

JAERI - M  
93-158

VME-busとCAMACシステムを用いた  
JT-60制御用データ伝送システムの開発

1993年8月

戸塚 俊之

JAERI-M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。  
入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11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。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division  
Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-  
mura, Naka-gun, Ibaraki-ken 319-11, Japan.

©Japan Atomic Energy Research Institute, 1993

編集兼発行 日本原子力研究所  
印 刷 いばらき印刷株式会社

VME-busとCAMACシステムを用いた  
JT-60制御用データ伝送システムの開発

日本原子力研究所那珂研究所核融合装置試験部  
戸塚 俊之

(1993年7月15日)

大型核融合実験装置JT-60の制御系におけるデータ収集・伝送システムは、CAMAC規格の機器を用いて構成されている。16ビット・マイクロコンピュータを組み込んだ、本データ収集・伝送システムはその完成から10年近くを経過し、最新の32ビット・マイクロコンピュータを利用したシステムに較べ性能やプログラム開発環境の点で明らかに劣ったものとなっている。この点を改善するため、VME-busモジュールの高速マイクロコンピュータとCAMACシリアルハイウェイドライバを用いたデータ伝送システムを導入することを計画している。これに対してVME-busシステムに組み込まれたマイクロコンピュータ上で動作するCAMACハンドラーの開発、及びシリアルハイウェイドライバの機能確認、データ伝送性能の評価を行った。本開発の結果、プログラム開発が容易で、十分な処理速度を有し、信頼性の高いデータ伝送システムが構築できる見通しを得た。

Development of a VME and CAMAC Based  
Data Acquisition and Transfer System for JT-60 Control

Toshiyuki TOTSUKA

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

(Received July 15, 1993)

Development of a VME and CAMAC based data acquisition and transfer system for JT-60 Control is reported. The present data acquisition and transfer system in JT-60 control is basically composed of CAMAC devices. Since the system equipped with 16-bit microcomputers was manufactured more than ten years ago, the performance and program development environment of the system are apparently worse than those of modern 32-bit microcomputers. To improve these disadvantages, a new data acquisition and transfer system using VME-based 32-bit microcomputers and CAMAC drivers is under design. Corresponding to this design, a CAMAC handler, which runs on the microcomputer, for the VME based CAMAC driver was newly developed. Moreover, the functions of the driver and data transfer performance of the VME and CAMAC complex system were tested. The test results show that the VME based microcomputer and CAMAC serial driver can be applied for the fast and reliable acquisition and transfer system for JT-60 control.

Keywords: VME-bus, CAMAC, Interface, Data Acquisition, Microcomputer Workstation, Ethernet, JT-60

## 目 次

1. はじめに .....	1
2. システム構成 .....	5
2.1 ハードウェア構成 .....	5
2.2 ソフトウェア構成 .....	6
3. CAMACハンドラー .....	7
3.1 動作環境 .....	7
3.2 基本機能 .....	7
3.3 使用上の注意点 .....	17
4. データ伝送試験 .....	18
4.1 試験項目と結果 .....	18
4.2 考察 .....	19
5. まとめ .....	21
謝 辞 .....	22
参考文献 .....	22
付録1 CAMACモジュールアクセスのプログラム例 .....	23
付録2 シリアルドライバのハードウェア設定 .....	25
付録3 シリアル・ドライバの機能概要 .....	27

## Contents

1. Introduction .....	1
2. System Configuration .....	5
2.1 Configuration of Hardware .....	5
2.2 Configuration of Software .....	6
3. CAMAC Handling Program .....	7
3.1 Run Environment .....	7
3.2 Functions .....	7
3.3 Notes .....	17
4. Data Transfer Test .....	18
4.1 Test Results .....	18
4.2 Discussion .....	19
5. Concluding Remarks .....	21
Acknowledgments .....	22
References .....	22
Appendix 1 Examples of CAMAC Access Programs .....	23
Appendix 2 Setup of Serial Driver .....	25
Appendix 3 Functions of the Serial Driver .....	27

# 1. はじめに

日本原子力研究所の臨界プラズマ試験装置 JT-60U は、欧州の JET、米国の TFTR と並ぶトカマク型の大型核融合実験装置である。JT-60U では、主半径 3.4 m、副半径 1.5 m のトーラス形状のプラズマを生成し、最大 6 MA の環状電流を 15 秒間維持する。

このようなプラズマ放電を行った結果収集されるデータ量は、1 ショット当たり約 8 メガバイトに上る。また、JT-60U の実験運転を維持するための各プラント設備からの機器状態データは、1 万数千チャンネルに及ぶ。これらのデータは、以下に示す JT-60 制御系の計算機 CAMAC<sup>\*1</sup> システムにより収集されさまざまな制御に使用されている（第 1.1 図）。

JT-60 制御システムは、プラズマ放電制御機能とプラント制御監視機能を司る複数の設備の制御装置とこれらを統括する「全系制御設備（以下「全系」と称す。）」から構成され、全系の制御用ミニコンピュータ（HIDIC-80E）のバスに接続されたシリアルハイウェイドライバから延びる CAMAC ハイウェイで接続されている。

各設備の制御装置の CAMAC システムには、マイクロコンピュータを搭載した補助コントローラが設けられ、メールボックス方式の通信用モジュールを挿入したコミュニケーションクレートを介して全系との交信を行う。また、このコミュニケーションクレートからそれぞれの設備内 CAMAC ハイウェイが設けられていて、設備内データの収集を行う。

HIDIC-80E で収集されたデータは、プロセス・データや放電結果データのサーバーとしてマン・マシンシステムとのインターフェイスの役割を果たすスーパーバイザ計算機（HIDIC-V90/45）に蓄えられ、イーサネットによるネットワークを通じてマン・マシン用ワークステーション上で参照される。

この JT-60 制御系の計算機・CAMAC システムは、その制作から 10 年以上を経過し、性能上や保守の点から以下の問題が生じてきている。

- ① 各設備制御装置では、Intel8086 や HIDIC-08L 等の 16 ビットのマイクロコンピュータが用いられており、新たな機能追加による性能向上の余地に乏しい。また、プログラムが ROM 化されているものがあるために、制御機能の変更に伴うプログラムの修正に時間を要する。
- ② また、老朽化にともない故障した CAMAC モジュールのメンテナンスに多くの時間と費用を要する。

<sup>\*1</sup> CAMAC: Computer Automated Measurement And Control の略で計算機間通信及びデータ入出力のインターフェイスに関する国際規格である。本規格は、データウェイ規格 (IEEE Std. 583-1982)、シリアルハイウェイ規格 (IEEE Std. 595-1982)、ブランチハイウェイ規格 (IEEE Std. 596-1982)、多重コントローラ規格 (IEEE Std. 675-1982)、ブロック転送規格 (IEEE Std. 683-1976 (R1971))、実時間 BASIC 規格 (IEEE Std. 726-1982)、サブルーチン規格 (IEEE Std. 758-1979 (R1981)) の 7 つの規格からなる。

③ さらに、収集した詳細なデータを設備内部で解析するための道具立て（マン・マシン機能）が不十分でありトラブルシューティングに時間を要する。

一方、最近のエレクトロニクス分野の進歩をみると、32ビット高速プロセッサを登載した高速汎用バス（VME-bus<sup>\*2</sup>）システムがさまざまな分野で普及しつつあり、また、VME-bus上のCPUで動作するいくつかのリアルタイムO/Sが開発され、これらの技術を上手く利用することにより、上述の問題点を解決し要求を満たすことが出来る。

また、ネットワークでは、10Mbit/sの転送速度をもつイーサネット<sup>2</sup>で接続されたワークステーションとVME-busシステム間的高速データ通信が可能となっている。

そこで、現状のCAMACシステムに代わるVME-busによる新しいシステムの構築を考えた場合、一度に全てのCAMACシステムをVME-busシステムに置き換えることは、時間と費用的な面から見て困難である。この解決法の一つとして設備のマイクロコンピュータの部分とCAMACシステムのシリアルハイウェイを接続するシリアルハイウェイドライバの部分<sup>3</sup>をVME-bus化することにより、CAMACシステムのI/OモジュールをVME-busシステムのCPUで制御することが出来る。即ち、一例として、第1.2図に示すような新たな通信システムが想定出来る。

この様な対応により、前述の問題点の①に関しては、イーサネット<sup>2</sup>で接続されたワークステーションをホスト計算機として、高級プログラム言語であるC言語または、FORTRAN等でソフトウェアのクロス開発が可能（セルフ開発のシステム構築も可能）である。また、ワークステーションの編集機能を使い容易にソフトウェアの変更、修正が可能となり、開発環境の向上が図れる。また②に関しては、CAMACモジュールの削減、構造の比較的簡単なI/Oモジュールだけになる等により、メンテナンスに要する時間及び故障の頻度が軽減出来る。さらに③に関しては、VME-busにて収集されたデータをネットワーク上のワークステーションに蓄積することにより、ワークステーションの高度な演算機能、グラフィック機能により、さまざまなデータ解析が可能となる。以上の様に設備では、機能面及び、メンテナンス性の面で大幅な強化が図れることになる。

しかし、全系の計算機システム、HIDIC-80Eには、イーサネット<sup>2</sup>等を使ったネットワークは接続出来ない。このため、HIDIC-80Eは、VME-busシステムと直接交信が出来ない。また、HIDIC-80Eのシステム使用率の点から見ると現状以上の機能開発、追加の余裕がない等の問題がある。これを解決するためには、HIDIC-80Eのソフトウェアを完全にワークステーションやVME-busのマイクロコンピュータで動作するソフトウェアに置き換える必要がある。しかし、この方法を実現するためには、各設備が一斉に前述のVME-bus化をおこなわなければならないことや、HIDIC-80Eのソフトウ

\*2 VME-bus: Versa Module Europe-bus の略で、IEEE P1014/D1.2 により規定された高速の32 bit バス規格である。

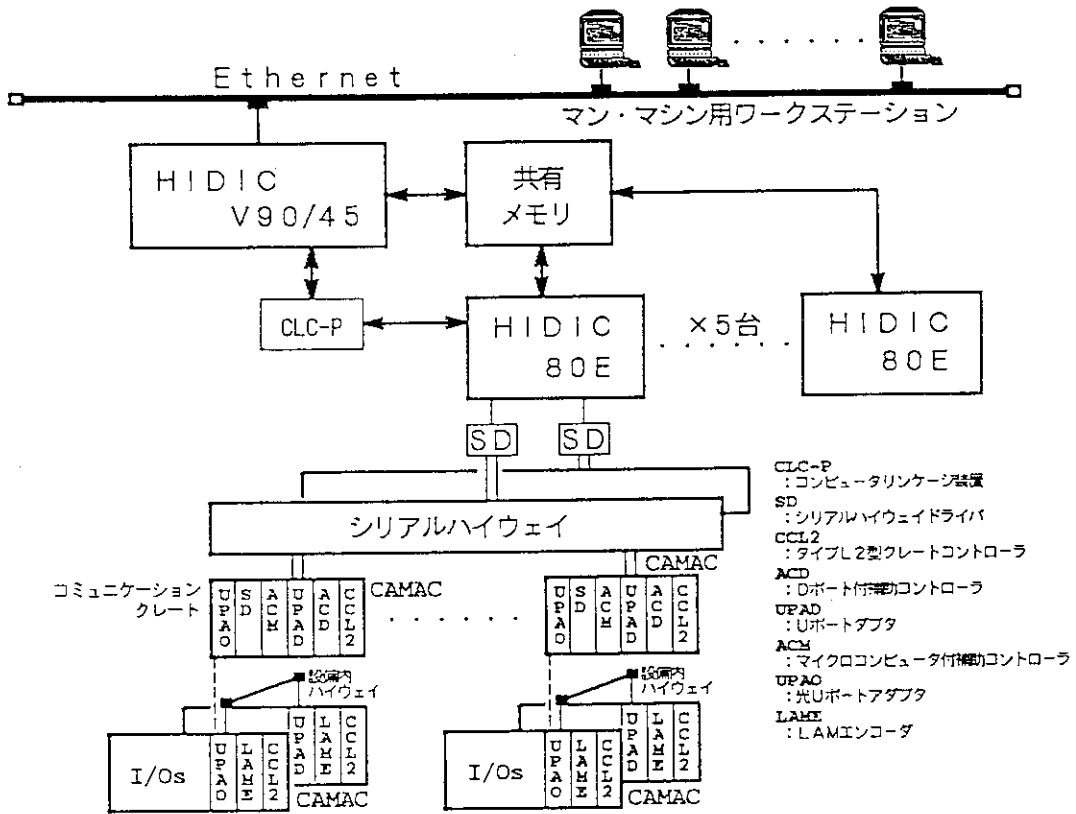


ウェアを移行するための開発に多くの時間と費用が必要である等の問題がある。

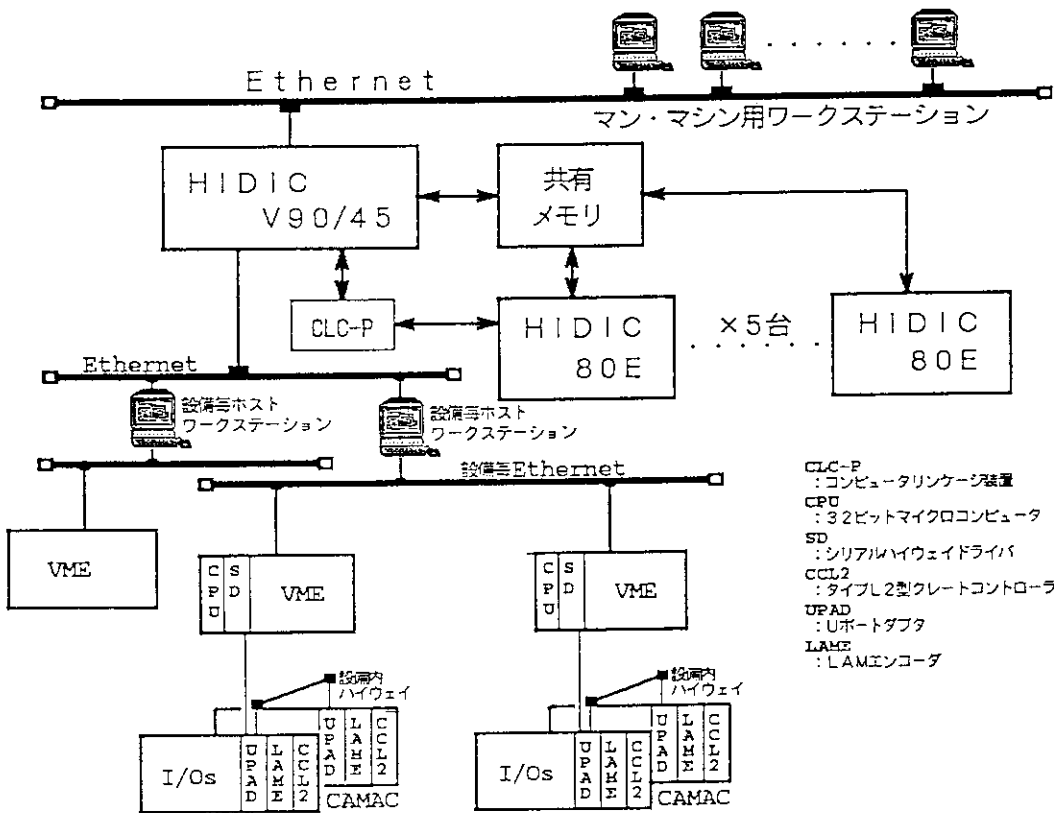
このためにもう一つの解決方法として、全系のマン・マシンシステムとのスーパーバイザであるHIDIC-V90/45を使用することが考えられる。HIDIC-V90/45は、イーサネットによるネットワーク通信及びグローバルメモリとコンピュータリンク装置を用いたHIDIC-80Eとの通信の両方を行える。そこで、各設備のVME-busシステムとの通信は、HIDIC-V90/45にて実施することとし、HIDIC-80E側の機能を順次移行して行く様にすればHIDIC-80Eに関する問題も解決する。

以上の観点からワークステーションとVME-busのモジュールである32ビットマイクロコンピュータ及び、シリアルドライバ(米国キネティック社製 SD-2140-Z1A)を用いてシリアルハイウェイで接続した試験システムを構築し、シリアルドライバの機能調査を行うと共にシリアルドライバを動作させるためのCAMACハンドリングプログラムを作成し、伝送性能の確認を行った。

第2章では、本試験システムのハードウェア構成及び、ソフトウェア構成を、第3章では、シリアルハイウェイドライバを動作させるためのCAMACハンドラーの機能を、第4章で試験結果と考察を述べる。最終章で、まとめを述べる。



第1.1図 現状の通信システムの概要



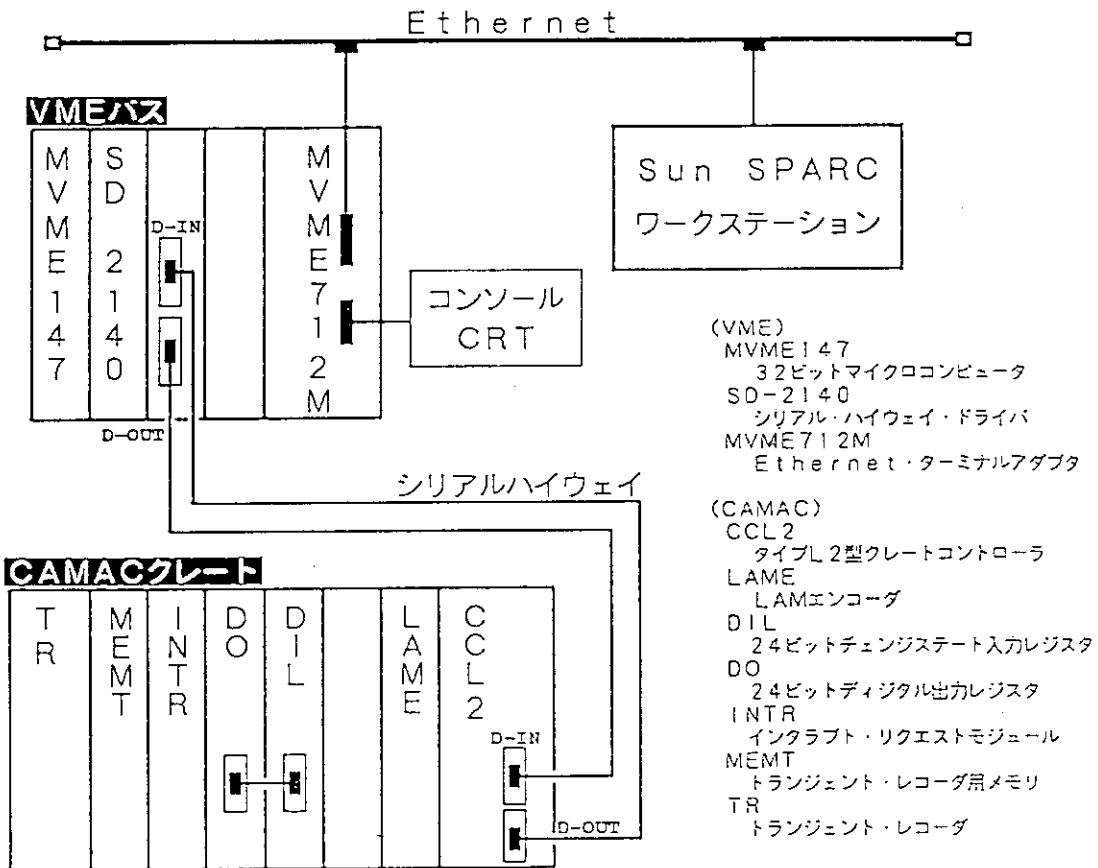
第1.2図 新通信システムの概要

## 2. システム構成

### 2.1 ハードウェア構成

本試験システムは、第2.1図に示すようにイーサネットに接続されたワークステーションとVME-busシステム及び、VME-busシステムとシリアルハイウェイで接続されたCAMACシステムからなる。

ワークステーション上で開発されたプログラムは、イーサネットを通じてVME-busシステムのCPUにダウンロードされる。VME-busシステムのCPUは、与えられたプログラムを実行してシリアル・ドライバからCAMACファンクションを各CAMACモジュールに発行する。本試験用システムでは、CAMAC入出力モジュールとして、DO（24ビットデジタル出力レジスタ）、DIL（24ビットチェンジステート入力レジスタ）、INTR（インタラプト・リクエストモジュール）、TR（トランジェント・レコーダ）を用い、それぞれのモジュールにVME-busシステムからCAMACコマンドを発行し、データ入出力の確認を行った。



第2.1図 シリアルハイウェイ・ドライバ試験システム

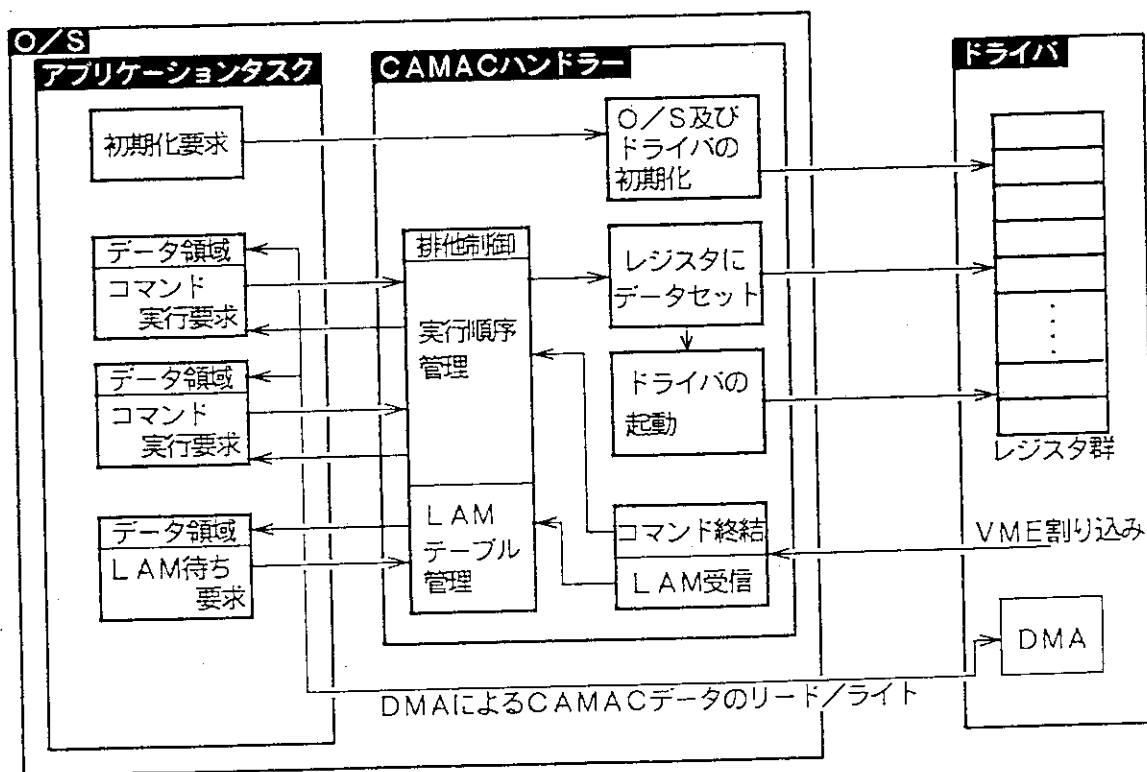
DO/DILについては、VME側からDOに書き込まれた24ビットデータをDILにて入力し、これをVME側に取り込むリードバック試験、INTRについては、INTRが発生したLAMをVME側にて受信出来ることの確認試験、さらにTRにて収集されたデータをブロック転送にてVME上のシリアル・ドライバに取り込みDMA(Direct Memory Access)機能を使用してCPUのメモリに書き込めることの確認等が行えるものとした。

## 2.2 ソフトウェア構成

本試験システムのソフトウェアは、第2.2図に示すように、VME-bus上のCPUで動作するO/S(オペレーティング・システム)の管理下に組み込まれたCAMACハンドラーとアプリケーションタスクから構成される。

アプリケーションタスクは、CAMACハンドラーに用意された各々のサブルーチンを使用した複数のタスクがCAMACファンクションの実行やLAM待ち要求を行う。

CAMACハンドラーは、複数のアプリケーションタスクからの処理要求に対して実行順序管理等の排他制御を行いシリアル・ドライバの競合使用を防ぐと共に、アプリケーションタスクから渡されたCAMACファンクションをシリアル・ドライバの各レジスタの設定し、シリアル・ドライバを起動する。また、VME割り込みにより受信したCAMACからのLAM信号をテーブル登録し、LAM待ち要求に対する配信処理等の制御を行う。



第2.2図 ソフトウェア構成

## 3. CAMACハンドラー

### 3.1 動作環境

本CAMACハンドラー（以下本ハンドラーと称す）は、VMEバス上のCPU（MVME 147S）からシリアル・ドライバ（SD-2140-Z1A）を通じてCAMACシリアルハイウェイ上のCAMACクレートに実装されている各種モジュールにアクセスするためのシリアル・ドライバ・インターフェイス・ルーチンである。

VME-busシステム上のユーザアプリケーションからシリアル・ドライバそのもののハードウェアを意識することなく、本ハンドラーを用いることにより、容易にCAMACシステムを動作させるためのソフトウェアの構築が出来る。

本ハンドラーの動作環境は、CPUとして、32ビットプロセッサ（MC68030）を搭載したモトローラ社製のMVME147シリーズを、また、リアルタイムO/Sとしては、Wind River Systems, Inc. のVxWorks Version 5.0である。

### 3.2 基本機能

本ハンドラーは、ハードウェア及びソフトウェアの初期化、CAMACコマンドの実行要求、LAM待ち要求の3つの基本機能から成るものとした。また、これらの機能をサポートするためのいくつかの機能を備えることとした。第3.1図にサブルーチン構成を示す。

#### 3.2.1 ハードウェア及び、ソフトウェアの初期化処理

##### (1) サブルーチン仕様

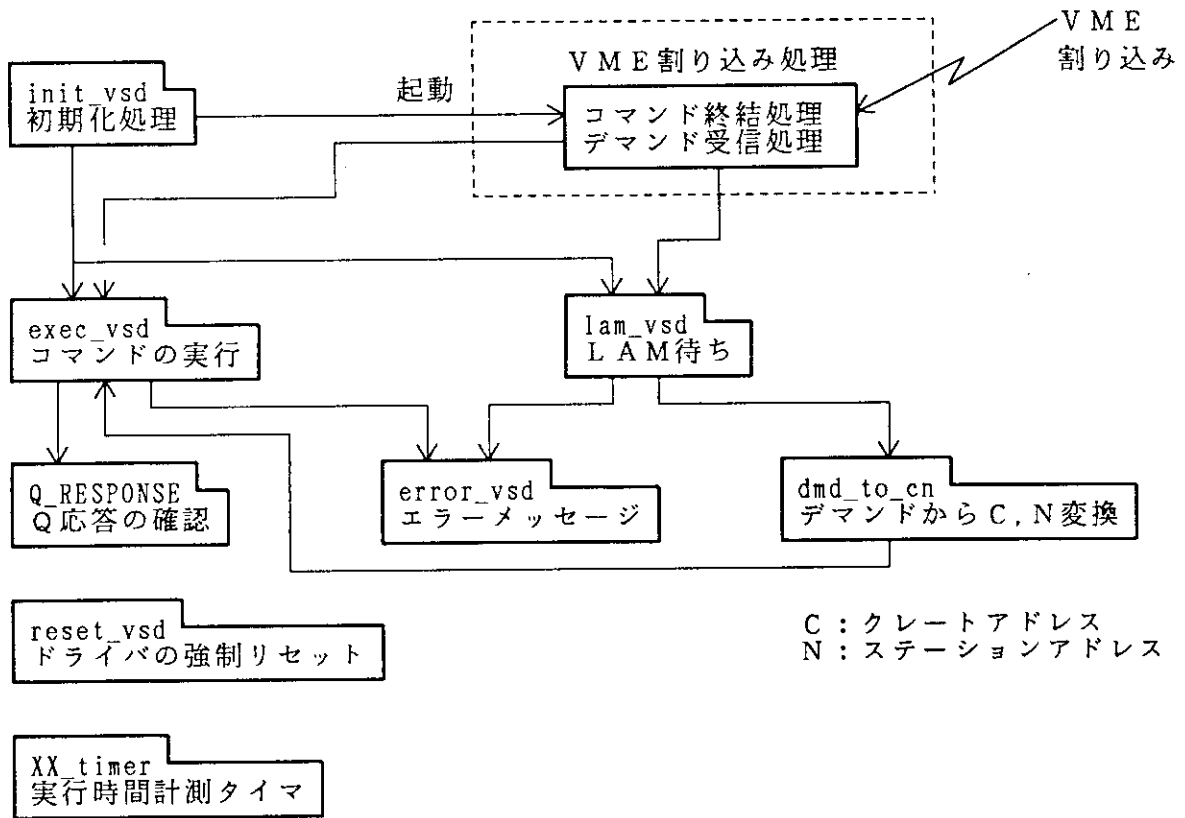
```
init_vsd()
```

引数、リターンコード無し。

##### (2) 処理詳細

CPU（MVME147S）、シリアルドライバ（SD-2140-Z1A）の各レジスタを初期化すると共にソフトウェアの動作環境を整える。

システム立ち上げ時の必ず本処理を1度実行する必要がある。これを行わないと以下CAMACコマンドの実行が出来ないものとした。



第3.1図 CAMACハンドラーのサブルーチン構成

### 3. 2. 2 CAMACコマンドの実行要求

#### (1) サブルーチン仕様

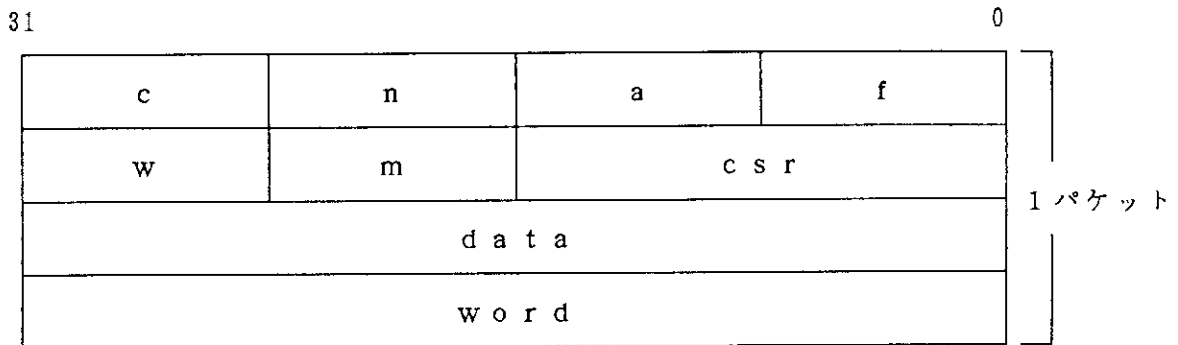
exec\_vsd(n, list)

##### ① 引数

int n : 処理すべきCAMACコマンドの数 (IN)

struct pram\_t \*list : 構造体(pram\_t) の先頭アドレス (IN)

##### ② 構造体(pram\_t)



```

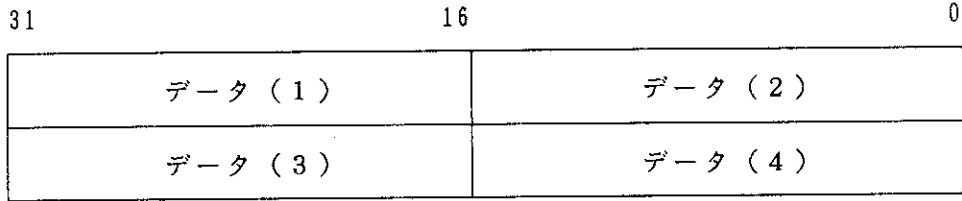
struct pram_t {
    u_char c, n, a, f; ・・・CAMACコマンドのクレート番号、ステーション番号、
                        サブアドレス、ファンクション (IN)
    u_char w;          ・・・データ幅。転送モードmが0以外の時のみ有効 (IN)
                        0 : 16ビット、1 : 24ビット
    u_char m;          ・・・転送モード
                        0 : シングルアクション
                        1 : Q-STOP
                        2 : Q-IGNOR
                        3 : Q-REPEAT
    u_short csr;      ・・・SD-2140-Z1AのレジスタCSR(Channel
                        Status Register)の上位16ビット (OUT)

    union{
        caddr_t p; ・・・ブロック転送時のデータ格納先頭アドレス
        u_long d; ・・・シングルアクション時のデータ値
    }data;          CAMACデータライト時 (IN)
                    CAMACデータリード時 (OUT)
    u_long word;    ・・・ブロック転送時の転送データ語数 (IN)
                    ブロック転送によって転送されたデータ語数 (OUT)
}
    
```





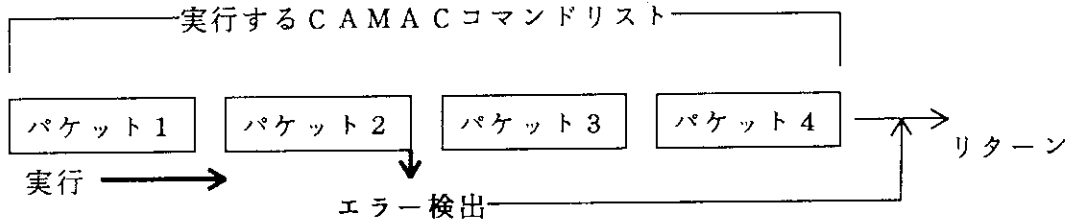
(16bitデータ)



16ビットデータのリードで転送語数が奇数の場合、最後の転送データが入った4バイトの下位2バイトには、1が詰められる。

④ コマンドの実行処理

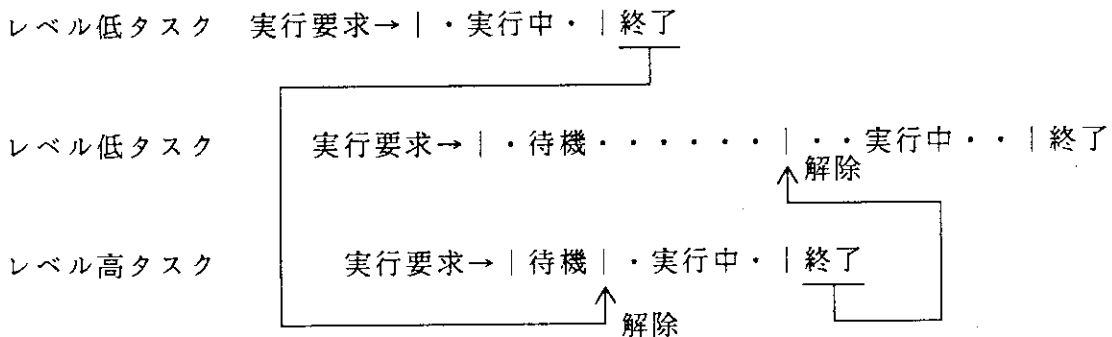
本処理では、処理すべきCAMACコマンドの数(n)で指定された分のCAMACコマンドをパッケージから読み込み順次実行する。実行途中にリターンコードで示すエラーを検出した場合、処理を中断し、ただちに終了する。



⑤ マルチタスクでの動作

本処理は、マルチタスクでの動作を前提とし、設計した。従って複数のタスクから同時に実行要求が行われた場合、先着優先とし実行中のCAMACコマンドリストが終了するまで、次の実行要求は、ブロックされ、シリアル・ドライバ・ハードウェアをタスク間で競合使用することを防いでいる。

また、タスクレベルとCAMACコマンドリストの実行順序の関係は、実行要求タスクのうち、タスクレベルの高いものから順次ブロックが解除されて実行する。



## ⑥ Q 応答の確認

シリアル・ドライバ・ハードウェアでは、Q 動作モードをハードウェアが制御しているため、Q 応答の結果（0、1）をソフトウェア上に取り込まない。

そこで、CAMAC コマンドの実行結果としてかく、各パッケージに格納された `csr` (Channel Status Register) の内容を入力とし、Q 応答結果に変換するマクロ命令を用意した。

## ・マクロ名

`Q_RESPONSE (csr)`

## ・使用例

```
task_A()
{
  struct pram_t pkt;
  int   rt;
  int   q;
  rt = exec_vsd(1, &pkt);
  q  = Q_RESPONSE(pkt.csr);
}
```

この例では、`pkt` に格納された CAMAC コマンドを実行し、その Q 応答の結果を変数 `q` に格納する。

### 3. 2. 3 LAM待ち要求

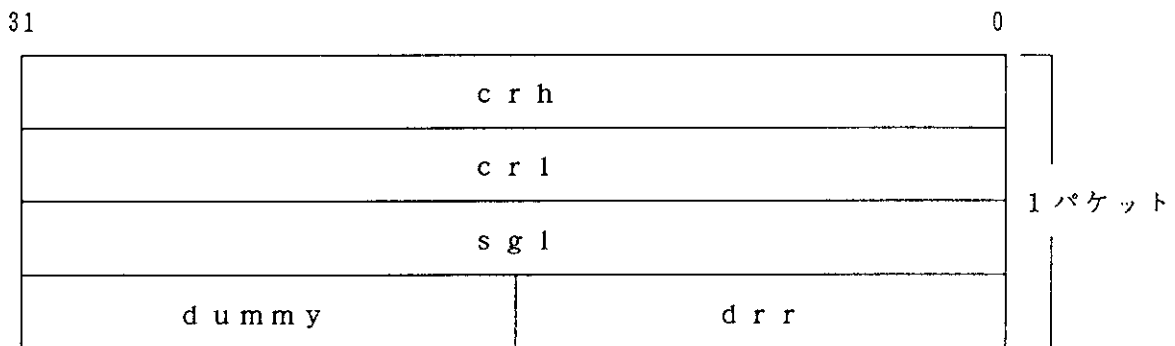
#### (1) サブルーチン仕様

lam\_vsd(rec, dmd, tmo)

#### ① 引数

int                   rec : 処理要求 1 : LAM待ち要求、2 : LAMクリア (IN)  
 struct dmd\_t       \*dmd : 構造体(dmd\_t) の先頭アドレス (IN)  
 int                   tmo : LAM待ち時間 0 : 無制限、0 < tmo : 秒 (IN)

#### ② 構造体(dmd\_t)



```

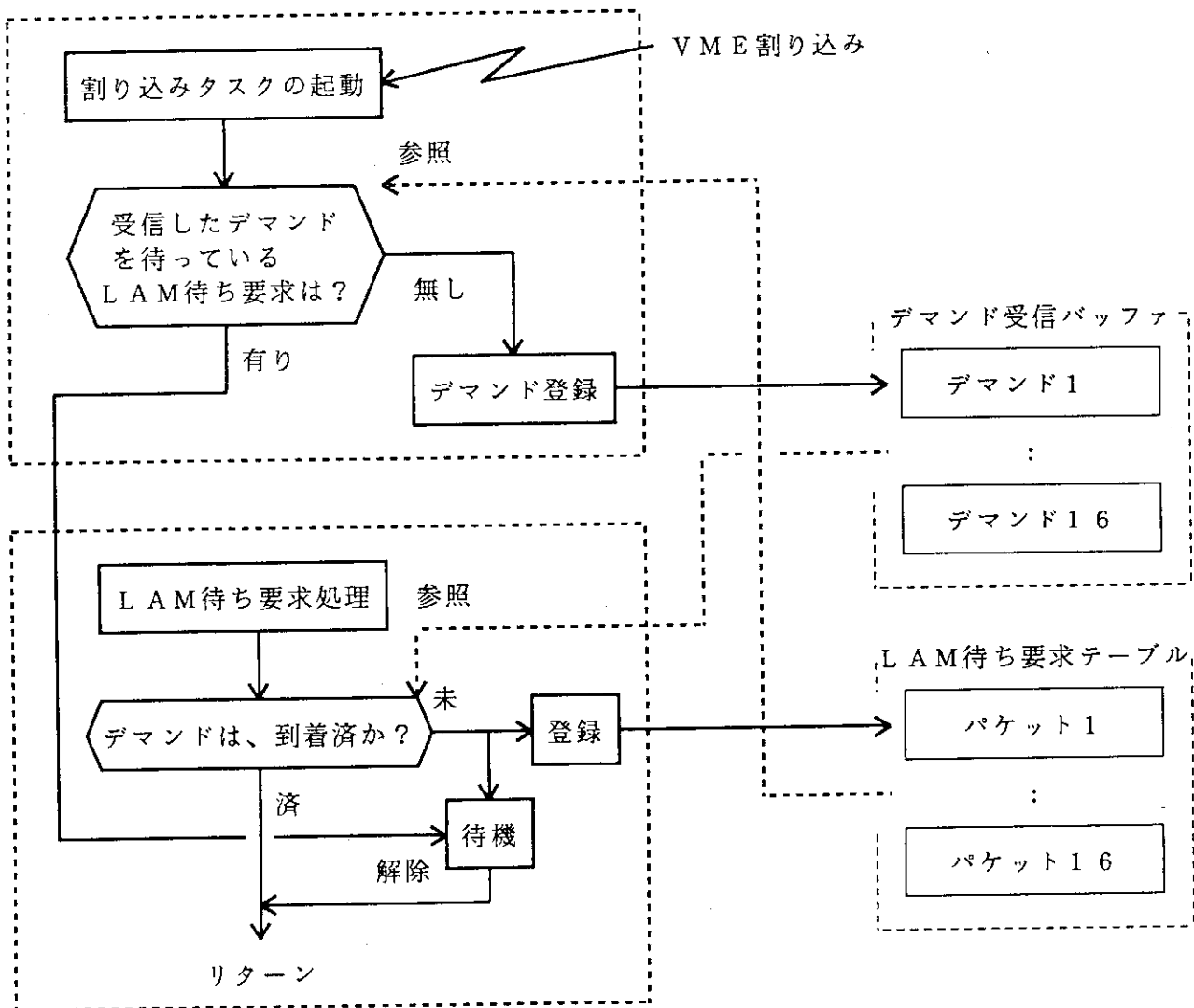
struct dmd_t{
    u_long   crh;   ... クレート番号62～32のどのクレートからのデマンドを
                  処理するか否かを表すビット列(MSBは、無効)(IN)
                  各ビット 1:処理、0:無視
    u_long   crl;   ... クレート番号31～1のどのクレートからのデマンドを
                  処理するか否かを表すビット列(LSBは、無効)(IN)
                  各ビット 1:処理、0:無視
    u_long   sgl;   ... エンコード化されたSGLのうちどのステーションアドレ
                  スのデマンドを処理するかを指定する。(IN)
    u_short  dummy;
    u_short  drr;   ... 受信したデマンドメッセージ(OUT)
}
    
```

#### ③ リターンコード(rt)

- 0:                   正常
- 11:ERROR\_LAM\_REC 処理要求が異常
- 12:ERROR\_LAM\_TMO LAM待ち時間がタイムアップした。
- 13:ERROR\_LAM\_SP 未処理のLAM待ち要求が16個を越えた。

#### (2) 処理詳細

LAM待ち要求は、lam\_vsd()をユーザアプリケーションからコールすることにより、crh, crl, sgl で指定されたデマンドメッセージを取り込む。LAM待ち要求は、同時に16個まで受付可能とした。第3.2図にLAM待ち要求の処理概略を示す。



第3. 2 図 LAM待ち要求の処理概略

② LAM割り込み受信（シリアルドライバからのVME割り込み）

シリアルドライバが受信したデマンドメッセージは、VME割り込みを通じてCPU（MVME147）のデマンド受信バッファに格納される。

デマンド受信バッファは、サイクリックに使用される16個のデマンドフィールドからなりLAM待ち要求が無い場合、LAM待ち割り込み処理がハングデマンド以外のデマンドを登録する。

これにより、LAM待ち要求以前に発生したデマンドの取りこぼしを防ぐ。

また、処理要求に 2 : LAMクリア を指定した場合、デマンド受信バッファに格納されている CRH, CRL, SGL で指定された既に到着していたデマンドを削除する。

## ③ マルチタスクでの動作

本処理は、マルチタスクでの動作を前提とし、設計した。従って複数のタスクから同時にL A M待ち要求が行われた場合、L A M待ち要求テーブルに指定されたパケットを登録すると共にL A M待ち割り込みタスクが指定されたデマンドを受信するまでまたは、指定された時間（秒）待機する。

また、それぞれの要求に対して既にデマンド受信バッファに要求デマンドが格納されていた場合、直ちに終了する。

## ④ デマンド受信バッファのフォーマット

デマンド受信バッファは、シリアルド・ライバのデマンドリクエストレジスタが受信した値そのものが格納される。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	S G L				0	0	C R A T E						

S G L : コード1～1 F、コード1 Fは、ハングデマンド。

C R A T E : 1～6 2

## 3. 2. 4 その他の機能

## (1) ドライバ強制リセット

何らかの異常でシリアル・ドライバがV M Eバスを解放しない場合等に使用する。

```
rt = reset_vsd()
```

- ・引数無し
- ・リターンコード ( r t )
  - 2 : シリアル・ドライバのD M A転送においてバスエラーが発生した。
  - 3 : D M A転送が終結していなかった。

## (2) デマンドからC, Nへ変換

L A M待ち要求で取り込んだデマンドをC A M A Cコマンドの実行要求パケットのc、nに変換して設定する。

```
dmd_to_cn(dmd, pkt, n)
```

- ・引数
  - struct dmd\_t \*dmd : 構造体(dmd\_t)の先頭アドレス ( I N )
  - struct pram\_t \*pkt : 構造体(pram\_t)の先頭アドレス ( O U T )
  - int n : 構造体(pram\_t)の数 ( I N )
- ・リターンコード無し

## (3) エラーメッセージ表示

CAMACコマンドの実行要求及び、LAM待ち要求のリターンコード及び、CAMACコマンドの実行要求パケットに格納されたcsrを入力としてそれぞれに対応した簡単なエラーメッセージを接続されたコンソールに表示する。

```
error_vsd(rt,csr,text)
```

## ・引数

```
int          rt  : コマンド実行要求、LAM待ち要求の実行結果リターンコード  
              (IN)
```

```
u_short     csr  : コマンド実行要求結果パケットのcsr (IN)
```

```
char        *text : エラーコードの他に表示したい文字列変数の先頭アドレス (IN)
```

## ・リターンコード無し

## (4) 実行時間計測タイマ

MVME147のハードタイマの起動、停止を行いタスクの実行時間及び、異なったタスク間同士の処理時間を6.25マイクロ秒単位で計測表示する。

計測範囲は、6マイクロ秒～1.6384秒とする。

```
start_timer() : タイマーを起動する。
```

```
stop_timer()  : タイマーを停止する。
```

```
print_timer(text): タイマー値を表示する。
```

## ・引数

```
char        *text : タイマー値の他に表示したい文字列変数の先頭アドレス (IN)
```

### 3. 3 使用上の注意点

- (1) LAM待ち要求を行うタスクは、必ず他のCAMACコマンドの実行要求を行うタスクより、タスクレベルを高くしておかないとCAMACコマンドの実行要求タスクがLAM受信処理の動作を妨げる。

VxWorksのタスクレベルは、高0～255低となっている。

システムを構築する上でこの調整をうまく行うことが必要である。

- (2) 本プログラムをシステムに組み込む場合、システムに応じた以下①～⑤のパラメータを変更することが出来る。変更箇所は、ヘッダファイル(VSD.h)で定義されている。

#### ① VMEアドレス

ハードウェアで設定したシリアルド・ライバのVMEアドレスと統一させること。

```
#define SD_ADDR 0xeffff00
```

#### ② LAM待ちテーブルの数

16以上必要時、変更する。但し、極端に多くすると割り込み処理のレスポンスが遅くなるので注意すること。

```
#define MAX_LAM 16
```

#### ③ デマンド受信バッファの数

16以上必要時、変更する。但し、極端に多くすると割り込み処理のレスポンスが遅くなるので注意すること。

```
#define MAX_DRR 16
```

#### ④ シリアル・ドライバから見たCPUメモリの24ビット境界以上アドレスとAMコード

MVME147のDRAMの先頭VMEアドレスが0X04000000であることを表す。

0d は、AMコード。付録3 5.(1). ⑧ レジスタAMRを参照。

```
#define AMR_INIT ((u_long)0x0000040d)
```

#### ⑤ VME割り込みレベル

シリアル・ドライバが発生する各割り込みのレベルを設定する。初期値はレベル5。

```
#define INTR_LEVEL 0x05
```

## 4. データ伝送試験

### 4.1 試験項目と結果

#### (1) 試験項目

本システムのデータ伝送速度の性能評価に関して以下のような試験を実施した。

シングルアクションCAMACコマンドの入出力機能の健全性の確認のために①～④、また、ブロック転送におけるDMA機能の健全性の確認のために⑤を実施した。さらに、⑥では、CAMACクレート内で発生したLAMの受信機能の確認を行うと共にそれぞれの実行時間を計測を行った。

- ① シングルアクション時の1CAMACコマンドの実行時間(INLINE WRITE)
- ② シングルアクション時の連続5CAMACコマンドの実行時間
- ③ シングルアクション時の連続10CAMACコマンドの実行時間
- ④ シングルアクション時の連続20CAMACコマンドの実行時間
- ⑤ ブロック転送時の3Kワード転送の実行時間
- ⑥ CAMACクレート内で発生したLAMをアプリケーションが取り込むまでの時間

#### (2) 試験結果

上記(1)の各項目をバイトシリアルモード、ビットシリアルモードでそれぞれの実行時間を計測した結果を第4.1表に示す。

尚、ハイウェイクロックは、5MHzで試した結果である。

第4.1表 実行時間計測結果

No.	モード		
	処理		
1	1コマンド	約120 $\mu$ s	約260 $\mu$ s
2	5コマンド	約520 $\mu$ s	約1200 $\mu$ s
3	10コマンド	約1000 $\mu$ s	約2540 $\mu$ s
4	20コマンド	約2100 $\mu$ s	約4950 $\mu$ s
5	ブロック3Kワード	約15ms	約138ms
6	LAM受信	200 $\mu$ s以内	350 $\mu$ s以内



## 4. 2 考察

### (1) ブロック転送について

No. ⑤の試験結果からDMAを使用したブロック転送における1語当たりの転送時間が求められる。即ち、全体の実行時間からハンドラーおよびO/Sのオーバーヘッドを引き、転送語数で割る。このようにして得られたバイトシリアル/ビットシリアルそれぞれの転送モードの1語当たりの実行時間は、 $5\mu\text{s}/46\mu\text{s}$ である。このうち、バイトシリアル時の実行時間は、ほぼカタログ値<sup>\*3</sup>の性能に等しい。従って、これらはシリアル・ドライバのハードウェアとしてのそれぞれの転送モードでの最高のスループットと考えられる。

### (2) シングルアクションについて

シングルアクションでの1語当たりの転送時間を以下の考え方で求める。No. ①~④の試験における実行時間は、語数Nに比例する時間とオーバーヘッド時間の和として次に示す式に従うものと仮定する。

$$T = aN + b$$

T : 実行時間

a : 1語転送に要する時間

N : CAMACコマンドの数

b : 本ハンドラーのオーバーヘッド

これをNo. ①~④の試験結果に適用すると、バイトシリアル/ビットシリアルの1語当たりの転送時間は、 $104\mu\text{s}/245\mu\text{s}$ となる。この係数aは、シリアル・ドライバのハードウェアとしての1語当たりの転送時間とO/Sが転送終了を通知する割り込み処理受信後に本ハンドラーの待機状態を解除するために費やす時間の和である。また、オーバーヘッドbは、本ハンドラーのCAMACコマンド実行要求処理を呼び出し、この処理がシリアル・ドライバの各レジスタを初期化するのに要する時間であり、 $20\mu\text{s}$ である。

シングルアクション時のバイトシリアル転送モードでのシリアル・ドライバのハードウェアとしての1語当たりの転送に要する時間として、カタログ値 $5.5\mu\text{s}$ を採用すると、O/Sが割り込み処理受信後に本ハンドラーの待機状態を解除するために費やす時間は、 $98.5\mu\text{s}$ となる。これをビットシリアル転送モードに適用し、1語当たりの転送に要する時間から差し引くとビットシリアルモード時のシリアル・ドライバのハードウェアの1語当たりの転送に要する時間は、約 $150\mu\text{s}$ となる。しかし、本来ビットシリアルでの転送所要時間は、バイトシリアルでのそれに較べ10倍になるはずであり、この値 $150\mu\text{s}$ は、

\*3 シリアル・ドライバのインストラクションマニュアルによるとハードウェアのスループットは、ハイウェイクロックが5MHz、バイトシリアルモードの場合、シングルアクションコマンド及びブロック転送では $4.5\mu\text{s}/1$ コマンド、インラインライトでは $5.5\mu\text{s}/1$ コマンドである。

ビットシリアル転送モード時の予想値  $55 \mu s$  に較べて大きすぎる。そこでシングルアクション時の本ドライバの処理として、転送モードに依存するコマンド送出準備時間がさらに含まれていると考えるのが妥当である。このような考えを基に、バイトシリアル/ビットシリアル転送モードでのドライバのハードウェアとしての転送所要時間を求めるとそれぞれ約  $15 \mu s / 150 \mu s$  となり、ソフトウェアの転送終了割り込み処理は、両方とも約  $90 \mu s$  となる。従って、シングルアクション転送時でのデータ送出準備に要する時間は、DMAを使用したブロック転送の結果から得られた送出時間を差し引いて、バイトシリアル/ビットシリアルそれぞれ  $10 \mu s / 100 \mu s$  と推定できる。

今後、本伝送システムの応用に関しては、より高速性を必要とする様な制御に使用する場合、ハードウェア (CPU) のグレードアップやO/Sのバージョンアップにより、数十  $\mu s$  での動作が可能と思われる。但し、上記で述べたようにビットシリアルモードでは、バイトシリアルモードのようにCPUやO/Sの変更による飛躍的な性能向上は望めない。

しかし、本プログラムを現状のJT-60U各設備のプラント監視機能や放電制御機能に利用する場合に於いては、リレーの動作速度や制御周期の面から見てビットシリアルモードでも十分に対応可能な伝送処理速度である。

## 5. まとめ

高速・高演算処理が実現可能であるVME-busシステムを既存のCAMACシステムに組み込むための前準備として、試験システムによるシリアルハイウェイドライバの機能確認及びCAMACハンドラーの開発を行い、ハードウェア/ソフトウェアの両面からの実用性の確認をおこなった。

本開発の結果、十分な処理速度を有し、信頼性の高いデータ伝送システムが構築できる見通しを得た。

そこで第1章にて述べた問題について、実用的な面から考察すると、プログラム開発環境の点については、ワークステーションを使ったプログラム開発が出来る事により、設備内の機能改良及び、さまざまなニーズに柔軟かつ迅速に対応出来る開発環境になる。次に老朽化の点については、構造の簡単なI/Oモジュールだけになることによるメンテナンスに要する時間の軽減をはじめ、全系との通信手段をイーサネット化することによりネットワークの信頼性が向上する。また、今後さらに新しい設備を追加する様な場合に於いても簡単に接続出来ることになる。さらにマン・マシン性の点については、ワークステーションの優れたマン・マシン性、高機能を利用したトラブルシューティング用ソフトウェアを開発することにより、異常時の対応やメンテナンスが容易になる。なお、プログラム設計・開発にあたっては、日進月歩で高性能化されるワークステーションのアップグレードや機種の変更等に柔軟に対応出来るような考慮が必要である。

この様にVME-busシステムとCAMACシステムの通信方法を確立することにより、現在の資源を有効に活用しつつJT-60U制御システムのアップグレード化に対応可能な信頼性、操作性、メンテナンス性に優れたシステム開発の見通しを得る事が出来た。また、一部設備では、これらを使用したシステムへ移行するための改造に取りかかっている。

今後のJT-60Uの制御システムを考えるとCAMACを使ったシステムを現状のまま維持することは、拡張性、柔軟性、また、経済性の面からみて不利であり、順次VME-busシステムへと移行することが重要な課題である。

## 謝辞

今回、本試験システムの開発に際し、VME-bus及びCAMACシステム全般にわたり数々の技術的助言及び協力をして頂いた日本原子力研究所炉心プラズマ研究部炉心プラズマ解析室 青柳 哲雄氏に心から深く感謝の意を表します。さらに、本作業を進めるに当たり、機能設計の検討において有益な議論及び、本論文をまとめるにあたって適切な助言をして頂いた日本原子力研究所核融合装置試験部JT-60第1試験室長代理 木村 豊秋氏、同室 米川 出氏にそれぞれ厚く御礼申し上げます。

## 参考文献

- ( 1 ) Model 2140 Z1A Enhanced Serial Highway Driver for INSTRUCTION MANUAL  
August, 1992.
- ( 2 ) MVME147S MPU VMEmodule User's Manual Motorola Inc.
- ( 3 ) VMEbus アーキテクチャ・マニュアル Revision C.1 VME MEMBER, 1986
- ( 4 ) VxWorks REFERENCE MANUAL Wind River Systems, Inc. REV:7 May 91
- ( 5 ) JT-60 CAMAC手帳 昭和58年10月 日本原子力研究所  
JT-60装置試験部 発行

## 謝辞

今回、本試験システムの開発に際し、VME-bus及びCAMACシステム全般にわたり数々の技術的助言及び協力をして頂いた日本原子力研究所炉心プラズマ研究部炉心プラズマ解析室 青柳 哲雄氏に心から深く感謝の意を表します。さらに、本作業を進めるに当たり、機能設計の検討において有益な議論及び、本論文をまとめるにあたって適切な助言をして頂いた日本原子力研究所核融合装置試験部JT-60第1試験室長代理 木村 豊秋氏、同室 米川 出氏にそれぞれ厚く御礼申し上げます。

## 参考文献

- ( 1 ) Model 2140 Z1A Enhanced Serial Highway Driver for INSTRUCTION MANUAL  
August, 1992.
- ( 2 ) MVME147S MPU VMEmodule User's Manual Motorola Inc.
- ( 3 ) VMEbus アーキテクチャ・マニュアル Revision C.1 VME MEMBER, 1986
- ( 4 ) VxWorks REFERENCE MANUAL Wind River Systems, Inc. REV:7 May 91
- ( 5 ) JT-60 CAMAC手帳 昭和58年10月 日本原子力研究所  
JT-60装置試験部 発行

## 付録1 CAMACモジュールアクセスのプログラム例

ここで示すプログラム例は、まずL2型クレートコントローラに対してデータウェイのクリア、イニシャライズ、アンバイパス、オンライン、デマンドイネーブルを一度のexec\_vsdで実行する。エラーの場合、次の処理を行わず終了する。

次にステーションアドレスが19、クレート番号が1～31のどれかに実装されたDIL(デジタル入力モジュール)のLAM待ち状態になり、LAMの発生を受信すると共に該当すDILに対して、インプットレジスタ及び、メモリレジスタの内容を読み込みLAMソースをクリアする。

## CAMACコマンド

```
(CCL2)
C(1), N(30), A(0), F(19), DATA(0X0003)---クリア、イニシャライズ
C(1), N(30), A(0), F(23), DATA(0X0800)---アンバイパス
C(1), N(30), A(0), F(17), DATA(0X0100)---オンライン、デマンドイネーブル
(DIL)
C(?), N(19), A(0), F(0), DATA(0X0000)---インプットレジスタリード
C(?), N(19), A(1), F(0), DATA(0X0000)---メモリレジスタリード
C(?), N(19), A(10), F(1), DATA(0X0000)---LAMソースのクリア
```

## プログラム

```
taskA()
{
int    rt;
struct dmd_t req_lam;
static struct pram_t ccl2[] = {{1, 30, 0, 19, 0, 0, 0, 0x0003, 0},
                                {1, 30, 0, 23, 0, 0, 0, 0x0800, 0},
                                {1, 30, 0, 17, 0, 0, 0, 0x0100, 0}};
static struct pram_t dil[] = {{0, 0, 0, 0, 1, 0, 0, 0x0000, 0},
                               {0, 0, 1, 0, 1, 0, 0, 0x0000, 0},
                               {0, 0, 0, 10, 1, 0, 0, 0x0000, 0}};

    rt = exec_vsd(3, ccl2);
    if(rt){
        error_vsd(rt, 0, "CCL2 Command Excute Error");
        return;
    }
    req_lam.crl = 0xffffffff;
    req_lam.sgl = 19;
    if(rt = lam_vsd(1, &req_lam, 0)){
        error_vsd(rt, 0, "LAM Request Error");
    }else{
        dmd_to_cn(req_lam.drr, dil, 3);
        rt = exec_vsd(3, dil);
        if(rt){
```

```
        error_vsd(rt,0,"DIL Command Excute Error");
    }
}
return;
}
```

## 付録2 シリアル・ドライバのハードウェア設定

本シリアル・ドライバの使用に際しては、ハードウェアの各種設定をインストラクションマニュアルに沿って確認すること。

特に確認と変更が必要となるのは、以下の5点である。

### (1) デバイスアドレス

インストラクションマニュアルによるとデバイスアドレスの初期設定は、ショートアドレス0XFF00 に設定されている。

しかし、本試験システムで使用しているCPU (MVME147) は、ショートアドレスI/O時のポートサイズがD16 (16ビットアクセス) のためシリアルドライバ上の32ビットレジスタにアクセス出来ない。このため、本試験システムでは、デバイスアドレスを0XEFFFFFF00 に設定した。

### (2) VMEバス

本ドライバは、受信ボードと送信ボードの2つのVMEモジュールから構成されている。

VMEバックプレーンのスロットに本ドライバを差し込む場合、通常受信ボードを左側に入れる様にする。この時、VMEバックプレーンのP2側を付属のジャンパーコネクタで接続すること。また、送信側にあるVMEバス獲得用及び、割り込み用の5つのジャンパーを抜いてしまうと (通常モジュールを挿入した場合には、抜く) それ以降のスロットにVMEバスリクエストが届かなくなるのでジャンパーは、取り付けておくこと。

### (3) スペースバイト数

5MHzのバイトシリアルモードで動作させる場合のみ、クレートコントローラによってスペースバイトが不足し、縦パリティエラーが発生する場合がある。

送信ボード内のMSB (More Space Bytes) ジャンパーポストをジャンパーすることにより、スペースバイトを数バイト増やすことが出来る。

### (4) バス要求レベル

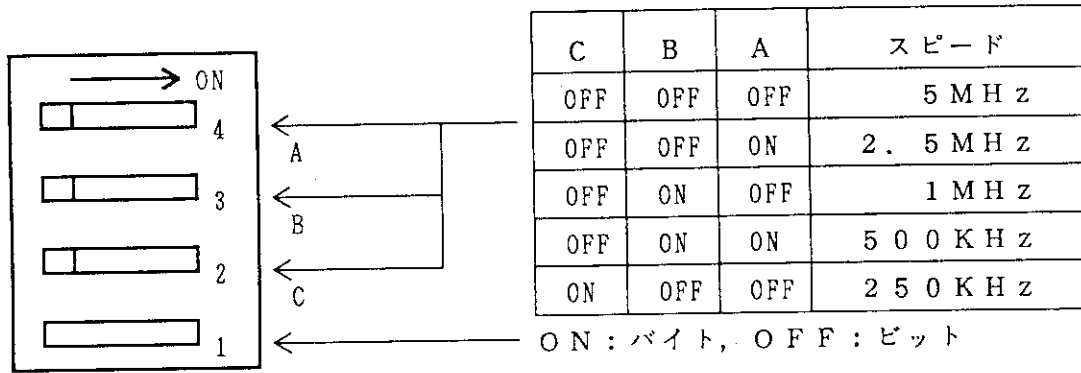
工場設定では、VMEバス要求レベル3になっている。

### (5) ビット・バイト切り替えと転送速度の切り替え

送信ボードフロントパネルに切り替え用ディップスイッチがある。

第1図にディップスイッチの設定方法を示す。





第1図 ディップスイッチの設定方法

## 付録3 シリアル・ドライバの機能概要

## 1. フロントパネル表示

受信ボードのフロントパネルLED表示により、ハードウェアの状態を確認する事が出来る。

- RUN : CAMACコマンドの出力からコマンドが終結するまでの間点灯する。
- LAM : シリアルハイウェイのデマンドを受信したときに点灯する。本ドライバは、512個のデマンド用FIFOに受信したデマンドを格納する。プログラムがデマンド用FIFOを読むか、デマンド用FIFOをクリアすることにより消灯する。
- NO SYNC : シリアルハイウェイ上のクロックが受信出来ない場合、点灯する。
- E1, E2, E3 : それぞれのビットパターンにより、エラー情報を表示する。
7. NO SYNC (E1:1, E2:1, E3:1)
  6. 指定したクレートアドレスは、存在しない。(E1:0, E2:1, E3:1)
  5. ハイウェイタイムアウト (E1:1, E2:0, E3:1)
  4. ハイウェイ出力失敗 (E1:0, E2:0, E3:1)
  3. 不使用
  2. X応答がない。(E1:0, E2:1, E3:0)
  1. Q応答がない。(E1:1, E2:0, E3:0)
  0. 正常。(E1:0, E2:0, E3:0)
- NO X : X応答がない。
- NO Q : Q応答がない。

## 2. レジスタレイアウト

本ドライバは、以下に示す22個の内部レジスタから構成されている。

(1) C S E R (Channel Status/Error Register)	SIZE:D16, OFFSET:0X00
(2) D O C (Device/Operation Control Register)	SIZE:D16, OFFSET:0X04
(3) S C C (Sequence/Channel Control Register)	SIZE:D16, OFFSET:0X06
(4) M T C (Memory Transfer Count Register)	SIZE:D16, OFFSET:0X0A
(5) M A C H I (Memory Address Counter High)	SIZE:D16, OFFSET:0X0C
(6) M A C L O (Memory Address Counter Low)	SIZE:D16, OFFSET:0X0E
(7) L A M C (LAM Interrupt Control Register)	SIZE:D16, OFFSET:0X40
(8) D O N E C (Done Interrupt Control Register)	SIZE:D16, OFFSET:0X42
(9) D M A C (DMA Buffer Empty Interrupt Control Register)	SIZE:D16, OFFSET:0X44
(10) L I S T A C (List Abort Interrupt Control Register)	SIZE:D16, OFFSET:0X46
(11) L A M V (LAM Interrupt Vector Register)	SIZE:D16, OFFSET:0X48
(12) D O N E V (Done Interrupt Vector Register)	SIZE:D16, OFFSET:0X4A
(13) D M A V (DMA Buffer Empty Interrupt Vector Register)	SIZE:D16, OFFSET:0X4C
(14) L I S T A V (List Abort Interrupt Vector Register)	SIZE:D16, OFFSET:0X4E
(15) A M R (Address Modifier Register)	SIZE:D32, OFFSET:0X60
(16) C M R (Command Memory Register)	SIZE:D32, OFFSET:0X64
(17) C M A (Command Memory Address Register)	SIZE:D32, OFFSET:0X68
(18) C W C (Command Word Count Register)	SIZE:D32, OFFSET:0X6C
(19) D R R (Demand Request Register)	SIZE:D32, OFFSET:0X70
(20) C D R (CAMAC Data Register)	SIZE:D32, OFFSET:0X74
(21) D B R (Data Burst Register)	SIZE:D32, OFFSET:0X78
(22) C S R (Control Status Register)	SIZE:D32, OFFSET:0X7C

3. 基本的なCAMACファンクションの発行手順

(1) シングルアクションのCAMACコマンド (非DMA, 非割り込み)

CAMACコマンド発行のための使用するレジスタを以下①～④に示す。

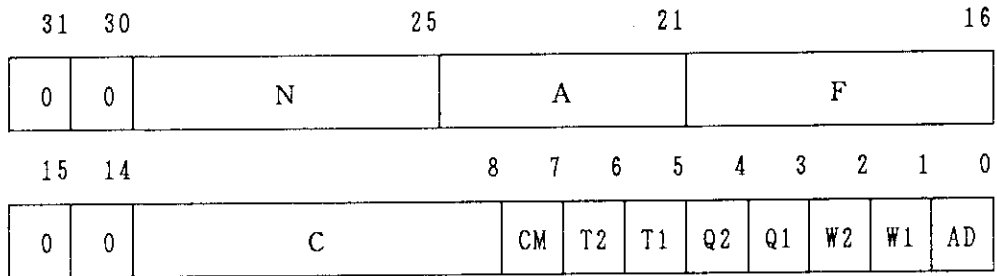
① CMR (Command Memory Register)

このレジスタは、CAMACコマンド (C: クレートアドレス, N: ステーションアドレス, A: サブアドレス, F: ファンクション) 及び、CAMACデータの転送モード等を書き込む。

8K \* 32bitのデータ空間から成り以下②で示すレジスタCMA (Command Memory Address register) に設定されているデータ空間のアドレスにCAMACコマンドが書き込まれる。

また、CAMACコマンドを並べて書き込んでおき、複数のCAMACコマンドを一度のオペレーションで出力することが出来る。

書き込んだCAMACコマンドの次にコマンドの終了を示すHALTを書き込まなければならない。



- N : ステーションアドレス
- A : サブアドレス
- F : ファンクション
- C : クレートアドレス
- CM : コマンドモード
  - 0: CAMACファンクション
  - 1: HALT
- T2, T1: 転送モード
  - 00: シングルアクション
  - 01: ブロック転送
  - 11: インライン・ライト
- Q1, Q2: Q応答モード
  - 00: Q-STOP
  - 01: Q-IGNORE
  - 10: Q-REPEAT
  - 11: Q-SCAN
- W1, W2: データサイズ (Word)
  - 00: 24bit
  - 01: 16bit (Normal)
  - 11: 16bit (Word Reversed)
- AD : アボートディゼーブル

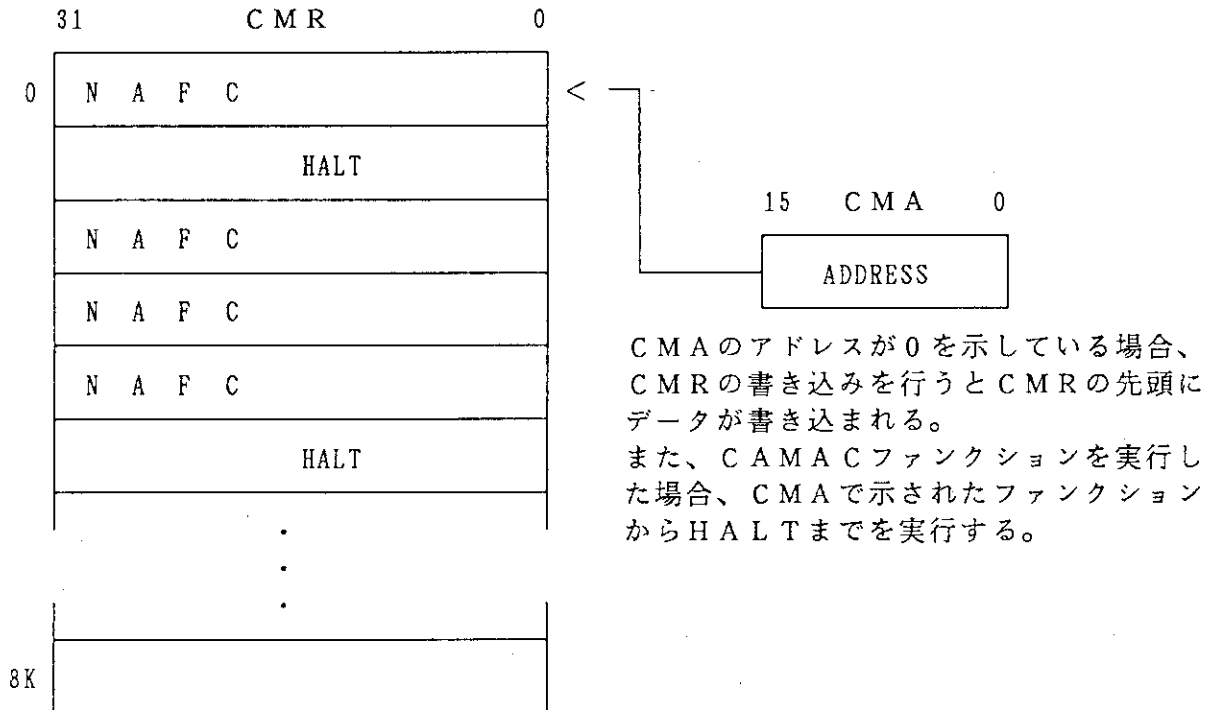
② CMA (Command Memory Address Register)

このレジスタは、上記①で示したレジスタCMRの書き込み位置のアドレスを示す。

CMRにデータが書き込まれる毎にCMAに書かれているアドレスは、インクリメントされる。

また、CMRのアドレスを書き換えることにより、既にCMRに書き込まれている任意のCAMACコマンドを出力することが出来る。

(CMRとCMAの関係)



③ CDR (CAMAC Data Register)

このレジスタは、リード及びライト時のCAMACデータが書き込まれる。

(24 bit データ)

31	24	0
ONES	データ (1)	
ONES	データ (2)	

(16 bit データ Normal)

31	16	0
データ (2)		データ (1)
データ (4)		データ (3)

(16 bit データ Word Reversed)

31	16	0
データ (1)	データ (2)	
データ (3)	データ (4)	

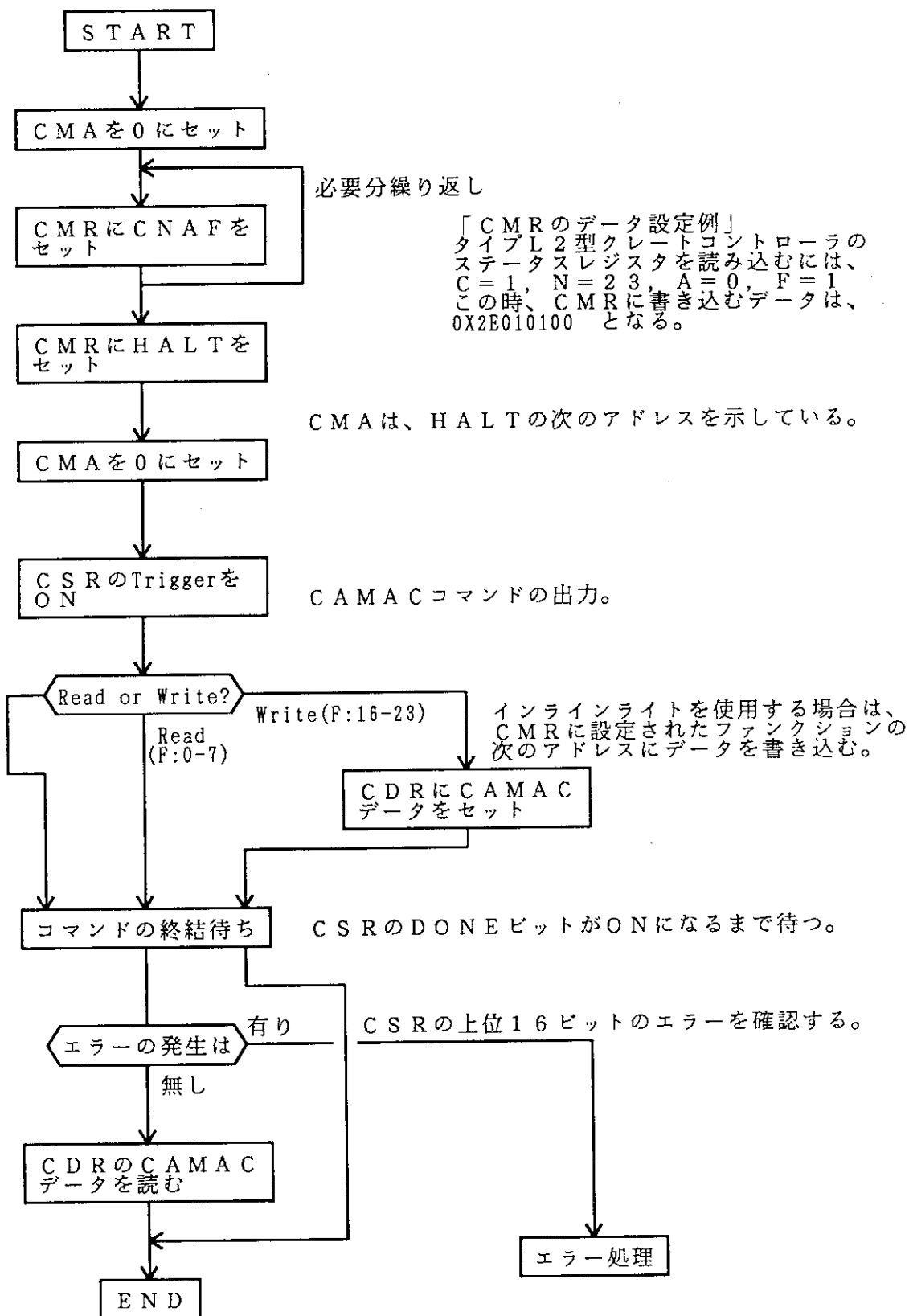
④ C S R (Control Status Register)

このレジスタの上位16ビットはCAMACコマンド実行時のエラー情報等を表し、下位16ビットは、本ドライバのコントロールレジスタである。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
w	v	0	0	u	t	s	r	q	0	p	o	n	m	l	k
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
j	i	h	g	f	e	d	c	b	a						

- a. Trigger : 本ビットに01を書き込むとCMRに登録されたCAMACコマンドを実行する。
- b. Timer : 0~Fを設定することにより、周期的にCAMACコマンドを実行する。
- c. Append : 本ビットを1にするとブロック転送時、CSRがデータの最終Wordに反映され転送される。
- d. Burst : 本ビットを1にするとDMA時、バースト転送となる。
- e. ERROR CODE : フロントパネル表示のエラー情報と同様。
- f. DONE : CAMACコマンドの実行が終了した。
- g. SHD Reset : 本ビットを1にするとドライバの異常がリセットされる。
- h. CLEAR DMD FIFO : 本ビットを1にするとデマンド受信FIFOがクリアされる。
- i. DMD OVER RUN : デマンド受信FIFOがオーバーフローした。
- j. DMD Pend : デマンド受信FIFOに処理されていないデマンドがある。
- k. NO Q : CAMACコマンドに対するQ応答がない。
- l. NO X : CAMACコマンドに対するX応答がない。
- m. DIR : 0:リードデータCAMACファンクション, 1:ライトデータCAMACファンクション
- n. ADNR : 存在しないクレートアドレスが指定された。
- o. TPE : 横パリティエラーを検出した。
- p. LPE : 縦パリティエラーを検出した。
- q. RPY ERR : リプライエラーを検出した。
- r. NO SYNC : シリアルハイウェイ上のクロックがない。
- s. TMO : ハードウェアタイムアウト
- t. DLY ERR : CAMACコマンドを実行する以前にエラーが発生していた。
- u. STE : シリアルハイウェイ出力失敗。
- v. N>23 : Q-SCANで動作中ステーションアドレスが23を過ぎた。
- w. ABT : CAMACコマンド出力中、エラーにより出力が中止された。

(2) 基本的なCAMACコマンドを出力する場合のプログラムの流れを以下に示す。



## 4. 割り込み処理

## (1) 発生するVME割り込みの種類

## ① L A M 割り込み

C A M A Cモジュールで発生したL A M信号がシリアルハイウェイ上にデマンドメッセージ及びハングデマンドとして反映された場合にデマンドメッセージを取り込みVME割り込みを発生する。

## ② D O N E 割り込み

1回のC A M A Cファンクションが終結した時、VME割り込みを発生する。

## ③ D M A B u f f e r E m p t y 割り込み

D M Aに於ける書き込み及び、読み込みが終了した時、VME割り込みが発生する。

## ④ L i s t A b o r t 割り込み

複数のC A M A Cファンクションを連続して実行中、何らかの異常でH A L Tまでのファンクションを発行出来なくなった時、VME割り込みを発生する。

上記①～④のVME割り込みを発生させるためには、各々に対応する割り込み制御レジスタ、割り込みベクタレジスタを設定しなければならない。

## (2) 割り込み制御レジスタ

## ① L A M C (L A M I n t e r r u p t C o n t r o l R e g i s t e r)

## ② D O N E C (D o n e I n t e r r u p t C o n t r o l R e g i s t e r)

## ③ D M A C (D M A B u f f e r E m p t y I n t e r r u p t C o n t r o l R e g i s t e r)

## ④ L I S T A C (L i s t A b o r t I n t e r r u p t C o n t r o l R e g i s t e r)

各レジスタのフォーマットは、全て共通で以下に示すとおりである。

15	8	7	6	5	4	3	2	1	0
不 使 用	FLAG	0	FLGC	INTE	INTC	IRQ2	IRQ1	IRQ0	

FLAG : ソフトウェアで使用する。

FLGC : 本ビットをONにするとVME割り込みが発生すると自動的にFLAGビットをOFFとする。

INTE : 本ビットをONにするとVME割り込みがイネーブルとなる。

INTC : 本ビットをONにするとVME割り込みが終了すると自動的にINTEビットをOFFとし、VME割り込みをディゼーブルとする。

IRQ2, IRQ1, IRQ0 :

VME割り込みレベルを設定する。(1~7)

(3) 割り込みベクタレジスタ

- ① L A M V (LAM Interrupt Vector Register)
- ② D O N E V (Done Interrupt Vector Register)
- ③ D M A V (DMA Buffer Empty Interrupt Vector Register)
- ④ L I S T A V (List Abort Interrupt Vector Register)

各レジスタのフォーマットは、全て共通で以下に示すとおりである。

	15		8	7	6	5	4	3	2	1	0
	不 使 用			割 り 込 み ベ ク タ *1							

(4) L A M 割り込み

(2)、(3)にて示したL A M 割り込みレジスタを設定することにより、本ドライバは、シリアルハイウェイ上のデマンドメッセージを検出した場合、V M E 割り込みを発生する。

デマンドメッセージは、本ドライバのレジスタD R R に格納される。

D R R は、5 1 2 個のデマンドメッセージ用F I F O からなり、デマンドの発生順に格納される。

アプリケーションは、V M E 割り込みルーチンの中でD R R を読み込むか、割り込み制御レジスタL A M C の割り込みオートクリアフラグ(F L G C)をO N にしておかないと、V M E 割り込みが終了しない等の異常動作が発生する。

D R R のフォーマット

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	S G L				0	0	C R A T E						

S G L : コード1 ~ 1 F、コード1 F は、ハングデマンド。

C R A T E : 1 ~ 6 2

D R R には、シリアルハイウェイ上にデマンドメッセージが発生する毎に常に書き込まれる。

L A M 割り込み受信処理を起動する以前に溜まったデマンドメッセージが不必要なものであれば、C S R を操作することにより、これをクリアすることが出来る。

\*1 割り込みベクタとは、V M E 割り込みレベル以外に受信側で受信した割り込みに対する処理を判断させるために使用する。V M E 割り込みと共に割り込みデータとして受信する。



5. ブロック転送とDMA

ブロック転送指定の時、本ドライバは、与えられたデータ語数分のCAMACコマンドを発行する。この時、CPU(MVME147)が、CSRのDONEフラグを見てデータの読み書きを行うことも出来るが、データ語数が非常に多い場合には、DMA転送を使うとCPUの負荷が軽減する。

DMA転送を使用する場合、本ドライバのDMAコントロールのための各レジスタ、DMAの終了をCPUに通知するための割り込み処理の設定等、各種設定を行わなければならない。

(1) DMA転送にて使用するレジスタを以下①～③に示す。

① CSER(Channel Status/Error Register)

このレジスタは、DMAコントローラの状態及びエラーコードが格納される。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
d	0	c	b	a	0	0	1	0	0	0	ERROR CODE				

ERROR CODE

- 0x01001 : バスエラーが発生した。
- 0x10001 : アプリケーションからDMAの中止が行われた。
- a. ACT : DMAが動作中であることを示す。
- b. ERR : DMAが異常終了したことを示す。
- c. NDT : CAMACコマンドリストが途中終了したことを示す。
- d. COC : DMAが正常に終了したか否かを示す。0:異常終了、1:正常終了

② D O C (Device/Operation Control Register)

このレジスタは、DMA転送に於ける転送方向、Wordサイズ等を定義する。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
d	0	1	1	c	0	0	0	b	0	a	0	0	1	0	

- a. SIZE : Wordサイズ。(本ドライバでは、常に11。1Word=32bit)
- b. DIR : DMA転送方向 0:CPUメモリ→ドライブ(Write)、1:ドライブ→CPUメモリ(Read)
- c. OCR : 本ドライバでは、常に1。
- d. REC : 転送モード 0:バーストモード、1:サイクルスチールモード

③ S C C (Sequence/Channel Control Register)

このレジスタを操作し、DMA転送の開始、中止等を制御する。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	c	0	0	b	a	0	0	0

- a. INTINA : 本ビットをONにするとDMA Buffer Empty割り込みがイネーブルになる。
- b. SOFTAB : 本ビットをONにするとDMAが中止される。
- c. START : 本ビットをONにするとDMAが開始される。

④ M T C (Memory Transfer Count Register)

このレジスタは、DMA転送語数（転送Word数）を指示する。

⑤ M A C H I, M A C L O (Memory Address Counter High, Low)

このレジスタは、DMA転送を行うアドレスを指示する。

⑥ D M A C (DMA Buffer Empty Interrupt Control Register)

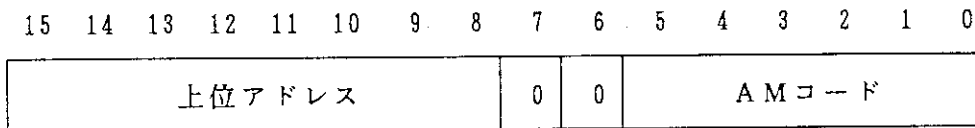
本文中 4. 割り込み処理参照。

⑦ D M A V (DMA Buffer Empty Interrupt Vector Register)

本文中 4. 割り込み処理参照。

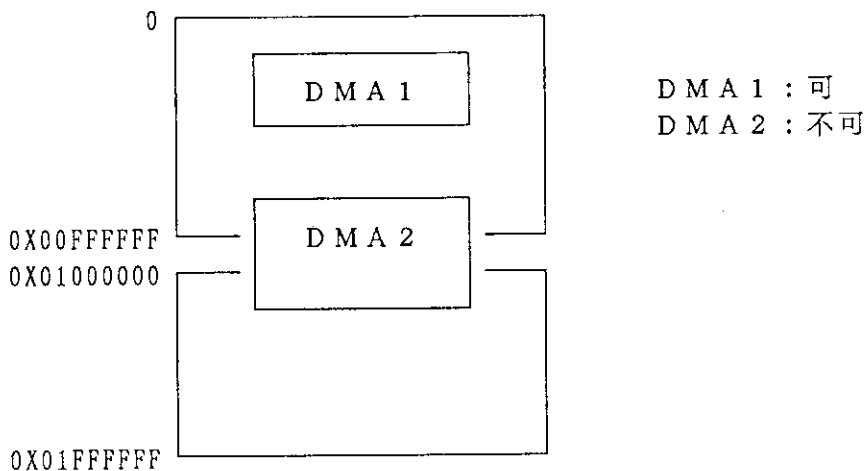
⑧ A M R (Address Modifier Register)

VMEアドレスモディファイアコード（AMコード）を指示する。



32ビットアドレスを使用する場合、24ビットアドレス境界以上の上位アドレスを設定する。この時、24ビットアドレス境界をまたぐDMA転送は、出来ない。

即ち、下図に示す様にCPU(MVME147)のメモリアドレス0X00FFFFFF以下のアドレスから0X01000000以上の連続する領域へのDMA転送は、バスエラーとなる。



⑨ C W C (Command Word Count Register)

— (DMA転送すべき語数) + (DMA転送を行った語数) が格納される。

何らかの異常でDMA転送が終結しなかった時、何ワード転送されたか確認出来る。

正常終了時は、0となる。

(2) 割り込み処理とDMAを使用したブロック転送時のプログラムの流れを以下に示す。

