Ứng dụng công nghệ FPGA xây dựng hệ thống điều khiển nhà thông minh

TÓM TẮT

Trong thời đại công nghệ 4.0, nhu cầu về các hệ thống nhà thông minh ngày càng tăng cao. Bài báo này trình bày việc ứng dụng công nghệ FPGA (Field Programmable Gate Array) để thiết kế và xây dựng một hệ thống điều khiển nhà thông minh có khả năng xử lý song song, tốc độ cao và linh hoạt trong cấu hình. Hệ thống được thiết kế nhằm điều khiển các thiết bị điện trong nhà và giám sát các thông số môi trường như nhiệt độ, độ ẩm thông qua các cảm biến và giao tiếp không dây. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng nhanh và có khả năng mở rộng cao.

Từ khóa: *FPGA, nhà thông minh, xử lý song song, giao tiếp không dây.*

Application of FPGA Technology for Developing a Smart Home Control System

ABSTRACT

In the era of Industry 4.0, the demand for smart home systems is increasing rapidly. This paper presents the application of FPGA (Field Programmable Gate Array) technology in the design and development of a smart home control system with parallel processing capability, high speed, and flexible configuration. The system is designed to control household electrical devices and monitor environmental parameters such as temperature and humidity through sensors and wireless communication. Experimental results show that the system operates stably, responds quickly, and offers high scalability.

Keywords: *FPGA, Smart Home, Parallel Processing, Wireless Communication.*

1. INTRODUCTION

In the era of the Internet of Things (IoT), smart homes have emerged as a prominent area of research and application, wherein electrical and electronic devices are interconnected and automatically managed to enhance the quality of life, optimize energy consumption, and ensure safety. The development of an effective smart home control system necessitates a hardware platform that offers high processing speed, low power consumption, and ease of customization. Within this context, Field Programmable Gate Array (FPGA) technology has gained significant attention as an ideal solution due to its parallel processing capabilities, low latency, and high flexibility in hardware design[1-4].

According to Magyari and Chen , FPGAs have been widely adopted in IoT network systems to optimize performance, security, and reliability, particularly in applications with stringent requirements on latency and power consumption[4]. Recent studies further demonstrate that FPGAs can replace CPUs and GPUs in numerous embedded applications due to their superior computational density and energy efficiency[5,6]. In the field of embedded systems, FPGAs have been employed to implement machine learning algorithms and deep neural networks (DNNs) on edge devices, thereby accelerating processing speed and reducing energy consumption. This advancement paves the way for smart home systems with enhanced learning and adaptive capabilities [7,8].

FPGA applications in smart homes are becoming increasingly diverse, ranging from lighting control and security monitoring to sensor data processing and artificial intelligence integration. The study by Guerrieri et al. demonstrated that FPGAs can be utilized to accelerate deep learning systems, process biomedical signals, and even perform fault monitoring in industrial equipment[5,6].

Table 1. Comparison of system processing time[9]

Sensor Type	FPGA (ms)	Microcontroller (ms)	Percentage of improvement (%)
Temperature	0.25	2.10	88.09
Humidity	0.30	2.25	86.67
Dust concentration	0.32	2.50	87.20
CO concentration	0.35	2.90	87.93

To evaluate the advantages of FPGA-based systems in real-time data processing applications, the authors in[9] employed an FPGA development board (Xilinx Spartan-6) and a microcontroller development board (ATmega328P), integrating temperature sensors, humidity sensors, dust concentration sensors, and carbon monoxide (CO) sensors.

Table 1 illustrates that: *(i)* the FPGA-based system demonstrates significantly faster processing speeds compared to the microcontroller; *(ii)* regardless of the sensor

type—temperature, humidity, dust concentration, or CO concentration—the processing time of the FPGA is consistently much lower than that of the microcontroller; and *(iii)* the performance improvement values converge around **87%**. These findings provide empirical evidence that FPGA systems exhibit superior efficiency and processing capability compared to traditional microcontrollers.

The remainder of this paper is organized as follows: Section 2 provides an overview of smart home systems and FPGA technology. Section 3 introduces the proposed system model. The implementation results are presented in Section 4. Finally, the conclusions of the paper are discussed in Section 5.

2. SMART HOME SYSTEMS AND FPGA TECHNOLOGY

2.1. Smart Home

A "smart home" is a modern type of residence organized to support human activities through automation and advanced technological devices. A smart home should be understood as a system that ensures safety, comfort, and resource efficiency for all users[10].

A smart home leverages both local storage and processing devices, such as gateways or hubs, in combination with cloud computing infrastructure[11-13]. With the advancement of edge computing, system performance is expected to improve significantly, as operations that do not require intensive computational resources can be executed locally. The benefits include reduced latency, load balancing, decreased communication traffic, and more flexible utilization of available resources.

The "Smart Home" system possesses both advantages and limitations that may play a crucial role in its deployment. Similar to other technological solutions, smart home systems offer a variety of benefits that make them worthy of implementation. These benefits include:

- Security: The system provides comprehensive control over the home and delivers notifications in the event of unauthorized access. In emergency situations, the Smart Home system seeks to prevent incidents, such as fire hazards.

- Ease of use: The entire system is managed through a single device, typically a mobile phone.

- Flexible configuration: The system enables users to customize device settings according to their needs and modify functionalities when required.

- Cost efficiency: A smart home reduces utility bills by automatically switching off devices when not in use. This not only lowers overall energy consumption but also decreases the load on the power grid.

- Automation: Most household appliances can be integrated into the smart home system, enabling automated control. This significantly reduces the time required for daily tasks.

- Design: All system components—including buttons, thermostats, sensors, outlets, and switches—are designed with a modern aesthetic that can easily blend into any interior environment.

However, the "Smart Home" system also presents several disadvantages, such as:

- Cost: Although the system primarily consists of basic sensors and cameras, the overall expense remains relatively high.

- System design challenges: A significant shortage of qualified experts in electronics, programming, and system design is considered a critical issue.

- Service and maintenance: Like any other device, the system may experience malfunctions. When this occurs, only highly skilled technicians are capable of resolving the issue, and such expertise is often difficult to find. Moreover, the failure of a single component can potentially affect the operation of other interconnected devices.

- Resource limitations: IoT devices have constrained resources, meaning they possess limited computational power, memory capacity, and energy efficiency. These restrictions can significantly reduce their ability to process data and interact with other devices.

- Security concerns: This issue arises from the fact that an increasing number of connected devices also increases the risk of cyberattacks and unauthorized access.

2.2. FPGA technology

2.2.1. Introduction

A Field-Programmable Gate Array (FPGA) is one of the key semiconductor technologies that enables hardware reconfiguration after manufacturing. With its parallel processing capability and high performance, FPGA has increasingly become a flexible computing platform for various domains such as telecommunications, IoT, artificial intelligence (AI), and cloud computing. Originating in the 1980s with the first commercial product introduced by Xilinx (now part of AMD), FPGA

differs fundamentally from CPUs and GPUs, which operate primarily in a sequential manner. Instead, FPGAs allow designers to construct customized logic structures at the hardware level, enabling low latency and high performance in specialized tasks.

2.2.2. FPGA Architecture

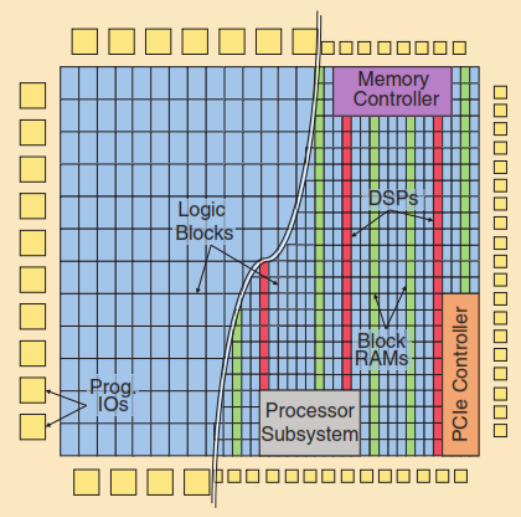

Figure 1. The original FPGA architecture with programmable logic and I/O compared to the modern heterogeneous FPGA architecture incorporating RAM, DSP, and other hard blocks. All blocks are interconnected through bit-level programmable routing.[14]

As illustrated in Figure 1, an FPGA consists of an array of various programmable blocks (logic, I/O, etc.) that can be flexibly interconnected through pre-fabricated routing channels, with programmable switches placed between them.

The functionality of all FPGA blocks and the configuration of routing switches are controlled by millions of static random-access memory (SRAM) cells, which are programmed at runtime to implement a specific function.

2.2.3. FPGA Programming Tools and Languages

FPGAs are programmed using Hardware Description Languages (HDLs), with VHDL and Verilog being the most widely used. The design flow typically involves several stages: hardware description, logic synthesis, mapping, routing, and bitstream generation for device configuration.

In recent years, High-Level Synthesis (HLS) tools have enabled FPGA design using C/C++ or OpenCL, thereby reducing development time and broadening accessibility for software programmers. Major vendors such as Xilinx (Vivado, Vitis), Intel (Quartus Prime, OneAPI), and Lattice (Radiant) have all provided their own toolchains to support the FPGA ecosystem.

3. SYSTEM MODEL

3.1. System Connection Block Diagram

In the scope of this paper, the authors develop a smart home control system based on FPGA technology. The core idea of this system is to collect environmental parameters from a DHT11 temperature and humidity sensor. Leveraging FPGA technology, these parameters are transmitted to a smartphone for monitoring the surrounding temperature and humidity, as well as for controlling the on/off operation of electrical appliances within the house. The proposed system is illustrated in Figure 2 and consists of the following fundamental blocks:

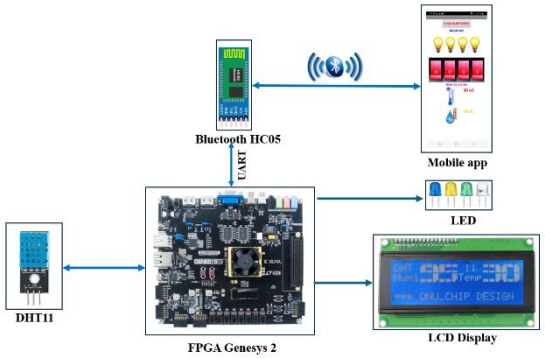

Figure 2. Block diagram of the FPGA-based smart home control system.

- *Sensor block:* This block employs the DHT11 sensor to measure environmental temperature and humidity. The DHT11 consists of a resistive humidity sensing element and an NTC thermistor, enabling simultaneous measurement of both parameters. It offers advantages such as fast response time, good noise immunity, and cost-effectiveness, making it suitable for IoT and smart home applications. The measurement range of the sensor covers humidity from 20% to 90% RH with an accuracy of ±5% RH, and temperature from 0 °C to 50 °C with an accuracy of ±2 °C.
- *FPGA central processing block:* This block serves as the core of the system, responsible for managing and coordinating all operations. The implementation is based on the Genesys 2 development board equipped with an AMD Kintex™-7 FPGA (XC7K325T-2FFG900C). The device integrates 50,950 logic slices, each comprising four 6-input LUTs and eight flip-flops, as well as 840 DSP slices, and supports an internal clock frequency of up to 450 MHz. The block is further organized into several internal submodules, including:

- DHT11_DATA block: This block is responsible for interfacing with the DHT11 sensor in order to acquire temperature and humidity values;
- LCD_DISPLAY block: This block handles the visualization of the acquired temperature and humidity values on an LCD screen;
- UART_CONTROLLER block: This block is designed to configure the baud rate and manage communication with the HC-05 Bluetooth module, enabling data transmission to the smartphone.

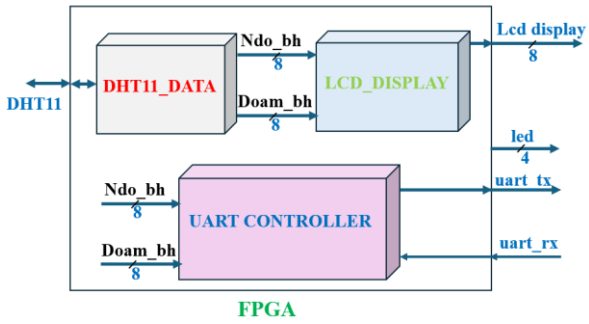

Figure 3. FPGA central processing block.

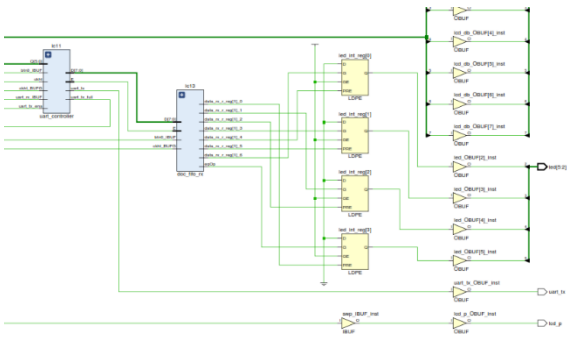

Figure 4. A partial architecture of the FPGA-based smart home control system.

- *Display block:* This block employs a 2004 LCD module to present temperature and humidity information acquired from the sensor block. At the same time, these parameters are also displayed on the smartphone application.

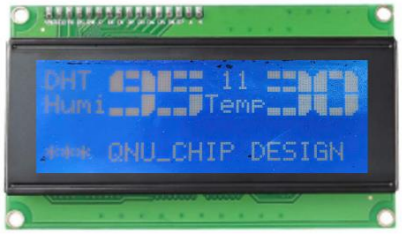

Figure 5. LCD 2004.

- *Bluetooth communication block:* This block utilizes the HC-05 module to transmit temperature and humidity data acquired from the sensor to the smartphone application. Simultaneously, it receives control commands from the application for switching electrical devices on or off, and forwards these commands to the FPGA central processing block.

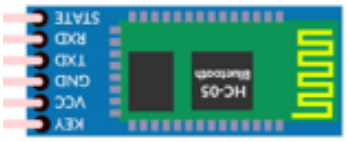

Figure 6. Module Bluetooth HC-05.

- *Smartphone application block:* This block, developed using App Inventor, is responsible for monitoring temperature and humidity data collected from the sensor, as well as controlling the on/off states of electrical devices.

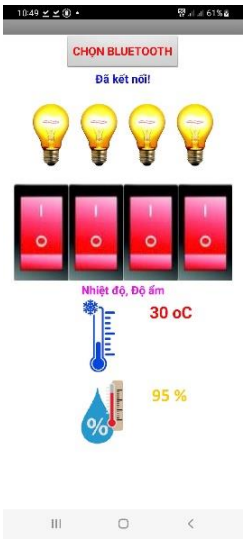

Figure 7. Smartphone Application.

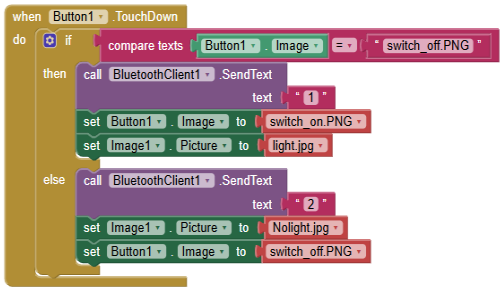

Figure 8. Programming with App Inventor.

3.2. Interfacing with the DHT11 Sensor

The DHT11 sensor employs a single-wire, two-way serial communication protocol (Single-Wire Two-Way). A complete communication process takes approximately 4 ms. Each full data transmission consists of 40 bits structured as follows: 8 bits for the integer part of relative humidity (RH), 8 bits for the decimal part of RH,

8 bits for the integer part of temperature (T), 8 bits for the decimal part of T, and 8 bits for the checksum. If the transmission is correct, the checksum value equals the sum of the 8-bit integer RH, 8-bit decimal RH, 8-bit integer T, and 8-bit decimal T, with only the least significant 8 bits retained.

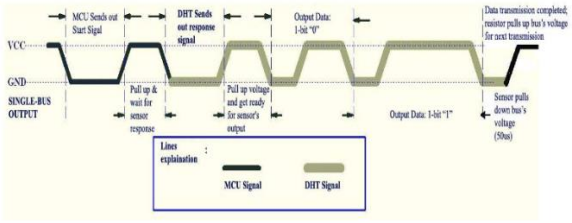

Figure 9. General Communication Process.

When the FPGA sends a Start Signal, the DHT11 sensor transitions from low-power mode to active mode, waiting for the FPGA to complete the start signal. Once the start signal is completed, the DHT11 transmits a 40-bit response containing temperature and humidity data to the FPGA. If no start signal is received from the FPGA, the DHT11 will not respond. After the data has been transmitted, the DHT11 returns to low-power mode until it receives the next start signal from the FPGA.

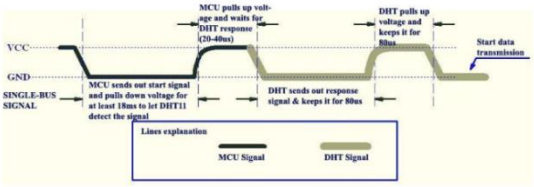

Figure 10. FPGA Start Signal Transmission and DHT11 Response.

The idle state of the single-wire data line is at a high voltage level. When communication between the FPGA and the DHT11 begins, the FPGA pulls the data line from high to low for at least 18 ms to ensure that the DHT11 can detect the start signal. Subsequently, the FPGA releases the line to high and waits for 20–40 µs to receive the response from the DHT11.

When the DHT11 detects the start signal, it sends a response signal at a low voltage level lasting 80 µs. Then, the internal program of the DHT11 sets the single-wire data line from low to high and maintains this high level for 80 µs to prepare for data transmission. When the single-bus DATA line is at a low voltage level, it indicates that the DHT11 is sending the response signal. Immediately after transmitting the response signal, the DHT11 pulls the line high for 80 µs to prepare for sending data. During data transmission to the FPGA, each data bit begins with a low voltage level lasting 50 µs, and the duration of the subsequent high voltage level determines whether the bit represents a “0” or a “1” (see Figures 11 and 12 below).

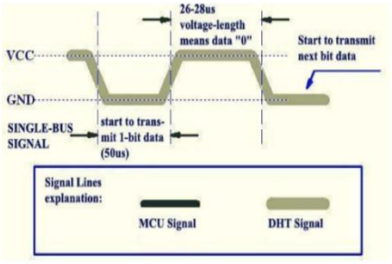

Figure 11. Signal Shape of Data “0”.

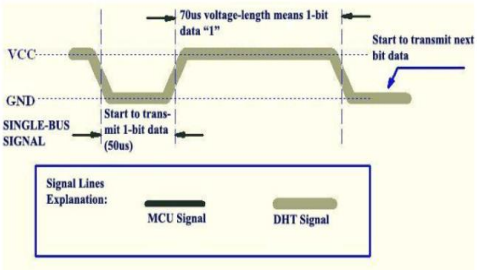

Figure 12. Signal Shape of Data “1”.

If the response signal from the DHT11 remains at a high voltage level, it indicates that the sensor is not responding correctly, and the connection should be checked. After the final data bit is transmitted, the DHT11 pulls the data line low and holds it for 50 µs. Subsequently, the single-bus line is pulled high by a resistor, returning it to the idle state.

4. RESULT

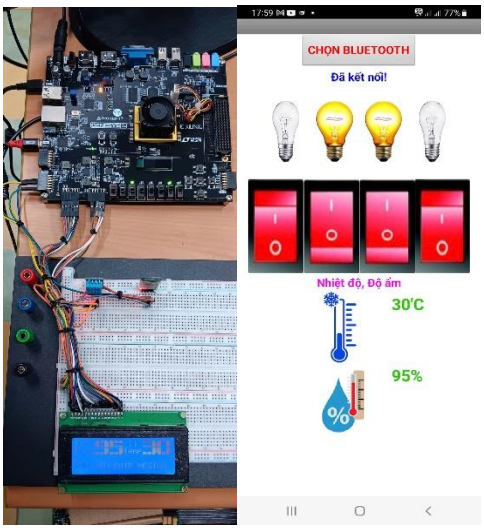

Figure 13. Experimental Model of the FPGA-Based Smart Home Control System.

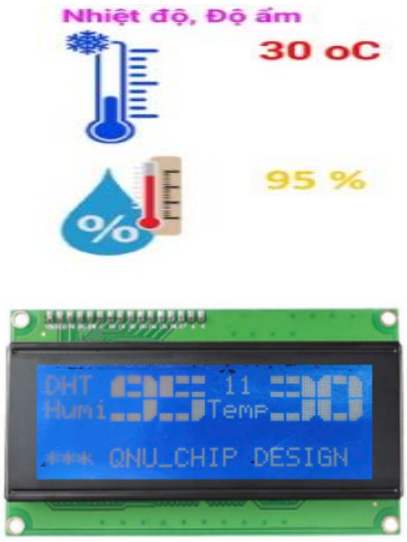

Figure 14. Results Displayed on LCD and Smartphone.

Figure 15. Results of Device On/Off Control.

Based on the system diagram illustrated in Figure 2, the authors implemented an experimental model of the FPGA-based smart home monitoring and control system, as shown in Figure 13. The results are presented in Figures 14 and 15. The measured temperature was 30 °C, and the humidity was 95%. These parameters were displayed on the LCD screen as well as on the smartphone application. Additionally, the states of the LEDs were controlled via wireless Bluetooth communication through the smartphone application.

5. CONCLUSION

Within the scope of this paper, the authors have investigated and developed a monitoring and control system for smart homes based on FPGA technology. With this system, users can monitor environmental parameters such as temperature and humidity, while simultaneously controlling the on/off states of electrical devices through a smartphone application. The results of this study further emphasize the significance of FPGA technology in handling high-speed, parallel-processing tasks with low latency, particularly in the context of mastering core technological capabilities.

Based on the results achieved in this study, the authors plan to further develop and integrate additional sensors and to build a more feature-rich system, particularly emphasizing IoT connectivity and AI capabilities.

REFERENCES

1. A. Magyari, Y. Chen. Review of State-of-the-Art FPGA Applications in IoT Networks, *Sensors*, **2022**, *19*, 7496.

2. D. G. Perera, K. F. Li. A design methodology for mobile and embedded applications on FPGA-based dynamic reconfigurable hardware, *Embedded Systems*, **2019**, *11* (5), 661–677.

3. S. Gandhare, B. Karthikeyan. *Survey on FPGA architecture and recent applications*, 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, Vellore, India, March 2019.

4. M. Joy Daniel, K. Siva Kumar M.E. Recent Trends and Improvisations in FPGA, *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, **2017**, *12* (3), 71-77.

5. A. Guerrieri, A. Upegui, L. Gantel. Applications Enabled by FPGA-Based Technology, *Electronics*, **2023,** *12*, 3302.

6. K. Guo, S. Zeng, J. Yu, Y. Wang, H. Yang. A Survey of FPGA- Based Neural Network Interference Accelerator, *ACM Transactions on Reconfigurable Technology and Systems*, **2017**, *9*(4), 11.

7. K. P. Seng, P. J. Lee, L. M. Ang. Embedded Intelligence on FPGA: Survey, Applications and Challenges, *Electronics*, **2021**, *10*, 895.

8. S. M. S. Trimberger. Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology, *IEEE Solid State Circuits Mag*, **2018**, *10*, 16–29.

9. Y. Zhu, K. Hou. Development and Implementation of an FPGA-Embedded Multimedia Remote Monitoring System for Information Technology Server Room Management, *Hindawi International Journal of Digital Multimedia Broadcasting*, **2024**, *2024*, 9426.

10. S. Poperehnyak, O. Bakaiev, Y. Shevchuk. *Construction of a Stable System of Interaction of IoT Devices in a Smart Home using a Generator of Pseudo-random Numbers*, Workshop on Cybersecurity

Providing in Information and Telecommunication Systems, CPITS 2025, Ukraine, 2025.

11. A. H. Alavi, P. Jiao, W. G. Buttlar, N. Lajnef. Internet of things-enabled smart cities: State-of-the-art and future trends, *Measurement*, **2018**, *129*, 589–606.

12. B. L. Stojkoska, K. V. Trivodaliev. A review of Internet of Things for smart home: Challenges and solutions, *Journal of Cleaner Production*, **2017**, *140*, 1454-1464.

13. S. Talari, M. Shafie-khah, P. Siano, V. Loia, A. Tommaetti, J. P. S. Catalao. A review of smart cities based on the Internet of Things concept, *Energies*, **2017**,*10*, 421.

14. A. Boutros, V. Betz. FPGA Architecture: Principles and Progression, *IEEE Circuits and Systems Magazine*, **2021**, *21*, 4-29.