PAM – Paraconsistent Analyser Module

João Inácio da Silva Filho
Jair Minoro Abe

Faculdade SENAC de Ciências Exatas e Tecnologia
Rua Galvão Bueno, 430
01506-000 São Paulo - SP- Brazil

Institute For Advanced Studies - University of São Paulo
Av. Prof. Luciano Gualberto, trav. J, 374, térreo,
05508-900 São Paulo - SP - Brazil

Abstract

This work is a sequel to some previous results obtained by us concerning implementation of paraconsistent electronic circuits. We investigate a projected circuit called *Paraconsistent Analyzer Module* (*PAM*) based on a kind of paraconsistent annotated logics. It was implemented by using *CMOS* transistors and its major function is to detect inconsistent (contradictory) signals and give a non-trivial treatment to these situations, provided by a multivalued and paraconsistent logic that underlies these studies.

Key Words: Paraconsistency and electronic circuits, paraconsistent logical circuits, paraconsistency and circuits.

1 Introduction

This work is a sequel to some previous results obtained by us concerning implementation of paraconsistent electronic circuits (v.g., [Abe & Da Silva Filho 96], [Da Silva Filho & Abe 00], [Da Silva Filho 97]).

We investigate a projected circuit called *Paraconsistent Analyzer Module* (*PAM*) based on a kind of paraconsistent annotated logics. It was implemented by using *CMOS* transistors and its major function is to detect inconsistent (contradictory) signals and give a non-trivial treatment to these situations, provided by a multivalued and paraconsistent logic that underlies these studies.

2 Implementing *PAM* circuits

A Paraconsistent Analyzer Module is composed by literal operations f_i (i = 1, ... , 6) and paraconsistent gates AND and OR. The following diagram shows the *PAM* in blocks.

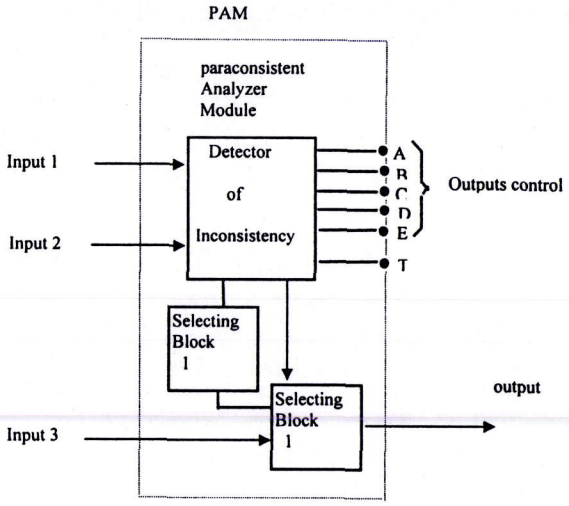

Figure 1 – Paraconsistent Analyser Module - *PAM*

The inconsistency detecting circuit is built with blocks consisting of literal operators and some usual logical gates which make the sensoring of the first two multivalued inputs. The circuits called selection 1 and selection 2 are composed by the paraconsistent primitive blocks OR and AND. These Paraconsistent analyzer modules linked in a convenient way allow a study of the signals, close to the studies of paraconsistent annotated logics $P\tau$. The diagram in blocks is shown in the following figure.

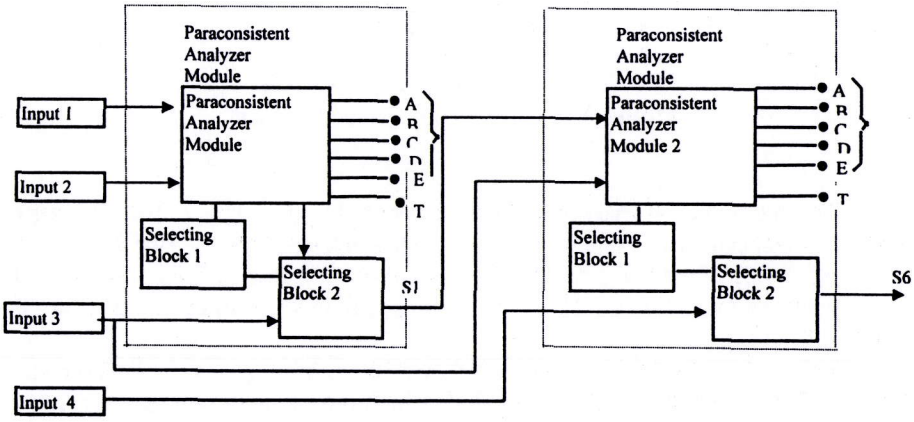

Figure 2 - A connection of Paraconsistent Analyzer Module.

Several *PAM* circuits can be interconnected allowing to work with contradictory signals in a nontrivial manner in its structure. These studies are inspired on a class of paraconsistent annotated logics (see, for instance, [Abe 92]).

In the above figure, the *PAM* 1 verifies the occurrence of conflicting signals between the first two sensors, choosing the logical signal which will appear in the output S_1. If inconsistent, the selection block 2 directs the signal of the sensor 3 to the output S_1.

If there is no inconsistency, the selecting block 1, directs the valued signal, following rules of the paraconsistent annotated logic $P\tau$, to the output S_1. Subsequently, the *PAM*2 detects if there is inconsistency between the resulting signal of the first analysis S_1 and the signal of the sensor 3. If there is inconsistency, the selection block 2 makes the output signal S_6 be the logical signal of the sensor 4. All logical signals presented in outputs are multivalued and the resulting signal has the logical signal according to the definitions of paraconsistent annotated logic.

The circuit which detects inconsistencies is built based on blocks consisting of literal operators and some of the usual logical gates. They perform the sensoring between the first two multi-valued inputs. The following figure shows how to detect inconsistencies by using literal operations:

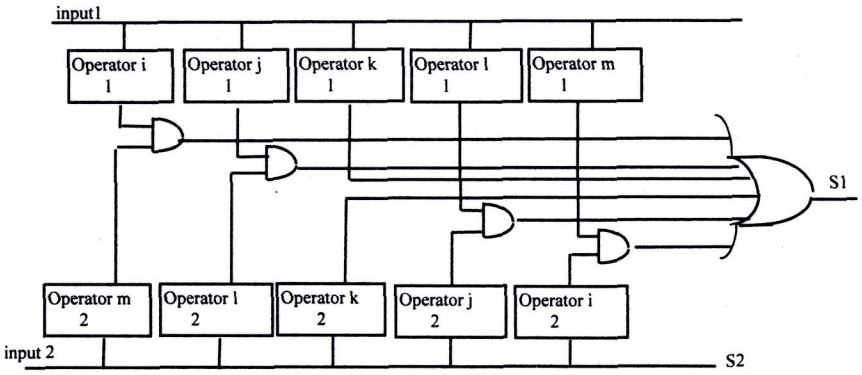

Figure 3 - Inconsistency detecting block

When the output of two complementary operators has logical level "1", we have an state of inconsistency. When an inconsistency appears among any signals of the inputs 1 and 2, one of the gates AND will have output "1". It is sufficient that one of the outputs of the gates AND be "1" in order to appear in S_1 (output of the gate OR) a logical signal "1".

The literal operators *k* are connected directly with the gate OR. This means that when one of the inputs presents the signal ½, which means inconsistency, the output S_1 has the signal "1". The occurrence of this combination of signals, in the inputs 1 and 2, results the action of the block of selection S_1 through of the output S_1, allowing the 3rd input to appear in the output of *PAM*. If the inputs 1 and 2 are the same logical signals,

then there is no contradiction and so the output S_1 has the signal "0". This value "0" applied to block selection 1 allows the signal S_2 to be selected as the output of *PAM*.

The block selection 2 will perform in "intermediate" situations, i.e., when there are not inconsistencies and also when the input signals are not equal. Such situations can be divided as follows.

Group 1.

This encompasses all situations where the input combinations of *PAM* have as output signal "1" or "0". This group does not offer difficulties. The possible combinations are

Input 1	Input 2	Resulting signal	Logical level
0	¼	False	0
¼	0	False	0
1	¾	True	1
¾	1	True	1

Group 2.

This encompasses all situations where the input combinations of *PAM* have as output signal "¼" (less false) or "¾" (less true). In these cases there are doubt to be considered. So, the circuit need find other evidences in order to improve the belief degree, making new consults. The combinations are the following.

Input 1	Input 2	Resulting signal	Logical level
¾	0	Less False	¼
0	¾	Less False	¼
1	¼	Less True	¾
¼	1	Less True	¾

The detecting circuit for the signals of group 1 is presented as follows.

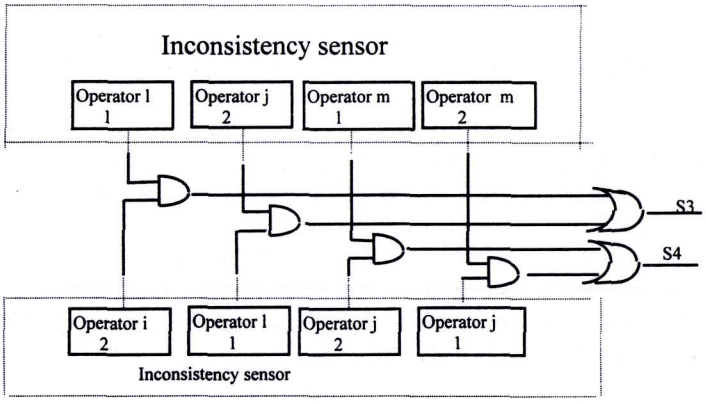

Figure 4 - Detecting circuit for the signals of group 1

The detecting circuit for the signals of group 2 is presented as follows.

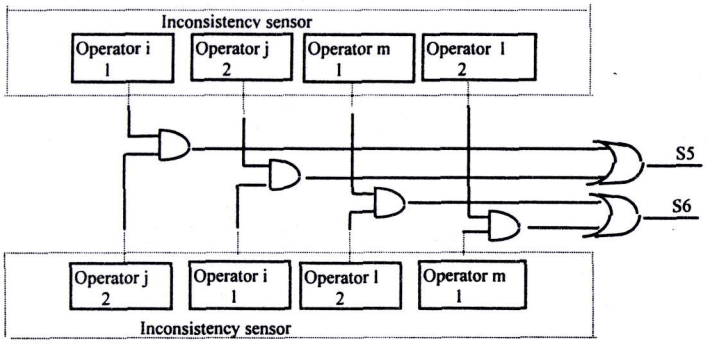

Figure 4 - Detecting circuit for the signals of group 1 - Equality detecting circuit

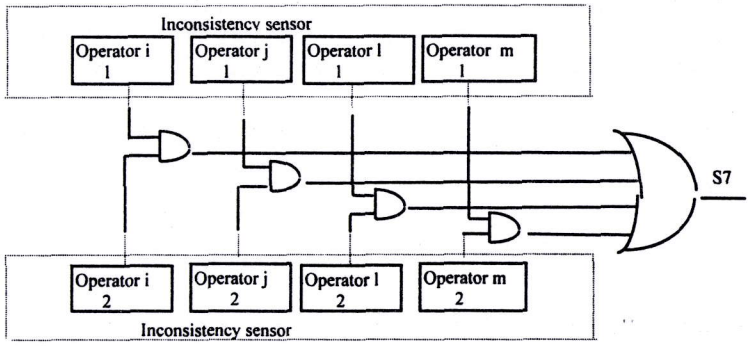

Figure 6 - Detecting circuit of the signals of group 2

The circuit of the Figure 8 below selects as output signal the input 3, when a logical signal "1" appears in S_1. In these conditions there is an inconsistency between inputs 1 and 2. A logical signal "0" in S_1 allows that the signal S_9 to be selected as output (S_{10}) of *PAM*.

In the presence of inconsistency, S_1 is 1, so the gate AND 1 has as output the logical level "0" and in the gate AND 2, the output signal are the values of the input 3. With this, the gate OR will have as input with logical level "0" and therefore it will allow the input signal 3 appears in the output S_{10}.

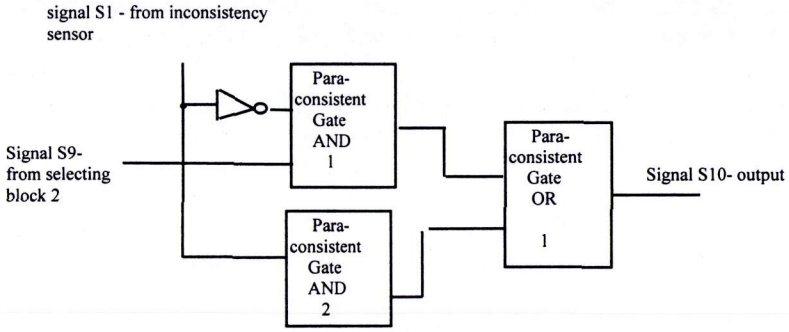

Figure 6 - Selecting block 1.

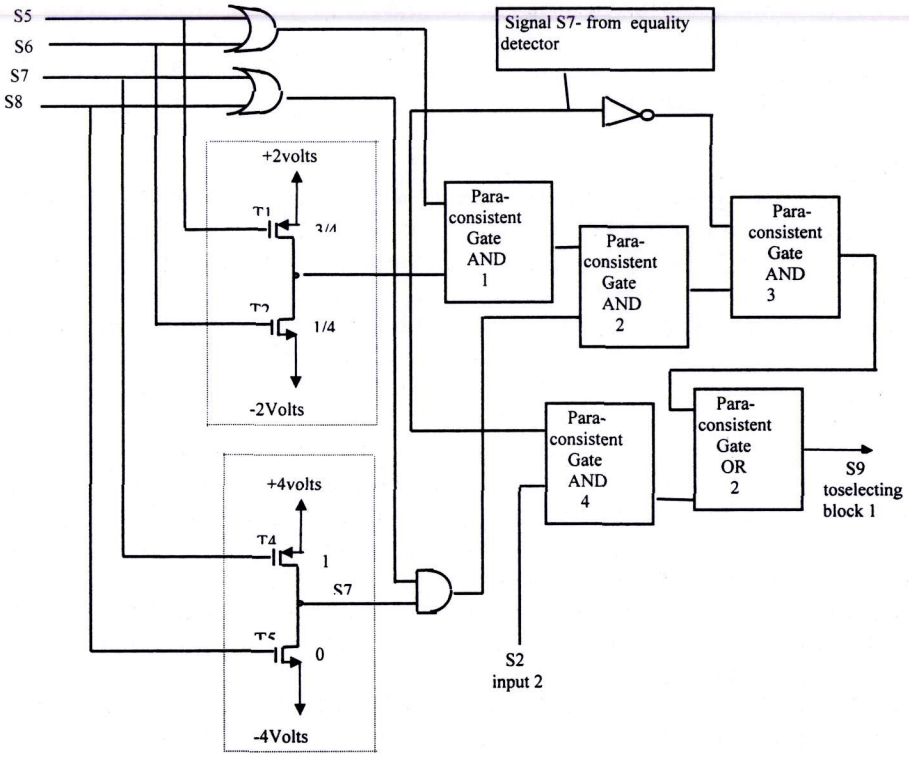

Figure 8 - Selecting block 2.

The selecting block 2 treats three signals. The first one is when the inputs 1 and 2 presents equal signals. In this case, S_7 is "1" and the gates AND 3 and 4, together with gate OR, releases the signal S_2 of input 2 to output S_9. The 2nd one is when the inputs are signals of the group 1. We have the input S_5 or S_6 equal to "1". This logical signal

"1" will polarize the transistors T_1 or T_2 and apply ±2 Volts in the input of the gate AND 1. As the other input of this gate is "1", this signal, ±2 Volts, equivalent to logical values ¼ or ¾, is applied in one of inputs of the gate 2. The outputs S_5 and S_6 of the group 2 have logical value "0". This guarantees that the output of the gate AND is "0". This value, applied to input of the gate AND 2, releases the logical signal coming from group 1 to output S_9.

The last one is the selection of the signal ±4 Volts, equivalently to logical signals "0" or "1". The inputs S_7 and S_8 with logical value "1", operate the transistors T_4 or T_5.

Conclusions

In this work we have presented a Paraconsistent Analyzer Module which allows work in the presence of contradictory signals. As far as we know, these treatments seem to be pioneering in the theory of electronic circuits. They matearize how to deal with the concept of contradiction in Hardware. It is perfectly possible to build paraconsistent digital systems, with an immediate application in several fields in Artificial Intelligence, Robotics, and Automation.

References

[Abe 92] ABE, J.M., *Fundamentos da Lógica Anotada* (Foundations of Annotated Logics) (in Portuguese), Ph.D. Thesis, University of São Paulo, São Paulo, 1992.

[Abe 97] ABE, J.M., Some Aspects of Paraconsistent Systems and Applications, *Logique et Analyse*, 157(1997), 83-96.

[Abe & Da Silva Filho 96] ABE, J.M. & J.I. DA SILVA FILHO, Implementação de Circuitos eletrônicos de funções lógicas paraconsistentes Radix N (in Portuguese), *Coleção Documentos*, *Série Lógica e Teoria da Ciência,* Institute For Advanced Studies, University of São Paulo, n° 22, 37p., 1996.

[Abe & Da Silva Filho 98] ABE, J.M. & J.I. DA SILVA FILHO, Inconsistency and Electronic Circuits, *Proceedings of The International ICSC Symposium on Engineering of Intelligent Systems* (EIS'98), Volume 3, Artificial Intelligence, Editor: E. Alpaydin, ICSC Academic Press International Computer Science Conventions Canada/Switzerland, ISBN 3-906454-12-6, 191-197, 1998.

[Da Costa, Abe & Subrahamanian 91] DA COSTA, N.C.A., J.M. ABE & V.S. SUBRAHMANIAN, Remarks on annotated logic, *Zeitschrift f. math. Logik und Grundlagen d. Math.* 37, pp 561-570, 1991.

[Da Silva Filho 97] DA SILVA FILHO, J.I., "Implementação de circuitos lógicos fundamentados em uma classe de Lógicas Paraconsistentes Anotadas", Dissertação de Mestrado-EPUSP, São Paulo, 1997.

[Da Silva Filho & Abe 99] DA SILVA FILHO, J.I. & J.M. ABE, PEC-Paraconsistent Electronic Circuits, to appear.

[Reiter 80] REITER, R. "A logic for Default Reasoning" *Artificial Intelligence* vol.13, 81-132, 1980.

[Rine 84] RINE, D.C & Z.G. VRANESIC & K.C. SMITH, "Electronic Circuits for Multi-valued Digital Systems", *New York: North-Holland*,1984.

[Weste & Eshraghian 85] WESTE, N.H.E. & K. ESHRAGHIAN, "Principles of CMOS VLSI VLSI Design" *Wesley Ed. Addison Publishing Company*, USA , 1985.