TESTING OF ASYNCHRONOUS SEQUENTIAL SWITCHING CIRCUITS

by A. THAYSE

Abstract

The Boolean differential calculus is used as a mathematical tool to perform a variable transformation which allows to obtain a synchronous model for any asynchronous network. This synchronous model is used to derive behavioural and structural test procedures for asynchronous networks.

1. Introduction

The purpose of this paper is to develop fault-detection test procedures for asynchronous, or unclocked, sequential networks. Prior work on fault analysis of sequential networks has been almost exclusively devoted to synchronous networks, that is network using synchronizing clock pulses. This seems mainly due to the fact that fault-detection experiments are generally performed through the use of homing, distinguishing and synchronizing experiments which are only defined for synchronous networks. It will be shown in this paper that the Boolean differential calculus, introduced by the author in ref. 5, allows us to build a synchronous model for an asynchronous network so that the fault-detection experiments, initially defined for synchronous networks, can be applied to asynchronous networks.

In sec. 2 the mode of operation of the asynchronous sequential network is defined and the adequate differential network model is given. The classical behavioural approach to the testing of synchronous networks is applied in sec. 3 to asynchronous networks through the use of the mathematical tool presented in sec. 2. A structural testing approach is developed in sec. 4, which makes use of prior work by the author relative to variational test procedures for combinatorial networks.

The notations used in this paper are those of refs 5 and 6; it is also assumed that the reader is familiar with the vocabulary relative to Boolean differential calculus, to diagnosis of synchronous networks and to operation modes of asynchronous networks.

2. Boolean differential calculus for asynchronous networks

Let us consider an asynchronous sequential network where the inputs are constrained so as to change only when the memory elements are all in stable conditions. The term fundamental mode is generally used for this mode of
operation. Assume a realization of the network having the following set of excitation equations:

\[ Y_i = f_i(x, y, y_i), \quad i = 1, 2, \ldots, m. \]  

(1)

By differentiating the expression of \( Y_i \) one obtains:

\[
\frac{dY_i}{dt} = \left[ \frac{\partial f_i}{\partial y_i} \right]_{x_i y_i} + \sum_{k=1}^{m} \left[ \frac{\partial f_i}{\partial x_k} \right]_{x_i y_i} \frac{dx_k}{dt}. \tag{2}
\]

The above differential relation has been partitioned into three parts. Part I of relation (2) defines uniquely the variation of the excitation variable \( Y_i \) if it is assumed that any transition leads to a stable state by the intermediary of no more than one unstable state: parts II and III are then identically zero. When the restriction that only one input variable at a time may change is added and when operation is in fundamental mode, the term \emph{normal fundamental mode} is often used. Let \( x^s, y^s \) be a total stable state of a normal-fundamental-mode flow table:

\[ (x^s y^s) = (x^s_1, \ldots, x^s_m, y^s_1, \ldots, y^s_m). \]

(3)

In view of (2) and (3) the excitation equations (1) may be rewritten in the following form:

\[
Y_i = y^s_i + \left( \frac{dY_i}{dt} \right)_{y^s}, \quad i = 1, 2, \ldots, m. \tag{4}
\]

If \( (x^s y^s), \) \( dx_j, \) are considered as independent variables instead of \( x_j, y_i, \) then relations (4) provide us with a synchronous model for any normal-fundamental-mode network. For sake of brevity it has been implicitly assumed that the output vector is included in the state vector, that is \( z \subseteq Y. \) This does not affect the generality of what precedes. The above variable transformation will now be illustrated by means of an example:

\[
Y = x_1 x_2 + (x_1 + x_2') y = z, \tag{5}
\]

\[
dY = [x_2 dx_1 \oplus (x_1 \oplus y) dx_2 \oplus dx_1 dx_2]_i \oplus [(x_2' \oplus dx_2) dy]_{III}.
\]

<table>
<thead>
<tr>
<th>( Y )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
In the excitation table I, the stable states have been underlined. One has \( \{x^s y^s\} = \{000, 100, 111, 101, 001, 010\} \). The next state equation:

\[
Y = y^s \oplus x^s_2 \, dx_1 \oplus (x^s_1 \oplus y^s) \, dx_2 \oplus dx_1 \, dx_2
\]  

(6) allows us to build table II. In table II the column \((dx_1 \, dx_2 = 11)\) may evidently be dropped. This table is clearly a synchronous table. A first general result was thus obtained, that is: any normal-fundamental-mode asynchronous network may be described by means of an asynchronous table when the input and internal variables are taken as independent variables and by means of a synchronous table when the stable total states and the differential of the input variables are taken as independent variables. It is also clear that any synchronous table defines uniquely a normal-fundamental-mode asynchronous table.

### TABLE II

<table>
<thead>
<tr>
<th>(x^s y^s)</th>
<th>(dx_1)</th>
<th>(dx_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
<td>010</td>
</tr>
<tr>
<td>01</td>
<td>101</td>
<td>010</td>
</tr>
<tr>
<td>10</td>
<td>111</td>
<td>010</td>
</tr>
<tr>
<td>11</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>101</td>
<td>001</td>
<td>010</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>010</td>
</tr>
<tr>
<td>010</td>
<td>010</td>
<td>010</td>
</tr>
</tbody>
</table>

Let us now consider a table where more than one intermediary unstable state between two stable states may be found. The flow table is then non-normal mode. If however one still requests that no internal variable may change more than once during any transition a synchronous table may still be derived by taking into account the following observations. A non-normal flow table is one in which a state variable is sensitized by means of another state variable instead of by means of an input variable. If no internal variable may change more than once during any transition, the only difference between a normal-mode network and a non-normal-mode network is that of the duration of the switching time of the network: a dependence on an internal variable instead of a dependence on an input variable manifests itself as a retardation or slow-
up in the otherwise correct signal response. This however does not affect the
input–output behaviour of a network when operating in fundamental mode.
Any non-normal-mode network may thus be substituted by a correspondent
normal-mode network without modifying its input–output behaviour. As
quoted above a synchronous model for that kind of network can then be built.
Let us however note that the substitution of a non-normal-mode network by
a correspondent normal-mode network affects its structure.

3. Behavioural testing of asynchronous networks

The following general scheme will be used to test asynchronous networks:
(1) A synchronous model of the asynchronous network will be obtained by the
intermediate of the mathematical tool presented in the precedent section.
(2) A fault-detection experiment will be constructed for the synchronous model.

Let us first recall the definitions of homing and of distinguishing sequences.

Definition 1

An input sequence $H_0$ is said to be a *homing sequence* if the final state of the
machine can be determined uniquely from the machine's response to $H_0$,
regardless of the initial state.

Definition 2

An input sequence $D_0$ is said to be a *distinguishing sequence* if the output
sequence produced by the machine in response to $D_0$ is different for each initial
state.

Let us briefly recall the fundamentals relative to the testing of synchronous
networks. The network is assumed to be reduced, strongly connected and com-
pletely specified and is available to the experimenter as a *black box*, which means
that he has access to its input and output terminals but cannot inspect the
internal devices and their interconnections. The experiments thus consist of a
set of input sequences and their correspondent output sequences. In addition
it will be assumed that the number of stable states does not increase under fault
conditions and that the network has a distinguishing sequence.

According to Kohavi 4), each fault-detection experiment can be summarized
as follows:
(1) A fault-detection experiment starts with a homing sequence, followed by
the appropriate transfer sequence, so as to manoeuvre the machine to an
initial prespecified state.
(2) The machine is next supplied with an input sequence which causes it to
visit each state and to display its response to the distinguishing sequence.
(3) Finally, the machine is made to go through every state transition, and in
each case the transition is verified by displaying its response to the distinguishing sequence.

Consider the asynchronous network described by means of tables I and II. Its observable output vector is evidently the vector of total states. In order to simplify the notations, we shall designate the binary observable outputs by decimal numbers, i.e. 000 will be designated by 1; 100 by 2; 111 by 3; 101 by 4; 001 by 5; and 010 by 6. Suppose that the fault-detection experiment is designed so that state 1 is the initial state to which it is necessary to transfer the network. To this end we apply the homing sequence \( d_2 = (dx_1 = 0; dx_2 = 1) \) and observe the response.

1. If the response is 010, the machine is in state 6. Apply the transfer sequence \( T(6 \rightarrow 1) = d_2 \) to transfer the network to state 1.
2. If it is 000, the machine is in state 1. (Part 1)
3. If it is 111, the machine is in state 3; apply the transfer sequence \( T(3 \rightarrow 1) = d_1 \cdot d_2 \cdot d_1 \) with \( d_1 = (dx_1 = 1; dx_2 = 0) \).
4. If it is 101, the machine is in state 4; apply the transfer sequence \( T(4 \rightarrow 1) = d_2 \cdot d_1 \cdot d_2 \).

This terminates part 1 of the fault-detection experiment. The sequence \( d_1 \) is a distinguishing experiment for the network. Part 2 of the fault-detection experiment can be achieved as follows:

**Input:**

\[
\begin{array}{cccccccc}
\text{} & d_1 & d_1 & d_2 & d_1 & d_1 & d_2 & d_1 & d_1 \\
\text{State:} & 1 & 2 & 1 & 6 & 3 & 6 & 3 & 4 & 5 & 4 & 5 \\
\end{array}
\]

(Part 2)

The above sequence thus verifies the existence of at least six states by means of the six different underlined output sequences as response to the same input sequence \( d_1 \). The last input sequence \( d_1 \) guarantees that the network terminates in state 5. This terminates part 2. To complete the experiment, it is now necessary to verify each state transition. Up to this point the remaining transitions to be verified are: (2 \( \rightarrow \) 3); (5 \( \rightarrow \) 6); (6 \( \rightarrow \) 1); (4 \( \rightarrow \) 3). Each transition will be checked by applying the suitable transfer sequence followed each by the distinguishing sequence. The experiment is thus achieved as follows:

**Input:**

\[
\begin{array}{cccccccc}
\text{} & d_2 & d_1 & d_1 & d_2 & d_1 & d_2 & d_2 & d_1 \\
\text{State:} & 5 & 6 & 3 & 6 & 1 & 2 & 3 & 6 & 3 & 4 & 3 & 6 \\
\end{array}
\]

(Part 3)

**Transition checking:**

\[
\begin{array}{cccc}
(5 \rightarrow 6) & (6 \rightarrow 1) & (2 \rightarrow 3) & (4 \rightarrow 3) \\
\end{array}
\]
The method outlined above can evidently be applied to any asynchronous network having at least one distinguishing sequence. It must be added that fault-detection experiments for networks which do not have any distinguishing sequence exist but are extremely complicated.

4. Structural testing of asynchronous networks

As in the precedent section it will be assumed that the asynchronous network is described by means of its synchronous model. The structural testing of asynchronous networks will mainly be grounded on a recent paper by Kohavi \(^8\) and on two previous papers by the author \(^6\),\(^9\). The behavioural testing experiment described in the precedent section distinguishes the given m-state network from all possible p-state machines \((p > m)\). Kohavi has shown that the class of machines that the given machine can be transformed into, as a result of stuck-faults type form a subset of the set of all \(p \leq m\)-state machines. An expected result of this is that the structural testing of a network will result in a shorter experiment than the behavioural testing.

---

**Fig. 1. Example of sequential network. Minimum number of state variables is 1.**

**Fig. 2. Example of sequential network. Minimum number of state variables is 2.**
Let us first assume that the sequential network is given by means of its logical scheme. One has first to determine the minimum number of state variables of the network in order to be able to write its excitation equations. A possible way of obtaining these variables was explained in ref. 9. It must be noted that generally several sets of state variables may exist for a given network. Let us consider e.g. the example already treated previously and a realization of which is given in fig. 1. The minimum number of state variables is 1. Each of the arcs 7 or 8 may be associated with the state variable. A somewhat more elaborated example is given in fig. 2. The minimum number of state variables is 2. Each of the following couples of arcs may be associated with the 2-dimension state vector: \((1, 8); (2, 8); (2, 9); (2, 10)\). The determination of the state variables allows us to obtain the excitation equations and consequently the synchronous model for the network. Let us now perform parts 1 and 2 of the behavioural testing experiment described in the precedent section. It is apparent that stuck-faults in the arcs likely to be associated with state variables cause the number of states of the network to decrease. Consequently this type of faults will be detected in part 2 of the experiment in which one identifies the \(m\) distinct states of the network. For example that part of the experiment would detect any stuck-faults in the arcs \((7, 8)\) and \((1, 2, 8, 9, 10)\) of the examples of figs 1 and 2 respectively. Thus the only faults that are still to be checked are the stuck-faults in the remaining arcs of the combinatorial part of the logic network. In principle we could use any one of the techniques described in the classical literature in order to test combinatorial networks. However, the variational diagnosis method described by the author in ref. 6 is particularly well suited in our case. Indeed, since tests for sequential networks are variational tests, the complete test sequence remains homogeneous in variational test patterns. Let us illustrate this by continuing the example of fig. 1. Parts 1 and 2 of the behavioural test procedure remain valid for the structural test procedure, while part 3 is performed by means of the following variational test patterns:

\[
x_2 \, dx_1 = 1, \\
\bar{y} \, x_1 \, dx_2 = 1,
\]

which are equivalent to the following sequence:

Input: 

\[
\begin{array}{c|c|c|c|c|c|c}
    & d_2 & d_1 & d_1 & d_2 & d_1 & d_2 \\
\hline
\text{State:} & 5 & 6 & 3 & 6 & 1 & 2 & 3
\end{array}
\]

\text{(Part 3')} 

Evidently the sequence of part 3' is shorter than that of part 3.

The following remarks must be made concerning the structural test procedure described hereabove. Parts 1 and 2 of the experiment are the same as for the behavioural test procedure. As a consequence each non-normal-mode flow table may be substituted by the correspondent normal-mode flow table. The behavioural test which can be deduced from the normal-mode flow table is
however automatically converted into a structural test by determining the arcs which are tested after parts 1 and 2 of the experiment have been performed. The real structure of the network must evidently be used when performing part 3 of the experiment. This however does not introduce any trouble when using the diagnosis method suggested in ref. 6. Moreover, it must be noted that the fault-detection test procedures, as were described in ref. 6, cannot always directly be applied to the combinatorial part of the sequential network in the way suggested hereabove. Indeed, part of the input vector \( y \) and part of the output vector \( Y \) are generally not available to the experimenter. Thus in general, the procedure of applying a set of input variations and observing the output response is not always applicable. Kohavi \(^8\) suggests the following procedure to overcome this drawback. Although it is impossible to apply directly an input combination to the combinatorial part of the network and observe the output response, it is possible to perform the test indirectly. In other words, since we cannot use inputs \( y \) and outputs \( Y \) in real time, we will force through the inputs \( x \) a desired input combination \( y \) and instead of observing the output response \( Y \) we will identify it by observing its response to a distinguishing sequence. This solves the problem entirely.

5. Conclusion

The fault-detection experiments described in this paper are grounded on a variable transformation which allows us to obtain a synchronous model for an asynchronous network. As a consequence most of the work which was devoted to the testing of synchronous networks can be applied to the diagnosis of faults in asynchronous networks. In particular, a behavioural and a structural fault-detection experiment have been described.

MBLE Research Laboratory

Brussels, November 1971

REFERENCES