

JAERI-Tech  
2000-043

JP0050679



# 新データ処理設備実時間計算機(RTP) の開発、および高性能化

2000年7月

坂田 信也・小岩 素直・青柳 哲雄\*・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。

入手の問合わせは、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）で複写による実費頒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, 2000

---

編集兼発行 日本原子力研究所

## 新データ処理設備実時間処理計算機（RTP）の開発、および高性能化

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

（2000年5月16日受理）

臨界プラズマ試験装置（JT-60）立ち上げ当初、データ処理設備実時間処理計算機（RTP）としてPANAFACOM U-1500ミニコンピュータを中心としたシステムが構築されていた。しかし、システムの老朽化に伴い、ハードウェア、ソフトウェア共に保守することが困難な状態になりつつあった。

これに対し、近年のUNIX系ワークステーションの性能向上は著しく、ユーザアプリケーションプログラムに対する柔軟性、ハードウェアのメンテナンスの容易さ、および周辺装置の拡張性を考慮し、UNIX系ワークステーションを中心とした新システムを構築した。

本報告書では、新データ処理設備実時間処理計算機（RTP）の概要、基本設計、および近年の高性能化改造について述べる。

Development and Upgrade of New Real Time Processor  
in JT-60 Data Processing System

Shinya SAKATA,Motonao KOIWA,Tetsuo AOYAGI\* and Toshiaki MATSUDA

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

(Received May 16 , 2000)

At the beginning of JT-60 experiments, the real time processor (RTP) in the data processing system was mainly constructed by PANAFACOM U-1500. As the computer became superannuated, however, it gradually became difficult to maintain both hardware and software.

A performance of a recent UNIX workstation has been remarkably progressed. The UNIX workstation has a large flexibility for user application programs, an easiness for maintenance of the hardware and an ability of expansion to peripheral devices. Therefore, the RTP system is newly reconstructed by using the UNIX workstation.

This report describes the overview, the basic design and the recent upgrade on the RTP in the data processing system.

Keywords: JT-60, Data Processing System, Real Time Processor, UNIX Workstation  
Real Time Process

---

\* Research Organization for Information Science & Technology

## 目 次

1. 概要 .....	1
2. 実時間処理の実現 .....	2
2. 1 マルチプロセス制御 .....	2
2. 2 マルチプロセス制御に於ける問題点 .....	3
2. 3 オペレーティングシステムの特徴 .....	4
2. 3. 1 プロセス割り当て機能 .....	5
2. 3. 2 割り込み制限機能 .....	5
2. 3. 3 固定優先度割り当て機能 .....	6
3. RTPの構成機器概要 .....	6
4. 実時間処理システム構成 .....	8
4. 1 ISPとの接続 .....	8
4. 2 計測装置との接続 .....	8
4. 2. 1 CAMAC入出力装置 .....	8
4. 2. 2 K2140用デバイスドライバの開発 .....	10
4. 2. 3 Dポート付デュアルポートメモリ (DPMD) .....	10
4. 2. 4 A/D変換器 .....	11
4. 3 データ処理設備タイミングサブシステムとの接続 .....	12
4. 4 全系との接続 .....	12
5. RTPの処理構成 .....	13
5. 1 オンラインプロセス .....	13
5. 1. 1 オンラインプロセスの起動／停止 .....	13
5. 1. 2 放電制御メッセージの送受信処理 .....	15
5. 1. 3 CAMAC初期設定処理 .....	20
5. 1. 4 リフレクティブメモリ初期設定処理 .....	26
5. 2 実時間処理プロセス .....	27
5. 2. 1 実時間処理プロセスの生成／消滅 .....	27
5. 2. 2 実時間処理環境の設定 .....	27
5. 2. 3 タイミング信号の検出 .....	29
5. 2. 4 実時間処理 .....	32
5. 2. 5 実時間帰還制御用機器健全性チェック .....	38
5. 2. 6 A/D変換器ベース電圧の測定 .....	40
5. 3 プロセス間、およびCPU間通信 .....	40
5. 4 ロギング機能 .....	41
5. 5 RTP不具合発生時の回避処理 .....	41
5. 6 リフレクティブメモリが持つ懸念事項 .....	41
6. RTP改造推移 .....	42

7. 改造結果 .....	4 5
8. まとめ .....	4 7
謝辞 .....	4 8
参考文献 .....	4 8

## Contents

1.	Overview .....	1
2.	Realization of Real Time Processing .....	2
2. 1	Multi Process Control .....	2
2. 2	Problem in Multi Process Control .....	3
2. 3	Characteristics of Operating System .....	4
2. 3. 1	Function to Assign Process .....	5
2. 3. 2	Function to Restrict Interrupt .....	5
2. 3. 3	Function to Assign Specified Priority .....	6
3.	RTP System Overview .....	6
4.	Configuration of Real Time Processing System .....	8
4. 1	Connection with ISP .....	8
4. 2	Connection with Diagnostic Subsystems .....	8
4. 2. 1	CAMAC Interface Unit .....	8
4. 2. 2	Development of Device Driver for K2140 .....	10
4. 2. 3	Dual Port Memory with D-port .....	10
4. 2. 4	A/D Converter .....	11
4. 3	Connection with Timing Subsystem .....	12
4. 4	Connection with ZENKEI .....	12
5.	Process in RTP Processes .....	13
5. 1	Online Process .....	13
5. 1. 1	Start and Stop .....	13
5. 1. 2	Process Sending and Receiving Discharge Control Messages .....	15
5. 1. 3	Initial Setup Process of CAMAC .....	20
5. 1. 4	Initial Setup Process of Reflective Memory .....	26
5. 2	Real Time Process .....	27
5. 2. 1	Creation and Termination .....	27
5. 2. 2	Setup for Real Time Process .....	27
5. 2. 3	Detection of Timing Signal .....	29
5. 2. 4	Real Time Processing .....	32
5. 2. 5	Hardware Check .....	38
5. 2. 6	Measurement of Base Voltage of A/D Converter .....	40
5. 3	Interprocess and Inter-CPU Communication .....	40
5. 4	Logging Function .....	41
5. 5	Function to Recover from Troubles .....	41
5. 6	Remarks About Reflective Memory .....	41
6.	Upgrade of RTP .....	42

7 . Results .....	4 5
8 . Concluding Remarks .....	4 7
Acknowledgements .....	4 8
References .....	4 8

## 1. 概要

J T - 6 0 データ処理設備実時間処理計算機（以下「R T P」と称す。）は、J T - 6 0 実験放電で生成されるプラズマに対し、実時間帰還制御を行うためのパラメータとなる複数の計測データを収集し、収集した各計測データに任意の演算処理を実施した後、全系制御設備（以下「全系」と称す。）に転送するという一連の処理を 1 m s 周期で実施している。この一連の処理を実時間処理<sup>\*1</sup>と呼ぶ。図 1 に実時間帰還制御系における R T P の位置づけを示す。

J T - 6 0 立ち上げ当初、R T P は、PANAFACOM U-1500 ミニコンピュータを中心としたシステムで構築されていたが<sup>1)、2)</sup>、システムの老朽化に伴い、ハードウェア、ソフトウェアの両面から保守することが困難な状態になりつつあったため、U N I X 系ワークステーションを中心とした新システムの構築を検討した。

しかし、実時間処理実行中においても、J T - 6 0 実験放電シーケンスと同期を図るために、R T P の上位計算機となるデータ処理設備ショット間処理計算機とのネットワークを介したメッセージの送受信処理を行わなければならず、通常の U N I X 系ワークステーションを導入した場合、マルチプロセスにて生じる割り込み処理が問題となることが懸念された。そこで R T P では、問題となる割り込み処理の発生をリアルタイム処理に対し特徴的な機能を有するオペレーティングシステムと 2 C P U を有する計算機を採用することで解決し、システムの再構築を実現した。

また、近年、実時間帰還制御の多様化に伴い、パラメータとなる計測データの増加、演算処理の高度化と同時に実時間処理のさらなる高速化が求められてきた<sup>3)、4)、5)、6)、7)、8)、9)</sup>。この要求に応えるため R T P は、幾つかの段階を経て実時間処理の高速化を実現した。

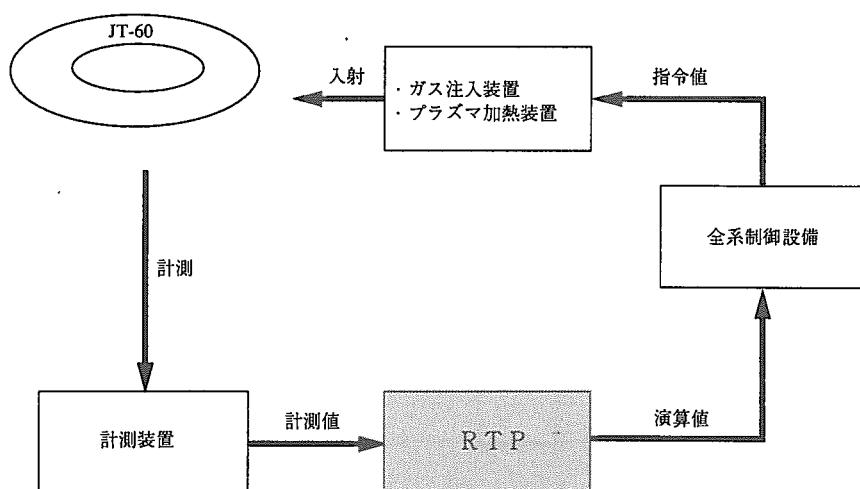


図 1 実時間帰還制御系における R T P の位置づけ

\* 1 : 本報告書内に記述する①実時間処理プロセス、および②実時間処理について以下のように定義する。

①実時間処理プロセス：オンラインプロセスにより起動され、プラズマ放電との同期を図りながら実時間処理を動作させるためのプロセス。

②実時間処理：実時間処理プロセス中の処理の一部であり、I p 励起開始タイミング以降、全系 1 m s クロックに同期し、計測データの収集、演算、格納、および全系に対し実時間帰還制御用パラメータデータの転送という一連の処理。

## 2. 実時間処理の実現

RTPは、JT-60実験放電で生成されるプラズマに対し、実時間系帰還制御を行うためのパラメータとなる複数の計測データを収集し、収集した各計測データに任意の演算処理を実施した後、全系に転送するという一連の処理を1ms周期で実施している。この一連の処理を実時間処理と呼ぶ。

一方、実時間処理実行中においても、JT-60実験放電シーケンス（以下「実験放電シーケンス」と称す。）と同期を図るため、RTPの上位計算機となるデータ処理設備ショット間処理計算機（以下「ISP」と称す。）とのネットワークを介したメッセージの送受信処理を行わなければならない。ここで問題となるのがマルチプロセス制御で生じる割り込み処理の発生である。RTPでは、問題となる割り込み処理の発生を以下のように解決し、JT-60実験放電に於ける実時間処理を実現した。

### 2. 1 マルチプロセス制御

通常、UNIXマシンを利用する場合、利用者はls、pwd等のユーザコマンドを実行する際に動作中の他のプロセスの有無を意識せずに使用することが可能である。また、ユーザが作成したアプリケーションプログラムを実行させる場合についても同じことが言える。

UNIXシステムは、プロセスをCPUで実行させるために各プロセスに対し公平にスケジューリングを行い、プロセスはCPUを時分割方式で共有する。このため実行中のプロセスは、CPUの割り当て時間が経過すると処理が中断され、別のプロセスが実行されるようなスケジューリングが行われる。また、処理を中断されたプロセスは再度実行待ち状態となり、その後、処理が再開される。このようなプロセスの制御方法をマルチプロセス制御と呼ぶ。図2.1にマルチプロセス制御プロセス実行遷移図を記す。

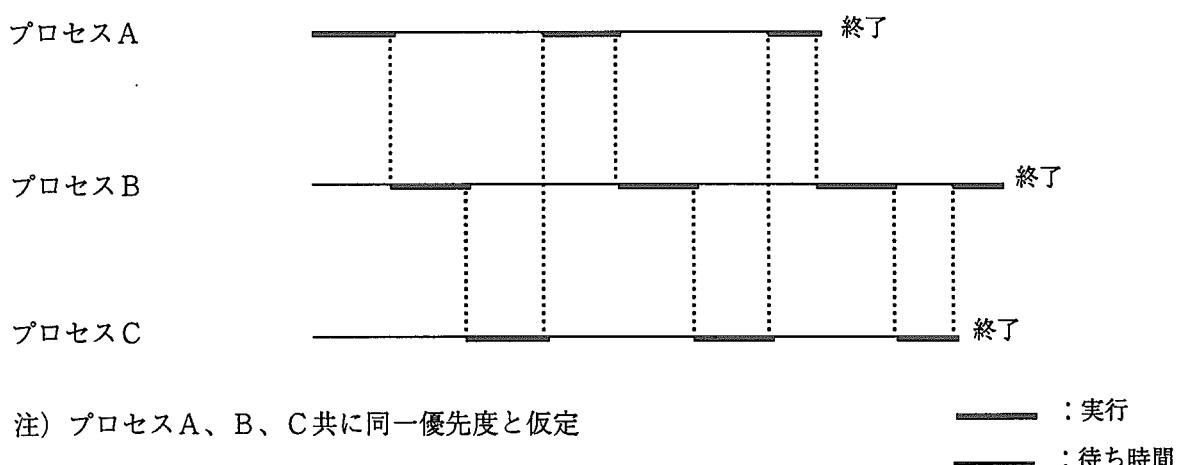


図2.1 マルチプロセス制御プロセス実行遷移図

ここで図2. 1中のプロセスA、B、Cが同一優先度であると仮定した場合、マルチプロセス制御により各々のプロセスに対し、ある一定（単位）時間、処理の実行権が与えられ、それを経過すると再度実行待ち状態とされるよう制御される。

## 2. 2 マルチプロセス制御に於ける問題点

実時間処理プロセスは、プラズマ放電中、全系から転送される1msクロックを検出し、実時間処理の起動タイミングとしている。しかし、実時間処理を実行させる上でマルチプロセス制御は、1msクロックの検出漏れという現象を生じさせる可能性がある。

ここで図2. 1中のプロセスAを実時間処理プロセスと置き換え、さらに実時間処理プロセス、プロセスB、およびCを同一優先度であると仮定した場合、図2. 2に記すように実時間処理プロセスが実行状態にあれば、1msクロックを検出することが可能であるが、待ち状態にあった場合、1msクロックを検出することが出来ない可能性が生じる。さらに1msクロックの検出漏れ、および検出漏れの回数を実時間処理プロセスが認識することは不可能である。こういった現象が発生すると、実時間帰還制御を実施する上で収集したデータの時間的な整合性にズレが生じる。

また、プロセスB、Cに対して、実時間処理プロセスの実行優先度を高くした場合でも、待ち時間がつくられるため、結果的に同じ現象を引き起こしてしまう可能性は残る。

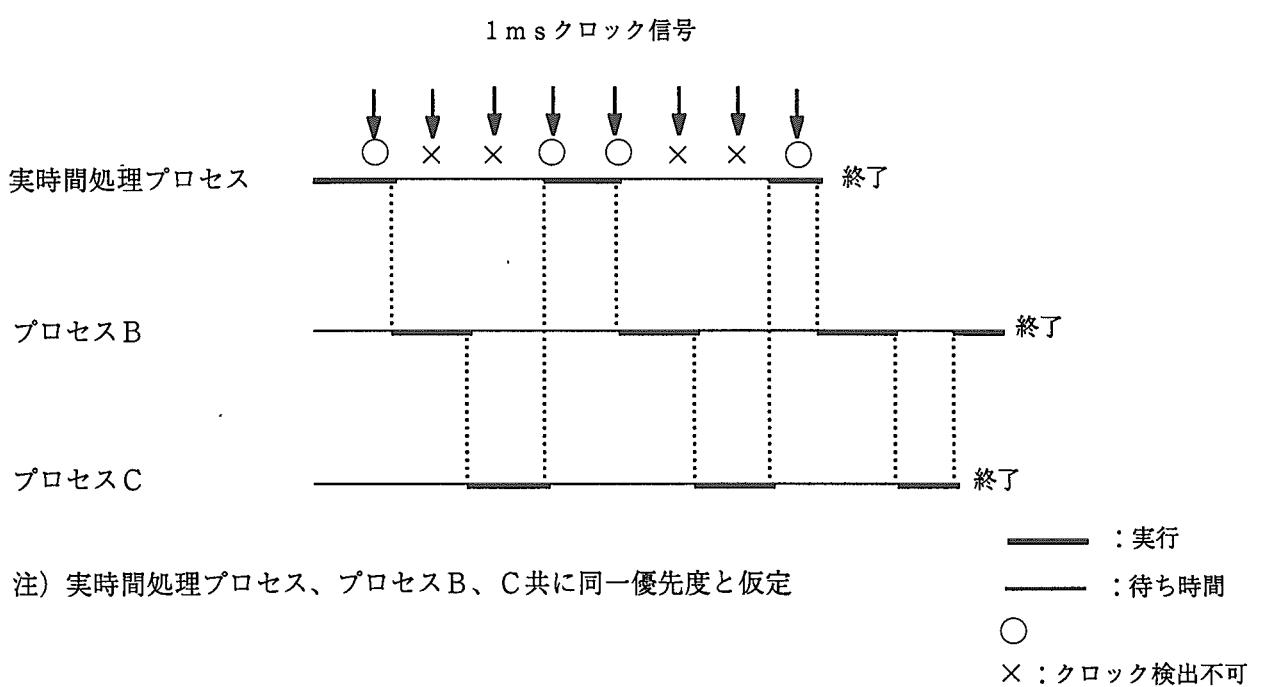


図2. 2 マルチプロセス制御に於ける問題点

## 2. 3 オペレーティングシステムの特徴

RTPで採用しているオペレーティングシステムはリアルタイム拡張並列UNIX<sup>13)</sup>(以下「RTU」と称す。)と呼ばれ、以下に述べる特徴的な機能を有している。

### ①プロセス割り当て機能

任意のプロセスをブートCPU以外の任意のCPU(以下「セカンドCPU」と称す。)に割り当てることが可能。

### ②割り込み制限機能

セカンドCPUに対し、システムからの割り込みを制限することが可能。

### ③固定優先度割り当て機能

セカンドCPU上で動作する任意のプロセスに対し、固定的な優先度を持たせることが可能。

の3点が挙げられる。これらの機能を利用し、セカンドCPUに対し実時間処理プロセスのみを割り当て、セカンドCPUに対してシステムからの割り込みを制限し、さらに実時間処理プロセスの実行優先度を高位にすることで上記2.2のマルチプロセス制御に於ける問題点を解決し、実時間処理を実現させた。図2.3に実時間処理プロセス実行時のCPU制御遷移図を記す。

また、RTUは、UNIXに準拠したOSであるため、TCP/IPプロトコルを用いたネットワーク通信やCPU間、およびプロセス間で共有メモリが使用可能である。以下に各々の特徴について述べる。

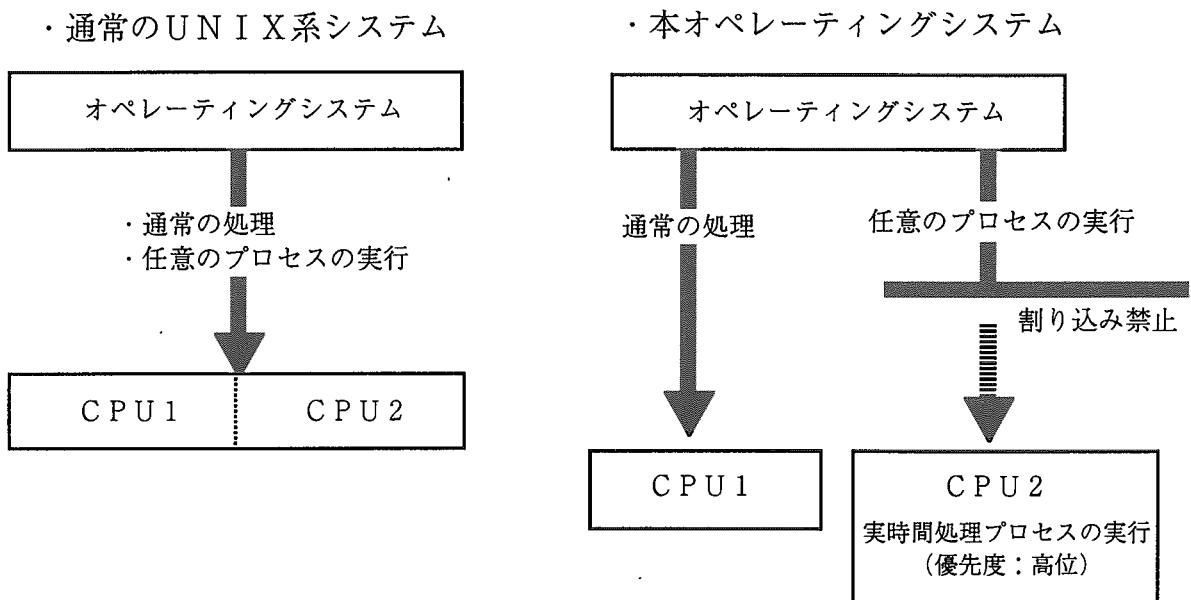


図2.3 CPU制御遷移図

### 2. 3. 1 プロセス割り当て機能

通常、多CPUを有するUNIX系マシン上で、ユーザアプリケーションプロセスを実行した場合、ユーザは、そのプロセスがどのCPU上で実行されているかを認識する必要性がない。また、認識するためのサービスも提供されていない。これに対しRTUでは、ユーザアプリケーションプログラムよりシステムコールを利用して、任意のプロセスをセカンドCPUに割り当てることが可能である。この機能は、以下に述べる2.3.2、2.3.3の機能を併用し、実時間処理を実現するために本システムの基盤となる非常に重要な機能である。

### 2. 3. 2 割り込み制限機能

図2.4に通常のUNIX系マシンにおける処理検出イメージ図を記す。通常のUNIXシステムは、割り込み処理を検出するために図2.4中Aに示すようなシステム固有の処理検出周期を有している。これは、図2.5に示す様にこの処理検出周期が短ければ、処理要求を高速に認識でき、逆に図2.6に示す様に処理検出周期が長ければ処理要求の認識が遅れると言える。しかし、言い方を変えれば、システムに対して処理要求が発生したと言う認識を遅らせることができる。RTUでは、この処理検出周期を60Hz～1000Hzの間で設定可能であり、セカンドCPUに対しては、禁止することも可能である。

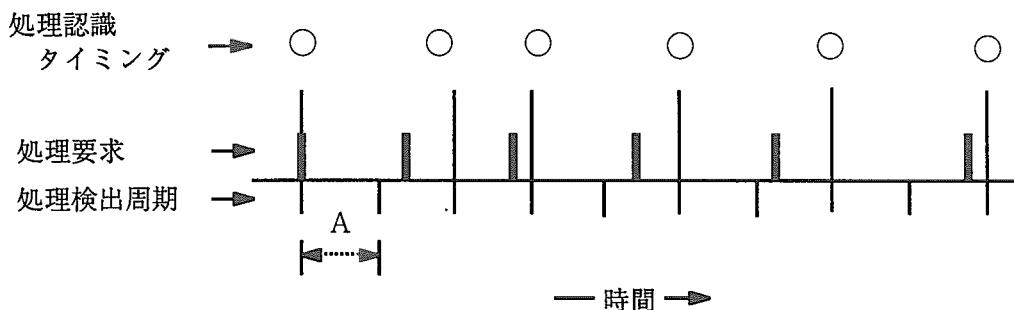


図2.4 処理検出イメージ図（1）

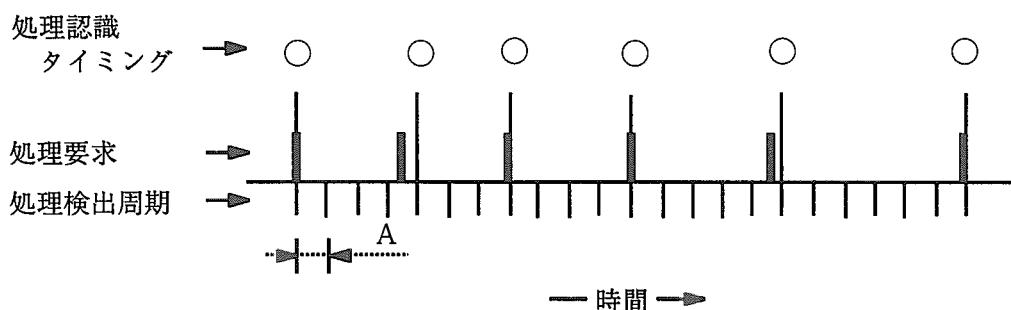


図2.5 処理検出イメージ図（2）

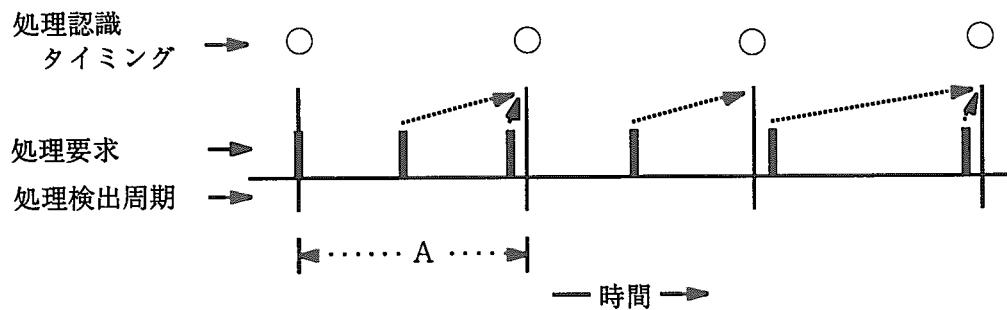


図2.6 処理検出イメージ図（3）

### 2.3.3 固定優先度割り当て機能

近年のUNIX OSでは、リアルタイムスケジューリングクラスを設定することにより、任意のプロセスに対して、実行優先度を調整することが可能となった。しかし、前項の問題点で記したように処理実行中のプロセスに対し待ち時間が生じ、結果的に同じ現象を引き起こしてしまう可能性は残る。これに対しRTUでは、任意のプロセスに対し優先度を高くすることが可能であり、かつ、その優先度より高い優先度を有するプロセスが実行可能な状態にならない限り実行権を放棄しない状態を作り出すことが可能である。

## 3. RTPの構成機器概要

RTPは、処理装置、CAMAC入出力装置、デジタル信号入力装置、A/D変換器、およびリフレクティブメモリにより構成される。図3にRTP外観図を記す。

### ①処理装置

コンカレント System 9200

- |                       |            |    |
|-----------------------|------------|----|
| (1) C P U             | MIPS R4400 | 2個 |
| (2) 主記憶               | 64 MB      |    |
| (3) システムクロック          | 150 MHz    |    |
| (4) 磁気テープ装置           |            |    |
| (5) Ethernet インターフェイス |            |    |
| (6) システムコンソール用キャラクタ端末 |            |    |

### ②CAMAC入出力装置

KINETIC Model 2140-Z1Aシリアルハイワイドライバ

デバイスドライバの開発により、ユーザレベルのアプリケーションプログラムからメモリマップ機能が使用可能

③デジタル信号入力装置

米国PENTLAND社製マルチファンクションボードMPV956  
入力チャンネル数 16CH

④A/D変換器

米国VMIC社製A/D変換ボードVMI VME-3123 2台

- (1)分解能 16 bit
- (2)入力方式 シングルエンド入力/ディファレンシャル入力
- (3)入力レンジ +/-10V、+/-5V
- (4)入力チャンネル数 16CH ( $\times 2$ )
- (5)A/D変換速度  $50\mu s / 16CH$

⑤リフレクティブメモリ

米国VMIC社製A/D変換ボードVMI VME-5576

- (1)メモリ容量 256 Kbyte
- (2)データ伝送方式 光ファイバケーブル

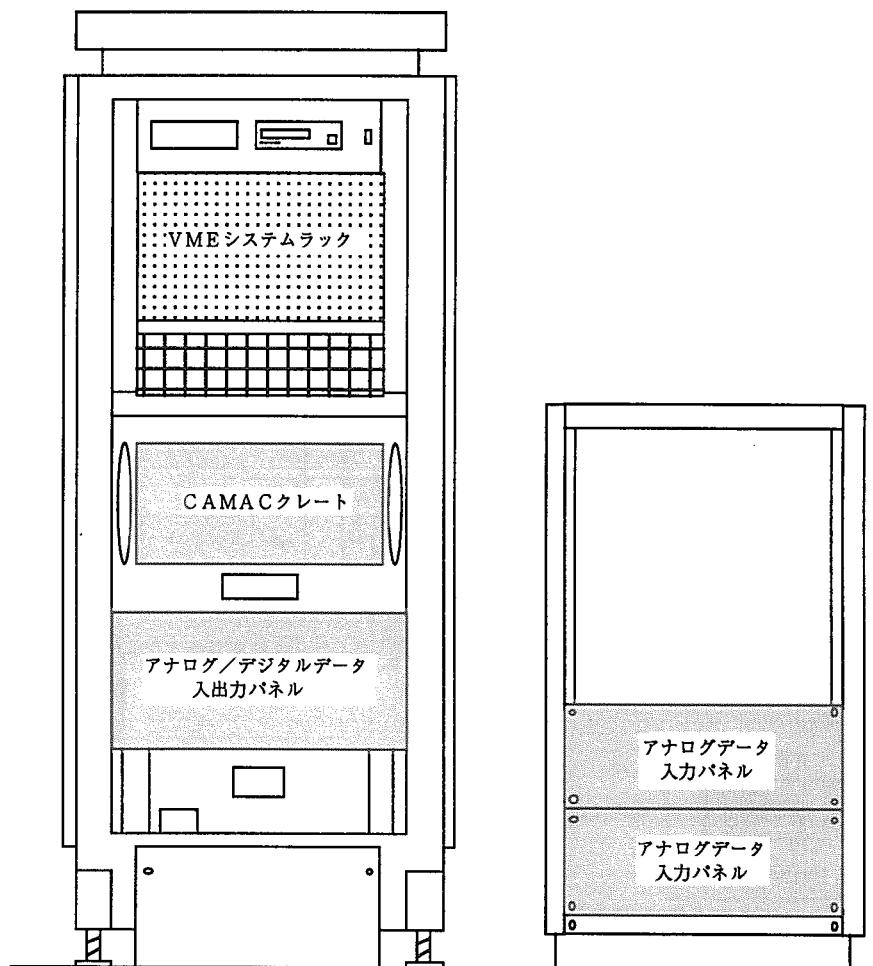


図3 RTP外観図

## 4. 実時間処理システム構成

RTPは実時間帰還制御を実現するため、様々なインターフェイス装置、および入出力装置を介し他システムと接続される。図4. 1に実時間制御系システム構成図を記す。

### 4. 1 ISPとの接続

上位計算機となるISPとはイーサネットを介し接続され、実験放電シーケンスに同期した放電制御メッセージの送受信処理を行う。

### 4. 2 計測装置との接続

実時間処理実行時、計測装置より計測データを収集するために接続される。接続方法は、以下の二通りである。

- ①CAMACシリアルハイウェイを介した接続
- ②A/D変換器を介した接続

#### 4. 2. 1 CAMAC入出力装置

RTPは、VMEバスをバックプレーンとするVMEモジュールで構成されるシステムであるが、実時間処理実行時、計測装置より計測データを収集するためにCAMACシリアルハイウェイを使用する必要があった。このためVME-CAMAC間の入出力装置として、また、CAMACシリアルハイウェイを構築する上での制御装置として、米国KINETIC社製Model 12140-Z1Aシリアルハイウェイドライバ<sup>14)</sup>（以下「K2140」と称す。）を採用した。図4. 2にCAMAC系計測データ収集構成図を記す。

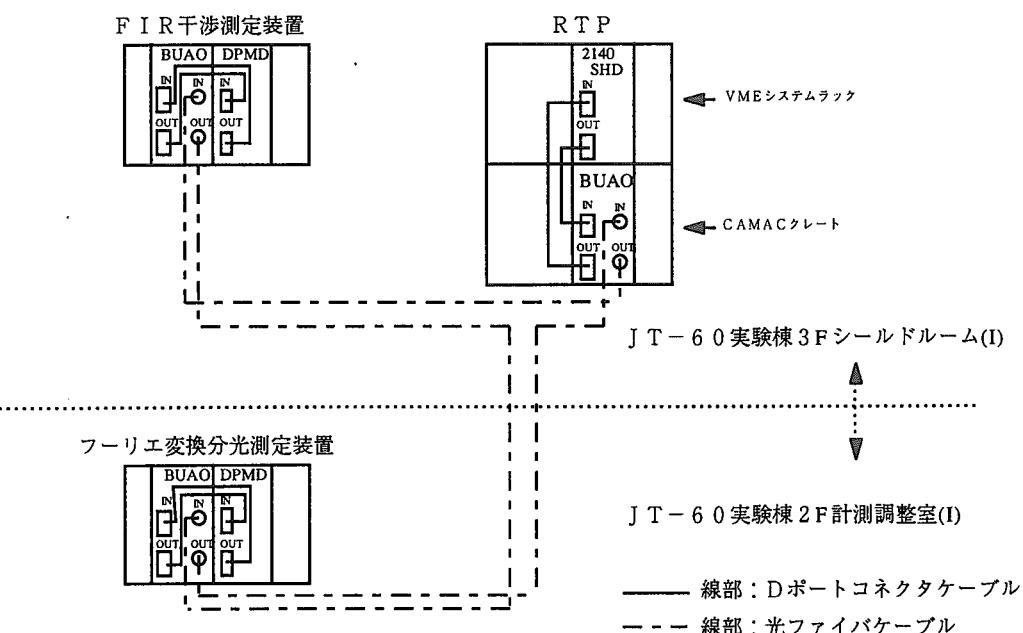


図4. 2 CAMAC系計測データ収集構成図

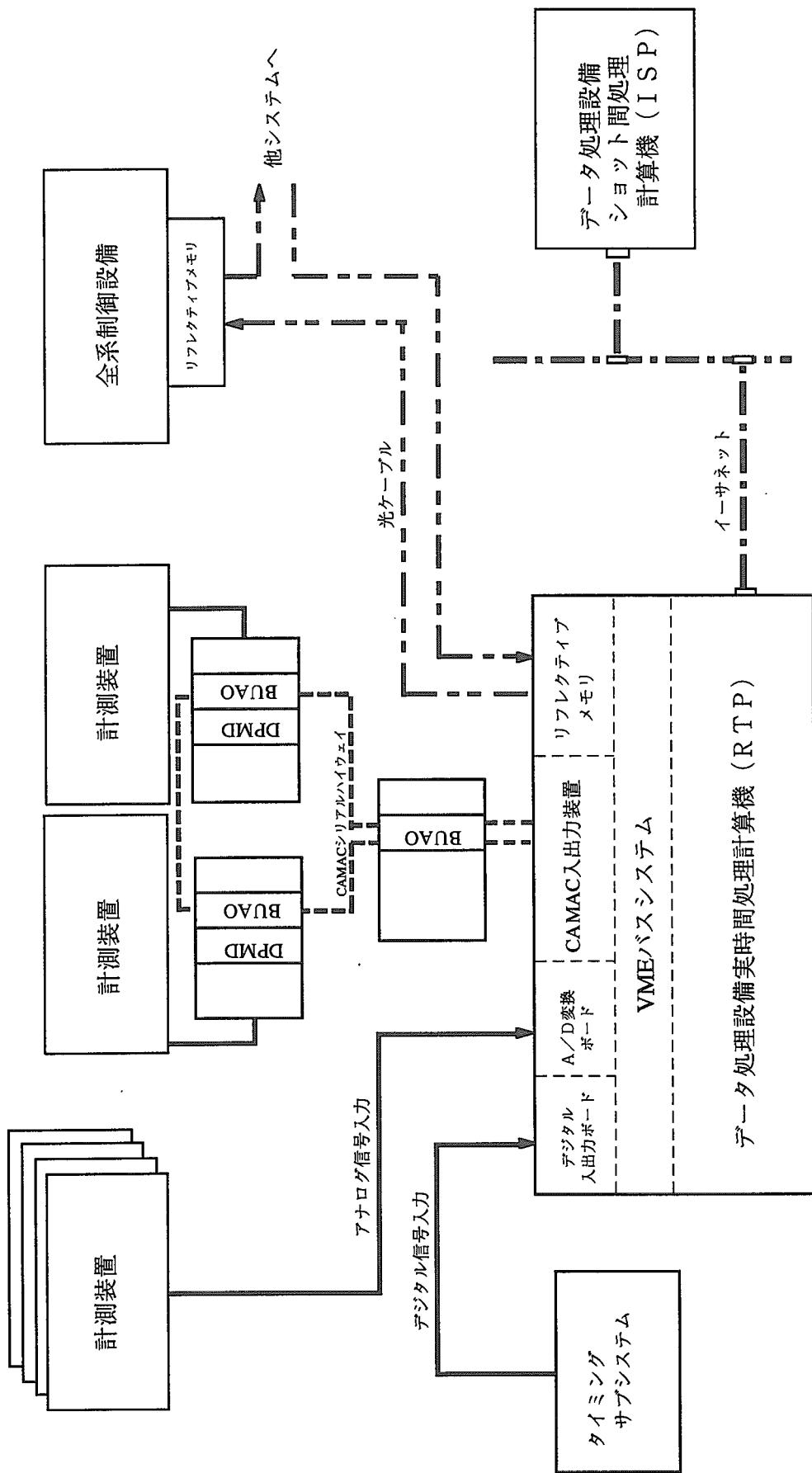


図4.1 実時間制御系システム構成図

#### 4. 2. 2 K2140用デバイスドライバの開発

CAMAC入出力装置K2140は市販の製品ではあるものの、システムに組み込み、オペレーティングシステムから制御を行うために必要となるデバイスドライバが装備されていなかった。このため、この部分については、KINETIC 2917-Z1Aのデスクトップ用SPARC (sun4c) 用のデバイスドライバ<sup>10)</sup>を基本にK2140用のデバイスドライバを構築し<sup>11)</sup>、本システムに導入することに成功した。

これにより、ユーザレベルのアプリケーションプログラムから、システムコールを利用したメモリマッピング機能が使用可能となり、オンライン処理におけるCAMAC初期設定処理、および結果データ転送処理、実時間処理における計測データ取り込み処理等のCAMACアクセス処理がユーザアプリケーションプログラム上から操作可能となった。

#### 4. 2. 3 Dポート付デュアルポートメモリ (DPM'D)

本装置はCAMAC規格のモジュールであり、CAMACシリアルハイウェイDポート、およびCAMACデータウェイの2方向からアクセス可能な2KWのメモリモジュールであり、実時間処理実行時、計測装置より計測データを収集するために用いる。

動作モードは、

- ①送信モード、または受信モード
- ②リザーブ方式、または非リザーブ方式

に分類され、それぞれの組み合わせにより、4通りの動作モードに分類される。

##### (1)送信モード

データウェイ側からメモリへデータを書き込み、シリアルハイウェイ側からそのデータを読み込む場合。

##### (2)受信モード

シリアルハイウェイ側からメモリへデータを書き込み、データウェイ側からそのデータを読み込む場合。

##### (3)リザーブ方式

メモリに対しての書き込み／読み込み動作にインターロックを取り、同時動作を禁止する方式。

##### (4)非リザーブ方式

メモリに対しての書き込み／読み込み動作を互いに意識することなく非同期での動作する方式。

この場合、オーバライト／オーバリードが問題となるが、この状況が生じた場合、その動作を禁止するQ=0モードと、オーバライト／オーバリードを意識しないオーバライト／リードモードがある。

また、送信モード／受信モードの切り替えは、前面パネルのスイッチよりハード的な切り替えで、リザーブ方式／非リザーブ方式の切り替えは、CAMACコマンドで設定する。

現在RTPで使用しているDPMの動作モードは、K2140が構成する光ループをCAMACシリアルハイウェイ側と定義し、「送信モード：非リザーブモード：オーバライト／リード方式」と設定している。図4.3にRTP-計測装置間DPM交信概略図を記す。

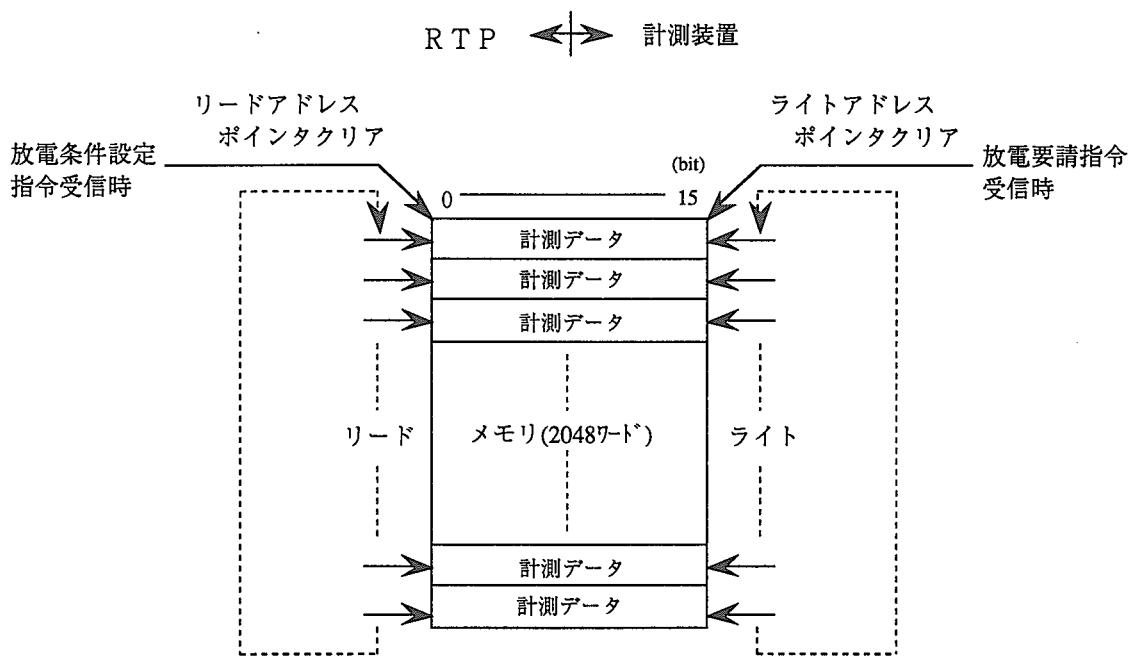


図4.3 RTP-計測装置間DPM交信概略図

#### 4.2.4 A/D変換器

計測装置よりアナログ入力により計測データの収集を行うため、VME基板のA/D変換ボードである米国VMIC社製VMI VME 5576を採用した。以下に本A/D変換器が有する機能について述べる。

##### ①A/D変換速度

$50 \mu s / 16 ch$

##### ②入力レンジ

$+/-10V, +/-5V$

ユーザアプリケーションプログラム上から選択可能である。現在は $+/-10V$ レンジを選択し、計測データの収集を行っている。

##### ③分解能

16 bit

入力レンジを $+/-10V$ 設定した場合、これを1 bitあたりの電圧値に換算すると約 $0.3051mV$  ( $+/-5V$ 選択時は、 $1.525mV$ ) となる。

##### ④データ入力形式

### ディファレンシャル入力／シングルエンド入力

原則としてディファレンシャル入力の形をとっているが、アナログ入力端子板のベース電圧入力用コネクタを短絡することで、シングルエンド入力も測定可能としている。この場合のベース電圧は、実験棟3Fシールドルーム(I)の接地電位と等値である。

#### 4.3 データ処理設備タイミングサブシステムとの接続

実時間処理プロセスは、放電1分前指令受信後～10ms制御停止指令受信の間、全系データ処理設備タイミングサブシステム(以下「タイミングサブシステム」と称す。)を経由し転送される各種タイミング信号により実験放電シーケンスと同期を図る。このため、デジタル信号入力装置として英国PENTLAND社製マルチファンクションボードMPV956を採用した。

また、本デジタル信号入力装置もシステムコールを利用し、ハードウェアのレジスタをユーザアプリケーションプログラム上から直接アクセスすることが可能である。

#### 4.4 全系との接続

実時間帰還制御用パラメータデータの転送を実施するため、全系-RTP間はリフレクティブメモリ<sup>15)</sup>を介し接続される。

リフレクティブメモリは、256Kbyteのメモリを有するVME基板のシステム間共有メモリモジュールであり、コミュニケーションバスは2本の光ファイバーケーブルにより接続される。また、同一路上に2～256台のリフレクティブメモリが接続可能で、ユーザアプリケーションプログラム上から容易にアクセス可能である。

リフレクティブメモリのデータ転送方式は、

- ①ローカルボード上に書き込まれたデータを瞬時に他のボード上のメモリにコピーする  
ブロードキャスト方式
- ②ノード番号を指定することで、任意のボード上に転送するPoint to Point方式

の二通りの機能を有しており、実時間帰還制御用パラメータ転送時は、①の方式を採用している。

また、このデータ転送に関する制御に関しては、ハードウェア上で実施されるため、ソフトウェアのオーバーヘッドがなく、ユーザアプリケーションプログラムからデータ転送時、処理時間を考慮する必要が殆どない。図4.4にリフレクティブメモリ機能概略図を記す。

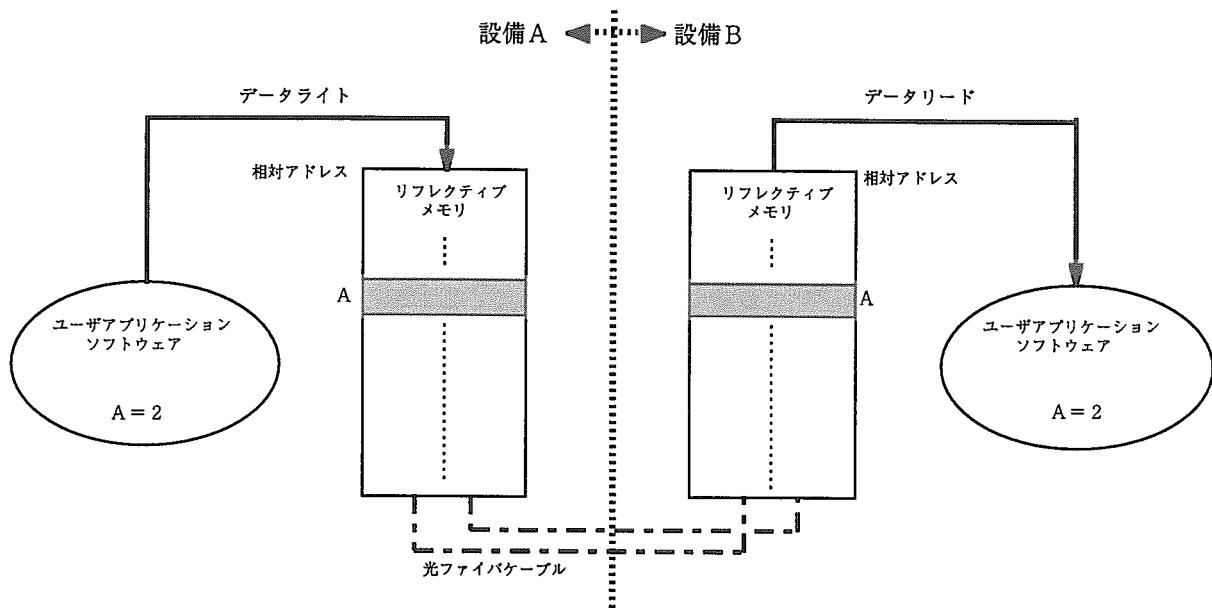


図4. 4 リフレクティブメモリ機能概略図

## 5. RTP の処理構成

RTPは、オンラインプロセスと実時間処理プロセスの2つのプロセスより構成される。

オンラインプロセスは、実験放電シーケンス実施時、上位計算機であるISPと放電制御メッセージの送受信処理を行い、RTPを構成する各機器の動作を実験放電シーケンスに同期させるためのメインプロセスある。

実時間処理プロセスはプラズマ放電中、実時間処理を実行させるためのプロセスであり、本システムが採用するリアルタイム拡張並列UNIXと呼ばれるOSの特徴を利用し、特殊な動作環境のもとで実時間処理を実行する。

また、RTP上で動作するこれらのプロセスは、プログラム改造の容易性、および拡張性を考慮し、C言語を用いてプログラムしている。

### 5. 1 オンラインプロセス

#### 5. 1. 1 オンラインプロセスの起動／停止

ISP主導型でプロセスの起動／停止が行われ、ISPからの接続要求時にinedit（インターネットデーモン機能）により起動され、切り離し要求時にプロセスが停止する。図5. 1にオンラインプロセス動作フローを記す。

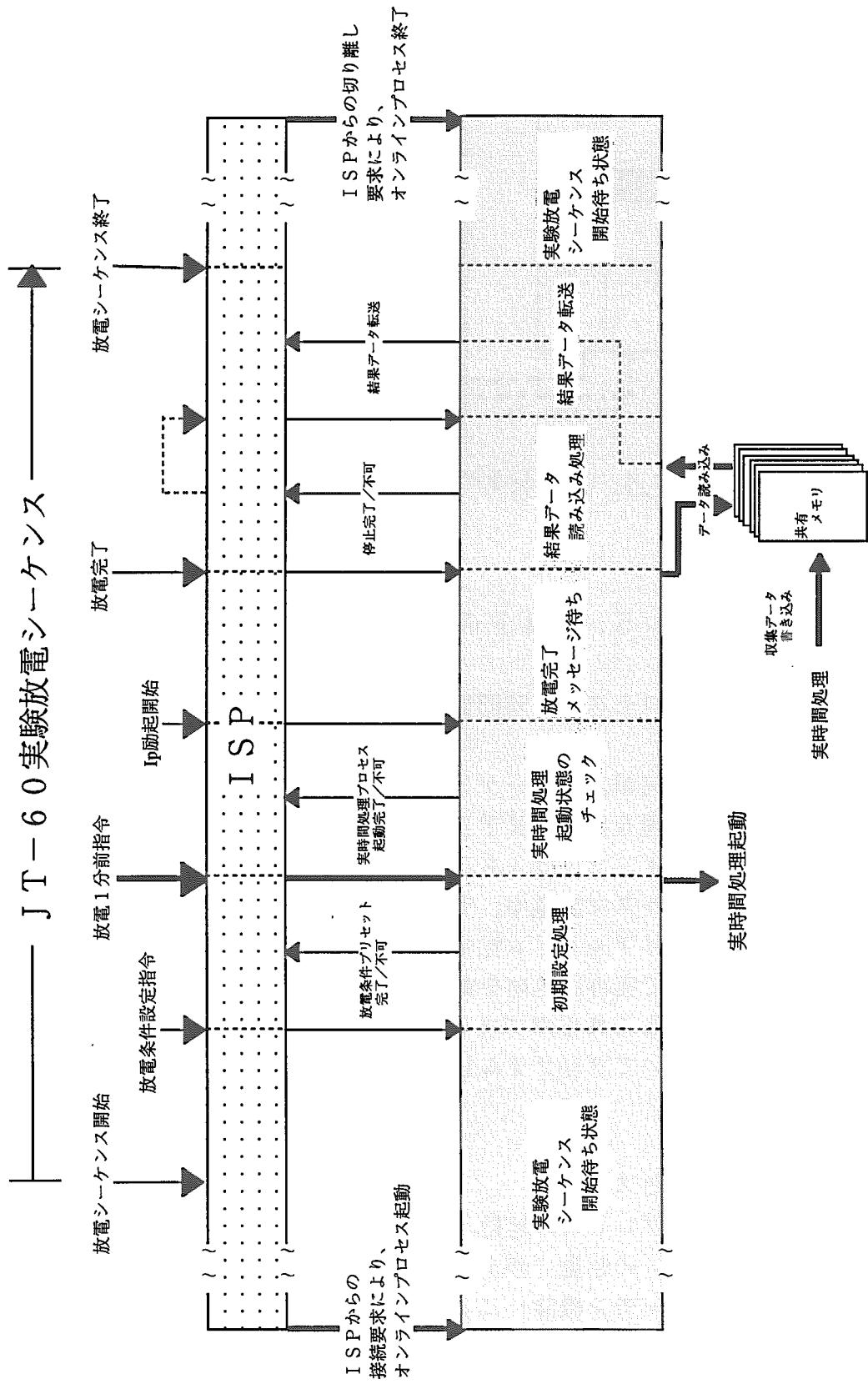
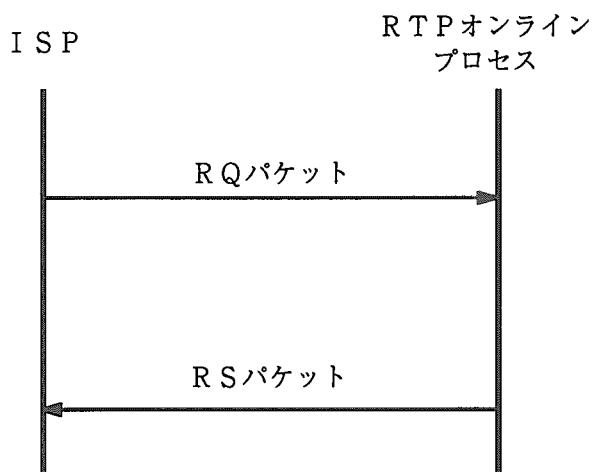


図5. 1 オンラインプロセス概略フロー

### 5. 1. 2 放電制御メッセージの送受信処理

実験放電シーケンスに同期してRTPを構成する各機器の動作、および後述する実時間処理プロセスを生成させるため、上位計算機となるISPと放電制御メッセージの送受信処理をイーサネットを介したパケット通信で実施する。以下にISPより受信する放電制御メッセージを、また、図5. 2にパケット通信基本プロトコルを、図5. 2a～図5. 2pに各々の放電制御メッセージパケット内容を記す。

#### ・一般パケット通信基本プロトコル



#### ・データ収集パケット通信プロトコル

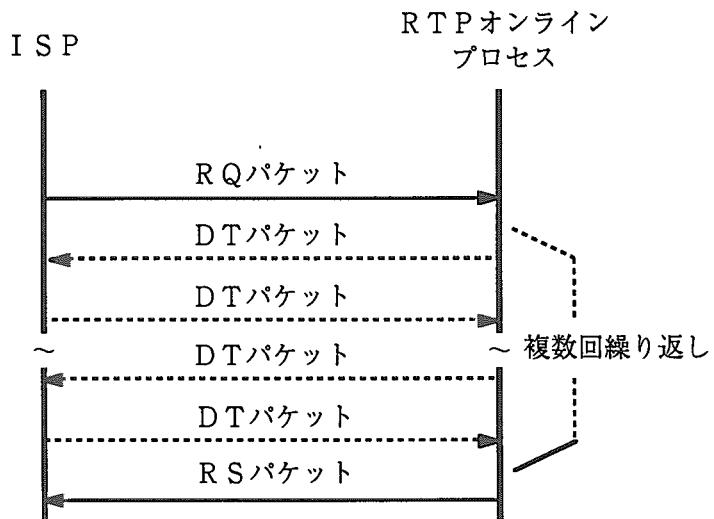
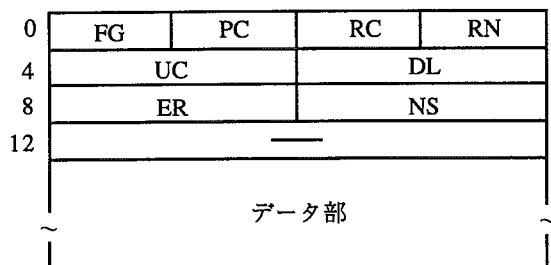


図5. 2 パケット通信基本プロトコル

#### ①放電条件設定指令受信時

- (1) CAMAC初期設定処理
- (2) リフレクティブメモリ初期設定処理

- (3)初期設定処理の正常動作判定  
 ②放電1分前指令  
   (1)実時間処理プロセスの生成  
   (2)正常動作の判定  
 ③放電完了  
   (1)実時間処理プロセスロギングメッセージの取り込み  
   (2)正常動作の判定  
 ④結果データ転送 (ISP - RTP間のみのメッセージ)  
   (1)ISPへの放電結果データ転送  
   (2)正常動作の判定  
 ⑤放電要請解除指令  
   放電シーケンス監視状態フラグの初期化  
 ⑥放電中止指令  
   実時間処理プロセスへの放電中止指令受信通知



FG : 0x7E固定  
 PC : パケットコード  
   0x01 RQパケット  
   0x02 DTパケット  
   0x04 RSパケット  
   0x10 REパケット  
 RC : 要求コード  
   0x22 放電制御指令／応答（固定）  
 RN : パケット通番 (1.....127、0、1.....)

UC : ユーザ定義  
   0x00AA RTP起動状態チェック  
   0x0010 放電条件設定  
   0x0030 放電1分前  
   0x0008 放電完了  
   0x0004 放電要請解除  
   0x0006 放電中止  
   0x00B0 データ収集  
 DL : データ部の長さ（バイト）最大4096バイト  
 ER : RS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  
   1 : エラー  
 DTパケットでは「0」、RQパケットでは不定  
 NS : DTパケットのみ有効でDTパケットの通番を示す。  
   (1.....255、0、1.....)  
 RQ、RSパケットでは、不定

図5. 2 a 一般フォーマット

0	7E	01	22	RN
4	UC=0x10	DL=0x28		
8	—	—		
12	—	—		
16	ショット番号			
20	放電方式	方式内区分	子 番	
24	放電条件 1	放電条件 2		
~	...	...		
52	放電条件 15	放電条件 16		

ショット番号 : interger  
 放電方式 : bit(8)  
 0x01 実験  
 方式内区分 : bit(8)  
 0x01 下記以外  
 0x02 動作試験放洗パターン  
 子番 : short  
 放電条件1～16 : short

図 5. 2 b 放電条件設定RQパケット

0	7E	04	22	RN
4	UC=0x10	DL=0x1C		
8	—	—		
12	—	—		
16	完了コード			
20	0			
24	0			
28	0			
32	c : エラーCAMACコマンド : N A F			
36	0   エラーCAMACコマンド CSR			
40	AI入力エラーコード			

完了コード : interger  
 0 : 放電条件設定完了  
 他 : 放電条件設定不可  
 エラーCAMACコマンド : bit(8)  
 0x01 実験  
 方式内区分 : bit(8)  
 0x01 下記以外  
 0x02 動作試験放洗パターン  
 子番 : short  
 放電条件1～16 : short  
 COD.C01 80～111バイトと同一  
 AI入力エラーコード : 0固定

図 5. 2 c 放電条件設定完了／不可RSパケット

0	7E	01	22	RN
4	UC=0x30	DL=0		
8	—	—		
12	—	—		

図 5. 2 d 放電1分前RQパケット

0	7E	04	22	RN
4	UC=0x30	DL=0x04		
8	ER=0	—		
12	—	—		
16	0	—		

図 5. 2 e 放電1分前RSパケット



0	7E	01	22	RN
4		UC=0x06		DL=0
8	—	—		—
12		—		

図5. 2 j 放電中止RQパケット

0	7E	04	22	RN
4		UC=0x06		DL=0
8	ER	—	—	—
12		—		

ER : 0 下記以外  
 1 エラー  
 放電条件設定RQパケット受信後～放電 1 分前  
 RQパケット受信前のタイミングで放電中止RQ  
 パケットを受信した場合

図5. 2 k 放電中止RSパケット

0	7E	01	22	RN
4		UC=0xB0		DL=0x04
8	—	—	—	—
12		—		
16		転送項目		

転送項目 : 1 ~ 10  
 1 : FIR電子密度データ  
 2 : 中性子発生率データ  
 3 : ダイバータ部放射損失量データ  
 4 : CO2密度データ  
 5 : ダイバータ密度データ  
 6 : 中性粒子圧力データ  
 7 : フーリエ温度データ  
 8 : プラズマ中心部放射損失量データ  
 9 : プラズマ周辺部放射損失量データ  
 10 : 中性粒子圧力比データ

図5. 2 l データ収集RQパケット

0	7E	02	22	RN
4		UC=0xB0		DL
8	—	—	NS	—
12		—	—	
16	データ 1	データ 2		
	⋮	⋮	⋮	⋮

データn : MAX1024点

図5. 2 m データ収集DTパケット



```

/*
 * 2 1 4 0 - Z 1 A用レジスタサイズの指定
 */
typedef struct {
    int          rq;      ---- 要求コード (= 1)
    int          ncl;     ---- 処理すべきCAMA Cコマンドの数
    int          rc;      ---- 戻りコード 0:正常 他:エラー
    int          ndone;   ---- 処理したCAMA Cコマンドの数
    caddr_t     p;       ---- ブロック転送用データアドレス
    int          l;       ---- 上記データ域のバイト長
    int          dum[2];
    parm_t      parm;    ---- CAMA Cコマンドリスト
} CAM_REQ;

typedef struct {
    unsigned char c;      ---- CAMA Cコマンドのクレートアドレス
    unsigned char n;      ---- CAMA CコマンドのステーションNo.
    unsigned char a;      ---- CAMA Cコマンドのサブアドレス
    unsigned char f;      ---- CAMA Cコマンドのファンクション
    unsigned long wd;    ---- m=0 && 16<=f<=23 の時
                           writeデータ (下位24bit)
                           m!=0 の時
                           最大データワード数
    unsigned char w;      ---- データ幅 m!=0 の時のみ有効
                           0:16bit 1:24bit
    unsigned char m;      ---- 転送モード
                           0:シングルアクション
                           1:Q-stop
                           2:Q-Ignore
                           3:Q-Repeat
    unsigned char csr;   ---- 2 1 4 0 - Z 1 AハードウェアのCSRの
                           下位16bit
    unsigned char rd;    ---- m=0 && 0<=f<=7 の時
                           readデータ下位24bit
                           m=0 && 0<=f<=7 の時
                           有効データワード数 (m!=0 の時)
} parm_t;
/*
 * デバイスオーブン処理
 */
*fd = open("/dev/csd0",O_RDWR);      ---- 2 1 4 0 - Z 1 A用デバイスオープン
if(*fd < 0){
    return(ERROR);
}
/*
 * CAMA Cコマンドの実行
 */
CAM_REQ->rq        =1;
CAM_REQ->ncl       =1;
CAM_REQ->rc        =0;
CAM_REQ->ndone     =1;
CAM_REQ->parm_t.wd =1;

```

```

CAM_REQ->parm_t.w      =1;
CAM_REQ->parm_t.m      =1;
CAM_REQ->parm_t.csr    =1;
CAM_REQ->parm_t.rd     =1;
CAM_REQ->parm_t.c      =1;
CAM_REQ->parm_t.n      =1;
CAM_REQ->parm_t.a      =1;
CAM_REQ->parm_t.f      =1;
if((sts_flag=write(*fd,CAM_REQ,sizeof(CAM_REQ)))=-1){ ---- CAMAC コマンドの実行
    return(ERROR);
}

```

- 以下に 2140-Z1A ハードウェアの CSR (Command Status Register) の下位 16 bit の内容を記す。

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ABT	STMO	NEX	N>23	STE	DERR	TMO	NSYC	RPYE	0	LPE	TPE	ADNR	0	NX	NQ

- ABT : ADNR、TMO、STE、NX 4つのOR  
 STMO : ソフトウェアタイムアウト  
 NEX : コマンドリストの実行が打ち切られたため、コマンド未実行  
 N>23 : CAMACステーションナバー (N) が23以上となった。  
 STE : 伝送エラー。RPYE、LPE、TPEのOR  
 DERR : 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  
 NSYC : 同期信号なし。  
 RPYE : エラービットが表示された。  
 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  
 NX : X=0である。  
 NQ : Q=0である。

図5. 3 ①「write システムコール」を利用した CAMAC コマンドの実行例



```

/*
 * CAMAC コマンドの実行
 */
    close_fd = OPEN_Z2140(&fd);           -- 2140-Z1A デバイスオープン関数の呼び出し
    *(Z2140_ADR+CMA) = 0;                -- CAMAC コマンドの設定
    *(Z2140_ADR+CMR) = A1a_READ;        -- CAMAC コマンドの設定
    *(Z2140_ADR+CMR) = CMD_HALT;        -- CAMAC コマンドの設定
    sts_flag = DPMD_EXEC(Z2140_ADR);    -- CAMAC コマンド実行関数の呼び出し
    rcv_data = *(Z2140_ADR+CDR);        -- リードデータの取り込み

/*
 * 2140-Z1A レジスタマッピング処理関数
 */

int
OPEN_Z2140(fd)
int      *fd;
{
unsigned char *sbrk();
unsigned char *addr;
int      sts_flag;

/* デバイスオープン処理 */
*fd = open( "/dev/csd0" , O_RDWR );
if (*fd < 0){
    exit();
}

/* デバイスアドレス取得 */
addr = sbrk(0);
addr = (unsigned char *)roundup(addr);
sts_flag = brk(addr + MAP_SIZE);
if(sts_flag < 0){
    exit();
}

/* レジスタマッピング処理 */
sts_flag = mmap(addr,MAP_SIZE,PROT_READ|PROT_WRITE,MAP_SHARED,*fd,0);
if(sts_flag < 0){
    exit();
}

/* レジスタアドレス設定 */
addr = (unsigned char *)((int)addr + Z2140_ADR_GET);
Z2140_ADR = (unsigned long *)addr;
return(*fd);

}

```

```

/*
 * CAMACコマンドの実行関数
 */

int
DPMD_EXEC(Z2140_ADR)
unsigned long      *Z2140_ADR;
{
unsigned long REG_CSR;
unsigned long STS_CSR;

/* CAMACコマンドの実行 */
*(Z2140_ADR+CMA) = 0;
STS_CSR = *(Z2140_ADR+CSR);
*(Z2140_ADR+CSR) = STS_CSR | TRIGGER_BIT;

/* CAMACコマンド終了監視 */
for(;;){
    REG_CSR = *(Z2140_ADR+CSR);
    if (CMD_END & REG_CSR){
        if(REG_CSR & CAMAC_EXEC){
            return(ERROR);
        }
        else{
            return(NORMAL);
        }
    }
}
}
}

```

図5. 4 ②「mmapシステムコール」を利用した  
CAMACコマンドの実行例

各々のCAMACコマンド実行方法の特徴は、

- ① 「writeシステムコール」を利用したCAMACコマンドの実行  
発行するシステムコールは、writeコマンドだけであるため、非常に単純にCAMACコマンドの実行が可能。但し、CAMACコマンド実行処理速度は、システム内の利用状況、他プロセスの動作状況に依存する部分が大きい。
- ② 「mmapシステムコール」を利用したCAMACコマンドの実行  
システムコールを発行してシステム上に割り当てられるメモリアドレス等を取得しなければいけないため、動作環境設定処理が複雑となる。但し、動作環境設定処理終了後は、ユーザアプリケーションプログラム上からK2140のレジスタを直接操作可能となるため、CAMACコマンド実行処理速度が高速である。  
これらのCAMACコマンド実行方法の特徴を利用し、①は、処理時間に比較的余裕のあるオンラインプロセスにて実行するCAMAC初期設定処理に、②は、処理速度を重要視する実時間処理実行時の計測データの収集に用いている。

また、当然のことではあるが、①、②ともに同一C A M A C コマンドを実行した場合、得られる結果は同一である。

以下に実際に使用しているC A M A C コマンドを記す。

- ・バイパスビットクリア : C 1 N 1 3 A 2 F 1 7  
                              : C 2 N 1 3 A 2 F 1 7
  - ・非リザーブモード設定 : C 1 N 1 3 A 2 F 1 6  
                              : C 2 N 1 3 A 2 F 1 6
  - ・リードアドレスポイントクリア : C 1 N 1 3 A 1 F 9  
                              : C 2 N 1 3 A 1 F 9
  - ・データリード : C 1 N 1 3 A 1 F 0  
                              : C 2 N 1 3 A 1 F 0
- C = 1 : F I R 干渉測定装置  
C = 2 : フーリエ変換分光測定装置

#### 5. 1. 4 リフレクティブメモリ初期設定処理

リフレクティブメモリは、256Kbyteのメモリを有するシステム間共有メモリモジュールであり、実時間処理実行時に全系-R T P 間の実時間帰還制御用パラメータデータの転送に用いている。

オンラインプロセスは、ISPより放電条件設定指令受信時にリフレクティブメモリ初期設定処理としてリフレクティブメモリ上の所定のアドレス（通番書き込みエリア）に「-1」を書き込む。

この処理は、実時間処理プロセスがI p 励起開始タイミング信号検出時に開始する実時間処理に於いて、1周期目の実時間帰還制御用パラメータデータをリフレクティブメモリに対して書き込む前に全系がリフレクティブメモリの該当アドレス上に残ったデータを初回時の実時間帰還制御用パラメータデータとして読み込むことを防ぐための処理であり、全系-R T P 間の実時間帰還制御用パラメータデータの時間的な整合性を図るための手段となっている。図5. 5にリフレクティブメモリ設備間メモリ割り当てを記す。

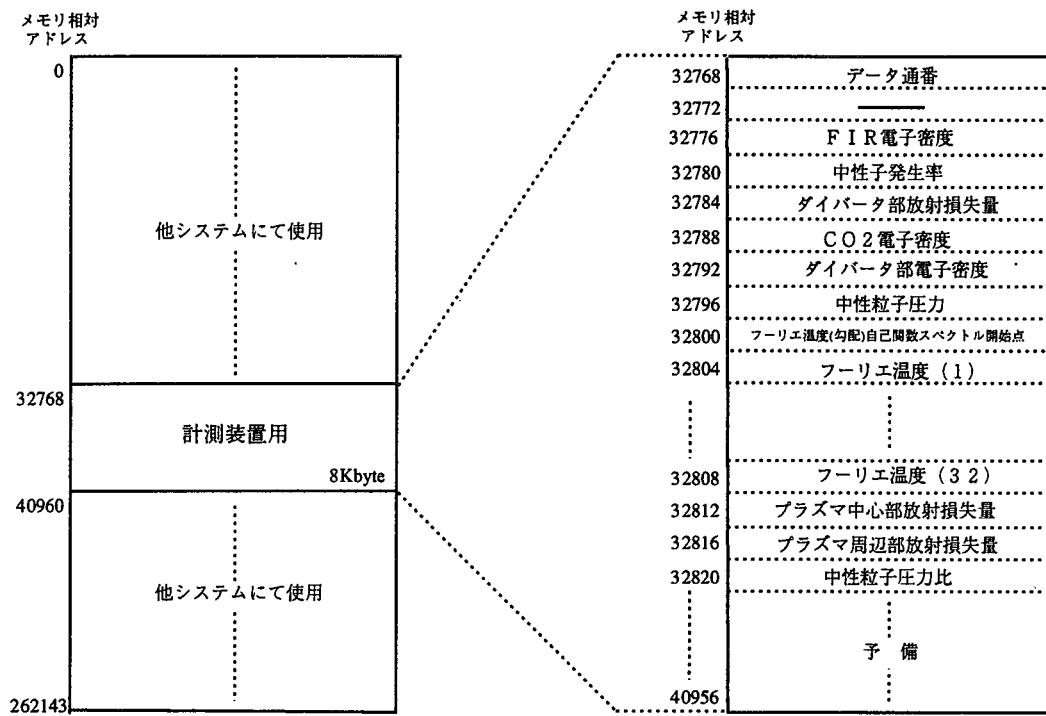


図 5. 5 リフレクティブメモリ設備間メモリ割り当て

## 5. 2 実時間処理プロセス

### 5. 2. 1 実時間処理プロセスの生成／消滅

オンラインプロセスが I S P より「放電 1 分前指令」メッセージ受信時にオンラインプロセスによりオンラインプロセスの子プロセスとして生成される。図 5. 6 に実時間処理プロセス動作フローを記す。

また、

- ① 10 ms 制御停止タイミング信号（タイミングパルス信号）
  - ② 放電中止指令（放電制御メッセージ）
- のいずれかの要因により自プロセスを消滅させる。

### 5. 2. 2 実時間処理環境の設定

実時間処理プロセスは、起動された時点ではオンラインプロセス下の子プロセスという状態であるため、特にシステム上での特権等は与えられていない。

このため、実時間処理プロセスは、自プロセスに対して以下に述べるような実時間処理環境の設定を行う。

- ① 実時間処理プロセスの優先度の設定
  - ② 実時間処理実行 C P U の設定
  - ③ 実時間処理実行 C P U に対する割り込み処理の禁止
- これらの設定を行うことで実時間処理環境を作り出し、実時間処理を実施している。

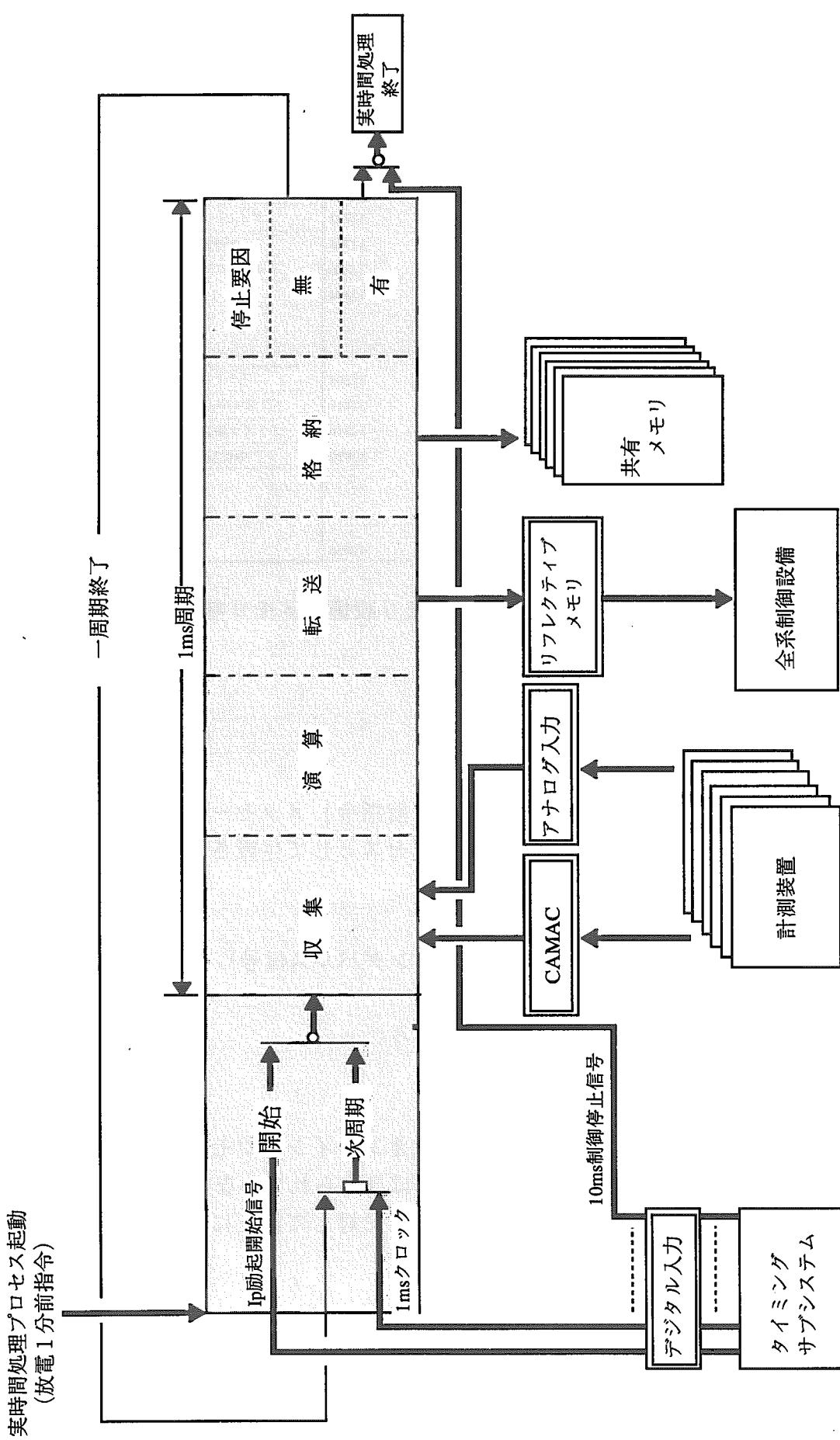


図 5. 6 実時間処理プロセス動作フロー

### 5. 2. 3 タイミング信号の検出

実時間処理プロセスは、起動～停止するまでの間、各種タイミング信号を検出し、放電シーケンスとの同期を図る。実時間処理プロセスが検出するタイミング信号は、

- (1) Tコイル励磁開始タイミング
- (2) 10ms制御準備タイミング
- (3) Ip励起開始タイミング
- (4) 10ms制御停止タイミング
- (5) 全系1msクロック信号

の五つである。また、検出するタイミング信号の種類は、

(1) タイミングパルス信号  
(2) あるタイミングを起動タイミングとし、それ以降連続的に発生するクロック信号の二つに大別できる。上記、実時間処理プロセスが検出するタイミング信号中、タイミングパルス信号に該当するタイミング信号は、

- (1) Tコイル励磁開始タイミング
- (2) 10ms制御準備タイミング
- (3) Ip励起開始タイミング
- (4) 10ms制御停止タイミング

であり、クロック信号に該当するタイミング信号は、

- ・全系1msクロック信号

である。

#### ① タイミングパルス信号の検出

タイミングパルス信号は、「ある特定の事象が発生した時のみ発生するタイミング信号」である。

実時間処理プロセスではタイミング信号検出ロジック中に意識的に無限ループを組み込むことで、任意のタイミング信号検出時にのみ無限ループを脱出可能となるようプログラミングし、実時間処理プロセスと放電シーケンスとの同期を図っている。

図5. 7にタイミングパルス信号の時間幅、図5. 8にタイミング信号検出ロジック、および図5. 9に放電シーケンスとの同期を記す。

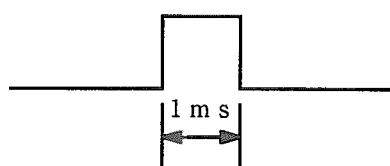


図5. 7 タイミングパルス信号の時間幅

```

for(;;){
    if(!(MPV956_ADR[0x50/2] & TCOIL_RAGE)){
        break;
    }
}

```

----- Tコイル励磁開始タイミング待ち  
----- 無限ループからの脱出

図5. 8 タイミング信号検出口ロジック  
(例: Tコイル励磁開始タイミング待ち)

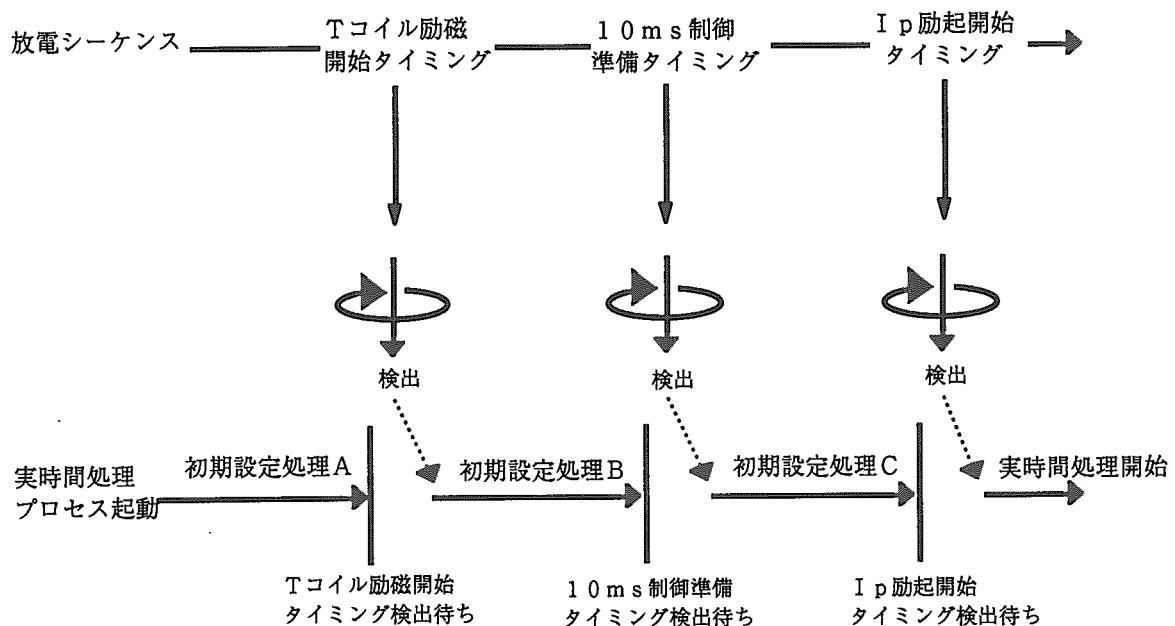


図5. 9 放電シーケンスとの同期

## ②クロック信号の検出

クロック信号は、「あるタイミングを起動タイミングとし、それ以降連続的に発生するタイミング信号」である。

実時間処理プロセスでは、このクロック信号を計測データ収集、演算、転送と言う一連の実時間処理の制御周期として使用している。

また、1 ms クロック信号検出中においても、実時間処理停止タイミングとなる「10 ms 制御停止指令」を検出しなければならないため、実時間処理実行中は、クロック信号、およびタイミングパルス信号の検出を同一ロジック上で行っている。

図5. 10 に全系1 ms クロック信号の時間幅、図5. 11 に全系1 ms クロック信号検出口ロジック、および図5. 12 に全系1 ms クロックとの同期を記す。

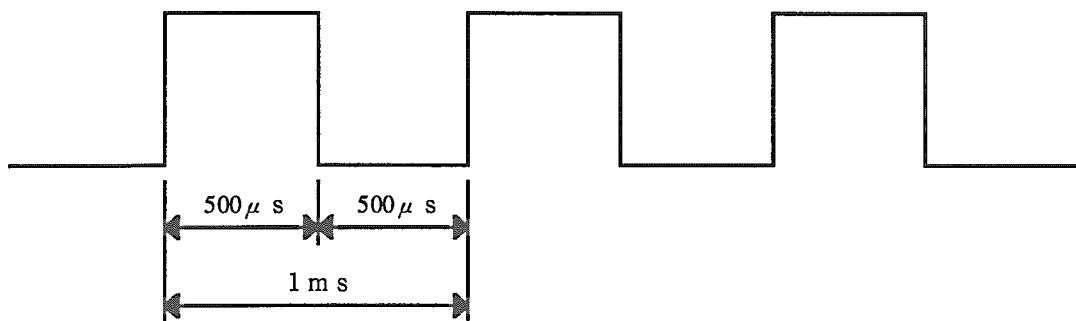


図 5. 10 全系 1 m s クロック信号の時間幅

```

for(;;){
    if(!(MPV956_ADR[0x50/2] & CNTL_STP_10MS)){           ----- 10 m s 制御停止指令の有／無確認
        return(ERROR);
    }
    if(!(MPV956_ADR[0x50/2] & CNTL_CYCLE)){             ----- 1 m s クロックの検出開始
        for(;;){
            if(!(MPV956_ADR[0x50/2] & CNTL_STP_10MS)){       ----- 10 m s 制御停止指令の有／無確認
                return(ERROR);
            }
            if(MPV956_ADR[0x50/2] & CNTL_CYCLE){           ----- 1 m s クロックの検出
                return(NORMAL);
            }
        }
    }
}
    
```

図 5. 11 全系 1 m s クロック信号検出ロジック

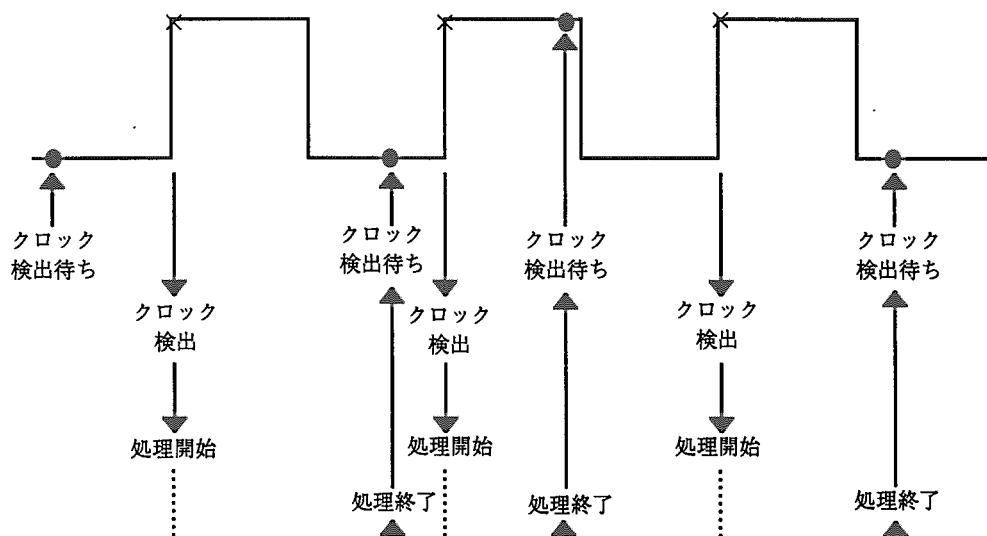


図 5. 12 全系 1 m s クロックとの同期

### ③タイミング信号の使用用途

以下に実時間処理に於ける各々のタイミング信号の使用用途を述べる。

#### (1) Tコイル励磁開始タイミング

演算処理用係数取り込み開始タイミングとして使用

#### (2) 10 ms 制御準備タイミング

A/D変換器ベース電圧の測定

#### (3) I<sub>p</sub>励起開始タイミング

実時間処理開始タイミングとして使用

#### (4) 10 ms 制御停止タイミング

実時間処理、および実時間処理プロセス終了タイミングとして使用

#### (5) 全系 1 ms クロック

実時間処理の周期用クロックとして使用

## 5. 2. 4 実時間処理

実時間処理プロセスは、I<sub>p</sub>励起開始信号受信～10 ms 制御停止タイミング受信までの間、全系制御設備より送信される1 ms クロックに同期し以下に記す実時間処理を実施する。

### ①実時間帰還制御用パラメータデータの取り込み

現在、実時間処理にて収集する計測データの収集方法は、

#### (1) CAMACシリアルハイウェイを介し、DPMD交信にて収集する方法

(2) A/D変換器を介しアナログデータ入力により収集する方法  
の2通りである。

以下に実時間処理にて収集する計測データを記す。

#### (1) FIR電子密度データ

#### (2) CO<sub>2</sub>電子密度データ

#### (3) 中性子発生率データ

#### (4) ダイバータ部放射損失量データ

#### (5) 中心部放射損失量データ

#### (6) 周辺部放射損失量データ

#### (7) ダイバータ領域密度データ

#### (8) ダイバータ部中性粒子圧力データ

#### (9) ダイバータ部中性粒子圧力比データ

#### (10) フーリエ分光器データ

### ②演算処理、および取得データの健全性のチェック

上記①中に記す10種類の計測データを収集した後、各々の計測データ毎に定められた演算処理を実施する。各々の計測データを用いた演算処理について、以下に記す。

## (1) F I R 電子密度

## ・演算式

なし。

収集した計測データに対し、健全性のチェック後、全系へ転送。

## ・データ収集方式

D P MD 交信

## ・データ不良判定条件

$$1) \quad Pd_{(n)} < 0$$

$$2) \quad Pd_{(n-2)} - Pd_{(n)} \leq 155 \quad (3 \text{ 周期連続検出時})$$

 $Pd_{(n)}$  : 今回計測値 $Pd_{(n-2)}$  : 前々回計測値但し、1)、2) 共に  $t = 110 \text{ ms}$  までは無効とする。

## (2) C O 2 電子密度

## ・演算式

なし。

収集した計測データに対し、健全性のチェック後、全系へ転送。

## ・データ収集方式

アナログデータ入力 1 c h

## ・データ不良判定条件

$CO2_V < 0$

 $CO2_V$  : 今回測定電圧値

## (3) 中性子発生率

## ・演算式

$$Sn = a_0 + a_1 \cdot V^2 + a_2 \cdot V^4 + a_3 \cdot V^6$$

 $Sn$  : 中性子発生率 $a_0, a_1, a_2, a_3$  : 係数 $V$  : 測定電圧

## ・データ収集方式

アナログデータ入力 1 c h

## ・データ不良判定条件

$Sn < 0$

## (4) ダイバータ部放射損失量

## ・演算式

$$Pr(n) [\text{Watt}/\text{m}^2] = \alpha \cdot V_{(n)} + \beta \cdot V_{(n-1)} + \gamma \cdot V_{(0)}$$

: 各チャンネル毎

$\alpha = a(1/\Delta t + 1/T)$

$\beta = -a/\Delta t$

$\gamma = -a/T$

 $Pr(n)$  : 放射強度今回値 $V_{(n)}$  : 今回測定電圧 $V_{(n-1)}$  : 前回測定電圧

$V_{(0)}$	: 初回測定電圧
$\alpha$	: 校正係数
T	: 冷却時間
$\Delta t$	: 時間ステップ (1 m s)

$P_{DIV} [MW] = 10^{-6} \cdot \sum_{k=1}^8 M_k \cdot C_k \cdot [P_k - (0.86P_{M6} + 0.4P_{M7})]$	
$C_k = 2\pi \cdot R_k \cdot W_k$	
$P_{DIV}$	: ダイバータ部放射損失量
$P_k$	: 各チャンネル毎放射強度値
$P_{M6}$	: 主プラズマ放射強度値 (チャンネル 6)
$P_{M7}$	: 主プラズマ放射強度値 (チャンネル 7)
$R_k, W_k$	: 計測点位置 (チャンネル毎に固定)
$M_k$	: 補間係数

- ・データ収集方式  
アナログデータ入力 10 c h
- ・データ不良判定条件  
 $P_{DIV} > 32767$

## (5) 主プラズマ中心部放射損失量

- ・演算式

$P_{r(n)} [Watt/m^2] = \alpha \cdot V_{(n)} + \beta \cdot V_{(n-1)} + \gamma \cdot V_{(0)}$	: 各チャンネル毎
$\alpha = a(1/\Delta t + 1/T)$	
$\beta = -a/\Delta t$	
$\gamma = -a/T$	
$P_{r(n)}$	: 放射強度今回値
$V_{(n)}$	: 今回測定電圧
$V_{(n-1)}$	: 前回測定電圧
$V_{(0)}$	: 初回測定電圧
$a$	: 校正係数
T	: 冷却時間
$\Delta t$	: 時間ステップ (1 m s)

$P_{MAIN} [MW] = 10^{-6} \cdot \sum_{m=1}^3 M_m \cdot C_m \cdot P_m$	
$C_m = 2\pi \cdot R_m \cdot W_m$	
$P_{MAIN}$	: 主プラズマ中心部放射損失量
$P_m$	: 各チャンネル毎放射強度値
$R_m, W_m$	: 計測点位置 (チャンネル毎に固定)
$M_m$	: 補間係数

- ・データ収集方式  
アナログデータ入力 3 c h

- ・データ不良判定条件  
 $P_{MAIN} > 32767$

## (6) 主プラズマ周辺部放射損失量

- ・演算式

$$Pr(n)[\text{Watt/m}^2] = \alpha \cdot V(n) + \beta \cdot V(n-1) + \gamma \cdot V(0)$$

: 各チャンネル毎

$$\alpha = a(1/\Delta t + 1/T)$$

$$\beta = -a/\Delta t$$

$$\gamma = -a/T$$

$Pr(n)$  : 放射強度今回値

$V(n)$  : 今回測定電圧

$V(n-1)$  : 前回測定電圧

$V(0)$  : 初回測定電圧

$\alpha$  : 校正係数

$T$  : 冷却時間

$\Delta t$  : 時間ステップ (1 ms)

$$P_{SID}[\text{MW}] = 10^{-6} \cdot \sum_{s=1}^6 M_s \cdot C_s \cdot P_s$$

$$C_s = 2\pi \cdot R_s \cdot W_s$$

$P_{SID}$  : 主プラズマ中心部放射損失量

$P_s$  : 各チャンネル毎放射強度値

$R_s, W_s$  : 計測点位置 (チャンネル毎に固定)

$M_s$  : 補間係数

- ・データ収集方式

アナログデータ入力 6 c h

- ・データ不良判定条件

$$P_{SID} > 32767$$

## (7) ダイバータ領域密度

- ・演算式

$$V_{D(n)} = V_{D(n)} - V_{D(0)}$$

$V_{D(n)}$  : 今回測定電圧値

$V_{D(0)}$  : 初回 (基準) 測定電圧値

但し、

$$V(y) = V_{D(n)} - V_{D(0)}$$

$V_{D(y)} - V_{D(n-1)} \geq 0.7$  が成立する場合

$$V(y) = V(y) + 0.5 \quad (\text{小数点1桁目を四捨五入})$$

$$V_{D(0)} = V(y)$$

$V_{(y-1)}$  : 前回算出電圧値

- ・データ収集方式

アナログデータ入力 1 c h

- ・データ不良判定条件  
 $V(y) < 0$

## (8) ダイバータ部中性粒子圧力

- ・演算式

$$P_{obs}^0 = a + b[I_i / (I_e - I_i)]$$

$$I_i = (S_i / G_i)V_i(t)$$

$$I_e = (S_e / G_e)V_e(t)$$

$P_{obs}^0$  : ダイバータ中性粒子圧力実測値(Pa)

a、b : 比例定数

$S_i$ 、 $S_e$  : 測定系の感度(A/D)

$G_i$ 、 $G_e$  : 測定系のゲイン

$V_i(t)$  : 測定電圧(イオン電圧)

$V_e(t)$  : 測定電圧(エミッション電圧)

- ・データ収集方式

アナログデータ入力 2ch

- ・データ不良判定条件

$$1) I_e < 10^{-6}$$

$$2) I_e > 10^{-3}$$

$$3) I_i \geq I_e$$

$$4) P_{obs}^0 < 0$$

## (9) ダイバータ部中性粒子圧力比

- ・演算式

$$P_{obs}^1 = a1 + b1[I_{i1} / (I_{e1} - I_{i1})]$$

$$I_{i1} = (S_{i1} / G_{i1})V_{i1}(t)$$

$$I_{e1} = (S_{e1} / G_{e1})V_{e1}(t)$$

$P_{obs}^1$  : ダイバータゲージ(1)中性粒子圧力実測値(Pa)

a1、b1 : ゲージ(1)比例定数

$S_{i1}$ 、 $S_{e1}$  : ゲージ(1)測定系の感度(A/D)

$G_{i1}$ 、 $G_{e1}$  : ゲージ(1)測定系のゲイン

$V_{i1}(t)$  : ゲージ(1)測定電圧(イオン電圧)

$V_{e1}(t)$  : ゲージ(1)測定電圧(エミッション電圧)

$$P_{obs}^2 = a2 + b2[I_{i2} / (I_{e2} - I_{i2})]$$

$$I_{i2} = (S_{i2} / G_{i2})V_{i2}(t)$$

$$I_{e2} = (S_{e2} / G_{e2})V_{e2}(t)$$

$P_{obs}^2$  : ダイバータゲージ(2)中性粒子圧力実測値(Pa)

a2、b2 : ゲージ(2)比例定数

$S_{i2}, S_{e2}$  : ゲージ(2)測定系の感度(A/D)  
 $G_{i2}, G_{e2}$  : ゲージ(2)測定系のゲイン  
 $V_{i2}(t)$  : ゲージ(2)測定電圧(イオン電圧)  
 $V_{e2}(t)$  : ゲージ(2)測定電圧(エミッション電圧)

$$r_{12} = P_{obs}^1 / P_{obs}^2$$

- ・データ収集方式  
アナログデータ入力 4 c h
- ・データ不良判定条件

$$1) I_{e1} < 10^{-6}$$

$$2) I_{e1} > 10^{-3}$$

$$3) I_{i1} \geq I_{e1}$$

$$4) P_{obs}^1 < 0$$

$$5) I_{e2} < 10^{-6}$$

$$6) I_{e2} > 10^{-3}$$

$$7) I_{i2} \geq I_{e2}$$

$$8) P_{obs}^2 \leq 0$$

#### (10) フーリエ分光器

- ・演算式

$$TECH_{(n)} = \sqrt{A_{(n)} * F_{(n)}} / (\sqrt{2} * 1024)$$

$$A = P * 2^{63}$$

$n$  : チャンネルNo.(1~32チャンネル)  
 $P$  : 読み込みデータ  
 $F_{(n)}$  : チャンネル毎変換係数

- ・データ収集方式  
DPM-D交信

- ・データ不良判定条件

$$1) TECH_{(n)} < 0$$

$$2) TECH_{(n)} > 32767$$

### ③ 実時間帰還制御用パラメータデータの転送

演算処理実施後のデータを実時間帰還制御用パラメータデータとしてリフレクティブメモリを介し、全系に転送する。

### ④ 結果データの格納

演算処理にて算出されたデータを放電後にオンラインプロセスが「放電結果データ」として I S Pへ転送するため、共有メモリ上に格納する。

### ⑤ 停止要因の判定

実時間処理の停止要因は、

- (1) 10 ms 制御停止指令

(2)放電中止指令受信  
のいずれかである。

### 5. 2. 5 実時間帰還制御用機器健全性チェック

実時間処理実行時、リフレクティブメモリを介し、全系-RTP間で以下のように機器健全性チェックを実施している。

#### ①計測装置健全性チェック

5. 2. 4 項②演算処理と同時に計測データの健全性のチェックを実施する。このチェックは、収集した計測データ、あるいは演算後のデータに対し、各々の計測装置毎に制限値やデータ不良判定条件を設け、その条件を逸脱した計測装置をデータ不良として判定し、データ不良を起こした計測装置を用いた実時間帰還制御を中止するよう、全系に対して通知するために行う。この通知は、全系-RTP間のリフレクティブメモリを介して実施される。通知方法は、あらかじめリフレクティブメモリ上に各計測装置の実時間帰還制御用パラメータデータを書き込むアドレスを全系-RTP間で取り決めておき、データ不良を起こした計測装置に該当するアドレスに対し、「-1」を15回連続で書き込むことで全系に対してデータ不良を通知している。この15回連続の書き込みは、実時間処理の制御周期(1ms)と全系の制御周期(10ms)が異なることを考慮した処理である。また、全系にデータ不良を送信している間も実時間処理は引き続き計測データの収集、演算処理を実施する。

図5. 13にデータ不良検出手順を記す。

#### ②通番チェック

実時間処理は、リフレクティブメモリ上の所定のアドレスに毎周期「通番」を書き込む。これは全系側でRTPの健全性の検査を行うためのもので、全系側が通番データ読み込み時、前回の通番と今回の通番が同一であった場合、RTPの異常と判断し、それ以降の実時間帰還制御を中止する。

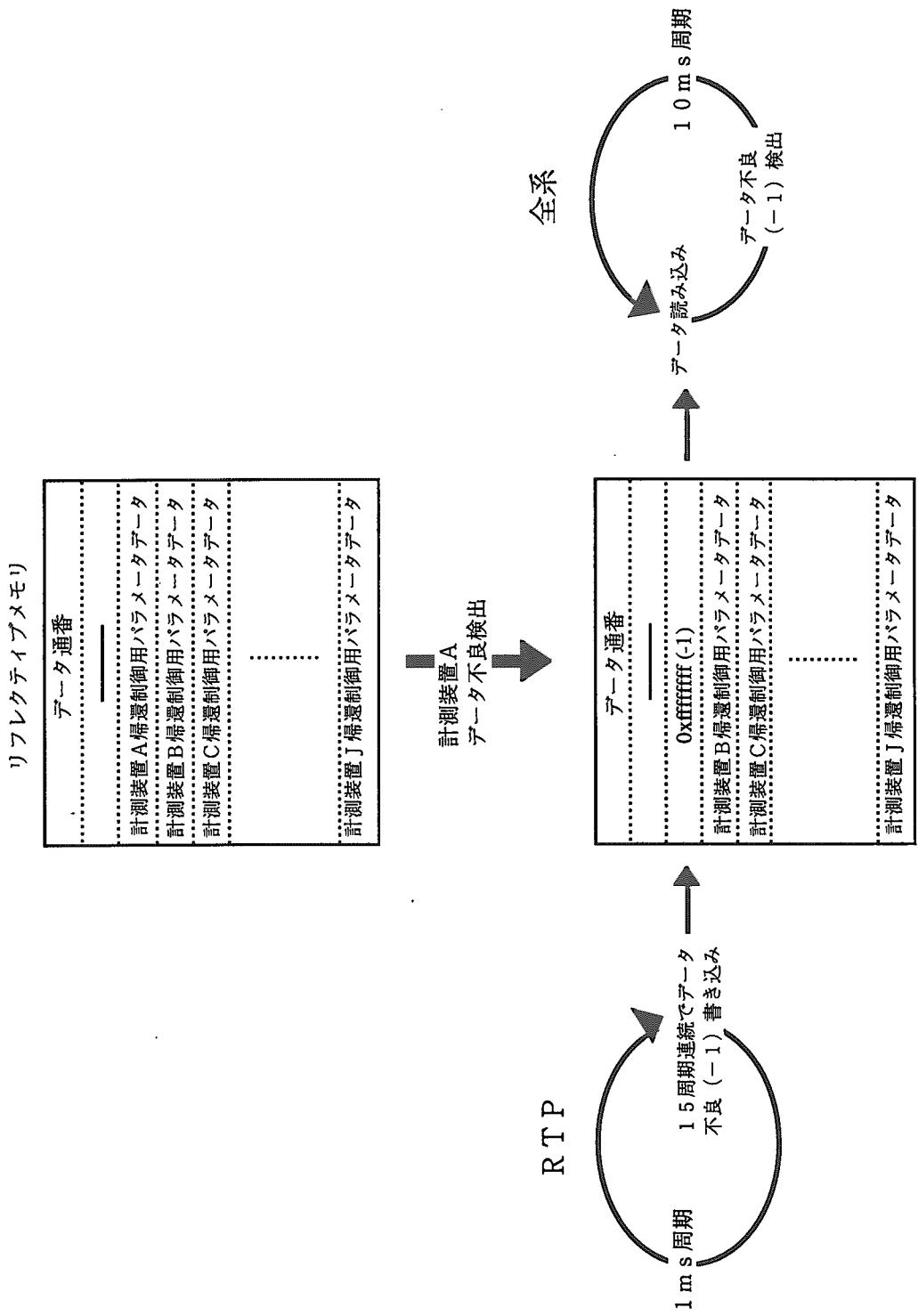


図5.13 データ不良検出手順

### 5. 2. 6 A/D変換器ベース電圧の測定

実時間処理プロセスは、10 ms 制御準備タイミング信号検出時、A/D変換器の各チャンネルのベース電圧の測定を行う。

この測定処理は各々のチャンネルに対し100回行われ、その平均値をベース電圧として実時間処理中に各々のチャンネルから測定した電圧データより差し引くことで測定誤差を小さくしている。

### 5. 3 プロセス間、およびC P U間通信

実験放電シーケンス中、以下の項目についてオンラインプロセス-実時間処理プロセス間、およびC P U間で共有メモリを介して情報の受け渡しを行う。

#### ①放電中止指令受信通知

放電1分前～I p 励起開始タイミング受信までの間、有効となる通知である。

放電1分前～10 ms 制御停止タイミング受信までの間、オンラインプロセスと実時間処理プロセスは、別C P U上で並列に動作する。この間、オンラインプロセスは I S Pからの放電制御メッセージ受信待ち状態となり、実時間処理プロセスはタイミング信号待ち状態となる。

放電中止指令は、放電制御メッセージとして I S Pからオンラインプロセスに対して通知されるため、オンラインプロセスが何らかの方法で実時間処理プロセスに対し、放電中止指令受信の通知を行わない限り実時間処理プロセスは放電中止指令受信を認識することが出来ず、その結果、タイミング信号待ち状態のままシステム上に残り続けるという現象が起こる。

このため、オンラインプロセス-実時間処理プロセス間の放電中止指令受信通知は、共有メモリを介して行っており、実時間処理プロセスはタイミング信号待ち状態であっても、共有メモリを介して放電中止指令受信通知の有無をチェックするようプログラムされている。

また、I p 励起開始タイミング受信以降については、設備間の取り決めにより10 ms 制御停止タイミング信号が必ず出力されることになっており、実時間処理プロセス自身がタイミング信号を検出することが可能であるため、放電中止指令受信通知に関するプロセス間、およびC P U間の通信は行わない。

#### ②放電結果データの受け渡し

実時間処理プロセスは、実時間処理実行時に実時間帰還制御用パラメータデータをリフレクティブメモリを介して全系へ転送すると同時に共有メモリ上に格納する。これは、オンラインプロセスが放電後に共有メモリ上に格納された実時間帰還制御用パラメータデータを読み出し、放電結果データとして I S Pに転送するためである。

なお、実時間帰還制御用パラメータデータを格納する場合のデータサンプリングピッチは、ユーザアプリケーションプログラムより容易に設定でき、1～10 ms の間で可変である。

## 5. 4 ロギング機能

I S Pより転送される放電制御メッセージ、および受信した放電制御メッセージに対応して実施した処理結果を記録する。

また、実時間処理実行時に収集する計測データ不良に関するエラーロギングは、初回検出時のみ記録している。これは、プラズマ放電中、実時間処理が1 m s周期で動作し続けるため、15秒放電を行うことを仮定した場合、一つのメッセージに対し最高1500回のロギング処理を行う可能性があり、ロギング処理にのみC P Uを占有されることで実時間処理が動作出来ず制御周期遅延となることを防止するための処理である。

これらのロギングの記録は、システムコンソールよりコマンドを入力することで、容易に表示可能である。

## 5. 5 R T P不具合発生時の回避処理

オンラインプロセスは、I S Pより「放電条件設定指令」を受信することで実験放電シーケンスが開始されたことを認識する。これは、I S Pが全系から転送される放電条件設定指令に含まれる放電条件データ中の「V276：R T P使用／不使用」項目が「使用」と選択された実験放電シーケンスのみR T Pに対して放電制御メッセージを転送するためである。

この処理はR T Pが何らかの理由により不具合が発生した場合、放電条件により「R T P不使用」を選択することで実験放電シーケンスを続行できるよう考慮しI S P側に設けたものである。但し、「R T P不使用」にて実時間帰還制御を実施することはできない。

## 5. 6 リフレクティブメモリが持つ懸念事項

リフレクティブメモリは、ユーザーアプロケーションプログラムから容易にアクセス可能であり、データ転送速度もC A M A Cシリアルハイウェイを用いたデータ転送と比較した場合、非常に高速であると言える。R T Pでは、リフレクティブメモリ導入以前、実時間帰還制御用パラメータデータの転送にC A M A Cシリアルハイウェイを介したD P M D交信を採用していた。以下に各々のデータ転送方式を用いた場合の実時間処理における実時間帰還制御用パラメータデータ転送所要時間実測結果を記す。

- |                   |             |
|-------------------|-------------|
| ①C A M A Cアクセス処理  | 20 μ s／アクセス |
| ②リフレクティブメモリアクセス処理 | 1 μ s／アクセス  |

上記の実測結果からもわかるようにリフレクティブメモリの採用により実時間帰還制御用パラメータデータの転送処理は、約20倍高速化することができた。

しかし一方で同一ループ上に多数のリフレクティブメモリが接続されることにより、各々の設備が管理するリフレクティブメモリに対して、任意のエリアにデータをライトすることが可能であるために予期せぬデータが書き替わる可能性があり、リフレクティブメモリ上のデータの信用性に問題が発生することが懸念される。

このため、全系が中心となり、設備間の取り合い事項として、同一ループ上に接続されるリフレクティブメモリに対するアクセスエリアの割り当てを行っている。今後、リ

フレクティブメモリを介したデータ転送を行う場合、ユーザアプリケーションプログラムの作成、特にリフレクティブメモリ上にデータをライトする場合は、細心の注意をすべきである。

## 6. RTP改造推移

1985年のJT-60立ち上げ以降から現在までのRTPの改造理由、および改造項目について以下に述べる。表6にRTP改造年表を記す。

1985年 データ処理設備実時間処理計算機としてPANAFACOM U-1500ミニコンピュータを使用したシステムが導入された。

1986年 プラズマ密度制御にRTPを使用開始。

1994年 システムの老朽化に伴い、ハードウェア、ソフトウェア共に保守することが困難な状態になりつつあったため、UNIX系の計算機を導入。また、他システムへのデータ転送、および将来のデータ収集方法を考慮し、以下のようなシステムを構築した。

①ホスト計算機 コンカレント社製System 7250

(1)CPU MC 68040 2個

(2)主記憶 32 MB

(3)システムクロック 33 MHz

②CAMAC入出力装置

米国KINETIC社製

Model 2140-Z1Aシリアルハイウェイドライバ

③マルチファンクションボード (A/D変換器、およびデジタル入出力装置)

米国PENTLAND社製マルチファンクションボードMPV956

(1)アナログ入力部

分解能 12 bit

入力方式 シングルエンド入力/ディファレンシャル入力

入力レンジ +/-10v, +/-5v

入力チャンネル数 16 ch

A/D変換速度 200 μs / 16 CH

(2)デジタル入力部

入力チャンネル数 16 ch

1997年 1996年度(平成8年度)に実時間帰還制御系開発予定が示され、1

997年度（平成9年度）に4ch、1998年度（平成10年度）に8chのアナログ信号入力の追加が予定されていた。

当時、RTPは、16chのアナログデータ入力部を有し、そのうちの7chを使用中であった。このため、開発予定に示される平成9年度分の追加については当時のシステムのままで対応可能であったが、平成10年度分の追加について、アナログデータ入力系統の増設が必要となった。

ここで懸念されたのが収集するアナログデータの増加に比例したA/D変換所要時間の増加であった。

このため、A/D変換器、およびデジタル入出力装置として使用していた米国PENTLAND社製マルチファンクションボードMPV956をデジタル入出力装置としてのみ使用し、より高速なA/D変換器を2台（32ch分）増設することを検討した。

また、収集するアナログ信号入力の増加に比例し、各々のデータについて演算処理ロジックの高度化が予想された。当時のシステムに於いて実時間処理を実施した場合、1周期中に最も処理時間を要する処理は、全系-RTP間の実時間帰還制御用パラメータデータの転送を行うためのCAMACシリアルハイウェイを用いたDPMID交信（約 $590\mu s$ /1周期）であり、演算処理ロジックをより高度化するためには、1周期中（1ms）における全系-RTP間の実時間帰還制御用パラメータデータの転送処理時間の短縮化が必要不可欠であった。

当時、全系は自システム内にリフレクティブメモリを用いたデータ伝送経路をすでに構築しており、RTP内にリフレクティブメモリを導入し、全系が構築するリフレクティブメモリループに接続することで容易にデータ転送が可能であることが判った。また、データ転送速度についてもCAMACシリアルハイウェイを用いたDPMID交信と比較し数倍高速であると判断したため、CAMA Cシリアルハイウェイを用いたDPMID交信にかかるデータ転送方式として、リフレクティブメモリを用いたデータ転送をRTP内に導入することを検討した。

さらに、演算処理の高度化、A/D変換器、およびリフレクティブメモリの導入を考慮した場合、重要なのはCPUの処理能力であった。とくに演算処理の高速性は、CPUそのものの能力に依存する。このため、上記A/D変換器、およびリフレクティブメモリをシステムに導入することを前提にし、まず第一段階として、これらの処理に耐えうる能力を有するCPUが必要であると判断し、ホスト計算機の入れ替えを実施した。

#### ホスト計算機 コンカレント社製System9200

- |           |            |    |
|-----------|------------|----|
| ①CPU      | MIPS R4400 | 2個 |
| ②主記憶      | 64MB       |    |
| ③システムクロック | 150MHz     |    |

1998年 上記計画に基づきA/D変換器の更新を実施し、同時にアナログ信号入力ch数を16chから32chへ増設。

米国VMIC社製 VMIVME-3123

- ①分解能 16 bit
- ②入力方式 シングルエンド入力／ディファレンシャル入力
- ③入力レンジ +/-10v、+/-5v
- ④A/D変換速度 50μs/16CH

1999年 1997年度（平成9年度）の計画に基づき全系へのデータ転送方式変更のためリフレクティブメモリを導入。

米国VMIC社製A/D変換ボードVMIVME-5576

- ①メモリ容量 256Kbyte
- ②データ伝送方式 光ファイバケーブル

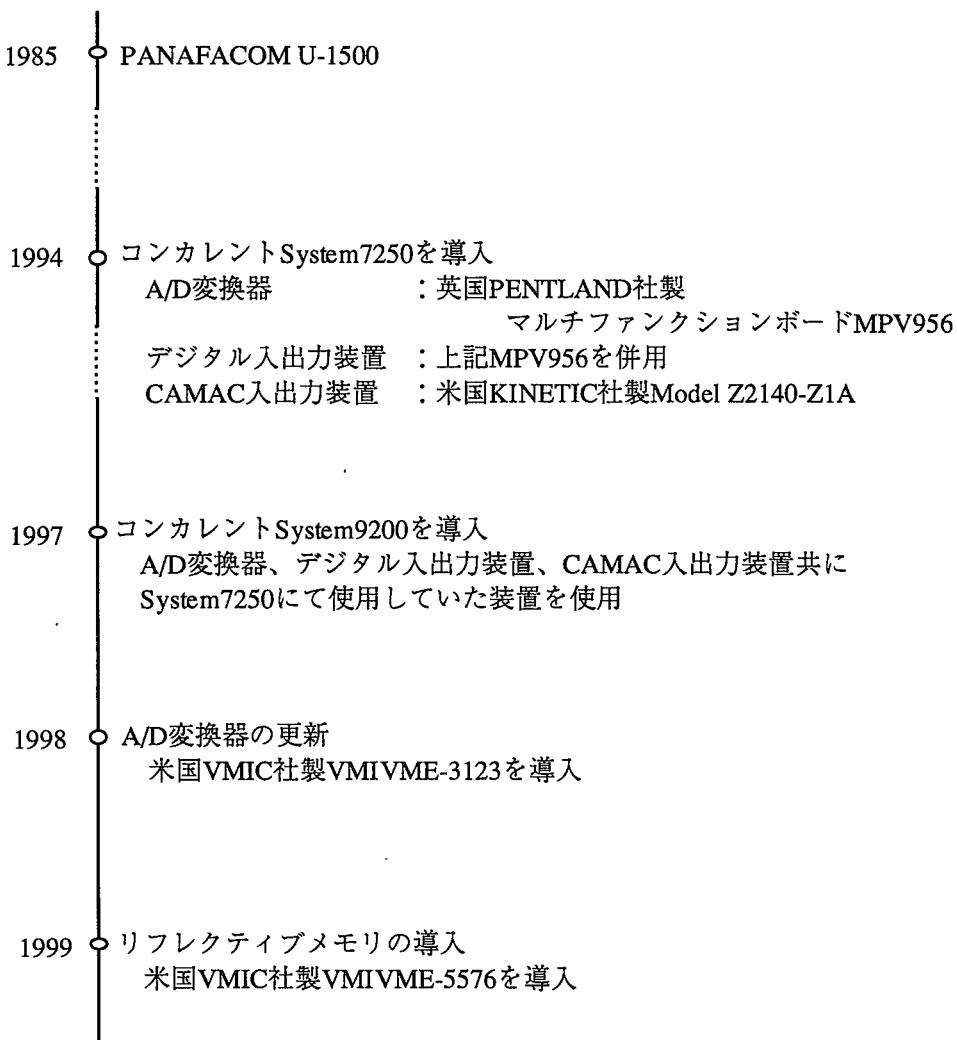


表6 RTP改造年表

## 7. 改造結果

これまでの改造の過程を次の3つに大別し、

- ① 1994年 System 7250導入時
- ② 1997年 System 9200導入時、およびA/D変換器更新時
- ③ 1999年 リフレクティブメモリ導入時

実時間処理の内容を次の3つに大別した場合、

- (a) CAMACアクセス処理
- (b) A/D変換処理
- (c) 演算処理

各々の改造時に於ける実時間処理処理別タイムチャートを図7に記す。

また、処理所要時間測定試験時、JT-60実験計画、および改造作業行程の都合上、実時間処理の内容を同一とした条件下で測定することができなかった。以下に各々の実時間処理処理所要時間測定試験時の処理内容を記す。

- ① 1994年 System 7250導入時

- (1) 演算項目数  
3項目
- (2) DPMOD交信項目数  
1項目
- (3) アナログデータ入力チャンネル数  
7チャンネル

- ② 1997年 System 9200導入時、およびA/D変換器更新時

- (1) 演算項目数  
6項目
- (2) DPMOD交信項目数  
1項目
- (3) アナログデータ入力チャンネル数  
11チャンネル

- ③ 1999年 リフレクティブメモリ導入時

- (1) 演算項目数  
10項目
- (2) DPMOD交信項目数  
2項目
- (3) アナログデータ入力チャンネル数  
23チャンネル

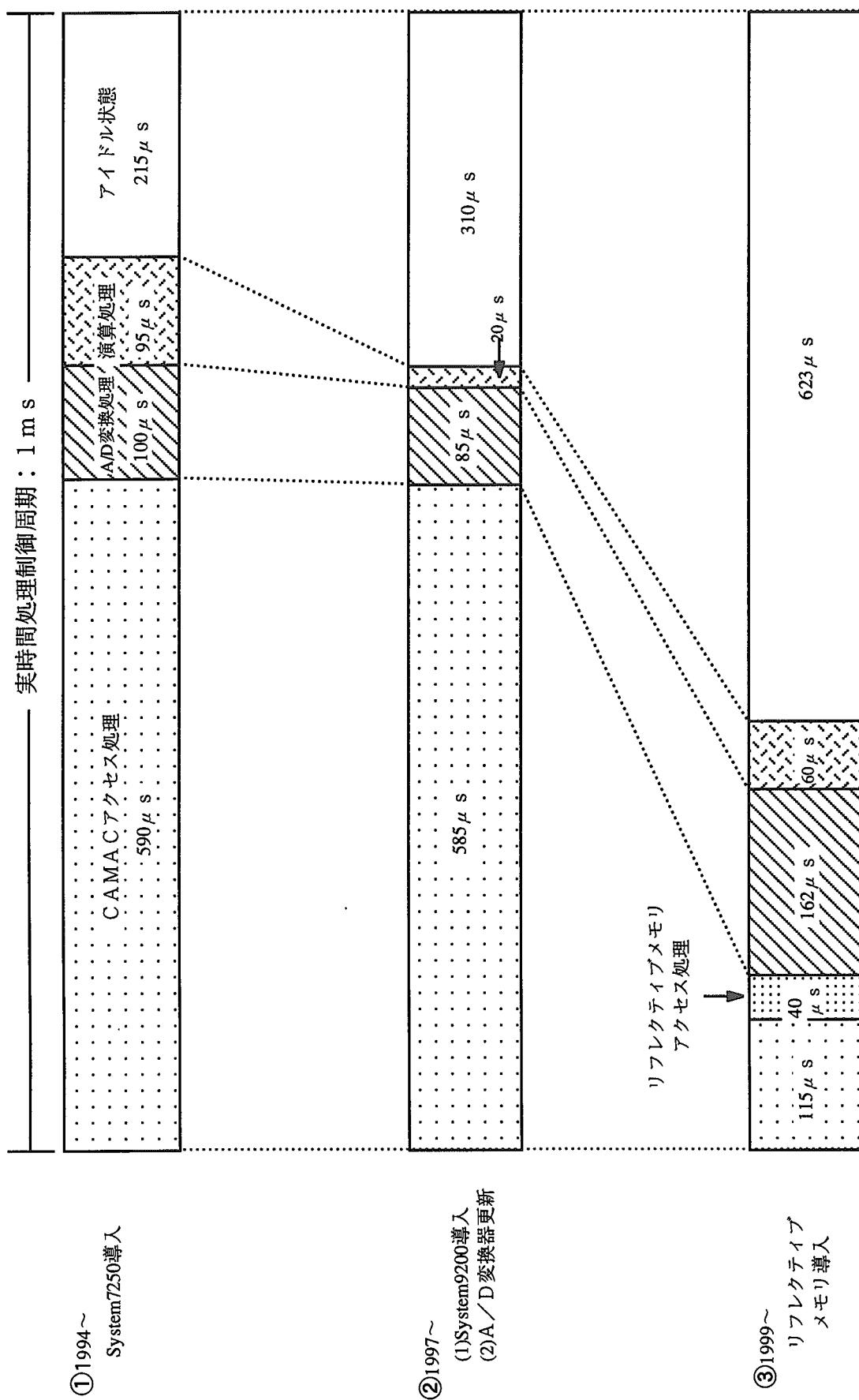


図 7 実時間処理処理別タイムチャート

図7を基に前記(a)～(c)の処理所要時間について述べる。

(a) CAMACアクセス処理

①と②を比較した場合、殆ど差がない。これは、CPUの処理能力に依存する処理ではなく、ユーザアプリケーションプログラムから、CAMAC(ハードウェア)に対してアクセスする固定的な時間が必要となるためである。

このため、全系-RTP間の実時間帰還制御用パラメータデータの転送方式をCAMACシリアルハイウェイを介したDPM-D交信から、リフレクティブメモリを用いたデータの転送方式に置き換えたことにより、全系-RTP間の実時間帰還制御用パラメータデータ転送処理の処理時間を大幅に短縮(約1/4)することができた。

(b) A/D変換処理

①と②を比較した場合、アナログデータ入力チャンネル数が7チャンネルから11チャンネルに増加しているにも関わらず、約1.25倍の処理速度となった。また、②と③を比較した場合、処理所要時間が約2倍となっているが、これは、アナログデータ入力チャンネル数が11チャンネルから23チャンネルに増加したためである。

(c) 演算処理

本処理は、CPUの処理能力に最も依存する処理である。①と②を比較した場合、演算処理項目が3項目から6項目に増加しているにも関わらず、約5倍の処理速度となった。これは、System7250のCPUのシステムクロックが33MHzであるのに対し、System9200のCPUのシステムクロックが150MHzであるためである。

また、②と③を比較した場合、処理所要時間が約3倍となっているが、これは、演算処理項目数の増加(6項目から10項目)、および演算ロジックの高度化のためと推測される。

## 8. まとめ

数段階のステップを経てシステムの改良を実施し、システムの置き換え、実時間処理の高速化を実現した。現システムでは実時間処理の制御周期を1msとしているが、さらに高速化することも十分可能である。言い方をかえれば、より多数の計測データの収集、より高度化した演算処理ロジックを組み込むための余裕が出来たと言える。

JT-60実験放電に於けるRTPの位置付けが実時間帰還制御系の試験装置である以上、現システムはあくまで開発途中のシステムである。今後も要求に対し、出来る限り柔軟に対応し、JT-60の実時間帰還制御系の高度化通信に貢献していく予定である。

## 謝辞

本システムの開発に際し、ご協力頂いた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炉心プラズマ研究部 小関隆久氏、菊池満氏、計算科学技術推進センター 平山俊雄氏に深く感謝致します。

## 参考文献

- 1) Oasa K., Toyokawa R., Mochizuki O., et al.: Fusion Engineering (Proc. 10th Symp., Philadelphia, 1983), IEEE, New York, p.274 (1983).
- 2) 長島 章, 次田友宣, 青柳哲雄 他: 核融合研究59別冊, 303 (1988).
- 3) Neyatani Y., Fukuda T., Nishitani T., et al.: Fusion Eng. Des., 36, 429 (1997).
- 4) Konoshima S., Hosogane N., Sakata S., et al.: Divertor Radiation Feedback Control, in JT-60 Team, Review of JT-60 Experimental Results from February to October, 1995, JAERI-Research 96-018, p.190 (1996).
- 5) Tamai H., Konoshima S., Hosogane N., et al.: Fusion Eng. Des., 39-40 , 163 (1998).
- 6) Chiba S., Kawano Y., Inoue A.: Real-time Processing of CO<sub>2</sub> Laser Interferometer for Density Feedback Control, in JT-60 Team, Review of JT-60 Experimental Results in 1997, JAERI-Research 98-039, p.14 (1998).
- 7) Tamai H., Hosogane N., Konoshima S., et al.: Feedback Control of Divertor Neutral Pressure, in JT-60 Team, Review of JT-60 Experimental Results in 1997, JAERI-Research 98-039, p.16 (1998).
- 8) Fukuda T., JT-60 Team: Fusion Eng. Des., 46 , 337 (1999).
- 9) 福田武司: プラズマ・核融合学会誌 75 , 1377 (1999).
- 10) Takeuchi Y., Tanimori T., Yasu Y.: KEK preprint 92-80, KEK, TITHEP-92-05, TITECH (1992).
- 11) 青柳哲雄: プラズマ・核融合学会誌 72 , 1370 (1996).
- 12) JT-60チーム: JT-60初期加熱実験結果のレビュー (昭和61年8月－11月) , JAERI-M 87-009 (1987).
- 13) コンカレント日本株式会社: RTUのリアルタイム拡張と並列処理
- 14) KineticSystem Corporation: Model 2140-Z1A Enhanced Serial Highway Driver for VMEbus System INSTRUCTION MANUAL
- 15) コンカレント日本株式会社: VMI-5576 光リフレクティブメモリ・ユーザガイド
- 16) コンピュータサイエンス誌: UNIXカーネルの設計



