1986

Implementation of testability in VLSI circuits /

Cu T. Than
Lehigh University

IMPLEMENTATION OF TESTABILITY

IN VLSI CIRCUITS

by

Cu T. Than

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1986

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

(date)
September, 18th, 1986

Professor in Charge
Gianni H. Colotti

Chairman of the Department
Lawrence J. Vanness

ACKNOWLEDGMENTS

The author wishes to express his appreciation to individuals whose contributions have made completion of this project possible: Dr. M. H. White for guidance and C. W. Spivak for review of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

IMPLEMENTATION OF TESTABILITY

IN VLSI CIRCUITS

by

Cu T. Than

ABSTRACT

As the complexity of digital circuits increases, the most outstanding problem has been how to efficiently test these large circuits . One solution for the testing problem is to incorporate the testability into circuits. Design for Test relies on three types of techniques: the ad hoc approaches, the structured approaches, and the built-in self test approaches. The ad hoc technique that is normally used for semi-custom chips is the test point approach. In term of area overhead incurred by test circuit, this approach is an efficient way to implement testability. However, its implementation is not always straightforward and usually requires long design time for manual test vector generation. The structured design for test approaches center around a uniform design method that converts latches into shift registers. The testability implementation with a structured design approach is simple and normally automated; however, significant area overhead is incurred in the process. It is this overhead that prevents the widespread use of the structured design approach in semi-custom designs.

This thesis proposes a design for test applicable to semi-custom designs, that minimizes area overhead while achieving high fault coverage. In this technique, a selective number of latches are chosen based on the functional test result and their testability values. The testability implementation and vector generation are automated with the use of an available testability system. The thesis reviews various design for test approaches and their potential applications in semi-custom designs. The testability measures and fault model are described in the thesis.

I. INTRODUCTION

A. Historical Review of Testability

Integrated Circuit Technology is moving from Large-Scale Integration (LSI) to Very-Large-Scale Integration (VLSI) and quickly approaching Ultra-Large-Scale Integration (ULSI). The increase in chip complexity has brought a decrease in cost per logic gate, along with improvement in performance. However, the problem of determining in a cost effective way whether a chip has been properly tested to insure its quality is still being solved. It is believed that economical testing can only be obtained if a testing strategy is adopted during the initial chip design.

Design for test has been employed to indicate those design techniques used to enhance chip testability. Most of the design for test techniques are attempts to enhance the observability and controllabilty of a circuit design. Three major types of design for test are: the ad hoc approaches, the structured approaches, and the built-in self test (BIST) approaches. The earliest design for testability employed ad-hoc guidelines in carrying out a design. Ad-hoc techniques are those techniques that can be applied to a given design but are not directed at solving the testability in general. Ad-hoc techniques have been an effective way in implementing testability without incurring significant area overhead. However, they have the disadvantage of depending heavily on the skill of the designer. Among those ad hoc techniques, the test point approach is the most popular and is being used in VLSI design to offer testability relief. This approach involves inserting test nodes to enhance both controllability and observabiltiy. Design for test has evolved in response to the increase in chip complexity. More recently, the proposed design for test are structured. The structured techniques, on the other hand, are attempts to solve the general test problem with a design methodology. The structured design for test requires a chip be designed by means of design rules. The structured design for test facilitates the use of complicated CAD tools for automated testability implementation and test

generation. The distinct advantage of the structured design for test over ad hoc guidlines is its ease of implementation. The BIST, which is not covered in this thesis, is an outgrowth of the structured design for test. The objective of BIST is to design chips so that they can test themselve if enough clock cycles are being applied. BIST has been used to self test chips with a structured array such as a RAM, where the linear feedback shift registers are used to automatically generate the address and data patterns for testing on chip RAM.

B. Scope of this thesis

The scope of this thesis is a review of commonly used design for testabiltiy techniques. Attributes of each technique are evaluated. An alternative design for test is investigated and the experimental results are reported. The concept of controllabilty and observability, testability measures and fault models are described in the thesis.

II. TESTABILITY MEASURES AND FAULT MODELS

A. Testability Measures

As the complexities of VLSI circuits increase, the need for designing them with testability is more important than ever. To facilitate this design objective, it is obvious to VLSI designers that there is a definite need to quantify the testability measure of digital circuits before test generation or verification actually is performed. This quantitative measure of testability also can be used to aid in designing more testable circuits.

In general, the testability measure is based on algebraic methods to calculate the controllability and observability of every node in the digital integrated circuits. The controllability is rephrased here as the ability to set a node to logic high and low; whereas the observability is the ability to propagate a fault on that node to an observable output. From these definitions, it is clear that the primary inputs and outputs of digital circuits are easiest to control and observe respectively. The more a node is embedded in a circuit, the more difficult it is to control or observe.

Stephenson and Grason [1] developed TMEAS, a testability program, for analyzing testability of digital circuits. In TMEAS, a circuit is considered a network of components interconnected links. The program calculates the controllability (CY) and the observabilty (OY) of each link. The overall testability value of a node is the geometric mean of CY and CO. Examples tend to show a good correlation between testability and actual test generation efforts [1]. TESTSCREEN is another testability analysis program [2]. In addition to the combinational controllability and observability similar to TMEAS, the sequential component test value of nodes is also reported. This sequential value indicates the number of test patterns needed to exercise the logic node.

However, the most publicized test analysis algorithm is SCOAP. SCOAP (Sandia Controllability and Observability Analysis Program) was originally developed by L. H. Goldstein

et. al. [3], and later modified for use in VLSI design at Bell Laboratories [4]. SCOAP is a topology analysis program that estimates the testability of digital circuits by calculating the effort of controlling and observing each node. A circuit is described to SCOAP as the interconnections of standard cells from a cell library. The program calculates six parameters that characterize the combinational and sequential testability properties of a circuit. However, the Bell Laboratories version of SCOAP does not consider the controllabilty and observability for sequential nodes because they have been found to produce no significant information [4] in determining testability measure of a digital circuit. Instead, three new parameters are introduced that are considered to be correlated with the difficulty of detecting stuck-at faults. The new parameters are: stuck-at zero testability (SA0), stuck-at one testability (SA1), and the sum testability (SUM) . The program also computes the figure of merit of testability for an entire circuit. This figure is very helpful if it can be correlated with number of test patterns needed to test the circuit under investigation.

Good testability information is extremely useful to designers especially if it is available early in the design process. The accurate testability results of internal nodes of a digital circuit could be used to identify the area of poor testability. Remedial actions can then be taken to improve testability. Testability algorithms can also be used as a computer aided design (CAD) tool to locate redundant logics, uncontrollable and unobservable nodes. Also, they can be used in locating long counters that requires large number of test patterns, or identifying sequential circuits that cannot be initialized very easily.

However, there are limitations associated with testability measures. Testability values are estimates from considerations of circuit topology alone [6], consequently the values are derived based on assumptions made by algorithms. V. D. Agrawal et. al. showed that SCOAP program produces different (CC0) values for two different realizations of the same inversion function. The amount of information degradation from assumptions will vary with the circuit

design and the way the values are interpreted. There is also poor correlation between testability values and the actual test generation effort. Testability measures produce their values on the basis of circuit structure; whereas the actual testing process depends on the ability and sophistication of the pattern generator as well as the circuit structure. Therefore, good engineering judgment is required in the interpretation of testability measures.

B. Stuck-at Fault model

A model of faults which has been used successfully throughout the industry is the stuck-at model. The stuck-at model assumes that in the presence of a fault a logic node is permanently fixed to either to a logic 1 or a logic 0. Basically, this model is based upon the assumption that a good circuit will have no faults. Therefore, an input pattern to a faulty logic gate can be considered a test for its stuck-at faults if the pattern can provoke a difference in the output response between the good logic gate and the faulty logic gate.

However, not all failures that occur can be modeled by the stuck-at model. It has been pointed out recently this classical model does not represents failures in CMOS devices. A number of faults in CMOS such as stuck-at open and stuck-at closed could change a combinational network into a sequential network. The stuck-at model is no longer effective in testing in all cases. Advanced fault simulator that can model digital integrated circuits both at logic gate and transistor level is available for fault simulation. In addition to the stuck-at faults, the simulator can also model stuck open and stuck closed faults.

III. DESIGN FOR TEST TECHNIQUES

A. Ad hoc techniques

Ad hoc techniques enhance testability for a given design and are not applicable to all designs. The first ad hoc technique ever reported is the partitioning approach [7]. As the name implied, the technique disconnects one portion of an integrated circuit from another portion of the circuit to make testing easier. Another approach that has been used by the microcomputer designers is the bus architecture technique [8]. This technique allows access to buses which go to many different modules on the computer board. The third technique which uses linear feedback shift registers is the signature analysis [9]. This technique requires some design rules at the board level, but is not aimed at the overall objective of enhance the ability to observe and control the state variables of a sequential circuit. The above techniques are more suited for board level design. The ad hoc technique which is normally used in custom-logic design is the test point approach. It is described in this thesis.

1. Test Point Technique Design for Test

The test point approach has been used to enhance the testability of digital circuits. The goal of this ad-hoc design for test is to make internal nodes deeply embedded in a circuit more controllable, more observable. A test point can be used to improve the controllabily when it is used as a primary input (Fig. 1a). To improve the observability, a test point can be used a primary output (Fig. 1 b). In some cases, a single test point can be used as both an input and output to achieve both control access and observation access (Fig. 1c). Test points are selected based on the designer's skill and knowledge of the circuit that he is designing. Experienced designers often have a clear idea where test points may be placed. Well placed test points will aid in detecting most faults thus high fault coverage can be achieved. Testability analysis algorithms, such as TMEAS or SCOAP mentioned previously, can be used in selecting test points.

One critical limitation of the test point technique involves the use of extra pins for test purpose. For the current VLSI technology, even a small number is likely to be expensive. However this drawback can be overcome if existing functional pins are utilized for test purpose with a multiplexing scheme. Instead of going directly to an extra output pin, the test data is multiplexed through the use of a data select. In the normal mode, the functional data goes to output pins. In the test mode, the data from the test point propagates to the existing output pins for observation. Similarly, the same scheme can be used to multiplex input data and test data to improve control access without using extra primary input pin.

B. Structured design for test techniques

1. Level Sensitive Scan Design (LSSD)

The most popular structured design technique for testability is the LSSD (Level Sensitive Scan Design) [10], currently in use at IBM. This technique enhances both controllability and observability of digital circuits by connecting all system latches into one or more shift registers. The registers are designed to operate in two distinct modes. In the normal mode, the circuit latches performs their design functions. In the test mode, all the system latches are reconfigured into serial shift registers. The testing of sequential circuitry is then achieved by shifting a sequence of known data into and out of the cascaded shift registers. The shifting process exercises the combinational circuitries, stores the results in the latches, and propagates the stores results out of the registers for evaluation. Since the data can be scanned in and out of the registers via primary inputs, the problems of fault detection in sequential networks then is reduced to the much easier task of detecting faults in combinational networks.

The two distinct attributes of the LSSD are "Scan" and "Level sensitive". Scan refers to the ability to shift into or out any state of the digital circuit. Level sensitive refers to the operation of sequential logic that is independent on rise time, fall time, and circuit delay. The second feature is realized with the use of level sentitive latches for circuit memory elements.

Figure 2a shows a shift register latch (SRL). The inputs D and C form the normal mode memory function while the inputs I, A and B of the functionally idle L2 latch makes up additional circuitry for the shift register function. The shift registers are formed by connecting L2 to input I as shown in Figure 2b. Scan in, A, B, and Scan out are extra pins needed to implement the LSSD technique. The clocking line A and B operate in a two-phase non-overlapping fashion.

There are several negative impacts associated with the LSSD technique. The SRLs in the shift registers are much more complex than single latches. Up to 4 additional input and ouput pins are required. However, the pin overhead can be reduced by making use of existing functional pins. Overall performance of the circuit may be degraded by the clocking requirement and by the routing connections of widely seperated latches. The area overhead has been reported up to 20%. In some optimized versions of LSSD [11], the L2 latch is modified for functional use along with the L1 latch thus reduces the significant overhead associated with the originally proposed LSSD .

2. Scan path design for test

The Scan Path technique has the same objective as the LSSD approach. However, the memory elements that are used to configure shift registers are D-type master slave flip-flops [12]. A multiplexer is added at the data input of the flip-flop (Fig. 3a) to permit the selection of two different data. The choice of data input depends on the logical value of the control input MODSW. In the normal mode, MODSW is held high to allow the functional data (D) to be gated into the flip-flop. In the test mode, MODSW is kept low and the test data is entered from Scan-In input.

Figure 3b shows the structure of a Scan Path design. With the system in the test mode, a test sequence is scanned into the shift registers via the Scan-In primary input. After the sequence has been loaded, the sytem is returned to the normal mode. The combinational

circuits act upon the flip flop contents and store results in the memory elements. By returning to the test mode again, the contents of flip-flops are scanned out to a primary output for evaluation. Unlike the LSSD, the same master and slave clocks are used in both normal and test modes. In a standard cell design, the slave clock can be easily generated locally from the master clock. Only one clock needs to be routed over the whole chip, thus eliminates the effort to reduce clock skew as required in a multiple clock design. Additionally, the delay induced by the data multiplexer in the data path can be alleviated if the multiplexer is implemented within the D flip-flop. The technique can be further optimized if the registers are kept at reasonable length. For an n-bit shift register, it will take n clock cycles to shift the first bit of a data stream into the last flip flop of the register. It is advantageous to keep all registers short to avoid handling large number of test sequences. In term of extra I/O pins, at most three pins are needed: mode control, scan-in, and scan-out. However functional inputs or output pins can be used as scan-in and scan-out signals for shift registers.

3. Scan Set Design for Test

This design technique for testability is similar to LSSD in a sense that shift registers are used to control and observe the state of internal latches. The distinct difference is that the shift registers are not in the circuit data path [13]. They are independent of all system latches. Being separate from the functional portion of the circuit, the scan/set design has small effect on the overall chip performance.

Figure 4 shows an example of Scan/Set design, referred to as bit-serial scan. The serial shift register, usually limited to 64 bits long, is designed with proper control signals to operate in two distinct modes: scan and set. In the first function, the internal nodes of the circuit to be scanned are parallel loaded into the bit-serial register with a single clock. Once the data bits are loaded, a shifting process will take place and the data will be scanned out through the scan output pin for observation. The reverse of the scan function is the set function. In this

mode, the test input pattern is serially shifted into the scan/set register and then gated into the system functional latches to set them to specific values.

The Scan/Set technique offers designers flexibility to select nodes to be sampled or set. The shift registers in the Scan/Set design can be used either to sample outputs from the functional logic to improve observability or to drive test data into the system latches to enhance controllabilty. Also the approach does not require the set function to set all system latches, nor the scan funtion scan all system latches. This design flexibility would allow designers to apply his skill and engineering judgment in selecting sample and set points to maximize testability while holding the overhead to a minimum. Normally, scan points should be assigned where they will do the most good in terms of aiding detection faults.

Another favorable attribute of this technique is that the scan function can occur during the system operation. The sampling pulse (load enable) to the shift register can be applied while the clock is being applied to the system sequential logic. States of the sequential machine can be obtained and off-loaded for observation with little effect on system performance.

4. Random Access Scan Design for Test

The Random Access Scan technique [14] enhances controllability and observability of internal system latches without employing shift registers. What is used in this design for testability approach is an addressing scheme that allows each storage element to be uniquely selected, so that it can be controlled or observed. The addressing structure is similar to that of a Random Access Memory.

The basic addressable latch configurations required for the Random Access Scan approach are shown in Figure 5a. This shows a single latch with extra inputs besides its functional inputs labeled Data and CK. The extra inputs are required only for testing purpose . The Scan Data In (SDI) is a scan data lead which is clocked into the latch by the SCK clock.

The SCK clock can only affect this addressable latch, if both the X-adr and Y-adr addresses are one. The Scan Data Out (SDO), which indicates the state of the latch, can be observed when both X-adr and Y-adr are pulled to a logic one. The observability mechanism is achieved by ANDing all SDO signals from all latches to produce chip scan out signal.

A circuit implemented with scan in/out network is shown in Figure 5b. Basically, there is a X-Y address decoder, the addressable latches, system clocks, and CLEAR function. There is also one logic gate necessary to create the preset function. An AND gate tree which is required to multiplex all latch state signals (SDO) is also shown. The Random Access Scan offers the controllabilty and observability of all system latches with fairly significant overhead. The overhead has been reported about three to four gates per storage element. The functionally idle address decoder and output AND gate tree are high price for testability. The number of address inputs and AND gates is proportional to the number of latches in the system. In terms of primary inputs and outputs, the overhead is between 10 to 20. Even though this number can be lowered by using the serial scan approach to implement the X and Y decoder, a small increase in pin count is still expensive for current VLSI technology. The technique can be used to a great advantage if the exsisting address decoder of an embedded memory is used to implement the addressing scheme for system latches.

IV. DESIGN FOR TEST IN ASICs

The Application Specific Integrated Circuits (ASICs) are characterized by a diversity of functions, rapid design changes and quick turnaround cycles. People in this emerging semiconductor business certainly agrees that Design for Testability is an important aspect of a whole design process. With the common belief that DFT offers high quality testing at reasonable cost, companies have adopted DFT as official design pratice of their products. IBM is an example with its LSSD.

However, the hardware overhead incurred by DFT especially by several structured approaches is unnecessarily high. The LSSD or Scan path design approach really pays off when applied at the board and system level . On per chip basis, the structured design is less attractive. The strongly competative nature of the ASICs business does not tolerate negative impacts that structured techniques have on area, yield and eventually cost. Therefore, it is not contradictory to suggest that the implementation of testability in ASICs should be carried out on discrete basis of individual design. Choice of DFT approach, ad-hoc or structured, for a given semicustom design should be made in considering the design structure, design schedule, and cost objective. By careful selection of DFT technique, the area overhead can be held to a minimum.

In considering the cost benefit ratio, the test point approach design for test is still popular and frequently used to cheaply achieve testability in ASICs. This ad-hoc approach requires a designer to have experience in logic design and switching theory. Nevertheless, the designer is well motivated to try harder if he is able to employ his skills, innovative techniques to achieve the design objective without resorting to using a structured approach design for testability. Furthermore, the demand in designer's skills is lessened with the availability of testability analysis programs. Automatic testability analysis can help the designer to strategically select test points without enduring the pain of manual analysis of the circuit. High number of

extra input and output pins associated with this approach can be often overcome by using existing functional pins and a mulptiplexing scheme.

There are similarities between LSSD and Scan Path design. Both techniques require digital circuits to be designed with a set of design rules, use shift registers to facilitate testing. The insignificant difference lies in the fact that LSSD uses level sensitive latches, whereas Scan Path technique employs edge-triggered D flip-flops for its bistable memory elements. It is unclear to see which technique has advantages over the other.

Since both technique require all memory elements to be connected into shift registers, computer algorithm is easier to develop to implement these design concepts into working CAD tools for design automation. Software is available for automatically implementing testability into a circuit using the Scan Path design after the functional design has been completed. When using this design tool, designers however have to identify and isolate existing functional shift registers and counters in the circuit so that not all flip flops in the registers and counters are unnessarily replaced by more complex and larger scannable flip flops.

In comparison with other structured design approaches, the area overhead incurred by LSSD or Scan Path is significantly high. An increase of 21% in cell area alone has been reported for an experimental Scan Path design [12]. Therefore, LSSD and Scan Path techniques should be considered for low production, poorly testable ASICs or those whose area is not as critically important as testing quality and design schedule.

The Scan/Set technique is the most flexible and has several interesting attributes among structured design for testability techniques. The shift registers which are independent from circuit latches are used to either sample outputs from the functional logic or drive data into selected number of circuit latches. Sample and set nodes can be limited to internal nodes which are difficult to control or observe. Careful selection of nodes which will aid most in fault detection, it is possible to enhance testability while keeping the area overhead to a

minimum. Designers have to rely on the experience in making good choice of internal test points to maximize testability. One drawback of this technique is that the registers are functionally idle. Since shift registers are not in the circuit data paths, there is little effect on the circuit performance. Therefore this technique should be considered for those ASICs whose speed requirement is stringent.

Finally, the Random Access Scan should be considered for ASICs with embedded memories where functional address decoders could be used to individually control or observe circuit latches. Otherwise, the extra decoder and AND gate tree required to simply implement testability make this technique less attractive. Despite there are many structured design for test techniques, they are not frequently used in ASICs. The reason is simple: the area increase is too high. Designers still use ad hoc techniques.

V. PARTIAL SCAN DESIGN FOR TEST

Ad-hoc technique design for test offers testability with reasonably low area overhead. The main disadvantage lies in long design time and designer's skill requirements. Structured design for test, on the other hand, enhances testability at the expense of excessively high area overhead, which has negative impact on chip yield, performance and eventually cost. High area overhead incurred by a structured approach is due to the redundancy effort in fault detection and global implementation of testability. In the Scan Path design, for example, all flip-flops in an integrated circuit are converted into bigger and more complex scannable flip-flops and then connnected into shift register for test purpose. Also in implementing testability, a structured design for test approach usually assumes that no faults have been detected. In a real design situation, a high percentage of faults are usually detected by functional test , which is essential in verifying every design intent. Therefore, the overhead can be minimized if the redundancy in fault detection is avoided and the testabiltiy is implemented to only detect those faults that have not been detected by functional test.

A. Methodology

A new technique called Partial Scan Design for test is investigated. The approach eliminates unnessary increase in area while still maintains high fault coverage normally achieved by structured design for testability. The reduction in area overhead is resulted from a systematic method in determining which flip flops need to be converted into scannable flip-flops and chained into shift registers.

Unlike the Scan path design, in the Partial Scan only about half of flip flops in a chip is modified for test purpose. The other half are systematically selected not to be connected into shift registers. The eliminating process is based on the following reasonings.

All inputs and outputs of a digital chip are highly controllable and observable. Therefore faults associated with input and output latches are highly probable of being detected by

functional test. Therefore, the output and input flip-flops can be excluded from scan chain without causing fault coverage degradation. Their exclusion from scan chain results in reduction of area overhead especially in digital chips with large number of I/O pins. Those flip-flops that have lower testability values relative to other flips-flops were carefully selected and dropped from scan chain until a given design objective was met. Potentially, some faults may not be covered because not all flip-flops are connected into shift registers, a list of undetected faults obtained from the functional fault simulation was used in aiding the selecting process. The remainder of flip-flops were then modified into scannable flip-flops and chained into shift register for testability. The conversion and vector generation were carried out using a design and test system available to the author. Vectors were generated to only detect faults that have not been covered thus reducing number of test sequences. A fault coverage figure was also obtained. This fault coverage indicates the percentage of faults that were additionally detected as the result of implementing testability.

B. Experimental Results

Two LSI chips were used to study the area saving and fault coverage offered by the Partial scan technique. On the average, the area overhead incurred by the proposed approach was only 13% compared to 22% incurred by a total scan technique. It was also found that no significant reduction in fault coverage.

The first LSI chip used in this study has 900 transistors and with a total of 21 flip-flops. TITUS [12] a program that automatically implements testability and generates test vectors was used as a CAD tool in this study. A 22% increase in cell area was obtained when all flip-flops in the chip were connected into a shift register for test purpose. However, a 13% increase in area was achieved when the testability was implemented using the proposed partial scan technique. Only eleven out of 21 flip-flops were included in the shift register. The other nine flip-flops were selectively eliminated from the shift register.

The second LSI chip used in this study has 4500 transistors and 117 flip-flops. With this level of complexity, the area saving as the result of using the proposed partial scan design was significant. A 25% increase in cell area when the total scan was tried. Whereas an increase of only 14% in area was achieved when 45 flip-flops were chosen not to be included in the shift register.

In the total Scan design, the area overhead increases as the chip complexity increases. In the partial Scan design, the area overhead can be controlled without fault coverage degradation by selectively or locally implementing testability. The ability to keep chip size as small as possible is very important. To have the advantages offered by the partial scan design, extra effort and time were required in running fault simulation and selecting flip-flops for testabiltiy. However, the testabiltiy implementation and vector generation can be automated via the use of available software. The crucial step in this design technique was to manually decide which flip-flops to include and which flip-flops to exclude from a shift register. Nevertheless, the partial Scan technique is a good temporary solution to the excessive area overhead incurred by structured design for test until a future testability system is developed , where the system will run functional fault similation, identify undetected faults, and then selectively implement testabiltiy to only detect faults that have not covered by the functional test. The designer controls the area overhead by supplying better and better functional test vectors to the system. Functional test that detects more faults will reduce the amount of test circuitry needed and thus keep area overhead low.

VI. CONCLUSION

As the chip complexity increases, design for test is vital especially in the area of semi-custom designs where a large number of different functions have to be produced. However, the large area overhead incurred by most structured design techniques is intorerable in most cases. To insure high quality testing and reduce area overhead, intelligent testabiltity systems should be developed. An ideal system would be able to implement testability to detect only faults that have not been covered by functional test. Only such a system will keep area overhead to an acceptable level. Bounded by his working environment, the semi-custom logic designer normally does not have the choice of selecting which design for test approach is best suited for his design. He has to decide whether or not to use the design for test software available to him to implement testability. The decision is usually made considering cost, performance and volume of a product. An area sensitive chip may make large area overhead incurred by a structured design for test intolerable. Due to many constraints, the designer more often resorts to the test point approach for implementing testability.

Finally, the partial Scan design technique studied in this work could be one alternative to any structured design for testabiltiy. Experimental results indicate that the proposed technique can be used to achieve high fault coverage testing without significant increase in chip area.

REFERENCES

1. J. E. Stephenson & J. Grason, "TMEAS, a Testability Measurement Program," 16th Design Automation Conf., June 1975, pp. 156-161.

2. P. G. Kovijanic, "Interactive Testability Analysis," 1979 Test conference, IEEE Pub. pp. 310-316, Oct. 1979.

3. L. H. Goldstein & E. L. Thigpen, "SCOAP, Sandia Controllabilty/ Observabiltiy Analysis Program," Proceedings of the 17th Design Automation Conference, Minneapolis, MN, June 1980, pp. 190-196.

4. D. M. Singer, "Friendly User Version of SCOAP Manual," Private Technical Memorandum, 1982.

5. V. D. Agrawal & M. R. Mercer, "Application for a Testability Measure to VLSI Design," Private Technical Memorandum, 1982.

6. V. D. Agrawal , M. R. Mercer, "Testability Measures - What do they tell us," Proceedings of 1982 IEEE Tets Conference , pp. 391-396.

7. S. B. Akers, "Partitioning for testability," J. Des. Automation Fault-Tolerant Computing., vol. 1, no. 2, Feb. 1977.

8. H. J. Nadig, "Signature analysis -concepts, examples, and guidelines," Hewlette-Parkard Journal, pp. 15-21, May,1977.

9. E. White, "Signature analysis, enhancing the serviceability of microprocessor-based industrial products," Proc. 4th IECI Annual Conf., IEEE Pub. 78CH1312-8, pp. 68-76, Mar. 1978.

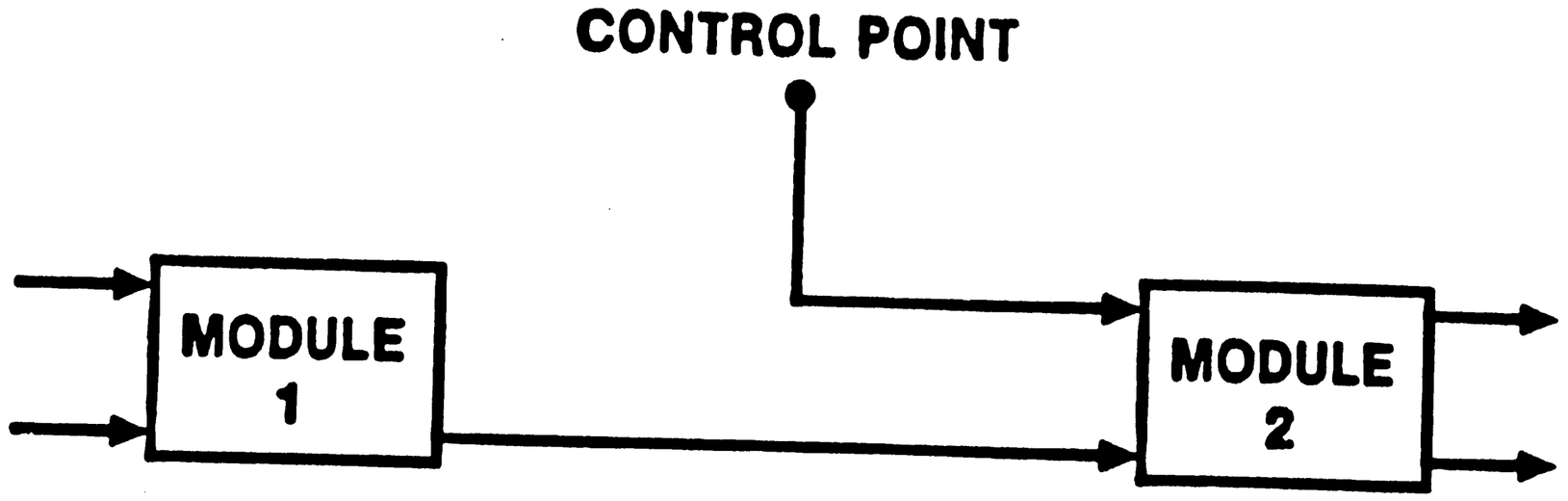
10. E. B. Eichelberger & T. W. Williams, "A Logic Design Structure for LSI Testabiltiy," 14th Design Automation Conference, 1977, New orleans, LA.

9 11. S. Dasgupta et al, "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability, and Serviceability," 11th Annual International Symposium on Fault-Tolerant Computing, 1981, Portland, ME.

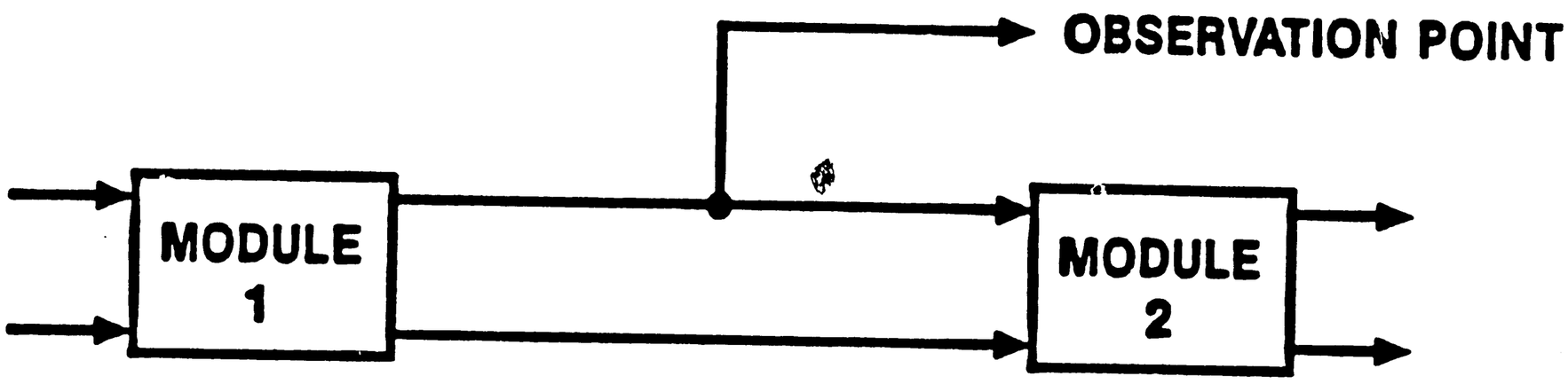
12. V. D. Agrawal et el, "A CAD system for Design for Testability," VLSI Design, Oct. 1984, pp. 46-54.

13. J. H. Stewart, "Future testing of large LSI circuits cards," Digest papers 1977 Semiconductor Test Symposium, IEEE Pub. 77CH1261-7C, pp. 6-17, Oct. 1977.

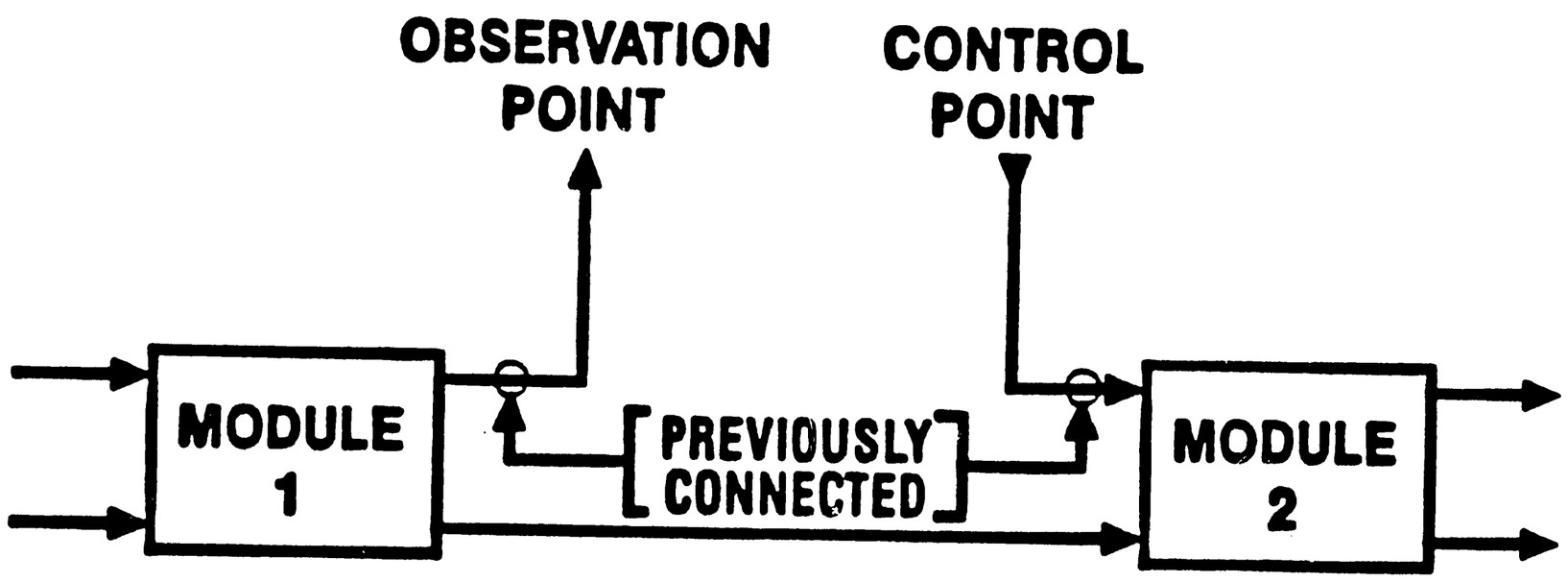
14. H. Ando, "Testing VLSI with Random Access Scan," Digest papers, Compcon 1980, IEEE Pub. 80CH11491-OC, pp. 50-52, Feb. 1980.



a. Test point to enhance controllability

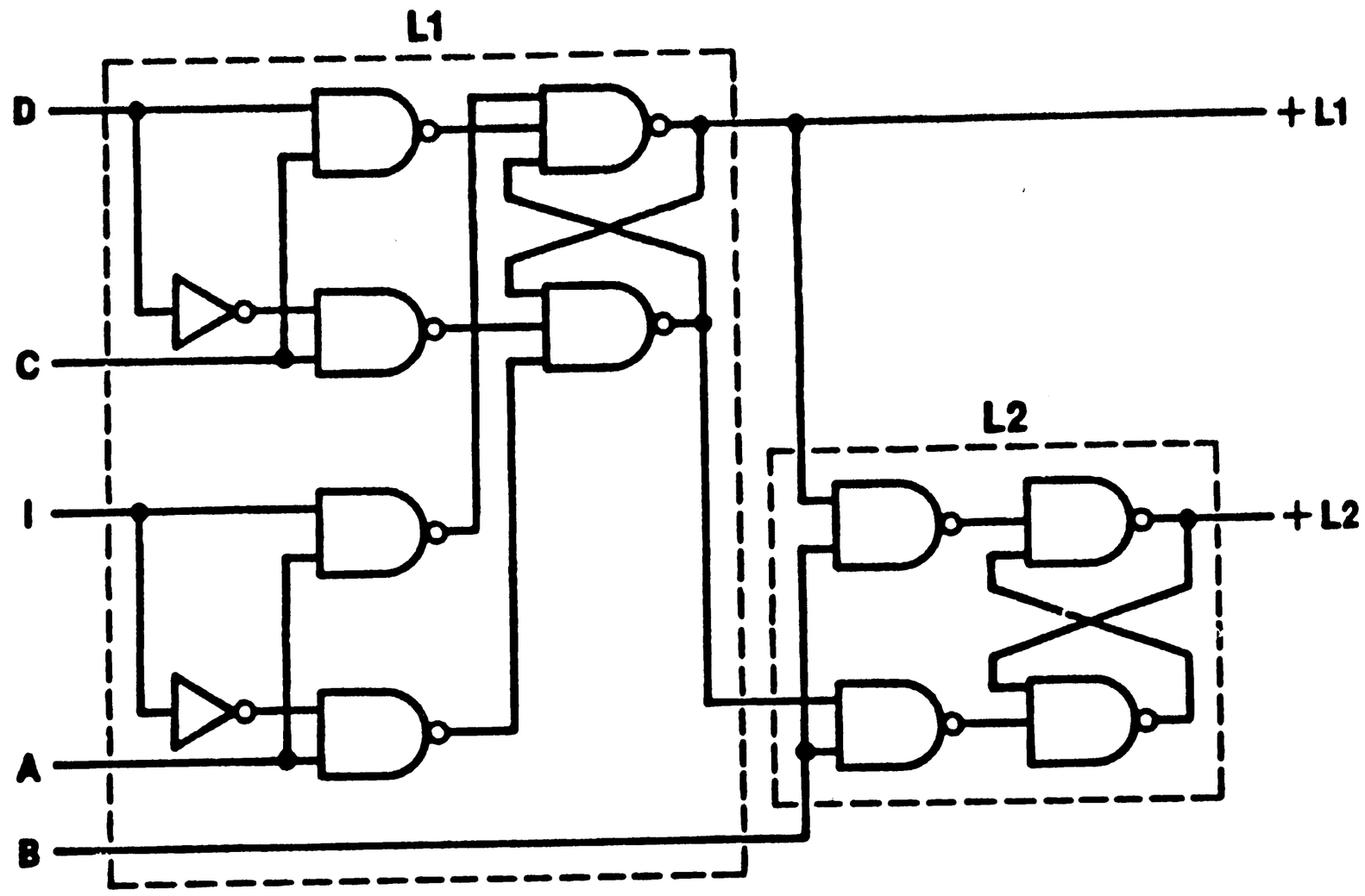


b. Test point to enhance observability

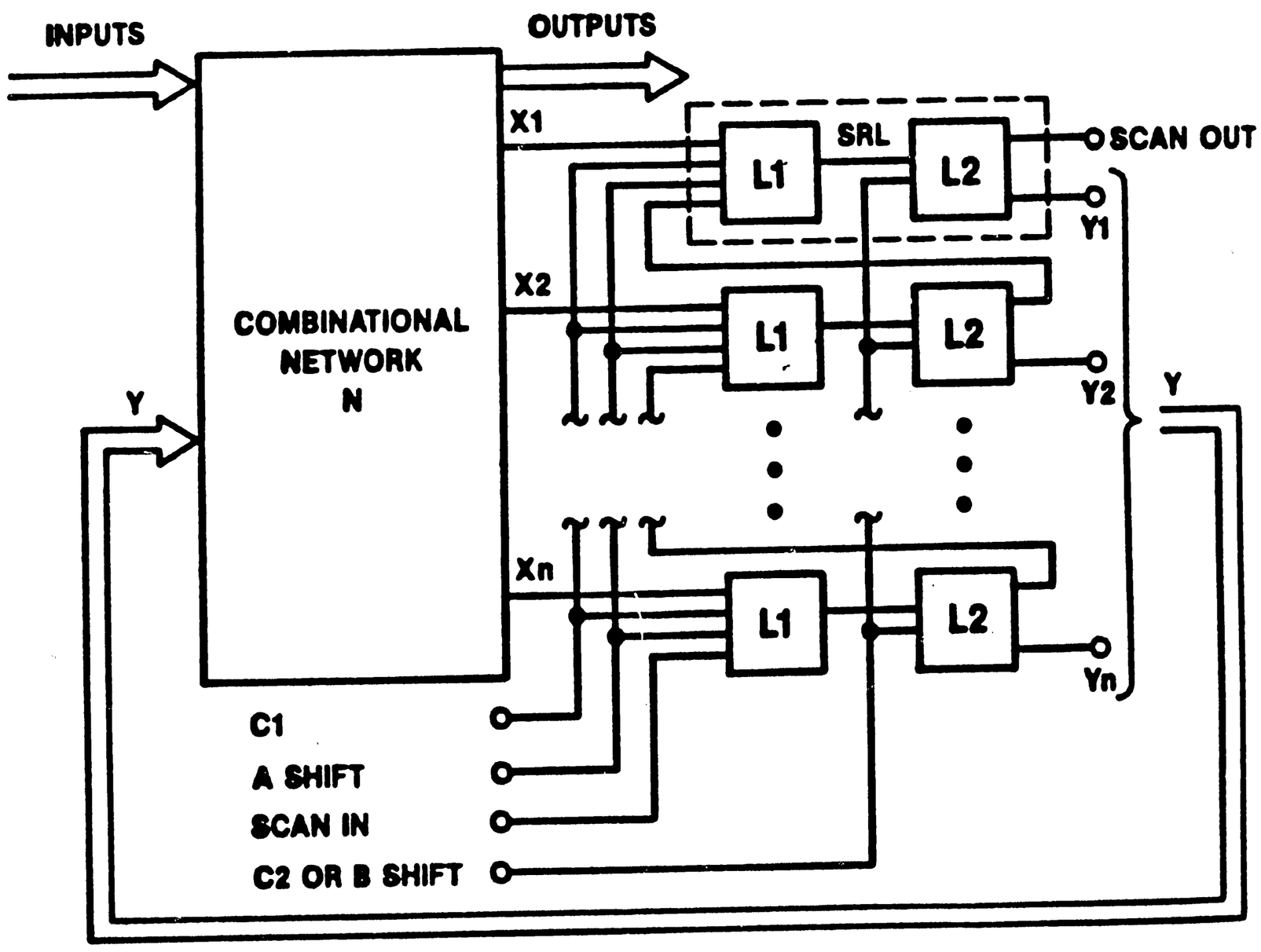


c. Test point to enhance both controllability and observability

Types of testpoints
Figure 1

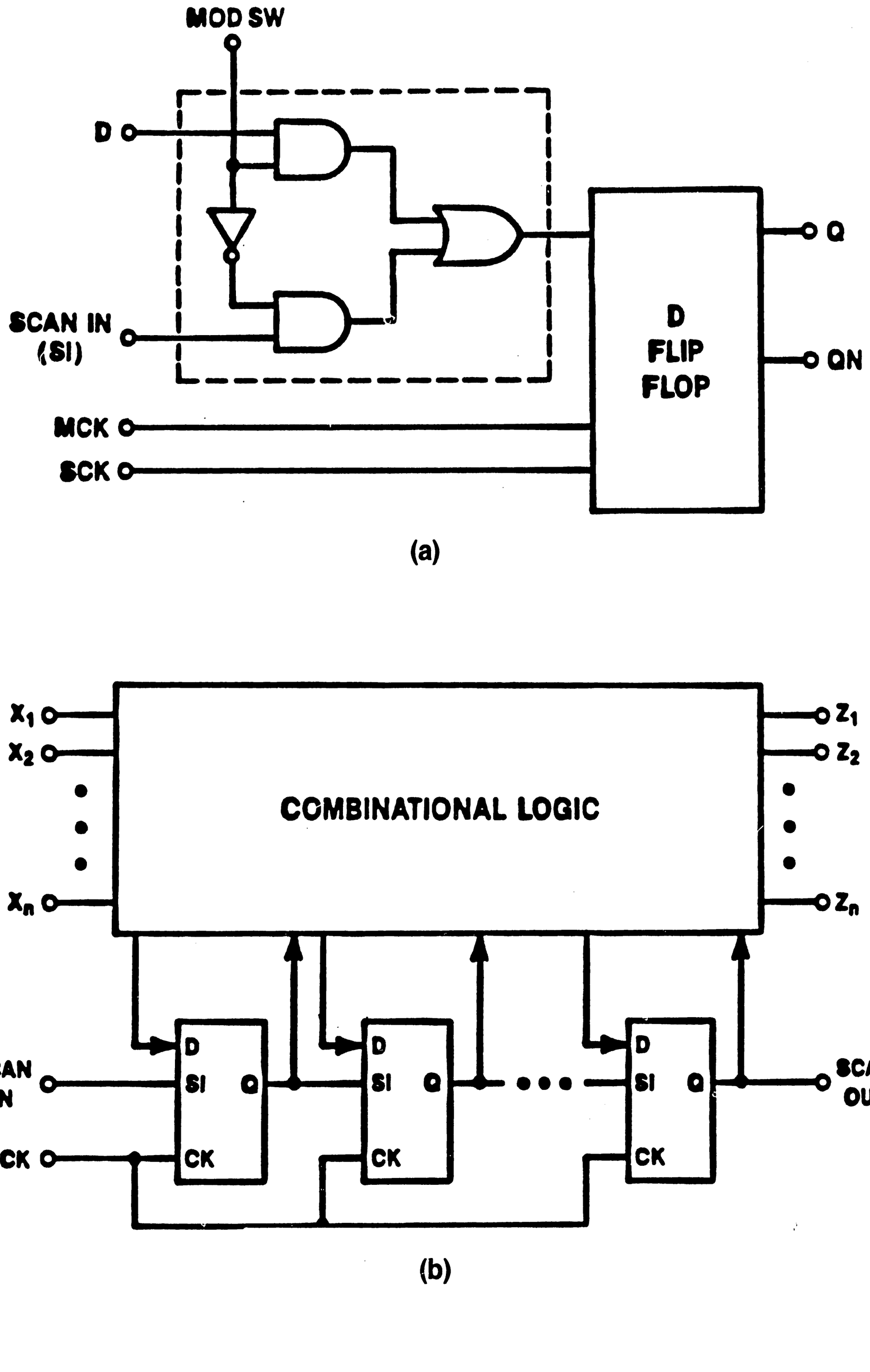


(a)

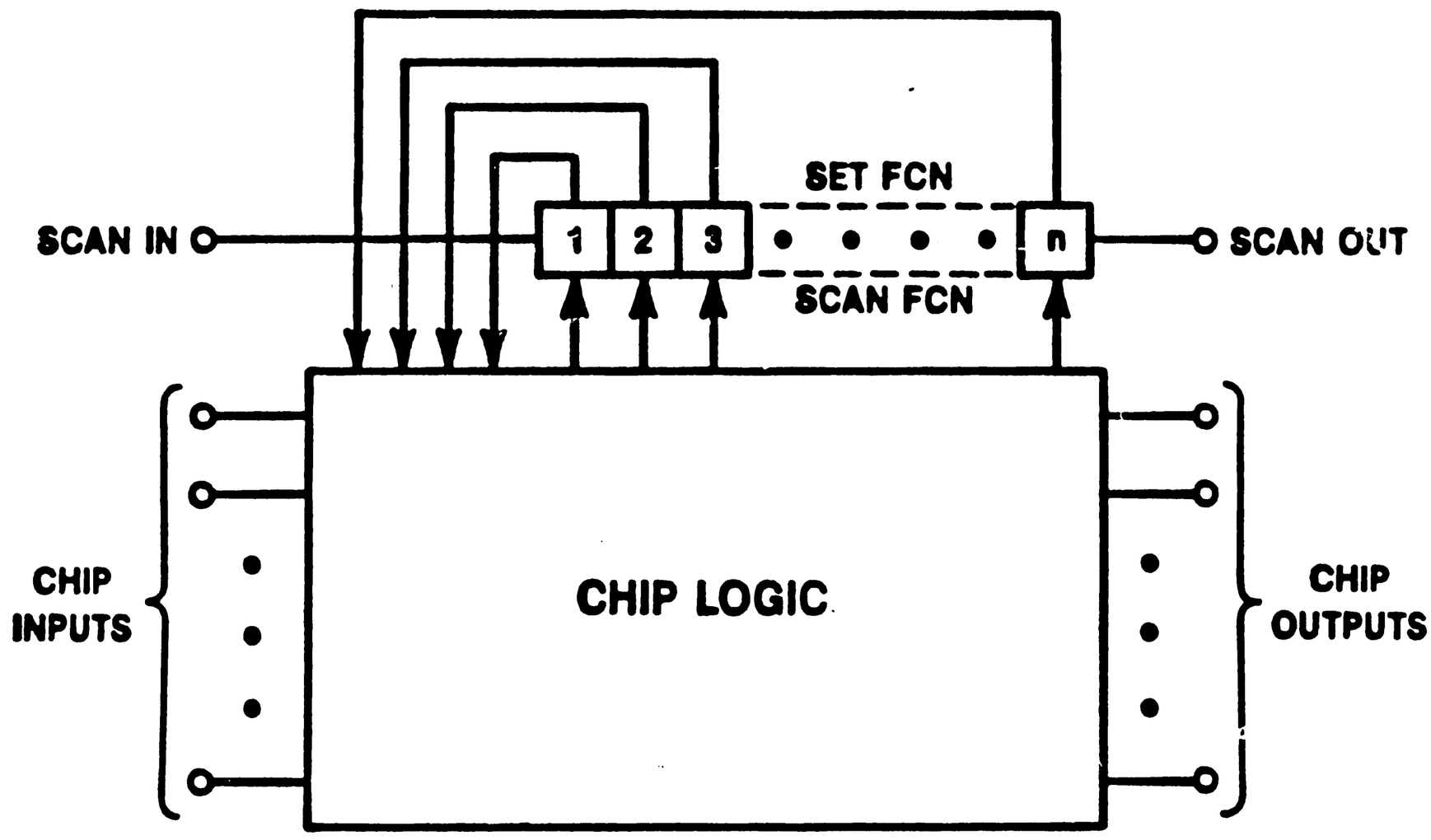


(b)

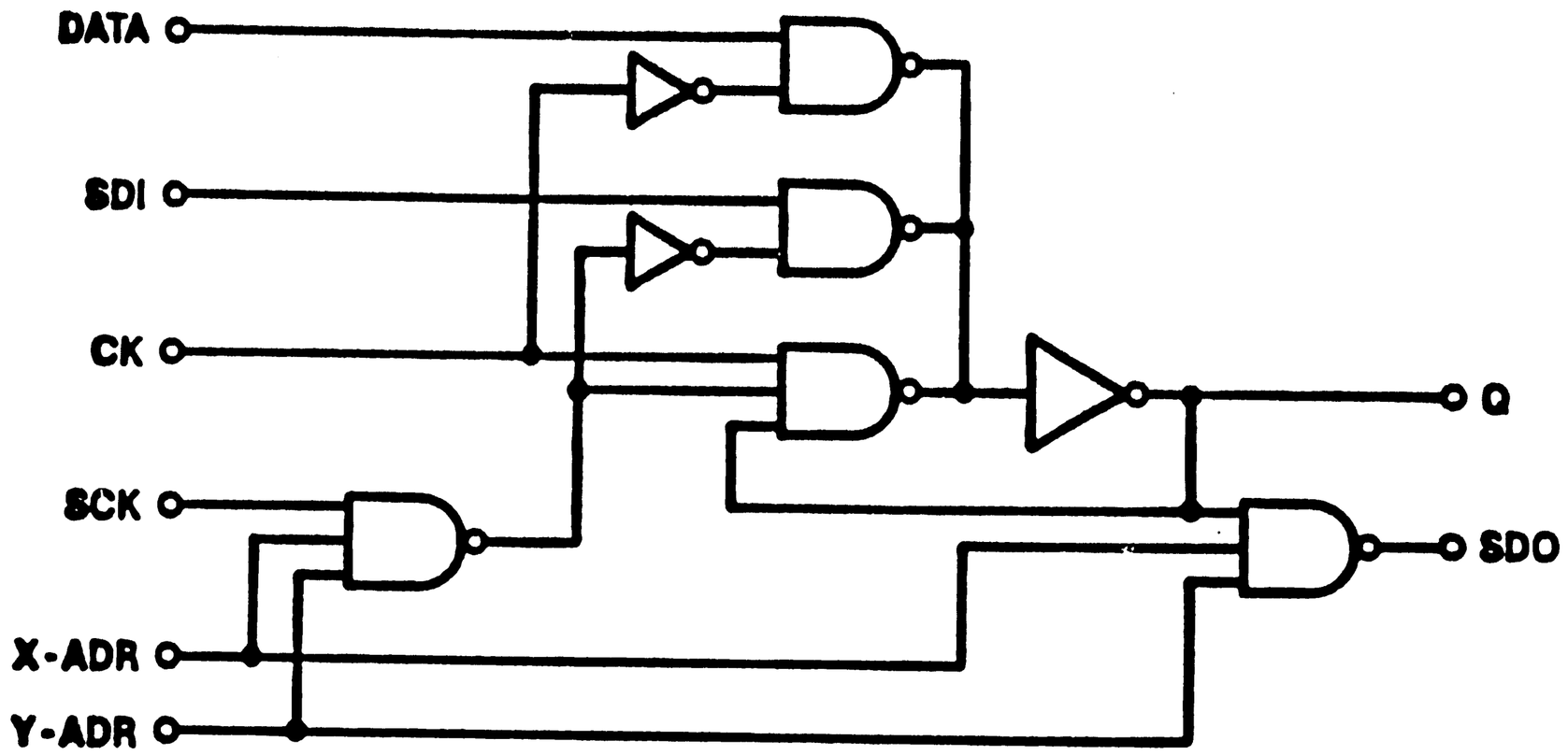
General structure of a LSSD network
Figure 2



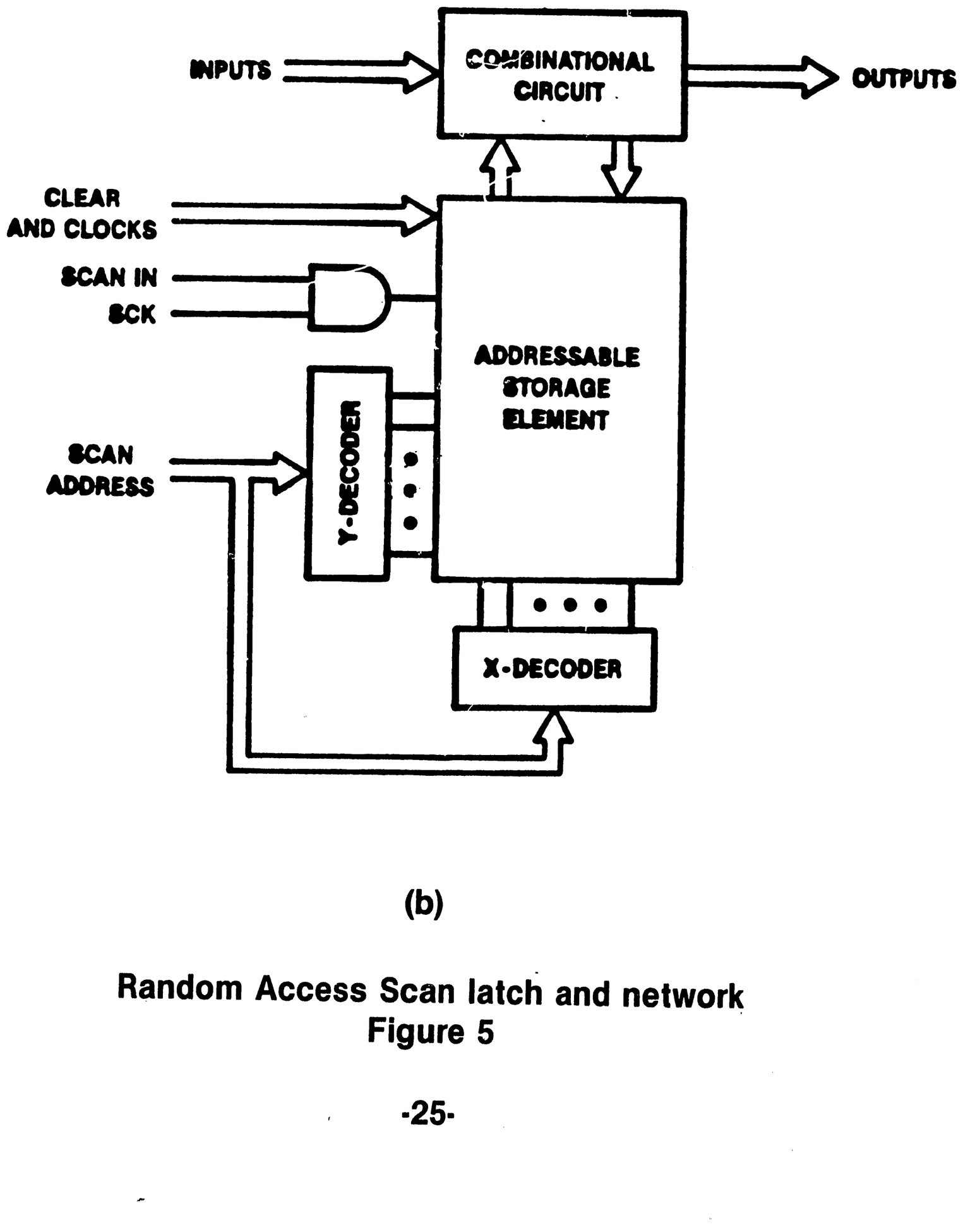
Cell and circuit implementation of scan design
Figure 3



General structure of Scan/Set design
Figure 4



(a)



(b)

Random Access Scan latch and network
Figure 5

VITA

CU TRONG THAN

Born October 10, 1950 in Vietnam. Immigrated to United States in August, 1975 and nationalized in June, 1982. Married to Phuoc Nguyen Than in June, 1974. Three children: Thuy (11), Lara (10) and Vi Than (7).

EDUCATION

Harrisburg Area Community College, Harrisburg, PA, Associate Degree in Electrical Engineering (Honor), 1979.

Pennsylvania State University (Capitol Campus), Middletown, PA, Bachelor of Science in Electrical Technology (Honor), 1980.

EXPERIENCE

Employed by AT&T Bell Laboratories of Allentown, PA., since 1981 as a Senior Technical Associate. Promoted to Member of Technical Staff in July, 1985. Past assignments have been related to the layout and physical design of Very Large Scale Integrated (VLSI) circuits. Present assignment is the logic design of a Asynchronous Receiver/Transmitter Interface.