

# 1. 加速器の制御システムとは

## 1. 1 加速器制御システムの発展

加速器の制御システムは、ごく初期の小型の加速器にあっても、加速器を構成する本質的な部分であった。そして、近年加速器が巨大化するにつれて、ますます重要となりつつある。

初期の加速器は非常に小さく、構成要素は制御システムを含めて一つの部屋におさまってしまった。この場合は、まとまった制御パネルというより、むしろ機器に付属したスイッチやメータの集合が制御システムをなしていたといえる。

加速器のサイズが大きくなるにつれて、その構成要素は一つの部屋に入りきれなくなり、加速器全体を統一してコントロールするコントロールルームが存在するようになる。このコントロールルームにあるコントロールパネル上のスイッチやメータと遠方にある機器は直接電線で結ばれる。ケーブルの本数はかなり多くなるが、まだ何とかやっていた。

しかし、さらに加速器の規模が大きくなると、何らかのマルチプレクシングが必要となってくる。マルチプレクシングとは、多数の線を何らかの方法で少数の線に多重化することである。加速器の規模の増大がマルチプレクシングを必要としはじめたときは、丁度エレクトロニクスの進歩と小型計算機の実用化の時代にあたり、ここから加速器の計算機制御が始まることになった（この時は同時に工業用計算機制御システムが普及していく時期でもある）。加速器が動き出す当初から計算機による制御を行った巨大加速器の世界最初のもは、1967年に動き始めたLOS ALAMOSのLAMPFである。その後のほとんどすべての巨大加速器は、動き出しの始めから計算機制御をすることがあたりまえとなっていく。また、すでに動いている加速器にも、計算機制御がつけ加えられるようになった。

1974年に運転を開始した、CERNのSPSの制御システムは、(1)小型計算機による分数処理、(2)制御のための応用プログラムの作成を専門のプログラマではなく、加速器の建設、運転そして研究にたずさわる科学者と技術者にゆだねる、という二点で、加速器の制御に新時代をきづいたものといえる。当初マルチプレクサの機能が主であった計算機は、この段階では、加速器を研究するための道具としての性格を強く持つようになった。別のことばでいうと、制御システムの力点はフレキシビリティにおかれるようになったといえる。

この数年のマイクロプロセサの驚異的な発展は、加速器の制御システムに大きな影響を及ぼしつつある。すなわち、(1)制御システム中の計算能力（例えばmips値の）大巾な増大、そして(2)分散の徹底化である。この点からすると現在CERNで建設中のLEPの制御システムは、次世代の加速器制御システムの典型といえるだろう（8. 1節参照のこと）。

## 1. 2 加速器制御システムの機能

これまで説明したように、現在の加速器制御システムは、必然的に計算機制御システムである。そして加速器の制御システムは次のような機能を持たなければならない。

### (1) 機器の遠方からの制御

機器を遠方から制御する、すなわち、機器のonやoff、機器のステータスの読みとり、機器へのパラメータの設定などは、加速器制御の基本である。巨大加速器においては、ほとんどすべての機器が遠方から統一的に制御されなければならない。遠方制御できない機器を一々現場にかけつけて操作することは現実的に不可能である。

### (2) 閉ループ制御

オペレータにより状態を設定された機器は、その状態を実現するべく、閉ループ制御される。この閉ループ制御は通常は、計算機を用いることなく実現される。

### (3) 開ループ制御

開ループ制御とはフィードバックループ中にオペレータが介入する制御と定義される。オペレータはまず測定をし、次に計算をし、そしてその結果何らかの操作を加える。このやり方は、通常は加速器のオペレーションそのものといえる。

測定—計算—結果に基づく操作の一連のつながりを、できるだけ自動化していくと、これは(4)にあげるシュミレーションとモデリングに近づいていく。

### (4) シュミレーションとモデリング

くり返しの早い加速器においては、あるパラメータを動かし、その結果を測定することを繰り返すことによって、最適のパラメータに到達することができる。しかしTRISTANのようなストレージングではこの方式はうまくいかないことが多い。何故ならば、ストレージングではくり返しが極端に遅いからである。

そこでストレージングにおいては、あらかじめ加速器の構造を計算機上にモデル化しておき、多数のパラメータの同時的な変化があるパラメータにどの

ような影響を与えるかを計算機上で予想する手法がとられる。一般にこのようなモデリング計算を短時間で処理するには大きな計算能力が必要となり、何らかの方法で汎用大型計算機またはスーパーミニコンを用いる必要がある。

制御システム内に加速器の構造の数学的モデルを作り加速器のふるまいをシミュレーションする技法は SLAC の SPEAR で開発され、その後ストレージリングでは標準的な技法となっている。

(5) サベイランス

サベイランスは加速器の状態を連続的に監視することである。加速器が巨大になるにつれ、監視しなければならないパラメータは多くなり、計算機による監視は不可欠となる。その際情報はコンパクトにまとめられていなければならないのはいうまでもない。そしてエラー発生時の報告だけでなく、エラーの同定と対処のし方をオペレータに教えられるシステムが理想である。

(6) データの記録

データの記録は、(a)以前に実現していた状態の再現、及び(b)オペレーションの統計に必要となる。

(7) オペレータのガイドとオートメーション

加速器制御システムによる加速器のオペレーションが進展するにつれて、ルーチン化できる部分が多くなる。この部分は、できるだけオートメーション化し、オペレータの負担を軽減することが必要である。

現代の巨大加速器では、構成要素である機器は空間的に広範囲に分散している。このため、加速器に関わる人々にとって、加速器へのインタラクション

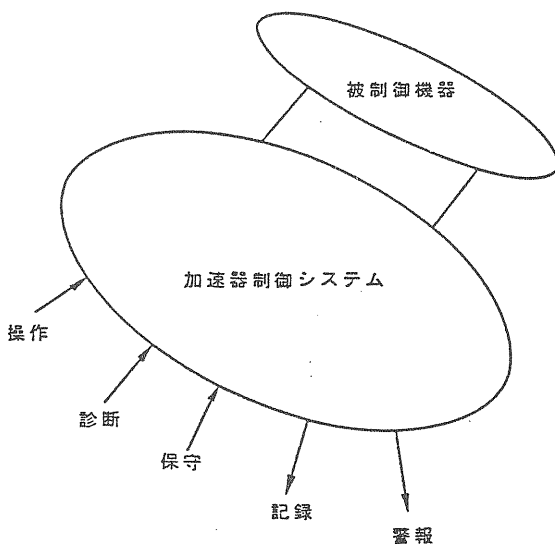


図 1. 1 加速器制御の概念

は計算機化された加速器制御システムを通さなければ不可能となってきた。結局私たちは、制御システムを通してのみ加速器と接していることになる。図 1-1 にこの様子を模式的に示している。加速器を構成する機器へのアクセスは原則として加速器制御システムを通してのものであり、この制御システムによりこの節であげた機能を実現されている。この意味において、制御システム上にいかに正確に加速器のイメージを作り上げるかが、制御システムの良し悪しを決めることになる。

1. 3 インダストリ制御システムと加速器制御システムの比較

加速器の制御システムは、広い意味ではインダストリ制御システムの一つといえるが、両者の間には次のような相異がみられる。

- (1) 加速器の制御システムにおいては、建設前であらかじめ制御システムに対する要請を定めてしまうことができない。すべての加速器は何らかの点で新しさを持っており、建設完了後の研究によりしだいに特性が判明していく。このことは、制御の内容、特にソフトウェアは、運転を開始した後も変更されることを意味する。
- (2) 加速器は物理実験のための道具としての性格を持つ。そこで物理学の発展にともなう研究方法の変化にともない、加速器の使われ方も変化していく。例えば CERN SPS は当初は加速された陽子ビームを外部にとりだして使用する加速器であったが、現在はさらに、陽子-反陽子の衝突実験のためのストレージリングとして使われている。さらに将来は LEP の入射器として電子と陽電子の加速器としても使われる予定である。
- (3) (1)と(2)の特徴から、加速器制御システムに対する最大の要請はフレキシビリティということになる。これはインダストリ制御システムと大きな対照をなす。例えば原子力発電所の制御では一旦つくられた制御システムはソフト及びハードを含めてまず変更されることは有り得ない。

1. 4 加速器制御システムのアーキテクチャ

初期の加速器制御システムでは、計算機のコストが高く、1台の計算機が各機器を制御した。

加速器が巨大化するにつれて、1台の計算機では制御しきれなくなる。そこで中央に大きな計算機を1台、周辺の機器の近くに小型計算機を複数台配置し、中央の計算機と小型計算機を 1 : N に結合する方法が

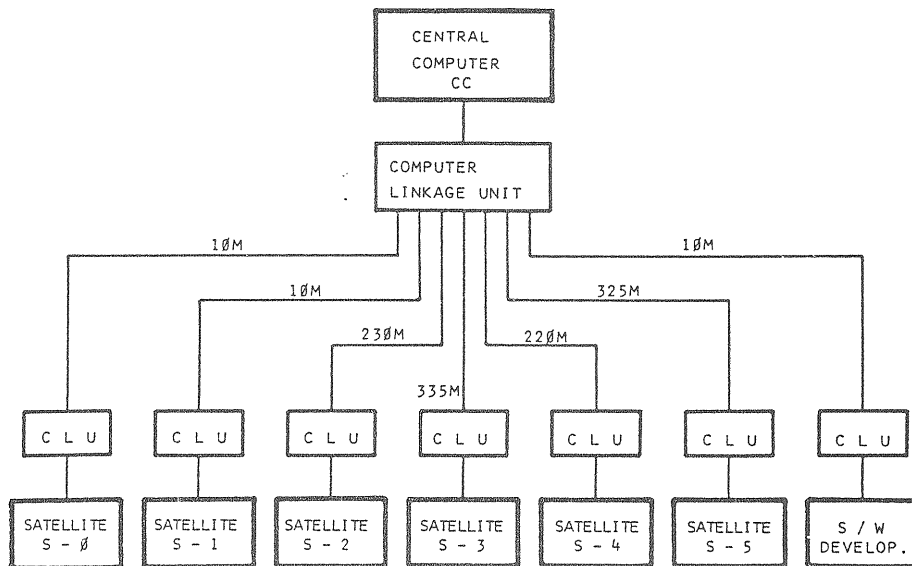


図 1-2 KEK PS 制御システムにおける計算機システムの結合状態。

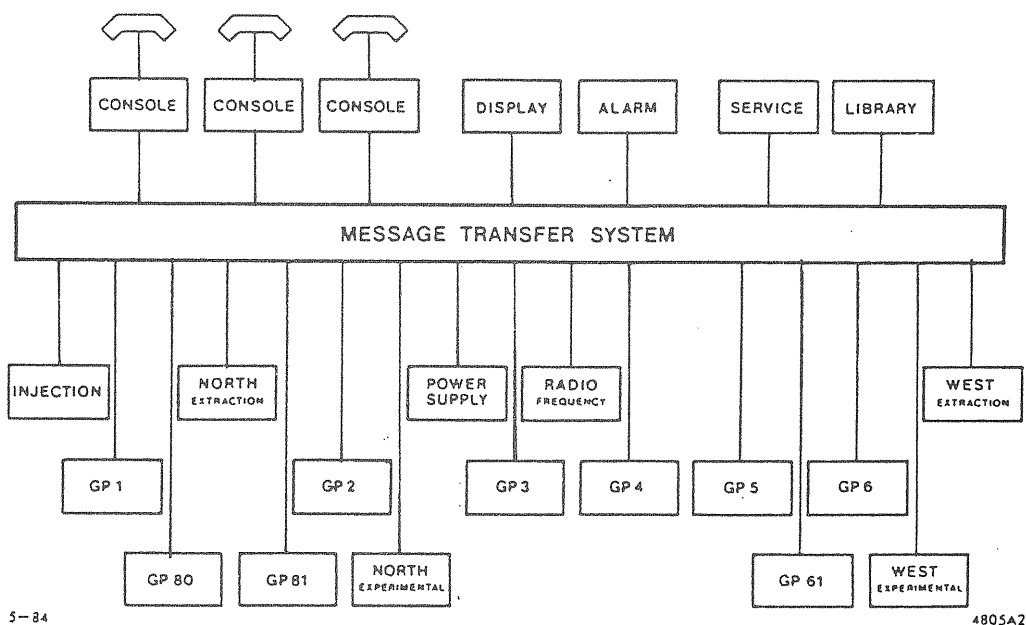


図 1-3 1976 年時点の SPS 制御システム。現在は大巾に拡張されている。

とられるようになった。このようなアーキテクチャの制御システムの例として、図 1-2 に KEK PS 制御システムを示す。このアーキテクチャは構成が単純であるという利点があり、現在でも多くの加速器制御システムに採用されている (SLAC 制御システム, PEP 制御システムなど)。弱点は、中央計算機に負荷が集中し、この計算機的能力により全システムの性能が決ってしまうことにある。

他の有力なアーキテクチャは、CERN SPS 制御に採用された N : N 結合である。このアーキテクチャでは、ほぼ同一の小型計算機が対等な資格で単一のネットワークに結合される。この方式では、図 1-3 の CERN SPS 制御システムの例に示すように、各計算機はそれぞれ分散された機能を持つ。

この例では、1 : N システムの中央計算機が果している役割が、複数台の計算機 (図では、display, alarm, service, library, そして 3 台の console 計算機) によって荷なわれることになる。この方式により、1 : N システムのボトルネックは解消することができる。しかしながら、N : N システムは本質的に複雑なシステムであり、システムの統合をうまくはかってやるのが難しくなる。SPS はこれを NODAL というマルチ計算機言語により解決している。NODAL については、5.2 節と 7.3, 7.4 節で解説する。KEK トリスタン制御システムは、N : N 構成をとる制御システムである。

## 2. 機器とのインターフェイス

計算機制御においては、各被制御機器を計算機に接続しなければならない。そして加速器が巨大化するにつれて、系統的かつ標準的な方法がとられるようになってきた。

機器を計算機に接続する最も単純でかつ効果の大きい方法は、計算機の内部のバスを延長し、それに機器の制御回路を接続することである。初期の制御システムでは（また現在の制御システムでもかなりのものは）個々の計算機メーカーが自分たち独自の方法でバスの延長の方法を定め、独自のモジュールを延長されたバスに接続している。しかしこの方法では、計算機が変わったときは、機器とのインターフェイスを全部やり直さなくてはならず、不便が大きい。

そこで、計算機から独立した規格を定める必要が生じてくる。このような規格の代表的なものが CAMAC 規格である。

### 2. 1 CAMAC

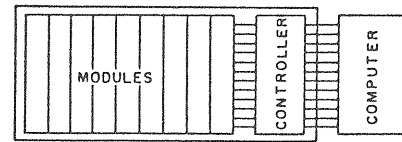
#### 2. 1. 1 CAMAC の概要

CAMAC は当初は、高エネルギー物理学や原子核物理学の実験のデータテキングのために考案された規格である。表 2-1 に CAMAC 規格の一覧を示す

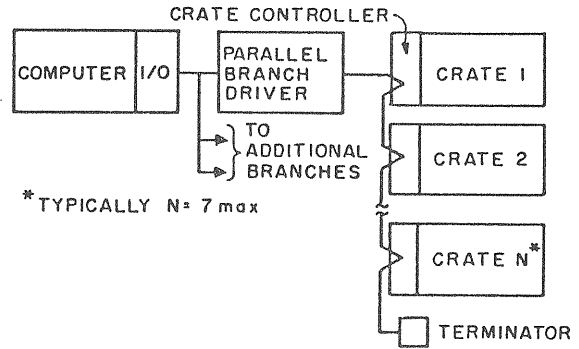
図 2-1 に CAMAC を使う際の三つの代表的な方法、(1)計算機とクレートを直結して用いる場合、(2)クレートをブランチハイウェイに接続する場合、そして(3)クレートをシリアルハイウェイに接続する場合の三つを示してある。極く小規模のシステムを除き、複数台のクレートを計算機に接続する必要が生じる。このとき、クレート及び計算機をパラレルラインで接続するものが(2)、シリアルラインで接続するものが(3)である。シリアルハイウェイについては、3章でデータウェイの一種として説明するので、ここで

表 2. 1 CAMAC の規格

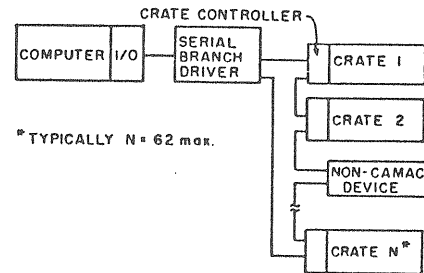
- Modular Instrumentation and Digital Interface System (CAMAC)  
ANSI/IEEE Std 583-1982
- Serial Highway Interface System (CAMAC)  
ANSI/IEEE Std 595-1982
- Parallel Highway Interface System (CAMAC)  
ANSI/IEEE Std 596-1982
- Multiple Controllers in a CAMAC Crate  
ANSI/IEEE Std 675-1982
- Block Transfers in CAMAC Systems  
ANSI/IEEE Std 683-1976 (R1981)
- Real-Time BASIC for CAMAC  
ANSI/IEEE Std 726-1982
- Subroutines for CAMAC  
ANSI/IEEE Std 758-1979 (R1981)



CAMAC System with Controller Communicating with Computer



Parallel Highway System



Serial Highway System

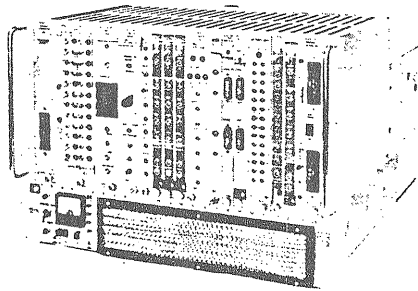
図 2. 1 CAMAC システムの三形態

は、3つの方法に共通する、CAMAC のバスの構造について示す。

CAMAC では、クレートと呼ばれる箱(図 2-2)が規定されている。このクレートには25ヶのステーションがある。1~23のステーションには通常のモジュールが、24及び25ステーションには2の slots を占拠する、クレートコントローラが入る。ステーション25はコントロールステーション、1~24はノーマルステーションと呼ばれる。クレートコントローラの基本的な働きは、計算機から延長されてきたバスを、クレート内部の CAMAC 標準バス(データウェイ)に変換することにある。そこで、計算機から延長されたバスが異なるに従って、そのためのクレートコントローラが必要となる(例えば、LSI-11用には Q バス用クレートコントローラ、PDP-11用には UNIBUS 用クレートコントローラがある)。

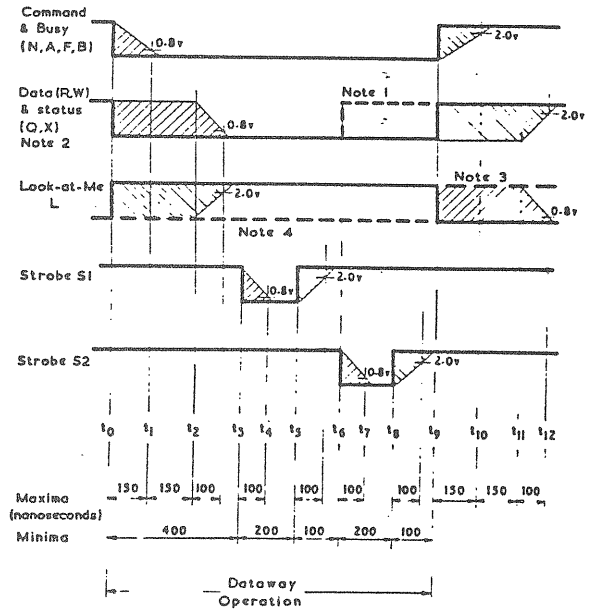
CAMAC のバスはデータウェイと呼ばれる。CAMAC データウェイの特徴を示すと、

- (1) データの巾が24ビット



**CAMAC Crate with Modules from Several Manufacturers**  
(Crate Controller Is at Extreme Right)

⊠ 2. 2



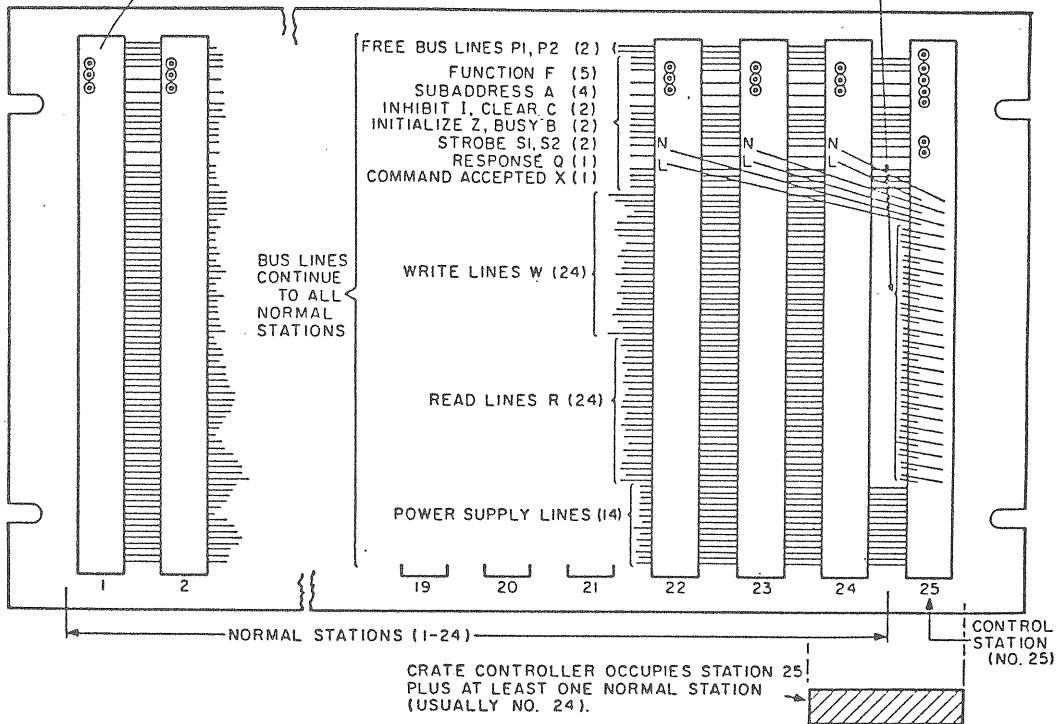
- NOTES:
- (1) Data and status may change in response to S2
  - (2) During some operations Q may change at any time
  - (3) LAM status may be reset during operation
  - (4) L signal may be maintained during operation
  - (5) For all signals the minimum rise or fall time is 10 ns. See 7.1.
  - (6) Signal transition at  $t_0$  or  $t_9$  may be absent if the signals on command or data lines are the same for the immediately preceding or following operations. See 7.1.3.1.

Timing of a Dataway Command Operation

⊠ 2. 3

CIRCLES ⊙ REPRESENT PATCH CONTACTS (3 PER NORMAL STATION, 7 FOR CONTROL STATION).

DIAGONAL LINES ARE N (STATION) AND L (LOOK-AT-ME) LINES BETWEEN CONTROL STATION AND INDIVIDUAL NORMAL STATIONS



Dataway Wiring, Front View of Twenty-Five Station Crate

⊠ 2. 4

- (2) リード (R ライン) とライト (W ライン) が独立している。
- (3) S1とS2というタイミングに同期する。データウェイ上のCAMAC 1回のトランザクションは1  $\mu$ secで終了する。(図2-3に同期のタイムチャートを示す)。
- (4) ノーマルステーションの各々には、スロット25からNラインがそれぞれ1本ずつ接続される。
- (5) 同様に各ノーマルステーションからLラインがそれぞれ1本ずつステーション25に接続される。このラインにより各モジュールからの

LAM (look-at-me) がクレートコントローラに対して教えられる。

- (6) 各モジュールに対する動作を示すために、5本のF(ファンクション)ラインがあり、この5本のラインによって、F=0からF=31までの31ヶの動作が区別できる。
- (7) 各モジュール内部のレジスタを区別するために、4本のA(サブアドレス)ラインがあり、A=0からA=15までの区別が可能である。図2-4に、データウェイの結線の概念を示す。表2-2にデータウェイの標準的な用法を、表2-3にCAMACのノーマルステーションでのコ

表2. 2  
Standard Dataway Usage

Title	Designation	Contacts	Use at a Module
<b>Command Station Number</b>	<i>N</i>	1	Selects the module (individual line from control station)
<b>Subaddress</b>	<i>A1, 2, 4, 8</i>	4	Selects a section of the module
<b>Function</b>	<i>F1, 2, 4, 8, 16</i>	5	Defines the function to be performed in the module
<b>Timing Strobe 1</b>	<i>S1</i>	1	Controls first phase of operation (Dataway signals must not change)
<b>Strobe 2</b>	<i>S2</i>	1	Controls second phase (Dataway signals may change)
<b>Data Write</b>	<i>W1-W24</i>	24	Bring information to the module
<b>Read</b>	<i>R1-R24</i>	24	Take information from the module
<b>Status Look-at-Me</b>	<i>L</i>	1	Indicates request for service (individual line to control station)
<b>Busy</b>	<i>B</i>	1	Indicates that a Dataway operation is in progress
<b>Response</b>	<i>Q</i>	1	Indicates status of feature selected by command
<b>Command Accepted</b>	<i>X</i>	1	Indicates that module is able to perform action required by the command
<b>Common Controls</b>			<i>Operate on all features connected to them, no command required</i>
<b>Initialize</b>	<i>Z</i>	1	Sets module to a defined state (Accompanied by S2 and B)
<b>Inhibit</b>	<i>I</i>	1	Disables features for duration of signal
<b>Clear</b>	<i>C</i>	1	Clears registers (Accompanied by S2 and B)
<b>Nonstandard Connections</b>			
<b>Free bus lines</b>	<i>P1, P2</i>	2	For unspecified uses
<b>Patch contacts</b>	<i>P3-P5</i>	3	For unspecified interconnections. No Dataway lines
<b>Mandatory Power Lines</b>			<i>The crate is wired for mandatory and additional lines</i>
<b>+24 V dc</b>	<i>+24</i>	1	
<b>+ 6 V dc</b>	<i>+ 6</i>	1	
<b>- 6 V dc</b>	<i>- 6</i>	1	
<b>-24 V dc</b>	<i>-24</i>	1	
<b>0 V</b>	<i>0</i>	2	Power return
<b>Additional Power Lines</b>			<i>Lines are reserved for the following power supplies:</i>
<b>+12 V dc</b>	<i>+12</i>	1	
<b>-12 V dc</b>	<i>-12</i>	1	
<b>Clean Earth</b>	<i>E</i>	1	Reference for circuits requiring clean earth
<b>Supplementary -6 V</b>	<i>Y1</i>	1	See Section 10.
<b>Supplementary +6 V</b>	<i>Y2</i>	1	See Section 10.
<b>Reserved undesignated</b>		3	
<b>Total</b>		<b>86</b>	

ネクタの各ピンの役割, 表2-4にコントロールステーションのコネクタの各ピンの役割を示す。また表2-5に, ファンクション F=0 から F=31の役割の定義を示す。

CAMACにおいては, あるモジュールに対する何らかの動作を定めるには, (1)ステーション N, (2)サブアドレス A, (3)ファンクション Fを指定しなければならない。データウェイ上に NAFが送出されると各モジュールは所定の動作を行うことになる。もし Fによる指定が0から7まで(リード)かあるいは16から23のとき(ライト)は, S1のタイミングで, データが R1~R24または W1~W24上になる。

る。

このように, CAMACの動作には Nが必ずともなう。Nを送出できるのは, ステーション25であるから, すなわち, 動作の主導権をにぎるのはクレートコントローラということになる。クレートコントローラは計算機のパスを CAMACデータウェイに変換する役割をしているだけであるので, CAMACシステムは結局計算機主導のシステムということになる。

## 2. 1. 1 CAMAC に対する批判

CAMACは, 規格が比較的単純であり, またファ

表 2. 3

### Contact Allocation at a Normal Station (Viewed From Front of Crate)

Bus line	Free Bus line	P1	B	Busy	Bus line
Bus line	Free Bus line	P2	F16	Function	Bus line
Individual patch contact		P3	F8	Function	Bus line
Individual patch contact		P4	F4	Function	Bus line
Individual patch contact		P5	F2	Function	Bus line
Bus line	Command Accepted	X	F1	Function	Bus line
Bus line	Inhibit	I	A8	Subaddress	Bus line
Bus line	Clear	C	A4	Subaddress	Bus line
Individual line	Station Number	N	A2	Subaddress	Bus line
Individual line	Look-at-Me	L	A1	Subaddress	Bus line
Bus line	Strobe 1	S1	Z	Initialize	Bus line
Bus line	Strobe 2	S2	Q	Response	Bus line
Twenty-four Write Bus lines		W24	W23		
W1 = least significant bit		W22	W21		
W24 = most significant bit		W20	W19		
		W18	W17		
		W16	W15		
		W14	W13		
		W12	W11		
		W10	W9		
		W8	W7		
		W6	W5		
		W4	W3		
		W2	W1		
Twenty-four Read Bus lines		R24	R23		
R1 = least significant bit		R22	R21		
R24 = most significant bit		R20	R19		
		R18	R17		
		R16	R15		
		R14	R13		
		R12	R11		
		R10	R9		
		R8	R7		
		R6	R5		
		R4	R3		
		R2	R1		
Power Bus lines	-12 V dc	-12	-24	-24 V dc	
	Reserved (c)*		-6	-6 V dc	
	Reserved (a)*	-	-	Reserved (b)*	
	Supplementary -6 V**	Y1	E	Clean Earth	
	+12 V dc	+12	+24	+24 V dc	
	Supplementary +6 V**	Y2	+6	+6 V dc	
	0 V (Power Return)	0	0	0 V (Power Return)	

\*Reserved (c) was previously assigned to +200 V dc, Reserved (a) to 117 V ac Line, and

Reserved (b) to 117 V ac neutral, all non-mandatory voltages.

Effective 1 January 1978, these assignments were cancelled in order to avoid hazardous voltages on the connectors.

\*\*See Section 10.

ンクションによる指定という人間的なところもあって、高エネルギー物理学の実験や加速器の制御システム、さらに核融合の研究装置などに広汎に使われてきた。しかし、CAMAC 規格の制定は1960年代にさかのぼり、現時点からみるとやや時代遅れの面がでてきた。CAMAC に対する主な批判には次のようなものがある。

- (1) データ巾の24は中途半ばであり、現代の主流的な小型計算機やマイクロコンピュータのデータ巾である16や32ビットと適合しない。

- (2) モジュールとデータウェイの接続用のカードエッジコネクタは信頼性が低く、故障が多い。  
 (3) シングルホスト構成のため、現代の進歩したマイクロプロセサの能力が使いにくい。

上記の批判を克服できるバスシステムはいろいろ考えられている。そのうちの代表的なものとして、VMEバスを次節でとりあげる。

## 2. 1. 2 CAMAC モジュールの例

よく使われる CAMAC モジュールは(1)ディジタ

表 2. 4

Contact Allocation at the Control Station  
(Viewed From Front of Crate)

Individual patch contact		P1	B	Busy	Bus line
Individual patch contact		P2	F16	Function	Bus line
Individual patch contact		P3	F8	Function	Bus line
Individual patch contact		P4	F4	Function	Bus line
Individual patch contact		P5	F2	Function	Bus line
Bus line	Command Accepted	X	F1	Function	Bus line
Bus line	Inhibit	I	A8	Subaddress	Bus line
Bus line	Clear	C	A4	Subaddress	Bus line
Individual patch contact		P6	A2	Subaddress	Bus line
Individual patch contact		P7	A1	Subaddress	Bus line
Bus line	Strobe 1	S1	Z	Initialize	Bus line
Bus line	Strobe 2	S2	Q	Response	Bus line
Twenty-four individual Look-at Me lines (L1 from Station 1, etc)		L24	N24	Twenty-four individual Station Number lines, (N1 to Station 1, etc)	
		L23	N23		
		L22	N22		
		L21	N21		
		L20	N20		
		L19	N19		
		L18	N18		
		L17	N17		
		L16	N16		
		L15	N15		
		L14	N14		
		L13	N13		
		L12	N12		
		L11	N11		
		L10	N10		
		L9	N9		
		L8	N8		
		L7	N7		
		L6	N6		
		L5	N5		
		L4	N4		
		L3	N3		
		L2	N2		
		L1	N1		
Power Bus lines	-12 V dc	-12	-24	-24 V dc	
	Reserved (c)*		-6	-6 V dc	
	Reserved (a)*	—	—	Reserved (b)*	
	Supplementary -6 V**	Y1	E	Clean Earth	
	+12 V dc	+12	+24	+24 V dc	
	Supplementary +6 V**	Y2	+6	+6 V dc	
	0 V (Power Return)	0	0	0 V (Power Return)	

\*Reserved (c) was previously assigned to +200 V dc,

Reserved (a) to 117 V ac Line, and

Reserved (b) to 117 V ac Neutral, all non-mandatory voltages.

Effective 1 January 1978, these assignments were cancelled in order to avoid hazardous voltages on the connectors.

\*\*See Section 10.



表 2. 5  
The Function Codes

Code F( )	Function	Use of R and W Lines	Function Signals					Code F( )
			F16	F8	F4	F2	F1	
0	Read Group 1 register	Functions using the R lines	0	0	0	0	0	0
1	Read Group 2 register		0	0	0	0	1	1
2	Read and Clear Group 1 register		0	0	0	1	0	2
3	Read Complement of Group 1 register		0	0	0	1	1	3
4	Nonstandard		0	0	1	0	0	4
5	Reserved		0	0	1	0	1	5
6	Nonstandard		0	0	1	1	0	6
7	Reserved		0	0	1	1	1	7
8	Test Look-at-Me	Functions not using the R or W lines	0	1	0	0	0	8
9	Clear Group 1 register		0	1	0	0	1	9
10	Clear Look-at-Me		0	1	0	1	0	10
11	Clear Group 2 register		0	1	0	1	1	11
12	Nonstandard		0	1	1	0	0	12
13	Reserved		0	1	1	0	1	13
14	Nonstandard		0	1	1	1	0	14
15	Reserved		0	1	1	1	1	15
16	Overwrite Group 1 register	Functions using the W lines	1	0	0	0	0	16
17	Overwrite Group 2 register		1	0	0	0	1	17
18	Selective Set Group 1 register		1	0	0	1	0	18
19	Selective Set Group 2 register		1	0	0	1	1	19
20	Nonstandard		1	0	1	0	0	20
21	Selective Clear Group 1 register		1	0	1	0	1	21
22	Nonstandard		1	0	1	1	0	22
23	Selective Clear Group 2 register		1	0	1	1	1	23
24	Disable	Functions not using the R or W lines	1	1	0	0	0	24
25	Execute		1	1	0	0	1	25
26	Enable		1	1	0	1	0	26
27	Test Status		1	1	0	1	1	27
28	Nonstandard		1	1	1	0	0	28
29	Reserved		1	1	1	0	1	29
30	Nonstandard		1	1	1	1	0	30
31	Reserved		1	1	1	1	1	31

表 2. 6 TRISTAN 制御に用いられる標準 CAMAC  
モジュール

SAD : 32-channel	Sample-and-hold A D C
DAD : 32-channel	Dual-slope A D C
DAC : 8-channel	D/A converter
SIG : 32bit	Status input gate
IIR : 16bit	Interlock input register
SOR : 16bit	Status output register
AIO : 16bit	Active I/O
PIO : 16bit	Passive I/O
PTG : 4-channel	Pulse train generator

ル入力, (2)デジタル出力, (3)アナログ入力, (4)アナログ出力モジュールに分類される。表 2-6 に、TRISTAN 制御で用いられている CAMAC モジュールの一覧を示す。また図 2-5 に CAMAC モジュールの写真を示す。

## 2. 2 VME バス

前節であげた CAMAC に対する批判を乗り越え

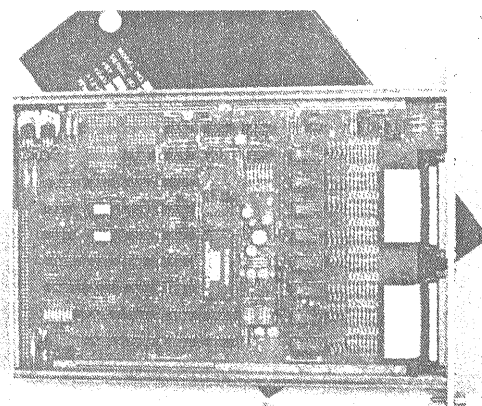


図 2. 5 CAMAC モジュールの例

るものとして、VME が注目されている。

VME バスは1979年にモトローラのバーサバスに始まり、1981年にはモトローラ、モステック、シグネティックスの3社によってVMEバス・マニファクチュア・グループが結成され、1983年にはIEEE P1014, IEC4713でVMEバスの標準化が始ま

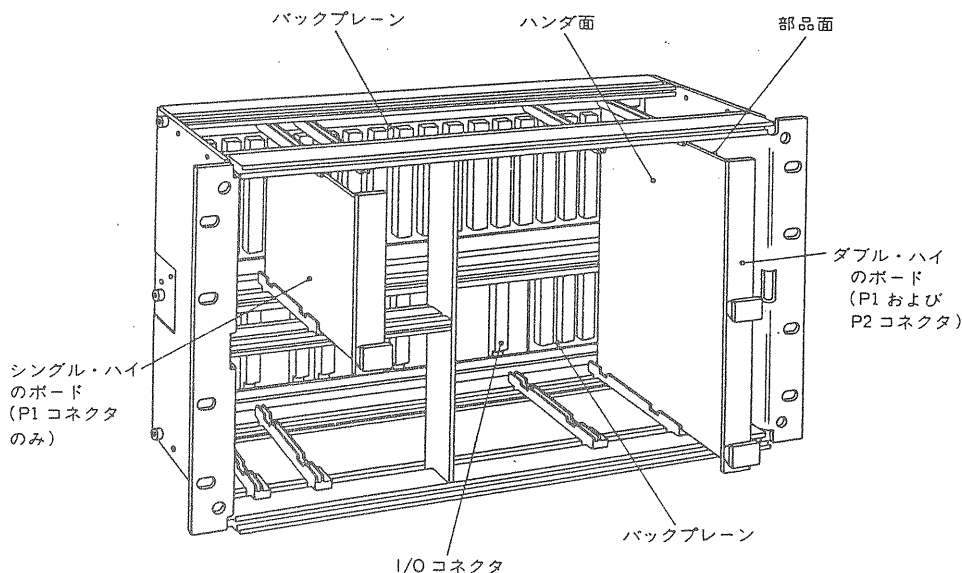


図 2. 6 VME のラック

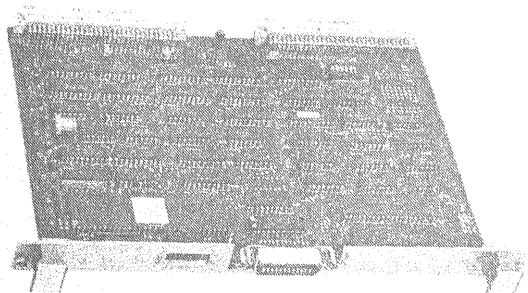


図 2. 7 VME モジュールの例

っている。1984年には世界で約80社が VME ボードの開発を行っており、日本でも10数社が VME ボードを作っている。

VME の特徴として

- (1) データ巾は 8, 16 または 32 ビット。
- (2) アドレス巾は 8, 24 または 32 ビット。
- (3) データ巾とアドレス巾はマスタによりデータ転送サイクルごとに指定される。
- (4) データ転送は非同期である。
- (5) 20 マスタまで制御できるバス調停機構を持つ。
- (6) 96 ピンの DIM コネクターを使用して信頼性が高い。

があげられる。

図 2-6 に VME バスのラックの構造を示す。図 2-7 に VME ボードの写真を示す。いずれにしても、VME はマルチマイクロプロセサに適したバスとして今後大いに発展するものと考えられる。

### 3. データハイウェイ

データハイウェイは計算機のバスを長く延長した

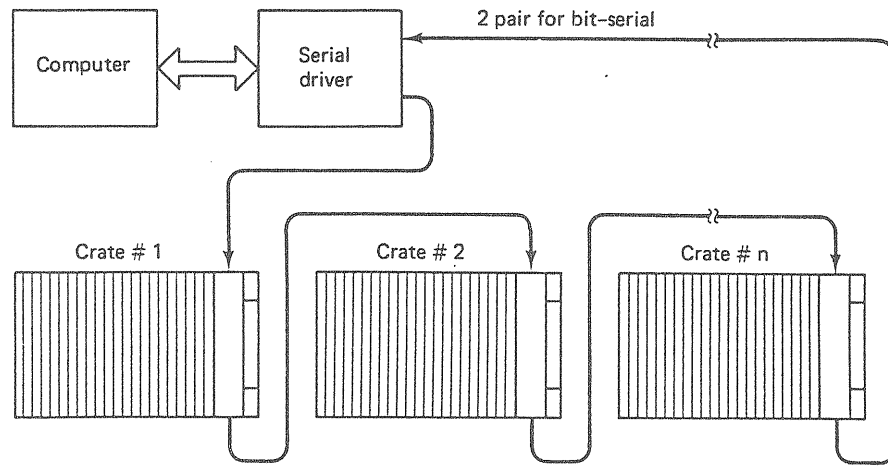
ものといえる。2章でとりあげた CAMAC のブランチハイウェイとシリアルハイウェイは、データハイウェイの例である。以下にはまず CAMAC シリアルハイウェイについて説明をする。そして、CAMAC シリアルハイウェイの弱点を克服するデータハイウェイの代表例として PROWAY について説明する。PROWAY は、4章で考えるローカルエリアネットワーク (LAN) の一つと考えることもでき、データハイウェイとローカルエリアネットワークの中間的なものといえる。

#### 3. 1 CAMAC シリアルハイウェイ

##### 3. 1. 1 CAMAC シリアルハイウェイの概要

前に示したように、計算機に複数台の CAMAC クレートを接続するには、計算機のバスを何らかの方法で延長しなければならない。もし、このとき延長距離が 100m から 1 km 程度になると、パラレルラインではコスト的に引き合わなくなり、シリアル転送方式をとらざるを得ない。CAMAC シリアルハイウェイは、最大 62 台のクレートを接続でき、また計算機一クレート、クレート一クレート間の距離を原理的にはどこまでものばすことができる規格であり、大規模な応用に適したデータハイウェイといえる。CAMAC シリアルハイウェイには、ビットシリアルとバイトシリアルのものがある。前者は 1 ペアのデータと 1 ペアのクロックの計 2 ペアのケーブルより成り立つデータハイウェイであり、後者は 8 ペアのデータと 1 ペアのクロックの計 9 ペアのケーブルから成り立っている。

CAMAC シリアルハイウェイにおいては、計算機



Standard CAMAC serial highway.

図 3. 1

		MSB						LSB			
		8	7	6	5	4	3	2	1		
b	0	SC32						SC1		HEADER BYTE	
b	0	0	0	SAB		SA1				BYTE 2	
b	0	1	SF16				SF1				BYTE 3
b	0	1	SN16				SN1				BYTE 4
b	0	SW24						SW19		BYTE 5 *	
b	0	SW18						SW13		BYTE 6 *	
b	0	SW12						SW7		BYTE 7 *	
b	0	SW6						SW1		BYTE 8 *	
b	0	c	c	c	c	c	c			SUM BYTE	
1	0	1	1	1	1	1	1			SPACE BYTES AS REQUIRED	
1	0	1	1	1	1	1	1				
1	1	1	0	0	0	0	0			END BYTE	

Command Message: Bit Assignments

- b odd byte-parity bits
- c Even column-parity bits
- \* Bytes 5, 6, 7, 8 included if SF16 = 1 and SF8 = 0

図 3. 2

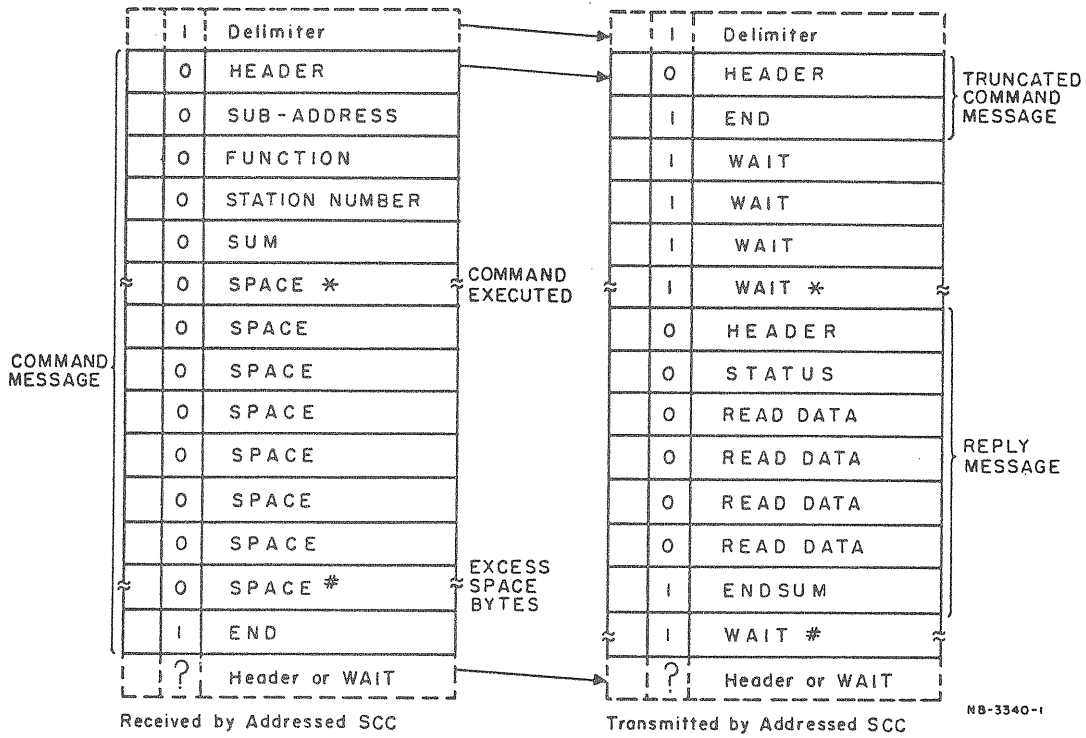
のバスはシリアルドライバーによりシリアルハイウェイに変換される。シリアルハイウェイは図 3-1 に示すようにループであり、シリアルドライバーから出たシグナルは、各クレーットの24と25ステーションに入れられたシリアルクレーットコントローラ (SCC) を一順して再びシリアルドライバーに戻ってくる。

シリアルハイウェイ上の信号は、バイト毎にまとまった意味を持って流れるバイトストリームである。アイドル状態では、wait byte と呼ばれる

(11100000) というパターンが流れている。

計算機はシリアルハイウェイに接続されているクレーット内のモジュールにアクセスするために、C (クレーット番号), N, A, F を指示しなければならない。また、ファンクションがライトのときは、24ビットのデータを与えなければならない。これらの情報は図 3-2 に示すようなバイトの集まりで与えられる。これらのバイトのうち第7ビットは、該当のバイトがデリミタバイト (delimiter byte) かそうでないかを示す。デリミタバイトのときはこのビットが1、その他の時は0となる。先にあげたウエイバイトの他にこの図中のエンドバイト (end byte) もデリミタバイトの一種である。第8ビットはこのバイトが全体でオッドパリティになるようにするためのビットである。これらの第7と第8ビットを除く残りの6ビットが情報を伝える。1バイト中のパリティビットの他、これらのバイトのたて方向のパリティをイーブンにするためのコラムパリティ (column parity) がサムバイト (sum byte) に存在している。

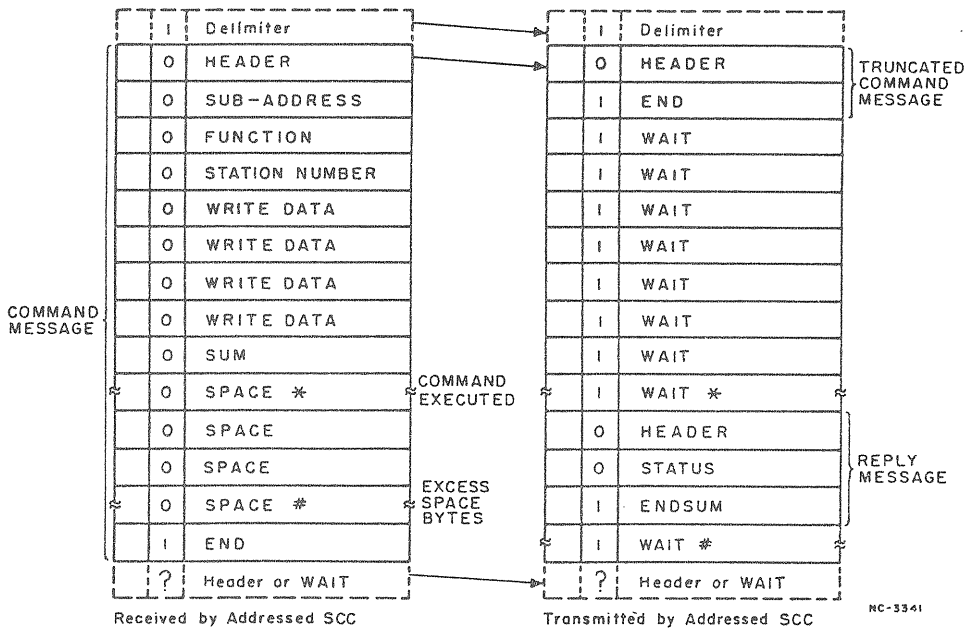
CAMAC シリアルハイウェイのコマンドリプライシーケンスを図 3-3 (リードオペレーション) と図 3-4 (ライトオペレーション) に示す。リードの場合には、コマンドメッセージ中のスペースバイト (space byte) とエンドバイトがヘッダ、ステータス、そして4バイトのリードデータとエンドサムバイトに書き直される。同時に A, F, N を含むバイトとエンドサムバイトは、エンドバイトとウエイバイトに書き直される。結局コマンドメッセージ (図 3-5) は、トランケーティドコマンドメッセージ (図 3-6) とリプライメッセージ (図 3-7) に書き直されたことになる。



Command/Reply Sequence: Read Operation, Bit-Serial Mode

- \* Number of bytes as required to accommodate execution of command: minimum number 0
- # Number of bytes as required to accommodate excess SPACE bytes: minimum number 0

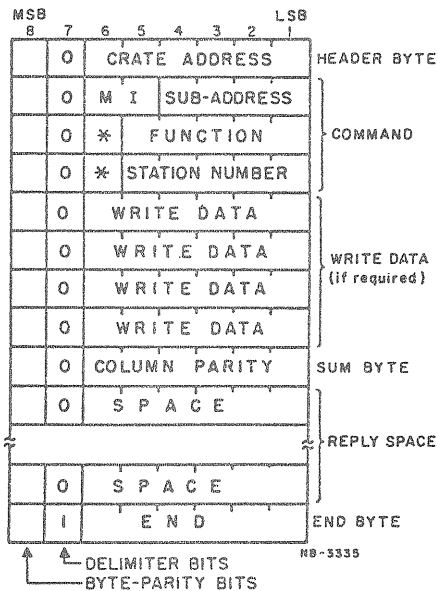
图 3. 3



Command/Reply Sequence: Write Operation, Bit-Serial Mode

- \* Number of bytes as required to accommodate execution of command: minimum number 0
- # Number of bytes as required to accommodate excess SPACE bytes: minimum number 0

图 3. 4



Command Message: Field Assignments

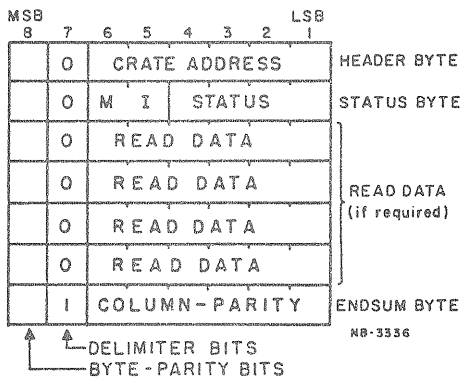
\* Reserved bits  
MI Message Identifier Field

図 3. 5



Truncated Command Message:  
Field Assignments

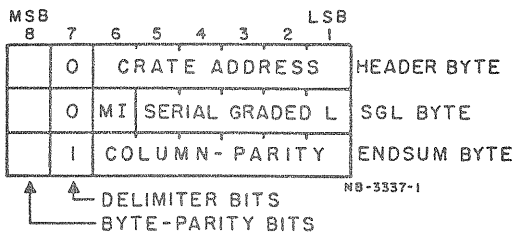
図 3. 6



Reply Message: Field Assignments

MI Message Identifier Field

図 3. 7



Demand Message: Field Assignments

図 3. 8

### Direct D-Port Interconnection

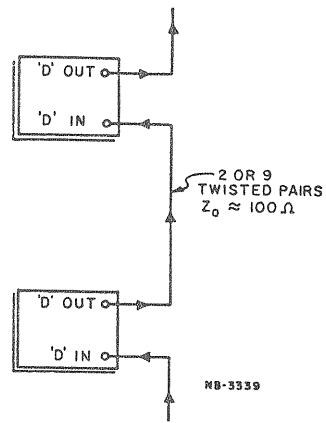
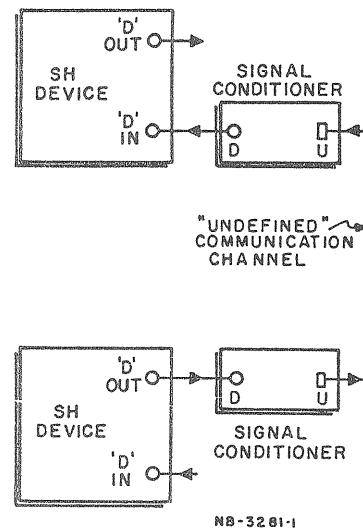


図 3. 9



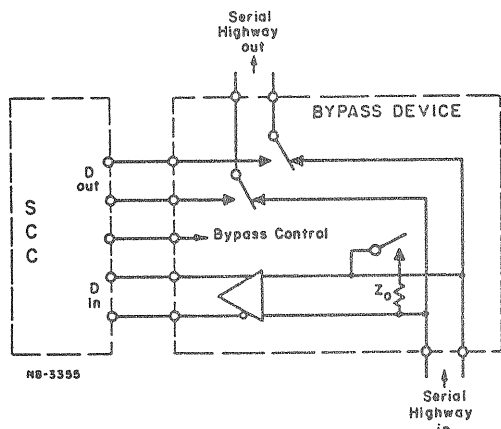
### Indirect Connection Via Undefined Standard

図 3. 10

以上のように CAMAC シリアルハイウェイでは命令とそれに対する応答がほとんど1メッセージ内で行われる。このような形式はコマンドリフライ方式と呼ばれ、応答の速さが特徴である。1クレート内での S1, S2 タイミングによる CAMAC サイクルが、シリアルハイウェイ上に自然に拡張されることが分かると思う。

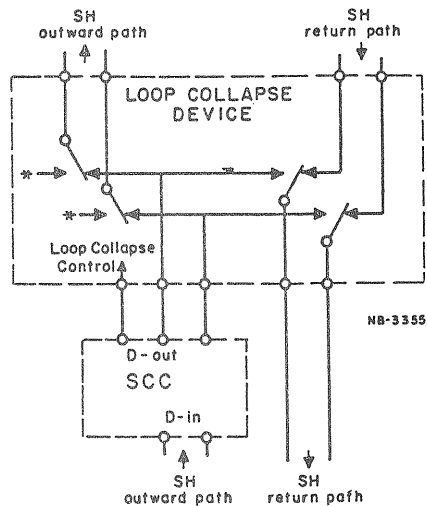
1クレート内の LAM に対応して、シリアルハイウェイでは、デマンドメッセージが定義されている(図 3-8)。モジュールからの LAM があると、SCC はシリアルハイウェイ上に 3 バイトのデマンドメッセージを送出する。デマンドメッセージには、クレートアドレスと、エンコードされた LAM のパターンがのせられる。

シリアルハイウェイの入出力の標準は D ポート(D は defined を意味する)として定義されている。D ポート間は 100Ω のインピーダンスを持つツイステッドペア線にて結ばれる(図 3-9)。D ポートによ



**Example of Bypass Switching for One D-Port Signal**

[In a Bypass Device for Bit-Serial D-Port Signals This Arrangement is Duplicated for Data and Clock Signals]



**Example of Loop Collapse Switching for One D-Port Signal**

[In a Loop Collapse Device for Bit-Serial D-Port Signals This Arrangement is Duplicated for Data and Clock Signals]

[The Clock Signal Output to the Disconnected Part of the SH Loop is Held in a Fixed Logic State by the Conditions at \*-\*]

図 3. 11

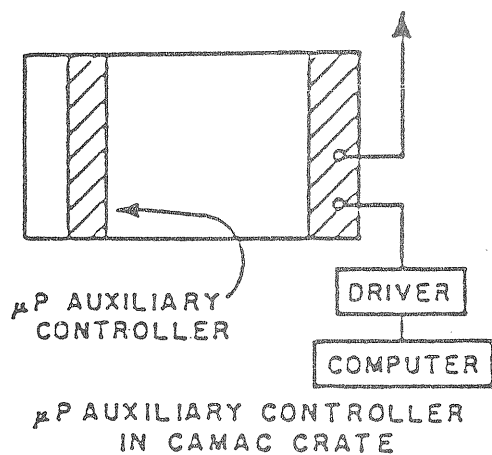


図 3. 12

る接続では、高々100m 程度の距離しかカバーできず、また D ポートではお互いを絶縁できない。そこで、Dポートの信号をシグナルコンディショナー(Uポートアダプタと呼ぶ)を通してUポート(Uはundefinedを意味する)に変えることがよく用いられる。Uポート間を結ぶケーブルには同軸ケーブルや光ファイバケーブルが用いられることが多い(例えば TRISTAN 制御では同軸ケーブルを用いて2.5Mbps を達成している)(図 3-10)。

シリアルハイウェイは、ループであるので、ループの途中のどれかの SCC が故障してしまうと全システムのダウンとなる。そこで、ある SCC のバイパス (by-pass) とループを途中で短絡してしまうコ

ラプス (collapse) の機能が備わっている (図 3-11 参照)。

### 3. 1. 2 シリアルハイウェイの速度

シリアルハイウェイ上の伝送速度は、最高 5 Mbps と規定されている。前に説明したように、1 回のリードまたはライトは少なくとも12バイトのデータをシリアルハイウェイ上に転送しなければならないから

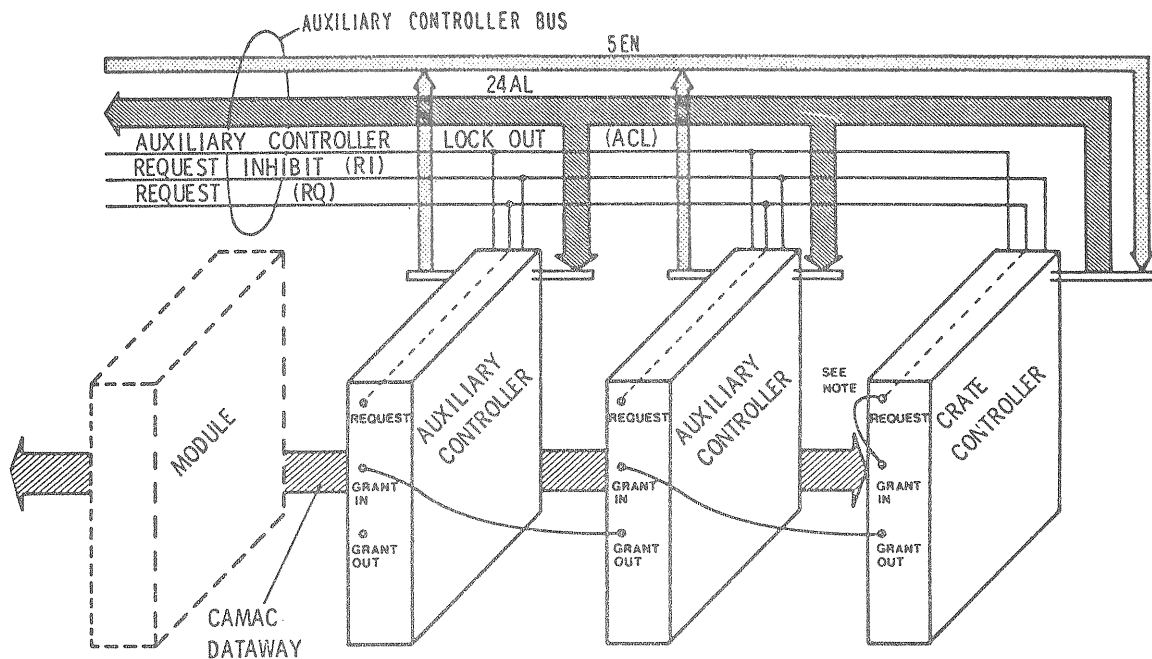
$$12 \times \frac{1}{5 \times 10^6} = 2.4 \mu\text{sec} \text{ (バイトシリアル)}$$

$$12 \times 10 \times \frac{1}{5 \times 10^6} = 24 \mu\text{sec} \text{ (ビットシリアル)}$$

が1回のリードまたはライトに要する最小時間となる。2番目の式で10がかかっているのは、ビットシリアルモードにおいては、1バイトの転送に10ビットを要する(8ビット以外に1ケのスタートビットと1ケのストップビットが必要)からである。いずれにしても、これらの速度は、通常の加速器の機器にとって十分速いものである。

### 3. 1. 3 クレート内の補助的な CPU

CAMAC シリアルハイウェイの規格は、クレート内に補助的なクレートコントローラを入れることを可能としている (図 3-12)。2章で説明したよう



NOTE: FRONT PANEL REQUEST TO GRANT-IN CONNECTION MADE AT HIGHEST PRIORITY CONTROLLER USING R/G PROTOCOL

Multiple Controllers in a CAMAC Crate

図 3. 13

に、CAMACにおいては、NとLラインはコントロールステーション（N=25）のみに接続されている。モジュールにアクセスするには必ずNラインを操作しなければならず、またモジュールからLAMを出すにはLラインを経由しなければならないから、N=25以外に入っているモジュールは他のモジュールをアクセスできないことになる。そこでシリアルハイウェイの規格では、SCCの背面にSGLエンコーダコネクタがとりつけられており、このコネクタを通して、ノーマルステーションのモジュールがN=25のNラインとLラインにアクセスができるようになっている。

さて2つ以上のコントローラが同一のクレート内に存在すると、同時にデータウェイへのアクセスが生じたときどうするかを解決しなければならない。シリアルハイウェイでは、このためにACL (auxiliary controller lockout) というラインがあり、SCCがデータウェイをアクセスしているときはACLが1になり他のコントローラはデータウェイへのアクセスが禁止される。(図3-13参照)。

### 3. 1. 4 CAMACシリアルハイウェイの弱点

CAMACシリアルハイウェイは、高速かつ高レスポンス、長距離がカバーできるという長所を持っているが、次にあげるような弱点がある。すなわちシリアルハイウェイは1ホストのシステムであり、クレート内のCPUがデマンドを送出する以外の方法

でシリアルハイウェイにアクセスできない。”

近年のマイクロプロセサの進歩にともない、この点を克服するデータハイウェイへの要求が高まってきた。そのような試みの一つとして次節ではPROWAYについて説明しよう。

### 3. 2 プロウェイ (PROWAY)

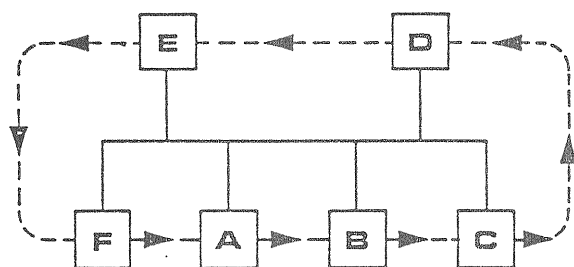
PROWAYは工業用データハイウェイの規格として1975年からIEC(国際電気標準会議)で議論が続けられてきている。ここで説明するのはそのうちのPROWAY-Bと呼ばれる規格である。PROWAYにはその他にPROWAY-Cという規格も検討されている。表3-1にPROWAYの概略仕様を示す。PROWAYは伝送路として75Ωの同軸ケーブルを用いたバス型のトークン方式のシステムである。

バス型のシステムではあるステーション(伝送路に接続されたコンピュータなど)から送出された信号は伝送路を通してすべてのステーションに伝播していく。つまりあるステーションからの情報は伝播遅れの範囲内ですべてのステーションに伝えられる。これでは2ヶ以上のステーションが同時に伝送路にアクセスしたときに混乱が生じるので、何らかの方法がとられなければならない。最も一般的に用いられているのは、(1)CSMA方式と(2)トークン方式である。(1)のCSMAについては4章で扱う。(2)のトークン方式がPROWAYの採用している方式である。

伝送路上にトークンと呼ばれ特定のメッセージが

表 3. 1  
PROWAYの概要

1	トポロジー	バス形
2	伝送路	75 Ω同軸ケーブル(トランク/ドロップ構成)
3	伝送距離	最大 2 Km
4	伝送速度	1 Mbps
5	ステーション数	最大 100
6	コーディング方式	マンチェスタ・コーディング
7	変調方式	位相連続FSK (6.25 MHz/3.75 MHz)
8	ラインアクセス 制御方式	PROWAY A/B:バトンパッシング方式 PROWAY C:トークパッシング(IEEE 802.4)方式
9	フレーム構成	PROWAY A:HDLC形フレーム(8ビットアドレス) PROWAY B:IEC/TC 57形フレーム(8ビットアドレス) PROWAY C:IEEE 802.4形フレーム(16ビットアドレス)
10	誤り制御	Immediate ACK監視による再送回復
11	見逃し誤り率	$3 \times 10^{-15}$ 以下
12	ワーストケース アクセスタイム	PROWAY A/B:20ms以下 PROWAY C:50ms以下
13	ユーザサービス (②③④はオプション)	① SDA(Send Data with Acknowledge) ② SDN(Send Data with No Acknowledge) ③ RDR(Request Data with Reply) ④ RSR(Remote Station Recovery) ⑤ MOP(Management of Local PROWAY)
14	ライブラリスト	PROWAY内でサポートされる
15	カップラインタフェース	RS-422/37ピンコネクタ(ISO 4902)
16	信号絶縁方式	トランス絶縁
17	冗長化構成	可能(ネットワーク層制御によるPROWAY完全2重化)



A Logical Ring on a Physical Bus

図 3. 14

順回している。バス上にはトークンの順回を定める自然な順序というものはないので、論理的にステーションに番号付けをして、この番号順にトークンは

順回することになる。メッセージを送りたいステーションは、まずトークンを獲得してからでないと送信ができない。つまりトークンを介して複数のステーションの間の衝突がさげられていることになる。(図3-14)。

PROWAYの具体的な情報伝達方法はSDA (send data with acknowledge) であり、送信ステーションからのデータを受信ステーションが受けとったことを示すACK (アクノリッジ) を送信ステーションがトークンを保持している内に返す。これを immediate ACK と呼び PROWAY の特徴である。

PROWAY と CAMAC シリアルハイウェイの違いは、PROWAY においては、伝送路に接続された



任意のステーションが自由にデータを伝送できることにある。しかしここでは CAMAC シリアルハイウェイにみられた C, N, A, F によりモジュールに任意にリードとライトができる能力は失われ、例えば A ステーションが B ステーションが管理しているデータを読みたいときは、まず A → B へのメッセージの送信で B へデータが読みたいことを伝え、次いで B → A へのメッセージにより欲するデータが得られることになる。

#### 4. ローカルエリアネットワーク (LAN)

加速器の規模が巨大化すると、どうしても 2 台以上の計算機による制御システムが必要となる。特に近年の計算機の発展にともなう価格の低下により、使用される計算機の台数は急激に多くなっている。例えば、CERN SPS では 50 台以上の NORD10 及び NORD 100, TRISTAN 制御システムでも 20 台を超える HIDIC80 が用いられている。

これらの多数の計算機は有機的に結合されて一つのシステムを構成しなければ、複雑化した加速器の制御を荷なうことはできない。そこで必然的に計算機ネットワークをいかに行うかが問題となってくる。計算機ネットワーク技術、特にそのうちのローカルエリアネットワーク技術 (LAN) は加速器制御技術の中心といえる。

LAN とは、お互いに独立した計算機を結合するネットワークであり、ネットワークの規模が一事業

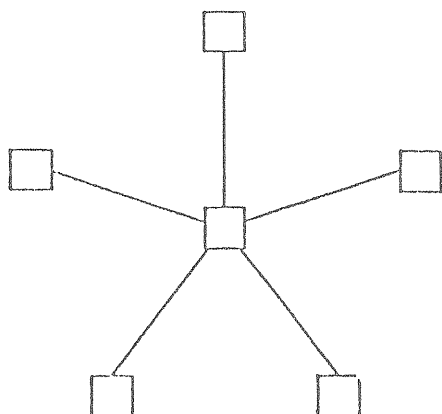


図 4. 1 スタートポロジ

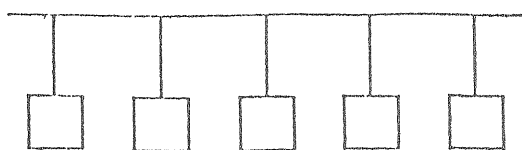


図 4. 2 バストポロジ

所や研究所内の敷地内に限られるものをいう。LAN はこの意味では分散処理の技術である。

#### 4. 1 LAN のトポロジー

LAN のトポロジーとは、ノードが幾何学的にどのように配置されているかを示すものである。代表的なトポロジーとしては以下にあげる、スター、バスそしてリングの三種がある。

##### 4. 1. 1 スター

図 4-1 にスタートポロジを示す。このトポロジーにおいてはセンターのノードが通信の管理を行う。あるノードはまずセンターのノードと通信を行い、次いでセンターノードは目的のノードと通信を行う。このトポジを用いた代表的な加速器制御システムは CERN の SPS である。

スタートポロジにおいては、(1)センターノードの故障は全体のダウンとなる、(2)センターノードにすべての通信が集中するので、スラッシュポイントがこの部分で制限されてしまう、という弱点を持つ。最近の加速器の制御システムでは、次にのべるバスまたはリングトポジを用いる傾向が次第に強くなってきている。

##### 4. 1. 2 バス

バストポロジの概念を図 4-2 に示す。バストポロジの特徴は、ある 2 つのノードの間のコミュニケーションは中継ノードを介することなく直接に行われることにある。そこで、バスの場合伝送路にノードを接続するには非常に簡単な方法をとることができ(しばしば受動素子のみで接続できる)、またあるノードが故障しても、残りのシステムにほとんど影響を与えずに済ませることができる。

バストポロジにおいて CSMA/CD 方式というアクセス方法(アクセス方法とは、伝送路の使い方をいう)を用いる代表例が 4. 2 節で紹介する Ethernet であり、トークン方式をとったときの例が 3. 2 節の PROWAY である。

##### 4. 1. 3 リング

リングトポジとは図 4-3 に示すように、ノードがリング状に結合されているものである。リングトポジにおいては、通常データは一方方向に伝送され、すべてのノードを経由して発送もとに戻ってくる。各ノードは信号を受けるとアクティブリピータにより増巾して次のノードに渡す。媒体には同軸ケーブルと光ファイバーケーブルがよく使われる。

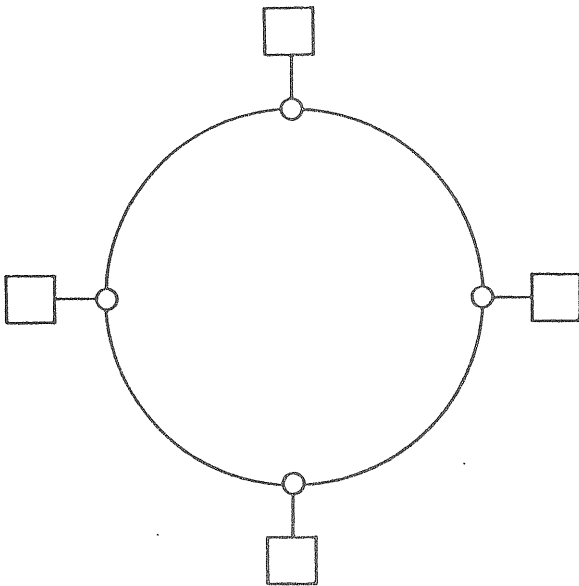


図4.3 リングトポロジー

リングトポロジーでは、一つのノードの故障が全システムのダウンにつながるが、これはバイパスやループバックにより保護されることが多い。4-3節において、リングトポロジーの代表例として TRISTAN 制御システムで用いられている DFW (データフリーウェイ) について説明する。

#### 4.2 Ethernet

バストポロジーにおいて CSMA/CD (Carrier-Sense Multiple-Access with Collision-Detection) というアクセス方式をとる代表が Ethernet である。Ethernet は Xerox, DEC, Intel 社により標準仕様が定められた LAN であり、構成が簡単であることから広く使われている。

Ethernet の特徴をあげると：

- |                 |            |
|-----------------|------------|
| (1) データ伝送速度     | 10Mbps     |
| (2) ステーション間最大距離 | 2.5km      |
| (3) 最大ステーション数   | 1024       |
| (4) 最大メッセージ長    | 1518バイト    |
| (5) 媒体          | 同軸ケーブル     |
| (6) 伝送方式        | ベースバンド伝送方式 |

任意のノードはデータを送出する前にバスの状態を調べ、もしバスが使用中でなければメッセージを送信する。もしバスが使用中であれば、バスが空くまで待ち、その後ある短い時間をおいてから送信を開始する。これでほとんどの衝突はさけられるが、バス上の信号の伝播に要する時間が有限であることから、衝突を完全に回避することは原理的に不可能である (ここまでが CSMA の説明)。そこで送信ノードは、メッセージを送信中に自分が送っているメ

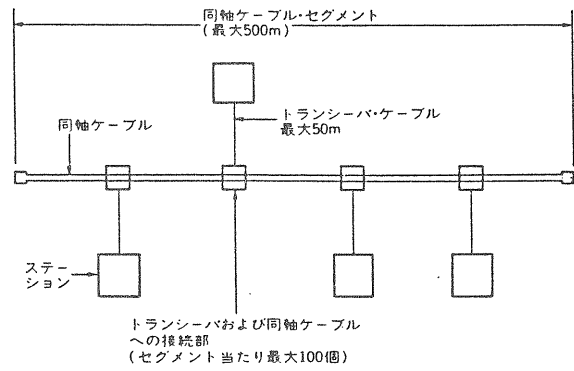


図4.4 Ethernet

ッセージが他のノードから送られたメッセージと干渉しないかどうかを調べる。もし干渉が検出されたときには、衝突が起ったものとして再送信をする。単純に再送信したのでは、再び衝突が発生するので、あるアルゴリズムに従って、再送信の前にランダムなディレイを加える (CD の説明)。

Ethernet は10Mbps と高速でありかつハードウェアも単純であり非常に魅力的である。しかしながら、衝突時における再送の遅延時間が確率的に決まり、また送信が完了するまで何回の送信が必要かが決定できないという弱点がある。この制約はオフィスオートメーションの世界では何の問題もおこさないが、リアルタイム性を要求するプロセス制御 (加速器制御もプロセス制御の一種である) では不都合なことがある。しかしながら、ネットワークの負荷が小さいときは、衝突の確率は小さく、負荷を十分小さくしておくという条件のもとで十分に役に立つ LAN といえる。図4-4に最も単純な Ethernet を示す。

#### 4.3 DFW (データフリーウェイ)

トークンリングの代表例として TRISTAN 制御システムで用いている DFW (データフリーウェイ, 日立製作所製) をとりあげる。DFW は、ステーションの間を光ファイバケーブルで接続している。光ファイバケーブル上の伝送速度は10Mbps である。DFW におけるアクセス方式は、次のようになっている。

- (1) アイドル状態においては、GA (go ahead) と NPOL (normal polling) というパターンがリング状を巡回している。これらのパターンがフリートークンの役割を果たしている。
- (2) あるステーションが別のステーションにメッセージを送ろうとするときは、送信元のステーションはまず GA と NPOL をつかまえないといけない。そしてこのステーションは NPOL

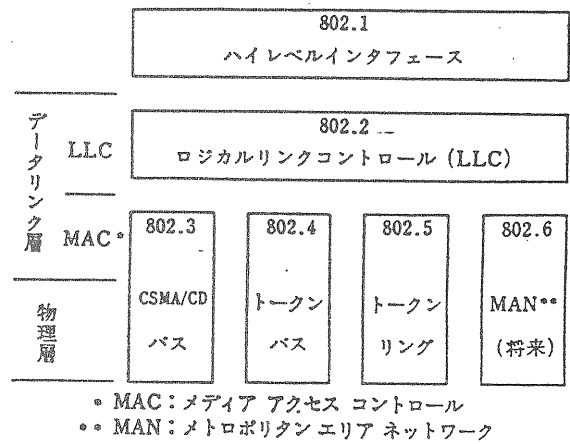
を RSV (reserve) というパターンに変え、送るべきデータを I-frame として送り出す。GA と RSV の組はビジートークン、すなわち、すでにトークンがあるステーションに保持されていることを示す。

- (3) 送信元のステーションより下流にあるステーションは I-frame の内容をチェックする (I-frame には送信元と送信先のアドレスが書かれている)。もし送信先が自分自身であればこのステーションは I-frame の内容をとり込む。違っているときは、メッセージを次のステーションに渡す。
- (4) 送信先のステーションは I-frame が正しく受信されたかを調べ、その結果を含んだ RESP-frame と GA と RPOL (retry-polling) を送り出す。
- (5) 送信元のステーションは RESP-frame の内容を調べ、その結果(4)で I-frame の受信がうまくいっていることが分ったときは RPOL を NPOL に変える。すなわちフリートークンを送出する。エラーが検出されたときは、RPOL を RSV に変え、(2)からくり返す。

トークンリングは、4. 2 節の Eathernet のところで説明した CSMA/CD と比較して、(1)トークンリングでは一時に一つのステーションからの送信だけが許可されるので、送信メッセージの間の衝突は発生しない、(2)トークンリングでは、あるステーションが送信できるまでの待ち時間の上限が定められている、という利点を持っている。このため、リアルタイム性の要求されるプロセス制御の世界ではよく使われる方法であり、将来性も大きいといえる。KEK の Linac の制御システムと TRISTAN 制御システムは加速器の制御システムに光ファイバトークンリング LAN を用いた世界初の試みである。

#### 4. 4 LAN の標準化

IEEE802委員会 は図 4-5 に示すように、CSMA/CD バス (IEEE802. 3)、トークンバス (IEEE802. 4)、トークンリング (IEEE802. 5) の三種の LAN の標準化の進めている。このうち 802. 3 はほぼ Eathernet であり、802. 4 は 3. 2 節で紹介した PROWAY の一種である。802. 5 のトークンバスは 1982 年に IBM が発表したものに基づいており、IBM リングと呼ばれている。今後規格が確定するにつれて、IEEE 標準の LAN の使用は急激に高まると予想され、加速器の制御においても、規格に基づいた製品の使用が一般化するであろう。



IEEE 802 LAN ファミリー

図 4. 5

## 5. ソフトウェアシステム

### 5. 1 オペレーティングシステム

加速器の制御に用いられる小型計算機のオペレーティングシステムとしてはマルチタスクオペレーティングシステムが用いられる。

マルチタスクオペレーティングシステムでは計算機上にタスクと呼ばれる処理の単位を考える。複数のタスクはお互いに計算機の資源(たとえば CPU, メモリ, 補助メモリ, プログラム)などを競合して使う。各タスクには優先レベルが割り当てられており、低レベルのタスクが CPU を使用しているときに、レベルの高いタスクが起動要求されると、低レベルタスクは高レベルタスクに CPU を明け渡し、高レベルタスクの実行が終了したあとで再び実行を開始することになる。

加速器の制御や一般にプロセス制御においては、タスクは外界の事象にもとづく割り込みによって起動されたり、計算機の内部クロックにより起動されることが多い。

### 5. 2 アプリケーションプログラム

初期の加速器制御システムでは、専門のプログラマが加速器制御のためのアプリケーションプログラムを書いていた。加速器を研究したり運転したりする人々は、専門のプログラマに自分は何をしたいかを伝え、それをプログラムにってもらう必要があった。専門のプログラマによるプログラミングは、(1)プログラミングはしばしばアセンブラにより行われていた、(2)マルチタスクオペレーティングシステムのものとのプログラミングはしろうとの手に余ることが多い、という理由から行われていた。

しかしながら、この方法はあまり能率的ではない。他人にプログラムしてもらうには、自分で考えていることを正確に伝えることが必要であり、根気と時間のいる作業となる。またほんの少しの変更にも作成者をわずらわさねばならない。

そこで、問題をかかえている人が自からプログラムを作ることができるようなシステムが望まれることになる。

この問題を解決するには、しろうとでもプログラミングができるような道具だてが必要となる。そこで(1)プログラミング作成の言語としてインタプリタを採用し、かつ(2)マルチタスクオペレーティングシステム、特にネットワークの部分についての詳細をユーザからかくすという方法が試みられ、大きな成功をおさめることになった。

インタプリタとは、ユーザが作成したプログラムをコンパイル（ソースプログラムを計算機が理解できる機械語に変換すること）することなく直接に実行する方式である。パーソナルコンピュータで普及している BASIC はインタプリタの代表的言語である。インタプリタの特徴は、プログラムの作成—実行—実行の停止—プログラムの修正—再実行—……という作業が能率よくできることにある。インタプリタにおいては、プログラムにエラーがあると実行はそこで停止され、ユーザにエラーの種類とエラーが発生した場所を教えてくる。そこでエラーを修正し、再実行させればよい。

インタプリタのもう一つの特徴は、インタプリタではタスクとして実行されるのはユーザが作成したプログラムではなく、プログラムを解釈実行するイ

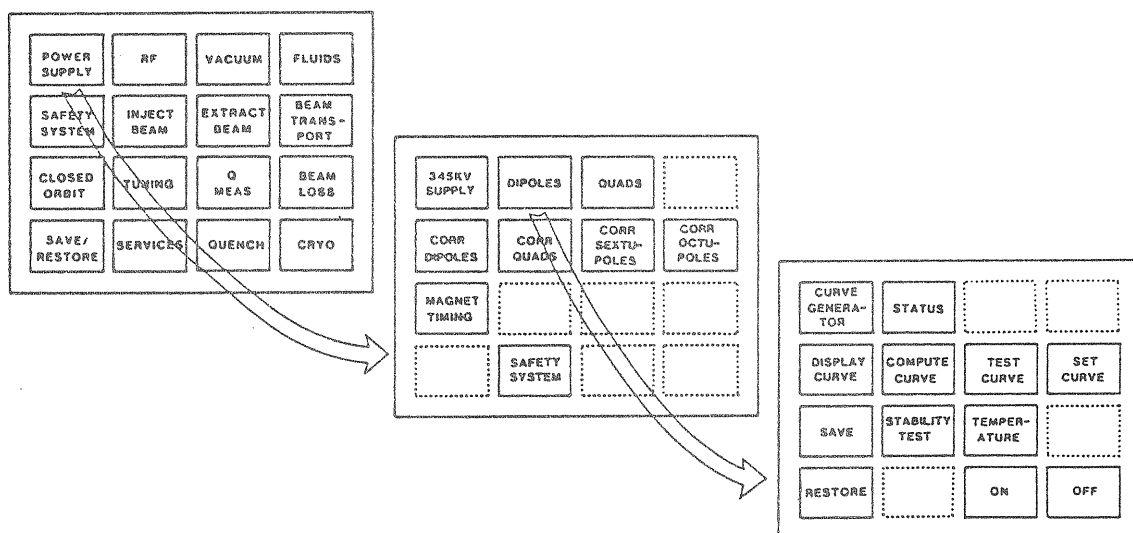
ンタプリタ自身であり、プログラムはこのインタプリタが処理するデータであることである。ネットワーク上で計算機間でやりとりされるのはデータである。インタプリタはプログラム自身がデータであることから、プログラムを計算機内で転送し合うことが可能となる。

これを利用してマルチ計算機ネットワーク上の計算機内でプログラムを転送し、転送先で送られたプログラムを解釈実行するインタプリタ言語をつくることができる。その最初の例が NODAL である。NODAL は CERN SPS の制御システムのためにつくられた言語であり、すでに10年以上の歴史を持っている。近年は CERN CPS, DESY PETRA, TRISTAN の制御にも NODAL または NODAL とよく似たインタプリタ言語が使われている。

このようなマルチ計算機インタプリタにより、プログラム作成者は複数の計算機にまたがる処理を一つのプログラムとしてまとめることができ、プログラム作成の能率は大巾に向上することになる。NODAL についてのさらに詳しい説明は 7. 3 節で行う。

## 6. マンマシンインターフェイス

マンマシンインターフェイスとは、加速器制御システムの使用者が制御システムに働きかけるために用いる機器のことである。制御システムの良さの指標として、しばしばマンマシンインターフェイスの使い易さが大きな要素を占めることが多く、この部分のデザインの制御システム全体に占めるウェイト



Touch-Button Tree

図 6. 1

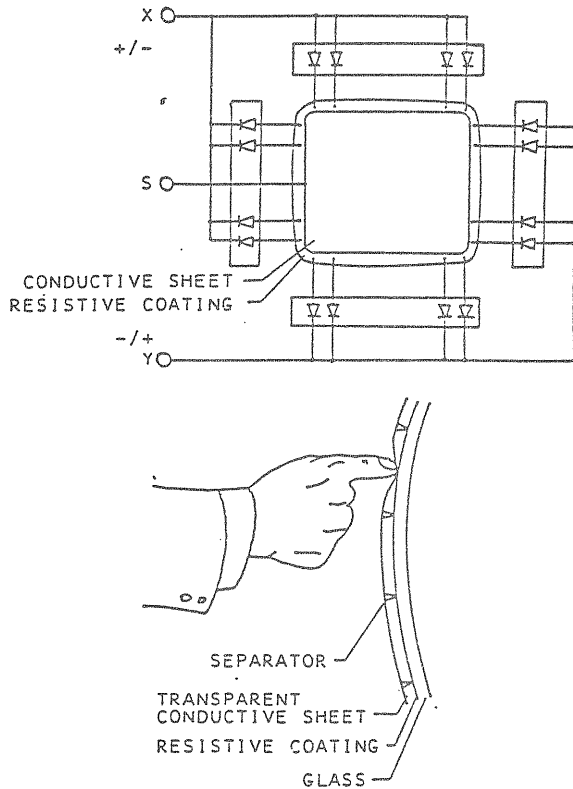


図 6. 2 タッチパネルの原理

は大きなものとなっている。

ここでは近代的な加速器の制御システムのマンマシンインターフェースとして良く用いられるタッチパネルとカラーグラフィックディスプレイについて解説する。

### 6. 1 タッチパネル

タッチパネルとは、押された位置を検出できる透明なパネルとその下にある CRT を組み合わせた装置である。計算機のソフトウェアにより CRT 上にボタンの画面を作り出し（ボタンの個数は  $4 \times 4$ 、 $5 \times 7$ 、 $7 \times 7$  など種々ある）そのうちのどのボタンが押されたかをソフトウェアで判断する。タッチパネルの最大特徴は、ソフトウェアで任意の形状のボタンとボタン上の文字を作り出せることにある。これは、階層構造に配列された項目を検索するに適している。図 6-1 に示すように、まず第一の画面で、あるボタンを選ぶ、すると選ばれたボタンに対応する次の層の画面が表われる。この画面からさらに一つのボタンを選び……と展開していくことができる。

TRISTAN 制御システムでは、任意の形状のボタンを作り出す能力をうまく使い、オペレータコンソール上から、テンキー、アルファニューメリックキーそして各種の押ボタンスイッチを追放すること

ができた。

図 6-2 に、TRISTAN 制御システムに用いているタッチパネルの原理を示す。

計算機制御の進展にともない、加速器の制御に使われるマンマシンインターフェイスも変化してきている。その傾向として

- (1) 計算機のフレキシビリティに頼り、ソフトウェアで使い方にバラエティを持たせる。
- (2) その結果、オペレーターズコンソールに用いられる機器の種類が減少している。

ということがあげられる。

### 6. 2 カラーグラフィック CRT

プロセス制御の世界では、フルグラフィックのカラー CRT の利用は一般化していき、セミグラフィック CRT が主に用いられている。これは、フルグラフィック CRT の価格がまだ高いこと、そしてフルグラフィック CRT 上に画面をつくるソフトウェアのオーバーヘッドが大きいためといえる。

それとは逆に加速器の制御システムにおいては、対象の複雑さを反映して、高解像度のフルグラフィックカラー CRT が多用される。例えば TRISTAN 制御システムにおいては、 $1024 \times 960$  のピクセルを持つカラー CRT が用いられている。図 6-3 に、この CRT 上の画面のハードコピーの一例を示す。

フルグラフィック CRT を用いる際に、インタラクティブに画面を作成できることは、プログラム作成の能率を大巾に左右する。TRISTAN 制御システムにおいては、前章でのべたインプリタ言語 NODAL にグラフィック CRT 用のファンクションが組み込まれており、プログラム作成の能率が大巾に向上している。

### 6. 3 オペレーターズコンソールの発展

小型の加速器から、大型の加速器へと発展していくにつれて、加速器制御のオペレーターズコンソールは、図 6-4 に示すような発展をして来ている。すなわち、

- (A) 加速器複合体の構成要素、すなわち LINAC, BOOSTER, MAIN RING などはそれぞれ独立した計算機で制御され、各計算機には独自のコンソールが付属している。
- (B) コンソールを司さる計算機とハードウェアの制御を司さる計算機は分離し、かつこれらの計算機はネットワークで結合される。しかし、各コンソールはいまだにそれぞれ独自なものとして存在している。

(C) コンソール用の計算機とコンソールは個性を失い複数台の同一のコンソールが並列して存在する(ちなみに、TRISTAN 制御システムのコンソールはこの方式を採用している)。どのコンソールからでも、どの加速器も制御できる。

## 7. トリスタン制御システム

### 7.1 はじめに

高エネルギー物理学研究所 (KEK) では、TRISTAN と呼ばれる巨大加速器を建設中である。TRISTAN は、2.5GeV 線形電子加速器、8 GeV 蓄積リング (AR)、30GeV 電子陽電子衝突リング (MR) という三つの加速器の複合体であり、電子と陽電子の衝突反応による素粒子実験のための装置である (図7-1 参照)。これらの三つの加速器のうち、線形電子加速器は、フォトンファクトリ光源用ストレージリングへの入射器としての役もはたしてお

## CLOSED ORBIT DISTORTION

85-07-03 09:45:21

• (MEAS)

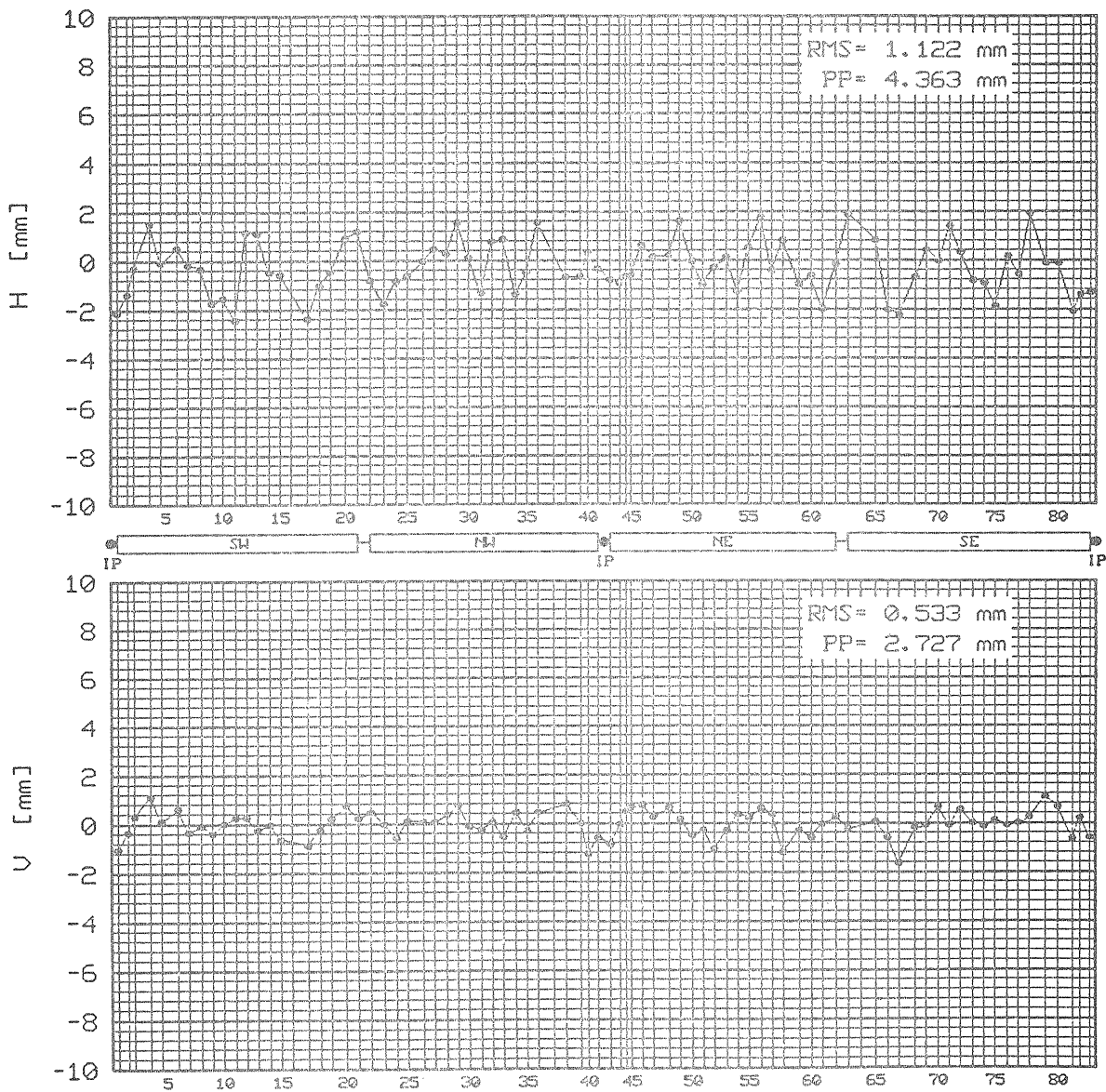
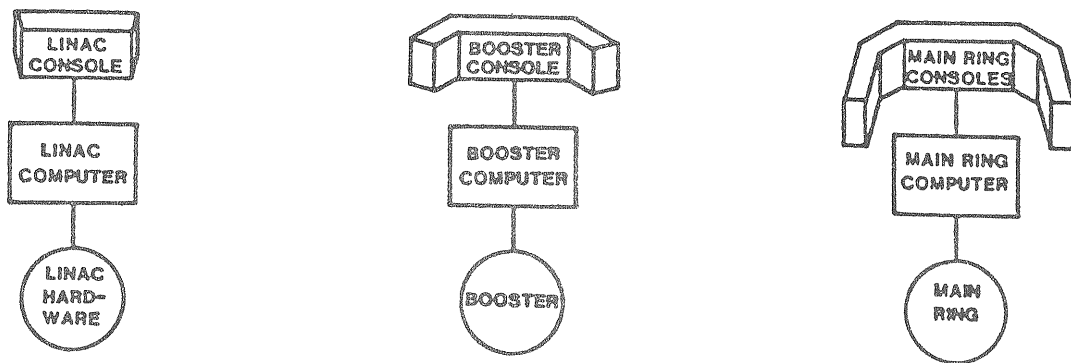
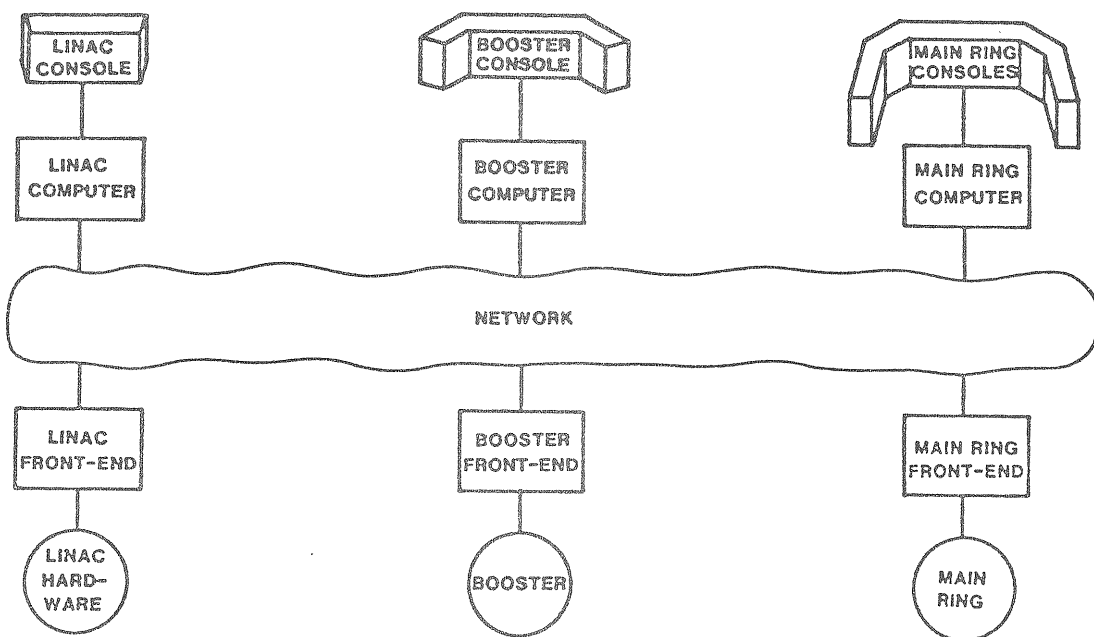


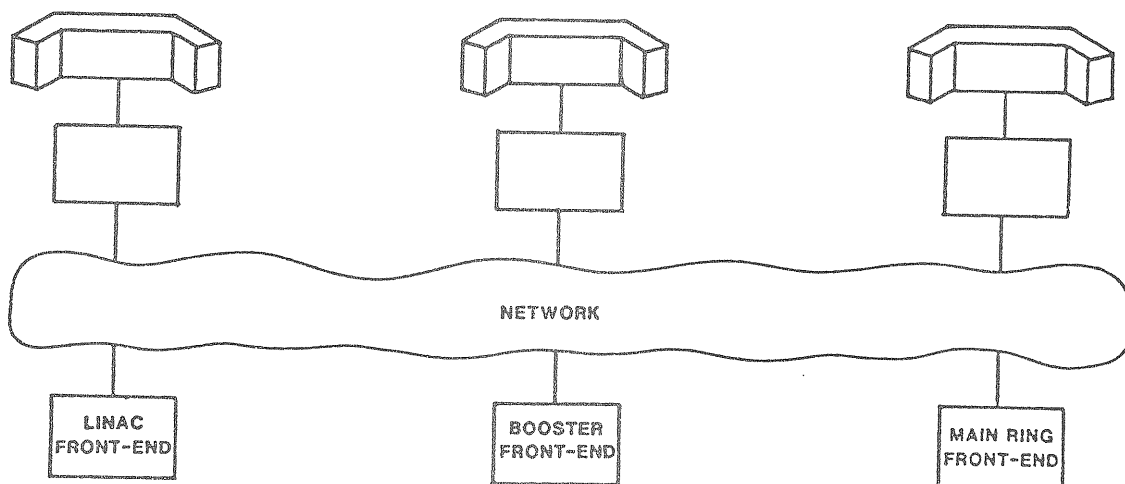
図 6. 3 グラフィック CRT 画面の一例



(A) Dedicated System

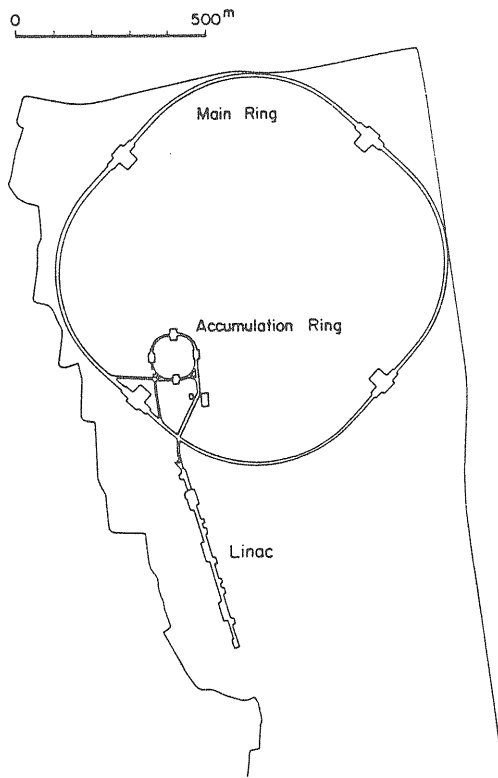


(B) System with Dedicated Computers and Consoles



(C) System with General Purpose Consoles and Console Computers

図 6. 4 オペレーターズコンソールの発展



TRISTAN 加速器の配置

図 7. 1

り、独立の制御システムにより制御されている。一方、AR と MR は、一つの制御システムのもとに統轄されており、同一のコントロールセンタから制御される。以下では、AR/MR 制御システムについて説明する。

TRISTAN を制御する上で、以下の事項を考慮しなければならない。

- (1) TRISTAN では直径 1 km 以上の領域に種々の被制御機器が分散している。また、制御点数は 2 万点以上となる。
- (2) 加速器は研究用の巨大機械であり、非常に複雑な装置である。加速器は動き出した当初から所期の性能を発揮することはまれであり、性能を向上するために不断に研究を続け、得られた結果に基づいて、ハードウェアの手直しや、制御方法の改善を行っていかねばならない。加速器の利用の形態が変わることにもなうシステムの変更も頻繁にある。このような不断の変更に対応するためには、制御システムはハードウェアとソフトウェアの両面で十分に柔軟でなければならない。
- (3) 特に、ソフトウェア、すなわち制御プログラムがシステムの変更に柔軟に対応できるためには、ソフトウェアの生産性が十分高いことが必

要である。それには専門のプログラマではない加速器の技術者が自分で必要とする応用プログラムを作成できるようにすることが有効である。

- (4) 加速器の動作のシュミレーションとそれに基づくフィードバックのため、しばしば非常に大きなプログラムを実行させる必要がある。このために必要な計算能力を制御システムは利用できなければならない。

私たちはこれらの問題を次のような方法で解決した。(a)分散配置された小型計算機を光ファイバにより結合し、N:N のトークン・リング・ネットワークからなる LAN を構成する、(b)ソフトウェアとして、マルチ計算機機能を持つインタプリタ言語である NODAL を採用する、(c)制御用の小型計算機ネットワークを高エネルギー物理学研究所の中央計算機と結合し、大型計算機をリアルタイム的に用いる。以下の章で、これらについて詳細に説明する。

## 7. 2 システムの構成

### 7. 2. 1 計算機ネットワークと小型計算機

TRISTAN の規模と複雑さに対応するため、私たちは分散制御システムのアーキテクチャを採用した。加速器の周辺に分散配置された 24 台の小型計算機（日立製作所 HIDIC 80E および HIDIC 80M）を光ファイバ・ケーブルにより結合して、N:N のトークン・リング・ネットワークを形成する。計算機の結合には、日立製作所の光データフリウェイ（DFW）を用いている。光ファイバ上の伝送速度は 10Mbps である。

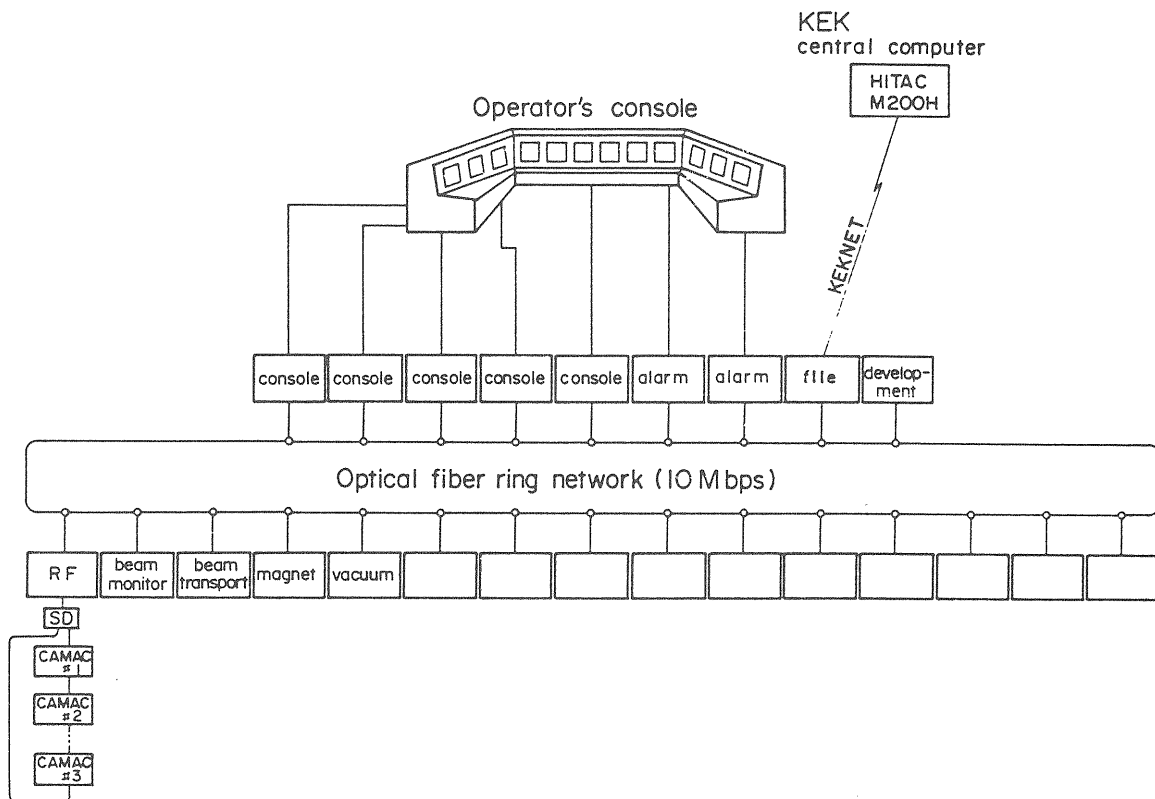
小型計算機はシステム計算機とデバイス制御計算機という二つのカテゴリーに分類される。システム計算機とは、コントロール・センタの機能を分担する計算機であり、コンソール計算機（5 台）、アラーム計算機（2 台）、ファイル計算機（1 台）、プログラム開発計算機（1 台）の計 9 台であり、中央コントロール室内に置かれている。

デバイス制御計算機は、高周波加速装置、電磁石と電源、真空装置などの加速器の機器を扱う計算機であり、加速器の周辺の側室に分散して配置されている。

それぞれの計算機からは、2.5Mbps のビット・シリアル CAMAC ハイウェイが被制御機器までのびている。AR に用いられる CAMAC クレートの数は約 40、MR 用は約 140 である。

図 7-2 は、TRISTAN 制御計算機システムのご概念を示す。





TRISTAN制御システムの概念図

図7. 2

### 7. 2. 2 大型計算機との結合

TRISTANのような複雑な加速器を制御したり、研究したりする上で、大きな計算能力をリアルタイムに用いなければならないことが多い。一例として、加速器内のビームの閉軌道の補正をあげる。

加速器を性能良く運転するには、加速器内のビーム軌道が設計値と一致するように補正を加えなければならない。このため、ARには83ヶ、MRには392ヶのビーム位置モニタと、適当な個数の補正用小型偏向電磁石が配置されている。ビーム軌道を補正するには、位置モニタで測定された位置情報を入力として、補正磁石の電流値を計算しなければならない。一般にこのような計算は、大きな次元のマトリックス演算をとめない、小型計算機の計算能力では、速度性が保証されない。

そこで、小型計算機ネットワーク上の一計算機(ファイル計算機)がKEKNETにより、高エネルギー物理学研究所の中央計算機に接続されている。KEKNETは、日立製作所のデータフリーウェイを応用した、高速のネットワークである。

## 7. 3 NODAL

### 7. 3. 1 NODALのマルチ計算機機能

TRISTAN制御システムのソフトウェアの体系は、CERN(欧州原子核研究所)のSPS加速器の制御のために開発されたNODALに基づいている。NODALはインタプリタ言語であり、プログラムをインタラクティブに開発できる。NODALの最大の特徴は、そのマルチ計算機機能にある。この機能により、TRISTAN制御用の分散された計算機が統合されることになる。

NODALのマルチ計算機プログラムの例を下に示す。

```

1.1 DIM A(10)
1.2 EXEC < 2 > 2 A ; WAIT < 2 >
1.3 FOR I = 1, 10 ; TYPE I A(I) !
1.4 END

2.1 FOR I = 1, 10 ; SET A(I) = MAG(I, 'CUR')
2.2 REMIT A

```

このプログラムが計算機1の上で解釈されるものとする。インタプリタは、行1.2“EXEC < 2 > 2 A”にいきつくと、グループ2の行(すなわち2.1と2.2)と配列Aを計算機2に転送する。計算機2の上のインタプリタは、これらの行をインタプリトし、REMITコマンドにより答を返送する。計算機1は、

“WAIT<2>”で、計算機2からの返送を待ち合せ、答を受けとったのち、1,3行で配列Aの値を印字する。

この例が示すように、一つのNODALプログラムはいくつかのサブプログラムに分割され、そのうちのあるものは他の計算機に転送されそこで解釈実行される。このようなマルチ計算機機能により、プログラム作成者は、複数台の計算機にまたがる処理を一つのプログラムとしてまとめることができ、プログラム作成の能率は大巾に向上する。

この他、NODALには、ネットワーク内の任意の計算機のファイルを自由にアクセスする機能がある。例えば、ある計算機のNODALプログラムのリストを、他の計算機、例えばDVOという計算機のラインプリンタに出力するには、

```
OPEN (11, 'W', 'DV 0/LP:')
LIST<11>
CLOSE (11)
```

というように、ファイル名(LP:)の前に計算機名DV0をつけてやればよい。

7.3.2 データモジュールと分散データベース  
マルチ計算機機能とならぶNODALのもう一つの重要な特徴に、データモジュールの概念がある。データモジュールは一種のデバイスハンドラであり、7.3.1節であげた例中の“MAG(I, 'CUR')”はその一例である。

```
SET A(I)=MAG(I, 'CUR')
```

ここで、MAGはデータモジュールの名前、Iはユニット番号、そしてCURはプロパティである。プロパティは機器に対する操作の種別を示し、ユニット番号は機器のグループ中の何番目に対する操作であることを示す。上例では、I番目の電磁石から電流を読み出し、配列AのI番目の要素に代入することを意味している。

各データモジュールはそれぞれデータテーブルと呼ばれる二次元のテーブルを一つずつ持っている。データテーブル中には、CAMACモジュールのアドレスや変換係数などの機器の操作に必要な各種のパラメータがしまわれている。データモジュールはこれらのパラメータを参照しながら、機器の制御を行うことになる。

データモジュールとデータテーブルは、機器が接続されている計算機上にロードされている。そこで、NODALのマルチ計算機機能は一種の分散データベース・ハンドラの働きをすることになる。

これらのデータモジュールは、リンクマンと呼ば

れるハードウェアを担当する人たちによって作成される。NODALのみを使うユーザにとっては、データモジュールは、システムに存在する外部手続きとなる。

## 7.4 KEK NODALの特徴

KEK NODALは、オリジナルのCERN SPSのNODALに、いくつかの改良を加えてある。主なものは、(1)中間語の採用による高速化、(2)スクリーン・エディタ機能の導入、(3)自己拡張の単純化、及び、(4)大型計算機との結合機能である。

### 7.4.1 高速化

インタプリタにつきものの、実行速度の遅さを克服するために、KEK NODALでは、ソースをいったん中間語に変換し、実行時に中間語を解釈する方式を採用した。

プログラマが端末上でソースを入力し、リターン・キーを押すと、NODAL構文解析タスクが起動され、ソースを中間語に変換する。変換された中間語はメモリ上に格納され、ソースはディスク上のバッファに格納される。NODALインタプリタ・タスクは実行時にこの中間語を解釈実行する。そこで、例えばASCIIストリングからなる定数を2進形式の浮動小数点に変換することや、オペレータとオペランドの順序の逆ポーランド方式への並びかえなどの作業は実行時には不要となり、その分だけ実行時間を短くすることができる。

NODALの実行速度を上げるその他の工夫として、(1)変数のサーチをハッシングで行う、(2)行番号、変数、関数用のキャッシュを用いる、の二つがとりいれられている。(2)は、いったん使われた行、変数、関数のアドレスをキャッシュ・テーブルにしまい、次回以降にはキャッシュ・テーブルを見ることにより、サーチの時間を短縮することを意味する。

### 7.4.2 スクリーン・エディタ

フルスクリーン・エディタは、編集の進行と結果が具体的にCRT上で見えるので、非常に使い易いエディタといえる。特に、TRISTAN制御システムのように、できるだけ多くの人が自ら応用プログラムを作成できるようにするという目標がある場合には、スクリーン・エディタの機能は必須である。KEK NODALにおいてはDECのVT100端末を標準端末として定め、VT100のキーパッドにスクリーン・エディタの機能を割り付けてある(図7-3)。

### 7.4.3 自己拡張性

T O P	PAGE +	PAGE -	BOTTOM
DEL EOL	DEL SOL	DEL LINE	DEL C
COMAND	FIND NEXT	CHANGE	PRINT
L TOP	L END	COPY	W I N D O W
OPEN LINE		REPLACE	

図7. 3 VT100キーボードへのエディタ機能の割り付け

2.2節で述べた、データモジュールは、KEK NODALにおいては、PCL（日立製作所 HIDIC 80シリーズ上で動く制御用コンパイラ言語であり、FORTRAN と非常に近い仕様をもつ）で書かれた外部手続きである。データモジュール以外にも、ユーザ関数と呼ばれる PCL で書かれる外部手続きがある。ユーザ関数の代表的な例として、グラフィック用関数や7.4.4節であげる大型計算機との通信用の関数をあげることができる。これらの外部手続きのシステムへの組み込みが簡単であることはシステムの柔軟性を高める上で非常に重要である。

KEK NODAL は、コアロード・イメージの形でシステム内にロードされているこれらの外部手続きのアドレスと引数の個数、型などを、システム中のテーブル上に保持している。データモジュールとユー

ザ関数は、実行時にこのテーブルの内容に基づいてダイナミックにリンクされる。外部手続き組み込み時に NODAL インタプリタの再コンパイル、再リンクは不要である。

#### 7. 4. 4 大型計算機との結合機能

KEK NODAL システムから大型計算機を用いるために、数個のユーザ関数が用意されている。これらの関数の中の主なものの機能を表7-1に示す。ユーザはこれらの関数を用いて、NODAL プログラム中から中央計算機のデータセットのリード/ライト、JOB のサブミット、JOB のステータスの読み出しができる。実際はこの関数はファイル計算機のみであり、他計算機は EXEC などの NODAL マルチ計算機コマンドを用いて、アクセスを行うことになる。

KEK NODAL システムでの大型計算機の利用の仕方は、JOB の入力用のデータセットに NODAL プログラムからデータを書き込み、次に JOB をサブミットし、JOB の実行終了後に JOB からの出力データセットを NODAL プログラムから読み出すという形をとる。この方法の利点は、大型計算機上のプログラムをオンライン用に改造しなくて良いことにある。

制御に必要な応答を保証するために、サブミットされるバッチジョブには特殊なプライオリティが与えられている。

#### 7. 5 システムの評価

##### 7. 5. 1 システムの現状と見通し

現在20台の小型計算機がネットワークに結合されており、1983年10月に完成した AR の制御と研究に使用されている。

各コンソール計算機にはそれぞれ500個の操作用プログラムが蓄積されており、このうちの200は実際

表7. 1 KEKNET 用 NODAL 関数

関数名	機能
FALOC	大型計算機のデータセットを定義し、次いで OPEN する
FCLOSE	大型計算機のデータセットを CLOSE する
FWARAY	大型計算機のデータセットに NODAL 配列の内容を書き込み1レコードとする
FRARAY	大型計算機のデータセットの1レコードを読み NODAL 配列にセットする
FWSTR	大型計算機のデータセットに NODAL の文字列を書き込み1レコードとする
FRSTR	大型計算機のデータセットの1レコードを読み NODAL 文字列にセットする
FSBMIT	大型計算機の JOB をサブミットする
FSTATU	大型計算機の JOB のステータスを読み出す

表7. 2 NODAL ベンチマークテストの結果

FOR LOOP	0.319 msec	DO 33.20	
SET A=1	0.463 msec	33.20 RET	0.612 msec
SET A=B	0.668 msec	DO 54	
SET A=A+1	1.066 msec	54.10 GOTO 54.20; RET	1.152 msec
SET A=A-1	1.063 msec	\$SET C="A"	0.879 msec
SET A=A*0.9999	1.066 msec	\$SET D="A" "B"	1.683 msec
SET A=A/0.9999	1.059 msec	\$SET DS(4)="A"	1.134 msec
SET A=A^0.9999	1.453 msec	\$IF C="X";	1.113 msec
SET D1(4)=1	1.150 msec	\$SET C=1	3.664 msec
SET D2(4,4)=1	1.414 msec	\$DO "S A=1"	265.306 msec
SET D3(4,4,4)=1	1.673 msec	SET A=SIN(1.001)	1.569 msec
IF B>1;	0.771 msec	SET A=SQRT(1.001)	1.522 msec
WHILE B<=1;	0.769 msec	SET A=DEF(1.001)	3.173 msec

に加速器の操作と研究に使われている。全システムでは、プログラムの総数は1,500以上となる。標準的なプログラムの大きさは、50~100行からなる。この本数のプログラムが約10人の人々により、1983年7月からの約1年間で製作された。さらにその後も日々プログラムの改良が続き、また新しいプログラムが作られつつある。このような効率的なプログラム開発ができたのは、次の要因に基づくと考えられる。

- (1) NODAL インタプリタによりインタラクティブなプログラム開発ができたこと。
- (2) データモジュールの作成がハードウェア担当のリンクマンにより行なわれ、NODAL プログラム作成者は、ハードウェアの詳細を知らなくてもよいこと。

### 7. 5. 2 NODAL の性能

表7-2にKEK NODALのベンチマークテストの結果を、各コマンドの実行に要する時間をmsecで示してある。一コマンドの実行時間はほぼ1 msecである。

### 7. 5. 3 ネットワークの性能

ネットワークの性能の指標にはレスポンスとスループットの二つがある。KEK NODAL システムのレスポンスを示すものとして、EXECの応答、また計算機ネットワークのスループットが十分であるか否かを示すものとして、ネットワーク上の回線負荷率の測定値を示す。

#### (1) EXEC の応答

二つの計算機間で、(a)最も単純な EXEC プログラム (すなわち、END 1行からなるプログラム) を実行したときの時間と、(b)若干のプログラムと配列 (954語) を送り、送り先では直ちにこの配列を REMIT したとき (この場合は、ほぼ1 K 語の

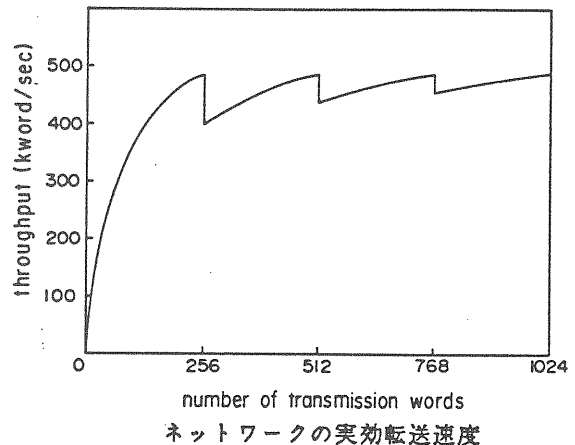


図7. 4

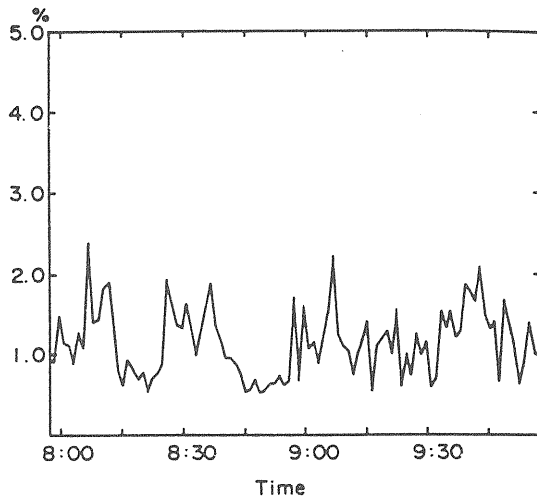
送信と受信となる) の、二つの場合の時間測定を行った。結果は、(a)の場合92msec、(b)の場合は160 msecであった。

TRISTAN 制御システム上の普通の EXEC でのデータ転送量はこの二つの範囲内にあり、100~150msecがこのシステムの EXEC の応答ということになる。この時間は、人間に対して遅れを感じさせる時間 (0.3秒) に比べて短く、また、加速器の標準的な機器の応答 (1秒) に比べても十分短いといえる。

#### (2) ネットワークのスループット及び回線負荷率

TRISTAN 制御の計算機間のネットワークに用いられているデータフリーウェイでは、パケットの平均語数、ループ長 (L)、ノード数 (N) によって最大実効転送速度が決定される。図7-4に、パケット平均語数を変えたときの最大実効転送速度がどのように変わるかを示す。このとき、Lとして1.5km、Nとして13を仮定した。

データフリーウェイではリング全体の制御をつかさどるノードである CST は回線情報のトレースを RAM 上に記録している。この RAM 上のデータを計算機で読み出し解析することで回線上のデータのやりとりの様子を知り、回線負荷率を計



回線負荷率の測定結果の一例

図 7. 5

算することができる。図 7-5 に、このようにして得られた回線負荷率の測定結果を示す。この例は、AR の定常的な運転に対応するもので、最高 2.4% の回線負荷率を示している。このときのパケットの平均語数は 36 語であり、これに対応する実効転送速度は 205K 語/秒であるから、1 秒間に 5 K 語のデータがネットワーク上を通過することになる。MR 完成後のシステムの回線負荷率は、これから推定すると、10~12% 程度と予想される。

## 8. 加速器制御システムの将来

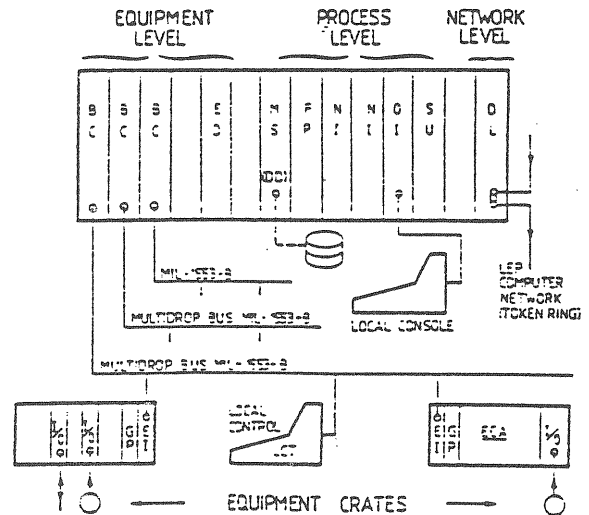
### 8. 1 LEP 制御システム

現在 CERN で建設が進んでいる LEP は直径 9 km の巨大ストレージングであり、1989 年の始めに運転開始の予定である。

LEP 制御システムは徹底した分散処理方式を採用している。このシステムにおいては、これまで加速器の制御システムに多用されてきた小型計算機を VME バスに基づくマルチマイクロプロセッサ (PCA : Process control assembly) により置きかえようとしている。

図 8-1 に PCA の概念を示す。PCA 中にはファンクショナルモジュール (functional module) と呼ばれるマイクロプロセッサが実装される (12MHz 68010 CPU)。ファンクショナルモジュールには次のような種類がある：

- (1) データリンクユニット (DL) : PCA を IEEE802.5 トークンリング (IBM リング) にインターフェースする。



- BC Bus Controller
- DD Disk Driver
- DL Data Link driver
- EI Equipment Interface
- GP General purpose Processor can act as
- SU Supervisor
- NI NODAL Interpreter
- MS Mass Storage (same)
- OI Operator Interface
- ED Equipment Directory
- FP FORTRAN Processor
- GP General Processor

図 8. 1 PAC (プロセスコントロールアンセブリ)

- (2) スーパーバイザリユニット (SU) : PCA 内のアクティビティの調整。
- (3) NODAL インタプリタモジュール (NI) : NODAL プログラムをインタプリトするモジュール。
- (4) イクイップメントディレクトリユニット (ED) : 機器操作のためのデータベース。
- (5) マルチドロップバスコントローラ (BC) : 機器を制御するための MIL 1553-B バスコントローラ。

一つの VME クレートにこれらのファンクショナルモジュールを適当に組み合わせて実装することにより、小型計算機上のタスクのそれぞれは、CPU とメモリを持つことになる。

このシステム構成は、制御システムをファンクショナルモジュールというシングルボード計算機の集合で構成しようとするに他ならない。そしてこのことはシステム構成の単純さからくる信頼性の向上とコストの低減につながる。またシステム全体での計算能力は非常に大きくなり、大巾なスループットの向上が期待できることになる。

図 8-2 に LEP 制御の計算機ネットワークを示す。

### 8. 2 残された問題

現在米国では直径 20~50km にも達する超巨大加速器 SSC が計画されている。SSC は超伝導加速器であり、必要とする超伝導電磁石の台数は一万近くに

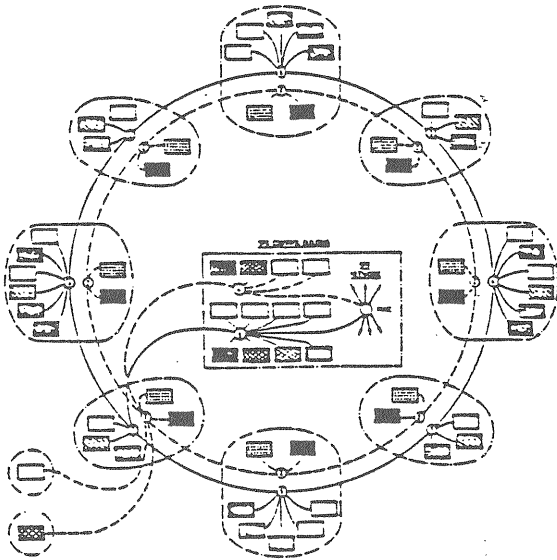


図 8. 2 LEP 制御システム概念図

なる。このような超巨大加速器を制御することはトリビアルとはいえず、次のような問題が立ちだかっている。

- (1) 信頼性をどうして保証するか？直径 1 km の TRISTAN と直径 50 km の SSC では、信頼性に対する要求はケタ違いである。規模が 50 倍となれば、単純に考えても信頼性は  $1/50$  となる。1 年に 1 回故障するシステムは 1 週に 1 回の故障となってしまう。
- (2) マルチプレッシングをどうするか？規模の増大にともないマルチプレクスしなければならない信号数はリニアに増加する。かくも多数の信号を有効にまとめられるであろうか。
- (3) 50 km に広がる空間でのリアルタイム性を持つ信号の伝達はどのように行えばよいか。例えば直径 50 km のトークンリングネットワークができたとする。リング上を信号は光定度で伝わったとしても 1 週には 0.5 msec かかる。これはどんなにがんばっても 1 秒に 2,000 回以上の転送はできないことを意味し、1 度に 1 kbyte 送ってもスループットは 2 Mbps にしかならない。

おわりに

このテキストを作成するにあたって、下にあげる文献から図表を転載した。

- (1) F. Beck and M. Gormley : Computer Control of Large Accelerators, Design Concepts and Methods, AIP Conference Proceedings No. 127 pp. 624~692.
- (2) IEEE : CAMAC Instrumentation and Interface Standards.

- (3) C. Weitzmann : Distributed Micro/Minicomputer Systems, Prentice-Hall, 1980.
- (4) M. C. Crowley-Milling : Evolution of Control Systems for Accelerators, 1982 CERN School of Computing.
- (5) J. Altaber et al. : The Message Architecture of the LEP Control System, CERN SPS/84-18 (ACC).
- (6) J. Altaber et al. : Multi-Microprocessor Architecture for the LEP Storage Ring Controls, CERN SPS/85-28 (ACC).
- (7) M. C. Crowley-Milling : Trends in Accelerator Control Systems, SLAC-PUB-3331.
- (8) 東山 尚, 小川敏夫 : システム制御における情報ネットワーク, 計測と制御 vol. 24, pp. 8~15.
- (9) 阿江 忠 : ローカル・ネットワーク技術の基礎と実際, CQ 出版社, 1983.
- (10) 今村博宣 : VME バスの規格と設計のポイント, インターフェイス, 1985年 8 月号, pp. 202~221.
- (11) 黒川真一, 秋山篤美, 生出勝宣, 加藤直彦, 菊谷英司, 小磯晴代 : トリスタン制御計算機システム—大規模分散型ラボラトリー・オートメーション—, 「LAN/マルチメディアの応用と分散処理」シンポジウム, 情報処理学会, 昭和59年10月.
- (12) 加藤直彦他 : KEK PS Control Computer System II, KEK Internal 80-6.