DESIGNING ASYNCHRONOUS SEQUENTIAL CIRCUITS FOR RANDOM PATTERN TESTABILITY

O. A. Petlin, S. B. Furber

(Department of Computer Science, The University, Manchester, M13 9PL, UK)

oleg@cs.man.ac.uk, sfurber@cs.man.ac.uk

A. M. Romankevich, V. V. Groll

(Department of Special Purpose Computers, Kiev Polytechnic, 252056, Ukraine)

Abstract

A resurgence of interest in asynchronous VLSI circuits is occurring because of their potential for low power consumption, design flexibility and the absence of the clock skew problem. In this paper, an approach to the design of asynchronous sequential circuits for random pattern testability based on the micropipeline design style is described. The test procedure for such asynchronous sequential circuits provides for the separate testing of the combinational logic block and the memory elements. The total number of random test patterns required to detect all the stuck-at faults in the data processing blocks and control blocks is determined by the total number of tests for the combinational logic block. A case study of a register destination decoder designed for random pattern testability is presented to demonstrate the practicability of the proposed design approach.

Keywords and phrases: asynchronous circuits, very large scale integration (VLSI), micropipelines, pseudo-random testing, random testing, random pattern testability.

1. Introduction

Asynchronous VLSI circuits have become a subject of intensive research due to important advantages compared to their synchronous counterparts such as the absence of the clock skew problem, the potential for lower power consumption and design flexibility [1]. There are several different approaches to designing asynchronous circuits. The main differences between them can be characterized in terms of their data representation and data processing methods [2,3].

One of the most promising approaches to the design of complex asynchronous VLSI circuits of great complexity is the micropipeline approach, described by Ivan Sutherland [4]. In the micropipeline approach, the delays of all the logic elements and wires are finite and subject to certain constraints. The data is represented in binary form and treated as a bundle (see Figure 1). When the data is ready the sender generates a request signal. After the data has been processed by the receiver it generates an acknowledge signal. Then the sender can produce new data for the receiver.

Using the micropipeline approach, an asynchronous version of the ARM6 processor has been designed by the AMULET research group in the Department of Computer Science at the University of Manchester and fabricated by GEC Plessey Semiconductors Ltd. [5,6]. The general structure of the asynchronous ARM processor is a large micropipeline in which streams of input data, addresses and output data are processed. All the computational blocks of the asynchronous ARM are also designed as micropipelines. The bundled data interface of the asynchronous ARM employs the two-phase transition signalling technique shown in Figure 2. In this technique, every transition (falling or rising) on a control line is considered as an event.

In production, a VLSI circuit must be tested for fabrication faults to ensure that it works correctly. Many methods and techniques for the derivation of tests for digital circuits have been proposed [7,8]. These test generation methods can be divided into two main groups: those which use algorithms and those which use random or, more correctly, pseudo-random techniques to derive tests. All the algorithmic approaches to generating tests assume that there is an algorithm which will yield all the necessary tests to detect faults from a certain class. Random testing of VLSI circuits is an alternative to algorithmic test generation methods. It is becoming increasingly attractive for test engineers because of the following advantages [8,9]:

- test patterns may be generated simply using pseudo-random pattern generators (PRPGs) which can also be used in built-in self test VLSI systems;
- pseudo-random testing does not depend on the particular properties of a test object except that it must have no disallowed input vector values.

An attempt to implement boundary-scan and pseudo-random BIST in an asynchronous transfer mode switch has been made by Thorel P. et al. [10]. The proposed approach detects stuck-open and simultaneous reading and writing faults in a Double Access RAM with a fault coverage of 99.9%. During the test, all addresses and input stimuli are produced by the PRPG and the responses are collected into a signature analyser. Unfortunately, these results are restricted to a particular class of test objects.

A scan test technique for asynchronous sequential logic circuits synthesized from either a Huffman model or a signal transition graph has been reported [11]. The proposed scan test procedure provides for the detection of stuck-at faults in the asynchronous sequential circuit, reducing the test generation problem to one of just testing the combinational circuit.

The realisation of a partial scan test for asynchronous circuits has been demonstrated using the example of a DCC error corrector [12]. An analysis of the proposed scan test approach shows a high stuck-at fault coverage and a small area overhead. However, these scan test techniques have not been developed for testing asynchronous sequential circuits based on the micropipeline design style.

The aim of this paper is to present a method for designing asynchronous sequential circuits for random pattern testability. The proposed method is based upon the micropipeline design style and the two-phase signalling protocol.

2. The general structure of an asynchronous sequential circuit

Figure 3 shows the general structure of an asynchronous sequential circuit which uses a bundled data convention. This structure contains the combinational logic block (CLB) which performs the basic logic operations, and two registers (Reg1 and Reg2) in the feedback loop which store the state of the sequential circuit. The asynchronous sequential circuit works as a micropipeline. In the initial state, all the latches of Reg1 are set to their initial states and both the C-elements are set to zero. The input data is generated on the primary inputs (PI) of the circuit by the sender which sends a request signal (Rin) to the sequential circuit. The request signal is delayed by the delay element for long enough for the output data to stabilize on the primary (PO) and internal (SO) outputs of the combinational circuit. As a result, a request signal (Rout) is produced for the receiver by the sequential circuit. After receiving an acknowledge signal (Aout) and storing a new state in Reg2 the circuit generates an acknowledge signal (Ain) for the sender simultaneously causing the copying of the content of Reg2 into Reg1. When a new request signal is sent by the sender the procedure of processing the data is repeated.

The process of latching and storing the data in the state registers of an asynchronous sequential circuit is usually controlled by a pair of transition signals such as ‘pass’ and ‘capture’ [4]. Such latches are called transition latches. The register latch control of the circuit shown in Figure 3 is implemented using an internal conversion of the two-phase transition signalling into four-phase signalling allowing the use of level-sensitive latches [6].

3. Testing micropipelines

Stuck-at faults in micropipelines can be detected in various ways [13]. It was observed that stuck-at faults on the inputs and outputs of the C-elements and the request and acknowledge lines of the micropipeline are detected easily since they cause a faulty micropipeline to halt. The detection of stuck-at faults in the processing logic, which is assumed to be a combinational circuit, can be achieved by applying tests derived using any known test generation technique [7,8]. Another type of stuck-at fault has been identified in the micropipeline latches. These stuck-at faults can put a micropipeline latch permanently in capture mode (stuck-at- capture faults) or pass mode (stuck-at-pass faults). Any stuck-at fault on the inputs or outputs of the stage register or stuck-at-capture fault of the transition latch is equivalent to the corresponding stuck-at fault in the combinational logic block. To detect a stuck-at-pass fault in the transition latch two test patterns are required. In this paper, we will consider only stuck-at faults on the inputs and outputs of the state registers of an asynchronous sequential circuit.

A scan test technique has been developed by Khoche and Brunvand for the testing of stuck-at and delay faults in micropipelines [14]. The micropipeline can perform in two modes: normal operation and scan test mode. The micropipeline performs to its specification in normal operation mode. In test mode, all the latches are configured into one shift register where each latch works as an ordinary master-slave flip-flop. The stage registers of the micropipeline are clocked through the control lines where the input Aout is used as a clock input. The C-elements pass their negated inputs onto their outputs forming a clocking line for the scan path. As a result, the test patterns are loaded from the scan-in input into all the latches of the micropipeline. Then the micropipeline is returned to normal operation mode and one request signal is generated. To observe the contents of the register latches the micropipeline is set again to scan test mode. The contents of all the latches are shifted out to the scan-out output. The test technique described allows the detection of all the stuck-at and delay faults in the micropipeline.

Another method to design and test asynchronous sequential circuits based on the micropipeline design style has been reported [15]. The proposed scan test approach is implemented using specially designed scan latches controlled by the scan test control logic.

The main drawback of the scan test techniques mentioned above is the need to shift an *n*-bit test pattern into the scan register before it can be applied to the inputs of the test object. This reduces the test performance of BIST structures where the application of a large number of pseudo-random test patterns generated by internal PRPGs is presumed. The proposed solution is to implement the scan testing of asynchronous sequential circuits by shifting a random test pattern bit serially with the concurrent observation of the test results.

4. Design of asynchronous sequential circuits for random pattern testability

The general idea of alleviating the test problem of an asynchronous sequential circuit (Figure 3) is common to all sequential circuits, i.e. during the test the whole sequential circuit must be divided into a combinational part and memory elements which are tested separately. Figure 4 illustrates the design of a testable sequential circuit. This circuit contains some additional elements such as a register (Reg3) for collecting the test data from the internal outputs of the combinational circuit, a block of XOR gates to mix the test data and the multiplexer to switch the data flow during the test phase. Also there are two XOR gates, multiplexers and a toggle element to provide the proper control signalling.

The sequential circuit works in two modes (normal operation and test mode) which are set by switching the Boolean signal on the operation mode input (OM). There are two additional pins, Sin and Sout, inserted in the schematic to scan test patterns into Reg1 for stimulating the internal inputs of the combinational circuit and scan its responses out during the test.

Test mode.

In test mode, a logical zero is set on the operation mode control input (OM=0). The control part of the circuit shown in Figure 4 is reconfigured to provide the desired asynchronous test control interface. Initially, all the latches of Reg1 are set to their initial states, all the latches of Reg3, all the C-elements and the toggle element are set to zero. The primary inputs (PI) of the combinational circuit and input Sin of the sequential circuit are coupled to the outputs of the asynchronous pseudo-random generator (APRPG) [16]. The responses from the primary outputs of the combinational circuit and output Sout of the sequential circuit are compressed by the asynchronous signature analyser (ASA) [16].

A request signal (Rin) from the APRPG is delayed for long enough for the output data to stabilize on the outputs of the combinational logic block. The data from the internal outputs of the combinational circuit is mixed with the output data of Reg3 in the block of two-input

XOR gates. The outputs of the XOR gates come through the multiplexer to the inputs of Reg2 and are latched in Reg2. After receiving an acknowledge signal from Reg2, which is steered by the toggle element, the content of Reg2 is copied into Reg3. When the data is captured by Reg3 it generates an acknowledge signal on its output Ack. This signal causes the multiplexer to connect the first (*n-1*) most significant bits of Reg1 and the scan-in input of the circuit under test to the inputs of Reg2. Simultaneously, a request signal is produced for the ASA on output Rout of the circuit. The data from the outputs of the multiplexer is captured by Reg2 when a new request signal appears on its input Rq (in fact, this is the acknowledge signal for Reg3 which is delayed until the multiplexer has finished switching). A new acknowledge signal from Reg2 is steered by the toggle element and passes to the corresponding input of the C-element where it waits for an acknowledge signal from the ASA. The primary outputs of the combinational circuit and the scan-out output of Reg3, which is actually the *n*th bit of Reg3, are collected by the ASA. Once an acknowledge signal is received on input Aout of the circuit under test:

1) the content of Reg2 is copied into Reg1;

2) an acknowledge signal is sent to the APRPG.

When the APRPG has finished producing a new test pattern a new request signal is generated on input Rin of the circuit under test and the test procedure is repeated again.

Figure 5 shows the mechanism for applying random test patterns to the inputs and compressing the responses from the outputs of the combinational circuit. The procedure for applying test patterns (Figure 5a) assumes that random test patterns are applied to the primary inputs of the combinational circuit and the scan-in input of the sequential circuit. During the test registers Reg1 and Reg2 are configured to shift a new test bit to Reg1 after receiving a request signal from the APRPG. The process of collecting and compressing test data from the outputs of the combinational circuit (Figure 5b) consists of two parts. The first one includes the direct analysis of the responses from the primary outputs of combinational circuit by means of the ASA. The second part is a signature analyser which compresses the responses from the internal outputs of the combinational circuit. Registers Reg2, Reg3 and the block of XOR gates are configured in such way that the current contents of Reg3 are mixed (with the help of the XOR operation) with a new response which is produced on the internal outputs of the combinational circuit. The contents of Reg3 are observed on its *n*th output.

The signature analyser used for collecting the test data from the internal outputs of the combinational circuit is illustrated in Figure 6. The general structure of this signature analyser (Figure 6a) is similar to the well known structure of the BILBO signature analyser [7,8]. The equivalent schematic of such a signature analyser (Figure 6b) shows that the procedure for compressing the test data from the internal outputs of the combinational circuit is similar to the XOR operation. After receiving each request signal (r) the input bits are delayed for a different number of steps (request signals) depending on their position numbers and then XORed.

Normal operation mode.

In normal operation mode, input OM of the sequential circuit is set to one. The outputs of the toggle element and the outputs of Reg3 are held at zero permanently. Initially, all the latches of Reg1 are set to their initial states and all the C-elements are set to zero. After receiving a request signal (Rin) from the sender data is processed by the sequential circuit in the same way as was described for the circuit shown in Figure 3.

5. Analysis of the structure of the testable asynchronous sequential circuit

5.1 Advantages

The random pattern testable sequential circuit shown in Figure 4 has some important features which simplify its random testing.

Complexity of the test procedure.

During the test the combinational part of the circuit is tested separately from the memory elements which makes the testing of the circuit much easier. The random test procedure is implemented asynchronously allowing the use of all the advantages of the asynchronous bundle data processing technique.

Test performance.

Compared with the scan test methods known so far the test procedure does not require a test pattern to be scanned into the shift register before the test and the test data to be scanned out after the application of this test. During the random testing of the circuit (Figure 5a), test patterns are produced on the internal inputs of the combinational circuit with the help of a one bit shift of the content of Reg1. A new test bit is loaded from the APRPG after receiving a request signal. The test data from the internal outputs of the combinational circuit is collected in Reg3 after each new test pattern is applied to the inputs of the combinational circuit (Figure 5b). There is no need to shift all the contents out of Reg3 after applying a new test pattern to the inputs of the circuit (the test data is compressed and stored into register Reg3 and observed on its *n*th output after the application of each test pattern). In this case the random pattern testing of such a circuit is approximately (*n-1*) times faster then a traditional scan test method, where *n* is the number of latches of Reg1.

Number of random test patterns.

There are two important characteristics of random testing: the number of patterns which must be produced by the test pattern generator to produce the desired set of test vectors, and the probability of detecting all possible faults from the predetermined class of the circuit's faults. The first parameter reflects the practical usability of random testing or simply the random pattern testability of the circuit. The second parameter is a characteristic of the quality of random testing.

The analysis of the circuit illustrated in Figure 4 shows that the number of random test patterns required to detect all the single stuck-at faults in it is equal to the number of test patterns for detecting all these stuck-at faults in the combinational part of the circuit under test. This is because of the following properties of the circuit:

- all the stuck-at faults on the inputs/outputs of registers Reg1 and Reg2 are equivalent to the appropriate faults on the internal inputs of the combinational logic block [13];
- all the stuck-at faults on the inputs/outputs of the block of XOR gates and Reg3 are detected easily during the test of the combinational circuit (the circuitry which collects the test data from the internal outputs of the combinational circuit (Figure 6b) is similar to the BILBO register [7,8]);
- stuck-at faults on the control lines involved in the control of the random testing of the circuit are detectable since either they cause a deadlock of the asynchronous sequential circuit or they change the data flow during the test which can be identified easily [13].

Suppose that to detect all the stuck-at faults from the predetermined class of the combinational network's faults it is necessary to generate on its N inputs (N=n+m, where n and m are the number of internal and primary inputs to the combinational circuit respectively) the set, Q_k, of k ($k \le 2^N$) test patterns. The test confidence probability threshold, p_t, is the probability that all necessary test patterns from Q_k will be applied to the inputs of the circuit under test. The escape probability threshold of the test, $q_t = 1 - p_t$, is the probability that at least one pattern from the set Q_k will not be applied to the inputs of the combinational logic block during the test. The total number, T_k, of equiprobable test patterns applied to the inputs of the combinational circuit can be estimated using the following formula [16]:

$$T_k \ge 2^N \log (k/q_t) \ . \tag{1}$$

All possible binary vectors ($k = 2^N$) must be applied to the inputs of the combinational circuit to test it exhaustively. As a result, all possible faults which do not transform a faulty circuit into a sequential one will be detected with probability p_t.

Random testing using only equiprobable random test patterns is not always the optimal test procedure for obtaining the minimal (or close to minimal) number of random test patterns in order to guarantee that all the tests from Q_k will appear on N inputs of the combinational circuit with probability p_t. To reduce the number of random test patterns, special methods were derived for achieving optimal output signal probabilities for generators of weighted pseudo-random test patterns [17,18]. The algorithm for calculating the lower bound for the number of test patterns for random testing by means of applying weighted test patterns can be described as the following sequence of steps [16]:

1) compute the minimal probability of a test pattern from set Q_k, p_{min};

2) calculate how many test patterns, g, have a probability of appearance no larger than twice the minimal probability p_{min};

3) estimate the number of random patterns as

$$T_k \geq p_{min}^{-1} \log (g / q_t) \ . \qquad (2)$$

There are some stuck-at faults (on the true inputs of the multiplexers) which cannot be detected in test mode. They will manifest themselves during normal operation mode by preventing activities on outputs Rout and Ain, hence causing the whole circuit to deadlock.

5.2 Penalties

Hardware redundancy.

The hardware redundancy of the random pattern testable asynchronous sequential circuit is comprised of register Reg3, the block of *n* XOR gates, two XOR gates, three multiplexers, one delay element and the toggle element. Clearly, the overall hardware redundancy of the testable circuit heavily depends on the complexity of the combinational logic block: the more complex the combinational circuit is, the less redundancy the testable sequential circuit has. As a result, this method of designing testable asynchronous sequential circuits is more effective in terms of hardware redundancy for complex sequential circuits.

Performance degradation.

There is some degradation in the performance of the testable sequential circuit during normal operation mode. This is caused by the additional circuits in the data paths which inevitably insert additional delays into these data paths. These additional delays must be taken into account during the design verification of the asynchronous circuit.

6. Experimental results

The asynchronous version of ARM has a circuit called the “register destination decoder” which can be designed using the technique described above [6].

Figure 7 shows a testable implementation of the register destination decoder. During *normal operation mode* (OM=1), the input data to this circuit is a 16-bit binary vector (I[15:0]) containing coded information about the availability of registers in the register bank. For instance, a one in the 5th position of the input vector means that the 5th register must be processed. The output from the register destination decoder includes:

1) the four-bit address of the least significant ‘one’ in the input vector (RD[3:0]);

2) an active high output (R15) which indicates that the output address is ‘15’;

3) an active low output (NTRM) which indicates that the output register address contains the address of the most significant ‘one’ in the input vector.

After receiving a request signal (Rin) from the instruction pipe the input data is stored in the RS-flip-flops of the input register (RSReg). This data is processed by the priority encoder (PenC) and then stored into the latches of the output register (RdGenLat). When the output data is available the register destination decoder generates a request signal (Rout) to the primary decoder. Concurrently, the output address is decoded by the address decoder (PdGenDec) and sent to the input register through the block of sixteen AND gates. As a result, the least significant bit of the input register previously set to a logical one is cleared. The modified vector is again sent to the priority encoder. The address of the least significant one is stored in the output register when an acknowledge signal is received on the control input (Aout) of the register destination decoder. The procedure described above is repeated until the input register is set to all zeros. Then the register destination decoder produces an acknowledge signal (Ain) which indicates that it is ready to process another input vector.

Test mode.

During random testing (OM=0), the test data from the outputs of the output register (RdGenLat) and the 15th output bit of the signature register (SgnLat) are collected by the external ASA. The stimuli produced by the external APRPG are applied to the sixteen inputs of the register destination decoder. During the test the priority encoder generates all possible address vectors on its outputs. The address decoder is tested exhaustively if all possible binary vectors are applied to its inputs. The responses from the priority encoder are stored into the output register and analysed in the external ASA whereas the responses from the address decoder are collected in the signature register. As a result, the outputs of the priority decoder can be used as test patterns for exhaustive testing of the address decoder. Clearly, if the priority encoder is tested exhaustively the address decoder will be tested exhaustively as well.

In the initial state, all the memory elements are set to zero. A test pattern is applied to the inputs of the register destination decoder and stored in the input register after receiving a request signal (Rin) from the APRPG. Its output data is loaded into the output register and a request signal (Rout) for the ASA is generated. Simultaneously, the data in the output register is latched and the register destination decoder produces an acknowledge signal (Ain) for the APRPG. All the RS flip-flops of the input register are set to ones by a clock produced by the clock generator (Cgen). The responses from the outputs of the address decoder are mixed with the content of the signature register and loaded into the input register. When an acknowledge signal comes from the ASA the contents of the input register are copied into the signature register as shown in Figure 6. The input register is set to zero. After receiving a new request signal from the APRPG the test procedure is repeated again.

A model of the testable register destination decoder (Figure 7) has been created using *View-Logic* CAD tools. Test patterns for testing the register destination decoder have been generated with the help of a program model of the APRPG which is able to produce pseudo-random sequences of any length with any desired probability of a one [16].

The structure of the testable register destination decoder has been simulated with the presence of a single stuck-at fault. Roughly speaking all the stuck-at faults of the testable structure can be divided into two groups:

1) stuck-at faults in the data paths and

2) stuck-at faults in the control logic blocks.

Clearly, stuck-at faults on all the control lines involved into the communication process will be detected easily since they manifest themselves by a deadlock during the test. All the stuck-at faults in the data processing blocks can be detected by means of applying test patterns to the inputs of the circuit. Since the priority encoder is the most complicated combinational logic block the time of its testing determines the total test length of the random test sequence applied to the inputs of the register destination decoder. Using the *ViewLogic* fault simulator a set of test patterns was found which detects all possible single stuck-at faults in the priority encoder. Actually, during the test all the stuck-at faults both in the logic blocks and in control lines have been detected. It was observed that all the single stuck-at faults in the control blocks which are not involved into the communication process in test mode caused a deadlock of the whole circuit during its normal operation.

The test set for the priority encoder consists of 47 test patterns including:

1) one test pattern which contains all zeros;

2) sixteen ‘running one’ test patterns;

3) thirty test patterns everyone of which includes only two ones and all zeros.

Suppose that the probability of a one on each input of the register destination decoder, p_i ($0 \le i \le 15$), is independent, constant and equal to p ($0 < p < 1$). Let P_1, P_2 and P_3 be the probabilities of an ‘all zeros’ test, a particular ‘running one’ test and a particular ‘just two ones’ test pattern respectively. Therefore,

$$P_1 = q^{16} ; P_2 = q^{15} \cdot p ; P_3 = q^{14} \cdot p^2 ;$$

where $p+q=1$.

Let us estimate the optimal input signal probability on each input of the register destination decoder in order to obtain the minimal random test sequence which will guarantee with the given probability p_t the appearance of all 47 test patterns from the test set.

As $0 < p < 0.5$ the following inequality takes place:

$$P_3 < P_2 < P_1 .$$

The optimal output probability for the APRPG can be found from the following expression:

$$\underset{p}{MAX} \; MIN(P_1, P_2, P_3) = \underset{p}{MAX}(P_3) \quad .$$

The extremums of function P_3 can be estimated as:

$$\frac{d}{dq}\left(q^{14} (1-q)^2 \right) = 0 \text{ or } 16q^2 - 30q + 14 = 0 . \qquad (3)$$

The solutions of equation (3) are $q_1 = 1$ and $q_2 = 0.875$. The first solution of the equation cannot be accepted whereas the second solution is that required by the test procedure. Therefore, $p_{opt} = 1 - q_2 = 0.125$.

Figure 8 illustrates a graph of the dependency of the percentage of undetected stuck-at faults in the priority encoder on its input signal probabilities after the application of 200 random test patterns to the inputs of the register destination decoder. The minimum of the graph corresponds to the optimal signal probability which is equal to 0.125.

The minimal length of the random test sequence which will detect all the stuck-at faults in the register destination decoder with probability p_t $(p_t + q_t = 1)$ can be calculated using equation (2) where $p_{min} = (1 - p_{opt})\, p_{opt}^2$ or $p_{min} = 2.4 \cdot 10^{-3}$, g=30. Therefore,

$$T_{30}\,(p_t = 0.99) = 3320;\ T_{30}\,(p_t = 0.999) = 4274.$$

The appropriate values for the test lengths of the test procedure with the use of equiprobable test patterns are as follows:

$$T_{47}\,(p_t = 0.99) = 554127;\ T_{47}\,(p_t = 0.999) = 705029;$$

As a result, the number of test patterns for random testing the register destination decoder using weighted random test patterns has been reduced by up to 165 times.

The implementation of the register destination decoder in CMOS technology requires 1011 transistors whereas the number of transistors for the implementation of the testable register destination decoder is 1290. Thus, the hardware redundancy of the testable register destination decoder is 27%. This high rate of hardware redundancy is a consequence of the relative simplicity of the combinational logic in the register destination decoder.

7. Conclusions

The approach described in this paper allows the design of asynchronous sequential VLSI circuits for random pattern testability. During test mode the asynchronous sequential circuit is tested asynchronously in a manner similar to well-known scan techniques: the combinational logic block and all the storage elements are tested independently which simplifies the test greatly. The proposed scan test technique provides for the bit serial scanning of test patterns into the state registers of the circuit and the bit serial scanning out of the responses of the combinational logic block from the internal signature register. This makes the scan testing faster then a traditional scan test. The random pattern test length for the testable asynchronous sequential circuit is equal to the test length of the random testing of the combinational circuit and can be estimated easily. All the single stuck-at faults in the data processing and control blocks can be detected during the test. The hardware redundancy of the proposed approach depends greatly on the complexity of the combinational logic block. This approach to designing testable asynchronous sequential circuits can be used effectively in developing BIST VLSI circuits where the test generator and the signature analyser are placed on the chip.

The random pattern testable structure for the register destination decoder has been considered. The results show that the proposed approach has practical flexibility which allows it to be used to design various kinds of asynchronous sequential circuits for random pattern testability.

8. References

[1] Mead C., Conway L. Introduction to VLSI systems. Addison-Wesley Publishing Company, 1980.

[2] Lavagno L., Sangiovanni-Vincentelli A. Algorithms for synthesis and testing of asynchronous circuits. Kluwer Academic Publishers, 1993.

[3] Hauck S. Asynchronous design methodologies: An overview. Proc. of IEEE, Vol. 83, No. 1, Jan., 1995, pp. 69-93.

[4] Sutherland I.E. Micropipelines. Communications of the ACM, Vol.32, no.6, June, 1989, pp. 720-738.

[5] Furber S. B., Day P., Garside J. D., Paver N. C., Woods J.V. A micropipelined ARM. Proceedings of the IFIP TC 10 WG 10.5 International Conference on VLSI, Grenoble, France, 6-10 September 1993, pp. 5.4.1-5.4.10.

[6] Paver N. The design and implementation of an asynchronous microprocessor. Ph.D. Thesis, University of Manchester, 1994.

[7] Russell G., Sayers I. L. Advanced simulation and test methodologies for VLSI design. Van Nostrand Reinhold (International), 1989.

[8] McCluskey E. J. Logic design principles: with emphasis on testable semicustom circuits. Prentice-Hall International Editions, 1986.

[9] Wagner K.D., Chin C.K., McCluskey C.J. Pseudorandom testing. IEEE Transactions on Computers, C-36(3), 1987, pp. 332-343.

[10] Thorel P., Rainard J. L., Botta A., Chemarin A., Majos J. Implementing boundary-scan and pseudo-random BIST in an asynchronous transfer mode switch. 1991 International Test Conference, 1991, pp. 131-139.

[11] Chin-Long Wey, Ming-Der Shieh, Fisher D. ASCLScan: a scan design for asynchronous sequential circuits. Proc. of IEEE Int. Conf. on Computer-Aided Design, 1993, pp. 159-162.

[12] Roncken M. Partial scan test for asynchronous circuits illustrated on a DCC error corrector. Proc. Int. Symposium on Advanced Research in Asynchronous Circuits and Systems (Async94), Utah, USA, Nov., 1994, pp. 247-256.

[13] Pagey S., Venkatesh G., Sherlekar S. Issues in fault modelling and testing of micropipelines. First Asian Test Symposium, Hiroshima, Japan, Nov., 1992.

[14] Khoche A., Brunvand E. Testing micropipelines. Proc. Int. Symposium on Advanced Research in Asynchronous Circuits and Systems (Async94), Utah, USA, Nov., 1994, pp. 239-246.

[15] Petlin O., Furber S. Scan testing of asynchronous sequential circuits. Proc. 5th Great Lakes Symposium on VLSI, New York, USA, March, 1995.

[16] Petlin O. Random testing of asynchronous VLSI circuits. M.Sc. Thesis, University of Manchester, 1994.

[17] Agrawal P., Agrawal V. D. On Monte Carlo testing of logic tree networks. IEEE Transactions on Computers, C-25(6), 1976, pp. 664-667.

[18] Waicukauski J. A., Lindbloom E., Eichelberger F. B. A method for generating weighted random test patterns. IBM J. Res. and Dev., 1989, no. 2, pp. 149-161.

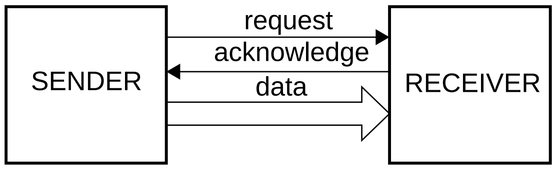


Figure 1 : *The standard bundled data interface*

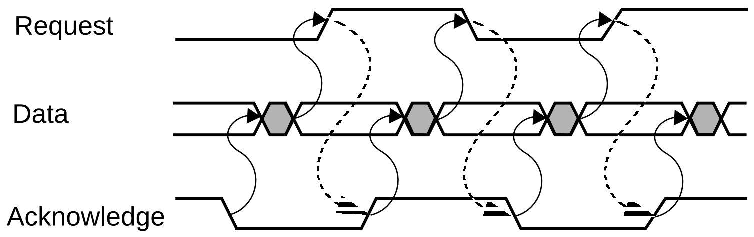


Figure 2 : *Two-phase bundled data communication protocol*

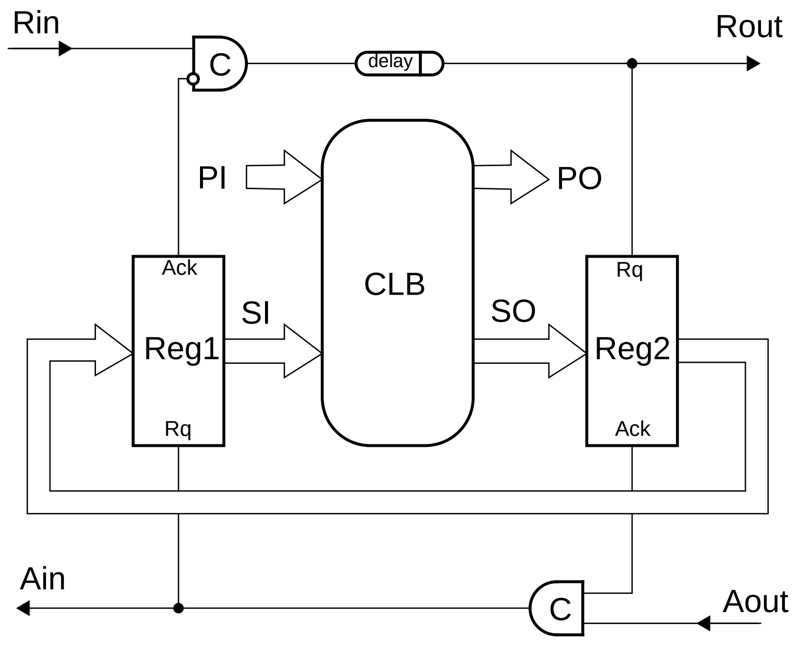


Figure 3 : *The general structure of an asynchronous sequential circuit*

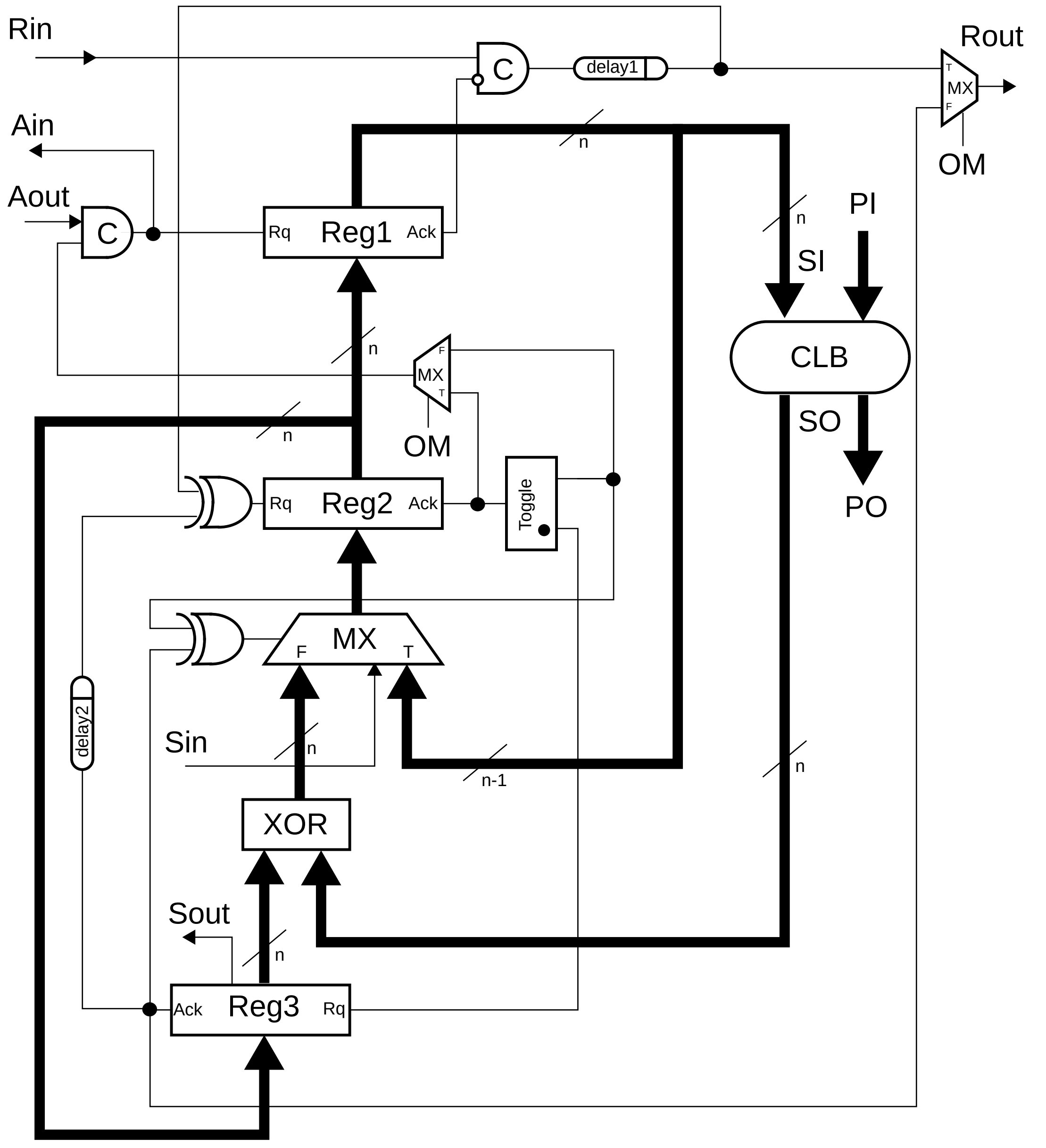


Figure 4 : *The testable asynchronous sequential circuit*

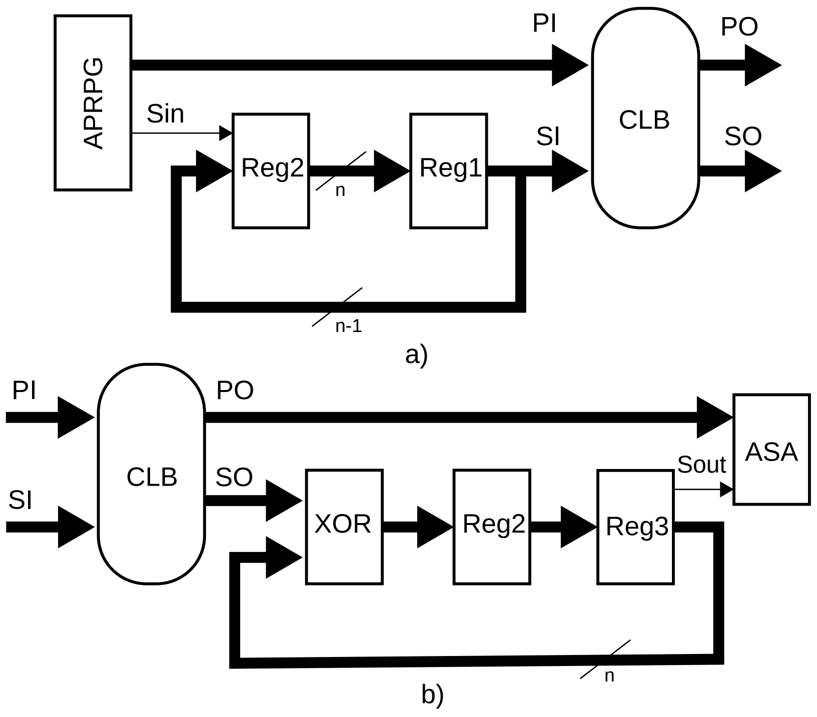
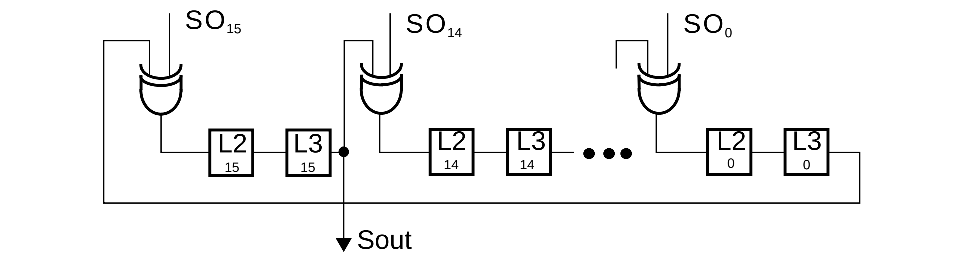
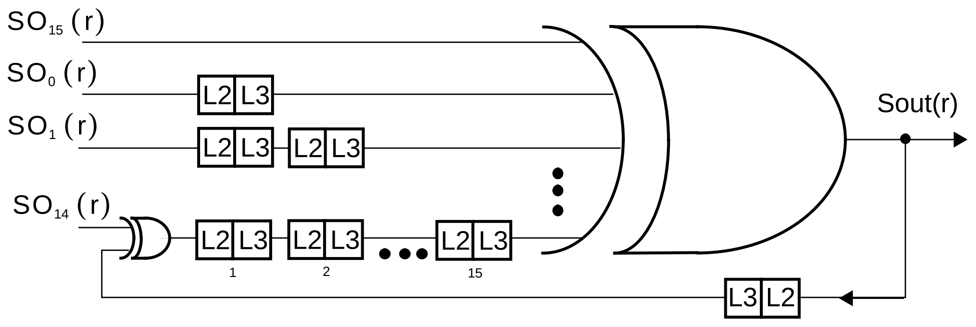


***Figure 5 :** The mechanism for a) applying test patterns to the inputs of the CLB and b) compressing the responses from the outputs of the CLB during the test*



a)



b)

***Figure 6 :** Compressing the test data from the internal outputs of the CLB: a) the structure of the signature analyser; b) the equivalent schematic of the signature analyser*

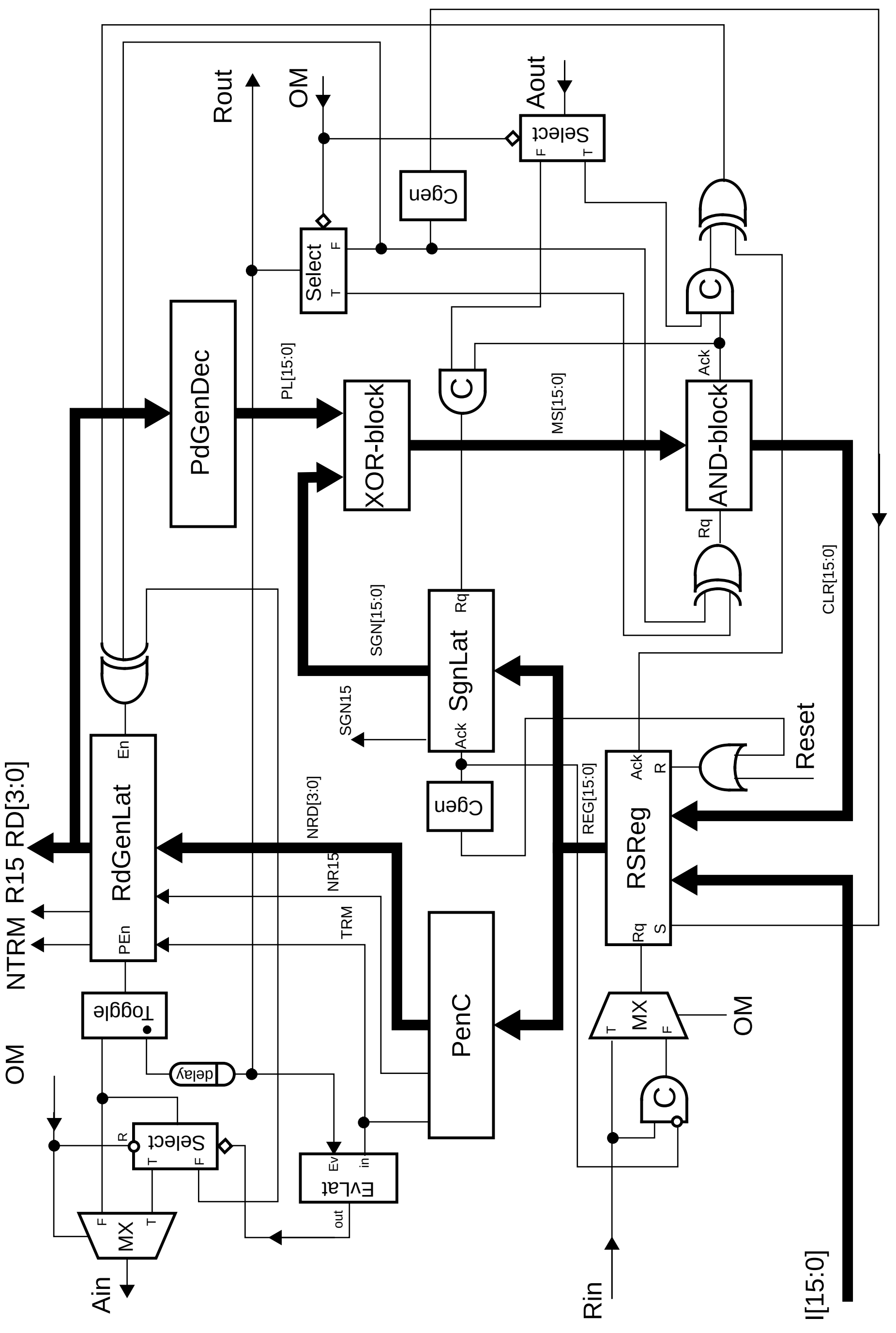


Figure 7 : *A testable structure of the register destination decoder*

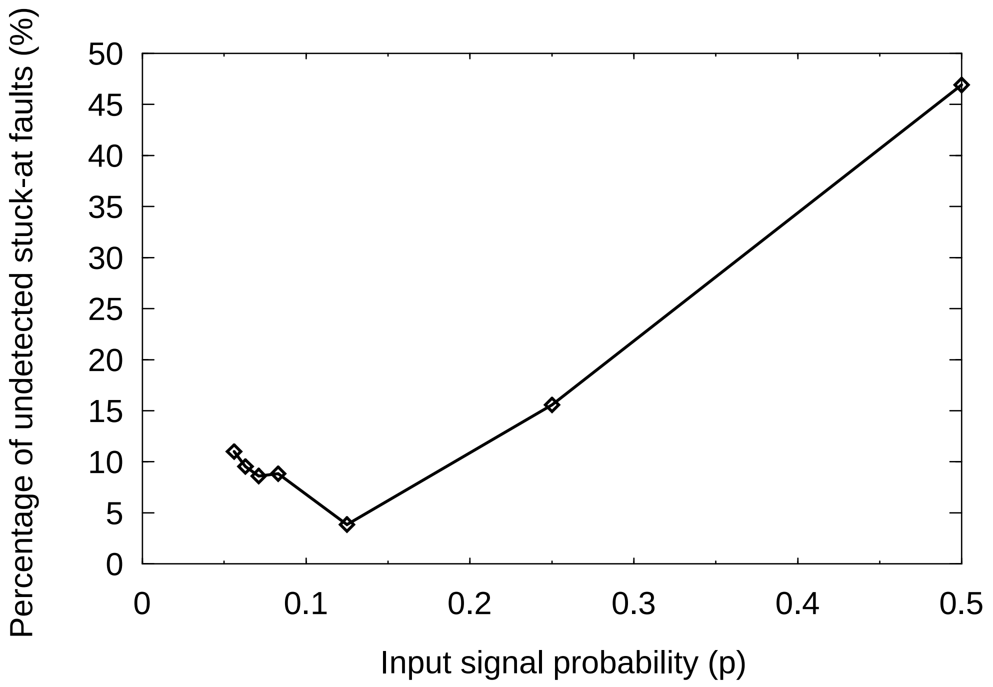


Figure 8 : *The dependency of the percentage of undetected stuck-at faults in the priority encoder on its input signal probabilities after the application of 200 random test patterns*