Analysis of Finite State Machines in Unknown CMOS Integrated Circuits

M. Franke[1,2], M. Brutscheck[1,2], A. Th. Schwarzbacher[2] and St. Becker[1]

[1]Department of Computer Science and Communications Systems,
University of Applied Sciences, Merseburg
GERMANY
E-mail: marco.franke@stud.hs-merseburg.de

[2]School of Electronic and Communications Engineering,
Dublin Institute of Technology,
IRELAND

Abstract:
The investigation of unknown integrated circuits is increasingly more important in today's digital integrated circuit analysis. Up until now, different invasive strategies have been developed for the analysis of unknown circuits. A novel non-invasive approach to determine combinatorial and sequential finite state machines will be described in this paper. Furthermore, an analysis environment will be used for the description of the transition and the result functions.

I Introduction

Finite state machines (FSM) are used in a wide variety of areas, including sequential and combinatorial circuits. Since the functions of FSMs are not always known, it is essential to determine their correct behaviour. This might be required when the label is missing or it is necessary to find out more about the internal structure of the integrated circuit (IC). The overall research objective of this work is to show how the internal functions of unknown CMOS integrated circuits can be determined in a non-destructive manner. This research deals with simple logic gates up to complex finite state machines. The finite state machines are distinguished the combinatorial and the sequential automaton. In the first section of this paper the background of the different finite state machines will be represented. Therefore, the particular automaton will be described in its structure and in its mathematical functions. It must be considered, that combinatorial and sequential finite state machines have different structures and behaviours and therefore they require several test sequences. In the second section, the analysis environment will be explained. This analysis environment includes the media surface, the hardware and the interface between personal computer (PC) and analysis board. The media surface for the creation of test vectors and the computation of the results is programmed with Matlab. The programming of the software for the analysis board and the interface will be carried out on a Field Programmable Gate Array (FPGA) using the hardware description language VHDL. With this environment several tests are carried out and the results are then compared against the theoretically determined values. Therefore, known finite state machines for the analysis are used to prove the correct test outputs. The examples given

range from simple combinatorial gates to complex sequential finite state machines. It will be shown that the behaviour of finite state machines is determined by the mathematical system analysis. Furthermore, it will be shown that the output values are controlled by the input vectors and the states in the sequential automaton. In the last section the test result of combinatorial and sequential finite state machines will be represented and from this it will be explained how the transition function of the finite state machine can be determined.

II Theory and Background

Digital systems can be divided into sequential and combinational systems. The sequential systems can be further subdivided in synchronous and asynchronous systems as is illustrated in Figure 1.

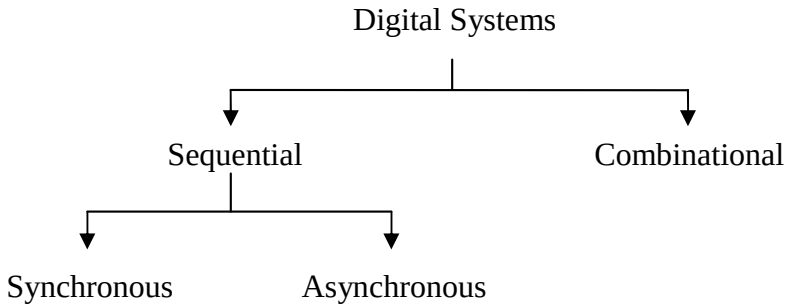

Figure 1: Overview of Digital Systems

In the following two basic finite state machines will be described which are explained detailed in [3]. The function and the behaviour of the combinatorial and sequential finite state machines will be shown in the next section.

a) *Combinatorial Finite State Machines*

The combinatorial finite state machines have a simple structure and are based on basic logic gates. The input values are interconnected directly to the output without buffering values. Therefore, the input alphabet is linked directly to the output alphabet. The combinatorial finite state machine is shown in Figure 2 below:

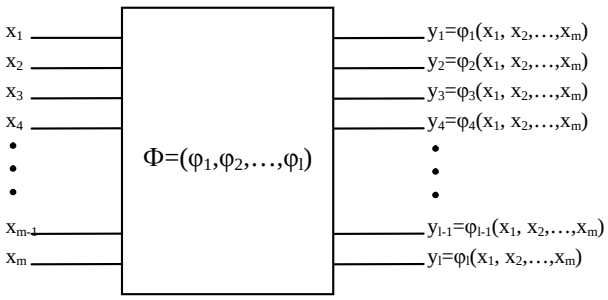

Figure 2: Combinatorial Finite State Machine

The combinatorial finite state machine is described by the transformation Φ. Thus, the transformation Φ consists of the amount l of type φ. The initial state is a transformation of the input values and it is described by $y = \varphi(x_1, x_2,\ldots,x_m)$. The combinatorial finite state machine processes input words at discrete moments. The same input words always lead to the same initial values.

b) *Sequential Finite State Machines*

To form the sequential finite state machines memory blocks are added to the combinatorial finite state machines. The memory provides the ability to store values and to realise system

states. The states of memory indicate how many input values can be stored. The output values *y(t)* are a function of the input values *x(t)* as well as of the stored values *x(t-1), x(t-2),…,x(t-n)*. They might also only be a function of the stored values *x(t-1), x(t-2),…,x(t-n)*, whereas *n* is the number of system states. Therefore, the number of system states *n* can be different. Figure 3 presents the block diagram of a sequential finite state machine.

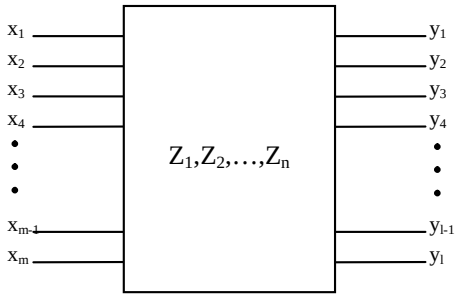

Figure 3: Sequential Finite State Machine

The sequential finite state machines can be divided into Mealy [7] and Moore [8] machines. The Mealy machine generates output values at each transition and the output values are based on the current state and the input value. The Moore machine also creates output values at each transition, but the output values are independent from the input values. The output values are only determined by the current state. The mathematical function of Mealy and Moore machine are described by a 6 – tuple $A = (X, Z, z_0, Y, \delta, \lambda)$, consisting of the finite set of input values called input alphabet X, a finite set of states Z, a initial state z_0, the finite set of output values called output alphabet Y, the state transition function $\delta(Z,X)$ and the output function $\lambda(Z,X)$. At sequential systems Mealy and Moore machine are not mutually exclusive. That means a sequential machine can be both a Mealy and a Moore machine. A sequential finite state machine is specified completely if for each state Z_i and input X_i the next state $Z_{i+1} = \delta(Z_i, X_i)$ and the output $Y_i = \lambda(Z_i, X)$ are specified. Otherwise, the automaton is specified partially.

III Analysis Environment

The analysis environment is divided into the media surface, the interface and the analysis board. The media surface is programmed with Matlab and it is used for input information, for the creation of test vectors and for the analysis of the results. The different pin types and the pin number belong to the input information. The different pin types are distinguished between input pins, output pins, ground pin, V_{CC} pin and not connected pins. With this information the calculation of test vectors is possible. For the test of finite state machines the input alphabet X is generated by the test vectors. These input alphabets are transmitted through the interface from personal computer to analysis board. The personal computer running the software Matlab and the interface works with the RS232 protocol. Therefore, an adaptation of the data structure occurs on the receiver with the interface. The interface is a component of the analysis board. The analysis board consists of a FPGA and the hardware which is needed for the measurement at the unknown state machine. The FPGA is used for the programming of the interface and for the programming of the test environment. The programming software is called Quartus II by the company Altera [6]. Therefore, the FPGA of the same company is used.

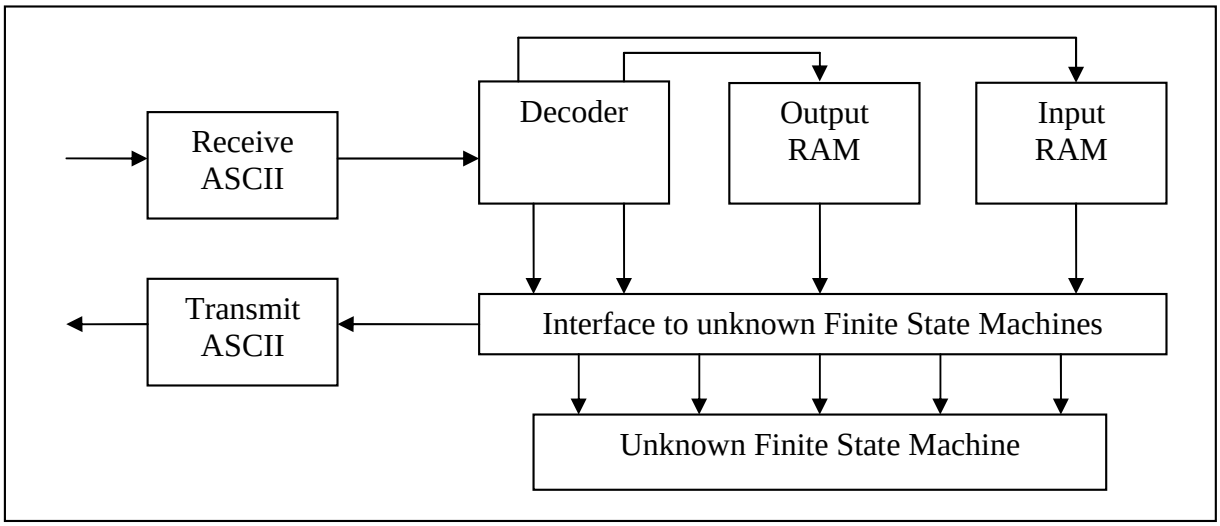

Figure 4: Analysis Board

The complete analysis board is illustrated in Figure 4. Six different VHDL modules are implemented into the FPGA. It includes the interface with a receiver and a transmitter, a decoder, an Input RAM an Output RAM and an interface to the unknown finite state machines. The task of the receiver is to get ASCII frames and after the frames have been received the user information will be passed to the decoder. The user information can be the pin types or the input alphabet. The decoder analyses the information and processes them correspondingly. The input pins are stored in the input RAM and the output pins are stored in the output RAM, because the pin values are required for the complete test sequence. The interface to the unknown finite state machine is the main block for the test of unknown state machine. If the decoder recognises the input alphabet, the information will be handed directly to the finite state machine to be tested. Furthermore, the interface module reads in the initial states. These values will pass to the transmitter interface. The information will be processed and sent using the RS232 protocol to the PC. The RS232 protocol has a baud rate of 9600Bd. Therefore, the unknown circuit can to be tested in real time.

IV Heuristics Results

The test of finite state machines is important to obtain an overview about the internal structure and the functions of the automaton. The tests of combinatorial and sequential finite state machines are carried out by different methods. This is because the process, the internal structure and thus the mathematical behaviour of the combinatorial and sequential finite state machines are different. The behaviour of the combinatorial automaton is more simply then the behaviour of sequential automaton Therefore, it is very difficult to test sequential finite state machines.

a) *Analysis of the Combinatorial Finite State Machine*

The combinatorial finite state machines consist of elementary gates and the input values are not buffered. Therefore, the output values depend directly of the input values $Y(t) = \Phi(X(t))$. This is because the output values only depends on the input values of the same time and not on values of previous times $(t-1)$ or later times $(t+1)$. All possible combinations are used as input values. That means if four inputs are available than 2^4 combinations are used. The following Table 1 shows the possible input combinations and the related output results of any automaton.

t	0	1	2	3
$x_1(t)$	0	1	0	1
$x_2(t)$	0	0	1	1
$y_1(t)$	0	1	1	0
$y_2(t)$	1	1	0	0

Table 1: Input and Output values of the Combinatorial FSM

During the test of the combinatorial finite state machines it has to be noticed, that the same input values always involve to the same initial states. The next step of the system analysis is the setup of the transition function. The transition function is specified by $y_n(t) = \varphi_n(x_1(t), x_2(t),\ldots,x_q(t))$ and can be generated conjunctively or disjunctively. For example the conjunctive transition functions are given by (1).

$$y_1(t) = \left(x_1(t) \vee x_2(t)\right) \wedge \left(\overline{x}_1(t) \vee \overline{x}_2(t)\right)$$
$$y_2(t) = \left(x_1(t) \vee \overline{x}_2(t)\right) \wedge \left(\overline{x}_1(t) \vee \overline{x}_2(t)\right) \quad \textbf{(1)}$$

The disjunctively transition functions are given by (2).

$$y_1(t) = x_1(t) \wedge \overline{x}_2(t) \vee \overline{x}_1(t) \wedge x_2(t)$$
$$y_2(t) = \overline{x}_1(t) \wedge \overline{x}_2(t) \vee x_1(t) \wedge \overline{x}_2(t) \quad \textbf{(2)}$$

The combinatorial finite state machines were tested with the analysis environment and following the results were compared against the mathematically certain values. In conclusion, with the test environment the transition functions of the combinatorial finite state machines can be determined.

b) *Analysis of Sequential Finite State Machines*

The analysis of sequential finite state machines is more complex. Therefore, it has to be divided into a number of steps. The first step is the determination of the number of states. Traditionally, the cross-correlation will be used to determine the time-delay in an integrated circuit. In Figure 5 the cross-correlation will be described to determine the number of internal states *z(t)*, where *x(t)* is the input values and *y(t)* is the output values.

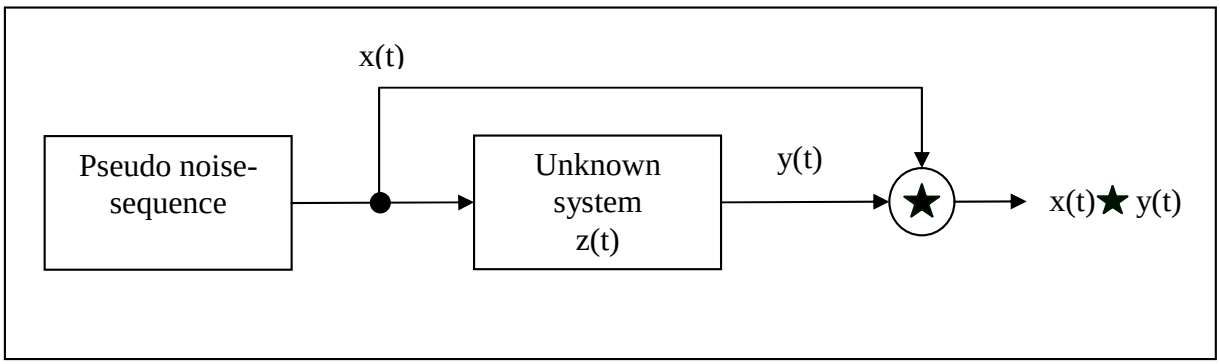

Figure 5: Block Diagram of the Determination of State Numbers

The following conditions have to be considered to use the measurement system above. Firstly, the period of the generated pseudo noise sequence has to be greater than the number of possible states in the unknown system to avoid repetitions of the pseudo noise sequence. Otherwise, this could lead to wrong results during the computation of the cross-correlation. In the moment, the unknown system must be non-recursively and clocked. The pseudo noise sequence is realised using fed back shift registers. The cross-correlation will be realised with

stages of D-latches. However, the number of D-latches must be greater or identical than the number of internal states in the unknown system to evaluate the cross-correlation exactly. This is necessary, because the cross-correlation calculates only the correct values if the signal propagation delay through the internal states and the D-latches will be identical. The outputs of the D-latches are separately linked modulo-2 with *y(t)*. The result of the cross-correlation leads to the number of D-latches. This number is equal to the number of states in the unknown sequential finite state machine. The whole system was implemented and tested into a digital simulator PC software.

V Conclusions

This paper has presented a novel procedure to analyse combinatorial and sequential finite state machines in a non-destructive manner. It has been shown that the transition function of combinational finite state machines can be determined and consequently the design of the combinatorial automaton. This is possible by using the presented analysis environment. Tests have shown that different combinatorial finite state machines can be tested with diverse transition functions. Tests of sequential finite state machines are much more complex. Therefore, the determination of the transition and result functions were carried out in a number of steps. This work presented the first approach for the determination of the states in sequential finite state machines using a cross-correlation method. The tests were carried out using a non-recursively and clocked sequential finite state machines. In every system tested the internal states could be determined. Current research focuses on a further determination of sequential finite state machines using the presented analysis environment.

VI Acknowledgements

The authors would like to thank Prof. Dr. Bundschuh for his helpful suggestions during the project.

VII References

[1] M. Brutscheck, A. Th. Schwarzbacher and St. Becker, “A test environment for analysing unknown integrated circuits,” *7th New Generation Scientist Conference*, Wernigerode, Germany, pp. 223-224, January 2006.

[2] M. Brutscheck, M. Franke, A. Th. Schwarzbacher and St. Becker, “Determination of pin types and minimisation of test vectors in unknown CMOS integrated circuits,” *accepted for Electronic Devices and Systems IMAP CS International Conference*, Brno, Czech Republic, September 2006.

[3] G. Wunsch and H. Schreiber, *Digitale Systeme*, 4th Edition, Springer-Verlag, Berlin, 1993.

[4] G. Wunsch, *Handbuch der Systemtheorie*, Akademie-Verlag, Berlin, 1986.

[5] A. M. Dewey, *Analysis and Design of Digital Systems with VHDL*, PWS Publishing Company, 1997.

[6] Quartus II Web Edition Software Version 6.0, http://www.altera.com.

[7] G. H. Mealy, *A Method for Synthesizing Sequential Circuits*, Bell System Technical Journal, vol. 34, pp. 1045-1079, September 1955.

[8] Edward. F. Moore, *Gedanken-Experiments on Sequential Machines*, Automata Studies pp 129-153 Princeton University Press, Princeton, N. J., 1956.