SIMD 型並列プロセッサを用いたフルテキスト検索†

宮原末治†† 近藤利夫†† 多田俊吉†††

文書データの効率的な運用を目指し，これまで文書画像処理や文字認識処理エンジンとして利用してきた小型並列プロセッサ (LISCAR) 上に，高速のフルテキスト検索機能を実現した．LISCAR は1ビットのプロセッサエレメント (PE) を64個搭載した AAP 2-LSI 4 個からなる256並列の SIMD 型2次元アレイプロセッサである．検索システムは，この LISCAR とホストコンピュータ（パソコン）で構成しており，あらかじめパソコンから検索プログラムと文書データとをロードしておき，必要に応じて検索単語と検索条件とを入力することで検索を実現している．LISCAR によるフルテキスト検索としては，文書データの水平格納によるビットシリアル型の処理方式を中心に検討した．実験では日本語文書データを用い検索処理の基本となる完全一致と部分一致の検索速度を評価した．その結果，完全一致検索では，①全文字照合（総当たり照合）の速度は文書データ量と検索単語の語長に比例し，4文字単語では2.2千万字/秒，②絞り込み照合（不一致後は次の文字列に移る照合）の速度は検索単語の語長にはほとんど依存せず，文書データ量に比例し，出現頻度の高い4文字単語の検索例では実効4千万字/秒程度になることがわかった．さらに，部分一致検索では異字許容照合や単語内ワイルドカード照合が，それぞれ全文字照合や絞り込み照合とほぼ同程度の速度で検索できることを示した．

1. はじめに

特許や文献，記事情報などを対象にした従来型の文書データベースでは，検索を容易にするため文書の分類やキーワード付けなどの事前作業が必要で，この作業に多大な労力を要している．そのため事前作業の不要なフルテキスト型の検索方式が注目されている[1]．この方式は必要と思われる文書情報を事前作業なしにコードの型でデータベース化しておき，情報が必要になった時点で，そのつど全データをサーチすることで，所望の情報を得ようとするものである．この方式をキーワード付きのデータベース検索方式と比較すると，

①データベースの構築が容易である．

②任意の言葉を用いて検索できる．

などの利点がある．しかし，一方では，

③データベース内のすべての文書データをサーチするので処理時間がかかる．

④1回の検索要求で，意味内容が一致，あるいは類似した所望の文書データの得られる割合が少ない．

などの問題点が指摘されている．もっとも④の問題については，検索と確認とを対話的に繰り返し行うことで解決できる部分がある．これは検索結果を検索条件にフィードバックし，検索を繰り返すことで，より多くの有用な情報を得ることができるからである[2),3)]．しかし，この方法は検索を繰り返すので③の処理時間の問題はますます重要になる．このため，専用のハードウェアによってフルテキスト検索を高速化する試みが報告されている[3)~6)]．しかし，文献 3)，4) ではシステム規模が大きく高価なものになる，文献 5)，6) ではデータ転送速度がネックになりシステムとしての性能が十分に発揮できない，などの問題があり適用域が限定されていた．特に，本論文と同種の試みに文献3）の“SIMD 型並列プロセッサによる文書検索”の例があるが，そのシステムはフルテキスト型の文字列検索では不要なハイパキューブ形のネットワークや高速の積和演算機能を有しており，その分高価なものとなっている．

筆者らは柔軟な構成と，小型・経済性を両立できるフルテキスト検索システムを実現するため，これまで文書画像処理や文字認識処理に利用してきた SIMD 型小型並列プロセッサ LISCAR (LIne Scannable Cellular ARray processor)[7] を，文字列検索に応用することを試みた．LISCAR は1チップに 64 PE を搭載したフルカスタム LSI と，安価な DRAM とをベースにして1ボード化することで，小型化，経済化をはかっている．本論文ではこれを用いたフルテキスト検索システムの構成と特長，検索処理のアルゴリズムと速度性能，検索法とその評価について述べる．

† Full-Text Retrieval Using a SIMD Parallel Processor by SUEHARU MIYAHARA, TOSHIO KONDO (Human and Multimedia Laboratory, NTT Human Interface Laboratories) and SYUNKICHI TADA (NTT Intelligent Technology Corporation).

†† NTT ヒューマンインタフェース研究所マルチメディア処理研究部

††† NTT インテリジェントテクノロジー(株)

2. 検索システムの構成

検索システムは**図1**に示すように，ホストコンピュータとしてのパーソナルコンピュータ(PC)にSCSIインタフェースを介して LISCAR ユニットを接続することで構成している．ユニットの構成単位である LISCAR ボードは，256個の1ビットのプロセッサエレメント PE (Processor Element) から成る SIMD 型の並列プロセッサであり，B4 判サイズの基板1枚で実現されている．このユニットは PC のバックエンドプロセッサとして動作し，ホストからプログラムやデータを受け取り，必要な処理を実行した後，結果をホストに引き渡す．LISCAR はそれ自体の汎用性に加え，ボードやユニット単位での増設が可能であり，文書データの容量拡大や検索の高速化などの要求に対しても柔軟に対応することができる．本システムの主な仕様を**表1**に示す．

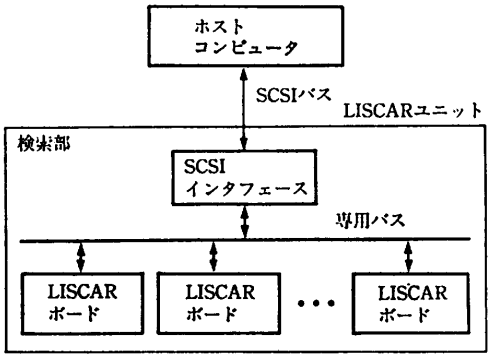


図1 検索システムの構成
Fig. 1 Retrieval system configuration.

表1 検索システムの主な仕様
Table 1 Main specifications of the retvieval system.

項目	内容	
検索部 (LISCAR)	基本ハードウェア	・256 PE による並列処理 (AAP2-LSI を4個使用) ・マシンサイクル：143 nsec
		アレイデータメモリ容量：32 MB/ボード
		プログラムメモリ容量：640 KB
	制御スカラ演算器	16 ビット固定小数点 DSP
	マイクロ命令長	80 ビット/インストラクション
	ユニットの大きさ	440×440×80 mm³
ホスト (PC 98)	- OS	MS-DOS
	I/F	SCSI
	CPT，キーボード	

2.1 LISCAR ボードの構成

LISCAR ボードは**図2**に示すように，プロセッサアレイ部，制御スカラ演算器，アドレス生成器，プログラムメモリ，データメモリ，アレイデータメモリ (ADM)，インタフェース等から成っている．この中でプロセッサアレイ部には，64個の1ビット PE を内蔵した AAP2-LSI[8] を4個搭載して，256 個の PE が並列動作する構成を採っている．データメモリはスカラ演算用のメモリであり，ADM はプロセッサアレイ部を構成する各 PE のローカルメモリ配列である．

(1) プロセッサアレイ

プロセッサアレイ部は 16×16 の2次元のアレイ構成をとっており，各 PE は隣接する8方向の PE と接続している．さらに，各 PE の上端と下端との間，左端と右端との間はそれぞれループ状に接続している．左右間はループを1段ずつずらした段違い接続とし，2次元のアレイ全体がスパイラル上に巻かれた1次元のアレイとしても機能するようになっている．この構造によって 16×16 サイズのプレーンを単位とする並列演算モードと，256

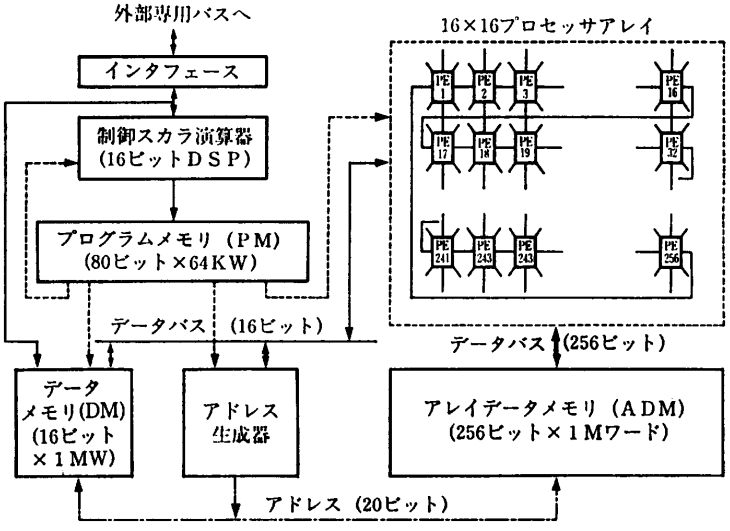


図2 並列プロセッサ LISCAR の構成
Fig. 2 Parallel processor LISCAR configuration.

長のラインを単位とする並列演算モードの両方の処理が可能である．

（2） プロセッサエレメント (PE)

1ビット幅の ALU，144 ビットのレジスタファイル (RF) 等から成る演算部と，セレクタやレジスタ等から成る PE 間データ転送部で構成される（図3）．

2.2 検索に必要な LISCAR の機能

パラレル-シリアル変換：本処理はデータメモリからプロセッサアレイ上の各列に文字単位で送り込まれたパラレルデータを，2次元モードで動作させて各パラレルデータの下位ビットから順に所定の行の 16 個の PE の ADM 上に流し込む処理である．この処理によって，16文字分のシリアルデータが所定の行の PE の ADM 内に作られる．この処理を16回繰り返すことで，PE ごとに1文字分のシリアルデータを転送することができる．

2.3 LISCAR の基本性能

LISCAR の基本性能を表2に示す．LISCAR は1ビットプロセッサで構成されているため，ビット単位の処理，例えば，算術論理演算や左右方向のシフトでは，実効速度が，それぞれ 1,600 MOPS，450 MOPS と極めて高い性能を示す．また，8ビット語長の演算でも乗除算が入らない場合，例えば，要素の語長が8ビット間の差の絶対値の累積では 100 MOPS とかなりの高い性能が得られる．

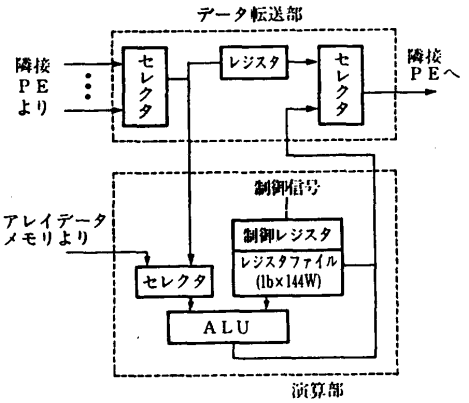


図3 プロセッサエレメントの構成
Fig. 3 Processor-element configuration.

表2 LISCAR の基本演算性能
Table 2 Basic performance of the LISCAR.

演算内容	所要時間 (nsec)	演算性能 (MOPS)
ビットライン間の算術論理演算	160	1,600
ビットライン間の左右シフト	570	450
グレイライン間の差の絶対値の累積*	7,300	100

* 要素の語長は8ビット．

3. フルテキスト検索

文献 4)～6) の検索手法は文字列照合用の専用のハードウェアの中に，文書データを1文字ずつ流し込み，比較器ごとに用意された複数の検索単語と比較し，一致した文字列を検出する手法である．これは並列処理による本手法とは異なり，データの流し込み速度や比較器の個数で検索性能が決まる．文献 3) の検索処理は本論文と同種の並列処理によるものであり，検索方式はプロセッサ内のメモリに対し，単語情報を PE の深さ方向に格納しておき，検索要求に対して検索単語と一致する単語を検出する方法である．そのため文書データに対して単語の切り出しや格納用のデータ変換処理などの前処理を必要としている．

ここで提案する文字列検索方式は，LISCAR のハード構成を生かしたもので，文書データを文字ごとに PE の深さ方向に割り付け，文字列が PE と水平方向に並ぶように ADM に格納し，1次元のライン処理と2次元のプレーン処理により，前処理なしで文書データを検索できるようにしたものである．ここでは提案した検索方法を用いて，完全一致検索と部分一致検索の実現法について考察し，その速度・性能を明らかにする．また，照合方式では全文照合と絞り込み照合とについて，その得失を比較する．

3.1 検索の機能分担と処理手順

本検索システムでは検索時間の大半を占める文字列検索の処理を LISCAR に行わせ，単語間の論理演算処理（AND，OR，NOT，記号のカッコなど）や距離値判定処理を PC，および制御スカラ演算器に，検索結果のディスプレイ表示や保存などを PC に分担させる構成にしている．この構成に基づき，フルテキスト検索の処理は以下の手順で行われる．① PC から検索ソフトウェアと文書データとを，それぞれ LISCAR のプログラムメモリと ADM とに転送・格納する．② PC のコンソールから各種の検索単語と論理演算式を入力して LISCAR 側に送る．③ LISCAR では検索単語と一致する文字列を検出し，PC 側へその存在位置を通知する．④ PC では論理条件や距離値を満足する文書を選択して提示する．

3.2 検索アルゴリズム

検索方式については，検索処理の基本となる完全一致検索を 3.3 節で述べ，部分一致検索を 3.4 節で述べ

る．本節では検索条件，文書データの格納形式とその問題点について述べる．

（1） PE，および ADM の使い方

LISCAR を用いた検索の処理方法ではビットパラレル型（BP 法）とビットシリアル型（BS 法）の2つを考えた．BP 法は PE を2次元（プレーン）配列として使用するもので，1文字（16ビット）に対して16個の PE を割り当て，1回の照合命令で16文字（256ビット）の照合を同時に実行する処理である（**図4**（a））．一方，BS 法は PE を1次元（ライン）配列として使用するもので，1文字に対して1個の PE を割り当て，PE の深さ方向に16ビットを使用して1文字を表現し，16回の照合命令で256文字の照合を同時に実行する処理である（図4（b））．BS 法と BP 法との比較では，BP 法は文献9）で示した実験結果，および本論文の BS 法の実験の結果から，①両処理法は PE での照合回数は同じであるが，BP 法は照合判定に処理速度の遅い PE 間演算処理が加わること（2.3節参照），16 文字ごとに初期設定（文書データの設定と検索単語や文書データの終了判定処理など）と一致単語判定を繰り返すため，処理制御のオーバヘッドが多くなり，処理速度が低下する．また，② BP 法は照合単位が 16 文字と短いために，文字列の折り返し数も多くなり処理量が増大する欠点がある．このことから，本論文では BS 法についてのみ論じる．

BS 法による文字列検索の処理は，あらかじめ LISCAR の ADM 上に文書データをロードしておき，そこから検索要求を受け付ける．ADM 内の文書データは，256文字（256ビット×深さ16ビット）単位で，文字列の層を形成するように格納しておく．なお，BS 法では文書データを256文字単位でシリアルデータに変換して格納しなければならないが，この処理時間は文書データのロードの際に，LISCAR が SCSI の転送速度を上まわる速度の 5 M バイト/秒でパラレル-シリアル変換を行えるので問題にはならない．

（2） 文書データの格納形式

文書データを ADM に格納する方法には，連続する文字列を，PE の深さ方向に記録する方法（垂直格納），PE の並びと同一方向に記録する方法（水平格納）がある．垂直格納の場合は文書データのロードを効率よく行うために，あらかじめホスト上でシリアルデータに変換する前処理が必要となり，その処理に時間がかかる．そのため，ここでは水平格納を採用する．

（3） 文書データの折り返し

BS 法の場合，ADM 内で水平格納された文書データは，256 文字単位で折り返しが生じる．文字列検出では，折り返しで境界に生じた検索単語を，どのようにして検出するかが問題となる．

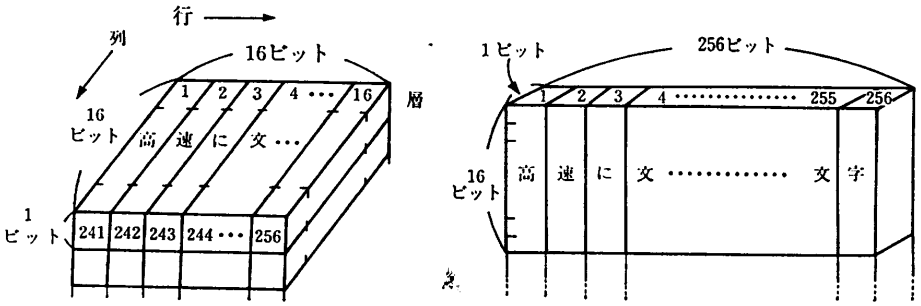
3.3 完全一致検索

BS 法を用いた完全一致検索の中から，全文字照合と絞り込み照合について検討した．全文字照合は文書データの文字と検索単語の文字とを総当たりで調べる方法であり，絞り込み照合は検索単語の先頭から一致をみていき，不一致が生じれば以後の文字照合は行わず，次の文書データとの照合に移る方法である．具体的な処理手順を以下に述べる．

3.3.1 全文字照合（BS 法 I）

（1） ライン（256）内の文字列検索

① 文書データのロード：ホスト上の文書データを LISCAR の ADM に転送して蓄積する．この時，文書データはシリアルデータに変換されており，ADM の各層（16ライン）に256文字が割り当てられ，層の連なりで文書データが表現される（**図5**の文書データ領域）．



（a） ビットパラレル型配置
（a） Bit parallel distribution.

（b） ビットシルアル型配置
（b） Bit serial distribution.

図4 ADM(RF) における文書データの配置（水平格納）
Fig. 4 Document distribution for ADM (Horizontal storage method).

② 検索単語の表現：N文字から成る検索単語をPCから受け取って文書データと同じようにシリアルデータに変換してADMのPEの深さ方向に配置する．これにより，ADM内の検索単語は，256 PEのすべてに対し，PEの深さ方向の$N\times16$ビットのメモリを使用して表現される（図5の検索単語領域，単語の例“文字列”）．

③ 文字照合：着目する層の文書データをRFにコピーし，RFの内容と検索単語との間で，256個のPEを用いて256個の文字（16ビット）を同時に照合させる（図5ではRFに文書データ領域の着目層をコピーして照合文書データとし，それとADM内の検索単語の“文”との間でビットごとの排他的論理和を採り，得られた16ビット内での論理和を採って文字ごとの照合結果の1ビットを得るようにしている）．

④ 一致文字の有無判定：全文字照合では文字ごとの一致判定は行わず，文字照合結果をADMに記録して（図5の照合結果），検索単語内の次の文字（図5の検索単語領域の“字”）の照合に移る．

⑤ 単語照合：検索単語のN文字の照合が終了した時点で，照合結果に対して照合順序に応じたシフト処理を施し，最終結果に一致情報が残っていれば（図5の一致検出テーブルの情報），一致情報から文字列の先頭位置を検出する．

⑥ 複数単語の照合：項番③，④，⑤の処理を検索単語数だけ繰り返し，層ごとにそれぞれの一致情報を検出・保存する．

⑦ 論理判定：指定された論理演算に従って一致情報を検査し，PC側で条件に合致した文字列の存在する文書データを抽出・表示する．

（2） ライン境界の文字列検索

⑤-（1）境界単語の判定：項番⑤の単語照合が終了した時点で，ライン末尾に検索単語候補の文字列が存在する場合は，不足した文字数sを検出する（図5の一致検出テーブルの一致候補）．

⑤-（2）境界単語の照合：ここでは256個のアレイを16×16の2次元配列とみなしてシフト処理を行い，境界文字列をライン内文字列に変えて照合を行う．すなわち，不足文字数sから$a=s/16+1$で表されるシフト量aを求め，aに相当する次層の先頭文書データの内容（$a\times16$文字$\times16$ビットの量）を着目する文書データの層にコピーする（図6のコピー文書データの

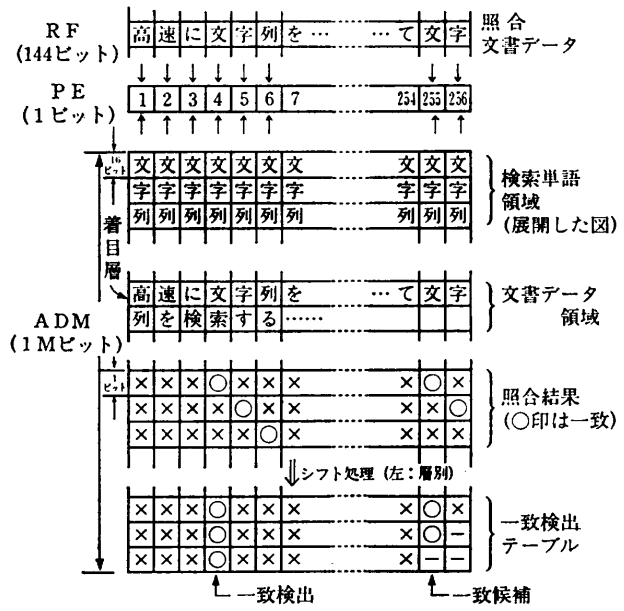


図5 文字列検索の様子
Fig. 5 Retrieval-word searching process.

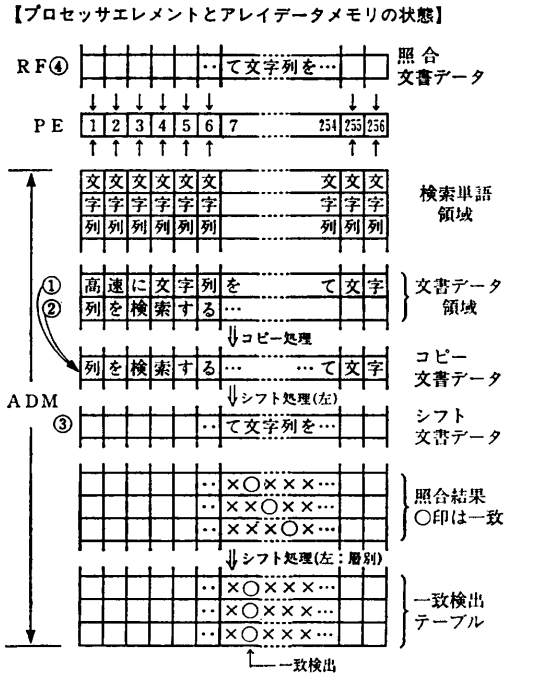


図6 折り返し境界の処理の様子
（○印内の番号は境界処理の順序を示す）
Fig. 6 Boundary-word searching process.

場合は $a=1$). 次に，アレイを2次元配列とみなして着目層を a 回だけ先頭文字列の方向（2次元配列の隣接 PE）にシフトさせ，分離されていた境界単語を連続単語にする（図6のシフト文書データ）．次に項番③，④，⑤のライン単位の処理を行い検索単語に一致するか否かを判定する．

3.3.2 絞り込み照合（BS 法 II）

3.3.1 項の全文字照合において，項番④の処理を次のように変更する．

④ 一致文字の有無判定：256 文字の照合結果に対し，AAP 2-LSI に用意されている全 PE の論理和演算を採る機能によって，高速に一致文字の有無を知ることができる．このとき，(イ)一致がなければ次の層（文書データ）の照合に移る．(ロ) 一致があれば一致情報を記録して，検索単語内の次の文字の照合に移る．以下，BS 法 I と同じ処理を繰り返す．

3.4 部分一致検索

文書データベースの検索で一般に用いられる部分一致検索の例として，異字許容照合と単語内ワイルドカード (don't care) 照合を採り上げ，これらの処理を LISCAR で行う際の実現法を以下に述べる．

（1） 異字許容照合

完全一致検索における全文字照合において，照合結果の一致判定の代わりに一致文字数を計数（$\log_2 N$ 語長の精度の加算を N 回繰り返す）し，計数結果と許容値とを比較して閾値より大きな値を示す文字列を検出する．この処理量 $N\times\log_2 N$ は全文字照合の約 12 %に相当し，処理速度は完全一致の全文字照合よりやや低下する．

（2） 単語内ワイルドカード照合

完全一致検索における絞り込み照合において，ワイルドカードに指定された文字部分の照合を省略してスキップすればよい．そのため，処理速度はワイルドカードの判定に要する時間とスキップによる効果とを差し引いたものになる．その結果，処理速度は検索単語長やワイルドカードに指定された文字数にも依存するが，4文字単語の中に1文字のワイルドカードがある場合をプログラムの処理量から換算すると約 5% の増加となり，完全一致型の絞り込み照合よりもわずかに遅くなる．

4. 評価（実験）

3.85 M バイトの新聞記事データ（約 10 日分）を実験対象とした．このデータを LISCAR の ADM に格納し，4文字から成る検索単語を用いて検索実験を行った．検索方法は，ビットシリアル型の処理方式の中から全文字照合（BP 法 I）と絞り込み照合（BP 法 II）とを選んだ．検索単語は新聞記事において出現頻度の高い 20 種（"経済協力" など）を選んだ[10]．ここで選んだ検索単語の語長と出現率との関係を調べると表3のようになる．これは 256 文字単位で集計したもので，絞り込み照合で検索単語の語長が増せば，文字列の連続する割合は減少し，照合の処理量も減少することを示している．

4.1 ライン内の文字列検索性能

完全一致検索における照合方式（全文字照合と絞り込み照合）と処理時間の関係を，実測値および測定に用いたプログラムの論理ステップ数から算出して比較したものを図7に示す．この図から以下のことがわかる．

（1） 全文字照合の処理時間は，検索単語の文字数に比例するが，1文字に対する初期設定と照合との占有時間比は 14 対7となり，検索単語数が増すと照合時間のみが増加する．

（2） 絞り込み照合の処理時間は，検索単語と一致する単語がすべての層で出現する場合の処理時間と，検索単語の1文字目の文字がすべての層に出現しない

表 3 検索単語の語長と出現率の関係
Table 3 Relation between character length of retrieval word and appearance rate.

状態 / 項目	検索単語の語長			
	1 文字	2文字まで	3文字まで	4文字まで
平均出現率* (%)	17.48	2.76	1.02	0.98

* 出現率の高い4文字単語を 256 単位で調査．

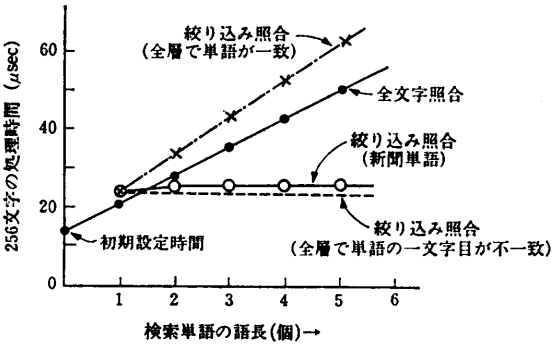


図 7 照合法と処理時間の関係
(完全一致検索の場合)
Fig. 7 Relation between retrieval time for the word length.

表 4 検索結果
Table 4 Retrieval result.

検索処理		性能	処理速度 (nsec/文字)
ビットシリアル形処理	完全一致	全文字照合	46
		絞り込み照合	23～25*
	部分一致	異字許容照合	55
		ワイルドカード照合	24～27*

* 出現率の高い検索単語の場合.

場合の処理時間との間に位置する．ただし，通常の場合は表3の新聞単語で示した値より小さな値を示す．このことから絞り込み照合の処理時間は，検索単語の語長にはほとんど依存せず，文書データ量に比例し，4文字単語の場合での1文字当たりの検索速度は 23～25 nsec 程度になる．

4.2 総合的な文字列検索性能

検索単語がラインの折り返し境界に存在した時の処理時間はライン内検索時間の約2倍かかる．全文字照合を基準に4文字単語を検索対象とした場合の検索結果の比較を表4に示す．このことから，LISCAR を用いたビットシリアル型のフルテキスト検索は，以下のような特長を有することがわかった．

（1） 完全一致検索における全文字照合の処理は，3.85 MB の全データと照合するのに 0.35 秒を要し，1文字当たり約 46 nsec（2.2 千万字/秒の速度）で処理でき，絞り込み照合検索では実効 25 nsec（約4千万字/秒の速度）程度で処理できる．

（2） 部分一致検索においても，異字許容照合については，完全一致検索の全文字照合よりわずかに増加し，55 nsec/字程度の速度で行うことができ，単語内ワイルドカード照合は絞り込み照合と同程度の速度 (27 nsec/字) の速度で行うことができる．

5. 考 察

本章では並列プロセッサによる文字列照合と，そこで用いたビットシリアル型の処理の長所・短所について考察する．

（1） 並列プロセッサは，プログラムの変更によって種々の検索アルゴリズムが比較的容易に搭載でき，ソフトウェアによる処理にもかかわらず高速な検索が実現できる．また，LISCAR による検索処理は，文書データを水平格納し，プロセッサアレイを文字列照合では1次元配列として使用，アレイ境界処理では2次元配列とみなしてシフト処理を行うことにより，効率のよい処理系を組むことができる．

（2） ビットシリアル型の処理（BS 法）における1文字の照合時間は図7の全文字照合を表す直線の勾配から，約 28 nsec と高速なことがわかる．しかし，初期設定と一致単語判定に約 56 nsec と多くの時間を要している．このことから，LISCAR は繰り返し演算を高速なビットライン間の算術論理演算で行うような処理に適し，ビットパラレル型の処理のように，16文字ごとの単語照合や状態設定，および PE 間のデータ転送を基本にしたものは不向きなことがわかった．

（3） PC と LISCAR とを接続したフルテキスト検索システムは，文書データを LISCAR 内にロードしておけば，高速な検索ができる．しかし，最初の文書データのロードに時間が掛かるため，データベースへの追加・変更の少ない検索に適している．

6. おわりに

SIMD 型の小型並列プロセッサ（LISCAR）を PC のバックエンドプロセッサとして用いた検索システムを試作し，フルテキスト検索の実験を行った．その結果，①本システムは完全一致検索では，4文字単語の場合，全文字照合の検索で 2.2 千万字/秒，絞り込み照合で実効4千万字/秒程度の高速な検索ができる，②部分一致検索でも完全一致型の検索とほぼ同程度の検索速度が得られる，③さらに高速化を図る場合や文書データの大容量化に対してはボードやユニットの増設で対応できる，などがわかった．このシステムは検索時間が短く小型で経済的なので，文書データの調査や分析などのようにいろいろな視点から検索を行う業務に威力を発揮するものと考える．今後は，LISCAR への文書データの高速な入力法，システムへのシソーラス[2]の搭載や言語解析への適用などについて検討していく予定である．なお，このシステムは既開発の文書認識ソフトウェアを搭載すれば，文書認識装置としても利用できる[11]．したがって，本検索機能が加わったことにより，LISCAR は文書の認識入力と検索の両方が実現でき，単独でコードベースの文書ファイリングシステムを構築できる．

謝辞 本研究の機会を与えていただいた当研究所遠藤部長，ご意見をいただいた小橋主幹員，中川主幹員を始め協力いただいた関係各位に深謝します．

参考文献

1) 根岸正光：フルテキスト・データベースの実用化における諸問題，情報処理学会情報学基礎研究会，14-1 (1989).

2) 宮原末治，鈴木　章，多田俊吉，壁谷喜義：文書情報の蓄積検索システムに関する検討，情報処理学会 HI 研究会，29-3 (1990).
3) Stanfill, C. and Kahle, B.: Parallel Free-text Search on the Connection Machine System, *Comm. ACM*, Vol. 29, No. 12, pp. 1229-1239 (1986).
4) 加藤寛治，藤沢浩道，大山光男，川口久光，畠山　敦：大規模文書情報システム用テキストサーチマシンの研究，情報処理学会情報学基礎研究会，14-6 (1989).
5) 安藤敦史，菅野祐司，伊東正雄，田村　登，鶴林　健，早川佳宏：フルテキスト・データベース・システム検蔵君，情報処理学会 AD シンポジウム，pp. 17-25 (1990).
6) 山田八郎，高橋恒介，平田雅規，永井　肇：あいまい検索が可能な文字列検索 LSI，日経エレクトロニクス，No. 422, pp. 165-181 (1987).
7) 多田俊吉，近藤利夫，宮原末治：小形高並列プロセッサとその文字認識への応用，信学論 (D), Vol. J 71-D, No. 8, pp. 1546-1552 (1988).
8) 近藤利夫，北村美宏，浜口重建，土屋敏雄：高並列処理用セルラアレイプロセッサ LSI-AAP 2，信学技報，SDM 87-104, pp. 25-30 (1987).
9) 宮原末治，近藤利夫，多田俊吉：SIMD 形並列プロセッサを用いたフルテキスト検索システム，並列シンポジュウム JSPP '91, pp. 61-68 (1991).
10) 国立国語研究所：電子計算機による新聞の語彙調査 (IV)，国立国語研究所 (1973).
11) 宮原末治，鈴木　章，近藤利夫，多田俊吉：LIS. CAR を用いた印刷文書読み取り，*NTT-R&D*, Vol. 39, No. 11, pp. 1593-1602 (1990).

(平成 3 年 8 月 7 日受付)
(平成 3 年 11 月 5 日採録)

宮原　末治 (正会員)

1946 年生. 1969 年熊本大学工学部電気工学科卒業. 1971 年同大学院工学研究科修士課程修了. 同年日本電信電話公社研究所入社. 以来，音声情報処理の研究，文字認識装置の研究実用化，ペン入力型文字認識の研究，文書検索・言語処理の研究実用化に従事. 現在，日本電信電話株式会社ヒューマンインタフェース研究所勤務. 電子情報通信学会会員.

近藤　利夫

1953 年生. 1976 年名古屋大学工学部電気工学科卒業. 1978 年同大学院工学研究科修士課程修了. 同年日本電信電話公社入社. 以来，プロセッサアレー LSI，これを用いた大規模並列プロセッサ，文字認識処理装置への応用等の研究に従事. 現在，NTT ヒューマンインタフェース研究所. 電子情報通信学会，IEEE 各会員.

多田　俊吉

1948 年生. 1971 年慶応義塾大学工学部電気工学科卒業. 1974 年同大学院工学研究科修士課程修了. 同年日本電信電話公社研究所入社. 機構部品の研究，文字認識装置の実用化に従事. 平成 2 年より NTT インテリジェントテクノロジー(株)へ出向. 小型並列プロセッサを用いた OCR 製品開発に従事. 電子情報通信学会会員.