

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4147005号
(P4147005)

(45) 発行日 平成20年9月10日(2008.9.10)

(24) 登録日 平成20年6月27日(2008.6.27)

(51) Int.Cl.		F I	
HO 1 L 21/82	(2006.01)	HO 1 L 21/82	A
G 1 1 C 29/12	(2006.01)	G 1 1 C 29/00	6 7 1
HO 1 L 21/822	(2006.01)	HO 1 L 27/04	T
HO 1 L 27/04	(2006.01)	HO 1 L 27/04	U

請求項の数 11 (全 34 頁)

(21) 出願番号	特願2000-611314 (P2000-611314)	(73) 特許権者	503121103 株式会社ルネサステクノロジ 東京都千代田区大手町二丁目6番2号
(86) (22) 出願日	平成11年4月14日 (1999.4.14)	(74) 代理人	100089071 弁理士 玉村 静世
(86) 国際出願番号	PCT/JP1999/001974	(72) 発明者	佐藤 正幸 日本国東京都小平市上水本町五丁目20番 1号 株式会社日立製作所 半導体グルー プ内
(87) 国際公開番号	W02000/062339	(72) 発明者	志水 勲 日本国東京都小平市上水本町五丁目20番 1号 株式会社日立製作所 半導体グルー プ内
(87) 国際公開日	平成12年10月19日 (2000.10.19)		
審査請求日	平成16年2月27日 (2004.2.27)		

最終頁に続く

(54) 【発明の名称】 半導体集積回路およびそのテスト方法並びに製造方法

(57) 【特許請求の範囲】

【請求項1】

半導体チップ上に、複数の基本論理セルからなり各基本論理セルごとに、その基本論理セルが正常か異常かを示す信号を出力可能で、かつそれぞれが正常を示す信号を出力する複数の基本論理セルで、任意の論理を構成可能な可変論理回路が搭載されてなる半導体集積回路であって、

上記可変論理回路は、相補的な信号を出力可能な2線ラインロジック回路と該2線ラインロジック回路の出力信号を比較して異常の有無を判定する判定手段とからなる基本論理セルと、これら基本論理セル間の信号線の接続を切り替え可能な複数の可変スイッチ回路とを備え、上記可変スイッチ回路による接続を切り替えることにより任意の論理を構成可能であって、上記2線ラインロジック回路から出力される信号が他の基本論理セル内の2線ラインロジック回路に供給され、上記判定手段の出力信号が、回路が正常か異常かを示す信号として各基本論理セルから出力されるように構成され、

上記可変スイッチ回路は、互いに交差する信号線間を接続遮断可能なスイッチ素子と、該スイッチ素子の状態を制御する情報を記憶する書込み読出し可能な記憶素子とからなり、上記判定手段から出力される回路が正常か異常かを示す信号が上記記憶素子に記憶されるように構成されていることを特徴とする半導体集積回路。

【請求項2】

少なくとも複数の論理セルからなり論理セルごとに回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能な可変論理回路と読出し書込み可能なメモリ回路とを

内蔵した半導体集積回路の検査方法であって、

上記可変論理回路により自己テストを行なわせ、その結果得られた不良個所を示す情報を用いて該可変論理回路内に、正常な論理セルで所定のアルゴリズムに従って所定のテスト信号および期待値信号を発生してテスト信号を上記メモリ回路に供給し、その結果メモリ回路から得られる出力信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成してチップ外部へ出力させるメモリテスト回路を構成して、上記メモリ回路を検査するようにしたことを特徴とする半導体集積回路の検査方法。

【請求項 3】

上記メモリ回路は、スタティック型ランダム・アクセス・メモリであることを特徴とする請求項 2 に記載の半導体集積回路の検査方法。

10

【請求項 4】

上記メモリ回路は、ダイナミック型ランダム・アクセス・メモリであることを特徴とする請求項 2 に記載の半導体集積回路の検査方法。

【請求項 5】

少なくとも複数の論理セルからなり論理セルごとに回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能な可変論理回路と、読出し書込み可能なメモリ回路と、カスタム論理回路または中央処理ユニットとを内蔵した半導体集積回路の検査方法であって、

上記可変論理回路により自己テストを行なわせ、その結果得られた不良個所を示す情報を用いて該可変論理回路内に、正常な論理セルで所定のアルゴリズムに従って所定のテスト信号および期待値を発生してテスト信号を上記メモリ回路に供給しその結果メモリ回路から得られる出力信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成してチップ外部へ出力させるメモリテスト回路を構築して、該メモリテスト回路により上記メモリ回路を検査し、

20

次に、上記可変論理回路内に、正常な論理セルで所定のアルゴリズムに従って所定のテスト信号および期待値信号を発生してテスト信号を上記カスタム論理回路または中央処理ユニットに供給しその結果カスタム論理回路または中央処理ユニットから得られる出力信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成してチップ外部へ出力させるロジックテスト回路を構築し、上記メモリ回路内にテストパターンもしくはテストパターン生成プログラムを格納して、上記ロジックテスト回路を起動させて上記カスタム論理回路または中央処理ユニットを検査するようにしたことを特徴とする半導体集積回路検査方法。

30

【請求項 6】

上記メモリ回路は、スタティック型ランダム・アクセス・メモリであることを特徴とする請求項 5 に記載の半導体集積回路の検査方法。

【請求項 7】

上記メモリ回路は、ダイナミック型ランダム・アクセス・メモリであることを特徴とする請求項 5 に記載の半導体集積回路の検査方法。

【請求項 8】

少なくとも複数の論理セルからなり論理セルごとに回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能な可変論理回路と、冗長回路を備えていない第 1 のメモリ回路と、冗長回路を備えている第 2 のメモリ回路と、カスタム論理回路または中央処理ユニットとを内蔵した半導体集積回路の製造方法であって、

40

上記可変論理回路により自己テストを行なわせ、その結果得られた不良個所を示す情報を用いて該可変論理回路内に、正常な論理セルで所定のアルゴリズムに従って所定のテスト信号及び期待値信号を発生してテスト信号を上記第 1 のメモリ回路に供給しその結果第 1 のメモリ回路から得られる出力信号と期待値とを比較して一致しない場合に不良を示す信号を形成してチップ外部へ出力させるメモリテスト回路を構築して、該メモリテスト回路により上記第 1 のメモリ回路を検査し、

次に、上記可変論理回路内に、正常な論理セルで所定のアルゴリズムに従って所定のテ

50

スト信号および期待値信号を発生してテスト信号を上記カスタム論理回路または中央処理ユニットに供給しその結果カスタム論理回路または中央処理ユニットから得られる出力信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成してチップ外部へ出力させるロジックテスト回路を構築するとともに上記第1のメモリ回路内にテストパターンもしくはテストパターン生成プログラムを格納して、上記ロジックテスト回路を起動させて上記カスタム論理回路または中央処理ユニットを検査し、

しかる後、上記可変論理回路内に、正常な論理セルで所定のアルゴリズムに従って所定のテスト信号及び期待値信号を発生してテスト信号を上記第2のメモリ回路に供給しその結果第2のメモリ回路から得られる出力信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成してチップ外部に出力させるメモリテスト回路を構築して、上記第2メモリ回路を検査し、

10

その後、上記第2のメモリ回路の検査結果に基づいて上記第2のメモリ回路に付随する冗長回路を用いて第2のメモリ回路内の欠陥ビットを予備のビットと置き替えるビット救済処理を行なうようにしたことを特徴とする半導体集積回路の製造方法。

【請求項9】

上記第1のメモリ回路はスタティック型ランダム・アクセス・メモリであり、上記第2のメモリ回路は、ダイナミック型ランダム・アクセス・メモリであることを特徴とする請求項8に記載の半導体集積回路の製造方法。

【請求項10】

上記第2のメモリ回路のビット救済処理後に、上記可変論理回路内に所望の論理を構成するようにしたことを特徴とする請求項8または9に記載の半導体集積回路の製造方法。

20

【請求項11】

上記第2のメモリ回路の検査の際に欠陥ビットの位置を示す情報を上記第1のメモリ回路内に記憶するようにしたことを特徴とする請求項8、9または10に記載の半導体集積回路の製造方法。

【発明の詳細な説明】

技術分野

本発明は、半導体集積回路（IC：Integrated Circuit）およびそのテスト技術並びに製造技術に関し、例えば故障を検出し故障箇所を回避して論理を構成可能な論理LSI（Large Scale Integration）を提供するものである。

30

背景技術

ロジックICのテスト方式としては、テストと呼ばれるテスト装置によりテストパターンデータを発生してICへ入力し、出力されたデータ信号と期待値とを比較して検出する方式が一般的であった。しかしながら、ロジックICはその論理の規模が大きくなるほどテストパターンのステップ数が長大になってしまい、テストパターンの作成およびそれを用いたテスト所要時間が非常に長くなってしまふ。

そこで、テストによるテストを容易にする方法として、ICの本来の機能を構成しているフリップフロップ等の順序回路を縦続接続してシフトレジスタを構成可能に設計しておいて、テスト時に前記シフトレジスタにテストパターンをシリアルに入力（スキャンイン）して取り込ませ（セット）、所望の組合せ論理回路にシフトレジスタに取り込んだテストデータを入力し、その後前記論理回路の出力データ信号をシフトレジスタに取り込んでシフトして外部へ取り出せる（スキャンアウト）ようにしたいわゆるスキャンパス方式と呼ばれるテスト容易化設計技術が開発され実用化されている。

40

外部からテストパターンを入力する方式では、論理回路が順序回路を含んでいると内部の状態によって出力が異なるので、ある論理回路を検査するにはその中に含まれる順序回路の状態をまずテストパターンで設定しなければならないためテストパターンが非常に長くなってしまふが、フリップフロップをシフトレジスタ構成にしてテストパターンを入力（スキャンイン）することでテストパターンを大幅に減らすことができる。

しかしながら、上記スキャンパス方式は、それまでのテスト方式に比べるとテストパター

50

ンの量が少なくなるもののテストパターンの生成が難しく不良検出率を上げにくいとともに、テストパターンをシリアルに入力（転送）することを繰り返し行なうためテスト時間が長くなると不具合もある。また、新たに開発する論理LSIがRAM（ランダム・アクセス・メモリ）、ROM（リード・オンリ・メモリ）のようなメモリ回路やCPU等の大型セル（マクロセルまたはIPコア：Intellectual Property Core）を備える場合、それらのセルについてもテストを行なおうとすると膨大なテストパターンの作成と入力が必要とされるため、事実上テストが行なえないという問題点がある。

一方、論理集積回路内に疑似乱数発生回路のようなランダムなテストパターンを発生するパターン発生回路を内蔵させたBIST（Built in self test）方式

10

のテスト技術が提案されている。しかしながら、BIST方式のテスト回路は、発生されたテストパターンが不良を検出するのに十分なテストパターンになるか保証されていない。そのため、テスト回路のテストパターンで十分な不良検出率が得られるかどうか別途検証する必要があるという課題があった。

さらに、従来のテスト回路を搭載したLSIはいずれも、そのテスト回路の自身の故障ないしは欠陥については何ら対策が施されていない。つまり、テスト回路そのものが故障した場合にはチップ本来の回路が正常であっても不良判定をしてしまうという欠点がある。そして、その対策としては、テスト回路の規模を最小にして故障や欠陥の発生を抑制する

20

しかないが、このことはテストの充分性すなわち不良検出率の向上という目的と矛盾した結果を招いてしまう。

この発明の目的は、高機能の外部テストを用いることなくLSI内部の回路のテストを行なうことが可能なテスト技術を提供することにある。

この発明の他の目的は、自分で故障箇所を検出しかつ自己救済することができる歩留まりの高いLSIを提供することにある。

この発明の他の目的は、ハードウェアのオーバーヘッドの少ない自己テスト機能を有するLSIを提供することにある。

この発明のさらに他の目的は、特に中央処理ユニットのような制御回路やメモリ回路、カスタム論理回路等を1つのチップ上に備えたいわゆるシステムLSIに適用して好適な製造技術を提供することにある。

30

この発明の前記ならびにそのほかの目的と新規な特徴については、本明細書の記述および添付図面から明らかになるであろう。

発明の開示

本願において開示される発明のうち代表的なものの概要を説明すれば、下記のとおりである。

すなわち、本発明に係る半導体集積回路は、チップ上に複数の基本論理セル（セル論理ブロック）からなり基本論理セルごとに回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能なFPGA（Field Programmable Gate Array）のような可変論理回路を搭載するようにしたものである。これにより、外部テストを使用することなく可変論理回路（FPGA）内に不良箇所があることおよびその位置を知ることができる。また、不良箇所を回避して論理を構成することにより歩留まりを向上させるとともに、この可変論理回路（FPGA）を用いてテスト回路を構築して他の内部回路をテストする場合にテスト回路自身の故障による誤ったテスト結果が出力されるのを回避することができる。

40

回路が正常か異常かを示す信号を出力可能な基本論理セルとしては、例えば相補出力を有する2線ラインロジック（論理積ゲート回路）とその出力を比較して異常の有無を判定する判定手段（排他的論理和ゲート）とからなる回路がある。基本論理セルをこのような構成とすることで、比較的小規模な回路で、回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能な可変論理回路（FPGA）を実現することができる。

50

また、本発明に係る半導体集積回路のテスト方法は、少なくとも複数の基本論理セル（セル論理ブロック）からなり基本論理セルごとに回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能な可変論理回路（FPGA）と読み出し書き込み可能なメモリ回路とを内蔵した半導体集積回路において、まず上記可変論理回路（FPGA）により自己テストを行なわせ、その結果得られた不良個所を示す情報を用いて不良個所を除いた基本論理セルのみで所定のアルゴリズムに従って所定のテスト信号および期待値信号を発生してテスト信号を上記メモリ回路に供給し、その結果メモリ回路から得られる出力信号（出力データ）と上記期待値信号（期待値データ）とを比較して一致しない場合に不良を示す信号を形成してチップ外部へ出力させるメモリテスト回路を構築するようにしたものである。

10

かかるメモリテスト回路は、所定のアルゴリズムに従ってメモリのテストパターンを生成するALPG（Algorithmic Memory Pattern Generator）と呼ばれるテストパターン生成器として構成することができる。さらに、このALPGは、命令コードを解読して演算器等を制御する公知のマイクロプログラム制御方式の制御部と、テストパターンを演算によって生成する演算器とにより構成する。そして、上記のようにFPGA内に構築されたALPGにテストパターン生成プログラムの命令コードを順次供給してテストパターンを発生させ、そのテストパターンを被テスト回路であるメモリ回路に供給して不良の有無を検査することができる。

さらに、上記テストパターン生成プログラムは、既存のテスト言語乃至はテスト言語を用いて記述され、命令コードによって定義される上記所定のアルゴリズムに従って、テストパターン（アドレスとデータ）が生成される。上記テスト言語は、アドレスとデータとを含むテストパターンを効率的に発生するための有効な命令言語と見なされる。上記テスト言語は、テスト業界で用いられているような言語とされ、たとえば、アドバンテスト社のテスト言語と互換性のある言語が用いられる。これにより、既存のテストパターンのプログラムデータが利用できるからである。上記所定のアルゴリズムを記述する言語は、テスト言語に限定されるものではなく、アドレスとデータとを含むテストパターンを発生することが可能な命令言語で有ればよい。

20

本発明に係る半導体集積回路の他のテスト方法は、半導体集積回路がメモリ回路と共にCPU（中央処理ユニット）やユーザが要求する論理機能を構成したユーザ論理回路のようなカスタム論理回路を含むような場合に、上記FPGAに所定のアルゴリズムに従ってCPUやカスタム論理回路のテスト信号および期待値信号を生成してテストを行なうテスト回路（ロジックテスト）を構築するとともに、既に前記テストにより正常と判定されたメモリ回路を使用して、そのメモリ回路にテストパターンもしくはテストパターン生成プログラムを格納し、FPGAに構築されたテスト回路を起動してメモリ回路からテストパターンもしくはテストパターン生成プログラムを読み出しながらCPUやカスタム論理回路のテストを行なうものである。

30

これによって、高機能の外部テストを用いることなくLSI内部の論理回路のテストを行なうことができる。

所定のアルゴリズムに従ってCPUやカスタム論理回路のテスト信号および期待値信号を生成してテストを行なうテスト回路は、命令コードを解読して制御信号を形成するマイクロ命令方式の制御部と、該制御部から出力される制御信号に基づいてCPUやカスタム論理回路のテスト信号および期待値信号を生成するとともにCPUやカスタム論理回路から出力される信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成する信号形成・比較手段をFPGA内に構築することで実現される。この場合、基準クロック信号に基づいて互いに位相やデューティの異なる複数のクロック信号を形成するタイミング発生回路をFPGA内に構築して、上記制御部からの制御信号に従って各テスト信号ごとにタイミングを設定できるようにするとよい。

40

このように、テスト回路は、所定のアルゴリズムに従ってCPUやカスタム論理回路のテスト信号および期待値信号を生成するための制御信号を形成するマイクロプログラム制御方式の制御部と、該制御部から出力される制御信号に基づいた内部論理回路のテスト信号

50

および期待値信号を生成するとともに内部論理回路から出力される信号と期待値信号とを比較して一致しない場合に不良を示す信号を形成する信号形成・比較手段とにより構成されるため、テストパターンをすべて内部のメモリに記憶する必要がないとともに、制御部の構成によって命令コード自身のデータ圧縮が可能となり、FPGA内に十分に構築することができる規模のものである。

発明を実施するための最良の形態

以下、本発明の好適な実施例を図面に基づいて説明する。

図1は、本発明を適用したシステムLSIの一実施例のブロック図で、公知の半導体集積回路の製造技術により単結晶シリコンのような1個の半導体チップ100上に構成される。

図1の符号110～180は上記半導体チップ100上に構成された内部回路、190はこれらの内部回路と外部装置との間の信号の入出力を行なうインタフェース回路、200は上記内部回路110～180相互間およびインタフェース回路190との間を接続する内部バスである。上記内部回路110～180のうち、110および120はユーザが要求する論理機能を構成するユーザ論理回路のようなカスタム論理回路で、このうち120はユーザが任意に論理を構成可能なFPGAにより構成されている。このカスタム論理回路は、ユーザ論理を構成せずそのまま残しておくようにしてもよい。

また、130はプログラムの命令を解読して対応する処理や演算を実行するCPU（中央処理ユニット）、140はスタティックRAM（ランダム・アクセス・メモリ）、150～180はダイナミックRAMである。さらに、この実施例のシステムLSIには、特に制限されるものでないが、内部回路のテスト時に外部のテスト500との間の信号の入出力を行なうため、IEEE1149.1規格で規定されているTAP（Test Access Port）210がインタフェース回路として設けられている。テスト500は、従来の論理LSIやメモリのテストのような高機能のものでなくデータの書き込みと読み出しおよび簡単なデータ処理が行なえるものでよく、パーソナルコンピュータを用いることも可能である。

上記CPU130は、狭義のCPUの他に、プログラムROM、ワーキングRAM、シリアルコミュニケーションインタフェースやタイマー回路、デジタル・アナログ変換回路などのいわゆるマイコン周辺回路を含んでマイクロプロセッサとして構成されていてもよい。

上記スタティックRAM140およびダイナミックRAM150～180は、内部バス200を介してアドレス信号が与えられたときに対応するメモリセルを選択するアドレスデコーダ等のメモリ周辺回路を含む。さらに、ダイナミックRAM150～180は、非アクセス時間が長くなってもメモリセルの情報電荷が失われないように周期的に疑似選択するリフレッシュ制御回路を含む。また、特に制限されるものでないが、この実施例では、ダイナミックRAM150～180には、メモリアレイ内に欠陥ビットがあった場合にその欠陥ビットを含むメモリ行もしくはメモリ列を予備のメモリ行もしくは予備のメモリ列と置きかえるいわゆる冗長回路がそれぞれ設けられている。

図2には、図1に示されている内部回路110～180のうち、カスタム論理回路120を構成するFPGAの具体例を示す。

この実施例のFPGAは、マトリクス状に配置された複数個のセル論理ブロックCLBと、各セル論理ブロックCLB間に設けられてセル間を接続するための配線群121、122と、外部から配線間接続の状態を変更することが可能なクロスポイントスイッチCSWとにより構成されている。上記クロスポイントスイッチCSWは、図3に示すように、例えば横方向の配線Lxと縦方向の配線Lyにそれぞれソート、ドレインが接続されたスイッチMOSFET Qswと、このスイッチMOSFET Qswのゲートに印加される制御情報を記憶する配線接続情報記憶用メモリセルSMCとから構成されている。上記配線群121および122は、多層配線技術を利用してそれぞれ異なる配線層によって互いに絶縁された状態で形成するのが望ましい。図2には、各セル論理ブロック間にそれぞれ2本ずつ配線が示されているが、実際にはより多くの配線が形成される。配線の数はセ

10

20

30

40

50

ル論理ブロックの数に比例して増加される。

特に制限されるものではないが、このFPGAブロックの周辺には、上記配線接続情報記憶用メモリセルSMCを選択してデータを書き込むためのXデコーダ回路やYデコーダ回路、書込み回路等を設けるようにしてもよい。また、上記配線接続情報記憶用メモリセルSMCは、スイッチMOSFET Q_{sw}と1対1で設ける代わりに、FPGAブロックの周辺にSRAMメモリアレイとして設けることも可能である。

セル論理ブロックCLBは、例えば図4(a)に示すように、AND論理とNAND論理のような相補出力を有する論理積ゲート回路(2線ラインロジック)LG1と、その相補出力を入力とする排他的論理和ゲート回路(比較手段)LG2とから構成することができる。このゲート回路LG2は、2つの入力信号が同一の論理レベルのときにロウレベルの出力信号を、また2つの入力信号の論理レベルが異なるときにハイレベルの出力信号を出力するので、論理積ゲート回路LG1に欠陥があつて相補出力となるべきところが同相出力となると、ゲート回路LG2の出力はロウレベルとなりゲート回路LG1に欠陥があることを知らせる。

このゲート回路LG2の出力信号は、そのままFPGAブロックの外部へ出力させるようにしてもよいが、この実施例では、図3に示されているクロスポイントスイッチCSWを構成する配線接続情報記憶用メモリセルSMCのノードN2に入力させて記憶できるように構成されている。

従つて、各セル論理ブロックCLBに電源電圧を与えてそのときのゲート回路LG2の出力状態を配線接続情報記憶用メモリセルSMCに記憶させてから、配線接続情報記憶用メモリセルSMCの記憶情報を外部へ読み出すことにより、セル論理ブロックCLBが正常に動作するか否かを知ることができる。また、各配線接続情報記憶用メモリセルSMCへチップ外部から順次データを書き込んで読み出すことでメモリセルが故障しているか否かも検出することができる。さらに、配線接続情報記憶用メモリセルSMCにデータを書き込んで所望のスイッチMOSFET Q_{sw}をオンさせて配線群121, 122を使用して外部より信号を入力しチェックすることで、スイッチMOSFET Q_{sw}が故障しているか否かも検出することができる。

なお、ゲート回路LG2の出力状態が入力される配線接続情報記憶用メモリセルSMCを構成するMOSFET Q1のゲート構造あるいはメモリセル回路を工夫することにより、ゲート回路LG2の出力がロウレベルのときは外部からのデータ入力によってMOSFET Q1が反転すなわちオフしないように構成してもよい。これによって、セル論理ブロックCLBに故障があるか否かを効率良く検出することができる。

かかる機能を実現する具体的方法としては、メモリセルSMCを構成するMOSFET Q1を、例えば図4(b)に示すように、コントロールゲートとフローティングゲートを有する構造とし、ゲート回路LG2の出力(ロウレベル)をインバータINVで反転してMOSFET Q1のコントロールゲートCGに高レベルの電圧V_pを印加してフローティングゲートFGに電荷を注入させ、外部からのデータ入力によってはメモリセルの状態が変化しないようにする方法が考えられる。

また、図4(a)に示すAND論理とNAND論理のような相補出力を有する論理積ゲート回路LG1としては例えば図4(c)のような回路が考えられる。すなわち、論理積ゲート回路LG1は、電源電圧端子V_{cc}と接地点との間に直列形態に接続されたMOSFET Q11~Q13からなる第1MOSFET列と、直列形態のMOSFET Q21~Q23からなる第2MOSFET列とからなり、Q12とQ21のゲートに第1の入力信号Xが供給されることにより、第2MOSFET列の出力ノードN12から入力信号X, Yの論理積出力Z(=X・Y)が、また第1MOSFET列の出力ノードN11から入力信号X, Yの論理積出力Zの反転出力/Z(=/X・Y)を出力するように動作する。なお、図に示されているMOSFET Q11およびQ23は、それぞれゲートとドレインが結合もしくはゲートに所定の電位が印加されることにより負荷として作用する。

図5(a)に上記セル論理ブロックCLBの他の構成例を示す。

この実施例のセル論理ブロックCLBは、BIST(Built in self te

10

20

30

40

50

s t) 内蔵型の論理ブロックであり、2つの入力信号X, Yをそれぞれラッチして各々正相と逆相の信号X, /X; Y, /Yを出力するフリップフロップFF1, FF2と、これら4つの出力信号を入力信号とし論理和や論理積、排他的論理和等複数の論理動作が可能な論理ユニットALUと、この論理ユニットの出力Zをラッチして正相と逆相の信号Z, /Zを出力するフリップフロップFF3と、上記論理ユニットの論理動作を指定する制御情報を記憶する読出し書き込み可能なメモリMEMと、テストパターンを乱数の形で発生する公知のLFSR(Linear Feedback Shift Register)と、コンパレータCMP等より構成されている。

上記論理ユニット制御情報を記憶するメモリMEMは外部から制御情報を書き込めるように構成されている。上記LFSRにはスイッチMOSFET G1~G7を介して入力信号X, Yと論理ユニット制御情報およびFF3の出力信号Z, /Zの伝送信号線と接続可能に構成されて、クロックCLKに同期して動作する。

この実施例のセル論理ブロックCLBを自己検査動作させる場合、上記スイッチMOSFET G1~G7のゲートに制御信号CHKを入力してオンさせる。すると、LFSRによりランダムなパターンが形成されてフリップフロップFF1, FF2および論理ユニットALUに供給されるとともに、発生されたパターンとフリップフロップFF3の出力とが論理合成、圧縮されてシグネチャ・パターンとしてコンパレータCMPに出力される。コンパレータCMPは、例えば図5(b)に示されているように、期待シグネチャ・パターンが格納されたリード・オンリ・メモリROM、イクスクルーシブORゲートEOR、出力ラッチOLT、クロックCLKよりラッチタイミング信号を発生するタイミング発生回路TMG等から構成されている。LFSRよりシグネチャ・パターンが入力されると、コンパレータCMPはイクスクルーシブORゲートEORによってリード・オンリ・メモリROMに格納されている期待シグネチャ・パターンとLFSRの出力パターンとを比較して一致しているとロウレベルの信号を、また一致していないとハイレベルの信号を出力する。この出力がラッチOLTにラッチされて良/不良を示す信号ERRとして出力される。

なお、LFSRの動作原理は既に公知であり、種々の文献等に記載されているので詳しい説明は省略するが、その原理に従い被検査論理回路に合った最適化が可能である。LFSRを適用したBISTを内蔵した一般の論理LSIでは、論理回路毎にLFSRの最適化が必要であったため設計上煩わしさがあつたが、本実施例のFPGAでは同一のセル論理ブロックCLBを使用しているため、その最適化は一律に行なうことができ、設計負担が軽減される。また、BISTを内蔵した従来のLSIは、1つのBISTがLSI内部回路全てを検査するグローバルなBISTであったため、生成されるテストパターンではテスト充分性が保証されないが、本実施例のBISTは各セル論理ブロックCLB内に設けられたローカルなBISTであるため、テスト充分性も保証される。

図6は、図1に示されているTAPを用いたインタフェース回路210の具体例を示す。TAPは前述したように、IEEE1149.1規格で規定されているスキャンテストやBIST回路のためのインタフェースおよび制御回路で、入力ポートからのテストデータを出力ポートへシフトするとき使用するバイパスレジスタ211、回路へ特定の信号を伝える場合に使用するデータレジスタ212、チップ固有の製造識別番号を設定するためのデバイスIDレジスタ213、データレジスタの選択や内部のテスト方法を制御する場合に使用するインストラクションレジスタ214、TAP回路全体を制御するコントローラ215等により構成されている。

上記データレジスタ212はオプション扱いのレジスタである。また、インストラクションレジスタ214に設定される命令には、4つの必須命令と3つのオプション命令が用意されている。コントローラ215には、専用の3つの外部端子から、テストモードを指定するためのテストモードセレクト信号TMS、テストクロックTCK、リセット信号TRSTが入力されており、これらの信号に基づいて上記レジスタ211~214やセレクト回路216~218に対する制御信号を形成する。

また、TAPにはテストデータTDIの入力端子とテスト結果データTDOの出力端子が

10

20

30

40

50

設けられており、入力されたテストデータT D Iは上記セクタ回路2 1 6を介して各レジスタ2 1 1 ~ 2 1 4または内部のスキャンパスI s c a n , B s c a nへ供給される。また、レジスタ2 1 1 ~ 2 1 4の内容および内部回路からのスキャンアウトデータは、セクタ回路2 1 7、2 1 8を介してチップ外部へ出力される。さらに、T A Pには、データレジスタ2 1 2とインストラクションレジスタ2 1 4の内容に従って内部のB I S T回路に対する信号が形成されて供給されると共に、B I S T回路から出力されたテスト結果を示す信号がセクタ回路2 1 7、2 1 8を介してチップ外部へ出力可能に構成されている。

図1の実施例のシステムL S Iでは、後に詳細に説明されるようにカスタム論理回路(F P G A) 1 2 0やC P U 1 3 0上に構築される自己テスト回路をB I S T回路とみなして、上記T A Pの有するB I S T回路用の信号入出力機能を利用して、カスタム論理回路(F P G A) 1 2 0やC P U 1 3 0に対する自己テストのための設定信号やデータを入力したり、テスト結果およびF P G A 1 2 0内のメモリセルやS R A M 1 4 0に記憶されているデータを出力したりするように構成されている。T A Pの有するスキャンテストのための機能は、図1の実施例のシステムL S Iでは使用されない。

なお、図6において、“I s c a n”は内部論理回路を構成するフリップフロップをチェーン状に結合したシフトレジスタをテストデータのスキャンパスとして使用して、内部論理回路の診断を行なうためのテストパスを意味する。また、“B s c a n”は信号入出力部(図1のインタフェース回路1 9 0)内に設けられているフリップフロップをチェーン状に結合したシフトレジスタをスキャンパスとして使用して、他の半導体集積回路との間の接続状態の診断(バウンダリスキャンテスト)を行なうためのテストパスを意味する。次に、図1に示されているシステムL S Iに本発明に係るテスト方法を適用した場合のテスト手順の一例を、図7~図11を用いて説明する。図7には、L S I全体のテスト手順の概略が、また図8~図11にはL S Iを構成する各ブロックのテスト手順の具体例が示されている。

本発明に係るテスト方法に従うと、図7に示されているように、先ずF P G A 1 2 0が前述したセル論理ブロックの機能を利用して検査され、不良の有無が判定されて不良があるときは不良個所の回避が行なわれる(ステップS 1 ~ S 3)。次に、F P G A 1 2 0内の上記不良個所を除いた部分にS R A M 1 4 0をテストするためのテスト回路(A L P G)が構築され、S R A M 1 4 0のテストが実行される(ステップS 4, S 5)。

S R A M 1 4 0に不良個所が発見されなかった場合には、F P G A 1 2 0内の上記不良個所を除いた部分にカスタム論理回路1 1 0およびC P U 1 3 0をテストするためのテスト回路(ロジックテスト)が構築され、カスタム論理回路1 1 0およびC P U 1 3 0のテストが実行される(ステップS 6 ~ S 8)。このとき、既に検査が終了しているS R A Mを利用してテストパターンもしくはテストパターン生成プログラムが格納される。

不良が発見されなかった場合には、F P G A 1 2 0内の上記不良個所を除いた部分にD R A M 1 5 0 ~ 1 8 0をテストするためのテスト回路(A L P G)が構築され、D R A M 1 5 0 ~ 1 8 0のテストが順次実行される(ステップS 9, S 1 0)。そして、不良個所が発見された場合には、それがS R A M 1 4 0あるいは外部の記憶装置に記憶されてから、D R A M 1 5 0 ~ 1 8 0に設けられている冗長回路を利用して不良ビットを救済するための救済プログラムがC P U 1 3 0に読み込まれ、C P U 1 3 0によってそのプログラムが実行されてビット救済が行なわれる(ステップS 1 1, S 1 2)。

その後、F P G A 1 2 0内の上記不良個所を除いた部分にユーザ論理等のカスタム論理の一部が構成されて、システムL S Iとして完成される(ステップS 1 3)。このステップS 1 3では、先ずT A P 2 1 0をF P G A 1 2 0内の接続情報記憶用メモリセルS M Cのアクセスモードに設定してから、ステップS 1で得られている不良個所を示す情報を利用してその不良個所を回避するようにユーザ論理を構成するデータをF P G A 1 2 0内の正常なクロスポイントスイッチのメモリセルS M Cに書き込むことによって所望の論理を構成する。

図8には、図7のフローチャートのステップS 1 ~ S 3におけるカスタム論理回路(F P

10

20

30

40

50

G A) 1 2 0 の自己検証のより詳しい手順が示されている。

本実施例のデバイス(システム L S I)に電源電圧が投入されると、F P G A 1 2 0 を構成するセル論理ブロック C L B の論理ゲート回路 L G 1 , L G 2 が能動化され、欠陥がある場合には論理ゲート回路 L G 1 の出力がロウレベルになってその出力状態が接続情報記憶用メモリセル S M C に記憶される(ステップ S 1 1 1)。

次に、テスト 5 0 0 を用いてテスト用インタフェース回路としての T A P 2 1 0 にテストモードセレクト信号 T M S やインストラクションレジスタ 2 1 4 に設定するコードを入力して、T A P 2 1 0 を、F P G A 1 2 0 内の接続情報記憶用メモリセル S M C のアクセスモードに設定する(ステップ S 1 1 2)。続いて、メモリセル S M C に対して正常を示すデータ(上記セル論理ブロック C L B の自己検証による欠陥状態を示す論理レベルと逆の論理レベル)を書き込む(ステップ S 1 1 3)。次に、上記メモリセル S M C のデータの読み出しを行なう(ステップ S 1 1 4)。

そして、読み出されたデータと書込みデータとを比較することでいずれのセル論理ブロック C L B に欠陥があるか判定する(ステップ S 1 1 5)。また、例えば上記書込みデータと逆のデータを書き込んで読み出すことによってメモリセル S M C 自身に欠陥のあるクロスポイントスイッチ C S W も検出することができる。

次に、テスト 5 0 0 において、上記判定結果に基づいて正常なクロスポイントスイッチ C S W およびセル論理ブロック C L B のマップを作成する(ステップ S 1 1 6)。作成したマップすなわちクロスポイントスイッチ C S W およびセル論理ブロック C L B の正常/異常を示す情報は、テスト 5 0 0 内の記憶装置等に格納しておく。それから、F P G A 1 2 0 上に構築する S R A M テスタ(A L P G)の H D L 記述をデータベース等から読み出してテスト 5 0 0 で論理合成等を行ない、上記マップに基づいて欠陥のあるクロスポイントスイッチ C S W およびセル論理ブロック C L B を回避して A L P G (A l g o r i t h m i c M e m o r y P a t t e r n G e n e r a t o r) を構築するデータを生成する(ステップ S 1 1 7)。そして、生成したデータをテスト 5 0 0 内の記憶装置等に格納する(ステップ S 1 1 8)。このデータは、正常なクロスポイントスイッチ C S W のスイッチ M O S F E T Q s w を、構成したい論理に応じて選択的にオンさせるデータである。

図 9 には、図 7 のフローチャートのステップ S 4 ~ S 5 における S R A M 部 1 4 0 の検査のより詳細な手順が示されている。

S R A M 部 1 4 0 の検査では、先ずテスト 5 0 0 から T A P 2 1 0 に制御信号を供給して F P G A 1 2 0 内のクロスポイントスイッチ制御情報を記憶するためのメモリセル S M C を選択状態にする(ステップ S 1 2 1)。それから、選択状態のメモリセル S M C に上記ステップ S 1 1 8 で記憶装置に記憶された A L P G を構築するデータを転送する(ステップ S 1 2 2)。これによって、F P G A 1 2 0 内に S R A M を検査するためのテストパターンを発生可能な A L P G を含むテスト回路が構築される。F P G A 1 2 0 内に構築される A L P G の構造および A L P G を構築する具体的な方法については、後に詳しく説明する。

次に、A L P G を動作させてテストパターンを発生させるためのプログラムをテスト 5 0 0 により T A P 2 1 0 を介して F P G A 1 2 0 内のメモリ回路に書き込む(ステップ S 1 2 3)。このメモリ回路は、ステップ S 1 2 2 での A L P G の構築の際に、F P G A 1 2 0 を構成するセル論理ブロック C L B とクロスポイントスイッチ C S W とによって構成されている。

続いて、テスト 5 0 0 から T A P 2 1 0 に制御信号を供給して S R A M 部 1 4 0 を選択状態にする(ステップ S 1 2 4)。そして、上記 F P G A 1 2 0 内の A L P G に起動をかけ、ステップ S 1 2 3 で書き込んだテストパターン生成プログラムを実行させてテストパターンを発生させ、発生されたテストパターンをバス 2 0 0 等を介して選択状態の S R A M 部 1 4 0 に供給してテストを行ない、そのテスト結果を T A P 2 1 0 を介して外部(テスト 5 0 0)へ出力する(ステップ S 1 2 5 , S 1 2 6)。

すると、テスト 5 0 0 は出力されたテスト結果から S R A M 部 1 4 0 内に不良があるか否

10

20

30

40

50

かを判定し良品と不良品を選別する(ステップS127)。なお、FPGA120内に構築されたALPGにより形成された書込みデータを期待値として外部へ出力させるとともに、SRAMから読み出されたデータも外部へ出力させることで、外部のテストで期待値と読み出しデータとを比較して不良の有無の判定を行なうように構成することも可能である。また、テストパターン生成プログラムをFPGA内に構成したメモリ回路に記憶される代わりに、外部からTAPを介して順次テストパターン生成プログラムを構成する命令コードを入力させるようにしてもよい。

図10には、図7のフローチャートのステップS6～S8におけるロジック部すなわちカスタム論理部110およびCPU部130の検査のより詳細な手順が示されている。

論理部110および130の検査では、まずテスト500において、FPGA120内にロジックテストを構築するためのデータを作成する(ステップS131)。このとき、図8のFPGA自己検証フローのステップS116で生成された正常なセル論理ブロックCLBとクロスポイントスイッチCSWのマップを用いて故障のある回路を回避してロジックテストを構築するようなデータが作成される。

次に、テスト500からTAP210に制御信号を供給してFPGA120内のクロスポイントスイッチ制御情報を記憶するメモリセルSMCを選択状態にする(ステップS132)。それから、選択状態のメモリセルSMCに上記ステップS131で記憶装置に記憶されたロジックテストを構築するデータを転送する(ステップS133)。これによって、FPGA120内に論理部を検査するためのテストパターンを発生可能なテスト回路が構築される。このとき、FPGA内にプログラムメモリを構成し構成されたメモリにロジックテストを動作させるためのプログラムも転送しておくようにしてもよい。FPGA120内に構築されるロジックテストの構造およびロジックテストを構築する具体的な手順については、後に詳しく説明する。

次に、テスト500からTAP210に制御信号を供給して、図9のフローの検査が終了したSRAM部140を選択する(ステップS134)。そして、予めテスト500内に用意されているカスタム論理部110を検査するためのテストパターンを発生するプログラムをテスト500によりTAP210を介してSRAM140内に書き込む(ステップS135)。SRAMのテストパターンを生成するプログラムのように、FPGA内のメモリ回路に格納せずに検査が終了したSRAM内に格納するのは、一般にメモリのテストパターンよりも論理回路のテストパターンの方が長くより大きなメモリ領域を必要とするためである。

続いて、上記FPGA120内のロジックテストに起動をかけ、ステップS135でSRAM140に書き込んだテストパターン生成プログラムを読み出してテストパターンを生成しながらカスタム論理部110に供給する(ステップS136)。そして、カスタム論理部110からの出力信号を期待値と比較してテスト結果をTAP210を介して外部(テスト500)へ出力する(ステップS137)。

すると、テスト500は出力されたテスト結果からカスタム論理部140内に不良があるか否かを判定し良品と不良品を選別する(ステップS138)。

次に、予めテスト500内に用意されているCPU130を検査するためのテストパターン生成プログラムをSRAM140へ転送する(ステップS139)。続いて、上記FPGA120内のロジックテストに起動をかけ、ステップS135でSRAM140に書き込んだテストパターン生成プログラムを読み出してテストパターンを生成しながらCPU130に供給する(ステップS140)。そして、CPU130からの出力信号を期待値と比較してテスト結果をTAP210を介して外部(テスト500)へ出力する(ステップS141)。

すると、テスト500は出力されたテスト結果からCPU130内に不良があるか否かを判定し良品と不良品を選別する(ステップS142)。

図11には、図7のフローチャートのステップS9～S12におけるDRAM部150～180の検査のより詳細な手順が示されている。

DRAM部150～180の検査では、まずテスト500において、FPGA120内に

10

20

30

40

50

DRAMを検査するためのテストパターンを発生可能なALPG (Algorithmic Memory Pattern Generator)を構築するためのデータを生成する(ステップS151)。このとき、図8のFPGA自己検証フローのステップS116で生成された正常なセル論理ブロックCLBとクロスポイントスイッチCSWのマップを用いて故障のある回路を回避してALPGを構築するようなデータが作成される。なお、DRAMを検査するALPGはSRAMを検査するALPGとほぼ同様であり、リフレッシュ動作の正常/異常を判定する処理が追加されている点が異なる。

次に、テスト500からTAP210に制御信号を供給してFPGA120内のクロスポイントスイッチ制御情報を記憶するメモリセルSMCを選択状態にする(ステップS152)。それから、選択状態のメモリセルSMCに上記ステップS151で作成されたALPGを構築するデータを転送する(ステップS153)。これによって、FPGA120内にDRAMを検査するためのALPGを含むテスト回路が構築される。FPGA120内に構築されるALPGの構造およびALPGを構築する具体的な方法は、後に詳しく説明されるSRAMを検査するためのALPGとほぼ同様である。

次に、ALPGを動作させてテストパターンを発生させるためのプログラムをテスト500によりTAP210を介してFPGA120内のメモリ回路に書き込む(ステップS154)。このメモリ回路は、ステップS153でのALPGの構築の際に、FPGA120を構成するセル論理ブロックCLBとクロスポイントスイッチCSWとによって構成されている。なお、テストパターン生成プログラムは、SRAM140内に記憶してもよいし、DRAM検査の際に外部から順次命令コードをFPGA内のALPGに入力するよう

にしてもよい。

続いて、テスト500からTAP210に制御信号を供給してSRAM部140を選択状態にして(ステップS155)、SRAM内に後述のDRAMテスト(ステップS159)で検出される欠陥ビットの位置を記憶するためのフェールメモリを構成する(ステップS156)。次に、テスト500からTAP210に制御信号を供給してCPU部130を選択状態にして(ステップS157)、DRAMの欠陥ビットを救済するための救済プログラムをCPU130内のメモリに転送する(ステップS158)。この救済プログラムはSRAM140内に入れておくようにしてもよい。

この救済プログラムは、所定の置換アルゴリズムに従って、上記DRAM150に付随して設けられている冗長回路内のアドレス変換回路に置換アドレスを設定し、欠陥ビットを含むメモリ行またはメモリ列を予備行または予備列に置きかえるものである。テスト結果に基づいて、最も適切な予備メモリ行または予備メモリ列を選択して、欠陥ビットと置き換える冗長置換アルゴリズムそれ自身は公知であり、それを使用することで、本実施例のために新規な置換アルゴリズムを必要としない。

それから、テスト500によりTAP210に制御信号を供給してDRAM150を選択状態にしてから上記FPGA120内のALPGに起動をかけ、ステップS154で書き込んだテストパターン生成プログラムを実行させてテストパターンを発生させ、発生されたテストパターンを選択状態のDRAM部150に供給してテストを行ない、その結果すなわち欠陥ビットの位置をSRAM140に構成されたフェールメモリに記憶させる(ステップS159)。なお、ステップS159でのDRAMテストでは、SRAMテストと同様のリード/ライトテストの他に、正常なリフレッシュ動作が行なわれるか否かのテストも行なわれる。

次に、CPU130に起動をかけて救済プログラムを実行させ、上記フェールメモリ(SRAM140)に記憶された欠陥ビットの情報に基づいて欠陥ビットを含むメモリ行またはメモリ列を予備行または予備列に置きかえるビット救済処理を行なう(ステップS160)。その後、正常に救済が行なわれたか否かのテストが救済プログラムの一連の動作として行なわれる(ステップS161)。そして、そのテスト結果がチップ外部のテスト500に出力される。

すると、テスト500は出力されたテスト結果からDRAM部150内に不良があるか否かを判定し良品と不良品を選別する(ステップS162)。DRAM部150のテストお

10

20

30

40

50

よびビット救済が終了すると、再びステップS159へ戻って他のDRAM部160、170および180についても同様にしてテストおよびビット救済とテスト結果の判定が行なわれる。

次に、前記SRAM140の検査の際にFPGA120内に構築されるALPGの詳細について説明する。ALPGは、例えば図12に示されているように、所定のテストパターン生成アルゴリズムに従って記述された複数のマイクロ命令群からなるマイクロプログラムが格納される命令メモリ411と、該命令メモリ411から読み出されるべきマイクロ命令の命令アドレスを指定するプログラムカウンタ412と、上記命令メモリ411から読み出されたマイクロ命令内の命令コードを解読してメモリ回路部に対する制御信号やプログラムカウンタ412等テスト回路内の内部回路に対する制御信号を形成するシーケンス制御回路413と、命令メモリ411より読み出されたマイクロ命令に従ってテストアドレスを生成するアドレス演算回路414と、読み出されたマイクロ命令に従ってテストデータおよび期待値データを生成するテストデータ生成回路415と、上記テストアドレスによってSRAM部140から読み出されたデータと上記テストデータ生成回路415により生成された期待値データとを比較し、正常な書込みがなされた否か判定する比較判定回路416等により構成されている。

上記比較判定回路416における判定結果は、データの一致/不一致を示す判定信号FとしてTAP210を介してチップの外部へ出力されるように構成されている。テストパターン生成アルゴリズム自体は公知であり、それらを適用することで、特に本実施例のために新規なアルゴリズムを必要としない。

上記命令メモリ411に格納される複数のマイクロ命令は、既存のテスト言語乃至はテスト言語を用いて記述することができる。上記複数の命令によって定義される上記所定のアルゴリズムに従って、テストパターン(アドレスとデータ)が生成される。上記テスト言語は、アドレスとデータとを含むテストパターンを効率的に発生するための有効な命令言語と見なされる。上記テスト言語は、テスト業界で用いられているような言語とされ、例えば、アドバンテスト社のテスト言語と互換性のある言語が用いられる。既存のテストパターンのプログラムデータが利用できるからである。上記所定のアルゴリズムを記述する言語は、テスト言語に限定されるものではなく、アドレスとデータとを含むテストパターンを発生することが可能な命令言語で有ればよい。

上記命令メモリ411がSRAMメモリセルによって構成される場合には、テストの実施に先立って、上記命令メモリにプログラムをロードする必要がある。FPGA120が不揮発性メモリを含む場合には、この不揮発性メモリの一部のアドレス領域に上記プログラムを格納しておき、SRAMテストモードへの移行にตอบสนองして、上記プログラムを格納するアドレス領域から上記命令メモリ411へ上記プログラムをロードするのがよい。命令メモリ411を設ける代わりに連続した命令コードを外部のテストから順次FPGA120内に構築されているALPGのシーケンス制御部414に入力してSRMAのテストパターンを発生させるように構成してもよい。

なお、本実施例が適用される図1のシステムLSIのようにメモリの他にCPU130やカスタム論理回路110を備えている場合には、図13に示すように、メモリ回路140(150~180)とCPU130及びカスタム論理回路110と、テスト回路が構築されるFPGA120との間に専用バス220および切替えスイッチSW1~SWnからなる信号切替え回路230を設けた可変バス構成としてもよい。

上記信号切替え回路230内の切替えスイッチSW1~SWnを制御する信号CSは、メモリ回路のテストの際には、ALPGにおいて命令によって発生させ、それをスイッチSW1~SWnへ与えるようにしてもよい。また、外部から供給されるテストモード指定信号に基づいて切替え制御信号CSを形成するように構成してもよい。

また、上記切替えスイッチSW1~SWnは、メモリ回路140とCPU130及びカスタム論理回路110との間、またはメモリ回路140とALPGが構築されたFPGA120との間の接続切り替えのみならず、CPU130及びカスタム論理回路110とFPGA120との間を接続できるように構成するのがよい。

10

20

30

40

50

図 1 4 には、上記命令メモリ 4 1 1 に格納されるマイクロ命令の命令フォーマットの構成例が示されている。前述のように、このマイクロ命令のフォーマットは、テスト言語に基づいている。

この実施例のマイクロ命令は、ジャンプ命令で使用する命令の飛び先番地を示す P C アドレスが格納されるアドレスフィールド M F a と、シーケンス制御コードが格納されるオペコードフィールド M F b と、命令の繰り返し数などが格納されるオペラントフィールド M F c と、アドレスやデータの出力およびリード/ライトを指示する制御コードが格納される制御フィールド M F d と、アドレス演算命令コードが格納されるアドレス演算コードフィールド M F e と、データ生成命令コードが格納されるデータ生成コードフィールド M F f 等からなる。アドレスフィールド M F a は、次に実行されるべき命令のアドレスを規定するフィールドと見なされる。

10

この実施例に使用されるテスト言語としてのマイクロ命令の特徴は、1つの命令でテスト用のアドレス演算とデータ演算とを指定して2つの演算を並列に実行する点にある。

図 1 5 には、シーケンス制御回路 4 1 3 の構成例が示されている。この実施例のシーケンス制御回路 4 1 3 は、オペコードフィールド M F b の制御コードを解読して制御信号を形成するデコーダなどからなる命令解読制御部 4 3 0 と、プログラムカウンタ 4 1 2 の値を「+1」するためのインクリメンタ 4 3 1 と、上記インクリメンタ 4 3 1 またはアドレスフィールド M F a 内の飛び先番地のいずれかを選択してプログラムカウンタ 4 1 2 へ供給するマルチプレクサ 4 3 2 と、オペラントフィールド M F c 内の繰り返し数を保持する複数のインデックスレジスタ 4 3 3 と、該インデックスレジスタ 4 3 3 の値を「-1」するためのデクリメンタ 4 3 4 と、「-1」された値を保持する複数のプレーンあるいはウェイから構成されるワーキングレジスタ 4 3 5 と、後述の j x d 命令(表 1 参照)で用いるデータ反転の有無を示すフラグ 4 3 6 と、j i n d e x 命令で用いるオペラントのプログラムカウンタ 4 1 2 への転送の有無を示す複数のフラグ 4 3 7 と、レジスタ 4 3 3 , 4 3 5 の値を選択的に上記デクリメンタ 4 3 4 に供給する複数のマルチプレクサ 4 3 8 と、デクリメンタ 4 3 4 の値をワーキングレジスタ 4 3 5 のいずれかのプレーンに分配するデマルチプレクサ 4 3 9 から構成される。

20

表 1 には、上記マイクロ命令内のオペコードフィールド M F b に定義乃至記述格納されるところの上記シーケンス制御に用いられるオペコードの種類とその内容が示されている。

【表 1】

30

コマンド	動作
nop	PC+1→PC
jindex1	jindexn(n=1~4) operand jfn=0 operand→PC
jindex2	idxn →idx idx →idxwn
jindex3	1 →jfn jfn=1 idxwn=0 PC+1 →PC
jindex4	0 →jfn idxwn ≠0 operand→PC idxwn →idx idx-1 →idxwn
jxd	jxd operand dflg=0 operand→PC 1→dflg dflg≠0 PC+1 →PC 0→dflg
jmp	jmp operand operand→PC
stop	stop

10

20

表1において、「nop」で示される命令は、プログラムカウンタ412の値をインクリメント431で「+1」してプログラムカウンタ412へ戻すことを指示するノーオペレーション命令つまりプログラムカウンタの更新以外に何の操作を行わずに次の命令に移ることを指令する命令である。

また、「jindex1」~「jindex4」は、ジャンプによる命令のループを回すために用意された命令である。メモリのパターンテストにおいては、ジャンプ命令を用いて同じ命令を何回も繰り返して実行することで命令数、すなわち、プログラム長を減らすことができる場合（例えば、アドレスを最終番地までインクリメントすることで、全メモリセルに「1」を書き込んで読み出すような場合）がある。このループ（ジャンプ）の回数を設定できるようにするために、本実施例では、インデックスレジスタ433を設けている。しかも、複数種類の判定方式を実行できるようにするために、ジャンプ命令とインデックスレジスタ433およびワーキングレジスタ435をそれぞれ4つずつ設けている。各ジャンプ命令は同じ制御内容であるので、以下「jindex1」による制御動作を説明し、他のジャンプ命令「jindex2」~「jindex4」の説明は省略される。オペコードフィールドMFbからjindex1命令が読み出されると、それが最初のjindex1命令であるか判定されて、その判定結果がフラグ437に反映される。具体的には、最初のjindex1のときはフラグjfn=0とされ、2回目以降はjfn=1とされる。

30

フラグjfn=0のときにjindex1命令が読み出されると、そのマイクロ命令のアドレスフィールドMFa内のジャンプ先アドレスとしてのPCアドレスがプログラムカウンタ412へ設定されるように、マルチプレクサ432が制御される。それによって、マイクロ命令の実行シーケンスはその番地にジャンプされるとともに、フラグjfnは「1」にセットされる。これと同時に、オペランドフィールドMFc内で定義されるループ回数がインデックスレジスタ433のidxに読み込まれる。

40

フラグjfn=1のときにjindex1命令が読み出されると、そのマイクロ命令のアドレスフィールドMFa内のPCアドレスがプログラムカウンタ412へ設定される。そして、インデックスレジスタ433のidx内のループ回数がマルチプレクサ438を介してデクリメント434に供給されデクリメント「-1」される。デクリメントされた値は、デマルチプレクサ439を介してワーキングレジスタ435のidxw1に格納さ

50

れる。そして、ワーキングレジスタ435の*idxw1*が「0」になると、マイクロ命令のアドレスフィールド*Mfa*内のPCアドレスをプログラムカウンタ412へ設定しないで代わりに、プログラムカウンタ412のアドレスをインクリメンタ431で「+1」してプログラムカウンタ412へ戻すようにマルチプレクサ432が制御される。

従って、マイクロ命令のオペコードフィールド*Mfb*に*jump*命令が定義乃至記述され、そのアドレスフィールド*Mfa*内に当該マイクロ命令のジャンプ先アドレスとしてのPCアドレスが定義乃至記述されていると、オペランドフィールド*Mfc*内で指定されたループ回数だけ同一の*jump*命令が繰り返して実行され、最後にプログラムカウンタ412がインクリメントされて、次のマイクロ命令へ進んでループから抜け出すような制御が行なわれる。

10

また、表1内の「*jump*」命令は、フラグ436内の*df1g*を参照し、その*df1g*フラグの値によってプログラムカウンタの値を操作する命令とされる。「*jump*」命令は、フラグ436内の*df1g*フラグが「0」のような第1の値のときは、オペランドをプログラムカウンタへ転送し、プログラムをオペランドの示す飛び先番地の命令へジャンプさせ、かつ、*df1g*フラグを「1」のような第2の値にセットする。一方、「*jump*」命令は、*df1g*フラグが「1」のような第2の値のときは、プログラムカウンタの値をインクリメントし、プログラムカウンタへ戻し、かつ、*df1g*フラグを「0」のような第1の値にリセットする。

「*jump*」命令は、オペランドをプログラムカウンタへ転送して、プログラムをオペランドの示す飛び先番地の命令へジャンプさせることを指令する命令である。

20

「*stop*」命令は、シーケンス制御を終了させる事を指示する停止命令である。

図16には、上記アドレス演算回路414の構成例が示されている。

この実施例のアドレス演算回路414は、大きく分けてXアドレスの生成を行なうXアドレス演算部441と、Yアドレスの生成を行なうYアドレス演算部442とにより構成されている。Xアドレス演算部441とYアドレス演算部442は、ほぼ同一の構成であるので、以下、Xアドレス演算部441の構成を説明し、Yアドレス演算部442の構成の説明を省略する。また、必要に応じて不可的なZアドレス演算部を設けることにより、部分的なパターン生成(パーシャルパターン)を行なわせるようにできる。

Xアドレス演算部441は、Xアドレスの初期値を格納する初期値レジスタ*Xhold*と、「0」を保持するゼロ設定手段443と、Xアドレスの初期値または「0」のいずれかを選択するマルチプレクサ*MUX1*と、選択された初期値または「0」を保持するベースレジスタ*Xbase*と、レジスタ*Xbase*の値を加算する第1の演算器*ALU1*と、演算器*ALU1*の演算結果または「0」または帰還値のいずれかを選択する第2のマルチプレクサ*MUX2*と、選択された値を保持するカレントレジスタ*Xcurrent*と、レジスタ*Xcurrent*の値を加算もしくは減算する第2の演算器*ALU2*と、この第2演算器*ALU2*または上記第1演算器*ALU1*の出力のいずれかを選択する第3のマルチプレクサ*MUX3*と、選択された出力を反転可能なインバータ*INV*とから構成されている。

30

このインバータ*INV*は、メモリのパターンテストにおいて、アドレス信号の切り換えノイズによる誤動作を試験する場合があり、その際にアドレス信号の反転信号を出力する必要があるため設けられる。このインバータ*INV*を使用することで、上記試験におけるアドレス信号の反転信号を容易に形成することができる。

40

特に制限されないが、この実施例では、上記Xアドレス演算部441の演算器*ALU1*、*ALU2*で生成されたXアドレスをYアドレス側へ、また、Yアドレス演算部442で生成されたYアドレスをXアドレス側へ出力できるように、それぞれの第3マルチプレクサ*MUX3*が構成されている。

これにより、複数の種類のメモリ、例えば、アドレスマルチプレックス方式のメモリおよびアドレスノンマルチプレックス方式のメモリのいずれのテスト回路としても使用できるように構成されている。つまり、命令メモリ411に格納されるマイクロ命令を書き換えるだけで、すべてのメモリに対してそれに必要なテストパターンの発生乃至検査を行なう

50

ことができる。

なお、上記Xアドレス演算部441とYアドレス演算部442の異なる点は、Xアドレス演算部441の第1演算器ALU1がオーバーフローしたときにYアドレス演算部442の第1演算器ALU1に対してはポロ-信号BRが供給されるようにされている点である。

表2には、上記マイクロ命令内の演算コードフィールドMFeに記述乃至定義格納されるところの上記Yアドレス演算部442の第1演算器ALU1でのYアドレス演算(ベース演算)に用いられる演算コードの種類とその内容が示されている。

【表2】

Ybase演算	動作
Ybase←Ybase	Non Operation
Ybase←0	0→Ybase
Ybase←Yhold	Yhold→Ybase
Ybase←Ybase+1	Ybase+1→Ybase
Ybase←Ybase-1	Ybase-1→Ybase
Ybase←Ybase+1(BR)	Xbase≠XMAX Ybase→Ybase Xbase=XMAX Ybase+1→Ybase
Ybase←Ybase-1(BR)	Xbase≠XMAX Ybase→Ybase Xbase=XMAX Ybase-1→Ybase
Ybase←Ybase+(BR)	Xbase≠XMAX Ybase+1→Ybase Xbase=XMAX Ybase+2→Ybase
Ybase←Ybase-(BR)	Xbase≠XMAX Ybase-1→Ybase Xbase=XMAX Ybase-2→Ybase

10

20

表2において、Ybase 0は、ベースレジスタYbaseの値を「0」にすることを指令する命令である。Ybase Yholdは、初期値レジスタYholdの内容をベースレジスタYbaseに入れることを指令する命令である。Ybase Ybase+1は、ベースレジスタYbaseの値をインクリメント(+1)してレジスタYbaseに戻すことを指令する命令である。Ybase Ybase+1(BR)は、ベースレジスタXbaseの値が最大値でなければYbaseの値をそのままにし、かつ、Xbaseの値が最大値であればYbaseの値をインクリメントしてレジスタYbaseに戻すことを指令する命令である。このとき、第1演算器ALU1から第2演算器ALU2へポロ-信号BRが供給される。

30

表3には、上記Xアドレス演算部441の第1演算器ALU1でのアドレス演算に用いられる演算コードの種類とその内容が示されている。

表4には、上記Yアドレス演算部442の第2演算器ALU2でのYアドレス演算(カレント演算)に用いられる演算コードの種類とその内容が示されている。

表5には、上記Xアドレス演算部441の第2演算器ALU2でのアドレス演算に用いられる演算コードの種類とその内容が示されている。

表4において、左側の欄はテスト言語を用いて記述されたALPG記述で、右側の欄はそれに対応した機能動作レベル記述である。この記述をハードウェア記述言語(HDL記述)のルールに従い記述して機能を表示できる。

40

【表3】

Xbase演算	動作
Xbase←Xbase	Non Operation
Xbase←0	0→Xbase
Xbase←Xhold	Xhold→Xbase
Xbase←Xbase+1	Xbase+1→Xbase
Xbase←Xbase-1	Xbase-1→Xbase

【表4】

50

Ycurrent;演算	動作
Ycurrent←Ycurrent	Non Operation
Ycurrent←Ybase	Ybase→Ycurrent
Ycurrent←0	0→Ycurrent
Ycurrent←Ycurrent+1	YC+1→YC
Ycurrent←Ybase+1	YB+1→YC
Ycurrent←Ycurrent-1	YC-1→YC
Ycurrent←Ybase1	Ybase-1→Ycurrent
Ycurrent←Ycurrent+1(BR)	Xbase≠XMAX Ycurrent+1→Ycurrent, Xbase=XMAX Ycurrent+2→Ycurrent
Ycurrent←Ybase+1(BX)	Xbase≠XMAX Ybase+1→Ycurrent, Xbase=XMAX Ybase+2→Ycurrent
Ycurrent←Ycurrent-1(BR)	Xbase≠XMAX Ycurrent-1→Ycurrent, Xbase=XMAX Ycurrent-2→Ycurrent
Ycurrent←Ybase-1(BR)	Xbase≠XMAX Ybase-1→Ycurrent, Xbase=XMAX Ybase-2→Ycurrent
Ycurrent←Ycurrent+1(BR)	Xcurrent≠XMAX Ycurrent+1→Ycurrent, Xcurrent=XMAX Ycurrent+2→Ycurrent
Ycurrent←Ybase+1(BR)	Xcurrent≠XMAX Ybase+1→Ycurrent, Xcurrent=XMAX Ybase+2→Ycurrent
Ycurrent←Ycurrent-1(BR)	Xcurrent≠XMAX Ycurrent-1→Ycurrent, Xcurrent=XMAX Ycurrent-2→Ycurrent
Ycurrent←Ybase-1(BR)	Xcurrent≠XMAX Ybase-1→Ycurrent, Xcurrent=XMAX Ybase-2→Ycurrent
Ycurrent←Ycurrent+1(BR)	Xbase≠XMAX Ycurrent→Ycurrent, Xbase=XMAX Ycurrent+1→Ycurrent
Ycurrent←Ybase+1(BR)	Xbase≠XMAX Ybase→Ycurrent, Xbase=XMAX Ybase+1→Ycurrent
Ycurrent←Ycurrent-1(BR)	Xbase≠XMAX Ycurrent→Ycurrent, Xbase=XMAX Ycurrent-1→Ycurrent
Ycurrent←Ybase-1(BR)	Xbase≠XMAX Ybase→Ycurrent, Xbase=XMAX Ybase-1→Ycurrent
Ycurrent←Ycurrent+1(BR)	Xcurrent≠XMAX Ycurrent→Ycurrent, Xcurrent=XMAX Ycurrent+1→Ycurrent
Ycurrent←Ybase+1(BR)	Xcurrent≠XMAX Ybase→Ycurrent, Xcurrent=XMAX Ybase+1→Ycurrent
Ycurrent←Ycurrent-1(BR)	Xcurrent≠XMAX Ycurrent→Ycurrent, Xcurrent=XMAX Ycurrent-1→Ycurrent
Ycurrent←Ybase-1(BR)	Xcurrent≠XMAX Ybase→Ycurrent, Xcurrent=XMAX Ybase-1→Ycurrent

10

20

30

【表5】

Xcurrent;演算	動作
Xcurrent←Xcurrent	Non Operation
Xcurrent←Xbase	Xbase→Xcurrent
Xcurrent←0	0→Xcurrent
Xcurrent←Xcurrent+1	Xcurrent+1→Xcurrent
Xcurrent←Xbase+1	Xbase+1→Xcurrent
Xcurrent←Xcurrent-1	Xcurrent-1→Xcurrent
Xcurrent←Xbase-1	Xbase-1→Xcurrent

40

図17には上記テストデータ生成回路415の構成例が示されている。
 この実施例のテストデータ生成回路415は、ライトデータまたは期待値の初期値を格納する初期値レジスタTpholdと、当該初期値レジスタTpholdの値を反転可能なインバータINVERT1と、出力すべきテストデータまたは期待値の基準データを保持

50

するベースデータレジスタTpと、ビットシフト機能を有する演算器ALUと、演算器ALUの出力を反転可能なインバートINVERT2とから構成されている。なお、上記実施例では、ビット幅を18ビットとしているが、必要に応じてテストデータ回路を拡大して、所望のビット幅に構成できる。

表6には、上記マイクロ命令内のデータ生成コードフィールドM F fに定義乃至は記述される上記テストデータ生成回路415での動作制御に用いられる制御コードの種類とその内容が示されている。表6において、表3～表5の命令と同一規則で表されている命令は、ほぼ同様の命令である。

Tp ← Tp * 2は、レジスタTpと演算器ALUを制御してレジスタTp内の18ビットのデータを演算器ALUで処理してビット列をMSB側もしくはLSB側へ1ビットシフトさせてレジスタTpに戻す命令である。この命令によって、メモリ部が1ワードあるいは1バイトのような単位でデータのリード・ライトが行なわれるタイプのメモリであっても、メモリセルに対して1ビットずつデータ「1」を書き込むためのテストデータを比較的容易に生成することができる。

【表6】

Tp出力	動作
Tp←Tp	Non Operation
Tp←0	0→Tp
Tp←Tphold	Tphold→Tp
Tp←Tp*2	Shift left

表7には、メモリの試験方法の1つであるところのマーチング試験に関するマイクロ命令のリストの一例が示される。

マーチング試験は、次のように行われる。メモリアレイ1内のメモリセルの全ビットにおいて、その中の1ビットが順番に選択される。選択されたメモリセルには、「0」のデータが書き込まれる。その後、書き込まれた「0」のデータの読み出しが行なわれる。続いて、全ビットにおいて、その中の1ビットが順番に選択される。選択されたメモリセルには「1」のデータが書き込まれる。その後、書き込まれた「1」のデータが読み出されて、それぞれ期待値と比較して欠陥の有無が判定される。

一例として、Xアドレスが4ビット、Yアドレスが4ビットで、256ビットの記憶容量を有するメモリアレイを想定したマーチング試験が説明される。

【表7】

PC	オペコード	PCアドレス	制御命令等
0	nop		Xbase←Xhold Ybase←Yhold Tp←Tphold
1	jindex1	1 W	X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR)
2	nop	R	X←Xbase Y←Ybase
3	jindex1	2 W	X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR) Dinvert
4	jindex1	4 R	X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR) Dinvert
5	jxd	1	

表7において、記号WおよびRは、制御フィールドM F dに記述乃至定義される場所のメモリに対するリード/ライトを指示する制御コードである。この制御コードは、Wに対応したビットとRに対応したビット、つまり、2ビットで表すようにされている。「PCアドレス」の欄に記載されている数字は、飛び先番地を示すもので、当該マイクロ命令実行後にPCの欄に記載されている番号のマイクロ命令に移行してループもしくはジャンプすることを意味している。また、「制御命令等」の欄に記載されている記号X ← Xbase, Y ← Ybaseは、それぞれベースレジスタXbase, Ybaseの値をXアドレス, Yアドレスとして出力することを意味している。

図18および図19には、前記マイクロプログラム制御方式の制御部と演算部とからなるテスト回路(ALPG)によりSRAM140のマーチング試験を行なう場合の表7のマイクロ命令リストに従った処理フローの一例を示す。このうち図18はマイクロ命令内のオペコードとオペランドとによるシーケンス制御のフローを示す。また、図19は、図1

8に示されるシーケンス制御のフローと並行して実行される場所のアドレス演算コードとデータ生成コードによるテストアドレスおよびテストデータ(ライトデータ、期待値データ)の生成フローを示す。これらの処理は、前述の表1～表6に記載されている命令コードを用いてシーケンス制御回路413およびアドレス演算回路414、データ生成回路415を動作させることにより実行することができる。

以下、図18と図19の各ステップを対応させながら、マーチング試験の手順を説明する。

外部からプログラムカウンタ412にスタートアドレスとして、表7のリストに示される最初の命令の番地がセットされると、図18および図19のフローが開始される。まず、命令メモリ411から表7の最初のマイクロ命令が読み出されると、オペコードは「nop」であるため、シーケンス制御回路413は何もせずただプログラムカウンタ412を1つだけ進める(ステップS201)。一方、アドレス演算回路414およびデータ生成回路415は、初期値レジスタXhold, Yhold, Tholdに入っている初期値「0」をベースレジスタXbase, Ybase, Tpにそれぞれ設定する(ステップS221)。

次に、上記ステップS201でプログラムカウンタ412がインクリメントされているため、表7の2番目のマイクロ命令が読み出される。第2のマイクロ命令のオペコードは「jindex1」であるため、シーケンス制御回路413ではオペランドの繰り返し回数がインデックスレジスタidx1にセットされる(ステップS202)。

インデックスレジスタidx1の最初の設定値はメモリアレイ1の容量(バイト数もしくはワード数)に対応される。一方、アドレス演算回路414では、演算コードXbase, YbaseによってベースレジスタXbase, Ybaseの値をXアドレス, Yアドレスとしてそれぞれ出力する。データ生成回路415からはレジスタTpの値「0」が出力される。そして、このとき、制御信号W(書き込み)が出力される。これによって、メモリアレイ1ではアドレス演算回路414から出力されたX, Yアドレスに対応したメモリエルが選択されてデータ「0」が書き込まれる(ステップS222)。

シーケンス制御回路413はオペコード「jindex1」によってインデックスレジスタidx1の値をデクリメントしてワーキングレジスタidxwに入れ、それが「0」になるまで同一命令を繰り返す(ステップS203, S204)。

また、アドレス演算回路414では演算コードXbase, Xbase+1によってベースレジスタXbaseの値すなわちXアドレスをインクリメントし、Ybase, Ybase+1(BR)によってベースレジスタXbaseが最大値になるとYアドレスをインクリメントする。上記インデックスレジスタidx1の最初の設定値はメモリアレイ1の容量と同じであるので、上記動作の繰り返しによって、メモリアレイ1内のすべてのメモリエルに順番にデータ「0」が書き込まれる(ステップS223, S224)。

ワーキングレジスタidxwの値が「0」になると、シーケンス制御回路413はプログラムカウンタ412の値をインクリメントして、表7の3番目のマイクロ命令を読み出す。そのオペコードは「nop」であるため、シーケンス制御回路413は何もせずプログラムカウンタ412を1つだけ進める(ステップS205)。

一方、アドレス演算回路414は、3番目のマイクロ命令の演算コードに従って、X, YアドレスをインクリメントしてからベースレジスタXbase, Ybaseに入っている値をXアドレス, Yアドレスとしてそれぞれ出力する。このときベースレジスタXbase, Ybaseの値は「0」に戻っている。データ生成回路415はこのとき「0」を出力する。また、制御フィールドMfdの制御コードに従って、制御信号R(読み出し)が出力される。これによって、当該アドレスのデータが読み出され、期待値データ「0」と比較される(ステップS225, S226)。

次に、上記ステップS205でプログラムカウンタ412がインクリメントされているため、表7の4番目のマイクロ命令が読み出される。第4のマイクロ命令のオペコードは「jindex1」であるため、シーケンス制御回路413ではオペランドの繰り返し回数がインデックスレジスタidx1にセットされる(ステップS206)。インデックスレ

10

20

30

40

50

レジスタ $id \times 1$ の最初の設定値はメモリアレイ 1 の容量 (バイト数もしくはワード数) に対応される。ただし、第 4 命令の $jindex \times 1$ 命令コードは第 2 命令の $jindex \times 1$ 命令コードと異なる PC アドレス (飛び先番地) を有するため、実行後に第 3 の命令 (オペコードが「nop」) に戻る。

一方、アドレス演算回路 414 では、演算コード $X \quad Xbase$, $Y \quad Ybase$ によってベースレジスタ $Xbase$, $Ybase$ の値を X アドレス, Y アドレスとしてそれぞれ出力する。また、データ生成コード $Dinvert$ によって、データ生成回路 415 からはレジスタ Tp の値「0」を反転したデータ「1」が出力される。そして、このとき、制御信号 W (書き込み) が出力される。これによって、メモリアレイ 1 では出力された X, Y アドレスのメモリセルにデータ「1」が書き込まれる (ステップ S227)。

シーケンス制御回路 413 はオペコード「 $jindex \times 1$ 」によってインデックスレジスタ $id \times 1$ の値をデクリメントしてワーキングレジスタ $id \times w$ に入れ、それが「0」になるまで「nop」命令に戻って繰り返す (ステップ S207, S208)。また、アドレス演算回路 414 では演算コード $Xbase \quad Xbase + 1$ によってベースレジスタ $Xbase$ の値すなわち X アドレスをインクリメントし、 $Ybase \quad Ybase + 1$ (BX) によってベースレジスタ $Xbase$ が最大値になると Y アドレスをインクリメントする。上記インデックスレジスタ $id \times 1$ の最初の設定値はメモリアレイ 1 の容量と同じであるので、上記手順によって、メモリアレイ 1 内のすべてのメモリセルに対して順番に読み出しが行なわれ期待値と比較されてから当該アドレスに「1」が書き込まれる動作を繰り返す (ステップ S227, S228)。

次に、表 7 の 5 番目のマイクロ命令が読み出される。第 5 のマイクロ命令のオペコードは「 $jindex \times 1$ 」であるため、シーケンス制御回路 413 ではオペランドの繰り返し回数がインデックスレジスタ $id \times 1$ にセットされる (ステップ S209)。インデックスレジスタ $id \times 1$ の最初の設定値はメモリアレイ 1 の容量 (バイト数もしくはワード数) に対応される。この第 5 の $jindex \times 1$ 命令コードは第 2 の $jindex \times 1$ 命令コードと同じ PC アドレスを有し、同一命令を繰り返す命令とされる。一方、アドレス演算回路 414 では、演算コード $X \quad Xbase$, $Y \quad Ybase$ によってベースレジスタ $Xbase$, $Ybase$ の値を X アドレス, Y アドレスとしてそれぞれ出力する。また、データ生成コード $Dinvert$ によって、データ生成回路 415 からはレジスタ Tp の値「0」を反転したデータ「1」が出力される。そして、このとき、制御信号 R (書き込み) が出力される。これによって、メモリアレイ 1 では出力された X, Y アドレスによって指定されたメモリセルからデータの読み出しが行われる (ステップ S229)。

シーケンス制御回路 413 はオペコード「 $jindex \times 1$ 」によってインデックスレジスタ $id \times 1$ の値をデクリメントしてワーキングレジスタ $id \times w$ に入れ、それが「0」になるまで「 $jindex \times 1$ 」命令を繰り返す (ステップ S210, S211)。また、アドレス演算回路 414 では演算コード $Xbase \quad Xbase + 1$ によってベースレジスタ $Xbase$ の値すなわち X アドレスをインクリメントし、 $Ybase \quad Ybase + 1$ (BR) によってベースレジスタ $Xbase$ が最大値になると Y アドレスをインクリメントする。上記インデックスレジスタ $id \times 1$ の最初の設定値はメモリアレイ 1 の容量と同じであるので、上記動作の繰り返しによって、メモリアレイ 1 内のすべてのメモリセルが順番に読み出しが行なわれて期待値「1」と比較される (ステップ S230, S231)。

次に、表 7 の 6 番目のマイクロ命令が読み出される。第 6 のマイクロ命令のオペコードは「 jxd 」であるため、シーケンス制御回路 413 ではアドレスフィールド Mfc 内の PC アドレスの示す飛び先番地へジャンプする。この実施例では、表 7 の第 2 の命令「 $jindex \times 1$ 」へ戻るようにされている。そして、再び $jindex \times 1$ 命令が読み出されたときにフラグ $dflg$ が「1」にセットされる。これにより、2 巡目のシーケン処理においては、図 18 のステップ S212 で「YES」と判定して、マーチング試験を終了する。

一方、2 巡目のマイクロ命令リストの実行に際しては、図 19 のステップ S222 ~ S231 において、データ生成回路 414 によりライトデータと期待値データを「1」として

10

20

30

40

50

全ビットへ連続して書き込みを行なってから、書き込んだデータを1ビットずつ読み出してそこへ反転データ「0」を書き込み、その後、全ビットの連続した読み出しが行なわれるようにされている。図19のフローでは、ステップS232で、ステップS222～S231による上記のような1巡目とは逆のデータの書き込み読み出しが終了したか否かの判定（裏データ終了判定）が行なわれるようにされている。なお、上記裏データによる検査を行なうための裏データ生成は図17のThresholdのINVERTとdf1gでコントロールして生成するようにされている。

上述したように、図18および図19のフローに従った制御は、表1～表6に記述されている命令コードを用いて命令メモリ411内のマイクロ命令を次の表7の命令リストのように記述することで、僅か6つのマイクロ命令によりマーチング試験を実行することができる。

10

さらに、前記実施例のテスト回路（ALPG）においては、表1～表6に記述されている命令コードを用いることにより上記マーチング試験の他に、すべてのメモリセルに「0」または「1」が書き込めるか検査するオール「0」判定試験またはオール「1」判定試験や、あるビットに「1」を書き込んでそれによって他のビットが誤書き込みされていないかすべてのビットについて検査するいわゆるN²乗パターン試験（もしくはギャロッピング試験）等も実行することができる。

表8に、N²乗パターン試験のためのマイクロ命令リストを示す。なお、ixdは、初期値レジスタTpholdの下にあるフラグdf1gでコントロールされるインバータINVERTをコントロールして裏テストパターンを生成する命令である。フラグdf1gが0のときは初期値レジスタTpholdの値をそのままレジスタTpに格納し、フラグdf1gが1のときはその逆のパターンがレジスタTpに設定される。

20

【表8】

PC	命令コード	PCアドレス	制御命令等
0	nop		Xbase←Xhold Ybase←Yhold Tp←Tphold
1	jindex1	1	W X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR)
2	jindex1	2	R X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR)
3	nop		W X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR) Xcurrent←Xbase Ycurrent←Ybase Dinvert
4	jindex1	4	R X←Xbase Y←Ybase Xbase←Xbase+1 Ybase←Ybase+1(BR)
5	jindex2	3	R X←Xcurrent Y←Ycurrent Xcurrent←Xcurrent+1 Ycurrent←Ycurrent+1(BR) Dinvert
6	jxd	1	

30

上記実施例においては、プログラムカウンタ412に最初に設定する値を、表7や表8に示される最初のマイクロ命令の命令番地に変更することで、マーチング試験やN²乗パターン試験を開始させることができる。

次に、前記カスタム論理部110およびCPU130の検査の際にFPGA120内に構築されるロジックテストの詳細について説明する。

40

ロジックテストは、メモリのテストパターンを生成する図12に示されているALPGと類似の構成を有し、例えば図20に示されているように、所定のテストパターン生成アルゴリズムに従って記述された複数のマイクロ命令群からなるマイクロプログラムが格納された命令メモリ411と、該命令メモリ411から読み出すべきマイクロ命令を指定するプログラムカウンタ412、命令メモリ411から読み出されたマイクロ命令内の命令コードを解釈して上記プログラムカウンタ412など制御部400内の回路に対する制御信号を形成する命令解釈制御回路430、基準クロック0に基づいてタイミング制御信号を形成するタイミング発生部420、マイクロ命令内のタイミング設定ビットMfd（TSビット）に基づいてタイミング発生部420に対する制御データを出力するデータレジスタセット417、マイクロ命令内のタイミング設定ビットMfd（TSビット）をデコ

50

ードしてデータレジスタセット417から制御データを読み出すデコーダ418などを備えている。

また、テスト対象となる論理回路の内、その機能が特定されている回路（例えば、ALU：Arithmetic Logic Unit）の場合には、すでに適切なテストパターン形成方式が確立されている場合が多いので、そのテストパターンの資産を利用することで、効率の良いテストパターンの生成が可能である。また、組合せ論理回路に関しては、故障仮定法および一つの回路には故障は一つであるという単一故障という考えに基づくDアルゴリズムと呼ばれる効率の良いテストパターンの生成方法が知られている。この手法を利用することによって、テストパターン生成のためのマイクロプログラムを短くすることができ、命令メモリ411の容量の増大を実現可能な程度まで抑えることができる。この実施例では、特に制限されないが、デコーダ418でデコードされるタイミング設定ビットTSは2ビットで構成され、データレジスタセット417には7個の制御データが格納されている。これらの制御データのうち一つはテスト・サイクルを規定するデータ“RATE”、残りの6個の制御データは、テスト専用バスの各信号線ごとにハイレベルもしくはロウレベルの信号の出力タイミングを与える2種類の制御データ“ACLK1”、“ACLK2”と、パルス信号の立上がりタイミングを与える2種類の制御データ“BCLK1”、“BCLK2”と、パルス信号の立下がりタイミングおよび期待値との比較出力タイミングを与える2種類の制御データ“CCLK1”、“CCLK2”である。

これらの各制御データが、タイミング発生部420に供給されると、制御データRATEに関しては予め規定されたタイミングの信号RATEがプログラムカウンタ412に供給されて命令メモリ411からのマイクロ命令コードの取り込みが行なわれる。また制御データとして“ACLK1”～“CCLK2”がタイミング発生部420に供給されると、タイミングクロックACLK1～CCLK2の中からその制御コードに対応するクロックが信号形成・比較部300に出力される。各クロックの使用のための接続や選択は必要に応じて適宜実施される。

さらに、ロジックテストには、上記プログラムカウンタ412の値を「+1」にインクリメントするためのインクリメント421や、上記インクリメント421またはアドレスフィールドMfa内の飛び先番地のいずれかを選択してプログラムカウンタ412へ供給するマルチプレクサ422、オペランドフィールドMfc内の繰り返し数を保持するインデックスレジスタ423、該インデックスレジスタ423の値を「-1」するためのデクリメント424、「-1」にデクリメントされた値を保持するワーキングレジスタ425、所定の命令で用いられるオペランドのプログラムカウンタ412への転送の有無を示すフラグ427、レジスタ423、425の値を選択的に上記デクリメント424に供給するマルチプレクサ428、デクリメント424の値をワーキングレジスタ425のいずれかのプレーンに分配するデマルチプレクサ429などが設けられている。

このロジックテストでは、マイクロ命令コードに命令の繰り返し数を格納するオペランドフィールドMfcを設けるとともに、その繰り返し数を保持するインデックスレジスタ423を設けているので、同一テスト信号を繰り返し生成するような場合に、必要なマイクロ命令数を減らしマイクロプログラムを短くすることができる。また、この実施例では、インデックスレジスタ423やワーキングレジスタ425、フラグ427が複数プレーン（図では4個）設けられていることにより、あるループ処理内におけるサブループ処理、さらにそのサブループ処理内におけるサブループ処理といったことを容易に実行することができ、マイクロプログラムを短くすることができる。

図21には、上記信号形成・比較部300の実施例が示されている。なお、図21の回路は、テスト専用バス220を構成する信号線のうちの1本に対応するドライバ/コンパレータ回路のみが代表的に示されているが、実際にはテスト専用バス220を構成する信号線の数だけ図21に示す回路が設けられる。

図21に示すように、この実施例のドライバ/コンパレータ回路は、テスト専用バスへ出力する信号を形成するドライバ回路（信号形成回路）340と、テスト専用バス上の信号と期待値信号とを比較して一致/不一致を比較するコンパレータ回路（比較回路）350

10

20

30

40

50

と、ドライバ回路340とコンパレータ回路350とを切り替える切替え回路360とから構成される。切替え回路360は、ドライバ回路340と入出力ノードNioとの間に設けられた伝送ゲートTG1と、入出力ノードNioとコンパレータ回路50と間に設けられた伝送ゲートTG2とから構成され、上記制御部400から供給される入出力制御ビットI/Oに応じていずれか一方が開かれ他方は遮断状態とされる。

ドライバ回路340は、タイミング発生部420から供給されるタイミングクロックACKiによって入出力制御ビットTPを取り込んで保持するエッジトリガ型フリップフロップ341と、タイミング発生部420から供給されるタイミングクロックBCKiとCCKiとの論理和をとるORゲート342と、このORゲート342の出力と上記エッジトリガ型フリップフロップ341の出力を入力信号とするJ/Kフリップフロップ343と、このJ/Kフリップフロップ343の出力と制御部400から供給される入出力制御ビットCONTとを入力信号とするANDゲート344と、上記エッジトリガ型フリップフロップ341の出力と制御部400から供給される入出力制御ビットCONTとを入力信号とするANDゲート345と、これらのANDゲート344, 345の出力によってテスト専用バスを駆動するドライバ346とから構成されている。

一方、コンパレータ回路350は、タイミング発生部420から供給されるタイミングクロックCCKiと制御部400から供給される入出力制御ビットCONTとを入力信号とするANDゲート351と、上記D型フリップフロップ341の出力(期待値)と伝送ゲートTG2を介して供給されるテスト専用バス上の信号とを入力信号とするエクスクルーシブORゲート352と、このエクスクルーシブORゲート352と上記ANDゲート351との出力を入力信号とするANDゲート353と、このANDゲート353の出力をラッチするフリップフロップ354とから構成されており、すべてのコンパレータ回路350の出力の回路の論理和をとった信号がトータル・フェイル信号TFLとして出力される。上記入出力制御ビットI/O、TP、CONTは、上記制御信号に相当する。

図20に示されているように、本実施例のテスト回路におけるマイクロ命令は、ジャンプ命令で使用する命令の飛び先番地を示すPCアドレスが格納されるアドレスフィールドMfaと、シーケンス制御コードが格納されるオペコードフィールドMfbと、命令の繰り返し数などが格納されるオペランドフィールドMfcと、上記データレジスタセット14からタイミング発生部420に対する制御信号を読み出すためのタイミング設定ビットTSが格納されるタイミング設定フィールドMfdと、上記信号形成・比較部300の入出力制御ビットが格納される入出力制御フィールドMfeとからなる。

上記タイミング設定フィールドMfdに格納されるタイミング設定ビットTSは、前述したようにこの実施例では2ビットであるが、3ビット以上設けてもよい。また、上記入出力制御フィールドMfeに格納される入出力制御ビットは、テスト専用バス220のn本の信号線に対応して、ドライバ・ビットTPとI/Oビットとコントロール・ビットCONTの3ビットを1セットとし、nセットだけ設けられている。これらのビットのうち、I/Oビットは入力か出力かを指定する制御ビットで“1”のときは伝送ゲートTG1を開きかつTG2を遮断してドライバの出力信号をテスト専用バス220の対応する信号線上へ出力し、“0”のときは伝送ゲートTG1を遮断しかつTG2を開いてテスト専用バス220の対応する信号線上の信号を比較用のゲート352へ入力させる。ドライバ・ビットTPおよびコントロール・ビットCONTは、その組合せに応じてハイ出力またはロウ出力か、正パルスもしくは負パルスの出力か、入力無効状態か、出力ハイインピーダンス状態かを指定する。

表9には、上記入出力制御ビットTP, I/O, CONTと信号形成・比較部300の動作状態との関係が示されている。

【表9】

10

20

30

40

出力テスト信号	TP	I/O	CONT	備考
1	1	1	1	Hiドライブ
0	0	1	1	Lowドライブ
P	1	1	0	Positive Pulse
N	0	1	0	Negative Pulse
H	1	0	1	Hi期待
L	0	0	1	Low期待
X	1	0	0	don't care
Z	0	0	0	Reserved(Hi-Z)

10

表9に示されているように、入出力制御ビットTP, I/O, CONTが「111」のときはドライバ回路340がハイレベルの信号を出力し、「011」のときはドライバ回路340がロウレベルの信号を出力し、「110」のときはドライバ回路340が正のパルス信号を出力し、「110」のときはドライバ回路340が負のパルス信号を出力するように制御が行なわれる。また、入出力制御ビットTP, I/O, CONTが「101」のときはコンパレータ回路350がハイレベルの入力信号を期待し、「001」のときはコンパレータ回路350がロウレベルの入力信号を期待し、「100」のときは入力信号を無効とするように制御が行なわれる。

なお、この実施例の信号形成・比較部300では、制御ビットTP, I/O, CONTが「000」となる状態は何ら意味を持たないように構成されている。ただし、制御ビットTP, I/O, CONTが「000」のときは、例えば伝送ゲートTG1を閉じてTG2を開き、かつエクスクルーシブORゲート352を上記ハイレベルとロウレベルの間にある2つのレベルで動作するシュミット回路としてその2つのレベル間にテスト専用バス220に接続された入出力ノードNiOの電位が存在する状態(ハイインピーダンス状態)を比較できるように信号形成・比較部300を構成しておくことも可能である。

20

図22には上記実施例におけるタイミング発生部420より供給されるタイミングクロックACLK1~CCLK2と信号形成・比較部300からテスト専用バス220上に出力される信号の一例が示されている。図22において、(a)は外部から供給される基準クロック0を、(b)~(g)はタイミングクロックACLK1~CCLK2の波形を、(h)は表9の出力テスト信号として「1」が指定されかつクロックとしてACLK1が選択された端子の出力信号の波形を示す。また、(i)は表9の出力テスト信号として「0」が指定されかつクロックとしてACLK2が選択された端子の出力信号の波形を示す。また、(j)は表9の出力テスト信号として「P」が指定されかつクロックとしてBCLK1, CCLK1が選択された端子の出力信号の波形を示す。さらに、(k)は表9の出力テスト信号として「N」が指定されかつクロックとしてBCLK2, CCLK2が選択された端子の出力信号の波形を示す。

30

図22から分かるように、入出力制御ビットTP, I/O, CONTが「111」に設定されクロックACLK1が指定された端子からはクロックACLK1に従い図22(h)のようなハイレベルの信号が出力され、TP, I/O, CONTが「011」に設定されクロックACLK2が指定された端子からはクロックACLK2に従い図22(i)のよ

40

うなロウレベルの信号が出力され、TP, I/O, CONTが「110」に設定されクロックACLK1, BCLK1, CCLK1が指定された端子からはクロックACLK1でセットされたデータに従いBCLK1, CCLK1をエッジとする図22(j)のような正パルスが出力され、TP, I/O, CONTが「010」に設定されクロックACLK2, BCLK2, CCLK2が指定された端子からはクロックACLK2でセットされたデータに従いBCLK2, CCLK2をエッジとする図22(k)のような負パルスが出力される。

また、図示しないが、入出力制御ビットTP, I/O, CONTが「101」に設定されクロックCCLK1が指定された端子では期待値をハイレベルとして図22(f)のクロックCCLK1をストローブ信号として比較が行なわれ、TP, I/O, CONTが「0

50

01」に設定されクロックCLK2が指定された端子では期待値をロウレベルとし図22(g)のクロックCLK2をストロブ信号として比較が行なわれる。なお、クロックの選択は上記に限定されず任意の組合せとすることができる。

以上説明したように上記実施例においては、半導体チップ上に複数の基本論理セル(セル論理ブロック)からなり基本論理セルごとに回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能なFPGA(Field Programmable Gate Array)のような可変論理回路を搭載するようにしたものである。これにより、外部テストを使用することなく可変論理回路(FPGA)内に不良個所があることおよびその位置を知ることができる。また、不良個所を回避して論理を構成することにより歩留まりを向上させるとともに、この可変論理回路(FPGA)を用いてテスト回路を構築して他の内部回路をテストする場合にテスト回路自身の故障による誤ったテスト結果が出力されるのを回避することができるという効果がある。

10

また、本発明に係る半導体集積回路のテスト方法は、回路が正常か異常かを示す信号を出力可能でかつ任意の論理を構成可能な可変論理回路(FPGA)と読み書き可能なメモリ回路とを内蔵した半導体集積回路において、まず上記可変論理回路(FPGA)により自己テストを行なわせ、その結果得られた不良個所を示す情報を用いて不良個所を除いた基本論理セルのみで所定のアルゴリズムに従ってメモリ回路を検査するメモリテスト回路を構築して上記メモリ回路を検査するようにした。これによって、高機能の外部テストを用いることなくLSI内部のメモリ回路のテストを行なうことができるという効果がある。

20

さらに、本発明に係る半導体集積回路の他のテスト方法は、半導体集積回路がメモリ回路と共にCPU(中央処理ユニット)やユーザが要求する論理機能を構成したユーザ論理回路のようなカスタム論理回路を含むような場合に、上記FPGAに所定のアルゴリズムに従ってCPUやカスタム論理回路のテスト信号および期待値信号を生成してテストを行なうテスト回路(ロジックテスト)を構築するとともに、既に前記テストにより正常と判定されたメモリ回路を使用して、そのメモリ回路にテストパターンもしくはテストパターン生成プログラムを格納し、FPGAに構築されたテスト回路を起動してCPUやカスタム論理回路のテストを行なうようにした。これによって、高機能の外部テストを用いることなくLSI内部のCPUや論理回路のテストを行なうことができるという効果がある。

さらに、本発明に係る半導体集積回路の製造方法は、まず上記可変論理回路により自己テストを行なわせ、その結果得られた不良個所を示す情報を用いて該可変論理回路内に、不良個所を除いた基本論理セルのみで所定のアルゴリズムに従って第1のメモリ回路を検査するメモリテスト回路を構築して、上記第1のメモリ回路を検査し、

30

次に、上記可変論理回路内に、不良個所を除いた基本論理セルのみで所定のアルゴリズムに従ってカスタム論理回路または中央処理ユニットを検査するロジックテスト回路を構築するとともに上記第1のメモリ回路内にテストパターンもしくはテストパターン生成プログラムを格納して、上記カスタム論理回路または中央処理ユニットを検査し、

しかる後、上記可変論理回路内に、不良個所を除いた基本論理セルのみで所定のアルゴリズムに従って第2のメモリ回路を検査するメモリテスト回路を構築して、上記第2のメモリ回路を検査し、

40

その後、上記第2のメモリ回路の検査結果に基づいて上記第2のメモリ回路に付随する冗長回路を用いて第2のメモリ回路内の欠陥ビットを予備のビットと置き替えるビット救済処理を行なうようにしたので、メモリ回路のビット救済を自動的に行なうことができ、半導体集積回路の歩留まりおよび生産性を向上させることができるという効果がある。

また、本発明に係る半導体集積回路の製造方法は、上記第2のメモリ回路のビット救済処理後に、上記可変論理回路内に所望の論理を構成するようにしたので、テスト終了後にLSIチップ内には無駄な回路が残らないので、テスト回路をLSI内部に設けることに伴うハードウェアのオーバーヘッドを減らすことができるという効果がある。

さらに、本発明に係る半導体集積回路の製造方法は、上記第2のメモリ回路の検査の際に欠陥ビットの位置を示す情報を上記第1のメモリ回路内に記憶するようにしたので、上記

50

第1のメモリ回路内に記憶されている情報を読み出すことにより不良解析を容易に行なうことができるという効果がある。

以上本発明者によってなされた発明を実施例に基づき具体的に説明したが、本発明は上記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば上記実施例においては、セル論理ブロックCLB内の回路が正常か否か検出して出力する排他的論理和ゲート回路LG2の出力状態をクロスポイントスイッチ回路CSW内のメモリセルSMCに記憶させるようにしているが、LSIチップの外部へ出力させるように構成してもよい。

また、上記実施例においては、DRAM150～180のテストの際に欠陥ビットの位置を記憶するフェールメモリをSRAM140内に構成するようにしているが、外部のテスト内にフェールメモリを設けるようにしてもよい。さらに、上記実施例ではテスト用のインタフェース回路としてIEEEで規定されているTAPを利用しているが、それに限定されるものでなく、本実施例に適した専用のインタフェース回路を別途設計して設けるようにしてもよい。

産業上の利用可能性

以上の説明では主として本発明者によってなされた発明をその背景となった利用分野であるCPUとSRAM、DRAMおよびカスタム論理回路を備えたシステムLSIにおける自己検証方法を例にとって説明したが、この発明はそれに限定されず、CPUとSRAMおよびカスタム論理回路を備えた論理LSI、CPUとDRAMおよびカスタム論理回路を備えた論理LSI、CPUとカスタム論理回路を備えた論理LSI、SRAMとDRAMおよびカスタム論理回路を備えた論理LSIさらには上記のようなデジタル回路の他にアナログ回路が搭載された半導体集積回路にも利用することができる。

【図面の簡単な説明】

図1は、本発明を適用した論理LSIの一実施例の全体構成を示すブロック図である。

図2は、本発明を適用した論理LSI内に設けられる可変論理回路(FPGA)の一実施例を示す回路構成図である。

図3は、FPGAを構成するクロスポイントスイッチの具体例を示すブロック図である。

図4は、FPGAを構成するセル論理ブロックの具体例を示す論理回路図および概念図である。

図5は、FPGAを構成するセル論理ブロックの他の例を示す論理回路図である。

図6は、本発明を適用した論理LSI内に設けられるテスト用インタフェース回路の一構成例を示すブロック図である。

図7は、本発明を適用した論理LSIにおける内部回路の検査手順の一例を示すフローチャート図である。

図8は、図7のフローチャートのステップS1～S3におけるFPGA部の検査手順の具体的内容を示すフローチャートである。

図9は、図7のフローチャートのステップS4～S5におけるSRAM部の検査手順の具体的内容を示すフローチャートである。

図10は、図7のフローチャートのステップS6～S8におけるカスタム論理回路およびCPU部の検査手順の具体的内容を示すフローチャートである。

図11は、図7のフローチャートのステップS9～S12におけるDRAM部の検査手順の具体的内容を示すフローチャートである。

図12は、FPGA内に構築されるALPGの構成を示すブロック図である。

図13は、ALPGを使用してメモリ回路をテストする際のバスの構成例を示すブロック図である。

図14は、ALPGの命令メモリに格納されるマイクロ命令の命令フォーマットの構成例を示す説明図である。

図15は、ALPGのシーケンス制御部の構成例を示すブロック図である。

図16は、ALPGのアドレス演算部の構成例を示すブロック図である。

図17は、ALPGのテストデータ生成演算部の構成例を示すブロック図である。

10

20

30

40

50

図18は、ALPGによりメモリ回路の検査を行なう場合のマイクロ命令のオペコードおよびオペランドを用いたシーケンス制御手順の一例を示すフローチャートである。

図19は、ALPGによりメモリ回路の検査を行なう場合のマイクロ命令の演算コードおよびデータ生成コードを用いたテストアドレスおよびデータの生成、出力手順の一例を示すフローチャートである。

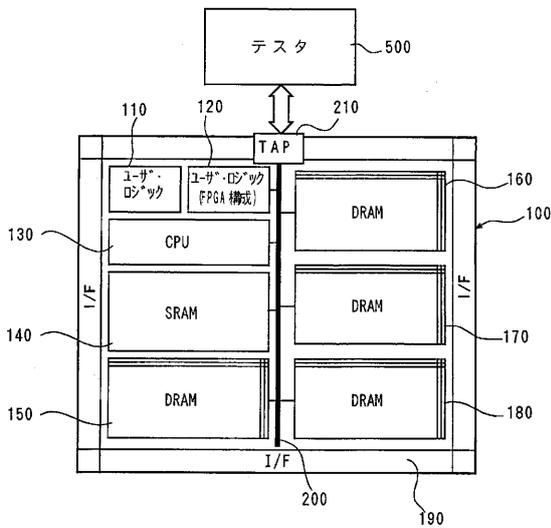
図20は、FPGA内に構築されるロジックテストの構成例を示すブロック図である。

図21は、ロジックテスト内の信号形成・比較手段の具体例を示す論理回路図である。

図22は、ロジックテストにより形成されるテスト用信号波形の一例を示す波形図である。

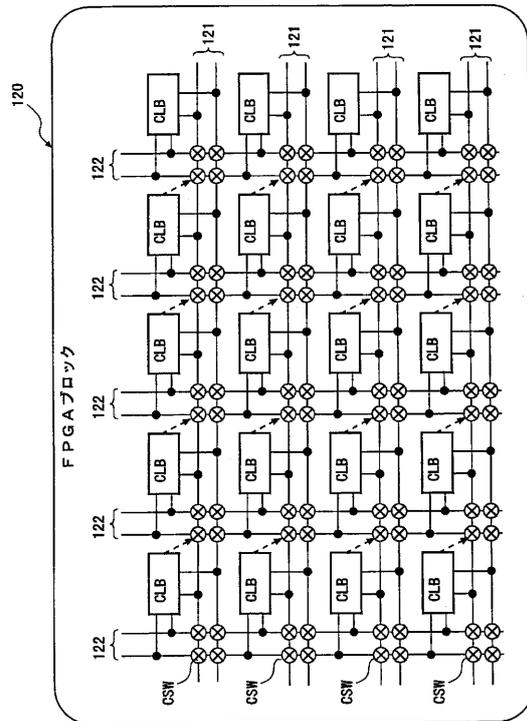
【図1】

図1



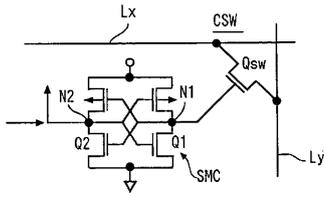
【図2】

図2



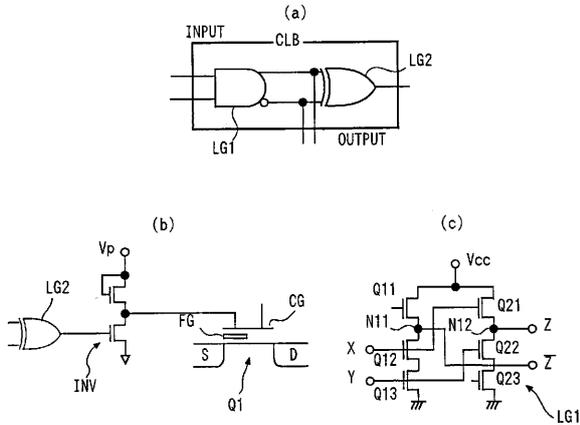
【図3】

図3



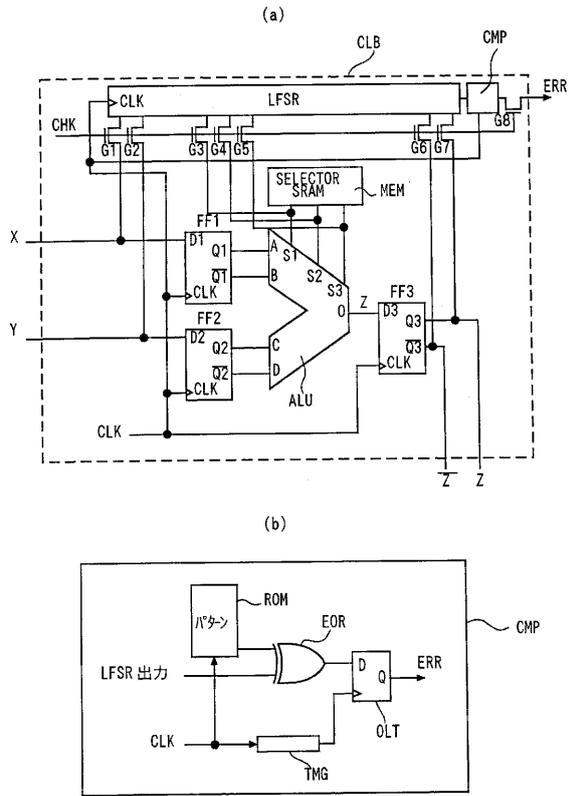
【図4】

図4



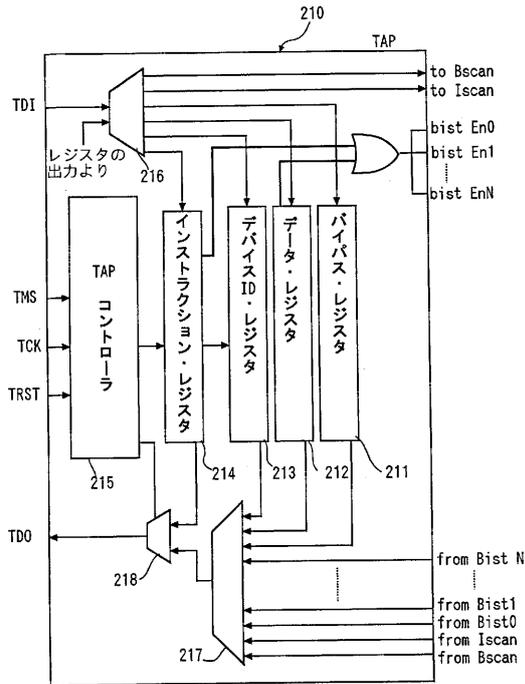
【図5】

図5



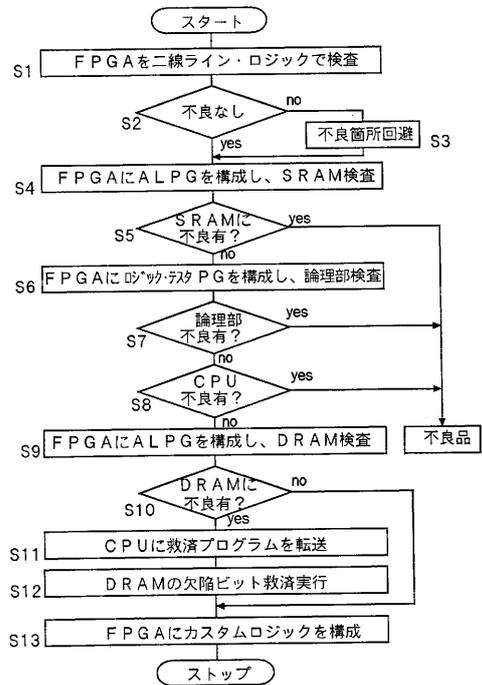
【図6】

図6



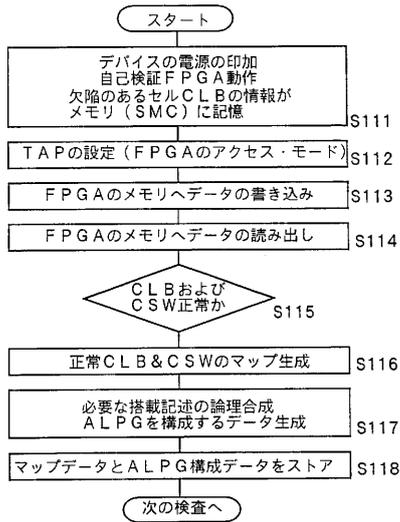
【図7】

図7



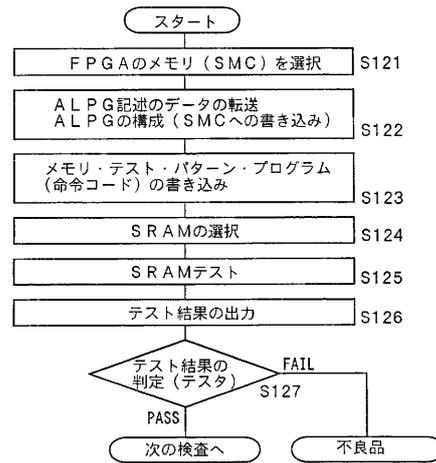
【図 8】

図 8



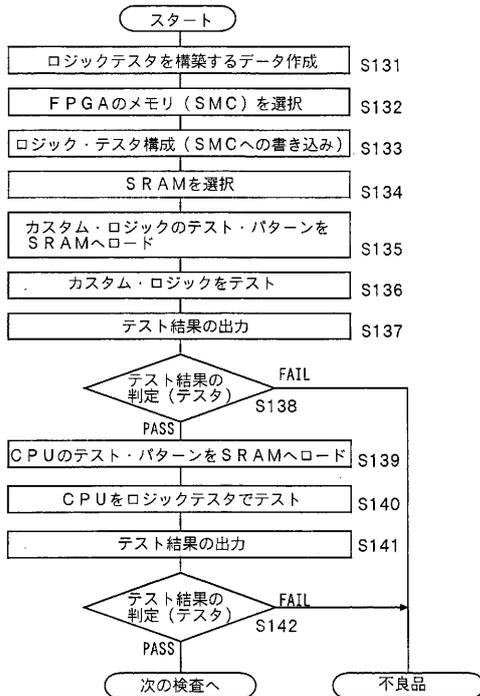
【図 9】

図 9



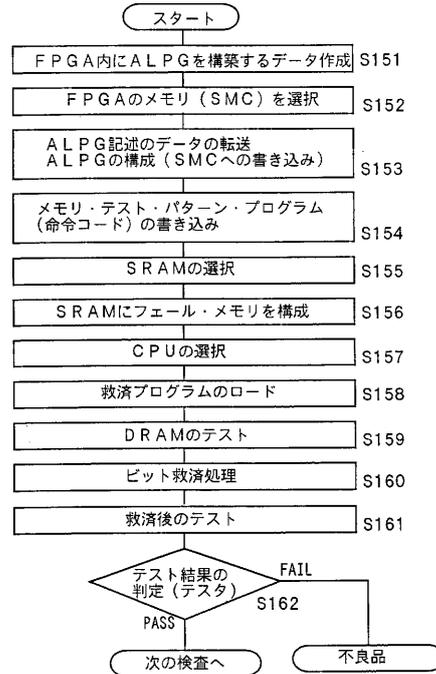
【図 10】

図 10



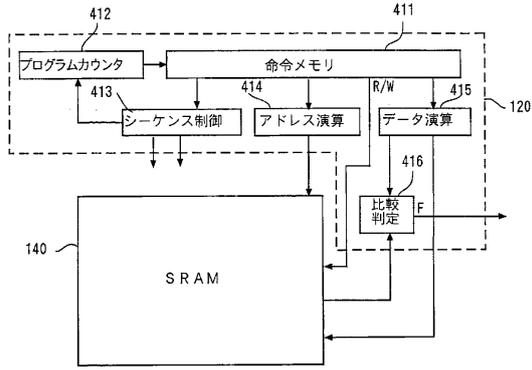
【図 11】

図 11



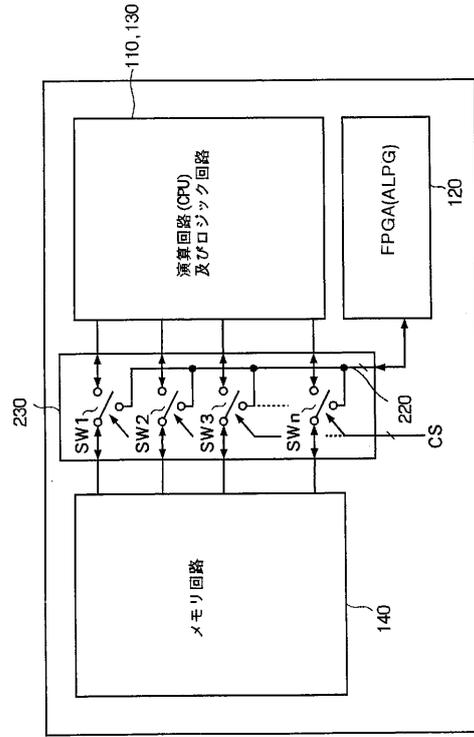
【図12】

図12



【図13】

図13



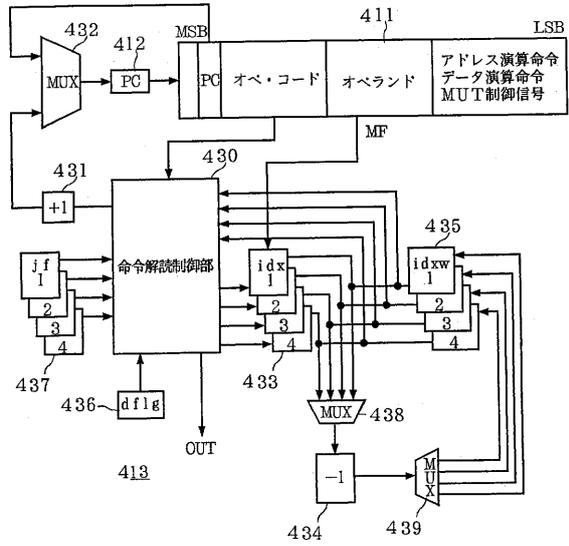
【図14】

図14



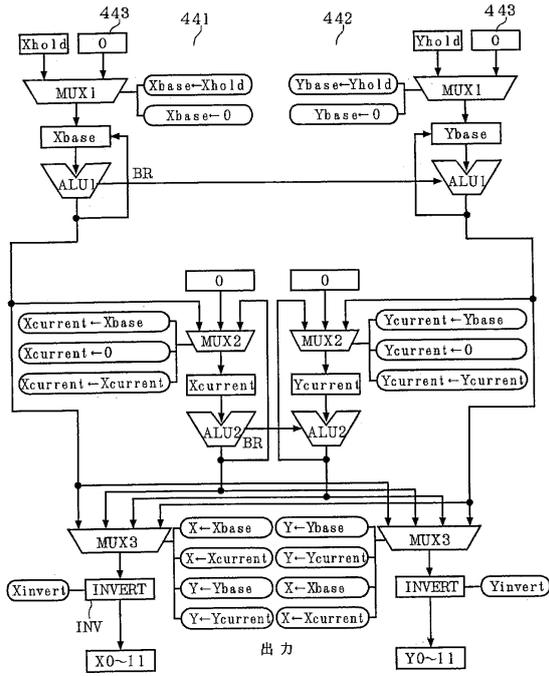
【図15】

図15



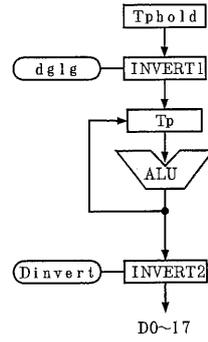
【図16】

図16



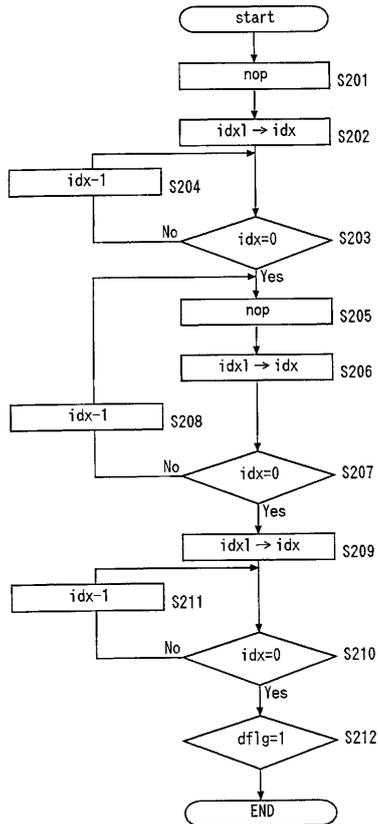
【図17】

図17



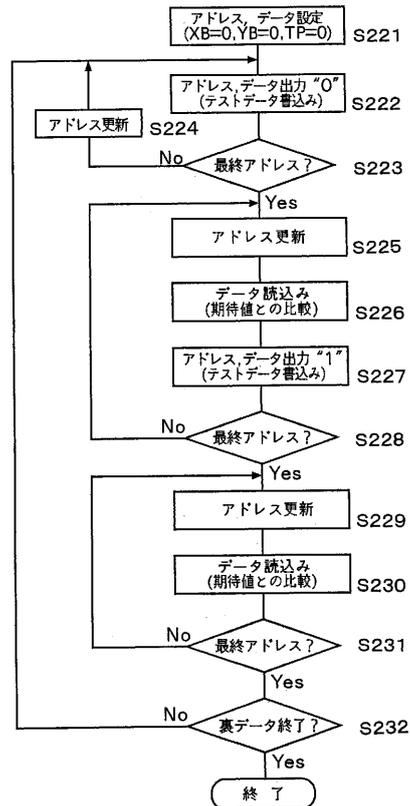
【図18】

図18



【図19】

図19



フロントページの続き

(72)発明者 高橋 秀明

日本国東京都小平市上水本町五丁目20番1号 株式会社日立製作所 半導体グループ内

審査官 宇多川 勉

- (56)参考文献 特開2000-252361(JP,A)
特開2000-124315(JP,A)
特開平10-313061(JP,A)
特開平10-302499(JP,A)
特開平10-242288(JP,A)
特開平10-241399(JP,A)
特開平10-233677(JP,A)
特開平10-223764(JP,A)
特開平10-221409(JP,A)
特開平08-298289(JP,A)
特開平08-095818(JP,A)
特開平07-099439(JP,A)
特開平06-045451(JP,A)
特開平04-368018(JP,A)
特開昭60-074643(JP,A)

(58)調査した分野(Int.Cl., DB名)

H01L 21/82

G11C 29/12

H01L 21/822

H01L 27/04