Introduction

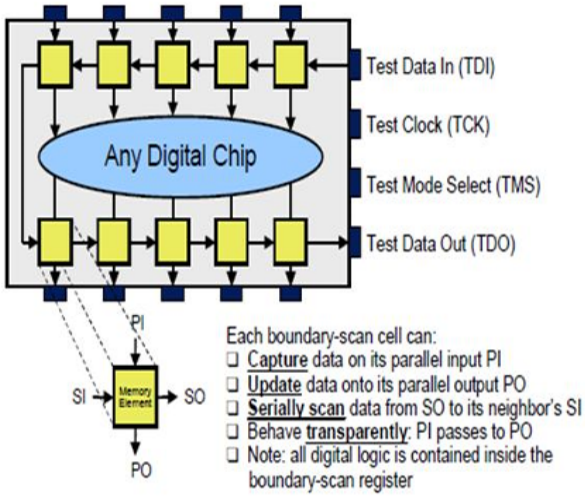
IEEE 1149.1: IEEE Standard for Test Access Port and Boundary-Scan Architecture

Design for Test (DFT) incorporates rules and techniques in product design to facilitate testing.

Structured DFT impacts all product phases, from circuit design to field service, managing complexity, minimizing development time, and reducing manufacturing costs.

Testing involves controllability (establishing a known state, supplying test data) and observability (checking if the system performs as designed).

Considering testing at the chip design level enables benefits at all electronic assembly levels.



IEEE 1149.1 Standard provides the foundation of embedded test technology in integrated circuits, facilitating test and programming capabilities.

Boundary Scan Register (BSR) is a serial scan path intercepting signals between core logic and pins, facilitating test mode operations.

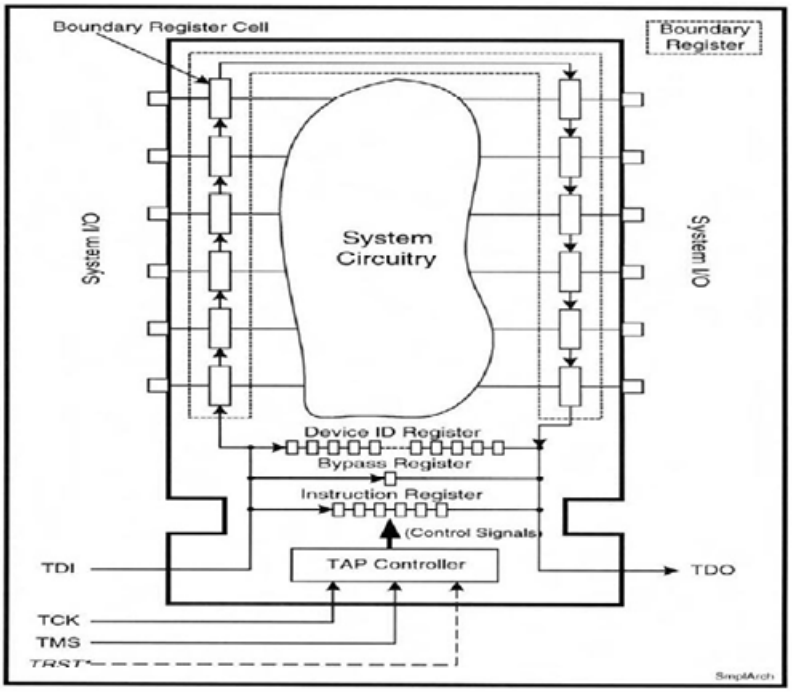


Fig. 3: Basic Boundary Scan Architecture.

There are various types of Boundary Scan Cells defined by the 1149.1 standards, with manufacturers able to define non-standard cell types to match hardware functionality.

Interface Signals

The JTAG interface, collectively known as a Test Access Port, or TAP, uses the following signals to support the operation of boundary scan.

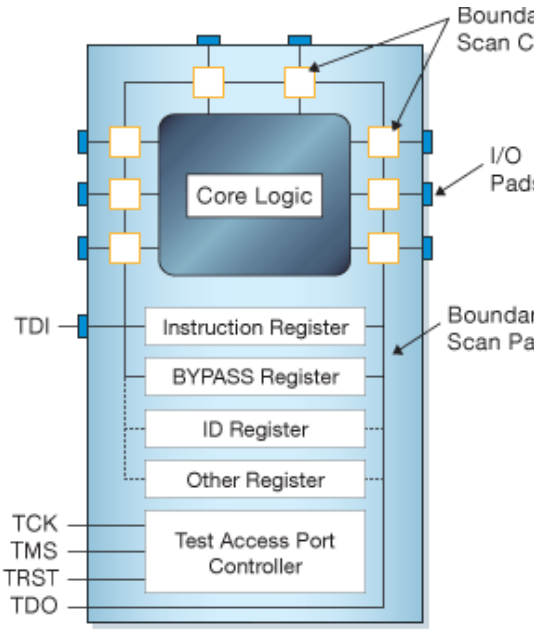


Fig1. JTAG TAP controller

TCK (Test Clock) – this signal synchronizes the internal state machine operations.

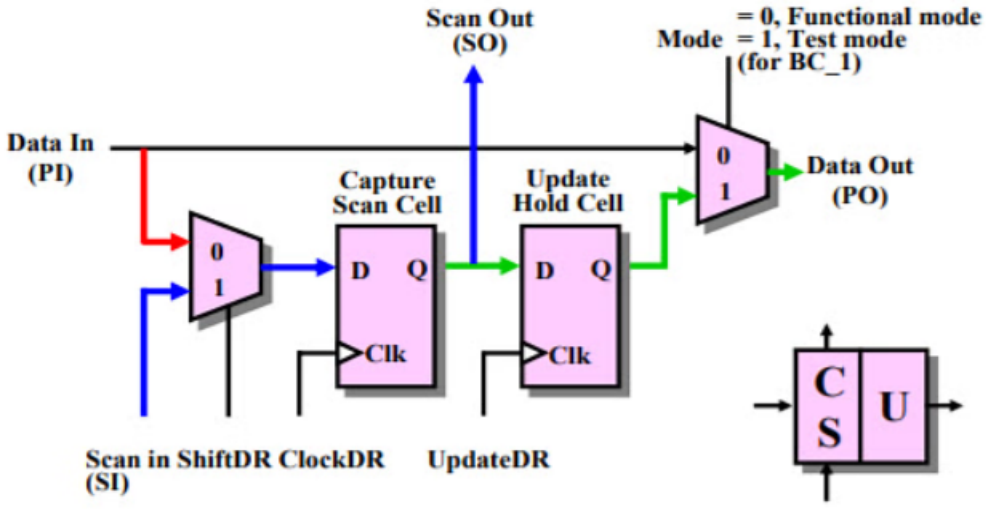
TMS (Test Mode Select) – this signal is sampled at the rising edge of TCK to determine the next state.

TDI (Test Data In) – this signal represents the data shifted into the device's test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.

TDO (Test Data Out) – this signal represents the data shifted out of the device's test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state.

TRST (Test Reset) – this is an optional pin which, when available, can reset the TAP controller's state machine.

Boundary Scan Cell



Boundary scan is the application of a scan path at the boundary (I/O) of ICs to provide controllability and observability access via scan operations. Figure 3-1 shows an IC with an application-logic section and related input and output, and a boundary-scan path consisting of a series of boundary-scan cells (BSCs), in this case one BSC per IC function pin.

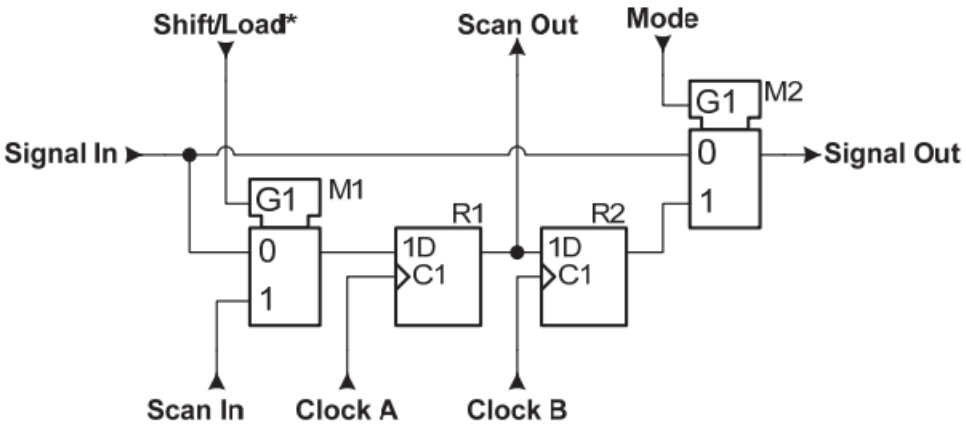
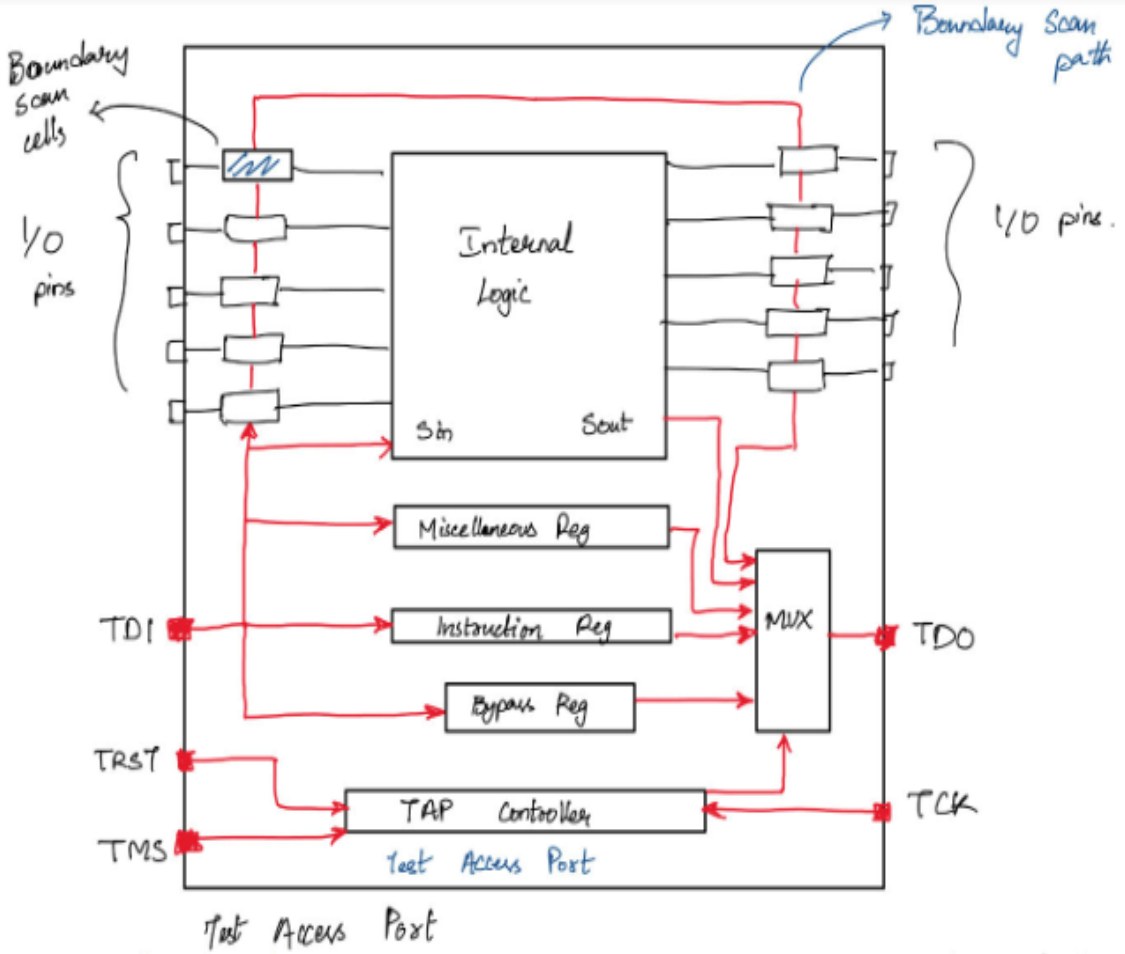


Figure 1-1—Boundary-scan register cell



Test Access Port

TAP controller is a FSM with 16 states to control these signals.

1. TDI : Test Data In
2. TDO : Test Data Out
3. TMS : Test Model Select
4. TRST : Test Reset
5. TCK : Test Clock

Mandatory

- Instruction Reg :- holds test instruct^{n} that is being loaded.
- Bypass Reg :- helps to bypass a chip
- Boundary Scan Reg

Registers

There are two types of registers associated with boundary scan. Each compliant device has one instruction register and two or more data registers.

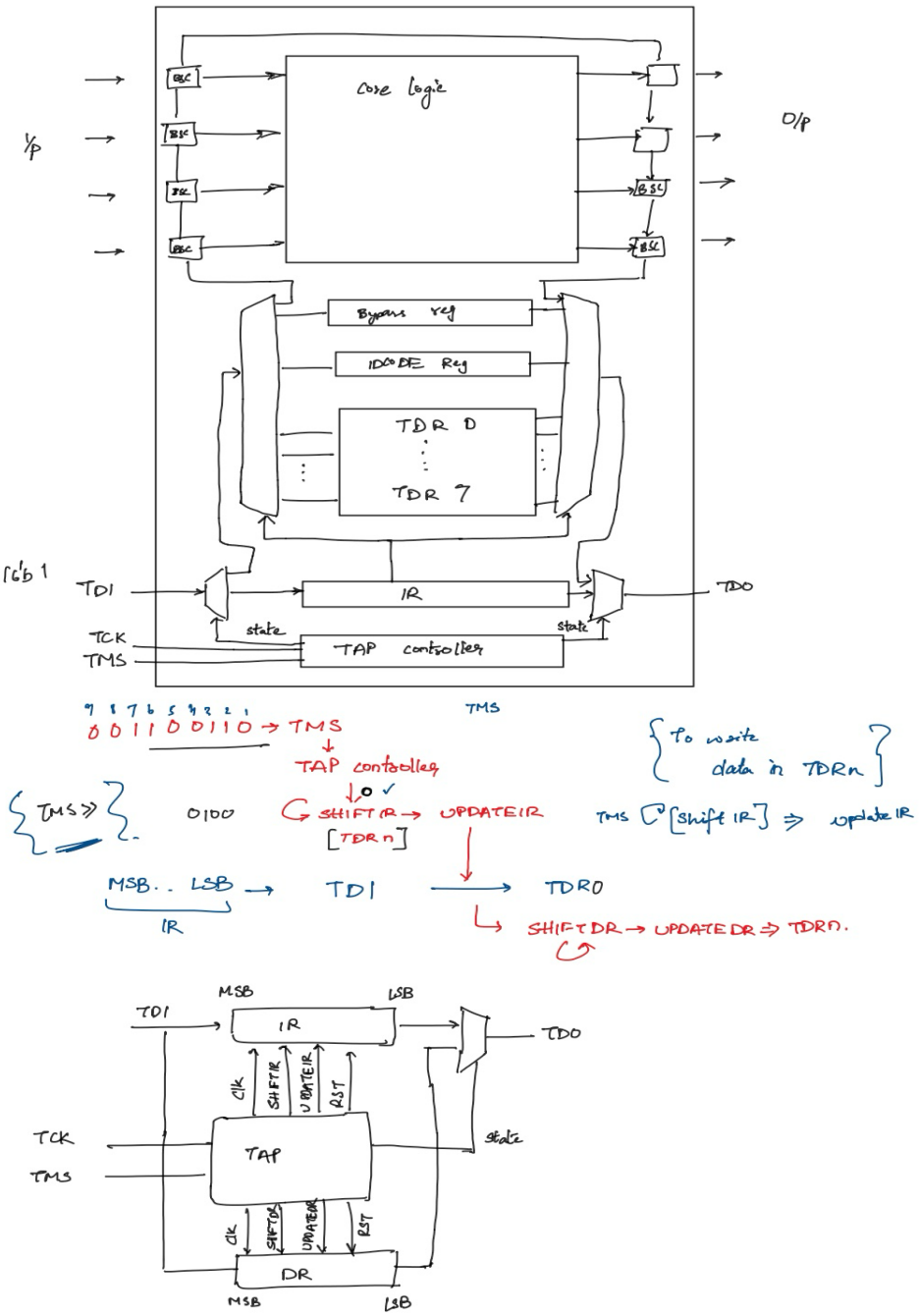
Instruction Register – the instruction register holds the current instruction. Its content is used by the TAP controller to decide what to do with signals that are received. Most commonly, the content of the instruction register will define to which of the data registers signals should be passed.

Data Registers – there are three primary data registers, the Boundary Scan Register (BSR), the BYPASS register and the IDCODES register. Other data registers may be present, but they are not required as part of the JTAG standard.

BSR – this is the main testing data register. It is used to move data to and from the I/O pins of a device.

BYPASS – this is a single-bit register that passes information from TDI to TDO. It allows other devices in a circuit to be tested with minimal overhead.

IDCODES – this register contains the ID code and revision number for the device. This information allows the device to be linked to its Boundary Scan Description Language (BSDL) file. The file contains details of the Boundary Scan configuration for the device.



Test Access Port (TAP) Controller

The TAP controller, a state machine whose transitions are controlled by the TMS signal, controls the behavior of the JTAG system. Figure below, shows the state-transition diagram.

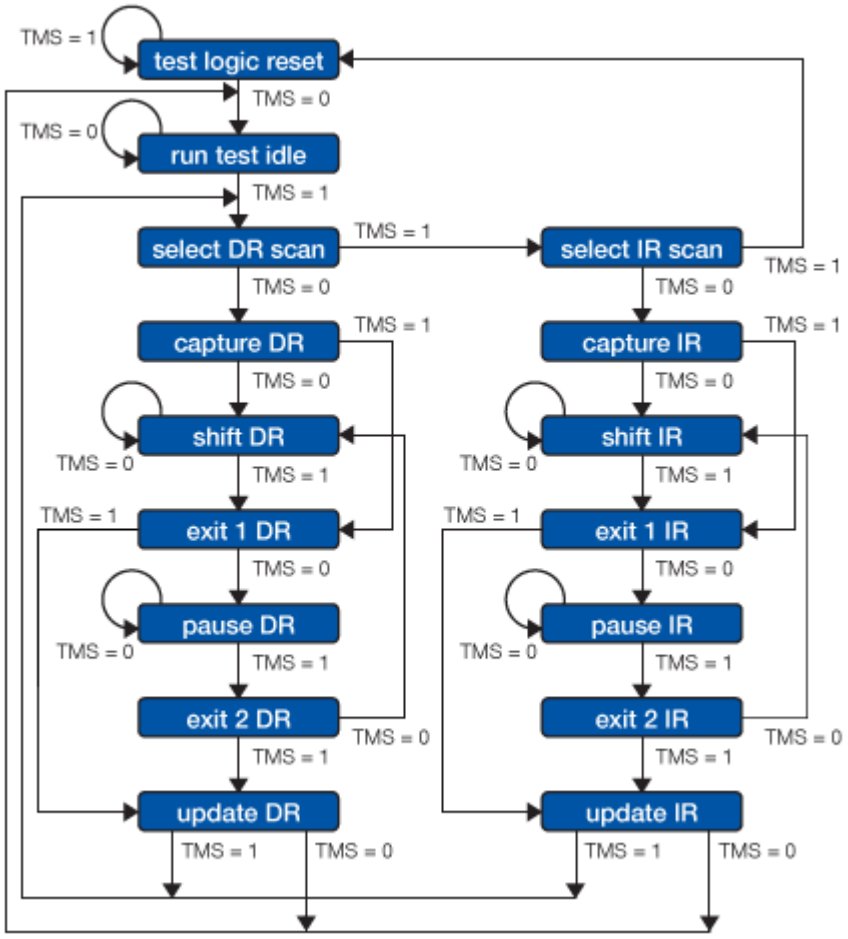


Fig: TAP Finite State Machine

The Tap Controller: The TAP controller is a finite state machine with a state diagram containing sixteen (16) states

The actions taken in each state are as follows:

Test-Logic-Reset: In this controller state, the test logic is disabled so that normal operation of the IC's system circuitry can proceed unhindered.

Run-Test/Idle: In this state, activity in selected test logic occurs only when certain instructions are present.

Select-DR-Scan , Select-IR-Scan: These are the starting states respectively for accessing one of the data registers (the boundary-scan or bypass register in the minimal configuration) or the instruction register.

Capture-DR , Capture-IR: These capture the current value of one of the data registers or the instruction register respectively into the scan cells. In instruction register we capture the status information, rather than the actual instruction with Capture-IR.

Shift-DR, Shift-IR: Shift a bit in from TDI (on the rising edge of TCK) and out onto TDO (on the falling edge of TCK) from the currently selected data or instruction register respectively.

Exit1-DR, Exit1-IR: These are the exit states for the corresponding shift state. From here the state machine can either enter a pause state or enter the update state.

Pause-DR , Pause-IR: Pause in shifting data into the data or instruction register. This allows for example test equipment supplying TDO to reload buffers etc.

Exit2-DR , Exit2-IR: These are the exit states for the corresponding pause state. From here the state machine can either resume shifting or enter the update state.

Update-DR , Update-IR: The value shifted into the scan cells during the previous states is driven into the chip (from inputs) or onto the interconnect (for outputs).

So we have a simple state machine, which allows either data registers or the instruction register to go through its capture-shift-update cycle, with an option to pause during the shifting.

All states have two exits, so all transitions can be controlled by the single TMS signal sampled on TCK. The two main paths allow for setting or retrieving information from either a data register or the instruction register of the device. The data register operated on (e.g. BSR, IDCODES, BYPASS) depends on the value loaded into the instruction register.

Boundary Scan Instructions

The IEEE 1149.1 standard defines a set of instructions that must be available for a device to be considered compliant. These instructions are:

BYPASS – this instruction causes the TDI and TDO lines to be connected via a single-bit pass-through register (the BYPASS register). This instruction allows the testing of other devices in the JTAG chain without any unnecessary overhead.

EXTEST – this instruction causes the TDI and TDO to be connected to the Boundary Scan Register (BSR). The device's pin states are sampled with the 'capture dr' JTAG state and new values are shifted into the BSR with the 'shift dr' state; these values are then applied to the pins of the device using the 'update dr' state.

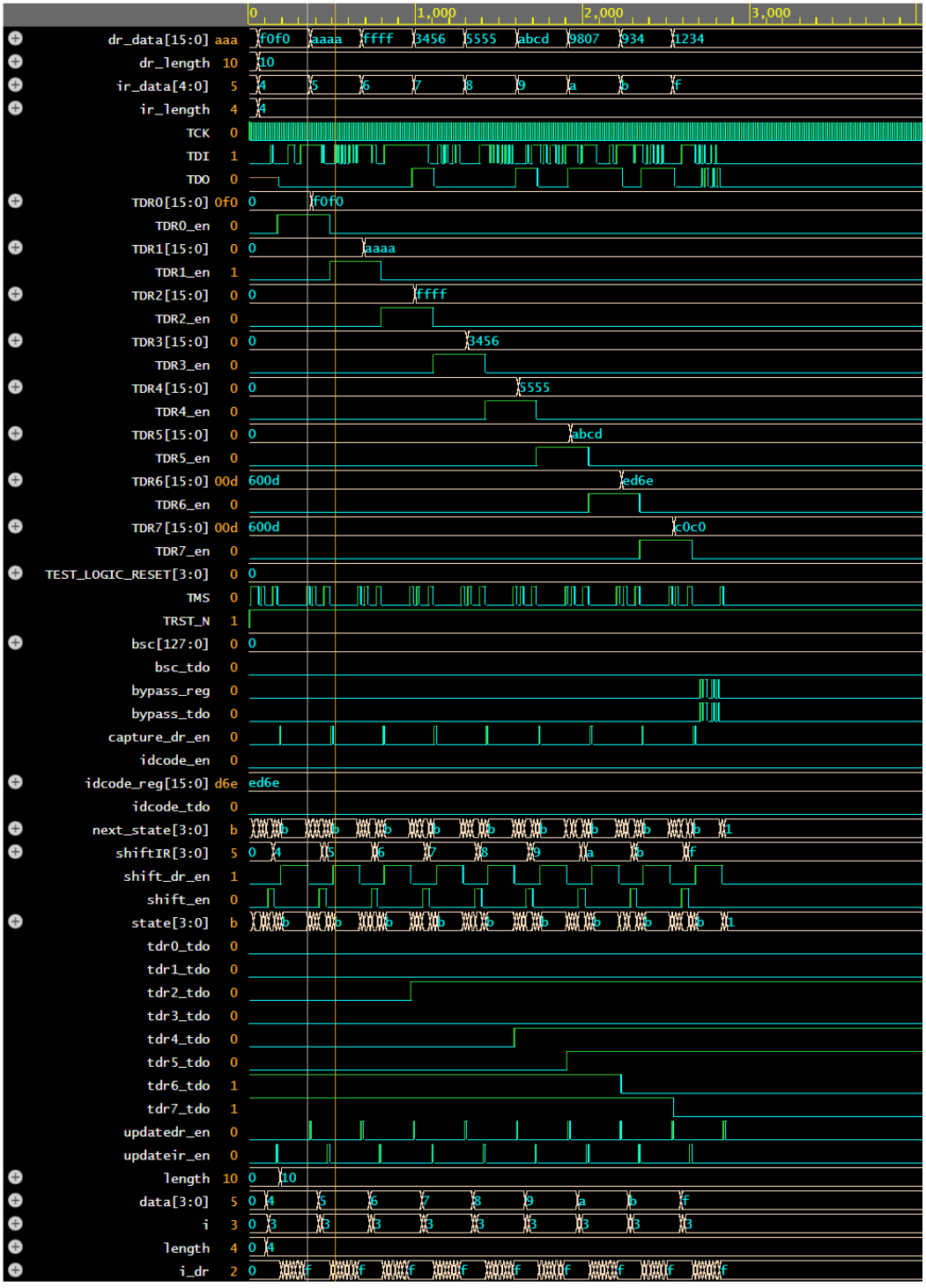
SAMPLE/PRELOAD – this instruction causes the TDI and TDO to be connected to the BSR. However, the device is left in its normal functional mode. During this instruction, the BSR can be accessed by a data scan operation to take a sample of the functional data entering and leaving the device. The instruction is also used to preload test data into the BSR prior to loading an EXTEST instruction.

Other commonly available instructions include:

IDCODE – this instruction causes the TDI and TDO to be connected to the IDCODE register.

INTEST – this instruction causes the TDI and TDO lines to be connected to the Boundary Scan Register (BSR). While the EXTEST instruction allows the user to set and read pin states, the INTEST instruction relates to the core-logic signals of a device.

Results


0
1,000
2,000
3,000
dr_data[15:0] aaa
dr_length 10
ir_data[4:0] 5
ir_length 4
TCK 0
TDI 1
TDO 0
TDR0[15:0] 0f0
TDR0_en 0
TDR1[15:0] 0
TDR1_en 1
TDR2[15:0] 0
TDR2_en 0
TDR3[15:0] 0
TDR3_en 0
TDR4[15:0] 0
TDR4_en 0
TDR5[15:0] 0
TDR5_en 0
TDR6[15:0] 00d
TDR6_en 0
TDR7[15:0] 00d
TDR7_en 0
TEST_LOGIC_RESET[3:0] 0
TMS 0
TRST_N 1
bsc[127:0] 0
bsc_tdo 0
bypass_reg 0
bypass_tdo 0
capture_dr_en 0
idcode_en 0
idcode_reg[15:0] d6e
idcode_tdo 0
next_state[3:0] b
shiftIR[3:0] 5
shift_dr_en 1
shift_en 0
state[3:0] b
tdr0_tdo 0
tdr1_tdo 0
tdr2_tdo 0
tdr3_tdo 0
tdr4_tdo 0
tdr5_tdo 0
tdr6_tdo 1
tdr7_tdo 1
updatedr_en 0
updateir_en 0
length 10
data[3:0] 5
i 3
length 4
i_dr 2
f0f0
aaaa
ffff
3456
5555
abcd
600d
ed6e
e0c0
1234
9807
934

References:

https://www.jtag.com/ieee-1149-1/

https://ieeexplore.ieee.org/document/6515989

https://www.iraj.in/journal/journal_file/journal_pdf/11-296-147564790336-40.pdf

https://www.xjtag.com/about-jtag/jtag-a-technical-overview/

https://medium.com/@aliaksandr.kavalchuk/diving-into-jtag-protocol-part-1-overview-fbdc428d3a16

https://reversepcb.com/swd-vs-jtag/#