## BOUNDARY SCAN

In built-in self test (BIST) design, parts of the circuit are used to test the circuit itself. Online BIST is used to perform the test under normal operation, whereas off-line BIST is used to perform the test off-line. The essential circuit modules required for BIST include:

*Pseudo random pattern generator (PRPG)

* Output response analyzer (ORA)

The roles of these two modules are illustrated in Fig. 1. The implementation of both PRPG and ORA can be done with Linear Feedback Shift Registers (LFSRs).

**Pseudo Random Pattern Generator :-**

To test the circuit, test patterns first have to be generated either by using a pseudo random pattern generator, a weighted test generator, an adaptive test generator, or other means. A pseudo random test generator circuit can use an LFSR, as shown in Fig. 2.

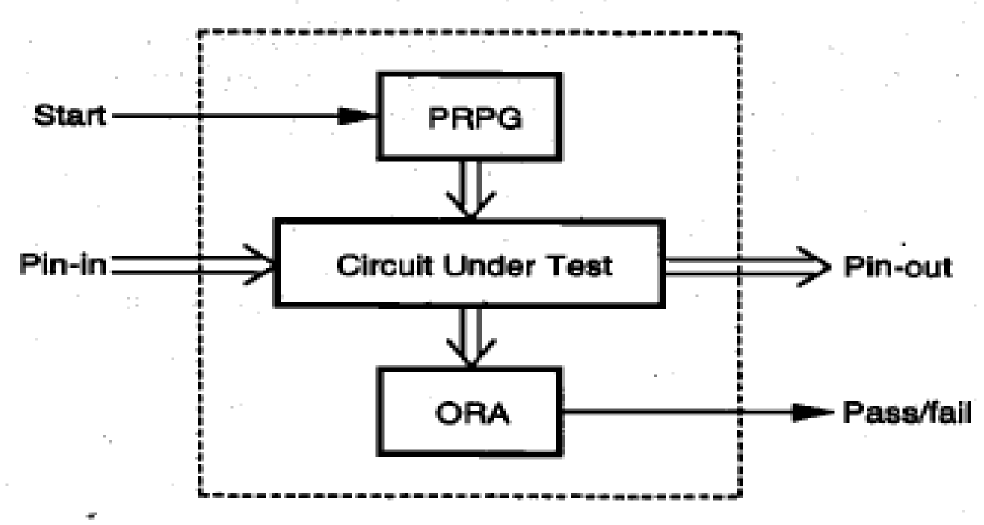

**Figure 5.6.1 : A procedure for BIST**

[Source: R.Jacob Baker, Harry W.LI., David E.Boyee, ―CMOS Circuit Design, Layout]

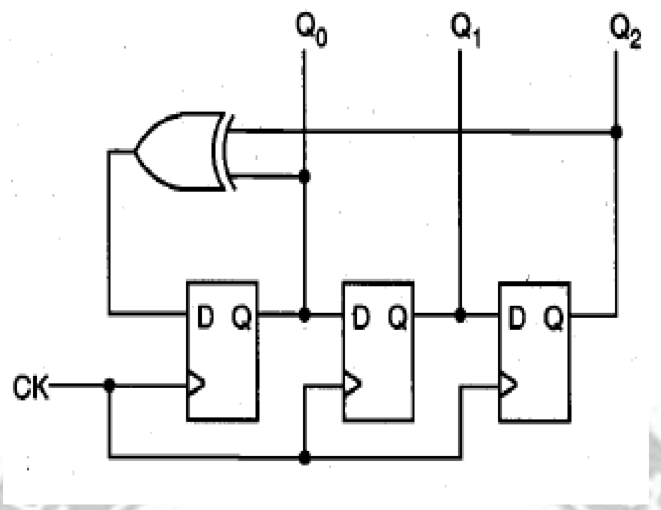

**Figure 5.6.2 : A pseudo-random sequence generator using LFSR**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

**Linear Feedback Shift Register as an ORA :-**

To reduce the chip area penalty, data compression schemes are used to compare the compacted test responses instead of the entire raw test data. One of the popular data compression schemes is the signature analysis, which is based on the concept of cyclic redundancy checking. It uses polynomial division, which divides the polynomial representation of the test output data by a characteristic polynomial and then finds the remainder as the signature. The signature is then compared with the expected signature to determine whether the device under test is faulty. It is known that compression can cause some loss of fault coverage. It is possible that the output of a faulty circuit can match the output of the fault-free circuit; thus, the fault can go undetected in the signature analysis. Such a phenomenon is called **aliasing.**

In its simplest form, the signature generator consists of a single-input linear feedback shift register (LFSR), as shown in Fig. 3 in which all the latches are edge-triggered. In this case, the signature is the content of this register after the last input bit has been sampled. The input sequence {an) is represented by polynomial G(x) and the output sequence by Q(x). It can be shown that G(x) = Q(x) P(x) R(x), where P(x) is the characteristic polynomial of LFSR and R(x) is the remainder, the degree of which is lower than that of P(x). For the simple case in Fig. 3 the characteristic polynomial is

$$P(x) = 1 + x^2 + x^4 + x^5$$

For the 8-bit input sequence { 1 1 1 1 0 1 0 1, the corresponding input polynomial is

$$G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1$$

and the remainder term becomes R(x) = $x^4$ $x^2$ which corresponds to the register contents of {0 0 1 0 1}.

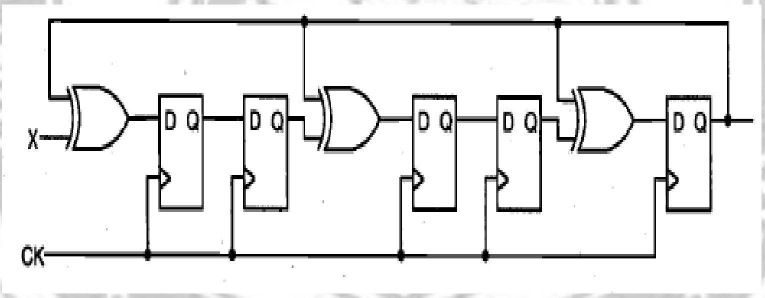

**Figure 5.6.3: Polynomial division using LFSR for signature analysis**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

Applications are found in high volume, high-end consumer products, telecommunication products, defense systems, computers, peripherals, and avionics. In fact, due to its economic advantages, some smaller companies that cannot afford expensive in-circuit testers are using boundary-scan.

The boundary-scan test architecture provides a means to test interconnects between integrated circuits on a board without using physical test probes. It adds a boundary-scan cell that includes a multiplexer and latches to each pin on the device.

Boundary-scan cells in a device can capture data from pin or core logic signals, or force data onto pins. Captured data is serially shifted out and externally compared to the expected results. Forced test data is serially shifted into the boundary-scan cells. All of this is controlled from a serial data path called the scan path or scan chain. Figure 1 depicts the main elements of a boundary-scan cell. By allowing direct access to nets, boundary-scan eliminates the need for a large number of test vectors, which are normally needed to properly initialize sequential logic. Tens or hundreds of vectors may do the job that had previously required thousands of vectors. Potential benefits realized from the use of boundary-scan are shorter test times, higher test coverage, increased diagnostic capability and lower capital equipment cost.

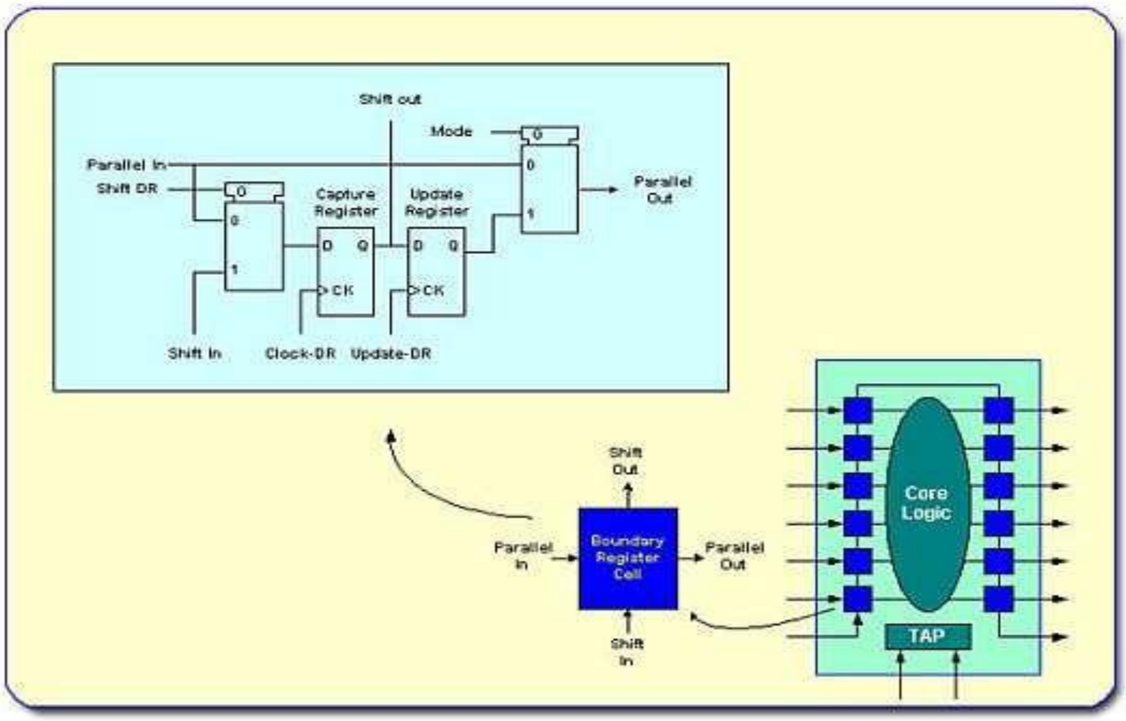

**Figure 5.6.4: Typical Boundary-Scan Cell**

[Source: Neil H.E. Weste, David Money Harris ―CMOS VLSI Design: A Circuits and Systems Perspective]

The principles of interconnect test using boundary-scan are illustrated in Figure 2. Figure 2 depicts two boundary-scan compliant devices, U1 and U2, which are connected with four nets. U1 includes four outputs that are driving the four inputs of U2 with various values. In this case, we assume that the circuit includes two faults: a short between Nets 2 and 3, and an open on Net 4. We will also assume that a short between two nets behaves as a wired-AND and an open is sensed as logic 1. To detect and isolate the above defects, the tester is shifting into the U1 boundary-scan register the patterns shown in Figure 2 and applying these patterns to the inputs of U2. The inputs values of U2 boundary-scan register are shifted out and compared to the expected results. In this case, the results (marked in red) on Nets 2, 3, and 4 do not match the expected values and, therefore, the tester detects the faults on Nets 2, 3, and 4.

Boundary-scan tool vendors provide various types of stimulus and sophisticated algorithms, not only to detect the failing nets, but also to isolate the faults to specific nets, devices, and pins.

ALKULAM, KANYAKUMARI
OBSERVE OPTIMIZE OUTSPREAD

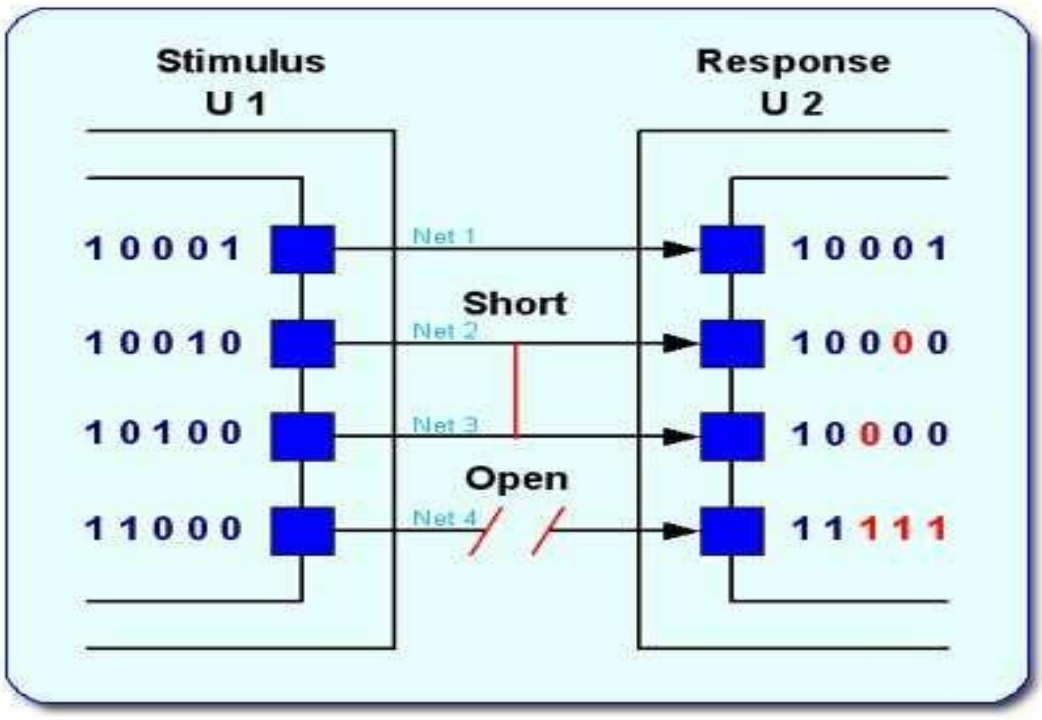

**Figure 5.6.5: Interconnect Test Example**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

**Boundary-Scan Chip Architecture**

- Chain integrity testing
- Interconnection testing between devices
- Core logic testing (BIST)
- In-system programming
- In-Circuit Emulation
- Functional testing

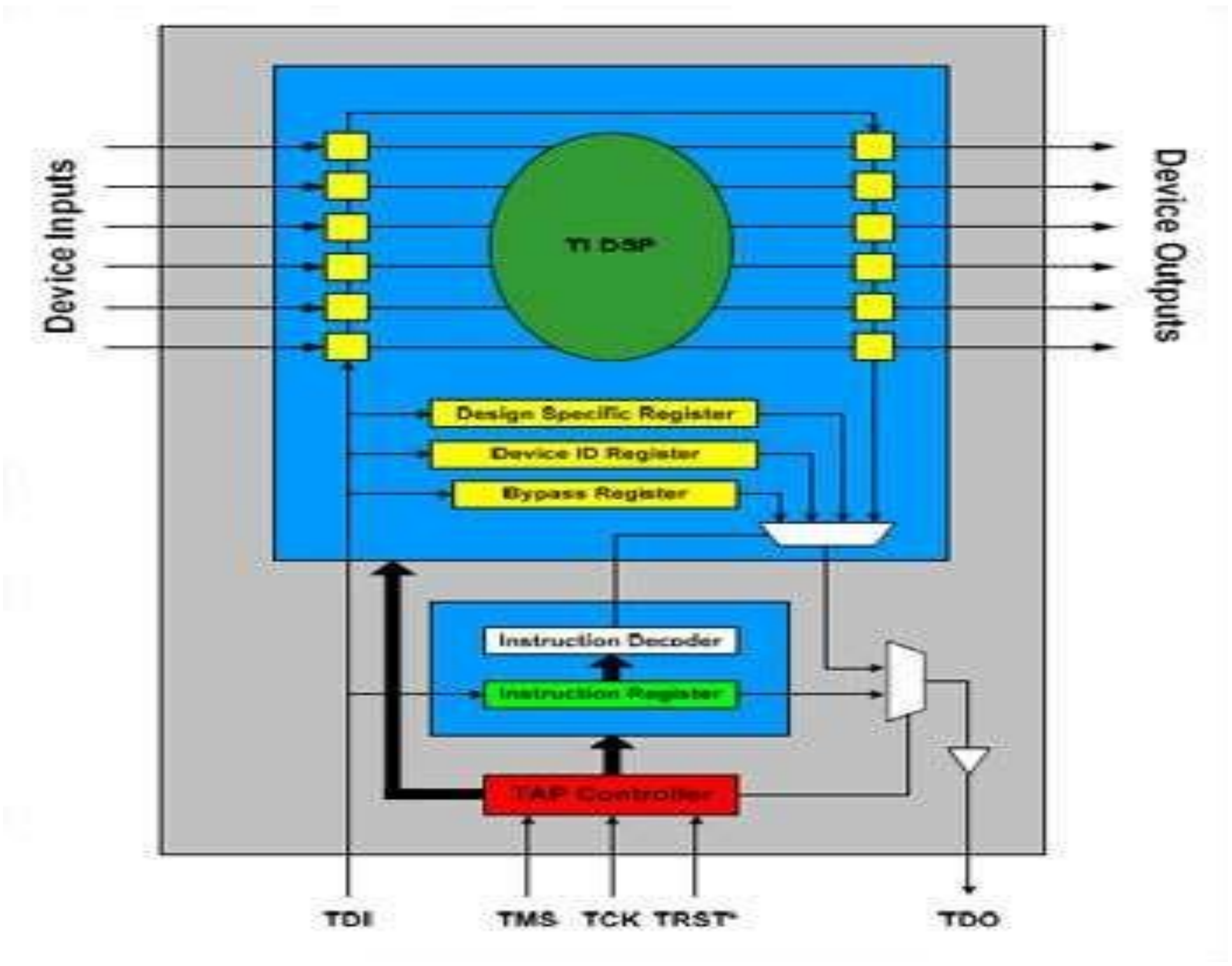

**Figure 5.6.6: Boundary-Scan Chain with Multiple Chips**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

OBSERVE OPTIMIZE OUTSPREAD

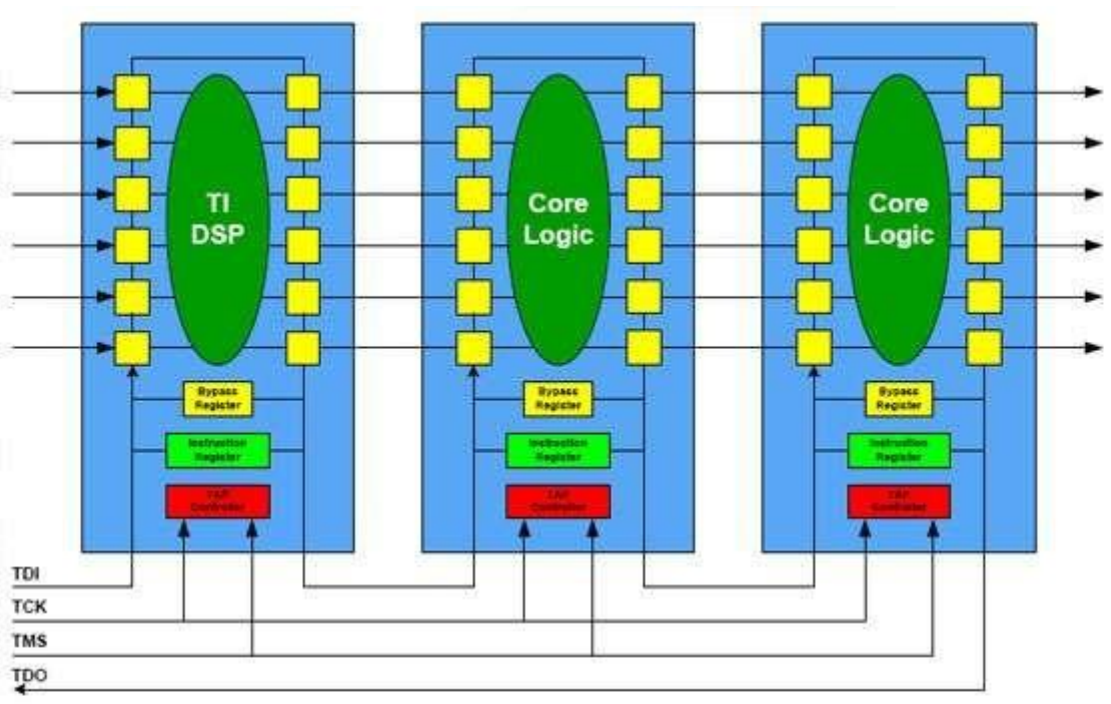

**Figure 5.6.7: Boundary-Scan Test Vectors**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

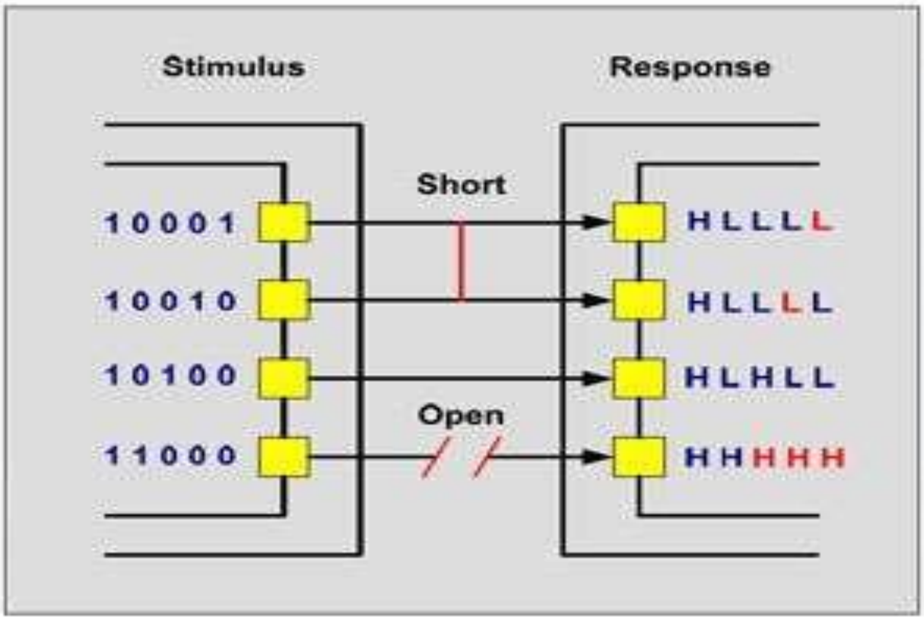

**Figure 5.6.8: Boundary-Scan Test Vectors**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

## JTAG Interface Test Access Port (TAP)

- **EXTEST**

  The EXTEST instruction performs a PCB interconnect test, compliant device into an external boundary test mode, and selects the boundary scanregister to be connected between TDI and TDO. During EXTEST instruction, the boundary scan cells associated with outputs are preloaded with test patterns to test downstream devices. The input boundary cells areset up to capture the input data for later analysis.

- **SAMPLE/PRELOAD**

  The SAMPLE/PRELOAD instruction allows an IEEE 1149.1 compliant device to remain in its functional mode and selects the boundary scan register to be connected between the TDI and TDO pins. During SAMPLE/PRELOAD instruction, the boundary scan register can be accessed through a data scan operation, to take a sample of the functional data input/output of the device. Test data can also be preloaded into the boundary-scan register prior to loading an EXTEST instruction.

- **BYPASS**

  Using the BYPASS instruction, a device's boundary scan chain can be skipped, allowing the data to pass through the bypass register. This allows efficient testing of a selected device without incurring the overhead of traversing through other devices. The BYPASS instruction allows an IEEE 1149.1 compliant device to remain in a functional mode and selects the bypass register to be connected between the TDI and TDO pins. Serial data is allowed to be transferred through a device from the TDI pin to the TDO pin without affecting the operation of the device.

## Boundary-Scan Applications

While it is obvious that boundary-scan based testing can be used in the production phase of a product, new developments and applications of the IEEE-1149.1 standard have enabled the use of boundary-scan in many other product life cycle phases. Specifically, boundary-scan technology is now applied to product design, prototype debugging and field service as depicted in Figure 3. This means the cost of the boundary-scan tools can be amortized over the entire product life cycle, not just the production phase.

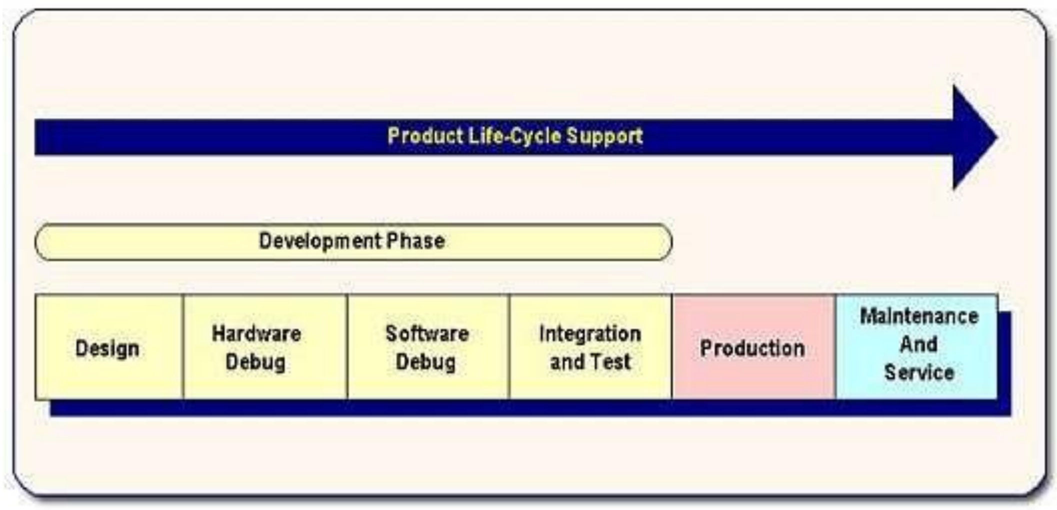

**Figure 5.6.9: Product Life Cycle Support**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

To facilitate this product life cycle concept, boundary-scan tool vendors such as Corelis offer an integrated family of software and hardware solutions for all phases of a product's life-cycle. All of these products are compatible with each other, thus protecting the user's investment.

## Applying Boundary-Scan for Product Development

The ongoing marketing drive for reduced product size, such as portable phones and digital cameras, higher functional integration, faster clock rates, and shorter product life-cycle with dramatically faster time-to- market has created new technology trends. These trends include increased device complexity, fine pitch components, such as surface-mount technology (SMT), systems-in-package (SIPs), multi-chip modules (MCMs), ball-grid arrays (BGAs), increased IC pin-count, and smaller PCB traces. These technology advances, in turn, create problems in PCB development:

- Many boards include components that are assembled on both sides of the board. Most of the through-holes and traces are buried and inaccessible.
- Loss of physical access to fine pitch components, such as SMTs and BGAs, makes it difficult to probe the pins and distinguish between manufacturing and design problems.
- Often a prototype board is hurriedly built by a small assembly shop with lower quality control as compared to a production house. A prototype generally will include more assembly defects than a production unit.
- When the prototype arrives, a test fixture for the ICT is not available and, therefore, manufacturing defects cannot be easily detected and isolated.
- Small-size products do not have test points, making it difficult or impossible to probe suspected nodes.
- Many Complex Programmable Logic Devices (CPLDs) and flash memory devices (in BGA packages) are not socketed and are soldered directly to the board.
- Every time a new processor or a different flash device is selected, the engineer has to learn from scratch how to program the flash memory.
- When a design includes CPLDs from different vendors, the engineer must use different in-circuit programmers to program the CPLDs.

Boundary-scan technology is the only cost-effective solution that can deal with the above problems. In recent years, the number of devices that include boundary-scan has grown dramatically. Almost every new microprocessor that is being introduced includes boundary-scan circuitry for testing and in-circuit emulation. Most of the CPLD and field programmable array (FPGA) manufacturers, such as Altera, Lattice and Xilinx, to mention a few, have incorporated boundary-scan logic into their components, including additional circuitry that uses the boundary-scan four-wire interface to program their devices in-system.

As the acceptance of boundary-scan as the main technology for interconnect testing and in-system programming (ISP) has increased, the various boundary-scan test and ISP tools have matured as well. The increased number of boundary-scan components and mature boundary-scan tools, as well as other factors that will be described later, provide engineers with the following benefits:

- Easy to implement Design-For- Testability (DFT) rules. A list of basic DFT rules is provided later in this article.
- Design analysis prior to PCB layout to improve testability.
- Packaging problems are found prior to PCB layout.
- Little need for test points.
- No need for test fixtures.
- More control over the test process.
- Quick diagnosis (with high resolution) of interconnection problems without writing any functional test code.
- Program code in flash devices.
- Design configuration data placement into CPLDs.
- JTAG emulation and source-level debugging.

What Boundary-Scan Tools are needed?

In the previous section, we listed many of the benefits that a designer enjoys when incorporating boundary-scan in his product development. In this section we describe the tools and design data needed to develop boundary-scan test procedures and patterns for ISP, followed by a description of how to test and program a board. We use a typical board as an illustration for the various boundary-scan test functions needed. A block diagram of such a board is depicted in Figure 4.

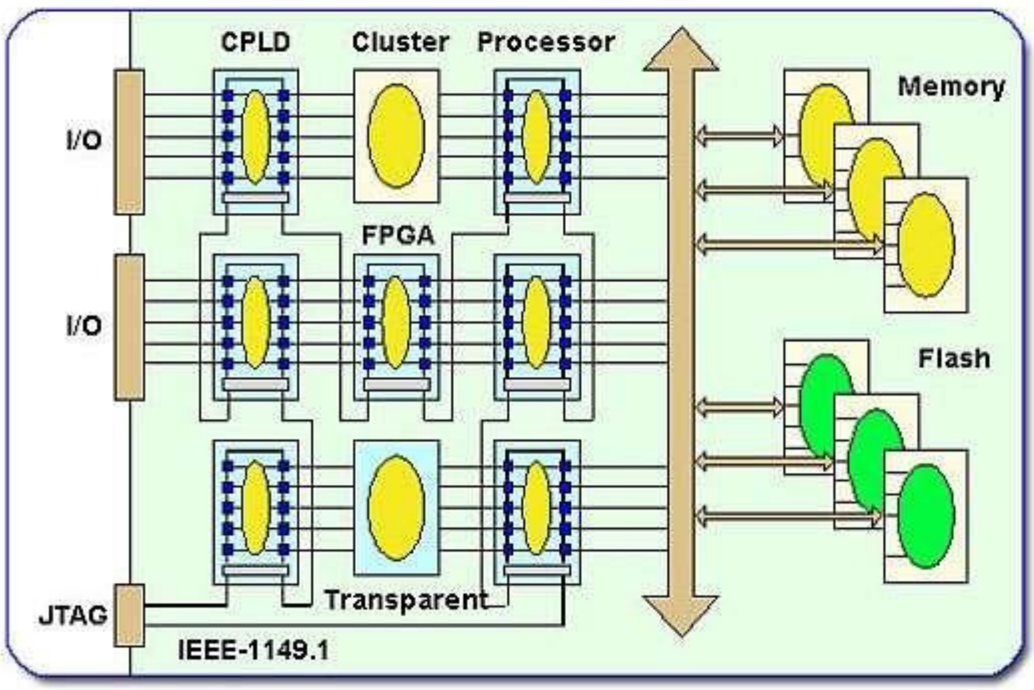

**Figure 5.6.10: Typical Board with Boundary-Scan Components**

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

A typical digital board with boundary-scan devices includes the following main components:

- Various boundary-scan components such as CPLDs, FPGAs, Processors, etc., chained together via the boundary-scan path.
- Non-boundary-scan components (clusters).
- Various types of memory devices.
- Flash Memory components.
- Transparent components such as series resistors or buffers.

Most of the boundary-scan test systems are comprised of two basic elements: Test Program Generation and Test Execution. Generally, a Test Program Generator (TPG) requires the netlist of the Unit Under Test (UUT) and the BSDL files of the boundary-scan components. The TPG automatically generates test patterns that allow fault detection and isolation for all boundary-scan testable nets of the PCB. A good TPG can be used to create a thorough test pattern for a wide range of designs. For example, ScanExpress TPG typically achieves net coverage of more than 60%, even though the majority of the PCB designs are not optimized for boundary-scan testing. The TPG also creates test vectors to detect faults on the pins of non-scannable components, such as clusters and memories that are surrounded by scannable devices.

Some TPGs also generate a test coverage report that allows the user to focus on the non-testable nets and determine what additional means are needed to increase the test coverage.

Test programs are generated in seconds. For example, when Corelis ScanExpress TPG™ was used, it took a 3.0 GHz Pentium 4 PC 23 seconds to generate an interconnect test for a UUT with 5,638 nets (with 19,910 pins). This generation time includes netlist and all other input files processing as well as test pattern file generation.

Test execution tools from various vendors provide means for executing boundary-scan tests and performing in-system programming in a pre-planned specific order, called a test plan. Test vectors files, which have been generated using the TPG, are automatically applied to the UUT and the results are compared to the expected values. In case of a detected fault, the system diagnoses the fault and lists the failures as depicted in Figure 5. Figure 5 shows the main window of the Corelis test execution tool, ScanExpress Runner™. ScanExpress Runner gives the user an overview of all test steps and the results of executed tests. These results are displayed both for individual tests as well as for the total test runs executed. ScanExpress Runner provides the ability to add or delete various test steps from a test plan, or re-arrange the order of the test steps in a plan. Tests can also be enabled or disabled and the test execution can be stopped upon the failure of any particular test.

Different test plans may be constructed for different UUTs. Tests within a test plan may be re-ordered, enabled or disabled, and unlimited different tests can be combined into a test plan. ScanExpress Runner can be used to develop a test sequence or test plan from various independent sub-tests. These sub-tests can then be executed sequentially as many times as specified or continuously if desired. A sub-test can also program CPLDs and flash memories. For ISP, other formats, such as SVF, JAM, and STAPL, are also supported.

PALKULAM, KANYAKUMARI
OBSERVE OPTIMIZE OUTSPREAD