

# PC/Linux による VMEbus ベースの データ収集システムの構築

仲吉一男<sup>a</sup>

kazuo.nakayoshi@kek.jp

安 芳次<sup>b</sup>

Yoshiji.YASU@kek.jp

<sup>a</sup> 高エネルギー加速器研究機構田無分室

<sup>b</sup> 高エネルギー加速器研究機構  
素粒子原子核研究所オンライングループ

## Abstract

PC/AT 互換機と Linux による VMEbus をベースとするデータ収集システムの構築を行った。このシステムの特長は、コストパフォーマンスが高いことである。システムの入出力性能は、Programmed Input/Output で 2MB/s、DMA で 6MB/s である。このシステムで使用する VME ライブラリを配付キットとしてまとめ WWW 上で公開している。

# 目次

<b>1</b>	<b>背景と目的</b>	<b>1</b>
1.1	背景	1
1.2	目的	1
<b>2</b>	<b>要求される機能</b>	<b>1</b>
<b>3</b>	<b>システムの構成</b>	<b>2</b>
3.1	ハードウェアの構成と特長	2
3.2	ソフトウェアの構成と特長	2
<b>4</b>	<b>システムの性能</b>	<b>3</b>
<b>5</b>	<b>Linux 用 VME ライブラリ</b>	<b>4</b>
<b>6</b>	<b>DAQ システム構築例</b>	<b>4</b>
<b>7</b>	<b>まとめ</b>	<b>6</b>
	謝辞	6
	参考文献	6
<b>A</b>	<b>VME ライブラリのインストール方法</b>	<b>8</b>
A.1	ハードウェアのセットアップ	8
A.2	ソフトウェアのセットアップ (vmehb-1.x, vmelib)	9
<b>B</b>	<b>サンプルプログラム intreg.c について</b>	<b>13</b>
B.1	ソースファイルの説明	13
B.2	Interrupt Register の設定	15
B.3	サンプルプログラムを実行する	15
<b>C</b>	<b>Troubleshooting</b>	<b>15</b>
C.1	LAN カード (3Com 3C509B-COMBO/ISA) が使えなくなった	15
C.2	insmod vmehb をすると initialization of vmehb failed となる	16

# 1 背景と目的

## 1.1 背景

VMEbus<sup>1)</sup> はワークステーション (WS) と CAMAC<sup>2)</sup> や TKO<sup>3)</sup> とのインターフェイスとして、また各種機器の制御用バスとして長年使用されてきた。Linux は POSIX 仕様でフリーの UNIX 互換 OS であり、現在多くのパッケージが存在する。また、Pentium に代表されるパーソナルコンピュータ (PC) の CPU 性能は、年々高くなっており、そのコストパフォーマンスは WS に比べて高い。その PC に Linux をインストールすると高性能な PC UNIX マシン (PC/Linux) となる。PC/Linux では CERN Library を使用することができるので、高エネルギー物理分野ではデータ解析にも使用されている<sup>4,5)</sup>

PC/Linux による VMEbus ベースのデータ収集 (DAQ) システム構築の背景として以下の 2 点が挙げられる。

- VME の様々なモジュールのデータを PC/Linux で読み取り処理したいという実験グループからの要求がある<sup>6)</sup>
- PC/Linux による DAQ システムは、これまで CAMAC<sup>7-9)</sup> を中心に開発が行われており VMEbus に関してはまだ行われていない。

## 1.2 目的

本システムの目的は「コストパフォーマンスの高い VMEbus ベースの DAQ システムを構築すること」である。本システムでは VMEbus システムのテストベンチや VMEbus ベースの小・中規模実験の DAQ システムとしての使用をターゲットとして考えている。

# 2 要求される機能

実験グループから出されている要求を考慮して本システムに対する具体的な要求を以下の 3 つにまとめた。

1. VME モジュールを PIO(Programmed Input/Output) で 1~2MB/s ,DMA(Direct Memory Access) で 5~6MB/s で読めること。
2. PC/Linux による既存の CAMAC システムと共存できること。
3. システムのトータルコストが低いこと。

## 3 システムの構成

### 3.1 ハードウェアの構成と特長

本システムで使用したハードウェアの構成を以下に示す。

- PCIバスをもつ PC/AT 互換機。
- Bit3 社 Model 617/616<sup>10)</sup>(PCI-VME Adaptor)。

最近の PC/AT 互換機は、ほとんど PCIバスを持っている。PC のスペックとしては、CPU : Pentium 120MHz、メモリ : 32MB 程度で十分である。

PCI-VME Adaptor の Model 617 は DMA 機能付き、Model 616 は、DMA 機能なしのモデルである。Model 617/616 は、PCI カードと VME ボードおよびそれらをつなぐケーブルで構成される。

本システムのハードウェアの特長は、

「PC の電源を切らずに VME クレーターの電源を ON/OFF できる」ということである。これは、本システムを VMEbus システムのテストベンチとして使用する場合には有効である<sup>11)</sup>

### 3.2 ソフトウェアの構成と特長

本システムのソフトウェア構成を以下に示す。

- Linux システム (Linux カーネル 2.0.x およびその配付パッケージ)。
- Bit3 社 Model 617/616 用デバイスドライバ vmehb<sup>12)</sup>。
- Linux 用 VME ライブラリ vmelib。

ソフトウェアの構成は、NIKHEF の Kruszynska-Straszewicz 氏が開発した Bit3 社 Model 617/616 用デバイスドライバ vmehb と新規に作成した VME ライブラリ vmelib から構成される。vmehb の特長は、

- ローダブルなデバイスドライバである。
- VMEbus へのアクセスは、アドレス幅 16/24/32bit、データ幅 16/32bit が可能。
- 存在しない VME アドレスにデータを書いてもシステムがクラッシュしない。

「ローダブルなデバイスドライバ」とは、動作中のカーネルにモジュール化したデバイスドライバを組み込んだり、取り除いたりすることができるという意味である。「存在しない VME アドレスにデータを書き込んでもシステムがクラッシュしない」という特長は、システムの安定運用の面で優れた特長であり、テストベンチとして使用する場合には特に有効である。

## 4 システムの性能

システムの性能を測定するため、以下のことを行った。

VMEbus 上の高速メモリモジュール (CI-VME64/32, 8192 x 32bits) に対してアドレス幅 32bit, データ幅 32bit でデータ量を変化させながら DMA および PIO によるリードとライトを各々 10000 回行い, その転送時間の分布から各データ量での転送時間を決定した。図 1 に DMA, PIO のデータ量と転送時間の関係を示す。

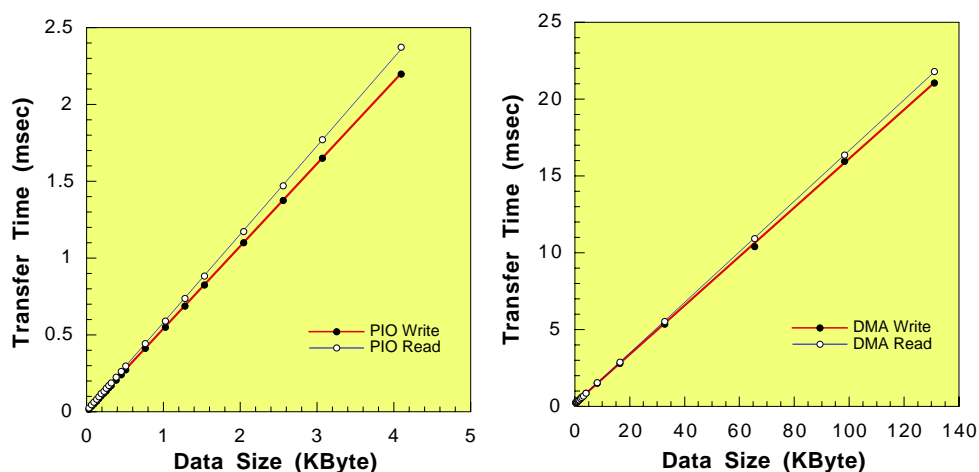


図 1: 本システムの DMA および PIO のデータ量と転送時間の関係 (左は PIO, 右は DMA). グラフの  $\bullet$  はライト,  $\circ$  はリードを表している. グラフの横軸はデータ量 (KB), 縦軸は転送時間 (msec).

図 1 のグラフを直線で近似し, DMA および PIO のスピードと DMA のオーバーヘッドを求めた. その結果を表 1 に示す. 測定には, Pentium 120MHz, Linux カーネル 2.0.29 の PC を使用した.

	PIO (MB/s)	DMA (MB/s)	DMA Overhead ( $\mu$ s)
READ	1.8	6.1	160
WRITE	1.9	6.3	180

表 1: DMA, PIO アクセス性能. (CPU: Pentium 120MHz, メモリ: 32MB, カーネル: Linux 2.0.29)

同様な測定を Pentium II 300MHz の PC でも行ったが, Pentium 120MHz と Pentium II 300MHz の PC では DMA 転送に関しては, 性能の向上が 10 数% 見られたが PIO に関してはほとんど変化が見られなかった。

## 5 Linux 用 VME ライブラリ

このシステムを使用してユーザが容易にプログラムを書けるように，KEK 標準 VME ライブラリ<sup>13)</sup>と同じ関数をもつ Linux 用のライブラリを新規に作った．ライブラリの構成は，表 2 のようになっている．このライブラリの関数をプログラム中でコールすることにより VME モジュールに容易にアクセスすることができる．

Function	Description
vme_mapopen()	VME アドレス空間をユーザプロセスのアドレス空間へ mapping する
vme_mapclose()	ユーザプロセスのアドレス空間へ mapping した VME アドレス空間の解放
vme_dmaopen()	DMA 転送モードの設定
vme_dmaclose()	DMA 転送モードの解除
vme_dmaread()	DMA 転送によるデータの読み込み
vme_dmawrite()	DMA 転送によるデータの書き込み

表 2: Linux 用 VME ライブラリの関数．

## 6 DAQ システム構築例

KEK 田無分室の電子シンクロトロン<sup>14)</sup>で実験中の ES159 実験グループ<sup>15)</sup>の DAQ システムを借用し，本システムを使用したデータ収集のテストを行った．

ES159 実験の DAQ システムと本システムの概略を図 2 に示す．

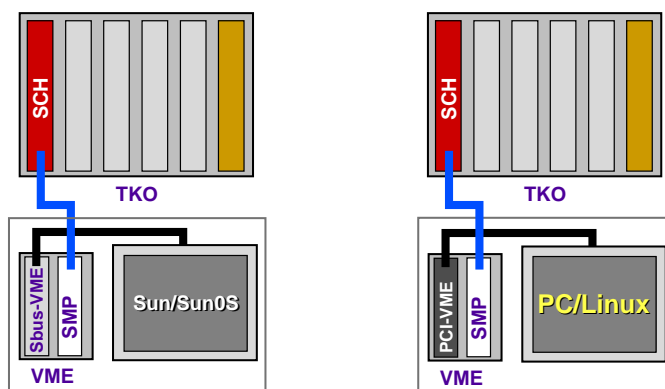


図 2: 左側の図が，ES159 実験の DAQ システム (SPARCstation IPX/SunOS と Sbus-VME Adaptor)，右側が本システム．

ES159 実験の DAQ システムのフロントエンドは、VMEbus ではなく、TKO システムである。TKO システムと計算機とのインターフェイスとして VME SMP<sup>16)</sup> モジュールを使用している。従って VMEbus 上の VME SMP モジュールへアクセスすれば TKO システムを制御し、データを取得することができる。ES159 実験の DAQ システムから本システムへの変更は、VME クレートから Sbus-VME 変換モジュールを外し、代わりに PCI-VME 変換モジュールを入れるだけである。

本システムから VME SMP へのアクセスは、KEK オンラインの五十嵐洋一氏が作成した TKO ライブラリを Linux 用に修正して使用した。

オンラインモニターには、CERN Library の Shared Memory を用いてデータのヒストグラミングと表示を行った。図 3 にその一例を示す。

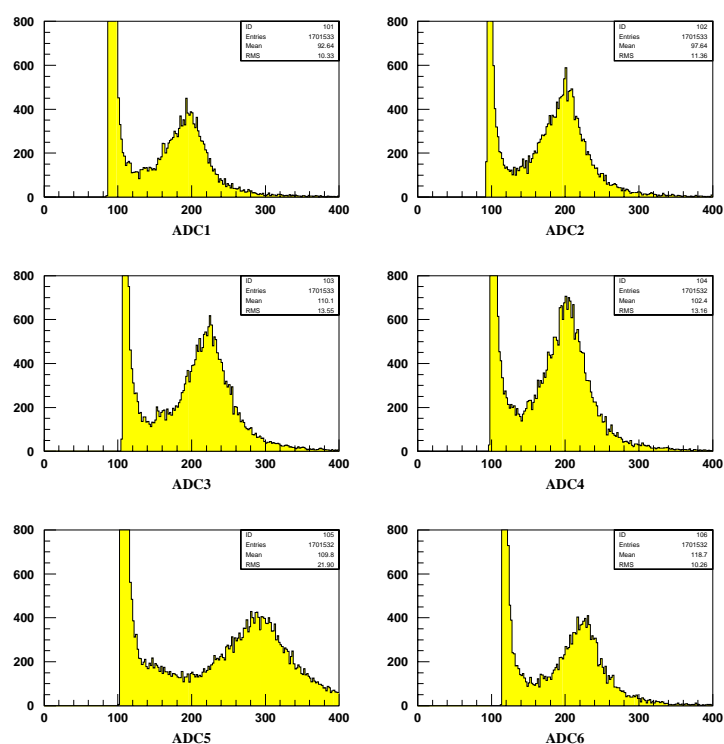


図 3: 本 DAQ システムでとった純 CsI 結晶カロリメータの宇宙線に対する応答のデータ例。

## 7 まとめ

PC と Linux による VMEbus をベースとする DAQ システムの構築を行った . このシステムの最大の特長は , コストパフォーマンスが高いことである . システムの性能は , Pentium 120MHz の PC を使用して PIO : 2MB/s , DMA : 6MB/s である (ともにリード/ライト) . Linux 用 VME ライブラリは , 配付キット<sup>17)</sup>としてまとめ WWW 上からダウンロードが可能である . 配付キットには , ハードウェアおよび VME ライブラリのインストール方法 , VME ライブラリの使用方法 , サンプルプログラムが含まれている .

## 謝辞

本システム導入の際に PC , VME クレートおよび作業スペース等を提供していただいた長基線ニュートリノ振動実験グループとオンライングループの方々に感謝致します . また , TKO を使った本システムのテストに御協力頂いた ES159 実験グループおよび ES グループの方々に感謝致します .

## 参考文献

- [1] Microprocessor Standards Committee of the IEEE, "VMEbus" A Standards specification for a Versatile Backplane Bus, P1014, March 1987.
- [2] Modular Instrumentation and Digital Interface System (CAMAC) ANSI/IEEE Std. 583-1982.
- [3] KEK Data Acquisition Development Working Group, TKO SPECIFICATION, KEK Report 85-10, 1985.
- [4] 例として「Linux Resources for High Energy Physics」というサイトを参照のこと . URL は , <http://hepwww.ph.qmw.ac.uk/HEPpc/> . DAQ 関連では , KEK オンライングループのホームページを参照のこと . URL は , <http://www-online.kek.jp/~online/Linux/linux.html> .
- [5] Stephen Wolbers, Commodity Hardware and Software Summary, CHEP97, 1997
- [6] 例えば , 長基線ニュートリノ振動実験 (E362) では , VME Flash ADC のデータを読みだし処理したいという要求があった .
- [7] 山形大学と KEK オンライングループにより開発された TOYO CC/7x00 による CAMAC DAQ システム . <http://www-online.kek.jp/~online/Linux/linux.html> .



- [8] 長崎総合科学大学と KEK オンライングループにより開発された Kinetic K3922 による CAMAC DAQ システム . <http://www.elc.nias.ac.jp/~daq/software/> .
- [9] E362 実験グループと KEK オンライングループにより開発された HOSHIN CCP による CAMAC/TKO DAQ システム . E362 実験グループでは , ビームライン・環境 ( 温度 , 湿度等 ) モニターの計算機に PC/Linux を採用することが決まっている . <http://neutrino.kek.jp/~kohama/ccp/ccp.html> .
- [10] 国内では , PCI-VME Adaptor はソリトンシステムズが販売している . <http://www.soliton.co.jp/sbo/> .
- [11] 例えば , SPARCstation と Sbus-VME Adaptor では , VME クレークの電源を切る場合には WS をいったん shutdown しなければならなかった .
- [12] 最新の vmehb は以下の URL からダウンロードが可能である . <ftp://nikhef.nikhef.nl/pub/projects/vmehb/> .
- [13] <http://www-online.kek.jp/~online/VME/> .
- [14] 田無分室 1.3GeV 電子シンクロトロンについては以下の URL を参照のこと . <http://www.tanashi.kek.jp/~okuno/> .
- [15] ES159:  $(\gamma, \eta)$  反応による原子核内 S11 状態の研究 .
- [16] VME SMP(Super Memory Partner) は , VMEbus と TKO システムのインターフェイス・モジュールで VMEbus 上で通常のメモリとして動作すると共に TKO システムを制御し TKO BOX にある各種のモジュールのデータを収集するモジュールである .
- [17] 本システムの説明と VME ライブラリの配付キットについては , 以下の URL を参照のこと . <http://onlax2.kek.jp/~nakayosi/> .

## A VME ライブラリのインストール方法

インストールには以下の環境が必要です。

- PCI バスをもち Linux システム (カーネル 2.0.x およびその配付パッケージ) がインストールされている PC/AT 互換機。
- Bit3 社 Model 617/616(PCI-VME Adaptor)。

Linux 2.0.x の配付パッケージやそのインストール方法については、数多くの書籍があるのでそれらを参照してください。

### A.1 ハードウェアのセットアップ

#### 【1】 PCI Adaptor Card のインストール

1. インストール先の PC のシステムを停止して、電源を切る。
2. 空き PCI bus スロットに PCI Adaptor Card を挿し込み、ねじで固定する。

#### 【2】 VME Adaptor Card のインストール

1. VME adapter card をシステムコントローラにするか決める。システムコントローラにするならスロット # 1 (左端) に挿す。以下では VME adapter card はシステムコントローラであることを前提に説明します。それ以外での使用は、617/616 のマニュアルを参照のこと。
2. VME adapter card を挿す VME クレータのスロットにあるバスリクエスト、バスグラント、割り込みアクノリッジの信号をつなぐためのジャンパを取る。
3. 以下のように VME adapter card の Bus Grant(BGO-BGI) と Bus Request(BR) のジャンパーをセットする。617, 616 共通。(下図の - はジャンパー IN を表している)

BGO-BGI	BR
o-o o o BA	
o-o o-o 3	o o 0
o o-o o 2	o o 1
o o-o o 1	o o 2
o o-o o 0	o-o 3

4. 以下のようにシステムジャンパーをセットする . 617 , 616 共通 .

- o o 1
- o-o 2
- o-o 3
- o-o 4
- o-o 5
- o-o 6
- o o 7

5. VME adapter card をスロット # 1 に挿し込み , ねじで固定する .

### 【3】 ケーブルの接続

1. VME adapter card のコネクタとケーブルの A/B ラベルを確かめて接続する .
2. PCI Adaptor Card のコネクタとケーブルを接続する .

### 【4】 PC と VME システムの電源を入れる

電源投入後 , 両システムのアダプターの LED が光っていることを確認する .

## A.2 ソフトウェアのセットアップ (vmehb-1.x, vmlib)

ソフトウェアの設定について説明します . 本文中で % typewriter とあった場合 , typewriter とタイプしてリターンキーを押すことを意味します . % は一般ユーザ , # はスーパーユーザのときの Shell のプロンプトを表します .

### 【1】 デバイスドライバ (vmehb-1.x) のファイルをダウンロードし , 展開する

1. <http://onlax2.kek.jp/~nakayosi/vmlib4linux.html> に Netscape 等のブラウザでアクセスする .
2. 最新の vmehb-1.x.tgz をダウンロードし保存する .
3. % tar zxvf vmehb-1.x.tgz

### 【2】 bit3\_def.h の Device ID と IRQ の修正

デバイスドライバ (vmehb-1.x) の初期設定では , Model 617 となっているため , Model 616 を使用する場合は , bit3\_def.h の PCI\_DEVICE\_ID\_BIT3\_VME を 0x1 から 0x3 へ , BIT3\_IRQ を 5 から 15 へ修正する . (以下参照)

```

/*****
** PCI configuration of bit3 & AMCC devices
*****/
#define PCI_VENDOR_ID_BIT3      0x108a
#define PCI_DEVICE_ID_BIT3_VME 0x1      0x3
                        (以下,省略)
/*****
** BIT3 configuration constants
*****/
#define BIT3_MAJOR      62
#define BIT3_IRQ        5      15
                        (以下,省略)

```

**【3】 README.vmehb ファイルを読んで, インストールを行う**

**【4】 root になり insmod vmehb を実行後, dmesg コマンドでドライバのロードが正常に行われたか確認する**

以下に成功した場合の出力例 (616 の場合) を示す .

```

% su
# insmod vmehb
# dmesg

```

```

vmehb: config the device 108a 3 on bus 0 fn 144

```

1(617 の場合)

```

----- start of PCI register dump for vmehb -----
PCI_VENDOR_ID:      0x108a
PCI_DEVICE_ID:      0x3      0x1(617 の場合)
PCI_COMMAND:        0x7      IO mapped
PCI_STATUS:         0x400
PCI_CLASS_REVISION: 0x680004e
PCI_CACHE_LINE_SIZE: 0x0
PCI_LATENCY_TIMER:  0x60
PCI_HEADER_TYPE:    0x0
PCI_BASE_ADDRESS_0: 0xfcc1
PCI_BASE_ADDRESS_1: 0xfffd0000
PCI_BASE_ADDRESS_2: 0xfffc0000
PCI_BASE_ADDRESS_3: 0xfc000000

```

```
PCI_INTERRUPT_LINE: 9
PCI_INTERRUPT_PIN: 1
PCI_MIN_GRANT: 4
PCI_MAX_LATENCY: 64
----- end of PCI register dump -----
```

```
PCI_conf._base0 = fcc1
Bit3_base0 = fcc0
IRQ - 2.0 DMA serialized with semaphores
```

### 【5】 proc ファイルを見て確認を行う

以下に成功例を示す .

```
% cat /proc/devices
```

```
Character devices:
 1 mem
 2 pty
 3 tty
 4 tty
 5 cua
 6 lp
 7 vcs
62 vmehb <=== OK
```

```
% cat /proc/ioports
```

```
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0080-009f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : npu
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0378-037f : lp
03c0-03df : vga+
03f0-03f5 : floppy
```

```
03f6-03f6 : ide0
03f7-03f7 : floppy DIR
03f8-03ff : serial(auto)
fca0-fcbf : 3c590 Vortex 10Mbps
fcc0-fcdf : bit3_ior_0          <=== OK
```

```
% cat /proc/interrupts
```

```
0: 105379180 timer
1: 73105 keyboard
2: 0 cascade
4: 723213 + serial
9: 0 + vmehb          <=== OK
10: 16645748 3c590 Vortex 10Mbps
13: 1 math error
14: 2810620 + ide0
```

#### 【6】 VME ライブラリのファイルをダウンロードし展開する

1. <http://onlax2.kek.jp/nakayosi/vmelib4linux.html> に Netscape 等のブラウザでアクセスする .
2. 最新の `vmelib0.x_linux.tar.gz` をダウンロードする .
3. `% tar zxvf vmelib0.x_linux.tar.gz`

#### 【7】 VME ライブラリ `libvmelib.a` を作る

1. `% cd vmelib0.x_linux/lib`
2. `% make`

コンパイルが成功したら , `libvmelib.a` というファイルができる .

#### 【8】 実行可能な example program を作る

1. `% cd vmelib0.x_linux/example`
2. `% make`

コンパイルが成功したら , `intreg` , `intreg2` , `vscaler` という実行可能なファイルができる .

## B サンプルプログラム intreg.c について

配付キットを展開してできる example ディレクトリの中の intreg.c について説明します。これは、KEK 仕様の「VME Interrupt and I/O Register Module(以下、Interrupt Register)」の I/O Register としての機能をテストするものです。具体的には、Interrupt Register の OUT レジスタにデータを書くことによりフロントパネルから NIM 信号を出力させ、次にケーブルで入力コネクタとつなぎ IN レジスタの値を読むことで入力信号があったかどうかを確認する。

### B.1 ソースファイルの説明

#### 【1】インクルードファイル intreg.h の説明

```
#define INTREG_ADDR 0x8F00
#define INTREG_SIZE 16
```

Interrupt Register の VME 上のアドレスを 0x8F00、サイズを 16Byte にする。

```
struct intreg {
    unsigned short latch[2];
    unsigned short flipflop;
    unsigned short in;
    unsigned short pulse;
    unsigned short out;
    unsigned short csr[2];
};
```

Interrupt Register のレジスタ構造と同じ構造体を定義する。レジスタ構造の説明はマニュアルを参照してください。

#### 【2】intreg.c の説明

```
#include <stdio.h>
#include <sys/types.h>
#include "vmelib.h"
#include "intreg.h"
```

```
volatile struct intreg *mm;
```

intreg 構造体を指すポインタ mm を定義します。

```

extern errno;

main(argc, argv)
int argc;
char **argv;
{
    volatile caddr_t vme_addr;
    int outreg, outbit = 1;

    vme_addr = (volatile caddr_t)vme_mapopen(VME_A16D16MODE, INTREG_ADDR,
                                             INTREG_SIZE);

    if (vme_addr == (volatile caddr_t)-1 ) {
        printf("intreg cannot open! \n");
        exit(1);
    }
}

```

VME ライブラリの `vme_mapopen()` をコールして、VME アドレス空間をプロセスのアドレス空間へ mapping を行う。mapping が失敗すれば -1 が、成功すればプロセスのアドレス空間の先頭アドレスが返される。プロセスのアドレス空間へ READ/WRITE することにより 対応する VME 空間のアドレスの READ/WRITE が実行される。

```

mm = (volatile struct intreg *)vme_addr;
printf("address = %x\n", mm);
mm->pulse = 0x0;

```

Interrupt Register からの出力信号はレベル出力にする。

```
mm->out = outbit;
```

出力信号は、フロントパネルの OUT1 から出力する。

```

/* display contents of registers... */
printf("Latch 0 = %4X\n",mm->latch[0] & 0xFF);
printf("Latch 1 = %4X\n",mm->latch[1] & 0xFF);
printf("in      = %4X\n",mm->in & 0xFF);
printf("pulse   = %4X\n",mm->pulse & 0xFF);
printf("out     = %4X\n",mm->out & 0xFF);
printf("csr 0   = %4X\n",mm->csr[0] & 0xFF);
printf("csr 1   = %4X\n",mm->csr[1] & 0xFF);
}

```

Interrupt Register のアドレスの状態を表示する。



## B.2 Interrupt Register の設定

### 【1】 VME ベースアドレスを設定する

VME ベースアドレスを 0x8F00 に設定する . SW2 は A04-07 , SW4 は A08-A15 に対応している . 従って SW2 は , 0x0 , SW4 は , 0x8F に設定する .

### 【2】 フロントパネル上の INPUT1 と OUTPUT1 をケーブルでつなぐ

【3】 VME クレー트에 Interrupt Register をセットしてクレートの電源を入れる

## B.3 サンプルプログラムを実行する

### 【1】 実行ファイル intreg をつくる

```
% make intreg
```

### 【2】 プログラムを実行する

コンパイルが成功して intreg というファイルができていたら

```
% intreg
```

とタイプしてプログラム実行させる . その結果 「in = 1」と表示されれば OK です . 次にケーブルを INPUT1 から INPUT2 へ挿し替えてもう一度実行する . その結果 , 「in = 2」と表示されれば OK です .

## C Troubleshooting

PCI Adaptor Card を PC にインストールした際 , 障害が発生したことがあったので実例とその対処法を参考のために示します .

### C.1 LAN カード (3Com 3C509B-COMBO/ISA) が使えなくなった

#### 【対策】

LAN カード付属のインストール用フロッピーディスクを使用して , カードのプラグアンドプレイ機能を OFF にする . その後 , マニュアルで衝突しない IRQ を設定する .

### 【手順】

1. % `cat /proc/interrupts` をして 何番の IRQ が衝突しているか確認する .
2. PC を DOS で立ち上げる .
3. LAN カード付属のフロッピーディスクを入れる .
4. % `3C59XCFG CONFIGURE /INT:5`  
衝突しない番号 (この例では 5) を指定
5. その後 , Linux を立ち上げる .

## C.2 `insmod vmehb` をすると `initialization of vmehb failed` となる

### 【対策】

PCIバスにあるカードの順番を変えてみる . LAN カード (3Com 3c590/PCI) と 616 の PCI adaptor card を入れ換えると OK になった .

### 【手順】

1. `dmesg , cat /proc/devices , cat /proc/ioports , cat /proc/interrupts` コマンドを実行をする . その出力を前記の「ソフトウェアのセットアップの成功例」と比べてみる .
2. PC を shutdown して電源を切り , PCIバスにあるカードを 1 枚残して , 残りをすべて外して電源を ON にする .
3. 問題がなければ , 次のカードを挿入し電源を ON にする . その作業を繰り返し PCI adaptor card を挿入した後 , 再び `insmod vmehb` を実行する .
4. それでもダメな場合は , 挿入する場所を変えて同じ作業をやる .