A NEW LOGIC PROGRAMMABLE AUDIO SWITCHING MATRIX

Giuseppe Magdalena Stephan[1], Fábio da Silva Dutra[2], Carlos Eduardo F. Savioli[1,2], , Madjer de Andrade Martins[1]
{ 26@ cetm.marinha.br, fabiodut@ lpc.ufrj.br, 264@ cetm.marinha.br, madjeram@ bol.com.br }

[1] Centro de Eletrônica da Marinha
Divisão Técnica
Praça Barão de Ladário S/N, Cep 20091-000
Rio de Janeiro / RJ – BRASIL

[2] Universidade Federal Do Rio De Janeiro
LPC – CT/UFRJ/PEE-COPPE – Bloco H s. 210
CP 68564, Cep 21945-970
Rio de Janeiro / RJ – BRASIL

Abstract – This article describes a logic programmable audio switching matrix developed by this Center to substitute the old mechanical ones in some of the Brazilian Navy ships. With the purpose to switch audio signals only, this new system allows a fast, friendly and easy switching control by an IBM-PC compatible computer and a specific FPGA-based hardware.

I. INTRODUCTION

Some of the Brazilian Navy ships still use arrays of mechanical hand-commanded switches, which are used to connect each endpoint of communication to a different transmitter or receiver, as shown in Figure 1. An endpoint of communication is any device, located anywhere inside the ship, able to transmit and receive signals in the audio spectrum of frequencies, like TTY (Teletype), Morse devices and so on.

This kind of equipment is mostly used during real exercise operations in the sea, when the enemy team (foe) tries to listen to our radio communications. The fast switching among the various transmitters and receivers is a way to turn this listening into a difficult task, because those switching techniques not only affect the carrier frequency, but also the type of the signal modulation, like SSB or DSB.

The changes must be done as fast as possible and simultaneously among all those communications devices. In order to improve the performance of operation performance, seriously affected by the slow nature of the human mechanical switching, a new approach was proposed, in which the current mechanical system is substituted by a computer and FPGA-based one.

II. PHYSICAL STRUCTURE

Since there is a complex wiring system already assembled in the Brazilian Navy ships, which would be hard to remake, the design was strongly oriented to conserve the previous mechanical structure.

The current structure is composed by a 4x10 matrix of identical mechanical modules, according to Figure 1. Each module has 10 switches and each switch has seven positions (off, receivers 1 to 5 and extension).

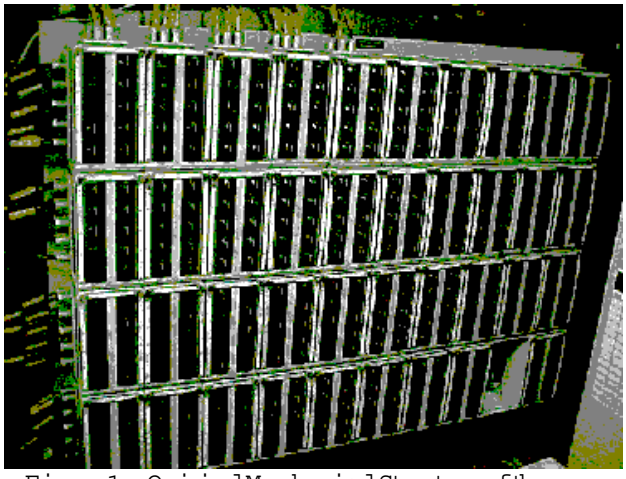

Figure 1 – Original Mechanical Structure of the Switching Matrix.

Therefore, the proposed solution was to customize a printed circuit board (PCB) to execute the function of a unique module. Care was taken in the design of the PCB to make it fit exactly in the mechanical console dimensions, avoiding the need for any external modifications.

The Figure 2 shows the electronic version of the system. Each module was substituted for a programmable logic-based board.

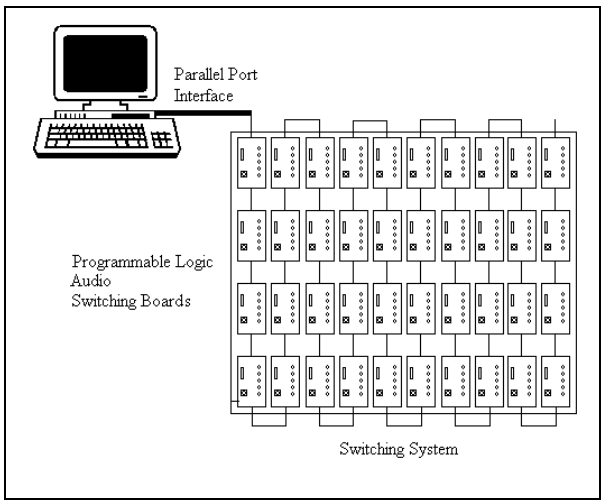

Figure 2 – Electronic Version of the Switching Matrix.

III. DEVELOPMENT

The new system was implemented with a mixed signal circuit, whose block diagram is shown in the Figure 3. This block diagram represents only a board.

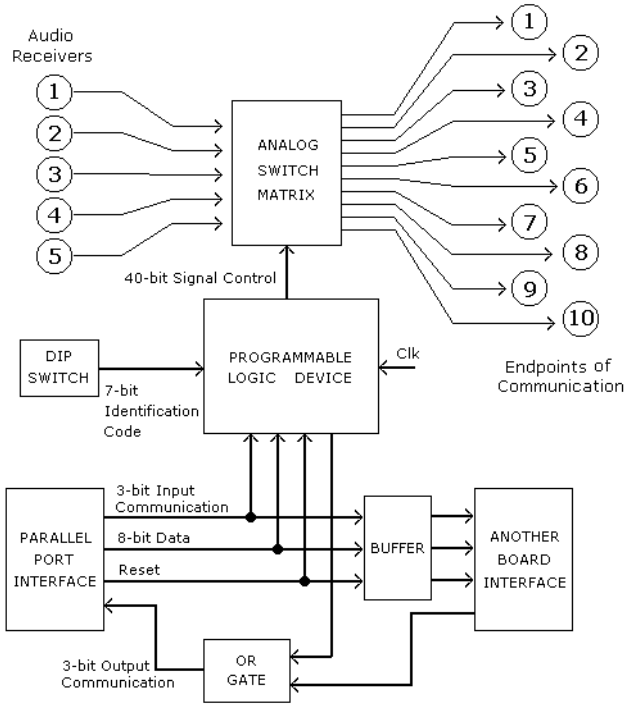

Figure 3 – Block Diagram of the Board.

The function of each block is explained in the further items:

A. AUDIO SWITCHING

Analog multiplex devices, 4052 [5], were used to do the audio switching. These devices cause negligible interference in the audio signal and are controllable by digital signals. The balanced specifications of the 4052 are critical for this application.

The programmable logic device allows a remote control of the switching matrix by the computer software.

B. IDENTIFICATION CODE

The switching matrix of the ship is composed by 40 identical boards, which must be correctly identified. In order to avoid possible errors in the switch selection operation, an identification code was implemented

The identification code is a 7-bit number, and so it is able to identify up to 128 different boards, more than necessary.

This code could be programmed in the logic device, but this procedure would make the device programming and system maintenance more complex. Then the proposed solution was to add a dip-switch in each board assigning a unique code for it.

The dip-switches must receive different values permitting to use the same VHDL programming logic in all devices.

C. COMMUNICATION PROTOCOL

A communication protocol among the programmable device and parallel port interface was implemented to allow the control of switching.

The signals used to implement the protocol were:

- SB (Select Board) indicates that the 8-bit data in parallel port is ready and represents the identification code of board;
- ACK_B (Acknowledgment Board) indicates the acceptance the SB signal;
- SSP (Select Switch and Position) indicates that the 8-bit data in the parallel port is ready and represents the selected switch and its position;
- ACK_SP (Acknowledgment Switch and Position) indicates that the board accepted the SSP signal;
- YACK (Yes Acknowledgment) indicates that the software accepted both ACK_B and ACK_SP signals;
- ERROR indicates that there was a parity error in the communication.

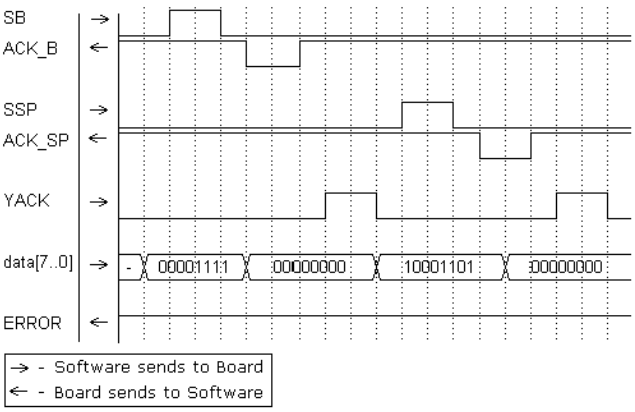

Figure 4 – Communication Protocol Handshaking.

To start the handshake, the information must be steady in the parallel port, and the software must send a pulsed SB signal.

The board whose identification code is coincident to the data in parallel port will accept the communication and then send the ACK_B signal to the computer. Note that ACK_B is negative logic.

The computer will receive the ACK_B and, to avoid a possible deadlock, it will send YACK signal to finish the communication.

After this identification procedure, the next step is to select one of the 10 switches and its position.

The information must be steady in the parallel port again, and the software must send the SSP signal.

The same board will accept the communication, and then send the ACK_SP signal to the computer. The selected switch will change to the new position. Note that ACK_SP is also in negative logic.

The computer will receive the ACK_SP, and then will send YACK signal to finish the communication. The handshake will be completed.

D. ERROR DETECTION

In order to avoid a possible error in the communication process, a bit of parity was added in the 8-bit data communication.

If any bit of that byte changes during the transmission, the board is able to detect it and send the negative logic ERROR signal, Figure 5.

After parity error detection, the protocol must reboot due to the board return to stand-by mode.

The time diagram of Figure 5 simulates a parity error. When the SB pulse is sent, the 8-bit data on parallel port is 10001111_b. This data has odd parity, so the logic detects the problem and sends a pulse of ERROR signal.

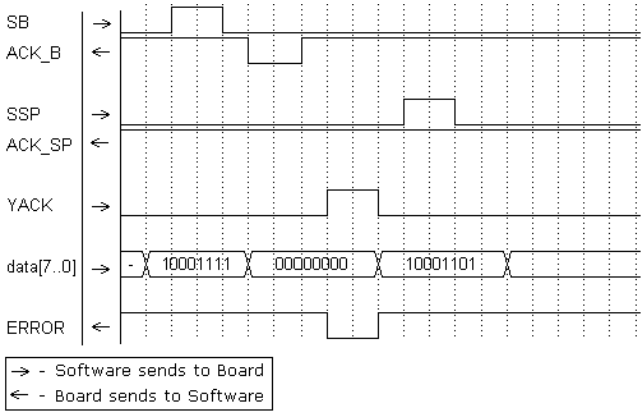

Figure 5 – Parity Error Detection.

E. DATA INFORMATION

Two 8-bit data are transmitted in each communication. The first one represents the identification code, and its structure is shown in Figure 6.

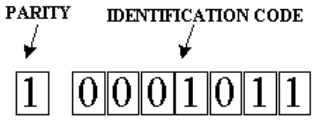

Figure 6 – First 8-bit Data Structure.

The second information represents the target switch and its desired position. Its data structure is shown in Figure 7.

Three bits are reserved to select up to eight positions, and four bits allows selecting up to 16 switches.

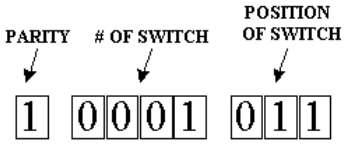

Figure 7 – Second 8-bit Data Structure

F. INTERFACE WITH NEIGHBOURS BOARDS

Each board has a buffer to drive the board it belongs. Therefore, the 3-bit input communication protocol (SB, SSP and YACK), 8-bit data and Reset signals are propagated from a board to its neighbors without loss of integrity.

The purpose of the OR gate shown in figure 3 was to process the 3-bit output communication protocol (ACK_B, ACK_SP and ERROR), dealing with the limitations due to the use of only one parallel port.

G. VHDL PROGRAMMING

A programmable logic device, MAX7000, from Altera family [4], was used to help the computer software to control the switching remotely.

The architecture developed was divided in two blocks: DATA PATH and CONTROLLER, as shown in Figure 8.

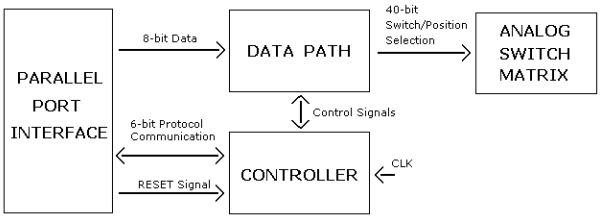

Figure 8 – Logic Block Diagram.

Both blocks were programmed in VHDL.

A finite-state machine [6] was created to implement the CONTROLLER. The DATA PATH was divided in four blocks, LATCH, PARITY, DECODE and COMPARATOR.

A detailed block diagram of the programmed logic is shown in Figure 9.

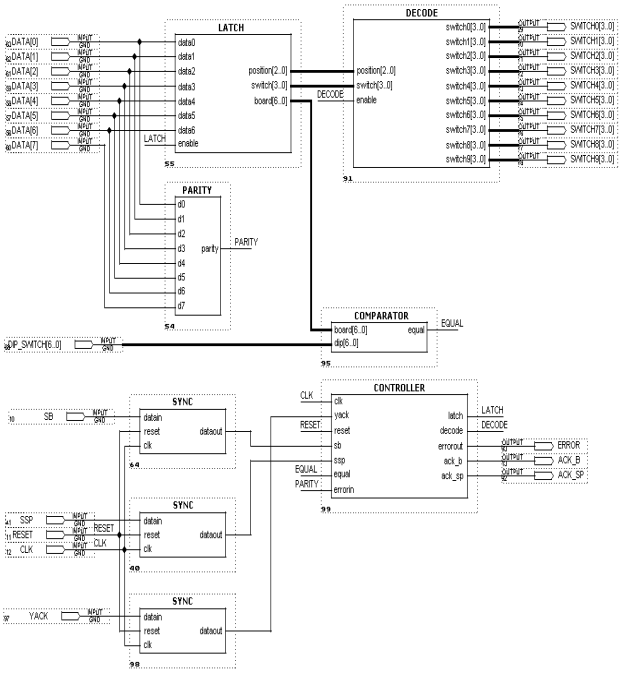

Figure 9 – Detailed Architecture of the Programmed Logic.

The function of each block is described below:

- LATCH reads the 8-bit data and divide it according to the data structures of Figures 6 and 7;
- PARITY verifies the parity of 8-bit data present on parallel port interface;
- COMPARATOR compares the value of the dip switch code and the code sent by the parallel port to identify the correct board;
- SYNC synchronizes the external control signal;
- DECODE receives the 8-bit data code and decodes it to select the desired switch;
- CONTROLLER controls the communication and switch selection.

In addition, the function of the each signal is described in the following items:

- Reset allows to reinitialize the programmable logic;
- DIP_SWITCH[6..0] is a 6-bit bus that defines the identification code;
- DATA[7..0] represents the 8-bit data on parallel port;
- SWITCH[3..0] are the 4-bit bus that controls the switch switching;
- PARITY internal signal, indicates a parity error;
- LATCH and DECODE internal signals, enable the LATCH and DECODE blocks, respectively;
- EQUAL internal signal indicates that the 8-bit data and dip-switches are identical.

The other signals are used in communication protocol whose description is explained in section III.c.

H. SOFTWARE

The platform used to this first prototype was Intel-Based running Windows 98 operating system. The software is Object-Oriented and was developed in Borland Delphi 5.0. Special care was taken in its graphic interface, which must be friendly and let fast human operation.

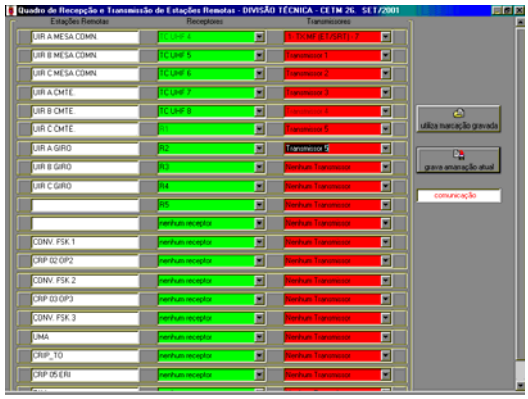

Figure 10 – Graphics User Interface.

For the selection of the transmitters and receivers, a combination of comboboxes, a kind of an object in the objected-oriented environment, were used, as shown in Figure 10. Another important feature of the system is possibility to program specific configurations among endpoints and receivers, to use them whenever necessary.

IV. RESULTS

The results of simulation are described in this section. A prototype of the system was implemented and an auxiliary circuitry was developed to simulate the system on the ship

The implementation of the system was divided in the following tasks:

A. DIGITAL SIMULATION

Each digital block of block diagram in Figure 9 was simulated, using the Altera's MAXplus II tool, to verify its working. The final simulation results are shown in Figure 11.

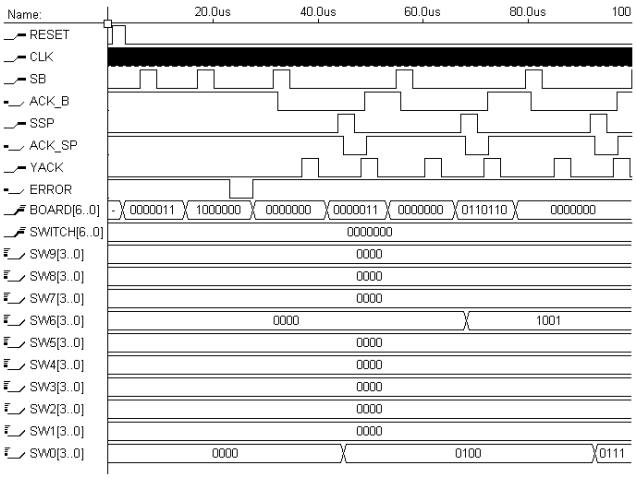

Figure 11 – Simulation Results.

Observing the Figure 11, the identification code is 00000000_b (SWITCH[6..0]). The communication begins when the first pulse of SB signal is sent. However, the BOARD[6..0] Bus (00000011_b) does not have the same value of SWITCH[6..0], what means the software wants to communicate with another board, whose identification code is 00000011_b. Therefore, the programmable logic sets aside the first pulse of the SB signal.

With the same code in SWITCH and BOARD bus, the logic selects the desired switch.

B. LABORATORY TESTS

To help the tests of the assembled system, it was developed a circuit tester based on the microcontroller PIC 16F84, from Microchip.

The tester creates 10 different set of pulses, generated by the microcontroller, which simulate the audio signals. In the panel of the tester, a 10 leds bar blinks this pulses. The first pulse blink once, the second twice, the third three times and so on until the tenth, that pulses ten times.

As show in figure 12, each pulse turns its remote audio unit led on.

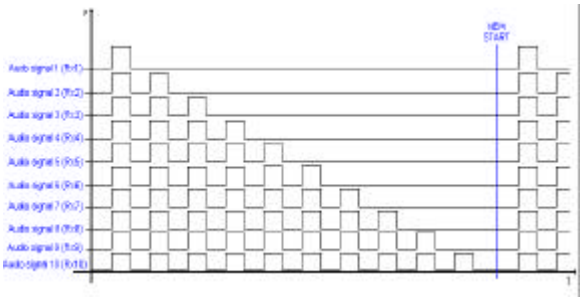

Figure 12 – Microcontroller's Pulses.

In the panel there are three bar leds, one denote the audio signals and the two other, the signal switched by two modules. Each module is connected to the tester. For instance, when channel one is switched to the third receiver, the led one of the bar leds will blink three times. The first five leds of audio signal bar leds are connected at a module and the last five leds, connected to the other module.

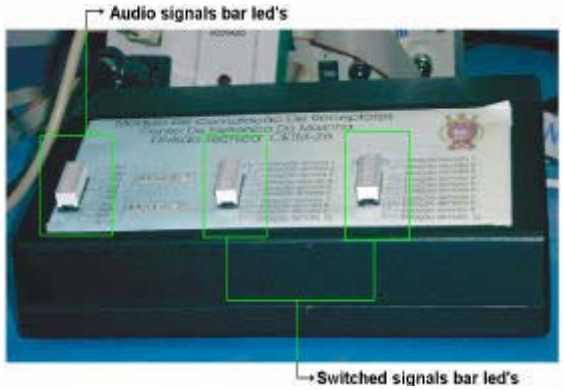

Figure 13 – Tester Front Panel

V. CONCLUSION

This new paradigm introduced with that prototype improves the speed of switching, if compared to the previous mechanical human switching, and turns that kind of operation more confident, since the operator selects exactly the desired transmitter or receiver with a few mouse clicks on the screen.

Besides bringing some technology independence to the country, updating the matrix, taking advantage of the basic previous mechanic structure and all its related wiring, represents a great effort in saving the financial navy resources, without lack of performance, if compared to some commercial switching matrices available in the worldwide weapons industry market.

This prototype, however, should be undoubtedly improved to comply the military standards requirements for ship equipment. The major changes would be in the software communication protocol, in order to reduce the probability of deadlocks, and in the software itself, which must be executed in a real-time platform, more robust than most of the commercial operating systems.

In hardware, minor changes should be done to improve the performance of the prototype, such as inclusion of a power-on reset circuitry and minimization of the physical dimensions and energy consumption.

Acknowledgments

The authors would like to thank the Vice-Adm. Pierantoni, Vice-Adm. Loesch and Capt. Gusmão, for their financial support to this project.

References

[1] CPDET "Manual Técnico do Módulo de Comutação de Receptores ET/URA-15", IMBEL, 1986.

[2] CPDET "Manual Técnico do Módulo de Comutação de Transmissores ET/URA-16", IMBEL, 1986.

[3] G. M. Stephan, M. C. Kireff, "Projeto e Montagem de Circuito de Comutação de Sinais de Áudio", Pesquisa Naval, nr.14, pp. 377-384, October 2001.

[4] ALTERA "Data Book", Altera Corporation, 1998

[5] RCA Solid State, "CMOS Integrated Circuits Data Book", RCA Corporation, 1983

[6] Daniel D. Gajski, "Principles of Digital Design", Prentice Hall, pp. 318-385, 1997;

[7] Peter J. Ashenden, "The VHDL Cookbook", first edition, pp. 1.1 – 2.1, July, 1990;

[8] K. Skahill, "VHDL for Programmable Logic", Addison-Weskley, pp. 1-17, 1996;

[9] Peter J. Ashenden, "The Designer's Guide to VHDL", Morgan Kaufmann Publishers, pp. 2 – 25, 1996.