9

Scan-Path Design

9.1 Introduction

Testing sequential circuits is a very complex process [Miczo 83]. Therefore, it has been important to find a scheme to facilitate their testing. The complexity of testing is a function of the number of feedback loops and their length. The longer a feedback loop, the more clock cycles are needed to initialize and sensitize patterns. In 1973, Williams and Angell proposed a design that increases the ease of testing of synchronous sequential circuits [Williams 1973]. Their scheme facilitates the initialization, controllability, and observability of the circuit.

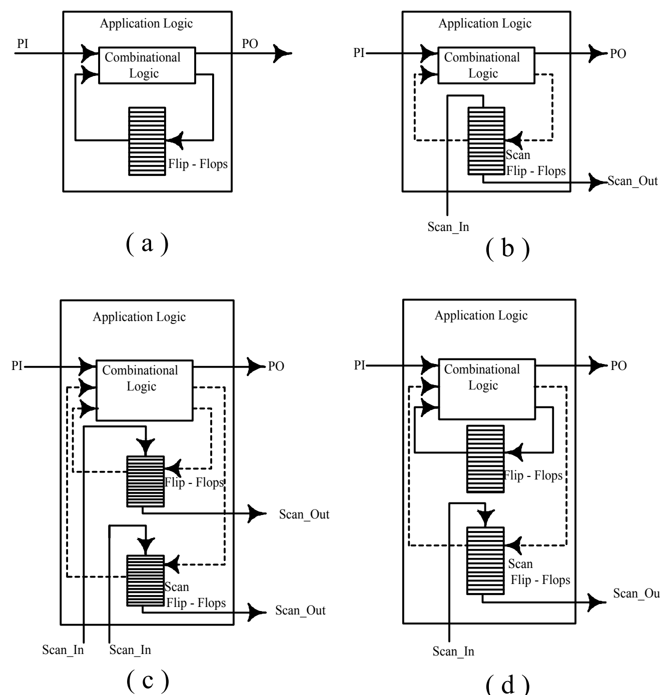


Figure 9.1 Scan-path Design

The suggested approach reduces the process of test pattern generation to that for combinational circuits. According to this scheme, a synchronous sequential circuit works in two modes, normal and test. The *normal mode* is depicted as shown in Fig. 9.1*a,* where the circuit is represented by the Huffman model introduced in Chapter 3. In the *test mode*, all the flip-flops are disconnected from the circuit and configured as a shift register, as illustrated in Fig. 9.1*b*. In this shift mode the flip-flops can be easily initialized. In this fashion, all the outputs of the flip-flops become pseudo primary inputs to the

Copyright © 2000 by John Wiley & Sons

circuit. After initializing the flip-flops, a test pattern is applied to the primary inputs, the results are latched at the flip-flops, and then they are propagated to the output by placing the circuit in a test mode and clocking enough times to capture the results. This arrangement makes the input to a flip-flop an observation point. All such observation points are pseudo primary outputs to the circuit. To switch between normal operation and shift modes, each flip-flop needs additional circuitry to perform the switch. There are different implementations of storage devices and they are discussed in Section 9.6.

Prior to the publication of scan-path methodology in 1973, the scheme was reported at a Japanese conference [Kobayashi 1963]. Also, at IBM, William C. Carter employed a very similar scheme for the purpose of design debugging [Carter 1964]. Scan-path design has evolved since Williams first proposed it. For example, modifications have been suggested for the storage devices. Instead of multiplexed flip-flops, two-port flip-flops or latches may be used. Also, different architectures have been recommended. It is possible to use a scan architecture that allows selective observation at any of the flip-flops in the chain. Another important approach is to organize the flip-flops in more than one scan chain, as illustrated in Fig. 9.1*c*. Instead of full-scan, it may be advantageous to scan just a few of the flip-flops. This approach, known as partial scan, is illustrated in Fig. 9.1*d*. Because of routing congestion in modern complex ICs, the placement order of the flip-flops present several problems we discuss in Section 9.11.

9.2 Scan-Path Design

Any synchronous sequential circuit may be modeled as shown in Fig. 9.2*a*, which is a detailed version of Fig. 9.1*a*. Here we assume that only D flip-flops are used. This assumption is very realistic and does not limit the applicability of the scheme to other flip-flop types. The combinational part represents the next-state forming logic and the output forming logic. For scan design, each flop-flop is preceded by a 2-to-1 MUX. The select lines of all flip-flops are connected together as shown in Fig. 9.2*b*. As this select line, scan enable (SE) is 0, the circuit works under normal operation. Since the other input of the MUX is connected to the output of the previous flip-flop, for SE = 1, all flip-flops are connected as a shift register. For the first flip-flop, the second input to the MUX is labeled as SI. Also, the output of the last flip-flop is labeled as SO.

Copyright © 2000 by John Wiley & Sons

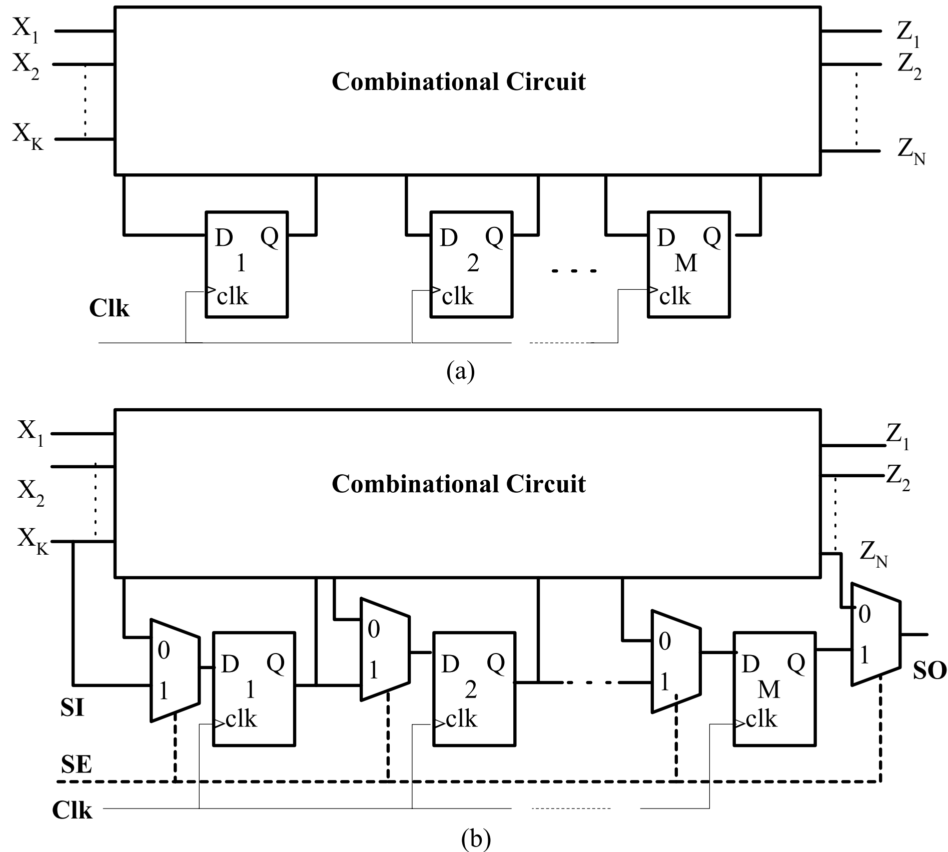


Figure 9.2 (a) A Generic for a Synchronous Sequential Circuit

(b) The Circuit Configured with Scan-path

This arrangement indicates that the scan-path approach requires three extra I/O pins for the signals scan enable (SE), scan-in (SI), and scan-out (SO). However, instead of connecting SI to an extra pin, this signal is multiplexed with one of the inputs. Similarly, the SO is multiplexed with one of the outputs. In Fig. 9.2*b*, we multiplexed SI with X3 and SO with Z3. The different alternatives of connecting these signals do not affect the operation of scanpath.

9.3 Test Pattern Generation

Test patterns are generated only for the combinational circuit based on the following assumptions:

1. No asynchronous signals are in the circuit, including set and reset of flip-flops.
2. Latches are controlled by nonoverlapping clocks.
3. For test pattern generation:

Copyright © 2000 by John Wiley & Sons

3.1. Propagation ends at an input of a flip-flop, which is considered as an output of the circuit.

3.2. Justification ends at an output of a flip-flop, which is considered an input to the circuit.

Thus a circuit configured in scanpath has $N + M$ primary inputs, where N is the original primary inputs of the circuit and M is the number of the flip-flops to be part of the scan chain. The inputs of the flip-flops are dispersed within the circuit and thus allow improved controllability of the nodes. The increased number of the outputs also improves the observability of the circuit and, in effect, partitions the circuit into smaller components.

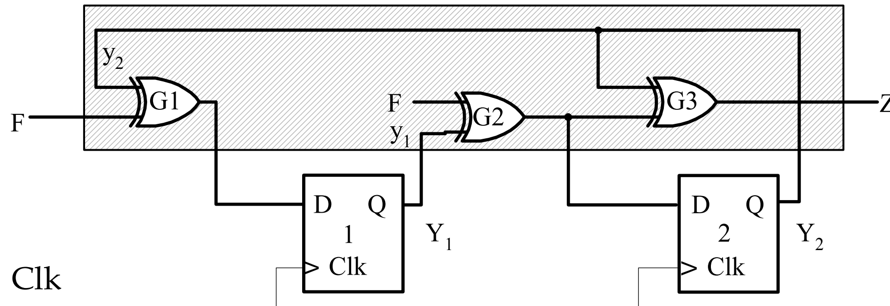


Figure 9.3 Circuit *S*1 to Illustrate Scan-Path Design

The example in Fig. 9.3 shows how the combinational part is actually partitioned. It consists of two parts. The first part consists only of gate 1; the second comprises gates 2 and 3. Each partition has one primary input, *F*, but while the first partition has one pseudoinput, Y_2, the other partition has two pseudoinputs, Y_1 and Y_2.

9.4 Test Pattern Application

Test application consists of two stages. The flip-flops are tested, followed by the combinational part of the circuit. Testing the flip-flops is done first since they will be used in controlling and observing the testing of the combinational part of the circuit.

9.4.1 Testing the Flip-Flops

The flip-flops are configured as a shift register, SE = 1 and patterns are applied through the SI input. The output is observed at the SO pin. The pattern applied may consist of M 0's followed by M 1's, where M is the number of flip-flops. Because of the risk of pattern sensitivity, which will be discussed in conjunction with RAM testing in Chapter 12, other variations of patterns may be used. It is possible, for example, to alternate 0's and 1's.

Copyright © 2000 by John Wiley & Sons

9.4.2 Testing the Combinational Part of the Circuit

Test patterns have to be generated for each partition in the circuit, and the patterns need to be organized in the proper order of application. For stuck-at faults, the order is irrelevant, however, in the case of fault models for which the order of the sequence is pertinent, this has to be taken into consideration. After the application of each pattern, the response is first latched in the flop-flops and then it is scanned out of the circuit by configuring the flip-flops in a shift register. This scheme is summarized in the following protocol:

Repeat until all patterns are applied.

a. Set SE = 1, shift in the initial values on the flip-flops. (These are the signals at the output of the latches for the first test pattern.)

b. SE = 0, apply a pattern at the primary inputs.

c. Clock the circuit once and observe the results at the primary outputs.

d. Clock the circuit *M* times.

End repeat.

Since each pattern includes signals on the output of the flip-flops, we entered these signals prior to the application of the other signals of the patterns at the primary inputs. However, the initialization of the flip-flops may be performed as the previous test is scanned out–step d of the test application protocol given above. We can modify the test application protocol as follows:

Initialize.

Set SE = 1; shift in the initial values on the flip-flops for the first pattern.

Repeat until all patterns are applied.

a. Set SE = 0, apply a pattern at the primary inputs.

b. Clock the circuit once and observe the primary outputs.

c. Set SE = 1.

- Apply the initialization for the next pattern at SI.
- Clock the circuit, *M* times.
- Observe at the primary outputs and the SO pins.

End repeat.

The protocol is used to apply test patterns on the circuit shown in Fig. 9.4. and is described in the next section.

Copyright © 2000 by John Wiley & Sons

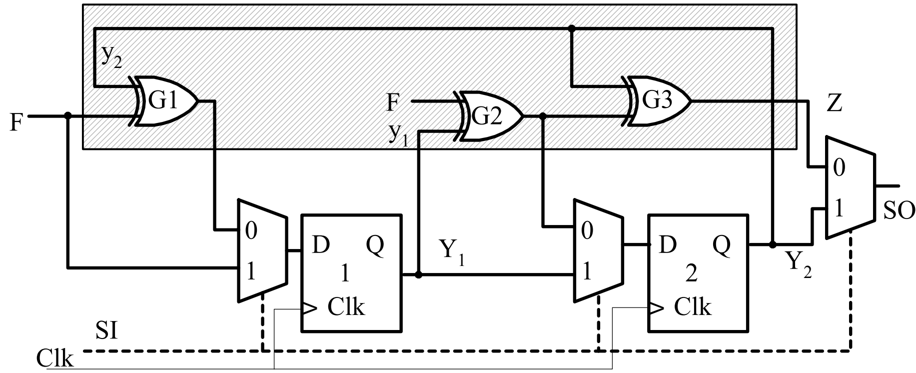
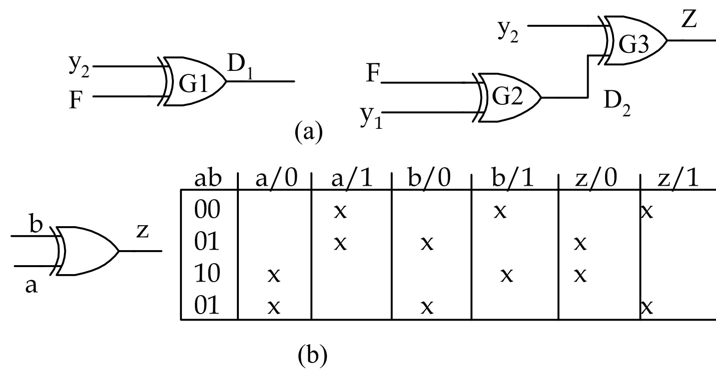


Figure 9.4 Circuit S1 with Scan-Path Inserted

9.5 Example for Scan-Path Testing



ab	a/0	a/1	b/0	b/1	z/0	z/1
00		x		x		x
01		x	x		x	
10	x			x	x	
01	x		x			x

Fig 9.5 The partitions of S1 and Test Set for An XOR gate

Originally, the circuit had one input (excluding the clock) and one output. With scan-path structure, the PI and PO have been increased by the number of flip-flops. Also, the combinational part of the circuit has been partitioned into two components, as illustrated in Fig. 9.5. XOR 1 has inputs F and y_2, and the second component consists of the two other XOR gates with inputs F, y_1,and y_2, and the two outputs Z and D_2.

The first step in testing this sequential circuit will be to generate the patterns for the combinational parts of the circuit. Here we have three XOR gates that can be tested simultaneously. We learned from previous chapters that three patterns are sufficient to test this gate. However, assuming any type of implementation, we will use an exhaustive test, four patterns. Let us assume that we will use the test in the following sequence: Fy_i = 00, 10, 01, 11, where i = 1 or 2, depending on the gate tested. Thus the patterns on the second partition are $FY1Y2$ = {000,100,011,111}.

Copyright © 2000 by John Wiley & Sons

We next apply the patterns. We summarize the steps taken in the application of the test in Table 9.1. We chose to apply the all-0's pattern followed by an all-1's pattern to check the flip-flops. This is given in the first four

Table 9.1. Summary of Test Application to the Circuit of Fig. 9.4.

clk	*SE*	*F*	*SI*	*PS* y_1y_2	*NS* Y_1Y_2	*Z*	*SO*
1	1	x	1	$x\,x$	1 x	x	X
2	1	x	1	1 x	1 1	x	Xx
3	1	x	0	1 1	0 1	0	1xx
4	1	x	0	0 1	0 0	1	11xx
5	1	x	0	0 0	0 0	0	011xx
6	0	0	x	0 0	D_1D_2	0	Z011xx
7	1	x	0	D_1D_2	0 D_1	x	$D_2$0011xx
8	1	x	0	0 D_1	0 0	x	$D_1D_2$0011xx
9	0	1	x	0 0	D_1D_2	1	$ZD_1D_2$0011xx
10	1	x	1	D_1D_2	1 D_1	x	$D_2ZD_1D_2$0011xx
11	1	x	1	1 D_1	1 1	x	$D_1D_2ZD_1D_2$0011xx
12	0	0	x	1 1	D_1D_2	0	$ZD_1D_2ZD_1D_2$0011xx
13	1	x	1	D_1D_2	1 D_1	x	$D_2ZD_1D_2ZD_1D_2$0011xx
14	1	x	1	1 D_1	1 1	x	$D_1D_2ZD_1D_2ZD_1D_2$0011xx
15	0	1	x	1 1	D_1D_2	1	$ZD_1D_2ZD_1D_2ZD_1D_2$0011xx
16	1		a	D_1D_2	a D_1	x	$D_2ZD_1D_2ZD_1D_2ZD_1D_2$0011xx
17	1		b	a D_1	b a	x	$D_1D_2ZD_1D_2ZD_1D_2ZD_1D_2$0011$xx$

lines of the table. The first four columns of this table list the inputs to the circuit: the clk, SI, *F*, and SE. The fifth and sixth columns give the present state (PS) and the next state (NS) of the flip-flops. The present state for each clock is, of course, the next state from the previous clock.

To apply pattern 00 on G1 and G2 means that $Fy_1 = 00 = Fy_2 = 00$. That is, the pattern is $Fy_1y_2 = 000$. Thus for $y_1y_2 = 00$, we clock the circuit twice with SE = 1 and SI = 0. However, since in testing the flip-flops we have already entered 0's on the flip-flops, the circuit was ready for the application of the first test. Thus we apply $F = 1$ while SE = 0 and clock the circuit once. Then while SE =1, we clock twice while applying 0 on SI. These two clock cycles will allow the observability of the response to the first pattern one and the initialization of the flip-flops for the second test pattern, $Fy_1y_2 = 100$. To apply this pattern, we simply apply 0 on *F* and clock once. To observe the pattern, we clock twice while SE = 1. Again, as the results of the second pattern are scanned out, the initialization for the third pattern is performed. Such initialization is not needed when the last pattern is applied.

Copyright © 2000 by John Wiley & Sons

Therefore, at the last clock cycles, 16 and 17, no particular signals are needed for subsequent test patterns. We deliberately indicated the data as *a* and *b* to emphasize that no particular signal is required.

9.6 Storage Devices

The most important requirement of storage device is the ability to work individually in normal mode or to be configured as a shift register in testing mode. In addition to multiplexed flip-flops, two other types are used: two-port flip-flops and level-sensitive latches. For a partial scan, the devices that are not scanned have an added requirement and that is of holding its value during shift mode.

9.6.1 Two-Port Flip-Flop

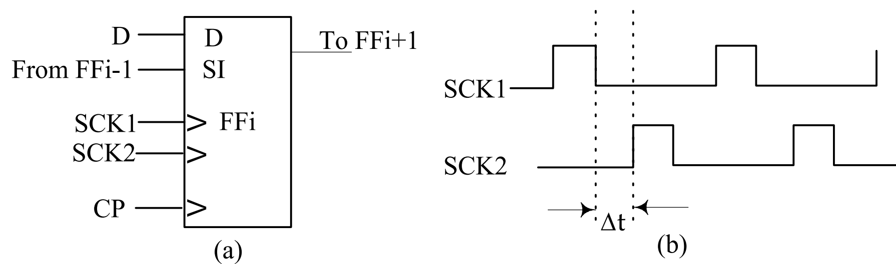


Figure 9.6 Two Port D Flip-flop (a) Symbol (b) Non-Overlapping Clocks

To configure the flip-flops in a shift register, we preceded each one by a 2-to-1 multiplexer. We can obtain the same functionality substituting a two-port storage device that was originally proposed to minimize the gate size and its delays [Funatsu 1978]. A possible version of this device is the scan flip-flop, FD3S2, shown in Fig. 9.6*a* [LSI 1992]. It has two data lines, D and SI and two clocking systems: CP and the combined S*CK*1 and S*CK*2 signals. The input at pin D is latched at the output at the positive edge of the system clock CP whereas transition the input at pin SI is captured after the pulses at SCK1 and SCK2 are applied. SCK1 and SCK2 are used for shift mode in a non-overlapping fashion shown in Fig. 9.6*b* in order to minimize the hazards. After the application of each test pattern, CP is clocked once. Then, SCK1 and S*CK*2 are clocked *M* times, where *M* is the number of two-port flip-flops in the scan chain.

9.6.2 Clocked Latch

The storage units in a sequential design do not have to be edge triggered. They may be latches. The latter devices are level sensitive and require a specific order on signal application. Figure 9.7 shows a clocked D-latch. Its next-state equation is $Y = CD + yC'$. However, to eliminate static hazard, the latch is redesigned as shown in Fig. 9.7*b*

Copyright © 2000 by John Wiley & Sons

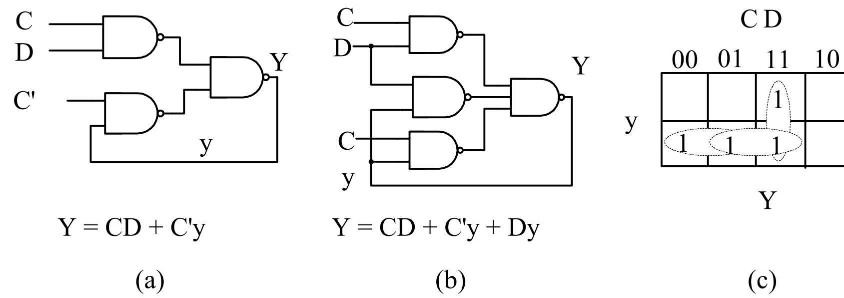


Figure 9.7 Hazard Free Clocked Latch [Eichelberger 1985]

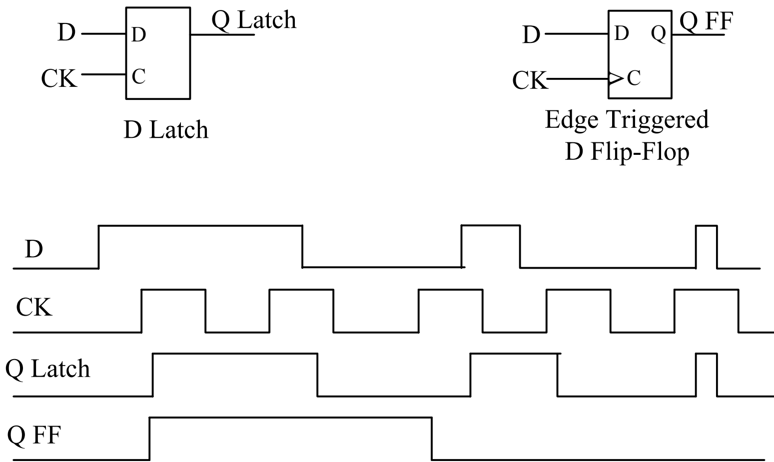


Figure 9.8 Comparison of Operations of a Clocked Latch and an Edge-triggered Flip-flop

[Eichelberger 1965]. The governing equation is then $Y = CD + yC' + Dy$. As $C = 0$, the latch's output is unchanged whereas when $C = 1$, any change on the input, D, is reflected immediately on the output, Y. This is known as the transparency property of latches. The clock changes to 1 only when the input, D, is stabilized. But the slew rate of the clock does not alter the final value of the latch. It may cause delays, but no change results in the level of the output. This property is illustrated in Fig. 9.8, where the edge-triggered flip-flop and the clocked latch responses to the same input waveform are shown. Notice that when the clock, CK, is high, the input on D is seen immediately on Q for the clocked latch. This is known as the transparency property [McCluskey 1996].

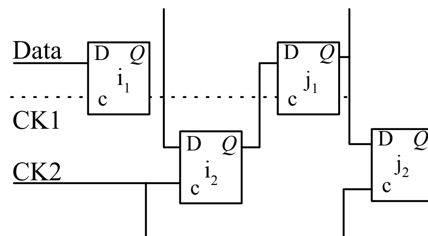


Figure 9.9 Formation of a Shift Register with Clocked Latches

Copyright © 2000 by John Wiley & Sons

Because of the transparency property, latches cannot be connected as a shift register while controlled by the same clock. When the clock is high, the value on the first latch will be passed immediately to all the next latches in a domino effect. For this, two latches are concatenated to perform the task of a flip-flop, as illustrated in Fig. 9.8. The circuit in Fig. 9.9 shows latches connected in a master–slave arrangement. However, in order to avoid essential hazards, the second latch is operated with clock CK2 which is nonoverlapping with clock CK1.

9.7 Scan Architectures

Scan-path design can be organized in various architectures [McCluskey 1986] and may be mixed with other DFT structures such as built-in self-test (BIST) that will be described in Chapter 11. In this section, we will examine only two different architectures: Level-sensitive scan design (LSSD) and scan-set. The first is a design that replaces flip-flops by latches and the second has the particular feature of being able to monitor the circuit on-line.

9.7.1 Level-Sensitive Scan Design

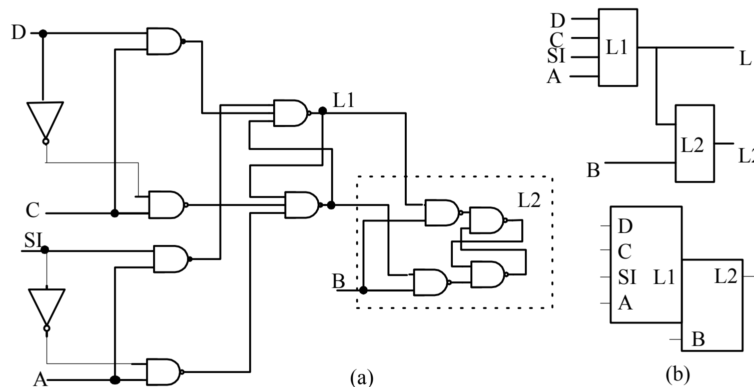


Figure 9.10 Polarity Hold Latch (a) Logic Diagram; (b) Symbolic Representations

As described at the end of Section 9.6, the latch cannot be used in a shift register because when the clock is high, the value on one latch will change immediately because of the transparency property. For this, two latches are concatenated to perform the task of a flip-flop. The circuit in Fig. 9.10 shows two latches connected in a master-slave arrangement [Eichelberger 1977]. However, to avoid essential hazard, the second latch is operated with a clock *B* that is nonoverlapping with clock *C*. This latch may then be used in a shift register as shown in Fig. 9.9, where the operation of the register is also illustrated.

To use the latch in a scan-path design, the design is augmented as shown in Fig. 9.7. This latch was proposed and patented by Edward Eichelberger and is the basis of IBM's level-sensitive scan design. Two clocks, *C*

Copyright © 2000 by John Wiley & Sons

and *B*, are needed to perform the shift. To multiplex between normal operation and shift modes, a third clock, *A*, is used. The normal operation of the latch implies the use of clocks *C* and *B*. For shift mode, clocks *A* and *B* are used.

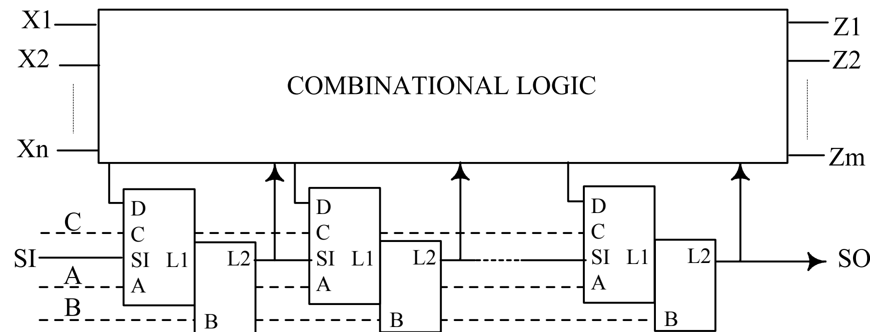


Figure 9.11 LSSD Design

Each flip-flop in Fig. 9.2*a* is substituted for by a level-sensitive latch, and the circuit is then transformed to the one shown in Fig. 9.11. The test is performed as described in Section 9.2.2, with the exception that every time a clock signal is applied, it is replaced by the sequence clock, *C*, followed by clock *B*, or by clock *A* followed by clock *B* for the shift mode. We list the steps here for convenience.

Test the latches.

Set $A = B = 1$.
Apply 0 and 1 alternatively at SI.
Clock *A*, then clock *B*, *N* times.

Initialize.

Shift in the initial values on the flip-flops. (These are the signals at the output of the latches for the first test pattern.)

Repeat until all patterns are applied.

a. Apply a pattern at the primary inputs.
b. Clock *C*; then clock *B* once and observe the results at the primary outputs.
c. Shift out the response

- Apply the initialization for the next pattern at SI.
- Clock *A*, then clock *B, M* times.
- Observe at the primary outputs and the SO pins.

9.7.2 Scan-Set Architecture

An interpretation of the scan-set architecture is shown in Fig. 9.12. The circuit storage devices are not configured as a shift register and a shift register is added to the circuit. It also consists of two-port flip-flops [Stewart 1978]. The

Copyright © 2000 by John Wiley & Sons

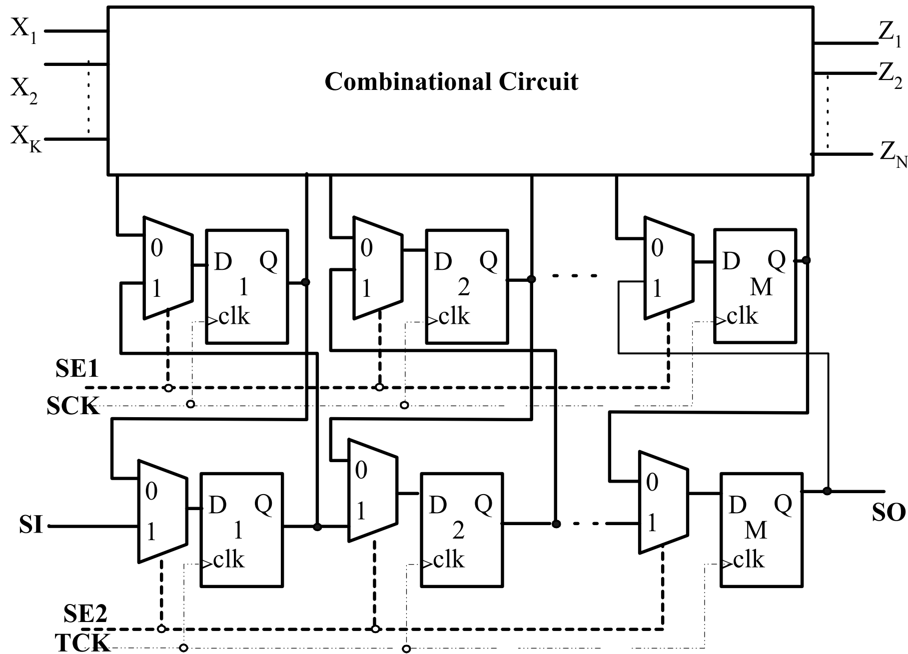


Figure 9.12 Scan Set Architecture

initialization data are introduced serially in the shift register, using clock TCK and SI port. The data are then transferred in parallel to all the latches of the circuit using the clock SCK. At this stage a test pattern is applied to the primary input and the system clock, SLK, is applied once to transfer the test response to the output of the latches. The shift register is then clocked *N* times with TCK, in order to scan the data to SO. The observation of the response is *on-line.* That is, it is possible to observe the circuit while it is working under normal operation. For both architectures described in this section, the storage devices may be arranged in one or multiple chains, as described next.

9.8 Multiple Scan Chains

Test application time is a function of the number of flip-flops scanned; the more flip-flops in the circuit, the longer the testing time will be. It is possible to reduce the time by forming more than one scan chain that can be operated in parallel. This arrangement is particularly effective for BIST design, which employs very long pseudo-random test sets. Also, some circuits use multiple clocks to control different parts of the circuit. It is easier in this case to configure the storage devices controlled by each clock separately. For externally applied test patterns, multiple chains require more pins or a more complex pin-multiplexing scheme. However, if BIST is used, there is no pin penalty.

Copyright © 2000 by John Wiley & Sons

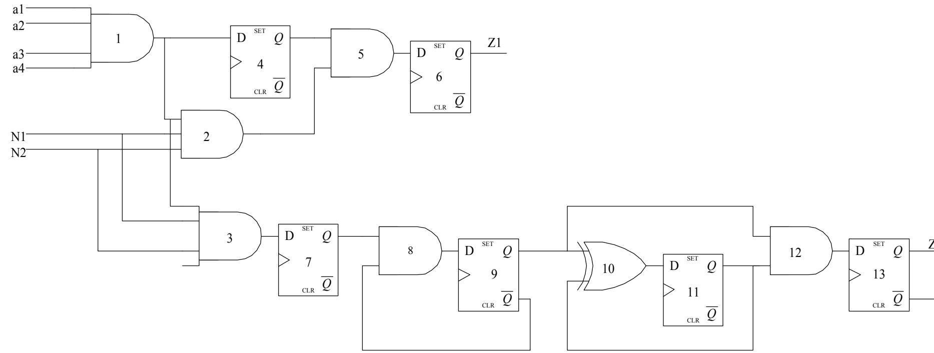


Figure 9.13 Example S2 for a Sequential Circuit

In general, selecting the appropriate devices in each scan chain is far from trivial. In addition to having to balance the number of storage devices in the parallel chains, it is important to select the devices such that the interconnect wires are minimized. This topic is discussed in Section 11.10. The topology of the circuit in Fig. 9.13 is, for example, amenable to two-scan chain organization.

9.9 Cost of Scan-Path Design

Scan-path design has greatly facilitated the testing of sequential circuits and has been widely accepted in industry. The ease of testing did not come free. The cost of scan-path design involves five principal issues: extra area, higher pin count, circuit performance, test application time, and power dissipation at speed testing.

9.9.1 Area and I/O Pins

The increase in silicon area is due to the complexity of the scan device, flip-flop, or latch. Although the overhead for one storage device may be high, the overall increase in hardware is of a lesser magnitude since all storage devices constitute a smaller percentage of the circuit. In addition, an efficient design of the scan cell can reduce the overhead. The most significant increase is due to the extra routing for connecting the flip-flops as a shift register. Efforts to order the device in the chain such that the area occupied by the interconnect wires is minimized can be done at placement and routing time. The extra pins are SI, SO, SE, or the second and third clocks that are used in multiple-port storage devices as described earlier in this chapter. SI and SO can possibly be multiplexed with input and output pins.

9.9.2 Performance

The major concern with performance is due to the added circuitry. In the multiplexed flip-flops, there is delay though the multiplexer. Two-port flip-flops (or latches) do not cause delay since the data can be available at the two ports simultaneously. The drawback of such types of devices is the need for a two-phase clock.

Copyright © 2000 by John Wiley & Sons

9.9.3 Test Application Time

As we have seen in Section 9.4.2, test application requires shifting response data for each test pattern. Thus test application time has been increased *M* times, where *M* is the number of flip-flops in the scan chain.

9.9.4 Heat Dissipation

Test application is usually done at a speed lower than the operating frequency of the circuit. However, when scan is used with built-in self-test (BIST) which is described in Chapter 11, testing is often done at speed, and in such a case the heat dissipation due to clocking all flip-flops is so excessive that it may burn the chip. At present, this problem is under investigation to minimize its impact.

9.10 Partial Scan Testing

The penalties of full scan methods can be partially alleviated by scanning some but not all flip-flops of the circuit [Trischler 1980]. Thus the circuit remains sequential and requires a sequential test pattern generator. It is assumed, however, that such an approach will minimize the cost of the DFT circuitry: overhead area and delays. A trade-off is made between added hardware, and fault coverage. Such a trade-off is shown in Fig. 9.14 for two circuits taken from ISCAS benchmark suite [Brglez 89].

The main issue in partial scan is the selection of the flip-flops to be included in the scan chain. Several studies have been conducted to make the best selection. By *best* it is meant a selection that minimizes the cost of such a DFT structure:

- The circuit is easier to test by the sequential ATPG.
- The area overhead is minimized.
- The placement of the flip-flops is such that the interconnects are minimized.
- The delays are shortened.

Easing test pattern generation seems to be the most important factor. It has been shown that the complexity of test pattern generation is a function of the number of flip-flops in feedback loops [Agarwal 1988, Cheng 1990]. It is also affected by the number of flip-flops between two scanned flip-flops. Approaches for the selection of the appropriate flip-flops to include in the scan chain are based on one of the following techniques: testability analysis [Trischler 1980, Abramovici 1991]; structural analysis [Cheng 1990, Lee 1990, Park 1992]; and test pattern generation [Ma 1988, Chickermane 1991, Parikh 1993].

Copyright © 2000 by John Wiley & Sons

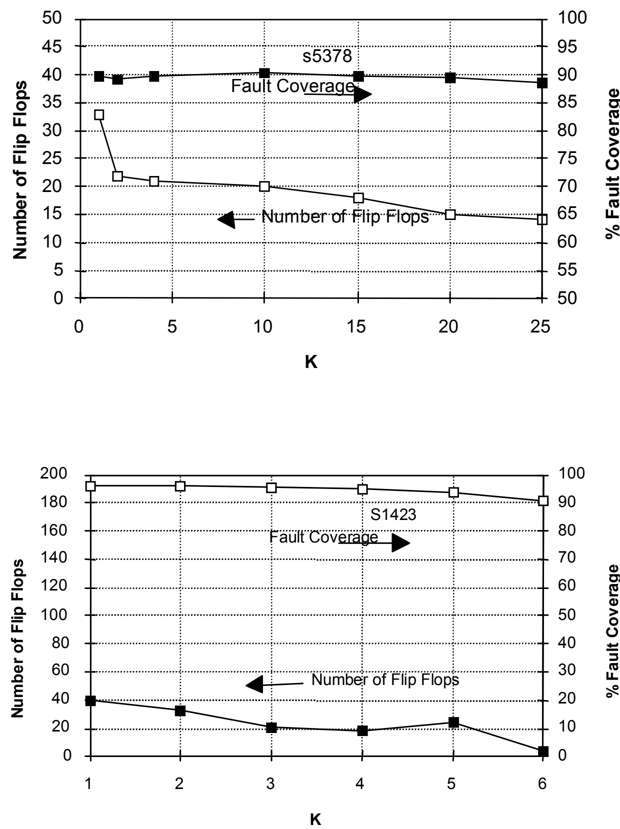


Figure 9.14 Partial Scan, Tradeoff of Fault Coverage and Hardware Overhead

Techniques based totally on testability analysis are not reliable since these measures are not necessarily good predictors of problems faced during test pattern generation [Savir 1983]. It is possible to check the contribution of each flip-flop by calculating the fault coverage when one flip-flop at a time is scanned. This approach will probably give the best results, but it is computationally expensive and it is better to use a less optimal approach that can be accomplished faster.

A more feasible approach is combining structural analysis with TM or test pattern generation. In structural analysis, the circuit is represented by a directed graph, the nodes of which are the flip-flops, with the edges being the combinational logic between the flip-flops. We will use some graph terminology that we have defined in Chapter 3 and list them here for ease of reference.

9.10.1 Definitions

An *S graph* is a cyclic graph, $G(V,E)$, that is associated with a sequential circuit. A vertex, v_i, represents a flip-flop, and the edge, (v_i,v_j), is a path between the output of one flip-flop (v_i) and the input of another (v_j).

Copyright © 2000 by John Wiley & Sons

A *path* is a sequence of edges from v_i to v_n, and its length is n-i.

A *cycle* is a path in which the starting vertex is also the final vertex. These cycles represent the feedback loops in the circuit.

The *sequential depth* of a circuit is the longest cycle in the circuit. This length causes difficulty in test pattern generation. Self-loops, feedback loops consisting of one flip-flop, are cycles of unit length. They usually do not cause the main problems in test pattern generation [Cheng 1990].

9.10.2 Selecting Scan Flip-flops

Using the graph for the circuit, the idea is to select the minimal number of flip-flops that change the S-graph to an acyclic graph. This minimal set of flip-flops is called the *minimal feedback vertex set* (MFVS). Under these conditions, it is possible to use, with some modifications, a combinational ATPG. Determining the MFVS is an NP-complete problem; therefore, heuristics are used to accomplish the task. The circuit used to illustrate scan-path test pattern application is shown in Fig. 9.3 and consists of two flip-flops, where the output of the second flip-flop feeds back into the first. Thus we have a cycle of depth 2. It is sufficient to scan one flip-flop and break the cycle. The circuit is so small that it really does not matter which flip-flop is scanned. But, in general, selecting the appropriate flip-flop is not as simple as it may seem.

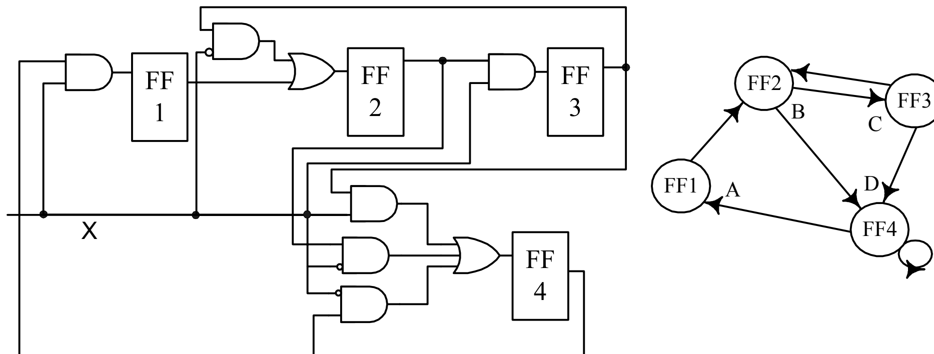


Figure 9.15 Sequential Circuit S3 and its S-Graph Representation [Park 1992]

We explain the approach using circuit S3, shown in Fig. 9.15. The figure also shows the S-graph of the circuit. The main idea is to select some of the flip-flops so that the graph becomes acyclic. The graph shows four cycles, of lengths, 1 (self-loop), 2, 3, and 4. The question is, then: What is the minimum number of flip-flops to scan (where to cut the cycles) such that the circuit becomes acyclic?

Copyright © 2000 by John Wiley & Sons

Table 9.2 Adjacency Matrix for the S-graph in Fig. 9.15

(a)

	*FF*1	*FF*2	*FF*3	*FF*4	Total
*FF*1	0	1	0	0	1
*FF*2	0	0	1	1	2
*FF*3	0	1	0	1	2
*FF*4	1	0	0	1	2
Total	1	2	1	3	

(b)

*FF*1	*FF*2	*FF*3	*FF*2T	Total
0	1	0	0	1
0	0	1	1	2
0	1	0	1	2
0	0	0	0	0
1	2	1	2	

Table 9.2*a* gives the adjacency matrix representing the S-graph in Fig. 9.15. An entry, a_{ij}, in the matrix indicates that flip-flop *i* feeds into flip-flop *j* if it is equal to 1, otherwise, it is 0. There are several approaches on how to use this matrix to search the graph for appropriate flip-flips to scan. One such approach uses the sum of indegree and outdegree for each vertex as an estimate of the number of possible loops through the vertex [Lee 1990]. This can be translated as adding the 1s in the corresponding flip-flop row and column, and then selecting the largest number. The last column of Table 9.2*a* lists all the outdegrees for the flip-flops and the last row gives the indegrees. For the example shown above, flip-flop 4 has the highest estimate of 5; this is followed by flip-flop 2. If we discount self-loops, both flip-flops would have the same number, 4. If we scan FF4, we cut all loops except that of depth 2, which is formed by FF2 and FF3. We can now update the matrix and calculate our cost function again. The updated values are shown in Table 9.2*b* and to make the circuit fully acyclic, we can then scan FF2. Notice that we reach the same result by scanning FF3. However, remember that we are using a heuristic. There is another factor that we need to enter in selecting the appropriate flip-flops to scan: the proximity of the scannable flip-flops to each other.

9.10.3 Test Application

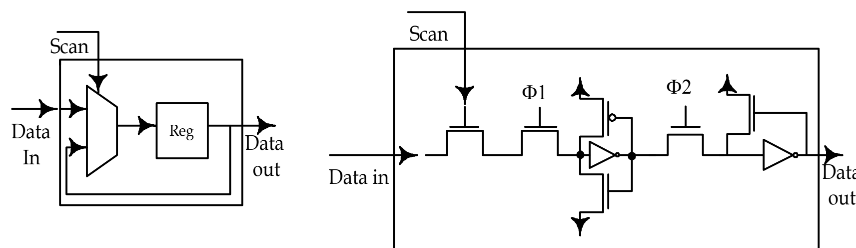


Figure 9.16 Hold Mode for Non-Scannable Flip-

Let us consider the most common approach to scan design using one clock and multiplexed flip-flop. After the test application, the system is clocked to latch the response on the flip-flops as usual. As the response is scanned out while clocking all flip-flops, the nonscanned flip-flops will latch to the scanned response. In other words, the circuit

Copyright © 2000 by John Wiley & Sons

will not remain in the same state it was in when the test pattern was applied. Thus some provision must be made to keep the non-scanable flip-flops in the same state. This can be accomplished using separate clocks for the scanable and nonscanable flip-flops. Another alternative is to add a hold-mode feature in the nonscanable flip-flops, as illustrated in Fig. 9.16 [Debaney 1994, Steensma 1993]. It is, of course, possible to use Scan-Set architecture described in Section 9.7.2 in conjunction with partial scan. In this case, no special provision has to be made to maintain the state of the circuit at scan-out time since this process is done independent of the normal operation of the circuit.

9.11 Ordering Scan Chain Flip-flops

The order of the flip-flops in the scan chain affects two of the scan-path costs discussed in the preceding section: additional length of the interconnect wires and increased testing application time. It seems that an optimal approach to solve this problem would be during circuit synthesis. We discuss this approach in Chapter 14. Here we assume that after the circuit has been synthesized, the resulting netlist can then be optimized for the order of the chain, then placed and routed. Another approach is to actually place the netlist, then find a way "to stitch" the storage device for optimal order. It is important to realize that optimizing for test application does not necessarily result in optimal routing, or vice versa.

9.11.1 Optimizing for Test Application

The most general way for this optimization is multiple chains but as we have mentioned before, this costs extra pins or multiplexers. In selecting the different devices in each parallel branch, it is better to do it after placement since the interconnect wires can be minimized. In some designs, such as datapath circuits, registers are used as output or input devices only; therefore, it is possible to take advantage of such structures to schedule multiple test sessions that minimize test application time. We illustrate this with an example taken from work done at University of Southern California [Narayanan 1992]. The circuit in Fig. 9.17 shows two arrangements of the registers for scan-path testing.

Figure 9.17 Ordering Flip-flops, Optimization for Test Application Time [Narayanan 1992]

There are five scan registers and two combinational partitions, K1 and K2, with test lengths of 50 and 300, respectively. The size of the registers is indicated in the figure and the total number of flip-flops used is 20. Assume that it is possible to merge the two tests and to apply only 300 patterns [Abramovici 1990]. Then, according to the

Copyright © 2000 by John Wiley & Sons

test application process discussed in Section 9.4, the number of clock cycles required for test application is: 300(20 + 1) + 20 = 6320. This will involve all the flip-flops in the test results simultaneously in test application time that is independent of the register order in the scan chain.

It is possible to exploit the circuit topology to use an overlapped scheme [Abramovici 1990]. Notice that registers R1 and R2 serve as input only to K1 and K2, they are called the *drivers*. Also, R4 and R5 serve as output only and they are called the *receivers*. Accordingly, R3 is a driver –receiver register. With this knowledge we

rearrange the registers so that each of the 50 common patterns is shifted in the first three registers that are of combined length 14. At the same time, the result of the previous pattern is scanned-out through the registers R3, R4, and R5 whose combined length is eight. The time taken for this test is then 50(14 + 1) = 750 cycles. The rest of the test, which consists of 250 patterns, is applied through drivers R2. This takes 12 cycles for the arrangement shown in Fig. 9.17*a*. Meanwhile, the previous pattern is scanned out through *R*3 and *R*5 as receivers. The total testing time for this session is then 250(12 + 1) = 3250. The overall total testing time is then 750 + 3250 + 14 = 4014 cycles. In this fashion, the test application time has been reduced by 34%. Notice that this reduction of test application time is accompanied by lowering the heat dissipation during testing.

This saving can be even greater if the flip-flops in the scan chain are reordered as shown in Fig. 9.17*b*. The first session will take 50(18 + 1) = 950 cycles. The 18 cycles are required because to apply the pattern through the drivers R1, R2, and R3, we have to go also through R4. For the other session, only R2 is used as a driver and both R3 and R5 are used as receivers. Then it takes only: 250(4 + 1) = 1250 cycles. The total test application time is 950 + 1250 + 18 = 2218 cycles. This is an improvement of 44.7% over the order shown in Fig. 9.17*a,* and of 66.9% over the traditional test application scheme. It is not realistic always to expect such a savings in test application time because the savinga is strongly dependent on the topology of the circuit and the length of test sets for the various partitions or segments of the combinational part of the circuit.

9.11.2 Optimizing Interconnect Wiring

Reordering the flip-flops in the scan chain to optimize on interconnect wire length is important for submicron technology chips. The delay due to the interconnect is becoming comparable to that of the logic gates. Also,

Copyright © 2000 by John Wiley & Sons

interconnects occupy an appreciable part of the chip area. Determining the order of the flip-flops in the scan chain in such a way as to minimize interconnect length cannot be done on the logic level only. It is important to consider it on the physical level as well. The problem is analogous to placement and routing that we reviewed in Chapter 4. However, while in generic placement there is no emphasis on a particular class of gates, here the focus is on minimizing the interconnect between the scanned flip-flops. The interconnect wires in this case consist of:

- The control signal SE
- The direct connection from the output of one flip-flop to the input of the next flip-flop in the chain
- The clock signal

It is possible then to perform placement to evaluate the length of the interconnect. This length can be used as a cost function in conjunction with a heuristic to reorder the position of the flip-flops. Such an approach may inadvertently cause an increase in the interconnects for the other gates of the circuit. To avoid this complication, it is also possible to decide about the order without changing the physical locations of the flip-flops after placement is completed. The problem can be considered more as a routing problem: given the locations of the various flip-flops, how to find the shortest route such that each flip-flop is visited once and only once. This can be modeled by a graph $G(V,E,W)$, where $v_i \in V$ is any of the flip-flops, (v_i,v_j), E is a connection from v_i to v_j, and w_{ij} W is the length of the edge (v_i,v_j). The adjacency matrix for this graph is such that the entries represent the weights w_{ij}. We can use a modification of a greedy algorithm such as Krusal's algorithm that we used to find the minimal spanning tree in Chapter 4. We start with the smallest weight, w_{ij}, and then add other paths that are either connected to v_i or v_j. The vertices v_i and v_j, are then removed from V so that they are no longer revisited. If the next flip-flop is v_k, the path is now v_i,v_j,v_k and we search for the next vertex that is nearest v_i or v_k. The process continues until all vertices have been visited.

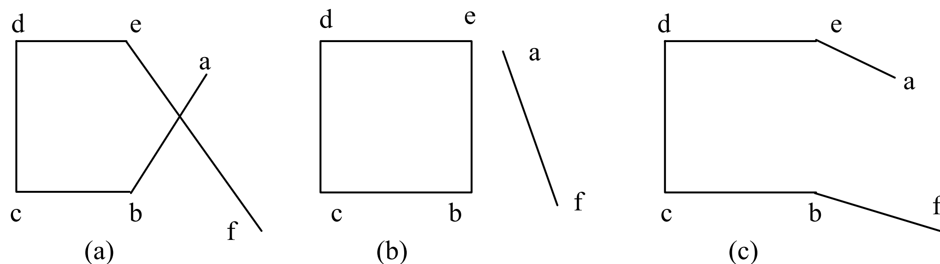


Figure 9.18 Re-routing for Minimization of Interconnect Wires [Makar 1998]

Copyright © 2000 by John Wiley & Sons

According to [Makar 1998], finding the distance between all pairs of flip-flops is problematic. For example, for only 1000 flip-flops, 1 million distance computations are required. However, any flip-flop will be connected only to, at most, two flip-flops in its neighborhood. Hence it is less computationally intensive to find the distances from one flip-flop to others in its neighborhood. The designer can determine the size of the neighborhood, which is a factor of the size of the set *V*. Once the neighborhoods are determined, it is possible to find subchains as discussed above. At the end of the process some flip-flops may not belong to any of the subchain. However, the number of such isolated flip-flops is not expected to be too large to cause a serious problem. When all subchains are merged in one path, it is possible that some of the wire segments intersect. These segments may be routed at different metal layers. Changing the order of the flip-flop from the onset is more advantageous. This is illustrated by the example shown in Fig. 9.18. In this figure there are six flip-flops, *a, b, c, d, e,* and *f*, that have been placed. One shorter stitching pattern is shown in Fig. 9.18*a*. The segments, *ef* and *ab*, intersect. To break the intersection, it is possible to connect *b* to *e* and *a* to *f*, or *e* to *a*, and *b* to *f,* as shown in Fig. 9.18*b* and *c*, respectively. The first solution is not acceptable since we obtain a loop. The second solution is acceptable and in addition, it has been proven that it results in an even shorter path.

9.12 References

Abramovici, M. and M. A. Breuer (1990), Digital Systems Testing and Testable Design, *IEEE Press,* Piscataway, NJ.

Abramovici, M., J. Kulikowski, and R. K. Roy (1991), The Best Flip-Flops to Scan, *Proc. IEEE International Test Conference*, pp. 166–172.

Agrawal, V. D., K. Cheng, D. Johnson, and T. Lin (1988), Designing circuits with Partial Scan, *IEEE Des. and test of Comput.*, Vol. 5, No. 4, pp. 8–15.

Brglez, F. et al. (1989), Combinational profiles of sequential benchmark circuits, *Proc. International Symposium on Circuits and Systems*, pp. 1929–1934.

Carter, W. C., et al. (1964), Design of serviceability features of the IBM System/360, *IBM J. Res. and Dev*, Vol. 8, No. 4, pp. 115–116.

Cheng, K., and V. D. Agrawal (1990), A partial scan method for sequential circuits with feedback, *IEEE Trans. Comput.*, Vol. C-39, No. 4, pp. 544–548.

Chickermane, V. and J. H. Patel (1991), A fault oriented partial scan design approach, *Proc. International Conference on Computer-Aided Design, pp.* 400–403.

Debaney, W. H. (1994), Quiescent scan design for testing digital logic circuits, *IEEE Dual Use Conf*erence, Session G4.

Copyright © 2000 by John Wiley & Sons

Eichelberger, E. B. and T. W Williams (1977), A logic design structure for LSI testability, *Proc. of 14th Des. Automation Conference,* pp. 462–468.

Eichelberger, E. B. (1965), Hazard detection in combinational and sequential circuits, *IBM J. Res. Dev.*, Vol. 9, No. 1, pp. 90–99.

Eichelberger, E. B. (1985), Latch design using level-sensitive scan design, Proc*. of Compcon*, pp. 380–383.

Funatsu, S., N., Wakatsuki, and A. Yamada (1978), Designing digital circuits with easily testable considerations, *IEEE Semiconductor Test Conference*, pp. 98–102.

Greene, B. and S. Mourad, (1998), *IEEE Instrumentation and Measurement Technology Conference*, pp. 423–427.

Kobayashi, A. et al. (1963) A Flip - Flop Circuit suitable for FLT (in Japanese), *Annual Meeting of the Institute of Electronics, Information and Communications Engineers*, Manuscript 892, p. 962.

Lee D. H., and S. M. Reddy (1990), On determining Scan flip-flops in Partial-Scan Designs, *Proc. ICCAD*, pp. 322–325.

LSI (1992), *Chip-Level Full Scan Design Methodology Guide,* LSI Logic Corporation, Milpitas, CA.

Ma, H. K.T., S. Devadas, A. R. Newton and A. Sangiovanni-Vincentelli (1988), An incomplete scan design approach to test generation for sequential machines", *Proc. IEEE Internaional Test Conference*, pp. 730–734.

Makar, S. (1998), A layout based approach for ordering scan chain flip-flops, *Proc. IEEE International Test Conference*.

McCluskey, E. J. (1986), A survey of design for testability scan techniques, *Summer 1986 Semicustom Design Guide*, VLSI System Design, CMP Publications, Manhasset.

Miczo, A. (1983), The sequential ATPG: A theoretical limit, *Proc. IEEE International Test Conf*erence, pp. 143–147.

Narayanan, S., C. Njinda, and M Breuer (1992), Optimal sequencing of scan registers, *Proc. IEEE International Test Conference*, pp. 293–302.

Parikh, P. S and M. Abramovici (1993), A cost based approach to partial scan, *Proc. 30th ACM/IEEE Design Automation Conference,* pp. 255–259.

Park S., and S. Akers (1992), A graph theoretic approach to partial scan design by K-cycle elimination, *Proc. IEEE International Test Conference*, pp. 303–311.

Savir J. (1983), Good controllability and observability do not guarantee good testability, *Proc. IEEE Trans. Comput.*, Vol. C-32, No. 12, pp. 1198–1200.

Stewart, J. H. (1978), Applicant of scan/set for error detection and diagnositics, *Proc. IEEE Semiconductor Test Conference*, pp. 152–158.

Steensma, J., F. Catthoor and H. De Man (1993), Partial scan at the register-transfer level, *Proc. IEEE International Test Conference*, pp. 488–497.

Trischler E. (1980), Incomplete scan path with an automatic test generation methodology, *Proc. IEEE International Test Conference,* pp. 153–162.

Williams, M. J. Y. and J. B. Angell (1973), Enhancing testability of large scale integrated circuit via test points and additional logic, *IEEE Trans. Comput.*, Vol. C-22, No. 1, pp. 46–60.

Copyright © 2000 by John Wiley & Sons

Problems

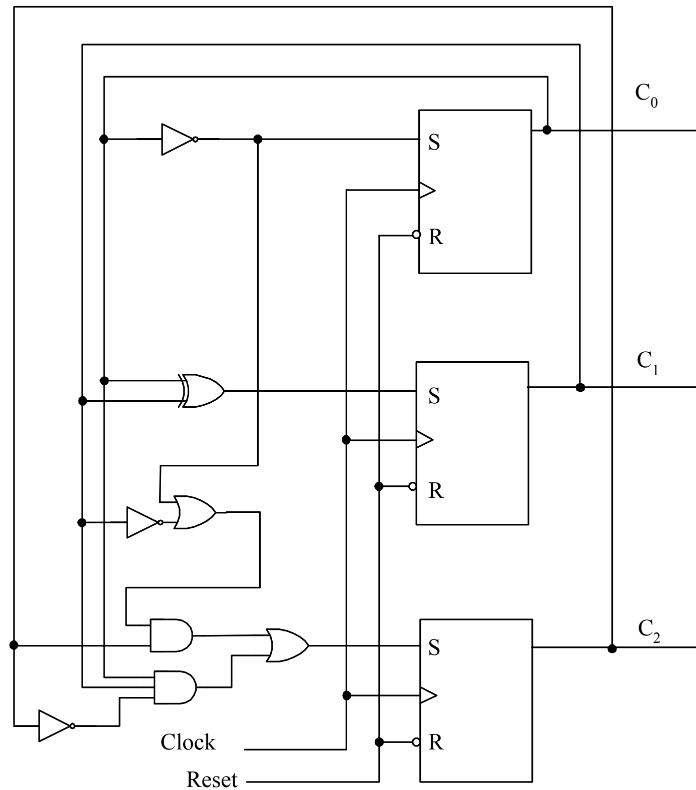


Figure 9.P1

9.1. Determine if the counter in Fig. P 9.1 has to go through an entire cycle to detect all stuck-at faults in the combinational part. How many test patterns are needed?

9.2. Repeat Problem 9.1 for the circuit shown in Fig. P9.2. Notice that the combinational circuit is the same for both circuits.

9.3. For the circuit in Fig. 9.13, how many scan-paths would you use to minimize test pattern application? Redesign the circuit using scan-path design and generate a test set to detect all stuck-at faults in the combinational part.

Copyright © 2000 by John Wiley & Sons

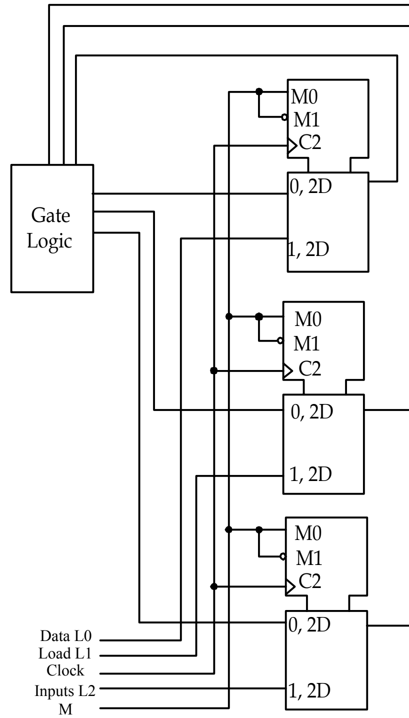


Figure 9.P2

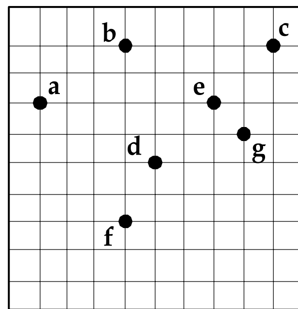


Figure 9.P3

9.4. The nodes on the grid shown in Fig. P9.4 are flip-flops that are to be stitched to form a scan path. Show how they can be connected for minimal interconnect length. Write a short algorithm to find the minimal spanning path.

9.5. Develop a test set for the example in Fig. 9.3 when only the first flip-flop is scanned. Describe in detail the application of the test.

9.6. Select the appropriate flip-flops for partial scan of the circuit in Fig. 9.15 and justify your selection. (*Hint*: Open the appropriate edges to make the S-graph acyclic.

Copyright © 2000 by John Wiley & Sons