マイクロコンピュータと
ワークステーション 54－1
（1989. 2. 21）

ワンチップマイコンを用いたマルチプロセッサ・システムの試作

御牧 義　南沢 正之
電気通信大学電子情報学科

上坂 達生
三菱電機セミコンダクタソフトウエア(株)

１６ケのシングルチップマイクロコンピュータで構成したマルチプロセッサ・システムを試作した。素子となるマイクロコンピュータは８ビットＣＰＵ，８Ｋバイトの ＥＰＲＯＭ，２５６バイトのＲＡＭ，５本の８ビットＩ/Ｏおよびシリアルポートなどをチップ上に持っている。
１６ケのマイクロコンピュータは４×４に配列し、縦にも横にも相互に接続しさらに端に位置した通信ポートは折り返して縦方向にも横方向にも環状になるように接続した。

A Multiprocessor System Composed of 16 Single-chip Microcomputers

Tadashi MIMAKI　Masayuki NANZAWA
University of Electro-Communications
Dept. of Communication and System

Tatsuo UESAKA
Mitsubishi Electric Semiconductor Software Corporation

A multiprocessor system composed of 16 single-chip microcomputers is constructed. Each microcomputer has 8-bit CPU, 8K-byte EPROM, 256-byte RAM, five 8-bit I/O ports, 8-bit input port and serial port.
Sixteen node processors are connected so as to form a 4×4 two dimensional network. Moreover, the processors disposed at edge position are connected together to form a ring in both longitudinal and transversal directions.

1．はじめに

例えば、雑音の統計的性質を調べる研究のように、多種類の単純な仕事を数１００万回繰り返す場合には、マルチプロセッサシステムが適している事は言うまでもない。仕事が単純であるからこれをシングルチップで実現することが望まれるが、現状では、これは価格の点等から実現が難しい。一方、ディスクリートなマイクロプロセッサを、複数個用いたマルチプロセッサシステムは多数提案され、構築されているが、プロセッサ自身を互いに接続する機構を有していないので、共通バスを用いるか、接続用のハードウエアを使用しなければならないという煩わしさがある。この点では、トランスピュータは接続用の手を有しているので良いが、高価で、数１０個、数１００個を使ったシステムを構築しにくいし、また、ＣＰＵに要求される機能もそれ程複雑でなくて良いことが多い。

種々の機器に組み込んで使用されているシングルチップマイクロコンピュータは、自身でＲＯＭ、ＲＡＭ、複数の入出力ポート、タイマを有しているので、これらを多数つなぎ合わせてシステムを作ることは容易である。

特に最近のシングルチップマイクロコンピュータ（以下単にマイクロコンピュータと言う）はＲＯＭがＥＰＲＯＭであったり、ＥＰＲＯＭをピギイバックで外付けできるようになっているので、マルチプロセッサの環境でのプログラム開発が容易になっている。さらにＣＰＵとしての機能も、８ビットのマイクロプロセッサの改良版になっているので、豊富であること、大量に生産されているので極めて低価格であること等が有利である。

以上のような考え方から、既存のマイクロコンピュータを用いてマルチプロセッサシステムの試作を開始したのであるが、マイクロコンピュータの接続方法を含むハードウエアの構造、これを制御するコントロールプログラム（ＯＳ）の開発、アプリケーションプログラムの構成、プログラム開発環境の整備、システムのシングルチップ化への可能性等、解決しなければならない興味あるテーマがいくつも存在している。

ここでは試作した既存のマイクロコンピュータを用いたマルチプロセッサシステムのパイロットモデルの主としてハードウエア構成について報告する。

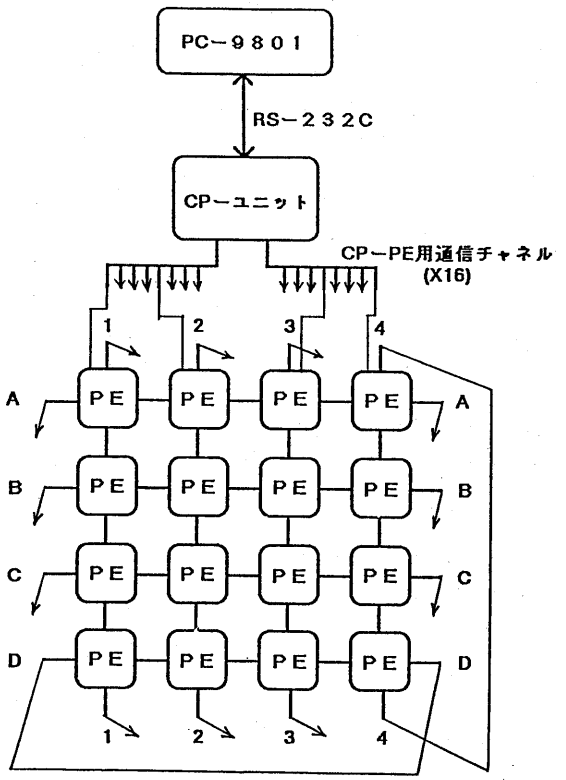
2．システム構造

パイロットモデルの全体の構成は第１図に示したものとなっている。マイクロコンピュータから構成されているＰＥユニット（Ｐｒｏｃｅｓｓｏｒ　Ｕｎｉｔ）は１６個あり、これを４×４の２次元状に接続し、さらに両端は縦方向にも横方向にも接続して環状としてある。これらの接続は８ビット並列のコミュニケーションポートで実現している。

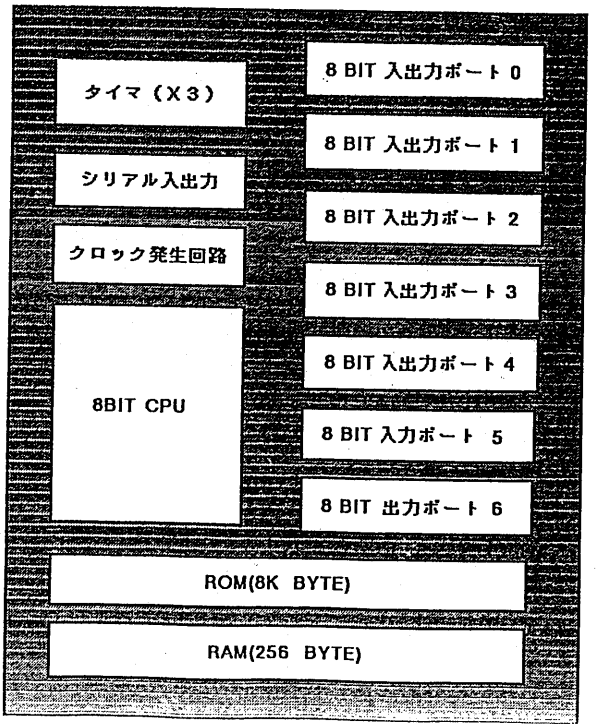
さらに別のもう１対の８ビットコミュニケーションポートは１６対をまとめて特別なプロセッサＣＰユニット（Ｃｏｎｔｏｒｏｌ　Ｐｒｏｃｅｓｓｏｒ）に接続してある。従って各ＰＥユニットは自分の隣の４つのＰＥユニットおよびＣＰユニットと通信ができるようになっている。

ＰＥユニットとＣＰユニットとに共通に使用したシングルチップマイクロコンピュータ Ｍ５０７４７Ｅは第２図に示すように

①８ＭＨｚで動作する８ビットのプロセッサ
②８ＫバイトのＥＰＲＯＭ
③２５６バイトのＲＡＭ
④５対の８ビット入出力ポート
⑤１対の８ビット出力ポート
⑥１対の８ビット入力ポート
⑦１対のシリアルＩ／Ｏ
⑧３対の８ビットタイマ



第１図
全システム



第２図
マイクロコンピュータの構成

をチップ上に備えたところの、電源と発振子をつければ独立して動作できるマイクロコンピュータである。これらの機能のうち⑥、⑧については今回は使用しなかった。このシングルチップマイクロコンピュータは元来、シングルチップモード、マイクロプロセッサモード、メモリ拡張モードおよびエバチップモードの４つのモードで使用できるようになっており、シングルチップモードでは、上述のすべての機能が使用できる。今回はＲＡＭを増強するためメモリ拡張モードで使用した。このモードにすると内部のバス信号をＩＣのピンから外部に引き出すため、④の８ビット入出力ポートがポートとして使えなくなる。このため、ＣＰユニット、ＰＥユニットの場合ともコミュニケーションポートに必要なだけＩＣの外部に入出力ポートを増設した。

ＰＥユニットは第３図に示すが、マイクロコンピュータをメモリ拡張モードにし、１６KバイトのＲＡＭと４つのコミュニケーションポートを増設したものである。ＰＥユニットーＰＥユニット間はこの増設ポートで接続し、ＣＰユニットーＰＥユニット間はマイクロコンピュータに元からあるポートを利用した。

ＣＰユニット（第４図）はこのマイクロコンピュータに３２KバイトのスタティックｒＡＭと１６ケのコミュニケーションポートを外付けしたもので、別に⑦のシリアルＩ／Ｏを通じて外部のパーソナルコンピュータと通信できる。

第１図にもどって、このシステム全体の役割分担および情報の流れについて述べると、まずＰＥユニットのＥＰＲＯＭ領域にはＰＥ個別システムソフトウエア（前述のＯＳ）　すなわちＰＥ自分自身のコントロールプログラム、ＰＥ－ＰＥ間の相互通信コントロールプログラム、ＣＰ－ＰＥ間の通信コントロールプログラムを含んだＯＳプログラムをあらかじめ入れておく。

ＣＰのＥＰＲＯＭ領域にはパーソナルコンピュータとの通信コントロールプログラム、各ＰＥとＣＰの間の通信コントロールプログラムを含んだＣＰ自分自身のコントロールプログラムを入れておく。

第１図において、パーソナルコンピュータはヒューマンインターフェースおよびプログラムの入出力部の役割を担っている。メッセージまたはプログラムは、パーソナルコンピュータから、シリアル通信を介してＣＰに送られる。

ＣＰは１６本の各ＰＥに名前（すなわちプロセッサアドレス、後述のＮＯＤＥーＩＤ）を与えたり、また、問題あるいは役割分担のプログラムを送り込む。

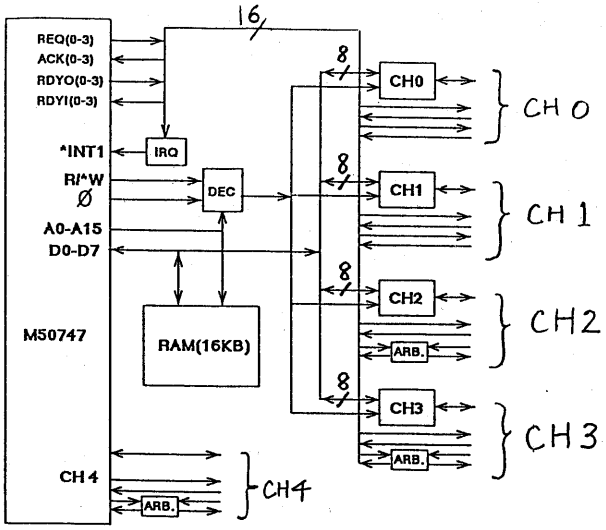
ＰＥ相互間で問題の処理が開始され、処理後　再び、ＣＰ－ＰＥ間の通信を介してメッセージがＣＰに送られる。

３．相互通信方式

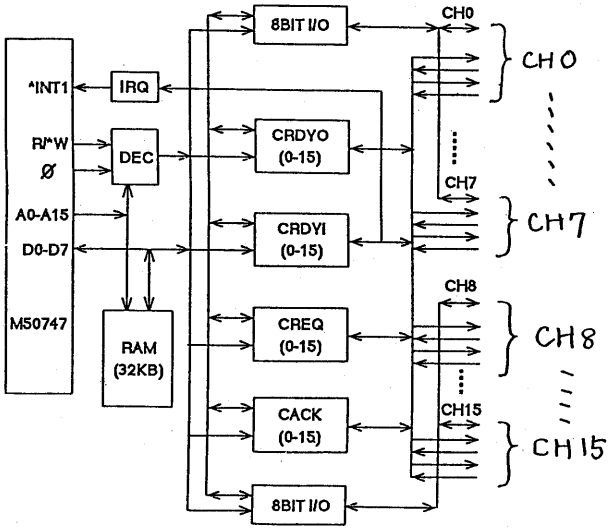
各プロセッサ間の通信は、第５図に示すように１ヶ所の接続につき８bitのＩ/Ｏ線と４本の制御線を用いて行っている。基本的には、セントロニクスインタフェースのようにハンドシェイクを用いた通信を行う。

以下に、１バイトデータをプロセッサＡからＢへ通信する方法を示す。

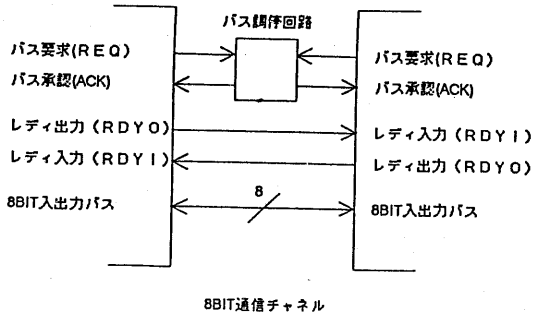
１．ＡはＲＥＱをアサートし、アクノリッジをチェックする。もしもアクノリッジが返ってこないようであれば、アクノリッジが得られるまで送信を待たなくてはいけない。
２．Ａはデータを８ｂｉｔのデータバスに乗せてＲＤＹＯをアサートする。
３．Ｂは、ＲＤＹＩのアサートにより割り込みをかけられるので、どのchannelから入力されたのかを決定した後、目的とするchannel対し受信処理を行うルーチンにとぶ。
４．Ｂは、目的とするchannelよりデータを読み込み、その後にＲＤＹＯをアサートする。
５．Ａは、ＲＤＹＩがアサートされるのを待ってデータバスを切り離しＲＤＹＯをネゲートする。
６．Ｂは、ＲＤＹＩがネゲートされた後、ＲＤＹＯをネゲートして受信を完了する。
７．Ａは、ＲＤＹＩがネゲートされた後ＲＤＹＯをネゲートし、ＲＥＱをネゲートして送信を完了する。



第3図
PEユニット



第4図
CPユニット



第5図
コミュニケーション
ポートの構成

ＣＰ－ＰＥ間は、第６図に示すように、実装上の理由から８bitのデータ線について１６台のＰＥと１つのＣＰのchannelを共用している。このため、２つ以上のＰＥユニットが送信を行おうとすると、バス上でデータが衝突することになる。
従ってＣＰ－ＰＥノード間の８bitデータバスは、双方の合意が成立するまでは必ず入力状態にしておかなければならない。
逆に、ＣＰが２台以上のＰＥに送信を行うことは可能である。この場合ＣＰはすべてのＰＥユニットのハンドシェイクラインを確認しつつ送信をおこなえばよい。ＣＰ－ＰＥ間も、ＰＥ－ＰＥ間も第５図に示すように、このシステムでは、メッセージの通信に必要なハードウエアは前記のような比較的単純なハンドシェイク制御線しか備えていない。
従って、それを用いて、具体的な通信をどのように行うかは、すべてソフトウエアに依存している。そこでここでは、考えられる１つの方法を述べる。以下の方法はあくまでも１つの方法論であり、ソフトウエア次第でこれはどのようにも変更できる。
ここですべてのＰＥは、初期化の段階でＮＯＤＥ－ＩＤと呼ばれるシステム上でただ１つの識別子を割り当てられているとする。これにより自分以外のＰＥが相対的にどの位置にいるのかがわかるのである。
相互通信をコントロールするプログラムは第７図に示すような３種の機能を含んでいることが必要である。
すなわち
- ＴＡＬＫＥＲ：自分自身のＮＯＤＥ－ＩＤとＬＩＳＴＥＮＥＲのＮＯＤＥ－ＩＤを知っているので、そちらの方向のコミュニケーションポートへ送信する。
- ＮＯＤＥ：メッセージを受信しＬＩＳＴＥＮＥＲのＮＯＤＥ－ＩＤを知り、自分でなければ、その番地の方向のコミュニケーションポートへ送信する。
- ＬＩＳＴＥＮＥＲ：自分の番地であれば、送信されてきたデータを受け取る。

４．集合体としてのコントロール方式について

現在集合体としてコントロールするプログラムを開発中であるが、既述したハードウエアはいろいろな場合にフレキシブルに対応できるように作られているのでソフトウエアにより

- プロセスをどのように　各ＰＥに割り当てるのか。
- プロセス間の同期はどうするか。
- コントロールを各ＰＥの上で分散的に実現できないか。

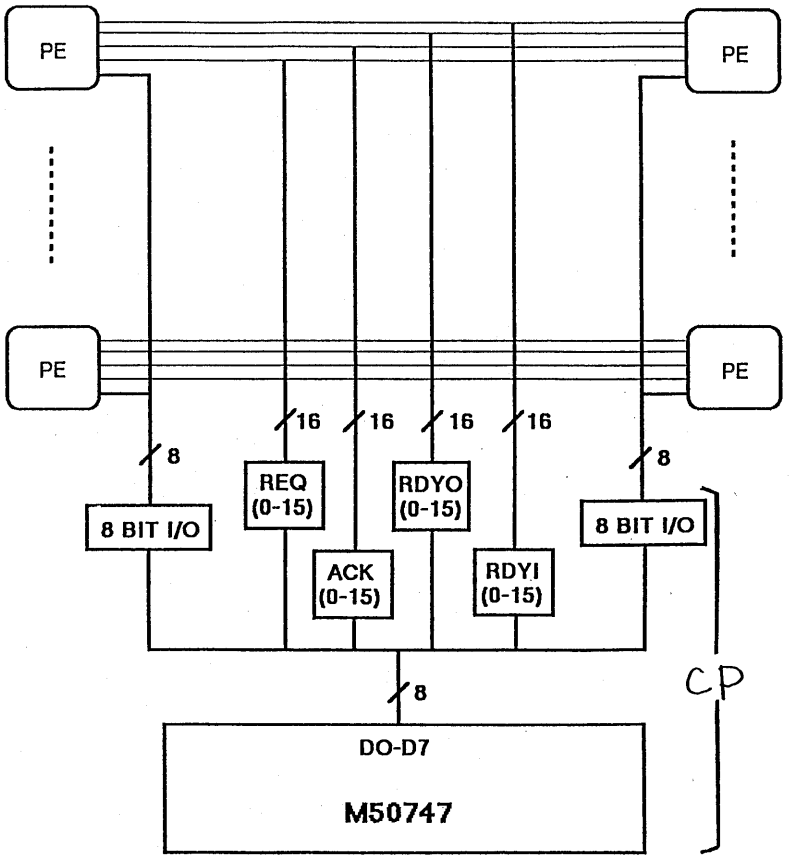
の問題に対応できる見通しである。

５．まとめ

既存のシングルチップマイクロコンピュータを用いたマルチプロセッサシステムのパイロットモデルを構築した。多数の並列Ｉ／Ｏポートを内蔵しているマイクロコンピュータは、相互接続が容易であるから、安価なマルチプロセッサシステムが構成できる。しかし、コントロールプログラムを、どのように乗せるか、アプリケーションをどのように処理するか、を定めるソフトウエアの開発が、システムの機能を定める鍵である。
今後は試作したハードウエア上でのソフトウエアの開発を急ぐ予定である。

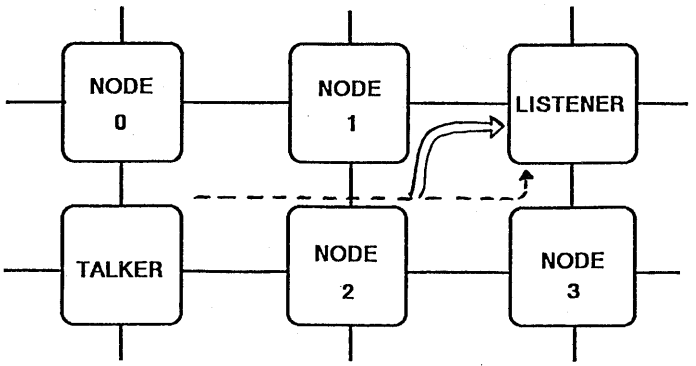
参考文献

1. William C. Athas and Charles L. Seitz：Multicomputers:Message－Passing Concurrent Computers, IEEE COMPUTER Vol. 21, No.8, pp. 9-24(1988).



第6図

CP-PE
間の接続



第7図

通信例