

内容検索メモリを用いた検索システムの速度性能の評価

正員 橋爪 正樹<sup>†</sup>      正員 山本 博資<sup>††</sup>  
 正員 為貞 建臣<sup>††</sup>      非会員 埴渕 敏明<sup>†††</sup>

Evaluation of a Retrieval System Using Content Addressable Memory

Masaki HASHIZUME<sup>†</sup>, Hirotsuke YAMAMOTO<sup>††</sup>, Takeomi TAMESADA<sup>††</sup>, *Members and Toshiaki HANIBUTI<sup>†††</sup>, Nonmember*

あらまし 半導体製造技術の発展により、従来に比べ安価に大容量 CAM を作製する技術的基盤ができています。しかし、我々が検索対象とするデータ量は急激に増加しており、すべての検索データを CAM 内に格納しておくことはコストの点から非現実的である。従って、CAM 内にすべての検索データを格納せず、検索データを別の記憶装置にブロック分割して格納しておき、各ブロックを順次 CAM に転送し、検索する利用方法が実用的である。このような利用形態では、データの転送ネックや処理のオーバーヘッドが生じ、どの程度の検索速度が得られるかが明確でない。そこで本論文では、CAM へのデータ転送を含めた CAM を用いた検索システムのモデリングを行い、検索時間を求め、速度性能を評価した。その結果、(1)CAM を用いた検索システムで高速性を得るには、CAM 自身の検索を高速化するより、データを高速に取り出すことやバス容量を増加させることなどによりデータ転送時間を減少させる方がより効果的であること、(2)検索データ量により高速検索のための最適な CAM の容量および形状が存在すること、(3)CAM を用いると CAM にデータを転送するための制御機構が必要となるが、その制御機構の速度によるシステム全体への影響は少ないことなどが明らかになった。

1. ま え が き

内容検索メモリ (Content Addressable Memory, CAM) は RAM のようなアドレスを用いた検索ではなく、検索データの内容によりメモリセル内のデータを並列検索できる記憶素子である。CAM は 1950 年代後半から画像処理、データベースマシンなどへの応用が考えられ、その処理アルゴリズムも検討されてきた。その当時は CAM として小容量なものしか開発できず、また RAM 等にくらべて 1 ビット当りのコストが非常に高価であったため、一部のシステムで用いられたに過ぎなかった。しかし VLSI 技術の発達により、比較的 low cost で 20 k ビット程度の大容量な半導体 CAM の

実現可能な基盤ができ、ますます CAM の利用が進むものと思われる<sup>(1),(6)</sup>。

しかし、低価格になったとはいえ、CAM は他の RAM などのメモリ素子に比べると、そのハードウェア量が多いため、まだ高価である。また、実用レベルにおいて CAM で処理したいデータ量は一般に CAM の容量より多い。従って、CAM を用いた検索システムでは、CAM は階層化されたメモリの一構成要素として使用するのが経済的であり実用的と考えられる。つまり、すべての検索すべきデータを高価な CAM 内に格納しておくのではなく、安価な RAM や外部記憶装置などに格納しておき、そのデータをいくつか分割して CAM に転送し、CAM で検索を行うのが当面の現実的な利用形態と考えられる。

この階層化メモリシステムでの検索速度に影響を与える要因は次のとおりである。なお、以後では検索対象データを格納したメモリから問合せデータに一致したデータを検索するための時間を「検索時間」と呼び、その逆数を「検索速度」と呼ぶ。

<sup>†</sup> 徳島大学工業短期学部電子工学科, 徳島市  
 Technical College of Tokushima University, Tokushima-shi, 770 Japan

<sup>††</sup> 徳島大学工学部電子工学科, 徳島市  
 Faculty of Engineering, Tokushima University, Tokushima-shi, 770 Japan

<sup>†††</sup> 三菱電機株式会社, 伊丹市  
 Mitsubishi Electric Corporation, Itami-shi, 664 Japan

(1) メモリが階層化されたこのシステムで高速性を引き出すには、CAMの一度の検索でできるだけ大量のデータを検索できることが望まれ、大容量かつ高速検索可能なCAMが要求される。

(2) CAMではCAMの容量単位で比較が行われるので、検索が進み、検索すべきデータ量が少なくなり、CAM容量以下になった場合にはCAMが大容量であるほどロスが生じ並列検索能力が活用できなくなる。

(3) CAMはメモリセル内のデータに対して並列に検索できる能力をもつが、メモリセルへのデータの格納および検索結果の取り出しはバス容量単位で逐次的に行われる。従って転送ネックが生じ、CAMを用いてもシステム全体の高速性は引き出せなくなる可能性がある。

(4) メモリを階層化したシステムでは外部メモリのどのデータをCAMに転送するかを制御するデータ転送制御機構が不可欠となる。この制御機構が低速であると、システム全体の検索速度は低下する。

しかし、現在までにCAMを用いたシステム全体の検索速度性能に上記(1)~(4)の要因がどのような影響を及ぼすのかが明確にされていない。また高速性を引き出すにはどうすべきかも明らかでない。

過去にはCAM自身の処理速度を評価したものはあるが、データ転送まで含めたCAMを用いたシステム全体の検索速度の評価は、STARANタイプの連想プロセッサ<sup>(2)</sup>を用いたデータベースマシンに対して検討されているのみである。そこでは、検索時間の大部分はCAMとのデータ転送時間になり、データを格納したディスクなどの外部記憶装置に対するデータの入出力時間をいかに短縮するかが、マシン全体の速度性能を決定することを、Berraらが報告している<sup>(3)</sup>。しかし彼らは完全並列方式でなく、ビットシリアル方式のSTARANタイプの連想プロセッサを対象としており、一般性に乏しく、また上記(1)~(4)の影響を考慮した評価が行われていない。

そこで、本論文では外部記憶装置とCAM間のデータ転送まで含めた検索システムの一致検索に関する速度性能を評価する。CAMは本論文で検討する一致検索だけでなく、あいまいな検索や画像処理などにも利用できる。しかし、一致検索以外の処理でも一致検索処理と同様な処理過程、つまり「外部メモリからCAMへのデータ転送、CAMでの検索、検索結果の転送」の三つの処理過程となる。更に、検索は他の処理の基本指標となるため、CAMを用いた検索システムの速度性能

を評価する。

本論文ではまず、CAMを用いた検索システムを一般化し、それに対して評価モデルを作成する。次にそのモデルから特性評価式を導出し、性能評価を行う。CAMの性能評価項目としては、あいまい性をもつ内容検索能力や並列処理能力が考えられるが、ここでは検索能力の一つである速度性能のみを検証する。以後、2.ではCAMの概要について、3.ではCAMを用いたシステムのモデリングについて述べ、4.で性能評価式の導出および性能評価結果について述べる。

## 2. 内容検索メモリ (CAM)

CAMは記憶機能だけでなく、論理機能を備えたメモリであり、連想処理に用いられるため連想メモリ (Associative Memory) とも言われる。論理機能として一致・不一致のほか greater than, less than, max, minなどを備えたCAM<sup>(4)</sup>も存在する。

CAMは機能的には図1に示すように次の四つの部分から構成される。

(1) データを格納する「メモリセル (memory array)」、

(2) 検索したいデータを格納する「問合せレジスタ (data register)」、

(3) 問合せレジスタ内のデータで検索したくないビットをマスクするための「マスクレジスタ (mask register)」、

(4) 検索結果を格納する「照合レジスタ (response store)」、

CAMのデータ検索動作の大略は次のようになる。CAMは、検索キーと一致するメモリセル内のデータを並列に検索することができる。検索キーは図1に示す

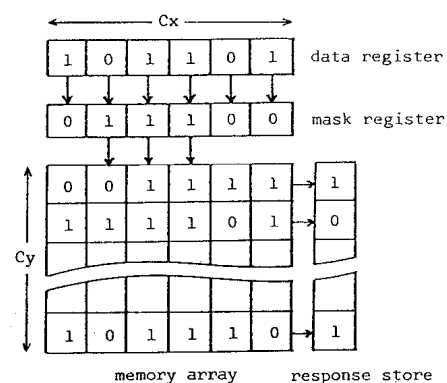


図1 CAMの機能ブロック図  
Fig. 1 Functional block diagram of CAM.

ように、問合せレジスタの内容(以後「問合せデータ」と呼ぶ)をマスクレジスタ内に格納した「マスクデータ」でマスクして作る。検索結果は照合レジスタに出力され、一致したデータに対応する照合レジスタの各ビットが1となる。

### 3. CAMを用いた検索システムの検索速度評価モデル

#### 3.1 評価モデル

本論文では図2に示すようなCAMを用いた検索システムのモデルを考える。本モデルでは、検索対象データ(以後「検索データ」と呼ぶ)はCAM以外の低価格大容量の記憶装置に格納しておく。検索はCAM容量単位(以下では「ブロック」と呼ぶ)で検索データを分割し、CAMに転送して検索する。この検索モデルは、記憶装置内のデータを小容量のCAMで分割検索し、図1と同じ処理を仮想的に実現したモデルであり、最も現実的なモデルと考えられる。

そこで図2の検索モデルを評価するために、図3に示す評価モデルを設定する。評価モデルは「検索データ格納装置(storage device)」、「CAM」、「CPU」および「バス」から構成し、それぞれ次のような役目をする。

[バス] CAMと検索データ格納装置間は幅 $B_w$ ビットのバスで結ばれ、バス容量 $B_w$ ビットを1サイクルでデータ転送できると仮定する。

[CAM] CAMはビット方向 $C_x$ でワード方向 $C_y$ 、つまり1ワード $C_x$ ビットのレコードを $C_y$ 個格納できる容量をもつものとする。また、CAMは照合レジスタに検索結果を $t_c$ サイクルで格納できると仮定する。CAM内に格納する検索データを高速に検索するために符号化する方法も考えられている<sup>(6)</sup>。しかし、その場合でもそれらはCAMで検索されるデータとなり、本モデルでモデル化できる。従って、ここでは検索データの内容を限定しない。なお、CAMでの検索の種類は一致検索に限定する。

[CPU] CPUはCAMと検索データ格納装置間のデータ転送を制御する機能を行う。

[検索データ格納装置] 検索データ格納装置は階層化された記憶装置<sup>(7)</sup>をモデル化したものである。ここでは検索データを格納し、その内容はブロックに分割され、CAMに転送される。階層化されたメモリ内には1レコード長 $T_x$ ビットのデータが $T_y$ レコード格納されているとする。また、検索すべきデータをCAMに転送する

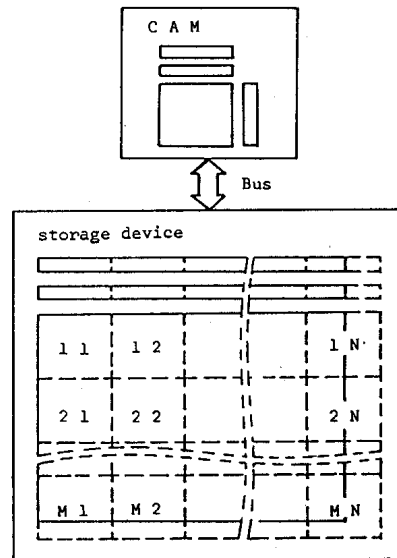


図2 CAMを用いたシステムの検索モデル  
Fig. 2 Retrieval model of CAM system.

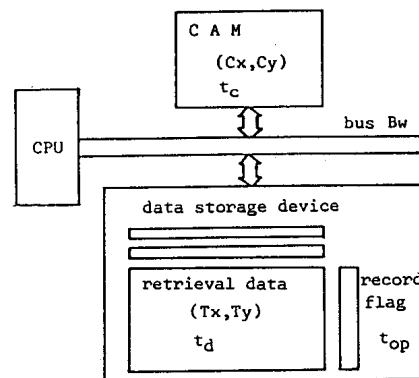


図3 評価モデル  
Fig. 3 Evaluation model.

ためにはそのデータを階層化されたメモリから取り出さねばならない。それに要する時間は階層化技術により、またデータによっても異なるが、ここでは簡単のため、1レコード当たり平均 $t_d$ サイクルとする。

CAMを用いた検索ではCAMにどのデータを転送すべきかのデータ転送制御機構が不可欠となる。そこで、検索データ格納装置内の各レコードに1ビットのフラグ(「レコードフラグ」と呼ぶ)を設け、このフラグが1のレコードを検索データとしてCAMに転送することにする。検索が進み、検索する必要のなくなったレコードを転送しないために、レコードフラグを0に書き換える必要がある。この書換えは、1レコード当たり $t_{op}$ サイクル要すると仮定する。なお、このデータ転送制御

機構はフラグを使用することなく実現可能であるが、本モデルのようにCAMの検索結果である照合レジスタの内容をCAMから検索データ格納装置に転送し書き換える方法がデータ転送量を最も少なくできるため、フラグを使用する。

### 3.2 検索アルゴリズム

図3のシステムの高速度性を評価するために、CAMのもつ並列検索能力をできる限り有効に利用し、かつ検索データ格納装置とCAM間のデータ転送量が少なくなる検索アルゴリズムを考える。図3のモデルでの検索アルゴリズムとして、検索データの列方向(ビット方向)を優先して検索する方法と行方向(ワード方向)を優先して検索する二つの方法が考えられる。

行方向優先のアルゴリズムは、図2において各ブロックを「 $11 \rightarrow 12 \rightarrow \dots \rightarrow 1N \rightarrow 21 \rightarrow 22 \rightarrow \dots \rightarrow 2N, \dots, M1 \rightarrow M2 \rightarrow \dots \rightarrow MN$ 」の順に検索する方法である。この方法ではブロック11を検索後、ブロック12のデータすべてをCAMで転送し検索する必要がない。そこで、ブロック11ではそのブロックの問合せデータと一致したデータのみをブロック12の検索時ではCAMに転送し検索するのが好ましい。しかしこの方法では、12, 13,  $\dots$ ,  $1N$ と検索が進むに従い、検索データがCAM容量より少なくなる。しかし、そうするとCAMのもつ並列検索能力を十分発揮できない。CAMの性能、特に並列処理能力を十分引き出すには、一度になるべく多くのデータを検索する必要がある。

一方、列方向優先の処理アルゴリズムは図2において、各ブロックを「 $11 \rightarrow 21 \rightarrow \dots \rightarrow M1 \rightarrow 12 \rightarrow 22 \rightarrow \dots \rightarrow M2 \rightarrow \dots \rightarrow 1N \rightarrow 2N \rightarrow \dots \rightarrow MN$ 」の順に検索する方法である。この方法によれば2列目以降では前列の検索で一致した解の候補と考えられるレコードのデータのみをCAMに転送すればよい。この方法は行方向優先のアルゴリズムに比べ、一度のCAMの検索でできる限り多くのデータを検索できるため、CAMの並列検索能力を最大限引き出すことができる。そこで、ここでは列方向優先アルゴリズムを検討する。

更に、図2のブロック $M1, MN, 1N$ のように検索データがCAM容量より少なくなる場合、不足分のデータをCAMに転送しない。またビット方向に余りが生じるブロック $1N$ のような場合はマスクデータは不足したビットに0を入れ、 $C_x$ ビットにして転送するが、問合せデータ、検索データはデータの存在する分のみを転送する。

以上より、次の列方向優先アルゴリズムを用いる。

[検索アルゴリズム] 検索データ格納装置に格納されている検索データを、CAM容量により $N \times M$ 個のブロックに分割し、各ブロックの内容を $T(i, j) (i=1, 2, \dots, M, j=1, 2, \dots, N)$ とする。問合せデータ、マスクデータは $C_x$ ビットごとに分割し、それらの内容を $D(j), M(j) (j=1, 2, \dots, N)$ とする。

(1)  $j=1$ とする。

(2)  $D(j), M(j)$ を検索データ格納装置からCAMに転送する。但し、 $M(N) < C_x$ ならば、「0」を追加し、 $C_x$ ビットにしたものをCAMに転送する。 $D(N) < C_x$ ならば、 $M(N)$ のようにデータを追加せず、CAMに転送する。

(3) 第 $j$ 列の検索対象となる検索データでまだ検索していないデータをCAMの容量分だけ検索データ格納装置からCAMに転送する。その際、CAMの容量を満たさない場合が生じても、余分なデータを追加せずにCAMに転送する。

(4) CAMで検索を行う。

(5) 照合レジスタの内容を検索データ格納装置に転送する。

(6) 第 $j$ 列での検索で検索すべきデータがなくなった場合または第 $N$ 列の検索が終了した場合、全処理を終了する。上記以外で、第 $j$ 列での検索対象となる検索データを検索し終わった場合、次の列の検索に移るため、 $j=j+1$ として(2)に戻る。それ以外の場合、(3)に戻る。

なお、上記アルゴリズムの第2列目以降の処理において処理(3)で検索データを転送する以前に、次のアルゴリズムでレコードフラグを更新する。また、 $T(1, 1)$ の検索前のレコードフラグはすべて1にしておく。

[レコードフラグ更新アルゴリズム] 図4に示すように、検索データ格納装置内の第 $j$ 列目の検索結果のレコードフラグが図4(a)であった場合、第 $j+1$ 列目の検索時には第 $j$ 列目のレコードフラグが1のレコードの検索

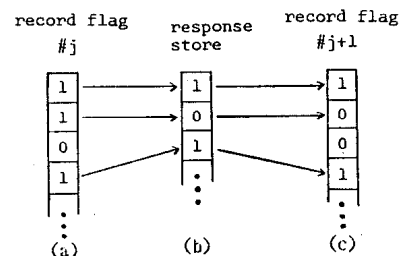


図4 レコードフラグ更新アルゴリズム  
Fig. 4 Record flag rewriting algorithm.

データのみをCAMに転送する。検索結果がCAMから検索データ格納装置内へ転送されたとき、第 $j$ 列目で1の立っているフラグとCAMから転送される照合レジスタの各ビットとの論理積演算の結果を第 $j+1$ 列の検索結果のレコードフラグとする。

#### 4. システム評価

##### 4.1 性能評価式の導出

検索データ格納装置内には問合せデータ、マスクデータが格納されているとする。その時点から検索データをCAMに転送し、問合せデータと一致するデータのレコードフラグが求まるまでの時間 $T$ を導出する。そのため、次の①～⑤の仮定を設ける。

① CAMと検索データ格納装置間のデータ転送に要する時間を基本サイクル(1サイクル)として、検索時間 $T$ を計算する。

② 大きさ $C_x \times C_y$ のCAMで一致検索を行う。またCAMに検索データを格納してから、照合レジスタに検索結果が出力されるのに必要な時間は、 $t_c$ サイクルとする。

③ 検索データ格納装置に格納している検索データの大きさは $T_x \times T_y$ とする。検索データ格納装置は検索データを記憶する以外に、CAMの照合レジスタの内容を基にレコードフラグを更新でき、更にこのレコードフラグを基に1レコード当り平均 $t_d$ サイクルでCAMに転送すべき検索データを検出できる機能をもつ。なお、この検出時間の総和は各検索列で一定とする。

④ バス容量を $B_w$ ビットとし、検索データ格納装置とCAM間では $B_w$ ビットのデータが並列にデータ転送できる。

⑤ レコードフラグの更新処理は、CAMへのデータ転送やCAMでの検索処理と時間的に並列処理可能とする。このレコードフラグ更新処理に要する時間は1レコード当り $t_{op}$ サイクルとする。

ここでは、CAMへのマスクデータ、問合せデータ、検索データの転送、検索、結果の格納に要する時間 $T_s$ と、レコードフラグ更新時間 $T_{rew}$ を別々に導出する。なお、以下において $[x]$ は $x$ 以上の最小の整数を、 $[x]$ は $x$ 以下の最大の整数を、 $R(x, y)$ は $x$ を $y$ で割ったときの余りをそれぞれ表す。

図2において第 $j$ 列の検索に要する時間を $T_s(j)$ とすると、 $T_s$ は各列の検索時間から、式(1)で求めることができる。

$$T_s = T_s(1) + T_s(2) + \dots + T_s(N) \quad (1)$$

ここで $N$ は式(2)から求められる。

$$N = \lceil T_x / C_x \rceil \quad (2)$$

但し、ある第 $k$ 列の検索で検索すべきデータ、つまり解の候補となるレコードがなくなった場合、 $T_s(j) = 0$  ( $j = k+1, \dots, N$ )とする。

$D(j)$ および $M(j)$ をCAMに転送するのに要する時間はそれぞれ $T_d(j)$ 、 $T_m(j)$ とすると、次の式(3)、(4)で表せる。

$$T_d(j) = \begin{cases} \lceil \{T_x - (j-1)C_x\} / B_w \rceil, & T_x < jC_x \text{ のとき} \\ \lceil C_x / B_w \rceil, & T_x \geq jC_x \text{ のとき} \end{cases} \quad (3)$$

$$T_m(j) = \lceil C_x / B_w \rceil \quad (4)$$

第 $j$ 列で解の候補となるレコード数を $N_r(j)$ とすると、第 $j$ 列の検索データをCAMに転送・検索し、検索結果を検索データ格納装置に転送するのに要する時間 $T_s(j)$ は次の式(5)から求めることができる。

$$T_s(j) = \begin{cases} T_d(j) + T_m(j) + T_{sn}(j) + T_{slast}(j), & R(N_r(j), C_y) > 0 \text{ のとき} \\ T_d(j) + T_m(j) + T_{sn}(j), & R(N_r(j), C_y) = 0 \text{ のとき} \end{cases} \quad (5)$$

ここで $T_{sn}(j)$ 、 $T_{slast}(j)$ は次式で与えられる。

$$T_{sn}(j) = \{T_d(j)C_y + T_y t_d C_y / N_r(j) + t_c + \lceil C_y / B_w \rceil\} \times \lceil N_r(j) / C_y \rceil \quad (6)$$

$$T_{slast}(j) = T_d(j)R(N_r(j), C_y) + T_y t_d R(N_r(j), C_y) / N_r(j) + t_c + \lceil R(N_r(j), C_y) / B_w \rceil \quad (7)$$

レコードフラグ更新手続きはCAMへのデータ転送や検索および結果の格納の処理と並列に実行でき、またCAMとのデータ転送時間に比べ短時間で行えるので、レコードフラグ更新時間が実質的に零となる列が多い。しかし、検索が進むに伴い、第 $j$ 列の最後のCAMによる検索結果を待たないと、 $N_r(j+1) \geq C_y$ とならない場合には、次の式(8)に示す時間が必要となる。この現象は検索列が第 $N$ 列に近づき、その列の $N_r(j)$ がCAM容量より少なくなった場合によく生じる。

$$T_{rew}(j) = \begin{cases} t_{op} C_y, & R(N_r(j), C_y) = 0 \text{ のとき} \\ t_{op} R(N_r(j), C_y), & R(N_r(j), C_y) \neq 0 \text{ のとき} \end{cases} \quad (8)$$

また、最後の列の検索が終了し、その検索結果からレコードフラグを更新する必要があるため、 $T_{rew}$ は次の式(9)で求められる。

$$T_{rew} = T_{rew}(1) + T_{rew}(2) + \dots + T_{rew}(N) \quad (9)$$

但し、ある第  $k$  列の検索で検索すべきデータがなくなった場合、 $T_{rew}(j) = 0$  ( $j = k+1, \dots, N$ ) とする。

全体の検索時間  $T$  は式(1), (9)より式(10)で求められる。

$$T = T_s + T_{rew} \quad (10)$$

ここで、多くの場合  $T_{rew}(j) = 0$  となり、 $T_{rew}$  は無視することができる。また、 $D(j), M(j)$  ( $j = 1, 2, \dots, N$ ) の転送に要する時間も検索データ量に比べ少ないため、無視することができる。更に、図2に示すようなCAMの大きさとの不整合から生じる並列検索能力の低下による影響も無視することができる。

そこで全体の検索時間は式(11)に示すように、検索すべきデータを取り出すために要する時間  $T_p$ 、検索データ格納装置からCAMへ検索データを転送するために必要な時間  $T_{dt}$ 、検索に要する時間  $T_{ss}$ 、検索結果を検索データ格納装置に転送するために要する時間  $T_{rt}$  から次式により近似的に求めることができる。

$$T \approx T_p + T_{dt} + T_{ss} + T_{rt} \quad (11)$$

ここで

$$T_p = (T_x / C_x) T_y t_d \quad (12)$$

$$T_{dt} = N_s C_x / B_w \quad (13)$$

$$T_{ss} = (N_s / C_y) t_c \quad (14)$$

$$T_{rt} = N_s / B_w \quad (15)$$

なお式(13)~(15)における  $N_s$  は各列の検索レコード数の和であり、式(10)から求めることができる。

$$N_s = N_r(1) + N_r(2) + \dots + N_r(N) \quad (16)$$

#### 4.2 検索システムの性能評価

データ転送に要する時間は転送データの読み出し、書き込みの2度のメモリオペレーションに要する時間で近似できる。従って、メモリに対するアクセスタイムを  $t_m$  とすると、1回のデータ転送に要する時間つまり1サイクルは  $2t_m$  で近似できる。また、CAMの検索速度はメモリのアクセスタイムとほぼ同じ程度であるため、 $t_c = t_m$  と考える。レコードフラグ更新処理は論理演算に要する時間が  $t_m$  に比べ短いため、検索前のレコードフラグの読み出しと更新したレコードフラグの書き込みの2回のメモリオペレーションで近似でき、 $t_{op} = 2t_m$  と考える。また、 $t_d$  は階層化された検索データ格納装置においてヒット率が90%と仮定すると、 $0.1 t_m$  となる。従って、以下の評価では  $t_{op} = 1$  (サイクル)、 $t_c = 0.5$  (サイクル)、 $t_d = 0.05$  (サイクル) として評価する。

図2の各ブロックで解の候補となるレコードの確率  $\alpha$  はブロックの位置によらず一定とする。実用的なデータの場合、 $\alpha$  の値は列の位置により大きく変化すると思

われるが、ここでは基本的な性能を調べるため、定量的な評価が簡単に行えるように  $\alpha$  を一定とする。この場合、 $N_r(j)$  は次の式(17)で表すことができる。

$$N_r(j) = \alpha^{j-1} T_y \quad (17)$$

なお、評価時には  $\alpha$  は検索キーと一致するレコードの総数が  $A$  となるように設定した。

例として  $C_x = C_y = 128$  (bit) のCAMによる検索で、各処理が占める割合を式(10)を用いて求めた結果を図5に示す。図5(a)において  $T_1$  はCAMによる検索時間、 $T_2$  は検索結果の格納に要する時間、 $T_3$  は  $D(j)$  と  $M(j)$  の転送に要する時間、 $T_4$  はレコードフラグの更新に要する時間の各割合を示している。また図5(b)の  $T_5$  は検索データを取り出すのに要する時間を、 $T_6$  は検索データをCAMに転送するのに要する時間を、 $T_7$  は式(11)で

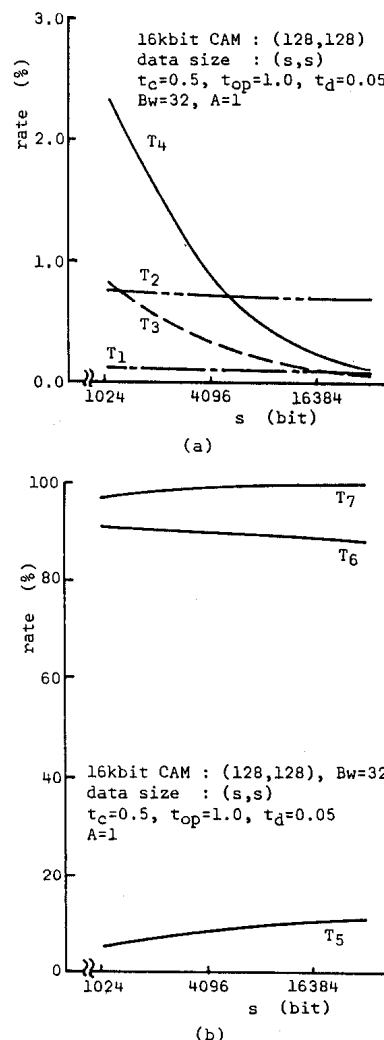


図5 各処理の占める割合  
Fig. 5 Rate of each operation.

考慮した処理に要する時間 ( $T_1 + T_2 + T_5 + T_6$ ) を示す。図 5 より、処理の大部分は CAM と検索データ格納装置間のデータ転送で、CAM による検索時間の占める割合は非常に少ない。そのため、CAM の検索速度がシステム全体の検索時間に与える影響は少なく、また図 2 の  $M1$  や  $1N$  のような不整合ブロックによる CAM の検索能力の低下の影響は少ない。更にレコードフラグ更新処理は検索データ量が多くなるとその影響が少なくなることがわかる。従って、システムの検索時間は正確な式(10)の代わりに式(11)で近似的に求められることがわかる。このことは、図 5(b) の  $T_7$  から明らかである。

式(11)より、システム全体の検索時間を短くするためには、次の 4 通りの方法が考えられる。

- (1) 検索データ格納装置からのデータの取り出しを高速化し  $T_p$  を小さくする、
  - (2) バス容量を増加させ、データ転送時間 ( $T_{dt} + T_{rt}$ ) を小さくする、
  - (3) 大容量の CAM を用いて  $T_p$  と  $T_{ss}$  を小さくする、
  - (4) データ転送量に対応する  $N_s$  を減少させる、
- 以下では各方法についてその効果を評価する。

データ取出し時間  $t_d$  による特性を図 6 に示す。メモリを階層化すればヒット率が最悪の場合でも  $t_d = 0.5$  (サイクル) となるため、図 6 より  $t_d$  による影響は少ない。しかし、階層化しない場合、 $t_d$  は  $10^3 t_m \sim 10^4 t_m$  程度になるため、 $t_d$  による影響が顕著に現れてくる。

図 7 にバス容量を増加させた場合の特性を示す。図 7 より、バス容量の増加による高速化効果が大きいこと

がわかる。

図 8 に一定量の検索データを、CAM 容量を変化させて検索した場合の結果を示す。また図 9 には図 8 の  $T_x = 32768$  (bit) のデータに対する各処理に要した時間を示す。図 8 より大容量 CAM を用いても、高速にはならないことがわかる。その理由は図 9 でデータ転送時間  $T_{dt}$  が大容量 CAM になると増加していることから、小容量 CAM を用いた場合に不一致となって以後転送されないデータも、大容量 CAM を用いた場合に転送する必要が生じ、 $N_s$  が増加したためである。このように図 2 のようなシステムでは  $N_s$  がシステム全体の検索時間に大きな影響を与えることがわかる。逆に図 9 から

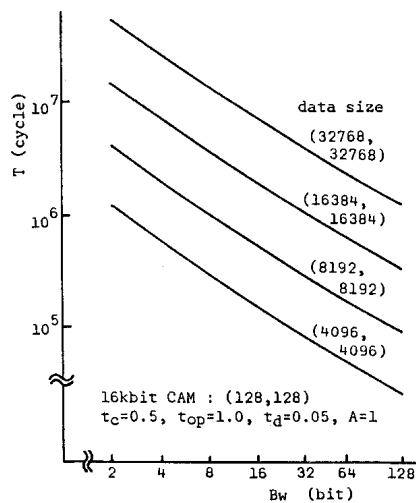


図 7  $B_w$  と  $T$  との関係  
Fig. 7 Relationship between  $B_w$  and  $T$ .

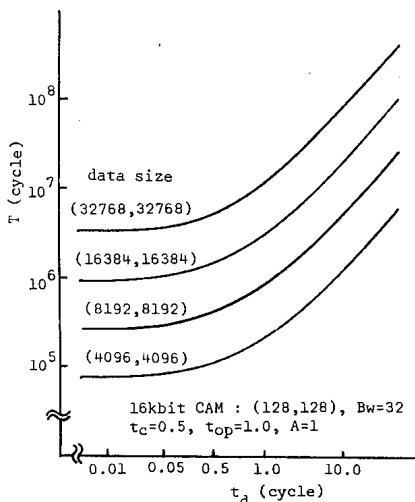


図 6  $t_d$  と  $T$  との関係  
Fig. 6 Relationship between  $t_d$  and  $T$ .

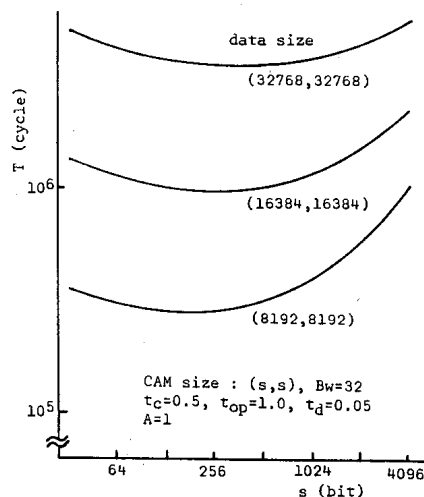


図 8 CAM の容量と  $T$  との関係  
Fig. 8 Relationship between CAM size and  $T$ .

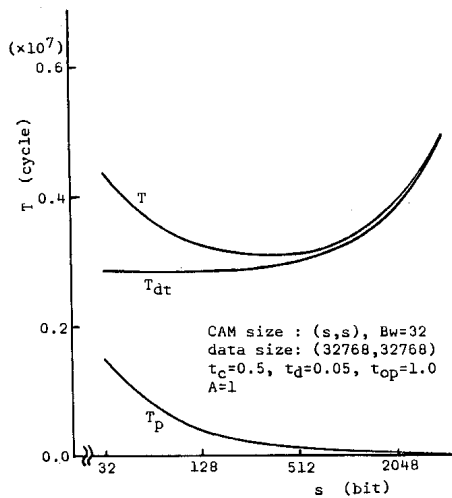


図9 CAMの容量とデータ転送時間との関係  
Fig. 9 Relationship between CAM size and data transfer time.

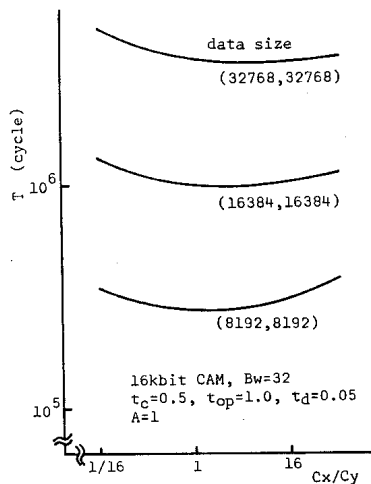


図10 CAMの形状とTとの関係  
Fig. 10 Relationship between  $C_x/C_y$  and  $T$ .

わかるように、小容量CAMではCAMに転送するデータの取り出し時間が増加し、検索時間が増加する。

同様な現象がCAMの容量が同じでもその形状つまり  $C_x/C_y$  を変えることにより生じる。このことは図10に示すように、同じ量の検索データを16 kbit CAMの形状を変えて検索した結果からも明らかである。従って、図9および図10からも明らかのように、検索データ量により、最短時間で検索可能なCAMの容量および形状が存在することがわかる。

## 5. むすび

本論文では、CAMへのデータ転送を含めたCAMを

用いた検索システムのモデリングを行い、処理時間を求め、速度性能を評価した。

その結果、以下のことが明らかになった。

(1) CAMを用いた検索システムで高速性を得るにはCAMの検索速度の高速化、CAMの大容量化、データ転送量の減少、データ転送の高速化が予想される。しかし、本性能評価例により、CAMを用いたシステムの処理時間の大部分はCAMとデータ格納装置間のデータ転送時間であり、システム全体を高速化するにはCAM自身の検索を高速化するよりも、データを高速に取り出すこと、バス容量を増加させデータ転送時間を短縮すること、データ転送量を減少させることの方がより効果的である。

(2) データ転送量は検索データが同じでも、CAM容量およびその形状により変化する。そのため、高速に検索するための最適なCAMの容量および形状が存在する。

(3) CAMを用いると次に検索すべきデータを転送するための制御機構が必要となるが、その処理はデータ転送処理と並列処理が可能で、その制御機構の速度によるシステム全体への影響は少ない。

本論文では簡単のため、一度のCAM検索で各レコードの解候補となる確率  $\alpha$  が一定と仮定して、CAMを用いた検索システムの評価を行い、上記のような基本的な特性を知ることができた。しかし、実際のデータでは図2の各列で  $\alpha$  が大きく変化する場合がある。このような場合には、 $\alpha$  が小さな列から優先的に検索した方が検索時間を短くできることが予想される。今後このような  $\alpha$  の変化による影響を調べる必要がある。また、CAMを用いた検索システムは内容により検索が行え、また検索キーを容易に変更でき、ハッシュ法では容易に行えない検索が行えるという特徴がある。CAMはこの特徴をいかしたシステムに多く導入されることが期待される。そこでCAMのもつ検索能力以外の処理能力を利用した画像処理用のCAMシステムなどの性能評価を、今後行う必要もあると思われる。

**謝辞** 本論文査読者には貴重な助言を頂きました。査読者に心から謝意を表します。

## 文 献

- (1) 市川, 平川: "連想処理とデータベース", 情報処理, 23, 8, pp. 748-756 (昭57-08).
- (2) J. A. Rudolph: "A production implementation of an associative array processor—STARAN", Fall Joint Computer Conference, 41, pp. 229-241 (1972).
- (3) P. B. Berra and E. Oliver: "The role of associative



array processors in data base machine architecture”,  
Computer, 12, 3, pp. 53-60 (1979).

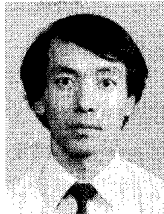
- (4) S. S. Yau and H. S. Fung : “Associative processor architecture—a survey”, Computing Surveys, 9, 1, pp. 3-28 (1977).
- (5) C. S. Roberts : “Partial-match retrieval via the method of superimposed codes”, Proc. IEEE, 67, 12, pp. 1624-1642 (1979).
- (6) 小倉, 山田 : “連想メモリ LSI の現状と今後”, 信学誌, 69, 7, pp. 745-751 (昭 61-07).
- (7) 高森, 伊藤 : “メモリシステム技術”, 情報処理, 27, 6, pp. 582-592 (昭 61-06).

(昭和 61 年 9 月 10 日受付, 62 年 3 月 23 日再受付)



埴淵 敏明

昭 57 徳島大・工・電子卒, 昭 59 同大大学院修士課程了。同年三菱電機株式会社入社。在学中, 連想記憶方式に関する研究, CAM の応用に関する研究に従事。



橋爪 正樹

昭 54 徳島大・工・電気卒, 昭 56 同大大学院修士課程了。同年日本電信電話公社武蔵野電気通信研究所入所。昭 58 徳島大工業短期大学部助手, 現在に至る。回路の論理設計用 CAD に関する研究に従事。



山本 博資

昭 50 静岡大・工・電気卒, 昭 55 東京大学大学院工学系研究科博士課程了。同年徳島大学工学部電子工学科助手。現在, 同助教授。主に通信路符号, 情報源符号, 暗号などにおけるシャノン理論の研究, および電子回路, マイコン応用などの研究に従事。

IEEE 会員, 工博。



為貞 建臣

昭 38 徳島大・工・電気卒, 昭 41 同大大学院・修士・電気専攻了。同年同大・電気工学教室助手。以後, 同電子工学教室助手, 講師, 助教授を経て, 現在同教室電子回路工学講座教授, 工博(京都大学)。主な研究テーマ: フリップ・フロップ回路の臨界トリガパルスと転移機構, 三相無・単・双・三安定回路, 多相マルチバイブレータ, 準安定状態をもつ順序回路の解析法と合成法, 電動義手の開発, 誘発筋電位の測定とその側弯症進行予測と治療への応用, 論理ゲート IC を用いたマルチバイブレータの構成法・動作解析・特性改善。興味あるテーマ: 超高速マルチの開発, 高速演算回路の開発, デジタル回路の CAD など。著書「電子回路 I」(朝倉書店), 「電子回路 II」(同), 「電子計算機 I」(同) などすべて共著で 5 冊, IEEE, 日本 ME 学会各会員。