

Testing and Troubleshooting PLDs

Testing a PLD

Testing a programmed PLD is the process of applying appropriate logic inputs to the device and then monitoring the outputs to see that they produce the correct logic levels at the desired times with the desired pattern of inputs.

The inputs can come from static sources such as switches in the smaller, simpler circuits. However, they usually come from a test instrument such as a pattern generator that produces desired bit patterns in a desired sequence for testing logic circuits. In a development system, the software can often be used to generate a set of inputs that will be applied to the PLD.

The outputs may be monitored on a logic analyzer or the development software can capture the outputs and analyze them to see that they match the results designated by the design.

In an automated manufacturing environment, the PLD may be tested by a special piece of automated test equipment (ATE) that is programmed to verify that the chip works as needed.

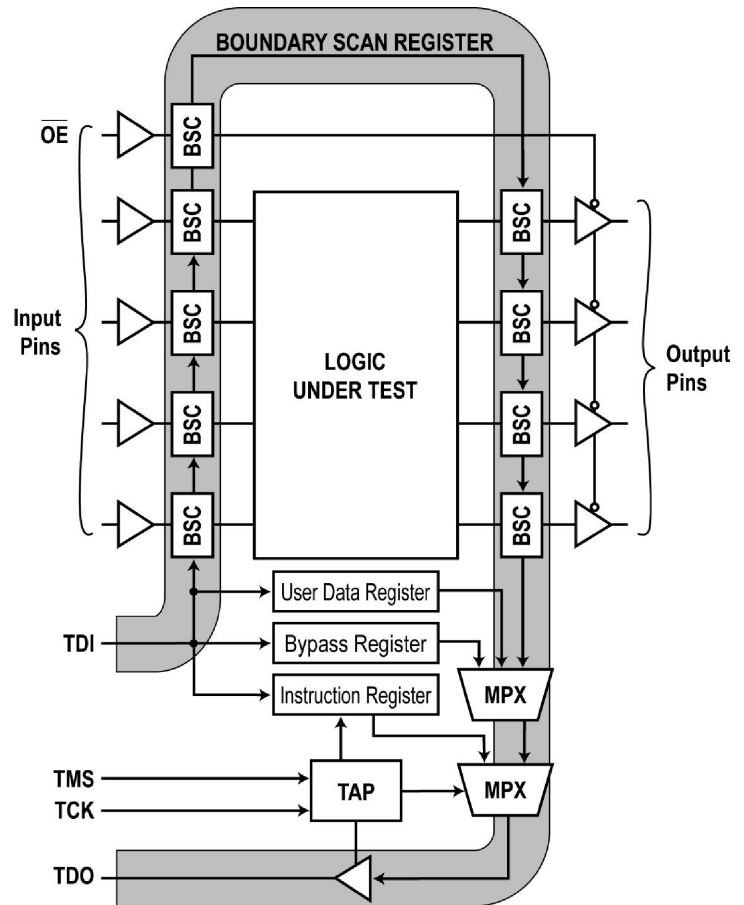
The JTAG Interface

Most large PLDs contain what is called a JTAG interface. JTAG means Joint Test Action Group, an industry consortium of companies that make chips, electronic equipment, and test instruments. JTAG defines a standard way to build test circuitry into large complex chips that makes it easier to test such chips. The JTAG defined a serial interface into and out of the chip that allows an external test generator to enter test commands and test bit patterns then receive outputs indicating the results of the test. The test procedure is called boundary scan.

JTAG is a formal IEEE standard, designated 1149.1. The boundary scan feature requires that a small amount of test logic be built into larger chips like FPGA to provide a way to not only test the chip itself but also to test larger assemblies of many interconnected chips in a manufacturing test environment.

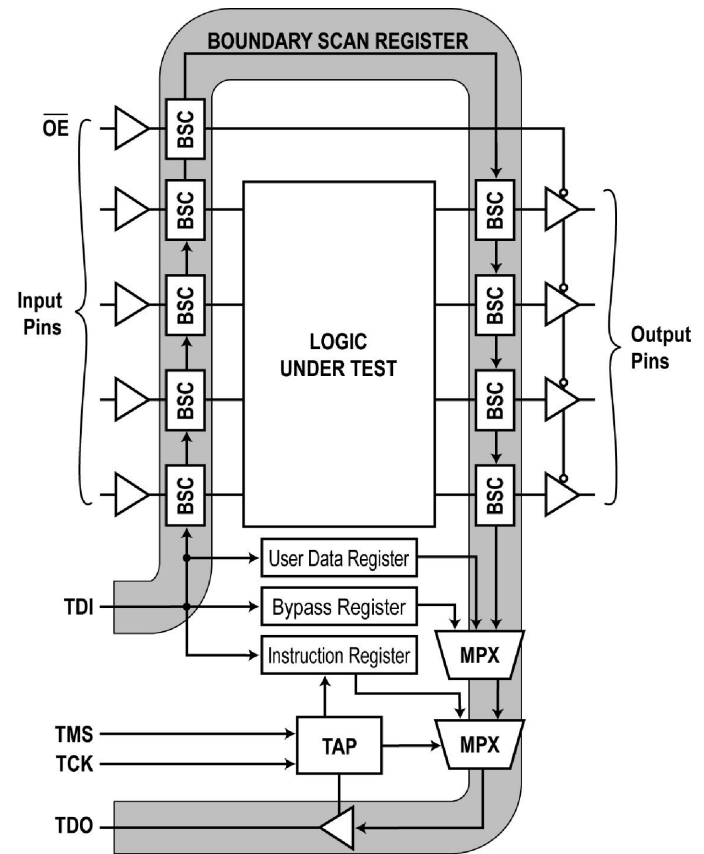
How Boundary Scan Works

This figure shows a general block diagram of the boundary scan architecture. It consists of multiple boundary scan cells (BSC). Each BSC contains a flip flop (FF) that is part of a large shift register called the boundary scan register. Each BSC is connected to one of the key points within the block of logic to be monitored or controlled.



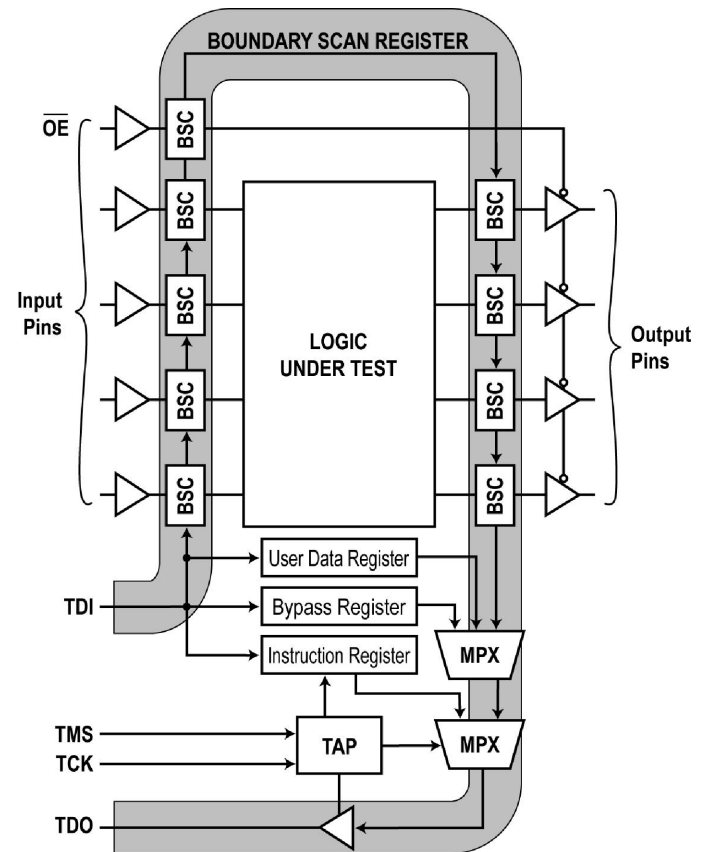
Normal Inputs and Outputs

The input pins on the left are the normal IC inputs. The BSC may allow these normal data inputs (NDI) to pass on through the PSC to the logic under test or a test input to be applied. The outputs on the right are the normal data outputs (NDO) from the IC which the BSC can allow to pass or the results of a test.

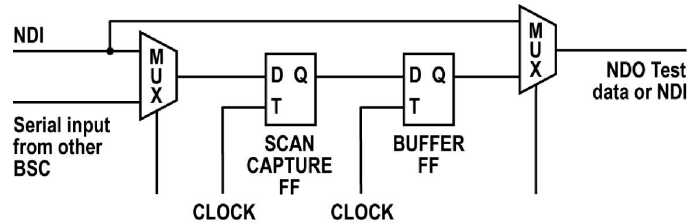


Test Data Inputs

Test patterns of bits can be loaded into the BSC from an external source called the test data input (TDI). These bits can be applied to the desired internal circuit inputs. Other BSCs monitor the logic outputs in response to the test inputs. The test results stored in the BSC flip flops can then be read out by shifting the BSC FFs until the test results appears at the test data output (TDO).



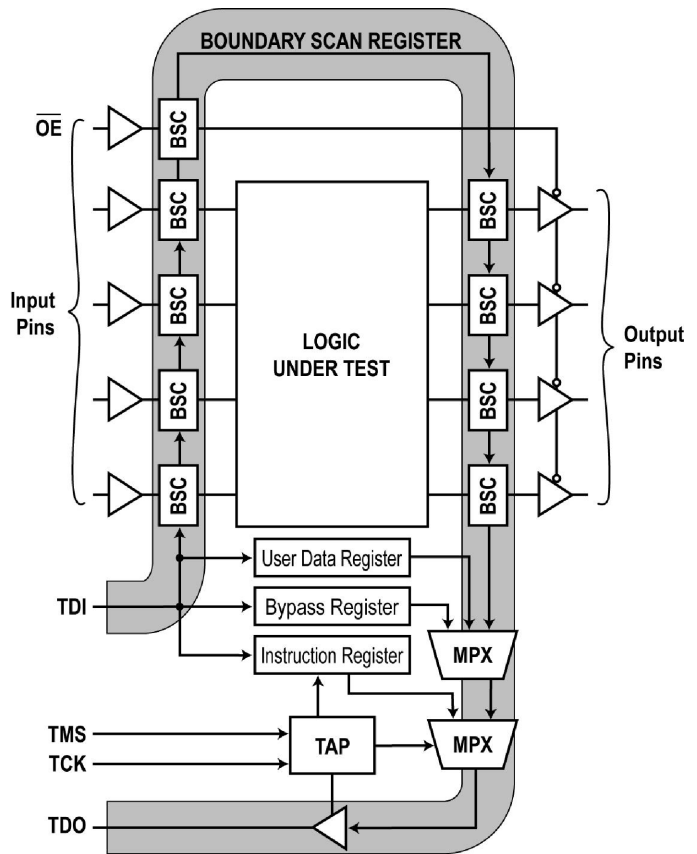
Boundary Scan Cell



This figure shows the details of each BSC. An input multiplexer selects either the NDI or a test serial data input from the TDI or another BSC to be stored in the scan capture register FF also called a boundary scan register (BSR). A buffer FF is also provided to store the state of the BSR FF. An output multiplexer allows the buffer output to be applied to the output or the NDI to be passed to the output.

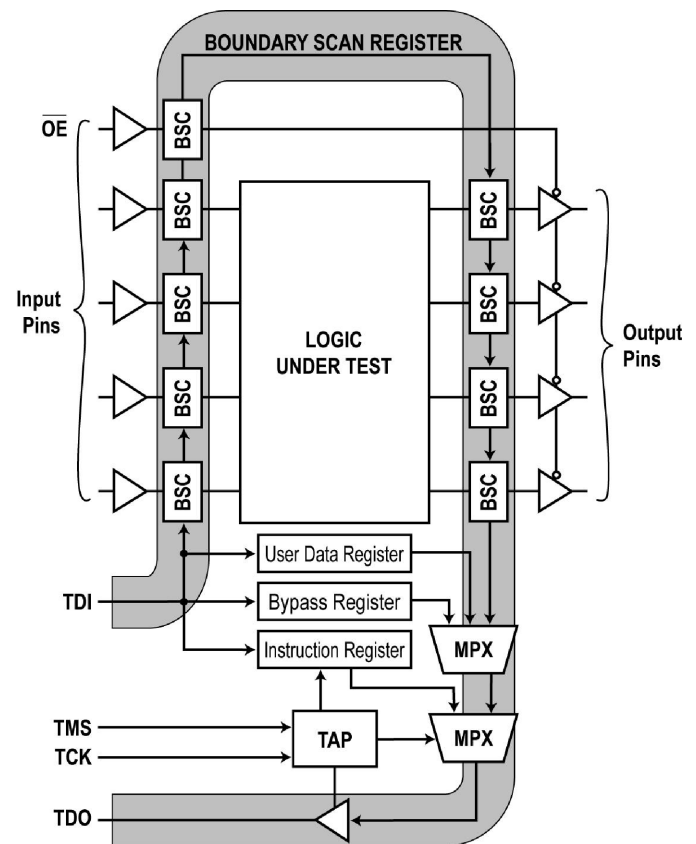
A Test Example

Referring again to the figure, note that the boundary scan process is controlled by logic called the test access port (TAP) controller. All functions are timed by the clock input (TCK). The test mode select (TMS) input determines if an instruction or test data is fed into the circuitry via the TDI input line.



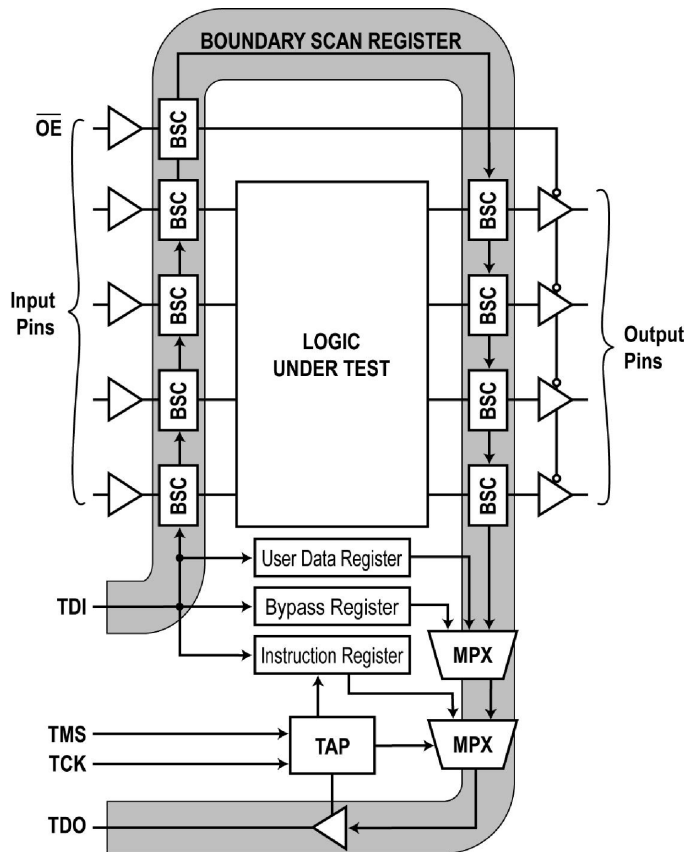
Testing

To initiate a test, a test instruction defined by a group of bits is loaded into the instruction register serially by way of the TDI line. The instruction register decoder identifies the test to be performed and then initiates operations in the TAP logic to make it happen.



Test Data Bit Patterns

Test data bit patterns are then fed into the TDI line where they are circulated in the BSR and used to provide a stimulus to the logic under test. Other BSCs capture the logic outputs that occur as the result of the test stimulus input. The BSR content is then shifted and continuously circulated or, if an output instruction is given, the data is fed to the TDO line.



Boundary Scan Benefits

The JTAG boundary scan method is about the only way to test large complex chips because the built in boundary scan circuits give access to points in the IC that are not connected to input or output pins. It provides a way to automate testing of the chip to see that it does what it is supposed to do. And it helps isolate problems if it does not work as desired.

The boundary scan method also provides a way to test for connections between two large chips on a PCB. Since most modern chips are of the surface mount type, once they are soldered to the PCB there is no access to the I/O pins because they are completely covered by the IC package. By using boundary scan tests on two interconnected chips, it is possible to determine if there is continuity between the output pin of one chip and the input to the other chip.

If boundary scan circuits are not built into the chip itself, special boundary scan ICs can be used with the ICs of the design to provide a way to test a complete PCB of multiple chips.

Troubleshooting a PLD

Like testing a PLD, troubleshooting involves applying appropriate inputs to analyze the operation of the device under varying conditions. At the same time, the outputs are monitored to determine the result. Simple PLDs can be tested like older discrete logic by monitoring with an oscilloscope or logic analyzer. Boundary scan tests can be developed and used to test the larger PLDs.

If the operation is not as desired, then the chip is defective in some way. Little can be done other than replace the chip.

One possible alternative is to remove the chip, if it can be removed, and to reprogram the chip. This is possible with chips that are installed in a socket on a PCB. PLDs are often installed this way. It is possible that some electrical disturbance caused a glitch in the internal programming of the chip. Reprogramming the chip may allow it to be reused.

Test your knowledge

Programmable Logic Devices Knowledge Probe 6 Testing and Troubleshooting PLDs

Click on [Course Materials](#) at the top of the page.
Then choose **Knowledge Probe 6**.