Design Method of Build-in Test Equipment Interface in FPGA System Based on JTAG

DENG Haitao¹, HE Juan², WANG Nantian³, HU Jiafei⁴
¹National University of Defense Technology, Changsha 410073, Hunan, China.
²61287 Army, Chengdu 610000, Sichuan, China.
³Hypervelocity Aerodynamics Institute, CARDC, Mianyang 621000, Sichuan, China.
⁴National University of Defense Technology, Changsha 410073, Hunan, China.
Email: denght.88@163.com

Abstract. Data transmission in typical Build-in Test Equipment (BITE) with FPGA system is based on FPGA’s input and/or output (I/O) pins. It occupies some IO resource, and cannot be used in having been designed system whose hardware change is not allowed. FPGA system usually has a JTAG (Joint Test Action Group) interface, which is not used when the system working. Therefore, we present a BITE data transmission interface design method using the idle JTAG interface in FPGA system. Principle and design method is introduced in detail, and a design demo is given. Experiment based on the design is conducted; results verify the effectiveness of the method.

1. Introduction
In recent years, with the performance growth of systems, their complexities also have increased a lot. And the Build-in Test (BIT) has been widely used for fault diagnosis and isolation in different layers such as PCB board layer or subsystem layer in them. Usually, test equipment is embedded in the tested object in design, which is called Build-in Test Equipment (BITE).

In electronic systems, FPGAs are widely used. How to design BIT for FPGA system comes into our view. A simplified general method is designing the test logics in the FPGA and sending out the test results using some input and/or output (I/O) pins. However, the simplified general method will cost some I/O resource. Furthermore, hardware change are not allowed in some in-using FPGA systems. The question is that can we design BIT for them? Usually, an FPGA design has some programmable logic resources in redundancy, and this could be used to design the test logics for sending out the test results. The FPGA system has a JTAG interface used for system programming and debugging, but it is not occupied with the system working. So it is a solution that using this JTAG interface to transmit the BIT data. Based on this point, we present our BITE data transmission interface design method using the idle JTAG interface of FPGA system.

2. Principle of JTAG
A boundary-scan technology to simplify the human test complexity and improve test efficiency was presented in 1985 by Joint Test Action Group (JTAG). Then it developed as the IEEE1149.1-1990 standard ---- Test Access Port and Boundary-Scan Architecture. This standard gives the basic architecture for a boundary-scan test system in which a chip is accessed using the Test Access Port
(TAP) with boundary-scan architecture. As it is first developed by JTAG, so we often call the boundary-scan test system as JTAG for short.

JTAG mainly consists of TAP controller, Instruction Register (IR) and Data Resistors (DRs). Generally, The DRs are consist of bypass register, IDCDOE register, boundary-scan register and some other shifters. The TAP controller includes five signal lines, which are the test clock line (TCK), the test mode select line (TMS), the test data input (TDI), the test data output (TDO) and test reset line (TRST). The TMS controls the state transfer on the edge of TCK and the data transfer from TDI into the system and then go out via TDO. TRST is used to force the state machine reset quickly, and it is not necessary. Fig. 1 shows the boundary-scan architecture in Virtex-6 FPGA as the IEEE1149.1 standard. The IR is a fixed length shifter to save the instructions or select the DR. Bypass register is a 1-bit register for bypassing the scan chain. IDCDOE register is a 32-bit shifter restore the chip’s information such as type, version, manufactary and et al. Boundary-scan register connects with the I/O pins. Some other shifter may also be in the system and selected by the IR. The TAP controller is a sixteen states state machine shown in fig. 2. It provides the boundary-scan test system’s working states. In order to understand the process more easily, we may focus on the simplified process as shown in fig. 3 which only includes seven states. And the test process generally runs as follows.

(A) Reset the TAP controller. If the TMS keeps to “1” for five times at rising edge of TCK, or the TRST is set to “1”, the TAP controller will go to the TEST-LOGIC-RESET state. If no data will be transmitted, keep the TAP controller at TEST-LOGIC-RESET state.

(B) IR transmission. When a data transmit start, state data is captured into the IR at CAPTURE-IR state. Then the captured state data in IR are shifted out to TDO and the instructions are shifted into the IR at SHIFT-IR state. When a piece of instruction has been shifted into the IR, it is conducted at UPDATE-IR state and the relative DR is selected.

(C) DR transmission. When the instructions have been conducted, the data should be shifted out and captured into DR at CAPTURE-DR state, and then shifting at SHIFT-DR state. Meanwhile, the data should be shifted in DR and loaded into destination at UPDATE-DR state.

(D) Transmission one again or exit. If a new transmission is required, the step (B) and/or step (C) may be conducted again. Otherwise, runs the step (A).

3. Communicating Interface Design Based on Embed JTAG in FPGA

3.1. JTAG Supporting in FPGA
Now, JTAG has been widely used in integrated circuits, and almost all the FPGAs support the JTAG. Xilinx is the largest FPGA manufacturer, so we introduce the JTAG supported of Xilinx FPGAs, such as the FPGA in spartan-6, virtex-6, kenneX-7 and zynq-7 series. It can be used to finish the general function such as FPGA’s configuration, virtual scope (the ChipScope). Furthermore, four external IR instructions, the USER1, USER2, USER3 and USER4, are provided for the users to build standard DR route as IEEE1149.1 standard for data transmission in FPGA’s function design. The fig. 4 shows Spartan 6 FPGA Configuration with four IR instructions. It also indicates the length of IR in Spartan-6 FPGA is 6 bit.

In our design, we’d better use the USER4 and USER3, but not the USER1 and USER2, because the USER1 or USER2 may be used by the Xilinx tools. For example, the ChipScope may use the USER1 in default.

Table 3-2: Spartan-6 FPGA Boundary-Scan Instructions

| Boundary-Scan Command | Instruction | Description |
|-----------------------|-------------|-------------|
| EXTEST                | 000111      | Enables boundary-scan EXTEST operation. |
| SAMPLE                | 000001      | Enables boundary-scan SAMPLE operation. |
| USER1                 | 000010      | Access user-defined register 1. |
| USER2                 | 000011      | Access user-defined register 2. |
| USER3                 | 011010      | User code that allows fabric access to/from the TAP controller from JTAG primitive instance 3. |
| USER4                 | 011011      | User code that allows fabric access to/from the TAP controller from JTAG primitive instance 4. |

Figure 3. Simplified TAP controller working process

In an FPGA system, the programmable logical resources of FPGA are usual redundant, the JTAG of FPGA is not used during its normal working and the Xilinx FPGA JTAG supports functional data transmission. These characters provide a new probable solution for BIT design: BIT designed in the redundant logical resources and the data transmitting via the FGPA.

3.2. Communicating interface design

In order to simplify the using complexity, Xilinx provides a primal language to use the JTAG’s reserved four IRs(USR1, USER2, USER3 and USER4)\(^{[8]}\). The instance name for Spartan-6 FPGA is BSCAN_SPARTAN6. The other serial’s FPGAs have the similar instance name and port name. The Fig. 5 getten from ISE13.2 shows the using codes of BSACN_SPARTAN6 based on Verilog HDL.

In the functional design based on the primal language, we have to provide a shifter SHIFT_DR used as the DR in the JTAG’s scan chain as Fig. 6 shows. The MSL (LSB) of SHIFT_DR connects to the TDI (TDO) of BSACN_SPARTAN6. When the TAP controller is at fig. 3 shows state 4, the CAPTURE signal goes into “1”, and it can be used to enable loading the test results or states into the shifter, preparing to be shifted out the FPGA. When TAP controller goes into state 5, the SHIFT signal is set to “1” and it is used to enable shifting: The loaded test results or states in shifter is shifted out via TDO, meanwhile the commands or parameters can be shifted into the shifter via TDI if needed. After the datas’ shifting finished, TAP controller goes into state 6, and the UPDATE signal equals to “1” which is used to load the command or parameter in the shifter into the user designed BIT (or system). The RESET signal can be used to reset the BIT. The four IRs’ select via setting the parameter JTAG_CHAIN, for example, the JTAG_CHAIN=4 means using the USER4.

Based on the provided method above, after the JTAG received the USER4’s IR (equals to 6’b011011 as shown in Fig. 4), the data on JTAG’ TDI pin can be shifted into the provided
SHIFT_DR, and the data in the SHIFT_DR load at CAPTURE state is shifted out through the JTAG’s TDO pin. Fig. 7 shows the apply scheme of the BITE interface design. In the BITE logics, a shifter is designed for buffing data. The shifter is used as DR in the boundary-scan system and the connected via the BSACN_SPARTAN6 primal language.

3.3. A design demo
Now we design an FPGA solder joint resistance BITE to further introduce our method. The BITE’s function includes the follows. It gets the FPGA pin’s solder joint resistance (can be used for PHM and et al.) through a method (not our focus, and similar to method in paper [9]). The current resistance and history maxima resistance (the resistance is proportional to a time variable $t$, so we record the $t$ as solder joint resistance) are recorded and send out via JTAG. The BIT can be enabled and disabled. The maxima resistance is clean when the BIT is disabled.
Fig. 8 shows the structure of the BITE. It includes four same BIT blocks, each block tests a solder joint. They are connected in serial to form a daisy chain: one block’s TDO connects to the next block’s TDI. The bit width of $t_i$ is 8, so the bit width of each block recording $t_i$ and $t_{i_{\text{max}}}$ is 16. It means the SHIFT-DR is 64 bit or 8 byte length, or the daisy chain’s DR is 64 bit length.

In each block, the tested $t_{i_{\text{max}}}$ and $t_i$ are loaded into SHIFT-DR while CAPTURE is high logical level. Then shifted out from TDO while SHIFT is high logical level, meanwhile, the block enable single is shifted into SHIFT-DR through TDI. Then the LSB of SHIFT-DR is loaded into EN register while UPDATE equals to 1. When EN=0, the $t_{i_{\text{max}}}$ is clear to zero. When EN=1, if $t_i > t_{i_{\text{max}}}$, the $t_{i_{\text{max}}}$ is replace by $t_i$ to keep the $t_{i_{\text{max}}}$ no less than $t_i$.

When the BIT is designed, we can read out the test results via the JTAG.

4. Experiment and results

Fig. 9 shows the experiment system. The large PCB board includes some solder joint test circuit, and a Xilinx spartan6 FPGA, XC6SLX9-TQG144, in which including the above mentioned BITE. Its JTAG port is located at the right-top corner, connected to the UART-JTAG convertor on the right-bottom corner in the figure with 10 pin cable. The UART-JTAG convertor[10] connects to the computer. The UART-JTAG convertor is used for data format changes between the UART data pack and the JTAG scan. The UART data pack consists of six parts: synchronous header, IR length, DR length, IR data, DR data, stop configuration, noted as: 5A-A5-50-05--IR_H-IR_L--DR_H-DR_L--IR_D0 ... IR_Dn--DR_D0 ... DR_Dm--AA.

(1) Synchronous header. It is fixed four bytes: 5A-A5-50-05(hexadecimal, Hex).
(2) IR length. It includes two bytes: IR_H-IR_L. the IR bit length equals to 256*IR_H+IR_L. The IR_H must no larger than 15 because IR_H larger than 15 is reserved.
(3) DR length. It includes two bytes: DR_H-DR_L, the DR bit length equals to 256*DR_H+DR_L.
(4) IR data. Its length is decided by the IR length. If IR length is 0, no this part in the pack. Otherwise, 8 bit format a byte: the LSB first shifted into TDI, and the first shifted from TDO save in the LSB. If less than 8 bit is left, the significant bit(s) is/are set to zero.
(5) DR data. It is Similar to IR data.
(6) Stop confirmation. One fixed byte 0xAA.

Fig. 10 shows the data packs sent to and received from the convertor. The test process goes as follows.

![Fig. 10 UART data packs send and received](image-url)
(1) Integrity test. Data pack for reading IDCODE command is send the check the scan chain. The
detail command is “5A A5 50 05 00 06 00 20 1B FF FF FF AA”. It shows the IR length is six
(0x0006), the DR length is thirty-two (0x0020), the IR data is “2’b011011” (6’h1B), and the data is
“FF FF FF FF”(Only thirty-two place holders are needed, and their real values are not concerned.). We
can find out the read back IDCODE is “h24002093” which is same as the shown in Spartan 6 FPGA’s
configuration user guide[8]. Result shows the data transmission is good.

(2) Disable the BITE, and clear its SHIFT_DR. we send the disable BITE data pack as shown in fig.10. The return data is the data in SHIFT_DR. Because we don’t know whether the BITE has been
enabled, this return value is useless.

(3) Enable the BITE. We send the enable BITE data pack, and the return data is zero, because the
BITE is enabled at first before state. We send the enable BITE data pack again, the return data pack
has including the tested \( t_s \) and \( t_{\text{max}} \).

(4) Changing the simulative solder joint resistance. We change the simulative solder joint
resistance, and different \( t_s \) of block 4 is get, and the result that the \( t_{\text{max}} \) of block 4 equals the largest
\( t_s \) has been tested. Five test results are shown in the Fig.8.

(5) Disable BITE. The data pack for disable BITE is send twice. The first return result including
the tested \( t_s \)s and \( t_{\text{max}} \)s. The second return tested values are zeros, meaning the BITE has been
disabled.

The test result shows the BITE has the capabilities of recording maxima tested value, test enable
and disable and et al, and sending test results via the JTAG with the UART-JTAG convertor. We can
conclude that the presented JTAG based BITE interface design method for FPGA is efficient.

5. Conclusions
Based on the provided TAP controller user interface by Xilinx in its FPGA, a design method for in-
FPGA BITE data transmission interface based on its JTAG is presented. A demo design is
implemented by the presented method, and an experiment is also conducted. The results show the
presented method can realize data transmission without using FPGA’s IO resource. It is valuable for
the BITE design with limited I/O resource. It also can be used for BITE update in a designed FPGA
system.

References
[1] P. H. Bardell, W. H. Mcanney, and J. Savir, "Built-in test VLSI: Pseudorandom techniques,"
Int.j.circ.theor.appl, vol. 40, pp. 37--47, 1987.
[2] J. Xu, J. Li, and Y. Jiang, "Components Locating in PCB Fault Diagnosis Based on Infrared
Thermal Imaging," in International Conference on Information and Computing Science,
2009, pp. 7-9.
[3] D. L. Bekker, J. F. L. Blavier, R. W. Key, and D. M. Rider, "An FPGA-based Focal Plane Array
interface for the Panchromatic Fourier Transform Spectrometer," in Aerospace Conference,
2011, pp. 1-10.
[4] M. Cardinali, R. Dzyhgadlo, A. Gerhardt, K. Götzen, R. Johler, G. Kalicy, H. Kumawat, D.
Lehmann, B. Lewandowski, and M. Patsyuk, "Frontend electronics for high-precision single
photo-electron timing using FPGA-TDCs," Nuclear Inst & Methods in Physics Research A,
vol. 766, pp. 231-234, 2014.
[5] L. Gao, Z. Chen, L. Wu, and N. Pu, "Design of digital transmitter based on FPGA," Electronic
Test, 2011.
[6] I. C. Society, "IEEE Std 1149.1-1990: IEEE Standard Test Access Port and Boundary Scan
Architecture," ed, 1990.04.
[7] Xilinx. Virtex-6 FPGA Configuration User Guide (v3.4) [Online].
[8] Xilinx. Spartan-6 FPGA Configuration User Guide (v2.5) [Online].
[9] N. Wang, Y. L. Li, Y. Zongyue, and Z. Ren, "Online test method of FPGA solder joint
resistance with low power consumption," Microelectronics Reliability, vol. 55, pp. 1867-1871, August 2015.

[10] N. WANG, J. ZHAO, G. CHAO, and Q. MAO, "JTAG-UART converter design for boundary-scan test" presented at the Seventh national academic exchange of China Institute of aerodynamics and measurement and control of Specialized Committee, 2017.(in Chinese)