(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4934825号
(P4934825)
(45)発行日　平成24年5月23日(2012.5.23)　(24)登録日　平成24年3月2日(2012.3.2)

(51)Int.Cl.　F I
G11C 15/04 (2006.01)　G11C 15/04 D

請求項の数 3　(全 19 頁)

(21)出願番号　特願2008-510867(P2008-510867)
(86)(22)出願日　平成19年3月27日(2007.3.27)
(86)国際出願番号　PCT/JP2007/056498
(87)国際公開番号　WO2007/119540
(87)国際公開日　平成19年10月25日(2007.10.25)
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2006-101106(P2006-101106)
(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　堀江　義隆

最終頁に続く

(54)【発明の名称】連想メモリ

(57)【特許請求の範囲】
【請求項１】
入力データに対しそのデータに対応する固有のインデックスを出力する連想記憶メモリであって、
入力データに対しそのデータに対応する固有のインデックスを出力する関数（以下「ＣＡＭ（Content Addressable Memory）関数」という。）fの無効出力値をドント・ケアで置き換えた関数（以下「簡略化ＣＡＭ関数」という。）gを表すＬＵＴ結合論理回路又はＰＬＡにより構成された簡略化関数演算部と、
前記ＣＡＭ関数fの逆関数f^{-1}が記憶された補助メモリと、
前記簡略化関数演算部の出力値が、前記入力データに対するＣＡＭ関数fの出力に一致するか否かを判定し、一致する場合には前記簡略化関数演算部の出力値を出力し、それ以外の場合は無効信号を出力する一致判定手段と、
を備え、
前記簡略化関数演算部は、前記入力データに対して前記簡略化ＣＡＭ関数gの演算値（以下「仮インデックス値」という。）を前記補助メモリの読み出しアドレスとして出力し、
前記補助メモリは、前記仮インデックス値が読み出しアドレスとして入力されると、その仮インデックス値に対する逆関数f^{-1}の値を出力し、
前記一致判定手段は、前記入力データと前記補助メモリが出力する逆関数f^{-1}の値とを比較して、両者が一致する場合は前記簡略化関数演算部の出力値を出力し、それ以外の場

合は無効信号を出力すること
を特徴とする連想メモリ。
【請求項２】
前記ＬＵＴ結合論理回路は、ＬＵＴカスケード論理回路であることを特徴とする請求項１記載の連想メモリ。
【請求項３】
前記ＬＵＴ結合論理回路は、ｐｑ回路網であることを特徴とする請求項１記載の連想メモリ。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、連想メモリ（内容検査メモリ：Content Addressable Memory：以下「ＣＡＭ」という。）に関し、特に、高速検索が可能で、消費電力を抑え、且つ小さい実装面積で実装可能な連想メモリに関する。
【背景技術】
【０００２】
通常のメモリは、与えられたインデックス（アドレス）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のインデックス（アドレス）を生成する（非特許文献１，２参照）。
【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。
【０００５】
〔定義１〕（ＢＣＡＭ）
ｎ入力で登録データ数ｐのＢＣＡＭテーブルは、ｐ個の異なる２値ベクトルを格納する。また、ｐ個のベクトルは、アドレス１からアドレスｐに順に格納されていると仮定する。また、各ベクトルのアドレスはｍビットで表現可能である。ｍは次式（１）により表される。
【０００６】
【数１】

$$m = \lceil \log_2(p+1) \rceil \qquad (1)$$

また、ｐ個のベクトルは、アドレス１からアドレスｐに順に格納されていると仮定する。対応するＢＣＡＭ関数$f:\{0,1\}^n \rightarrow \{0,1\}^m$は、以下の条件を満たす：
f(x)は入力xと同じベクトルがＢＣＡＭテーブル中にあるとき、ベクトルxを格納するＣＡＭのアドレス（１からｐの値の何れか）を出力する。入力xと同じパターンがＢＣＡＭテーブル中に無い場合には、f(x)の値は0である。
（定義終り）
【０００７】
（例１）
（表１）は、７個の２値ベクトルを格納するＢＣＡＭを示す。対応するＢＣＡＭ関数を（表２）に示す。何れも、入力データに完全に一致したベクトルを格納するアドレスを３

ビットの数（例えば、‘０１１’）として出力する。入力ベクトルと一致するものがＢＣＡＭ中に格納されていない場合には、０を出力する。

（例終わり）

【０００８】

【表１】

表 1: BCAM 関数の例

アドレス	ベクトル
1	0010
2	0111
3	1101
4	0101
5	0011
6	1011
7	0001

【０００９】

【表２】

表 2: BCAM テーブルの例

x_1	x_2	x_3	x_4	f_2	f_1	f_0
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	0	0	1
0	0	1	1	1	0	1
0	1	0	0	0	0	0
0	1	0	1	1	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	0
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	1	1	0
1	1	0	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	0	0	0
1	1	1	1	0	0	0

【００１０】

〔定義２〕（ＴＣＡＭ）

ｎ入力で要素数ｐのＴＣＡＭテーブルは、ｐ個の３値ベクトルを格納する。また、ｐ個のベクトルは、アドレス１からアドレスｐに順に格納されていると仮定する。また、各ベクトルのアドレスはｍビットで表現可能である。ここで、ｍは上記式（１）により表される。各３値ベクトルは、０，１，又は＊（ドント・ケア）から構成される。対応するＴＣＡＭ関数：$f:\{0,1\}^n \rightarrow \{0,1\}^m$は、以下の条件を満たす：

入力xに対して、xと一致するベクトルがＴＣＡＭテーブル中にある場合、出力f(x)は、

一致したベクトル中の最小のアドレスを示す。また、xと一致するベクトルがＴＣＡＭテーブル中に無い場合には、０を出力する。

（定義終り）

【００１１】

（例２）

（表３）に示すＴＣＡＭは、７個の３値ベクトルを格納する。対応するＴＣＡＭ関数を（表４）に示す。入力x=(1,0,1,1)は、アドレス５及び６に蓄えられているパターンと一致する。５の方が小さいので、出力は(0,1,0,1)となる。

（例終わり）

【００１２】

【表３】

表 3: TCAM 関数の例

アドレス	ベクトル
1	*010
2	0011
3	1101
4	1100
5	*011
6	1*11
7	*001

【００１３】

【表４】

表 4: TCAM テーブルの例

x_1	x_2	x_3	x_4	f_2	f_1	f_0
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	1	0
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	0	1	1	1	0	1
1	1	0	0	1	0	0
1	1	0	1	0	1	1
1	1	1	0	0	0	0
1	1	1	1	1	1	0

【００１４】

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（半導体メモリ）を用いてＣＡＭを実現することが多い。以下、ハードウェアで構成された従来のＣＡＭについて説明する。

【００１５】

図８は、従来のＣＡＭ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）。ＣＡＭ１００は、比較レジスタ１０１、検索ビット線ドライバ１０２、ｎ個のワードW_1～W_n、ｎ個の一致センス回路MSC_1～MSC_n、ｎ個の一致フラグレジスタMFR_1～MFR_n、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。

【００１６】

比較レジスタ１０１は、ｍ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１０２は、比較レジスタ１０１の各ビットを検索ビット線上にドライブする。各ワードW_1～W_nは、それぞれｍビットのＣＡＭセルを備えている。

【００１７】

図９は、図８のＣＡＭセルの構成回路図である。図９に例示したＣＡＭセル１０３は、不一致検出型のものである。ＣＡＭセル１０３は、メモリ・セル１０４及び一致比較回路１０５から構成される。メモリ・セル１０４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９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１０５は、メモリ・セル１０４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一致線ＭＬ上に出力する。

【００１８】

一致比較回路１０５は、３つのｎＭＯＳトランジスタ（以下「ｎＭＯＳ」という。）１０６，１０７，１０８を備えている。ｎＭＯＳ１０６，１０７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。ｎＭＯＳ１０６，１０７のゲートは、それぞれ、メモリ・セル１０４のデータＤ，反転データＤＮに接続されている。ｎＭＯＳ１０８は、一致線ＭＬとグランドとの間に接続されている。ｎＭＯＳ１０８のゲートは、ｎＭＯＳ１０６，１０７の間のノード１０９に接続されている。

【００１９】

まず、検索を行う前に、ＣＡＭ１００のそれぞれのワードW_1～W_nに、検索対象であるデータが記憶される。各ワード内の各ＣＡＭセル１０３において、メモリ・セル１０４へのデータの書き込み及びメモリ・セル１０４からのデータの読み出しは、通常のＳＲＡＭと同様にして行われる。

【００２０】

検索時には、まず、比較レジスタ１０１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１０２により、各々対応する検索ビット線上にドライブされる。

【００２１】

各々のワードW_1～W_n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～ML_n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～MSC_nに入力される。各一致センス回路MSC_1～MSC_n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～MT_n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～MFR_nに格納され、一致フラグ出力として一致フラグ出力出力線MF_1～MF_n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。

【００２２】

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_nに向かうに従って順次優先順位が低くなるものとする。

【００２３】

尚、各ワードW_1～W_n内の各ＣＡＭセル１０３における一致検索は、次のようにして実行される。

【００２４】

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１０４に記憶されているデータに応じて、一致比較回路１０５のｎＭＯＳ１０６，１０７のうち一方がオン状態、他方がオフ状態となる。従って、ｎＭＯＳ１０６，１０７のうちオン状態の方を介して、両者の間のノード１０９のレベルが‘Ｌ’となり、ｎＭＯＳ１０８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。

【００２５】

次に、検索ビット線を介して比較レジスタ１０１に記憶された検索データの各ビットが各ＣＡＭセル１０３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。

【００２６】

メモリ・セル１０４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、ノード１０９のレベルは‘Ｌ’であり、ｎＭＯＳ１０８はオフ状態に保持される。

【００２７】

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１０９のレベルは‘Ｈ’となり、ｎＭＯＳ１０８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。

【００２８】

ｍビットのＣＡＭセル１０３からなるＣＡＭワードの一致線ＭＬは、各ＣＡＭセル１０３のｎＭＯＳ１０８がパラレル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１０３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１０３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。

【００２９】

例えば、検索の結果、一致フラグレジスタMFR_1～MFR_n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W_2，W_3，…，W_{n-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W_2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MFR_2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W_3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。

【００３０】

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をともに‘Ｌ’（＝‘０’）としておけばよい。

【００３１】

図１０は、図８のＣＡＭセルの別の例の構成回路図である。図１０に示すＣＡＭセル１０３’は一致検出型のものであり、図９と同様、ＳＲＡＭ構成のメモリ・セル１０４及び一致比較回路１０５を備えている。ＣＡＭセル１０３’は、図９のＣＡＭセル１０３において、一致比較回路１０５のｎＭＯＳ１０８の接続が異なる。図ＤのｎＭＯＳ１０８は、一致線ML_aと一致線ML_bとの間に接続されている。ｎＭＯＳ１０８のゲートは、ｎＭＯＳ１０６，１０７の間のノード１０９に接続されている。

【００３２】

ＣＡＭセル１０３’では、検索時には、初期化動作として、ビット線対ＳＬ，ＳＬＮが共に‘Ｈ’とされる。一方、メモリ・セル１０４に記憶されているデータに応じて、一致比較回路１０５のｎＭＯＳ１０６，１０７のうち一方がオン状態、他方がオフ状態となる。従って、ｎＭＯＳ１０６，１０７のうちオン状態の方を介して、両者の間のノード１０９のレベルが‘Ｈ’となり、ｎＭＯＳ１０８はオン状態となる。この状態で、一致線ＭＬの一端が‘Ｈ’（＝‘１’）状態にプリチャージされる。尚、一致線ＭＬは‘Ｈ’が「不一致」を表す。

【００３３】

ｍビットのＣＡＭセル１０３’からなるＣＡＭワードの一致線ＭＬは、各ＣＡＭセル１０３’のｎＭＯＳ１０８がシリアルに接続されたＡＮＤ回路を構成する。従って、各々のＣＡＭセルの一致線ML_a，ML_bは、各々のＣＡＭセル１０３’のｎＭＯＳ１０８を介して‘Ｈ’にプリチャージされる。

【００３４】

その後、検索ビット線を介して比較レジスタ１０１に記憶された検索データの各ビットが各ＣＡＭセル１０３’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。

【００３５】

メモリ・セル１０４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、ノード１０９のレベルは‘Ｈ’であり、ｎＭＯＳ１０８はオン状態に保持される。

【００３６】

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１０９のレベルは‘Ｌ’となり、ｎＭＯＳ１０８はオフ状態になる。

【００３７】

ＣＡＭワードのｍビットのＣＡＭセル１０３’のすべての状態が確定した後、一致線ＭＬの一方の端部からディスチャージを開始し、他方の端部で一致比較結果を判定する。このとき、１ビットでも不一致のＣＡＭセル１０３’がある場合には、一致比較結果は‘Ｈ’、すなわち、不一致の状態に保持される。一方、すべてのＣＡＭセル１０３’で一致が検出された場合のみ、一致比較結果は‘Ｌ’、すなわち一致状態となる。

【００３８】

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をともに‘Ｈ’（＝‘１’）としておけばよい。

【特許文献１】特開２００４－２９５９６７号公報
【特許文献２】特願２００３－３８９２６４号明細書
【特許文献３】特開２００４－２５８７９９号公報
【特許文献４】特開2004-258799号公報
【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，倍風館，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.
【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 and M.Matsuura, "BDD representation for incompletely specified multiple-output logic functions and its applications to functional decomposition," Design Autonmation Conference, June 2005, (pp.373-378).
【非特許文献６】井口、笹尾、”LUTカスケード・アーキテクチャについて、” 平成15年電気学会 電子・情報・システム部門大会、MC2-4、2003年8月29日～30日、秋田大学。

【発明の開示】
【発明が解決しようとする課題】
【００３９】
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、デバイスの構成は複雑となる。そのため、ＣＡＭの１ビットあたりの価格（ビットコスト）は、ＲＡＭに比べると１０～３０倍程度、高価なものになる。
【００４０】
また、１ビットあたりの消費電力がＲＡＭに比べて遙かに大きい（非特許文献３参照）。これは、上で説明したように、すべてのＣＡＭ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ＲＡＭの約５０倍程度にもなる。
【００４１】
そこで、本発明の目的は、検索の高速性を維持しつつも、消費電力を抑え且つデバイスの構造を簡単化して小さい実装面積で実装可能な連想メモリを提供することにある。
【課題を解決するための手段】
【００４２】
以下、本明細書において使用する用語の定義及び本発明の前提となる理論を説明し、その後本発明の構成及び作用について説明する。
【００４３】
〔１〕ＣＡＭ関数の性質
〔定義３〕（分解表、基本分解表、列複雑度）
関数$f(X) : B^n \to B^q$、及び$X=(x_1,x_2,\cdots,x_n)$が与えられているものとする。ここで、$B=\{0,1\}$である。(X_L, X_H)をXの分割とする。fの「分解表」とは、二次元のマトリックスであって、列のラベルは、X_LにBの構成要素をすべての可能な組み合わせに対して割り当てたものであり、また行のラベルは、X_HにBの構成要素をすべての可能な組み合わせに対して割り当てたものである。また、対応するマトリックスの値は$f(X_L,X_H)$の値に等しい。
関数fの分解表のうちで、$X_L=(x_1, x_2, \cdots, x_{nL})$且つ$X_H=(x_{nL+1}, x_{nL+2}, \cdots, x_n)$となる分解表を「基本分解表」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」という。
尚、分解表の特別な場合として、$X_L=X$の場合も考える。
（定義終り）
【００４４】
〔定義４〕（Ｃ尺度）
変数の順序を$(x_1,x_2,\cdots,x_n)$としたとき、論理関数fの基本分解表の列複雑度の最大値を、fの「Ｃ尺度」という。
（定義終り）
【００４５】
（例３）
$f_1=x_1x_2 \vee x_3x_4 \vee x_5x_6$のＣ尺度は３であるが、$f_2=x_1x_5 \vee x_2x_6 \vee x_3x_4$のＣ尺度は８である。
（例終わり）
【００４６】
分解表の列複雑度は、ＭＴＢＤＤ（多端子二分決定グラフ）の幅に等しい。従って、論理関数のＣ尺度は、与えられた入力変数の順におけるＭＴＢＤＤの幅の最大値に等しい。与えられた論理関数$f(x_1,x_2,\cdots,x_n)$に対して、Ｃ尺度は容易に計算でき、一意的に定まる。後述するように、Ｃ尺度が小さい関数は、ＬＵＴカスケード（LUT (Lookup-table) cascade）で効率的に実現可能である。従って、Ｃ尺度は、論理関数をＬＵＴカスケードで実現する際の複雑さの尺度となる。
【００４７】
〔補題１〕
与えられた関数fに対して、非零出力を生ずる入力の組み合わせの個数をpとする。この

とき、fのＣ尺度は高々p+1である。

（補題終り）

【００４８】

〔定理１〕（ＢＣＡＭ関数のＣ尺度）

要素数pのＢＣＡＭテーブルが与えられたとき、そのＢＣＡＭ関数のＣ尺度は高々p+1である。

（定理終り）

【００４９】

〔定理２〕（ＴＣＡＭ関数のＣ尺度）

ＴＣＡＭテーブルがp個のベクトルを格納し、各ベクトルが高々k個のドント・ケアを有するとき、対応するＴＣＡＭ関数のＣ尺度は高々2^kp+1である。

（定理終り）

【００５０】

〔２〕ＬＵＴカスケード

ＣＡＭ関数は通常のＲＡＭでも実現可能である。例えば、上記（表１）に示した要素７のＢＣＡＭ関数は、（表２）に示したように、１６ワードのＲＡＭにより実現することができる。ここで、各ワードは３ビットである。ｎ入力のＣＡＭ関数を単一のＲＡＭで実現する場合、ＲＡＭの大きさは、ＢＣＡＭがわずか数個のベクトルしか含まない場合であっても、2^nに比例して大きくなる。そこで、ＬＵＴカスケードを用いることにより、必要なメモリ量を大幅に削減することが可能となる（特許文献３参照）。

【００５１】

〔定理３〕

与えられた関数fに対して、X_Lを分解表の列に対応する変数、X_Hを行に対応する変数とし、μを分解表の列複雑度とする。このとき、関数fは図１に示すような回路で実現することが可能である。この場合、二つのブロックＨとＧの間を結ぶ信号線数（以下「レイル数」という。）は

【００５２】

【数２】

$$\lceil \log_2 \mu \rceil \qquad (2)$$

である。

（定理終り）

【００５３】

二つのブロック間を結ぶ信号線数がX_L中の変数の個数よりも少ないとき、関数を実現するためのメモリ量を削減できる可能性がある。この手法を「関数分解」という。与えられた関数を繰り返し関数分解することにより、図２に示すようなＬＵＴカスケードが得られる（非特許文献４参照）。ＬＵＴカスケードは「セル」から構成され、隣接するセル間を接続する信号線を「レイル」という。Ｃ尺度が小さな関数は小型のＬＵＴカスケードで実現可能である。Ｃ尺度を求めるには、必ずしも分解表を使用する必要はなく、多出力関数の特性関数を表現する二分決定グラフ（Binary Decision Diagram for Characteristic Function：以下「BDD_for_CF」という。）から効率的に計算することができる（特許文献２，非特許文献５参照）。

【００５４】

〔定理４〕

Ｃ尺度がμの論理関数は、入力数が高々

【００５５】

【数３】

$$\lceil \log_2 \mu \rceil + 1 \qquad (3)$$

出力数が

【００５６】
【数４】

$$\lceil \log_2 \mu \rceil \tag{4}$$

のセルから構成されたＬＵＴカスケードで実現可能である。

（定理終り）

【００５７】
〔定理５〕
関数fを実現するＬＵＴカスケードを考える。いま、nを外部入力変数、sをセル数、rを最大レイル数（即ち、セル間の信号線数）、kをセルの最大入力数、μを関数fのＣ尺度とする。

【００５８】
【数５】

$$k > r,\ r = \lceil \log_2 \mu \rceil \tag{5}$$

が成立するとき、以下の関係を満たすfを実現するＬＵＴカスケードが存在する。

【００５９】
【数６】

$$s \leq \left\lceil \frac{n-r}{k-r} \right\rceil \tag{6}$$

（定理終り）

【００６０】
〔３〕ドント・ケアを用いた設計法
〔３－１〕ＢＣＡＭ関数の設計法
ＢＣＡＭ関数では、真理値表において、非零出力の値が、全体の組み合わせの数2^nに比べて大幅に少ない。つまり、次の（仮定１）が成立するものとする。

【００６１】
（仮定１）　ＢＣＡＭテーブルの入力ビット数をn、ベクトル数をpとするとき、$p \ll 2^n$。

【００６２】
例えば、n=32でp=1000のＢＣＡＭを考えるとき、非零出力の割合は全最小項の個数の1000／2^{32}=2.3×10^{-7}となる。

【００６３】
ＢＣＡＭ関数をＢＤＤで表現すると、ＢＤＤの最大幅が、Ｃ尺度を超えないこと、及び〔定理１〕から、レイル幅はp+1を超えないことがわかる。しかし、全体の幅はp+1近くなる。従って、ＢＣＡＭ関数をＬＵＴカスケードで実現する場合、入力数が

【００６４】
【数７】

$$\lceil \log_2(p+1) \rceil + 1 \tag{7}$$

のセルが多数必要となる。そこで、ここではドント・ケアの概念を用いて、ＢＣＡＭ関数を実現するハードウェア量を削減する方法について、図３を参照しながら説明する。

【００６５】
〔アルゴリズム１〕
（１）fをＢＣＡＭ関数とする。また、fにおいてＣＡＭに登録されていないデータに対する出力値をドント・ケアとした関数をgとする。

【００６６】
（２）gの特性関数を表現する二分決定グラフ（BDD_for_CF）を生成し、簡単化を行う。

【００６７】
（３）簡単化したＢＤＤから、ＬＵＴカスケード１を生成する。一般に、ＬＵＴカスケード１は、fを実現するＬＵＴカスケード（これを「厳密なＬＵＴカスケード」と呼ぶ。）よりも簡単になる。
【００６８】
（４）検索データが登録データと一致するときは、ＬＵＴカスケード１は正しい値を出力する。検索データが登録データと一致しないときは、ＬＵＴカスケード１は、誤った値を出力する可能性がある。
【００６９】
（５）誤りを補正するために、m入力n出力の補助メモリ２を用いる。ここで、mは次式で表される。
【００７０】
【数８】

$$m = \lceil \log_2(p+1) \rceil \qquad (8)$$

この補助メモリ２には、各アドレスにＢＣＡＭテーブルの対応するデータが格納されている。
【００７１】
（６）ＬＵＴカスケード１の出力のインデックスを補助メモリ２に供給し、補助メモリ２内の登録データを読み出す。そして、一致回路３により、入力データと補助メモリ２の出力データとを比較する。両者が一致すれば、ＬＵＴカスケード１の出力値は正しいことが保証される。従ってエンコーダ４はＬＵＴカスケード１の出力のインデックスをそのまま出力する。一方、補助メモリ２の出力データと入力データとが一致しないときは、ＣＡＭ内にそのデータは登録されていない。従って、そのときには、エンコーダ４は無効のインデックス（０）を出力する。

（アルゴリズム終り）

【００７２】
補助メモリ２の全ビット数は、$n2^m$であり、ハードウェアのコストは、ＬＵＴカスケード１のコストに比べ無視できる程度である。
【００７３】
〔３－２〕ＴＣＡＭ関数の設計法
ＴＣＡＭ関数の場合、ＴＣＡＭテーブルは３値となるので、補助メモリ２は、m入力2n出力となる。また、一致回路３で、ドント・ケアに対応するビットは無視する。他の部分はＢＣＡＭと同じである。
【００７４】
〔１〕本発明の構成
本発明に係る連想メモリの構成は、入力データに対しそのデータに対応する固有のインデックスを出力する連想記憶メモリであって、
入力データに対しそのデータに対応する固有のインデックスを出力する関数（以下「ＣＡＭ（Content Addressable Memory）関数」という。）fの無効出力値をドント・ケアで置き換えた関数（以下「簡略化ＣＡＭ関数（reduced Content Addressable Memory function）」という。）gを表すＬＵＴ結合論理回路又はＰＬＡ（Programmable Logic Array）により構成された簡略化関数演算部と、
前記ＣＡＭ関数fの逆関数f^{-1}が記憶された補助メモリと、
前記簡略化関数演算部の出力値が、前記入力データに対するＣＡＭ関数fの出力に一致するか否かを判定し、一致する場合には前記簡略化関数演算部の出力値を出力し、それ以外の場合は無効信号を出力する一致判定手段と、
を備え、
前記簡略化関数演算部は、前記入力データに対して前記簡略化ＣＡＭ関数gの演算値（以下「仮インデックス値」という。）を前記補助メモリの読み出しアドレスとして出力し

、
前記補助メモリは、前記仮インデックス値が読み出しアドレスとして入力されると、その仮インデックス値に対しする逆関数f^{-1}の値を出力し、
前記一致判定手段は、前記入力データと前記補助メモリが出力する逆関数f^{-1}の値とを比較して、両者が一致する場合は前記簡略化関数演算部の出力値を出力し、それ以外の場合は無効信号を出力することを特徴とする。
【００７５】
この構成によれば、ＬＵＴ結合論理回路は、通常のＲＡＭを複数個用いて構成することができる。また、補助メモリも通常のＲＡＭで構成できる。また、ＣＡＭ関数fそのものを表すＬＵＴ結合論理回路ではなく、ＣＡＭ関数fの無効出力値をドント・ケアで置き換えた簡略化関数gを表すＬＵＴ結合論理回路又はＰＬＡを用いることによって、当該ＬＵＴ結合論理回路又はＰＬＡを構成するために必要なメモリ量を大幅に削減することができる。従って、全体として、従来のＣＡＭに比べ、デバイスの構造を簡単化して小さい実装面積により実装が可能となる。
【００７６】
また、通常のＲＡＭで構成することによって、専用のＣＡＭ回路を必要としない。故に、ＡＳＩＣで構成する以外に、汎用のＲＡＭを内蔵したＦＰＧＡ（Field Programmable Gate Array）やＣＬＤ（Complex Programmable Logic Device）のようなプログラマブル・デバイスを用いて簡単に低コストで構成することもできる。
【００７７】
また、一致判定手段以外はすべて通常のＲＡＭで構成できる。そして、１回の検索動作に対して、数回（ＬＵＴ結合論理回路でのＲＡＭのアクセス回数＋１）のＲＡＭのアクセスで仮インデックス値が得られる。各ＲＡＭのアクセスでは、ＲＡＭ内の１つのアドレスのみがアクセスされる。全体として、数回のＲＡＭのメモリ・アクセスのみで仮インデックス値が得られる。従って、従来のＣＡＭに比べると、消費電力を大幅に低減することが可能となる。

【００７８】
一方、高速性の面では、従来のＣＡＭに比べると遅くなるものの、通常のＲＡＭをＣＰＵを用いて検索する方法に比べると遙かに高速に検索を行うことができる。
【００７９】
ここで、「ＬＵＴ結合論理回路」とは、複数のＬＵＴ（Look-Up Table）をカスケード状又はネットワーク状に結合した回路をいうが、必ずしもハードウェア的に複数のＬＵＴを配置・結合する必要はない。例えば、１つのメモリ内に複数のＬＵＴを記憶させ、ＬＵＴの選択を順次切り替えながら、メモリの出力値をメモリの読み出しアドレスにフィードバックすることによってＬＵＴ結合を実現するような回路であってもよい。ＬＵＴ結合論理回路としては、ＬＵＴカスケード論理回路の他に、ＬＵＴをネットワーク状に結合したＬＵＴネットワークなどを使用することも可能である。
【００８０】
尚、本発明はＢＣＡＭ及びＴＣＡＭの双方に適用することが可能である。
【発明の効果】
【００８１】
以上のように、本発明によれば、検索の高速性を維持しつつ、消費電力を抑え且つデバイスの構造を簡単化して小さい実装面積で実装可能な連想メモリを提供することができる。
【図面の簡単な説明】
【００８２】
【図１】論理関数の関数分解を表す図である。
【図２】中間出力を有するＬＵＴカスケードを表す図である。
【図３】ドント・ケアを用いたＣＡＭ関数の実現法を説明する図である。

【図４】本発明の実施例１に係る連想メモリの全体構成を表す図である。
【図５】図４の簡略化関数演算部５の構成を表す図である。
【図６】図４の一致判定手段７の構成を表す図である。
【図７】実施例２に係る連想メモリの一致判定手段７の構成を表す図である。
【図８】従来のＣＡＭの基本構成の一例を表すブロック図である。
【図９】図８のＣＡＭセルの構成回路図である。
【図１０】図８のＣＡＭセルの別の例の構成回路図である。
【符号の説明】
【００８３】
１　ＬＵＴカスケード
２　補助メモリ
３　一致回路
４　エンコーダ
５　簡略化関数演算部
６　補助メモリ
７　一致判定手段
１０　連想メモリ
１１　入力変数レジスタ
１２－１～１２－ｓ　論理関数メモリ
１３　出力変数レジスタ
２１　ＥＸＮＯＲゲート
２２　ＡＮＤゲート
２３　ＡＮＤゲート
３１　ｐｑ素子
【発明を実施するための最良の形態】
【００８４】
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】
【００８５】
図４は、本発明の実施例１に係る連想メモリの全体構成を表す図である。本実施例の連想メモリ１０は、簡略化関数演算部５、補助メモリ６、及び一致判定手段７を備えている。
【００８６】
連想メモリ１０は、外部回路から入力されるｎビットの入力データ$X=(x_1,\cdots,x_n)$に対しその入力データXに対応するｍビットの固有のインデックス$A=(a_1,\cdots,a_m)$を出力する。入力データXは、簡略化関数演算部５及び一致判定手段７に入力される。
【００８７】
以下では、入力データXに対しその入力データXに対応する固有のインデックスAを出力するＣＡＭ関数を、Fと記す。また、ＣＡＭ関数Fの出力のうち、無効なインデックス値をドント・ケアで置き換えた簡略化ＣＡＭ関数を、Gと記す。
【００８８】
簡略化関数演算部５は、簡略化ＣＡＭ関数Gを演算するＬＵＴ結合論理回路により構成されている。簡略化関数演算部５は、入力データXに対し仮インデックス$A'=G(X)$を演算する。この仮インデックスA'は、補助メモリ６及び一致判定手段７に出力される。
【００８９】
補助メモリ６は、ＣＡＭ関数Fの逆関数F^{-1}、すなわち、固有のインデックスAに対しそのインデックスAに対応するデータXを出力するデータ出力関数が、ＬＵＴとして記憶されている。補助メモリ６には、簡略化関数演算部５が出力する仮インデックスA'が入力され、それに対する逆算データ$X'=F^{-1}(A')$が出力される。仮インデックスA'は、一致判定手段７に出力される。

【００９０】
一致判定手段７は、入力データXと逆算データX'とを比較し、両者が一致するか否かを判定する。そして、両者が一致する場合には仮インデックスA'を出力インデックスAとして出力し、一致しない場合には無効値を出力する。
【００９１】
図５は、図４の簡略化関数演算部５の構成を表す図である。本実施例１においては、簡略化関数演算部５として、ＬＵＴカスケード論理回路を用いる。
【００９２】
簡略化関数演算部５は、入力変数レジスタ１１、論理関数メモリ１２－１～１２－ｓ、及び出力変数レジスタ１３を備えている。
【００９３】
入力変数レジスタ１１は、外部から入力される入力データXを一時的に保持するレジスタである。論理関数メモリ１２－１～１２－ｓは、簡略化ＣＡＭ関数Gを関数分解して得られるＣＡＭ関数Gの部分関数$G_1,\cdots,G_s$がＬＵＴとして格納されている。ここで、Xの分割を$X=(X_1,X_2,\cdots,X_s)$とした場合、各部分関数$G_1,\cdots,G_s$は次のようになる。
【００９４】
【数９】

$$\begin{aligned}(\boldsymbol{A}'_1,\boldsymbol{R}_1)&=\boldsymbol{G}_1(\boldsymbol{X}_1)\\(\boldsymbol{A}'_2,\boldsymbol{R}_2)\quad&\boldsymbol{G}_2(\boldsymbol{X}_2,\boldsymbol{R}_1)\\&\vdots\\(\boldsymbol{A}'_{s-1},\boldsymbol{R}_{s-1})&=\boldsymbol{G}_{s-1}(\boldsymbol{X}_{s-1},\boldsymbol{R}_{s-2})\\\boldsymbol{A}'_s&=\boldsymbol{G}_s(\boldsymbol{X}_s,\boldsymbol{R}_{s-1})\\\boldsymbol{A}'\quad&(\boldsymbol{A}'_1,\boldsymbol{A}'_2,\ldots,\boldsymbol{A}'_s)\end{aligned}\tag{9}$$

【００９５】
ここでは、ベクトル$A_1',A_2',\cdots,A_{s-1}'$は、ＬＵＴカスケードの中間出力であり、ベクトル$A_s'$はＬＵＴカスケードの最終出力である。仮インデックスA'は、これらのベクトルの合成として表される。また、ベクトル$R_1,R_2,\cdots,R_{s-1}$は、ＬＵＴカスケードの中間変数を表す。
【００９６】
出力変数レジスタ１３は、各論理関数メモリ１２－１～１２－ｓが出力する中間出力$A_1',A_2',\cdots,A_{s-1}'$及び最終出力$A_s'$を保持し、仮インデックスA'として出力するレジスタである。
【００９７】
尚、ＬＵＴカスケード論理回路の詳しい動作については、特許文献４及び非特許文献６に記載されているため、ここでは説明を省略する。
【００９８】
図６は、図４の一致判定手段７の構成を表す図である。図６では、一例として、入力データXが８ビット、出力インデックスAが８ビットの場合を示しているが、入力データX及び出力インデックスAのビット数はこれに限られるものではない。
【００９９】
一致判定手段７は、入力データXの各ビットに対応して設けられたＥＸＮＯＲゲート２１、１個のＡＮＤゲート２２、及び出力インデックスAの各ビットに対応して設けられたＡＮＤゲート２３により構成される。
【０１００】
入力データX及び逆算データX'の各ビットx_i, x_i'（i=1,2,…,8）は、それぞれ対応するＥＸＮＯＲゲート２１に入力され、EXNOR演算が行われる。各ＥＸＮＯＲゲート２１の演算結果は、ＡＮＤゲート２２に入力され、AND演算が行われる。このＡＮＤゲート２２の出力Qを、一致判定信号と呼ぶ。

【０１０１】
一方、仮インデックスA'の各ビットa_1',a_2',…,a_8'は、それぞれ各ＡＮＤゲート２３の一方の入力ノードに入力される。また、各ＡＮＤゲート２３の他方の入力ゲートには、一致判定信号Qが入力される。
【０１０２】
以上のように構成された本実施例に係る連想メモリ１０について、以下その動作を説明する。
【０１０３】
まず、外部回路から入力データXが入力されると、簡略化関数演算部５は、ＣＡＭ関数G(X)の演算を行い、その結果を仮インデックスA'として出力する。ここで、入力データXに対する真のインデックスが存在する場合、仮インデックスA'の値は真のインデックスの値に一致する。一方、真のインデックスが存在しない場合には、仮インデックスA'の値はドント・ケアとなる。従って、仮インデックスA'は真のインデックスを含むものの、その値が真のインデックスを表すのかドント・ケアを表すのかの判別はできない。
【０１０４】
次に、仮インデックスA'は補助メモリ６に入力される。補助メモリ６は、仮インデックスA'に対してデータ出力関数$F^{-1}(A')$のＬＵＴ演算を行い、その結果を逆算データ$X'= F^{-1}(A')$として出力する。このとき、仮インデックスA'は真のインデックスを表す場合には、逆算データX'は入力データXと同じ値となる。しかし、仮インデックスA'がドント・ケアの場合には、仮インデックスA'はデタラメな値なので、逆算データX'には無効値が出力される。
【０１０５】
次に、一致判定手段７のｉ番目のＥＸＮＯＲゲート２１（i=1,2,…,8）には、それぞれ入力データXのｉ番目の成分x_iと逆算データX'のｉ番目の成分x_i'とが入力される。ＥＸＮＯＲゲート２１は、論理演算
【０１０６】
【数１０】

$$q_i = \overline{x_i \oplus x_i'} \qquad (10)$$

を行い、演算値q_iを出力する。演算値q_iの値は、成分x_iと成分x_i'とが一致する場合は１、一致しない場合は０となる。
【０１０７】
各演算値q_iは、ＡＮＤゲート２２に入力され、ＡＮＤ演算が行われる。これにより、入力データXと逆算データX'の各成分が完全に一致する場合には一致判定信号Qは１となり、それ以外の場合には一致判定信号Qは０となる。従って、この一致判定信号Qを用いることによって、仮インデックスA'が真のインデックス値を表しているか否かの判別ができる。
【０１０８】
ｉ番目のＡＮＤゲート２３（i=1,2,…,8）には、それぞれ仮インデックスA'のｉ番目の成分x_iと一致判定信号Qが入力される。そして、各ＡＮＤゲート２３はこれらのAND演算を行い、その結果を出力インデックスAのｉ番目の成分a_iとして出力する。これにより、仮インデックスA'が真のインデックス値を表している場合には、出力インデックスAとして真のインデックス値が出力され、それ以外の場合は出力インデックスAとして０が出力される。
【０１０９】
以上のようにして、入力データXに対するCAM関数の演算が行われる。上述のように、本実施の形態では、簡略化関数演算部５及び補助メモリ６はすべてメモリで構成されているため、通常のRAMを用いて構成することが可能である。従って、プロセスの微細化が容易であり回路規模を小さくすることができる。また、メモリであるため、使用しない期間を省電力状態として消費電力の節減を図ることも可能である。

【実施例２】

【０１１０】

本実施例２に係る連想メモリの全体構成及び一致判定手段７の構成は、図４，図６と同様であり、説明は省略する。実施例２に係る連想メモリは、簡略化関数演算部５の構成が実施例１とは異なる。

【０１１１】

図７は、実施例２に係る連想メモリの簡略化関数演算部５の構成を表す図である。本実施例の簡略化関数演算部５は、複数のｐｑ素子３１を樹形状に結合した構成からなる。ｐｑ素子３１は、ｐ入力ｑ出力のメモリである。各ｐｑ素子３１のｐ，ｑの値はそれぞれのｐｑ素子３１ごとに任意に設定される。各ｐｑ素子３１には、簡略化ＣＡＭ関数Gを関数分解して得られる部分関数G_1,G_2,…がＬＵＴとして格納されている。このように複数のｐｑ素子３１を樹形状に結合した回路をここでは「ｐｑ回路網」と呼ぶ。

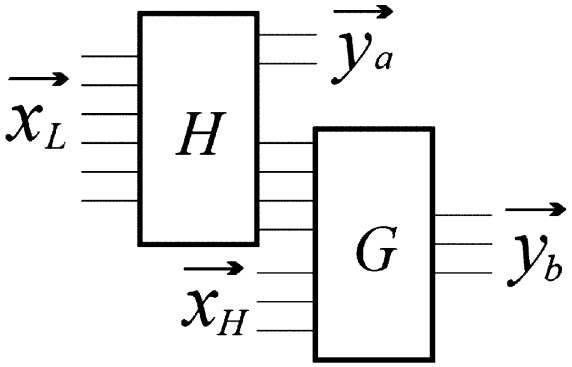
【０１１２】

尚、図７の構成は一例であり、ｐｑ素子３１の結合の仕方は、ＣＡＭ関数Gにより適宜選択される。

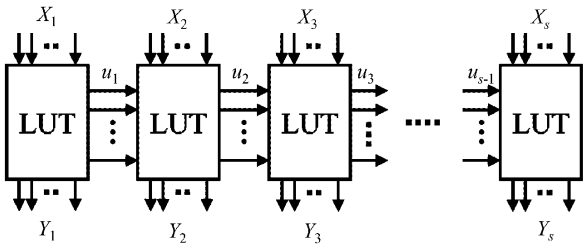
【０１１３】

このように、簡略化関数演算部５としてｐｑ回路網を使用しても、実施例１と同様の連想メモリを構成することができる。

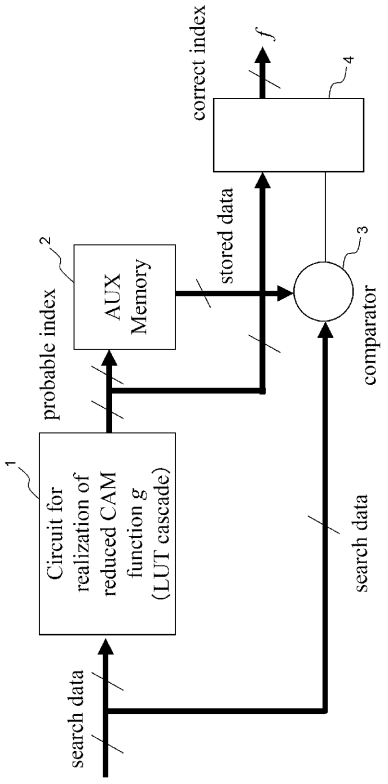
【図１】



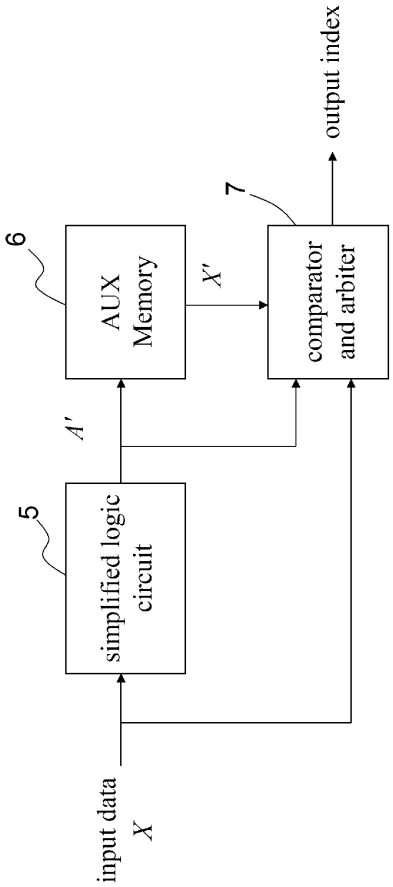
【図２】



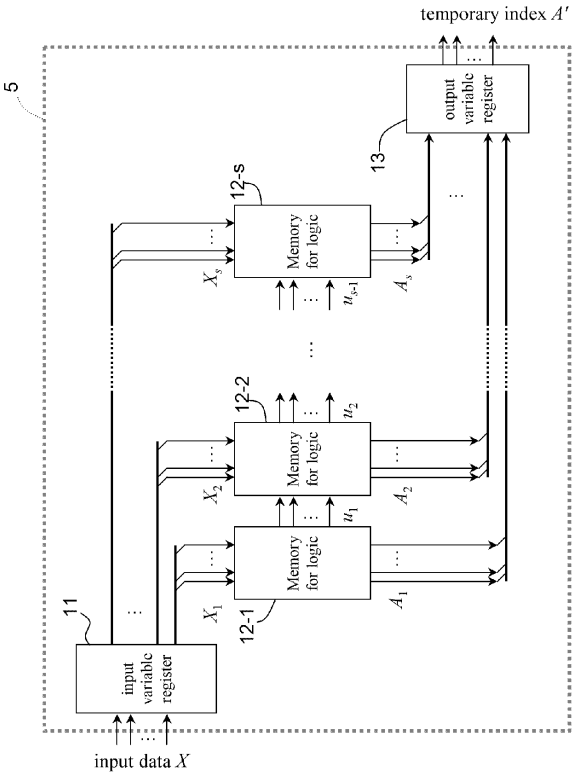
【図３】



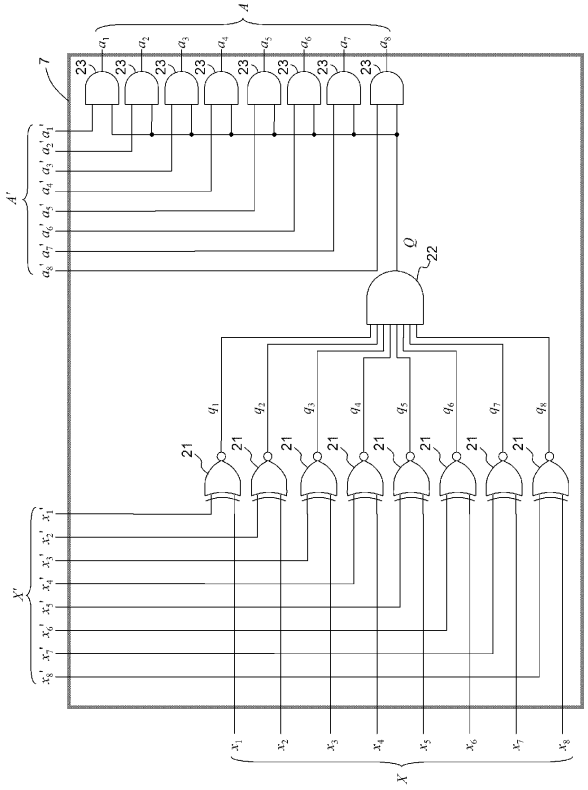
【図４】


output index
7
6
AUX Memory
X'
comparator and arbiter
A'
5
simplified logic circuit
input data X

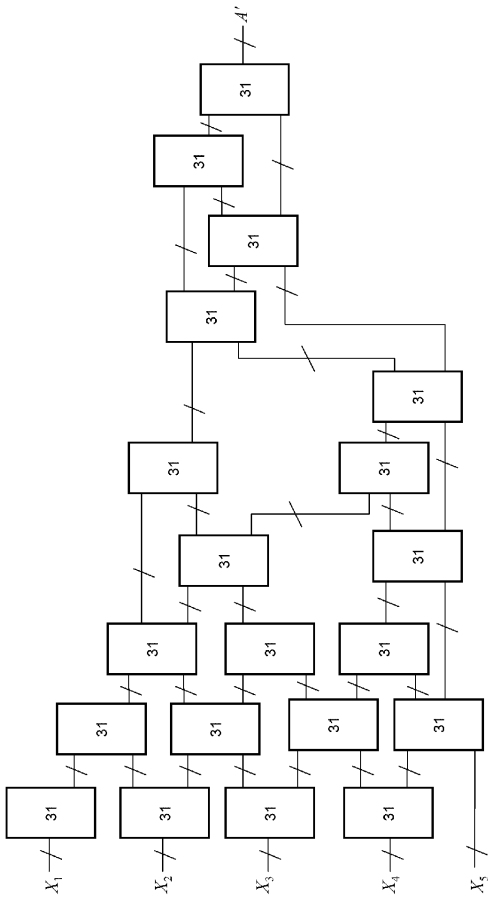
【図５】


temporary index A'
5
output variable register
13
12-s
Memory for logic
12-2
12-1
input variable register
11
input data X

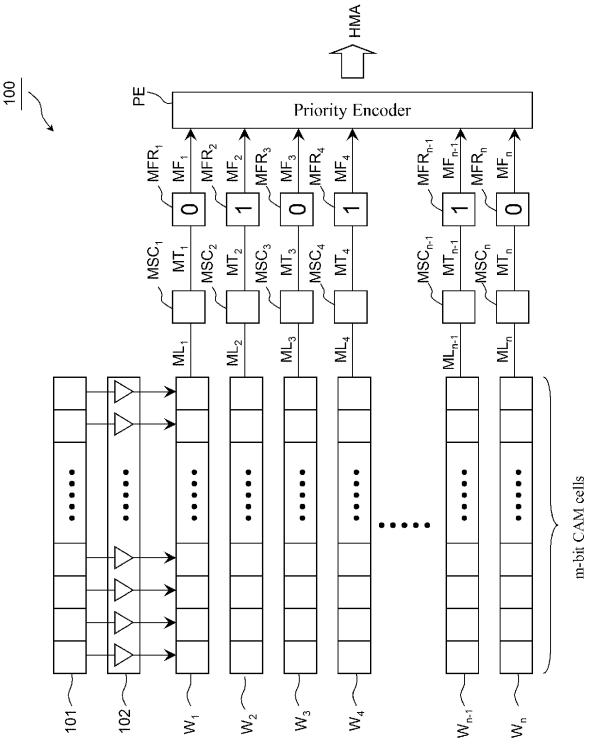
【図６】


A
7
23
A'
Q
22
21
X'
X

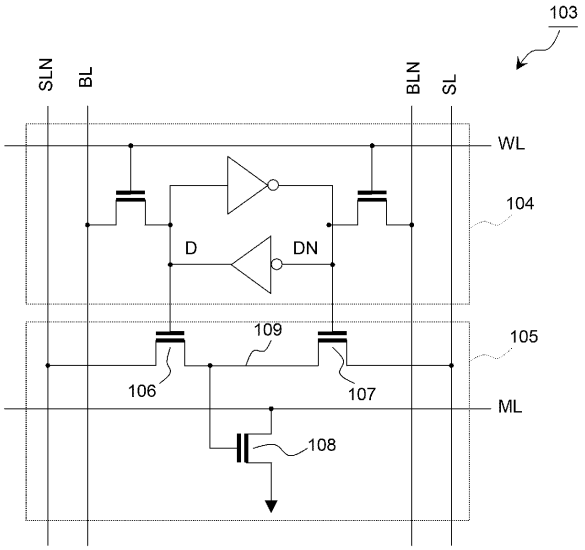
【図７】


A'
31
X1
X2
X3
X4
X5

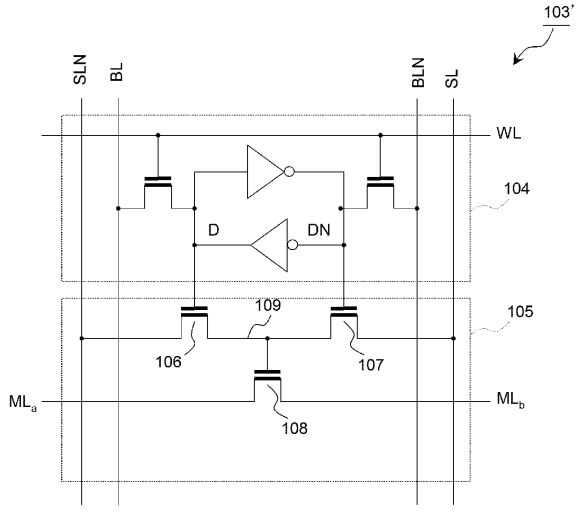
【図８】


HMA
100
PE
Priority Encoder
m-bit CAM cells

【図９】


103
SLN
BL
BLN
SL
WL
104
D
DN
109
105
106
107
ML
108

【図１０】


103'
SLN
BL
BLN
SL
WL
104
D
DN
109
105
106
107
108

フロントページの続き

(56)参考文献 特開２００４－２２９１６３（ＪＰ，Ａ）
特開２００２－３３４１１４（ＪＰ，Ａ）
特開２０００－１０５７７０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G11C 15/04
G06F 7/00
G06F 17/30