

マルチマイクロプロセッサシステム α -16 の アーキテクチャ†

井上 倫夫†† 小林 康浩††

安価な1チップ μ P (マイクロプロセッサ) を複数結合して、演算処理能力を高める方法の一つに、共有バス結合による資源共有型システムがある。この方式は、各 μ P の機能を均質にできること、処理アルゴリズムに柔軟性があること等により、小規模システムで広く用いられている。しかし、接続可能な μ P の台数に物理的制限があり、処理能力の向上を困難にしている。本報告では、安価で容易に入手可能な1チップ汎用 μ P を用い、これを比較的多数(数十台)接続できる多重化共有バス結合のマルチ μ P システム α -16 のアーキテクチャについて述べる。 α -16 では、共有バスのアクセス方式を簡易化して、ソフトウェアの負担を軽減し、低性能の μ P で複数の共有バスを能率よく利用できるようにした。また、非同期2進木バスアービタの採用によって、同一共有バスでの複数 μ P のアクセス競合を短時間 $O(\log_2 n)$ で調停し、共有バス利用時のオーバーヘッドを極力減少させている。さらに、 μ P 相互の通信制御、共有資源利用時の相互排他的制御などのハードウェア化により、プログラミングの負担の軽減も図っている。 α -16 による FFT (高速フーリエ変換) の並列処理例では、 μ P の使用台数に比例した処理速度が得られている。

1. はじめに

安価な1チップ汎用 μ P (マイクロプロセッサ) を利用して、実験室レベルでデータ処理を容易に実現できるようにした。

多量のデータを入力・保存し、出力することは、比較的高速に行うことができる。しかし、演算処理能力については不満が残る。集中管理された高速の汎用処理装置(大型計算機)、または、データ処理専用機(アレイプロセッサ等)に頼らざるをえない。これら高速の処理装置は、システム価格、運転経費がともに高価となり、とても1ユーザが長時間独占して使用することは困難である。実験室内で時々刻々観測されるデータ(時系列データ等)を、できるだけ速く処理し分析したいのはユーザ心理であり、低価格で、処理速度もある程度満足できるデータ処理装置の実現が望まれる。

その一つの方法として、安価な1チップ μ P を複数結合して、演算処理能力の向上を図るマルチ μ P システムが各種提案されている^{1)~6)}。なかでも、星野の PACS (科学技術用並列計算機) は、問題向きにシステムを構成した好例であり、安価な μ P を用いている点からも、価格性能比のすぐれたシステムとして注目される^{4)~7)}。PACS の成功例からも、大規模並列処理

システムでは、問題(処理アルゴリズム)に適したシステムを構成することが望ましい。しかし、実験室レベルでの特定ユーザ向け専用機(小規模システム)としては、なるべく柔軟性を有しているほうが種々のアルゴリズムに対応でき、システムの利便性が良い。

小規模で、比較的汎用性を有するシステムとして、一般に、共有バス結合による資源(メモリ、I/O 装置)共有型システムが考えられる^{3),5)}。しかし、単一共有バス方式では、 μ P の台数が増加するとアクセス競合による待ち時間を多く必要とし、これによる処理時間の増加(性能低下)が懸念される。樋口、内田らは、高機能接続装置を用いて共有バスでのアクセス競合を低減している⁵⁾。

複数の共有バスを利用することによって、各共有バスでのアクセス競合は容易に緩和される。しかし、すべての共有バスに等しい機能を付加し、同一の共有資源へ複数のアクセスバスを可能とすることは、①システムの構成部品が増大し、コストが割高となる、②ソフトウェアの負担が増大する、などの問題点があり、低価格(μ P が低性能)システムへの適用は困難である。

筆者らは、現在、低性能の μ P を16台用いた多重化共有バス結合の並列処理システム α -16 を稼働させている¹⁰⁾。 α -16 では、複数の共有バスを能率よく利用するために、以下のようにした。

(1) アドレス空間の分割による多重化共有バス方式の採用

共有バスのアクセス方式を簡易化し、これによ

† Architecture of the Multiple Microprocessor System α -16 by MICHIO INOUE and YASUHIRO KOBAYASHI (Department of Electrical Engineering, Faculty of Engineering, Tottori University).

†† 鳥取大学工学部電気工学科

て、制御ロジック回路をシンプルにすると同時に、ソフトウェアの負担を軽減する。

(2) マトリックススイッチ⁸⁾の採用

複数の μ P のアクセスバスと共有バスとを1カ所で集中的に配線し、システム構成部品(素子数、ケーブル長、コネクタ数等)の低減を図る。

また、同一共有バスでの複数 μ P のアクセス競合を短時間で調停するために、非同期2進木バスアービタ^{11),12)}を設けた。これによって、共有バスアクセス時のオーバヘッドを極力減少させている。さらに、 μ P 相互の通信制御、共有資源利用時の相互排他的制御のハードウェア化により、プログラミングの負担の軽減を図っている。

以下、本報告では、多重化共有バス結合による資源共有型マルチ μ P システム α -16 のシステム構成を示し、各部の機能について述べる。

2. α -16 のシステム構成

α -16 は、安価なデータ処理システムを実現するための実験機として製作された。本システムは、資源(メモリ、I/O装置)を複数の共有バスに分散して配置し、全 μ P で共同利用する多重化共有バス結合の密結合計算機複合体である。 μ P と共有バスとの接続は、マトリックススイッチで行っている。図1に、 α -16 のブロック図を示す。 α -16 では、PU(処理ユニット)として、安価で容易に入手可能な1チップ汎用 μ P (8ビ

ット μ P: i8085 A-2¹³⁾、基本クロック 5 MHz) を採用し、さらに、4バイト浮動小数点の演算能力を強化するために、算術演算専用プロセッサ (i8231¹³⁾) を付加している。16台の PU はマトリックススイッチを介して8組の共有バス (CBUS₀~CBUS₇) に接続され、共有資源利用時のアクセス競合を緩和するように意図されている。

また、 α -16 では、メインメモリを 512 kB (kB: キロバイト) と大容量にし、16台の PU とグラフィックコントローラとで時分割利用させている。グラフィック表示機能は、256×256 画素、濃淡 256 レベル/画素である。これを利用して、多量の計算結果をただちに表示し、処理結果を直接視覚に訴えることができるようにしてある。

このシステムの運用は、実験室レベルで特定のユーザを対象とするので以下のようにした。

- (1) データは、共有メモリに格納しておき、各 PU からいつでも任意にアクセスできるようにする。
- (2) プログラムは、アクセス競合を考慮して、たとえ同一の処理手順であっても各 PU 個別にローカルメモリへ格納する。
- (3) タスク単位に並列動作させる。このとき、複数のジョブを同時に動作させることはしない。一つのジョブを複数の並列タスクに分割し、各タスクをそれぞれ PU に分配して処理させる。ここでは、メインタスクを実行する PU を指定し、その管理下ですべての PU を動作させる。
- (4) データの入出力も行う自立型システムとする。データの演算処理は、すべての PU を用いて並列処理させる。データの入出力は、機能分散を図り、複数の PU で平行処理させる。これによって、システムのトータルスループットを高める。

α -16 では、8組の共有バスに接続された共有資源と16台の PU とを効率よく利用するために、

- 1) 複数共有バスへの結合制御、
- 2) 同一共有バスでのアクセス競合の制御、
- 3) μ P 相互の通信制御、

の3点をとくに配慮し、それぞれ専用のハードウェアを設けている。また、共有バス制御のハードウェアをシンプルにし、高速動作させるために、バスアクセス方式の簡易化を図っている。

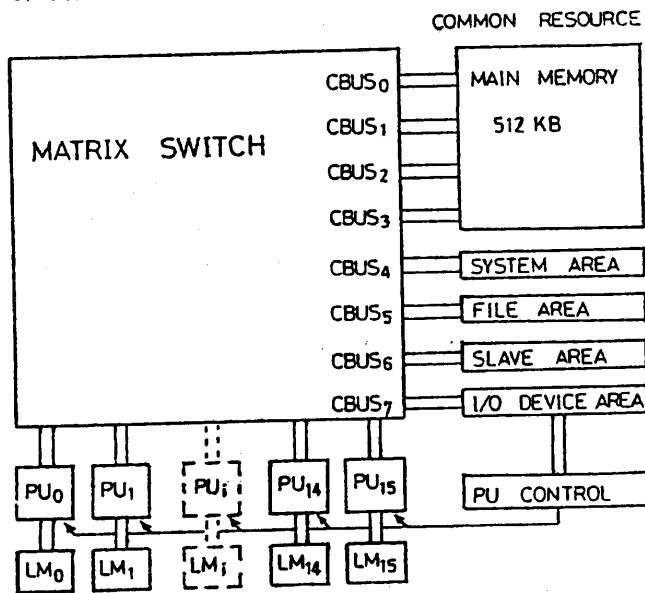


図1 α -16 システムのブロック図

Fig. 1 Block diagram of the α -16 system. PU_i: Processing unit (μ P: i8085A, APU: i8231), LM_i: Local memory.

3. アドレス空間の分割による多重化共有バス

並列処理システムで複数の共有バスを利用する場合、クロスバースイッチ方式のように同一の共有資源に対して複数のアクセスパスが存在することは、システムの信頼性からは望ましいことである。しかし、ハードウェアが複雑となる上、ソフトウェアの負担が増大する。 α -16では、使用する μ Pの処理能力がそれほど高くないことから、各共有バスを使用目的別に専用化して利用することとした。

図2に、 α -16のアドレスマップを示す。共有領域を拡大するために、バンクレジスタを設けPUのアクセス可能な物理アドレス空間を増加させている。各共有資源へのアクセスパス(使用する共有バス)は、 μ Pの上位アドレスビットとバンクレジスタとの内容によって決定する。この方法は、共有資源へのアクセスパスが一つに限定される欠点があるが、ハードウェアが簡単、そのため動作(アクセス時間)が速い、ソフトウェアの負担がほとんどない、などの特徴を有している。 α -16でのアドレス空間と共有バスとの対応を以下に示す。

i) プライベート領域

ローカルメモリ μ P固有のバス

ii) 共有領域

メインメモリ CBUS₀~CBUS₃

システム領域 CBUS₄

ファイル領域 CBUS₅

スレイブ領域 CBUS₆

I/O領域 CBUS₇

実際の処理過程(動作時)は、ローカルメモリをプログラム格納場所、メインメモリをデータ格納場所として使用する。システム領域、ファイル領域、I/O領域等は、特定のPUのアクセスに限定するように(プ

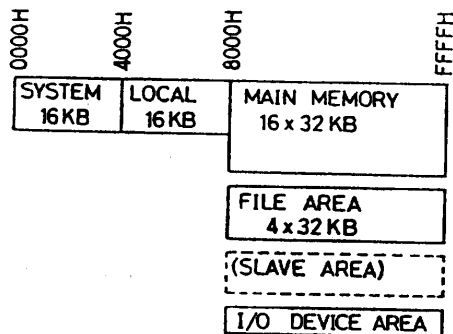


図2 α -16システムのアドレスマップ
Fig. 2 Address map of the α -16 system.

ログラム)する。したがって、仮に、全PUのアクセスがこれらの領域(共有バス: CBUS₄~CBUS₇)で一時的に競合(電源投入後の初期設定; プログラムのロード等)しても、単期間であり、ジョブレベルではその影響を無視できる。

問題となるのは、メインメモリへのアクセス頻度である。信号処理や画像処理を行う場合、16台のPUがほぼ同一のプログラム(タスク)を用いてそれぞれ異なるデータを取り扱う。このときほぼ同時刻にすべてのPUのアクセスがメインメモリへ集中する。多量のデータを処理するとき、この状態が長時間続く。このとき、アクセス競合による待ち時間の累積が増大し、システムの処理速度の低下が懸念される。そこでメインメモリでのアクセス競合を緩和するために、4インタリーブ方式を採用した。データはほぼ連続する領域に格納される。また、4バイト浮動小数点データの場合、連続して4回のアクセスが行われる。このとき、各共有バスのアクセス頻度がほぼ平等となるように配慮した。

メインメモリは、四つのブロックに分割され、それぞれ異なる共有バス(CBUS₀~CBUS₃)に接続する。使用する共有バスは、アドレス情報の下位2ビットの内容によって以下のように決定する。

アドレス	共有バス
4i	CBUS ₀
4i+1	CBUS ₁
4i+2	CBUS ₂
4i+3	CBUS ₃

ただし、 $i=0, 1, 2, \dots$

α -16では、各共有資源をアクセスするとき、それぞれ専用化された8組の共有バスを利用する。使用する共有バスは、PUのアドレス情報によって選択される。これによって、共有バスのアクセス手順、制御ロジック回路、ともに単純化が可能である。

4. 共有バスの制御

一般に、メモリを共有する密結合システムでは、バス結合の制御が頻繁に行われる。したがって、バスアクセス手順の簡易化、アクセス競合制御の高速化などが望まれる。以下、本章では、マトリックススイッチ、バスアービタの制御方式について述べる。

4.1 マトリックススイッチ

α -16では、マトリックススイッチ内に全共有バスを統一的に制御する機能は設けない。各共有バスは、

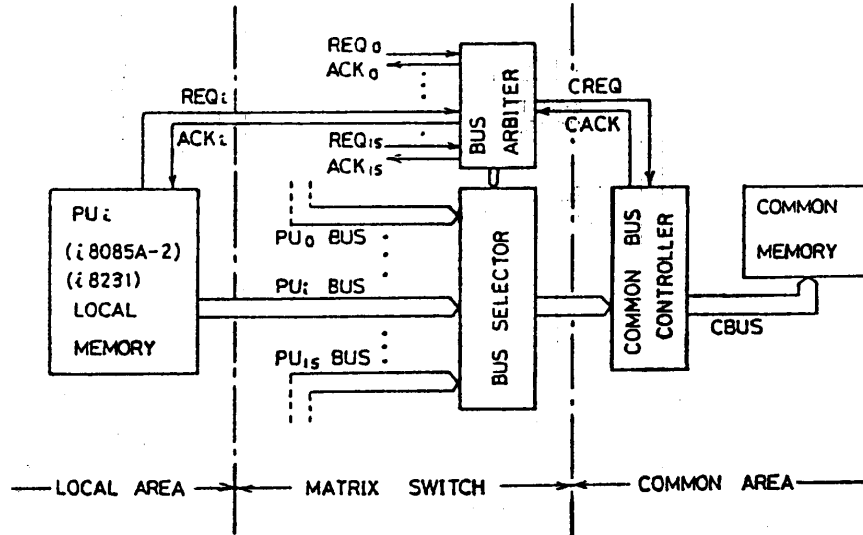
図3 μ P と共有メモリとの接続図

Fig. 3 Block diagram of an access-path from a microprocessor to the common memory.

それぞれ他と独立に動作させる。PU と共有バスとの接続は、各共有バス独立にマルチプレクサによって行う。そのマルチプレクサの制御は、各共有バスに設けたバスアービタが行うようにする。

各PUは、そのつど使用したい共有バス（バスアービタ）へアクセスを要求することとする。複数のPUからのアクセス要求は、各共有バスのバスアービタによって制御（交通整理）される。図3に、PU と共有バスとの接続例を示す。ここでは、①16台のPUは、共有バス（共有資源）を平等に利用し、特定のPUによる共有バスの長時間専有を避ける、②制御回路を単純化する、などを考慮して、共有バスのアクセス方式を以下のように決定した。

(1) 1回に送受（アクセス）するデータは1バイト単位とする。

各PUは、共有バスのアクセス権を得たのち、1バイトのデータのリード/ライト実行後、必ずバスコントローラ（バスアービタ）へアクセス権を返す。これをバスアクセスの1サイクルとする。16台のPUが同時に同一の共有バスの使用要求を出している場合、各PUは15サイクル後には必ずそのバスのアクセス権を得ることができるようにする。また、1バイト単位のアクセス方式は、 μ P自身のアクセス方式と同一にできるため、共有バスの制御回路を単純化できる。

(2) 共有バスのアクセス権は、巡回型優先順位方式によって決定する。

巡回型優先順位方式では、各PUのバスアクセス権の優先順位を各バスサイクルごとに変化させ、最後に

共有バスを利用したPUのアクセス権が最も低くなる。これによって、すべてのPUに平等にアクセス権を渡すことができる。

4.2 バスアービタ

バスアービタは、複数のPUが同一の共有バスを利用するとき生ずるアクセス競合を調停する。 α -16では、アクセス要求を出すPUが16台接続される。バスアービタの性能がシステムの処理能力に大きく影響する。したがって、

- 1) アクセス要求を出しているPUを速やかに選択する、
- 2) すべてのPUが平等に共有バスを利用できるようにする、

などの機能をバスアービタに与えたい。実現の容易さを考え、非同期2進木（バイナリトリー）方式^{11),12)}を採用した。

図4に、バスアービタのブロック図を示す。アクセス要求信号 ($REQ_i: i=0, 1, 2, \dots$) の調停回路は、二者択一を行う2入力基本アービタをトリー状に接続することによって実現した。ここで、16台のPUのアクセス権の優先順位を決定する。

図5に、2入力NAND素子で構成した2入力アービタの基本回路とその動作を示す。初段のNAND素子で構成されたラッチ回路において生ずるクリティカルレースを利用して、二つの要求信号のうち早いほうを選択する。競争に負けた要求信号は、ラッチ回路の入力位置で待機する。この要求信号は、次の機会に優先される。要求信号の選択（調停）に要する時間は、

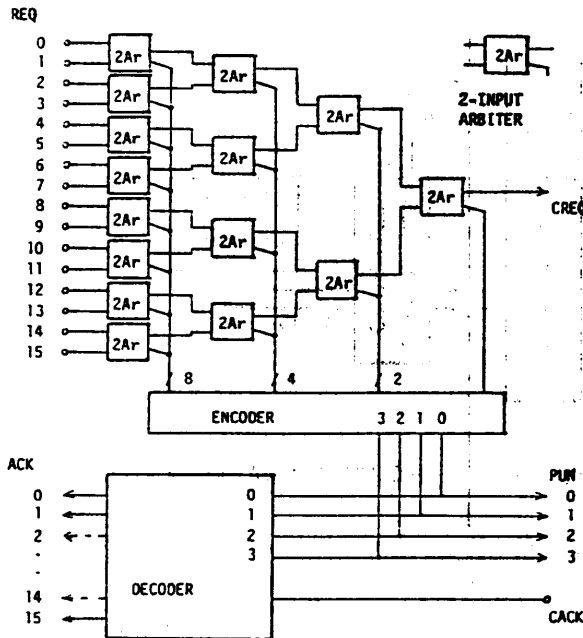


図 4 2進木バスアービタのブロック図

Fig. 4 Block diagram of a binary-tree bus arbiter.

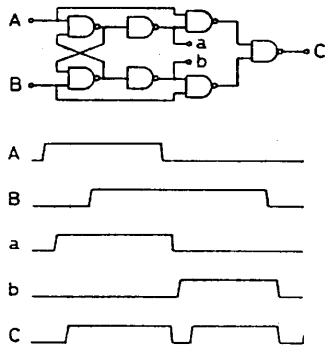


図 5 2入力アービタ

Fig. 5 2-input arbiter.

NAND 素子のゲート遅延時間の 2~5 倍でありきわめて短時間である。

2入力基本アービタの調停時間を τ [s] としたとき、トリー状に接続した n ($n=2^l$, l は整数) 入力 2進木アービタの調停時間 τ_n [s] は、次式で与えられる。

$$\tau_n \approx \tau_0 \log_2 n \text{ [s]} \quad (1)$$

したがって、調停時間 τ_n が入力数 n に比例するアービタ(リング方式ほか)に比較して、 $\alpha=16$ で採用した非同期 2進木アービタは要求信号の調停時間がきわめて短い。これによって、共有バスアクセス時のオーバヘッドを短縮できる。

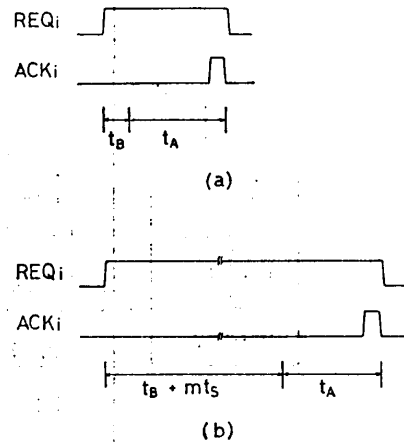


図 6 共有バスのアクセスタイミング

Fig. 6 Timing, for access to the common bus. (a) in the case of no conflict, (b) in the case of conflicts. Where, $m=0, 1, 2, \dots, 15$, $t_B=200$ [ns]: Delay-time of the common bus, $t_A=800$ [ns]: Mean access-time of the main memory, $t_s=800$ [ns]: Cycle-time of the main memory.

非同期 2進木バスアービタは、各 PU のアクセス要求頻度によって調停(優先順位の設定)が次のように異なる。

(1) 共有バスでのアクセス競合がほとんど起こらないとき

共有バスのアクセス権は、ほぼ先着順に $O(\log_2 n)$ の調停時間(遅延時間)で各 PU へ渡される。図 6 (a)に、アクセス競合がないとき、 PU_i が共有バスをアクセスするタイミングを示す。ここで、 t_B [s] は、ドライバ、ケーブル、マトリックススイッチ(アービタの調停時間 τ_n を含む)による遅延時間、 t_A [s] は、共有資源の平均アクセス時間を表す。

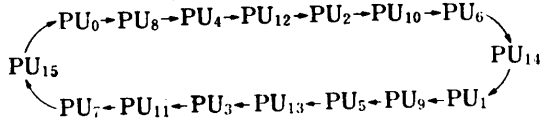
(2) 共有バスでのアクセス競合が頻繁に発生するとき

各 PU は、共有バスのアクセス権が渡されるまで待機しなければならない。図 6 (b)に、アクセス競合が頻発しているとき、 PU_i が共有バスをアクセスするタイミングを示す。ここで、 PU_i の待ち時間は、共有資源のサイクル時間 t_s の m ($m=0, 1, \dots, 15$) 倍である。アクセス権の優先順位は、このとき動作(同一の共有バスにアクセスを要求)している PU の台数、番号(アービタの入力位置)によって次のように異なる。

(i) 16 台すべての PU が同一の共有バスを頻繁にアクセスするとき

アクセス権の優先順位は、巡回的に決定される。た

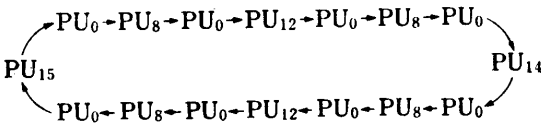
たとえば、以下のような順位でアクセス権が渡される。



このとき、 PU_i へのサービス頻度は $S_i = 1/16$ ($i=0, 1, 2, \dots, 15$) となり、各 PU とも平等である。

(ii) 一部の PU のみが同一の共有バスを頻繁にアクセスするとき

アクセス権の優先順位は、固定される¹²⁾。たとえば、 $PU_0, PU_8, PU_{12}, PU_{14}, PU_{15}$ の5台が同一の共有バスを頻繁にアクセスするとき、以下のような順位でアクセス権が渡される。



このとき、 PU_i へのサービス頻度 S_i は以下のようになる。

$$\left. \begin{aligned} S_0 \approx \frac{8}{16}, S_8 \approx \frac{4}{16} \\ S_{12} \approx \frac{2}{16}, S_{14} = S_{15} \approx \frac{1}{16} \end{aligned} \right\} \quad (2)$$

したがって、全 PU を利用しない処理 (I/O 動作等) で同一の共有バスでアクセス競合が予想されるときには、上記のことを考慮してプログラム (PU へのタスクの配分) すれば、一部の PU へアクセス権の優先権を与えて動作させることができる。ただし、最も低い優先権の PU_i でも、サービス頻度は $S_i = 1/16$ が確保される。

5. プロセッサの制御

プロセッサの制御については、できるだけ、プログラミングの負担を軽減するために、ハードウェア化を図った。以下、PU 相互の通信制御、共有資源利用時の相互排他的制御について述べる。

5.1 PU 相互の通信制御

タスクの起動、複数のタスクの同期を行うためには、PU 間相互の通信が必要である。この PU 間の通信を能率よく行うために、PU 間相互の割り込みを制御する専用回路を共有 I/O 空間に2組設けている。これらは、PU (タスク) の起動、起動の確認、タスク終了の確認などを行うときに利用する。タスクの受渡しは、必要な情報をメールボックス (システム共有空間) に書いておき、相手 PU へ割り込み信号を送るこ

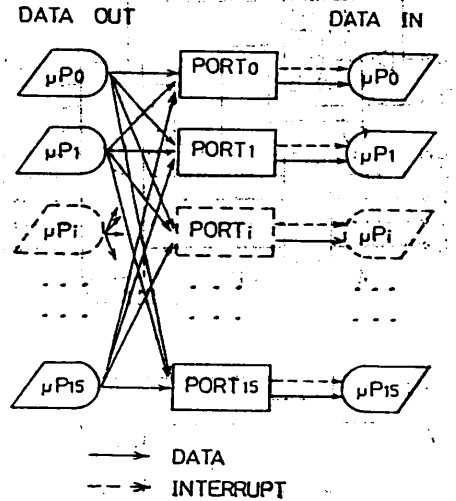


図 7 μP-μP 割り込み制御回路

Fig. 7 Interruption for respective microprocessors.

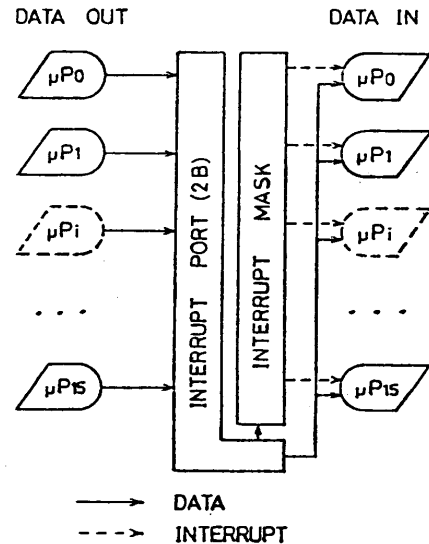


図 8 全 μP への割り込み制御回路

Fig. 8 Interruption for all microprocessors

とによって行うようにする。

1) 特定 PU への通信

図 7 に、特定 PU (μP) へ割り込み信号を送る制御回路のブロック図を示す。おもに、異なるタスクを個々の PU で個別に処理させるとき利用する。

2) 複数 PU への同時通信

図 8 に、複数 PU (μP) へ同時に割り込み信号を送る制御回路のブロック図を示す。複数の PU を同時に起動し、同一のタスクを並列に実行させるとき利用する。このとき、特定の PU をマスクできる。

これらの制御回路は、ソフトウェア上処理に矛盾が

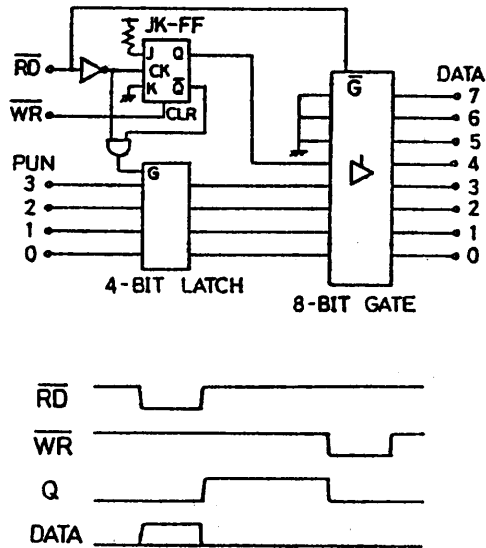


図9 Test and Set 回路
Fig. 9 Diagram of a Test and Set.

生じないかぎり、すべての PU が平等に利用できる。また、自分自身を含めてすべての PU へ割り込み信号を送出できる。

5.2 相互排他的制御

マルチプロセッサシステムにおいてソフトウェア上重要なものに、共有資源（システムの管理テーブル、I/O 装置等）利用時の相互排他的制御がある。ここでは、他の PU のアクセスを一時禁止しなければならない。そのために、各 PU（タスク）はあらかじめ定められた特定のフラグの状態を確認してから目的の資源を利用するようにする。α-16 で用いる μP には、このセマフォを実現できる命令がないので、図9に示すような Test and Set 回路を8組設けた。この回路（ポート）には、フラグの状態と現在その資源を利用している PU の番号がセットされている。フラグのセットは、PU がこのポートをアクセスしたとき、アクセス終了と同時に回路によって自動的にセットされる。したがって、資源の使用権を得た PU は、再度共有バス（ポート）をアクセスしてそのフラグをセットする必要はない。α-16 では、8組までの共有資源を相互排他的に利用できるようにしてある。

6. 並列処理実施例

α-16 による並列処理の具体例を示す。ここでは、信号処理でよく利用される高速フーリエ変換 (FFT) を行った。

データ数 $N=2^7$ 、基数2、インプレイス型の FFT

表1 α-16 による高速フーリエ変換の処理時間
Table 1 Time-consuming in FFT by 4 byte-floating point operations.

サンプル数 N	μP の台数 n				
	1	2	4	8	16
4,096	45,632 (100)	21,348 (46.78)	10,239 (22.43)	5,036 (11.04)	2,501 (5.48)
2,048	19,566 (100)	9,363 (47.85)	4,621 (23.62)	2,284 (11.67)	1,139 (5.82)
1,024	8,539 (100)	4,178 (48.93)	2,073 (24.28)	1,032 (12.09)	516 (6.04)
512	3,743 (100)	1,852 (49.48)	923 (24.66)	461 (12.32)	231 (6.17)

Where (): Ratio in time-consuming to the single processor system.

アルゴリズムは r ステージに分割され、各ステージでは $N/2$ 回の FFT バタフライ演算が必要である⁹⁾。各ステージ内のバタフライ演算は、どれを先に処理してもよく、最も並列処理に適しているといえる。ここでは、このバタフライ演算 $N/2$ 回を n 台の PU に平等に分割し、これを1タスク ($N/(2n)$ 回のバタフライ演算) とした。PU 間の同期は、各ステージ (タスク) 終了ごとに行う。表1に、データ数 $N=512 \sim 4,096$ の FFT の処理時間を示す。演算には、4バイト浮動小数点形式を用いている。

FFT の並列処理では、 n 台の PU による処理時間を、1台の PU による処理時間のほぼ $1/n$ 倍 (ただし、 $n=2, 4, 8, 16$) にできる。各 μP の処理能力の低下はほとんど見られない。この事実は、α-16 ではメインメモリでのアクセス競合に対する調停が効果的に機能していることの証明でもある。

7. おわりに

多重化共有バス結合による資源共有型並列処理システム α-16 のアーキテクチャを紹介した。

α-16 では、共有バスのアクセス方式をシンプルにすることによって、低性能の μP で複数の共有バスを能率よく利用できるようにした。また、非同期2進木バスアービタによって、同一共有バスでのアクセス競合を短時間 $O(\log_2 n)$ で調停し、共有バス利用時のオーバーヘッドを極力短くできた。さらに、μP 相互の割り込み制御回路、Test and Set 回路の付加によって、タスクおよびジョブレベルでの μP の制御を効率よく行うことができるようにし、プログラミングの負担の軽減を図っている。

FFT の並列処理実施例では、 α -16 の処理速度は μ P 単体のほぼ 16 倍であり、 μ P の使用台数に比例した処理速度が得られた。

α -16 システムの性能評価については、別に報告する¹⁴⁾。

α -16 では、比較的低い性能の μ P を用いている。したがって、システムの処理能力は、さほど高くない¹⁴⁾。現在、 α -16 とほぼ同様のアーキテクチャを用い、高性能 16 ビット μ P によるシステムを検討中である。

謝辞 最後に、 α -16 システムの製作にあたり、援助を賜った日進ソフトウェア(株)ならびに同社総合システム本部田淵孝志氏に、感謝いたします。

参 考 文 献

- 1) 高橋義造：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).
- 2) Bear, J.L.: Multiprocessing Systems, *IEEE Trans. Comput.*, Vol. C-25, No. 12, pp. 1271-1277 (1976).
- 3) 大森他：計算機複合体 MICS-II の設計思想と構成，情報学論，Vol. 20, No. 3, pp. 235-242 (1979).
- 4) 星野 力：偏微分方程式解析のためのマイクロプロセッサ複合体，情報処理，Vol. 20, No. 11, pp. 974-982 (1979).
- 5) 樋口，内田：高機能接続装置による計算機複合体の接続方式とその評価，通信学論，Vol. J 65-D, No. 2, pp. 282-289 (1982).
- 6) 小畑，金田，前川：ブロードキャストメモリ結合形マルチマイクロプロセッサシステムの試作，情報処理学会論文誌，Vol. 24, No. 3, pp. 351-356 (1983).
- 7) 星野，白川：並列演算シミュレータ PACS-32, シミュレーション第2回研究会資料，pp. 13-18 (1982).
- 8) 小林，井上，大北：マルチマイクロプロセッサシステム α -4，鳥大工研究報告，Vol. 12, No. 1, pp. 77-83 (1981).
- 9) 小林，井上，大北：マルチマイクロプロセッサシステム α -4 による高速フーリエ変換，鳥大工研究報告，Vol. 12, No. 1, pp. 92-102 (1981).
- 10) 井上，小林：マイクロプロセッサを用いた並列処理システム α -16，シミュレーション第2回研究会資料，pp. 19-24 (1982).
- 11) Plummer, W. W.: Asynchronous Arbiters, *IEEE Trans. Comput.*, Vol. C-21, No. 1, pp. 37-42 (1972).
- 12) 小池，大森：マルチマイクロプロセッサの制御方式，信学会研資，EC 74-11, pp. 15-23 (1974).
- 13) Intel Component Data Catalog 1980, Intel Corp. (1980).
- 14) 井上，小林： α -16 マルチマイクロプロセッサシステムの性能評価，情報処理学会論文誌，Vol. 25, No. 4, pp. 640-646 (1984).

(昭和 58 年 8 月 17 日受付)

(昭和 59 年 1 月 17 日採録)