INTTの初期化について

蜂谷　崇

はじめに

- Half Entry問題
  - データ信号についてパルス測定からわかった事
    - パルス幅が狭い　(5ns -> 3.5ns)
    - パルスのjitterがある(1ns程度)
  - 考えられること
    - データ受信時を上手く取り込めていないのでは。
  - 対策
    - Jitterを減らす：　FPHXチップの初期化に失敗しているせいではないかと予想し、初期化が成功するまで、リセットを繰り返す。
  - 疑問：
    - SlowControlの設定で何が初期化できるのか。

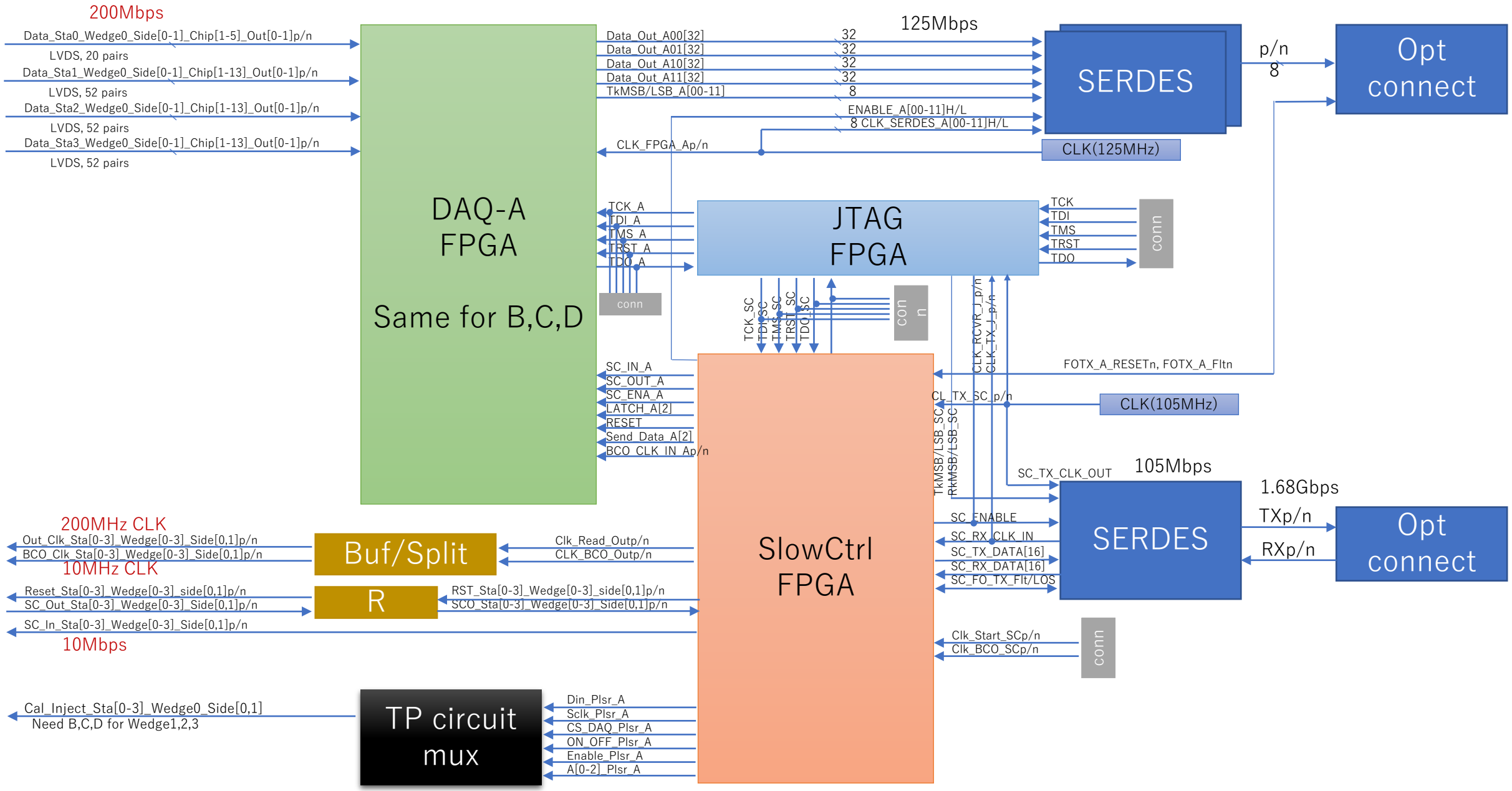
GlobalStartボタンのコマンド列

- send_fo_sync()　：　SlowControlのFiberをSyncする
- send_fpga_reset()　：　FEM, FEM-IB, ROCのFPGAをリセットする
- time.sleep(2)
- #send_fo_sync()
- send_reset(regpanels)　：　FFR　FPHXチップをハードリセットする
- send_init(regpanels)：　GUIの初期値をFPHXに送る
- send_enable_ro(regpanels)　：　FPHXをReadOutデータ出力モードにする
- send_latch()：　ROCのデータファイバをSyncする。FPHXチップからのデータパルスのPhaseを決める
- #send_latch()
- #send_latch()
- send_fem_lvl1_delay(int(fem_lvl1_delay_var.get()))
- send_pulse_module(pulse_module_var, pulse_wedge_var, femaddr_var)
- send_bco_start()：　FEMとFPHXにBCOを送り、ビームカウンタをスタートさせる
- send_calib()
- start_daq_prog(regpanels)

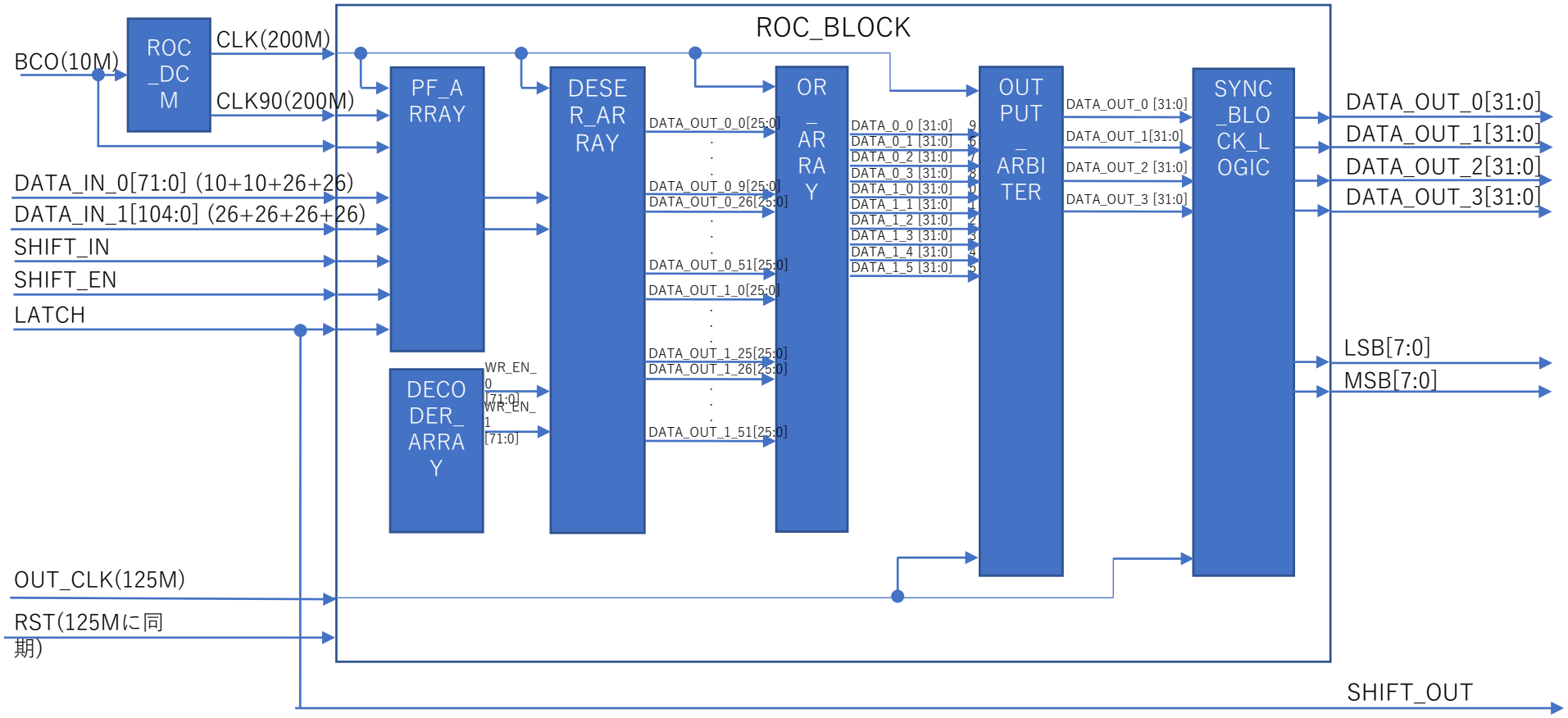
SlowControlコマンド

If you would like to run manually, some of the basic DAQ buttons are described here, in the order in which they are generally pushed when collecting data (and the buttons can be found in the image below, primarily in the upper right-hand corner):

- **FPGA Reset:** This issues a reset to the FPGAs on the FEM, FEM Interface and ROC boards. It should be issued after you first power up, and we generally issue one before data collection. The FPGA Reset sequence ends with a SC FO sync command being sent so you should see the FO Sync LED light up at the end of RESET also.
- **FO Sync:** This generates the fiber optic synchronization sequence on the slow control fiber. If the synchronization has been successfully achieved in both directions then the top, left-most LED on the FEM will light up. If it isn't achieved, just hit the button again until sync is achieved.
- **FFR:** This used to be the general reset button but now it just issues a fire-fighter reset to the FPHX chips. This should be issued after power-up of a wedge and before sending an INIT command.
- **INIT:** Download parameters to FPHX chips. This needs to be issued after FFR. If INIT has worked properly, you should see the digital current draw for the wedges drop from the value that you have after a FFR.
- **Enable RO:** This sends and enable readout command to the FPHX chips. It is necessary to hit this after an INIT to get the data words to come out of the FPHX chips.
- **Latch FPGA:** This LATCH command serves two purposes - first, it initiates the fiber optic synchronization command to the ROC data fibers. If synchronization is achieved, then all 8 lights for each fiber that is connected to the FEM should become lit on the FEM front panel (3rd and 4th rows of LEDs go with the bottom fiber on the FEM, 5th and 6th rows go with the top fiber on the FEM). Second, the LATCH readies the ROC data FPGAs for data collection by determining the phase of the data lines coming from the FPHX chips.
- **Start DAQ** This readies the NI for data collection.
- **BCO Start** This command is needed to synchronize the beam-clock counters on the FPHX chips and the FEM. It simultaneously issues a start counter command to the FEM and down to the FPHX chips. The result is that the two will be counting in sequence but it does **not** cause the BCO clock for a calibration, for instance, to be on the same clock each time you run calibration.
- **Calib:** Issue a calibration command to the ROC board. After the command is received, the ROC will initiate a full calibration sequence for the chips.
- **Set Module:** This must be set each time you want to change the module that is being calibrated, but does not need to be sent again for subsequent runs that are to use the same module. This button is a bit of a misnomer right now because it actually selects one of 8 possible "sides" to calibrate in a phi slice. "0" corresponds to station 0, side 0, "1"=station 0, side 1, "2"=station 1, side 0, etc. This should be changed to specify the phi slice, the module, and side to be calibrated, and allow for multiple modules to calibrate at once.


200Mbps
Data_Sta0_Wedge0_Side[0-1]_Chip[1-5]_Out[0-1]p/n
LVDS, 20 pairs
Data_Sta1_Wedge0_Side[0-1]_Chip[1-13]_Out[0-1]p/n
LVDS, 52 pairs
Data_Sta2_Wedge0_Side[0-1]_Chip[1-13]_Out[0-1]p/n
LVDS, 52 pairs
Data_Sta3_Wedge0_Side[0-1]_Chip[1-13]_Out[0-1]p/n
LVDS, 52 pairs
DAQ-A FPGA
Same for B,C,D
Data_Out_A00[32]
Data_Out_A01[32]
Data_Out_A10[32]
Data_Out_A11[32]
TkMSB/LSB_A[00-11]
32
32
32
32
8
125Mbps
SERDES
p/n
8
Opt connect
ENABLE_A[00-11]H/L
8 CLK_SERDES_A[00-11]H/L
CLK_FPGA_Ap/n
CLK(125MHz)
TCK_A
TDI_A
TMS_A
TRST_A
TDO_A
conn
JTAG FPGA
TCK
TDI
TMS
TRST
TDO
conn
TCK_SC
TDI_SC
TMS_SC
TRST_SC
TDO_SC
con n
CLK_RCVR_J_p/n
CLK_TX_J_p/n
SC_IN_A
SC_OUT_A
SC_ENA_A
LATCH_A[2]
RESET
Send_Data_A[2]
BCO_CLK_IN_Ap/n
FOTX_A_RESETn, FOTX_A_Fltn
CL_TX_SC_p/n
CLK(105MHz)
TkMSB/LSB_SC
RkMSB/LSB_SC
SC_TX_CLK_OUT
105Mbps
1.68Gbps
SC_ENABLE
SC_RX_CLK_IN
SC_TX_DATA[16]
SC_RX_DATA[16]
SC_FO_TX_Flt/LOS
SERDES
TXp/n
RXp/n
Opt connect
SlowCtrl FPGA
200MHz CLK
Out_Clk_Sta[0-3]_Wedge[0-3]_Side[0,1]p/n
BCO_Clk_Sta[0-3]_Wedge[0-3]_Side[0,1]p/n
Buf/Split
Clk_Read_Outp/n
CLK_BCO_Outp/n
10MHz CLK
Reset_Sta[0-3]_Wedge[0-3]_side[0,1]p/n
SC_Out_Sta[0-3]_Wedge[0-3]_Side[0,1]p/n
R
RST_Sta[0-3]_Wedge[0-3]_side[0,1]p/n
SCO_Sta[0-3]_Wedge[0-3]_Side[0,1]p/n
SC_In_Sta[0-3]_Wedge[0-3]_Side[0,1]p/n
10Mbps
Clk_Start_SCp/n
Clk_BCO_SCp/n
conn
Cal_Inject_Sta[0-3]_Wedge0_Side[0,1]
Need B,C,D for Wedge1,2,3
TP circuit mux
Din_Plsr_A
Sclk_Plsr_A
CS_DAQ_Plsr_A
ON_OFF_Plsr_A
Enable_Plsr_A
A[0-2]_Plsr_A

DATA_FPGA


ROC_BLOCK
BCO(10M)
ROC_DCM
CLK(200M)
CLK90(200M)
DATA_IN_0[71:0] (10+10+26+26)
DATA_IN_1[104:0] (26+26+26+26)
SHIFT_IN
SHIFT_EN
LATCH
PF_ARRAY
DECODER_ARRAY
WR_EN_0 [71:0]
WR_EN_1 [71:0]
DESER_ARRAY
DATA_OUT_0_0[25:0]
DATA_OUT_0_9[25:0]
DATA_OUT_0_26[25:0]
DATA_OUT_0_51[25:0]
DATA_OUT_1_0[25:0]
DATA_OUT_1_25[25:0]
DATA_OUT_1_26[25:0]
DATA_OUT_1_51[25:0]
OR_ARRAY
DATA_0_0 [31:0] 9
DATA_0_1 [31:0] 6
DATA_0_2 [31:0] 7
DATA_0_3 [31:0] 8
DATA_1_0 [31:0] 0
DATA_1_1 [31:0] 1
DATA_1_2 [31:0] 2
DATA_1_3 [31:0] 3
DATA_1_4 [31:0] 4
DATA_1_5 [31:0] 5
OUTPUT_ARBITER
DATA_OUT_0 [31:0]
DATA_OUT_1[31:0]
DATA_OUT_2 [31:0]
DATA_OUT_3 [31:0]
SYNC_BLOCK_LOGIC
DATA_OUT_0[31:0]
DATA_OUT_1[31:0]
DATA_OUT_2[31:0]
DATA_OUT_3[31:0]
LSB[7:0]
MSB[7:0]
OUT_CLK(125M)
RST(125Mに同期)
SHIFT_OUT

FPGAコードと回路図を見て分かった事

- DataFPGAには、チップ事のデータをマスクする機能はない。
  - 実装が途中で止まっている。

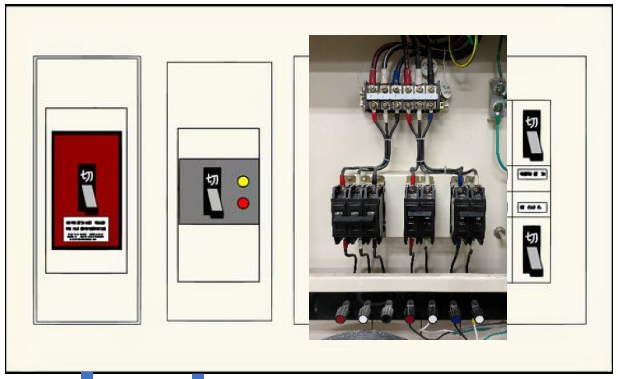
- データ用optic fibreのコネクタをリセットする機能はない
  - LEDが光るとコネクタがFailしている。
  - リセットや状態異常のラインはSlowControlFPGAに繋がっているが使っていない。

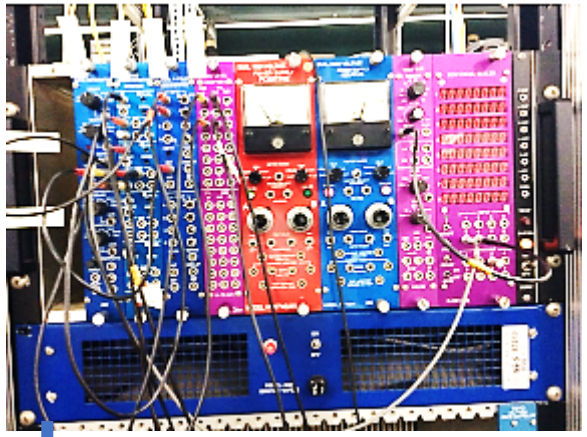
- Latchコマンド
  - 今調べ中です。

疑問

- FEM, FEM-IBは本番でも使うのか？
  - FPHXからの読み出しは、FELIXボードを使う
  - BCO_CLOCKの供給、SlowControlの制御などはどのようにするのか？
    - これまでどおりFEM, FEM-IB？　FELIXボードにSlowControlを制御する機能はあるのか？　→　多分ない。
    - Calibデータはどのようにして取るのか？
      - AmplitudeデータはSlowControlのリードバックラインを経由して吸い上げ、FEMでFPHXからの読み出しデータと結合される。　読み出しデータが別になった場合、これができなくなる。

電源の接続

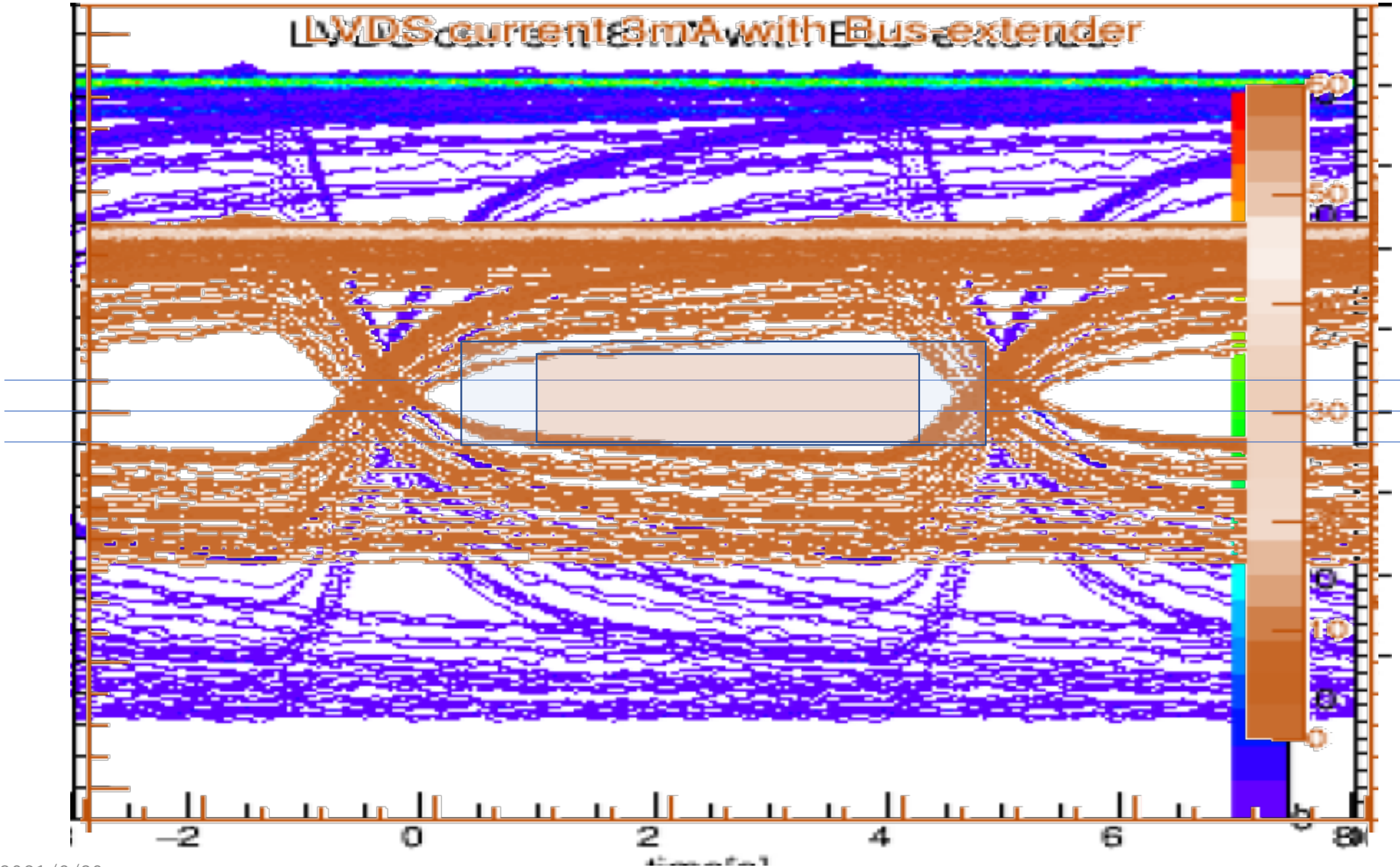




Ladder 1

4つ(ROC 3＋Ladder 1)

- ラダーへの電源が2系統に分かれている。


LVDS current 3mA with Bus extender
60
50
40
30
20
10
0
−2
0
2
4
6
8