[1.] Corina Daniela CUNŢAN, [2.] Ioan BACIU, [3.] Mihaela OSACI

MECHANISM FOR OPERATING AN ACCESS WAY IN A BUILDING WITH COMMAND MADE IN XILINX

[1-3.]University Politehnica Timişoara, University Politehnica Timişoara, Faculty of Engineering of Hunedoara, Electrical Engineering and Industrial Informatics Department, Hunedoara, ROMANIA

Abstract: The work presents a control circuit for closing or opening an access way to a building, made in Xilinx. The control circuit allows the entry of a numeric code. Recognizing the established code determines the action of the mechanism for opening / closing the door. The numeric code consists of three digits, which can be identical, and which are entered from the switches on the development board in the programming state. You can increase the number of digits in the code by adding circuits identical to the existing ones. Switching to the standby state is done with a command on the development board, after which the numerical code can be entered again, the circuit passing into the operating state when the motor for opening the access path is commanded. The programming and actuation states are highlighted with an LED light signal. Entering the wrong code has the effect of blocking the drive circuit.
Keywords: complex logic circuits, development board, control circuit, electromagnetic actuation

1. INTRODUCTION

The Xilinx program is designed for the synthesis, simulation and implementation of fpga circuit projects.
The realization of the control circuit for the operation of the closing/opening mechanism, using the Nexis 4 DDR development board, is done by implementing its electronic scheme in the Xilinx program.
In order to obtain the output command, it is necessary for the control circuit to recognize an access code that is inserted from the existing switches on the development board.
The code number consists of three digits that are inserted in a serial way respecting the sequence established in the memorized code. There is no prohibition to repeat the same number in the code sequence.[2][3]
If a wrong code is entered, the circuit locks and does not accept other commands. In order to unlock the circuit and return to the programming state, it is necessary to activate the circuit reset switch.
If you want to extend the access code, the scheme can be completed with several identical functional blocks and the allocation of additional switches corresponding to the number of code digits added.
The Nexys 4 DDR board is a complete, ready-to-use digital circuit development platform based on the latest Artix-7 Field Programmable Gate Array (FPGA) from Xilinx. With its large, high-capacity FPGA, generous external memories, and collection of USB, Ethernet, and other ports, the Nexys4 DDR can host designs ranging from introductory combinational circuits to powerful embedded processors.[1]
Usually, the Nexis 4 DDR board is powered via a USB cable, but it is also fitted with a connector for an external power source, for example a battery of 3.5 V – 5.5 V.
The Nexis 4DDR board contains: 15,850 logic slices, each with four 6-input LUTs and 8 flip-flops 4, 860 Kbits of fast block RAM, internal clock speeds exceeding 450 MHz, on-chip analog-to-digital converter (XADC), USB-UART Bridge, 12-bit VGA output, Digilent USB-JTAG port for FPGA programming and communication, four Pmod ports and other.

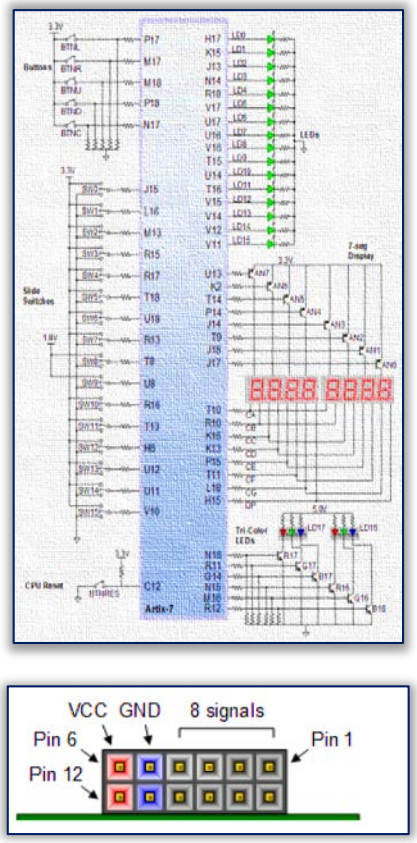

Figure 1. General purpose I/O devices and PMOD connectors on the Nexis4 DDR [1]

The Nexys4 DDR board includes two tri-color LEDs, sixteen slide switches, six push buttons, sixteen individual LEDs, and an eight-digit seven-segment display, as shown in Figure 1. [1]

The Pmod ports are arranged in a 2×6 right-angle, and are 100-mil female connectors that mate with standard 2×6 pin headers. Each 12-pin Pmod port provides two 3.3V VCC signals (pins 6 and 12), two Ground signals (pins 5 and 11), and eight logic signals, as shown in Figure 1. The V_{CC} and Ground pins can deliver up to 1A of current. Pmod data signals are not matched pairs, and they are routed using best-available tracks without impedance control or delay matching. [1]

2. DESCRIPTION OF APPLICATION

In order to implement the electronic circuit on the Nexis 4 DDR development board, it was necessary to design it in Xilinx. The design of the circuit was made hierarchically, in this way having the possibility to check the functioning of each block of the circuit structure.

This circuit must read the code at the input to compare it with the previously memorized code. If the code entered does not correspond to the one stored, the circuit will be blocked. If the code is correctly entered, the command to open the access door is activated. For this, an interface circuit is interposed between the development board and the drive motor for adapting the current and voltage levels between them.

The block diagram of the control circuit (figure 2) made on the basis of functional blocks allows obtaining at the output two signals corresponding to the programming and programming mode respectively.[2][3]

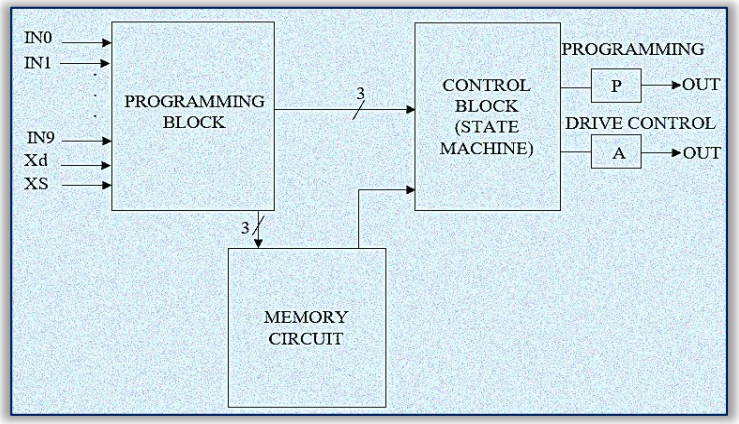

Figure 2. The block diagram of the system

The code introduced in the programming block consists of 3 digits from 1 to 9, the digit 0 is excluded, being used to reset the counting circuit from the programming block structure. The Xd and XS inputs are used to obtain the necessary commands to pass the circuit in the programming state respectively in the actuation state. The simultaneous action of the two has the effect of resetting the circuit in case of entering the wrong code.

The design of the control circuit in Xilinx was made hierarchically using the schematic way of working.

The programming block contains in its structure an encoder, a counter, memory and demultiplexation circuits.

The access code is brought to the encoder entry where the three-digit sequence is formed, which will be stored in the programming state of the circuit in order to compare with the sequence introduced later for action.

The encoder scheme implemented in Xilinx (figure 3) has 9 inputs corresponding to the 9 input digits, respectively four outputs where the binary sequence is obtained for each digit.[2]

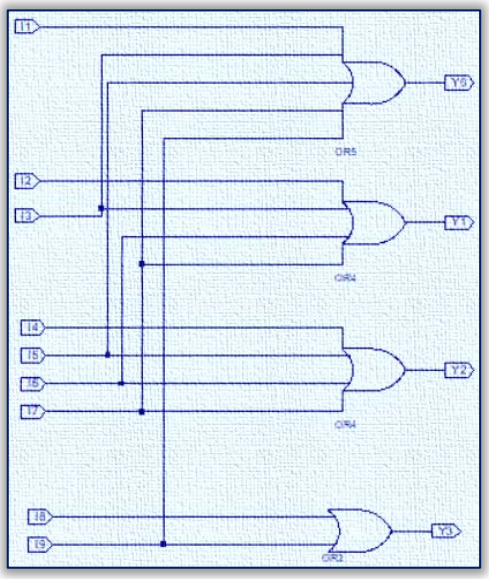

Figure 3. Encoder scheme

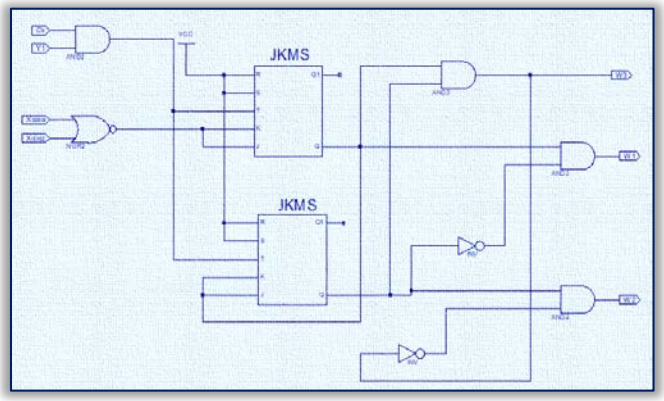

Figure 4. Counter diagram

The diagram of the three-bit counter made in Xilinx (figure 4) allows the selection of each memory circuit in accordance with the figure entered at the input. When entering the first digit of the code, the counter selects the first circuit of memory in which the four-bit value corresponding to the entered digit is recorded. When entering the second digit of the code, the counter selects the next memory circuit in which the binary value of the entered digit will be recorded. Identically, the circuit corresponding to the memorization of the third digit is selected.[2][3][5]

The memory circuit made in Xilinx (figure 5) is selected by the binary sequence at the output of the counter. At its exit, the digital sequence corresponding to the number inserted at the entry of the programming circuit is obtained.

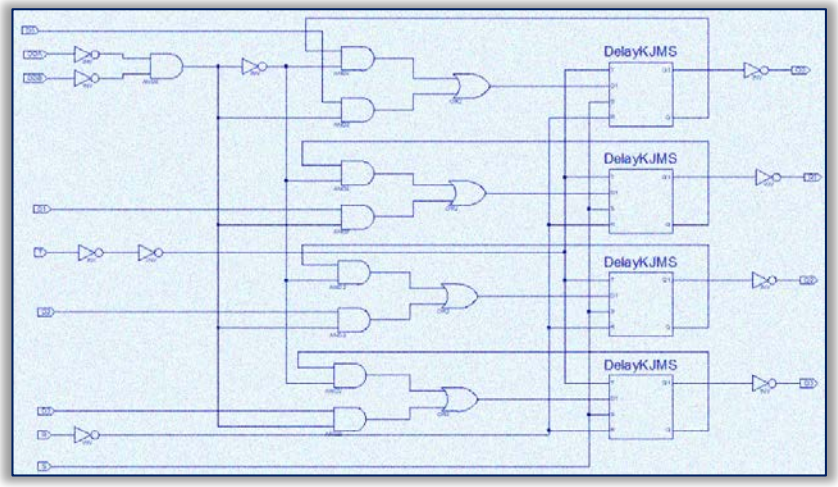

Figure 5. Memory circuit

The demultiplexer made in Xilinx (Figure 6) has 16 analogue inputs and four digital inputs.[5] The 9 switches at the circuit input are connected to 9 inputs of the demultiplexor, which will be transferred at the output depending on the command received from the memory circuit and when accessing the code for operation.[2][3]

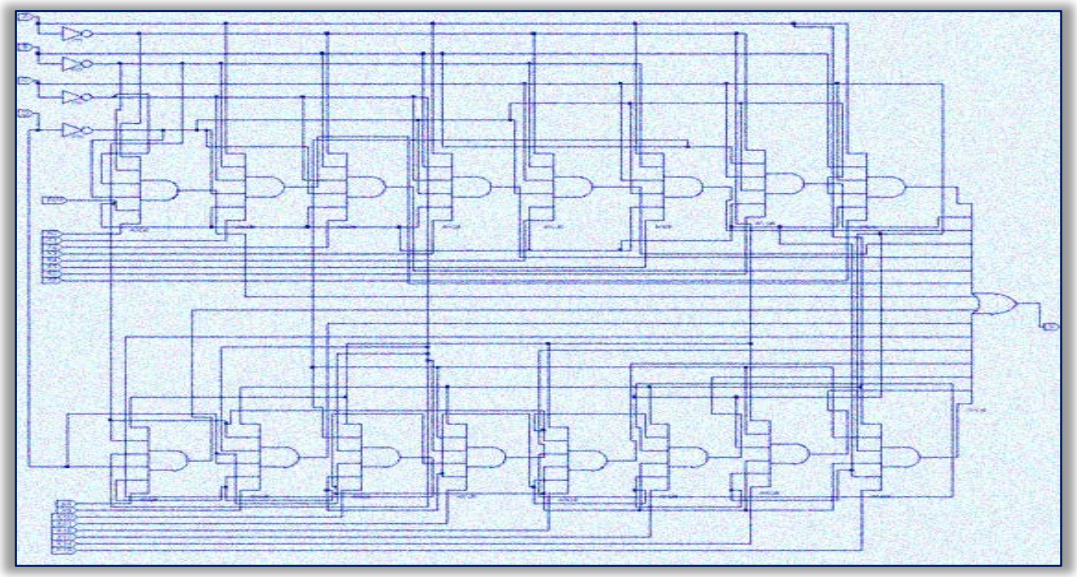

Figure 6. Demultiplexer diagram

At the exit of the demultiplexor, the state of 0 will be maintained until the entry of the figure identical to the memorized figure will be inserted. If the input figure is other than the one memorized, the output of the demultiplexor will not pass into 1 logical and the circuit will lock. [8]

The command block made in Xilinx (Figure 7) provides two signals at the output. One of programming that is activated at the start of the circuit and one of actuation that is activated at the recognition of the entered code.[4]

The block receives at the input the three signals from the outputs of the three demultiplexers when entering the code for the actuation. If the code entered is wrong, the output of the demultiplexor whose input was entered wrongly does not pass into the logical state of 1, which leads to blocking the scheme.[8]

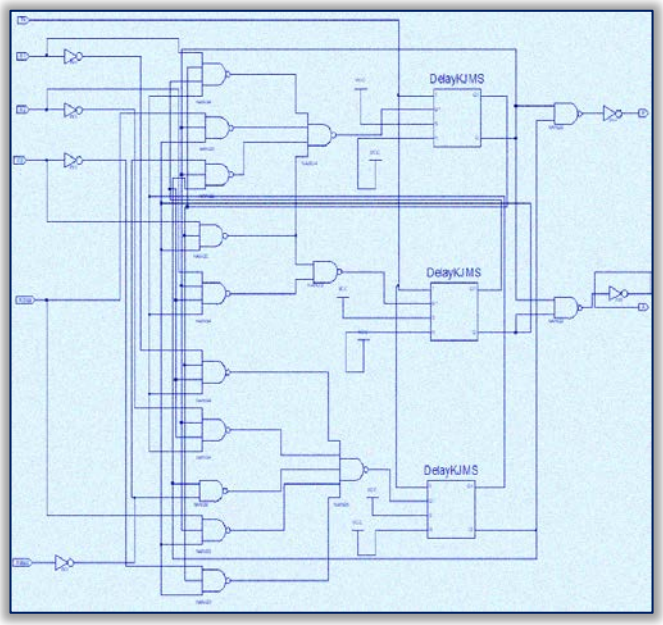

Figure 7. Command block diagram

In order to test the scheme on the development board, each input and output terminal is associated with a pin of the FPGA circuit. For example, the inputs connect to the switches, and the outputs connect to the LEDs and control pins. This is achieved by having a file with the .ucf) extension (User Constraints File (Figure 8) that contains an association between the input/output ports and the pins of the FPGA circuit.[2][4]

```
NET "IN0" LOC=J15 | IOSTANDARD=LVCMOS33; #IO_L24N_T3_RS0_15
NET "IN1" LOC=L16 | IOSTANDARD=LVCMOS33; #IO_L3N_T0_DQS_EMCCLK_14
NET "IN2" LOC=M13 | IOSTANDARD=LVCMOS33; #IO_L6N_T0_D08_VREF_14
NET "IN3" LOC=R15 | IOSTANDARD=LVCMOS33; #IO_L13N_T2_MRCC_14
NET "IN4" LOC=R17 | IOSTANDARD=LVCMOS33; #IO_L12N_T1_MRCC_14
NET "IN5" LOC=T18 | IOSTANDARD=LVCMOS33; #IO_L7N_T1_D10_14
NET "IN6" LOC=U18 | IOSTANDARD=LVCMOS33; #IO_L17N_T2_A13_D29_14
NET "IN7" LOC=R13 | IOSTANDARD=LVCMOS33; #IO_L5N_T0_D07_14
NET "IN8" LOC=T8 | IOSTANDARD=LVCMOS18; #IO_L24N_T3_34
NET "IN9" LOC=U8 | IOSTANDARD=LVCMOS18; #IO_25_34
NET "Xd" LOC=R16 | IOSTANDARD=LVCMOS33; #IO_L15P_T2_DQS_RDWR_B_14
NET "Xs" LOC=T13 | IOSTANDARD=LVCMOS33; #IO_L23P_T3_A03_D19_14
NET "Tact" LOC=H6 | IOSTANDARD=LVCMOS33; #IO_L24P_T3_35
NET"Progr" LOC=D14 | IOSTANDARD=LVCMOS33; #IO_L20N_T3_A19_15
NET"Action" LOC=F16 | IOSTANDARD=LVCMOS33; #IO_L21N_T3_DQS_A18_15
NET "n1" LOC=J13 | IOSTANDARD=LVCMOS33; #IO_L17N_T2_A25_15
NET "n2" LOC=N14 | IOSTANDARD=LVCMOS33; #IO_L8P_T1_D11_14
NET "n3" LOC=R18 | IOSTANDARD=LVCMOS33; #IO_L7P_T1_D09_14
NET "q10" LOC=V17 | IOSTANDARD=LVCMOS33; #IO_L18N_T2_A11_D27_14
NET "q21" LOC=U14 | IOSTANDARD=LVCMOS33; #IO_L22P_T3_A05_D21_14
NET "q32"  LOC=V11 | IOSTANDARD=LVCMOS33; #IO_L21N_T3_DQS_A06_D22_14
```

Figure 8. UCF Files

It will then generate the file with the extension .bit necessary for programming the development board.
The programming command is obtained at pin D14 of the NEXIS4DDR board and the drive command is obtained at pin F16 according to figure 8. For the actual command of the execution element, it is necessary to introduce an interface circuit that makes the adaptation between the output of the development board and the power element. (Figure 9)

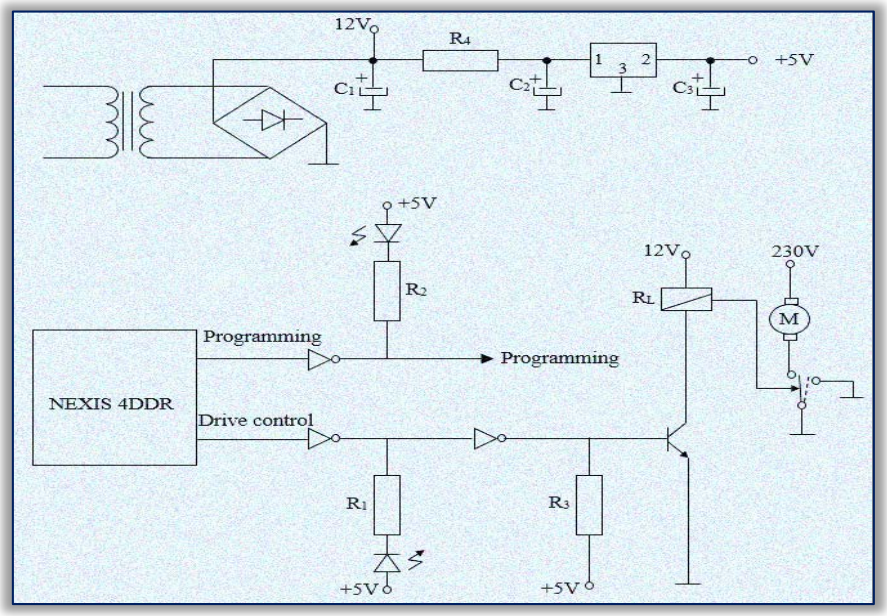

Figure 9. Interface circuit diagram

The drive motor of the access way fed to a single-phase voltage is operated by means of a relay supplied at the voltage of 12V, voltage that is obtained according to figure 9. The relay is operated by means of a medium power transistor that receives the actuation output from the corresponding pin on the development board by means of two inverters that achieve the adaptation in the current. The signalling of the two programming and actuation commands is made with the help of two LEDs of different colours.[9][10]

3. EXPERIMENTAL RESULTS. CONCLUSIONS

The scheme of the control circuit of the actuation mechanism was made in Xilinx. Each block of the control scheme component was individually designed and tested on the Nexis4DDR board.
Through the proper interconnection of the individually realized blocks, the ordering scheme was obtained based on which the file necessary for the programming of the developed board used was generated.
The verification of the correct functioning of the control circuit on the NEXIS4DDR board can be done both optically, by assigning the existing LEDs on the board, and by obtaining at the output pins the signals necessary for the designed programming / operating circuit.

At start-up, the circuit passes into the programming state, signalled by the ignition of the green LED connected in the adaptation circuit. (Figure 10)

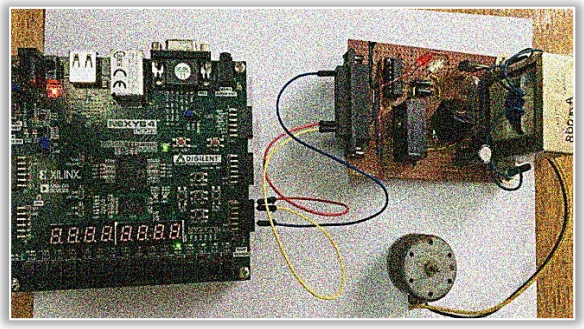

Figure 10. Ignition of the green LED connected in the adaptation circuit (start-up)

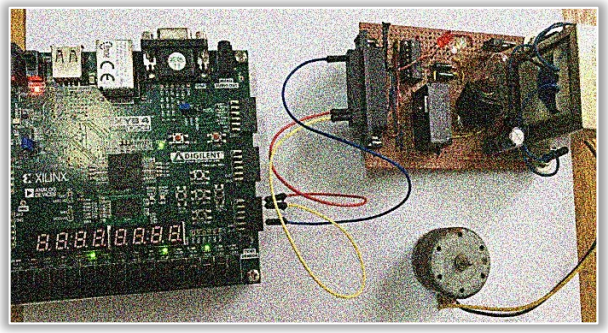

Figure 11. The programming LED

The code will be inserted from the related switches on the development board and then the circuit will be passed to the waiting state at which point the programming LED is turned off. (Figure 11)

When you want to activate the drive command, the correct code is inserted and it is obtained at the output of the command circuit level 1 logically at the pin corresponding to the drive output. This is evidenced by the ignition of the red LED on the adaptation circuit and the start of the engine. (Figure 12)

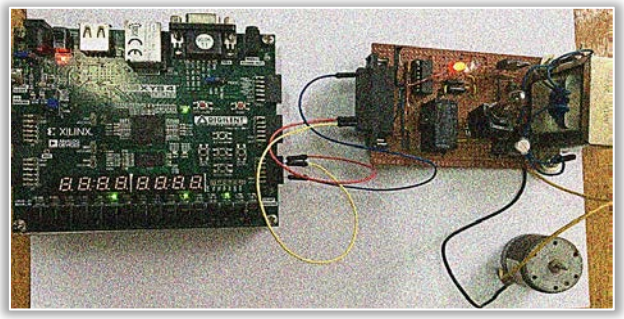

Figure 12. The adaptation circuit and the start of the engine

References

[1] https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual
[2] Pârvu, C., Note de aplicaţie în electronica digitală, Editura: Editura Albastră, Timişoara, 2008;
[3] Haskell R. E., & Hanna, D. M. Digital Introduction to Digital Design Using Digilent FPGA Boards -VHDL Edition. LBE Books, 2009;
[4] Ionel Petrescu, Ionel-Bujorel Păvăloiu, George Drăgoi, Digital Logic Introduction Using FPGAs, Procedia - Social and Behavioral Sciences, Volume 180, 5 May 2015, Pages 1507-1513;
[5] Wakerly J., Circuite Digitale. Principiile şi practicile folosite în proiectare, Editura Teora, Bucuresti, 2002;
[6] Haskell R.E., Hanna D.M., Digital Design Using Digilent FPGA Boards, Publisher LBE Books, Rochester Hills, 2012;
[7] Bostan, I., Metode clasice şi moderne în studiul circuitelor digitale – Lucrări practice de laborator, Editura: Matrix Rom, Bucuresti, 2006;
[8] G. C. Naguboina, K. Anusudha, "Design and synthesis of combinational circuits using reversible decoder in Xilinx," 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), 2017, pp. 1-6
[9] Stojanov I., Borcoci E., Tomescu N., Bulik D., Petec M., Petec C., De la poarta TTL la Microprocessor, Editura Tehnica, Bucuresti, 1987;
[10] Ardelean I., Giuroiu H., Petrescu L.L., Circuite integrate CMOS, Editura Tehnica Bucuresti, 1986;

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665