マイクロコンピュータ応用機器開発用汎用援助システムについて

浅田勝彦　　寺田浩詔
（大阪大学工学部）

1　はじめに

マイクロコンピュータの応用は，あらゆる産業分野に広がり，ますます多様化し，マイクロコンピュータの応用技術者の数は大幅に増加している。またマイクロプロセッサ／マイクロコンピュータの利用形態も各種の方法がとられている。

マイクロプロセッサとその周辺ＬＳＩは，現在すでにかなり高機能化し，複雑化している。具体的にはかつてのミニコンピュータ以上で，いわゆる小形・中形（ファミリィ）機種と呼ばれるほどの性能をもつマイクロプロセッサが出現し，また1つのシリコンチップの上にコンピュータの全機能（ＣＰＵ＋ＲＡＭ＋ＲＯＭ＋Ｉ/Ｏポート）を集積化したワンチップ・マイクロコンピュータも，並列処理幅，メモリサイズも大きくなった高機能なものが発表されている。今後も更に高機能化は進むものと思われる。

このような背景から，マイクロコンピュータ応用製品の開発を効率よく行なうため，「開発援助システム」は重視されており，すでに現在多くの開発援助システムが発表され，市販されている。

開発援助システムはこれまでどちらかというと開発設計援助システムという，狭義の意味で考えられていた。しかし，先にも述べたように，マイクロプロセッサとその周辺ＬＳＩは高機能化し，複雑化している現状から開発援助システムは，設計開発技術者の設計・調整の個別的な援助にとどまらず，システムの企画・設計・製造・運用／保守にまで活用できるものが望まれるようになってくる。このような統合的な開発援助システムは，その応用製品にたずさわるすべての要員が利用できる総合的かつ一貫性あるデータベースとしての性格をもつものでなければならない。

本稿では一般的なマイクロコンピュータ応用製品の開発の流れで，各段階における開発援助システムの役割を述べ，つづいて現在市場にある開発設計援助システムを分類し，それらの機能・特徴を解説する。最後に我々の研究室で，検討を行なっているマルチプロセッサシステムの一応用として考えているマルチユーザ向きの汎用開発援助システムについて紹介する。

2．マイクロコンピュータの利用形態と開発法

マイクロコンピュータの利用形態を分類すると，ほぼ次のように分類できる。[1)]

(1)　（パーソナル／スタンドアロン）コンピュータ形式
(2)　汎用ワンボードコンピュータ形式
(3)　専用ボード形式（ＣＰＵ組込み）
(4)　ワンチップ・コンピュータ形式

一般的に，先にあげたものほどハードウエアの機能は保証されており，したがってソフトウエア開発の比重は，ハードウエア開発に比べて相対的に高くなる傾向にある。また，後にあげた形式の応用製品ほど，量産される数は一般に大きくなる。

ここであげたコンピュータ形式の利用法とは，いわゆるパーソナル・コンピュ

ータあるいはミニコンピュータ形式のマイクロコンピュータをシステム内に組み込む形式の利用法を想定している。この形式では，ごく特別なハードウエアを付加することを除けば，そのハードウエア技術はすでに保証されている。したがってシステムにコンピュータを組み込む開発過程は，もっぱらソフトウエアの開発過程に終始することになる。この形式のコンピュータは一般にＴＴＹあるいはＣＲＴディスプレー等の標準入出力機器を持っており，これを用いてそれ自身でソフトウエア開発が行なえる。この機能を利用して，同種のマイクロプロセッサを利用した他の形態の応用システムのソフトウエア開発援助システムとして用いられることもある。この形態で応用されるコンピュータは，ソフトウエア開発の援助機能，直接接続・制御可能な周辺機器，それにユーティリティプログラム等のソフトウエア的な処理機能とで評価される。

これに対し，もう一方の極端であるワンチップ・マイクロコンピュータ形式の応用では，コンピュータのもっとも基本的なアーキテクチャを取扱わねばならず，小型ではあるが，コンピュータシステム設計そのものと同様な開発過程が要求される。原則としてこの形式の応用では，限定された入出力ポート数，チップ上のＲＯＭおよびＲＡＭ容量の制限内で，必要とする機能を実現しなければならない。このような制約のもとに，ハードウエアとソフトウエアの機能分担を考え，ハードウエア設計，ソフトウエア開発を行なうのは容易な作業でなく，高度のシステム的技術が要求される。さらに最近のワンチップ・マイクロコンピュータでは，適応能力を高めるため，アーキテクチャの一部にＰＬＡを組み込み，可変にしている場合もあるが，開発過程はさらに複雑になる。

ワンチップ・マイクロコンピュータはコンピュータの全機能がチップ上に組み込まれているので，実用される形そのままでの開発は不可能であり，通常プロトタイプという試作システムを用いて開発が行なわれる。プロトタイプ設計用に，ほとんどのワンチップ・マイクロコンピュータは，評価用（エバリュエーション）チップと呼ばれる開発援助用のチップをもっている。これは主としてチップ上のプログラムメモリであるＲＯＭを外部に付加できるようハードウエアを改造したものであり，ソフトウエア開発援助の機能はほとんど持っていない。

汎用ワンボード・コンピュータ形式ではいわゆるワンボード・コンピュータと一般に対象とするシステムに適応したインタフェースボードを組合せてシステムを構成する。ワンボード・コンピュータは，８ないし16ビットマイクロプロセッサ，メモリ（ＲＡＭ，ＲＯＭ），標準周辺機器とのインタフェース（ＲＳ－２３２Ｃ，カーレントループ）および標準的な外部バス系（Ｓ－100バス，マルチバス等）と接続用のバス・インタフェースを備えている。

この形式の応用では，ワンボード・コンピュータおよびインタフェース・ボードは，いずれもそのハードウエア機能が保証されているものと考えてよく，(1)の形式の場合と同様に応用プログラムについてのソフトウエア開発が中心となる。

専用ボード形式の利用形態は，対象システム向きにボードを設計し，利用する場合をさし，この形態は今後ますます拡大していくように思われる。この応用形式の場合には，ハードウエア開発とソフトウエア開発とを並行かつ総合的に行なわねばならず，ワンチップ・マイクロコンピュータの開発と同様な問題が生じる。

現在ＬＳＩ技術の進歩により，プロセッサチップの高性能化，メモリチップの大容量化，周辺制御ＬＳＩの高機能化が進んでおり，チップ間接続も両立性が保

たれるよう考慮され，またファミリィチップとして統一された周辺制御ＬＳＩが多く用意され，さらに単一電源で動作するようになってきている。このことから比較的簡単に高機能のボードをハードウエア設計できるようになってきた。しかしボード内部の複雑さは大きくなっており，ハードウエアの細かな調整をしながらソフトウエア開発を進める作業は非常に煩らわしいものになっている。このためシステム設計時のハードウエア機能とソフトウエア機能との分担の決定はさらに慎重に行なう必要がある。

3．開発の流れと開発援助システム

マイクロプロセッサ応用システムの開発課程は一般的に図１に示すような手順をとる。最初にも述べたように開発援助システムは，同図の全課程にわたって支援すべきものであるが，このような統合的な開発援助システムは現在のところみあたらず，現在一般にはⅢ～Ⅴの課程を支援する狭義の開発援助システムが使われている。

統合的な開発援助システムが，狭義の開発設計援助システムのもつ機能の他に持つべき機能は，議論のあるところであるが，次のような機能を考えることができる。

(1)開発に必要なデータベースを持つこと。

(2)開発／製造工程の管理を行なう機能をもつこと。

このような統合的な開発援助システムは相当大規模なものになる。しかしマイクロコンピュータ応用システム開発において，通常Ⅲ～Ⅴの課程が反復して行なわれるため開発の期間の大半を占め，その間開発援助システムを接続したままで行なわれることが多い。このことから統合的な開発援助システムを個々の開発技術者が持つということは贅沢なことになり，その維持も大変である。したがって開発段階で，応用システムの実時間動作のモニタと解析を行なう機能

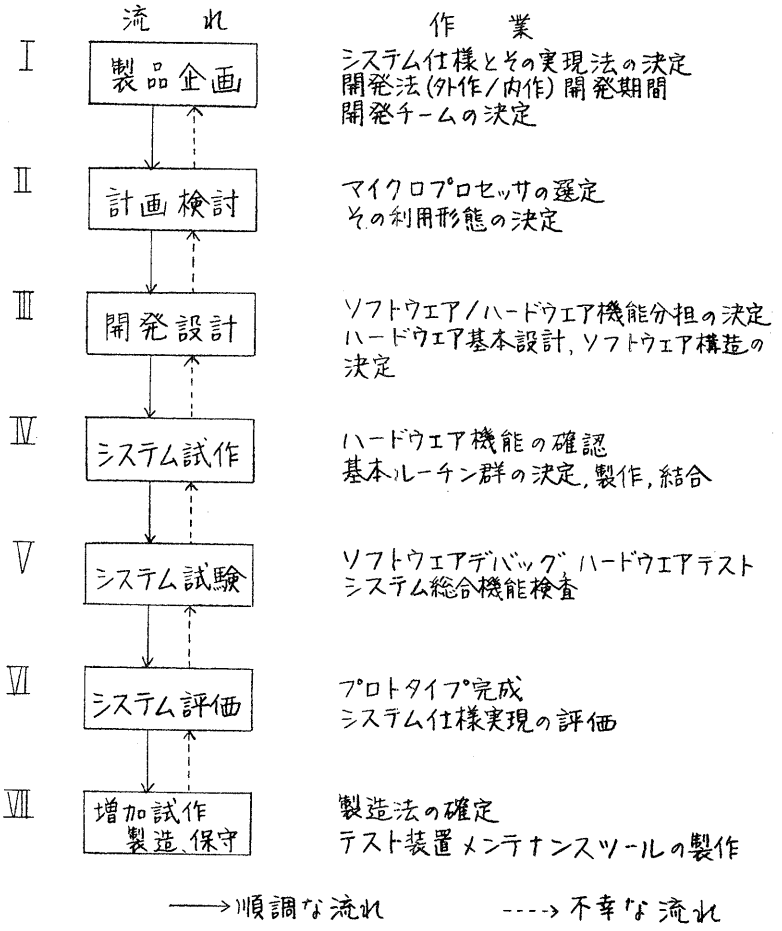

図１　マイクロプロセッサ応用システムの開発の流れ

を持つ端末を使って開発を行ない，データベースへのアクセス，工程管理，応用プログラムのコンパイル／アセンブル等の高度の処理は，ホストの高性能コンピュータで行なうＴＳＳ形の開発援助システムの形態が多くなると思われる。現在すでにこのような形態をとる開発援助システムは２，３発表されており，これに上記のような機能が付加されると，本格的な開発援助システムが得られるようになる。

マイクロプロセッサ応用システムの開発課程でⅢ～Ⅴの段階が開発期間の大半を占め，この間の開発を支援する狭義の開発援助システムが多く使われる。したがってこれからは狭義の開発援助システムについて述べる。

この課程の開発の流れを詳しく示すと図２のようになる。ハードウエアとソフトウエアが並行して開発が進められ，最後に総合し，動作が検証される。この課程では，ワンチップ・マイクロコンピュータ形式，あるいは専用ボード形式のものではプロトタイプという最終製品とは異なる試作システム上で開発が進められる。すなわちワンチップ・マイクロコンピュータ形式では，評価用チップあるいはＣＰＵエミュレータを利用して開発用のプロトタイプシステムを作るが，これは実際の製品より形状的に大きなものになる。専用ボード形式のものでは，開発に伴なって生ずる手直し，あるいは開発の初期からプロトタイプシステム上で行なえることが要求されるため，ワイアラップ等の手配線で試作したものが利用されることが多い。

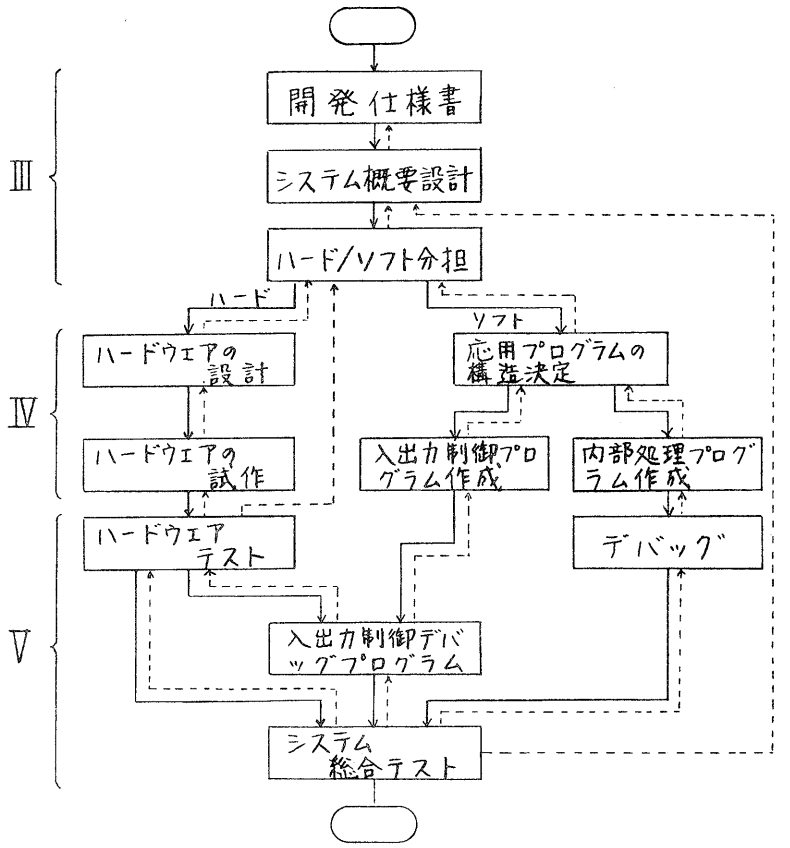

図２．ハードウエア・ソフトウエア調整の流れ

このようなプロトタイプの上で開発が進められるのがマイクロコンピュータ応用製品開発の特長になるが，これとは別に最終製品の設計も必要になり，図２に示した流れのプロトタイプ開発に並行して，最終製品設計が行なわれる。システム仕様を満足するものをプロトタイプ上で開発できると，そのハードウエア／ソフトウエアを最終製品設計に移植して，製品開発が完了することになる。

ソフトウエア開発に於いて，プログラムの作成とそのデバッグが開発期間で大

きな部分を占める。試行と錯誤の繰返しで進められるこの期間の短縮は，この過程のサイクル数とサイクル時間を減少させることによる。これにはプログラマの資質がもちろん大きく作用し，またどのような言語（アセンブラ・コンパイラ）を用いるかによるが，サイクル時間は，機能の高い開発援助システムを活用することによって短縮することが可能である。

ソフトウエア開発に，マイクロプロセッサ・ベースのスタンドアロン・システムを用いることもあるが，強力なホストプロセッサを持つクロスシステムを用いた方が有利である。この理由は，一般にホスト・システム側にある強力なソフトウエアと高速の処理能力が利用できることが第一にあげられる。それと同時に強力な入出力機器，大容量のファイル媒体の利用によって，個々の作業間の手待ち時間を非常に短縮できることも大きな理由になっている。

ハードウエア開発において，設計段階では論理設計援助プログラム，試作段階ではプリント基板作成援助プログラム等が考えられる。この段階では現在ＬＳＩ自体の機能が向上し，比較的簡単にハードウエア設計ができるようになり，また，プリント基板は外部機関に依頼して作成することが多いため，開発援助システムはあまり必要としない。

ハードウエアテストの段階では，開発援助システムに負うところが多く，後述のように各種の動作解析用の開発援助システムが現在市販されている。マイクロプロセッサ応用システムの開発はいつもハードウエアとソフトウエアが絡み合っており，ハードウエアテストもテストプログラムというソフトウエアの援助で行なわれる。このとき動作を確認しようとするハードウエアを不確実な（バグを含むかもしれない）ソフトウエアを用いてテストするという矛盾した作業を行なわねばならず，開発援助システムが必要となるわけである。ハードウエア動作解析のためには，多点のデータを相互の正確な時間係を含めて並列に観測し，解釈しなければならず，それらのデータは系列として意味を持つものであるから，集録すべきデータは非常に大量になる。したがってこれらのデータを見やすく整理して表示する機能は重要な援助手法となる。

入出力プログラムの作成とデバッグは，ソフトウエア開発に含めて考えることもできるが，この部分がハードウエアとソフトウエアの接点となり，開発に苦労するところである。またソフトウエア的にも，ハードウエアとの接点をできるだけ局限化し，プログラム上も局在化させる方針でシステム設計した方が，開発を効率よく行なうことができるので，内部処理プログラムと分けて開発の流れに示している。この部分はハードウエアテストが終了してから本格的なデバッグに入ることができる。

これまで述べたようにマイクロコンピュータ応用システムの開発作業はそのほとんどが試行錯誤の過程であり，単調な作業の繰返しで進められる。このような環境では，開発技術者に焦りと疲労を感じさせることになる。このことから開発援助システムは，充分人間工学的に配慮されたものが要求される。

4．開発援助システムの現状

ここでは現在市販され利用できるマイクロプロセッサ応用システムの開発援助システムを開発における役割によって分類し，その特徴と要求される性能・機能について述べる。現在非常に多くの援助システムがあり，ここで行なう分類には

入らないものもあるが，基本的には次のような機能を持つ援助システムがある。

4．1　プログラム開発援助システム

これには，先にも述べたように，各種の形態のものが利用されている。すなわち通信回線利用によって大型センタの資源を利用する方法，インハウスシステムのクロスソフトウェア機能を利用する方法，マイクロプロセッサベースのスタンドアロン／パーソナルコンピュータを利用する方法，汎用ボードに標準周辺機器を付加したシステムを利用する方法，等がある。これらは，対象とするプロセッサの種類，開発すべきソフトウェア量，言語の選択など，種々の条件によって選ぶことができる。これらのシステムは次のような機能によってソフトウェア開発を支援している。

- プログラム編集機能（テキスト・エディタ）
- 言語処理機能（アセンブラ・コンパイラ）
- プログラム結合援助機能（リンカー，ロケータ）
- 動作解析，プログラム修正機能（シミュレータ，トレーサ，デバッガ，エミュレータ，逆アセンブラ，モニタ）

最近は，これらのソフトウェア開発援助プログラムでユーザ所有のシステムへの移植が容易に行なえるものが提供されている。すなわち，ユーザは所有のシステム個有の入出力機器インタフェースに合わせて物理的な入出力制御プログラムを作り，提供される管理プログラムと結合すれば，提供される開発援助プログラムすべてが利用できるようになる。これで開発できるマイクロプロセッサの種類は限られるが，かなりの機能をもつ開発援助プログラムが提供されるので，入出力制御プログラムだけ用意し，最初からそれを使えることをうたった開発援助システムも多く発表されている。

開発されたプログラムをＰＲＯＭに書き込むためにＰＲＯＭ書き込み器が付加されるものもある。

4．2　ロジックアナライザ

ハードウエアの動作解析では，多点のデータを相互の正確な時間関係を含めて解釈する要求が強く，ロジックアナライザはこの目的に使われる。

通常８～３２チャネルの信号を同時に観測することができる。各信号は内部クロック，あるいは解析システムから得られる外部クロックによって最高数10MHzの速度でサンプルされ，ロジックアナライザ内部のメモリに取込まれる。信号は連続して数百ワード取込み可能であり，取込みクロック，サンプル開始，終了のトリガ条件，データの選択的取込みの条件等が各種選択できるようになっている。このようにして取込んだ大量のデータを見やすくするためＣＲＴ画面上にタイムチャートで波形表示したり，２進，８進，16進等で表現した文字列で表示される。

ロジックアナライザでは各信号はシンクロスコープのように連続して取扱わず，クロックでサンプルした時点の信号のみが取込まれる。したがって原理的にサンプルクロックの間にある細かな信号変化は観測できないことになるが，そのような信号変化は一般にグリッチと呼ばれ，誤動作を起こす要因にもなるので，このような信号も１サンプルデータとして取込むモードを持つものもある。

ロジックアナライザは，対象システムの周辺から受動的にその動作を実時間的に解析する，いわゆるリアルタイムトレースを得意とする援助システムで，対象システムの内部状態（メモリ，レジスタ，I/Oポートの内容）を直接調べたり，

変更することはできない。

ロジックアナライザは，これまで述べたことからもわかるように汎用の開発援助システムで，広くデジタル制御システムの開発に利用される。しかし個々のマイクロプロセッサ用に専用のアダプタを付加できるものも発表されている。これを用いると対象システムへの接続が容易になり，また逆アセンブラも組込まれていて，トレースデータを，命令略号を使ってＣＲＴ画面上に表示し，解析を容易にしている。

４．３　マイクロプロセッサアナライザ

専用ボード形式の利用形態のシステムを開発する場合，ハードウエアをできるだけ簡略にすませる設計がとられるため，開発援助の機構は殆んど付加されない。このようなシステム開発のために，ミニコンピュータのコンソールパネルのような機能を，ＣＰＵチップのソケットまたはシステムバスに接続して，実現したものをマイクロプロセッサアナライザと呼んでいる。

これは通常，小形で持ち運びが便利なように作られ，開発だけでなく，保守にも用いられる。これの持つ機能としては，マイクロプロセッサの実行制御と，ＣＰＵレジスタ，メモリ，Ｉ／Ｏポートの内容の読み出しと，その内容を変更できる機能をもつが，ロジックアナライザのようなトレース能力を持たないものが多い。当然のことながら特定のマイクロプロセッサに対してのみ適用されるものになる。実行制御も，単に起動・停止だけでなく，ブレークポイントを設定して，特定の個所で実行停止できる機能を持つものもあるが，その機能はあまり高くない。

４．４　インサーキット・エミュレータ（ＩＣＥ）

ＩＣＥは，簡単に言うと，ロジックアナライザとマイクロプロセッサアナライザの両方の機能を合せ持つものである。したがってこれの機能の実現にはかなり高度の処理が要求されるため，これ自身もマイクロプロセッサを利用して構成されるか，また，４．１で述べたソフトウエア開発援助システムの一つの付加機構として組込まれて実現されている。特に後者は，ソフトウエア開発援助システムと統合されるため，これだけで，ソフトウエア，ハードウエアの開発が行なえる特長を持つことになる。

ＩＣＥは，ハードウエアの調整の初期の段階から，ソフトウエアとハードウエアを総合した調整段階まで，有効に開発を援助してくれる。すなわちＩＣＥは，対象とするターゲット系のＣＰＵのソケットから接続されるから，ＣＰＵ周辺の回路の動作から調整を始め，メモリ，Ｉ／Ｏポート等のハードウエア調整を順次行ない，ターゲットシステム上の確実に動作するハードウエアを広げていく。すべてのハードウエア調整が終ると，リアルタイムトレース機能を利用して入出力プログラム，さらには内部処理プログラムのデバッグにも活用できる。したがってＩＣＥがソフトウエア開発援助システムと，非常に効率よく開発できることが類推できよう。

ＩＣＥもマイクロプロセッサアナライザと同様に，通常支援するプロセッサの種類は一種に限定される。

４．６　その他の開発援助ツール

ここでは，システムにまで行かないが，開発に必要な測定器，簡便な援助ツールをとりあげることにする。

まず才一にあげられるのがオッシロスコープ（シンクロスコープ）である。こ

れはあらゆる電子制御システムの開発に用いられる汎用の測定器であるが，先に述べた(3),(4)の形態での応用のようにハードウエアからシステムから設計を行なわねばならないときには，不可決な援助システムである。

ロジックプローブも簡単であるが調整用に便利なツールである。これは通常ペン形をしており，測定値のHigh／lowをランプ／周波数(音)で知らせるものである。繰返し信号，単発パルスの現象を観測しやすくするため，周波数を下げたり，パルス幅を広くして表示してくれるので，オッシロスコープでは取らまえ難い単発現象に効果がある。ロジックプローブは，回路上に無理にパルスを送り，回路の状態を変えることを行なうロジックパルサと組合せて用いられることもある。

開発援助システムが乏しい状況では，プログラムを直接ＰＲＯＭに書き込み，調整する場合があるが，修正個所が発見できると，消去したＰＲＯＭに新しいプログラムを書くという効率の悪い開発を進めなければならない。このＰＲＯＭをＲＡＭ化し，ＰＲＯＭのソケットへ接続し，そのＲＡＭ上でデバッグできるようにし，ＰＲＯＭ書き込み機を組合せたＰＲＯＭ置換器も出ている。

5．開発中のマルチユーザ向汎用開発援助システム

我々の研究室では，ＭＩＭＤ形の処理を共有メモリを介して実行するマルチプロセッサアーキテクチャの一形式としてPASSIVE MEMORYLESS ARCHITECTURE（ＰＭＡ）を提案し，その試作を行なっている。ＰＭＡは並列処理要求がタスク以下のレベルで陽に示される応用に対して有効で，一般に多端子の実時間制御システムへの応用が考えられる。[2) 3)]

ＰＭＡシステムでは，各々独立に処理要求を発生するサブシステム（ＳＳ）群，これらの処理要求を受理して並列実行するプロセッサ（ＰＵ）群，さらにこれらに共通なメモリモジュール(M)群が，ＰＭＡバスといわれる共通バスに結合される構成をとる。これは従来から提案されているＭＩＭＤ形マルチプロセッサと同様な構造をとるが，その動作モードに特徴をもっている。すなわち，ＰＭＡにおけるＰＵの特徴は，ＰＵがシステム内のあらゆる資源の分配に関して受動的(passive)であり，同時にサブシステムに関する動的な情報は共有メモリ内に置き，ＰＵ内には保持されない(Memoryless)ことである。この特徴により，システム内での資源分配の制御をＰＵの機能から取り除き，ソフトウエア的手法によるＰＵ間協調のためのオーバヘッドの軽減をはかり，マルチプロセッサ化に伴うソフトウエア構造の複雑化が避けられる。ＰＭＡはデータフローマシンとなじむところがあり，データフローマシン実現のためにソフトウエアの構造について検討を進めている。

図3に現在試作中のマルチユーザ／マルチプロセッサ向の開発援助システムの構成を示す。本システムはマルチプロセッサの開発を主たる対象としているために，協調して動作する複数のＩＣＥ機能およびマルチプロセッサシステム内の複数の信号経路（たとえばバス系）にアクセスするための複数の

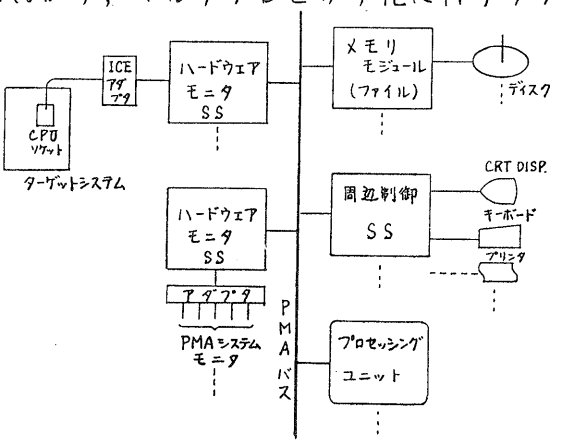

図３．ＰＭＡによる開発援助システム

アダプタを接続しうる構造をとり，このシステム自体が前述のＰＭＡ形式の動作モードをもつマルチプロセッサシステムとなっている。

本システムでは，ハードウエアモニタで収集されたデータが，簡単な前処理をうけたのち，データパケットとして後段の処理に必要な処理機能（プロセッシングユニット）を起動する。この際，プロセッシングユニットは，もし必要ならば，関連するデータを他のモジュール（メモリ，プロセッシングユニット，ハードウエアモニタ，周辺制御ユニット）等から，受動的に受けとることができる。処理済のデータは，つねに受動的にではあるが，他のモジュールにデータパケットとして送出可能である。

以上のような動作モードをとるために，システム内のモジュールは互に独立性が極めて高く，このシステムはまた複数のユーザが見かけ上独立な複数の開発援助システムとして利用する場合にも，複雑な管理ソフトウエアを必要としない点に大きな特徴がある。

本システム上で動作するソフトウエアシステムとしては，汎用アセンブラ（ＣＲＡＭＢＯ）とこれに対応する汎用逆アセンブラ（ＵＤＩＳ）がすでに準備されており，市販のほとんどのマイクロプロセッサを取扱える。現在は，ソフトウエアシステムをマルチプロセッサシステムを対象とするものに拡張中である。[4) 5)]

図３のシステムで，汎用のＩＣＥとなりあるいはＰＭＡシステム監視を行なう，ハードウエアモニタの構造を図４に示す。このシステムは先に開発したヘテロマルチプロセッサシステムのものを発展させたものである。これは汎用化をめざしており，個々のプロセッサに対し，簡単な回路で構成できるＩＣＥアダプタを付加して適用させることができる。

ハードウエアモニタは図示したようにトレース部，エミュレーション部，マップドメモリ部からなる。トレース部は，ターゲットシステムのリアルタイムトレースを行なう機能をもち，40ビット幅（アドレス16，データ16，コントロール８ビット）のデータを２Ｋワード最高８ＭＨｚのサンプル速度でトレースでき，16ビットプロセッサまで解析できる。この制御部では，トレー

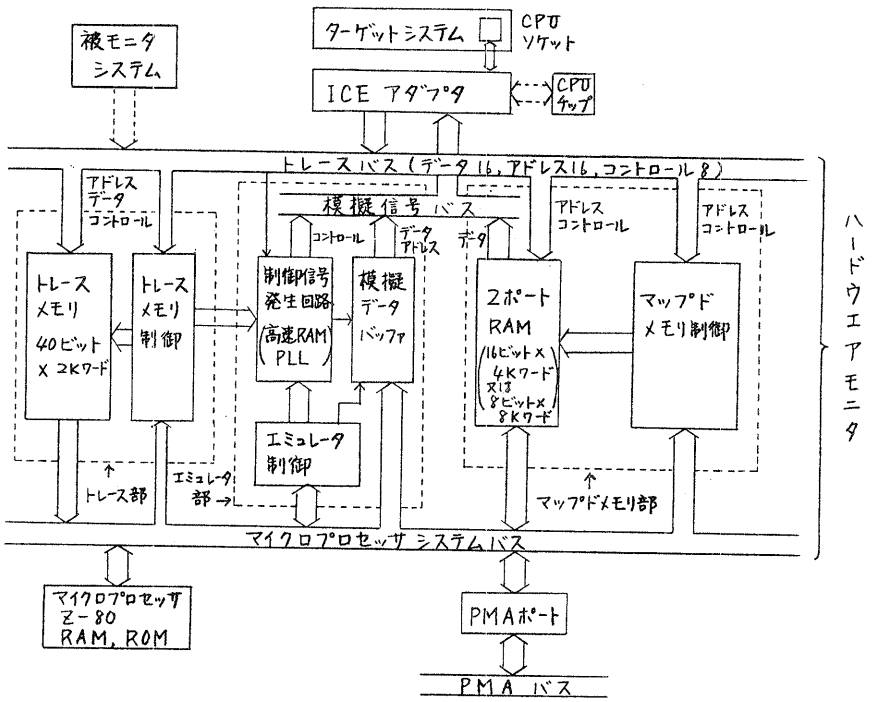

図５．ハードウエアモニタのブロック図

スの起動，終了の条件が設定できる。その条件として，トレースデータのパターン，サンプル数，時間等が組合せて設定できる。またここには汎用カウンタを持ち，特定事象の計数，事象間の時間測定に利用できる。

エミュレーション部は，ターゲットシステム制御用の制御信号，データを発生する機能を持ち，ターゲットシステムのハードウエア動作解析を効果的に進めることができ，故障診断としても有効になる。制御信号発生部は高速のＲＡＭ（８ビットＸ64ワード）で構成され，それを高速に連続的に読み出すことによって制御信号を発生する方法をとり，非常に柔軟性の高いものになる。このとき模擬データバッファは，ターゲットシステムのデータ／アドレスバスにその内容を制御信号と同期して送出するのに使われる。ここでターゲットシステムの実行制御も行なう。

マップドメモリ部はターゲットシステムのメモリの機能を代行するものである。このメモリは16ビットＸ４Ｋワードあるいは８ビットＸ８Ｋワードのサイズの２ポートＲＡＭで構成され，ターゲットシステムのメモリアドレス空間におけるこのメモリの割当ては制御部で設定される。このメモリをターゲットシステムのＰＲＯＭのアドレス空間に割当てると，ホストシステムで作成したＰＲＯＭに書かれるべき応用プログラムをこの上でデバッグでき，開発を効率よく行なえる。

これらの各ブロックはマイクロプロセッサＺ－80によって制御される。またＰＭＡバスにはＰＭＡポートを介してサブシステムとして接続することが可能である。現在は当マイクロプロセッサにフロッピィディスク，ＣＲＴディスプレーをインタフェースしたスタンドアロンシステムが稼動中である。

６．むすび

マイクロプロセッサ応用システムの開発には，さまざまな援助手法があり，それを支援する多くの開発援助システムが市販されている。しかし，それらは開発のごく一部を支援するものにすぎない。開発の全課程を支援する，さらに機能を拡大した，また開発の心理的側面を人間工学的によく配慮した開発援助システムが期待されている。

我々の研究室でも開発援助システムを試作しているが，まだまだ機能は高いとは言えない。マルチプロセッサシステム－ＰＭＡ－の開発に伴ない，さらに機能の高いシステムの検討を行なっていく。

参考文献

1) 寺田浩詔“マイコン・システムの開発支援はどうあるべきか”，インタフェース No.36，PP74，1980年５月

2) 寺田浩詔，浅田勝彦 "Passive Memoryless Architecture"，情報処理学会 アーキテクチャ研究会31－2，1978年６月

3) 浅田勝彦，寺田浩詔，小谷武史 "Passive Memoryless Architecture の一実現法" 信学会技法 ＥＣ79－76，1980年２月

4) 武田浩一，寺田浩詔，浅田勝彦“マイクロプロセッサ用汎用クロスリロケータブルアセンブラーＣＲＡＭＢＯ－”信学会技報，ＥＣ76－70，1976年12月

5) H. TERADA, K. ASADA, A. FILP "Debugging and Monitoring Aids in an Integrated Multi-Computer Development System", Technology Rept., Osaka Univ., Vol.28, No.1439，1978年５月