By eliminating redundancy and providing a reduced fault coverage, it is possible to test most combinational logic blocks with a limited set of input vectors. This does not solve the sequential problem, however. To test a given fault in a state machine, it is not sufficient to apply the correct input excitation; the engine must be brought to the desired state first. This requires that a sequence of inputs be applied. Propagating the circuit response to one of the output pins might require another sequence of patterns. In other words, testing for a single fault in an FSM requires a sequence of vectors. Once again, this might make the process prohibitively expensive.

One way to address the problem is to turn the sequential network into a combinational one by breaking the feedback loop in the course of the test. This is one of the key concepts behind the *scan-test* methodology described later. Another approach is to let the circuit test itself. Such a test does not require external vectors and can proceed at a higher speed. The concept of *self-test* will be discussed in more detail later. When considering the testability of designs, two properties are of foremost importance:

1. **Controllability,** which measures the ease of bringing a circuit node to a given condition using only the input pins. A node is easily controllable if it can be brought to any condition with only a single input vector. A node (or circuit) with low controllability needs a long sequence of vectors to be brought to a desired state. It should be clear that a high degree of controllability is desirable in testable designs.
2. **Observability,** which measures the ease of observing the value of a node at the output pins. A node with a high observability can be monitored directly on the output pins. A node with a low observability needs a number of cycles before its state appears on the outputs. Given the complexity of a circuit and the limited number of output pins, a testable circuit should have a high observability. This is exactly the purpose of the test techniques discussed in the sections that follow.

Combinational circuits fall under the class of easily observable and controllable circuits, since any node can be controlled and observed in a single cycle.

Design-for-test approaches for the sequential modules can be classified in three categories: ad hoc test, scan-based test, and self-test.

H.3.2   Ad Hoc Testing

As suggested by the title, ad hoc testing combines a collection of tricks and techniques that can be used to increase the observability and controllability of a design and that are generally applied in an application-dependent fashion.

An example of such a technique is illustrated in Figure H-3a, which shows a simple processor with its data memory. Under normal configuration, the memory is only accessible through the processor. Writing and reading a data value into and out of a single memory position requires a number of clock cycles. The controllability and observability of the memory can be dramatically improved by adding multiplexers on the data and address busses (Figure H-3b).

(a) Design with low testability   (b) Adding a selector improves testability
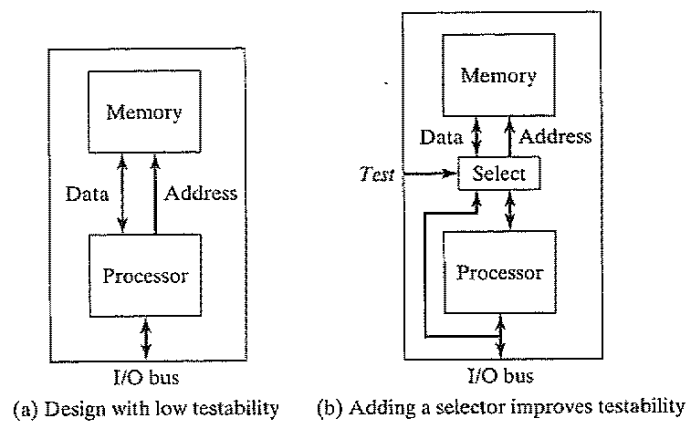
Figure H-3   Improving testability by inserting multiplexers.

During normal operation mode, these selectors direct the memory ports to the processor. During test, the data and address ports are connected directly to the I/O pins, and testing the memory can proceed more efficiently. The example illustrates some important design-for-testability concepts.

- It is often worthwhile to introduce *extra hardware* that has no functionality except improving the testability. Designers are often willing to incur a small penalty in area and performance if it makes the design substantially more observable or controllable.
- Design-for-testability often means that extra I/O pins must be provided besides the normal functional I/O pins. The *test* port in Figure H-3b is such an extra pin. To reduce the number of extra pads that would be required, one can multiplex test signals and functional signals on the same pads. For example, the I/O bus in Figure H-3b serves as a data bus during normal operation and provides and collects the test patterns during testing.

An extensive collection of ad hoc test approaches has been devised. Examples include the partitioning of large state machines, addition of extra test points, provision of reset states, and introduction of test busses. While very effective, the applicability of most of these techniques depends upon the application and architecture at hand. Their insertion into a given design requires expert knowledge and is difficult to automate. Structured and automatable approaches are more desirable.

H.3.3   Scan-Based Test

One way to avoid the sequential-test problem is to turn all registers into externally loadable and readable elements. This turns the circuit-under-test into a combinational entity. To control a node, an appropriate vector is constructed, loaded into the registers and propagated through the logic. The result of the excitation propagates to the registers and is latched, after which the con-
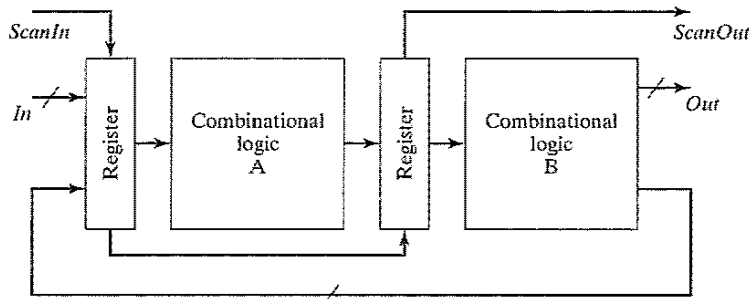
Figure H-4   Serial-scan test.

tents are transferred to the external world. Connecting all the registers in a design to a test bus regrettably introduces an unacceptable amount of overhead. A more elegant approach is offered by the serial-scan approach illustrated in Figure H-4.

The registers have been modified to support two operation modes. In the normal mode, they act as N-bit-wide clocked registers. During the test mode, the registers are chained together as a single serial shift register. A test procedure now proceeds as follows.

1. An excitation vector for logic module A (and/or B) is entered through pin *ScanIn* and shifted into the registers under control of a test clock.
2. The excitation is applied to the logic and propagates to the output of the logic module. The result is latched into the registers by issuing a single system-clock event.
3. The result is shifted out of the circuit through pin *ScanOut* and compared with the expected data. A new excitation vector can be entered simultaneously.

This approach incurs only a minimal overhead. The serial nature of the scan chain reduces the routing overhead. Traditional registers are easily modified to support the scan technique, as demonstrated in Figure H-5, which shows a 4-bit register extended with a scan chain. The only addition is an extra multiplexer at the input. When *Test* is low, the circuit is in normal operation
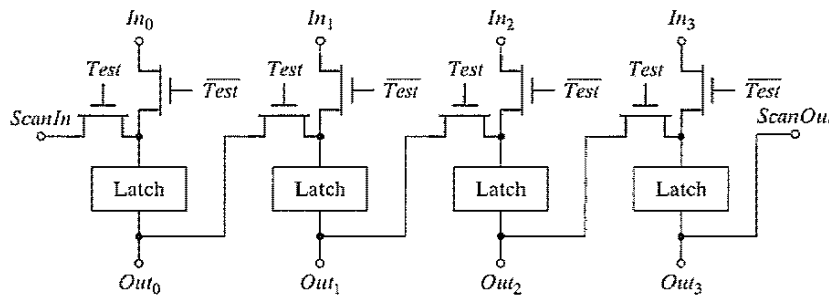

Figure H-5   Register extended with serial-scan chain.

mode. Setting *Test* high selects the *ScanIn* input and connects the registers into the scan chain. The output of the register *Out* connects to the fan-out logic, but also doubles as the *ScanOut* pin that connects to the *ScanIn* of the neighboring register. The overhead in both area and performance is small and can be limited to less than 5%.

Problem H.1   Scan-Register Design

Modify the static, two-phase master-slave register of Figure 7-10 to support serial scan.

Figure H-6 depicts the timing sequence that would be employed for the circuit in Figure H-4 under the assumption of a two-phase clocking approach. For a scan chain N registers deep, the *Test* signal is raised, and N clock pulses are issued, loading the registers. *Test* is lowered, and a single clock sequence is issued, latching the results from the combinational logic into the registers under normal circuit-operation conditions. Finally, N extra pulses (with *Test* = 1) transfer the obtained result to the output. Note again that the scan-out can overlap with the entering of the next vector.

Many variants of the serial-scan approach can be envisioned. A very popular one, which was actually the pioneering approach, was introduced by IBM and is called *level-sensitive scan design* (LSSD) [Eichelberger78]. The basic building block of the LSSD approach is the *shift-register latch* (SRL) shown in Figure H-7. It consists of two latches $L1$ and $L2$, the latter being
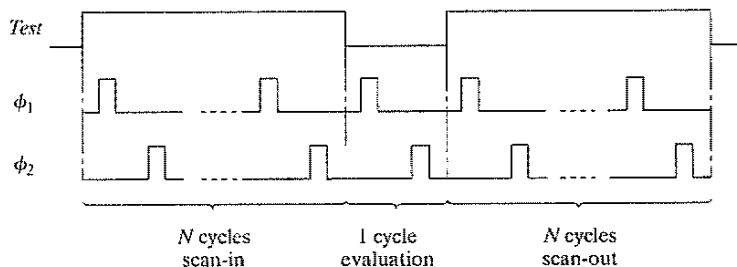

Figure H-6   Timing diagram of test-sequence. N represents the number of registers in the test chain.
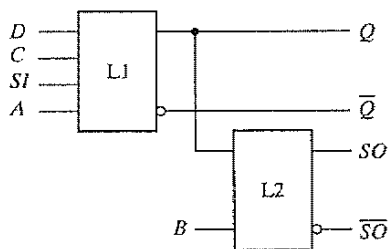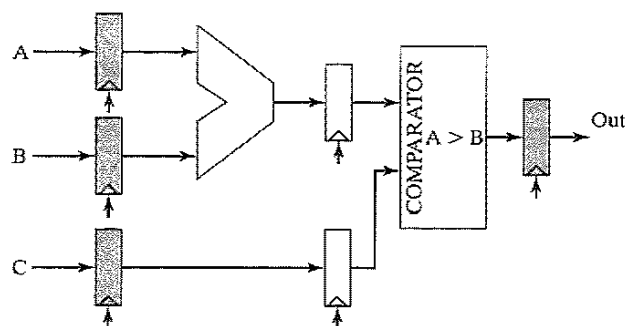

Figure H-7   Shift-register latch.

Figure H-8 Pipelined datapath using partial scan. Only the shaded registers are included in the chain.

present only for testing purposes. In normal circuit operation, signals D, Q ($\overline{Q}$), and C serve as latch input, output, and clock. The test clocks A and B are low in this mode. In scan mode, SI and SO serve as scan input and scan output. Clock C is low, and clocks A and B act as nonoverlapping, two-phase test clocks.

The LSSD approach represents not only a test strategy, but also a complete clocking philosophy. By strictly adhering to the rules implied by this methodology, it is possible to automate to a large extent the test generation and the timing verification. This is why the use of LSSD was obligatory within IBM for a long time. The prime disadvantage of the approach is the complexity of the SRL latch.

It is not always necessary to make all the registers in the design scannable. Consider the pipelined datapath of Figure H-8. The pipeline registers in this design are only present for performance reasons and do not strictly add to the state of the circuit. It is, therefore, meaningful to make only the input and output registers scannable. During test generation, the adder and comparator can be considered together as a single combinational block. The only difference is that during the test execution, two cycles of the clocks are needed to propagate the effects of an excitation vector to the output register. This approach is called *partial scan* and is often employed when performance is of prime interest. The disadvantage is that deciding which registers to make scannable is not always obvious and may require interaction with the designer.

H.3.4   Boundary-Scan Design

Until recently, the test problem was most compelling at the integrated circuit level. Testing circuit boards was facilitated by the abundant availability of test points. The through-hole mounting approach made every pin of a package observable at the back side of the board. For test, it was sufficient to lower the board onto a set of test probes (called "bed-of-nails") and apply and observe the signals of interest. The picture changed with the introduction of advanced packaging techniques such as surface-mount or multichip modules (Chapter 2). Controllability and observability are not as readily available anymore, because the number of probe points is dramatically

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.