

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4488595号
(P4488595)

(45) 発行日 平成22年6月23日(2010.6.23)

(24) 登録日 平成22年4月9日(2010.4.9)

(51) Int. Cl. F I
 GO 1 R 31/3183 (2006.01) GO 1 R 31/28 Q
 GO 1 R 31/28 (2006.01) GO 1 R 31/28 F

請求項の数 14 (全 53 頁)

(21) 出願番号	特願2000-171765 (P2000-171765)	(73) 特許権者	390005175 株式会社アドバンテスト 東京都練馬区旭町1丁目32番1号
(22) 出願日	平成12年6月8日(2000.6.8)	(74) 代理人	100121706 弁理士 中尾 直樹
(65) 公開番号	特開2001-349932 (P2001-349932A)	(74) 代理人	100128705 弁理士 中村 幸雄
(43) 公開日	平成13年12月21日(2001.12.21)	(74) 代理人	100066153 弁理士 草野 卓
審査請求日	平成19年1月25日(2007.1.25)	(74) 代理人	100100642 弁理士 稲垣 稔
		(72) 発明者	石田 雅裕 東京都練馬区旭町1丁目32番1号 株式会社アドバンテスト内

最終頁に続く

(54) 【発明の名称】 テストパターン生成方法

(57) 【特許請求の範囲】

【請求項1】

半導体集積回路に含まれ、ゲート遅延故障または断線遅延故障からなる遅延故障を過渡電源電流試験法で検出するためのテストパターン系列を生成する方法であって、

(a) 被試験半導体集積回路において発生する可能性がある、全ての遅延故障を登録した故障リストを生成するステップと、

(b) 上記故障リストから1つずつ故障を選択された故障として逐次選択するステップと、

(c) 上記選択された故障に対応する上記半導体集積回路中の故障箇所である論理ゲートまたは信号線の出力に上記故障の初期信号論理値をあたえる初期化テストパターンと、上記故障箇所の出力信号の状態を変化させ、かつ上記故障箇所に入力が接続した論理ゲートの出力に、上記故障箇所の出力信号の状態変化を伝搬させるための伝搬テストパターンとからなるテストパターン系列を含意操作により生成するステップと、

(d) 上記テストパターン系列生成ステップで生成に成功したテストパターン系列をテストパターンリストに登録するステップと、

(e) 上記故障選択ステップ(b)において未選択の故障がなくなるまで、各ステップ(b)から(d)を繰り返すステップと、

を有することを特徴とするテストパターン生成方法。

【請求項2】

請求項1に記載したテストパターン生成方法において、

クライアントと複数のサーバとを用い、
クライアントは、上記故障リスト生成ステップ (a) を実行し、複数のサーバを起動し、
上記故障リストを複数のサブ故障リストに分割して、複数のサーバに送信し、
上記各サーバは、上記故障選択ステップ (b) を実行し、上記選択された故障に対し、
上記テストパターン系列生成ステップ (c) を実行し、生成に成功したテストパターン系
列をクライアントに返送すると共に上記サブ故障リストに処理されていない他の故障が存
在するか否かを確認し、
上記クライアントは、さらに、上記サーバから返送されたテストパターン系列をテスト
パターンリストに登録して上記登録ステップ (d) を実行する、
 ことを特徴とするテストパターン生成方法。

10

【請求項 3】

請求項 2 に記載したテストパターン生成方法において、
 上記サーバは、上記テストパターン系列生成ステップ (c) でテストパターン系列を求
 めることができなかった場合は、処理する対象故障をクライアントに要求するステップを
 有することを特徴とするテストパターン生成方法。

【請求項 4】

請求項 1 に記載したテストパターン生成方法において、
クライアントと複数のサーバとを用い、
クライアントは、上記故障リスト生成ステップ (a) を実行し、複数のサーバを起動し
、
上記各サーバは、上記クライアントの上記故障リストから 1 つずつ故障を逐次選択して
上記故障選択ステップ (b) を実行し、
上記テストパターン系列生成ステップ (c) を実行し、生成に成功したテストパターン
系列をクライアントに返送し、
上記クライアントは、さらに、上記サーバから返送されたテストパターン系列をテスト
パターンリストに登録して上記登録ステップ (d) を実行する、
 ことを特徴とするテストパターン生成方法。

20

【請求項 5】

請求項 1 乃至 4 の何れかに記載の方法において、
 さらに、
 (f) 上記テストパターンリストに登録したテストパターン系列をそれぞれ用いて、上
記半導体集積回路に対し過渡電源電流故障シミュレーションをおこない、上記用いたテス
トパターン系列に対応して過渡電源電流試験で検出可能な故障を被検出故障としてリス
トに格納する被検出故障リスト作成ステップと、
 (g) 上記被検出故障リストに登録した故障と同じ故障が、上記被検出故障リスト生成
ステップ (f) で生成した故障リスト中に存在するとき、その故障を故障リストから削除
するステップとを備え、
上記故障リスト中に残存する故障のうちから、1 つずつ故障を選択された故障として逐
次選択して、上記選択ステップ (b) を実行する、
 ことを特徴するテストパターン生成方法。

30

40

【請求項 6】

請求項 1 に記載したテストパターン生成方法において、
クライアントと複数のサーバとを用い、
クライアントは、上記故障リストを複数に分割したサブ故障リストを生成して上記故
障リスト生成ステップ (a) を実行し、生成したサブ故障リストを上記複数のサーバにそ
れぞれ送信し、
上記各サーバは、上記クライアントから送信された上記サブ故障リストを基にして、上
記故障選択ステップ (b) を実行し、
上記テストパターン系列生成ステップ (c) を実行し、生成に成功したテストパターン
系列を上記クライアントに返送し、

50

上記クライアントは、さらに上記サーバから返送されたテストパターン系列をテストパターンリストに登録して上記登録ステップ(d)を実行する、
ことを特徴とするテストパターン生成方法。

【請求項7】

請求項1または4、又は6に記載したテストパターン生成方法において、
 上記テストパターン系列生成ステップ(c)でテストパターン系列を求めることができなかつた場合は、上記故障選択ステップ(b)に移ることを特徴とするテストパターン生成方法。

【請求項8】

請求項1乃至7の何れかに記載したテストパターン生成方法において、
 上記テストパターン系列生成ステップ(c)は、上記対象故障に対応する故障箇所上記故障を活性化するための初期値をあたえる初期化テストパターンを求めるステップと、
 上記故障箇所が上記初期値に縮退する故障を仮定し、上記故障箇所を入力とする次段の論理ゲートの出力に上記縮退故障を伝搬する伝搬テストパターンを求めるステップとよりなることを特徴とするテストパターン生成方法。

10

【請求項9】

請求項1乃至7の何れかに記載の方法において、
 上記テストパターン系列生成ステップ(c)は、上記選択された故障に対応する上記半導体集積回路中の故障箇所である論理ゲートまたは信号線の出力に、上記故障に対応する5値論理システムにおける信号を出力させ、かつ上記故障箇所に入力が接続した論理ゲートの出力に、上記信号の影響を伝搬させる5値論理システムにおけるテストパターン系列を
 含意操作により生成するステップであることを特徴とするテストパターン生成方法。

20

【請求項10】

半導体集積回路のパス遅延故障を過渡電源電流試験法で検出するための所望のテストパターン系列を生成する方法であって、

(h)被試験半導体集積回路の全ての可能性があるパス遅延故障の故障リストを生成する故障リスト生成ステップと、

(i)2つ以上のテストパターンからなる上記半導体集積回路に印加すべきテストパターン系列を発生するテストパターン系列生成ステップと、

(j)上記印加すべきテストパターン系列を上記半導体集積回路に与えた場合の遷移シミュレーションをおこない、上記半導体集積回路内部の信号線に生じる遷移信号値列を計算する遷移信号値列計算ステップと、

30

(k)上記遷移信号値列を用いて、上記印加すべきテストパターン系列を上記半導体集積回路に与えた場合に過渡電源電流試験で検出可能なパス遅延故障を被検出故障として、被検出故障リストに登録する被検出故障リストを作成するステップと、

(m)上記被検出故障リストに登録された被検出故障で上記故障リスト中に存在しているものを上記故障リストから削除するステップと、

(n)上記故障リスト中に存在している被検出故障の検出に用いられた、上記印加すべきテストパターン系列を所望のテストパターン系列としてテストパターン系列リストに登録する登録ステップと、

40

(o)上記パターン系列生成ステップに戻り、上記登録ステップまでを繰り返すステップと、を有することを特徴とするテストパターン生成方法。

【請求項11】

請求項10に記載したテストパターン生成方法において、
クライアントと複数のサーバとを用い、
クライアントは、
上記故障リスト生成ステップ(h)を行い、かつ、
上記テストパターン系列生成ステップ(i)を行い、複数の印加すべきテストパターン系列を生成し、

この印加すべきテストパターン系列を、処理をおこなっていない各サーバに、1つずつ

50

繰り返し送信し、

上記各サーバは、

上記遷移信号値列計算ステップ(j)をおこない、クライアントから送信された印加すべきテストパターン系列に基き遷移信号値列を計算し、

上記被検出故障リスト生成ステップ(k)を行い、

上記削除ステップ(m)を行い、上記クライアントの上記故障リスト中に存在する上記被検出故障リストに含まれる被検出故障を上記故障リストから削除し、かつ、

上記登録ステップ(n)を行い、上記クライアントのテストパターン系列リストに対し、上記所望のテストパターン系列を登録し、かつ、その処理終了に際し上記クライアントに通知することを特徴とするテストパターン生成方法。

10

【請求項12】

請求項10に記載したテストパターン生成方法において、

クライアントと複数のサーバとを用い、

クライアントは、

上記故障リスト生成ステップ(h)を行い、かつ、

上記テストパターン系列生成ステップ(i)を行い、

複数の印加すべきテストパターン系列を生成し、

上記各サーバは、

上記クライアントが印加すべきテストパターン系列を1つずつ繰り返し取り込み、上記遷移信号値列計算ステップ(j)をおこない、その取り込んだ印加すべきテストパターン系列に基き遷移信号直列を計算し、上記被検出故障リスト生成ステップ(k)を行い、上記削除ステップ(m)を行い、

20

上記クライアントの上記故障リスト中に存在する被検出故障を上記故障リストから削除し、かつ、上記登録ステップ(n)を行い、上記クライアントのテストパターン系列リストに対し、上記所望のテストパターン系列を登録することを特徴とするテストパターン生成方法。

【請求項13】

請求項10に記載したテストパターン生成方法において、

クライアントと複数のサーバとを用い、

上記クライアントは、

上記故障リスト生成ステップ(h)を行い、かつ、

上記テストパターン系列生成ステップ(i)を行って、全ての可能な印加すべきテストパターン系列のリストを作成し、この全ての可能な印加すべきテストパターン系列のリストを複数のテストパターン系列サブリストに分割し、かつ、それらサブリストを複数のサーバに送信し、

30

上記各サーバは、

クライアントから送信されたサブリストから印加すべきテストパターン系列を1つずつ繰り返して取り出し、上記遷移信号値列計算ステップ(j)を行い、その取り出した印加すべきテストパターン系列に基き遷移信号値列を計算し、上記被検出故障リスト生成ステップ(k)を行い、上記削除ステップ(m)を行い、

40

上記クライアントの上記故障リスト中に存在する被検出故障を上記故障リストから削除し、上記登録ステップ(n)を行い、上記クライアントのテストパターン系列リストに対し上記所望のテストパターン系列を登録し、上記テストパターン系列サブリスト内に印加すべきテストパターン系列がなくなると印加すべきテストパターン系列の取り出し処理を停止するとともに処理終了を通知することを特徴とするテストパターン生成方法。

【請求項14】

請求項1乃至13の何れかに記載の方法において、クライアントは、上記故障リスト内のパス遅延故障数が所定値以下になった場合、処理をおこなっている複数のサーバの処理を直ちに終了させることを特徴とするテストパターン生成方法。

【発明の詳細な説明】

50

【 0 0 0 1 】

【 発明の属する技術分野 】

この発明は半導体集積回路試験に用いられるテストパターンの生成方法にかかわり、とくに、ゲート遅延故障やゲート遅延故障を生じる断線故障、およびパス遅延故障を過渡電源電流信号を用いて検出するためのテストパターン系列を生成する方法に関する。

【 0 0 0 2 】

【 従来技術 】

従来、被試験回路における論理ゲートの遅延時間が異常となるゲート遅延故障 (gate delay fault) を試験するためのテストパターンは、被試験論理ゲートの出力をある信号値 (初期論理値) に初期化するためのテストパターンと被試験論理ゲートの出力が上記初期論理値に固定してしまう故障 (縮退故障) をテストするためのテストパターンを求めることにより生成されていた。図 1 (a) に、被試験集積回路の一例を示す。この回路は入力端子 x_2 と x_3 が NAND ゲート G_1 の入力側に接続され、また入力端子 x_3 と x_4 が NAND ゲート G_2 の入力側に接続され、NAND ゲート G_1 と G_2 の各出力が NAND ゲート G_3 へ供給され、NAND ゲート G_3 の出力と入力端子 x_1 の入力が NOR ゲート G_4 に供給され、NAND ゲート G_3 の出力と入力端子 x_5 の入力が NAND ゲート G_5 へ供給され、NOR ゲート G_4 と NAND ゲート G_5 の各出力が NOR ゲート G_6 へ供給され、NOR ゲート G_6 の出力側が出力端子 z_1 に接続される。

【 0 0 0 3 】

この被試験集積回路において、論理ゲート G_3 における立ち上がり遷移が遅くなるゲート遅延故障 (slow-rise) に対するテストパターン生成は、はじめに図 1 (a) 中に示すように故障ゲート G_3 の出力に初期値 “0” をあたえるテストパターン $x_1 = x, x_2 = x, x_3 = 0, x_4 = x, x_5 = x$ 、 x は “0” と “1” の何れでもよいことを示す (このテストパターンを以下 $v_1 = (x x 0 x x)$ のように表記する) を求め、つぎに故障ゲート G_3 の出力が設定した初期値 “0” に固定してしまう縮退故障を仮定し、この縮退故障を集積回路の出力信号線で検出するためのテストパターン $v_2 = (0 x 1 1 1)$ (図 1 (b)) を求めることによりおこなわれる。したがって、ゲート G_3 における slow-rise 故障を検出するテストパターン系列は $T = \langle v_1, v_2 \rangle = \langle “x x 0 x x”, “0 x 1 1 0” \rangle$ と求められる。ゲート遅延故障に対するテストパターン生成法については、たとえば、E.S.Park and M.R.Mercer, “An Efficient Delay Test Generation System for Combinational Logic Circuits,” Transactions on Computer-Aided Design, 11(7), pp.926-938, 1992 や U.Mahlstedt, “DELTEST: Deterministic Test Generation for Gate Delay Faults,” Proceedings of IEEE International Test Conference, pp.972-980, 1993 などに記載されている。

【 0 0 0 4 】

また回路における信号伝搬経路をパス (paths) と呼ぶ。パスの入力信号線から出力信号線まで信号が伝搬する時間をパス遅延時間 (path delay time) と呼び、パス遅延時間が所定の値より大きくなると回路は正常に動作しなくなる (これをパス遅延故障 (path delay fault) と呼ぶ)。従来、集積回路内のパス遅延故障を試験するためのテストパターン系列 (一般に、遅延故障を試験するためには 2 つのテストパターンが必要であり、これら 2 つのテストパターンはテストベクトルペアまたはテストパターン系列と呼ばれる) は、図 2 に示す 5 値論理 (five logic value system) を用い、図 3 に示す含意表 (implication table) や図 4 に示す経路活性化表 (sensitizing table) にしたがって、被試験回路内の各信号線に上記論理値を割り当てることによって生成されていた。図 2 は、 S_0 はパターン v_1 と v_2 の何れでも “0” であり S_1 はパターン v_1 と v_2 の何れでも “1” であり、 U_0 はパターン v_1 で “x”、パターン v_2 で “0” であり、 U_1 はパターン v_1 で “x”、パターン v_2 で “1”、 XX はパターン v_1 と v_2 の何れでも “x” であることを示す。

【 0 0 0 5 】

図 3 (a) は AND ゲートの二つの入力端子 x_1, x_2 の各入力それぞれ $S_0, U_0,$

10

20

30

40

50

S 1 , U 1 , X X の何れかとなった場合の各組み合わせにおける出力の状態を示している。例えば x_1 が S 0 , x_2 が U 0 は $v_1 = < 0 , x > , v_2 = < 0 , 0 >$ であるから出力は $< 0 , 0 > = S 0$ となることを示している。図 3 (b) は NOR ゲートの二つの入力端子 x_1 , x_2 の各入力それぞれ S 0 , U 0 , S 1 , U 1 , X X の何れかとなった場合の各組み合わせにおける出力の状態を示し、図 3 (c) はインバータの入力端子 x の入力が S 0 , U 0 , S 1 , U 1 , X X である場合の出力端子 z の出力がどのようなになるかを示している。

【 0 0 0 6 】

また図 4 (a) は AND ゲートの一方の入力が “ 0 ” から “ 1 ” に変化した時に、出力の状態が変化する、つまり回路を活性化するのに必要な他方の入力信号値を示しており、この場合は U 1 つまり “ x ” から “ 1 ” にすればよい。また、図 4 (b) に示すように AND ゲートの一方の入力が “ 0 ” から “ 1 ” に変化する場合は、出力の状態を変化させる (回路を活性化する) ために他方の入力は S 1、つまり “ 1 ” のままにすればよい。図 4 (c) に AND ゲート又は NAND ゲートおよび OR ゲート又は NOR ゲートの各一方の入力が “ 0 ” から “ 1 ” に立上った場合と、“ 1 ” から “ 0 ” に立下った場合における、これらゲートを活性化するのに必要な他方の入力信号値をそれぞれ示す。

【 0 0 0 7 】

いま図 5 (a) の集積回路は、入力端子 x_1 と x_2 の各入力が AND ゲート G_1 に供給され、入力端子 x_3 と x_4 の各入力が OR ゲート G_2 に供給され、入力端子 x_1 の入力がインバータゲート G_3 に供給され、インバータゲート G_3 の出力と入力端子 x_5 の入力が AND ゲート G_4 に供給され、ゲート G_1 と G_2 の各出力が NOR ゲート G_5 へ供給され、ゲート G_5 と G_4 の出力が OR ゲート G_6 へ供給されている。この回路において図中に太線で示すパス P、つまり入力端子 $x_3 - G_2 - G_5 - G_6 -$ 出力線におけるパス遅延故障に対するテストパターンを生成するには、まず、図 5 (a) 中に示すようにこの被試験パス P 上の各信号線に遷移信号を設定し、つぎに図 5 (b) に示すようにパス P 上の論理ゲートのパス P 上にない入力信号線 (サイド入力、side-inputs) に図 4 (c) に示した経路活性化表を用いてパス P の各ゲートを活性化する信号値をあたえ、最後に図 6 に示すように各信号線にあたえた信号値をもとに図 3 に示す含意表を用いて各入力信号線にあたえる信号値を決定する。図 5 および図 6 の例では、パス P のパス遅延故障を試験するテストパターン系列は $T = < v_1 , v_2 > = < “ S 0 X X U 1 S 0 U 0 ” > = < “ 0 x 0 0 x ” , “ 0 x 1 0 0 ” >$ と求められる。パス遅延故障に対するテストパターン生成法については、たとえば、C.J.Lin and S.M.Reddy, “ On Delay Fault Testing in Logic Circuits, ” Transactions on Computer-Aided Desing, CAD-6(5), pp.694-703, 1987 (文献 1) や K.-T.Cheng, A.Krstic, and H.-C.Chen, “ Generation of High Quality Tests for Robustly Untestable Path Delay Faults, ” Transactions on Computers, 45(12), pp.1379-1392, 1996 などに記載されている。

【 0 0 0 8 】

【 発明が解決しようとする課題 】

しかし、図 1 を参照して説明したテストパターン生成方法は、基本となる試験法が被試験回路の出力端子 (ピン) の電圧信号を観測する手法であるため、故障の影響を被試験回路の出力端子 (ピン) まで伝搬しなければならないという制約があり、テストパターン生成が困難であった。とくに、遅延時間の増分が被試験回路のクロック周期以下である微小なゲート遅延故障 (small gate delay fault) の試験には、故障の影響を観測するために、故障ゲートの出力信号を、もっとも遅延時間が大きい信号伝搬パスをとおして出力端子 (ピン) に伝搬しなければならない、故障を活性化する 2 つのテストパターン (テストベクトルペアまたはテストパターン系列と呼ぶ) を生成することは非常に困難である。

【 0 0 0 9 】

また、従来 of テストパターン生成方法は、ゲート遅延故障および断線故障の影響を出力信号線まで伝搬しなければならないため、テストパターン生成の含意操作において信号線の論理値に矛盾が生じる確率が高くなり、テストパターン生成において信号設定のやり直し

10

20

30

40

50

(バックトラック)回数が大きくなり、テストパターン生成に時間がかかるという問題がある。

このため、集積回路のゲート遅延故障やゲート遅延故障を引き起こす断線故障に対し、テストパターン系列を容易にかつ高速に生成できるテストパターン生成方法が必要となる。

【0010】

また図5及び図6を参照して説明したテストパターン生成方法は、被試験パスを活性化させるために被試験パス上のすべての論理ゲートのサイド入力(被試験パス上にないすべての入力信号線)に非制御入力値(non-controlling input value,各論理ゲートの出力を一意的に決定できない論理値、たとえば、ANDゲートやNANDゲートの非制御入力値は論理値“1”であり、ORゲートやNORゲートの非制御入力値は論理値“0”である)を設定するという制約と、パス遅延故障試験をロバストにおこなう(試験結果が出力電圧サンプリングのタイミングに依存しない)テストパターン系列を生成するために被試験パスおよびサイド入力にハザード(ひげ状の電圧パルス)を生じないという制約を満たさなければならない。この結果、テストパターン系列を生成することが非常に困難であった。さらに、テストパターン生成できないパス遅延故障の数が大きくなり、パス遅延故障の故障検出率が低いという問題がある。

10

【0011】

このため、集積回路のパス遅延故障に対し、テストパターン系列を容易にかつ効率的に生成できるテストパターン生成方法が必要となる。

この発明の目的は、回路の過渡現象を試験できる可観測性の高い過渡電源電流(I_{DDT} , transient power supply current)試験法を用いることによって、半導体集積回路内の遅延故障や断線故障に対するテストパターン系列を容易にかつ高速に生成できるテストパターン生成方法を提供することにある。

20

【0012】

この発明の他の目的は、回路の過渡現象を試験できる可観測性の高い過渡電源電流(I_{DDT} , transient power supply current)試験法を用いることによって、半導体集積回路内のパス遅延故障に対するテストパターン系列を容易に生成でき、効率的なテストパターン系列を作成できるテストパターン生成方法を提供することにある。

【0013】

【課題を解決するための手段】

第1発明の半導体集積回路をテストするテストパターン系列を生成する方法の一例においては、被試験半導体集積回路に対する故障リストを生成し、上記故障リストからテストパターン生成をおこなう対象故障を選択ステップで選択し、上記対象故障に対応する故障箇所を上記故障を活性化するための初期値をあたえる初期化テストパターンを求め、上記故障箇所が上記初期値に縮退する故障を仮定し、上記故障箇所を入力とする次段の論理ゲートの出力に上記縮退故障を伝搬する伝搬テストパターンを求め、上記求めた初期化テストパターンと伝搬テストパターンで構成されるテストパターン系列をテストパターンリストに登録し、上記故障リストに処理されていない故障が存在しなくなるまで上記選択ステップ以下の処理を繰り返す。

30

【0014】

上述におけるテストパターン系列の生成は、上記対象故障に対応する故障箇所を上記故障を活性化する、5値論理システムの信号値をあたえ、上記故障箇所を入力とする次段の論理ゲートの出力に上記信号値を伝搬するテストパターン系列を求めてもよく、その他の手法によってもよい。

40

更にそれぞれ電子計算機を備えるクライアントと複数のサーバとを用いて、クライアントで上記故障リストの生成を行い、その故障リストを複数のサブ故障リストに分割して複数のサーバに送信し、各サーバでクライアントから送信された上記サブ故障リストからテストパターン生成をおこなう対象故障を選択して、上記対象故障に対応する故障箇所を上記故障を活性化する信号を与え、その故障箇所を入力とする次段の論理ゲートに信号をテストパターン系列を求めてクライアントに返送し、サブ故障リストに処理されていない故障

50

が存在しなくなると処理終了をクライアントへ通知し、クライアントは複数のサーバから返送されたテストパターン系列をテストパターンリストに登録し、すべてのサーバからテストパターン生成処理が終了したという通知を受信して処理を終了する。

【0015】

このように分散処理によりテストパターン系列を高速に生成できる。

上述において登録するテストパターン系列を用いて過渡電源電流故障シミュレーションをおこない、上記テストパターン系列で検出可能な故障のリスト（被検出故障リスト）を作成し、その被検出故障リスト内の故障を上記故障リストから削除する。

このようにして故障リスト内の故障処理の完了を速くすることができる。

【0016】

クライアントと複数のサーバを用いる場合に、故障リストを分割して一度にサーバに送ってしまうことなく、処理の済んだサーバから逐次1つずつ送って処理させ、特にサーバから返送されたテストパターン系列、つまり登録するテストパターン系列を用いて過渡電源電流故障シミュレーションをおこない、上記テストパターン系列で検出可能な故障のリスト（被検出故障リスト）を作成して、上記被検出故障リスト内の故障を上記故障リストから削除することにより、全体としての処理速度を速めてもよい。あるいはサーバがクライアントの故障リストから故障を選択して処理し、かつ故障リストに処理していない故障があるかを調べるようにしてもよい。

【0017】

また、上記テストパターン系列を生成するステップは、ゲート遅延故障を単位としてテストパターン系列を生成することが望ましい。

また、上記テストパターン系列を生成するステップは、断線故障を単位としてテストパターン系列を生成することを特徴とすることが望ましい。

第2発明のテストパターン生成方法においては被試験半導体集積回路に対するパス遅延故障のリスト（故障リスト）を生成し、上記半導体集積回路にあたる2つ以上のテストパターンからなるテストパターン系列を生成し、このテストパターン系列に対し、被試験半導体集積回路について、遷移シミュレーションをおこない、回路内部の信号線に生じる遷移信号値列を計算し、上記遷移シミュレーションにより計算された各信号線の遷移信号値列を用いて上記テストパターン系列により過渡電源電流試験で検出可能な故障のリスト（被検出故障リスト）を生成し、上記被検出故障リスト内の故障を上記故障リストから削除し、上記テストパターン系列をテストパターン系列リストに登録し、上記故障リスト内のパス遅延故障が所定数以下になるまで、上記テストパターン系列の生成以下の処理を行う。

【0018】

この場合もクライアントと複数のサーバを用いて、クライアントでパス遅延故障のリスト（故障リスト）を生成し、テストパターン系列を生成して処理をおこなっていない1以上のサーバにそれぞれ1つずつテストパターン系列を送信し、各サーバでクライアントから送信されたテストパターン系列に対し遷移シミュレーションをおこない、回路内部の信号線に生じる遷移信号値列を計算し、その遷移信号値列を用いて上記テストパターン系列により過渡電源電流試験で検出可能な故障のリスト（被検出故障リスト）を生成し、上記被検出故障リスト内の故障を上記故障リストから削除し、上記故障リスト内のパス遅延故障数が所定値以下になるとクライアントが処理終了を各サーバに通知する。更にクライアントで予め十分な数のテストパターン系列のリストを生成し、これを複数のサブテストパターンに分別して、それぞれ各サーバに送信し、各サーバはそのサブテストパターン系列リストからテストパターン系列を選んで処理をするようにしてもよい。

[作用]

以下では、もっとも一般的な半導体集積回路であるCMOS集積回路を例に説明する。

CMOS論理ゲートの過渡電流

図7に、CMOSインバータの過渡応答(transient response)を示す。この過渡応答は、回路シミュレータで求めた。図7aは、過渡状態における入力電圧 V_{IN} に対する出力電

10

20

30

40

50

圧 V_{OUT} の応答と、電源から CMOS インバータに流れ込む電流 I_{DD} の応答である。この電流を過渡電流 (transient current) と呼ぶ。インバータの入力が “ 1 ” から “ 0 ” に遷移するとき (図 7 b)、入力電圧が n - MOS のしきい電圧より高く p - MOS のしきい電圧より低いあいだ、n - MOS と p - MOS が一瞬だけ同時にオンとなり、電源からグランドへ貫通電流 (short circuit current) I_S が流れる。このとき、インバータの出力信号線は “ 0 ” から “ 1 ” に遷移するため、貫通電流と同時にインバータの出力信号線に接続された寄生容量 (parasitic capacitance) C_{load} への充電 (capacitance charging) にともなう電流 I_C が電源端子 T_{VD} から流れる。したがって、インバータの入りに立ち下がり遷移が生じるとき (添字 “ f ” で記述する)、インバータに流れ込む過渡電流 I_{Gf} は、貫通電流 I_{Sf} と充電電流 (capacitance charging current) I_C の和であたえられる。

10

【 0 0 1 9 】

$$I_{Gf} = I_{Sf} + I_C \quad (1)$$

一方、入力が “ 0 ” から “ 1 ” に遷移するとき (出力が “ 1 ” から “ 0 ” に遷移するとき (添字 “ r ” で記述する) (図 7 c))、出力信号線に接続された寄生容量 C_{load} からの放電 (capacitance discharging) が起こり、放電電流 (capacitance discharging current) I_D が生じるが、電源端子 T_{VD} からインバータへ流れ込む電流 I_{Gr} は貫通電流 I_{Sr} だけである。このため、この電流ピークは、立ち下がり遷移時の過渡電流 I_{Gf} ピークよりわずかに小さくなる。

【 0 0 2 0 】

$$I_{Gr} = I_{Sr} \quad (2)$$

20

CMOS インバータの伝達特性は、図 8 a に示すように入力電圧 V_{IN} の変化に対し三角パルス状の電流 I_S を示す。このため、CMOS インバータの入力が立ち上がり遷移をもつとき、CMOS インバータを流れる貫通電流波形 I_{Sr} は、入力電圧 V_{IN} がランプ状に遷移すると仮定すると、図 8 b において “ I_S ” で示した三角パルスで近似できる。また、図 8 b に示した入力信号の立ち上がり遷移に対する CMOS インバータの貫通電流波形 I_{Sr} は、つぎのような近似式であたえられる。

【 0 0 2 1 】

【 数 1 】

$$I_{Sr} = \begin{cases} 0, & t \leq \frac{V_{THN}}{V_{DD}} t_r \\ \frac{V_{DD} \cdot I_{Smax}}{(V_{SP} - V_{THN}) \cdot t_r} t - \frac{V_{THN} \cdot I_{Smax}}{(V_{SP} - V_{THN})}, & \frac{V_{THN}}{V_{DD}} t_r < t \leq \frac{V_{SP}}{V_{DD}} t_r \\ \frac{V_{DD} \cdot I_{Smax}}{(V_{SP} - V_{DD} + V_{THP}) \cdot t_r} t - \frac{(V_{DD} - V_{THP}) \cdot I_{Smax}}{(V_{SP} - V_{DD} + V_{THP})}, & \frac{V_{SP}}{V_{DD}} t_r < t \leq \frac{V_{DD} - V_{THP}}{V_{DD}} t_r \\ 0, & t \geq \frac{V_{DD} - V_{THP}}{V_{DD}} t_r \end{cases} \quad (3)$$

30

40

【 0 0 2 2 】

ここで、 I_{Smax} は CMOS インバータに流れ込む過渡電流 (貫通電流) の最大値、 V_{DD} は電源電圧、 V_{THN} は n - MOS トランジスタのしきい電圧、 V_{THP} は p - MOS トランジスタのしきい電圧、 t_r は入力信号の立ち上がり遷移時間である。ただし、 V_{THP} は絶対値で示した。また、近似式を簡略化するため、入力電圧 V_{IN} の遷移開始時刻は 0 とし、遷移終了時刻 t_r で入力電圧は V_{DD} になるとした。

入力信号の立ち下がり遷移に対する CMOS インバータの貫通電流波形 I_{Sf} についても同様に、

【 0 0 2 3 】

【 数 2 】

50

$$I_{sf} = \begin{cases} 0, & t \leq \frac{V_{THP}}{V_{DD}} t_f \\ \frac{V_{DD} \cdot I_{Smax}}{(V_{DD} - V_{THP} - V_{SP}) \cdot t_f} t - \frac{V_{THP} \cdot I_{Smax}}{(V_{DD} - V_{THP} - V_{SP})}, & \frac{V_{THP}}{V_{DD}} t_f < t \leq \frac{V_{DD} - V_{SP}}{V_{DD}} t_f \\ \frac{V_{DD} \cdot I_{Smax}}{(V_{THN} - V_{SP}) \cdot t_f} t - \frac{(V_{DD} - V_{THN}) \cdot I_{Smax}}{(V_{THN} - V_{SP})}, & \frac{V_{DD} - V_{SP}}{V_{DD}} t_f < t \leq \frac{V_{DD} - V_{THN}}{V_{DD}} t_f \\ 0, & t \geq \frac{V_{DD} - V_{THN}}{V_{DD}} t_f \end{cases} \quad (4)$$

10

【0024】

と求めることができる。ここで、 t_f は入力信号の立ち下がり遷移時間である。念のためにこの場合の電源電流の立ち上がり開始時刻、最大値 I_{Smax} の時刻、立ち下り終了時刻を図8bに、括弧を付けて示した。

さらに、CMOSインバータの出力信号線の寄生容量 C_{load} への充電電流 I_C は、出力信号線の電圧変化を $v_{out}(t)$ とすると次式であたえられる。

【0025】

20

【数3】

$$I_C = C_{load} \frac{dv_{out}(t)}{dt} \quad (5)$$

【0026】

これらの式は、インバータ以外の論理ゲートに対しても同様に求めることができる。

論理ゲートに流れ込む過渡電流 I_G は、 I_C が I_S より十分小さいと仮定すると、図8bに I_S で示すような三角パルスで近似できる。実際、CMOSインバータの過渡電流波形 I_G は、図7aに示すような三角パルスを示す。また、 I_G が最大値 I_{Smax} となるのは、入力電圧 V_{IN} が論理ゲートのスイッチング電圧 V_{SP} になったときである。すなわち、図8bに示すように、 I_G がピークとなる時刻と論理ゲートの入力遷移の時刻が一致する。論理ゲートは遅延時間をもつため、論理ゲートの出力遷移の時刻は入力遷移の時刻よりわずかに遅れる。すなわち、 I_G がピークとなる時刻は論理ゲートの出力遷移の時刻よりわずかに先行する。この場合、過渡電流波形 I_G の立ち下がりエッジ（立ち下り部分）が、出力遷移の時刻に一致すると考えることができる。逆に、 I_C が I_S より大きいとき、過渡電流波形のピークは I_C のピークと一致する。充電電流 I_C は出力信号線の電圧遷移に関係するため、 I_G のピークは論理ゲートの出力の遷移時刻とほぼ一致する。

30

【0027】

また、論理ゲートの過渡電流波形 I_G のパルス幅は、式(1)、(2)、(3)、(4)より、入力電圧の遷移時間（たとえば立ち上がり遷移時間 t_r ）に比例する。ここで、入力電圧の遷移時間の変化に対して出力電圧の遷移時間の変化は十分小さいため、式(1)における I_C の項は無視した。

40

さらに、式(3)および式(4)より、貫通電流 I_{Sr} および I_{Sf} の時間積分値 Q_{Sr} および Q_{Sf} は、それぞれ、

【0028】

【数4】

$$Q_{Sr} = \int_{-\infty}^{\infty} I_{Sr} dt = \frac{I_{S \max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} t_r \quad (6)$$

$$Q_{Sf} = \int_{-\infty}^{\infty} I_{Sf} dt = \frac{I_{S \max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} t_f \quad (7)$$

【 0 0 2 9 】

であたえられる。したがって、スイッチングのとき論理ゲートを流れる貫通電流の積分値 Q_S は、

【 0 0 3 0 】

【 数 5 】

$$Q_S \equiv \int_{-\infty}^{\infty} I_S dt = \frac{I_{S \max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} t_T \propto t_T \quad (8)$$

【 0 0 3 1 】

であたえられる。ここで、 t_T は入力信号の遷移時間である。すなわち、論理ゲートに流れ込む貫通電流 I_S (I_{Sr} または I_{Sf}) の積分値 Q_S は、論理ゲートの入力遷移時間 t_T に比例する。また、 Q_S は、入力信号の遷移方向が立ち上がり遷移であるか、立ち下がり遷移であるかに関係ないことがわかる。

C M O S インバータの出力負荷容量 C_{load} への充電電流 I_C の積分値 Q_C は式 (5) より

【 0 0 3 2 】

【 数 6 】

$$\begin{aligned} Q_C &= \int_{-\infty}^{\infty} I_C dt = \int_{-\infty}^{\infty} C_{load} \frac{dv_{out}(t)}{dt} dt \\ &= C_{load} [v_{out}(t)]_{-\infty}^{\infty} = C_{load} (V_{DD} - 0) = C_{load} V_{DD} \end{aligned} \quad (9)$$

【 0 0 3 3 】

であたえられ、C M O S インバータの入力遷移時間 t_T には依存しない。

したがって、論理ゲートを流れる過渡電流 I_{Gf} および I_{Gr} の積分値 Q_{Gf} および Q_{Gr} は、式 (1) , (2) , (8) , (9) より、

【 0 0 3 4 】

【 数 7 】

$$Q_{Gf} = \int_{-\infty}^{\infty} (I_{Sf} + I_C) dt = \frac{I_{S \max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} t_T + C_{load} V_{DD} \propto t_T \quad (10)$$

$$Q_{Gr} = \int_{-\infty}^{\infty} I_{Sr} dt = \frac{I_{S \max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} t_T \propto t_T \quad (11)$$

【 0 0 3 5 】

とも定められる。すなわち、論理ゲートの過渡電流の積分値は論理ゲートの入力遷移時間に比例する。

C M O S 集積回路の過渡電源電流 (I_{DDT} , transient power supply current)

C M O S 集積回路の電源電流は、C M O S 集積回路に流れ込む電源電流であり、集積回路内の各論理ゲートを流れる電流の和で表される。

図 9 a に示す C M O S 集積回路は、図 7 に示したインバータを直列に 4 つ (G_1 , G_2 , G_3 , G_4) 組み合わせたものであり、各インバータを流れる過渡電流 (I_{G1} , I_{G2} , I

10

20

30

40

50

I_{G3}, I_{G4}) は通常一つの電源から供給される。このため、集積回路の過渡電源電流 I_{DDT} 応答は、図 9 b に示すように各論理ゲートを流れる過渡電流の和として表され次式で求まる。

【 0 0 3 6 】

$$I_{DDT} = \sum_{n=1}^N I_{Gn} \quad (12)$$

ここで、 N は入力されたテストパターン系列によりスイッチングする論理ゲートの数であり、図 9 a の例では $N = 4$ である。このため、集積回路内に論理ゲートの過渡電流を変化させる故障があり、その故障を活性化できるテストパターン系列が存在すれば、集積回路の I_{DDT} を観測することにより故障を検出することができる。

また、過渡電源電流 I_{DDT} の積分値 Q_{DDT} も、各論理ゲートを流れる電流の積分値 Q_{Gn} ($1 \leq n \leq N$) の和として表される。

【 0 0 3 7 】

【 数 8 】

$$Q_{DDT} = \int_{-\infty}^{\infty} I_{DDT} dt = \int_{-\infty}^{\infty} \left(\sum_{n=1}^N I_{Gn} \right) dt = \sum_{n=1}^N \int_{-\infty}^{\infty} I_{Gn} dt = \sum_{n=1}^N Q_{Gn} \quad (13)$$

【 0 0 3 8 】

図 9 の例では、過渡電源電流 I_{DDT} の積分値 Q_{DDT} は、各インパルスを流れる電流の積分値 ($Q_{G1}, Q_{G2}, Q_{G3}, Q_{G4}$) の和として表される。したがって、集積回路内に論理ゲートの過渡電流やその時間積分値を変化させる故障があり、その故障を活性化できるテストパターンがあれば、集積回路の過渡電源電流 I_{DDT} の積分値 Q_{DDT} を観測することにより故障を検出することができる。

ゲート遅延故障 (gate delay fault) の検出

ゲート遅延故障は、論理ゲートの入力から出力までの信号伝搬遅延が大きくなる故障である。論理ゲート出力の立ち上がり遷移が遅くなる故障を立上り遅れ (slow-to-rise) 故障と呼び、逆に立ち下がり遷移が遅くなる故障を立下り遅れ (slow-to-fall) 故障と呼ぶ。CMOS 集積回路内の論理ゲートにゲート遅延故障が生じると、論理ゲートの出力遷移時刻が遅れる。このため、この故障論理ゲートの出力を入力とする論理ゲートの過渡電流ピークのタイミングが遅くなる。したがって、あるテストパターン系列が故障論理ゲートを活性化し上記故障ゲートにより駆動される論理ゲートをスイッチングさせることができれば、CMOS 集積回路の I_{DDT} における電流ピークの遅れを観測することにより上記ゲート遅延故障を検出することができる。

断線故障 (open fault) の検出

つぎに、遅延故障を生じる断線故障について定義する。断線故障は、故意でない電氣的不連続であり、ある信号線を二つ以上の異なる信号線に分割する。断線故障には、金属の欠損や酸化膜による断線コンタクト、パターンング不良やエッチング不良などによる金属配線断線、マスク不良などによる拡散層あるいはポリシリコンにおける断線などが含まれる。また、断線故障は、図 10 a に示すように“論理故障”を生じる断線故障と、図 10 b に示すように“遅延故障”を生じる断線故障の二つのタイプに分類できる。論理故障を生じる断線故障は、断線の規模が大きく故障の両端の信号線に電圧をあたえても電流が流れないため、信号遷移に伴う寄生容量の充放電がおこなわれず論理がある一定の値に固定される論理故障を生じる。これに対し、遅延故障を生じる断線故障では、故障の両端の信号線に電圧をあたえると微小な電流が流れるが、その電流量は正常時より小さいため、信号遷移に伴う寄生容量の充放電が遅くなり、回路の遅延時間が増加する。遅延故障を生じる断線故障には、コンタクトなどの不良により信号線間の抵抗値が正常値より大きくなってしまったり、信号線に生じた不良などにより信号線の抵抗値が正常値より大きくなってしまったり、トンネル効果により断線故障を通してごく微量なリーク電流が流れる非常に小さな (亀裂の幅が 100 nm より小さい) 微小断線故障がある。微小断線故障を流れるトンネル電流については、例えば、C.L.Henderson, J.M.Soden, and C.F.Hawkin

10

20

30

40

50

s, "The Behavior and Testing Implications of CMOS IC Logic Gate Open Circuits," Proceedings of IEEE International Test Conference, pp.302-310, 1991.に記載されている。この明細書では、遅延故障を生じる断線故障を対象とし、これを単に断線故障と呼ぶ。

【0039】

断線故障は、故障をとおして小さな電流が流れるため、大きな抵抗値の抵抗素子 R_{open} でモデル化できる。図11aに、断線故障をもつCMOS集積回路の例を示す。このCMOS集積回路は、2つのインバータ G_1 , G_2 によって構成されており、インバータ G_1 と G_2 を接続する信号線Mに断線故障をもつ。その断線箇所は抵抗 R_{open} で接続されていると等価的に考えられる。インバータ G_1 がスイッチングし、信号線Mに信号遷移が生じるとき、断線故障の存在によりインバータ G_2 の入力寄生容量 C_{in} の充電または放電に要する時間（断線故障の抵抗値と寄生容量の積に比例する）が長くなるため、等価抵抗素子 R_{open} のインバータ G_2 側の信号線Mの信号遷移が遅くなる（図11b）。このため、入力に断線故障をもつインバータ G_2 の過渡電流波形は図12に示すように正常なインバータの過渡電流波形に比べ電流パルスの幅、および、過渡電流の時間積分値が大きくなる。したがって、上記CMOS集積回路の I_{DDT} または Q_{DDT} を観測することによりインバータ G_2 の入力段に存在する断線故障を検出することができる。インバータ G_1 の出力段における断線故障は、インバータ G_2 の入力段における断線故障と等価であり、同様に検出可能である。一般に、集積回路のすべての信号線における断線故障に対し、あるテストパターン系列で断線故障を活性化し上記断線故障により駆動される論理ゲートをスイッチングさせることができれば、集積回路の I_{DDT} または Q_{DDT} を観測することにより上記断線故障を検出することができる。

パス遅延故障

信号が伝搬する集積回路内の経路をパス (paths) と呼ぶ。パスの始点 (入力信号線) から終点 (出力信号線) までの遅延時間 (これをパス遅延時間 (path delay time) と呼ぶ) が既定値より大きく (小さく) なると、集積回路は異常な動作を示す。これをパス遅延故障 (path delay faults) と呼ぶ。つぎに、パス遅延故障について定義する。

【0040】

CMOS論理回路において、2つのテストパターン v_1 , v_2 をもつテストパターン系列 $T = \langle v_1, v_2 \rangle$ (パターン v_1 の次にパターン v_2 が続くことをあらわす) を用いてパス $P = \{ g_0, g_1, g_2, \dots, g_m \}$ を活性化するときを考える。ここで、 g_0 はパスPの入力信号線であり、 g_1, g_2, \dots, g_m はパスP上の論理ゲート G_1, G_2, \dots, G_m の出力信号線である。同時に、 g_0, g_1, \dots, g_{m-1} はパスP上の論理ゲート G_1, G_2, \dots, G_m の入力信号線でもある。各信号線 g_0, g_1, \dots, g_m の信号遷移の時刻 (電圧信号が $V_{DD}/2$ をよこぎる時刻) をそれぞれ t_0, t_1, \dots, t_m とすると、パスP上の各論理ゲート G_1, G_2, \dots, G_m のゲート遅延時間 t_{gdi} , $1 \leq i \leq m$ は、それぞれ

$$t_{gdi} = t_i - t_{i-1} \quad (14)$$

であたえられる。したがって、パスPのパス遅延時間 t_{pd} は、ゲート遅延時間 t_{gdi} の和として、

【0041】

【数9】

$$t_{pd} = \sum_{i=1}^m t_{gdi} = \tau_m - \tau_0 \quad (15)$$

【0042】

で求めることができる。しかし、実際のゲート遅延時間 t_{gdi} は、故障の影響により変動する。

$$t_{gdi} = t_{gdi,typ} + \delta_i, \quad 1 \leq i \leq m \quad (16)$$

ここで、 $t_{gdi,typ}$ は論理ゲート G_i のゲート遅延時間の典型値であり、 δ_i はゲート遅延

時間の変動成分である。例えば、断線故障は、故障をもつ論理ゲートのゲート遅延時間のみを増加させ、ほかの論理ゲートの遅延時間を増加させない。また、プロセスパラメータの異常は、すべての論理ゲートの遅延時間を増加させる。ゲート遅延時間の変動にともなって、パス遅延時間 t_{pd} も同様に変動する。

【 0 0 4 3 】

【 数 1 0 】

$$t_{pd} = t_{pd,typ} + \Delta = \sum_{i=1}^m (t_{gdi,typ} + \delta_i) \quad (17)$$

10

【 0 0 4 4 】

ここで、 $t_{pd,typ}$ はパス P のパス遅延時間の典型値であり、 Δ はパス遅延時間の変動成分である。

図 1 3 に遅延故障試験方法の基本原則を模式的に示す。図 1 3 a の被試験回路 (CUT) が正常に動作するためには、入力ラッチに生成された信号遷移がパス P をとおって出力ラッチまで既定の時間内に伝搬しなければならない。したがって、図 1 3 b に示す入力 V_{IN} 及び出力 V_{OUT} の関係と、システムクロック CLK との関係からパス P のパス遅延時間 t_{pd} は以下の条件を満たさなければならない。

【 0 0 4 5 】

$$t_{pd} + T_{SU} < T_{CLK} - T_{SKW} \quad (18)$$

20

ここで、 T_{SU} は信号のセットアップ時間、 T_{CLK} はシステムクロックの周期、 T_{SKW} はシステムクロックのクロックスキュー (ジッタでありエッジが \pm に変動する) である。変形すると、

$$t_{pd} < T_{CLK} - T_{SKW} - T_{SU} - T \quad (19)$$

となる。すなわち、パス P のパス遅延時刻 t_{pd} は、クロック周期からセットアップ時間やクロックスキューなどのマージンをひいた時間 (これを T とする) より小さくしなければならない。もし、 t_{pd} が T より大きければ、パス P に沿った信号伝搬がシステムクロックに間に合わず、回路は正しい動作をしない。この状態を遅延故障と定義する。すなわち、 t_{pd} がある既定の時間 T より大きいとき、パス P は遅延故障をもつと定義する。ここで、 T は、許容できる遅延時間の上限値である。

30

パス遅延故障の検出 (過渡電源電流のパルス幅を利用)

論理ゲートの過渡電流波形のピーク (または、立ち下がリエッジ) がその論理ゲートの出力の遷移時刻に対応することから、CMOS 集積回路の過渡電源電流波形の最終ピーク (最終の立ち下がリエッジ) が CMOS 集積回路において最後にスイッチングする論理ゲートの出力遷移時刻に一致する。したがって、CMOS 集積回路の過渡電源電流波形の最終ピーク (最終の立ち下がリエッジ) を検出し入力遷移の時刻と比較することにより、回路のパス遅延時間を求めることができる。ここで、過渡電源電流の最終の立ち下がリエッジの時刻は、たとえば、過渡電源電流がある既定の電流値となる時刻の最大値として求めることができる。この電流値は、被試験パス上の最後の論理ゲートの出力が電源電圧の半分の値となるときの電源電流の値であり、被試験回路に対する回路シミュレーションや実デ

40

【 0 0 4 6 】

つぎに、上で述べた過渡電源電流のパルス幅を利用してパス遅延故障を検出する方法について述べる。上記方法は、被試験回路の電源電流波形のパルス幅を測定し、既定の時間と比較する方法である。上記方法の基本原則を図 1 4 に示す。CMOS 論理回路において、2 つのテストパターン v_1, v_2 を持つテストパターン系列 $T = \langle v_1, v_2 \rangle$ により複数のパス P_1, P_2, \dots, P_n が活性化されるとする。パス P_i 上の入力から数えて j 番目の論理ゲートがスイッチングする時刻を t_{ij} とすると、各パス P_1, \dots, P_n により論理ゲ

50

トの数は異なり、パス P_1, P_2, \dots, P_n においてもっとも遅くスイッチングする論理ゲート G_{final} の出力遷移の時刻 t_{max} は、

【 0 0 4 7 】

【 数 1 1 】

$$t_{max} = \max_{i,j} \{t_{ij}\}, \quad 1 \leq i \leq n, 1 \leq j \quad (20)$$

【 0 0 4 8 】

であたえられる。したがって、パス P_1, P_2, \dots, P_n におけるパス遅延時間の最大値 $t_{pd,max}$ は、 t_{max} と入力遷移の時刻 t_0 のあいだの時間間隔としてつぎのように求めることができる。

$$t_{pd,max} = t_{max} - t_0 \quad (21)$$

一方、CMOS論理回路の過渡電源電流波形のパルス幅 t_{PW} を、回路入力信号遷移の時刻 t_0 と過渡電源電流波形の最終ピーク（立ち下がりエッジ）の時刻 t_{IDD} のあいだの時間間隔と定義する。

【 0 0 4 9 】

$$t_{PW} = t_{IDD} - t_0 \quad (22)$$

前に述べたように、過渡電源電流波形の最終ピークの時刻 t_{IDD} は、最後にスイッチングする論理ゲート G_{final} の出力遷移時刻 t_{max} に一致するか t_{max} より先行することから、過渡電源電流波形のパルス幅 t_{PW} は、テストパターン T により活性化されるパス P の遅延時間 $t_{pd,max}$ に対応する。

$$t_{PW} = t_{IDD} - t_0 \leq t_{max} - t_0 = t_{pd,max} \quad (23)$$

もし、 t_{PW} が許容できる遅延時間の上限値 T より大きいならば、

$$T < t_{PW} \leq t_{pd,max} \quad (24)$$

となる。もっとも大きな遅延時間 $t_{pd,max}$ をもつパスにおいて、信号の伝搬がシステムロックに間に合わない。すなわち、回路には遅延故障が存在する。したがって、T より大きい t_{PW} は活性化されたパスのいずれかに遅延故障があることを示し、T より小さい t_{PW} は活性化されたパスのいずれにも遅延故障が存在しないことを示す。

【 0 0 5 0 】

遅延故障なし、 $t_{PW} \leq T$

遅延故障あり、 $t_{PW} > T$ (25)

以上のように、過渡電源電流波形のパルス幅 t_{PW} を既定の時間 T と比較することにより、回路のパス遅延故障をテストすることができる。

パス遅延故障の検出（過渡電源電流の瞬時値を利用）

また、図7に示すように、論理ゲートの過渡電源電流はピーク値以降単調に減少することから、図9に示すCMOS集積回路の電源電流は、同集積回路において最後にスイッチングする論理ゲートの出力遷移時刻以降単調に減少する。つまり、故障のないCMOS集積回路において、最後にスイッチングする論理ゲートの出力遷移時刻を t_{max} とし、時刻 t_{max} における過渡電源電流の瞬時値を I_{max} とすると、 t_{max} 以後CMOS集積回路の過渡電源電流が I_{max} より大きくなることはない。

【 0 0 5 1 】

この原理をもちい、ある既定時刻におけるCMOS集積回路の過渡電源電流の瞬時値を測定することにより、被試験回路における遅延故障を検出することができる。ここで、故障検出の判断基準となる電流値 I_{th} は、被試験パス上の最後の論理ゲートの出力が電源電圧の半分の値となるとき電源電流の値であり、被試験回路に対する回路シミュレーションや実デバイスをもちいた統計データなどから求めることができる。

【 0 0 5 2 】

つぎに、上で述べた過渡電源電流の瞬時値を利用してパス遅延故障を検出する方法について述べる。上記方法は、規定の時刻における被試験回路の過渡電源電流の瞬時値を測定し

、遅延故障のないゴールデン回路の過渡電源電流値と比較する方法である。上記方法の基本原理を図15に示す。

CMOS論理回路において、テストパターン系列 $T = \langle V_1, V_2 \rangle$ により複数のパス P_1, P_2, \dots, P_n が活性化されるとする。パス P_i 上の入力から数えて j 番目の論理ゲートがスイッチングする時刻を t_{ij} とすると、パス P_1, P_2, \dots, P_n においてもっとも遅くスイッチングする論理ゲート G_{final} の出力遷移の時刻 t_{max} は、

【0053】

【数12】

$$t_{max} = \max_{i,j} \{t_{ij}\}, \quad 1 \leq i \leq n, 1 \leq j \tag{26}$$

10

【0054】

であたえられる。したがって、パス P_1, P_2, \dots, P_n におけるパス遅延時間の最大値 $t_{pd,max}$ は、 t_{max} と入力遷移の時刻 t_0 の間の時間間隔としてつぎのように求めることができる。

$$t_{pd,max} = t_{max} - t_0 \tag{27}$$

前にも述べたように、論理ゲートの出力遷移の時刻が論理ゲートの過渡電源電流のピークまたは立ち下がり時刻に一致するので、 t_{max} は回路の過渡電源電流波形 I_{DDT} の最終ピークまたは立ち下がりエッジの時刻 t_{DD} に対応する。論理ゲートの電源電流 I_G は三角波で近似でき、 G_{final} は最後にスイッチングするゲートであるから、 t_{max} 以降に電源電流のピークをもつ論理回路は存在しない。したがって、 $t \geq t_{max}$ なる時刻 t において電源電流波形関数 $i_{DDT}(t)$ は、単調減少関数である。すなわち、電源電流波形の時間関数を $i_{DDT}(t)$ とし、時刻 t_{max} における電源電流の瞬時値を

$$I = i_{DDT}(t_{max}) \tag{28}$$

とすると、 $t \geq t_{max}$ なる t において、

$$i_{DDT}(t) \leq i_{DDT}(t_{max}) = I, \quad t \geq t_{max} \tag{29}$$

が成立する。回路が正常に動作するためには、 $t_{pd,max}$ が遅延時間の上限値 T ($= T_{CLK} - T_{SKEW} - T_{SU}$) より小さくなければならない。

【0055】

$$t_{pd,max} = t_{max} - t_0 < T \tag{30}$$

したがって、回路に故障がないとき、 $t = T + t_0 > t_{max}$ なる時刻 t において、式(29)より

$$i_{DDT}(T + t_0) \leq I \tag{31}$$

が成立する。もし、 $T + t_0$ における i_{DDT} の瞬時値が I より大きいならば、つまり、

$$i_{DDT}(T + t_0) > I = i_{DDT}(t_{max}) \tag{32}$$

ならば、式(29)より $T + t_0$ が t_{max} より大きいことはありえないので、

$$t_{max} > T + t_0 \tag{33}$$

$$t_{pd,max} = t_{max} - t_0 > T \tag{34}$$

40

となる。もっとも大きな遅延時間 $t_{pd,max}$ をもつパスにおいて、信号の伝搬がシステムクロックに間に合わない。すなわち、回路にはパス遅延故障が存在する。したがって、時刻 $T + t_0$ における電源電流値 $i_{DDT}(T + t_0)$ が I より大きいということは、活性化されたパスのいずれかにパス遅延故障があることを示す。逆に、 $i_{DDT}(T + t_0)$ が I より小さいということは、活性化されたパスのいずれにもパス遅延故障が存在しないことを示す。

【0056】

遅延故障なし、 $i_{DDT}(T + t_0) \leq I$

遅延故障あり、 $i_{DDT}(T + t_0) > I$

$$\tag{35}$$

以上のように、ある既定の時刻における i_{DDT} の瞬時値を故障のない回路の i_{DDT} レベル

50

と比較することにより、回路の遅延故障を検出することができる。

パス遅延故障の検出（過渡電源電流の時間積分値を利用）

さらに、過渡電源電流 I_{DDT} の積分値 Q_{DDT} を用いることにより、断線故障を原因とするパス遅延故障を検出することができる。つぎに、過渡電源電流の時間積分値を利用してパス遅延故障を検出する方法について述べる。上記方法は、被試験回路の過渡電源電流の積分値を測定し、所定の値と比較することによってパス遅延故障を評価する方法である。

【0057】

過渡電源電流 I_{DDT} の積分値 Q_{DDT} は、各論理ゲートを流れる電流の積分値 Q_{Gn} ($1 \leq n \leq N$) の和として表される。

【0058】

【数13】

$$Q_{DDT} = \int_{-\infty}^{\infty} I_{DDT} dt = \int_{-\infty}^{\infty} \left(\sum_{n=1}^N I_{Gn} \right) dt = \sum_{n=1}^N \int_{-\infty}^{\infty} I_{Gn} dt = \sum_{n=1}^N Q_{Gn} \quad (13)$$

【0059】

各論理ゲートを流れる電流の積分値 Q_{Gn} ($1 \leq n \leq N$) は、式(10)または式(11)に示すように、各論理ゲートの入力遷移時間 t_{Tn} ($1 \leq n \leq N$) にそれぞれ比例することから、 Q_{DDT} は t_{Tn} ($1 \leq n \leq N$) の線形多項式であたえられる。例えば、図9の例では、 Q_{DDT} は各インバータの入力遷移時間 ($t_{T1}, t_{T2}, t_{T3}, t_{T4}$) の線形多項式であたえられる。

【0060】

【数14】

$$Q_{DDT} = \sum_{n=1}^N Q_{Gn} = \sum_{n=1}^N Q_{Sn} + \sum_{n=1}^N Q_{Cn} = \sum_{n=1}^N a_n t_{Tn} + b \quad (36)$$

【0061】

式(36)において、 a_n は論理ゲート G_n の貫通電流の積分値 Q_{Sn} と論理ゲート G_n の入力遷移時間 t_{Tn} の間の比例係数であり、 b は各論理ゲートに流れ込む充電電流 Q_{Cn} の和であらわされる定数項である。

断線故障は、故障をとおして小さな電流が流れるため、大きな抵抗 R_{open} でモデル化できる。図16aに、入力に断線故障をもつCMOSインバータの例を示す。入力信号線Aに図16bに示す信号遷移が生じるとき、断線故障により、断線箇所に後続する信号線Aの信号遷移が図16cに示すように遅くなる。このとき、信号線Aの信号遷移時間 t_T は、断線故障の抵抗を R_{open} 、インバータの入力における寄生容量を C_{in} とすると、

【0062】

【数15】

$$t_T \approx t_{T,typ} + 2.2R_{open}C_{in} \quad (37)$$

【0063】

であたえられる。ここで、 $t_{T,typ}$ は故障がない場合の入力信号の遷移時間の典型値であり、遷移時間 t_T は電圧値が $0.1V_{DD}$ から $0.9V_{DD}$ まで立ち上がる（または、電圧値が $0.9V_{DD}$ から $0.1V_{DD}$ まで立ち下がる）に要する時間としてもとめた。 $2.2R_{open}C_{in}$ は C_{in} が $0.1V_{DD}$ から $0.9V_{DD}$ になるまでの時間であり $\log_e(0.9V_{DD}/0.1V_{DD}) \times R_{open}C_{in}$ により求めた値である。すなわち、インバータの入力信号の遷移時間の増分は、断線故障の抵抗値 R_{open} に比例する。したがって、被試験パス上の k 番目のインバータの入力に断線故障があるとき、CMOS集積回路の電源電流の積分値 Q_{DDT} は、式(36)、(37)より、式(38)が求まり、断線故障の抵抗値 R_{open} に応じ

10

20

30

40

50

て線形に変化し、その増分は断線故障の抵抗値 R_{open} に比例する。

【 0 0 6 4 】

【数 1 6】

$$Q_{DDT} = \sum_{n=1}^N a_n t_{Tn} + b = \left(\sum_{n=1}^N a_n t_{Tn,typ} + b \right) + 2.2 a_k C_{in} R_{open} \quad (38)$$

$$= Q_{DDT,typ} + 2.2 a_k C_{in} R_{open} \propto R_{open}$$

【 0 0 6 5 】

10

ここで、 $Q_{DDT,typ}$ は故障がない場合の電源電流の積分値の典型値とする。式(38)の右辺第2項の $2.2 a_k C_{in} R_{open}$ は k 番目のインバータの入力断線故障にもとづく加算量である。この式(38)は、図17に示す R_{open} に対する Q_{DDT} の変化のシミュレーション結果とも一致する。図17は、図9に示した回路において、インバータIN2の入力信号線に断線故障があるときの、断線故障の抵抗値 R_{open} に対する Q_{DDT} の変化をプロットしたものである。

【 0 0 6 6 】

また論理ゲートのゲート遅延時間 t_{gd} は、入力信号の遷移時間 t_T に比例する。(Neil H. E. Weele 著者 “Principles of CMOS VLSI Design-A Systems Perspective” Second Edition. Addison-Wesley Publishing Company. 1999年発行の216～217頁の式4.52, 式4.53による)

20

【 0 0 6 7 】

【数 1 7】

$$t_{gd} = t_{gd,step} + \frac{1}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) t_T \quad (39)$$

【 0 0 6 8 】

ここで、 $t_{gd,step}$ は遷移時間0のステップ入力に対する故障のないインバータの遅延時間である。また、 V_{TH} は p -MOS または n -MOS のしきい値電圧であり、入力の立ち上がり遷移に対しては $V_{TH} = V_{THN}$ 、入力の立ち下がり遷移に対しては $V_{TH} = V_{THP}$ であたえられる。したがって、入力信号線上に抵抗 R_{open} でモデル化できる断線故障をもつ論理ゲートのゲート遅延時間 t_{gd} は、論理ゲートの入力遷移時間が式(37)であたえられることから、式(39)に式(37)を代入して、

30

【 0 0 6 9 】

【数 1 8】

$$\begin{aligned}
t_{gd} &= t_{gd,step} + \frac{t_T}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) \\
&= t_{gd,step} + \frac{t_{T,typ} + 2.2R_{open} C_{in}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) \\
&= t_{gd,step} + \frac{t_{T,typ}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) + \frac{2.2C_{in}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) R_{open} \\
&= t_{gd,typ} + \frac{2.2C_{in}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) R_{open} \propto R_{open}
\end{aligned} \tag{40}$$

10

【 0 0 7 0 】

と求めることができる。ここで、 $t_{gd,typ}$ は故障のない論理ゲートのゲート遅延時間の典型値である。すなわち、断線故障をもつ論理ゲートのゲート遅延時間 t_{gd} は故障の抵抗値 R_{open} によって変化し、ゲート遅延時間の増分は故障の抵抗値 R_{open} に比例する。ゆえに、被試験パス上のいずれかの論理ゲートの入力に断線故障があるとき、被試験パスのパス遅延時間 t_{pd} も R_{open} に比例する。このことを式で示すと、式(17)に式(40)を代入して式(41)が得られることから理解される。

20

【 0 0 7 1 】

【数19】

$$\begin{aligned}
t_{pd} &= \sum_{i=1}^m t_{gdi} = \sum_{i=1}^m (t_{gdi,typ} + \delta_i) \\
&= \sum_{i=1}^m t_{gdi,typ} + \frac{2.2C_{ink}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) R_{open} \\
&= t_{pd,typ} + \frac{2.2C_{ink}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) R_{open} \propto R_{open}
\end{aligned} \tag{41}$$

30

【 0 0 7 2 】

これは、図18に示す R_{open} に対する t_{pd} の変化のシミュレーション結果とも一致する。図18は、図9に示した回路において、インバータIN2の入力信号線に断線故障があるときの、断線故障の抵抗値 R_{open} に対する t_{pd} の変化をプロットしたものである。パスP上のある論理ゲート G_k の入力に断線故障が存在するとき、 G_k の貫通電流の積分値 Q_{Sk} は、式(8)および式(37)より、

【 0 0 7 3 】

【数20】

40

$$\begin{aligned}
Q_{Sk} &= \frac{I_{S\max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} t_{Tk} \\
&= \frac{I_{S\max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} (t_{Tk,typ} + 2.2R_{open} C_{ink}) \\
&= \frac{I_{S\max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} \cdot t_{Tk,typ} + \frac{I_{S\max} (V_{DD} - V_{THN} - V_{THP})}{2V_{DD}} \cdot 2.2R_{open} C_{ink} \\
&= Q_{Sk,typ} + \frac{2.2I_{S\max} (V_{DD} - V_{THN} - V_{THP}) C_{ink}}{2V_{DD}} \cdot R_{open}
\end{aligned} \tag{10}$$

【0074】

ともとめられる。したがって、集積回路の過渡電源電流の積分値 Q_{DDT} は、式(36)より、

【0075】

【数21】

$$\begin{aligned}
Q_{DDT} &= \sum_{n=1}^N Q_{Gn} \\
&= \sum_{n=1}^N Q_{Sn} + \sum_{n=1}^N Q_{Cn} \\
&= \sum_{n=k}^N Q_{Sn,typ} + Q_{Sk,typ} + \frac{2.2I_{S\max} (V_{DD} - V_{THN} - V_{THP}) C_{ink}}{2V_{DD}} \cdot R_{open} + \sum_{n=1}^N Q_{Cn} \\
&= \sum_{n=1}^N Q_{Sn,typ} + \sum_{n=1}^N Q_{Cn} + \frac{2.2I_{S\max} (V_{DD} - V_{THN} - V_{THP}) C_{ink}}{2V_{DD}} \cdot R_{open} \\
&= Q_{DDT,typ} + \frac{2.2I_{S\max} (V_{DD} - V_{THN} - V_{THP}) C_{ink}}{2V_{DD}} \cdot R_{open}
\end{aligned} \tag{42}$$

【0076】

となり、回路の過渡電源電流の積分値 Q_{DDT} も、断線故障の抵抗値 R_{open} に比例する。したがって、式(41)および式(42)より、断線故障をもつパスPの遅延時間 t_{pd} は、CMOS集積回路の過渡電源電流の積分値 Q_{DDT} に対して線形に変化する。これは、図19に示す Q_{DDT} に対する t_{pd} の変化のシミュレーション結果とも一致する。図19は、図9に示した回路において、インバータIN2の入力信号線に断線故障があるときの、過渡電源電流の積分値 Q_{DDT} に対する t_{pd} の変化をプロットしたものである。

式(42)より求めた R_{open} を式(41)に代入すると式(43)が得られる。

【0077】

【数22】

$$\begin{aligned}
 t_{pd} &= t_{pd,typ} + \frac{2.2C_{ink}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) R_{open} \\
 &= t_{pd,typ} + \frac{2.2C_{ink}}{6} \left(1 - 2 \frac{V_{TH}}{V_{DD}} \right) \cdot \frac{(Q_{DDT} - Q_{DDT,typ}) \cdot 2V_{DD}}{2.2I_{Smax} (V_{DD} - V_{THN} - V_{THP}) C_{ink}} \quad (43) \\
 &= t_{pd,typ} + \frac{V_{DD} - 2V_{TH}}{3I_{Smax} (V_{DD} - V_{THN} - V_{THP})} (Q_{DDT} - Q_{DDT,typ})
 \end{aligned}$$

10

【0078】

パス遅延時間 t_{pd} が許容できる遅延時間の上限値 T となるときの過渡電源電流の積分値を Q_{max} とすると、式(43)において $t_{pd} = T$, $Q_{DDT} = Q_{max}$ とおいて Q_{max} を求めると式(44)となる。

【0079】

【数23】

$$Q_{max} = Q_{DDT,typ} + \frac{3I_{Smax} (V_{DD} - V_{THN} - V_{THP})}{V_{DD} - 2V_{TH}} (T' - t_{pd,typ}) \quad (44)$$

20

【0080】

この Q_{max} は、パス遅延故障のないCMOS集積回路の過渡電源電流の積分値 Q_{DDT} の上限値である。すなわち、 Q_{DDT} が Q_{max} より小さいときCMOS集積回路には遅延故障が存在せず、 Q_{DDT} が Q_{max} より大きいときCMOS集積回路に断線故障によるパス遅延故障が存在すると判断することができる。

遅延故障なし, $Q_{DDT} \leq Q_{max}$

遅延故障あり, $Q_{DDT} > Q_{max}$ (45)

以上のように、過渡電源電流の積分値 Q_{DDT} を既定値 Q_{max} と比較することにより、回路のパス遅延故障をテストすることができる。ここで、既定値 Q_{max} は、回路シミュレーションや統計データから式(44)を用いて求めることができる。

30

【0081】

過渡電源電流は、集積回路の電源ピンを流れる過渡電流であり、電圧信号より高い可観測性が保証されている。このため、過渡電源電流をもちいた遅延故障試験方法は、電圧信号をもちいた遅延故障試験方法より、高い遅延故障の可観測性を保証することでできる。例えば、電圧信号をもちいた遅延故障試験方法は、電圧信号が集積回路の出力信号線まで伝搬しなければ遅延故障を検出できないのに対し、過渡電源電流信号をもちいた遅延故障試験方法は、たとえ電圧信号が集積回路の出力信号線まで伝搬しなくても、電圧信号が伝搬したパスの遅延時間に対応するパルス幅をもつ過渡電源電流信号が観測可能であるので、遅延故障や断線故障を検出することができる。また、過渡電源電流信号をもちいた遅延故障試験方法は、電圧信号を集積回路の出力信号線まで伝搬させる必要がないので、電圧信号を集積回路の出力信号線まで伝搬させる必要がある電圧信号をもちいた従来の遅延故障試験方法に比べ、テストパターン生成の制約が少ない。このため、テストパターン生成を容易にすることができる。

40

テストパターン生成方法(ゲート遅延故障)

つぎに、単一のゲート遅延故障にたいする第1発明のテストパターン生成方法について説明する。図20に被試験CMOS集積回路の一例を示す。被試験集積回路は図1に示した回路と同一であって5つの入力端子 x_1, x_2, x_3, x_4, x_5 、1つの出力端子 z_1 、5つの内部信号ノード n_1, n_2, n_3, n_4, n_5 、6つの論理ゲート $G_1, G_2, G_3, G_4, G_5, G_6$ をもつ。 n_1 と n_2 はゲート G_1 及び G_2 とゲート G_3 との各接

50

続点、 n_3 はゲート G_3 とゲート G_4 及び G_5 との接続点、 n_4 及び n_5 はゲート G_4 及び G_5 とゲート G_6 との各接続点である。図に示していないがゲート $G_1 \sim G_6$ の接続端子は共通の電源に接続されている。論理ゲート G_3 における立ち上がり遷移が遅くなるゲート遅延故障 (slow-rise) にたいするテストパターン生成を考える。

【0082】

はじめに、図20(a)中に示すように故障ゲート G_3 の出力にslow-rise故障の初期値“0”をあたえるテストパターン $v_1 = "xx0xx"$ を含意操作 (implication) によりもとめる。ここで、含意操作とは、集積回路内の信号線の論理値に基づき、その信号線に接続される論理ゲートの入出力信号線に一意的に選択される論理値を順次設定していく操作であり、回路の入力側(後方操作)および出力側(前方操作)の両方についておこなわれる。また、信号“0”、“1”、“x”は、それぞれLowレベル信号、Highレベル信号、Don't Care信号(LowレベルまたはHighレベルのどちらでもかまわない信号)を意味する。たとえば、NANDゲート G_3 の出力に信号値“0”を設定するには、ゲート G_3 のすべての入力信号値を“1”とする必要がある。すなわち、論理ゲート G_1 および G_2 の出力信号値は、上記の含意操作により“1”ともとめることができる。同様に、NANDゲート G_1 および G_2 の出力信号値を“1”とするには、NANDゲート G_1 および G_2 のそれぞれにたいし、そのNANDゲートの二つの入力のどちらかを“0”とすればよいので、入力端子 x_3 の信号値を含意操作により“0”をもとめることができる。このとき、他の入力端子の値は“0”、“1”のどちらでもかまわない。このようにして、故障ゲート G_3 の出力に初期値“0”をあたえるテストパターン v_1 を含意操作により“xx0xx”ともとめることができる。含意操作については、たとえば、藤原秀雄、コンピュータの設計とテスト、工学図書株式会社、1990の第8章に記載されている。

【0083】

つぎに、故障ゲート G_3 の出力が設定した初期値“0”に固定してしまう縮退故障を仮定し、この縮退故障を論理ゲート G_4 の出力に伝搬するためのテストパターン $v_2 = "0x11x"$ を含意操作によりもとめる(図20(b))。以上により、ゲート G_3 におけるslow-rise故障を過渡電源電流試験法で検出するテストパターン系列は $T = \langle v_1, v_2 \rangle = \langle "xx0xx", "0x11x" \rangle$ ともとめることができる。

【0084】

あるいは、故障ゲート G_3 の出力にslow-rise故障に対応する信号“U1”をあたえ、上記slow-rise故障の影響を論理ゲート G_4 の出力に伝搬するテストパターン系列 $T' = "S0 S0 U1 S1 XX"$ を含意操作 (implication) によりもとめる(図20(c))。ここで、信号“S0”、“S1”、“U0”、“U1”、“XX”は、C.J.Linらによって開発された5値論理システムの信号値であり、図2に示したようにそれぞれ常時Lowレベル信号($\langle "0", "0" \rangle$)、常時Highレベル信号($\langle "0", "0" \rangle$)、最終値がLowレベルとなる信号($\langle "X", "0" \rangle$)、最終値がHighレベルとなる信号($\langle "X", "1" \rangle$)、Don't Care信号($\langle "X", "X" \rangle$)を意味する。また、5値論理システムの含意操作とは、集積回路内の信号線の信号値に基づき、その信号線に接続される論理ゲートの入出力信号線に一意的に選択される信号値を順次設定していく操作である。

【0085】

たとえば、NANDゲート G_3 の出力に信号値“U1”を設定するには、ゲート G_3 の片方の入力信号値を“U0”とし、もう片方の入力信号値を“S1”とする必要がある。ここでは、論理ゲート G_1 の出力信号値を“S1”、ゲート G_2 の出力信号値を“U1”と仮定する。つぎに、NANDゲート G_1 の出力信号値を“S1”とするには、ゲート G_1 の入力の何れかを“S0”と設定すればよいので、ここでは入力端子 x_2 に信号“S0”をあたえる。一方、NANDゲート G_2 の出力信号値を“U0”とするには、ゲート G_2 の入力の片方を“U0”、もう片方を“S1”とすればよいので、入力端子 x_3 の信号値を“U0”とし、入力端子 x_4 の信号値を“S1”とする。同様に、故障ゲート G_3 の影

10

20

30

40

50

響“U1”をNORゲートG₄の出力に伝搬するには、G₄のもう一方の入力、すなわち、入力信号線を“S0”とすればよい。したがって、故障ゲートG₃の出力にslow-to-rise故障に対応する信号“U1”をあたえ、上記slow-to-rise故障の影響を論理ゲートG₄の出力に伝搬するテストパターン系列T'は、上記の含意操作により“S0 S0 U1 S1 XX”ともとめることができる。5値論理システムについては、先に述べたようにたとえば、文献1に記載されている。以上により、ゲートG₃におけるslow-to-rise故障を過渡電源電流試験法で検出するテストパターン系列はT' = “S0 S0 U1 S1 XX” = < v1, v2 > = < “0001x”, “0011x” > ともとめることができる。

【0086】

論理ゲートG₃がゲート遅延故障をもつとき、論理ゲートG₃の出力遷移時刻が遅くなる。この故障の影響をテストパターン系列TまたはT'により次段の論理ゲートG₄(の出力)に伝搬すると、論理ゲートG₄の過渡電源電流波形が正常時と異なるため、被試験集積回路の過渡電源電流は異常を示す。したがって、テストパターン系列TまたはT'をあたえ被試験集積回路の過渡電源電流に異常が生じるか否かを観測することによって、論理ゲートG₃におけるslow-to-rise故障が存在するか否かを判別することができる。

【0087】

以上により、被試験集積回路内のあるゲート遅延故障にたいするテストパターン系列を生成することができる。さらに、この発明のテストパターン生成方法は、ゲート遅延故障単位でのテストパターン生成に限定されるものではなく、集積回路内部の信号線に断線故障を仮定することにより断線故障を単位としてテストパターン系列を生成することもできる。

テストパターン生成方法(断線故障)

つぎに、単一の断線故障にたいする第1発明のテストパターン生成方法について説明する。図21に被試験CMOS集積回路の一例を示す。被試験集積回路は、図20の回路と同一であって、入力端子x₃とゲートG₁及びG₂間の信号線m₁及びm₂、ゲートG₁及びG₂とゲートG₃間の信号線m₃及びm₄、ゲートG₃の出力側の信号線m₅、信号線m₅とゲートG₄及びG₅との間の信号線m₆及びm₇、ゲートG₄及びG₅とゲートG₆内の信号m₈及びm₉の9つの内部信号線があり、信号線m₆における断線故障(x印で示す)にたいするテストパターン生成を考える。はじめに、故障信号線m₆にslow-to-rise故障の初期値“0”をあたえるテストパターンv1 = “x x 0 x x”を含意操作によりもとめ、図21aに示す信号状態とする。つぎに、故障信号線m₆の論理値が設定した初期値“0”に固定してしまう縮退故障を仮定し、この縮退故障を論理ゲートG₄の出力に伝搬するためのテストパターンv2 = “0 x 1 1 x”を含意操作によりもとめ、図21aに示す信号状態とする。以上により、信号線m₆における断線故障を過渡電源電流試験法で検出するテストパターン系列はT = < v1, v2 > = < “x x 0 x x”, “0 x 1 1 x” > ともとめることができる。

【0088】

あるいは、故障信号線m₆にslow-to-rise故障に対応する信号“U1”をあたえ、上記slow-to-rise故障の影響を論理ゲートG₄の出力に伝搬するテストパターン系列T = “S0 S0 U1 S1 XX”を含意操作(implication)によりもとめる(図21c)。以上により、信号線m₆における断線故障を過渡電源電流試験法で検出するテストパターン系列はT = “S0 S0 U1 S1 XX” = < v1, v2 > = < “0001x”, “0011x” > ともとめることができる。

【0089】

信号線m₆が断線故障をもつとき、論理ゲートG₄の入力遷移時間が大きくなる。この故障の影響をテストパターン系列TまたはT'により論理ゲートG₄(の出力)に伝搬すると、論理ゲートG₄の過渡電源電流波形が正常時と異なるため、被試験集積回路の過渡電源電流は異常を示す。したがって、テストパターン系列TまたはT'をあたえ被試験集積回

10

20

30

40

50

路の過渡電源電流に異常が生じるか否かを観測することによって、信号線 m_6 における断線故障が存在するか否かを判別することができる。

【0090】

以上により、被試験集積回路内のある断線故障にたいするテストパターン系列を生成することができる。さらに、この発明のテストパターン生成方法は、論理ゲートを接続する信号線における断線故障単位でのテストパターン生成に限定されるものではなく、論理ゲート内部の信号線に断線故障を仮定することにより論理ゲート内部の断線故障もテストパターン生成の対象故障とすることも可能である。

【0091】

また、この発明のテストパターン生成方法は、フリップフロップをもたない組合せ論理回路に限定されるものではなく、フリップフロップなどの記憶素子をもつ順序回路にも適用することができる。

10

また、この発明のテストパターン生成方法は、CMOS集積回路に限定されるものではなく、他のタイプの半導体集積回路にも適用することができる。

I_{DDT} 故障シミュレーション方法 (ゲート遅延故障)

つぎに、ゲート遅延故障単位で被検出故障リストを生成する過渡電源電流故障シミュレーション (以下、 I_{DDT} 故障シミュレーションと書く) 方法について説明する。

【0092】

図22に被試験CMOS集積回路を示す。被試験集積回路は、3つの入力端子 x_1, x_2, x_3 、2つの出力端子 z_1, z_2 、5つの論理ゲート G_1, G_2, G_3, G_4, G_5 、5つの内部信号ノード n_1, n_2, n_3, n_4, n_5 をもつ。また、出力端子 z_1, z_2 は出力バッファ G_6, G_7 介して接続されているものとする。つまり入力端子 x_1 はインバータ論理ゲート G_1 の入力側に接続され、入力端子 x_2, x_3 はそれぞれNAND論理ゲート G_2 の入力側に接続され、論理ゲート G_1, G_2 の各出力側はノード n_1, n_2 を通じてNAND論理ゲート G_3 の入力側に接続され、論理ゲート G_3 の出力側はノード n_3 を通じてインバータ論理ゲート G_4 の入力側と、NOR論理ゲート G_5 の一方の入力側とに接続され、論理ゲート G_5 の他方の入力側に入力端子 x_3 が接続され、論理ゲート G_4 の出力側はノード n_4 、バッファ G_6 を通じて出力端子 z_1 に接続され、論理ゲート G_5 の出力側はノード n_5 、バッファ G_7 を通じて出力端子 z_2 に接続される。図に示していないが論理ゲート $G_1 \sim G_5$ および出力バッファ G_6, G_7 の各電源端子は共通の電源に接続されている。

20

30

【0093】

上記被試験CMOS集積回路にたいしておこなった I_{DDT} 故障シミュレーション結果の一例を図23に示す。図23において、左から第1列目はテストパターン系列の識別子を示す。第2列目は被試験CMOS集積回路の入力端子 x_1, x_2, x_3 にあたえる入力信号を示し、第3列目は各テストパターン系列をあたえたとき被試験CMOS集積回路の内部信号ノード n_1, n_2, n_3, n_4, n_5 に生じる信号を示し、第4列目は各テストパターン系列をあたえたとき被試験CMOS集積回路の出力端子 z_1, z_2 に生じる信号を示す。ここで、図22の第2、第3、第4列における信号“L”、“H”、“R”、“F”は、それぞれ、常時ローレベルの信号 (< “0”, “0” > (< > 内の第1要素は初期信号値を示し、第2要素は最終信号値を示す)、常時ハイレベルの信号 (< “1”, “1” >)、ローレベルからハイレベルへの立ち上がり信号 (< “0”, “1” >)、ハイレベルからローレベルへの立ち下がり信号 (< “1”, “0” >) を示す。

40

【0094】

このため、各テストパターン系列は2つのテストパターンからなり、たとえば、テストパターン系列 $T_1 = \text{“LLR”}$ は $x_1 x_2 x_3 = \text{“000”, “001”}$ を意味する。図23の第5列目は、各テストパターン系列をあたえたとき過渡電源電流をもちいた試験で検出可能なゲート遅延故障の集合 (被検出故障リスト) を示す。論理ゲートがゲート遅延故障をもつとき、論理ゲートの出力遷移時刻が遅れる。これに伴って、この故障論理ゲートの出力を入力とする論理ゲートの過渡電流ピークのタイミングが遅くなり、被試験集

50

積回路の過渡電源電流は異常を示す。このため、あるテストパターン系列をあたえ過渡電源電流に異常が生じるか否かを観測することによって、上記入力テストパターン系列によりスイッチング動作（立ち上がり遷移または立ち下がり遷移）を生じる論理ゲートにたいし、これらの論理ゲートの出力信号線を入力とする論理ゲートがスイッチング動作を生じるとき、これらの論理ゲートが故障しているか否かを判別することができる。

【0095】

たとえば、図22に示す被試験CMOS集積回路にテストパターン系列T2をあたえると、被試験CMOS集積回路内の論理ゲート G_2 、 G_3 、 G_4 、 G_5 および出力バッファ G_6 、 G_7 にスイッチング動作を生じ、信号線 n_2 、 n_4 、 n_5 および出力端子 z_1 、 z_2 に立ち下がり遷移が生じ、信号線 n_3 に立ち上がり遷移が生じる。したがって、論理ゲート G_2 、 G_4 、 G_5 のいずれかにslow-to-fall故障が存在するとき、または、論理ゲート G_3 にslow-to-rise故障が存在するとき、テストパターン系列T2をもちいた過渡電源電流試験において過渡電源電流に異常が観測される。すなわち、テストパターン系列T2をもちいた過渡電源電流試験により、論理ゲート G_2 、 G_4 、 G_5 のslow-to-fall故障および論理ゲート G_3 のslow-to-rise故障を検出できる。したがって、テストパターン系列T2にたいする被検出故障リストは、以上の I_{DDT} 故障シミュレーションにより{G2F, G3R, G4F, G5F}とまとめることができる。ここで“F”および“R”は、それぞれslow-to-fall故障あるいはslow-to-rise故障であることを示す。

【0096】

以上により、あるテストパターン系列で検出可能なゲート遅延故障単位の被検出故障リストを生成することができる。

I_{DDT} 故障シミュレーション方法（断線故障）

つぎに、断線故障単位で故障リストを生成する I_{DDT} 故障シミュレーション方法について説明する。図24に被試験CMOS集積回路を示す。これは図22に示したものと同一であるが、各入力端子 x_1 、 x_2 、 x_3 と論理ゲートとの接続信号線、各論理ゲート間の接続信号線をそれぞれ信号線 m_1 、 m_2 、…、 m_{12} により識別させている。ここで、信号線には入出力信号線も含むものとし、分岐した信号線はそれぞれ別々の信号線とした。また、出力信号線 m_{11} 、 m_{12} は出力バッファ G_6 、 G_7 に接続されているものとする。

【0097】

上記被試験CMOS集積回路にたいしておこなった I_{DDT} 故障シミュレーション結果の一例を図25に示す。図25において、左から第1列目はテストパターン系列の識別子を示し、第2列目は被試験CMOS集積回路の入力端子 x_1 、 x_2 、 x_3 にあたえる入力信号を示し、第3列目は各テストパターン系列をあたえたとき被試験CMOS集積回路の信号線 m_1 、 m_2 、…、 m_{12} に生じる信号を示し、第4列目は各テストパターン系列をあたえたとき被試験CMOS集積回路の出力端子 z_1 、 z_2 に生じる信号を示す。ここで、信号“L”、“H”、“R”、“F”は、図23で説明したものと同一であり、たとえば、テストパターン系列T1 = “LLR”は $x_1 x_2 x_3 = < “000” , “001” >$ を意味する。図25の第5列目は、各テストパターン系列をあたえたとき過渡電源電流をもちいた試験で検出可能な断線故障の信号線の集合、すなわち、被検出故障リストを示す。集積回路内部の信号線が断線故障をもつとき、故障信号線を入力とする論理ゲートのスイッチング動作が遅くなり、これに伴って論理回路の過渡電源電流波形が変化するため、被試験集積回路の過渡電源電流は異常を示す。このため、あるテストパターン系列をあたえ過渡電源電流に異常が生じるか否かを観測することによって、上記入力テストパターン系列によりスイッチング動作を生じる信号線にたいし、これらの信号線を入力とする論理ゲートがスイッチング動作を生じるとき、これらの論理回路が故障しているか否かを判別することができる。

【0098】

たとえば、図24に示す被試験CMOS集積回路にテストパターン系列T6をあたえると、被試験CMOS集積回路内の信号線 m_2 、 m_7 、 m_8 、 m_9 、 m_{10} 、 m_{11} にスイッチン

グ動作が生じ、さらに被試験CMOS集積回路内の論理ゲート G_2 、 G_3 、 G_4 および出力バッファ G_6 にスイッチング動作が生じる。したがって、信号線 m_2 、 m_7 、 m_8 、 m_9 、 m_{11} のいずれかに断線故障が存在するとき、テストパターン系列T6をもちいた過渡電源電流試験において過渡電源電流に異常が観測される。すなわち、テストパターン系列T6をもちいた過渡電源電流試験により、信号線 m_2 、 m_7 、 m_8 、 m_9 、 m_{11} における断線故障を検出できる。したがって、テストパターン系列T6にたいする被検出故障リストは、以上の I_{DDT} 故障シミュレーションにより $\{m_2, m_7, m_8, m_9, m_{11}\}$ ととめることができる。ここで、断線故障のリストは、断線故障が生じうる信号線で示す。

【0099】

以上により、あるテストパターン系列で検出可能な断線故障単位の被検出故障リストを生成することができる。さらに、上記 I_{DDT} 故障シミュレーション方法は、論理ゲートを接続する信号線における断線故障に限定されるものではなく、論理ゲート内部の信号線上に故障を仮定することにより論理ゲート内部の信号線における断線故障を対象とすることも可能である。

I_{DDT} 故障シミュレーションを利用したテストパターン生成方法

つぎに、第1発明のテストパターン生成方法について説明する。上記の I_{DDT} 故障シミュレーションの結果にも示すとおり、ひとつのテストパターン系列により複数のゲート遅延故障あるいは断線故障を検出できる。このため、ある故障(対象故障)にたいし生成されたテストパターン系列により他の故障(被検出故障)を検出できるとき、これらの被検出故障にたいしあらためてテストパターン生成をおこなう必要はない。また、 I_{DDT} 故障シミュレーションは、入力信号にたいする内部信号線の論理信号値を求める論理シミュレーションにより実現できるため、内部信号値から入力信号を逆算するテストパターン生成に比べ、一般的に処理時間が短い。したがって、ある対象故障にたいしてテスト生成をおこなった後、 I_{DDT} 故障シミュレーションによって被試験故障を対象故障のリストから削除することにより、テストパターン生成の処理時間を短縮することができる。

被検出故障リスト生成方法(パス遅延故障)

つぎに、被検出故障リストを生成する方法について説明する。図26に被試験CMOS集積回路の一例を示す。被試験集積回路は、4つの入力端子 x_1 、 x_2 、 x_3 、 x_4 、2つの出力端子 z_1 、 z_2 、5つの内部信号ノード n_1 、 n_2 、 n_3 、 n_4 、 n_5 、5つの論理ゲート G_1 、 G_2 、 G_3 、 G_4 、 G_5 、および、出力バッファ G_6 、 G_7 をもつ。入力端子 x_1 、 x_2 はANDゲート G_1 の二つの入力側に接続され、ゲート G_1 の出力側は信号ノード n_1 を介してインバータゲート G_2 とORゲート G_3 の各入力側に接続され、インバータゲート G_2 の出力側はノード n_2 を介してANDゲート G_4 の一方の入力側に接続され、ゲート G_3 の出力側はノード n_3 を介してANDゲート G_4 の他方の入力側とANDゲート G_5 の一方の入力側に接続され、入力端子 x_3 はORゲート G_3 の他方の入力側に接続され、入力端子 x_4 はANDゲート G_5 の他方の入力側に接続され、ANDゲート G_4 及び G_5 の各出力側はそれぞれノード n_4 、 n_5 を介し、更に出力バッファ G_6 、 G_7 を介して出力端子 z_1 、 z_2 に接続される。ゲート $G_1 \sim G_7$ の各電源端子は図に示していないが共通の電源に接続されている。

【0100】

また、ゲート G_1 、 G_2 、 G_3 、 G_4 、 G_5 、 G_6 、 G_7 は、それぞれ1、1、3、2、1、1、1のゲート遅延(伝搬遅延)をもつと仮定する。上記被試験CMOS集積回路に対しておこなった故障シミュレーション結果の一例を図27に示す。図27において、左から第1列目はテストパターン系列の識別子を示し、第2列目は被試験CMOS集積回路の入力端子 x_1 、 x_2 、 x_3 、 x_4 にあたる入力信号を示し、第3列目は各テストパターン系列をあてたとき被試験CMOS集積回路の内部信号ノード n_1 、 n_2 、 n_3 、 n_4 、 n_5 に生じる遷移信号値列を示し、第4列目は各テストパターン系列をあてたとき被試験CMOS集積回路の出力端子 z_1 、 z_2 に生じる遷移信号値列を示す。信号“L”、“H”、“R”、“F”は、図23、図25の場合と同様であり、たとえば、テストパターン系列T1 = “FHHL”は $x_1 x_2 x_3 x_4 = < “1110”, “0110” >$ を

10

20

30

40

50

意味する。また、信号値の下の () 内の数値は、入力信号の遷移時刻を 0 としたときの、信号の遷移時刻を表す。たとえば、 $R(3)$ は、時刻 3 に立ち上がり遷移信号 (“ R ”) が生じることを示す。

【 0 1 0 1 】

図 27 の第 5 列目は、各テストパターン系列をあたえたとき過渡電源電流を用いた試験で検出可能な遅延故障パスの集合 (被検出故障リスト) を示す。被試験集積回路がパス遅延故障をもつとき、故障パスの出力遷移時刻が遅れる。これに伴って、この故障パス上のすべて、または、いずれかの論理ゲートの出力遷移時刻が変化することによって上記論理ゲート出力を入力とする次段の論理ゲートの過渡電流ピークのタイミングが遅くなり、被試験集積回路の過渡電源電流は異常を示す。このため、あるテストパターン系列をあたえ過渡電源電流に異常が生じるか否かを観測することによって、上記入力テストパターン系列によりパス上のすべての論理ゲートがスイッチング動作 (立ち上がり遷移または立ち下がり遷移) を生じる信号伝搬パスに対し、パス遅延故障が生じているか否かを判別することができる。

10

【 0 1 0 2 】

たとえば、上記被試験 CMOS 集積回路にテストパターン系列 T1 をあたえると、図 28 a に示すように、被試験 CMOS 集積回路内の論理ゲート G_1 の出力 n_1 に遷移信号 $F(1)$ が生じる。同様に、論理ゲート G_2 、 G_4 および出力バッファ G_6 の出力には、それぞれ信号 $R(2)$ 、 $R(4)$ 、 $R(5)$ が生じる。したがって、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ 上のすべての論理ゲートがスイッチングするため、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ にパス遅延故障があるときテストパターン系列 T1 を用いた過渡電源電流試験において過渡電源電流に異常が観測される。すなわち、テストパターン系列 T1 を用いた過渡電源電流試験により、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ におけるパス遅延故障を検出できる。したがって、テストパターン系列 T1 に対する被検出故障リストは、以上の故障シミュレーションにより $\{\{x_1, n_1, n_2, n_4, z_1\}\}$ と求めることができる。

20

【 0 1 0 3 】

別の例では、上記被試験 CMOS 集積回路にテストパターン系列 T2 をあたえると、図 28 b に示すように、被試験 CMOS 集積回路内の論理ゲート G_1 の出力 n_1 に遷移信号 $F(1)$ が生じ、同様に、論理ゲート G_2 、 G_3 、 G_4 および出力バッファ G_6 の出力には、それぞれ信号 $R(2)$ 、 $F(4)$ 、 $R(4) - F(6)$ 、 $R(5) - F(7)$ が生じる。したがって、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ および $\{x_1, n_1, n_3, n_4, z_1\}$ 上のすべての論理ゲートがスイッチングするため、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ または $\{x_1, n_1, n_3, n_4, z_1\}$ の何れかにパス遅延故障があるときテストパターン系列 T2 を用いた過渡電源電流試験において過渡電源電流に異常が観測される。しかし、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ のパス遅延時間が増加すると、出力信号線 z_1 における立ち上がり遷移 $R(5)$ が遅延し、 z_1 における信号遷移が生じなくなる恐れがある。このため、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ のパス遅延時間を過渡電源電流試験により測定することができなくなり、信号伝搬パス $\{x_1, n_1, n_2, n_4, z_1\}$ のパス遅延故障を検出できない。すなわち、テストパターン系列 T2 を用いた過渡電源電流試験により検出できるパス遅延故障は、信号伝搬パス $\{x_1, n_1, n_3, n_4, z_1\}$ におけるパス遅延故障のみである。したがって、テストパターン系列 T2 に対する被検出故障リストは、以上の故障シミュレーションにより $\{\{x_1, n_1, n_3, n_4, z_1\}\}$ と求めることができる。

30

40

【 0 1 0 4 】

以上により、あるテストパターン系列で検出可能なパス遅延故障の被検出故障リストを生成することができる。ここで、被検出故障リストに登録されるパス遅延故障をもつ信号伝搬パスは、被試験回路の入力端子から出力端子まで到達するパスに限定されるものではなく、たとえば図 26 に示した半導体集積回路の $\{x_1, n_1\}$ 、または $\{x_1, n_1, n_3\}$ のように出力端子まで到達しない信号伝搬パスでもよい。

50

【0105】

さらに、このような被検出故障リストの生成はCMOS集積回路に限定されるものではなく、他のタイプの半導体集積回路にも適用することができる。

テストパターン生成方法

つぎに、第2発明のテストパターン生成方法について説明する。図27の遷移シミュレーションによる被検出故障生成方法の結果にも示すとおり、単一のテストパターン系列T3により複数のパス遅延故障を検出できる。このため、ある故障(対象故障)に対し生成されたテストパターン系列により他の故障(被検出故障)を検出できるとき、これらの被検出故障に対してあらためてテストパターン生成をおこなう必要はない。また、上記遷移シミュレーションは、入力信号に対する内部信号線の論理信号値を求める論理シミュレーションにより実現できるため、内部信号値から入力信号を逆算するテストパターン生成に比べ、一般的に処理時間が短い。したがって、テストパターン生成の処理時間を短縮することができる。さらに、この第2発明の基礎となる過渡電源電流試験法は、被試験パス上の信号線やサイド入力にハザードを生じる場合でもパス遅延故障を効率的に試験できる。たとえば、図28bに示すように被試験パス $\{x_1, n_1, n_3, n_4, z_1\}$ の出力にハザードが生じている場合でも、過渡電源電流試験法はテストパターンT2により被試験パスのパス遅延故障を検出できる。このため、単一のテストパターン系列で検出可能なパス遅延故障の数が大きくなる可能性があり、結果として被試験半導体集積回路のパス遅延故障を試験する全テストパターン系列の数を小さくすることができる。

10

【0106】

さらに、この第2発明のテストパターン生成方法はCMOS集積回路に限定されるものではなく、他のタイプの半導体集積回路にも適用することができる。

20

【0107】

【発明の実施の形態】

以下、この発明の実施例について説明する。

図29は第1発明のテストパターン生成方法の実施例の処理手順を示す。はじめに、ステップ101において、被試験半導体集積回路にたいするすべての故障を登録した故障リストを作成する。この故障リストの作成は被試験半導体集積回路の例えば各論理ゲートについてslow-to-fall故障(GF)とslow-to-rise故障(GR)を人手で順次列挙してもよく、あるいは回路情報から電子計算機により各論理ゲート(各内部信号線)を自動的に取出して作成してもよい。つぎに、ステップ102において、上記故障リストからテストパターン生成をおこなう対象故障を一つ選択する。ステップ103において、上記選択した対象故障に対応する故障箇所(故障箇所)に故障を活性化するための初期値をあたえる初期化テストパターンv1を含意操作によりもとめる。ステップ104において、上記初期化テストパターンv1を生成するための含意操作により上記被試験半導体集積回路の入力信号線および内部信号線の論理信号に矛盾を生じることなく初期化テストパターンを生成することができたか否かを確認し、矛盾を生じない初期化テストパターンを生成することができたならばステップ105に移行し、矛盾を生じない初期化テストパターンを生成することができないならばステップ108に移行する。

30

【0108】

つまりステップ103で含意操作により初期化テストパターンv1の生成を試み、その生成ができたならばステップ104で矛盾なく生成できたことになる。ステップ105において、上記故障箇所が上記初期値に縮退する故障を仮定し、上記故障箇所を入力とする次段の論理ゲートの出力に上記縮退故障を伝搬する伝搬テストパターンv2を含意操作によりもとめることを試み、ステップ106において、その伝搬テストパターンv2を生成するための含意操作により上記被試験半導体集積回路の入力信号線および内部信号線の論理信号に矛盾を生じることなく伝搬テストパターンを生成することができたか否かを確認し、つまり、伝搬テストパターンを生成することができたならばステップ107に移行し、伝搬テストパターンを生成することができないならばステップ108に移行する。

40

【0109】

50

ステップ107において、上記ステップ103および105で求めた初期化テストパターンv1と伝搬テストパターンv2でテストパターン系列T = (v1, v2)を構成し、テストパターン系列Tをテストパターンリストに登録する。最後に、ステップ108において、上記故障リストに処理されていない他の故障が存在するか否かを確認し、処理されていない故障が存在するならば上記ステップ102, 103, 104, 105, 106, 107, 108を繰り返し、処理されていない故障が存在しないならば処理を終了する。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。

【0110】

図30は第1発明のテストパターン生成方法の他の実施例の処理手順を示している。はじめに、ステップ201において、被試験半導体集積回路にたいするすべての故障を登録した故障リストを作成する。つぎに、ステップ202において、上記故障リストからテストパターン生成をおこなう対象故障を選択する。つぎに、ステップ203において、上記対象故障に対応する故障箇所を上記故障を活性化するための、5値論理システムにおける信号値をあたえ、上記故障箇所を入力とする次段の論理ゲートの出力に上記信号値を伝搬するテストパターン系列Tを含意操作によりもとめることを試み、ステップ204において、上記含意操作により上記被試験半導体集積回路の入力信号線および内部信号線の信号値に矛盾を生じることなくテストパターン系列Tを生成することができたか否かを確認し、つまりテストパターン系列Tを生成することができたならばステップ205に移行し、テストパターン系列Tを生成することができないならばステップ206に移行する。ステップ205において、上記ステップ203で求めたテストパターン系列Tをテストパターンリストに登録する。最後に、ステップ206において、上記故障リストに処理されていない他の故障が存在するか否かを確認し、処理されていない故障が存在するならばステップ202, 203, 204, 205, 206を繰り返し、処理されていない故障が存在しないならば処理を終了する。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。

【0111】

図31は第1発明のテストパターン生成方法の更に他の実施例の処理手順を示している。はじめに、クライアント(テストパターンを管理する電子計算機)が、ステップ301において、被試験半導体集積回路にたいするすべての故障を登録した故障リストを作成する。つぎに、クライアントは、ステップ302において、上記故障リストを複数のサブ故障リストに分割し、複数のサーバに送信する。その後、クライアントは、ステップ303において、各サーバからのテストパターン系列返送を待機し、テストパターン系列Tが返送されたならばステップ304においてテストパターン系列Tをテストパターンリストに登録する。クライアントは、最後に、ステップ305において、すべてのサーバのテストパターン生成処理が終了したか否かを確認し、すべてのサーバの処理が終了していないならば上記ステップ303, 304, 305を繰り返し、すべてのサーバの処理が終了したならば処理を終了する。

【0112】

一方、各サーバは、図32に示すように、はじめにステップ401において、クライアントから送信されたサブ故障リストを受信し、サブ故障リストを記憶手段に格納する。つぎに、サーバは、ステップ402において、上記記憶手段内のサブ故障リストからテストパターン生成をおこなう対象故障を選択する。サーバは、ステップ403において、選択した対象故障に対応する故障箇所に故障を活性化するための初期値をあたえる初期化テストパターンv1を含意操作によりもとめることを試み、サーバは、ステップ404において、その初期化テストパターンv1を生成するための含意操作により上記被試験半導体集積回路の入力信号線および内部信号線の論理信号に矛盾を生じることなく初期化テストパターンを生成することができたか否かを確認し、つまり初期化テストパターンを生成することができたならばステップ405に移行し、初期化テストパターンを生成することができないならばステップ408に移行する。

10

20

30

40

50

【0113】

サーバは、ステップ405において、上記故障箇所が上記初期値に縮退する故障を仮定し、上記故障箇所を入力とする次段の論理ゲートの出力に上記縮退故障を伝搬する伝搬テストパターン v_2 を含意操作によりもとめることを試み、サーバは、ステップ406において、上記伝搬テストパターン v_2 を生成するための含意操作により上記被試験半導体集積回路の入力信号線および内部信号線の論理信号に矛盾を生じることなく伝搬テストパターンを生成することができたか否かを確認し、つまり伝搬テストパターンを生成することができたならばステップ407に移行し、伝搬テストパターンを生成することができないならばステップ408に移行する。サーバは、ステップ407において、上記ステップ403および405で求めた初期化テストパターン v_1 と伝搬テストパターン v_2 でテストパターン系列 $T = \langle v_1, v_2 \rangle$ を構成し、テストパターン系列 T を上記クライアントに返送する。サーバは、最後に、ステップ408において、上記記憶手段内のサブ故障リストに処理されていない他の故障が存在するか否かを確認し、処理されていない故障が存在するならば上記ステップ402, 403, 404, 405, 406, 407, 408を繰り返し、処理されていない故障が存在しないならばテストパターン生成処理終了をクライアントに報告して処理を終了する。このように複数サーバにテストパターン系列により故障を検出できるか否かを調べる処理を行わせることにより、全体の処理速度を向上させることができる。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。

10

【0114】

図33は第1発明のテストパターン生成方法の更に他の実施例の処理手順を示している。はじめに、クライアントが、ステップ501において、被試験半導体集積回路にたいするすべての故障を登録した故障リストを作成する。クライアントは、ステップ502において、上記作成した故障リストを複数のサブ故障リストに分割し、複数のサーバに送信する。その後、クライアントは、ステップ503において、各サーバからのテストパターン系列返送を待機し、テストパターン系列 T が返送されたならばステップ504においてテストパターン系列 T をテストパターンリストに登録する。クライアントは、最後に、ステップ505において、すべてのサーバのテストパターン生成処理が終了したか否かを確認し、すべてのサーバの処理が終了していないならば上記ステップ503, 504, 505を繰り返し、すべてのサーバの処理が終了したならば処理を終了する。

20

30

【0115】

一方、各サーバは、図34に示すようにステップ601において、クライアントから送信されたサブ故障リストを受信し、サブ故障リストを記憶手段に格納する。つぎに、サーバは、ステップ602において、上記記憶手段内のサブ故障リストからテストパターン生成をおこなう対象故障を選択する。つぎに、サーバは、ステップ603において、上記選択した対象故障に対応する故障箇所に上記故障を活性化するための5値論理システムにおける信号値をあたえ、上記故障箇所を入力とする次段の論理ゲートの出力に上記信号値を伝搬するテストパターン系列 T を含意操作によりもとめることを試み、サーバは、ステップ604において、上記テストパターン系列 T を含意操作により上記被試験半導体集積回路の入力信号線および内部信号線の信号値に矛盾を生じることなくテストパターン系列 T を生成することができたか否かを確認し、つまりテストパターン系列 T を生成することができたならばステップ605に移行し、テストパターン系列 T を生成することができないならばステップ606に移行する。サーバは、ステップ605において、上記ステップ603で求めたテストパターン系列 T を上記クライアントに返送する。最後に、ステップ606において、上記サブ故障リストに処理されていない他の故障が上記記憶手段内に存在するか否かを確認し、処理されていない故障が存在するならばステップ602, 603, 604, 605, 606を繰り返し、処理されていない故障が存在しないならばテストパターン生成処理終了をクライアントに報告して処理を終了する。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。

40

50

【 0 1 1 6 】

図 3 5 は第 1 発明のテストパターン生成方法の更に他の実施例の処理手順を示している。この処理手順は、図 2 9 に示した実施例とほぼ同様であるが、異なる点はステップ 1 0 7 において、上記ステップ 1 0 3 および 1 0 5 で求めた初期化テストパターン v_1 と伝搬テストパターン v_2 でテストパターン系列 $T = \langle v_1, v_2 \rangle$ を構成し、テストパターン系列 T をテストパターンリストに登録した後、ステップ 7 0 1 において、上記ステップ 1 0 7 でもとめたテストパターン系列 T をもちいて過渡電源電流故障シミュレーション (I_{DDT} 故障シミュレーション) をおこない、上記テストパターン系列で検出可能な故障のリスト (被検出故障リスト) を作成する。つぎに、ステップ 7 0 2 において、上記ステップ 7 0 1 で生成された被検出故障リスト内の故障を上記ステップ 1 0 1 で作成した故障リストから削除する。最後に、ステップ 1 0 8 において、上記故障リストに処理または削除されていない他の故障が存在するか否かを確認し、処理または削除されていない故障が存在するならば上記ステップ 1 0 2, 1 0 3, 1 0 4, 1 0 5, 1 0 6, 1 0 7, 7 0 1, 7 0 2, 1 0 8 を繰り返し、処理または削除されていない故障が存在しないならば処理を終了する。前記削除処理によりテストパターン系列生成の終了がはやくなる。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。

10

【 0 1 1 7 】

図 3 6 は第 1 発明のテストパターン生成方法の更に別の実施例の処理手順を示している。これは、図 3 0 に示した処理手順に、図 2 9 に対し図 3 5 で加えたと同様の処理を加えたものである。つまり、図 3 0 に対し異なる点はステップ 2 0 5 でテストパターン系列 T をテストパターンリストに登録した後、ステップ 8 0 1 において、上記ステップ 2 0 3 でもとめたテストパターン系列 T をもちいて過渡電源電流故障シミュレーション (I_{DDT} 故障シミュレーション) をおこない、上記テストパターン系列で検出可能な故障のリスト (被検出故障リスト) を作成する。つぎに、ステップ 8 0 2 において、上記ステップ 8 0 1 で生成された被検出故障リスト内の故障を上記ステップ 2 0 1 で作成した故障リストから削除する。最後に、ステップ 2 0 6 において、上記故障リストに処理または削除されていない他の故障が存在するか否かを確認し、処理または削除されていない故障が存在するならばステップ 2 0 2, 2 0 3, 2 0 4, 2 0 5, 8 0 1, 8 0 2, 2 0 6 を繰り返し、処理または削除されていない故障が存在しないならば処理を終了する。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。ステップ 2 0 5 の登録とステップ 8 0 1, 8 0 2 の削除とは順序を逆にしてもよい。

20

30

【 0 1 1 8 】

図 3 7 および図 3 8 は第 1 発明のテストパターン生成方法のさらに別の実施例の処理手順を示している。図 3 1 及び図 3 2 に示した実施例と異なる点を主に説明する。クライアントは、ステップ 3 0 1 で故障リストを作成した後、ステップ 9 0 1 において、テストパターン生成処理をおこなう複数のサーバを起動する。つぎに、クライアントは、ステップ 3 0 3 で、各サーバからのテストパターン系列返送を待機し、テストパターン系列 T が返送されたならばステップ 3 0 4 でテストパターン系列 T をテストパターンリストに登録する。つぎに、クライアントは、ステップ 9 0 2 において、上記テストパターン系列 T をもちいて過渡電源電流故障シミュレーション (I_{DDT} 故障シミュレーション) をおこない、上記テストパターン系列で検出可能な故障のリスト (被検出故障リスト) を作成する。つぎに、クライアントは、ステップ 9 0 3 において、上記ステップ 9 0 2 で生成された被検出故障リスト内の故障を上記ステップ 3 0 1 で作成した故障リストから削除してステップ 3 0 5 に移る。

40

【 0 1 1 9 】

各サーバは、図 3 8 に示すようにクライアントから起動されると、ステップ 1 0 0 1 において、クライアントが保持する上記故障リストからテストパターン生成をおこなう対象故障を選択する。その選択した対象故障を検出するための初期化テストパターン v_1 と、伝

50

搬テストパターン v_2 を生成しこれらよりなるテストパターン系列 $T = \langle v_1, v_2 \rangle$ を構成して上記クライアントに返送する処理は図32と同様である。ステップ407でテストパターン系列を返送した後、サーバは、最後に、ステップ1002において、上記故障リストに処理または削除されていない他の故障が存在するか否かを確認し、処理または削除されていない故障が存在するならばステップ1001, 403, 404, 405, 406, 407, 1002を繰り返し、処理または削除されていない故障が存在しないならばテストパターン生成処理終了する。上記テストパターン生成方法は、テストパターン生成の対象故障をゲート遅延故障としてもよいし、断線故障を対象とすることもできる。この図37、図38においてクライアントはサーバを起動すると同時に、故障リストから各1つずつ対象故障をサーバに送り、サーバからテストパターン系列が送られてくると、登録と、シミュレーションによる被検出故障リストの作成、その故障の故障リストからの削除を行った後、故障リストから未処理の1つを対象故障を選択してそのサーバへ送り、サーバからテストパターン系列の返送があり、それに対する処理をした状態で故障リストに未処理の故障がない場合は、そのことをそのサーバへ通知し、サーバはその通知を受けると処理を終了する。またクライアントは、故障リストに未処理の故障がなくなり、かつ、すべてのサーバから最後のテストパターン系列の返送を受信すると、その登録を行って処理を終了してもよい。

【0120】

図39及び図40は第1発明のテストパターン生成方法のさらに別の処理手順を示している。これは図31及び図32に示した実施例を図37及び図38に示した実施例に変更したと同様に図33及び図34に示した実施例を変更したものである。よって、図39及び図40中の各ステップで図33及び図34、また図37及び図38中のステップ同様の処理を行うものに、同一番号を付け重複説明を省略する。この場合も、テストパターン生成の対象故障は、ゲート遅延故障、断線故障の何れでもよい。

【0121】

次に複数のサーバを用いる場合に、クライアント側で対象故障の選択と未処理故障の存在確認を行う第1発明の実施例を、図41に示すクライアント側の処理手順と、図42に示すサーバ側の処理手順とを参照して説明する。

クライアント側では、ステップ301で故障リストを作成し、ステップ309で複数のサーバを起動する。その後、ステップ1201でサーバから処理する故障の要求を受信したかを調べ、受信していたら、ステップ1202で故障リストから対象故障を1つ選択して、要求のあったサーバへ送信する。次にステップ303でサーバからテストパターン系列の返送があったかを調べ、返送があればステップ304でそのテストパターン系列をテストパターンリストに登録する。次にステップ1203で故障リスト内に処理されていない故障が存在するかを調べ、また存在する場合はステップ1201に戻る。

【0122】

ステップ1201で故障要求が受信されていなければ、ステップ303に移り、ステップ303でテストパターン系列が返送されていなければ、ステップ1203に移る。ステップ1203で故障リスト中に未処理故障が存在してなければ、ステップ1204で各サーバへ処理が終了したことを通知して動作を停止する。

サーバ側においては図42に示すようにクライアントから起動されるとステップ1301で処理する故障をクライアントに要求し、ステップ1302でクライアントから送信された対象故障の受信を待ち、対象故障が受信されると、図32中のステップ403～407を実行し、つまり、初期化テストパターン v_1 、伝搬テストパターン v_2 を含意操作によりもとめることを試み、求めることができたなら、そのテストパターン系列 $T = \langle v_1, v_2 \rangle$ をクライアントに返送する。その返送後、あるいはテストパターン系列を生成できなかった場合は、ステップ1303でクライアントから処理終了通知を受信したかを調べ、受信していなければステップ1301へ戻り、受信していれば動作を停止する。

【0123】

各サーバで対象故障の選択と、未処理故障の存在確認を行う場合の実施例では、クライア

10

20

30

40

50

ントの処理手順は図43に示すように、図37中のステップ902と903を省略したものでよい。サーバの処理は図38に示した処理と実質的に同一である。ただ図38中のステップ1002において、故障リスト中に処理されていない故障が存在するかを調べることになり、クライアント側でステップ902, 903の省略に伴い、故障リストから故障の削除は行われぬ。

【0124】

次に図41、42で示した実施例において、サーバから返送されたテストパターン系列Tを用いて、故障シミュレーションを行い、被検出故障リストを作成し、その被検出故障リスト内の故障を故障リストから削除する場合のクライアントの処理手順を、図44に示す。この場合は、図41に示した処理手順において、ステップ304の次に図37中のステップ902と903を実行してステップ1203に移すようにする。この場合のサーバの処理手順は図42に示した手順と同一となる。

10

【0125】

なお図41乃至図44を参照して説明した実施例ではテストパターン系列の生成を初期化テストパターンと伝搬テストパターンの系列を求めて行ったが、図39、図40に示した実施例のように、5値論理システムにおける信号値を用いてテストパターン系列Tを生成するようにしてもよい。

図45は第2発明のテストパターン生成方法の処理手順を示している。はじめに、ステップ101において、被試験半導体集積回路に対するすべてのパス遅延故障を登録した故障リストを作成する。この作成は回路を手でたどって各パス遅延故障(遅延する際の可能性のあるパス)を列挙する、あるいはこの被試験半導体集積回路の回路情報から電子計算機によりパス遅延故障を列挙することもできる。ステップ102において、上記被試験半導体集積回路にあたる2つ以上のテストパターンからなるテストパターン系列を発生する。つぎに、ステップ103において、上記テストパターン系列に対して遷移シミュレーションをおこない、上記被試験半導体集積回路内部の信号線に生じる遷移信号値列を計算する。ステップ104において、上記遷移シミュレーションにより計算された各信号線の遷移信号値列を用いて上記テストパターン系列により過渡電源電流試験で検出可能なパス遅延故障のリスト(被検出故障リスト)を生成する。つぎに、ステップ105において、上記被検出故障リスト内の故障が上記ステップ101で作成した故障リストに存在するかどうかを確認し、被検出故障リスト内の故障が上記故障リストに存在するならばステップ106に移行し、被検出故障リスト内の故障が上記故障リストに存在しないならば上記ステップ102, 103, 104, 105を繰り返す。

20

30

【0126】

ステップ106において、上記被検出故障リスト内の故障を上記ステップ101で作成した故障リストから削除する。つぎに、ステップ107において、上記テストパターン系列をテストパターン系列リストに登録する。最後にステップ108において、上記故障リストが空になったか否かを確認し、故障リストが空になっていないならば上記ステップ102, 103, 104, 105, 106, 107, 108を繰り返し、故障リストが空になったならば処理を終了する。上記ステップ102のテストパターン系列を発生するステップは、2つ以上のランダムパターンをテストパターン系列として発生してもよいし、あらかじめ作成されたテストパターン系列群の中からひとつ選択して発生してもよい。また、上記ステップ108において、故障リストが空になったか否かを確認するステップは、故障リストに残った故障がある所定の数(0を含む正数)以下になったか否かを確認することによって処理終了を判断してもよい。ただし、被試験半導体集積回路のパス遅延故障検出率を向上するためには、上記所定のしきい値は可能な限り小さいほうが望ましい。残りのパス遅延故障が少なくなると、これを検出可能なテストパターン系列を得るのに時間がかかる。一方、回路によっては故障検出率が例えば99%以上であればよいとされているものもあり、これらを考慮した場合は、その要件を満す程度で残りのパス遅延故障の数を増加させればよい。

40

【0127】

50

図46は、図45中のステップ104における被検出故障リスト生成方法の処理手順を示している。はじめに、ステップ201において、被試験半導体集積回路内に生じうるパス遅延故障を図45中のステップ101で作成した故障リストから1つ選択し、初期設定する。つぎに、ステップ202において、遷移シミュレーションの結果をもとに上記パス遅延故障をもつパス（故障パス）上のすべての論理ゲートがスイッチングするか否かを確認し、上記故障パス上のすべての論理ゲートがスイッチングしているならば、ステップ203に移行し、上記故障パス上のすべての論理ゲートがスイッチングしていないならば、ステップ206に移行する。

【0128】

ステップ203において、遷移シミュレーションの結果をもとに上記故障パス上に図45中のステップ102で発生されたテストパターン系列により複数回スイッチングしている論理ゲートが存在するか否かを確認し、複数回スイッチングしている論理ゲートが存在するならば、ステップ204に移行し、複数回スイッチングしている論理ゲートが存在しないならば、ステップ205において上記パス遅延故障を被検出故障リストに登録する。ステップ204において、上記複数回スイッチングするすべての論理ゲートのオンパス入力（故障パス上の入力信号線）が過渡電源電流試験による故障検出条件を満たすか否かを確認し、過渡電源電流試験による故障検出条件を満たすならば、ステップ205において上記パス遅延故障を被検出故障リストに登録し、過渡電源電流試験による故障検出条件を満たさないならば、ステップ206に移行する。

【0129】

ステップ204では、複数回スイッチングするすべての論理ゲートに対し、上記論理ゲートのオンパス入力に対応する出力遷移信号が、その論理ゲートの出力信号線における最終の遷移信号と同じ種類の遷移信号であるか否かを確認する。例えば図28(b)中の論理ゲート G_4 が複数回スイッチングしている例であり、ゲート G_4 のオンパス入力 n_3 の入力遷移信号“F(4)”に対応するゲート G_4 の出力遷移信号“F(6)”は論理ゲート G_4 の最終の遷移信号に一致しているため、オンパス入力 n_3 は過渡電源電流試験による故障検出条件を満たす。最後に、ステップ206において、他に処理されていないパス遅延故障が存在するか否かを確認し、処理されていないパス遅延故障が存在するならば、ステップ207において起こりうるつぎのパス遅延故障を設定して上記ステップ202, 203, 204, 205, 206を繰り返し、処理されていないパス遅延故障が存在しないならば、処理を終了する。

【0130】

図47および図48は第2発明のテストパターン生成方法の他の実施例の処理手順を示している。はじめに、クライアントが、ステップ301において、被試験半導体集積回路に対するすべてのパス遅延故障に登録した故障リストを作成する。つぎに、クライアントは、ステップ302において、上記半導体集積回路にあたえる2つ以上のテストパターンからなるテストパターン系列を発生し、処理をおこなっていない1以上のサーバにそれぞれ1つずつテストパターン系列を送信する。その後、クライアントは、ステップ303において、上記複数のサーバから処理終了が通知されるのを待機し、処理終了が通知されたならばステップ304に移行する。ステップ304において、上記故障リストが空になったか否かを確認し、故障リストが空になっていなければ、上記ステップ302, 303, 304を繰り返し、故障リストが空になったならば、ステップ305においてすべてのサーバの処理を強制終了して、処理を終了する。

【0131】

一方各サーバは、図48に示すように、ステップ401において、上記クライアントから送信されたテストパターン系列に対し遷移シミュレーションをおこない、被試験半導体集積回路内部の信号線に生じる遷移信号値列を計算する。つぎに、サーバは、ステップ402において、上記遷移シミュレーションにより計算された各信号線の遷移信号値列を用いて上記テストパターン系列により過渡電源電流試験で検出可能な故障のリスト（被検出故障リスト）を生成する。サーバは、ステップ403において、ステップ402で作成され

10

20

30

40

50

た被検出故障リスト内の故障が上記クライアントが保持する上記故障リスト（ステップ301で作成）に存在するか否かを確認し、被検出故障リスト内の故障が上記故障リストに存在するならばステップ404に移行し、被検出故障リスト内の故障が上記故障リストに存在しないならばステップ406に移行する。

【0132】

サーバは、ステップ404において、上記被検出故障リスト内の故障を上記故障リストから削除する。つぎに、サーバは、ステップ405において、クライアントが保持するテストパターン系列リストにテストパターン系列を登録する。最後に、サーバは、ステップ406において、処理終了をクライアントに通知し、処理を終了する。

図47中のステップ302のテストパターン系列を発生するステップは、2つ以上のランダムパターンをテストパターン系列として発生してもよいし、あらかじめ作成されたテストパターン系列群の中からひとつ選択してもよい。また、図48中のステップ402の被検出故障リストを生成するステップは、図46に示した被検出故障リスト生成方法を用いておこなうことができる。図47中のステップ304において、故障リストが空になったか否かを確認するステップは、故障リストに残った故障がある所定の数（しきい値）より小さくなったか否かを確認することによって処理終了を判断してもよい。ただし、被試験半導体集積回路のパス遅延故障検出率を向上するためには、上記所定のしきい値は可能な限り小さいほうが望ましい。

【0133】

第2発明においてクライアントと複数のサーバとを用いる場合に、サーバでクライアントからテストパターン系列を取出して処理する場合の実施例を説明する。クライアントの処理手順は図49に示すように、ステップ301で被試験半導体集積回路に対するパス遅延故障のリスト（故障リスト）を生成し、ステップ1401で被試験半導体集積回路にあたる2つ以上のテストパターンからなるテストパターン系列のリストを生成し、その後ステップ901で複数のサーバを起動する。ステップ1402で複数のサーバからテストパターン系列リストにテストパターン系列が登録されるのを待機し、ステップ1403で故障リスト内のパス遅延故障数が所定値以下になるまで、あるいは、すべてのサーバが処理を終了するまでパターン系列登録待機ステップ1402以後を繰り返し、故障リストが空になるか、全てのサーバが処理を終了すると、ステップ1406で各サーバの処理を強制終了させて動作を停止する。

【0134】

各サーバは、図50に示すように、クライアントから起動されると、ステップ1404で、クライアントが保持するテストパターン系列リストからテストパターン系列を1つ選択し、以後は図48中のステップ401～405を実行し、そのテストパターン系列により、過渡電源電流試験で検出可能な故障のリスト（被検出故障リスト）を生成し、その故障をクライアントが保持する故障リストから削除し、更にそのテストパターン系列をクライアントが保持するテストパターン系列リストに登録する。その後、ステップ1405でクライアントのテストパターン系列リスト内に選択していないテストパターン系列が存在するかを調べ、存在すればステップ1404に戻り、存在しなければステップ406でテストパターン生成処理を停止するとともに処理終了をクライアントに通知する。

【0135】

パス遅延故障を検出するテストパターン系列の生成を、クライアントと複数のサーバを用いる他の実施例のクライアントの処理手順を図51に、サーバの処理手順を図52に示す。クライアントは図49の場合と同様にステップ301と1401でパス遅延故障の故障リストを作り、更にテストパターン系列のリストを作成するが、その後、この実施例ではステップ1501でテストパターン系列リストを複数のサブパターン系列リストに分割し、各サブパターン系列リストを1つずつ各サーバに分配送信する。その後の処理は図49の場合と同様である。サーバは図52に示すように、ステップ1502でクライアントからサブテストパターン系列リストを受信すると、それを記憶部に一旦格納し、ステップ1503でそのサブテストパターン系列リストから1つのテストパターン系列を選択し、

10

20

30

40

50

その後は図50中のステップ401～405を実行し、次にステップ1504で記憶部に格納したサブテストパターン系列リストにまた処理されていないテストパターン系列が存在するかを調べ、存在する場合はステップ1503に戻り、存在しなければ、ステップ406で処理終了をクライアントに通知する。

【0136】

上述したパターン生成は例えばコンピュータにより行わせることができる。例えば第1発明の方法においては図53に示すように、CPU11がメモリ12内のプログラムを実行することにより全体の処理を行う。まず入力手段13により例えば被試験回路の回路情報をダウンロードして記憶部14に一旦格納し、メモリ15内の故障リスト作成プログラムを実行して、得られた故障リストをメモリ16に格納する。次に前述したように、故障リストメモリ16内から対象故障を選択して、その故障箇所を活性化し、その箇所の次の論理ゲートに信号を送るテストパターン系列を、前記初期化テストパターンv1の生成と、伝搬テストパターンv2の生成により、又は5値論理システムにおける信号値を与えて、何れも含意操作により求めるプログラムを、それを格納したメモリ17を用いて実行し、得られたテストパターン系列をテストパターン系列メモリ18に格納(登録)し、以下、順次、故障リストメモリ16から対象故障を取り出して行う。

10

【0137】

図35、図36に示した実施例の場合は、テストパターン系列メモリ18に登録した又は登録するテストパターン系列でIDD_T故障シミュレーションを、メモリ19内のプログラムを実行して行い、更にメモリ21内のプログラムを実行してこのテストパターン系列で検出可能な故障のリストを作成し、そのリストをCPU11内のRAMに一時保持した後、故障リストメモリ16から該当する故障を削除する機能が付加される。

20

【0138】

図31と図32、図33と図34に示した実施例の場合は図54Aにクライアントの機能を示すように、図53中のCPU11、プログラムメモリ12、入力手段13、記憶部14、故障リスト作成プログラムメモリ15、テストパターン系列メモリ18を備え、更にサーバとの送受信部22を備える。ただしメモリ12内プログラムの内容は図53の何れとも異なり、図31又は図33に示した処理を行うようにされている。サーバは図54Bに示すように、CPU23によりメモリ24内のプログラムを実行することにより全体を機能させ、クライアントと通信する送受信部26を介して送られたサブ故障リストはサブ故障リストメモリ25に格納されて処理される。図53中の故障箇所活性化テストパターン系列生成プログラムを格納したメモリ17が設けられる。

30

【0139】

図37と図38又図39と図40に示した実施例の場合はクライアントは図54A中に破線で示すように、図53中のIDD_T故障シミュレーションプログラム格納メモリ19と、被検出故障リスト作成プログラムを格納したメモリ21が付加され、またサーバとの情報の授受などメモリ12内のプログラムが、図37又は図39の処理を行うように作成されている。サーバは図54B中のサブ故障メモリ25が省略され、メモリ24内のプログラムが図38又は図40の処理を行うように作られる。図41と図42に示した実施例の場合はクライアントは図54A中のメモリ19, 21が省略され、サーバとの情報の授受などメモリ12内のプログラムが図41の処理を行うように作成されている。サーバは図54B中のサブ故障メモリ25が省略され、メモリ24内のプログラムが図42の処理を行うように作られている。図43および図44に示した実施例の場合も、図54Aに示した構成と同様にしてコンピュータにより機能させることができる。

40

【0140】

図45に示した実施例の場合は、図55に示すように、CPU31がメモリ32内のプログラムを実行して、全体の機能を行い、入力手段33を通じて、被試験回路の回路情報がダウンロードされて記憶部34に一旦格納され、メモリ35内の故障リスト作成プログラムを記憶部34内の回路情報について実行して、その全てのパス遅延故障を求めて故障リストメモリ36に格納する。次にメモリ37内のテストパターン発生プログラム37を実

50

行してテストパターン系列を生じ（この生成はハードウェアにより行ってもよい）、メモリ38内のプログラムを実行してそのテストパターン系列を被試験回路に印加した場合の遷移シミュレーションを行い、メモリ39内のプログラムを実行して被検出故障リストを作り、これをCPU31内のRAM又は記憶部34に一時蓄積して、この被検出故障（パス）と同一のものを、故障リストメモリ36内から削除して、前記テストパターン系列をテストパターン系列メモリ41に格納する。故障リストメモリ36内のパス遅延故障の数が所定数（0を含む）以下になるまで、同様のことを実行する。

【0141】

図47と図48、図49と図50、図51と図52にそれぞれ示した実施例の場合、クライアントでは図56Aに示すように図55中から遷移シミュレーションプログラムを格納したメモリ38、被検出故障リスト作成プログラムを格納したメモリ39が省略され、サーバと通信を行う送受信部42が設けられる。またメモリ32内のプログラムの内容は図47、図49、図51の各処理とそれぞれ対応したものとなる。

10

【0142】

一方、サーバは図56Bに示すように、送受信部43によりクライアントと通信を行い、被検出故障リストを作成のために、クライアントで作成した故障リストがメモリ44内に予め転送されてある。図55中の遷移シミュレーションプログラムを格納したメモリ38、被検出故障リスト作成プログラムを格納したメモリ39が設けられ、図48の例ではクライアントからテストパターン系列が送られてくるごとにCPU45がメモリ46のプログラムを実行して、図48に示した処理を行い、図50の例ではクライアントからテストパターン系列を取り出して図50に示した処理を行い、図52の例では図56Bにクライアントから受信したサブテストパターン系列リストを格納する記録部が設けられ、図52に示した処理を行う。

20

【0143】

【発明の効果】

第1発明のテストパターン生成方法によれば、可観測性が高くゲート遅延故障や断線故障を効率よく試験できる過渡電源電流試験法を用いることにより、ゲート遅延故障や断線故障に対するテストパターンを容易に生成することができ、テストパターン生成の効率を大幅に改善することができる。

また、第1発明のテストパターン生成方法によれば、ゲート遅延故障および断線故障の影響を出力信号線まで伝搬する必要がないため、テストパターン生成の含意操作において信号線の論理値に矛盾が生じる確率が小さくでき、初期化テストパターン生成および伝搬テストパターン生成において入力信号設定のやり直し（バックトラック）回数を削減できるため、テストパターン生成に要する時間を大幅に削減することができる。

30

【0144】

また、第1発明のテストパターン生成方法の実施例によれば、テストパターン生成より高速な過渡電源電流故障シミュレーション（ I_{DDT} 故障シミュレーション）によりテストパターン生成をおこなう対象故障数を削減できるため、テストパターン生成に要する時間をさらに削減することができる。

第2発明のテストパターン生成方法によれば、可観測性が高くゲート遅延故障や断線故障を効率よく試験できる過渡電源電流試験法を用いることにより、パス遅延故障に対するテストパターンを容易に生成することができ、テストパターン生成の効率を大幅に改善することができる。

40

【0145】

また、第2発明のテストパターン生成方法によれば、テストパターン生成より高速な遷移シミュレーションを用いて被検出故障リストを生成し、テストパターン生成の対象故障のリストから削除していくことにより、テストパターン生成に要する時間を削減することができる。

【図面の簡単な説明】

【図1】従来のテストパターン生成法を説明するための論理回路を示す図。

50

- 【図 2】従来のテストパターン生成法に用いられる 5 値論理システムの各信号値を示す図。
- 【図 3】従来のテストパターン生成法で用いられる含意表を示す図。
- 【図 4】従来のテストパターン生成法で用いられる経路活性化表を示す図。
- 【図 5】従来のテストパターン生成法の一例を説明するための回路図。
- 【図 6】図 5 の続きを示す図。
- 【図 7】a は C M O S インバータの入力電圧 V_{IN} 、出力電圧 V_{OUT} の経時変化とその電源電流 I_{DD} の過渡応答の一例を示す図、b はその C M O S インバータ回路と出力の立ち上がり遷移時に流れる電源電流を示す図、c はその C M O S インバータ回路と出力立ち下り遷移時に流れる電源電流を示す図である。 10
- 【図 8】C M O S 論理ゲートの過渡応答の典型例を示し、a は入力電圧 V_{IN} 、出力電圧 V_{OUT} 、電源電流 I_S の伝達特性図、b は過渡電流の近似波形を示す図である。
- 【図 9】a は C M O S 集積回路の例を示す回路図、b はその集積回路に対する入力電圧、出力電圧の変化と判定する過渡電源電流応答 I_{DDT} の様子を示す図である。
- 【図 10】断線故障を模式的に説明する図。
- 【図 11】a は断線故障をもつ C M O S 集積回路の一例を示す図、b、c はその断線無し、有りの出力波形例を示す図である。
- 【図 12】断線故障をもつ C M O S 集積回路の過渡電源電流応答の一例を示す図。
- 【図 13】a はパス遅延故障試験方法の基本原則を模式的に示す図、b はその入出力とクロックの関係を示す図である。 20
- 【図 14】過渡電源電流のパルス幅を利用した過渡電源電流試験方法の原理図。
- 【図 15】過渡電源電流の瞬時値を利用した過渡電源電流試験方法の原理図。
- 【図 16】a は C M O S インバータの入力信号線に存在する微小オープン欠陥のモデルを示す図、b は信号遷移時間を模式的に示す図である。
- 【図 17】C M O S 集積回路内に存在する微小オープン欠陥の抵抗値に対する C M O S 集積回路の過渡電源電流の積分値の変化を示す図。
- 【図 18】C M O S 集積回路の被試験パス上に存在する微小オープン欠陥の抵抗値に対する被試験パスのパス遅延時間の変化を示す図。
- 【図 19】C M O S 集積回路の被試験パス上に微小オープン欠陥が存在すると仮定したときの、C M O S 集積回路の過渡電源電流の積分値と被試験パスのパス遅延時間の間の線形性を示す図。 30
- 【図 20】第 1 発明のテストパターン生成方法の一例を説明するために用いられる、被試験 C M O S 集積回路の一例を示す図。
- 【図 21】第 1 発明のテストパターン生成方法の他の例を説明するために用いられる、被試験 C M O S 集積回路の別の一例を示す図。
- 【図 22】第 1 発明のテストパターン生成方法で用いられる I_{DDT} 故障シミュレーションを説明するための、被試験 C M O S 集積回路の一例を示す回路図。
- 【図 23】第 1 発明のテストパターン生成方法で用いられる I_{DDT} 故障シミュレーションのシミュレーション結果の一例を示す図。
- 【図 24】第 1 発明のテストパターン生成方法で用いられる I_{DDT} 故障シミュレーションを説明するための、被試験 C M O S 集積回路の別の一例を示す回路図。 40
- 【図 25】第 1 発明のテストパターン生成方法で用いられる I_{DDT} 故障シミュレーションのシミュレーション結果の別の一例を示す図。
- 【図 26】第 2 発明の故障シミュレーション方法を説明するために用いられる、被試験 C M O S 集積回路を示す図。
- 【図 27】第 2 発明のテストパターン生成方法に用いられる故障シミュレーションのシミュレーション結果の一例を示す図。
- 【図 28】図 27 の故障シミュレーション結果を用いて第 2 発明の方法を説明するための回路図。
- 【図 29】第 1 発明のテストパターン生成方法の処理手順の例を示すフローチャート。 50

- 【図30】第1発明のテストパターン生成方法の処理手順の別の例を示すフローチャート。
- 【図31】第1発明のテストパターン生成方法のクライアント側の処理手順の例を示すフローチャート。
- 【図32】図31のクライアントに対するサーバ側の処理手順の例を示すフローチャート。
- 【図33】第1発明のテストパターン生成方法のクライアント側の処理手順の別の例を示すフローチャート。
- 【図34】図33のクライアントに対するサーバ側の処理手順の例を示すフローチャート。
- 【図35】第1発明のテストパターン生成方法の更に別の処理手順を示すフローチャート。
- 【図36】第1発明のテストパターン生成方法の更に別の処理手順を示すフローチャート。
- 【図37】第1発明のテストパターン生成方法の更に別のクライアント側の処理手順を示すフローチャート。
- 【図38】図37のクライアントに対するサーバ側の処理手順を示すフローチャート。
- 【図39】第1発明の方法による更に別のクライアント側の処理手順を示すフローチャート。
- 【図40】図39のクライアントに対するサーバ側の処理手順を示すフローチャート。
- 【図41】第1発明の方法による更に別のクライアントの処理手順を示すフローチャート。
- 【図42】図41のクライアントに対するサーバの処理手順を示すフローチャート。
- 【図43】第1発明の方法による更に別のクライアントの処理手順を示すフローチャート。
- 【図44】第1発明の方法による更に別のクライアントの処理手順を示すフローチャート。
- 【図45】第2発明の方法の処理手順の例を示す流れ図。
- 【図46】図45中の被検出故障リスト作成ステップの具体例を示すフローチャート。
- 【図47】第2発明の方法を適用したクライアント側の処理手順の例を示すフローチャート。
- 【図48】図47のクライアントに対するサーバ側の処理手順を示すフローチャート。
- 【図49】第2発明の方法を適用したクライアントの別の処理手順を示すフローチャート。
- 【図50】図49のクライアントに対するサーバの処理手順を示すフローチャート。
- 【図51】第2発明の方法を適用したクライアントの更に別の処理手順を示すフローチャート。
- 【図52】図51のクライアントに対するサーバの処理手順を示すフローチャート。
- 【図53】第1発明の方法を適用した装置の機能構成例を示す図。
- 【図54】第1発明の方法を適用したクライアント及びサーバの各機能構成例を示す図。
- 【図55】第2発明の方法を適用した装置の機能構成例を示す図。
- 【図56】第2発明の方法を適用したクライアント及びサーバの各機能構成を示す図。

10

20

30

40

【 図 1 】

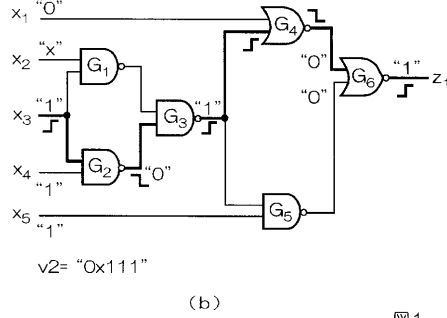
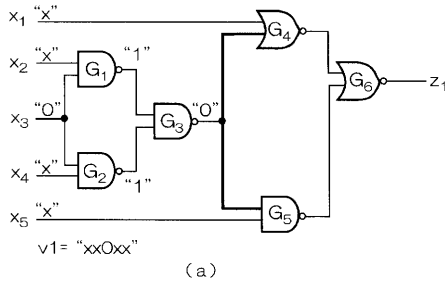


図1

【 図 2 】

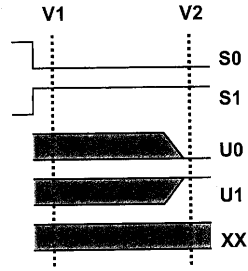


図2

【 図 3 】

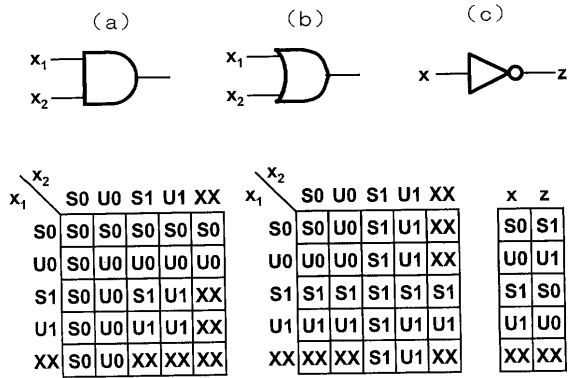


図3

【 図 4 】

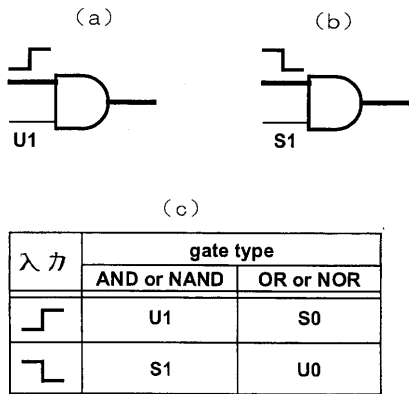


図4

【 図 5 】

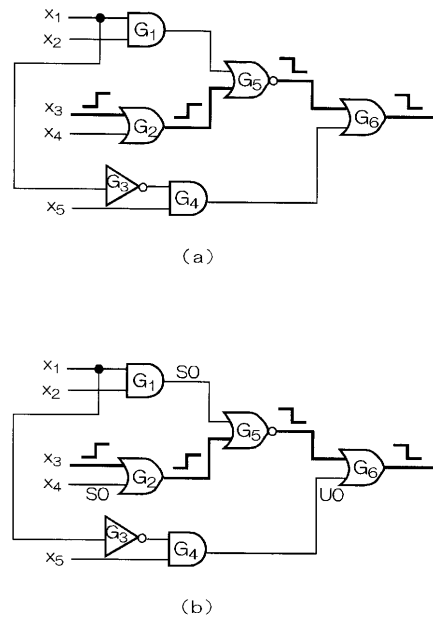


図5

【 図 6 】

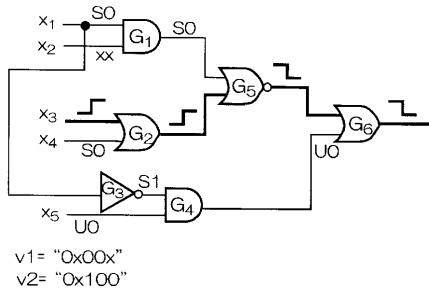


図6

【 図 7 】

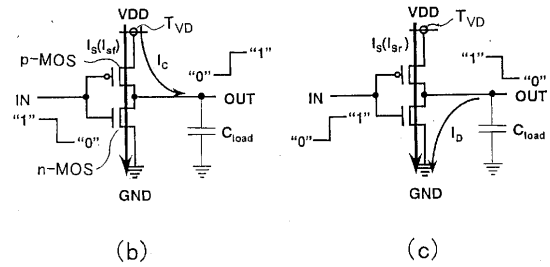
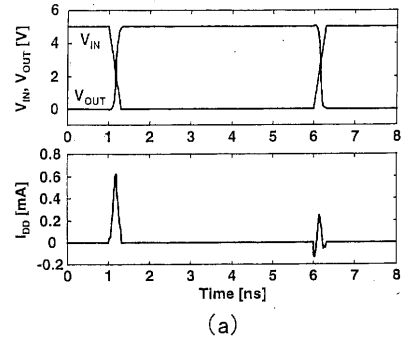


図7

【 図 8 】

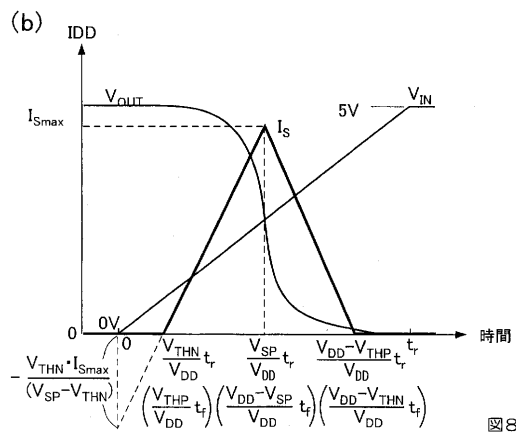
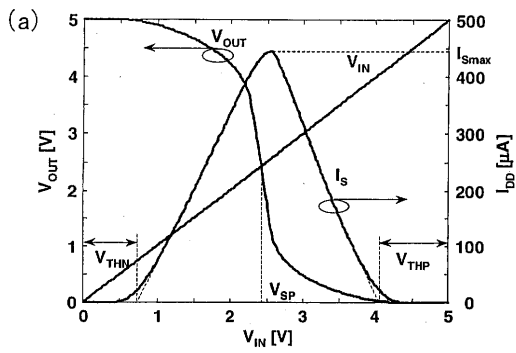


図8

【 図 9 】

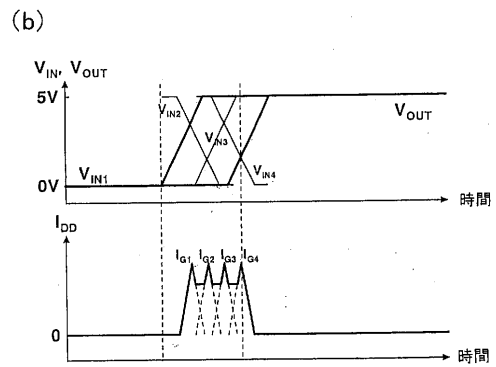
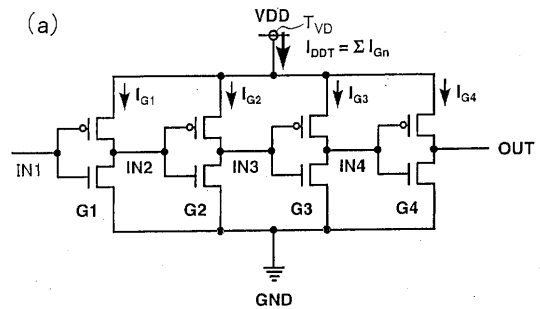
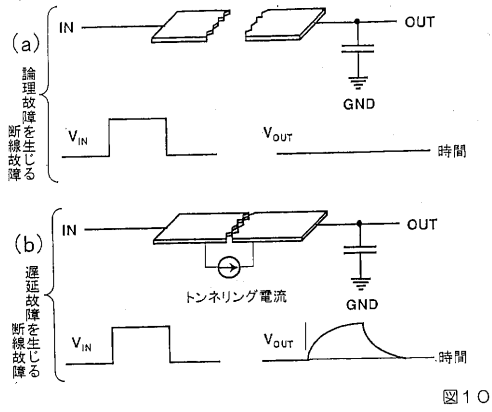
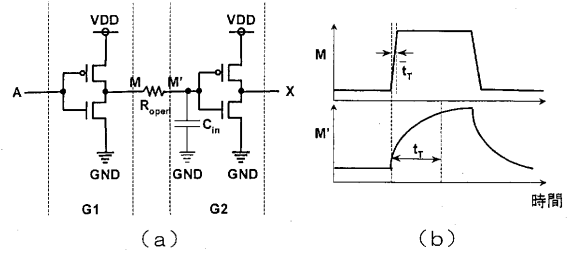


図9

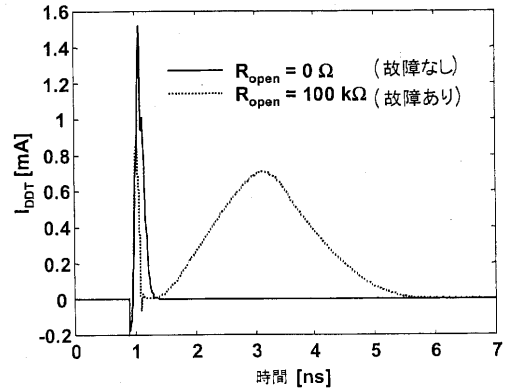
【図10】



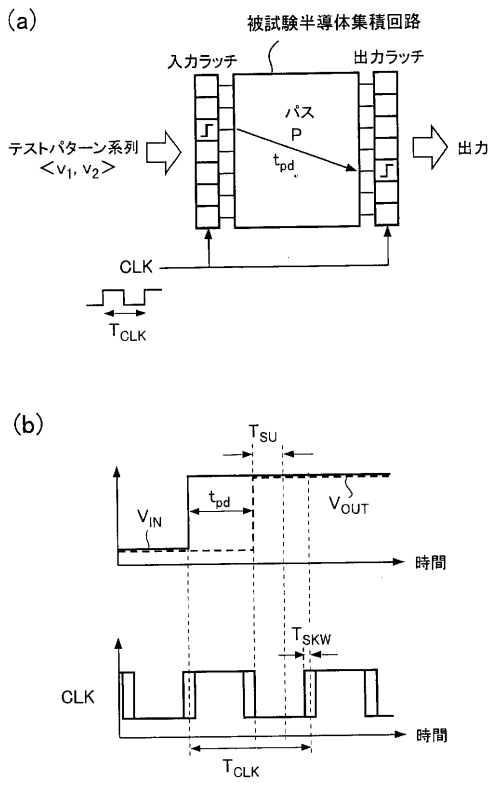
【図11】



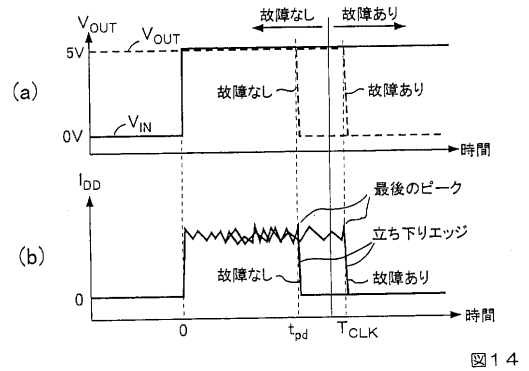
【図12】



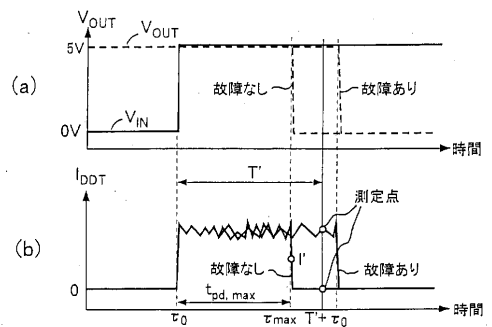
【図13】



【図14】



【図15】



故障なし: $i_{DDT}(T + \tau_0) \leq I'$

故障あり: $i_{DDT}(T + \tau_0) > I'$

図15

【 図 16 】

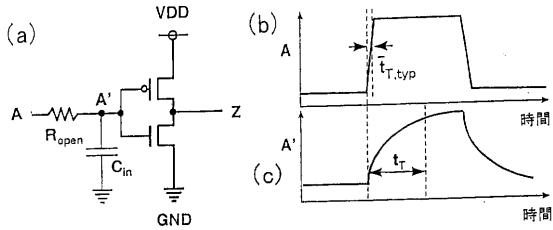


図 16

【 図 18 】

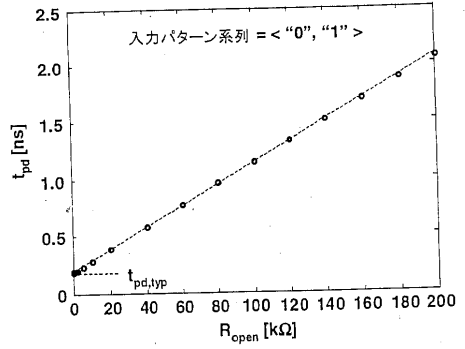


図 18

【 図 17 】

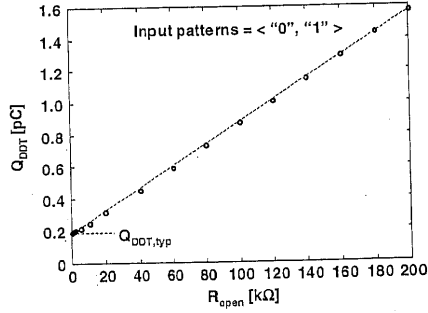


図 17

【 図 19 】

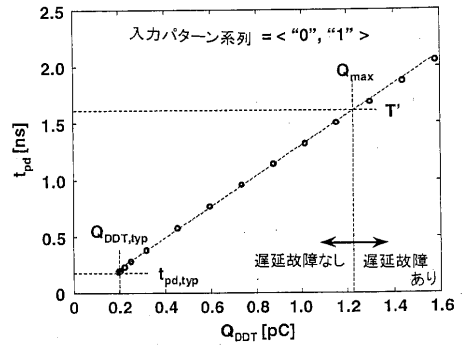
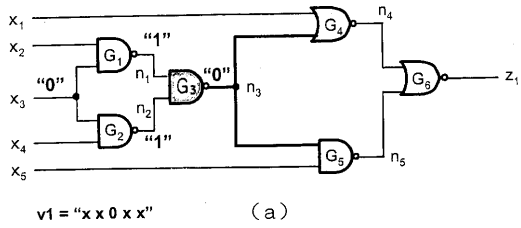
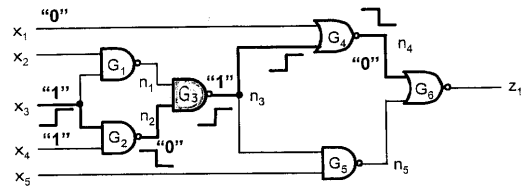


図 19

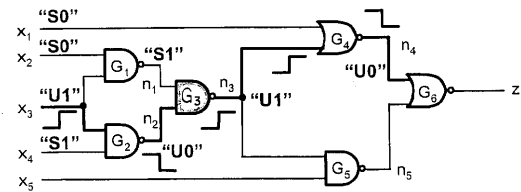
【 図 20 】



v1 = "x x 0 x x" (a)



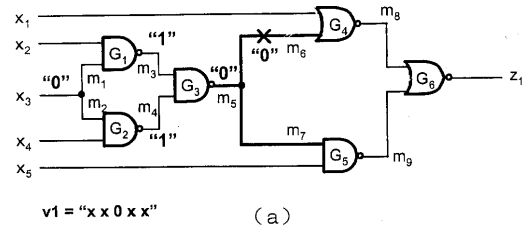
v2 = "0 x 1 1 x" (b)



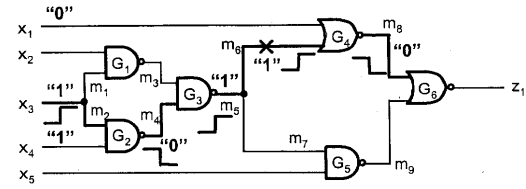
T' = "S0 S0 U1 S1 XX" = < "0 0 0 1 x", "0 0 1 1 x" > 図 20

(c)

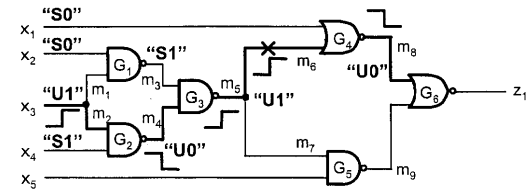
【 図 21 】



v1 = "x x 0 x x" (a)



v2 = "0 x 1 1 x" (b)



T' = "S0 S0 U1 S1 XX" = < "0 0 0 1 x", "0 0 1 1 x" > 図 21

(c)

【 図 2 2 】

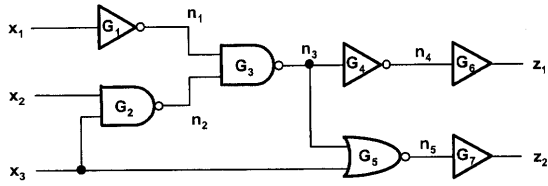


図 22

【 図 2 3 】

テスト/故障 シナリオ識別子	入力端子 x1, x2, x3	内部端子 n1, n2, n3, n4, n5	出力端子 z1, z2	検出可能 断線故障
T1	L L R	H H L H F	H F	GF
T2	L H R	H F R F F	F F	G2F, G3R, G4F, G5F
T3	H L R	L H H L L	L L	-
T4	H H R	L F H L L	L L	-
T5	L R L	H H L H H	H H	-
T6	L R H	H F R F L	F L	G2F, G3R, G4F
T7	H R L	L H H L L	L L	-
T8	H R H	L F H L L	L L	-
T9	R L L	F H R F F	F F	G1F, G3R, G4F, G5F
T10	R L H	F H R F L	F F	G1F, G3R, G4F
T11	R H L	F H R F F	F F	G1F, G3R, G4F, G5F
T12	R H H	F L H L L	L L	-
...

図 23

【 図 2 4 】

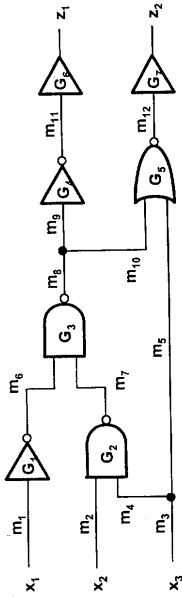


図 24

【 図 2 5 】

テスト/故障 シナリオ識別子	入力端子 x1, x2, x3	内部端子 (m)	出力端子	検出可能 断線故障
T1	L L R	1 2 3 4 5 6 7 8 9 10 11 12	1 2	m3, m5, m12
T2	L H R	L R R R H H H H H H H H	F F	m3, m4, m5, m7, m8, m9, m10, m11, m12
T3	H L R	L R R R L H L L L L L L	L L	-
T4	H H R	L R R R L F L L L L L L	L L	m3, m4
T5	L R L	L R L L H H H H H H H H	H H	-
T6	L R H	L R H H H F F F F F F L	F L	m2, m7, m8, m9, m11
T7	H R L	L L L H L L L L L L L L	L L	-
T8	H R H	L L L H L L F L L L L L	L L	m2
T9	R L L	L L L F F F F F F F F F	F F	m1, m4, m8, m9, m10, m11, m12
T10	R L H	L L H H F F F F F F F L	F L	m1, m8, m9, m10, m11
T11	R H L	L L L F H F F F F F F F	F F	m1, m8, m9, m10, m11, m12
T12	R H H	L L L F L L L L L L L L	L L	m1
...

図 25

【図26】

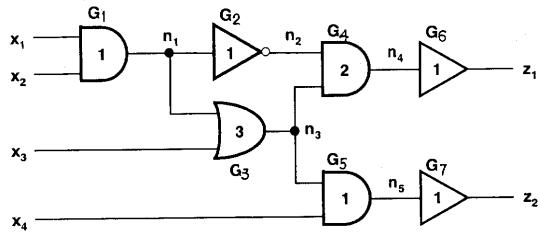


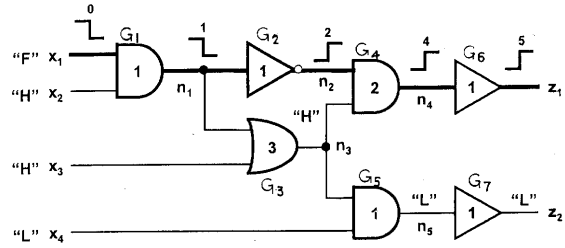
図26

【図27】

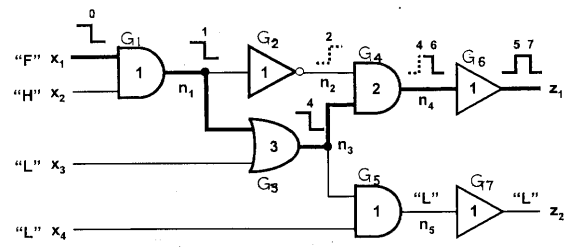
テストパターン シリーズ識別子	入力端子				内部番号							出力端子		検出可能 遅延故障パス
	x ₁	x ₂	x ₃	x ₄	n ₁	n ₂	n ₃	n ₄	n ₅	z ₁	z ₂	z ₃		
T1	F	H	H	L	F (1)	R (2)	F (1)	L	L	R (5)	L	R (6)	{x ₁ , n ₁ , n ₂ , n ₄ , z ₁ }	
T2	F	H	L	L	F (1)	R (2)	F (1)	L	L	R (5)	L	R (6)	{x ₁ , n ₁ , n ₃ , n ₄ , z ₁ }	
T3	F	H	L	H	F (1)	R (2)	F (1)	L	L	R (5)	L	R (6)	{x ₁ , n ₁ , n ₃ , n ₄ , z ₁ }, {x ₁ , n ₁ , n ₃ , n ₅ , z ₂ }	
T4	R	H	L	L	R (1)	F (2)	F (1)	L	L	L	L	L	-	
T5	R	H	L	H	R (1)	F (2)	F (1)	L	L	L	L	R (6)	{x ₁ , n ₁ , n ₃ , n ₅ , z ₂ }	
...	

図27

【図28】



(a)



(b)

図28

【図29】

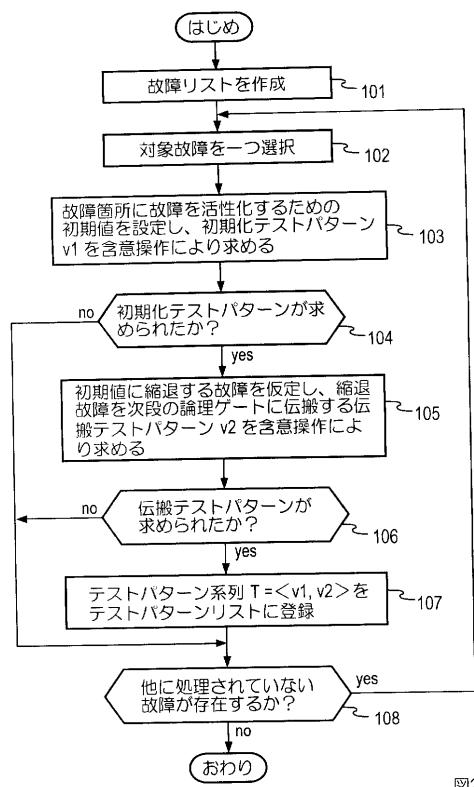


図29

【 図 3 0 】

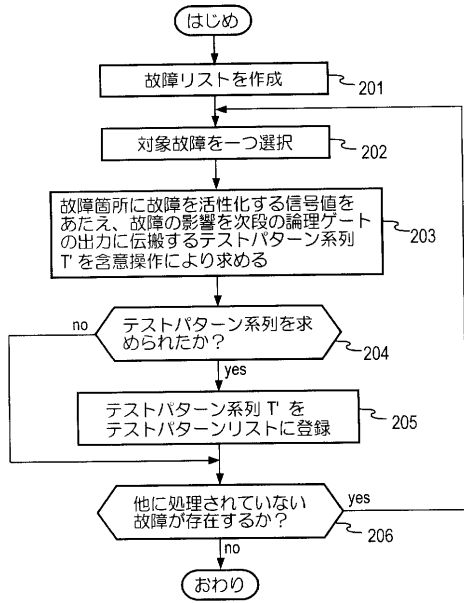


図30

【 図 3 1 】

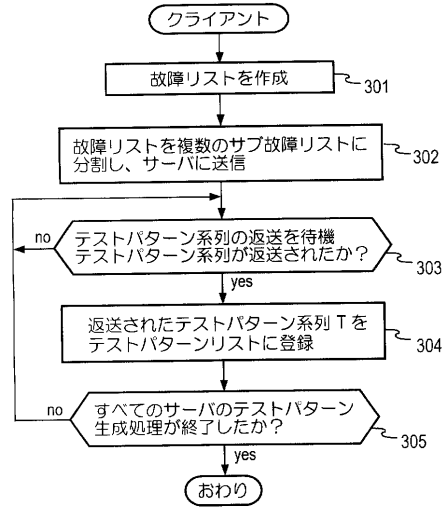


図31

【 図 3 2 】

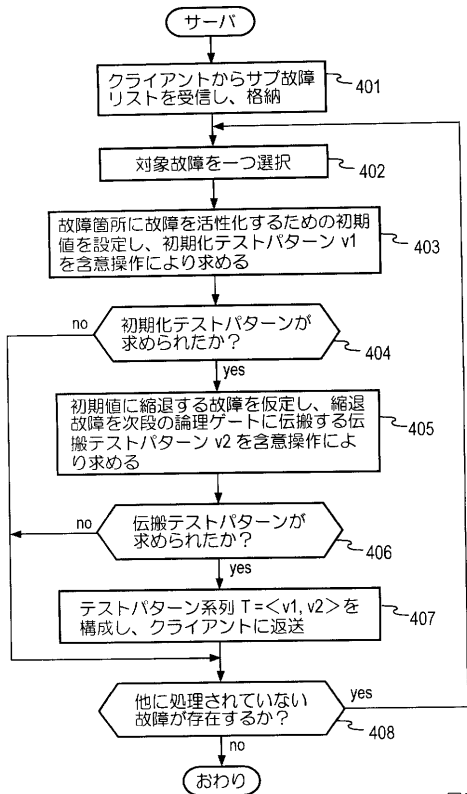


図32

【 図 3 3 】

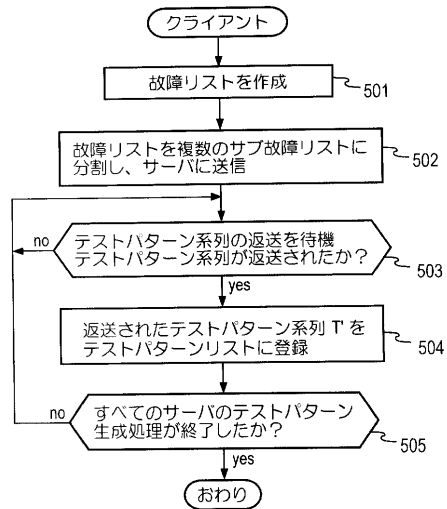


図33

【図34】

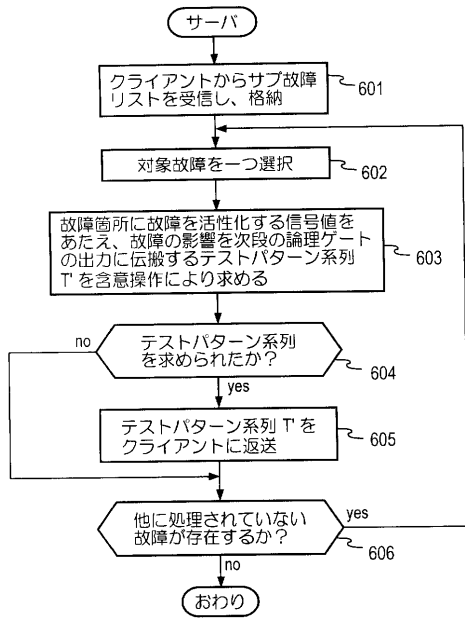


図34

【図35】

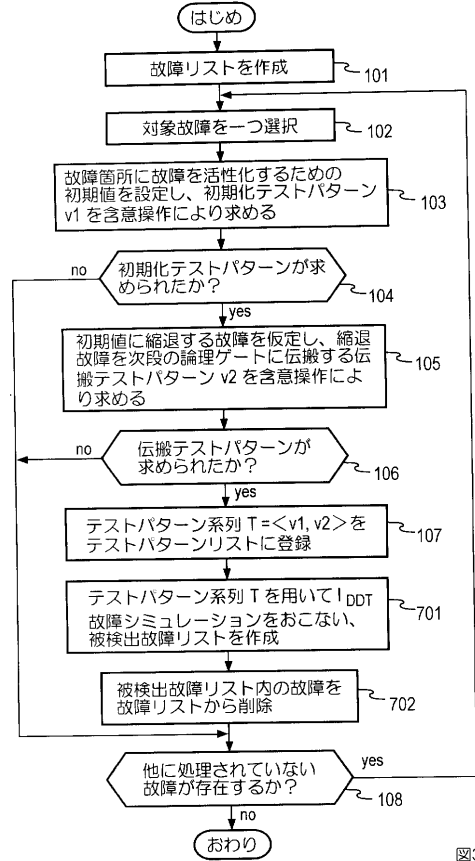


図35

【図36】

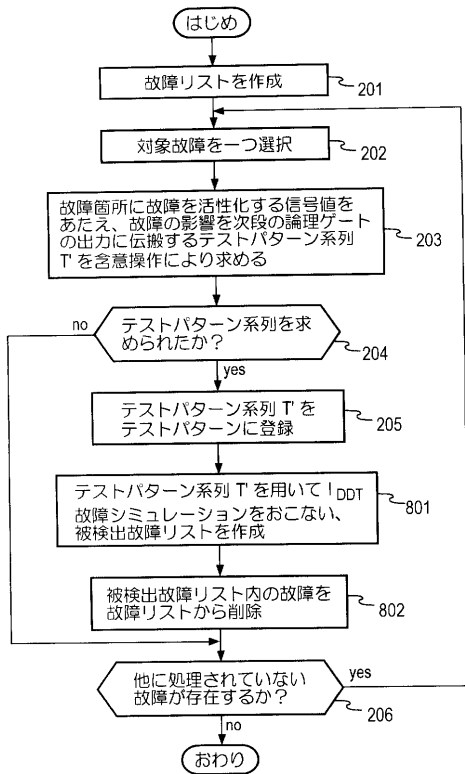


図36

【図37】

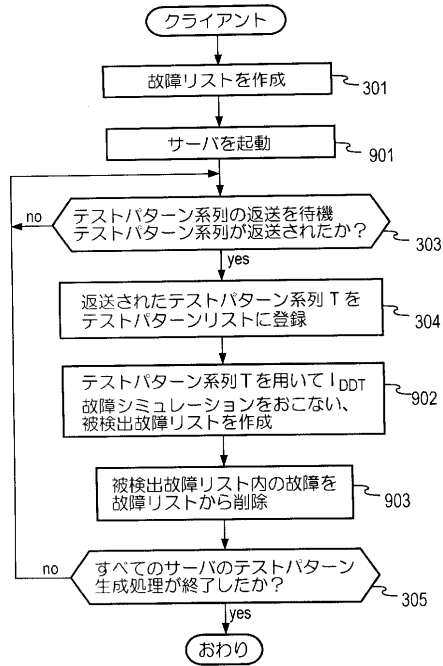


図37

【図38】

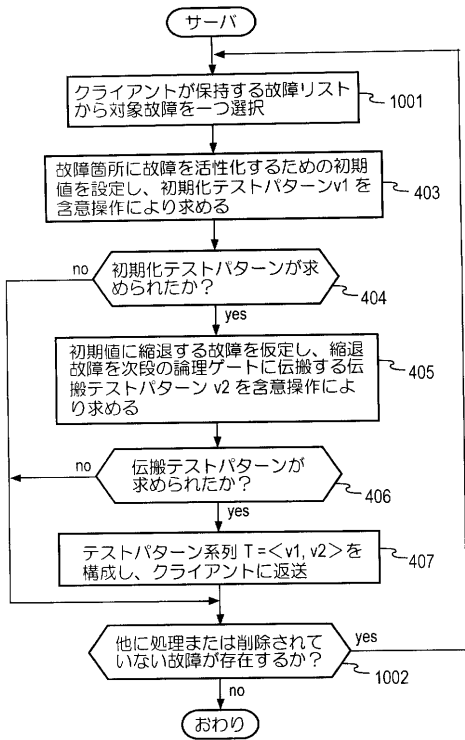


図38

【図39】

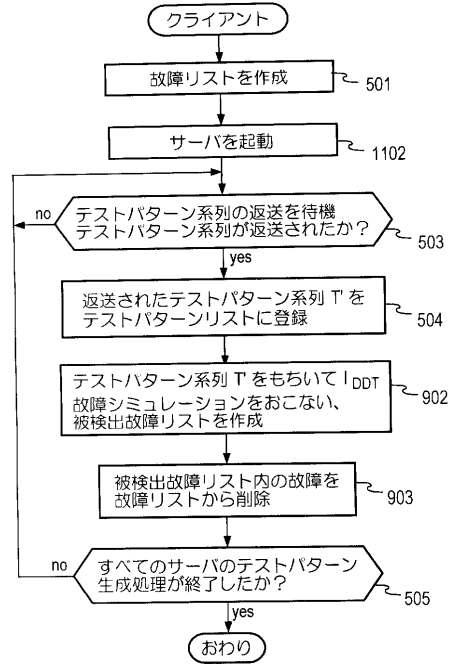


図39

【図40】

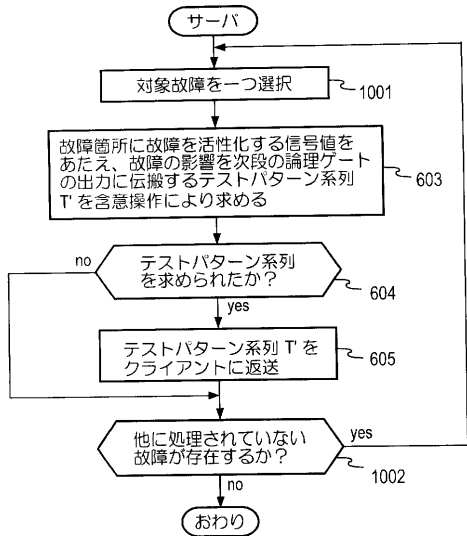


図40

【図41】

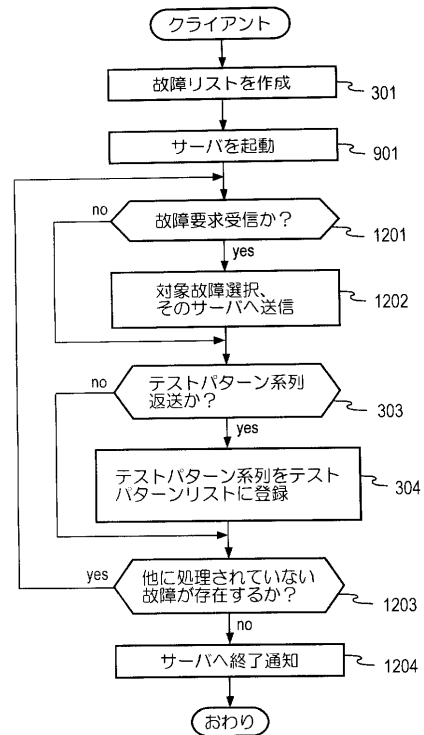


図41

【図42】

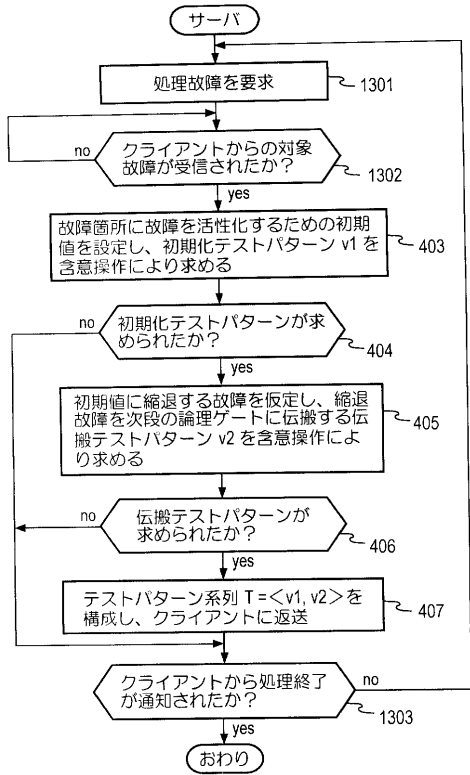


図42

【図43】

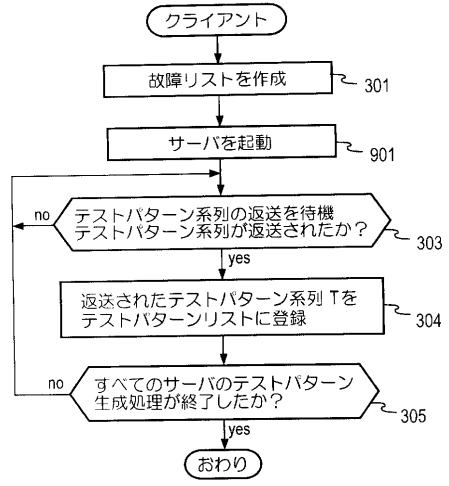


図43

【図44】

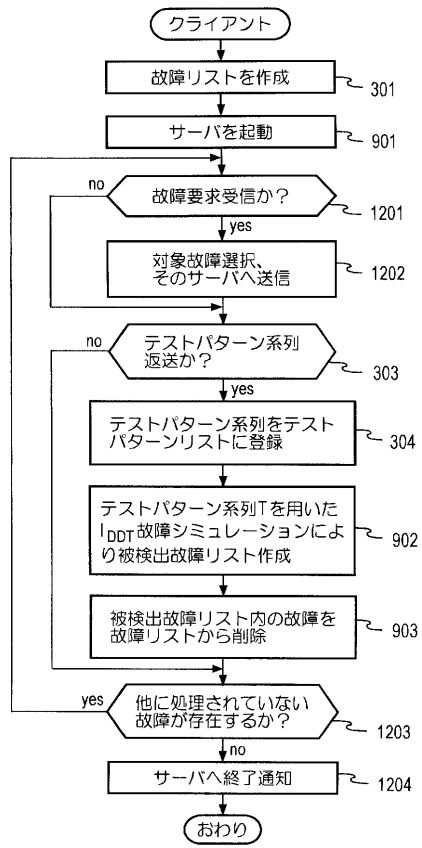


図44

【図45】

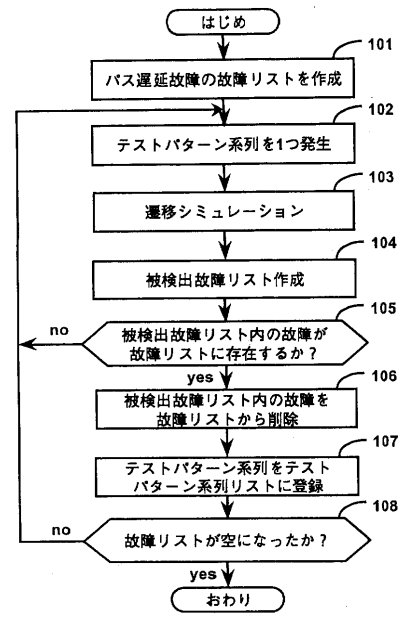


図45

【図46】

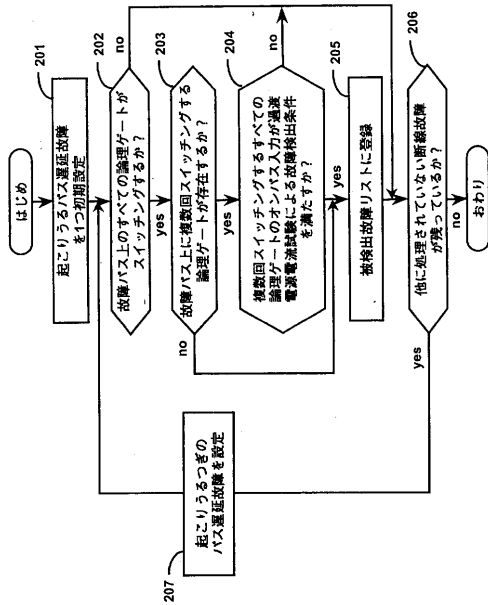


図46

【図47】

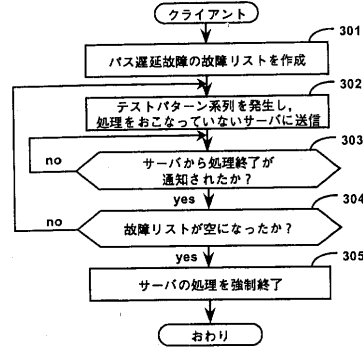


図47

【図48】

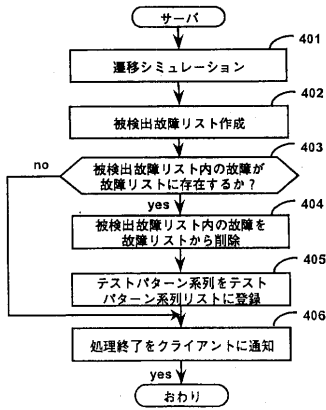


図48

【図49】

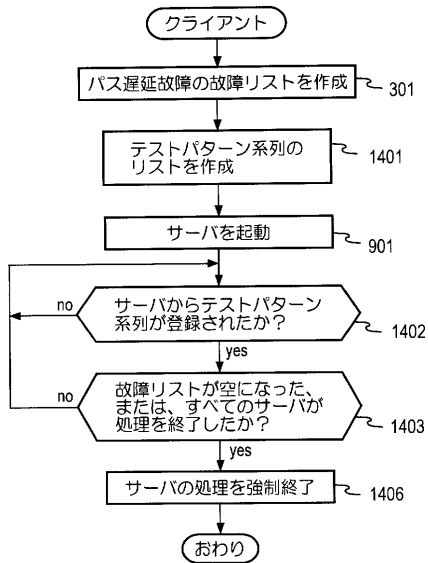


図49

【図50】

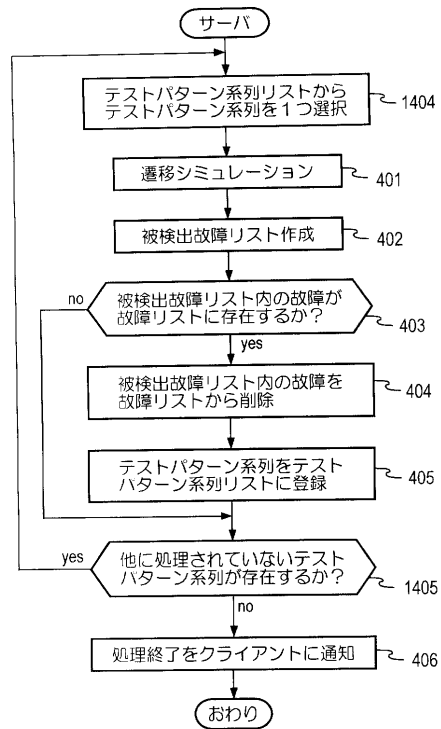


図50

【 図 5 1 】

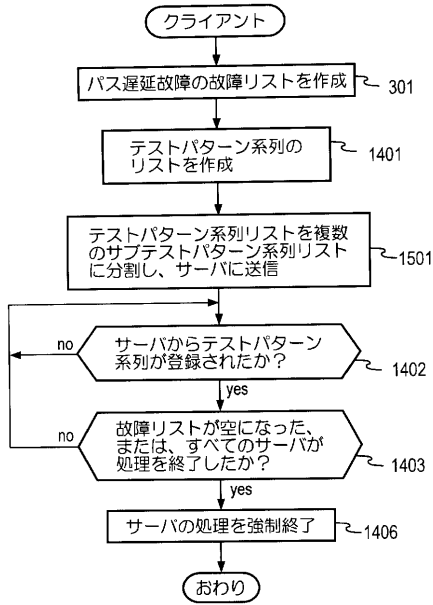


図51

【 図 5 2 】

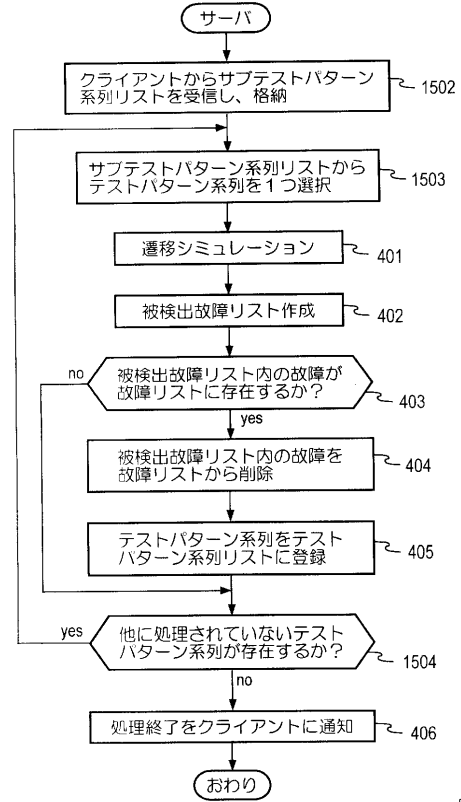


図52

【 図 5 3 】

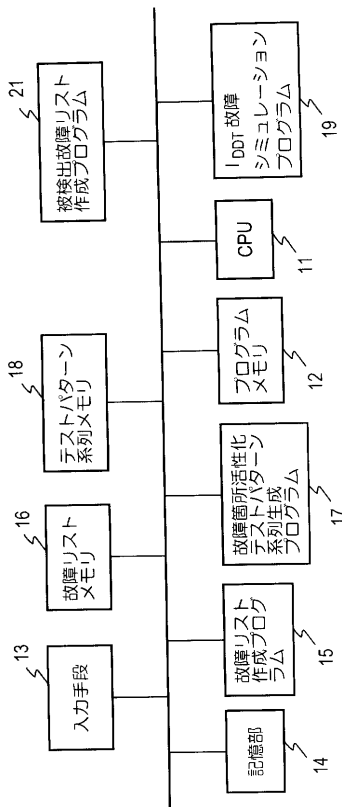


図53

【 図 5 4 】

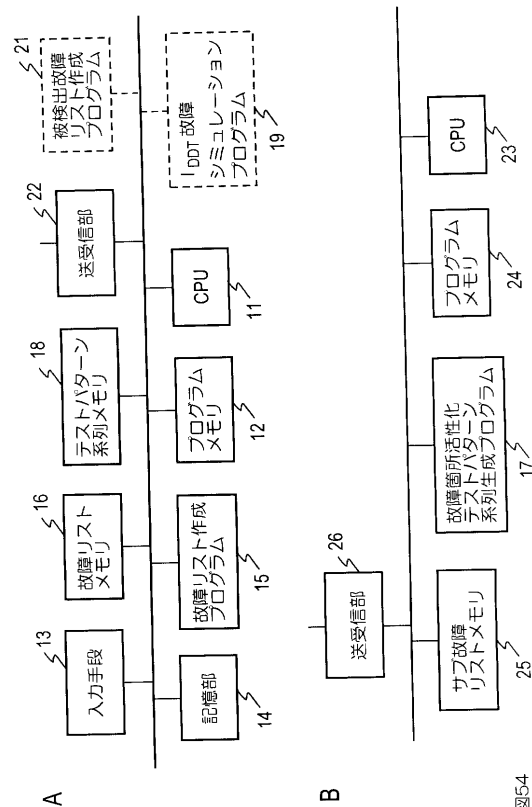


図54

【図55】

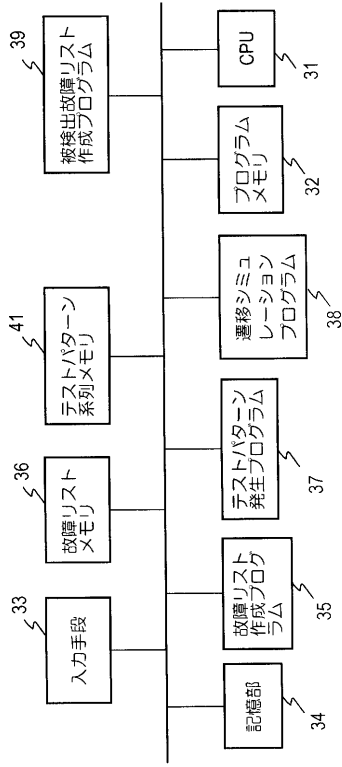


図55

【図56】

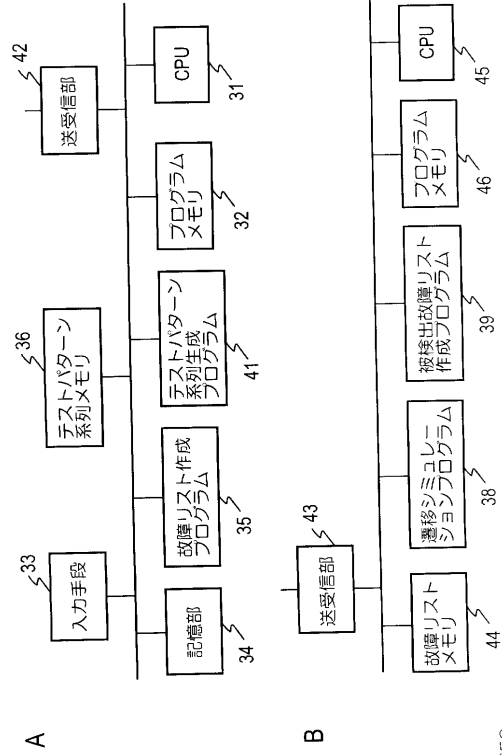


図56

フロントページの続き

(72)発明者 山口 隆弘

東京都練馬区旭町1丁目3番1号 株式会社アドバンテスト内

審査官 神谷 健一

(56)参考文献 特開平02-276980(JP,A)

特開平03-120485(JP,A)

特開平04-293165(JP,A)

特開平04-344481(JP,A)

特開平05-072287(JP,A)

特開平08-084067(JP,A)

特開平09-080114(JP,A)

特開平09-145800(JP,A)

特開平09-269959(JP,A)

特開平10-247208(JP,A)

国際公開第92/001943(WO,A1)

B. KRUSEMAN, et al., Transient current testing of 0.25 μm CMOS devices, Proceedings of International Test Conference, 1999, 米国, IEEE, 1999年, pp. 47-56

CHIN JEN LIN, et al., On Delay Fault Testing in Logic Circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 米国, IEEE, 1987年, Vol. CAD-6, No. 5, pp. 694-703

M. SACHDEV, et al., Defect detection with transient current testing and its potential for deep sub-micron CMOS ICs, Proceedings of International Test Conference, 米国, IEEE, 1998年, pp. 204-213

A. WALKER, et al., An approach for detecting bridging fault-induced delay faults in static CMOS circuits using dynamic power supply current monitoring, IEEE International Workshop on IDDQ Testing, 1997. Digest of Papers., 米国, IEEE, 1997年, pp. 73-77

WANG JIANG, et al., IC test using the energy consumption ratio, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 米国, IEEE, 2000年 1月, VOL. 19, NO.1, pp. 129-141

J. F. FRENZEL, et al., A comparison of methods for supply current analysis, Proceedings of the IEEE 1991 Custom Integrated Circuits Conference, 米国, IEEE, 1991年, pp. 13.3.1-13.3.4

(58)調査した分野(Int.Cl., DB名)

G01R 31/28-31/3193

G06F 11/22-11/26

G06F 17/50

H01L 21/82

H01L 27/04

H03K 19/00

H03K 19/01-19/082

H03K 19/09

H03K 19/094-19/096