

メモリーボードへのアクセスについての確認メモ

高エネルギー物理学研究所 飛山真理

VME からは word access を前提とする。

vme_membase <- メモリーのベースアドレス。0 を設定。

vme_iobase <- I/O アクセスのベースアドレス。\$200000 を設定。

1) フィルターボードの読みとり方法 (これは大丈夫のはず)

A) stop address に関係なく、メモリーの頭から読む場合

```
for k:= 0 to 127 do
  for j := 0 to 255 do
    for i := 0 to 31 do {ボードのループ}
      begin
        v1 := j+k*256;
        VME_address := vme_membase+ i*65536+(v1)*2;
        mem[i+v1*32] := byte(vme_16read(VME_address));
      end;
```

説明)メモリーマップは以下のようになっている

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ボード切り替え					ボード毎のメモリーアドレス															

word アクセスでデータを読みとった場合、上位 8 ビットと下位 8 ビットは同じデータが入っているはず。(A バンク B バンクの異なるメモリー上のデータだが、同じデータを書き込んでいるので同じはず)

あるデータの次のクロックのデータは、となりのボードにいつている。ボードは 16 枚あるが、中ではさらに 2 つに分かれていて、計 32 個見る必要がある。

B) stop address から読み込む場合

```
stop_address := vme_16read(vme_iobase+8);
for k:= 0 to 127 do
  for j := 0 to 255 do
    for i := 0 to 31 do {ボードのループ}
      begin
        v1 := j+k*256;
        v2 := (v1+stop_address+1) mod 32768;
        VME_address := vme_membase+ i*65536+(v2)*2;
```

```
mem[i+v1*32] := byte(vme_16read(VME_address));  
end;
```

説明)

stop_address の一つ先から読み始める。各ボードへの書き込みは同時なので、ボード毎の時間差は気にする必要はない。ボード内アドレスが 32767 までいくと、次は 0 に戻って stop_address まで読みとる。

I/O 操作について

vme_iobase +0

0: 0 でフィルター動作、1 で停止=VME よりメモリアクセス可)

1: 0 でアドレス enable、1 で disable ???

2: 0 でクリア、1 で動作 ???

3: 1 で同期、0 で非同期

4: 1 で 2 倍モード

5: 1 で 4 倍モード

6: 1 で 8 倍モード

7: 1 で 16 倍モード ???

4,5,6,7 が全て 1 の時はスルーモード

1) シンクロナイズ

vme_iobase に 1、9、12 を書く。

2) タップ設定

vme_iobase+4 がタップ 1、vme_iobse+8 がタップ 2

3) ステータス

vme_iobase を読む

0: 0 でフィルター動作中、1 で停止中

1: 1 でクロック動作、0 でクロック停止

2: 1 でファン動作、0 でファン停止

3: 1 で同期モード、0 で非同期モード

4: 0 で演算モード、1 でスルーモード

5: 1 で 2 倍

6: 1 で 4 倍

7: 1 で 8 倍

2) メモリーボード読みとりソフトの読み方

B) Stop address から読み始める時

```
const b_array : array[0..15] of integer = (0,1,4,5,8,9,12,13,16,17,20,21,24,25,28,29);
```

```
function s_ad(i : longint);  
begin  
  s_ad := (i+sb0) mod 20;  
end;
```

注：上位アドレスの値を計算する。ストップアドレスは sb0 に入っている。ループの値(i) は 0 から 19 まで変わるので、これからアクセスすべき値を計算する。

```
function a_ad(i : longint) : longint;  
var temp : longint;  
begin  
  temp := i+a_0;  
  if (temp = 32768) then  
    begin  
      sa := sa+1;  
      vme_16write(vme_iobase+4,s_ad(sa));  
    end;  
  a_ad := (i+a_0) mod 32768;  
end;
```

注：下位アドレスの値を計算する。ストップアドレスは a_0 に入っている。アドレスのエンド返くると、頭に戻るだけでなく、上位アドレスを一つ進める必要がある。これは 1 回しかしてはならない。

```
stop1 := vme_16read(vme_iobase+8);    下位アドレスのストップポイント  
stop2 := vme_16read(vme_iobase+12);   上位アドレスのストップポイント  
stop1 := stop1 and $7fff;  
stop2 := stop2 and $1f;   マスク。多分、変なデータは入っていないはずだが、念のため。
```

```
stop_address := stop1+stop2*32768;    全体のストップアドレス
```

```
sb0 := (stop_address+1) div 32768;    ストップアドレスから一つ先が先頭  
a_0 := (stop_address+1) mod 32768;
```

```

sa := 0; s := 0;
vme_16write(vme_iobase+4,s_ad(sa)); 読み始めの上位アドレスをセット
for s := 0 to 19 do
  for a := 0 to 32767 do
    for b := 0 to 15 do
      begin
        val := vme_membase +
              (b_array[b]*32768+
              a_ad(a))*2;
        d := vme_16read(val);
        if odd(b) then
          begin
            mem[b*2-1+a*32+s*1048576] := up(d);
            mem[b*2+1+a*32+s*1048576] := down(d);
          end
        else
          begin
            mem[b*2+a*32+s*1048576] := up(d);
            mem[b*2+2+a*32+s*1048576] := down(d);
          end;
        end;
      end;
    end;
  end;
end;

```

説明)

メモリーマップは以下のようにになっている。

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ボード切替					ボード内下位アドレス															

これとは別に、ボード内上位アドレスを iobase+4 に 0 から 19 まで設定する。

ワードアクセスをしたとき、出てくるデータは、[d15..d8][d7..d0]の上位、下位の8バイトずつ、それぞれボード番号が0なら(0、2)、1なら(1、3)という風に入っている。ここで重要なポイントがある。ボード0のデータとボード2のデータは全く同じとなっている。そこで、ボード切り替えをbとすると、

B	上位	下位	B	上位	下位
0	0	2	16	16	18
1	1	3	17	17	19

2	0	2	18	16	18
3	1	3	19	17	19
4	4	6	20	20	22
5	5	7	21	21	23
6	4	6	22	20	22
7	5	7	23	21	23
8	8	10	24	24	26
9	9	11	25	25	27
10	8	10	26	24	26
11	9	11	27	25	27
12	12	14	28	28	30
13	13	15	29	29	31
14	12	14	30	28	30
15	13	15	31	29	31

のようなマップとなっている。

ロングワードでアクセスする場合

ボード切り替えはワードアクセス時と同じく $Adr[20-16]$ を使う。但し、0~7 までしか使わない。

ボード番号	D31-24	D32-16	D15-8	D7-0
0	3	1	2	0
1	7	5	6	4
2	11	9	10	8
3	15	13	14	12
4	19	17	18	16
5	23	21	22	20
6	27	25	26	24
7	31	29	30	28

$640\text{kbyte} = 655,360 = 0xA0000 = 1010\ 0000\ 0000\ 0000\ 0000$

各ボード $20,480 = 0x5000 = \quad\quad\quad 0101\ 0000\ 0000\ 0000$

読むこと。

$20\text{Mbyte} = 20,971,520 = 0x1400000 = 0001\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000$

バンクが 20 あるので、1 バンクにつき 1,048,576 データ

各ボードは $32,768 = 0x8000 = 1000\ 0000\ 0000\ 0000$

読むこと。