Electronic Engineering and Informatics
G. Izat Rashed (Ed.)

doi:10.3233/ATDE240075

FPGA Chip-Based Multi-Module System Verification Methodology

Haojie CHEN and Shikai ZUO[1]
School of Optical and Communication Engineering, Xiamen University of Technology, Xiamen, China

Abstract. This paper proposes a functional verification method for multi-module system field programmable gate array (FPGA) chips. According to the FPGA integrated netlist information, multiple modules to be tested as well as the connection relationship between modules to be tested and the system are all extracted accurately to get a brand-new netlist of modules to be tested. Then the verification excitation is inputted from the outside to verify the extracted netlist, which reduces the compilation time of the large-scale FPGA chip, as well as reduces the configuration time of the chip function verification, accelerates the speed of the simulation, and improves the simulation and verification efficiency. This method has been successfully applied to the FPGA chip circuit function verification engineering practice.

Keywords. FPGA chip verification; multi-module system; netlisting

1. Introduction

Multi-module systems are widely used in FPGA design because they enable a complete system to be implemented quickly and efficiently. For example, such systems can be used to implement embedded systems, programmable logic controllers (PLCs), and digital signal processing (DSP) systems. At the same time, multi-module systems are highly scalable, and modules can be easily added or removed to fulfill the Adequate for specific needs.

Table 1 lists the frequently used modules and their functions in a particular FPGA chip:

Table 1. Modules and Functions

Module (in software)	Functionality
Configurable Function Unit (CFU)	Performing specific computational and logical operations. Basic lookup tables, arithmetic logic Four operating modes: cell, static random memory, and read-only memory
Input-Output Module (IOB)	Receiving signals from external sources via I/O ports
Block Static Random Memory Module (BSRAM)	Providing fast read/write speeds and efficient data storage

[1] Corresponding author: Shikai ZUO, School of Optical and Communication Engineering, Xiamen University of Technology, e-mail: 408596563@qq.com

Module (in software)	Functionality
Digital Signal Processing Module (DSP)	Processing and analyzing digital signals
MIPI D-PHY	A digital serial interface protocol for transferring high-speed data within or between chips

In the multi-module system FPGA chip verification, the verification focus and difficulties are mainly concentrated on the intricate connection relationship between multiple modules and between each module and the system, as well as the correctness of the configuration process and global signal wiring [9]. In order to improve the verification coverage, the verification object needs to contain a large number of verification modules, consuming a long time of verification. And each module verification contains the connection relationship in the whole chip circuit netlist, resulting in the verification of the signal connection unrelated to the module accounting for about 2/3 of the whole verification time [10]. Therefore, to accelerate the simulation speed and improve the verification efficiency, this paper proposes a multi-module system FPGA chip function verification method.

2. FPGA Functional Simulation Verification Scheme for Conventional Multi-Module Systems

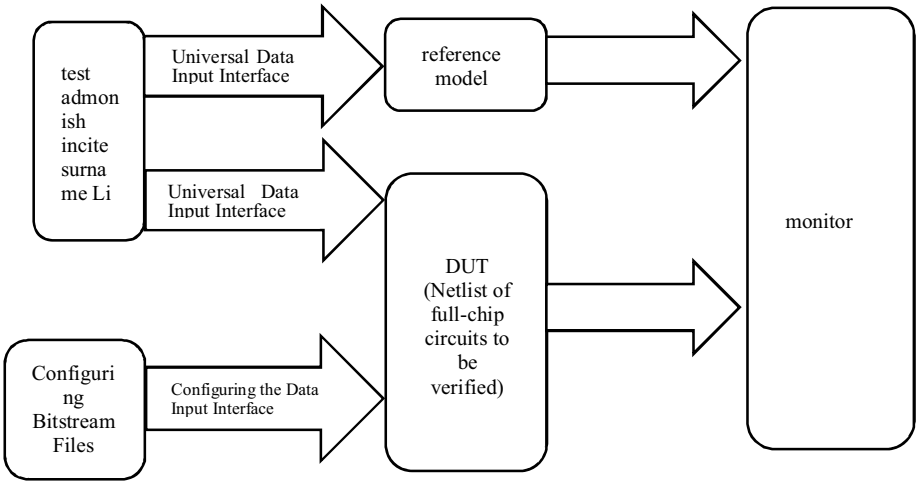

Figure 1. Full chip verification platform

The full-chip verification platform is shown in Figure 1, which mainly includes verification excitation, configuration bitstream file, circuit under test (DUT), reference model, and monitor. The DUT is the full-chip circuit netlist to be verified, the configuration bitstream file is input to the DUT through the configuration data input interface of the DUT, and the corresponding verification excitation is input to the DUT through the general data input interface of the DUT. At the same time, the corresponding verification excitation is input into the DUT through the generic data input interface of the DUT, while the verification excitation is input into the reference model through the

generic data input interface of the DUT, and the monitor is used to monitor whether the output of the DUT is consistent with the expected value of the reference model [1], [2].

In traditional multi-module system verification, the DUT is a full-chip circuit netlist that encompasses the entire circuit module. When more than one module to be verified in this netlist is verified, the other modules of the whole circuit will be involved in the verification project together, and the multi-module system verification only accounts for a small part of the whole circuit module. So this will share most of the unnecessary simulation and verification resources, resulting in prolonged simulation and verification time, reduced coverage, and low verification efficiency. With the number of FPGA chip integration resources and the complexity of the design increasing day by day, the disadvantages of using this method to verify the multi-module system are becoming more and more obvious [3].

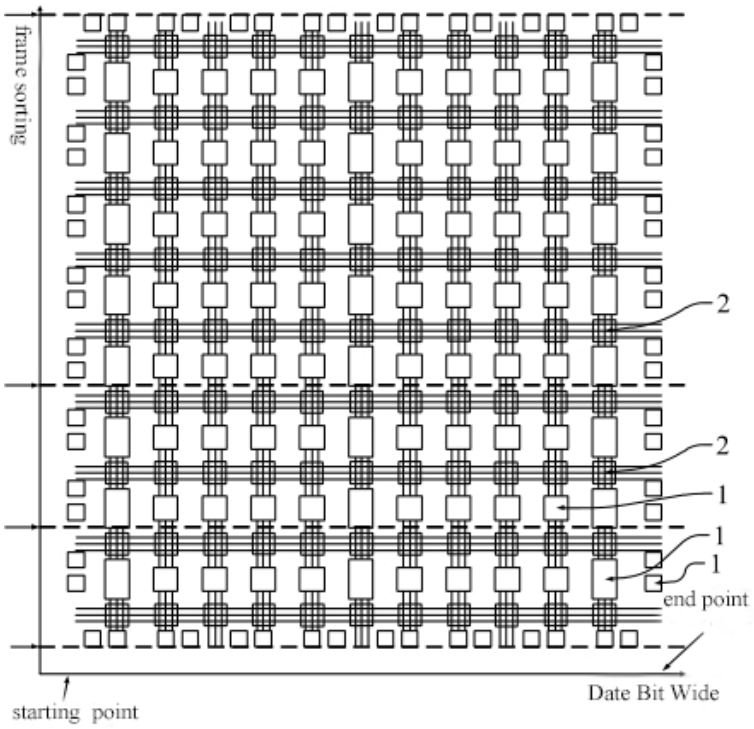

Figure 2. Logic resource arrangement structure in FPGA chip and its frame distribution

In addition, as shown in Figure 3, the configuration bitstream file also limits the efficiency of the multi-module system simulation and verification. The logic resources in the FPGA chip and its frame distribution are shown in Figure 2, which mainly includes VERSATILE and interconnect resources [4]. The first frame of the configuration bit stream corresponds to the first line of circuits at the top of the chip, and the last frame corresponds to the last line of circuits at the bottom of the chip, and it takes a long time to download all the configuration bit streams from the first frame to the last frame. If the verification case contains more chip resources, the simulation and verification time will be longer [5], [6].

3. Improved Functional Simulation Verification Method for FPGA Chips in Multi-Module Systems

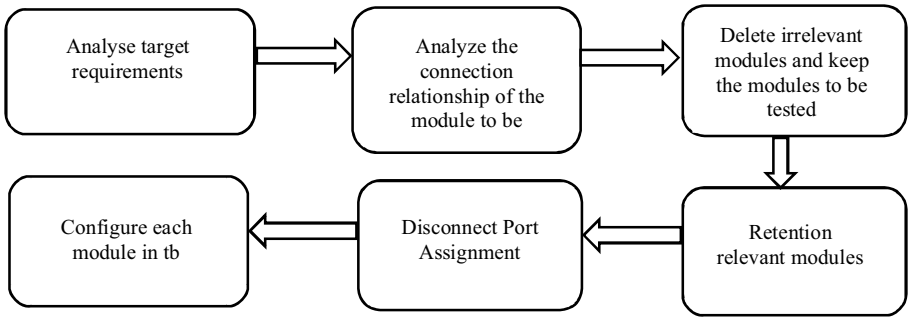

Figure 3. Flowchart of FPGA chip function simulation verification for multi-module system

Multi-module system FPGA chip function simulation verification specific flow is shown in Figure 3. A complete netlist of FPGA is input, and the target demand to be tested is analyzed. According to the demand to analyze the need to cover the module, the development of a set of special tools is used to automatically extract the sub-modules from the netlist, the connection relationship is analyzed, the connection relationship of the sub-modules to be retained is extracted, and the other irrelevant modules in the netlist will be deleted [7]. According to the netlist of the extracted sub-modules, the analyzer will analyze the default state of the corresponding input/output connections, and assign corresponding signals to the disconnected ports. The transmission of the configuration bitstream file to DUT is canceled, the configuration code stream used by the sub-module to be verified in the configuration bitstream file is analyzed and extracted, and backdoor access is used to set the code stream configuration data of the corresponding sub-module directly to the corresponding sub-module, and the function of the sub-module is turned on [8]. After all the sub-modules are configured sequentially, the simulation system is configured in tb, and the re-testing of each sub-module is carried out sequentially.

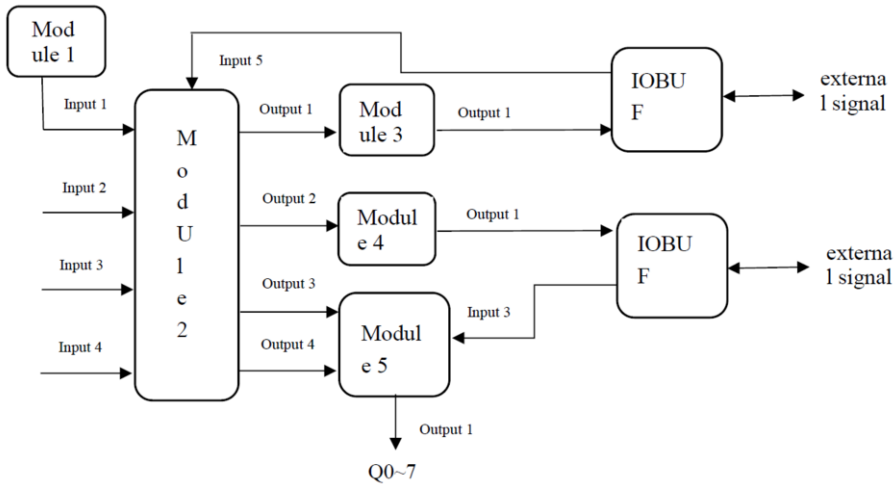

Figure 4. Connection relationship of each sub-module after an FPGA netlist cropping

Taking a certain FPGA chip as an example, when multiple sub-modules are verified, the DUT in the verification platform is a full-chip circuit netlist, which means that when verification is performed, all modules in the netlist are involved in the verification. To improve the verification efficiency, the main purpose is to use a special tool to extract the sub-module to be tested from the netlist, and all the remaining modules are deleted. In the netlist, the sub-module to be tested has an intricate connection with other sub-modules, and when separating the sub-module to be tested from other sub-modules, it is necessary to analyze the connection relationship between the two and clarify the signal with high and low levels. The input and output ports of the sub-module to be tested are confirmed, and each module has a lot of input and output relationships with other modules. So the inputs and outputs used by the sub-module to be tested will be retained, and the rest of the sub-module to be tested will be deleted. At this time, the reserved input and output ports are disconnected and suspended. And according to the signal interaction between the sub-module to be tested and other sub-modules in the network table, a fixed value is assigned to the disconnected input port, so that even if the input is disconnected, it can still work normally as it did before cutting the network table.

As shown in Figure 4, Module 1, Module 2, Module 3, Module 4, Module 5, and the IOB module originally had a large number of connections to each other, but after removing unnecessary connections, a small number of connections were retained as shown in the figure. Input 1 of Module 2 is connected to the output of Module 2, Input 5 is connected to the output of the IOBUF module, and the rest of the input ports are disconnected and suspended. The suspended input ports of Input 2, Input 3, and Input 4 are assigned corresponding signals based on the information of the netlist. Output 1 of Module 2 is connected to the input of Module 3, Output 2 is connected to the input of another Module 4, the other two output ports of Module 2, Output 3, and Output 4 are connected to the input of Module 5, and the other input of Module 5 is connected to the output of the IOBUF module, which outputs the Q0~7 data from Output 1. The output of Module 3 is connected to one of the IOBUF module inputs, and the output of Module 4 is connected to another IOBUF module input. Also, it is worth noting that the cib serves as the internal base unit of the FPGA chip. For the cib inside the sub-module to be tested, it is also necessary to disconnect the redundant connections, keep the required ones, and assign the disconnected inputs to fixed values. The original configuration bitstream file is canceled, and the configuration bitstream file for the configuration code stream used by the sub-module to be verified is analyzed and extracted through the script to automatically assign the corresponding configuration code stream data to the module.

4. Simulation verification results

System Verilog, an object-oriented verification language, is used to establish the Verilog functional verification environment. The simulation tool of Synopsys VCS is used to design the simulation verification [10]. Through the Verdi waveform viewer tool, the signal timing waveforms between the various sub-modules retained can be visualized, as shown in Figure 5.

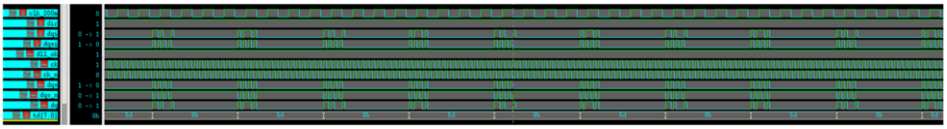

Figure 5. Signal timing waveform

By comparing the DUT of traditional multi-module system verification and the DUT file size of the improved multi-module system verification, as shown in Figure 6 and Figure 7, it can be seen that the cropped DUT can effectively reduce the occupation of redundant sub-modules in terms of verification resources, shorten the time of simulation and verification, and improve the verification efficiency.

Figure 6. Comparison of file size between full chip circuit netlist and cropped netlist file size

```
$finish called from file "../src/tb.sv", line 79.
$finish at simulation time              50000000
           V C S   S i m u l a t i o n   R e p o r t
Time: 50000000 ps
CPU Time:    271.590 seconds;       Data structure size:  28.5Mb
Fri Jul 28 17:38:26 2023
CPU time: 5.871 seconds to compile + 2.683 seconds to elab + .271 seconds to link + 271.644 seconds in simulation

$finish called from file "fc_tb.pat", line 109.
41360000:fc_tb: ---- SIMULATION PASSED ----
$finish at simulation time              41360200
           V C S   S i m u l a t i o n   R e p o r t
Time: 41360200 ps
CPU Time:    358.430 seconds;       Data structure size: 17836.4Mb
Mon Jul 31 15:47:59 2023
CPU time: 303.031 seconds to compile + 974.848 seconds to elab + .529 seconds to link + 359.299 seconds in simulation
```

Figure 7. Comparison of running time between traditional and improved methods

5. Conclusions

In this paper, the FPGA chip integration verification method for multi-module systems is investigated and the method of removing redundant sub-modules and retaining the sub-modules to be tested is proposed, which can only allow the to-be-tested submodules to carry out the verification. Thereby, this verification method can only allow the sub-module to be tested to reduce the occupation of verification resources by redundant sub-modules. The code streams required by the sub-module to be tested in the configuration bitstream file are directly assigned to the sub-module to be tested in the DUT to reduce the time of downloading and configuring the bitstream data, thus greatly reducing the simulation time, improving the verification efficiency, and satisfying the needs of large-scale circuit function simulation.

References

[1] ACCELLERA. Universal verification methodology (UVM) 1.1 class reference [Z/OL]. (2011-06-30)[2022-8-17] https://www.docin.com/p-226774464.html.
[2] JIA Xun, QIAN Lei, WU Guiming, et al. Research status and future challenges of FPGA application to high-performance computing[J]. Computer Science, 2019, 11(46): 11-19.
[3] Xueya Zhuang, Leilei Li. A configuration chip design based on SRAM-based FPGA [J]. Electronics and Packaging, 2018, 18(3): 29-32.

[4] SAMANTA P, CHAUHAN D, DEB S, et al. UVM-based STBUS verification IP for verifying SoC architectures [C]//18th International Symposium on VLSI Design and Test. ie, 2014: 14-16.
[5] Cao Yang, Hu Yue Li. Functional verification of storage controller based on UVM [J]. Computer Measurement and Control, 2015, 23(3): 834-837.
[6] Lou Xiuli, Cong Hongyan, Yan Hua, et al. An efficient FPGA integration verification method: CN112597721A[P]. (2021-07-05)[2022-03-23].
[7] Sun T. Research on IP soft core verification methodology [D]. Beijing: Beijing Jiaotong University, 2009.
[8] Cong Hongyan, Yu Zongguang, Yan Hua, et al. An efficient and high-speed function verification method for high-capacity FPGA circuits [J]. Semiconductor Technology, 2017,42 (1):25-27.
[9] Research on Modeling and Verification Technology of Functional Model Based on SystemC. He Mingyue; Peng Desheng. Electronic Quality, 2016
[10] Universal Verification Methodology (UVM)1.1 Class Reference.Candence Design Systems Inc.2011.