Boundary Scan Architecture for a Double Precision Floating Point Subtractor

Anirudh Kashyap[1],Kusuma Keerthi[2] and Dr. Shilpa D.R.[3]
[1]*Student, R.V. College of Engineering, Bengaluru*
[2]*Head of Skill Development, Silival Pvt. Ltd., Bengaluru*
[3]*Associate Professor, R.V. College of Engineering, Bengaluru*
[1]*anirudhkashyap.ec17@rvce.edu.in,* [2]*kusumakeerthi99@gmail.com,* [3]*shilpadr@rvce.edu.in*

Abstract: *The boundary scan logic for testing was developed in order to make the process of testing easier for System-on-Chip (SoC) [1] architectures. The proposed work focuses on designing a boundary scan logic for a 64-bit floating point subtractor unit. The TAP controller designed is capable of executing the three mandatory Joint Test Action Group (JTAG) instructions of the IEEE 1149 standard. The testing architecture has the potential to not only test the functionality of the core logic but also to test single stuck-at faults for all the inputs and outputs of the core logic. A provision for bypassing the core logic was made in order to skip the IC while testing numerous ICs together. A simulation was also performed to demonstrate the above procedures. The designed module can further be used in a larger circuit with other ICs [2]containing a similar boundary scan structure with individual TAP controllers.*

Keywords: *Boundary Scan, TAP Controller, JTAG, IEEE 1149, State Machine*

1. Introduction

Testing refers to the process of checking the correctness of a manufactured device by applying certain test stimuli and observing the output corresponding to those stimuli. It may seem obvious that the process of testing will vary from one manufactured device to another. Each device according to the above assumption would require a unique testing equipment known as Design for Test. In today's world there are a large number of designs and it is close to impossible to think of a dedicated testing unit for each of those devices [3]. IEEE therefore created a standard for testing known as IEEE 1149 by the Joint Test Action Group (JTAG) in the 1980s. This standard completely scraps the idea of creating a dedicated testing unit for different designs. Instead, it defines a method of testing any Printed Circuit Board or Integrated Circuit (IC) by making use of 5 pins and a controller[4]. These pins are: Test Mode Select (TMS), Test Data In (TDI), Test Data Out (TDO), Test Reset (TRST) and Test Clock (TCK). The insertion of Design for Testability is an important step in the digital design flow and it occurs before logic synthesis. The controller is known as the Test Access Port (TAP) Controller. From a hardware standpoint the JTAG testing standard defines a wrapper around the core logic consisting of a scan chain of registers. They are each linked to the input and output of the core logic with their corresponding parallel inputs and parallel outputs. They form a scan chain with their serial inputs and serial outputs. With these pins, a complete wrapper is formed around the core logic and boundary scan operations can be performed.

2. Boundary Scan with Core Logic

The Boundary Scan Logic is a well-planned architecture that is linked to the periphery of any design that forms the core logic. The logic consists of a Test Access Port (TAP) controller and multiple Boundary Scan Cells (BSCs) or a Boundary Scan Register (BSR) [5]. The connection of the Boundary Scan Logic wrapper [6] is shown in Figure 1. These cells are linked to the primary inputs and outputs of the core logic. Therefore, the size of the boundary scan register will be equal to the sum of the number of primary inputs and primary outputs on the core logic. The cell in itself consists of two inputs and two outputs. The inputs are - Parallel In and Scan In and the outputs are: Parallel Out and Scan Out. The core logic used in the stated design is a Double Precision Floating Point Subtractor. This subtractor incorporates a 52-bit Brent-Kung Adder module which proves to be highly area efficient among all the other prefix-tree adders [7]. The subtractor design also accommodates for all the exceptions like - positive and negative zeros, positive and negative infinity, underflow and overflow. These are drawn as outputs of the core logic along with the 64-bit difference. The size of the boundary scan register therefore is computed as 200 bits. Thescan chain of

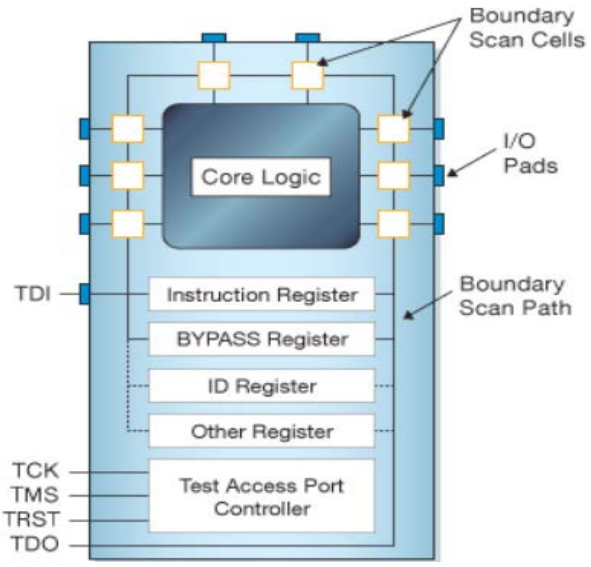

Figure 1. Boundary Scan Wrapper

registers start from the outputs and moves all the way to the last bit of the input. The boundary scan register requires four control signals that are linked to each cell for its operation, namely: ShiftDR, ClockDR, UpdateDR and Mode.

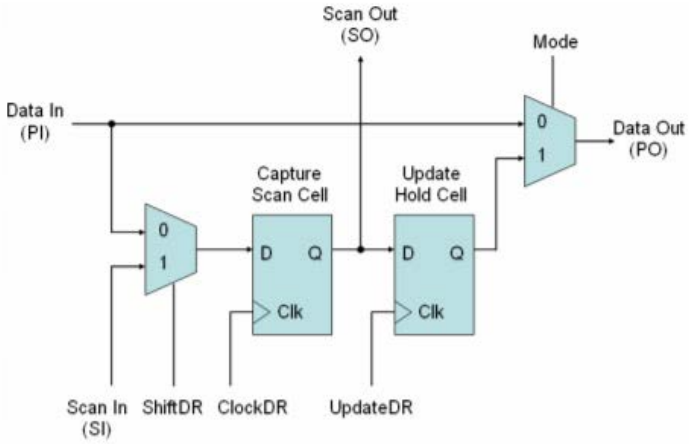

Figure 2. Schematic of one Boundary Scan Cell

These control signals are provided by the TAP controller based on the instruction that is loaded in the Instruction Register. The TAP controller [8] works on the basis of a state diagram and each of these states is responsible to trigger one or more control signals linked to the BSR. The ShiftDR signal is a select signal that is used to choose between the Parallel Input and the Scan Input. The ClockDR signal is used to trigger the capture flip-flop to propagate the bit coming from the input multiplexer. To further propagate the captured bit, the UpdateDR signal is used and it triggers the update flip-flop. The Mode signal, similar to the ShiftDR signal, is used as a select line for the multiplexer that chooses between the Parallel Input and the output of the update flip-flop. The connection of all these control signals to a single cell is shown in Figure 2.

3. Design and Implementation

The design of the Boundary Scan Logic consists of many submodules - Boundary Scan Register, Instruction Register (IR), Bypass Register (BR) and the TAP Controller [9]. Since the core logic, that is, the Double Precision Floating Point Subtractor, has a total of 200 pins including inputs and outputs, the BSR should also consist of 200 bits. The block diagram displaying the primary inputs and outputs including the testing pins are shown in Figure 3.

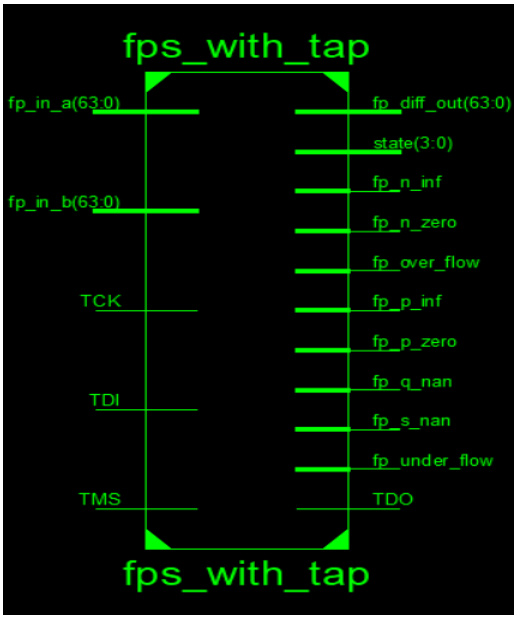

Figure 3. Black-box Diagram of the overall design

The instruction register however is designed for accommodating the mandatory instructions of the 1149 standard which are three in number. Hence the size of the IR is 2 bits. The BR is used only to skip the entire chip while testing and therefore is always one bit wide. The TAP Controller [10], as mentioned in the previous section, is based on a finite state machine that consists of 16 states as shown in Figure 4. The state diagram is divided into two main verticals. The first vertical controls the BSRs also known as Data Registers (DR) and the second deals with controlling the Instruction Register (IR) [11]. The state transition of the controller is governed by a single bit called Test Mode Select (TMS). A unique characteristic of this TAP controller is that, irrespective of its current state, by asserting and holding the TMS pin for 5 clock cycles, the controller reaches the reset state. The design of the TAP controller is used to perform many instructions. Each instruction that a TAP Controller [12] performs is characterized by a code in binary and is referred to as that instruction's op-code. The op-code is loaded to the IR from the TDI pin and the loaded op-code is decoded by the Instruction Decoder. In combination with the Instruction Decoder, the TAP controller accordingly manages the control signals that are connected to the BSR or BR. According the IEEE 1149 standard there is a set of three mandatory instructions [13] that are to be a part of the JTAG Boundary Scan architecture and they are listed below

3.1 Extest

Extest [14]is an operation that is used to test the inter-connects between two ICs on the board. This is done by performing the shift operation through the TDI pin of one IC all the way to the interconnect under test. The update pulse will then move this data to the Parallel Out pin of that particular Boundary Scan Cell. Since there is an inter-connection between the ICs, the Parallel Out of one is connected to the Parallel In of the other. ShiftDR for the second IC is de-asserted for the bit to move to the capture flop and a capture pulse through clockDR is given to further move it to the TDO pin. From TDO, it can be configured in the shift register mode and can be shifted out to the main TDO pin. The actual output can then be matched with the expected one. The op-code according to IEEE 1149 standard is supposed to be all 0s.

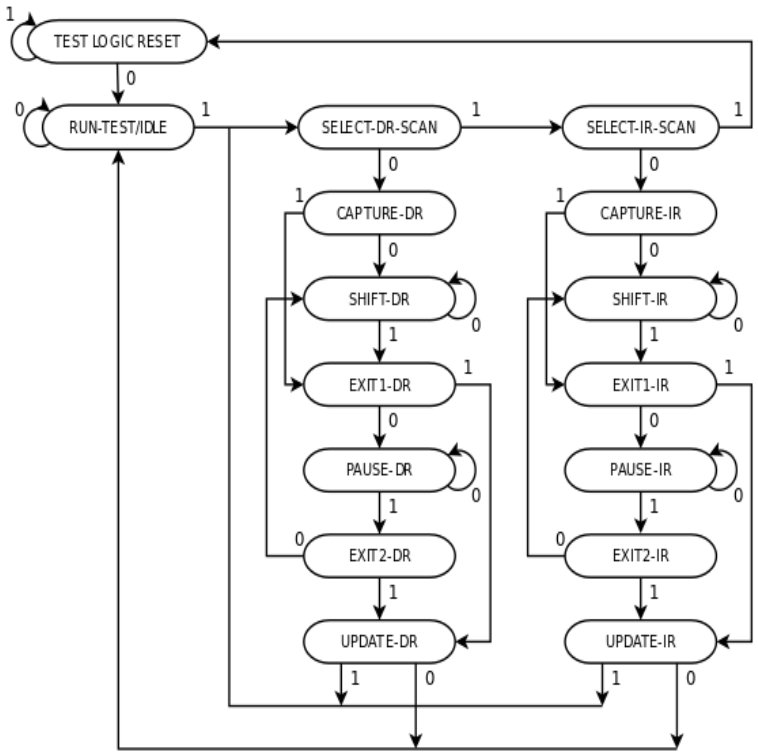

Figure 4. TAP Controller State Machine

3.2 Bypass

The Bypass operation is the simplest operation among the three. It is used in order to skip testing an IC on the board. The bypass select signal (SelectBR) is provided by the TAP controller in conjunction with the Instruction Decoder of that particular IC. Asserting this control signal aids in moving the bits from TDI to the BR and out of TDO. IEEE 1149 standard states that the op-code for this instruction is all 1s.

3.3 Sample/Pre-load

This instruction is often used as a prerequisite to the Extest instruction. The control signals coming from the TAP controller to the BSR connects the parallel input to the parallel output

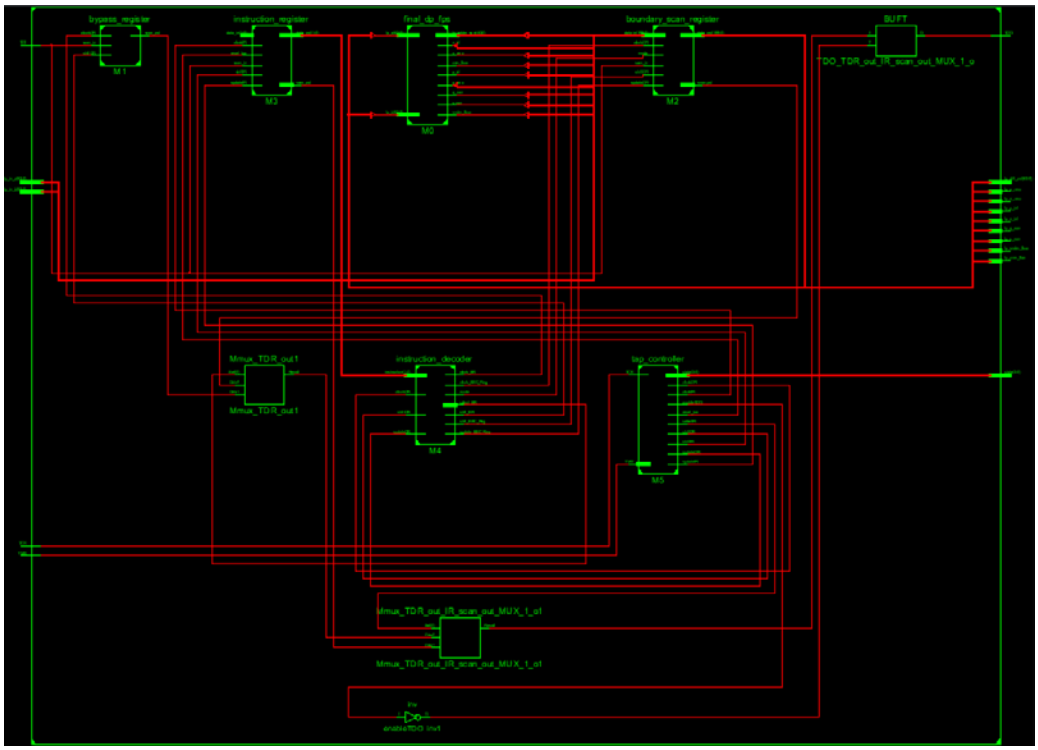

Figure 5. RTL Schematic of the Implemented Design

and also connects TDI to TDO. It is important to note that if a parallel value is loaded to the inputs of the BSR, the core logic will compute its result and will be available at the output. There is no restriction on the op-code for this instruction. In the implemented 2-bit IR, the op-code given to this was 01.The implemented design of the complete device consisting of the Double Precision Floating Point Subtractor as well as the boundary scan architecture is shown in Figure 5. The three instructions described above are verified on the implemented design in the following section.

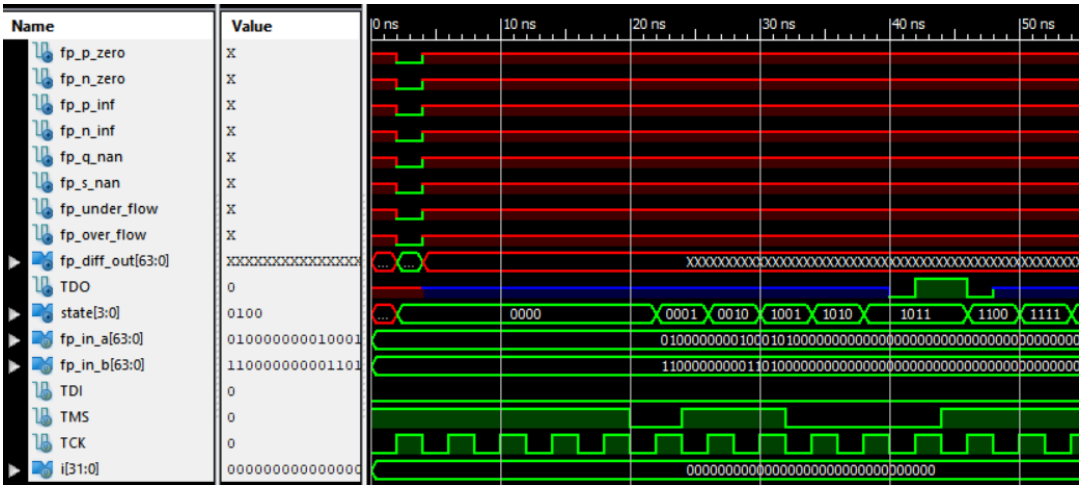

Figure 6. Loading Extest Instruction

4. Results

The implemented design is programmed to execute three mandatory instructions as stated in the IEEE 1149 standard. A testbench was written in verilog and simulated using Xilinx ISE 14.7 software. Extest instruction as mentioned in the previous section, is responsible to check for faults [15] at the interconnection of two ICs. The faulty net in the design is assumed tobe the output “fp_p_zero” and the fault is Stuck-at 0. To test this net, an opposite value is forced into it with the help of the extest instruction. The op-code is first loaded and updated into the IR as shown in Figure 6. An appropriate bitstream is provided at TDI during the state machine’s DR vertical execution. The bitstream is chosen such that the net mentioned above lands with a logic 1. This is depicted in Figure 7.

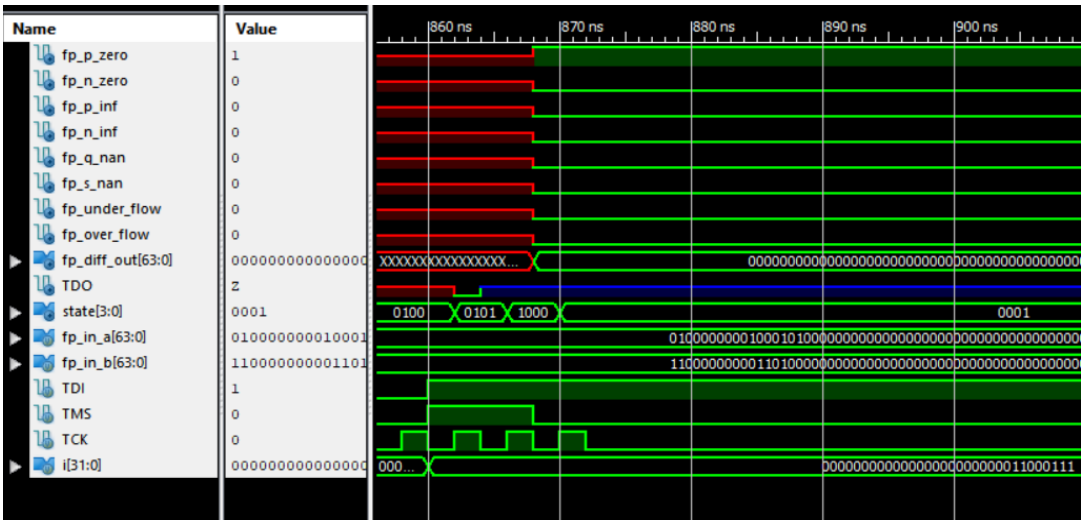

Figure 7. Testing Stuck-at 0 Fault at “fp_p_zero”

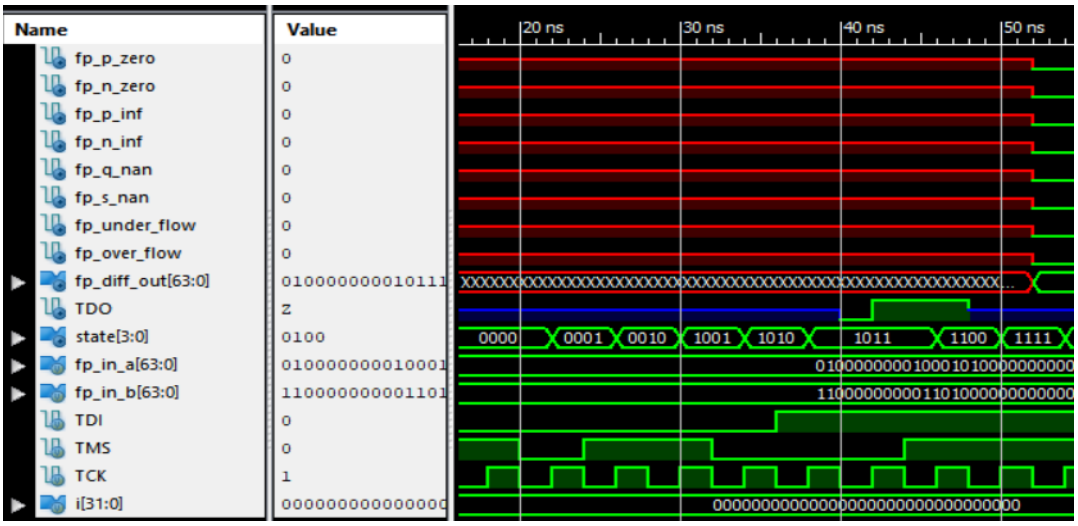

Figure 8. Loading Bypass Instruction

The instruction register is being loaded with the Bypass instruction in Figure 8. According to the controller, when the CaptureIR state is reached, the loading of the op-code to TDI must begin. This is displayed as a waveform in Figure 9. Once an instruction is loaded, the controller moves to the DR vertical of the state machine. To demonstrate the Bypass instruction, a string of 1s is fed into TDI and at the end of the DR state transitions, as seen in Figure 9 is forced out of TDO thereby skipping the subtractor logic entirely. The third

instruction is Sample/Pre-load with op-code 01. In Figure 10, the IR is loaded with 01 which is fed in through TDI. Since this instruction allows parallel inputs, the floating-point operands loaded are +9.25 and -6.5, forming the parallel inputs fp_in_a and fp_in_b respectively. According to the instruction's functionality, the data coming at TDI should be available at TDO along with the computed result of the core logic. In Figure 10, the core logic provides its output as fp_diff_out (when translated from binary gives +15.75) along with exception outputs. It also shows that TDO follows TDI after the controller passes the DR vertical.

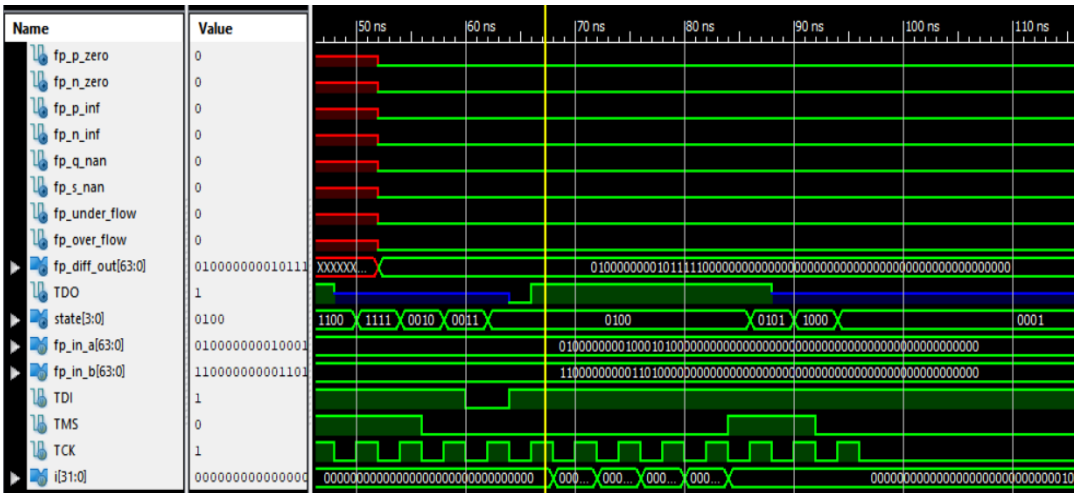

Figure 9. A string of 1s is bypassed

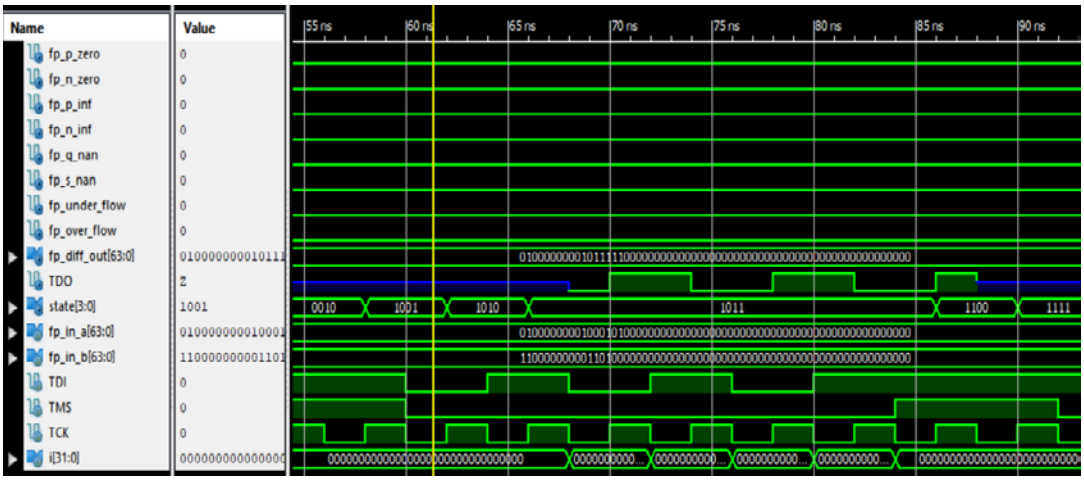

Figure 10. Output of Sample/Preload

5. Conclusion

The process of testing finds its application in various semiconductor industries. It plays an important role in determining the number of good and bad ICs after manufacturing, aiding the supply of high-quality products in the semiconductor market. One of the major components of testing is the boundary scan logic. The boundary scan architecture dedicated for a double precision floating-point subtractor was designed on Xilinx ISE 14.7 using Verilog HDL. It was also successfully synthesized on a Virtex – 6 FGPA. A suitable testbench covering all the mandatory instructions of boundary scan dictated by the JTAG

standard was created for the designed unit. Analyzing the resulting waveforms showed that the architecture was compliant with all the rules of boundary scan testing. Some plausible improvements for the stated design would be to optimize the data register path based on the fault model, moving from TDI to TDO which would reduce the consumed area. The second improvement to the design would be to create customised boundary scan cells that are capable of executing more than the three instructions presented in this design.

REFERENCES

[**1**] S. F. Oakland, "Considerations for implementing IEEE 1149.1 on system-on-a-chip integrated circuits", Proceedings International Test Conference 2000 (IEEE Cat. No.00CH37159), 2000, pp. 628-637, doi: 10.1109/TEST.2000.894257.

[**2**] R. Burgess, P. Nagaraj and M. N. Waseq, "The boundary scan," in IEEE Potentials, vol. 14, no. 3, pp. 11-12, Aug.-Sept. 1995, doi: 10.1109/45.464687.

[**3**] S. Srinivas and H. N. Sheshagiri, "Design and implementation of boundary scan testing of core logic on FPGA", 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 327-331, doi: 10.1109/RTEICT.2017.8256609.

[**4**] IEEE Standard 1149.1, “Standard Test Access Port and Boundary-Scan Architecture", 2013.

[**5**] Z.Stamenkovic, M. Giles and F. Russi, "Combining internal scan chains and boundary scan register: A case study," IEEE EUROCON 2009, 2009, pp. 2064-2069, doi: 10.1109/EURCON.2009.5167932.

[**6**] X. Wang, L. Liang and X. Wang, "A New Solution to Implement Multi-Full-Scan-Chain Test with JTAG", 2006 8th International Conference on Solid-State and Integrated Circuit Technology Proceedings, 2006, pp. 2155-2157, doi: 10.1109/ICSICT.2006.306667.

[**7**] Anirudh Kashyap, Aniruddh M, Kusuma Keerthi. “Performance Analysis of Double-precision Floating-point Adder using Prefix Tree Adders”, International Journal of Advanced Science and Technology, 2020, 29(7), 2225-2233.

[**8**] A. Sguigna, "JTAG/Boundary Scan for Built-In Test", 2018 IEEE AUTOTESTCON, 2018, pp. 1-3, doi: 10.1109/AUTEST.2018.8532506.

[**9**] Yuejian Wu and Paul Soong, "Interconnect delay fault testing with IEEE 1149.1", International Test Conference 1999. Proceedings (IEEE Cat. No.99CH37034), 1999, pp. 449-457, doi:10.1109/TEST.1999.805767.

[**10**] S. Mitra, E. J. McCluskey and S. Makar, "Design for testability and testing of IEEE 1149.1 TAP controller", Proceedings, 20th IEEE VLSI Test Symposium (VTS 2002), 2002, pp. 247-252, doi: 10.1109/VTS.2002.1011145.

[**11**] L. Zhiwei and P. Zhongliang, "Realization of Integrity Test of Boundary-Scan Structure", 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2020, pp. 722-724, doi: 10.1109/ICAICA50127.2020.9182579.

[**12**] S, Shelja& R, Nandakumar & C, Muruganantham. “Design of IEEE 1149.1 Tap Controller IP Core”, 2016, 107-118. 10.5121/csit.2016.60910.

[**13**] Texas Instruments, “IEEE std 1149.1(JTAG) Testability, Primer”, Texas Instruments,1997.

[**14**] Sungju Park and Taehyung Kim, "A new IEEE 1149.1 boundary scan design for the detection of delay defects," Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000 (Cat. No. PR00537), 2000, pp. 458-462, doi: 10.1109/DATE.2000.840311.

[**15**] Vishwas K. Chaudhary, Manish J Patel, “A Research Paper on Designing a TAP (Test Access Port)”, Journal of Information, Knowledge and Research in Electronics and Communication Engineering, Nov 2012 to Oct 2013, Vol. 2 Issue 2, pp. 479 - 483, ISSN: 0975-6779.