UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

An Optimistic Ternary Simulation of Gate Races

C.-J. Seger
J.A. Brzozowski

CS-86-22

June, 1986

An Optimistic Ternary Simulation of Gate Races

C-J. Seger and J.A. Brzozowski

Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

ABSTRACT

The detection of timing problems in digital networks is of considerable importance. In particular it is desirable to have efficient methods for discovering critical races and hazards. Unfortunately, commercial simulators rarely provide such facilities; in fact, the simulators usually assume that all the gate delays are exactly equal. In contrast to this, binary race analysis frequently assumes that gate delays can be arbitrarily large, though finite. An exception to this is the Almost-Equal-Delay race model, where gates have different delays, but the difference between any two delays cannot be arbitrary. The difficulty with the use of this model is that it is computationally very inefficient. In this paper we define a new ternary model which is very closely related to the binary Almost-Equal-Delay model. Moreover, the ternary model is considerably more efficient, as efficient as the unit delay model; consequently, it could easily be incorporated in simulators.

† This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant A0871.

1. Introduction

The ideas to be presented will be introduced by examples; note that these examples were chosen for their simplicity and not necessarily their usefulness. Consider the circuit of Fig. 1; the behavior of this circuit is governed by the following equations:

$$Y_1 = x', \quad Y_2 = xy_1, \quad Y_3 = y_2 + y_3,$$

where, for each gate, y_i denotes the present output of gate i and Y_i, called the *excitation* of gate i, gives the value of the boolean function computed by gate i. When $y_i = Y_i$ the gate has no tendency to change, and we say that it is *stable*. If $y_i \neq Y_i$, then the gate is *unstable* and the output y_i tries to change to Y_i. This change does not always happen because it is possible that an earlier change in some other gate may cause Y_i to become equal to y_i. This corresponds to the fact that the delay associated with gate i is inertial, in the sense that short periods of instability are tolerated without any change. Now suppose the circuit of Fig. 1 is started with $x = 0$ and $y = y_1, y_2, y_3 = 1, 0, 0$, which is a stable state. What happens when x is changed to 1? This is the type of problem that we are concerned with in this paper.

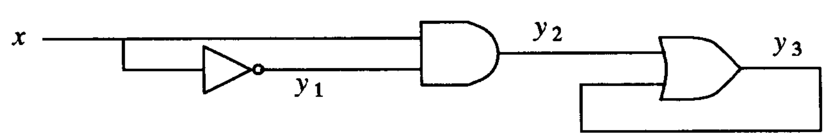

Figure 1. Network N_1.

Any state in which more than one gate is unstable at the same time is called a *race*. The outcome of a race depends very much on the model used. Commercial simulators like SILOS [S85] or MOSSIM [B84] use the "unit delay" (UD) model, in which all gates are assumed to have exactly equal delays. In the state $x = 1$, $y = 1\,0\,0$ (commas omitted from $1, 0, 0$ for simplicity) gates 1 and 2 are unstable, and will both change. Consequently the next state is $0\,1\,0$. Now gates 2 and 3 are unstable, and state $0\,0\,1$ is reached. This state is stable. In summary, the unit delay model predicts that the final outcome of this transition is the stable state $0\,0\,1$; see Fig. 2, where unstable gates are indicated by underlining.

In contrast to this, the General Multiple Winner (GMW) model [BY79] permits the possibility of unequal delays. The model assumes that any nonempty subset of the set of unstable gates can change in any race. In Fig. 3 we show the GMW analysis of network N_1. If gate 1 is faster in

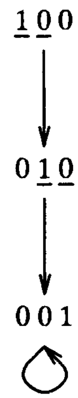

Figure 2. Unit delay analysis of network N_1.

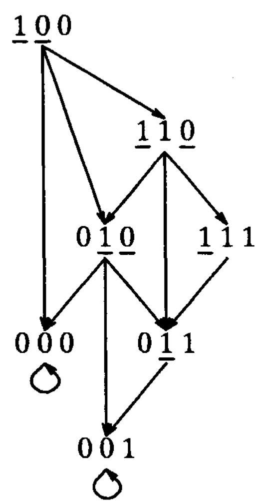

Figure 3. Race analysis of N_1 according to the GMW model.

state 1 0 0, the state 0 0 0 may be reached. This is also a stable state, and represents a likely outcome. This shows that the UD model is inaccurate. In fact, the only justification of the use of the UD model appears to be its simplicity. On the other hand, the GMW model is too "pessimistic" as we show below.

Consider the circuit of Fig. 4 started in the stable state $x = 0$, $y = 1\,0\,1\,0\,1\,0\,0$, and let x change to 1. It is reasonable to assume that y_5 will change to 0 before y_4 changes to 1, and that the only likely final outcome is the stable state 0 1 0 1 0 0 0. However, the GMW model will

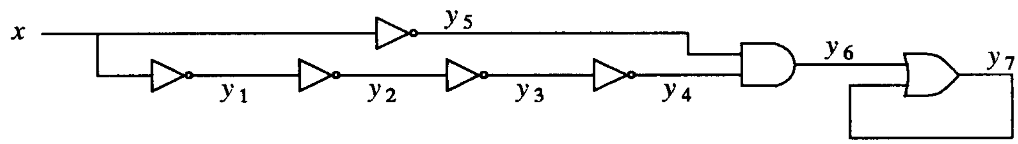

Figure 4. Network N_2.

allow the possibility that y_4 changes before y_5 and that the state 0 1 0 1 0 0 1 is also reachable.

The "Almost-Equal-Delay" (AED) model, formally defined later, represents an attempt to reduce the above pessimism by assuming that (roughly speaking) no gate delay exceeds the sum of any two gate delays. Thus the AED model predicts only the state 0 1 0 1 0 0 0 as the outcome for the circuit of Fig. 4, because the delay of gate 5 is assumed to be smaller than the sum of delays of gates 1, 2, 3 and 4. Note that the AED model (extended to multiple winners) is clearly more informative than the UD model, since it always includes the outcome predicted by the UD model.

In this paper we define a new stepwise AED model and compare it with the original AED model. We show that the two models are equivalent with respect to their capability of predicting the outcome of a transition. However, the new stepwise model has the attractive property of being closely related to a time scale; hence, it is possible to obtain some timing information from the analysis.

A major difficulty with models such as the GMW and AED is that the number of steps involved can be exponential in the number of gates. To overcome this, ternary simulation has been used [JMV69, B83] to perform the analysis more efficiently. However, the ternary simulation algorithm, as suggested by Eichelberger [E65], corresponds to a binary GMW analysis of the network assuming both gate and wire delays [BS85,BS86], and is therefore even more pessimistic than the GMW analysis. To overcome this, we propose a new ternary algorithm, called the ternary almost-equal-delay (TAED) model. This is a stepwise algorithm of the same complexity as the UD method, but it takes into account possible races. In fact, we show that the TAED model is closely related to the stepwise AED model.

2. The Binary Almost-Equal-Delay Model

The "almost-equal-delay" (AED) model was originally defined using the "single-winner" concept for simplicity [BY75, BY76]. In this paper we consider the more general "multiple-winner" version of the model which is described below. This generalization is quite straightforward.

The basic idea is illustrated in Fig. 5. Suppose network N_1 of Fig. 1 starts in state 1 0 0 at time 0 with gates 1 and 2 unstable as shown in Fig. 5. Suppose now that gate 2 wins the race at time t and that, as a result of that change in gate 2, gate 3 becomes unstable. Under the almost-equal-delay assumption, it is unreasonable to let gate 3 win the new race between gates 1 and 3, since gate 1 has already been "waiting" for t units of time. The model will remember this fact and will only permit gate 1 to change in state 1 1 0, predicting the next state as 0 1 0. Informally, we can consider that at time 0 a "race unit" has started involving gates 1 and 2. No other gate can "enter" this race unit until all the gates in the original unit are somehow "satisfied". A gate becomes satisfied if it either changes or becomes stable as a result of some other change. These ideas will be made precise below.

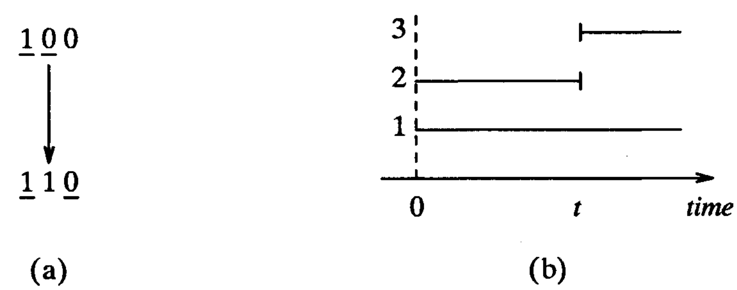

Figure 5. Illustrating the "AED" idea:
(a) possible transition; (b) timing diagram.

Consider a network with n gates whose outputs are given by the vector $y = y_1, \cdots, y_n$, and with m inputs given by $x = x_1, \cdots, x_m$. With each gate y_i we associate its boolean function $Y_i(x, y)$. In the AED model a certain amount of previous race history is necessary to determine the outcome of a race. We therefore define a *race state* to be an ordered pair $<y, S>$, where $y \in B^n$ denotes a state of the network, and S denotes the set of gates that are unstable in that state and are candidates to change according to the AED model, as explained below.

Given any gate state $y = y_1, \cdots, y_n$, and any subset P of the set $\{1, \cdots, n\}$ of gate subscripts, we define $y^{(P)}$ to be the vector obtained from y by complementing all the y_i such that i is in P.

We now define a set K of race states and a binary relation ρ on K. We begin with a stable state x, y^0, and change the input to $\tilde{x}$. The initial race state of the network is $<y^0, U(y^0)>$, where, for any y, $U(y)$ is the set of subscripts of unstable gates in state y, i.e.

$$U(y) = \{j : Y_j(\tilde{x}, y) \neq y_j\}.$$

(Note that the input will be held constant at the value $\tilde{x}$ for the rest of the analysis; thus the dependency of U on $\tilde{x}$ is suppressed.)

The set K and the relation ρ are now defined inductively as follows:

Basis: $<y^0, U(y^0)> \in K$.

Induction Step: Given $<y, V> \in K$,

1. If $V = \varnothing$, then $<y, V> \rho <y, V>$.
2. If $V \neq \varnothing$, then for each nonempty subset P of V compute $W_P = (V - P) \cap U(y^{(P)})$.
   (a) If $W_P = \varnothing$, then $<y^{(P)}, U(y^{(P)})> \in K$ and $<y, V> \rho <y^{(P)}, U(y^{(P)})>$.
   (b) If $W_P \neq \varnothing$, then $<y^{(P)}, W_P)> \in K$ and $<y, V> \rho <y^{(P)}, W_P)>$.

Nothing else is in K or ρ.

The set K defined above represents the set of all possible race states reachable from the initial race state $<y^0, U(y^0)>$, and the relation ρ describes the immediate successor race states. In particular, given $<y, V>$, if $V = \varnothing$ then y is a stable state, and this is indicated by stating that $<y, \varnothing>$ is the only successor of itself. Next, if $V \neq \varnothing$, it turns out by this definition that V will always represent a subset of the gates which are unstable in y. The model now assumes, according to the multiple winner principle, that the gates in any nonempty subset P of V may all change and so state $y^{(P)}$ may be reached. The gates in P are considered to have completed the race they were involved in. As for the gates in $V - P$, one of two things can happen. Either a gate y_i remains unstable in $y^{(P)}$, i.e. $i \in W_P$, or the instability of y_i is removed when y changes to $y^{(P)}$. In the latter case, $i \notin W_P$. Now, if $W_P = \varnothing$, all the instabilities have been removed from V, one way or another. Thus we consider the previous race unit as completed, and now start a new one by entering the race state $<y^{(P)}, U(y^{(P)})>$ where each unstable gate of $y^{(P)}$ has an equal chance of winning. If $W_P \neq \varnothing$, then some of the instabilities of V still remain unsatisfied; these are precisely all the gates in W_P. These gates are given preference over any new instabilities that may have been introduced by the change from y to $y^{(P)}$.

We illustrate this definition by the following example. Consider the network N_1 of Fig. 1 in state $x = 0$, $y^0 = 1\,0\,0$ which is stable. We now let $\tilde{x} = 1$ and note that $U(y^0) = \{1, 2\}$. Thus the network starts in the race state $<1\,0\,0, \{1, 2\}>$. The graph of the relation ρ is shown in Fig. 6, where each edge has a label showing the gates that change during the transition corresponding to the edge. (For now, ignore the * marking on some edges.)

The relation graph shown in Fig. 6 describes all the possible states that can be reached during the transition from the initial state $y^0 = 1\,0\,0$, when the input changes from 0 to 1. However, we are normally interested only in the "final" outcome of the transition. Thus we will consider only the set of cycles in the relation graph of ρ. Note that, by the definition of ρ, there cannot be any transient cycles in the graph. (A cycle is said to be transient if there is a gate that has the same value and is unstable in all the states of the cycle [BY79].) In this case the network may end up in either one of two possible cycles, namely the stable states 0 0 0 and 0 0 1. Thus the initial race is *critical* because the final outcome depends on the relative delays in the circuit.

For technical reasons which will become clearer later, we mark certain edges by a * in the graph of ρ as follows. All edges of the type $(<y, \emptyset>, <y, \emptyset>)$ are marked. Also, every edge $(<y, V>, <y^{(P)}, U(y^{(P)})>)$ added to ρ by Rule 2(a), i.e. with $W_P = \emptyset$, is marked. No other edges are marked. (Note that if an edge is introduced because of Rule 2(a), that same edge cannot be also generated by Rule 2(b). This follows because y, V, and $y^{(P)}$ uniquely determine W_P.) For example, Fig. 6 shows all the marked edges for the previous example.

3. The Stepwise AED Model

In this section we introduce a new binary race model, very closely related to the AED model of Section 2. This model, which will be called the *stepwise AED* model, has the advantage of showing more clearly certain timing information. Also·the stepwise model will be used later to establish a related ternary model.

To obtain more timing information from ρ we now define a new relation R derived from ρ as described below. Intuitively, one can interpret this relation R in the following way. Suppose that all the gates have approximately the same delay, $\delta \pm \epsilon$. Then any race unit lasts for

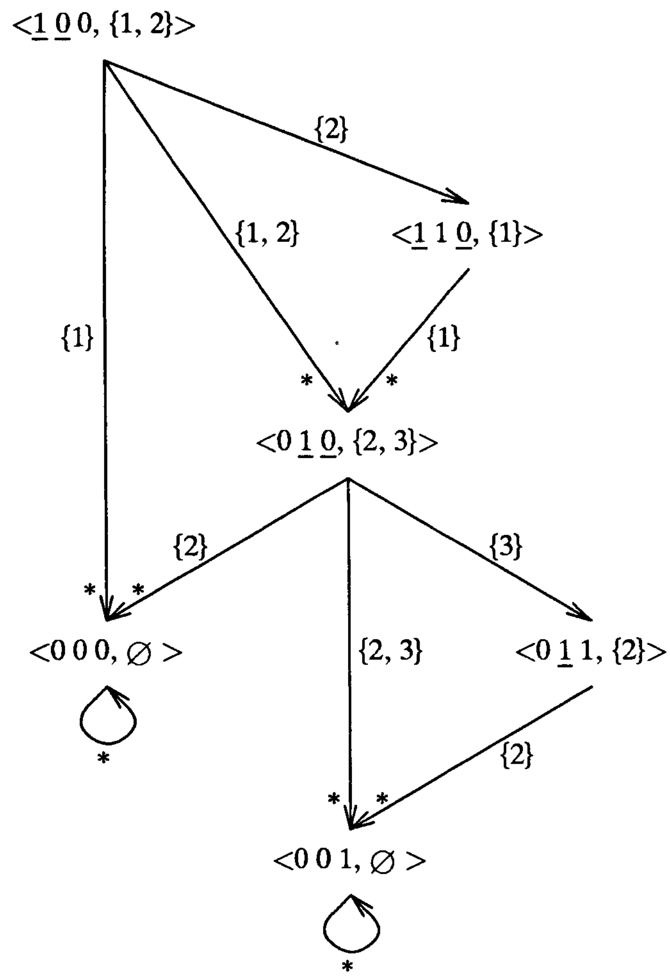

Figure 6. Race analysis of N_1 according to the AED method.

approximately δ units of time. Thus a transition between two distinct states related by R represents roughly δ units of time. In contrast to this, consider the relation ρ of Fig. 6 and the sequence <1 0 0, {1, 2}>, <1 1 0, {1}>, <0 1 0, {2, 3}> when consecutive race states are related by ρ. Here the first transition takes about δ units of time, whereas the second — only ϵ units of time, because gate 1 became unstable in the first state and lagged behind gate 2 by a small amount of time.

Note that the above intuitive explanation is adequate as long as ϵ is much smaller than δ. Note also that the * -marking introduced in Section 2 has the following meaning: A transition is marked iff that transition completes a race unit, or it is a self-loop on a stable state.

Formally, the relation R is defined on a subset Q of the set K of race states as follows:

$$Q = \{<y^0, U(y^0)>\} \cup$$
$$\{<y, V> : \text{there is a marked edge into } <y, V> \text{ in the graph of } \rho\}$$

For $<y, V>, <\bar{y}, \bar{V}> \in Q$, we define $<y, V> R <\bar{y}, \bar{V}>$ iff there exists a path from $<y, V>$ to $<\bar{y}, \bar{V}>$ in the graph of ρ, such that only the last edge of the path is marked. Note, in particular, that $<y, \emptyset> R <y, \emptyset>$ for all the stable states $<y, \emptyset>$ in Q.

To illustrate this consider the example of Fig. 6. We find

$$Q = \{<1\,0\,0, \{1,2\}>, <0\,1\,0, \{2,3\}>, <0\,0\,0, \emptyset>, <0\,0\,1, \emptyset>\},$$

and the reader can verify that R is as shown in Fig. 7.

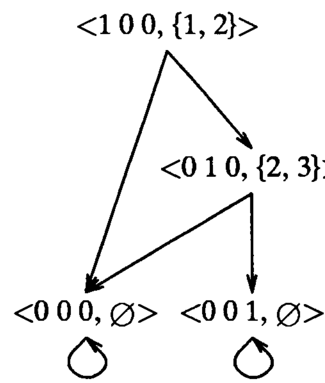

Figure 7. Relation R derived from ρ of Fig. 6.

The reader should note that the knowledge of a race state $<y, V>$ alone does not provide sufficient timing information about race units. Consider the network N_3 described by the following equations:

$$Y_1 = xy_2', \qquad Y_2 = x, \qquad Y_3 = y_2$$

started in the stable state $x = 0$, $y = 0\,0\,0$ and with the new input $\tilde{x} = 1$. (We use some strange gates in this network in order to keep the example small; similar phenomena occur in more realistic larger circuits.) In Fig. 8 we show the graph of the relation ρ for this transition. It is easy to verify that the states $<0\,0\,0, \{1,2\}>$, $<0\,1\,0, \{3\}>$, and $<1\,1\,0, \{1,3\}>$ are all in Q, and that $<0\,0\,0, \{1,2\}> R <0\,1\,0, \{3\}>$ because the change in gate 2 completes the race unit started in $<0\,0\,0, \{1,2\}>$. When state $<0\,1\,0, \{3\}>$ is reached in this way the instability of gate 3 represents a new race unit.

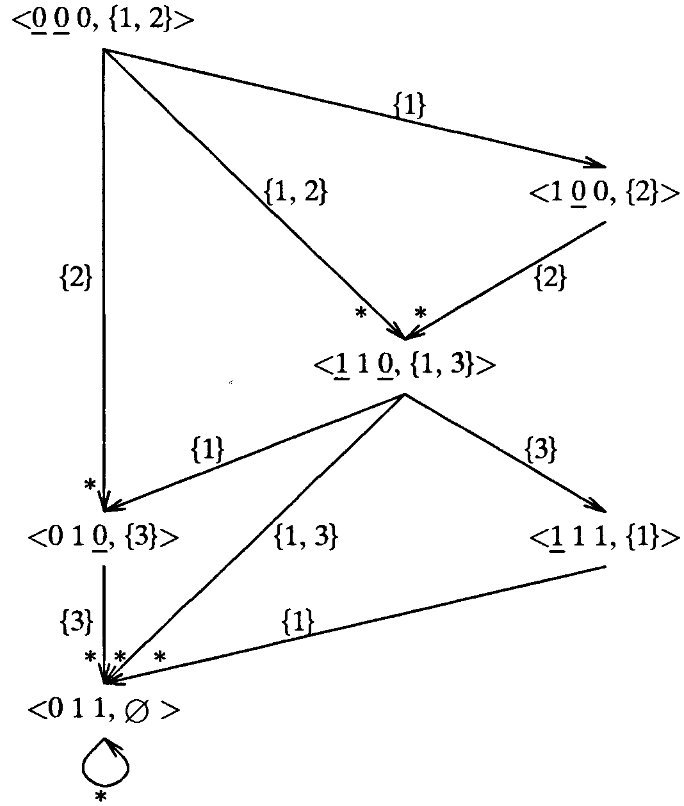

Figure 8. AED analysis of network N_3.

However, despite the fact that the states <1 1 0, {1, 3}> and <0 1 0, {3}> are both in Q and <1 1 0, {1, 3}> ρ <0 1 0, {3}>, they are not related by R. The reason for this is that the transition from <1 1 0, {1, 3}> to <0 1 0, {3}> does not represent the completion of a "race unit", because gate 3 is still unstable, and that condition started in state <1 1 0, {1, 3}>. By requiring that the *last* edge in the ρ-path be marked, we make sure that the R relation holds only between states reachable by complete race units.

We will show that the R relation contains all the "useful" information of the old AED relation ρ. Assume the network is started in the stable state x, y^0 and that the new input is $\tilde{x}$. For convenience, let G_ρ be the directed graph of the relation ρ, and G_R be the directed graph of the relation R. Note that G_ρ and G_R depend not only on the network, but also on the initial state and the input change.

Proposition 1: In the graph G_R every node $<y, V> \in Q$ is reachable from the initial race state $<y^0, U(y^0)>$.

Proof: First note that, by the definition of ρ, all nodes in G_ρ are reachable from the initial race state $<y^0, U(y^0)>$. Consider $<y, V> \in Q$; either $<y, V>$ is the initial state or it has a marked edge $(<\bar{y}, \bar{V}>, <y, V>)$ in G_ρ into it. Obviously, if it is the initial state there is nothing to prove. Otherwise, study the state $<\bar{y}, \bar{V}>$. Note that $<\bar{y}, \bar{V}>$ must be reachable from $<y^0, U(y^0)>$ in G_ρ, and hence there must exist a path in G_ρ from $<y^0, U(y^0)>$ to $<y, V>$ ending with a marked edge. In this path some other edges might also be marked, but that will only mean that we can reach $<y, V>$ in G_R by going through more than one node. Hence the claim follows. □

Proposition 2: The stable states of G_ρ are the same as the stable states of G_R.

Proof: All stable states of G_ρ are of the form $<y, \emptyset>$, with an edge $(<y, \emptyset>, <y, \emptyset>)$. Note that this type of edge in G_ρ is always marked and hence will also exist in G_R. Conversely, every stable state of G_R is a stable state of G_ρ by construction of R. □

For the next proposition we need a new concept. Given any cycle $<y^1, V^1>, \cdots, <y^r, V^r>, <y^1, V^1>$ in the graph G_ρ, we call a race state $<y^i, V^i>$ *initiating in this cycle* iff the edge $(<y^{i-1}, V^{i-1}>, <y^i, V^i>)$ is marked, where $<y^0, V^0>$ is interpreted as $<y^r, V^r>$.

Proposition 3: Every simple cycle C in G_ρ of length ≥ 2 must contain at least two distinct initiating states in C.

Proof: For any edge $(<y^{i-1}, V^{i-1}>, <y^i, V^i>)$ in such a cycle, either the edge is marked (Case 2(a)) or V^i is a proper subset of V^{i-1} (Case 2(b)). Thus from any race state in C we must reach an initiating state in C in a finite number of steps. Starting from any initiating state $<y, V>$ in C we eventually must reach another initiating state $<\hat{y}, \hat{V}>$ in C. We argue that these two race states must be different. This is because the edge leaving $<y, V>$ involves changing at least one variable, say y_i, from V. This variable cannot change again until a new initiating state in C is reached. Thus $\hat{y}$ and y differ at least in y_i. □

We now wish to show that the cyclic behavior predicted by ρ is in some sense preserved in R. For any cycle C of race states in the graph G_ρ, let $\Pi(C)$ be the sequence of initiating states of C in the same order as in C. The AED model of Section 2 (relation ρ) and the stepwise model of this section (relation R) are related by the following result. Assume that the network is started in the stable state x, y^0 and the input changes to $\tilde{x}$. Compute ρ and R starting from this condition.

Theorem 1 For every cycle C in the graph G_ρ there exists a cycle $\Pi(C)$ in the graph G_R. Conversely, for every cycle D in G_R there exists a cycle C in G_ρ such that $D = \Pi(C)$. Furthermore, given two corresponding cycles C_ρ in G_ρ and C_R in G_R, and any variable y_i, either y_i has the same binary value in all the states of C_ρ and C_R or y_i oscillates (assumes the values 0 and 1) in both C_ρ and C_R.

Proof: Consider a cycle C_ρ in G_ρ. If the length of the cycle is 1, i.e. the state is a stable state, then the result follows immediately from Proposition 2. Hence assume the length of the cycle is ≥ 2. By the definition of an initiating state in a cycle and the definition of $\Pi(C_\rho)$, it follows that if $<y, V>$ and $<\bar{y}, \bar{V}>$ are any two consecutive race states in $\Pi(C_\rho)$, then the path in C_ρ from $<y, V>$ to $<\bar{y}, \bar{V}>$ will have only the last edge marked. Therefore we can conclude that $<y, V> R <\bar{y}, \bar{V}>$. This, together with Proposition 3 shows that $\Pi(C_\rho)$ is a cycle in G_R of length ≥ 2.

Conversely, let D_R be a cycle of G_R. According to the definition of R, for any two consecutive race states $<y, V>$ and $<\bar{y}, \bar{V}>$ in D_R, there must exist a path in the graph G_ρ from $<y, V>$ to $<\bar{y}, \bar{V}>$ with only the last edge marked. Hence there must exist in G_ρ a cycle C_ρ corresponding to the cycle D_R in G_R, such that $\Pi(C_\rho) = D_R$.

The final part of Theorem 1 follows immediately from the observation that a gate can change at most once between two initiating states. □

4. Race Units

The definition of Q and R as given above can be simplified since it is possible to avoid using race states. This follows because, in any state $<y, V>$ in Q, the set V is uniquely determined by y ($V = U(y)$). Below we give a different algorithm for computing Q and R, where we do not explicitly form the graph of ρ. The reader can easily verify that the result is equivalent to the previous definition.

As before, we consider a gate network defined by x, y, and Y. A sequence, $(z^0, S^0), (z^1, S^1), \cdots, (z^k, S^k)$, $k \geq 0$, is called a *race unit* if

i) $z^0 \in B^n$ is a gate state and $S^0 = U(z^0)$,

ii) z^{i+1} is state z^i with all the gate outputs in the set Δ^i complemented, where Δ^i is any nonempty subset of S^i. Also $S^{i+1} = (S^i - \Delta^i) \cap U(z^{i+1})$, and

iii) $S^k = \varnothing$.

Note that $S^0 \supset S^1 \supset \cdots \supset S^k$, where $\supset$ denotes proper containment.

We compute the set Z and also the relation σ on Z inductively as follows:

Algorithm 1: Let x , y^0 be a stable total state and let the input change to $\tilde{x}$.

Basis $y^0 \in Z$.

Induction step: For every $y \in Z$, if there exists a race unit beginning with $(y, U(y))$ and ending with $(z, \varnothing)$, then $z \in Z$ and $y \sigma z$.

Now to obtain Q and R , replace each state $y \in Z$ by $<y, U(y)>$ in Q and let $<y, U(y)> R <z, U(z)>$ iff $y \sigma z$. From now on we will be working with Z and σ rather than with Q and R when referring to the stepwise AED model.

To illustrate these ideas, consider network N_1 of Fig. 1, as before started in the stable state $x = 0$, $y^0 = 1\,0\,0$ and with the new input $\tilde{x} = 1$. Since $U(y^0) = \{1, 2\}$, there are only three possible race units starting with $(1\,0\,0, \{1, 2\})$:

1 $(1\,0\,0, \{1, 2\})\ (0\,0\,0, \varnothing)$,

2 $(1\,0\,0, \{1, 2\})\ (1\,1\,0, \{1\})\ (0\,1\,0, \varnothing)$, and

3 $(1\,0\,0, \{1, 2\})\ (0\,1\,0, \varnothing)$.

Therefore we add the states $0\,0\,0$ and $0\,1\,0$ to Z, and add the pairs (100, 000) and (100, 010) to the relation σ. It is easy to show that from the state $0\,0\,0$ we can only go to $0\,0\,0$, and from state $0\,1\,0$ we can end up in $0\,0\,0$ or $0\,0\,1$. From $0\,0\,1$ we can only reach. $0\,0\,1$. In summary, $Z = \{100, 000, 010, 001\}$, and the relation σ over Z is as illustrated by the graph of Fig. 9. Note the correspondence between Fig. 9 and Fig. 7.

It is now convenient to interpret Fig. 9 as a nondeterministic finite state machine with initial state $1\,0\,0$, and δ as its only input letter. After one race unit (i.e. after roughly δ units of time), we may nondeterministically reach the states $0\,0\,0$ or $0\,1\,0$, etc. Let $Z^0 = \{y^0\}$ and let Z^i be the set of states of Z reachable after i steps, i.e. $Z^i = \{z : y^0 \sigma^i z\}$. Note that these sets are the same as the subsets constructed by the subset construction when converting the nondeterministic finite state machine to a deterministic finite state machine. In Fig. 10 we show the deterministic finite state machine corresponding to the nondeterministic machine of Fig. 9.

We close this section with a discussion of some limitations of the stepwise AED model. One of the basic assumptions in this model is that delays are only associated with gates, and that delays are inertial in their nature. In many real circuits these assumptions can be well justified. However, there is also a danger that the model may become unrealistic under certain conditions. The model is only accurate as long as races from

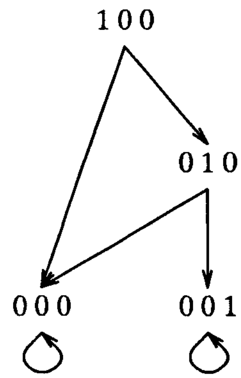

Figure 9. Stepwise AED analysis of N_1.

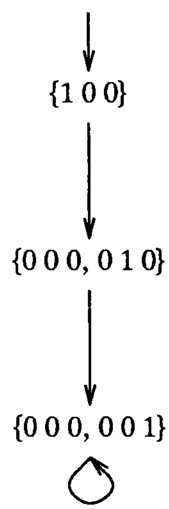

Figure 10. Corresponding deterministic finite state machine to Fig. 9.

different race units do not get "mixed up". In general, under the assumption that all gates have delays Δ_i in the interval $[\delta-\epsilon, \delta+\epsilon]$, the following condition makes sure that the k^{th} race unit does not get mixed up with the $(k-1)^{st}$:

$$k(\delta - \epsilon) > (k - 1)(\delta + \epsilon).$$

The condition states that the time to complete k changes in the fastest gates (each with delay $\delta - \epsilon$) should be greater then the time required for $k - 1$ slowest gates (each with delay $\delta + \epsilon$). What can happen, if the above condition is not satisfied, is that the model may omit certain races that potentially could exist. In summary, a sufficient condition for the stepwise AED model to be accurate for at least k steps, is that

$$\frac{\epsilon}{\delta} < \frac{1}{2k - 1}.$$

For example, if the uncertainty of the delays in the gates is 10%, the stepwise AED model is accurate for at least 5 steps. However, the reader should note that the above condition is only a sufficient condition; in many cases the AED model is accurate for more than k steps.

5. The TAED Model

In this section we describe a ternary simulation method which is related to the stepwise AED model. For more details about ternary simulation in general, the reader should refer to [BY79]. Let $T = \{0,1,½\}$. The values 0 and 1 represent the usual logic levels and ½ represents an unknown value. We will use the following convention. Variables like y_i, etc. which take values from $B = \{0, 1\}$ will have corresponding variables $\mathbf{y}_i$, etc. taking values from T. The *partial order* $\leq$ on T is defined by

$$\mathbf{t} \leq \mathbf{t} \text{ for all } \mathbf{t} \in T,$$
$$0 \leq ½ \text{ and } 1 \leq ½.$$

We extend the partial order $\leq$ to T^m in the usual way:

$$\mathbf{t} \leq \mathbf{r} \textit{ iff } \mathbf{t}_i \leq \mathbf{r}_i \text{ for all } i = 1, \cdots, m.$$

The statement $\mathbf{t} \leq \mathbf{r}$ means that whenever r_i is binary then t_i has the same binary value as r_i, but $\mathbf{r}$ may contain more unknown components (i.e. components with value ½). Thus $\mathbf{r}$ has more "uncertainty" than $\mathbf{t}$.

For any vector of boolean functions $f : B^r \rightarrow B^p$ its *ternary extension* $\mathbf{f}: T^r \rightarrow T^p$ is defined by

$$\mathbf{f}(\mathbf{t}) = l.u.b. \{f(a) : a \in B^r \text{ and } a \leq \mathbf{t}\}.$$

It follows that, for $t \in B^r$, $\mathbf{f}(t) = f(t)$, i.e. on binary vectors the ternary extension agrees with the original function. The ternary extension obeys the following monotonicity property [BY79]:

$$\mathbf{t} \leq \mathbf{r} \text{ implies } \mathbf{f}(\mathbf{t}) \leq \mathbf{f}(\mathbf{r}).$$

The underlying idea behind the TAED method, formally defined below, is to find all the unstable gates in a state $\mathbf{y}$, and then determine which of these unstable gates that *must* change in this race unit. Consider network N_4 of Fig. 11(a) started in the stable state $x = 0$, $y = 1\,1$ and with the new input $\hat{x} = 1$. The first step of the TAED algorithm is to calculate the *l.u.b.* of the present state, and the excitation of the gate network. In this intermediate state, called $\mathbf{t}$ in the algorithm, a gate will have the value ½ if it is unstable. In network N_4 both gate 1 and gate 2 are unstable and hence $\mathbf{t} = ½\,½$. In order to determine which of the unstable gates must change in the present race unit, the excitations of the unstable gates (i.e. the gates for which $\mathbf{t}_j = ½$) are re-evaluated. However, this re-evaluation is done assuming $\mathbf{t}$ is the total state of the network. If the excitation of an unstable gate j is still binary, then, independently of changes in the other unstable gates, this gate has to change before the present race unit can finish. On the other hand, if the "new" excitation of an unstable gate j is ½, then that instability "depends" on some gates that are also unstable. Hence it is possible to change these other unstable gates first and thereby remove the instability of gate j. In network N_4 we get $\mathbf{Y}_1(1, ½½) = 1' = 0$ and $\mathbf{Y}_2(1, ½½) = (1½)' = ½$, so the new state we reach is 0 ½. It is easy to see that gate 1 must change in any race unit, since the excitation of the gate depends only on the new input value. On the other hand, the instability of gate 2 depends on gate 1 which is unstable; if gate 1 changes first, gate 2 becomes stable. Hence in the stepwise AED model we can reach in one race unit the states 0 0 or 0 1. Note that in the next race unit, gate 2 will again be "unstable" (= ½) but when re-evaluated, it will be "forced" to the binary value 0. This is also consistent with the stepwise AED model.

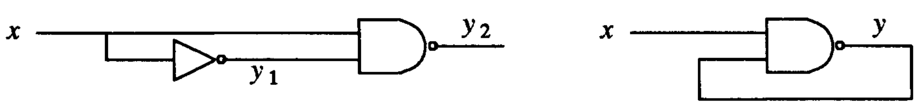

Figure 11 (a) Network N_4; (b) network N_5.

In the description above we simplified the re-evaluation of the excitation of the unstable gates slightly. As the following example shows, it is not correct to use the values of $\mathbf{t}$ for all gates. Study network N_5 of Fig. 11(b) started in the stable state $x = 0$, $y = 1$ and with the new input $\tilde{x} = 1$. We immediately get $\mathbf{t} = ½$, but if we use that value for the re-evaluation, the TAED model will predict that the state after one race unit will be ½. However, the stepwise AED model only predicts the state $y = 0$. The reason for this discrepancy is that, when there is a self-loop, the ½ that the gate depends on, is coming from itself. Hence the only way to remove the instability of the gate is to change its state. The solution to this problem is simply to use the "old" gate state for gate j when re-evaluating gate j. We will use the notation $\mathbf{t}^{(j)}$ to denote the vector obtained from $\mathbf{t}$ by replacing the j^{th} component by $\mathbf{y}_j$. In network N_5 we get $\mathbf{Y}(\tilde{x}, \mathbf{t}^{(1)}) = \mathbf{Y}(1, 1) = (11)' = 0$, and hence the TAED model will correspond to the stepwise AED model. We now formally define the TAED method.

We define the function **next** as follows. Let $\tilde{x}$ be the new binary input, and let $\mathbf{y}$ be any ternary gate state. We calculate the successor state of $\mathbf{y}$ as shown below.

```
function next(y ∈ T^n) ∈ T^n
{
    for j = 1 to n do              -- First calculate the intermediate state t
    {
        t_j = l.u.b (y_j , Y_j (x̃ , y))
    }
    for j = 1 to n do              -- Re-evaluate the excitation
    {
        if t_j = ½ then
            ỹ_j = Y_j (x̃ , t^(j))     -- t^(j) = t_1 · · · t_{j-1} y_j t_{j+1} · · · t_n
        else
            ỹ_j = y_j
    }
    return(ỹ)
}
```

The TAED algorithm consists of repeatedly applying **next**. Note that this will either lead to a stable ternary state, or an oscillation. In this respect the TAED method is similar to the unit-delay method used by most commercial simulators.

To illustrate the algorithm, consider network N_1 of Fig. 1 started in the stable state $x = 0$, $y^0 = 1\,0\,0$ and with the new input $\tilde{x} = 1$. The first time we call **next** we get the following intermediate values:

$\mathbf{t}_1 = l.u.b.\ \{y_1^0, \mathbf{Y}_1(\tilde{x}, y^0)\} = l.u.b.\ \{1, \tilde{x}'\} = l.u.b.\ \{1, 0\} = ½$,

$\mathbf{t}_2 = l.u.b.\ \{y_2^0, \mathbf{Y}_2(\tilde{x}, y^0)\} = l.u.b.\ \{0, (\tilde{x} y_1^0)\} = l.u.b.\ \{0, 1\} = ½$, and

$\mathbf{t}_3 = l.u.b.\ \{y_3^0, \mathbf{Y}_3(\tilde{x}, y^0)\} = l.u.b.\ \{0, (y_2^0 + y_3^0)\} = l.u.b.\ \{0, 0\} = 0$.

Hence, $\mathbf{t} = ½\ ½\ 0$. In the second step of the algorithm, the gates with ½ in the intermediate state are re-evaluated using the intermediate state, except for the value of the gate itself for which we use the "old" value. For network N_1 we re-evaluate gates 1 and 2, and we get the following values:

$$\tilde{\mathbf{y}}_1 = \mathbf{Y}_1(\tilde{x}, \mathbf{t}^{(1)}) = \tilde{x}' = 0, \text{ and}$$
$$\tilde{\mathbf{y}}_2 = \mathbf{Y}_2(\tilde{x}, \mathbf{t}^{(2)}) = (\tilde{x}\ \mathbf{t}_1) = (1\ ½) = ½.$$

Hence, we obtain the new state $\tilde{\mathbf{y}} = 0\ ½\ 0$. In Fig. 12(a) we give the complete TAED analysis of network N_1.

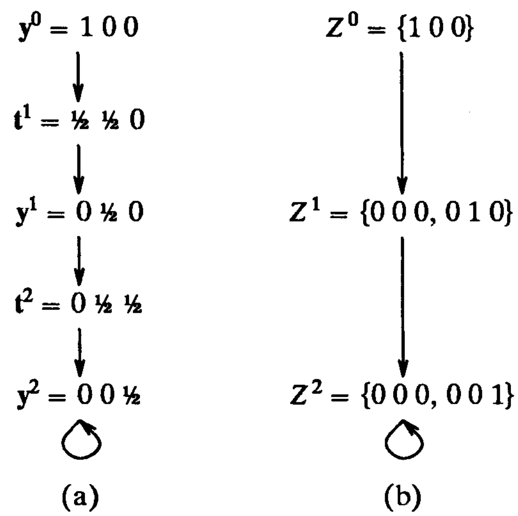

Figure 12. Analysis of N_1: (a) TAED model;
(b) states reachable in stepwise AED model.

It is interesting to note that the *l.u.b.* of the states reachable after i race units in the stepwise AED model, corresponds exactly to the outcome of the TAED method after i steps; see Fig. 12(a) and (b). In the next section we explore the relationship between the two models.

6. A Partial Characterization of the TAED Method

In this section we show a partial correspondence between the binary stepwise AED model and the results of the TAED algorithm described above. The main result is:

Theorem 2: Let x, y^0 be a stable total state, and let $\tilde{x}$ denote a new input vector. Let $\mathbf{y}^i$ denote the results of the TAED method after $i \geq 0$ steps starting in $\mathbf{y}^0 = y^0$. Then:

$$\mathbf{y}^i \geq l.u.b.(Z^i),$$

where Z^i is the set of states reachable in i steps in the stepwise AED model.

Before proving the theorem we will establish Lemma 1 below. The following observation is useful in proving the lemma. If $\mathbf{y}$ is the input to **next**, then $\mathbf{t}^{(j)} \geq \mathbf{y}$. Hence, by the monotonicity property of the ternary extension,

$$\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) \geq \mathbf{Y}_j(\tilde{x}, \mathbf{y}). \tag{1}$$

Lemma 1: Let $\mathbf{y} \in T^n$ and let $\tilde{\mathbf{y}}$ be **next**(y). Then

$$w \leq \mathbf{y} \text{ and } w\,\sigma z \text{ implies } z \leq \tilde{\mathbf{y}}.$$

Proof: If $\tilde{\mathbf{y}}_j = ½$, then for any such z we will trivially have $\tilde{\mathbf{y}}_j \geq z_j$. Hence, study the cases when $\tilde{\mathbf{y}}_j \in B$. Note that, by the definition of **t**, it follows that $\mathbf{y}_j = ½$ implies $\mathbf{t}_j = ½$. Therefore there are only four cases to consider.

	1	2	3	4
$\mathbf{y}_j =$	b	b	b	½
$\mathbf{t}_j =$	b	½	½	½
$\tilde{\mathbf{y}}_j =$	b	b	b'	b

Here b stands for some binary value (0 or 1), and b' denotes the complement of b.

Case 1 $\mathbf{y}_j = b$, $\mathbf{t}_j = b$, and $\tilde{\mathbf{y}}_j = b$.
By the definition of $\mathbf{t}_j$, $\mathbf{t}_j = b$ implies that $\mathbf{Y}_j(\tilde{x}, \mathbf{y}) = b$. Furthermore, by the definition of ternary extension, we must also have $Y_j(\tilde{x}, w) = b$ for every $w \leq \mathbf{y}$. Since $w_j \leq \mathbf{y}_j = b$, gate j is stable in any such state w. Consequently, gate j remains unchanged in any race sequence beginning in state w. Therefore, if $w\,\sigma z$, we must have $z_j = b$. By assumption, $\tilde{\mathbf{y}}_j = b$ and $\tilde{\mathbf{y}}_j \geq z_j$ holds.

Case 2 $\mathbf{y}_j = b$, $\mathbf{t}_j = ½$, and $\tilde{\mathbf{y}}_j = b$.
If $\mathbf{t}_j = ½$ then $\mathbf{Y}_j(\tilde{x}, \mathbf{y})$ is either ½ or b'. By (1) $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) \geq \mathbf{Y}_j(\tilde{x}, \mathbf{y})$. Thus $\tilde{\mathbf{y}}_j$, as computed by **next**, must have the value ½ or b'. But we have assumed $\tilde{\mathbf{y}}_j = b$. Hence this case is impossible.

Case 3 $\mathbf{y}_j = b$, $\mathbf{t}_j = ½$, and $\tilde{\mathbf{y}}_j = b'$.
We first prove that in any state $w \leq \mathbf{y}$, gate j is unstable. Since $\mathbf{t}_j = ½$ we know that $\tilde{\mathbf{y}}_j = \mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)})$. Also, by assumption, $\tilde{\mathbf{y}}_j = b'$. Altogether we have $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) = b'$. By (1) we know that $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) \geq \mathbf{Y}_j(\tilde{x}, \mathbf{y})$, so $b' = \mathbf{Y}_j(\tilde{x}, \mathbf{y})$. Furthermore, by the definition of ternary extension, it follows that $\mathbf{Y}_j(\tilde{x}, \mathbf{y}) \geq Y_j(\tilde{x}, w)$ when $w \leq \mathbf{y}$. Hence $b' = Y_j(\tilde{x}, w)$ for $w \leq \mathbf{y}$. Since $w \leq \mathbf{y}$, and $\mathbf{y}_j = b$, we must have $w_j = b$. Together, this shows that gate j is unstable in any state $w \leq \mathbf{y}$.

Secondly, we show that for any race sequence starting in state w, gate j must change, and therefore for any z such that $w\,\sigma z$ we will have $z_j = b'$. We prove this by contradiction. Assume there exists a race sequence $<z^0, S^0>, <z^1, S^1>, \cdots, <z^k, S^k>$

$(k \geq 0)$ with $z^0 = w$, $S^0 = U(w)$, and $S^k = \emptyset$, such that $z_j^k = b$. Since, according to the first part above, gate j is unstable in any state $w \leq \mathbf{y}$, we can conclude that $k \geq 1$. Furthermore, since $w \leq \mathbf{y}$, and, by assumption, $\mathbf{y}_j = b$, we must have $z_j^0 = b$. However, by the definition of a race sequence it follows that a gate can change at most once during a race sequence; since $z_j^0 = b$, and z_j^k is assumed to be b, it follows that $z_j^p = b$ for $p = 0, \cdots, k$. Since $\mathbf{t} = l.u.b.\,(\mathbf{y}, \mathbf{Y}(\tilde{x}, \mathbf{y}))$, it follows by the definition of ternary extension that $\mathbf{t} \geq l.u.b.\,(w, Y(\tilde{x}, w))$ for $w \leq \mathbf{y}$. Furthermore, by the definition of a race unit, it follows that for $j = 1, \cdots, n$, and $p = 0, \cdots, k$, we have that z_j^i is either equal to w_j or $Y_j(\tilde{x}, w)$. We can therefore conclude that $z^p \leq l.u.b.\,(w, Y(\tilde{x}, w))$, and hence $\mathbf{t} \geq z^p$ for $p = 0, \cdots, k$. This, together with the fact that $z_j^p = b = \mathbf{y}_j$ for $p = 0, \cdots, k$ shows that $\mathbf{t}^{(j)} \geq z^p$ for $p = 0, \cdots, k$. By the monotonicity of ternary extension $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) \geq \mathbf{Y}_j(\tilde{x}, z^p)$ for $p = 0, \cdots, k$. Furthermore, since $\mathbf{t}_j = ½$ we have $\tilde{\mathbf{y}}_j = \mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)})$, and, since $\tilde{\mathbf{y}}_j = b'$, by assumption, it follows that $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) = b'$. Together the above two facts imply that $Y_j(\tilde{x}, z^p) = b'$ for $p = 0, \cdots, k$. However, this implies that j is in $U(z^p)$ for $p = 0, \cdots, k$ contradicting the assumption that $S^k = \emptyset$. Therefore the assumption must have been false, and we can conclude that all states z, such that $w \sigma z$, must have $z_j = b'$. Hence, $\tilde{\mathbf{y}}_j \geq z_j$ holds.

Case 4 $\mathbf{y}_j = ½$, $\mathbf{t}_j = ½$, and $\tilde{\mathbf{y}}_j = b$.
Since $\mathbf{t}_j = ½$ it follows that $\tilde{\mathbf{y}}_j = \mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)})$, and since we assumed $\tilde{\mathbf{y}}_j = b$, it follows that $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) = b$. By (1) we know that $\mathbf{Y}_j(\tilde{x}, \mathbf{t}^{(j)}) \geq \mathbf{Y}_j(\tilde{x}, \mathbf{y})$, so we can conclude that $\mathbf{Y}_j(\tilde{x}, \mathbf{y}) = b$. By the definition of ternary extension it follows that $Y_j(\tilde{x}, w) = b$ for any $w \leq \mathbf{y}$. Study any state $w \in B^n$, $w \leq \mathbf{y}$. We have two cases:

(i) $w_j = b$.
Since $w_j = b$ and $Y_j(\tilde{x}, w) = b$, we have that gate j is stable, and, as in Case 1 above, we can conclude that for any state z such that $w \sigma z$ we have $z_j = b$. Hence, $\tilde{\mathbf{y}}_j \geq z_j$ holds.

(ii) $w_j = b'$.
Since $w_j = b'$ and $Y_j(\tilde{x}, w) = b$ we have that gate j is unstable. Using similar arguments as in the second part of Case 3 above, we can conclude that for any race sequence

from the state w, gate j must change and thus we have that if $w \sigma z$, then $z_j = b$. Hence, $\tilde{\mathbf{y}}_j \geq z_j$ holds.

We have shown that $\tilde{\mathbf{y}}_j \geq z_j$ for all possible cases, and hence that $\tilde{\mathbf{y}} \geq z$ for any state z, such that $w \sigma z$, where $w \leq \mathbf{y}$. □

We are now in position to prove Theorem 2.

Proof of Theorem 2: We want to show that if $\mathbf{y}_j^i \in B$, then $l.u.b. \{z_j : y^0 \sigma^i z\} = \mathbf{y}_j^i$. We will prove this by induction on i.

Basis: $i = 0$. Trivially true.

Induction Hypothesis: Assume that for all $k \leq i$, we have that $\mathbf{y}^k \geq l.u.b. \{z : y^0 \sigma^k z\}$.

Induction Step: We need to show that for any z such that $y^0 \sigma^{i+1} z$, we have $\mathbf{y}^{i+1} \geq z$. But $y^0 \sigma^{i+1} z$ implies there exists w such that $y^0 \sigma^i w$ and $w \sigma z$. By the induction hypothesis $w \leq \mathbf{y}^i$, and Lemma 1 applies. Thus $\mathbf{y}^{i+1} \geq z$. □

The following example shows that the inequality of the theorem can not be replaced by equality, i.e. that the TAED model is sometimes overly pessimistic. Study network N_6 of Fig. 13, started in the stable state $x = 0$, $y = y_1 \cdots y_6 = 1\,1\,1\,0\,0\,0$ and with the new input $\tilde{x} = 1$. In Fig. 14 we show the binary stepwise AED analysis of the race and in Fig. 15 we show the TAED analysis. Note that the ½ for gate 5 in state $\mathbf{y}^2$ is incorrect. The reason for this discrepancy between the ternary simulation and the binary race analysis is that in the state $\mathbf{y}^1$ not all binary states $\leq \mathbf{y}^1$ are reachable, and in particular, the state $y = 0\,1\,1\,1\,0\,0$ is not reachable from the initial state. It is this state that causes gate 5 to be unstable in the ternary simulation, and eventually leads to the ½ in $\mathbf{y}_5^2$. In general, the discrepancy occurs because of the loss of information in the ternary simulation where we only use the "average" of the states reachable. It appears that this pessimism occurs only in pathological examples, and that for most practical circuits, the ternary and the binary AED analysis correspond exactly. However, the problem of characterizing the class of networks for which the two models agree remains open.

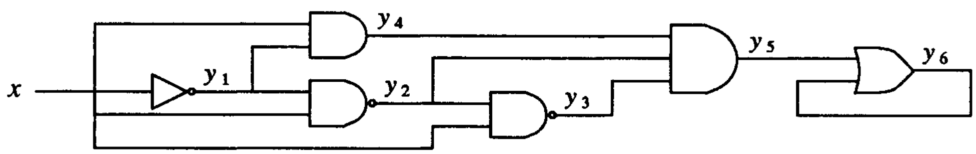

Figure 13. Network N_6.

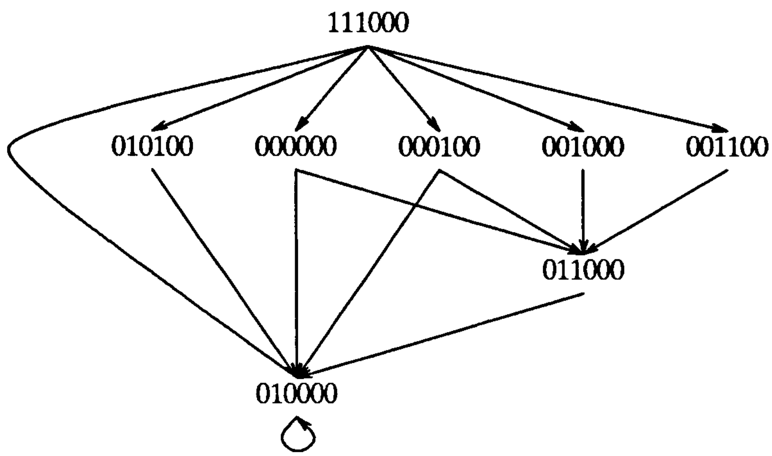

Figure 14. Stepwise AED analysis of N_6.

(a)	(b)
$\mathbf{y}^0 = 1\,1\,1\,0\,0\,0$	$l.u.b.\,(Z^0) = 1\,1\,1\,0\,0\,0$
$\mathbf{t}^1 = \tfrac12\tfrac12\tfrac12\tfrac12\,0\,0$	
$\mathbf{y}^1 = 0\tfrac12\tfrac12\tfrac12\,0\,0$	$l.u.b.\,(Z^1) = 0\tfrac12\tfrac12\tfrac12\,0\,0$
$\mathbf{t}^2 = 0\tfrac12\tfrac12\tfrac12\tfrac12\,0$	
$\mathbf{y}^2 = 0\,1\tfrac12\,0\tfrac12\,0$	$l.u.b.\,(Z^2) = 0\,1\tfrac12\,0\,0\,0$
$\mathbf{t}^3 = 0\,1\tfrac12\,0\tfrac12\tfrac12$	
$\mathbf{y}^3 = 0\,1\,0\,0\,0\tfrac12$	$l.u.b.\,(Z^3) = 0\,1\,0\,0\,0\,0$

Figure 15. (a) TAED analysis of N_6;
(b) *l.u.b.* of the stepwise AED analysis.

7. Summary

In this paper we have presented a new ternary simulation algorithm that can easily replace the unit-delay algorithm in simulators. The algorithm is very closely related to the binary almost-equal-delay model, and hence is capable of detecting critical races under the assumption that all delays are approximately, but not exactly, equal. Computationally the ternary algorithm is of the same order of complexity as the unit-delay method (in the worst case it involves twice as many function evaluations). A major disadvantage with both this ternary algorithm and with the unit-delay method, is that the algorithms are not guaranteed to halt. One can easily test whether a ternary stable state is reached by comparing $\mathbf{y}$ and $\tilde{\mathbf{y}}$. If $\tilde{\mathbf{y}} = \mathbf{y}$ and $\mathbf{y}$ is binary, we know the circuit reliably reaches a unique stable state. Otherwise, a critical race or an oscillation is likely, and further analysis is required. Since there are at most 3^n reachable ternary states, it is decidable whether the analysis predicts a unique binary stable state. However, for large networks this decision may involve an excessive amount of computation. For such reasons commercial simulators simply terminate the computation when the number of steps exceeds some arbitrarily chosen relatively small limit. The same approach can be used here. One can argue that a practical circuit that does not stabilize in (say) 10 race units is not very well designed. Consequently the method can be made practical and should produce useful results.

References

[B83] R.E. Bryant, "Race Detection in MOS Circuits by Ternary Simulation", In F. Anceau and E.J. Aas (eds.), *VLSI '83*, pp. 85-95, Elsevier Science Publishers B. V. (North-Holland), 1983.

[B84] R.E. Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems", *IEEE Transactions on Computers*, Vol. C-33, No. 2, pp. 160-177, Feb. 1984.

[BS85] J. A. Brzozowski and C.-J. Seger, "A Characterization of Ternary Simulation of Gate Networks", *Tech. Rep. CS-85-37*, University of Waterloo, Oct. 1985.

[BS86] J.A. Brzozowski and C-J. Seger, "Correspondence between Ternary Simulation and Binary Race Analysis in Gate Networks", to appear in Proceedings of ICALP '86.

[BY75] J.A. Brzozowski and M. Yoeli, "Models for Analysis of Races in Sequential Networks", In A. Blikle (ed.), *Lecture Notes in Computer Science*, pp. 26-32, Springer Verlag, 1975.

[BY76] J.A. Brzozowski and M. Yoeli, *Digital Networks*, Prentice-Hall, Englewood Cliffs, N. J., 1976.

[BY79] J.A. Brzozowski and M. Yoeli, "On a Ternary Model of Gate Networks", *IEEE Transactions on Computers*, Vol. C-28, No. 3, pp. 178-183, Mar. 1979.

[E65] E.B. Eichelberger, "Hazard Detection in Combinational and Sequential Switching Circuits", *IBM J. Res. Dev.*, Vol. 9, pp. 90-99, Mar. 1965.

[JMV69] J.S. Jephson, R.P. McQuarrie, and R.E. Vogelsberg, "A Three-Level Design Verification System", *IBM Systems Journal*, Vol. 8, No. 3, pp. 178-188, Mar. 1969.

[S] SimuTec, "SILOS: Logic Simulator User's Manual".