

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6229990号
(P6229990)

(45) 発行日 平成29年11月15日(2017.11.15)

(24) 登録日 平成29年10月27日(2017.10.27)

(51) Int.Cl. F I
G 1 1 C 15/04 (2006.01) G 1 1 C 15/04 6 3 1 W
G 0 6 F 17/30 (2006.01) G 0 6 F 17/30 4 1 2

請求項の数 15 (全 29 頁)

| | |
|--|--|
| <p>(21) 出願番号 特願2016-544229 (P2016-544229)</p> <p>(86) (22) 出願日 平成27年8月19日 (2015.8.19)</p> <p>(86) 国際出願番号 PCT/JP2015/073227</p> <p>(87) 国際公開番号 W02016/027829</p> <p>(87) 国際公開日 平成28年2月25日 (2016.2.25)</p> <p>審査請求日 平成28年12月22日 (2016.12.22)</p> <p>(31) 優先権主張番号 特願2014-168777 (P2014-168777)</p> <p>(32) 優先日 平成26年8月21日 (2014.8.21)</p> <p>(33) 優先権主張国 日本国(JP)</p> <p>(出願人による申告) 平成25年度 独立行政法人科学技術振興機構 研究成果展開事業 研究成果最速展開支援プログラム 産業技術力強化法第19条の適用を受ける特許出願</p> | <p>(73) 特許権者 801000027 学校法人明治大学 東京都千代田区神田駿河台1-1</p> <p>(74) 代理人 100064908 弁理士 志賀 正武</p> <p>(74) 代理人 100106909 弁理士 棚井 澄雄</p> <p>(74) 代理人 100108578 弁理士 高橋 詔男</p> <p>(74) 代理人 100126882 弁理士 五十嵐 光永</p> <p>(72) 発明者 笹尾 勤 神奈川県川崎市多摩区東三田1-1-1 学校法人明治大学 生田キャンパス内</p> |
|--|--|

最終頁に続く

(54) 【発明の名称】 連想記憶装置、インデックス生成器、及び登録情報更新方法

(57) 【特許請求の範囲】

【請求項1】

複数のハッシュ関数を用いて入力ベクトルの特徴量を生成し、前記特徴量に基づき登録ベクトルに関する登録情報を検索することにより前記入力ベクトルに対応したインデックスを生成する複数のインデックス生成部と、

前記複数のインデックス生成部の各出力を結合して前記入力ベクトルに対応したインデックスを出力する出力部と、

前記登録情報の更新を制御する制御部と、
を備え、

前記複数のインデックス生成部のそれぞれは、

前記特徴量に基づき生成されたインデックスが前記登録情報として存在するか否かを示す第1信号を生成して前記制御部に供給すると共に、前記特徴量に基づき生成されたインデックスを用いて前記入力ベクトルの再生ベクトルを生成し、前記再生ベクトルが前記入力ベクトルと一致するか否かを示す第2信号を生成して前記制御部に供給する、連想記憶装置。

【請求項2】

前記複数のインデックス生成部のそれぞれは、

前記入力ベクトルから前記入力ベクトルの特徴量を算出するためのハッシュ回路と、

前記登録ベクトルのインデックスが前記登録情報として格納された主記憶部を有し、前記主記憶部から前記特徴量に対応した仮のインデックスを読み出して出力する仮インデッ

クス生成回路と、

前記仮インデックス生成回路から出力された仮のインデックスに基づいて前記第 1 信号を生成して出力する衝突信号生成回路と、

前記登録ベクトルが前記登録情報として格納された副記憶部を有し、前記副記憶部から前記仮のインデックスに対応したベクトルを読み出して前記再生ベクトルとして出力する再生ベクトル生成回路と、

前記再生ベクトルと前記入力ベクトルとを比較し、前記再生ベクトルと前記入力ベクトルとが一致するか否かを示す前記第 2 信号を出力する比較回路と、

前記第 2 信号が前記再生ベクトルと前記入力ベクトルとの一致を示す場合、前記仮のインデックスを前記入力ベクトルに対応した固有のインデックスとして出力し、前記第 2 信号が前記再生ベクトルと前記入力ベクトルとの不一致を示す場合、無効値を出力する出力回路と、

を備えた請求項 1 に記載の連想記憶装置。

【請求項 3】

前記主記憶部と前記副記憶部が、1つのメモリに統合された、請求項 2 に記載の連想記憶装置。

【請求項 4】

前記主記憶部は、前記特徴量を示すビット列がアドレス信号として入力されるメモリから構成された、請求項 2 または 3 に記載の連想記憶装置。

【請求項 5】

前記副記憶部は、前記仮のインデックスを示すビット列がアドレス信号として入力されるメモリから構成された、請求項 2 から 4 の何れか 1 項に記載の連想記憶装置。

【請求項 6】

前記複数のインデックス生成部のうちの一部のインデックス生成部に代えて、または、前記複数のインデックス生成部に加えて、CAMセルを有する連想メモリを備えた、請求項 1 から 5 の何れか 1 項に記載の連想記憶装置。

【請求項 7】

前記制御部は、

前記登録情報として登録ベクトルとそのインデックスとを追加する場合、追加すべき登録ベクトルが前記登録情報として存在しない旨を示す前記第 1 信号を供給するインデックス生成部を選択し、選択した当該インデックス生成部の登録情報として、前記追加すべき登録ベクトルに対応するインデックスを追加する、請求項 1 から 6 の何れか 1 項に記載の連想記憶装置。

【請求項 8】

前記制御部は、

前記登録情報として登録された登録ベクトルのインデックスを削除する場合、前記複数のインデックス生成部のうち、削除すべき登録ベクトルのインデックスを用いて生成された再生ベクトルが前記入力ベクトルと一致する旨を示す前記第 2 信号を供給するインデックス生成部を選択し、選択した当該インデックス生成部の登録情報から、前記登録ベクトルに対応するインデックスを削除する、請求項 1 から 7 の何れか 1 項に記載の連想記憶装置。

【請求項 9】

入力ベクトルから前記入力ベクトルの特徴量を算出して出力するハッシュ回路と、

登録ベクトルに関する登録情報として前記登録ベクトルに対応するインデックスが格納された主記憶部を有し、前記主記憶部から前記特徴量に対応した仮のインデックスを読み出して出力する仮インデックス生成回路と、

前記仮インデックス生成回路から出力された仮のインデックスに基づいて、前記特徴量に対応した仮のインデックスが前記登録情報として存在するか否かを示す第 1 信号を生成して出力する信号生成回路と、

前記登録ベクトルに関する登録情報として前記登録ベクトルが格納された副記憶部を有

10

20

30

40

50

し、前記副記憶部から前記仮のインデックスに対応したベクトルを読み出して再生ベクトルとして出力する再生ベクトル生成回路と、

前記再生ベクトルと前記入力ベクトルとを比較し、前記再生ベクトルと前記入力ベクトルとが一致するか否かを示す第2信号を出力する比較回路と、

前記第2信号が前記再生ベクトルと前記入力ベクトルとの一致を示す場合、前記仮のインデックスを前記入力ベクトルに対応した固有のインデックスとして出力する出力回路と、

を備えたインデックス生成器。

【請求項10】

請求項1から8の何れか1項に記載の連想記憶装置の登録情報を更新する情報更新方法であって、

前記連想記憶装置は、動作モードとして、検索モードと更新モードとを備え、

前記連想記憶装置に備えられた制御部は、前記更新モードにおいて前記登録情報を更新するための制御を実施する、情報更新方法。

【請求項11】

前記検索モードにおいて、前記複数のインデックス生成部と前記出力部とが連想メモリとして機能する、請求項10に記載の情報更新方法。

【請求項12】

前記更新モードにおいて、前記制御部は、前記登録情報として登録された登録ベクトルに対応するインデックスを削除した後、前記登録情報として追加すべき登録ベクトルに対応するインデックスを追加することにより、前記登録情報を更新する、請求項10または11に記載の情報更新方法。

【請求項13】

前記更新モードにおいて、

削除すべき登録ベクトルを前記入力ベクトルとして前記連想記憶装置に備えられた複数のインデックス生成部に供給する第1段階と、

前記複数のインデックス生成部のそれぞれについて、前記制御部により、前記第2信号に基づき、前記削除すべき登録ベクトルに対応した固有のインデックスが前記登録情報として存在するか否かを判定する第2段階と、

前記複数のインデックス生成部の何れかにおいて、前記削除すべき登録ベクトルに対応した固有のインデックスが前記登録情報として存在する場合、前記制御部により、前記登録情報として存在する前記削除すべき登録ベクトルに対応するインデックスの値を有効値から無効値に変更する第3段階と、

を含む請求項12に記載の情報更新方法。

【請求項14】

前記更新モードにおいて、

追加すべき登録ベクトルを前記入力ベクトルとして前記複数のインデックス生成部に供給する第1段階と、

前記複数のインデックス生成部のそれぞれについて、前記制御部により、前記第1信号に基づき、前記追加すべき登録ベクトルに対応した仮のインデックスが前記登録情報として存在するか否かを判定する第2段階と、

前記複数のインデックス生成部の何れかにおいて、前記追加すべき登録ベクトルに対応した仮のインデックスが前記登録情報として存在しない場合、前記追加すべき登録ベクトルに対応したインデックスの値として有効値を設定する第3段階と、

を含む請求項12または13に記載の情報更新方法。

【請求項15】

前記複数のインデックス生成部の何れかにおいて、前記追加すべき登録ベクトルに対応した仮のインデックスが前記登録情報として存在する場合、前記制御部により、前記追加すべき登録ベクトルに対応した仮のインデックスが存在しない他のインデックス生成部において、前記追加すべき登録ベクトルに対応したインデックスの値として有効値を設定す

10

20

30

40

50

る、請求項 1 4 に記載の情報更新方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、連想記憶装置、インデックス生成器、及び登録情報更新方法に関し、特に、連想記憶装置に備えられたインデックス生成器に登録された登録情報の更新を高速化するための技術に関する。

【背景技術】

【0002】

通常メモリは、与えられたアドレス（インデックス）に対して、そのアドレスに格納されている登録データを生成する。一方、CAM(Content Addressable Memory)は、与えられた検索（入力）データに対して、それを格納するCAMのアドレスを生成する（非特許文献1, 2参照）。

10

【0003】

CAMは、パターン・マッチング、インターネットのルータ、プロセッサのキャッシュ、TLB(Translation Lookaside Buffer)、データ圧縮、データベースのアクセラレータ、ニューラルネット、メモリパッチなど幅広い分野において利用されている。

【0004】

通常、CAMは、その機能から、2値CAM(Binary CAM:以下「BCAM」という。)及び3値CAM(Ternary CAM:以下「TCAM」という。)の二種類に分類される。BCAMでは、各セルに0及び1を格納する。TCAMでは各セルに0, 1, 及び*を格納する。ここで、「*」はドント・ケア(don't care)を表し、0と1の両方にマッチする。

20

【0005】

CAMの機能をソフトウェアで実現することも可能であるが、ソフトウェアで実現したものは大幅に低速である。そのため、専用のハードウェアを用いてCAMを実現することが多い。

【先行技術文献】

【特許文献】

【0006】

30

【特許文献1】特開2004-295967号公報

【非特許文献】

【0007】

【非特許文献1】菅野卓雄監修, 香山晋編, 「超高速デバイス・シリーズ2 超高速MOSデバイス」, 初版, 培風館, 1986年2月, pp. 324-325.

【非特許文献2】電子情報通信学会編, 「LSIハンドブック」, 第1版, オーム社, 1994年11月, pp. 523-525.

【非特許文献3】Kostas Pagiamtzis and Ali Sheikholeslami, "A Low-Power Content-Addressable Memory (CAM) Using Pipelined Hierarchical Search Scheme", IEEE Journal of Solid-State Circuits, Vol.39, No.9, Sept.2004, pp.1512-1519.

40

【非特許文献4】T.Sasao, M.Matsuura, and Y.Iguchi, "A cascade realization of multi-output function for reconfigurable hardware", International Workshop on Logic and Synthesis (IWLS01), Lake Tahoe, CA, June 12-15, 2001, pp.225-230.

【非特許文献5】T.Sasao, "Memory-Based Logic Synthesis", Springer 2011.

【非特許文献6】笹尾勤, "パターンマッチング用プログラマブル論理回路とその設計法", 電子情報通信学会誌, Vol. 96, No.2, pp.100-104, 2013年2月.

【発明の概要】

【発明が解決しようとする課題】

【0008】

上記従来のCAMは、RAMに比べると、並列に検索可能であるため高速であるが、デ

50

バイスの構成は複雑となる。そのため、C A Mの1ビットあたりの価格(ビットコスト)は、R A Mに比べると10~30倍程度、高価なものになる。また、1ビットあたりの消費電力がR A Mに比べて遙かに大きい(非特許文献3参照)。これは、すべてのC A Mセルを同時にアクセスするためである。そのため、1ビットあたりの消費電力は、通常のR A Mの約50倍程度にもなる。

従って、上記従来のC A Mによれば、ビットコスト及び消費電力の観点から記憶容量の大規模化は困難である。

【0009】

本発明は、上記事情に鑑みてなされたものであり、記憶容量の大規模化を可能としつつ、登録情報の更新を高速に行うことができる連想記憶装置、インデックス生成器、及び登録情報更新方法を提供することを目的とする。

10

【課題を解決するための手段】

【0010】

本発明の一態様による連想記憶装置は、複数のハッシュ関数を用いて入力ベクトルの特徴量を生成し、前記特徴量に基づき登録ベクトルに関する登録情報を検索することにより前記入力ベクトルに対応したインデックスを生成する複数のインデックス生成部と、前記複数のインデックス生成部の各出力を結合して前記入力ベクトルに対応したインデックスを出力する出力部と、前記登録情報の更新を制御する制御部と、を備え、前記複数のインデックス生成部のそれぞれは、前記特徴量に基づき生成されたインデックスが前記登録情報として存在するか否かを示す第1信号を生成して前記制御部に供給すると共に、前記特徴量に基づき生成されたインデックスを用いて前記入力ベクトルの再生ベクトルを生成し、前記再生ベクトルが前記入力ベクトルと一致するか否かを示す第2信号を生成して前記制御部に供給する、連想記憶装置の構成を有する。

20

【0011】

本発明の一態様によるインデックス生成器は、入力ベクトルから前記入力ベクトルの特徴量を算出して出力するハッシュ回路と、登録ベクトルに関する登録情報として前記登録ベクトルに対応するインデックスが格納された主記憶部を有し、前記主記憶部から前記特徴量に対応した仮のインデックスを読み出して出力する仮インデックス生成回路と、前記仮インデックス生成回路から出力された仮のインデックスに基づいて、前記特徴量に対応した仮のインデックスが前記登録情報として存在するか否かを示す第1信号を生成して出力する衝突信号生成回路と、前記登録ベクトルに関する登録情報として前記登録ベクトルが格納された副記憶部を有し、前記副記憶部から前記仮のインデックスに対応したベクトルを読み出して再生ベクトルとして出力する再生ベクトル生成回路と、前記再生ベクトルと前記入力ベクトルとを比較し、前記再生ベクトルと前記入力ベクトルとが一致するか否かを示す第2信号を出力する比較回路と、前記第2信号が前記再生ベクトルと前記入力ベクトルとの一致を示す場合、前記仮のインデックスを前記入力ベクトルに対応した固有のインデックスとして出力する出力回路と、を備えたインデックス生成器の構成を有する。

30

【0012】

本発明の一態様による情報更新方法は、前記連想記憶装置に登録された登録情報を更新する情報更新方法であって、前記連想記憶装置は、動作モードとして、検索モードと更新モードとを備え、前記連想記憶装置に備えられた制御部は、前記更新モードにおいて前記登録情報を更新するための制御を実施する、情報更新方法の構成を有する。

40

【発明の効果】

【0013】

本発明の一態様によれば、記憶容量の大規模化を可能としつつ、登録情報の更新を高速に行うことができる。

【図面の簡単な説明】

【0014】

【図1】本発明の第1実施形態による連想記憶装置の構成例を示すブロック図である。

【図2】本発明の第1実施形態による連想記憶装置に備えられたインデックス生成部の構

50

成例を示すブロック図である。

【図 3 A】本発明の第 1 実施形態による連想記憶装置に備えられた第 1 のインデックス生成部のハッシュ回路の構成例を示す回路図である。

【図 3 B】本発明の第 1 実施形態による連想記憶装置に備えられた第 2 のインデックス生成部のハッシュ回路の構成例を示す回路図である。

【図 3 C】本発明の第 1 実施形態による連想記憶装置に備えられた第 3 のインデックス生成部のハッシュ回路の構成例を示す回路図である。

【図 4 A】本発明の第 1 実施形態による連想記憶装置に備えられたインデックス生成部の比較回路の構成例を示す図である。

【図 4 B】本発明の第 1 実施形態による連想記憶装置に備えられたインデックス生成部の出力回路の構成例を示す図である。

【図 4 C】本発明の第 1 実施形態による連想記憶装置に備えられたインデックス生成部の衝突信号生成回路の構成例を示す図である。

【図 5】本発明の第 1 実施形態による連想記憶装置に備えられた出力部の回路構成例を示す図である。

【図 6】本発明の第 1 実施形態による連想記憶装置に登録される登録情報の一例を示す図である。

【図 7】本発明の第 1 実施形態による連想記憶装置に備えられた 1 番目のインデックス生成部で実現される登録情報を説明するための図である。

【図 8】本発明の第 1 実施形態による連想記憶装置に備えられた 2 番目のインデックス生成部で実現される登録情報を説明するための図である。

【図 9】本発明の第 1 実施形態による連想記憶装置に備えられた 3 番目のインデックス生成部で実現される登録情報を説明するための図である。

【図 10】本発明の第 2 実施形態による連想記憶装置の構成例を示すブロック図である。

【図 11】本発明の第 2 実施形態による連想記憶装置に備えられた連想メモリの基本構成の一例を示すブロック図である。

【図 12】CAMセルの回路構成例を示す図である。

【図 13】本発明の第 1 実施形態による登録情報の削除方法の一例を示すフローチャートである。

【図 14】本発明の第 1 実施形態による新規情報の追加方法の一例を示すフローチャートである。

【発明を実施するための形態】

【0015】

本発明の実施形態を説明する前に、本明細書において使用するいくつかの用語を定義しておく。

〔定義 1〕（登録ベクトル）

相異なる k 個 (k は自然数) の n ビット (n は自然数) のベクトルの集合を考える。このベクトルの集合を登録ベクトル集合 (set of registered vectors) といい、登録ベクトル集合に属する各ベクトルを登録ベクトルという。

【0016】

〔定義 2〕（インデックス生成関数，インデックス生成器，入力ベクトル）

登録ベクトル集合の各要素と一致する入力に対して 1 から k までの固有インデックスに単射し、それ以外の入力に対して 0 となる関数をインデックス生成関数という。

即ち、関数 $f(X) : B^n \rightarrow \{0, 1, \dots, k\}$ ($B = \{0, 1\}$, k 自然数) 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 $f(a_i) = i$ ($i = 1, 2, \dots, k$) が成立し、それ以外の ($2^n - k$) 個の入力ベクトル a_i に対しては、 $f(a_i) = 0$ が成立するとき、 $f(X)$ を重み k のインデックス生成関数という。インデックス生成関数は、 k 個の異なる 2 値ベクトルに対して、1 から k までの固有インデックスを生成する。なお、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。

10

20

30

40

50

【 0 0 1 7 】

また、上記のインデックス生成関数の演算を行う回路をインデックス生成器という。インデックス生成器に入力されるベクトルを、入力ベクトルという。入力ベクトルが n 次元ベクトルであるインデックス生成関数を、 n 入力のインデックス生成関数という。また、 n 入力のインデックス生成関数の演算を行う回路を n 入力のインデックス生成器という。

【 0 0 1 8 】

〔定義 3〕（ハッシュ関数）

ハッシュ関数とは、集合 S から整数の集合 $\{0, 1, \dots, m-1\}$ への写像である。ここで、 m は、集合 S の要素を超えない自然数である。

【 0 0 1 9 】

以下、図面を参照して、本発明の実施形態を説明する。

（第 1 実施形態）

1. 構成の説明

図 1 は、本発明の第 1 実施形態による連想記憶装置 100 の構成例を示すブロック図である。連想記憶装置 100 は、検索対象の n ビット（ n は自然数）の入力ベクトル X （ x_1, x_2, \dots, x_n ）に対応した固有のインデックス A を生成して出力するためのものであり、 N 個（ N は 2 以上の自然数）のインデックス生成部 110-1, 110-2, 110-3, ..., 110-N、出力部 120、制御部 130 を備えている。以下では必要に応じて、 N 個のインデックス生成部 110-1, 110-2, 110-3, ..., 110-N のうちの任意の一つを「インデックス生成部 110- i 」（ i は、1 から N までの自然数）と表す。また、インデックス生成部 110-1, 110-2, 110-3, ..., 110-N のそれぞれは、インデックス生成器、インデックス生成装置等として単体で取り扱うことができる。

【 0 0 2 0 】

N 個のインデックス生成部 110-1, 110-2, 110-3, ..., 110-N は、入力ベクトル X の特徴量を生成し、その特徴量から、入力ベクトル X に対応した固有のインデックス A_1, A_2, \dots, A_N をそれぞれ生成して出力するように構成されている。即ち、 N 個のインデックス生成部 110-1, 110-2, 110-3, ..., 110-N は、それぞれ、ハッシュ関数を用いて入力ベクトル X の特徴量を生成し、この特徴量に基づいて、各インデックス生成部に登録された登録ベクトルに関する登録情報を検索することにより、入力ベクトル X に対応した固有のインデックス $A_1, A_2, A_3, \dots, A_N$ を生成する。本実施形態では、後述するように、ハッシュ関数として、排他的論理和ゲート回路のみで構成された回路で実現できる線形関数を用いる。

なお、 N 個のインデックス生成部 110-1, 110-2, 110-3, ..., 110-N でそれぞれ用いられる N 個のハッシュ関数は、上述の〔定義 3〕に従う限り、相互に同一の関数であってもよく、相互に異なる関数であってもよい。即ち、 N 個のインデックス生成部 110-1, 110-2, 110-3, ..., 110-N のそれぞれは、他のインデックス生成部で用いられるハッシュ関数と同一のハッシュ関数を用いてもよく、他のインデックス生成部で用いられるハッシュ関数とは異なるハッシュ関数を用いてもよい。

【 0 0 2 1 】

また、インデックス生成部 110-1, 110-2, 110-3, ..., 110-N のそれぞれは、上記の特徴量に基づき生成されたインデックスが上記の登録情報として存在するか否かを示す第 1 信号（以下、「衝突信号」と称す。） $CD_1, CD_2, CD_3, \dots, CD_N$ を生成すると共に、上記の特徴量に基づき生成されたインデックスを用いて入力ベクトル X の再生ベクトルを生成し、再生ベクトルが入力ベクトル X と一致するか否かを示す第 2 信号（以下、「一致信号」と称す。） MT_1, MT_2, \dots, MT_N を生成する。ここで、「再生ベクトル」とは、入力ベクトル X の特徴量から得られるインデックスに基づいて元の入力ベクトル X を再現したベクトルを指す。衝突信号 CD （ $CD_1, CD_2, CD_3, \dots, CD_N$ ）と一致信号 MT （ MT_1, MT_2, \dots, MT_N ）は制御部 130 に供給される。

10

20

30

40

50

【 0 0 2 2 】

出力部 1 2 0 は、N 個のインデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3, ..., 1 1 0 - N の各出力 (インデックス A 1, A 2, A 3, ..., A N) を結合して入力ベクトル X に対応した固有のインデックス A を出力するように構成されている。

制御部 1 3 0 は、N 個のインデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3, ..., 1 1 0 - N のそれぞれに登録された上記の登録情報の更新を制御するためのものである。制御部 1 3 0 からインデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3, ..., 1 1 0 - N に対し、登録情報である登録ベクトルの書き込み動作を制御するためのライトイネーブル信号 W E と、登録ベクトルのインデックス I D X と、登録情報を書き込むべきインデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3, ..., 1 1 0 - N の何れか指定するための選択信号 S E L が供給される。

10

なお、制御部 1 3 0 は、指令信号 U P D が供給された場合に、動作モードを更新モードに変更する。また、制御部 1 3 0 は、指令信号 U P D とインデックス I D X とが供給された場合に、新規な登録情報を追加する動作を行う。また、制御部 1 3 0 は、指令信号 U P D のみが供給された場合に、既存の登録情報を削除する動作を行う。

また、制御部 1 3 0 は、例えば、更新モードなどの動作中に、動作中であることを示す信号 B S Y (ビジー) を出力する。また、制御部 1 3 0 は、例えば、新規な登録情報を追加する動作が失敗した場合に、失敗したことを示す信号 F A L (フェイル) を出力する。

【 0 0 2 3 】

ここでは、説明の簡略化のため、 $N = 3$ とし、図 1 に示す連想記憶装置 1 0 0 は、3 個のインデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3 を備えるものとして説明する。従って、 i は、1, 2, 3 のうちの任意の自然数を指し、インデックス生成部 1 1 0 - i は、インデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3 のうちの一つを表す。ただし、この例に限定されず、インデックス生成部の個数 N は任意に設定し得る。また、 $n = 6$ とし、インデックス生成部 1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3 には、それぞれ、6 ビットの入力ベクトル X ($x_1, x_2, x_3, x_4, x_5, x_6$) が入力されるものとする。

20

【 0 0 2 4 】

次に、インデックス生成部 1 1 0 - i 、出力部 1 2 0、制御部 1 3 0 の各構成の詳細について説明する。

30

図 2 は、本発明の第 1 実施形態による連想記憶装置 1 0 0 に備えられたインデックス生成部 1 1 0 - i の構成例を示すブロック図である。インデックス生成部 1 1 0 - i は、ハッシュ回路 1 1 1、仮インデックス生成回路 1 1 2、再生ベクトル生成回路 1 1 3、比較回路 1 1 4、出力回路 1 1 5、衝突信号生成回路 1 1 6 を備えている。このうち、ハッシュ回路 1 1 1 は、ハッシュ関数を用いて、 n ビットの入力ベクトル X から、入力ベクトル X の特徴量を表す p ビット (p は n よりも小さい自然数) のベクトル Y_1 を生成するためのものである。ここでは、 $p = 3$ とする。ハッシュ関数は、異なる入力に対して同じ値を与える場合があり、それをハッシュ衝突 (collision) と称す。本実施形態による連想記憶装置 1 0 0 では、或るハッシュ関数でハッシュ衝突が発生した場合、別のインデックス生成部を用いることによりハッシュ衝突を回避する。その詳細については後述する。

40

【 0 0 2 5 】

仮インデックス生成回路 1 1 2 は、入力ベクトル X の特徴量を表す p ビットのベクトル Y_1 に対応した q ビットの仮のインデックス A' を生成するものである。ここで、 $q = \lceil \log_2 (k_i + 1) \rceil$ であり、 k_i は、このインデックス生成部 1 1 0 - i で実現される登録ベクトルの数を表し、 $\lceil \log_2 (k_i + 1) \rceil$ は切り上げ関数を表す。仮インデックス生成回路 1 1 2 は、登録ベクトルのインデックスが登録情報として格納された主記憶部を有しており、主記憶部から入力ベクトル X の特徴量に対応した仮のインデックス A' を読み出して出力する。上記の主記憶部は、上記の特徴量を表すビット列がアドレス信号として入力される D R A M (Dynamic Random Access Memory) や S R A M (Static Random Access Memory) 等の汎用の半導体メモリから構成されている。ただし、上記の主記憶部

50

は、D R A M等の汎用メモリと同様に1回のアクセスでメモリセルアレイの一部のメモリセルを選択する方式を採用するメモリであれば、汎用メモリである必要はなく、専用に設計されたメモリであってもよい。

【0026】

再生ベクトル生成回路113は、仮のインデックスA'に対応した入力ベクトルXを再生するためのものであり、登録ベクトルが登録情報として格納された副記憶部を有している。再生ベクトル生成回路113は、副記憶部から仮のインデックスA'に対応した登録ベクトルを読み出して、(n-p)ビットの再生ベクトルX2'として出力する。(n-p)ビットの再生ベクトルX2'は、入力ベクトルX(X1, X2)のX2を再生した要素である。ここで、X1は入力ベクトルXのビットx1, x2, x3を指し、X2は、入力ベクトルXのビットx4, x5, x6を指す。上記の副記憶部は、上記の仮のインデックスA'を示すビット列がアドレス信号として入力されるD R A MやS R A M(Static Random Access Memory)等の汎用の半導体メモリから構成されている。上記の副記憶部についても、上記の主記憶部と同様に、専用に設計されたメモリであってもよい。

なお、上記のハッシュ関数として非線形関数を用いた場合、再生ベクトル生成回路113は、入力ベクトルXの全ビットに対応したnビットの再生ベクトルX'を出力する。

【0027】

比較回路114は、再生ベクトルX2'と入力ベクトルXのX2(x4, x5, x6)とが一致するか否かを検証するためのものである。再生ベクトルX2'と入力ベクトルXのX2とが一致するか否かを検証することは、仮のインデックスA'が入力ベクトルXに対応した固有のインデックスであるかどうかを検証することを意味する。比較回路114は、再生ベクトルX2'と入力ベクトルXのX2とを比較し、この比較の結果に基づき、再生ベクトルX'と入力ベクトルXのX2とが一致するか否かを示す一致信号MTiを出力する。

【0028】

出力回路115は、一致信号MTiに基づき、入力ベクトルXに対応した固有のインデックスAiまたは所定の無効値を出力するためのものである。即ち、出力回路115は、一致信号MTiが再生ベクトルX2'と入力ベクトルXのX2との一致を示す場合、仮のインデックスA'を固有のインデックスAiとして出力し、一致信号MTiが再生ベクトルX2'と入力ベクトルXのX2との不一致を示す場合、所定の無効値を出力する。本実施形態では、インデックスAiの全てのビットの値が'0'である場合、インデックスAiは所定の無効値を示す。ただし、この例に限定されず、無効値の定義は任意である。例えば、全てのビットの値が'1'であるインデックスAiを無効値としてもよい。

【0029】

衝突信号生成回路116は、更新モードにおいて、仮インデックス生成回路112から出力される仮のインデックスA'に基づいて衝突信号CDiを生成して出力するためのものである。第1実施形態では、衝突信号生成回路116は、更新モードにおいて、新規な登録ベクトルを登録情報として追加する場合、仮インデックス生成回路112から出力される仮のインデックスA'に基づいて衝突信号CDiを出力する。ここで、衝突信号CDiは、更新モードでハッシュ衝突の有無を示す信号であり、更新対象の入力ベクトルXに対して仮インデックス生成回路112から出力される仮のインデックスA'が無効値でない場合、即ちハッシュ衝突が発生した場合に値'1'となる信号である。なお、ハッシュ衝突が発生した場合の衝突信号CDiの値は任意に設定し得る。

【0030】

図3A~図3Cは、本発明の第1実施形態による連想記憶装置100に備えられたインデックス生成部110-iに備えられたハッシュ回路111の構成例を示す回路図である。ここで、図3Aは、インデックス生成部110-1に備えられたハッシュ回路111(i=1)の構成例を示し、図3Bは、インデックス生成部110-2に備えられたハッシュ回路111(i=2)の構成例を示し、図3Cは、インデックス生成部110-3に備えられたハッシュ回路111(i=3)の構成例を示している。図3A~図3Cでは、イ

10

20

30

40

50

インデックス生成部 110 - i に入力される入力ベクトル X として、6 ビット ($n = 6$) の入力ベクトル X ($x_1, x_2, x_3, x_4, x_5, x_6$) を想定している。

【0031】

図 3 A に示すハッシュ回路 111 ($i = 1$) は、インデックス生成部 110 - 1 において入力ベクトル X の特徴量を算出するためのハッシュ関数を実現する。図 3 A の例ではハッシュ回路 111 は、入力ベクトル X の 6 ビット ($x_1, x_2, x_3, x_4, x_5, x_6$) のうち、 $X_1 (x_1, x_2, x_3)$ の 3 ビットを、入力ベクトル X の特徴量を表す 3 ビット ($p = 3$) のベクトル $Y_1 (y_1, y_2, y_3)$ に変換するハッシュ関数を実現する。具体的には、図 3 A に示すハッシュ回路 111 は、3 個の排他的論理和ゲート回路から構成される。これら 3 個の排他的論理和ゲート回路の第 1 入力部には、 $X_1 (x_1, x_2, x_3)$ の各ビットがそれぞれ供給され、第 2 入力部には、値 '0' が共通に供給される。これにより、図 3 A のハッシュ回路 111 は、入力ベクトル X の $X_1 (x_1, x_2, x_3)$ を 3 ビットのベクトル $Y_1 (y_1, y_2, y_3)$ として出力する。

10

【0032】

図 3 B に示すハッシュ回路 111 ($i = 2$) は、インデックス生成部 110 - 2 において入力ベクトル X の特徴量を算出するためのハッシュ関数を実現する。図 3 B の例ではハッシュ回路 111 は、入力ベクトル X の 6 ビット ($x_1, x_2, x_3, x_4, x_5, x_6$) のうち、 $X_2 (x_4, x_5, x_6)$ の 3 ビットを、入力ベクトル X の特徴量を表す 3 ビット ($p = 3$) のベクトル $Y_1 (y_1, y_2, y_3)$ に変換するハッシュ関数を実現する。具体的には、図 3 B に示すハッシュ回路 111 は、3 個の排他的論理和ゲート回路から構成される。これら 3 個の排他的論理和ゲート回路の第 1 入力部には、 $X_2 (x_4, x_5, x_6)$ の各ビットがそれぞれ供給され、第 2 入力部には、値 '0' が共通に供給される。これにより、図 3 B のハッシュ回路 111 ($i = 2$) は、入力ベクトル X の $X_2 (x_4, x_5, x_6)$ を 3 ビットのベクトル $Y_1 (y_1, y_2, y_3)$ として出力する。

20

【0033】

図 3 C に示すハッシュ回路 111 ($i = 3$) は、インデックス生成部 110 - 3 において入力ベクトル X の特徴量を算出するためのハッシュ関数を実現する。図 3 C の例ではハッシュ回路 111 は、入力ベクトル X の 6 ビット ($x_1, x_2, x_3, x_4, x_5, x_6$) のうち、ビット x_1 とビット x_6 の排他的論理和を演算し、その演算結果を、入力ベクトル X の特徴量を表す 3 ビット ($p = 3$) のベクトル $Y_1 (y_1, y_2, y_3)$ として出力するハッシュ関数を実現する。

30

【0034】

具体的には、図 3 C に示すハッシュ回路 111 ($i = 3$) は、3 個の排他的論理和ゲート回路 111 - 31, 111 - 32, 111 - 33 から構成される。排他的論理和ゲート回路 111 - 31 の入力部には、入力ベクトル X のビット x_1 とビット x_6 が供給される。排他的論理和ゲート回路 111 - 31 によりビット x_1 とビット x_6 との排他的論理和が演算され、その演算結果はベクトル Y_1 のビット y_1 の値とされる。また、排他的論理和ゲート回路 111 - 32 の入力部には、入力ベクトル X のビット x_2 とビット x_5 が供給される。排他的論理和ゲート回路 111 - 32 によりビット x_2 とビット x_5 との排他的論理和が演算され、その演算結果はベクトル Y_1 のビット y_2 の値とされる。

40

【0035】

更に、排他的論理和ゲート回路 111 - 33 の入力部には、入力ベクトル X のビット x_3 とビット x_4 が供給される。排他的論理和ゲート回路 111 - 33 によりビット x_3 とビット x_4 との排他的論理和が演算され、その演算結果はベクトル Y_1 のビット y_3 の値とされる。

このように、3 個のハッシュ回路 111 ($i = 1$), 111 ($i = 2$), 111 ($i = 3$) は、それぞれ異なるハッシュ関数を実現している。

【0036】

インデックス生成部 110 - 1, 110 - 2, 110 - 3 の各ハッシュ回路 111 で実現されるハッシュ関数は、入力ベクトル X のビット数 n よりも少ないビット数 p で登録ベ

50

クトルを有意に識別し得る特徴量が得られるように、登録ベクトルの内容に応じて設定される。その詳細については後述する。

【 0 0 3 7 】

次に、図 2 に示すインデックス生成部 1 1 0 - i に備えられた仮インデックス生成回路 1 1 2 の構成を説明する。

仮インデックス生成回路 1 1 2 は、前述のように、D R A M 等の汎用の半導体メモリから構成され、この汎用の半導体メモリは仮インデックス生成回路 1 1 2 の上記の主記憶部を形成する。データの書き込みと読み出しの両方が可能であることを限度に、仮インデックス生成回路 1 1 2 の主記憶部を形成するメモリとして、強誘電体メモリや磁性体メモリ等、任意の方式のメモリを用いることができる。仮インデックス生成回路 1 1 2 の主記憶部を形成するメモリには、登録ベクトルのインデックスに関する情報が登録情報として格納されている。仮インデックス生成回路 1 1 2 の主記憶部を形成する汎用メモリのアドレス入力端子には、ハッシュ回路 1 1 1 から出力される p ビットのベクトル Y 1 の各ビットがアドレス信号として供給される。

10

【 0 0 3 8 】

ここで、ベクトル Y 1 の各ビットの値の組み合わせによって指定される上記の主記憶部を形成する汎用メモリのメモリセルには、登録ベクトルのインデックスに関する登録情報として、その特徴量に対応したインデックスを示すデータが格納されている。従って、ベクトル Y 1 に基づいて上記の主記憶部を形成する汎用メモリをアクセスすることにより、上記の主記憶部から入力ベクトル X の特徴量に対応した q ビットのインデックスが仮のインデックス A ' として読み出される。

20

【 0 0 3 9 】

次に、図 2 に示すインデックス生成部 1 1 0 - i に備えられた再生ベクトル生成回路 1 1 3 の構成を説明する。

再生ベクトル生成回路 1 1 3 は、上述の仮インデックス生成回路 1 1 2 と同様に、D R A M 等の汎用の半導体メモリから構成され、この汎用の半導体メモリは再生ベクトル生成回路 1 1 3 の上記の副記憶部を形成する。データの書き込みと読み出しの両方が可能であることを限度に、再生ベクトル生成回路 1 1 3 の副記憶部を形成するメモリとして、強誘電体メモリや磁性体メモリ等、任意の方式のメモリを用いることができる。再生ベクトル生成回路 1 1 3 の副記憶部を形成するメモリには、登録ベクトルのビットの値が登録情報として格納されている。再生ベクトル生成回路 1 1 3 の副記憶部を形成する汎用メモリのアドレス入力端子には、仮インデックス生成回路 1 1 2 から出力された q ビットの仮のインデックス A ' の各ビットがアドレス信号として供給される。

30

【 0 0 4 0 】

本実施形態では、ハッシュ回路 1 1 1 においてハッシュ関数として線形関数を用いていることから、後述の比較回路 1 1 4 において入力ベクトルと再生ベクトルとの一致を検証するのに一部のビットを比較すれば足りる。このため、仮のインデックス A ' の各ビットの値の組み合わせによって指定される上記の副記憶部を形成する汎用メモリのメモリセルには、登録ベクトルに関する登録情報として、仮のインデックス A ' に対応した登録ベクトルの一部のビットである (n - p) ビットの値が格納されている。ただし、上記のハッシュ関数として非線形関数を用いる場合、上記の副記憶部を形成するメモリには、登録ベクトルの全ビット (即ち n ビット) の値が登録情報として格納される。

40

【 0 0 4 1 】

ここで、仮のインデックス A ' は、入力ベクトル X に対応している。従って、仮のインデックス A ' に基づいて再生ベクトル生成回路 1 1 3 の副記憶部を形成する汎用メモリをアクセスすることにより、上記の副記憶部から入力ベクトル X の X 2 に対応した登録ベクトルの一部のビットが再生ベクトル X 2 ' として読み出される。

【 0 0 4 2 】

なお、仮インデックス生成回路 1 1 2 の主記憶部と、再生ベクトル生成回路 1 1 3 の副記憶部は、D R A M 等の 1 つの半導体メモリに統合されてもよい。これにより、構成部品

50

数を低減するとともに、メモリアクセス時間を削減することができる。

【0043】

次に、図2に示すインデックス生成部110-iに備えられた比較回路114の構成を説明する。

図4Aは、本発明の第1実施形態による連想記憶装置100に備えられたインデックス生成部110-iの比較回路114の構成例を示す図である。比較回路114は、一致ゲートである否定的排他的論理和ゲート回路1144, 1145, 1146から構成されている。否定的排他的論理和ゲート回路1144の入力部には、入力ベクトルXのX2のビットx4と再生ベクトルX2'のビットx4'が供給される。また、否定的排他的論理和ゲート回路1145の入力部には、入力ベクトルXのX2のビットx5と再生ベクトルX2'のビットx5'が供給される。更に、否定的排他的論理和ゲート回路1146の入力部には、入力ベクトルXのX2のビットx6と再生ベクトルX2'のビットx6'が供給される。否定的排他的論理和ゲート回路1144, 1145, 1146の各出力部は論理積ゲート回路1147の入力部に接続され、論理積ゲート回路1147の出力信号は一致信号MTiとされる。

10

【0044】

上記の比較回路114の構成によれば、入力ベクトルXのX2のビットx4, x5, x6の各値と再生ベクトルX2'のビットx4', x5', x6'の各値が完全に一致した場合のみ、一致信号MTiとして値'1'を示すハイレベルの論理信号が出力される。

【0045】

次に、図2に示すインデックス生成部110-iに備えられた出力回路115の構成を説明する。

図4Bは、本発明の第1実施形態による連想記憶装置100に備えられたインデックス生成部110-iの出力回路115の構成例を示す図である。この例では、k=15を考慮して、仮のインデックスA'が4ビット(a1', a2', a3', a4')で表される場合を想定している。ただし、この例に限定されず、仮のインデックスA'のビット数は任意の値に設定し得る。

20

【0046】

出力回路115は、仮のインデックスA'のビット数(4ビット)に対応する4個の論理積ゲート回路1151, 1152, 1153, 1154から構成されている。論理積ゲート回路1151, 1152, 1153, 1154の各第1入力部には、それぞれ、仮のインデックスA'のビットa1', a2', a3', a4'が供給される。また、論理積ゲート回路1151, 1152, 1153, 1154の各第2入力部には一致信号MTiが共通に供給される。

30

【0047】

上記の出力回路115の構成によれば、一致信号MTiの値が'1'(ハイレベル)である場合、論理積ゲート回路1151, 1152, 1153, 1154は、仮のインデックスA'のビットa1', a2', a3', a4'の値を、そのまま、固有のインデックスAiのビットa1i, a2i, a3i, a4iの値として出力する。また、一致信号MTiの値が'0'(ローレベル)である場合、論理積ゲート回路1151, 1152, 1153, 1154は、仮のインデックスA'のビットa1', a2', a3', a4'の値とは関係なく、値'0'(ローレベル)の信号を出力し、固有のインデックスAiのビットa1i, a2i, a3i, a4iの値はすべて'0'となる。これは無効値を示す信号である。

40

【0048】

次に、図2に示すインデックス生成部110-iに備えられた衝突信号生成回路116の構成を説明する。

図4Cは、本発明の第1実施形態による連想記憶装置100に備えられたインデックス生成部110-iの衝突信号生成回路116の構成例を示す図である。この例でも、k=15を考慮して、仮のインデックスA'が4ビット(a1', a2', a3', a4')

50

で表される場合を想定している。

【0049】

衝突信号生成回路116は、仮のインデックスA'のビット数(4ビット)に対応する4入力の論理和ゲート回路1161から構成されている。論理和ゲート回路1161は、仮のインデックスA'のビットa1', a2', a3', a4'の論理和を演算し、その演算結果を衝突信号CDiとして出力する。衝突信号生成回路116は、仮のインデックスA'のビットa1', a2', a3', a4'の何れかに'1'の値が含まれている場合、衝突信号CDiとして値'1'を示すハイレベルの論理信号を出力する。この衝突信号CDiの値'1'は、更新モードにおいてハッシュ衝突が発生していることを示す。これに対し、仮のインデックスA'のビットa1', a2', a3', a4'の全ての値が'0'である場合、即ち、仮のインデックスA'が無効値を示す場合、衝突信号生成回路116は、衝突信号CDiとして値'0'を示すローレベルの論理信号を出力する。この衝突信号CDiの値'0'は、更新モードにおいてハッシュ衝突が発生していないことを示す。

10

以上で、図1に示す連想記憶装置100に備えられたインデックス生成部110-iの構成を説明した。

【0050】

次に、図1に示す連想記憶装置100に備えられた出力部120の構成を説明する。

図5は、本発明の第1実施形態による連想記憶装置100に備えられた出力部120の回路構成例を示す図である。

20

出力部120は、インデックス生成部110-1, 110-2, 110-3の個数に対応した3個の論理和ゲート回路1201, 1202, 1203から構成されている。論理和ゲート回路1201の入力部には、インデックス生成部110-1から出力されたインデックスA1のビットa1と、インデックス生成部110-2から出力されたインデックスA2のビットa1と、インデックス生成部110-3から出力されたインデックスA3のビットa1とが供給される。論理和ゲート回路1201は、インデックス生成部110-1, 110-2, 110-3からそれぞれ入力されたビットa1の論理和を演算し、その演算結果を、インデックスAのビットa1の値として出力する。

【0051】

また、論理和ゲート回路1202の入力部には、インデックス生成部110-1から出力されたインデックスA1のビットa2と、インデックス生成部110-2から出力されたインデックスA2のビットa2と、インデックス生成部110-3から出力されたインデックスA3のビットa2とが供給される。論理和ゲート回路1202は、インデックス生成部110-1, 110-2, 110-3からそれぞれ入力されたビットa2の論理和を演算し、その演算結果を、インデックスAのビットa2の値として出力する。

30

【0052】

更に、論理和ゲート回路1203の入力部には、インデックス生成部110-1から出力されたインデックスA1のビットa3と、インデックス生成部110-2から出力されたインデックスA2のビットa3と、インデックス生成部110-3から出力されたインデックスA3のビットa3とが供給される。論理和ゲート回路1203は、インデックス生成部110-1, 110-2, 110-3からそれぞれ入力されたビットa3の論理和を演算し、その演算結果を、インデックスAのビットa3の値として出力する。

40

【0053】

上記の論理演算を実施することにより、出力部120は、インデックス生成部110-1, 110-2, 110-3から出力されたインデックスA1, A2, A3の対応ビットを結合してインデックスAを生成し出力する。ここで、上記論理演算の過程で、回路の構成法から、高々1個のインデックス生成部のみが有効値を生成し、他のインデックス生成部は無効値を生成する。従って、上記構成によれば、有効値のみが出力部120からインデックスAとして出力される。

なお、全てのビットが'1'からなる仮のインデックスA'を無効値として定義した場

50

合、出力部 120 は、論理和ゲート回路 1201, 1202, 1203 に代えて、論理積演算を実施する論理積ゲート回路を備えればよい。

【0054】

次に、図 6 から図 10 を参照して、連想記憶装置 100 で実現される登録情報を説明する。

図 6 は、本発明の第 1 実施形態による連想記憶装置 100 に登録される登録情報の一例を示す図である。図 6 の登録情報は、 $n = 6$ 、 $k = 15$ 、 $X1 = (x1, x2, x3)$ 、 $X2 = (x4, x5, x6)$ としたときに、入力ベクトル $X(x1, x2, x3, x4, x5, x6)$ の各値と、入力ベクトル X の固有のインデックス A を表すインデックス IDX の値との対応関係を規定している。例えば、入力ベクトル $X(x1, x2, x3, x4, x5, x6)$ の値 $(0, 1, 1, 0, 0, 0)$ に対し、この入力ベクトル X の値に固有のインデックス A を表すインデックス IDX の値として値「1」が対応付けられている。

【0055】

前述したインデックス生成部 110-1, 110-2, 110-3 の各ハッシュ回路 111 で実現されるハッシュ関数は、入力ベクトル X のビット数 n よりも少ない p ビットのベクトル $Y1$ によって、登録情報に登録された登録ベクトルを有意に識別する特徴量を表現できるように、登録ベクトルの内容に応じて設定される。図 6 の例では、例えば $X1$ のビット $x1$ の値に着目した場合、インデックス IDX の値「1」及び「2」にそれぞれ対応付けられている登録ベクトルのビット $x1$ の値は共に「0」である。従って、ビット $x1$ のみに基づいて、インデックス IDX の値「1」及び「2」にそれぞれ対応付けられている登録ベクトルを識別することはできない。そこで、ビット $x2$ に着目すると、インデックス IDX の値「1」及び「2」にそれぞれ対応付けられている登録ベクトルのビット $x2$ の値は共に「1」である。従って、ビット $x1$ とビット $x2$ の二つのビットに着目しても、インデックス IDX の値「1」及び「2」にそれぞれ対応付けられている登録ベクトルを識別することはできない。

【0056】

そこで、更にビット $x3$ に着目すると、インデックス IDX の値「1」に対応付けられている登録ベクトルのビット $x3$ の値は「1」であり、インデックス IDX の値「2」に対応付けられている登録ベクトルのビット $x3$ の値は「0」である。従って、ビット $x3$ に着目すれば、インデックス IDX の値「1」及び「2」にそれぞれ対応付けられている登録ベクトルを識別することができる。即ち、インデックス IDX の値「1」及び「2」に対応する登録ベクトルの特徴量は、 $X1$ のビット $x1, x2, x3$ を用いて表現可能である。

ただし、この例では、ビット $x2, x3$ のみからインデックス IDX の値「1」及び「2」に対応する登録ベクトルを識別することが可能であり、この場合の登録ベクトルの特徴量としてビット $x2, x3$ の値を用いることもできる。このようにして、 $X1$ によって識別可能な登録ベクトルに対応したインデックス IDX の集合が得られる。

【0057】

第 1 実施形態では、上述の図 6 に示す登録情報のうち、 $X1$ を用いて識別可能な登録ベクトルとインデックス IDX との対応を規定した登録情報は、3 個のインデックス生成部 110-1, 110-2, 110-3 のうち、1 番目のインデックス生成部 110-1 で実現される。

【0058】

図 7 は、本発明の第 1 実施形態による連想記憶装置 100 に備えられた 1 番目のインデックス生成部 110-1 で実現される登録情報を説明するための図である。

図 7 に示すように、インデックス生成部 110-1 で実現される登録情報は、上述の図 6 に示す登録情報のうち、インデックス IDX の 8 個の値「4」、「5」、「2」、「1」、「11」、「10」、「15」、「3」と、これらのインデックス IDX の値に対応した 8 個の登録ベクトルのビット $(x1, x2, x3, x4, x5, x6)$ の値との対応関係を規定するように決定される。

【 0 0 5 9 】

このようにインデックス生成部 1 1 0 - 1 で実現すべき登録情報が決定されると、この登録情報の登録ベクトルを有意に識別するために必要なビット x_1, x_2, x_3 が入力ベクトル X から得られるように、インデックス生成部 1 1 0 - 1 のハッシュ回路 1 1 1 で実現すべきハッシュ関数が設定される。そして、このようなハッシュ関数を実現するために、前述したように、インデックス生成部 1 1 0 - 1 のハッシュ回路 1 1 1 ($i = 1$) は、入力ベクトル X のビット $x_1, x_2, x_3, x_4, x_5, x_6$ のうち、 $X_1 (x_1, x_2, x_3)$ をベクトル $Y_1 (y_1, y_2, y_3)$ として仮インデックス生成回路 1 1 2 に供給するように構成されている。

【 0 0 6 0 】

ここで、図 7 に示す登録情報のうち、登録ベクトルのビット x_1, x_2, x_3 に対応付けられているインデックス IDX の各値は、インデックス生成部 1 1 0 - 1 の仮インデックス生成回路 1 1 2 の主記憶部に仮のインデックス A' として格納されている。この場合、インデックス IDX を表す仮のインデックス A' は、そのインデックス IDX に対応する登録ベクトルのビット x_1, x_2, x_3 によって指定されるメモリセルに格納されている。このように、登録ベクトルとインデックス IDX とが対応付けられて登録情報として仮インデックス生成回路 1 1 2 の主記憶部に格納されている。

【 0 0 6 1 】

また、図 7 に示す登録情報のうち、登録ベクトルのビット $x_1, x_2, x_3, x_4, x_5, x_6$ の各値は、再生ベクトル生成回路 1 1 3 の副記憶部に格納される。即ち、仮インデックス生成回路 1 1 2 から q ビットの仮のインデックス A' としてインデックス IDX が再生ベクトル生成回路 1 1 3 に供給され、この仮のインデックス A' によって指定される副記憶部のメモリセルには、登録ベクトルのビット $x_1, x_2, x_3, x_4, x_5, x_6$ の各値が格納されている。このように、登録ベクトルの各ビットの値が、仮のインデックス A' に対応付けられた登録情報として再生ベクトル生成回路 1 1 3 の副記憶部に格納されている。ただし、前述したように、上記のハッシュ関数として線形関数を用いた場合、登録ベクトルの全ビットを再生ベクトル生成回路 1 1 3 の副記憶部に格納する必要はなく、登録ベクトルの一部のビット x_4, x_5, x_6 を再生ベクトル生成回路 1 1 3 の副記憶部に格納すれば足りる。

【 0 0 6 2 】

上述のインデックス生成部 1 1 0 - 1 は、検索モードにおいて、入力ベクトル X が供給されると、入力ベクトル X の全ビットのうち、ビット x_4, x_5, x_6 の値が、再生ベクトル生成回路 1 1 3 の副記憶部を構成するメモリに貯えられた値と一致する場合、比較回路 1 1 4 の一致信号 MT_1 の値が ' 1 ' となり、インデックス IDX の値がインデックス A_1 として出力される。また、入力ベクトル X のビット x_4, x_5, x_6 の値が上記の副記憶部を構成するメモリに貯えられた値と一致しない場合、比較回路 1 1 4 の一致信号 MT_1 の値が ' 0 ' となり、インデックス IDX の値を表すインデックス A_1 として無効値が出力される。この場合、インデックス生成部 1 1 0 - 1 から出力されるインデックス A_1 の全ビットが ' 0 ' となる。

【 0 0 6 3 】

上述したように、インデックス生成部 1 1 0 - 1 では、上述の図 6 に示す登録情報のうち、インデックス IDX の 8 個の値「 4 」、「 5 」、「 2 」、「 1 」、「 1 1 」、「 1 0 」、「 1 5 」、「 3 」と、これらのインデックス IDX の値に対応した登録ベクトルとの対応関係を規定する登録情報が実現される。しかしながら、インデックス IDX の残りの 7 個の値「 7 」、「 9 」、「 1 3 」、「 1 4 」、「 1 2 」、「 6 」、「 8 」をインデックス生成部 1 1 0 - 1 では実現できない。

【 0 0 6 4 】

そこで、続いて、2 番目のインデックス生成部 1 1 0 - 2 では、登録情報の登録ベクトルのビット x_4, x_5, x_6 に着目して、上述の図 6 に示す登録情報のうち、インデックス IDX の 5 個の値「 7 」、「 1 3 」、「 1 2 」、「 6 」、「 8 」と、これらのインデッ

10

20

30

40

50

クスID Xの値に対応した登録ベクトルとの対応関係を規定する登録情報が実現される。この場合、登録ベクトルのX 2のビットx 4, x 5, x 6を用いて、インデックス生成部1 1 0 - 2で実現する登録情報のインデックスID Xが決定される。

【0 0 6 5】

図8は、本発明の第1実施形態による連想記憶装置1 0 0に備えられた2番目のインデックス生成部1 1 0 - 2で実現される登録情報を説明するための図である。この例では、上述の図6に示す登録情報のうち、インデックスID Xの5個の値「7」、「1 3」、「1 2」、「6」、「8」に関する登録情報が実現されている。しかしながら、インデックスID Xの残りの2個の値「9」、「1 4」をインデックス生成部1 1 0 - 2では実現できない。

10

なお、図8に示す例では、入力ベクトルXの全ビットのうち、ビットx 1, x 2, x 3の値が再生ベクトル生成回路1 1 3の副記憶部を構成するメモリに貯えられた値と一致する場合、比較回路1 1 4の一致信号MT 2の値が「1」となる。

【0 0 6 6】

そこで、最後に、3番目のインデックス生成部1 1 0 - 3では、図3 Cを参照して説明したように、登録情報の登録ベクトルのビットx 1, x 2, x 3とビットx 4, x 5, x 6との排他的論理和の演算結果として得られるビットy 1, y 2, y 3の値によって指定される上記の主記憶部及び副記憶部の各メモリセルに、残りのインデックスID Xの値「9」、「1 4」に関する登録情報が格納される。この場合、登録情報の登録ベクトルのビットx 1, x 2, x 3とビットx 4, x 5, x 6との排他的論理和の演算結果として得られるビットy 1, y 2, y 3を用いて、インデックス生成部1 1 0 - 3で実現する登録情報のインデックスID Xが決定される。

20

【0 0 6 7】

図9は、本発明の第1実施形態による連想記憶装置1 0 0に備えられた3番目のインデックス生成部1 1 0 - 3で実現される登録情報を説明するための図である。

この例では、上述の図6に示す登録情報のうち、インデックスID Xの5個の値「9」、「1 4」に関する登録情報が実現されている。この場合、登録情報のうち、インデックスID Xの値「9」、「1 4」に関する情報は、ビットy 1, y 2, y 3の値によって指定される仮インデックス生成回路1 1 2の主記憶部のメモリセルに格納される。また、インデックスID Xの値「9」、「1 4」に対応した登録ベクトルのビット(y 1, y 2, y 3, x 4, x 5, x 6)の各値は、インデックスID Xの値「9」、「1 4」を表す仮のインデックスA'によって指定される再生ベクトル生成回路1 1 3の副記憶部のメモリセルに格納される。

30

なお、図9に示す例では、入力ベクトルXの全ビットのうち、ビットx 4, x 5, x 6の値が再生ベクトル生成回路1 1 3の副記憶部を構成するメモリに貯えられた値と一致する場合、比較回路1 1 4の一致信号MT 3の値が「1」となる。

【0 0 6 8】

このように、3個のインデックス生成部1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3を用いることにより、それぞれにおいてハッシュ衝突が発生したとしても、図6に示す1 5個のインデックスに関する登録情報の全てを実現することができる。

40

【0 0 6 9】

2. 動作の説明

次に、第1実施形態による連想記憶装置1 0 0の通常の検索モードでの動作を説明する。通常の検索モードでの動作時には、入力ベクトルXに回答して、3個のインデックス生成部1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3のうち、一つのインデックス生成部1 1 0 - iが有効値を示すインデックスA iを出力する。他の二つのインデックス生成部1 1 0 - iは無効値を出力する。出力部1 2 0は、3個のインデックス生成部1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3からそれぞれ出力されたインデックスA 1, A 2, A 3を結合してインデックスAを生成して出力する。このインデックスAは、3個のインデックス生成部1 1 0 - 1, 1 1 0 - 2, 1 1 0 - 3から出力されたインデックスA 1, A 2, A 3のうち、有

50

効値を示すインデックス A_i と同値になる。入力ベクトル X がインデックス生成部 110-1, 110-2, 110-3 に設定された登録ベクトルのいずれとも一致しない場合は、すべてのインデックス生成部 110-1, 110-2, 110-3 が無効値を生成する。

【0070】

以上により、通常の検索モードでは、3個のインデックス生成部 110-1, 110-2, 110-3 の何れかから、入力ベクトル X に対応した固有のインデックス A が得られ、このインデックス A が出力部 120 から出力される。従って、検索モードにおいて、3個のインデックス生成部 110-1, 110-2, 110-3 と出力部 120 とが一体となって、一つの連想メモリとして機能する。

10

【0071】

次に、連想記憶装置 100 の更新モードでの動作を説明する。

更新モードでの動作として、既存の登録情報を削除する動作と、新規な登録情報を追加する動作がある。更新モードにおいて、制御部 130 は、登録情報として登録された既存の登録ベクトルが存在する場合、その既存の登録ベクトルに対応するインデックスを削除した後、登録情報として新規な登録ベクトルのインデックスを追加することにより、登録情報を更新する。更新モードにおいて登録情報を更新するための動作は制御部 130 により制御される。

【0072】

2-1. 既存の登録情報を削除する動作

20

ここでは、図13を参照して、上述した制御部 130 による既存の登録情報を削除する動作の一例について説明する。

図13は、本発明の第1実施形態による登録情報の削除方法の一例を示すフローチャートである。

既存の登録情報を削除する場合、連想記憶装置 100 を管理するシステム（図示なし）から、連想記憶装置 100 の制御部 130 に、更新モードの起動を示す指令信号 UPD が供給される。そして、図13に示すように、指令信号 UPD が制御部 130 に供給されると、制御部 130 は、削除すべき登録ベクトルを入力ベクトルとして、複数のインデックス生成部 110- i に供給する（ステップ $S101$ ）。すなわち、3個のインデックス生成部 110-1, 110-2, 110-3 には、削除すべき登録ベクトルが入力ベクトル X として共通に供給される。

30

【0073】

次に、各インデックス生成部 110- i は、該当する登録ベクトルが存在すれば、一致信号 $MT_i = '1'$ を出力する。また、各インデックス生成部 110- i は、該当する登録ベクトルが存在しなければ、一致信号 $MT_i = '0'$ を出力する（ステップ $S102$ ）。すなわち、インデックス生成部 110-1, 110-2, 110-3 は、それぞれ、該当する登録情報が存在すれば、一致信号 MT_1, MT_2, MT_3 として値 '1' の信号を制御部 130 に供給し、該当する登録情報が存在しなければ、一致信号 MT_1, MT_2, MT_3 として値 '0' の信号を制御部 130 に供給する。

【0074】

40

次に、制御部 130 は、一致信号 MT_i を参照して、削除すべき登録ベクトルが存在するインデックス生成部 110- i を特定する（ステップ $S103$ ）。すなわち、制御部 130 は、インデックス生成部 110-1, 110-2, 110-3 からそれぞれ供給される一致信号 MT_1, MT_2, MT_3 の値を参照して、削除すべき登録ベクトルが存在するインデックス生成部を特定する。

ここで、削除すべき登録ベクトルが存在すると、衝突が発生するので、衝突信号 CD_i および一致信号 MT_i が値 '1' を示す。この一致信号 MT_i から削除すべき登録ベクトルに対応するインデックス生成部を特定することができる。

【0075】

次に、制御部 130 は、特定されたインデックス生成部 110- i があるか否かを判定

50

する(ステップS104)。制御部130は、特定されたインデックス生成部110-iがある場合(ステップS104: YES)に、処理をステップS105に進める。また、制御部130は、特定されたインデックス生成部110-iがない場合(ステップS104: NO)に、処理を終了する。

【0076】

ステップS105において、制御部130は、削除すべき登録ベクトルが存在するインデックス生成部110-iに対して、WE = '1'として、既存の登録ベクトルのインデックスの値を無効値に変更する。すなわち、制御部130は、書き込み動作を制御するためのライトイネーブル信号WEにより、3個のインデックス生成部110-1, 110-2, 110-3のうち、削除すべき登録インデックスに対応するインデックス生成部の主記憶部及び副記憶部をライトモードに設定し、これらの記憶部に格納された登録情報を削除する。具体的には、削除すべき登録インデックスが格納されたインデックス生成部110-iの主記憶部に格納された仮のインデックスの値を有効値から無効値に書き換える。例えば、インデックス生成部110-1の登録ベクトル(x_1, x_2, x_3) = (0, 1, 1)に対するインデックスIDXを無効値に書き換える場合、インデックス生成部110-1の主記憶部を構成するメモリのアドレス(0, 1, 1)におけるインデックスIDXの内容を(0, 0, 0)に書き換える。これにより、システムから制御部130に供給された登録ベクトルに関する登録情報が削除された状態になる。

【0077】

上述の登録情報の削除に関する動作をまとめると、制御部130は、登録情報として登録された既存の登録ベクトルに対応するインデックスを削除する場合、複数のインデックス生成部110-1, 110-2, 110-3のうち、削除すべき登録ベクトルのインデックスを用いて生成された再生ベクトル X_2' が入力ベクトルXの X_2 と一致する旨を示す一致信号MTiを供給するインデックス生成部を選択する。制御部130は、選択した当該インデックス生成部の登録情報から、既存の登録ベクトルのインデックスを削除する。

【0078】

更新モードにおいて登録情報を削除する場合、制御部130による制御は次の第1段階から第3段階を含む。

第1段階：更新モードにおいて、削除すべき既存の登録ベクトルを入力ベクトルXとして複数のインデックス生成部110-1, 110-2, 110-3に供給する。

第2段階：複数のインデックス生成部110-1, 110-2, 110-3のそれぞれについて、制御部130は、一致信号MTiに基づき、削除すべき既存の登録ベクトルに対応した固有のインデックスが登録情報として存在するか否かを判定する。

第3段階：複数のインデックス生成部110-1, 110-2, 110-3の何れかにおいて、既存の登録ベクトルに対応した固有のインデックスが登録情報として存在する場合、制御部130は、登録情報として存在する既存の登録ベクトルのインデックスの値を有効値から無効値に変更する。

【0079】

このように、既存の登録情報を削除する場合、制御部130が、一致信号MTiの値から、削除すべき既存の登録ベクトルが格納されているインデックス生成部を特定し、制御部130が各インデックス生成部の主記憶部と副記憶部を直接的にアクセスして登録情報を削除するので、登録情報を高速に削除することができる。従って、既存の登録情報の削除を伴う更新動作を高速に実施することが可能になる。

【0080】

2-2. 新規な登録情報を追加する動作

ここでは、図14を参照して、上述した制御部130による新規な登録情報を追加する動作の一例について説明する。

図14は、本発明の第1実施形態による新規情報の追加方法の一例を示すフローチャートである。

10

20

30

40

50

新規な登録ベクトルを追加する場合、連想記憶装置100を管理するシステムから、連想記憶装置100の制御部130に、更新モードの起動を示す指令信号UPDが供給される。また、同システムから、追加すべき登録ベクトルのインデックスIDXが供給される。そして、図14に示すように、指令信号UPDとインデックスIDXとが制御部130に供給されると、制御部130は、追加登録ベクトルを入力ベクトルとして、複数のインデックス生成部110-iに供給する(ステップS201)。すなわち、3個のインデックス生成部110-1, 110-2, 110-3には、追加すべき登録ベクトルが入力ベクトルXとして供給される。

【0081】

次に、各インデックス生成部110-iは、該当する登録と衝突するベクトルが存在すれば、衝突信号CDi = '1'を出力する。また、各インデックス生成部110-iは、該当する登録と衝突するベクトルが存在しなければ、衝突信号CDi = '0'を出力する(ステップS202)。すなわち、インデックス生成部110-1, 110-2, 110-3は、それぞれ、該当する登録情報の仮のインデックスA'が存在すれば、衝突信号CD1, CD2, CD3として値'1'の信号を制御部130に供給し、該当する登録情報が存在しなければ、衝突信号CD1, CD2, CD3として値'0'の信号を制御部130に供給する。

【0082】

次に、制御部130は、衝突信号CDiを参照して、追加登録すべきベクトルと衝突する登録ベクトルが存在しないインデックス生成部110-iを特定する(ステップS203)。すなわち、制御部130は、インデックス生成部110-1, 110-2, 110-3からそれぞれ供給される衝突信号CD1, CD2, CD3の値を参照して、追加すべき登録ベクトルを登録可能なインデックス生成部を特定する。ここで、追加すべき登録ベクトルと衝突する登録ベクトルが存在すると、ハッシュ衝突が発生するので、衝突信号CDiが値'1'を示す。この衝突信号CDiの値から登録ベクトルを追加可能なインデックス生成部を特定することができる。

【0083】

次に、制御部130は、特定されたインデックス生成部110-iがあるか否かを判定する(ステップS204)。制御部130は、特定されたインデックス生成部110-iがある場合(ステップS204: YES)に、処理をステップS205に進める。また、制御部130は、特定されたインデックス生成部110-iがない場合(ステップS204: NO)に、処理をステップS206に進める。

【0084】

ステップS205において、制御部130は、追加登録すべきインデックス生成部110-iに対して、WE = '1'として、主記憶部に追加登録すべきベクトルのインデックスの値を書き込む。すなわち、制御部130は、ライトイネーブル信号WEにより、3個のインデックス生成部110-1, 110-2, 110-3のうち、追加すべき登録ベクトルと衝突する登録ベクトルが存在しないインデックス生成部の主記憶部及び副記憶部をライトモードに設定し、これらの記憶部に新規な登録情報を追加する。具体的には、追加すべき登録ベクトルと衝突する登録ベクトルが存在しないインデックス生成部の主記憶部に仮のインデックスの有効値を書き込む。例えば、インデックス生成部110-3に追加すべき登録ベクトルと衝突する登録ベクトルが存在しない場合、インデックス生成部110-3の主記憶部を構成するメモリのアドレス(y1, y2, y3) = (1, 1, 0)にインデックスの有効値を書き込む。これにより、システムから制御部130に供給された登録ベクトルに関する登録情報が連想記憶装置100に追加された状態になる。

【0085】

また、ステップS206において、制御部130は、追加不可を示す信号FALを出力する。すなわち、制御部130は、3個のインデックス生成部110-1, 110-2, 110-3の全てが、追加すべき登録ベクトルと衝突する登録ベクトルが存在する場合に、新規な登録情報の追加を行わない。そして、この場合、制御部130は、追加不可を示

10

20

30

40

50

す信号 F A L を出力する。

【 0 0 8 6 】

上述の登録情報の追加に関する動作をまとめると、制御部 1 3 0 は、登録情報として新規な登録ベクトルのインデックスを追加する場合、複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 のうち、新規な登録ベクトルが登録情報として格納されていない旨を示す衝突信号 C D i を供給するインデックス生成部を選択する。制御部 1 3 0 は、選択した当該インデックス生成部の登録情報として、上記の追加すべき新規な登録ベクトルのインデックスを追加する。また、複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 の何れかにおいて、追加すべき新規な登録ベクトルに対応した仮のインデックスが登録情報として格納されている場合、制御部 1 3 0 は、新規な登録ベクトルに対応した仮のインデックスが存在しない他のインデックス生成部において、新規な登録ベクトルに対応したインデックスの値として有効値を設定する。

10

【 0 0 8 7 】

更新モードにおいて登録情報を追加する場合、制御部 1 3 0 による制御は次の第 1 段階から第 3 段階を含む。

第 1 段階：追加すべき新規な登録ベクトルを入力ベクトル X として複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 に供給する。

第 2 段階：複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 のそれぞれについて、制御部 1 3 0 により、衝突信号 C D i に基づき、追加すべき新規な登録ベクトルに対応した仮のインデックスが登録情報として存在するか否かを判定する。

20

第 3 段階：複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 の何れかにおいて、新規な登録ベクトルに対応した仮のインデックスが登録情報として存在しない場合、新規な登録ベクトルに対応したインデックスの値として有効値を設定する。

【 0 0 8 8 】

このように、新規な登録情報を追加する場合、制御部 1 3 0 が、一致信号 M T i の値から、追加すべき登録ベクトルと衝突しないインデックス生成部を特定し、制御部 1 3 0 が各インデックス生成部の主記憶部と副記憶部を直接的にアクセスして登録情報を追加するので、登録情報を高速に追加することができる。従って、新規な登録情報の追加を伴う更新動作を高速に実施することが可能になる。

【 0 0 8 9 】

30

上述したように、第 1 実施形態によれば、複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 において、複数のハッシュ関数を用いて入力ベクトルの特徴量を抽出し、この特徴量に基づいて仮のインデックス A ' を生成するようにしたので、ハッシュ関数に起因したハッシュ衝突が発生しても、このハッシュ衝突を回避しつつ、汎用の半導体メモリを用いて登録情報を連想記憶装置 1 0 0 上に実現することができる。

【 0 0 9 0 】

また、複数のインデックス生成部 1 1 0 - 1 , 1 1 0 - 2 , 1 1 0 - 3 , ... , 1 1 0 - N のそれぞれに備えられたハッシュ回路 1 1 1 が入力ベクトル X のビット数 n よりも小さい p ビットのベクトル Y 1 を生成するようにしたので、ベクトル Y 1 がアドレスとして入力される仮インデックス生成回路 1 1 2 の主記憶部を構成するメモリのアドレス空間を小さくすることができる。従って、上記の主記憶部を構成する半導体メモリの記憶容量を削減することができる。

40

【 0 0 9 1 】

更に、ハッシュ回路 1 1 1 がハッシュ関数として線形関数を用いて入力ベクトル X の特徴量を抽出するようにしたので、再生ベクトル生成回路 1 1 3 の副記憶部には、登録ベクトルのビットの値として、登録ベクトルの一部の (n - p) ビットの値を格納しておけばよい。従って、再生ベクトル生成回路 1 1 3 の副記憶部を構成する半導体メモリの記憶容量を有効に削減することができる。

従って、第 1 実施形態によれば、ビットコストと消費電力の上昇を抑制しつつ、連想記憶装置 1 0 0 の記憶容量を大規模化することが可能となる。

50

【 0 0 9 2 】

また、第1実施形態によれば、複数のインデックス生成部110-1, 110-2, 110-3のそれぞれにおいて、衝突信号CD_iと一致信号MT_iを発生させ、これらの信号を制御部130に供給するようにしたので、更新モードにおいて、制御部130は、衝突信号CD_iと一致信号MT_iを参照することにより、削除すべき登録情報が存在するインデックス生成部と、登録情報を追加可能なインデックス生成部とを直接的にアクセスすることにより登録情報を更新することができる。従って、登録情報の更新を高速化することが可能になる。

【 0 0 9 3 】

(第2実施形態)

次に、本発明の第2実施形態を説明する。

図10は、本発明の第2実施形態による連想記憶装置200の構成例を示すブロック図である。

【 0 0 9 4 】

連想記憶装置200は、上述の第1実施形態による図1に示す連想記憶装置100の構成において、複数のインデックス生成部110-1, 110-2, 110-3, ..., 110-Nに加えて、または、複数のインデックス生成部110-1, 110-2, 110-3, ..., 110-Nのうちの一部のインデックス生成部に代えて、連想メモリ110-Cを備えている。連想メモリ110-Cは、CAMセルを有する内容検索型の半導体メモリであり、例えば従来型のCAMを用いることができる。連想メモリ110-Cの動作(検索モード及び更新モードでの動作)は、制御部130により制御される。

【 0 0 9 5 】

また、連想記憶装置200は、出力結合部220を更に備えている。出力結合部220は、第1実施形態による連想記憶装置100に備えられた出力部120から出力されるインデックスAと、連想メモリ110-Cから出力されるインデックスBとを結合して一つのインデックスCとして出力するためのものである。出力結合部220は、出力部120から出力されるインデックスAと、連想メモリ110-Cから出力されるインデックスBとのいずれか一方のインデックスを出力する。出力結合部220は、例えば、図5に示す出力部120と同様に、論理和ゲート回路から構成されている。その他は、第1実施形態と同様である。

【 0 0 9 6 】

次に、図11及び図12を参照して、連想記憶装置200に備えられた連想メモリ110-Cについて説明する。

図11は、本発明の第2実施形態による連想記憶装置200に備えられた連想メモリ110-Cとして用いられる従来型のCAM1000の基本構成の一例を表すブロック図である(特許文献1参照)。CAM1000は、比較レジスタ1001、検索ビット線ドライバ1002、m個のワードW₁~W_m、m個の一致センス回路MSC₁~MSC_m、m個の一致フラグレジスタMFR₁~MFR_m、及びプライオリティ・エンコーダ(優先度付符号化回路)PEを備えている。

【 0 0 9 7 】

比較レジスタ1001は、nビットの検索データを格納するレジスタである。検索ビット線ドライバ1002は、比較レジスタ1001の各ビットを検索ビット線上にドライブする。各ワードW₁~W_mは、それぞれnビットのCAMセルを備えている。

【 0 0 9 8 】

図12は、図11のCAMセルの構成回路図である。図12に例示したCAMセル1003は、不一致検出型のものである。CAMセル1003は、メモリセル1004及び一致比較回路1005から構成される。メモリセル1004は、1ビットのデータを記憶するSRAM構成のメモリセルである。図12においてDがデータ、DNが反転データを表す。一致比較回路1005は、メモリセル1004に記憶された1ビットのデータと検索ビット線対SL, SLN上にドライブされる検索データとを比較し、その一致比較結果を

10

20

30

40

50

一致線ML上に出力する。

【0099】

一致比較回路1005は、3つのnMOSトランジスタ(以下「nMOS」という。)1006, 1007, 1008を備えている。nMOS1006, 1007は、検索ビット線SLNと検索ビット線SLとの間に直列に接続されている。nMOS1006, 1007のゲートは、それぞれ、メモリセル1004のデータD, 反転データDNに接続されている。nMOS1008は、一致線MLとグランドとの間に接続されている。nMOS1008のゲートは、nMOS1006, 1007の間のノード1009に接続されている。

【0100】

まず、検索を行う前に、CAM1000のそれぞれのワード $W_1 \sim W_m$ に、検索対象であるデータ(登録ベクトル)を記入する。各ワード内の各CAMセル1003において、メモリセル1004へのデータの書き込み及びメモリセル1004からのデータの読み出しは、通常のSRAMと同様に行われる。

【0101】

検索時には、まず、比較レジスタ1001に検索データが格納される。検索データの各のビットは、検索ビット線ドライバ1002により、各々対応する検索ビット線上にドライブされる。

【0102】

各々のワード $W_1 \sim W_m$ では、各CAMセルに予め記憶されているデータと検索ビット線上にドライブされた検索データとの一致検索が同時(並列)に実行され、その結果が一致線 $ML_1 \sim ML_m$ 上に出力される。これらの検索結果は、それぞれ一致センス回路 $MSC_1 \sim MSC_m$ に入力される。各一致センス回路 $MSC_1 \sim MSC_m$ は、各検索結果を増幅し、一致センス出力として一致センス出力線 $MT_1 \sim MT_m$ に出力する。各一致センス出力は、一致フラグレジスタ $MFR_1 \sim MFR_m$ に格納され、一致フラグ出力として一致フラグ出力線 $MF_1 \sim MF_m$ に出力される。一致フラグは、「1」が「一致あり」、「0」が「一致なし」を表すものとする。

【0103】

各一致フラグ出力は、プライオリティ・エンコーダPEに入力される。プライオリティ・エンコーダPEでは、所定の優先順位付けに従って、一致が検出されたワードの中から最優先順位のワードのアドレス(最優先一致アドレス:HMA)を選択し出力する。各ワードの優先順位は、ワード W_1 が最も高く、 W_m に向かうに従って順次優先順位が低くなるものとする。

【0104】

尚、各ワード $W_1 \sim W_m$ 内の各CAMセル1003における一致検索は、次のようにして実行される。

【0105】

まず、初期化動作を実行する。初期化動作では、検索ビット線対SL, SLNがともに「L」(=「0」)とされる。一方、メモリセル1004に記憶されているデータに応じて、一致比較回路1005のnMOS1006, 1007のうち一方がオン状態、他方がオフ状態となる。従って、nMOS1006, 1007のうちオン状態の方を介して、両者の間のノード1009のレベルが「L」となり、nMOS1008はオフ状態となる。この状態で、一致線MLが「H」(=「1」)状態にプリチャージされる。尚、一致線MLは「H」が「一致」を表す。

【0106】

次に、検索ビット線を介して比較レジスタ1001に記憶された検索データの各ビットが各CAMセル1003に入力される。これにより、検索データSに応じて、検索ビット線対SL, SLNの何れか一方が「H」、他方が「L」となる。

【0107】

メモリセル1004に記憶されているデータDと検索データSとが一致する場合、ノー

10

20

30

40

50

ド1009のレベルは‘L’であり、nMOS1008はオフ状態に保持される。

【0108】

一方、データDと検索データSとが一致しない場合、ノード1009のレベルは‘H’となり、nMOS1008はオン状態になる。これにより、一致線MLはディスチャージされて‘L’状態となる。

【0109】

nビットのCAMセル1003からなるCAMワードの一致線MLは、各CAMセル1003のnMOS1008が平行に接続されたワイヤードOR回路を構成している。従って、1ワードを構成するmビットのCAMセル1003のすべてにおいて一致が検出された場合に限り、一致線MLは‘H’（「一致」）の状態に保持される。一方、1ビットでもCAMセル1003で不一致が検出されると、一致線MLは‘L’（「不一致」）の状態となる。

10

【0110】

例えば、検索の結果、一致フラグレジスタMFR₁～MFR_mに、一致フラグとして‘0’、‘1’、‘1’、‘0’、…、‘1’、‘0’が格納されたとする。この場合、ワードW₂、W₃、…、W_{m-1}で一致が検出されている。従って、プライオリティ・エンコーダPEは、最も優先順位が高いワードW₂のアドレスをHMAとして出力する。また、一致フラグレジスタMFR₂に格納された一致フラグを‘0’にクリアすることで、その次に優先順位が高いワードW₃のアドレスをHMAとして出力することができる。以下同様にして、一致が検出されたワードのアドレスを順次出力することができる。

20

【0111】

尚、TCAMとして使用する場合、ドント・ケアのビットについては、検索ビット線対SL、SLNとともに‘L’（＝‘0’）としておけばよい。

以上で、第2実施形態による連想記憶装置200に備えられた連想メモリ110-Cとして用いられる従来型のCAM1000を説明した。

【0112】

上述した第2実施形態によれば、複数のインデックス生成部110-1、110-2、110-3、…、110-Nと共に連想メモリ110-Cを備えたので、110-1、110-2、110-3、…、110-Nにおいてハッシュ衝突を回避することができない場合であっても、連想メモリ110-Cは、任意の登録ベクトルの登録を実現することができるので、ハッシュ衝突に関する問題を完全に解消することができる。また、登録ベクトルのほとんどは、複数のインデックス生成部110-1、110-2、110-3、…、110-Nで実現できるので、連想メモリ110-Cの記憶容量は小さくてもよい。このため、連想メモリ110-Cの消費電力を抑制することができる。従って、第2実施形態よれば、第1実施形態と同様に、ビットコストと消費電力の上昇を抑制しつつ、連想記憶装置200の記憶容量の大規模化を実現することができると共に、ハッシュ関数に起因したハッシュ衝突を完全に回避することが可能になる。

30

【0113】

次に、参考として、第2実施形態の連想記憶装置200による更新動作の実験結果を述べる。4個のインデックス生成部110-iと1個のCAMとを組み合わせた場合、5個のインデックス生成部110-iと1個のCAMとを組み合わせた場合のそれぞれについて、C言語で開発したシミュレータを用いて更新動作の実験を実施した。この実験では、ランダムなベクトルを100万回更新するものとし、インデックス生成部110-iで発生したハッシュ衝突を回避するために必要なCAM（連想メモリ110-C）の記憶容量の大きさを求めた。この実験では、n=32、k=100000とし、入力ベクトルを発生させる乱数の生成法としてメルセンヌツイスタ法を用いた。

40

【0114】

実験での更新手順は次の通りである。

（手順1）k個の登録ベクトルを生成し、本発明による連想記憶装置100の複数のインデックス生成部110-iとCAMに格納する。

50

(手順2) 複数の登録ベクトルの中からランダムに1つの登録ベクトルを取り出して、その登録ベクトルを連想記憶装置100から削除する。

(手順3) 新規な登録ベクトルを生成し、連想記憶装置100に格納する。

【0115】

上記実験において、CAMに登録されたベクトルの個数の平均値と最大値を求める。上記実験によるシミュレーションの結果から得られた値と、確率計算により得られた値は、ほぼ一致した。連想記憶装置の実現に用いるFPGA(Field Programmable Gate Array)の制約を考慮すると、3個のインデックス生成部110-iと1個のCAMとの組み合わせの場合、CAMの語数が多くなり過ぎる。そのため、5個のインデックス生成部110-iと1個のCAMとの組み合わせが望ましい。

10

上記の実験により、上述の実施形態による連想記憶装置の有効性を確認することができた。

【0116】

以上、本発明の実施形態を説明したが、本発明は上述の実施形態に限定されるものではなく、本発明の要旨を逸脱しない範囲で種々の変形、変更、修正、置換等が可能である。

【符号の説明】

【0117】

100, 200 ... 連想記憶装置

110-1, 110-2, 110-3, 110-i, 110-N ... インデックス生成部

110-C ... 連想メモリ

20

111 ... ハッシュ回路

112 ... 仮インデックス生成回路

113 ... 再生ベクトル生成回路

114 ... 比較回路

115 ... 出力回路

116 ... 衝突信号生成回路

120 ... 出力部

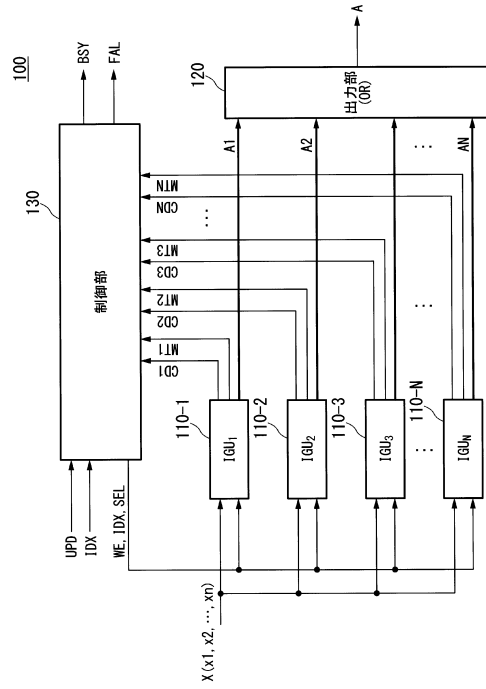
130 ... 制御部

1000 ... CAM

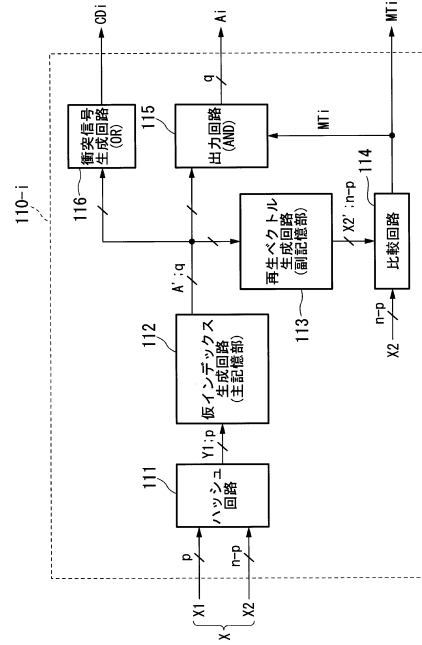
1003 ... CAMセル

30

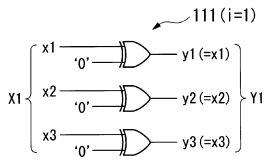
【図 1】



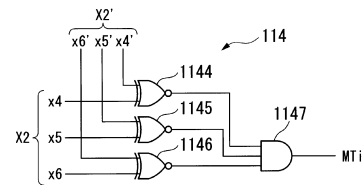
【図 2】



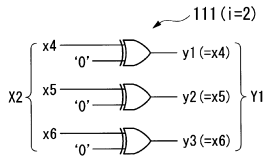
【図 3 A】



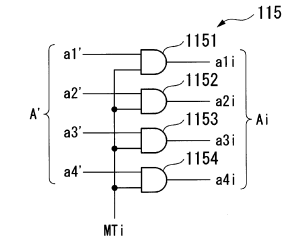
【図 4 A】



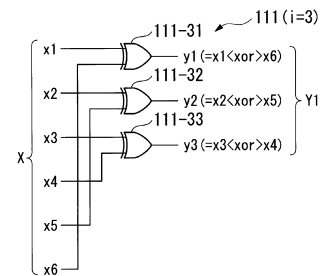
【図 3 B】



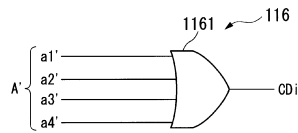
【図 4 B】



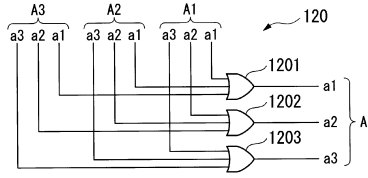
【図 3 C】



【図 4 C】



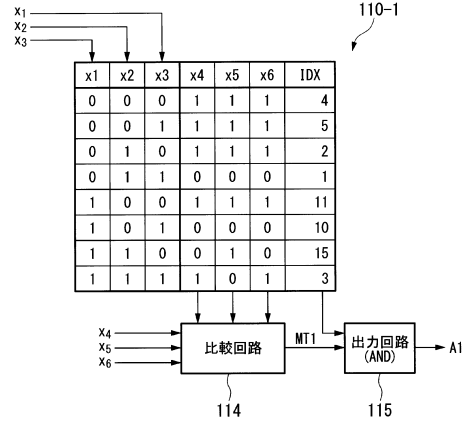
【図5】



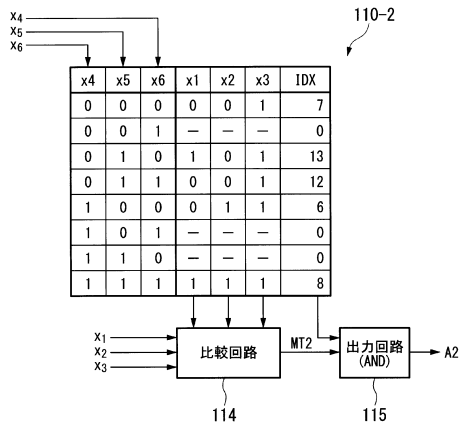
【図6】

| x1 | x2 | x3 | x4 | x5 | x6 | IDX |
|----|----|----|----|----|----|-----|
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 2 |
| 1 | 1 | 1 | 1 | 0 | 1 | 3 |
| 0 | 0 | 0 | 1 | 1 | 1 | 4 |
| 0 | 0 | 1 | 1 | 1 | 1 | 5 |
| 0 | 1 | 1 | 1 | 0 | 0 | 6 |
| 0 | 0 | 1 | 0 | 0 | 0 | 7 |
| 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| 1 | 1 | 1 | 0 | 1 | 1 | 9 |
| 1 | 0 | 1 | 0 | 0 | 0 | 10 |
| 1 | 0 | 0 | 1 | 1 | 1 | 11 |
| 0 | 0 | 1 | 0 | 1 | 1 | 12 |
| 1 | 0 | 1 | 0 | 1 | 0 | 13 |
| 0 | 0 | 1 | 1 | 0 | 0 | 14 |
| 1 | 1 | 0 | 0 | 1 | 0 | 15 |

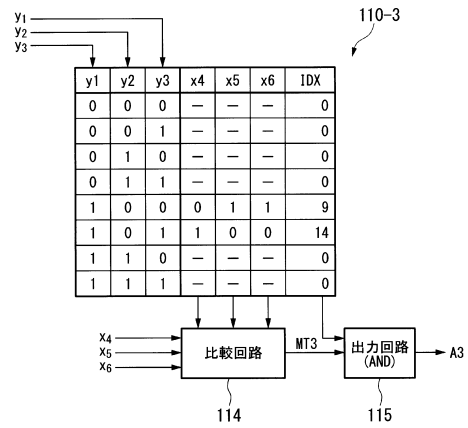
【図7】



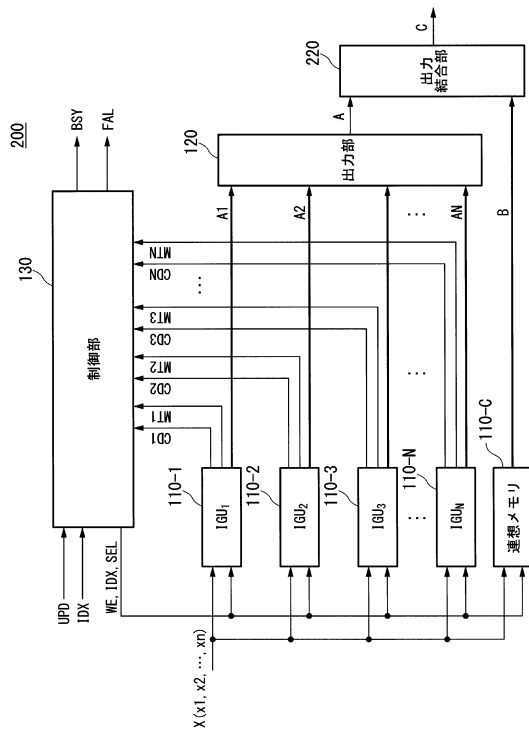
【図8】



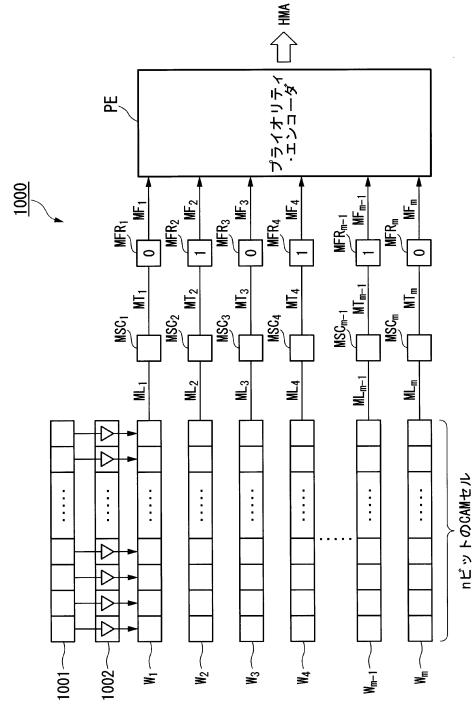
【図9】



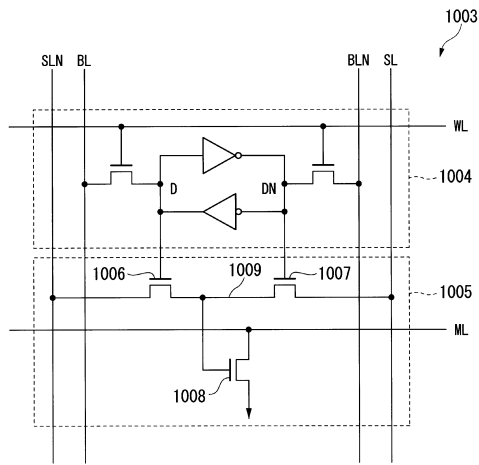
【図10】



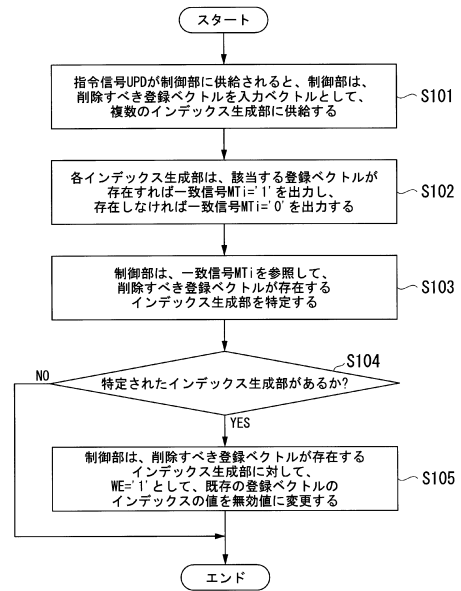
【図11】



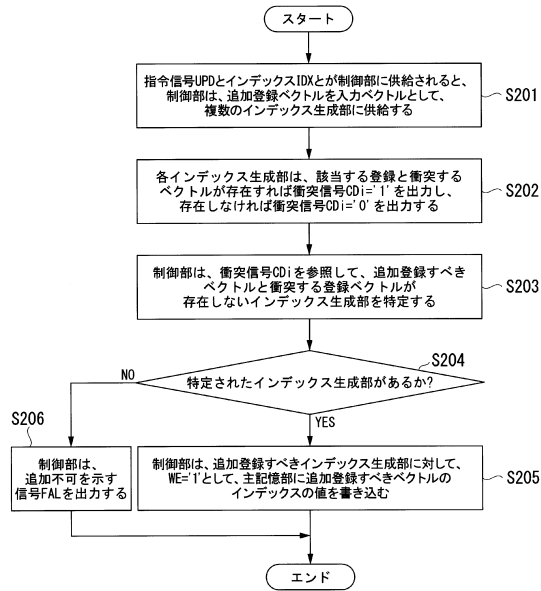
【図12】



【図13】



【図14】



フロントページの続き

審査官 後藤 彰

- (56)参考文献 国際公開第2007/119539(WO, A1)
国際公開第2007/119540(WO, A1)
中原 啓貴, " 並列ふるい法とMPUを用いたウイルス検出エンジンについて ", 電子情報通信学会技術研究報告 RECONF2009-41-RECONF2009-53, 日本, 社団法人電子情報通信学会, 2009年12月3日, 第109巻第320号, 第25頁-第30頁
中原 啓貴, " 並列インデックス生成器を用いたIPv6検索アーキテクチャに関して ", 情報処理学会研究報告 2013 February システムLSI設計技術(SLDM), 日本, 一般社団法人情報処理学会, 2013年2月15日, No.159 (5)

(58)調査した分野(Int.Cl., DB名)

G11C 15/04
G06F 17/30