

JAERI-Tech  
97-073



## 新CICUの開発

1998年2月

青柳哲雄\*・佐藤 稔・坂田信也・松田俊明

日本原子力研究所  
Japan Atomic Energy Research Institute

本レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合せは、日本原子力研究所研究情報部研究情報課（〒319-1195 茨城県那珂郡東海村）あて、お申し越しください。なお、このほかに財団法人原子力弘済会資料センター（〒319-1195 茨城県那珂郡東海村日本原子力研究所内）で複写による実費領布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195, Japan.

© Japan Atomic Energy Research Institute, 1998

編集兼発行 日本原子力研究所

印 刷 株式会社原子力資料サービス

## 新CICUの開発

日本原子力研究所那珂研究所炉心プラズマ研究部  
青柳 哲雄\*・佐藤 稔・坂田 信也・松田 俊明

(1997年12月12日受理)

JT-60データ処理設備では、各システムの内、老朽化しハードウェア、ソフトウェア共保守困難なミニコンピュータを順次ワークステーションに更新している。1993年のマンマシンコミュニケーション計算機MMI、1994年の実時間処理計算機RTPに引き続き、ショット間処理計算機ISPとCAMACシステムを接続する基幹装置であるCAMACインターフェース制御装置CICUについても後継機の開発を行い、試験運用を実施してきた。現在、全面的に更新するための調整試験を行っている。

本報告書では、ワークステーションにVMEbusを組み合わせた構成で新規開発した、新CICUの概要およびソフトウェアの基本設計について述べる。

## Development of New CICU

Tetsuo AOYAGI\*, Minoru SATO, Shinya SAKATA  
and Toshiaki MATSUDA

Department of Fusion Plasma Research  
Naka Fusion Research Establishment  
Japan Atomic Energy Research Institute  
Naka-machi, Naka-gun, Ibaraki-ken

(Received December 12, 1997)

In JT-60 data processing system, workstations have been used to replace aged mini-computers which have been difficult to maintain their hardware and software. The communication computer MMI was replaced in 1993 and the real time processor RTP in 1994. The new CAMAC interface control unit CICU has been developed and tested. It is now at the final step to replace the old mini-computer system.

In this report, an outline of new CICU consisting of a workstation with VMEbus and a software design are described.

**Keywords:** JT-60, Data Processing System, Workstation, Mini-Computer, CAMAC, VMEbus

---

\* Research Organization for Information Science & Technology

## 目 次

第1編 新CICUの概要.....	1
1. 概 要.....	3
2. 基本システムの構成 .....	5
3. CICU ソフトウェア .....	10
4. 運用サポート .....	17
5. システムの構築 .....	23
6. CICU1,2,3 の構築 .....	32
付 錄 .....	36
第2編 新CICUの基本設計 .....	37
1. 概 要.....	39
2. 構 成.....	40
3. 機 能.....	42
3.1 ACMA 通信機能 .....	42
3.2 高速データ収集機能 .....	57
3.3 単一要求処理機能 .....	60
3.4 ダミー処理機能 .....	61
3.5 システム監視機能.....	61
3.6 コンソール機能 .....	62
4. 共有資源 .....	64
4.1 MBX .....	64
4.2 セマフォ .....	66
4.3 データバッファ .....	66
4.4 ロギングエリア .....	66
5. ISPとの通信 .....	67
5.1 通信の概要 .....	67
5.2 通信プロトコル .....	68
5.3 フォーマット .....	69
6. ACMAとの通信 .....	82
6.1 通信の概要 .....	82
6.2 使用ハードウェアと CAMAC コマンド.....	83
6.3 通信プロトコル .....	84
7. ハードウェア操作 .....	88
7.1 BSD-A シングルアクションの実行 .....	88
7.2 BSD-A 連続コマンドの実行 .....	89

7.3 BSD-A 高速データ収集の実行 .....	90
謝 辞 .....	91
参考文献 .....	91

## Contents

<b>Part1 Outline of New CICU.....</b>	<b>1</b>
1. Overview .....	3
2. Configuration of New CICU .....	5
3. New CICU Software.....	10
4. Operation Procedure .....	17
5. Setup of Prototype CICU .....	23
6. Setup of New CICU1, 2 and 3.....	32
Appendix .....	36
<b>Part2 Software Design of New CICU .....</b>	<b>37</b>
1. Overview .....	39
2. Configuration of New CICU .....	40
3. Software Functions .....	42
3.1 ACMA Communication .....	42
3.2 Fast Data Acquisition.....	57
3.3 Single Request .....	60
3.4 Dummy Function.....	61
3.5 System Monitoring .....	61
3.6 Console Function .....	62
4. Shared Resources .....	64
4.1 MBX .....	64
4.2 Semaphore.....	66
4.3 Data Buffer .....	66
4.4 Logging Area.....	66
5. Communication with ISP .....	67
5.1 Overview .....	67
5.2 Communication Protocol .....	68
5.3 Packet Format .....	69
6. Communication with ACMA .....	82
6.1 Overview .....	82
6.2 CAMAC Modules and Commands .....	83
6.3 Communication Protocol .....	84
7. Hardware Operation and Control .....	88
7.1 BSD-A Single Action .....	88
7.2 BSD-A Continuous Commands .....	89

7.3 BSD-A Fast Data Acquisition.....	90
Acknowledgments .....	91
References .....	91

## 第1編 新CICUの概要

## 1. 概 要

CICU (CAMAC インターフェース制御装置) は、JT-60 計測装置で使用している CAMAC クレートと A7 データ処理設備ショット間処理計算機（以下 ISP という）を接続するための装置である<sup>1),2),3)</sup>。CICU の主な機能は計測装置を制御しているマイクロコンピュータ付補助クレートコントローラタイプA（以下 ACMA という）と ISP 内の応用システム間の通信を仲介すること、ISP の指示により計測データを高速で収集すること、及びデジタル入出力信号を取り扱うことの 3 つである。これらを実現するため CICU は PFU-1200 にチャネルインターフェース装置（M-Adapter； MADP）、バイトシリアルハイウェイドライバ（BSD-B）、及びデジタルユニット（ICUD-BB）を実装したものを使用していた。CICU として使用していた PFU-1200 はすでに製造されておらず、部品の供給や、ソフトウェア障害、改良時のプログラマの確保に支障が生じる恐れがあった。またワークステーションの性能向上と価格の低下は著しく、PFU-1200 の保守料と比較できるほどになってきた。

そこで今回、この CICU をワークステーションで置き換えることにした。ここで ISP とのインターフェースは MADP より Ethernet へ、BSD-B は VMEbus のバイトシリアルハイウェイドライバ（Kinetic 2140）へ置き換えることにする。ICUD-BB で実現していた機能のうち、デジタル出力については現在使用されておらず、また割り込み入力については MMI 計算機に同一機能があるので CICU からは削除する。CICU を利用している ISP 側の応用プログラムは CICU 情報交換（IRTRCV）、ACMA 情報交換（ACMIH）、高速データ収集プログラム（ACQ）、模擬 ACMA 情報交換（DPCM）、オンライン要求コマンド（ACMA）、CICU テストプログラムなどがある。これらの応用プログラムはユーザ SVC を介して CICU コミュニケータに要求を出し、CICU コミュニケータが CICU を駆動して所定の処理を行わせている。今回の置き換えではユーザ SVC のインターフェースに互換性を持たせることにより、これらの応用プログラムを無修正で新しい CICU に接続させることができる。新しい CICU の全体構成を図 1 に示す。

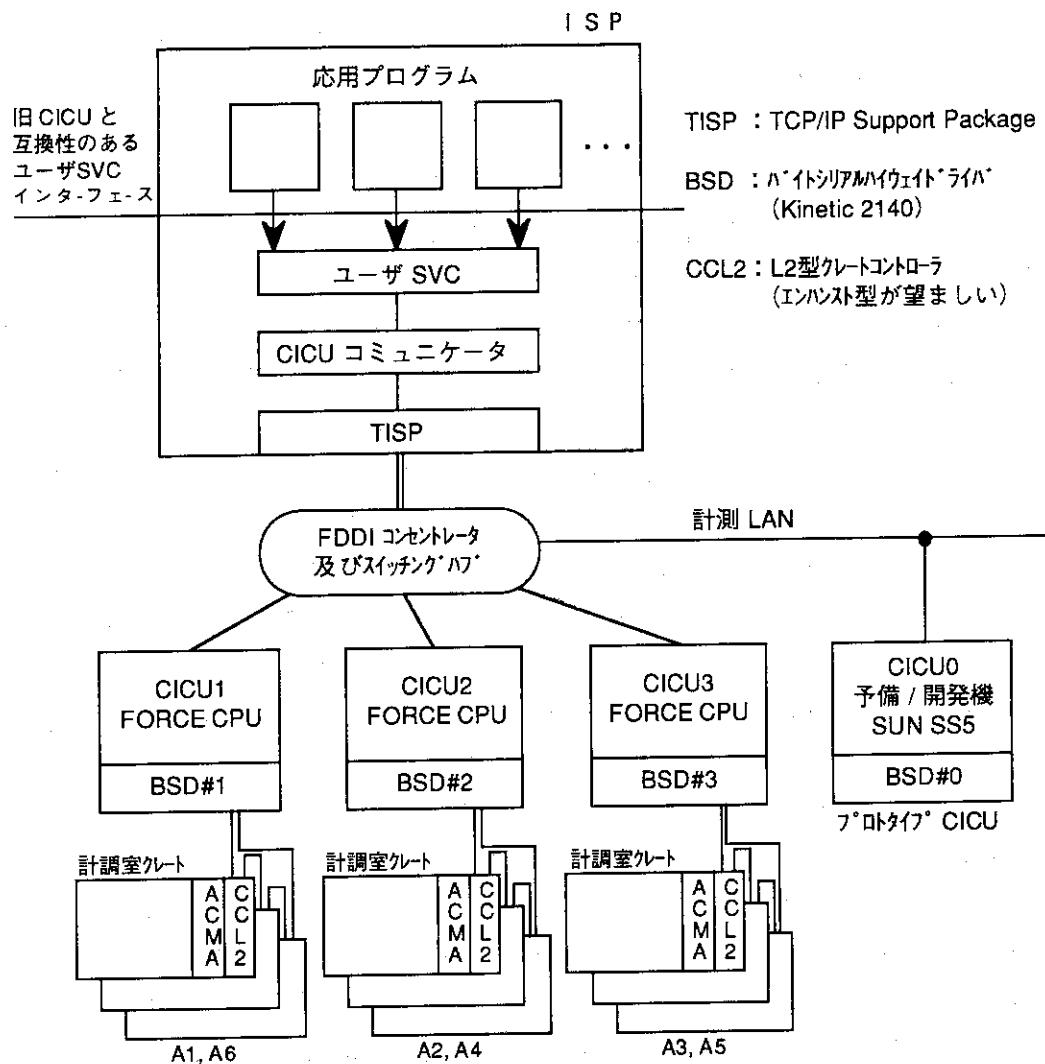


図1 CICUシステムの全体構成図

## 2. 基本システムの構成

### 2.1 CICU 基本システム

1996年1月からA1及びA6計測サブシステム用の旧CICU 1号機をプロトタイプCICUで置き換え、テストしている。プロトタイプCICUでは基本システムとして、Kinetic 2140と接続実績のあるSUN Microsystems社のSPARC Station 5ワークステーションと、Solflower社のSFVME-116 VMEbus変換装置を選んだが、VMEbus内蔵型のSUN互換機であるFORCE社のCPUシリーズを使用した方が、コスト、信頼性、及び設置スペースの点で優れているので、後者を採用することにする。

本体 FORCE CPU 5VT 3台

32MB メインメモリ

2GB ハードディスクドライブ

フロッピーディスクドライブ

ケーブル 他

VME シャーシ picoFORCE 3台

19インチラックマウントタイプ

スロット 1 CPU

ク 2

ク 3 Kinetic 2140

ク 4 Kinetic 2140

基本ソフト Solaris 2.5/5v

TCP チェックサムエラーが解決済のパッチレベル以上

#### 台数について

プロトタイプCICUでのCPU負荷率は10%以下であり、CICUソフトウェアも対応できるので1台のCPUに複数のシリアルハイウェイドライバを実装することも可能である。しかし、故障時の切り分けや交換のことを考えると、3台が独立している方が保守上すぐれているので、今回は3台独立方式とする。

#### 基本ソフトについて

CICUソフトウェアはSolaris 2.xのリアルタイムスケジューリングクラスを使って実行の優先度を調整しているので、Solaris 2.x以降でなければならない。又、プロトタイプCICUではSolaris 2.4のTCPモジュール内でチェックサムエラーが発生し、これが引き金となって著しい送信遅延が発生することが判明しているので、この問題が解決されていることが必要である。（BUG ID 1224148、パッチ 101945-38で解決）

## 開発システムについて

Kinetic 2140 デバイスドライバ及び CICU ソフトウェアのコンパイル及びリンクはプロトタイプ CICU で行い、実行形式ファイルを各 CICU にインストールすれば問題なく動作するはずなので、C コンパイラを含む開発システムを各 CICU に組み込む必要はない。

## 2.2 シリアルハイウェイドライバ

シリアルハイウェイドライバには、Kinetic 社の 2140 エンハンストシリアルハイウェイドライバ（以下 K2140 と呼ぶ）を使用する。K2140 は旧 CICU の BSD-B を代替するものである。高速データ収集は従来の BSD-B のハードウェア機能に依存した方法で行われてきたので、今回の置き換え時には K2140 とのハードウェア機能との差が問題となる。それらの差は、K2140 が DMA を行う際、データが 4 バイト境界になければならないこと、K2140 には Q リピートオーバーの検出機能が事実上ないこと、24 ビット転送において、上位 1 バイトが 0 でなく X'FF' になること、及び通常モードのデータ転送では、シリアルハイウェイからリプライが返ってきてから次のコマンドを出すことしかできないことの 4 点である。

### ① 4 バイト境界

高速データ収集機能転送処理では、読み込み処理に指示するメモリの読み込み開始位置を私用共有メモリの先頭にしている。共有メモリはページ境界から始まるので、自動的に 4 バイト境界に合う。16 ビット転送で奇数ワードを転送する場合、共有メモリ上には 2 バイトの余分なデータが転送されるが、ISP に転送する時はワード数からデータ長を決定しているので、余分なデータは ISP に転送されない。

### ② Q リピートオーバー

BSD-B では UQC モードでのハングアップを防ぐため最大 256 回連続して Q=0 となると転送を終結する機能がある。K2140 にも UQC モードで 7 秒間又は 15 秒間コマンドが終結しないとタイムアウトを検出するが、この時間はあまりにも長すぎる。そこでデバイスドライバ側でデータ転送量に応じたソフトウェアタイムアウトを設けている。タイムアウト発生時は K2140 をリセットする。この時 ACMA からの LAM が消失する可能性があるので、計調室クレートのハングデマンドを有効とし、ACMA 通信機能 LAM 受信処理で ACMA からのハングデマンドを正しく処理するようにしている。

### ③ 上位バイトの FF

24 ビット転送の時は CAMAC アクセス共通ルーチンの中で、上位 8 ビットを 0 にしている。

#### ④ コマンド送出タイミング

BSD-B ではコマンドのリプライがシリアルハイウェイから戻ってくる前に次のコマンドを送出する”B モード”という実行モードがある。本システムのように 1 つのループに多数の CCL2 や BUAO が接続され、リプライが遅くなる場合でもシリアルハイウェイの最高速度を維持できる。一方、K2140 の通常のデータ転送ではリプライを待ってから次のコマンドを出力することしかできないため、プロトタイプ CICU でのテストではデータ転送速度が 1/1.5 になってしまった。K2140 にはエンハンストモードと呼ばれる Kinetic 社独自の転送方法があり、これを使えば従来と同様な転送速度が得られるので、これを採用した。ただしこの転送方法は、エンハンストモードを受け入れられる CCL2 でないと実行できない。プロトタイプ CICU では A6c と、A6f のみエンハンストモードの CCL2 に交換しているが、3 台の CICU を交換する場合には他の計測サブシステムのうち、データ量の多いものをエンハンストモードの CCL2 に交換する必要がある。エンハンストモードでの転送では未調査のトラブルが発生している。

① BSD-A にシリアルハイウェイ速度 5MHz ではアクセスできない。

2.5MHz では問題ない。

② LAME にアクセスできない。

全面的にエンハンストモードに移行する前に CAMAC データウェイの信号をモニタし、CAMAC 規格との関係を調べてどちらに規格逸脱があるのか明らかにした方が良いと思われる。

シリアルハイウェイドライバは、旧 CICU ラック内にある BSD-SW を経由して BUAO に接続する。K2140 側の D ポートケーブルに特に細工をしなくとも、旧 CICU 4 号機からの指令で BSD-SW を切り換えられることがわかった。CE による保守点検や障害切り分け作業の際、ケーブルの接続替えを手作業で行う必要はない。

### 2.3 シリアルハイウェイドライバ用デバイスドライバ

Solaris 2.4 のデバイスドライバ開発手順に基づいて K2140 のデバイスドライバを作成している<sup>4)</sup>。FORCE CPU 上でも問題なく動作することを確認している。

### 2.4 計測 LAN

CICU は 1 台当たり約 250KB/秒の速度でデータを ISP に転送するので、3 台接続した場合には Ethernet の能力を越えてしまう。そこで 3 台導入時には図 2 のようにスイッチングハブ、FDDI 経由で ISP と接続する。

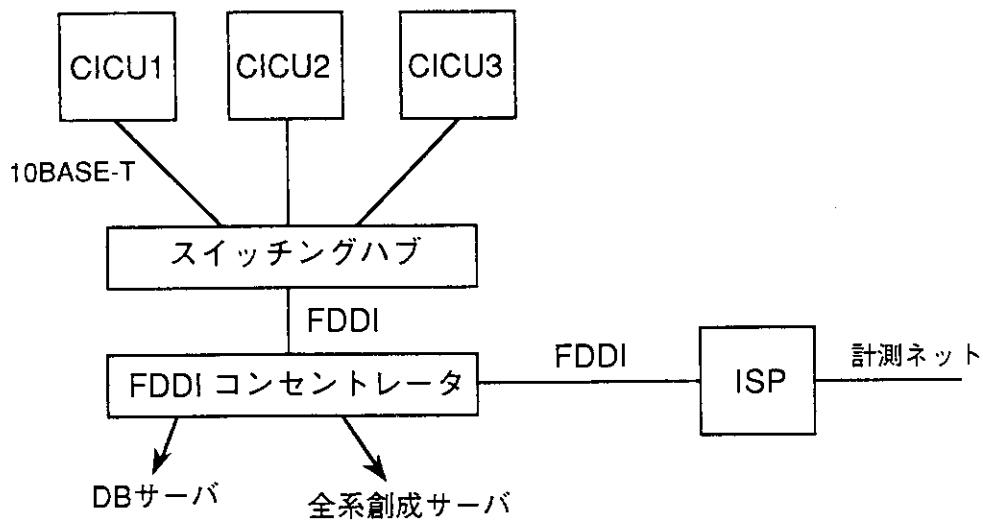


図 2 FDDI 直結型のスイッチングハブ使用

## 2.5 旧 CICU システムとの関係

旧 CICU 4 号機のみを保守用に残し、他をすべて撤去する。残す 4 号機も ISP との接続や、ディジタル入出力が不要となるため CICUSW や、ICUD-BB、RCVI モジュールが不要となる。図 3 に旧 CICU システムと新 CICU システムの構成を示す。

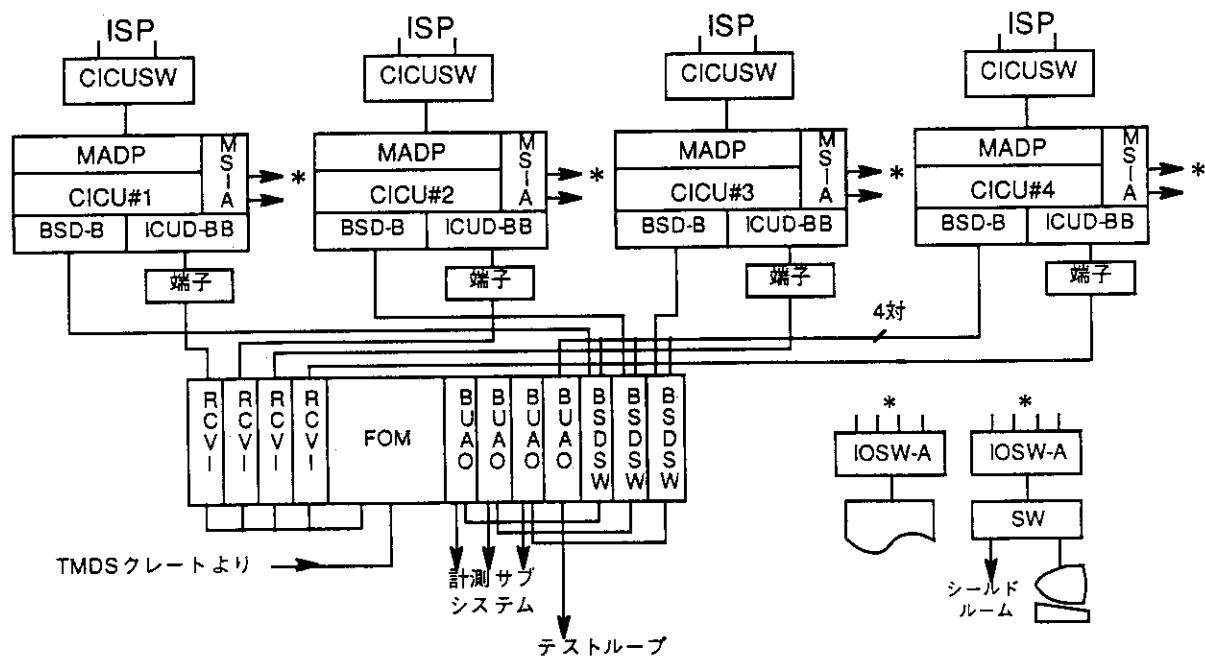


図 3 a 旧 CICU システム

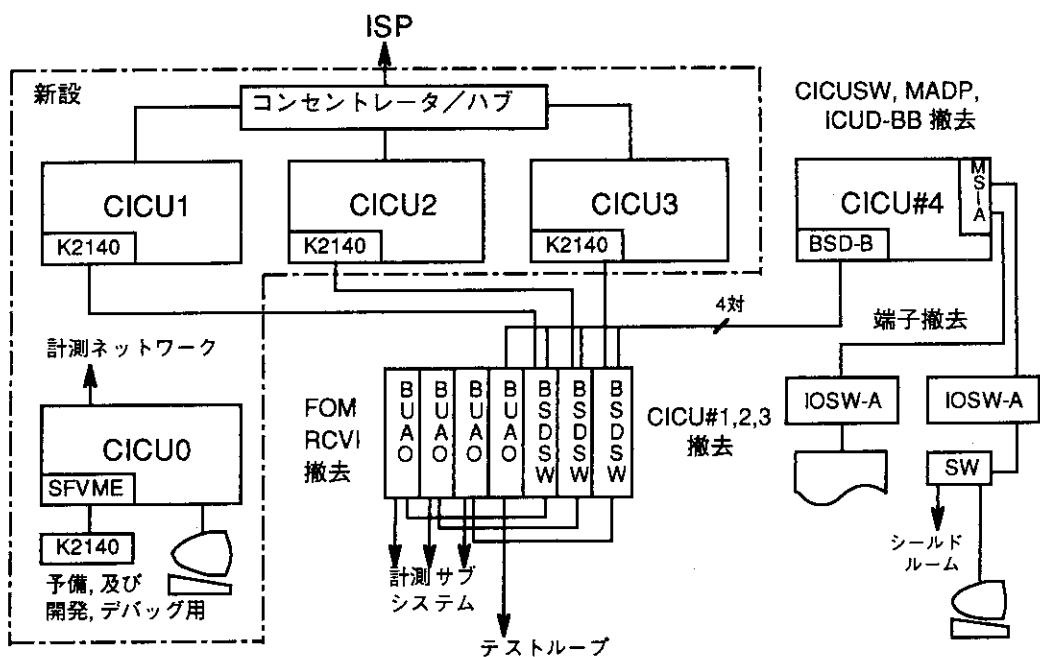


図3 b 新 CICU システム

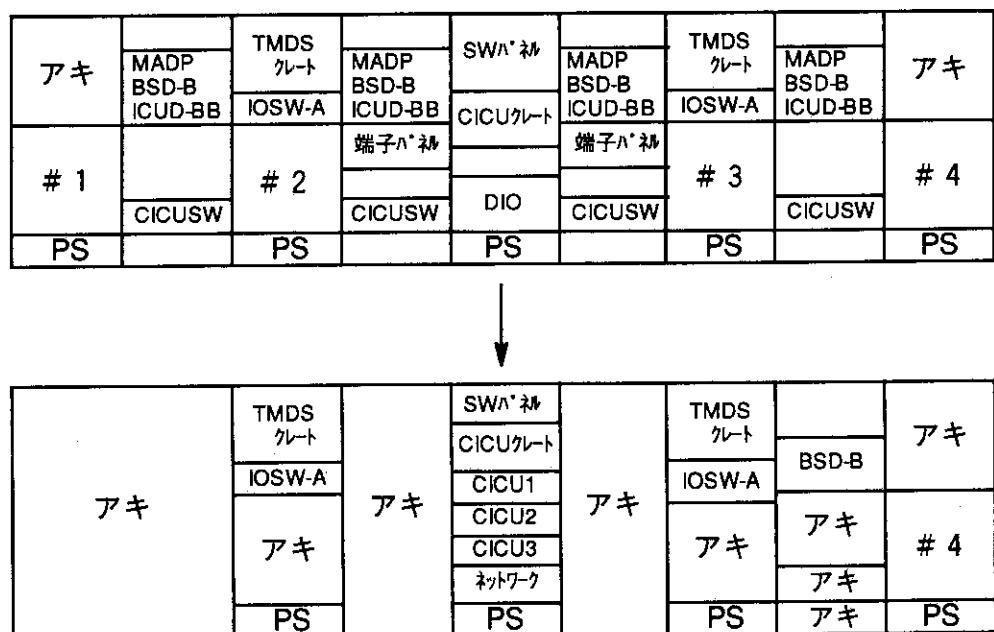


図3 c 新 CICU システムのラック配置案

### 3. CICU ソフトウェア

CICU ソフトウェアは CICU コミュニケータからの各要求を処理する ACMA 通信機能、高速データ収集機能、単一要求処理機能、ダミー処理機能、全常駐プロセスを管理し、共有メモリ等の退避／復元を行うシステム監視機能、及びコンソールからの要求を処理するコンソール機能から構成される。これらの各機能は1つ又は複数のプロセスで構成される。各プロセスはシステムに常駐するものと、CICU コミュニケータからのコネクション開設ごとにインターネットサーバ (inetd) より起動されるものがある。

#### 3.1 ACMA 通信機能

ACMA 通信機能は ACMA 通信処理、LAM 受信処理、RQPUT 中継処理、及び RQGET 中継処理の4つの処理より構成されている。ACMA 通信処理、及び LAM 受信処理は1つの K2140 に対して1つ存在し、ACMA との通信を管理する。RQPUT 中継処理は ISP 起動 ACMA 交信用の処理で、コミュニケーションからの接続要求がある毎に inetd より起動される。RQGET 中継処理は ACMA 起動 ACMA 交信用の処理で、同様に inetd より起動される。4つの処理はメールボックスを介して通信する。メールボックス (ACMA 通信要求 MBX) は1つの K2140 毎に1つ存在し、メッセージタイプを上手に使うことにより通信を制御する。コミュニケーションとのコネクションが閉設された場合、RQGET 中継処理は ACMA から受信した ENQ メッセージを ISP に送信できない。送信不可の場合は同メッセージを ACMA 通信要求 MBX へ戻し、自身は終了する。これにより、すでに他のコネクションが開設されている場合は、そちらの RQGET 中継処理を経由して ENQ メッセージを ISP へ送信することができる。メッセージタイプの使い方を示す。

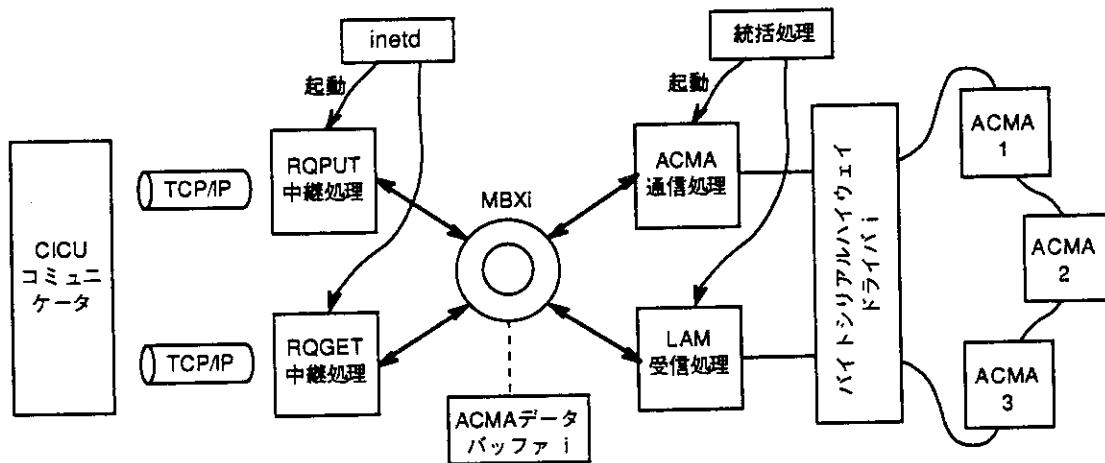
- ① ACMA 通信処理は通常すべてのメッセージタイプを受信している。これはメールボックスにゴミが残るのを防ぐためである。この段階で有効なメッセージは RQPUT 中継処理からの通信開始要求 (ENQ メッセージ) と LAM 受信処理からの ENQ メッセージだけである。他のメッセージは読み捨てる。
- ② ACMA 通信処理は RQPUT 中継処理又は LAM 受信処理からの ENQ メッセージを受信すると、対応する ACMA との通信を開始する。つまり、ACMA に対して ENQ を送信するか、又は RQGET 中継処理に通信開始要求 (ENQ メッセージ) を通知し、時間監視を開始する。時間監視中、ACMA 通信処理はすべての ACMA か、又は通信中の中継処理からのメッセージのみ受信する。
- ③ LAM 受信処理は ACMA からの割り込みがあれば、その ACMA のクレートアドレスに対応したメッセージタイプのメッセージを作成してメールボックスに通知する。ACMA 通信処理が特定の ACMA - 中継処理と通信中の場合、他の ACMA からのメッセージは ACMA 通信処理が受信し、Busy-Reject をその ACMA に通知する。

- ④ RQPUT 中継処理はコミュニケータからの通信開始要求 (RQ パケット) を受信すると、特定の（どの ACMA クレートアドレスとも一致しない）メッセージタイプのメッセージをメールボックスに通知する。ACMA 通信処理が特定の ACMA 一中継処理と通信中の場合、他の中継処理からのメッセージはメッセージタイプが異なるので ACMA 通信処理は受信しない。このメッセージは通信が終了するまでメールボックス内に残され、通信終了と同時に①の処理が発生する。
- ⑤ LAM 受信処理又は中継処理からの応答のみを受信した ACMA 通信処理は中継処理又は ACMA にその応答を通知する。中継処理への応答はプロセス id に対応するメッセージタイプで行われるので、他の中継処理はこのメッセージを受信できず、通信中の処理だけに通知できることになる。
- ⑥ 通信終了直後、ACMA 通信処理は ACMA のクレートアドレスに対応するメッセージタイプと RQPUT 中断処理からの通信開始要求に対応するメッセージタイプのみ受信する。これは最後に中継処理に通知したメッセージを自分自身で受信しないためである。30秒後、再びすべてのメッセージタイプを受信するようにして①へ戻る。

旧 CICU の ACM 交信処理では何らかの理由で通信開始要求を処理できない場合、ISP には CICU 戻りコード X'FFFF' を、ACMA には MNO=X'FF' のメッセージを返し、Busy-Reject であることを通知、応用プログラムは一定時間後、再度通信開始要求を行うようになっていた。新 CICU では処理が単に待たされるように設計することも可能であった。このように設計した場合、要求元には Busy-Reject は戻らず、単に処理可能になるまで待たされるだけとなるので、応用プログラムは再試行の処理が不要となって好ましいと言える。一方、CICU での処理がさらに遅れると、ISP ではコミュニケータがコネクションを切断して要求元にエラーコード 3-3 を返し、ACMA では ISP 通信ソフトが要求元に ISP ダウンを通知してしまう。このタイムアウト時間は次のとおりである。

ISP 側応用プログラム	受信待ち時間	送信待ち時間 (参考)
ACQ, DACM	不定	デフォルト値
ACMIH	デフォルト値	10秒
IRTRCV	9999秒	9999秒
ACMA コマンド	デフォルト値	デフォルト値
ACMA 側プログラム		
ISP 通信ソフト	10秒	—

デフォルト値の場合はコミュニケータの初期設定ファイルより値が補充されるので、応用プログラムを変更せずに調整可能である。ACQ, DACM が不定なのは初期値の代入忘れがあるので、デフォルト値とするよう、修正が必要である。IRTRCV は十分な時間なので、問題ない。新 CICU では高速データ収集と並行して ACMA 通信が可能ではあるものの、放電完了時に多数の ACMA から同時に要求が発生することを考慮すると ACMA 側の 10 秒固定のタイムアウト時間では不安が残る。そこで今回は、ISP からの要求は待たせることとし、タイムアウト時間は実機での遅延時間を見て調整することにした。一方、ACMA 側は調整不可能なので、従来どおり Busy-Reject を返すようにした。ACMA 通信機能及びメッセージタイプの意味を図 4 に示す。



メッセージ・タイプ*	意味	送 信	受 信
1 ~ 62	ACMA j と通信中 (j = 1~62)	LAM 受信, 中継 (通信中)	ACMA 通信
64	ISP 起動通信開始要求	RQPUT 中継 (初回)	ACMA 通信
65	ACMA 起動通信開始要求	ACMA 通信	RQGET 中継 (初回)
100 ~ (pid + 100)	中継処理への応答	ACMA 通信	中継 (通信中)

注) CICU コミュニケータとの各 TCP/IP コネクションは、SVC 番号、通信要求コード、方向、相手ホスト、ポート番号、相手ループ番号の各属性を持っている。CICU コミュニケータは CICU に接続時、最初にループ番号及び方向標識を送信する。CICU では、この情報でシリアルハイウェイドライバのオープン、又は対応する MBXi の割り当て、あるいは RQPUT/RQGET の処理の違いを認識する。

図 4 ACMA 通信機能

### 3.2 高速データ収集機能

旧 CICU では高速データ収集は BSD-A、BSD-B 及び MADP のハードウェア機能を最大限に利用して実現されていた。その結果、高速ではあるものの柔軟性に欠けており、それをすべてソフトウェアでカバーしていた。例えば、データ転送は一時的に停止させることができない。その結果、ISP の処理が遅れるとオーバーランが発生してしまうので、高速データ収集の応用プログラムはこれに対応しなければならなかった。また MADP がデータチェインモードに切り換わったかどうか、ISP に CICU のセンス情報を取り込ませるために、通常の I/O 機器では故障時にしか使用しないユニットチェックビットを使用するなど特殊な工夫をしていた。新 CICU では BSD-A のハードウェア機能だけをそのまま使って同等の転送速度を得ている。ただし、2.2④で述べたように、K2140 の通常の動作では遅延が大きいので、エンハンストモードで転送を行うようにしている。

高速データ収集機能は転送処理と読み込み処理より構成される。転送処理はコミュニケーションからの接続要求がある毎に inettd より起動され、BSD-A の制御権を得た後、BSD-A のマスクレジスタの退避等の前処理を行う。次に最大 1MB のデータ収集要求を私用共有メモリの大きさ（256KB）に分割して読み込み処理に通知する。読み込み処理は転送処理から起動され、BSD-A のコマンドバッファの設定を行った後、BSD-A のデータバッファよりデータを私用共有メモリに転送し、実行結果を BSD-A より読み込んで転送処理に返す。処理を 2 つに分けたのは BSD-A からの読み込みと、ISP へのデータ転送を並行処理させるためである。当初の設計では転送処理が BSD-A のコマンドバッファの設定を行い、読み込み処理で 1MB のデータを適当に分割して読み込む予定だったが、K2140 のエンハンストモードの Q リピートモードでは、K2140 が出すコマンド数を正確に制御できないことがわかったので、この方法はあきらめた。高速データ収集機能を図 5 に示す。なおメールボックス、共有メモリ共、転送処理が私用属性で作成／消去するので、メッセージタイプによるメッセージの複雑な選別は必要ない。その代わり、BSD-A の競合を防ぐため BSD-A ロックセマフォを使用する。

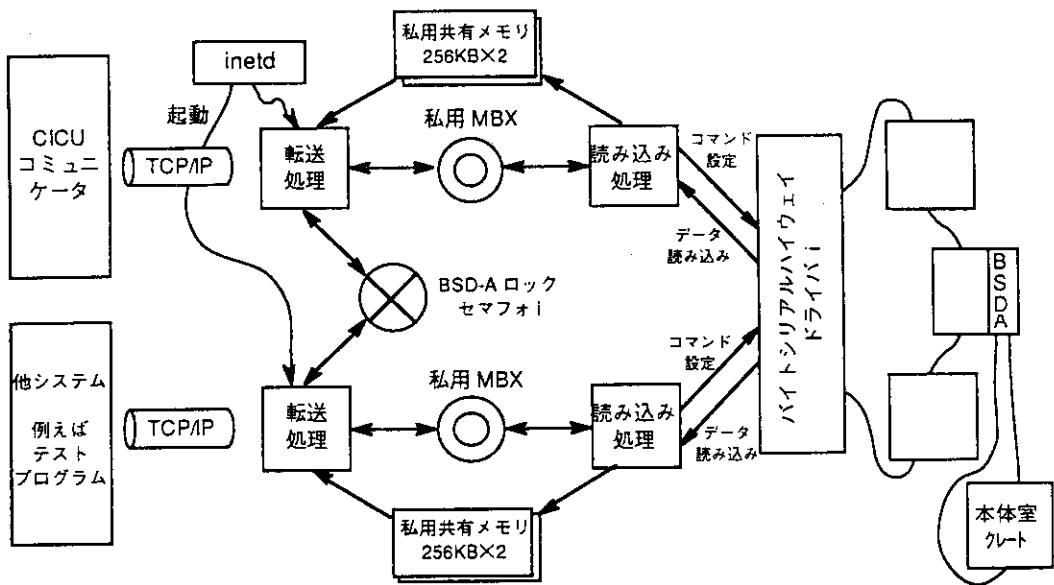


図5 高速データ収集機能

### 3.3 単一要求処理機能

单一要求処理機能は本体室アクセス処理、計調室アクセス処理、オンライン要求処理、及び初期設定要求処理の4つの処理の総称である。これらはいずれもコミュニケーションからの接続要求がある毎にinetdより起動され、コミュニケーションより指示されたシリアルハイウェイドライバをオープンして必要な処理を行う。旧CICUとの対応を下表に示す。

新 CICU	旧 CICU	応用プログラム
本体室アクセス処理	本体室クレートアクセス要求	ACQ
計調室アクセス処理	一般 CAMAC サポート処理	DACM
オンライン要求処理	共通機能、オンライン要求	ACMA (コマンド)
初期設定要求処理	共通機能、初期設定要求	IRTRCV

单一要求処理機能を図6に示す。

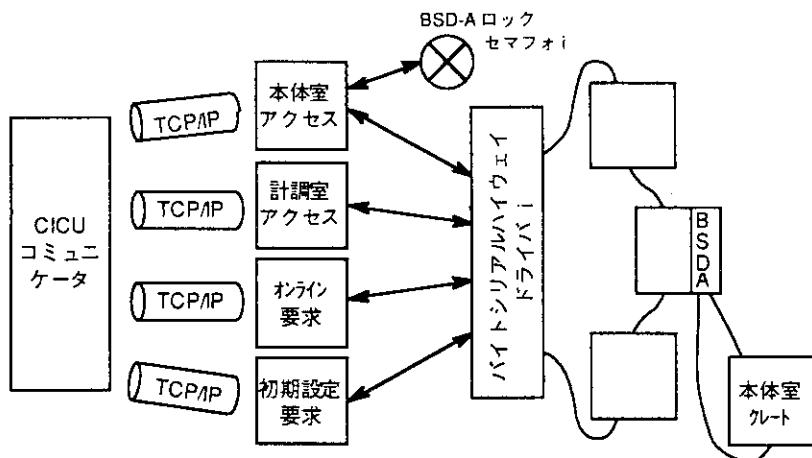


図 6 単一要求処理機能

### 3.4 システム監視機能

システム監視機能は統括処理とロギング処理から構成される。統括処理は起動後、共有メモリの復元、メールボックス、セマフォの割り当てを行った後、接続しているシリアルハイウェイドライバ毎に ACMA 通信処理、及び LAM 受信処理を起動し、これらのプロセスの終了監視と再起動処理を行う。本処理が終了する時は共有メモリをファイルに退避する。シグナルを受信して終了する場合も同様である。ロギング処理は独自のプロセスは持たず、組み込みサブルーチンとして、他の各処理に組み込まれ、イベントロギング、通信ロギング、及びエラーロギングを行う。多数のプロセスから書き込まれるイベントロギングとエラーロギングについてはセマフォによるガードを行う。

### 3.5 コンソール機能

主に障害追跡のため、CICU のコンソールより CAMAC コマンドの発行、ロギングの出力、シリアルハイウェイのループテスト、CCL2 のオンライン、及び共有メモリの参照を行うことができる。

### 3.6 削除した機能

旧 CICU にあった下記機能は削除、ダミールーチン、又は代替処理で対応している。

#### ① ICUD-BB サポート機能

新 CICU にはデジタル入出力がないので削除したが、IRTRCV が受信要求を出すので、とりあえずダミールーチンで対応するようになっている。CICU を 3 台とも入れ替えた場合、IRTRCV は（初期設定要求を除いて）無意味なものとなるので、この段階で初期設定要求のみを行って終了するプログラムに置き換えた方が

良いと思われる。タイミング信号は親タイミングクレートから MMI を経由して ISP に通知されるようになっているので、スケジューラのルーティングパラメタを後日修正する。

② 通信管理部

RQ, DT, RS, RE の各パケットフォーマットだけ旧 CICU に合わせているが、他はすべて TCP/IP による通信に置き換えている。また高速データ収集時のデータも拡張 DT パケットで ISP に送信するように変更している。3.1 で述べたように ISP に対する Busy-Reject はなくなった。

③ 処理管理部

旧 CICU では各要求処理中に他の要求を受け付けられるか判断しているが、新 CICU では各要求は完全に独立して処理されるので、要求が拒否されることはない。これは高速データ収集処理においても例外ではなく、高速データ収集中に他のクレートの ACMA とは通信が可能となるので、計測完了を早く ISP に伝えることができるようになる。ただし、高速データ収集自身の時間はあまり変わらないので、このことは計測完了の通知完了から高速データ収集完了の受信までの時間を長引かせることになる。幸いプロトタイプ CICU では問題とならなかつたが計測完了から高速データ収集完了まで時間監視を行っている計測サブシステムがある場合は、タイムアウトの危険があるので事前に調べておく必要がある。

④ MDR サポート処理

MDR はすでに存在しないので、対応する処理もない。

⑤ 共通機能

オンライン要求、及び初期設定要求以外はすべて削除した。

## 4. 運用サポート

### 4.1 起動・停止

#### 4.1.1 起動

- ① まず CICU クレートの電源入、及び  $N = 17 \sim 19, 20 \sim 22, 23 \sim 25$  の BSDSW の A ランプが点灯していることを確認する。もしなっていない場合は CICU 4 号機から BSDSW の切替えを行う。
- ② 図 7 a に示す 3 つの電源を入れる。電源を入れると UNIX が自動的に立ち上がり、CICU ソフトウェアも起動する。
- ③ ISP から CICU コミュニケータを立ち上げる。これで通信可となる。

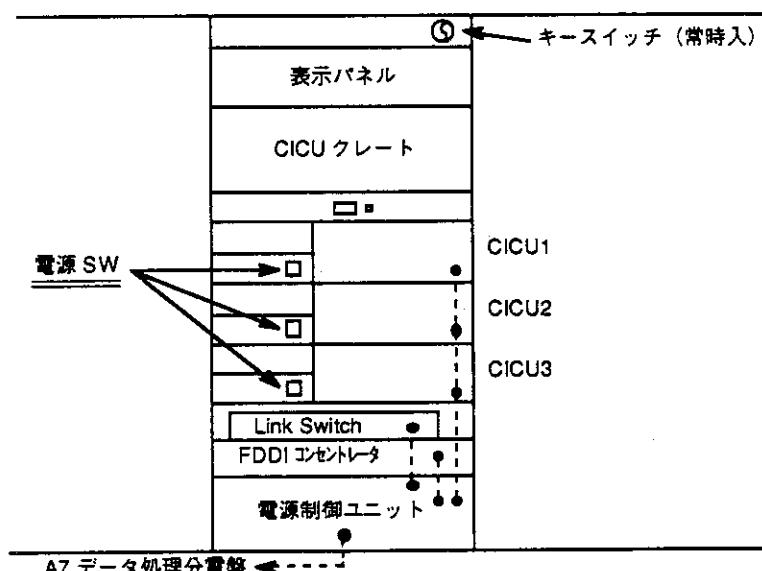


図 7 a CICU システムの電源

#### 4.1.2 停止

- ① 他のワークステーションからログインする。  
telnet cicu1e [return] (又は cicu2e, cicu3e)  
 又は telnet 157.111.131.67 [return] (又は 68, 69)  
 $\vdots$   
cicu1% su [return]  
password: パスワード [return]  
cicu1# halt [return]  
 connection closed by foreign host.  
 1 分ほど待ち、電源を切る。

コンソール CRT が接続されている場合は、そこから root でログインし、halt コマンドを投入することもできる。

#### 4.1.3 CICU プロトタイプの起動・停止

- ① [起動] 図 7 b の (1) ~ (3) の順序で電源を入れる。なお CRT (3) は必要になつた時あとで入れることもできる。電源を入れると UNIX が自動的に立ち上がり、CICU ソフトウェアも起動する。

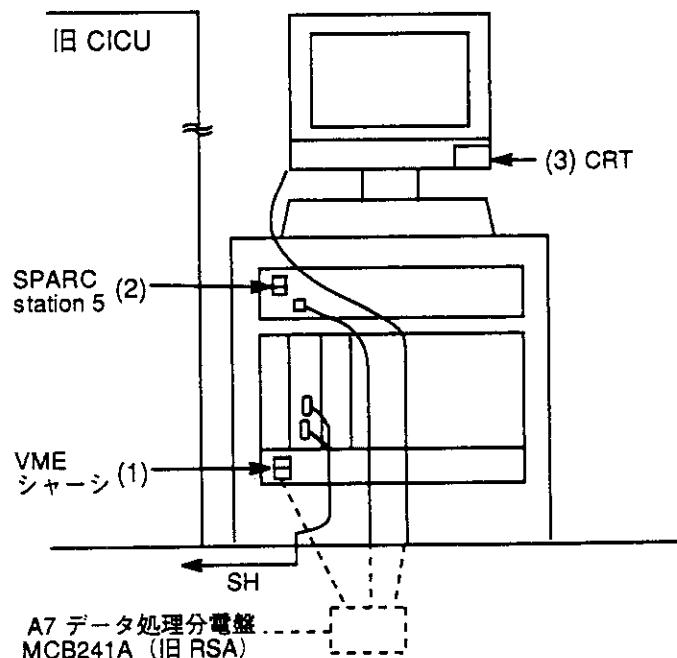


図 7 b CICU プロトタイプシステム電源系

#### ② [停止]

```
console login: の状態から
  console login: root [return]
  password: パスワード [return]
  :
  # halt [return]
  :
  ok
```

ok プロンプトが出たら (3)(2)(1) の順序で電源を切る。

#### 4.2 ログインとログアウト

以下すべてのコマンドはユーザ名 cicu でログインした後のみ実行できる。

##### CICU コンソールからのログイン方法

プロトタイプ CICU の場合はコンソールから、他の CICU はコンソール用 CRT を接続すればコンソールからログインできる。

```
console login : cicu [return]
password :xxxxxxxx [return] (エコーされない)
```

プロトタイプ CICU の場合は OpenWindows が自動起動する。メニューの中からコマンドツール又はシェルツールを起動して、その中で作業する。作業が終了したら Openwindow の「終了」を選ぶ。その後 logout コマンドを入力する。これで元の状態に戻る。

他のワークステーションから CICU に入ることもできる。

##### プロトタイプ CICU (予備、開発用)

diag7% telnet cicu0 [return]	又は telnet 157.111.131.49 [return]
CICU1 (A1, A6 ループ)	
diag7% telnet cicule [return]	又は telnet 157.111.131.67 [return]
CICU2 (A2, A4 ループ)	
diag7% telnet cicu2e [return]	又は telnet 157.111.131.68 [return]
CICU3 (A3, A5 ループ)	
diag7% telnet cicu3e [return]	又は telnet 157.111.131.69 [return]

作業が終了したら logout しておく。

#### 4.3 CAMAC コマンドの発行

CICU コンソールから CAMAC コマンドを発行することができる。コマンドは計調室クレートに対してだけでなく、本体室クレートに対しても出すことができる。本体室クレートにコマンドを出す場合、ACMA と競合する可能性があるので、ACMA が停止中か、又は ACMA を IPL しても問題ない時だけ行う。

```
cicu0% ccamac [return]
```

ここでコマンド入力待ちとなる。プロンプトはない。この状態から抜けるには ctrl-D (コントロールキーを押したまま d キーを打つ。これは End of File を示す) を入力する。コマンドは次のように入力する。

```
C3 N8 c1 n23 a0 f0/ [return]
```

スラッシュで実行である。スラッシュを入れないと何も実行しない。C (大文字) と数字で計調室クレートアドレス、N (大文字) と数字で BSD-A ステーションアドレス、c, n, a, f は本体室クレートのクレートアドレス、ステーション番号、サブアドレス、及びファンクションを表す。この他に d と数字で 10 進データを入力できる。C, N, c, n, a, f, d は 1 回入力すると覚えているので必要な所だけ変更してコマンドを再実行させることができる。

C3 N8 c6 n23 a0 f0/ [return]

c5/c4/c3/c2/c1 [return] ← c5 ~ c1 に同じコマンドを実行する。

////////// [return] ← c1 n23 a0 f0 を 10 回実行する。

データに 16 進数を含めたい時は a ~ f 又は A ~ F を含めるか、文字 x を含める。

d135 . . . 10 進数 135 を指定

d16a . . . 16 進数 0x16a を指定

dx135 . . . 16 進数 0x135 を指定

計調室クレートに対してコマンドを出す時は c を 0 とする。

c0 n1 a0 f0/ [return]

応答には "local" が表示され、計調室アクセスであったことを示す。

コンソールから毎回コマンドを入力する代わりに実行するコマンドをファイルに入れておき、まとめて実行することができる。

cicu0% ccamac ファイル名 [return]

ファイル名を省略すると標準入力、すなわちコンソールからの入力待ちとなる。

ファイル中のコマンドの記述方法はコンソールから入力するのとまったく同じである。

#### 4.4 ロギングの出力

CICU は ACMA 通信ロギング、イベントロギング、及びエラーロギングの 3 つのロギングを行っている。

cicu0% clog ロギング名 [出力数] [return]

出力数を省略すると最新の 10 コが出力される。出力は常に逆順になる。すなわち最新のロギングが先頭に表示される。

- ACMA 通信ロギング

cicu0% clog mbx0 [出力数] [return]

CICU と ACMA との通信 (CCRR, CCRS) ロギングが出力される。

- イベントロギング

cicu0% clog evnt [出力数] [return]

プロセスの起動・停止、CAMAC コマンドの実行結果のロギング、各処理の処理結果のロギングが出力される。ロギングに表示されるプロセス名は以下のとおりである。

hsd0 . . . 高速データ収集処理	rac0 . . . 本体室アクセス処理
acm0 . . . ACMA 通信処理	lac0 . . . 計調室アクセス処理
lam0 . . . LAM 受信処理	onl0 . . . オンライン要求処理
pgt0 . . . 中継処理	ini0 . . . 初期設定要求処理
svc . . . 統括処理	irt0 . . . IRT 処理

・エラーロギング

cicu0% clog err [出力数] [return]

CICU 内の応用プログラムが検出した各種のエラーのロギングが出力される。統括処理が回復不可能なエラーを検出した場合はコンソールの画面上に直接エラーメッセージが表示され、応用プログラム全体が停止する。この時このメッセージはロギングには残らない。

・ファイルへの保存

障害追跡のためロギングをファイルに残す場合は、標準出力をファイルにリダイレクトする。

cicu0% clog err 100 > errlog\_96\_02\_19 [return]

これで最新のエラー 100 個がファイル "errlog\_96\_02\_19" に保存される。

#### 4.5 ループテストとオンライン

CAMAC のシリアルハイウェイをテストする目的で CICU のコンソールからループテストを実行することができる。なお、以下のコマンドは control+z で停止する。

cicu0% looptest n [return]

n は計調室クレートアドレスである。looptest コマンドは指定されたクレートの N23 の LAME に対し 4KB のデータを書き込み、直ちに読み込んでデータの比較を行う。コマンドのエラー又は比較エラーで停止する。書き込み、読み込み及び比較を 100 回行うごとにコンソールにループ回数を表示する。

2.2 で述べたようにエンハンストモードでは LAME にアクセスできないので、BSD-A のコマンドバッファに読み書きするコマンドも用意してある。

cicu% emodetest n [return]

n は計調室クレートアドレスである。emodetest コマンドは指定されたクレートの N8 又は N9 の BSD-A のコマンドバッファに対し 4KB のデータを書き込み、直ちに読み込んでデータの比較を行う。コマンドのエラー又は比較エラーで停止する。N8 か N9 かは自動的に判定される。またこのコマンドは CCL2 のステータスピットを見てエンハンストモードか、通常モードか自動的に判定し、可能ならばエンハンストモードで動作する。コマンドオプションとして "-n" がある。

cicu% emodetest -n n [return]

これは CCL2 がエンハンストモードをサポートしていても、通常のモードで動作する。このオプションと反対の動作をする "-e" オプションもある。

今回のループテストの特徴は、複数の計調室クレートに同時に実施しても問題ないことである。emodetest も同様である。

cicu% looptest 1 & looptest 2 [return]

これで計調室クレートのアドレス 1 とアドレス 2 が同時に（と言っても実際には交互にだが）テストできる。書き込みと読み込みは不可分のコマンドとして実行されているので、同一クレートに対して同時に複数のループテストを実施するこ

とさえできる。

ハードウェアを交換後ループテストを実施することを考えて CCL2 をアンバイパス、オンラインにするコマンドもある。

cicu% online n [return]

n は計調室クレートアドレスである。ハイウェイの物理的切替えを伴うため、このコマンドは通常エラーを出力するが、問題ない。

#### 4.6 旧 CICU との切り換え

旧 CICU 4 号機からの指令により BSD-SW を切り換えて、各シリアルハイウェイを旧 CICU 4 号機から制御することができる。これはハードウェアの保守点検や、4.5 のループテストでは検出できないハードウェアの障害を追跡する時に使用する。

#### 4.7 プロトタイプ CICU との切り換え

CICU に障害が発生した場合、1 台だけならプロトタイプ CICU と交換することができる。ここでは CICU1 が故障した場合を想定して切り換え手順を示す。

- ① プロトタイプ CICU の電源を入れ、システムが立ち上がるのを待つ。次に放電シーケンスの停止を確認する。またすべての ACM-A をオフラインにする。
- ② CICU コミュニケータを停止する。IRTRCV からエラーメッセージが、又 ACMIH からはエラーメッセージが定期的に発生するはずである。（一度どうなるか、テストした方が良いと思われる。）
- ③ CICU クレートの電源を落とし、ステーション 17 ~ 19 の BSDSW の A ポートを切り離し、プロトタイプ CICU からのケーブルをこのポートに取り付ける。

障害 CICU	ループ	BSDSW ステーション
CICU1	A1, A6	17 ~ 19
CICU2	A2, A5	20 ~ 22
CICU3	A3, A4	23 ~ 25

- ④ ISP の CICU コミュニケータ定義ファイルを編集する。

HOST = 'CICU1' → HOST = 'CICU0' (8ヶ所)

HNAME(1) = 'CICU1' → HNAME(1) = 'CICU0' (1ヶ所)

とする。

- ⑤ CICU クレートの電源を投入する。

- ⑥ CICU コミュニケータを再立ち上げする。IRTRCV が初期設定を行うことを確認する。また ACMIH からのエラーメッセージも停止するはずである。このあと ACM-A をオンラインにする。

## 5. システムの構築

このセクションでは、バックアップテープのない状態からプロトタイプ CICU を構築する方法を説明する。基本的なやり方は他の CICU でも同じだが、異なる点を 6. にまとめてある。

### 5.1 ハードウェアのインストール

#### 5.1.1 SFVME のインストール

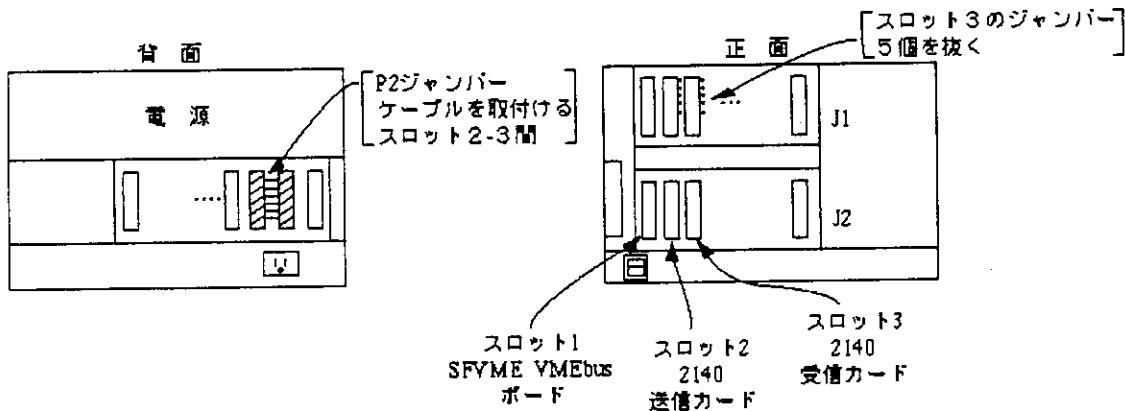
平成 6 年度に「CICU 計算機基本システム」の一部としてソルフラワー社の SFVME-110B を CTC 経由で購入したが、この製品は DMA で CAMAC 側に書き込むとデータが化ける設計上の不具合がある。平成 8 年 8 月には改良版でも試したが完全な解決には至っていない。日商エレクトロニクスの野口氏との話し合いにより、問題が解決するまで旧製品である SFVME-116 を借用することにしている。不良のSFVME-110B はファイル室に保管してある。SFVME-116 は借用品であるためマニュアルはないが、MMI 計算機更新の時購入した「SFVME SERIES USER'S MANUAL and INSTALLATION GUIDE」や、SFVME-110B 付属の「SFx10 SERIES USER'S MANUAL」が参考になる。また、SPARC station 5 については「SPARC station 5 Installation Guide : 4, 5 システムユニットの開閉、7 SBus カードの取付け」を参照する。概略は以下のとおりである。

- ① SFVME SBus ボードのジャンパーは借用時のままとなっている。
- ② システムユニットの電源を落とす。
- ③ 「SPARCstation 5 Installation Guide : 4 システム本体を開ける」の手順に従ってカバーを固定しているネジを緩め、ロックブロックを外す。カバーを持ち上げ前面側に開く。
- ④ 「SPARCstation 5 Installation Guide : 7 SBus カードを取り付ける」の手順に従って slot2 のフィラーパネルをはずし、SFVME のSBus ボードを押し込む。
- ⑤ 「SPARC station 5 Installation Guide : 5 システム本体を閉じる」の手順に従ってカバーを閉じ、ロックブロックとカバーを固定しているネジを締める。

### 5.1.2 VMEbus ボードのインストール

モトローラ MVME945B VME シャーシには SFVME VMEbus ボード、2140シリアルハイウェイドライバ受信及び送信ボードの3枚の VMEbus ボードを取り付ける。

- ① VME シャーシの背面パネルをはずし、2140 に付属の P2 ジャンパーケーブルを P2 のスロット 2 と 3 の間にバックプレーンの裏側から取り付ける。



- ② VME シャーシの前面パネルをはずし、バックプレーンの表側にある BG 及び IACK ジャンパーのうちスロット 3 用の 5 つのジャンパーピンを抜く。
- ③ SFVME の VMEbus ボードのジャンパーを確認しておく。すべて借用時ままになっている。
- ④ 2140 シリアルハイウェイドライバ送信ボードのジャンパーを確認する。ボード上の 4 本のネジをはずし、「MSB」と書いてあるジャンパーが入っていることを確認する。また前面パネルの速度選択を 2.5MHz に変更する。詳細については「Model 2140-Z1A Enhanced Serial Highway Driver for VMEbus Systems INSTRUCTION MANUAL p4 CHART I」を見る。
- ⑤ 2140 シリアルハイウェイドライバ受信ボードのジャンパーを確認する。すべて工場設定のままである。
- ⑥ SFVME の VMEbus ボードをスロット 1 に、2140 シリアルハイウェイドライバ送信ボードをスロット 2 に、同じく受信ボードをスロット 3 に差し込む。残った前、背面パネルをすべて取り付ける。
- ⑦ SFVME の SBus ボードと VMEbus ボードを接続ケーブルでつなぐ。また 2140 シリアルハイウェイドライバ送信カードの D ポートをハイウェイに接続する。

## 5.2 OS のインストール

SPARC station 5 に日本語 Solaris 2.4 をインストールする。

「日本語 Solaris 2.4 Hardware : 11/94 for SMCC」の箱の中の「日本語 Solaris 2.4 - 11/94 ハードウェアリリースご使用にあたって（追補）-SMCC」、「日本語 Solaris 2.4 - 11/94 ハードウェアリリース SPARC ハードウェアプラットフォームガイド -SMCC」、「日本語 Solaris™ のインストール（SPARC 版）」、「日本語 Solaris 2.4 最新情報」を参照する。概略は次のとおりである。

① 日本語 Solaris 2.4 Hardware : 11/94 for SMCC CD-ROM メディアを準備する。他の 2 枚 (Hardware : 11/94 アップデイトソフトウェア及びシステム管理 AnswerBook) は今回はインストールしない。なお 1/4 テープドライブを SCSI ポートに接続しておく。

② CD-ROM メディアを CD-ROM ドライブにセットする。VME シャーシの電源投入後、本体の電源を入れ、L1 キーを押しながら a キーを押す。コンソールに ok プロンプトが出る。

③ ok boot cdrom [return] と入力する。

④ 自動的にインストールが開始される。各質問に以下のように答える。

ホスト名	cicu0
ネットワーク接続	Yes
IP アドレス	157.111.131.49
ネームサービス	None
サブネット	Yes
サブネットマスク	255.255.255.0
タイムゾーン	Asia Eastern, Japan → 時刻の設定

インストール方法	Initial
システムタイプ	Standalone
言語選択	Japanese を追加 ([Add >]) する。
ソフトウェア	Entire Distribution (525MB)
ディスク選択	c0t3d0 [ ] 両方追加 ([Add >]) する。 c0t1d0 [ ]
ディスク配置	Auto-layout

自動配置対象ファイルシステム：以下のすべてを選択する。

/, swap, /usr, /opt, /export, /usr/openwin, /var

ここで確認の画面が出るので Customize を選択する。

/export/home 20MB → 10MB ] に変更する。  
 / 15MB → 25MB ]  
 (これは後にパッチ 101945-34を入れるのに必要となる。)

インストール開始、終了後に Rebootしない を選択する。(パッチを入れるために) 約 1 時間ほどかかる。

⑤ パッチをインストールする。CD-ROM はそのままである。

```
# cd /cdrom/Patches return
# ./install_patches /a return パッチ (日本語 Solaris 2.4 最新情報参照)
      約 30 分ほどかかる。
# halt return
ok boot -s return
```

⑥ CD-ROM メディアを排出する。

```
# eject cdrom return
```

⑦ /export/home0 を手動でマウントする。

```
# mount /dev/dsk/c0t1d0s7 /export/home0 return
デバイス名が不安な場合は grep /export/home0 /etc/vfstab return で確認する。
```

⑧ パッチ 101945-38 をインストールする。1/4 テープメディア「T101945-38 Sun OS 5.4 patch」を準備する。このメディアを 1/4 テープドライブにセットする。

⑨ /export/home0 の下にパッチを展開する。

```
# cd /export/home0 return
# tar -xf /dev/rmt/0 return
```

⑩ パッチのインストールスクリプトを起動する。

```
# cd T101945-38 return
# ./installpatch . return
      15 分ほどかかる。
```

⑪ パッチファイルを削除し、システムを再起動する。

```
# cd /export/home0 return
# rm -r T101945-38 return
# reboot return
ルートパスワードを設定する。
```

### 5.3 csd のインストール

SPARC station 5 に SFVME 及び CSD をインストールする。SFVME のインストールについては残念ながら正確なマニュアルがない。参考として「SFx10 Series User's Manual」がある。CSD については「KINETIC Model 2140-Z1A デバイスドライバ利用の手引き」を参照する。概略は次のとおりである。

- ① 「SFVME Solaris 2.x Pkgadd format Date 1/24/95」 フロッピーディスク、「K2140 デバイスドライバ for Sun OS 5.4」 フロッピーディスクを用意する。以下の作業はすべてスーパーユーザで行う。

- ② vold を落とす。

```
# ps -ef | grep vold [return]
root 318 1 68 16:03:17 ? 0:01 /usr/sbin/vold
# kill 318 [return] (ps コマンドで出力される pid 番号を入力)
```

- ③ フロッピーディスクドライブに SFVME のフロッピーディスクをセットし、下記のコマンドを入力する。

```
# pkgadd -d /dev/rfd0c [return]
```

終了したらフロッピーディスクを排出する。

```
# eject [return]
```

- ④ フロッピーディスクドライブに CSD のフロッピーディスクをセットし、下記のコマンドを入力する。

```
# cd /opt [return]
# mkdir csd [return]
# cd csd [return]
# tar -xvf /dev/rfd0c [return]
# eject [return]
```

- ⑤ CSD をデバイスドライバがあるディレクトリへコピーし、実行権を与える。

```
# cp csd /usr/kernel/driv [return]
# cp csd.conf /usr/kernel/driv [return]
# chmod a+x /usr/kernel/driv/csd [return]
```

- ⑥ /etc/devlink.tab を修正した後 CSD ドライバを有効とする。

```
# vi /etc/devlink.tab [return]
```

最終行に type=ddi\_pseudo ; name=csd \_\_\_\_\_ \M0 を追加  
 ↑ (ここは TAB)

```
# add_drv -m '* 0666 root sys' csd [return]
```

⑦ /dev/csd0 ができているか確認する。もし異なるデバイス（例えば /dev/csd2）ができた場合は csd0 に変更する。

例) /dev/csd2 があり、/dev/csd0 がない時

```
# mv /dev/csd2 /dev/csd0 return
```

#### 5.4 C コンパイラのインストール

SPARC station 5 に C コンパイラをインストールする。応用ソフトウェアが完成し、ある程度の運用実績ができればコンパイラをここにインストールする必要はなくなる。しかしトラブルは突然起こるので、コンパイラを入れておいた方が安心である。「SPARCworks Professional C」の箱の中の「SunSoft 開発支援ツールのインストール」を参照する。概略は以下のとおりである。

① 「SunSoft WorkShop DEVELOPER PRODUCTS 日本語版 Volume 3, Number 2JA」 CD-ROM メディアを準備する。

② CD-ROM メディアを CD-ROM ドライブにセットし、次のコマンドを入力する。

```
# csh return
# setenv LANG ja return           -- 日本語環境
# /usr/openwin/bin/openwin return -- オープンウインドウの起動
# mount -o ro /dev/sr0 /cdrom return -- CD-ROM 手動マウント
# cd /cdrom return
# ./spro_install_tool return      -- 本体のインストール
ここで [製品] より SPARCworks Prof C3.0.1 [ ] を選択してインストールする。
ライセンス用ソフトウェアに  マーク
```

③ パッチをインストールする。CD-ROM メディアはそのままである。

```
# ./spro_install_patch return     -- パッチのインストール
パッチ対象を聞いてくるので cc_sw [return] と入力する。
```

④ ライセンスを設定する。CICU0 用にすでに取得してあるので、それを使用する。マシンを交換する場合（hostid が変わる場合）はライセンスを再取得する必要がある。

```
# /etc/opt/licenses/lit return
ライセンスインストレーションツール画面が表示される。
[製品の選択] ボタンで製品を順次選択しパスワード、使用権数、及び有効期限を設定し [ライセンス入力終了] をクリックする。
```

製 品	使 用 権 数	パスワード	チ ェ ッ ク サ ム
SPARCworks 3.0	1	07B0AD0D189DD8ACF9945	cb
SPARCworks Common	1	01B8A70D1F7308C74639F	84
SPARCompiler C3.0.1	1	0FB6A70E13F478DF2C748	61

有効期限は 01-jan-0 とします。

3回入力したのち [インストールを行って終了] をクリックして終了する。

- ⑤ CD-ROM メディアを排出する。

# eject cdrom [return]

## 5.5 応用プログラムのインストール

### 5.5.1 /etc ファイルの修正

ファイルの修正はスーパーユーザで行う。

/etc/resolv.conf ← ファイルを新しく作る。

```
domain naka.jaeri.go.jp
nameserver 157.111.133.11
nameserver 157.111.169.12
```

/etc/nsswitch.conf

hosts: files dns ... の部分を追加する。

/etc/inet/services

```
cicuhsd 911/tcp #(01) Acq request
cicuacmi 912/tcp #(02) ACMA request (ISP ini)
ciculac 913/tcp #(03) Local access request
cicuini 914/tcp #(06) Initialize request
cicuonl 915/tcp #(08) Online request
cicurac 916/tcp #(09) Remote access request
cicuacma 917/tcp #(12) ACMA request (ACMA ini)
cicuirt 918/tcp #(15) IRT request
```

の 8 行を追加する。なお # 以降はコメントなので任意である。

/etc/init/initd.conf

```
cicuhsd stream tcp nowait root /export/home0/cicu/app/sbin/hsd hsd
cicuacmi stream tcp nowait root /export/home0/cicu/app/sbin/pgt pgt
ciculac stream tcp nowait root /export/home0/cicu/app/sbin/lac lac
cicuini stream tcp nowait root /export/home0/cicu/app/sbin/ini ini
cicuonl stream tcp nowait root /export/home0/cicu/app/sbin/onl onl
cicurac stream tcp nowait root /export/home0/cicu/app/sbin/rac rac
cicuacma stream tcp nowait root /export/home0/cicu/app/sbin/pgt pgt
cicuirt stream tcp nowait root /export/home0/cicu/app/sbin/irt irt
```

の 8 行を追加する。

**/etc/system**

共有メモリの制限をとりのぞくため  
set shmsys : shminfo\_shmseg = 10  
を追加する。

**/etc/rc2.d/S97svc ← ファイルを新しく作る。**

CICU の応用システムを自動起動するためのスクリプトを作成する。

```
# !/bin/sh
echo 'starting CICU supervisor task:c'
if [ -x /export/home0/cicu/appl/sbin/svc ]
then
  /export/home0/cicu/appl/sbin/svc & echo 'svc\c' > /dev/console
fi
echo '!' > /dev/console
(S90loc.ja.cssd を参考にして作成する。)
```

**/etc/hosts**

他の cicu 名を登録しておく。

```
157.111.131.67 cicu1e
157.111.131.68 cicu2e
157.111.131.69 cicu3e
```

を追加する。

**5.5.2 ユーザ cicu の設定**

① スーパーユーザでログイン、OpenWindows を起動後、ユーザを設定する。

# admintool **return**

- User Account Manager を選択する。
- ネームサービス無を選択する。

Username	→ cicu
UserID	→ 102
Primary Group	→ staff
Login Shell	→ C(/bin/csh)
Create Home Directory	→ Yes
Path	→ /export/home0/cicu
Server	→ cicu0

これらを記入して  
**ADD** をクリックする。

② Open Windows を終了し、ログアウトする。次に cicu でログインし、パスワードを設定する。

```
# logout return
  console login : cicu return
  New Password : xxxxxxxx return
  Verification : xxxxxxxx return
```

③ ユーザ環境を設定しておく。

```
cicu0% cp /etc/skel/local.login ~/.login return
cicu0% vi ~/.login return
  stty erase ^H ] 2行追加する。
  setenv LANG ja
cicu0% cp /etc/skel/local.cshrc ~/.cshrc return
cicu0% vi ~/.cshrc return
  set path=(/opt/SUNWspro/bin /usr/ccs/bin $path ~/appl/bin)
  setenv MANPATH /opt/SUNWspro/man:/usr/man
  setenv LD_LIBRARY_PATH /opt/SUNWspro/lib:/usr/lib
の3行を追加する。
```

### 5.5.3 CICU ソフトウェアのインストール

CICU ソフトウェアのインストールはユーザ cicu で行う。「CICU ソフトウェア」フロッピーディスクをフロッピーディスクドライブにセットする。

```
cicu0% cd return
cicu0% tar -xvf /dev/rfd0c return
cicu0% eject return
cicu0% cd appl return
cicu0% make install return
```

これですべての作業が終了した。システムを停止し、再立ち上げします。ハードウェアを確実にクリアするため電源を1度切り、30秒後に立ち上げる。

```
cicu0% su return
password: パスワード return
# halt return
:
ok
```

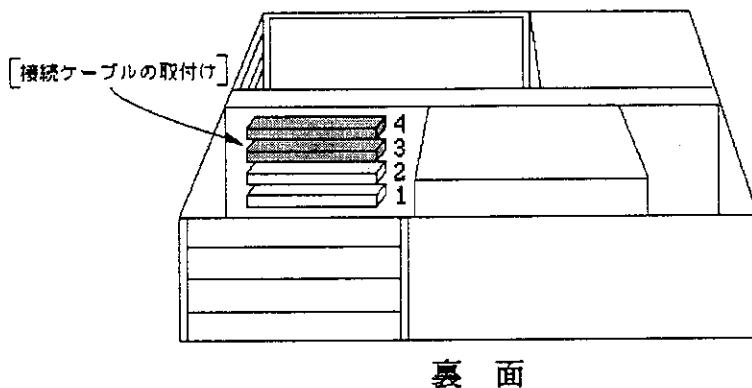
電源切 → 30秒後に VMEbus シャーシの電源投入後、本体の電源を入れる。

## 6. CICU 1, 2, 3 の構築

### 6.1 ハードウェアのインストール

#### 6.1.1 2140 シリアルハイウェイドライバの取り付け

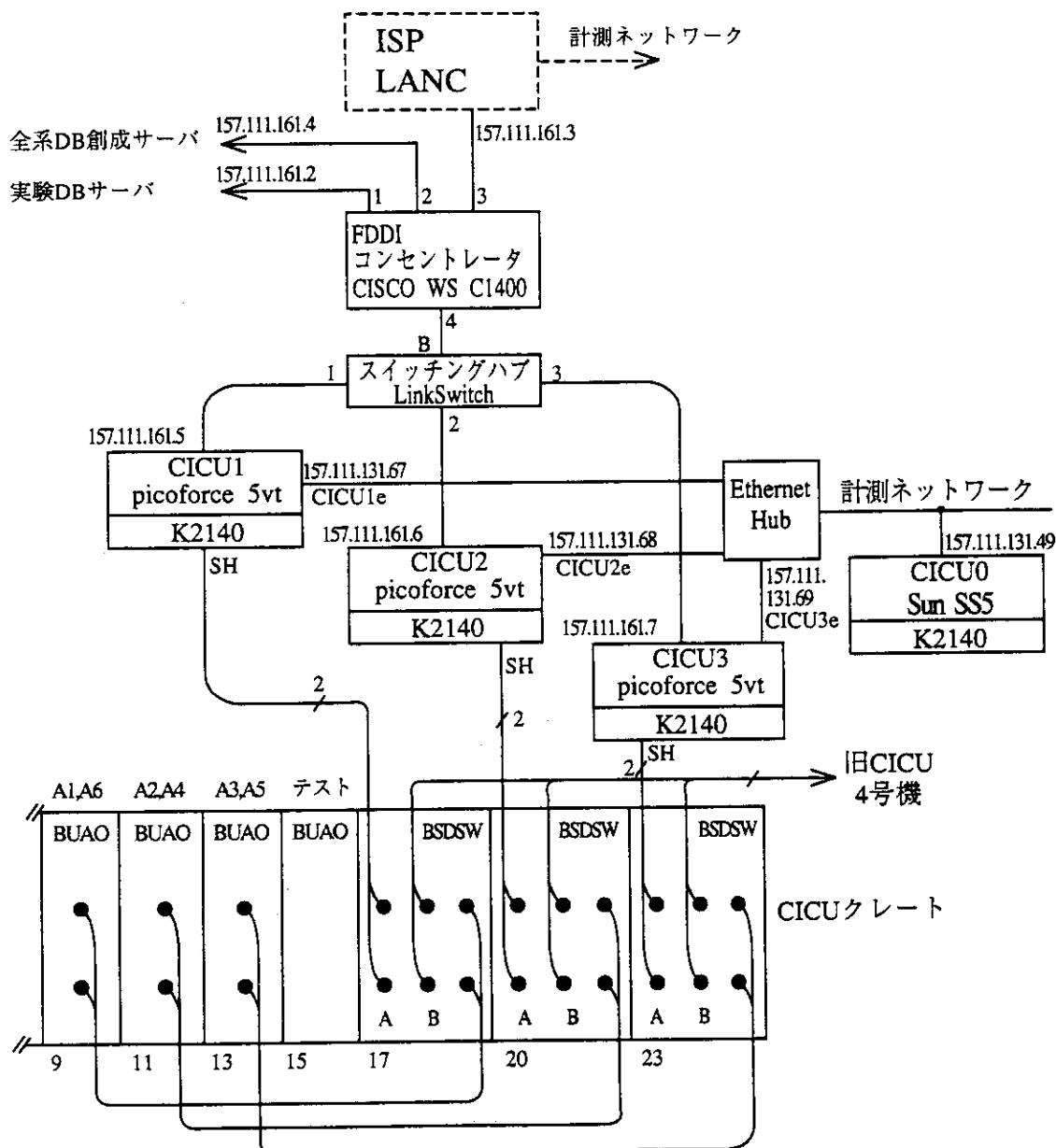
- ① picoFORCE の上面カバーを開け、2140 に付属の P2 ジャンパーケーブルを P2 のスロット 3 と 4 の間にバックプレーンの裏側から取り付ける。



- ② 2140 シリアルハイウェイドライバ送信ボードのジャンパーを確認する。ボード上の 4 本のネジをはずし「MSB」と書いてあるジャンパーが入っていることを確認する。また前面パネルの速度選択を 2.5 MHz に変更する。詳細については「Model 2140-Z1A Enhanced Serial Highway Driver for VMEbus Systems INSTRUCTION MANUAL p4 CHART I」を見る。
- ③ 2140 シリアルハイウェイドライバ受信ボードのジャンパーを確認する。すべて工場設定のままである。
- ④ 2140 シリアルハイウェイドライバ送信ボードをスロット 3 に、同じく受信ボードをスロット 4 に差し込む。

### 6.1.2 ネットワークとハイウェイの接続

ネットワーク及びシリアルハイウェイを下図のように接続する。



## 6.2 OS のインストール

cicu1, 2, 3 用には Solaris 2.5 が納入されているので多少手順は異なる。CD-ROM ドライブ、コンソール CRT を接続してからシステムを立ち上げる。5.2 を参考にしてインストールする。異なる点は以下のとおりである。

④ ホスト名	cicu1	cicu2	cicu3
IP アドレス	157.111.161.5	157.111.161.6	157.111.161.7
第 2 ホスト名	cicu1e	cicu2e	cicu3e
第 2 アドレス	157.111.131.67	157.111.131.68	157.111.131.69

ディスクは自動配置で、Customize の必要はない。

⑦～⑪パッチ T101945-38 はインストール不要である。

## 6.3 csd のインストール

- ・ SFVME の代わりに FORCE VME driver をインストールする。
- ・ csd のインストールは同じなので 5.3 を参照する。

## 6.4 C コンパイラのインストール

コンパイラはインストールしない。

## 6.5 応用プログラムのインストール

### 6.5.1 /etc ファイルの修正

- ・ 6.2 OS のインストール時に第 2 アドレスの問い合わせがなかった場合は /etc/hostname.le1 を作り、ここに第 2 ホスト名を入れる。また /etc/hosts にも第 2 アドレスと第 2 ホスト名を追加する。
- ・ /etc/notrouter ファイルを作成する。中身は空とする。
- ・ /etc/host.equiv ファイルを作成し、「cicu0」のみ入れる。また /etc/hosts ファイルに「157.111.131.49 cicu0」行を追加する。
- ・ 他は 5.5.1 を参照する。

### 6.5.2 ユーザ cicu の設定

- ① useradd コマンドを使ってユーザを設定する。

```
# useradd -d /export/home0/cicu -g staff -m -u 102 -s/bin/csh cicu [return]
# passwd cicu [return]
New password : xxxxxxxx [return]
Re-enter new password : xxxxxxxx [return]
```

② ログアウトし、cicu でログインする。

```
# logout [return]
console login :cicu [return]
password :xxxxxxxx [return]
```

③ ユーザ環境を設定する。

```
% vi .cshrc [return]
set path=($path ~/appl/bin)
stty erase ^H
```

の2行を入れる。

#### 6.5.3 CICU ソフトウェアのインストール

CICU ソフトウェアのインストールはユーザ cicu で行う。第2 Ethernet が有効ならば cicu0 から rcp コマンドでインストールができる。

cicu0% rcp -rp ~/appl cicu1:appl [return] (cicu1への場合)

第2 Ethernet がまだ無効な場合は1回リブートしてから実行する。

注意 CICU ソフトウェアのインストールディレクトリは、プログラム中 PROJROOTDIR として固定されている。再コンパイルするのでなければ必ず /export/home0/cicu/app1 にインストールする。

## 付録 K2140 CSR の読み方

Kinetic 2140 シリアルハイウェイドライバのステータスレジスタ (csr) のビット配置を示す。

																LSB
MSB																
ABT	STMO	NEX	0	STE	DLE ERR	TMO	NO SYNC	RPY ERR	0	LPE	TPE	ADNR	0	NO X	NO Q	

ABT : ADNR, TMO, STE, NOX 4つの OR

STMO : ソフトウェアタイムアウト (TMO より短い)

NEX : 他のエラー要因により、このコマンドは実行されなかった。

STE : 伝送エラー。RPYERR, LPE, TPE 3つの OR

DLEERR : 直前のコマンド実行時にエラーが発生していた。

TMO : ハードウェアタイムアウト (15秒)

NOSYNC : 同期信号なし

RPYERR : CCL2 よりエラービットが示された。

LPE : 垂直パリティエラー

TPE : 水平パリティエラー

ADNR : クレート番号エラー

NOX : X=0 である。

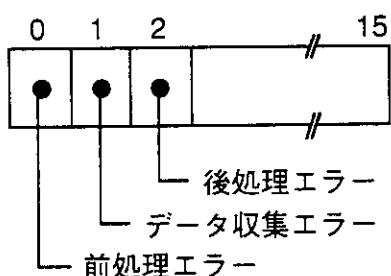
NOQ : Q=0 である。

高速データ収集では、従来の BSD-B の LER に似せるようにするため、ビットの一部を動かしている。“LER”として表示されるビットの配置を示す。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DLE ERR	NOQ	NOX	STE	STMO	NEX	0	STE	DLE ERR	TMO	NO SYNC	RPY ERR	0	LPE	TPE	ADNR

(反転)(反転)

また DET (詳細情報) は以下のようになる。



## 第2編 新CICUの基本設計

## 1. 概 要

A7 データ処理設備 CAMAC インターフェース制御装置（以下 CICU と呼ぶ）は JT-60 計測装置の CAMAC クレートと A7 データ処理設備ショット間処理計算機（以下 ISP と呼ぶ）を接続するもので、JT-60 計測装置のマイクロコンピュータ付補助クレートコントローラ（以下 ACMA と呼ぶ）と ISP との通信を仲介して JT-60 計測装置を JT-60 の放電シーケンスに同期させると共に、放電終了時に ISP の指示に基づいて ACMA を介さずに計測データの収集と、ISP への転送を行うものである。

## 2. 構成

CICU は中央処理装置（磁気ディスク、CRT、キーボードを含む）、VMEbus インターフェース装置、CAMAC シリアルドライバ、基本処理プログラム及び CICU ソフトウェアにより構成される。CICU の構成を図 1 に示す。

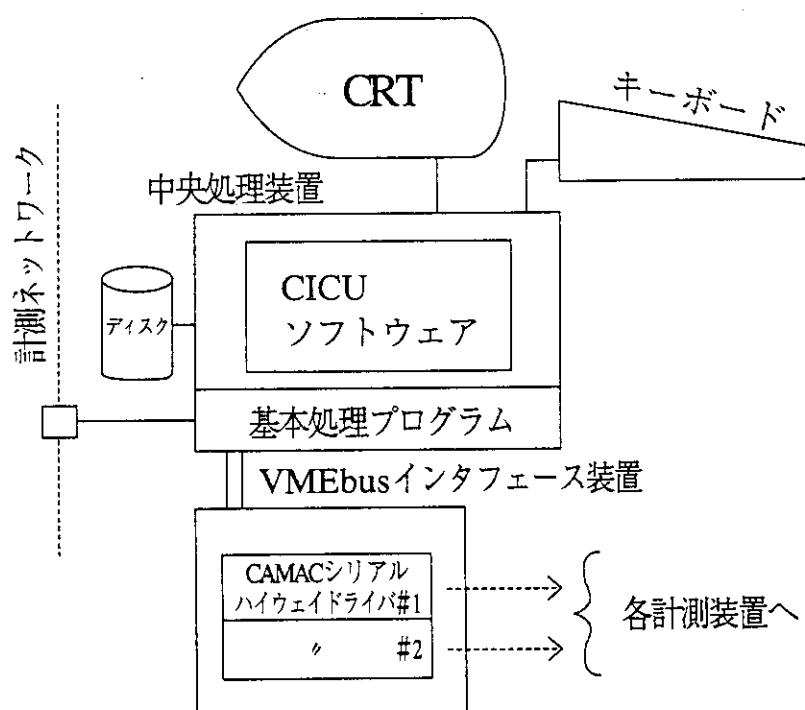


図 1 CICU の構成

CICU は計測ネットワークに接続され、このネットワークを使用して ISP 内の応用プログラムである ACMA 情報交換、高速データ収集プログラム、CICU 情報交換、模擬 ACMA 情報交換、及び ACMA コマンドと接続される。また CICU は CAMAC シリアルハイウェイにより JT-60 計測装置と接続されている。CICU と他装置との関係を図 2 に示す。

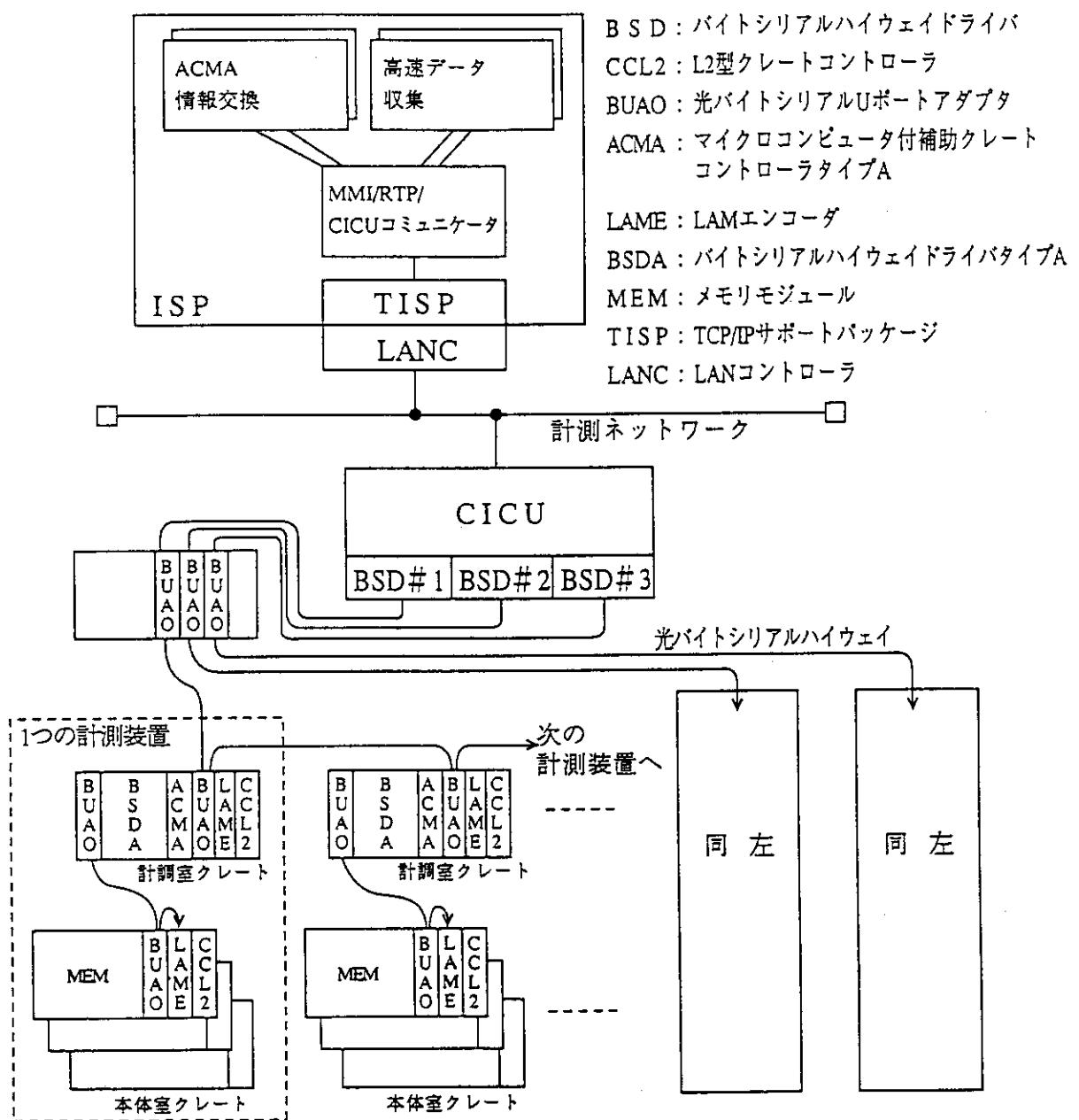


図 2 CICU と他装置との関係

### 3. 機能

CICU ソフトウェアは ACMA と ACMA 情報交換間の通信を仲介する ACMA 通信機能、本体室クレートより計測データを収集し ISP に転送する高速データ収集機能、本体室アクセス要求など ISP からの单一要求を処理する单一要求処理機能、旧 CICU との互換性確保のため存在するダミー処理機能、CICU ソフトウェア全体を監視するシステム監視機能、及びコンソールからのコマンドを処理するコンソール機能の 6 つの機能より構成される。各機能の関係を図 3 に示す。

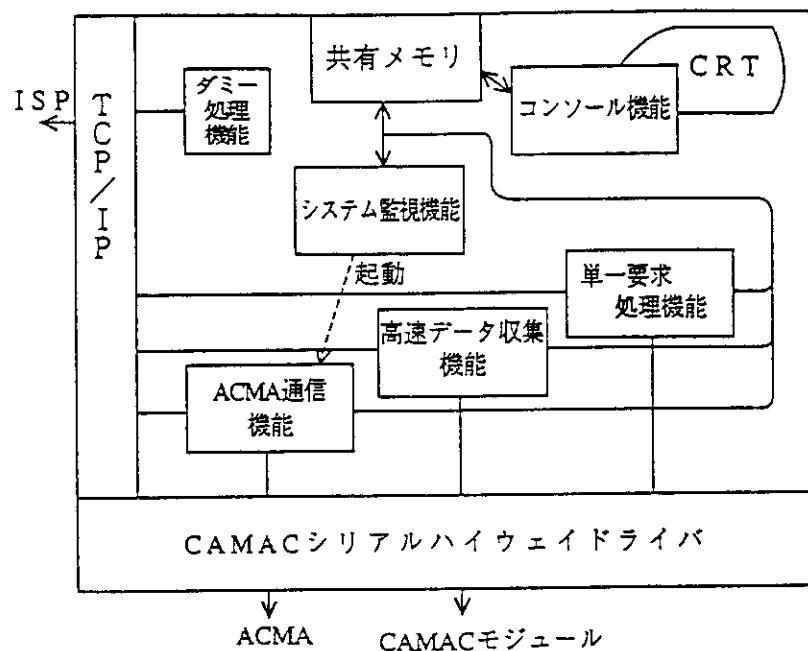


図 3 CICU ソフトウェア各機能の関係

#### 3.1 ACMA 通信機能

ACMA 通信機能は ACMA 通信処理、LAM 受信処理、RQPUT 中継処理、及び RQGET 中継処理の 4 つの処理より構成される。ACMA 通信機能の各処理はシリアルハイウェイに対して 1 組ずつ存在し、同様に各シリアルハイウェイに 1 つずつ割り当てられる ACMA 通信要求メールボックス（以下 ACMA 通信要求 MBX と呼ぶ）を中心に処理要求の待ち合わせ、及び処理結果の通知を行う。ACMA 通信処理及び LAM 受信処理はシステム監視機能統括処理より各シリアルハイウェイごとに起動される。RQPUT 中継処理及び RQGET 中継処理（以下、両者を区別しない場合は単に中継処理と呼ぶ）は ISP から接続要求があるごとに inetcより起動される。ACMA 通信機能の通信シーケンスの例を図 4 a ~ p に示す。

##### 3.1.1 ACMA 通信処理

ACMA 通信処理は ACMA 通信機能の中核を成すもので ISP 及び ACMA の通信シーケンスの管理を行う。ACMA 通信処理の状態遷移図を図 5 に、決定表を表 1 に示す。

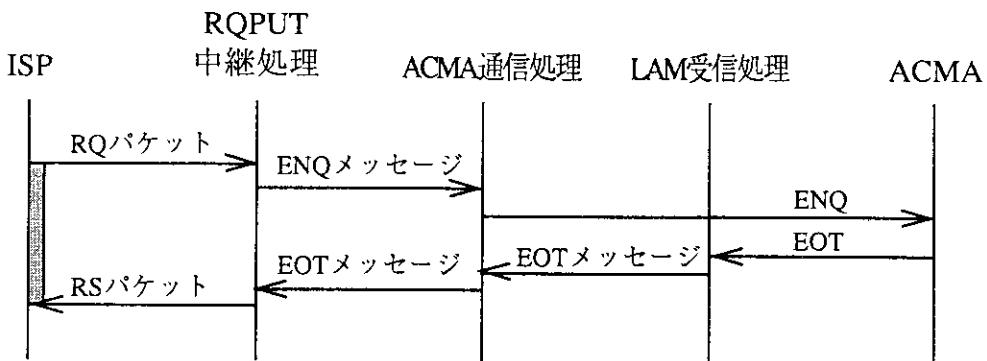


図4a 通信シーケンスの例 ISP起動データ転送なし

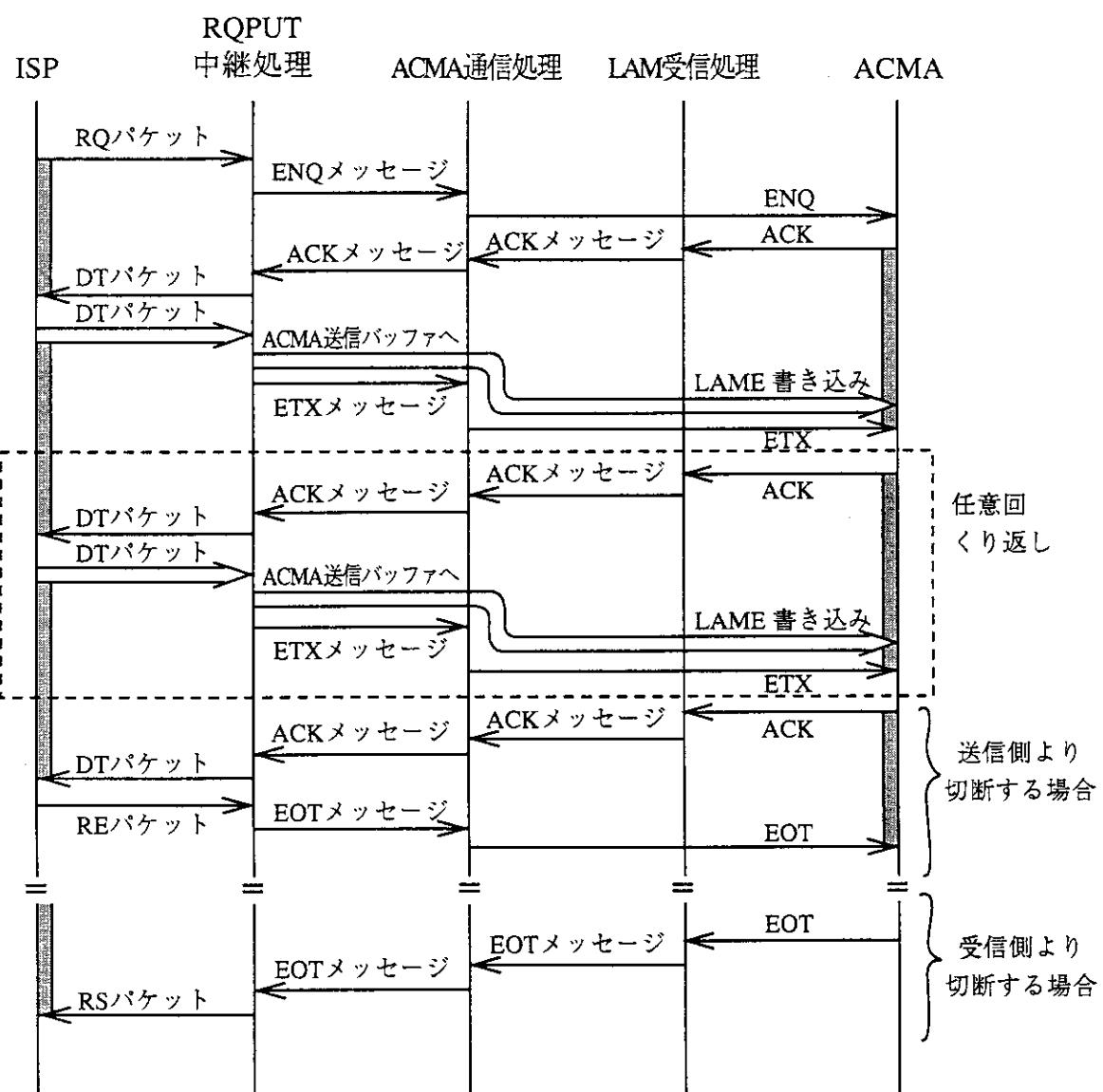


図4b 通信シーケンスの例 ISP起動 ISP→ACMAへデータ転送

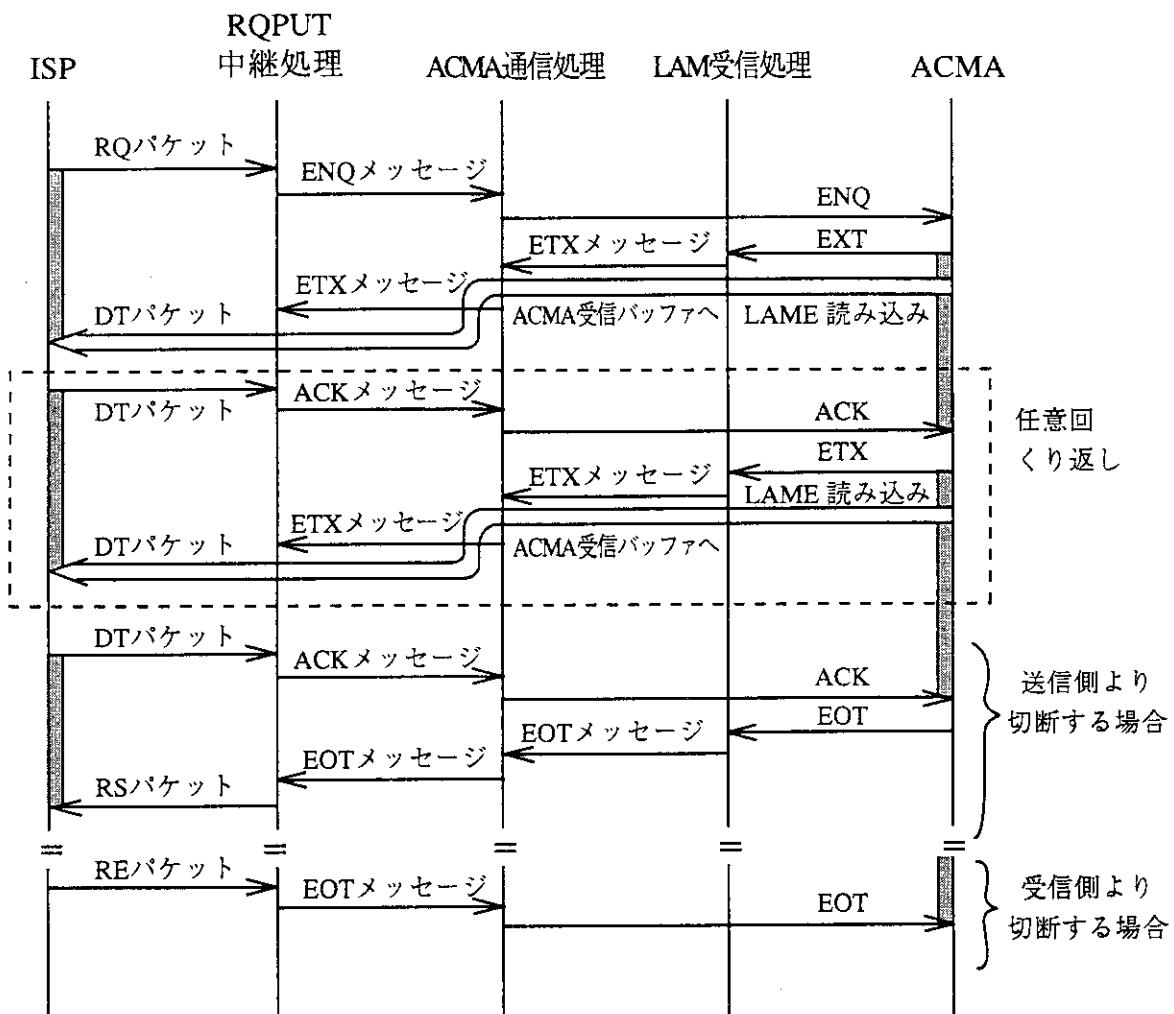


図4c 通信シーケンスの例 ISP起動 ACMA→ISPへデータ転送

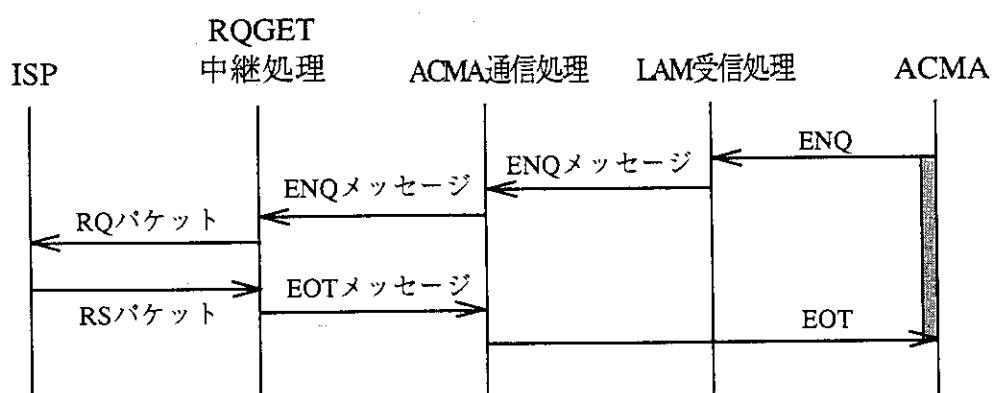


図4d 通信シーケンスの例 ACMA起動 データ転送なし

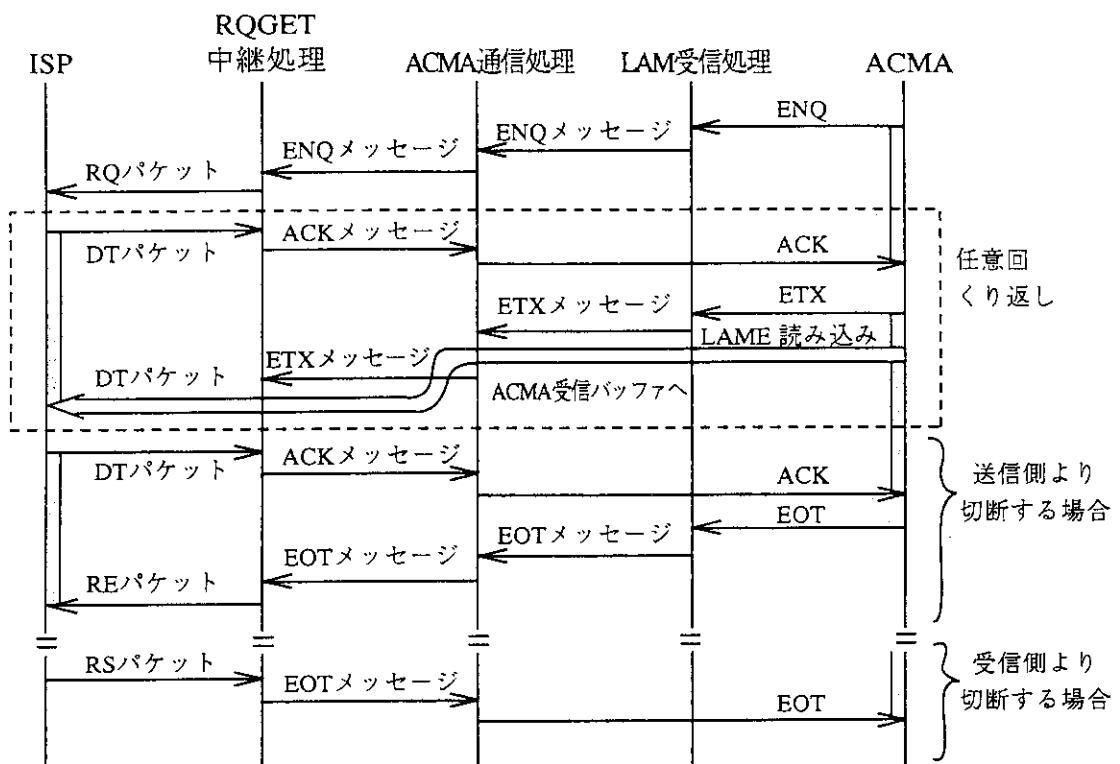


図4e 通信シーケンスの例 ACMA起動 ACMA→ISPへデータ転送

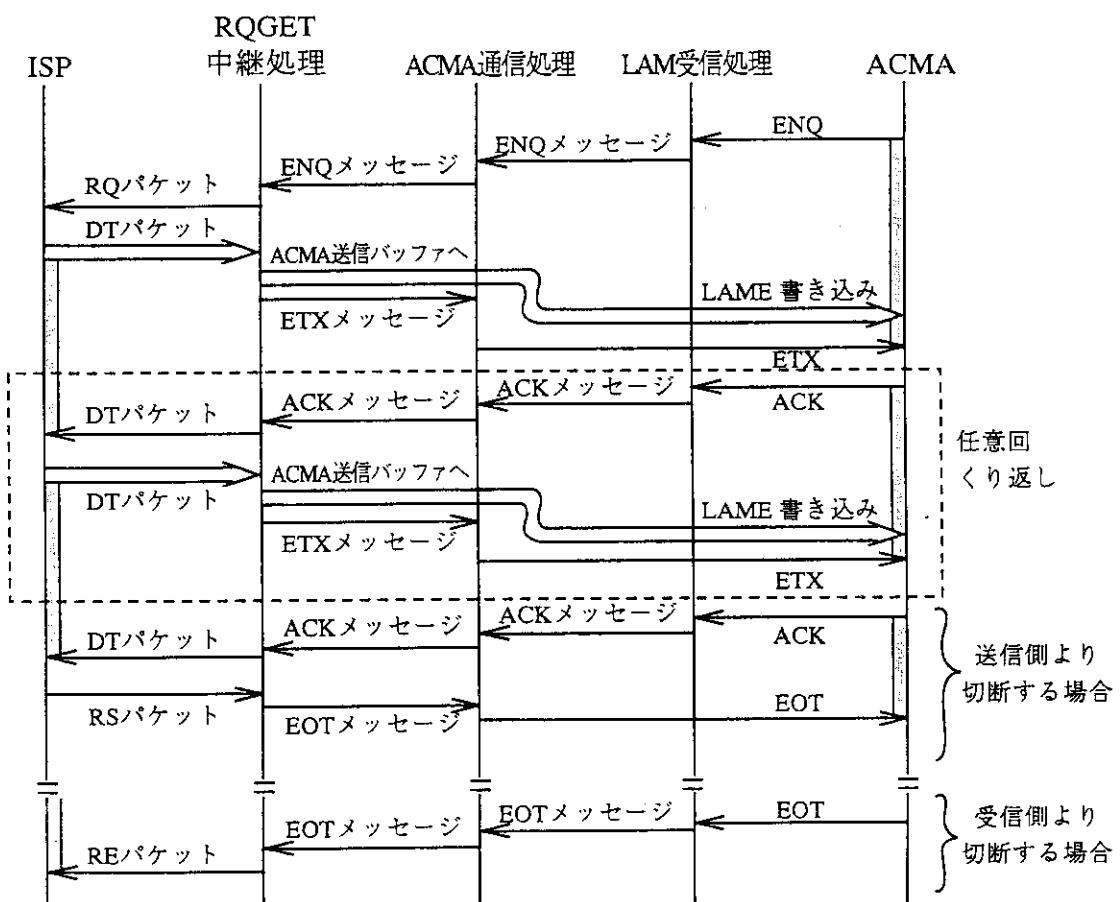


図4f 通信シーケンスの例 ACMA起動 ISP→ACMAへデータ転送

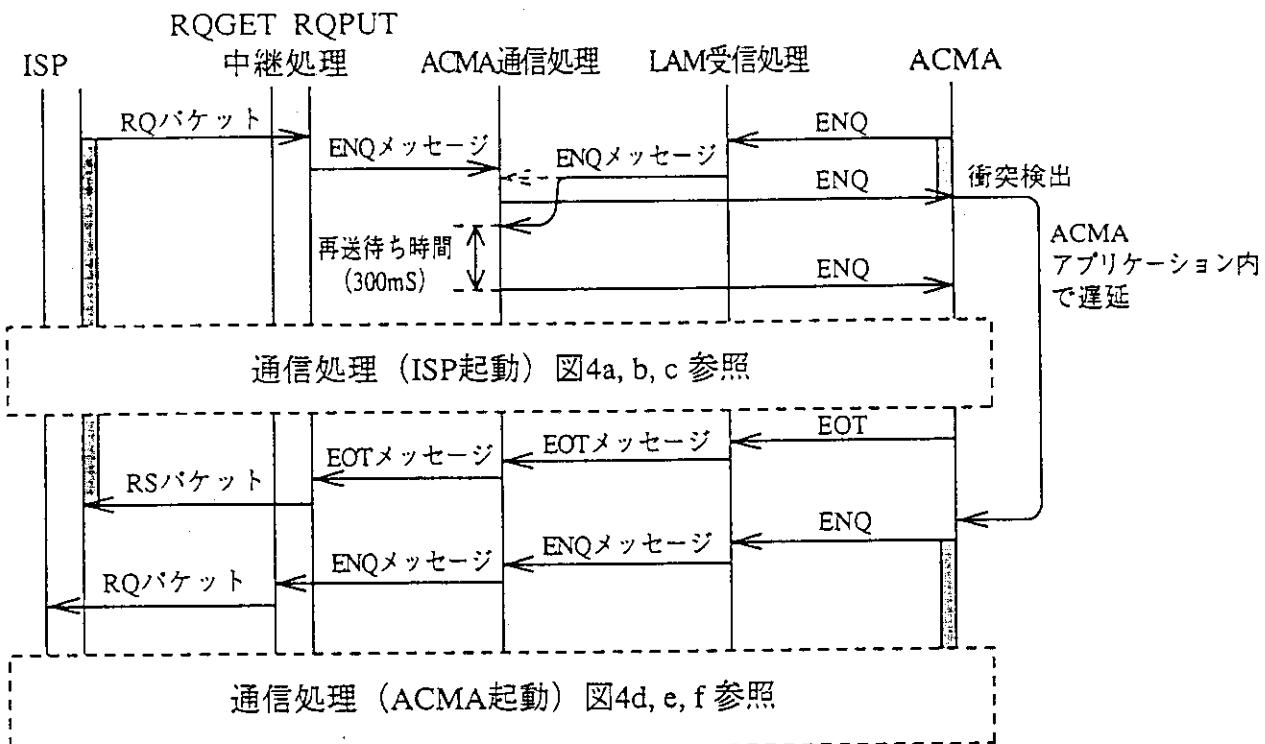


図 4 g 通信シーケンスの例 CICU 内 ENQ 衝突処理；再送

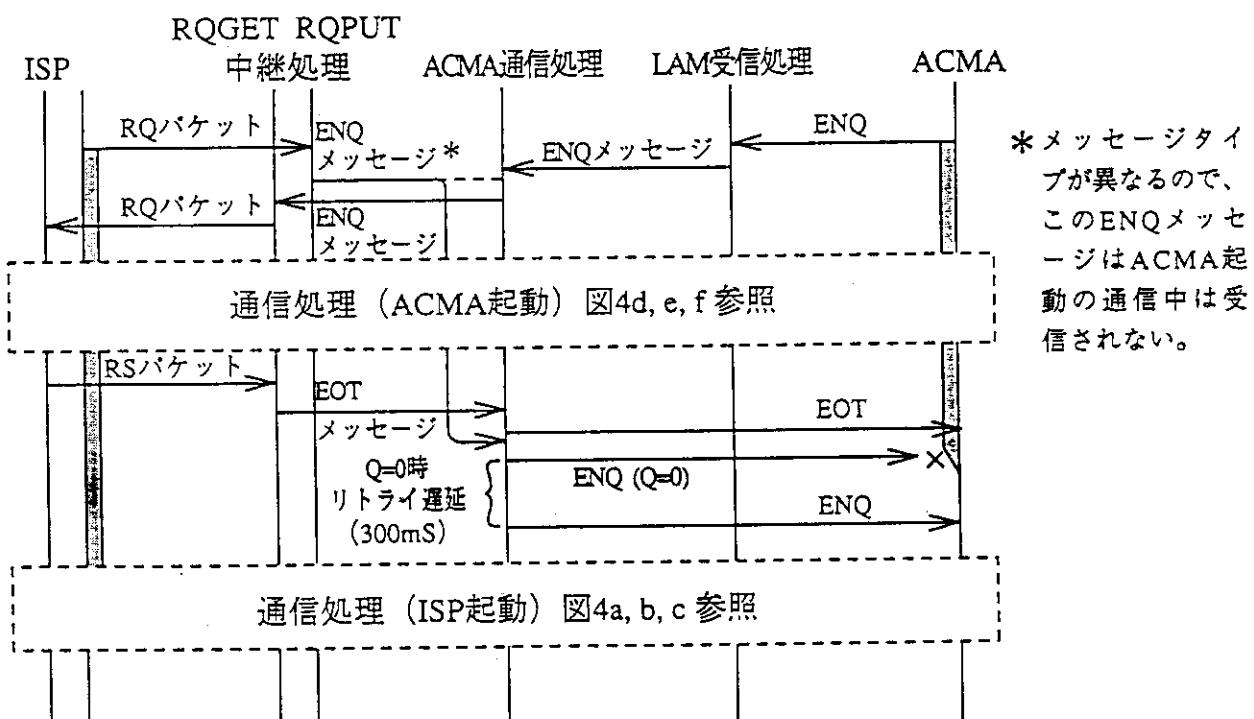


図 4 h 通信シーケンスの例 CICU 内 ENQ 衝突処理；回避

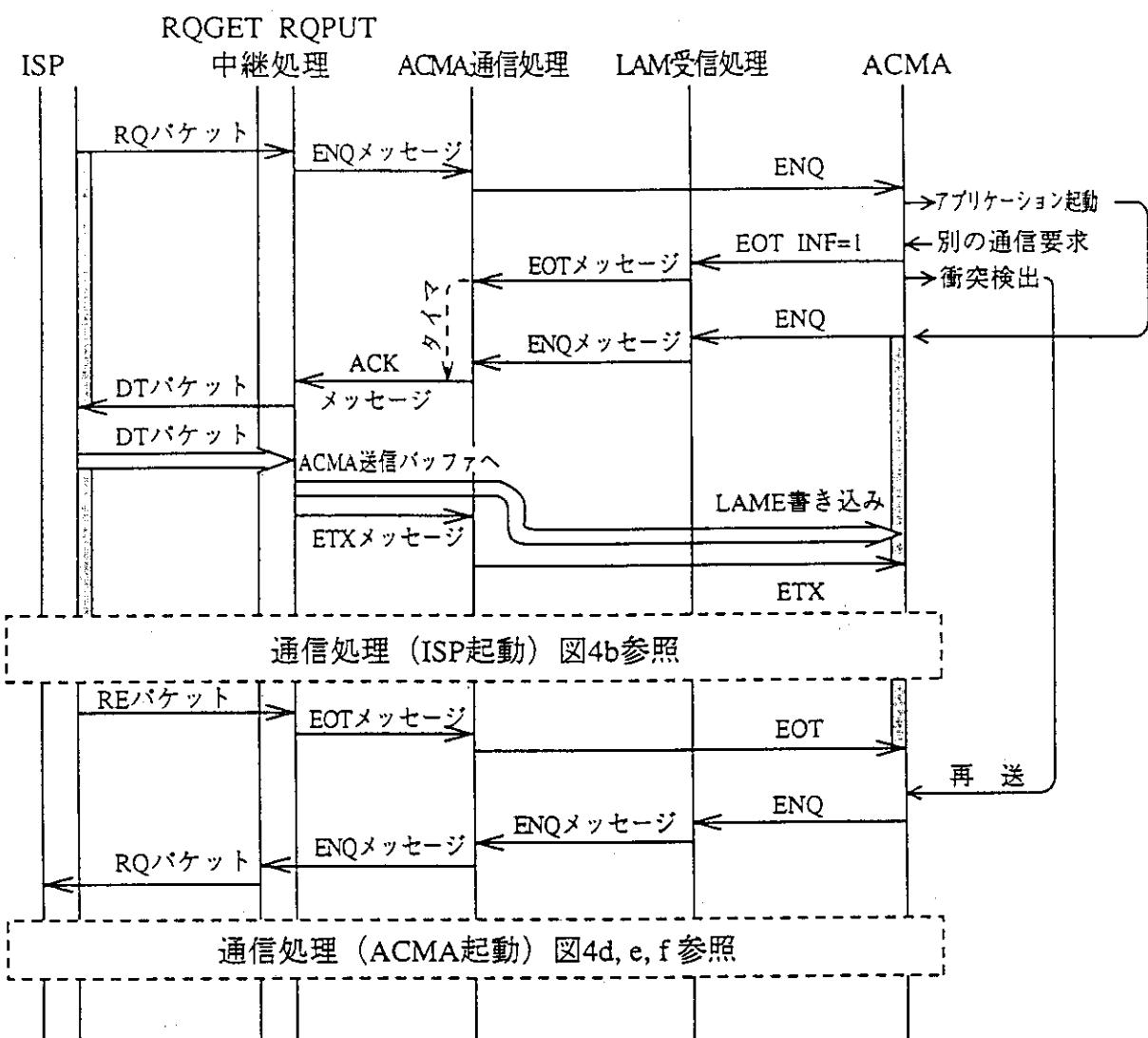


図4-i 通信シーケンスの例 ACMA内ENQ衝突処理  
データ転送方向 ISP → ACMA

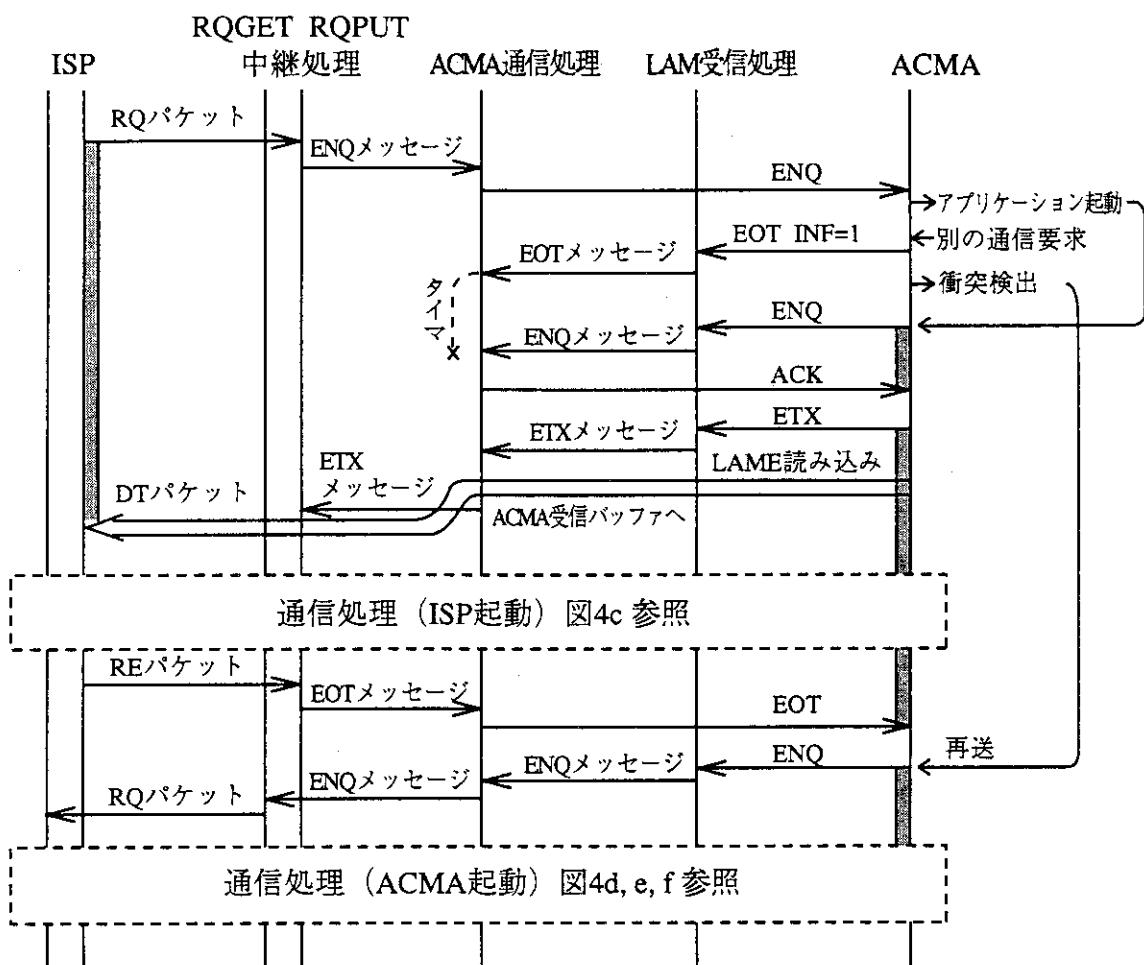


図 4 j 通信シーケンスの例 ACMA 内 ENQ 衝突処理  
データ転送方向 ACMA → ISP

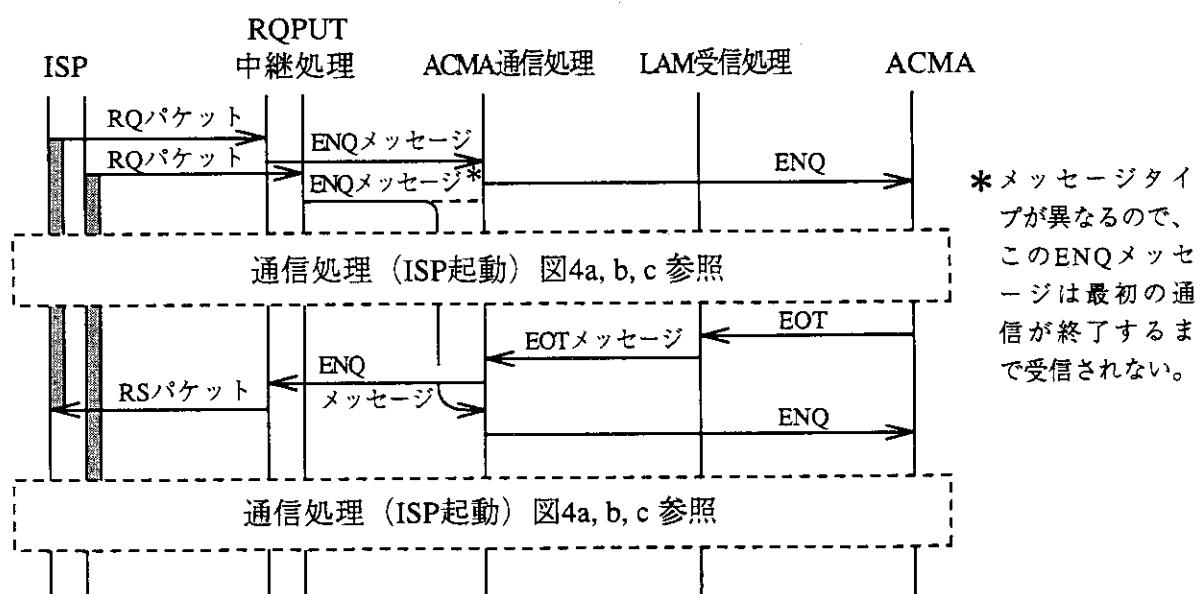


図 4 k 通信シーケンスの例 同時通信要求 (ISP)

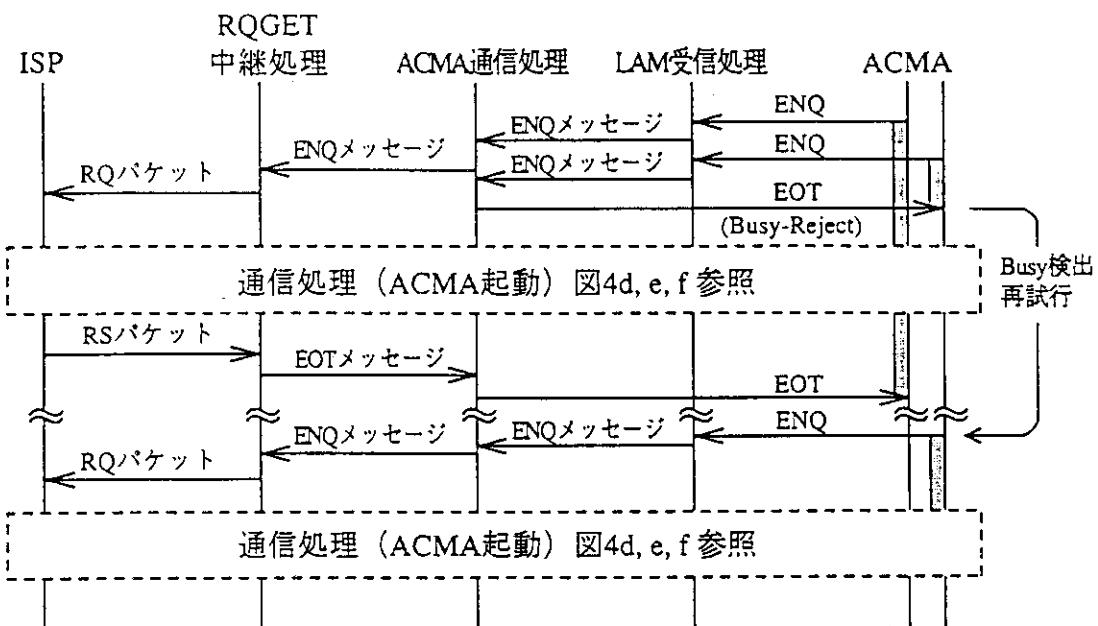


図4 l 通信シーケンスの例 同時通信要求 (ACMA)

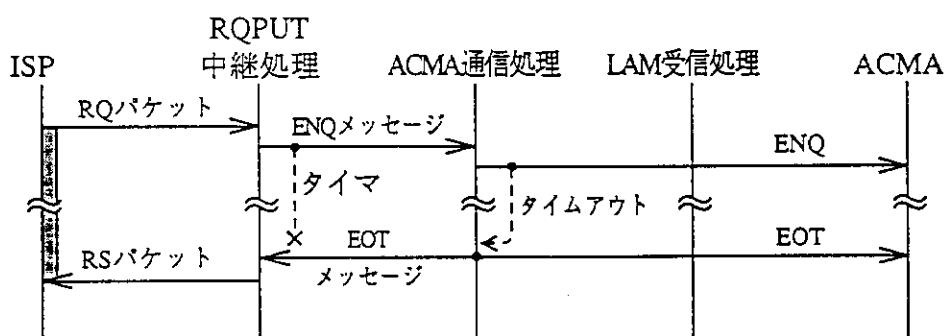


図4 m 通信シーケンスの例 ACMA 応答待タイムアウト

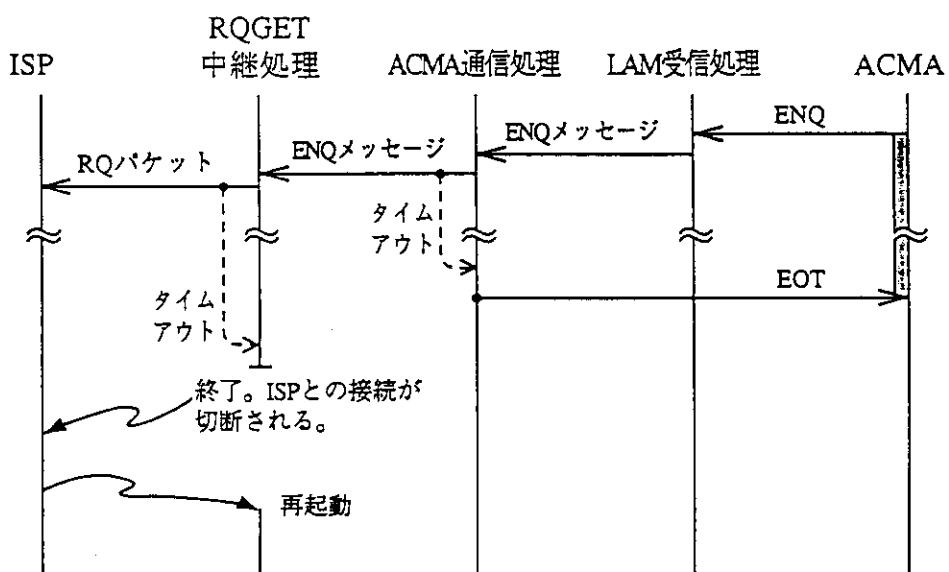


図4 n 通信シーケンスの例 ISP 応答待タイムアウト

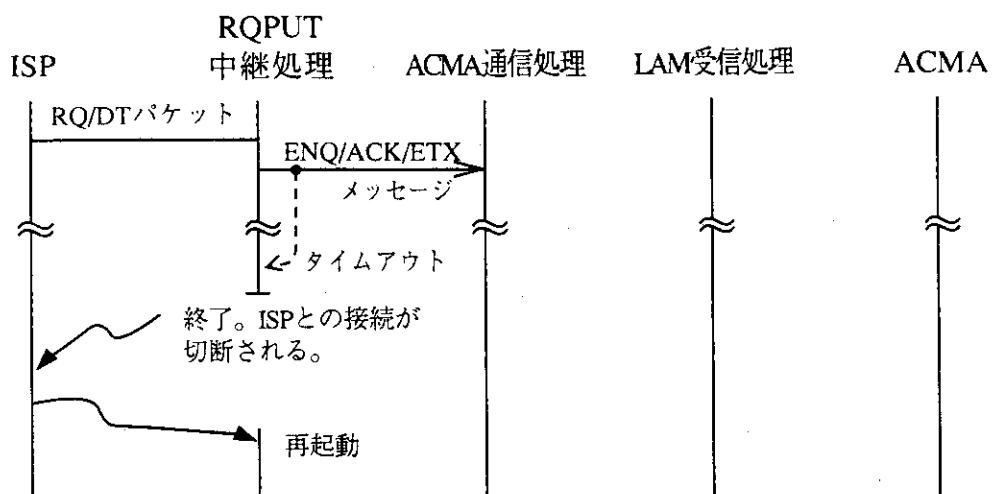


図 4 o 通信シーケンスの例 ACMA 通信処理タイムアウト

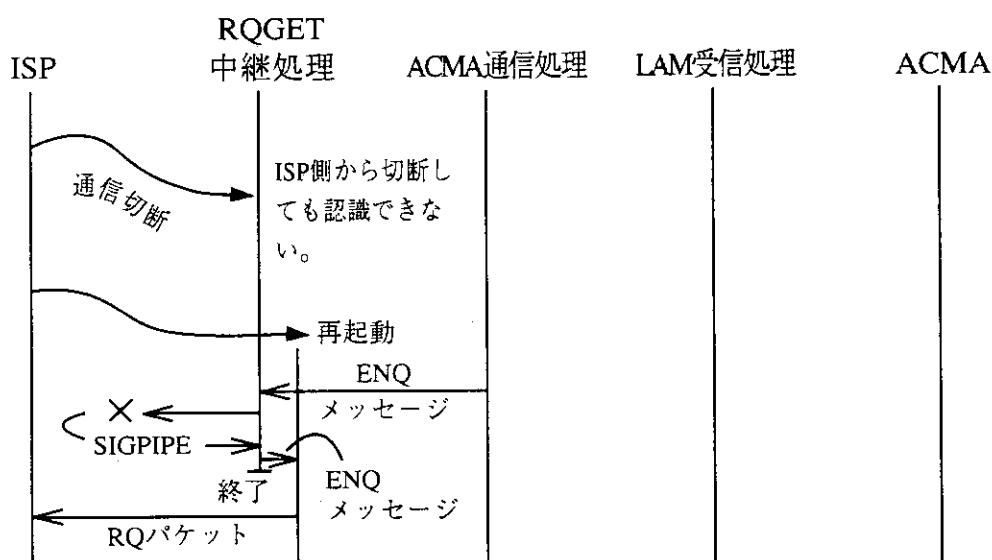
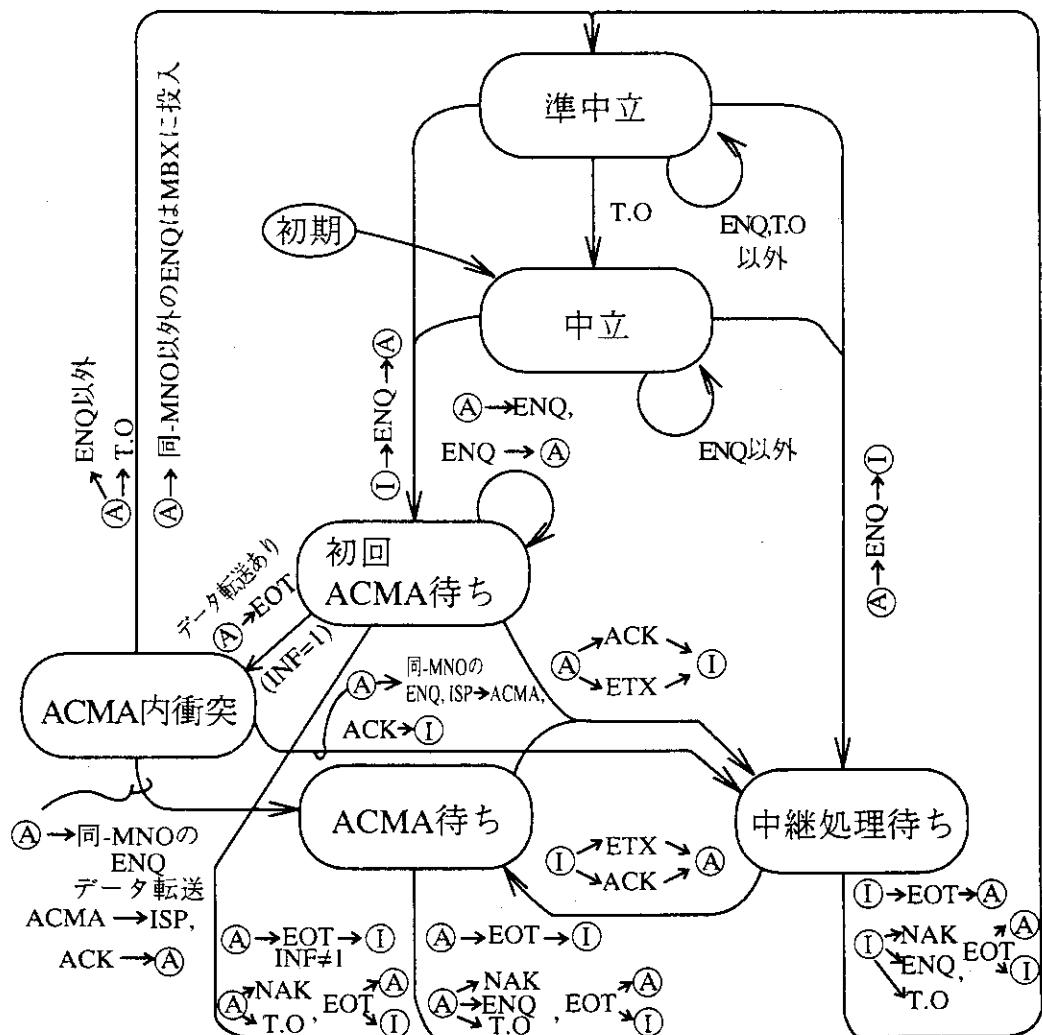


図 4 p 通信シーケンスの例 RQGET 中継処理の再起動



凡例 ⑧ → EOT → ① ACMA より EOT を受信した場合は EOT を中継処理へ通知する。

①→T.O, EOT→Ⓐ 中継処理待ちタイムアウトの場合はEOTをACMAへ送信する。

図5 ACMA通信処理状態遷移図

表1 ACMA通信処理決定表

状態 受信	中立	準中立	データ転送なし		ISP→ACMA		データ転送方向		ISP→ISP		データ転送方向	
			初回 ACMA待ち	中継処理待ち	初回 ACMA待ち	ACMA待ち	中継処理待ち	ACMA待ち	中継処理待ち	ACMA待ち	ACMA→ISP	ISP→ACMA
受信メッセージタイプ	全て	64以下 ENQメッセージのみ受信	62以下。通信中の(ACMA内衝突時に衝突が発生した)ACMA以外からのメッセージに対するはただちにEOT(Busy-Reject)を返し、状態も変化しない。								ACMA内衝突	
中継処理よりENQ	ENQをACMAへ送信 →初回ACMA待ち		(メッセージタイプ=64のため受信しない)									
ACMAよりENQ	ENQをRQGET中継処理へ通知 →中継処理待ち		300ms後にENQをACMAに再送	エラー(下記参照)	300ms後にENQをACMAに再送	エラー(下記参照)	300ms後にENQをACMAに再送	エラー(下記参照)	エラー(下記参照)	エラー(下記参照)	同一MNOならACKを中継ACMAに通知	同一MNOならACKを中継ACMAに通知
ACK					ACKを中継処理へ送信 →中継処理待ち		ACKを中継処理へ送信 →中継処理待ち		ACKをACMAへ送信 →ACMA待ち		→ACMA待ち	
ETX	無視する		(下記参照)		エラー(下記参照)		LAMEデータ書き込み。 ETXを中継処理へ送信 →ACMA待ち		エラー(下記参照)		→準中立	
EOT					終了情報をEOTより設 定。EOTを中継処理に通知 →準中立		INF=1なら →ACMA内衝突 INF=0なら右欄と同一処理 →準中立		終了情報をINF=1なら EOTより設定。EOTを中継処理に通知 →準中立		終了情報をINF=0なら EOTより設定。EOTを中継処理に通知 →準中立	
NAK												
T.O	—		→中立									
エラー… { ACMA及び中継処理にEOT(エラー)を通知 ., →準中立												

- ① 共有メモリを割り当て、起動メッセージをロギングする。起動パラメタからループ番号を読み込み、ループ番号に対応するシリアルドライバ及び ACMA 通信要求 MBX を決定する。シリアルドライバをオープンする。ACMA 通信要求 MBX をクリアし、ACMA 通信情報の初期化（通信ステータスを中立に、中継処理 pid、ACMA クレートアドレス、及び MNO を 0 に、終了情報を正常に、データ転送方向をデータ無とする）を行う。SIGALARM シグナルでシステムコールを再始動しないよう指定する。正常に終了した場合は②へ、エラーの場合はロギングし、処理を終了する。
- ② 通信ステータスに応じた時間監視を開始する。③へ行く。（表 2 参照）  
(ualarm (3))
- ③ 通信ステータスに応じたメッセージタイプの要求のみを ACMA 通信要求 MBX で待ち合わせる。（表 2 参照）タイムアウトの場合は④へ行く。メッセージを受信した場合は必要ならば時間監視を解除し、残り時間を退避する。通信ロギングを行う。通信ステータスが中立、又は準中立の場合は⑤へ行く。メッセージの送信元が（通信中の）ACMA クレートアドレスと異なる場合は EOT (Busy-Reject) を送信元に送信する。エラーの場合はエラーロギングを行う。正常ならば通信ロギングを行う。いずれの場合も退避した残り時間を調べ 1 秒以上の場合はその時間で時間監視を開始し③へ行く。1 秒未満の場合は④へ行く。メッセージの送信元が ACMA クレートアドレスと同一の場合は、通信ステータスが ACMA 内衝突である場合は⑥へ、それ以外の場合は⑦へ行く。
- ④ [タイムアウト] 通信ステータスにより以下のように処理する。
- ・ 準中立 → 通信ステータスを中立とし②へ行く。
  - ・ ACMA 内衝突 → 通信ステータスを準中立とし②へ行く。
  - ・ その他 → 終了情報にエラーを設定し⑪へ行く。

表 2 通信ステータスと時間監視、受信メッセージタイプ

通信ステータス	時間監視	受信メッセージタイプ
中立	行わない	全部
準中立	30秒	64 以下
(初回) ACMA 待ち	ACMA 待ち T.O 時間	62 以下
中継処理待ち	中継処理待ち T.O 時間	同上
ACMA 内衝突	3秒	同上

⑤ [中立、準中立] ENQ メッセージ以外はすべて無視し、②へ行く。ENQ メッセージの場合は MNO、データ転送方向を保存する。中継処理より受信した場合は要求元の pid を中継処理 pid に設定し、通信相手先の ACMA に ENQ を送信する。Q=0 の場合は 300mS 待ち、再度送信する。再度 Q=0 となった場合、又はその他のエラーの場合はエラーをロギングし、終了情報をエラーとして⑭へ行く。正常ならば通信ロギングを行い、ACMA クレートアドレスを設定し、通信ステータスを初回 ACMA 待ちとする。②へ行く。LAM 受信処理より受信した場合は ACMA クレートアドレスを設定し、RQGET 中継処理にメッセージタイプ 65 で ENQ メッセージを通知する。通信ロギングを行い、通信ステータスを中継処理待ちとする。②へ行く。

⑥ [ACMA 内衝突] ENQ メッセージ以外はすべて読み捨て、通信ステータスを準中立とし②へ行く。保存してある MNO と異なる MNO を持つ ENQ メッセージを受信した場合は ACMA 通信要求 MBX に ACMA クレートアドレスのメッセージタイプで同一メッセージを送信し、通信ステータスを準中立として②へ行く。保存してある MNO と同一の MNO を持つ ENQ メッセージを受信した場合は以下の処理を行う。データ転送方向が ISP → ACMA の場合は ACK メッセージを中継処理に通知し、通信ロギングを行って、通信ステータスを中継処理待ちとする。②へ行く。データ転送方向が ACMA → ISP の場合は ACK を ACMA に送信する。エラーの場合はロギングし、終了情報をエラーとして⑭へ行く。正常ならば通信ロギングを行い、通信ステータスを ACMA 待ちとする。②へ行く。

⑦ [（初回）ACMA 待ち、中継処理待ち] 通信ステータス、データ転送方向、受信メッセージの組み合わせで表 3 のように処理する。

表 3 （初回）ACMA 待ち、中継処理待ちにおける受信メッセージの処理

	通信ステータス	データ転送方向	受信メッセージ	送信元	処理
1	初回 ACMA 待ち	(任意)	ENQ	ACMA	→ ⑧
2	初回 ACMA 待ち又は ACMA 待ち	ISP → ACMA	ACK	ACMA	→ ⑨
		ACMA → ISP	ETX		
3	中継処理待ち	ISP → ACMA	ETX	中継処理	→ ⑩
		ACMA → ISP	ACK		
4	初回 ACMA 待ち	ISP → ACMA 又は ACMA → ISP	EOT, INF=1	ACMA	→ ⑪
5	4. 以外のすべての EOT 受信				→ ⑫
6	上記 1 ~ 5 以外のすべての組み合わせ				→ ⑬

- ⑧ [ENQ 衝突] 300mS 待ち、再び ENQ を通信相手先の ACMA に送信する。エラーの場合は終了情報をエラーとし ⑭ へ行く。正常ならば通信ロギングを行い ② へ行く。
- ⑨ [ACMA より受信] ETX メッセージの場合は対応する ACMA の LAME よりデータを ACMA データバッファへ読み込む。ブロック転送エラーの場合は 1 回だけ再試行する。他のエラー、又は 2 回目のブロック転送エラーの場合はエラーをロギングし、終了情報をエラーとして ⑭ へ行く。ACK メッセージの場合、又は ETX メッセージで LAME の読み込みが正常だった場合はメッセージタイプを中継処理 pid 相当とし、受信メッセージを中継処理に通知する。通信ロギングを行い、通信ステータスを中継処理待ちとする。② へ行く。
- ⑩ [中継処理より受信] 要求元の pid を中継処理 pid に設定する。ETX メッセージの場合は ACMA データバッファより対応する ACMA の LAME にデータを書き込む。ブロック転送エラーの場合は 1 回だけ再試行する。他のエラー、又は 2 回目のブロック転送エラーの場合はエラーをロギングし、終了情報をエラーとして ⑭ へ行く。ACK メッセージの場合、又は ETX メッセージで LAME への書き込みが正常だった場合は受信メッセージを通信相手先の ACMA に送信する。エラーの場合はロギングし、終了情報をエラーとして ⑭ へ行く。正常ならば通信ロギングを行い、通信ステータスを ACMA 待ちとする。② へ行く。
- ⑪ [EOT, INF=1] 通信ステータスを ACMA 内衝突とし ② へ行く。
- ⑫ [EOT 受信] 終了情報を EOT の終了情報から取り出して設定する。中継処理から受信した場合は中継処理 pid を 0 とする。LAM 受信処理から受信した場合は ACMA クレートアドレスを 0 とする。⑭ へ行く。
- ⑬ [プロトコルエラー] エラーをロギングし、終了情報をエラーとして ⑭ へ行く。
- ⑭ [終結] 中継処理 pid が 0 でない場合は、メッセージタイプを中継処理 pid 相当とし、終了情報を EOT メッセージに設定して、中継処理に EOT メッセージを通知する。中継処理 pid を 0 とする。通信ロギングを行う。ACMA クレートアドレスが 0 でない場合は終了情報を EOT に設定して ACMA に EOT を通知する。エラーの場合はロギングする。正常ならば通信ロギングを行う。どちらの場合でも ACMA クレートアドレスを 0 とする。通信ステータスを準中立とし ② へ行く。

### 3.1.2 LAM 受信処理

- ① 共有メモリを割り当て起動メッセージをロギングする。起動パラメタからループ番号を読み込み、ループ番号に対応するシリアルドライバ及び ACMA 通信要求 MBX を決定する。シリアルドライバをオープンする。エラーの場合はロギングし処理を終了する。正常ならば②へ行く。
- ② 全下位クレートのハングデマンド及び ACMA から LAM を待つ。ハングデマンドを受信した場合はハングデマンドをクリアし、ACMA からのハングデマンドを確認後 CCRS を読み込む。ACMA からの LAM を受信した場合は CCRS を読み込む。エラーの場合はロギングし②へ行く。正常ならば読み込んだメッセージを ACMA 通信処理へ通知する。②へ行く。

### 3.1.3 中継処理

- ① 共有メモリを割り当て起動メッセージをロギングする。標準入力からループ番号及び方向標識を読み込み、ループ番号に対応する ACMA 通信要求 MBX、及び自身の動作を RQPUT 中継処理とするか、RQGET 中継処理か決定する。SIGALARM シグナルでシステムコールを再始動しないよう指定する。また SIGPIPE シグナルを受信できるよう設定する。RQPUT 中継処理の場合は②へ、RQGET 中継処理の場合は③へ行く。
- ② [RQPUT 中継処理] ISP からの RQ パケットを待ち合わせる。メッセージタイプを 64 とし、ENQ メッセージを ACMA 通信処理に通知する。④へ行く。受信エラー、又はフォーマットエラーの場合はエラーをロギングし終了する。
- ③ [RQGET 中継処理] ACMA 通信要求 MBX でメッセージタイプ 65 のみを待ち合わせる。ENQ メッセージを受信した場合は ISP へ RQ パケットを送信する。⑤へ行く。
- ④ [ACMA 待ち] 時間監視を開始し ACMA 通信要求 MBX で自己の pid 相当のメッセージタイプのみを待ち合わせる。メッセージを受信した場合は時間監視を解除する。データ転送方向が ISP → ACMA で ACK メッセージを受信した場合は DT パケットを ISP に送信する。データ転送方向が ACMA → ISP で ETX メッセージを受信した場合は ACMA データバッファよりデータを読み込み DT パケットを組み立てて ISP に送信する。EOT メッセージを受信した場合は⑥へ行く。ISP 送信時に SIGPIPE シグナルが発生した場合は受信したメッセージをそのまま ACMA 通信要求 MBX に通知し、ロギングして終了する。上記以外のメッセージの組み合わせ、タイムアウト、又は送信エラーの場合はエラーをロギングし終了する。

- ⑤ [ISP 待ち] 時間監視を開始し ISP からの DT, RS 又は RE パケットを待ち合わせる。DT パケットを受信した場合で、データ転送方向が ISP → ACMA の場合は受信データを ACMA データバッファに書き込んだ後、ACMA 通信タスクに ETX メッセージを通知する。DT パケットを受信した場合でデータ転送方向が ACMA → ISP の場合は ACMA 通信タスクに ACK メッセージを通知する。RQGET 中継処理で RS パケットを受信した場合、又は RQPUT 中継処理で RE パケットを受信した場合は⑦へ行く。上記以外のパケットの組み合わせ、タイムアウト、又は受信エラーの場合はエラーをロギングし終了する。
- ⑥ [EOT 受信] RQGET 中継処理の場合は RE パケットを ISP に送信する。③へ行く。RQPUT 中継処理の場合は RS パケットを ISP に送信する。②へ行く。送信エラーの場合はエラーをロギングし終了する。
- ⑦ [RS/RE 受信] ER=X'FFFF' の場合はエラーの EOT メッセージを作成する。ER=0 ならば UC の内容のまま ACMA 通信処理に EOT メッセージを通知する。RQGET 中継処理の場合は③へ、RQPUT 中継処理の場合は②へ行く。

### 3.2 高速データ収集機能

高速データ収集機能は読み込み処理と転送処理の 2 つの処理により構成される。高速データ収集機能の各処理はシリアルハイウェイに対して対になって存在し、データ収集と転送を行う。転送処理は ISP からの接続要求があるごとに inetd より起動される。読み込み処理は転送処理より起動され、私用 MBX で読み込み要求を待ち合わせる。高速データ収集機能のシーケンスの例を図 6 に示す。

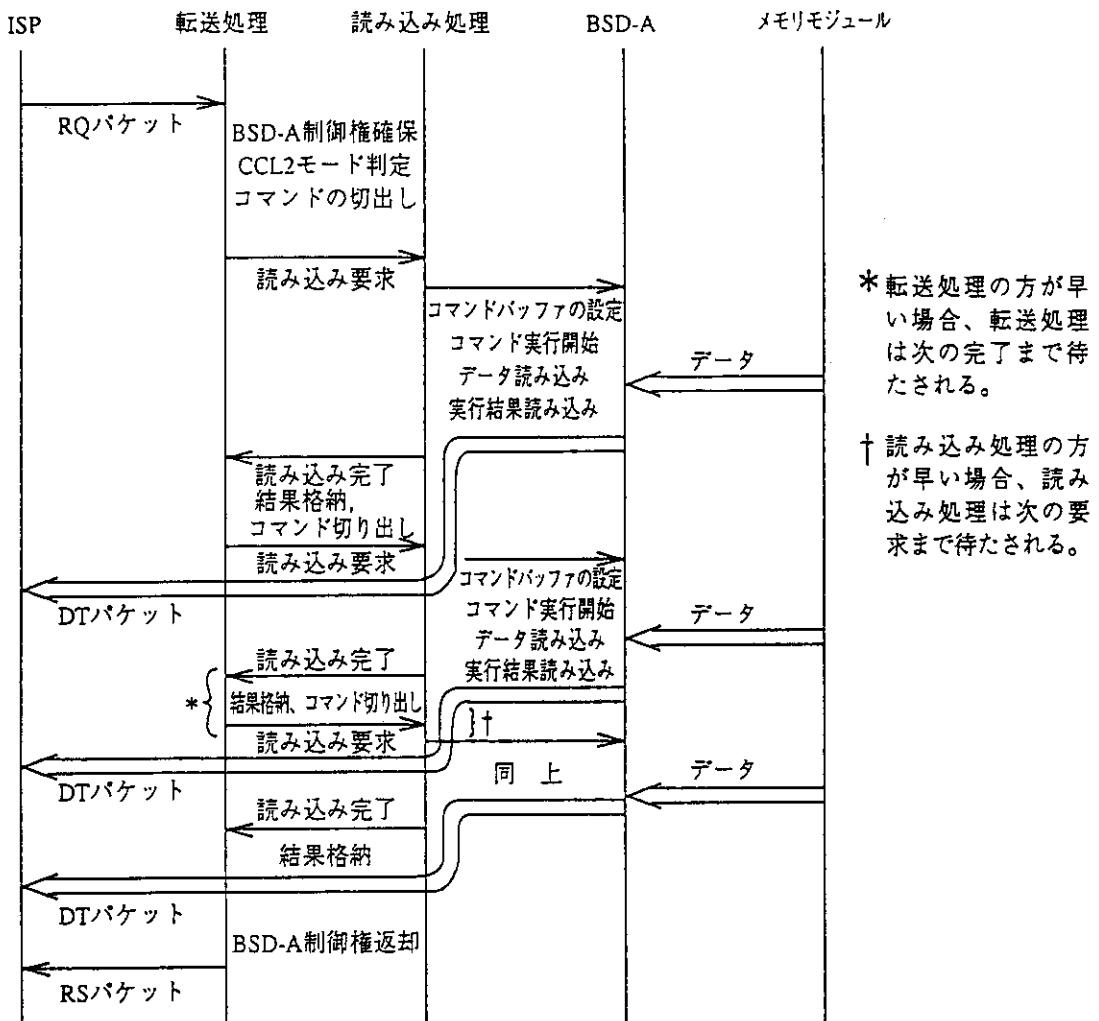


図6 高速データ収集機能 シーケンスの例

### 3.2.1 転送処理

①共有メモリを割り当て起動メッセージをロギングする。標準入力からループ番号及び方向標識を読み込み、ループ番号に対応するシリアルドライバを決定してオープンする。読み込み処理との通信用の私用 MBX 及び私用共有メモリを割り当て、読み込み処理を起動する。SIGTERM シグナル受信時に ⑥に行くよう設定する。SIGALARM シグナルでシステムコールを再始動しないように指定する。以上が正常に終了した場合は ②へ行く。エラーの場合はロギングして終了する。

②ISP からの RQ パケットを待ち合わせる。受信エラー、又はフォーマットエラーの場合はエラーをロギングして ⑥へ行く。正常ならば CCL2 の動作モード（エンハンスト／通常）の判定を行う。またループ番号に対応した BSD-A ロックセマフォより UNDO オプション付で BSD-A の制御権を獲得する。ISP から指示された CAMAC コマンドのワードカウント (WC) を調べ、バッファ長を越えない範囲でコマンドを集め、読み込み処理に読み込み要求を通知する。③へ行く。  
(WC=0 は 65536 ワードであることに注意、16/24 ビットも考慮)

- ③ 時間監視を開始し、私用 MBX で読み込み完了又は読み込みエラーメッセージのみを待ち合わせる。タイムアウトの場合はエラーをロギングして⑥へ行く。メッセージを受信した場合は時間監視を解除する。コマンド実行結果をメッセージより取り出し、RS パケットに設定する。読み込み完了の場合は④へ、読み込みエラーの場合は⑤へ行く。
- ④ [読み込み完了] ISP より指示された CAMAC コマンドが残っている場合は、ワードカウント (WC) を調べ、バッファ長を越えない範囲でコマンドを集め、読み込み処理に読み込み要求を通知する。DT パケットを ISP に送信する。送信エラーの場合はロギングし⑥へ行く。正常ならば③へ行く。ISP より指示された CAMAC コマンドが残っていない場合は DT パケットを ISP に送信する。送信エラーの場合はロギングし⑥へ行く。正常ならば⑤へ行く。
- ⑤ [後処理] BSD-A の制御権を解放する。RS パケットを組み立て、ISP に送信する。送信エラーの場合はロギングし⑥へ行く。正常ならば②へ行く。
- ⑥ [エラー] 読み込み処理に SIGTERM シグナルを送信する。読み込み処理の終了を待ち合わせた後、私用 MBX、私用共有メモリを削除する。シリアルドライバをクローズ、終了をロギングしてプログラムを終了する。

### 3.2.2 読み込み処理

読み込み処理は転送処理との通信用の私用 MBX、私用共通メモリの割り当て、及びシリアルドライバのオープンが完了した状態で転送処理より起動される。

- ① 起動メッセージをロギングする。②へ行く。
- ② 私用 MBX で読み込み要求のみを待ち合わせる。BSD-A の初期化（コマンドバッファ、データバッファのクリア、BSD-A 動作モードの設定等）、コマンドバッファの設定、及び BSD-A の起動を行う。CAMAC コマンドの実行結果をロギングする。エラーの場合は前処理エラーフラグをオンとして④へ行く。前処理が正常ならば、BSD-A のデータバッファから指定されたデータ数、及びモードで指定された私用共有メモリにデータを読み込む。CAMAC コマンドの実行結果をロギングする。24ビットモードの場合は各ワードの先頭1バイトを0に設定する。（この処理は camacb() 内で実行されている）。エラーの場合はデータ収集エラーフラグをオンとして③へ行く。正常ならば③へ行く。（データ収集エラーは再試行しない）
- ③ BSD-A が動作中の場合は終結させる。BSD-A より処理結果を読み込む。BSD-A の設定を元に戻す。CAMAC コマンドの実行結果をロギングする。エラーの場合は後処理エラーフラグをオンとする。④へ行く。
- ④ 処理結果をロギングし、結果を転送処理に通知する。②へ行く。

### 3.3 単一要求処理機能

単一要求処理機能は ISP からの要求に従って処理を行い、RS パケットのみを返す処理の総称で、本体室アクセス処理、計調室アクセス処理、オンライン要求処理、及び初期設定要求処理の 4 つの処理より構成される。これらの処理は ISP からの接続要求があるごとに inetd より起動される。

#### 3.3.1 本体室アクセス処理

- ① 共有メモリを割り当て起動メッセージをロギングする。標準入力からループ番号及び方向標識を読み込み、ループ番号に対応するシリアルドライバをオープンする。エラーの場合はロギングし終了する。正常ならば②へ行く。
- ② ISP からの RQ パケットを待ち合わせる。受信エラー又はフォーマットエラーの場合はエラーをロギングして終了する。正常ならばループ番号に対応した BSD-A ロックセマフォより UNDO オプション付で BSD-A の制御権を獲得する。次に BSD-A の LAM マスクレジスタの内容を退避後割り込み禁止とし、転送モードを A モードに、コマンドバッファ及びデータバッファを設定して BSD-A を起動する。CAMAC コマンドの実行結果をロギングする。以上正常なら③へ、エラーなら④へ行く。
- ③ 20, 40, 80, … 640 mS 遅延して、BSD-A の動作完了をチェックする。完了した場合はデータバッファを読み込む。完了しなかった場合はエラーとして、BSD-A を強制終結する。BSD-A より処理結果を読み込む。BSD-A の設定を復旧する。CAMAC コマンドの実行結果をロギングする。④へ行く。
- ④ BSD-A の制御権を解放する。ISP に RS パケットを送信する。送信エラーの場合はロギングし終了する。正常ならば②へ行く。

#### 3.3.2 計調室アクセス処理

- ① 共有メモリを割り当て起動メッセージをロギングする。標準入力からループ番号及び方向標識を読み込み、ループ番号に対応するシリアルドライバをオープンする。エラーの場合はロギングし終了する。正常ならば②へ行く。
- ② ISP からの RQ パケットを待ち合わせる。受信エラー又はフォーマットエラーの場合はエラーをロギングして終了する。パケットで指定された CAMAC コマンドを実行する。コマンドの実行結果をロギングする。ISP に RS パケットを送信する。送信エラーの場合はロギングし終了する。正常ならば②へ行く。

### 3.3.3 オンライン要求処理

- ① 共有メモリを割り当て起動メッセージをロギングする。標準入力からループ番号及び方向標識を読み込み、ループ番号に対応するシリアルドライバをオープンする。エラーの場合はロギングし終了する。正常ならば②へ行く。
- ② ISPからのRQパケットを待ち合わせる。受信エラー又はフォーマットエラーの場合はエラーをロギングして終了する。正常ならば指定されたクレートのCCL2, ACMA, 及び LAMEに対しアンバイパス、オンライン、CCRSの空読み、CCRSのLAMイネーブル、LAME N=20イネーブル、CCL2デマンドイネーブルを行う。アンバイパスコマンドはエラーをチェックせず、かつ次のコマンド発行まで300mS以上遅延するものとする。また次のコマンドのエラーもチェックしない。結果をRSパケットに設定しISPに送信する。送信エラーの場合はロギングし終了する。正常ならば②へ行く。

### 3.3.4 初期設定要求処理

基本的にオンライン要求処理と同じである。違いはRQパケットのフォーマット及びCICUの時刻設定を行う点である。

## 3.4 ダミー処理機能

ダミー処理機能は旧CICUとの互換性を確保するために存在するもので、IRT処理のみから構成される。

### 3.4.1 IRT処理

- ① 共有メモリを割り当て起動メッセージをロギングする。標準入力からループ番号、及び方向標識を読み込む。方向標識にかかわらずISPよりデータを待ち合わせる。データを受信しても受信エラーになってもロギングし終了する。

## 3.5 システム監視機能

システム監視機能はシステム全体の初期化と常駐プロセスを監視する統括処理、及び各種の事象を記録するロギング処理の2つの処理より構成される。統括処理はCICUの起動時にrcファイルより起動される。ロギング処理は特定のプロセスではなく、各処理に組み込まれて使用される。

### 3.5.1 統括処理

- ① 共有メモリを割り当てる。起動パラメタから退避ファイル名を取り出す。取り出せない場合はデフォルト値を使用する。ファイルが存在する場合はファイルより共有メモリを初期設定する。ファイルが存在しない場合は初期設定ファイルを読み込み、共有メモリを初期設定する。ACMA通信要求MBX, BSD-Aロックセマフォをシリアルハイウェイ数だけ割り当て、初期設定する。ロギングロックセマフォを割り当て初期設定する。

ループ番号を引数とし ACMA 通信処理、及び LAM 受信処理を起動する。各処理の再起動カウンタを 0 とする。SIGTERM シグナルを受信した場合③へ行くよう設定する。以上が正常に終了した場合は②へ、エラーの場合はコンソールにエラーを表示して終了する。

- ② 子プロセスの終了を待ち合わせる。終了した場合は終了した処理を調べ、再起動カウンタを 1 増やす。再起動カウンタが起動回数の上限に達していない場合は該当する処理を再起動する。起動回数の上限に達している場合、又は再起動エラーの場合は③へ行く。正常な場合は②へ行く。
- ③ [終了] 起動中の子プロセスに SIGTERM シグナルを送信する。共有メモリを退避ファイルに退避する。終了する。

### 3.5.2 ロギング処理

- ① ロギング用共有メモリが割り当てられていない場合は割り当てる。イベント又はエラーロギングでロギングロックセマフォが割り当てられていない場合は割り当てる。イベント又はエラーロギングの場合はロギングロックセマフォをロックする。②へ行く。
- ② 日付、時刻及び現処理名を取り出す。サブルーチン名によりエラー、イベント又は通信のいずれかのロギングエリアにロギングを書き込む。ロギングの現ポインタを 1 つ進める。イベント又はエラーロギングの場合はロギングロックセマフォをアンロックする。③へ行く。
- ③呼び出し元ヘリターンする。

## 3.6 コンソール機能

コンソール機能は CICU のコンソールからコマンドを受け付け、これを処理するもので、単発の CAMAC コマンドを実行する CAMAC コマンド実行処理、ロギングデータを表示するロギング出力処理、及び共有メモリの表示、修正を行う表示処理の 3 つの処理より構成される。

### 3.6.1 CAMAC コマンド実行処理

- ① 起動パラメタから入力ファイル名を得てオープンする。引数がない場合は標準入力からコマンドを入力するものとする。引数エラー、又はオープンエラーの場合はエラーメッセージを出力し終了する。正常ならば②へ行く。
- ②シリアルドライバをオープンエラーになるまで順次オープンする。1 つもオープンできない場合はエラーメッセージを出力し終了する。正常ならば③へ行く。

- ③ ファイル又は標準入力よりハイウェイ番号 B、計調室クレートアドレス C、BSD-Aステーションアドレス N、本体室クレートアドレス c、ステーション番号 n、サブアドレス a、ファンクション f、及びデータ d を読み込み、シングルアクションの CAMAC コマンドを発行する。c=0 の場合は計調室クレートに対して、c≠0 の場合は BSD-A を使って本体室クレートに対してアクセスを行う。結果を標準出力に表示する。③へ行く。入力が EOF になった場合は④へ行く。
- ④ [EOF] オープンしているシリアルドライバをクローズする。入力ファイルをオープンしている場合は、これをクローズする。プログラムを終了する。

### 3.6.2 ロギング出力処理

- ① 起動パラメタから出力するロギングのテーブル名、及び出力行数を得る。出力行数を得られない場合はデフォルトを 10 とする。テーブル名を得られない場合、又はテーブル名が間違っている場合はエラーメッセージを出力し終了する。正常ならば②へ行く。
- ② 指定されたテーブル名に相当する共有メモリを割り当てる。ロギングを出力行数又は実ロギング行数のうち短い方で決まる行数だけ出力する。プログラムを終了する。

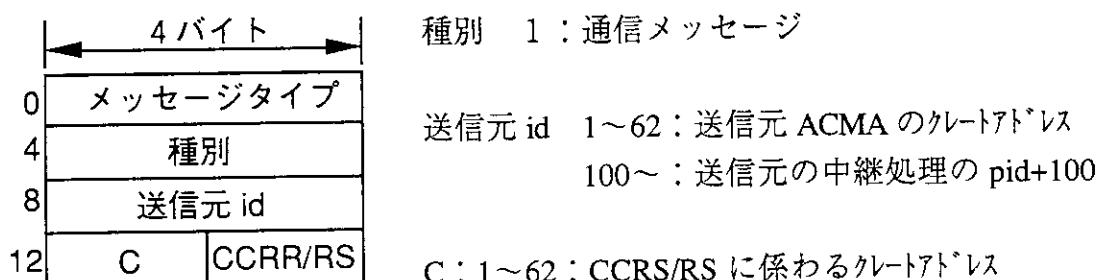
### 3.6.3 表示処理

- ① 起動パラメタから表示、変更する共有メモリのテーブル名を得る。テーブル名を得られない場合、又はテーブルが間違っている場合はエラーメッセージを出力して終了する。正常ならば②へ行く。
- ② 標準入力よりコマンドを読み込む。指定されたオフセットとデータ長に従って表示を行う。終了コマンドで終了する。

## 4. 共有資源

### 4.1 MBX

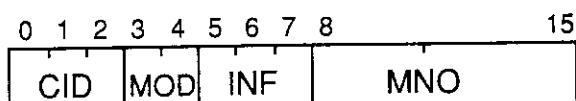
#### 4.1.1 ACMA 通信要求 MBX (シリアルハイウェイ毎に存在)



#### メッセージタイプ

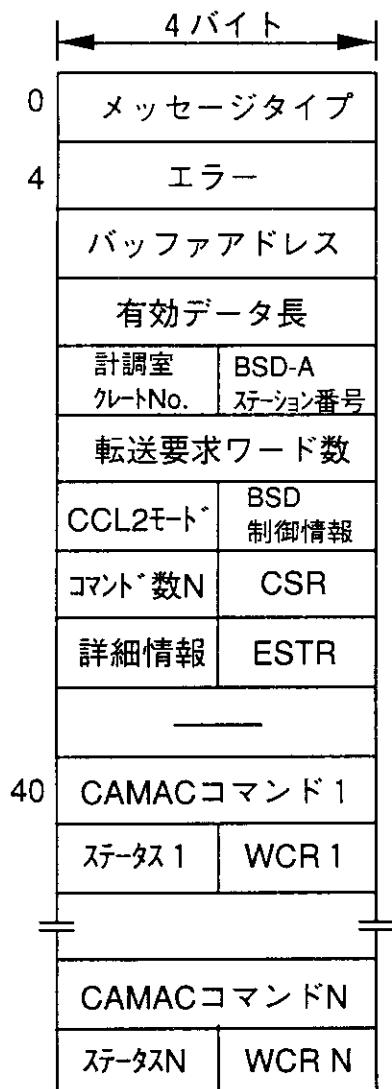
- 1~62 : 通信中の中継処理、又は ACMA からのメッセージ
- 64 : RQPUT 中継処理からの通信開始要求
- 65 : RQGET 中継処理への通信開始要求
- 100~ : 通信中の中継処理へのメッセージ

#### CCRR/RS



種別 1 の場合→詳細は 6.3 参照

## 4.1.2 高速データ収集用私用 MBX (高速収集のコネクション毎に1つ)



メッセージタイプ 1 : 読み込み要求  
2 : 読み込み完了／エラー

エラー 0 : 正常  
≠ 0 : エラー

バッファアドレス : 読み込み先頭アドレス

有効データ長 : 収集できたデータ長 (バイト)

CCL2 モード : 

0	1	2														15
-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

  
 0 : 通常モード  
 1 : エンハンストモード

その他は 5.3.5 及び 5.3.6 参照のこと

#### 4.2 セマフォ

##### ① BSD-A ロックセマフォ

あるシリアルハイウェイ下の BSD-A の制御権の逐次化に使用する。

シリアルハイウェイごとに存在する。初期値 1

##### ② ロギングロックセマフォ

イベントロギング及びエラーロギング書き込みの逐次化に使用する。初期値 1

#### 4.3 データバッファ

##### ① ACMA データバッファ

ACMA との通信時、DT パケット及び LAME のデータの中継を行う。大きさは 4 KB。内容については CICU は関知しない。シリアルハイウェイごとに 1 つ存在する。

##### ② 高速データ収集用私用データバッファ

高速データ収集の収集データのバッファリングに使用する。大きさ 256 KB。

高速データ収集のコネクションごとに 2 つずつ存在し、交互に読み込みと ISP への転送を行う。

#### 4.4 ロギングエリア

##### ① イベントロギング

要求の受付、プロセスの起動、CAMAC コマンドの実行結果など CICU の主要な動作を記録する。

##### ② エラーロギング

イベントロギングのうち、エラーに係わるものここに記録する。

##### ③ 通信ロギング

ACMA 通信処理のメッセージ送受信記録である。シリアルハイウェイごとに 1 つ存在する。このロギングは ACMA 通信処理のみが行うのでロギングロックセマフォを使用する必要はない。

## 5. ISP との通信

### 5.1 通信の概要

本通信は計測ネットワークを介して ISP の ACMIH、ACQ 等の応用プログラムとの間でデータを伝送するものである。通信は半2重であるが、ISP と CICU のどちらかが送信権を持つようになっているので競合は発生しない。本通信の概要を図 7 に示す。

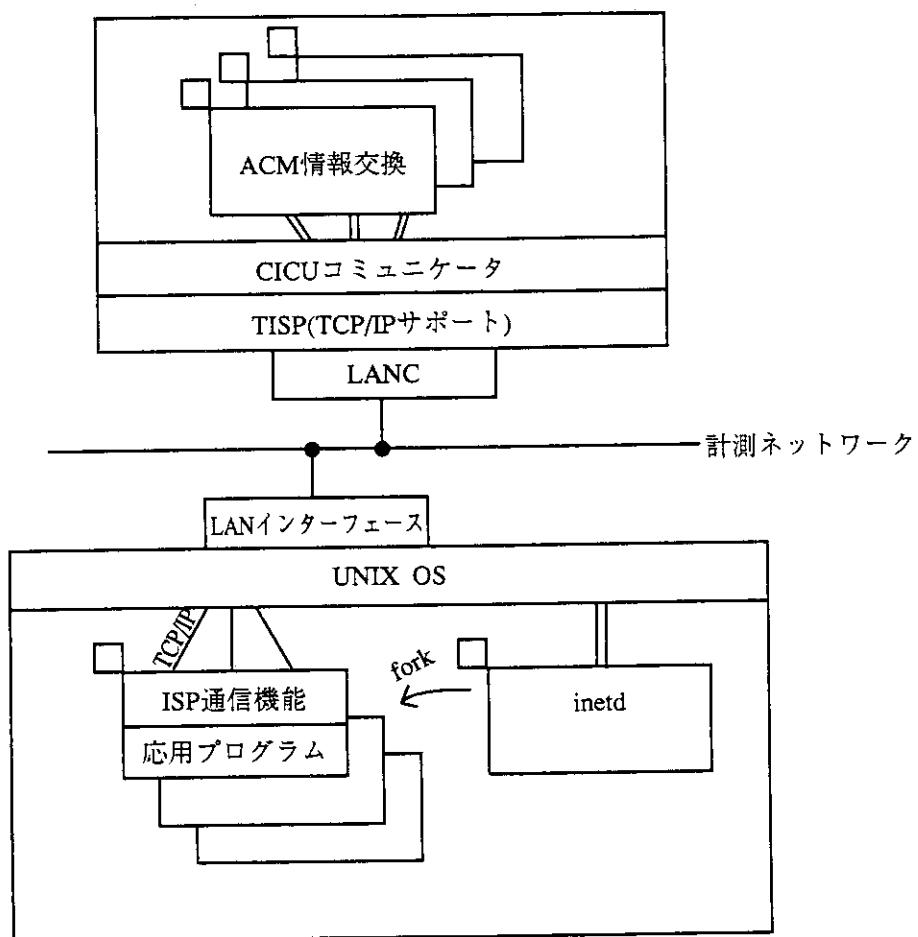


図 7 ISP との通信の概要

## 5.2 通信プロトコル

### 5.2.1 伝送媒体

伝送媒体として TCP/IP のコネクションを使用する。CICU は通信要求コードに対応するポートでコネクション開設要求を待ち合わせる。開設されたコネクションを使用してデータを伝送する。

### 5.2.2 シーケンス

各通信要求コードに対してどちらが通信を開始できるかあらかじめ決定されている。通信の開始は RQ パケットに始まり RS (又は RE) パケットにより終了する。通信シーケンスを図 8 に示す。

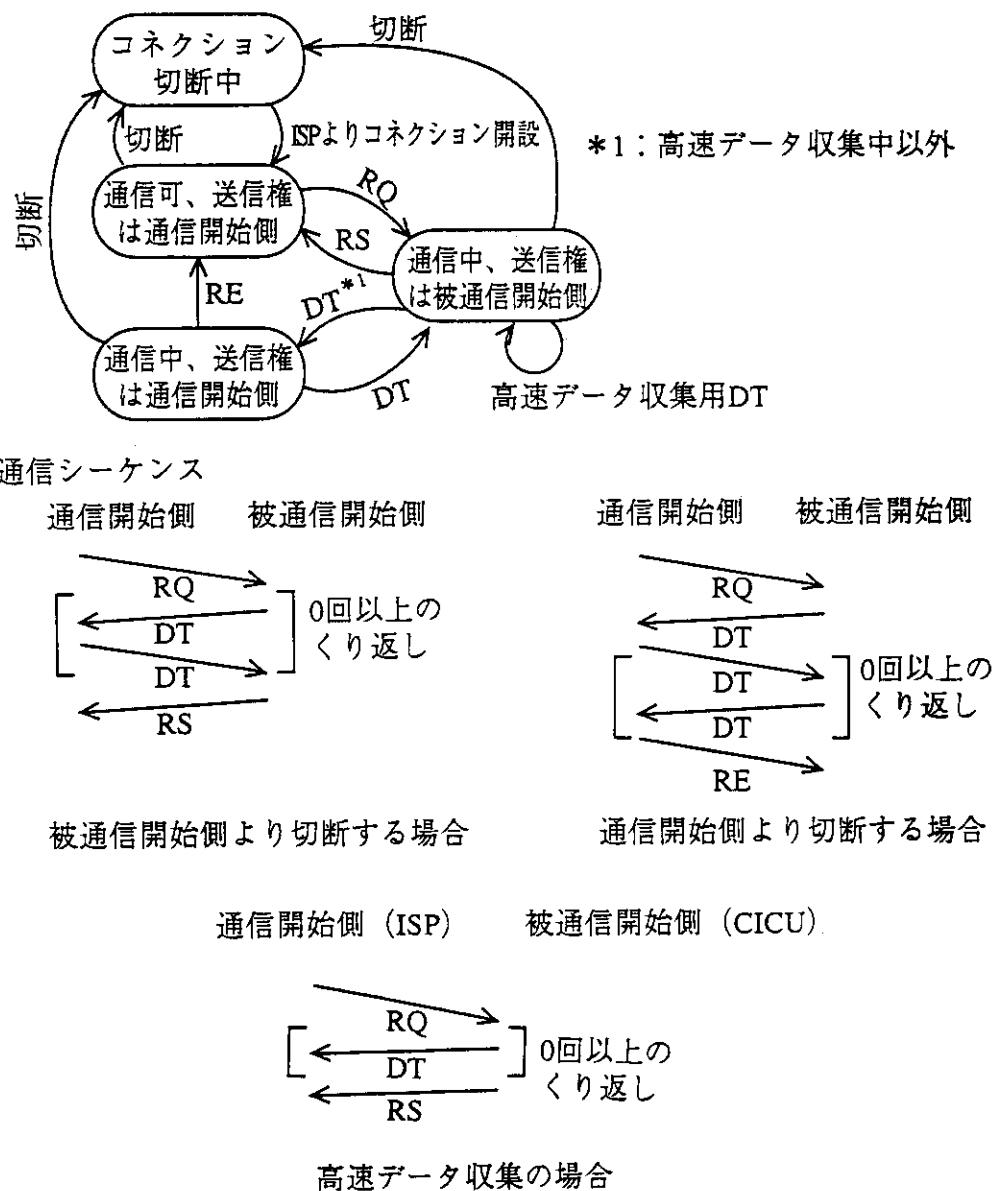
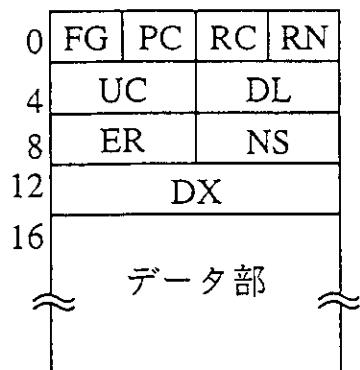


図 8 ISP との通信シーケンス

### 5.3 フォーマット

#### 5.3.1 一般フォーマット



FG : X'7E 固定

PC : パケットコード

X'01' RQパケット X'02' DTパケット

X'04' RSパケット X'10' REパケット

RC : 通信要求コード

X'06' 初期設定要求

X'01' 高速データ収集要求

X'02' ACMA交信要求 (ISP起動)

X'12' ノード (ACMA起動)

( X'15' IRT要求…通信なし )

X'09' 本体室アクセス要求

X'03' 計調室アクセス要求

X'08' オンライン要求

RN : 通信要求コードごとのパケット通番。RQパケットが発生すると

$$RN_{new} = (RN_{old} + 1) \bmod 128$$

により新しいRNがRQパケットに割り当てられる。最初のRQパケットのRNは1である。引き続くDT又はRSパケットのRNはRQパケットのものと同一の値である。

UC : ユーザ定義 (任意)

DL : データ部の長さ(バイト)ただし高速データ収集の場合はDXフィールドが使用される。

ER : RS/RE パケットのみ有効で通信システムのエラーコードを示す。

0 : 正常 他 : エラー

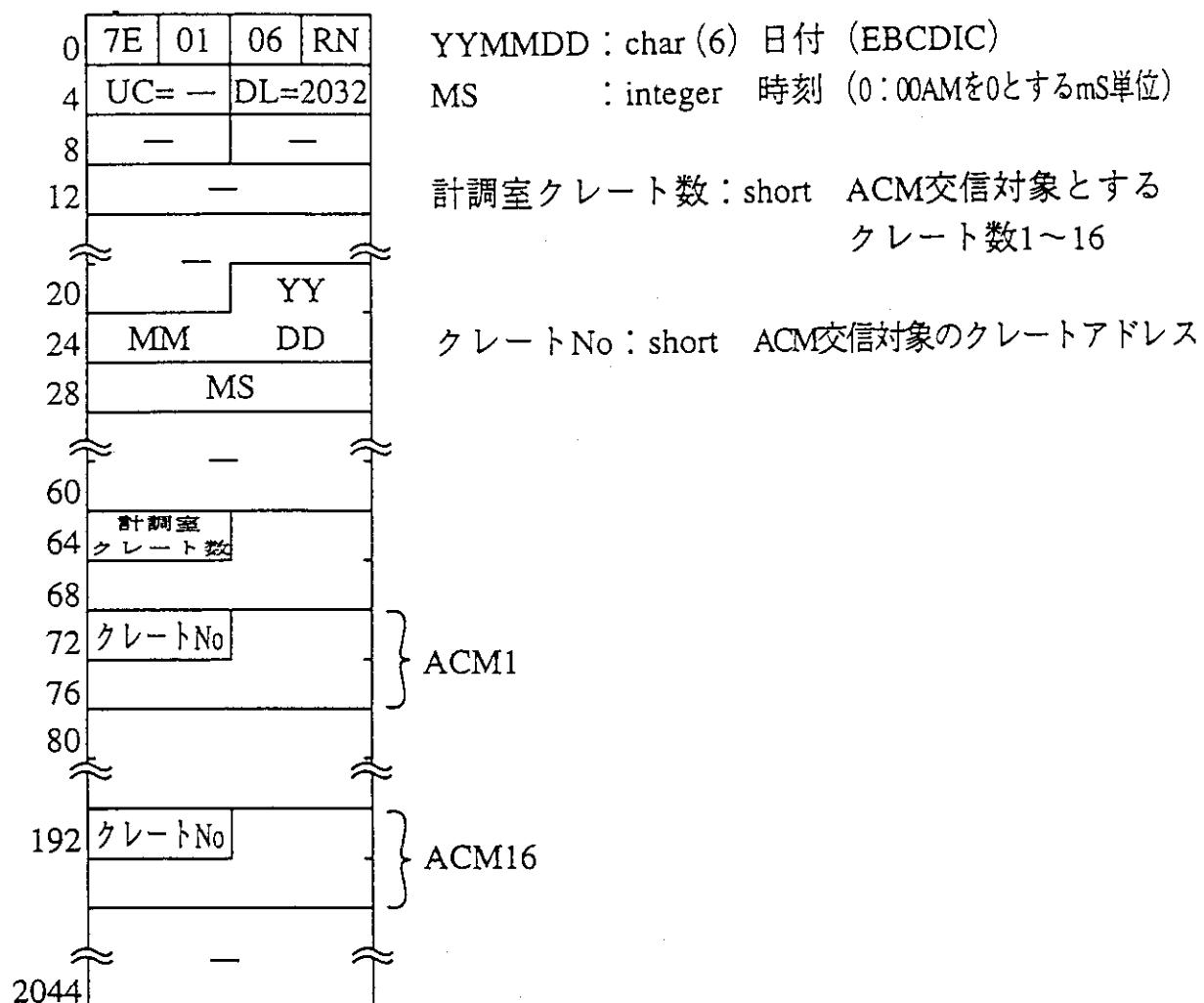
DTパケットでは0、RQパケットでは不定となる。

NS : DTパケットのみ有効でDTパケットの通番を示す。初期値1

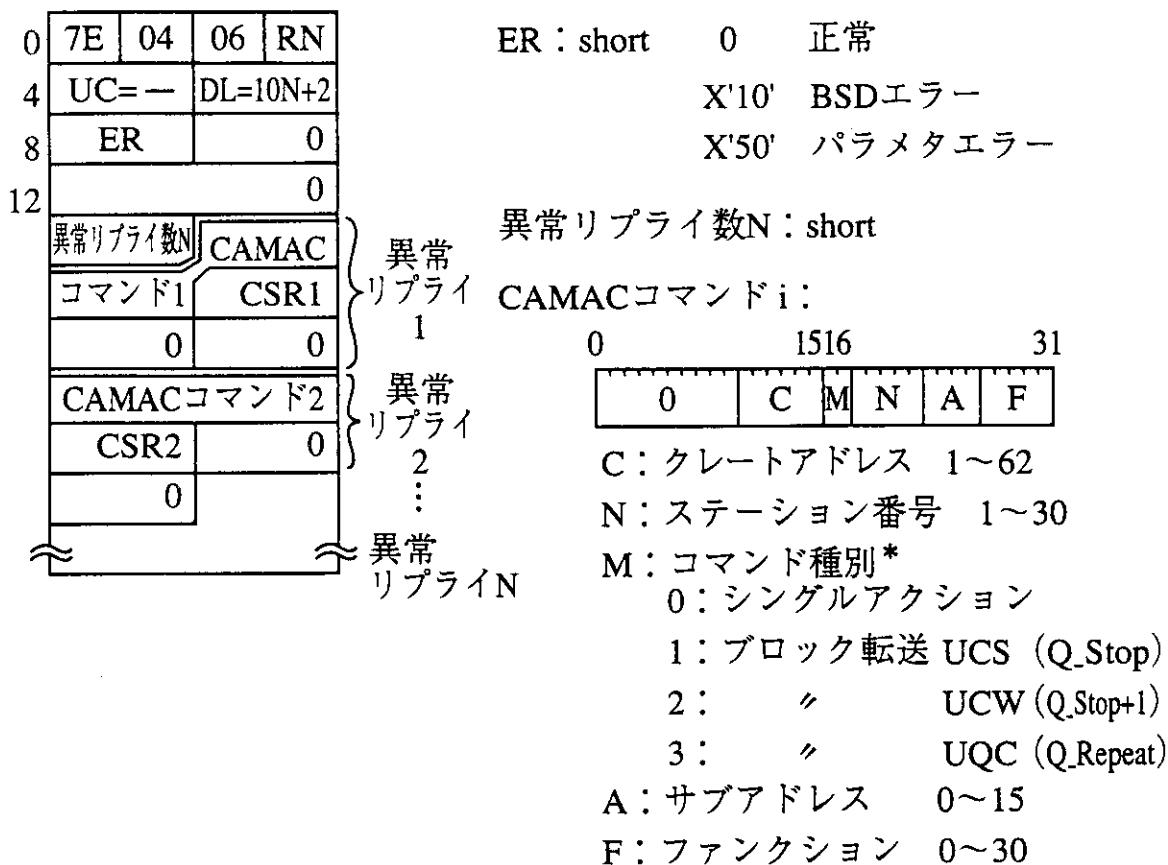
次のDTパケットのNSは $NS_{new} = (NS_{old} + 1) \bmod 256$ により割り当てられる。RQ, RS, 及びREパケットでは不定。

DX : 高速データ収集のDTパケットで、データ部の長さ(バイト)を表わす。他の場合は不定となる。

## 5.3.2 初期設定要求 RQ パケット



### 5.3.3 初期設定要求 RS パケット



## CSR i：BSDのステータス情報

bit 0	DERR	前コマンドのエラー	bit 8	DERR	bit0と同じ
bit 1	SQ	Q=1	bit 9	TMO	ハード検出タイムアウト(7秒)
bit 2	SX	X=1	bit 10	NSYNC	同期ずれ
bit 3	STE	传送エラー(bit11,13,14のOR)	bit 11	RPYE	クレートでエラー検出
bit 4	STMO	ソフトウェアタイムアウト	bit 12	0	
bit 5	NEX	実行されなかった	bit 13	LPE	垂直パリティエラー
bit 6	N>23	0	bit 14	TPE	水平パリティエラー
bit 7	STE	bit3と同じ	bit 15	ANDR	クレートアドレスが認識されない。

\* 初期設定要求RSパケットではM=0である。

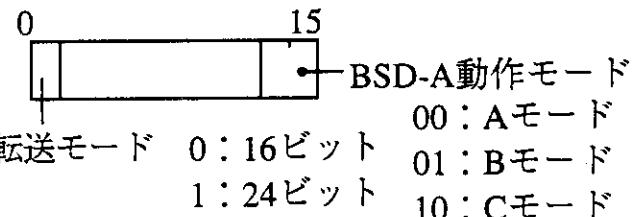
## 5.3.4 高速データ収集要求 RQ パケット

0	7E	01	01	RN
4	UC=—	DL=	4336	
8	—	—		
12	—	—		
16	計調室クレート No	BSD-A ステーション番号		
20	転送ワード数			
24	—	BSD制御情報		
28	コマンド数N			
32				
36				
40	CAMACコマンド1			
44	—	WCR1		
48	~~	~~	~~	~~
52	CAMACコマンドN			
56	—	WCRN		
60	~~	~~	~~	~~

4352

計調室クレートNo : short  
 BSD-Aステーション番号 : short  
 転送ワード数 : integer BSD-Aのデータバッファ  
 から読み込むデータ数

## BSD 制御情報 :



コマンド数N : short CAMACコマンド数  
 CAMACコマンドi : 5.3.3参照 ただし、Mはブロック転送  
 WCRi : unsigned short BSD-Aの最大コマンドくり  
 返し回数。  
 0~65535。  
 0は65536回と見なす。

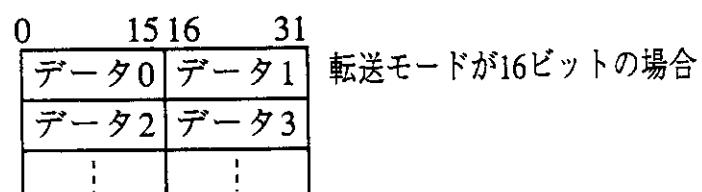
## 5.3.5 高速データ収集要求 DT パケット

0	7E	02	01	RN
4	UC=—	DL=—		
8	0	NS		
12	DX			
16	~~	~~	~~	~~
20	収集データ			
24	~~	~~	~~	~~

NS : short DTパケット通番。初期値1 225→0

DX : integer 収集データ長 (バイト)

## 収集データ :



## 5.3.6 高速データ収集要求 RS パケット

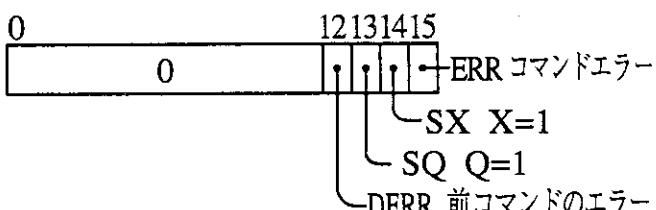
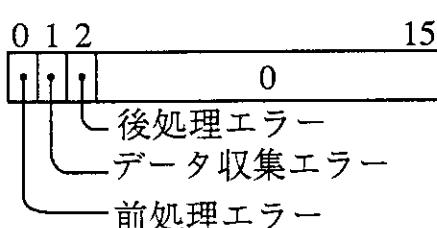
0	7E	04	01	RN
4	UC=--	DL=8N+24		
8	ER	—		
12	—			
16	計調室クレート No	BSD-A ステーション番号		
20	転送ワード数			
24	—	BSD制御情報		
28	リプライ数N	CSR		
32	詳細情報	ESTR		
36	—			
40	CAMACコマンド1		リプライ1	
	ステータス1	WCR1		
<hr/>				
	CAMACコマンドN		リプライN	CSR : BSDエラー発生時のみ有効で
	ステータスN	WCRN		BSDのステータス情報が設定される。5.3.3参照

ER : short 0 正常  
 X'10' BSD エラー  
 X'20' BSD-A エラー  
 X'50' パラメタエラー

計調室クレートNo :  
 BSD-Aステーション番号 : RQパケットの値を  
 転送ワード数 : そのまま設定する。  
 BSD制御情報 : 5.3.4参照

リプライ数N : short  
 BSD-Aコマンドバッファより読み出しができたリプライの数

CAMACコマンドi : 5.3.3参照  
 WCRI : unsigned short 転送できなかったワード数  
 ステータスi :



ESTR : BSD-Aのステータス情報

bit 0	DERR	前コマンドのエラー	bit 10	RPE	Q=0リピート回数オーバー
bit 1	SQ	Q=1	bit 11	SERR	同期はずれ
bit 2	SX	X=1	bit 12	TMO	350mS経過してもリプライなし
bit 3	ERR	传送エラー	bit 13	TPE	垂直パリティエラー
bit 6	PER4	コマンドバッファパリティエラー	bit 14	LPE	水平パリティエラー
bit 7	PER3	コマンドク	bit 15	ADNR	クレートアドレスが認識されない。
bit 8	PER2	リードFIFOク			
bit 9	PER1	ライトFIFOク	bit4, 5	は0	

## 5.3.7 ACMA 交信要求 RQ パケット

0	7E	01	RC	RN
4	UC=—	DL=16		
8	0	0		
12	0			
16	発信アドレス	着信アドレス		
20	—			
24	—			
28	—			

RC : byte      X'02' ISP起動ACMA交信要求  
                   X'12' ACMA起動 ク

UC : CCRR/RS フォーマット

0	234578	15
001	MOD	MNO

MOD : 0 データ無  
       1 ISP→ACMA  
       2 ACMA→ISP

MNO : メッセージ番号

発信アドレス short : ISP起動の場合 X'0080'  
                           ACMA起動 クレートアドレス

着信アドレス short : ISP起動の場合 相手クレートアドレス  
                           ACMA起動 ク X'0080'

## 5.3.8 ACMA 交信要求 DT パケット

0	7E	02	RC	RN
4	UC	DL		
8	0	NS		
12	0			
16	発信アドレス	着信アドレス		
20	—			
24	0			
28	—			
32	送受信データ (最大4096バイト)	DL		

RC : 5.3.7参照

UC : CCRR/RS フォーマット

0	234578	15
CID	0 INF	MNO

CID : 2 ACK    INF : CID=3のとき  
       3 NAK            2 マイクロディスクアクセスエラー  
       5 ETX            3 受信データオーバー  
                         4 CAMACアクセスエラー

MNO : メッセージ番号

発信アドレス : } RQパケットの値をそのまま設定する。

着信アドレス : } 5.3.7参照

送受信データ : CID=5の場合のみ存在する。

## 5.3.9 ACMA 交信要求 RS/RE パケット

0	7E	PC	RC	RN
4		UC		DL=16
8		ER		0
12			0	
16	発信アドレス	着信アドレス		
20	0	0		
24	0		0	
28				

PC : byte X'04' RSパケット

X'10' REパケット

RC : 5.3.7参照

UC : CCRR/RS フォーマット

0 12345678 15



INF : 0 正常

- 1 ACMAでリジェクト
- 2 マイクロディスクアクセスエラー
- 3 受信データオーバー
- 4 CAMACアクセスエラー

MNO : メッセージ番号

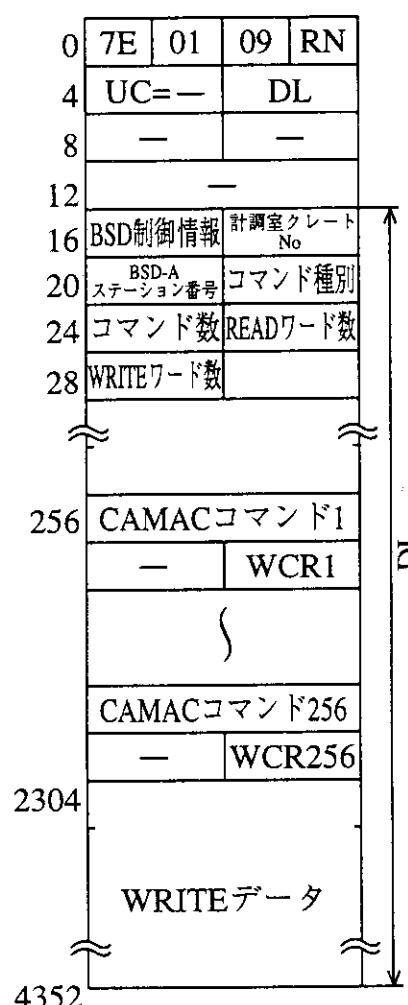
発信アドレス : RQパケットの値をそのまま設定する。

着信アドレス : 5.3.7参照

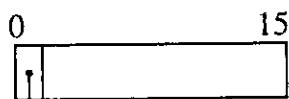
ER : X'FFFF' ISP内で応用プログラムタイムアウトが発生。  
この場合UCのCIDはEOTになっていない。

0 上記以外

## 5.3.1.0 本体室アクセス要求 RQ パケット



## BSD制御情報：



転送モード 0: 16ビット  
1: 24ビット

計調室クレートNo : short

BSD-Aステーション番号 : short

コマンド種別 : short BSD-Aコマンド種別

0 : シングルアクション	2 : ブロック転送UCW
1 : ブロック転送UCS	3 : ブロック転送UQC

コマンド数 : short CICUが実行すべきコマンド数

コマンド種別がブロック転送の場合 1  
シングルアクションの場合 1~256

READワード数 : short

コマンドの中で指定されるREADデータの総数

転送モードが16ビットの場合 0~1024  
24ビットの場合 0~512

WRITEワード数 : short

コマンドの中で指定されるWRITEデータの総数

転送モードが16ビットの場合 0~1024  
24ビットの場合 0~512

CAMACコマンドi : 5.3.3参照

WCRI : unsigned short

コマンド種別がブロック転送の場合は

BSD-Aの最大コマンドくり返し回数。

転送モードが16ビットの場合 0~1024

24ビットの場合 0~512

コマンド種別がシングルアクションの場合は不定

WRITEデータ : WRITEしないコマンドのためのスペースは確保されない。

0	15	16	31
データ0	データ1		
データ2	データ3		

転送モードが  
16ビットの場合

0	7	8	24
—	データ0		
—	データ1		

転送モードが  
24ビットの場合

## 5.3.1.1 本体室アクセス要求 RS パケット

0	7E	04	09	RN	
4	UC=—	DL			
8	ER	—			
12 —					
16	BSD制御情報	計調室クレート No			
20	BSD-A ステーション番号	コマンド種別			
24	コマンド数	READワード数			
28	WRITEワード数	—			
32	エラー発生種別	CSR			
36	ESTR	—			
400 DL —					
256	CAMACコマンド1				
	ステータス1	WCR1			
		}			
	CAMACコマンド256				
	ステータス256	WCR256			
2304	READデータ				
4352 DL —					

ER : short CICU復帰情報

X'0000' 正常

X'0010' BSDエラー

X'0020' BSD-Aでエラー発生

X'0050' パラメタエラー

BSD制御情報 :

計調室クレートNo :

BSD-Aステーション番号 : } RQパケットの値をそのまま設定する。

コマンド種別 : } 5.3.10参照

コマンド数 :

READワード数 :

WRITEワード数 :

エラー発生種別



CSR : BSDのステータス情報 5.3.3参照

ESTR : BSD-Aの ク 5.3.6参照

CAMACコマンドi : 5.3.3参照

ステータスi : 5.3.6参照

WCRI : unsigned short コマンド種別がブロック転送の場合は実行できなかったコマンド数

コマンド種別がシングルアクションの場合は0

READデータ : READしないコマンドのためのスペースは確保されない。

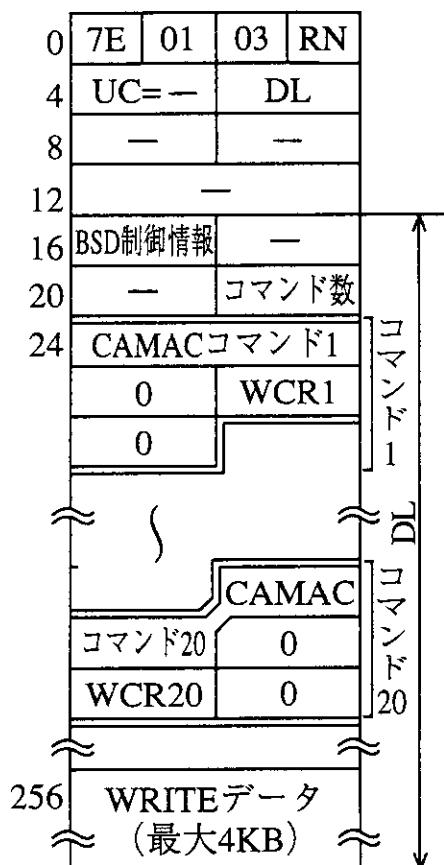
転送モードが16ビットの場合

転送モードが24ビットの場合

0	15	16	31
データ0	データ1		
データ2	データ3		
:	:		

0	78	24
0	データ0	
0	データ1	

## 5.3.1.2 計調室アクセス要求 RQ パケット



## BSD制御情報

0	15
---	----

転送モード 0:16ビット

1:24ビット

コマンド数: short

Mがシングルアクションの場合 1~20

Mがブロック転送の場合 1

CAMACコマンドi: 5.3.3参照

WCRi: unsigned short

Mがブロック転送の場合

転送モードが16ビットの場合 1~2048

" 24ビットの場合 1~1024

Mがシングルアクションの場合 不定

WRITEデータ:

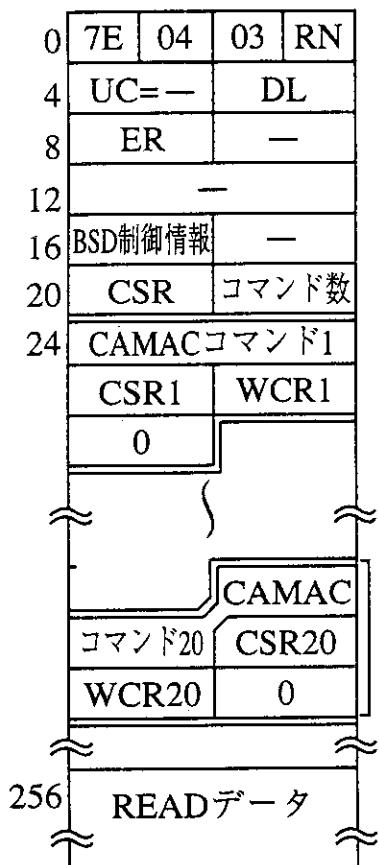
WRITEしないコマンドに対してはスペースが確保される。

0	15	16	31
データ0	データ1		転送モードが
データ2	データ3		16ビットの場合
!	!		

0	78	31
—	データ0	
—	データ1	
—	—	

転送モードが  
24ビットの場合

## 5.3.1.3 計調室アクセス要求 RS パケット



ER : short CICU復帰情報

X'0000' 正常

X'0010' BSDエラー

X'0050' パラメタエラー

BSD制御情報 : } RQパケットの値をそのまま設定する。  
 コマンド数 : } 5.3.12参照

CSR : ERがBSDエラーの場合、BSDのステータス  
 情報が設定される。5.3.3参照

CAMACコマンドi : 5.3.3参照

CSRi : BSDステータス情報。5.3.3参照

WCRi : unsigned short

Mがブロック転送の場合は実行できなかったコマンド数

Mがシングルアクションの場合は0

READデータ :

READしないコマンドに対してはスペースが確保される。

0	15	16	31
データ0		データ1	
データ2		データ3	
:		:	

転送モードが  
16ビットの場合

0	78	31
0	データ0	
0	データ1	
	:	

転送モードが  
24ビットの場合

## 5.3.1.4 オンライン要求 RQ パケット

0	7E	01	08	RN
4	UC=—	—	DL=48	
8	—	—		
12	—	—		
16	クレート数	クレートアドレス1		
20	クレートアドレス2	4	3	
24	—	—	—	
28	—	—	—	
32	—	—	—	
36	—	—	—	
40	—	—	—	
44	—	—	—	
48	—	—	—	
52	—	—	—	
56	—	—	—	
60	—	—	—	

クレート数 : short 初期化するクレート数 1~16

クレートアドレスi : short 初期化するクレートアドレス 1~62

## 5.3.1.5 オンライン要求 RS パケット

0	7E	04	08	RN
4	UC=—	—	DL=10N+2	
8	ER	—	0	
12	—	—	0	
16	異常リプライ数 N	CAMAC		
18	コマンド1	CSR1		
22	0	0		
26	CAMACコマンド2			
30	CSR2	0		
34	0			
38	—	—	—	
42	—	—	—	
46	—	—	—	
50	—	—	—	
54	—	—	—	
58	—	—	—	
62	—	—	—	

ER :

異常リプライ数 :

CAMACコマンドi :

CSRi :

5.3.3参照

異常

リプライ

1

異常

リプライ

2

異常

リプライ

N

5.3.1.6 ループ番号及び方向標識

-7E	80	80	00
—	—	0	—
—	—	—	—
LPNO	DIR		

LPNO : short ループ番号 0, 1, 2  
DIR : short 方向標識 1…ISP起動  
2…CICU起動

## 6. ACMAとの通信

### 6.1 通信の概要

本通信は CAMAC シリアルハイウェイ上に接続されたクレートに存在する富士通製マイクロコンピュータ付補助クレートコントローラタイプ A (ACMA) と CICU との間でデータを伝送するものである。通信は半2重コンテンション方式で、制御のやりとりは ACMA 内部の CCRR と CCRS レジスタで、データのやりとりは富士通製 LAM エンコーダ (LAME) 内の 2 KW (4 KB) のバッファを介して行う。この様子を図 9 に示す。

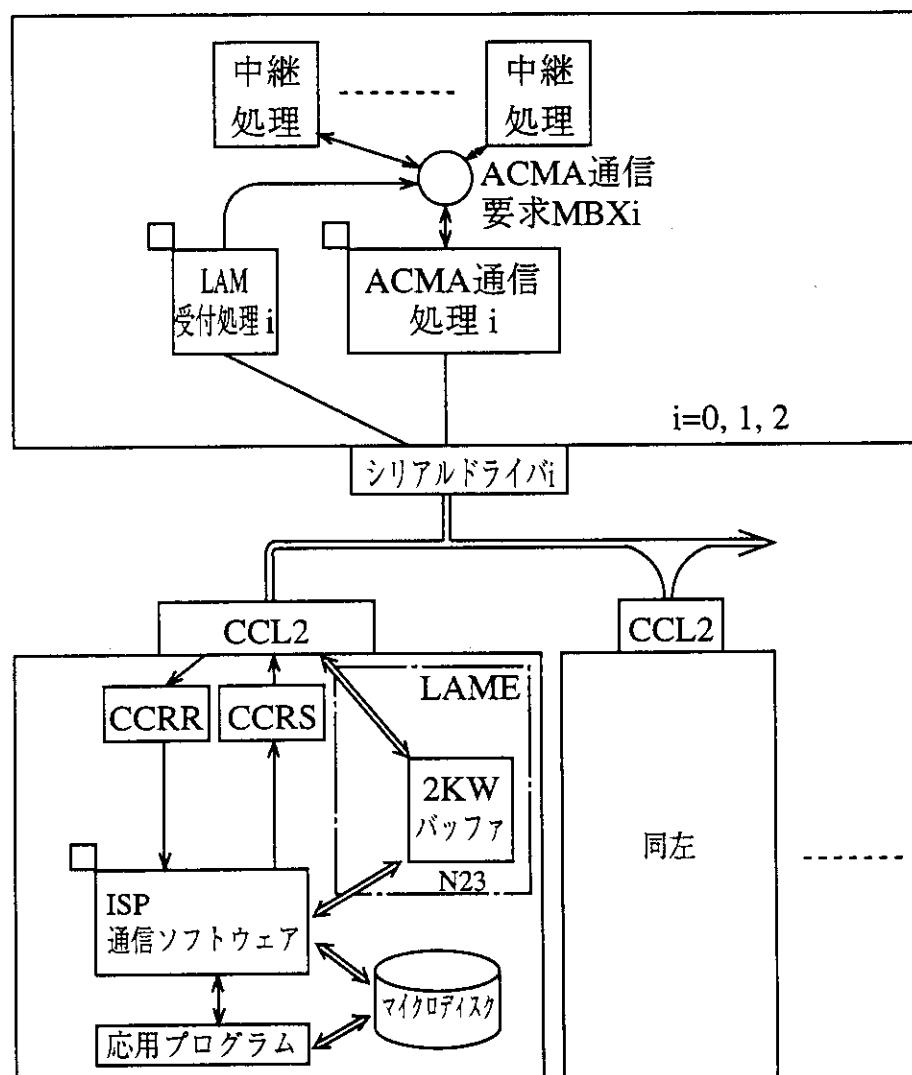


図 9 ACMA との通信の概要

## 6.2 使用ハードウェアとCAMACコマンド

CCL2 KS3952

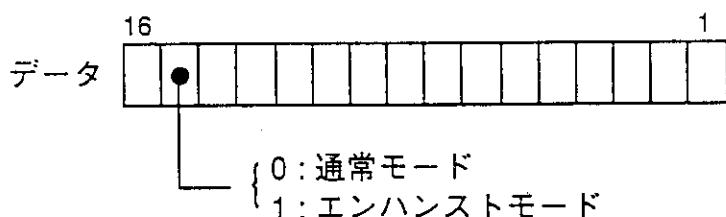
クレートアドレス C(1) ~ C(62)

N(30)A(0)F(23)W(800h) アンバイパス。このコマンドはハイウェイを物理的に切り替えるので、コマンドはエラーとなる。次のコマンドを実行するまで 300 mS 待つ。また次のコマンドが遅延エラーとなることもある。

N(30)A(0)F(17)W(100h) オンライン、デマンド許可、I 停止

エンハンストモードの判定

N(30)A(0)F(1) STR リード



LAME F7822CA

ステーションアドレス N(23)

N(23)A(1)F(24)	デマンド発生不許可	LAME 初期化
N(23)A(1)F(25)	ハングデマンドフラグクリア	
N(23)A(13)F(11)	LER クリア	
N(23)A(1)F(16)W(0)	ADR クリア	

N(23)A(13)F(20)W(14h) LER N20 デマンド許可

N(23)A(1)F(26) デマンド発生許可

N(23)A(0)F(0)	パッファ読み込み & ADR インクリメント	パッファの読み書き
N(23)A(0)F(16)	パッファ書き込み & ADR インクリメント	
N(23)A(1)F(16)	ADR 設定	

ACM-A F7821AA

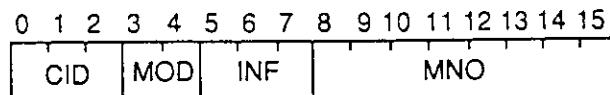
ステーションアドレス N(20) [右スロット]

N(20)A(1)F(24)	PSR による LAM 禁止	CCRS 割り込み許可
N(20)A(0)F(26)	CCRS による LAM 許可	
N(20)A(0)F(0)	CCRS 読み込み Q=1 OK, Q=0 テータ無	
N(20)A(0)F(16)	CCRR 書き込み Q=1 OK, Q=0 前のデータが残存	

## 6.3 通信プロトコル

### 6.3.1 CCRR/RS 形式

CCRR/RS は 16 ビットのレジスタで下記の形式を持つ。



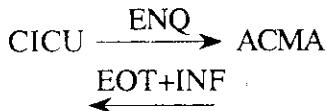
CID : 1 ENQ	MOD : CID=1 のとき	INF : CID=3 又は 4 のとき
2 ACK	0 : データ無	0 : 正常
3 NAK	1 : ISP → ACMA	1 : ACMA 側でリンクエラー
4 EOT	2 : ACMA → ISP	2 : マイクロ・アクセスエラー
5 ETX	CID ≠ 1 のときは 0	3 : 受信データオーバー
		4 : CAMAC アクセスエラー

MNO : 一覧を下表に示す。

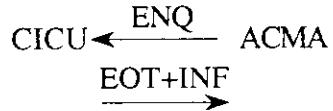
メッセージ	MNO	MOD	方向	メッセージ	MNO	MOD	方向
放電要請	10	0	↓	高速較正データ収集要求	9C	0	↑
放電条件設定	11	1	↓	高速較正データ収集要求OK	0C	2	↓
放電条件設定完了	91	2	↑	高速較正データ収集要求NG	1C	0	↓
計測準備完了	92	2	↑	オンライン要求	85	0	↑
放電完了	13	0	↓	オンライン要求OK	06	0	↓
計測完了	93	2	↑	オンライン要求NG	05	0	↓
高速データ収集完了	15	0	↓	オフライン要求	87	0	↑
計測条件解除	1E	0	↓	オフライン要求OK	08	0	↓
計測中止	1F	0	↓	オフライン要求NG	07	0	↓

### 6.3.2 シーケンス

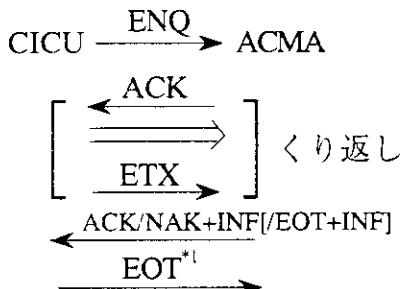
ISP 起動 データ無 (MOD=0)



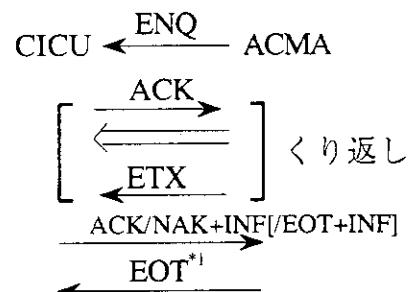
ACMA 起動 データ無 (MOD=0)



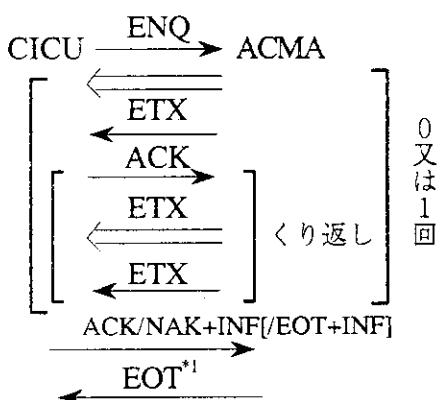
ISP 起動 ISP → ACMA データ有  
(MOD=1)



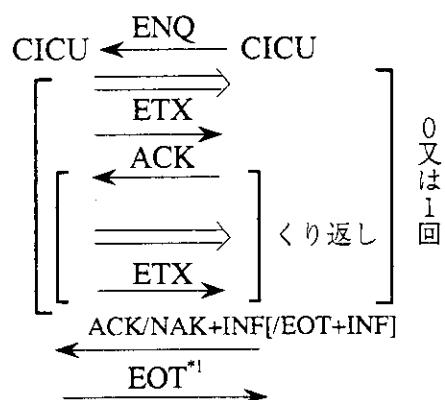
ACMA 起動 ACMA → ISP データ有  
(MOD=2)



ISP 起動 ACMA → ISP データ有  
(MOD=2)



ACMA 起動 ISP → ACMA データ有  
(MOD=1)



今回はこのシーケンスは存在しない。

\* 1 EOT をすでに受信している場合は送信しない。

### 6.3.3 回線管理

- ・ ENQ - EOT までは排他的に回線を使用する。
- ・ ENQ コンテンションウィナーは CICU である。その場合 ACMA は一定時間後 ENQ を再送する。CICU はただちに ENQ を再送する。
- ・ データ有の ENQ メッセージは ACMA 内で他の ENQ と衝突することがあり得る。その場合 INF=1 の EOT メッセージが CICU に戻されるが、直後に ACMA は同一の ENQ メッセージを CICU に送信し、通常の手順に従って通信を継続する。衝突した ENQ は一定時間後に再送される。
- ・ データ送受信の物理的な単位は最大 2 KW (4 KB) である。

### 6.3.4 通信の開始

- ① CICU は CCL2 アンバイパス、オンライン、ACMA のデマンド発生許可とする。
- ② CCRS を空読みする。
- ③ CICU は ACMA からの割り込みを待つ。CCRS を読む。もし 0 なら ③ に戻る。  
0 以外ならメッセージとみなして通信処理を行う。

☆ 電源断の状態から先に CICU が稼動した場合

- ① CCL2 アンバイパス、オンライン等に失敗する。
- ② ACMA の電源 ON、IPL を行っても CICU に通知できないので、
- ③ ACMA コマンドによりオンライン要求を出す。その結果 CCL2 アンバイパス、オンライン、デマンド発生許可、CCRS 空読みが行われる。

☆ 電源入の状態から CICU が稼動した場合、又は運用中に ACMA を IPL した場合

- ① CCL2 アンバイパス、オンライン等は正常である。もし ACMA に何か送信しようとした場合はタイムアウトエラーとなる。
- ② ACMA の IPL 後 CCRS に 0 が書き込まれる。CICU はこれを読み捨てる。ただし IPL で Z 信号が出る場合はデマンド禁止となってしまうため ACMA コマンドの投入が必要となる。

☆ CICU が後から稼動した場合

- ① CICU は CCL2 アンバイパス、オンライン、デマンド発生許可、CCRS の空読みを行う。

### 6.3.5 データ部のフォーマット

データ部のフォーマットは CICU では関知しない。

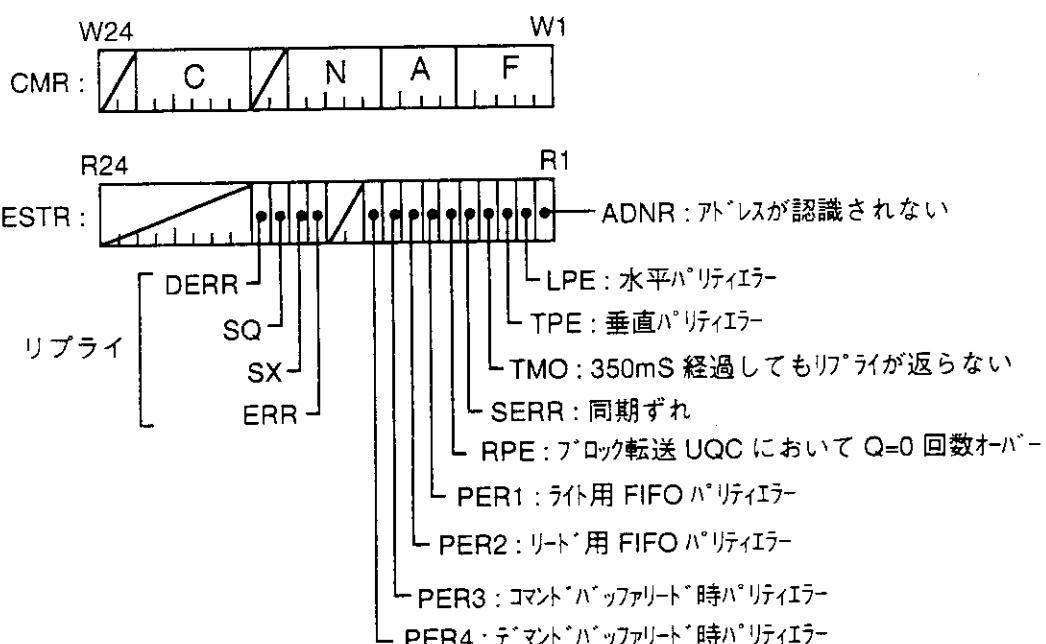
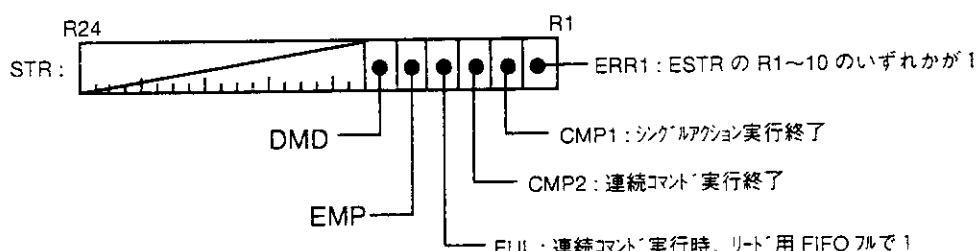
注) LAME のアドレスポインタは送信後必ず 0 に戻すこと。ACMA の ISP 通信  
ソフトはアドレスポインタを操作しない。

## 7. ハードウェア操作

### 7.1 BSD-A シングルアクションの実行

BSD-A はシングルアクション用のコマンドレジスタ (CMR) を持っていて、ここに CNAF を設定することによりコマンドを実行する。データはシングルライトデータレジスタ (SWDR)、シングルリードデータレジスタ (SRDR) を使用する。BSD-A のステーションアドレスは各処理単位に ISP より指定される。

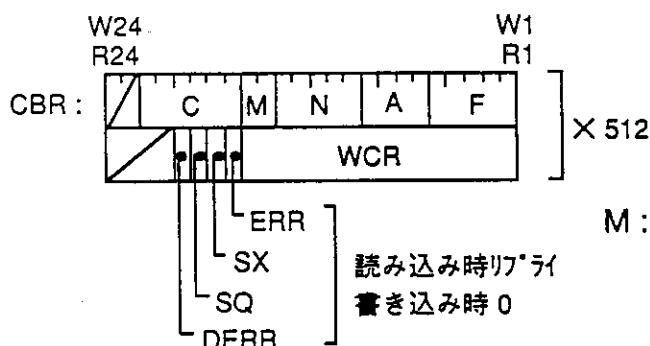
F(1)A(13)	リード MKR	マスクレジスタの内容を読む
F(17)A(13)W(0)	ライト MKR	割り込みをすべてマスクする
F(17)A(3)W(データ)	ライト SWDR	ライトファンクション時はデータを書き込む
F(17)A(0)W(CNAF)	ライト CMR	コマンド実行
⌚ F(1)A(12)	リード STR	CMP1=1 を待つ
F(23)A(12)W(3)	セレクティブクリア STR	CMP1, ERR ← 0 とする
F(1)A(11)	リード ESTR	結果の確認
F(1)A(4)	リード SRDR	リードファンクション時はデータを読み込む
F(17)A(13)W( )	ライト MKR	マスクレジスタの値を復旧する



## 7.2 BSD-A 連続コマンドの実行

BSD-A のコマンドバッファ (CBR) にあらかじめ複数のコマンドを設定し、BSD-A を起動することにより、連続的にコマンドを実行させることができる。コマンドはシングルアクションでも、ブロック転送でもかまわないが、今回のCICU では両方の種類のコマンドを混在して実行することはない。コマンドの実行に必要なライトデータはライト用 FIFO に、リードデータは実行後にリード用 FIFO から読み込む。データ数は FIFO の容量の範囲内とするので、実行中に書き込み、又は読み込みを行うことはない。

F(1)A(13)	リード MKR	マスクレジスタの内容を読む
F(17)A(13)W(0)	ライト MKR	割り込み禁止
F(17)A(2)W(0)	ライト CTR	A モードとする
F(9)A(0)	クリア CBR アドレス	
F(9)A(3)	クリア WDR アドレス	各 FIFO 初期化
F(9)A(4)	クリア RDR アドレス	
⌚ F(16)A(3)	ライト WDR	ライトファンクションがある時はデータを書き込む
⌚ F(16)A(0)	ライト CBR	コマンド列書き込み
F(25)A(2)	起動	
⌚ F(1)A(12)	リード STR	CMP2=1 を待つ
F(23)A(12)W(0x1d)	セレクティブクリア STR	EMP, FUL, CMP2, ERR をクリア
(F(9)A(2)	強制終結	ソフトウェアタイムアウトの時は強制終了させる)
F(1)A(11)	リード ESTR	結果の確認
⌚ F(0)A(4)	リード RDR	リードファンクションがある時はデータを読み込む
F(9)A(0)	クリア CBR アドレス	
⌚ F(0)A(0)	リード CBR	コマンド毎の結果の確認
F(17)A(13)W( )	ライト MKR	マスクレジスタの値を復旧する



- |        |            |
|--------|------------|
| M : 00 | シングルアクション  |
| 01     | ブロック転送 UCS |
| 10     | ブロック転送 UCW |
| 11     | ブロック転送 UQC |

WCR : ブロック転送時のみ有効

書き込み時 : くり返し回数。0 ~ 65535。0 は 65536 回とみなす。

読み込み時 : 未実行回数

WDR, RDR : 24 ビット × 4096

### 7.3 BSD-A 高速データ収集の実行

BSD-A のコマンドバッファ (CBR) にあらかじめ複数のコマンドを設定し、BSD-A を起動することにより、連続的にリードファンクションを実行させる。読み込んだデータはリード用 FIFO に入ってくるので、これを BSD に連続的に読み込む。BSD-A の動作が早くリード用 FIFO がフルになると、空ができるまで BSD-A の読み込みが停止する。BSD の動作が早くリード用 FIFO が空になると Q=0 が返り、BSD は Q=1 になるまでコマンドを再試行する。(ブロック転送 UQC)

F(1)A(13)	リード MKR	マスクレジスタの内容を読む
F(17)A(13)W(0)	ライト MKR	割り込み禁止
F(17)A(2)	ライト CTR	BSD-A モード設定
F(17)A(10)W(0)	ライト WBR	ウェイトパットを 0 にする
F(9)A(0)	クリア CBR アドレス	各 FIFO 初期化
F(9)A(4)	クリア RDR アドレス	
○ F(16)A(0)	ライト CBR	コマンド列書き込み
F(25)A(2)	起動	
○ F(0)A(4)	リード RDR	Q=1 で有効
F(1)A(12)	リード STR	CMP2=1 を確認
(F(9)A(2)	強制終結	終結していなければ強制終結)
F(23)A(12)W(0x1d)	セレクティブクリア STR	EMP, FUL, CMP2, ERR をクリア
F(1)A(11)	リード ESTR	
F(9)A(0)	クリア CBR アドレス	
○ F(0)A(0)	リード CBR	コマンド毎の結果の確認
F(17)A(13)W( )	ライト MKR	マスクレジスタの値を復旧する
W24		
CTR : 		

起動前に ESTR のエラービット、STR の ERR ビットが 1 となっていると正常に動作しない。これを防止するには起動前に

F(1)A(11)	リード ESTR
F(11)A(12)	クリア STR

を実行すればよい。

## 謝　　辞

新C I C Uの開発に際し、ご協力頂いた渡辺 和彦氏（現在(株)原子力資料サービス）をはじめとする炉心プラズマ解析室A-7関係者の方々に感謝致します。また、本開発を進めるにあたり暖かい御支援を頂いた、菊池 満炉心プラズマ解析室長、平山 俊雄プラズマ理論研究室長、安積 正史炉心プラズマ部長に感謝致します。

## 参考文献

- 1) 山本 需 他：FUJITSU **37**, 49 (1986).
- 2) 長島 章 他：核融合研究 **59** 別冊, 303 (1988).
- 3) 青柳 哲雄：プラズマ・核融合学会誌 **72**, 1370 (1996).
- 4) Y. Takeuchi, et al., KEK preprint 92-80, KEK, TITHEP-92-05, TITECH (1992).

## 謝 辞

新CICUの開発に際し、ご協力頂いた渡辺 和彦氏（現在(株)原子力資料サービス）をはじめとする炉心プラズマ解析室A-7関係者の方々に感謝致します。また、本開発を進めるにあたり暖かい御支援を頂いた、菊池 満炉心プラズマ解析室長、平山 俊雄 プラズマ理論研究室長、安積 正史炉心プラズマ部長に感謝致します。

## 参 考 文 献

- 1) 山本 需 他：FUJITSU **37**, 49 (1986).
- 2) 長島 章 他：核融合研究 **59** 別冊, 303 (1988).
- 3) 青柳 哲雄：プラズマ・核融合学会誌 **72**, 1370 (1996).
- 4) Y. Takeuchi, et al., KEK preprint 92-80, KEK, TITHEP-92-05, TITECH (1992).