

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3937034号

(P3937034)

(45) 発行日 平成19年6月27日(2007.6.27)

(24) 登録日 平成19年4月6日(2007.4.6)

(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 G

GO 6 F 11/22 (2006.01)

GO 1 R 31/28 M

GO 6 F 11/22 3 1 O C

GO 6 F 11/22 3 6 O P

請求項の数 7 (全 31 頁)

(21) 出願番号 特願2000-378423 (P2000-378423)
 (22) 出願日 平成12年12月13日(2000.12.13)
 (65) 公開番号 特開2002-181905 (P2002-181905A)
 (43) 公開日 平成14年6月26日(2002.6.26)
 審査請求日 平成17年3月14日(2005.3.14)

(73) 特許権者 000005108
 株式会社日立製作所
 東京都千代田区丸の内一丁目6番6号
 (73) 特許権者 000233295
 日立情報通信エンジニアリング株式会社
 神奈川県横浜市戸塚区戸塚町393番地
 (74) 代理人 110000350
 ポレール特許業務法人
 (74) 代理人 100068504
 弁理士 小川 勝男
 (74) 代理人 100086656
 弁理士 田中 恭助
 (74) 代理人 100094352
 弁理士 佐々木 孝

最終頁に続く

(54) 【発明の名称】 半導体集積回路のテスト方法及びテストパターン発生回路

(57) 【特許請求の範囲】

【請求項1】

一定のビット数及び時刻数のパターン列が複数個集まったパターン列群を1回以上発生する回路であって、

上記パターン列群内のパターン列は全て同一である同一パターン列発生部と、前記同一パターン列発生部から発生されたパターン列群を入力とし、前記パターン列群内のパターン列における一部のビットを反転させ、パターン列群及びパターン列群内パターン列番号及びパターン列内時刻に応じて、反転するパターン内ビット位置を変更する回路であるビット反転制御部とを有することを特徴とするテストパターン発生回路。

【請求項2】

請求項1において、前記同一パターン列発生部が線形フィードバックシフトレジスタと、前記パターン列群の1つを発生する間、前記線形フィードバックシフトレジスタ内のレジスタの初期値を保持し続けるレジスタを少なくとも1つ有することを特徴とするテストパターン発生回路。

【請求項3】

請求項1又は2において、前記ビット反転制御部が、入力されるパターン列群内で、反転ビットのないパターン列、1パターンの一部又は全部のビットが反転したパターン列、連続又は所定のパターン数分の間隔があいた複数パターンの一部又は全部のビットが反転したパターン列の複数パターン列の全て、あるいは上記複数パターン列の一部のパターン列を含むようにビット反転を制御する制御回路を有することを特徴とするテストパターン

10

20

発生回路。

【請求項 4】

請求項 1 ないし 3 のいずれか 1 つに記載されたテストパターン発生回路又は前記ビット反転制御部と、テスト対象回路とが集積され、前記テスト対象回路内に貼られたスキランチェーンの入力信号線又はテスト対象回路への外部入力信号線と前記テストパターン発生回路又は前記ビット反転制御部の出力信号線が接続されたことを特徴とする半導体集積回路。

【請求項 5】

スキランチェーンの入力信号線又は外部入力信号線を持つテスト対象回路にテストパターンの信号を加え、上記テスト対象回路の応答パターンを期待値と比較する半導体集積回路のテスト方法であって、上記テストパターンの信号の発生ステップとして、一定のビット数及び時刻数のパターン列が複数個集まったパターン列群内のパターン列が全て同一である同一パターン列を発生するステップと、

前記同一パターン列のパターン列群のパターン列における一部のビットを反転させ、パターン列群及びパターン列群内パターン列番号及びパターン列内時刻に応じて、反転するパターン内ビット位置を変更するステップを有することを特徴とする半導体集積回路のテスト方法。

【請求項 6】

請求項 5 記載のテスト方法において、前記同一パターン列を発生するステップの同一パターン列生成方法が、

仮定した故障を検出するテストパターンの集合を生成する第一のステップと、前記テストパターンの集合をパターン列の集合に変換して考え、設定すべきパターン列内時刻とパターン内ビット位置の組の集合が一致し、かつ、所定のハミング距離以下のパターン列同士を同じクラスタとなるように前記パターン列の集合を分類する第二のステップと、前記パターン列のクラスタそれぞれに対し、属するパターン列からパターン列内時刻及びパターン内ビット位置ごとの多数決により求まるパターン列を生成する第三のステップを有することを特徴とする半導体集積回路のテスト方法。

【請求項 7】

請求項 5 記載のテスト方法において、前記同一パターン列を発生するステップが、仮定した故障を検出するテストパターンの集合を生成する第一のステップと、前記テストパターンの集合をパターン列の集合に変換して考え、設定すべきパターン列内時刻とパターン内ビット位置の組の集合が一致し、かつ、所定のハミング距離以下のパターン列同士を同じクラスタとなるように前記パターン列の集合を分類する第二のステップと、前記パターン列のクラスタそれぞれに対し、属するパターン列からパターン列内時刻及びパターン内ビット位置ごとの多数決により求まるパターン列を生成する第三のステップと、前ステップで求めた各パターン列をLFSRのレジスタ初期値に変換する第四のステップと、前ステップで求められたLFSRレジスタ初期値の中で、パターン列群に展開すると新規に故障を検出できるあるいはテストパターンを含むものを選択する第五のステップを有することを特徴とする半導体集積回路のテスト方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、半導体集積回路のテスト方法及びそのテスト方法に使用するテストパターン発生回路並びに半導体集積回路に関するものである。

【0002】

【従来の技術】

半導体集積回路の良否を検査するテスト方法として、ストアドテスト方式と組込み自己テスト (Built-In Self-Test、BIST) 方式が代表的である。ストアドテスト方式は、仮定した故障に対しアルゴリズムによりテストパターンを求め、それを記憶したテスト装置によりテスト対象回路に印加してテスト対象回路からの応答パターンを期待値と比較するテ

10

20

30

40

50

スト方法である。BIST方式は、擬似乱数パターン発生器と符号圧縮器を半導体集積回路に内蔵して、テスト対象回路に大量の擬似乱数パターンを与え、応答パターンの圧縮結果を期待値と比較するテスト方法である。

【0003】

ストアドテスト方式では、大規模のテスト対象回路に対し高い故障検出率を得るにはテストパターン数あるいはテストデータ量が増大し、半導体集積回路テスト装置に記憶しきれないという問題がある。BIST方式では、線形フィードバックシフトレジスタ(LFSR)で発生される擬似乱数パターンを用いるため、テストデータ量は少ないが、大規模なテスト対象回路に対しては限られた数の擬似乱数パターンでは高い故障検出率が得られないという問題がある。

10

【0004】

また、BIST方式の故障検出率を改善する案が多数提案されている。特開平10-197601号公報や特開平11-142481号公報に記載された検査点挿入方式では、検査点と呼ぶ回路をテスト対象回路中に付加することにより乱数パターンでも高い故障検出率を得る。文献Proceeding of Design Automation Conference 97 (1997年) 472-477頁に掲載されたK. H. Tsai等の論文“STARBIST: Scan Autocorrelated Random Pattern Generation”で記述された方式では、基準となる1つのパターンに対し1ビットずつ反転した近傍パターンを特定の確率で発生するために、LFSRによる乱数パターンの重みを制御する回路とビット反転を制御する回路をスキャンチェーンの途中に付加し、故障を効率的に検出する。文献Proceeding of International Test Conference 98 (1998年)1057-1064頁に掲載されたG. Kiefer等の論文“Deterministic BIST with Multiple Scan Chains”で記述された方式では、LFSRで発生した擬似乱数パターンを類似したテストパターンに変更するために一部のビットを反転させる論理を付加する。文献Proceeding of International Test Conference 92 (1992年)120-129頁に掲載されたS. Hellebrandの論文“Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift registers”で記述されたReseeding方式では、LFSRの初期値(seedと呼ぶ)を発生したいテストパターンから計算し、そのseedを次々に入れ換える。

20

【0005】

【発明が解決しようとする課題】

上述のBIST方式の故障検出率を改善するいずれの方法も大規模な半導体集積回路に適用するには課題がある。まず、検査点挿入方式では、テスト対象の半導体集積回路(以下テスト対象回路とも略称)中のパスに検査点を挿入するため回路の動作速度を遅くするオーバーヘッドがあるという問題に加え、テスト対象回路に依存して検査点を変更する必要があるため、検査点の挿入位置が決まらなるとテスト対象回路のレイアウトや配線が局所的にも決まらなるという半導体集積回路の設計期間が増大するという問題がある。

30

【0006】

また、K. H. Tsai等の方法やG. Kiefer等の方法では、テスト対象回路に対し、時間のかかるテストパターン生成処理結果に依存してビット反転等を制御する回路やスキャンチェーンを変更しなければレイアウトや配線ができないという設計期間が増大するという問題のほかに、テスト対象回路が大規模回路になると、H. Tsai等の方法ではスキャンチェーンの貼り方の制限による配線オーバーヘッドの増加、G. Kiefer等の方法ではビット反転を制御する回路の増大がある。さらに、Reseeding方式では、オリジナルのBIST方式と比べてハード面でのオーバーヘッドや設計期間の増大の問題はないが、seed数がストアドパターン数と同等以上になることが予想され、BIST方式本来の目的であるテストデータ量削減の効果が小さいという問題がある。

40

【0007】

本発明の主な目的は、少ないテストデータ量で高い故障検出率を得る半導体集積回路等のテスト方法、特にBIST方式を用いたテスト方法を実現することである。

本発明の他の目的は、上記テスト方法を実施するため、付加する回路がテスト対象回路に依存せず、テストのための回路設計が容易な半導体回路装置を実現することである。

50

本発明の更に他の目的は、上記テスト方法を実施するため付加する回路による動作速度のオーバーヘッドがなく、ゲートや配線のハード的なオーバーヘッドも小さいテストのための回路装置を実現することである。

【0008】

【課題を解決するための手段】

上記目的を達成するため、本発明の半導体集積回路のテスト方法は、擬似乱数パターンであるテストパターン信号（以下単にテストパターンと呼ぶ）をテスト対象回路に加え、そのテスト対象回路の応答パターンを期待値と比較するテスト方法であって、

上記テストパターンを発生するステップが、一定のビット数及び時刻数のパターン列が複数個集まったパターン列群であって、そのパターン列群内のパターン列は全て同一である同一パターン列であるパターン列群を一回以上発生する第1ステップと、前記パターン列群内のパターン列における一部のビットを反転させる第2ステップと、前記第2ステップでパターン列群、パターン列群内パターン列番号、パターン列内時刻に応じて、反転するパターン内ビット位置を変更する第3ステップとを有する。

10

【0009】

また、本発明の半導体集積回路テスト方法の好ましい実施形態では、スキャンチェーン本数や外部入力端子数から定まるビット数及び最大スキャンチェーン長や単位テスト系列長から定まる時刻数のパターン列が複数個集まったパターン列群を1回以上テスト対象回路に印加し、パターン列群内では1つの基準となるパターン列が存在し、その基準となるパターン列に対し、反転ビットのないパターン列、1パターンの一部又は全部のビットが反転したパターン列、連続あるいは所定のパターン数分の間隔があいた複数パターンの一部又は全部のビットが反転したパターン列の、全て、あるいは一部のパターン列を用いる。

20

【0010】

また、上記半導体集積回路のテスト方法を実施するため、本発明のテストパターン発生回路は、一定のビット数及び時刻数のパターン列が複数個集まったパターン列群を1回以上発生する回路で、そのパターン列群内のパターン列は全て同一である同一パターン列発生部と、前記同一パターン列発生部から発生されたパターン列群を入力とし、前記パターン列群内のパターン列における一部のビットを反転させ、パターン列群及びパターン列群内パターン列番号及びパターン列内時刻に応じて、反転するパターン内ビット位置を変更する回路であるビット反転制御部とをもつ。

30

【0011】

本発明の好ましい実施形態では、上記パターン発生回路とテスト対象回路とを単一の半導体集積回路として構成する場合、上記パターン発生回路のビット反転制御部とテスト対象回路とを単一の半導体集積回路として構成する場合、上記パターン発生回路をテスト対象回路と独立した半導体集積回路として構成する場合があり、上記テスト対象回路と分離されたパターン発生部、あるいはパターン発生回路は、これらを含む半導体回路テスト装置を構成し、半導体回路テスト装置を構成するプローブを上記テスト対象回路の外部入力端子に接続してテストを実行する。

【0012】

また、同一パターン列発生部は、線形フィードバックシフトレジスタと、1つのパターン群を発生する間、その線形フィードバックシフトレジスタ内の初期値を保持するレジスタを持ち、更に、線形フィードバックシフトレジスタ内のレジスタに、シリアル又はパラレルにコピーする機能を持つ。また、前記複数のパターン列群を発生するため、前記線形フィードバックシフトレジスタ内のレジスタの初期値を複数個保存するレジスタを設ける場合もある。

40

【0013】

【発明の実施の形態】

以下、本発明の実施の形態について、図面を参照し説明する。

【0014】

図1は、本発明によるテスト方法を実施する回路の一実施例の構成を示す。

50

【 0 0 1 5 】

本テスト方法を実施するためには回路は、同一パターン列発生部 1 1 0 と、ビット反転制御部 1 2 0 と、テスト対象回路 1 3 0 から構成される。同一パターン列発生部 1 1 0 の出力線群 P T 1、P T 2 ... P T n は、ビット反転制御部 1 2 0 に入力され、ビット反転制御部 1 2 0 の出力線群は、テスト対象回路 1 3 0 の入力端 I N 1、I N 2 ... I N n に入力される。

【 0 0 1 6 】

同一パターン列発生部 1 1 0 は、クロックに同期して出力線数 n をビット幅とするパターンを出力する。同一パターン列発生部 1 1 0 が出力するパターンで、所定の時刻数分のパターンをパターン列と呼ぶとき、同一のパターン列を複数個順次出力し、この複数個の同一のパターン列をパターン列群と呼び、ビット反転制御部 1 2 0 から異なるパターン列群を順次出力する。

10

【 0 0 1 7 】

ビット反転制御部 1 2 0 は、入力されるパターン列群の各パターンに対し、パターン列番号やパターン列内時刻に応じて、パターン中の一部のビットを反転する回路である。ビット反転制御部 1 2 0 は、各時刻に対し入力パターン中のビット反転位置を表現したパターン（ビット反転位置に対応するビットのみ論理値 1）を出力する反転制御回路 1 2 1 と、ビット反転制御部 1 2 0 の入力パターンの各ビットを反転制御回路 1 2 1 の出力したパターンの各ビットに応じて排他的論理和を得る回路 1 2 2 ~ 1 2 4 をもつ。

【 0 0 1 8 】

さらに、反転制御回路 1 2 1 の出力するパターン列群は、全て論理値 0 の成分をもつパターン列と、1パターンだけ一部又は全部のビットが論理値 1 であるパターン列と、連続あるいは所定のパターン数分の間隔があいた複数パターンの一部又は全部のビットが論理値 1 であるパターン列の、全て、あるいは一部のパターン列を含む。このとき、ビット反転制御部 1 2 0 の出力する各パターン列群は、1つの基準となるパターン列に対し、反転ビットのないパターン列、1パターンの一部又は全部のビットが反転したパターン列、連続あるいは所定のパターン数分の間隔があいた複数パターンの一部又は全部のビットが反転したパターン列の、全て、あるいは一部から構成される。テスト対象回路 1 3 0 は、論理設計者が設計した回路の中でテストの対象にする半導体集積回路等である。

20

【 0 0 1 9 】

図 2 (a) は、上記テスト対象回路の第一の回路例を示す。第一の回路例は、フルスキャン設計されたテスト対象回路 2 0 0 であり、テスト対象回路 2 0 0 中の全ての記憶素子（2 1 1、2 1 2 ... 2 3 3）に対し、テスト中に設定、読み出しが可能となるようにスキャン機能をつけ、テスト対象回路 2 0 0 を組合せ回路としてテストする。図 2 (b) は上記スキャン機能付き記憶素子の回路 2 4 0 を示す。クロック入力 C、データ入力 D、出力 Q を持つ記憶素子 2 4 1 に対し、セレクト 2 4 2 を付加し、入力線 SE の入力が論理値 0 のときは、入力線 D の入力の値を格納する通常動作で、入力線 SE の入力が論理値 1 のときは入力線 S I の入力値を格納するスキャン動作となる。

30

【 0 0 2 0 】

図 2 (a) に戻って、スキャン機能付き記憶素子 2 1 1 ~ 2 1 3、2 2 1 ~ 2 2 3、2 3 1 ~ 2 3 3 は、それぞれスキャンチェーン 2 0 1、2 0 2、2 0 3 上に直列に接続される。なお、図には記憶素子のみ記載しており、通常動作で使われる組合せ回路部分は省略している。外部入出力端子にはバウンダリスキャン記憶素子が挿入されている（図示せず）。テスト対象回路 2 0 0 の動作は、端子 2 0 4 のスキャンイネーブル信号 SEN が 0 のとき端子 2 0 5 のクロック CLK を叩けば回路本来の通常動作となり、信号 SEN が 1 のときクロック CLK を叩けば各スキャンチェーン 2 0 1、2 0 2、2 0 3 上でスキャンシフト動作となる。テスト対象回路 2 0 0 の入力 IN1、IN2、...、INn が有効となるのは、スキャンシフト動作のときである。

40

【 0 0 2 1 】

図 2 (c) は、上記テスト対象回路の第二の回路例を示す。第二の回路例は、ノンスキャン

50

設計されたテスト対象回路260であり、記憶素子251、252、261、262、271、272等にはスキャン機能を付けず、外部入力端子と等価な信号線をテスト対象回路260の入力INnとする。この場合、テスト対象回路260は順序回路としてテストする。

【0022】

上述のように、本実施例によれば、フルスキャン設計又はノンスキャン設計のいずれの回路に対しても、同一パターン列発生部110とビット反転制御部120を備えるパターン発生回路により、テストパターンとして故障検出に有効とされる近傍パターン、即ち、基準となる1つのパターンに対し数ビット反転したパターンを発生することができるので、Re seeding方式と組み合わせることにより、BIST方式で高い故障検出率を得ることができる。

10

【0023】

ここで、同一パターン列発生部110の構成例の一部である線形フィードバックシフトレジスタ(以下、LFSRと略称)を先に説明する。

図3は、線形フィードバックシフトレジスタの回路例で、従来知られているパターン発生器のものと同一である。LFSR300は、シフトレジスタとして動作する記憶素子群301~303と、最下段の記憶素子303の値と特定の記憶素子の値との排他的論理和を得る回路304、305の出力をセレクタ306を通して最上段の記憶素子301にフィードバックする。なお、記憶素子301~303はクロック入力に変化した時にデータ入力を取り込むエッジトリガ型とする。以下の説明では、LFSR300内のシフトレジスタの状態をseedと呼ぶ。

20

【0024】

図9の(a)は、LFSR300の動作モードを示す。入力INTSELを論理値1のとき初期設定モードと呼び、記憶素子群301~303は入力BRCに同期してシフトし、入力SEEDINからseedの初期値を設定できる。入力INTSELが論理値0のときパターン発生モードと呼び、入力BRCに同期して、各記憶素子の出力PT1、PT2、...、PTnから疑似乱数パターンを発生する。

【0025】

LFSRの性質として、nビットのLFSRにおける記憶素子の出力をフィードバックに用いる排他的論理和の入力に用いるか否かを論理値0、1で表現したn個のビット列と、2の剰余系でn次多項式の1次からn次の係数(0又は1)を対応させたとき、そのn次多項式が原始多項式、即ち既約であるならば、nビットのLFSRはseedの成分が全て論理値0でないとき発生するパターンの周期は最大、即ち2のn乗から1を減じた数となる。例えば、2の剰余系で4次多項式 $X^4+X+1=0$ は原始多項式であるので、LFSR300が4ビットとすると、出力PT1とPT4の排他的論理和をフィードバックすれば、発生するパターンの周期は最大の15となる。

30

【0026】

図4、図5、図6、図7及び図8はいずれも、上記本発明の実施例における同一パターン列発生部110の回路例を示す。

図4は、同一パターン列発生部110の第一の回路構成例で、パラレルなseed復元が可能なLFSR400を示す。シフトレジスタとして動作する記憶素子群401~403と、最下段の記憶素子403の値と特定の記憶素子の値との排他的論理和404、405をセレクタ406を通して最上段の記憶素子401にフィードバックする部分の構成は、図3のLFSR300と同じである。seedを記憶するために、記憶素子群401~403にそれぞれ対応した記憶素子群407~409を備え、セレクタ410~412でseedの復元を制御する。

40

【0027】

LFSR400の動作モードを図9(b)に示す。入力INTSELが論理値1、入力RDSELが論理値0でクロックBRCが印加されるとき初期設定モードと呼び、記憶素子群401~403はクロックBRCに同期してシフトし、入力SEEDINからseedの初期値を設定する。そのseedを

50

記憶素子群 4 0 7 ~ 4 0 9 にパラレルにコピーする。入力 INTSEL が論理値 0、RDSEL が論理値 0 のときパターン発生モードと呼び、クロック BRC に同期して、各記憶素子 4 0 0 1 ~ 4 0 3 の出力 PT1、PT2、...、PTn から疑似乱数パターンを発生し、記憶素子群 4 0 7 ~ 4 0 9 の値は保持される。入力 INTSEL が 0、RDSEL が 1 のとき seed 復元モードと呼び、クロック BRC に同期して、記憶素子群 4 0 7 ~ 4 0 9 の値が記憶素子群 4 0 1 ~ 4 0 3 に seed としてパラレルに（一斉に）復元される。入力 INTSEL が 1、RDSEL が 0 でクロック BRC が印加されないとき、seed 更新モードと呼び、記憶素子群 4 0 1 ~ 4 0 3 の値を記憶素子群 4 0 7 ~ 4 0 9 にパラレルにコピーする。

【 0 0 2 8 】

図 5 は同一パターン列発生部 1 1 0 の第二の回路構成例で、シリアルな seed 復元が可能な LFSR 4 2 0 を示す。シフトレジスタとして動作する記憶素子群 4 2 1 ~ 4 2 3 と、最下段の記憶素子 4 2 3 の値と特定の記憶素子の値との排他的論理和回路 4 2 4、4 2 5 の出力をセレクタ 4 2 6 を通して先頭の記憶素子 4 2 1 にフィードバックする部分の構成は、前記 LFSR 3 0 0 と同じである。seed を記憶するために、記憶素子群 4 2 1 ~ 4 2 3 にそれぞれ対応した記憶素子群 4 2 7 ~ 4 2 9 を備え、セレクタ 4 3 0 で seed の復元を制御する。

【 0 0 2 9 】

LFSR 4 2 0 の動作モードを図 9 (c) に示す。入力 INTSEL が論理値 1、入力 RDSEL が論理値 0 でクロック BRC が印加されるとき初期設定モードと呼び、記憶素子群 4 2 1 ~ 4 2 3 はクロック BRC に同期してシフトし、入力 SEEDIN から seed の初期値を設定する。その seed を記憶素子群 4 2 7 ~ 4 2 9 にシリアルにコピーする。入力 INTSEL が論理値 0、入力 RDSEL が論理値 0 のときパターン発生モードと呼び、クロック BRC に同期して、各記憶素子の出力 PT1、PT2、...、PTn から疑似乱数パターンを発生し、記憶素子群 4 2 7 ~ 4 2 9 の値は保持される。入力 INTSEL が 0、入力 RDSEL が 1 のとき seed 復元モードと呼び、クロック BRC に同期して、記憶素子群 4 2 7 ~ 4 2 9 の値が記憶素子群 4 2 1 ~ 4 2 3 に seed としてシリアルに復元される。

【 0 0 3 0 】

図 6 は、同一パターン列発生部 1 1 0 の第三の回路構成例で、パラレルな seed 復元される複数の多項式に対応した LFSR 4 4 0 を示す。図では 4 ビットの例を示す。シフトレジスタとして動作する記憶素子群 4 4 1 ~ 4 4 4 と、最下段の記憶素子 4 4 4 の値と最上段の記憶素子 4 4 1 の値との排他的論理和回路 4 4 5 をセレクタ 4 4 6 を通して最上段の記憶素子 4 4 1 にフィードバックする構成と、seed を記憶するための記憶素子群 4 4 7 ~ 4 5 0 と、seed の復元を制御するセレクタ 4 5 1 ~ 4 5 4 は、前記 LFSR 4 0 0 と同じである。LFSR 4 4 0 では、排他的論理和 4 4 5 の入力に AND 素子 4 5 6 の出力端を接続し、記憶素子 4 5 5 で設定された値に応じてフィードバックに用いる排他的論理和の入力の一部をマスクする構成を追加した。

【 0 0 3 1 】

この構成により、LFSR に対応した 2 の剰余系の多項式を複数個もつことができる。図の例では、 $x^4+x+1=0$ (原始多項式) と、 $x^4+1=0$ (シフトレジスタ) である。また、LFSR 4 4 0 の動作は LFSR 4 0 0 の動作とほぼ同様で、異なる点は、初期設定モード時にフィードバックに用いる排他的論理和の入力を制御する記憶素子 4 5 5 の値を設定する必要があることである。

【 0 0 3 2 】

図 7 は、同一パターン列発生部 1 1 0 の第四の回路構成例で、パラレルな seed 復元及びシフトによるパターン発生を可能にした LFSR 4 6 0 を示す。LFSR 4 6 0 の 4 ビットのシフトレジスタ部分 4 6 1 ~ 4 6 4 及び seed 復元機能 4 6 7 ~ 4 7 0 の構成は前記 LFSR 4 4 0 と同じである。ただし、パターン発生モードで、記憶素子 4 7 5 の値が論理値 0 のときは記憶素子 4 6 1 と 4 6 4 の値の排他的論理和を記憶素子 4 6 1 にフィードバックするが、記憶素子 4 7 5 の値が論理値 1 のときは入力 SEEDIN からの値を記憶素子 4 7 5 をバイパスしてシフトレジスタ 4 6 1 ~ 4 6 4 に取り込む。これは、後で図 1 7 説明する複数個の LFSR をもつ同一パターン列発生部で、別の LFSR の状態を利用する時に用いる。

10

20

30

40

50

【 0 0 3 3 】

図 8 は、同一パターン列発生部 1 1 0 の第五の回路構成例で、シフトレジスタ群 4 8 0 を示す。記憶素子群 4 8 1、4 8 2、4 8 3 とセレクタ 4 8 4、記憶素子群 4 8 5、4 8 6、4 8 7 とセレクタ 4 8 8、記憶素子群 4 8 9、4 9 0、4 9 1 とセレクタ 4 9 2 は、入力 INTSEL が 1 のとき、入力 BRC に同期してシフトすることにより入力群 SEEDIN1、SEEDIN2、... SEEDINn から初期値を設定し、入力 INTSEL が論理値 0 のとき、それぞれがシフトレジスタとして動作して、出力 PT1、PT2、...、PTn からは最終記憶素子 4 8 3、4 8 7、4 9 1 の値からなるパターン信号を出力する。

【 0 0 3 4 】

以上の同一パターン発生部 1 1 0 の回路構成例 5 つに対して、その特長をまとめる。第一の回路構成例 LFSR 4 0 0 は、seed の復元が平行にできるためテスト時間を短くできる。第二の回路構成例 LFSR 4 2 0 は、seed の復元はシリアルなためテスト時間は長くなるがゲートオーバーヘッドが LFSR 4 0 0 より小さい。第三の回路構成例 LFSR 4 6 0 は、LFSR 多項式を複数もつことができる。第四の回路構成例 LFSR 4 8 0 は、複数個の LFSR をもつときに別の LFSR の状態を利用してパターンを発生できる。第五の回路構成例では、ゲートオーバーヘッドや全記憶素子に設定するためのデータ量は大きい、パターン列として全ての組合せを表現できる。

【 0 0 3 5 】

次に、反転制御回路 1 2 1 の詳細な構成について説明する。なお、パターン列内パターン数は 2 5 6 として説明する。

図 1 0 は、反転制御回路 1 2 1 の第一の回路構成例を示す。反転制御回路 6 0 0 では、1 パターンだけ全部のビットが論理値 1 であるパターン列を 2 5 6 通り出力する。反転制御回路 6 0 0 は、出力 C1-C8 をもつパターン列内時刻用 8 bit カウンタ 6 0 1 と出力 C1-C8 をもつパターン列番号用 8 bit カウンタ 6 0 2 と、2 つのカウンタ 6 0 1 及び 6 0 2 の値が一致しているか否かを求める比較器 6 0 4 を備え、その出力はマスク用の AND 素子 6 0 5 を通して n 個の出力 RVS1、RVS2、...、RVS_n に分配される。クロック BRC は、パターン列内時刻用 カウンタ 6 0 1 のクロック C として使われ、パターン列番号用 カウンタ 6 0 2 のクロック C はクロック BRC と入力 HCCKEN との論理積素子 6 0 3 で供給される。

【 0 0 3 6 】

ここで、反転制御回路 6 0 0 に使われる n ビットカウンタの回路例を図 1 4 (a) に示す。n ビットカウンタ 7 0 0 は、カウンタの状態を表す記憶素子 7 0 1 ~ 7 0 4 と、排他的論理和を求めるための素子 7 1 2 ~ 7 1 4 と、カウンタの 0 リセットをする素子 7 2 1 ~ 7 2 4 と、カウンタとしての動作かシフト動作かを選択するセレクタ 7 3 1 ~ 7 3 4 から構成され、カウンタの状態を n 個の出力 C1、C2、...、Cn によって出力する。

【 0 0 3 7 】

図 1 4 (b) は、カウンタ 7 0 0 の動作モードを示す。入力 SFTEN が論理値 1、入力 R が論理値 0 のときシフトモードと呼び、入力 SFTIN よりクロック C に同期してカウンタ内記憶素子 7 0 1 ~ 7 0 4 をシフトすることでカウンタの初期値を設定する。入力 SFTEN が論理値 0、R が論理値 1 のときリセットモードと呼び、クロック C に同期してカウンタを 0 に設定する。入力 SFTEN が論理値 0、R が論理値 0 のときインクリメントモードと呼び、クロック C に同期してカウンタのもつ値を 1 増加させる。

【 0 0 3 8 】

反転制御回路 6 0 0 の説明に戻り、その動作モードを説明する。入力 INTSEL が論理値 1、入力 RDSEL が論理値 0、入力 HCCKEN が論理値 1 のとき、クロック BRC に同期して 2 つのカウンタ 6 0 1、6 0 2 内の記憶素子の内容がシフトするので、入力 CTIN よりそれらの初期値を設定できる。入力 INTSEL が論理値 0、入力 RDSEL が論理値 1 のとき、クロック BRC に同期してパターン列内時刻用 カウンタ 6 0 1 を値 0 (記憶素子を全て 0) に設定する。入力 INTSEL が論理値 0、入力 RDSEL が論理値 0 のとき、入力 HCCKEN が論理値 0 であればクロック BRC でパターン列内時刻用 カウンタ 6 0 1 のみインクリメントし、入力 HCCKEN が論理値 1 であればクロック BRC で 2 つのカウンタ 6 0 1、6 0 2 をインクリメントする。さらに、パ

10

20

30

40

50

ターン列内で常に入力NBENが論理値1のとき、2つのカウンタの値が一致する場合のみ成分が全て論理値1であるパターンを出力し、それ以外のパターンでは全て論理値0であるパターンを出力する。パターン列内で常に入力NBENが論理値0のとき、全て論理値0の成分をもつパターン列が出力される。

【0039】

図11は、反転制御回路121の第二の回路構成例を示す。反転制御回路では、1パターンだけ半分のビットが論理値1であるパターン列を512通り出力する。反転制御回路620は、パターン列内時刻用カウンタ621と、比較器624を備え、マスク用のAND素子625、入力HCCKENとの論理積素子623は、反転制御回路600のAND素子605及び論理積素子603と同じである。パターン列番号用カウンタ622は反転制御回路600のパターン列番号用カウンタ602に9ビット目C9が追加された構成であり、ビットC9の論理値あるいはその反転値との論理積素子626～630を通してn個の出力RVS1、RVS2、...、RVS_nに接続する。図11の回路の場合、パターン列番号用カウンタ622の9ビット目C9が論理値0であれば、RVS1、RVS3、...のみ論理値1をもつパターンを出力し、9ビット目C9が論理値1であれば、RVS2、RVS4、...、RVS_n...のみ論理値1をもつパターンを出力する。

10

【0040】

図12は反転制御回路121の第三の回路構成例を示す。反転制御回路640では、1パターンだけ全ビットが論理値1であるパターン列を256通り、周期128の間隔をもつ2パターンの全ビットが論理値1であるパターン列を256通り、計512通りのパターン列を出力する。反転制御回路640は、パターン列内時刻用カウンタ641を備え、マスク用のAND素子645、入力HCCKENの論理積643は、それぞれ前記反転制御回路600のAND素子605及び論理積素子603と同じである。

20

【0041】

パターン列番号用カウンタ642はAND素子605及び論理積素子603に9ビット目C9が追加されており、回路644は前記比較器604の機能が一部変更されている。回路644は、パターン列番号用カウンタ642の9ビット目C9が論理値0のとき従来の8ビット比較器と同じ動作をするが、その9ビット目C9が論理値1のとき下位7ビットの比較器となる。この結果、パターン列番号用カウンタ642の9ビット目C9が論理値0のとき1パターンだけ全ビットが論理値1であるパターン列を出力し、パターン列番号用カウンタ642の9ビット目C9が論理値1のとき周期128ので出現する2パターンの全ビットが論理値1であるパターン列を出力する。

30

【0042】

図13は、反転制御回路121の第四の回路構成例を示す。反転制御回路660は、1パターンだけ全ビットが論理値1であるパターン列を256通り、時刻の連続した2パターンの全ビットが論理値1であるパターン列を256通り、1時刻分間隔のある2パターン全ビットが論理値1であるパターン列を256通り、2時刻分間隔のある2パターン全ビットが論理値1であるパターン列を256通り、計1024通りのパターン列を出力する。

【0043】

反転制御回路660は、パターン列内時刻用カウンタ661と、比較器664と、マスク用のAND素子665、入力HCCKENと信号BRDの論理積回路663とを備える。これらはそれぞれ前記反転制御回路600のパターン列内時刻用カウンタ601と、比較器604と、マスク用のAND素子605及び論理積回路603と同じである。パターン列番号用カウンタ662は反転制御回路600のカウンタ602に9、10ビット目C9、C10が追加された構成であり、AND素子666、667、668はカウンタ602に9、10ビット目C9、C10の出力を制御する。上記9、10ビット目C9、C10の論理値がどちらも0のときAND素子666、667、668の出力は全て論理値0、9ビット目C9が論理値1で10ビット目C10が論理値0のときAND素子666の出力のみが論理値1、9ビット目C9が論理値0で10ビット目C10が論理値1のときAND素子667の出力のみが論理値1、上記9、10ビッ

40

50

ト目C9,C10がどちらも論理値1のときAND素子668の出力のみが論理値1となる。

【0044】

記憶素子669～672は、順にその時点での時刻、1時刻前、2時刻前、3時刻前における比較器664の出力値を記憶する。OR素子673には、4つの入力、すなわち、比較器664の出力値、1時刻前の比較器664の出力値をもつ記憶素子670とAND素子666との出力値との論理積、2時刻前の比較器664の出力値をもつ記憶素子671とAND素子667との出力値との論理積、3時刻前の比較器664の出力値をもつ記憶素子672とAND素子668との出力値との論理積があり、そのいずれか1つが1であれば、出力RVS1、RVS2、...、RVSnへ論理値1が伝わる。この結果、パターン列番号用カウンタ662の9、10ビット目C9,C10の論理値がどちらも0のとき1パターンだけ全ビットが論理値1であるパターン列を出力し、9ビット目C9が論理値1で10ビット目C10が論理値0のとき時刻の連続した2パターンの全ビットが論理値1であるパターン列を出力し、9ビット目C9が論理値0で10ビット目C10が論理値1のとき1時刻分間隔のある2パターン全ビットが論理値1であるパターン列を出力し、9、10ビット目C9,C10がどちらも論理値1のとき2時刻分間隔のある2パターン全ビットが論理値1であるパターン列を出力する。

10

【0045】

ここで、反転制御回路121内のカウンタのビット数について、例えば、反転制御回路600ではパターン列内時刻用カウンタ601のビット数を8としたが、この数から求まるカウンタの周期256はパターン列内のパターン数上限を意味するもので、8に限定されない。もしパターン列内のパターン数がパターン列内時刻用カウンタ601の周期より大きい場合、反転制御回路600は、全部のビットが論理値1であるパターンがその周期で出現するパターン列を順次出力する。

20

【0046】

一方、反転制御回路600ではパターン列番号用カウンタ602のビット数を8としたが、この数から求まるカウンタの周期256はパターン列群内のパターン列の個数上限を意味するもので、この例のように8でなくても良い。パターン列群内のパターン列の個数がパターン列番号用カウンタ602の周期より大きいと同じパターン列が出現するだけであるので、この場合は想定しない。以上のカウンタのビット数に関する注意は、反転制御回路121の他の構成例でも同じである。

30

【0047】

好ましい実施形態では、上述の各パターン発生部、ビット反転制御部、反転制御回路部は、単独又は組み合わせた形態で単一の半導体集積回路で構成される。以下の本発明による半導体集積回路を説明する。

図15の(a)は、本発明によるテストパターン発生回路を持つ半導体集積回路の第一の実施例の構成を示す。半導体集積回路800は、同一パターン列発生部810と、ビット反転制御部820と、パターン発生制御部830が1つの半導体集積回路(テストパターン発生器)として構成される。同一パターン列発生部810は、図4、図5、図6に示した同一パターン列発生部400、420、440のいずれかとする。ビット反転制御部820内の反転制御回路821は、図10、図11、図12、図13に示した反転制御回路600、620、640、660のいずれかとする。パターン発生制御部830は、2つの入力BINIT、NHGENから出力INTSEL、RDSEL、NBEN、HCKENを4つの信号に設定するデコーダで、上記2つの入力に対する出力の組合せ及び動作モードの呼称を図15の(b)に示す。

40

【0048】

初期設定モード841では、同一パターン列発生部810及び反転制御回路821内の記憶素子に対し、入力SEEDINからシリアルに同一パターン列発生器810の初期値を設定する。パターン発生モード842では、同一パターン列発生部810はLFSRとして動作して疑似乱数パターンを順次発生し、反転制御回路821は全ての成分RVS1、RVS2...RVSnが論理値0であるパターンを出力し続けるため、パターン発生器800は、同一パターン列

50

発生部 810 の発生する疑似乱数パターンをそのまま出力する。seed 復元モード 843 では、同一パターン発生部内の seed が記憶されていたレジスタにパラレルあるいはシリアルに復元され、反転制御回路 821 内のパターン列内時刻用カウンタが 0 にリセットされる。

【0049】

近傍パターン発生モード 844 では、近傍パターン群、すなわち、基準となるパターン列に対し、反転ビットのないパターン列、1 パターンの一部又は全部のビットが反転したパターン列、連続あるいは所定のパターン数分の間隔があいた複数パターン的一部又は全部のビットが反転したパターン列の、全て、あるいは一部のパターン列を発生する。なお、以下の説明では、パターン発生器 800 が発生する近傍パターン群の中で、基準となるパターン列のことを親パターン、それ以外の基準となるパターン列と一部のビットが反転しているパターン列のことを子パターンと呼ぶ。

10

【0050】

図 16 は、本発明によるテストパターン発生回路を持つ半導体集積回路の第二の実施例の構成を示す。半導体集積回路 900 は、パターン発生器 901 とテスト対象回路 902 から構成される。パターン発生器 901 は、図 15 (a) のパターン発生器 800 と同じ構成で、同一パターン列発生部 810 とビット反転制御部 820 の組合せは前述の回路のいずれの組み合わせでも良い。テスト対象回路 902 は、図 2 の (a)、(c) に示したテスト対象回路 200、260 のいずれでもよい。本実施例では、故障検出に有効な近傍パターンを発生できるので、少ないテストデータ量で高い故障検出率を得ることができる。

20

【0051】

図 17 は、本発明によるパターン発生回路を持つ半導体集積回路の第三の実施例の構成を示す。半導体集積回路 920 は、複数のパターン発生器 921 ... 922 と一つのテスト対象回路 923 から構成される。パターン発生器 921 ... 922 のそれぞれは、前記パターン発生器 800 と同じで、同一パターン列発生部 810 とビット反転制御部 820 の組合せはいずれの組み合わせでも良い。テスト対象回路 923 は、図 2 (a)、(c) に示したテスト対象回路 200、260 のいずれでもよい。本実施例では、少ないテストデータ量で高い故障検出率を得ることができる効果に加えて、サイズの小さいパターン発生器を複数もち半導体集積回路内に分散して配置することにより、テスト対象回路内のスキャンチェーンを配線するオーバーヘッドを低減できる。

30

【0052】

図 18 は、本発明によるパターン発生回路を持つ半導体集積回路の第四の実施例の構成を示す。半導体集積回路 940 は、パターン発生器 941 とテスト対象回路 942 とパターン圧縮器 943 が一つ半導体集積回路を構成する。本実施例は本発明に基づいた BIST 方式の標準的な構成である。パターン発生器 941 は、前記パターン発生器 800 と同じ構成で、同一パターン列発生部 810 とビット反転制御部 820 の組合せはいずれの組み合わせでも良い。テスト対象回路 942 は、図 2 (a) に示したフルスキャン設計されたテスト対象回路 200 と同じ構成である。

【0053】

図 19 は、図 18 のパターン圧縮器の回路を示す。パターン圧縮器 943 の回路 960 は、BIST 方式で一般に使われる Multiple Input Signature Resister (MISR) と同じ構成である。MISR は、記憶素子 961 ~ 964 と排他的論理和 965、966 及びそのフィードバックからなる部分は LFSR で、各入力 SA1、SA2、...、SAN の論理値を各記憶素子 961 ~ 964 の直前で排他的論理和 967 ~ 980 をとる構成である。入力 SA1、SA2、...、SAN から入力されるパターン列を順次クロック BMC に同期して記憶素子 961 ~ 964 にコード化 (圧縮) していく。LFSR に対応する 2 の剰余系の多項式が原始多項式であれば、誤り見逃し (入力パターン列に誤りがある場合に最終的な記憶素子 961 ~ 964 のビット列に誤りがない) 確率は非常に小さい事が知られている。すなわち、MISR の最終状態の期待値のみを読み出した結果と比較することは、各パターンごとの期待値と比較する場合と同等の故障検出能力があることを意味する。従って、図 19 の回路構成では、さらに少ないテ

40

50

ストデータ量で高い故障検出率を得ることができる。

【0054】

図20は、本発明によるパターン発生回路を持つ半導体集積回路の第五の実施例の構成を示す。半導体集積回路980は、初期値保存レジスタ981とパターン発生器941とテスト対象回路942とパターン圧縮器943から構成される。前記第四の実施例である半導体集積回路940と比較して、入力TDIとパターン発生器941との間に初期値保存レジスタ981が追加されている点が異なり、第四の実施例である半導体集積回路940と同じ部分は同じ符号で示している。初期値保存レジスタ981は、入力BINITが論理値1、入力NHGENが論理値0の初期設定モードのときに、クロックBRCに同期してシフトレジスタとして動作する。記憶素子983～985はその数がパターン圧縮器943内の記憶素子数で、パターン圧縮器943の初期値を格納する。記憶素子986～988、989～991、992～994は、それぞれ、その数がパターン発生器941内の記憶素子数で、パターン発生器941の第一回目の初期値、第二回目の初期値、第s回目の初期値を格納する。なお、sはパターン列群の数である。前述の第四の実施例では、パターン列群毎に半導体集積回路テスト装置からパターン発生器941の初期値を与えるため、半導体集積回路テスト装置とのインターフェースにおけるオーバーヘッドがあった。しかし、第五の実施例では、複数のパターン列群に対するパターン発生器941の初期値を最初に格納できるため、半導体集積回路テスト装置とのインターフェースにおけるオーバーヘッドが小さくなる。

10

【0055】

以上の第一ないし第五の半導体集積回路の構成では、本発明による同一パターン列発生部110とビット反転制御部120が半導体集積回路内に内蔵されているが、これらがテスト対象回路と別の半導体集積回路テスト装置として構成してもよい。また、上記半導体集積回路テスト装置が同一パターン列発生部110を内蔵し、テストの対象となる半導体集積回路がビット反転制御部120を内蔵してもよい。さらに、同一パターン列発生部110やビット反転制御部120は、半導体集積回路外のウェハー部分に作成して、いわゆるウェハー上での半導体集積回路テスト装置に適用してもその効果は失われない。

20

【0056】

次に、本発明によるパターン発生回路を持つ半導体集積回路の動作をタイムチャートを使用して説明する。

30

図21は、図18に示した半導体集積回路940の基本的な動作のタイムチャートを示す。まず、BIST初期設定モード1001では、パターン発生器941(図15の800と同じ)内のパターン発生制御部830を初期設定モード841に設定し、パターン発生器941とパターン圧縮器943の記憶素子数分、クロックBMCとクロックBRCを交互にパルスを与え、入力TDIからseedを含むパターン発生器941とパターン圧縮器943の初期値を順に入力する。

【0057】

親パターン設定モード1002では、パターン発生器941内のパターン発生制御部830をパターン発生モード842に設定、スキャンイネーブルSENを論理値1に設定し、最大のスキャンチェーン長分、スキャンチェーンシフト用クロックCLKとクロックBRCに交互にパルスを与えることで、テスト対象回路942内の記憶素子全てに論理値が設定され、テスト対象回路942をテストする親パターンとなる。

40

【0058】

クロックアドバンス及びseed復元モード1003では、パターン発生器941内のパターン発生制御部830をseed復元モード843に設定、スキャンイネーブルSENを論理値0に設定し、データキャプチャー用のクロックCLKとseed復元用のクロックBRCに1回ずつパルスを与える。

【0059】

近傍パターン設定・圧縮モード1004では、パターン発生器941内のパターン発生制御部830を近傍パターン発生モード844に設定、スキャンイネーブルSENを論理値1

50

に設定し、最大のスキャンチェーン長分、クロックBMC、スキャンチェーンシフト用クロックCLK、クロックBRCに順にパルスを与え、クロックアドバンス結果の圧縮と次の子パターンの設定を同時に行う。そして、モード1003とモード1004を親パターン用の1回分+子パターン数分ほど繰り返す。ここまでのモード1001~1004までの手順で、パターン発生器941において1つのパターン列群を発生し、テスト対象回路942に対し親パターン1つと子パターン群によるテストを行い、その応答パターン結果をパターン圧縮器943の最終状態(全記憶素子の値)に圧縮している。

【0060】

このモード1001~1004までの手順をseed数、即ちパターン列群の数ほど、繰り返す。なお、2回目以降のBIST初期設定モード1001では、パターン発生器941とパターン圧縮器943の初期値設定と同時に、テストの応答パターンを圧縮した結果であるパターン圧縮器943の最終状態も読み出す。そして、上記手順の繰り返しの最終回については、圧縮結果抽出モード1005でパターン圧縮器943の最終状態を読み出す。

10

【0061】

ここで、図18に示した半導体集積回路940と図21に示したタイムチャートで、テスト対象回路942をどのようなパターンでテストできるかを説明する。説明のため、フルスキャン設計であるテスト対象回路942のスキャンチェーン長を259、スキャンチェーン本数をn個とする。従って、パターン発生器941の発生するパターン列内のパターン数は259となる。1つのパターン列群において、基準となるパターン列をスキャン展開したパターン、即ち、親パターンに対し、どのような子パターンが発生されるのかを、

20

【0062】

図24~図27の各図は図2(a)に示したテスト対象回路200を簡略化したもので、1つ1つの小さな正方形はスキャン機能付き記憶素子に対応し、ハッチングがあるときは親パターンと反転したビットであることを表す。また、子パターンの番号は、その子パターンが生成されたときのパターン列番号用カウンタの値を表す。

【0063】

図24は、反転制御回路600を用いた場合の子パターン256個を示す。子パターン0~2は周期256で2列が反転したパターンで、子パターン3~255は1列のみ反転したパターンとなる。

30

【0064】

図25は、反転制御回路620を用いた場合の子パターン512個を示す。反転ビットのある列は、子パターン0~255と子パターン256~511が図24に対応している。ただし、列内での反転しているビットは、子パターン0~255では奇数行目(IN1、IN3、...で接続されるスキャンチェーン上の記憶素子)で、子パターン256~511では偶数行目(IN2、IN4、...で接続されるスキャンチェーン上の記憶素子)である。

【0065】

図26は、反転制御回路640を用いた場合の子パターン256~384を示す。子パターン384個の中で、子パターン0~255は図24と同じであるため省いている。子パターン256~258は周期128で3列が反転したパターンで、子パターン259~383は周期128で2列反転したパターンとなる。

40

【0066】

図27は、反転制御回路660を用いた場合の子パターン1024個の中で、子パターン256~1023を示す。子パターン0~255は図24と同じであるため省かれている。子パターン256~511は周期256で連続した2列が反転したパターンで、子パターン256、257は計4列、子パターン258は計3列、子パターン259~511は2列が反転したパターンである。子パターン512~767は周期256で1列間隔のある2列が反転したパターンで、子パターン512は計4列、子パターン513、514は計3列、子パターン515~767は2列が反転したパターンである。子パターン768~1023は周期256で2列間隔のある2列が反転したパターンで、子パターン76

50

8 ~ 770 は計 3 列、子パターン 771 ~ 1023 は 2 列が反転したパターンである。

【0067】

図 28 は、本発明によるパターン発生回路を持つ半導体集積回路の第六の実施例の構成を示す。半導体集積回路 1200 は、パターン発生器 1210 とテスト対象回路 1220 から構成される。テスト対象回路 1220 は、8 個の記憶素子 1221 ~ 1228 と、それらの出力 x1 ~ x8 を入力とする AND 素子 1229 と、AND 素子 1229 の出力 x0 を入力する記憶素子 1230 をもち、記憶素子 1230 の出力 y0 は記憶素子 1221 ~ 1228 のそれぞれの入力 D に加えられる。

【0068】

テスト対象回路 1220 はフルスキャン設計されており、記憶素子 1230 はスキャンチェーン 1231 上にあり、記憶素子 1221 ~ 1228 はスキャンチェーン 1232 上にある。パターン発生回路 1210 は、図 15 に示したパターン発生器 800 と同じ構成である。ただし、その発生するパターンのビット数は出力 PG1、PG2、PG3、PG4 の 4 ビットである。同一パターン列発生部 810 は図 6 に示した同一パターン列発生部 440 と同じ構成である。反転制御回路 821 は図 10 に示した反転制御回路 600 と同様の構成で、2 つのカウンタ 601、602 のビット数はどちらも 3 ビットとする。

【0069】

テスト対象回路 1220 で、記憶素子の出力 x0、x1、...、x8、y0 に対し 0 縮退故障と 1 縮退故障を仮定したときのテストパターン集合を図 28 (b) に示す。列 1241 は各テストパターンの番号、列 1242 は各テストパターンを記憶素子出力名に対する論理値の組で表現したもの、列 1243 は各テストパターンで検出可能な縮退故障を「信号線名 / 縮退値」で表現し列挙したものである。なお、列 1242 で論理値が X となっているのは、不定値、即ち、論理値 0 でも論理値 1 でもよいことを表す。

【0070】

図 28 (b) で示したテストパターン集合の性質を述べる。パターン番号 2 ~ 9 のパターンは、記憶素子の出力 x1 ~ x8 の 8 ビット全てが論理値 1 であるパターン番号 1 のパターンに対し、1 ビットずつ反転したパターンである。即ち、パターン番号 1 のパターンとハミング距離が 1 である近傍パターンでほとんどの故障を検出できることがわかる。前述した K. H. Tsai 等の論文によると、多くの回路に対し、近傍パターンがテストパターンとして有効であることが指摘されている。

【0071】

図 29 は、上記半導体集積回路 1200 のテスト時における詳細な動作を示す。タイムチャートは図 21 で示したものに従うが、パターン圧縮器を持たないためクロック BMC はない。図中の「P」はクロックにパルスが与えられることを示す。また、パターン発生器の初期値に関しては、フィードバックイネーブル用の記憶素子の値を 0、LFSR 部分の seed を (1, 1, 1, 1)、パターン列番号用 3 ビットカウンタの値を 7 に設定する。なお、パターン列内時刻用 3 ビットカウンタの値は、子パターン出現ごとに 0 にリセットされるため、初期化する必要はない。

【0072】

時刻 1 ~ 11 はパターン発生器の初期値設定で、時刻 11 で上記の設定が完了する。時刻 12 ~ 19 はスキャンのシフトによる親パターン設定で、時刻 19 の記憶素子 y0、x1、... x8 の親パターンが設定される。時刻 20 ではクロックアドバンスにより、親パターンに対する応答結果を記憶素子 y0、x1、... x8 に格納する。時刻 21 ~ 28 では、親パターンに対する応答結果をスキャンのシフトにより外部出力端子 01、02 で読み出すと同時に、時刻 28 の記憶素子 x0、x1、... x8 に子パターン 0 (x8 のみ論理値 0 に反転) が設定される。このように、クロックアドバンスと子パターン設定・結果読み出しを繰り返す。時刻 91 で記憶素子 y0、x1、... x8 に子パターン 7 を設定し、時刻 92 でクロックアドバンスにより応答パターンを記憶素子 y0、x1、... x8 に格納する。そして、最後に、時刻 93 ~ 100 で、子パターン 7 の応答パターンを読み出す。

【0073】

10

20

30

40

50

以上の動作によって実現できるテストパターンは、図28(b)で示したテストパターン集合の全てをカバーする。即ち、親パターンはパターン番号1とパターン番号10をカバーし、子パターン0はパターン番号2とパターン番号10、子パターン1はパターン番号3とパターン番号10、子パターン2はパターン番号4とパターン番号10、子パターン3はパターン番号5とパターン番号10、子パターン4はパターン番号6とパターン番号10、子パターン5はパターン番号7とパターン番号10、子パターン6はパターン番号8とパターン番号10、子パターン7はパターン番号9とパターン番号11を全て含む。

【0074】

よって、半導体集積回路1200の全縮退故障をテストするためのテストデータ量は、パターン発生器1210の初期値の1seed、10ビット(カウンタの初期値を含む)である。一方、仮にテスト対象回路1220をストアドテスト方式で全縮退故障をテストする場合のテストデータ量は、11パターン、計74ビットである。このように、本発明を用いた半導体集積回路は、そのテストに必要なデータ量を著しく減らすことができる。

10

【0075】

次に、本発明による半導体集積回路について、疑似乱数パターンを発生させるタイムチャートと、複雑なタイムチャートとそのときのテストパターンを説明する。

図22は、図18に示した半導体集積回路940を例に、従来のLFSRと同様な疑似乱数パターンを発生するタイムチャートを示す。まず、BIST初期設定モード1011と初期パターン設定モード1012は、それぞれ図21のモード1001、1002と同じである。クロックアドバンスモード1013と乱数パターン設定・圧縮モード1014では、パターン発生モード842(図15(b))に設定する。このとき、図4~図7に示した同一パターン列発生部400、420、440、460はいずれもLFSRそのものと同じ動作をするため、その出力パターンは疑似乱数パターンとなる。最後の圧縮結果判定モード1015は、図21のモード1005と同じである。

20

【0076】

図23は、図18に示した半導体集積回路940に対する複雑なタイムチャートの例を示す。図18は図21で示したタイムチャートに比べて、親パターン設定モード1022~1025と、近傍パターン設定・圧縮モード1027~1030が複雑になっており、seed更新モード1031が新たに追加されている。以下変更があった部分についてのみ説明する。

30

【0077】

親パターン設定モードにおいて、空回し1022では、クロックBRCのみパルスを与えるため、LFSRのみ状態を遷移し、スキャンチェーンはシフトしない。これを指定された空回し数分繰り返す。空送り1023では、クロックCLKのみパルスを与えるため、LFSRの状態は変わらずにスキャンチェーンのみはシフトする。これを指定された空送り数分繰り返す。設定1024では、クロックCLKとクロックBRCに交互にパルスを与えることで、スキャンチェーンはシフトし、LFSRの状態も遷移する。以上1022~1024のモードを、テスト対象回路942内の記憶素子全てに論理値が設定されるまで繰り返す。その後、空送り追加1025で、クロックCLKのみパルスを与えるためスキャンチェーンのみシフトする。これを指定された空送り追加数分繰り返す。なお、空回し数はスキャンチェーン本数以下、空送り数は最大スキャンチェーン長以下、空送り追加数は空送り数以下でなければならない。

40

【0078】

また、近傍パターン設定・圧縮モード1027~1030も、パターン圧縮回路943を動作させるためのクロックBMCを除いては、親パターン設定モード1022~1025と同じである。seed更新モード1031については、パターン発生器941内のパターン発生制御部830を初期設定モード841に設定し、クロックBRCにパルスを与えないので、LFSRの状態をseed保存用記憶素子群にコピーする。

【0079】

上記の「空回し」、「空送り」、「seed更新」という動作について説明する。図18に

50

示した半導体集積回路 9 4 0 を例に、パターン発生器 9 4 1 は 4 ビットの LFSR をベースにした同一パターン列発生器 4 4 0 (図 6) と同様で、テスト対象回路 9 4 2 内のスキャンチェーン本数は 4 つ、スキャンチェーン長は全て 5 とする。このとき、テスト対象回路 9 4 2 に設定されたテストパターンと設定完了時点の LFSR の状態を、図 3 1 ~ 3 2 に示す。図では、パターン発生器 9 4 1 内の論理値を表現した部分 1 4 0 1 は LFSR の 4 ビットのみを記述し、フィードバック制御 1 4 0 2 と seed 入力 1 4 0 3 を模式化している。LFSR の初期値 (seed) は、s1、s2、s3、s4 が順に seed 入力 1 4 0 3 からシリアルに設定され、LFSR の初期値設定後に生成する乱数列を、r1、r2、r3、... と表現する。もし、フィードバック制御 1 4 0 2 の入力が 0 であれば、 $r1=s1$ 、 $r2=s2$ 、 $r3=s3$ 、... である。フィードバック制御 1 4 0 2 の入力が 1 であれば、 $r1=s1+s4$ 、 $r2=s2+r1$ 、 $r3=s3+r2$ 、... となる。

10

【 0 0 8 0 】

図 3 1 の (a) は、空回し数 = 0、空送り数 = 0 の場合で、テスト対象回路 1 4 1 1 で設定されたテストパターンは、右上から左下に向かって記憶素子間の相関があることがわかる。図 3 1 の (b) は、空回し数 = 1、空送り数 = 0 の場合で、テスト対象回路 1 4 1 2 で設定されたテストパターンは、右上から左下に向かって 1 列ずつあいた記憶素子間の相関があることがわかる。図 3 1 の (c) は、空回し数 = 0、空送り数 = 1、空送り追加数 = 0 の場合で、テスト対象回路 1 4 1 3 で設定されたテストパターンは、スキャンチェーンのシフト方向に 2 個ずつ同じ論理値が設定されており、右上から左下への相関もあることがわかる。図 3 1 の (d) は、空回し数 = 0、空送り数 = 1、空送り追加数 = 1 の場合で、テスト対象回路 1 4 1 4 で設定されたテストパターンは、テスト対象回路 1 4 1 3 で設定されたテストパターンを 1 つ右にシフトしたもので、記憶素子間の相関関係が変わっていることがわかる。また、図 3 1 の (e) は、空回し数 = 0、空送り数 = 0 で、seed 更新を行った場合を示しており、テスト対象回路 1 4 1 5 で設定されたテストパターンは、LFSR から発生する擬似乱数パターンの 2 つ目であるし、LFSR の初期値を r2、r3、r4、r5 と設定した場合のテストパターンとみることにもできる。

20

【 0 0 8 1 】

このように、「空回し」動作は、テストパターン中の記憶素子の相関を変えて、故障検出率を向上できるという効果がある。「空送り」動作は、スキャンチェーンシフト中やクロックアドバンス時の遷移信号発生割合を抑える効果があり、テスト時のノイズを低減できる。「空送り追加」動作は、「空送り」動作において生じる記憶素子の相関を変えて、

30

【 0 0 8 2 】

さらに、他の実施例として、図 1 7 で示した半導体集積回路 9 2 0 のように、複数のパターン発生器をもつときのテストパターンとその効果を述べる。各パターン発生器 9 2 1 ~ 9 2 2 には、パターン発生時に擬似乱数発生とシフト動作を切替えることができる同一パターン列発生部 4 6 0 (図 7) をもつ。テスト対象回路 9 2 3 のスキャンチェーンは 1 2 本で 4 本ずつパターン発生回路 9 2 1 ~ 9 2 2 に接続し、スキャンチェーン長は 5 とする。このとき、全ての同一パターン列発生部 4 6 0 の記憶素子 4 7 5 に論理値 0 を設定したときのテストパターンを図 3 2 の (a) に示し、初期設定用の入力 TDI に最も近いパターン発生器 1 4 2 3 の記憶素子 4 7 5 のみ論理値 0 で、それ以外のパターン発生器 1 4 2 1、1 4 2 2 の記憶素子 4 7 5 に論理値 1 を設定したときのテストパターンを図 3 2 の (b) に示す。

40

【 0 0 8 3 】

同一パターン列発生部 1 4 2 1、1 4 2 2、1 4 2 3 内の LFSR の初期値を、順に、(s11、s12、s13、s14)、(s21、s22、s23、s24)、(s31、s32、s33、s34) とする。また、LFSR の初期値設定後に生成する乱数列を、順に、(r11、r12、r13、...)、(r21、r22、r23、...)、(r31、r32、r33、...) とし、これらの値はそれぞれ (s11、s12、s13、s14)、(s21、s22、s23、s24)、(s31、s32、s33、s34) の一次式で表現できる。

【 0 0 8 4 】

ここで、図 3 2 の (a) の場合と図 3 2 の (b) の場合を比較して、seed 変換の容易さを説明す

50

る。テストパターン中の設定すべき記憶素子の集合は、1つのパターン発生器から接続されたスキャンチェーン上にあると仮定する。このとき、記憶素子群1431、1432、1433、1443における次元数、即ち、連立一次方程式を立てたときの変数の数は4であるが、記憶素子群1441、1442における次元数は8（記憶素子群1441における変数は、s11、s12、s13、s14、s21、s22、s23、s24。記憶素子群1442における変数は、s21、s22、s23、s24、s31、s32、s33、s34。）である。次元数が大きいほどseed変換成功の可能性が高いといえるため、図32(b)のシフト動作によるパターン発生はseed変換に有利だといえる。このように、パターン発生時に擬似乱数発生とシフト動作を切換えることができる同一パターン列発生部460は、複数のパターン生成期をもつ場合に、seed変換成功の可能性を高くするという効果がある。

10

【0085】

以下では、本発明による半導体集積回路のテスト方法に必要な情報の抽出方法、特に、同一パターン列発生部110における発生するパターン列の決定方法の一実施例を述べる。説明の前提となるハード構成として、同一パターン列発生部110は同一パターン発生部400、420、440、460のようなLFSRをベースにした構成、テスト対象回路はテスト対象回路200のようなフルスキャン設計された場合について述べる。

【0086】

テスト手順は、100%に近い故障検出率を効率的に得るために、まず、BIST方式で図22に示した乱数パターン発生モード1014で所定のパターン数分テストした後、図21に示した近傍パターン発生モード1004を実行し、最後にBIST方式による故障検出率を補うストアドテスト方式によるテストがあるという前提で述べる。従って、求めるべき情報は、同一パターン列発生部で用いるseed集合と、故障検出率を補う追加テストパターン集合である。なお、seed集合の各要素は、パターン発生器の初期値であり、その各初期値には、LFSRのseedの他に、空回し数、空送り数、空送り追加数、LFSR多項式切換え、パターン列番号用カウンタの初期値を含む。

20

【0087】

まず、seed集合及び追加テストパターン集合を求める手順の概略を図33に示す。ステップ1501では、BIST方式の乱数パターンモードにおける期待値計算と故障シミュレーションを実行し、故障検出情報1512を出力する。ステップ1502では、故障検出情報1512からわかる未検出故障に対し、テストパターンの生成あるいは冗長判定を行い、テストパターン集合1513を出力するとともに、故障検出情報1512に冗長判定結果を反映する。ステップ1503では、テストパターン集合1513を利用して近傍パターンの中心パターンである親パターンを生成し、親パターン集合1514を出力する。ステップ1504では、親パターン集合1514をBIST近傍パターン発生モードを実行できる情報であるseed集合1515に変換し、そのときの期待値計算と故障シミュレーションを実行して、故障検出情報1512を更新する。ステップ1505では、テストパターン集合1513と故障検出情報1512から故障検出率を補うのに必要なテストパターンを抽出して、追加テストパターン集合1516を出力し、故障検出情報1512を更新する。

30

【0088】

ステップ1502のテストパターン生成では、後で述べる親パターン生成1503でのパターンクラスタが効果的になるために、各パターンは設定不要な記憶素子はできるだけ不定値となるように生成する。例えば、1つの故障に対するテストパターンを既存のテストパターン生成アルゴリズムを利用して生成し、設定不要な記憶素子は不定値のままとして故障シミュレーションを行い、同時に検出される故障を見つける、といった処理を繰り返せばよい。ただし、生成したパターン同士の不定値部分を利用してマージするといった、いわゆるテストパターン圧縮は行わない方が望ましい。

40

【0089】

図34は前記親パターン生成処理1503の手順を示す。ステップ1701では、テストパターン集合を、各テストパターンの論理値0又は1を設定された記憶素子の集合が一致するグループに分割する。ステップ1702では、ステップ1701で求めたパターンの

50

グループをさらに分割し、分割された各グループ内のテストパターンが、使用する反転制御回路 121 の機能に応じて定まるハミング距離以下とする。例えば、反転制御回路 600、620 の場合はハミング距離が 1 以下で、反転制御回路 640、660 の場合はハミング距離が 2 以下とすればよい。このようにして生成されたテストパターンのグループの 1 つ 1 つをパターンクラスタと呼ぶ。ステップ 1703 では、各パターンクラスタに対し、各ビットごとに多数決をとることで求まるパターンを親パターンとする。

【0090】

図 35 は、前記 seed 集合生成処理 1504 の手順を示す。ステップ 1811 で、親パターン集合 1514 から未試行の親パターンを選択し、ステップ 1812 で、空回し数、空送り数、空送り追加数、LFSR 多項式の与えられたパターン生成条件の中から、未試行の条件を選択する。選択した親パターン、パターン生成条件に対し、以下の処理を行う。ステップ 1813 では、スキャンチェーンの接続と空回し数、空送り数、空送り追加数に依存する記憶素子間の相関についてチェックする。もし、実現できないパターンと判明すれば、ステップ 1814 で未試行のパターン生成条件があればステップ 1812 に戻り、未試行のパターン生成条件がなければステップ 1820 に進む。もし、ステップ 1813 での相関チェックが問題無ければ、ステップ 1815 へ進み、選択した LFSR 多項式から発生する乱数列より親パターンを実現できるように、LFSR の seed を連立一次方程式の解法を利用して求める。もし条件を満たす seed が存在しない場合は、ステップ 1816 からステップ 1614 へ進む。条件を満たす seed が存在する場合はステップ 1817 へ進み、親パターン及び使用する反転制御回路 121 の機能に応じて発生される子パターン群に対し、故障シミュレーションを行う。このとき、ステップ 1818 で新規に検出される故障がなければステップ 1820 へ進む。新規に検出される故障があれば、有効な子パターンの番号の集合を記憶しておく。そして、ステップ 1819 では、LFSR の seed と、選択した LFSR 多項式から求まる LFSR フィードバック位置を指定する記憶素子の値と、有効な子パターンの最初の番号から求まるパターン列番号用カウンタの値から決まるパターン発生器の初期値を採用するとともに、空回し数、空送り数、空送り追加数と、有効な子パターンの最初の番号と最後の番号からそれらの差に 1 加えた値として求まる子パターン数を記憶する。ステップ 1820 では、未試行の親パターンがあればステップ 1811 に戻り、全ての親パターンに対し試行したならば、処理を終了する。

【0091】

図 36 は、前記追加テストパターン生成処理 1505 の手順を示す。ステップ 1931 では、テストパターン集合 1513 から未試行のテストパターンを選択し、そのパターンに対しステップ 1932 で故障シミュレーションを行う。ステップ 1933 で、新規に故障を検出した場合、ステップ 1934 へ進み、そうでなければステップ 1937 へ進む。ステップ 1934 では、試行中のテストパターンが既に保存してあるパターンにマージ可能な場合は、ステップ 1936 でマージし、そうでなければ、ステップ 1935 で新規にそのパターンを保存する。ステップ 1937 では未試行のテストパターンがあれば、ステップ 1931 へ進み、テストパターン集合 1513 のテストパターン全てについて試行したならば、処理を終了する。

【0092】

ここで、図 28 で示した半導体集積回路 1200 に対し、ステップ 1502、1503、1504 を実行した例を説明する。まず、テストパターン生成処理 1502 の結果であるテストパターン集合 1513 は、図 28 (b) のテストパターン 1242 である。次に、親パターン生成 1503 における処理を説明する。ステップ 1701 により、テストパターンの番号 1 ~ 11 を、信号線 x1、x2、...、x8 を設定する必要があるパターン番号 1 ~ 9 のグループと、信号線 y0 を設定する必要があるパターン番号 10 ~ 11 のグループに分類する。ステップ 1702 で、所定のハミング距離を 1 とするとき、パターン番号 1 ~ 9 のグループはパターン番号 1 の近傍であり、パターン番号 10 ~ 11 のグループはパターン番号 10 の近傍であることから、それぞれのグループがパターンクラスタとなりうる。ステップ 1703 により、パターン番号 1 ~ 9 のグループでビット毎多数決をとると親パター

10

20

30

40

50

ンはパターン番号 1 に一致し、同様に、パターン番号 1 0 ~ 1 1 のグループでは親パターンはパターン番号 1 0 に一致する。

【 0 0 9 3 】

次に、seed生成処理 1 5 0 4 に進む。ステップ 1 7 1 1 で親パターンであるパターン番号 1 を選択し、ステップ 1 7 1 2 で空回し数 = 0、空送り数 = 0、LFSR多項式を $x^4+1=0$ (シフトレジスタとして動作) を選択する。ステップ 1 6 1 3 で記憶素子間の関連チェックをするが、テスト対象回路 1 2 2 0 では関連がなくOKである。ステップ 1 6 1 5 では、seedを (s1、s2、s3、s4) とするとき、記憶素子 1 2 2 8 ~ 1 2 2 1 に対する条件として、順に、s1=1、s2=1、s3=1、s4=1、r1=1、r2=1、r3=1、r4=1、r5=1となり、この連立一次方程式を解く。ただし、r1、r2、...、r5は、LFSRの初期値設定後に生成する乱数列で、r1=s1、r2=s2、r3=s3、r4=s4、r5=s1である。この連立一次方程式の解は存在し、s1=1、s2=1、s3=1、s4=1である。即ち、seed変換が成功したので、ステップ 1 6 1 7 でこのseedから生成される親パターンと子パターン 8 個に対する故障シミュレーションを実行すると、図 2 9 で示したテストパターンが出現するので、全縮退故障を検出する。従って、ステップ 1 6 1 9 でこのseedを採用し、seedである (1、1、1、1) と、パターン生成条件である、空回し数 = 0、空送り数 = 0、LFSR多項式を $x^4+1=0$ という情報を記憶する。そして、ステップ 1 6 1 1 に戻り、親パターンであるパターン番号 1 0 に対する処理がステップ 1 6 1 2、1 6 1 3、1 6 1 5、1 6 1 6、1 6 1 7 と進み、(X、1、X、X) というseedを求められるが、ステップ 1 6 1 8 にて新規故障検出がないため、このseedは採用しない。

【 0 0 9 4 】

図 3 7 は、本発明のテストパターン発生回路における反転制御回路の別の回路を示す。図 1 1 の回路では、図 2 5 に示すように偶数列又は奇数列での反転制御を可能にする。これに対して、図 3 7 の回路では、任意の列に対して、反転制御を可能にする。パターン列内時刻用カウンタ 6 8 1、パターン列番号用カウンタ 6 8 2、比較器 6 8 4、AND素子 6 8 5 は、図 1 0 に示すパターン列内時刻用カウンタ 6 0 1、パターン列番号用カウンタ 6 0 2、比較器 6 0 4、AND素子 6 0 5 とそれぞれ同じ機能を持つ構成要素であり、ここでは説明を省略する。本構成回路の特徴は、列ごとに反転を制御するために反転情報レジスタ 6 9 4 を設けた点にある。反転情報レジスタ 6 9 4 は n 個のレジスタを含み、各レジスタの出力と AND素子 6 8 5 の出力との論理積が各出力 R V S 1、R V S 2、... R V S n に出力される。

【 0 0 9 5 】

反転情報レジスタ 6 9 4 には、入力 C T I N から列反転パターンが入力される。図 3 8 は、(レジスタ 1、レジスタ 2、レジスタ 3 ... レジスタ n) (尚、レジスタ i は出力 R V S i に対応するレジスタである。) に列反転パターンとして (1、1、0、... 1) が入力された場合の子パターンの出力を示したものである。この場合、反転行 (例えば子パターン 0 における第 3 行) において、第 1、2、n 列は親パターンの反転パターンであり、第 3 列は非反転パターンとなる。もちろん、本回路構成に対して、図 1 2 の回路構成を適用することにより反転する行の周期を可変としてもよく、図 1 3 の回路構成を適用することにより反転する行を複数行にしてもよい。これらの場合においても反転行において、所定の列が非反転パターンとなる。

【 0 0 9 6 】

また、反転情報レジスタに格納する値は、レジスタに格納された値としたが、子パターンごとに変化させることもできる。この場合、所望の列反転パターンを形成するための回路を別途反転情報レジスタ 6 9 4 に付加する必要がある。また、図 3 7 の回路構成では別途反転情報レジスタ 6 9 4、カウンタ 6 8 1、6 8 2 が一本のスキランチェーンにつながれているが、反転情報レジスタ 6 9 4 についてはカウンタとは別に初期値を入力するように構成してもよい。この場合、反転情報レジスタ 6 9 4 がカウンタとは独立にその値を制御することができる。

以上のように、上記の処理に従えば、本発明による半導体集積回路のテスト方法に必要な情報、即ち、パターン発生器の初期値を計算する事ができる。

10

20

30

40

50

【 0 0 9 7 】

【 発 明 の 効 果 】

本発明による半導体集積回路は、近傍パターンを発生する回路をテスト対象回路と独立に付加することにより、設計期間の増大の問題がなく、テスト対象回路の動作速度オーバーヘッドがなく、ハード的なオーバーヘッドが小さく、そして、少ないテストデータ量で、高い故障検出率を得ることができる。

【 図 面 の 簡 単 な 説 明 】

【 図 1 】 本発明によるテスト方法を実施する回路の一実施例の構成を示す図。

【 図 2 】 上記テスト対象回路の回路を示す図。

【 図 3 】 LFSR (LFSR) の回路図。

【 図 4 】 本発明によるパターン発生回路の実施例における同一パターン発生部の第一の回路構成例の回路図。

【 図 5 】 本発明によるパターン発生回路の実施例における同一パターン列発生部の第二の回路構成例の回路図。

【 図 6 】 本発明によるパターン発生回路の実施例における同一パターン列発生部の第三の回路構成例の回路図。

【 図 7 】 本発明によるパターン発生回路の実施例における同一パターン列発生部の第四の回路構成例の回路図。

【 図 8 】 本発明によるパターン発生回路の実施例における同一パターン列発生部の第五の回路構成例の回路図。

【 図 9 】 同一パターン列発生部の動作を表す図。

【 図 1 0 】 本発明によるパターン発生回路の実施例における反転制御回路の第一の回路構成例の回路図。

【 図 1 1 】 本発明によるパターン発生回路の実施例における反転制御回路の第二の回路構成例の回路図。

【 図 1 2 】 本発明によるパターン発生回路の実施例における反転制御回路の第三の回路構成例の回路図。

【 図 1 3 】 本発明によるパターン発生回路の実施例における反転制御回路の第四の回路構成例の回路図。

【 図 1 4 】 上記反転制御回路に使用される n ビットカウンタの回路図と、その動作を表す図。

【 図 1 5 】 本発明によるパターン発生回路を持つ半導体集積回路 (パターン発生器) の第一の実施例の構成及びその動作を表す図。

【 図 1 6 】 本発明によるパターン発生回路を持つ半導体集積回路の第二の実施例の構成を図。

【 図 1 7 】 本発明によるパターン発生回路を持つ半導体集積回路の第三の実施例の構成を示す図。

【 図 1 8 】 本発明によるパターン発生回路を持つ半導体集積回路の第四の実施例の構成を示す図。

【 図 1 9 】 図 1 8 のパターン圧縮器の回路図。

【 図 2 0 】 本発明によるパターン発生回路を持つ半導体集積回路の第五の実施例の構成を示す図。

【 図 2 1 】 図 1 8 に示した半導体集積回路の基本的な動作のタイムチャート図。

【 図 2 2 】 図 1 8 に示した半導体集積回路の疑似乱数パターンを発生するタイムチャート図。

【 図 2 3 】 図 1 8 に示した半導体集積回路に疑似乱数パターンを発生するタイムチャート図。

【 図 2 4 】 図 1 0 の反転制御回路を用いた場合の子パターンの様子を示す図。

【 図 2 5 】 図 1 1 の反転制御回路を用いた場合の子パターンの様子を示す図。

【 図 2 6 】 図 1 2 の反転制御回路を用いた場合の子パターンの様子を示す図。

10

20

30

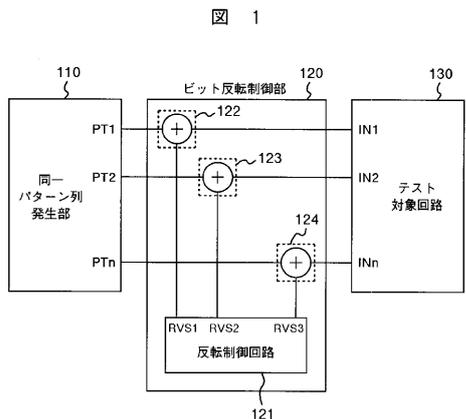
40

50

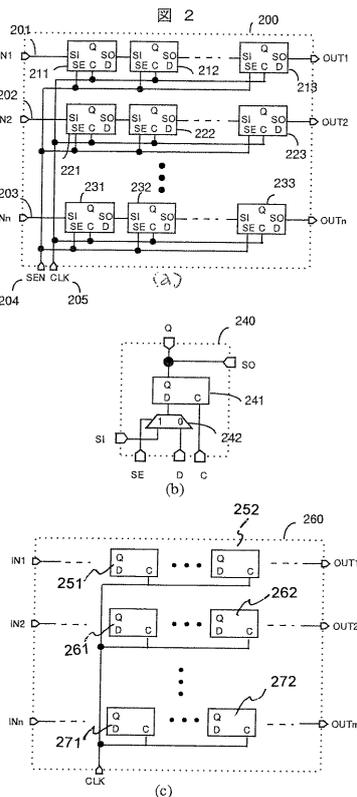
- 【図 27】 図 13 の反転制御回路を用いた場合の子パターンの様子を示す図。
- 【図 28】 本発明によるパターン発生回路を持つ半導体集積回路の第六の実施例の構成及び動作を示す図。
- 【図 29】 図 28 の半導体集積回路のテスト時における詳細な動作を示す図。
- 【図 30】 図 28 の半導体集積回路のテスト時における詳細な動作を示す図。
- 【図 31】 本発明による半導体集積回路において、パターン発生条件による動作の違いを説明する模式図。
- 【図 32】 本発明による半導体集積回路において、パターン発生条件による動作の違いを説明する模式図。
- 【図 33】 本発明による半導体集積回路のテストに必要な情報の抽出方法の一実施例の処理フロー図。 10
- 【図 34】 図 33 の親パターン生成処理の詳細な処理フロー図。
- 【図 35】 図 33 のseed集合生成処理の詳細な処理フロー図。
- 【図 36】 図 33 のテストパターン生成処理の詳細な処理フロー図。
- 【図 37】 本発明によるパターン発生回路の実施例における反転制御回路の別の回路図。
- 【図 38】 図 37 の反転制御回路を用いた場合の子パターンの様子を示す図。
- 【符号の説明】
- 1 1 0 ... 同一パターン列発生部、
- 1 2 0 ... ビット反転制御部、 1 2 1 ... 反転制御回路
- 1 2 2 , 1 2 3 , 1 2 4 ... 排他論理和回路、
- 1 3 0 ... テスト対象回路、
- 8 0 0 ... パターン発生器。

20

【図 1】

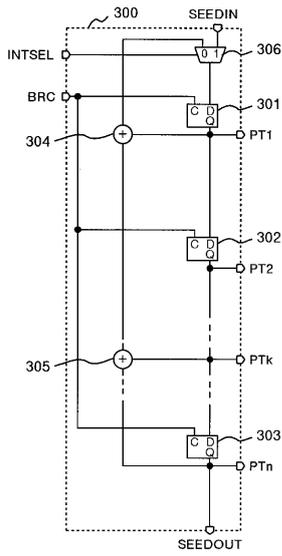


【図 2】



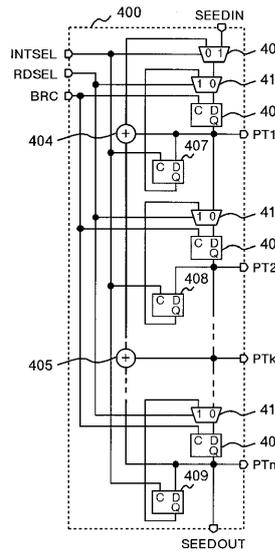
【 図 3 】

図 3



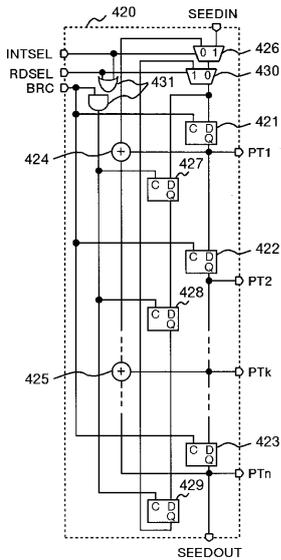
【 図 4 】

図 4



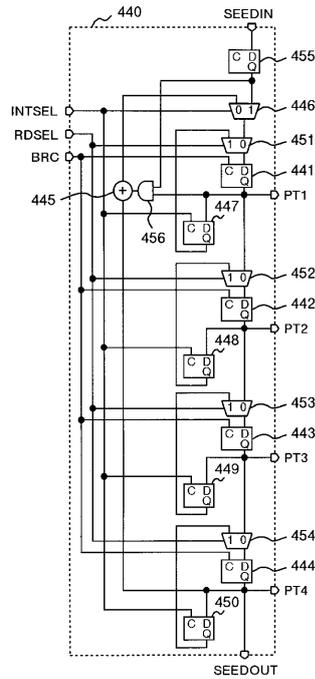
【 図 5 】

図 5



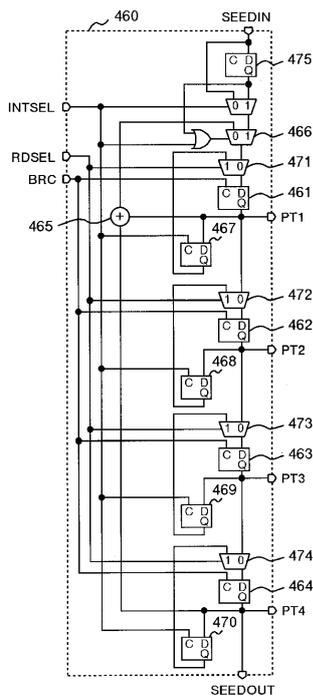
【 図 6 】

図 6



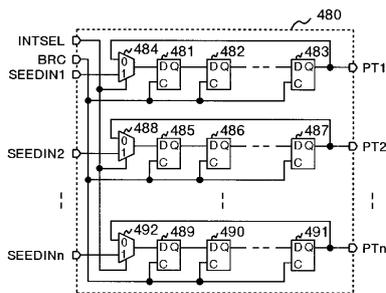
【 図 7 】

図 7



【 図 8 】

図 8



【 図 9 】

図 9

(a)

動作モード	INTSEL	BRC	リツタ
初期設定	1	↑	シフト
パターン発生	0	↑	乱数

(b)

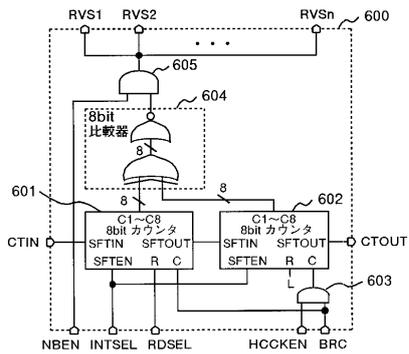
動作モード	INTSEL	RDSEL	BRC	リツタ	seed 記憶素子
初期設定	1	0	↑	シフト	seed コピー
パターン発生	0	0	↑	乱数	保持
Seed 復元	0	1	↑	復元	保持
Seed 更新	1	0	—	—	seed コピー

(c)

動作モード	INTSEL	RDSEL	BRC	リツタ	seed 記憶素子
初期設定	1	0	↑	シフト	seed コピー
パターン発生	0	0	↑	乱数	保持
Seed 復元	—	1	↑	復元	保持

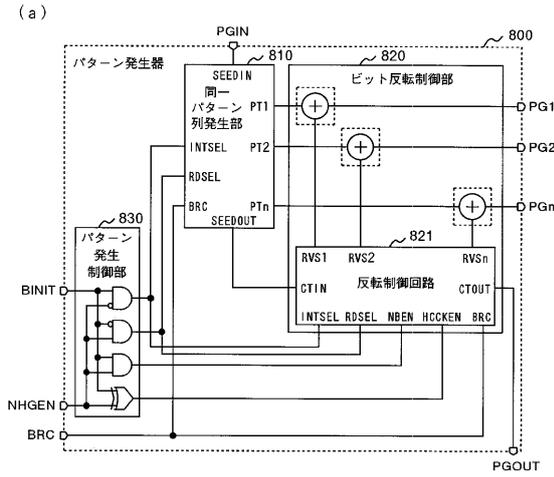
【 図 10 】

図 10



【 図 1 5 】

図 1 5

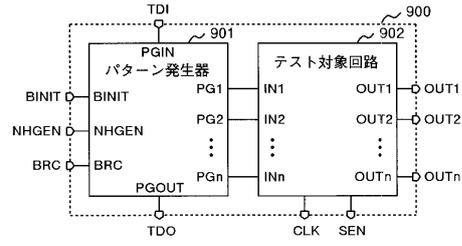


(b)

動作モード	BINIT	NHGEN	INTSEL	RDSEL	NBEN	HCKKEN
841 初期設定	1	0	1	0	0	1
842 パターン発生	0	0	0	0	0	0
843 seed 復元	0	1	0	1	0	1
844 近傍パターン発生	1	1	0	0	1	0

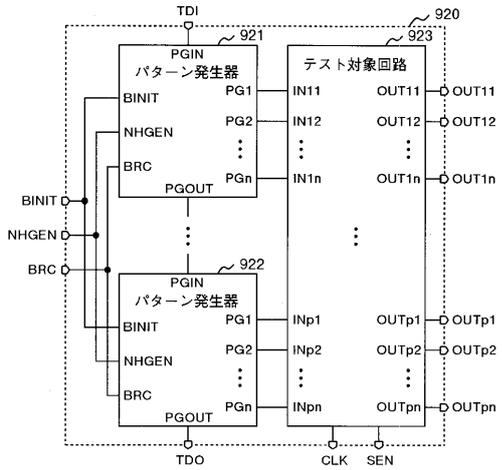
【 図 1 6 】

図 1 6



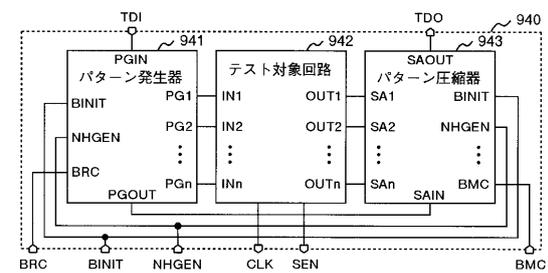
【 図 1 7 】

図 1 7

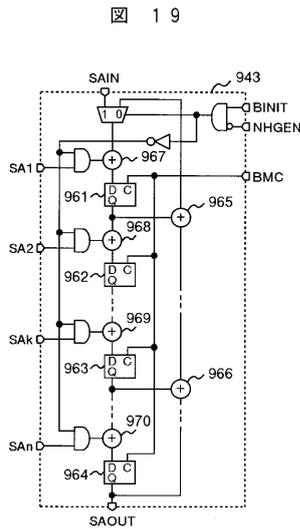


【 図 1 8 】

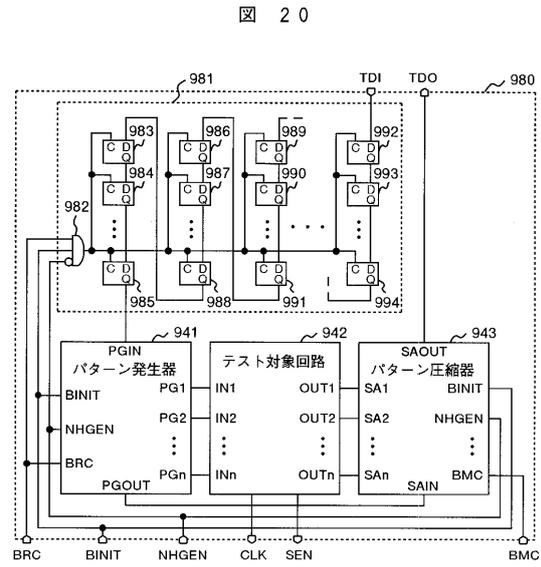
図 1 8



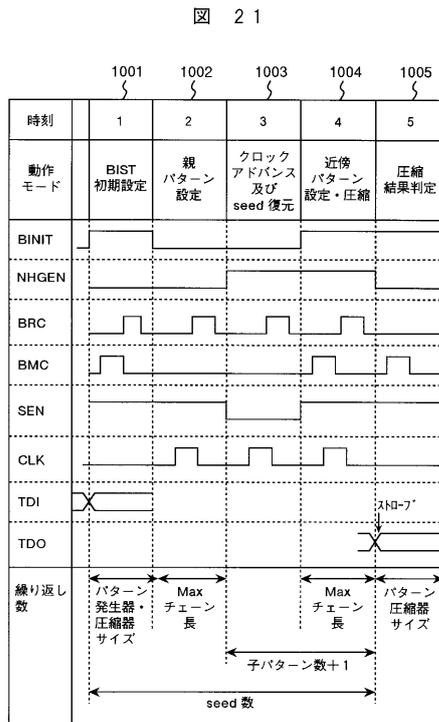
【図 19】



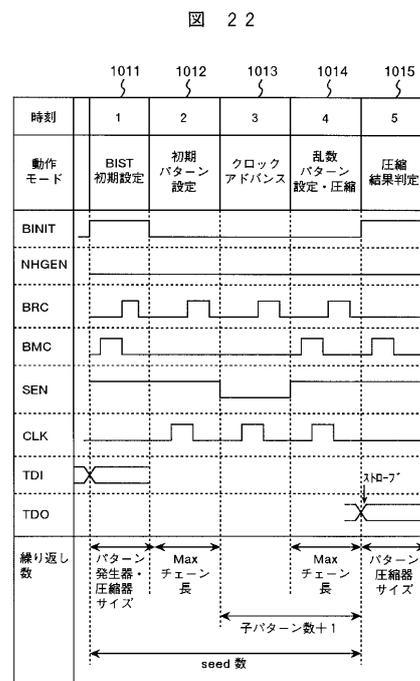
【図 20】



【図 21】



【図 22】



【 図 27 】

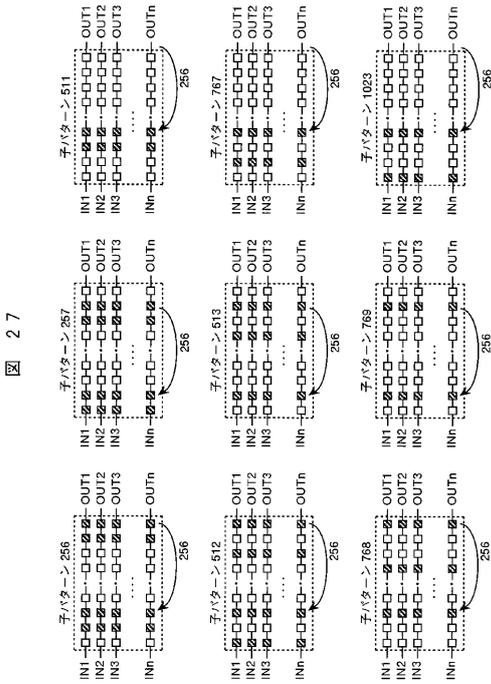
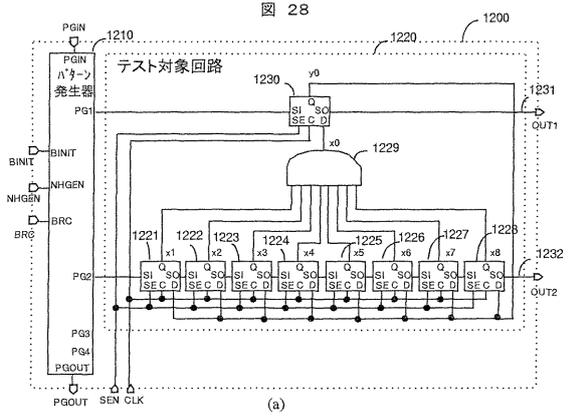


図 27

【 図 28 】



パターン 番号	テストパターン x1: x2: x3: x4: x5: x6: x7: x8: y0								検出可能な縮退故障 (信号線名/縮退値)					
	x1	x2	x3	x4	x5	x6	x7	x8	y0	x1/0	x2/0	x3/0	x4/0	
1	1	1	1	1	1	1	1	1	X	x0/0	x1/0	x2/0	x3/0	x4/0
2	0	1	1	1	1	1	1	1	X	x0/1	x1/1			
3	1	0	1	1	1	1	1	1	X	x0/1	x2/1			
4	1	1	0	1	1	1	1	1	X	x0/1	x3/1			
5	1	1	1	0	1	1	1	1	X	x0/1	x4/1			
6	1	1	1	1	0	1	1	1	X	x0/1	x5/1			
7	1	1	1	1	1	0	1	1	X	x0/1	x6/1			
8	1	1	1	1	1	1	0	1	X	x0/1	x7/1			
9	1	1	1	1	1	1	1	0	X	x0/1	x8/1			
10	X	X	X	X	X	X	X	X	1	y0/0				
11	X	X	X	X	X	X	X	0	y0/1					

(b)

【 図 29 】

図 29

時刻	動作	BI NIT	NH GEN	BRC REN	SEN CLK	TDI	FB1 EN	PG1	PG2	PG3	PG4	LC1	LC2	LC3	HC1	HC2	HC3
0	初期設定	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
12	親パターン設定	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
13		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
14		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
15		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
16		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
17		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
18		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
19	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
20	知リフトパルス	0	1	P	0	P	X	0	1	1	1	1	0	0	0	0	0
21	子パターン0設定 (親パターン結果)	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
22		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
23		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
24		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
25		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
26		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
27		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
28	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0	
29	知リフトパルス	0	1	P	1	P	X	0	1	1	1	1	0	0	0	0	0
30	子パターン1設定 (子パターン0結果)	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
31		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
32		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
33		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
34		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
35		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
36		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
37	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0	
38	知リフトパルス	0	1	P	1	P	X	0	1	1	1	1	0	0	0	0	0
83	知リフトパルス	0	1	P	1	P	X	0	1	1	1	1	0	0	0	0	0
84	子パターン7設定 (子パターン6結果)	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
85		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
86		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
87		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
88		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
89		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
90		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
91	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0	
92	知リフトパルス	0	1	P	1	P	X	0	1	1	1	1	0	0	0	0	0
93	子パターン7結果	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
94		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
95		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
96		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
97		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
98		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
99		1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0
100	1	1	P	1	P	X	0	1	1	1	1	1	0	0	0	0	

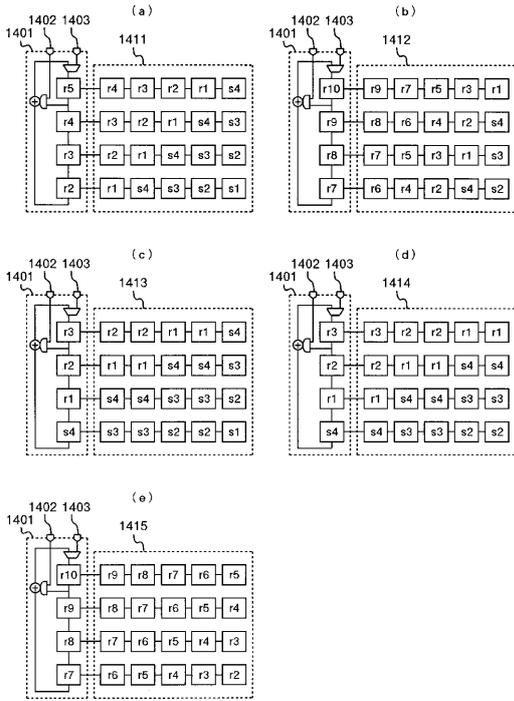
【 図 30 】

図 30

時刻	動作	y0	x1	x2	x3	x4	x5	x6	x7	x8	o1	o2
0	初期設定	X	X	X	X	X	X	X	X	X	X	X
1		X	X	X	X	X	X	X	X	X	X	X
2		X	X	X	X	X	X	X	X	X	X	X
3		X	X	X	X	X	X	X	X	X	X	X
4		X	X	X	X	X	X	X	X	X	X	X
5		X	X	X	X	X	X	X	X	X	X	X
6		X	X	X	X	X	X	X	X	X	X	X
7		X	X	X	X	X	X	X	X	X	X	X
8		X	X	X	X	X	X	X	X	X	X	X
9		X	X	X	X	X	X	X	X	X	X	X
10		X	X	X	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X	X	
12	親パターン設定	1	1	X	X	X	X	X	X	X	X	X
13		1	1	1	X	X	X	X	X	X	X	X
14		1	1	1	1	X	X	X	X	X	X	X
15		1	1	1	1	1	X	X	X	X	X	X
16		1	1	1	1	1	1	X	X	X	X	X
17		1	1	1	1	1	1	1	X	X	X	X
18		1	1	1	1	1	1	1	1	X	X	X
19	1	1	1	1	1	1	1	1	1	X	X	
20	知リフトパルス	1	1	1	1	1	1	1	1	1	1	X
21	子パターン0設定 (親パターン結果)	0	0	1	1	1	1	1	1	1	1	1
22		1	1	0	1	1	1	1	1	1	1	0
23		1	1	1	0	1	1	1	1	1	1	1
24		1	1	1	1	0	1	1	1	1	1	1
25		1	1	1	1	1	0	1	1	1	1	1
26		1	1	1	1	1	1	0	1	1	1	1
27		1	1	1	1	1	1	1	0	1	1	1
28	1	1	1	1	1	1	1	1	0	1	1	
29	知リフトパルス	0	1	1	1	1	1	1	1	1	1	1
30	子パターン1設定 (子パターン0結果)	1	1	1	1	1	1	1	1	1	0	1
31		0	0	1	1	1	1	1	1	1	1	1
32		1	1	0	1	1	1	1	1	1	1	1
33		1	1	1	0	1	1	1	1	1	1	1
34		1	1	1	1	0	1	1	1	1	1	1
35		1	1	1	1	1	0	1	1	1	1	1
36		1	1	1	1	1	1	0	1	1	1	1
37	1	1	1	1	1	1	1	0	1	1	1	
38	知リフトパルス	0	1	1	1	1	1	1	1	1	1	1
83	知リフトパルス	0	1	1	1	1	1	1	1	1	1	1
84	子パターン7設定 (子パターン6結果)	1	1	1	1	1	1	1	1	1	0	1
85		1	1	1	1	1	1	1	1	1	1	1
86		1	1	1	1	1	1	1	1	1	1	1
87		1	1	1	1	1	1	1	1	1	1	1
88		1	1	1	1	1	1	1	1	1	1	1
89		1	1	1	1	1	1	1	1	1	1	1
90		1	1	1	1	1	1	1	1	1	1	1
91	1	1	1	1	1	1	1	1	1	1	1	
92	知リフトパルス	0	0	0								

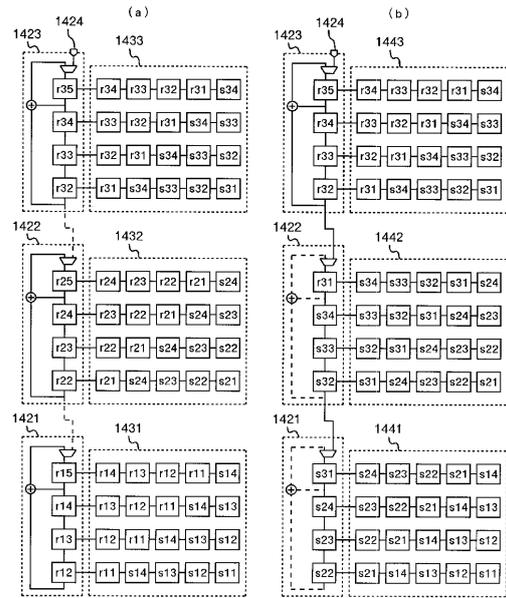
【 図 3 1 】

図 3 1



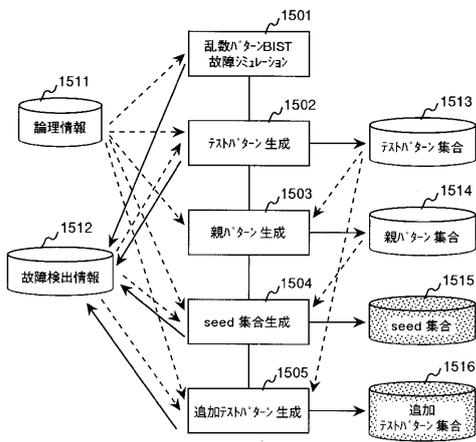
【 図 3 2 】

図 3 2



【 図 3 3 】

図 3 3



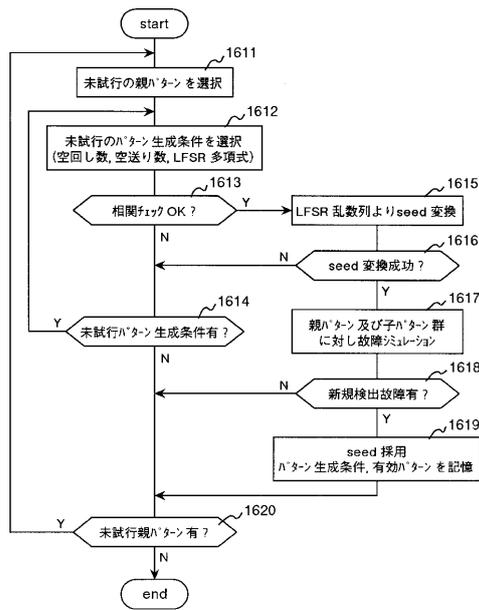
【 図 3 4 】

図 3 4



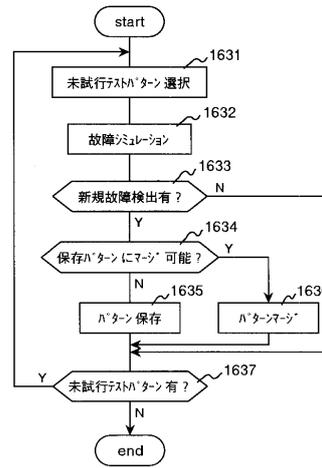
【 図 3 5 】

図 3 5



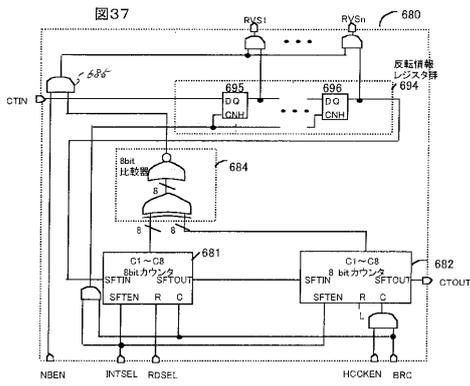
【 図 3 6 】

図 3 6



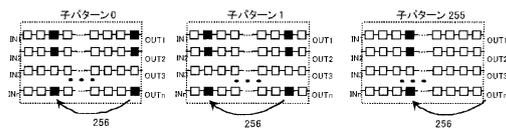
【 図 3 7 】

図37



【 図 3 8 】

図38



フロントページの続き

- (72)発明者 中尾 教伸
東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内
- (72)発明者 畠山 一実
東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内
- (72)発明者 夏目 幸一郎
東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内
- (72)発明者 清重 賢一
東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内
- (72)発明者 河野 正樹
東京都青梅市新町六丁目16番地の3 株式会社日立製作所 デバイス開発センタ内
- (72)発明者 浜本 正人
東京都青梅市新町六丁目16番地の3 株式会社日立製作所 デバイス開発センタ内
- (72)発明者 吉田 英文
神奈川県足柄上郡中井町境456番地 株式会社日立インフォメーションテクノロジー内
- (72)発明者 中村 知司
神奈川県足柄上郡中井町境456番地 株式会社日立インフォメーションテクノロジー内

審査官 神谷 健一

- (56)参考文献 特開昭55-060872(JP,A)
特開昭56-127253(JP,A)
特開平02-103481(JP,A)
特開平04-232480(JP,A)
特開平05-040151(JP,A)
特開平06-003424(JP,A)
特開平06-148282(JP,A)
特開平09-106418(JP,A)
特開平09-257885(JP,A)
特開2000-266815(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G01R 31/28-31/3193
G06F 11/22-11/26