TESTCHIP: A Chip for Weighted Random Pattern Generation, Evaluation, and Test Control

Albrecht P. Ströle, Hans-Joachim Wunderlich, Oliver F. Haberl

University of Karlsruhe
Institute of Computer Design and Fault Tolerance
(Prof. D. Schmid)
P. O. Box 6980
D-7500 Karlsruhe 1
Federal Republic of Germany

ABSTRACT: A chip is presented that generates weighted random patterns, applies them to a circuit under test and evaluates the test responses. The generated test patterns correspond to multiple sets of weights. Test response evaluation is done by signature analysis. The chip can easily be connected to a micro computer and thus constitutes the key element of a low-cost test equipment.

1. INTRODUCTION

Conventional test strategies using deterministic test pattern generation and an automatic test equipment cause high costs and thus lead to severe problems, particularly for ASICs with their moderate production volumes. As an alternative random pattern testing is very attractive, since the computation intensive automatic test pattern generation is eliminated. The expensive automatic test equipment can be avoided, if pattern generation, test response compression, and test control are implemented on the chip itself (built-in self-test) or moved to an extra chip specially designed for this purpose. An extra chip is often more advantageous, as the design effort and the silicon area for a built-in self-test are saved.

Such an approach for external random pattern generation was reported in [BaMc82] and [EiLi83]. A higher fault coverage is attainable if weighted patterns are used. In [WLEF89] a complex system for weighted random pattern testing is described. At IBM it is used for the production test of LSSD logic chips.

In this paper we present a chip that performs the main tasks of an automatic test equipment:

- Pattern generation: Weighted random patterns corresponding to multiple sets of weights can be generated.
- Test response compression: Signature analysis is performed.
- Test control.

The central parameters have to be programmable in order to adapt them to a wide range of circuits under test (CUT). This makes the designed chip a key element in building a low-cost test equipment. Our test system is for example intended to be used for testing chips of multichip projects designed by students.

In section 2 we discuss the characteristic features and advantages of random pattern testing corresponding to multiple sets of weights. The proposed test configuration and the chip design are presented in section 3. Section 4 describes the application of the chip. Finally, section 5 concludes with a short summary.

2. RANDOM PATTERN TEST CORRESPONDING TO MULTIPLE SETS OF WEIGHTS

When random patterns are used, the expensive automatic test pattern generation is eliminated. Compared to a deterministic test pattern set, random patterns can be generated faster, but a greater number of patterns is required. On the other hand this increase in the test length improves the ability to detect nonmodeled faults. To reduce the random test length, weighted random patterns are applied. Here the probabilities of a logical "1" (weights) for each position of a pattern are optimized in order to get a short test length. In [Wu88] it is shown that this can reduce test lengths by orders of magnitude. Multiple sets of weights can shorten test lengths even more. To test the ISCAS'85 benchmark circuit c880 [BrPH85] for example only 660 optimized random patterns are required rather than 37000 patterns without optimization. Methods to determine the weights for a given CUT are presented in [Wu85], [LBGG86], [Wu88], and [WLEF89]. Thus all combinational circuits can be random-tested and the time for the random test execution is comparable with the time for a deterministic test. Today the test by weighted random patterns is widely accepted. In the following we present a test configuration for the same test strategy. Instead of an expensive test equipment, however, we only need low-cost hardware, an ASIC (TESTCHIP) and a personal computer.

3. TESTCHIP

3.1. CHIP ARCHITECTURE

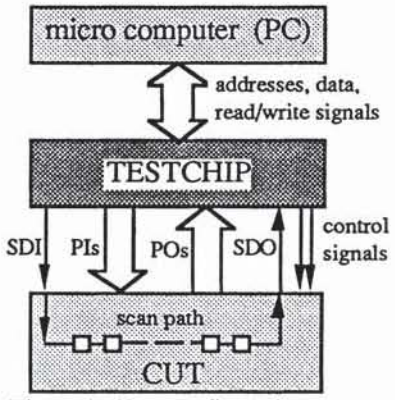

Figure 1: Test configuration

Figure 1 shows the test configuration. On one side TESTCHIP is coupled with a personal computer (PC) or another micro computer, that supplies the user interface, initializes the test execution and evaluates the results. On the other side TESTCHIP is connected to the CUT. As it is usual for sequential circuits, the CUT is provided with a scan path. In the test mode the circuit is then partitioned into a combinational logic part and a set of storage elements configured into a shift register (SDI: serial data in, SDO: serial data out). TESTCHIP generates patterns for the primary inputs (PIs) and the scan path of the CUT. It controls the test execution by means of a clock signal and a mode switching signal (test mode or normal mode). Finally TESTCHIP compresses the test responses from the primary outputs (POs) and the scan path of the CUT.

A more detailed view of TESTCHIP is presented in figure 2. It contains two completely separated pattern generators to guarantee that the pseudo-random test patterns produced for the primary inputs and the scan path of the CUT are statistically independent. The patterns produced serially by pattern generator 2 are immediately shifted into the scan path. The patterns from pattern generator 1 are first shifted into an internal shift register of TESTCHIP (externally extensible) and then applied in parallel to the primary inputs. With independently programmable weights for each position this requires much less hardware than generating parallel patterns at once. In general this does not increase the test execution time, since the scan path has to be loaded serially anyway and in most cases the number of scan path elements is larger than the number of the primary inputs.

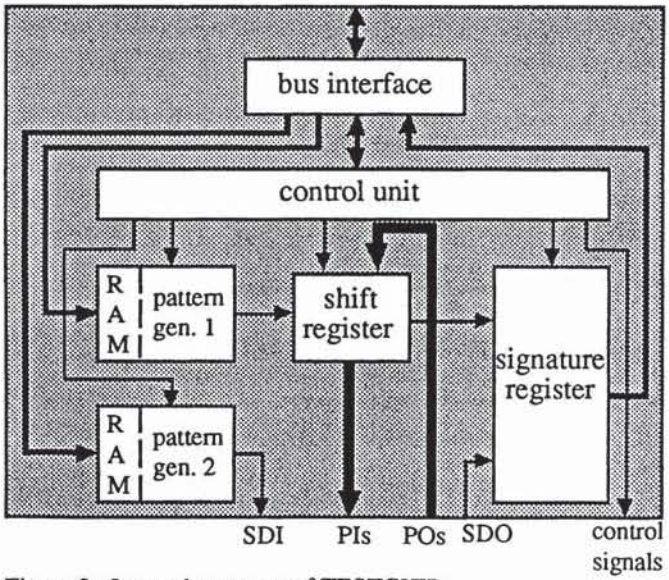

Figure 2: Internal structure of TESTCHIP

When test patterns have been applied, the CUT is switched to normal mode and clocked once. Thus we get a test response in the scan path. At the same time the results of the primary outputs are loaded into the internal shift register of TESTCHIP (the data for the primary inputs are not required any more at that time). Afterwards the CUT is switched back to test mode. The contents of the shift register and the scan path are transfered to the 2-input signature register, that compresses the test responses bit by bit. The implemented signature register uses a primitive feedback polynomial of degree 32. The probability of aliasing (i.e. a faulty circuit leads to the same signature as the faultless circuit and thus the fault cannot be detected) is 2^{-32} for long test lengths [WiDa87]. Simultaneously to signature analysis the next test patterns are generated. When the test execution is completed, the control unit signals the test end.
All the parameters, instructions and status information are kept in registers, accessible to the micro computer. So TESTCHIP is programmable and the test execution can be adapted to practically all CUTs with a scan path. TESTCHIP can be used to test circuits with up to 127 primary inputs, 127 primary outputs and 511 scan path elements.

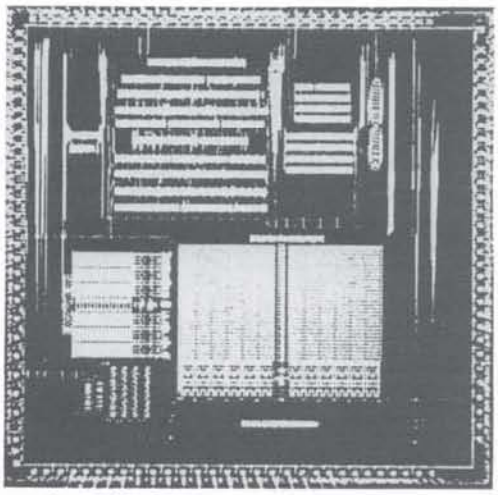

Figure 3: TESTCHIP

Up to 4 sets of weights and test lengths from 1 to 10^6 for each set are programmable. If the requirements of a CUT exceed these features, several TESTCHIPs can be combined. The pattern generators and the signature analyzer can operate at a speed of $2*10^7$ bits/s.

TESTCHIP has been implemented using the standard cell design system VENUS [HöNS86]. Samples of the CMOS-chip, that contains about 64000 transistors, have been produced and completely satisfy the specification. The chip photo (figure 3) shows the RAMs containing the sets of weights (the block in the middle of the lower half and the smaller block to the left of it). The large standard cell block above them is the control unit. The three smaller standard cell blocks contain the LFSRs for pattern generation and signature analysis.

3.2. PATTERN GENERATORS

In the last few years, several hardware structures have been proposed that generate weighted pseudo-random patterns ([Wu87], [WLEF89], [BrGK89]). All of them are based on linear feedback shift registers (LFSRs). The LFSR is tapped at some stages and the pseudo-random bits at these stages are combined using boolean functions and multiplexers to get the desired weights. The approaches differ in the selection of the tap positions and in the way the bits are combined.

The pattern generator proposed in [WLEF89] has two disadvantages. Firstly, the possible weights are not distributed uniformly. There are more weights near 0 and near 1 than in the neighborhood of 0.5. Secondly, to avoid direct dependences the bits used in producing a single weighted bit must be shifted out of the LFSR before succeeding weighted bits are created.

We designed a serial version of the parallel pattern generator presented in [Wu87]. The two implemented pattern generators are based on modular 32-bit-LFSRs with different primitive feedback polynomials. There are several feedback connections between the three tap positions. So the pseudo-random bits at the tap positions are practically independent. The weights $\frac{1}{8},\frac{1}{4},\ldots,\frac{7}{8}$ are realized by boolean functions and can be selected by a multiplexer. The associated RAMs contain sequences of 3-bit-codes that control the multiplexer and determine the weight for each position in the generated patterns and for each set of weights.

The pattern generator of [BrGK89] does not offer any advantages compared with our design. A greater choice of weights does not decrease the test length significantly. Even replacing the LFSR by a linear cellular automaton with maximum period does not give any benefits, since each cell of such a cellular automata produces (apart from a phaseshift) exactly the same pseudo-random bit sequence as a stage of the corresponding LFSR. (For lack of space the proof using basic facts of linear algebra is omitted here.)

4. APPLICATION

The low-cost test configuration of figure 1 can be used for the test of all circuits with a scan path. As an example we take the ISCAS'89 benchmark circuit s1196 of [BrBK89] with 14 primary inputs, 14 primary outputs, and a scan path of 18 storage elements. The first step is to calculate the sets of weights for the random patterns generated to test the circuit s1196. The resulting 4 sets (using only weights $\frac{1}{8},\frac{1}{4},\ldots,\frac{7}{8}$) and the test lengths to get a desired fault coverage of 99.9% are listed in table 1.

	set 1	set 2	set 3	set 4
test length (#patterns)	30145	49073	49073	25313
weights for the primary inputs	4,7,6,6,2,3,5, 4,6,5,4,7,4,2	7,5,6,7,4,1,7, 7,7,1,2,2,1,1	3,7,7,7,1,4,7, 7,7,7,2,6,6,2	3,6,7,7,3,3,6, 4,2,5,3,7,4,2
weights for the scan path elements	7,1,3,1,7,1,7, 7,7,7,1,7,1,1, 7,7,1,4	1,1,1,7,7,1,7, 7,7,7,1,1,1,1, 7,7,1,4	1,1,1,1,7,1,7, 1,1,1,1,1,1,1, 7,1,1,7	1,7,7,1,7,1,7, 7,7,4,1,1,1,1, 7,1,1,6

Table 1: Sets of weights for the circuit s1196 (units of $\frac{1}{8}$)

Then the fault coverage is validated by fault simulation and the signature for the faultless circuit is determined. The parameter registers and the RAMs of TESTCHIP are loaded with the characteristic data of the circuit s1196: 14 primary inputs, 14 primary outputs, 18 scan path elements and the test lengths and weights of table 1.

Whereas all these preparations have been done on the PC, now TESTCHIP carries out the test. The test execution takes 0.3 s. Afterwards the signature register of TESTCHIP is read and compared to the expected signature in order to decide whether the circuit under test is faulty or not.

5. CONCLUSIONS

A chip for weighted random pattern generation corresponding to multiple sets of weights, test response compression, and test control has been designed. Samples of this CMOS-chip have been produced and tested. The chip can be used for the test of a wide range of circuits with a scan path. Together with a personal computer a complete low-cost test equipment can be built.

REFERENCES

BaMc82 P. H. Bardell, W. H. McAnney, "Self-Testing of Multichip Logic Modules", Proc. International Test Conference, pp. 200-204, 1982

BrBK89 F. Brglez, D. Bryan, K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits", Proc. International Symposium on Circuits and Systems, pp. 1929-1934 Portland, Oregon, 1989

BrGK89 F. Brglez, C. Gloster, G. Kedem, "Hardware-Based Weighted Random Pattern Generation for Boundary Scan", Proc. International Test Conference, pp. 264-274, Washington 1989

BrPH85 F. Brglez, P. Pownall, R. Hum, "Accelerated ATPG and Fault Grading via Testability Analysis", Proc. International Symposium on Circuits and Systems, pp. 695-698, 1985

EiLi83 E. B. Eichelberger, E. Lindbloom, "Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test", IBM Journal of Research and Development, Vol. 27, No. 3, May 1983

HöNS86 E. Hörbst, M. Nett, H. Schwärtzel, "VENUS - Entwurf von VLSI-Schaltungen", Springer, 1986

LBGG86 R. Lisanke, F. Brglez, A. de Geus, D. Gregory, "Testability-Driven Random Pattern Generation", Proc. International Conference on Computer-Aided Design, pp. 144-147, 1986

WiDa87 T. W. Williams, W. Daehn, M. Gruetzner, C. W. Starke, "Aliasing Errors in Signature Analysis Registers", IEEE Design&Test, April 1987, pp. 39-45

WLEF89 J. A. Waicukauski, E. Lindbloom, E. B. Eichelberger, O. P. Forlenza, "A Method for Generating Weighted Random Test Patterns", IBM Journal of Research and Development, Vol. 33, No. 2, pp. 149-161, March 1989

Wu85 H.-J. Wunderlich, "PROTEST: A Tool for Probabilistic Testability Analysis", Proc. 22nd Design Automation Conference, pp. 204-211, Las Vegas 1985

Wu87 H.-J. Wunderlich, "Self-Test Using Unequiprobable Random Patterns", Proc. 17th International Symposium on Fault-Tolerant Computing FTCS-17, pp. 258-263, Pittsburgh 1987

Wu88 H.-J. Wunderlich, "Multiple Distributions for Biased Random Test Patterns", Proc. International Test Conference, pp. 236-244, Washington 1988