

KOCOS の アーキテクチャ (1)

— システムフィロソフィとシステム構成 —

上林憲行 徳田英幸 竹山明 石塚朝生 (慶大工学部)
西垣秀樹 平塚良治 (沖電気工業(株))

1. はじめに

コンピュータの利用分野が拡大してきたので、汎用大型計算機をどのすべての分野に適合させる事が難しくなってきた。そこで、近年各分野においてミニコンピュータやマイクロプロセッサを多数結合してコストパフォーマンスの優れたシステムを構築しようという研究が、活発になってきている。[11]

今までに提案され、又稼動しているミニコンピュータ・コンプレックスには、大きく分けて、次の二つのアプローチがあると思われる。

- [1] 総合システム指向型 : 大型計算機などに見られるように、ハードウェア技術やソフトウェア開発がある息詰まりをきたしている現状に対して、安価で高性能なミニコンピュータを多数結合して大型計算機に優るコストパフォーマンスを実現しようとするアプローチ。[4],[8],[12]
- [2] 資源共有指向型 : 現に普及している数多くのミニコンピュータを結合して互いにハードウェア資源と各々のミニコンピュータに蓄積されてきたソフトウェア資源を共有することによって特殊装置・高価な入出力装置等の有効利用およびソフトウェアの生産性向上が、はかれるシステムを実現し、あわせて信頼性も向上させようとするアプローチ。[2],[5],[13]

従来のミニコンピュータ・コンプレックスの多くは、その研究の重点が、計算機間の結合方式におかれ、コンピュータ・コンプレックスの持つ潜在的な能力を十分に引き出すために必要な次の様な研究が、十分な成果をあげていないように思われる。

- (1) 効率の良い、且つシンプルなプロセス間通信方式の研究。[7]
- (2) コンピュータ・コンプレックスの運用を支えるオペレーティング・システム(OS)の管理形態、機能についての研究。
- (3) ユーザが、コンピュータ・コンプレックスの環境を、十分に活用できるサポートシステムの研究・開発。

本システムでは、設計にあたって上記の(1)~(3)を、重視し、アーキテクチャに、それを、十分反映させる方針をとって[1],[2]の二つのアプローチを調和した異種ミニコンピュータ・コンプレックス KOCOS (Keio-Ōki no Complex System) を設計した。KOCOS は、まず、リソースシェアリング、ロードシェアリングを指向し、最終ゴールとして、リソースを有機的に結合した並列処理を指向するシステムである。本論文では、KOCOS の設計思想、ハードウェア的側面からのシステム構成について述べる。

2. システムフィロソフィ

2.1 設計思想

一般には、コンピュータ・コンプレックスは、アプリケーション・オリエンテ

ドであると言われている。つまり、システムの最適なアーキテクチャは、そのアプリケーションの要求によって構築され、自らその結合状態、管理方式は、決定される。KOCOSの様な、特定のアプリケーションを指向しないシステムではそのアーキテクチャ設計に対する要求は、あいまいである。そこで第一章で述べた事を背景に KOCOSに実験研究システムという性格を加味して、次の設計方針を設定した。

- [1] 本大学に散在するミニコンピュータの記憶容量が限られている事を配慮し、参加するプロセッサの負荷を、最小限にして、実質的機能を向上させる結合損の少ないシステムを設計する。
 - [2] 参加するプロセッサのハードウェアの変更は、メンテナンスの問題点や異機種種の条件からいっさいしない。又、異機種間の結合方式を研究課題として、それら相違を吸収するシステムとする。
 - [3] ARPA ネットワークなどに見られたプロトコル方式から観点を変えた^[7]ミニコンピュータ・コンプレックスにふさわしいシステム空間を統一的、体系的に扱う事のできるプロセス間通信方式を開発する。
 - [4] ミニコンピュータ・コンプレックスにマッチしたオペレーティング・システムの構成、管理形態を研究する。
 - [5] リアルタイム処理可能な“実効転送スピード”の速いシステムとする。
 - [6] 機能的にも十分な融通性・可変性・拡張性が保証されるシステムとする。
 - [7] 並列処理環境とそのサポート・システムに留意したシステムとする。
 - [8] 半導体技術の進歩によって可能となったLSI、特にマイクロプロセッサやマイクロプログラミング技術を積極的に採用したシステムとする。
- 以上の裏から物理的結合度は比較的ゆるいが、論理的には密結合であるシステムを目指して、KOCOSに課された拡張性と結合度という相反する要求を、満たす事にした。

2.2 システムの性格と特徴の概要

KOCOSは 慶應義塾大学工学部矢上台キャンパスを占めている異種ミニコンピュータ・コンプレックスである。参加するプロセッサは、EP (Element Processor) と呼ばれ、Prototype-KOCOSでは、2台のOKITAC-4300Cと2台のNOVA-1200と HITAC-70 マルチシステムから構成される。(図.1)

[1] 計算機間結合方式 [3],[6],[14]

計算機の結合方式には、幾つかの方法がある。即ち単一バス、多重バス、リング結合、マトリクススイッチ、チャネル結合方式等があり、各々に一長一短がある。KOCOSは以下の基準から結合方式を決定した。

- (1) コンピュータの改造が必要のない、技術的に実現性が高く、拡張が容易な方式。
- (2) 効率の良いプロセス間通信方式を実現できる方式。
- (3) 総合システムとしての発展を考慮した結合密度を提供する方式。

以上の理由から、結合方式としては単一バス(C-Busと呼ぶ)による方式を、採用し、各EPはその中枢部を、マイクロプロセッサで構成されるBIU (Bus Interface Unit)を介してこの単一バスに接続される結合方式をとった。

BIUとは、入出インターフェースを通じて接続される。

[2] プロセス間通信機能 : IPCF (Inter Process Communication Facility)

KOCOSの計算機結合方式は、比較的ゆるい結合であるが、KOCOSのIPC F によつて、論理的に非常に緊な結合を実現している。従来のネットワークやコンプレックスにおいて、システム空間を体系的、統一的に取り扱うプロセス間通信方式を実現した例は少ない。^[7] その為、メッセージを他のプロセッサへ転送する場合、転送上のオーバーヘッドと複雑な割り込み処理が各プロセッサへのオペレーティング・システムの負担となり、実質的にその結合密度を下げている。その為、本システムではプロセス間の円滑な通信が可能でミニコンピュータ・コンプレックスにふさわしいIPC F を設計した。^[1]

* KOCOS プロセス間通信機能の基本性格 *

(1) プロセスとプロセスの直接的なメッセージ転送の実現：

実際に転送されるメッセージDM (Data Message) と制御情報、IPC M (Inter Process Communication Message) を本質的に区別し、IPC M によつて“ロジカル・リンク”を、確立して、その転送されるDM は送信BIUの管理のもとに、送信プロセスの番地空間から受信プロセスの番地空間へダイレクトに転送される。そのために各EPは、全く、転送用バッファを持つ必要がなくまた、速い実効転送スピードを提供している。

(2) プロセス間通信基本命令によるきめの細かい相互制御の実現：

プロセス間通信基本命令 (IPC primitives) によつて相手プロセスの制御が可能で、ユーザは開放された豊富なIPC primitives を使用することで、コンプレックス・システムの環境を利用したコンカレント・プロセスが実現できる。

(3) 多様な転送機能：

3種類の同期モード、論理的ブロードキャスト転送、部分転送、離散型データの転送等の機能を有している。また同一プロセッサにおいて多重送信、受信が可能である。

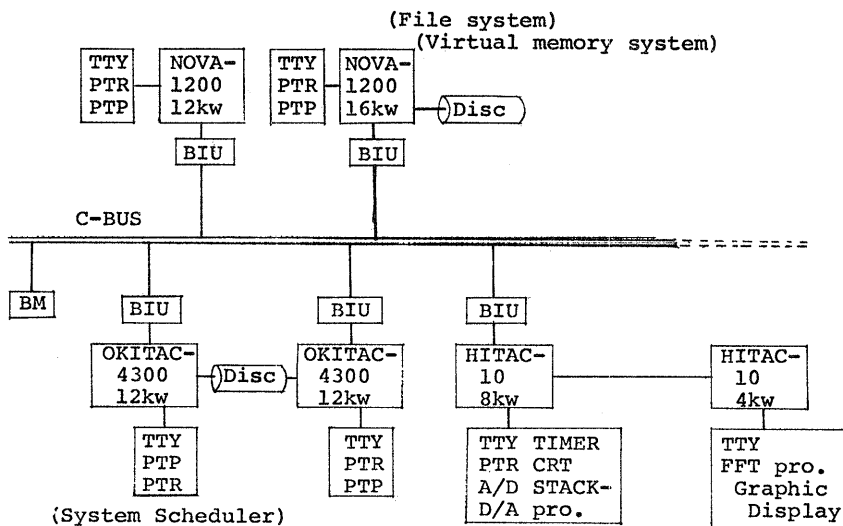


図.1 System configuration of KOCOS

[3] BIUの機能

BIUにはKOCOSの設計思想が集約されている。これは単に各EPとC-Busとのインターフェースを行なうハードウェア装置ではなく、IPCFの物理的・論理的な中核として機能する。KOCOSに要求される課題を考慮すると、BIUは単なるハードウェア構成ではなく、知的機能を持った独立性、自律性に富んだ構成が望ましい。その為、中核部はマイクロプロセッサを中心とするLSIで構成されており、その知的な機能を十分活用して、KOCOSの課題であった可変性、強力なIPCFを実現し、各プロセッサを複雑な転送上の処理から解放し各EPにマルチプログラミングの環境を提供する。

[4] ソフトウェア構成

コンピュータ・コンプレックスを設計するに際して、[1],[2]と共に、大きな要素となるのが、オペレーティング・システムのシステム管理方式である。^[15]コンプレックス・システムでは、信頼性の観点からみると、システム管理機構は、できるだけ集中しない方が望ましい。しかしPrototype KOCOSでは、EPにかなりの制約がありシステム管理機構を、各EPのLOS (Local Operating System) に独立して持たせると、LOSの負荷が増加し、システム全体として冗長である。そのために、リソースの管理はSS (System Scheduler) による集中管理を、各EPで実行される動的プロセスの管理は、LOSによる分散管理方式を採用した。以上の様な“集中分散管理方式”をとる事によって、各プロセッサの独立性と、システム全体の体系化との調和をはかっている。^[1]

[5] バス制御方式

DBC (Distributed Bus Controller) は、単一バスにおけるバス使用の競合を防止するためにKOCOSで開発した完全分散型バス制御方式であり、これによりラウンド・ロビンとラウバスの公平な使用を実現でき、信頼性の向上にも寄与している。

[6] リソース・シェアリング

ユーザは、そのリソースが、物理的にバス上のいかなる位置にあってもよく、システム・コマンドを介してリソースシェアリングを実現できる。

[7] 高速な実効転送速度

システムの性能指数として、転送速度があげられるが、今までは単にバスや回線上の速度が問題にされていたきらいがある。KOCOSではバス上の転送速度、OS, BIUを含めて、送信プロセスから受信プロセスの実効転送速度に注目して、高速な実効転送速度を実現した。C-Bus上では、バースト転送であり、非同期の可変ブロック長転送である。

[8] 論理的システム

BIUのコントロールは原則として、マイクロプロセッサのBIU制御プログラム (BIU-CP) によるので、IPCFの拡充、修正は、BIU-CPの変更で容易に実現でき、ハードウェア的制約を受けないシステムであり機能上の拡張性、成長性を保証している。

[9] 端末装置や大型機の接続

高価な入出力装置や特殊装置の共有の為に直接これらの装置をC-Busへ接続する事が可能であるし、大型機や他のネットワークとのリンクも可能である。

3. システム構成

K₀C₀Sを構成するハードウェア・エレメントとしては、EP, 入出力装置等のリソース, BIU, DBCとC-Busのトラフィックの測定及びシステム障害対策用のBM (Bus Monitor) がある。

3.1 C-Bus (Common Bus)

C-Busは、EP間のデータ転送を行なう為の双方向性の単一バスである。K₀C₀Sでは、初期設備の簡便化、将来の拡張性及び柔軟性を考えて単一バスを採用した。C-Busは、計32本のツイステッドペアで、非同期の転送方式である。

表. 1. C-Busのライン構成

分類	名称	略称	本数	用途
データ転送ライン	Data line	DATA	16	データ転送ライン
	Parity line	PAR	1	パリティライン
データ転送 制御ライン	Control Indicator line for Receiver	CINR	2	転送データのタイプ(ファストアダプ セカドアダプ, データ, フロクエンド)表示
	Data Accept line	DTAT	1	転送データのレディ表示(スロフ用)
	Control Indicator line for Sender	CINS	2	受信BIUの受信表示(正常受信, 異常受信, フロクエンド受信, BIUエラー)
	Data Ready line	DTRY	1	データアクセプト用表示(スロフ用)
システム・障害用 ライン	Abort line	ABT	1	転送異常時アバート用
	Clock Interrupt line	CLKI	1	システム・クロック割り込み用
	Fault Mode	FLTM	1	障害表示
DBC用ライン	Mask line	MASK	3	マスク・ライン (単方向)
	Wait line	WAIT	2	ウェイト・ライン
	Bus Busy line	BBSY	1	バス・ビジー・ライン

3.2 BM (Bus Monitor)

BMの目的は、K₀C₀Sにフィードバックするための各種統計量の収集、異常状態のチェック及びリアルタイム・クロックの供給である。上記の目的の為にBMは、C-Busの使用時間; 頻度や転送語数のカウント及びIPC Mをサンプリングしてトレースし、それらを記録する機能を果たしている。

3.3 DBC

K₀C₀Sは、単一バスによる複合体であるので、バスの競合を防止する必要がある。⁽⁹⁾単一バスで問題となるバスの公平な使用と信頼性の向上という点を考慮して、K₀C₀Sでは、完全分散制御方法のバス・コントローラ(DBC)を開発した。DBC間の接続を、図2に示し

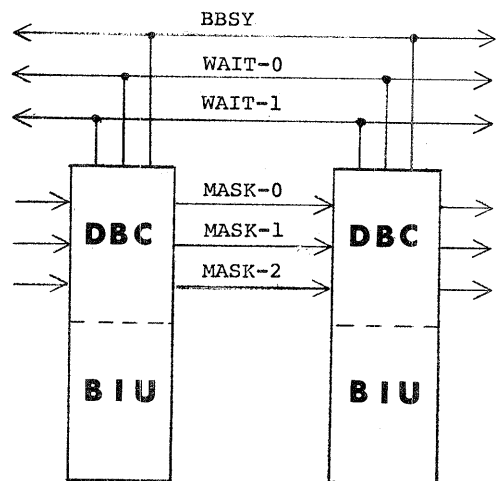


図2. DBC 接続図

DBCの構成を、図3に示す。

DBCの動作を以下に説明する。

- (1) バスを使用するBIUは、自分のDBCに対してREQUESTを出す。
- (2) バスが、ビジーでなければ、RF (Reserve Flag) をセツトし、BBSYにビジー信号がのる。ビジーならば、それが解消するまで待つ。
- (3) RFをセツトすると同時に、自分よりプライオリティの低いDBCに対して、バス使用権を自分のバス使用が終了するまでMASKを用いてマスクする。プライオリティはバス上の物理的位置によって決まる。
- (4) 各DBCは、競合防止のためのディレイタイム (Mini ϕ , Max $2 \mu\text{sec}$) 後にマスクされなければ、バスを使用できる。
- (5) 各DBCは、バス使用終了後まで、RFを維持し、BBSY上のビジー信号を保持する。BBSYがビジーである間は、プライオリティにかかわらず、新たなリザーブは禁止され、RFをセツトしたDBCのリクエストがすべて受けられた後でなくては、新たなリクエストのためのリザーブはできない。これにより、ラウンド・ロビンが可能となった。

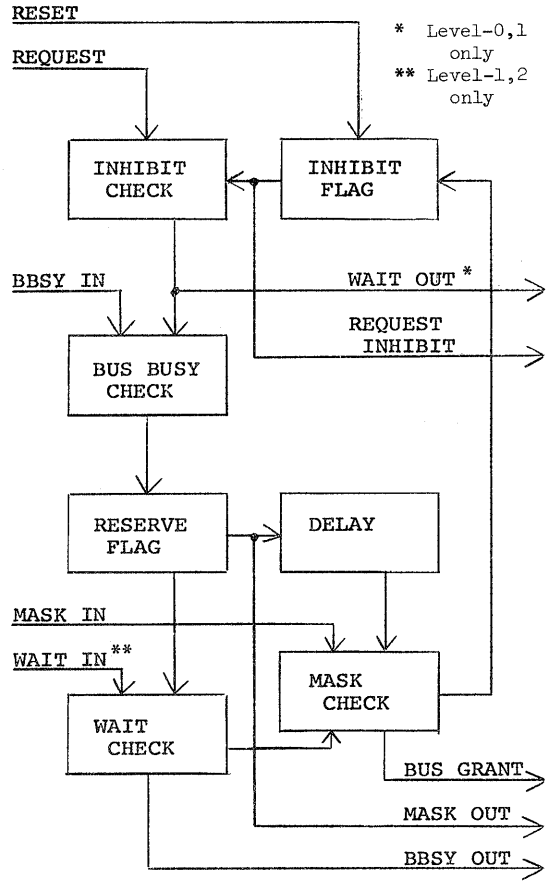


図3. DBCのブロック・ダイアグラム

KBCでは、3レベルのプライオリティ・レベルを、設けている。レベル0が一番高いレベルで、障害対策用。以下、レベル1は、リアルタイム処理用、レベル2は一般のEP用である。低いレベルのリクエストの待ち行列がある時に、高いレベルのリクエストが発生すると、WAITを用いてRFをセツトしているDBCのRFは保持させたままBBSYを、強制的におろさせて、高いレベルのリザーブが可能となる。低いレベルの待ち行列はそのまま保持される。バス使用権を得たBIUが故障した時には、それ以後の他のBIUのバス使用が、許されないためにデッド・ロックが生じる。これを防止するためBMでタイムアウト・チェックを行ない、故障を検出し、故障BIUは、ハード的に切り離される。これによって、システム・ダウンを防止している。あるレベル全体が故障した時には、残りの2レベルを用いてバス・コントロール論理を、再編成してシステムを維持できる。

以上の様に DBC は バス・コントローラを、分散型にした事によりラウンド・ロビンが可能にしており、又、高い信頼性も可能にした。そして、分散型

の場合に問題となるハードウェア量についても、バス・コントロール用ラインの本数を最少限にできたので、従来の方式よりも減少させる事ができた。⁽¹⁰⁾

3.4 C-Bus 上でのフィジカルなデータ転送

本システムにおいては、異ったプロセッサ上に存在するプロセス間で行なわれるコミュニケーションが C-Bus を介したメッセージ交換という形で実現されている。此外に現われるメッセージは、プロセスの処理対象となる生データ(DM) プロセス間通信に必要な IPCM (WRITE, READ, START, STOP 等) の 2 つであり、以下に述べるフィジカルな転送データ・フォーマット、転送方式に従って、C-Bus 上を転送されている。

3.4.1 転送方式

メッセージを、データレize、C-Bus 上で転送する際の転送制御方式、転送単位、C-Bus 使用権の占有単位は、次の様になっている。

- (i) 転送制御方式：異種リソース間の動作速度の違いに影響を受けないようにするために、ハンドシェイク(Data Ready - Data Accept)を使った非同期方式による転送を行なっている。(図.5.)
- (ii) 転送単位：標準コンピュータの語長にあわせて、16ビットをデータ転送単位として、パラレル転送している。
- (iii) データ転送の方法(C-Bus 使用権の占有単位)：バーストで転送しており、1データ毎のC-Bus 要求、獲得、解放は行なっていない。可変長のブロック(max 4 kwords)が、C-Bus 使用権の占有単位である。

3.4.2 メッセージのデブロッキングとブロックフォーマット。

(1) メッセージのデブロッキング

C-Bus のデータ転送は、バーストで行なわれるが、ファイル転送の様な長いメッセージが、長時間 C-Bus を占有すると、より緊急なメッセージの転送が相当待たされることになり好ましくない。この為、長いメッセージは、複数のブロックに分割されて、ブロック単位でバースト転送される。

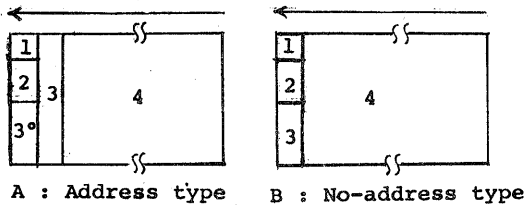
このデブロッキングはBIU内の制御プログラムにより行なわれ、EPは一切タッチしていない。ブロック長については、システムのトラフィックやメッセージの緊急度に応じて、動的に変化させることができる。

(2) ブロックのタイプとフォーマット

C-Bus 上で転送されるデータ・ブロックは、本システムにおけるプロセス間通信の性格から2つのタイプに分類される。

- (a) アドレス・タイプ(生データ)：送信側メイン・メモリ → 受信側メイン・メモリのルートで転送されるデータ・ブロックであり、ブロックのヘッダに受信側メモリ・アドレスが伴なうタイプ。
- (b) 非アドレス・タイプ：IPCM送信側メモリ → 受信側BIUのバッファ(FIFO G1)、あるいは送信側BIUのマイクロプロセッサ → 受信側BIUのバッファ(FIFO G1)のルートで転送されるデータ・ブロックであり、受信側メモリ・アドレスが伴わないタイプ。

一方各ブロックは、転送の際に目的先BIU番号、ブロック・タイプの識別タグと、ともに伴なう受信側メモリ・アドレスが、BIUによりヘッダとして付加され送られる。このヘッダは、受信側BIUの受信制御部で解読され、対応するルートでデータが取りこまれる。図.4 にブロック・フォーマットを示す。



- 1 : tag : 1/0 address type/no-address type
- 2 : destination BIU number
- 3 : address in receiving EP
- 3° : 拡張 address
- 4 : data

図4. Block format.

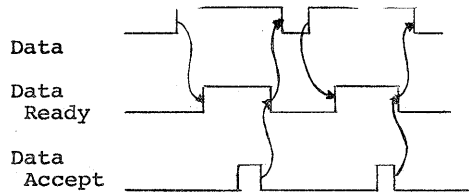


図5. 転送制御のハンドシェイク

3.4.3 多重通信とファイナル・メッセージ(FIM)

本システムのプロセス間通信では、一つのプロセッサでの多重通信を許している。この為、受信側では、メッセージ転送終了時にどのプロセスへの転送が、終わったかを知る必要がある。メッセージが、IPC Mの場合は、その中にプロセス名が入っているため、問題は無いが、生データの場合は、入っていないので、生データとは、別に、この情報を、付加する必要がある。この為、設けられたIPC Mが、ファイナル・メッセージ(FIM)で生データ・メッセージ転送終了後に、該プロセス名の格納されたこのFIMが、送信BIUから送信プロセスへ送られる。

Final Message

F I M	3
source process name	
destination process name	

3.5 BIU (Bus Interface Unit)

BIUは、C-Busと、エレメントプロセッサ間のインターフェースを行なっており、

- 1) C-Busを介するメッセージ転送のフィジカルなサポート
- 2) プロセス間通信におけるロジカルなランデブー・ポートの提供。

という2つの主要な機能を持っている。そして、これらの機能を知的に遂行するために、ユニットの中核制御部にマイクロプロセッサが使われ、BIU全体がスタッド・ロジックで制御されている。

3.5.1 BIUの構成と機能

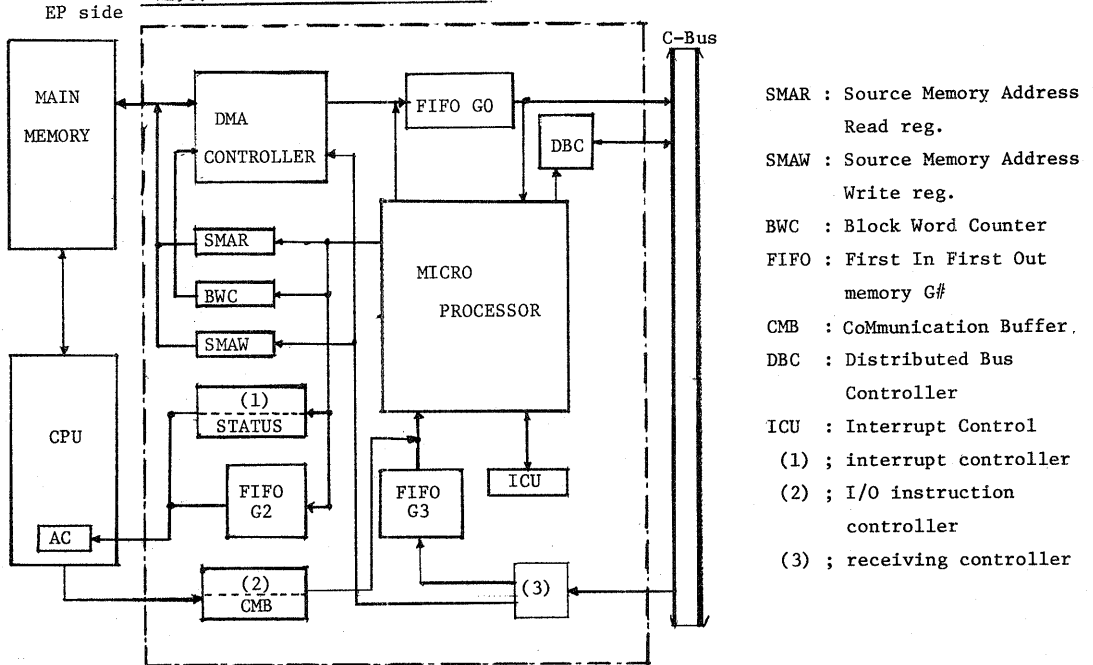
BIUが、果している機能を細かく見ると、それらは、

- a) EPからの入出力命令に対するインターフェース。
- b) EPへの割り込み制御。
- c) DMA制御。
- d) C-Bus送受信制御。
- e) テブロッキング
- f) ランデブー・ポートの提供(転送リンクの確立、抹消)
- g) エラー処理。

であり、こうした処理機能が図6のBIU構成で実現されている。

* BIUで使用されているマイクロプロセッサは、マイクロプログラマブルなビットスライスのパイプラインのマイクロプロセッサであり、BIUの制御に適した効率の良いコントロールとスピードを提供している。

図. 6. BIU (Bus Interface Unit)



- SMAR : Source Memory Address Read reg.
- SMAW : Source Memory Address Write reg.
- BWC : Block Word Counter
- FIFO : First In First Out memory G#
- CMB : CoMmunication Buffer.
- DBC : Distributed Bus Controller
- ICU : Interrupt Control
- (1) ; interrupt controller
- (2) ; I/O instruction controller
- (3) ; receiving controller

表. 2 EP-BIUの間の交換情報

識別	名称	摘要	モード
入出力命令	LIO	コマンド名・BCW語数・優先度・情報手渡し, etc.	プログラム
	SIO	BCWアドレスの手渡しとBIU起動	
	TIO1	割り込情報のリード	
	TIO2	FIFO G2 リード	
	SENSE1	割り込センス	
	SENSE2	CMBの空/塞センス	
割り込	割り込信号	トラップ信号	DMA
	割り込情報	割り込ステータス	
コマンド	BCW	BIUへのコマンド・ワード	DMA

表. 3 Auto Interrupt の割り込み

割り込みレベル	対象
R7	IPL (BIU電源ONの初期設定)
R6	エラー処理要求
R5	C-Bus 送信終了
R4	システムクロック割り込み
R3	IPCM 受信終了
R2	LIO
R1	SIO
R0	BCW READ 終了

(1) EPからの入出力命令及びコマンドの処理

EPとBIUの結合は、入出力バスで行なわれており、両者間で交換される情報は、表. 2に示される。入出力命令の内、SIO、LIO命令はEPのACの内容（BCWアドレス、BCW語数、コマンド名、及び優先度）がCMBにセットされた後、マイクロプロセッサに割り込みがかかる。マイクロプロセッサは、この情報をもとにSMAR、BWCをセットしてDMAによりBCWをMM → FIFO G0 → マイクロプロセッサのルートで読み込む。その他の入出力命令は、ハードウェアで処理される。一方EPへの割り込み情報は、ハードウェア（DMAエラー、BIU電源異常等）とマイクロプロセッサ（メッセージ受信、メッセージ送信終了等）によりセットされ、EPに読み込まれる。

* BCW (BIU Control Words) は、チャネル・コマンド・ワードを機能拡張したものに相当する。

(2) C-Bus への送信.

C-Bus への送信は、基本的に FIFOGR 内のデータを、C-Bus へ出力する事で行なわれるが、FIFOGR にセットされる送信データは、EP の MM から、DMA で転送される場合（生データ・IPCM）と、マイクロプロセッサから、転送される場合（FIM）がある。前者では、マイクロプロセッサが、FIFOGR へのヘッダのセットと SMAR, BWC のセットの後、又 後者では、FIFOGR へのヘッダと転送データのセットの後、DBC へ対して要求が出力される。DBC から許可がおりると送信側制御部が 前者では DMA を起動しながら 後者では、起動せずに C-Bus 送信を行なう。送信が 終了した時、あるいは 異常終了した場合、マイクロプロセッサに割り込みがかかる。（表.3）

(3) C-Bus からのデータ転送

C-Bus 上に先頭のヘッダ・データが流れると、その BIV 番号部分が調べられる。その番号が、自分のものと一致すると次にタグ部分が調べられ、データのルートが 設定され受信が始まる。受信するブロックが アドレス・タイプの場合、ヘッダのメモリ・アドレス部が SMAR にセットされた後、DMA が 起動され受信データが 順次 MM に転送される。受信の終了は、CINR のブロック・エンドで 検出され、DMA を ストップする。

この際、受信側のマイクロプロセッサへの割り込みは 行なわれない。一方、受信するブロックが、非アドレス・タイプの場合には 受信データは FIFOGR1 に転送される。受信の終了は、前者と同じ方法で 検出され、この場合には、受信側マイクロプロセッサに割り込みが、かけられる。

(4) 転送エラーの処理

C-Bus を介したデータ転送中にエラー（パリティエラー、アドレスエラー、等）が生じると、次の様な動作が行なわれる。

- (a) 送信側で検出した場合：受信側 BIV に、ABT 信号を送り、送信を中断して、送信側マイクロプロセッサに割り込む。受信側では受信を中止し受信側マイクロプロセッサに割り込む。
- (b) 受信側で検出した場合：送信側 BIV に、CINS でエラーを知らせ受信を中止する。送信側では、送信を中止し、マイクロプロセッサに割り込む。エラーが検出されるとマイクロプロセッサがブロック単位でのリトライを行なう。

(5) FIFOGR を介した EP-BIU 間の情報転送.

EP への割り込みに関する詳細情報と、EP 内のプロセスに渡さねばならない IPCM は、FIFOGR を介して BIU から EP に渡される。マイクロプロセッサがこれらの情報を、FIFOGR にセットした後、EP に割り込みをかけると EP が FIFOGR から読み込む。読み込み終了は LIO 命令により BIU に知らされ FIFOGR の解放が

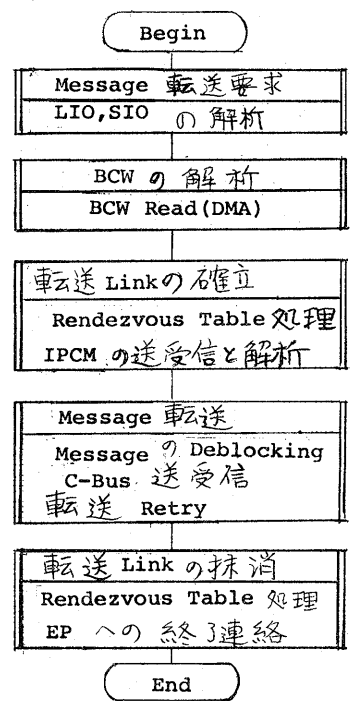


図.7. Flow Chart

行なわれる。

(b) ランデブー・ポートの提供

プロセス間通信におけるランデブー・ポートを実現する為には、マイクロプロセッサのRAMの一部が、ランデブー・テーブルに割り当てられ、このランデブー・テーブル上で、通信要求の登録、通信リンクの確立、抹消が、行なわれている。この処理には BCW と IPCM が、使われる為、他BIUから送られてくる IPCM は C-Bus → FIFOGI → マイクロプロセッサへのルートで、必ずマイクロプロセッサに取り込まれて解読される。

3.6.1. DMの転送メカニクス

DMが転送される場合にBIUで行なわれる処理メカニクスの概略を図.7のフローチャートで示す。

謝辞

最後に、KOCOSの開発設計にあたり御協力、御指導頂いた沖電気工業(株) 杉浦宣紀氏、林茂史氏、松下温氏、遠藤美見氏、本学相磯秀夫教授、清水洋彦氏(現日本航空)に深謝致します。また本論文の作成にあたって協力していただいた阿多靖広君、石井知善君、瀧塚博志君、西尾忠幸君、松尾繁樹君、福島知善君に深謝致します。

参考文献

- [1] 徳田,上林,竹山,石塚,平塚,西垣 : " KOCOSのアーキテクチャ(2)「プロセス間通信方式とソフトウェア構成」". 情報処理学会 計測機アーキテクチャ研究会資料 1974年7月
- [2] 坂井,田畑,大西,北澤 : " イニハウス・コンピュータ・ネットワークとHOSTコンピュータ" 情報処理, VOL.15 NO.12, PP448-454.
- [3] 元岡 : " コンピュータ・コンピュータの展望 ". 情報処理, VOL.15, NO.7 pp525-535.
- [4] Wulf, W. A., Bell, C. G. : C.mmp-A multi-mini-processor, Proc. FJCC, pp.765-777, 1972.
- [5] Farber, D. J., Larson, K. C. : The System Architecture of the Distributed Computer System-The Communications System, Proc. Symp. on Computer-Communications Network and Teletraffic, April, 1972.
- [6] Davis, R. L., Zucker, S., Campbell, C. M. : A Building Block Approach to Multi-processing, Proc. SJCC, pp.685-703, 1972.
- [7] Akkoyunlu, E., Bernstein, A., Schantz, R. : Interprocess Communication Facilities for Network Operating Systems, Computer, vol.7, no.6, pp.45-55, 1974.
- [8] Heart, F. E., et al. : A New Minicomputer/Multiprocessor for the ARPA Network, Proc. NCC, pp.529-537, 1973.
- [9] Thurber, K. J., et al. : A Systematic Approach to the Design of Digital Bussing Structures, Proc. FJCC, pp.719-740, 1972.
- [10] Lochkeed Electronics Company : SUE Computer Handbook, Los Angeles, 1972.

- [11] 飯塚, 藤井, 弓場, 島田 : "新しい計算機システムの調査と評価"
電子技術総合研究所調査報告. 第179号.
- [12] Baskin, H. B., Borgerson, B. R., Roberts, R. : PRIME-A Modular Architecture for Terminal-oriented Systems, Proc. SJCC, pp.431-437, 1972.
- [13] Farber, D. J., Larson, K. C., Loomis, D. C., Rowe, L. A. : The Distributed Computing System, Proc. COMPCON, pp.31-34, 1973.
- [14] Enslow, H. : Multiprocessors and Parallel Processing, A Wiley- Interscience publication, 1974.
- [15] Farber, D. J., Larson, K. C. : The Structure of A Distributed Computing System-Software, Proc. Symp. on Computer-Communications Network and Teletraffic, April, 1972.