ON A TERNARY MODEL OF GATE NETWORKS

by

J.A. Brzozowski
Dept. of Comp. Science
University of Waterloo
Waterloo, Ont., Canada

M. Yoeli
Dept. of Comp. Science
Technion
Haifa, Israel

Research Report CS-76-40
September 1976

This work was done when the second author was visiting the University of Waterloo. This research was supported by the National Research Council of Canada under Grant No. A-1617.

ON A TERNARY MODEL OF GATE NETWORKS

by

J.A. Brzozowski and M. Yoeli

Abstract

In this paper we formalize a ternary model which is being used to study the behavior of binary sequential gate networks. We first introduce a general binary model which is capable of a detailed description of network behavior, but involves a number of steps that grows exponentially in the number of gates. The complexity of the ternary model is linear in the number of gates; however, only partial information is obtained in general. A mathematical theory is developed, making precise these two models and the comparison between them. A number of examples illustrate these results. This work generalizes previously reported research.

1. Introduction

Eichelberger [EIC] has shown how ternary logic can be used in order to detect hazards in combinational gate networks, as well as races and oscillations in sequential networks. The mathematical theory of hazard detection was further developed in [BR-YO]. The methods described in [EIC] have been used in [JE-MC-VO], [PU-RO], and elsewhere for implementing a ternary simulator of digital systems.

However, a variety of "pitfalls" of such ternary simulators have been encountered by their users. An informal explanation of some of these pitfalls is given in [BRE]. In [YO-BR] we extended the binary GSW (General Single Winner) race model of [BR-YO] to ternary gate networks. This framework enabled us to establish a mathematically precise result indicating the limitations of Eichelberger's race detection techniques.

This paper provides further insight into the features of ternary simulators based on Eichelberger's method. First, we introduce a more general binary race model, namely the GMW (General Multiple Winner) model. We then generalize the result obtained in [YO-BR], and provide a detailed proof. Finally we give a detailed comparison between the results obtained by the binary and ternary models.

2. The GMW Model

We are considering networks constructed from gates such as INVERTERS and multi-input AND, OR, NAND and NOR gates shown in Fig. 1. However, the mathematical theory need not be restricted to these gate types; in fact, any n-input, one-output combinational network can be considered as a generalized gate. With each such gate we associate a Boolean function $f:B^n \rightarrow B$, where $B \triangleq \{0,1\}$. (The symbol $\triangleq$ means 'is by definition'.)

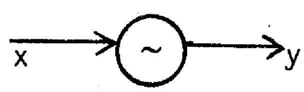
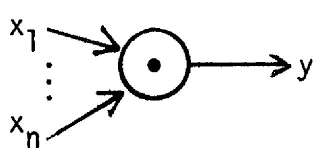
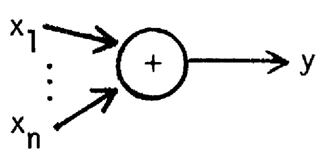
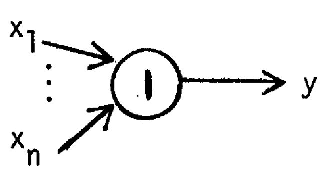
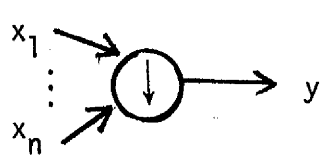
GATE TYPE	SYMBOL	BOOLEAN FUNCTION
INVERTER		$f(x) = x'$
AND		$f(x_1,\ldots,x_n) = x_1 \cdot \ldots \cdot x_n$
OR		$f(x_1,\ldots,x_n) = x_1+\ldots+x_n$
NAND		$f(x_1,\ldots,x_n) = (x_1\cdot\ldots\cdot x_n)'$
NOR		$f(x_1,\ldots,x_n) = (x_1+\ldots+x_n)'$

Fig. 1 Gate symbols

Figure 2 shows the logic diagram of a gate network (a NOR latch with inputs tied together, to keep the example simple). When $x = 1$, the state $y_1 = y_2 = 0$ is stable. Our problem is to predict the behavior of a network when it is started in some such stable state y and the input changes (to $x = 0$, in our example). We assume that the new input value will not change until the network reaches a new steady-state condition. Under these assumptions, the complete analysis of any gate network can be carried out by repeating the analysis for all such states y for each input x. For a more detailed discussion of the analysis problem see [BR-YO].

More formally, let the network have n binary inputs $x_1,\ldots,x_n$ and s gates $G_1,\ldots,G_s$ whose outputs are $y_1,\ldots,y_s$. Let $x \triangleq x_1,\ldots,x_n$, $y \triangleq y_1,\ldots,y_s$ and $Y \triangleq Y_1,\ldots,Y_s$, where $Y_i \triangleq f_i(x,y)$, f_i being the Boolean function associated with gate G_i. We call Y the <u>excitation</u> of the network. The entire logical structure of the network can be concisely described by a <u>network function</u> $F:B^n \times B^s \to B^s$. Let $[F(x,y)]_i$, the ith coordinate of F, be the Boolean function f_i associated with gate G_i.

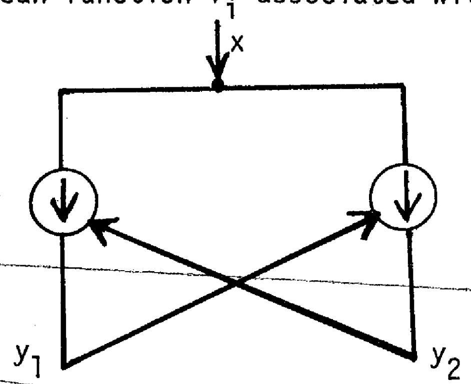


Fig. 2 Network N_1

In our example, we have the excitation equations:

$$Y_1 = (x+y_2)' \qquad Y_2 = (x+y_1)'$$

The network function F is given in Fig. 3.

y \ x	0	1
00	1 1	0 0
01	0 1	0 0
10	1 0	0 0
11	0 0	0 0

Y(x,y)

Fig. 3 Network function for N_1.

A state y of a network is primary iff there exists an input $\bar{x} \in B^n$ such that $F(\bar{x},y) = y$; i.e. $(\bar{x},y)$ is a stable total state. We assume that we are interested only in analyzing the behavior under any input x when the network is started in a primary state y . This analysis is carried out with the aid of a binary relation R_x on the set B^s of gate states. We now define this relation for the GMW (General Multiple Winner) model of races.

Let $y,\bar{y} \in B^s$ be gate states. We define the interval $[y,\bar{y}]$ between y and $\bar{y}$ as follows:

$$[y,\bar{y}] = \{z \in B^s \mid z_i = y_i \text{ or } z_i = \bar{y}_i \text{ for all } i \in \{1,\ldots,s\}\}.$$

For example, $[00,01] = \{00,01\}$, and $[1100,1111] = \{1100,1101,1110,1111\}$. We will be using the interval between a state y and its excitation $Y_x = F(x,y)$. This interval $[y,Y_x]$ consists of all the states that can be immediate successors of state y under input x, on the assumption that any subset of unstable gates can change simultaneously (multiple winners of a race are allowed). We now define the GMW relation R_x. For $y,\bar{y} \in B^s$,

$$y \; R_x \; y \quad \text{iff } y = F(x,y)$$

and for $\bar{y} \neq y$,

$$y \; R_x \; \bar{y} \quad \text{iff } \bar{y} \in [y,F(x,y)].$$

The relation R_x is conveniently represented by its relation diagram, where nodes correspond to internal states and an arrow from node y to node $\bar{y}$ indicates that y is related to $\bar{y}$, i.e. $y \; R_x \; \bar{y}$. The relation diagram of R_0 for the network of Fig. 2 is shown in Fig. 4. This is obtained from the $x = 0$ column of Fig. 3 by applying the definition of interval. In Fig. 4 we show that gate G_i is unstable, i.e. $Y_i \neq y_i$, by underlining y_i. If the network starts in the primary state $y = 00$ with $x = 0$, both variables are unstable. This is a race. It is seen that it is possible to reach two stable states, 01 and 10, depending on the relative magnitudes of the gate delays. Such a possibility is called a critical race. If the delays are equal it is possible for the network to oscillate between the states 00 and 11. However, it is unrealistic to assume that this oscillation will persist indefinitely, since it is unlikely that the gate delays remain perfectly matched.

This indicates that the GMW model may be somewhat pessimistic, since it predicts an unlikely oscillation.

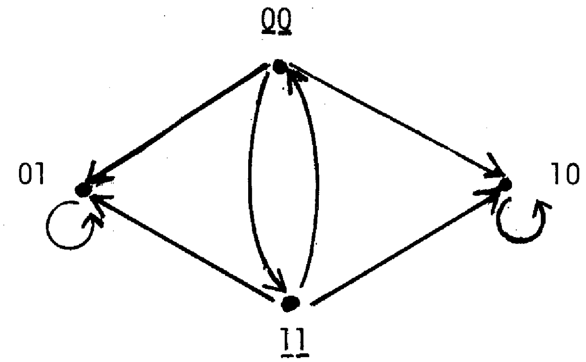


Fig. 4 GMW relation R_0 for N_1.

A second example of an oscillation that is unlikely to persist is provided by the network N_2 of Fig. 5 (a). Note that F(0,011) = 011. Hence 011 is a primary state. The GMW analysis of this state for x = 1 is shown in Fig. 5(b). Note that we only show that part $R_x(y)$ of the relation R_x in which all nodes are reachable from the primary state y; the remaining states are of no interest. Figure 5(b) shows the cycle {011,010} indicating an oscillation. Observe, however, that the first gate is unstable in both states of the cycle and its output will eventually change from 0 to 1. The network will then stop oscillating and reach the unique stable state 101.

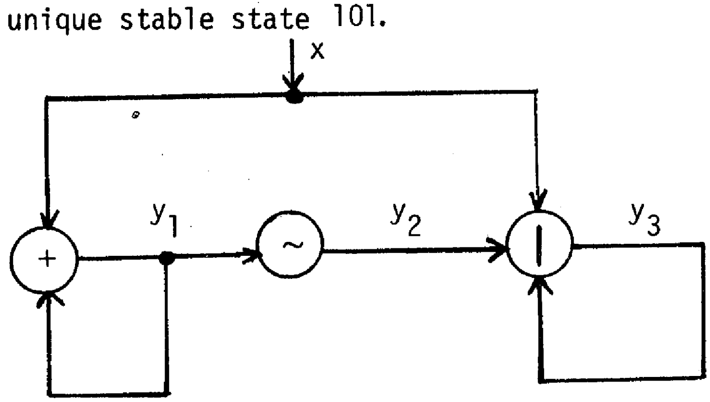


Fig. 5(a) Network N_2

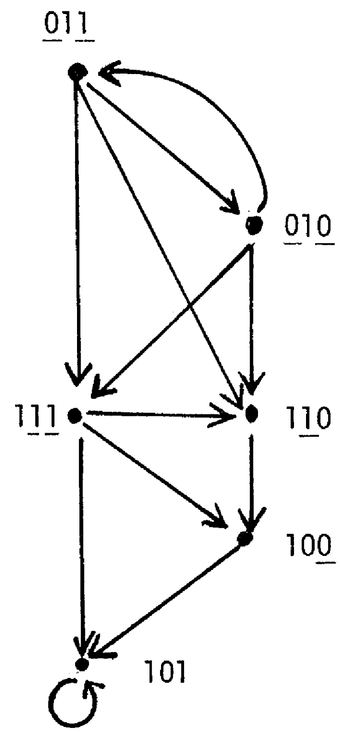


Fig. 5(b) Relation R_1 (011) for N_2.

A third type of oscillation is illustrated in Fig. 6(b) for the network N_3 of Fig. 6(a). Here the oscillation will continue indefinitely.

We now return to the more formal approach. The relation diagram is always finite since the set B^s is finite. Consequently, if we start with y and follow any directed path in the diagram we must reach a cycle. We say that a cycle C is sustained iff for all y in C, $F(x,y)$ differs from y in no more than one coordinate. (See Fig. 6.). A cycle is transient iff there exists i, $1 \le i \le s$ such that gate G_i is unstable and y_i has the same value for all the states in the cycle. A cycle C is match-dependent iff there exist states y and z in C such that z is the successor of y in C and y and z differ in at least two coordinates. (See Fig. 4.)

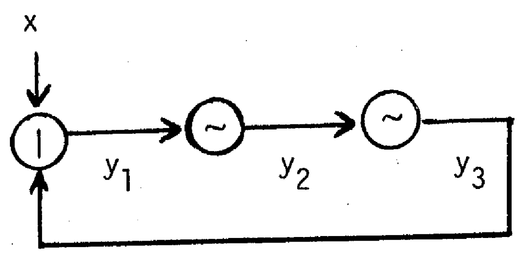


Fig. 6(a) Network N_3

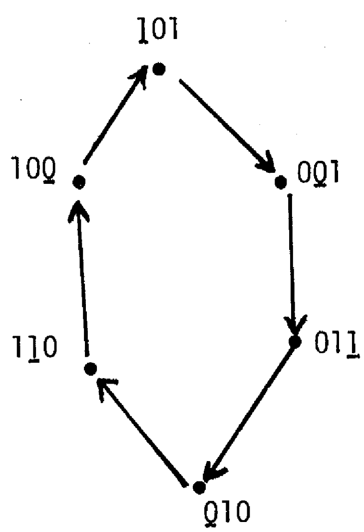


Fig. 6(b) Relation R_1 (101) for N_3.

Given any set S and a binary relation R on S, define for any $y \in S$:

$$\underline{\text{cycl}}\ (R,y) \triangleq \{z \in S \mid y\ R^* z \text{ and } z\ R^+ z\},$$

where R^+ and R^* are the transitive and the transitive-and-reflexive closures of R respectively. Thus cycl (R_x,y) for $x \in B^n$, $y \in B^s$ is the set of all cyclic states reachable from y, where a state is cyclic iff it appears in some cycle of the relation diagram of R_x. Also let trans $(R_x,y) \triangleq \{z \in \text{cycl}\ (R_x,y) \mid z$ appears in only transient cycles of $R_x(y)\}$ and out $(R_x,y) \triangleq$ cycl (R_x,y) - trans (R_x,y).

The set out—the "outcome" of state y under input x — represents the set of all states that the network can be in, under non-transient situations. We can distinguish the following cases:

1) A unique stable state, i.e. cycl$(R_x,y) = \{z\}$ for some $z \in B^s$. This is usually the desirable case, when a transition takes place from y to z under input x and this transition is independent of the relative delays of the gates.

2) cycl(R_x,y) consists of a single cycle of length more than one. Note that this cycle is necessarily sustained. This represents an oscillation that will continue until the input changes again.

3) out(R_x,y) contains two or more cycles. Here the behavior depends on which gate wins the race, and the race is critical.

We interpret transient cycles as follows: it is possible for the network to "go around" a transient cycle several times.

However, eventually, the network will leave the cycle. We point out that match-dependent cycles are somewhat similar to transient cycles because it is improbable that the network will remain in a match-dependent cycle indefinitely. Disregarding match-dependent cycles will therefore result in a more optimistic model. One such model is the GSW (General Single Winner) model [BR-YO], in which $y\ R_x\ \bar{y}$ implies that y and $\bar{y}$ differ in at most one coordinate.

As we shall see, the ternary simulation results will provide some information about out (R_x, y).

3. Ternary Model

In this section we introduce a ternary model of binary gate networks and describe the corresponding simulation procedure.

Let $T \triangleq \{0, \frac{1}{2}, 1\}$. Intuitively, $\frac{1}{2}$ represents the fact that the value of a binary signal is unknown. We introduce a partial order relation $\sqsubseteq$ on T given by the relation diagram of Fig. 7.

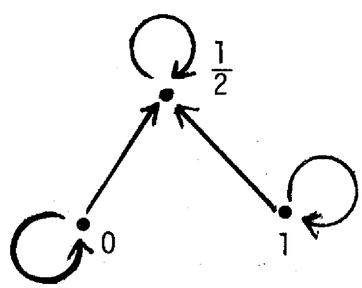


Fig. 7 Relation diagram of $\sqsubseteq$.

Thus, we have $t \sqsubseteq t$, for every $t \in T$, as well as $0 \sqsubseteq \frac{1}{2}$ and $1 \sqsubseteq \frac{1}{2}$. We now extend the partial order $\sqsubseteq$ to T^m, $m > 1$, in the usual way; namely, let $t, r \in T^m$, then

$$t \sqsubseteq r \text{ iff } t_i \sqsubseteq r_i \text{ for every } i \in \{1,\dots,m\}.$$

For example, $(0, \frac{1}{2}, 1, 1) \sqsubseteq (\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2})$.

Next, we set $\mu\{0\} = 0$, $\mu\{1\} = 1$ and $\mu\{0,1\} = \frac{1}{2}$, and for any nonempty subset A of B^s, $s > 1$, we set

$$\mu A \triangleq (\mu A_1, \dots, \mu A_s)$$

where $A_i \triangleq \{a_i | (a_1, \dots, a_s) \in A\}$, $1 \le i \le s$.

For example,

$$\mu\{(0,0,1),(0,1,0)\} = \mu\{(0,0,0),(0,1,1)\} = (0,\tfrac{1}{2},\tfrac{1}{2}).$$

Clearly, if $a \in A \subseteq B^s$, then $a \sqsubseteq \mu A$.

With any function $F: B^m \to B^s$ ($m \geq 1$, $s \geq 1$) we associate its <u>ternary</u> <u>extension</u> $\underline{F}: T^m \to T^s$, where for every $t \in T^m$

$\underline{F}(t) = \mu\{F(x) \mid x \in B^m \text{ and } x \sqsubseteq t\}$. Note that for $t \in B^s$, $F(t) = \underline{F}(t)$. For example, if F is the NOR function, $F = (x_1 + x_2)'$, then $\underline{F}$ is given by Fig. 8.

t_1 \ t_2	0	$\frac{1}{2}$	1
0	1	$\frac{1}{2}$	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
1	0	0	0

$\underline{F}(t_1,t_2)$

Fig. 8 The ternary extension $\underline{F}$ of $F = (x_1 + x_2)'$.

Let now $F: B^n \times B^s \to B^s$ be the network function for some binary gate network N. Let (x,y) be a primary total state of N, i.e. $F(\bar{x},y) = y$ for some $\bar{x} \in B^n$. To obtain information about the behavior of N, starting from its total state (x,y), the following algorithms A and B may be applied (cf. [EIC]).

Algorithm A.

1. Set $p \leftarrow \mu\{x,\bar{x}\}$;

2. Set $r \leftarrow y$;

3. If $\underline{F}(p,r) = r$ then stop; else

   set $r \leftarrow \underline{F}(p,r)$ and repeat step 3.

We show in Section 5 that this Algorithm must terminate. The value r obtained by Algorithm A is used as the initial value in the following:

Algorithm B.

1. Set $t \leftarrow r$;

2. If $\underline{F}(x,t) = t$ then stop; else

   set $t \leftarrow \underline{F}(x,t)$ and repeat step 2.

The termination of this algorithm will also be shown in Section 5.

The main result of this paper is the following result:

(*) Let t be the value obtained by Algorithm B; then $\bar{y} \sqsubseteq t$ for every $\bar{y} \in$ out (R_x,y). See Section 5 for the proof of this result.

We now illustrate Algorithms A and B and (*) by several examples.

Example 1. Network N_1 of Fig. 2.

Note that $F(1,00) = 00$. Let $\bar{x}=1$ and $x=0$. For Algorithm A we have:

Step 1. $p \leftarrow \mu\{0,1\} = \frac{1}{2}$

Step 2. $r \leftarrow y = 00$

Step 3. $\underline{F}(\frac{1}{2},00) = \frac{1}{2}\frac{1}{2} \neq 00$

$r \leftarrow \frac{1}{2}\frac{1}{2}$

Step 3. $\underline{F}(\frac{1}{2},\frac{1}{2}\frac{1}{2}) = \frac{1}{2}\frac{1}{2}$; stop.

For Algorithm B we get

Step 1. $t \leftarrow \frac{1}{2}\frac{1}{2}$

Step 2. $\underline{F}(0,\frac{1}{2}\frac{1}{2}) = \frac{1}{2}\frac{1}{2}$; stop.

Clearly (*) is satisfied. Note also that for this example $\mu(out(R_0,00)) = \frac{1}{2}\frac{1}{2}$. Compare the outcome of the ternary procedure with Fig. 4. We shall return to this example later.

Example 2. Network N_2 of Fig. 5. Note that $F(0,011) = 011$. Let $\bar{x}=0$, $x=1$, and $y=011$. The reader will verify that Algorithm A yields $r = \frac{1}{2}\frac{1}{2}\frac{1}{2}$ and Algorithm B yields $t=101$. Compare with Fig. 5.

Example 3. Consider the network N_4 of Fig. 9.

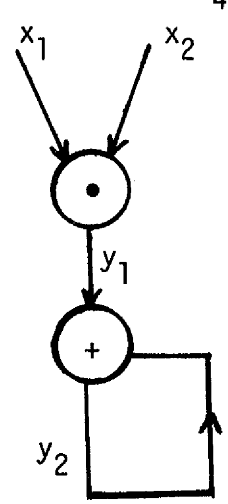


Fig. 9 Network N_4 .

Let $\bar{x}$=01, x=10, y=00.

One easily verifies that

$$\mathrm{cycl}(R_x,y) = \{00\} = \mathrm{out}(R_x,y)$$

The outcome of Algorithm A is $r = \frac{1}{2}\frac{1}{2}$

and of Algorithm B is $t = 0\frac{1}{2}$.

Here, $t \neq \mu(\mathrm{out}(R_x,y))$.

Example 4. Consider the network N_5 of Fig. 10, where the 1-input AND gate represents a delay. Let $\bar{x}$=01, x=10, y=001. One verifies that

$$\mathrm{cycl}(R_x,y) = \{000,010\} = \mathrm{out}(R_x,y)$$

consists of two stable states (critical race). The outcomes of Algorithms A and B are $r = \frac{1}{2}\frac{1}{2}\frac{1}{2}$ and $t = 0\,\frac{1}{2}\,0$, i.e.

$$t = \mu(\mathrm{out}(R_x,y)).$$

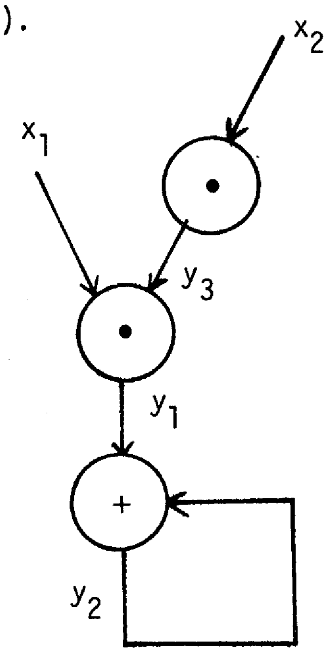


Fig. 10 Network N_5 .

4. Evaluation of the Ternary Procedure

In this section we compare the ternary procedure described in Section 3 with the GMW-model. This comparison will clarify both the advantages as well as the limitations of the ternary procedure.

Refer now to Algorithm A, and let $\bar{r}$ be some intermediate value of r. In Section 5 we show that $\bar{r} \sqsubseteq \underline{F}(p,\bar{r})$. Thus $\underline{F}(p,\bar{r})$, the next value of r, is obtained from $\bar{r}$ by changing some of its binary entries into $\frac{1}{2}$'s. It follows that Algorithm A must terminate after at most s applications of Step 3. A similar argument (see Section 5) shows that Algorithm B must terminate after at most s applications of Step 2. Thus the complexity of the ternary procedure grows linearly with s, whereas the complexity of a procedure based on the GMW-model grows exponentially with s because there are 2^s states. This is, of course, the basic advantage of the ternary procedure.

Assume now that the outcome t of Algorithm B is binary, i.e. $t \in B^s$, as in Example 2 of Section 3. In view of (*),

$$\text{out}(R_x,y) = \text{cycl}(R_x,y) - \text{trans}(R_x,y) = \{t\}.$$

Thus, $\text{cycl}(R_x,y)$ consists of a cycle of length 1, namely $\{t\}$, and, perhaps, one or more transient cycles (see Fig. 5(b) for an example with one transient cycle). It follows that the network in question will eventually reach the stable state t. Hence the ternary procedure provides, in this case, complete information about the steady-state behavior of the network.

However, if the outcome t of Algorithm B is not binary, i.e. $t \in T^s - B^s$, the interpretation of this outcome is not straightforward. We consider the various possibilities leading to a non-binary outcome t.

1) $out(R_x, y)$ consists of a single cycle of length more than one (see Fig. 6). This indicates that the steady-state behavior of the network consists of an oscillation. It easily follows from our theorem (see also Section 5) that

$$\mu(out(R_x, y)) \sqsubseteq t .$$

Hence t must have one or more $\frac{1}{2}$ - entries. If the occurrence of $\frac{1}{2}$ - entries in t is interpreted as "critical race or oscillation", then the outcome of the ternary procedure in case of an oscillation can be considered satisfactory.

2) $out(R_x, y)$ contains at least two cycles which are not match-dependent (as is the case in Fig. 2). This indicates a critical race. Again, since

$$\mu(out(R_x, y)) \sqsubseteq t$$

t cannot be binary, and the outcome of the ternary procedure can be considered satisfactory.

3) Consider now the Network N_1 of Fig. 2 with the addition of an AND-gate as shown in Fig. 11.

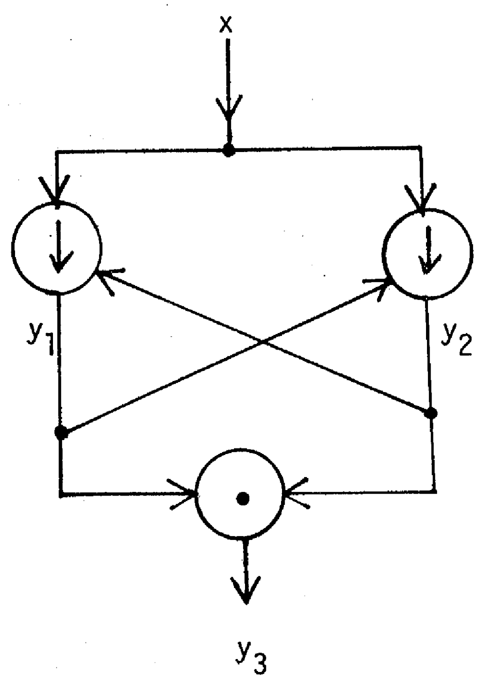


Fig. 11 Network N_6

Let $\bar{x}$=1, y=000, and x=0. The outcome of Algorithm B is $t = \frac{1}{2}\frac{1}{2}\frac{1}{2}$.
The corresponding relation $R_0(000)$ is shown in Fig. 12.

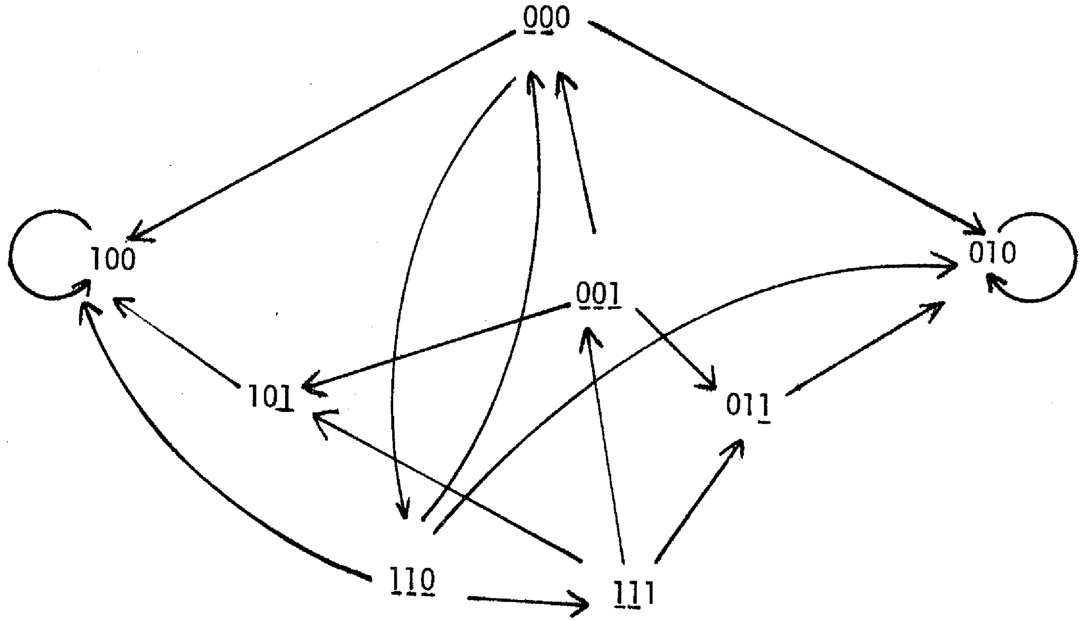


Fig. 12 Relation $R_0(000)$ for N_6

Since we may assume that the network will not move along match-dependent cycles infinitely many times, it follows from Fig. 12 that y_3 will eventually become 0. However, this fact will not be detected by the ternary procedure, since $t_3 = \frac{1}{2}$. Indeed, the ternary procedure treats match-dependent cycles in the same way as sustained cycles. This is the kind of 'pitfall' of the ternary procedure, referred to in [BRE].

(4) Next consider Example 3 of Section 3. Here $out(R_x,y) = \{00\}$. Since $out(R_x,y)$ consists of a single cycle of length 1, one would expect the outcome of Algorithm B to be binary. However, $t = 0\frac{1}{2}$.

On the other hand, if one compares Example 3 with Example 4, it becomes evident that the GMW-model does not take into account line delays. In order to account for such delays one must include them explicitly in the model, as in Example 4. As for the ternary procedure, one easily verifies that this procedure always takes into account line delays, even if such delays have not been explicitly included in the representation of the network. This explains the outcome of the ternary procedure in Example 3. Whether this feature of the ternary procedure is desirable or not, will depend on the particular technology in use. Frequently, line delays may be neglected. In these cases the feature in question constitutes a limitation of the ternary procedure.

5) The network N_5 (see Fig. 10) of Example 4 is obtained from the network N_4 (see Fig. 9) of Example 3 by the addition of an input line delay. However, the preceding discussion is not limited to input line delays. To illustrate this, consider network N_7 of Fig. 12.

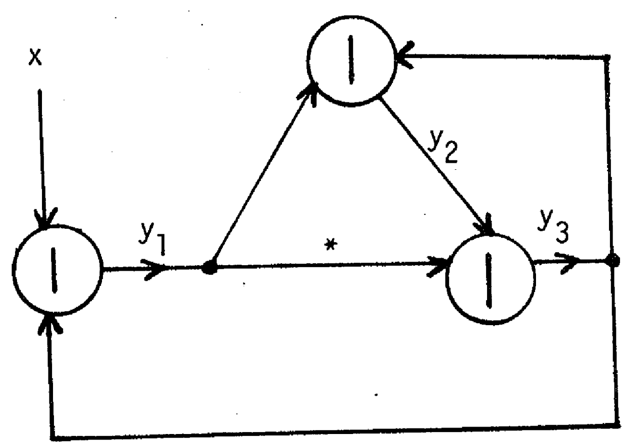


Fig. 12 Network N_7 .

Let $\bar{x}=0$, $y=101$, and $x=1$. In the GMW-model we obtain $cycl(R_x,y) = out(R_x,y) = \{011\}$, whereas the outcome of Algorithm B is $t = \frac{1}{2}\frac{1}{2}\frac{1}{2}$.

However, if we insert a line delay in the connection indicated by * in Fig. 12, the GMW model will yield a critical race. This shows that the ternary procedure also takes into account internal line delays.

In summary, we point out the following:

1. The ternary procedure disregards transient cycles. From the point of view of steady-state behavior this is the desirable approach.

2. Match-dependent cycles are treated by the ternary procedure in the same way as sustained cycles. In view of the fact that oscillations resembling match-dependent cycles can appear in certain networks (the "glitch phenomenon" [CH-MO]), the feature of the ternary model may be quite useful. On the other hand, match-dependent cycles are unlikely to persist, and the ternary model is overly pessimistic with respect to the steady-state behavior.

3. The ternary model does not distinguish between oscillations and critical races, where by critical race we mean that it is possible for the network to reach two or more nontransient cycles. This is probably acceptable for most purposes.

4. The ternary model takes into account line delays. This may or may not be applicable, depending on the technology.

It follows that, if the outcome of the ternary procedure is a binary state, the steady-state analysis is complete. Otherwise, further analysis (using, the GMW model for example) may be required.

Note that we have not given a precise characterization of the result of the ternary procedure. We conjecture the following:

Let N be a gate network, and let $\tilde{N}$ be the network obtained from N by inserting line delays in all fan-out connections and in all input lines. Let $\tilde{R}_x$ be the GMW relation for $\tilde{N}$ under x. Then for $y \in B^s$,

$$\mu(out(\tilde{R}_x, y)) = t$$

where t is the outcome of the corresponding ternary procedure applied to N.

5. Mathematical Theory

In this section we develop the mathematical background required to prove the theorem. We will need the following properties of the partial order $\sqsubseteq$.

Proposition 1. Let $F:B^m \to B^s$, $m,s \geq 1$, be a function and let $\underline{F}:T^m \to T^s$ be its ternary extension. Then

(a) $q \sqsubseteq r$ implies $\underline{F}(q) \sqsubseteq \underline{F}(r)$ for all $q,r \in T^m$;

(b) $t \sqsubseteq r$ and $q \in [t,r]$ implies $t \sqsubseteq q \sqsubseteq r$;

(c) $t,r \sqsubseteq u$ and $q \in [t,r]$ implies $q \sqsubseteq u$.

Proof This is easily verified. □

From now on we use the notation given below without explicit reference.

We consider a network function $F:B^n \times B^s \to B^s$, $s \geq 1$, $n \geq 1$. For $x \in B^n$, $y \in B^s$ the relation R_x is the GMW relation defined by (F,x), and $R_x(y) = \{(z,\bar{z}) \in R_x | yR_x^* z\}$. The ternary extension of F is $\underline{F}:T^n \times T^s \to T^s$. For $p \in T^n$, $\underline{R}_p$ denotes a relation analogous to R_x, and defined by $(\underline{F},p)$ as follows:

$$q\underline{R}_p q \text{ iff } q = \underline{F}(p,q), \text{ and for } q \neq \bar{q}$$

$$q\underline{R}_p \bar{q} \text{ iff } \bar{q} \in [q,\underline{F}(p,q)],$$

where the interval is as defined in the binary case.

For $p \in T^n$ and $q \in T^s$, we say that q is positive in $(\underline{F},p)$ iff $q \sqsubseteq \underline{F}(p,q)$. Similarly, q is negative in $(\underline{F},p)$ iff $q \sqsupseteq \underline{F}(p,q)$, where $\sqsupseteq$ is the converse of $\sqsubseteq$.

Lemma 1 Let $p \in T^n$ and let $q,r \in T^s$ be such that q is positive in $(\underline{F},p)$ and $q \; \underline{R}_p^* \; r$. Then $q \sqsubseteq r$ and r is positive in $(\underline{F},p)$.

Proof $q \; \underline{R}_p^* \; r$ implies $q \; \underline{R}_p^k \; r$ for some $k \geq 0$. The proof is by induction on k. If $k=0$ then $q=r$ and the lemma holds. Now assume $k > 0$, and suppose $q \; \underline{R}_p^{k-1} \; t$, and $t \; \underline{R}_p \; r$. By the induction hypothesis $q \sqsubseteq t$ and t is positive. Now $t \; \underline{R}_p \; r$ implies $r \in [t,\underline{F}(p,t)]$. Since t is positive $t \sqsubseteq r \sqsubseteq \underline{F}(p,t)$, by Prop. 1(b). By Prop. 1(a), $\underline{F}(p,t) \sqsubseteq \underline{F}(p,r)$. Altogether, $q \sqsubseteq t \sqsubseteq r \sqsubseteq \underline{F}(p,t) \sqsubseteq \underline{F}(p,r)$. Thus the lemma holds. □

Lemma 2 Under the conditions of Lemma 1, let $u \in T^s$ be such that $q \sqsubseteq u$ and $\underline{F}(p,u) = u$. Then $r \sqsubseteq u$.

Proof Again assume $q \; \underline{R}_p^k \; r$, $k \geq 0$ and proceed by induction on k. The basis, $k=0$, is trivial. Hence suppose $q \; \underline{R}_p^{k-1} \; t$ and $t \; \underline{R}_p \; r$. By the induction hypothesis $t \sqsubseteq u$. Hence $\underline{F}(p,t) \sqsubseteq \underline{F}(p,u) = u$. Since $q \; \underline{R}_p^* \; t$, we can apply Lemma 1, yielding $q \sqsubseteq t$ and t positive in $(\underline{F},p)$. Now $t \; \underline{R}_p \; r$ implies $t \sqsubseteq r \sqsubseteq \underline{F}(p,t)$. Thus $r \sqsubseteq \underline{F}(p,t) \sqsubseteq \underline{F}(p,u) = u$, and the lemma holds. □

Lemma 3 Let $q \in T^s$ be positive in $(\underline{F},p)$. Then $\mathrm{cycl}(\underline{R}_p,q) = \{r\}$, for some $r \in T^s$.

Proof Suppose $r,t \in \mathrm{cycl}(\underline{R}_p,q)$ appear in the same cycle, i.e. $r \; \underline{R}_p^* \; t$ and $t \; \underline{R}_p^* \; r$. Clearly $q \; \underline{R}_p^* \; r$. By Lemma 1 r is positive. Also by Lemma 1 (since $r \; \underline{R}_p^* \; t$), $r \sqsubseteq t$. By symmetry $t \sqsubseteq r$, and it follows that r=t. Hence every cycle reachable from q in $\underline{R}_p$ is of length 1.

Next suppose q $\underline{R}_p^*$ r and q $\underline{R}_p^*$ u, where {r} and {u} are two cycles of length 1. By Lemma 1, q ⊑ u. Also $\underline{F}(p,u)$ = u. By Lemma 2 r ⊑ u. By symmetry, u ⊑ r. Hence u=r. □

It is easily verified that Prop. 1 and Lemmas 1, 2 and 3 remain correct if the relation ⊑ is replaced by the converse relation ⊒, and "positive" is replaced by "negative". We will refer to these "dual" results as Proposition 1^D and Lemmas 1^D, 2^D and 3^D.

<u>Corollary 1</u> Let $\bar{x} \in B^n$ and $y \in B^S$ be such that $y = F(\bar{x},y)$ and $\bar{x} \sqsubseteq p$. Then $cycl(R_p,y) = \{r\}$ for some $r \in T^S$.

<u>Proof:</u> $y = F(\bar{x},y) = \underline{F}(\bar{x},y) \sqsubseteq \underline{F}(p,y)$, by Prop. 1. Hence y is positive in $(\underline{F},p)$ and the claim follows by Lemma 3. □

<u>Corollary 2</u> Let $p \in T^n$ and $r \in T^S$ be such that $r=\underline{F}(p,r)$ and $p \sqsupseteq x$. Then $cycl(\underline{R}_x,r) = \{t\}$, for some $t \in T^S$.

<u>Proof:</u> $r = \underline{F}(p,r) \sqsupseteq \underline{F}(x,r)$, by Prop. 1. Hence r is negative in $(\underline{F},x)$. The claim follows by Lemma 3^D. □

<u>Lemma 4</u> Suppose $y \in B^S$ is positive in $(\underline{F},p)$, y R_x^* $\bar{y}$ for some $\bar{y} \in B^S$, and y $\underline{R}_p^*$ r for some $r \in T^S$ such that $r = \underline{F}(p,r)$. Then $x \sqsubseteq p$ implies $\bar{y} \sqsubseteq r$.

<u>Proof:</u> We must have y $\underline{R}_x^k$ $\bar{y}$ for some $k \geq 0$. The proof is by induction on k. If k=0, $\bar{y}$=y. Since y is positive in $(\underline{F},p)$ and y $\underline{R}_p^*$ r, we have

$y \sqsubseteq r$, by Lemma 1.

Assume now that $y \, R_x^{k-1} \, \hat{y}$ and $\hat{y} \, R_x \, \bar{y}$. By the induction hypothesis, $\hat{y} \sqsubseteq r$. Now $\bar{y} \in [\hat{y}, F(x,\hat{y})]$, and $F(x,\hat{y}) \sqsubseteq \underline{F}(x,r) \sqsubseteq \underline{F}(p,r) = r$. By Prop. 1(c), $\bar{y} \sqsubseteq r$. □

<u>Lemma 5</u> Let $x \in B^n$ and let C be a nontransient cycle of R_x. Let $r \in T^s$ be such that r is negative in $(\underline{F},x)$ and $r \sqsupseteq y$, for every y in C . Suppose $r \, R_x^* \, t$ for some $t \in T^s$. Then $t \sqsupseteq y$, for every y in C .

<u>Proof</u> We must have $r \, \underline{R}_x^k \, t$ for some $k \geq 0$. We use induction on k. If $k=0$, the claim is obvious. Otherwise, let $r \, \underline{R}_x^{k-1} \, u$ and $u \, \underline{R}_x \, t$. By the induction hypothesis $u \sqsupseteq y$ for every y in C . By Lemma 1^D, u is negative in $(\underline{F},x)$. Thus $u \sqsupseteq t \sqsupseteq \underline{F}(x,u)$ since $u \, \underline{R}_x \, t$. We will show that $\underline{F}(x,u) \sqsupseteq y$, for each y in C . Suppose this is not the case. Then $z_i \triangleq [\underline{F}(x,u)]_i$ must be in $\{0,1\}$ and $y_i = z_i'$. Now $u \sqsupseteq y$ implies $z_i = [\underline{F}(x,u)]_i \sqsupseteq [F(x,y)]_i$. Since $z_i \in \{0,1\}$, we have $z_i = [F(x,y)]_i$. Hence $y_i \neq [F(x,y)]_i$. Since the cycle C is non-transient there must exist $\bar{y} \in C$ such that $\bar{y}_i = [F(x,\bar{y})]_i$. Since $\bar{y} \sqsubseteq u$, $\bar{y}_i = [F(x,\bar{y})]_i \sqsubseteq [\underline{F}(x,u)]_i = z_i$. Consequently there must be two states in C, say $\bar{y}$ and y, such that $\bar{y} \, R_x \, y$, $\bar{y}_i = z_i$ and $y_i = z_i'$. But $y \in [\bar{y}, F(x,\bar{y})]$ and $\bar{y}_i = [F(x,\bar{y})]_i = z_i$. Hence $y_i = z_i$, which is a contradiction, and the lemma holds. □

<u>Theorem</u> Let $x,\bar{x} \in B^n$ and $y \in B^s$. Let $p = \mu(x,\bar{x})$, and assume $F(\bar{x},y) = y$. Then

(A) $\text{cycl}(\underline{R}_p, y) = \{r\}$ for some $r \in T^s$.

(B) $\text{cycl}(\underline{R}_x, r) = \{t\}$ for some $t \in T^s$.

(C) Let C be a nontransient cycle of R_x reachable from y, and let $\bar{y} \in C$. Then $\bar{y} \sqsubseteq t$.

<u>Proof</u> Since $\bar{x} \sqsubseteq \mu(x,\bar{x}) = p$, Corollary 1 yields (A). Similarly $x \sqsubseteq p$, and (B) follows by Corollary 2. Since $y = F(\bar{x},y) \sqsubseteq \underline{F}(p,y)$, y is positive in $(\underline{F},p)$. Also $y \; R_x^* \; \bar{y}$, $y \; \underline{R}_p^* \; r$ and $r = \underline{F}(p,r)$ yields $\bar{y} \sqsubseteq r$ by Lemma 4. Finally $r = \underline{F}(p,r) \sqsupseteq \underline{F}(x,r)$. Thus r is negative in $(\underline{F},x)$. By Lemma 5 $\bar{y} \sqsubseteq t$.

The mathematical results above are applied as follows. Part (A) of the Theorem applies to any primary state y and any input x. It states that, in the ternary model, the result of starting the network in state y with input $p = \mu(\bar{x},x)$ is a unique stable ternary state r. This resulting state r is independent of the path chosen from y in $\underline{R}_p(y)$, and this justifies Algorithm A, where we take the "maximum winner" path i.e. the path consisting of steps of the form $t \; \underline{R}_p \; \underline{F}(p,t)$. Similarly, Part (B) justifies algorithm B, and part (C) provides a proof of (*) of Section 2.

References

[BRE] Breuer, M.A., "A Note on Three-Valued Logic Simulation". IEEE Trans. Computers, Vol. C-21, 1972, pp. 399-402.

[BR-YO] Brzozowski, J.A., and M. Yoeli, Digital Networks. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1976.

[CH-MO] Chaney, T.J., and C.E. Molnar, "Anomalous Behavior of Synchronizer and Arbiter Circuits". IEEE Trans. Computers, Vol. C-22, 1973, pp. 421-422.

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

[JE-MC-VO] Jephson, J.S., R.P. McQuarrie, and R.E. Vogelsberg, "A Three-Value Computer Design Verification System". IBM Syst. J., Vol. 8, 1969, pp. 178-188.

[PU-RO] Putzolu, G.R., and J.P. Roth, "A Heuristic Algorithm for Testing Asynchronous Circuits". IEEE Trans. Computers, Vol. C-20, 1971, pp. 639-647.

[YO-BR] Yoeli, M. and J.A. Brzozowski, "Ternary Simulation of Binary Gate Networks". (To appear.)