

記号処理カーネルSILENTのハードウェア構成

吉田雅治*, 天海良治**, 山崎憲一**,
中村昌志**, 竹内郁雄***, 村上健一郎***

- * NTT ヒューマンインタフェース研究所
- ** NTT 基礎研究所
- *** NTT ソフトウェア研究所

記号処理システムの核となるプロセッサSILENTのアーキテクチャ, 実験機構成, 及び簡単な性能評価について述べる. SILENTは記号処理のみならず, 知能ロボットやコンピュータグラフィックス等の分野への適応も考慮し, 画像生成計算機SIGHT-2で提案したTARAI演算器と密結合マルチプロセッサを構成することを前提としている.

SILENT-CPUは0.7 μ CMOSゲートアレーで作成した. LSIの回路規模は97k gates, 33MHzで動作している. SILENTアーキテクチャの性能評価のために実験機を作成し, マイクロプログラムでLispの処理系を実装した. 関数呼び出しのオーバーヘッドを計測するtarai関数の実行時間を比較したところ, SILENTはELISの10倍以上の性能を得た.

A List Processor SILENT

Masaharu Yoshida*, Yoshiji Amagai**, Kenichi Yamazaki**,
Masashi Nakamura**, Ikuo Takeuchi***, Kenichiro Murakami***

- * NTT Human Interface Labs.
- ** NTT Basic Research Labs.
- *** NTT Software Labs.

SILENT is a Lisp processor. The main target of SILENT is not only artificial intelligence research but also computer graphics and autonomous robot control. To improve the performance of numerical data processing, SILENT will be tightly coupled with the TARAI unit which was proposed as a part of SIGHT-2. The SILENT CPU LSI employs 0.7 μ CMOS gate array technology. The circuit has 97 kilo gates, operating at 33MHz. A SILENT prototype was build to evaluate the system performance. The list processing system is programmed by micro program. The performance of SILENT is more than ten times faster than ELIS.

1. はじめに

計算機の応用分野の拡大に伴って、記号処理はますます広い分野で使用されつつある。例えば、コンピュータグラフィックス^[1]、工業用ロボットの行動計画の策定^[2]、あるいはTCP/IPネットワークシステムの制御^[3]等がある。記号処理言語を実行するには専用機による方法と汎用マシン上での言語処理系によるものがある。記号処理専用機として80年代中頃には各種のLispマシンが開発されてきたが今では見るべきものはない。現在ではRISC型ワークステーションにLispの処理系を搭載したシステムが使用されているが、実世界の問題に適用するには次のような欠点がある。

- (1) 処理系自身がOS上の一アプリケーションに過ぎないため、OSの持つ実時間スケジューリング機能、プロセス間通信機能などが十分に生かせず、実時間処理の応用に適さない。
- (2) デバイス制御など低レベルな部分まではLispで記述できないため、均一な環境下での開発ができない。
- (3) デバッグのためのインタプリタ実行と実処理でのコンパイル実行の速度の差が大き過ぎるため、実時間制御等をするプログラムのデバッグが困難である。

これらの問題を解決するため、新しい記号処理カーネルを提案する。新しい記号処理カーネルに求められるものは高い数値演算能力や他の計算機システムとの親和性である。例えばロボットコントローラに使用できるためには、実時間でロボットの軌道生成処理を行える数値計算能力だけでなく、サーボモータコントローラやセンサインタフェースプロセッサとヘテロなマルチプロセッサを構成できることが求められる。

そこで我々はこれまでの経験、コンピュータグラフィックス専用計算機SIGHT-2の数値計算用プロセッサのTARAIユニット^[4]、記号処理計算機ELISでの記号処理専用アーキテクチャ^[5]、その基本言語であるTAO^[6]等を基に新しい記号処理カーネルであるSILENTのアーキテクチャを検討してきた^[7]。TAOは手続き型プログラム(Lisp)、論理型プログラム、オブジェクト指向プログラム

の3種類のプログラミングパラダイムを統合した言語である。TAOはSILENT上での実装を機会に再設計され、より高性能でエレガントな言語へとリファインされた。

SILENTはELISの流れをくみ、TAOを基本言語とする専用機であるが、

- (1) 他のプロセッサとの親和性を高めるため、32ビットデータ/アドレスを基本としたバイトアドレッシングを採用。
 - (2) コンピュータグラフィックス、知能ロボット等の分野での大量の数値計算を処理するため、TARAIユニットを接続。
 - (3) 機械語としていわゆるバイトコード(実際は10ビット長)を導入し、マイクロプログラムにて解釈実行。
 - (4) 水平型マイクロプログラム制御による並列処理の徹底。
 - (5) TAOに適合した専用ハードウェア。
- 以上の点を主たる特徴とする。ELISとの互換性は考慮していない。

SILENTは現在、プロトタイプシステムのハードウェアのデバッグを終了し、マイクロプログラムの実装を進めている。本稿ではSILENTのアーキテクチャ、プロトタイプシステムの概要、及び簡単な性能評価を示す。

2. SILENTアーキテクチャ

Lisp, 論理型プログラミング, オブジェクト指向プログラミングの中心となるのは記号処理である。Fortranに代表されるような数値演算処理と比較して記号処理の特徴には次のようなものがある。

- (1) メモリのアクセスはポインタベースであり、アクセスのローカリティは少ない。
- (2) しばしばプログラムとデータが等価に扱われる。このため、インタプリタ形式の処理系が必要になる。
- (3) オブジェクト指向プログラミングの長所を生かすためには、実行時のデータタイプのチェックやデータのメモリ割りつけが必要になる。
- (4) メモリの使用状態を予測できないため、ガー

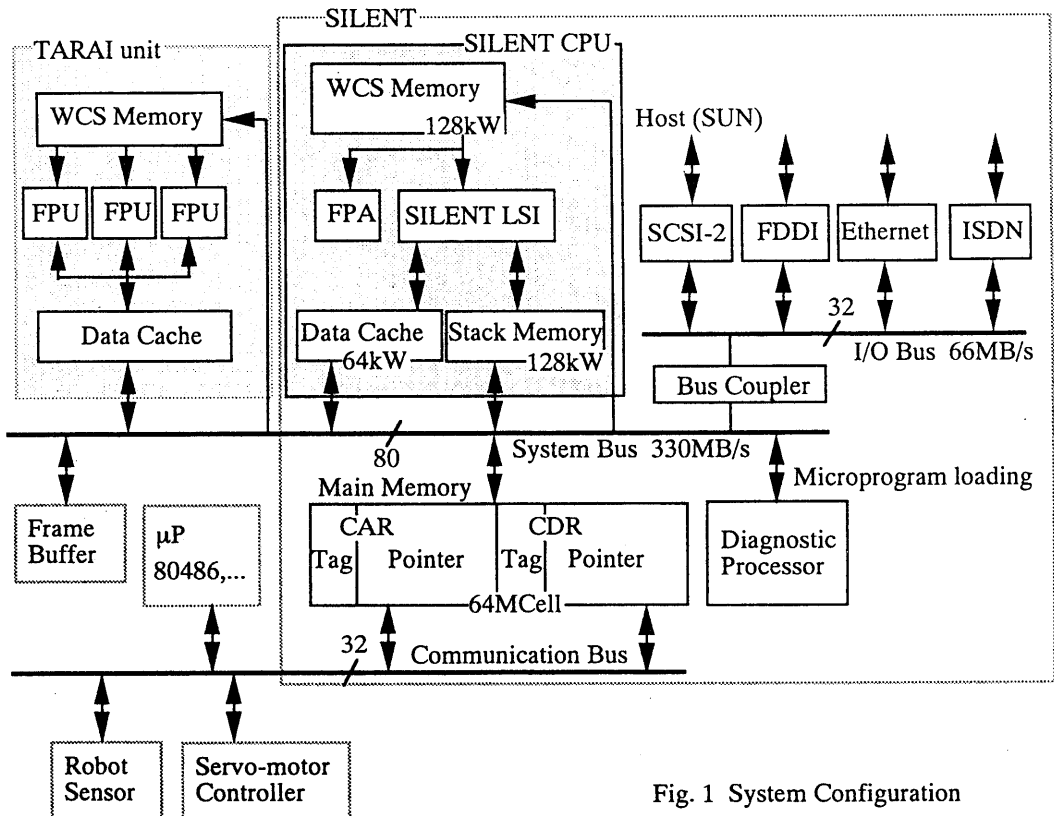


Fig. 1 System Configuration

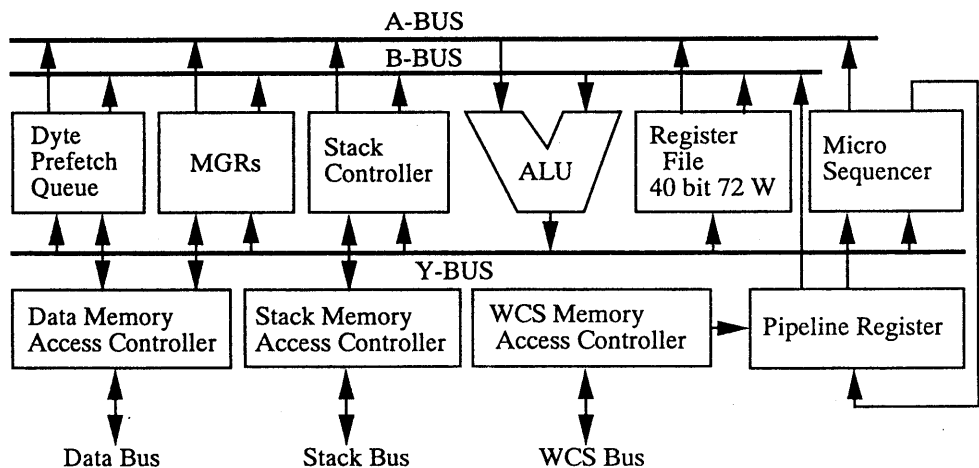


Fig. 2 Data Path Structure

ページコレクションが必要になる。

以上の特徴から記号処理専用ハードウェアとしては次のようなことが求められる。

- ① ポインタの処理の高速化。
- ② スタックの強化。
- ③ タグの処理の高速化。
- ④ パイプライン・レーテンシの低減。

④については実行時にタグを始めとして各種のチェックを行うため、条件分岐命令の発生頻度が高く、パイプライン・レーテンシが長いとパイプライン・ハザードのペナルティが大きく、実質的な性能が得られないためである。

このような特徴は数値演算処理とは異なった面を持っているので、知能ロボットやコンピュータグラフィックスのように記号処理と数値演算処理の双方に高い処理能力を求めるアプリケーションに対しては、記号処理に特化したプロセッサと数値演算処理に特化したプロセッサのマルチプロセッサ形式を採るのが自然であると考えられる。

図1にSILENTプロトタイプ機のシステム構成を示す。TARAIユニットは数値演算処理に特化したプロセッサ、フレームバッファはコンピュータグラフィックスに不可欠のものであり、実装を予定している。80486等の他のプロセッサとの接続や、ロボット関連のプロセッサとの接続も予定している。

SILENTはバックエンドプロセッサに位置付けられ、ファイルシステムやユーザインタフェースはホストプロセッサが担当する。ホストとの接続にはSCSI-2を使用するため、市販のほとんどのワークステーション等と接続できる。SILENTは、FDDI、イーサネット、ISDNの各ネットワークをサポートする。新しいインテリジェント・ネットワーク・プロトコルの開発を予定している。

診断プロセッサは486にMSDOS及びLispを実装したプロセッサで、ハードウェアの診断、メモリの初期化、ブート用マイクロプログラムや後述するByteコードのロード等を担当している。ブート後はSILENTが自身で管理し、診断プロセッサはプログラムのデバッグ等に使用している。

2.1 SILENTデータバス系

SILENT CPUのコアであるSILENT LSIのデータバス系を図2に示す。LSIのゲート数の制限から浮動小数点演算器(FPA)は外付けとなった。システムクロックを低く設定して消費電力を抑え、パイプライン段数を極力減らすため、SILENTでは多くのユニットを並列動作させている。図2のうち、レジスタファイルを除く各ユニットは並列動作する。また、プログラマに見えるパイプライン段数は1段である。

SILENTのALU部には、図3に示すように2個のALU(8ビットTag ALU, 32ビットData ALU), 32ビット乗算器, 64ビットバレルシフト, ファインドビットユニット(FBIT), 及びマッシュユニット(MASH)がある。Tag ALU, 乗算器は他のユニットと並列動作する。Tag ALUはタグに対して論理演算をサポートし、Data ALUはデータに関して整数演算及び論理演算をサポートする。Data ALUの語長は命令毎に、32, 16, 8の3種を選択することが可能で、文字列処理やデバイス制御のプログラムで効果的である。記号処理プログラムではほとんどのデータ演算でタグを操作するのでTag ALUを実装したことにより命令内並列度が上がった。SILENTはナンバクランチャであるTARAIユニットと接続することを前提としているが、マルチプロセッサ構成でのオーバーヘッドを勘案し、少量の浮動小数点演算については自前で処理することができる。

レジスタファイルは3アドレス(2R/1W)形式のもので、ライトバッファを有し、読出しと書き込みを同時に行うことができる。書き込みのために余分なパイプライン段数が必要にならぬように、読出しアドレスと書き込みアドレスの間にはなんら制約がないように回路を構成した。この結果、SILENT LSIのタイミングのクリティカルパスはレジスタファイルの読出し時間+ALUの処理時間となった。72個のレジスタのうち、8個は浮動小数点演算器とのインタフェース用で、64個が汎用レジスタである。また、このうちの32個はレジスタ間接アドレッシングが可能である。

メモリ汎用レジスタ(MGR)はELISで導入されたもので、car用に4個、cdr用に4個の計8個の

レジスタがある。ロード/ストアバッファとして機能するデータバスアクセス・レジスタと、文字列(バイト列)操作用レジスタの2種類の機能を持つ。後者は8個のレジスタをリングバッファとして扱い、任意の位置から文字列(バイト列)を切り出すことができるので文字列操作の高速化に貢献している。

スタックコントローラは4組のスタックレジスタ及びスタックトップレジスタから構成される。スタックトップレジスタは文字通りスタックトップの値を常時保持しているレジスタで、スタック操作の高速化に寄与している。更にスタックベースレジスタや、スタックのバウンダリをチェックするハードウェアを実装し、スタックが論理アドレスでアクセスできるようにした。記号処理プログラムではスタック操作が多用されるためスタック操作が効率良く並列に実行できることが重要である。この結果、スタック管理が容易になり、プロセススイッチ時間も短縮できた。

図には示していないが、スタックコントローラとMGRの間には専用のデータ転送パスを設けて、ALU部を介さずにスタックとメモリ間のデータ交

換を可能にしている。これはスタックマシンにおけるデータ処理のスループットを上げることに寄与している。

SILENTはマイクロプログラムで制御される。マイクロプログラムフィールド構成を図4に示す。Tag ALU, Data ALU, MGR等は、独立したマイクロフィールドを有し、並列動作することができる。ALUは3アドレス方式を採っている。更に、Dyteと呼ぶ10ビット長の機械語を設けた。通常1~3個のDyteコードで1機械語命令を構成する。Dyteコードで直接汎用レジスタを指定することも可能であり、Dyteコードを解釈実行するマイクロプログラムの効率を上げている。更にシステム性能を向上させるために、Dyte命令ポインタ、Dyteプリフェッチキューやそのための独立したシーケンサ等を装備している。Dyte命令ポインタの更新は、Dyteコードの参照に伴って自動的に行ない、Dyteのプリフェッチもハードウェアが自動的に行うため、処理の高速化が可能になった。

SILENTの基本データ形式は図5に示すような、8ビットのタグ、32ビットのデータからなるもので、Lispのcarやcdrに対応している。SILENTは

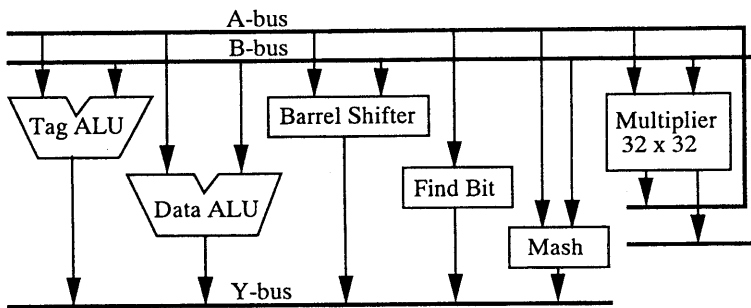


Fig. 3 ALU

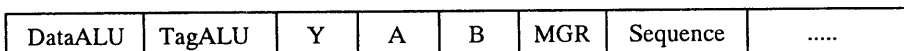


Fig. 4 SILENT Microinstruction Format

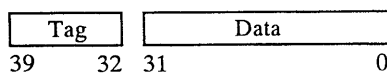


Fig. 5 The Basic Data Structure

80ビットデータバスを使用しているので、carとcdrからなる1セルを同時にアクセスすることができる。1セルは80ビット長であるが、この大きさは、① 8個のDyteコードを収容できる、② 80ビットの拡張浮動小数点形式のデータを格納できる、の2点でも有用である。後者はロボット制御で使用される初等関数の演算精度を満たすために必要である。

SILENTはバイトアドレッシングを採用しているが、タグ部分にはアドレスはない。データ部分のアドレスから一意にタグ部分が特定できるからである。また、こうすることでデータ部分のアドレスが連続することになるため、通常のプロセッサと接続した時にデータとアドレスの対応付けが共通化できる。更に、アドレッシングとしては、1セル = 8バイト = 8 Dyte となるので、Dyteコードのアドレス管理も容易になった。

2.2 SILENTの専用ハードウェア

以下の節ではSILENTの特徴的な部分について説明する。

FBITは入力オペランドの"1"の立っているビットの位置をMSB側から、またはLSB側から検索してそのビット番号を返すユニットで、プロセス・プライオリティ・キュー、メモリ管理、デバイスドライバ、ロボット制御、グラフィックスデータ等の処理で有効である。

MASHは入力オペランドの簡易なハッシュ値を返すユニットで、ハッシュテーブルの管理ルーチンで使用される。その一部を以下に示す。

```
Y15=(A10$A20$A21$A31)&B15
```

```
Y14=(A0 $A1 $A11$A22)&B14
```

.....

ここで、\$は排他的論理和、&は論理積、A、B、YはそれぞれA入力、B入力、Y出力、数字はビット位置を示している。このビットパターンは実験によって求めたもので、4種類のパターンを実装し、再ハッシュの時に切り換えている。MASHのハードウェア量は僅かなものであるが、記号処理のプログラムではシンボル管理の処理の占める割合が高いため、メソッド探索、名前表管理、変数

管理等の処理の高速化に効果的である。コストパフォーマンスの高いハードウェアである。

記号処理プログラムをサポートするレジスタにRPR(Replacement Register)とRPF(Replacement Flag)がある。最後に参照した変数やデータフィールドの値や型を保持するレジスタで、代入やユニフィケーションの高速化と高機能化に効果がある。両レジスタに対する値の書き込みは参照と同時にわれ、このための遅延は発生しない。

タグ、データ、アドレスフィールドを参照する多数の条件分岐命令や多重分岐命令もマイクロプログラムのダイナミックステップ数の減少に効果的である。例えばデータの型の判別のためのタグフィールドの判定も1命令で処理できるため、通常条件分岐命令をネストするよりもステップ数を減少できる。現行のSILENTはソフトウェアから見たパイプライン段数が1段しかないため、あまり意味はないが、次のバージョンで段数が増えた時には分岐命令の頻度を低下させたことでパイプラインハザードの低減が期待される。

ロボットの制御やコンピュータグラフィックス等では多数の割り込み要因に対応する処理能力や実時間性が求められる。このため、通常マイクロプログラム制御で用いられるソフトウェアポーリングの他にハードウェアによるマイクロプログラムへの1レベルの割り込み制御をサポートし、処理の効率化を図った。

3. SILENTプロトタイプシステム

SILENT LSIは3層メタル配線の0.7 μ CMOSチャンネルレス・ゲートアレーで作成した。使用したプロセスはLCA210K⁽⁸⁾、回路規模は97kgateである。15mm角のチップを391pinのPGAに実装した。33MHzの2相クロックを使用している。LSIメーカーの提供するライブラリのメモリセルの性能が不足していたため、レジスタファイルは全てロジックで組んだ。この部分だけで約27kgateである。また図2ではデータバスをバス構成で示しているが、実際には全てマルチプレクサ構成を使用しており、いわゆるバスはLSI内部には存在しない。バス構成による性能の低下を嫌ったためである。

但しこのことはLSIのレイアウト上のネックとなり、2層では配線し切れず、3層となってしまった。

SILENT CPUとメインメモリを搭載したプリント基板の写真を図6に示す。プリント基板の寸法は400x320mm、基板面積の大半を占めているのはメモリで、主として32MBのDRAMモジュールと1MbのSRAMで構成している。IC類は片面実装で、反対側にはCR類のみ表面実装している。

メモリキャッシュは8Kエントリ、1エントリは8セルである。TAOのデータ構造とキャッシュ境界をあわせるようにデータ構造が設計されており、キャッシュのヒット率を上げている。

SCSI-II、FDDI等の各インタフェースは別基板に搭載した。

4. 性能評価

SILENTの基本言語であるTAOの現在の開発状況は、SILENTの機械語であるDyteコードのインタプリタのマイクロプログラムの作成を終了し、ユー

ザ記述から機械語へのコンパイラ/インタプリタとマイクロOSカーネルを現在作成中である。以下に簡単なベンチマークテストの実行結果を記す。使用したテストプログラムは、関数呼び出しのオーバーヘッドを計測するtarai関数(名前は似ているがTARAIユニットとは無関係)とフィボナッチ関数である。プログラムは付録に掲載した。SILENT、ELISと、Franz社のAllegro Common LispをHP社のHP9000/755/125上で実行した3種の比較を表1にしめす。

tarai関数の実行時間の比較でELISの約12倍、HPの約半分の性能となった。クロック周波数の比較ではELISの約6倍、HPの約1/4であるから、このような単純なベンチマークに関しては専用アーキテクチャの効果は約2倍といえる？。

表1. 性能評価

System	Clock	tarai 12 6 0	fib 30
SILENT	33MHz	11.155秒	2.746秒
ELIS	5.6MHz	134.037秒	28.595秒
HP9000	125MHz	5.27 秒	1.37 秒

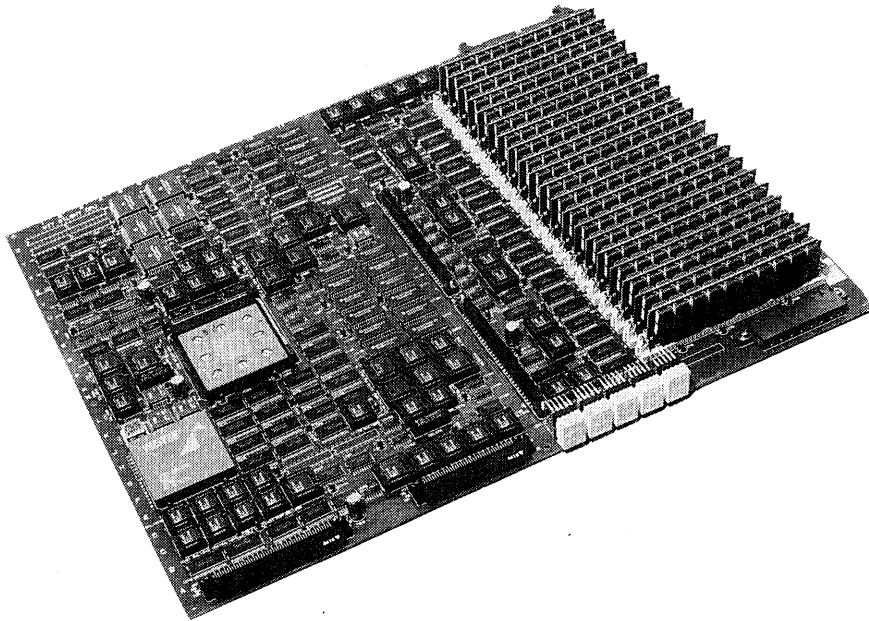


Fig. 6 SILENT CPU Board

現在作成中のマイクロOSカーネルではプロセススイッチが典型例で約3 μ 秒、プロセス生成からトリビアルなタスク実行を終えて資源を開放するまでの時間が約17 μ 秒と、マイクロプログラムで書いただけの性能がでており、実時間処理への適応が期待できる。

フィボナッチ関数の再帰をプロセス生成にした多重プロセス版フィボナッチ関数のベンチマークではELISの約300倍以上の性能を得ている。スタック関連のハードウェアや、各ユニットの並列処理が予想以上に効果的であった。

SILENTにはこうした性能評価用のハードウェア(主としてカウンタ類)を幾つか実装してあるので、今後、アーキテクチャの効果について詳細な検討を進めたい。

5. 終わりに

コンピュータグラフィックス、知能ロボット、知的ネットワーク制御等の分野に多様なプログラミングパラダイムを提供するためのベースとなる記号処理核であるSILENTのアーキテクチャを提案し、プロトタイプ機を作成した。現在、SILENTの基本言語であるTAOの実装を進めている。

SILENTはELISの流れをくむ記号処理カーネルであり、

- (1) 32ビットデータを基本としたバイトアドレッシングを採用。
- (2) TARAIユニットとマルチプロセッサ構成。
- (3) 機械語としてDyteコードを導入。
- (4) 水平型マイクロプログラム制御による並列処理の徹底。
- (5) TAOの処理系に適した専用ハードウェア。等の点を特徴としている。

CPUは0.7 μ CMOSゲートアレーで作成した。約97kgateの回路規模となった。プリント基板に実装し、動作を確認した。簡単なベンチマークテストではELISの約12倍、簡単な多重プロセスの実験では300倍強の性能を得た。今後詳細な性能評価を進める予定である。

参考文献

- (1) Reynolds W. "Flocks, Herds, and Schools: A Distributed Behavioral Model", ACM SIGGRA H '87. Conf. Proc. Vol.21, No.4, p.25-34.
- (2) 白井, 辻井「人工知能」岩波講座 情報科学 22
- (3) K. Murakami, "An Object-Oriented Implementation of TCP/IP Network Protocols", in Advances in Software Science and Technology, Vol.3, p.37-52, 1991
- (4) M. Yoshida, T. Naruse, T. Takahashi, "A Dedicated Graphics Processor SIGHT-2", in Advances in Computer Graphics Hardware-IV R. L. Grimsdale and W. Strasser Eds., Springer-Verlag, p.151-169, 1991.
- (5) 渡辺, 石川, 山田, 日比野「32ビットAIチップELISのアーキテクチャ」信学会計算機アーキテクチャ 69-10, 1988
- (6) I. Takeuchi, H. Okuno, N. Osato, "A List Processing Language TAO with Multiple Programming Paradigms", New Generation Computing, Vol.4, No.4, Ohmsha & Springer 1986.
- (7) 吉田, 竹内, 山崎, 天海, 「新しい記号処理カーネルSILENTの設計」情処学会記号処理 56-1, 1990
- (8) 0.7-Micron Array-Based Products Databook LCA200K/LEA200K Compacted Array Turbo Series, April 1992, LSI Logic Corp.

付録 ベンチマークテストプログラム

- (1) tarai


```
(defun tarai (x y z)
  (if (<= x y) y
      (tarai (tarai (1- x) y z)
              (tarai (1- y) z x)
              (tarai (1- z) x y))))
```
- (2) fib


```
(defun fib (n)
  (if (<= n 1) 1
      (+ (fib (1- n)) (fib (- n 2)))))
```