The automation of the car access system using logical circuits FPGA

GABRIEL PREDUSCA, FLORIAN ION, EMIL DIACONU, CATALIN DUSCU
Department of Electronics, Telecommunications and Energy Engineering
Valahia University of Targoviste
Bd.Carol I, Nr.2,130024
ROMANIA
gpredusca@gmail.com, ionflorian@yahoo.com, emy_diaconu@yahoo.com, catalinduscu@yahoo.com

Abstract: - Parking is an increasingly difficult problem for many drivers in crowded cities with limited parking spaces. Parking space management plays a decisive role in the urban transport planning process. This article proposes to complete the automation of a parking access system using programmable logic circuits - FPGA for the integrated management system. The proposed system using FPGA is easy to implement and it efficiently monitors all functions by using a single integrated circuit.

Key-Words: - circuit FPGA, ISE WebPACK, automated parking system, board Nexys2.

1 Introduction

Problems caused by lack of parking spaces are becoming increasingly acute in many cities in Romania and there is no proper management of these parking areas. This confirms the importance of promoting urban parking management and control policies that are compatible with current regulations regarding the transportation planning process.

In Romania, in 1990, the fleet was 1.9 million units, namely a car for 12 people. In 2010, the fleet reached 5.4 million units which is, a car for 4 people [1]. Referring to the above, we can say that the current infrastructure and parking facilities are insufficient to support the flow of vehicles.

Managing car parking spaces therefore plays a decisive role in the planning of land transport, and it can significantly contribute to reducing congestion and lead to the elimination of problems caused by illegal parking.

In the transport sector, the use of telematics is associated with the development of what are known as "Intelligent Transport Systems" (ITS), with very different applications. The main research topics covered in this subject are well documented in several investigations reported by Predusca [2], Chowdhury and Sadek [3], Ghosh [4], Sussman [5], Tsopelas (2000) [6], Yass and Yasin [7]. These resources are currently being implemented in various functional areas, and can also be applied to manage parking.

2 The design process with programmable logical circuits

The FPGA circuit contains a large number of programmable logic blocks - CLB (Configurable Logic Blocks), distributed in programmable wiring matrix, surrounded by programmable I/O blocks – Fig.1. With them, more complex logical functions can be created using electronic schemes and also written codes in a hardware descriptive language - VHDL, Verilog or ABEL [8]. The FPGA circuit was invented by Xilinx in 1984.

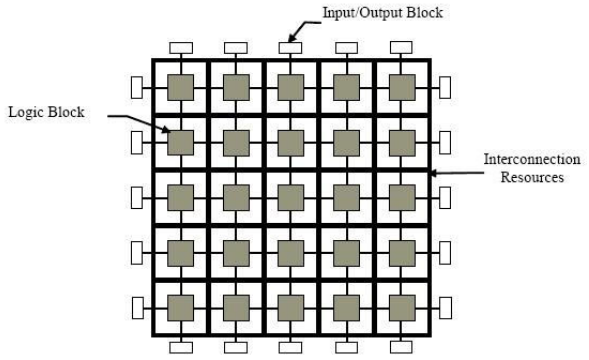

Fig.1. The structure of an FPGA

As the FPGA devices are used in advanced applications, they must support a growing number of standards of the signals that are connected to the I/O pins of the circuit [9].

In many cases, the facilities of the I/O block are automatically selected by the instruments used to implement the application. The user can specify their status by declaring certain components at the beginning of the VHDL code or schematic. The

Xilinx libraries contain the following templates that can be used to declare the components of the I/O circuits: IBUF (input buffer), IBUFG (clock input buffer), OBUF (output buffer), OBUFT (three state output buffer) IOBUF (input/output buffer).
The ISE WebPACK software package, of Xilinx, is an integrated environment for the design of digital systems using FPGA circuits.
For simulating the descriptions, the WebPACK software package contains a version of the ModelSim simulator of Model Technology, called MXE (ModelSim Xilinx Edition). The WebPACK package also contains a graphical interface for specifying test vectors. The WebPACK software package allows the use of FPGA Xilinx devices:
• FPGA Circuits from the Spartan family: Spartan-2, Spartan-2E, Spartan-3, Spartan-3E;
• FPGA Circuits from the Virtex family: Virtex-E, Virtex-2, Virtex-2 Pro.
Fig.2 shows the main modules of the system software design, Xilinx ISE WebPACK.

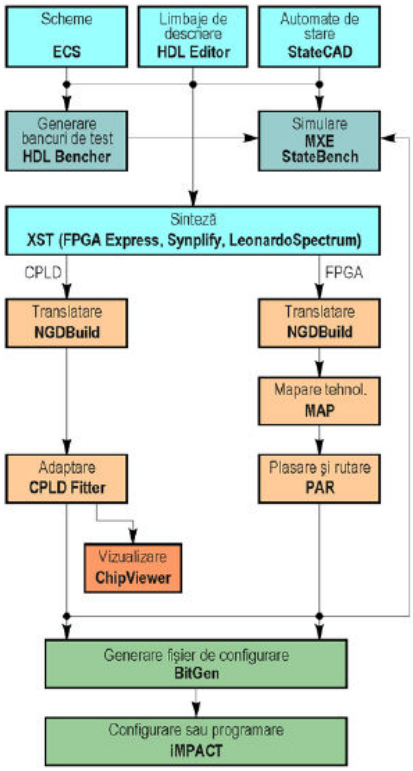

Fig.2. The structure of system Xilinx ISE WebPACK.

The modules for the description of the digital system include the schematic editor ECS (Engineering Capture System), the HDL editor for hardware description languages and the editor for StateCAD state diagrams. The schematic editor uses a module that contains the library primitives for FPGA circuits that can be used to implement the digital system. The HDL editor allows editing source files containing the description of the digital system in VHDL or Verilog language. The StateCAD editor allows the description of finite state machines using state diagrams. The user can draw the states machine, the transitions between states and can specify the output signals to be generated in each state.
The HDL Bencher module allows an easier testing of the digital systems. With this module, the user can enter input signals in a graphical form as timing diagrams.
These modules make the synthesis of the digital system description, generating an internal representation of the system as a list of connections. The results of synthesis can be controlled by specifying various properties. One of the properties that is usually specified indicates how to optimize the results of the synthesis, either in terms of the resources used to implement or of the operating speed.
The Synthesis module that is integrated within the WebPACK system is XST (Xilinx Synthesis Technology). In addition to this module, the following synthesis modules can be used, for which the system contains integrated interfaces: FPGA Express (Synopsys, Inc.) Synplify / Synplify Pro (Synplicity, Inc.) or LeonardoSpectrum (Copy Logic, Inc.).
The iMPACT module allows the configuration or programming of the FPGA circuit used for implementation. To do this, a parallel or a JTAG cable must be connected to the parallel port of the computer (in some cases, you can use a serial or a USB cable).
The iMPACT module contains a program (PROM File Formatter) to generate the required file to programme a PROM memory. Such a non-volatile memory can be used to preserve the configuration information required by the FPGA devices based on the SRAM technology that requires configuration after each interruption of the voltage supply. The file generated by the iMPACT can be a MCS-86 (Intel), EXORMACS (Motorola) or a TEKHEX (Tektronix). A **.hex** file can also be generated containing the hexadecimal representation of the string configuration.
Assign Package Pins is a new sub process, called PACE, with which you can view and enter the I/O pins of the FPGA circuit, where the implementation is made (Fig.3, Fig.4). Here, the program area constraints can be entered. Each type of pin is marked with a certain colour - as shown in Fig.5, or with a package symbol in the view mode.

ISBN: 978-1-61804-099-2

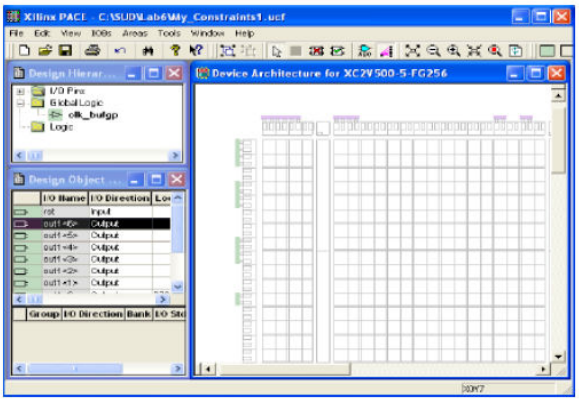

Fig.3. Xilinx PACE.

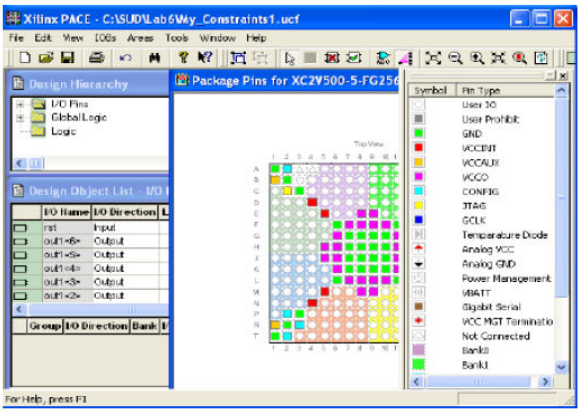

Fig.4. PACE based on colour coding.

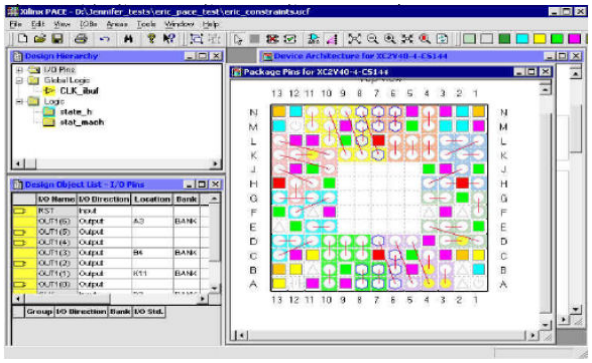

Fig.5. Assign pins I/O using PACE.

Ngdbuild is a process that identifies the basic module and its ports with the FPGA circuit pins. It also enables a sub-set voltage to be applied to the I/O port.

The Synthesize process is the following process invoked. The Synthesis of complex circuits (e.g. the synthesis of a circuit for implementing floating-point arithmetic operations) can take hours and it requires tens of GB free space on the hard drive. Running this process implies the existence of at least one of the following tools: Xilinx XST Compile, Synplicity or Leonardo Spectrum. Xilinx XST Compile is a tool and Xilinx proprietary test environment dedicated hardware circuits containing only FPGA/CPLD of Xilinx. The other two utilities also do the synthesis for other FPGA/CPLD, such as the ALTERA circuits. This process contains a total of four sub-processes. It should be noticed that all sub-processes are executed in a predetermined order. Running a specified sub process, involves executing in order all sub processes that precede the chosen sub process.

Analyze Power (XPower) - is a tool used to estimate the power consumption of an FPGA circuit. An example can be seen in Fig.6. This tool takes the FPGA implementation files of the circuit and also the input of the project to produce a very accurate estimation for each voltage supply. XPower can process the files obtained from the process simulation results, files in **.vcd**.

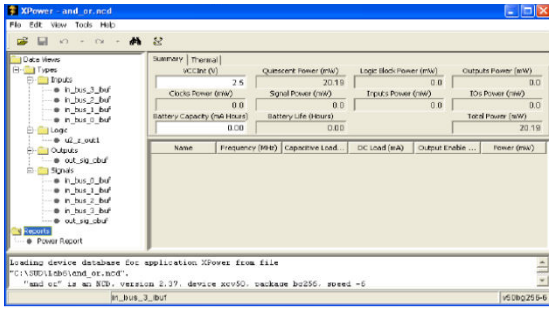

Fig.6. Results using XPower tool.

After running this process, you can select the RSFF Report option to view and analyze the obtained results. This report will contain information of the synthesis results obtained for each case tested. The symbol "*" is the one that indicates exactly what was saved, as shown in Fig.7.

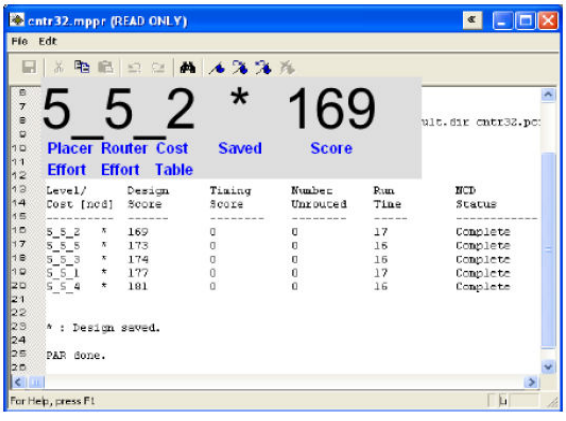

Fig.7. MFPR Report.

3 The system implementation

The obtained system allows monitoring:

- I/O by using an input respectively, output sensor;
- the initial number of seats;
- the number of places available.

ISBN: 978-1-61804-099-2

The access in the parking is made by raising a barrier if there are vacancies. If it is a paid parking, with the lifting of the barrier a ticket is issued with a client code and there starts a timer for measuring the time left in the parking. If the client was charged the exit is allowed by raising another barrier to leave the parking. The logic signals for the two barriers were named O1 – for the entry barrier and O2 - for the exit barrier. The analogue signals from the sensors are transferred through a digital analogue converter as input signals in the FPGA.
The implementation algorithm of the state machine is shown in Fig.8.

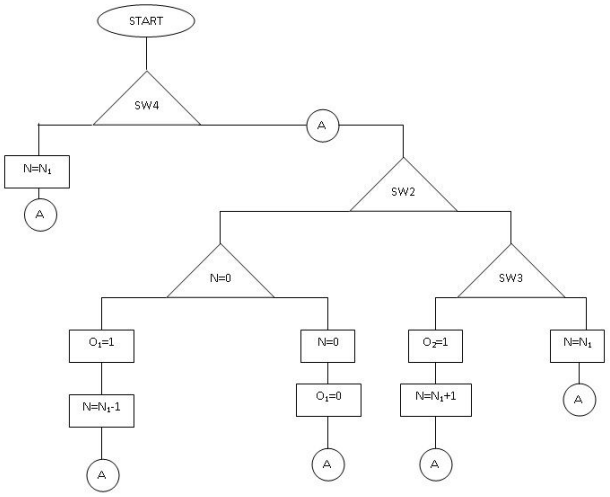

Fig.8. Machine algorithm of implementation

The variables used in the system are:
SW2 = input sensor - input request,
SW3 = output sensor - output request, charging for parking
SW4 = setting / operating
N = the number of places available,
N1 = the initial number of places available,
O1 = lifting of the entry barrier, release client code, start parking timer,
O2 = lifting exit barrier
A = original operating condition
If the SW4 switch is logic 0, a button is pressed to set the initial number of available places - N1, that will be displayed, then proceed to point A. If the SW4 switch is logic 1 and the SW2 switch is logic 0 and the N = 0 condition is not met, the „O1" LED turns on - which means raising the barrier to entry, the displayed number N decreases, then, pass to the point A, if the SW3 switch is logic 0 is, it passes to point A. If the N = 0 condition is fulfilled 0 is displayed, the barrier is not lifted to access and it passes to point A. If the SW4 switch is logic 1 and the SW2 switch is logic 0 and the SW3 switch is logic 1 the „O2" LED turns on - which means raising the exit barrier, the N number is incremented and it passes to point A. The transition table corresponding to the algorithm in Figure 8 is as follows:

Table 1. Table of transitions

SW2	SW3	N
0	0	$N=N_1$
0	1	$N=N_1+1$
1	0	$N=N_1-1$
1	1	$N=N_1$

The logical equations corresponding to the two outputs are:
O2 = SW3 * Paid (if parking is paid)
O1 = SW2 * N (the access barrier is lifted if there are available places).
To achieve the automatic system, the following material resources were used:
- NEXYS2 Simulation board with XILINX Spartan 3E chipset – Fig.9;
- Software for programming the FPGA circuit - Xilinx ISE WebPACK 8.1.i;
- IBM compatible PC.

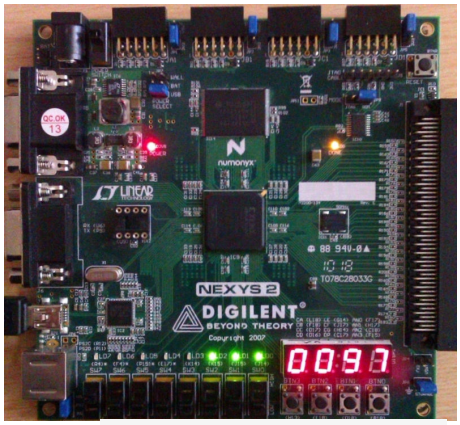

Fig.9. Board simulation NEXYS 2

The Spartan 3E chipset contains 500,000 logic gates and it has the following features – Fig.10, Fig.11, Fig.12:
- 1 USB port;
- 50MHz oscillator;
- 8 LEDs, 4-digit 7-segment display, 4 buttons, 8 switches;
- 16 MB flash memory;
- 16 MB SDRAM;
- PS/2 port, RS232 port;
- VGA interface.

ISBN: 978-1-61804-099-2

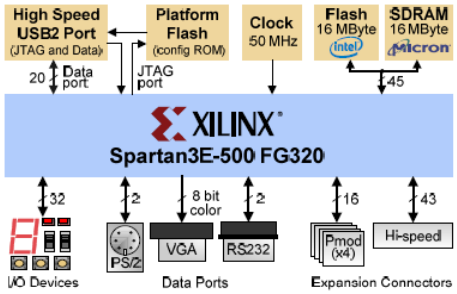

Fig.10. Block diagram of switches and buttons on the board NEXYS2

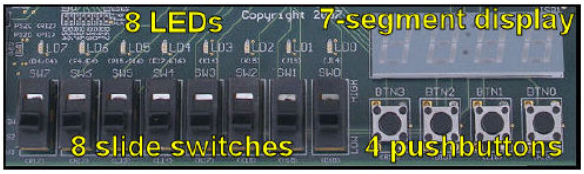

Fig.11. Inputs/Outputs NEXYS 2

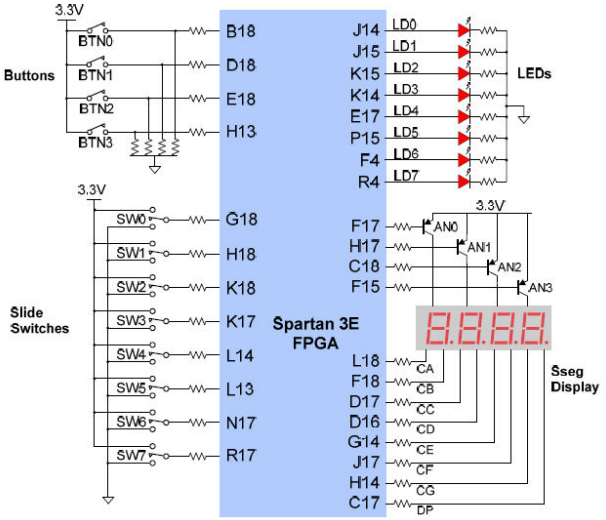

Fig.12. The pins corresponded I / O board NEXYS2.

3 Conclusions

The proposed and tested system is suitable for current lifestyles. The use of new technologies (FPGA circuits, simulation board Nexys 2) in parking management can significantly reduce parking problems. The automated parking systems are more efficient, have a very low maintenance cost, and future developments - the attachment of new facilities, are very easy to achieve.

References:

[1] http://m.hotnews.ro/stire/9861708.

[2] G. Predusca, Sisteme de comunicaţii în transporturi utilizând tehnologie GPS, Teza doctorat, Universitatea Politehnica Bucureşti, Facultatea de Automatică şi Calculatoare, domeniul Automatică, nr. 4871 / 07.08.2006.

[3] M.A. Chowdhury, A. Sadek, *Fundamentals of Intelligent Transportation Systems Planning (Artech House Its Library) (1 edition)*, Artech House Publishers, 2003.

[4] S. Ghosh, *Intelligent Transportation Systems: New Principles and Architectures,* CRC Press, 2000.

[5] J.S. Sussman, *Perspectives on Intelligent Transportation Systems (ITS)*, Publisher Springer, 2005.

[6] I. Tsopelas, Car Parching and Transport Telematics: Improved User Information, *In UTSG-32nd Annula Conference*, Vol.3, 5-7 January, University of Liverpool, 2000.

[7] Ayad Ahmed Yass, Norizan Mohd Yasin, Using the concept of management information system to improve an effusive intelligent parking system, *International Proceedings of Computer Science and Information Technology*, Vol.2, 2009, (2011) IACSIT Press, Singapore, pp.355-359.

[8] Ion F., *Electronică digitală – memorii ROM, RAM, circuite CPLD, FPGA*, Editura Bibliotheca, Târgovi□te, 2009.

[9] Brown S., Rose J., *Architecture of FPGAs and CPLDs: A tutorial*, Departament, of Electrical and Computer Engineering, University of Toronto, http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf.

ISBN: 978-1-61804-099-2