electroscope

Analysis of Unknown Nonlinear Integrated Circuits

M. Brutscheck[1,2], B. Schmidt[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:
The analysis of unknown integrated circuits (ICs) has become very important over the last decade. In this context different invasive and non-invasive procedures have been developed. However, destructive procedures are not suitable because they always damage the IC under investigation. Non-invasive analysis procedures have the disadvantage that ICs are analysed using very complex and time consuming algorithms. This paper presents the first novel non-invasive procedure to determine nonlinear binary multi-input multi-output (MIMO) ICs only by its input-output behaviour. The algorithm presented in this paper solves unknown ICs by the abstraction of automata theory. The overall identification procedure was simulated and fully tested on IEEE ISCAS benchmark models as well as user defined models of real ICs. This paper will show that for every circuit under test the function has been successfully determined by the proposed identification procedure.

INTRODUCTION

Nowadays, the investigation of unknown CMOS integrated circuits has become very important [1], [2], [3], [4]. Current ICs consist of very complex structures with a great variety of functions and different behaviours. Since these functions are not always known it can be essential to correctly determine their behaviour. This is for example required when an IC is obsolete and it is necessary to find out more about the internal structure of this integrated circuit. Furthermore, it is conceivable to use structures of discontinued ICs in new IC designs or to add new functionality to an existing system. There is a need to divide the overall analysis into different parts to make a structured analysis of these ICs possible. The determination of pin types is the first analysis step which was described by the authors in detail in [5]. This is followed by a preliminary investigation of the IC under test which results in combinatorial, sequential linear or sequential nonlinear behaviour as described in [6]. Here, it was demonstrated that a real IC can be abstracted using the traditional model of automaton [7]. However, a large number of unknown ICs have a nonlinear behaviour. Therefore, this paper will discuss the particular problem of the identification of unknown nonlinear ICs represented by sequential deterministic finite state machines. The overall identification procedure consists of three parts the separation into Moore or Mealy automaton, the preparation algorithm and the main algorithm. Figure 1 shows the main parts of the identification procedure for nonlinear finite state machines in general which are described in detail in the following sections.

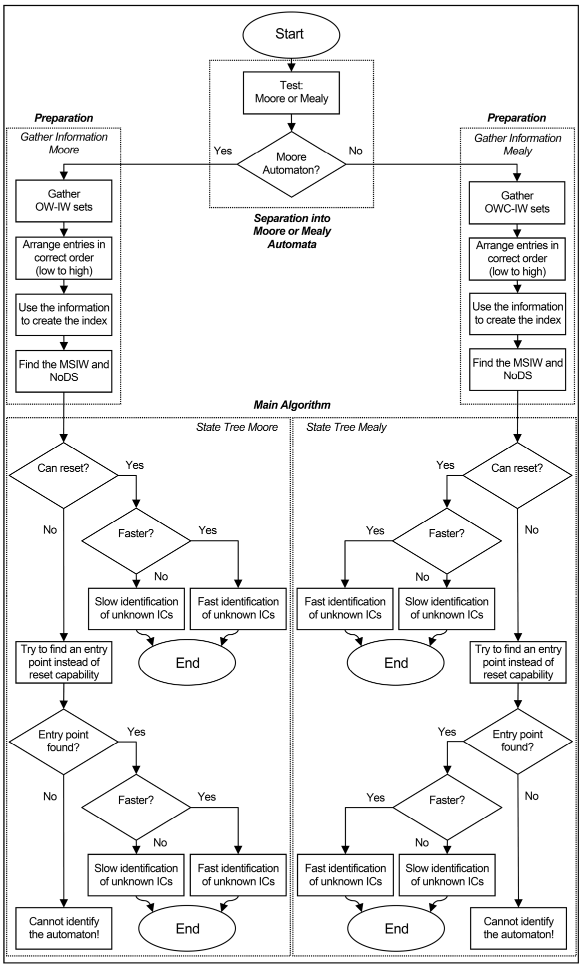


Fig. 1: Identification Procedures in Principle

SEPARATION INTO MOORE OR MEALY AUTOMATA

The separation into Moore or Mealy is the first step of the overall identification procedure. It is an important improvement of the novel procedure to other methods described to simplify the following analysis steps. The different behaviour of Moore and Mealy automata is used for a general classification. The output of a Moore automaton only depends on the internal states. Additionally, if the storages and the inputs are connected by combinatorial circuitry the automaton is of type Mealy. The test run is started while a random input word is applied and then a clock pulse is given to the circuit under test. After this step, the first input word (‘0’) is applied to the automaton and the resulting output word is stored. In the following loop all other input words are applied to the circuit, but no clock pulse is caused. The output words which appear are compared to the stored one and in case of any differences the automaton is classified as a Mealy automaton and the procedure is finished. If all output words are identical the next random step is made until the maximum number of cycles is reached. If the last cycle is executed and no variance in output words was found, the automaton will be considered as a Moore automaton during the following analysis.

PREPARATION ALGORITHM

After the type of the automaton was determined the main algorithm is prepared by several process steps. This preparation is an important step to provide an efficient mode of operation of the analysis. However, the identification procedure must firstly find one initial state of the automaton. In the simplest case the initial state can be reached by a reset pin control. In other ICs the reset can be carried out by disconnection of power supply which is also called power-on-reset. However, an initial state can also be found using suitable process steps if such a reset capability does not exist. After an initial state was found the preparation can be carried out. First, the information about the input-output words (OWs) or input-output word combinations (OWCs) are gathered. These sets are recorded as shown in Equation (1).

$$\begin{aligned} &\{1\} \quad OW\left(t-1\right); IW; OW\left(t\right) \\ &\{2\} \quad OWC\left(t-1\right); IW; OWC\left(t\right) \end{aligned} \qquad (1)$$

If the analysis has identified a Moore automaton set {1} is used. Set {2} is used in the case of a Mealy automaton. The following steps and parts of the algorithm are explained for a Mealy automaton and are carried out in an analogous manner for Moore automata. However, for a Moore automaton only a single output word is processed instead of all output word combinations. The different states are separated relative to their obvious differences in their output behaviour before the algorithm is started. Therefore, random input words are applied and a clock pulse is generated afterwards. In regular intervals a reset is applied to resettable automata to restart the algorithm from well defined initial states. The previous and the following output word are recorded at each step. Additionally, the input word which has caused the step is stored. Repeating combinations of these three values are not saved. However, all differences are collected using the described procedure. As always the first *OW*(*t*-1*)* the change of the *OW*(*t*) to multiple applications of different input words is investigated and is used for the result. If the current output word has two or more following output words when applying the same input word then the number of these output words relates to the minimum number of states which share the first output word. This is valid for deterministic automata. Each information set as illustrated in Equation (1) is checked if it has already occurred. If it has not it is added to the current list. The number of output words found is stored. The number of states, the output words, the input words and the type of the automaton are the basic information. The combinations are checked for their first output word. Due to their order each alteration implies a new output word. If entries exist where at any time input words and output words are equal but the following states are different, then the list is rearranged. The detection of such entries is proof that a minimum of two states exists with the same output word. The input word, which causes most output words following a particular output word, is labelled as most significant input word (MSIW). With the help of the most significant input words it is possible to separate states that have the same output word and the same input word is applied but the following output word is a different one. For instance, if there would be three such entries there will be at least three states related to this output word. After all entries are made the information is interpreted. The total number of finally found differences forms the number of securely distinguishable states. This means, that it is possible to compare two sets of *OW*(*t*-1); *IW*; *OW*(*t*) for a Moore automata or *OWC*(*t*-1); *IW*; *OWC*(*t*) for a Mealy automata there is no difference in the actual sets but only in the result of the investigation. The use of a number of distinguishable states severely reduces the necessary investigation depth of the integrated circuit. Equation (2) shows the calculation of the number of distinguishable states (NoDS).

$$NoDS = \sum_{MSIW} significance \qquad (2)$$

If this number is equal to the real number of states, then the automaton can be identified directly. Otherwise, it is not possible to determine the automaton in only one step. The classification into Moore or Mealy automata as well as the consideration of the number of distinguishable states are important parts of the analysis procedure. The preparation results are used to fully solve such problems by the main algorithm which will be described in the next section.

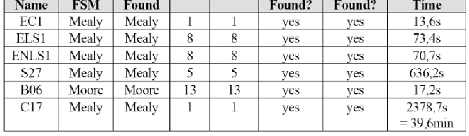
MAIN ALGORITHM

After the initial investigation of the unknown IC the main algorithm is carried out which is the major part of the analysis procedure for nonlinear FSMs. It consists of several blocks and works similar for Moore and Mealy automata. First, the required length of the investigation tree is determined. To determine the state tree length is important to record the state transitions and in order to afterwards correctly identify the unknown IC. After the determination of the tree length the IC under investigation is checked if a reset capability exists or an entry point is determinable. In the next step the algorithm queries the solution type. These are the fast or the slow identification. Basically, both the fast and the slow analysis produced the same results. Normally, the IC under test is analysed by the fast identification. However, in case of insufficient RAM it is not possible to process the algorithm using the fast identification. Therefore, it automatically switches to the slower solution which uses less RAM but requires more evaluation time. The maximum number of states needs not to be known to proceed with the algorithm. In most cases the number of states (NoS) can be calculated using iteration. The initial value can be either given by the user or is determined from the number of distinguishable states. If the real number of states is not known the number of distinguishable state for the initial value can be calculated NoS = NoDS + 2. The added two is based on fact that it is the number of guessed and not identified states. This number can be chosen in a free range. If a higher value will be chosen the likelihood increases to find distinguishable states in each identification cycle which would previously not have been detected. At the same time the investigation complexity increases. If the addend ‘2’ is too high this advantage could be lost and converted into a disadvantage. The iteration to the real number of states is carried out after each cycle of the main algorithm. The idea behind the identification of unknown ICs is similar to the general classification of states using the data prepared in the previous determination of the number of distinguishable states. Using deterministic automata a state has to have the same response at the output caused by the same input word which means the achievement of the same following state. If two states differ in their internal bit combination but always respond equally at the outputs then the algorithm identifies these states as only one state. With this, the automaton is not only identified but also reduced. However, several states can share the same output word. This is valid for all output words. Therefore, it is possible that all output words of two states are identical without any redundancy. For a final distinction of states their state trees are investigated. A state tree contains information of particular output words which are causes by the respective input word applied. As previously described the evaluation of the following output word is an adequate further distinctive feature. Therefore, all following output words (FOWs) of the previous final points are also gathered. This classification is continued until the significance of the trees is sufficient to clearly separate occurring states. Traditional solutions require the knowledge of the maximum number of states [3]. This is an essential disadvantage as the maximum number of states is not available in practice. However, the restriction to resettable automata or automata with a definable entry point provides the possibility to determine the number of states using an iterative approximation without any knowledge of the real number of states. The more precisely the initial value is predefined the faster and safer the solution of the investigated unknown automata is found out. As previously described the initial value can be either given by the user or is derived from the number of distinguishable states. In this case the number of discriminable states represents the minimum number of states. A predefined number of states is added to this number of distinguishable states. From this predefined number it is expected that many other similar states exist, which are not distinguishable by only one step. Here, a preliminary reduction is possible because the states are compared in relation to their current output word as well as their following output word.

RESULTS

In this section the results of the nonlinear identification procedure will be discussed. The theory presented in this paper was verified using both simulation and real hardware tests. The IC models were analysed having unknown as well as known number of internal states using MATLAB [8]. The following tables will show the results of the simulation and the hardware analysis of the nonlinear identification procedure. Furthermore, for each model the result with unknown as well as known number of states is shown. Table 3a first shows the simulation results where NoFS is the number of states found. Moreover, STT is the state transition table and OF represents the output function.

Table 3a: Simulation Results (Unknown NoS)

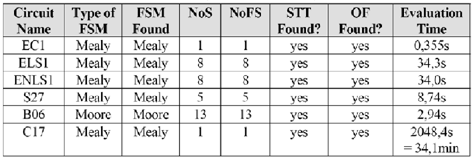


Circuit Name	Type of FSM	FSM Found	NoS	NoFS	STT Found?	OF Found?	Evaluation Time
EC1	Mealy	Mealy	1	1	yes	yes	13,6s
ELS1	Mealy	Mealy	8	8	yes	yes	73,4s
ENLS1	Mealy	Mealy	8	8	yes	yes	70,7s
S27	Mealy	Mealy	5	5	yes	yes	636,2s
B06	Moore	Moore	13	13	yes	yes	17,2s
C17	Mealy	Mealy	1	1	yes	yes	2378,7s = 39,6min

It can be seen from Table 3a that the algorithm found the correct type of all unknown ICs under investigation. Moreover, the correct number of states was always found. Hence, the correct state table as well as the correct output function were in all cases successfully found. In case, that the exact number of states is known Table 3b shows the simulation results. Here, the same implementations were analysed with known number of states instead the initial number of states (NoS) equal to zero.

However, the algorithm introduced was developed to analyse nonlinear FSM.

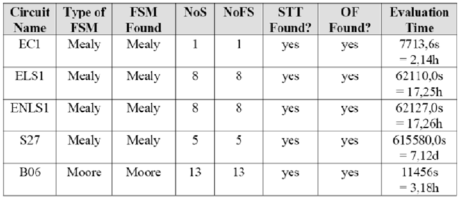
Table 3b: Simulation Results (Known NoS)



Circuit Name	Type of FSM	FSM Found	NoS	NoFS	STT Found?	OF Found?	Evaluation Time
EC1	Mealy	Mealy	1	1	yes	yes	0,355s
ELS1	Mealy	Mealy	8	8	yes	yes	34,3s
ENLS1	Mealy	Mealy	8	8	yes	yes	34,0s
S27	Mealy	Mealy	5	5	yes	yes	8,74s
B06	Moore	Moore	13	13	yes	yes	2,94s
C17	Mealy	Mealy	1	1	yes	yes	2048,4s = 34,1min

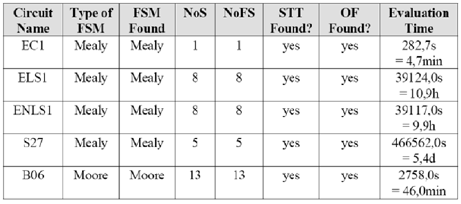
As can be seen from Table 3a and 3b combinatorial as well as linear sequential FSM can also be identified using the novel algorithm. Furthermore, the evaluation time in the right column shows that the simulation is accomplished within less than an hour for even complex circuits. In the next step all IC models were implemented into hardware which are presented in Table 4a and Table 4b.

Table 4a: Hardware Analysis (Unknown NoS)



Circuit Name	Type of FSM	FSM Found	NoS	NoFS	STT Found?	OF Found?	Evaluation Time
EC1	Mealy	Mealy	1	1	yes	yes	7713,6s = 2,14h
ELS1	Mealy	Mealy	8	8	yes	yes	62110,0s = 17,25h
ENLS1	Mealy	Mealy	8	8	yes	yes	62127,0s = 17,26h
S27	Mealy	Mealy	5	5	yes	yes	615580,0s = 7,12d
B06	Moore	Moore	13	13	yes	yes	11456s = 3,18h

Table 4b: Hardware Analysis (Known NoS)



Circuit Name	Type of FSM	FSM Found	NoS	NoFS	STT Found?	OF Found?	Evaluation Time
EC1	Mealy	Mealy	1	1	yes	yes	282,7s = 4,7min
ELS1	Mealy	Mealy	8	8	yes	yes	39124,0s = 10,9h
ENLS1	Mealy	Mealy	8	8	yes	yes	39117,0s = 9,9h
S27	Mealy	Mealy	5	5	yes	yes	466562,0s = 5,4d
B06	Moore	Moore	13	13	yes	yes	2758,0s = 46,0min

As can be seen in Table 4a and Table 4b in each case the novel algorithm found the correct type of unknown IC. The evaluation time in the right column in Table 4a shows that about one week is needed to identify the complex benchmark S27. The other IC models can be determined in less than a day. However, it is even possible to identify the expected state transition table as well as the correct output function. In case that the exact number of states is known the hardware analysis shown as presented in Table 4b was used. From Table 4b it can be seen that in each case the nonlinear detection algorithm found the correct type of the unknown IC.

CONCLUSION

This paper has proposed a novel identification procedure to fully determine nonlinear ICs which possesses several vital improvements compared to traditional procedures. Traditional non-invasive identification procedures always required some prior knowledge of the number of internal states to correctly determine the internal function of the IC under investigation. In practice the number of states is however, mostly unknown. Therefore, a novel iteration procedure was developed by which the algorithm firstly independently approximates the number of states. From this approximation the novel algorithm is then capable to determine the real number of states. Furthermore, an automatic separation into Moore or Mealy automaton was developed, which is based on their different logic structures. This separation was achieved by applying random input words to the unknown system. From the output responses of the automata their behaviour was then determined. The correct operation was verified through the implementation of several IEEE benchmark ICs as well as user defined IC models. The procedure described successfully solves the identification problem of nonlinear finite state machines for the first time. Therefore, in conclusion this paper has presented a novel non-destructive reverse engineering procedure for structured analysis of nonlinear digital unknown CMOS ICs.

REFERENCES

[1] Jarzabek, St., Keam, T. P.: Design of a generic reverse engineering assistant tool, *Proceedings of 2nd Working Conference on Reverse Engineering*, Toronto, Canada, July 14-16, 1995, pp. 61-70.

[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, issue 2, 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., Franke, M., Schwarzbacher, A. Th., Becker, St.: Determination of pin types and minimisation of test vectors in unknown CMOS integrated circuits, *13th* Electronic Devices and Systems IMAP CS International Conference, Brno, Czech Republic, September 2006, pp. 64-69.

[6] Brutscheck, M., Berger St., Franke, M., Schwarzbacher, A. Th., Becker, St.: Classification procedure for finite state machines in unknown CMOS integrated circuits, 9th New Generation Scientist Conference, Köthen, Germany, April 2008, pp. 65-70.

[7] Kohavi, Z.: Switching and finite automata theory, McGraw-Hill Book Company, Second Edition, New York, 1978.

[8] The MathWorks Incorporation [Online]. Available: http://www.mathworks.com, [Accessed: June 2, 2009].