NOVEL LOGIC CIRCUITS DYNAMIC PARAMETERS ANALYSIS

Nicolae Galupa

Higher Colleges of Technology – Ras al Khaimah, UAE
ngalupa@hct.ac.ae
Department of Computer Engineering, Technical University Iasi – Romania
nky@cs.tuiasi.ro

ABSTRACT

Combinational logic circuit timing analysis is an important issue that all designers need to address. The present paper presents a simple and compact analysis procedure. We follow the guidelines drawn by previous methods, but we shall define new time-dependent logic variables that help us improve their efficiency. By using the methodology suggested, we shall replace a very laborious technique (pure delay circuit + time constants method) with a simpler procedure that can pinpoint the specific conditions for a logic circuit's anomalous behaviour within a few simple steps. Considering the logic function implemented the methodology presented will require analysis of only a limited number of situations/combinations to determine the presence of an anomalous behaviour. When anomalous behaviour is identified, the methodology provides a clear timing description.

KEYWORDS

Logic Design, Timing, Time Dependant Logic Variables

1. INTRODUCTION

The present work focuses on issues regarding the anomalous functioning of logic circuits. We shall address the static and dynamic hazards defined by J.Beister, E.J. McCluskey, R.F. Tinder and J. Brzozowski [1-3].

At present, we distinguish two analysis methodologies to determine and eventually describe the presence of a hazard in a logic circuit output. The first approach is a purely algebraic one that considers the logic function implemented by a logic circuit and identifies specific algebraic patterns that are responsible for the presence of a hazard. As presented by E.J. McCluskey and R.F. Tinder [2-3], these are $x+\bar{x}$, $x\cdot\bar{x}$ for a static hazard and $x+x\cdot\bar{x}$, $x\left(x+\bar{x}\right)$ for a dynamic hazard, where x is a component of the input vector driving the analysed logic function. This method reaches its goal by algebraically manipulating the logic function and using binary decision graphs, as presented by R. Bryant, S. Ackers, S.M. Nowick, C. Jeong, Berthomieu B. and J. Brzozowski [4-12]. However, this method, will not describe the hazard's evolution (at least not completely), meaning that no timing information will be revealed. Additionally, one can easily note that this method requires a high computational effort.

David C. Wyld et al. (Eds) : CSEN, SIPR, NCWC - 2016
pp. 01–20, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.61001

This paper presents an improvement of the above mentioned method, improvement that allows the designer to determine whether a hazard is present by simply analysing the individual terms present in the logic function's expression, either in SOP form (disjunctive form) or in POS form (conjunctive form). It is my opinion that the improvement presented, if used in conjunction with the classical method, will maintain reliability and will lower the computational effort required

The second approach considers the implementation of a logic function, so analysis will be performed on a completely defined combinational logic circuit (CLC). Basically, we use a pure delay circuit model plus individual in-out path definition, as presented by E.J. McCluskey [2]. Following this procedure, all distinct in → out paths are revealed, and we determine:

- a completely defined delay vector for the circuit,
- the association of each input variable to the path (paths) it crosses towards the output + its specific delay, and
- the ideal logic circuit implementing the logic function.

The method has been described by J. Beister [1], developed by O. Maler and A. Martello [13-15] and exceptionally applied by R.K Brayton [16]. An improvement that considers the inequality between the specific delays for "1"↓"0"and "0"↑"1" transitions has been presented by N. Galupa [17-18]. The rules applied for operating with the gate-specific delays have been presented by K.S. Stevens, R.B. Salah and M. Bogza [19-21]. Please bear in mind that the method presented is not confined only to acyclic combinational circuits, as proven by M. Riedel [22]. The method, also known as the time constants method, allows us to determine the moment of time when the circuit's output has stabilized. However, it will not easily provide information on the behaviour of the logic circuit output prior to stabilization. Should the analysed CLC be used to implement an automaton, its dynamic parameters are critical.

The second section will present a methodology that allows us to easily determine (logic computations only) the dynamic parameters of the circuit output (including active hazard).

Both procedures presented are based on describing the logic variables involved (and, of course, the associated electric signals) with respect to two notions:

- Logic Value – the normal use of a logic variable
- Time – will express the evolution of the variable with respect to time. This is why we shall refer to these variables as time-dependent logic variables (TDLVs).

The present paper is organized as follows:

- Definitions - in this section, we will define the TDLVs
- Properties – in this section, we present and prove the specific properties of TDLVs showing why these variables are useful for the analysis of logic circuit behaviour.

- Analysis procedure for a logic function. We prove, within this section, that the fulfilment of a simple condition will pinpoint the presence of a hazard on the circuit output, thus drastically reducing the computations required.

- Analysis methodology. This section will make use of the TDLVs to determine the hazard generating patterns associated with the logic patterns described by E.J. McCluskey and R.F. Tinder [2-3]

- Finally, a complete example using these methodologies.

2. DEFINITIONS

2.1. Time-Dependent Logic Variables (TDLVs)

Whenever a logic variable applied on a logic gate input changes value, it triggers a process that can be observed on the gate's output connection. However, the gate's output will maintain its previous level for a predetermined period of time – specifically, the gate's propagation time. This situation is bothersome when we expect the gate's output to switch to its complementary value as a result of the input change. Therefore, we shall define a logic variable, denoted τ, to be used for describing the gate's output level evolution as a result of an input variable change, with respect to time.

$$\tau_x=\tau\,(t_x-t) = \begin{cases} 0 \text{ for } t<t_x \\ 1 \text{ for } t\geq t_x \end{cases}$$

One can easily note that τ will help us describe a specific moment of time that is generally associated with a level transition in a signal. Obviously, we also need to be able to describe a time interval, so we shall define the logic variable δ as follows. Let us consider two moments of time ta and tb respecting ta<tb. Under these conditions, δ is defined as:

$$\delta(t_a,t_b)=\delta_{a,b} = \begin{cases} 1 \text{ for } t_a \leq t<t_b \\ 0 \quad \text{otherwise} \end{cases}$$

2.2. Term weight

The weight of a logic term (conjunctive or disjunctive form) is a vector $w=(w_1,w_2,\ldots w_{n-1},w_n)$, where $w_k \{0,1\}$. If $w_k=0$, then the k component of the term is complemented; otherwise, it is in its direct form.

Example: w=0101 → the logic term is $\overline{a_1}\cdot a_2\cdot\overline{a_3}\cdot a_4$ (conjunctive form) or $\overline{a_1}+a_2+\overline{a_3}+a_4$ (disjunctive form)

3. PROPERTIES

3.1. Properties of τ and δ variables

Let us consider three ordered time stamps, t_1, t_2, and t_3 respecting $t_1<t_2<t_3$. We shall consider all possible logic terms that can be defined using three independent logic variables (both conjunctive and disjunctive forms) that have these three time stamps associated as switching moments. The reduced terms that result when τ and δ logic variables are used to express logic terms are presented in table 1. Proof has been provided by Galupa [18].

Table 1 τ,δ MINTERM AND MAXTERM PROPERTIES

Weight	Term
000	$\overline{\tau_1}\cdot\overline{\tau_2}\cdot\overline{\tau_3}=\overline{\tau_1}$ $\overline{\tau_1}+\overline{\tau_2}+\overline{\tau_3}=\overline{\tau_3}$
001	$\overline{\tau_1}\cdot\overline{\tau_2}\cdot\tau_3=0$ $\overline{\tau_1}+\overline{\tau_2}+\tau_3=\overline{\delta_{2,3}}$
010	$\overline{\tau_1}\cdot\tau_2\cdot\overline{\tau_3}=0$ $\overline{\tau_1}+\tau_2+\overline{\tau_3}=1$
011	$\overline{\tau_1}\cdot\tau_2\cdot\tau_3=0$ $\overline{\tau_1}+\tau_2+\tau_3=\overline{\delta_{1.2}}$
100	$\tau_1\cdot\overline{\tau_2}\cdot\overline{\tau_3}=\delta_{1,2}$ $\tau_1+\overline{\tau_2}+\overline{\tau_3}=1$
101	$\tau_1\cdot\overline{\tau_2}\cdot\tau_3=0$ $\tau_1+\overline{\tau_2}+\tau_3=1$
110	$\tau_1\cdot\tau_2\cdot\overline{\tau_3}=\delta_{2,3}$ $\tau_1+\tau_2+\overline{\tau_3}=1$
111	$\tau_1\cdot\tau_2\cdot\tau_3=\tau_3$ $\tau_1+\tau_2+\tau_3=\tau_1$

Clearly, if we consider n distinct time stamps t_1, t_2,…, t_n respecting $t_1<t_2<\ \ldots<t_{n-1}<t_n$ and following the same logic path as above, we will reach the reduced expressions for the n-component logic terms, as presented in table. II.

Table 2. GENERAL τ,δ MINTERM AND MAXTERM PROPERTIES

Weight		
000…000	$\overline{\tau_1}\cdot\overline{\tau_2}\cdot\overline{\tau_3}\cdot\ldots\cdot\overline{\tau_{n-2}}\cdot\overline{\tau_{n-1}}\cdot\overline{\tau_n}=$ $\overline{\tau_1}+\overline{\tau_2}+\overline{\tau_3}+\ldots+\overline{\tau_{n-2}}+\overline{\tau_{n-1}}+\overline{\tau_n}=$	$\overline{\tau_1}$ $\overline{\tau_n}$
000…001	$\overline{\tau_1}\cdot\overline{\tau_2}\cdot\overline{\tau_3}\cdot\ldots\cdot\overline{\tau_{n-2}}\cdot\overline{\tau_{n-1}}\cdot\tau_n=$ $\overline{\tau_1}+\overline{\tau_2}+\overline{\tau_3}+\ldots+\overline{\tau_{n-2}}+\overline{\tau_{n-1}}+\tau_n=$	0 $\overline{\delta_{n-1,n}}$
000…010	$\overline{\tau_1}\cdot\overline{\tau_2}\cdot\overline{\tau_3}\cdot\ldots\cdot\overline{\tau_{n-2}}\cdot\tau_{n-1}\cdot\overline{\tau_n}=$ $\overline{\tau_1}+\overline{\tau_2}+\overline{\tau_3}+\ldots+\overline{\tau_{n-2}}+\tau_{n-1}+\overline{\tau_n}=$	0 1
…	…	…

00...01...11	$\overline{\tau_1}\cdot\ldots\cdot\overline{\tau_{i-1}}\cdot\tau_i\cdot\ldots\cdot\tau_n =$ $\overline{\tau_1}+\ldots+\overline{\tau_{i-1}}+\tau_i+\ldots+\tau_n =$	0 $\overline{\delta_{i-1,i}}$
…	…	…
011…111	$\overline{\tau_1}\cdot\tau_2\cdot\tau_3\cdot\ldots\cdot\tau_{n-2}\cdot\tau_{n-1}\cdot\tau_n =$ $\overline{\tau_1}+\tau_2+\tau_3+\ldots+\tau_{n-2}+\tau_{n-1}+\tau_n =$	0 $\overline{\delta_{1,2}}$
100…000	$\tau_1\cdot\overline{\tau_2}\cdot\overline{\tau_3}\cdot\ldots\cdot\overline{\tau_{n-2}}\cdot\overline{\tau_{n-1}}\cdot\overline{\tau_n} =$ $\tau_1+\overline{\tau_2}+\overline{\tau_3}+\ldots+\overline{\tau_{n-2}}+\overline{\tau_{n-1}}+\overline{\tau_n} =$	$\delta_{1,2}$ 1
…	…	…
111…110	$\tau_1\cdot\tau_2\cdot\tau_3\cdot\ldots\cdot\tau_{n-2}\cdot\tau_{n-1}\cdot\overline{\tau_n} =$ $\tau_1+\tau_2+\tau_3+\ldots+\tau_{n-2}+\tau_{n-1}+\overline{\tau_n} =$	$\delta_{n-1,n}$ 1
111…111	$\tau_1\cdot\tau_2\cdot\tau_3\cdot\ldots\cdot\tau_{n-2}\cdot\tau_{n-1}\cdot\tau_n =$ $\tau_1+\tau_2+\tau_3+\ldots+\tau_{n-2}+\tau_{n-1}+\tau_n =$	τ_n τ_1

Please observe that in table 2, only the extremities (meaning the border time stamps, expressed by the terms associated with weights w=000…000 and w=111…111) are coherent when operating with τ. Therefore, we can safely state that when operating the terms present in a function equation with respect to time (meaning, using τ and δ), only $\tau_1=\tau(t-t_1)$ and $\tau_n=\tau(t-t_n)$ will be present in the final expression (according to the specific function's equation).

On the other hand, whenever the term in question is characterized by an ordered weight (i.e., w=000…01…111 or w=111…10…000), that term will contribute to the final expression only by pinpointing a time interval (t_{k-1}, t_k) by means of δ.

The other terms present in the analysed logic expression are either logic "1" (disjunctive) or "0" (conjunctive).

4. ANALYSIS PROCEDURE

Let there be a logic function y=f(a,b,c,d,…). According to the time constants method (Beister[1], McCluskey[2]), we define the individual in-out pathways by virtually replicating all gates characterized by a fan-out larger than one, replace the gates with their equivalent ideal gates (zero delay) plus their specific propagation time (delay operator) and propagate all delay operators from output → input. Finally, we will reach a structure that will present all individual in-out pathways and their specific delays followed by an ideal logic circuit.

Let us consider that logic variable a (input vector component) crosses n distinct paths towards the output, so that primary input variable a generates n distinct secondary variables $(a_1,a_2,\ldots,a_n)$ characterized by n distinct delays $(t_1, t_2,\ldots,t_n)$. We assume that the secondary input vector has been organized so all specific path delays respect $t_1<t_2<\ldots<t_{n-1}<t_n$.

Therefore, a change in input variable a will generate a sequence of changes in the secondary input variables $(a_1, a_2,\ldots,a_n)$ ordered and spaced in time according to $(t_1, t_2,\ldots,t_n)$. Subsequently, because the secondary input vector is driving an instantaneous ideal logic circuit, we shall observe a sequence of transitions in the circuit's output.

Considering the definition of τ, we can write:

$$a_x(t)=a_{xinitial}\oplus\tau(t_x-t) \quad (4.1.)$$

where $a_{xinitial}$ is the initial value from which a_x evolves. Note that $a_x(t)$ will maintain $a_{xinitial}$ level until we reach t_x time stamp and switches to the complementary value afterwards.

The function's expression becomes:

$$y(t)=f[a_1(t),a_2(t),...,a_n(t)]=f[a\oplus\tau_1(t_1-t),a\oplus\tau_2(t_2-t),...,a\oplus\tau_n(t_n-t)] \quad (4.2.)$$

Ultimately, a logic function can be expressed in a conjunctive or disjunctive form, therefore, considering y(t) presented by eq. 4.2 and the properties presented in table 2, we conclude that only a strictly limited and well determined number of terms will be present.

To present how this works, we shall first consider a particular case – only three secondary variables for the primary input variable considered, a_1, a_2, and a_3, characterized by t_1, t_2, and t_3, respecting $t_1< t_2< t_3$. Afterwards, we shall generalize to n secondary variables.

The function's expression becomes:

$$y(t)=\alpha_{000}\overline{a_1(t)}\cdot\overline{a_2(t)}\cdot\overline{a_3(t)}+\alpha_{001}\overline{a_1(t)}\cdot\overline{a_2(t)}\cdot a_3(t)+...+\alpha_{111}a_1(t)\cdot a_2(t)\cdot a_3(t) \quad (4.3.)$$

$$y(t)=[\overline{\alpha_{000}}+\overline{a_1(t)}+\overline{a_2(t)}+\overline{a_3(t)}]\cdot[\overline{\alpha_{001}}+\overline{a_1(t)}+\overline{a_2(t)}+a_3(t)]\cdot...\cdot[\overline{\alpha_{111}}+a_1(t)+a_2(t)+a_3(t)] \quad (4.4.)$$

where, $\alpha_{w,y,z}$ $\{0,1\}$ and is defined as follows:

- $\alpha_{w,y,z}= 0\rightarrow$term characterised by weight wyz is not present;
- $\alpha_{w,y,z}=1\rightarrow$term characterised by weight wyz is present.

For further computations we'll consider the function's expression presented by eq. 4.3. Computations for the other form (eq. 4.4.) are similar.

$$y(t)=\alpha_{000}\overline{a\oplus\tau_1}\cdot\overline{a\oplus\tau_2}\cdot\overline{a\oplus\tau_3(t)}+\alpha_{001}\overline{a\oplus\tau_1}\cdot\overline{a\oplus\tau_2}\cdot a\oplus\tau_3+...+\alpha_{111}a\oplus\tau_1\cdot a\oplus\tau_2\cdot a\oplus\tau_3 \quad (4.5.)$$

with a being the initial value for the variable switching during the process. Now, we shall operate each term and reduce it according to the properties presented for τ.

$$w=000\rightarrow\overline{a\oplus\tau_1}\cdot\overline{a\oplus\tau_2}\cdot\overline{a\oplus\tau_3}=(a\tau_1+\bar{a}\bar{\tau}_1)(a\tau_2+\bar{a}\bar{\tau}_2)(a\tau_3+\bar{a}\bar{\tau}_3)=a\tau_1\tau_2\tau_3+\bar{a}\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3=a\tau_3+\bar{a}\bar{\tau}_1 \quad (4.6.)$$

$$w=001\rightarrow\overline{a\oplus\tau_1}\cdot\overline{a\oplus\tau_2}\cdot a\oplus\tau_3=(a\tau_1+\bar{a}\bar{\tau}_1)(a\tau_2+\bar{a}\bar{\tau}_2)(\bar{a}\tau_3+a\bar{\tau}_3)=a\tau_1\tau_2\bar{\tau}_3+\bar{a}\bar{\tau}_1\bar{\tau}_2\tau_3=a\delta_{2,3}+\bar{a}0=a\delta_{2,3} \quad (4.7.)$$

……..

$$w=111\rightarrow(a\oplus\tau_1)\cdot(a\oplus\tau_2)\cdot(a\oplus\tau_3)=(\bar{a}\tau_1+a\bar{\tau}_1)(\bar{a}\tau_2+a\bar{\tau}_2)(\bar{a}\tau_3+a\bar{\tau}_3)=a\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3+\bar{a}\tau_1\tau_2\tau_3=a\bar{\tau}_1+\bar{a}\tau_3 \quad (4.8.)$$

Following the same procedure and using the properties listed above, we find:

Table 3. REDUCED TIME-DEPENDENT SECONDARY TERMS

weight		
000	$\overline{a\oplus\tau_1}\cdot\overline{a\oplus\tau_2}\cdot\overline{a\oplus\tau_3}=a\tau_1\tau_2\tau_3+\bar{a}\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3=$	$a\tau_3+\bar{a}\bar{\tau}_1$
001	$\overline{a\oplus\tau_1}\cdot\overline{a\oplus\tau_2}\cdot a\oplus\tau=a\tau_1\tau_2\bar{\tau}_3+\bar{a}\bar{\tau}_1\bar{\tau}_2\tau_3=$	$a\delta_{2,3}$
010	$\overline{a\oplus\tau_1}\cdot a\oplus\tau_2\cdot\overline{a\oplus\tau_3}==a\tau_1\bar{\tau}_2\tau_3+\bar{a}\bar{\tau}_1\tau_2\bar{\tau}_3=$	0
011	$\overline{a\oplus\tau_1}\cdot a\oplus\tau_2\cdot a\oplus\tau_3=a\tau_1\bar{\tau}_2\bar{\tau}_3+\bar{a}\bar{\tau}_1\tau_2\tau_3=$	$a\delta_{1,2}$
100	$a\oplus\tau_1\cdot\overline{a\oplus\tau_2}\cdot\overline{a\oplus\tau_3}=a\bar{\tau}_1\tau_2\tau_3+\bar{a}\tau_1\bar{\tau}_2\bar{\tau}_3=$	$\bar{a}\delta_{1,2}$
101	$a\oplus\tau_1\cdot\overline{a\oplus\tau_2}\cdot a\oplus\tau_3=a\bar{\tau}_1\tau_2\bar{\tau}_3+\bar{a}\tau_1\bar{\tau}_2\tau_3=$	0
110	$a\oplus\tau_1\cdot a\oplus\tau_2\cdot\overline{a\oplus\tau_3}=a\bar{\tau}_1\bar{\tau}_2\tau_3+\bar{a}\tau_1\tau_2\bar{\tau}_3=$	$\bar{a}\delta_{2,3}$
111	$a\oplus\tau_1\cdot a\oplus\tau_2\cdot a\oplus\tau_3=a\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3+\bar{a}\tau_1\tau_2\tau_3=$	$a\bar{\tau}_1+\bar{a}\tau_3$

Therefore, the function's expression becomes:

$$y(t)=\alpha_{000}[a\tau_3+\bar{a}\bar{\tau}_1]+\alpha_{001}a\delta_{2,3}+\alpha_{011}a\delta_{1,2}+\alpha_{100}\bar{a}\delta_{1,2}+\alpha_{110}\bar{a}\delta_{2,3}+\alpha_{111}[a\bar{\tau}_1+\bar{a}\tau_3] \qquad (4.9.)$$

Note that only the border terms influence the final expression ($a\cdot\tau_3+\bar{a}\cdot\bar{\tau}_1$ if α_{000}=1, meaning a term with weight w=000 - $\bar{a}_1\cdot\bar{a}_2\cdot\bar{a}_3$ - is present, and / or $a\cdot\bar{\tau}_1+\bar{a}\cdot\tau_3$ if α_{111}=1, meaning a term with weight w=111 - $a_1\cdot a_2\cdot a_3$ - is present).

Considering that the time stamps are ordered $t_1<t_2<t_3$, the individual time intervals (t_1,t_2) and/or (t_2,t_3) will be pinpointed by $\delta_{1,2}$ and $\delta_{2,3}$ if α_{011} and α_{001}, respectively, are logic 1 and / or $\overline{\delta_{1,2}}$ and $\overline{\delta_{2,3}}$ if α_{011} and α_{001}, respectively, are logic 1.

Please observe that in the final expression, we have succeeded in significantly decreasing the number of terms involved. Also note that considering eq. 4.9, we can trace the output's behaviour with respect to time.

First, we identify the influence that each member of eq. 4.9. has on the overall output presented in Table. 4.

Table. 4. OUTPUT TRACE FOR VARIABLE a

	t_{init}	t_1	t_2	t_3	t_{final}
1. α_{000}=1⇒y(t)=	$\bar{a}$	0	0	a	
2. α_{001}=1⇒y(t)=	0	0	a	0	
3. α_{011}=1⇒y(t)=	0	a	0	0	
4. α_{100}=1⇒y(t)=	0	$\bar{a}$	0	0	
5. α_{110}=1⇒y(t)=	0	0	$\bar{a}$	0	
6. α_{111}=1⇒y(t)=	a	0	0	$\bar{a}$	

- The border terms (w=111 and w=000 – lines 1 and 6) will never generate a hazard by themselves, independent of variable a's initial value.

	t_{init}	t_1	t_2	t_3	t_{final}
α_{000}=1and a=0⇒y(t)=	1	0	0	0	
α_{000}=1and a=1⇒y(t)=	0	0	0	1	
α_{111}=1and a=0⇒y(t)=	0	0	0	1	
α_{111}=1and a=1⇒y(t)=	1	0	0	0	

- The terms characterized by a uniform ordered weight will generate a static hazard.

	t_{init}	t_1	t_2	t_3	
α_{001}=1and a=1⇒y(t)=	0	0	1	0	Static hazard for "1"↓"0"transition
α_{011}=1and a=1⇒y(t)=	0	1	0	0	Static hazard for "1"↓"0"transition
α_{100}=1and a=0⇒y(t)=	0	1	0	0	Static hazard for "0"↑"1"transition
α_{110}=1and a=0⇒y(t)=	0	0	1	0	Static hazard for "0"↑"1"transition

- A dynamic hazard will be present if a combination of border terms and ordered weight terms is encountered

	t_{init}	t_1	t_2	t_3	
α_{000}=1and a=0⇒y(t)=	1	0	0	0	
α_{110}=1and a=0⇒y(t)=	0	0	1	0	
Overall behavior	**1**	**0**	**1**	**0**	**Dynamic hazard for "0"↑"1" transition**
α_{000}=1and a=1⇒y(t)=	0	0	0	1	
α_{011}=1and a=1⇒y(t)=	0	1	0	0	
Overall behavior	**0**	**1**	**0**	**1**	**Dynamic hazard for "1"↓"0" transition**
α_{111}=1and a=0⇒y(t)=	0	0	0	1	
α_{100}=1and a=0⇒y(t)=	0	1	0	0	
Overall behavior	**0**	**1**	**0**	**1**	**Dynamic hazard for "0"↑"1" transition**
α_{111}=1and a=1⇒y(t)=	1	0	0	0	
α_{001}=1and a=1⇒y(t)=	0	0	1	0	
Overall behavior	**1**	**0**	**1**	**0**	**Dynamic hazard for "1"↓"0" transition**

Additionally, please observe that the previous analysis presents us with a way to mask an existing dynamic hazard. If we encounter such a situation, all we should do is activate the appropriate term (if algebraically possible) that will fill in the glitch forming the dynamic hazard. Possible cases are listed below:

Table. 5. HAZARD MASKING CASES

CASE 1	t_{init}	t_1	t_2	t_3	
α_{000}=1and a=0⇒y(t)=	1	0	0	0	Dynamic hazard for a→"0"↑"1"
α_{110}=1and a=0⇒y(t)=	1	0	1	0	
Initial behavior	**1**	**0**	**1**	**0**	
α_{100}=1and a=0⇒y(t)=	0	1	0	0	Added term (if possible)
Final overall behavior	**1**	**1**	**1**	**0**	Hazard masked

CASE 2	t_{init}	t_1	t_2	t_3	
α_{000}=1and a=1⇒y(t)=	0	0	0	1	Dynamic hazard for a→"1"↓ "0"
α_{011}=1and a=1⇒y(t)=	0	1	0	0	
Initial behavior	**0**	**1**	**0**	**1**	
α_{001}=1and a=1⇒y(t)=	0	0	1	0	Added term (if possible)
Final overall behavior	**0**	**1**	**1**	**1**	Hazard masked

CASE 3	t_{init}	t_1	t_2	t_3	
α_{111}=1and a=0⇒y(t)=	0	0	0	1	Dynamic hazard for a→"0"↑"1"
α_{100}=1and a=0⇒y(t)=	0	1	0	0	
Initial behavior	**0**	**1**	**0**	**1**	
α_{110}=1and a=0⇒y(t)=	0	0	1	0	Added term (if possible)
Final overall behavior	**0**	**1**	**1**	**1**	Hazard masked

CASE 4	t_{init}	t_1	t_2	t_3	
α_{111}=1and a=1⇒y(t)=	1	0	0	0	Dynamic hazard for a→"1"↓ "0"
α_{001}=1and a=1⇒y(t)=	0	0	1	0	
Initial behavior	**1**	**0**	**1**	**0**	
α_{011}=1and a=1⇒y(t)=	0	1	0	0	Added term (if possible)
Final overall behavior	**1**	**1**	**1**	**0**	Hazard masked

The instrument presented above works when dealing with minterms / maxterms. Obviously, this is not always the case. However, whenever we address a term that is not complete (meaning that it lacks input components), we shall analyse whether its weight is ordered. If not, no further inquiries are required, as the term will not generate anomalous behaviour. On the other hand, if the weight is ordered, the term should be expanded to canonical form (without altering the function's truth value) and an analysis performed.

In case of an n-component secondary input vector (a_1, a_2,…,a_n) generated by primary input variable a, characterized by t_1, t_2, …, t_n ordered timestamps ($t_1<t_2<\ldots<t_n$), the function's expression becomes:

$$y(t)=\alpha_{000\ldots000}[a\tau_n+\bar{a}\bar{\tau_1}]+\alpha_{000\ldots001}\delta_{n-1,n}+\ldots+\alpha_{000\ldots01\ldots111}a\delta_{i-1,i}+\ldots+\alpha_{011\ldots11\ldots111}a\delta_{1,2}+$$
$$+\alpha_{100\ldots000}\bar{a}\delta_{1,2}+\ldots+\alpha_{111\ldots10\ldots000}\bar{a}\delta_{j-1,j}+\ldots+\alpha_{111\ldots110}\bar{a}\delta_{n-1,n}+\alpha_{111\ldots110}[a\bar{\tau_1}+\bar{a}\tau_n] \quad (4.10.)$$

Therefore, the function's output trace is presented by Table.6.

Once again, note that we have succeeded in significantly decreasing the number of terms involved. Just identifying the presence of specific terms (border and/or weight ordered) will be enough to conclude whether anomalous behaviour is present. Furthermore, in particular situations, we can also pinpoint the methodology for masking the anomalous behaviour, should this be our goal.

Table. 6. GENERAL OUTPUT TRACE FOR VARIABLE a

	t_{init}	t_1	t_2	t_3	…	t_{n-2}	t_{n-1}	t_n t_{final}
$\alpha_{000\ldots000}=1\rightarrow y(t)=$	$\bar{a}$	0	0	0	…	0	0	a
$\alpha_{000\ldots001}=1\rightarrow y(t)=$	0	0	0	0	…	0	a	0
$\alpha_{000\ldots011}=1\rightarrow y(t)=$	0	0	0	0	…	a	0	0
……								
$\alpha_{001\ldots111}=1\rightarrow y(t)=$	0	0	a	0	…	0	0	0
$\alpha_{011\ldots111}=1\rightarrow y(t)=$	0	a	0	0	…	0	0	0
$\alpha_{100\ldots000}=1\rightarrow y(t)=$	0	$\bar{a}$	0	0	…	0	0	0
$\alpha_{110\ldots000}=1\rightarrow y(t)=$	0	0	$\bar{a}$	0	…	0	0	0
……								
$\alpha_{111\ldots100}=1\rightarrow y(t)=$	0	0	0	0	…	$\bar{a}$	0	0
$\alpha_{111\ldots110}=1\rightarrow y(t)=$	0	0	0	0	…	0	$\bar{a}$	0
$\alpha_{111\ldots111}=1\rightarrow y(t)=$	a	0	0	0	…	0	0	$\bar{a}$

The singular presence of a term with the weight ordered is a sufficient condition for the presence of a static hazard.

5. CIRCUIT ANALYSIS METHODOLOGY

The previous paragraph presented how to determine whether a logic function output exhibits anomalous behaviour. Another approach is required, if our goal is to determine the occurrence of an anomalous pulse(s) on a logic circuit's output and eventually determine its dynamic parameters.

Basically, we shall determine the secondary input vector and its associated delays vector using already established methods (J. Beister[1], McCluskey[2]). At this point, we will use the TDLVs (τ and δ) to express all variables considering the time variable. We reduce the logic function's expression to a minimal one (Boolean rules), and we will attempt to identify the hazard generating patterns (static or dynamic).

Notation rules:

- delay for NOT gate → t_N / and associated $\tau_N=\tau\ (t_N\text{-}t)$
- delay for AND gate → t_A / and associated $\tau_A=\tau\ (t_A\text{-}t)$
- delay for OR gate → t_O / and associated $\tau_O=\tau\ (t_O\text{-}t)$
- delay for NAND gate → t_{NA} / and associated $\tau_{NA}=\tau\ (t_{NA}\text{-}t)$
- delay for NOR logic gate → tNO / and associated $\tau_{NO}=\tau\ (t_{NO}\text{-}t)$
- $t_N+t_O = {}_{tN+O}$ / and associated $\tau_{N+O}=\tau\ (t_{N+O}\text{-}t)$, so.on.

To determine the time-dependent hazard generating patterns, we shall start from the circuit patterns as presented by E.J. McCluskey and R.F. Tinder [2-3].

5.1. Static - Case 1 Algebraic description → $y = x \cdot \bar{x}$

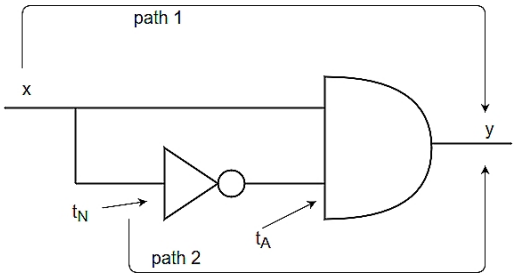

Fig.1.Hazard generating circuit – case 1

Delays: → path 1- t_A / associated $\tau_A=\tau(t_A-t)$; non inverting
→ path 2- $t_N+t_A=t_{N+A}$ / associated $\tau_{N+A}=\tau(t_{N+A}-t)$; inverting

$$y(t) = (x \oplus \tau_A) \cdot (\bar{x} \oplus \tau_{N+A}) = (x\bar{\tau}_A + \bar{x}\tau_A)(\bar{x}\overline{\tau_{N+A}} + x\tau_{N+A}) = x\bar{\tau}_A\tau_{N+A} + \bar{x}\tau_A\overline{\tau_{N+A}} \quad (5.1.)$$

Considering that $t_A<t_N + t_A=t_{N+A}$ according to table 1, we know that $\bar{\tau}_A \cdot \tau_{N+A} = 0$ and $\tau_A \cdot \overline{\tau_{N+A}} = \delta_{A,N+A}$, thus rendering the above equation as follows:

$$y(t) = \bar{x} \cdot \delta_{A,N+A} \quad (5.2.)$$

Eq. 5.2. provides us with the static "0" hazard pattern.

One can easily observe that x=0 → $y(t) = \delta_{A,N+A}$ and x=1 → *y(t)=0*, meaning that for transition x "0"↑"1", the circuit's output will exhibit a pulse valued "1" beginning at t_A and ending t_{N+A}, while for transition x "1"↓"0", the circuit's output will maintain a constant "0" value.

5.2. Static - Case 2 Algebraic description → $y = x + \bar{x}$

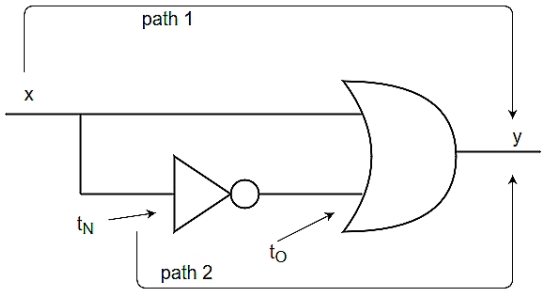

Fig.2.Hazard generating circuit – case 2

Delays: →path 1- t_O / associated $\tau_O=\tau(t_O-t)$; non inverting
→path 2- $t_N + t_O = t_{N+O}$ / associated $\tau_{N+O}=\tau(t_{N+O}-t)$; inverting

$$y(t) = (x \oplus \tau_O) + (\bar{x} + \tau_{N+O}) = x\bar{\tau}_O + \bar{x}\tau_O + x\tau_{N+O} + \bar{x}\overline{\tau_{N+O}} = x(\bar{\tau}_O + \tau_{N+O}) + \bar{x}(\tau_O + \overline{\tau_{N+O}}) \quad (5.3.)$$

Considering that $t_O<t_N+t_O=t_{N+O}$ according to table 1, we know that $\overline{\tau_O}+\tau_{N+O}=\overline{\delta_{O,N+O}}$ and $\overline{\tau_O}+\tau_{N+O}=1$, thus rendering the above equation as follows:

$$y(t)=\bar{x}+x\cdot\overline{\delta_{O,N+O}} \tag{5.4.}$$

Eq. 5.4. provides us with the static "1" hazard pattern.

One can easily observe that x=0→$y(t)=1$ and x=1→ $y(t)=\overline{\delta_{O,N+O}}$, meaning that for transition x "1"↓"0", the circuit's output will exhibit a pulse valued "0" beginning at t_O and ending at t_{N+O}, while for transition x "0"↑"1", the circuit's output will maintain a constant "1" value.

5.3. Dynamic - Case 3 Algebraic description → $y=x+x\cdot\bar{x}$

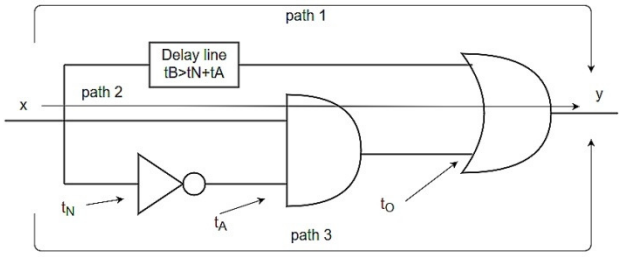

Fig.3.Hazard generating circuit – case 3

Delays: →path1– t_B+t_O / associated $\tau_{B+O}=\tau(t_{B+O}-t)$; non inverting
→path2- $t_A+t_O=t_{A+O}$ / associated $\tau_{A+O}=\tau(t_{A+O}-t)$; non inverting
→path3-$t_N+t_A+t_O=t_{N+A+O}$/associated$\tau_{N+A+O}=\tau(t_{N+A+O}-t)$; inverting

$$y(t)=(x\oplus\tau_{B+O})+(x\oplus\tau_{A+O})\cdot(\bar{x}+\tau_{N+A+O})=(x\overline{\tau_{B+O}}+\bar{x}\tau_{B+O})+(x\overline{\tau_{A+O}}+\bar{x}\tau_{A+O})\cdot(x\tau_{N+A+O}+\bar{x}\overline{\tau_{N+A+O}})=$$
$$=x(\overline{\tau_{B+O}}+\overline{\tau_{A+O}}\tau_{N+A+O})+\bar{x}(\tau_{B+O}+\tau_{A+O}\overline{\tau_{N+A+O}}) \tag{5.5.}$$

Considering that $t_{A+O}<t_N+t_A+t_O=t_{N+A+O}$ according to table 1, we know that $\overline{\tau_{A+O}}\tau_{N+A+O}=0$ and $\tau_{A+O}\overline{\tau_{N+A+O}}=\delta_{A+O,N+A+O}$, rendering the above equation as follows:

$$y(t)=x\overline{\tau_{B+O}}+\bar{x}(\tau_{B+O}+\delta_{A+O,N+A+O}) \tag{5.6.}$$

Eq. 5.6. provides us with the dynamic "1" hazard pattern. One can easily observe that:

x="1"→ $y(t)=\overline{\tau_{B+O}}$, meaning that for transition x"1"↓"0", the circuit's output will switch to "1"↓"0" after t_{B+O}, and no further transitions will occur afterwards (normal output transition – no hazard present)

x="0"→ $y(t)=\tau_{B+O}+\delta_{A+O,N+A+O}$, meaning that for transition x"0"↑"1" and considering that $t_B>t_{N+A}$ (see the figure above), the circuit's output will present a

- "0" until t_{A+O},
- "1" from t_{A+O} to t_{N+A+O} (term $\delta_{A+O,N+A+O}$),
- "0" between t_{N+A+O} → t_{B+O}, and
- "1" after t_{B+O}(term τ_{B+O}).

Please observe that if $t_B \leq tN+A$, the output will switch to "1" after t_{B+O} (term τ_{B+O})$\leq t_{N+A+O}$ (term $\delta_{A+O,N+A+O}$), rendering the hazard invisible (masked but not non-existent).

5.4. Dynamic - Case 4 Algebraic description → $y = x \cdot (x + \bar{x})$

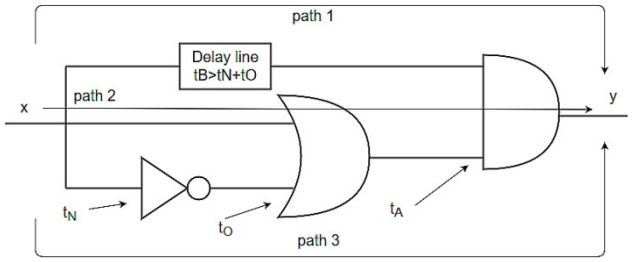

Fig.4.Hazard generating circuit – case 4

Delays: →path 1– t_B+t_A / associated $\tau_{B+A}=\tau(t_{B+A}-t)$; non inverting
→path 2- $t_O+t_A=t_{O+A}$ / associated $\tau_{O+A}=\tau(t_{O+A}-t)$; non inverting
→path3-$t_N+t_O+t_A=t_{N+O+A}$/associated $\tau_{N+O+A}=\tau(t_{N+O+A}-t)$; inverting

$$y(t) = (x \oplus \tau_{B+O}) \cdot [(x \oplus \tau_{O+A}) + (\bar{x} \oplus \tau_{N+O+A}) = (x\overline{\tau_{B+O}} + \bar{x}\tau_{B+O}) \cdot (x\overline{\tau_{O+A}} + \bar{x}\tau_{O+A} + x\tau_{N+O+A} + \bar{x}\overline{\tau_{N+O+A}}) =$$

$$= x\overline{\tau_{B+O}}(\overline{\tau_{O+A}} + \tau_{N+O+A}) + \bar{x}\tau_{B+O}(\tau_{O+A} + \overline{\tau_{N+O+A}}) \quad (5.7.)$$

Considering that $t_O+t_A=t_{O+A}<t_N+t_O+t_A=t_{N+O+A}$ according to table 1, we know that $\overline{\tau_{O+A}} + \tau_{N+O+A} = \overline{\delta_{O+A,N+O+A}}$ and $\tau_{O+A} + \overline{\tau_{N+O+A}} = 1$, thus rendering the above equation as follows:

$$y(t) = x \cdot \overline{\tau_{B+O}} \cdot \overline{\delta_{O+A,N+O+A}} + \bar{x} \cdot \tau_{B+O} \quad (5.8.)$$

Eq. 5.8. provides us with the dynamic "1" hazard pattern. One can easily observe that:

x="0"→$y(t)=\tau_{B+O}$, meaning that for transition x "0"↑"1", the circuit's output will switch to "0"↑"1" after t_{B+O}, and no further transitions will occur afterwards (normal output transition – no hazard present)

x="1"→ $y(t) = x \cdot \overline{\tau_{B+O}} \cdot \overline{\delta_{O+A,N+O+A}}$, meaning that for transition x "1"↓"0" and considering that $t_B>t_{N+A}$ (see fig. 5.4.), the circuit's output will present a

- "1" until t_{O+A},
- "0" from t_{O+A} to t_{N+O+A} (term $\overline{\delta_{O+A,N+O+A}}$),
- "1" between t_{N+O+A} → t_{B+O},
- "0" after t_{B+O} (term τ_{B+O}).

Please observe that if $t_B \leq t_{N+A}$, the output will switch to "0" after t_{B+O}(term $\overline{\tau_{B+O}}$)$\leq t_{N+A+O}$ (term $\overline{\delta_{O+A,N+O+A}}$), rendering the hazard invisible (masked but not non-existent)

As a conclusion, we can state the following:

- Let $t_1,t_2,\ldots,t_n$, be n timestamps respecting $t_1< t_2<\ldots< t_n$,

- Let there be a logic circuit implementing a logic function $f(x,a_0,a_1,\ldots,a_{m-2})$, where $\{x,a_0,a_1,\ldots, a_{m-2}\}$ is the logic function's m-component input vector and x is the input variable to be analysed.
- Let us assume that input variable x is characterized by n distinct in→out paths, therefore generating an n-component secondary input vector $\{x_1, x_2,\ldots, x_n\}$ with associated specific propagation delays of $\{t_1, t_2, t_3,\ldots, t_n\}$
- Rewrite the secondary logic function's equation:

$$F= f(x_1, x_2,\ldots, x_n, a_0, a_1,\ldots, a_{m-2}) \quad (5.9.)$$

with respect to TDLV (τ,δ) using $x_k=x\oplus\tau_k$, for $k\in\{1,2,\ldots, n\}$

$$F= f(x\oplus\tau_1, x\oplus\tau_2,\ldots, x\oplus\tau_n, a_0, a_1,\ldots, a_{m-2}) \quad (5.10.)$$

and reduce its expression according to (τ,δ) properties:

$$F= f(x, \tau_1, \tau_2,\ldots, \tau_n, \delta_{1,2},\ldots, \delta_{i-1,i},\ldots, \delta_{n-1,n}, a_0, a_1,\ldots, a_{m-2})$$

where x is the initial value for input variable x.

By determining a specific combination for the remainder of the input vector $\{a_0, a_1,\ldots, a_{m-2}\}$ that would render the logic function's expression identical to one of the patterns presented above, we can state that hazard exists and we can specify the moment of the time when it occurs.

Assuming that $t_{init}<t_\alpha<t_\beta<t_\omega<t_{final}$, the logic function's expression is reduced to:

- $F=\bar{x}\cdot\delta_{\alpha,\beta}$ - static "0" beginning at t_α and ending at t_β
- $F=\bar{x}+x\cdot\overline{\delta_{\alpha,\beta}}$ - static "1" beginning at t_α and ending at t_β
- $F=x\cdot\overline{\tau_\varpi}+\bar{x}\cdot(\tau_\varpi+\delta_{\alpha,\beta})$ - dynamic "0" begins at t_α and ends at t_ω
- $F=x\cdot\overline{\tau_\varpi}\cdot\overline{\delta_{\alpha,\beta}}+\bar{x}\cdot\tau_\varpi$ - dynamic "1" begins at t_α and ends at t_ω

6. ANALYSIS EXAMPLE

6.1. Analysis example 1:

$$f(a,b,c)=a\cdot(b+c)+\bar{a}\cdot\bar{b}\cdot c$$

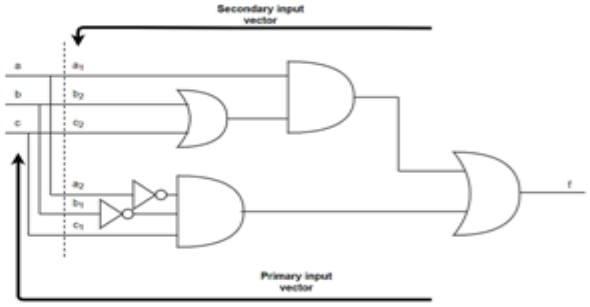

Fig.5. Analysis example – circuit 1

Each component of the primary input vector is characterized by two distinct in→out paths. However, analysis will be performed only for variables a and b. Variable c, although characterized by two paths, does not exhibit a complementary relationship between them, so it will not generate a hazard (E.B.EICHELBERGER [6]). We have computed for each path its specific propagation delay time and defined the secondary input vector by associating with each path a unique secondary variable derived from a primary one (as presented in fig.5.). For presentation purposes we've considered the circuit to be implemented using classic logic gates. The method works the same for any technology chosen to implement the circuit analysed.

Table. 7 CIRCUIT 1-PATH DELAYS FOR VARIABLE a and b

Path delay	FAST Schottky	LS TTL
t_{a1}	13.2 ns	37 ns
t_{a2}	18.2 ns	52 ns
t_{b1}	18.2 ns	52 ns
t_{b2}	19.8 ns	59 ns

Please note that $t_{a1} < t_{a2}$ and $t_{b1} < t_{b2}$.

6.1.1. Analysis for input (a) transition

Function's expression considering the secondary inputs becomes:

$$f(a_1, a_2, b, c) = a_1 \cdot (b + c) + \overline{a_2} \cdot \overline{b} \cdot c \tag{6.1.}$$

Function's expression with respect to TDLV (τ,δ) is:

$$f(a, \tau_{a1}, \tau_{a2}, b, c) = (a \oplus \tau_{a1}) \cdot (b + c) + \overline{(a \oplus \tau_{a2})} \cdot \overline{b} \cdot c =$$
$$= a \cdot [(b + c) \cdot \overline{\tau_{a1}} + \overline{b} \cdot c \cdot \tau_{a2}] + \overline{a} \cdot [(b + c) \cdot \tau_{a1} + \overline{b} \cdot c \cdot \overline{\tau_{a2}}] \tag{6.2.}$$

Now, we can perform the analysis on variable a by assuming values for the *(b+c)* and $\overline{b} \cdot c$ terms:

- $(b + c) = \overline{b} \cdot c = 0$ → f(a,τ_{a1},τ_{a2},b,c)=0; NO output anomalies present (output is a constant "0")
- $(b + c) = 0, \overline{b} \cdot c = 1$ → NO solution for the logic equations system → this situation will never occur
- $(b + c) = 1, \overline{b} \cdot c = 0 \rightarrow f(a, \tau_{a1}, \tau_{a2}, b, c) = a \cdot \overline{\tau_{a1}} + \overline{a} \cdot \tau_{a2} \rightarrow$ both transitions for a will provide a singular transition on the output (τ_{a1} respectively $\overline{\tau_{a1}}$) → NO output anomalies present

$(b + c) = \overline{b} \cdot c = 1 \rightarrow f(a, \tau_{a1}, \tau_{a2}, b, c) = a \cdot \left(\overline{\tau_{a1}} + \tau_{a2}\right) + \overline{a} \cdot \left(\tau_{a1} + \overline{\tau_{a2}}\right) \rightarrow$ Considering $t_{a1}<t_{a2}$ and the (τ,δ) properties, we know that $\overline{\tau_{a1}} + \tau_{a2} = \overline{\delta_{1,2}}$ and $\tau_{a1} + \overline{\tau_{a2}} = 1$, thus rendering the function's equation to be:

$$f = a \cdot \overline{\delta_{1,2}} + \overline{a} \tag{6.3.}$$

Eq. 6.3 presents a pattern identifying a static "1" hazard for input transition **a** "1"↓"0" and b=0, c=1. By placing the time origin at the moment variable a switches, the output will evolve as presented in fig.6.:

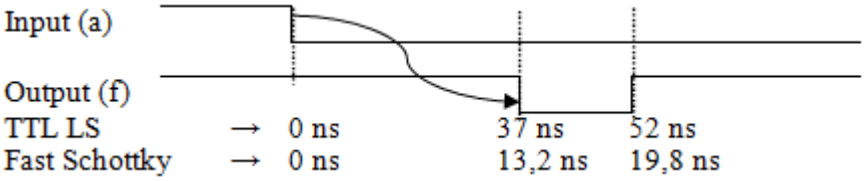

Fig.6. Output waveform for input a transition "1"↓"0"

The analysis for input b follows a similar path

6.2. Analysis example 2:

$$f(a,b,c) = a \cdot (b+c) \cdot \overline{(a+c) \cdot b}$$

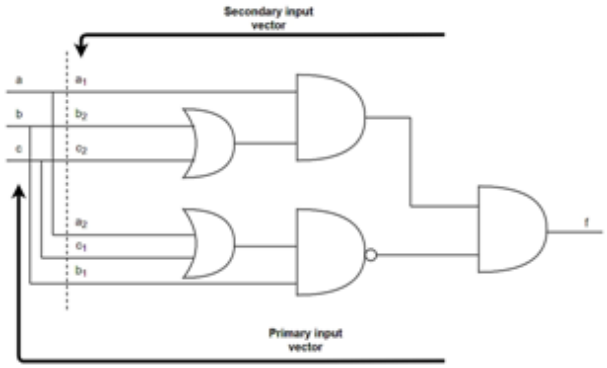

Fig.7. Analysis example – circuit 2

Each component of the primary input vector is characterized by two distinct in → out paths.We have computed a specific propagation delay time for each path and defined the secondary input vector by associating with each path a unique secondary variable derived from a primary one (as presented in fig.7.).

Table. VIII CIRCUIT 2-PATH DELAYS FOR VARIABLES **a,b,c**

Path delay	FAST Schottky	LS TTL
t_{a1}	13.2 ns	30 ns
t_{a2}	18.2 ns	52 ns
t_{b1}	18.2 ns	30 ns
t_{b2}	19.8 ns	52 ns
t_{c1}	19.2 ns	52 ns
t_{c2}	19.8 ns	52 ns

Please note that $t_{a1}<t_{a2}$ and $t_{b1}<t_{b2}$. However, we encounter a most interesting situation when we consider variable c. In the case of LS TTL implementation, both paths are characterized by the same propagation delay time, meaning that although this input variable respects all conditions that would make it a candidate for hazard analysis, we do not need to analyse the circuit's behaviour when variable c switches because any complementary output switch should occur at the same moment of time, therefore cancelling each other. In the case of FAST Schottky implementation, tc1 < tc2, so analysis should be performed.

6.2.1. Analysis for input (c) transition

Function's expression considering the secondary inputs becomes:

$$f(a,b,c_1,c_2) = a \cdot (b + c_2) \cdot \overline{(a + c_1) \cdot b} \tag{6.4.}$$

Function's expression with respect to TDLV (τ,δ) is:

$$f(a,b,c,\tau_{c1},\tau_{c2}) = a \cdot (b + c \oplus \tau_{c2}) \cdot \overline{(a + c \oplus \tau_{c1}) \cdot b} == a \cdot \bar{b} \cdot (c \oplus \tau_{c2}) =$$
$$= c \cdot a \cdot \bar{b} \cdot \overline{\tau_{c2}} + \bar{c} \cdot a \cdot \bar{b} \cdot \tau_{c2} \tag{6.5.}$$

In this case, only τ_{c2} is present in the function's expression, meaning that only one delay will be visible at the circuit output (t_{c2}) if the proper conditions are met (a=1, b=0)

→ No anomalous behaviour possible →

variable c switch will not generate an output anomalous behaviour under any circumstances.

6.2.2. Analysis for input (a) transition

Function's expression with respect to the secondary input vector is:

$$f(a_1,a_2,b,c) = a_1 \cdot (b + c) \cdot \overline{(a_2 + c) \cdot b} = a_1 \cdot \overline{a_2} \cdot b \cdot \bar{c} + a_1 \cdot \bar{b} \cdot c \tag{6.6.}$$

Function's expression with respect to TDLV (τ,δ) is:

$$f(a,\tau_{a1},\tau_{a2},b,c) = (a \oplus \tau_{a1}) \cdot \overline{(a \oplus \tau_{a2})} \cdot b \cdot \bar{c} + (a \oplus \tau_{a1}) \cdot \bar{b} \cdot c =$$
$$= a \cdot \left(b \cdot \bar{c} \cdot \overline{\tau_{a1}} \cdot \tau_{a2} + \bar{b} \cdot c \cdot \overline{\tau_{a1}}\right) + \bar{a} \cdot \left(b \cdot \bar{c} \cdot \tau_{a1} \cdot \overline{\tau_{a2}} + \bar{b} \cdot c \cdot \tau_{a1}\right) \tag{6.7.}$$

Now, we can perform the analysis on variable a by assuming values for $b \cdot \bar{c}$ and $\bar{b} \cdot c$ terms:

- $b \cdot \bar{c} = \bar{b} \cdot c = 1$ → NO solution for the logic equations system
- $b \cdot \bar{c} = \bar{b} \cdot c = 0$ → f(a,τ_{a1},τ_{a2},b,c)=0; NO output anomalies
- $b \cdot \bar{c} = 0, \bar{b} \cdot c = 1$ → both transitions for a will provide a singular transition on the output (τ_{a1} respectively $\overline{\tau_{a1}}$) → NO output anomalies present
- $b \cdot \bar{c} = 1, \bar{b} \cdot c = 0$ → Considering $t_{a1}<t_{a2}$ and the (τ,δ) properties, we know that $\overline{\tau_{a1}} \cdot \tau_{a2} = 0$ and $\tau_{a1} \cdot \overline{\tau_{a2}} = \delta_{1,2}$, thus rendering the function's equation to be:

$$f = \bar{a} \cdot \delta_{1,2} \tag{6.8.}$$

Eq. 6.8. presents a pattern identifying a static "0" hazard for input transition **a** "0"↑"1" and b=1, c=0. By placing the time origin at the moment variable a switches, the output will evolve as presented in fig.8.:

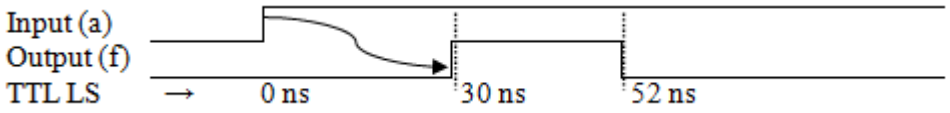

Fig.8. Output waveform for input a transition "0"↑"1"

7. CONCLUSION

Both approaches presented will either improve or ease the use of the classic methods while maintaining their reliability. Please bear in mind that it is not our intention to state that the classic techniques are failing, but we simply wish to demonstrate that the same results may be reached using less computation, and if we use the second approach, the outcome will provide more information as far as timing is concerned.

The first approach presented may be used when the behavior of a combinational logic circuit needs to be analyzed. One can easily determine if the CLC's output may present a hazard simply by identifying specific terms in its expression. At this point, we may choose not to use that circuit or perform structural changes (if possible) to mask the hazardous behavior. However, choosing the second approach (masking anomalous behavior) comes with a cost, as masking is performed by entering redundant terms into the logic function equation or by using dummy gates to equalize the different path propagation delays. That means an increase in the implementation cost and a decrease in speed.

The second approach, if used to analyze an already implemented CLC, will provide detailed information on the output's behavior with respect to the time axis. Having this information available, we shall be able to design a hierarchically superior circuit that uses the present circuit's outputs in such a manner that the time frames, when the circuit output is malfunctioning, are not to be considered.

The second approach also presents us with a possible development path in the area of asynchronous automata. It is well known that the proper design and operation of these devices is dependent on strict timing specifications. The proposed approach is to design the automaton as a synchronous machine, use the methodology presented to map the outputs of the input group of functions CLC (next state CLC – Mealy or Moore) and define and design a variable time pulse generator to be used as a synchronizing signal (instead of a fixed parameter clock signal) that will use the earliest mapped moment of time to change the state of the automation. Thus, the automaton will not be an asynchronous one but rather a pseudo synchronous one. The advantages would be the ease of design and less susceptibility to timing issues while retaining most of the advantages of the asynchronous structure, such as high speed and fast response.

REFERENCES

[1] Beister J.: "A unified approach to combinational hazards", IEEE Trans.Comp.VolC-23, pp. 566-575, 1974, 10.1109/T-C.1974.223996

[2] McCluskey E. J.:"Logic Design Principles",Prentice-Hall, Englewood Cliffs, NJ, 1986.

[3] Tinder Richard F.: "Engineering Digital Design", Second Edition, Elsevier - ACADEMIC PRESS, 2001

[4] Bryant R.:"Graph-based Algorithms for Boolean Function Manipulation," IEEE Trans.Comp., 677-691 (1986), 10.1109/TC.1986.1676819

[5] Akers S.: "Binary Decision Diagrams," IEEE Trans.Comp., C-27, 509-516 (1978). DOI:10.1109/TC.1978.1675141

[6] Nowick S.M.,O'Donnell C.W.:"On the existence of hazard-free multi-level logic" Proceedings / IEEE ASYNC 03, May12-16,2003, DOI: 10.1109/ASYNC.2003.1199171

[7] Jeong C., Nowick S.M.: "Fast hazard detection in combinational circuits" ACM DAC 04, June 7-11, 2004, 10.1109/DAC.2004.240453

[8] Berthomieu B., Diaz M.:"Modeling and Verification of Time Dependent Systems using Time Petri Nets", IEEE Transactions on Software Engineering 17,259-273, 1991DOI: 10.1109/32.75415

[9] J.A. Brzozowski and C.J.H. Seger: "Advances in Asynchronous circuit theory Part II - Bounded Inertial Delay Model, MOS Circuits, Design Techniques", EATCS Bulletin 43, 199-263, 1991

[10] J. Brzozowski, B. Li, Y. Ye: "On the Complexity of the Evaluation of Transient Extensions of Boolean Functions", 12th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2010,

[11] Y. Ye., J. Brzozowski: "Covering of Transient Simulation of Feedback-Free Circuits by Binary Analysis",Int. J.Found. Comput. Sci.17,949-973, 2006

[12] M. Gheorghiu, J. Brzozowski:"Simulation of Feedback-Free Circuits in the Algebra of Transients"Internat. J.Found. Comput.Sc.,14,1033-1054, 2003.

[13] Oded Maler, Amir Pnueli: "Timing Analysis of Asynchronous Circuits using Timed Automata", Correct Hardware Design and Verification Methods, Volume 987, 1995,pp189-205,Springer2005,DOI: 10.1007/3-540-60385-9_12

[14] Alan R. Martello, Steven P. Levitan: "Temporal analysis of time bounded digital systems", Correct Hardware Design and Verification Methods, Volume 683, Springer 1993, pp 27-38, DOI: 10.1007/BFb0021712

[15] Asarin A., Maler O. and Pnueli A.: "Symbolic Synthesis of Discrete and Timed Systems" in A. Nerode (Ed), Hybrid Systems II, Springer LNCS, 1995, DOI: 10.1.1.43.5633 1995

[16] William K.C. Lam, Robert K. Brayton: "Timed Boolean Functions – A unified formalism for exact timing analysis", Kluwer Academic Publishers, 1994, ISBN 0-7923-9454-2

[17] Galupa N.: "Increase of Sequential Systems Performance Using Digital Hazard Analysis", ICCS/ISITA '92, 1096-1100, ISBN: 0-7803-0803-4, 10.1109/ICCS.1992.255092

[18] Galupa N.: "TIME/LOGIC VARIABLES USED FOR DIGITAL HAZARD SEARCH,"Proceedings of IEEE CCECE 2008, DOI: 10.1109/CCECE.2008.4564553

[19] Stevens K.S., Ginosar R., and Rotem S.: Relative timing [asynchronous design], in IEEE Transactions on VLSI Systems, pp. 129-140, ISSN 1063-8210, 2002, 1.1109/TVLSI.2002.801606

[20] R. Ben Salah, Bozga M. and Maler O.: "On Timing Analysis of Combinational Circuits," In FORMATS'03, LNCS 2791, pages 204-219. Springer (2003)

[21] R. Ben Salah, Bozga M. and Maler O.: "On Timed Components and their Abstraction," In SAVCBS'07 Workshop, ACM ISBN 978-1-59593-721-6/07/0009 (2007)

[22] Riedel M.D., Bruck J.: "Timing Analysis of Cyclic Combinational Circuits," Technical Report, Parallel and Distributed Systems Group, CaltechPARADISE:2004.ETR060, 25 Feb 2014

AUTHORS

Nicolae Galupa (B.Sc and M.Sc. Hons 1988, PhD 2003) received his BSc and MSc degree from Technical University Iasi, Faculty of Automatic Control and Computer Engineering, Iasi, Romania in 1988 after a 5 years training programme and PhD degree from the same University in 2003 after a 6 years training programme.

He has been with the Faculty of Automatic Control and Computer Engineering, Technical University Iasi, Iasi, Romania since 1990. Also since 2006 he joined the digital circuits design team of DICODE Ltd. and focused on timing issues in digital circuits.

He joined IEEE in 2006 and has been a member of IEEE Region 8 ever since.