# ALGEBRAIC ANALYSIS OF NONDETERMINISTIC BEHAVIOR

S. Leinwand and T. Lamdan
Dept. of Applied Mathematics
The Weizmann Institute of Science, Rehovot, ISRAEL

## ABSTRACT

This paper is concerned with the analysis of design errors that lead to unpredictable response of digital systems. Besides classical topics, such as hazards and races, the analysis of malfunctions in real circuits is also included. After defining the notion of behavior and nondeterministic response, a general approach for detecting such design problems through algebraic analysis is presented. Compared with existing simulation methods, the algebraic technique provides results of improved accuracy. Another basic advantage is the ability to accomodate modular synthesis of digital systems.

Examples show how the proposed methods deal with sequential circuits under various delay assumptions. In particular, analysis of designs based on nominal delay parameters and on window delays is presented. A novel method, aiming at spike detection, is also presented. The ability of the algebraic analysis to detect errors in a modular design environment is illustrated by means of an example.

Finally, the topic of nondeterministic behavior at RTL is briefly discussed. Notably, an algebraic method for deriving setup and hold time constraints from the circuit delay parameters is proposed.

## 1. NONDETERMINISTIC BEHAVIOR OF DIGITAL CIRCUITS

In a previous paper [2], the importance of functional abstraction for the verification of digital systems has been shown. Functional abstraction is defined as the generation of the behavior for a given digital system. Typically, a system is described by a network of interconnected modules and its behavior, as seen at the interface, is generated. Often, digital systems yield unpredictable response. The term nondeterministic behavior refers to the possibility of a digital system to experience changes that are dependent not only on its inputs and its starting state, but also on some implementation details not totally controlable. The qualification of functional is used for a system operated such that its response is always uniquely defined.

Functional abstraction [2] assumes that the digital system under consideration is used only with deterministic results. Thus, it is very important to detect any attempt of using the system in the nonfunctional domain.



Although theoretically any digital system can be analyzed and its behavior fully defined, many factors not completely controlable occur in practice. For example, variations in propagation delay parameters, due to fabrication tolerance and environmental factors, may change the circuit's behavior. Since it is not feasible to fully control these parameters, the presence of nondeterministic behavior must be assumed.

### 1.1 Factors for Consideration

Three possible approaches for dealing with the effect of propagation delays on the design are:

a) Unbounded delay designs are able to accomodate any propagation delays and thus are insensitive to implementation factors. The philosophy of speed independent design is an offspring of this approach to digital systems.

b) Nominal delay designs consider that full control of the propagation delays is possible.

c) Window delay designs take into consideration the upper and lower limits of the delay parameters. Usually, these limits are determined by means of a probabilistic analysis of the dispersion in delay parameters. A more pessimistic solution assumes that worst case delays are given and the design must be able to work under this constraints.

An important factor affecting the analysis is the operation mode intended for the system. Relevant cases are:

a) Fundamental mode operation with single input changes [5] such that the input changes are separated by delays sufficient to allow the circuit to stabilize.

b) Fundamental mode operation with multiple input changes, but still allowing the circuit to stabilize before application of a new input combination.

c) Nonfundamental mode operation, such that the circuit does not stabilize before a new input combination is applied.

Usually, circuits are expected to operate in fundamental mode with single input changes and the occurrence of multiple input changes or short input pulses is due to design errors.

### 1.2 Advantages of Algebraic Analysis

The methods currently used for the analysis of nondeterministic behavior of digital circuits can be grouped into two categories:

a) To the first one belong analytical methods able to detect classical design problems, such as hazards and races [1]. All these methods analyze functional behavior under the assumption of unbounded delays. A design passing such a test is insensitive to the

implementation and can operate deterministically with any delay parameters. Actually, designs are seldom meant to be speed independent, therefore the practical value of these methods is quite low.

b) The prevailing tool for analysis of digital circuits in the presence of implementation parameters is logic simulation. It uses a mostly structural description of the analyzed circuit, with built in routines describing the behavior of basic components [4]. This provides an effective method for checking the circuit's behavior for a given input combination. However, exhaustive testing of all possible input combinations can not be used in practical circuits. In [2], a similar argument against verification by means of simulation was presented. When dealing with nondeterministic behavior, the situation is much worse. In addition to all possible input combinations, the relative delay between application of any two such inputs plays an important role. Thus, analysis in the present context means derivation of admissible inputs including allowed delays between input changes. While simulation can be usefull for checking correct behavior for some typical situations, it can not be considered a general analysis tool.

### 1.3 Algebraic Analysis of Nondeterministic Behavior

The algebraic analysis of nondeterministic behavior relies on the following assumptions:

i) Nondeterminism is exhibited by the appearance of erroneous internal states in the circuit. Internal combinational parts of the circuit are never considered to exhibit nondeterminism by themselves. They are included in the analysis only through their effect on internal states. Combinational parts feeding the outputs are excepted from this rule. Since the analyzed module can feed another sequential circuit, enough information must be stored to enable detection of any nondeterministic features when the module is used as a component of complex systems.

ii) All cases of nondeterministic behavior are instances of the system's behavior being sensitive to the relative order of the occurrences of changes. Even when the exact delay parameters are known, the relative order of two internal changes can be ascertained only if both are generated by the same input transition. Therefore, the presence of nondeterministic features can be tested only with respect to a single input change, that brings forth multiple internal transitions propagating on reconvergent fanout branches.

iii) Input changes that can not be traced to a common input transition can not be ascertained as generating nondeterministic behavior, since their relative order is not yet known. The solution is to set up admissible relative delays between unrelated input transitions. When the module is used as part of a complex system, the limitations of admissible inputs are checked and if they are infringed, the system behaves nondeterministically.

iv) The algebraic analysis proceeds backwards, starting with feedback signals and deriving the local input transitions that give rise to problematic behavior. These local combinations are then expressed in terms of primary inputs and define the unallowed input combinations.

The analysis of nondeterministic behavior is integrated in the process of design verification based on functional abstraction.

## 2. FUNDAMENTALS OF ALGEBRAIC ANALYSIS

### 2.1 Dynamic Boolean Algebras

The algebraic framework is provided by a dynamic boolean algebra (DBA) with four predicates for each signal, representing its enabled state, its disabled state, its rising state (from disabled to enabled) and its dropping state (from enabled to disabled). Corresponding to a signal X, these four predicates are denoted by X, ~X, *X and *~X , respectively. Considerations on the implications of using dynamic boolean algebras can be found in [3]. The circuit components AND, OR and INVERTER are denoted by . , + and ~. The basic postulates are:

i) Orthogonality: $X + \sim X + {}^*X + {}^*\sim X = 1$ ,
$X \cdot \sim X = X \cdot {}^*X = X \cdot {}^*\sim X = \sim X \cdot {}^*X = \sim X \cdot {}^*\sim X = {}^*X \cdot {}^*\sim X = 0$

ii) $\sim\sim X = X$ , $^*(\sim X) = {}^*\sim X$ , $^*\sim(\sim X) = {}^*X$

iii) $\sim(X + Y) = \sim X \cdot \sim Y$ , $\sim(X \cdot Y) = \sim X + \sim Y$

iv) $^*(X + Y) = \sim X \cdot {}^*Y + {}^*X \cdot \sim Y + {}^*X \cdot {}^*Y + {}^*X \cdot {}^*\sim Y + {}^*\sim X \cdot {}^*Y$
$^*\sim(X + Y) = \sim X \cdot {}^*\sim Y + {}^*\sim X \cdot \sim Y + {}^*\sim X \cdot {}^*\sim Y + {}^*\sim X \cdot {}^*Y + {}^*X \cdot {}^*\sim Y$
$^*(X \cdot Y) = X \cdot {}^*Y + {}^*X \cdot Y + {}^*X \cdot {}^*Y + {}^*X \cdot {}^*\sim Y + {}^*\sim X \cdot {}^*Y$
$^*\sim(X \cdot Y) = X \cdot {}^*\sim Y + {}^*\sim X \cdot Y + {}^*\sim X \cdot {}^*\sim Y + {}^*\sim X \cdot {}^*Y + {}^*X \cdot {}^*\sim Y$

As shown in [3], this algebra has several cases of inconsistency. Thus, the final result may depend on the order of reductions performed. The practical solution is to define a strategy of reductions, tailored to the intended application.

### 2.2 Treatment of Sequential Components

The key to the analysis of nondeterministic features is the treatment of sequential components. Topologically, they are easily identified by the presence of feedback lines. In the following, a sufficient set of feedback signals (such that any cycle contains at least one feedback signal) describe the internal states.

Based on the above algebraic system, the following procedure is used to analyze digital circuits. Consider that a set of feedback signals has been selected. Be X one of them. Then:

1. The next state expression for X can be described as a function X' = f(X).
Based on this expression, the dynamic expressions for *X' and for *~X' are computed. They include both the functional and the nondeterministic behavior of X.

2. In the expression of *X', all minterms containing X or *X can be eliminated, since a rise on X assumes that its initial value is not "enabled", and since the terms in *X show only the latching effect. Similarly, the minterms containing ~X or *~X occuring in the expansion of *~X' can be neglected.

3. The dynamic behavior of feedback signal X can be factorized as follows:
$^*X' = L + M \cdot {}^*\sim X$
$^*\sim X' = P + R \cdot {}^*X$
where L, M, P and R are boolean expressions not containing *X or *~X. An equivalent formulation is obtained from the orthogonality postulate:
$^*X' = L + M \cdot \sim R \cdot {}^*\sim X + M \cdot {}^*R \cdot {}^*\sim X + M \cdot {}^*\sim R \cdot {}^*\sim X + M \cdot R \cdot {}^*\sim X$

*~X' = P + ~M . R . *X + *M . R . *X + *~M . R . *X + M . R . *X

The boolean equation M . R = 1 defines an oscillation condition. Thus one can write:

*X' = L + M . ~R . *~X + M . *R . *~X + M . *~R . *~X + M . R . oscillation

*~X' = P + ~M . R . *X + *M . R . *X + *~M . R . *X + M . R . oscillation

4. By replacing all occurences of *X by the expansion of *X', and similarly replacing *~X, the result is:

*X' = L + M . ~R . P + M . *R . P + M . *~R . P + M . R . oscillation

*~X' = P + ~M . R . L + *M . R . L + *~M . R . L + M . R . oscillation

5. Usually, the circuit comprises several feedback signals. Having built the dynamic behavior of each such signal, the circuit is described by a system of dynamic boolean equations. The next reduction step consists of normalizing the dynamic system. This has a direct correspondent in flow table normalization [1], by which all chains of internal state transitions are replaced by single transitions, leading directly from one stable state to another. In the system of dynamic equations under consideration, normalization can be performed by replacing any further references to *X and *~X by the corresponding expansions of *X' and *~X'. Upon completion of the process, the only dynamic terms appearing in the dynamic expansions of feedback signals refer to the primary inputs.

6. A further simplification consists of removing all minterms corresponding to unstable total states. For example, the dynamic minterm X . *I can be removed if the value of internal state X is zero whenever input I is disabled. Obviously, the rise of I starts from a disabled value, internal state X is zero and therefore the minterm can be neglected.

7. Finally, all occurences of X in the expansion of *~X' can be replaced by one, since the drop of X starts from X enabled. Similarly, occurences of ~X in *X' are replaced by one.

The interpretation of the results depends on the particular assumptions on delay parameters and on the allowed operation mode.

## 3. UNBOUNDED DELAY ANALYSIS

### 3.1 DBA for Unbounded Delay Analysis

The algebraic framework for unbounded delay analysis is provided by the dynamic boolean algebra presented in section 2.1. As discussed above, this algebra has several cases of inconsistency: when the postulates defining it are used during algebraic manipulations, the result may depend on the order of reductions performed. The source of these inconsistencies is the requirement of orthogonality - any signal can not be in more than one state during a considered operation period. In circuits displaying hazardous behavior, this postulate can not be enforced and problems arise. As a practical application, any inconsistency can be readily associated with the effects of hazards in digital systems. Thus, when used for analyzing nondeterministic behavior all conditions producing inconsistencies have to be pointed out.

In order to generate all possible inconsistencies, the dynamic operators are distributed over static operators before any other reductions. The only properties of switching algebra that are preserved in this dynamic framework are commutativity, associativity, idempotency and the deMorgan laws. Thus, reductions based on these four postulates can be performed at any time, without changing the dynamic result.

All mixed direction minterms in the AND and OR expansions correspond to logic or function hazards. The hazard minterms are underscored in the following examples.

### 3.2 Unbounded Delay Analysis of Circuits

Although the analysis of designs under the assumption of unbounded delays has been treated by many researchers and has several solutions by now classical [1], the algebraic method is included for comparison.

#### 3.2.1 Hazards in Combinational Circuits

The dynamic expansion of the combinational function (as defined by the circuit) is developed. When only function hazards are sought, the dynamic expansion of a canonical function representation is built, by considering the join of all its minimal implicants. Thus, the details of circuit implementation are not taken into consideration. (It is well known that this canonical representation of functions contains no logic hazards).

As an example, consider the circuit shown in figure

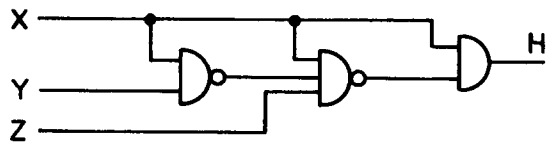


Figure 1: Circuit with Hazards

1. The dynamic expression for the rise of H (developed from the circuit) is separated into groups of terms, according to their nature:

*H = X . ~Y . *~Z + X . Z . *Y + ~Z . *X (a)
+ X . *Y . *~Z (b)
+ X . (*Y . *Z + *~Y . *~Z) + *X . (*Y . *Z + (c)
*~Y . *~Z + ~Y . *Z + Z . *~Y + *~Y . *Z) (c)
+ *~X . (~Y . *~Z + Z *Y + *Y . *~Z (c)
+ *Y . *Z + *~Y . *~Z) (c)
+ *X . (Y . Z + Y . *Z + Y . *~Z + ~Y . *~Z + (d)
Z . *Y + *Y . *~Z + *~X . (Y . Z + Y . *~Z + (d)
Y . *Z + ~Y . *Z + Z . *~Y + *~Y . *Z) (d)
+ ~Y . Z . (*X + *~X) (e)

The function hazards are obtained by expanding the canonical representation of H, denoted by |H|:

|H| = X . Y + X . ~Z
*|H| = X . ~Y . *~Z + X . Z . *Y + ~Z . *X (a)
+ X . *Y . *~Z (b)
+ Y . *X . (Z + *Z + *~Z) + ~Y . *X . *~Z (f)
+ Z . *X . *Y + *X . *Y . *~Z (f)
+ X . (*Y . *Z + *~Y . *~Z) + *X . (*Y . *Z + (c)
*~Y . *~Z + ~Y . *Z + Z . *~Y + *~Y . *Z) (c)
+ *~X . (~Y . *~Z + Z . *Y + *Y . *~Z (c)
+ *Y . *Z + *~Y . *~Z) (c)

Interpretation:

1. Minterms in group (a) correspond to hazard free effects of single input changes.

2. Minterms in group (b) correspond to hazard free effects of multiple input changes.

3. Minterms (c) correspond to function hazards due to multiple input changes.

4. Minterms (d) correspond to dynamic logic hazards due to either single or multiple input changes.

5. Minterms (e) correspond to static logic hazards.

6. Minterms (f) correspond to "clean" transitions of the function, that are hazardous in the circuit implementation.

### 3.2.2 Races in Sequential Circuits

The hazard minterms in the dynamic expansion of feedback signals reflect the effect of races. For example,

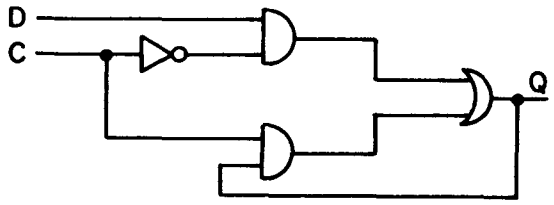


Figure 2: Level Latch

consider the level latch shown in figure 2. The dynamic expansions of Q (after step 2 of algorithm 2.2) are:

$$*Q' = \tilde{}Q \cdot D \cdot *\tilde{}C + \tilde{}C \cdot *D + \tilde{}Q \cdot *\tilde{}C \cdot *D + \tilde{}Q \cdot (*C \cdot *D + *\tilde{}C \cdot *\tilde{}D) + *\tilde{}Q \cdot (D \cdot *\tilde{}C + \tilde{}D \cdot *C + D \cdot *C + *D \cdot *C + *D \cdot *\tilde{}C + *\tilde{}D \cdot *\tilde{}C + *\tilde{}D \cdot *C)$$

$$*\tilde{}Q' = Q \cdot \tilde{}D \cdot *\tilde{}C + \tilde{}C \cdot *\tilde{}D + Q \cdot (D \cdot *\tilde{}C + D \cdot *C + *D \cdot *C + *D \cdot *\tilde{}C + *\tilde{}D \cdot *\tilde{}C + *\tilde{}D \cdot *C) + *Q \cdot (D \cdot *\tilde{}C + \tilde{}D \cdot *\tilde{}C + D \cdot *C + *D \cdot *C + *D \cdot *\tilde{}C + *\tilde{}D \cdot *\tilde{}C + *\tilde{}D \cdot *C)$$

After performing the reductions presented in section 2.2, (namely oscillation analysis - steps 3, 4 - and simplification of the resulting dynamic expressions - step 7) one obtains:

$$*Q' = D \cdot *\tilde{}C + \tilde{}C \cdot *D + *\tilde{}C \cdot *D + *C \cdot *D + *\tilde{}C \cdot *\tilde{}D$$

$$*\tilde{}Q' = \tilde{}D \cdot *\tilde{}C + \tilde{}C \cdot *\tilde{}D + D \cdot *\tilde{}C + D \cdot *C + *D \cdot *C + *D \cdot *\tilde{}C + *\tilde{}D \cdot *\tilde{}C + *\tilde{}D \cdot *C$$

A further interesting phenomenon is the presence of multiple output changes on Q. The difference between a multiple output change and an oscillation is that the later will go on as long as the enabling condition stays true, while a multiple output change stops after a finite number of transitions. The condition for a multiple output change is:

$$MOC\text{-}Q = *Q' \cdot *\tilde{}Q' = D \cdot *\tilde{}C + *\tilde{}C \cdot *D + *C \cdot *D + *\tilde{}C \cdot *\tilde{}D$$

The dynamics of Q guaranteed to be hazard free are termed clean. Eliminating the terms listed in the expression of MOC-Q, results in:

$$*Q'\text{-clean} = \tilde{}C \cdot *D$$
$$*\tilde{}Q'\text{-clean} = \tilde{}D \cdot *\tilde{}C + \tilde{}C \cdot *\tilde{}D$$

The functional behavior is described by means of transition expressions [2] as:

$$| \tilde{}C \cdot (*D + *\tilde{}D) + \tilde{}D \cdot *\tilde{}C | \quad Q \leftarrow \tilde{}C \cdot *D$$

Interpretation:

1. Combination $\tilde{}C \cdot *D$ causes a clean rise of Q.

2. Each of the combinations $\tilde{}D \cdot *\tilde{}C$ and $\tilde{}C \cdot *\tilde{}D$ brings forth a clean drop of Q.

3. For $D \cdot *\tilde{}C$ or $*\tilde{}C \cdot *D$, Q rises with a possible dynamic hazard. Notice that the hazard terms in $*\tilde{}Q'$ are eventually overcome by the regular terms in $*Q'$.

4. For $D \cdot *C$ and $*C \cdot *\tilde{}D$, hazard minterms in $*\tilde{}Q'$ not occurring in MOC, an erroneous value of zero can be latched into Q. This is the effect of a race between the two inputs to the OR gate.

5. For $*C \cdot *D$ and $*\tilde{}C \cdot *\tilde{}D$, hazard minterms occurring both in $*Q'$ and in $*\tilde{}Q'$, the output may experience a multiple change.

### 3.2.3 Essential Hazards in Sequential Circuits

Only the essential hazards intrinsic to the function (as a join of minimal implicants) are considered. Be the excitation functions:

$$Y' = X \cdot Z + Y \cdot \tilde{}X + Y \cdot Z$$
$$Z' = \tilde{}X \cdot \tilde{}Y + X \cdot Z + \tilde{}Y \cdot Z$$

The dynamics of Y and Z are:

$$*Y' = X \cdot \tilde{}Y \cdot *Z + \tilde{}Y \cdot Z \cdot *X + \tilde{}Y \cdot *X \cdot *Z + \tilde{}Y \cdot (*X \cdot *\tilde{}Z + *\tilde{}X \cdot *Z)$$

$$*\tilde{}Y' = Y \cdot \tilde{}Z \cdot *X + X \cdot \tilde{}Y \cdot *\tilde{}Z + Y \cdot (*X \cdot *Z + *X \cdot *\tilde{}Z + *\tilde{}X \cdot *Z + *\tilde{}X \cdot *\tilde{}Z)$$

$$*Z' = \tilde{}Y \cdot \tilde{}Z \cdot *\tilde{}X + \tilde{}X \cdot \tilde{}Z \cdot *\tilde{}Y + \tilde{}Z \cdot *\tilde{}X \cdot *\tilde{}Y + \tilde{}Z \cdot (*X \cdot *\tilde{}Y + *\tilde{}X \cdot *Y)$$

$$*\tilde{}Z' = \tilde{}X \cdot Z \cdot *Y + Y \cdot Z \cdot *\tilde{}X + Z \cdot (*X \cdot *Y + *X \cdot *\tilde{}Y + *\tilde{}X \cdot *Y + *\tilde{}X \cdot *\tilde{}Y)$$

After normalization (step 5 in section 2.2), the dynamic system becomes:

$$*Y' = \tilde{}Y \cdot Z \cdot *X + \tilde{}Y \cdot \tilde{}Z \cdot *\tilde{}X$$
$$*\tilde{}Y' = Y \cdot \tilde{}Z \cdot *X + Y \cdot Z \cdot *\tilde{}X$$

$$*Z' = \tilde{}Y \cdot \tilde{}Z \cdot *\tilde{}X + Y \cdot \tilde{}Z \cdot *X$$
$$*\tilde{}Z' = Y \cdot Z \cdot *\tilde{}X + \tilde{}Y \cdot Z \cdot *X$$

After performing step 7, the system is reduced to:

$$*Y' = Z \cdot *X + \tilde{}Z \cdot *\tilde{}X$$
$$*\tilde{}Y' = \tilde{}Z \cdot *X + Z \cdot *\tilde{}X$$

$$*Z' = \tilde{}Y \cdot *\tilde{}X + Y \cdot *X$$
$$*\tilde{}Z' = Y \cdot *\tilde{}X + \tilde{}Y \cdot *X$$

Interpretation: Due to the presence of races between secondary variable changes and their static previous value, the sequential machine can perform hazardous transitions.

## 4. ANALYSIS OF CIRCUITS WITH NOMINAL DELAY PARAMETERS

The issue of analyzing nondeterministic behavior of circuits with nominal delay parameters is extremely important for hardware verification. A method for analysis of nondeterministic features, taking into account delay parameters, is reported in [6]. The use of the boolean differential calculus in that report provides a tool somehow similar to the dynamic boolean algebra described here. The differences and the similarities between these two approaches are listed in [3].

### 4.1 DBA for Nominal Delay Analysis

The dynamic boolean algebra described in section 2.1 is modified, so as to keep track of propagation delays. Clearly, delay parameters are meaningfull only for dynamic terms. These parameters are generated by associating a delay parameter with each component of the circuit and by summing up delays (along propagation paths) during the expansion of dynamic terms. In the following, the propagation delays are denoted by underscripts: numbers or lower case letters (a, b etc.) standing for actual delays in generic rules.

The inconsistencies associated with the DBA can be solved once the delay parameters are known. It has been shown in [3] that the inconsistencies are due to the

presence of reconvergent fan out branches with a different number of inverters. Thus, the treatment of reconvergent fan out is basic to this extended DBA. Using manipulations (presented in section 2.2), the dynamic expansions of the primary outputs are described in terms of primary inputs and feedback signals. During these manipulations, a reconvergent fan out signal may become manifested only after traversing many intervening gates. Since the type of the gate in which fan out branches reconverge is of paramount importance, the algebraic system must associate with each gate submitted to multiple input changes enough information to enable correct treatment of possible reconvergent fan-outs.

The following cases of multiple input changes can be distinguished for two input gates (gates with more inputs are treated by induction based on the basic rules, since the associativity postulates hold):

i) The output of an AND gate rises when:

   a) Two rising edges are applied on the inputs - the output change is triggered by the last one.

   b) A rise is applied on one input, while the second input drops later than the rising one - the output change is triggered by the rising input.

ii) The output of an AND gate drops when:

   a) Two dropping edges are applied on the inputs - the output change is triggered by the first input change.

   b) A rise is applied on one input, while the second input drops later than the rising one - the output change is triggered by the dropping input.

iii) The output of an OR gate rises when:

   a) Two rising edges are applied on the inputs - the output change is triggered by the earliest one.

   b) A rise is applied on one input, while the second input drops earlier than the rising one - the output change is triggered by the rising input.

iv) The output of an OR gate drops when:

   a) Two dropping edges are applied on the inputs - the output change is triggered by the last one.

   b) A rise is applied on one input, while the second input drops earlier than the rising one - the output change is triggered by the dropping input.

The above case rules are sufficient for performing nominal delay analysis of any AND or OR gate. In fact, the propagation of changes in logical simulators uses quite a similar mechanism. The problem is to integrate these case evaluations into an algebraic system.

The listed local cases can be summarized by a few general rules, relying on only two items of information:

- The presence of mixed direction changes must be recorded, since in this case the relative delay between the input changes dictates whether an output transition will occur. The information used in this case relies on distinguishing active transitions from passive ones. Active transitions are those input changes that dictate the direction of the output change. When mixed direction input changes occur, the transitions opposite to the active ones are termed passive and marked as such. Given the changes $^{*}X$ or $^{*\sim}X$, their passive versions are denoted by $'^{*}X$ and $'^{*\sim}X$.

- The information used for selecting the triggering change shows whether the earliest or the latest input change (both having the same direction) brings forth an output transition. This information is recorded by means of two types of parantheses. The dynamic terms clustered by < and > transmit the earliest incomming active change. Similarly, terms grouped by [ and ] transmit the latest active transition. In the case of mixed type transitions, these parantheses show their relative delay required in order that the change be transmitted to the output. These parantheses have only a sintactic role, enabling correct choice of the triggering change.

Based on this information, the DBA defined in section 2.1 is changed by replacing postulates (iv) with:

1. The first group of dynamic rules is concerned with backward algebraic substitution of signals in the presence of delay parameters:

   a) $^{*}(X_a + Y_b)_c = {}^{\sim}X \cdot {}^{*}Y_{b+c} + {}^{*}X_{a+c} \cdot {}^{\sim}Y + \langle {}^{*}X_{a+c} \cdot {}^{*}Y_{b+c}\rangle + \langle {}^{*}X_{a+c} \cdot {}'^{*\sim}Y_{b+c}\rangle + \langle {}'^{*\sim}X_{a+c} \cdot {}^{*}Y_{b+c}\rangle$

   b) $^{*\sim}(X_a + Y_b)_c = {}^{\sim}X \cdot {}^{*\sim}Y_{b+c} + {}^{*\sim}X_{a+c} \cdot {}^{\sim}Y + [{}^{*\sim}X_{a+c} \cdot {}^{*\sim}Y_{b+c}] + [{}^{*\sim}X_{a+c} \cdot {}'^{*}Y_{b+c}] + [{}'^{*}X_{a+c} \cdot {}^{*\sim}Y_{b+c}]$

   c) $^{*}(X_a \cdot Y_b)_c = X \cdot {}^{*}Y_{b+c} + {}^{*}X_{a+c} \cdot Y + [{}^{*}X_{a+c} \cdot {}^{*}Y_{b+c}] + [{}^{*}X_{a+c} \cdot {}'^{*\sim}Y_{b+c}] + [{}'^{*\sim}X_{a+c} \cdot {}^{*}Y_{b+c}]$

   d) $^{*\sim}(X_a \cdot Y_b)_c = X \cdot {}^{*\sim}Y_{b+c} + {}^{*\sim}X_{a+c} \cdot Y + \langle {}^{*\sim}X_{a+c} \cdot {}^{*\sim}Y_{b+c}\rangle + \langle {}^{*\sim}X_{a+c} \cdot {}'^{*}Y_{b+c}\rangle + \langle {}'^{*}X_{a+c} \cdot {}^{*\sim}Y_{b+c}\rangle$

2. The second group of dynamic rules is concerned with the generation of canonical forms for dynamic minterms, in the presence of propagation delays. In the following, A stands for either $^{*}X$ or $^{*\sim}X$ , B for $^{*}Y$ or $^{*\sim}Y$ and C for either $^{*}Z$ or $^{*\sim}Z$. Note that only active dynamic terms are treated in these rules. Based on rule (iv.2.d), equivalent expressions can be derived for all passive dynamic terms. The treatment of mixed type terms (active and passive) presents difficult problems that were not yet solved in a general way. In the examples, such situations are reduced by using case analysis.

   a) $\langle\langle A \cdot B\rangle \cdot C\rangle = \langle A \cdot B \cdot C\rangle$
   $[[A \cdot B] \cdot C] = [A \cdot B \cdot C]$
   These express a new kind of associativity, holding between dynamic wavefronts in the DBA with delay parameters.

   b) $\langle [A \cdot B] \cdot C\rangle = [\langle A \cdot C\rangle \cdot \langle B \cdot C\rangle]$
   $[\langle A \cdot B\rangle \cdot C] = \langle [A \cdot C] \cdot [B \cdot C]\rangle$

These rules express the distributivity of wavefronts over different types of trigger change selection.

c) $\langle A \cdot [A \cdot B]\rangle = A \quad , \quad [A \cdot \langle A \cdot B\rangle] = A$
These rules express the idempotency of dynamic wavefronts. Note that the dynamic associativity, distributivity and idempotency rules are different from the corresponding static properties.

d) $'\langle {}^*A \cdot {}^*B\rangle = ['{}^*A \cdot '{}^*B]$
$'[{}^*A \cdot {}^*B] = \langle '{}^*A \cdot '{}^*B\rangle$
These show the different treatment of triggering change selection by active and passive transitions.

e) $'({}^*X) = '{}^*X \ , \ '('{}^*X) = {}^*X$
These rules show the convolution of the marking for passive signals, due to the cancellation of any even number of inverters. In practice, reliance on convolution results sometimes in difficulties that have to be treated by case analysis.

3. The last group of dynamic rules is concerned with the treatment of dynamic changes generated by a common input signal (reconvergent fan out). Only the case of rise dynamic terms is presented, since the dropping ones obey the same rules.

a) $\langle {}^*X_a \cdot {}^*X_b\rangle = {}^*X_{min(a,b)}$

$[{}^*X_a \cdot {}^*X_b] = {}^*X_{max(a,b)}$

These express the rules for selecting the triggering change when the local input changes have the same direction (due to reconvergent fan out branches with equal parity).

b) $\langle {}^*X_a \cdot '{}^*X_b\rangle = {}^*X_a$ if a>b and 0 otherwise

$[{}^*X_a \cdot '{}^*X_b] = {}^*X_a$ if a<b and 0 otherwise

These rules express the conditional transmitting of mixed direction input transisions (reconvergent fan out branches with different parity).

c) $\langle '{}^*X_a \cdot '{}^*X_b\rangle = '{}^*X_{max(a,b)}$

$['{}^*X_a \cdot '{}^*X_b] = '{}^*X_{min(a,b)}$

This is required by the dynamic associativity postulate. Indeed, consider the expression $\langle {}^*X_a \cdot '{}^*X_b \cdot '{}^*X_c\rangle$. By associativity, this equals either:

$\langle\langle {}^*X_a \cdot '{}^*X_b\rangle \cdot '{}^*X_c\rangle = \langle {}^*X_a \cdot '{}^*X_c\rangle$ if a>b = ${}^*X_a$ if a>b and a>c , and 0 otherwise.

By a different association, this gives

$\langle {}^*X_a \cdot \langle '{}^*X_b \cdot '{}^*X_c\rangle\rangle = \langle {}^*X_a \cdot '{}^*X_{max(a,b)}\rangle$

that equals the previous result.

## 4.2 Nominal Delay Analysis of Circuits

The analysis of designs assuming nominal delay propagation is very important. Two types of delays occur in practice: transport and inertial ones. In the sequence, only transport delays are considered. This introduces a pessimistic approach to the treatment of short pulses, that are usually quelled by inertial delays. The treatment of inertial delays could be accomodated by recording a threshold propagation width along any dynamic path.

Using the dynamic boolean algebra described in section 4.1, the following sections show the detection of typical problems.

### 4.2.1 Spikes in Combinational Circuits

Spikes are defined as pulses (successive opposite transitions of the same signal) having a width less than some critical parameter. They are not considered an error by themselves, but can induce errors in sequential circuits. Consider the circuit shown in

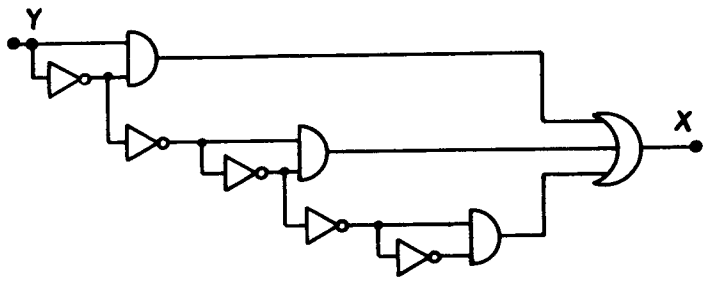


Figure 3: Circuit Producing Spikes

figure 3. Assuming that all INVERTERs introduce a unit delay, while AND and OR gates present no delay, the dynamic behavior is:

$${}^*X = \langle {}^*Y_0 \cdot {}^*Y_2 \cdot {}^*Y_4\rangle + \langle {}^*Y_0 \cdot {}^*Y_2 \cdot '{}^*Y_5\rangle + \langle {}^*Y_0 \cdot '{}^*Y_3 \cdot {}^*Y_4\rangle + \langle '{}^*Y_1 \cdot {}^*Y_2 \cdot {}^*Y_4\rangle + \langle {}^*Y_0 \cdot '{}^*Y_3 \cdot '{}^*Y_5\rangle + \langle '{}^*Y_1 \cdot {}^*Y_2 \cdot '{}^*Y_5\rangle + \langle '{}^*Y_1 \cdot '{}^*Y_3 \cdot {}^*Y_4\rangle = {}^*Y_0 + \langle {}^*Y_0 \cdot '{}^*Y_5\rangle + \langle {}^*Y_0 \cdot '{}^*Y_3\rangle + \langle '{}^*Y_1 \cdot {}^*Y_2\rangle + \langle {}^*Y_0 \cdot '{}^*Y_5\rangle + \langle {}^*Y_2 \cdot '{}^*Y_5\rangle + \langle '{}^*Y_3 \cdot {}^*Y_4\rangle = {}^*Y_0 + {}^*Y_2 + {}^*Y_4$$

$${}^{*\sim}X = [{}^*Y_1 \cdot {}^*Y_3 \cdot {}^*Y_5] + [{}^*Y_1 \cdot {}^*Y_3 \cdot '{}^*Y_4] + [{}^*Y_1 \cdot '{}^*Y_2 \cdot {}^*Y_5] + ['{}^*Y_0 \cdot {}^*Y_3 \cdot {}^*Y_5] + [{}^*Y_1 \cdot '{}^*Y_2 \cdot '{}^*Y_4] + ['{}^*Y_0 \cdot {}^*Y_3 \cdot '{}^*Y_4] + ['{}^*Y_0 \cdot '{}^*Y_2 \cdot {}^*Y_5] = {}^*Y_5 + [{}^*Y_3 \cdot '{}^*Y_4] + ['{}^*Y_2 \cdot {}^*Y_5] + ['{}^*Y_0 \cdot {}^*Y_5] + [{}^*Y_1 \cdot '{}^*Y_2] + ['{}^*Y_0 \cdot {}^*Y_3] + ['{}^*Y_0 \cdot {}^*Y_5] = {}^*Y_1 + {}^*Y_3 + {}^*Y_5$$

Interpretation: a rise of Y is transmitted as three rises of X, after 0, 2 and 4 delay units. The same change on Y also brings forth drops of X, after 1, 3 and 5 delay units. If this circuit feeds a sequential construct that is not able to operate with pulses of only one delay unit width, the final state of that unit is unpredictable. The reader is reminded that the algebraic model used here does not include the effect of inertial delays. Thus, this interpretation of spikes is somehow pessimistic.

### 4.2.2 Instabilities in Sequential Circuits

Under the assumption of complete control over propagation delays, only oscillations and spikes can give rise to nondeterministic behavior in real circuits. The issue of oscillations has been treated in section 2.2. The actual classification of an oscillation as an error or as a desired feature rests eventually on the user.

The result of spikes is the occurrence of instabilities in sequential constructs. This instability is manifested by uncontrolled oscillations or even metastable states, with an unpredictable final value.

In order to cope with nofundamental mode operation, due to short pulses applied on the inputs, the orthogonality postulates have to be modified. Thus, spikes input to a sequential construct can be analyzed and their critical width ascertained. Short pulses are modeled by the expressions:

a) $[*I_0 \cdot '*\sim I_d]$ and $<*\sim I_d \cdot '*I_0>$ define a negative pulse: a drop of I followed within d time units by a rise.

b) $[*\sim I_0 \cdot '*I_d]$ and $<*I_d \cdot '*\sim I_0>$ define a positive pulse: a rise of I followed within d time units by a drop.

These terms are not to be reduced to zero (as required by the orthogonality postulates of the dynamic boolean algebra), and are used for analyzing the circuit's behavior in non fundamental mode operation. The delay parameters of the circuit define the critical width, below which such a term is able to induce instabilities in sequential constructs.

Instabilities in sequential circuits are detected as follows. The algorithm described in section 2.2 (valid for active changes) is performed in the algebraic framework provided by the DBA of section 4.1. This results in expressions of the form:

*X' = A + B . '*~X
*~X' = C + D . '*X

where A , B , C and D do not contain terms in X. The expansion of *~X' is used to replace '*~X in *X' and similarly for '*X in *~X'. The result is:

*X' = A + B . 'C . instability
*~X' = C + D . 'A . instability

The terms A and C show the regular operation of the sequential construct. The term B . 'C lists the conditions for X to experience a rise while a previously attempted drop of X did not latch yet. Similarly, the term D . 'A lists the conditions for X to experience a drop while a previous rise did not yet settle. These two situations correspond to instabilities of the feedback link with all the undesirable associated phenomena. Thus, the complete behavior of X can be described by:

*X' = A
*~X' = C
Instability-X = B . 'C + D . 'A

Due to the features of the algebraic framework used, it is guaranteed that all minterms of A and B . 'C are distinct (and similarly for the minterms of C and D . 'A). Nevertheless, it is possible that the expressions A and D . 'A, respectively C and B . 'C have overlapping terms. These have to be discarded from the functional behavior, since they correspond to instabilities.

The following example shows the operation of the described procedure. Consider the SR flip flop shown in figure 4. Assume that the NAND gates have unit propagation delay. The dynamics of the feedback signal Q are:

$$*Q' = (\sim R + \sim Q) \cdot *\sim S_1 + S \cdot [*R_2 \cdot '*\sim Q_2] + <*\sim S_1 \cdot [*R_2 \cdot '*\sim Q_2]> + R \cdot <*\sim S_1 \cdot '*\sim Q_2> + <*\sim S_1 \cdot ['*\sim R_2 \cdot '*\sim Q_2]> + <'*S_1 \cdot [*R_2 \cdot '*\sim Q_2]>$$

$$*\sim Q' = S \cdot Q \cdot *\sim R_2 + \sim R \cdot *S_1 + Q \cdot [*S_1 \cdot *\sim R_2] + Q \cdot [*S_1 \cdot '*R_2] + Q \cdot ['*\sim S_1 \cdot *\sim R_2] + S \cdot <*\sim R_2 \cdot '*Q_2> + [*S_1 \cdot <*\sim R_2 \cdot '*Q_2>] + R \cdot [*S_1 \cdot '*Q_2] + [*S_1 \cdot <'*R_2 \cdot '*Q_2>] + ['*\sim S_1 \cdot <*\sim R_2 \cdot '*Q_2>]$$

These expressions are reduced by replacing the value of ~Q in *Q' by 1, and similarly for Q in *~Q' (as described in step 7 of 2.2)

$$*Q' = *\sim S_1 + S \cdot [*R_2 \cdot '*\sim Q_2] + <'*S_1 \cdot [*R_2 \cdot '*\sim Q_2]>$$

$$*\sim Q' = S \cdot *\sim R_2 + \sim R \cdot *S_1 + [*S_1 \cdot *\sim R_2] + [*S_1 \cdot '*R_2] + ['*\sim S_1 \cdot *\sim R_2] + R \cdot [*S_1 \cdot '*Q_2] + [*S_1 \cdot <'*R_2 \cdot '*Q_2>]$$

The presented method for detecting instabilities gives the results:

$$*Q' = *\sim S_1 + S \cdot [*R_2 \cdot '*\sim R_4] \cdot \text{instability} + <'*S_1 \cdot [*R_2 \cdot <'*S_3 \cdot '*\sim R_4>]> \cdot \text{instability} + <'*S_1 \cdot [*R_2 \cdot <'*S_3 \cdot *R_4>]> \cdot \text{instability} + <'*S_1 \cdot [*R_2 \cdot <*\sim S_3 \cdot '*\sim R_4>]> \cdot \text{instability}$$

$$*\sim Q' = S \cdot *\sim R_2 + \sim R \cdot *S_1 + [*S_1 \cdot *\sim R_2] + [*S_1 \cdot '*R_2] + ['*\sim S_1 \cdot *\sim R_2] + R \cdot [*S_1 \cdot '*\sim S_3] \cdot \text{instability} + [*S_1 \cdot <'*R_2 \cdot '*\sim S_3>] \cdot \text{instability}$$

The following reductions are performed:

- All instabilities are joined into a separate condition, Instability-Q.
- $<'*S_1 \cdot [*R_2 \cdot <'*S_3 \cdot *R_4>]> = <'*S_1 \cdot [*R_2 \cdot '[*S_3 \cdot '*R_4]]> = <'*S_1 \cdot [*R_2 \cdot '*S_3]>$
- Based on the formula
  $[A \cdot <'B \cdot 'C>] = <[B \cdot 'C] \cdot [A \cdot 'B]> + <[C \cdot 'B] \cdot [A \cdot 'C]> + <'B \cdot [A \cdot 'C]> + <'C \cdot [A \cdot 'B]>$
  one can perform the transformation
  $<'*S_1 \cdot [*R_2 \cdot <'*S_3 \cdot '*\sim R_4>]> = <'*S_1 \cdot [*S_3 \cdot '*\sim R_4] \cdot [*R_2 \cdot '*S_3]> + <'*S_1 \cdot [*\sim R_4 \cdot '*S_3] \cdot [*R_2 \cdot '*\sim R_4]> + <'*S_1 \cdot '*S_3 \cdot [*R_2 \cdot '*\sim R_4]> + <'*S_1 \cdot '*\sim R_4 \cdot [*R_2 \cdot '*S_3]>$
- The terms $<'*S_1 \cdot [*S_3 \cdot '*\sim R_4] \cdot [*R_2 \cdot '*S_3]>$ and $<'*S_1 \cdot '*\sim R_4 \cdot [*R_2 \cdot '*S_3]>$ are covered by the term $<'*S_1 \cdot [*R_2 \cdot '*S_3]>$.

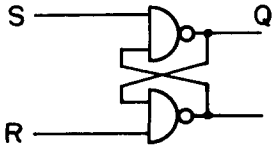


Figure 4: SR Flip Flop

o $\langle'^*S_1 \cdot [^*\sim R_4 \cdot '^*S_3] \cdot [^*R_2 \cdot '^*\sim R_4]\rangle$ is equal to zero, since *R and *~R can not be both active.

o Based on the same formula, another transformation is performed:
$[^*S_1 \cdot \langle'^*R_2 \cdot '^*\sim S_3\rangle] = \langle[^*R_2 \cdot '^*\sim S_3] \cdot [^*S_1 \cdot '^*R_2]\rangle + \langle[^*\sim S_3 \cdot '^*R_2] \cdot [^*S_1 \cdot '^*\sim S_3]\rangle + \langle'^*R_2 \cdot [^*S_1 \cdot '^*\sim S_3]\rangle + \langle'^*\sim S_3 \cdot [^*S_1 \cdot '^*R_2]\rangle$

o $\langle[^*R_2 \cdot '^*\sim S_3] \cdot [^*S_1 \cdot '^*R_2]\rangle$ is covered by $\langle'^*S_1 \cdot [^*R_2 \cdot '^*S_3]\rangle + \langle'^*R_2 \cdot [^*S_1 \cdot '^*\sim S_3]\rangle$

o The term $\langle[^*\sim S_3 \cdot '^*R_2] \cdot [^*S_1 \cdot '^*\sim S_3]\rangle$ is equal to zero, since *S and *~S can not be both active.

o The term $\langle'^*\sim S_3 \cdot [^*S_1 \cdot '^*R_2]\rangle$ is covered by $\langle'^*\sim S_3 \cdot {}^*S_1\rangle$ that corresponds to a pulse of negative width, therefore equals zero.

o The term $\langle'^*S_1 \cdot [^*R_2 \cdot \langle'^*\sim R_4 \cdot {}^*\sim S_3\rangle]\rangle$ is covered by $\langle'^*S_3 \cdot [^*R_2 \cdot '^*\sim R_4]\rangle$

o $[^*S_1 \cdot '^*R_2] = [^*S_1 \cdot '^*R_0] + \langle'^*R_0 \cdot [^*S_1 \cdot '^*R_2]\rangle$ where the second term overlaps with instability minterms.

The final expressions for the dynamics of Q are thus:

$^*Q' = {}^*\sim S_1$

$^*\sim Q' = S \cdot {}^*\sim R_2 + \sim R \cdot {}^*S_1 + [^*S_1 \cdot {}^*\sim R_2] + [^*S_1 \cdot '^*R_0] + [^*\sim R_2 \cdot '^*\sim S_1]$

Instability-Q $= S \cdot [^*R_2 \cdot '^*\sim R_4] + R \cdot [^*S_1 \cdot '^*\sim S_3] + \langle'^*S_3 \cdot [^*R_2 \cdot '^*\sim R_4]\rangle + \langle'^*R_2 \cdot [^*S_1 \cdot '^*\sim S_3]\rangle + \langle'^*S_1 \cdot [^*R_2 \cdot '^*S_3]\rangle$

The transition expression (described in [2]) for the SR flip flop is then:

$| \; ^*\sim S_1 + S \cdot {}^*\sim R_2 + \sim R \cdot {}^*S_1 + [^*S_1 \cdot {}^*\sim R_2] + [^*S_1 \cdot '^*R_0] + [^*\sim R_2 \cdot '^*\sim S_1] \; | \; Q <\!- \; ^*\sim S_1$

Interpretation of the result:

1. A negative pulse on R, shorter than two delay units, while S is either permanently enabled, or rises at least one delay unit before the rise of R, brings forth an instability on Q.

2. A negative pulse on S, shorter than two delay units, while R is either permanently enabled, or rises more than one delay unit before the rise of S, brings forth an instability on Q.

3. The multiple input change: rise of R and rise of S, with a relative delay between them less than one delay unit, brings forth an instability.

4. The multiple input change: rise of R and rise of S, where S occurs at least one delay unit before the rise of R brings forth a drop of Q.

5. A drop of both S and R, such that the drop of R is at least one delay unit before the drop of S, brings forth a drop of Q. Notice that this results finally in a dynamic hazard, since the drop of S forces the output Q back to enabled.

When the flip flop is used as a component of more complex circuits the dynamics of its output are:

$^*Q' = (\sim R + \sim Q) \cdot {}^*\sim S_1$

$^*\sim Q' = Q \cdot S \cdot {}^*\sim R_2 + \sim R \cdot {}^*S_1 + [^*S_1 \cdot {}^*\sim R_2] + [^*S_1 \cdot '^*R_0] + Q \cdot [^*\sim R_2 \cdot '^*\sim S_1]$

Another important condition is the test for occurrencies of spikes on Q. A quite pessimistic approach is taken, by assuming that any instability of Q is able to produce a spike, irrespective of the delay parameters of the sequential circuit fed by Q. The conditions for negative and positive spikes on Q, when the sequential component fed by it is unable to react to pulses of width less than some parameter d are:

NegSpike-Q $= [^*Q'_0 \cdot '^*\sim Q'_d]$ + Instability-Q $= \sim R \cdot [^*\sim S_1 \cdot '^*S_{1+d}] + \langle'^*\sim R_{2+d} \cdot [^*\sim S_1 \cdot '^*S_{1+d}]\rangle + [^*\sim S_1 \cdot '^*S_{1+d} \cdot '^*R_d] + Q \cdot \langle'^*\sim S_1 \cdot [^*\sim R_{2+d} \cdot '^*\sim S_{1+d}]\rangle + S \cdot [^*R_2 \cdot '^*\sim R_4] + R \cdot [^*S_1 \cdot '^*\sim S_3] + \langle'^*S_3 \cdot [^*R_2 \cdot '^*\sim R_4]\rangle + \langle'^*R_2 \cdot [^*S_1 \cdot '^*\sim S_3]\rangle + \langle'^*S_1 \cdot [^*R_2 \cdot '^*S_3]\rangle$

PosSpike-Q $= [^*\sim Q'_0 \cdot '^*Q'_d]$ + Instability-Q $= \sim R \cdot [^*S_1 \cdot '^*\sim S_{1+d}] + \langle'^*\sim R_2 \cdot [^*S_1 \cdot '^*\sim S_{1+d}]\rangle + [^*S_1 \cdot '^*\sim S_{1+d} \cdot '^*Rd] + S \cdot [^*R_2 \cdot '^*\sim R_4] + R \cdot [^*S_1 \cdot '^*\sim S_3] + \langle'^*S_3 \cdot [^*R_2 \cdot '^*\sim R_4]\rangle + \langle'^*R_2 \cdot [^*S_1 \cdot '^*\sim S_3]\rangle + \langle'^*S_1 \cdot [^*R_2 \cdot '^*S_3]\rangle$

## 5. ANALYSIS OF CIRCUITS WITH WINDOW DELAY PARAMETERS

In an industrial environment, it is not feasible to completely control the delay parameters. A common approach is to rely on minimum and maximum delay parameters. An alternative is to associate with each component an average delay and a standard deviation.

### 5.1 DBA for Window Delay Analysis

A further step towards faithfull modeling of digital systems is based on a DBA that takes into account window propagation delays. In the following, it is assumed that delay parameters have a normal distribution. Each basic component is characterized by two parameters:

a) The mean propagation delay $T_m$ , corresponds to the statistical mean of the propagation delays over a considered range of products.

b) The deviation $T_v$ shows the standard deviation from the nominal propagation delay.

Given these two factors, a change transmitted through a logic element is usually considered to occur within the window:

a) Minimal propagation delay defined by:
$T_{min} = T_m - 3 \cdot T_v$

b) Maximal propagation delay defined by:
$T_{max} = T_m + 3 \cdot T_v$

The addition rules for two window propagation delays T' and T" are:
$T_m = T'_m + T"_m$
$T_v^2 = T'_v{}^2 + T"_v{}^2$

For any dynamic term *X, its delay parameters are denoted by $*X_{(Tm,Tv)}$, where Tm is the mean delay and Tv is the variance. The DBA presented in section 4.1 is slightly modified. The rules appearing in the group of dynamic rules (iv.3) are replaced, in order to accomodate window propagation delays:

a) $\langle *X_{(a,s)} \cdot *X_{(b,t)} \rangle$ = if (a - 3 . s > b + 3 . t) then $*X_{(b,t)}$ else if (a + 3 . s < b - 3 . t) then $*X_{(a,s)}$ else $*X_{(\min(a,b),\max(s,t))}$

A pessimistic value for the variance is used when the two windows overlap.
$[*X_{(a,s)} \cdot *X_{(b,t)}]$ = if (a - 3 . s > b + 3 . t) then $*X_{(a,s)}$ else if (a + 3 . s < b - 3 . t) then $*X_{(b,t)}$ else $*X_{(\max(a,b),\max(s,t))}$

Note again the pessimistic approach to overlapping windows.

b) $\langle *X_{(a,s)} \cdot '*X_{(b,t)} \rangle$ = if (a - 3 . s > b + 3 . t) then $*X_{(a,s)}$ else if (a + 3 . s < b - 3 . t) then 0 else $*X_{(a,s)}$

$[*X_{(a,s)} \cdot '*X_{(b,t)}]$ = if (a + 3 . s < b - 3 . t) then $*X_{(a,s)}$ else if (a - 3 . s > b + 3 . t) then 0 else $*X_{(a,s)}$

Notice that hazardous transition are possible. The underscored minterms express a malfunction - a dynamic output change that may occur or not, depending on implementation factors out of control.

### 5.2 Window Delay Analysis of Circuits

The use of simulation in the presence of window delays is characterized by pessimistic results. This is due to the presence of correlations between the propagation delay parameters of different signals, having a common path. Some proposals have been made to correct this shortcoming [4]. The algebraic approach provides a good solution. The following steps of substitution guarantee a result faithfull to the actual behavior of the circuit:

a) All reconvergent fanouts are identified.

b) The dynamic expressions of outputs, fanout and feedback signals are built, in terms of primary inputs and points of reconvergent fanout.

c) Reductions are performed on these intermediary results. Since the reconvergent fanout signals appear as algebraic terms, no correlations between propagation delay parameter are present.

d) Finally, the reconvergent fanout signals are replaced by their expansions in the dynamic system. Thus, only primary inputs and state signals remain.

e) Due to the basic approach of DBA, namely that of discriminating between different working conditions and representing each of them as a separate term, the correct choice of delays becomes straightforward. In contrast to simulation, where it is not known in advance whether the minimal or the maximal propagation delay along some sensitive path is needed, the algebraic model builds the dynamics of signals as disjoint sets of minterms, each corresponding to a different combination of sensitive paths. Each such minterm determines a worst case delay distribution, that can be computed without any interference from other paths.

## 6. EXTENSION OF LOCAL LIMITATIONS INTO GLOBAL ONES

When a complex system, composed of several interconnected basic modules is analyzed, the local limitations associated with each module are extended into global limitations. By that, input restrictions that guarantee functional behavior of all component modules are generated. These global input restrictions are relative to the system's interface signals, in contrast to the local limitations that were stated in term of internal signals.

The extension mechanism is described in the sequence. Be a complex system M that includes the component module m, as shown in figure 4. The inputs of m are denoted by x and its outputs are y. Considering the complex M, the primary inputs are X and the primary outputs are Y. The local behavior of module m is described by:

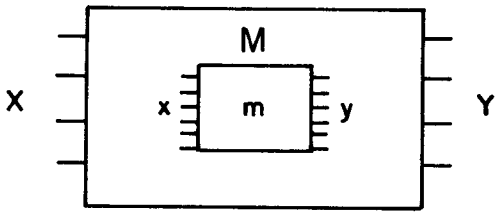


Figure 5: Extension of Limitations

m :: y = f(x) + g(x) . Error-m

where the local limitations of m are summarized by the boolean function g(x)[1]. Substituting the values assigned to the local interfaces of m by the complex system M, results in:

M :: Y = K(y,X) and x = H(X)

where K and H are boolean functions. The behavior of m effects the complex as follows:

M :: Y = K(f(H(X)) + g(H(X) . Error-m) , X)

This can be expanded into:

M :: Y = F(X) + G(X) . Error-M

The local limitations g(x) are thus extended into the global limitations G(X).

### 6.1 Example of Limitations Extension

The use of the RS flip flop of figure 5 as a building component of a more complex circuit, illustrates the modularity of the algebraic approach. Consider the D flip flop (similar to TTL 7474N) shown in figure 6. The system is composed of three SR flip flops. Interpreting the behavior of the component units in the actual configuration, results in:

$*X' = (\tilde{}Z + \tilde{}X) \cdot *\tilde{}C_1$

---

[1] In fact, g(x) describes the input combinations resulting in a nonfunctional behavior of m. The admissible input domain is described by $\tilde{}g(x)$.

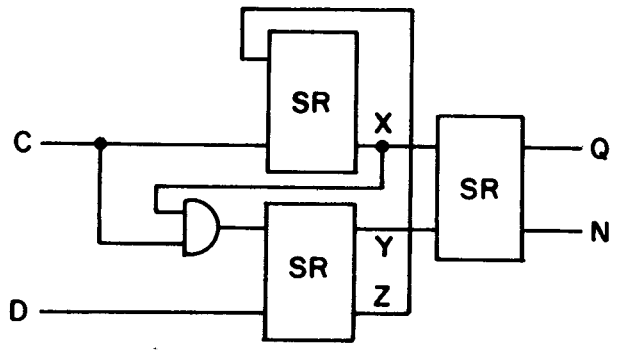


Figure 6: D Flip Flop

$$*\sim X' = C \cdot X \cdot *\sim Z_2 + \sim Z \cdot *C_1 + [*\sim Z_2 \cdot *C_1] + X \cdot [*\sim Z_2 \cdot '*\sim C_1] + [*C_1 \cdot '*Z_0]$$

$$\text{Instability-X} = Z \cdot [*C_1 \cdot '*\sim C_3] + \langle '*Z_2 \cdot [*C_1 \cdot '*\sim C_3]\rangle + C \cdot [*Z_2 \cdot '*\sim Z_4] + \langle '*C_3 \cdot [*Z_2 \cdot '*\sim Z_4]\rangle + \langle '*C_1 \cdot [*Z_2 \cdot '*C_3]\rangle$$

$$*Y' = (\sim D + \sim Y) \cdot *\sim (X \cdot C)_1$$

$$*\sim Y' = \sim D \cdot *(X \cdot C)_1 + X \cdot C \cdot Y \cdot *\sim D_2 + [*\sim D_2 \cdot *(X \cdot C)_1] + Y \cdot [*\sim D_2 \cdot '*\sim (X \cdot C)_1] + [*(X \cdot C)_1 \cdot '*D_0]$$

$$\text{Instability-Y} = D \cdot [*(X \cdot C)_1 \cdot '*\sim (X \cdot C)_3] + \langle '*D_2 \cdot [*(X \cdot C)_1 \cdot '*\sim (X \cdot C)_3]\rangle + X \cdot C \cdot [*D_2 \cdot '*\sim D_4] + \langle '*(X \cdot C)_3 \cdot [*D_2 \cdot '*\sim D_4]\rangle + \langle '*(X \cdot C)_1 \cdot [*D_2 \cdot '*(X \cdot C)_3]\rangle$$

$$*Z' = (\sim X + \sim C + \sim Z) \cdot *\sim D_1$$

$$*\sim Z' = (\sim X + \sim C) \cdot *D_1 + D \cdot Z \cdot *\sim (X \cdot C)_2 + [*\sim (X \cdot C)_2 \cdot *D_1] + Z \cdot [*\sim (X \cdot C)_2 \cdot '*\sim D_1] + [*D_1 \cdot '*(X \cdot C)_0]$$

$$\text{Instability-Z} = D \cdot [*(X \cdot C)_2 \cdot '*\sim (X \cdot C)_4] + \langle '*D_3 \cdot [*(X \cdot C)_2 \cdot '*\sim (X \cdot C)_4]\rangle + X \cdot C \cdot [*D_1 \cdot '*\sim D_3] + \langle '*(X \cdot C)_2 \cdot [*D_1 \cdot '*\sim (X \cdot C)_3]\rangle + \langle '*D_1 \cdot [*(X \cdot C)_1 \cdot '*D_3]\rangle$$

$$*Q' = (\sim Y + \sim Q) \cdot *\sim X_1$$

$$*\sim Q' = \sim Y \cdot *X_1 + X \cdot Q \cdot *\sim Y_2 + [*\sim Y_2 \cdot *X_1] + Q \cdot [*\sim Y_2 \cdot '*\sim X_1] + [*X_1 \cdot '*Y_0]$$

$$\text{Instability-Q} = X \cdot [*Y_2 \cdot '*\sim Y_4] + \langle '*X_3 \cdot [*Y_2 \cdot '*\sim Y_4]\rangle + Y \cdot \langle [*X_1 \cdot '*\sim X_3] + \langle '*Y_2 \cdot [*X_1 \cdot '*\sim X_3]\rangle + \langle '*X_1 \cdot [*Y_2 \cdot '*X_3]\rangle$$

After performing the reductions presented in section 2.2 (namely normalization and elimination of unstable starting total states) the dynamic system becomes:

$$*Q' = \sim Q \cdot D \cdot *C_2 + \sim Q \cdot \langle *C_2 \cdot '*D_4\rangle + \sim Q \cdot [*C_2 \cdot '*\sim D_2]$$

$$*\sim Q' = Q \cdot \sim D \cdot *C_3 + Q \cdot [*C_3 \cdot '*D_2] + Q \cdot \langle *C_3 \cdot '*\sim D_5\rangle$$

$$\text{Instability-Q} = \langle '*\sim D_2 \cdot [*C_2 \cdot '*\sim D_4]\rangle + \langle '*D_1 \cdot [*C_2 \cdot '*D_4]\rangle + \sim D \cdot [*C_3 \cdot '*\sim C_5] + D \cdot [*\sim C_1 \cdot '*C_3] + C \cdot [*D_2 \cdot '*\sim D_4]$$

The transition expression for Q is obtained thus as:

$$| \; D \cdot *C_2 + \langle *C_2 \cdot '*D_4\rangle + [*C_2 \cdot '*\sim D_2] + \sim D \cdot *C_3 + [*C_3 \cdot '*D_2] + \langle *C_3 \cdot '*\sim D_5\rangle \; | \quad Q \leftarrow D \cdot *C_2 + \langle *C_2 \cdot '*D_4\rangle + [*C_2 \cdot '*\sim D_2]$$

Interpretation:

a) When D is enabled and a positive pulse on C shorter than two delay units occurs, or D is disabled and a negative pulse on C occurs, Q enters the instability state.

b) When C is enabled and a negative pulse on D shorter than two delay units occurs, Q enters the instability region.

c) A simultaneous rise of both C and D signals, with D occurring less than two delay units before C but less than one delay unit after C, leads to an instability of Q.

d) A simultaneous rise of C and drop of D, with D occurring less than two delay units before C but at most at the same time as C, induces an instability on Q.

e) A simultaneous rise of both C and D, with D occurring more than two delay units before C, brings forth a rise of Q.

f) A rise of D occurring more than one delay unit after a rise of C, brings forth a drop of Q.

g) A drop of D, occurring after a rise of C, brings forth a rise of Q.

h) A drop of D occurring more than two delay units before the rise of C, brings forth a drop of Q.

i) Finally, when C rises and D is stable, the value of D is stored into Q.

## 7. NONDETERMINISTIC BEHAVIOR AT RTL

The main advantage of using RTL descriptions during functional abstraction [2] is the conciseness achieved by clustering together signals into vectorial constructs. This reduction comes at the price of imposing certain restrictions on the legal sequences of activation.

A main assumption at RTL is that signals can be discriminated according to their use into data and control. It is further assumed that data signals are not used for activating RTL constructs, this task being restricted to control signals. This enforces a special-

ization of signals into carriers of changes and of conditions, although at gate level there is no difference between control and data signals.

For example, a two input AND gate described at logical level is assumed to experience a rise whenever one input is enabled and the second one is rising, or when both inputs are changing. An RTL description of the same gate (used to transfer data from one register to another one), assumes that the inputs are specialized: one of them carries data levels, while the second one is the clock, used to trigger the data transfer. Thus, the dynamics of the same AND gate at RTL take into account only the combination of the data enabled while the clock rises.

For the purpose of functional abstraction, the use of RTL constructs with input transitions not included in the high level description introduces problems similar to the presence of nondeterministic behavior. Thus, the use of RTL descriptions introduces a new dimension of nondeterminism: illegal activations.

### 7.1 Analysis of Illegal Activations

In order to obey the specialization of signals into data and control, typical for RTL descriptions, the circuit level behavior (including nominal or window delay parameters) is analyzed in order to extract the following data:

a) The setup time is defined as the minimal admissible delay between the latest data change and the occurrence of a control change that triggers the data transfer. This minimal delay guarantees that the changes introduced by the data signals settle down before the activation of a control input.

b) The hold time is defined as the minimal admissible delay between the triggering of a data transfer and any further change of the data input, such that this data change is not included in the transfer.

Equivalent parameters may be defined at circuit level, in order to guarantee deterministic operation of a sequential construct. At RTL, the meaning of these parameters is extended, so as to deal with illegal activations.

Given the functional behavior of a sequential construct and assuming that the set of signals {Dj} have been specialized as data and the rest, forming the set {Ci}, are used for activation, the first step in generating a RTL description is to eliminate any dynamic minterm containing changes on data signals. In order for this description to be faithfull to the original one, the hold and the setup time are derived from dynamic minterms refering to multiple changes occurring simultaneously on control and data signals:

- Any dynamic minterm that requires the activation on $Ci_a$ to occur after a transition of the data signal $Dj_b$ defines a setup time for the signal Dj with respect to the control signal Ci :
  Setup(Dj, Ci) = b - a

- Any dynamic minterm that requires the activation of $Ci_a$ to occur before a transition of the data signal $Dj_b$ defines a hold time for signal Dj with respect to control Cj :
  Hold(Dj, Ci) = a - b

- For each pair of signals Dj and Ci , the final setup and hold times are computed as the maximal delays over all the minterms considered.

Consider for example the D flip flop presented in figure 6. Assuming that the signal D has been specialized for carrying only level information, while signal C is used for activation of the flip flop, the usable RTL behavior is:

$$| D \,.\, {}^{*}C_2 + \tilde{}D \,.\, {}^{*}C_3 \; | \quad Q \;\leftarrow\!- D \,.\, {}^{*}C_2$$

The following multiple input transitions are considered:

1. $\langle {}^{*}C_2 \,.\, '{}^{*}D_4 \rangle$ defines a setup time of 2 time units for D with respect to the rise of C.

2. $[{}^{*}C_2 \,.\, '{}^{*}\tilde{}D_2]$ defines a hold time of 0 time units for D with respect to the rise of C.

3. $[{}^{*}C_3 \,.\, '{}^{*}D_2]$ defines a hold time of 1 time unit for D with respect to *C.

4. $\langle {}^{*}C_3 \,.\, '{}^{*}\tilde{}D_5 \rangle$ imposes a setup time of 2 time units for D with respect to *C.

5. Thus, the final setup time for D with respect to C is 2 time units, while the final hold time for the same pair of signals is 1 time unit.

The limitations due to illegal activations of RTL descriptions are treated similary to those describing circuit level nondeterminism. In particular, the algorithm for extending limitations of a module to global limitations of a complex system (chapter 6) can be used without any modification.

## 8. CONCLUSION

In this paper, an algebraic method for analyzing digital circuits for the presence of nondeterministic behavior has been shown. The method gives results similar to those of existing algorithms for circuits with unbounded delays. For the more practical case of nominal and window delay analysis, there are no known methods for performing this task. Although the cost of a design verification system based on the proposed approach is expected to be several orders of magnitude higher than that of classical simulation (for a single input vector), the quality of the results justifies the investment.

## REFERENCES


[1] A. D. Friedman and P. R. Menon - Theory and Design of Switching Circuits - Computer Science Press, 1975.

[2] S. Leinwand and T. Lamdan - Design Verification Based on Functional Abstraction - The 16th Design Automation Conference, San Diego, 1979.

[3] S. Leinwand and T. Lamdan - Dynamic Boolean Algebras - unpublished report.

[4] B. Magnhagen - A Comparison Between a Minimum - Maximum Delay and a Most Likely Delay Logic Simulator - Int. Conf. on CAD and Manufacture of Electronic Components, Circuits and Systems, Brighton, July 1979.

[5] E. J. McCluskey - Introduction to the Theory of Switching Circuits - Mc. Graw Hill, N. Y. 1965.

[6] A. Thayse - La Detection des Aleas dans les Circuits Logiques au Moyen du Calcul Differentiel Booleen - Digital Processes, vol. 1 (1975), pg. 141 - 169.