RP Current Trends in Engineering and Technology
Vol. 2, No. 3 (July – September 2023) pp. 98–117
e-ISSN: 2583-5491

RESEARCH PLATEAU PUBLISHERS

Cite this article: Aadya, Verifying the compliance of a general-purpose microcontroller with the IEEE (JTAG) 1149.1 standard, *RP Cur. Tr. Eng. Tech.* **2** (2023) 98–117.

Original Research Article

Verifying the compliance of a general-purpose microcontroller with the IEEE (JTAG) 1149.1 standard

Aadya*
Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala, India
*Corresponding author, E-mail: aadyashamina@gmail.com

ARTICLE HISTORY
Received: 28 June 2023
Revised: 29 August 2023
Accepted: 30 August 2023
Published online: 17 Sept. 2023

KEYWORDS
Boundary scan testing; IEEE 1149.1; Compliance; Microcontroller; Scan chain.

ABSTRACT
Boundary scan testing is a widely used technique in the semiconductor industry for testing and debugging integrated circuits. The objective of this article is to describe the design of a general-purpose microcontroller and explain the process to check the compliance of design with the IEEE 1149.1 Standard, which is the basic standard to describe the interconnections and placement of the components in a design for its better testing. The implementation process involves identifying the key digital components that need to be included in the boundary scan chain & analyzing the microcontroller's architecture. The article discusses the challenges encountered during the implementation and verification process and proposes solutions or workarounds. The compliance is successfully proved for a sample design of a general-purpose microcontroller. Overall, this showcases the importance and benefits of implementing a design of a general-purpose microcontroller in compliance with IEEE 1149.1 Standard.

1. Introduction

DFT (Design for Testability): [1] Integration of design and testing circuitry is called design for testability. Various testability measures and ad hoc testability enhancement methods were proposed and used in the 1970s and 1980s to serve the purpose of testability improvement and increased controllability &observability of the design. Although these methods improved the testability and eased the procedure of generating ATPG (Automatic Test Pattern Generation) for the sequential circuits, their fault coverage is low. For combinational circuits, many innovative ATPG algorithms have been developed for automatically generating test patterns within a reasonable amount of time. But for sequential circuits automatically generating test patterns met with limited success, due to the existence of numerous internal states that are difficult to set and the presence of feedback loops in these circuits. Difficulties in controlling and observing the internal states of sequential circuits led to the adoption of structured DFT approaches in which direct external access is provided for storage elements. These reconfigured storage elements with direct external access are referred to as scan cells. Once the capability of controlling and observing the internal states of a design is added, the problem of testing the sequential circuit is transformed into a problem of testing the combinational logic, for which many solutions already existed.

Scan design is currently the most popular structured DFT approach. It is implemented by connecting selected storage elements of a design into multiple shift registers, called scan chains, to provide them with external access. Scan design accomplishes this task by replacing all selected storage elements with scan cells, each having one additional SI (Scan Input) port and one shared/additional SO (Scan Output) port. By connecting the SO port of one scan cell to the SI port of the next scan cell, one or more scan chains are created.

Scan Cell Designs

1. Full Scan Design, in this design all storage elements are selected for scan insertion.
2. Almost Full Scan Design, in this design almost all (more than 98%) storage elements are selected for scan insertion.
3. Partial Scan Design, in this design some storage elements are selected and sequential ATPG is applied for scan insertion.

Approaches of DFT

Ad-hoc approaches: In this approach good design practices learned through experience are applied on design and a bad design practice is replaced with a good one. Test Point Insertion is the most common and applicable ad-hoc approach. This technique is for improving the controllability and observability of internal nodes. Testability analysis is typically used to identify the internal nodes where test points should be inserted, in the form of control or observation points, controllability of the circuit nodes is dramatically improved. This, however, results in additional delay to the logical path. So, control points should not be inserted on a critical path. Moreover, it is preferable to add a scan point, which is a combination of a control point and an observation point, instead of a control point, as this allows us to observe the source end as well. Thus, in Ad-hoc approach scan point must be inserted instead of test point because it can improve both controllability and observability of the circuit.

Structured approaches: A methodical and systematic approach with much more predictable results. Scan design is the most favorable and appropriate structured approach in

Copyright: © 2023 by the authors. Licensee Research Plateau Publishers, India
This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

which testability of a circuit is improved by improving the controllability and observability of the storage elements in sequential design.

In the Scan design method, the sequential circuit is converted to a scan design with three modes of operation: normal mode, shift mode, capture mode. In Scan design approach the storage elements in design are provided with external access by converting them to the scan cells and then stitching them together to form one or more shift registers, called scan chains. Scan cells have two different input sources that can be selected. The first input, data input, is driven by the combinational logic of a circuit, while the second input, SI, is driven by the output of another scan cell to form scan chains. These scan chains are made externally accessible by connecting the scan input of the first scan cell in a scan chain to a primary input and the output of the last scan cell in a scan chain to a primary output. Because there are two input sources in a scan cell, a selection mechanism must be provided to allow a scan cell to operate in two different modes: normal/capture mode and shift mode. In normal/capture mode, data input is selected to update the output. In shift mode, scan input is selected to update the output. This makes it possible to shift in an arbitrary test pattern to all scan cells from one or more primary inputs while shifting out the contents of all scan cells through one or more primary outputs.

Boundary Scan: It is a method of testing interconnects on printed circuit boards by addition of a scan cell at each pin of the device to selectively override the functionality of the pin (without interfering with the normal functionality of the circuit). Boundary Scan cells for the pins of a component are connected to form a shift register chain around the border of design. This path is provided with serial input & output connections & appropriate clock &control signals.Basic idea about the boundary scan can be interpreted from Figure 1.

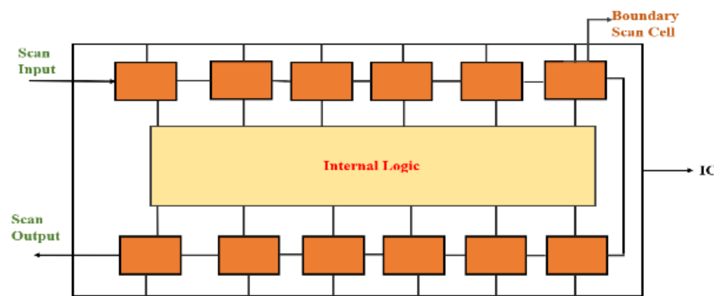

Figure 1: Basic idea of boundary scan.

We can interconnect multiple IC by using interconnects & scan them accordingly. Test data can be shifted into all BSC (Boundary Scan Cell) and loaded in parallel through the component interconnections. As shown in the Figure 2, there are two types of interconnections:

Serial Test Interconnections: The connections are used for connecting BSC.

System Interconnections: The connections are used for connecting the BSC of different testing IC's (Integrated Circuits).

Both can be used to confirm that each component performs the required function& components are connected in correct manner.

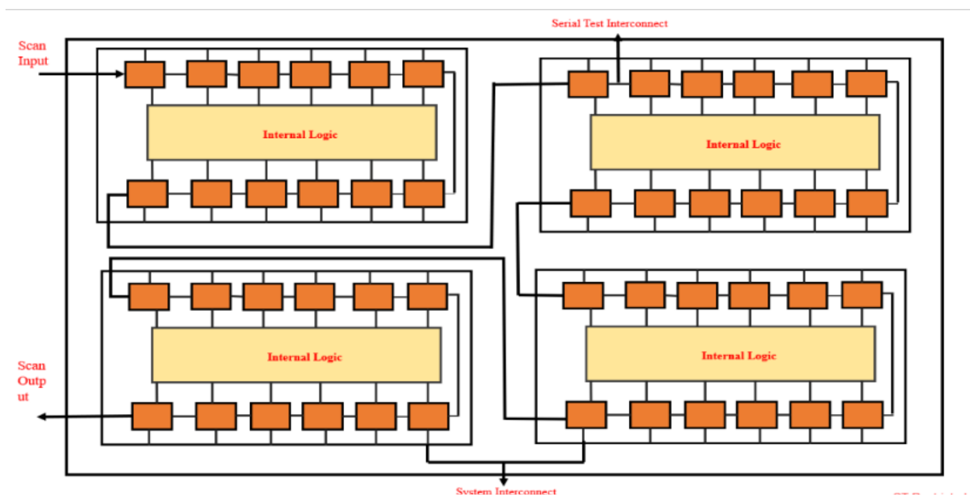

Figure 2: A board containing 4 ICs with boundary scan.

1149.1 IEEE JTAG Standard: [2]This standard tells about the test logic and approaches to test the interconnection between IC's and to test the observation/modification of circuit activity. Test logic is composed of BSR (Boundary Scan Register) and other building blocks that can be accessed through TAP (Test Access Port). This standard explains a circuitry that allows test instructions, test data to be fed into a component and the results of execution are read out where the

test instructions, test data, results are always put to the system in the serial format. Sequence of operations in the design are controlled by ATE (Automatic Test Equipment). Signals are controlled by: TCK (Test Clock), TMS (Test Mode Select) signals.

Objective of standard: Defining the boundary scan architecture that can be adopted as a standard feature of IC designs, thus allowing the required test framework to be created on assembled PCB (Printed Circuit Board).

1149.1Boundary Scan Architecture: Figure 3 shows the basic architecture of design with boundary scan.

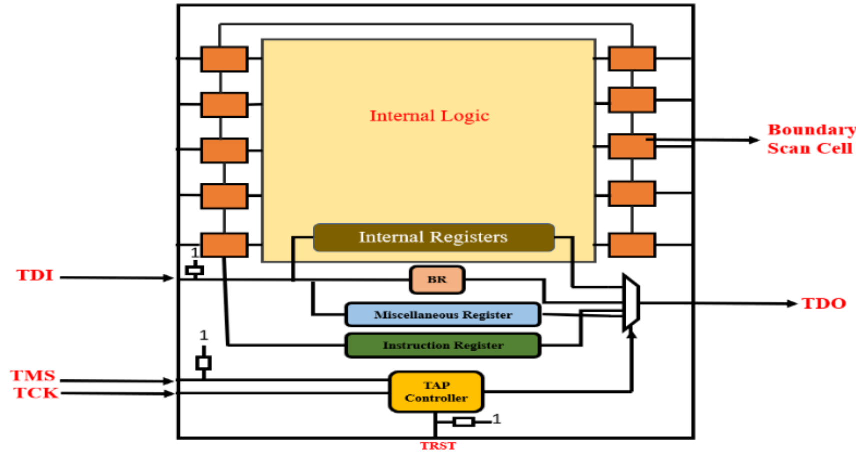

Figure 3: 1149.1 boundary scan architecture.

Basic operation steps of boundary scan

- Instruction is sent serially through TDI (Test Data Input) into the instruction register.
- The instruction register tries to configure the test circuitry to respond to the instruction.
- Test pattern is shifted into selected data register& applied to logic to be tested.
- Test response is captured into some data register.
- Captured response is shifted out& new test pattern is shifted in simultaneously.

Ports

1. **TDI:** The test data & test instructions are obtained serially by test logic at TDI. Signal presented at TDI shall be sampled into test logic at the rising edge of TCK.On shifting data from TDI to TDO (Test Data Output) no inversion should occur even after the multiple rising & falling edges of TCK so that the operation of components is simplified. Values presented at TDI are clocked into the selected register on the rising edge of TCK.
2. **TDO:** It is the serial output of test instruction & data from test logic. Change of state occurs at the falling edge of TCK or TRST.Valid states of this signal are: 0,1, Z(undriven).To ensure race free operation TDO changes on falling edge of TCK.
3. **TMS:** Its value is checked at the rising edge of TCK.Value of TMS at rising edge of TCK tells the next state of TAP controller. The load presented by TMS should be least or smallest.
4. **TCK:** If TCK=0 value of state remains same as the value of previous state. If TCK=1 the system functions according to the instruction to be executed. The load presented by TCK should be small.
5. **TRST:** (Test Reset.) This provides asynchronous initialization. Used when test logic is not powered up in a known and uncontrolled state. TRST should not be used to initialize the system logic within the component because this can also cause asynchronous initialization of other test logic in the design. If TMS=0, TRST=1, TCK=1 race condition will occur and TAP controller will remain either in TLR or run test/idle state. <u>COMPONENTS OF 1149.1 BOUNDARY SCAN ARCHITECTURE.</u>

- TAP Controller
- Instruction Register
- Test Data Registers

1. BR(BYPASS Register).
2. BSR(Boundary Scan Register).
3. Miscellaneous Registers

Instruction registers and test data registers are separate shift registers that are connected in parallel & have a common serial data input & output.

TAP Controller: This is a 16 State synchronous FSM, that controls the sequence of operations of circuits. It responds to the changes on TCK and TMS signals, thus these two signals act as input to the controller. The state of controller changes when TCK =1 (high level/rising edge) and values of TMS changes. It is a mandatory controller in the design. The following (Fig. 4) shows the TAP controller states.

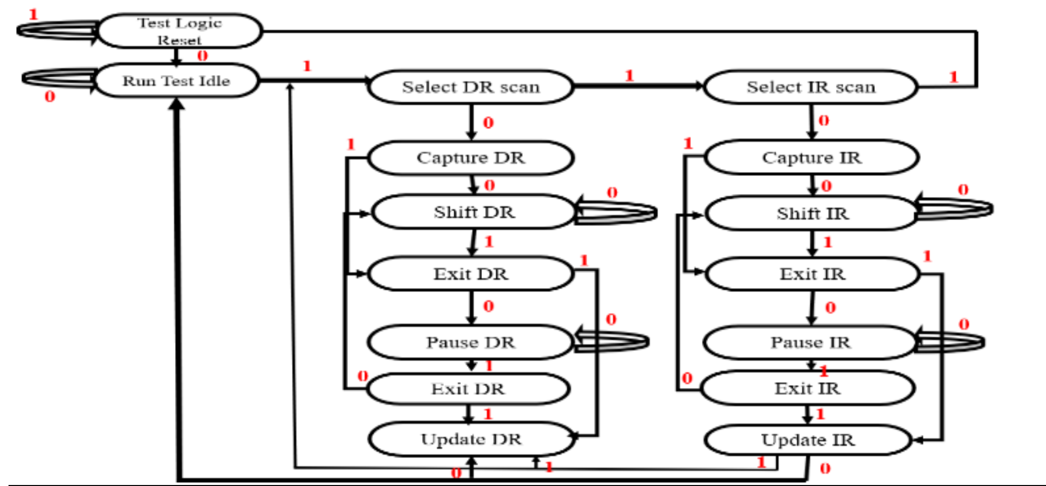

Figure 4: TAP controller states.

It controls the behavior of test logic &maintains synchronization across all components on a test scan chain to permit shifting, capturing, updating of data.

Explanation of States of TAP Controller

1. **Test Logic Reset:** The state has no effect on system logic. The controller remains in this state if it is not given any instruction to be executed. This state is acquired when either of the given conditions are satisfied: TCK=1 at least 5 times continuously & TMS=1.TCK=1 after 3 to 5 times continuously & TMS =1 (when some external interference arises.) or TRST= 0.
2. **Run-Test Idle:** In this state no action occurs in the design. This is a state between scan operations. If due to availability of instructions to execute, the controller is in this state then time spent in this state constitutes a delay. This state is acquired when TMS = 0 and will remain in the state until the value of TMS changes.
3. **Select-DR-Scan:** Temporary controller state. If TCK=1 the controller will exit from this state.

Table 1: Description of State Select-DR-Scan for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state it will go to capture state & scan sequence is initiated for data register.
1	1	Controller goes to Select IR scan state thus, it changes from Data Register to Instruction Register.

1. **Select-IR-Scan:** Temporary controller state in which all test data registers retain their previous state.

Table 2: Description of State Select-IR-Scan for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state it will go to capture state & scan sequence is initiated for instruction register.
1	1	Controller goes to TLR state.

2. **Capture -DR:** In this state data is loaded parallelly into the shift capture path of test data registers. If parallel data exists in a state & TCK=1, TAP exits the capture DR state. If no parallel data exists & TCK= 0, TAP retains its previous state.

Table 3: Description of State Capture-DR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state, it will go to shift-DR state.
1	1	Controller goes to Exit-1 state.

3. **Shift -DR:** Data is shifted one stage forward from TDI towards its serial output.

Table 4: Description of State Shift-DR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state, it will remain in the same state.
1	1	Controller goes to Exit-DR state.

4. **Exit-1-DR:**

Table 5: Description of State Exit-1-DR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state it will go to Pause-DR state.
1	1	Controller goes to Update-DR state & scanning process is terminated.

5. **Pause-DR:** Shifting of data between TDI & TDO is temporarily halted.

Table 6: Description of State Pause-DR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state it will remain in the same state.
1	1	Controller goes to Exit-2 DR state

6. **Exit-2 -DR:**

Table 7: Description of State Exit-2-DR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	If the controller is in this state it will go to Shift-DR state.
1	1	Scanning process is terminated and it goes to update stage.

7. **Update-DR:** Some test data registers may be provided with the latched parallel output to prevent changes at the parallel output, while data are shifted in the associated shift-capture path in response to certain instructions. All shift-capture paths in test data registers selected by the current instruction retain their previous state unchanged.

Table 8: Description of State Update-DR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller enters the Run-Test/Idle state.
1	1	Controller enters the Select-DR scan state.

8. **Capture-IR:** A pattern of fixed logic values is loaded parallelly into specific bits of IR on the capture path at TCK=1.

Table 9: Description of State Capture-IR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller enters the Shift IR state.
1	1	Controller enters the Exit 1-IR state.

9. **Shift-IR:** The data is shifted one stage from TDI to TDO on each rising edge of TCK.

Table 10: Description of State Shift-IR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller remains in same state.
1	1	Controller enters the Exit 1-IR state.

10. **Exit-1-IR:**

Table 11: Description of State Exit-1-IR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller goes to Pause-IR state.
1	1	Controller enters the Update-IR state & Scanning process is terminated.

11. **Pause-IR:** The shifting of IR is to be halted temporarily.

Table 12: Description of State Pause-IR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller remains in same state.
1	1	Controller enters the Exit-2-IR state.

12. **Exit-2 IR:**

Table 13: Description of State Exit-2-IR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller enters the Shift-IR state.
1	1	Controller enters the Update -IR state & scanning process is terminated.

13. **Update-IR:** In this state the bits in the IR shift-capture path are latched onto the parallel output on the falling edge of TCK.

Table 14: Description of State Update-IR for particular values of TCK, TMS.

Value of TMS	Value of TCK	Description
0	1	Controller enters the Shift-IR state.
1	1	Controller enters the Run-Test/Idle state.

TAP Controller operates only when either of the given conditions are satisfied TCK=1 (rising edge) is applied or TRST=0.

TAP Controller can be initialized by forcing it to TLR state at power-up by using TRST signal. It should not be initialized by system reset. A dedicated reset pin (TRST) is provided to allow initialization of TAP asynchronously.

IR (Instruction Register): It is formed by the combination of instruction cells & allows instruction to be shifted into design. Instruction is used to select the test to be performed or the test data register to be accessed or both. It shall include at least 2 shift register based cells capable of holding instruction data. No inversion of data occurs between serial input & serial output of instruction registers.IR must allow selection of Bypass register and access to Boundary Scan Register. Figure 5 shows an Instruction Register.

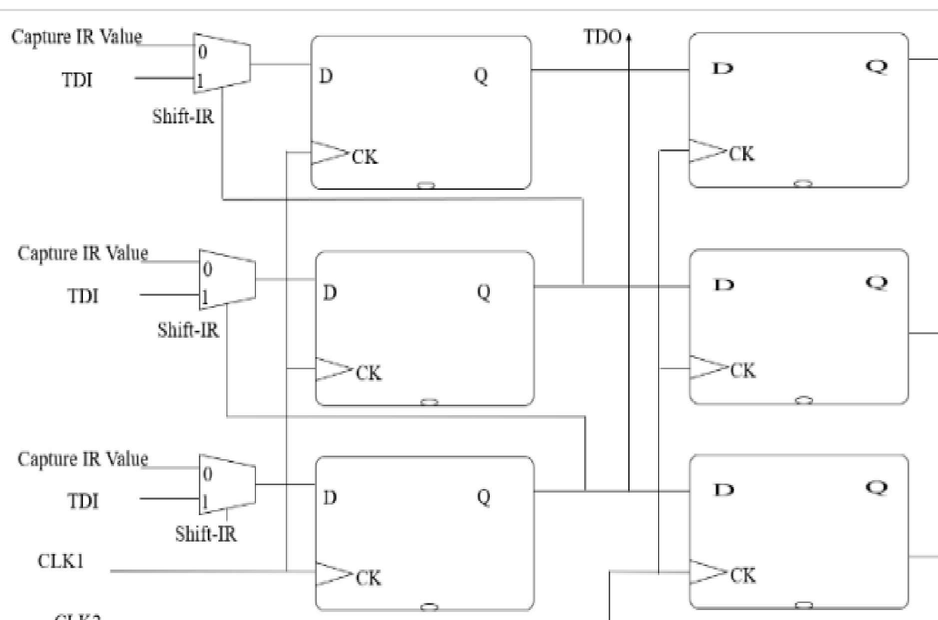

Figure 5: Instruction Register.

IR Cell:

Figure 6 below shows the IR cell formed by the Two D flip flops.

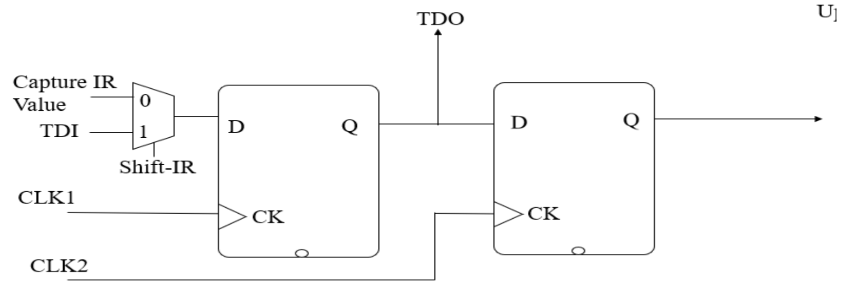

Figure 6: IR Cell.

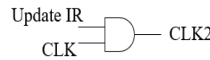

Figure 7: Formation of CLK2 for IR Cell.

Working: If CLK1 =1 Shift IR is 1(shift stage) then TDI will be available at the TDO without any inversion.

If CLK1=1 Shift IR is 0 (capture stage) then the instruction will be captured and loaded in the register. If CLK1=0, the cell is neither in shift stage nor in capture stage. If CLK1=0, CLK2=1, considering Figure 7 we can say that Update IR=1 (update stage), the loaded instruction in the register will be updated with a new instruction.

Instructions: Entered serially by an instruction register. Each instruction shall completely define the set of test data registers that operate or interact with on chip system logic when instruction is active. The test data registers that are active during an instruction execution should not interfere with the normal functioning of the logic. Each instruction shall select a single serial test data register path to be enabled to shift data between TDI and TDO in Shift-DR controller state. Active instruction can do two functions: 1) Selects the serial test data register path to shift data between TDI and TDO. 2) Defines the set of test data registers that may be used while instruction is active. Multiple instructions can be merged if they have the same binary code for operation and behaviors are not different e.g., SAMPLE & PRELOAD instructions can be combined to form a single instruction known as SAMPLE/PRELOAD. Where binary code is a sequence of data bits shifted serially into the instruction register from TDI.

Types of Instructions:

1. ***Public instructions:*** Available for use by the purchasers of the component. BYPASS, SAMPLE, PRELOAD, EXTEST these instructions must be provided if a component is compliant with the IEEE standard.
2. ***Private instructions:*** Allow component manufacturers to use TAP and test logic to gain access to test features embedded in the design for design verification, production testing, fault diagnosis.
3. ***Device identification register instructions:*** Use of optional device identification register allows a code to be serially read from the component that shows manufacturer's identity, part number, version number for the part. Device identification register is selected for scan by two instructions: IDCODE, USERCODE.

Description of Instructions:

BYPASS instruction: Bypass register contains a single shift register stage and provides a minimum length serial path between the TDI and TDO, for more rapid movement of test data from one component to another. The Bypass Instruction shall select the bypass register to be connected for serial access between TDI and TDO in the shift-DR controller state. This instruction can be entered by holding TDI to a constant value and completing an instruction scan cycle.

SAMPLE instruction: This allows a snapshot to be taken of the states of component's input and output signals without interfering with the normal operation of the logic. In this instruction data received at the system input pins is supplied without modifications to the on-chip system logic, thus data from the on-chip system logic is driven without modification through the system output pins.

PRELOAD instruction: This allows scanning of the boundary scan register without causing interference with the normal functioning of the system logic. The initial data patterns are to be placed at the latched parallel outputs of boundary scan register cells before the selection of another test operation. Data received at the system output pins will be supplied till the output pins without any modifications.

EXTEST instruction: This allows circuitry external to the component package (board interconnect) to be tested. It also allows testing of blocks of components that do not themselves incorporate boundary scan registers. The data shifted out of the component in response to this instruction is not altered by presence of faults in the system logic.

INTEST instruction: This is an optional instruction that allows slow speed testing of the system logic with each test pattern and response being shifted through the boundary scan register. This requires the on-chip system logic to be operated in a single step mode, where the circuitry moves one step forward in its operation.

CLAMP instruction: This is an optional instruction that allows the state of signals driven from component pins to be determined from the boundary scan register, while the bypass register is selected as the serial path between TDI and TDO.

IDCODE instruction: This instruction is put into the instruction register's parallel output latches during TLR controller state. Permits blind interrogation of the components assembled on the PCB. Its purpose is to verify that the device at a board location matches the boundary scan description language used to generate the test. If the value read during the test does not match the expected

device identification register result that could mean that the device is not compatible with the pin layout configuration.

USERCODE instruction: This allows a user-programmable identification code to be loaded and shifted out for examination. It is required only for programmable components. It allows the programmed function of the component to be determined and verified.

HIGHZ instruction: This is an optional instruction that places a component in a state in which all its system logic outputs are placed in a high impedance state.

Test Data Registers:

- The group of test data registers includes the following:

1. BR
2. BSR
3. Device Identification Register.
4. Electronic chip Identification Register.

The registers numbered as 1 and 4 are optional or miscellaneous registers.

BR, BSR and optional test data registers can be realized as a set of shift register based elements connected in parallel between a common serial input and a common serial output. Selection of a register that forms a serial path between the TDI and TDO is controlled by the instruction register.

Design and construction of DR (Data Register):

1. Each DR shall be given a unique name.
2. The DR should be designed such that, when data are shifted through it, data applied to TDI appears without inversion at TDO after multiple TCK cycles.
3. The length of each of each DR shall be fixed, independent of the instruction by which the DR is controlled.
4. All the registers should have a minimum length of 1.
5. Each DR cell shall be able to respond to TAP controller being compliant with the standard.

Figure 8 shows a test data register.

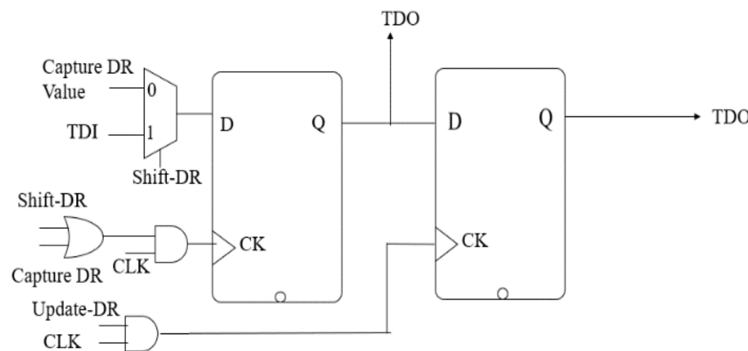

Figure 8: Test Data Register.

Working:

If CLK(Clock)=1, Update DR=0, Capture DR=0, Shift DR =1, TDI will be directly shifted towards TDO.

If CLK=1, Update DR=0, Shift DR=0, Capture DR=1, Capture DR state is activated, the data will be loaded and captured.

If CLK=1, Update DR=1, Shift DR=0, Capture DR=0, Update DR state is activated, the captured value will be updated and available at TDO.

Types of Test Data Registers:

1. **BR:** It provides a minimum length serial path for the movement of test data between TDI and TDO and consists of a single shift register stage. This register allows bypassing of segments of the board level serial test data register that are not required for specific test. The circuitry used to implement the BR shall not be used to perform the any other system function. When Bypass Register is selected for inclusion in the serial path between TDI and TDO by the current instruction, the shift register stage shall be 0 on rising edge of TCK in capture -DR stage. Figure 9 shows a BYPASS Register.

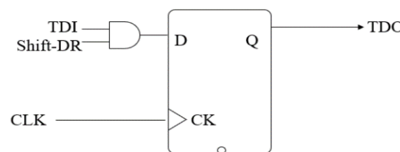

Figure 9: BYPASS Register.

2. **BSR:** Formed by a combination of Boundary Scan cells. It allows testing of circuitry external to the component along with the system signals flowing into and out of system logic to be sampled and examined without causing interference with the normal operation of system logic. It is the most complex register in the way register is connected to the on-chip system logic and its response to the instructions. Each BSR cell contains a single shift register stage and shall have a serial input and serial output terminal by which the cell is linked to the logic. BSR is selected as the serial path between TDI and TDO in the shift-DR controller state the data entered at TDI will appear without inversion at TDO after certain clock pulses. Figure 10 shows a Boundary scan cell.

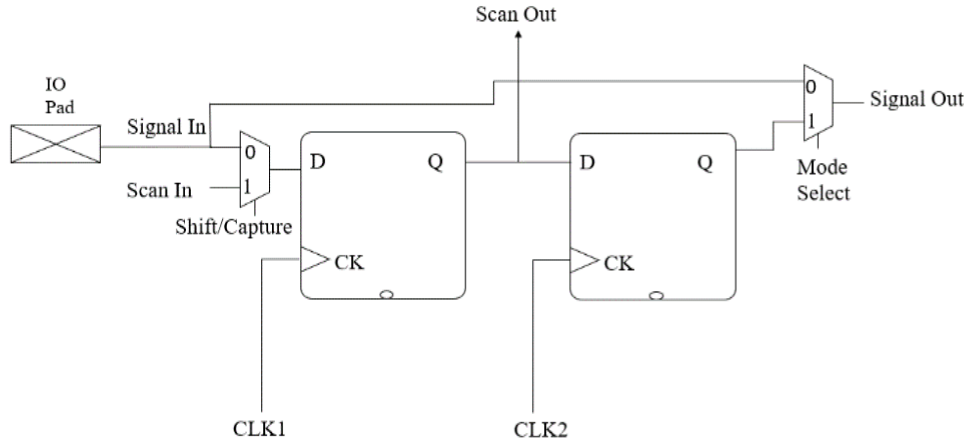

Figure 10: Boundary Scan Register Cell.

Pads:

They are present at the periphery of any chip and are used by a microprocessor to interact with the outer world (power supply source or IP (Intellectual Property) of another microcontroller etc.). Communication between the IP's can be controlled by controlling the interaction of interconnects. They are organized into a rectangular frame of metals & controlled by enable signals. The input/output pads are spaced with a pad pitch. They are used in the design as voltage level shifters are a part of their circuitry and this circuitry is used because core and peripheral chips might be working on different voltage levels.

So, we need voltage level shifters so that all required signals can be shifted to the required voltage level for the operation of microprocessor, if these level shiftings are not done it will impact the reliability of the device. Moreover, during handling and connections there might be ESD (Electrostatic Discharge) through human touch or any other sudden electric surge event which can cause a large amount of current to flow through the devices which could permanently damage them, and this can be prevented by ESD protection circuitry present in the pads.

Structure of Pads: It includes the: Bonding Pad (Area in which the bond wire is soldered. The wire goes from bonding pad to chip pin)., ESD Protection Circuitry (Consist of a pair of PMOS, NMOS in a reverse biased diode structure). Driving and Logic circuitry.

Types of Pads: According to logic directions: There are three types of pads.

1. *Input Pad:* They receive input signals from external sources.
2. *Output Pad:* They receive output signals from external sources.
3. *Bidirectional Pad:* They are interfaces that allow for the bidirectional transfer of signals and can handle both input and output operations on the same pin.

IO Pad Organization: It consists of a square of top-level metal that is soldered to bond wire connecting to a package. Pad refers to the complete I/O Cell containing the metal, ESD protection circuit, I/O Transistors along with built in receiver & driver circuits to perform level conversion & amplification. Figure 11 shows the IO Pad Organization in a design.

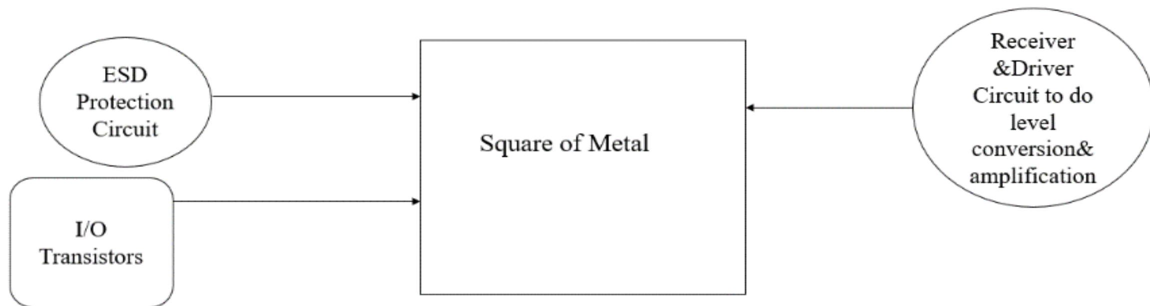

Figure 11: IO Pad Organization.

Block diagram of IO Pads:

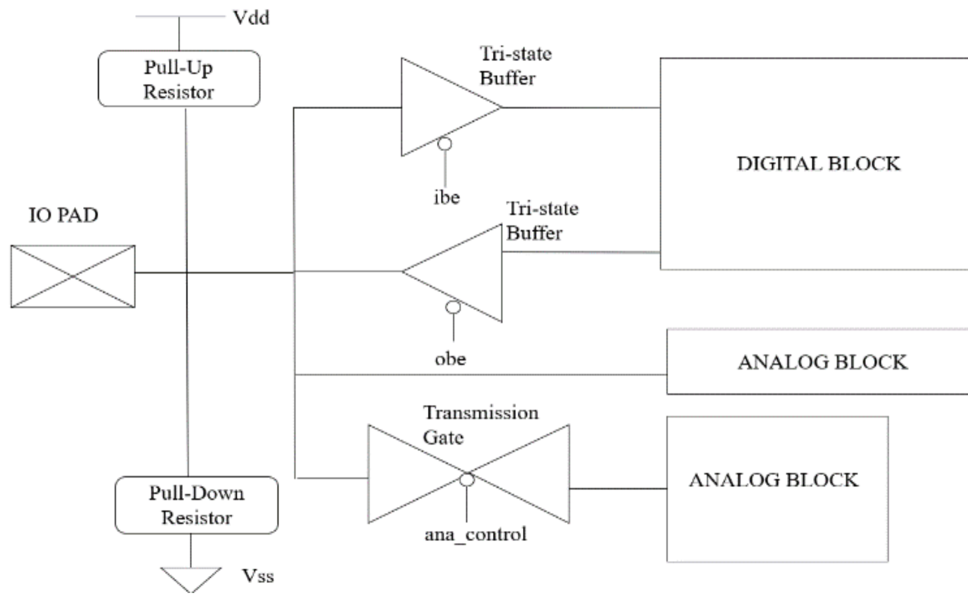

Figure 12: Basic Block Diagram of IO Pad.

Description:

- Analog Block includes ADC (Analog to Digital Converter) and PLL (Phase Lock Loop).
- Digital Block includes communication IP's and cores of microprocessors.
- Obe: Output buffer enable (output with enable signal).
- Ibe: Input buffer enable (input with enable signal).
- Tri-state buffers: Have an enable signals which blocks or enables communication.
- Pull-up Resistor: They are the resistors connected between the voltage supply and the IO Pad. This pulls the signal value to a high state (1).
- Pull-down Resistor: They are the resistors connected between the ground supply and the IO Pad. This pulls the signal value to a low state (0).
- To communicate with the digital block of IP, the connection of I/O pad to the digital block is gated through tri-state buffers. The output of core or blocks are considered as inputs for the pads, and the outputs of pad are considered as input for blocks.
- The device in which pads are inserted may act as analog or digital device at a time, not both simultaneously. We need to ensure that whenever analog block is enabled, the digital block is disconnected and vice versa. If both the blocks are enabled simultaneously then we run into contention and the signals will get corrupted. Transmission gates are used with analog blocks because they require continuous data for working which is provided by transmission gates only. Transmission gates are controlled by ana_control. There is an uncontrolled path to the analog block, thus whatever appears at the IO Pad is directly available at the analog block.
- The pull down and pull up resistors are used to avoid the floating or fluctuation of states among the three states 0,1, Z that can be acquired by the pin of a microcontroller.

Table 15: Truth Table of Output Buffer Enable.

Obe	Pad Port/Output
1	Digital Block drives the IO Pad.
0	Undriven.

Table 16: Truth Table of Input Buffer Enable.

Ibe	Pad Port/Output
1	IO Pad drives the Digital Block.
0	Undriven.

Table 17: Truth Table of Transmission Gate.

ana_control	Pad Port/Output
1	Bidirectional connection between IO Pad and analog blocks.
0	Undriven.

2. Materials and methods

1. Studying the Internal Logic of GPIO and Presence of Pull-Up & Pull-Down in GPIO:

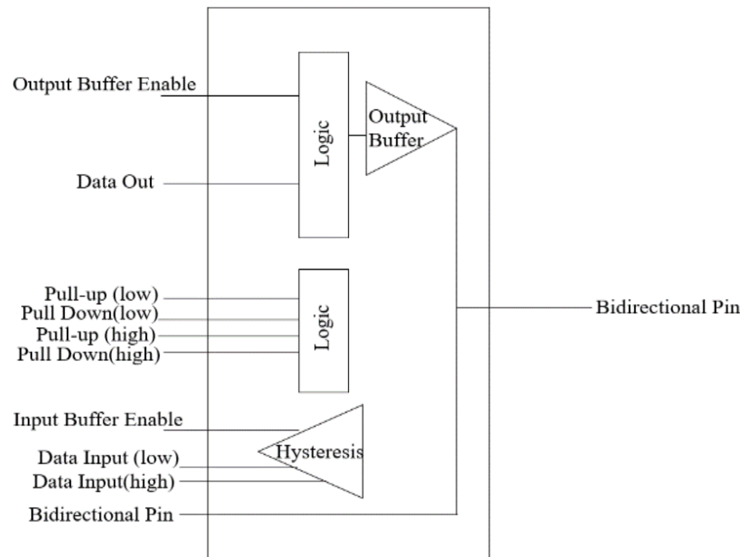

Figure 13: GPIO Symbolic Diagram.

Reading the Verilog file of the design of GPIO (General-purpose IO), Figure 14 can be interpreted for the logic blocks of GPIO.

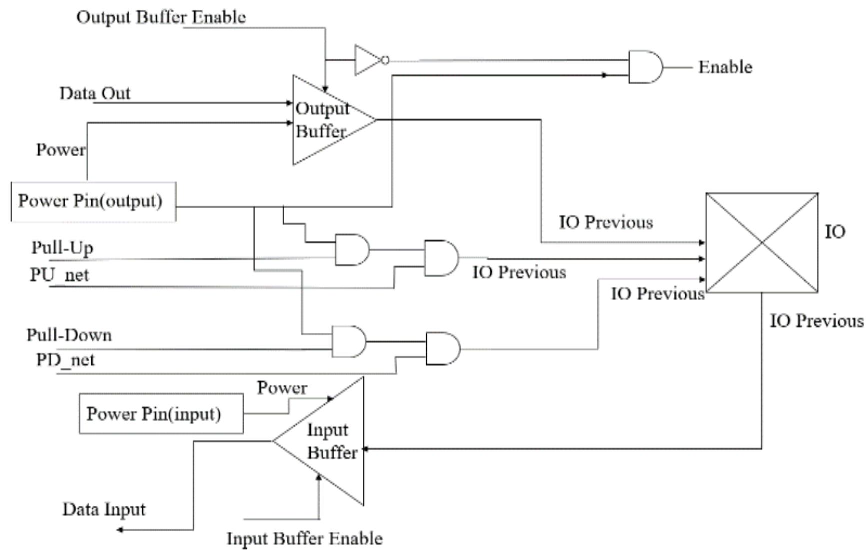

Figure 14: Internal Circuit of Logic Blocks of GPIO.

In Figure 14, If output buffer enable is high, Output Buffer is active, Input Buffer is inactive, Enable=0.The output data from the core of the design (data out) will be available as IO previous to the pad. Power pin (output) is providing power to the output buffer.

As either pull up (PU) or pull down (PD) will be active at a time Thus two cases arise:1)PU=1, PU_net=1 and power pin(output) is providing the power then IO Previous will be available to the pad and the floating pins will get the high value as pull up is active.2)PD=1, PD_net=1 and power pin(output) is providing the power then IO Previous will be available to the pad and the floating pins will get the low value as the pull down is active. If input buffer enable is high, input Buffer is active, output Buffer is inactive. The output data from the IO will be available to the core of the design as Data Input to the core and Power pin (input) is providing power to the input buffer.

2. Applying the Pads with the Data Register:

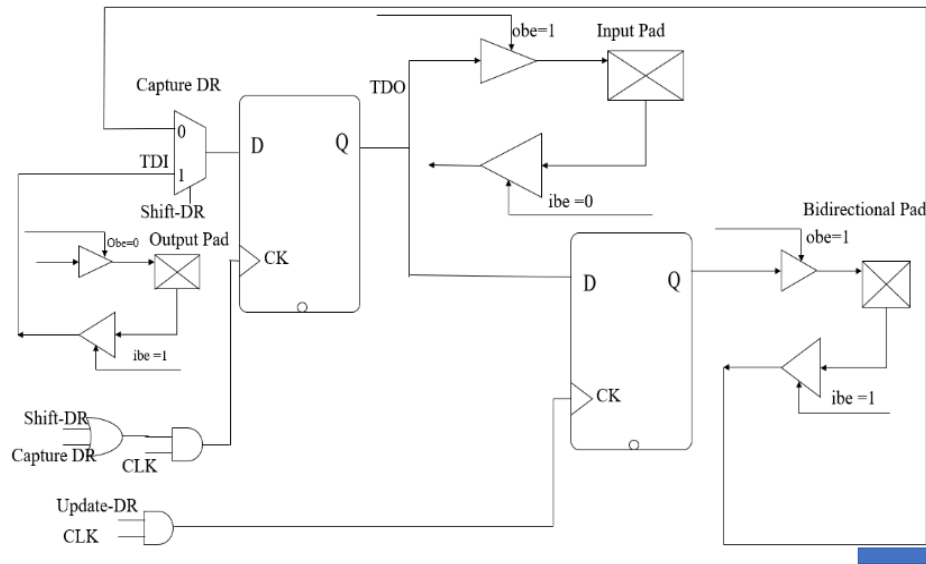

Figure 15: Combining IO PADS with Data Register.

If Shift-DR=1(shift stage), CLK=1: The TDI will be shifted in the register and available at TDO and is provided further to the Input pad. For TDI to be available, ibe=1.The IO pad is used as an Output pad.

If Shift-DR=0 (Capture stage), CLK=1.

If Update-DR=1 (Update Stage), CLK=1: The value present at TDO will be updated in this stage and the output will be provided to the Bidirectional pad with both the enable pins as 1. The pad output will be fed as Capture data at capture stage.

3. **Spyglass:** Spyglass [3] is a widely used structural analysis tool provided by Synopsys, a leading electronic design automation company. It helps in the static analysis and verification of digital designs at the RTL (Register Transfer level). The tool performs a variety of checks on the design, including linting checks, clock domain crossing analysis, FSM (Finite State Machine) analysis, coding style checks, DRC (Design Rule Check) and many other structural analysis tasks. It assists designers in identifying potential design issues, improving design quality, and ensuring adherence to design guidelines and industry standards.

This tool was used to trace the pins of microcontroller to identify the presence of BSR in the design of microcontroller. The results are shown in the results section of the article.

After identifying the existence of BSR, the compliance was proved for the same design by using the tool Design Compiler.

4. ***Design Compiler***: [4] It is a popular synthesis tool used in digital IC (Integrated Circuit) design. It takes an HDL (Hardware Description Language) code, such as VHDL or Verilog, and converts it into a gate-level representation suitable for physical design and manufacturing. Once the synthesis process is complete, it generates a gate-level netlist in a format suitable for downstream physical design tools. It may also produce various reports, such as area estimation, power consumption, and timing analysis reports.

3. Results and discussion

The following are the results drawn from tracing the pins by the tool Spyglass.

- Tracing the output pin:

Figures 16 and 17 shows the entire path traced for the output pin of the microcontroller to identify the location of the BSR.

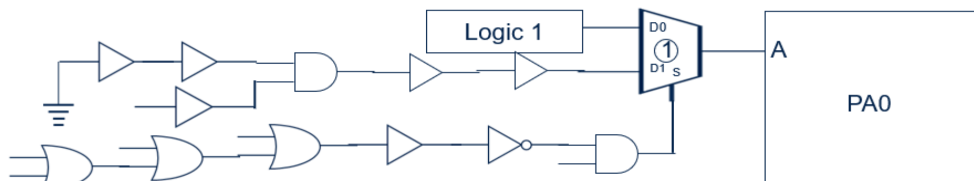

Figure 16: Components Immediately after Output pin.

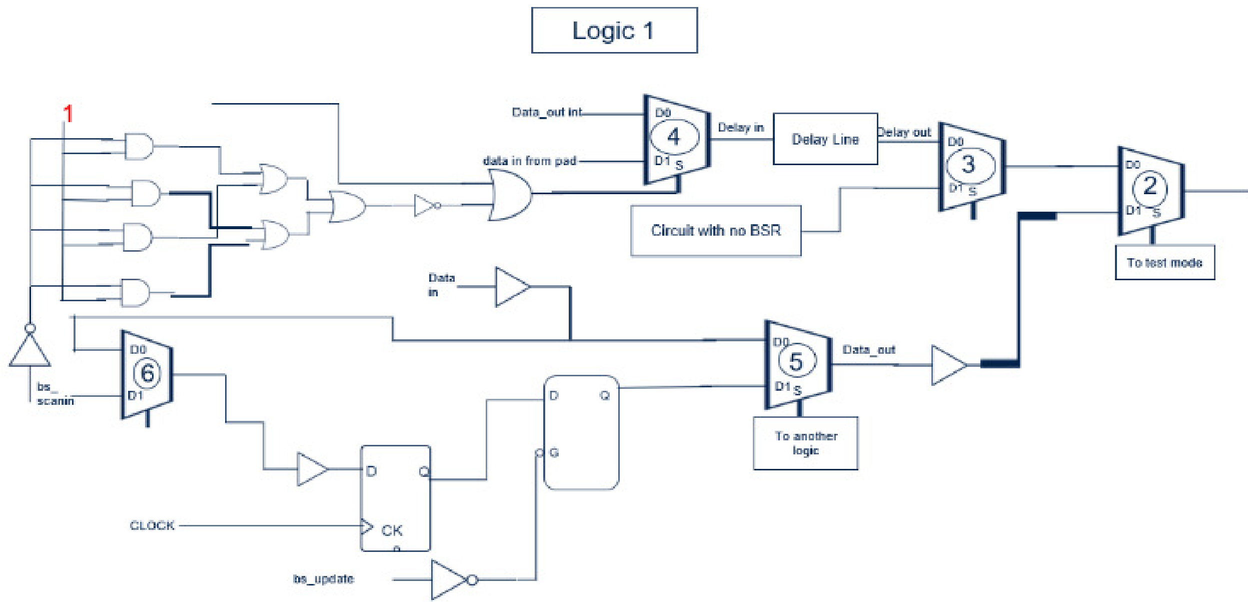

Figure 17: Identifying the BSR in Logic1 after the Output pin.

- Tracing the Data Input (high) pin. This pin, when traced showed that it is finally merging into the path of the data out pin and thus BSR is also available in that path only.

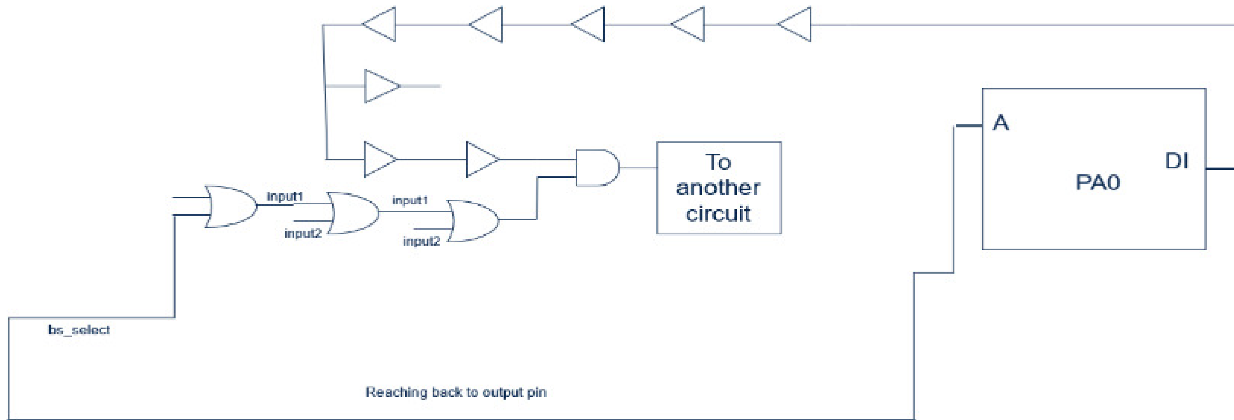

Figure 18: Tracing the path of Data Input(high) pin.

- Tracing the Data Input (low) pin

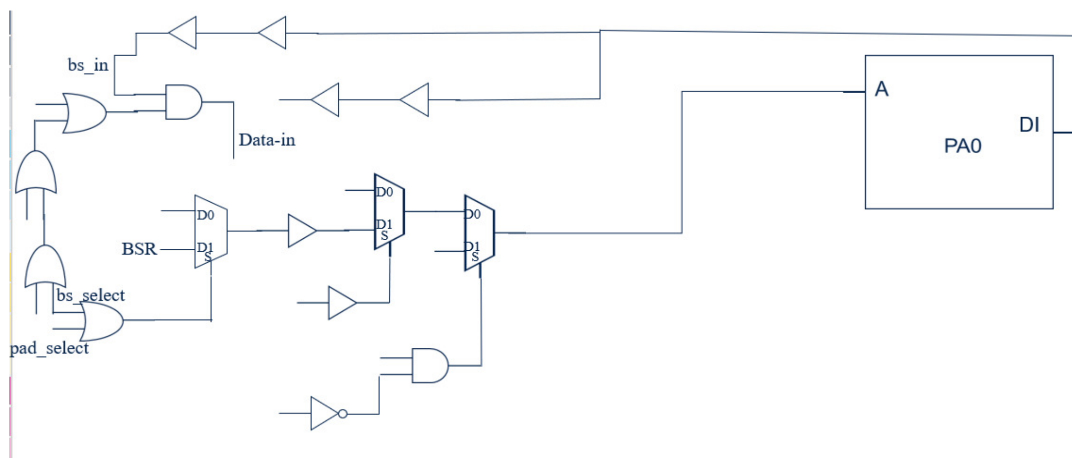

Figure 19: Tracing the path of Data Input(low) pin.

- Tracing the Pull -down (low) pin.

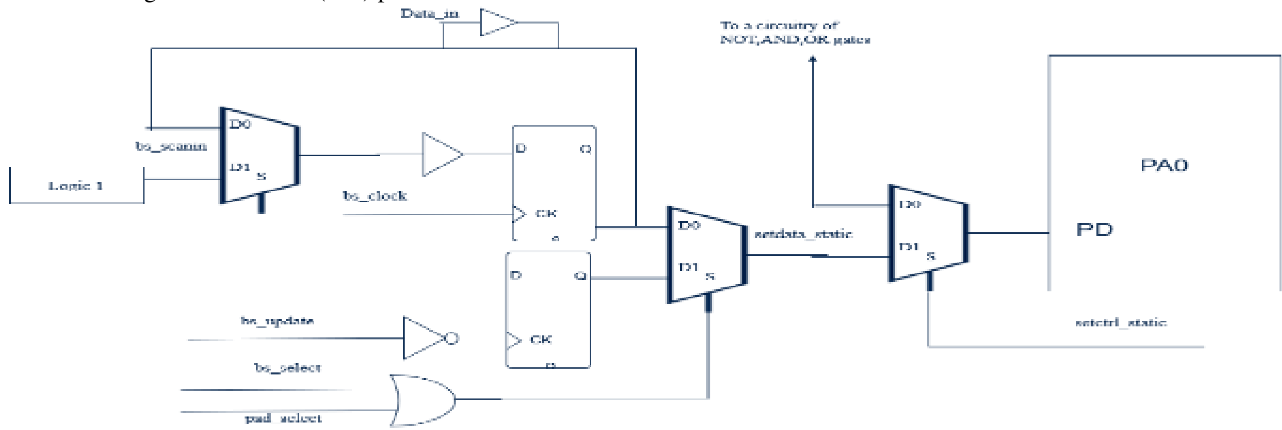

Figure 20: Tracing the pull-down(low)pin and identifying the BSR.

- Tracing the Pull -down (high) pin.

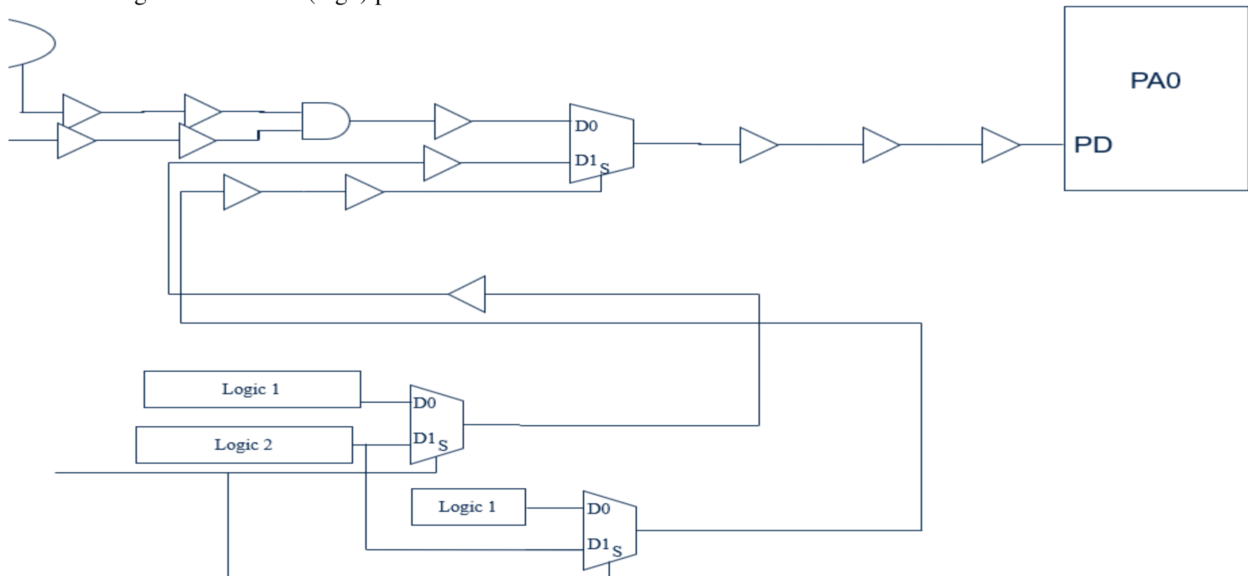

Figure 21: Components Immediately after pull -down (high) pin.

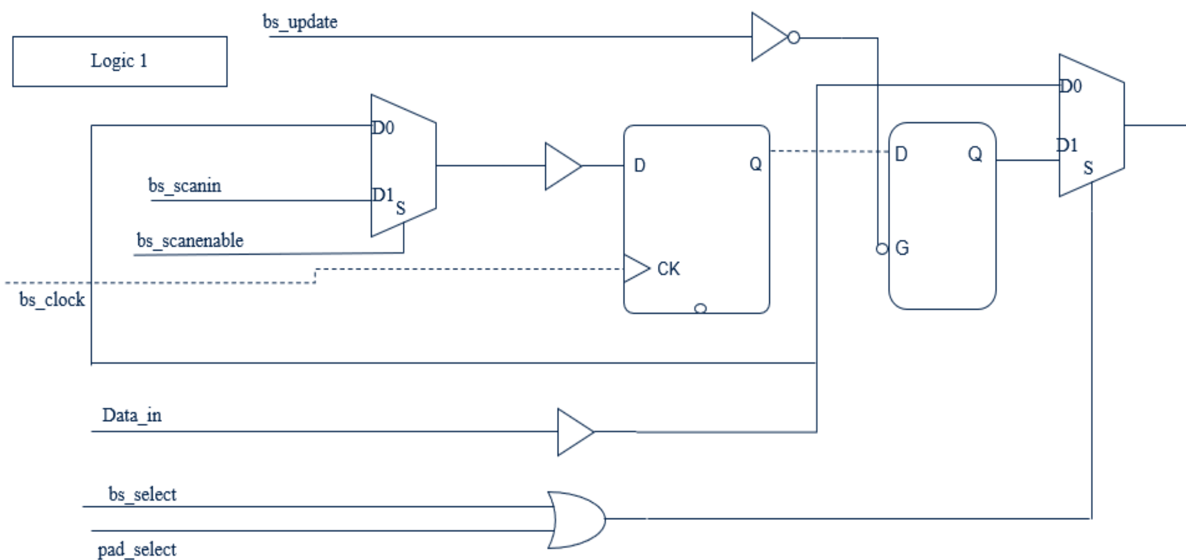

Figure 22: Elaborating the Logic1 to identify the presence of BSR.

- Tracing the Pull -up (low) pin.

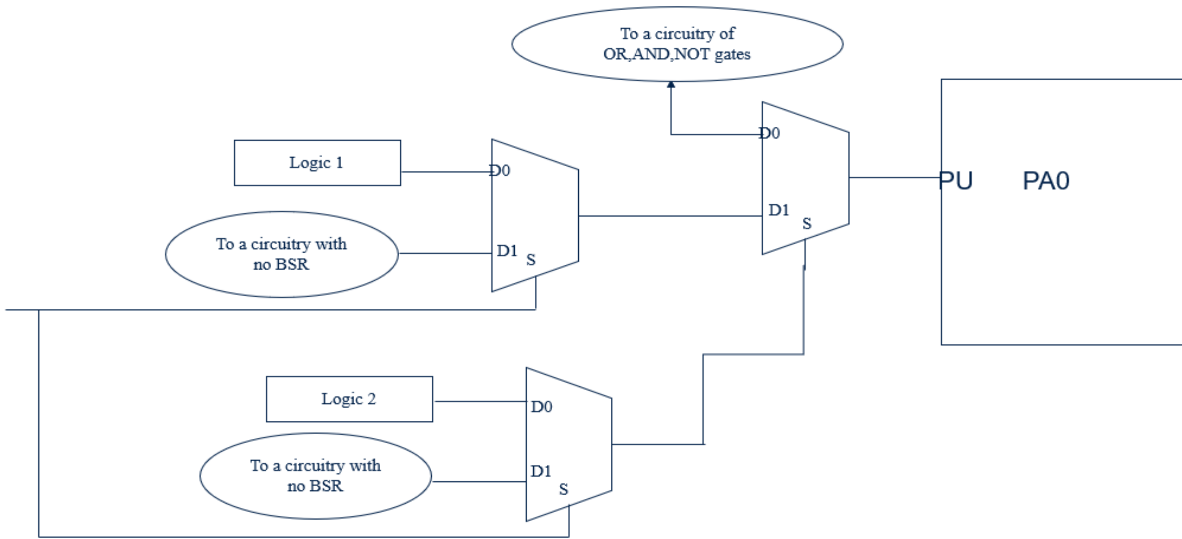

Figure 23: Components Immediately after pull-up (low) pin.

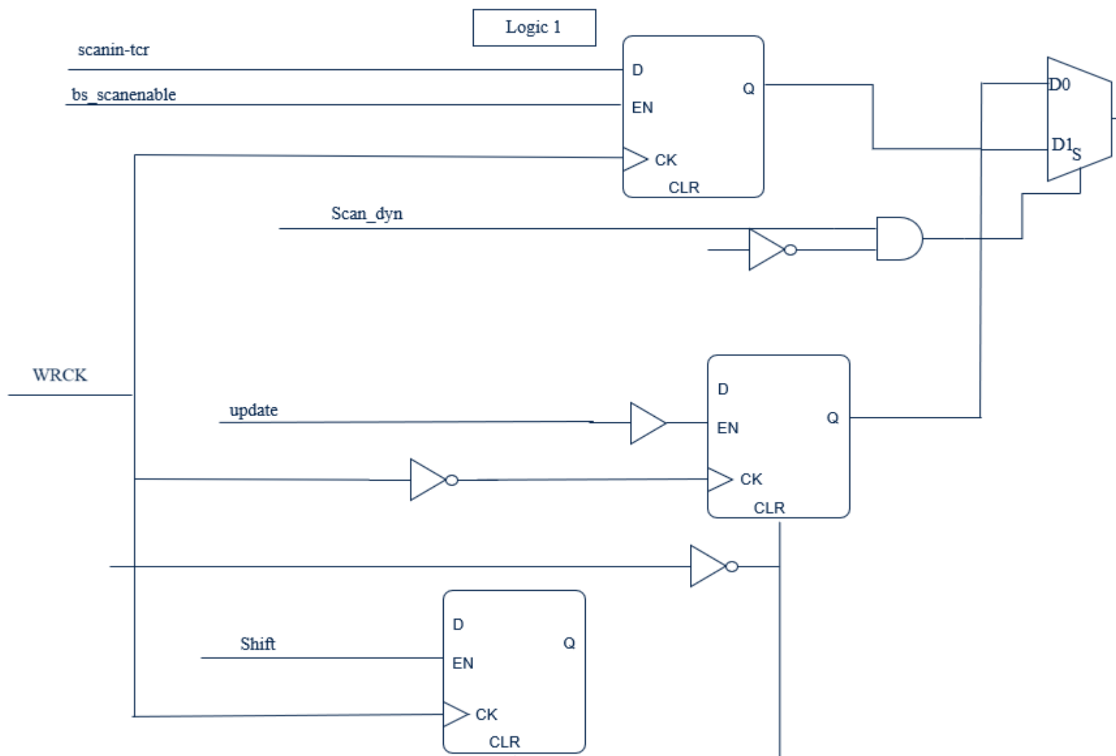

Figure 24: Elaborating the Logic1 present in path of pull-up(low) pin.

Figure 25: Elaborating the Logic2 to identify the location of BSR.

- Tracing the Pull -up (high) pin.

Figure 26: Components Immediately after pull-up (high) pin.

Figure 27: Elaborating the Logic1 to identify the location of BSR.

- Tracing the Input Buffer Enable pin.

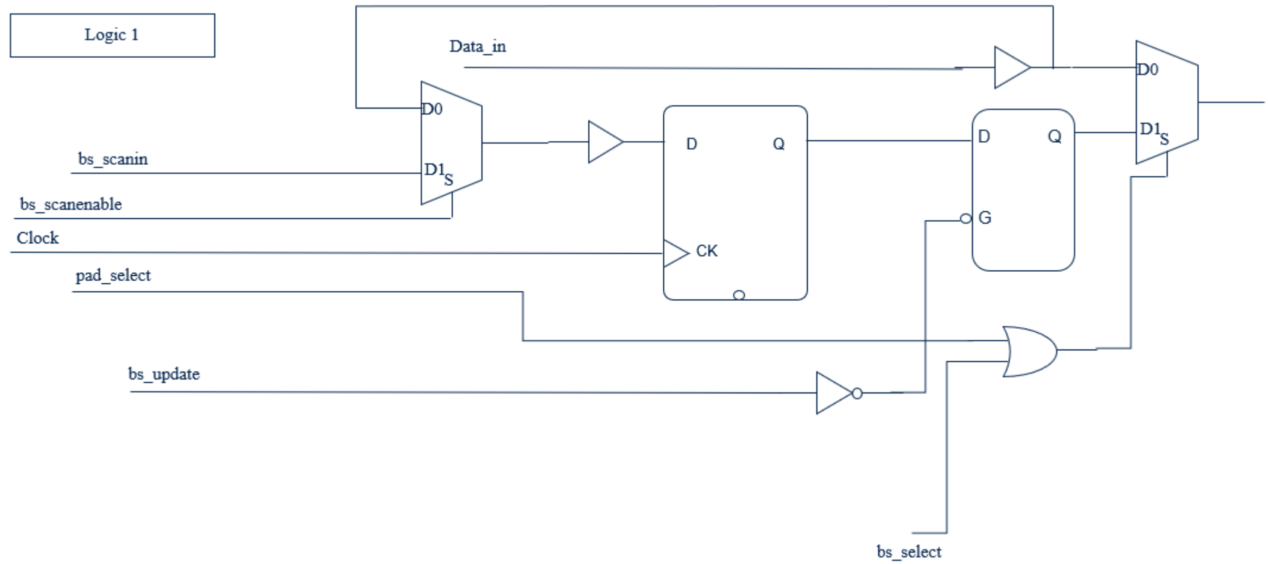

Figure 28: Components Immediately after Input Buffer Enable pin.

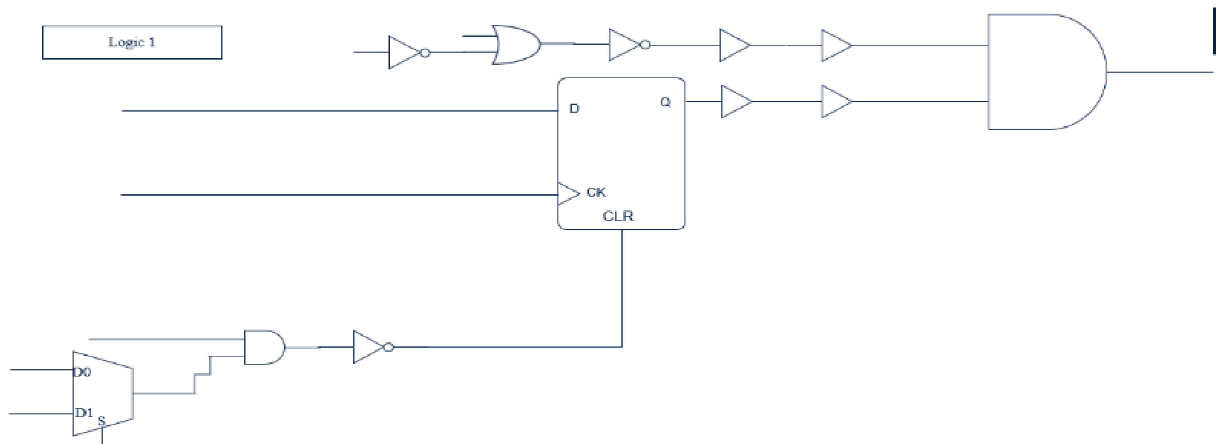

Figure 29: Elaborating the Logic1.

No BSR is available in the path of this pin, here all the signals to the various components have a constant value.

- Tracing the Output Buffer Enable pin.

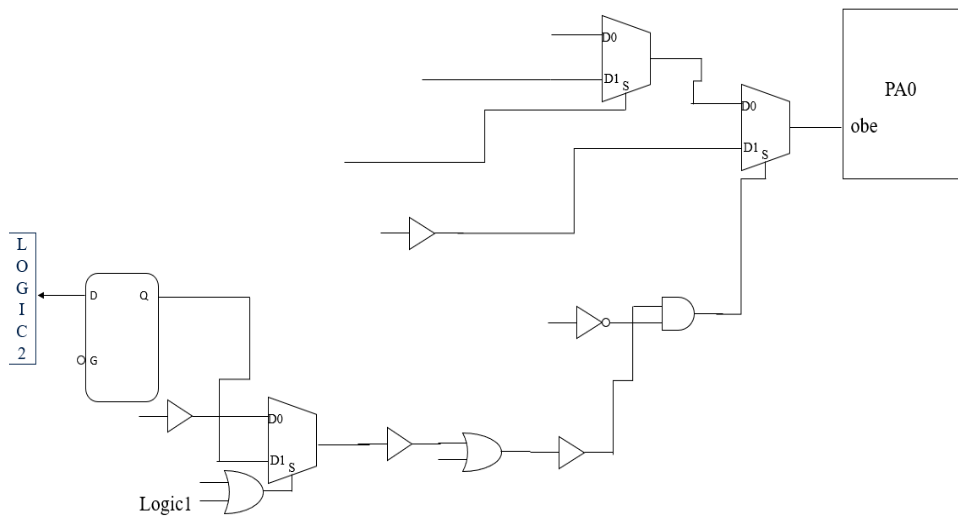

Figure 30: Components Immediately after Output Buffer Enable pin.

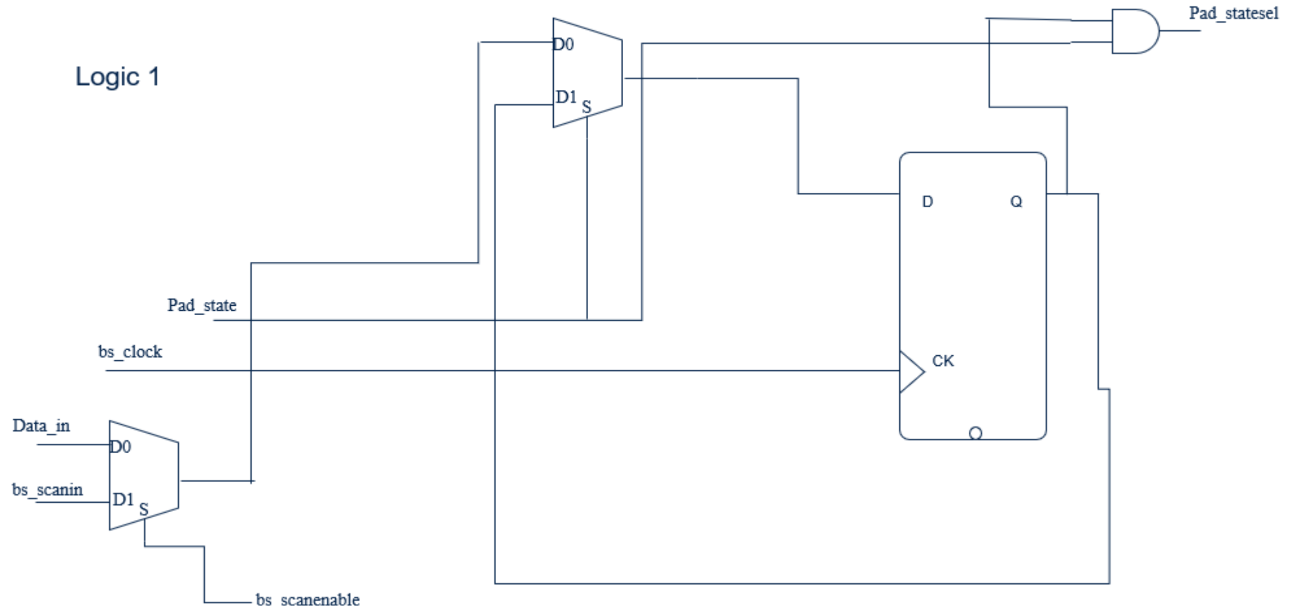

Figure 31: Elaborating the Logic1.

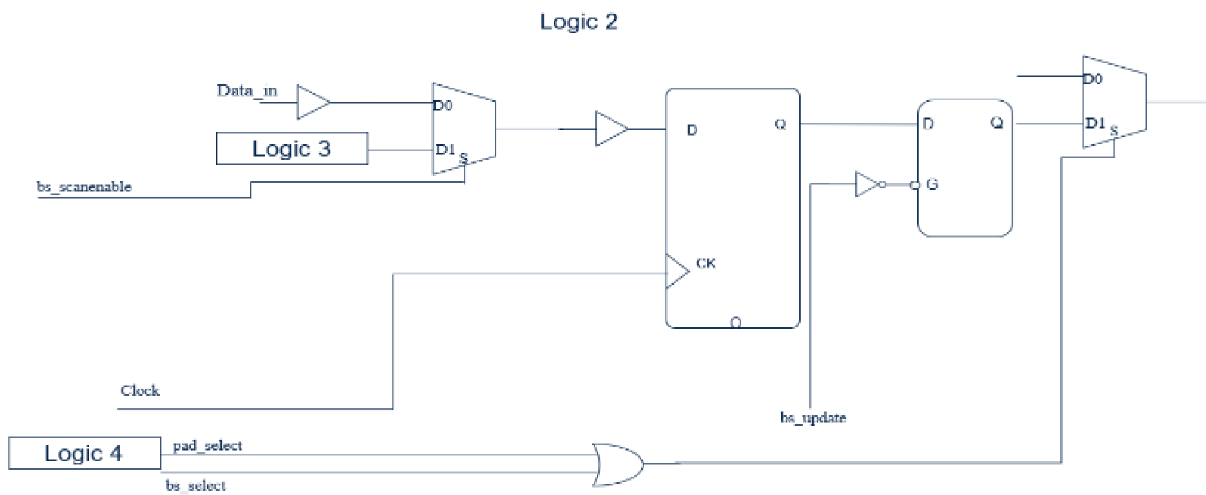

Figure 32: Elaborating Logic2 to identify the location of BSR.

The following (Fig. 33, 34, 35, 36) show the successful compliance of a sample testcase with the IEEE Standard 1149.1 by using the tool Design Compiler.

```
set_dft_configuration -scan disable -bsd enable
Accepted dft configuration specification.
1
##desing configuration for 1149.1, optional
set_bsd_configuration -std {ieee1149.1_2001}
Accepted bsd configuration specification.
1
#Specify TAP ports using set_dft_signal
# tdi tdo tms tck trst_n
set_dft_signal -type tdi  -port tdi
Accepted dft signal specification for modes: all_dft
1
set_dft_signal -type tdo  -port tdo
Accepted dft signal specification for modes: all_dft
1
set_dft_signal -type tck  -port tck
Accepted dft signal specification for modes: all_dft
1
set_dft_signal -type tms  -port tms
Accepted dft signal specification for modes: all_dft
1
set_dft_signal -type trst -port trst_n
Accepted dft signal specification for modes: all_dft
1
```

Figure 33: Acceptance of commands for TAP ports by DC Shell.

```
write_bsdl successful.
1
check_bsd -verbose
...Starting IEEE 1149.1 Compliance Checking.
...Finding set of sequential elements.
...Analyzing TAP and TAP Controller.
......Analyzing TAP.
......Finding the set of TAP controller sequential elements.
......Pruning set of TAP Controller sequential Elements
...Checking the TAP controller initialization.
...Analyzing the TAP controller reset condition.
...Traversing the TAP controller states.
...Inferring TAP controller clock outputs.
...Analyzing TRST port.
Information: Inferred pullup by traversal.
...Analyzing TMS Port.
Information: Inferred pullup by traversal.
...Analyzing TCK halt state.
Information: Inferred pullup by traversal.
...Analyzing the instruction register.
......Finding the update flops.
...Analyzing the BYPASS register.
...Analyzing the DIR register.
Device Identification Register doesn't exists
...Analyzing the boundary scan register.
......Finding the update flops.
......Finding the BSR cells controlling the design ports.
......Finding the BSR cells sensing the design ports.
......Finding the BSR cells driving the design ports.
...Using the implemented instructions information.
```

Figure 34: The successful start of compliance check.

```
****************************************************
            IEEE 1149.1 Summary
****************************************************
  Test Logic Reset Method: Synchronous and Asynchronous(TRST)

  16 state elements found in the TAP controller
  TOP_DW_tap_inst/U1_current_state_reg_0_
  TOP_DW_tap_inst/U1_current_state_reg_10_
  TOP_DW_tap_inst/U1_current_state_reg_11_
  TOP_DW_tap_inst/U1_current_state_reg_12_
  TOP_DW_tap_inst/U1_current_state_reg_13_
  TOP_DW_tap_inst/U1_current_state_reg_14_
  TOP_DW_tap_inst/U1_current_state_reg_15_
  TOP_DW_tap_inst/U1_current_state_reg_1_
  TOP_DW_tap_inst/U1_current_state_reg_2_
  TOP_DW_tap_inst/U1_current_state_reg_3_
  TOP_DW_tap_inst/U1_current_state_reg_4_
  TOP_DW_tap_inst/U1_current_state_reg_5_
  TOP_DW_tap_inst/U1_current_state_reg_6_
  TOP_DW_tap_inst/U1_current_state_reg_7_
  TOP_DW_tap_inst/U1_current_state_reg_8_
  TOP_DW_tap_inst/U1_current_state_reg_9_
```

Figure 35: Summary of compliance check for the sample testcase.

```
            IEEE 1149.1 Violation Summary

****************************************************
check_bsd successful.
1
#create BSD patterns using create_bsd_patterns
create_bsd_patterns -type all
.....Generating vectors to test the asynchronous test logic reset
.....Generating vectors to test the synchronizing sequence of 5 1's on tms
.....Generating vectors to test the TAP FSM
.....Generating vectors to test boundary scan instructions
.......Generating vectors to test the 'HIGHZ' instruction.
.......Generating vectors to test the 'BYPASS' instruction.
.......Generating vectors to test the 'EXTEST' instruction.
.......Generating vectors to test the 'SAMPLE' instruction.
.......Generating vectors to test the 'PRELOAD' instruction.
.......Generating vectors to test the 'CLAMP' instruction.
.....Generating vectors to test the boundary scan register
.....Generating vectors to perform leakage test.
Created bsd patterns.
1
```

Figure 36: The compliance is proved for the sample testcase.

Table 18: Description of commands used for proving the compliance.

Commands used	Description
read_verilog	Reads the design or library files in Verilog format.
current_design	Sets the working design.
link	Links all the library components and designs for the completion and functionality of a particular design.
set_dft_configuration -scan disable -bsd enable	Sets the DFT configuration for the current design with disabled scan insertion& enabled insertion of IEEE Std 1149.1compliant boundary-scan circuitry.
set_bsd_configuration -std{ieee1149.1_1993}	The boundary scan design configuration is compliant to IEEE Std 1149.1.

set_dft_signal -type tdi -port tdi	Specifies the DFT signal types tdi specifies the TDI port of TAP controller.
set_dft_signal -type tdo -port tdo	Specifies the DFT signal types tdo specifies the TDO port of TAP controller.
set_dft_signal -type tck -port tck	Specifies the DFT signal types tck specifies the TCK port of TAP controller.
set_dft_signal -type tms -port tms	Specifies the DFT signal types tms specifies the TMS port of TAP controller.
set_dft_signal -type trst -port trst_n	Specifies the DFT signal types trst specifies the TRST port of TAP controller.
set_bsd_instruction (instruction name) -code [list (opcode of the instruction)] -reg (Register for the instruction)	Specifies boundary-scan instructions (EXTEST, SAMPLE, CLAMP, HIGHZ) along with the opcode for the instruction and the targeted register for the instruction.
set_bsd_linkage_port -port_list {dummy}	Specifies the linkage port (port on which we don't want to insert boundary scan).
preview_dft -bsd all	It reports detailed information about one or more sections of the boundary-scan architecture.
insert_dft	Inserts DFT logic in the current design.
read_pin_map	Reads in a port-to-pin mapping file, which defines the design port-to-package pin mapping for a boundary-scan design.
set_bsd_configuration -default_package "name of package"	Specifies the boundary-scan configuration for a design also specifies the default package for a boundary-scan design. The package_name value must correspond to one of the packages for the design read in using the read_pin_map command.
write_bsdl	It generates the BSDL output file for the boundary-scan design.
check_bsd -verbose	Checks whether a design's boundary-scan implementation is compliant with IEEE Std 1149.1.
create_bsd_patterns -type all	Generates all the functional test patterns for a boundary-scan design.

The results showcase the successful compliance of the design with the IEEE 1149.1 Standard. The following Table 18 lists the commands used in the task.

4. Conclusions

The study gives a better understanding of the concepts of Design for Testability and boundary scan along with an in depth understanding of the IEEE 1149.1 JTAG Standard. The reader will get an insight about tools like Design Compiler, Spyglass.

Acknowledgements

The author is highly thankful to the faculties of ECED, T.I.E.T Patiala for their valuable guidance and encouraging me. I would also like to thank my mother and my friends who always supported me.

References

[1] Design For Testability, VLSI Test Principles and Architectures edited by Laung-Terng Wang Cheng-Wen Wu Xiaoqing Wen. IEEE Standard

[2] IEEE Standard for Test Access Port and Boundary-Scan Architecture.

[3] About the Spyglass Tool: Synopsys SpyGlass Products

[4] About Design Compiler Tool: Design Compiler (synopsys.com)

Publisher's Note: Research Plateau Publishers stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.