Ročník 2008 electroscope Číslo II

Investigation and Implementation of Test Vectors for Efficient IC Analysis

M. Brutscheck[1,2], M. Franke[2], A. Th. Schwarzbacher[1], St. Becker[2]
[1]School of Electronic and Communications Engineering,
Dublin Institute of Technology, IRELAND
[2]Department of Computer Science and Communications Systems,
University of Applied Sciences Merseburg, GERMANY
E-mail: michael.brutscheck@hs-merseburg.de

Abstract:
Up until now, the efficient and structured analysis of unknown CMOS integrated circuits (ICs) has become a topic of great relevance. In the last decade different invasive and non-invasive strategies have been developed to analyse unknown ICs. However, invasive procedures always lead to the destruction of the system under investigation. Non-invasive approaches published so far have the disadvantage that ICs are analysed using very complex algorithms. Here, no division is carried out to avoid extensive analysis times in the case that only simple structures are investigated. This paper describes the investigation and implementation of test vectors for efficient IC analysis. To demonstrate the correct operation the non-invasive classification procedure will be used. Furthermore, in this research the properties of several test vectors will be analysed and implemented into analysis environment. All sections of the procedure proposed are simulated and fully tested on ISCAS-85, ISCAS-89 and ISCAS-99 benchmark or user defined models of real ICs and the results are presented in this paper. In every circuit analysed the behaviour has been successfully determined by the use of the proposed test vectors.

INTRODUCTION

Today's unknown CMOS integrated circuits have very complex structures and are designed using a great variety of functions and behaviours. Since the functions of ICs are not always known it can be essential to determine their correct behaviour. This may be required when the label is lost or it is necessary to find out more about the internal structure of the integrated circuit. Moreover, it is conceivable to use structures of ICs discontinued in new IC designs or to add new functionality to an existing system. Here, finite state machines (FSMs) are the main component in today's integrated circuits. To make a structured analysis of these ICs possible, it is essential to create an abstract model of the real IC. Therefore, the theory of automata will be used to abstract the unknown system. Automata can be divided into deterministic and non-deterministic state machines. Real digital integrated circuits work as deterministic state machines. Only in the case of a malfunction of an integrated circuit non-deterministic behaviour may occur. In this research properly operating systems will be considered. That means that only complete and consistent systems will be investigated. Furthermore, it must be distinguished between infinite and finite state machines. Usually, the IC designers develop and realise integrated circuits as finite state machines. Therefore, only finite state machines are considered in this paper. It can be said that for a classification of unknown systems deterministic finite state machines must be considered.

In general, the classification procedure determines not the complete internal function of IC investigated, but general behaviour of the FSM, which can be divided into blocks with combinatorial, linear or nonlinear behaviour using automata theory. The separation procedure used was already fully implemented by the authors and is described in detail in [5]. The overall research objective of this project is to find non-invasive reverse engineering procedures to extract the function of unknown CMOS integrated circuits.

THEORY AND BACKGROUND OF TEST VECTORS

In the classification procedure linear maximum sequences for the generation of pseudo noise (pn) sequences were used. They are an important subclass of pseudo random sequences [10]. In general, linear homogeneous finite difference equations can be written as (1).

$$\begin{aligned} &c_0 a_i + c_1 a_{i-1} + \ldots + c_n a_{i-n} = 0, \\ &i = n, n+1, \ldots, c_0 c_n \neq 0 \end{aligned} \tag{1}$$

Using equation (1) an infinite sequence of elements as in (2) can be defined.

$$\{a_i\}_0^\infty = a_0, a_1, a_2, \ldots \tag{2}$$

Rewriting (1) these sequences are recursive, because each element a_i can be calculated from n previous elements as illustrated in (3).

$$a_i = -\frac{1}{c_0}\sum_{v=1}^{n} c_v a_{i-v}, \quad c_v, a_i \in GF(q) \qquad (3)$$

The elements of the sequences considered and the recursive coefficients will be taken from a Galois field. A formal power series as in (4) can be assigned to a sequence $\{a_i\}_0^\infty$.

$$G(D) = a_0 + a_1 D + a_2 D^2 + \ldots + a_i D^i + \ldots \qquad (4)$$

Equation (4) is also called a generating function or D-transformed of the sequence. Linear recursive sequences, which are generated by (1) with (3) are always periodic. Using (3) it can be seen in (5) that the particular initial condition according to Equation (5) only repeats itself and has the period $N = 1$.

$$(a_{-n}, a_{-n+1}, \ldots, a_{-1}) = (0, 0, \ldots, 0) \qquad (5)$$

Consequently, the maximum period length is illustrated in (6), because there are q^n-1 different n-digit vectors $s \neq 0$ with elements from *GF(q)*.

$$N = N_{\max} = q^n - 1 \qquad (6)$$

A linear recursive sequence, which satisfies (1) and has a period length as in (6), can be called q-significant maximum sequence of order n. Therefore, the period length N of a maximum sequence is independent of its initial condition. Moreover, all maximum sequences, which satisfy one and only one finite difference equation, only differ in shifting. These two properties result in many advantages due to practical realisations of these sequences, for example no specific initial conditions are necessary except $s \neq 0$. Short disturbances do not influence the generated sequence. For this, the maximum sequence will be generated from the set of linear recursive sequences. Regarding the periodicity as in (4) the equation can be rewritten as in (7):

$$G(D) = \left[1 + D^N + D^{2N} + \ldots\right] \sum_{i=0}^{N-1} a_i D^i \qquad (7)$$

Using the identity as described in (8)

$$1 = \left(1 - D^N\right)\left(1 + D^N + D^{2N} + \ldots\right) \qquad (8)$$

the special binary case can be written as follows:

$$\frac{1 - D^N}{c(D)} = \sum_{i=0}^{N-1} a_i D^i \qquad (9)$$

If the exponent e is equal to the period length N then it follows from (9), that the characteristic polynomial *c(x)* is the divider of binomial $(1\text{-}D^e)$. In the case of maximum period all initial conditions $s \neq 0$ are equitable. This means that a special initial state represents no constraint. Furthermore, an essential condition for the maximum period length is a characteristic polynomial. The minimum number e for which $(1\text{-}x^e)$ can be divided without a remainder is called exponent or period of polynomial *f(x)*. Therefore, a polynomial with maximum exponent $e = N_{max}$ must be irreducible. A polynomial *f(x)* of the order n can be called primitive, if $(x^N\text{-}1)$ can be divided by *f(x)* without a remainder with $N = q^N\text{-}1$. This is not valid for $N < q^N\text{-}1$.

To realise a pseudo-noise random sequence, it is important that each sequence consists of uniformly distributed values *'0'* and *'1'*. The autocorrelation function (ACF) of maximum sequences can be completely determined [10]. The special characteristic depends on the choice of amplitude values A_i assigned to its elements of *GF(q)*. The autocorrelation function of binary maximum sequences using the assignment $A_0 = -1, A_1 = +1$, $x_i \in \{A_0, A_1\}$ is shown in (10).

$$R_{x,x}(s) = \begin{cases} 1 & s \equiv 0 \bmod N \\ -\dfrac{1}{2^n - 1} & otherwise \end{cases} \qquad (10)$$

Figure 1 illustrates equation (10).

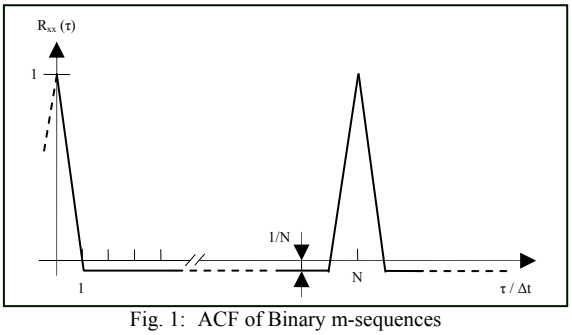

Fig. 1: ACF of Binary m-sequences

For each input pin of the unknown IC a completely different sequence has to be generated, as it is necessary to create an amount of different input values as high as possible. Table 1 shows the characteristics of binary maximum sequences used in this research with their generator polynomials and initial states. Here, generator polynomials of length $l = 2^{10}\text{-}1 = 1023$ were used and their autocorrelation functions were determined. Moreover, the polynomials were successfully checked towards that their ACF leads to maximum at 0 as illustrated in Figure 1. For the generation of test sequences a linear feedback shift register (LFSR) is used, which is easy to implement into the analysis environment. Afterwards, i different pn-sequences have to be applied to the inputs in $2^i\text{-}1$ different combinations, where i is the number of input pins.

Table. 1: Generator Polynomials for pn-sequence Generation

Number of pn-sequence	*Generator Polynomial*	*Initial State*
1	[1 0 0 1 0 0 0 0 0 0 1]	[0 0 0 0 0 0 0 0 0 0 1]
2	[1 1 0 1 1 0 0 0 0 0 1]	[0 0 0 0 0 0 0 1 0 0]
3	[1 1 1 0 0 1 0 0 0 0 1]	[0 0 0 0 0 0 0 1 1 1]
4	[1 0 1 1 0 1 0 0 0 0 1]	[0 0 0 0 0 0 1 0 1 0]
5	[1 0 1 0 0 1 1 0 0 0 1]	[0 0 0 0 0 0 1 1 0 1]
6	[1 1 1 1 0 1 1 0 0 0 1]	[0 0 0 0 0 1 0 0 0 0]
7	[1 0 0 0 0 0 0 1 0 0 1]	[0 0 0 0 0 1 0 0 1 1]
8	[1 1 0 1 0 0 0 1 0 0 1]	[0 0 0 0 0 1 0 1 1 0]
9	[1 0 1 0 0 0 1 1 0 0 1]	[0 0 0 0 0 1 1 0 0 1]
10	[1 1 1 0 1 0 1 1 0 0 1]	[0 0 0 0 0 1 1 1 0 1]

In the example illustrated in Table 2 the number of input is $i = 5$. Next, for every pn-sequence combination the results of each output have to be recorded and stored in a data table, e.g. for Table 2 31 output sequences have to be recorded.

Table. 2: Generator Polynomials for pn-sequence Generation

Number of Combination	*Inputs x_i*				
	x_1	x_2	x_3	x_4	x_5
1	pn_1	0	0	0	0
2	0	pn_2	0	0	0
3	pn_1	pn_2	0	0	0
…	…	…	…	…	…
30	0	pn_2	pn_3	pn_4	pn_5
31 (2^i-1)	pn_1	pn_2	pn_3	pn_4	pn_5

Finally, the dependencies can be solved by forming equations for each output. If an output reacts to a particular combination of applied pn-sequences then the output depends on the appropriate inputs. For example if an output responds to the 30th combination represented in Table 2, it depends on the inputs x_2-x_5. If the resulting sequence of the same output with the 31st combination applied is equal to the former combination, the output depends on every input except the first one. Each output, which has the same input dependencies, belongs to the same internal block.

IMPLEMENTATION AND RESULTS

The separation procedure from [5] was used to prove the correctness of the test vectors implemented. In principle the classification procedure works in three steps, the determination of independent blocks, the division into combinatorial or sequential FSMs and the division into linear or nonlinear sequential FSMs. Here, different combinations of test sequences are applied at the input, then the output values have to be stored and at last, equations must be solved. In the next step, HDL-code of benchmark ICs of the ISCAS-85, ISCAS-89 and ISCAS-99 series [8] and [9] and user defined IC models were implemented into MATLAB [7] and used to prove the correct mode of operation of the classification procedure. Table 3 shows the results of IC models implemented into the analysis environment. Here, *C* stands for combinatorial, *L* for linear sequential and *NLS* for non-linear sequential, respectively.

Table. 3: Result Table of IC Models Analysed

IC	*IB*	*FCNo of IB*	*Exp. Beh.?*	*Found Beh.?*	*Sim. Time [s]*
C17	2	Yes	2xC	Yes	14,6
S27	1	Yes	NLS	Yes	4,3
S298	1	Yes	NLS	Yes	19,6
B01	1	Yes	NLS	Yes	3,91
B02	1	Yes	NLS	Yes	1,14
B03	1	Yes	NLS	Yes	24,6
B06	1	Yes	NLS	Yes	7,4
B11	1	Yes	NLS	Yes	169,7
EC1	1	Yes	C	Yes	12,7
EC2	4	Yes	4xC	Yes	25,6
EC3	4	Yes	4xC	Yes	88,1
ELS1	1	Yes	LS	Yes	1,9
ELS2	2	Yes	2xLS	Yes	23,2
ELS3	4	Yes	4xLS	Yes	240,6
ELS4	5	Yes	5xLS	Yes	905,0
ENLS1	1	Yes	NLS	Yes	2,0
ENLS2	2	Yes	2xNLS	Yes	22,7
ENLS3	4	Yes	4xNLS	Yes	241,0
ENLS4	5	Yes	5xNLS	Yes	621,3

In Table 3 *IB* is the independent block and *FCNo* means found correct number. To demonstrate the operation of the classification procedure several benchmarks models were also combined and implemented as can be seen in Table 4.

Table. 4: Result Table of Mixed IC Models Analysed

IC Mix	*Total IB*	*FCNo of IB*	*Exp. Beh.?*	*Found Beh.?*	*Sim. Time [s]*
2xC17	4	Yes	C17: 2xC	Yes	582,9
C17, S27	3	Yes	C17: 2xC S27: NLS	Yes	201,2
2xS27	2	Yes	S27: NLS	Yes	78,6
S27, 2xB03	3	Yes	S27: NLS B03: NLS	Yes	984,6
B01, 2xB03	3	Yes	B01: NLS B03: NLS	Yes	971,1
B01, B06	2	Yes	B01: NLS B06: NLS	Yes	34,8
B03, B06	2	Yes	B03: NLS B06: NLS	Yes	180,8
B01, B11	2	Yes	B01: NLS B11: NLS	Yes	854,2
B03, B11	2	Yes	B03: NLS B11: NLS	Yes	1070,9
B01, B02, B06	3	Yes	B01: NLS B02: NLS B06: NLS	Yes	153,1
EC1, ELS1, ENLS1	3	Yes	EC1: C ELS1: LS ENLS1: NLS	Yes	78,9
EC2, ENLS1, ENLS2	7	Yes	EC2: 4xC ENLS1: 1xNLS ENLS2: 2xNLS	Yes	991,3
ELS1, ELS4	6	Yes	ELS1:1xLS ELS4: 5xLS	Yes	1099,3
ENLS1, ENLS4	6	Yes	ELS1: 1xNLS ENLS4: 5xNLS	Yes	942,6
EC1, ENLS3	5	Yes	EC1: 1xC ENLS3: 4xNLS	Yes	987,2
ELS1, ENLS4	6	Yes	ELS1: 1xNLS ENLS4: 5xNLS	Yes	816,4
EC1, ELS3	5	Yes	EC1: 1xC ELS3: 4xLS	Yes	1007,0
EC3, ELS2	6	Yes	EC3: 4xC ELS2: 2xLS	Yes	1042,0

As can be seen in Table 3 and Table 4, all IC models analysed were successfully solved and the independent blocks as well as the behaviour was correctly found. Furthermore, the simulation time in the right column shows that the analysis is

accomplished within less than an hour for even complex circuits.

CONCLUSIONS

This paper has presented the investigation and implementation of test vectors for finite state machines in unknown binary multi input – multi output CMOS integrated circuits. The analysis was performed using statistical investigation of its input-output behaviour and the classification procedure described in [5]. For this purpose all analysis steps described were implemented into MATLAB and verified using the analysis environment, as was described by the authors in [6]. It was demonstrated that the results obtained fully agree with the theoretical assumptions made. The correct operation was verified through the implementation of several benchmark ICs of the ISCAS-85, ISCAS-89 and ISCAS-99.
This paper has demonstrated that the test vectors described can be used in combination with the separation procedure described. Therefore, this paper has presented the successful investigation and implementation of test vectors for structured analysis of unknown CMOS ICs.

REFERENCES

[1] Suhua, J., Jingfeng, N., Jiaji, W.: Investigation on properties of focused ion beam enhanced etching, *IEEE Proceedings of 6th International Conference on Solid-State and Integrated-Circuit Technology*, vol. 2, Shanghai, China, October 2001, pp. 1091-1094.

[2] Blythe, S., Fraboni, B., Lall, S., Ahmed, H., Riu, U. de: Layout reconstruction of complex silicon chips, *IEEE Journal of Solid-State Circuits*, vol. 28, no. 2, February 1993, pp.138-145.

[3] Lee, D., Yannakakis, M.: Principles and methods of testing finite state machines - a survey, *Proceedings of the IEEE*, vol. 84, no. 8, August 1996, pp. 1090-1123.

[4] Kuroe, Y.: Learning and identifying finite state automata with recurrent high-order neural networks, *SICE Annual Conference*, Sapporo, August 2004, pp. 2241-2246.

[5] Brutscheck, M., Berger St., Franke, M., Schwarzbacher, A. Th., Becker, St.: Structural division procedure for efficient IC analysis, 16th IET Irish Signals and Systems Conference. June 2008, Galway, Ireland, pp. 18-23.

[6] Brutscheck, M., Schwarzbacher, A. Th., Becker, St.: A test environment for analysing unknown integrated circuits, 7th New Generation Scientist Conference. January 2006, Wernigerode, Germany, pp. 223-224, ISBN 3-00-018148-2

[7] Mathworks, www.mathworks.com, June 2008.

[8] Hansen, M. C., Yalcin, H., Hayes, J. P.: Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering, *IEEE Design and Test of Computers*, pp. 72-80, 1999.

[9] ISCAS Benchmark Circuits, www.fm.vslib.cz/~kes/asic/iscas, June 2008.

[10] Finger, A., Pseudorandom-Signalverarbeitung, Teubner Verlag Stuttgart, 1997.