

B188 での PSB 動作検証

長谷川慧

更新：2009年12月7日

1 はじめに

検査項目

- レジスタ読み書きのテスト
- レジスタ “Module” 値の確認
- sld all test pulse のテスト
- asd all test pulse のテスト

を纏める。

検査はリセットケーブルの接続や、asd tp の為に準備するシグナルケーブルの本数の都合から mother 1,2 の二つに分けて行う事にする。

特記すべき準備物は asd tp 用にシグナルケーブル 10 本 + ASD 20 個。

2 セットアップ

まず検査 PSB に対して、必要な配線を準備する。

- 電源ケーブル (電源より)
- clock 配布線 (SPP より。等長の LVDS ケーブル。L1A,tp も運ぶ)
- CAT6 (SSW より。適切な socket とつなく)
- Reset ケーブル (SPP または隣の PSB とつなく)
- シグナルケーブル + ASD (threshold は気にしない)

特に reset ケーブルに関しては、接続に際してインターフェイスを合わせる必要がある。例えば ESD 1 に対しては SPP → ESD 0 → ESD 1 というように中継させる。さらに、中継している PSB に対しても

- 電源供給線 (Reset signal が AND 素子を通るため)
- LVDS 線 (コントロールのため。厳密に必要なかわからないけど、まあつけておく)
- コントロール用 CAT6 (Adjacent や OR チャンネルが有る場合、調べたい PS ボードの読み出しに影響が出ないようにマスクするため。)

を行う。

3 計算機 (PC,SBC)

作業環境は、コミショニングで使い慣れた ManualTtc 等を使えるように、tdaq-01-09-01 を使う。そのために.bashrc には”source ~/testDAQ/.setup191.sh” としてあるが、何らかの変更があって、外してある場合は、手で source する。

執筆時 (2009 年 4 月) の段階での、PSB 検査時の PC,SBC の役割を書いておく。後述する各種作業は、適切な計算機上で行う事とする。

```
tgjpc1 pt5 データ読み出し
tgjvme2 CCI,TTCvi コントロール
tgjvme3 pt5 configuration via VME bus
```

4 パラメタファイル

ファイルの生成場所、マスクの準備などに便利なファイルの置き場所を示しておく。これらファイルから使うパラメータを抽出^{*1}して使う。詳しくは後述する試験手順を参照。

```
parameter file : ~/parameter/psb,ssw
    PSB, SSW のパラメータファイル。
PSB Module Type : ~/parameter/common/BW
    PSB の module type の設定ファイル。
```

5 PS ボード試験

5.1 PsReset

まず PSB, PsReset が働く事を確認する。

```
ManuatTtc 0xff1 W Bchan/PsReset do
JtagAccess -i hogehoge.prm -w
JtagAccess -i hogehoge.prm -r
ManuatTtc 0xff1 W Bchan/PsReset do
JtagAccess -i hogehoge.prm -r
```

作業の意味を説明する。

- 1 行目 電源 ON 後のレジスタ操作をする前のリセット
- 2 行目 適当な psb parameter を入れた hogehoge.prm でコンフィグする
- 3 行目 読み返して問題が無い事を確認

^{*1} grep で psb の address 0x01820003 など引っかけるとよい。

4 行目 PS reset 発行

5 行目 PS reset により、パラメータがデフォルト値に戻っている事を確認

5.2 レジスタ読み書きテスト

この作業は (執筆時) 最新のソフトを使うので環境を変える。CCI が触れる sbc でターミナルを一つ新しく開いてそれをつかうとよい。

```
cd /home/tgc07/users/okumura/project/tgc_online.tmp
source setup-dev.sh
```

その後、そのターミナルで以下のようにする。

```
testPsboard -c 0x01000000 -s 0x0083000000 -m 0x00000004
```

- -c : CCI のアドレス。B-188 テストベンチでは変更しない
- -s : 使う SSW の ID
- -m : 使う SSW の socket (mouth) 番号

5.3 Module type の確認

PS ボード、Module レジスタの読み出しを確認する。節 4 で書いたファイルから抽出してきて準備すれば良い。

```
JtagAccess -i module.prm -r
```

5.4 slb test pulse

まず PSB,SSW の Configuration file を準備する。

検査対象の PS ボード の分は slb all 用パラメータ、PS Reset 中継用の PS ボードの分は slb no track 用のパラメータを、節 4 で書いておいたファイルから抽出して準備しておく。

SSW のパラメータも、基本的には PS ボードと同様に parameter file から抽出して作る。ただし PS ボード一個だけでの試験を行うので、専用の Tx, Rx を用意する。

RxInitialSettingRx

図 1 を参照する。RxInitialSettingRx* を決めるには RXFPGA *番に、それぞれ何個の Slb が付いているかを見る。例えば RxFPGA0 には SSW0,1 それぞれ 2 個の合計 4 個がある。この 4 個 (時には 3or2 個) それぞれに対して、current, next,previous のそれぞれのバンチ読み出しを決定する。ON=1 が” 使う” の意味。さらに 1 bit で edge の選択を決め打でする。

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
register	edge = 1	slb3-n	slb3-c	slb3-p	0	2-n	2-c	2-p	0	1-n	1-c	1-p	0	0-n	0-c	0-p

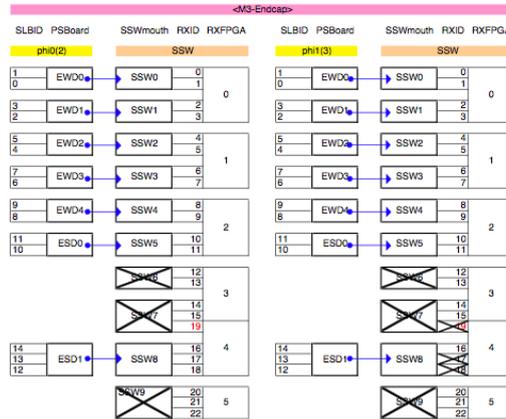


図1 SSW の便利テーブルの例。

TX MASKL, MASKH Tx の mask は L が RxID 11-0、H が 22-12 までを担当する。図表 1 を見ながら、使っていない Rx の ID を mask する。0 が mask をする、の意味。

bit	11	10	9	8	7	6	5	4	3	2	1	0
mask L	RxID 11	10	9	8	7	6	5	4	3	2	1	0
mask H	-	RxID 22	21	20	19	18	17	16	15	14	13	12

検査の手順は次の通り

- PS Reset
- PSB config (検査対象 PSB, 連携用 PSB 共に)
- SSW reset, config, mask
- pt5 config
 - これには PT5 の刺さっている VME の sbc で “PT5_config” を実行する。
- Depth の設定
 - 19 が良い (適当に探して)
 - depth_set コマンドが用意されている。usage が出るのでそれを参照。
- DAQ を立ち上げておく
 - これには PT5 からの S-Link ケーブルが届いている計算機で SectorDAQ.D を使う。GUI が立ち上がるので後は分かる。
- test pulse 発行
 - “ManualTtc 0xff1 W Bchan/TpTrigOn do”。渡す値は 37 と 6a。
- DAQ start, 一定時間で終了
- test pulse 停止
 - “ManualTtc 0xff1 W Bchan/TpTrigOff do”

たまに PT5 がこけるが、config し直せばよい。

データは ~ /DSTdat 以下に、TGCBrun***.dst として出来る (***)は run 番号)。デコードしてヒストグラム化するまでのスクリプトを作っておいた。

```
cd ${HOME}/DSTdat
sh profile.sh ***
```

ヒットが読み出せているか確認する。ヒストグラムを profile.root に吐き出させてあるのでそちらでズームなどして。

6 asd all tp

ASD all tp は基本的に前節 slb test pulse と変わらない。特別な事を以下に挙げる。

- シグナルケーブル,ASD を配線する。threshold のことは気にしなくていい。
- PS ボードに対して、asd tp 用のパラメタファイルを準備する。
- depth は 19 くらいが良い。

最適 depth 値が違っててもよく (ケーブル長まちまちだし)、数回に分けて run を走らせるなどして、すべてのチャンネルに欠けが無い事を確認する。

ただし、検査中の PSB 上の SLB が担当するチャンネルに注意する。つまり、SLB が buffer に入れるのは自分の PSB 上の PP への入力だけでなく、隣の PSB,PP への入力分も受ける場合がある。どの Slb がどのチャンネルを担当するかは資料 (<http://www.hepl.phys.nagoya-u.ac.jp/hasegawa.nagoya/archives/mappingTable.elsx>) をみる。

付録 A PS ボードの組み替え

交換用 PS ボードが無い場合は、 $C \rightarrow A, A \rightarrow C$ の組み替えが必要になる。組み立ての際には次の点に気をつける。

mother ボードの上下反転

組み替えの際には、PS ボードを構成している 2 枚の mother board の上下が反転する。その結果、どちらのサイドでも、phi0 あるいは 2 がラダー側に来る。PS8 を作っている、FWD 0,1 の上下関係は、どちらのサイドでも FWD 0 がラダー側である。

DCS ID

PS ボードに乗っかっている DCS の ID にも注意する必要がある。ID は図 2 に載せた。ID は、写真 3 の用に 8bit ある dip switch で設定する。まず、1,2 は ID ではないので触らず ON(bit 番号が書いてある側) にしておく。3-8 で ID を決めている。3 が MSB(2^5),8 が LSB(2^0) である。

付録 B LTP の初期設定

電源投入後の LTP の設定では Clock, Trigger ソースとして CTP link in をみるようになっている。これをそれぞれ内部クロック、Front Panel(lemo input) に変えてやる必要がある。menuRCFLtp コマンドを使って変更する。

```
$ menuRCDLtp
```

phi0/1>		phi2/3>		phi0/1>		phi2/3>	
PsBoard	eLMB	PsBoard	eLMB	PsBoard	eLMB	PsBoard	eLMB
FWD0-0	1	FWD0-1	21	FT0-0	40	FT0-1	50
FWD1-0	2	FWD1-1	22	FT1-0	41	FT1-1	51
FSD0-0	3	FSD0-1	23	EWT0-0	42	EWT0-2	52
EWD0-0	4	EWD0-2	24	EWT0-1	43	EWT0-3	53
EWD0-1	5	EWD0-3	25	EWT1-0	44	EWT1-2	54
EWD1-0	6	EWD1-2	26	EWT1-1	45	EWT1-3	55
EWD1-1	7	EWD1-3	27	EWT2-0	46	EWT2-2	56
EWD2-0	8	EWD2-2	28	EWT2-1	47	EWT2-3	57
EWD2-1	9	EWD2-3	29	EST0-0	48	EST0-2	58
EWD3-0	10	EWD3-2	30	EST0-1	49	EST0-3	59
EWD3-1	11	EWD3-3	31				
EWD4-0	12	EWD4-2	32				
EWD4-1	13	EWD4-3	33				
ESD0-0	14	ESD0-2	34				
ESD0-1	15	ESD0-3	35				
ESD1-0	16	ESD1-2	36				
ESD1-1	17	ESD1-3	37				

図 2 DCS カードの ID 表

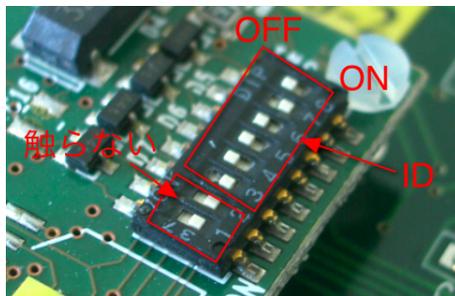


図 3 DCS ID の dip switch

<base>?
0xff4000

あとは適当にいじればわかる。

付録 C PT5 Configuration での error

以下のメッセージは P55 が正しく VME クレートに挿入されていないときに見られる。あるいは、Daughter ボードの接触を確認する。

```
err> bus error at 0910001c, am = 09, # of bus errors = 1
err> bus error at 0910001c, am = 09, # of bus errors = 1
err> bus error at 0910001c, am = 09, # of bus errors = 1
err> bus error at 0910001c, am = 09, # of bus errors = 1
err> bus error at 0910001c, am = 09, # of bus errors = 1
err> bus error at 0910001c, am = 09, # of bus errors = 1
```