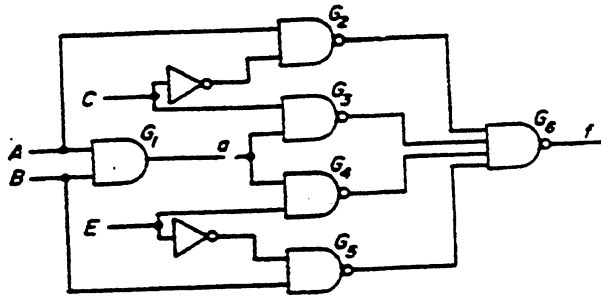


In general alternatives must be considered in every step of path sensitization algorithm



Fault a s-a-0

Requires $G_1 = 1 \Rightarrow A = B = 1$

Path (Choice 1) $G_3 G_6$

$$G_3 \Rightarrow C = 1$$

$$G_6 \Rightarrow G_2 = G_4 = G_5 = 1$$

$$G_2 = 1 \Rightarrow \underline{A = 0} \text{ or } \underline{C = 1}$$

No Contradiction ok

$$G_4 = 1 \Rightarrow \underline{E = 0} \text{ or } \underline{AB = 0} \Rightarrow \underline{A = 0 \text{ or } B = 0}$$

Contradiction

$$G_5 = 1 \Rightarrow \underline{E = 1} \text{ or } \underline{B = 0}$$

Contradict Contradict

No Test ?

Try Boolean Difference

4.21

$$f = A\bar{C} + G_1C + G_1E + B\bar{E}$$

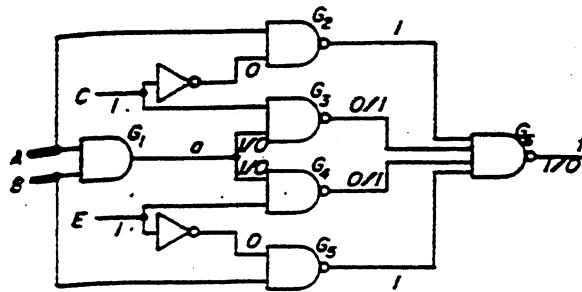
$$\frac{df}{dG_1} = (A\bar{C} + B\bar{E}) \oplus (A\bar{C} + C + E + B\bar{E}) = C(\bar{B} + E) + E(\bar{A} + C)$$

$$G_1 = AB.$$

The set of tests which detects G_1 s-a-0 is defined by the Boolean expression

$$G_1 \frac{df}{dG_1} = AB(C(\bar{B} + E) + E(\bar{A} + C)) = ABCE. \quad (1,1,1,1)$$

Analyze circuit for test input

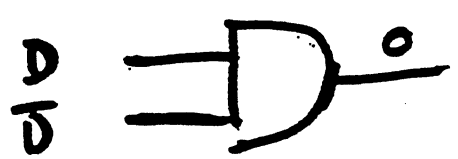


Note: Multiple paths sensitized
both $G_3 G_6$ and $G_4 G_6$

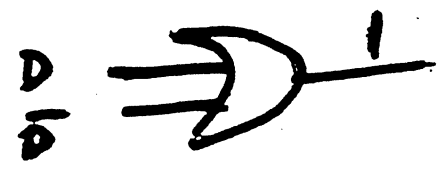
Use $D = \begin{matrix} \swarrow \\ 1 \\ \searrow \\ 0 \end{matrix}$ ← Normal Circuit Signal Value
← Faulty Circuit Signal Value

$$\bar{D} = 0/1$$

Note



$$D \cdot \bar{D} = 0$$



$$D + \bar{D} = 1$$

AND

	0	1	D	$\bar{D}$
0	0	0	0	0
1	0	1	D	$\bar{D}$
D	0	D	D	0
$\bar{D}$	0	$\bar{D}$	0	$\bar{D}$

(a)

OR

	0	1	D	$\bar{D}$
0	0	1	D	$\bar{D}$
1	1	1	1	1
D	D	1	D	1
$\bar{D}$	$\bar{D}$	1	1	$\bar{D}$

(b)

Fault Equivalence, Dominance, Collapsing

Set of tests which detect α $T_\alpha = f \oplus f_\alpha$

Set of tests which detect β $T_\beta = f \oplus f_\beta$

Set of tests which distinguish α, β

$$T_\alpha \oplus T_\beta = (f \oplus f_\alpha) \oplus (f \oplus f_\beta) = f_\alpha \oplus f_\beta$$

If $f_\alpha = f_\beta \Rightarrow T_\alpha = T_\beta$

$\Rightarrow$ no tests distinguish α, β

$\Rightarrow \alpha$ and β are equivalent



$2(n+1)$ single stuck faults
all s-a-0 equivalent
Reduce to $n+1$ faults

$2(n+1)$ faults
all s-a-1 equivalent
Reduce to $n+1$ faults

Equivalence Fault
Collapsing

α dominates B if $T_B \subset T_\alpha$

(detecting B insures α will also be detected)



output s-a-1 $\rightarrow$ detected by $\overline{x_1 x_2 \dots x_n} = (\overline{x_1} + \overline{x_2} + \dots + \overline{x_n})$

input x_1 s-a-1 $\rightarrow$ detected by $\overline{x_1} x_2 \dots x_n$

input x_n s-a-1 $\rightarrow$ detected by $x_1 x_2 \dots \overline{x_n}$

output s-a-1 dominates input s-a-1



output s-a-0 $\rightarrow$ tested by $(x_1 + x_2 + \dots + x_n)$

input x_1 s-a-0 $\rightarrow$ tested by $x_1 \overline{x_2} \dots \overline{x_n}$

output s-a-0 dominates input s-a-0

AND (NAND) gate

(1) - (n): each input s-a-1
(n + 1): any input s-a-0

OR (NOR) gate

(1) - (n): each input s-a-0
(n + 1): any input s-a-1

Figure 2.27 The Set of $n + 1$ Basic Faults Associated With Each Gate Type

Theorem 2.2: In a fanout free combinational circuit C , any set of tests which detects all stuck faults on primary inputs will detect all stuck faults.

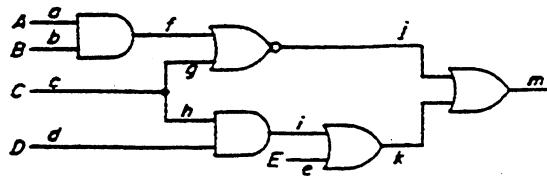
Proof: Assume a set of tests T detects all stuck faults on primary inputs of C but does not detect all internal faults. Then there must be some gate G

in C such that T detects all faults on the inputs of G but does not detect some output fault. Assume G is an AND gate. Then the output of G $s-a-0$ is equivalent to any input $s-a-0$ and the output $s-a-1$ dominates any input $s-a-1$. Therefore if all input faults of G are detected both output faults will also be detected. A similar proof holds if G is an OR, NAND, or NOR gate. $\square$

Theorem 2.3: In a combinational circuit any test which detects all single (multiple) stuck faults on all primary inputs and all branches of fanout* points detects all single (multiple) faults. The set of primary inputs and branches of fanout points are called the *checkpoints* of the circuit.

Proof: For single faults the proof is similar to that of Theorem 2.2. The proof for multiple faults follows from the fact that if line i is a predecessor of line j (i.e., there is a connected path from i to j) then any multiple fault of the form $(i, j), a, b = 0$ or 1^{**} dominates the single fault j_0 or j_1 . Thus any multiple stuck fault can be reduced to a set of stuck faults, no one of which is a predecessor of any other. Each of these dominates a single stuck fault on some checkpoint and hence the multiple fault dominates a multiple stuck checkpoint fault. $\square$

Example 2.11: The circuit of Figure 2.28 has 24 single stuck faults. Using



Theorem 2.3, this set of faults can be collapsed to 10 input faults plus 4 faults on branches of fanout points $\{g_0(g \text{ s-a-0}), g_1, h_0, h_1\}$. This set of fourteen faults can then be further collapsed. Since $A \text{ s-a-0}$ is equivalent to $B \text{ s-a-0}$, we can delete the latter. The fault g_1 is equivalent to f_1 which dominates $A \text{ s-a-1}$. Therefore, g_1 can be eliminated. Since h_0 is equivalent to $D \text{ s-a-0}$, the latter can be eliminated. Finally $E \text{ s-a-1}$ is equivalent to i_1 which dominates h_1 . Therefore $E \text{ s-a-1}$ can be eliminated. The original set of 24 faults has thus been reduced to 10.

$a_0 a_1$
 $b_0 b_1$
 $c_0 c_1$
 $d_0 d_1$
 $e_0 e_1$
 $f_0 f_1$
 $g_0 g_1$
 $h_0 h_1$
 $i_0 i_1$
 $j_0 j_1$
 $k_0 k_1$
 $e_0 e_1$
 $m_0 m_1$

$\Rightarrow$
Th 2.3

$a_0 a_1$
 $b_0 b_1$
 $c_0 c_1$
 $d_0 d_1$
 $e_0 e_1$

} input

$g_0 g_1$
 $h_0 h_1$

} fanout points

$a_0 a_1$
 $b_0 b_1$
 $c_0 c_1$
 $d_0 d_1$
 $e_0 e_1$
 $g_0 g_1$
 $h_0 h_1$

- ① $a_0 = b_0$
- ② $g_1 = f_1, d a_1$
- ③ $h_0 = d_0$
- ④ $e_1 = i_1, d h_1$

(24)

Of course this simple method of fault collapsing may fail to discover some equivalent faults in a circuit such as c s-a-1 and d s-a-1 in the circuit of Figure 2.29. However determination of equivalent faults which are not

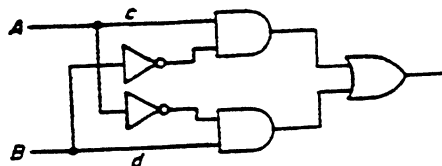


Figure 2.29

structurally related (in the manner specified in Theorem 2.3) would require a prohibitive amount of computation.

$$\begin{array}{lll}
 c \text{ s-a-1} & A=0, B=0 & T_\alpha = \bar{A}\bar{B} \\
 d \text{ s-a-1} & B=0, A=0 & T_\beta = \bar{A}\bar{B}
 \end{array}$$

$$T_\alpha = T_\beta \Rightarrow \alpha \text{ equiv. to } \beta.$$

Redundant Circuits

(4.28)

If $f = f_\alpha$ then $T = f \oplus f_\alpha = 0$

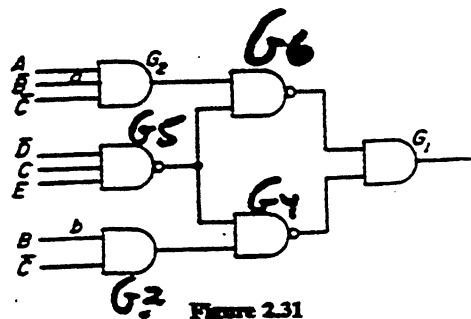
(no tests exist which detect fault α)

α is Undetectable

Circuit is redundant with respect to α . For any circuit which is redundant with respect to a stuck type fault, circuit can be simplified.

<u>Undetectable fault</u>	<u>Simplification Rule</u>
AND(NAND) input $s-a-1$	Remove input
AND(NAND) input $s-a-0$	Remove gate, replace by 0(1)
OR(NOR) input $s-a-0$	Remove input
OR(NOR) input $s-a-1$	Remove gate, replace by 1(0)

Figure 2.30 Simplification of Redundant Circuits



Fault α (a s-a-1)

$$G_2 \quad \bar{B}=0 \Rightarrow B=1 \quad A=1, \bar{C}=1 \rightarrow C=0$$

$$G_6 \quad G_5=1 \Rightarrow \bar{D}=0 \quad \text{or } \underbrace{C=0}_{\text{already specified}} \quad \text{or } E=0$$

$$G_1 \quad G_4=1 \Rightarrow \underbrace{G_5=0}_{\text{Contradict}} \quad \text{or } G_3=0$$

$$G_3=0 \Rightarrow \underbrace{B=0}_{\text{Contradict}} \quad \text{or } \underbrace{\bar{C}=0}_{\text{Contradict}}$$

No test exists.

Fault β (b s-a-0)

$$B=1, \bar{C}=1 \rightarrow C=0 \quad (\text{Gate } G_3)$$

$$G_4: G_5=1 \rightarrow \bar{D}=0 \quad \text{or } \underbrace{C=0}_{\text{already specified}} \quad \text{or } E=0$$

$$G_1: G_6=1 \rightarrow G_2=0 \quad \text{or } \underbrace{G_5=0}_{\text{Contradiction}}$$

$$G_2=0 \rightarrow A=0 \quad \text{or } \underbrace{\bar{B}=0}_{\text{already spec}} \quad \text{or } \underbrace{\bar{C}=0}_{\text{Contradict}}$$

One test is ~~$A\bar{B}\bar{C}$~~ ABC
Another test is $\bar{A}B\bar{C}$

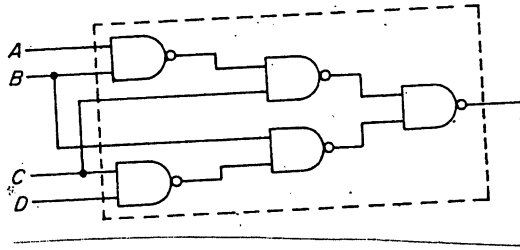
However, in the presence of the undetectable fault α ($a s-a-1$), the fault $b s-a-0$ is not detectable by ABC . The reason for this is as follows. In order to detect the fault $b s-a-0$ the test must generate a 1 on the top input to gate G_1 . This is done for the test ABC by setting line a to 0 and thus producing a zero output from gate G_2 . However, since the fault $a s-a-1$ is not detectable, this path will be deactivated for the test ABC if this undetectable fault occurs, and the test ABC will not detect the fault $b s-a-0$. This fault can be detected only by $\bar{A}B\bar{C}$ in the presence of $a s-a-1$. The set of tests $\{A\bar{B}CD, \bar{A}\bar{B}\bar{C}, \bar{A}\bar{B}\bar{C}DE, ABCDE, BCD\}$ detects any detectable single fault in the original circuit but does not detect $b s-a-0$ in the presence of the undetectable fault $a s-a-1$.

Thus in deriving a complete test set for a redundant circuit we must generate tests for any multiple fault which can arise due to a sequence of undetectable faults. This greatly increases the difficulty of test generation for redundant circuits. Many other problems can arise in redundant circuits including the following: (1) if α is a detectable fault and γ is an undetectable fault, fault α may be undetectable in the presence of fault γ (see Problem 2.13). The latter fault is called a *second-generation* redundant fault. (2) An undetectable fault α may become detectable in the presence of another undetectable fault β (See Problem 2.14). Therefore, a complete test set must contain a test which detects the multiple fault (α, β) although it does not detect any of the corresponding single faults in the original redundant circuit. (3) If α and β are two distinguishable faults and γ is an undetectable fault, faults α and β may become indistinguishable in the presence of fault γ (See Problem 2.15).

Because of the difficulties inherent in test generation for redundant circuits, it is desirable to eliminate such redundancy via circuit simplification as specified in the table of Figure 2.30. However, in general there is no simple manner to determine redundancy in combinational circuits other than proving that no test exists for some fault or faults.

In presence of α (which is undetectable) β is only tested by $\bar{A}B\bar{C}$

Multiple stuck faults



Multiple fault $\{B \text{ s-a-1}, C \text{ s-a-1}\}$.

$$F' = \bar{A} + \bar{D}$$

$$f = (\bar{A} + \bar{B}) \cdot C + (\bar{C} + \bar{D}) \cdot B$$

$$T = f \oplus f' = \bar{A}\bar{B}\bar{C} + \bar{B}\bar{C}\bar{D} + AD(\bar{B}\bar{C} + \bar{B}C)$$

set of tests which detect multiple fault

$$T' = \{ABCD, \bar{A}BCD, ABC\bar{D}, A\bar{B}\bar{C}D, A\bar{B}C\bar{D}, \bar{A}B\bar{C}D\}$$

T' detects all single faults

but does not detect multiple fault

thus it is possible for single faults to be detected while multiple faults are not (but not probable) unlikely

Some Theoretical Results about Multiple Faults in Combinational Circuits

4.29

- (1) In irredundant two-level combinational circuits, any set of tests which detects all single stuck faults also detects all multiple stuck faults.
 - (2) In fanout-free circuits (i.e., circuits where each primary input and each gate output are inputs to at most one gate), there exists a single stuck fault test set of minimal cardinality which detects all multiple stuck faults.
 - (3) The *checkpoints* of a combinational circuit consist of all inputs and branches of fanout points. A set of tests which detects all multiple stuck faults on the checkpoints of a circuit will detect all multiple stuck faults in the circuit.
 - (4) In a combinational circuit C any set of tests which detects all single stuck faults will also detect all multiple stuck faults unless C contains a subcircuit corresponding to the circuit shown within dotted lines in Figure 2.36 [27].
-

An erroneous signal (D or $\bar{D}$) may propagate onto a line l which is also faulty due to a multiple fault.

Value of Fault Signal Propagated onto line l	Fault on line l	Resultant Signal on line l
D	$s-a-0$	D
D	$s-a-1$	1
$\bar{D}$	$s-a-0$	0
$\bar{D}$	$s-a-1$	$\bar{D}$

Figure 2.37

Example 2.12: Consider the circuit of Figure 2.38(a) and the multiple fault $\{A \text{ s-a-1}, h \text{ s-a-1}\}$.

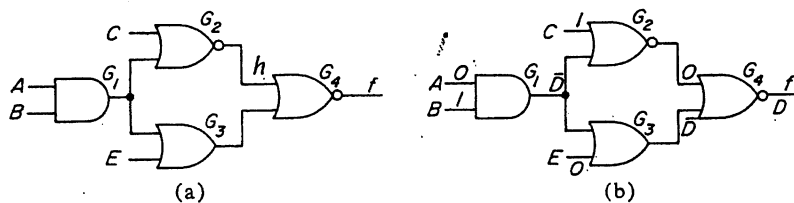


Figure 2.38

To generate a $\bar{D}$ at the output of G_1 we specify $A = 0, B = 1$. To propagate along the path G_2G_4 we specify $C = 0$, thus generating a D on the faulty line h . From the table of Figure 2.37 the propagation of a D onto a $s\text{-a-1}$ line results in a 1 signal. Thus the fault cannot be propagated along this path. Alternatively to propagate the $\bar{D}$ along G_3G_4 we specify $E = 0$. In justifying the input to G_4 we determine that the signal on h cannot be set to 0 but can be made $\bar{D}$ by specifying $C = 1$. Thus the test $\bar{A}BCE$ detects the multiple fault. Figure 2.38(b) shows the resultant signal propagation. $\square$

Generation of tests to Distinguish Two Faults

Fault α from Fault β

Must detect α but not β or vice-versa

Must keep track of 3 values for each signal,
normal value, value for α , value for β

$$D(\bar{D}) = \overset{\leftarrow \alpha}{1/0} (\%)$$

↑
Normal

$$E(\bar{E}) = \overset{\leftarrow \beta}{1/0} (\%)$$

↑
Normal

At most one fault is present.

If D, E propagate to the same gate ^(inputs of the)
the following shows the propagation
of errors to the output of that gate

AND				OR			
		E	$\bar{E}$			E	$\bar{E}$
D		D, E	$\bar{E}$	D		1	D
$\bar{D}$		$\bar{D}$	0	$\bar{D}$		E	$\bar{D}, \bar{E}$
(a)				(b)			

Figure 2.45

For a test output signal should be
 D or $\bar{D}$ or E or $\bar{E}$
(not D, E) $\leftarrow$ detects both α and β

Example 2.15: For the circuit of Figure 2.46, we wish to derive a test to distinguish the “D” fault A s-a-1 and the “E” fault C s-a-0. We begin by trying to propagate a faulty signal from A s-a-1. Setting $A = 0$, $B = 1$,

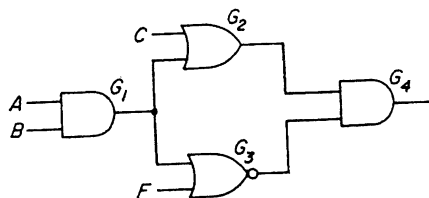


Figure 2.46

results in $G_1 = \bar{D}$. If we attempt to propagate this $\bar{D}$ through G_2 we observe that if $C = 0$ the output of G_2 is $\bar{D} + 0 = \bar{D}$. However if $C = 1$ the

output of G_2 is $\bar{D} + E = E$. Similarly the output of G_3 is D if $F = 0$ and is 0 if $F = 1$. Thus the input $\bar{A}BC\bar{F}$ results in an output of G_4 equal to $E \cdot D = D, E$ which does not distinguish the faults. The input $\bar{A}B\bar{C}F$ results in G_4 equal to $\bar{D} \cdot D = 0$ which does not distinguish the faults, and the inputs $\bar{A}BF$ result in the output 0. Hence the faults cannot be distinguished if the error signal D is propagated from A s-a-1. We must thus propagate a faulty signal E from C s-a-0. Setting $C = 1$, $B = 0$ results in the output of G_2 of $E + 0 = E$. Setting $F = 0$ results in the output of G_4 of $E \cdot 1 = E$. Thus $\bar{B}C\bar{F}$ distinguishes the faults. If $B = 1$ the output of G_1 is 1 if $A = 1$ and is $\bar{D}$ if $A = 0$. In either case no distinguishing test can be generated. Hence the only tests which distinguish these faults are defined by the Boolean expression $\bar{B}C\bar{F}$. $\square$

Non-adaptive testing

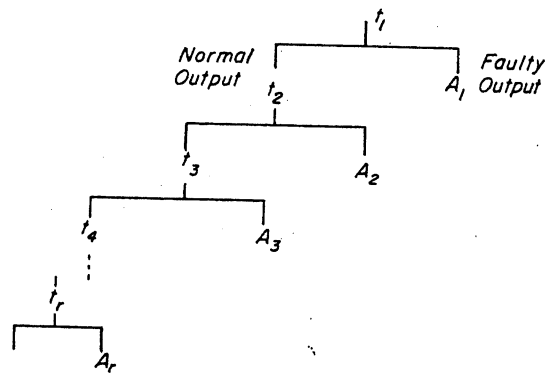


Figure 2.47

Test sequence fixed

$t_1, t_2, \dots, t_r$

(independent of results)

Used for fault detection

Adaptive Testing

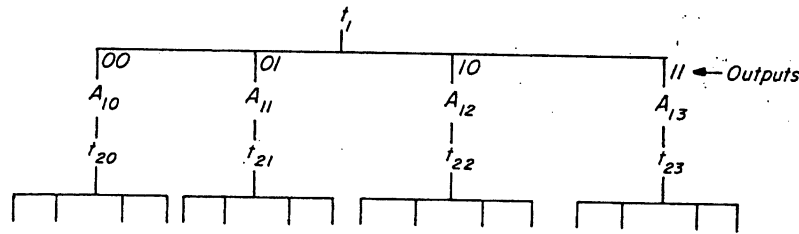


Figure 2.48

i th test applied depends on
results of all previous tests
(1, 2, 3... $i-1$)

Used for fault diagnosis

Adaptive testing may substantially
reduce the average number of
tests required for fault diagnosis.

Critical Path Test Generation

4.36

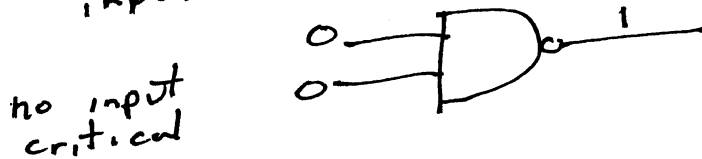
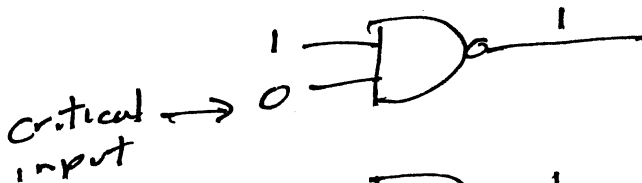
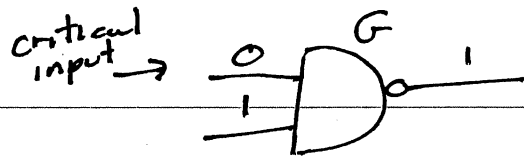
Test Generation: not based on specific fault

Random Test Generation

CPTG - along a "sensitized" path
half of stuck type faults
are detected by a test.

Goal

Generate tests which produce
long sensitized paths



if "critical" input changes^{value} the
output value will change

* Attempt to drive critical inputs
back from output to primary inputs
of circuit.

Example

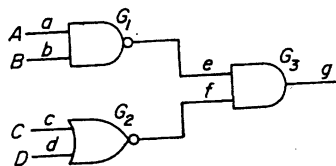


Figure 2.24

$$\begin{aligned}
 g = 0^c &\rightarrow e = 0^c, f = 1 \rightarrow A = 1^c, B = 1^c, C = 0, D = 0 \\
 &\text{OR } e = 1, f = 0^c \rightarrow A = 0 \text{ or } B = 0, C = 1^c, D = 0 \\
 &\text{OR } C = 0, D = 1^c
 \end{aligned}$$

$$\begin{aligned}
 g = 1^c &\rightarrow e = 1^c, f = 1^c \rightarrow A = 1, B = 0^c \text{ AND } C = 0^c, D = 0^c \\
 &\text{OR } A = 0, B = 1^c
 \end{aligned}$$

Example 2.9: Consider the circuit of Figure 2.24. Initially set $g = 0$. To define critical inputs to G_3 we set $e = 0, f = 1$ or $e = 1, f = 0$. Consider-

ing the first alternative, if $e = 0$, then $A = B = 1$ and G_1 has both inputs critical. The sensitized path has been driven back to the inputs. The condition $f = 1$ requires $C = D = 0$. Thus the test $(1, 1, 0, 0)$ has been gener-

ated. Considering the second alternative the critical signal $f = 0$ is driven back through G_2 by specifying $C = 0, D = 1$, or $C = 1, D = 0$. The condition $e = 1$ is justified by $A = 0$ or $B = 0$. (Since $e = 1$ is not a critical signal no purpose is served in defining A or B as critical signals). Additional tests can be generated by initially defining the output $g = 1$. This implies $e = f = 1$ with both signals critical. The critical signal $e = 1$ can be driven back by specifying $A = 0, B = 1$ or $A = 1, B = 0$. The critical signal $f = 1$ requires $C = D = 0$. Thus the two tests $(0, 1, 0, 0)$ and $(1, 0, 0, 0)$ are generated. $\square$

The Critical Path Test Generation algorithm for combinational circuits is thus as follows:

Procedure 2.3 (*Critical Path Test Generation*):

- (1) Select an output line. Define it as a critical 0 and drive this critical value back towards the inputs using sensitizing cubes. Whenever a choice exists, select one alternative and backtrack later to consider all possibilities. For each choice made, all implications of that choice are carried out, and all critical lines generated are marked. When all critical lines have been driven back, all non-critical lines are justified using primitive cubes. If a critical line cannot be driven back via a sensitizing cube because of an inconsistency, backtrack and use a primitive cube.
 - (2) Repeat for the output line initialized as a critical 1.
 - (3) Repeat 1,2 for the rest of the output lines. □
-

Example 2.10: Consider the circuit of Figure 2.25.

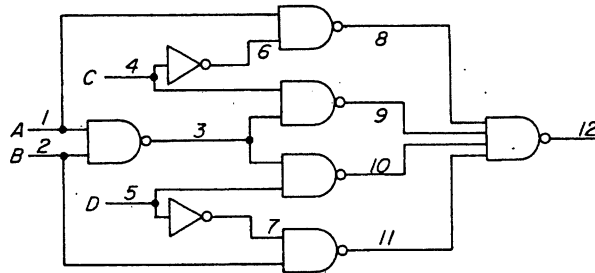


Figure 2.25

The table of Figure 2.26 shows the computation for the critical path test derivation procedure for this circuit.

	row	1	2	3	4	5	6	7	8	9	10	11	12
	1												0 ^c
	2								1 ^c	1 ^c	1 ^c	1 ^c	0 ^c
	3	0 ^c					1		1 ^c	1 ^c	1 ^c	1 ^c	0 ^c
(Implication)	4	0 ^c	0 ^c	1	0 ^c	0 ^c	1	1	1 ^c	1 ^c	1 ^c	1 ^c	0 ^c
	5	1					0 ^c		1 ^c	1 ^c	1 ^c	1 ^c	0 ^c
(Implication)	6	1	1	0 ^c	1	1	0 ^c	0 ^c	1 ^c	1 ^c	1 ^c	1 ^c	0 ^c
	7								0 ^c	1	1	1	1 ^c
	8	1 ^c	0		0 ^c	0	1 ^c		0 ^c	1	1	1	1 ^c

Figure 2.26

Starting with a critical 0 (denoted as 0^c) on line 12 implies critical 1's on lines 8, 9, 10, 11 (second row of table of Figure 2.26). We now have a choice as to which of these critical 1's to drive backwards and we elect to

drive back line 8, and select the sensitized cube having 0 on line 1 and 1 on line 6 (third row of table). This implies the value 0 on lines 2, 4, 5 and the value 1 on lines 3 and 7 and results in the critical values indicated by c's in the fourth row of the table of Figure 2.26. We then backtrack to our last choice and justify line 8 by the other sensitized cube having a 0 on line 6 and a 1 on line 1 (rows 5 and 6 of Figure 2.26). At this stage we would ordinarily backtrack to consider backward drive on lines 9, 10, 11. However in the course of the backward drive on line 8 critical values have already been generated along these paths. We next begin with a critical 1 on line 12 and justify it with a critical 0 on line 8 and 1's on lines 9, 10, 11 (row 7). The critical 0 on line 8 implies critical 1's on lines 1 and 6 and a critical 0 on line 4. No more sensitized paths can be generated and the 1's on lines 10 and 11 are justified by 0's on lines 2 and 5. We would next backtrack to consider a critical 0 on lines 9, 10, 11 respectively. □