U.D.C. 621.3.049.774′14:681.323-181.48

システム集積形オリジナルマイクロプロセッサH16

System Integration Oriented Original Microprocessor H16

馬場志朗* *Shirô Baba*
渡辺　坦** *Tan Watanabe*
前島英雄*** *Hideo Maejima*
倉員桂一* *Keiichi Kurakazu*

情報機器，産業分野の機器に内蔵して使われる制御用マイクロコンピュータには，現在，8ビットシングルチップマイクロコンピュータが使われているが，応用機器の高度化に伴い，高性能化の要求が強い。H16は，制御応用をねらって開発されたマイクロプロセッサであり，オリジナル命令セットを持つ内部32ビット構成のCPUを中心に，システム構成に必要な周辺機能をオンチップ化している。

CPUアーキテクチャでは，リングバンクモード付きのマルチレジスタバンク方式を開発することにより，制御応用で重要なリアルタイム応答性を従来方式に比べて約9倍，高級言語での副プログラムの呼出し速度を約2.5倍に改善した。また，周辺機能の内蔵により，応用システムのプリント基板サイズ縮小が可能となった。

1 緒　　言

マイクロプロセッサの開発では，32ビットの高性能マイクロコンピュータが次々と発表され，話題は32ビットに集中している。しかし，実際の市場では，依然として8ビットあるいは16ビットが大量に使われている。特にHD6301に代表される8ビットシングルチップマイクロコンピュータは，OA (Office Automation)，産業，民生などの分野の機器に組み込まれて広く使われており，その用途はますます拡大している。一方，これらの応用機器の機能，性能の向上が進むにつれ，最近では既存のシングルチップマイクロコンピュータでは能力不足となるケースが増加している。例えば，プリンタの場合，レーザービーム形プリンタは従来のワイヤドット形に比べて，扱うデータ量及び処理速度が飛躍的に増大しており，8ビットシングルチップマイクロコンピュータでは対応が難しい。このような場合，現状では，16ビット又は32ビットのマイクロプロセッサが使われている。しかし，シングルチップマイクロコンピュータに比べて，コストが増加すること，部品点数が増えてプリント基板サイズが大きくなることなどの問題がある。

H16はこれらの問題点の解決を目指して開発された16ビット高性能・高集積形マイクロコンピュータであり，主に機器に内蔵して使われるコントローラ応用をねらいとしている。オリジナル命令セットを持つ32ビット構成のCPU (Central Processing Unit)を中心に，RAM (Random Access Memory)，DMAC (Direct Memory Access Controller)，タイマ，シリアル通信機能などをワンチップに集積している。H16の開発に当たっては，以下を目標に開発を行った。

(1)　高性能CPU

リアルタイム応答性に優れ，大量のデータを高速に処理できる32ビットオリジナルCPUの実現

(2)　システムインテグレーション

システムコストの低減，プリント基板サイズの縮小を実現するため，システム構成に必要な周辺機能をオンチップ化する。

(3)　高級言語の効率的実行

ソフトウェアの生産性向上の流れに対応するため，C言語などの高級言語を効率的に実行できるCPUアーキテクチャと，それを最大限に活用する高効率Cコンパイラの実現

(4)　総合開発環境の提供

LSI，アセンブラ，Cコンパイラ，リアルタイムエミュレータなどを含む使いやすいソフトウェア開発環境の提供

2 概　　要

表1に仕様の一覧を，**図1**にブロック図をそれぞれ示す。32ビット構成のCPUを核に，タイマ，シリアル通信機能，DMACなど，システムの構成に必要な周辺機能を内蔵しており，いわゆるシステム集積指向マイクロプロセッサである。

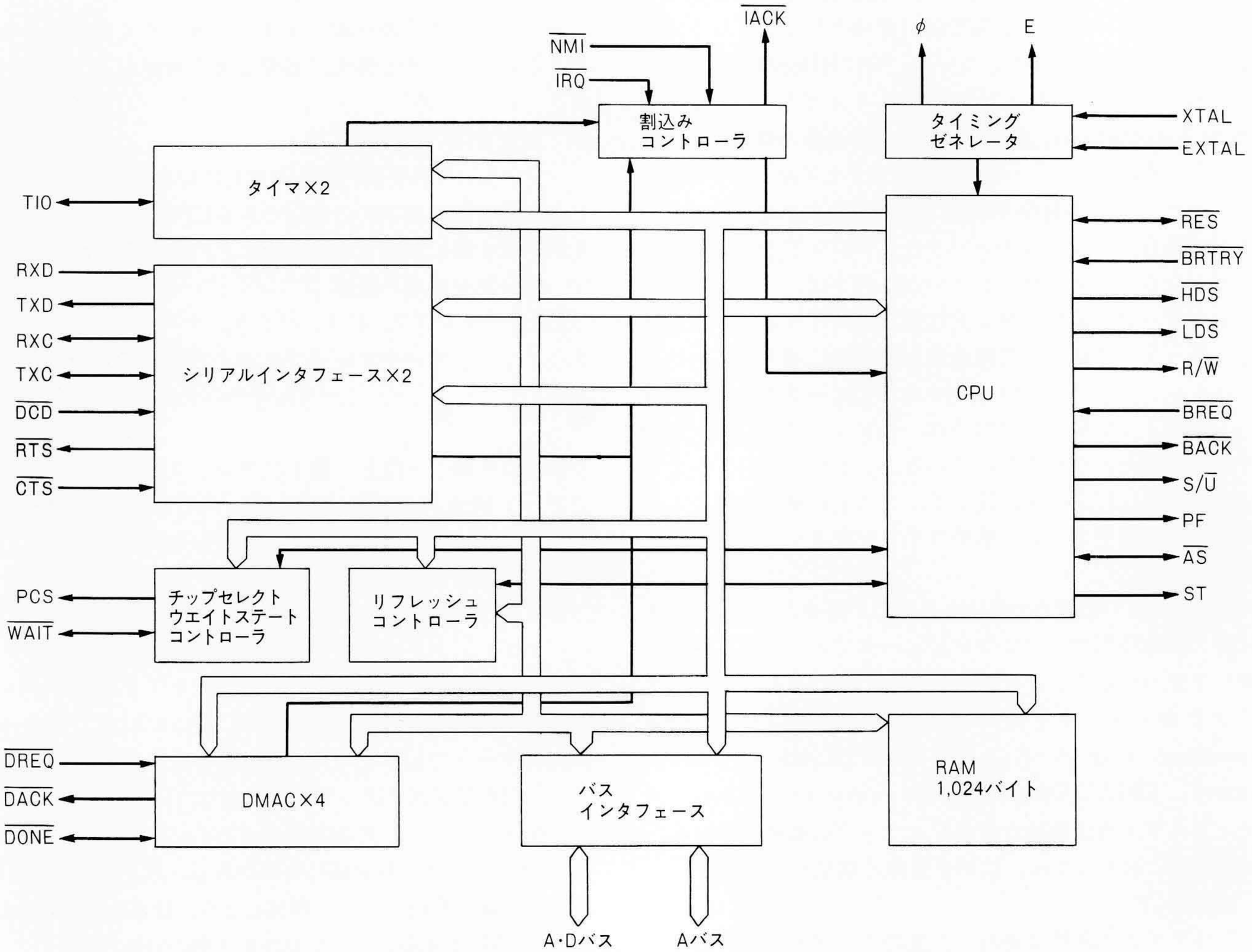
特に，DRAM (Dynamic Random Access Memory)のリフレッシュコントローラやウエイトステートコントローラ，チップセレクトコントローラなどのメモリアクセスサポート機能を内蔵しており，従来標準TTL (Transistor Transistor Logic)ゲートで組まなければならなかったいわゆる「雑ロジック」の大幅削減を図っている。また，DMACの内蔵と，それに伴うバスアービタの内蔵もバスインタフェースを簡単にし，「雑ロジック」の削減に効果がある。以上のように，周辺機能の内蔵とTTLゲートの削減により，H16は従来LSI 4個，TTL約15個で実現していたものを1個のLSIで実現している。

* 日立製作所武蔵工場　** 日立製作所システム開発研究所　*** 日立製作所日立研究所 工学博士

表1 H16の仕様一覧表 H16は，32ビット構成のCPUに加えて，豊富な周辺機能を内蔵している。

項目		仕様
CPU	レジスタ	32ビット×16本×最大16組み
	アドレス空間	16Mバイト(将来4Gバイトへ拡張)
	基本命令数	98
	アドレスモード	13
	演算器	32ビットALU，バレルシフタ内蔵
	命令体系	2アドレス方式・命令長はバイト単位
	最小命令実行時間	0.5μs(8MHz)(レジスタ間のデータ転送)
内蔵周辺機能	RAM	1,024バイト レジスタバンク領域，又は高速内蔵メモリとして使用可
	DMAC	4チャネル メモリ↔メモリ，メモリ↔I/O， メモリ↔内蔵I/O転送 シングル・デュアル アドレスモード コンティニュー動作可
内蔵周辺機能	タイマ	16ビット2チャネル インターバルタイマ，PWM出力，2相パルス出力 ワンショットパルス出力，イベントカウントなど各種動作可
	シリアルインタフェース	2チャネル 調歩同期・クロック同期選択可能 ボーレートゼネレータ内蔵
	割込みコントローラ	割込み端子3本 4レベルの優先順位 外部ベクタモード・内部ベクタモード選択可能
	メモリアクセスサポート機能	DRAMリフレッシュコントローラ プログラマブルウエイトステートコントローラ チップセレクトコントローラ
プロセス		1.3μm CMOS
パッケージ		84ピンPLCC

注：略語説明 DMAC(Direct Memory Access Controller)
ALU(Arithmetic Logic Unit)
I/O(入出力装置)
PWM(Pulse Width Modulation)
DRAM(Dynamic Random Access Memory)
CMOS(Complementary Metal Oxide Semiconductor)
PLCC(Plastic Leaded Chip Carrier)



注：略語説明 CPU(Central Processing Unit)

図1 H16ブロック図 32ビット構成のCPUを中心に，RAM，タイマ，シリアルインタフェース，DMACなどシステムの構成に必要な周辺機能を内蔵している。

これによりシステムを構成した場合のプリント基板サイズを，機器内蔵形の応用に十分使用できるサイズに縮小することが可能となった。

また，システムの部品点数の減少は，システム設計工数の低減にも効果がある。例えば，DMA(Direct Memory Access)機能を含むような複雑なシステムを，8ビットシングルチップマイクロコンピュータを使用したシステムの設計を行うのと同様な手軽さで実現できる。

このように，H16は，機器内蔵形マイクロコンピュータの機能，性能をプリント基板サイズ，設計の手軽さを損なうことなく大幅に向上している。

3　CPUの特長

H16CPUは，一般的な処理速度の向上に加えて，コントローラ応用で特に重要なリアルタイム応答性の改善を目標に開発されている。また，今後重要になると予測される文字と図形の混在データの処理に必要なビットフィールドデータ処理能力の強化も目標のひとつであった。一方，ソフトウェアの生産性向上を考えると，今後は制御応用といえどもC言語に代表される高級言語の使用が必要であり，これへの対応も考える必要があった。

以下，H16 CPUアーキテクチャの概要を説明する。なお，高級言語指向アーキテクチャの詳細については，**4**章で詳述する。

3.1　レジスタ構成

図2にレジスタの構成を示す。16本のはん(汎)用レジスタと12本の専用レジスタを持つ。はん用レジスタはすべて32ビット長であり，いずれもデータの格納，アドレスポインタ又はインデックスレジスタの目的に使用できる。また，レジスタ，プログラムカウンタが32ビット長であるので，最大4 Gバイトのリニアアドレス空間を持つことができる(現行のH16は端子の制約により16Mバイトとなっている)。また，内蔵ALU(Arithmetic Logic Unit)も32ビット幅となっているため，32ビット長のデータの基本演算を16ビットデータと同じ速度で実行可能である(8MHz動作時レジスタ間の加算が0.625μs)。このように内部32ビット構成となっているため，レーザビームプリンタや日本語ワードプロセッサに代表される数メガバイトものデータの高速処理を必要とする応用にも対応可能である。

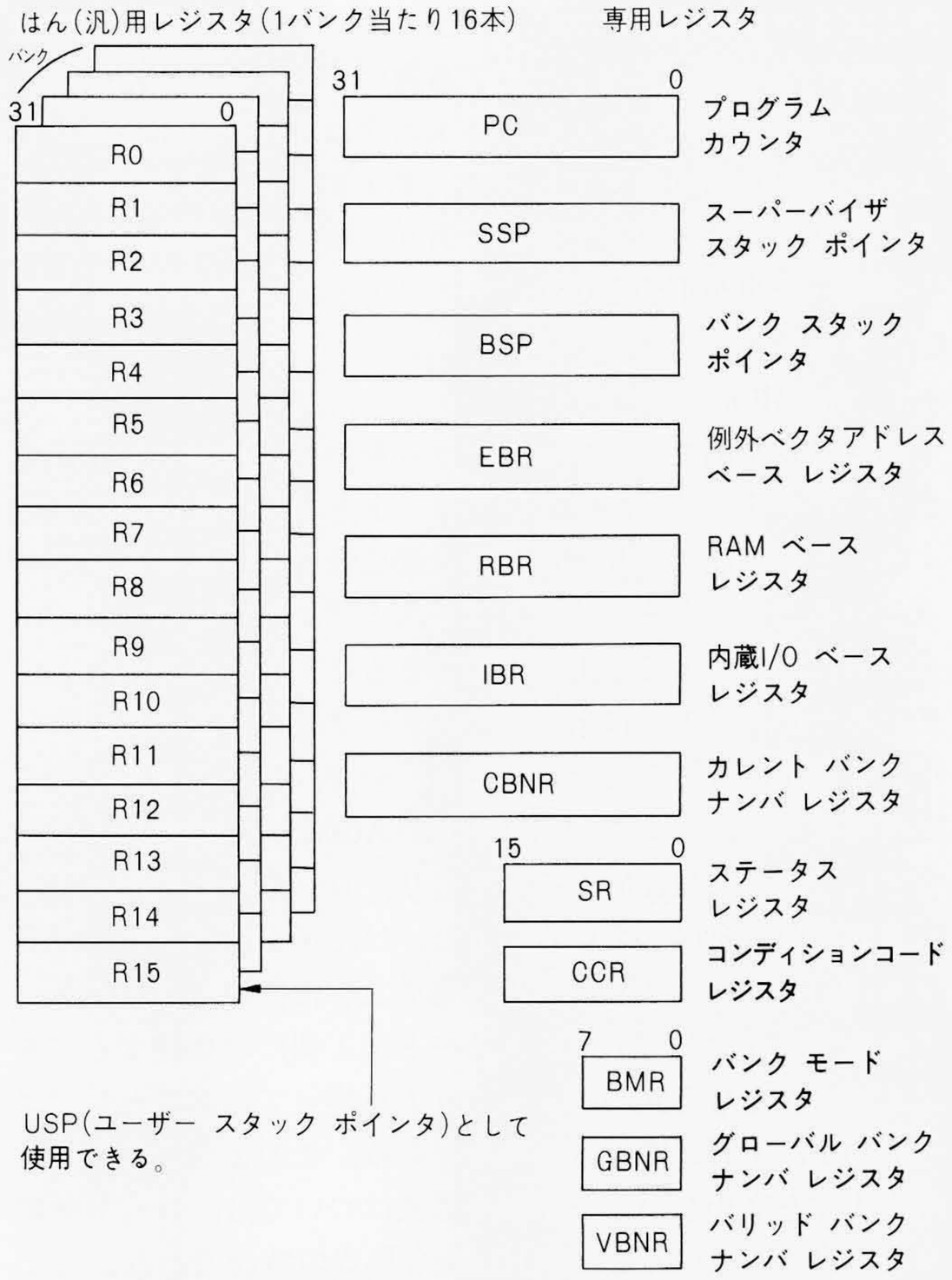
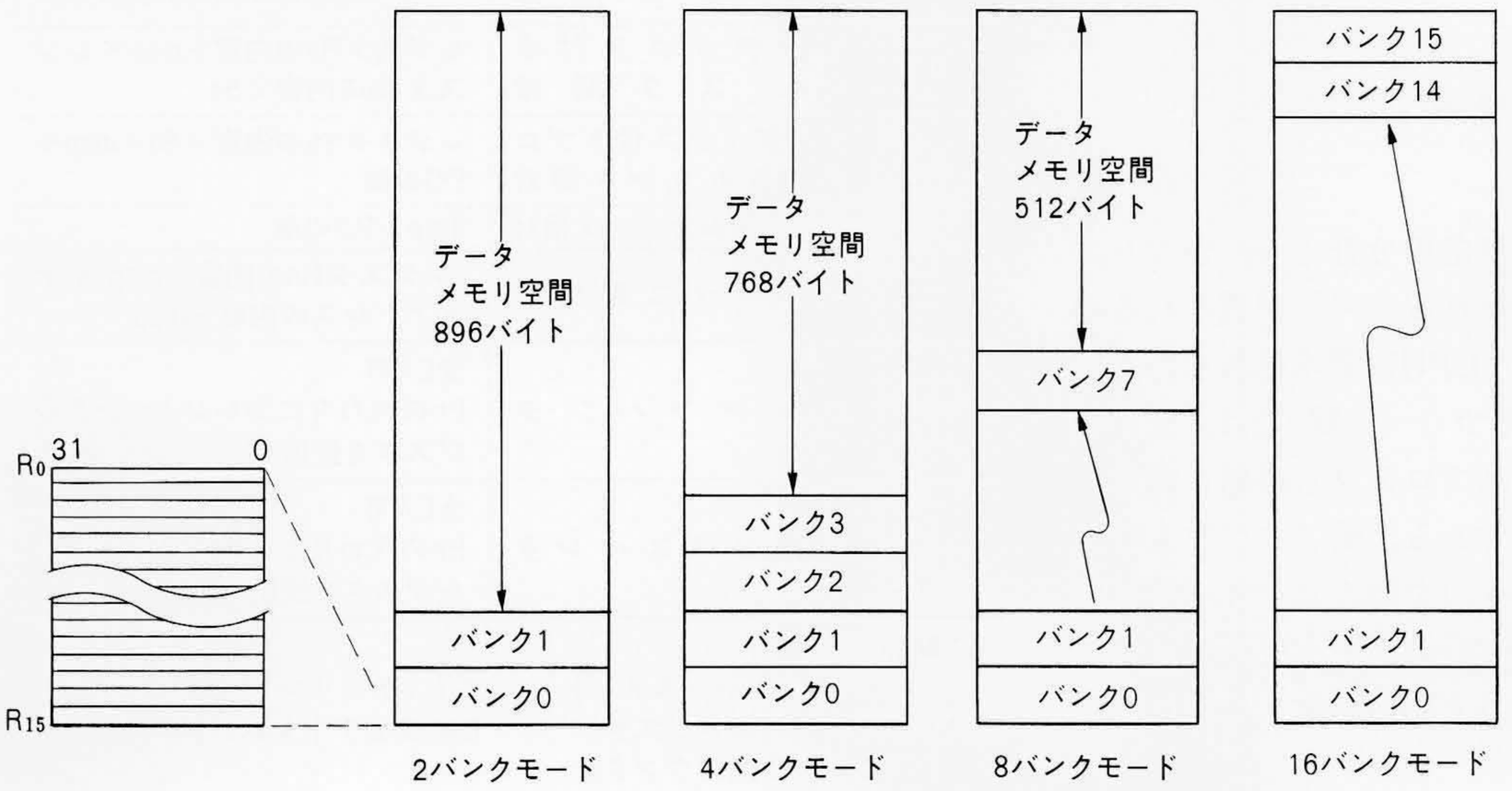


図2　レジスタ構成　32ビット長のはん用レジスタを16本，専用レジスタを12本持つ。

次に，リアルタイム応答性の向上，高級言語の高速実行を目的としたレジスタバンク方式について説明する。H16は16本のはん用レジスタセット(レジスタバンク)を最小2組み，最大16組み持つことができる。**図3**に示すように，これらのはん用レジスタの実体は，1,024バイトの内蔵RAM上に存在し，



注：略語説明
R_0～R_{15}(はん用レジスタ0～15)

図3　レジスタバンクの構成
はん用レジスタの実体は内蔵RAM上に存在する。16本のはん用レジスタを最小2組み，最大16組み(バンク)持つことができる。レジスタ領域以外は，高速のオンチップデータメモリとして使用できる。

例えば2バンクモードの場合，128バイトがレジスタ領域，残り896バイトが通常のオンチップ高速データメモリ領域となる。リアルタイムプログラムで，各タスクにそれぞれ1個のレジスタバンクを割り当てておけば，タスクのスイッチはレジスタバンクの切替えだけで可能であり，時間のかかるレジスタの退避・回復が不要となり，タスクスイッチ時間を約$\frac{1}{9}$に短縮できる[1)]。

3.2 命令体系

図4に基本的な命令フォーマットを示す。特長としては，命令，データサイズ，アドレスモードをそれぞれ独立に指定することが可能であり，いわゆる直交性の高い命令体系となっていることである。また，ソースとディスティネーションの両方にアドレスモードを指定できる2アドレス方式をとっているため，メモリにある二つのデータをレジスタを経由しないで直接演算することができる。一方，プログラム中での出現頻度の高いレジスタ間のADD(Addition)やMOVE命令などについては短縮形を持っており，コード効率の向上，実行速度の改善を図っている。また，はん用レジスタのうちの1本(R_0)をアキュムレータとみなすことにより，2オペランド命令でありながらアドレスモードを1個だけ指定するアキュムレータ命令形式のフォーマットを持っていることも，コード効率，実行速度改善に効果がある。このように柔軟性，はん用性を高めながら一方頻出形式については，コード効率などを改善していることがH16命令形式の特長である。

表2にアドレスモードの，**表3**に命令セットの概要を示す。

3.3 特色ある命令

特に特色のある命令としては以下がある。

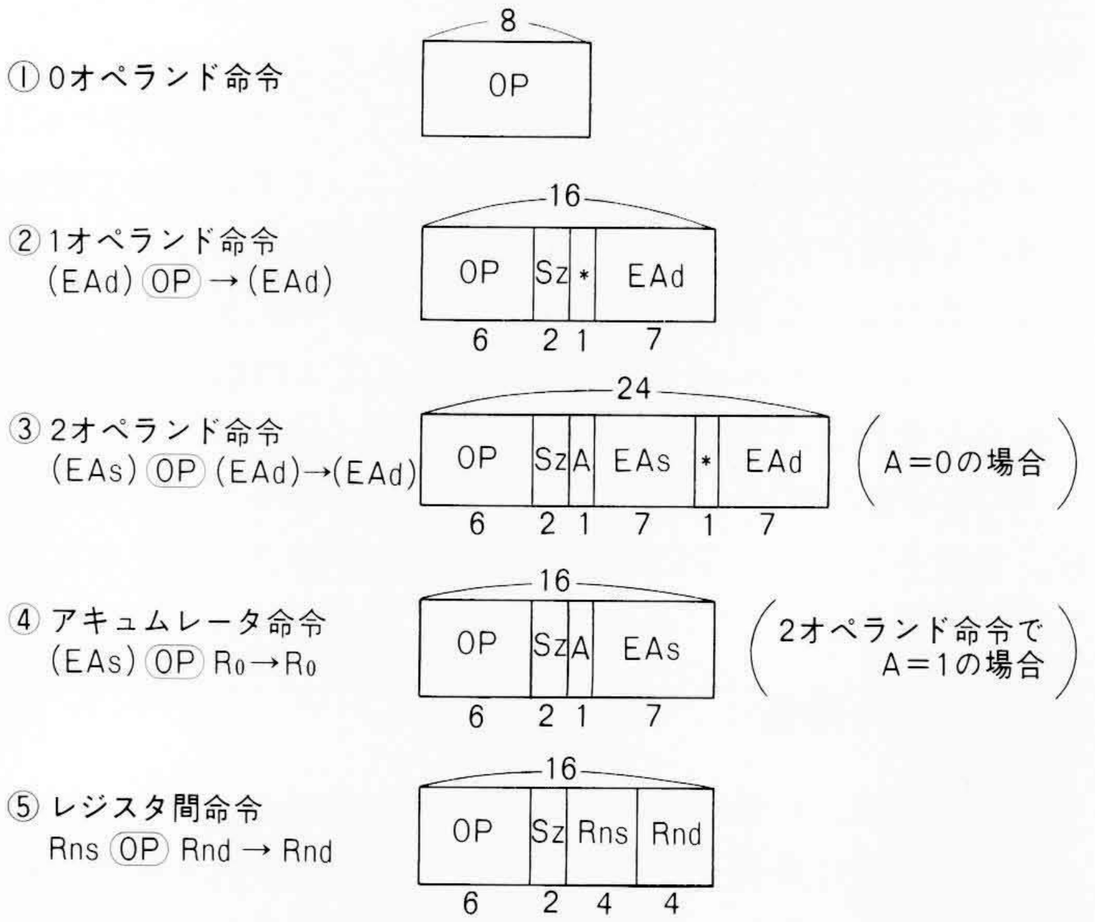
(1) バンクの切替えを行うバンクスイッチ命令
(2) ビットマップ表示に便利なビットフィールド命令
(3) ストリングデータの転送・比較命令
(4) 低消費電力モードへ移行するためのSLEEP命令

これらのうち，ビットフィールド命令やシフト命令は，内蔵しているバレルシフタを活用して，ビット位置にかかわらず一定時間で実行する仕様となっている。**図5**にビットフィールドMOVE命令の動作を示す。この命令は，**図5**に示す動作を一命令で実行する。したがって，例えば，フォントデータをビットマップ表示用の画像フレームメモリへ高速に転送するのに有効な命令である。

4 高級言語指向アーキテクチャ

4.1 高級言語プログラムの特性

今後は機器内蔵形のコントローラ応用分野でも，かなりの部分のプログラムが高級言語で書かれるものと予想されるので，それに適したアーキテクチャがCPUに要求される。高級言語プログラムの実行特性調査に基づく一見解[2)]では，文の種類別実行時間比率は，代入13%，副プログラム呼出し32%，分岐21%，反復32%，その他2%である。副プログラム呼出しは，内部状態の退避・回復を伴うので他の文よりも実行時間が長い。高級言語プログラムの実行速度を上げるには，このような大きい実行時間比率を持つ文を高速化しなければならない。



注：略語説明
OP(オペレーション指定フィールド)　Rns/Rnd(レジスタ番号指定フィールド)
EAs/EAd(実効アドレス指定フィールド)　A(アキュムレータ指定ビット)
Sz(サイズ指定フィールド)　＊(未使用)

図4　命令フォーマット　5種類の基本命令フォーマットを持つ。オペレーション指定フィールドのほかに最大2個のオペランド指定フィールドを持つ。

表2　アドレスモード　13種のアドレスモードを持つ。

No.	アドレスモード	実効アドレス
1	レジスタ直接	レジスタRn
2	レジスタ間接	レジスタRnの内容＋disp
3	レジスタ間接オートインクリメント	レジスタRnの内容(実行後Rnを自動的にインクリメント)
4	レジスタ間接オートデクリメント	レジスタRnの内容を自動的にデクリメントした値
5	イミディエート	オペランドデータはプログラム中の値がとられ，符号拡張される。
6	絶対アドレス	EA拡張部が実効アドレスになり，符号拡張される。
7	スケーリング付きレジスタ間接	レジスタRnの内容×Sf＋disp
8	インデックス付きレジスタ間接	レジスタRnの内容＋disp＋レジスタXnの内容×Sf
9	インデックス付きプログラムカウンタ相対	レジスタXnの内容×Sf＋disp＋PCの値
10	プログラムカウンタ相対	disp＋PCの値
11	レジスタ二重間接	{レジスタRnの内容＋disp}が示すアドレスの内容＋disp
12	カレントバンク	全EA可 Rnの代わりにカレントバンクレジスタを使用。
13	プレビアスバンク	全EA可 Rnの代わりにプレビアスバンクレジスタを使用。

注：略語説明
disp(ディスプレースメント)　Sf(スケーリングファクタ)
Rn，Xn(はん用レジスタ)　EA(実効アドレス)
PC(プログラムカウンタ)

表3　命令セットの概要　　算術，論理，比較，シフト，分岐などの基本命令に加えて，ビットフィールド命令，ストリング命令，バンク操作命令などを持っているところに特長がある。

命令区分	種類	代表例
算術演算	加算	ADD：G　2進加算
		ADD：R　レジスタ間加算
	減算	SUB：G　2進減算
		SUB：R　レジスタ間減算
	乗算	MULXS　符号付き乗算
		MULXU　符合なし乗算
	除算	DIVXS　符合付き除算
		DIVXU　符合なし除算
	符合反転	NEG　符合反転
論理演算	論理積	AND　論理積
	論理和	OR　論理和
	排他的論理和	XOR　排他的論理和
	否定	NOT　否定
比較	比較	CMP：G　比較
		CMP：R　レジスタ間比較
テスト	テスト	TST　テスト
シフト・ローテート	シフト	SHAL　算術左シフト
		SHLL　論理左シフト
	ローテート	ROTR　右ローテート

命令区分	種類	代表例
分岐系	分岐	Bcc：G　条件分岐
		BSR　サブルーチンコール
		RTS　サブルーチンからの復帰
単項演算	クリア	CLR　クリア
	サイズ拡張	EXTS　符合付き拡張
		EXTU　符合なし拡張
ビット操作	ビット操作	BTST　ビットテスト
		BSET　ビットテスト及びセット
ビットフィールド	抽出	BFEXT　抽出
	挿入	BFINS　挿入
	“1”のサーチ	BFSCH　“1”のサーチ
	繰返し転送	BFMOV　繰返し転送
ストリング	転送	SMOV　転送
	比較	SCMP　比較
転送	転送	MOV：G　転送
		MOV：R　レジスタ間転送
バンク	バンク切替え	CGBN　グローバルバンク切替え
		ICBN　リングバンク切替え
その他	スリープ	SLEEP　低消費電力モード

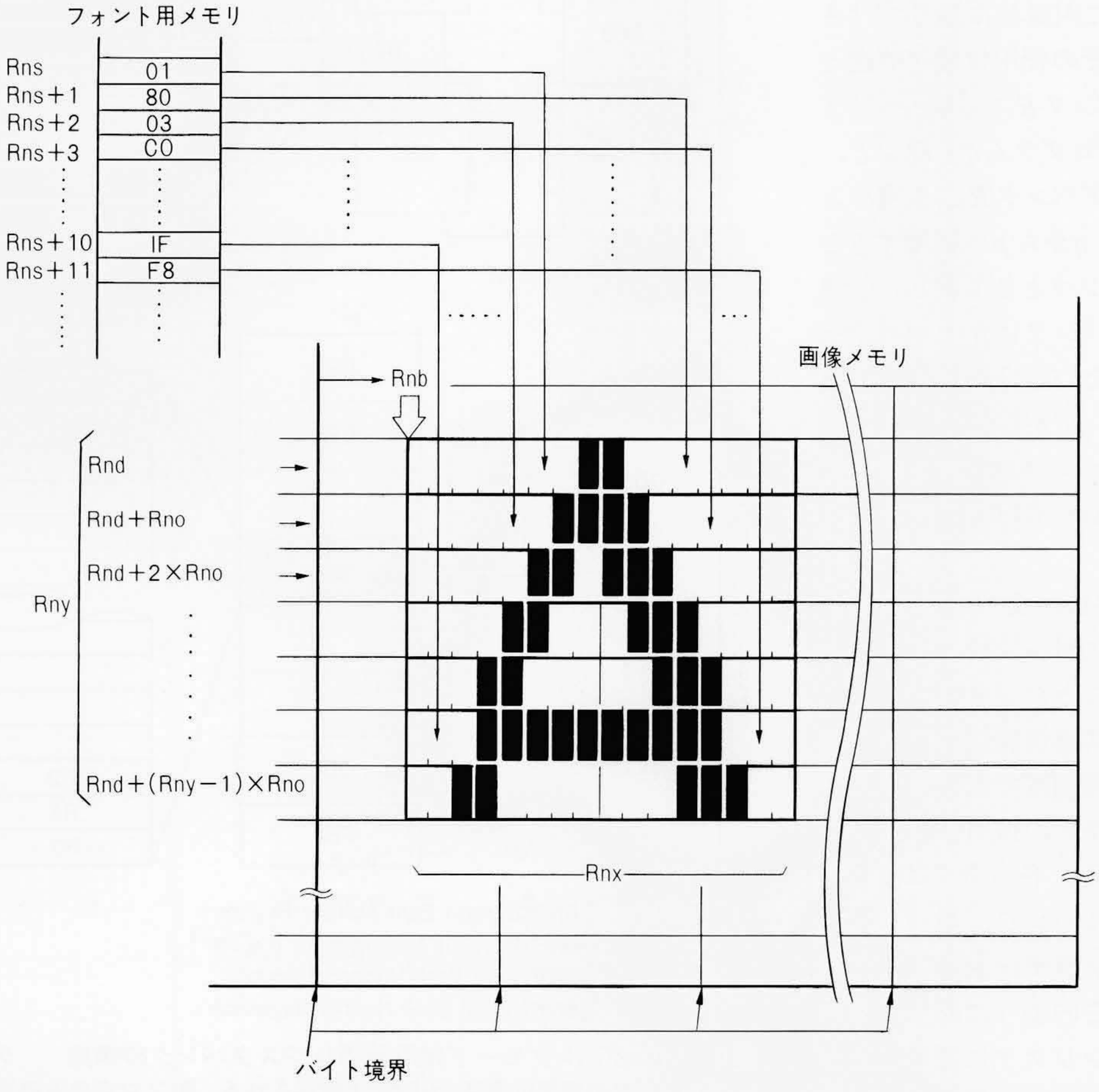


図5　ビットフィールドMOVE命令の動作　　Rnsで指定したロケーションからRnxバイトのデータをデスティネーションRndとビットポジションRnbで指定するビットフィールドへ連続転送する。その後RndにRnoを加え，次のデータを転送する。以上の動作をRny回繰り返す。以上の動作を一命令で実行する。この命令は，フォント用メモリの内容の画像メモリへの転送に使うと効果的である。

4.2　高級言語に適したざん(斬)新アーキテクチャ

H16ではハードウェアとソフトウェアを並行して開発し，高級言語に適した全く新規なアーキテクチャとした。

CPU内処理に比べて時間のかかるCPU～メモリ間データ転送を減らすには，レジスタ数を増すかCPU内に高速メモリを設ければよい。レジスタ数を増すと副プログラム呼出しやタスク切替えに伴うレジスタの退避・回復に時間がかかる。CPU内高速メモリは，OS(オペレーティングシステム)などでは有効に使えるが，不特定多数のプログラムでは共用が難しい。副プログラム呼出しに伴うレジスタ退避・回復の高速化方法として，呼出しの深まるごとに別レジスタを割り付ける方法があるが，これはレジスタが不足したときの割付け変更に時間がかかる。

H16では上記の問題を一挙に解決できるグローバルバンク付きリングバンク方式[1]と，それを活用するC言語コンパイラを開発した。**3**章で述べたように，H16は32ビットのレジスタ16本から成るレジスタバンクを最大16組み，合計256組みのレジスタをCPUチップに搭載している。**図3**で示したように，レジスタバンクを2組みないし16組み持つ通常モードに加えて，リングバンクモードという特別なモードを持っている。リングバンクモードに設定されると，レジスタバンクは副プログラム間で共通に使うグローバルバンク8組みと，副プログラムごとの局所データを入れるリングバンク8組みに分けられる(**表4**参照)。

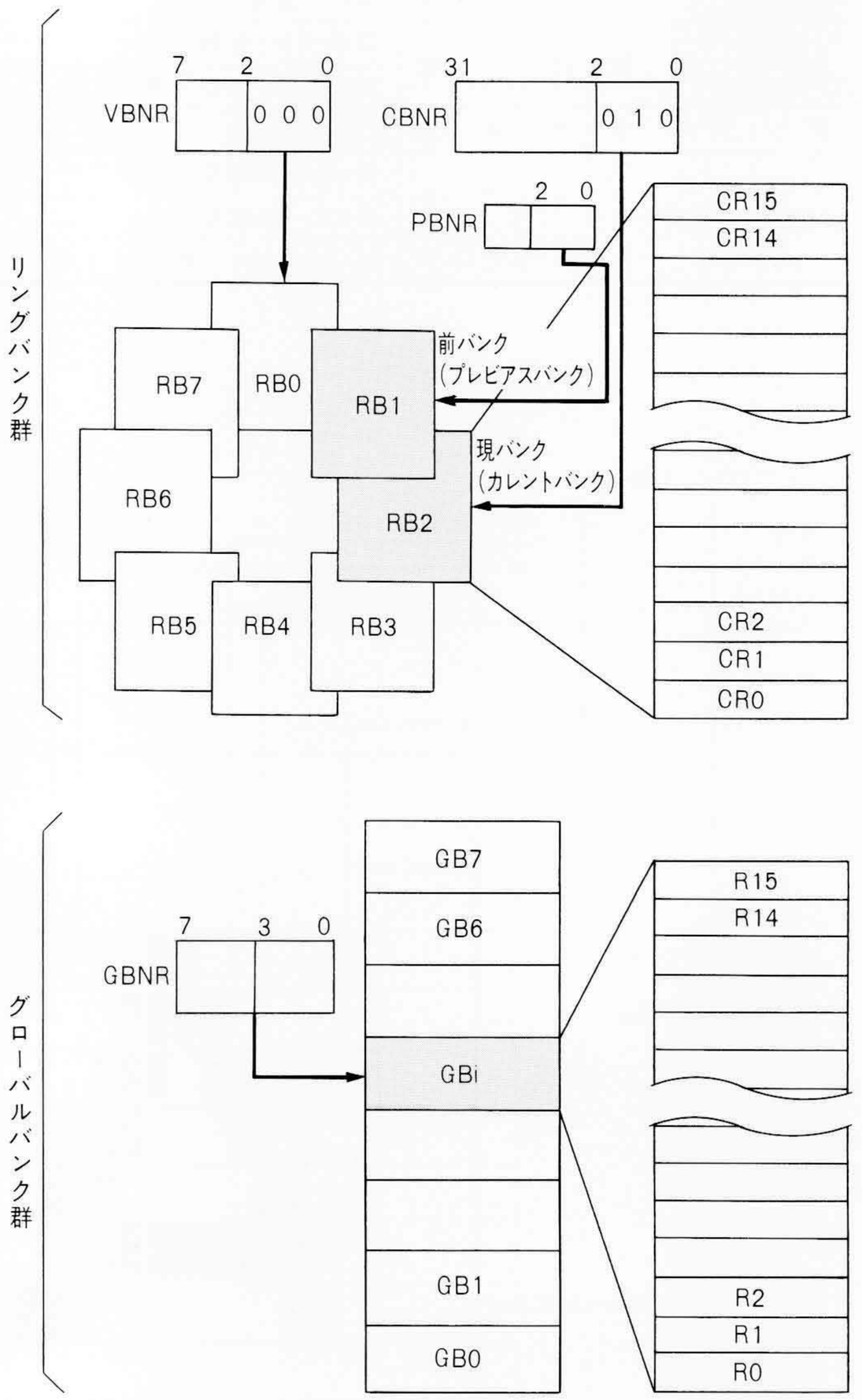
8組みのリングバンクは**図6**のように円環状に番号づけされ，副プログラムの呼出し・復帰ごとにその使用位置が前進・後退する。詳しく言うと，i番のリングバンクR_iを「現バンク」として使っている副プログラムpが副プログラムqを呼ぶと，qでは円環状番地でみて$i+1$番のリングバンクR_{i+1}を現バンクとし，R_iを「前バンク」として使う。qからpへ復帰するとpは再びR_iを現バンクとし，R_{i-1}を前バンクとして使う。前進時にリングが一巡して以前使っていたバンクにオーバーラップする場合は，そのオーバーラップしたバンクをメモリへ自動的に退避する。同様に，後退時にリングが一巡して元に戻った場合は，逆にメモリから自動的にバンクの内容を回復する。副プログラム呼出しの多重度の変動が7以内の間は，リングバンクの退避・回復を行わない。

プログラムでは，前バンク(**図6**のRB1)，現バンク(同図のRB2)，グローバルバンク(同図のGBi)の計三つのバンク，合わせて48本のレジスタを各時点で使うことができる。

プログラム全体で高頻度で使われる共通情報はグローバルバンクに入れ，一つの副プログラムで高頻度で使う情報と子へ渡す実引数は現バンクに入れる。親からの仮引数は前バンクにあり，親へ返す値は前バンクに入れる。各命令では，0～15のレジスタ番号と，それがグローバルバンク，現バンク，前バンクのいずれに含まれるかをアドレスモードによって指定する。各時点で使えるレジスタは，このように48本あるので，高頻度で使われるデータは大部分レジスタに置くことができる。並列実行されるタスクの各々には，一般にグローバルバンクを一つずつ割り当てる。高速応答を要求されるタスクは一般に処理が短いので，グローバルバンクだけを割り当

表4　H16レジスタバンクモード　　図4に示す2組みないし16組みのレジスタバンクを持つ通常モードに加えて，8組みのリングバンクと8組みのグローバルバンクを持つリングバンクモードがある。

レジスタモード	通常モード				リングバンクモード
	2バンク	4バンク	8バンク	16バンク	
グローバルバンク数	2	4	8	16	8
リングバンク数	0	0	0	0	8
CPU内高速メモリ	896バイト	768バイト	512バイト	0	0



注：略語説明　CBNR(Current Bank Number Register)
PBNR(Previous Bank Number Register)
VBNR(Valid Bank Number Register)
GBNR(Global Bank Number Register)

図6　リングバンクモード使用時のレジスタバンクの構成　　グローバルバンクの使用位置は，GBNRで指示される。副プログラムの呼出し・復帰に伴い，リングバンクの使用位置ポインタCBNRとPBNRが前進・後退し，それらがVBNRを越したときにだけレジスタの自動退避・回復が行われる。

て，長大な処理を要求するタスクにはグローバルバンクとリングバンクを割り当てるならば，タスク切替え時にレジスタバンクの退避・回復を必要としない。

H16では，上記機構によって，多くの処理をレジスタだけを使って処理でき，副プログラム呼出しもタスク切替えも極めて高速に実行できる。

4.3　新アーキテクチャ向きCコンパイラ

H16の開発に当たっては，LSIと並行して，高級言語“C”のコンパイラを開発した。

H16Cコンパイラは，C言語のソースプログラムを高速のオブジェクトプログラムへ変換するため，多数のレジスタを最大限に利用する。コンパイラでは，どんな値がどこで算出され，どこで使われるかを解析し，広域的にみて最適なレジスタ割付けをする。そこでは，Cのプログラム構成単位としての関数全体の処理の流れを解析し，個々の変数や式，部分式の値の算出位置と参照位置，値が不変に保たれる「生存区間」を調べ，これらの生存区間の各々に仮想的なレジスタを割り付ける。生存区間の重なり合わない仮想レジスタは，同一の実レジスタに割り付けできる。許容される実レジスタの本数の範囲内で重ね合わせできれば問題ないが，重ね合わせて実レジスタが不足すれば，使用頻度の低い仮想レジスタをメモリに割り付ける。どれをメモリに割り付けるかは，対応する変数や式，部分式の使われ方と使用頻度，生存区間の長さなどから算出するプロフィット関数によって決定する。レジスタ割付けには，引数や関数値のように特定レジスタに割り付けるといった制約がある。これらの制約もすべてプロフィット関数の計算因子として与える。

このようにして，数個の文から成る小区間ではなく，関数全体という大きい単位でレジスタ割付けを最適化し，メモリアクセスを極小化することによって実行速度を上げる。

4.4　アーキテクチャとコンパイラの評価

本アーキテクチャとそのコンパイラの効果を調べるため，H16で，全プログラムでレジスタバンクを一つだけ使う場合と，リングバンクモード(グローバルバンク8組み，リングバンク8組み)を使う場合とを比較した。前者は単一バンクを持つ従来のマイクロコンピュータの場合に類似している。**表5**にその結果を示す。後者の多バンクが前者よりも副プログラム呼出しで約2.5倍，タスク切替えで約9倍高速であり，ベンチマークテストでも大幅に速い場合が多い。

実際のプログラムの特性を調べた例では，副プログラム呼出しの段数の変動が7を超えるのは数十回に一度であり，タスク切替えの回数は副プログラム呼出し回数の$\frac{1}{50}$程度であった。また，関数で使われる局所変数の数は多くなく，H16ではその大部分をレジスタに割り付けることができる。これらの点からみて，H16で開発したアーキテクチャとコンパイラ方式は，高性能化に大きい効果を発揮すると言えよう。

5　開発サポートツール

H16を使ったシステムを開発するために用意された開発サポートツールを**表6**に，開発ツールのハードウェア構成例を**図7**に示す。ホストコンピュータ上で，クロスアセンブラやクロスCコンパイラを使って開発されたプログラムは，まずクロスシミュレータでデバッグされ，次にRS-232Cインタフェースを通じてリアルタイムエミュレータ(H16ASE)へロードされ，実機でデバッグされる。なお，アセンブラなどのクロスソフトは，いずれもそれ自身がC言語で記述されており，各種ホストコンピュータへの移植性の向上が考慮されている。

表5　レジスタバンク群の使用効果　　グローバルバンクを8組み，リングバンクを8組み使う多バンク形態では，レジスタバンクを1組みだけ使う単一バンク形態に比べて，副プログラム呼出しやタスク切替えなどが著しく高速化される。

比較項目		多バンク/単一バンク処理時間比
副プログラム呼出し		0.41
タスク切替え		0.11
ベンチマークプログラム	探索	0.58
	ソート	0.39
	素数計算	0.88

表6　H16開発サポートツール一覧表　　システム開発及びプログラム開発を効率よく行うために，数々の開発ツールを用意している。

分類	サポートツール
クロスソフトウェア	アセンブラ
	リンケージエディタ
	ライブラリアン
	Cコンパイラ シミュレータ・デバッガ
レジデントソフトウェア	リアルタイムOS
ハードウェア	ASE(リアルタイムエミュレータ)
	シングルボードコンピュータ

注：略語説明　ASE(Adaptive System Evaluator)

6　応　用

以上述べてきたように，H16はレーザービームプリンタ，日本語ワードプロセッサ，FA(Factory Automation)，ロボット，端末など大量のデータを高速処理する必要のある分野への応用に適している。

7　結　言

以上，機器内蔵形のコントローラ応用分野をねらって開発した高集積形マイクロコンピュータH16について，アーキテクチャの特長を中心に述べた。特に，マルチレジスタバンク構成をとることによって，リアルタイム応答性の改善と高級言語の高速実行を実現した。また，システム構成に必要な周辺機能を内蔵することで，16ビットマイクロプロセッサを使った高性能な応用システムが機器内蔵可能な基板サイズで実現できるようになった。このように，H16は機器内蔵形マイクロコンピュータの性能・機能を飛躍的に改善し，応用機器の高度化に大きく寄与するものと考える。

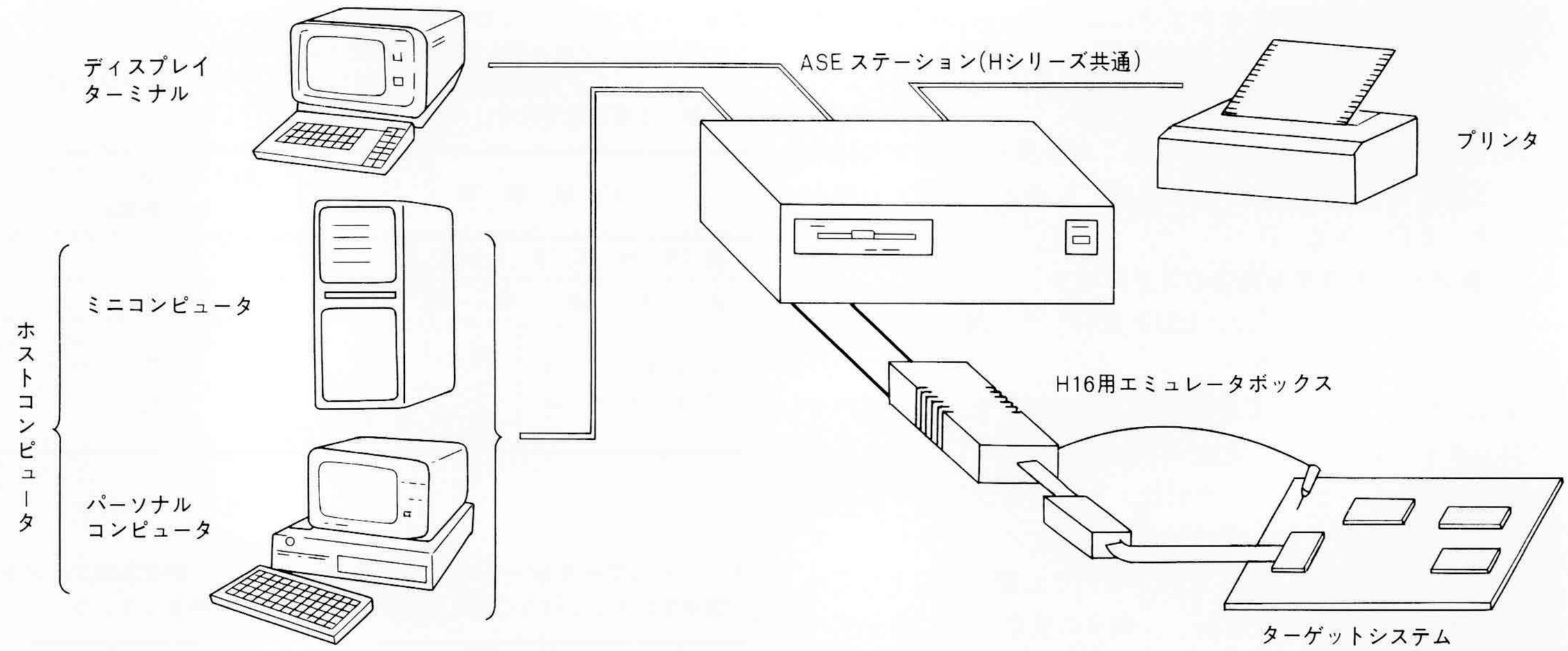


図7 H16開発サポートシステムのハードウェア構成例 Hシリーズ共通のASEステーションを中心に，H16専用のエミュレータボックス，ミニコンピュータ，パーソナルコンピュータ等のホストコンピュータなどで構成されている。

今後，各応用分野に最適な周辺機能の集積化，アドレス空間の拡張，高速化などの要求が強くなるものと考えられる。今回開発したH16CPUを核に，周辺機能のバリエーションを変えたファミリー製品群の開発を行い，これらの要求にこたえてゆきたいと考えている。

参考文献

1) Maejima, H. et al.：A 16-bit microprocessor with multi-register bank architecture, Proc. of FJCC, pp.1014-1019 (1986-11)
2) Patterson, D.A. et al.：A VLSI RISC, COMPUTER, September 1982, pp.8～45(1982-9)

論文抄録

アドレスラッチ機能内蔵形 高速4kビットバイポーラRAM

日立製作所 南部博昭・山口邦彦・他5名

電子情報通信学会論文誌 J70C－1，1～10（昭62-1）

バイポーラRAM (Random Access Memory)は，大形計算機の性能を左右するキー デバイスであるため，その高速化に対する要求は常に強い。

この要求にこたえるため，既に筆者らは，(1) RAMチップにラッチ機能を付加し，メモリシステムを実効的に高速化するアドレスラッチ機能内蔵技術，(2) ダミーセルを用いて高速化する新センス回路，及び(3) ワード線を高速に充電するダーリントン駆動回路，などの高速化回路技術を提案した。

本論文は，上記(1)，(2)のアクセス時間の高速化効果，及び(3)がメモリセルの情報保持特性に与える影響について，定量的に解析した結果の報告である。

まず，アドレスラッチ機能内蔵技術では，RAMチップ自身にラッチ機能を付加することによって，従来論理チップで構成していたラッチ回路が不要となるため，システム的にみて実効的に1 ns(20%)高速化できることを明らかにした。また，新センス回路では，ワード線の駆動振幅を20%低減できるため，0.5ns(10%)高速化できることを明らかにした。また，ダーリントン回路については，ワード線電位の立上りが極めて急しゅん(峻)になるため，メモリセルの情報保持特性が損なわれる可能性がある。そこで，この回路がセルの情報保持特性に与える影響を，回路シミュレーションによって検討した。その結果，ダーリントン回路を使用した場合のメモリセルの情報保持特性は，セルの負荷容量によらず従来と同等であることを示した。

更に，上記議論の妥当性を確認するために，高集積用1.5μmU溝アイソレーションプロセスで4kビットRAMを試作し，評価を行った。得られた性能は，アクセス時間3.5 ns(ラッチ機能内蔵により，システム的にみて従来のRAMのアクセス時間2.5nsに相当)，チップ面積12.7mm^2，消費電力1Wであり，上記回路技術で，システム的にみて30%の高速化を達成できることを実験的に明らかにした。なお，従来の金属探針法で観測不可能であったメモリセル内の電位波形を，時間分解能500psの電子ビームテスタで初めて観測し，ダーリントンワード線駆動回路を用いても，予想どおりメモリセルが安定に動作することを実証した。