

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4742281号
(P4742281)

(45) 発行日 平成23年8月10日(2011.8.10)

(24) 登録日 平成23年5月20日(2011.5.20)

(51) Int.Cl. F I
G06F 7/00 (2006.01) G O 6 F 7/00 2 O 4
H03K 19/173 (2006.01) H O 3 K 19/173 1 O 1

請求項の数 11 (全 40 頁)

(21) 出願番号 特願2008-508466 (P2008-508466)
 (86) (22) 出願日 平成19年3月2日(2007.3.2)
 (86) 国際出願番号 PCT/JP2007/054100
 (87) 国際公開番号 W02007/113964
 (87) 国際公開日 平成19年10月11日(2007.10.11)
 審査請求日 平成21年7月3日(2009.7.3)
 (31) 優先権主張番号 特願2006-101107 (P2006-101107)
 (32) 優先日 平成18年3月31日(2006.3.31)
 (33) 優先権主張国 日本国(JP)

(73) 特許権者 504174135
 国立大学法人九州工業大学
 福岡県北九州市戸畑区仙水町1番1号
 (74) 代理人 100121371
 弁理士 石田 和人
 (72) 発明者 笹尾 勤
 福岡県飯塚市川津680-4 九州工業大
 学情報工学部内
 審査官 緑川 隆

最終頁に続く

(54) 【発明の名称】 多段論理回路の再構成装置及び再構成方法、論理回路修正装置、並びに再構成可能な多段論理回路

(57) 【特許請求の範囲】

【請求項1】

入力変数をXとする目的論理関数F(X)を関数分解して得られる部分関数のLUTが記憶された複数のpq素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、入力ベクトルbに対する前記目的論理関数F(X)の出力ベクトルF(b)を無効値に変更する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成装置であって、

未修正の前記pq素子のうち出力側に最も近いpq素子E_Gから順次選択する素子選択手段と、

前記pq素子E_GのLUTの出力ベクトルのうち前記入力ベクトルbに対する出力ベクトルをcとしたとき、当該出力ベクトルcに対応する入力ベクトルが、前記入力ベクトルb以外にも存在するか否かを検査する対応検査手段と、

前記pq素子E_Gを修正済みとするとともに、当該pq素子E_GのLUTにおいて、前記入力ベクトルbが前記出力ベクトルcに一対一に対応する場合、前記入力ベクトルbに対する出力ベクトルcを無効値に書き換える削除修正手段と、を備えたことを特徴とする多段論理回路の再構成装置。

【請求項2】

入力変数をXとする目的論理関数F(X)を関数分解して得られる部分関数のLUTが記憶された複数のpq素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、前記目的論理関数F(X)の出力ベクトル集合に入力ベクトルbに

10

20

に対する出力ベクトル a を追加する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成装置であって、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応付手段と、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする追加修正手段と、を備えていることを特徴とする多段論理回路の再構成装置。

10

【請求項3】

入力変数 X の目的論理関数 $Q(X)$ を演算する主論理回路について、前記入力変数 X として入力される各入力ベクトル b のうち特定の対象入力ベクトル b_i に対する主論理回路の出力ベクトル $Q(b_i)$ を、修正出力ベクトル $Q'(b_i)$ に変更する論理回路修正装置であって、

前記各対象入力ベクトル b_i に対応して、前記各出力ベクトル $Q(b_i)$ を修正出力ベクトル $Q'(b_i)$ に修正するための修正用ベクトル P_i が所定のアドレス A_i に登録される補助メモリと

前記補助メモリが出力する修正用ベクトル P_i を出力した場合、当該修正用ベクトル P_i 及び前記主論理回路が出力する出力ベクトル $Q(b_i)$ に基づいて、前記修正出力ベクトル $Q'(b_i)$ を出力する修正手段と、

20

前記入力変数 X に対して、当該入力変数 X の値が前記対象入力ベクトル b_i に等しい場合は前記修正用ベクトル P_i が格納された前記補助メモリのアドレス A_i を出力するアドレス生成関数 $F(X)$ の演算を行うアドレス生成回路と、を備え、

前記アドレス生成回路は、前記アドレス生成関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路により構成されており、

前記修正用ベクトル P_i は、対象入力ベクトル b_i に対する主論理回路の出力ベクトル $Q(b_i)$ との排他論理和が、前記修正出力ベクトル $Q'(b_i)$ となる値に設定され、

30

前記補助メモリは、前記アドレス生成回路が出力するアドレス A_i が入力されると、前記修正手段に前記修正用ベクトル P_i を出力し、それ以外の場合は0を出力するものであり、

前記修正手段は、前記補助メモリの出力値と前記主論理回路との排他論理和演算を行う $EXOR$ ゲートであることを特徴とする論理回路修正装置。

【請求項4】

入力変数 X の目的論理関数 $Q(X)$ を演算する主論理回路について、前記入力変数 X として入力される各入力ベクトル b のうち特定の対象入力ベクトル b_i に対する主論理回路の出力ベクトル $Q(b_i)$ を、修正出力ベクトル $Q'(b_i)$ に変更する論理回路修正装置であって、

前記各対象入力ベクトル b_i に対応して、前記各出力ベクトル $Q(b_i)$ を修正出力ベクトル $Q'(b_i)$ に修正するための修正用ベクトル P_i が所定のアドレス A_i に登録される補助メモリと

40

前記補助メモリが出力する修正用ベクトル P_i を出力した場合、当該修正用ベクトル P_i 及び前記主論理回路が出力する出力ベクトル $Q(b_i)$ に基づいて、前記修正出力ベクトル $Q'(b_i)$ を出力する修正手段と、

前記入力変数 X に対して、当該入力変数 X の値が前記対象入力ベクトル b_i に等しい場合は前記修正用ベクトル P_i が格納された前記補助メモリのアドレス A_i を出力するアドレス生成関数 $F(X)$ の演算を行うアドレス生成回路と、を備え、

前記アドレス生成回路は、前記アドレス生成関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路により構成されており、前記入力変数 X に対して、当該入

50

力変数 X の値がいずれの前記対象入力ベクトル b_i とも等しくない場合には、無効値を出力するものであり、

前記補助メモリは、前記アドレス生成回路が出力するアドレス A_i が入力されると、前記修正手段に前記修正用ベクトル P_i を出力し

前記修正手段は、前記主論理回路及び前記補助メモリの出力段にそれぞれ設けられたトライ・ステート・バッファであり、

前記主論理回路の出力段の前記トライ・ステート・バッファは、前記アドレス生成回路の出力値が無効値でない場合にはハイ・インピーダンス、それ以外の場合にはロー・インピーダンス状態となり、

前記補助メモリの出力段の前記トライ・ステート・バッファは、前記アドレス生成回路の出力値が無効値の場合にはハイ・インピーダンス、それ以外の場合にはロー・インピーダンス状態となることを特徴とする論理回路修正装置。

【請求項 5】

前記補助メモリは、前記アドレス生成回路の最終段の p q 素子であることを特徴とする請求項 3 又は 4 記載の論理回路修正装置。

【請求項 6】

前記アドレス生成回路において、入力ベクトル b に対する前記アドレス生成関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記アドレス生成回路の再構成を行う再構成装置を備え、

前記再構成装置は、

未修正の前記 p q 素子のうち出力側に最も近い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトルを c としたとき、当該出力ベクトル c に対応する入力ベクトルが、前記入力ベクトル b 以外にも存在するか否かを検査する対応検査手段と、

前記 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の LUT において、前記入力ベクトル b が前記出力ベクトル c に一対一に対応する場合、前記入力ベクトル b に対する出力ベクトル c を無効値に書き換える削除修正手段と、を備えたことを特徴とする請求項 3 乃至 5 の何れか一記載の論理回路修正装置。

【請求項 7】

前記アドレス生成回路において、前記アドレス生成関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、前記アドレス生成回路の再構成を行う再構成装置を備え、

前記再構成装置は、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応付手段と、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする追加修正手段と、を備えていることを特徴とする請求項 3 乃至 5 の何れか一記載の論理回路修正装置。

【請求項 8】

入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、入力ベクトル b に対する前記目的論理関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成方法であって、

10

20

30

40

50

未修正の前記 p q 素子のうち出力側に最も近い p q 素子 E_G から順次選択する素子選択ステップと、

前記 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトルを c としたとき、当該出力ベクトル c に対応する入力ベクトルが、前記入力ベクトル b 以外にも存在するか否かを検査する対応検査ステップと、

前記 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の LUT において、前記入力ベクトル b が前記出力ベクトル c に一対一に対応する場合、前記入力ベクトル b に対する出力ベクトル c を無効値に書き換える削除修正ステップと、を繰り返し実行することを特徴とする多段論理回路の再構成方法。

【請求項 9】

入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、前記目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成方法であって、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択ステップと、

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応付ステップと、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする追加修正ステップと、を繰り返し実行することを特徴とする多段論理回路の再構成方法。

【請求項 10】

入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された再構成可能な多段論理回路であって、

入力ベクトル b に対する前記目的論理関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記多段論理回路の再構成を行う再構成回路を備え、

前記再構成回路は、

未修正の前記 p q 素子のうち出力側に最も近い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトルを c としたとき、当該出力ベクトル c に対応する入力ベクトルが、前記入力ベクトル b 以外にも存在するか否かを検査する対応検査手段と、

前記 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の LUT において、前記入力ベクトル b が前記出力ベクトル c に一対一に対応する場合、前記入力ベクトル b に対する出力ベクトル c を無効値に書き換える削除修正手段と、を備えたことを特徴とする再構成可能な多段論理回路。

【請求項 11】

入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された再構成可能な多段論理回路であって、

前記目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、前記多段論理回路の再構成を行う再構成回路を備え、

前記再構成回路は、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択手段と、

10

20

30

40

50

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の L U T の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応手段と、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G の L U T の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする追加修正手段と、
を備えていることを特徴とする再構成可能な多段論理回路。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、論理関数の関数分解を繰り返して論理設計される多段論理回路の再構成を行う再構成装置と、それを用いた論理回路修正装置に関する。

【背景技術】

【0002】

一般に、論理回路は専用の L S I (A S I C : Application Specific Integrated Circuit) を用いて構成されることが多い。しかしながら、A S I C の開発コストは高価であり、その修正や費用にも時間がかかる。一方、論理構成の容易な F P G A (Field Programmable Gate Array) も存在するが、現状では消費電力や性能の点で問題がある。

【0003】

20

そこで、通常、A S I C は、修正を可能とする余分な論理回路をあらかじめ組み込んだ状態に設計される。これにより、軽微な機能変更にある程度柔軟に対応させることが可能となる。

【0004】

軽微な機能変更を行う論理回路修正装置としては、例えば、特許文献 1 に記載のものが公知である。特許文献 1 に記載の論理回路修正装置は、C P U で実行するプログラムが記憶された R O M の出力値を修正するためのものであり、F P G A を用いて構成されている。

【0005】

図 20 は、特許文献 1 に記載の論理回路修正装置 103 の構成を示す図である。R O M 101 には、C P U 102 で実行するプログラムが格納されている。C P U 102 は、R O M 101 に対してアドレスバス 104 を介してアドレスを送る。R O M 101 は、当該アドレスに対し、第 1 のデータバス 106 にデータを出力する。

30

【0006】

論理回路修正装置 103 には、アドレスバス 104 からアドレスが入力され、それに対するデータが第 1 のデータバス 106 から入力される。論理回路修正装置 103 内の修正アドレス記憶部 111 には、修正を行うべきデータが格納された R O M 101 のアドレスが登録されている。比較回路 113 は、修正アドレス記憶部 111 に記憶されたアドレス値にアドレスバス 104 から入力されるアドレス値に一致するものがあるか否かを判定し、一致するものがある場合は一致信号を、一致するものがない場合には不一致信号を出力する。一致信号又は不一致信号は、データ選択回路 114 に出力される。

40

【0007】

一方、修正データ格納部 112 には、修正アドレス記憶部 111 に登録された各アドレス値に対応して、修正を行うべきデータが登録されている。データ選択回路 114 は、比較回路 113 から一致信号が入力された場合には、修正データ格納部 112 から読み出したデータを第 2 のデータバス 105 に出力する。一方、不一致信号が入力された場合には、データ選択回路 114 は、第 1 のデータバス 106 から入力されるデータを第 2 のデータバス 105 に出力する。第 2 のデータバス 105 に出力されたデータは、読み出しデータとして C P U 102 に入力される。

【0008】

50

このようにして、修正したいROM101のアドレスを修正アドレス記憶部111に登録するとともに、修正したいデータを修正データ格納部112に登録することによって、ROM101の内容の軽微な修正が行われる。

【特許文献1】特開2002-297408号公報

【特許文献2】特開2004-258799号公報

【非特許文献1】T. Sasao and M. Matsuura, "BDD representation for incompletely specified multiple-output logic functions and its applications to functional decomposition," Design Automation Conference, Anaheim, CA, June 13-17, 2005, pp.373-378.

【発明の開示】

【発明が解決しようとする課題】

【0009】

ところで、設計段階においては、もとの論理回路のどの部分の修正が必要となるのかは分からない。従って、論理回路修正装置は、もとの論理回路の任意の出力に対して修正が可能であることが求められる。そのためには、回路の論理構成を自在に変更できることが必要である。

【0010】

一方で、論理回路修正装置は、論理回路の修正がない限り、本来の論理回路の動作には関係しない余分な回路である。従って、論理回路修正装置は、その実装面積ができる限り小さく、その消費電力も可能な限り小さいことが好ましい。

【0011】

上記従来の論理回路修正装置は、回路に汎用性を持たせるためにFPGAを用いて構成されている。そのため、広い配線領域が必要とされ、論理回路修正装置の実装に比較的大きなチップ面積が必要であるという問題がある。

【0012】

また、FPGAの再構成を行う場合、一般に、配線の変更がなされる。従って、配線遅延の影響なども考慮に入れた変更が必要とされる。また、変更を誤ると、最悪の場合には回路を破損するおそれがあり、回路変更には十分な注意が必要とされる。更に、配線の変更を伴うことから、回路の変更は、論理回路の動作の停止中に行う必要がある。従って、論理回路の動作期間中にリアルタイムで動的な変更が要求されるような用途（例えば、辞書やニューラル・ネットワーク等における学習に伴う論理回路構成の変更。）に適用することは困難である。

【0013】

ところで、論理構成の再構成が可能な論理回路として、LUTカスケード論理回路が知られている（特許文献2，非特許文献1参照）。LUTカスケード論理回路は、二分決定グラフ（BDD：binary decision diagram）を用いて目的論理関数を入出力関係が直列状態となる複数の部分関数に関数分解し、各部分関数をLUTとしてメモリで実現したものである。LUTカスケード論理回路は、殆どがメモリにより構成されるため、高集積化が容易であり実装面積を小さくできる。

【0014】

しかしながら、LUTカスケード論理回路の再構成を行う場合、従来は、多くの計算処理が必要とされていた。再構成を行う場合、まず、目的論理関数のBDDを作成し、BDDの分割と再構成を繰り返すことによって最適な関数分解方法を探査する。そして、その結果得られる部分関数をLUT化してメモリに格納する。目的論理関数の入力変数が多い場合、この処理には大容量メモリを備えたコンピュータで長時間かけて行う必要がある。そのため、簡易な書き換えが必要とされる論理回路修正装置にLUTカスケード論理回路を適用することは、従来行われていない。

【0015】

しかしながら、LUTカスケード論理回路の論理構成の再構築を簡易化し、軽微な論理の修正を容易に行う手段があれば、論理回路修正装置をLUTカスケード論理回路に適用

10

20

30

40

50

すれば、小実装面積化と低消費電力化のアプローチと成り得ると考えられる。

【0016】

そこで、本発明の目的は、上記LUTカスケード論理回路と類似のLUT型の再構成可能な論理回路を用いて、回路の論理構成の変更の自由度を維持しつつ、従来に比べて実装面積を縮小することができ、消費電力も小さい論理回路修正装置を提供することにある。また、論理回路の動作期間中にリアルタイムで動的な修正を可能とする論理回路修正装置を提供することにある。

【0017】

また、本発明の他の目的は、上記論理回路修正装置などにおいて使用される、論理変更が可能で且つ小実装面積・低消費電力の多段論理回路の再構成を簡易に行うことが可能な多段論理回路の再構成装置を提供することにある。

10

【0018】

また、本発明の他の目的は、論理関数の軽微な再構成を容易に行うことができ、論理回路の動作期間中にリアルタイムで動的な再構成を実行することを可能とする再構成可能多段論理回路を提供することも目的としている。

【課題を解決するための手段】

【0019】

〔1〕用語の説明及び本発明の原理的背景

まず、本明細書において使用される主な用語の定義を行い、本発明に関するいくつかの理論的背景について説明する。

20

【0020】

〔定義1〕（アドレス生成関数，アドレス生成論理関数）

関数 $F(X):B^n \rightarrow \{0,1,\dots,k\}$ (X は n 次元ベクトル, $B=\{0,1\}$) において k 個の異なる登録ベクトル (registered vector) $a_i \in B^n$ ($i=1,2,\dots,k$) に対して、 $F(a_i)=i$ ($i=1,2,\dots,k$) が成立し、それ以外の (2^n-k) 個の入力ベクトルに対しては、 $F=0$ が成立するとき、 $F(X)$ を重み k のアドレス生成関数(address generation function)という。アドレス生成関数は、 k 個の異なる2値ベクトルに対して、1から k までの固有アドレスを生成する。アドレス生成関数の出力値を2進数で表現する多出力論理関数をアドレス生成論理関数(address generation logic function)という。

(定義終了)

30

本明細書においては、 k の値は入力ベクトルの組み合わせ総数 2^n に比べて十分に小さい($k \ll 2^n$)と仮定する。

【0021】

〔定義2〕（アドレス検出関数）

関数 $f(X):B^n \rightarrow \{0,1,\dots,k\}$ (X は n 次元ベクトル) において k 個の異なる登録ベクトル $a_i \in B^n$ ($i=1,2,\dots,k$) に対して $f(a_i)=1$ ($i=1,2,\dots,k$) が成立し、それ以外の (2^n-k) 個の入力ベクトルに対しては、 $f=0$ が成立するとき、 $f(X)$ を重み k のアドレス検出関数 (address detection function) という。

(定義終了)

40

【0022】

〔定義3〕（アドレス表，アドレス生成回路）

アドレス生成論理関数 $F(X):B^n \rightarrow \{0,1,\dots,k\}$ の k 個の登録ベクトル $\{a_i | i=1,2,\dots,k; a_i \in B^n\}$ に対し、出力ベクトル $F(a_i) \in B^m$ を対応させた表をアドレス表 (address table) という。アドレス表を実現する回路をアドレス生成回路 (address generation circuit) という。

(定義終了)

【0023】

〔定義4〕（分割）

入力ベクトルを $X = (x_1, x_2, \dots, x_n)$ とする。 X の変数の集合を $\{X\}$ で表す。 $\{X_1\} \cup \{X_2\} = \{X\}$ 且つ $\{X_1\} \cap \{X_2\} = \emptyset$ のとき、 $X = (X_1, X_2)$

50

を X の分割 (partition) という。ここで、 \emptyset は空集合を表す。

(定義終り)

【0024】

〔定義5〕(分解表, 基本分解表, 列複雑度)

完全定義関数 $f(X) : B^n \rightarrow B^q$ ($B = \{0, 1\}$, $X = (x_1, x_2, \dots, x_n)$, n, q 自然数) が与えられているとする。 (X_L, X_H) を X の分割とする。X の次元 (変数の個数) を $d(X)$ と記す。関数 $f(X)$ 及び X の分割 $X = (X_L, X_H)$ に対して、以下の (1) ~ (3) の条件を満たす表を f の分解表 (decomposition chart) という。

(1) 2^{n_L} 列 2^{n_H} 行の表である。ここで、 $n_L = d(X_L)$, $n_H = d(X_H)$ とする。 10

(2) 各行, 各列に 2 進符号のラベルを持ち、列及び行のラベルの集合は、それぞれ n_L, n_H ビットのすべてのパターンを要素とする。

(3) 表の各要素が、その要素に対応する列及び行のラベルの組み合わせ (X_L, X_H) に対する f の真理値 $f(X_L, X_H)$ である。

X_L を束縛変数 (bound variables)、 X_H を自由変数 (free variables) という。分解表の異なる列パターンの個数を分解表の列複雑度 (column multiplicity) といい、 μ と記す。分解表の特別な場合として、 $X_L = X, X_H = \emptyset$ の場合も考える。

また、関数 f の分解表のうちで、 $X_L = (x_1, x_2, \dots, x_{n_L})$ 且つ $X_H = (x_{n_L+1}, x_{n_L+2}, \dots, x_n)$ となるものを基本分解表 (standard decomposition chart) という。 20

(定義終り)

【0025】

(例1)

(表1) の分解表では、 $n_L = 3, n_H = 2$ である。また、分解表の列パターンは、 $(0110)^T$ と $(1101)^T$ の2つなので $\mu = 2$ である。

(例終り)

【0026】

【表1】

表 1: 分解表の例

		X_L									
		0	0	0	0	1	1	1	1	x_3	
		0	0	1	1	0	0	1	1	x_2	
		0	1	0	1	0	1	0	1	x_1	
X_H	0	0	0	1	0	0	0	1	1	0	
	0	1	1	1	1	1	1	1	1	1	
	1	0	1	0	1	1	1	0	0	1	
	1	1	0	1	0	0	0	1	1	0	
	x_5	x_4									

30

40

【0027】

〔補題1〕

重み k のアドレス生成 (検出) 関数の分解表の列複雑度は高々 $k + 1$ である。

(補題終り)

【0028】

〔補題2〕

F がアドレス生成関数のとき、次式 (1) の関係を満たす二つのアドレス生成関数 G と H は存在し、F の重みと G の重みは等しい。

【0029】

50

【数1】

$$F(\mathbf{X}_1, \mathbf{X}_2) = G(H(\mathbf{X}_1), \mathbf{X}_2) \tag{1}$$

(補題終り)

【0030】

(証明)

X_1 を束縛変数、 X_2 を自由変数とする分解表を考える。分解表の各列の値の最大値の順位(最大値が0のときは、順位を0とする。)を $H(X_1)$ の値とするアドレス生成関数 H を考える。このとき、関数 G では、出力値がすべて零の列のみが縮約される。そのため、 G の非零出力の個数は、 F の非零出力の個数と等しい。また、 G もアドレス生成関数になっている。

10

(証明終り)

【0031】

(例2)

(表2)に示す5変数のアドレス生成関数 $F(X)$ の分解表を考える(図1参照)。いま、関数 $F(X)$ を $F(X_1, X_2) = G(H(X_1), X_2)$ と分解する。ここで、 $X_1 = (x_1, x_2, x_3, x_4)$ 、 $X_2 = (x_5)$ である。このとき、(表2)の列複雑度は7である。 H は(表3)に示す4入力3出力関数であり、重み6のアドレス生成論理関数となっている。また、関数 G の分解表を(表4)に示す。このように、重み7のアドレス生成関数 F を分解して得られる関数 G もまた、重み7のアドレス生成関数となる。

20

(例終り)

【0032】

【表2】

表2: 関数 F の分解表

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

30

$x_5 = 0$		1			3		4	5								
$x_5 = 1$			2					6					7			

【0033】

【表 3】

表 3: 関数 H の真理値表.

x_1	x_2	x_3	x_4	y_1	y_2	y_3
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	0	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0

10

20

【 0 0 3 4 】

【表 4】

表 4: 関数 G の分解表

y_1	0	0	0	0	1	1	1	1
y_2	0	0	1	1	0	0	1	1
y_3	0	1	0	1	0	1	0	1

30

$x_5 = 0$		1		3	4	5		
$x_5 = 1$			2			6	7	

【 0 0 3 5 】

〔補題 3〕

f がアドレス検出関数のとき、下式 (2) の関係を満たす、アドレス検出関数 g とアドレス生成関数 H が存在し、 g の重みは f の重みを超えない。

【 0 0 3 6 】

【数 2】

$$f(\mathbf{X}_1, \mathbf{X}_2) = g(H(\mathbf{X}_1), \mathbf{X}_2) \quad (2)$$

40

(補題 終り)

これは、補題 2 と同様に証明できる。アドレス検出関数は単一出力であり、対応するアドレス生成回路よりも複雑になることはない。

【 0 0 3 7 】

〔定義 6〕(p q 素子)

p q 素子 (pq -element) とは、任意の p 入力 q 出力論理関数を実現するメモリであり、そのメモリ量 (memory size) は 2^{pq} である。

50

(定義終了)

【0038】

〔定理1〕

重み k のアドレス生成論理関数は、 p q 素子を下式(3)で表される個数だけ用いて合成可能である。

【0039】

【数3】

$$\left\lceil \frac{n-q}{p-q} \right\rceil, \quad (\text{ここで, } p > q, q = \lceil \log_2(k+1) \rceil) \quad (3)$$

10

(定理終了)

【0040】

(証明)

重み k のアドレス生成論理関数 F は、下式(4)のように関数分解が可能である。

【0041】

【数4】

$$F(X_1, X_2) = G(H(X_1), X_2) \quad (4)$$

このとき、補題1より、 $G(X_1', X_2)$ も重み k のアドレス生成関数であり、入力数は $n - (p - q)$ に減っている。この操作を

20

【0042】

【数5】

$$\left\lceil \frac{n-q}{p-q} \right\rceil \text{回} \quad (5)$$

繰り返すことにより、入力数を q まで削減することが可能である。従って、 p q 素子のみを用いて、アドレス生成回路を構成することが可能である。

(証明終了)

【0043】

(例3)

(表2)に示した5変数のアドレス生成関数 $F(X)$ の非零出力数は、 $k=7$ である。従って、式(6)となり、図1に示すように、4入力3出力素子で実現可能である。

【0044】

【数6】

$$q = \lceil \log_2(k+1) \rceil = \lceil \log_2(7+1) \rceil = 3 \quad (6)$$

(例終了)

【0045】

p q 素子を用いて回路合成をする際、 p を大きくすると、回路を実現するために必要な p q 素子の数は減るが、総メモリ量は増える。一方、 p を小さくすると、総メモリ量は減るが、 p q 素子数は増える。アドレス生成回路を p q 素子で合成する場合、総メモリ量を増やさずに素子数を最小化するような p の決定方法は、次の定理により示される。

40

【0046】

〔定理2〕

アドレス生成論理関数を p q 素子の多段論理回路として実現する場合、総メモリ量の上限は、 $p - q = 1$ 又は $p - q = 2$ のときに最小となる。

(定理終了)

【0047】

50

(証明)

アドレス生成回路を p q 素子を用いて関数分解する場合、1回分解する毎に、回路の入力線を r=p-q本削減することができる。n本の入力線を q本まで削減するためには、

【0048】
【数7】

$$s = \lceil \frac{n-q}{r} \rceil \text{回} \quad (7)$$

の関数分解が必要である。また、回路実現のためには、s個の p q 素子が必要である。従って必要な総メモリ量は、

【0049】
【数8】

$$MEM = s \cdot 2^p q \quad (8)$$

となる。nが十分に大きい場合には、

【0050】
【数9】

$$MEM \simeq \left(\frac{2^r}{r} \right) \cdot (n-q) \cdot 2^p q \quad (9)$$

と近似できる。nとpは与えられた問題により固定されているので、rのみを変化させることができる。2^r/rは、r=1又はr=2のとき最小値をとる。従って、上記定理が証明される。

(証明終り)

【0051】

通常は、回路の段数は少ない方が望ましいので、r=p-q=2として回路を構成する。(定理1)は、関数分解を繰り返すことにより、アドレス生成回路を p q 素子の多段論理回路として合成可能であることを示している。次に示す(例3)は、関数分解の方法によって、LUTカスケード論理回路を含む、種々の多段論理回路を合成することが可能であることを示している。

【0052】
(例4)

入力数n=48、重みk=255のアドレス生成回路を構成する。

【0053】
【数10】

$$q = \lceil \log_2(255+1) \rceil = 8 \quad (10)$$

である。これより、メモリ量を最小にし、段数も減らすpの値はp=10である。p q 素子 1個に対して、入力線数が2減るので、p q 素子を20個用いることで、入力数を8まで削減できる。20個の p q 素子で、例えば、図2のようなアドレス生成回路を実現可能である。この場合、段数は10段、メモリ量は160kbitである。

【0054】

また、入力数の大きい素子を用いることにより、素子数と遅延時間を削減することができる。図3は p = 11, q = 8の例である。この場合、素子数は(48-8)/(11-8)=14、段数は8段、メモリ量は212kbitである。また、図4は、p=12,q=8の例である。この場合、素子数は(48-8)/(12-8)=10、段数は5段、メモリ量は320kbitである。

(例終り)

【0055】

10

20

30

40

50

定理 2 は、一般のアドレス生成関数を実現する際、総メモリ量の上限を最小とする条件を示している。特定のアドレス生成関数に対しては、 $p-q=2$ 以外の場合に、総メモリ量が最小になることもある。

【 0 0 5 6 】

次に、アドレス生成関数のアドレス表を変化させた場合における $p q$ 素子の多段論理回路の再構成方法について説明する。多段論理回路の修正は、

(1) 登録ベクトルの除去

(2) 登録ベクトルの追加

の 2 つの基本操作に分解することができる。そこで、以下は上記 (1) (2) の場合について考察する。

【 0 0 5 7 】

また、上述したように、 $p q$ 素子の多段論理回路は、目的論理関数 F に対して関数分解を繰り返したものである。従って、目的論理関数 F のアドレス表を変化させた場合に、関数分解 $F(X_1, X_2) = G(H(X_1), X_2)$ において、関数 H と関数 G の修正法を示せば十分である。

【 0 0 5 8 】

尚、以下の説明において、「入力変数 (input variable) 」, 「出力変数 (output variable) 」というときは、値がまだ決まっていない場合をいう。また、「入力ベクトル (input vector) 」, 「出力ベクトル (output vector) 」というときは、入力変数, 出力変数の具体的に決まった値をいう。

【 0 0 5 9 】

(1) 登録ベクトルの除去

入力変数 X (B^n) の分割を $X = (X_1, X_2)$ とおく。アドレス生成関数 $F(X)$ の関数分解を $F(X_1, X_2) = G(H(X_1), X_2)$ と記す。ベクトル a (B^n) はアドレス生成関数 $F(X)$ の登録ベクトルであるとする。また、入力変数の分割 $X = (X_1, X_2)$ に対応するベクトル a の分割を $a = (a_1, a_2)$ と記す。

【 0 0 6 0 】

アドレス生成関数 $F(X)$ のアドレス表から登録ベクトル a を除去する場合を考える。尚、以下ではアドレス表が $L U T$ カスケード論理回路で実現されている場合について考える。

【 0 0 6 1 】

アドレス表からベクトル a を除去した場合、 $F(a) = 0$ となる。従って、 $G(H(a_1), a_2) = 0$ である。

【 0 0 6 2 】

一方、部分関数 $G(Y_1, X_2)$ の一方の変数 Y_1 を固定値 $H(a_1)$ として変数 X_2 を変化させた場合、 $X_2 = a_2$ 以外に $G(H(a_1), X_2) = 0$ となる変数 X_2 が存在しない場合、もとのアドレス表においてベクトル a 以外の登録ベクトルの出力ベクトルは、部分関数の値 $H(a_1)$ には依存しない。従って、この場合、 $H(a_1)$ は無効な値であるため $H(a_1) = 0$ としてもよい。

【 0 0 6 3 】

以上より、アドレス表からの登録ベクトル a の除去に伴う $L U T$ カスケード論理回路の修正は、次のようなアルゴリズムに従って行うことができる。

【 0 0 6 4 】

(アルゴリズム 1)

(Step 1) 未修正の $p q$ 素子のうち、最も出力側に近いものを選択する。選択された $p q$ 素子を E_G と記す。 $p q$ 素子 E_G により表される部分関数を $G(H(X_1), X_2)$ とする。ここで、 (X_1, X_2) X かつ $X_1, X_2 =$ である。 (X_1, X_2) に対する入力ベクトルを (a_1, a_2) とする。但し、 $p q$ 素子 E_G が最も出力側から遠い場合は $X_1 =$, $H(X_1) =$ とする。

【 0 0 6 5 】

(Step 2) $p q$ 素子 E_G に記憶された $L U T$ の出力ベクトルのうち、入力ベクトル $(H(a_1), a_2)$ に対応する出力ベクトルを $G(H(a_1), a_2) = 0$ とする。 $p q$ 素子 E_G を修正済とする。

【 0 0 6 6 】

(Step 3) 出力ベクトル $G(H(a_1), X_2)$ が非零となる部分変数 X_2 に対する入力ベクトル

10

20

30

40

50

ル a_0 が存在するか検査する。

【 0 0 6 7 】

(Step 4) a_0 が存在しない場合、Step1に戻る。そうでなければ、処理を終了する。

(アルゴリズム終り)

【 0 0 6 8 】

上記アルゴリズムを一般の p q 素子の多段論理関数の場合に拡張するのは容易である。一般の p q 素子の多段論理関数の場合は次のようになる。

【 0 0 6 9 】

(アルゴリズム 2)

(Step 1) 素子数が s の p q 回路網は、関数分解を $s-1$ 回繰り返し適用して構成できる。各関数分解で、入力側から順に p q 素子を抽出し、各 p q 素子に 1 から s までの番号を付与する。出力の p q 素子の番号は s である。番号 1 の素子 E_1 は、出力から最も離れている。

10

【 0 0 7 0 】

(Step 2) p q 素子の集合を p とし、初期値を $p=\{E_s\}$ とする。

【 0 0 7 1 】

(Step 3) 多段論理回路において、 p の要素とそれ以外の素子で構成された回路の関数分解 $G(H_1(Y_1), H_2(Y_2), \dots, H_m(Y_m))$ を考える。ここで、 Y_i は中間変数で、特別な場合として $H_k(Y_k)=Y_k$ も考える。

【 0 0 7 2 】

20

(Step 4) $G(H_1(a_1), H_2(a_2), \dots, H_2(a_2))=0$ とする。ここで、 $a=(a_1, a_2, \dots, a_m)$ は、中間変数のベクトルである。 $H_i(a_i)=d \neq 0$ のとき、他の登録ベクトル b に対して、 $H_i(a_i)=d$ となるものが存在しないとき、 $H_i(a_i)=0$ とする。

【 0 0 7 3 】

(Step 5) p q 素子の集合 p の要素数が $s-1$ ならば処理を終了する。そうでなければ、 p に素子番号が最も大きい素子を追加して、Step3に戻る。

(アルゴリズム終り)

【 0 0 7 4 】

(2) 登録ベクトルの追加

次に、アドレス生成関数 $F(X)$ のアドレス表に登録ベクトル a を追加する場合を考える。新たに追加する登録ベクトル a に対する出力ベクトルを c とする。尚、以下ではアドレス表が LUTカスケード論理回路で実現されている場合について考える。

30

【 0 0 7 5 】

アドレス表にベクトル a を追加した場合、 $F(a)=c$ となる。従って、 $G(H(a_1), a_2)=c$ である。

【 0 0 7 6 】

ここで、登録ベクトル a に対して $F(a)$ の値を一对一に対応させるためには、部分関数 $H(a_1)$ の値は 0 であってはならない。そこで、もし部分関数 $H(a_1)$ の出力ベクトルが 0 となっている場合には 0 以外の値に修正する必要がある。この際、他の登録ベクトルの出力に影響を及ぼさないようにする必要があるため、部分関数 $H(a_1)$ の出力ベクトルは未使用の値に変更する。例えば、部分関数 $H(X_1)$ の非零出力の最大値よりも大きい未使用の値のうち、最小のものを選択することができる。

40

【 0 0 7 7 】

以上より、アドレス表への登録ベクトル a の追加に伴う LUTカスケード論理回路の修正は、次のようなアルゴリズムに従って行うことができる。

【 0 0 7 8 】

(アルゴリズム 3)

(Step 1) 未修正の p q 素子のうち、最も出力側から遠いものを選択する。選択された p q 素子を E_G と記す。 p q 素子 E_G により表される部分関数を $G(H(X_1), X_2)$ とする。ここで、 (X_1, X_2) X かつ X_1 $X_2=$ である。 (X_1, X_2) に対する入力ベクトルを (a_1, a_2) とす

50

る。但し、 p q 素子 E_G が最も出力側から遠い場合は、 $X_1=$, $H(X_1)=$ とする。

【 0 0 7 9 】

(Step 2) $e_0=G(H(a_1), a_2)$ とする。 pq 素子 E_G が最も出力側の場合、 $e=c$ 。そうでない場合、 $e_0=0$ ならば $e=e_0$ 、 $e_0=0$ ならば pq 素子 E_G で可能な出力値のうち、未使用の正の出力値で、最小のものを e とする。

【 0 0 8 0 】

(Step 3) pq 素子 E_G に記憶されたLUTの出力ベクトルのうち、入力ベクトル($H(a_1)$, a_2)に対応する出力ベクトルを $G(H(a_1), a_2)=e$ とする。

【 0 0 8 1 】

(Step 4) 未修正の p q 素子が残っていれば、Step 1に戻る。そうでなければ、処理を終了する。

(アルゴリズム 終り)

【 0 0 8 2 】

上記アルゴリズムを一般の p q 素子の多段論理関数の場合に拡張するのは容易である。一般の p q 素子の多段論理関数の場合は次のようになる。

【 0 0 8 3 】

(アルゴリズム 4)

(Step 1) 素子数が s の p q 回路網は、関数分解を $s-1$ 回繰り返し適用して構成できる。各関数分解で、入力側から順に p q 素子を抽出し、素子に 1 から r まで番号を付与する。出力の素子の番号は s である。番号 1 の素子 E_1 は、出力から最も離れている。入力変数 X に対する入力ベクトルを a とする。番号 1 の p q 素子 E_1 を E_G とする。

【 0 0 8 4 】

(Step 2) 多段論理回路に入力ベクトル a を入力する。このときの E_G の出力ベクトルを e_0 とする。

【 0 0 8 5 】

(Step 3) p q 素子 E_G が最も出力側の場合、 $e=c$ 。そうでない場合、 $e_0=0$ ならば $e=e_0$ 、 $e_0=0$ ならば p q 素子 E_G で可能な出力値のうち、未使用の正の出力値で、最小のものを e とする。

【 0 0 8 6 】

(Step 4) p q 素子 E_G に記憶された L U T の出力ベクトルのうち、入力ベクトル a に対応する出力ベクトルを e とする。

【 0 0 8 7 】

(Step 5) E_G が最も出力側の p q 素子ならば、処理を終了する。そうでなければ、Step 1で順序づけた、 E_G の次の p q 素子を E_G として、Step 2に戻る。

(アルゴリズム 終り)

【 0 0 8 8 】

以上のような登録ベクトルの削除と登録ベクトルの追加を行うことにより、アドレス生成関数のアドレス表を変化させた場合における p q 素子の多段論理回路の再構成が可能となる。 p q 素子の多段論理回路が L U T カスケードの場合、セル数及び登録ベクトル数 (アドレス数) に比例する時間で、アドレス表のアドレスの追加及び除去が可能となる。

【 0 0 8 9 】

(2) 本発明の構成及び作用

本発明に係る多段論理回路の再構成装置の第 1 の構成は、入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の L U T が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、入力ベクトル b に対する前記目的論理関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成装置であって、

未修正の前記 p q 素子のうち出力側に最も近い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G の L U T の出力ベクトルのうち前記入力ベクトル b に対する出力ベクト

10

20

30

40

50

ルをcとしたとき、当該出力ベクトルcに対応する入力ベクトルが、前記入力ベクトルb以外にも存在するか否かを検査する対応検査手段と、

前記 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の L U T において、前記入力ベクトルbが前記出力ベクトルcに一対一に対応する場合、前記入力ベクトルbに対する出力ベクトルcを無効値に書き換える削除修正手段と、を備えたことを特徴とする。

【 0 0 9 0 】

この構成によれば、

(1) まず、素子選択手段は未修正の p q 素子のうち出力側に最も近い p q 素子 E_G から順次選択する。

(2) 次に、対応検査手段は、 p q 素子 E_G の入力ベクトルbに対する出力ベクトルcが入力ベクトルbと一対一対応か否かを検査する。

(3) 次に、入力ベクトルbが前記出力ベクトルcに一対一に対応する場合には、削除修正手段は、 p q 素子 E_G の L U T において、入力ベクトルbに対する出力値を無効値に書き換え、当該 p q 素子 E_G を修正済みとする。

【 0 0 9 1 】

以上のような (1) ~ (3) の処理を繰り返すことにより、上述のアルゴリズム 1 又は 2 で説明したように、 p q 素子の多段論理回路において、入力ベクトルbに対する出力ベクトル $F(b)$ を無効値に変更する論理変更を行うことができる。

【 0 0 9 2 】

ここで、「多段論理回路」とは、上述したような複数の p q 素子がカスケード状又は樹形状に接続した論理回路であり、 L U T カスケード論理回路も含まれる。「無効値」とは、出力ベクトル $F(b)$ が無効であることを表す値であり、通常は 0 とされるが、必ずしもそれに限られるものではない。

【 0 0 9 3 】

本発明に係る多段論理回路の再構成装置の第 2 の構成は、入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の L U T が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、前記目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトルbに対する出力ベクトルaを追加する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成装置であって、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の L U T の出力ベクトルのうち前記入力ベクトルbに対する出力ベクトルcが無効値の場合、当該出力ベクトルcを、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応付手段と、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G の L U T の出力ベクトルのうち前記入力ベクトルbに対する出力ベクトルcを出力ベクトルaに書き換え、当該 p q 素子 E_G を修正済みとする追加修正手段と、を備えていることを特徴とする。

【 0 0 9 4 】

この構成によれば、

(1) まず、素子選択手段は、未修正の p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する。

(2) 次に、素子選択手段が選択した p q 素子 E_G が最も出力側ではない場合には、対応付手段は、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G の L U T の出力ベクトルのうち前記入力ベクトルbに対する出力ベクトルcが無効値の場合、当該出力ベクトルcを、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する。

(3) 一方、素子選択手段が選択した p q 素子 E_G が最も出力側の場合、追加修正手段は、 p q 素子 E_G の L U T において、入力ベクトルbに対する出力ベクトルcを出力ベクトルaに書き換え、当該 p q 素子 E_G を修正済みとする。

以上のような (1) ~ (3) の処理を繰り返すことにより、上述のアルゴリズム 3 又は 4

10

20

30

40

50

で説明したように、 p q 素子の多段論理回路において、入力ベクトル b に対する出力ベクトル a を追加する論理変更を行うことができる。

【 0 0 9 5 】

本発明に係る論理回路修正装置の第 1 の構成は、入力変数 X の目的論理関数 $Q(X)$ を演算する主論理回路について、前記入力変数 X として入力される各入力ベクトル b のうち特定の対象入力ベクトル b_i に対する主論理回路の出力ベクトル $Q(b_i)$ を、修正出力ベクトル $Q'(b_i)$ に変更する論理回路修正装置であって、

前記各対象入力ベクトル b_i に対応して、前記各出力ベクトル $Q(b_i)$ を修正出力ベクトル $Q'(b_i)$ に修正するための修正用ベクトル P_i が所定のアドレス A_i に登録される補助メモリと

10

前記補助メモリが出力する修正用ベクトル P_i を出力した場合、当該修正用ベクトル P_i 及び前記主論理回路が出力する出力ベクトル $Q(b_i)$ に基づいて、前記修正出力ベクトル $Q'(b_i)$ を出力する修正手段と、

前記入力変数 X に対して、当該入力変数 X の値が前記対象入力ベクトル b_i に等しい場合は前記修正用ベクトル P_i が格納された前記補助メモリのアドレス A_i を出力するアドレス生成関数 $F(X)$ の演算を行うアドレス生成回路と、を備え、

前記アドレス生成回路は、前記アドレス生成関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路により構成されており、

前記補助メモリは、前記アドレス生成回路が出力するアドレス A_i が入力されると、前記修正手段に前記修正用ベクトル P_i を出力することを特徴とする。

20

【 0 0 9 6 】

この構成によれば、まず、主論理回路に入力変数 X として入力ベクトル b が入力された場合、主論理回路は目的論理関数を演算し、出力ベクトル $Q(b)$ を出力する。一方、アドレス生成回路は、入力ベクトル b に対してアドレス生成関数 $F(b)$ を演算する。入力ベクトル b が対象入力ベクトル b_i であれば、アドレス生成回路は、アドレス A_i を出力する。このアドレス A_i は、補助メモリに入力される。補助メモリは、アドレス A_i に対して修正用ベクトル P_i を出力する。そして、修正手段は、修正用ベクトル P_i と出力ベクトル $Q(b_i)$ に基づいて、修正出力ベクトル $Q'(b_i)$ を出力する。これにより、目的論理関数の出力値の変更が行われる。尚、入力ベクトル b が対象入力ベクトル b_i 以外の場合には、アドレス生成回路は、アドレス A_i を出力しないため、補助メモリは修正用ベクトル P_i を出力しない。従って、修正手段は、出力ベクトル $Q(b_i)$ の修正を行うことなく、出力ベクトル $Q(b_i)$ がそのまま出力される。

30

【 0 0 9 7 】

本発明においては、アドレス生成回路として p q 素子の多段論理回路を使用するため、上に説明したように少ないメモリ容量でアドレス生成回路を実現することが可能である。従って、論理回路修正装置の実装面積を小さくでき、省電力化も図られる。また、上述したアルゴリズム 1, 2 及びアルゴリズム 3, 4 により、アドレス生成回路の論理変更も可能であり、要求に応じて主論理回路の出力を自在に修正することができる。

【 0 0 9 8 】

また、本発明に係る論理回路修正装置の場合、アドレス生成回路の再構成を行う際に、 p q 素子(メモリ)の内容を書き換えるのみであり、配線経路の変更は行う必要がない。従って、配線遅延の影響などを考慮する必要がなく容易にアドレス生成回路の変更を行うことができる。更に、配線の変更を伴わないことから、主論理回路の動作期間中にリアルタイムで動的な変更を行うことが可能である。

40

【 0 0 9 9 】

本発明に係る論理回路修正装置の第 2 の構成は、前記第 1 の構成において、前記修正用ベクトル P_i は、対象入力ベクトル b_i に対する主論理回路の出力ベクトル $Q(b_i)$ との排他論理和が、前記修正出力ベクトル $Q'(b_i)$ となる値に設定され、

前記補助メモリは、前記アドレス生成回路が出力するアドレス A_i が入力されると、前記

50

修正手段に前記修正用ベクトル P_i を出力し、それ以外の場合は0を出力するものであり、
前記修正手段は、前記補助メモリの出力値と前記主論理回路との排他論理和演算を行う
E X O Rゲートであることを特徴とする。

【0100】

本発明に係る論理回路修正装置の第3の構成は、前記第1の構成において、前記修正手段は、前記主論理回路及び前記補助メモリの出力段にそれぞれ設けられたトライ・ステート・バッファであり、

前記アドレス生成回路は、前記入力変数 X に対して、当該入力変数 X の値がいずれの前記対象入力ベクトル b_i とも等しくない場合には、無効値を出力するものであり、

前記主論理回路の出力段の前記トライ・ステート・バッファは、前記アドレス生成回路の出力値が無効値でない場合にはハイ・インピーダンス、それ以外の場合にはロー・インピーダンス状態となり、

前記補助メモリの出力段の前記トライ・ステート・バッファは、前記アドレス生成回路の出力値が無効値の場合にはハイ・インピーダンス、それ以外の場合にはロー・インピーダンス状態となることを特徴とする。

【0101】

本発明に係る論理回路修正装置の第4の構成は、前記第1乃至3の何れか一の構成において、前記補助メモリは、前記アドレス生成回路の最終段の p q 素子であることを特徴とする。但し、最終段の素子の出力数は、 q 以上になってもよいと考える。

【0102】

本発明に係る論理回路修正装置の第5の構成は、前記第1乃至4の何れか一の構成において、前記アドレス生成回路において、入力ベクトル b に対する前記アドレス生成関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記アドレス生成回路の再構成を行う再構成装置を備え、

前記再構成装置は、

未修正の前記 p q 素子のうち出力側に最も近い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G のLUTの出力ベクトルのうち前記入力ベクトル b に対する出力ベクトルを c としたとき、当該出力ベクトル c に対応する入力ベクトルが、前記入力ベクトル b 以外にも存在するか否かを検査する対応検査手段と、

前記 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G のLUTにおいて、前記入力ベクトル b が前記出力ベクトル c に一対一に対応する場合、前記入力ベクトル b に対する出力ベクトル c を無効値に書き換える削除修正手段と、を備えたことを特徴とする。

【0103】

本発明に係る論理回路修正装置の第6の構成は、前記第1乃至4の何れか一の構成において、前記アドレス生成回路において、前記アドレス生成関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、前記アドレス生成回路の再構成を行う再構成装置を備え、

前記再構成装置は、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G のLUTの出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応付手段と、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G のLUTの出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする追加修正手段と、を備えていることを特徴とする。

【0104】

本発明に係る多段論理回路の再構成方法の第1の構成は、入力変数を X とする目的論理

10

20

30

40

50

関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の pq 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、入力ベクトル b に対する前記目的論理関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成方法であって、

未修正の前記 pq 素子のうち出力側に最も近い pq 素子 E_G から順次選択する素子選択ステップと、

前記 pq 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトルを c としたとき、当該出力ベクトル c に対応する入力ベクトルが、前記入力ベクトル b 以外にも存在するか否かを検査する対応検査ステップと、

前記 pq 素子 E_G を修正済みとするとともに、当該 pq 素子 E_G の LUT において、前記入力ベクトル b が前記出力ベクトル c に一対一に対応する場合、前記入力ベクトル b に対する出力ベクトル c を無効値に書き換える削除修正ステップと、を繰り返し実行することを特徴とする。

【0105】

本発明に係る多段論理回路の再構成方法の第2の構成は、入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の pq 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された多段論理回路において、前記目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、前記多段論理回路の再構成を行う多段論理回路の再構成方法であって、

未修正の前記 pq 素子のうち出力側から最も遠い pq 素子 E_G から順次選択する素子選択ステップと、

前記 pq 素子 E_G が最も出力側ではない場合において、当該 pq 素子 E_G を修正済みとするとともに、当該 pq 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 pq 素子の出力ベクトルとして使用していないベクトル値に変更する対応付ステップと、

前記 pq 素子 E_G が最も出力側の場合において、前記 pq 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 pq 素子 E_G を修正済みとする追加修正ステップと、を繰り返し実行することを特徴とする。

【0106】

本発明に係る再構成可能多段論理回路の第1の構成は、入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数の LUT が記憶された複数の pq 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された再構成可能な多段論理回路であって、

入力ベクトル b に対する前記目的論理関数 $F(X)$ の出力ベクトル $F(b)$ を無効値に変更する論理変更に伴い、前記多段論理回路の再構成を行う再構成回路を備え、

前記再構成回路は、

未修正の前記 pq 素子のうち出力側に最も近い pq 素子 E_G から順次選択する素子選択手段と、

前記 pq 素子 E_G の LUT の出力ベクトルのうち前記入力ベクトル b に対する出力ベクトルを c としたとき、当該出力ベクトル c に対応する入力ベクトルが、前記入力ベクトル b 以外にも存在するか否かを検査する対応検査手段と、

前記 pq 素子 E_G を修正済みとするとともに、当該 pq 素子 E_G の LUT において、前記入力ベクトル b が前記出力ベクトル c に一対一に対応する場合、前記入力ベクトル b に対する出力ベクトル c を無効値に書き換える削除修正手段と、を備えたことを特徴とする。

【0107】

この構成により、上述したとおり、再構成回路は pq 素子(メモリ)の内容を書き換えるのみで、容易に目的論理関数 $F(X)$ の出力ベクトル $F(b)$ を削除することができる。また、配線の変更を伴わないことから、論理回路の動作期間中にリアルタイムで動的な変更を行うことが可能である。

【0108】

10

20

30

40

50

本発明に係る再構成可能多段論理回路の第1の構成は、入力変数を X とする目的論理関数 $F(X)$ を関数分解して得られる部分関数のLUTが記憶された複数の p q 素子が、前記各部分関数の入出力の接続関係に従って回路的に接続された再構成可能多段論理回路であって、

前記目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、前記多段論理回路の再構成を行う再構成回路を備え、

前記再構成回路は、

未修正の前記 p q 素子のうち出力側から最も遠い p q 素子 E_G から順次選択する素子選択手段と、

前記 p q 素子 E_G が最も出力側ではない場合において、当該 p q 素子 E_G を修正済みとするとともに、当該 p q 素子 E_G のLUTの出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値の場合、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する対応付手段と、

前記 p q 素子 E_G が最も出力側の場合において、前記 p q 素子 E_G のLUTの出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする追加修正手段と、を備えていることを特徴とする。

【0109】

この構成により、上述したとおり、再構成回路は p q 素子(メモリ)の内容を書き換えるのみで、容易に目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更を行うことができる。また、配線の変更を伴わないことから、論理回路の動作期間中にリアルタイムで動的な変更を行うことが可能である。

【発明の効果】

【0110】

以上のように、本発明に係る多段論理回路の再構成装置によれば、 p q 素子で構成される多段論理回路の論理変更に伴う再構成を、短時間に行うことができる。また、再構成処理は少ない計算量で短時間に行うことができ、再構成計算に必要なとされるメモリ容量も小さいため、チップ内に専用回路として組み込むことが可能である。

【0111】

また、本発明に係る論理回路修正装置によれば、アドレス生成回路として p q 素子の多段論理回路を使用するため、上述したように少ないメモリ容量でアドレス生成回路を実現することが可能である。従って、論理回路修正装置の実装面積を小さくでき、省電力化も図られる。また、アドレス生成回路の論理変更も可能であり、要求に応じて主論理回路の出力を自在に修正することができる。

【図面の簡単な説明】

【0112】

【図1】アドレス生成論理関数 F の関数分解を示す図である。

【図2】 p q 素子によるアドレス生成回路($p = 10$)の例を示す図である。

【図3】 p q 素子によるアドレス生成回路($p = 11$)の例を示す図である。

【図4】 p q 素子によるアドレス生成回路($p = 12$)の例を示す図である。

【図5】本発明の実施例1に係る再構成可能な多段論理回路の構成を表す図である。

【図6】 s 個の p q 素子 $E(1)$, $E(2)$, ..., $E(s)$ が直列に接続されたカスケード論理回路を示す図である。

【図7】再構成装置1による目的論理関数 $F(X)$ の登録ベクトルの削除処理を表すフローチャートである。

【図8】再構成装置1による目的論理関数 $F(X)$ の登録ベクトルの追加処理を表すフローチャートである。

【図9】例5のアドレス生成関数 $F(X)$ を実現するLUTカスケード論理回路である。

【図10】本発明の実施例2に係る論理回路修正装置20の構成を示す図である。

【図11】本発明の実施例3に係る論理回路修正装置20の構成を示す図である。

【図12】本発明の実施例4に係る再構成可能な多段論理回路の構成を表す図である。

10

20

30

40

50

【図13】Xilinx社のFPGAで用いられている4入力LUTである。

【図14】4入力LUTを3個用いた12入力1出力の回路である。

【図15】LUTカスケード論理回路を一般的に表したものである。

【図16】参照テーブル9のテーブルの例である。

【図17】インデックス・テーブルの例である。

【図18】実施例4に係る再構成可能な多段論理回路の登録ベクトルの追加処理のフローチャートである。

【図19】実施例4に係る再構成可能な多段論理回路の登録ベクトルの削除処理のフローチャートである。

【図20】特許文献1に記載の論理回路修正装置103の構成を示す図である。

10

【符号の説明】

【0113】

1, 1' 再構成回路

2 素子選択手段

3 素子選択手段

4 対応検査手段

5 削除修正手段

6 対応付手段

7 追加修正手段

8 インデックス・テーブル

20

9 参照テーブル

10 多段論理回路

11 入力データバス

12 出力データバス

13 pq素子

20 論理回路修正装置

21 アドレス生成回路

22 補助メモリ

23 修正回路

24 EXORゲート

30

30 主論理回路

31 入力バス

32 出力バス

41 __0 ~ 41__15 Dフリップ・フロップ(DFF)

42 マルチプレクサ(MUX)

43 クロック線

44 登録値入力線

44 データ入力線

【発明を実施するための最良の形態】

【0114】

40

以下、本発明を実施するための最良の形態について、図面を参照しながら説明する。

【実施例1】

【0115】

図5は、本発明の実施例1に係る再構成可能な多段論理回路の構成を表す図である。この再構成可能な多段論理回路は、多段論理回路10と再構成装置1とが組み合わされたシステムとして構成されている。再構成回路1は、多段論理回路10の論理変更に伴い、多段論理回路10の再構成を行う装置である。

【0116】

多段論理回路10は、入力変数をXとする目的論理関数F(X)の演算を行う論理回路である。この多段論理回路10は、複数のpq素子13を備えている。各pq素子13には、

50

目的論理関数 $F(X)$ を関数分解して得られる部分関数のLUTが記憶されている。また、それぞれの p q 素子13は、各部分関数の入出力の接続関係に従って回路的に接続されている。多段論理回路10の構造としては、LUTカスケードや図2～図4に示したような p q 回路網などを用いることができる。

【0117】

入力変数 X は n 次元の2値ベクトルの変数である。目的論理関数 $F(X)$ の出力変数 Y は m 次元の2値ベクトルの変数である。入力変数 X は 2^n 個の値を取り得る。そのうちの k 個($k < 2^n$)の入力ベクトルを「登録ベクトル(registered vector)」と呼ぶ。それ以外のベクトルを「無効ベクトル(invalid vector)」と呼ぶ。

【0118】

多段論理回路10の入力データバス11からは、入力変数 X として入力ベクトル b が入力される。また、多段論理回路10の出力データバス12からは、出力変数 $Y=F(X)$ として出力ベクトル a が出力される。

【0119】

再構成回路1は、素子選択手段2, 3、対応検査手段4、削除修正手段5、対応付手段6、追加修正手段7、及び多段論理回路10を備えている。

【0120】

目的論理関数 $F(X)$ の入力ベクトル b に対する出力ベクトル $F(b)$ を無効値にする論理変更に伴い、削除処理が実行され、多段論理回路10の再構成が行われる。この削除処理において、素子選択手段2は、未修正の前記 p q 素子13を、出力側に最も近い p q 素子 E_G から、順次選択する。対応検査手段4は、 p q 素子 E_G のLUTにおいて、入力ベクトル b が出力ベクトル c に一対一に対応しているか否か検査する。削除修正手段5は、 p q 素子 E_G のLUTにおいて、入力ベクトル b が出力ベクトル c に一対一に対応している場合に、入力ベクトル b に対する出力値を無効値に書き換え、当該 p q 素子 E_G を修正済みとする。

【0121】

一方、目的論理関数 $F(X)$ の出力ベクトル集合に入力ベクトル b に対する出力ベクトル a を追加する論理変更に伴い、追加処理が実行され、多段論理回路10の再構成が行われる。この追加処理において、素子選択手段3は、未修正の p q 素子13を、出力側から最も遠い p q 素子 E_G から、順次選択する。対応付手段6は、 p q 素子 E_G が最も出力側ではない場合、当該 p q 素子 E_G を修正済みとする。更にこの場合において、対応付手段6は、当該 p q 素子 E_G のLUTの出力ベクトルのうち前記入力ベクトル b に対する出力ベクトル c が無効値ならば、当該出力ベクトル c を、当該 p q 素子の出力ベクトルとして使用していないベクトル値に変更する。追加修正手段7は、 p q 素子 E_G が最も出力側の場合、 p q 素子 E_G のLUTの入力ベクトル b に対する出力値を出力ベクトル a に書き換え、当該 p q 素子 E_G を修正済みとする。

【0122】

以上のように構成された本実施例1に係る多段論理回路の再構成回路1について、以下その動作を説明する。

【0123】

尚、以下の説明において、多段論理回路10は、図6に示したような s 個の p q 素子 $E(1)$, $E(2)$, ..., $E(s)$ が直列に接続されたカスケード論理回路とする。各 p q 素子は、入力側から $E(1)$, $E(2)$, ..., $E(s)$ の順に接続されている。

【0124】

入力変数を X とする。入力変数 X の分割を $X=(X_1, X_2, \dots, X_s)$ とする。また、入力変数 X に対する目的論理関数 $F(X)$ の出力変数を $Y=F(X)$ とする。目的論理関数 $F(X)$ を次式(11)のように部分関数 G_1, G_2, \dots, G_s に関数分解する。式(11)において U_i ($i=1, 2, \dots, s-1$)は中間変数である。

【0125】

10

20

30

40

【数 1 1】

$$\begin{aligned}
 U_1 &= G_1(X_1) \\
 U_2 &= G_2(X_2, U_1) \\
 &\vdots \\
 U_s &= G_s(X_s, U_{s-1}) \\
 Y &= U_s
 \end{aligned}
 \tag{11}$$

【 0 1 2 6 】

p q 素子E(i) (i=1, 2, ..., s)には、中間関数G_iのLUTが記憶されている。p q 素子E(1) 10
は、入力変数X₁が入力される。それに対し、p q 素子E(1)は出力変数U₁=G₁(X₁)を出力する。p q 素子E(i) (i=2, 3, ..., s)は、前段のp q 素子E(i-1)の出力変数U_{i-1}と入力変数X_iが入力される。それに対し、p q 素子E(i)は出力変数U_i=G_i(X_i, U_{i-1})を出力する。また、p q 素子E(s)の出力変数U_sは出力変数Yである。

【 0 1 2 7 】

最初に、目的論理関数F(X)の登録ベクトルの削除処理について説明する。図7は、再構成回路1による目的論理関数F(X)の登録ベクトルの削除処理を表すフローチャートである。

【 0 1 2 8 】

ここでは、目的論理関数F(X)の登録ベクトルbを削除する論理変更に伴う、多段論理回路10の再構成について説明する。尚、入力変数Xの部分変数X_iに対応する入力ベクトルbの部分値をb_iと記す。 20

【 0 1 2 9 】

まず、ステップS1において、削除する登録ベクトルbが入力されると、素子選択手段2はインデックスiをsに設定する。インデックスiはp q 素子の選択番号を示すインデックスである。これにより、最も出力側のp q 素子E(s)が選択される。

【 0 1 3 0 】

次に、ステップS2において、素子選択手段2は、p q 素子E(i)のLUTを読み出す。

【 0 1 3 1 】

次に、ステップS3において、削除修正手段5は、読み出されたLUTにおいて、入力ベクトルbに対応する出力ベクトルu_i(b)を0とする。入力ベクトルbに対応する出力ベクトルu_i(b)は、前段のp q 素子E(i-1)の出力ベクトルu_{i-1}(b)及び部分値b_iに対応する出力ベクトルである。前段のp q 素子E(i-1)の出力ベクトルu_{i-1}(b)は、多段論理回路10に入力ベクトルbを入力して演算を行い、中間変数U_{i-1}を取り出すことにより得られる。 30

【 0 1 3 2 】

次に、ステップS4において、修正されたLUTをp q 素子E(i)に書き込むことにより、p q 素子E(i)を更新する。

【 0 1 3 3 】

次に、ステップS5において、素子選択手段2は、インデックスiが2以上であるか否かを判定する。インデックスiが1の場合には、登録ベクトルの削除処理を終了する。そうでない場合には、次のステップS6に移行する。 40

【 0 1 3 4 】

ステップS6において、素子選択手段2は、選択番号を示すインデックスiを1だけデクリメントする。

【 0 1 3 5 】

次に、ステップS7において、素子選択手段2は、p q 素子E(i)のLUTを読み出す。

【 0 1 3 6 】

次に、ステップS8において、対応検査手段4は、p q 素子E(i)の入力ベクトルbに対する出力ベクトルu_i(b)を抽出する。入力ベクトルbに対応する出力ベクトルu_i(b)は、前段のp q 素子E(i-1)の出力ベクトルu_{i-1}(b)及び部分値b_i (i=1の場合は、部分値b_i) 50

応する出力ベクトルである。前段の p q 素子 $E(i-1)$ の出力ベクトル $u_{i-1}(b)$ は、多段論理回路 10 に入力ベクトル b を入力して演算を行い、中間変数 U_{i-1} を取り出すことにより得られる。

【0137】

次に、ステップ S9 において、対応検査手段 4 は、出力ベクトル $u_i(b)$ に対応する入力ベクトルが、入力ベクトル b 以外にもあるか否かを判定する。これは、LUT の出力値を調べ、出力ベクトル $u_i(b)$ と同じ値が 2 個以上あるか否かを検査すればよい。対応する入力ベクトルが 2 個以上あった場合には、登録ベクトルの削除処理を終了する。そうでない場合には、次のステップ S3 に戻る。

【0138】

以上の一連の処理により、目的論理関数 $F(X)$ の登録ベクトル b を削除する論理変更に伴う多段論理回路 10 の再構成を行うことができる。この処理は、各 p q 素子 $E(i)$ の LUT の読み出しと書き換えのみであり、高速に実行することができる。また、複雑な演算処理も必要としないため、素子選択手段 2、対応検査手段 4、及び削除修正手段 5 は簡単な回路で実現することができる。

【0139】

次に、目的論理関数 $F(X)$ の登録ベクトルの追加処理について説明する。図 8 は、再構成回路 1 による目的論理関数 $F(X)$ の登録ベクトルの追加処理を表すフローチャートである。

【0140】

ここでは、目的論理関数 $F(X)$ の登録ベクトル b を追加する論理変更に伴い、多段論理回路 10 の再構成を行う。登録ベクトル b に対応する目的論理関数 $F(X)$ の出力ベクトルを a とする。尚、入力変数 X の部分変数 X_i に対応する入力ベクトル b の部分値を b_i と記す。

【0141】

まず、ステップ S11 において、追加する登録ベクトル b 及び出力ベクトル a が入力されると、素子選択手段 3 は、 p q 素子の選択番号を示すインデックス i を 1 に設定する。これにより、最も入力側の p q 素子 $E(1)$ が選択される。

【0142】

次に、ステップ S12 において、素子選択手段 3 は、 p q 素子 $E(i)$ の LUT を読み出す。

【0143】

次に、ステップ S13 において、対応付手段 6 は、インデックス i が最大値 s か否かを検査する。 $i = s$ の場合にはステップ S20 に移行し、そうでない場合には次のステップ S14 に移行する。

【0144】

ステップ S14 において、対応付手段 6 は、読み出された LUT において、入力ベクトル b に対応する出力ベクトル $u_i(b)$ を計算する。入力ベクトル b に対応する出力ベクトル $u_i(b)$ は、前段の p q 素子 $E(i-1)$ の出力ベクトル $u_{i-1}(b)$ 及び部分値 b_i ($i=1$ の場合は、部分値 b_i) に対応する出力ベクトルである。前段の p q 素子 $E(i-1)$ の出力ベクトル $u_{i-1}(b)$ は、多段論理回路 10 に入力ベクトル b を入力して演算を行い、中間変数 U_{i-1} を取り出すことにより得られる。

【0145】

次に、ステップ S15 において、対応付手段 6 は、出力ベクトル $u_i(b)$ が 0 か否かを検査する。 $u_i(b) = 0$ の場合には、ステップ S19 に移行する。 $s = 0$ の場合には、次のステップ S16 に移行する。

【0146】

ステップ S16 において、対応付手段 6 は、 p q 素子 $E(i)$ の LUT の出力ベクトルを検査し、出力ベクトルとして未使用の値のうち最小の値 (ベクトル) e を抽出する。

【0147】

次に、ステップ S17 において、対応付手段 6 は、 p q 素子 $E(i)$ の LUT の入力ベクトル b に対応する出力ベクトル $u_i(b)$ を e に変更する。

10

20

30

40

50

【 0 1 4 8 】

次に、ステップ S 1 8 において、対応付手段 6 は、p q 素子 E(i) に、更新した L U T を書き込むことにより、p q 素子 E(i) を更新する。

【 0 1 4 9 】

次に、ステップ S 1 9 において、素子選択手段 3 は、選択番号を示すインデックス i を 1 だけインクリメントし、ステップ S 1 2 に戻る。

【 0 1 5 0 】

一方、ステップ S 1 3 において $i = s$ の場合、ステップ S 2 0 において、追加修正手段 7 は、p q 素子 E(i) の L U T の入力ベクトル b に対応する出力ベクトル $u_i(b)$ を出力ベクトル a に変更する。

10

【 0 1 5 1 】

そして、ステップ S 2 1 において、追加修正手段 7 は、p q 素子 E(i) に、更新した L U T を書き込むことにより、p q 素子 E(i) を更新する。

【 0 1 5 2 】

以上の一連の処理により、目的論理関数 F(X) の登録ベクトル b を追加する論理変更に伴う多段論理回路 1 0 の再構成を行うことができる。この処理は、各 p q 素子 E(i) の L U T の読み出しと書き換えのみであり、高速に実行することができる。また、複雑な演算処理も必要としないため、素子選択手段 2 及び削除修正手段 5 は簡単な回路で実現することができる。

20

【 0 1 5 3 】

また、本実施例の多段論理回路 1 0 では、再構成はメモリの書き換えのみによって実行される。すなわち、多段論理回路 1 0 の再構成に伴う配線構造の変更はない。従って、再構成にあたり配線遅延の影響などをする必要がなく、容易に実行することができる。さらには、論理回路の動作期間中にリアルタイムで動的に論理回路の再構成を行うことも可能である。

【 0 1 5 4 】

(例 5)

(表 5) に示したような 5 変数のアドレス生成関数 F(X) を考える。

【 0 1 5 5 】

【表 5】

30

x_5	x_4	x_3	x_2	x_1	f_3	f_2	f_1
0	0	1	1	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	0	1	1	0	0
1	0	0	1	0	1	0	1
1	1	0	0	0	1	1	0

40

【 0 1 5 6 】

入力変数 X の分割 $X=(X_1, X_2)$, $X_1=(x_5, x_4, x_3, x_2)$, $X_2=(x_1)$ について関数分解 $F(X)=G(H(X_1), X_2)$ を行い、図 9 に示したような 2 つの p q 素子から成る L U T カスケード論理回路によりアドレス生成関数 F(X) を実現する。このとき、部分関数 H, G の真理値表は (表 6), (表 7) のようになる。入力側の p q 素子には (表 6) の部分関数 H の真理値表が L U T として格納される。また、出力側の p q 素子には (表 7) の部分関数 G の真理値表が L U T として格納される。(表 6) (表 7) の各行の左側に付された数字は、その行に対応する登録ベクトルのインデックスを表す。

【 0 1 5 7 】

【表 6】

関数Hの真理値表

x_5	x_4	x_3	x_2	h_3	h_2	h_1	
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	2
0	1	1	0	0	1	1	3, 4
1	0	0	1	1	0	1	5
1	1	0	0	1	1	0	6

10

【 0 1 5 8 】

【表 7】

関数Gの真理値表

h_3	h_2	h_1	x_1	f_3	f_2	f_1	
0	0	1	1	0	0	1	1
1	0	1	0	0	1	0	2
0	1	1	0	0	1	1	3
0	1	1	1	1	0	0	4
1	0	1	0	1	0	1	5
1	1	0	0	1	1	0	6

20

【 0 1 5 9 】

このアドレス生成関数F(X)について、登録ベクトル $b_7=(x_5 \ x_4 \ x_3 \ x_2 \ x_1)=(11001)$ を追加する。(表 8)は、登録ベクトル b_7 の真理値表を表す。登録ベクトル b_7 に対する出力ベクトルには(111)を割り当てることとする。

30

【 0 1 6 0 】

【表 8】

x_5	x_4	x_3	x_2	x_1	f_3	f_2	f_1	
1	1	0	0	1	1	1	1	7

【 0 1 6 1 】

次に、図 8 のフローチャートに示した方法に従って、各 p q 素子の L U T の書き換えを行う。まず、入力側の p q 素子に格納された部分関数 H の入力ベクトル b に対応する出力ベクトルは、(表 6)の $(x_5 \ x_4 \ x_3 \ x_2)=(1100)$ の欄を参照すると、 $(h_1, h_2, h_3)=(110)$ である。これは 0 ではないため、部分関数 H の L U T は変化させない。このとき、入力ベクトル b は(表 9)に示したように、 $(x_5 \ x_4 \ x_3 \ x_2)=(1100)$ の欄に対応付けられる。

40

【 0 1 6 2 】

【表 9】

関数Hの真理値表

x_5	x_4	x_3	x_2	h_3	h_2	h_1	
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	2
0	1	1	0	0	1	1	3, 4
1	0	0	1	1	0	1	5
1	1	0	0	1	1	0	6, 7

10

【 0 1 6 3】

次に、出力側の p q 素子に格納された部分関数 G について考えると、入力ベクトル b に対応する部分関数 G の入力ベクトルは $(h_1, h_2, h_3, x_1) = (1101)$ である。従って、(表 10) に示したように、部分関数 G の LUT に、 $(h_1, h_2, h_3, x_1, f_1, f_2, f_3) = (1101111)$ を追加する。これにより、登録ベクトル b_7 の追加に伴う各 p q 素子の再構成が完了する。

【 0 1 6 4】

【表 10】

関数Gの真理値表

h_3	h_2	h_1	x_1	f_3	f_2	f_1	
0	0	1	1	0	0	1	1
1	0	1	0	0	1	0	2
0	1	1	0	0	1	1	3
0	1	1	1	1	0	0	4
1	0	1	0	1	0	1	5
1	1	0	0	1	1	0	6
1	1	0	1	1	1	1	7

20

【 0 1 6 5】

次に、登録ベクトルの削除に伴う各 p q 素子の再構成について考える。例として、(表 5) のアドレス表に登録された登録ベクトルから、登録ベクトル $b_3 = (01100)$ を削除する場合について考える。

【 0 1 6 6】

図 7 のフローチャートに示した方法に従って、まず出力側の p q 素子の LUT の書き換えを行う。(表 10) を参照すると、登録ベクトル b_3 に対する部分関数 G の入力ベクトルは、 $(h_1, h_2, h_3, x_1) = (0110)$ である。これに対応する出力ベクトルを (000) に変更し、出力側の p q 素子の LUT を (表 11) に示したように書き換える。

40

【 0 1 6 7】

【表 1 1】

関数Gの真理値表

h_3	h_2	h_1	x_1	f_3	f_2	f_1	
0	0	1	1	0	0	1	1
1	0	1	0	0	1	0	2
0	1	1	0	0	0	0	3
0	1	1	1	1	0	0	4
1	0	1	0	1	0	1	5
1	1	0	0	1	1	0	6
1	1	0	1	1	1	1	7

10

【 0 1 6 8】

次に、入力側の p q 素子について考える。この場合、更新前の部分関数 G の真理値表 (表 1 0) を参照すると、前段の L U T の出力ベクトル (h_1, h_2, h_3) が登録ベクトル b_3 に対する出力ベクトル (011) と同じで、非零の出力を持つ登録ベクトルとして、 b_3 の他に $b_4=(0111)$ が存在する。従って、入力側の p q 素子については L U T の書き換えは行われぬ。以上により、登録ベクトル b_3 の削除に伴う各 p q 素子の再構成が完了する。

20

【 0 1 6 9】

次に、さらに登録ベクトル $b_4=(01101)$ を削除する場合について考える。図 7 のフローチャートに示した方法に従って、まず出力側の p q 素子の L U T の書き換えを行う。(表 9) を参照すると、登録ベクトル b_4 に対する部分関数 G の入力ベクトルは、(h_1, h_2, h_3, x_1)=(0111) である。これに対応する出力ベクトルを (000) に変更し、出力側の p q 素子の L U T を (表 1 2) に示したように書き換える。

【 0 1 7 0】

【表 1 2】

関数Gの真理値表

h_3	h_2	h_1	x_1	f_3	f_2	f_1	
0	0	1	1	0	0	1	1
1	0	1	0	0	1	0	2
0	1	1	0	0	0	0	
0	1	1	1	0	0	0	4
1	0	1	0	1	0	1	5
1	1	0	0	1	1	0	6
1	1	0	1	1	1	1	7

30

【 0 1 7 1】

次に、入力側の p q 素子について考える。この場合、更新前の部分関数 G の真理値表 (表 1 1) を参照すると、前段の L U T の出力ベクトル (h_1, h_2, h_3) が登録ベクトル b_4 に対する出力ベクトル (011) と同じで、非零の出力を持つ登録ベクトルは存在しない。従って、入力側の p q 素子について、登録ベクトル b_4 に対応する部分関数 H の入力ベクトル (x_5, x_4, x_3, x_2)=(0111) に対する出力ベクトル (h_1, h_2, h_3)=(100) を (000) に変更する。従って、入力側の p q 素子の L U T は (表 1 3) のように更新される。これにより、登録ベクトル b_4 の削除に伴う各 p q 素子の再構成が完了する。

40

【 0 1 7 2】

50

【表 1 3】

関数Hの真理値表

x_5	x_4	x_3	x_2	h_3	h_2	h_1	
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	2
0	1	1	0	0	0	0	4
1	0	0	1	1	0	1	5
1	1	0	0	1	1	0	6

10

(例終り)

【実施例 2】

【0173】

図 10 は、本発明の実施例 2 に係る論理回路修正装置 20 の構成を示す図である。図 10 において、主論理回路 30 は、論理修正の対象となる論理回路であり、専用の L S I (A S I C) として構成されたメモリ、C P U 等である。主論理回路 30 の入力変数を $X=(x_1, x_2, \dots, x_n)$ ($B^n, B=\{0,1\}$)、出力変数を $Q=Q(X)=(q_1, q_2, \dots, q_m)$ (B^m) とし、論理回路修正装置 20 により修正された出力変数を $Q'=(q_1', q_2', \dots, q_m')$ (B^m) とする。主論理回路 30 は、入力変数 X に対して目的論理関数 $Q(X)$ を演算する。

20

【0174】

論理回路修正装置 20 は、入力変数 X として入力される各入力ベクトル b のうち特定の対象入力ベクトル b_i に対する主論理回路 30 の出力ベクトル $Q(b_i)$ を、修正出力ベクトル $Q'(b_i)$ に変更する。論理回路修正装置 20 は、アドレス生成回路 21、補助メモリ 22、修正回路 23、及び再構成回路 1 を備えている。尚、再構成回路 1 は、実施例 1 において説明した多段論理回路の再構成回路 1 と同様のものである。

【0175】

補助メモリ 22 は、各対象入力ベクトル b_i に対応して、各出力ベクトル $Q(b_i)$ を修正出力ベクトル $Q'(b_i)$ に修正するための修正用ベクトル $P_i=(p_{i1}, p_{i2}, \dots, p_{im})$ が所定のアドレス A_i に登録されるメモリである。補助メモリ 22 は、修正用ベクトル P_i が登録されたアドレス A_i が入力された場合には、修正用ベクトル P_i を出力し、それ以外の場合は、無効値として 0 を出力する。

30

【0176】

修正回路 23 は、補助メモリ 22 が出力する修正用ベクトル P_i を出力した場合、当該修正用ベクトル P_i 及び主論理回路 30 が出力する出力ベクトル $Q(b_i)$ に基づいて、修正出力ベクトル Q_i' を出力する。ここで、本実施例 2 においては、修正回路 23 は、主論理回路 30 の各出力線に対応して設けられた E X O R ゲート 24 により構成されている。各 E X O R ゲート 24 は、出力ベクトル Q の各成分 q_1, q_2, \dots, q_m と、修正用ベクトル P_i の各成分 $p_{i1}, p_{i2}, \dots, p_{im}$ との排他論理和により、修正出力ベクトル $Q_i'=(q_{i1}', q_{i2}', \dots, q_{im}')$ の演算を行う。

40

【0177】

【数 1 2】

$$q_j' = q_j \oplus p_{ij} \quad (i = 1, 2, \dots, k; j = 1, 2, \dots, m) \quad (12)$$

【0178】

従って、修正用ベクトル P_i の値は、上記式 (1 2) により目的とする修正出力ベクトル Q_i' が得られるような値に設定される。

50

【 0 1 7 9 】

アドレス生成回路 2 1 は、入力変数 X として入力される各入力ベクトル b_i に対して補助メモリ 2 2 のアドレスを出力する回路である。ここで、アドレス生成回路 2 1 は、各対象入力ベクトル b_i に対しては修正用ベクトル P_i が登録されたアドレス A_i を出力し、それ以外の入力ベクトル b_i に対しては無効値 (0) を出力するようなアドレス生成関数 $F(X)$ の演算を行う。

【 0 1 8 0 】

また、アドレス生成回路 2 1 は、上述したような p q 素子を多段に結合して構成された多段論理回路によって構成されている。

【 0 1 8 1 】

再構成回路 1 は、アドレス生成回路 2 1 が演算するアドレス生成関数 $F(X)$ の修正に伴い、アドレス生成回路 2 1 の各 p q 素子の L U T の再構成を行う。尚、再構成の方法に関しては、実施例 1 において述べた通りである。

【 0 1 8 2 】

以上のように構成された本実施例 2 の論理回路修正装置 2 0 について、以下その動作を説明する。

【 0 1 8 3 】

まず、入力変数 X として、対象入力ベクトル b_i 以外の入力ベクトル b_i が入力された場合、アドレス生成回路 2 1 は出力ベクトル a として 0 (無効値) を出力する。従って、補助メモリ 2 2 は、修正用ベクトル P として 0 を出力する。各 E X O R ゲート 2 4 は、主論理回路 3 0 の出力値 q_j と 0 との排他論理和、すなわち出力値 q_j を出力する。従って、この場合は出力ベクトル Q は修正されず、そのまま出力されることになる。

【 0 1 8 4 】

一方、入力変数 X として、対象入力ベクトル b_i が入力された場合、アドレス生成回路 2 1 は出力ベクトル a として、修正用ベクトル P_i が格納された補助メモリ 2 2 のアドレス値 A_i を出力する。従って、補助メモリ 2 2 は、修正用ベクトル P として修正用ベクトル P_i を出力する。各 E X O R ゲート 2 4 は、主論理回路 3 0 の出力値 q_j と修正用ベクトルの成分 p_{ij} との排他論理和 q_{ij}' を出力する。これにより、出力ベクトルの修正が行われる。

【 0 1 8 5 】

本実施例においては、アドレス生成回路 2 1 として p q 素子を多段に結合して構成された多段論理回路を使用したことで、論理回路修正装置 2 0 の実装面積を小さくすることが可能となる。また、論理変更が追加して行われた場合にも、再構成回路 1 を用いて容易にアドレス生成回路 2 1 の再構成を行うことができる。

【実施例 3】

【 0 1 8 6 】

図 1 1 は、本発明の実施例 3 に係る論理回路修正装置 2 0 の構成を示す図である。図 1 1 において、再構成回路 1、アドレス生成回路 2 1、補助メモリ 2 2、及び主論理回路 3 0 は、図 1 0 と同様のものである。本実施例においては、補助メモリ 2 2 には、各対象入力ベクトル b_i に対して修正出力ベクトル Q_i' がそのまま格納されている。また、補助メモリ 2 2 及び主論理回路 3 0 の出力段には、修正手段としてトライ・ステート・バッファ (図示せず) が設けられている。

【 0 1 8 7 】

アドレス生成回路 2 1 は、入力バス 3 1 から入力される入力変数 X に対して、当該入力変数 X の値がいずれの対象入力ベクトル b_i とも等しくない場合には、無効値を出力する。主論理回路 3 0 の出力段のトライ・ステート・バッファは、アドレス生成回路 2 1 の出力値が無効値でない場合にはハイ・インピーダンス、それ以外の場合にはロー・インピーダンス状態となる。また、補助メモリ 2 2 の出力段のトライ・ステート・バッファは、アドレス生成回路 2 1 の出力値が無効値の場合にはハイ・インピーダンス、それ以外の場合にはロー・インピーダンス状態となる。

【 0 1 8 8 】

10

20

30

40

50

これにより、入力変数 X の値が対象入力ベクトル b_i 以外の場合には、出力バス 3 2 には、主論理回路 3 0 の出力ベクトルが出力され、入力変数 X の値が対象入力ベクトル b_i の場合には、出力バス 3 2 には、補助メモリ 2 2 の出力ベクトルが出力される。

【実施例 4】

【0189】

図 1 2 は、本発明の実施例 4 に係る再構成可能な多段論理回路の構成を表す図である。図 1 2 において、図 5 と同様な構成部分については、同符号を付す。

【0190】

本実施例の再構成可能な多段論理回路においては、再構成回路 1 ' に新たに、インデックス・テーブル 8 と参照テーブル 9 とが追加されている。

10

【0191】

多段論理回路 1 0 は、実施例 1 において説明したとおり、入力変数を X とする目的論理関数 $F(X)$ の演算を行う論理回路である。多段論理回路 1 0 は、複数の $p q$ 素子 1 3 が多段に接続された論理回路である。この、多段論理回路 1 0 の具体的な構造としては、複数の $p q$ 素子 1 3 がカスケード状に接続された LUT カスケード論理回路(図 6, 図 1 5 参照)や、複数の $p q$ 素子 1 3 が樹形状に接続された LUT ツリー論理回路(図 2 ~ 4 参照)を用いることができる。

【0192】

再構成回路 1 は、この多段論理回路 1 0 の論理変更に伴って、多段論理回路 1 0 の再構成を行う回路である。再構成回路 1 は、素子選択手段 2, 素子選択手段 3, 対応検査手段 4, 削除修正手段 5, 対応付手段 6, 追加修正手段 7, インデックス・テーブル 8, 及び参照テーブル 9 を備えている。素子選択手段 2, 素子選択手段 3, 対応検査手段 4, 削除修正手段 5, 追加修正手段 7, 及び対応付手段 6 については、実施例 1 で既に説明をした通りであるので、説明は省略する。

20

【0193】

尚、 $p q$ 素子 1 3 は、必ずしもメモリを用いて構成する必要はなく、メモリと同様の機能を有する回路であればどのような回路を用いてもよい。

【0194】

例えば、図 1 3 は、Xilinx社のFPGAで用いられている4入力 LUT である。この4入力 LUT は、16個の同期型のDフリップ・フロップ $41_0 \sim 41_15$ と、1つの16入力1出力のマルチプレクサ42を備えている。

30

【0195】

この4入力 LUT では、モードを切り替えることで、シフトレジスタとして使用でき、書き換えが可能である。

【0196】

図 1 3 (a) は $SRL16$ モードであり、このモードではクロック線 4 3 から各 $DFF41_0 \sim 41_15$ にクロックが入力される。従って、 $DFF41_0 \sim 41_15$ はシフトレジスタとして機能する。この状態で登録値入力線 4 4 から登録データをシリアルに入力することにより、各 $DFF41_0 \sim 41_15$ に登録データが記憶される。

【0197】

40

一方、図 1 3 (b) は4入力 LUT モードであり、このモードではクロック線 4 3 にクロックの入力はなく、この回路は LUT として機能する。データ入力線 4 5 から $MUX42$ に4ビットのデータが入力されると、 $MUX42$ は、その入力データに応じて $DFF41_0 \sim 41_15$ の何れか一つを選択し、選択した DFF に記憶されたデータを出力する。従って、この回路は4入力1出力の LUT として機能する。

【0198】

図 1 4 に、4入力 LUT を3個用いた12入力1出力の回路を示す。図 1 4 に示すように、FPGA内部のマルチプレクサと組み合わせることで、複数の LUT の出力の論理積を計算できる。尚、この回路は1出力なので、複数ビットの出力を得たい場合には、図 1 4 の回路を必要なビット数だけ並列に並べればよい。

50

【 0 1 9 9 】

p q 素子 1 3 としては、図 1 3 に示したような L U T を使用することも可能である。

【 0 2 0 0 】

参照テーブル 9 は、各 p q 素子 1 3 毎に割り当て済みのレイル・ベクトル (rail vector) を保持するテーブルである。

【 0 2 0 1 】

ここで、「レイル (rail) 」とは、複数の p q 素子が多段に接続された多段論理回路において、前段の p q 素子の出力線のうち、後段の p q 素子の入力に接続された接続線をいう。

【 0 2 0 2 】

図 1 5 は、L U T カスケード論理回路を一般的に表したものである。図 1 5 の p q 素子 E(1) の出力線のうち p q 素子 E(2) の入力に接続されたものがレイルである。また、p q 素子 E(2) の出力線のうち p q 素子 E(3) の入力に接続されたものがレイルである。以下同様である。図 2 ~ 4 の L U T ツリー論理回路についても、同様にレイルが定義される。

【 0 2 0 3 】

また、「レイル・ベクトル」とは、ある p q 素子の出力側のレイルに出力される出力変数のベクトルをいう。

【 0 2 0 4 】

参照テーブル 9 の例を図 1 6 に示す。参照テーブル 9 は、多段論理回路 1 0 の各 p q 素子毎につずつ設けられる。1 つの p q 素子に対する参照テーブル 9 は、その p q 素子の各レイル・ベクトルとそれに対応する参照ベクトル数とのテーブルとで構成される。尚、「参照ベクトル数」とは、そのレイル・ベクトルを参照する登録ベクトルの個数である。

【 0 2 0 5 】

インデックス・テーブル 8 は、ベクトルが登録済みであるか否かを記憶するテーブルである。インデックス・テーブル 8 は、1 つだけ設けられる。図 1 7 に、インデックス・テーブルの例を示す。インデックス・テーブル 9 は、登録可能な各ベクトルに割り当てられたインデックスと、そのインデックスに対応する登録ベクトルが登録済み (使用されている) か否かを示す登録済フラグとで構成される。

【 0 2 0 6 】

以上のように構成された本実施例に係る再構成可能な多段論理回路について、以下その登録ベクトルの追加・削除の動作について説明する。

【 0 2 0 7 】

尚、ここでは説明を分かりやすくするため、多段論理回路 1 0 として、図 6 に示したような L U T カスケード論理回路が用いられていることとする。

【 0 2 0 8 】

(1) 登録ベクトルの追加

本実施例の再構成可能な多段論理回路では、登録ベクトルを新たに追加する場合は、インデックス・テーブル 8 及び参照テーブル 9 を使用し、処理時間の短縮が図られる。

【 0 2 0 9 】

図 1 8 は、実施例 4 に係る再構成可能な多段論理回路の登録ベクトルの追加処理のフローチャートである。

【 0 2 1 0 】

まず、ステップ S 3 1 において、登録する入力ベクトル b が、再構成回路 1 ' に入力される。

【 0 2 1 1 】

次に、ステップ S 3 2 において、追加修正手段 7 は、インデックス・テーブル 8 を参照し、入力ベクトル b は多段論理回路 1 0 に登録済みか否かを検査する。ここで、インデックス・テーブル 8 の入力ベクトル b に対する登録済フラグが 1 の場合には終了する。

【 0 2 1 2 】

次に、ステップ S 3 3 において、素子選択手段 3 は、選択素子の番号を表すインデック

10

20

30

40

50

ス i を 1 に設定する。

【 0 2 1 3 】

次に、ステップ S 3 4 において、対応付手段 6 は、多段論理回路 1 0 に入力ベクトル b を入力し、 $p q$ 素子 $E(i)$ の出力ベクトル $u_i(b)$ を読み出す。

【 0 2 1 4 】

次に、ステップ S 3 5 において、対応付手段 6 は、出力ベクトル $u_i(b)$ が 0 か否かを判定する。0 の場合には、ステップ S 3 8 に移行し、0 以外の場合には次のステップ S 3 6 に移行する。

【 0 2 1 5 】

次に、ステップ S 3 6 において、対応付手段 6 は、 $p q$ 素子 $E(i)$ に対応する参照テーブル 9 を検索し、未参照のレイル・ベクトル a を 1 つ索出する。そして、ステップ S 3 6 において、追加修正手段 7 は、 $p q$ 素子 $E(i)$ の出力ベクトル $u_i(b)$ がレイル・ベクトル a となるように $p q$ 素子 $E(i)$ のメモリ内容の書き換えを行う。

【 0 2 1 6 】

ステップ S 3 8 において、対応付手段 6 は、 $p q$ 素子 $E(i)$ に対応する参照テーブルの登録ベクトル $u_i(b)$ の参照ベクトル数を 1 だけインクリメントする。

【 0 2 1 7 】

次に、ステップ S 3 9 において、素子選択手段 3 は、選択素子の番号を表すインデックス i を 1 だけインクリメントする。

【 0 2 1 8 】

次に、ステップ S 4 0 において、素子選択手段 3 は、 $i = s$ か否かを判定する。 $i < s$ ならばステップ S 3 4 に戻り、 $i = s$ ならば次のステップ S 4 1 に進む。

【 0 2 1 9 】

ステップ S 4 1 において、追加修正手段 7 は、 $p q$ 素子 $E(s)$ の出力ベクトル $u_s(b)$ を $F(b)$ とばるように、 $p q$ 素子 $E(s)$ のメモリ内容の書き換えを行う。

【 0 2 2 0 】

最後に、ステップ S 4 2 において、追加修正手段 7 は、インデックス・テーブル 8 の登録ベクトル b に対する登録済フラグを 1 に設定して、登録ベクトル追加処理を終了する。

【 0 2 2 1 】

(2) 登録ベクトルの削除

図 1 9 は、実施例 4 に係る再構成可能な多段論理回路の登録ベクトルの削除処理のフローチャートである。

【 0 2 2 2 】

まず、ステップ S 5 1 において、削除する入力ベクトル b が、再構成回路 1 ' に入力される。

【 0 2 2 3 】

次に、ステップ S 5 2 において、削除修正手段 5 は、インデックス・テーブル 8 を参照し、入力ベクトル b は多段論理回路 1 0 に登録済みであるか否かを検査する。インデックス・テーブル 8 の入力ベクトル b に対する登録済フラグが 0 の場合は、登録ベクトルの削除処理を終了する。

【 0 2 2 4 】

次に、ステップ S 5 3 において、素子選択手段 2 は、選択素子の番号を表すインデックス i を s に設定する。

【 0 2 2 5 】

次に、ステップ S 5 4 において、対応検査手段 4 は、多段論理回路 1 0 に入力ベクトル b を入力し、 $p q$ 素子 $E(i)$ の出力ベクトル $u_i(b)$ を読み出す。

【 0 2 2 6 】

次に、ステップ S 5 5 において、対応検査手段 4 は、 $p q$ 素子 $E(i)$ に対応する参照テーブルの登録ベクトル $u_i(b)$ の参照ベクトル数を 1 だけデクリメントする。但し、このとき登録ベクトル $u_i(b)$ の参照ベクトル数が 0 の場合には、登録ベクトル $u_i(b)$

10

20

30

40

50

b) の参照ベクトル数は 0 のままとする。

【0227】

次に、ステップ S 5 6 において、削除修正手段 5 は、登録ベクトル $u_i(b)$ の参照ベクトル数が 0 か否かを判定する。登録ベクトル $u_i(b)$ の参照ベクトル数が 0 でないならばステップ S 5 8 に戻り、0 ならば次のステップ S 5 7 に進む。

【0228】

ステップ S 5 7 において、削除修正手段 5 は、pq 素子 $E(i)$ の出力ベクトル $u_i(b)$ が 0 となるように、pq 素子 $E(i)$ のメモリ内容の書き換えを行う。

【0229】

ステップ S 5 8 において、素子選択手段 2 は、選択素子の番号を表すインデックス i を 1 だけデクリメントする。

【0230】

次に、ステップ S 5 9 において、素子選択手段 2 は、 $i = 0$ か否かを判定する。 $i > 0$ ならばステップ S 5 4 に戻り、 $i = 0$ ならば次のステップ S 6 0 に進む。

【0231】

ステップ S 6 0 において、削除修正手段 5 は、インデックス・テーブル 8 の登録ベクトル b に対する登録済フラグを 0 に設定して、登録ベクトル削除処理を終了する。

【0232】

以上の処理によって、多段論理回路 10 に登録ベクトルを 1 つ追加するのに必要なステップ数は、以下の式で見積もることができる。

【0233】

【数13】

$$Add_{cas} = Op.Cas + s(Op.Cas + k \times Acc.Mem + Acc.Mem) \quad (13)$$

【0234】

ここで、 $Op.Cas$ は LUT カスケードにアクセスするために必要なステップ数、 $Acc.Mem$ は参照テーブルにアクセスするために必要なステップ数、 s はセル数、 k は登録ベクトル数を示す。第 1 項は、上記登録ベクトルの追加処理のうちステップ S 3 1 ~ S 3 4 の処理を行うのに必要なステップ数の見積りを示し、第 2 項はそれ以外の処理を行うのに必要なステップ数の見積りを示す。

【0235】

同様に、多段論理回路 10 から登録ベクトルを 1 つ削除するのに必要なステップ数は以下の式で見積もることができる。

【0236】

【数14】

$$Del_{cas} = Op.Cas + s(Op.Cas + Acc.Mem) \quad (14)$$

【0237】

第 1 項は上記登録ベクトルの削除処理のうちステップ S 5 1 ~ S 5 4 の処理を行うのに必要なステップ数の見積りを示し、第 2 項はそれ以外の処理を行うのに必要なステップ数の見積りを示す。

【0238】

式 (13), (14) より、多段論理回路 10 が LUT カスケード論理回路の場合、登録ベクトルの追加に要するステップ数は s と k に、削除に要するステップ数は s にほぼ比例する事が分かる。

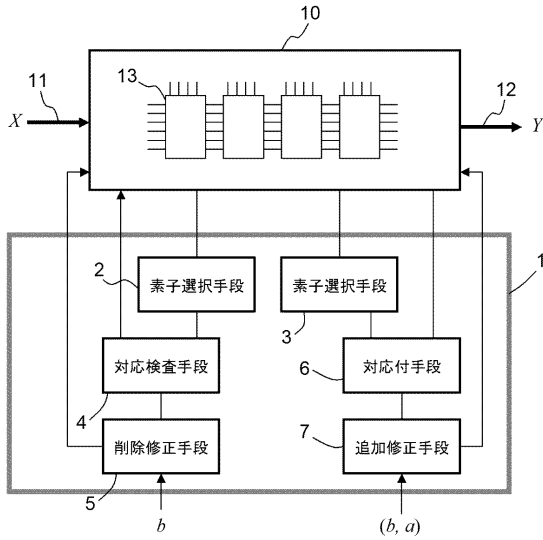
10

20

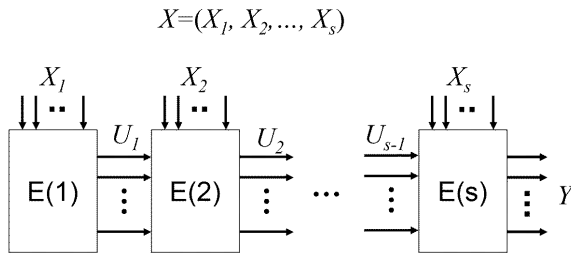
30

40

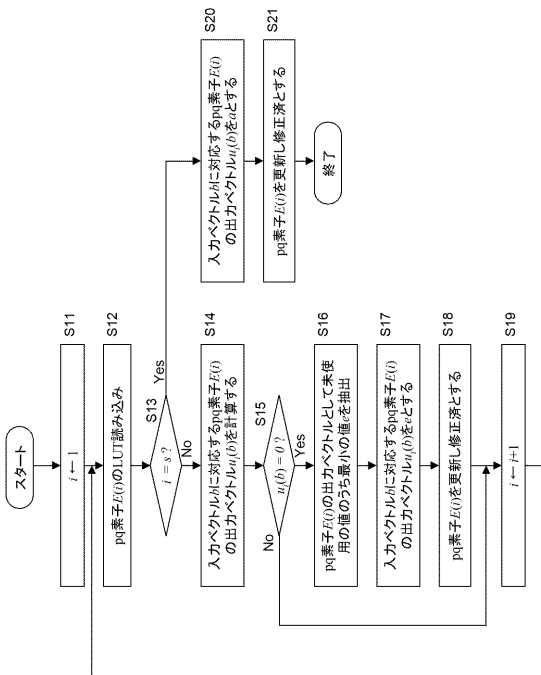
【図5】



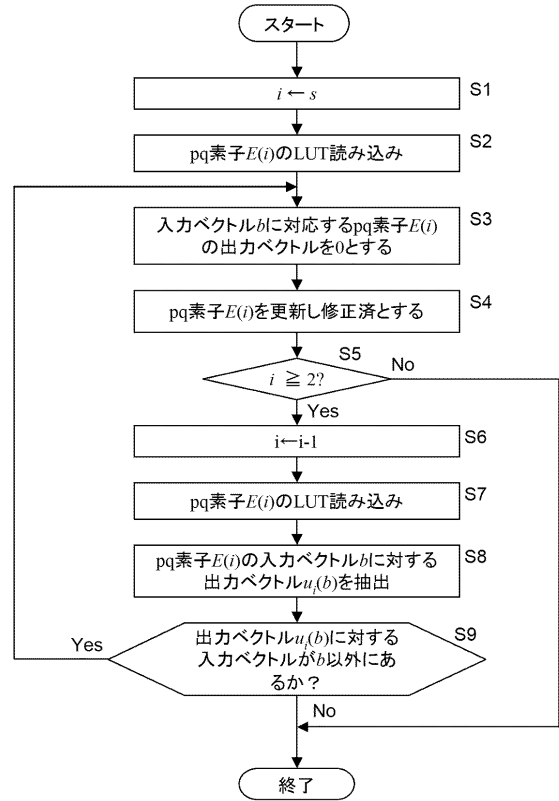
【図6】



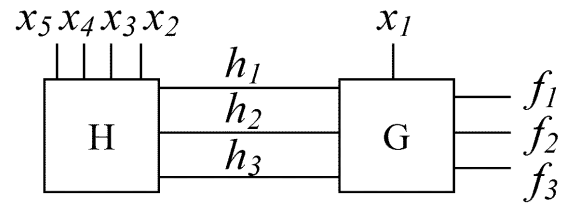
【図8】



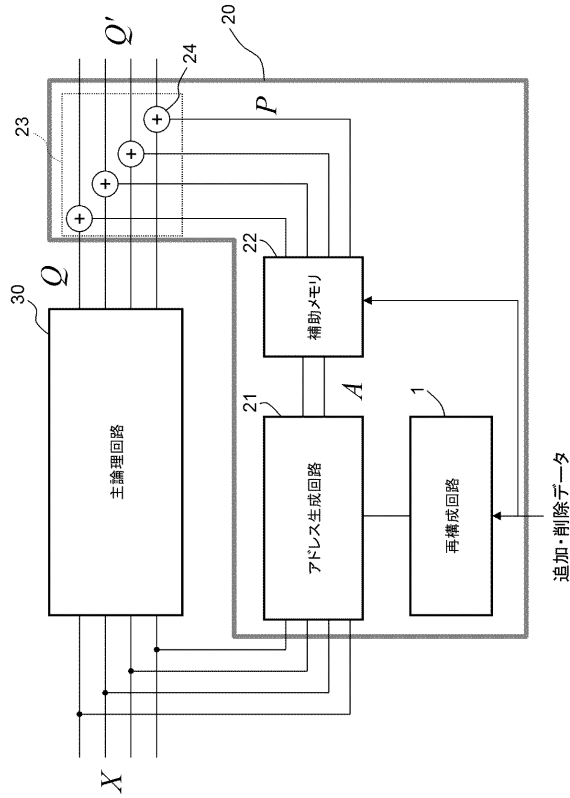
【図7】



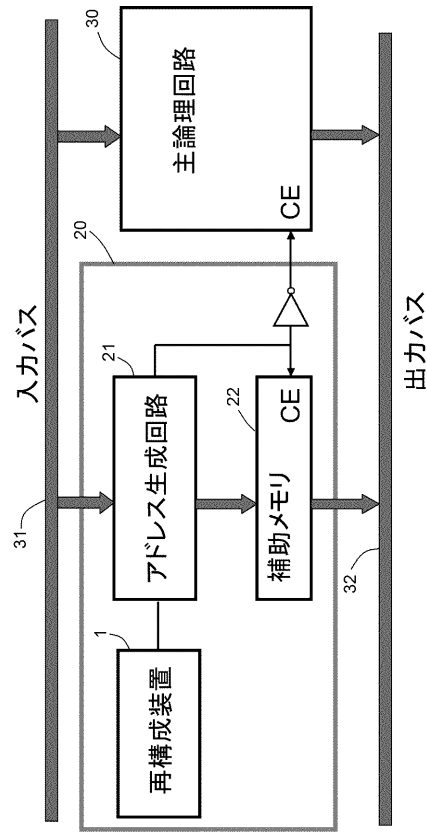
【図9】



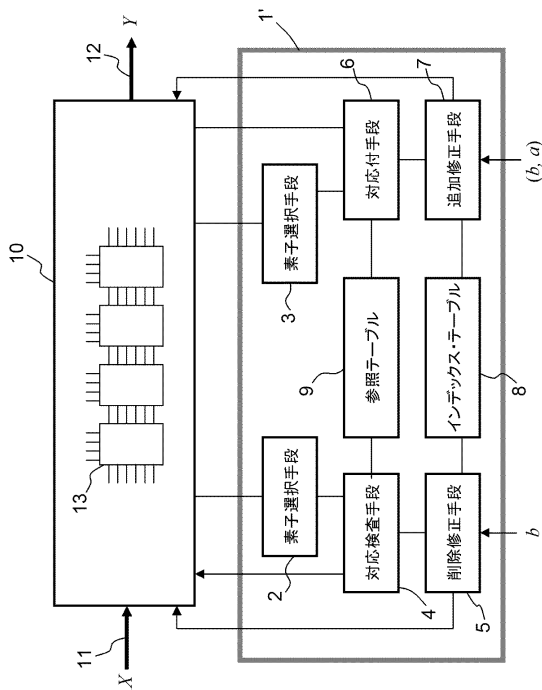
【図10】



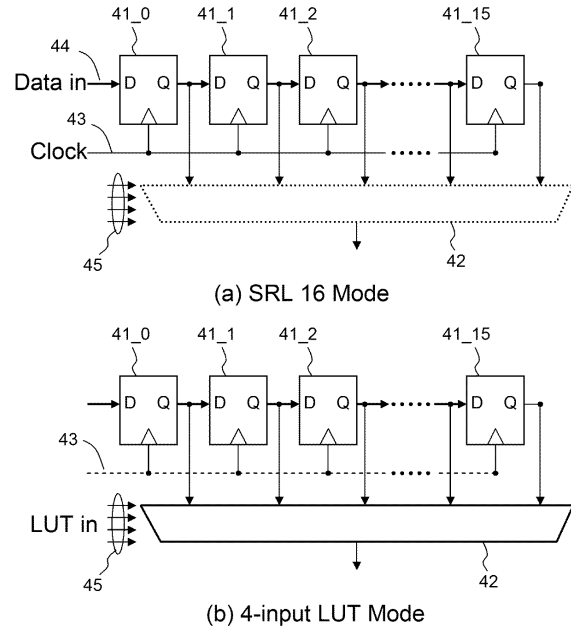
【図11】



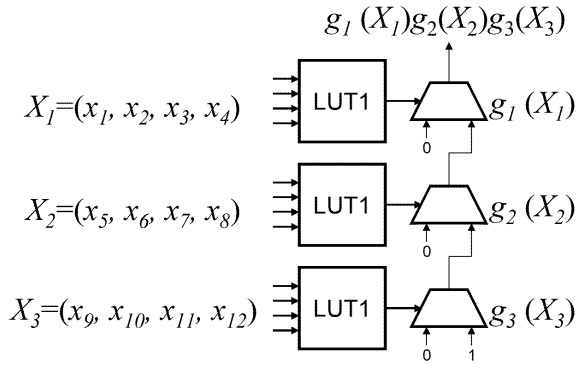
【図12】



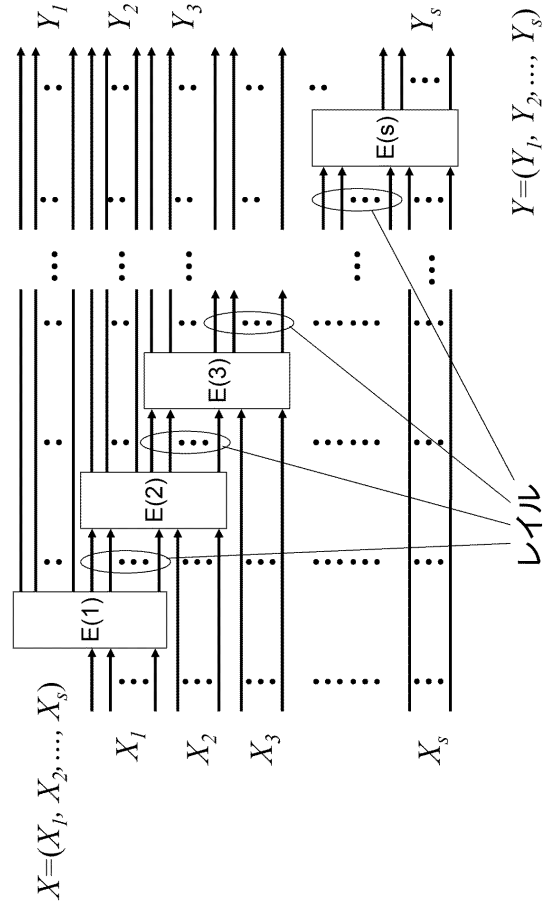
【図13】



【図14】



【図15】

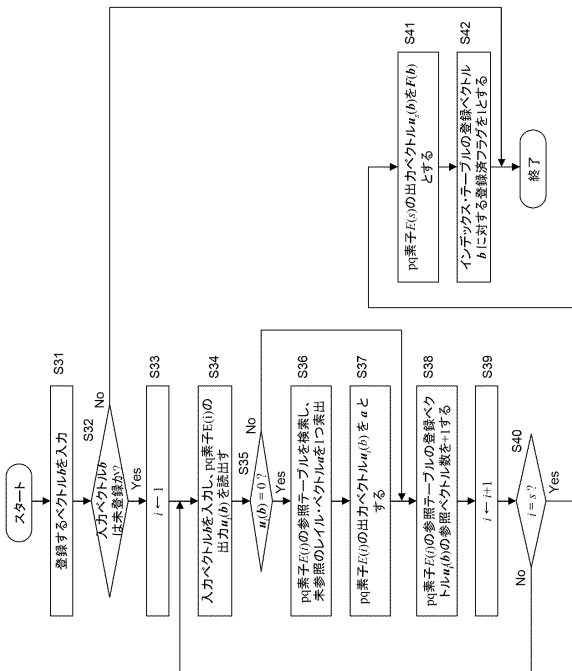


【図16】

address=rail vector	# of referenced vectors (0 denotes non-referenced)
00000000	1
00000001	3
00000010	2
00000011	1
00000100	4
00000101	0
...	...
11111111	0

of registrable vectors (# of registrable outputs)

【図18】

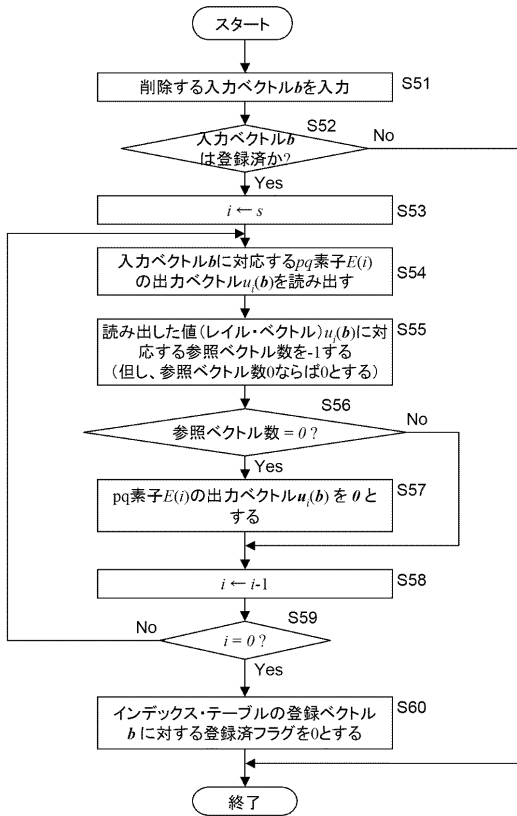


【図17】

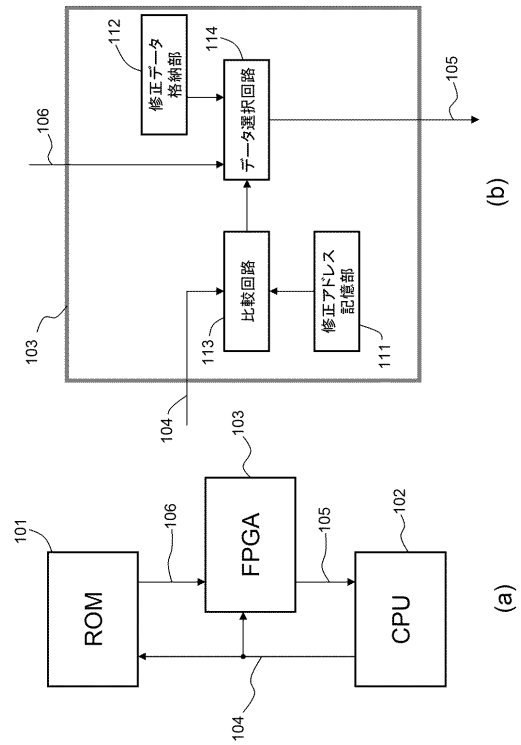
address=index for registered vector	0:non-used 1:used
00000001	1
00000010	0
00000011	1
00000100	1
00000101	0
00000110	0
...	...
11111111	0

of registrable vectors

【図19】



【図20】



フロントページの続き

(56)参考文献 特開平06-004266(JP,A)

特開平09-297776(JP,A)

Hui Qin, Tsutomu Sasao, Jon.T. Butler, Implementation of LPM address generator on FPGAs, Reconfigurable Computing: Architectures and Applications. Second International Workshop, ARC 2006. (Lecture Notes in Computer Science Vol. 3985), 2006年 3月 3日, p.170-181, URL, http://www.lsi-cad.com/sasao/Papers/files/ARC2006_qin.pdf

(58)調査した分野(Int.Cl., DB名)

G06F 7/00

H03K 19/173