Proceedings Seventeenth Asilomar Conf. on Circuits Systems & Computers, Oct 31-Nov 2, 1983, Pacific Grove

IMPROVING TESTABILITY BY USING ADDITIONAL CIRCUITS *

Vojin G. Oklobdzija

IBM T.J.Watson Research Center
P.O.Box 218
Yorktown Heights, NY 10598

ABSTRACT

The problem of testing VLSI logic can be simplified by proper partitioning. In this way the logic can be tested either by the random test patterns, or by deterministic test vectors. However, the technique used to generate partitions needs to be adapted to the VLSI environment where any impact on the area and speed should be minimized. In this paper, the circuit structures (C- logic) are introduced, and their impact on testability and circuit performance is evaluated. A testing methodology using these structures is proposed. Through the use of the multiplexing properties of these structures (C' , C'' logic) logic can be made random pattern testable [3]. This same C-logic when applied to the sequential circuit with particular properties, makes possible the test which does not suffer from the test time penalties of LSSD. It is possible to create the partitions in combinational logic and simplify the testing of hardware by the "divide and conquer" method.

INTRODUCTION

Several methods to improve testability of VLSI logic have been proposed.

To improve observability and controllability of digital logic, input and output test points are added [5],[10]. However, this method is not convenient for the LSI/VLSI circuits due to the pin limitation. Other methods such as LSSD suffer from the number of shift-in/shift-out sequences which contribute substantially to the testing time and make the testing method inherently dependent on the external tester.

The technique proposed in [7] adds multiplexers to facilitate partitioning of the chip into partitions which can be tested exhaustively, so that fault-modeling can be eliminated. The added multiplexing circuitry introduces an additional delay in the signal propagation path and considerable hardware overhead [6].

The method proposed here is to embed the desired features in the existing circuitry. The notion of "controllable logic" is introduced. This is achieved by adding an extra MOS device at a particular place in the combinational logic. In this way the overhead and the decrease in performance are minimized.

CONTROLLABLE LOGIC [8]

The principal idea of controllable logic is to allow for two modes of operation of each C-gate: **normal** mode and **test** mode. In the **test** mode, selected logic gates are forced to the stage where the logic state at the output is solely dependent on the logic state of a preselected priority input. The mode of operation is determined by the control signal **C** which has control over the C-gates. When the control signal C has the logic value of C=0 (**normal** mode) the specified function is implemented by the given logic. The output of the logic in the **normal** mode assumes the logic value determined by its truth-table.

In the case where C=1, (**test** mode), the logic value of the output becomes independent of all but one input which has been previously assigned. The controllable logic becomes an inverter. Function **f** of the controllable logic gate is described as:

$$f(x1,x2,\ldots,xn) = \begin{cases} q(x1,x2,\ldots,xn) & \text{if } C=0 \\ \text{NOT}(xk) & \text{if } C=1 \end{cases}$$

Logic representation of the C-gate is shown in Fig.1. An arrow is used to indicate the priority input.

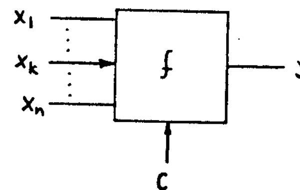


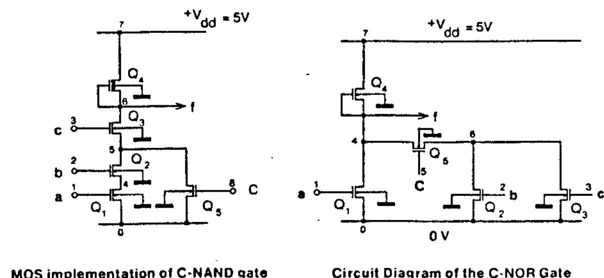
Fig. 1.

*Supported in part by the Office of Naval Research Contract No.N00014-79-C-0866

CH1939-8/83/0000/0118$01.00©1984 IEEE

In the examples given throughout the text n-MOS technology will be considered. However, the concept of C-logic is not limited only to n-MOS technology and could be applied to the other technologies as well.

The n-MOS implementation of the controllable NAND and NOR gate is shown.



MOS implementation of C-NAND gate — Circuit Diagram of the C-NOR Gate

TEST	a b c C	f
1	0 0 0 1	1
2	1 0 0 1	0
3	0 1 0 1	0
4	0 0 1 1	0
5	0 1 - 0	1

TEST	a b c C	f
1	1 1 1 0	0
2	1 1 0 0	1
3	1 0 1 0	1
4	0 1 1 0	1
5	0 - 1 1	0

Fig. 2.

It is important to note the effect of adding an extra device (transistor Q5) to the gate in terms of the increase of the propagation delay caused by such a modification. The actual layout of the gate, the effects on wirability and increase in area are considered. Newly introduced potential faults and their detectability by such a method is also studied [4],[8],[9].

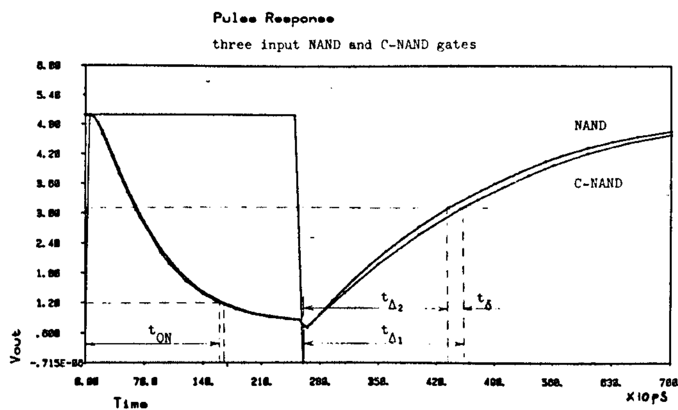
Propagation Delay of Controllable NAND and NOR

The addition of transistor Q5 is expected to have an effect in increasing propagation delay of the gate. The transistor Q5 has associated capacitances which will be added to the node, and in the case of a C-NOR gate the resistance of the transistor connected in series is also taken into account.

To obtain realistic results, a layout of the NAND and NOR gates is performed assuming 2.5 micron geometry. A simulation is done using the circuit simulation program SPICE [10], taking into account all the parasitics introduced [8].

It is observed in both cases that the increase in the propagation delay is not significant.The increase amounts to only 8.6% , for NAND (Fig. 3. a.), and 20% for NOR (Fig. 3. b.) of the propagation delay of the gate [8],[9].

Area increase is estimated to be approximately 20% of the gate area, due to the introduction of the extra device. The total increase is dependent on the number of gates replaced with the C-gates and it is estimated by dividing the area increase with the total area of the logic. In the example which was considered,area increase amounted to a total of 2% [8],[9].



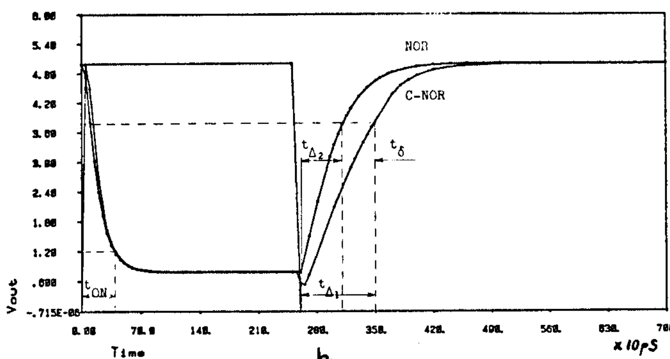
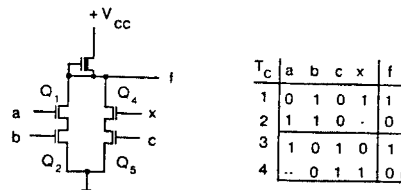


Fig. 3. a. b.

MULTIPLEXING WITH C-LOGIC

Improvement in testability is achieved by gaining access to the internal nodes of the logic [5]. This is achieved by adding multiplexers to the points which are directly connected to the primary inputs / outputs. One such place is logic associated with the off-chip drivers or input pads. Through the use of multiplexing, internal logic blocks can be accessed from the primary inputs / outputs and tested as isolated partitions. If the partitions are of sufficiently small size, exhaustive tests can be applied by using random patterns (RP) as input test vectors. However, in order to improve testability without excessive time / area penalties, some sort of low cost multiplexing circuitry is needed [3]. The addition of this circuitry should not have a detrimental effect on the overall testability of the given logic. It is proposed here [8],[9] to use internal logic, which is done without significantly degrading the circuit performance. In the **test** mode this multiplexer allows for sharing of the logic used in the **normal** mode of operation. By introducing multiplexers, the primary inputs / outputs of the VLSI circuit are shared during the **test** mode of operation. Which nodes will be multiplexed to which I/O paths, and the time sequence used in the test mode is determined in the best way from the given circuit topology. For this purpose

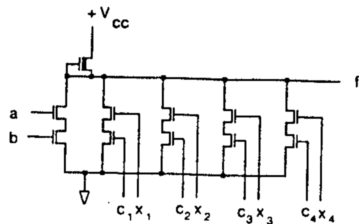
we are introducing a special logic with multiplexing properties; **C'-logic** (Fig. 4.) .



Tc	a	b	c	x	f
1	0	1	0	1	1
2	1	1	0	-	0
3	1	0	1	0	1
4	--	0	1	1	0

(a.) (b.)

C'-gate : (a) Circuit Diagram (b) Test for the C'-gate



Modification of the C'-gate

Fig. 4.

When the output of the C'-gate is directly connected to the primary output, the propagation mode allows the node connected to line x to be directly observable. Function of the C'-gate f' is :

$$f' = \text{NAND}(a,b) \quad \text{if } C=0$$

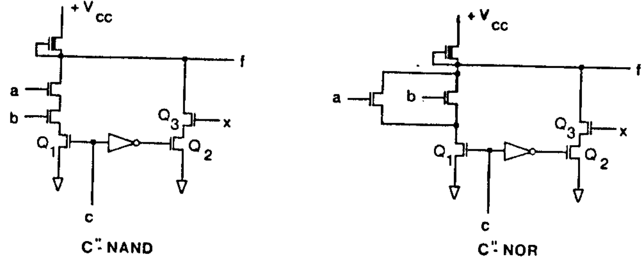
$$f' = \overline{x} \quad \text{if } C=1 ,\ ab=0$$

This has a favorable impact on testing: by gaining access to the nodes which are not easily observable, it reduces the length of the test. The C'-gate provides the means of inserting the low cost multiplexers embedded in the logic. The overhead is not significant since only two extra devices are being used for each access point. However, due to the introduction of the additional capacitance, propagation delay of the C' gate is increased, compared to the regular NAND gate. This increase is comparable to the additional delay introduced by C-NAND, and is significantly smaller than the increase resulting from the addition of an extra gate. It is found that the degradation in performance amounts to approximately 10% [8].

The condition for setting one of the remaining inputs to 0 is not difficult to meet since in the propagation mode the gate is used only to propagate the information from some other part of the logic under test [1]. Here, we assume that the C' logic block is chosen in such a way that the test to be propagated has no effect on the status of the logic containing the C' logic. The logic itself is tested in the **normal** mode.

The C''-logic illustrates the multiplexing function embedded in the existing logic of a VLSI structure. Implementation details regarding commonly used NAND and NOR gates are shown in **Fig.** 5.

$$f1 = \text{NAND}(a,b) \quad \text{if } C=0 \quad \text{else} \quad f1 = \overline{x}$$
$$f2 = \text{NOR}(a,b) \quad \text{if } C=1 \quad \text{else} \quad f2 = \overline{x}$$



Circuit Diagram for the C''-gate

Fig. 5.

Three extra devices, Q1 , Q2 , and Q3, and an extra inverter are used. This amounts to approximately two extra gates per control point C. The time penalty for introducing this type of multiplexer amounts to 10%. It should be noted that this is due to the implementation of the multiplexer, where the devices Q1 and Q2 are connected to the ground side of the gate. The extra capacitance introduced by devices Q1 and Q2 is minimized because they have virtually been grounded. The circuit is affected by the ON resistance of device Q1 (which can be made small by properly dimensioning Q1), and the capacitance introduced by device Q3. With careful circuit design steps, the time-area penalty caused by introducing the multiplexer, is minimized.

TESTABILITY

This method is based on the partitioning of the combinational part of the digital logic into relatively small partitions which can be tested in two ways :

1. Exhaustively

The elimination of the need for test pattern generation has these beneficial effects :

(a) the cost of testing is reduced by eliminating the high cost associated with generation of test patterns

(b) the storage requirement for the test patterns is reduced

(c) it is possible to generate the test on the chip itself, a significant step forward in achieving the self-test feature on the chip

2. By generating test patterns for the given partitions.

(a) Partitioning of the logic helps to simplify the problem of test generation by reducing the cost associated with it.

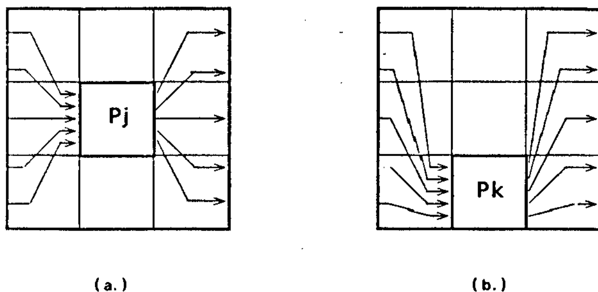
The main objective of this method is to achieve testability without excessive time and area penalties. It is very important that the introduction of testability has minimal or no effect on the performance. Thus, any technique which requires significant additional chip area is ruled out.

With the method proposed here, testability of the combinational portion of the VLSI logic is enhanced in the following way (Fig.6.) :

(a) In the test mode, the logic is partitioned into small manageable partitions. The way in which the logic is partitioned and the size of the partitions are decided during the design stage. This involves the knowledge of the function, and ideally, the circuit is divided into logically and functionally complete blocks

(b) By creating the paths to and from the partition in order to control the inputs and observe the outputs

(c) If the partition is not testable in one of the two ways; the logic of the partition is modified in the test mode, or the partition is further subdivided



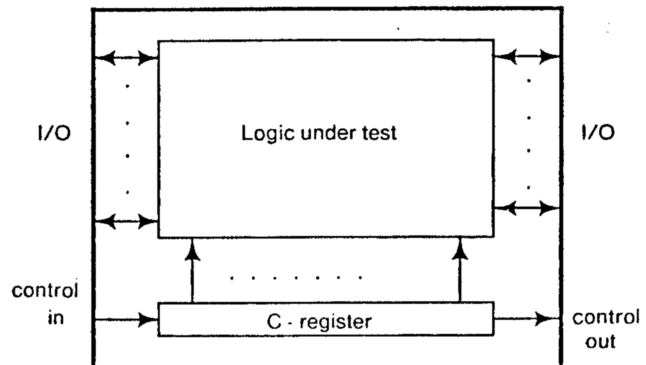
Testing the Combinational Logic of a VLSI Chip by Partitioning

Fig. 6.

During the design stage, the number of gates which are replaced by C-logic is determined in such a way that the prohibition of certain paths and the creation of others facilitate the division of the logic into sub-blocks or partitions. When the logic does not physically consist of separable partitions or the partitions cannot be accessed by "sensitization", C-logic block is introduced.

In the test mode, the partition to be tested **Pk** is chosen and then the logical values on the control lines are selected in such a way that the partitioning is achieved and the access to the partition from (to) the primary inputs (outputs) is created. The test for this particular partition is applied from the primary inputs and the test response is observed on the primary outputs. If it passes the test, another partition **Pj** is chosen and the process is repeated until the entire logic is covered.

Since the number of control lines can be significant, and therefore prohibitive to the use of external I/O pins, it is proposed that the control lines be controlled from an internal register (C-register) the content of which is "scanned-in" from the I/O pin added for that purpose.



Control Register C in the VLSI Circuit

Fig. 7.

The C-register is a serial-input parallel-output shift register (Fig.7.). Each cell of the register has a control over one or several C-logic blocks. The content of the C-register is observed through the additional I/O pin which is introduced in order to test the C-register. If the VLSI chip is designed with self-test capabilities, the C-register can be treated as an internal register which can be loaded from the micro-control memory.

The test procedure is performed in the following way:

(a) the partition to be tested is selected. The content of the C-register is loaded and the test for Pk is applied from the primary inputs

(b) the expected value (signature) is compared with the obtained result

(d) upon completion of the test for the particular partition Pk, select the next partition, Pj, and repeat steps (a) and (b) until the entire logic is covered

TESTING SEQUENTIAL LOGIC

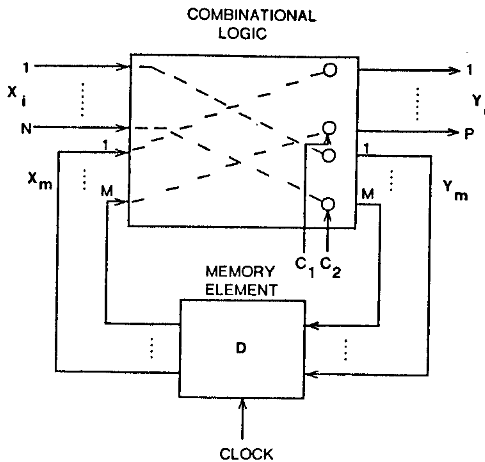
Testing of sequential logic is much more difficult than that of combinational logic [1],[2].

One of the most successful approaches yet, has been the LSSD approach [12],[13]. However, the LSSD test approach suffers from the excessive execution times which are caused by the long sequences of serial shift-in and shift-out operation. The length of the scan is the longest of the total of the length of all the registers connected together in the LSSD

chain, which can be quite a long sequence. The serial link between all of the registers is one of the essential requirements of the LSSD approach.

The goals of the proposed method are to minimize the execution time and make the time needed for setting the initial conditions shorter.

The general model represents the logic as a block consisting of the combinational logic and the memory elements in the feedback (Fig.8.). There are N primary input lines, Xi, and P primary output lines, Yi, through which the circuit communicates with the outside world. There are M feedback lines (M < P, M < N), Ym, going from the outputs of the combinational part to the memory elements, and M lines, Xm, going from the memory elements to the input lines of the combinational part. The circuit is synchronous which assures that there are no races and the circuit is hazard free in both modes of operation : **test** and **normal** .



General Model of a Sequential Circuit

Fig. 8.

The total of 2M control nodes, Ci, is introduced: M at the primary outputs, Yi, and M at the outputs of Ym. They consist of the mix of C, C' and C'' logic, which is used in the **normal** mode of operation.

In the **normal** mode both control lines C1 and C2 are set to the values (N) which enable the signal flow in the **normal** operation.

In the **test** mode, a single-step mode of operation is set and the test is performed according to the procedure S. Control lines C are selectively set to the value (T) which enables the signal flow for the **test** mode. In the single-step mode the control over the clock is relinquished to the tester.

PROCEDURE S:

1. Testing Memory Elements

a. Set the test vector Xi at the primary inputs; set , C1 = N , C2 = T.

b. Apply one clock cycle and store the outputs Ym into the memory elements.

c. Set C1 = T ; C2 = N , and read the contents of memory elements from the primary outputs Yi.

d. Repeat the process until the memory test is completed.

2. Testing Combinational Logic

a. Set the test vector Xi' to appear on Xm Set C2 = T , C1 = N.

b. Advance the clock and store Xi in the memory elements

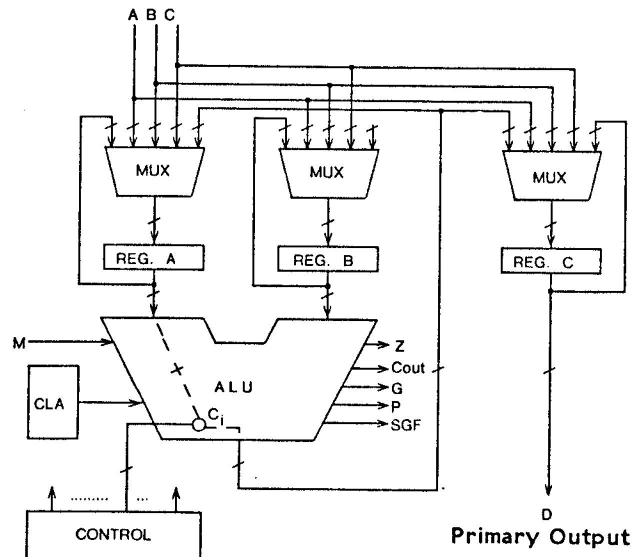
c. Set the test vector Xi'' on the input and observe the output. The output on Yi' is a response to the test Xi' Xi'' on the outputs Yi.

d. Advance the clock and store Ym into the memory elements.

e. Set C1 = T , C2 = N and observe Yi'' which is the response of the test Xi' , Xi'' at the Ym outputs.

3. Repeat until completed

The limitation of this approach is that the method is applicable only when M < N and M < P . In



Example of a Testable RALU Unit

Fig. 9.

the other cases other solutions, such as LSSD, are favorable. However, in many practical cases, the situation where M ≤N , M ≤ P is quite common. One such case occurs in the data path where the width of the registers is equal to the width of the busses (Wr = Wb). An example of a RALU unit [14] is shown in Fig. 9. By multiplexing only one path through the ALU, this RALU is tested by the procedure described.

CONCLUSION:

The problem of modifying an existing circuit by adding a limited amount of logic to simplify fault detection and diagnosis is of great importance in designing for testability. One way of improving diagnosibility is by adding lines from the internal points to the externally observable test points [11]. These points can be accessed temporarily by the test probes or they might be connected to the primary outputs of the circuit. However, the number of test points which can be inserted is severely limited by the pin limitation of the VLSI circuit and is impractical, given the very high gate to pin ratio. This limitation is overcome by using the multiplexing properties of C-logic, in which case commonly used I/O pins are used for propagation of the signals otherwise deeply buried inside the VLSI logic.

This paper addresses the problem of VLSI testing and designing of testable structures. It presents an approach for improving the testability of VLSI systems by introducing the concept of controllable logic (C-logic) and a design technique for its use. The primary objective of the approach is to increase the access to the internal points in the VLSI system without significantly changing the area and timing.

By replacing selected portions of the VLSI logic with the C-logic, RP (Random-Pattern) testability is achieved through partitioning into manageable partitions. Also, when a deterministic test is selected the C-logic in the test mode is utilized to create conveniently chosen partitions of the logic. By testing the partitions rather than the whole system, the problem of testing a large structure is simplified using the "divide and conquer" approach.

REFERENCES

[1] J.P.Roth, Computer Logic Testing and Verification, Computer Science Press, 1980.

[2] H.Y.Chang,L.Manning,G.Metze, Fault Diagnosis of Digital Systems, Wiley-Interscience, 1970.

[3] E.B.Eichelberger and E.Lindbloom, Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test, IBM Journal of Research and Development, Vol.27, No.3, pp.197, May 1983.

[4] J. Galiay et. al., Physical Versus Logical Fault Models in MOS LSI Circuits, Impact on Their Testability, Proceedings of FTCS-9, Madison-Wisconsin, p.195, June 1975.

[5] L.H.Goldstein, Controllability/Observability Analysis of Digital Circuits, IEEE Transaction on Circuits and Systems, Vol.CAS-26, No.9, p.685, September 1979.

[6] E.J.Mc Cluskey, S.Bozurgui-Nesbat, Design for Autonomous Test , IEEE Transaction on Computers, Special issue on Design for Testability, Vol. C-30,NO.11,p.866, November 1981.

[7] E.J.Mc Cluskey, S.Bozurgui-Nesbat, Design for Autonomous Test , Digest of Papers 1980 Test Conference, Cherry Hill, New Jersey ,1980.

[8] Vojin G. Oklobdzija, Design for Testability of VLSI Structures through the Use of Circuit Techniques, Report No. CSD 820820 , University of California Los Angeles, August 1982.

[9] Vojin G. Oklobdzija et all, Testability Enhancement of VLSI Using Circuit Structures, Proceedings of the International Conference on Circuits and Computers ICCC82, September 28 - October 1, 1982, New York Hilton, New York.

[10] SPICE-Users Manual, University of California Berkeley, 1978.

[11] J.P.Hayes, A.D.Friedman, Test Point Placement to Simplify Fault Detection, IEEE Transaction on Computers, Vol.C-23, No.7.,p.727, July 1974.

[12] M.J.Y.Williams, J.B.Angell, Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic, IEEE Transactions on Computers, Vol.C-22,p.46., January 1973.

[13] E.B.Eichelberger, T.W.Williams, A Logic Design Structure for LSI Testing, Proc. 14th Design Automation Conference, p.462,June 1977.

[14] Fumihiko Sato et all, A 400-pS Bipolar 18b RALU, Digest of Papers IEEE International Solid-State Circuits Conference, Vol.23., No. 80CH1490-2SCC, p.114., February 14, 1980.