-mail: jic@aquila.com.br

Porto Alegre, May, 1998.

1 Introdution

The intent of this research is presents the Logical **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 Neumanns Bottleneck* from the current computers. Enable, also, a deterministic perception of the variable inputs and working in real time. From its datas, the theoretical aspects of this new computing system was investigated and tried to use it in lots of systems to confirm its functioning and evaluate the advantages against the conventional controllers.

The significance of this work is the contribution for a search to eliminate the Von Neumann bottleneck from the present computational architecture. It deals by putting in practice the "General Model for a Digital System", (PHISTER, 1958)[**?**], that has a entire 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 will deal, arrange and decide the destiny that the program flow must to take.

In other way 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.

2 How it does work

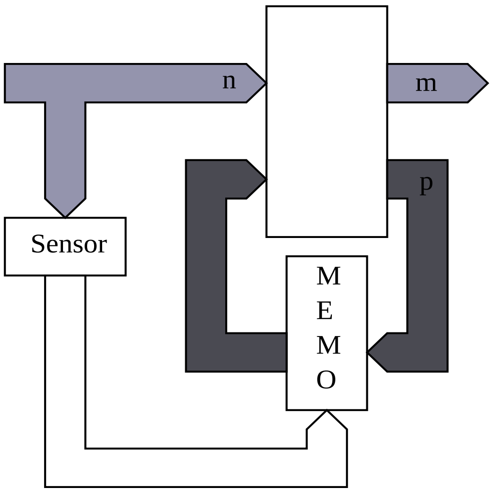
Following the theories developed in the thesis published in the book "Esção (n m p) - A Non Von Neumann Computer", from the teacher W. W. Martins[**?**], the **Esção** is a sequential machine that use a lot of devices from the family **ROM** (read only memory), who can works in asynchronous mode, without 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 $X\oslash$, who is input undependable variable.

It has then two kinds of 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

[*]Correspondence should be addressed to this autor at: Pontificia Universidade Catolica do Rio Grande do Sul, Mechanical and Mecatronics Engineering Dept. Av. Ipiranga, 6681. Predio 30/Bloco 7. CEP 90619-900. Porto Alegre-RS. Brazil. Phone: 55-51-3320-3584. Fax: 55-51-3320-3625.

the programs the suitable outputs to the aleatoric occurrences of the inputs.



- n number of **independents** 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-probabilistic 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.

2.1 Understanding Esção (n-m-p)

To really understand the functioning of a **Esção** is important to know two 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[?], 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** Boolean variables (input)
- m number of **dependable** Boolean variables (output)
- p number of **internal** Boolean variables (feedback or memories)

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.

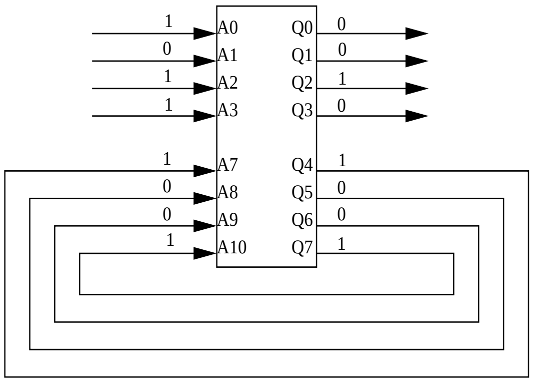
$$2^p = EIE$$

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 for a Digital System"* using the VLSI technology.

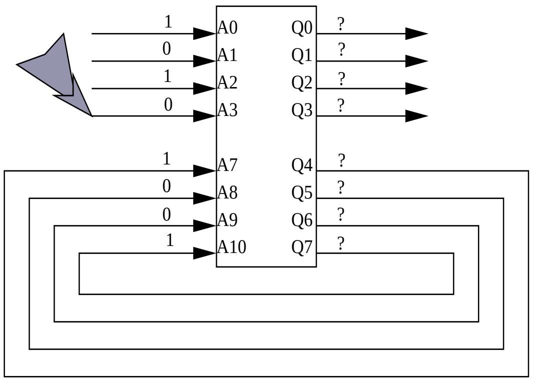
The amount of **EIE** will define the space to store the sequential programs that the machine will execute. So, for a **Esção** machine, as a **EIE** is simply a memory space, or else, a BYTE. As each byte uses one address, and to access must be gived the right address in the inputs of memory device.

Remembering about input variables, it always will be in the EPROMs 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!

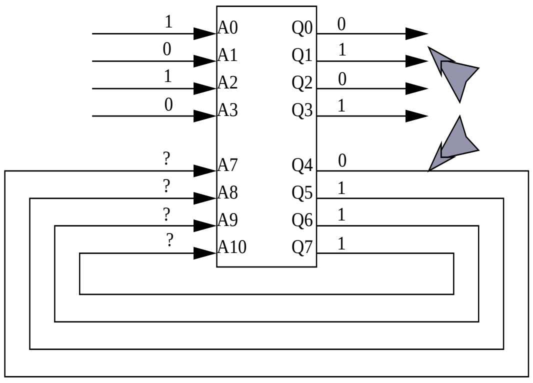
1. The EPROM found itself in a Stable Internal State (**EIE**), a constant input, and the feedback is identical as in the input as in the output. This state can be called of standby, because it is waiting a new order.



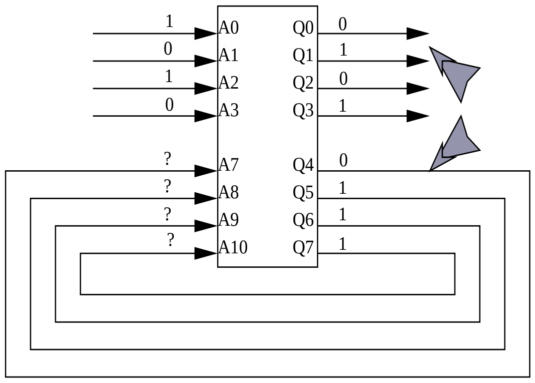
2. Happens a variation in the inputs.



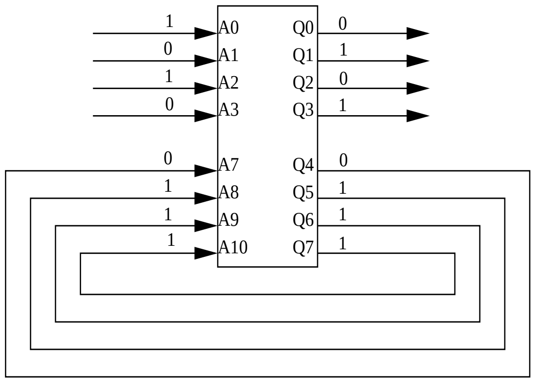
3. The EPROM gets it as a new address and immediately sends a new content in the output.



4. 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.



5. 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.



6. 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 **DDD** is made uniquely **EIE**.

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.

2.2 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.

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.

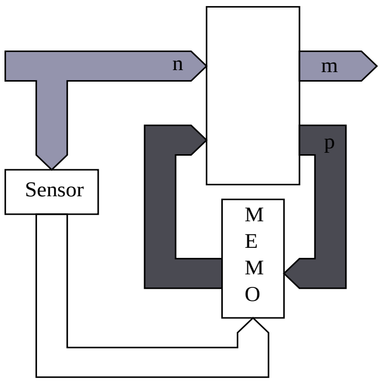
3. Drawing of the correspondent GRAPH with or without the temporal step (variable $X\oslash$), depending on the application.

4. The $X\oslash$ 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. Its 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, well 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 lets 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 memorys 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.

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

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.

References

[1] Wagner Waneck Martins. *Esção (n m p) - Um Computador Não-Von Neumann.* Editora Cartograf, 1st edition, 1985.

[2] Montgomery Phister. *Logical Design of Digital Computers.* John Wiley & Sons, Inc., 1958.