DOI: 10.35598/mcfpga.2026.025

Application of Integrated Logic Analyzer for Debugging FPGA-Based Systems

Oleksandr Vorgul
ORCID 0000-0002-7659-8796
dept. Department of Media Engine-ering and Information Radioelectronic Systems
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleksandr.vorgul@nure.ua

Olga Markina
ORCID 0000-0002-3917-2008
dept.Foreign Languages
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
olha.markina@nure.ua

Matvii Tkalenko
dept. Department of Computer-Integrated Technologies and Robotics
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
matvii.tkalenko@nure.ua

Abstract–This paper investigates the use of the Integrated Logic Analyzer (ILA) for debugging FPGA-based systems implemented on the Nexys 4 DDR platform with Xilinx Artix-7 FPGA. A UART communication module was implemented as a test system. Internal signals were analyzed in real time using Vivado ILA, enabling efficient observation of system behavior and timing.

Keywords— FPGA, Artix-7, Nexys 4 DDR, Logic Analyzer, Vivado, Integrated Logic Analyzer, Digital Design.

I. INTRODUCTION

The increasing complexity of FPGA projects requires efficient debugging tools capable of analyzing internal hardware operation. Logic analyzers are among the most important instruments used for monitoring signals, detecting timing violations and verifying digital designs. The purpose of this work is to investigate the use of logic analyzers in FPGA projects developed on the Nexys 4 DDR Artix-7 FPGA Trainer Board.

Field-Programmable Gate Arrays (FPGAs) have become an important component of modern embedded systems due to their flexibility, reconfigurability, and ability to implement highly parallel processing algorithms. Unlike traditional microcontrollers and processors, FPGAs allow designers to create custom hardware architectures optimized for specific application requirements. As a result, FPGA devices are widely used in industrial automation, telecommunications, aerospace systems, automotive electronics, digital signal processing, and scientific research equipment.

The growing complexity of modern FPGA designs has introduced new challenges in the verification and debugging process. Contemporary systems often consist of multiple interconnected modules operating simultaneously and exchanging large amounts of data. Under such conditions, identifying the source of functional errors, timing violations, or synchronization problems becomes increasingly difficult. Traditional debugging approaches based solely on simulation may not always reveal issues that occur under real operating conditions after implementation on physical hardware.

Therefore, integrated debugging solutions provided by modern development environments play a crucial role in ensuring correct hardware operation and reducing development costs. Among these tools, logic analyzers provide designers with the ability to observe signal behavior, analyze timing relationships, and verify communication between hardware modules. Such capabilities significantly simplify the debugging process and improve the reliability of FPGA-based systems.

Modern FPGA development platforms, including devices manufactured by AMD Xilinx, incorporate advanced debugging technologies directly into the design environment. One of the most widely used solutions is the Integrated Logic Analyzer (ILA), which allows monitoring of internal FPGA signals in real time without requiring external measurement equipment. The use of such tools accelerates hardware verification, simplifies fault localization, and increases overall development efficiency.

This paper focuses on the practical application of the Vivado Integrated Logic Analyzer for debugging FPGA-based systems implemented on the Nexys 4 DDR development board equipped with the Xilinx Artix-7 FPGA. The study investigates the principles of ILA operation, its integration into FPGA projects, and its effectiveness in monitoring internal hardware signals during experimental testing.

II. FEATURES OF THE NEXYS 4 DDR PLATFORM

The Xilinx Artix-7 XC7A100T FPGA used on the Nexys 4 DDR board provides a significant amount of programmable logic resources, making it suitable for implementing a wide variety of digital systems. The device contains configurable logic blocks, embedded memory resources, dedicated DSP slices, and flexible input/output interfaces. These resources enable the development of applications ranging from simple digital controllers to advanced communication and signal-processing systems.

An important characteristic of the platform is the availability of external DDR2 memory, which can be utilized for data buffering, storage of intermediate computation

results, and implementation of memory-intensive applications. Access to external memory allows designers to develop more sophisticated systems and evaluate memory management techniques commonly used in modern digital devices.

The board also supports the implementation of various communication protocols and embedded hardware architectures. Through the available interfaces and expansion connectors, developers can integrate sensors, displays, communication modules, and other peripheral devices into their projects. This flexibility makes the platform suitable for both laboratory experiments and practical engineering applications.

Furthermore, the combination of hardware resources and software development tools enables comprehensive investigation of FPGA design methodologies. Students and researchers can perform the complete development cycle, including hardware description, simulation, synthesis, implementation, debugging, and system verification, using a single integrated platform. As a result, the Nexys 4 DDR board serves not only as a development tool but also as an effective educational environment for learning modern digital design techniques.

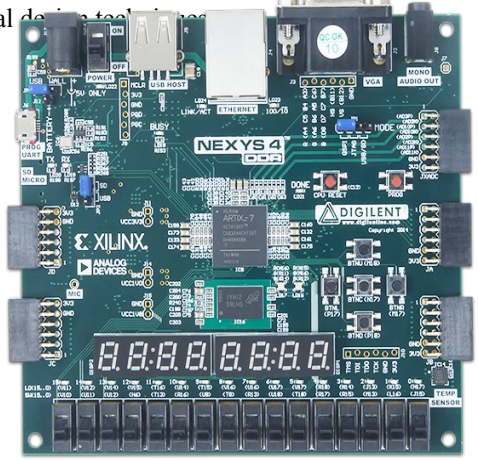

Fig. 1. Nexys 4 DDR Artix-7 FPGA Trainer Board

III. LOGIC ANALYZER IN FPGA DEVELOPMENT

The development of FPGA-based systems requires effective methods for monitoring internal signals and verifying hardware functionality. Unlike software applications, internal FPGA processes cannot be observed directly during operation.

In the Xilinx Vivado Design Suite, the Integrated Logic Analyzer (ILA) can be embedded directly into the FPGA design and connected to selected internal signals. This allows observation of register behavior, finite-state machine transitions, communication buses and timing relationships between modules in real time. Captured data is transferred through the JTAG interface and displayed as timing diagrams inside Vivado.

Compared to external logic analyzers, the integrated ILA solution does not require additional measurement equipment and provides access to internal FPGA resources that are normally unavailable on external pins.

The use of logic analyzers becomes particularly important when debugging communication protocols such as UART, SPI, I2C, and Ethernet controllers. These interfaces often involve strict timing requirements that cannot always be verified through simulation alone.

Modern FPGA debugging environments also support advanced triggering mechanisms. Triggers may be configured to activate on specific signal values, combinations of events, or state machine transitions.

Another significant benefit of integrated debugging tools is the reduction of development costs. Since the ILA utilizes resources already available within the FPGA device, there is no need for expensive external measurement equipment.

Another important advantage of the ILA tool is its flexibility during the verification process. The designer can select different groups of internal signals, modify trigger conditions, and reconfigure capture parameters without changing the functional behavior of the implemented hardware system.

In addition, the use of embedded debugging tools contributes to a better understanding of hardware timing characteristics. Engineers can observe signal propagation delays, synchronization mechanisms, and interactions between independent modules.

The Integrated Logic Analyzer can also be combined with other Vivado debugging features, including Virtual Input/Output (VIO) modules and hardware management tools.

The interaction between the FPGA design and the Inte

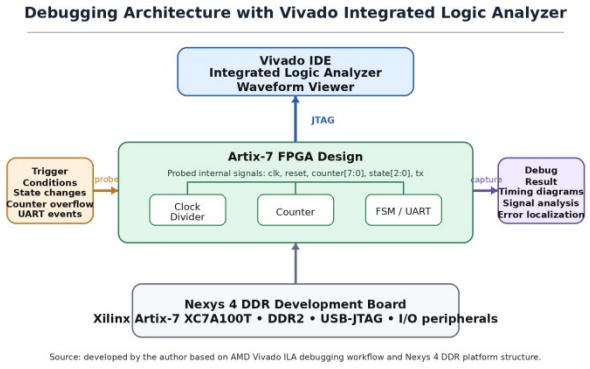

Fig. 2. Architecture of the debugging system based on the Vivado Integrated Logic Analyzer. Source: developed by the author.

IV. EXPERIMENTAL INVESTIGATION

The experimental design was implemented on the Nexys 4 DDR development board based on the Xilinx Artix-7 XC7A100T FPGA. The project included several digital modules, namely a clock divider, binary counter, UART

communication interface, finite-state machine, and LED control subsystem. These modules were selected because they represent common building blocks used in FPGA-based systems.

The UART communication module was selected as the primary object of investigation because it represents one of the most frequently used communication interfaces in FPGA-based embedded systems. UART is commonly employed for communication between programmable logic devices, microcontrollers, sensors, and personal computers. Due to its simplicity and widespread adoption, it provides an appropriate platform for evaluating the effectiveness of hardware debugging techniques.

To investigate the debugging capabilities of the Integrated Logic Analyzer, several internal signals were connected to the ILA core. The monitored signals included clock pulses, counter outputs, state machine registers, UART transmission signals, and control flags generated by the hardware logic. The ILA was configured to capture data using trigger conditions based on state transitions and signal changes.

Several experimental scenarios were developed to assess the capabilities of the Integrated Logic Analyzer under different operating conditions. These scenarios included normal UART data transmission, activation of reset signals, counter overflow events, and transitions between finite-state machine states.

During testing, the FPGA operated with a system clock frequency of 100 MHz. The captured timing diagrams provided detailed information about signal propagation, synchronization processes, and interactions between hardware modules. The obtained waveforms allowed direct observation of the behavior of internal registers and communication signals without routing them to external FPGA pins.

Particular attention was paid to synchronization processes occurring between the individual modules of the system. The captured waveforms enabled verification of clock signal distribution, timing relationships between hardware blocks, and proper synchronization of sequential logic elements. This information proved valuable for confirming that the implemented design operated according to the expected functional specification.

Additional measurements were performed to analyze the behavior of the finite-state machine controlling the communication process. The recorded data showed that all state transitions occurred according to the implemented design logic. The timing diagrams also provided a clear representation of the interaction between the state machine, counter modules, and UART controller.

The collected timing diagrams were compared with the results obtained from functional simulation in the Vivado environment. The comparison demonstrated a high level of consistency between simulated and measured behavior. At the same time, hardware-based observation provided additional insight into implementation-specific timing effects that cannot always be identified during simulation alone.

The use of the Integrated Logic Analyzer proved particularly valuable during the investigation of reset conditions. By observing internal registers and control signals immediately after reset activation, it was possible to verify that all modules returned to their initial states correctly.

An additional benefit observed during the investigation was the ability to perform repeated measurements under identical operating conditions. This made it possible to compare different design revisions and evaluate the effectiveness of implemented corrections.

The performed experiments demonstrated that the use of the Integrated Logic Analyzer significantly simplifies the debugging process. Several synchronization issues related to reset signal timing and state transitions were identified and corrected. The ability to monitor internal FPGA resources in real time reduced the time required for fault localization and improved the reliability of the developed hardware design.

V. PERSPECTIVES FOR FURTHER RESEARCH

While the current study successfully demonstrated the effectiveness of the Integrated Logic Analyzer for debugging standard-speed digital modules, future research will focus on more complex and high-speed FPGA-based systems. The transition to advanced communication interfaces, such as PCIe, Gigabit Ethernet, and DDR memory controllers, introduces significant challenges in signal integrity and timing closure. Investigating the application of high-speed serial ILA probes and advanced trigger mechanisms will be essential for analyzing these modern protocols.

Another promising direction is the integration of automated analysis and machine learning techniques into the hardware debugging workflow. As FPGA designs grow in complexity, the volume of captured waveform data increases exponentially, making manual inspection by engineers increasingly time-consuming. Future work will explore the development of algorithms capable of automatically detecting timing violations, identifying anomalous signal behavior, and predicting potential hardware failures based on historical ILA data.

Furthermore, the scope of research will be expanded to include Hardware-Software Co-debugging for System-on-Chip (SoC) architectures, such as the Xilinx Zynq platform. Combining the FPGA ILA with software debuggers will allow for a comprehensive analysis of complex embedded systems, bridging the gap between hardware logic and embedded software execution.

CONCLUSIONS

This paper investigated the application of logic analyzers in FPGA projects implemented on the Nexys 4 DDR Artix-7 development platform. The analysis demonstrated that modern debugging tools are essential for the successful development and verification of complex digital systems.

Particular attention was given to the Vivado Integrated Logic Analyzer, which enables direct observation of internal FPGA signals during system operation. Unlike external measurement instruments, the ILA provides access to internal registers, communication buses, and state machine signals without requiring additional hardware resources. This

capability significantly improves the visibility of hardware processes and facilitates the identification of timing and synchronization issues.

Experimental studies performed on the Nexys 4 DDR board confirmed the effectiveness of the proposed approach. The use of the logic analyzer improved debugging efficiency, simplified error detection, and reduced development time. The captured timing diagrams provided detailed information about signal interactions and enabled verification of the correct operation of individual hardware modules.

The conducted investigation also demonstrated the educational value of integrated debugging tools. By visualizing internal FPGA activity in real time, students and engineers can gain a deeper understanding of digital design principles, finite-state machine operation, communication protocols, and timing behavior. Such practical experience contributes to more effective learning and improves the quality of FPGA development projects.

Another important result of this work is the confirmation that hardware debugging and simulation should be used together as complementary verification methods. While simulation provides insight into expected system behavior, the Integrated Logic Analyzer allows validation of the implemented design under real operating conditions. The combination of these approaches increases confidence in the correctness and reliability of FPGA-based systems.

Future work may focus on the analysis of high-speed communication interfaces, implementation of advanced verification techniques, and integration of hardware debugging tools into more complex FPGA-based systems. Additional research may investigate automated debugging methodologies, real-time performance analysis, and the application of machine-learning techniques for processing and interpreting captured waveform data.

REFERENCES

[1] Digilent Inc. Nexys 4 DDR FPGA Board Reference Manual.

[2] AMD Xilinx. Vivado Design Suite User Guide: Programming and Debugging.

[3] AMD Xilinx. Artix-7 FPGA Data Sheet.

[4] Pong P. Chu. FPGA Prototyping by Verilog Examples.

[5] Steve Kilts. Advanced FPGA Design.

[6] Brown S., Vranesic Z. Fundamentals of Digital Logic with Verilog Design.