

An Efficient Single-Latch Scan-Design Scheme

By

Uma R. Panda

Department of Electrical Engineering

McGill University

Montréal

© March 1985

An Efficient Single-Latch Scan-Design Scheme

By

Uma R. Panda

A thesis submitted to the Faculty of Graduate Studies

and Research in partial fulfillment of

Master of Engineering

Department of Electrical Engineering

McGill University

Montréal

March 1985

©by Uma R. Panda

Abstract

With the increasing complexity of logic implemented on a single Very Large Scale Integrated (VLSI) circuit chip, there is a growing problem of checking the logical behavior of the chips manufactured. The problem is particularly acute for sequential circuits, due to difficulties in initializing, controlling and observing the state of the system. A possible solution to this problem is to incorporate Scan-Path into the sequential circuits. One of these Scan-Path designs is Level-Sensitive Scan Design.

A variation of Single-Latch Level-Sensitive Scan Design is described in this thesis. The new scheme eliminates the two shift clocks, thus considerably reducing the area overhead. A '*Mode Control*' signal switches the circuit between a 'Scan Mode' and a 'Normal Mode'. The same system clocks are used in both scan and normal modes. The system performance is not degraded with the use of latches proposed in this work. Advantages and cost impact of this scheme are also discussed. Technical details of the proposed shift register latch are documented and performance improvements are identified through extensive simulation. An estimate of area overhead and performance, and a set of design rules that will result in Level-Sensitive logic are also described.

Résumé

Avec la complexité croissante des circuits à haut niveau d'intégration (VLSI), la vérification du fonctionnement logique des circuits fabriqués devient de plus en plus problématique. Les circuits séquentiels offrent le plus de difficultés, dû aux problèmes d'initialisation, de commande, et d'observation de l'état du système. Une solution aux problèmes des circuits séquentiels est un type de conception appelé 'Scan Path'. Un cas particulier de conception 'Scan Path' s'appelle 'Level Sensitive Scan design'.

Cette thèse décrit une modification apportée au type appelé 'Single-Latch Level Sensitive Scan Design'. La modification élimine les deux signaux de commande 'Shift Clocks' normalement utilisés avec 'Scan Path', ainsi réduisant la superficie de silicium additionnelle requise. Un seul signal, 'Mode Control', commande le fonctionnement du circuit (entre les modes 'Scan' et 'Normal'). Les mêmes signaux de synchronisation ('clocks') peuvent donc être utilisés pour les deux différents modes de fonctionnement. En utilisant le nouveau type de 'Latch' ('Shift Register Latch') proposé, la performance d'un circuit n'est pas détériorée. Cette thèse présente en détails, les avantages et coûts de la solution proposée, ainsi que les caractéristiques techniques concernant le nouveau type de 'Latch'. Les analyses de performances sont appuyées par des résultats de simulations. Enfin, cette thèse comprend une estimation de l'aire additionnelle requise par la modification proposée, ainsi qu'un ensemble de règles de construction requises si la solution proposée était appliquée aux circuits de type 'Level Sensitive Logic'.

Acknowledgement

The author expresses sincere thanks to Prof. V.K. Agarwal for his excellent guidance and cooperation. Appreciation is also due to Prof. N. Rumin and Dr. Janusz Rajski for their advice and help.

TABLE OF CONTENTS

	page
Abstract	iii
Résumé	iv
Acknowledgement	v
Table of Contents	vi
List of Illustrations	viii
1 Introduction	1
2 Review of Existing Schemes	5
2.1 Fault Modeling and Analysis	5
2.2 Controllability and Observability	7
2.3 Structured Design For Testability Methods	9
2.3.1 Level-Sensitive Scan Design	10
2.3.2 Scan Path Design	19
2.3.3 Scan/Set Technique	21

2.3.4	Random Access Scan	23
2.4	Variations of LSSD Scheme	26
2.4.1	Saluja's Scheme	30
2.4.2	L2* Scheme	33
3	Proposed Scheme	39
3.1	Testing of MSRL	52
3.2	Design Rules	53
3.3	Area and Performance Overhead	55
3.4	Comparison with Other Schemes	62
4	Simulation	66
4.1	CMOS Implementation	66
4.2	Physical Design Considerations	66
4.2.1	Latch Output Considerations	69
4.2.3	Latch Scan-Output Considerations	70
5	Conclusion	75
	References	77
	Appendix A	81

LIST OF ILLUSTRATIONS

	page
1 Fig. 2.1(a) Fault-Free AND Gate	6
2 Fig. 2.1(b) AND Gate with Signal Stuck-At 0	6
3 Fig. 2.2 Sequential Circuit with Clocked D Flip-Flop	11
4 Fig. 2.3 Modified Sequential Circuit	12
5 Fig. 2.4 Double-Throw Switch	13
6 Fig. 2.5 Logic Implementation of Hazard-Free Polarity-Hold Latch	16
7 Fig. 2.6 Classical Model of a Sequential Network Modified for Shifting	17
8 Fig. 2.7 LSSD Shift Register Latch	18
9 Fig. 2.8 Raceless D-Type Flip-Flop with Scan Path	20
10 Fig. 2.9 Scan/Set Logic	21
11 Fig. 2.10 Random Access Scan	24
12 Fig. 2.11 Polarity-Hold Addressable Latch	25

13	Fig. 2.12 LSSD Double-Latch Design	27
14	Fig. 2.13 Single-Latch LSSD Configuration	29
15	Fig. 2.14 Polarity-Hold PSRL	31
16	Fig. 2.15(a) Normal Mode	32
17	Fig. 2.15(b) Test Mode	32
18	Fig. 2.16 SRL with L2* Latch	34
19	Fig. 2.17 Untestable Configuration	35
20	Fig. 3.1 Latch Structure in L2* Scheme	40
21	Fig. 3.2 Steady-State Hazard Condition	41
22	Fig. 3.3 Transient Hazard Condition	42
23	Fig. 3.4 Hazard-Free SRL using L2*	44
24	Fig. 3.5(a) Proposed Scheme (MSRL)	46
25	Fig. 3.5(b) Proposed Scheme (MSRL)	47
26	Fig. 3.6(a) Truth Table for MSRL	49
27	Fig. 3.6(b) Waveforms for MSRL	50
28	Fig. 3.7 Single-Latch Structure using MSRL	51
29	Fig. 3.8 Polycell Design Layout	56
30	Fig. 3.9 Polycell Design Layout	58
31	Fig. 3.10 Layout of Scan Circuit	59

32	Fig. 4.1 CMOS Implementation of MSRL.....	67
33	Fig. 4.2 Schematic of L1-L2 Latch Pair	68
34	Fig. 4.3 Output Response Time Vs W/L Ratio of Pull-Up Device	72
35	Fig. 4.4 Output Response Time Vs W/L Ratio of Pull-Down Device	73
35	Table 3.1 Comparison of Area Overhead for MSRL and L2*, n_1 Varies	63
36	Table 3.2 Comparison of Area Overhead for MSRL and L2*, $\frac{S}{C}$ Varies	64
37	Table 3.3 Comparison of MSRL with other Schemes.....	65
38	Table 4.5 Capacitive Load Vs Circuit Response Time	74

Introduction

1. Introduction

Testing is one of the major technical problems encountered while designing LSI/VLSI chips[1, 2, 3]. This testing problem can be substantially reduced or completely solved by adopting a design discipline, such that the chips are designed for testability[4]. In addition, an appropriate design philosophy can be adopted for the higher level machine components[5] (e.g., boards, systems).

Testing of chips is carried out by applying bit patterns to stimulate the logic inputs and by comparing the actual output response to an expected one, which has been precalculated. Generation of the stimuli and responses is achieved by simulating the function of the chip[2, 6]. Such patterns can be sometimes obtained as a by-product of design verification. Generally such functional test patterns produce only a limited test coverage, since they reflect the designer's objective as simulated against a good machine[2].

To achieve a more complete test coverage, the test patterns must permit to distinguish the good machine from all possible faulty machines. Such a set of test patterns will guarantee that the machine is free of faults. One model that is widely used to represent a faulty machine is the *Stuck-At* fault model[1, 2, 3, 7]. Test patterns that can detect such *Stuck-At* faults can be derived for combinational circuits by using algorithmic methods yielding 100% test coverage. However, for sequential circuits, test pattern generation algorithms are difficult to apply, and, in practice, test coverage is generally low[8].

Introduction

It has been established that test generation for a single *Stuck-At* fault is NP complete [9]. This means that there will be a number of circuits for which test generation time is an exponential function of circuit size. However, the theory seems extremely pessimistic compared with experience of well designed algorithms [15, 16, 17]. It has been observed [9] that the computer run time to do test generation and fault simulation is approximately proportional to the number of logic gates to the power of 3. Hence, small increases in gate count will yield high-increasing run times. The following equation shows this relationship :

$$T = KN^3$$

Where T is the expected computer run time, N is the number of gates, and K is the proportionality constant. It has been observed that computer run time just for fault simulation is proportional to N^2 without the test generation phase.

To find tests for sequential circuits, the problem that must be solved is determining a test pattern sequence which brings the memory elements into a state needed for applying the tests. Historically, designers improved the testability of sequential logic by practicing different design tricks. These are called the 'Ad Hoc' methods [4]. One of these 'Ad Hoc' methods most commonly practiced provides a common reset input [10] for counter or shift register latches. The reset input is used to obtain a defined machine state. Tests could then be designed by stimulating the logic from this known state to a state needed to apply a test. An obvious disadvantage with this approach is that long test pattern sequences are required since many tests originate from the same reset state. To overcome this disadvantage, an improvement to the above approach consists of breaking counters into subsections, each of which can reset individually. This will lead to shorter test pattern sequences. In addition to reset inputs, test points can be

Introduction

added to the circuit. This will permit to apply tests more readily and observe tests results better. Designing with these approaches requires much designer ingenuity and lacks a general applicability. In addition, while chip testability may be improved, such methods often do not assist in card testing.

Another way to improve testability, a more structured approach[11], is to design a chip in a fashion such that the combinational elements and the sequential elements are partitionable. In general, this partitioning does require some additional circuits such that connection of all latches on a chip to one or more shift registers becomes possible. This creates access to the combinational logic partition and test patterns can be supplied via the shift registers. This structured approach to testable design is called 'Scan-In, Scan-Out' or more commonly, just 'Scan'[12]. One of these 'Scan' based designs is 'Level-Sensitive Scan Design' (LSSD) [13]

All the testable design methods have the same objective : to reduce the cost of testing. An empirical relationship[11] that has been used for estimating the cost of finding a faulty device is that the cost will increase by a factor of 10 as fault-finding moves from one level to the next, i.e., if it cost \$0.30 to detect a fault at the chip level, then it would cost \$3 to detect that same fault when it was imbedded at the board level; and \$30 when it was imbedded at the system level, and \$300 when it is imbedded at the system level but has to be found in the field. Thus, if a fault can be detected at a chip or board level, then significantly larger costs per fault can be avoided at subsequent levels of packaging.

In VLSI, the inadequacy of automatic test pattern generation and fault simulation, makes it difficult to obtain a level of testability required to achieve acceptable defect levels. If the defect level of boards is too high, the cost of field repairs is also too high.

Introduction

These costs, and in some cases, the inability to obtain a sufficient test, have led to the need to have 'Design For Testability'.

In this thesis report, Chapter 2 gives a review of existing structured Design For Testability (DFT) Schemes. Chapter 3 describes the proposed scheme, which is a variation of 'Level-Sensitive Scan Design'. Chapter 4 presents one way of implementing a circuit design of the proposed 'Modified Shift Register Latch' (MSRL) with detailed simulation results.

Review of Existing Schemes

2. Review of Existing Schemes

With the utilization of LSI and VLSI technology, it has become apparent that testability has to be considered as a design parameter[11]. This has led to rigorous and highly structured design practices. Most structured design practices are built upon the concept that if the values in all the latches can be controlled to any specific value, and if they can be observed with a very straight forward operation, then the test generation, and possibly the fault simulation task, can be reduced to that of doing test generation and fault simulation for a combinational logic network. A control signal can switch the memory elements from their normal mode of operation to a mode that makes them controllable and observable[14]. This chapter describes the basic concepts in testing, beginning with the fault models and carrying through to the different variations of Level-Sensitive Scan Design [13] which was proposed by IBM.

2.1 Fault Modeling And Analysis

A model of faults which does not take into account all possible defects, but is a more global type of model, is the *Stuck-At* model[1, 2, 3, 7]. This is the most widely used model. The *Stuck-At* model assumes that a logic gate input or output is fixed to either a logic 0 or a logic 1.

For example, consider the fault-free AND gate G shown in fig. 2.1(a). Fig 2.1(b) shows the same gate with its input A stuck at zero. In the presence of a stimulus, $A=1$ and $B=1$, the faulty gate output is $C=0$, and the fault-free output is $C=1$. The inputs conditions that cause a faulty circuit to behave differently from a good circuit are

Review of Existing Schemes

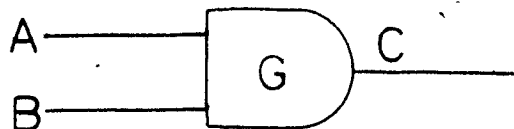


Fig. 2.1(a) Fault-Free AND Gate

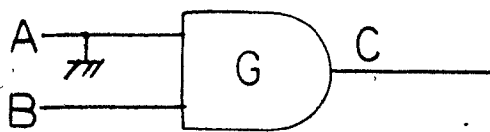


Fig. 2.1(b) AND Gate with Signal Stuck-At 0

Review of Existing Schemes

considered to be a test for that particular fault. In this case, the fault (A stuck at zero) is detected by the test $A=1, B=1$.

Test generation techniques are available for strictly combinational circuits[1]. A combinational circuit containing N inputs can use 2^N patterns, to verify each entry in the truth table. For larger networks, heuristic or random patterns can produce an acceptable starting point for test generation. But very large and complex designs require a deterministic or algorithmic approach. The most widely used test-generation algorithm is the D-algorithm[15]. This algorithm is still the basis for most of the techniques in use today. Many recent algorithms such as PODEM[16] and FAN[17], however seem to have a better performance than the D-Algorithm.

Unfortunately, sequential circuits rapidly increase the complexity of test generation and, in VLSI designs, make the test generation nearly useless. Many of the approaches that use structured testability reduce the test-generation problem by converting sequential circuits into combinational ones in a 'test' mode.

Finally, several physical circuit failures depend on the technology used. These failures often cannot be described by the single *Stuck-At* model. For example, bridging faults [18] produce behavior that cannot be modeled with a single stuck at model. Some failures in CMOS devices can cause a combinational network to behave like a sequential element[19].

2.2 Controllability And Observability

There are two key concepts in testability: Controllability and Observability. Controllability[12] is defined as the ease of setting a particular internal logic node to either

Review of Existing Schemes

logic 1 or logic 0. Observability [12] is defined to be the ease of observing the response of an internal logic node. An internal node is controlled from the primary inputs and observed at the primary outputs and the process of test-pattern generation relies on the ability both to control and to observe each node in the circuit. A measure for nodal testability can therefore be quantified in terms of nodal controllability and nodal observability values. Circuit testability [20] can then be determined from a knowledge of the circuit.

Control and observation of network nodes are central to implementing test procedure. For example, considering the case of fig. 2.1(a), in order to be able to test the A input *Stuck-At* 1, it was necessary to control the A input to 0 and the B input to 1 and be able to observe the C output to determine whether a 0 was observed or a 1 was observed. The 0 is the result of the good machine, and the 1 would be the result of faulty machine. If this AND block is embedded into a much larger sequential network, the requirement of being able to control the A and B inputs to 0 and 1, respectively, and being able to observe the output C, through some other logic blocks, still remains. Therein lies part of the problem of being able to generate tests for a network

In essence, therefore, testability measures based on controllability and observability features are really only a measure of the ease of generating test patterns. Because of the need to determine if a network has the attributes of controllability and observability that are desired, a number of programs [21, 22, 23] have been written which essentially give analytic measures of controllability and observability for different nodes in a given sequential network. Of necessity, such measures can only produce coarse results since, in reality, the only real measure of testability is the cost of producing an adequate set of tests for the circuit. Nevertheless, there would seem to be many uses [12] for such measures, such as:

Review of Existing Schemes

- (a) Advising on the better of two designs (a revised design may have a higher testability rating than original);
- (b) Allowing a judicious selection of test points (nodes with low observability are obviously good candidates);
- (c) Identifying potentially difficult nodes to test (low controllability and observability)

However, to be useful, a testability measure should be inexpensive to compute in comparison with the costs of deriving the tests.

2.3 Structured Design For Testability Methods

The most widely accepted structured testability technique is the Level-Sensitive Scan Design (LSSD) method proposed by Eichelberger and Williams [13]. This method is actually a combination of two separate design strategies. Level Sensitivity implies the operation of a logical network that is independent of internal circuit delays and primary input skew. Design rules are specified to guarantee this effect. Scan design embodies two functions for all sequential circuit elements. An auxiliary mode is available that lets all memory devices be connected as a shift register. With this connection, testing the sequential devices becomes a matter of simply shifting alternating sequences of 1's and 0's through the register and verifying the patterns at the output stage. One immediate benefit of scan design is that it reduces the testing problem to that of testing the remaining combinational logic of the circuit.

The shift register modification approach was first presented by Williams and Angell in 1973 [14]. This approach uses clocked D flip-flops as the storage elements, as shown in

Review of Existing Schemes

fig. 2.2. The structure of 'Shift Register Modification' approach is illustrated in fig. 2.3. The modification is done by inserting a double-throw switch at each input lead of every flip-flop, and in the lead that drives one of the primary outputs of the circuit. Each of the double-throw switches may be implemented as shown in fig. 2.4. The modified sequential circuit can operate either in its normal mode or shift register mode. When the mode signal is set to a 0, the circuit operates in the normal mode, i.e., it behaves exactly as it did before modifications were carried out. When the mode signal is set to 1, all flip-flops in the circuit are connected in a chain and form a shift register. In this shift register mode, the first flip-flop can be set directly from a primary input, and the output of the last flip-flop can be directly monitored at a primary output. Hence, the modified circuit can easily be set to any desired internal state by supplying the corresponding values to the shift register and further the internal state of the circuit can easily be observed by shifting out the contents of the shift register.

The other widely accepted structured techniques are generally very similar to LSSD. Scan Path design[24], proposed by Nippon Electric Company (NEC), implements the scan register by using D-type flip-flops. Scan/Set technique [25], put forth by Sperry-univac has a shift register path, but these shift registers are not in the data path. In the Random Access Scan[26], proposed by Fujitsu, shift registers are not employed but an addressing scheme is provided which allows each latch to be either controlled or observed. This section takes a closer look at all the above mentioned structured testability methods.

2.3.1 Level-Sensitive Scan Design

Level-Sensitive Scan Design (LSSD) introduced by Eichelberger and Williams [13] ensures race-free system operation as well as race-free testing. To provide reliable

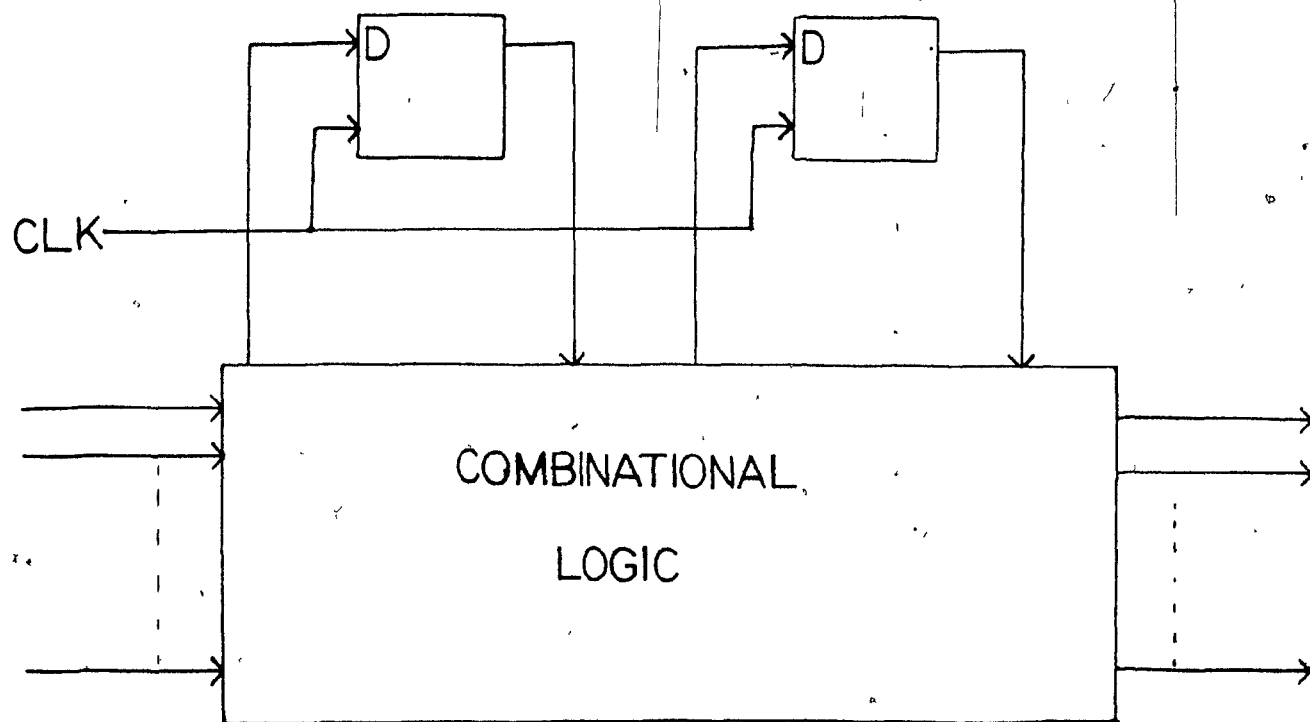


Fig. 2.2 Sequential Circuit with Clocked D Flip-Flop

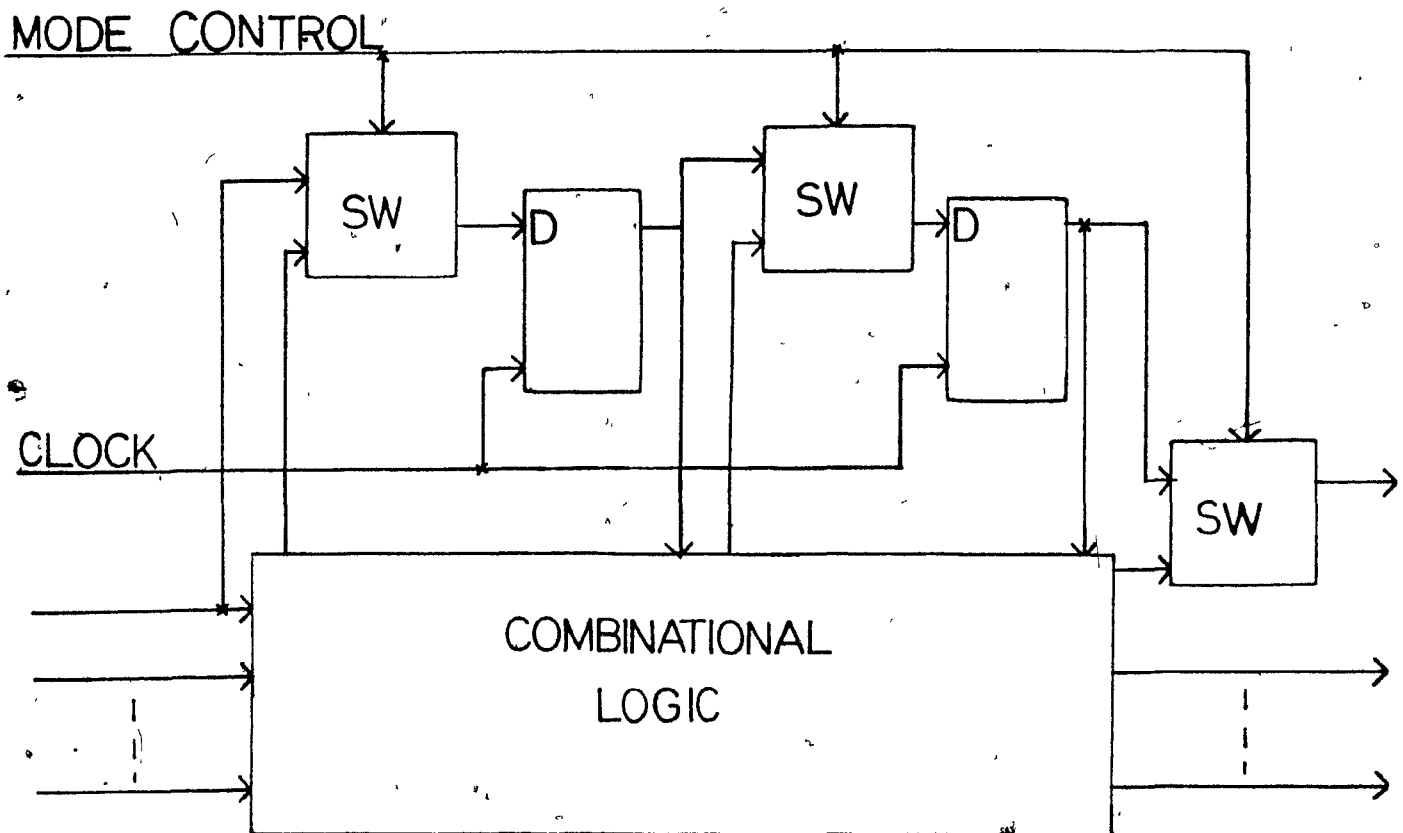


Fig. 2.3 Modified Sequential Circuit

Review of Existing Schemes

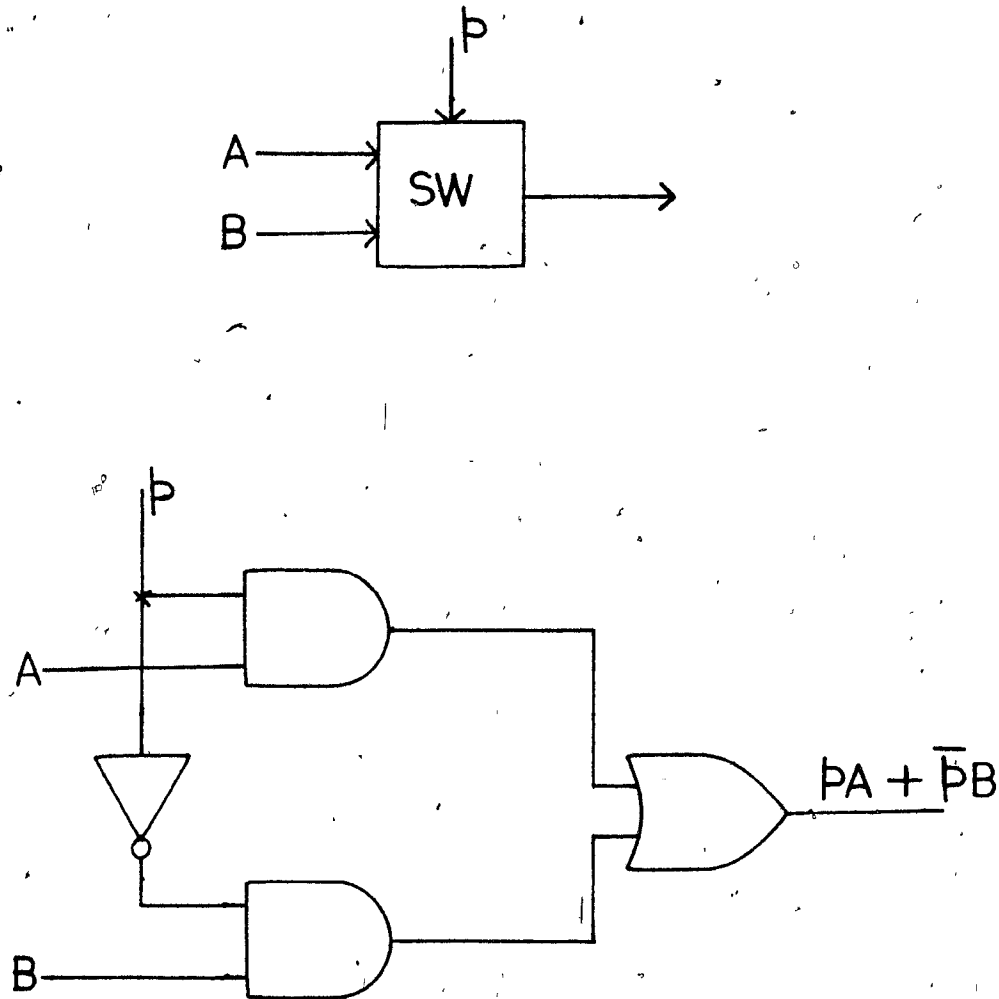


Fig. 2.4 Double-Throw Switch

Review of Existing Schemes

operation of a circuit, the designer must consider testing several ac design parameters such as rise time, fall time and delay. However, in LSI/VLSI it will become impossible or impractical to test all the ac design parameters in each circuit. The LSSD approach aims at obtaining logic circuits that are insensitive to those ac characteristics. The term 'Level-Sensitive' is defined by Eichelberger and Williams[13] as follows:

'A logic subsystem is Level-Sensitive if and only if the steady-state response to any allowed input state change is independent of the circuit and wire delays within the subsystem. Also, if an input state change involves the changing of more than one input signal, then the response must be independent of the order in which they change. Steady-state response is the final value of all logic gate outputs after all change activity has terminated.'

It is clear from this definition that level-sensitive operation is dependent on having only 'allowed' input changes. Thus, a level-sensitive design method will, in general, include some restrictions on input changes and are applied mostly to the clock signals. Other input signals have almost no restrictions on when they may change.

A level-sensitive subsystem is assumed to operate as a result of a sequence of allowed changes to allow the subsystem to stabilize in the new internal state. This time duration is normally ensured by means of clock signals that control the dynamic operation of the logic network.

A principle objective in establishing design constraints is to obtain logic subsystems that are insensitive to ac characteristics such as rise time, fall time, and minimum circuit delay. Consequently, the basic storage element should be a level-sensitive device that does not contain a hazard or race condition.

Review of Existing Schemes

The polarity-hold latch [13] as shown in fig. 2.5 has two input signals. When $C=0$, the latch cannot change state. When $C=1$, the internal state of the latch is set to the value of the excitation input D . Under normal operating conditions, the clock signal C is 0 during the time when the excitation signal D may be changed. This prevents the changing of D from immediately altering the internal state of the latch. The clock signal will normally occur after the excitation has become stable at either a 1 or a 0. This causes the latch to be set to the new value of the excitation signal when the clock signal occurs. The correct changing of the latch is dependent not on the rise or fall time of the clock signal, but only on the clock signal's being 1 for a period equal to or greater than T_0 , where T_0 is the time required for the signal to propagate through the latch and stabilize. This polarity-hold latch is further augmented to include shift capability.

With the concept that the memory elements in an integrated circuit can be threaded together into a shift register the memory element values can be both controlled and observed. Fig. 2.6 shows the familiar generalized sequential circuit model [11] modified to use a shift register. This technique enhances both controllability and observability, allowing to augment testing by controlling inputs and internal states, and easily examining internal state behavior.

Fig. 2.7 shows the latch called the Shift Register Latch (SRL) which is used in the LSSD as the basic memory element. The polarity-hold SRL [13] consists of two latches, $L1$ and $L2$, which have the scan input I , the data input D , the system clock C and two shift control inputs, A and B . In the normal operation mode, the shift signals A and B are both set to 0 and the $L1$ latch operates exactly like a polarity-hold latch. The clock signal C is 0 during the time when the data input D may be changed. After the data input has become stable at either a 1 or a 0, the clock C will change to 1, which causes the $L1$ latch to be set to the value of the data input D .

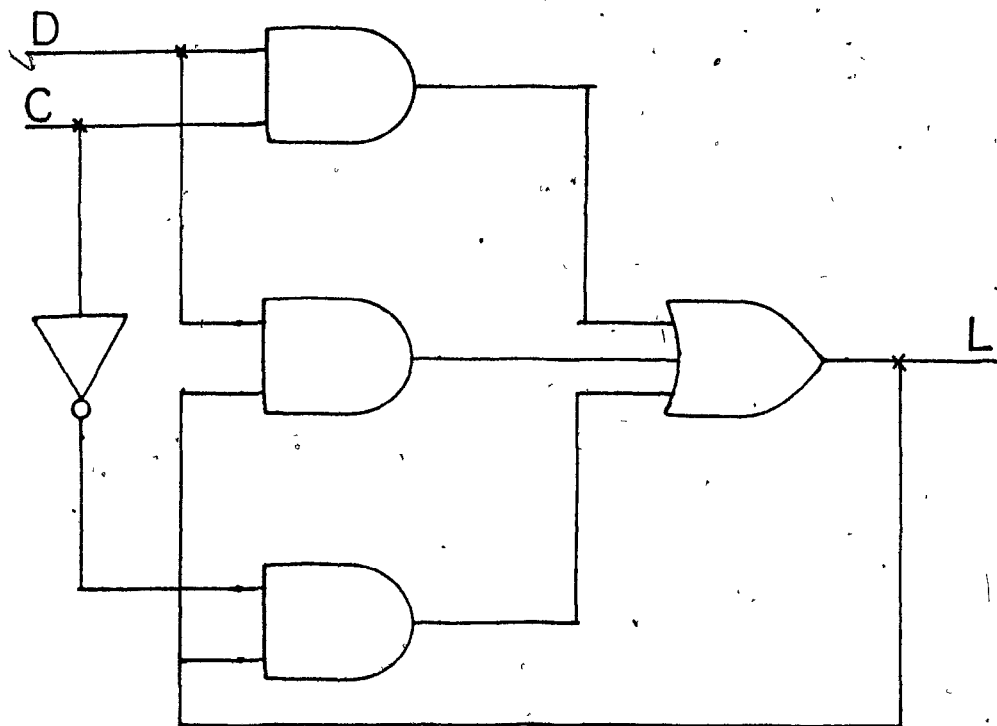


Fig. 2.5 Logic Implementation of Hazard-Free
Polarity-Hold Latch

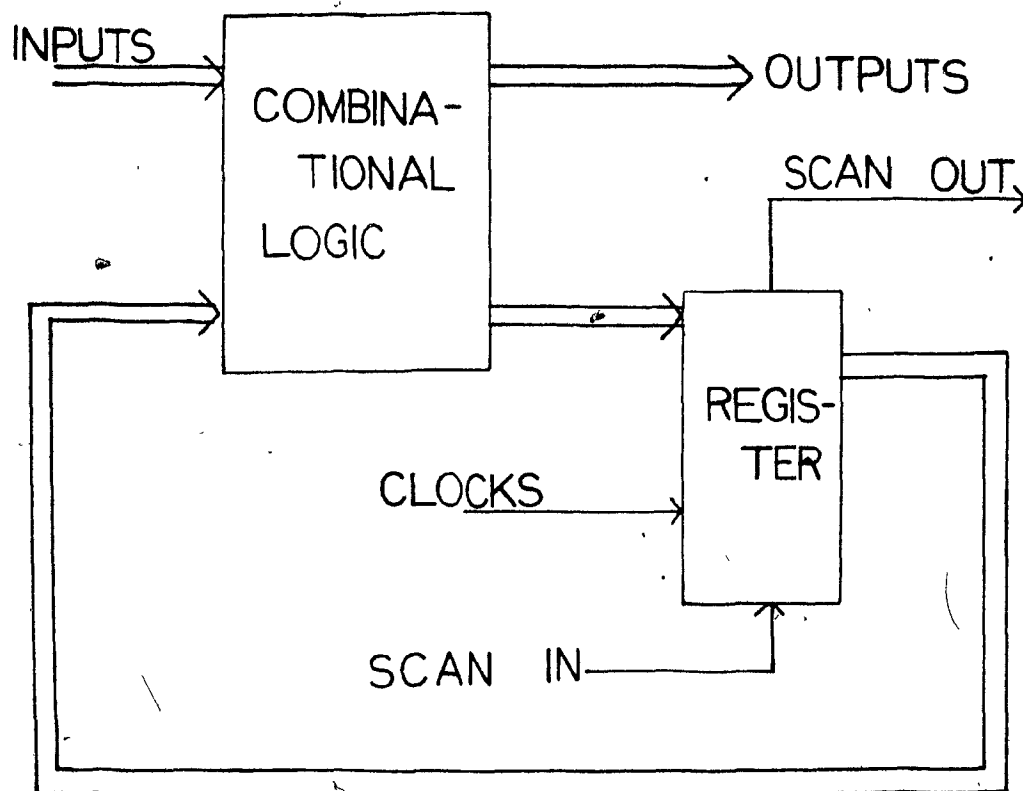


Fig. 2.6 Classical Model of a Sequential Network
Modified for Shifting

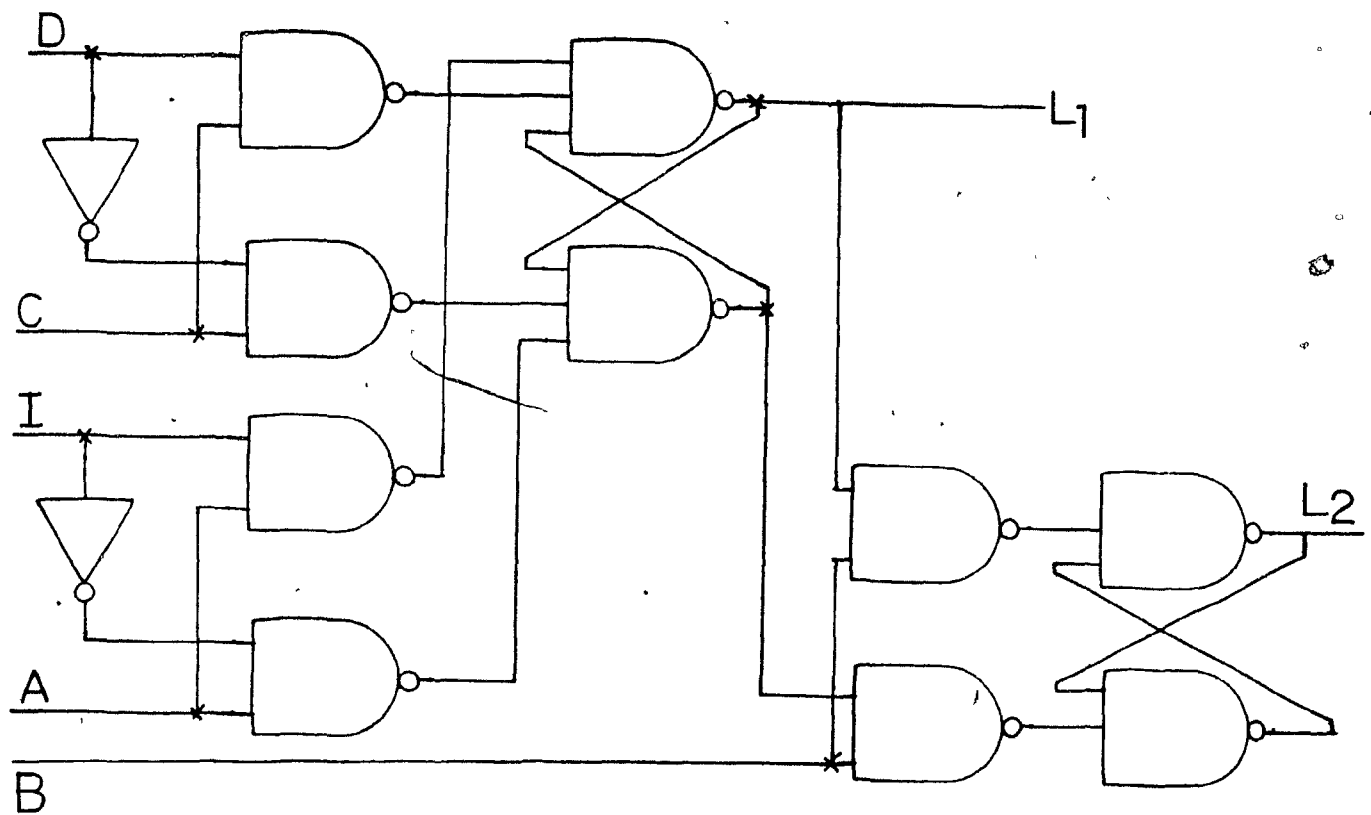


Fig. 2.7 LSSD Shift Register Latch

Review of Existing Schemes

In the shift register mode, the clock C is set to 0 and both shift signals A and B are alternately changed to shift data through latches L1 and L2. First, by changing A to 1, data from the preceding stage can be loaded into the latch L1 through the scan input I. Then after A has changed back to 0, the B shift signal changes to 1 to load the data from latch L2. Output L2 of latch L2 is connected to scan input 'I' of the next stage SRL.

Eichelberger and Williams[13] presented a set of design rules or constraints that will result in level-sensitive and scan design. These rules are given in Appendix A. Whether a logic circuit is designed in compliance with these rules can be automatically checked by a method developed by Godey etc[27].

2.3.2 Scan Path Design

The objectives of the Scan Path [24] technique are the same as the LSSD approach which has been described above. The memory elements that are used in the Scan Path approach are raceless D-type flip-flops and are shown in Fig. 2.8. In normal mode of operation, clock 2 is at a logic 'high' for the entire period. This prevents the test input from affecting the data in the first latch. Also, by having clock 2 at a logic 'high', the data in latch 2 is not disturbed. Clock 1 is the sole clock in system operation for this D-type flip-flop. When clock 1 is at logic 'low', the system data input can be loaded into latch 1. Clock 1 should be 'low' for sufficient time to latch up the data. As clock 1 turns 'high', latch 2 is sensitive to the data output of latch 1. As long as clock 1 is 'high' so that data can be latched up into latch 2, reliable operation will occur. This assumes that the output of latch 2 does not come around and feed the system data, input to latch 1 and change it during the time that the inputs to both latch 1 and latch

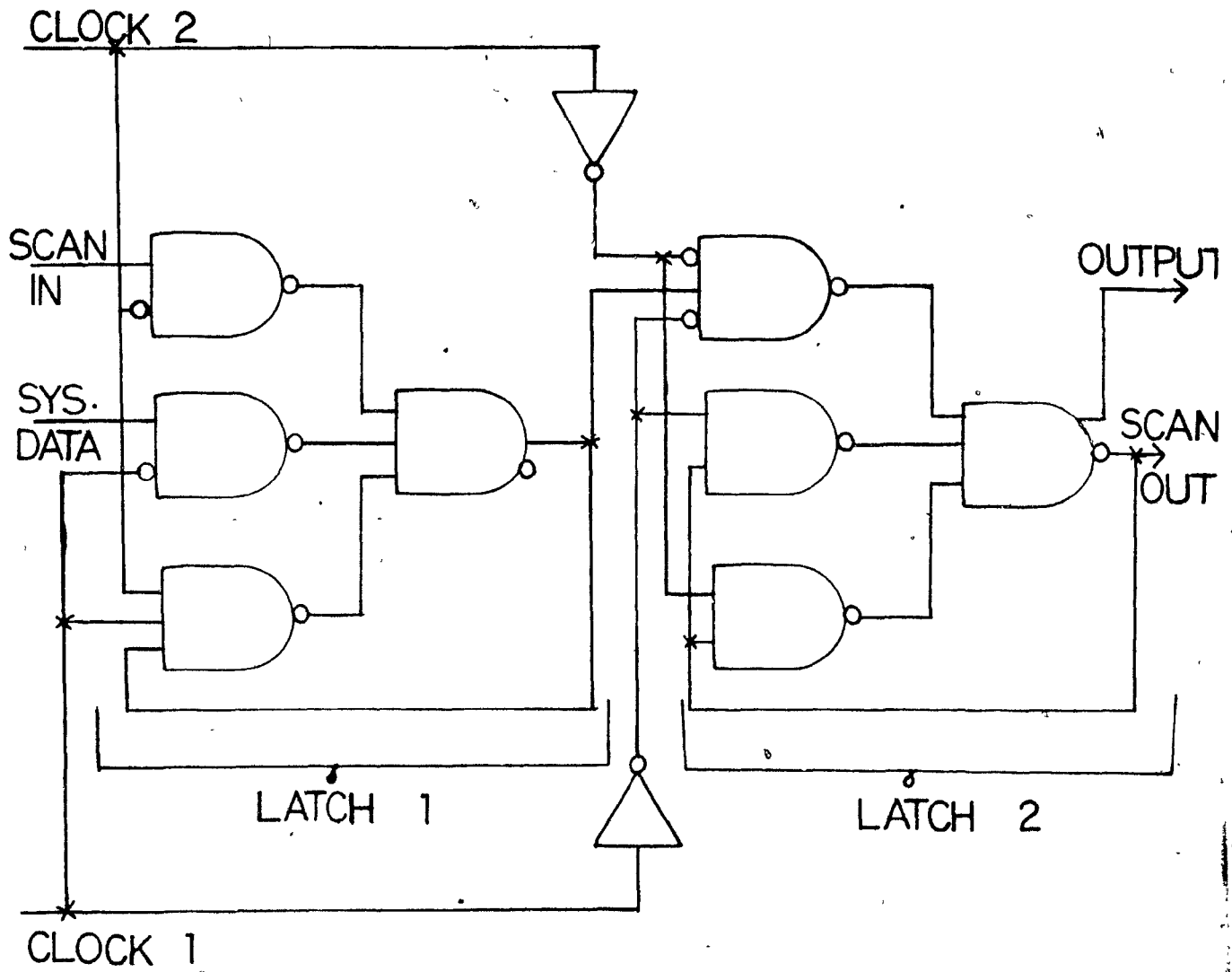


Fig. 2.8 Raceless D-Type Flip-Flop with Scan Path

Review of Existing Schemes

2 are active. The period of time when this can occur is related to the delay of the inverter for clock 1. A similar phenomena will occur with clock 2 and its associated inverter. This race condition comes from the use of only one system clock. In the scan mode of operation, the scan input is clocked into the L1 latch by clock 2, when clock 2 is 'low' and the result of the L1 latch is clocked into latch 2 when clock 2 is 'high'. Other than the lack of the Level-Sensitive property, the Scan Path approach is very similar to the LSSD technique.

2.3.3 Scan/Set Technique

The basic concept of the Scan/Set [25] technique is to have shift registers, as in Scan Path or in LSSD, but these shift registers are not in the data path. Fig. 2.9 shows an example of the Scan/Set logic. The basic concept is that the sequential network can be sampled at up to 64 points. These points can be loaded into the 64-bit shift register with a single clock. Once the 64 bits are loaded, a shifting process will occur, and the data will be scanned out through the scan-out pin. In the case of the Set function, the 64 bits can be transferred into the system logic, and then the appropriate clocking structure required to load data into the system latches is required in the system logic.

An advantage of this technique is that the scan function can occur during system operation, the sampling pulse to the 64-bit serial shift register can occur while system clocks are being applied to the system sequential logic, so that a snapshot of the sequential logic can be obtained and off-loaded without any degradation in system function.

Review of Existing Schemes

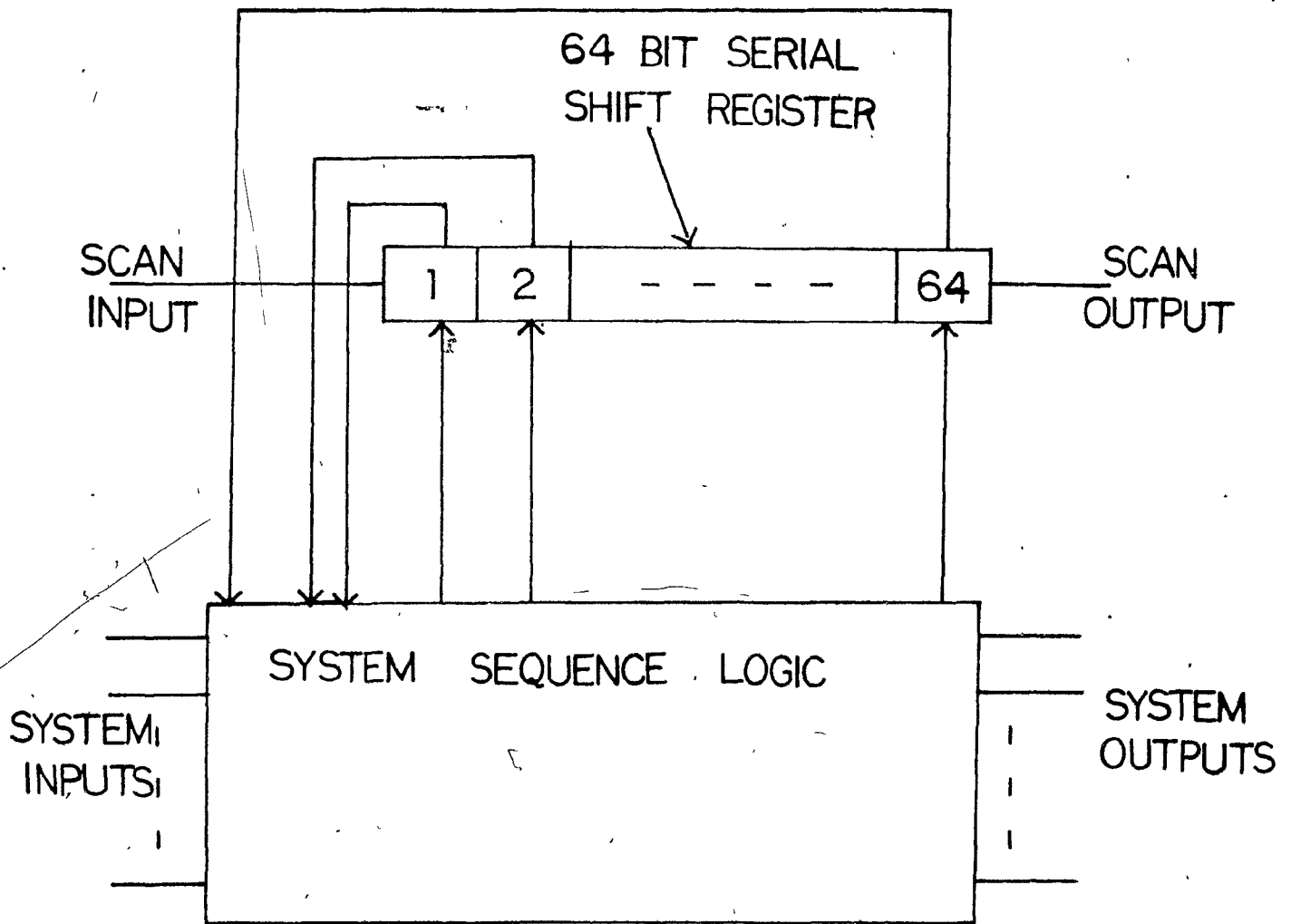


Fig. 2.9 Scan/Set Logic

Review of Existing Schemes

2.3.4 Random Access Scan

The principle objective of the Random Access Scan [26] design technique is to allow each stored-state device to be separately addressed in order that it can be independently set or preset, or its output value observed. Fig. 2.10 illustrates the principle of the technique and Fig. 2.11 shows one particular implementation of the stored-state device. In Fig. 2.10, each latch is individually selected via the decoded output of the scan-address register

Fig. 2.11 shows an implementation of an addressable latch for use in a Random Access Scan environment. Normal operation requires the Scan Clock (SCLK) to be held low, in which case changes in System Data (D) are transferred through to Q when the System Clock (CLK) is low. The last value on D is latched as CLK goes low to high. Scan operation is controlled similarly by the Scan Clock (SCLK) and requires the System Clock (CLK) to be held high. When the latch is selected, the latch output can be set to the value on Scan Data In (SDI) or the latched value observed on Scan Data Out (SDO).

Random Access Scan differs in one respect from the basic scan-path philosophy insofar as it does not contain a scan path as such. Individual SDO lines are normally high (for non-addressed latches) and can be tied together and brought out as a single Scan Output (SO) line. If the selected latch has $Q = 0$, then there is no change in the observed SO value. If the selected latch has $Q = 1$, then the SDO value will go low, pulling the main SO line low. The SDO values of all latches are determined by cycling through all addresses

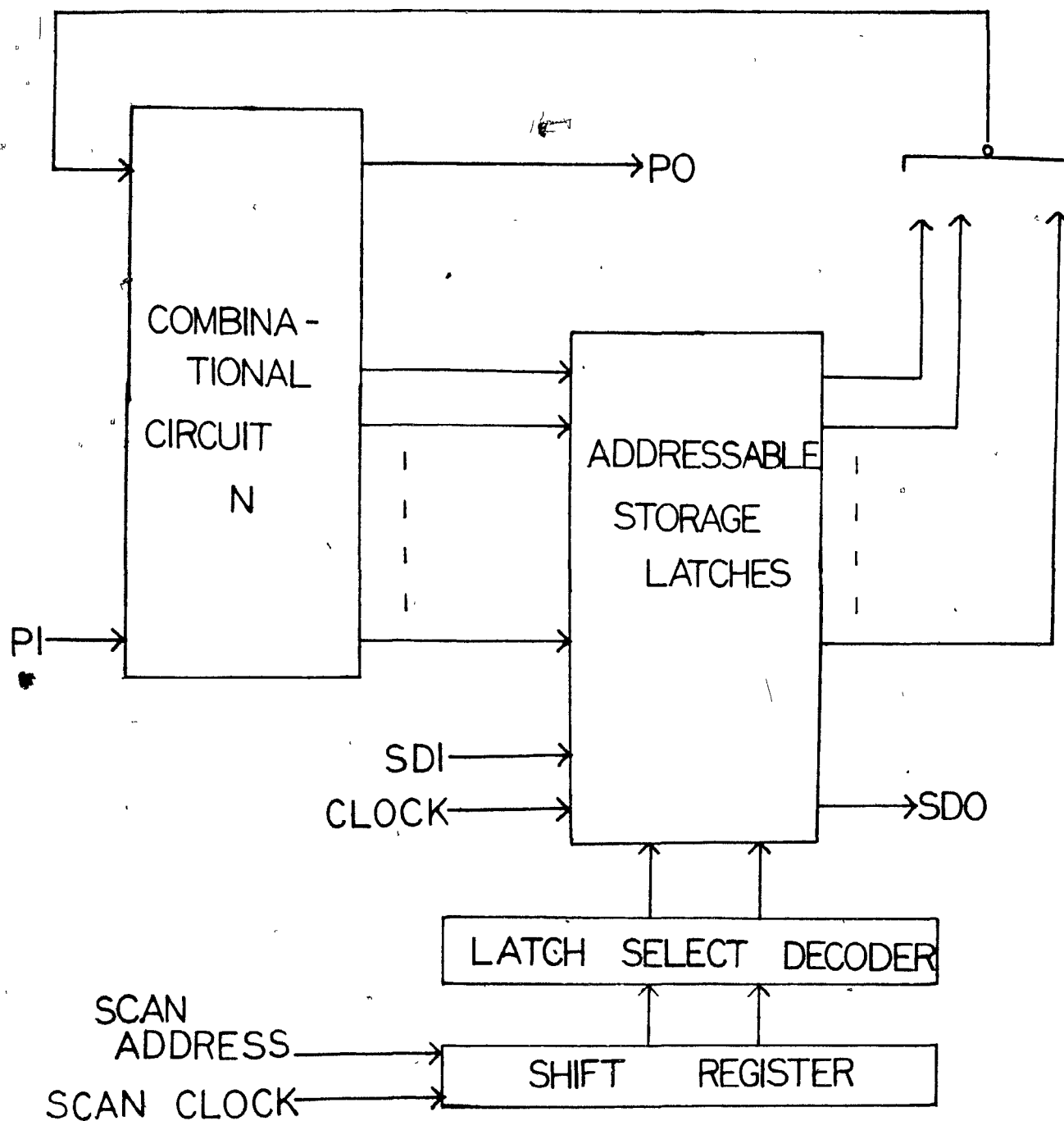


Fig. 2.10 Random Access Scan

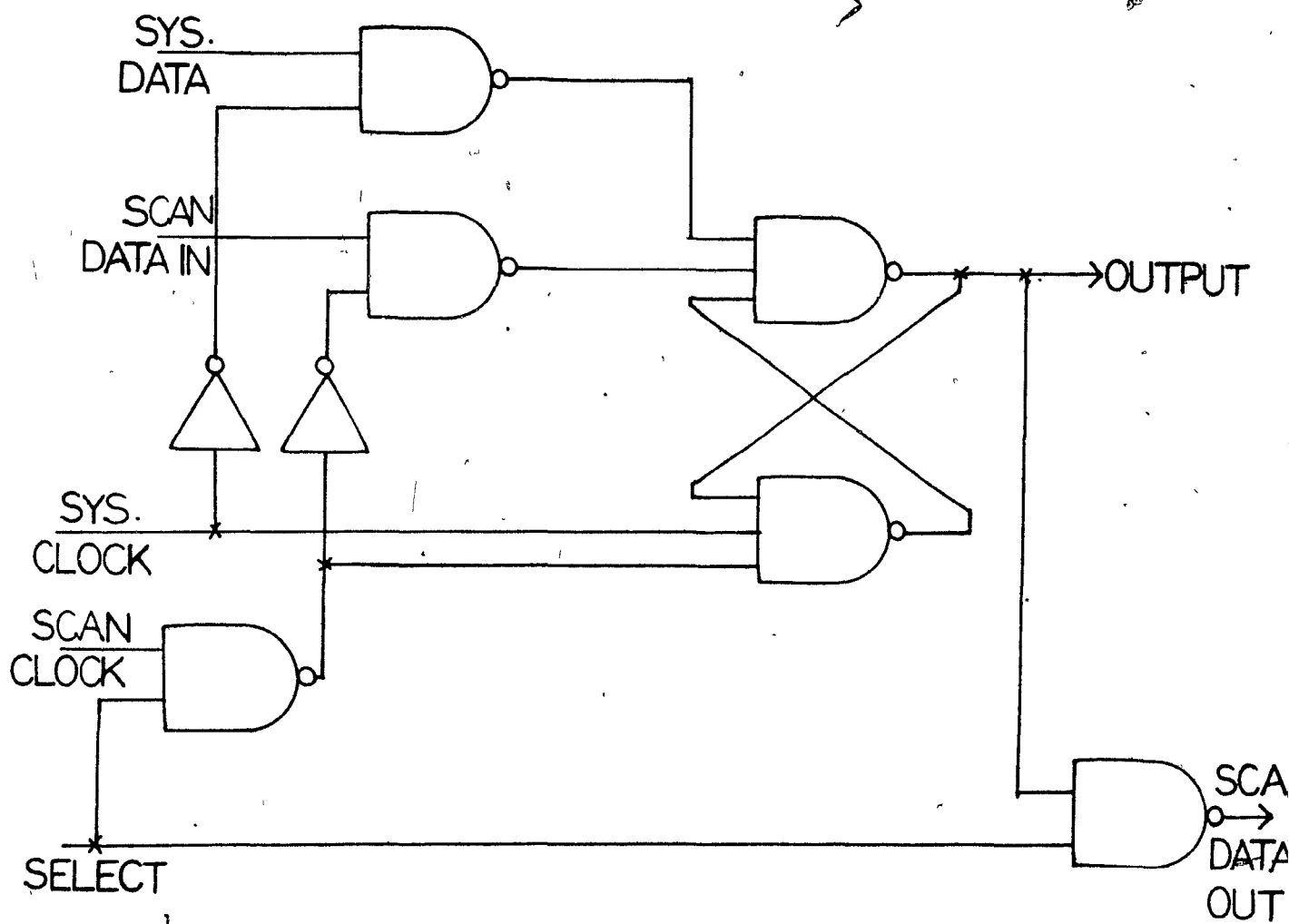


Fig. 2.11 Polarity-Hold Addressable Latch

Review of Existing Schemes

The major penalty with the Random Access Scan approach is the amount of time necessary to set the test input values into the latches and subsequently to observe the latched response. Also the overhead in additional gates is relatively high.

2.4 Variations of LSSD Scheme

In a practical LSSD circuit, the Shift Register Latches (SRL) are connected permanently to form a scan-path shift register by connecting the L2 output of one SRL to the Scan In (SI) of another SRL. The two scan clocks, A and B, are common to all SRLs. Fig. 2.12 shows a general structure for a logic circuit that follows the LSSD rules. The circuit in fig. 2.12 is called 'double-latch' design, since both latches are in the system path.

All storage elements are implemented as a set of master-slave latches L1 and L2. Each of the master-slave latches is connected in series and clocked by two non-overlapping clocks C1 and C2, where C2 is equivalent to B. At C1 time, C2 is zero and the inputs and outputs of N are stable. Some of the L1 latches change their states while C1 is 1. As soon as C1 is changed back to 0, the next clock C2 occurs, i.e., C2 changes to 1. The values of the L1 latches are loaded into the L2 latches while C2 is 1.

In the shift register mode, the SRLs are chained to form a shift register under the control of clocks A and B. Test patterns are applied to the combinational circuit by scanning them into the shift register and applying them at the primary inputs. Then the clock C1 is set to 1 and the response of the combinational circuit is captured in the L1 latches and at the primary outputs. The result of the test captured in the register

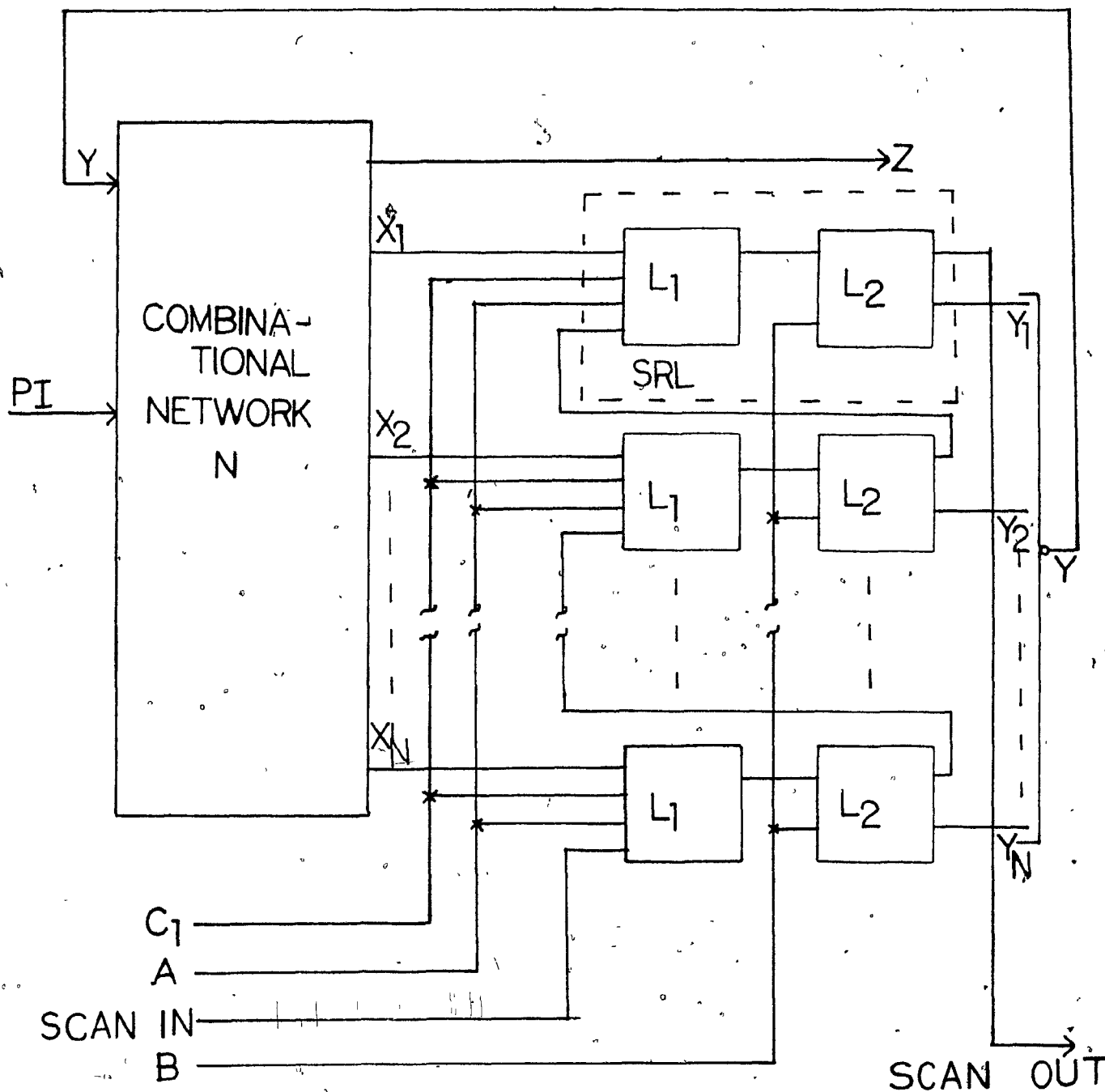


Fig. 2.12 LSSD Double-Latch Design

Review of Existing Schemes

is then scanned out. Race-less behavior is therefore guaranteed in either mode of operation.

Fig. 2.13 shows an alternative way of using SRLs in an LSSD environment, called the 'Single-Latch' configuration[12]. This configuration makes use of the L1 output as the system output and avoids the potential race condition by partitioning the combinational logic into two disjoint sets, denoted N1 and N2 in fig. 2.15. System clocks into the N1 and N2 SRLs are denoted C1 and C2 respectively. The outputs of the SRLs associated with N1 become the secondary variable inputs to N2, and vice versa. System operation is controlled by the two system clocks, C1 and C2, which operate in such a way as to ensure that only one clock is active (high) at any one time, i.e., C1 and C2 are non-overlapping. In this way potential race conditions are avoided. The name 'single-latch' comes from the fact that only one latch is used in the system path at a time.

The essential difference between the double-latch and single-latch configuration lies in the speed with which the circuit primary outputs can change as a result of primary input and clock changes. The double-latch system requires two independent and non-overlapping clocks (C and B) to change before signal-value changes can be propagated through the L1 and L2 latches and hence through the combinational circuit N to produce a stable primary output value. The single-latch configuration on the other hand only requires the appropriate single clock to change (C1 or C2) to cause propagation through the L1 latch before the appropriate combinational circuit outputs (N1 or N2 respectively) can stabilize. In both cases, the fastest operating speed is governed by the propagation delay of the combinational logic circuit. If this delay is denoted by $N\text{-delay}(\max)$, then the maximum clock rate on the system clock, C for double-latch and C1 or C2 for single-latch, is given by:

Review of Existing Schemes

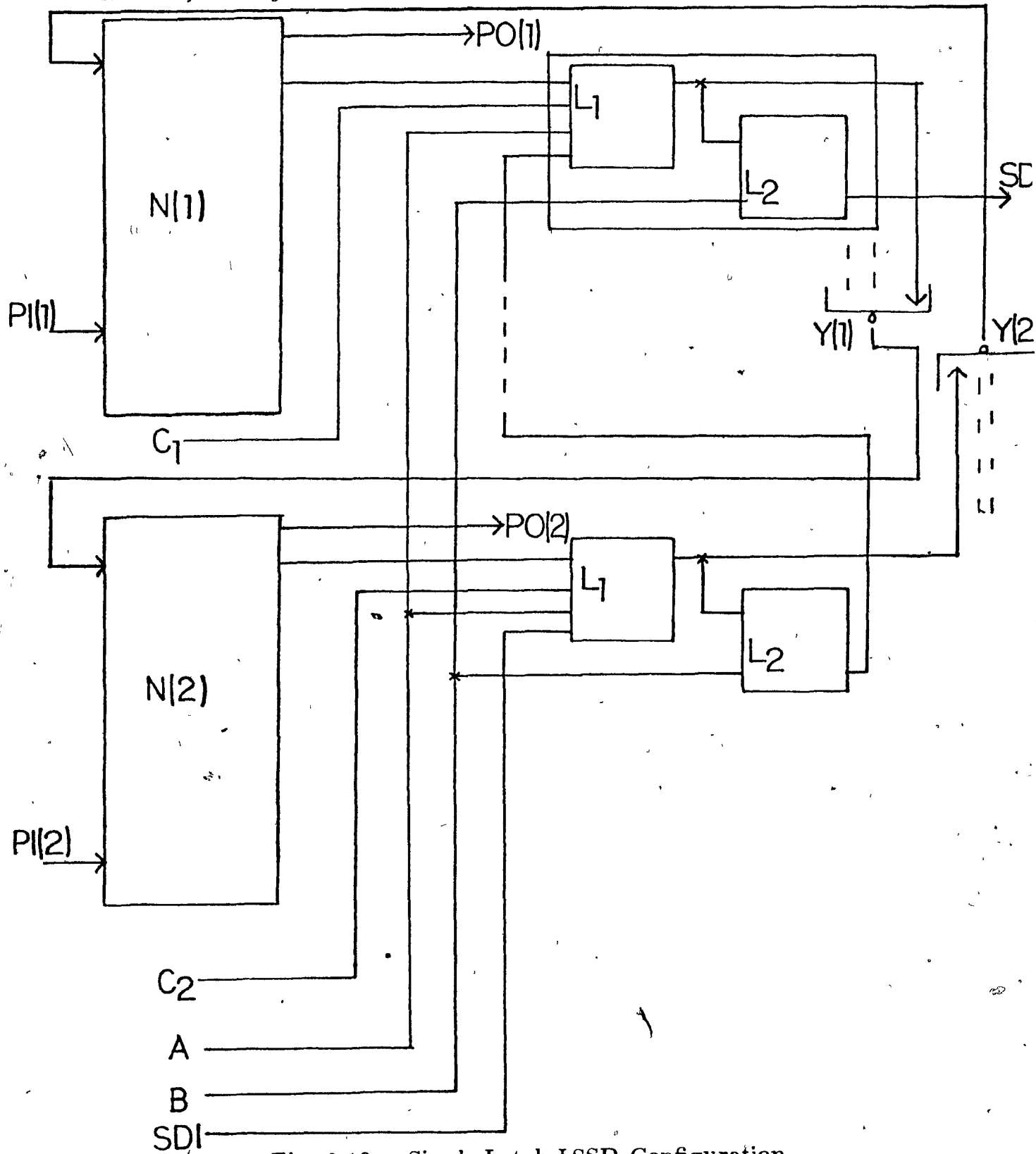


Fig. 2.13 Single-Latch LSSD Configuration

Review of Existing Schemes

$$C(max) < N\text{-delay}(max)$$

A disadvantage of the Single-Latch configuration based on the SRL of Fig. 2.13 is that the L2 latch has no role to play in system operation. In that sense, the L2 latches are redundant and represent a high overhead for testability. A variation to LSSD that solves the above problem is presented by Dasgupta etc which is called L2* Scheme [17]. This section describes the L2* Scheme and the other variations of Single-Latch LSSD approach.

2.4.1 Saluja's Scheme

A variation of the SRL was reported by Saluja in 1982 [28]. Fig. 2.14 shows this latch called a polarity-hold Parallel and Shift-Register Latch (PSRL). The PSRL has two modes of operation as shown in Fig. 2.15 and are as follows:

1. Mode 1 (Normal mode of operation): Under this mode the two latches L1 and L2 work in parallel and accept excitation signal D when the system clock C is at logic 1. In this mode A and B are held at logic 0. This mode is shown symbolically in fig. 2.15(a).

2. Mode 2 (Test mode of operation): Under this mode the latches work as shift register with IN as input and Q2 output of L2 as output (fig. 2.15(b)).

It is interesting to note that in mode 1, the uncomplemented and the complemented outputs are obtained from two different latches and in mode 2, the PSRL latch behaves

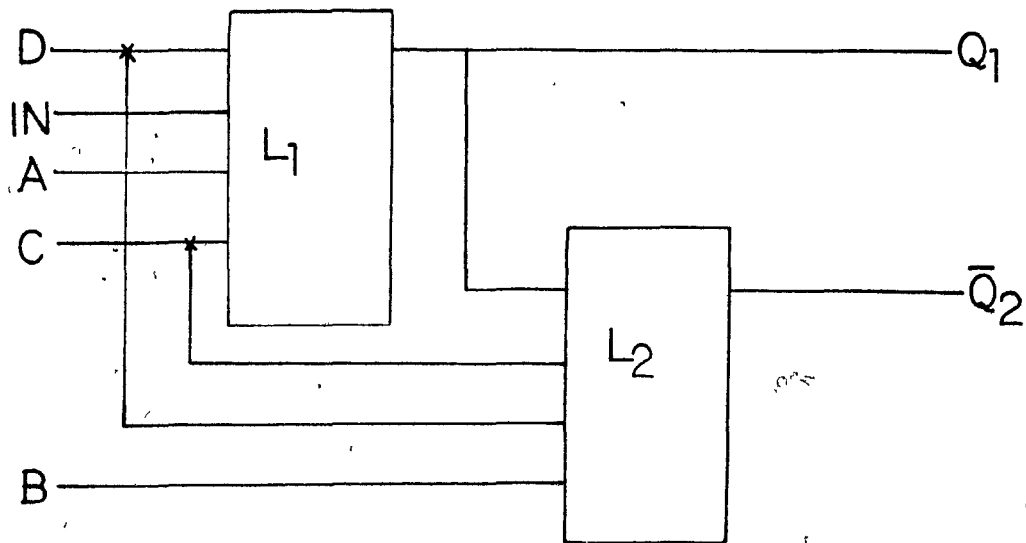


Fig. 2.14 Polarity-Hold PSRL

Review of Existing Schemes

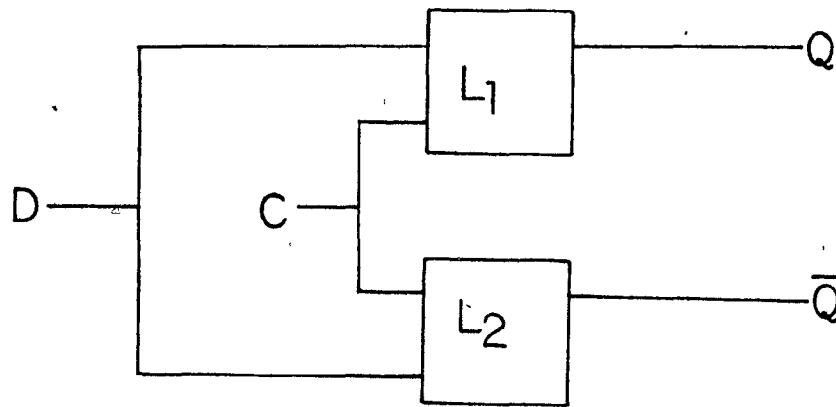


Fig. 2.15(a) Normal Mode

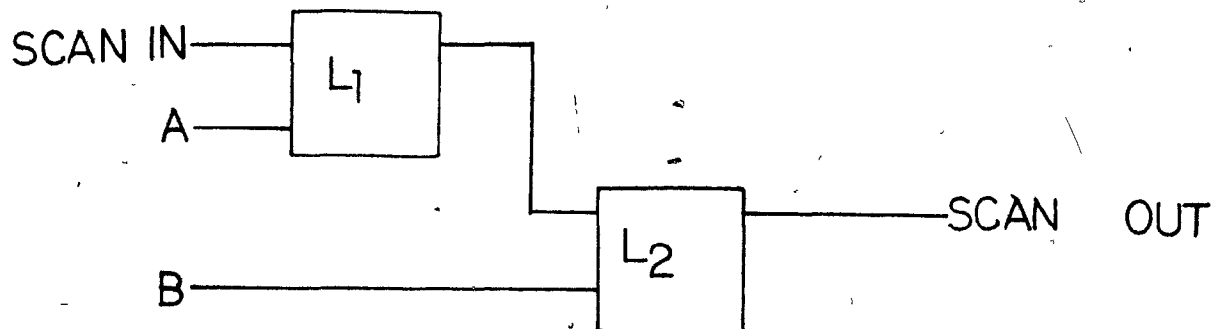


Fig. 2.15(b) Test Mode

Review of Existing Schemes

exactly the same as SRL. All the PSRLs are interconnected to form a shift register similar to LSSD approach.

2.4.2 L2* Scheme

Fig. 2.16 shows a variation of LSSD proposed by Dasgupta et al [29]. The difference between the L2 of fig. 2.7 and the L2* of fig. 2.16 is that the L2* latch has two independent data ports. The first port is fed by the related L1 latch and clocked by shift clock B. This allows the L2* latch to perform its traditional role as the slave latch in the shift register path. The second data port serves as an independent system data port clocked by system clock C* to permit different system data to be stored in the L2* latch during system operation.

From the designer's standpoint, the best feature of the L2* latch is that it requires no new design rules. However, one old rule needs greater attention now:

System outputs to a network can be taken from either the L1 or L2 latch of an SRL but not from both.

This rule is necessary to ensure that whatever test pattern is generated can actually be applied. If both latches of SRLs feed common logic, a situation could arise in which it might not be possible to shift in the required pattern. Fig. 2.17 shows one example. The L1 latches of two successive SRLs in the shift register path are required to have a value of 1, while the L2 latch in between must have the opposite value. This pattern cannot be shifted in

Testing of LSSD networks using the L2* latch, proceeds as follows:

Review of Existing Schemes

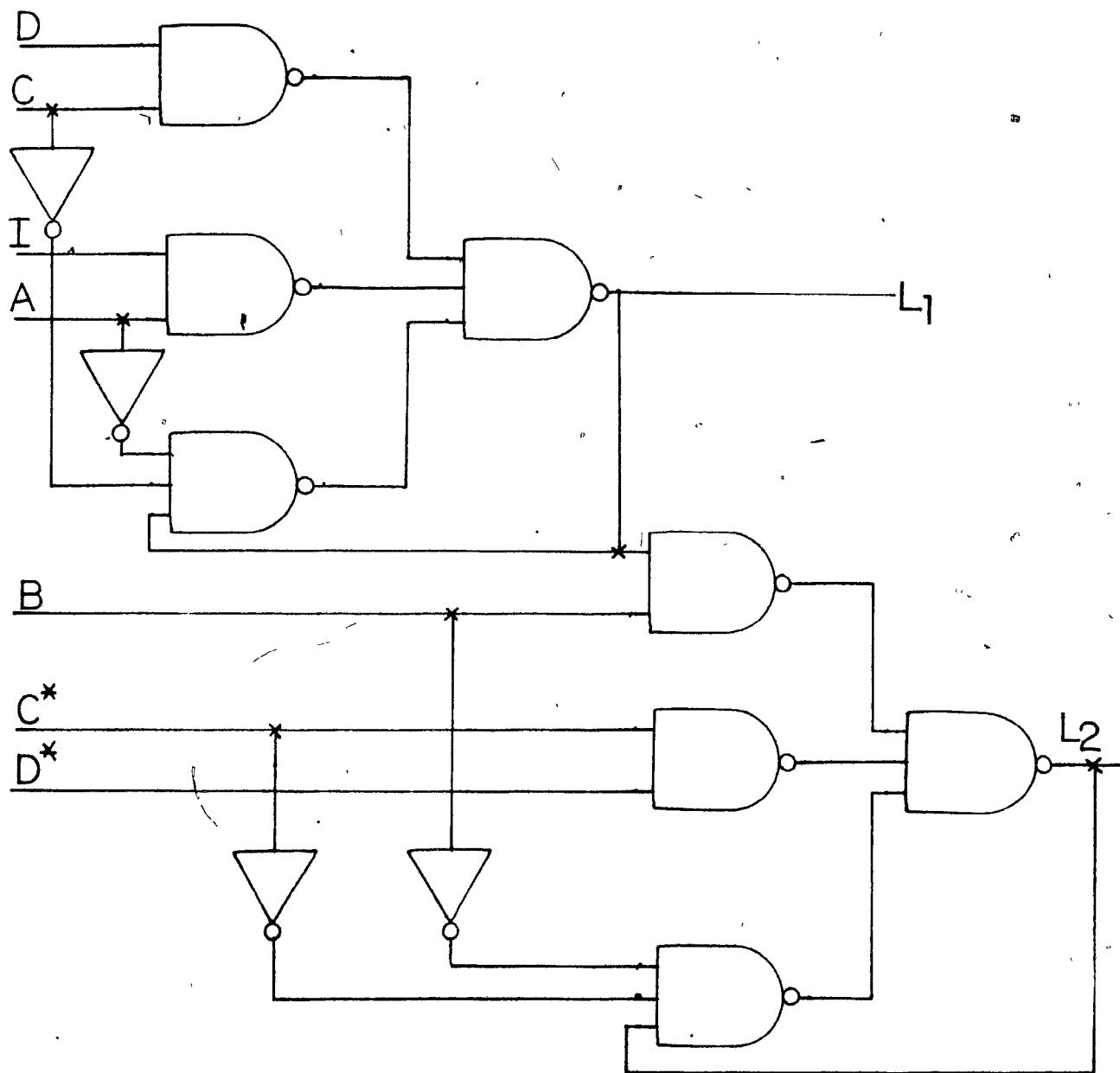


Fig. 2.16 SRL with L2* Latch

Review of Existing Schemes

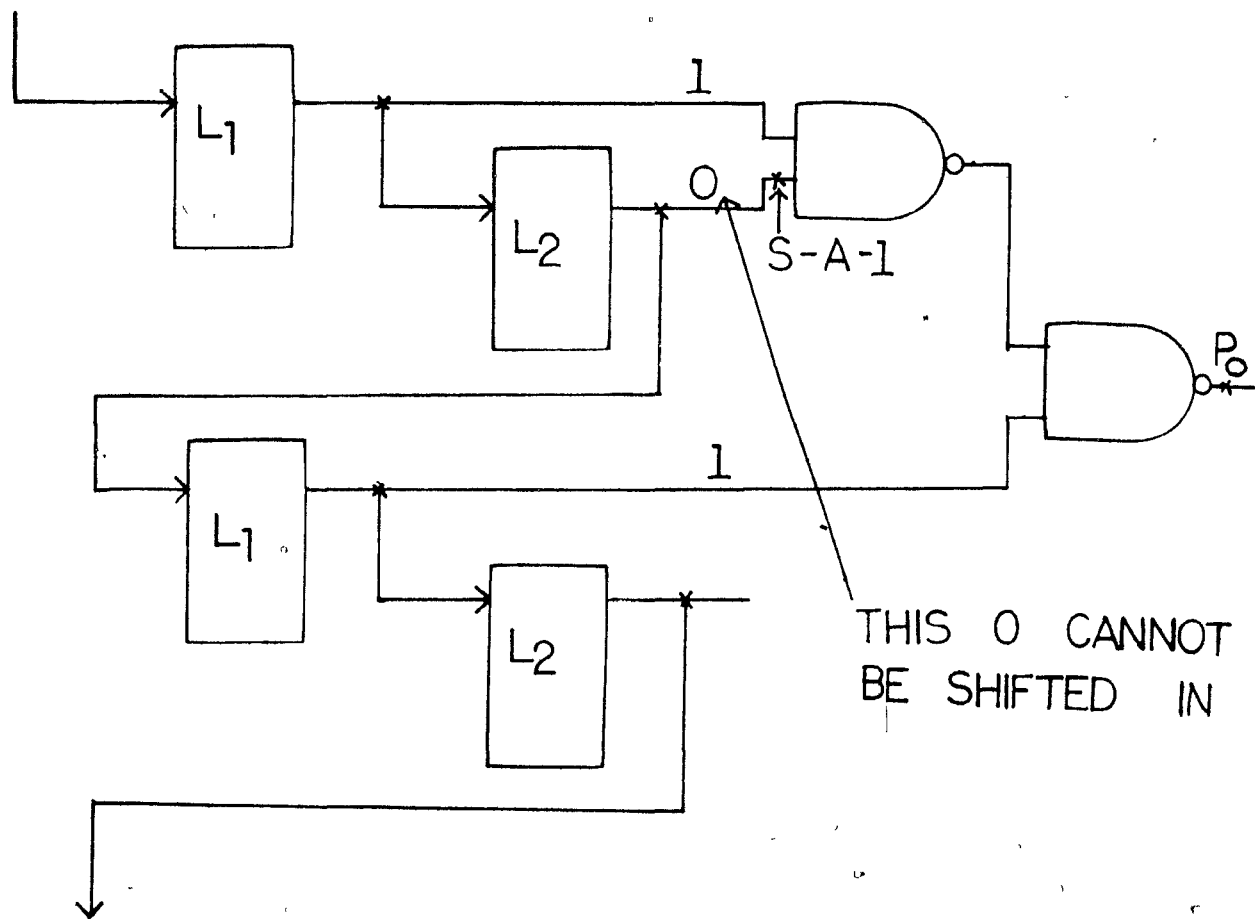


Fig. 2.17 Untestable Configuration

Review of Existing Schemes

Step 1: Set up the input state to the combinational logic by loading the shift registers and applying the desired values at primary inputs.

Step 2: Pulse the proper system clock to capture the result of the test in the L1 latches of SRLs at the outputs of the logic network.

Step 3: Pulse shift clock B to transfer the test values to the L2 latches of the SRLs.

Step 4: Unload the shift registers by repeating the operations 'Pulse shift clock A, pulse shift clock B' and measuring the 'scan-out' primary output.

Step 5: Repeat steps 2, and 4, this time pulsing the system clock feeding the L2⁺ latch.

We will now discuss the important question of how much the LSSD and its variations cost in logic gates and operating speed with level-sensitive design. The polarity-hold latches in the shift registers are logically two to three times as complex as simple latches. The logic gate overhead for implementing the level-sensitive design ranges between 4% to 20%. Four additional Input/Output pins are required for controlling the shift operation. This is a serious problem, since routing of three additional signals may add significantly to the area of the chip. External asynchronous input signals must not change more than once every clock cycle. This constraint is required so that level-sensitive logic subsystems will result. All timing within the subsystem is controlled by externally generated clock signals. The overall performance of the subsystem will be degraded by the clocking requirement. The increased clock delay is due to the connection of the output of flip-flop to the scan input of the next flip-flop in the scan register chain. This results in extra capacitive loading.

The main drawback of the Scan based designs, in particular LSSD is large test-application time. The testing strategy for these Scan designs requires the circuit to cycle from 'Scan' mode to 'Normal' mode and back again as each test is loaded and

Review of Existing Schemes

applied. The serial nature of the 'Scan-In, Scan-Out' mechanism can create long test-application times. For LSSD, Goel^[9] has projected that the total test-application time is proportional to G^2 where G is the gate count in excess of 100,000, the total test-application time becomes extremely large.

In the $L2^*$ scheme, the $L2$ latch is fully used even in the normal operation. This utilization of the $L2$ latch substantially decreases the area overhead attributed to SRL implementations. The total number of non-overlapping clocks which are required for the proper functioning of the system using $L2^*$ are same as in systems using SRLs. The overall system performance is not affected. Thus systems using $L2^*$ or SRLs can run at the same speed.

The design of latches in Saluja's scheme result in a reduction of effort in test pattern generation and provide a better fault coverage. As mentioned earlier, all Y inputs to the combinational logic are obtained from $Q1$ outputs of PSRLs and $\bar{Q}2$ outputs of PSRLs. For test generation purpose, $Q1$ and $\bar{Q}2$ are considered as independent variables. This process will increase the number of controllable inputs to the combinational logic. One of the time consuming operation in D-algorithm^[15] is consistency operation. By considering uncomplemented and complemented variables as independent variables, less inconsistencies result. Also, independence of Y from $\bar{Y}$ helps sensitize many paths. There is a little difference between the total number of gates used in a PSRL^[28] and a SRL. Two additional NAND gates are used in PSRL. However, while using SRLs, some of the $L2$ latches can be used for other system latches whereas in design's using PSRL, this cannot be possible. As the two sublatches in PSRLs work in parallel the overall system performance is not effected. Thus systems using SRLs or PSRLs can run at the same speed.

Review of Existing Schemes

To summarize, the main advantage of the LSSD techniques, because of scan capability, reduces the sequential test generation problem to a combinational one and enables logical partitioning of the circuit. Another advantage of the LSSD is that ac testing as well as test generation and fault simulation are greatly simplified, since the correct operation of the logic circuit is nearly independent of the ac characteristics and also the polarity-hold latch is free of hazards and race conditions.

Proposed Scheme

3. Proposed Scheme

As discussed before, the level-sensitive design introduced by Eichelberger and Williams [13] has the drawback that the basic memory element, the shift register latch (SRL) must have two latches L1 and L2, which are connected in master-slave configuration. For most of the designs, the master latch L1 is sufficient to achieve the required system function. The functionally idle L2 latch is useful only for shifting and therefore is an overhead for testability in Single-Latch designs. A variation to LSSD that solves the above problem is presented by Dasgupta et al [29]. This design has the disadvantage that four clock signals have to be routed over the whole chip. Routing of more than one clock can introduce major layout and timing problems since routing of several clocks can introduce time skews between the clock signals. Hence, from a layout point of view this scheme seems to be having considerable area overhead and also three additional input pins are required.

Another important point that should be noted is that the basic latch structure presented by Dasgupta et al [29] is not completely hazard-free. To illustrate this point, consider the basic latch structure of the SRL using L2* as shown in fig. 3.1. Referring to fig. 3.1, if we assume that the state of node 'p' changes that of node 'r' in response to a change of clock C from 1 to 0, then steady-state hazard exists as shown in fig. 3.2. The existence of steady-state hazard can be defined [30] as: If the circuit fails only in that, immediately after certain input changes, the system enters the wrong stable state, then steady-state hazard results. In fig. 3.2, it is assumed that the zero value of 'Y' is fed back rapidly enough to hold the state node 'r' at value 1.

In fig. 3.3, it is assumed that the state of node 'r' responds more quickly to a change of input C from 0 to 1 than does node 'p'. If we consider only gate delays.

Proposed Scheme

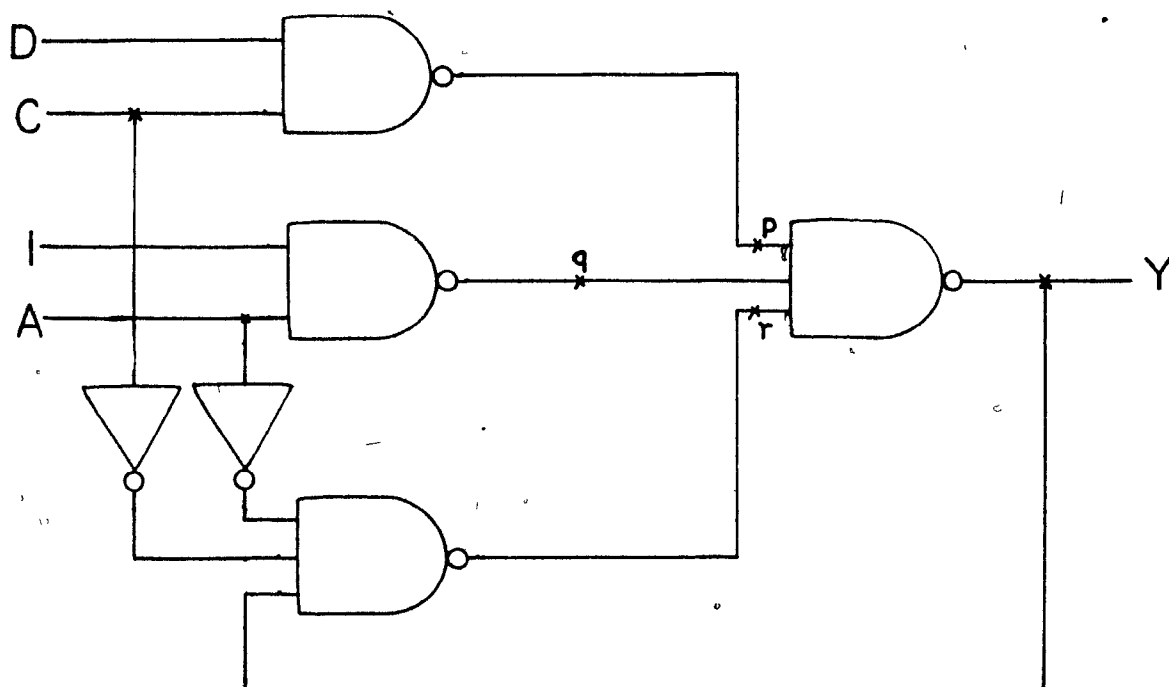


Fig. 3.1 Latch Structure in L2* Scheme

Proposed Scheme

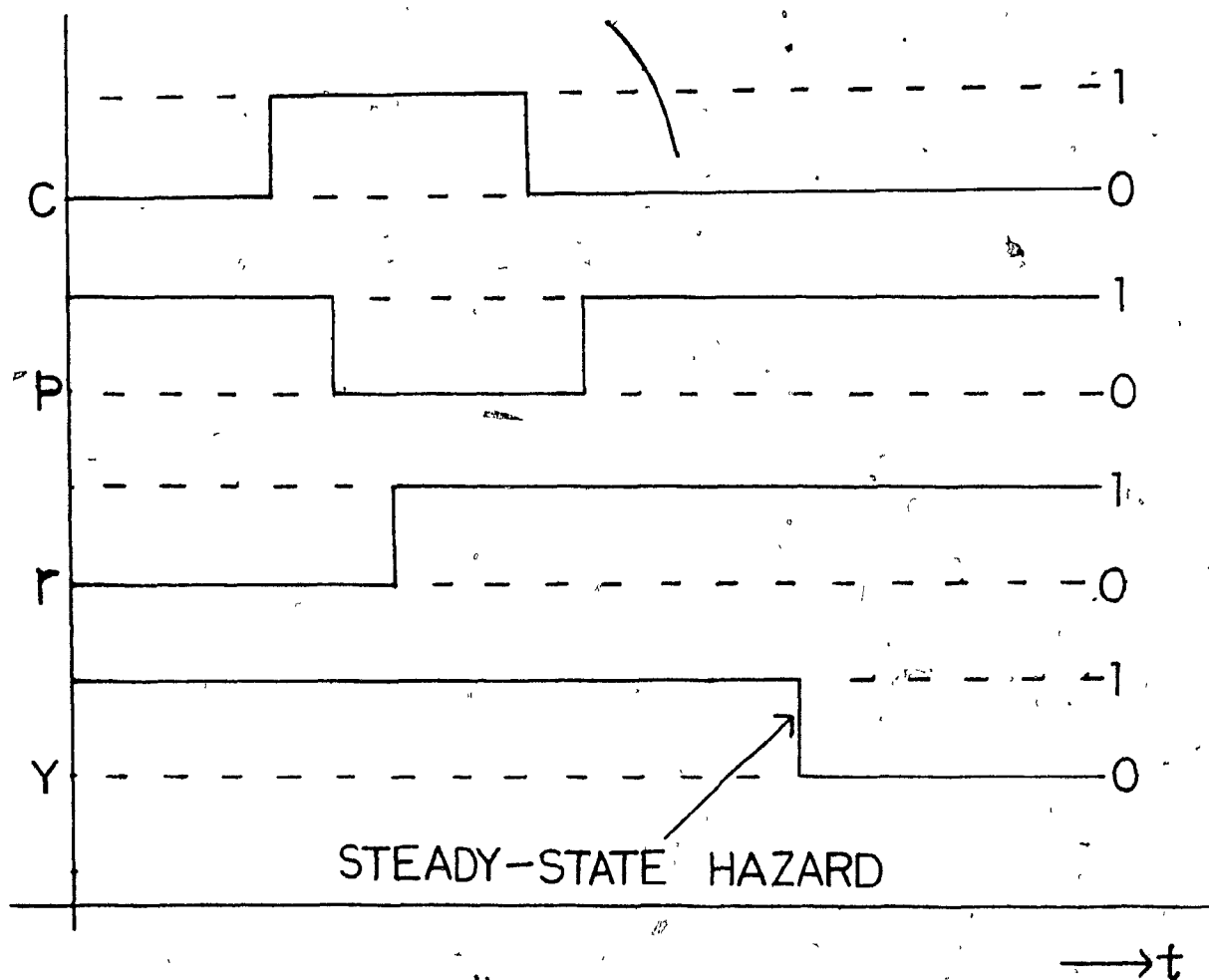


Fig. 3.2 Steady-State Hazard Condition

Proposed Scheme

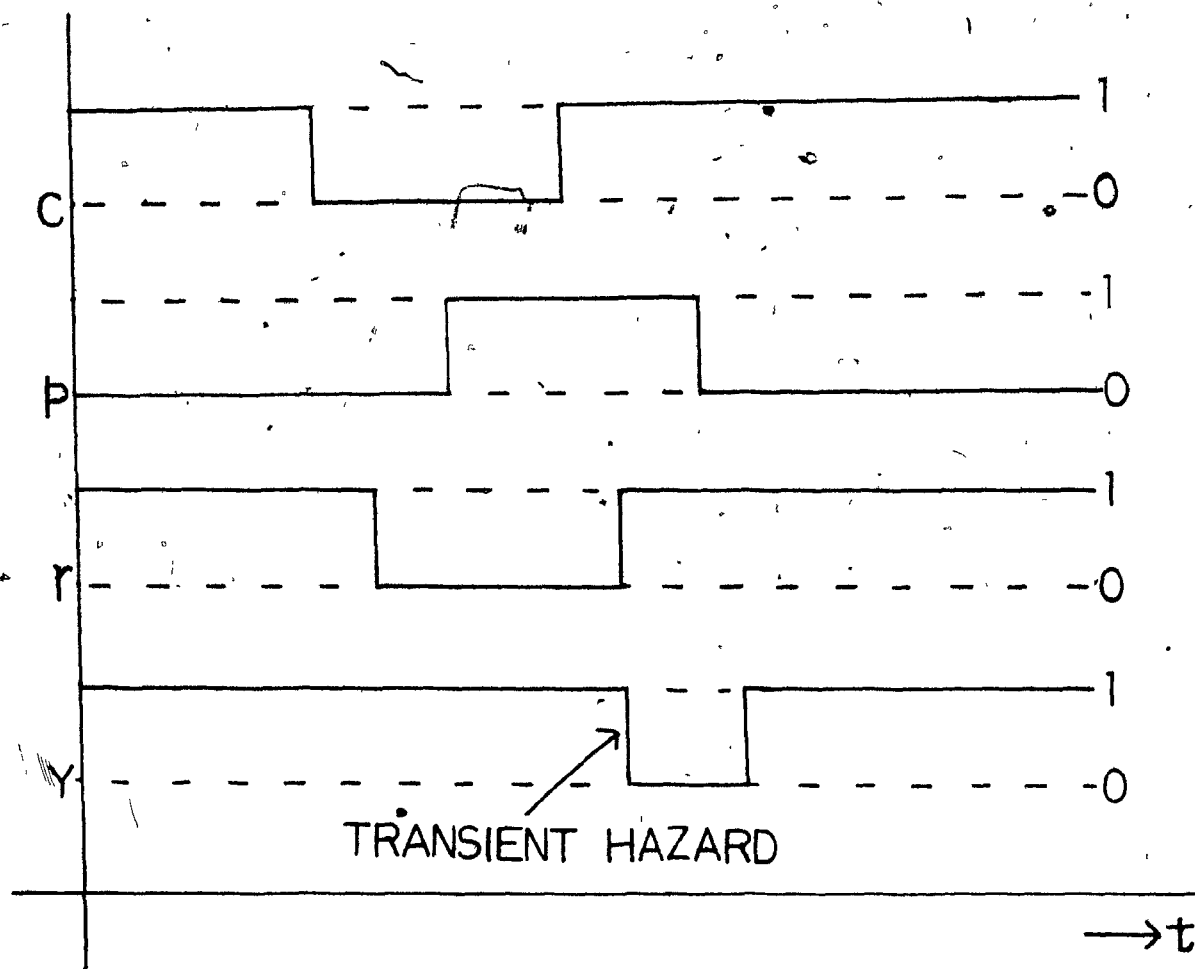


Fig. 3.3 Transient Hazard Condition

Proposed Scheme

then this possibility appears unlikely since two gates are involved for node 'r' and only one for node 'p'. A transient hazard results. The existence of transient hazard can be defined[30] as: If the circuit fails only in that, immediately after certain input changes, pairs of false output changes occasionally occur on some output leads, then transient hazards are said to exist.

It is not possible, however, to have both types of hazards in a particular circuit since a given distribution of time delay will result in one form of hazard but not both as an inspection of fig. 3.2 and fig. 3.3 shows. If sufficient timely delay is inserted in the feedback loop, then the steady-state hazard can be eliminated since the value of node 'r' will become equal to 0 before the output can be fed back to maintain it at the value 1. Another way of eliminating the static hazards is by using redundant gates. Then care should be taken to determine that the transitions for which redundant terms would be added can actually occur.

A hazard-free polarity²-hold SRL using $L2^*$ can be designed with a structure similar to that of the original SRL proposed by Eichelberger and Williams and is shown in fig. 3.4. It consists of two latches, $L1$ and $L2^*$. As long as the shift signals A and B are both 0, the $L1$ and $L2^*$ latches operate exactly like a polarity-hold latch. Terminal I is the input to the shift register, and L2 is the output. When the latch is operating as a shift register, data from the preceding stage are gated into the polarity-hold latch $L1$ via I, by a change of the A shift signal to 1. After A has changed back to 0, the B shift signal gates the data in the latch $L1$ into the output latch connected to the output terminal L2. Clearly, A and B can never be 1 at the same time if the shift registers are to operate properly. When the latch is operating in the normal mode, data are gated into the polarity-hold latch $L1$ via D, by a change of the C clock signal to 1. After C

Proposed Scheme

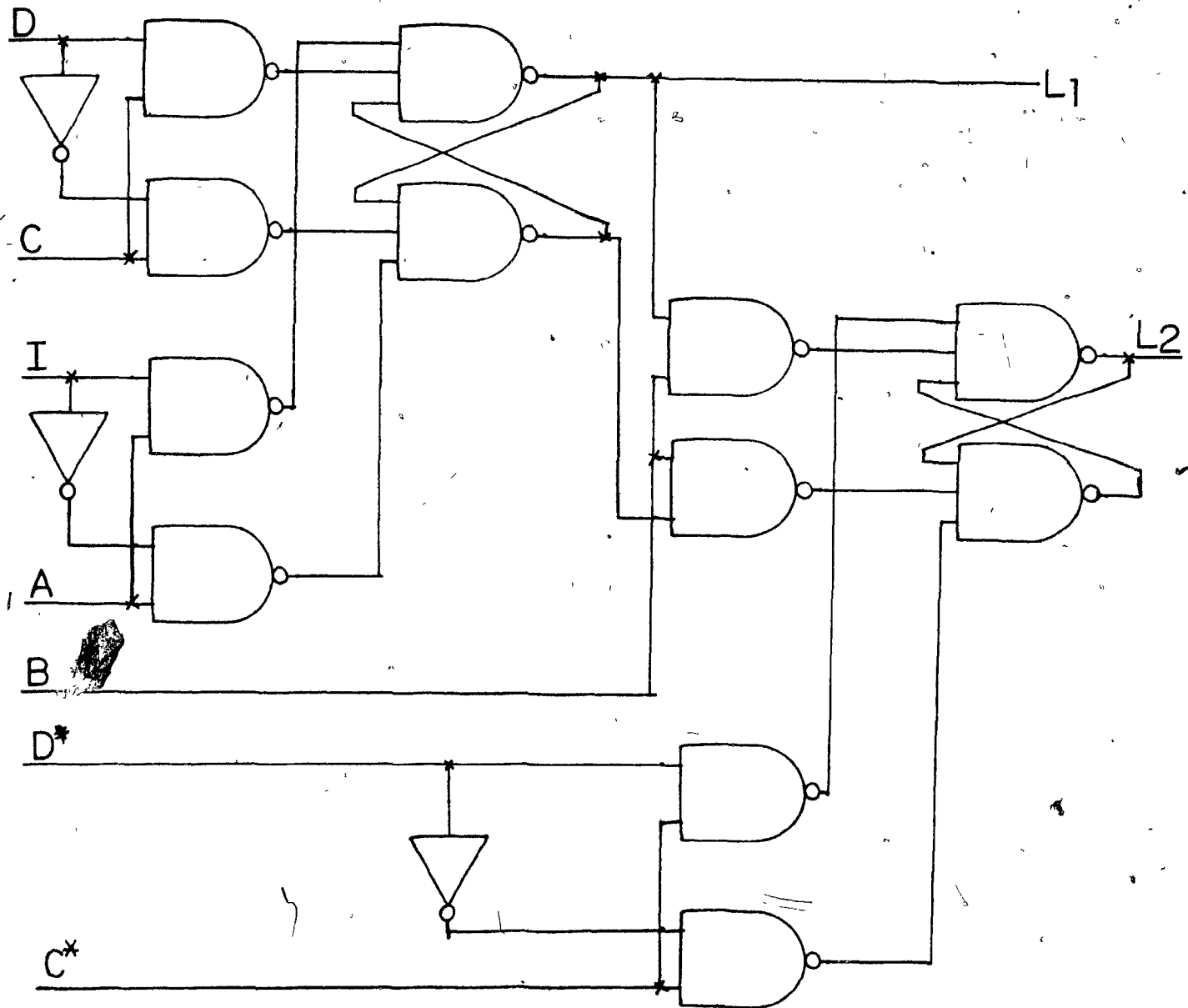


Fig. 3.4 Hazard-Free SRL using L2*

Proposed Scheme

has changed back to 0, the C^* clock signal gates the system data into the $L2^*$ latch via D^* .

As discussed above, the $L2^*$ design has the disadvantage that four clock signals have to be routed over the whole chip. This is important in chip layouts, since routing of several clocks can introduce time skews between the clock signals. In an attempt to solve the routing and timing problems, a modification of the $L2^*$ latch will be discussed in this section. This modification is shown in fig. 3.5((a) and (b)). We shall call this MSRL (Modified Shift Register Latch). A 'Mode-Switch' input selects the normal or scan shift mode (Mode switch = 1 for normal mode). The same system clocks C and C^* are used in both scan and normal modes. In the normal operation, inputs C and C^* are used as system clocks, while the input MS is held 'high'. For the scan operation, the two clocks C and C^* are used as scan clocks, while the mode-switch MS is held 'low' to detach the normal data lines D and D^* . Hence, in this scheme two clock signals and a mode-switch have to be routed over the whole chip. Whereas in the $L2^*$ scheme, two system clocks C and C^* , and two scan clocks A and B need to be routed.

An important characteristic of this latch (MSRL) is that no race or hazard conditions are present during normal operation. In other words, the latch can be used as a level-sensitive latch. As shown in fig. 3.5((a) and (b)), the MSRL consists of two latches $L1$ and $L2$. Latch $L1$, using clock C , gates system data D . Similarly, $L2$ operates independently of $L1$, using D^* and C^* . When the SRL is operating as a shift register, the mode-switch (MS) is reset to zero, input from SI is gated into latch $L1$ when the shift signal C changes from 0 to 1. When latch $L1$ is stable and C is changing back to 0, the C^* shift signal gates the $L1$ data into $L2$ by changing from 0 to 1. The two-phase shifting operation is a characteristic of LSSD. It is important to properly control the routing delays of C and C^* signals to preserve their nonoverlapping nature. An overlap of these

Proposed Scheme

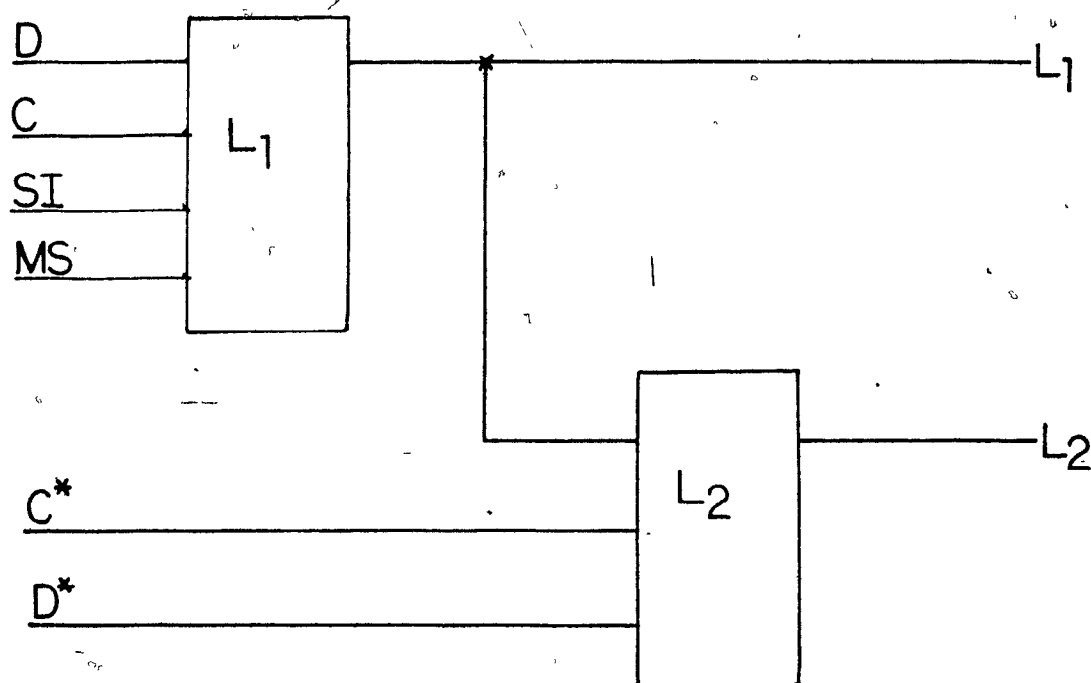


Fig. 3.5(a) Proposed Scheme (MSRL)

Proposed Scheme

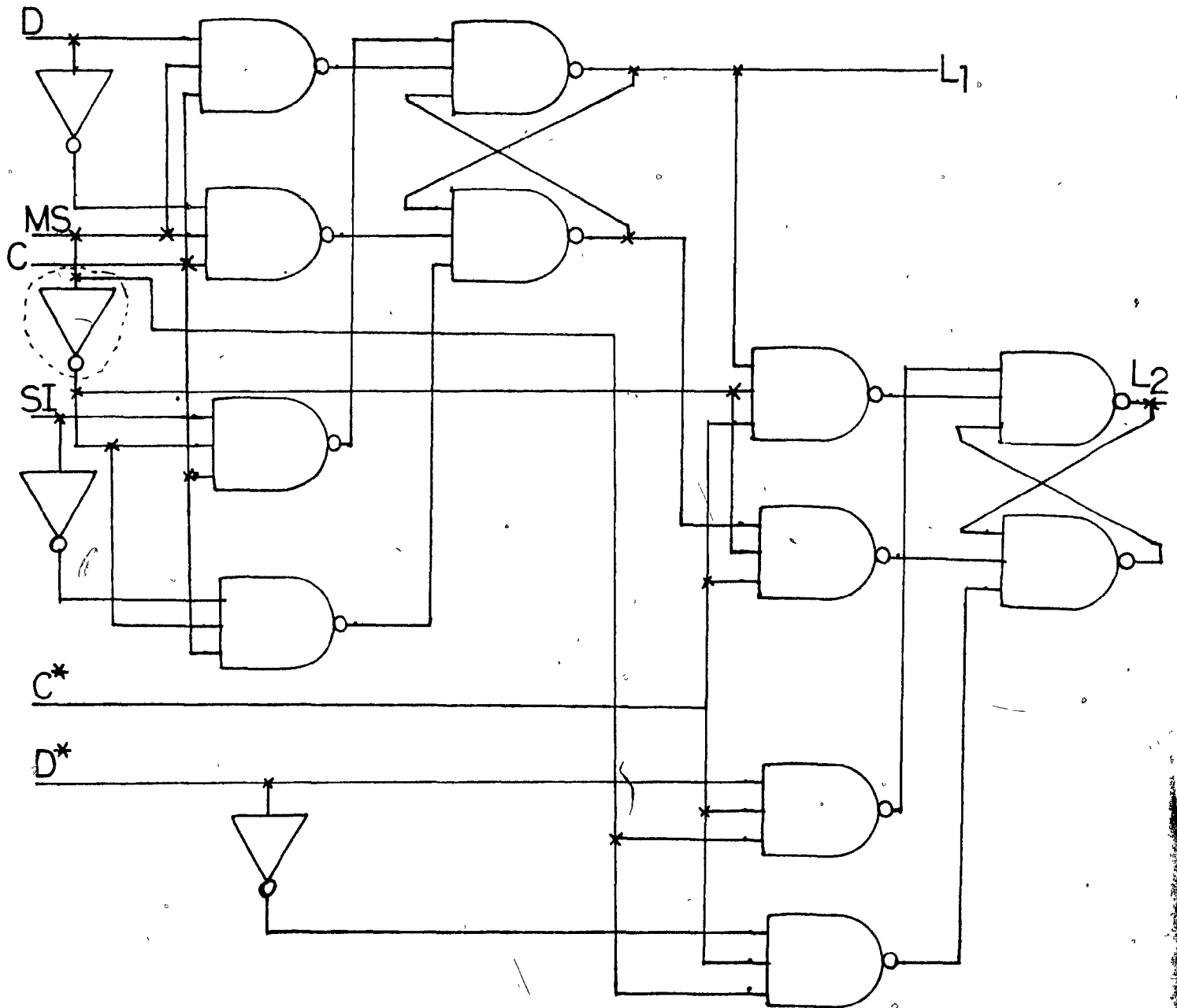


Fig. 3.5(b) Proposed Scheme (MSRL)

Proposed Scheme

signals (i.e., C and C^* being high simultaneously) can cause incorrect operation. The truth table (shown in Fig. 3.6(a)) and the suggested waveforms for MS , C and C^* are illustrated in fig. 3.6(b).

Thus, designing with MSRL would result in saving of one input pin and the routing which would otherwise have to connect this pin to all flip-flop, thus decreasing the area. It is well recognized that in VLSI circuits, long routing paths are more expensive in terms of chip area than a few devices which are locally connected.

A logic subsystem using MSRL will have the structure shown in fig 3.7. As shown in the figure, the two clock signals partition the logic subsystem into two parts, each composed of a combinational network and a set of MSRLs. Each of the combinational networks, $N1$ and $N2$, is a multiple-output logic network. $P1$ and $P2$ are primary inputs to the network, and $Z1$ and $Z2$ are for primary outputs. C and C^* are the two system clock signals. The operation of the subsystem is controlled by the clock signals. At C time, C^* is zero and the inputs and outputs of $N1$ are stable (assuming that the external inputs $P1$ are also stable). The clock signal C is then allowed to pass to the MSRL system clock input. The system clock C may be gated by signals from network $N1$ such that C reaches the MSRL if and only if the gate is active. Thus some of the latches may change at C time. These signal changes immediately propagate through network $N2$. As soon as C is changed back to 0 and all $L1$ signals have finished propagating, the next clock signal, C^* may occur. For correct operation of the subsystem, all that is needed is for the clock signals to be long enough to allow all latch changes to finish propagating. This structure meets the requirements for level-sensitive operation and ensures that there is little or no dependence on ac circuit parameters. For proper operation of the logic subsystem, all that is needed is that the delay through the combinational networks $N1$ and $N2$ be less than the corresponding time between the clock signals.

Proposed Scheme

MS	C	C*								
1	0	0	NONE	ACTIVE						] NORMAL MODE
1	1	0	LOAD	L ₁ WITH	SYS.	DATA	D			
1	0	1	LOAD	L ₂ WITH	SYS.	DATA	D*			
1	1	1	NOT	ALLOWED						
0	0	0	NONE	ACTIVE						] SCAN MODE
0	1	0	LOAD	L ₁ WITH	SCAN	DATA	SI			
0	0	1	LOAD	L ₂ WITH	SCAN	DATA	FROM	L ₁		
0	1	1	NOT	ALLOWED						

Fig. 3.6(a) Truth Table for MSRL

Proposed Scheme

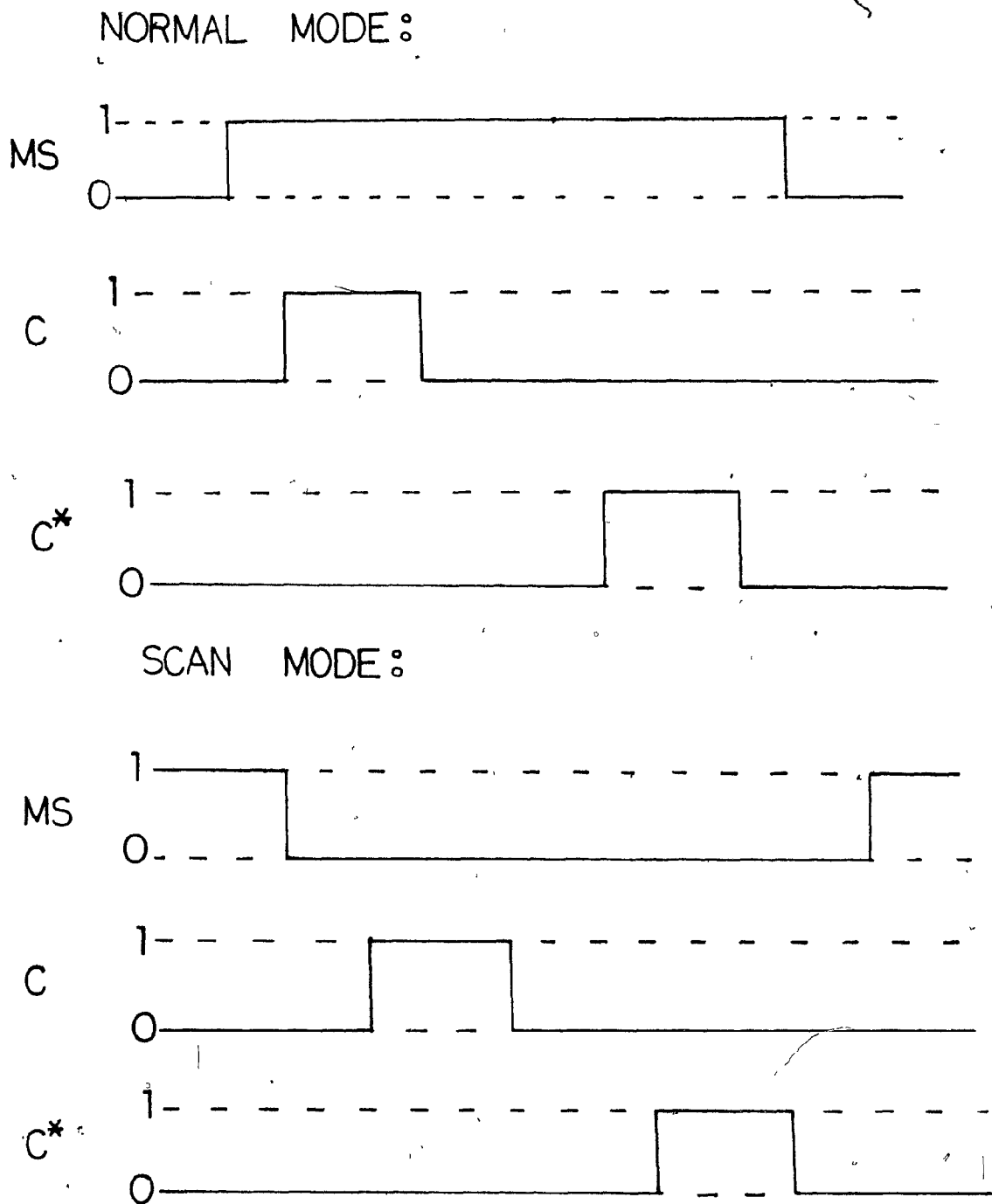


Fig. 3.6(b) Waveforms for MSRL

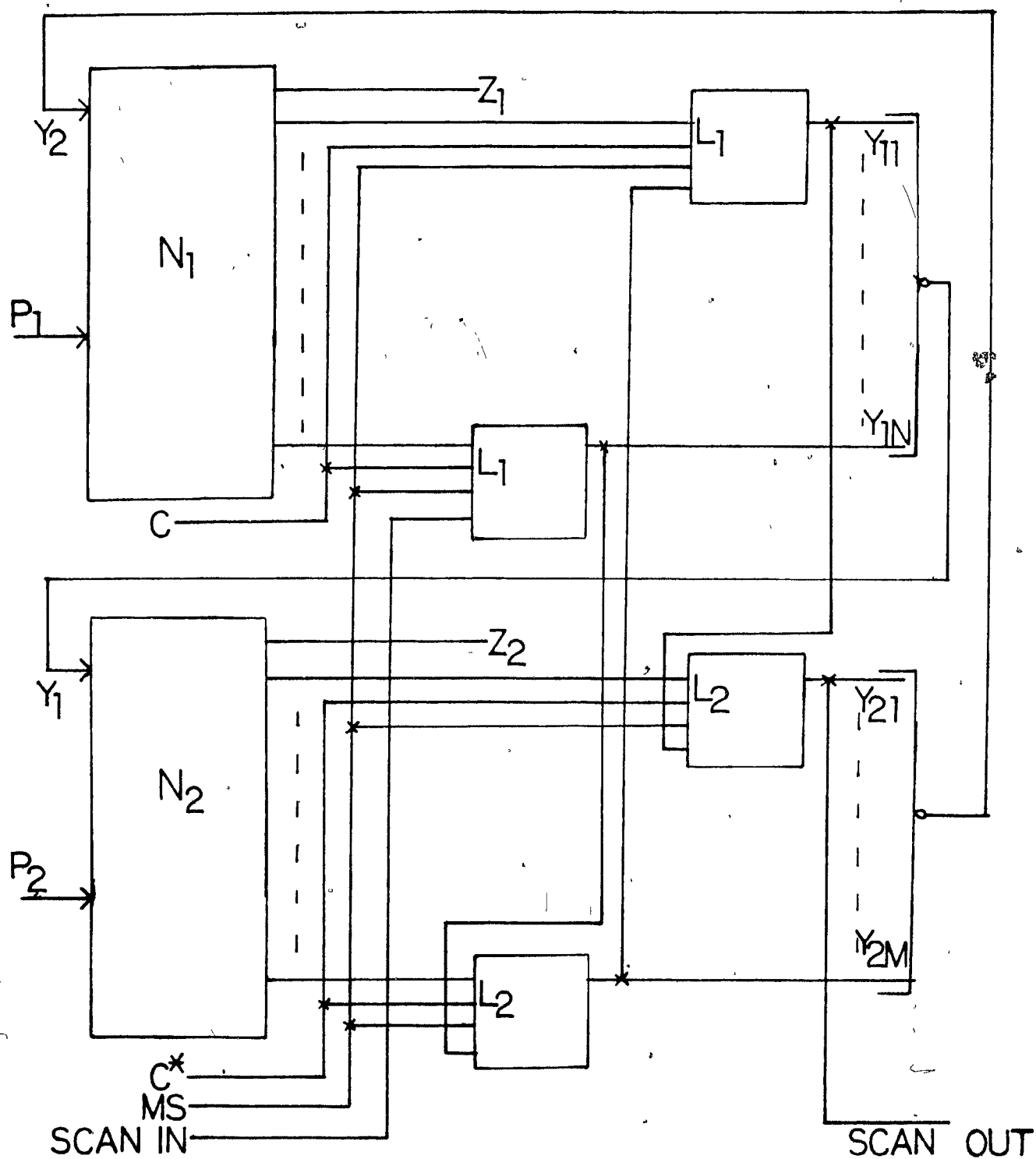


Fig: 3.7 Single-Latch Structure using MSRL

Proposed Scheme

3.1 Testing The MSRL

In general, in sequential circuits, the future state of the stored-state devices depends on both the primary inputs and the current recorded state of the stored-state devices themselves. It is this dependency of the future state on the present state that causes all the problems in test generation. The primary inputs are the only inputs over which the test programmer has direct control. Similarly, the primary outputs are the only outputs that can be observed directly. Control and observation of the stored-state devices is indirect through the combinational section of the circuit. The problem is - which section do we test first given that neither section is directly controllable or observable and that the sections are mutually dependent on each other for correct operation. The scan-design technique provides a solution to this problem by reducing the complexity of the circuit structure. The testing strategy for the MSRL and all the scan methods described in chapter 2 is now as follows.

STEP 1: Select the scan mode, i.e., all latches are reconfigured into a shift register. Test the status and operation of each latch using the Scan In, L2 output and system clock facilities. A suitable test for the shift register is as follows:

- (a) Shift test. In this test, the sequence 00110011.... is shifted through the register. This sequence exercises each latch through all combinations of present state and future state.

STEP 2: Determine a set of tests for the two combinational logic blocks, assuming

- (a) total control of all inputs (primary and from the latches);
- (b) direct observability of all outputs (primary and to the latches).

STEP 3: Apply each test in the following way:

Proposed Scheme

- (a) Select scan mode. Preload the latches with test input values and establish additional test input values on the primary inputs.
- (b) Select normal mode. The steady-state output response of one combinational logic block can now be clocked into the corresponding latch (L1 or L2).
- (c) Return to scan mode and clock out the contents of the latches. Compare these values, plus the values directly observable on the primary outputs, with the expected fault-free response.

STEP 4: Repeat step 3 for the other combinational logic block.

The 'divide-and-conquer' philosophy of the scan design approach can now be seen more clearly. Rather than test the circuit as a single entity, the addition of the shift path allows each major segment to be tested separately and in a procedural manner. Furthermore, if we assume a standard test for the latches (Step 1 above), the only test generation problem is to generate tests for the combinational segment. This problem has been well researched and a variety of programmable procedures exist [8].

3.2 Design Rules

A specific set of design rules will be described below, that will result in a design suitable for scan implementation with MSRL. The rules are simple to follow and can be checked automatically by a CAD tool. These rules result in a hazard-free and race-free sequential design and still provide considerable flexibility to the designer. These rules are designed to preserve the level-sensitive property and the scan property.

Rule 1: All internal memory elements must be implemented in MSRL type flip-flop.

Proposed Scheme

Rule 2: MSRLs are controlled by two non-overlapping clocks such that :

- (a) the L1 or L2 output of MSRL(1) can be used to gate a clock C to produce a gated clock, C(G). C(G) can then be used to clock another latch, MSRL(2), provided MSRL(1) is not being clocked by C;
- (b) subject to this restriction, the outputs of MSRL(1) may feed the data inputs of MSRL(2).

Rule 3: It must be possible to identify a set of MSRL that are directly controllable.

This means that:

- (a) all clock inputs can be held inactive independently;
- (b) any single clock can be made active while the others are maintained in their inactive state.

Rule 4: Clock primary inputs can only be connected to MSRL clock inputs. They cannot be connected to MSRL data inputs, either directly or through the combinational logic circuit.

Rule 5: System outputs to a network can be taken from either the L1 or L2 latch of an MSRL network but not from both.

This rule needs greater attention for a single-latch design. Since in a single-latch design both the latches are used for system function, extra care should be taken to ensure that both L1 and L2 outputs of the MSRL do not feed common logic. This also ensures that no hazards or races occur in the circuit. Rules 1-5 constitute a check for the property of level sensitivity. Rules 6-8 are for the scan mode verification.

Proposed Scheme

Rule 6: All MSRLs are permanently connected to form a shift register with a scan-in primary input, scan-out primary output and accessible control clocks.

Rule 7: There must exist a circuit configuration state, directly controllable from the primary inputs called the 'scan' state. One primary input pin must be allocated for specifying the mode (scan or normal mode).

Rule 8: When the mode specification line is in scan mode then the output of a flip-flop or scan-out primary output should be a function of only the preceding flip-flop output or scan-in primary input of the shift register.

3.3 Area And Performance Overhead

It is evident that all the advantages which apply to LSSD are also applicable to designs using MSRL. System performance is not dependent on hard-to-control ac circuit parameters such as rise time, fall time, or minimum delay. Test generation and testing are simplified to the well understood method of combinational logic network testing.

As pointed out earlier, the speed of the system is not degraded since the additional gates added to the latch are not in the data path. The area overhead for the scan design depends very much on the circuit structure. Other factors are the proportion of flip-flops of the whole circuit and their distribution over the chip. As suggested earlier, the proposed scheme reduces the required routing area in comparison to the L2* scheme, at the expense of some additional logic (1 inverter) in each flip-flop. In the polycell layout of the chip, the area is divided between the cell rows and the routing channels as shown in fig. 3.8. An estimate of area overhead is given below since a chip design using MSRL has not been completed.

Proposed Scheme

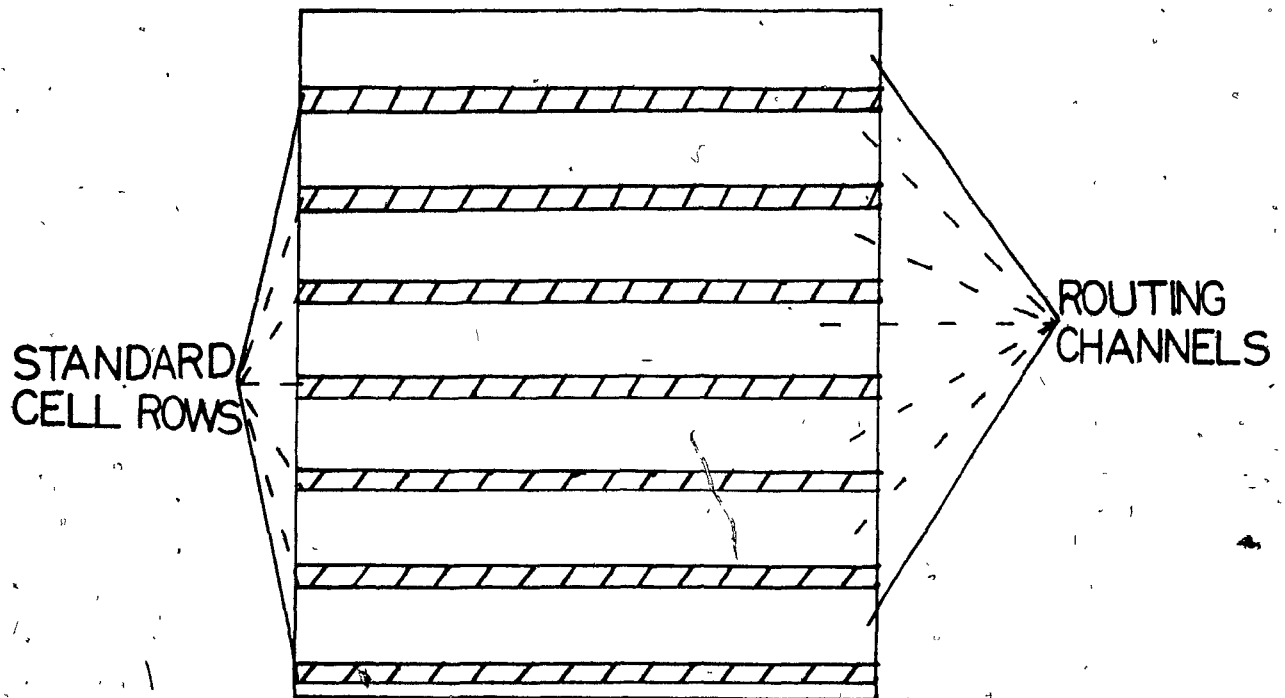


Fig. 3.8 Polycell Design Layout

Proposed Scheme

Theoretical Calculation of Area Overhead:

The polycell layout style consists of standard cells placed on grids in the rows of the layout as shown in fig. 3.9. The polycells contain simple boolean or memory cells. One dimension (height) of the cells is fixed to allow for an arrangement in rows. The width of the polycells varies. The rows of polycell are separated by routing space. The routing space consists of routing channels. Routing is mainly done in channels between the adjacent rows of cells.

The implementation of scan design increases the area of the chip in two ways [31]. First, the width of the scan flip-flop is larger than that of ordinary flip-flop (height of both flip-flops remain the same). The larger flip-flop size is reflected by the increase in width of polycell row. Secondly, the scan design requires at least two additional routing channels per pair of polycell rows. One of these channels is for mode specification line and the other channel is for scan data line. This is shown in Fig. 3.10.

The increase in area, due to larger scan flip-flops, is dependent on the fraction of chip area that is occupied by the flip-flops. The total increase in area can be theoretically calculated as follows:

Let

'C' be the number of combinational cells per row

'S' be the number of sequential cells per row

'N' be the number of polycell rows

' n_1 ' be the number of control lines per polycell row without scan

' n_2 ' be the number of control lines per polycell row with scan

'h' be the height of the pre-scan cell

Proposed Scheme

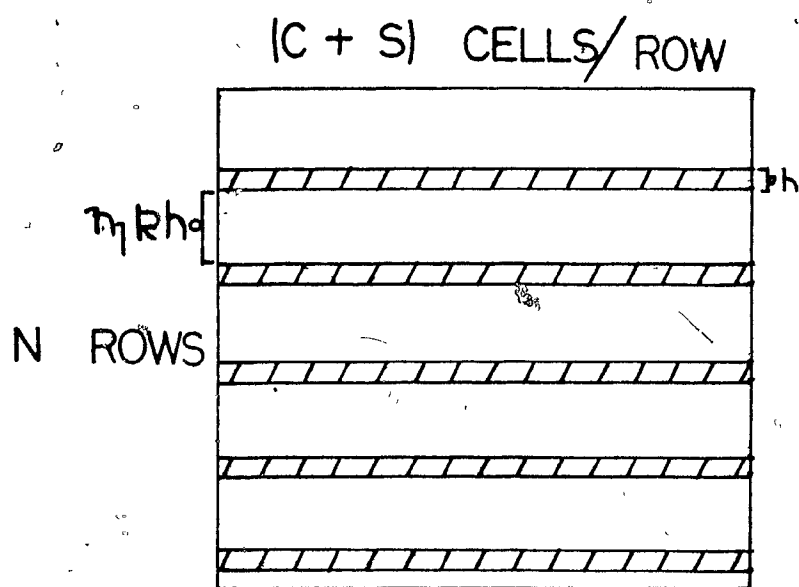


Fig. 3.9 Polycell Design Layout

Proposed Scheme

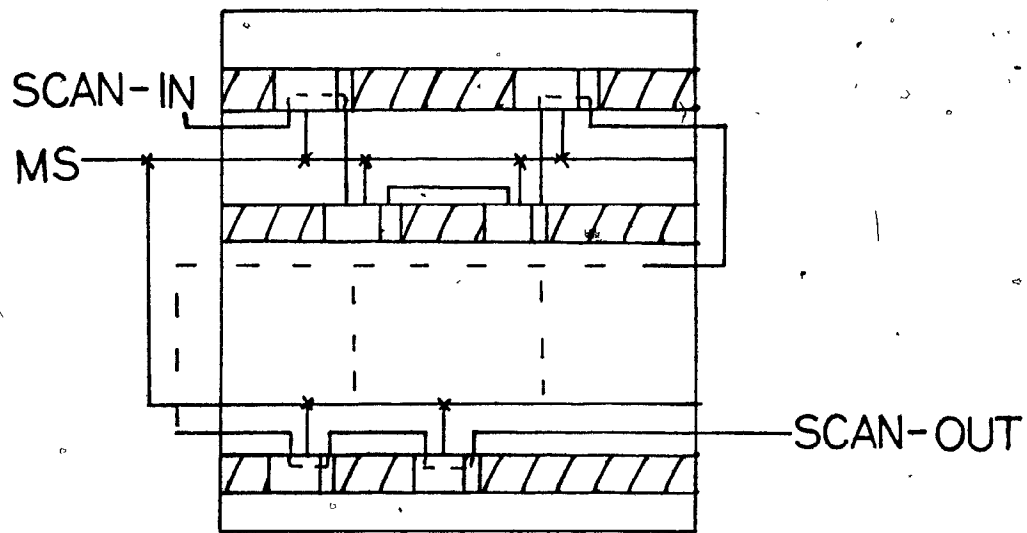


Fig. 3.10 Layout of Scan Circuit

Proposed Scheme

'w' be the width of the pre-scan cell

' h_s ' be the height of the scan cell

' w_s ' be the width of the scan cell

' kh ' be the height of each routing track for some $0 \leq k < 1$

Assumptions made

1. All standard cells on the chip are of same height and width (for simplicity)
2. All standard cells are square i.e $h = w$. Also, height of pre-scan FF and scan FF remains same. $h = h_s$

Total chip area without scan $= A_1$

$$\begin{aligned}
 &= \text{Area of polycell rows} + \text{Area of routing channels} \\
 &= Nh(C + S)w + Nn_1kh(C + S)w \\
 &= Nh^2[1 + n_1k][C + S]
 \end{aligned}$$

Area of the scan register cell,

$$h_s w_s = h w_s = h\left(\frac{5}{3}w\right) = \frac{5}{3}h^2 \quad \text{Increase in width } \frac{5}{3} \text{ is due to scan logic}$$

Chip area using MSRL $= A_2$

$$\begin{aligned}
 &= Nh^2C + \frac{5}{3}Nh^2S + Nn_2kh^2C + \frac{5}{3}Nn_2kh^2S \\
 &= Nh^2C[1 + n_2k] + \frac{5}{3}Nh^2S[1 + n_2k] \\
 &= Nh^2[C + \frac{5}{3}S][1 + n_2k]
 \end{aligned}$$

Proposed Scheme

$$\begin{aligned}
 \text{Area overhead} &= \frac{(A_2 - A_1)}{A_1} \\
 &= \frac{Nh^2[C + \frac{5}{3}S][1 + n_2k] - Nh^2[1 + n_1k][C + S]}{Nh^2[1 + n_1k][C + S]} \\
 &= \frac{(C + \frac{5}{3}S)(1 + n_2k) - (1 + n_1k)(C + S)}{(1 + n_1k)(C + S)} \\
 &= \frac{[(1 + \frac{5}{3}\frac{S}{C})(1 + n_2k)] - (1 + n_1k)(1 + \frac{S}{C})}{(1 + n_1k)(1 + \frac{S}{C})}
 \end{aligned}$$

For designs using MSRL, $n_2 = n_1 + 2$

$$\begin{aligned}
 \frac{(A_2 - A_1)}{A_1} &= \frac{(1 + \frac{5}{3}\frac{S}{C})(1 + (n_1 + 2)k) - (1 + n_1k)(1 + \frac{S}{C})}{(1 + n_1k)(1 + \frac{S}{C})} \\
 &= \frac{2k + \frac{2}{3}\frac{S}{C} + \frac{2}{3}n_1k\frac{S}{C} + \frac{10}{3}k\frac{S}{C}}{(1 + n_1k)(1 + \frac{S}{C})}
 \end{aligned}$$

Let, $n_1 = 30$ (the no. of routing tracks per polycell)

$k = 0.1$ (height of each routing track is 1/10 of height of basic cell)

$\frac{S}{C} = 0.5$ (fraction of sequential cells on the chip)

$$\begin{aligned}
 \text{Therefore, } \frac{(A_2 - A_1)}{A_1} &= \frac{0.2 + \frac{2}{3} * 0.5 + \frac{2}{3} * 30 * 0.1 * 0.5 + \frac{10}{3} * 0.1 * 0.5}{(1 + 30 * 0.1)(1 + 0.5)} \\
 &= 28.33 \%
 \end{aligned}$$

Table 3.1 and Table 3.2 show a comparison of area overhead for MSRL and L2* Scheme. From the two tables, it is clearly shown that designs using MSRLs will reduce area overhead by 2 - 8% when compared to designs using L2* Scheme.

Proposed Scheme

3.4 Comparison With Other Schemes

As discussed earlier, designs using MSRLs can reduce area overhead by 2 - 8 % when compared to designs using L2* scheme. With Saluja's Scheme, it is evident from the logic diagrams for MSRL and PSRL that there is little difference in the cost of the two designs. However, while using PSRLs, the L2 latches cannot be used for other system latches unless $\bar{Q}2$ output is not to be utilized. Whereas, No such restrictions exist for designs using MSRLs. In Saluja's scheme, as pointed out earlier, a better test coverage is possible with reduced test generation effort. Whereas, practical systems using MSRLs need to be investigated to determine fault coverage. The number of additional input/output pins required for MSRL and PSRL are 3. for L2* scheme 4 additional input pins are required. The speed of operation of the three systems is identical. Table. 3.3 lists the important features of MSRL and other existing systems.

Proposed Scheme

$$k = 0.1$$

$$\frac{S}{C} = 0.5$$

n_1	MSRL	L2*
5	38.5%	46.6%
10	34.4%	40.5%
20	30.3%	34.4%
30	28.3%	31.5%
40	27.2%	29.5%

Table 3.1 Comparison of Area Overhead for MSRL and L2*

$\frac{S}{C}$ Constant, n_1 varies

Proposed Scheme

$$k = 0.1$$

$$n_1 = 30$$

$\frac{s}{c}$	MSRL	L2*
0.10	11.3%	13.2%
0.20	16.5%	19.4%
0.30	21.1%	24.0%
0.40	25.0%	27.9%
0.50	28.3%	31.5%
0.60	31.2%	34.3%
0.70	33.8%	37.0%
0.80	36.1%	39.3%

Table 3.2 Comparison of Area Overhead for MSRL and L2*

n_1 Constant, $\frac{s}{c}$ Varies

Proposed Scheme

<i>Schemes</i>	LSSD	L2* Scheme	Saluja's Scheme	Proposed Scheme
<i>Characteristics</i>				
Latch Type	Single	Single	Single	Single
No. of Clock Lines	4 (C1,C2,A,B)	4 (C,C*,A,B)	3 (C,A,B)	2 (C,C*)
No. of Additional Lines due to Scan	3 (SI,A,B)	3 (SI,A,B)	3 (SI,A,B)	2 (SI, MS)
Race-Free	Yes	Yes	Yes	Yes
Hazard-Free	Yes	No	Yes	Yes
Level-Sensitive	Normal and Scan Mode	None	Normal and Scan Mode	Normal and Scan Mode
Performance (Clock Speed)	No Degradation	No Degradation	No Degradation	<u>No</u> Degradation
Design Rules	Comb. has to be partitioned	Comb. has to be partitioned	Comb. has to be partitioned	Comb. has to be partitioned
Other Constraints	None	None	L2 Output can be only used for Complemented Value	None

Table 3.3 Comparison of MSRL with other Schemes

Simulation

4. Simulation

There are many design considerations relating to the design of a level-sensitive polarity-hold shift register latch pair. High performance, low power dissipation, small size and stability are some of the major requirements for a good design. As with any design, there are engineering trade-offs which need to be taken into account to ensure a successful design. The technical details associated with the design of MSRL are documented in this section.

4.1 CMOS Implementation

The MSRL cell is implemented in Static CMOS technology. The circuit was implemented without transmission gates. Since, the logical behavior and faults for the transmission gates are generally not treated by existent Automatic Test Pattern Generators[15, 16, 17]. Furthermore, the failure modes of circuits with such devices can introduce non-classic logic faults[19].

Fig. 4.1 shows a MSRL implemented with 62 transistors. The L1 and L2 latches are constructed using 30 transistors each. $MS=1$ feeds normal data and $MS=0$ feeds scan data to the L1 latch. $C=1$ activates the L1 latch and the L2 latch is activated by $C^k=1$. The two transistors which are encircled in the fig 4.1 represent the additional overhead of the proposed design. The waveforms for this latch are the same as shown in fig. 3.6.

4.2 Physical Design Considerations

The basic latch pair is made of two similar latches. Each latch output utilizes an buffer to eliminate latch degradation due to loading. The latch pair schematic design is

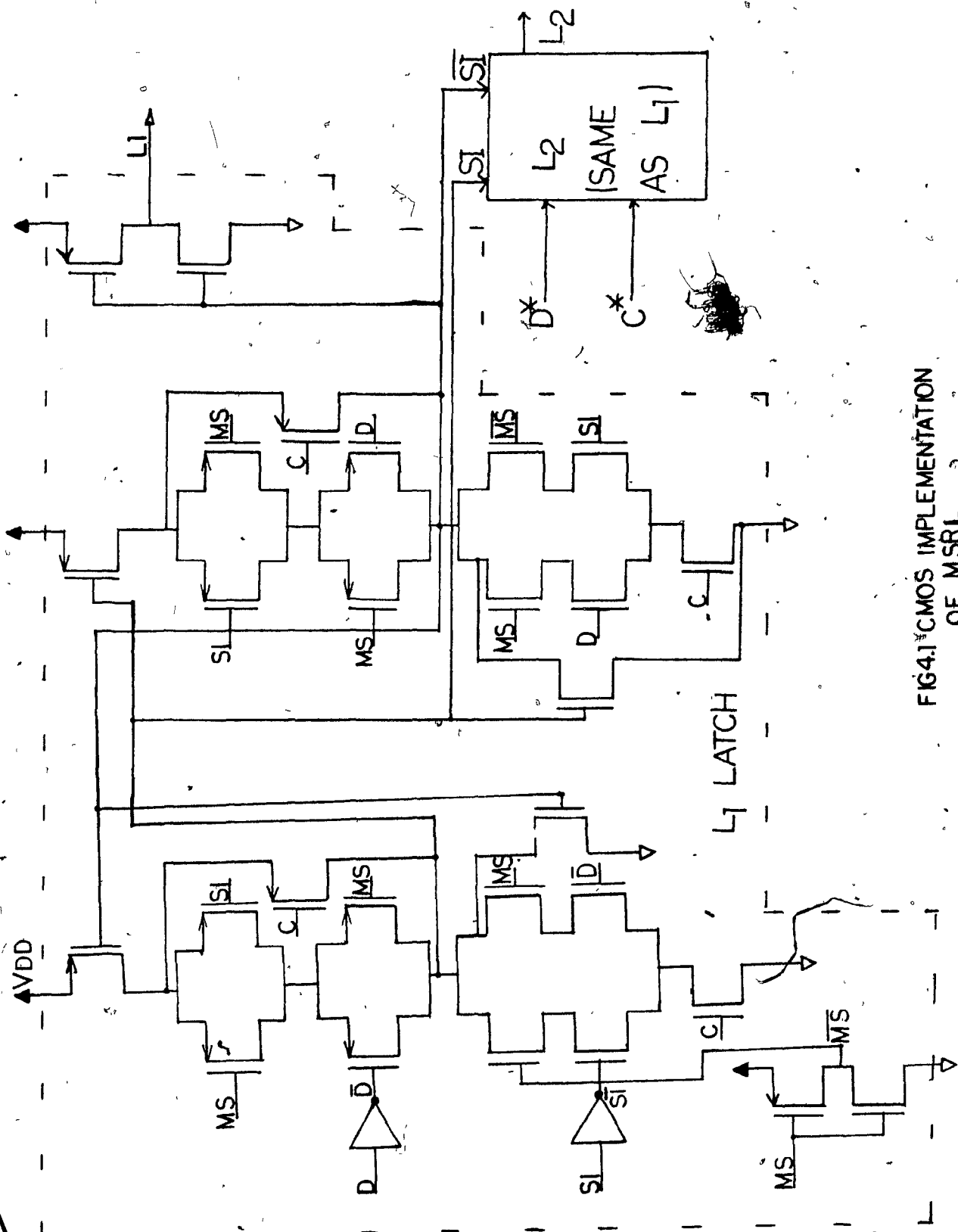
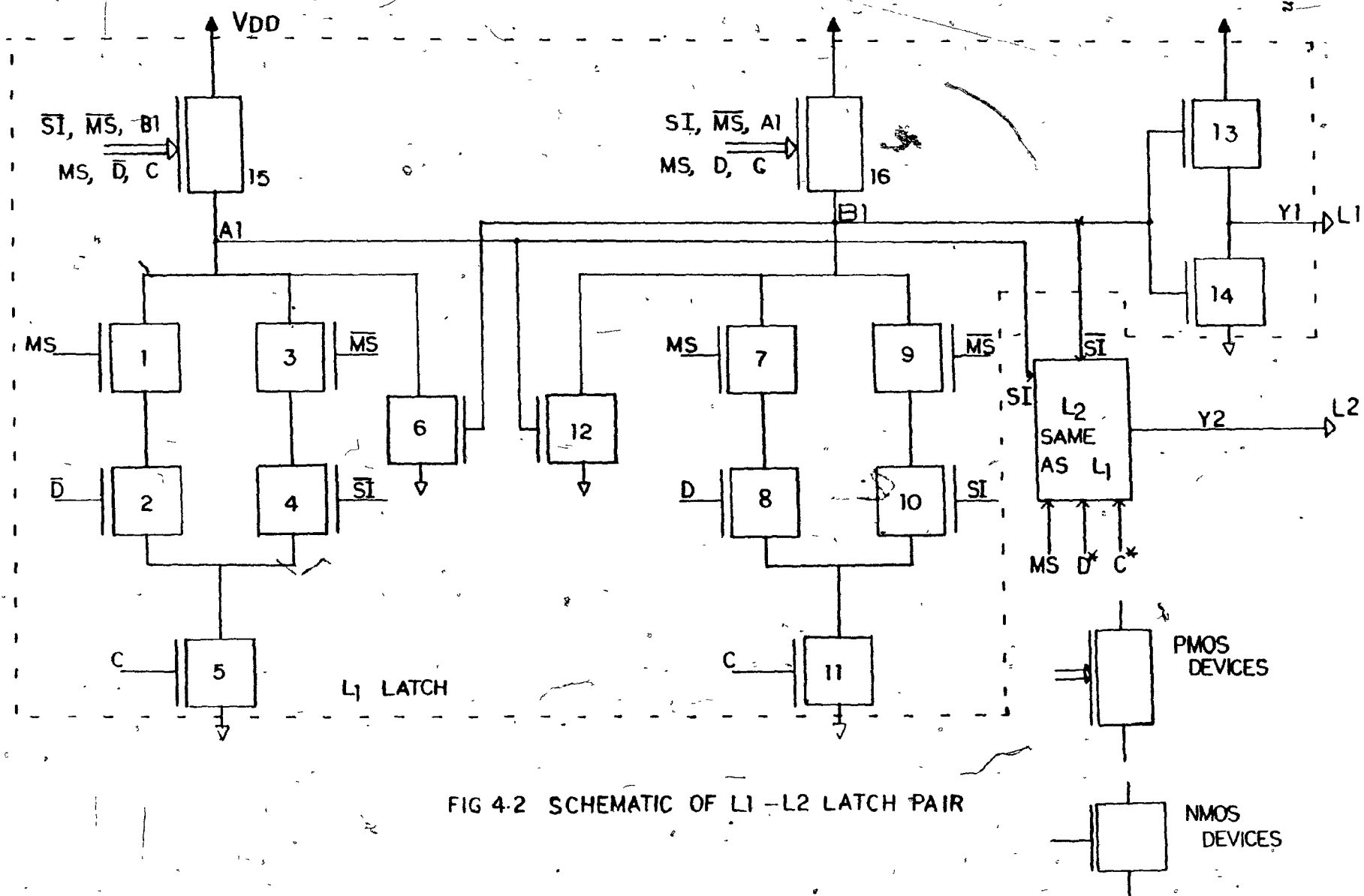


FIG4.1 CMOS IMPLEMENTATION OF MSRL



Simulation

shown in fig. 4.2. With respect to figure, the application of clock 'C' signal allows data present on the data input of the L1 to be latched. The scan input port is held inactive during normal operation and is utilized only during testing.

4.2.1 Latch Output Considerations

Through extensive circuit simulations using SPICE[34], performance improvements to the basic latch structure are identified while minimizing area. Also the device sizes selected for the latch implementation must provide good performance at the lowest power level and be insensitive to process variation. The output buffer provides load isolation to the latch, thereby improving latch set up time and output response. Since the buffer isolates the capacitive load, the power of the latch internal stages can be reduced while the overall performance is improved. The internal nodes of the latch have relatively low values of capacitance (less than 0.2pf) and hence high power circuits are not required. For such low-power latch internal stages the maximum L1 and L2 (Node A and Node B in Fig. 4.2) external capacitance should be limited to 0.1pf. Since this internal node is used only for scan out to an adjacent latch and for driving the output buffer, the above restriction is easily satisfied.

The characteristics of the latch output versus output device sizes are illustrated in fig. 4.3 and fig. 4.4. Here, T_{on} is defined as the delay for the output to fall to 2.5 volts with respect to the clock rising to the 2.5-volt level, T_{off} is defined as the delay for the output to rise to 2.5 volts with respect to the clock rising to the 2.5-volt level. T_{rise} and T_{fall} are delays of the output rising from 1 volt to 3 volts and falling from 3 volts to 1 volt, respectively. T_{avg} is defined as $(T_{on} + T_{off})/2$. Fig. 4.3 shows the delay and output rise time of the latch output as a function of the output pull-up device(device

Simulation

13 in Fig. 4.2) W/L ratio for a fixed capacitive load of 2.0 pF. The device W/L ratio selected was 8. This selection provides performance which is optimized. Fig. 4.4 shows the delay and output fall time as a function of the output pull-down device (device 14 in Fig. 4.2) W/L ratio for the same capacitive load. The device W/L ratio selected was 3. This device size results in an output fall time which is nearly optimum. Fig. 4.5 shows output performance of the latch versus capacitive load. The delays assume data valid prior to the arrival of the clock signal.

4.2.2 Latch Scan-Output Considerations

The L1 output to the L2 latch can be taken from either the A1 or B1 nodes in Fig. 4.2, both of which are internal L1 latch nodes. The A1-node transfer results in noninverted data to the L2, whereas the B1-node transfer results in out-of-phase data to the L2. To obtain an L2 output which is noninverting, the B1-node of the L1 is used for transferring data to the L2. For an inverting L2 output, the L1 latch A1-node is used for transfer to the L2 latch. For MSRL, the scan-output is taken from B1-node instead of from Y1. The speed of the shifting operation is increased by by-passing the large output buffer by 3.4%. At 2pf load, the output response time is 28 nano-seconds when the Scan-Output is taken from Y1. Whereas the output response time is only 21 nano-seconds when the Scan-Output is taken from B1-node. It should be noted that scan-output node has a very limited drive capability, since it is an internal latch node. It is intended to drive the scan-input of an adjacent L1-L2 latch pair only. Therefore, for heavy loading, the L2 latch output Y2 (fig.4.2) should be utilized as the LSSD scan-output.

Simulation

This section has described many design considerations of a MSRL latch pair. The original design goals of good performance, low power dissipation, and small size were met by using high-performance output buffers. Performance characteristics, along with device size selection of the latch, were also presented.

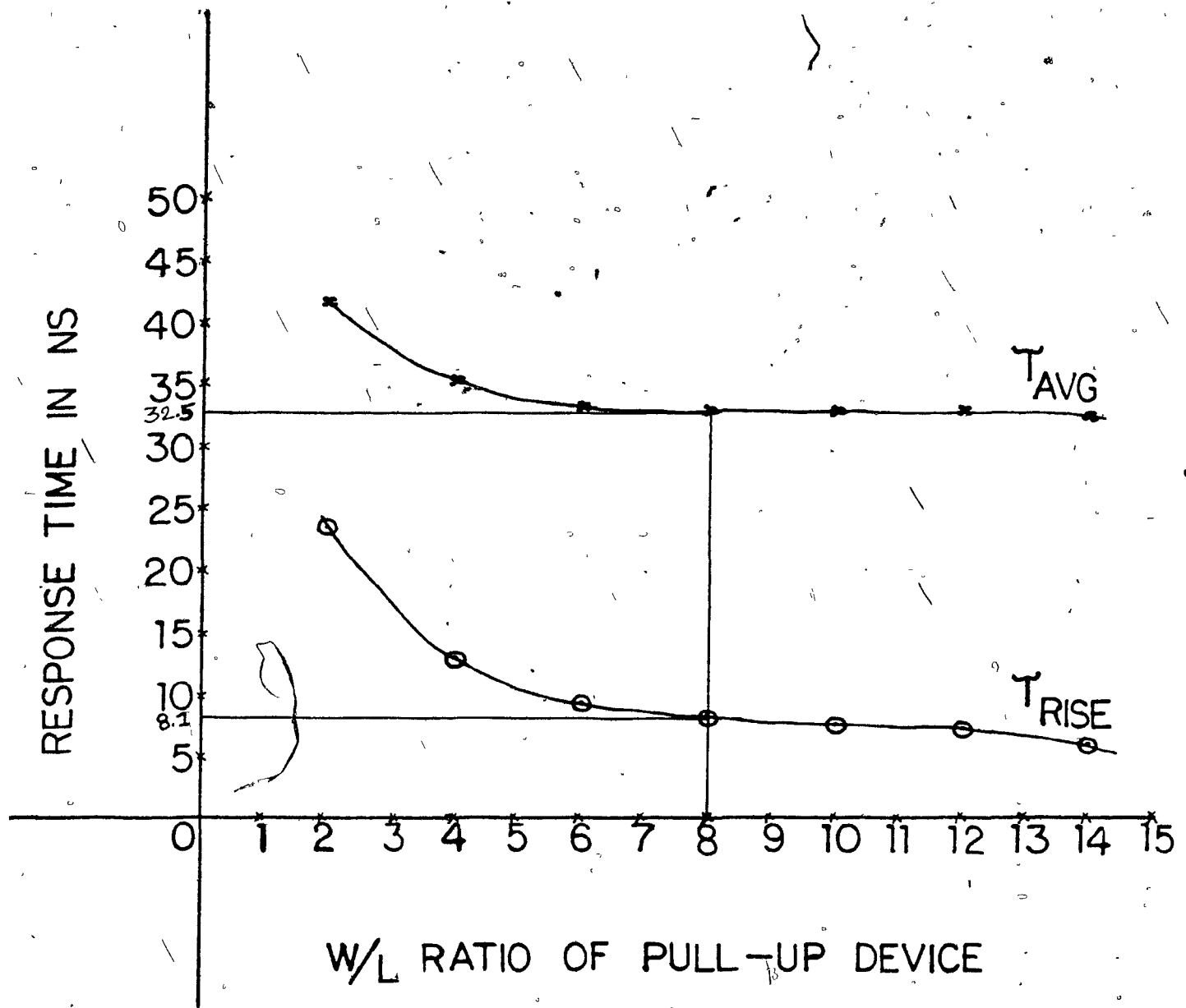


FIG. 4.3 OUTPUT RESPONSE TIME VS W/L RATIO OF OUTPUT PULL-UP DEVICE

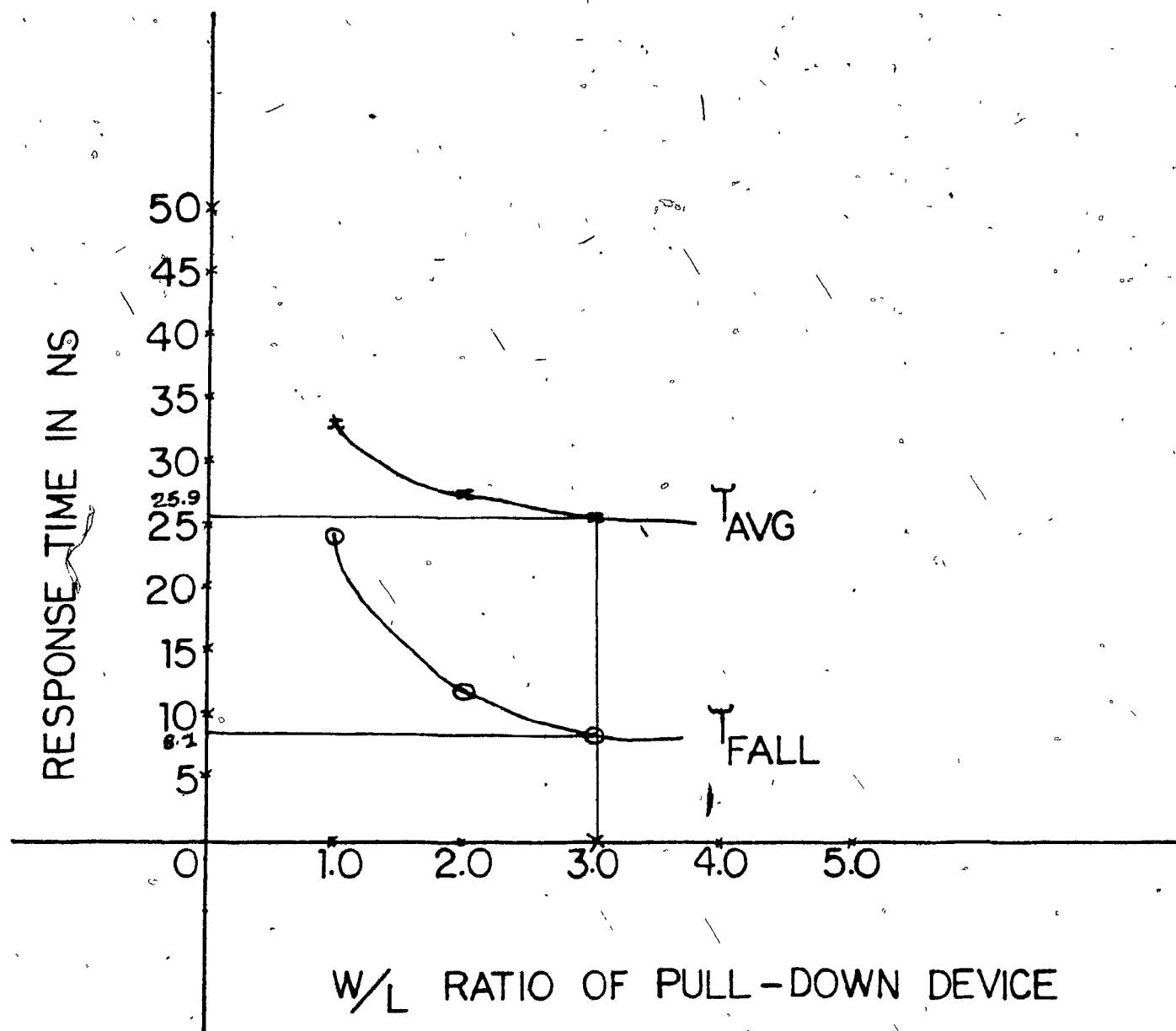


FIG. 4.4 OUTPUT RESPONSE TIME VS W/L RATIO OF OUTPUT PULL-DOWN DEVICE

Simulation

C_L	T_{on}	T_{off}	T_{fall}	T_{rise}
0.25Pf	18.3ns	12.7ns	2.1ns	2.1ns
0.50Pf	20.0ns	15.0ns	3.4ns	3.4ns
1.00Pf	22.5ns	17.1ns	5.0ns	5.0ns
2.00Pf	27.3ns	21.7ns	8.2ns	8.2ns
3.00Pf	31.5ns	26.3ns	12.3ns	12.3ns

Table 4.5 Capacitive Load Vs Circuit Response Time

Conclusion

5. Conclusion

In this thesis, a modified latch was proposed and a study was made of the use of modified latch for designing logic circuits. The latch design proposed is a variation of the existing L2* scheme used in LSSD environment. It was shown that the proposed design reduces the silicon cost of implementing the scan design.

In the proposed latch called MSRL, a 'Mode-Switch' input selects the 'Normal' or 'Scan' mode. The same system clocks C and C* are used in both 'Scan' and 'Normal' modes, thereby reducing area overhead considerably. The reduction of area overhead is due to the elimination of the scan clocks A and B from the shift register latch of the L2* scheme, at the expense of some additional logic (1 inverter per latch pair) in each flip-flop. Hence, in the proposed scheme two clock signals and a mode-switch have to be routed over the whole chip. Whereas in the L2* scheme, two system clocks C and C*, and two scan clocks A and B need to be routed. Typically, in LSI/VLSI circuits, long routing paths are considered more expensive compared to the addition of a few local logic gates. An example of a scan register flip-flop is given to illustrate that the additional logic can be implemented with just two extra transistors.

An important characteristic of the MSRL is that no race or hazard conditions are present during normal operation. The latch is Level-Sensitive. Also, the system performance is not degraded with the use of MSRLs. Advantages and cost impact of this scheme were discussed and a comparison was made with other existing systems. A set of design rules that will result in Level-Sensitive logic were also described.

The technical details associated with the design of the shift register latch were documented and performance improvements were identified through extensive circuit

Conclusion

simulations. The design goals of good performance, low power dissipation, and small size were met by using high performance output buffers. Since the latch is used as a storage element whose output performance is essentially determined by its output buffer, the power of the latch internal stages is reduced while the overall performance is improved. Performance characteristics, along with device size selection of the latch were also presented.

References

- [1] Breuer, M.A, *Diagnbsis and Reliable Design of Digital Systems*. Woodland Hills, California 1976.
- [2] Chang, H.Y., Manning, E.G. and Metze, G., *Fault Diagnosis of Digital Systems*. New York : Wiley-Interscience, 1970.
- [3] Friedman, A.D. and Menon, P.R., *Fault Detection in Digital Circuits*. Englewood Cliffs, New Jersey : Prentice- Hall, 1971.
- [4] Williams, T.W and Parker, K.P, 'Testing Logic Networks and Design for Testability', IEEE Computer, Volume 12, October 1979, pp. 9-21.
- [5] Batni, R.P. and Kime, C.R., 'A Module Level Testing Approach for Combinational Networks', IEEE Transactions on Computers, Volume C-25, June 1976. pp. 594-604.
- [6] Breuer, M.A. and Friedman, A.D., 'Functional Level Primitives in Test Generation', IEEE Transactions on Computers, Volume C-29, March 1980, pp. 223-234.
- [7] Schertz, D.R and Metze, G., 'A New Representation for Faults in Combinational Digital Circuit', IEEE Transactions on Computers, Volume C-21, August 1972, pp. 858-866.
- [8] Bennetts, R.G., *Introduction to Digital Board Testing* New York : Crane Russak and Company, Inc. 1982.
- [9] Goel, P., 'Test Generation Costs Analysis and Projections', IEEE Proceedings 17th Design Automation Conference, June 1980, pp. 77-84.
- [10] Hayes, J.P and Friedman, A.D, 'Test Point Placement to Simplify Fault Detection', Digest of Papers International 1973 Symposium on Fault-Tolerant Computing, June 1973, pp. 73-78.

- [11] Williams, T.W., and Parker, K.P., 'Design for Testability - A Survey', IEEE Transaction on Computers, Volume C-31, January 1982, pp. 2-15.
- [12] Bennetts, R.G., *Design of Testable Logic Circuits* Addison-Wesley Publishing Company, 1984.
- [13] Eichelberger, E.B and Williams, T.W., 'A Logic Design Structure for LSI Testing', IEEE Proceedings 14th Design Automation Conference, June 1977 pp. 462-468.
- [14] Michael, J.Y and Angell, J.B., 'Enhancing Testability of LSI Circuits Via Test Points and Additional Logic', IEEE Transactions on Computers, Volume C-22, January 1973, pp. 46-60.
- [15] Roth, J.P, Bouricuis, W.G and Schneider, P.R., 'Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits', IEEE Transactions on Electronic Computers, Volume EC-16, October 1967, pp. 567-580.
- [16] Goel, P and Rosales, B.C., 'Podem-X An Automatic Test Generation System for VLSI Logic Structures', Digest of Papers, 1980 International Symposium on Fault-Tolerant Computing, October 1980, pp. 145-151.
- [17] Fujiwara, H. and Shimono, T. 'On the Acceleration of Test Generation Algorithms', Digest of Papers 1983 International Symposium on Fault-Tolerant Computing, June 1983, pp. 98-105.
- [18] Mei, K.C.Y., 'Bridging and Stuck-At Faults', IEEE Transactions on Computers, Volume C-23, July 1974 pp. 720-727.
- [19] Wadsack, R.L., 'Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits', The Bell System Technical Journal, Volume 57, Number 5, May-June 1978, pp. 1449-1473.
- [20] Goldstein, L.H., 'Controllability/Observability Analysis for Digital Circuits', IEEE Transactions on Circuits and Systems, September 1979, Number 9, pp. 685-693.

- [21] Bennetts, R.G., Maunder, C.M., and Robinson, G.D., 'CAMELOT : A Computer-Aided Measure for Logic Testability', IEE Proceedings, 1981, Volume 128, Part E, Number 5, pp. 177-189.
- [22] Grason, J., 'TMEAS - A Testability Measurement Program', IEEE Proceedings 16th Design Automation Conference, June 1979, pp. 156-161.
- [23] Goldstein, L.M. and Thigpen, E.L., 'SCOAP : Sandia Controllability / Observability Analysis Program', IEEE Proceedings 17th Design Automation Conference, June 1980, pp. 190-196.
- [24] Fanatsu, S., Wakatsuki, N. and Arima, T., 'Test Generation Systems in Japan', IEEE Proceedings 12th Design Automation Conference, June 1975, pp. 114-122.
- [25] Stewart, J.H., 'Future Testing of Large LSI Circuit Cards', Digest of Papers Semiconductor Test Symposium, October 1977, pp. 6-15.
- [26] Ando, H., 'Testing VLSI with Random Access Scan', Proceedings 20th IEEE Computer Society International Conference, February 1980, pp. 50-57.
- [27] Godoy, H.C., Franklin, G.C. and Bottorff, P.S., 'Automatic Checking of Logic Design Structures for Compliance with Testability Ground Rules', IEEE Proceedings 14th Design Automation Conference, June 1977, pp. 469-478.
- [28] Saluja, K.K., 'An Enhancement of LSSD to Reduce Test Pattern Generation Effort and Increase Fault Coverage', IEEE Proceedings 19th Design Automation Conference, June 1982, pp. 489-494.
- [29] Dasgupta, S., Goel, P., Walther, R.G. and Williams, T.W., 'A Variation of LSSD and its Implications on Design and Test Pattern Generation in VLSI', IEEE Digest of Papers International Test Conference, November 1982, pp. 63-66.
- [30] Edwards, H.F., *The Principles of Switching Circuits*. Cambridge : The M.I.T Press, 1973.

- [31] Agrawal, V.D., Jain, S.K. and Singer, D.M., 'Automation in Design For Testability', IEEE Proceedings 21th Design Automation Conference, June 1984, pp. 159-163.
- [32] Nagel, L.W., 'SPICE2: A Computer Program to Simulate Semiconductor Circuits', ERL Memo. Number ERL-M520, Electronics Research Laboratory, University of California, Berkeley, 1975.

Appendix A

Eichelberger and Williams[13] presented a set of design rules or constraints that will result in Level-Sensitive and Scan Design:

Rule 1: All internal storage is implemented in hazard-free polarity-hold latches.

Rule 2: The latches are controlled by two or more non-overlapping clocks such that:

- (a) A latch, X, may feed the data port of another latch, Y, if and only if the clock that sets the data into latch Y does not clock latch X.
- (b) A latch, X, may gate a clock C_i to produce a gated clock C_{ig} which drives another latch, Y, if and only if clock C_{ig} does not clock latch X, where C_{ig} is any clock derived from C_i .

Rule 3: It must be possible to identify a set of clock primary inputs from which the clock inputs to SRLs are controlled either through simple powering trees or through logic that is gated by SRLs and/or non-clock primary inputs. Given this structure, the following rules must hold :

- (a) All clock inputs to all SRLs must be at their 'off' states when all clock primary inputs are held to their 'off' state
- (b) The clock signal that appears at any clock input of an SRL must be controllable from one or more clock PIs such that it is possible to set the clock input of the SRL to an 'on' state by turning any one of the corresponding clock PIs to its 'on' state and also setting the required gating conditions from SRLs and/or non-clock PIs.
- (c) No clock can be ANDed with either the true or complement value of another clock.

Rule 4: Clock primary inputs may not feed the data inputs to latches either directly or through combinational logic, but may only feed the clock input to the latches or primary outputs.

Rule 5: All SRLs must be interconnected into one or more shift registers, each of which has an input, an output and shift clocks available at the terminals of the package.

Rule 6: There must exist some primary input sensitizing condition (referred to as the scan state) such that:

- (a) Each SRL or scan-out PO is a function of only the single preceding SRL or scan-in PI in its shift register during the shifting operation.
- (b) All clocks except the shift clocks are held 'off' at the SRL inputs
- (c) Any shift clock to an SRL may be turned 'on' and 'off' by changing the corresponding clock primary input for each clock.