DOI: 10.2478/sbeef-2025-0001

APPLICATION DEVELOPMENT ON THE NEXYS 4 DDR PLATFORM: TECHNIQUES AND IMPLEMENTATIONS

A.I. OLARU[1], G. PREDUSCA[2]
Valahia University of Targoviste, Faculty of Electrical Engineering, Electronics, and Information Technology
E-mail: alexolaru27@yahoo.com, gabriel.predusca@valahia.ro

Abstract. ***The article explores the application of digital circuits, such as logic gates, logic functions, flip-flop, and automata, using the Nexys 4 DDR platform from Xilinx and the Vivado software. The Nexys 4 DDR, featuring the Artix-7 FPGA, provides a robust environment for designing and testing digital systems. It allows for efficient implementation of various digital functions through hardware programming and simulation. The use of Vivado software eases the creation, simulation, and deployment of custom digital circuits, highlighting the versatility and power of FPGA technology in real-world applications. This paper highlights key principles, practical implementations, and design considerations involved in using these tools for digital circuit applications.***

Keywords: Nexys 4 DDR, FPGA, USB ports, Vivado.

1. INTRODUCTION

Understanding digital electronics is crucial for designing and operating a wide array of applications, ranging from consumer, communication systems, security equipment, and military technology. As these devices become increasingly compact and integrate more advanced technologies, it is essential for engineers and students to develop a strong grasp of the fundamental principles of digital electronics. This includes not only knowledge of digital devices and integrated circuits but also the ability to implement efficient and tailored solutions. Such expertise empowers them to meet specific technical requirements effectively and optimize their designs for various applications [1-4].

Presents are two primary methods of determining the mathematical values of physical quantities: analog and digital. Analog representation expresses values as a continuous range between two limits, allowing theoretically infinite values. The temperature of an oven can be measured as 64°C, 64.96°C, or even 64.9579°C, depending on the precision of the device. Also, voltage in a circuit can vary continuously, such as 6.5V or 6.4969V. Digital representation uses discrete values, typically represented in binary. Temperatures can be expressed in a range of 1°C: 14°C, 15°C, 16°C, etc. Thus, analog representation produces continuous outputs, while digital representation provides discrete outputs. The corresponding systems process the quantities according to their specific form [1, 3-4].

Digital techniques offer advantages such as precision, programmability, high immunity to noise, easy storage, and small size due to integration into circuits. However, most physical quantities, such as speed or temperature, are analog. To receive help from digital advantages, these variables are digitized at input through analog-to-digital converters and reconverted at output with digital-to-analog converters.

A numerical system is defined by its base (radix), the positions of digits, and the maximum number of possible values. The most important parameter is the base, i.e., the number of distinct digits. The decimal system operates on a base of 10, utilizing digits 0 through 9. In contrast, the binary system has a base of 2 and employs only the digits 0 and 1. Meanwhile, the octal system is based on 8, and the hexadecimal system on 16, both of which are widely used in various technical fields.

Automation involves the use of control systems for working equipment, ranging from simple thermostats to complex industrial systems. Applications include manufacturing, network management, vehicle control, and more, reducing human intervention. Automation varies in complexity, from simple on-off control to advanced algorithms.

There are two types of control loops:

- Open-loop, where the controller's action is independent of the output, such as a timer-regulated boiler.
- Closed loop, where control depends on the output, using sensors for feedback, like a thermostat that supports a set temperature.

Closed-loop controllers provide constant feedback, being essential for precise and efficient processes.

Discrete (on-off) control is a simple model used in household applications, such as thermostats, which activate or deactivate an electrical contact based on the system's state.

The PID (Proportional-Integral-Derivative) controller is widely used in industrial systems to correct errors between reference values and process variables, with proportional, integral, and derivative terms [5].

Computerized control systems, including PLCs, enable sequential control and feedback. Modern computers can integrate and analyse data from PLC networks, offering advanced control, graphical views, and real-time reports for operators and engineers [6].

2. FEATURES OF THE NEXYS 4 DDR DEVELOPMENT PLATFORM

The Nexys 4 DDR development platform is an enhanced version of the earlier Nexys 4 system. The key improvement is 128 MiB of DDR2 SDRAM. To simplify the complexity of DDR2, Digilent offers a VHDL reference model, ensuring compatibility with Cellular RAM's SRAM interface. To accommodate the new memory, the FPGA bank output pin has been changed, as well as the constraint file for the projects [1, 7-9].

The Nexys 4 DDR provides an expanded array of ports and peripherals, including those shown in Figure 1: switches, USB-UART port, VGA output, accelerometer, LEDs, USB-JTAG port, displays, Pmod ports, keyboard, and USB stick

The Artix-7 FPGA is optimized for advanced logic, providing superior capabilities, improved performance, and expanded resources. This includes:

- Logic cells (15,850), each with four LUTs with inputs (6) and flip-flops (8).
- 4850 Kbit fast block RAM
- 6 clocks
- Internal clock frequency surpassing 450 MHz
- On-chip integrated analog-to-digital converters.

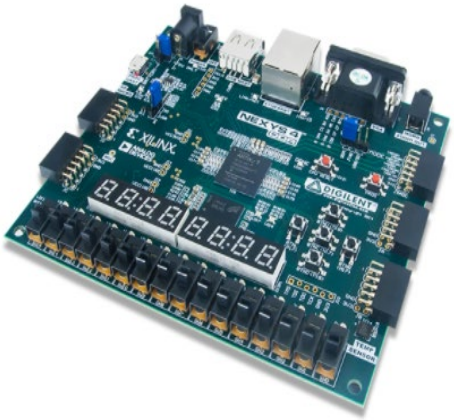

Figure 1. The Nexys 4 DDR development platform [7].

The programming data for the FPGA is stored in bitstream files with the "bit" extension, generated by Xilinx's ISE or Vivado software. These bitstreams are produced using VHDL or schematics stored in the FPGA's SRAM memory. The bitstreams define the logical operates and circuit connections and remain valid until they are removed by removing the power or overwriting with another file. The bitstream for an Artix-7 FPGA has approximately 30 Mb and requires a considerable amount of time for transport.

Xilinx software communicates with the FPGA via the JTAG port, using Digilent USB-JTAG circuits or external programmers such as the Digilent JTAG-HS2. After powering on the Nexys 4 DDR, JTAG programming can be performed, and if the FPGA is previously configured, the active organization is overwritten. The JTAG jumper setting prevents programming from another bitstream source. Using the Vivado hardware server or the iMPACT server from ISE can take approximately five seconds to configure.

The FPGA on the Nexys 4 DDR is volatile, and to keep the configuration between power cycles, a Quad-SPI flash memory is used. During the Master SPI programming process, the FPGA acts as the primary and reads the programming file from the flash memory at startup. Flash programming is done through a two-step process called indirect programming. After configuration, the FPGA can be programmed automatically at startup or reset. Programming may start to 5 minutes due to memory erasure, but FPGA configuration is fast. The configuration speed depends on the file compression and the width of the SPI bus.

The Nexys 4 DDR board has two external memory types: a 1Gib DDR2 SDRAM and a 128Mib non-volatile sequential Flash module. The DDR2 memory is integrated into the platform and communicates with the FPGA via a standard interface. The flash memory is connected to a reserved SPI bus that supports four operational modes (x4).

The Nexys 4 DDR is equipped with a Micron MT47H64M16HR-25: H 16-bit DDR2 memory, powered by 1.8V and connected to the FPGA HR bank. For proper operation, a memory controller and a physical interface (PHY) are needed. There are two options for integrating these: a simple one, using a DDR-to-SRAM adapter compatible with earlier boards, and a more complex one, using the MIG (Memory Interface Generator) expert to generate a native FIFO or AXI4 interface. Although the maximum data rate is 667 Mbps, the 100 MHz system clock limits frequencies to 650 Mbps for simplicity. MIG requires pin-out validation before generating the IP core, and a UCF file for this process is available on the Digilent website.

FPGA programming files are stored in the Quad-SPI flash memory (S25FL128S), and the module sets enable the FPGA to quickly read its organization from this tool at startup. The Artix-7 requires fewer than 4 MiB of memory, which is 77% free for customer data. If the FPGA is constructed from an alternative source, the entire memory becomes available for custom data storage. After configuration, the SPI bus signals, excluding for SCK, are repurposed as typical I/O pins, with access to SCK provided through the STARTUPE2 primitive (Figure 2).

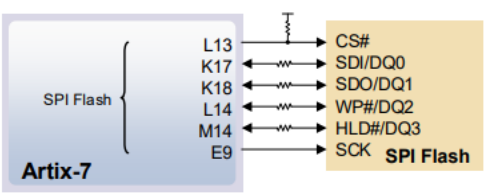

Figure 2. Nexys 4 DDR SPI flash pinout [7].

The Nexys 4 DDR platform has a crystal oscillator (for 100 MHz) connected to the pin E3 (E3 - MRCC input on bank 35). This oscillator can create MMCMs or PLLs to provide diverse frequencies and established stage connections throughout the project. Certain constraints may limit the MMCMs and PLLs driven by the oscillator.

The Nexys 4 DDR platform incorporates an FTDI FT2232HQ USB-UART port (J6), enabling communication with PC applications through standard Windows COM port commands. The PC and FPGA are connected via a two-wire serial port (TXD/RXD) with possible hardware flow inspection (RTS/CTS). LEDs LD20 (send) and LD19 (receive) show data flow. FT2232HQ also manages the USB-JTAG circuitry, with UART and JTAG functioning independently, allowing simultaneous use. Power and configuration are achieved via a Micro USB cable (Figure 3).

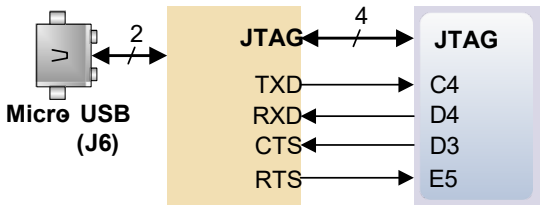

Figure 3. Nexys 4 DDR FT2232HQ links [7].

The microcontroller uses the USB HID protocol and emulates a PS/2 bus, which is a keyboard or mouse. This allows the reuse of existing PS/2 IP cores. The peripherical use a two-wire serial bus (clock and data) to connect to the host. PS/2 data packets contain 11 bits, including a start bit, a data byte, a parity bit, and a stop bit. The keyboard interface enables bidirectional data transfers (Figure 4 and Tabel 1).

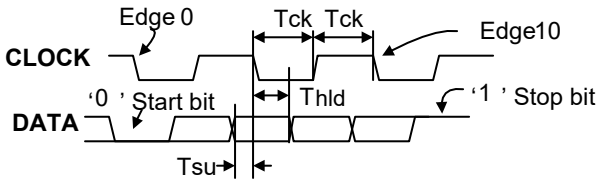

Figure 4. PS/2 host-device timing diagram [7].

Table 1. Timing parameters [7]

Symbol	Parameters	Min	Max
T_{CK}	Clock	30µs	50µs
T_{SU}	Data-to-clock setup time	5µs	25µs
T_{HLD}	Data-to-clock hold time	5µs	25µs

The clock and data signals are only active during data transfers; however, they remain in a high-impedance idle state.

The Nexys 4 DDR board has two tricolour LEDs, 16 slide switches, 6 buttons, 16 individual LEDs, and a seven-segment display. The FPGA's buttons and switches are equipped with resistors to provide short-circuit protection. The buttons are "momentary," and the "CPU RESET" button is used to reset the processor. The slide switches provide constant inputs of value 1 or 0 in condition on their location.

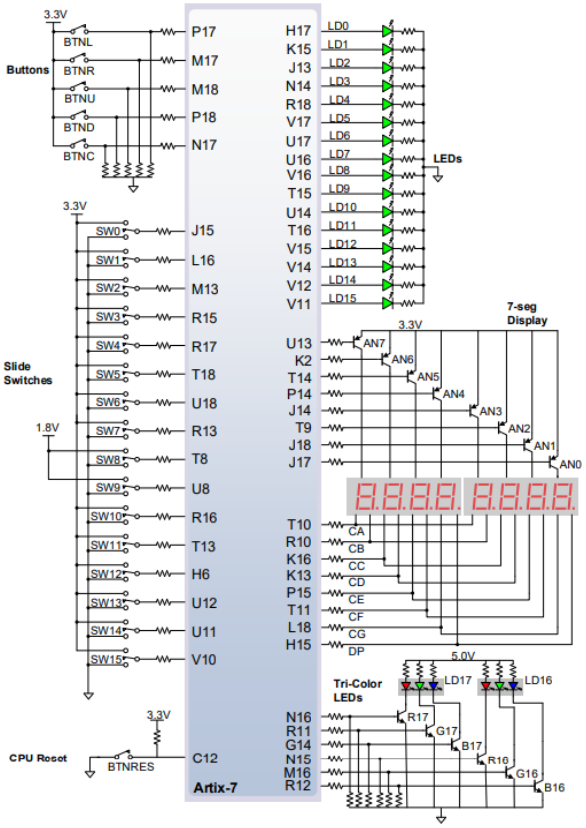

Figure 5. The Nexys 4 DDR has a general-purpose I/O [7].

The Pmod ports are arranged in a 2 (columns) x 6 (rows) matrix and are female connectors that pair with typical 2x6-pin headers. Each 12-pin Pmod interface features two 3.3V VCC signals (on pins 6 and 12), two ground signals (on pins 5 and 11), and eight logic signals (Figure 18). The VCC and Ground pins can supply a maximum current of 1A. The Pmod data signals are not connected and are routed using the optimal paths without impedance control or delay matching.

Figure 6. Pmod ports; front view, loaded on the PCB [7].

3. IMPLEMENTATION OF APPLICATIONS USING NEXYS 4 DDR

Vivado is a comprehensive design suite developed by Xilinx, primarily used for FPGA and SoC design. It supports both RTL (Register Transfer Level) and high-level synthesis, enabling efficient creation of digital circuits. Vivado offers a range of tools for design entry,

simulation, synthesis, implementation, and debugging. It includes a user-friendly graphical interface and integrates with HDL (Hardware Description Language) design tools like VHDL and Verilog. Vivado also offers IP (Intellectual Property) cores, optimization features, and support for advanced debugging techniques to streamline the development process [10-17].

Using the Vivado software and the Nexys4DDR development platform, the following applications were developed and analysed:

1.1 Implementation of logical functions

The following function is proposed for analysis and implementation using Vivado and Nexys4DDR:

$$m = ((a+b)\cdot c)'$$

The truth table corresponding to the function is described below (Table 2).

Table 2. Truth table of the function m

a	b	c	m
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

The waveform corresponding to the function obtained in Vivado is shown in Figure 7.

Figure 7. The waveforms correspond to the function m.

The description of the function using the Nexys4DDR platform is given in Figures 8 and 9.

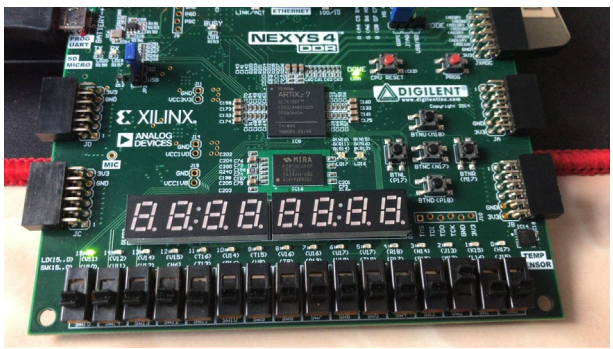

Figure 8. The function m – case abc=100.

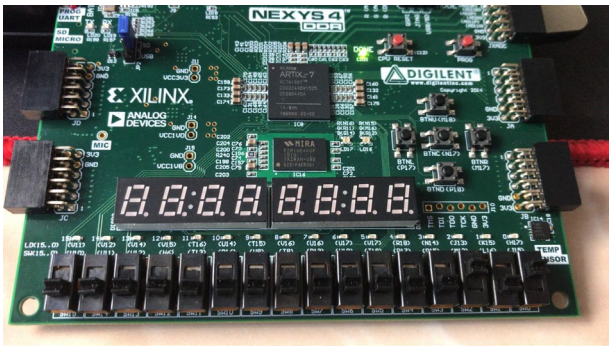

Figure 9. The function m – case abc=101.

The variables a, b, and c are represented by switches 2, 1, and 0 (on the bottom-right of the platform image), and for the output m, I have chosen LED15 (the last LED on the bottom-left). To demonstrate the functionality of the function, I selected two specific cases: case abc=100, where the result is LED15 lit (Figure 7); from the truth table (Table 2), we can see that for this combination, the result is "1"; case abc=101, where the result is LED15 off (Figure 8); from the truth table (Table 2), we can see that for this combination, the result is "0".

1.2 Implementation of combinational logical circuits

The following function with DCD, with active outputs on "0", is proposed for analysis and implementation using Vivado and Nexys4DDR.

$$F=P_1+P_3+P_5+P_7+P_9+P_{11}+P_{13}+P_{15}$$

The circuit implemented with DCD has the schematic shown in Figure 9.

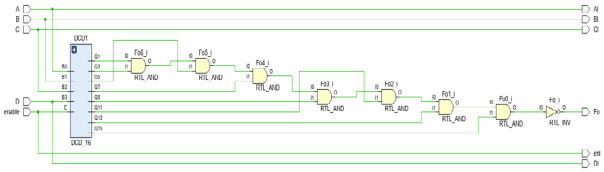

Figure 10. The implementation of the function with DCD in Vivado.

The waveforms corresponding to the function obtained in Vivado are shown in Figure 11.

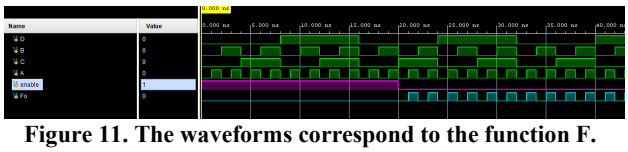

Figure 11. The waveforms correspond to the function F.

The results obtained using the Nexys platform are shown in Figures 11 and 12.

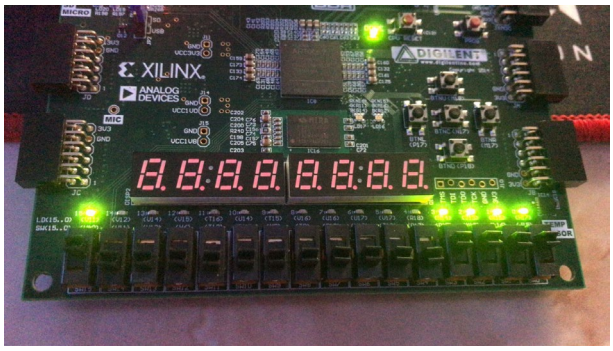

Figure 12. The implementation of the function with DCD – case ABCD = 1111, enable = 0.

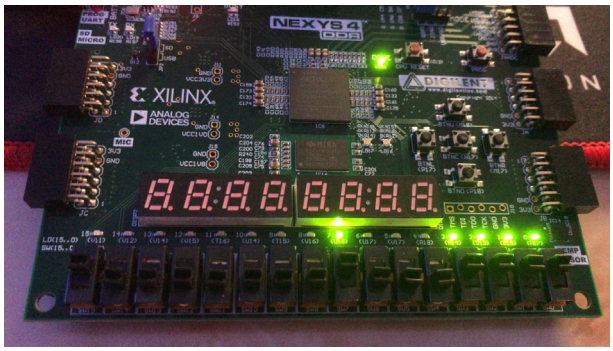

Figure 13. The implementation of the function with DCD – case ABCD = 1111, enable = 1.

The selection bits A, B, C, D are represented by switches 3, 2, 1, 0 (on the bottom-right side of the platform image), and for the output F, I chose LED 15 (the last LED on the bottom-left side). The activation of DCD is done using switch 7 (enable: when it is set to 0, it activates DCD and causes the circuit to display the result, i.e., F=1. When it is set to 1, DCD is deactivated, and the result is F=0). To prove the functionality of the function, I chose two specific cases: when switch 7=0 (enable=0), ABCD=1111, which causes output Q15 to be activated (P_{15}'). But,

$$F = \overline{\bar{P}_1 \cdot \bar{P}_3 \cdot \bar{P}_5 \cdot \bar{P}_7 \cdot \bar{P}_9 \cdot \bar{P}_{11} \cdot \bar{P}_{13} \cdot \bar{P}_{15}} \Rightarrow$$
$$F = \overline{\bar{1} \cdot \bar{1} \cdot \bar{1} \cdot \bar{1} \cdot \bar{1} \cdot \bar{1} \cdot \bar{1} \cdot \bar{1}} \Rightarrow$$
$$F = \overline{0 \cdot 0 \cdot 0 \cdot 0 \cdot 0 \cdot 0 \cdot 0 \cdot 0} => F = 1$$

So, the result is LED15 turned on (Figure 12); from the waveform (Figure 11), for this combination, the result is "1". In the case of switch 7=1 (enable=1), ABCD=1111, output Q15 is activated (P_{15}'). However, when DCD is active on 0 (enable=0), the result is LED15 turned off (Figure 13); from the waveform (Figure 11), for this case, the result is "0".

1.3 Implementation of sequential logical circuits

The sequential logic circuit proposed for analysis and implementation using Vivado and Nexys4DDR is shown in Figure 14.

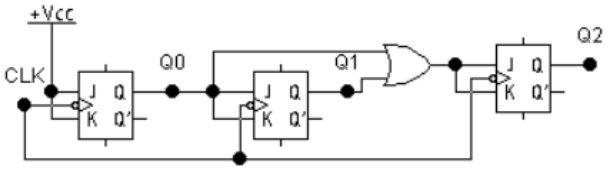

Figure 14. Implementation of sequential logic circuits.

The equations that describe the operation of the circuit are:

$$J_0=K_0=+Vcc=1$$
$$J_1=K_1=Q_0$$
$$J_2=K_2=Q_0+Q_1$$

To stand for the waveform (states of the circuit), we choose the first state $Q_0Q_1Q_2$=000 (Figure 15). The JK flip-flops toggle on the falling edge of the clock pulse.

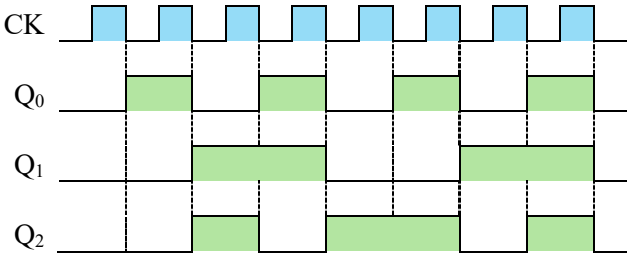

Figure 15. The obtained waveforms.

The obtained states are: $Q_0Q_1Q_2$ = 000 → 100 → 011 → 110 → 001 → 101 → 010 → 111 → 000, which in decimal is 0 → 4 → 3 → 6 → 1 → 5 → 2 → 7 → 0. For the implementation in Vivado, I created the schematic in figure 16. The corresponding waveforms obtained in Vivado are shown in figure 17. Comparing figure 15 with figure 17, the results are identical, so the operation is correct.

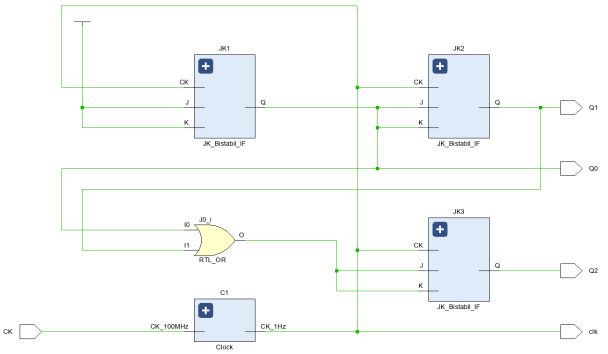

Figure 16. Implementation of sequential logic circuit in Vivado.

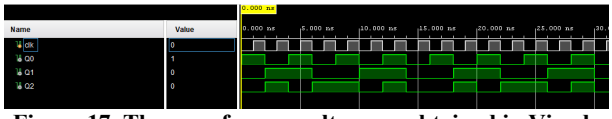

Figure 17. The waveform results were obtained in Vivado.

To verify the operation of the circuit on the Nexys4DDR platform, I used the LED on position 15 to show output

Q_0, the LED on position 14 to show output Q_1, and the LED on position 13 to show output Q_2. I used switch 7 as a clock pulse, which I set to one second. The results are shown in Figures 18-22 and match the earlier solutions (Figure 15 and Figure 17).

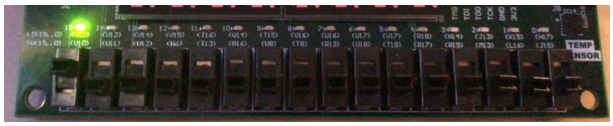

Figure 18. SLC results using the Nexys4DDR platform – Case 1 – CK=0, $Q_0Q_1Q_2$=100.

Figure 19. SLC results using the Nexys4DDR platform – Case 2 – CK=1, $Q_0Q_1Q_2$=011.

Figure 20. SLC results using the Nexys4DDR platform – Case 3 – CK=0, $Q_0Q_1Q_2$=110.

1.4 Implementation of finite state machine

It is proposed to analyse and implement the elementary finite state machine using Vivado and Nexys4DDR as shown in Figure 21.

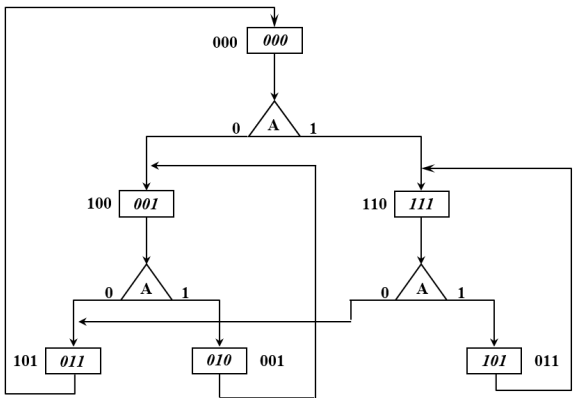

Figure 21. The operational flowchart of the finite state machine.

To implement the finite state machine, the following steps must be completed:

a. The transition table corresponding to the inputs and outputs.

b. Karnaugh diagrams for the flip-flop inputs and outputs.

c. The functions obtained after minimizing the Karnaugh diagrams.

d. The implementation of the sequential circuit using JK flip-flops and gates.

e. Creation of transition tables for states and outputs.

The transition table corresponding to the inputs and outputs is presented below (Table 3).

Table 3. The transition table corresponds to the state machine.

Q_2	Q_1	Q_0	A	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1	K_1	J_0	K_0	Y_2	Y_1	Y_0
0	0	0	0	1	0	0	1	X	0	X	0	X	0	0	0
0	0	0	1	1	1	0	1	X	1	X	0	X	0	0	0
0	0	1	0	1	0	0	1	X	0	X	X	1	0	1	0
0	0	1	1	1	0	0	1	X	0	X	X	1	0	1	0
0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	0	1	1	0	1	X	X	0	X	1	1	0	1
0	1	1	1	1	1	0	1	X	X	0	X	1	1	0	1
1	0	0	0	1	0	1	X	0	0	X	1	X	0	0	1
1	0	0	1	0	0	1	X	1	0	X	1	X	0	0	1
1	0	1	0	0	0	0	X	1	0	X	X	1	0	1	1
1	0	1	1	0	0	0	X	1	0	X	X	1	0	1	1
1	1	0	0	1	0	1	X	0	X	1	1	X	1	1	1
1	1	0	1	0	1	1	X	1	X	0	1	X	1	1	1
1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

In the truth table: Q_2, Q_1, Q_0 represent the current states; Q_2^+, Q_1^+, Q_0^+ represent the next states; J_2K_2, J_1K_1, J_0K_0 are the inputs corresponding to the flip-flops; Y_2, Y_1, Y_0 are the outputs corresponding to the states.

The functions obtained after minimizing the Karnaugh diagrams:

$$J_2 = 1,\ K_2 = A + Q_0$$
$$J_1 = Q_2'Q_0'A,\ K_1 = Q_0'A'$$
$$J_0 = Q_2,\ K_0 = 1$$
$$Y_2 = Q_1,\ Y_1 = Q_1 \oplus Q_0,\ Y_0 = Q_1 + Q_2$$

The circuit designed and simulated in Vivado is shown in Figure 22, and the corresponding waveforms are depicted in Figure 23.

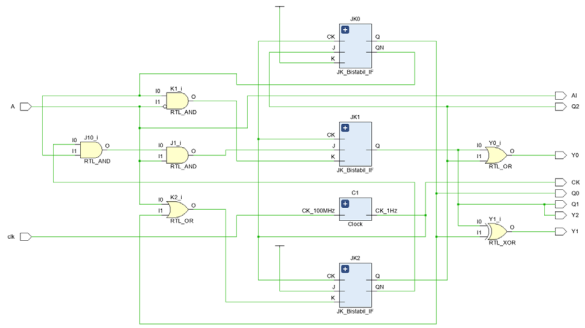

Figure 22. The finite state machine implemented in Vivado.

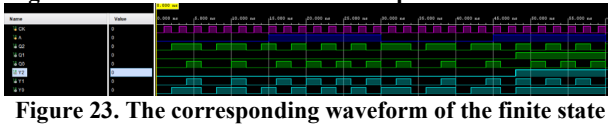

Figure 23. The corresponding waveform of the finite state machine.

To verify the operation of the circuit on the Nexys4DDR platform, I used the LED on position 15 to show the

output Q_2, the LED on position 14 to show the output Q_1, and the LED on position 13 to show the output Q_0. LEDs 11, 10, and 9 were used to display the results corresponding to the outputs Y_2, Y_1, and Y_0. Switch 7 was used as the clock pulse, which was set to one second. Switch 0 is used for the synchronization input A of the automaton. The results are shown in Figures 24-26 and correspond to the earlier solutions (Figure 23 and Table 3).

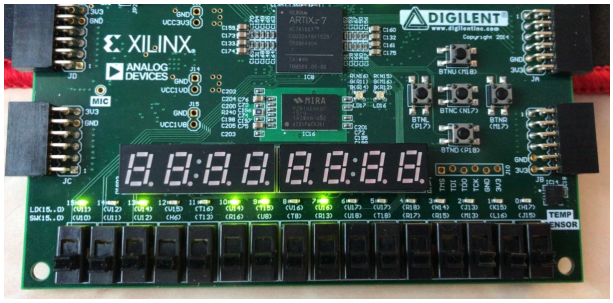

Figure 24. Operation of the finite state machine – case 1 A=0, $Q_2Q_1Q_0$=101, $Y_2Y_1Y_0$=011.

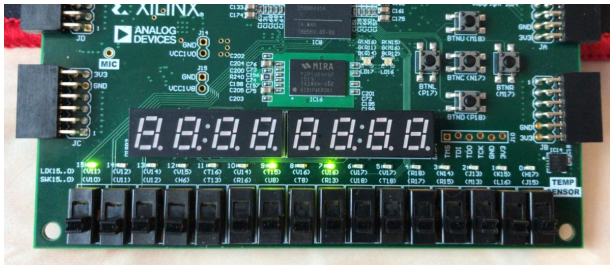

Figure 25. Operation of the finite state machine – case 2 A=0, $Q_2Q_1Q_0$=100, $Y_2Y_1Y_0$=001.

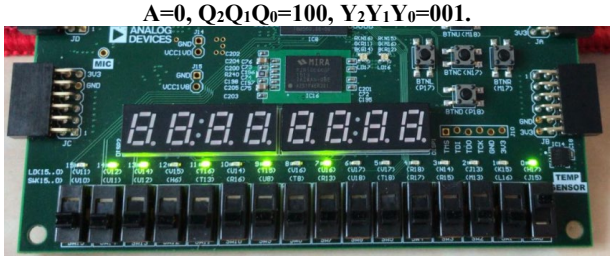

Figure 26. Operation of the finite state machine – case 3 A=1, $Q_2Q_1Q_0$=011, $Y_2Y_1Y_0$=101.

The transition tables for states and outputs are given below (Table 4 - 5).

Table 4. The transition table corresponds to the states of the finite state machine.

$Q_2Q_1Q_0$ / A	000	001	010	011	100	101	110	111
0	100	100	xxx	110	101	000	101	xxx
1	110	100	xxx	110	001	000	011	xxx

Table 5. The transition table corresponds to the outputs of the finite state machine.

$Q_2Q_1Q_0$ / A	000	001	010	011	100	101	110	111
0	001	001	xxx	111	011	000	011	xxx
1	111	001	xxx	111	010	000	101	xxx

4. CONCLUSIONS

The Nexys 4 DDR development platform, equipped with the Artix-7 FPGA, offers high performance and extensive resources for the development and testing of digital circuits. Components such as DDR2 memory, USB-JTAG interface, and multiple ports and peripherals make this platform a versatile choice for educational and research applications [18-20].

Vivado, by integrating design and simulation tools, helps the development of combinational logic circuits, sequential circuits, and finite state machines. The analysis of results obtained on the Nexys 4 DDR platform confirms the correctness of the designed circuits' functionality.

The article proved the successful implementation of logic functions, combinational and sequential circuits, as well as a finite state machine on the Nexys 4 DDR platform. These implementations highlight the platform's usefulness for direct learning and application development.

The integration of FPGAs in automated systems opens new possibilities for precise industrial process control and reducing human intervention. The use of controllers such as PID and PLCs, combined with digital platforms, offers advanced solutions for automation [5, 21-22].

The Nexys 4 DDR platform, together with the Vivado software, is a valuable tool for students and engineers, providing an in-depth understanding of digital electronics principles and their implementation in real-world applications.

5. REFERENCES

[1] Alexandru-Ionuț OLARU, Dezvoltarea de aplicații utilizând Nexys 4 DDR, Proiect de diplomă, Târgoviște, 2021.

[2] F. Ion, G. Predușcă, Analiza și sinteza circuitelor numerice. Aplicații și probleme, Ed. Bibliotheca, Târgoviște, 2006.

[3] S.L. Harris, D.M. Harris, Digital Design and Computer Architecture: ARM Edition, Morgan Kaufmann Publishers Inc., San Francisco, 2015.

[4] M.M. Mano, C.R. Kime, T. Martin, Logic and Computer Design Fundamentals (5th edition), Pearson, 2015.

[5] J. Graf, PID control fundamentals, CreateSpace Independent Publishing Platform, 2016.

[6] W. Bolton, Programmable Logic Controllers (PLCs), Newnes, 2015.

[7] [Online], available: https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-getting-started-with-microblaze/, accessed on December 14, 2024.

[8] [Online], available: https://www.xilinx.com/support/university/boards-portfolio/xup-boards/DigilentNexys4DDR.html#docs, accessed on December 14, 2024.

[9] R.J. Kachhadiya, Y. Agrawal, R. Parekh, Implementation of ALU using RTL to GDSII flow and on NEXYS 4 DDR FPGA board, Second International Conference on

Electronics and Sustainable Communication Systems (ICESC), pp.481-488, 2021.
[10] [Online], available: https://www.xilinx.com/products/design-tools/vivado.html, accessed on December 14, 2024.
[11] S. Churiwala, Designing with Xilinx® FPGAs: Using Vivado, Spinger, 2017.
[12] R.E. Haskell, D.M. Hanna, Digital Design Using Digilent FPGA Boards: VHDL Vivado Edition, LBE Books, 2019.
[13] V.A. Pedroni, Finite State Machines in Hardware: Theory and Design (with VHDL and SystemVerilog), The MIT Press, 2013.
[14] C.M. Singh, H.B. Singh, G. Ahmad, Implementation of Logic Gates and Combinational Circuits Using Neural Network On FPGA For Neuromorphic Hardware, IEEE World Conference on Applied Intelligence and Computing (AIC), pp.113-118, 2023.
[15] P.P. Chu, FPGA Prototyping by VHDL Examples: Xilinx MicroBlaze MCS SoC, John Wiley & Sons, 2017.
[16] C. Unsalan, B. Tar, Digital System Design with FPGA: Implementation Using Verilog and VHDL (1st edition), McGraw Hill, 2017.
[17] F. Bruno, FPGA Programming for Beginners, Packt Publishing, 2021.
[18] [Online], available: https://web.mit.edu/6.111/www/f2016/handouts/labs/lab4nv2.html, accessed on December 14, 2024.
[19] [Online], available: https://www.secs.oakland.edu/~llamocca/Tutorials/VHDLFPGA/Unit%207.pdf, accessed on December 14, 2024.
[20] [Online], available: https://ren-fengbo.lab.asu.edu/content/cse-320-designsynthesis-digital-hardware, accessed on December 14, 2024.
[21] [Online], available: https://www.microchip.com/en-us/solutions/industrial/fpga, accessed on December 14, 2024.
[22] C. Stanescu, F.G. Apostol, G. Predusca, D. Circiumarescu, N. Angelescu, D. Puchianu, IoE simulation with Cisco Packet Tracer, 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp.1-6, 2023.