

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4795025号
(P4795025)

(45) 発行日 平成23年10月19日(2011.10.19)

(24) 登録日 平成23年8月5日(2011.8.5)

(51) Int. Cl. F 1
G 0 6 F 15/80 (2006.01) G O 6 F 15/80
G 0 6 F 11/20 (2006.01) G O 6 F 11/20 3 1 O F

請求項の数 3 (全 15 頁)

<p>(21) 出願番号 特願2006-6295 (P2006-6295) (22) 出願日 平成18年1月13日(2006.1.13) (65) 公開番号 特開2007-188315 (P2007-188315A) (43) 公開日 平成19年7月26日(2007.7.26) 審査請求日 平成21年1月6日(2009.1.6)</p> <p>前置審査</p>	<p>(73) 特許権者 000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号 (74) 代理人 100125254 弁理士 別役 重尚 (72) 発明者 相澤 栄治 東京都大田区下丸子3丁目30番2号 キ ヤノン株式会社内 審査官 井上 宏一</p>
---	--

最終頁に続く

(54) 【発明の名称】 ダイナミックリコンフィギャラブルデバイス、制御方法、及びプログラム

(57) 【特許請求の範囲】

【請求項1】

構成情報を保持する保持手段と、
 演算器を有する処理ユニットを複数備え前記構成情報に従った前記演算器の構成及び前記処理ユニット間の接続に基づいて処理を行うデータ処理手段と、
 前記データ処理手段の備える前記複数の処理ユニットの各々に対応し、対応する処理ユニットにエラーが発生していることを検出する複数の検出手段と、
前記検出手段によりエラーが発生している処理ユニットが検出されたときに前記データ処理手段が処理していたデータの再読み込みを行うために、次にデータを読み出すアドレスと前記複数の処理ユニットで処理中のデータ数とに基づいて再読み込みを行う記憶手段のアドレスを計算する計算手段と、

10

前記検出手段の結果に応じて、前記検出手段によりエラーが検出された処理ユニットが処理に使用されないように、前記演算器の構成及び前記処理ユニット間の接続を動的に変更させる構成情報を前記保持手段に設定し、前記処理ユニットによって前記計算手段により計算したアドレスからデータを読み込ませる再構成処理手段と、
 を備えることを特徴とするダイナミックリコンフィギャラブルデバイス。

【請求項2】

構成情報を保持する保持手段と、演算器を有する処理ユニットを複数備え前記構成情報に従った前記演算器の構成及び前記処理ユニット間の接続に基づいて処理を行うデータ処理手段と、を備えるダイナミックリコンフィギャラブルデバイスの制御方法であって、

20

前記データ処理手段の備える前記複数の処理ユニットの各々に対応する複数の検出手段が、前記ダイナミックリコンフィギュラブルデバイス内の前記処理ユニット毎にエラーの発生を検出する検出手順と、

前記検出手順によりエラーが発生している処理ユニットが検出されたときに前記データ処理手段が処理していたデータの再読み込みを行うために、次にデータを読み出すアドレスと前記複数の処理ユニットで処理中のデータ数とに基づいて再読み込みを行う記憶手段のアドレスを計算する計算手順と、

前記検出手順におけるエラー検出に伴い前記エラーが検出された処理ユニットが処理に使用されないように、前記演算器の構成及び前記処理ユニット間の接続を動的に変更させる構成情報を前記保持手段に設定し、前記処理ユニットによって前記計算手順により計算したアドレスからデータを読み込ませる再構成処理手順と、を備えることを特徴とする制御方法。

10

【請求項 3】

構成情報を保持する保持手段と、演算器を有する処理ユニットを複数備え前記構成情報に従った前記演算器の構成及び前記処理ユニット間の接続に基づいて処理を行うデータ処理手段と、を備えるダイナミックリコンフィギュラブルデバイスの制御方法をコンピュータに実行させるコンピュータ読み取り可能なプログラムであって、

前記データ処理手段の備える前記複数の処理ユニットの各々に対応する複数の検出手段からの信号に基づいて、前記ダイナミックリコンフィギュラブルデバイス内の前記処理ユニット毎にエラーの発生を検出する検出手順と、

20

前記検出手順によりエラーが発生している処理ユニットが検出されたときに前記データ処理手段が処理していたデータの再読み込みを行うために、次にデータを読み出すアドレスと前記複数の処理ユニットで処理中のデータ数とに基づいて再読み込みを行う記憶手段のアドレスを計算する計算手順と、

前記検出手順におけるエラー検出に伴い前記エラーが検出された処理ユニットが処理に使用されないように、前記演算器の構成及び前記処理ユニット間の接続を動的に変更させる構成情報を前記保持手段に設定し、前記処理ユニットによって前記計算手順により計算したアドレスからデータを読み込ませる再構成処理手順と、

をコンピュータに実行させるプログラム。

【発明の詳細な説明】

30

【技術分野】

【0001】

本発明は、実行中に動的再構成可能デバイスの故障が発生した後も処理を継続可能にする場合に好適なダイナミックリコンフィギュラブルデバイス、制御方法、及びプログラムに関する。

【背景技術】

【0002】

近年、LSIの設計期間を短縮する技術に対する必要性が高まっている。これに伴い、LSI設計期間を短縮する1つの解決策として、動的再構成可能デバイス(ダイナミックリコンフィギュラブルデバイス)が注目されている。動的再構成可能デバイスとは、LSI中に演算器(ALU)を有する処理ユニット(プロセッシングエレメント)を複数組み込み、処理ユニット内部のALU制御及び処理ユニット間の接続を動的(LSI稼動中)に切り替え可能としたデバイスである。

40

【0003】

他方、回路を再構成可能なデバイスとしては、FPGA(Field Programmable Gate Arrays)のようにトランジスタ間の接続を変更可能なアーキテクチャもある。しかしながら、FPGA自体は、回路を動的に変更するには時間がかかるため、アプリケーションプログラム(以下アプリケーション)の実行中に回路を動的に制御することは難しい。

【0004】

これに対し、動的再構成可能デバイスは、回路を高速に切り替えることが可能である。

50

その理由は、FPGAはゲートレベルで回路の変更を行うのに対して、動的再構成可能デバイスはプロセッシングエレメント（PE）単位で回路変更を行うためである。これにより、回路の変更に必要な構成情報が少なく済むことに加えて、構成を変更するための回路が簡単になることで、高速な構成情報の変更が可能となる。

【0005】

動的再構成可能デバイスは、アプリケーションの実行中に回路を動的に切り替えることにより、同一のLSIで複数のアプリケーションに対応することが可能となる。代表的な動的再構成可能デバイスとしては、DAPDNA、DRPなどがある。どちらも、数百のプロセッシングエレメントを内部に備えることで、高い処理性能を維持している。

【0006】

上述した動的再構成可能デバイスに関しては各種の技術が提案されている（例えば、特許文献1参照）。

【特許文献1】特開2004-005739号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

通常、LSIには、LSI製造後のアプリケーションの実行中に故障（エラー）が発生した場合に対応するため、エラー原因が何かを特定するためのエラー検出回路が組み込まれている。そのため、LSIは、アプリケーションの実行中にエラーが発生した場合にエラーを即座に検出することができ、エラー内容に対応したメンテナンスが行われる。発生したエラーが致命的なエラーであった場合はLSIの交換が行われ、発生したエラーが致命的でない場合は性能を落として処理を継続させるなどの措置がとられる。

【0008】

しかしながら、上記従来の動的再構成可能デバイス自体はエラー検出手段を備えていない。そのため、LSI製造後のアプリケーションの実行中に動的再構成可能デバイスでエラーが発生した場合、エラーを即座に検出することができない。これは、ミッションクリティカルなシステム（高度な信頼性や耐障害性が必要な基幹系システム）で利用するには致命的である。更に、動的再構成可能デバイスがエラー検出手段を備えていないことは、致命的でないエラーが発生した場合にもLSIを交換しなければならず、コストが増加するという問題もある。

【0009】

本発明の目的は、動的再構成可能なデバイスが有する複数のプロセッシングエレメントのいずれかに故障などのエラーが発生した場合であっても、動的に且つ高速に回路を再構成することを可能としたダイナミックリコンフィギュラブルデバイス、制御方法、及びプログラムを提供することにある。

【課題を解決するための手段】

【0010】

上述の目的を達成するために、本発明のダイナミックリコンフィギュラブルデバイスは、構成情報を保持する保持手段と、演算器を有する処理ユニットを複数備え前記構成情報に従った前記演算器の構成及び前記処理ユニット間の接続に基づいて処理を行うデータ処理手段と、前記データ処理手段の備える前記複数の処理ユニットの各々に対応し、対応する処理ユニットにエラーが発生していることを検出する複数の検出手段と、前記検出手段の結果に応じて、前記検出手段によりエラーが検出された処理ユニットが処理に使用されないように、前記演算器の構成及び前記処理ユニット間の接続を動的に変更させる構成情報を前記保持手段に設定する再構成処理手段と、を備えることを特徴とする。

【発明の効果】

【0011】

本発明によれば、動的再構成可能なデバイスが有する複数のプロセッシングエレメントのいずれかに故障などのエラーが発生した場合であっても、動的に且つ高速に回路を再構成することができる。特に、簡単な構成によって、プロセッシングエレメントのエラー検

10

20

30

40

50

出後も処理を継続することが可能となる。

【発明を実施するための最良の形態】

【0012】

以下、本発明の実施の形態を図面を参照して説明する。

【0013】

図1は、本発明の実施の形態に係るデバイス故障検出装置としての動的再構成可能デバイス故障検出装置の構成を示すブロック図である。

【0014】

図1において、動的再構成可能デバイス故障検出装置は、エラー情報保持部105、プロセッサ107、プロセッシングエレメントマトリックスコア120、プロセッシングエレメントマトリックスリセット制御部121、外部メモリ145を備えている。更に、プロセッシングエレメントマトリックスコア120は、PEマトリックス101、ロードバッファ110、ストアバッファ111、入力データ合計値制御部112、入力データ値バッファ113を備えている。以下の説明ではプロセッシングエレメントを適宜PEと表記する。

【0015】

動的再構成可能デバイス故障検出装置は、LSI製造後のアプリケーションの実行中に動的再構成可能デバイスが故障した場合、構成情報の変更及びエラーが発生した処理の再実行を行うことで、故障が発生した後も処理を継続可能にするものである。

【0016】

PEマトリックス101は、内部に1つ以上の演算器を有する処理ユニット(プロセッシングエレメント)を複数備え、演算器の構成及び処理ユニット間の接続を、後述の構成情報の変更により動的に変更可能な動的再構成可能デバイスである。PEマトリックス101は、最小の処理ユニットであるプロセッシングエレメントの2次元アレイ構造を内部に備える。ロードバッファ110は、プロセッシングエレメントに外部メモリ145から入力される入力データを一時的に保存する。

【0017】

ストアバッファ111は、プロセッシングエレメントから出力されるデータを一時的に保存する。ストアバッファ111の出力データは、外部メモリ145に送出される。入力データ合計値制御部112は、PEマトリックス101内で現在使用している入力データ数の合計値を算出する制御を行うものであり、PEマトリックス101内で処理中のデータ数の算出に使用する。入力データ値バッファ113は、ストアバッファ111内のデータが使用した入力データ数を保存する。

【0018】

エラー情報保持部105は、PEマトリックス101で発生したエラーを示すエラー情報を保持する。プロセッサ107は、構成情報書き換え信号144により外部メモリ145内の構成情報の書き換えを行い、また、リセット指示信号142によりPEマトリックスリセット制御部121に対しリセット指示を行う。PEマトリックスリセット制御部121は、PEマトリックス101をリセットする制御を行う。外部メモリ145は、入力データ、出力データ、構成情報104を記憶する。構成情報104は、プロセッシングエレメントのコンフィグレーション情報である。

【0019】

割り込み信号140は、エラー情報保持部105にエラー情報がセットされたことをプロセッサ107に通知するための信号である。処理停止信号139は、エラー情報保持部105にエラー情報がセットされた際にPEマトリックス101を停止するための信号である。リセット信号143は、PEマトリックス101をリセットするための信号である。前記信号以外の他の信号については後述する。

【0020】

図9は、PEマトリックス101内部の最小構成要素であるプロセッシングエレメントの構成を示すブロック図である。

【 0 0 2 1 】

図9において、ALU102は、プロセッシングエレメント150内に設けられた演算器である。エラー検出回路103は、プロセッシングエレメント150内に設けられると共にALUを内蔵しており、該ALUの演算結果と上記ALU102の演算結果とを比較することで（ALUの二重化）、プロセッシングエレメントのエラーの検出を行う。加算器114は、プロセッシングエレメントで処理する入力データの数の加算に使用する。入力データ130、131は、ALU102に入力されるデータであり、他のプロセッシングエレメントまたはロードバッファ110から入力される。

【 0 0 2 2 】

信号132、133は、プロセッシングエレメントへの入力データ130、131にそれぞれ対応する入力データ数を示す信号であり、入力データ130、131が作成される際に元データが幾つ消費されたかを示している。信号135は、ALU102で計算された計算結果を示す信号である。信号136は、エラー検出回路103でエラー検出された結果を示す信号である。信号137は、加算器114で加算された入力データ数を示す信号である。

10

【 0 0 2 3 】

次に、本実施の形態の動的再構成可能デバイス故障検出装置によるエラー検出処理及びエラー制御方法について図1及び図9に基づき説明する。

【 0 0 2 4 】

例えば、LSI製造後にプロセッシングエレメント内のALU102が故障した場合、プロセッシングエレメント内のエラー検出回路103は、ALU102の出力結果とエラー検出回路103の出力結果とを比較する。そして、出力結果が異なっていた場合に、エラー検出回路103はエラーを検出する。エラー検出回路103で検出されたエラー情報は、フリップフロップ151を経由してエラー信号136として出力される。エラー信号136として出力されたエラー情報は、エラー情報保持部105にセットされ保持される。尚、エラー情報はプロセッシングエレメント毎に通知される。

20

【 0 0 2 5 】

エラー情報保持部105にエラー情報の値が新たにセットされると、エラー情報保持部105は、割り込み信号140をアサートし、プロセッサ107に対して割り込みを通知する。同時に、エラー情報保持部105にエラー情報の値がセットされると、エラー情報保持部105は、PEマトリックスコア120の処理を停止する処理停止信号139をアサートし、PEマトリックスコア120での処理を停止させる。

30

【 0 0 2 6 】

信号141は、エラー情報保持部105内のエラー情報として出力する信号である。プロセッサ107は、割り込み信号140を受信した際、信号141（エラー情報）をリードすることで、故障したプロセッシングエレメントに関する情報を取得する。割り込みマスク信号160は、割り込み信号140をマスクするための信号である。プロセッサ107は、割り込みマスク信号160をアサートすることで、割り込み信号140をマスクする。

【 0 0 2 7 】

プロセッサ107は、故障したプロセッシングエレメントに関する情報を取得した後、故障したプロセッシングエレメントを使用しないようにするため構成情報104を書き換える。また、プロセッサ107は、入力データ合計値制御部112からPEマトリックスコア120内で使用されている入力データ数を信号612より取得し、入力データを再ロードする際の外部メモリアドレス算出論理（後述）を用いて再ロードアドレスを算出する。その後、プロセッサ107は、リセット指示信号142をアサートする。

40

【 0 0 2 8 】

これに伴い、PEマトリックスリセット制御部121は、リセット信号143をアサートすることで、PEマトリックスコア120をリセットする。リセット後、プロセッサ107は、信号190をアサートすることにより処理停止信号139をデアサートすること

50

で、PEマトリックスコア120を処理可能にする。更に、プロセッサ107は、上記算出した再ロードアドレスから入力データをPEマトリックスコア120に再ロードする。これにより、PEマトリックスコア120における処理が正常に再開される。

【0029】

図2は、動的再構成可能デバイス故障検出装置のエラー情報保持部105の構成を示すブロック図である。

【0030】

図2において、エラー情報保持部105は、複数のフリップフロップ202、フリップフロップ204、複数のフリップフロップ210、フリップフロップ221、レジスタ222、複数のアンドゲート、複数のオアゲートを備えている。

10

【0031】

エラー信号136は、PEマトリックス101からエラー情報保持部105に出力された信号(エラー情報)である。フリップフロップ202は、エラー信号136(エラー情報)の値を内部にセットする。フリップフロップ202は、セットされた値がエラーであった場合、フリップフロップ内に値を保持する。フリップフロップ202は、セットされた値がエラーでない場合は、PEマトリックス101から出力された次のエラー信号136(エラー情報)をセットする。

【0032】

リセット信号180は、PEマトリックスリセット制御部121からエラー情報保持部105に出力された信号である。リセット信号180は、動的再構成可能デバイス故障検出装置の電源投入時にアサートされ、エラー情報保持部105内の全フリップフロップをリセットする際に利用される。

20

【0033】

割り込みマスク信号160は、プロセッサ107からエラー情報保持部105に出力された信号である。フリップフロップ210は、割り込みマスク信号160の値をセットする。フリップフロップ210は、エラー情報のマスク用のフリップフロップである。フリップフロップ202から出力されたエラー情報と、フリップフロップ210にセットされたマスク情報との“AND”(論理積)をとることで、エラー情報をマスクする。

【0034】

信号140は、フリップフロップ210によりマスクされた後の全エラー情報の“OR”(論理和)をとった信号であり、エラー情報保持部105からプロセッサ107に対する割り込み信号140として使用される。

30

【0035】

信号190は、プロセッサ107からエラー情報保持部105に出力される、PEマトリックスコア120の処理を停止する処理停止信号139をマスクするための信号である。フリップフロップ221は、処理停止信号139のマスク用のフリップフロップである。フリップフロップ221は、信号190の値をセットする。

【0036】

レジスタ222は、処理停止信号139を出力する。レジスタ222は、エラー情報保持部105内の全エラー情報の“OR”(論理和)をとった値(信号)と、フリップフロップ221にセットされた値との“AND”(論理積)をとった値(信号)をセットする。レジスタ222は、内部にエラー情報がセットされると、フリップフロップ221の値によりエラー情報がクリアされるまで処理停止信号139をアサートし続ける。

40

【0037】

処理停止信号139は、PEマトリックスコア120の処理を停止する信号として使用される。信号141は、フリップフロップ202から出力された信号(エラーPE情報)である。プロセッサ107は、信号141(エラーPE情報)に基づき、故障したプロセッシングエレメントの判定を行う。

【0038】

次に、本実施の形態における入力データの再ロード時のアドレスの算出方法について説

50

明する。

【 0 0 3 9 】

本実施の形態では、プロセッシングエレメントでエラーが発生した場合、一旦、エラー情報保持部 1 0 5 により P E マトリックスコア 1 2 0 の処理を停止し、P E マトリックスリセット制御部 1 2 1 により P E マトリックスコア 1 2 0 内の全データを無効化する。更に、エラーが発生したプロセッシングエレメントを使用しないようにするためプロセッサ 1 0 7 により構成情報を変更した後、無効化したデータに相当する入力データを外部メモリ 1 4 5 から再ロードして P E マトリックスコア 1 2 0 の処理を再実行する。これにより、P E マトリックスコア 1 2 0 の処理を継続する方式を採用する。

【 0 0 4 0 】

そこで、上記方式を実現するため、外部メモリ 1 4 5 から P E マトリックスコア 1 2 0 に再ロードする入力データのアドレスの計算を行う必要がある。再ロードするアドレスは、正常に処理を継続していた場合に次にデータを読み出すアドレスから、P E マトリックス 1 0 1 内で使用中の入力データ数を差し引いたアドレスとなる。再ロード時のアドレスの計算式を次に示す。

【 0 0 4 1 】

$$\begin{aligned} \text{再ロード時のアドレス} &= \text{次にデータを読み出すアドレス} \\ &\quad - \text{P E マトリックス内で処理中の入力データ数} \end{aligned}$$

ここで、P E マトリックス 1 0 1 内で処理中の入力データ数とは、P E マトリックスコア 1 2 0 内で処理中の入力データ数のことである。即ち、P E マトリックス 1 0 1 内で処理中の入力データ数と、ロードバッファ 1 1 0 とストアバッファ 1 1 1 で処理中の入力データ数とを合計したものである。ストアバッファ 1 1 1 は、少なくとも P E マトリックス 1 0 1 でエラーが発生してから P E マトリックスコア 1 2 0 が停止するまでの間、P E マトリックス 1 0 1 から出力された全てのデータを保持可能な容量を有する。

【 0 0 4 2 】

次に、P E マトリックス 1 0 1 の全プロセッシングエレメント内で処理中の入力データ数を算出する方法を図 3 乃至図 5 を用いて説明する。

【 0 0 4 3 】

図 3 は、P E マトリックス内データ数計算例で用いた配線情報を示す図である。図 4 は、P E マトリックス内データ数計算例で最初の入力状態の入力データ数を示す図である。図 5 は、P E マトリックス内データ数計算例で図 4 の状態から 4 クロックサイクル後の入力データ数を示す図である。

【 0 0 4 4 】

図 3 乃至図 5 において、図 3 は、4 × 4 プロセッシングエレメントの接続関係を示したものである。3 0 1 は、プロセッシングエレメントである。3 0 4 は、プロセッシングエレメント間の配線である。接続関係がないプロセッシングエレメント間には配線 3 0 4 を図示していない。プロセッシングエレメントの入力データ 3 0 2 は、ロードバッファ 1 1 0 から入力され、プロセッシングエレメントの出力データ 3 0 3 は、ストアバッファ 1 1 1 へ出力される。また、入力データ数信号 3 0 5 は、入力データ値バッファ 1 1 3 へ出力される。

【 0 0 4 5 】

図 4、図 5 は、図 3 のプロセッシングエレメント構成を用いた場合の各プロセッシングエレメントで処理に使用した入力データ数のクロックによる違いを示したものである。図 4 は、ロードバッファ 1 1 0 から出力された入力データ 3 0 2 が最初に P E マトリックス 1 0 1 内に入力された時の、各プロセッシングエレメント内で使用した入力データ数を示している。

【 0 0 4 6 】

プロセッシングエレメント内に記述された値は、各プロセッシングエレメント内で使用した入力データ数を示す。ロードバッファ 1 1 0 からの入力が 2 系統あるプロセッシングエレメントの入力データ数は“ 2 ”となり、ロードバッファ 1 1 0 からの入力が 1 系統の

10

20

30

40

50

プロセッシングエレメントの入力データ数は “ 1 ” となる。

【 0 0 4 7 】

図 5 は、図 4 の状態から 4 クロックサイクル後の各プロセッシングエレメントで処理に使用した入力データ数を示している。プロセッシングエレメント内に記述された値は、各プロセッシングエレメント内で使用した入力データ数を示す。その値は、接続関係のあるプロセッシングエレメント内の入力データ値を加算した値となる。また、PE マトリックス 1 0 1 からは、出力データ 3 0 3 及び入力データ数信号 3 0 5 の両方が出力され、それぞれストアバッファ 1 1 1 及び入力データ値バッファ 1 1 3 へ送出される。ここで、入力データは、ロードバッファ 1 1 0 から毎サイクル入力されているものとする。

【 0 0 4 8 】

図 5 の時点で、全プロセッシングエレメントで処理中の入力データ数は、各プロセッシングエレメント内の入力データ数の合計 2 6 であることが分かる。従って、加算器 1 1 4 (図 9) により各プロセッシングエレメントについて入力データ数の加算を行い合計を算出することで、全プロセッシングエレメント内で現在処理中の入力データ数を算出することができる。更に、加算器 1 1 4 によりロードバッファ 1 1 0 内のデータ数及びストアバッファ 1 1 1 内のデータ数に対応する入力データ数を加算することで、PE マトリックス内で処理中の全入力データ数を算出することができる。

【 0 0 4 9 】

図 6 は、ストアバッファ 1 1 1 及び入力データ値バッファ 1 1 3 の概略構成を示す図である。

【 0 0 5 0 】

図 6 において、ストアバッファ 1 1 1 及び入力データ値バッファ (入力データ数バッファ部) 1 1 3 の 2 つのブロックと信号の入出力関係は、ストアバッファ 1 1 1 内のデータに対応する入力データ数を算出するための方法を示している。データ信号 3 0 3 は、PE マトリックス 1 0 1 から出力された信号である。ストアバッファ 1 1 1 は、出力データ 3 0 3 を出力データ 6 0 3 としてバッファリングする。データ信号 6 0 4 は、外部メモリ 1 4 5 へ出力する信号である。

【 0 0 5 1 】

信号 3 0 5 は、PE マトリックス 1 0 1 から出力された入力データ数を示す信号である。入力データ数 6 1 3 は、出力データ 6 0 3 に対応する。入力データ合計数 6 1 2 は、入力データ値バッファ 1 1 3 内の全入力データ数を合計した値であり、入力データ合計値制御部 1 1 2 へ出力する信号である。入力データ合計数 6 1 2 は、出力データ 6 0 3 に応じて増減する。

【 0 0 5 2 】

PE マトリックス 1 0 1 から出力された出力データがストアバッファ 1 1 1 の出力データ 6 0 3 としてセットされる毎に、入力データ値バッファ 1 1 3 に対応する入力データ数がセットされる。ストアバッファ 1 1 1 の出力データ 6 0 3 が出力される毎に、入力データ値バッファ 1 1 3 から対応する入力データ数が削除される。従って、入力データ合計数 6 1 2 の値は、ストアバッファ 1 1 1 の出力データ 6 0 3 が使用した入力データ数の合計となる。

【 0 0 5 3 】

また、アルゴリズムによっては、プロセッシングエレメントの出力データと対応する入力データ数の比が一定の場合がある。そのような場合は、ストアバッファ 1 1 1 内の出力データ数を整数倍して入力データ数を求めても良いことは明らかである。

【 0 0 5 4 】

図 7 は、入力データ合計値制御部 1 1 2 の構成を示すブロック図である。

【 0 0 5 5 】

図 7 において、入力データ合計値制御部 1 1 2 は、上記図 1 で説明したように、PE マトリックス 1 0 1 内の全入力データ数を合計する。信号 6 1 2 は、入力データ値バッファ 1 1 3 から出力された信号である。信号 1 3 7 は、入力データ数 1 3 7 (図 9) の信号に

10

20

30

40

50

対応しており、全プロセッシングエレメントからの出力信号を示している。入力データ数 704 は、ロードバッファ 110 内の入力データ数である。入力データ合計値制御部 112 は、複数の加算器 705 及び加算器 706 により全ての入力信号の加算を行う。加算結果は、信号 707 としてプロセッサ 107 へ出力され、再ロードアドレス計算に使用される。

【0056】

次に、本実施の形態の動的再構成可能デバイス故障検出装置内部の各部の動作を図 8 及び図 10 乃至図 13 を参照しながら説明する。

【0057】

図 8 は、動的再構成可能デバイス故障検出装置内部の各部間のデータ処理タイミング及びデータ処理の流れを示す図である。図 10 は、エラー情報保持部 105 の処理を示すフローチャートである。図 11 は、プロセッサ 107 の処理を示すフローチャートである。図 12 は、PE マトリックスリセット制御部 121 の処理を示すフローチャートである。図 13 は、PE マトリックスコア 120 の処理を示すフローチャートである。

【0058】

図 8 において、上方から下方に向かい時間の経過を示している。各部（エラー検出回路 103、エラー情報保持部 105、PE マトリックスコア 120、入力データ合計値制御部 112、PE マトリックスリセット制御部 121、プロセッサ 107、外部メモリ 145）の間の矢印は各部間の信号を示している。また、矢印は信号の向かう方向を示している。

【0059】

図 8 及び図 10 乃至図 13 において、エラー検出回路 103 は、プロセッシングエレメントのエラーを検出した場合、エラー信号 810 をエラー情報保持部 105 へ出力し、エラー情報を伝達する。これに伴い、エラー情報保持部 105 は、エラー情報を保持する（ステップ S1）。また、エラー情報保持部 105 は、割り込み信号 812 をプロセッサ 107 へ出力し、エラー情報を通知する（ステップ S2）。また、エラー情報保持部 105 は、PE マトリックスコア 120 の処理を停止するために、機能停止信号 811 を PE マトリックスコア 120 へ出力し、PE マトリックスコア 120 の処理を停止する（ステップ S3）。

【0060】

プロセッサ 107 は、エラー情報保持部 105 から割り込み信号 812 を受信した後、エラー情報 814 を取得する（ステップ S11）。プロセッサ 107 は、エラー情報 814 を取得した後、割り込みマスク信号 840 をエラー情報保持部 105 へ出力し、割り込み信号 812 をマスクする（ステップ S12）。プロセッサ 107 は、割り込み信号 812 をマスクした後、外部メモリ 145 から構成情報 816（図 1 の 104）を取得する。更に、プロセッサ 107 は、上記エラー情報 814 を基に、故障したプロセッシングエレメントを使用しないようにするため、構成情報変更処理 817 により外部メモリ 145 の構成情報を新しい構成情報に書き換える（ステップ S13）。

【0061】

その後、プロセッサ 107 は、取得処理 841 により入力データ合計値制御部 112 から PE マトリックス内の入力データ数を取得し、アドレス計算処理 818 により PE マトリックス用入力データの再ロードアドレスを計算する（ステップ S14）。プロセッサ 107 は、アドレス計算処理 818 が終了した後、リセット信号（PE マトリックスデータクリア指示）819 を PE マトリックスリセット制御部 121 へ出力する（ステップ S15）。

【0062】

PE マトリックスリセット制御部 121 は、プロセッサ 107 からリセット信号 819 を受信した後（ステップ S21）、リセット信号 842 を PE マトリックスコア 120 へ出力し、PE マトリックスコア 120 内の全データを無効化する（ステップ S22）。この場合、PE マトリックスリセット制御部 121 により無効化されたデータは、入力デー

10

20

30

40

50

タ合計値制御部 112 により計数する。プロセッサ 107 は、PE マトリックス 101 のリセット後、PE マトリックス機能停止解除信号 820 をエラー情報保持部 105 に出力する(ステップ S16)。

【0063】

エラー情報保持部 105 は、PE マトリックス機能停止解除信号 820 を受信した後、デアサート処理 811 により機能停止信号をデアサートし、PE マトリックス 101 を処理可能状態にする(ステップ S4)。その後、PE マトリックスコア 120 は、外部メモリ 145 から構成情報 843 を PE マトリックスコア 120 にロードし、新しい構成情報に変更する(ステップ S31)。更に、PE マトリックスコア 120 は、処理 821 により入力データを PE マトリックスコア 120 に再ロードする(ステップ S32)。これにより、プロセッシングエレメントが実行中に故障した場合も、LSI を交換することなく正常に処理を再開することができる。

10

【0064】

以上説明したように、本実施の形態によれば、LSI 製造後のアプリケーション実行中に動的再構成可能デバイス内のプロセッシングエレメントでエラーが発生した場合、エラー情報保持部 105 により PE マトリックスコア 120 の処理を停止する。

【0065】

更に、プロセッサ 107 によりエラーが発生したプロセッシングエレメントから外部メモリ 145 へエラーデータが出力されないように制御し、PE マトリックスリセット制御部 121 により PE マトリックスコア 120 内の全データを無効化する。

20

【0066】

更に、エラーが発生したプロセッシングエレメントを使用しないようにするため、プロセッサ 107 により構成情報を変更した後、無効化したデータに相当する入力データを外部メモリ 145 から再ロードして PE マトリックスコア 120 の処理を再実行する。

【0067】

これにより、LSI 製造後のアプリケーションの実行中に PE マトリックス 101 内のプロセッシングエレメントでエラーが発生した場合、プロセッシングエレメントのエラーを即座に検出することが可能となる。また、プロセッシングエレメントのエラー検出後も、PE マトリックス 101 で処理を継続することが可能となる。

【0068】

30

[他の実施の形態]

また、本発明の目的は、前述した各実施の形態の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ(または CPU や MPU 等)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても達成される。

【0069】

この場合、記憶媒体から読み出されたプログラムコード自体が前述した各実施の形態の機能を実現することになり、そのプログラムコード及び該プログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0070】

40

また、プログラムコードを供給するための記憶媒体としては、例えば、フロッピー(登録商標)ディスク、ハードディスク、光ディスク、磁気テープ、不揮発性のメモリカード、ROM 等を用いることができる。または、プログラムコードをネットワークを介してダウンロードしてもよい。

【0071】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した各実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動している OS (オペレーティングシステム) 等が実際の処理の一部または全部を行い、その処理によって前述した各実施の形態の機能が実現される場合も含まれる。

50

【 0 0 7 2 】

更に、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、次のプログラムコードの指示に基づき、その拡張機能を拡張ボードや拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって前述した各実施の形態の機能が実現される場合も含まれる。

【 図面の簡単な説明 】

【 0 0 7 3 】

【 図 1 】 本発明の実施の形態に係るデバイス故障検出装置としての動的再構成可能デバイス故障検出装置の構成を示すブロック図である。

10

【 図 2 】 動的再構成可能デバイス故障検出装置のエラー情報保持部の構成を示すブロック図である。

【 図 3 】 PEマトリックス内データ数計算例で用いた配線情報を示す図である。

【 図 4 】 PEマトリックス内データ数計算例で最初の入力状態の入力データ数を示す図である。

【 図 5 】 PEマトリックス内データ数計算例で図 4 の状態から 4 クロックサイクル後の入力データ数を示す図である。

【 図 6 】 ストアバッファ及び入力データ値バッファの概略構成を示す図である。

【 図 7 】 入力データ合計値制御部の構成を示すブロック図である。

【 図 8 】 動的再構成可能デバイス故障検出装置内部の各部間のデータ処理タイミング及びデータ処理の流れを示す図である。

20

【 図 9 】 PEマトリックス内部の最小構成要素であるPEの構成を示すブロック図である。

【 図 1 0 】 エラー情報保持部の処理を示すフローチャートである。

【 図 1 1 】 プロセッサの処理を示すフローチャートである。

【 図 1 2 】 PEマトリックスリセット制御部の処理を示すフローチャートである。

【 図 1 3 】 PEマトリックスコアの処理を示すフローチャートである。

【 符号の説明 】

【 0 0 7 4 】

1 0 1 PEマトリックス (ダイナミックリコンフィギャラブルデバイス)

30

1 0 2 ALU (演算器)

1 0 3 エラー検出回路 (検出手段)

1 0 4 構成情報

1 0 5 エラー情報保持部

1 0 7 プロセッサ (計算手段)

1 1 0 ロードバッファ

1 1 1 ストアバッファ

1 1 2 入力データ合計値制御部

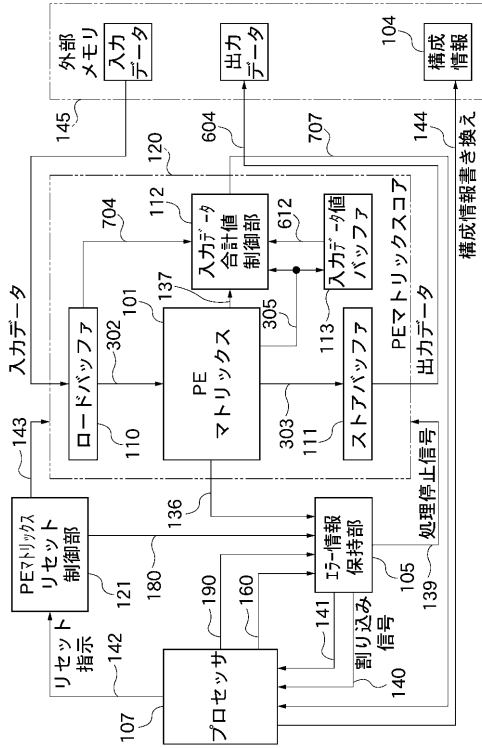
1 2 0 PEマトリックスコア

1 2 1 PEマトリックスリセット制御部

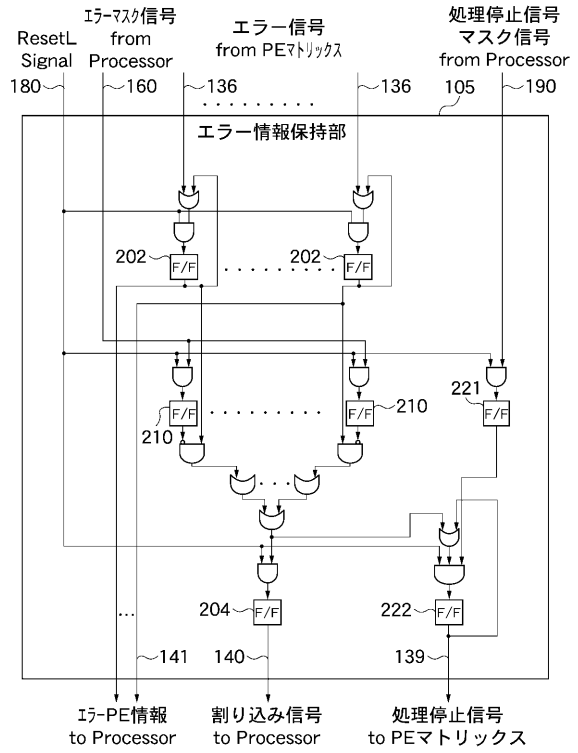
40

1 4 5 外部メモリ (記憶手段)

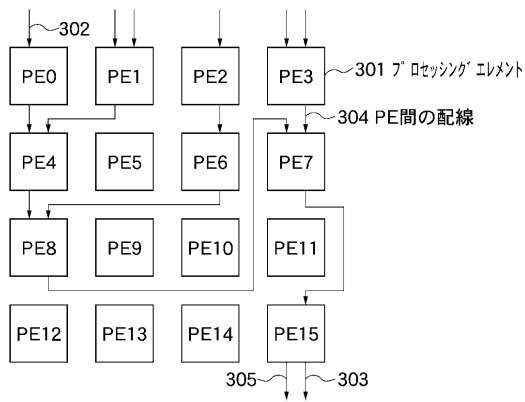
【図1】



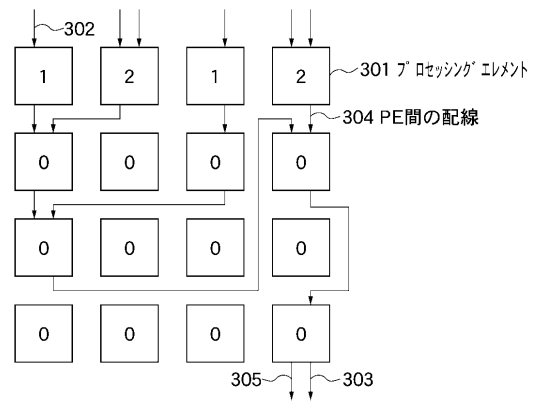
【図2】



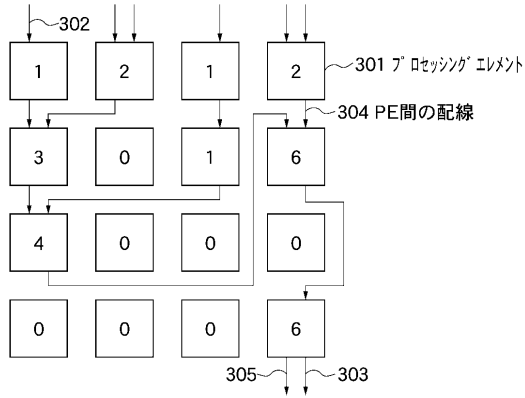
【図3】



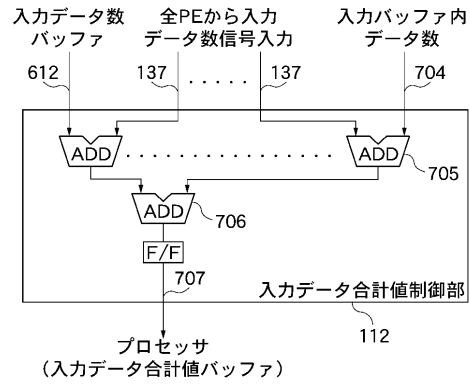
【図4】



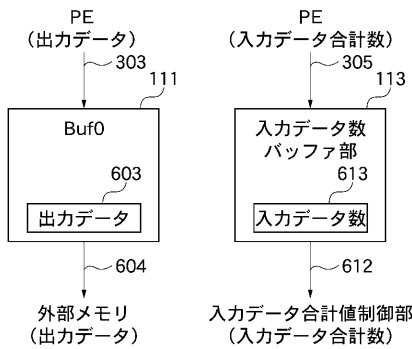
【図5】



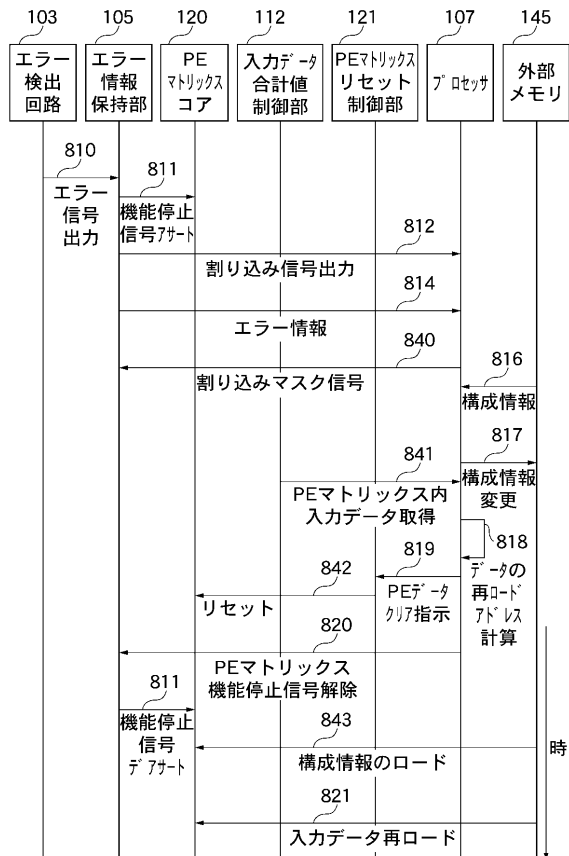
【図7】



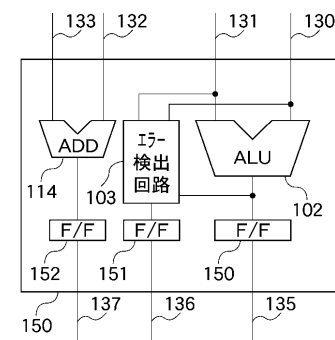
【図6】



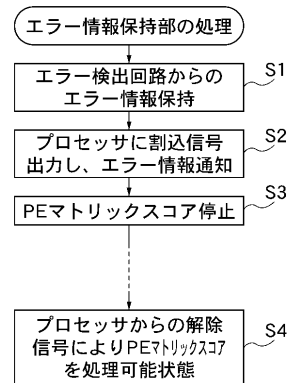
【図8】



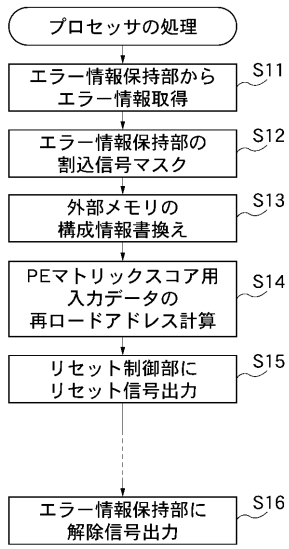
【図9】



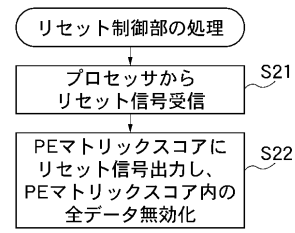
【図10】



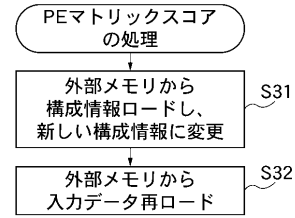
【図 1 1】



【図 1 2】



【図 1 3】



フロントページの続き

- (56)参考文献 特開平10-222389(JP,A)
特開昭57-143647(JP,A)
特開2000-311156(JP,A)
特開2000-81991(JP,A)
特開平8-44581(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 15/80
G06F 15/16 - 15/177
G06F 11/20