

知識情報処理用プロセッサAI32

Symbolic Data Processor AI32

知識情報処理言語(AI言語)の処理システムに対して、低価格化の要求が増大している。日立製作所は、このための処理回路とマイクロプログラム記憶をワンチップ化したAI言語用プロセッサAI32^{※1)}のアーキテクチャ確立とその性能評価を行った。

マイクロプログラム記憶にはEPROM(データを消去することができる書き込み可能な読み出し専用メモリ)を使用しており、目的とするAI言語に応じてプログラムすることができる。

AI32を用いることによって、低価格でより高性能・高度なAI処理システムが実現できるため、高機能ワークステーションだけでなくパーソナルコンピュータへの応用も期待できる。

河崎俊平* *Shumpei Kawasaki*

野尻 徹** *Tôru Nojiri*

迫田行介** *Kôsuke Sakoda*

天野亘孝** *Nobutaka Amano*

1 緒 言

従来から知識情報処理機能を持つAI(Artificial Intelligence)言語の有用性は、コンピュータ科学の研究者の間では広く認められていた。ところが、近年プログラム開発効率などの点に着目して、産業界でもAI言語の実用価値が認められ、AI言語処理能力を持つワークステーションが販売され始めた。AIワークステーションの普及に伴い、より安価で高性能なAI言語処理システムが求められてきている。

従来のAI言語処理システムは、ビットスライス形のはん(汎)用LSI、高速RAM(Random Access Memory)及びバイポーラ形論理LSIなどから構成されていた。これらのシステムでは、特定のAI言語を対象とした処理を行うマイクロコードを持ち、その言語に特有の処理を効率よく実行できる。対象とするAI言語を変更する場合は、その言語を処理するマイクロコードを高速RAMで構成されるマイクロ記憶に再実装すればよい。しかし、この種のシステムは構成チップ数が多く、コスト高となるため、適用できる応用分野が限られていた。

AI32ではAI言語を高速に実行するハードウェアとEPROM(Erasable Programmable Read Only Memory)で構成されるマイクロプログラムメモリを単一チップ上に集積している。ユーザーは対象とするAI言語に応じてマイクロプログラムをAI32内のEPROMに格納する。従来、バイポーラ個別部品でAI言語専用プロセッサを供給していたメーカーは、AI32上にマイクロコードを開発することによって、従来のコンパイラ、デバッガを含めアプリケーションソフトの変更なしにシステ

ムの低価格化や高性能化も可能となる。

また、ユーザーがマイクロプログラミングを行う場合に必要サポートツールとして、マイクロアセンブラなどのソフトウェアと評価ボードなどのハードウェアをキットとして供給する予定である。このため、ユーザーでのシステム開発費の低減も可能となる。

本稿では、AI32の特徴、AI32のアーキテクチャ、システム開発手順とサポートツール及び応用方法について論じる。

2 AI32の特徴

AI32は図1に示すとおり、ホストMPU(Microprocessor Unit)から起動されて、AI言語を高速に実行する周辺チップである。日立製作所の高性能32ビットMPUであるH32ファミリーのLSIであり、H32との接続に適しているが、H32以外のMPUとも接続が可能である。このためAI32は高性能システムだけでなく、メモリとのバスバンド幅が比較的狭いパーソナルコンピュータの上でも高速にAI言語の処理が可能である。以下にAI32の特徴について述べる。

(1) AI言語の高速処理

AI言語の中間言語を、少ないマシンサイクル数でエミュレートし高速動作を実現するために、水平形のマイクロ命令フォーマットを用いて、六つの独立したブロックを並列に制御する。更に、コンテキストやデータのキャッシング方法を含めた内部アーキテクチャの工夫によって、同じテクノロジーで作成されたはん用32ビットマシンよりも4倍近い性能向上を達成することを目的としている。高速化を図るためのアーキテクチャ上の特徴とその効果を図2に示す。

(2) 多種類のAI言語への対応

※1) AI32は、元東京大学助教授鈴木則久氏の設計した「刀32」の概念をもとに、日立製作所が開発したものである¹⁾。

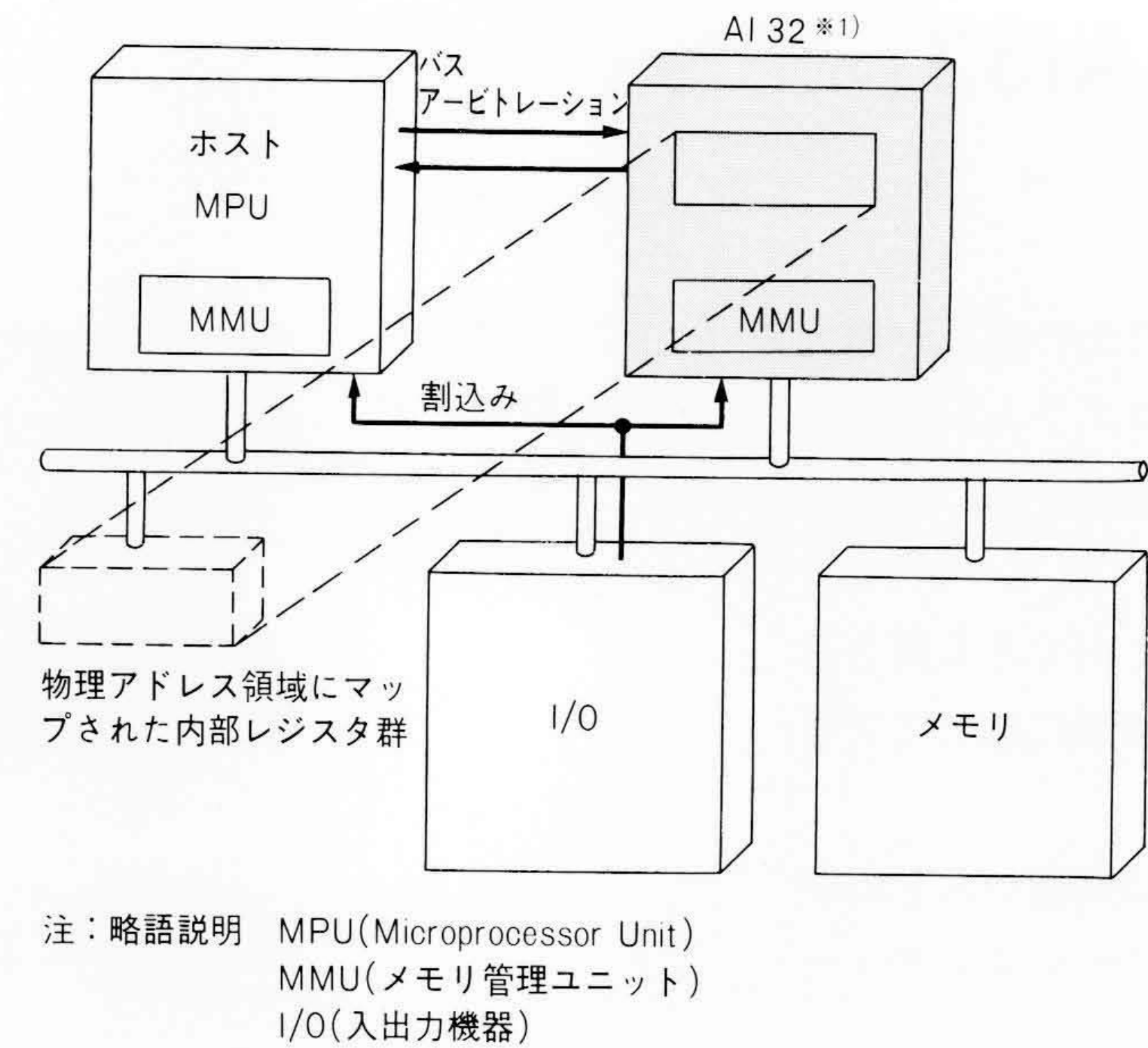


図1 システム構成図 AI32は自らアドレスを出力してメモリをアクセスするが、MPUからの位置づけは周辺チップとなるため、既存システムへの組込みは容易である。

AI32は、使用される高級言語がバイトフォーマットを持つ命令セットにコンパイルされるものであれば、内蔵マイクロ記憶容量の許す範囲で、どんなものもデコードし実行できる。また、通信分野やグラフィック用途の演算処理又はエキスパートシステム構築用ツールなどのように、定型的なオペレーションが頻繁に出現する分野に対してもAI32は有効である。これらの定型的なオペレーションをマイクロ化することによって高速化が可能だからである。表1にAI32を使用することによって、性能向上が見込める言語の例を掲げる。

(3) 既存のシステムへの組込みが容易

ユーザーは既存のシステムにAI32を追加して容易に高性能化を実現できる。MPUがメモリマップされたAI32の内部レジスタに制御情報を格納するだけで、後はAI32がMPUからバス権を獲得して、AI言語の中間言語をフェッチし、メモリとのオペランド転送を行いながらAI言語処理を行う。このため、MPUプログラムの追加・変更は少なく済む。

(4) フォン・ノイマン・ボトルネックの回避

現在のMOS・LSI(Metal Oxide Semiconductor・LSI)では、プロセスの改良によってチップ内部信号の遅延時間が年々減少する。しかし、チップ間、特にシステムバスでの遅延時間はほとんど改善が見られない。特に、高性能はん用マイクロプロセッサでは命令やデータをメモリからフェッチするために多くの時間を費やし、性能の多くの部分はメモリのバンド幅で制限される。従来、AI32では外部メモリに存在していた命令やデータを、内部のマイクロプログラムやレジスタファイルに取り込んでいる。このため、外部メモリのアクセス頻度が大幅に減少しており、メモリの遅延時間による性能への影響は少ない。つまり、安価な低速メモリを用いても高性能システムが構築できる。また、将来AI32のクロック周波数を上げた場合、ほぼ周波数に比例して性能を上げることがで

表1 AI32を使用して性能向上が見込める言語 AI32を使用して、高級言語の中間言語を直接実行した場合、下記の各種言語について高速化が行える。

AI32を使用して性能向上が見込める言語
Smalltalk-80 ^{※2)}
LISP
Prolog
FORTH
PostScript ^{※3)}
APL
Pascal
modula-II
Ada ^{※4)}

きる。つまり、将来の半導体技術の進歩に追従して容易に性能向上を図ることができる。Smalltalk-80^{※2)}のベンチマーク²⁾にみられるメモリ参照数の減少効果を図3に示す。

3 AI32アーキテクチャの概要

本章では、AI32システムをソフトウェア構成、外部インタフェース及び内部アーキテクチャの三つの観点から説明する。

3.1 ソフトウェア構成

AI言語とコンピュータの機械語との間のセマンティックギャップ(意味の隔たり)が大きいため、AI言語で作成されたソフトウェアは通常その言語に対応する命令セット(中間言語であり、バイト基調であるためバイト命令とも呼ばれる。)にコンパイルされる。AI言語の処理システムを実現するには、上記のバイト命令をエミュレートするソフトウェアをホストマシンの上に作成すればよい。エディタ、コンパイラ、デバッガなどのシステムソフトウェアもみなこのバイト命令にコンパイルされているため、バイト命令のインタプリタを実現すれば、これらのシステムソフトウェアも利用可能になる。

従来、AI言語処理システムの高性能化を行うために、システムメーカーは大別して二つのアプローチを採っている。一つの方法は、コンパイラやデバッガなどのソフトウェアを設計し直して現行のはん用マイクロプロセッサの命令セットに直接コンパイルし、効率よく実行する直接コンパイル方式である。もう一つは、専用VLSIによってAI言語の中間言語をそのまま実行する直接実行方式である。これらのそれぞれの方式の利害得失を表2に示す。ここに報告するAI32は後者のアプローチを採った。AI32はコンパイラでAI言語を中間言語に

※2) Smalltalk-80は、米国ゼロックス社の登録商標である。
※3) Post Scriptは米国におけるAdobe Systems Incorporatedの登録商標である。
※4) Adaは米国国務省の開発したプログラミング言語であり、米国における米国政府AJPO(Ada Joint Program Office)の登録商標である。

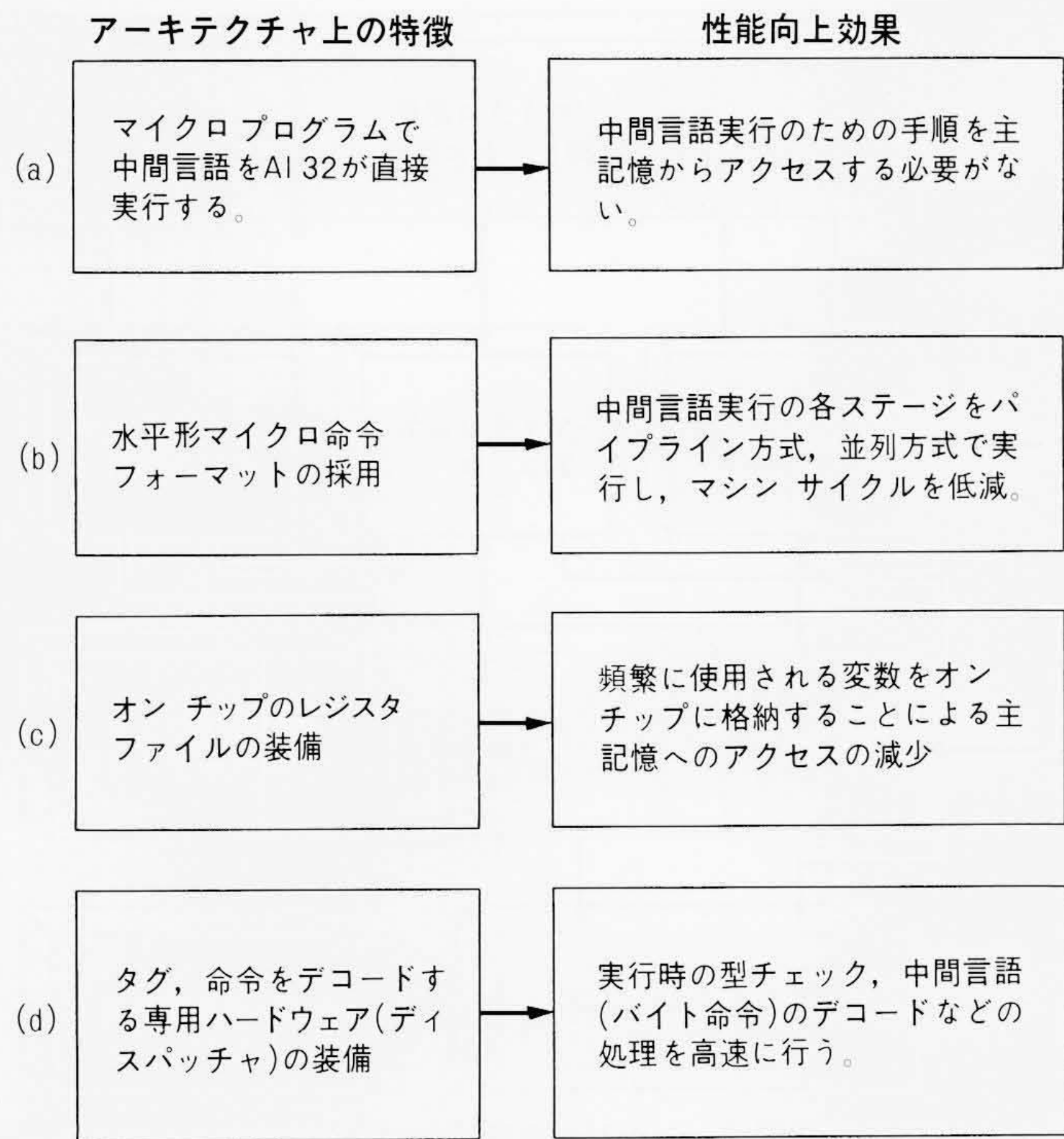


図2 AI32で性能向上が得られる理由 AI32では様々なアーキテクチャ上の工夫により、AI言語実行の際の性能向上を得ている。

変換し、内部のマイクロプログラムによりAI言語の中間言語をエミュレートして直接実行するプロセッサである。

3.2 外部インタフェース

AI32はMPUの管理下で動作する周辺LSIである。AI32の内部リソース(制御用レジスタや演算器、レジスタファイルなど)は、すべてメモリマップされている。AI32はアドレッシング能力を持ち、バスアービトレーションによりバスマスタになる。MPUがAI32の制御用レジスタ中のスタートビットをセ

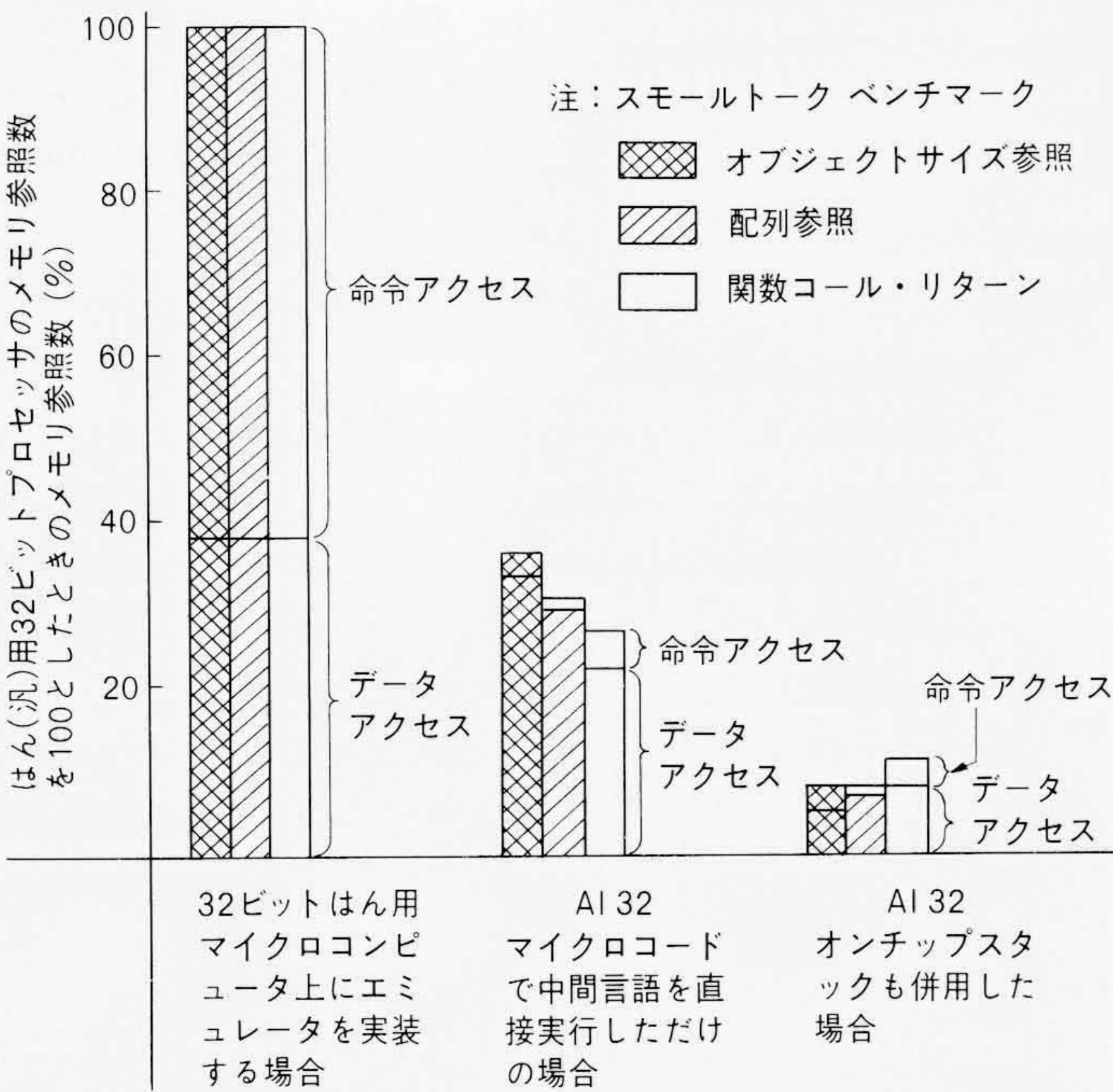


図3 ベンチマーク実行の際のメモリ参照頻度の比較 AI32では、AI言語の中間言語を直接実行することによって命令アクセス頻度を減らし、レジスタファイルを使用することによってデータアクセス頻度を減らし、メモリアクセス数をトータルで1けた減らすことができる。

ットすることによって、AI32は動作を開始する。バスアービトレーションを行い、バス権を獲得後所定のメモリロケーションにアクセスしてプログラムの実行を始める。AI32は必要に応じて外部メモリをアクセスするので、AI32が実行するときにMPUはバスを放棄した状態で待機している。AI32が外部メモリアクセス中にページフォールトが発生したり、入出力機器からMPUへの割込みが発生した場合、AI32はバスを解放してMPUに処理をゆだねる。

表2 AI言語の各アプローチによる利害得失 専用VLSIの導入によって、AI言語ソフトウェアのデバッグ性と移植性を損なわずにAI言語システムの開発が行え、ユーザーはソフトウェア開発費を大幅に節減できる。

	機械語への直接コンパイル方法	AI32, 専用VLSIでの直接実行方式
長所	●はん用MPUが使用でき、ハードウェアの付加なしにある程度の高速化が得られるため、はん用ハードウェアがそのまま使える。	●処理系の構成が単純であるため、AI言語システムが短期開発できる。 ●異なるシステム間での移植性が良い。 ●メモリ参照が少ないために、安価なシステムでも高速実行が可能である。 ●言語↔マシン間のセマンティックギャップが小さいため、プログラムのデバッグ効率が良い。 ●コンパイル時間が短い。
短所	●ソフトウェア開発(コンパイラ、デバッガ)に長期間を要する。プロセッサ、システムが変わった場合に、移植性が悪い。 ●実行時のエラーチェック機能が弱い。 ●機械語オブジェクトをメモリ上に格納するため、スペース効率が悪い。	●専用プロセッサを組み込まねばならず、システム面の開発を要する。 ●マイクロプログラムの開発を要する。

AI32のシステム内の位置づけは、図1に示したとおりである。最近の32ビットMPUでは、仮想アドレスから実アドレスへの変換の高速化のために、MMU(Memory Management Unit:メモリ管理ユニット)を内蔵しているものが多い。これらのMPUに対応するために、AI32もMMUを内蔵して仮想メモリに適応可能としている。

ユーザーがメモリを仮想化しないでAI32を使用する場合には、内蔵のMMUを40ビット入力・40ビット出力のアソシエティブキャッシュとして使用できる。ユーザーは40ビットのメモリアドレスを40ビットのアドレスへ変換するためのアドレス変換用テーブルを、このアソシエティブキャッシュに格納することができる。例えば、オブジェクトテーブルやメソッドキャッシュなど、AI言語で頻繁に参照されるテーブル類をキャッシュすることにより、AI32のメモリ参照頻度は更に減少する。

3.3 内部構成

多種類のAI言語にこのAI32を適応させながらそれぞれのAI言語を高速に実行するために、AI32では128ビットのワード幅を持つ水平形のマイクロ命令フォーマットを採用している。一つのマイクロ命令は六つの独立したフィールドから構成され、これらのフィールドによって、六つの独立した機能ブロックを並列に制御できる。各機能ブロックに定義されたオペレーションは、基本演算、スタック基本オペレーション、データフィールドによるディスパッチなどの基本オペレーションだけであり複合オペレーションはないため、特定の言語に依存するものではない。更にオンチップスタックも各ポイントレジスタに1本ずつ最大8本まで使用可能である。AI32のアーキテクチャの特徴を図2にまとめて示す。図4にAI32の内部構成を示す。AI32は、マイクロ記憶、マイクロ命令デコーダ、マイクロシーケンス制御、実行ユニット、アドレス変換ユニット及びバス制御ユニットの六つの各ブロックから成る。表3に各ブロックのハードウェア規模と仕様の一覧を示す。

4 マイクロプログラミングサポートツールの検討

本章ではAI32上にAI言語を処理するマイクロプログラムを開発する場合の手順、ツールなどの検討状況について述べる。

4.1 マイクロプログラム開発

ホスト機とAI32を搭載した拡張ボードを接続したシステムを用いて、対象とするAI言語を処理するマイクロプログラムをAI32上で開発する。図5に示したフローでマイクロプログラムを開発することによって、所望のAI言語を移植できるであろう。

4.2 マイクロプログラム開発用ツール

マイクロプログラムを、ユーザーに対して解放する際に必要となるツールとして下記のものがある。

- (1) マイクロアーキテクチャの詳細仕様を、正確にユーザーに伝達できるドキュメントを開発ツールの一つとしてユーザーに提供する。
- (2) ユーザーがチップの評価やデバッグを効率よく行える開発用ツールを供給する。

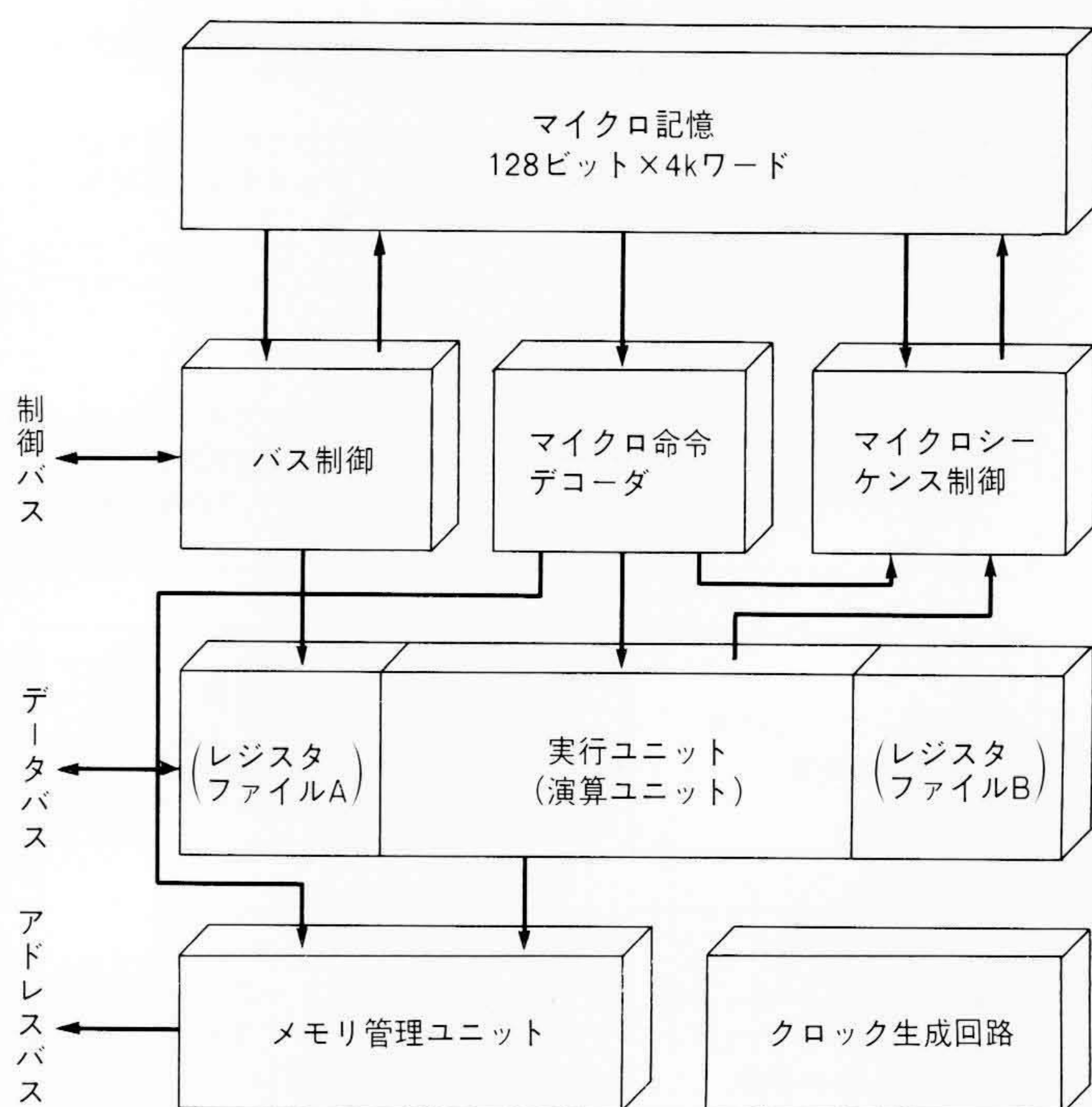


図4 AI32の内部構成 AI32の内部は六つの構成要素から成り立っている。各ブロックの機能は、マイクロプログラマにとって容易に理解できるように明快な構成にしてある。

4.2.1 マイクロアーキテクチャの記述方法

AI32の一命令は6個のフィールドから成り、これらが更に合計44個のサブフィールドから成る。サブフィールドの値は独立にセットされ、これらの値に基づいて44個のマイクロオペレーションが並列に実行される。各マイクロオペレーション間の同期が、どのように行われるかを詳細に示すことのできるソフトウェアシミュレータを公開することを予定している。

4.2.2 AI32評価キットの構成

ユーザーが所望のシステム上にAI32を搭載する場合に、最終的にはターゲットシステム上でマイクロコードをデバッグすることが必要となる。しかし、初期のマイクロプログラム開発及び部分デバッグを効率よく行うために、AI32専用の評価ツールが必要となろう。ここではそのために使われるツールの内容について説明する。

(1) ホストマシン

顧客が容易に購入できる安価なパーソナルコンピュータをホストマシンとする。

(2) 評価システム

評価システムの構成を図6に示す。ユーザーは、この評価システム上でマイクロプログラムの開発、テスト、デバッグ、ERPOMのプログラムを行うことができる。マイクロプログラムの動作を1ステップごとにモニタしながらトレース作業を行えるシングルステップトレース機能、RAM上にマイクロプログラムを置いて、これを1ステップずつ実行する実チップシミュレーション機能など、マイクロプログラム開発を行う際に便利な機能が盛り込まれている。また、ユーザーによって作成されたデバッグ前のマイクロプログラムを格納した

表3 各ブロックのハードウェア規模と仕様一覧 AI32は、現在最高のプロセス技術を利用して、AI言語を高速に実行するハードウェアをワンチップ上に集積している。

項 目		仕 様
プ ロ セ ス		CMOS 1.3μm 2層アルミ
マ イ ク ロ	ロ 記 憶	オンチップEPROM 4kワード×128ビット
マイクローシークエンス制御	マイクロ命令パイプライン段数	3段
	マイクロサブルーチンネストレベル	3レベル
マイクロ命令実行部資源	レジスタファイル	40ビット×256レジスタ×2ファイル
	ポインタレジスタ	40ビット×4レジスタ×2ファイル
	演算ハードウェア	●40ビットはん用ALU ●ポインタレジスタ専用加算器×2 ●バレルシフタ
	内部バス	32ビット 40ビットのオプションあり。
メモリ管理ユニット部分	論理アドレス領域	～1Tバイト
	物理アドレス領域	～1Tバイト
	アドレス変換バッファ	40ビット フルアソシヤティブ
	機能	●デマンドページ仮想メモリサポート ●テーブル変換用キャッシュ
外部バス構成	アドレスバス	40ビット
	データバス	40ビット
	転送方式	同期式
	バスサイジング	16ビット・32ビット・40ビット
動作モード		●マスタモード(AI32がバスマスタとなる。) ●スレーブモード(AI32はチップセレクトされる。)
マイクロプログラムのデバッグ機能		●マイクロ命令トレース機能 ●マクロ命令トレース機能 ●実チップシミュレーション機能
パッケージ		ピングリッド アレー形(170～240)
ディスプレイタッチャ		●256方向分岐 ●タグマスク機能付き
EPROMプログラム方式		実装状態でプログラム可能

注：略語説明など EPROM(Erasable Programmable Read Only Memory)
レジスタファイル(関数呼出し用スタックなどに使われる大容量レジスタ)

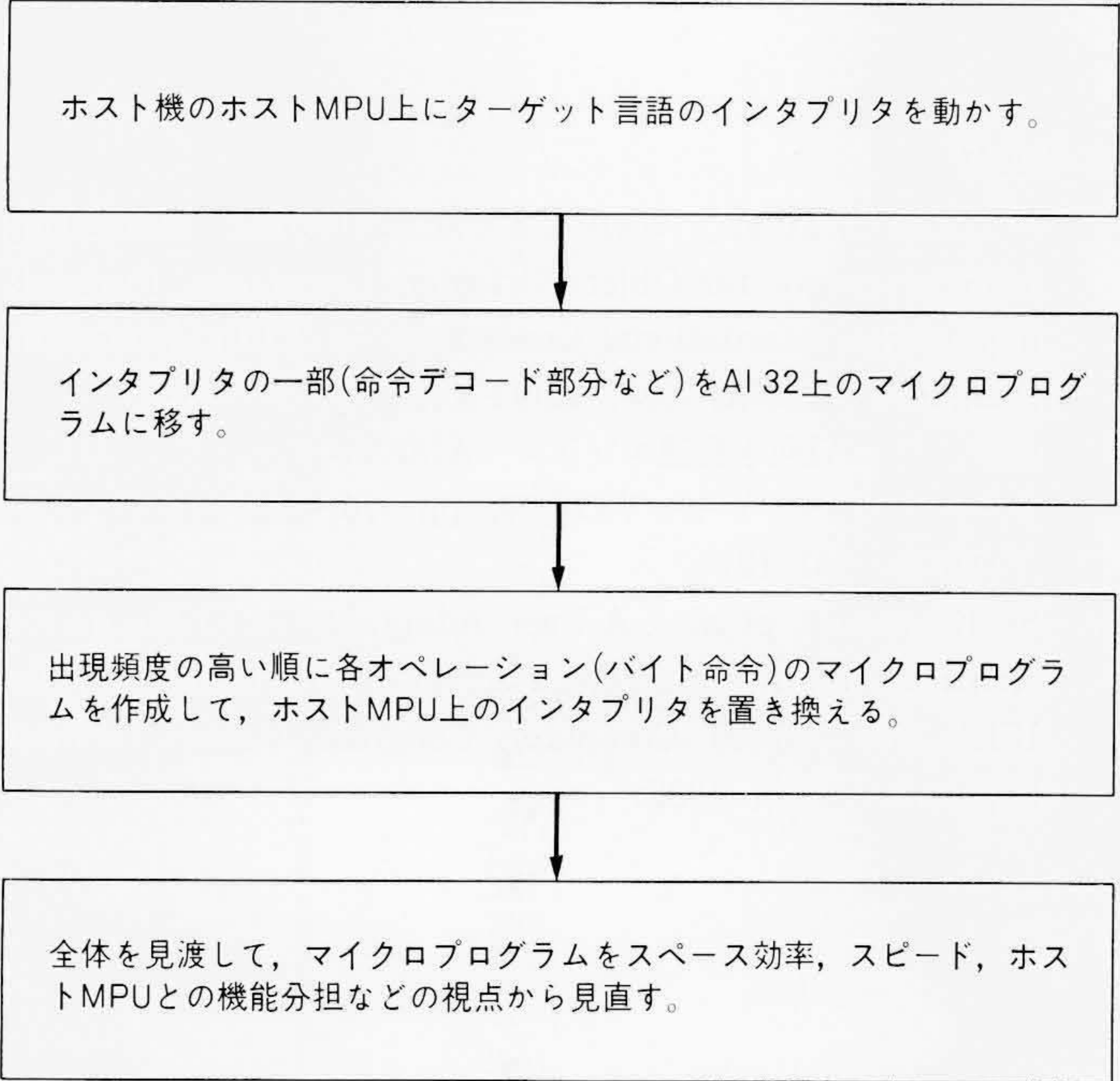


図5 AI言語用マイクロプログラムの開発フロー ホストシステムのMPU上に、AI言語用インタプリタを作成し、徐々にその機能をAI32のマイクロプログラムに移していくことによって、連続性を持ったシステム開発が可能である。

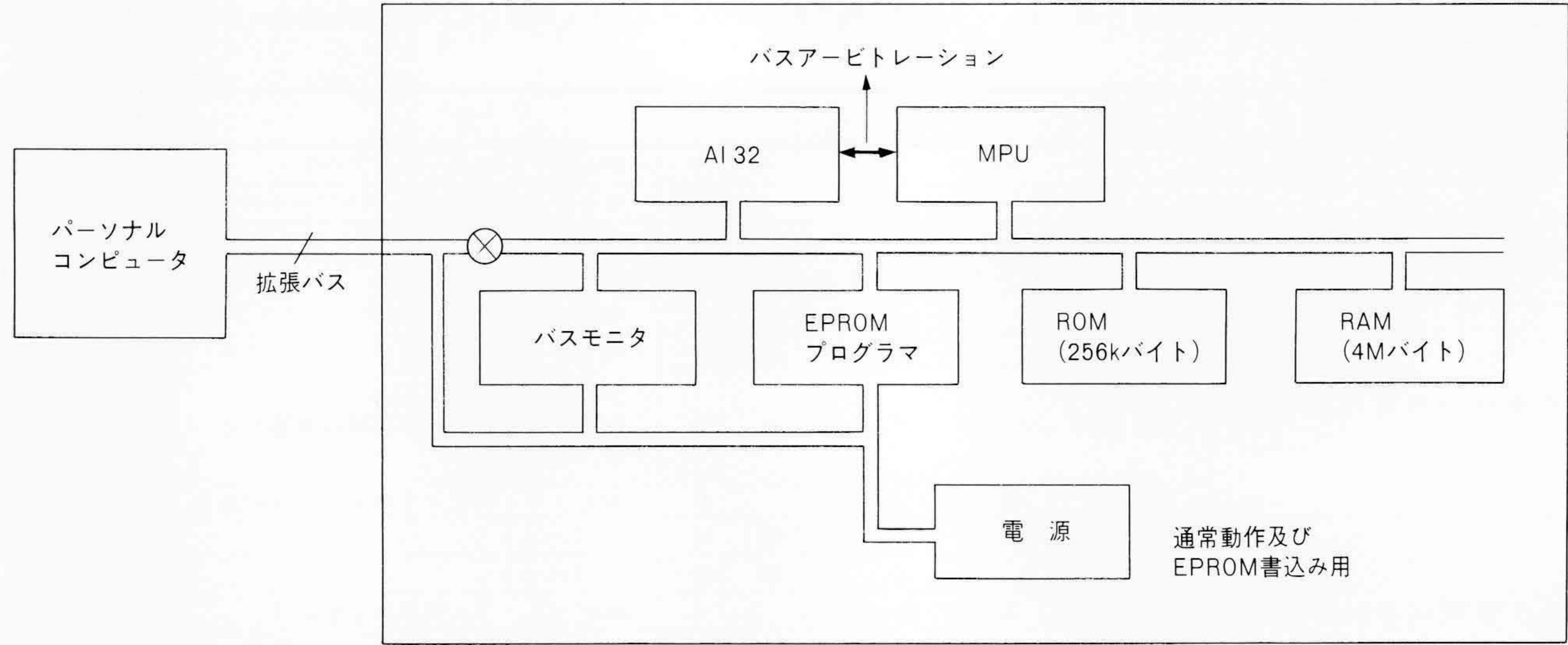
AI32がホストとなるパーソナルコンピュータ上のモニタソフトウェアを破壊することがないようにバススイッチが設けられており、AI32をホストシステムから切り離して動作させることができるようになっている。この場合には、ユーザーはバスモニタを介してAI32のバスオペレーションをモニタすることができるようにする。

(3) サポートソフトウェア

図7に評価システム上で動くサポートツールソフトウェア構成を示す。サポートソフトウェアには、マイクロアセンブラ、デバッガモニタ、EPROMプログラマなどがある。ユーザーは、これらを用いてホスト上でアセンブル、コンパイルしたホストMPU用のロードモジュールやAI32のマイクロプログラムオブジェクトを評価ボードにダウンロードし、起動、デバッグすることができる。マイクロプログラムのシミュレーションは実チップ上で高速に行うことができる予定である。

5 結 言

AI32の特長、応用範囲、性能など、システム開発の実際、サポートツール検討状況について述べた。特に性能に関しては、同一プロセス技術で設計された32ビットMPUに比較して



RAM (マイクロアセンブラで作成のマイクロコードとダウンロードワーク用)
バスモニター (バススイッチを開放して、ボードをパーソナルコンピュータから切り離れたときにバスの状態をモニターする。)
EPROMプログラマ (AI 32上のEPROMを実装状態でプログラムする。)
注: 略語説明 RAM(Random Access Memory), ROM(Read Only Memory), EPROM(Erasable Programmable ROM)

図6 AI32サポートツールのハードウェア構成 AI32のサポートツールの一環として、AI32を搭載したパーソナルコンピュータ用拡張ボードを検討中である。ユーザーは、この上でマイクロプログラムの開発、AI32EPROMの書き込み、マイクロプログラムの評価を行うことができる。

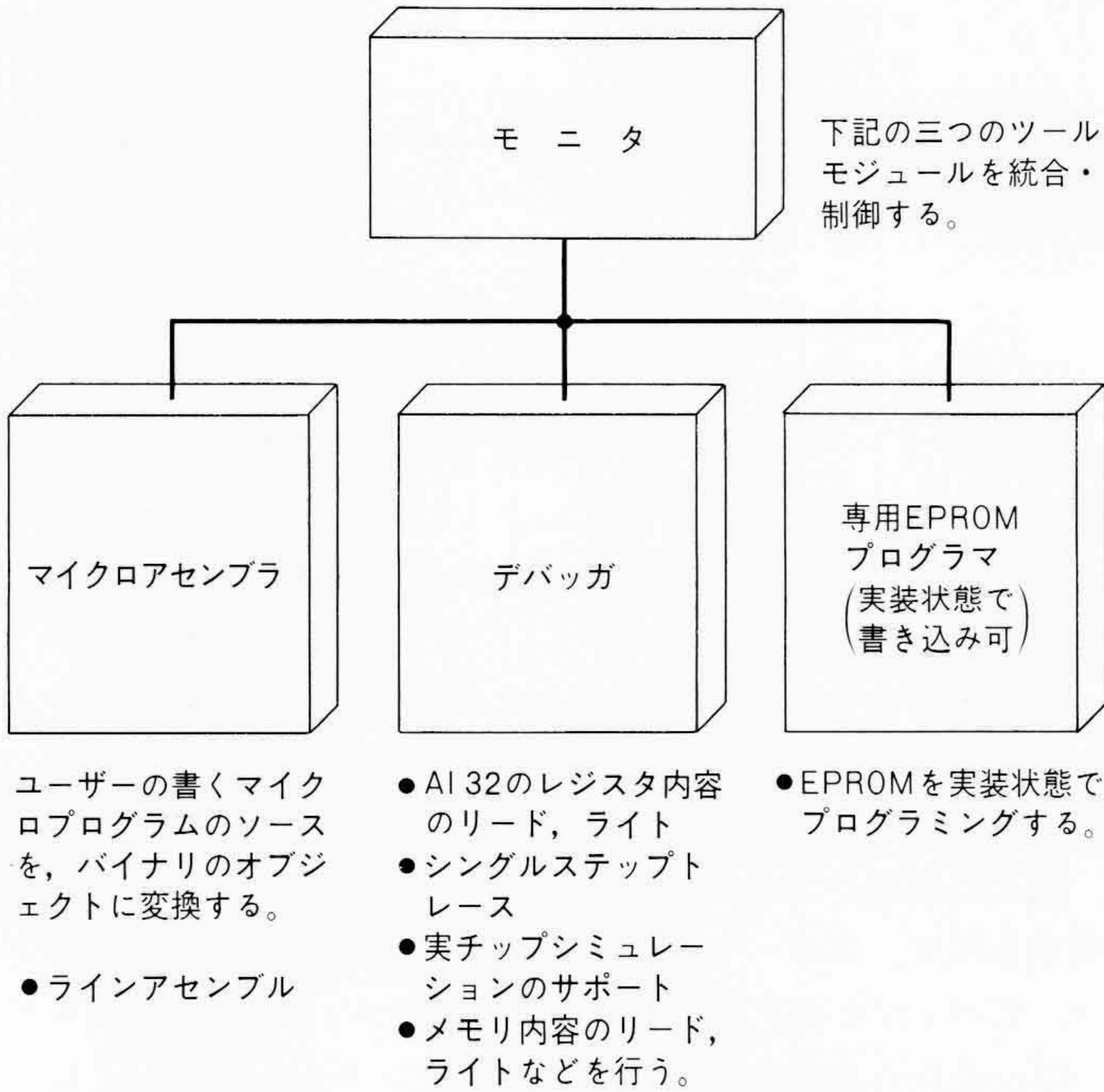


図7 AI32のサポートツールのソフトウェア構成 AI32のサポートツールは、マイクロアセンブラ、デバッガ、EPROMプログラマなどを統合している。マルチウインドウを基本としたツール体系を検討中である。

も4倍強の性能が得られる見通しである。AI32を部品として用いることによって、AI言語が本来持っている特色(移植性やデバッグの容易性及びシステムの構成)を損なうことなく、高速なAIシステムを構築することが可能となる。

参考文献

- 1) N. Suzuki, et al. : Sword32 : A Bytecode Emulating Microprocessor for Object-Oriented Languages, Proceedings of the International Conference on Fifth Generation Computer Systems 1984, ICOT 1984 pp. 389~396.
- 2) 稲吉, 外 : AI向き高速エンジン「AI32コプロセッサ」, 32ビット・マイクロ・プロセッサの全容, pp. 76~84, 日経データ・プロ編集(昭61-12)
- 3) S. Kawasaki, et al. : A User-Adaptable VLSI Engine for Artificial Intelligence, pp. 367~372, Proceedings of the IFIP 10th World Computer Congress, Dublin, Ireland, September 1~5, 1986.