

## 階層型共通バス方式による複合マイクロ コンピュータシステム†

石田 秀昭<sup>††</sup> 畑 田 稔<sup>††</sup> 桧 山 邦 夫<sup>†††</sup>

複数のプロセッサを結合する方式の一つに共通バス方式がある。この方式は、一般に、プロセッサの増設性がよいことと、プロセッサ間において、メッセージ伝送の高速化が図れること等により、広く用いられているが、結合できるプロセッサ数に物理的制限があることと、共通バスの競合制御をいかに行うかが、システム構成上の課題となっている。そこで、本稿では、これらを解決することと、マイクロプロセッサの小型、安価という特徴を生かした、コンパクトで性能/価格比の高い結合方式の実現をねらって、階層型共通バス方式を提案する。本方式では、ユニフォームな階層構造に着目したルーティング方式の採用により、多数のマイクロプロセッサの結合を可能としており、また、集中回路のない分散型バス制御方式に基づく唯一種類のバス制御で、バス間の結合が可能であることから、融通性・拡張性のあるシステムの容易な構築を実現している。また、コンパクトで低価格なハードウェアの開発をねらうとともに、ハードウェア化による性能向上、および、ソフトウェアの負担の軽減を十分考慮した、ソフト・ハード・インタフェースを設定している。これらの方式の概念と、具体的実現手段を明らかにし、実際に開発した複合プロセッサシステムにおいて実測した結果を示す。データ転送には、DMA 転送方式を採用しており、現状システムでは、1 MB/s の転送速度が得られている。

### 1. ま え が き

急速な半導体高集積化技術の進展により、小型、安価で高性能なマイクロプロセッサが使用できるようになり、これらを多数個結合した複合プロセッサシステムの研究が内外で活発化している。複合化の利点としては、(1)高性能化、(2)拡張性、(3)高信頼化などがあげられる。このような利点を実現するための研究課題として、

- (1) 個々のプロセッサに、どのように仕事を割り当てるか
- (2) 多数個のプロセッサを、どのように結合するか

があげられる。(1)については、従来、1台のプロセッサで行われていた処理を、自然な形で並列化、モジュール化し、各プロセッサに固定した仕事を割り当てる方式が多く採られる傾向にあることから<sup>1)</sup>、本論文では、(2)の結合方式を採り上げた。

プロセッサの結合方式は、効率のよい複合プロセッサシステムを構成する上から、重要な課題であり、種類の結合方式が考えられている<sup>2)-7)</sup>。これらのうち、

バス結合方式は、共通バスの競合制御が必要となるが、プロセッサ増設の容易さ、データの高速度伝送、および、すべてのプロセッサの平等化の実現が可能のため、広く使用されている<sup>8)</sup>。本論文では、その一実現方式として、階層型共通バス方式について述べる。

一般に、共通バス結合方式では、結合できるプロセッサ数に物理的制限があることと、バス競合による処理効率低下の問題がある。このため、バススイッチを用いて何本かのバスを切り換えて接続することにより、プロセッサ数を増加させる工夫や、プロセッサ間を複数のバスを用いて結合し、バス競合の緩和を図る方法が考えられているが、ハードウェア上の複雑化を伴っている。

一方、マイクロプロセッサの小型、安価という特徴を生かした、コンパクトで性能/価格比の高い結合方式が要求されている。そこで、本研究では、プロセッサ間における通信量が多くないが、多数個のプロセッサの結合が可能で、かつ、低価格で容易にシステムの実現・拡張を行いたい場合を前提に、共通バス方式の検討を行い<sup>9)</sup>、16ビット・マイクロプロセッサ 68000を使用した階層型共通バス方式による複合マイクロコンピュータシステムの開発を行った。

本方式では、上記観点から、共有メモリをもたない構成を前提に、プロセッサ間のデータ転送を高速に行えることと、各階層のバスが競合しなければ、ローカルなデータ転送が同時に行えるようにすることによって、バス競合の緩和を図った。また、プロセッサ間

† A Hierarchical Common Bus Structure for Multi-microcomputer Systems by HIDEAKI ISHIDA, MINORU HATADA (Microelectronics Products Development Laboratory, Hitachi, Ltd.) and KUNIO HIYAMA (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所マイクロエレクトロニクス機器開発研究所

††† (株)日立製作所システム開発研究所

は、単一バス方式とすることにより、バス競合制御に要するハードウェア量を最小に抑えるとともに、集中回路のない分散制御方式とすることにより、システム拡張時に必要な付加ハードウェアが少なくて済むことを狙いとした。さらに、ハードウェア量の最小化を狙う反面、ソフトウェアとのインタフェースの単純化を重視し、ハードウェア化を図ることによって、プログラミングの負担を軽減することも開発の目標とした。

以下、システム構成、および、バス制御の実現方式について、概要、および、その特徴を述べる。

## 2. システム構成

本章では、階層型共通バスシステムの全体構成と構造解析、および、プロセッサ間データ転送時のルーティング方式について述べる。

### 2.1 全体構成

図1に、階層型共通バスシステムによる複合マイクロコンピュータシステムの構成例を示している。システムは、共通バスごとに設けたBC (Bus Coupler) と呼ぶバスの競合制御部と、マイクロプロセッサを含むCM (Computer Module) と呼ぶ構成単位とから成っており、従来のバス方式のうち、最も単純な単一バス方式を基本としている。

BC は、各階層間でのバスの競合制御を行っており、各CM が平等にバス使用権を得られる方式(後述)を採用している。本方式は、一つの共通バスに最大8個のBCの接続が可能であり、増設により、方式上、最大5階層、 $8^5=32,768$  台のCMを結合することができる拡張性を備えており、しかも、すべて同じBCにより拡張できるという特徴も持っている。

また、各階層のバスは独立しているため、データ転送が局所的で、他のデータ転送とのバスの競合がなければ、図1に例を示すように、同時に複数組のマイクロプロセッサ間の転送が可能であり、システム全体としての転送能力の増大を図っている。なお、データ転送には、DMA (Direct Memory Access) 方式を採用している。

CM は、マイクロプロセッサ、メモリ、I/Oポート、バスアービタから成り、これらはすべて、ボード構成で市場に出回っている。CM内は、図1に示すように、標準的なマイクロプロセッサシステムのバス(ここでは、CMバスと呼ぶ)で結合されており、複数のプロセッサによる密結合構成が可能で、バス競合制御は集中型のバスアービタで行われている<sup>10),11)</sup>。

### 2.2 構造解析

ここでは、解析に基づく本システムの特徴を述べる。Wuらは、マルチコンピュータ向けの結合方式の

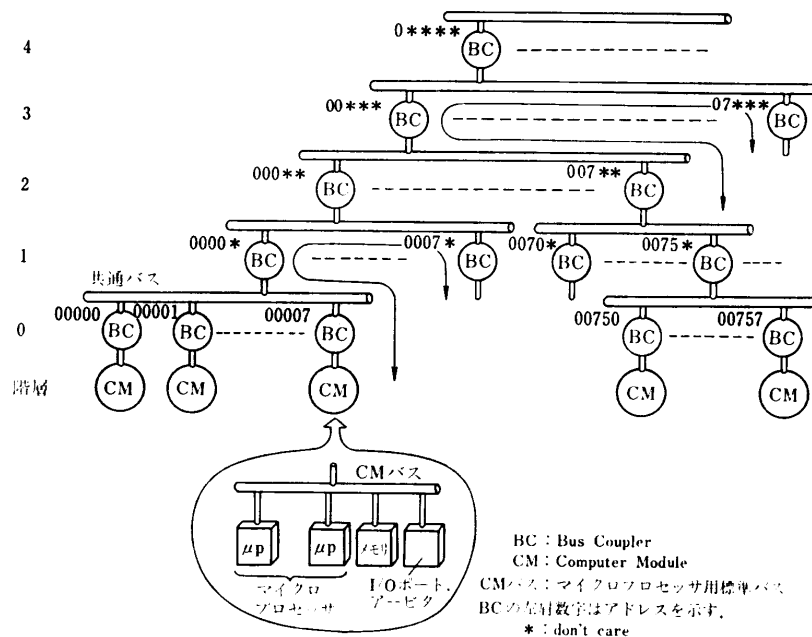


図1 階層型共通バスによる複合マイクロコンピュータシステムの構成(例)

Fig. 1 Structure of multi microcomputer system by hierarchical common bus (example).

うち、三つの代表的な構成 (hypercube, hierarchy, tree) について、定量的な解析を行っている<sup>12)</sup>。この分類によれば、本稿の階層型共通バス方式は、最下位階層にのみプロセッサ (コンピューティング・ノードと呼んでいる) が存在することから、システム構成上、tree に属しており、解析式も、そのまま適用可能となっている。

この結果、階層型共通バス方式は、最下位階層内のメッセージ転送が大部分を占める場合には、最も多くのプロセッサを結合できる構成であることが明確となっている。また、システムを構成するノードが、本方式では、プロセッサとしてのノードと、スイッチとしてのノードに、機能的に分かれていることから、システム構築の単純化が図れることも示される。

### 2.3 ルーティング

データ転送時のルーティングを行うために、各 BC には固有の番号 (アドレス) を付けている。BC のアドレスは、5 桁の 8 進数で表され、その番号の付け方を図 1 に示す。第 0 階層以外の BC のアドレスは、5 桁すべてを必要としないため、不要部分を \* で示している。ルーティングの原理について、以下に述べる。

データ転送に先だて、送信元の CM は、宛先アドレス (受信側の第 0 階層の BC アドレス) を伝送する。各 BC は、このアドレスと自分自身の BC アドレスを比較し、その結果に従ってスイッチ操作を行い、転送ルートを設定する。このスイッチング条件を、図 2

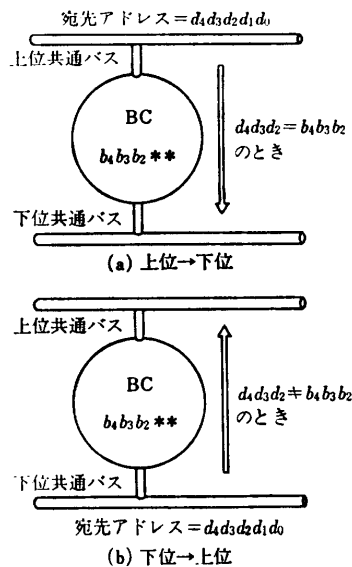


図 2 バスカプラのスイッチング条件 (例: 第 2 階層)  
Fig. 2 Switching condition on bus coupler  
(example: the second level).

の例 (第 2 階層の BC) に示す。

第  $i$  階層 ( $1 \leq i \leq 5$ ) の BC では、上位バスに転送されてきた宛先アドレスと自 BC アドレスのおのおの上位 ( $5-i$ ) 桁同士を比較し、一致すれば、上位から下位方向へのスイッチングを行う (図 2 (a))。また、下位バスに転送されてきた宛先アドレスと自 BC アドレスのおのおの上位 ( $5-i$ ) 桁同士を比較し、一致しなければ、下位から上位方向へのスイッチングを行う (図 2 (b))。

### 3. バスカプラ (BC) のハードウェア構成

前述したように、共通バス方式では、結合できるプロセッサ数に物理的制限があることと、共通バスの競合制御をいかに行うかが、システム構成上の課題となっている。プロセッサの結合方式については、前章で述べたごとく、バス間を BC で結合したユニフォームな階層構造により、多数のプロセッサの結合を可能とする方式を採用している。このため、BC においては、バス間のスイッチ制御、および、共通バスの競合制御として、コンパクトで拡張性のよい簡易な制御方式の実現が必要となる。

メモリを共有する密結合システムでも、メッセージ交換を主体とする粗結合システムでも、バス競合制御の機能は同じであるが、設計目標という観点からは大きな違いがある。密結合システムでは、一般に、バス競合制御を行う頻度が高いことから、信号線の増加や回路の複雑化を抑えるより、少しでも競合制御に要する時間を短縮する工夫が必要である。しかし、粗結合システムでは、いったん共通バスの使用権を得たら、たとえば、10~100 バイトのメッセージを伝送してからバスを解放するので、競合制御時間は、平均バス使用時間に比べ十分小さければよい。むしろ、信号線の削減、拡張性、高信頼化等が重要な設計目標となる。

本開発では、共通バス方式が、プロセッサの増設性、データ転送の高速性の点で優れていることから、メモリの共有のない粗結合の観点にたち、バス制御回路 (バスアービタ) には、信号線が少なく済み、拡張性、信頼性もよい分散制御方式を採用している。

以下、BC でのバス競合制御の概要を述べる。第 0 階層の BC と第 1 階層以上の BC では、両者の機能は若干異なっている。しかし、図 3 の BC の構成図に示すごとく、共通部分が多いことと、システム内のすべての BC が同一であるほうが、システム構築上、融通性があることから、本システムでは、両 BC の機能を

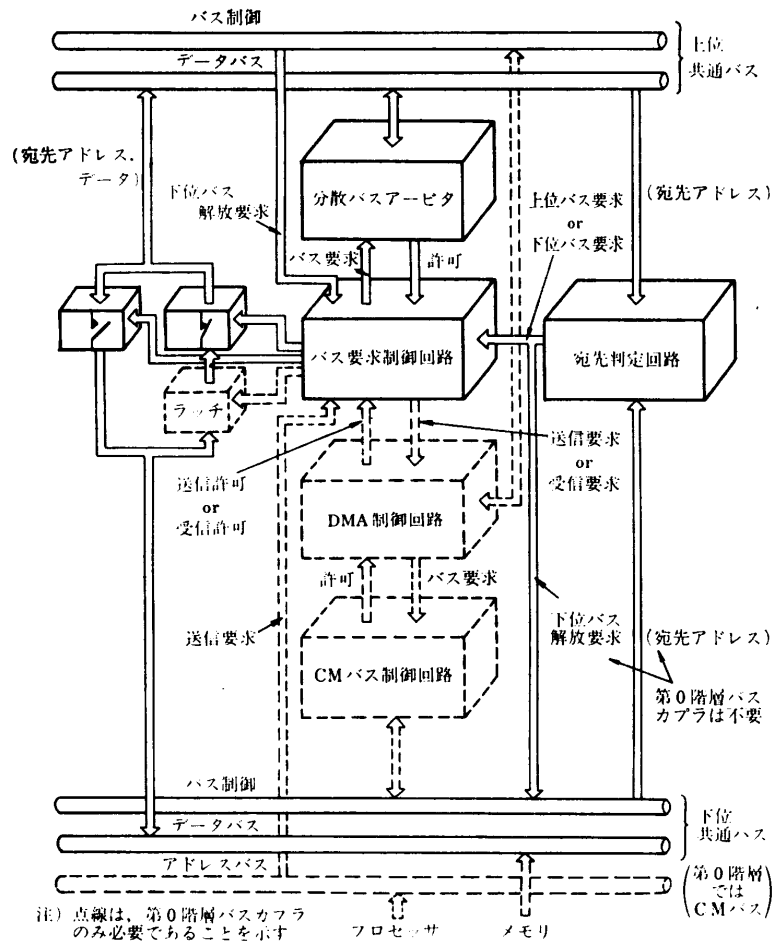


図3 バスカプラの構成

Fig. 3 Structure of a bus coupler.

兼ね備えたものを使用している。

また、本システムでは、いずれかの BC で送信要求が発生すると、すべての BC のタイミング回路が同時に起動される同期方式を採用している。このため、図中のほとんどすべてのブロックに対して、タイミング回路からの信号が供給されているが、図では煩雑さを避けるため省略している。

以下、主要ブロックの制御概要を述べる。

### 3.1 バス要求制御回路

本ブロックでは、他ブロックからのバス要求に基づいて、上位方向、または、下位方向のバス・スイッチングを行う。

#### (1) 上位バス要求

上位バス要求が入力された際、使用する同じ上位バスに対して、他の複数の BC が同時に要求を出してい

る可能性があるため、分散バスアービタに競合制御を依頼する。許可が得られれば、上位方向のバス・スイッチングを行い、得られなければ、得られるまで待つ。

#### (2) 下位バス要求

上位からの下位バスに対する使用要求と、同じバスに対する下位からの要求が衝突した場合、デッドロックを生ずる可能性がある。本方式では、これを避けるため、上位からの下位バス要求を優先としており、下位バス要求発生と同時に、下位の BC に対して下位バス解放要求を出力している。このため、下位の BC では、下位から上位へのスイッチングに成功していても、下位バス解放要求を受け取ると、それをキャンセルし、上位からのバス要求にバスを譲ることとなる。

#### (3) 第0階層 BC でのバス要求

第0階層の BC では、第1階層以上の BC におけるバス要求制御と若干異なる。

送信元 BC では、CM 内のマイクロプロセッサが特定番地に宛先アドレスを書き込むことにより、バス要求制御回路に送信要求が入力され、BC が起動されるようにしている。このとき、バス要求制御回路は、宛先アドレスを

ラッチさせた後、第1階層以上の BC と同様、分散バスアービタからの許可を得て、上位方向のバス・スイッチングを行って、宛先アドレスを上位バスへ出力する。

また、このとき、DMA 制御回路に対して送信要求を出力しておき、受信側へのルーティング完成後、得られる DMA 制御回路からの送信許可によって、宛先アドレスのラッチをやめ、データ転送のためのバス・スイッチングを行う。

受信側 BC では、第1階層以上の BC と同様、下位バス要求によって、下位方向のバス・スイッチングを行うが、無条件に行うのではなく、DMA 制御回路へ受信要求を出力した後、得られる受信許可を待ってから行っている。

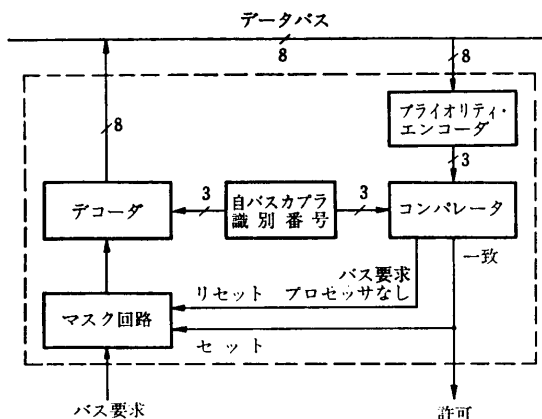


図4 分散バスアービタ  
Fig. 4 A distributed bus-arbiter.

### 3.2 分散バスアービタ

本システムでは、前述したように、共通バスの制御を、信頼性、拡張性の観点から、集中回路のない分散制御方式を採用している。また、おのおののマイクロプロセッサがバスを平等に使用できる方式を採っている。分散バスアービタの制御の概略を図4に示す。

バス要求制御回路から分散バスアービタにバス要求が出力されるとともに、すべてのBCのタイミング回路が起動されることは前に述べた。このとき、すでに、他のBCのバス要求によってタイミング回路が起動されている場合は、その転送が終了するまで、バス要求は保留される。バス要求が出力された後、分散バスアービタはタイミング回路からの指令により、自BCの識別番号を共通バス内のデータバスへ出力する。識別番号は、0~7で、同一共通バスを共有するBC間では、重複しない番号である。このとき、同時に他のBCも同一バスへ、おのおのの識別番号を出力している可能性がある。このため、識別番号の合成結果を各BCがプライオリティ・エンコーダを通して取り込み、この結果を解釈して、自らがバス使用権を得たか否かを判定する方式を採っている。バス使用権を得られなかった場合は、バスが空くまで、バス要求は保留となる。

さらに、本方式では、同一共通バスを共有するBC間で、バス使用の平等化を図るために、バス要求を受け付ける部分にマスク回路を設けている。これは、バスの使用権を得たときにセットし、同一共通バスに対して、バス要求を出している他のすべてのBCのバス使用がひととおり完了したとき、リセットする。これによって、同一バスに対して複数のBCによるバス要

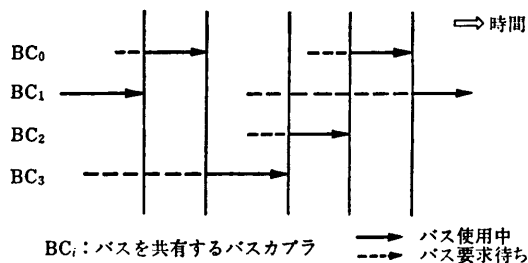


図5 バス・アービトレーション (例)  
Fig. 5 Bus arbitration (example).

求がある間は、同一BCによるバス要求を受け付けないようにして、平等化を図っている。図5に、以上の論理に基づく、バス・アービトレーションの一例を示す。

### 3.3 宛先判定回路

本ブロックの制御は、2.3節で述べたルーティングの原理に基づいている。すなわち、宛先判定回路は、共通バスから取り込んだ宛先アドレスと自BCアドレスを比較し、一致、または、不一致により、バス要求制御回路に対して、上位バス要求、または、下位バス要求を出力する。また、前述したように、下位バス要求出力の場合は、同時に、下位に対して下位バス解放要求を出力している。ただし、第0階層のBCでは、下位はCMであるため、これは不要である。

### 3.4 DMA 制御回路

バス要求制御回路からの送信要求、または受信要求により、DMA制御回路は、CMバス制御回路へバス使用要求を出す。使用権が得られたならば、バス要求制御回路へ許可を伝達するとともに、データ転送を開始する。

送信側と受信側のデータ転送の同期は、DMA制御回路間のハンド・シェイクで行う。送信が完了すると、送信側から受信側へ送信完了を知らせ、双方のBC、および、転送ルート上の各BCは、使用バスを解放してデータ転送は終了する。

### 3.5 割込み制御

以上述べた各ブロックの機能のほか、データ転送中のパリティチェックや、割込み制御機能を設けている。

割込み制御機能については、表1に示すような割込み要因を設けている。割込み制御回路は、割込み要因を検出後、マイクロプロセッサに対して割込みを発生させ、割込み要因ごとの割込みベクトル・アドレスをデータバスへ出力している。

項番1と2の受信終了、送信終了は、CM間のデー

表 1 割り込み要因  
Table 1 Factors of interruption.

項番	要因
1	送信終了
2	受信終了
3	パリティエラー(送信・受信)
4	タイムアウト
5	プロセッサ間同期制御

タ転送の終了をマイクロプロセッサに知らせるためのもので、これについては、次章のソフト・ハード・インタフェースで触れる。

項番4のタイムアウトは、BCが共通バスを獲得後、一定時間経過してもデータ転送を完了せず、したがって共通バスを解放しないことを監視するためのものである。

項番5は、複数のマイクロプロセッサ間で同期制御を行うためのもので、たとえば、特定のプロセッサからの指令で、同時に他のすべてのプロセッサの処理を

開始させるとか、すべてのプロセッサが一定の処理を完了したことを、特定のプロセッサで把握すること等を可能にしている。

#### 4. ソフト・ハード・インタフェース

ソフト・ハード・インタフェースについては、できるだけ、プログラミングの容易性を考慮し、プログラム上のハード制御にかかわる部分を最小に抑えるようにしている。以下、CM間でデータ転送を行う際のソフト・ハード・インタフェースについて述べる。

図6は、送信・受信時のソフト・ハード・インタフェースを示している。転送するデータは、たとえば、数十〜数百ワードを1ブロックとして、DMA転送により連続的に転送される。

送信側では、プログラムにおいて、特定番地に宛先アドレスを書き込むことによりBCが起動される。その後、転送ルート上の各BCは共通バスを獲得し、ルーティング完了後、データ転送が始まる。データ転送は、BC起動直前にプログラムが指定した転送語長とデータ先頭アドレスに基づき、ハードウェアによって自動的に1語ずつ読み出されて行われる。送信終了は、ハードウェアが管理しており、プログラムは、割り込みによって送信終了を知ることができ、パリティチェック結果により、送信がうまくいったか否かを判断する。

ところで、プログラムにより送信要求が出されても、必ずしも、BCは送信を開始するとは限らず、バスの競合制御により、先に受信しなければならないこともある。ハードウェアは、この状況を自動的に判断して、受信を先に行う場合は、受信処理終了後、送信を行っている。

受信側では、ハードウェアが自動的にデータの受信を受け付け、特定番地のバッファエリアにデータを格納する。ハードウェアは、1ブロック分受信後、セマフォカウンタをアップし、割り込みを発生させて、プログラムに受信終了を知らせる。プログラムは割り込みにより、データをバッファエリアから取り出した後、ソフト的にセマフォカウンタをダウンし、パリティチェック結果により、受信がうまくいったか否かを判断する。

バッファエリアの大きさは有限であるため、ハードウェア的なサイクリック制御方式\*で対処

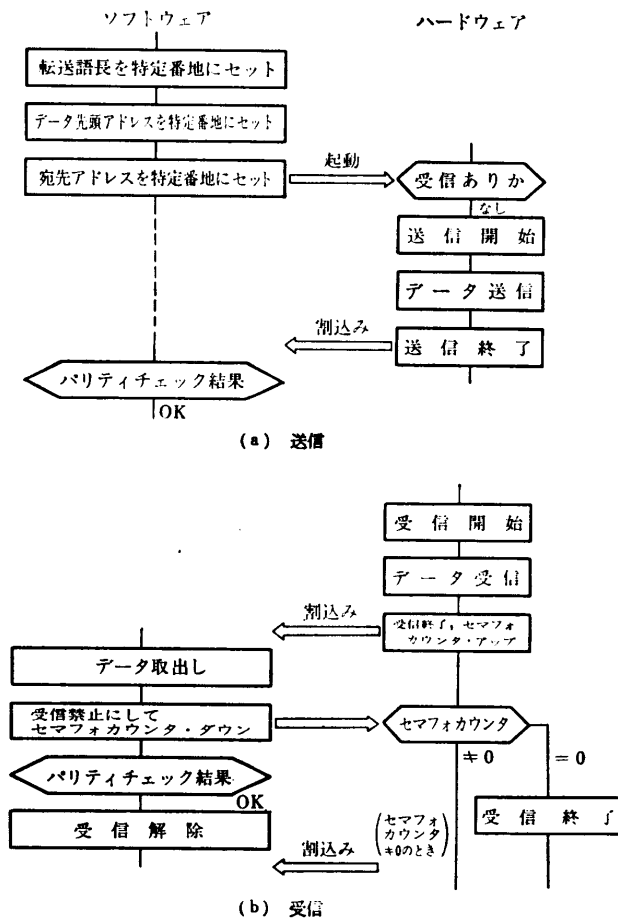


図 6 ソフト・ハード・インタフェース  
Fig. 6 Interface between software and hardware.

\* DMAC (DMA Controller) のもつデータチェイン機能を用いている。

している。受信側では、バッファエリアに空きがあるか否かをハードウェアが判断し、満杯のときは、送信元へ送信禁止を知らせる。送信側では、ハードウェア的に、再送信要求を出力するようにしている。

セマフォカウンタは、ハードウェアが、複数ブロックのデータを次々と受信できるようにするためのものである。上述したように、セマフォカウンタは、1ブロック受信終了時、ハードウェアにより+1され、バッファエリアからデータを取り出す際、プログラムが-1する。このとき、セマフォカウンタの減算結果が0以外であれば、ハードウェアは、再度、受信割込みを発生し、プログラムに、未取出しのデータがあることを知らせる。このため、セマフォカウンタをソフト的に-1するときは、ハードウェアによるデータ受信を一時的に禁止する必要がある、ソフト・ハード・インタフェース上、ソフト的に受信禁止できる機能を設けている。

### 5. 結果の考察

#### 5.1 ハードウェア・サイズ

現状のBCは、A4サイズより、やや小さめの、約20cm×25cmのプリント基板1枚に実装されている。本システム開発時には、68000用DMAC(DMA Controller)が販売されていなかったため、6800用DMACを使用している。このため、6800用DMACとCM内のバスとのタイミング調整上のインタフェース回路に、かなりのICを使用している。したがって、68000用DMACを使用すれば、性能、機能の向上とともに、BCのコンパクト化、低価格化が図れる見込みである。

#### 5.2 性能実測

図7(a)は、性能実測時のシステム構成で、下記の二つのケースについて、データ転送時の測定を行った。

テスト1：データ転送が下位バスでクローズしているケース

テスト2：データ転送が上位バス経由で行われるケース

全体としてのデータ転送時間は、図7(b)に示すテスト・プログラムを用いて測定した。すなわち、最初にタイマLSIから時刻を読み、その後、送信要求を出してデータ転送を行い、送信終了の割込み検出後、再度時刻を読んで、両者の差を求めた。しかし、図7(c)のタイム・チャートに示すように、テスト・プロ

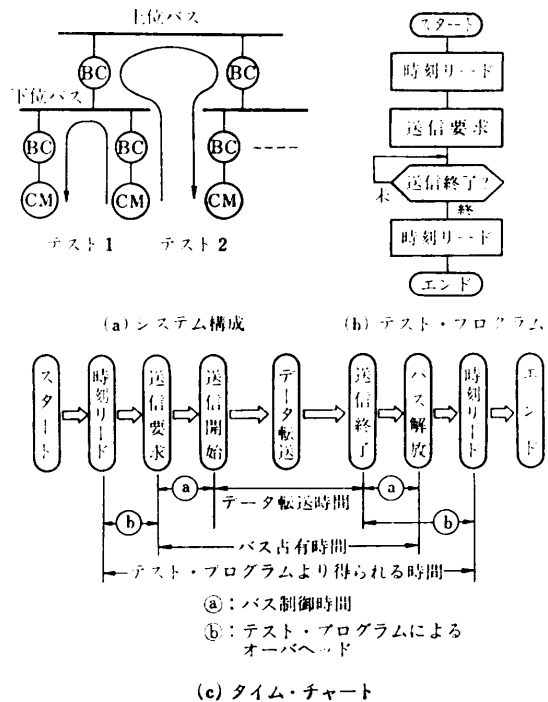


図7 性能実測方法  
Fig. 7 An ability test method.

グラムにより得られた時間のうち、最初の時刻リードから送信要求を出すまでと、送信終了検出から二度目の時刻リードまでは、テスト・プログラムのオーバーヘッド(25ステップ、115μs)であり、バス占有時間には含まれない。また、プログラムによる送信要求から送信開始までと、送信終了からバス解放までのバス制御時間は、転送語長には無関係である。これらの時間の測定はロジック・アナライザを用いた。この測定結果を表2に示す。

以上を考慮して得られたデータ転送時間と転送語長の関係を図8(a)に示す。図からわかるように、測定結果は完全に直線にのらず、ばらつきが発生している。これは、タイマLSIの最少時間単位が20μsであったことと、すべてのCMは自らのクロックで動作する非同期方式を採用していることによる。

データ転送速度は、図8(a)から、下位バス内転送

表2 ロジック・アナライザによる測定結果  
Table 2 Results of measurement by logic-analyzer.

項目	テスト	
	テスト1 (下位バス内)	テスト2 (上位バス経由)
送信要求から送信開始まで	24.4 μs	28.2 μs
データ転送時間(語長20バイト)	23.0	26.0
送信完了からバス解放まで	0.16	0.16

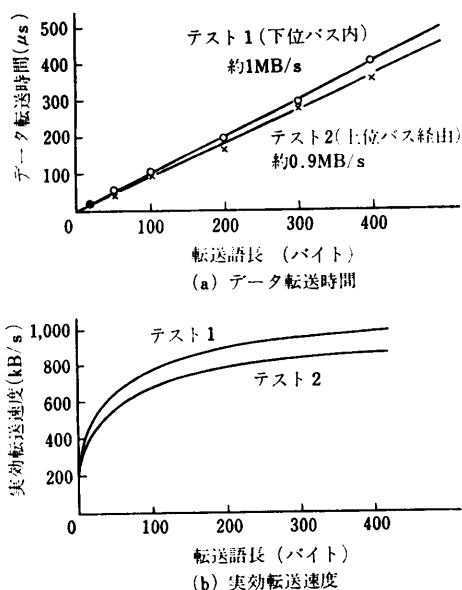


図 8 実測結果

Fig. 8 Measured performance.

のときは約 1 MB/s であり，上位バス経由のときは約 0.9 MB/s である。共通バスのスループットとか，待ち時間を算出するときは，データ転送時間にバス制御時間を加えたバス占有時間を使用しなければならない。すなわち，転送語長を  $n$  (バイト) とすると，バス占有時間  $t(\mu s)$  は，

i) 下位バス内転送のとき，

$$t = 24.6 + 1.0n$$

ii) 上位バス経由転送のとき，

$$t = 28.4 + 1.1n$$

と表される。

実効転送速度は， $1,000n/t$  (kB/s) と表され，計算結果を図 8 (b) に示す。図から， $n=200$  バイトのとき，バス制御の占めるオーバヘッドは約 10% である。

## 6. 階層型共通バス方式の特徴

本方式による複合マイクロコンピュータ・システムの特徴を以下に述べる。

### (1) 多数のプロセッサの接続が可能

従来の共通バス方式では，結合できるプロセッサ数に制限があったが，ユニフォームな階層構造に着目したルーティング方式によって，方式上，最大 5 階層，32,768 台というきわめて多数のプロセッサの結合が可能である。また，転送メッセージをすべて固定長としたモデルに基づく解析により，下位階層間のメッセージが大多数を占めるシステムでは，バス負荷上か

らも，多数のプロセッサ結合が可能であるという特徴が得られている。

### (2) システムの拡張容易性

本システムを構成するプロセッサは，すべて対等であること，バス制御を集中回路のない分散制御方式で行っていること，および，唯一種類の BC により結合が可能なこと等により，融通性のあるシステム構築が可能で，かつ，拡張が容易な構成となっている。さらに，ハードウェア量の少ない簡易なバス制御方式を採用しているため，プロセッサの増設に伴うハードの追加は少なく，コスト的にも優位にあると考えている。

### (3) データ転送の高速化

データ転送は，DMA 転送方式を採用しており，現状のシステムでは，最下位階層内の転送で，約 1 MB/s の転送速度が得られている。また，プロセッサ間の通信が局所的であり，バスの競合がなければ，同時に複数組の通信を可能にしており，システム全体としてのデータ転送能力の増大を図っている。

### (4) ソフトウェア負担の軽減

プロセッサ間のデータ転送については，プログラムが送信要求を出しておけば，ハードウェアが自動的に，データの送信，および受信を行う方式としている。プログラムは，ハードウェアからの割込みによって，送信終了，または受信終了を検出でき，プログラミングの簡略化を図っている。また，送信と受信の衝突時の制御，受信バッファの管理等をハードウェアが自動的に行い，ソフトウェアの負担の軽減を配慮している。

## 7. むすび

以上，階層型共通バス方式による複合マイクロコンピュータシステムについて，その概要，特徴について述べた。

本方式は，バス制御の簡易化，および，分散化により，高信頼化を図っている。また，ハードウェア量が少なく済むことから，コンパクトで低価格な複合システムを実現でき，システムの拡張性に富んでいると考えている。今後，さらに，新素子の導入等により，システムの小型化，性能向上を推進していく予定である。

**謝辞** 最後に，本研究の機会を与えていただいた，日立製作所取締役・コンピュータ事業本部長 三浦武雄博士，同社システム開発研究所長 川崎淳博士，ならびに，本研究の遂行にあたり日頃，ご指導いただい



た同研究所の各位、および、本システムのハードウェア調整にご協力いただいた、日立マイクロコンピュータエンジニアリングの三浦明義、松下正年の両氏に感謝いたします。

### 参 考 文 献

- 1) 日本電子工業振興会：マイクロコンピュータに関する調査報告書（動向調査編），pp. 325-362 (1978).
- 2) 高橋：並列処理のためのプロセッサ結合方式，情報処理，Vol. 23, No. 3, pp. 201-209 (1982).
- 3) Horowitz, E. and Zorat, A.: The Binary Trees and Interconnection Network: Applications to Multiprocessor Systems & VLSI, *IEEE Trans. Comput.*, Vol. C-30, No. 4, pp. 247-253 (1981).
- 4) 畑田， 松山：マルチループ構成によるマイクロコンピュータネットワークシステム，マイクロコンピュータ応用国際コンファレンス，pp. 813-820 (1980).
- 5) Lawrie, D. H.: Access and Alignment of Data in an Array Processor, *IEEE Trans. Comput.*, Vol. C-24, No. 12, pp. 1145-1155 (1975).
- 6) Sulliran, H. and Bashkow, T. R.: A Large Scale, Homogeneous, Fully Distributed Parallel Machine, I, *Proc. 4th Annu. Symp. Comput. Archit.*, pp. 105-117 (1978).
- 7) 相原， 他3名：並列パイプライン・プロセッサ UNIP の応用，信学技報，EC-82-31 (1982).
- 8) 有田， 他1名：階層ルーチングバスとその応用，情報処理学会，計算機アーキテクチャ研資，35-8 (1981).
- 9) 畑田， 他2名：階層共通バス方式による複合マイクロコンピュータシステム，情報処理学会，マイクロコンピュータ研資，20-1 (1982).
- 10) 日立製作所：日立マイクロコンピュータシステム，68000 シングルボードコンピュータ，ユーザーズマニュアル (1982).
- 11) 日立製作所：日立マイクロコンピュータシステム，68000 モニタボード，ユーザーズマニュアル (1982).
- 12) Wu, S. B. and Liu, M. T.: A Cluster Structure as an Interconnection Network for Large Multimicrocomputer Systems, *IEEE Trans. Comput.*, Vol. C-30, No. 4, pp. 254-264 (1981).

(昭和57年10月27日受付)

(昭和58年5月10日採録)