

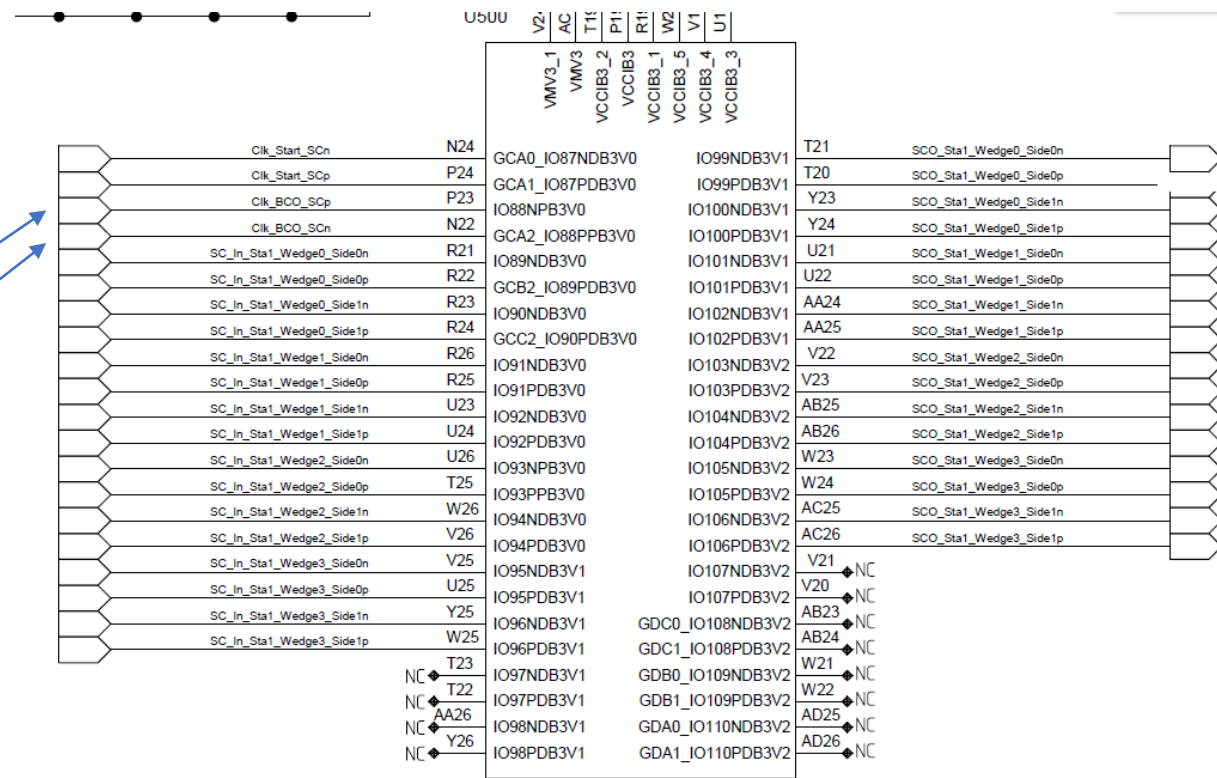
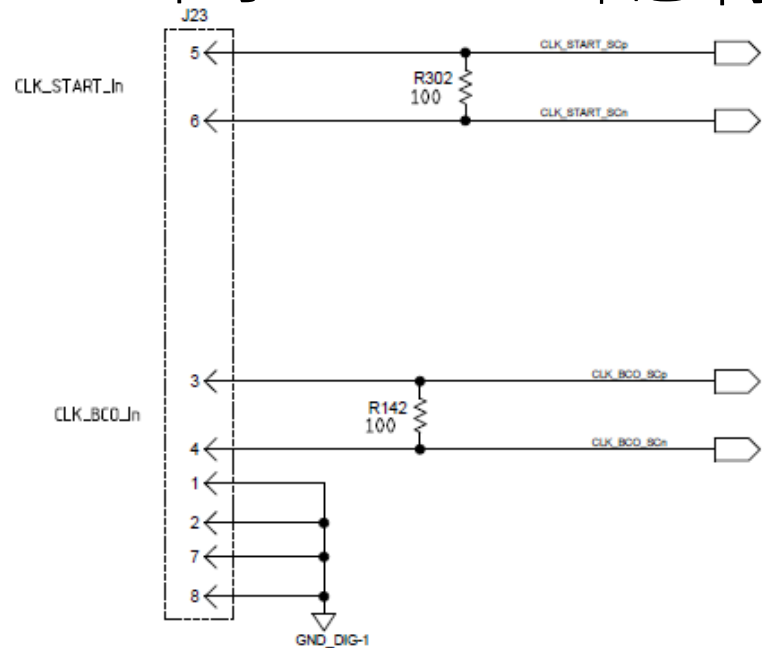
# Readbacker memo

## Debug status of Readbacker w/ BEX

T. Hachiya

Nara Women's University & RIKEN BNL

# ROC内のBCO配線



SlowCtrlFPGA

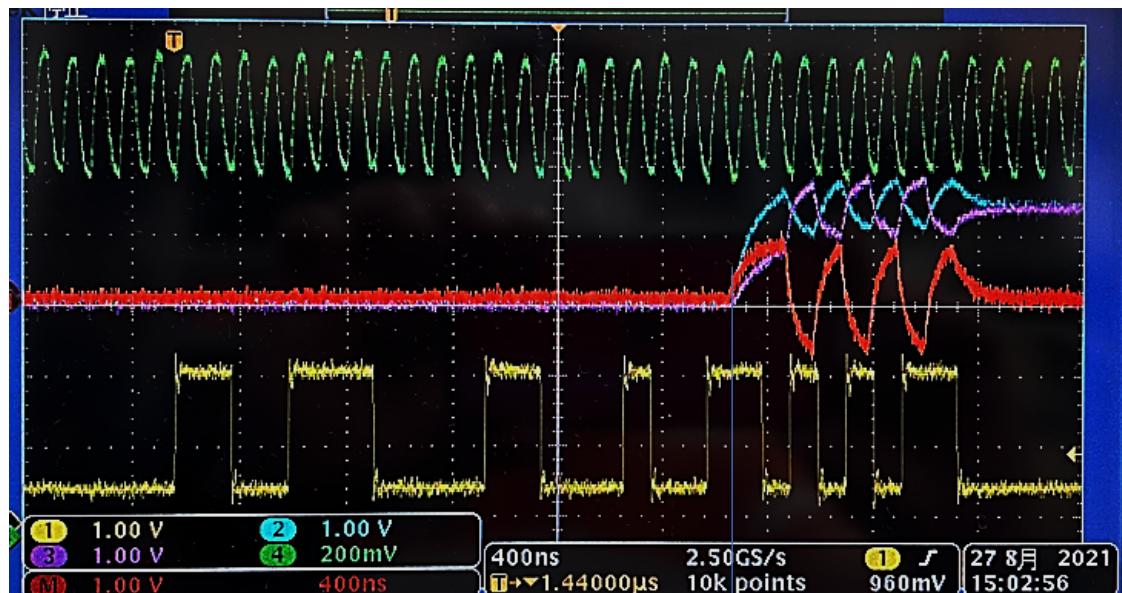
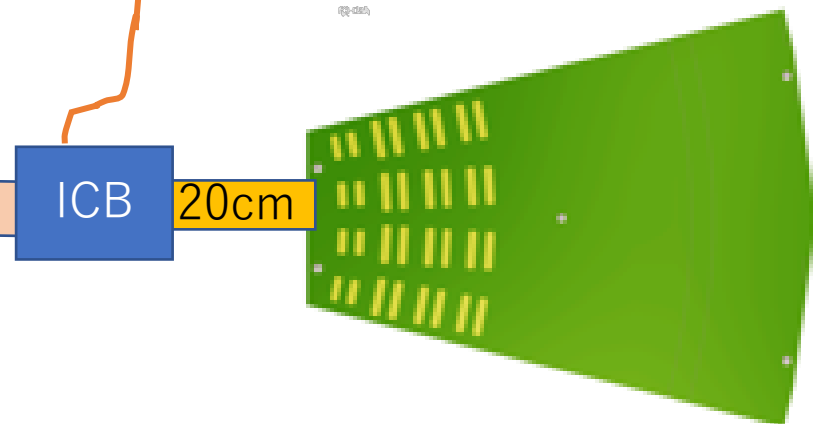
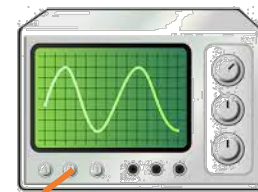
- 回路図
  - $BCO_p(n) = P23 (N22)$
- FPGA
  - `set_io BCO_CLK_0_n -pinname P23 -fixed yes`
  - `set_io BCO_CLK_0_p -pinname N22 -fixed yes`
- 反転している。

# Status

- BEXをつながない時(変換基板のみ)は正常に動作する
  - 設定値をそのまま読み出せる
- BEX + 変換基板をつなぐと、読み出し結果が設定値と異なる。
- 状況をもう少し調べた

Reg	Desc	To Chip	From Chip	Reg
*	Wild	0		Reg
1	Mask	0		Reg
2	Dig Ctrl	5	15	Reg
3	Vref	1		Reg
4	DAC0	8	16	Reg
5	DAC1	16	32	Reg
6	DAC2	30	60	Reg
7	DAC3	35	71	Reg
8	DAC4	40	80	Reg
9	DAC5	45	91	Reg
10	DAC6	50	100	Reg
11	DAC7	55	111	Reg
12	N1Sel <3:0>	6		Reg
	N2Sel <7:4>	4	3	

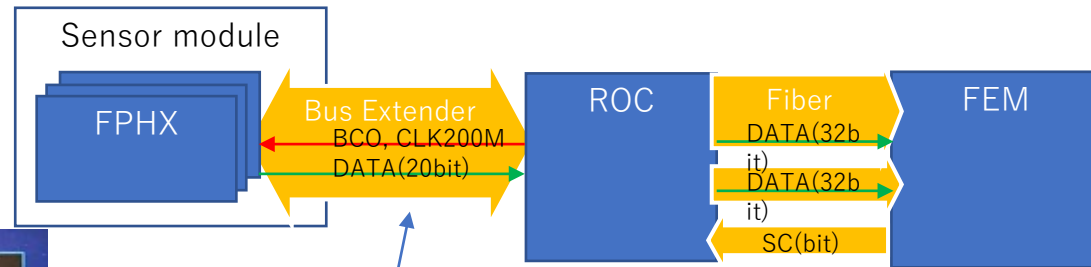
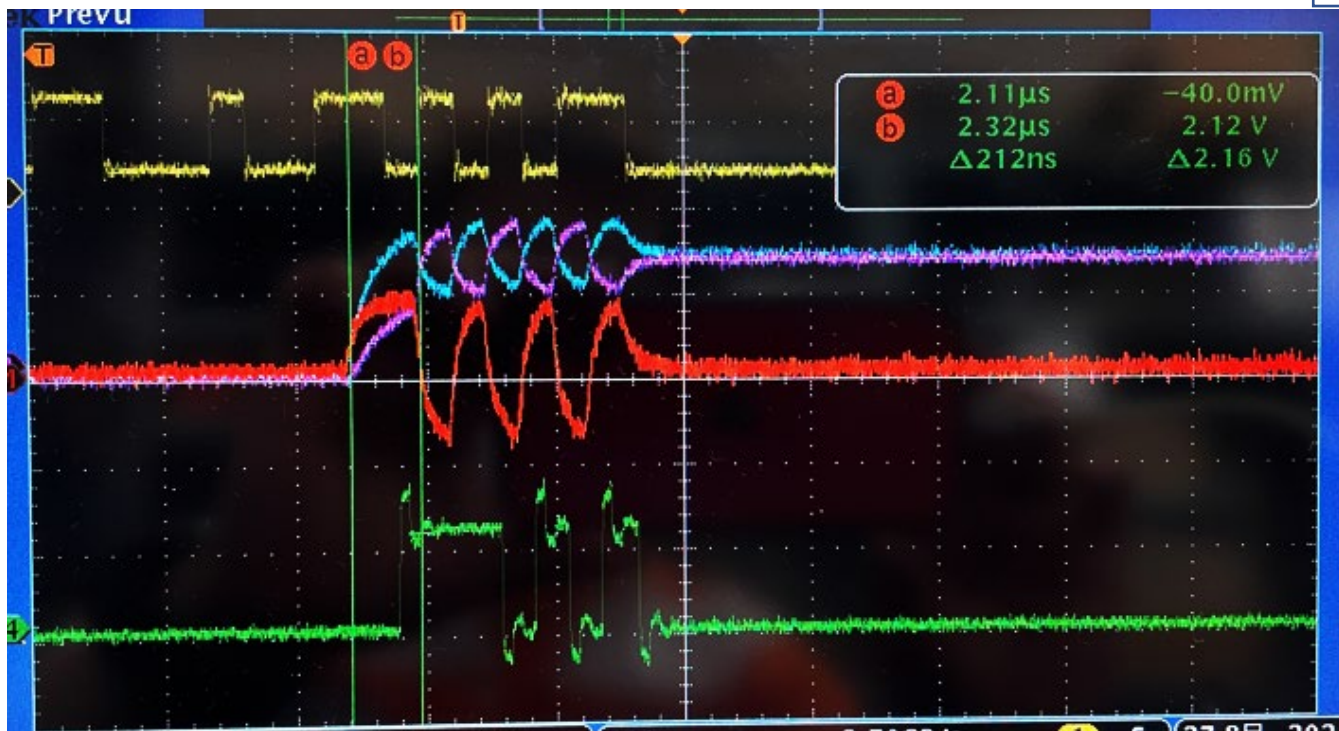
# 測定のセットアップ



BCO(10MHz)  
SC\_OUTp  
SC\_OUTn  
Diff(SC\_OUT)  
SC\_IN

- 入力：10101011
- 出力：11010101
  - 反転してでてくる。
- 入力と出力が一致している。

# 読み出しデータ



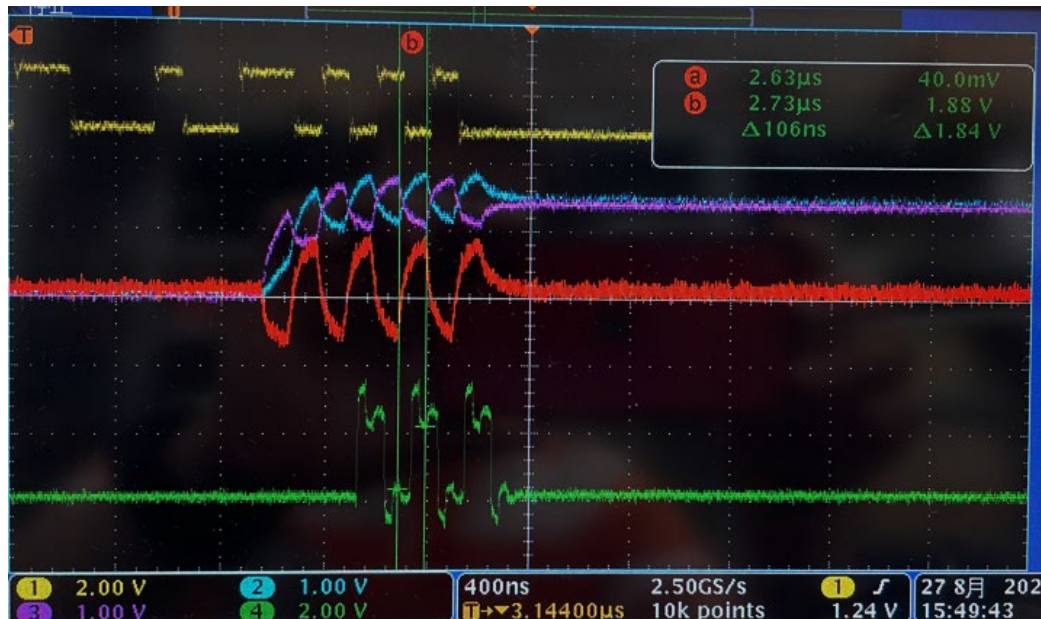
SC\_IN  
 SC\_OUTp  
 SC\_OUTn  
 Diff(SC\_OUT)

SC\_OUT@FEM FEMで受信したデータをTPで出力

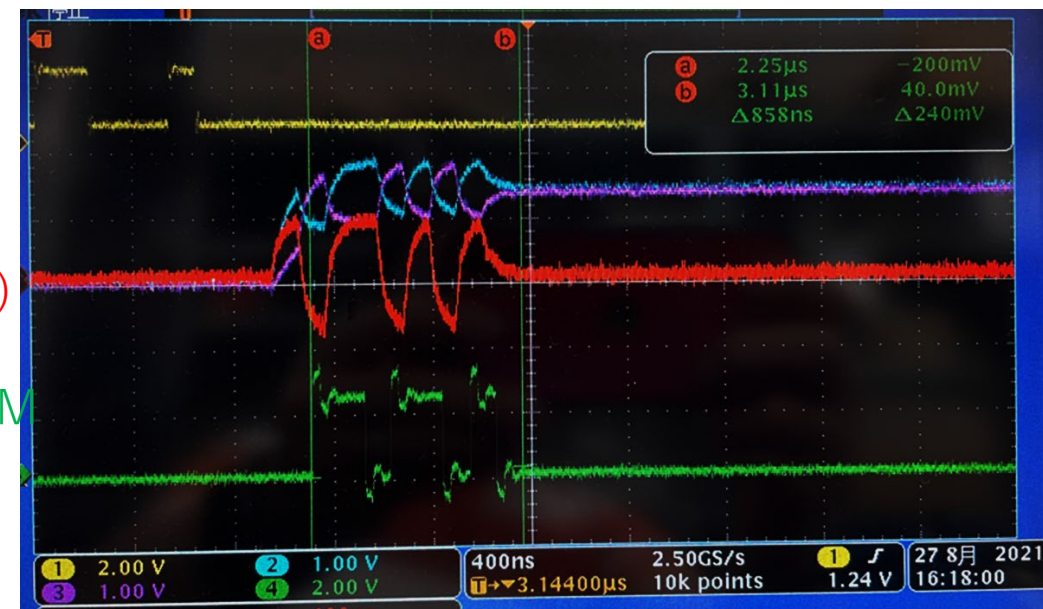
ICB上では、SC\_IN = SC\_OUT  
 FEM前後で、SC\_OUTが変化  
 → ROCでデータが化けている

- SC\_IN: 10101011
- SC\_OUT: 11010101 (SC\_INに一致)
- SC\_OUT@FEM : 11101010 (違う。1ビットずれ+最初のビットが違う)
- GUI:87 (01010111)

# 読み出しデータ



SC\_IN  
SC\_OUTp  
SC\_OUTn  
Diff(SC\_OUT)  
SC\_OUT@FEM



- SC\_IN: 10101010
- SC\_OUT: 01010101 (SC\_INに一致)
- SC\_OUT@FEM : 00101010
  - (違う。1ビットずれ+最初のビットが違う)
- GUI:84 (00101010)

- SC\_IN: 173
- SC\_OUT: 10110101 (SC\_INに一致)
- SC\_OUT@FEM : 11011010
  - (違う。1ビットずれ+最初のビットが違う)
- GUI:91 (00101010)

読み出しデータの1ビット目は、2ビット目といつも同じ

# データまとめ

GUI INPUT	SC_OUT(ICB)	SC_OUT(FEM)	GUI ReadBack	
171 (10101011)	11010101	11101010 (87)	87 (01010111)	
170 (10101010)	01010101	00101010 (84)	84 (01010100)	
172 (10101100)	00110101	00011010 (88)	0	読み出し結果がおかしい
173 (10101101)	10110101	11011010 (91)	91	
174 (10101110)	01110101	00111010 (92)	92	
138 (10001000)	01010001	00101000 (20)	20 (00010100)	
122 (01111010)	01011110	00101111 (244)	244 (11110100)	
20 (00010100)	00101000	00010100 (40)	40 (00101000)	
40 (00101000)	00010100	00001010 (80)	80	
25 (00011001)	10011000	11001100 (51)	51	
30 (00011110)	01111000	00111100 (60)	60	
35 (00100011)	11000100	11100010 (71)	71	

FEM前後で、SC\_OUTが変化 → ROCでデータが化けている。

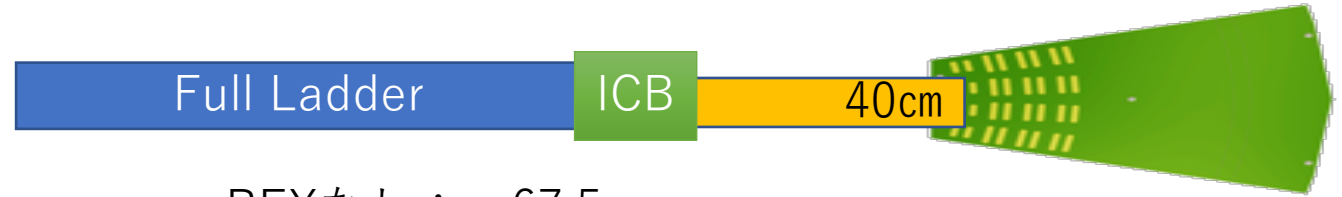
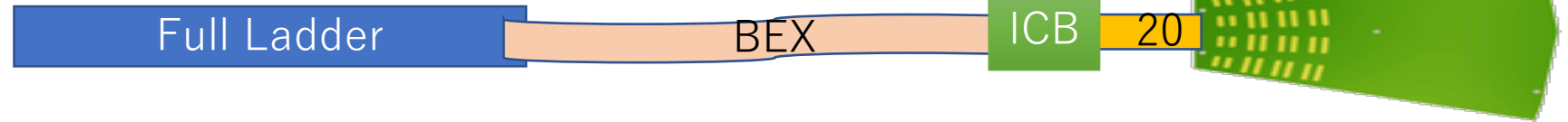
- ・ 1ビットずれ (後ろに)
- ・ 読み出しデータの1ビット目は、2ビット目と同じ

# ありえる原因

- ROC上で、FPHXデータの受信失敗
  - (距離が長い?ため、)データが遅れ、受信タイミングがずれてしまう
    - 典型的には1クロックおくれる
- 観測結果からこれが推測される
  - バスエクステンダを使ったときに起きる。
    - より下流(FEMなど) が原因とは考えにくい。
  - Readback波形の遅れ+なまりが原因と推測

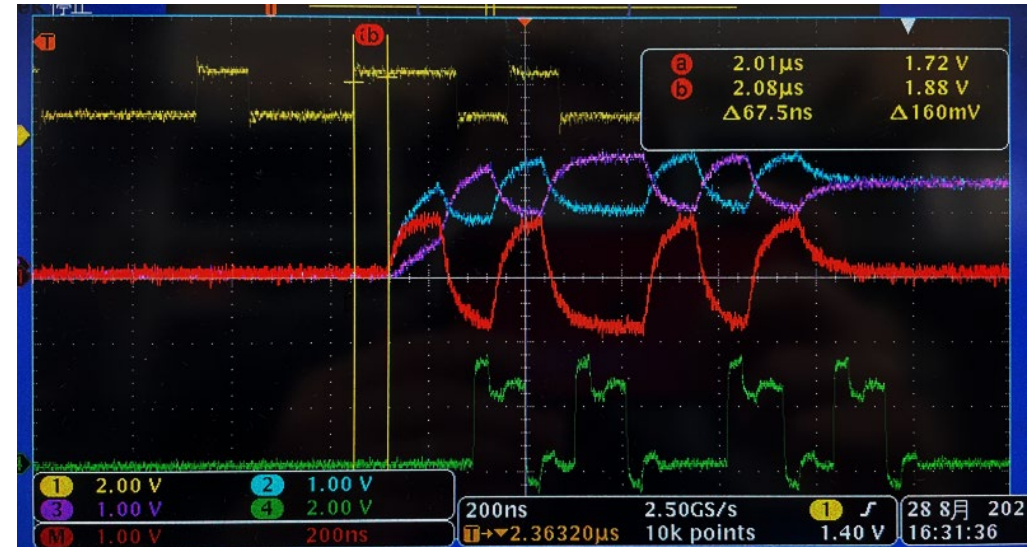
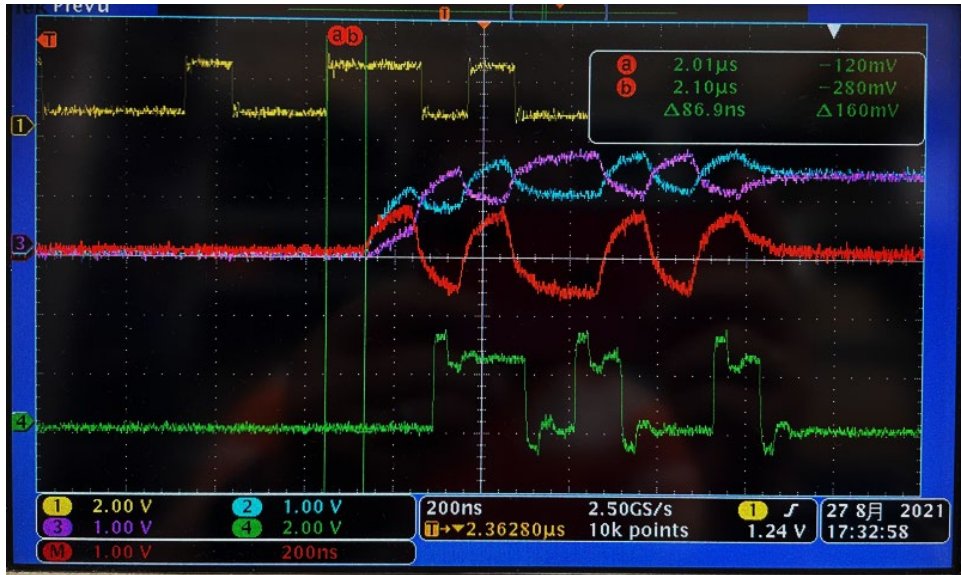


# Readback信号の戻り時間



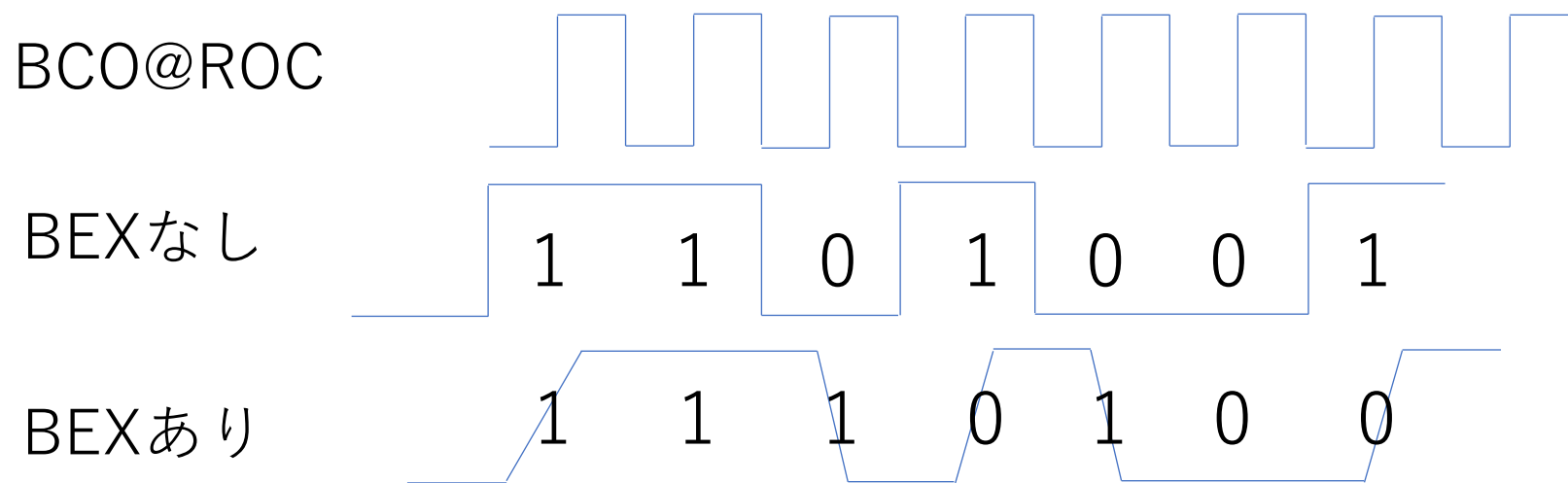
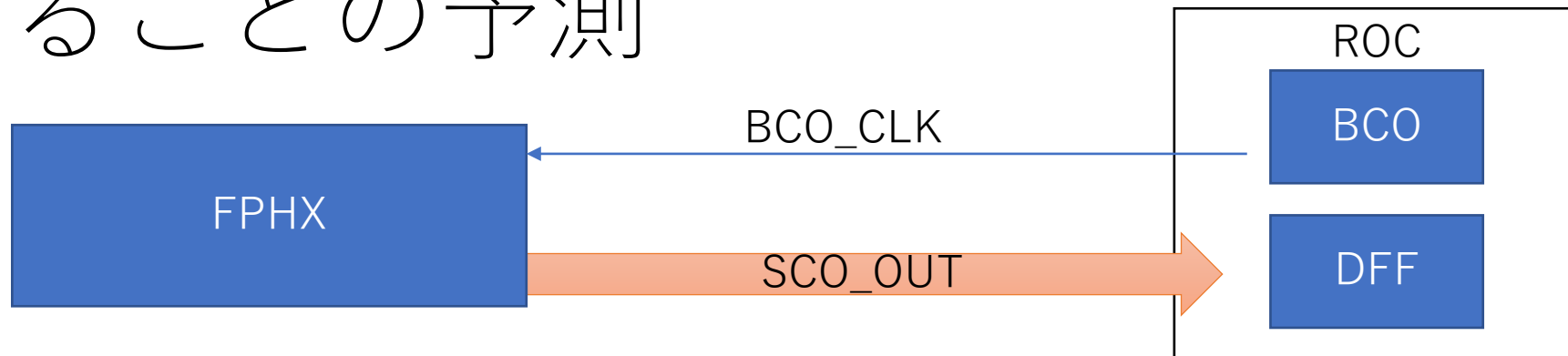
BEXあり : 86.9ns

BEXなし : 67.5ns



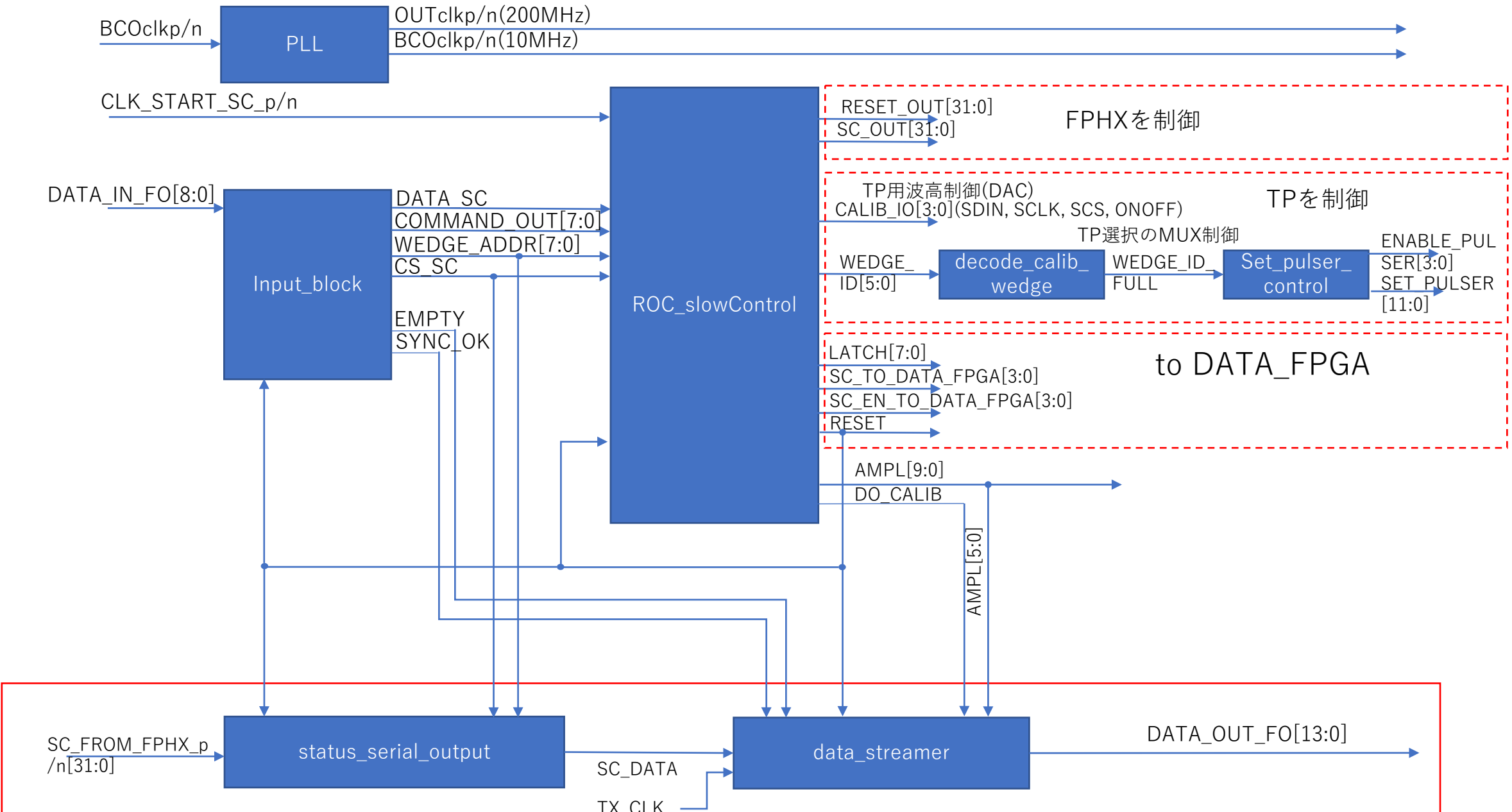
- 時間差 :  $86.9 - 67.5 = 19.4\text{ns}$  @ 130cm BEX
  - 130cm の往復でかかる時間 : 13ns (20cm/1nsで換算)
  - BCOの鈍りで遅れていることも考えられる。

# 起こっていることの予測



- 20ns程度遅れる
- 最初のスロープが緩やか (Z -> ロジック に切り替わる)

# ROC\_Slow\_control\_top



# Status\_serial\_output

```
elsif falling_edge(BCO_CLK) then

--Create an FPHX send_command trigger, and upon this trigger start a counter which will
--tell us when to extract the status word from the FPHX SC line:
SEND_COMMAND_BUF <= SEND_COMMAND;
SEND_COMMAND_2BUF <= SEND_COMMAND_BUF;
SEND_COMMAND_TRIG <= SEND_COMMAND_BUF and (not SEND_COMMAND_2BUF);

if SEND_COMMAND_TRIG = '1' then
    CE <= '1';
end if;

--After 24 clocks, read the 8-bit FPHX status word:
if ADDR = "010010" then
    READ_DATA_FPHX <= '1';
end if;
if ADDR = "011010" then
    READ_DATA_FPHX <= '0';
    CE <= '0';
end if;

if (READ_DATA_FPHX = '1') then
    STATUS_DATA <= SC_FROM_FPHX_int;
    CS_DATA <= '1';
else
    STATUS_DATA <= '0';
    CS_DATA <= '0';
end if;
```

SC送信後、Wait時間が固定(24clock)。

解決案： これを1つずらせばよい。

高度な解決案。

- Wait時間をコマンドで可変にする。(初期値は設定する)
- PhaseShifterを導入する。(高速データと同じ)

# Data\_streamerへのインプット(16ビット)

DATA_TO_SC_FO	
(15 downto 13)	(others => '0')
12	LATCH_int(3); --BCO_CLK;
11	SC_EN_TO_DATA_FPGA_int(1);
10	FPHX_CS_DATA;
9	FPHX_STATUS_DATA;
(8 downto 3)	AMPLITUDE(5 downto 0);
2	CALIB_IO_int(1);
1	not(SC_FIFO_EMPTY_int);
0	CALIB_BUSY;

Readback bit[12-0]	
12	
11	CLK_START_SC
10	CS_FROM_ROC;
9	DATA_FROM_ROC
8 - 3	AMPLITUDE(5 downto 0);
2	CALIB_TRIGGER
1	BUSY_ROC_SC
0	BUSY_CALIB

2の下に以下のコメント (しかし、CS\_SCは送らないように変更されている)

```
--Send the CS_SC back up to the FEM, to be used as a BUSY so that it will not send another  
--command until the last one is completed. Needed because ROC SC communication runs on a  
--slow clock than SC from FEM
```

# 対応するFEM側のSlowControl Readback data

- 13 bit words, each bit has different meanings

Readback bit[12-0]		
12		Not used in FEM
11	CLK_START_SC	Write Enable to DATA FPGA on ROC
10	CS_FROM_ROC;	
9	DATA_FROM_ROC	FPHX readback data (8 serial bit/word. Need 8 clocks to receive)
8 - 3	AMPLITUDE(5 downto 0);	Test pulse amplitude
2	CALIB_TRIGGER	
1	BUSY_ROC_SC	
0	BUSY_CALIB	--BUSY for calibration