

マルチミニコンピュータシステムにおける 共有 I/O バス制御方式とその評価†

小林 芳樹** 坂 東 忠 秋** 奥 田 健 三***
川 本 幸 雄** 浜 田 長 晴** 井 手 寿 之****

近年、マルチミニコンピュータシステムが広く用いられてきているが、このようなシステムにおいて CPU 間で共有する I/O の接続方式が重要な課題となっている。実用のシステムでは、CPU に固有の I/O を接続する CPU バスと、CPU 間で共有の I/O を接続する共有 I/O バスとを分離し、両バス間をバスカプラーで結合するバス構成をとる例が多い。本稿でもこのようなバス構成のシステムを扱うが、従来のシステムでは、CPU バスと共有 I/O バスとを直結して共有 I/O のデータ転送を行うため、バスカプラーの素子の遅れによる転送時間の増大、他の I/O 転送との相互干渉による転送時間の増大といった転送速度上の問題があった。

本稿では、上述の問題点を解決する新たな共有 I/O バス制御方式について論じる。この共有 I/O バス制御は、バスカプラーにバッファを設置し、転送データをバッファ間で受け渡しする形で転送するもので、バッファリレー制御方式と呼ぶ。

本文では、バッファリレー制御方式のバッファ割当方法の検討を行い、その制御方式について述べる。次に、方式評価のために行ったシミュレーション結果について考察する。

1. ま え が き

ハードウェア技術の進歩により、マルチミニコンピュータの研究、開発が活発である。マルチミニコンピュータシステムについては、文献^{1)~3)}に様々な観点からの分類や代表的なシステムが紹介されている。例えばリソース間の結合については、単一バス、多重バス、専用バス、専用バスカプラー、環状バス、マルチポート、マトリックス・スイッチと分類され評価されている³⁾。実用のマルチミニコンピュータの結合方式としては、シングルシステムのバス構造が複雑にならずかつマルチシステムへの拡張が容易なことから、専用バスカプラーに属するものが多い。本稿でも、CPU に固有の I/O を接続する CPU バスと、CPU 間で共有の I/O を接続する共有 I/O バスとを分離し、これらのバス間をバスカプラーで結合したマルチミニコンピュータシステムを扱う。

一般に共有 I/O バスにはデータ転送速度に関して 2 つの問題がある。一つは中継素子や距離等により転送

時間が大きくなること、一つは他の I/O 転送との相互干渉によりバスの負荷が増大すること、である。このため高速のファイル装置を共有 I/O バスに数多く接続することが難しくなっていた。

これに対し本稿では、前述の転送速度の問題点を解決し、高速なファイル装置をより多く接続できる新たな共有 I/O バス制御方式を提案している。この制御方式は、バスカプラー等の中継装置にバッファを設置し、データをバッファ間で受け渡しする形で転送するもので、バッファリレー制御方式と呼ぶ。本稿では、先ずこのバッファリレー制御方式について述べ、次に方式評価のために行ったシミュレーション結果について述べる。なお従来、バス結合の方式⁴⁾や解析⁵⁾及びシミュレーション⁶⁾を扱った文献はあるが、本稿のバッファリレー制御方式に関するものは見当たらない。

2. 従来の共有 I/O バス制御とその問題点

2.1 従来の共有 I/O バス制御

本稿で扱うマルチミニコンピュータシステムの構成例を図 1 に示す。CPU に固有の I/O を接続する CPU バスと、CPU 間で共有の I/O を接続する共有 I/O バスとを分離し、この間をバスカプラーで結合する。このバス構成において、CPU 固有の I/O は互いに独立にデータ転送を行うことができる。一方、共有 I/O については、共有 I/O バス及び転送先の CPU バスの両方

† Evaluation of Common I/O Bus Control in Multiminicomputer System by YOSHIKI KOBAYASHI, TADA AKI BANDO, KENZO OKUDA, YUKIO KAWAMOTO, NAGAHARU HAMADA (Hitachi Research Laboratory, Hitachi Ltd.), and JUSHI IDE (Omika Works, Hitachi Ltd.).

** (株)日立製作所日立研究所

*** (株)日立製作所日立研究所 (現在 宇都宮大学工学部)

**** (株)日立製作所大みか工場

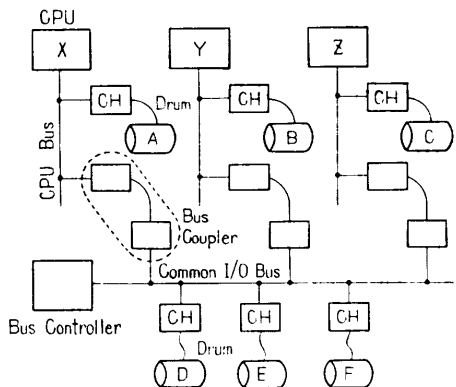


図 1 共有 I/O バス
Fig. 1 Common I/O bus.

のバスを占有してからデータ転送を行う直結制御方式が従来の制御方式である。なお、この直結制御方式には、両方のバスを占有する方法により

- 〔A〕 共有 I/O バスを先に占有する方法
- 〔B〕 CPU バスを先に占有する方法

の 2 方法に分類される。方法 A では共有 I/O バスの負荷に CPU バス占有待ち時間が加わり、方法 B では 2.2 節(2) にて述べるように、CPU バスの負荷に共有 I/O バス占有待ち時間が加わる。システム構成によりどちらの方法が良いか決まるが、本稿では 4 章のシミュレーション条件において良い結果を得た方法 B を従来方式として選び、Type I と呼ぶことにする。

2.2 従来方式の問題点

従来の共有 I/O バス制御の問題点として、データ転送速度が低下することがあり、この要因として次の 2 つが挙げられる。

- (1) データ転送時の信号遅延時間の増大

共有 I/O バスと CPU バスとはバスカプラで結合されるが、回路的には各々のバスカプラに信号のドライバ、レシーバが入ること、2 つのバスを占有するための時間、バス間の距離等により、共有 I/O のデータ転送時間が長くなる。

- (2) 他の I/O 転送との相互干渉

図 1 のシステムにおいて、ドラム A, D が CPUX と、ドラム B, E が CPUY と転送している場合のタイムチャート例を図 2 に示す。この例では、CPUX に固有のドラム A は、共有ドラム D, E の転送の間待たされることになる。このように CPU バスの負荷は、固有 I/O と転送相手の共有 I/O との負荷の和だけでなく、最悪の場合を考えると、固有 I/O とすべての共有 I/O との負荷の和となる。

従来の共有 I/O バス制御においては上記のような転

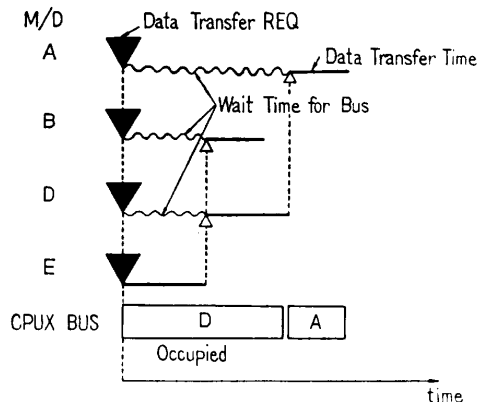


図 2 データ転送のタイムチャート
Fig. 2 Time chart of data transfer.

送速度が低下する問題があり、共有 I/O バスに高速な装置を数多く接続することが難しくなっていた。

3. バッファリレー制御方式

共有 I/O バスにおいて転送速度上の問題が生じた原因を取り除き、高速な装置をより多く接続するために、筆者らはバッファリレー制御方式による共有 I/O バスを提案し、方式検討を行った。

バッファリレー制御方式は、バスカプラにバッファを置き、データをバッファ間で受け渡す形で転送し、各転送におけるバス占有制御を分離して独立に行うようにしたものである。図 3 に基本構成を示す。この図において、CPU 側バスカプラをウィンドウと呼び、また共有 I/O バス側バスカプラをバスコントローラ (バスコンと略す) に集中化している。

さて、本方式で最大の課題であるバッファの割当には次の 2 方法がある。これを従来の直結制御方式につけた Type I に対し、Type II, III と呼ぶ。

- (Type II) I/O ごとにバッファを割当する方法……

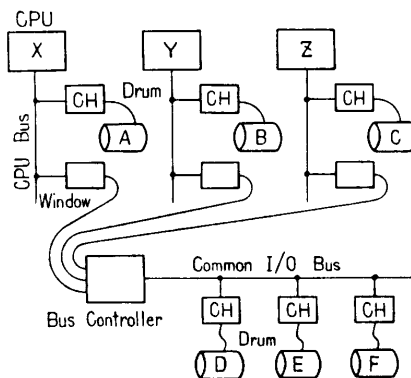


図 3 バッファリレー制御方式の共有 I/O バス
Fig. 3 Common I/O bus of buffer relay control.

ウィンドウ、バスコン各々に、共有 I/O に対応するバッファを割当る。

〔Type III〕 CPU ごとにバッファを割当る方法……ウィンドウには入力、出力バッファを一組だけ置き、バスコンには CPU に対応する入力バッファを割当て、出力バッファは一つだけとする。

さて、Type II は I/O ごとにバッファを割当るため、転送速度の面では有利であるが、バッファ量が多く転送制御が複雑でハードウェア量が大きくなる欠点がある。これに対し Type III は、CPU バス及び共有 I/O バスにおけるデータ転送が 1 語単位であることを考慮して各中継装置に 1 語のバッファを割当るようにしたものであり、バッファ量が少なく転送制御も簡単になる。このような観点から、筆者らは Type III を採用することにした。

バッファリレー制御方式 Type III のバスコン及びウィンドウの内部構成を図 4 に示す。バスコンでは共有 I/O ごとの転送先 CPU 番号及びデータ転送方向を管理し、一方ウィンドウでは共有 I/O ごとの転送先メモリアドレスを管理する。そして次のような制御によりデータ転送を行う。

(1) 共有 I/O ⇒ CPU (メモリ書込)……図 5

5

① バスコンでは、転送相手の CPU 入力バッファが空き状態である共有 I/O の転送要求 (REQ) を優先判定する。

② その結果選択 (SEL) された共有 I/O のデバイスアドレス (DVA) 及び転送データ (DATA) をバッファにセットする。

③ ウィンドウの入力バッファが空き状態であれば、上記の DVA, DATA をウィンドウに転送する。この結果、バスコンのバッファは空き状態に戻る。

④ ウィンドウでは、バッファに情報がセットされると CPU バス占有要求 (REQ) を出す。

⑤ その結果選択 (SEL) されると、DVA に対応する転送先メモリアドレス (ADDR) で指定されるメモリに DATA を書込む。その後 ADDR を +1 する。この転送の結果、ウィンドウのバッファは空き状態に戻る。

(2) CPU ⇒ 共有 I/O (メモリ読出)……図 6

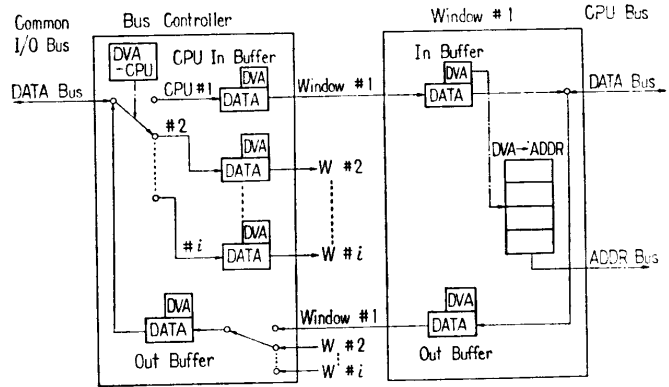


図 4 CPU ごとのバッファ割当によるバッファ構造
Fig. 4 Buffer structure of CPU-buffer assignment technology.

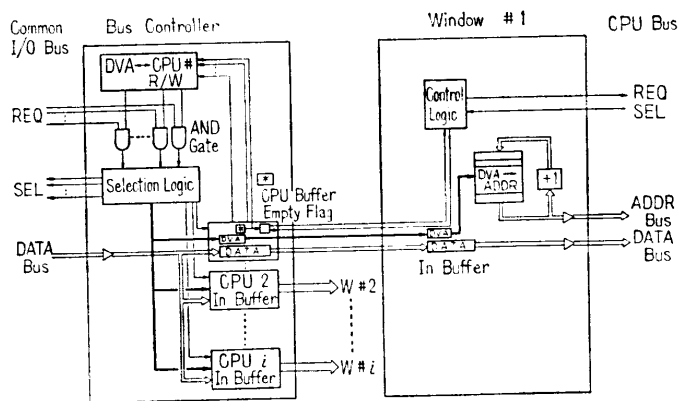


図 5 共有 I/O → CPU へのデータ転送の機構
Fig. 5 Common I/O → CPU data transfer scheme.

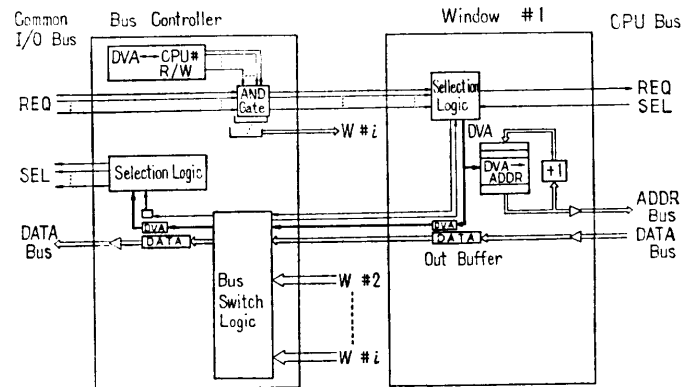


図 6 CPU → 共有 I/O へのデータ転送の機構
Fig. 6 CPU → Common I/O data transfer scheme.

① バスコンでは、共有 I/O からの転送要求 (REQ) を転送相手の CPU ウィンドウに伝える。

② ウィンドウでは、出力バッファが空き状態であれば CPU バス占有要求 (REQ) を出す。

③ その結果選択 (SEL) されると、共有 I/O の

DVA に対応する ADDR で指定されるメモリから DATA を読み出し、DVA と共にバッファにセットする。その後 ADDR を +1 とする。

④ バスコンの出力バッファが空き状態であれば、上記の DVA, DATA をバスコンに転送する。ただし、多くのウィンドウからの要求がくると優先判定され1つが選択されることになる。この転送の結果、ウィンドウのバッファは空き状態に戻る。

⑤ バスコンでは、バッファに情報がセットされると、DVA に対応した共有 I/O を選択 (SEL) し、DATA をその共有 I/O に転送する。この結果、バスコンのバッファは空き状態に戻る。

さて Type III の CPU ごとにバッファを割当する方法は CPU にとってはバッファが1個しかないため、共有 I/O の転送が1つの CPU に集中した場合に、バッファへのデータ集中が問題となる。この影響による転送速度の低下を評価し、同時に従来方式に対する性能向上を評価するため、次のシミュレーションを行った。

4. シミュレーション・モデル

シミュレーションは、図1,3の CPU 3台、固有ドラム各1台及び共有ドラム3台のシステムを対象とする。そして共有ドラムと CPU との転送関係により

① 共有ドラムが各々別の CPU と転送するものを Type I~III-1

② 共有ドラム2台が1つの CPU と、他の1台が別な CPU と転送するものを Type I~III-2

③ 共有ドラム3台が1つの CPU と転送するものを Type I~III-3

と分類し、以下のようにドラム転送速度をパラメータとして変化させ、他は現状のミニコンピュータやロジック素子、メモリ素子の技術レベルにより固定の値に設定してシミュレーションを行った。

(1) ドラムのモデル化

ドラムのデータ転送速度をこの逆数、すなわちデータ転送時間 a μ s/語とする。ドラム・チャネルはドラムからのデータ転送要求が発生すればバスに占有要求を出し、データ転送要求が複数個たまれば連続的にバス占有要求を出すものとする。

(2) CPU のモデル化

CPU の命令実行に伴うメモリアクセスは命令の処理内容により異なるが、平均 1 μ s 間隔で ± 0.3 μ s 幅だけばらついて発生するものとする。なお、入出力命

令実行時にはバスを長く占有するため、これと区別して扱う。一般にプログラム入出力の最大転送速度は 40 k 語/s 程度であるため、発生間隔を 25 ± 10 μ s/語とする。

(3) CPU バスのモデル化

優先順位は、高い順に、共有ドラム、固有ドラム、CPU と固定にし、1語転送終了時に新たに優先判定を行うものとする。

データ転送における CPU バス占有時間は、CPU 及び固有ドラムの場合、一般のメモリ素子のアクセス時間 0.3 μ s にバス信号線を介するデータ転送のためのオーバヘッド時間としての 0.4 μ s を加え 0.7 μ s とする。入出力命令の場合は、デバイスによってはより長くバスを占有するものもあるため 3 μ s とする。共有ドラムの場合はタイプにより異なり、Type II, III ではウィンドウのバッファとの転送であるため固有ドラムと同じ 0.7 μ s とする。一方 Type I では、2.2 節(1)で述べたように転送時間が増加する。この増加時間を現状の技術より 0.5 μ s とし、CPU バス占有時間を 1.2 μ s とする。

(4) 共有 I/O バスのモデル化

優先順位は、複数の共有ドラムの間で固定にし、1語転送終了時に新たに優先判定を行うものとする。

データ転送における共有 I/O バス占有時間は、Type II, III では共有ドラムとバスコンのバッファとの転送であり、CPU バスにおける占有時間に比してメモリアクセス (0.3 μ s) が不用のため 0.4 μ s とする。Type I の場合は、CPU バス占有時間 1.2 μ s に対し、共有 I/O バス占有処理時間等が省れるため 1 μ s とする。

(5) バス間の転送制御のモデル化

バッファリレー制御方式におけるバスコンとウィンドウの間の転送制御は、Type III では転送するレジスタ間の関係が 1:1 であり、レジスタ間を直結して転送を行うことができるため、現状の技術で、0.2 μ s と高速に転送できるものとする。しかし、Type II では転送するレジスタ間が $n:n$ に対応しており、インタフェースの優先制御が複雑となる。ここでは単純に共有ドラムごとに 0.3 μ s のタイムスロットを割当て、共有ドラムが最大5台つくものとして 1.5 μ s 周期のサイクリック転送を行うものとする。

さて、シミュレーション・モデルの一例として Type III-1 を図7に示す。Type III ではドラム転送方向をバスの負荷にとって厳しい CPU \Rightarrow 共有 I/O の向きに

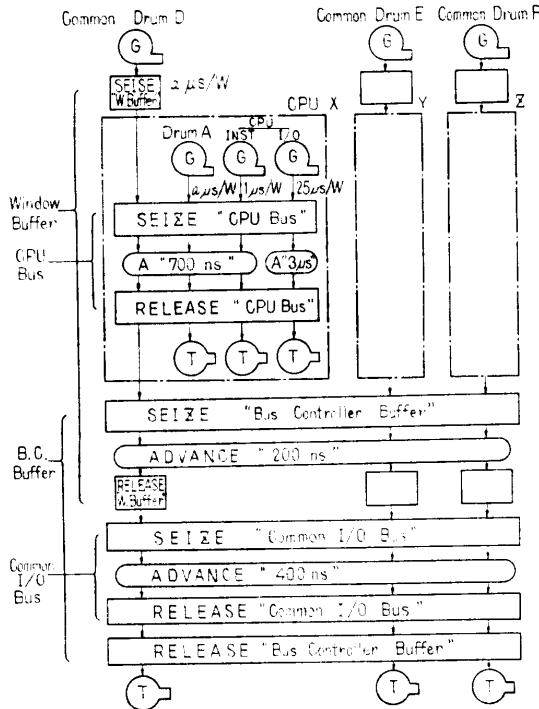


図 7 シミュレーション・モデルの例 (Type III-1)
Fig. 7 Example of simulation model (Type III-1).

している。シミュレーションは、このようなモデルを GPSS (General Purpose Simulation System) で記述し、ドラムのデータ転送時間々隔 $a \mu\text{s}/\text{語}$ をパラメータとして実行した。これによりバスの負荷を実質的に変化させて、ドラムのデータ転送要求の発生から終了までの時間を評価分析した。

5. シミュレーション結果の考察

5.1 バッファ割当の評価

Type II, III のシミュレーション結果を、ドラムのデータ転送発生時間々隔 $a \mu\text{s}/\text{語}$ と転送要求の最大待ち時間とについて整理した結果を図 8 に示す。転送要求の最大待ち時間とは、ドラム・チャンネルに 1 語のバッファがあるとして、ドラムのデータ転送要求が発生してからこのバッファとの間でデータ転送を行うまでのバッファ占有待ち時間の最大値を指す。したがってドラム・チャンネルに 1 語のバッファしかない場合には、転送要求の最大待ち時間が 0 でないものはオーバーランすることになる。オーバーランしないためには図に示すようなバッファ容量が必要となる。

ここで、ドラム・チャンネルのバッファ容量をいくら増加しても、ドラムの転送速度を上げるとオーバーランしてしまう限界の $a \mu\text{s}/\text{語}$ を比較すると、複数の共有

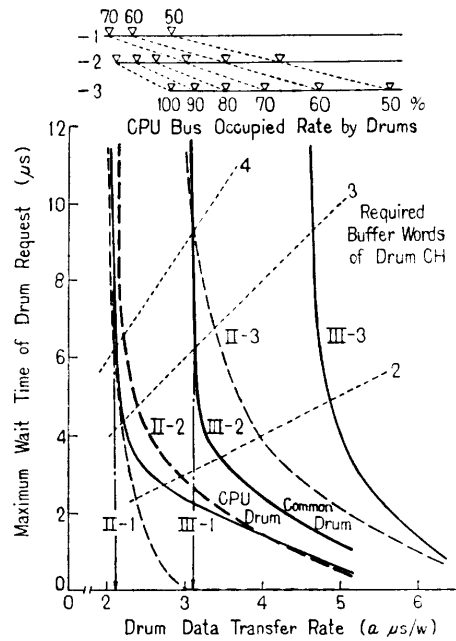


図 8 ドラム・データ転送の最大待ち時間
Fig. 8 Maximum wait time of drum request.

ドラムが 1 つの CPU に集中する場合には、

$$\text{Type III-2/II-2} = 3.1 \mu\text{s} / 2.1 \mu\text{s}$$

$$\text{Type III-3/II-3} = 4.6 \mu\text{s} / 2.9 \mu\text{s}$$

と Type III の方が II より 1.5~6 倍大きくなっており、性能的に落ちることがわかる。

この対策については次節で述べるが、ここではドラムのデータ転送による CPU バスの負荷と CPU の性能低下の関連について考察する。図 8 の上部に Type II, III-1~3 のドラム転送による CPU バスの負荷率を

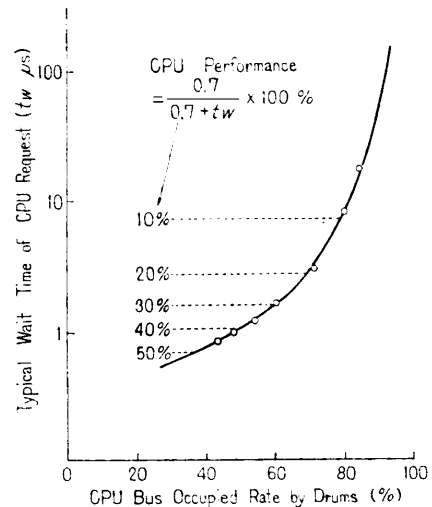


図 9 CPU バス占有率と CPU のバス待ち時間
Fig. 9 Typical wait time of CPU request.

示しているが、この CPU バス負荷率と CPU のメモリアクセスの平均待ち時間とは Type に依らず図 9 に示す関係になる。CPU のメモリアクセス待ち時間 (t_w) の増大は、CPU 処理の性能低下を招くことになる。この CPU の性能を簡単に、メモリアクセス時間に対する待ちを含めたメモリアクセス時間の比

$$\{0.7/(0.7+t_w)\} \times 100\%$$

で表現することにする。

一般にプロセス制御のリアルタイム処理においては、例えば制御対象からの割込みに対するレスポンスをあまり遅らせることはできない。このため一般にバスの負荷を 60% 以下に抑える必要があるといわれるが、この負荷率における CPU の性能は 30% となっており、CPU 処理速度は 3 倍強の余裕を持つ必要がある。

このように通常は CPU バスの負荷率を制限するの必要があり、例えば負荷率 60% までとすると Type II, III における限界の $a \mu\text{s}/\text{語}$ は同一となる。しかし、Type II は III に対し、CPU の性能を犠牲にすればドラムの転送速度を上げることができ、また同じ転送速度でもオーバランに対する余裕度が大きい等の差があるため、Type III の性能低下の要因を分析し対策することにした。

5.2 Type III の制御方法の改良

Type III の制御方式の改良のため、III-2 における 1 番優先順位の低い共有ドラムのデータ転送要求発生から終了までの各経路に要する平均時間を整理した。これを図 10 に示すが、この図から $a=3.1 \mu\text{s}/\text{語}$ において、1 語転送処理時間=データ転送発生時間々隔= $3.1 \mu\text{s}$ と限界に達していることがわかる。この原因は、ウィンドウ・バッファの空き待ち時間が急激に増大して $1.12 \mu\text{s}$ となったためであり、結局、ウィンドウ・バッファにデータ転送が集中したためであると推定できる。このためウィンドウ・バッファの占有時間を図から算出して求めると $1.55 \mu\text{s}$ となっている。Type III-2 では 2 台の共有ドラムと転送を行うため、 $2 \times 1.55 \mu\text{s} = 3.1 \mu\text{s}$ がウィンドウ・バッファの占有されている時間となる。 $a=3.1 \mu\text{s}/\text{語}$ であるから、ウィンドウ・バッファの占有時間が上限に達したことが明らかである。

したがって Type III の CPU ごとにバッファを割当する方法において、ウィンドウ・バッファの占有時間を短縮することにより性能が向上することが予測される。これには、次の 2 方法が考えられる。

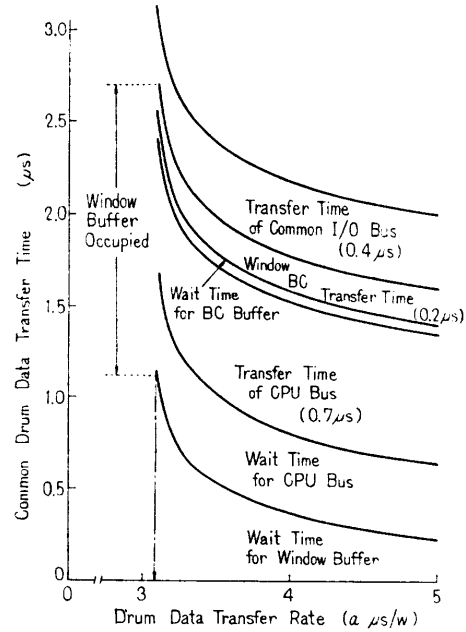


図 10 ドラム・データ転送時間の分析

Fig. 10 Analysis of drum data transfer time.

- (X) ウィンドウ・バッファを 2 個持つ方法
- (Y) ウィンドウ・バッファの空き状態を待たず、CPU バスを先に占有する方法

方法 X は単純にウィンドウ・バッファの占有時間を半分にするものである。方法 Y はウィンドウ・バッファの占有時間の要素である CPU バス占有待ち時間を短縮するもので、方法 X より制御が簡単である。

転送速度に関しては、どちらの方法でも、本稿のシミュレーション・モデルにおいては同一結果となった。この結果を図 11 に示す。なお、本節の改良を加えたものを Type III と区別して III' と呼ぶことにするが、Type III' は完全に Type II のレベルに達している。

5.3 従来方式との比較

改良された Type III' と従来の直結制御方式 Type I との限界の $a \mu\text{s}/\text{語}$ を比較すると、

$$\text{Type III}'-1/I-1=2.0 \mu\text{s}/3.3 \mu\text{s}$$

$$\text{Type III}'-2/I-2=2.2 \mu\text{s}/3.7 \mu\text{s}$$

$$\text{Type III}'-3/I-3=2.9 \mu\text{s}/4.6 \mu\text{s}$$

といずれの場合も約 1.6 倍高速なドラムの接続が可能になることがわかる。また転送速度をドラム台数に換えて、1.6 倍の数のドラムの接続が可能であるともいえる。なお Type III', I-2 におけるドラム転送による CPU バス負荷率を図の上部に示す。この負荷率についても Type III' と I とで 1.6 倍近くの差があり、

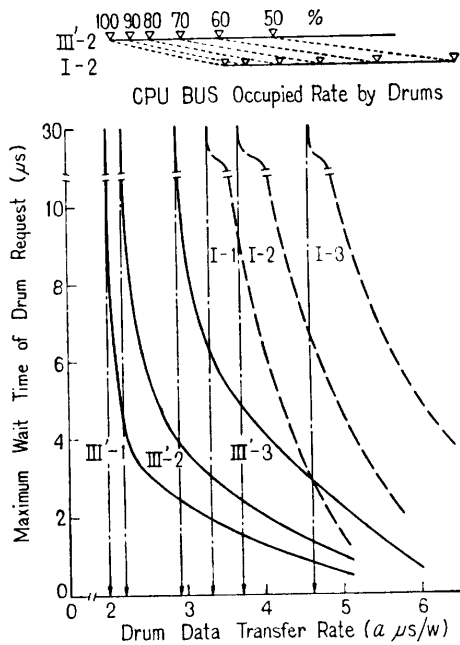


図 11 改良方法 Type III' のドラム最大待ち時間
Fig. 11 Max. wait time of drum in new Type III'.

したがって CPU バスの負荷率を制限しても両者の間に 1.6 倍の転送速度の差がでることになる。

6. むすび

マルチミニコンピュータシステムにおいて、新たなバッファリレー制御方式による共有 I/O バスの方式検討を行い、シミュレーションにより評価した。

共有 I/O バスは、マルチミニコンピュータシステムの中核となる部分であるが、転送時間の増大や他の I/O 転送との干渉により転送速度が低下するという問題がある。筆者らは、この原因を取り除くために、デ

ータをバッファ間で受け渡しする形で転送するバッファリレー制御方式を提案した。本方式における最適なバッファの割当方法として、CPU ごとにバッファを割当てする方法を採用し、その方式検討を行った。

また、CPU 3 台、固有ドラム各 1 台及び共有ドラム 3 台のシステム構成で、バッファリレー制御方式と従来方式の共有 I/O バスのシミュレーションを行い、バッファリレー制御方式の効果が大きいことを確認した。

最後に、本研究の遂行にあたり日頃御指導いただく日立製作所大みか工場桑原部長、同日立研究所平沢主任研究員に深謝いたします。

参 考 文 献

- 1) Kinney, L. L., et al.: A systematic approach to the design of digital bussing structure, Proc. of FJCC, Vol. 41, pp. 719-740 (1972).
- 2) 元岡 達: コンピュータ・コンプレックスの展望, 情報処理, Vol. 15, No. 7, pp. 525-533 (1974).
- 3) 電子技術総合研究所: パターン情報処理システム調査, 電総研々究報告, PIPS-R-No. 3 (1973).
- 4) 上林憲行ほか: ミニコンピュータ複合システム KOCOS の分散バス制御方式と知的インタフェース, 電子通信学会論文誌, Vol. J61-D, No. 11, pp. 842-849 (1978).
- 5) 古谷立美: バス結合マルチプロセッサシステムの解析モデルと解析, 情報処理, Vol. 17, No. 5, pp. 394-401 (1976).
- 6) 古谷立美: ポリプロセッサ・シミュレーション・システム—PPSS, 情報処理, Vol. 18, No. 6, pp. 534-541 (1977).

(昭和 54 年 4 月 3 日受付)

(昭和 54 年 10 月 25 日採録)