DE GRUYTER OPEN

DOI: 10.1515/SBEEF-2017-0019

MONITORING A SEMAPHORED CROSSROADS USING FPGA DE0

G. PREDUSCA, M. BOGDAN
Department of Electronics, Telecommunications and Energy Engineering, Valahia University of Targoviste
E-mail: gpredusca@gmail.com, marin_fynutu@yahoo.com

Abstract. ***Intelligent Transport System (ITS) is a worldwide movement meant to use advanced technology to make the ground transport systems more efficient, clogging-free, more secure and less polluting. The present technologies permit the development of applications for traffic monitoring in a semaphored crossroads. The solution proposed is a low-cost one, using FPGA DE0 development platform. The implementation and simulation of the application have been realized in Quartus II 13.1 and the diagrams corresponding to the outputs and inputs have been obtained using ModelSim-Altera 10.1d.***

Keywords: semaphored crossroads, FPGA, DE0, Quartus, ModelSim.

1. INTRODUCTION

Advanced Traffic Management Systems (ATMS) involve the use of sophisticated techniques to control traffic in the transport network. An important element of ATMS is the advanced traffic control system, which has in view all the semaphores in a certain zone creating a functionality such as a green wave for vehicles.

The traffic control system can have the following subsystems described in Figure 1:
- sensors, such as video camera, to detect the presence of vehicles (sensors/ positioning system);
- central command computer;
- semaphores (interface man-machine execution element);
- communication of information taken from the field from road sensors to the command computer, and transmission of information back from the command computer to the semaphores;
- an operator interface for the command computer [1-2].

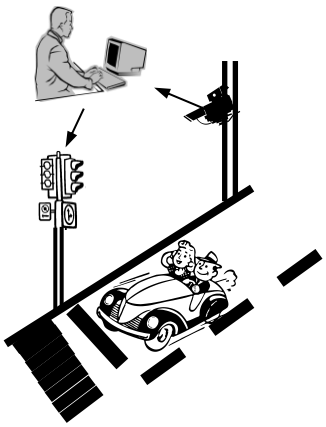


Figure 1. Traffic control system.

PLD (Programmable Logic Devices) are integrated circuits containing a large number of logical gates that can be connected via programmable connections to implement any logical function desired. Programming, as for PROM memories, can be done by the producer via masks or by the user using a dedicated programmer. Most PLD circuits can be re-programmed; consequently, they are particularly useful for the realization of prototypes [3-5].

Programmable logical circuits are of several types:
- programmable logical networks - PLA (Programmable Logic Array), PAL (Programmable Array Logic) and SPLD (Simple Programmable Logic Device);
- Complex Programmable Logic Device – CPLD;
- Field Programmable Gate Array - FPGA.

Programmable logical circuits FPGA are matrices of programmable logical gates in electrical field [3].

The implementation of traffic control systems using FPGA has had a rapid development after its invention by the firm Xilinx in 1984 [6-15].

Cyclone III family combines low cost, low energy consumption and great functionality to maximize the competitive advantage. The features and the architecture of Cyclone III family offer the ideal solution for complex applications [16-18].

2. FPGA DE0 ASPECT AND FEATURES

In Figure 2 is presented the development platform FPGA DE0 [19].

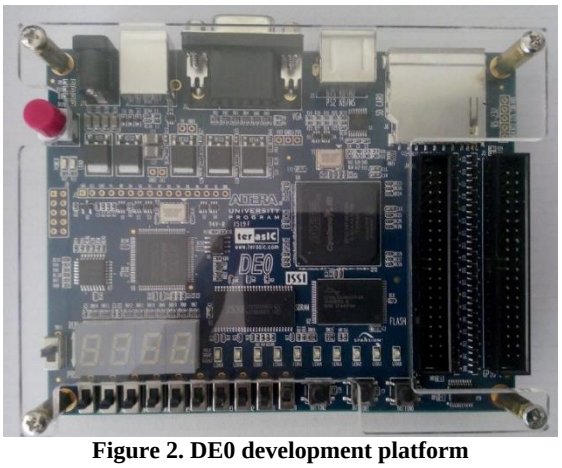


Figure 2. DE0 development platform

DE0 development platform contains the following hardware components [19]**:**

- Altera Cyclone® III 3C16 - FPGA device;
- Altera Serial for configuring the device – EPCS4;
- USB Blaster for programming and management by the user;
- 8-Mbyte SDRAM;
- 4-Mbyte flash memory;
- 10 switches;
- 10 LED's;
- 4 displays of 7 segments (active on 0 logic);
- 50 MHz oscillator;
- RS 232;
- 2x40 connectors.

In Figure 3 we presented the block diagram of DE0 development platform. To offer maximum flexibility for the user, all the connections are realized via the FPGA Cyclone® III device [18]. Consequently, the user can configure the FPGA to implement any design system.

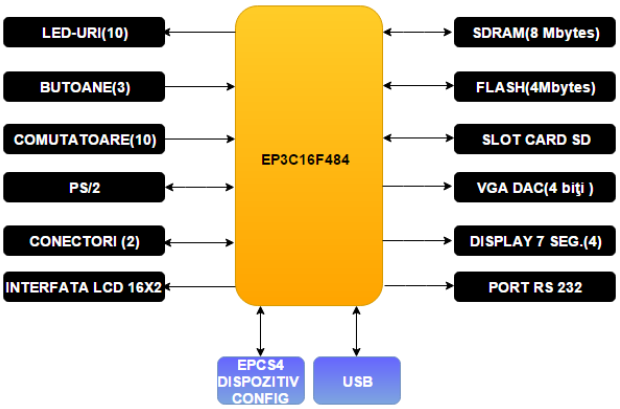


Figure 3. Block diagram of DE0 platform

Cyclone III 3C16 [18]:

- 15,408 logical elements;
- 56 M9K memory blocks;
- 504 Kbits RAM memory;
- 56 multipliers;
- 4 PLL;
- 346 I/O pins;
- 484 pins FBGA (FineLine Ball-Grid Array).

3. APPLICATION

To implement and create the application for monitoring a semaphored crossroads, we used the software Quartus II [20], and for simulation we used ModelSim-Altera [21] (Figure 4).

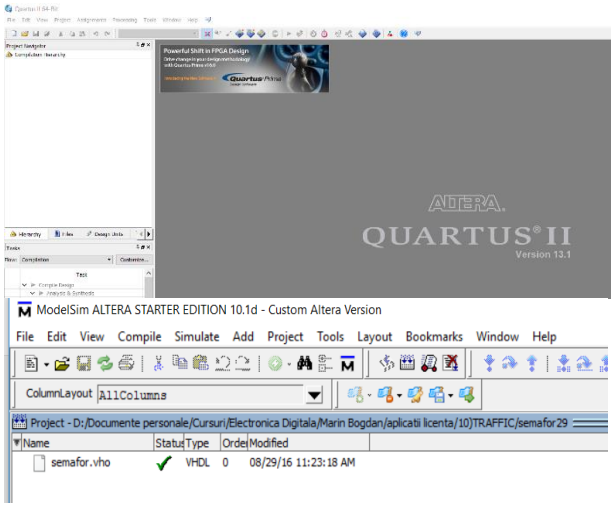


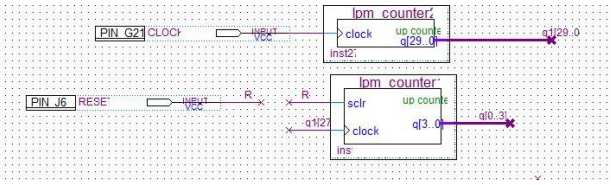
Figure 4. Interface of software's Quartus and ModelSim

The main objective of this work was to design a real-life traffic management system, safe and efficient, using FPGA DE0 platform.

In Figure 5 we presented the crossroads for which we made the application, namely a 4-way crossroads: from North to South and East to West.

Figure 5. Four-way semaphored crossroads (North-South and East-West)

To realize the application in Figure 5 we used the following components:
- lpm_counter2 to divide the 50 MHz frequency;
- lpm_counter1 to realize the reset function based on the frequency divided by lpm_counter2;
- 10 output commanding the LED's of each semaphore (traffic and pedestrians);
- logical gates (Figure 6).



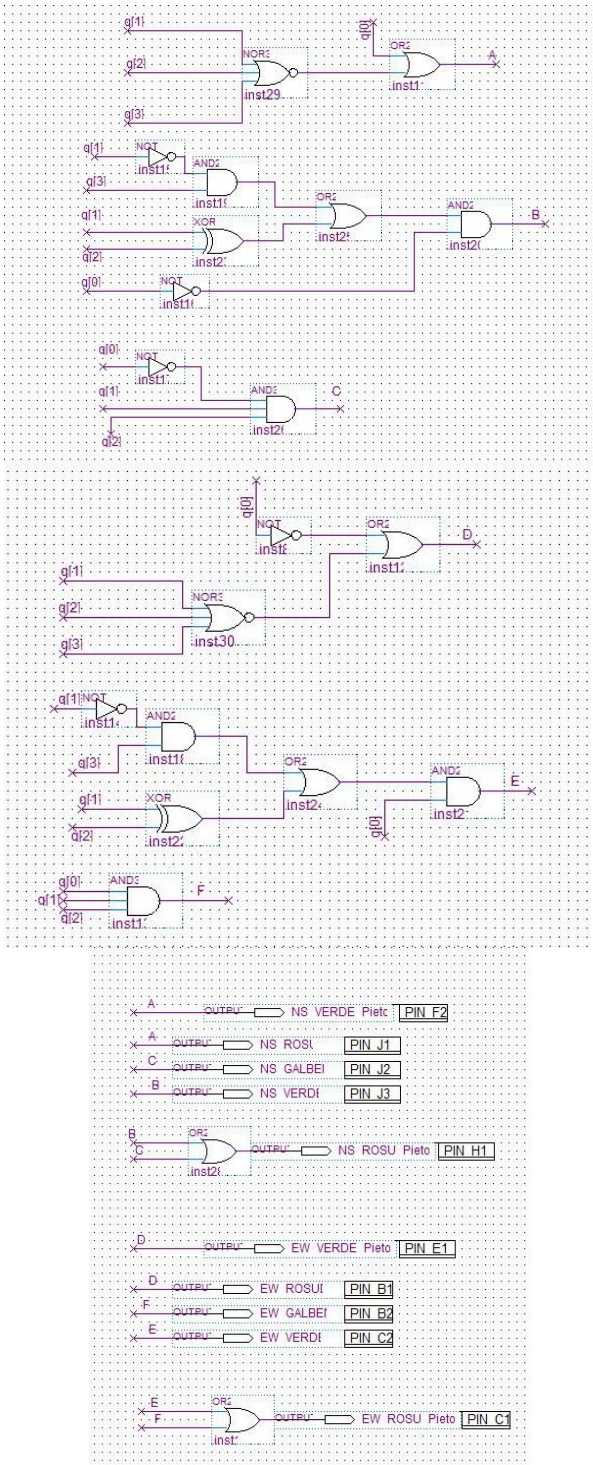


Figure 6. Logical scheme for the application of the semaphored crossroads

To light the colours, we used the variables q[3], q[2], q[1] and q[0] generated by block lpm_counter1 and logical gates. For example, for green colour for pedestrians in the direction NS, the variables q[3], q[2], q[1] and q[0] must take the values 0000, 0001, 0011, 0101, 0111, 1001, 1011, 1101 and 1111.

To simulate the semaphores, we used the 10 green LED's of the platform, transposed from the right to the left of the development platform (Figure 7).

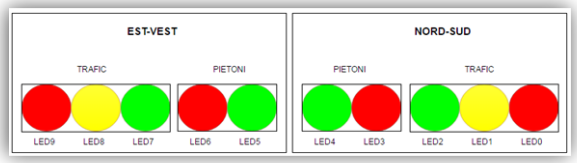


Figure 7. Transposition of the semaphores LED's

The association of the pins between FPGA and the DE0 platform is made according to Figure 8 and the description of the pins is given in Table 1.

Named: * Edit:

Node Name	Direction	Location	I/O Bank	VREF Group
CLOCK	Input	PIN_G21	6	B6_N1
EW_GALBEN	Output	PIN_B2	1	B1_N0
EW_ROSU_Pietoni	Output	PIN_C1	1	B1_N0
EW_ROSUD	Output	PIN_B1	1	B1_N0
EW_VERDE	Output	PIN_C2	1	B1_N0
EW_VERDE_Pietoni	Output	PIN_E1	1	B1_N0
NS_GALBEN	Output	PIN_J2	1	B1_N1
NS_ROSU	Output	PIN_J1	1	B1_N1
NS_ROSU_Pietoni	Output	PIN_H1	1	B1_N1
NS_VERDE	Output	PIN_J3	1	B1_N1
NS_VERDE_Pietoni	Output	PIN_F2	1	B1_N0
RESET	Input	PIN_J6	1	B1_N0
<<new node>>				

Figure 8. Association of the pins between FPGA and DE0

Where: PIN_G21 - internal clock of the platform; PIN_J6 - first switch used to reset the application; PIN_J1, PIN_J2, PIN_J3 - first 3 LED's on the right of the platform, representing the semaphore along the direction North-South; PIN_H1, PIN_F2 – LED's 3 and 4 used for pedestrians along the direction North-South; PIN_E1, PIN_C1 – LED's 5 and 6 used for pedestrians along the direction East-West; PIN_C2, PIN_B2, PIN_B1 - last 3 LED's (7...9), representing the semaphore along the direction East-West.

The way of functioning of the crossroads is described in Table 1 and Figure 9.

Table 1. Transitions of the semaphores

	Traffic East-West			**Pedestrians East-West**	
transitions	red	yellow	green	red	green
1	1	0	0	0	1
2	1	0	0	0	1
3	1	0	0	0	1
4	0	0	1	1	0
5	0	1	0	1	0
	Traffic North-South			**Pedestrians North-South**	
transitions	red	yellow	green	red	green
1	1	0	0	0	1
2	0	1	0	0	1
3	0	0	1	1	0
4	0	0	1	1	0
5	0	0	1	1	0

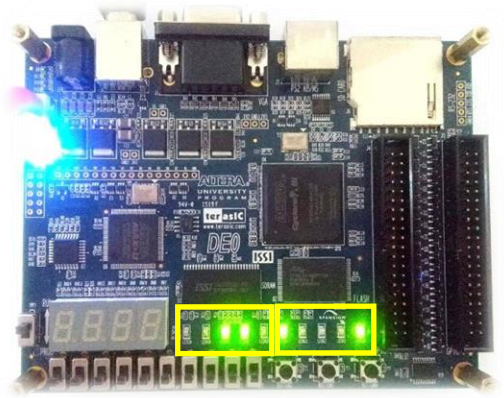


Figure 9. Functioning of the application semaphored crossroads

In the case presented in Figure 9 and out of Table 1, we can notice:
- LED's 0...4 belonging to the semaphore along the direction North-South (for pedestrians LED's 3 and 4 and for cars LED's 0, 1 and 2) are on ⇒ RED - for traffic and GREEN for pedestrians ⇒ cars stop, and pedestrians can cross the street.
- LED's 5...9 belonging to the semaphore along the direction East-West (for pedestrians LED's 5 and 6 and for cars LED's 7, 8 and 9) are on ⇒ RED - for pedestrians and GREEN for traffic ⇒ cars are on the move, and pedestrians cannot cross the street.

In Figure 10 we presented the waveforms corresponding to the application obtained in ModelSim.

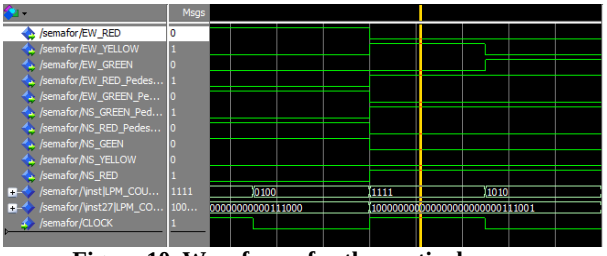


Figure 10. Waveforms for the particular case

Out of Figure 10 one can note that the system implemented passes from the initial state 10001.10001 corresponding to transition 3 according to Table 1, when block lpm_counter1 takes the value 1000, in state 10001.01100, corresponding to transition 1 according to Table 1, when block lpm_counter1 takes the value 0010.

The system designed permits pedestrian traffic monitoring and display by adding a digital clock. To do this, we used the following components:
- 3 counters 7492;
- 3 counters 7490;
- 6 BCD 7 segments to display seconds, minutes and hours;
- one input specific of a switch to reset time;
- a lpm_counter to divide the internal frequency of 50 MHz to 1Hz, value specific of bite 25 of the counter (Figure 11).

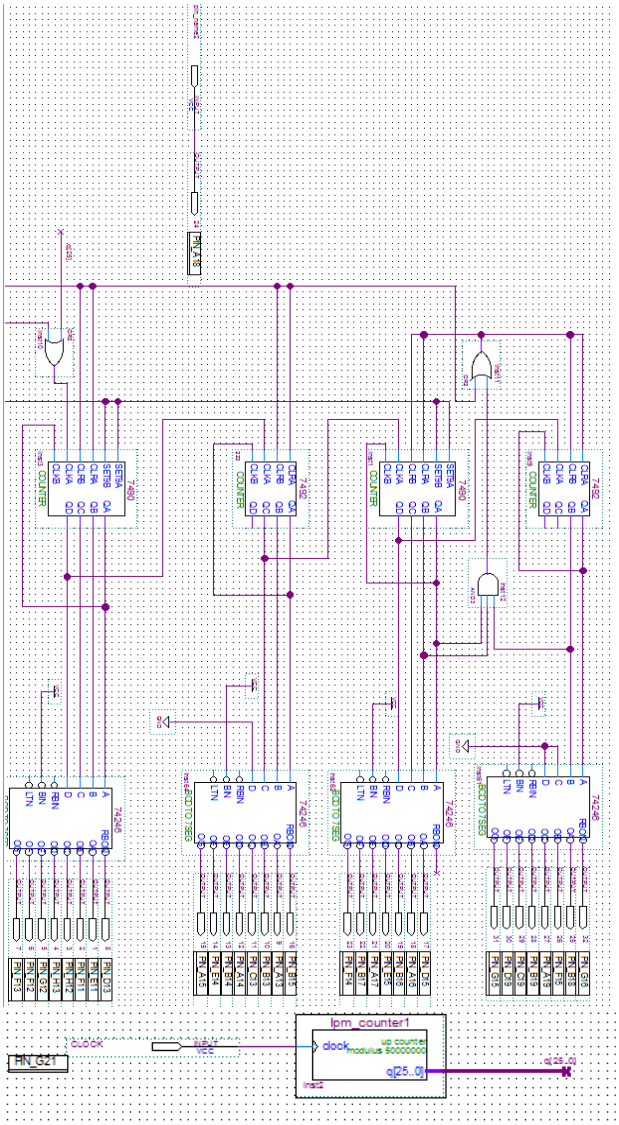


Figure 11. Logical scheme for digital clock

The association of the pins between DE0 and FPGA is given in Figure 12 and the description of the pins allotted to the digital clock application is: PIN_G21 - internal clock (50 MHz); PIN_J6 - switch 1 for resetting the clock (SW[0]); PIN_B1 - last LED on the platform, which lights up at 1 second (LEDG[9]); PIN_E11, PIN_F11, PIN_H12, PIN_H13, PIN_G12, PIN_F12, PIN_F13, PIN_D13 - pins allotted to the first display - 7 segments, this display exhibiting seconds; PIN_A13, PIN_B13, PIN_C13, PIN_A14, PIN_B14, PIN_E14, PIN_A15, PIN_B15 - pins allotted to the second display - 7 segments, this display exhibiting tens of seconds; PIN_D15, PIN_A16, PIN_B16, PIN_E15, PIN_A17, PIN_B17, PIN_F14, PIN_A18 - pins allotted to the third display - 7 segments, this display exhibiting minutes; PIN_B18, PIN_F15, PIN_A19, PIN_B19, PIN_C19, PIN_D19, PIN_G15, PIN_G16 - pins allotted to the fourth display - 7 segments, this display exhibiting tens of minutes.

13	Output	PIN_B14	7	B7_N1	2.5 V (default)
16	Output	PIN_B15	7	B7_N1	2.5 V (default)
19	Output	PIN_B16	7	B7_N1	2.5 V (default)
22	Output	PIN_B17	7	B7_N1	2.5 V (default)
25	Output	PIN_B18	7	B7_N0	2.5 V (default)
28	Output	PIN_B19	7	B7_N0	2.5 V (default)
11	Output	PIN_C13	7	B7_N1	2.5 V (default)
29	Output	PIN_C19	7	B7_N0	2.5 V (default)
8	Output	PIN_D13	7	B7_N1	2.5 V (default)
17	Output	PIN_D15	7	B7_N0	2.5 V (default)
30	Output	PIN_D19	7	B7_N0	2.5 V (default)
1	Output	PIN_E11	7	B7_N1	2.5 V (default)
14	Output	PIN_E14	7	B7_N1	2.5 V (default)
20	Output	PIN_E15	7	B7_N0	2.5 V (default)
2	Output	PIN_F11	7	B7_N1	2.5 V (default)
6	Output	PIN_F12	7	B7_N1	2.5 V (default)
7	Output	PIN_F13	7	B7_N1	2.5 V (default)
23	Output	PIN_F14	7	B7_N0	2.5 V (default)
26	Output	PIN_F15	7	B7_N0	2.5 V (default)
5	Output	PIN_G12	7	B7_N1	2.5 V (default)
31	Output	PIN_G15	7	B7_N0	2.5 V (default)
32	Output	PIN_G16	7	B7_N0	2.5 V (default)
CLOCK	Input	PIN_G21	6	B6_N1	2.5 V (default)
3	Output	PIN_H12	7	B7_N1	2.5 V (default)
4	Output	PIN_H13	7	B7_N1	2.5 V (default)
RESETCLOCK	Input	PIN_J6	1	B1_N0	2.5 V (default)

Figure 12. Association of the pins between DE0 and FPGA for digital clock application

The functioning of BCD-7 segments is given in Table 2. If at entry, we have DCBA=0000 then the result displayed is 0; if DCBA=0001 then the result displayed is 1 etc.

Table 2. Functioning of BCD-7 segments.

Count	Output			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

The functioning of the application is given in Figure 13. The setting of the crossing time for pedestrians is 2 minutes and for cars 4 minutes. In Figure 13 one can note that the cars have 53 seconds to go.

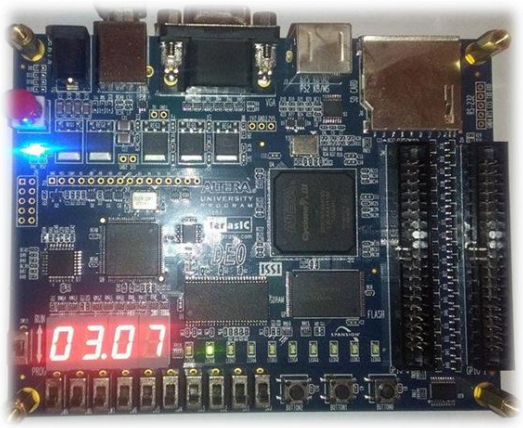


Figure 12. Functioning of the application for the particular case 3 min and 7 seconds

4. CONCLUSIONS

The applications have been realized on the logical circuit level, their implementation being realized via USB using the JTAG programming mode. The implementation of the applications was made using the FPGA of the platform Cyclone III (EP3C16F484C6N), and the results of the applications were displayed via (LED's, displays 7 segments).

The testing of the applications was realized using the software Quartus II and their simulation was realized in ModelSim, both software's being products of the firm Altera.

The proposed *control system of the crossroads semaphores* reduces the waiting time of both cars and pedestrians depending on the traffic data, as it can be programmed by user. DE0 development platform represents a low-cost solution, offering a strong support for the design and implementation as fast as possible of the applications.

5. REFERENCES

[1] G. Predusca, Sisteme de comunicaţii în transporturi utilizând tehnologia GPS, teza de doctorat, Bucuresti, 2006.

[2] Puchianu D., Predușcă G., Simulation model for traffic control on intelligent highways, Cea de-a 14 Conferință Internațională CSCS, 2003, pp.529-534.

[3] F. Ion, G. Predusca, Analiza și sinteza circuitelor numerice. Aplicații și probleme, Editura Bibliotheca, Targoviste, 2006.

[4] Florian Ion, Electronică digitală – memorii ROM, RAM, circuite CPLD, FPGA, Editura Bibliotheca, Târgoviște, 2009.

[5] John F. Wakerly, Circuite digitale - Principiile și practicile folosite în proiectare, Editura Teora, 2002.

[6] Wim Roelandts, 15 years of innovation, Xcell Journal, Issue 32, second quarter, 1999, pp.4.

[7] WM El-Medany, MR Hussain, FPGA – Based advanced real traffic light controller system design, 4th IEEE Workshop on Intelligent data aquisition and advanced computing systems: tehnology and applications, IDAACS, 6-8 September 2007, pp.100-105.

[8] F. Ion, G. Predusca, E. Diaconu, I. Cretoi, Automatic lighting system using complex programmable logic circuits, Scientific Bulletin of the Electrical Engineering Faculty – Year 11 No. 3 (17), 2011, pp.56-60.

[9] G. Predusca, F. Ion, E. Diaconu, C. Duscu, The automation of the car access system using logical circuits FPGA, 13th WSEAS International Conference on Automation & Information (ICAI '12), June 13-15, 2012, Iași, România, publicat in Latest Advances in Information Science, Circuits and Systems, pp.128-132.

[10] Parasmani, Shri Gopal Modani, FPGA – Based advanced traffic light controller simulation, International Journal of Scientific & Engineering

research, volume 4, issue 9, September 2013, pp. 2314-2317.
[11] R. Gao, Design of traffic light circuit based on FPGA, PUBLIC COMMUNICATION OF SCIENCE & TECHNOLOGY, (3):84-84, 81, 2014.
[12] V.V. Dabahde, R.V. Kshirsagar, FGA – based intelligent traffic light controller system design, International journal of innovative science, engineering & technology, vol.2, issue 4, April 2015, pp.1268-1271.
[13] Shilpa U. Holambe, D.B. Andore, Advance traffic light control system based on FPGA, International journal of scientific engineering and research (IJSER), volume 3, issue 11, November 2015, pp.31-35.
[14] P. Swetha Haritha, M.S. Manohar, FPGA implementation of advanced traffic light controller, International Journal of Scientific Engineering and Technology Research, vol.04, Issue 56, December 2015, pp.11926-11930.
[15] Jing Fang, Intelligent traffic light controller design using FPGA digest of technical papers, IEEE International Conference on Consumer electronics (ICCE), 7-11 Jan. 2016, pp.449-452.
[16] http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=364.
[17] http://www.xilinx.com/products/boards_kits/spartan.htm.
[18] Altera, Cyclone III Device Handbook 2012.
[19] Altera, DE0 User Manual.
[20] Introduction to the Quartus® II Software, Version 10.0, Altera, USA, San Jose, California, 2010.
[21] ModelSim-Altera Software Simulation. User Guide, Version 12.1, Altera, USA, San Jose, California, 2013.