

Readbacker updates

蜂谷 崇、高濱 瑠菜

Status

- 前回： ほぼできた。 確認・調整が必要。
→
- 今回： しました。 これで完成(のはず)
- GUIも調整した。
- 開発の間に分かったこともメモ(TIPPS)

Module=15, Side=15のときだけワイルドカード(モジュール・サイド)

Module # (15) → Port Name (A1)

All / D3 | Mod A0 | Mod B0 | Mod C0 | Mod D0 | Mod A1 | Mod B1 | Mod C1 | Mod D1 | Mod A2 | Mod B2 | Mod C2 | Mod D2 | Mod A3 | Mod B3 | Mod C3 |

Reg

Desc

To Chip

From Chip

Chip Command

*

Wild

0

Read

Write

Set255

Reset

Default

1

Mask

0

Read

Write

Set255

Reset

Default

2

Dig Ctrl

0

1

Read

Write

Set255

Reset

Default

3

Vref

0

1

Read

Write

Set255

Reset

Default

4

DAC0

20

8

Read

Write

Set255

Reset

Default

5

DAC1

25

16

Read

Write

Set255

Reset

Default

6

DAC2

30

32

Read

Write

Set255

Reset

Default

7

DAC3

35

48

Read

Write

Set255

Reset

Default

8

DAC4

40

80

Read

Write

Set255

Reset

Default

9

DAC5

45

112

Read

Write

Set255

Reset

Default

10

DAC6

50

144

Read

Write

Set255

Reset

Default

11

DAC7

55

176

Read

Write

Set255

Reset

Default

12

N1Sel <3:0>

0

6

Read

Write

Set255

Reset

Default

N2Sel <7:4>

0

4

13

FB1Sel <3:0>

0

4

Read

Write

Set255

Reset

Default

MaskSel <7:4>

0

0

14

P3Sel <1:0>

0

0

Read

Write

Set255

Reset

Default

P2Sel <7:4>

0

4

15

GSel <2:0>

0

1

Read

Write

Set255

Reset

Default

BWSel <7:3>

0

4

16

P1Sel <2:0>

0

5

Read

Write

Set255

Reset

Default

InjSel <5:3>

0

0

17

LVDS Current

17

170

Read

Write

Set255

Reset

Default

18

Reset

0

Read

Write

Set255

Reset

Default

Chip Control

Display/Modify Configuration for Chip ID 2 Side 0

Channel Mask

[Red = Off, Green = On]

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

Mask All

Unmask All

Toggle All

Send

Chip Side Enable

0

15

0

1

8

15

0

1

16

15

0

1

24

15

0

1

1

15

0

1

9

15

0

1

17

15

0

1

25

15

0

1

2

15

0

1

10

15

0

1

18

15

0

1

26

15

0

1

3

15

0

1

11

15

0

1

19

15

0

1

27

15

0

1

4

15

0

1

12

15

0

1

20

15

0

1

28

15

0

1

5

15

0

1

13

15

0

1

21

15

0

1

29

15

0

1

6

15

0

1

14

15

0

1

22

15

0

1

30

15

0

1

7

15

0

1

15

15

0

1

23

15

0

1

31

15

0

1

Side=1

All以外

Readbackerとその他

Side=15: both
0: top
1: bottom

All以外は0, 1しか動作しないように制限

TestStand
C Spartan3 C ROC C ROC+FEM FEM Addr 6 DB Access
C On C Off

Global Chip/DAQ Operations
Enable RO Latch FPGA Core Reset Start DAQ Check GLINK test
Intr Disable RO Calib JTAG Sync Stop DAQ Check FEM
IO Sync Set I-1 Delay 3 BCO Start Global Start Self Trig
FPGA RST Er. EEPROM Write Page Read Page Write All Cosmic Start

DAQ Configuration
C:/Users/sphenix/Docum Browse
NI DAQ Sample Rate (MHz) 5
Num of Chans (0==inf) 0
Duration (HH:MM:SS) 00:00:00
Print Output Print Off

Run Number 0
Filename C:/Users/sphenix/Documents/INTT_test
Beam Species None
Beam Energy 0

Pulsar Configuration
Pulse amplitude (10 bits max) 255 Config Amp Pulse
Num of Pulses 1 Pulse Train
BCOs between pulses 1023 Wedge 0
Module 0 Set Module

Module Enable
Module 15 On Off Both Side 0 Side 1 Module 7 On Off Both Side 0 Side 1
Module 0 On Off Both Side 0 Side 1 Module 8 On Off Both Side 0 Side 1
Module 1 On Off Both Side 0 Side 1 Module 9 On Off Both Side 0 Side 1
Module 2 On Off Both Side 0 Side 1 Module 10 On Off Both Side 0 Side 1
Module 3 On Off Both Side 0 Side 1 Module 11 On Off Both Side 0 Side 1
Module 4 On Off Both Side 0 Side 1 Module 12 On Off Both Side 0 Side 1
Module 5 On Off Both Side 0 Side 1 Module 13 On Off Both Side 0 Side 1
Module 6 On Off Both Side 0 Side 1 Module 14 On Off Both Side 0 Side 1

Manual Packet Send
Packet file to send Browse Send Read

Communications
C:\Users\sphenix\Documents\INTT_test\daq\daqmon

ReadBackerの使い方

1. ROCのポートに対応するModule#のタブを選ぶ

2. 設定したいチップとサイドを選ぶ

3. FEMのIDを選ぶ(基板上のロータリースイッチ)

4. Readボタンを押す

タブ

The screenshot displays the ReadBacker software interface with several key components highlighted by red boxes and numbered 1 through 3:

- Box 1:** Points to the top navigation bar containing tabs for Module 0 through Module 14. The 'Module 12' tab is currently selected.
- Box 2:** Points to the 'Chip Control' section, specifically the 'Display/Modify Configuration for Chip ID' dropdown menu, which is set to '2 Side 1'.
- Box 3:** Points to the 'Read' button in the 'Reg' table on the left side of the interface.

The interface also includes a 'Global Chip/DAQ Operations' panel with buttons like FEM, Enable RO, Latch FPGA, Core Reset, Start DAQ, Check GLINK, and test. Below this is a 'DAQ Configuration' section with fields for DAQ Program, NI DAQ Sample Rate, Num of events, Duration, and Print Output. The 'Pulsar Configuration' section includes Pulse amplitude, Num of Pulses, and BCOs between pulses. The 'Module Enable' section shows a grid of checkboxes for enabling various modules (Module 0 to Module 14) on both Side 0 and Side 1. At the bottom, there is a 'Manual Packet Send' section with a 'Packet file to send' field and 'Browse', 'Send', and 'Read' buttons.

変更点

- FEM-IB FPGA
 - USBドライバ(FTDI)のRDを読めるように、FPGAのピンアサインをした(A12)
 - USB-IFモジュールの改造
 - RD=Hiのとき、バスの方向をFPGA->USBにして、ReadBackしてきたデータを流す
 - Receiverモジュールの改造
 - ReadBackしてきたデータをずっと保存。(次のReadBackデータが来るまで保存)
- Nevis_gui_for_calib3.py
 - Read_register関数を改造
 - READ (0) インストラクションを使用してレジスタを読み出す
 - 書き込むときはWRITE(1)やSET(2) とすることでレジスタに書き込みつつ、データを読み出す。
 - READ(0) では書き込まず、読み出しデータだけを取り出すことができる。
 - 実行シーケンス
 - Write_bytes_to_target関数でREADコマンドをFPHXチップに送信： 同期BitBangで送信
 - Read_bytes_to_USB関数でFEM_IBに戻ってきたReadBackデータを取り出す： 非同期BitBang
 - Swap_bit: データのビット列を反転する。
- fphxtb.py
 - Read_bytes_to_USB関数を追加
 - Async_bitbangモードでデータバスの情報をIOする。
 - 実装はWrite_byte_to_USBのほぼコピー
 - swap_bit 関数の追加
 - 読み出してきた8bitデータのビット位置を反転する。

2021/6/22

Readbackerとその他

• 01234567 -> 76543210 : FPHXからのデータは0ビット目が先頭になっているので、7ビット目が先頭になるように入れ替える

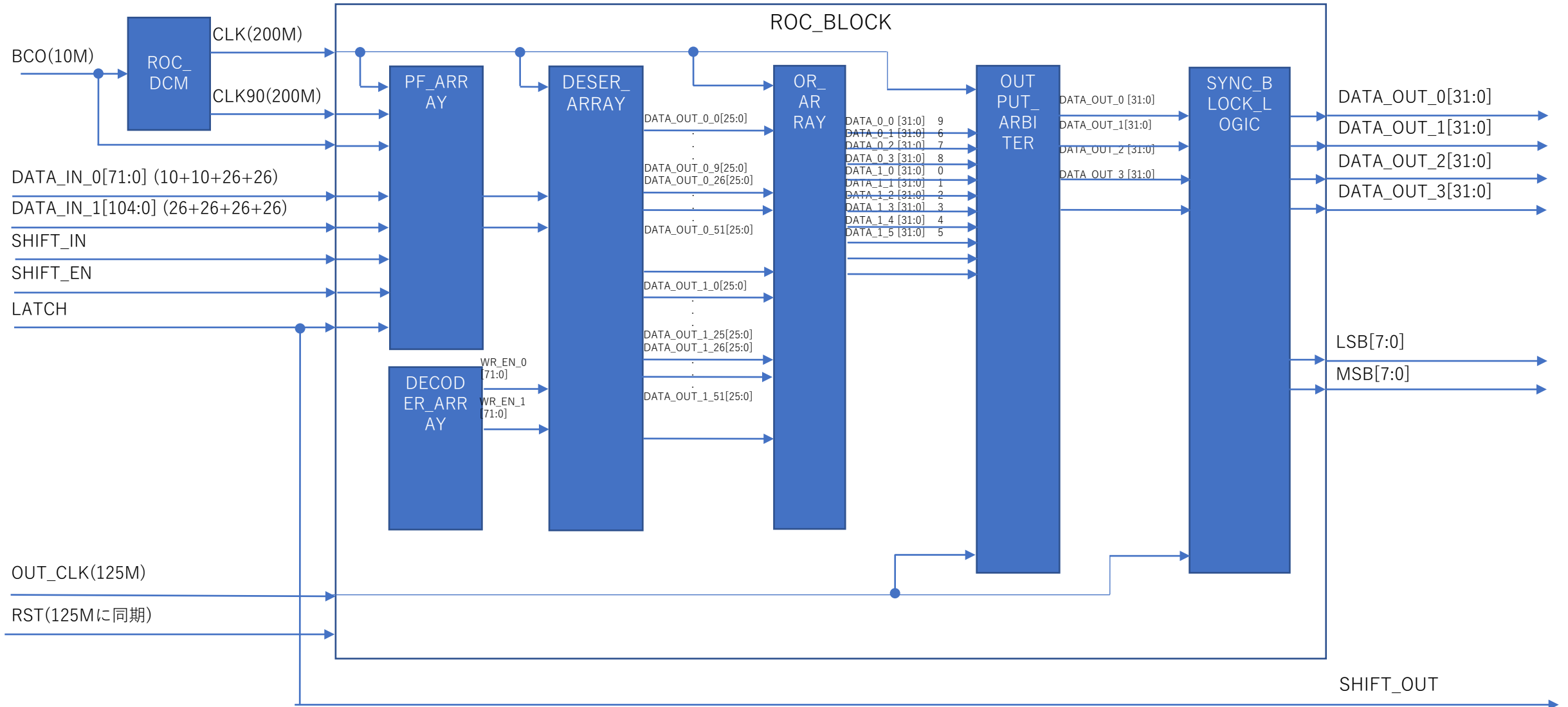
TIPPS

- 動作が変な時の対応
- まずは、GLOBAL_START で全体の動作確認をする。
 - ADCデータが1つでも取れたらOK。 DAQ_STOPする。
- GLOBAL_START後、DAQストップするとFPGAがおかしくなっている。 ReadBackする前にFPGA__RESETする必要がある。
(FFRなどはいらない。FFRすると設定した情報が消える)
- Readボタンを押すとUSB-IFがおかしくなることがある。
 - あらゆるコマンドが効かない。
 - FEM-IB上のリセットボタンを押すと治る

GUIの今後：

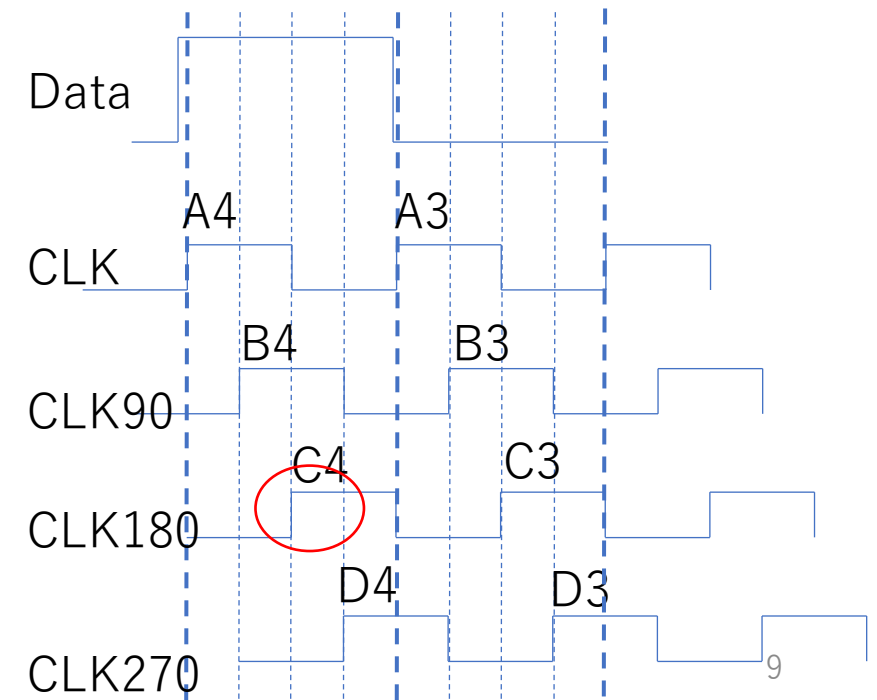
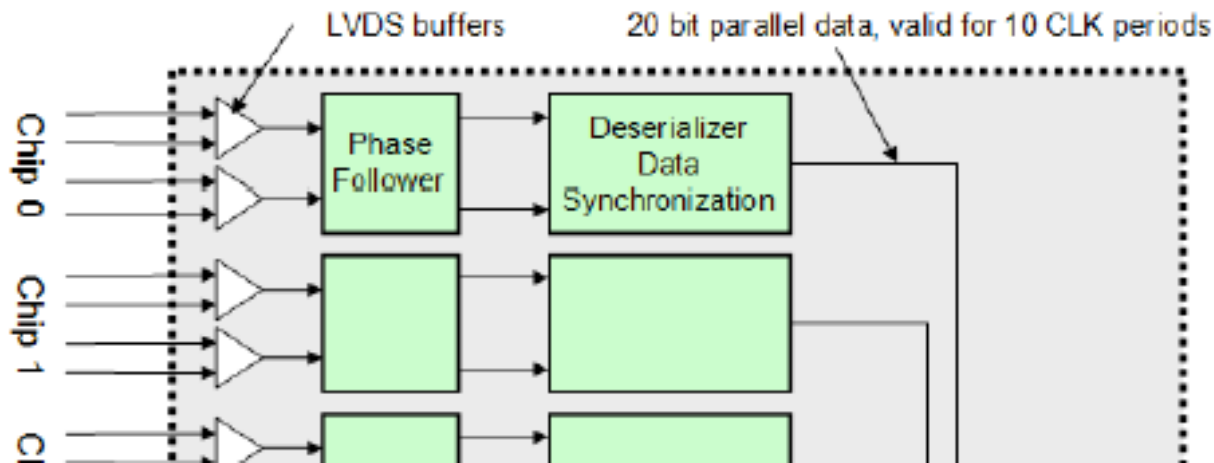
- Chipの全レジスタや、ラダーの全チップなど、まとめて読みだす機能＋表示GUIを追加
 - 今の方法では多分時間がかかる。
 - レジスタ毎にUSB-IFをOpen/Close(+安定化待ち時間)する。
- GlobalStartを押すという解決方法は気持ち悪いので、どうすればそれ以外の方法で動作するかを確認する。
 - 各Stepでなにをやっているか確認する。
 - 例えば、FEM/FEM-IBの電源をOFFにした場合、ROC/モジュールもOFFにする必要があるか？ 今は必要だがなぜ？
 - FPGA_RESET: ROC, FEM, FEM-IBのFPGAにリセット信号を送る
 - FFR: FPHXチップにHWリセットを送る
- 疑問：
 - 電源投入後の正しい起動シーケンスは？
 - FO_SYNC, LatchFPGA, BCO_START, CoreRESET, Enable ROなどは何をしているのか。
 - 実際の運用時のROがうまくいかない時の情報になる

Data Receiver (DATA_FPGA)の構造



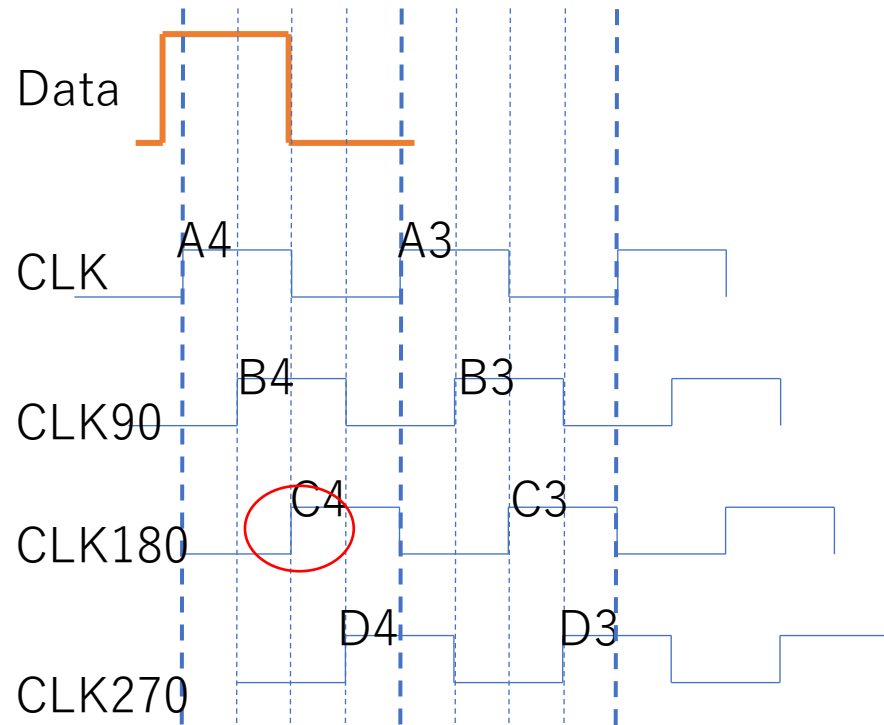
Phase follower

- 26x2 LVDS are connected to Data FPGA
 - No PLL (for each data line) and No clock recovery algorithm.
- Phase follower is implemented at 1st stage of receiving data
 - Algorithm to latch the receiving data with appropriate timing.



Concern with bus-extender

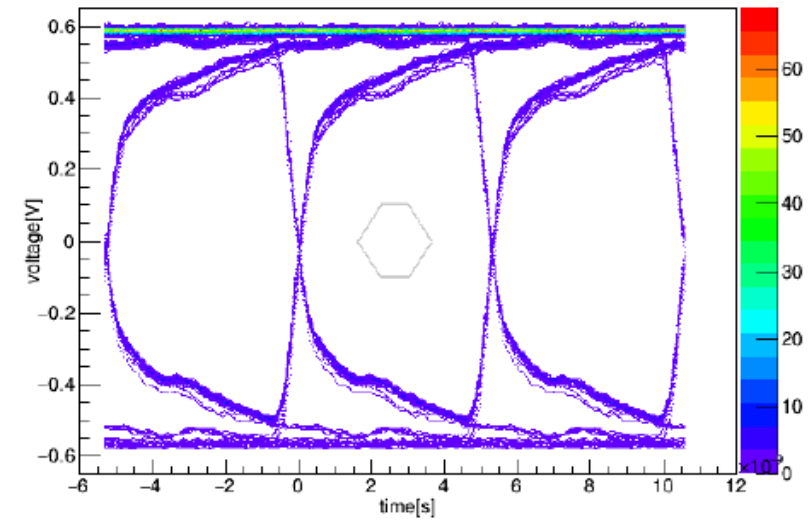
- If the waveform of data is distorted by bus-extender too much, the data timing needs to be reconsidered.
 - 90 degree should be better than 180?



2021/6/22

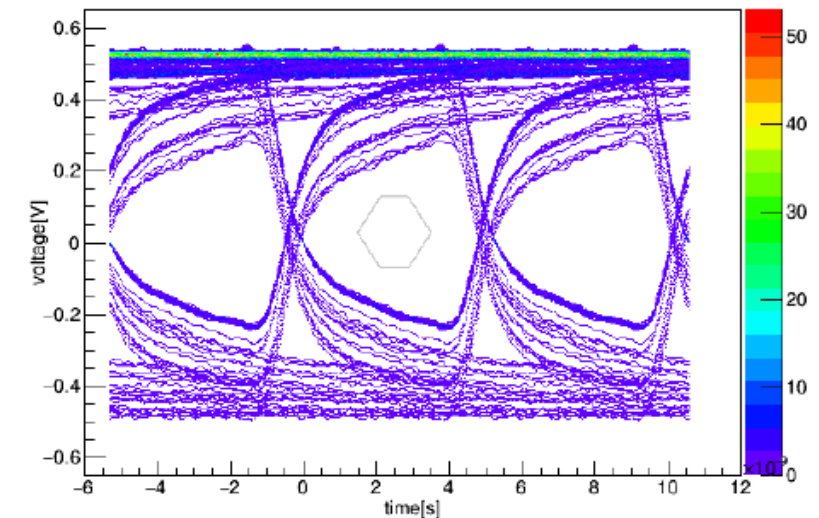
Eye-diagram w/o bus-extender

LVDS current 8mA w/o Bus-extender

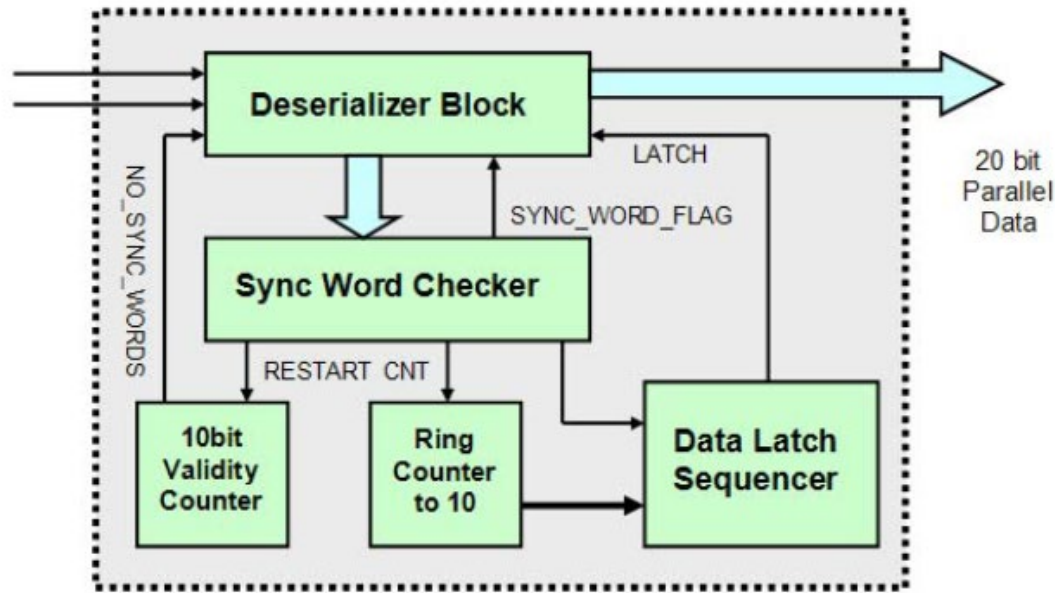


Eye-diagram w/ bus-extender

LVDS current 8mA with Bus-extender



DESER_ARRAYの中身



Deserialized 20 bit Data Word is sampled on every clock for a Sync Word pattern

If Sync Word is found, **ring counter to 10** is restarted

If there were no Sync Words detected in 1024 consecutive clocks, RST is issued to deserializer

Data Latch Sequencer sends **Latch** request whenever

- Sync Word is detected
- Counter reached threshold value (assume Data Word)

If **Latch** is sent for the Sync Word

→ **DATA_OUT = 0 (remove Sync Words)**

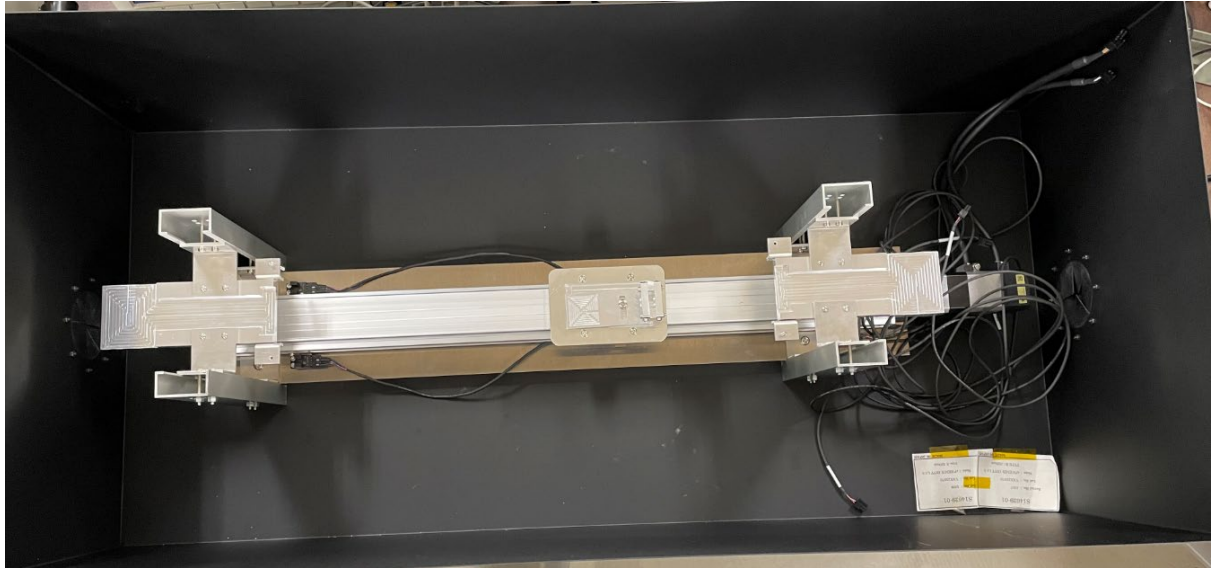
If **Latch** is sent for the Data Word

→ **DATA_OUT = DESERIALIZED_DATA**

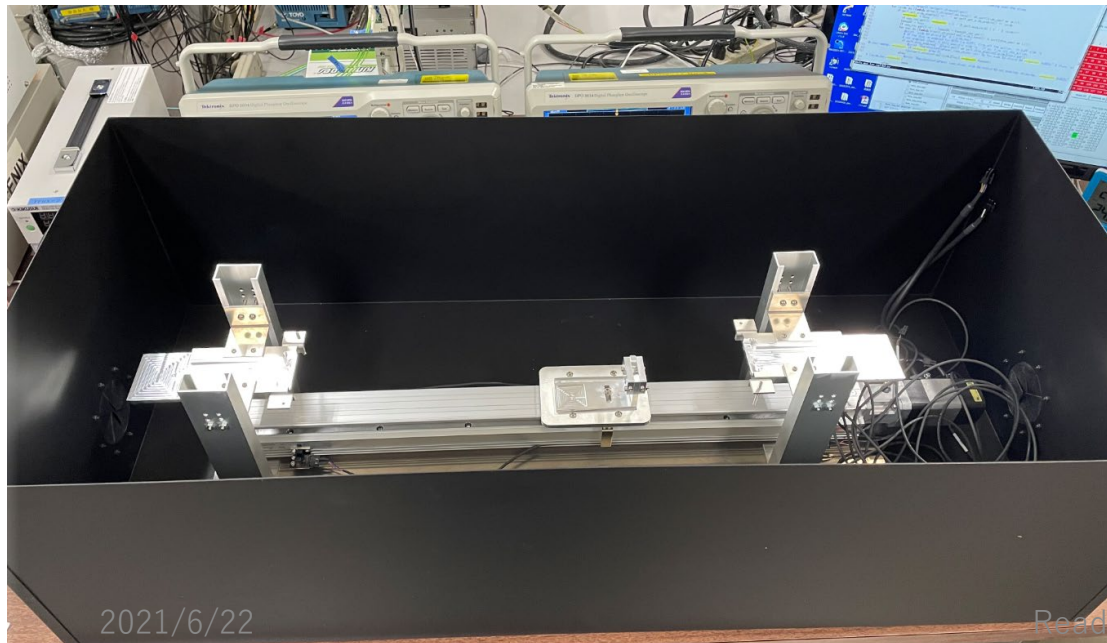
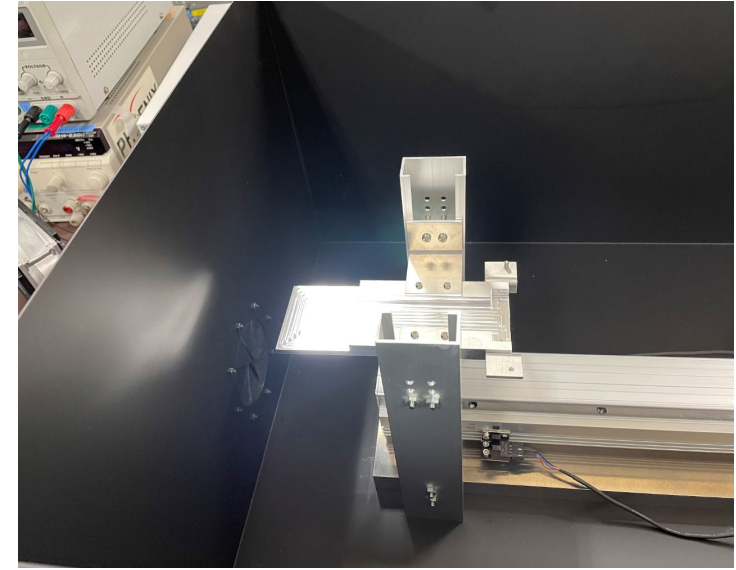
- Sync Word Checker
 - VHDL : sync_block.vhd
 - Sync word : 0000_0000_0000_0001の時
SYNC_WORD=1を出力
- Deserializer_FPHX
 - SYNC_FLAG: 一度Syncワードを発見したらずっと1。リセットされるとSyncが切れる
- 動作
 - CNT = 00101(17のこと)。
 - つまり、データを20Bit分読み込んだ時に相当する(はず)
 - Dataかどうかの判定
 - Syncワード(0000_0000_0000_0001) = Clock、それ以外はデータワード
 - Latch = WRITE_EN
 - カウンタでループする

Chipからの2つのデータラインを1つにマージ(OR)する。Feedbackerとその他

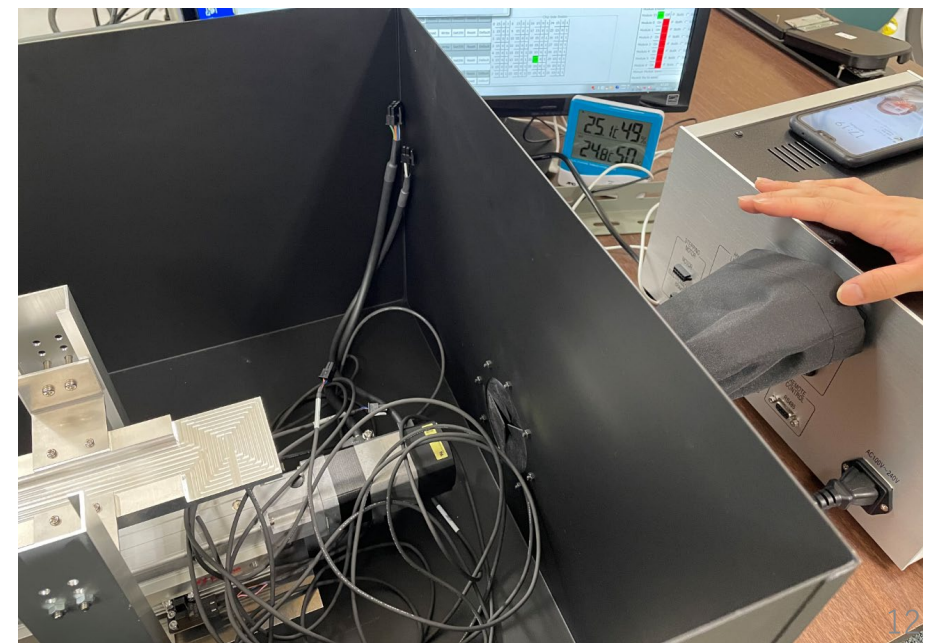
暗箱 上開き



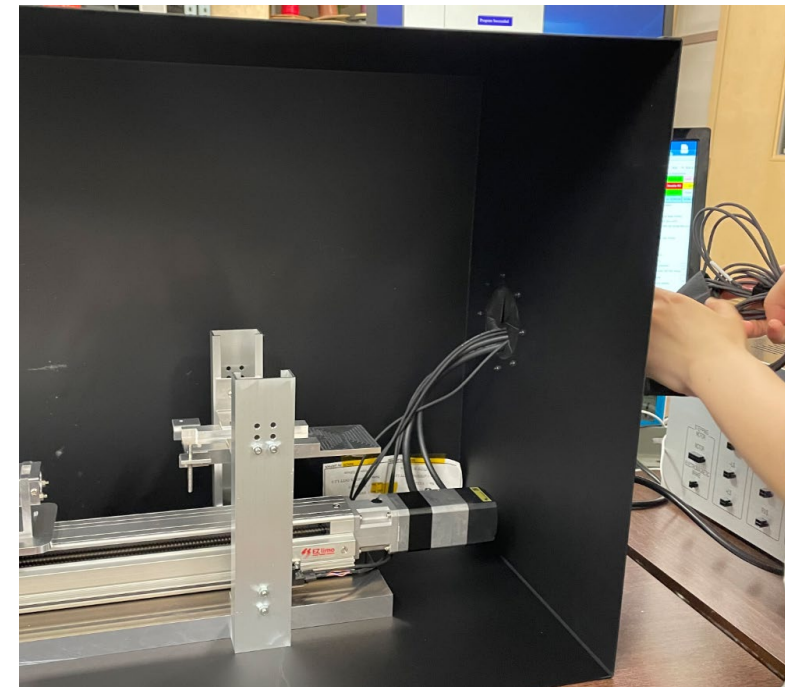
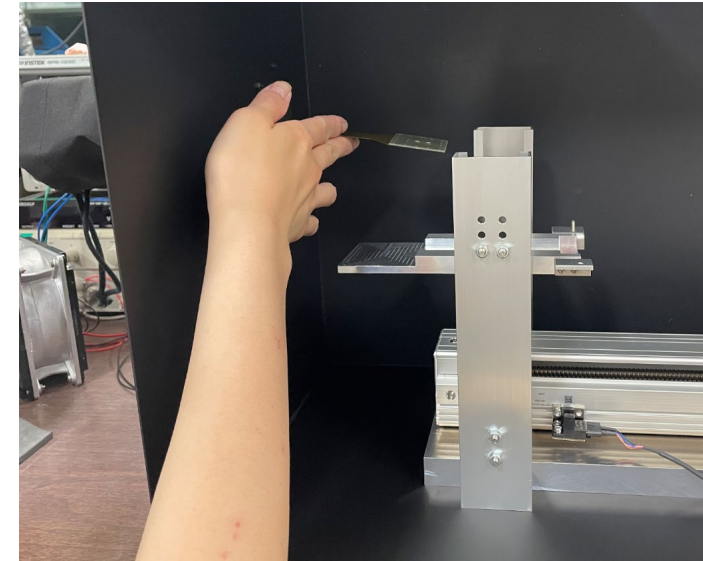
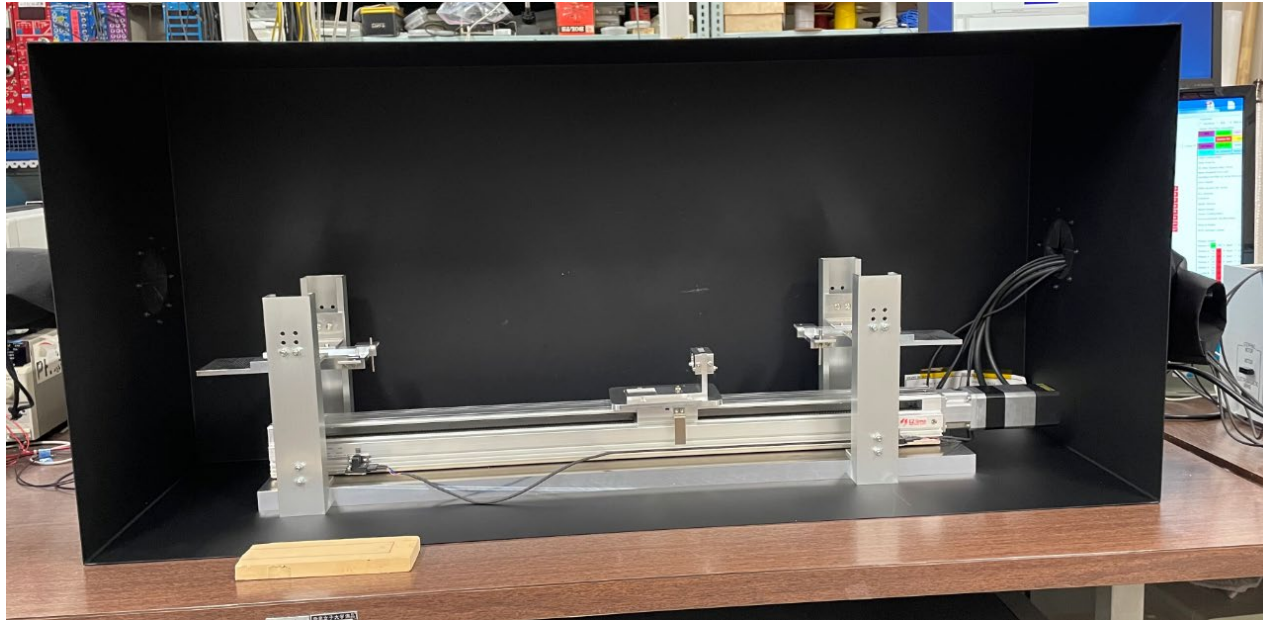
横穴の位置が低い



Readbackerとその他

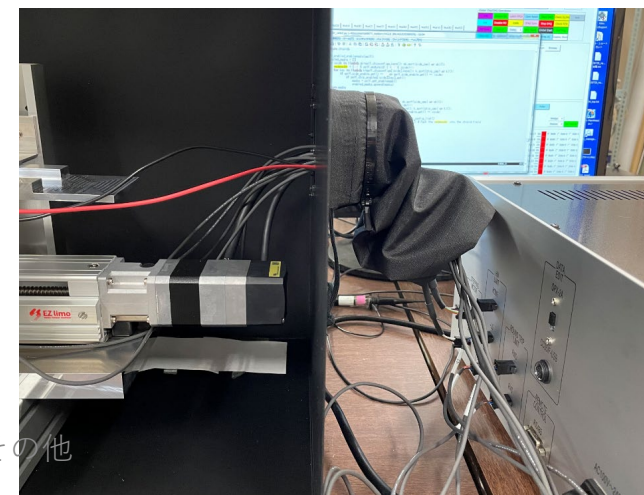
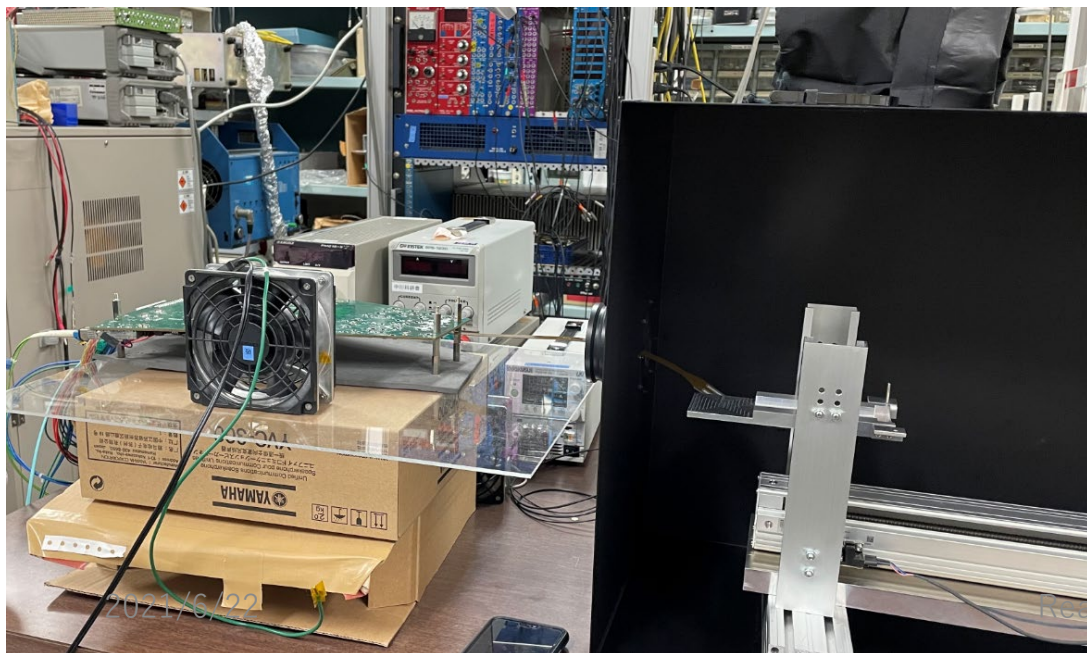
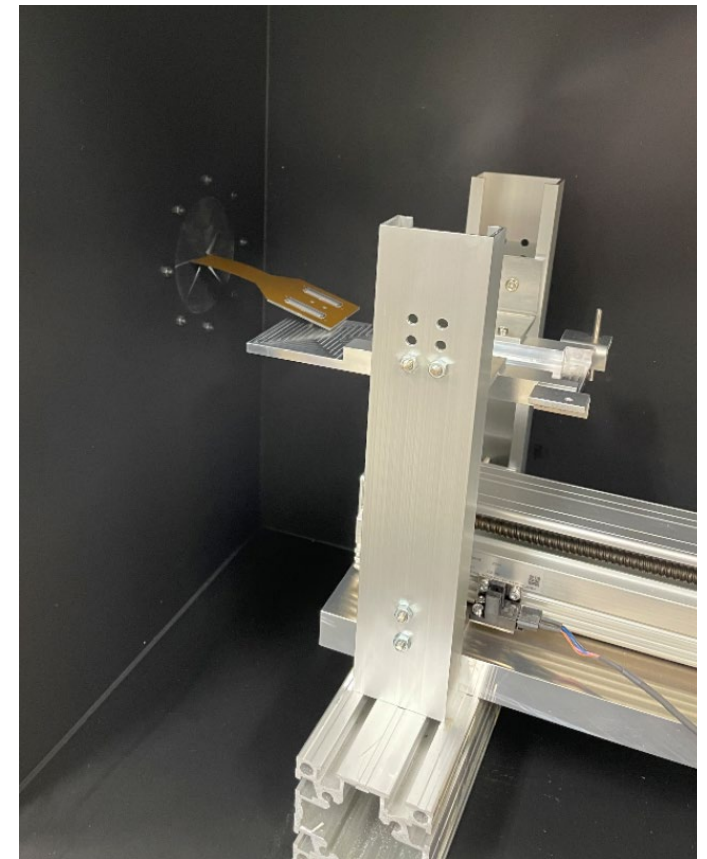
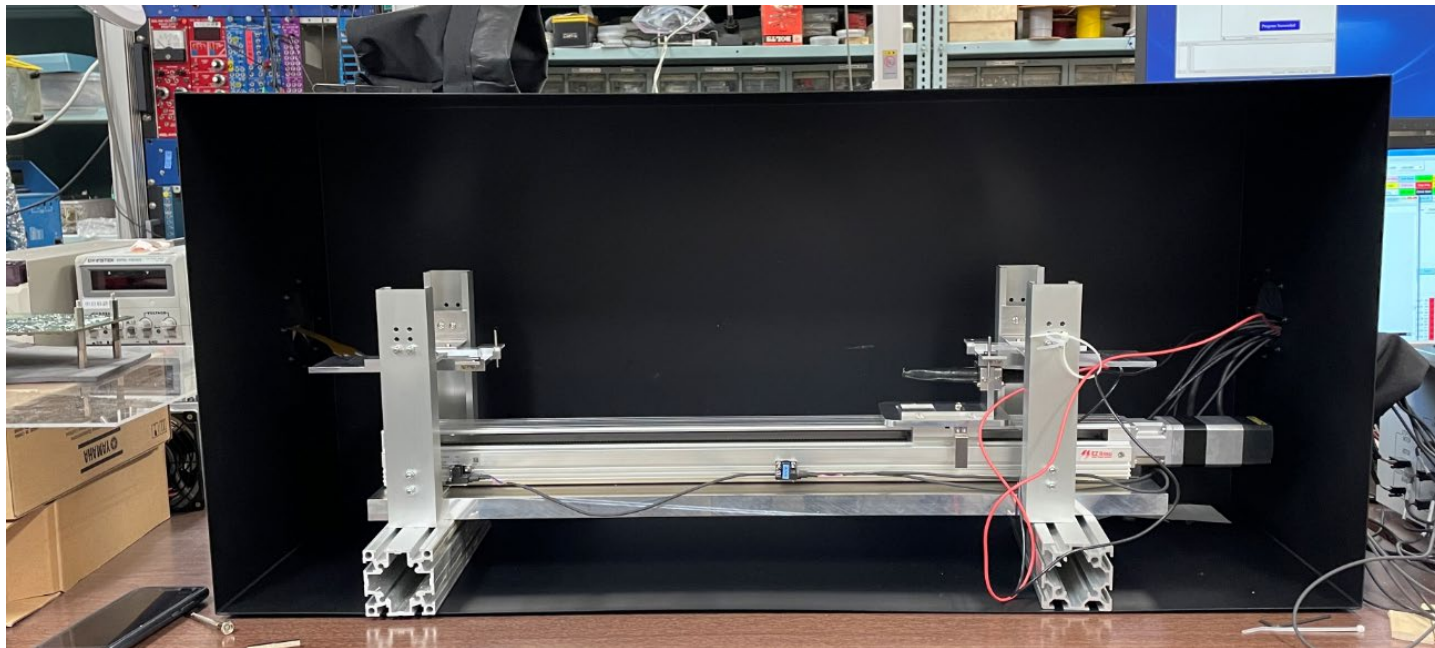


暗箱 横開き



2022/6/22

Readbackerとその他



モーターと電源間の
ケーブルが短い

追加購入

ラチェットバークランプ(ふたのおさえ簡易版)



ゴム足： 暗箱の底を持ち上げる。 ふたを閉めるため



パッチン錠 暗箱をしめるため： 上2つ、左右1つずつ

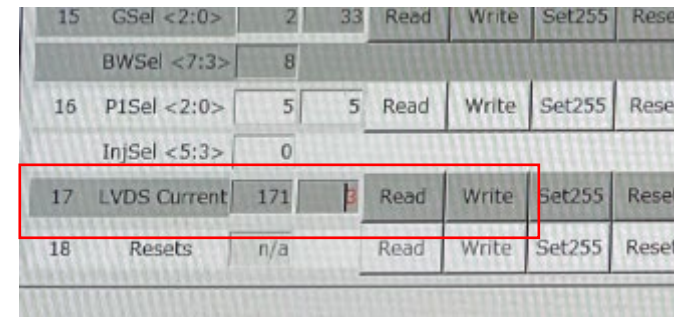


backup

ReadBackerの動作確認2

1. 設定値が読み出せるかテスト

- ① LVDS電流に値を設定しWriteボタンを押す
- ② Readボタンを押し、設定値が読み出せるか確認



2. リセット後のデフォルト値が読み出せるかテスト

- ① FFRボタンを押して、チップをハードリセット
- ② Readボタンを押し、設定値が読み出せるか確認
 - 仕様書の初期値か？ 結果はLVDS電流値は16だった。仕様書と同じ
 - Register 15の初期値が仕様書と違う値だった。
 - Writeボタンの設定値と違う値か。

15	GSel	2:0	2	Gain Select
15	BWSel	7:3	4	Bandwidth Select
16	PISel	2:0	5	

Reg	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
読出値	1	1	8	16	32	48	80	112	144	176	70	4	64	33	5	16
初期値	1	1	8	16	32	48	80	112	144	176	4 6 70	0 4 4	4 0 64	4 2 34	0 5 5	16

Address	Name	Bits	Default	Notes
1	Mask	N/A	N/A	Set Command = Mask Channel Reset Command = Unmask Channel Data bits D6-D0 = Channel Address Data bit D7 = Global Command (i.e. Mask all channels or Unmask all channels)
2	Digital Control	7:0	1	Bit 0 = Active Serial Lines (1=Two, 0=One) Bit 1 = Accept (1=Accept Hits, 0=Reject) Bit 2 = Global Inject Enable Bit 3 = Serial Output Order
3	Vref	1:0	1	
4	Threshold DAC 0	7:0	8	
5	Threshold DAC 1	7:0	16	
6	Threshold DAC 2	7:0	32	
7	Threshold DAC 3	7:0	48	
8	Threshold DAC 4	7:0	80	
9	Threshold DAC 5	7:0	112	
10	Threshold DAC 6	7:0	144	
11	Threshold DAC 7	7:0	176	
12	N1Sel	3:0	6	
12	N2Sel	7:4	4	
13	FB1Sel	3:0	4	
13	LeakSel	7:4	0	
14	P3Sel	1:0	0	
14	P2Sel	7:4	4	
15	GSel	2:0	2	Gain Select
15	BWSel	7:3	4	Bandwidth Select
16	P1Sel	2:0	5	
16	InjSel	5:3	0	
17	LVDS Current	7:0	16	1mA LVDS drive current for each active bit
18	Programmable Resets	N/A	N/A	Reset 1 - Set Command – core Reset Reset 2 - Reset Command – time Reset Reset 3 – Default Command – unused Reset 4 – Write Command - unused

2021/6/22

Readbackerとその他

Readbackerを作るために変更したこと

- FPGA

- FEM-IB :

- USBドライバ(FTDI)のRDを読めるように、FPGAのピンアサインをした(A12)
 - USB-IFモジュールの改造
 - RD=Hiのとき、バスの方向をFPGA->USBにして、ReadBackしてきたデータを流す
 - Receiverモジュールの改造
 - ReadBackしてきたデータをずっと保存。(次のReadBackデータが来るまで保存)
 - テストピンの出力

- FEM-SlowControl

- テストピンの出力

テストPin#	出力
15	ROC_DATA
14	DATA_TO_SERIAL_OUTPUT
13	DATA_OUT_SC_int
12	SC_DATA_IN(9)
11	WE_SC_VME_int

Readbackerとその他

テストPin#	出力
1	DI_SC_VME
2	DATA_TO_USB(0)
3	DATA_TMP(7)
7	READ_USB
8	DATA_STRB(WR)

GUIソフトウェア

- Nevis_gui_for_calib3.py
 - Read_register関数を改造
 - READ (0) インstrukションを使用してレジスタを読み出す
 - 書き込むときはWRITE(1)やSET(2) とすることでレジスタに書き込みつつ、データを読み出す。
 - READ(0) では書き込まず、読み出しデータだけを取り出すことができる。
 - 実行シーケンス
 - Write_bytes_to_target関数でREADコマンドをFPHXチップに送信： 同期BitBangで送信
 - Read_bytes_to_USB関数でFEM_IBに戻ってきたReadBackデータを取り出す： 非同期BitBang
 - Swap_bit：データのビット列を反転する。
 - ファイル中のバイナリデータをUSBに転送するボタンにReadボタンを追加
 - 実験用。
- fphxtb.py
 - Read_bytes_to_USB関数を追加
 - Async_bitbangモードでデータバスの情報をIOする。
 - FPGAからのデータ(Readbackデータ)を取り出す
 - 実装はWrite_byte_to_USBのほぼコピー
 - swap_bit 関数の追加
 - 読み出してきた8bitデータのビット位置を反転する。

2021/6/22 01234567 -> 76543210 : FPHXからのデータは0ビット目が先頭になっているので、7ビット目が先頭になるように入れ替える

Algorithm of phase follower

- Data is latched with 4 different phases of the clock(200MHz).
 - 0, 90, 180, 270 degree shifted respectively
- If data is found at CLK first, data at **CLK180** is received.
- Algorithm is explained in
 - <https://www.phenix.bnl.gov/WWW/publish/brooks/silicon/reviews/Nov08/FVTX%20Readout%20Electronics%20Details.pdf>

