

FPGA Experimental Teaching Based on Engineering Development Process

Xiulei Zhang¹, Jin Xiao^{1,*} and Haosong Yue¹

¹ *School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*

^{*} *Corresponding author. Email: xiaojin@buaa.edu.cn*

ABSTRACT

The conventional Field Programmable Gate Array (FPGA) experimental teaching failed to give sufficient attention to the integration with engineering application. As a result, students cannot form the FPGA engineering development and application concept, and cannot understand the FPGA engineering development process, which will hinder the improvement of students' engineering practice ability. Based on the existing problems in FPGA experimental teaching, this work explored and studied the use of SignalTap II logic analyzer, the application of instruments and equipment, the development of engineering basic example modules, etc., and proposed a FPGA experimental teaching mode based on engineering development process, which is expected to benefit students' FPGA engineering development ability cultivation.

Keywords: *FPGA experimental teaching; engineering development process; SignalTap II logic analyzer;*

engineering basic example

1. INTRODUCTION

FPGA is the abbreviation of field-programmable gate array, which is the product of further development after programmable devices such as PAL, GAL, CPLD, etc. It is a semi-custom circuit in the field of application-specific integrated circuit (ASIC), which not only solves the shortcomings of customized circuit, but also overcomes the limited number of original programmable devices and gates [1]. In the field of high-speed data acquisition, FPGA is superior to MCU or DSP to realize the sequential logic function of data acquisition such as video Decoder / Encoder. As for logic interface, FPGA can realize rich digital interface logic such as PCI, PCIe, USB, SDRAM communication interface, etc. With regard to level interface, it covers new level standards such as LVDS, HSTL, GTL / GTL+, SSTL, etc., in addition to LVTTTL and LVCOMS, which meet the interface level requirements of most electronic products.

FPGA plays a vital role in the development of military and civil products, and its application is increasingly extensive. Many universities have set up FPGA experiment courses to cultivate students' engineering practice ability. However, the traditional FPGA experimental teaching focuses on functional verification, and the experimental content does not pay attention to the combination with practical engineering application. As a result, the students do not master the FPGA design concept, and they are also ignorant for FPGA design process [2]. It is particularly important that students don't form the concept of FPGA engineering development and application, and don't know how to link the designed module with engineering application. Also, they don't understand the FPGA engineering development process and the commonly used

FPGA programming skills in engineering applications, which is not conducive to improving students' engineering practice ability. This work put forward FPGA experimental teaching mode based on the engineering development process, and made some exploration and research on it, hoping to draw lessons from others.

2. TEACHING STATUS AND EXISTING ISSUES

Many experimental courses offered by the school's electrical and electronic center contain FPGA experimental contents, and have been running independently for many years. Through 8 to 16 class hours of classroom experiments, students are familiar with the development environment QuartusII, learn the programming language VHDL, and design digital logic circuits combined with FPGA experimental box, including basic gate circuits, combinational circuits, sequential circuits and integrated application circuits. Due to the limitation of class hours, students need to master the development environment and programming language after class, complete the program design, and carry out function debugging and verification in class. After many years, the implementation of this teaching mode gradually presents some problems, which are summarized as follows.

2.1. FPGA program debugging method is single

FPGA program debugging method is single, which is quite different from the debugging process in practical

engineering, thus limiting the way and method of solving problems for students. At present, students only use the function simulation software to debug the FPGA program, and think that the program is designed successfully after the simulation is passed. In fact, functional simulation (RTL behavior level simulation) is only used to check the syntax errors in the code and the correctness of the code behavior. It does not consider competitive risk and gate circuit delay information, which has a certain degree of ideality and can't really reflect the reality [3]. In many cases, the simulation passed. However, there will be various errors when the actual hardware circuit runs. This excessive reliance on idealized simulation limits students' thinking, isolates the truth of the fact, prolongs the debugging time, and stretches the distance of the actual project debugging. It is urgent to introduce other efficient debugging tools or methods to guide students to find the root of the problem.

2.2. FPGA experiment content with high capacity and low quality

At present, FPGA experiment content is designed based on different difficulties such as simplicity, generality, synthesis, improvement, etc. In the process of experiment, students tend to pay attention to the number of experiments and do not understand the problems, and the final result is not satisfactory. Taking the basic and key signal processing problems often encountered in practical engineering as an example, students' experimental results are tested in the communication with students after class. Some students answered "I'm not clear", and some answered "Simulation is OK, and the actual circuit results should not be wrong", etc. In a word, no matter how many questions are done without grasping the essence, just repeat, the actual engineering development ability is improved little. Instead of this, it is better to focus on making students understand and grasp the basic and key processing ideas and methods in practical engineering, such as signal anti-shake processing, clock frequency division or doubling, signal synchronous cross-linking, edge detection, signal establishment or holding time, etc. Based on this, the function module design close to engineering application is realized, which is small in quantity but high in quality. It not only increases students' engineering application experience, but also improves students' ability to solve engineering problems.

2.3. FPGA experiment is disconnected from the hardware circuit

As we all know, FPGA program developers belong to the hardware engineer series, and need to have a certain hardware foundation. Before the FPGA experiment, students should have some understanding for experimental verification carrier (FPGA experimental box), and thoroughly understand the principle and signal input-output relationship of the hardware circuit part used. Thus, the clear circuit structure, signal transmission line and signal timing relationship in their minds are helpful for fault analysis and troubleshooting when students encounter the program running fault. It is found in the current teaching that most students lack the understanding of experimental hardware circuit. When they encounter hardware failure, they will not analyze any module circuit, and then there is no engineering consciousness to assist in locating and locating faults. The hardware circuit is not clear, and the connection between the signals is not clear. The meaning and location of the basic signal terminals and the reference terminals are not clear. Even if the students have the consciousness of using the equipment to troubleshoot, they will feel confused and at a loss about how the equipment tests and what variables are detected. Therefore, teachers should do a good job in corresponding explanation, demonstration, teaching, guidance, etc., with regard to students' understanding, familiarity and mastery of hardware circuits, and students' understanding of using relevant instruments and equipment to detect hardware circuits. Students are served to experience and learn the actual engineering development process, so that students can achieve the purpose of engineering development training.

3. EXPLORATION AND RESEARCH ON NOVEL TEACHING MODE

Based on the change of FPGA programming language, the introduction of SignalTap II logic analyzer and the addition of external instrument application, the basic module example based on engineering application is designed. Students' basic FPGA application ability and mastering basic digital signal processing technology are exercised, and then some comprehensive experimental content is extracted from the actual project to improve the students' ability of function expansion, module integration and overall joint debugging. Finally, the FPGA teaching content in the experimental course should be optimized and improved, and a new experimental teaching mode with strong operability, continuity and expansibility should be formed. The structure diagram of FPGA experiment teaching based on engineering development flow is shown in Fig. 1.

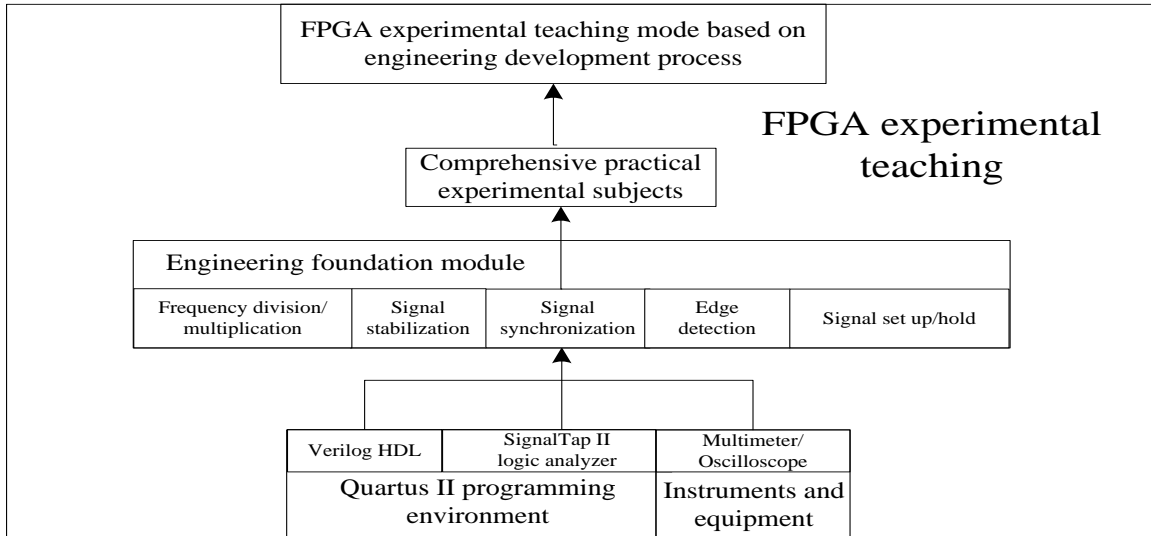


Figure 1. Structure diagram of FPGA experimental teaching mode based on engineering development flow

3.1. Increasing SignalTap II logic analyzer tools

With the increase of FPGA capacity, the design of FPGA becomes increasingly complex, and the design and debugging become a very heavy task. In order to put the design into the market as soon as possible, designers need a simple and effective testing tool to shorten the debugging time [4]. In the test of complex FPGA design, the traditional logic analyzer will face the following problems. First, there is a lack of spare I/O pins. The selection of device in the design is based on the design scale. Generally, the number of I/O pins for selected devices exactly matches the design requirements. Second, I/O pins are difficult to lead out. In order to reduce the area of PCB, designers mostly adopt the technology of thin spacing. It is very difficult to lead out the I/O pin without changing the PCB wiring. Third, the external logic analyzer may change

the original state of the signal in FPGA design, and thus it is difficult to ensure the integrity of the signal. Fourth, the traditional logic analyzer is expensive, which will increase the economic burden of the designer.

With the rapid development of EDA tools, the SignalTap II logic analyzer in Quartus II, a new debugging tool, meets the requirements of hardware debugging in FPGA development. It is characterized with no interference, easy to upgrade, simple to use, low price, etc. The SignalTap II logic analyzer is integrated into the Quartus II software, which can capture and display the real-time signal state in the design of system on a programmable chip (SOPC). Thus, developers can observe the interaction between hardware and software at the system level in the whole design process [5]. The SignalTap II logic analyzer interface is shown in Fig. 2. It offers designers with real-time visibility of the industry-leading SOPC design, which can greatly reduce the time spent in the verification process.

Type	Alias	Name	27	Value	28	-128	-112	-96	-80	-64	-48
		ADS_		1							
		⊕ A		37FFFFFFh							
		BLAST_		1							
		...R_		1							
		⊕ Q		0000h							
		READY_		1							
		⊕ fifo_bmk:U8 data		0001h						1008h	
		⊕ fifo_bmk:U8 q		0000h							
		⊕ fifo_bmk:U8 wrusedw		008h						00Ah	
		⊕ fifo_bmk:U8 rdusedw		008h						00Ah	
		⊕ ...ceive_LVDS_bmk:U4 bmkaddr		00h						08h	
		⊕ ...ceive_LVDS_bmk:U4 bmkdata		0000h						1008h	

Figure 2. SignalTap II logic analyser interface

SignalTap II logic analysis module is embedded into the FPGA. The logic analysis module captures the tested signal, and the data is transmitted from FPGA to Quartus II software for display through JTAG interface. SignalTap II eliminates the need for additional logical analysis equipment and simply connects a JTAG interface download cable to the FPGA device to be debuggable [6]. The SignalTap II logic analyzer replaces the external special instrument to capture signals inside the device to analyze and judge system faults, which offers a good way for FPGA program debugging. In the process of FPGA experiment, students can not only rely on the simulation software for functional verification, and the simulation results can not completely replace the actual operation results. By using the SignalTap II logic analyzer, the logic timing relation of the signal in the actual hardware circuit is captured. This is convenient to find and locate the cause of hardware circuit fault, and is very helpful to improve students' ability to solve practical engineering problems.

3.2. Increasing basic routine design for engineering applications

By setting up FPGA experimental courses, students can understand and master the basic knowledge of programming environment, programming language, programming architecture, etc., required for FPGA development. More importantly, students can be close to the actual engineering development and application. Taking engineering application as the starting point and FPGA programming skills as the support, this part focuses on the design of basic module examples based on engineering application. It can effectively enrich students' engineering experience, cultivate students' engineering awareness, and lay the engineering foundation for students to engage in FPGA development in the future. The basic routines of design and development are as follows.

3.2.1. Signal anti-shake processing module

In practical engineering, the digital pulse generated by relay close/open, key/button, dial switch, etc., is easy to generate jitter in the process of the jump. If the special anti-shake circuit is not added, the programmer needs to process the signal jitter through the program, thus avoiding the timing confusion caused by the signal jitter. Digital signal anti-shake processing is frequently used in practical engineering projects, and FPGA flexible programming method can design various anti-shake processing methods to adapt to different application scenarios.

3.2.2. Clock frequency division/doubling module

Clock is a necessary signal of sequential logic circuit. A system, especially an asynchronous system, needs various frequency clock signals. FPGA programming to achieve a

variety of frequency clock signals is the basic skills for engineering developers. Students are trained to use counter or IP core to design clock frequency division and frequency multiplication module, and master the basic skills of this project flexibly.

3.2.3. Signal synchronization processing module

The complex control logic based on FPGA may contain many asynchronous modules, and the clock domain of each module is different. The synchronization problem needs to be considered in signal interaction between modules. Otherwise, the timing confusion will be caused by the lack of signal establishment / holding time or the untimely signal state capture. Asynchronous system is one of the difficulties in FPGA development. It is an important skill to understand the interaction mechanism of asynchronous modules and master the processing method of signal interaction between asynchronous modules to solve the timing problem in engineering.

3.2.4. Signal edge detection module

In the digital system, the problem of signal edge detection will be involved when the signal edge is needed to trigger the corresponding action. For students, safe and reliable detection signal edge can not only develop their divergent thinking ability, but also cultivate their engineering application ability.

3.2.5. Signal establishment/holding time processing module

In the integrated circuit, most of the communication links between modules are made up of a group of buses, and the data interaction between modules is completed by the specified communication protocol. In the communication protocol, the time sequence relationship between signal and signal is defined. In the time sequence relationship, the signal establishment and holding time are two very important indexes. For example, it is required that the signal be stable at t time before the clock rising edge comes when the clock rising edge captures the signal. If the time is less than t , the receiver has unstable data reception when the clock rising edge comes. The retention time is also similar. Currently, integrated chips or modules in the market have their own establishment / holding time requirements. Users must meet their requirements before they can complete their interactive communication. Students can understand and understand the meaning of establishing/holding time by designing examples. Then through a simple self-checking communication example, students can master the specific operation process of the signal establishment/holding time, laying the foundation for the subsequent complex functional module integration.

3.3. Increasing hardware circuit detection

FPGA program design and debugging process is the design and debugging process for hardware circuit. Even with the powerful debugging tool of SignalTap II logic analyzer, traditional multimeter, oscilloscope, etc., are also indispensable equipment for hardware developers [7]. There are two main reasons.

First, the use of SignalTap II logic analyzer is conditional, such as triggering clock. If the clock in the circuit fails, SignalTap II will fail, and external test instruments and equipment are required to assist in troubleshooting. On the other hand, even if SignalTap II can capture the signal change state, the debugger will actually use external equipment for inspection and confirmation in order to troubleshoot reliably.

Second, hardware circuits with certain functions cannot be controlled by FPGA completely. There will always be some circuits that work independently of FPGA. In order to debug FPGA programs, developers need to control this part of circuits. At this time, multimeter, oscilloscope and other test instruments will be used in the field.

In brief, students can have a certain understanding to engineering development and debugging process through this link of learning. Combined with the simple use of instruments and hands-on practice process, they can achieve better results.

4. CONCLUSION

FPGA experimental teaching mode based on engineering development process is not only helpful for students to consolidate and deepen theoretical knowledge learning, but also offers students with training opportunities close to the FPGA engineering development process in society. It is necessary to learn to use Verilog HDL programming language in line with the market trend, design and verify examples of FPGA programming skills related to digital signal processing, and use flexible and convenient SignalTap II logic analyzer tools. Combined with the measuring instruments and equipment of multimeter, oscilloscope, signal generator, etc., it seems to be a technician who develops and debugs on the job using FPGA. It is helpful to broaden students' vision and thinking mode, enhance students' self-confidence and

competitiveness, and lay a solid foundation for their future career or postgraduate study after graduation.

ACKNOWLEDGMENT

The author is grateful for the 2019-2020 teaching reform project granted by Beihang University: Research and Practice of FPGA Experimental Teaching Model Based on Engineering Development Process.

REFERENCES

- [1] Liu Tao, Lou Xinghua. Example Navigation of FPGA Digital Electronic System Design and Development [M]. Beijing: People's Posts and Telecommunications Press, 2005: 1-2.
- [2] Qin Hongying. Discussion on the Improvement of FPGA Experimental Teaching Method [J]. Laboratory Science, 2012, 15(4): 79-82.
- [3] Wang Gesi, Liu Mian, Yi Zongqi. Design and Application of FPGA Innovative Development Experiment Teaching Platform [J]. China Electric Power Education, 2009(10): 145-146.
- [4] Guo Jiajia, Hu Xiaojing, Wang Yongliang. Using SignalTap II Logic Analyzer to Debug FPGA [J]. Today Electronics, 2005(5): 45-47.
- [5] Deng Cheng, Zhang Yani, Bai Lin, et al. Application of Embedded Logic Analyzer in FPGA Design [J]. Modern Electronic Technology, 2006(2): 76-84.
- [6] Wu Zhaohui, Zhang Zhijian. Application of Embedded Logic Analyzer in FPGA Design [J]. Hebei Industrial Science and Technology, 2010, 27(1): 41-46.
- [7] Lu Youliang, Jiang Shuyan. Challenging Experimental Teaching Design of Digital Design FPGA Application Courses [J]. Experimental Science and Technology, 2018, 16(6): 82-85.