

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4317338号
(P4317338)

(45) 発行日 平成21年8月19日 (2009. 8. 19)

(24) 登録日 平成21年5月29日 (2009. 5. 29)

(51) Int. Cl.

F I

G 1 1 C 29/44 (2006. 01)

G 1 1 C 29/00 6 5 5 Z

G O 1 R 31/28 (2006. 01)

G O 1 R 31/28 B

G O 1 R 31/28 Y

請求項の数 2 (全 27 頁)

(21) 出願番号 特願2001-332489 (P2001-332489)
 (22) 出願日 平成13年10月30日 (2001. 10. 30)
 (65) 公開番号 特開2002-203399 (P2002-203399A)
 (43) 公開日 平成14年7月19日 (2002. 7. 19)
 審査請求日 平成16年10月15日 (2004. 10. 15)
 (31) 優先権主張番号 09/702631
 (32) 優先日 平成12年10月31日 (2000. 10. 31)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 506347045
 ヴェリジー (シンガポール) プライベ
 ト リミテッド
 Verigy (Singapore) P t
 e. Ltd.
 シンガポール 7 6 8 9 2 3、 ロット
 1 9 3 7 シー、 1 9 3 5 エックス、 1
 9 7 5 ピー、 ナンバー 1 イーシュン・
 アベニュー 7
 No. 1 Yishun Ave. 7,
 Lot 1937C, 1935X,
 1975P, Singapore 7 6
 8 9 2 3
 (74) 代理人 100087642
 弁理士 古谷 聡

最終頁に続く

(54) 【発明の名称】 高機能化された後デコードを有するメモリテスト

(57) 【特許請求の範囲】

【請求項 1】

被験メモリの試験結果をメモリテストで解析する方法であって、

- (a) 被験メモリに試験処理を実施して結果データを生成するステップと、
 (b) 作業位置に前記結果データを前もって記憶することなく、前記結果データにマスクを適用するステップと、
 (c) 前記結果データに前記マスクを適用して得られたデータに基づいて、次の結果データに対する新たなマスクを作成するステップ
 を含む、

前記新たなマスクは、前記結果データの全ビット位置中の 1 以上の所定のビット位置におけるビットをマスクするためのマスクデータとされ、該マスクデータは、前記所定のビット位置におけるビットエラーの発生数を該所定のビット位置毎にカウントした値と該所定のビット位置に対応してそれぞれ設定された閾値との比較に基づいて該所定のビット位置毎に作成され、該所定のビット位置毎に作成されたマスクデータの各々が、次の結果データ中の対応するビット位置におけるビットをマスクために用いられる、方法。

【請求項 2】

前記ステップ (b) におけるマスクの適用後に、前記結果データ中に残るエラー表示に従って前記ビットエラーの発生数を前記所定のビット位置毎にカウントするためのカウンタを起動するステップと、

前記所定のビット位置毎のビットエラーの発生数のカウント値を該所定のビット位置に

10

20

対応してそれぞれ設定された前記閾値と比較し、該閾値を満たしたときにエラー信号を生成するステップ

を更に含む、請求項 1 の方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

(関連特許出願について)

本願の主題は、2000年9月28日に出願された、“MEMORY TESTER HAS MEMORY SETS CONFIGURABLE FOR USE AS ERROR CATCH RAM, TAG RAM's, BUFFER MEMORIES AND STIMULUS LOG RAM”と題する米国特許出願(出願番号は不明)に開示された内容に関連する。この

10

米国特許出願には、本願が対象とするアドレス選別及びデータ選別の動作態様が開示されている。このため、この米国出願を参照により本願に組み込むものとする。
更に、上記米国出願自体も2000年9月20日に出願された、“ERROR CATCH RAM FOR MEMORY TESTER HAS MEMORY SETS CONFIGURABLE FOR SIZE AND SPEED”と題する米国特許出願第09/665,892号を組み込んだものである。この後者の出願には、以下に説明する好適な実施態様において使用されるメモリシステムが詳細に記載されており、これは本願に必ずしも不可欠のものではないが、現在推奨されるシステムである。その属性を抜きにして本願の開示を説明することは困難であり、従って米国特許出願第09/665,892号もまた、参照により本願に組み込むものとする。

【 0 0 0 2 】

20

【従来の技術】

電子装置やその機能の日常生活における浸透は極めて著しい。人々の多くは多岐にわたる雑多な目的で、家庭内でパーソナルコンピュータを使用すると共に複数の生産的ツールを持ち運んでいる。多くの個人向けの生産的電子装置は何らかの形態で不揮発性メモリを含んでいる。携帯電話は電源を切ってもユーザーがプログラムした電話番号や設定を保持出来るように不揮発性メモリを用いている。P C M C I Aカードの場合、コンピュータのスロットから取り外された状態でも情報を維持することが出来るように不揮発性メモリを使う。他の多くの一般的な電子装置も同様に、電源が供給されない組立部品において不揮発性メモリの長期保存能力を利用しているのである。

【 0 0 0 3 】

30

電子装置メーカーに不揮発性メモリを販売する不揮発性メモリの製造者は、自社で製造したメモリが適正な動作を行うかを、テスターを用いて確認しなければならない。不揮発性メモリは大量に一貫して低コストで製造、販売される為、個々の製品の試験(テスト)に要する時間は可能な限り短くすることが非常に重要である。不揮発性メモリの購入者は、購入したメモリ素子をより高額な組立品へと組み込む際、コストを節約する為に最小限度の試験のみ、或いは無試験で済むように望むことから、メモリ製造者は高い出荷歩留まりを要求される。従ってメモリの試験手順は、不適合品の大部分、望ましくは全ての不適合品を1回の試験工程において識別し得るように、十分に高い効率を提供するものでなくてはならないのである。

【 0 0 0 4 】

40

不揮発性メモリの大型化、高密度化及び複雑化が進むに伴い、テスターはこれらの寸法の増大及び複雑化への対応を、試験時間を大幅に増大させることなく行うことを要求される。メモリテスターは、連続的に動作することが多く、試験時間は、最終品のコストの大きな部分を占めるものと考えられる。また、テスターはメモリ素子の進化及び改良に伴ってメモリ素子に加えられるこれらの変化に容易に適合出来なければならない。不揮発性メモリ試験において特徴的な他の問題は、メモリのセルへの書き込みが何度も繰り返されると、その部品の全体的な寿命性能が劣化する可能性があるという点である。不揮発性メモリの製造者は、特別な試験モードをメモリ素子中へと組み込むことにより試験に関わる多くの問題に対処している。これらの試験モードはメモリの購入者によって使用されるものではなく、製造者がアクセスすることでメモリの全て、又は大部分を可能な限り短時間でか

50

つ効率良く試験することが出来るものである。不揮発性メモリの中には試験工程中に修復可能なものもある。従ってテスターは、修復の必要性、修復箇所、必要な修復の種類を識別するものでなくてはならず、また適切な修復を実施出来なければならない。このような修復工程には、メモリ中の特定の不適合部分を検出、分離することが出来るテスターが必要である。修復機能並びにこの特別な試験モードの利点を有効に活用する為には、素子から期待される応答に基づいた条件分岐をサポートする試験プログラムを実行することが出来るテスターが有益である。

【 0 0 0 5 】

メモリを試験するプロセスは、概念的観点から見るとアルゴリズム的プロセスである。例えば、一般的な試験にはメモリセルに 0 又は 1 を書き込みながらメモリアドレスを順次増分又は減分して行くことが含まれる。1 メモリサイクル中に書き込まれる、又は読み出される 1 及び 0 の集合を、一般的に「ベクター」と呼び、一連のベクターを「パターン」と呼ぶ。試験には、メモリ空間に、チェッカーボード、移動する 1 のパターン及びバタフライ型パターンのようなパターンを書き込むことが含まれるのが一般的である。試験開発者はアルゴリズム的構造の助けを借りることにより、より容易かつ効率的に試験パターン（テストパターン）を作成する為のプログラムを生成することが出来る。アルゴリズム的に一貫した試験パターンはデバッグが容易であり、期待したように実行しないパターン部分を分離する為に論理的方法を使用している。プログラミングループ中で繰り返す命令及びコマンドを使用してアルゴリズム的に作成された試験パターンは、使用するテスターのメモリ空間が小さい。従って、メモリテスターはアルゴリズム的試験パターン作成能力を有

【 0 0 0 6 】

信号エッジの正確な配置及び検出もまた、不揮発性メモリのテスターの効率に関連して考慮されるものである。中央部ではほぼ適合するが、規定されたマージンの範囲では適合しない部品を捕捉するためには、不揮発性メモリテスターは各信号エッジを他の信号エッジに対して時間的に正確に配置出来なければならない。更に、どの時点において信号エッジが受信されたかを正確に測定する能力も重要である。従って不揮発性メモリテスターは、刺激及び被験装置（DUT）であるメモリからの応答のタイミング及び配置に対する充分な柔軟性と制御性を有していなければならない。

【 0 0 0 7 】

メモリテストは、DUTへと印加（刺激）する送信ベクターを生成し、期待される返信（応答）としてのベクターを受信するものである。メモリテストは信号をDUTに接触する端子へ、及び/または、その端子からルーティングする為のマッピング機構を含んでいる為、これらのベクターを生成するアルゴリズム的ロジックは、一般的にこれらの動作をベクター中の特定のビットがどのようにDUT中の特定の信号パッドへと、あるいは、信号パッドから送られるかを考慮することなく行うことが出来る。アルゴリズムパターン生成、閾値設定、信号条件設定及び比較機構、そしてこれらをDUTへと接続するプローブの集合をテストサイト（試験サイト）と呼ぶ。単純な場合には、各テストサイトに1つのDUTが存在する。

【 0 0 0 8 】

メモリテスターは試験工程を容易にする為に使用する内部試験メモリ（内部テストメモリ）を有する。この内部試験メモリは幾つかの目的に使用することが出来、その目的の中には（送信ベクターをリアルタイムに生成するかわりに）送信ベクターの事前記憶や期待受信ベクターの記憶、そして各種エラー表示や試験中に得られたDUT挙動に関わるその他の情報の記憶等が含まれる（更にRAMを用いたメモリテスターの内部動作におけるハウスキーピング処理もあり、これは「内部メモリ」という語の範囲に入れることができる。これらはテスターの内部動作に専用のものであり、アルゴリズムレベルでは見えず、実行可能命令記憶部及び内部制御レジスタに相当するものである。このメモリを「内部制御メモリ」と呼び、DUTへの刺激、及びDUTからの応答に直接関係するビットパターンを記憶する為に用いる、本願において言うところの「内部試験メモリ」から除外するものとす

る)。この内部試験メモリは、実施する試験と少なくとも同じ速度で動作する必要があることは明白である。非常に一般的な例は、内部試験メモリ（又はその一部）をDUTへと印加するものと同じアドレス（又はそれから派生したもの）によりアドレス指定することである。この場合、内部試験メモリのアドレス指定された位置に記憶されるのは、そのアドレスにおいてDUTに実施される試験処理中のDUTの挙動を示す何等かの情報である。試験プログラム内のアルゴリズム的検討は、連続送信ベクターに関連する一連のアドレスを任意とすることができることを意味する場合がある。従って、内部メモリは高速性とランダムアドレス指定能力という2つの特性を持っていなければならない。ここで最初に思い浮かぶのは、高速であり、かつ制御が容易で、完全なランダムアドレス指定が可能なSRAMである。実際、従来のメモリテスターはその内部試験メモリとしてSRAMを使用して来た。

10

【0009】

【発明が解決しようとする課題】

残念なことにSRAMは非常に高価である為、メモリテスタに用いる内部試験メモリの容量は限られていた。この結果、メモリ不足によりメモリテスタの機能性には限界があった。DRAMはこれよりかなり安価であるが、ランダムアドレス指定ができない上に、高速動作も出来ない。

【0010】

【課題を解決するための手段】

SRAMに代えてDRAMをメモリテスタの内部試験メモリとして使用することは可能である。以下に簡単に説明するが、内部試験メモリとして使用する為にDRAMの処理速度を上げるといふ課題は、高速化を試みるかわりに、使用するDRAMの容量を増やすことにより解決することができる。複数の同一のDRAMバンクをグループとして取り扱う。1グループ中の異なるメモリバンクに対する信号をインターリーブすることと、これらのバンクグループ間の多重化とを組み合わせることにより、メモリトラフィックをいずれのバンクに対してもそのバンクが扱い得る速度まで遅くするのである。

20

【0011】

内部試験メモリ構成の最上レベルにおいては、4つのメモリセットが存在し、メモリセットの各々は独自に別個のアドレス空間を有し、要求されたメモリトランザクションを実行する。その2つは上述したSDRAMであり、他の2つはSRAMである。メモリセットの各々は独自のコントローラを有し、メモリトランザクションはそのコントローラへと向けられる。外部から見えるメモリとしての処理能力に関して言えば4つのメモリセットは全て基本的に同一のものである。これらはメモリ空間のサイズ及び内部的にどのように実現されているかについてのみ異なっている。SRAMメモリセットはそもそも十分に高速である為、多重化もインターリーブ処理も利用しない。（SRAM又はDRAMの）同じ種類のメモリセットは、それぞれに独立はしているものの、これらを「積み重ねる」、即ちより大きな1つのアドレス空間として取り扱うことが出来る。

30

【0012】

従って、テスタの内部試験メモリは4つのメモリセットへと分割されており、そのうちの2つが「内部」SRAMであり、他の2つが「外部」DRAMである。誤解の無いように触れておくが、これらのメモリは全て、物理的にメモリテスタ内部にある。「内部」及び「外部」という語は、集積レベルに関係するものである。SRAMはテスタの中心的な機能回路に関わるVLSI（超大規模集積回路）の内蔵部品であり、一方、DRAMは個別にパッケージングされ、VLSIに隣接して実装されている部品である。SRAMの容量はかなり小さい（例えば1メモリセットあたり1メガビット程度）が、DRAMの容量はかなり大きく、（例えば1メモリセットあたり128～1024メガビットの範囲で）選択可能である。SRAMメモリセットは常に存在し、ROM（読み出し専用メモリ）であるDUTの期待される内容を記憶する等、適切な目的に使用することが出来る。DRAMメモリセットは、実際にはオプションであり、一般に修復に先立つ解析用に記録を作成しておく為に使用されるが、他にも用途がある。テスタは、どちらのメモリをどの用途に利

40

50

用するかについて、S R A MとD R A Mのメモリセットとの間で原則として区別することはしない。いくつかの実際の選別は主にそのサイズにより決まる。S R A Mメモリセットは小さいが、D R A Mメモリセットは潜在的に大きい。各種メモリセットをどのように利用するかについての決定は、試験プログラムの作成者が一般に行うものである。しかしながら、メモリテストの特定の処理機能に特定のメモリセットを利用しなければならないといった若干の区別は存在する。これらは、メモリセットへの専用のハードウェア経路を必要とする経済的又は性能的配慮から通常生じるものである。これらのメカニズムを一般化することができるが、適当なものを単純に選ぶのが都合が良い。これについては、これ以上は追求しないことにする。

【 0 0 1 3 】

このような大容量の内部試験メモリ（D R A Mメモリセット）の出現を受けて次に考慮されるのは、メモリテストに望まれる機能の動作を容易にする為に、この追加メモリ容量をどのように使用することができるかである。本願で対象とするテストにおいては、内部試験メモリのサブシステムは非常に柔軟であり、32ビットという固有ワード長を有しながらも、有効ワード長を2の任意の冪数（最高 $2^5 = 32$ まで）としつつ、これに対応させてより狭幅ワード用のアドレス空間を増大させることが出来る。D U Tのアドレス指定及び内部試験メモリのアドレス指定の両方における広範囲のアドレスマッピング能力、そして複数のT a g R A M及び他のエラー解析ツールを助けるデータ選別及びアドレス選別機構があり、これらは全て内部試験メモリの容量を大きくすることで実現することが容易になる。更に、これらの改良は、より多くのメモリを空の状態にしないことによって可能となった。これらの改良は、特定の種類のメモリ部品を試験する上で非常に貴重である。

【 0 0 1 4 】

メモリテストに「後デコード」（ポストデコード、または、後復号化）と呼ばれる機能を設けることは一般的である。これは、D U Tの試験実施後、幾つかの（表やリスト、トレース又は対象事象の「イメージ」等、様々に編成された）セクションに分割された内部試験メモリが、すぐに検分することが出来るデータを含んでいることを利用するものである。この検分処理の大部分は不良表示を発見することに費やされるが、対象システムにおいては、不良表示は0で符号化されている。これは単純な例であるが、0の数を内部試験メモリのある構造中に記憶することがしばしば必要であり、これは、後デコード機構の機能としてしばしば期待されるものである。従来技術によるメモリテストにおいては、これまで、試験データを後デコード機構へと印加する前に内部試験メモリ内のあるメモリ構造中（例えばエラー捕捉R A M等）に記憶しておく必要があった。

【 0 0 1 5 】

後デコード機能を持つ従来型のメモリテストは、現在のところ試験時間の増大を引き起こすものとして認められている手段に限定されている。大容量メモリ部品のメーカーは「テストにおける時は金なり」という考えから、試験時間を短縮する手法を好む。内部試験メモリの容量が増大したこと、及び、複数のT a g R A M（高度に精錬された意味を持ち、従って基本的にD U T自体と同等のサイズであるアドレス対エラーの単純イメージ等と比較すると小さいエントリを含む表）の作成を可能にするアドレス及びデータ選別技術によって、従来の後デコード技術は、試験時間を短縮するという目的においては制約材料となってしまう。この状況はメモリ（D U T）の大型化及びその内部の複雑化が進むにつれて、単純に悪化する。D U Tの複雑性が増すということは、後デコード機構に適用される内部試験メモリのメモリ構造（T a g R A M等）が増加するということを意味し、また更にD U Tが大型化するということは、これらのメモリ構造もまた大型化することを意味する。そしてこれらの事実は試験時間を更に増大させるものである。

【 0 0 1 6 】

後デコードをより高速で効率良く行う為には何が出来るのか？また、改良された後デコード機構が実施し得る、そしてメモリテストの有用性を増す新たな測定方法が存在するのか？

【 0 0 1 7 】

後デコード処理を改良する為に複数のことが可能である。以下に説明する実施態様においては、後デコード部へと向かうデータ経路を変更して、好適なデータが内部試験メモリ中の宛先メモリ構造へと書き込まれる前又は書き込まれるときに「オンザフライ」でそのデータを後デコード機構が処理できるようにしている。それ以外のデータは相変わらず後デコード機構へと印加される前に最初に内部試験メモリ中のメモリ構造へと書き込まれる。エラーテーブル（エラー表）のコピーと共に、広範囲にわたるマスク能力により新たな試験の増分的後デコード解析が可能となり、先に実施した試験中に既に不良となっていることがわかっている位置におけるエラーの計数を回避することが出来る。異なる範囲、即ち分解能でエラーを累積することができる。ワード中のエラーとビットエラーの両方を累積することができる。改良型後デコード機構の内部アーキテクチャは、オンザフライで印加されたデータ、或いは内部試験メモリ中の構造から得たデータを一回調べることによって複数のタイプの結果を生成できることが多い。後デコード機構はその中に多数のカウンタを有している。これらのカウンタは、エラーの作用についての何らかの判断を下す為の閾値を表す、事前にロードされた値からカウントダウンを実施するものである。カウンタは、ゼロまでカウントダウンするとカウント終了フラグを生成する。種々のカウント終了フラグの値は、内部試験メモリの構造中に記憶されるべきデータ（これらの値が記録される）としていつでも使用可能である。最後に、試験の各段階が終了すると、次の段階の準備としてカウンタの各々に、その初期計数値を再ロードする場合が多い。バスを介して各種初期値を再度送信するかわりに、単一コマンドの受信によりカウンタに再ロードする初期値レジスタをそれぞれのカウンタに設ける構成とすることによって、ある程度のオーバーヘッドを節約すると共に、有用性を増すことができる。更に、あるカウンタのカウント終了フラグが存在する場合、そのカウンタの初期値レジスタからの再ロードを禁止できるようになっていることが望ましい。

【 0 0 1 8 】

【 発明の実施の形態 】

まず、図 1 を参照するが、ここには本発明の原理に基づいて構築された不揮発性メモリテストシステムの概略ブロック図 1 が描かれている。具体的には、図示のシステムは、一度に最高 3 6 個の個別の D U T を各々最高 6 4 もの試験個所（テストポイント）において同時に試験することが出来るものであり、6 4 箇所よりも多い試験個所を有する D U T を試験する為に試験リソース集合の構成要素を結合することが出来るように再構成機能を備えている。これらの試験個所は、未だ切断（またはダイシング）やパッケージングされていない集積回路ウェーハの一部分上の位置であっても、或いはパッケージングされた部品の端子であっても良い。「試験個所」という語は、信号を印加すること（例えば電源、クロック、データ入力）が出来る、又は信号を測定すること（例えばデータ出力）が出来る電氣的部位を指す。本願においては、当該産業分野における習慣に従い、試験個所を「チャネル」と呼ぶものとする。先に述べた、「結合される試験リソースの集合」という語は、3 6 個ものテストサイトを意味しており、各テストサイトはテストサイトコントローラ 4、（6 4 チャネル）D U T テスタ 6、及び、D U T 1 4 に実際の電気接続を行う（6 4 チャネル）ピン電子回路 9 の集合を含んでいる。ある D U T の試験において 6 4 以下のチャネルを要する場合は、この D U T の試験には単一のテストサイトで充分であり、この場合を、例えば「テストサイト No. 1（図 1 に示す）が『単一サイトテストステーション』を形成する、或いは『単一サイトテストステーション』として動作する」と表現する。一方、上述した再構成が何らかの形で行われる場合、2 つ（以上）のテストサイトが「結合」して 1 2 8 チャネルを有する、1 つのより大きな同等のテストサイトとして動作する。従ってこの場合を、ここでも図 1 を例にとると、「テストサイト No. 3 5 及び No. 3 6 が『2 サイトテストステーション』を形成する」と表現する。

【 0 0 1 9 】

次に逆の場合を簡単に考察するが、留意すべきは、1 つのテストサイト全体が 1 つの D U T を試験するために必要とされる、或いは 1 つのテストサイトは、1 つの D U T しか試験できない、ということではない点である。ウェーハが 2 つ、3 つ、または 4 つのダイ（恐

10

20

30

40

50

らくは隣接しているものであるが、必ずしもそうでなくとも良い)を含み、その試験チャネル要件の和が64以下であると仮定する。このようなDUT(15a~d)は、単一のテストサイト(例えば、図2に示すようなテストサイトNo.2)で同時に試験することが出来る。これが可能なのは、後で説明する特定のハードウェア機能によって強化された、各テストサイトの汎用的なプログラム機能による。原則として、そのテストサイトが実行する試験プログラムを、そのテストサイトのリソースの一部が一方のDUTの試験に使用されることになるように、そして他の部分が他方のDUTの試験に使用されることになるように書くことが出来る。要するに、先に述べた2つのDUTの論理結合である第三のDUTがあるとして、この第三のDUTを単一のテストサイトで試験することが出来れば、その「構成要素であるDUT」をそのまま同様に試験することが出来るはずである。主な違いは、「第三」のDUTに対する単純な統合的な回答が出るのではなく、2つの「構成要素DUT」が合格したか不合格となったかの記録が個別に取られるということである。即ち、「第三」のDUTのどの部分が不合格となったかの問題となるのである。更に、不良DUTへの駆動信号の除去又は制限や、試験プログラムをどのDUTが不良を示したかに基づいて分岐させつつ、同時に、テストプログラムが非現実的な程度にまで多重化することがないようにすること等、他の問題も存在する。単一のテストサイトにおけるこの「マルチDUTテストステーション」能力の特定の態様はごく単純であるが、他のものは複雑である。マルチDUTテストを、2つ以上のテストサイトを結合するという概念と混同してはならない。

【0020】

このテストサイト再構成の概念抜きにしては、テストサイトとテストステーションの間には何等の相違も無いことになり、いずれか一方の語を使用すれば良いことになる。しかしながら明らかなように、テストステーションの数は必ずしもテストサイトの数と同じではない。過去においても、単純なマルチDUTテストを行うために、(DUTが1つのテストサイトを使い切るほど複雑ではない場合)テストサイトを分割してより多数のテストステーションを作成することで双方の数が異なることとなる場合があった。しかしながら、ここでの数の違いは、(DUTが単一のテストサイトで試験するには複雑過ぎる場合)テストサイトを結合してマルチサイトテストステーションを形成することによっても生じる。

【0021】

説明を進める。テストシステム(試験システム)コントローラ2は、システムバス3を介して最高36個のテストサイトコントローラ(No.1~No.36)4a~4z(符号a~zは1~26文字迄であり、36個に対応しないが、数字符号の後にまぎらわしい数字を使用するより好ましいと考えた)に接続している。テストシステムコントローラ2は、不揮発性メモリの試験タスクに関わる好適なテストシステム制御プログラムを実行するコンピュータ(例えばNTを実行するPC)である。テストシステム制御プログラムは、所望の試験を行う為の仕事(及び複雑性)の階層において最高レベルにある抽象概念を表すものである。テストシステムコントローラはそれぞれのテストサイトによってどのプログラムが実行されているかを判定すると共に、必要に応じて試験プローブ及びDUTを動かすロボットシステム(図示せず)を監督する。テストシステムコントローラ2は、「いくつかのテストサイトが、マルチDUTテストステーションのような単一サイトテストステーションとして動作するようにプログラムされ、他のテストサイトを結合してマルチサイトテストステーションを形成するようにプログラムされる」という概念を支持するように機能することが出来る。このような状況においては、異なる部品が試験されていることは明らかであるが、最も好ましくは、異なる部品に異なる試験が用いられることである。同様に、全ての単一サイトテストステーションが同じ様式の部品を試験しなければならないという条件は無く、マルチサイトテストステーションにもそのような条件は無い。従って、テストシステムコントローラ2は、必要とされるテストサイト結合を実施するコマンドを発し、その後、使用される様々なテストステーションに適正な試験プログラムを実行させるようにプログラムされているのである。テストシステムコントローラ2は更に、試

験により得られた結果に関する情報の受信も行う為、不良品を破棄する為の適正な行動を取ることが出来、そして例えば工場のセッティングにおける製造工程を制御するために使用可能な様々な解析用のログを維持することが出来る。

【 0 0 2 2 】

テストシステム自体は相応に大型の複雑なシステムであり、一般的に、ロボットサブシステムを使用してウェーハをステージ上に搭載し、続いてピン電子回路 9 に接続するプローブの下に 1 つ以上の将来のダイを配置するようになっている。そして、それらの将来のダイ（この時点ではまだウェーハはダイシングされていない）がテストされる。テストシステムはまた、好適なキャリア上に搭載されたパッケージ済み部品の試験にも使用することが出来る。（後述するように）テストステーションを構成するためにいくつかのテストサイトが用いられているか、或いは 1 テストサイトにいくつかのテストステーションがあるかにかかわらず、使用されるテストステーションの各々に関連付けられた少なくとも 1 つのテストサイトコントローラが存在する。テストサイトコントローラは、例えば、36 ~ 64 MB のプログラム及びデータメモリの組み合わせを含み、V O S (V e r s a T e s t O / S。これはアジレント社の V 1 3 0 0 又は V 3 3 0 0 等の不揮発性メモリを試験する為の初期の製品にも使用された）と呼ばれる自社開発 O S を実行するインテル社の i 9 6 0 プロセッサ等の埋込みシステムである。ここではしばらくの間、単一サイトテストステーションについてのみ考える。具体的な事例として、テストサイト N o . 1 がテストステーション N o . 1 として機能して、W H I Z C O 部品番号 0 0 1 3 を試験するものとする。試験には 1 0 0 種類程度の異なるタイプの試験（電圧レベル、パルス幅、エッジ位置、及び遅延を変化させて監視すること、及び選択された情報のパターンを何度も単純に記憶して取り出すこと）が含まれ、各タイプの試験には D U T に対して何百万ものメモリサイクルが必要である。最も高位のレベルにおいて、テストシステムのオペレータがテストシステムコントローラ 2 に対してテストステーション N o . 1 を使って W H I Z C O 0 0 1 3 の試験を開始するように命令する。テストシステムコントローラ 2 は、その処理の過程でテストサイトコントローラ N o . 1 (4 a) (埋込み [コンピュータ] システム) に対して関連する試験プログラム（例えば T E S T _ W H I Z _ 1 3 ）を実行するように指示する。このプログラムがテストサイトコントローラ N o . 1 の環境中に既にある場合は、単純にこのプログラムが実行される。ない場合は、プログラムはテストシステムコントローラ 2 によって供給されることになる。

【 0 0 2 3 】

ここで基本的には、プログラム T E S T _ W H I Z _ 1 3 は完全に独立したものであっても良い。しかしその場合、ほぼ間違いなくそのプログラムは大きくなるため、テストサイトコントローラ 4 a の埋込みシステムのプロセッサがこれを十分に高速で動作させて、試験を所望の速度で生成すること、或いは D U T メモリサイクル毎に均一な速度で生成することさえ難しくなる。従って、書き込むべき、或いは読み出し処理で期待されるアドレス及び関連データのシーケンスを生成する下位レベルのサブルーチンタイプの動作は、必要に応じて D U T テスター 6 中にあるプログラム可能なアルゴリズム的機構により実行されるが、これはテストサイトコントローラ 4 中の埋め込みシステムが実行するプログラムと同期して動作する。このことを、ある低レベルサブルーチンと同様の動作と D U T メモリサイクルを起動するタスクとを D U T 1 4 のハードウェア環境により近い機構（D U T テスター）にエクスポートすることであると考える。すると、一般的に、テストシステムコントローラ 2 がテストサイトコントローラに試験プログラムを供給する度に、テストシステムコントローラ 2 は、また、テストサイトコントローラに対するプログラミングにより記述される、又は要求される、全体的な動作の実行に要する適正な低レベル実行ルーチン（恐らくは試験するメモリに固有である）を、関連する D U T テスターに供給する。低レベル実行ルーチンは「パターン」と呼ばれ、高レベルのプログラミング言語における機能や変数に名前があるのと同様に、これらにも一般的に名称が付けられている。

【 0 0 2 4 】

各テストサイトコントローラ N o . n (4) はサイトテストバス N o . n (5) により関

10

20

30

40

50

連するDUTテストターNo. n (6) に結合されている。テストサイトコントローラは、サイトテストバス5を使用してDUTテストターの動作を制御し、そしてそこから試験結果に関する情報を得る。DUTテストター6は試験に関わる様々なDUTメモリサイクルを高速で生成することが出来、また、Read (読み出し) メモリサイクルの結果が期待通りのものであるかどうかを判定する。基本的に、DUTテストターは対応する有用なRead及びWrite (書き込み) DUTメモリサイクルのシーケンスを起動することによりテストサイトコントローラから送られるコマンド又は処理コード (「名前の付けられたパターン」) に応答する (すなわち、対応するパターンを実行する) 。概念的にDUTテストター6はDUTに印加されるべき刺激情報を出力し、また、そこから返される応答情報を受信する。この刺激 / 応答情報7aはDUTテストター6aとピン電子回路No. 1アセンブリ9aとの間でやり取りされる。ピン電子アセンブリ9aは最高64本までのプローブをDUT14へと接続することが出来るようにサポートする。

10

【0025】

上述した刺激情報は、DUTテストター中に使用されているいくつかの論理素子系列の電圧レベルに基づいて表現された、単なる並列ビットパターンのシーケンス (すなわち、 「送信ベクター」 及び期待される 「受信ベクター」 のシーケンス) である。刺激 / 応答中のビット位置とダイ上のプローブとの間には構成可能なマッピングが存在し、このマッピングはDUTテストター6が把握している。個々のビットはそれらのタイミング及びエッジ配置に関して適正なものであるが、マッピングに加え、それらのビットをDUTに印加できるようになる前に電圧レベルシフトが必要な場合もある。同様に、刺激に続いてDUTから出力される応答も、それをDUTテストターに送り返すのに適した状態になったとみなすことができるようになる前にバッファリング及び (逆) レベルシフトが必要な場合もある。これらのレベルシフトタスクはピン電子回路9aの守備範囲にある。WHIZCO 0013の試験に必要なピン電子回路の構成は、ACME社の部品を試験するためには (WHIZ社の他の部品を試験するためにでさえ) 、おそらく適用できないであろう。従って、ピン電子回路アセンブリもまた構成可能であることが要求されることがわかる。このような構成能力の提供はPE構成ライン8aの機能である。

20

【0026】

以上、1つのDUTを試験する為に1つのテストサイトがどのように構築されるかについて構造的な概要を述べた。次に、多数のテストサイトがある場合にどれを動作させるかに関連して生じる問題について触れる。準備として、複数のテストサイトを有するテストシステムを構築する為の好適な実施態様を説明する。これからここに記載するいくつかの情報は、多くの点において顧客嗜好についての市場調査及び費用便益分析に基づく選択の問題である。ともかく、これらの事項のいずれか1つを確立するには、明確な選択をしなければならず、一度この選択を行うと、システム全体を通じて特定の結果が明らかとなる。ここではテストシステムのハードウェア特性のより広範囲な概要を少なくとも一般的な方法で説明することが有用であると思われる。これらの特性のいくつかは、条件により変化するものではあるが、それでもそれらを知っておくことは本発明を説明するために提示する様々な例を理解する上での一助となる。

30

【0027】

まず始めに、4個の比較的大型のカードケージを考える。各カードケージは、電源及び水冷機構 (クリーンルーム環境下においてファンは汚染源となる可能性がある。また、冷水は、負荷が満載されたシステムから数十KWの熱を放熱させるのには、空調を使用するよりも安価である) の他に、マザーボード、前面及び後面を有する。各カードケージには最高9個のアセンブリを挿入することが出来る。各アセンブリはテストサイトコントローラ、DUTテストター及びピン電子回路を備える。テストサイトコントローラをどのように結合するかについての一般的な概略を説明するが、これには幾つかのバスを使ってディジーチェーン接続を作ることが含まれる。

40

【0028】

本題から若干それるが、最初にこの 「ディジーチェーン接続」 について簡単に触れる。シ

50

システム構成要素 A、B、C 及び D があるものとする。これらがこの順番でディジーチェーン接続されたと想定する。すると、そこには A を出て B に向かう情報又は制御経路があり、B は B を出て C に向かうトラフィックを選択的に送出することが出来、C は D に向かうトラフィックを選択的に送出することが出来る。同様の構成が他方向のトラフィックにも存在し得る。ディジーチェーン接続は優先順位機構を作る為に良く採用されているもので、ここでは種々のテストサイトコントローラの間にマスター・スレーブ関係を作る為に用いる。これらのディジーチェーン方式による通信構成を、末尾に「BUS」ではなく「DSY」を付すことにより示している。従って、例えば「コマンド/データ BUS」ではなく、「コマンド/データ DSY」としている。次に、情報が「B に入り、選択的に送出される」という概念は、そのトラフィックが、送出される前に、別個の導体セット上に複製されることを示唆している場合もある。しかしながら、性能上の理由から、これはどちらかと言えばアドレス指定可能なエンティティを有する通常のバスである。プログラム可能なアドレスマッピング構成及び下流のテストサイトコントローラの一部を「スリープ」状態にする能力により、1つのバスを論理的には複数のディジーチェーンであるかのように見せる（機能させる）ことが出来る。最後に、ディジーチェーンはコマンド及び制御情報の経路として高い性能を持つものであり、もし、そうでなかったらマスター・スレーブの組み合わせ（マルチサイトテストステーション）が1つのテストサイトと同じ速度で処理を行うことは望めないことが理解されよう。ディジーチェーン性能の恩恵を受けるために、各種 DSY はそれぞれのカードケージを出ない。これを実施する為に、どの（従って幾つの）テストサイトを結合できるかについて幾つかの制限が設けられる。基本的にはこの制限を設ける根本的な必要性があるわけではなく、技術的実用性が本質的に欠如しているわけでもない（即ち実施可能である）。これは単純に、既に9個のテストサイトがカードケージにあるのだから、DSYを延長することは多大なコストを払ってわずかな追加利益を得ることにしかないとの考えからである。

【0029】

図1の説明に戻り、各種テストサイトコントローラ 4a ~ 4z を考察するが、これらは4個のカードケージ内にそれぞれ9個ずつ収容されている。これらを 4a ~ 4f、4g ~ 4m、4n ~ 4t、4u ~ 4z とする（先にも述べたように、アルファベットは本来26文字しかないが、もう10文字がその中のどこかに存在するものと想定する）。CMD/DAT DSY 17a（コマンド/データディジーチェーン）は1つのカードケージ中にあるテストサイトコントローラ 4a ~ 4f を相互接続し、別の CMD/DAT DSY 17b は他のカードケージ中にあるテストサイトコントローラ 4g ~ 4m を相互接続する。残りのカードケージとテストサイトコントローラ 4n ~ 4t 及び 4u ~ 4z についても同様の構成がそれぞれ存在する。先にも述べたように「DSY がカードケージを出ない」と言った場合、その DSY を実際に構成するバスの最終端がカードケージを出ずに、他のカードケージにおける次のセグメントのヘッド（先端部）になる、ということである。テストシステムコントローラ 2 からのシステムバス 3 が、全てのテストサイトコントローラに向かい、各々が、カードケージを出ない DSY セグメントのヘッドにおいてマスターになることが出来る。

【0030】

ここまで説明して来た CMD/DAT DSY 17a ~ 17d はそれぞれのテストサイトコントローラ 4a ~ 4z の間に存在する。SYNC/ERR DSY 18a ~ 18d 及び DUT テスター 6a ~ 6z も同様の構成である。SYNC/ERR DSY 18 によって運ばれる同期及びエラー情報により、DUT テスターは調和して動作することが出来る。これら2つのディジーチェーン（17 及び 18）は若干異なる種類の情報を搬送するものではあるが、各々は1つ以上のテストサイトを結合してテストステーションを構成する為の同じ全体機構の一部として存在している。

【0031】

図2は、図1のDUTテスター6の概略を示す拡大したブロック図である。36個ものテスターがあるが、ここではそのうちの1つを説明する。図2を一目見れば明らかなように

、概略ブロック図にしては相当な数の構成要素が描かれている。DUTテスター6に含まれるもので、このブロック図に描かれた要素の中には、機能的に非常に複雑であり、市販品としては入手できないものもある。ここで2つの点について明記しておく。第一に、図2の主要目的は、不揮発性メモリテストシステム1全体の重要な動作環境の基本特性を説明することである。図3以降の図に関連して詳細に説明する本発明は、図2について以下に説明する機構の拡張であるか、或いはその存在根拠を図2に見出すことが出来る新たな機構である。いずれにしても、この説明を書いている段階では読者がこれらの機構のどれに直面するかは明確にはわからない。現時点での目標は様々な好適な実施態様の多数の異なる詳細な説明の開始点として簡潔でありながら十分な情報を提供することにより、これらの実施態様の各々を簡明かつ適正に説明することである（異なる発明の各々について全てを開示する「長大な」明細書にならないように）。第二に、発展させた、すなわち拡張した実施態様の中には、図2に示す構成に大体においては一致するものの、その簡略化されたバージョンとは正確には「整合」しない情報を含むものもある。しかしこれは誤りや決定的な矛盾ではなく、全体像を完全な縮小版で表すように物事を簡略化して示すことが時に困難、或いは不可能な場合があるからである。この状況は地図と似ている。例えばコロラド州の標準サイズの道路地図では、州間高速自動車道I-70に乗って東へ向かえば、デンバーでI-25に入って北へ向かうことが出来ることがわかる。これは左折のように見える。実際、昔は左折であったが、現在は違う。このインターを詳しく記した地図を見ると、一連の部分的進路変更と割り込み車線が示されている。しかしながら標準サイズの道路地図が間違っていると言う者はいない。このレベルの抽象性においては正しいからである。同様に、図2は相当詳しく描かれているようではあるが、実際は中程度の抽象性で描かれた簡略図であり、一見「左折」に見える部分でも、実際は単純な「左折」とは言えない場合もある。

【0032】

図1に示したように、DUTテスタ6への主な入力は、対象とするDUTテスタ6のインスタンスに関連するテストサイトコントローラ4を起点としたテストサイトバス5のインスタンスである。テストサイトバス5は、テストサイトバス上のトラフィックをリングバス85又はVTバス89のトラフィックへと変換するマルチバスコントローラ88に結合している。更にリングバストラフィックをVTバストラフィックへと、そしてVTバストラフィックをリングバストラフィックへと変換することも出来る。図2に示したほぼ全ては、大規模集積回路の一部である。便宜上、1つのエンティティとして示してはいるが、タイミング/フォーマッティング及び比較回路52（以下に説明）は実際に8個のそのようなICである。各種外部DRAM（これらの一部も内部試験メモリ87の一部である。図3参照）を除いては、図2に示した構成要素の殆どは、APG（自動パターン発生器）と呼ばれる他の大型ICの一部である。リングバス85は、DUTテスタ6のAPG部における主要構成要素の設定や動作モードの設定等を行う為の汎用の機構間通信経路である。APGの各構成要素間には、様々な専用の広く高速な経路も存在する。VTバス89はDUTテスタ自体内で用いられる、IC間のバスである。

【0033】

リングバス85は、テストサイトコントローラがDUTテスタ6のAPG部と通信を行う為に使用する機構である。リングバス85は、マイクロコントローラシーケンサ19に結合しているが、このシーケンサは特殊用途向けマイクロプロセッサにたえることができる。これは次アドレス計算機102が作成するアドレスを用いてプログラムメモリ中に記憶されたプログラムから命令をフェッチするものであるが、このプログラムメモリはマイクロコントローラシーケンサ19内部のもの（プログラムSRAM20）でも、外部のもの（外部DRAM21）でも良い。これらの2つのメモリは基本的に、論理的に共通の、プログラムカウンタ（又は命令フェッチアドレス）として作用するアドレス63によってアドレス指定されているかのように見え、また、いずれも実行すべきプログラムのソースになりうるものではあるが、（1）どの期間においてもいずれか一方のメモリしか命令フェッチメモリサイクルを実行せず、（2）実際、これらは電氣的に異なる信号によってア

ドレス指定される、という点に留意すべきである。SRAMは高速であり、真のランダムアクセスが可能であるが、マイクロコントローラシーケンサ19（大型APG ICの一部）中の貴重な空間を使用する為、そのサイズは限られている。外部DRAMはかなりの容量に設定することが出来るが、線形的に実行しかつ分岐せずに連続した領域をアクセスするときのみ高速に動作することができる。SRAM20におけるプログラミングはアルゴリズム性の高いものが多いが、外部DRAM21は初期化ルーチンやランダム又は不規則データのようなアルゴリズム処理では容易に生成出来ないものに最も適している。

【0034】

次アドレス計算機102は、無条件ジャンプ命令又は条件ジャンプ命令、或いは各種プログラム制御フラグ25、その他のフラグ55及びマルチDUT処理用に供給される他の特定の信号（便宜上、DFE0:3 103及びDPE0:3 104として別途示した）によって条件付けられた条件サブルーチン命令に応答して、実行中の試験プログラムに分岐を生じさせることが出来る。

【0035】

マイクロコントローラシーケンサ19がフェッチ及び実行する命令語は208ビットと、かなり長い。これは13個の16ビットフィールドから構成される。これらのフィールドはマイクロコントローラシーケンサ19本体の外部にある機構用にフェッチした命令情報を表すことが多い。このようなフィールドは、それらの関連する機構専用のものである。1セットのALU命令22が8個の16ビットALU24の集合へと印加され、他のセットはDUTテスト内に分散する他の様々な機構に分配される。後者の状態は、「各種制御値及び命令」42と表示したライン及び文字で表している。

【0036】

8個の16ビットALU24は、各々それに関連する16ビット結果レジスタ（各ALUは他のレジスタも幾つか含んでいる）の周囲に構築された算術命令の従来のレパトリを有する。結果レジスタの内3つと、それらが関連するALUは、X、Y及びZのアドレス成分27を生成する為のものであり、アドレス成分は様々に組み合わせられてDUTへと供給される完全なアドレスになる。8個のALU/レジスタ（DH及びDL）のうちの2つ以上は、32ビットデータパターン28をアルゴリズム的に作成するのを助けるものであり、32ビットデータパターン28は最大有効部分（DH）と最小有効部分（DL）に分割される。最後の3つのALU/レジスタ（A、B、C）はカウンタとして使用され、様々なプログラム制御フラグ25の生成に寄与する。これらのプログラム制御フラグは、プログラムで指定された繰り返し回数又は他の数値的条件が完了した時点で実施されるプログラム制御及び分岐を補助するものである。これらのプログラム制御フラグ25はマイクロコントローラシーケンサ19へと返送され、マイクロプログラムされた実行機構を熟知する者には周知の方法で命令フェッチアドレスの値に作用することになる。更に各種の「他のフラグ」55も存在し、これらもプログラム分岐を行う為に用いることが出来る。これらはフェッチした命令語のそれぞれのフィールドによって制御されるDUTテスト6中の他の種々の機構から発生する。1つの特定の追加フラグとしてPD_ERR90を個別に明示的に示した。これは後デコード機構60を起点としてプログラムSRAM20へと送られるもので、後デコード機構60がエラーを発見したことを示すものである。このような追加フラグの他の例としては、VEC_FIFO_FULL26がある。より簡略化した他の図においては、これもその他のフラグ55に包含されている。これをここで個別に示したのは、マイクロコントローラシーケンサ19の動作の一側面の説明を助ける為である。

【0037】

VEC_FIFO_FULLの機能は、マイクロコントローラシーケンサ19によるプログラムの更なる実行を（一時的に）停止させることである。マイクロコントローラシーケンサ19がフェッチした命令と、DUTに印加すべき試験ベクターを最終的に送出する機構との間には多段のパイプラインが存在する。更に、DUTへの印加に向けて搬送されるときにベクターを伴うバゲージの一部は、最終的なベクター印加速度、或いは各ベクターの

10

20

30

40

50

持続時間に関する情報である。従って、DUTへのベクター印加速度は一定である必要は無く、具体的には、あるベクター群の印加時間は、生成にかかった時間よりも長くても良い。マイクロコントローラシーケンサは、単純にプログラムをその最高速度で実行する。しかしながら、「ベクター消費」速度が「ベクター生成」速度と平均で等しくなければ、パイプラインにはほぼ無限の柔軟性を持たせなくてはならなくなることは明白である。以下に説明するアドレスマップ29の出力にはベクターFIFO45があり、これがパイプライン中の柔軟な容量として作用する。信号VEC_FIFOFULLは、パイプラインの先頭部(ヘッド)における新たなベクターの生成を一時的に中断することにより、パイプライン中の限界段数を超過しないようにするために使用される。

【0038】

更に説明を続けると、(3×16=48ビットの)X、Y及びZアドレス成分27がアドレスマップ29に印加される。アドレスマップ29の出力は、順序付けられた48ビットアドレス空間において、事前に選択されほぼ任意に再構成されたアドレス値である。この理解を助ける為に本題から離れるが、このアドレスマップ29を、48ビットアドレス空間を完全に収容するメモリとし、各アドレスには48ビット値が保持されていると想定する(このようなメモリを現在実現しようとするれば大型冷蔵庫大となるであろうが、このことは一時的に無視して考える)。このようなメモリがあれば、どのようなアドレスが与えられても、他の任意に選択された、後に置き換えアドレスとして使用できる48ビット値へとマッピングすることが可能なルックアップテーブルを実現することができる。このようなアドレスマッピングが望まれる理由は、X、Y及びZアドレス成分が一般的に、特定のDUT内部構造のコンテキストにおいて、1つの大型リニアデコーダでは実現されにくい有用な意味を持つ為である。行、列、層、ブロック或いはページといった概念はテストエンジニアにとっては非常に便利なものであり、物理的に近い位置で生じた不具合は、それらのX、Y及びZアドレスにおいてこれに対応する近接性を持っている。試験結果におけるこのようなパターンは、何が悪いのかを評価し、欠陥部分の動作を予備部分の動作に分岐するために、一部分を、設計レベル、又は製造レベルで再プログラミングして修復する上で有用なものである。このような考え方からは、2つの問題が生じる。第一の問題は、48ビットを削減してDUTへと印加すべき実際のビット数(例えば32又は16)にすることである。この削減がどのように行われるかを簡単に説明すれば、これはXから何個かのビットを得、Yから何個かのビットを得、そしてZから残りのビットを得ということである。しかし全てではなく、いくつかのアドレスは、回路の他の部分の左右(又は左右及び上下)の鏡象である回路中に存在する可能性がある為、これが第二の問題となる。これには、連続するアドレス値が回路内において物理的な順序で並んでいる限り、ビットの意味が変わるという作用がある。このチップレイアウト特性は何回も生じ得るもので、ある1群のビット(例えばY)をどのように解釈するかは、それに付随する他(例えばZ)のビット値に依存する場合がある。アドレスマップ29はX、Y及びZの生アドレスを「再パッケージング」することでこの種の状況を反映できるようにするために設けられたものであり、このような内部構造を有するメモリを試験する人々の便宜をはかるものである。実際にこれがどのように行われるかを説明する。アドレスマップ29は、相互接続されたかなり多くの数のマルチプレクサ(MUX)から構成される。これは、説明の便宜上先に一時的に想定した、完全に収容されたメモリデコード方式における完全に任意のルックアップテーブル機能を実施することは出来ない。しかしながら、具体的には48ビットから実際に必要な数に削減する他の機構がある為、X、Y及びZアドレス成分のサブフィールドを必要に応じて並べ替えることが出来る。アドレスマップ29は、制限された任意のマッピングを局所的な範囲において実施することが可能な、3個の16ビット(アドレス)ルックアップテーブルを更に有する。

【0039】

アドレスマップ29のマッピングされたアドレス出力30は、アドレスとして各種バッファメモリ及び/又はTagRAM31A-B、そしてエラー捕捉RAM1/2(32A/B)へと印加される。これらのメモリは別個の機能を有してはいるが、集合的に内部試験

10

20

30

40

50

メモリ 87 を構成する 4 つのメモリセット中の選択可能な区画として実現することが出来る。マッピングされたアドレス出力 30 は、また、アドレスビット選択回路 37 (その多重化機能は後に説明する) への 1 入力としても印加される。内部試験メモリは、異なる機能に使用される様々な RAM ベースのメモリ構造を多数含むように構成することが出来る。これは異なるメモリセットの特定の部分を、関連する目的に使用することを宣言することにより実現される。図 2 に示されているのがこのような構成の 1 つであり、試験の進行に伴って構成を変えることが出来、このメモリセット用途に関わる処理全体は非常に動的であると考えられる。内部試験メモリ中に存在するいずれの部分 (例えばエラー捕捉 RAM 32A - B) も、恒久的に固定されたハードウェアではない。不変なのは 4 つのメモリセットだけである。しかしながら、いずれの時点においても、(もし実際に定義されるとすれば) どのメモリセットのどの部分がエラー捕捉 RAM であるかに関しては、現在確立されている構成に依存する。

【0040】

バッファメモリ 31A 及び 31B について説明する。それらの機能は、DUT に印加することが可能なデータパターン 33 及びアドレス 34 を保持することである。これらは、実際には、それらの関連するバッファメモリからの別個の出力である。バッファメモリは「デュアルポートメモリ」ではないが、2 つの異なるメモリセットの部分から構成されるのが望ましい。この場合、記憶済みデータ 33 が一方のメモリセットに保持され、記憶済みアドレス 34 が他方のメモリセットに保持されるようになっていることが望ましい。更に、バッファメモリへの書き込みを行う機構を明示していないが、これは 1 つには、実行されているプログラムの命令でテストサイトコントローラ 4 が起動するアドレス指定されたバスの動作により行うことができる。(図が非常に見にくくなる為に大部分を省いたが) リングバス 85 と呼ばれる「床下」の「ユティリティサービス」バスがあり、これが図 2 に示した要素のほぼ全てへと通じている。メモリセットへと情報を書き込む為の他のより高速な方法は、図 3 に関連して説明する。

【0041】

エラー捕捉 RAM 32A - B はバッファメモリ 31 に印加されたのと同じアドレスによりアドレス指定される。これらは、エラーに関する情報の記憶及び取り出しを行うもので、その動作は後述する後デコード回路と連係して実施される。バッファメモリ 31A - B からの経路 33、34 と同様に、エラー捕捉 RAM 132A からの経路 62A ~ D は、リングバス (図示せず) によって配信される構成情報に基づいたメモリセット部分 (エラー捕捉 RAM として動作するよう構成されている部分) からの多重化出力であることが望ましい。

【0042】

留意すべきは、データマルチプレクサ 35 はその入力として ALU 集合 24 中のレジスタ DH 及び DL からのデータ 28 を受けると共にバッファメモリ 31 からの記憶済みデータ出力 33 をも受けるという点である。データマルチプレクサ 35 はこれらの入力 (28、32) のうち、どちらをその出力 38 として提供するかについて、プログラム SRAM 20 中に記憶された値 36 に基づいて最初の選択を行う。その出力 38 は、以下に説明するような変更が加えられない限りは、2 つのベクター構成要素のいずれか一方として (もう一方の構成要素はアドレスビット選択回路 37 の出力 39 である) 送信ベクターマッパ/シリアライザ/受信ベクター比較データ回路 40 へと印加される。

【0043】

回路 40 は 3 つのベクター関連機能を実行することが出来る。すなわち、ベクター構成要素 (38、39) を組み合わせて DUT に印加 (伝送) するベクター全体を順序付けられた論理表現にすること、送信ベクターの論理表現の順序付けられたビットと、信号 (ベクター中のそのビット) を届ける為に DUT に接触するピン電子回路 (例えばプローブ先端等) の実際の物理チャネル番号との間に動的な任意の対応付け (マッピング) を実施すること、そしてそれを受け入れる DUT に対し、別個に、順に (連続的に) 印加されるように全体の論理ベクターを部分に分割する際にコンパイラと連係して作用することである。

これらの機能のうち、どれを実行するかは、SRAM 41からの制御信号により決定されるが、この制御信号もまた、マイクロコントローラシーケンサ19によりフェッチされた208ビット命令中の1フィールドに従ってアドレス指定される。

【0044】

更に回路40に含まれているのは、DUTディスエーブル(DUT禁止)論理部90である。これが存在する理由は、最高4つのDUTのうち、1つ以上のどのDUTを無効(ディスエーブル)にするのかを表示する、ある程度静的で、ある程度試験結果により変わるが、全てプログラマ的に規定される様々な状況に対応する為である。これらの表示は、4つの信号DD0:3 44b(DUT0、DUT1等に対するDUTディスエーブル)により伝達される。これは1つのテストサイトでのマルチDUTテストをサポートするものであり、この詳細は組み込んだ関連特許出願に記載されている。回路40の出力は最高64ビットまでのベクター44aであり、DUTディスエーブル信号44bと共に、ベクターFIFO45へと印加される。FIFO45は、完全に埋まると信号VEC__FIFO__FULL26を生成するが、この信号の意味及び用途は先に説明した。期間発生器49(後に触れる)からの信号VEC__FIFO__UNLOAD47を受けると、ベクターFIFO45の最上部にあるベクターがそこから取り出される。取り出されたベクター46は、関連するピン電子回路9を介してDUTに接続するタイミング/フォーマッティング及び比較回路52に印加される。即ち、(種々のテストサイト内の)ピン電子回路9の各々が、送信及び受信ベクター7及びピン電子回路構成情報8を、関連するタイミング/フォーマッティング及び比較回路52から受信する。

【0045】

タイミング/フォーマッティング及び比較回路52は、構成及び制御情報を受信する為にVTバス89に接続している。先にも説明したように、タイミング/フォーマッティング及び比較回路52は実際には8つのICであるが、本明細書においては、便宜上1つのエンティティとして取り扱うものとする。

【0046】

タイミング/フォーマッティング及び比較回路52は、マイクロコントローラシーケンサ19のプログラムSRAM20と同じ命令アドレス(小円で囲まれたA)によりアドレス指定される内部SRAM54を有する(外部DRAM53を内部SRAM54の代わりに使用しても良いが、これは図示していない増分カウンタにより局所的にアドレス指定される)。内部SRAM54(又は外部DRAM53)は、関連するフォーマットを有するドライブサイクル及び比較サイクルの生成を助けるものである。ドライブサイクルは、RAMの54又は53の一方から供給される事前に選択されたフォーマットを使って送信ベクターをDUTへと印加するものである。比較サイクルはDUTから供給されるベクターを受信して検査し、先に供給された比較データと一致するかどうかを判別するもので、この試験もまた、RAMから供給される事前選択されたフォーマットに基づいて行われる。ドライブサイクル及び比較サイクルはいずれも、その持続時間に関して調整可能であり、負荷を印加するかどうか、そして何時印加するのか、何時データをラッチ又はストローブするのか、信号がReturn-To-Zeroであるのかないのか、被駆動信号をその補完信号で囲むのかどうか等について適切に調整可能である(これらのオプションが上述した様々なフォーマットである)。

【0047】

タイミング/フォーマッティング及び比較回路52によって生成される比較結果には、論理値が誤りだったためにそのチャンネルが不良となったのか(機能エラー)、及び/又はその電気特性が許容範囲を超えていた為に不良となったのか(パラメータエラー)について、チャンネル毎の情報を含んでいる。更に、組み込んだ特許出願に記載されているように、複数のDUT試験が実施される場合、どのチャンネルがどのDUTに関連しているのかは既知である。これにより、4つの信号DFE0:3(いずれの番号のDUTの機能エラー)103及び4つの信号DPE0:3(いずれの番号のDUTのパラメータエラー)104の生成が可能となる。

【 0 0 4 8 】

タイミング/フォーマッティング及び比較回路 5 2 により実施される比較処理でも 6 4 ビット値が生成され、これが、回路 4 0 の論理的反転機能を有すると考えることが出来る受信ベクター逆マップ/デシリアライザ 5 7 へと供給される。(回路 5 7 の動作は S R A M 5 8 により制御されるが、これは回路 4 0 の S R A M 4 1 による制御に対応する)。そして回路 5 7 の出力 5 9 は後デコード回路 6 0 及びエラー捕捉 R A M 1 3 2 A へと印加される。後デコード回路 6 0 は入力エラー情報 5 9 と、エラー捕捉 R A M 1 3 2 A に以前に記憶されたエラー情報の両方をプログラムの基準に基づいて検分し、凝縮した、より容易に解釈出来るエラー情報を生成することが出来る。この生成したエラー情報は経路 6 1 を介して他のエラー捕捉 R A M 2 3 2 B に記憶しておくことが出来る。一例として、特定のアドレス範囲でエラーが生じた回数の計数値を作成すること等があげられるが、このような情報はいつ代替回路を起動してオンチップ修復を試みるかを決定する上で有用である。

10

【 0 0 4 9 】

次に期間発生器 4 9 及びこれに関連するタイミング S R A M 5 1 について説明する。これらは 8 ビット信号 T _ S E L 4 3 に応答して、マイクロコントローラシーケンサ 1 9 によりフェッチされた 2 0 8 ビット命令の各々について、タイミング/フォーマッティング及び比較回路 5 2 の関連する動作の継続時間を決定する。T _ S E L 4 3 は、フェッチされた命令中の異なるフィールドによって表される各種制御値及び命令 4 2 の要素である。この信号は、8 ビット値なので、2 5 6 の異なる「事象」を表す、即ち符号化することが出来る。この場合、これらの「事象」は、タイミング S R A M 5 1 に記憶され、T _ S E L によりアドレス指定される 2 8 ビット値である。アドレス指定された 2 8 ビット値 2 3 の各々は 1 9 . 5 ピコ秒の分解能で所望の継続時間を指定するものである。アクセスされた 2 8 ビット継続時間値 2 3 のシーケンスは期間 F I F O 5 0 中に記憶され、そのシーケンスの個々の要素が取り出されて、それらの意図した対応ベクター(このベクターは、ベクター F I F O 4 5 に記憶される)の取り出しに同期して印加されるようになっている。

20

【 0 0 5 0 】

F I F O 5 0 の最も古いエントリ中にある粗タイミング値フィールドは 5 ナノ秒の分解能で継続時間情報を含んでおり、この情報から、次の送信ベクターを、ベクター F I F O 4 5 からタイミング/フォーマッティング及び比較回路 5 2 に送る信号 V E C _ F I F O _ U N L O A D 4 7 が生成される。随伴信号 T I M I N G _ R E M A I N D E R 4 8 もまた、回路 5 2 へと印加される。ここで 1 9 . 5 ピコ秒という最終分解能に達する。

30

【 0 0 5 1 】

次に図 3 を参照するが、これは図 2 のブロック図に示した内部試験メモリ 8 7 の概略ブロック図 6 4 である。メモリ 8 7 は、アドレスマップ 2 9 から、種々のアドレス選別部 7 7、7 8、7 9 へと印加される 4 8 ビットのマッピング済みアドレス 3 0 を受ける。アドレス選別部はメモリセット 7 3 ~ 7 6 に関連付けられている。これらのメモリセットの各々は、例えば E C R 3 2 等の種々の機能を個別に実施することが出来る完全なメモリ機構である。これらのメモリセットのうちの 2 つ(7 3、7 4)は外部 D R A M に属し、他の 2 つは内部 S R A M に属している。2 つの外部 D R A M メモリセットは常に同じアドレス選別機能を実行するもので、従って同じアドレス選別部 7 7 を共用している。一方、内部 S R A M メモリセット 7 5、7 6 は、それら自体に関連付けられたアドレス選別部 7 8、7 9 をそれぞれ有する。これらのアドレス選別部は、アドレスを変えずにそのまま、或いは変更したアドレスを出力することができるが、そのやり方については、組み込んだ特許願の 1 つにある程度詳細に記載されている。

40

【 0 0 5 2 】

各メモリセットはメモリセットコントローラを備えている。外部 D R A M メモリセット 7 3、7 4 は D R A M メモリセットコントローラ 6 5、6 6 をそれぞれ含み、内部 S R A M

50

メモリセット75、76はSRAMメモリセットコントローラ67、68をそれぞれ含む。DUTの試験中、これらメモリセットのいずれかに向けられたメモリトランザクション用アドレスは、それぞれに関連するアドレス選別部から関連するメモリセットコントローラに到達する。4つのメモリセットは全て、基本的には内部試験メモリへと入る従来の経路（後デコード回路60をバイパスする経路）である経路59を介してデータを直接受信することが出来る。DUTの試験中は、後デコード回路60から供給され、エラー捕捉RAM232Bへと書き込まれることになるエラーデータ61は、メモリセット2及び3にそれぞれ関連しているデータ選別部82～83へと最初に印加される。データ選別部80～83は、それらがどのように構成されているか、及び、それらが実行する機能に応じて、供給されるデータに変更を加える場合もあれば加えない場合もある。アドレス及びデータ選別部はそれぞれアドレス及びデータの高速度経路であり、これらは必要とされる最高速度で動作するよう意図されている。リングバス（ここでは図示していない）が、他の手法でアドレス及びデータをメモリセットに搬送することについても後で手短かに説明する。

10

【0053】

この時点において、各々に（選別された）アドレス及び（選別された）データが入力される4つのメモリセットコントローラ65～68がある。これらのメモリセットコントローラの各々は、対応するメモリに結合している。即ち、DRAMメモリセットコントローラ73及び74はそれぞれ、外部DRAM69及び70に結合しており、SRAMメモリセットコントローラ75及び76はそれぞれ、内部SRAM71及び72に結合している。これらの構成が4つのメモリセット73～76を形成しており、これらのうちの2つ（75、76）は、容量が比較的少ない高速SRAMを有し、他の2つ（73、74）は、大容量のより遅いDRAMを有している。ここで重要な点は、DRAMメモリセットをどのようにしてSRAMメモリセットと等しい速さにするか、そしてユーザーの嗜好や試験プログラム方針に応じて、DRAMのいくつかの代替構成をどのようにして組み込むかである。従って、DRAMメモリセットコントローラ65、66は、構成可能であり、異なる種類のメモリトランザクションを実施するものでなければならず、より単純なSRAMメモリセットコントローラ67、68と全く同じ構成というわけにはいかない、ということがわかる。図を簡潔にする為に図3にはこの柔軟性を提供する構造を示していない。ここでは、各メモリセットコントローラに特定の動作モード及び所望の構成において指示を行うリングバス（ここでは図示していない）に、各メモリセットコントローラが接続されていると言うに留める。これらのモードの一部はデータの記憶方式に関わるものであり、他の一部はデータの再取得に関わるものである。結論として、各メモリセットは対応するデータ出力（62A～D）を有し、これらは後の処理の為に後デコード機構60に送られる、ということに留意されたい。更に、メモリセット0及び2からのデータ出力は、マルチプレクサ（以下、マルチプレクサをMUXとも記す）84へと印加され、その出力がデータマルチプレクサ35へと送られる記憶済みデータ33となる点にも留意されたい。同様に、メモリセット1及び3からのデータ出力はマルチプレクサ127へと印加され、その出力がアドレスビット選択マルチプレクサ37へと送られる記憶済みアドレスとなる。マルチプレクサ84及び86の存在理由及びこれらがどのように制御されるのかについての詳細は本願においては重要ではない。これらについては、組み込んだ特許出願に記載されている。

20

30

40

【0054】

次に図2に示した後デコード回路60の概略ブロック図91である図4A及び図4Bを参照する。図4Aには、図2に示すものと同様に、後デコード機構60への入力として機能する2つのデータ経路が示されている。これらの一方は受信ベクター逆マップ/デシリアライザ57からの単一ソース59であり、他方はECR（ECRを、これまで、エラー捕捉RAM132A、または、ECR1として示してきた）として動作するように構成されたメモリセットのいずれか1つに対する内部テストメモリ87からの4つのソース62A～Dである。4つのソース62A～Dについては、いずれのメモリセットにおいても、一度には1つの（アクティブな）ECRしか存在しない。5つのソース全てがマルチプレ

50

クサ 9 4 へと入力されるが、マルチプレクサ 9 4 はその中の 1 つを、種々の入力マスク 9 6、9 7、1 0 5、1 0 7 へと、及び、マルチプレクサ 9 4 の出力とこれら種々の入力マスクの出力との間で選択を行うマルチプレクサ 1 0 8 へと印加すべき入力として、選択する。

【 0 0 5 5 】

本願においては、「マスク」という語を、データフィールド内の選択したビットを何らかの配慮に基づいて排除又は包含する為のパターン及びそれに関連する機構を指す語として従来通りに使用している。マスクパターンは対象とするデータフィールドと同じ数のビットを有しており、データフィールドビットと一対一で対応しているものと考えられる。本願で使用するマスクは、無視すべき位置、即ち考慮対象から「マスクアウト」すべき位置に 1 を有している。データフィールドは、不良を 0 で表すという慣例に従う。通常、考慮対象は、マスク適用後のデータフィールド中にある 0 の存在、又はその数である。実際にマスク機能を実施する回路は後で説明する。現時点では、図 4 A 及び図 4 B の一般的な性質を検討するので、マスク処理をブロック図レベルで単純に取り扱うものとする。入力マスクのブロック図レベルにおける性質とは、マスクアウトすべきビット位置に 1 を設定して、その位置が不良表示に影響することがないようにするものである。

【 0 0 5 6 】

続いてジャムマスクレジスタ 9 3 について説明するが、これは 3 2 ビットのマスク値を受ける為にリングバス 8 5 に結合している。このマスク値は入力マスク 9 6 へと供給される一方で、そのビット毎の補数がインバータ 9 8 の処理により入力マスク 9 7 へと供給される。これらのマスクは所与のパターンの持続時間中は一定である。即ち、あるデータ値から次のデータ値へと、ルーチン的に変化することは想定されていないということである。入力マスク自体は入力データとして、マルチプレクサ 9 4 が P D 制御レジスタ 9 2 (これもリングバス 8 5 に接続している)からの信号 E R R _ D A T A _ S E L に基づいて選択したソースを受信するものである。入力マスク 9 6、9 7 の出力は、追加入力としてマルチプレクサ 1 0 8 へと供給される。

【 0 0 5 7 】

マルチプレクサ 1 0 8 は更に 2 つのマスクされた入力を受ける。これらは入力マスク 1 0 5 及び 1 0 7 (これらへの入力もまた、マルチプレクサ 9 4 の出力であり、これらのマスクもまた、インバータ 1 0 6 によって処理された補数である)の出力である。しかしながらこの場合のマスク値のソースは、マルチプレクサ 9 5 及びマルチプレクサ制御信号 B M _ S E L により選択されたバッファメモリであることが予想される。この為に、マルチプレクサ 9 5 は、入力として 4 つのソース 6 2 A ~ D を有するということに留意されたい。この構成によれば、パターンが実施されているときに、入力マスク 1 0 5 及び 1 0 7 により使用されるマスクをサイクル毎に変化させることが出来る。

【 0 0 5 8 】

明らかなように、マルチプレクサ 1 0 8 への入力として提供されるのは、アドレス指定されたデータに様々なマスクをかけたもの 4 つと、アドレス指定されたデータにマスクをかけていないもの 1 つである。マルチプレクサ 1 0 8 がこれら 5 つのデータからどれを選択し、マスクされたデータ 1 1 4 として送出するのかは、マルチプレクサ制御信号 P D _ M A S K _ M O D E により決まる。

【 0 0 5 9 】

これまでに述べた 3 つのマルチプレクサ制御信号 (E R R _ D A T _ S E L、B M _ S E L、P D _ M A S K _ M O D E) 及び後に説明するもう 1 つの信号 (P D _ E R R _ S E L) は、全て、リングバス 8 5 を介して P D 制御レジスタ 9 2 に送られるトラフィックによりその内容がどのように設定されるかに従って、P D 制御レジスタ 9 2 からの出力として生成される。

【 0 0 6 0 】

次に図 4 B を参照する。この図はどちらかというと込み入った図のようであるが、見た目程複雑ではない。左側上隅の部分について説明する。ここには解析マスク及び解析カウン

10

20

30

40

50

タ機構が32個示されている。これらがどのような機構であることを説明する。

【0061】

まず、このブロック図レベルにおいて、解析マスク (ANYL.MASK__0...ANYL.MASK__31、110a~z) は、マスクされていないデータビットの1つ以上が0であった場合に0で表示し、マスクされていないビット中に不良表示(0)が全くなかった場合に1を表示する一ビット出力を生成するという特性を有する。各マスク回路110a~zにより使用されるマスクは、関連するMASK__0...MASK__31レジスタ109~zにより供給されるが、これらのレジスタはその値を受ける為にリングバスに接続している。ここで留意すべきは2つの点である。第一には、解析マスク回路110a~zの出力が対応する解析カウンタ (ANYL.CNTR__0...ANYL.CNTR__31、112a~z) へと送られるという点である。これらのカウンタはそこに送られた0の数を計数し、それに関連するマスクにより (例えばパターン中で) エラーが発見された回数の合計を得る為に使用することができる。第二には、32のマスク出力全てがその組113内に集められ、それらを (経路61を介して) ECRに記憶するか、或いはPDエラーマスク120によって更に解析することができるという点である。

【0062】

更に説明を続けると、解析カウンタ112a~zは以下のブロック図レベルの特性を有している。これらは、対応するロードレジスタ (CNTR__0 LOAD REG., ... CNTR__31 LOAD REG., 111a~z) 中に記憶された内容によって与えられた、事前にロードされた値からカウントダウンを行うものである。ロードレジスタに記憶される内容は全て、リングバスを介してそれらに個別に送られるトラフィックにより別々に設定される。これにより、カウンタを、カウントダウンを開始する閾値に事前に設定することが出来る。これらがゼロまで計数すると、対応するカウント終了フラグ (TCF__0, ... TCF__31、115a~z) は0を表示し、それ以上のカウントダウンが禁止される (カウンタは0で止まる)。これらは、信号LOAD__COUNTERS (この信号が生じると、常にリロードが強制される) の作用により閾値がリロードされるまで0のまま留まる。また、QUALIFIED__LOADという信号もあるが、これはカウンタが未だ0に到達していない場合にそれぞれのレジスタ111a~zからカウンタ112a~zの各々へリロードを行うものであり、この場合は個別のリロードは禁止される。

【0063】

カウント終了フラグ115a~zは、集合116として集められて、解析マスク110a~zからの集合113との交互入力としてマルチプレクサ118に供給されるということに留意されたい。集合116、113はまた、マルチプレクサ117にも供給されるが、ここから、それらから選択されたうちの1つを、経路61を介してバッファメモリに格納することができる。マルチプレクサ117及び118の両方における選択は、PD制御レジスタ92から送られる制御信号PD__ERR__SELにより決定される。

【0064】

次に図4Bの他の部分を説明するが、前述した解析マスク及び解析カウンタ機構により、事前選択した閾値を超える計数値に関して生成された表示を用いて、様々なエラー状況を検出し計数することが可能であることが理解されよう。出力113及び116に存在する事象の状態を、マルチプレクサ117及び経路61を介してメモリ中に記憶することが出来る。同じ状態の事象は、マルチプレクサ118及びPDエラーマスク120 (入力型のマスクであり、そのマスクはリングバスにより設定されるエラーマスクレジスタ (ERROR MASK REG.) 119の内容に含まれる) を介して更なる解析を受けることも可能である。PDエラーマスク (PD ERROR MASK) 120からの32ビット出力121は、NANDゲート122へと供給され、そこで他の2つの「0がエラーである信号」 (ゲート129及び136の出力) と論理和演算される。これら後者の信号については後で説明する。従って、このように得られた論理和は、1が表明された信号を意味する信号PD__ERR__90として生成され、マイクロコントローラシーケンサ119に供給される。

【 0 0 6 5 】

マスクされたデータ 1 1 4 は更に図 4 B の下部に示した他の 2 つのマスク / カウンタ構造にも供給される。まず、ACCUM . ANYL . MASK 1 2 3 (そのマスクはリングバスにより設定される ACCUM . MASK REG . 1 2 4 に含まれる) について説明する。これはマスク 1 1 0 a ~ z と同様に動作する。その出力もまた、マスク 1 2 3 により 0 が提供されたときにカウントダウンを開始し、信号 LOAD __ ACCUM __ CNTR が作用したときに ACCUM . LOAD REG . 1 2 5 により閾値に事前設定される ACCUM . CNTR . 1 2 6 に供給される。レジスタ 1 2 5 中の閾値は、リングバスにより供給される。カウンタ 1 2 6 の出力は、0 までカウントダウンすることが出来、また、0 までカウントダウンすると、カウント終了フラグ TCF __ ACCUM 1 2 7 が 0 を表明し、カウンタ自体はリセットされるまで 0 のまま留まる。この時点に至るまでは要素 1 2 3 ~ 1 2 7 の構造は、解析マスク及び解析カウンタに関連して先に説明したものと本質的に同一である。相違点は、結果をどのように出力することが出来るか、及び、意図するマスクが何かについてである。要素 1 2 3 ~ 1 2 7 の目的は、エラーを含むワード全体の数を累積し、これを所望の閾値と比較することである。説明を先に進めると、要素 1 2 7 からの信号 TCF __ ACCUM は、1 ビットラッチ A __ L a t c h 1 2 8 及び OR ゲート 1 2 9 から成る 1 ビットマスクに供給される。OR ゲート 1 2 9 の出力は、閾値に一致した場合は、0 が故障を検出したことを意味する論理で表示を行うが、この出力は、PD __ ERR 9 0 の生成におけるもう 1 つの入力である。

10

【 0 0 6 6 】

図 4 B には更にもう 1 つのマスク及び累積機構が示されている。BITWISE ANYL . MASK 1 3 0、BITWISE MASK REG . 1 3 1 及び BITWISE LOAD REG . 1 3 2 は、先に説明した、これらに対応する要素 (例えば 1 2 3 ~ 1 2 5) と同じである。唯一の相違点は、レジスタ 1 3 2 へのロード信号が、その独自の信号 LOAD __ BITWISE __ ACCUM であるという点である。ブロック図レベルの特性において生じる相違点は、ラッチ BITWISE ACCUM . 1 3 3 が他のカウンタのような単なるカウンタではないという点である。要素 1 3 3 は、マスク処理後に、到達する各マスク結果中の 0 の数を計数し、これをレジスタ 1 3 2 中の初期の閾値から減分する数量として累積する。閾値が 0 まで減分されると、ラッチ 1 3 4 のフラグ TCF __ BITW は低 (ロー) を表明する。前述と同様のことが、B __ L A T C H 1 3 5 及び OR ゲート 1 3 6 から成る 1 ビットマスクにより再び開始され、この 1 ビットマスクの出力は、NAND ゲート 1 2 2 による PD __ ERR 9 0 の生成に寄与する最後の要素となる。

20

30

【 0 0 6 7 】

後デコード機構 6 0 の概要の説明の最後に、図 4 B の上部右隅に位置する 4 ビット書き込み専用レジスタ 1 3 7 の機能について簡単に説明する。これはリングバスにより設定されるもので、その出力ビットの 1 つがクロックに同期して True (1 : 真値) として出力されたときに、それに関連する名称の制御信号としてそれ自体を表明するものである。これらは上記の説明において既に触れた信号であり、上述したカウンタにより使用される各種 LOAD 信号である。これらの LOAD 信号が残留しつづけると機能障害が生じる可能性がある為、それらの信号は、所定の機能を果たした後消滅するように設定されている。この為には、OR ゲート 1 3 0 はこれらの LOAD 信号のうち 1 つ以上が True であることを検出し、これを AND ゲート 1 3 9 へと供給し、次のクロックサイクルが生じたときに、レジスタ 1 3 7 の内容を全て 0 にリセットする。これはワンショットタイプの機構である。

40

【 0 0 6 8 】

最後に、これまでに参照した図には示していないが (図 7 において図示) 、各種カウンタ中の値は、リングバス上に読み出すことが出来る。

【 0 0 6 9 】

次に図 5 を参照する。図 5 は、各種マスクがどのように機能するかを示すブロック図 1 6

50

5である。入力マスクは32ビットの入力データ値166と、32ビットのマスク値167とからなる。マスク値167はレジスタから得るようにしても（静的方式の場合）、或いはメモリからマスクデータを読み出す等、より一時的な方法で得るようにしても良い。いずれにしても、2つの要素166と167の間には、ゲート168a～zによるビット毎の論理和演算が実施される。この結果得られた32ビット数値169が入力マスクに対する出力となる。解析マスクについては、数値169がANDゲート170へと供給され、ANDゲート170は、数値169のいずれかのビットが0であった場合にその出力として0を生成する。

【0070】

ビット毎累積機構（BITWISE ACCUM.）133の性質を理解する為に、この回路のブロック図140である図6を参照する。これは、マスクレジスタ（BITWISE MASK REGISTER）141を含んでおり、このレジスタの32ビット値142は32ビットの入力データ値143とビット毎にNOR演算される。この時点における図5のマスク構成との相違点は、ゲート144a～zがORゲートではなく、NORゲートであるという点である。この違いの結果、マスク処理後に残った不良に対して生成されるのは、0ではなく1ということになる。これは、このようになっていれば不良表示の発生数の和を取って総数を得ることがより容易な為である。これらはNORゲートをグループ化して対とし、これらのNORゲート対の出力を加算器145a～oに供給することにより加算される。加算器145a～oはそれら自体も対を構成しており、それらの出力は更に他の加算器146a～hにさらに供給される。この方法によれば、入力値143中の0の総数は、加算器147の出力に出力されるまで個別に累積されることになる。そしてこの出力は、BITWISE LOADレジスタ（BITWISE LOAD REG.）132内の閾値から開始される値を減分するために利用される。この閾値は、マルチプレクサ150へのマルチプレクサ制御信号LOAD__BITWISE__ACCUMの最初の表明によって、BITWISE ACCUM. レジスタ151に入れられたものである。これはレジスタ151中の値が加算器147の出力により減分されるように設定される。これは減算回路148により実施され、その後その出力はマルチプレクサ150を通過してレジスタ151中に記憶される次の値となる。この結果、レジスタ151は、加算器147が生成した和によりカウントダウンされることになる。レジスタ151が0を過ぎると、カウント終了フラグ機構134が0を表明する。

【0071】

図7は上述した解析及び累積カウンタ156の性質を説明する為のブロック図154である。解析カウンタ及びそれらのロードレジスタ157は24ビットであるが、累積用のものは32ビットである。いずれの場合も、マスクから計数される0はNORゲート158へと供給されるが、このNORゲート158に対する他の入力信号はYES__ZERO164である。YES__ZEROは、カウンタ156の（True：真値）出力により駆動され、カウンタ156の内容が0のときには常にYES__ZERO=ONE（1）を生成する機能を有するNORゲート159により生成される。YES__ZEROは2つの役割を持つ。第一には、マスクからの入力にかかわらず、NORゲート158の出力を強制的に0とすることにより更なる計数を禁止することである。カウンタ156は、実際には1を計数するので、計数はされないことになる。第二には、TCフラッチ160を設定することであり、これによりそのNOT-Q出力は0となる。

【0072】

解析カウンタについては、0となるTCF信号はANDゲート161を通じて作用し、信号QUALIFIED__LOADの作用を禁止する。これにより、前述したようにレジスタ156のロードが阻止される。QUALIFIED__LOADが禁止されない場合、これはORゲート162を通過し、ここでQUALIFIED__LOADもしくはLOAD__COUNTERSによりカウンタ156がレジスタ157中の値でロードされ、更にTCフラッチ160がリセットされることになる。これは点線171中の回路で示されている。

10

20

30

40

50

【 0 0 7 3 】

累積カウンタについては、点線 1 7 1 内の要素が点線 1 7 2 中に示される単純な構成に置き換えられる。この場合、直接的にカウンタ 1 5 6 をロードし、T C F ラッチ 1 6 0 をリセットする為に信号 L O A D _ A C C U M _ C N T R が印加される。

【 0 0 7 4 】

カウンタ 1 5 6 の出力はバスインターフェース 1 5 5 に供給されということに留意されたい。その値は、このバスインターフェースからリングバス上に送ることができる。

【 0 0 7 5 】

まとめとして、後デコード機構にとって望ましい特徴を幾つか述べる。ここでは、図 4 A、図 4 B、図 5、図 6 及び図 7 の主要構成をどのように利用して所望の目的を達成することができるかについて簡単に説明する。改良型後デコード機構の「オンザフライ解析」機能は、経路 5 9 がマルチプレクサ 9 4 に接続していることにより実現される。これにより試験結果を内部試験メモリ中に事前記憶する必要なく解析を続行することが可能となる。

【 0 0 7 6 】

オンザフライで解析を行う能力と、アドレス毎に変えることが出来るマスクとしてバッファメモリの内容を使用することができる能力とが結びつくことにより、新たに生じたエラーだけを認識することが出来る機能が助長される。これを「増分的後デコード解析」と呼ぶものとする。これは、E C R 内のエラーデータを先の試験中に得て、そのエラーデータをバッファメモリへとコピーし、その後そのバッファメモリをマスクとして使用することで新たな試験中は古いエラーを隠し、新たなエラーだけを識別して更なる処理を実施できるようにするというものである。この機能は経路 6 2 A ~ D がマルチプレクサ 9 5 に結合することでサポートされている。これはまた、入力マスク 1 0 5 に結合したインバータ 1 0 6 によってもサポートされている。これは、これらのインバータが、「0 は不良を表示する」というフォーマットを、マスクにより隠されるビット位置を示す論理 1 に反転することができるからである。しかしながら最初の試験においては、その試験の第一のパスをマスクなしで捕捉できるようにする為に、入力マスク 9 6 及びマルチプレクサ 9 4 と共に、ジャムマスクレジスタ 9 3 中に全 0 のマスクを使用することが望ましい。その後、マルチプレクサ 9 4 の代わりにマルチプレクサ 9 5 を使用して増分型処理モードを開始することが出来る。

【 0 0 7 7 】

上述した増分的分析機構は E C R からバッファメモリへのコピー能力に依存するものである。内部試験メモリであるメモリシステム自体は、そのようなコピー処理を実現する手段を持たない。しかしながら、この処理は上述した改良型後デコード機構の助けを借りれば可能である。具体的に言えば、経路 6 2 A ~ D のいずれかを起点とし、経路 6 1 を終点とする無変更データの伝送用経路が存在する。この無変更経路はマルチプレクサ 9 4 から始まり、そこからマルチプレクサ 1 0 8 へと入り、適正な解析マスク 1 1 0 a ~ z 構成を通過し、経路 1 1 3 に沿ってマルチプレクサ 1 1 7 にて終わる。適正な解析マスク 1 1 0 a ~ z 構成とは、各々が、そのマスク番号に対応するビット位置（この位置においては 0）以外の位置において全て 1 であるマスクを有するというものである（この一見奇異とも思われる条件は、各解析マスクが、単一ビットしか出力せず、及び、データを変化させることなく通過させるように複数のこれらのマスクを構成するという事実によるものである）。

【 0 0 7 8 】

改良型後デコード機構 6 0 によれば、いくつかの場合には、同時解析をおこなうことができる。例えば、各累積解析機構 1 2 3 ~ 1 2 9 及びビット毎解析機構 1 3 0 ~ 1 3 6 が別々にそれらのタスクを実施すると同時に、解析マスク 1 1 0 a ~ z の集合は、そのタスクを実行することが出来る。これが可能であるのは、これらの機構の各々が、別々に、そして同時にそのデータ 1 1 4 にさらされる為である。

【 0 0 7 9 】

累積解析機構 1 2 3 ~ 1 2 9 が経路 1 1 4 上に出現するデータワード中に差別化されていないエラーの存在又は不在に関して（すなわち、データワード内のエラーが複数あるかな

10

20

30

40

50

いかを区別することなく)作用するものであることが理解されよう。即ち、データワード中の1つ以上のエラーの存在は、単一のエラーと同じように取り扱われるのである。対照的に、ビット解析機構130~136は経路114上に現れるデータワード中の実際のエラーの数に応答する。

【0080】

最後に留意すべきは、カウント終了フラグの集合値116がマルチプレクサ117へと印加され、これをマルチプレクサ117から経路61を介してバッファメモリへと記録することが出来るという点である。以下は、この機能が有用である理由を示す例である。ECRがZ(すなわち、Zブロックの)TagRAMとして動作するように構成されていると想定する。各ブロックに関連するエラーに毎に個々の閾値を設けることが望ましい。これを行うには、各ブロックアドレスに対するカウント終了フラグをECRに記録し、その後ブロック間の解析カウンタをリセットする。記録された情報は、それぞれの閾値を超えたブロックを示すことになる。

10

【0081】

以下においては、本発明の種々の構成要件の組み合わせからなる例示的な実施態様を示す。

1. 被験メモリ(14)の試験結果をメモリテスト(1)で解析する方法であって、
(a)被験メモリに試験処理を実施して結果データ(56)を生成するステップと、
(b)作業位置に前記結果データを前もって記憶することなく、前記結果データにマスク(96,97,105,110,120,167,170)を適用するステップと、
(c)前記マスクの適用後に、前記結果データ中に残るエラー表示に従ってカウンタ(112)を起動するステップ

20

からなる、方法。

2. 前記カウンタの内容を閾値(151/157)と比較し(148/156)、閾値を満たしたときにエラー信号を生成するステップを更に含む、上項1の方法。

【0082】

本願の概要は以下の様である。後デコード機構(60)へのデータ経路を変更することにより、後デコードは、データが内部テストメモリ(87)内の宛先メモリ構造(32A)に配置される前または配置されるときに、そのデータを処理することが可能になる。他のデータは、後デコード機構に供給される前に、まず内部テストメモリ内のメモリ構造に依然として配置される。広範なマスク機能(96,97,105,107,110,120,167,170)をエラーテーブルのコピー機能と組み合わせることにより、新しいテストに対して増分式後デコード解析を可能にすると共に、以前のテスト中に不良として既にわかっている位置におけるエラーをカウントすることを回避する。ワード内のエラーとビットエラーの両方を累算することができる。後デコード機構は、しばしば、データがオンザフライで供給されるか、内部テストメモリ内の構造から供給されるかに関係なく、データを一回調べることにより複数のタイプの結果を生成することができる。後デコード機構は、エラーのアクティビティに関するものを決定するための閾値を表す予めロードされた値からカウントダウンを行うカウンタ(112)を備えている。カウントが0になるとカウント終了フラグ(160)が生成される。種々のカウント終了フラグの値が、データを内部テストメモリに記録するときにいつでも使用できる。カウンタには、しばしば、1つのテスト段階の終了時及び次の段階の準備の際に、初期カウント値(111)がリロードされる。カウンタが各初期値レジスタを有し、単一コマンドを受信したときにそれらのレジスタがカウンタにリロードすることができるように、カウンタを構成することによって、オーバーヘッドのいくつかを削減し、有用性を増すことができる。カウンタに対してカウント終了フラグを提示することにより、カウンタがその初期値レジスタからリロードしないようにすることができる(161)。

30

40

【0083】

【発明の効果】

本発明の改良された後デコード機構によれば、メモリシステムの試験に関するオーバーヘッドを削減することができるので、試験時間を短縮することが可能となる。

50

【図面の簡単な説明】

【図 1】本発明に従って構成された、広範囲に再構成可能な不揮発性メモリテストの略ブロック図である。

【図 2】図 1 の DUT テスタ 6 を拡大して示した略ブロック図である。

【図 3】図 2 のブロック図に示した内部試験メモリ機構の機能ブロックの略図である。

【図 4 A】図 2 に示した後デコード機構 60 の略ブロック図である。

【図 4 B】図 2 に示した後デコード機構 60 の略ブロック図である。

【図 5】図 4 のブロック図中に用いたマスク回路のブロック図である。

【図 6】図 4 のブロック図中に用いたマスク及びビットワイズカウンタ回路のブロック図である。

【図 7】図 4 B に用いた特定のカウンタ及びカウント終了フラグ回路の略ブロック図である。

【符号の説明】

1 メモリテスト

14 被験メモリ

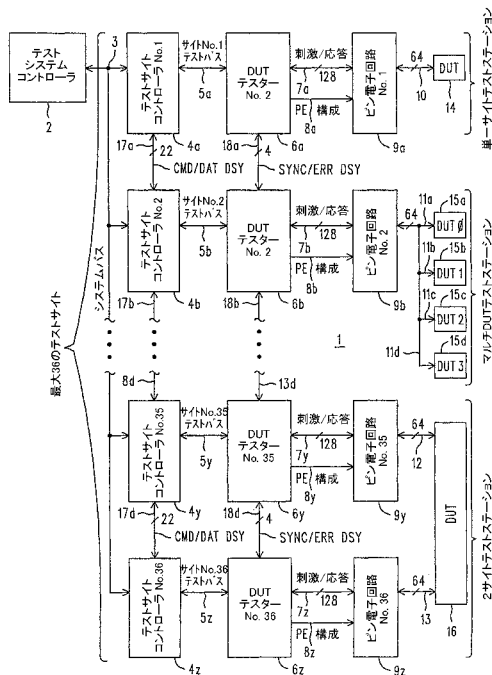
56 結果データ

96、97、105、110、120、167、170 マスク

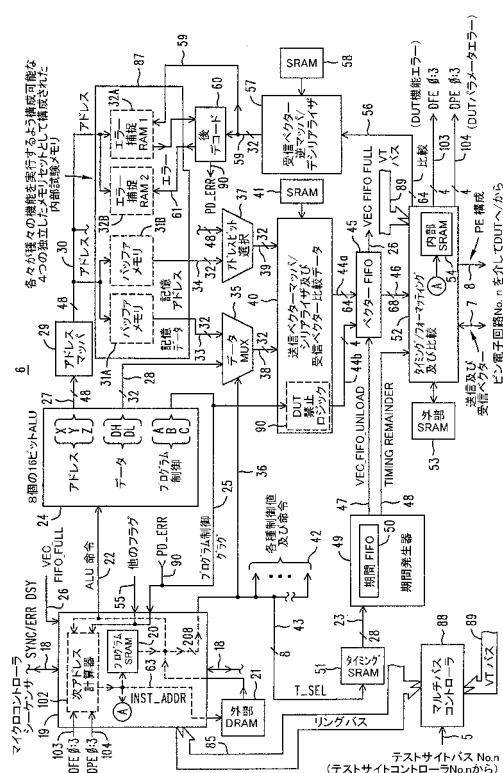
112 カウンタ

10

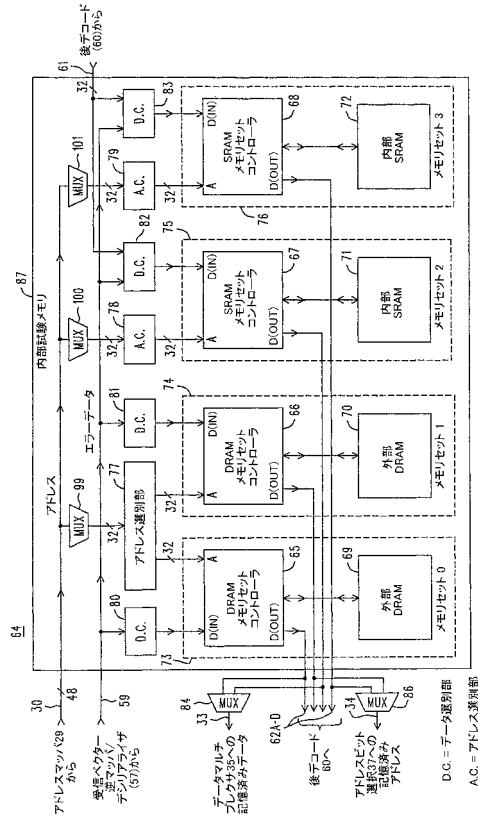
【図 1】



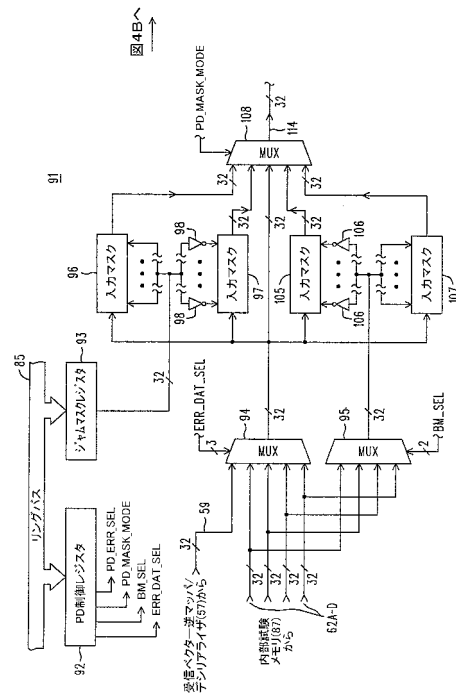
【図 2】



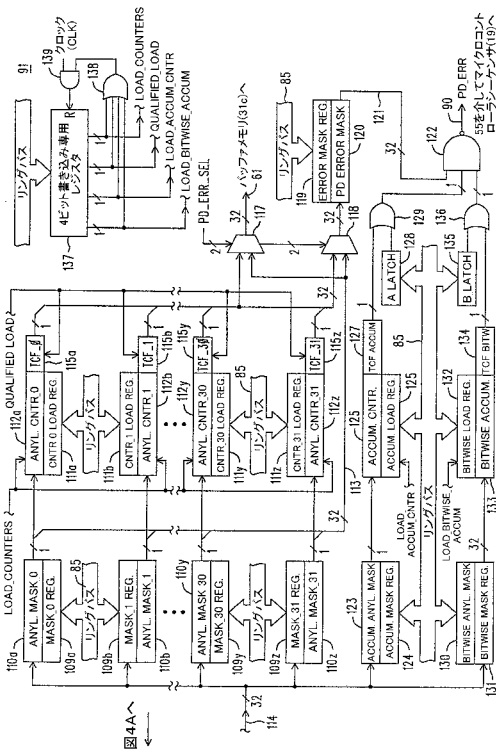
【図 3】



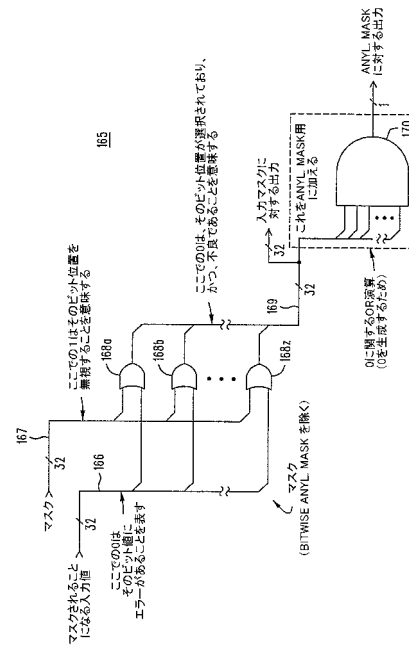
【図 4 A】



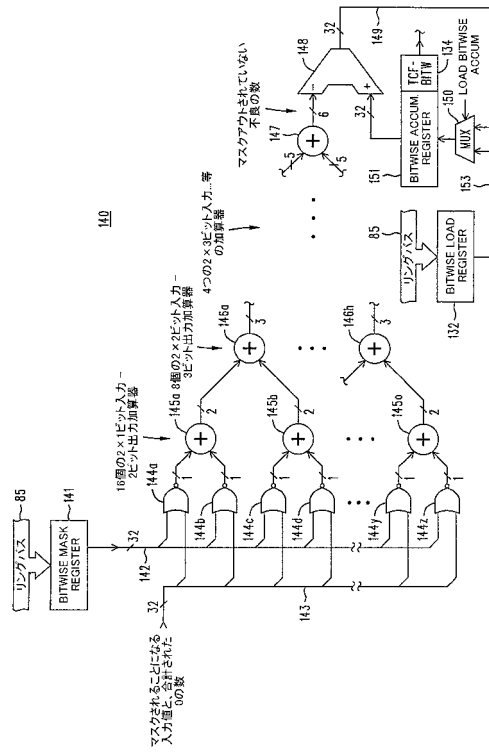
【図 4 B】



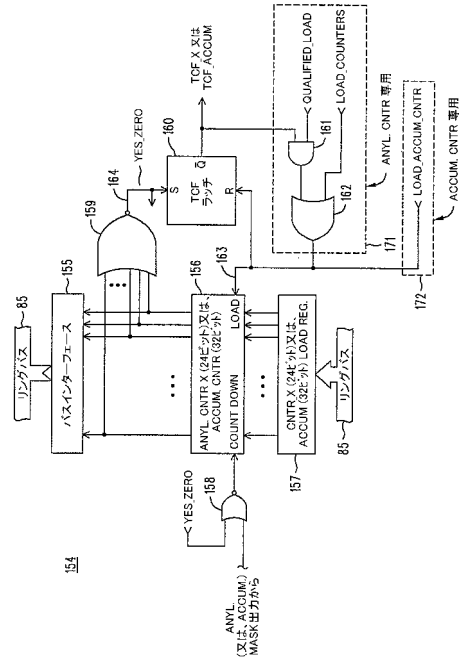
【図 5】



【図 6】



【図 7】



フロントページの続き

(74)代理人 100107364

弁理士 斉藤 達也

(72)発明者 ステファン・ディー・ジョーダン

アメリカ合衆国コロラド州 8 0 5 2 8 , フォートコリンズ , ファントム・クリーク・コート・2 5
2 1

(72)発明者 ジョン・エム・フリースマン

アメリカ合衆国コロラド州 8 0 5 2 6 , フォートコリンズ , グレイ・フォックス・ロード・4 5 0
7

(72)発明者 サミュエル・ユー・ウォン

アメリカ合衆国カリフォルニア州 9 5 0 1 4 , クパチーノ , リリー・アベニュー・8 7 3

審査官 須原 宏光

(56)参考文献 特開平 1 1 - 0 9 6 7 9 2 (J P , A)

特開平 0 9 - 0 6 3 3 0 0 (J P , A)

特開平 1 1 - 0 8 6 5 9 3 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G11C 29/00