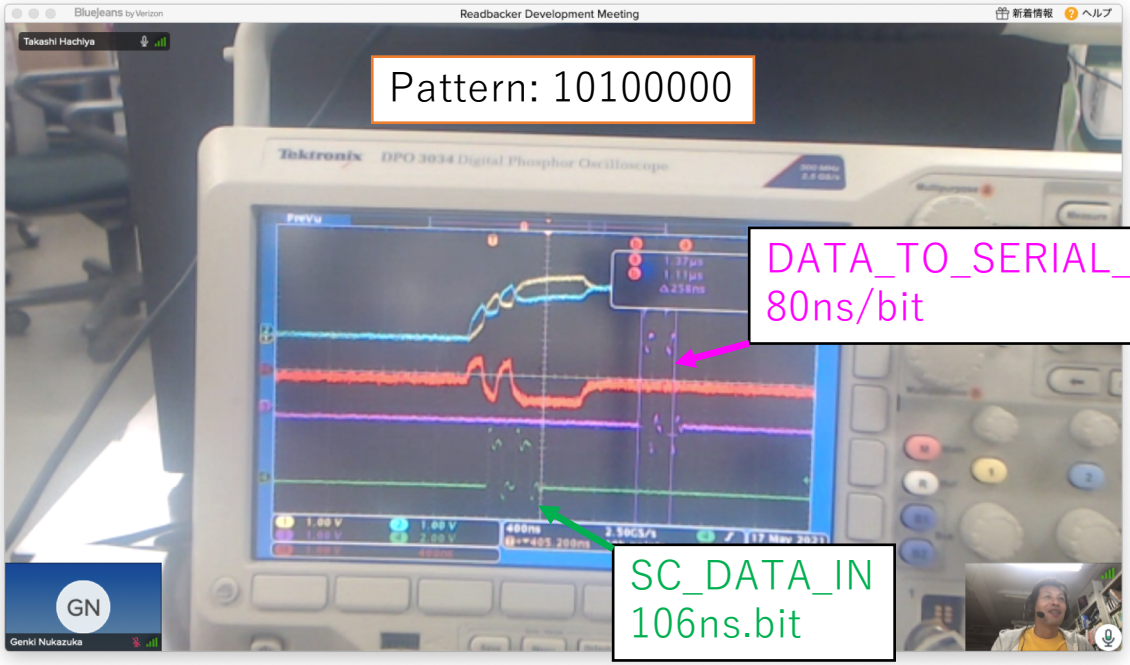
2021/5/17 Debugging

Takashi Hachiya, Maya Shimomura (NWU)
Genki Nukazuka, Itaru Nakagawa (RIKEN)

Checking Upstream Variables than ROC_DATA

@FEM Slow Control FPGA



CH1: SC_OUT_0N
CH2: SC_OUT_0P
M: CH 1 -CH 2
CH3: DATA_TO_SERIAL_OUTPUT
CH4: SC_DATA_IN

Most upstream variable at SC_FPGA@FEM

Seem to be all consistent between SC_OUT@FPHX = SC_DATA_IN = DATA_TO_SERIAL_OUTPUT

Various Pattern Attempts

Pattern	pattern	SC_DATA_IN	DATA_TO_SERIAL_OUTPUT	Consistency Level
	10100000	10100000	10100000	2
	01000000	01000000	01000000	2
	01100000	01100000	01100000	2
	11000000	11000000	11000000	2
	10110000	10110000	10110000	2
	11010000	11010000	11010000	2
	10101000	10101000	10101000	2
	10111000	10111000	10111000	2
	10010100	10010100	10010100	2
	11001100	11001100	11001100	2
	10101011	10101011	10101011	2
128	10000000	00000000	00000000	1
129	10000001	10000000	10000000	1
130	10000010	10000000	10000000	1
131	10000011	10000011	10000011	2

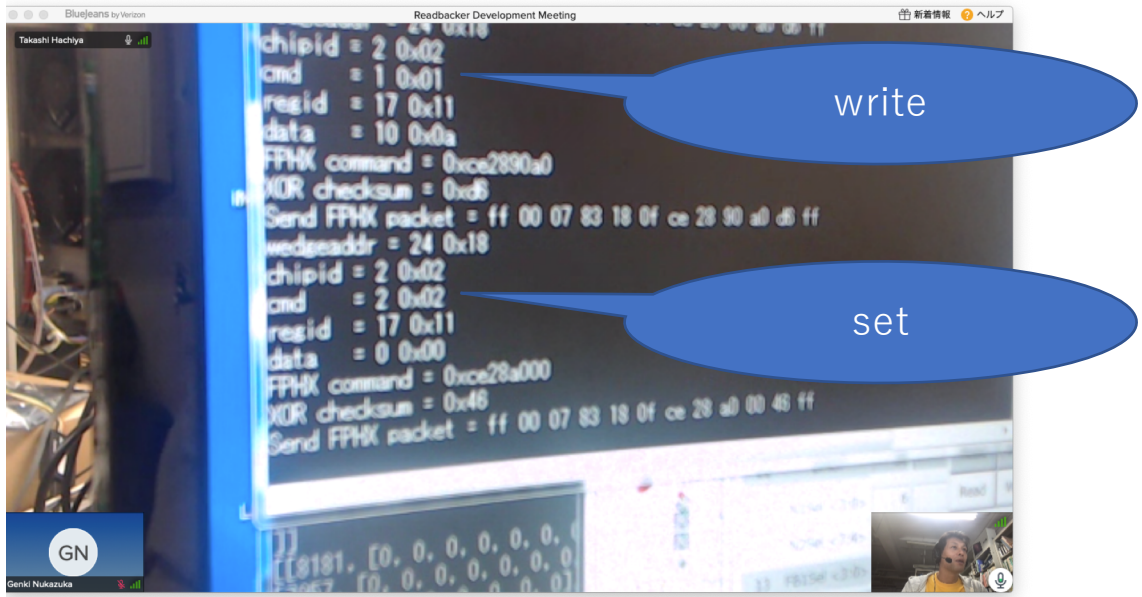
2: Perfect
1: Partially
0: Bad at all

Chip SC_OUT is as expected, but missing bits in FEM SC_FPGA. Chip-2, Chip-3 both behaves the same.

Slow Control Command Sequence

- Tab-8 -> Chip-2 -> Side-1
- Type in LVDS Current
- Press “**Write**” -> “**Write**”

This procedure is different from December Press “**Set**” -> “**Write**”

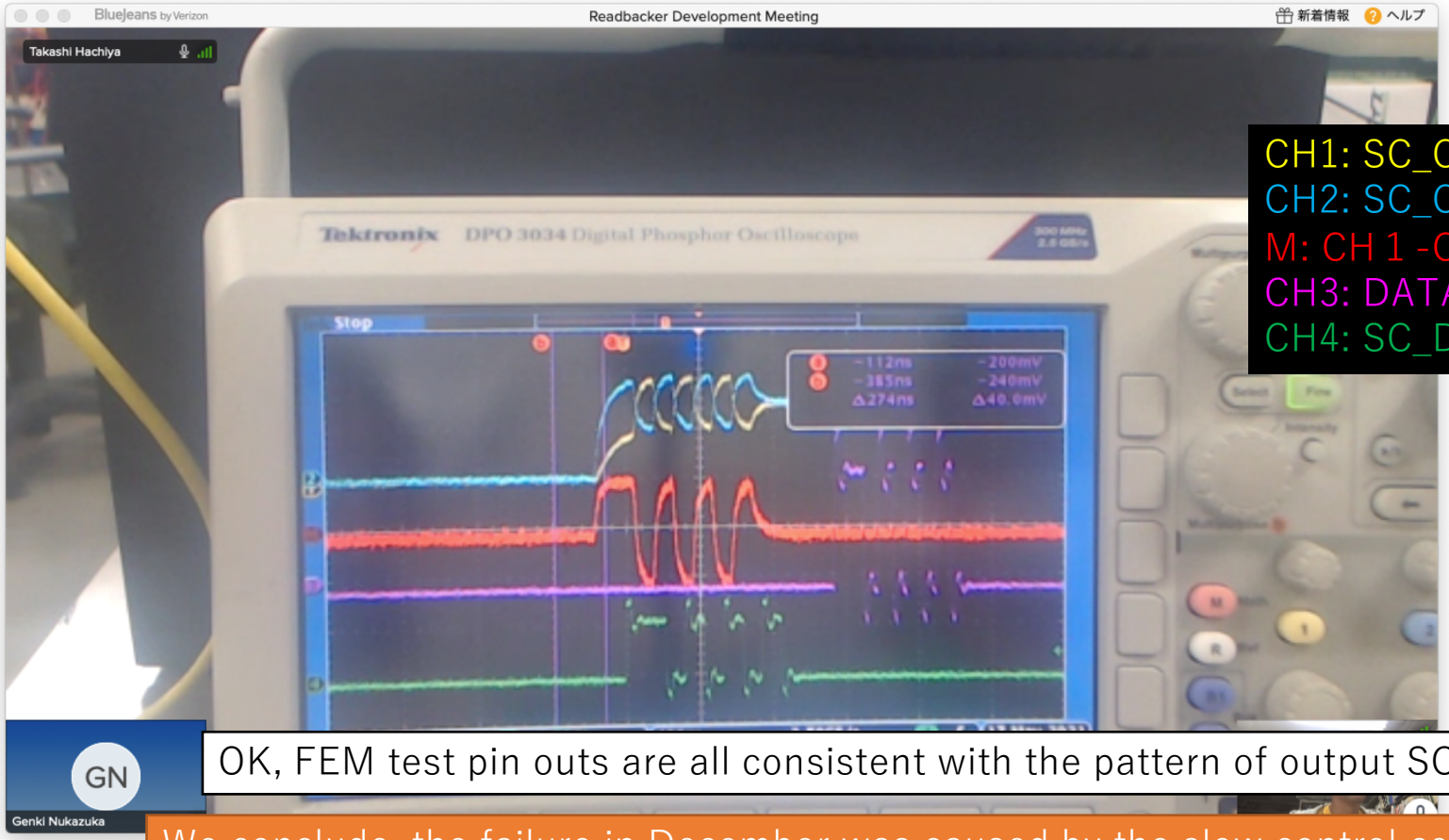


The sequence in December was inadequate.
According to FPHX manual,
“**Set**” commands resets all register bits to be 1

Table 1 - Slow Controller Instruction Set

I_2	I_1	I_0	Instruction	Effect
0	0	0	No effect	
0	0	1	**Write**	The eight data bits (D_7-D_0) are written into the register
0	1	0	**Set**	All the register bits are set to 1 [D_7-D_0 are ignored]
0	1	1	No effect	
1	0	0	No effect	
1	0	1	**Reset**	All the register bits are reset to 0 [D_7-D_0 are ignored]
1	1	0	**Default**	The register bits are sent to a preset, hardwired value [D_7-D_0 are ignored]
1	1	1	No effect	

Attempt Failed Pattern in December



CH1: SC_OUT_0N
CH2: SC_OUT_0P
M: CH 1 -CH 2
CH3: DATA_TO_SERIAL_OUTPUT
CH4: SC_DATA_IN

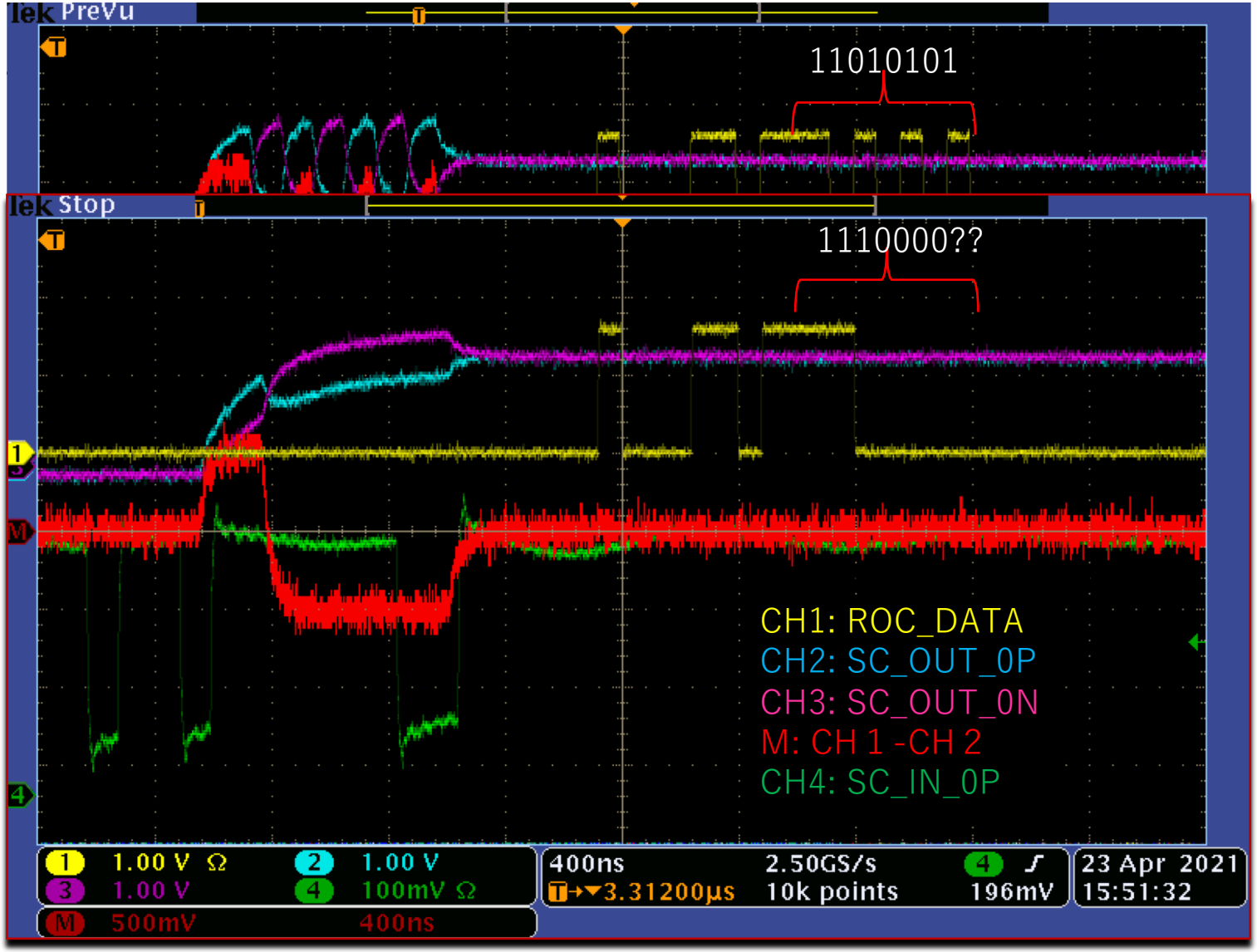
LVDS = 171mA
Pattern : 10101011

OK, FEM test pin outs are all consistent with the pattern of output SC @ FPHX

We conclude, the failure in December was caused by the slow control command execution procedure procedure as pointed out in the previous page.

DATA_OU

April Record



LVDS=171mA (10101011)

Same pattern as previous page, but failed in April attempt. We concluded this is caused by the inadequate SC command sequence.

Correct SC Command Sequence

STEP1

- Select Tab (8)
- Select Chip (2)
- Select Side (1)

Also, specify FEM address on GUI and it has to be matched with the assigned FEM address by the rotary switch!!

STEP2

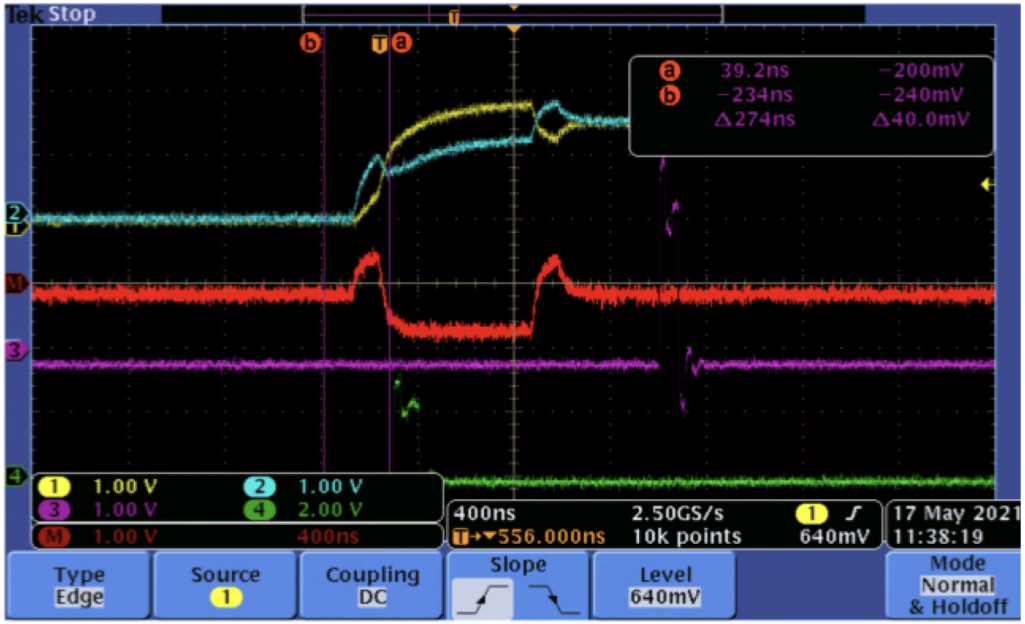
- Type in LVDS Current

STEP3

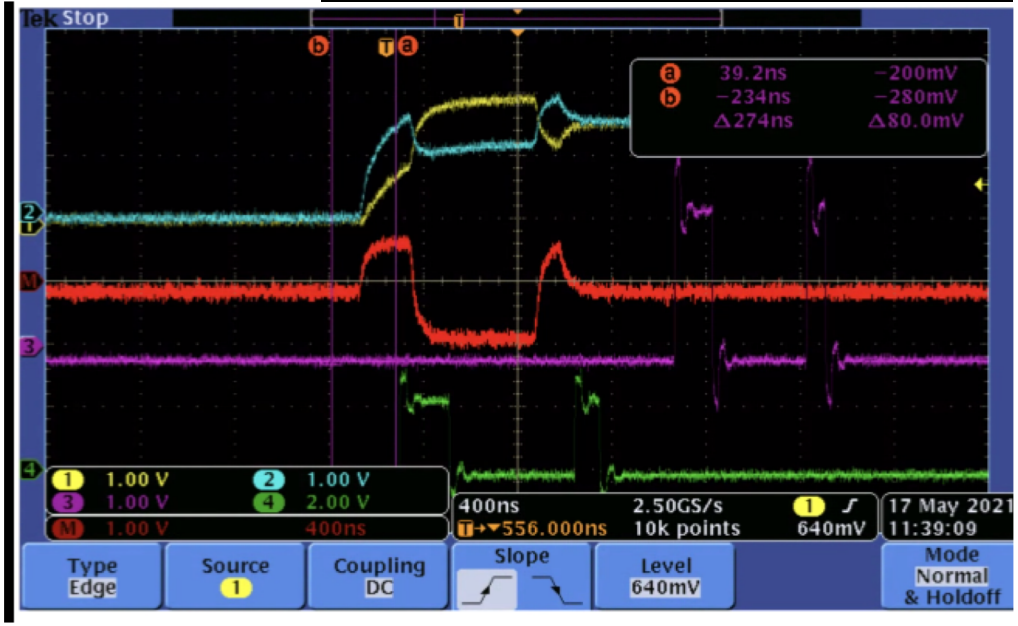
- Press “Write” + “Write”
- Need to press “Write” twice to readback updated registered value (otherwise, readback value is the left over.)

Patterns around 128

CH1: SC_OUT_0N
CH2: SC_OUT_0P
M: CH 1 -CH 2
CH3: DATA_TO_SERIAL_OUTPUT
CH4: SC_DATA_IN



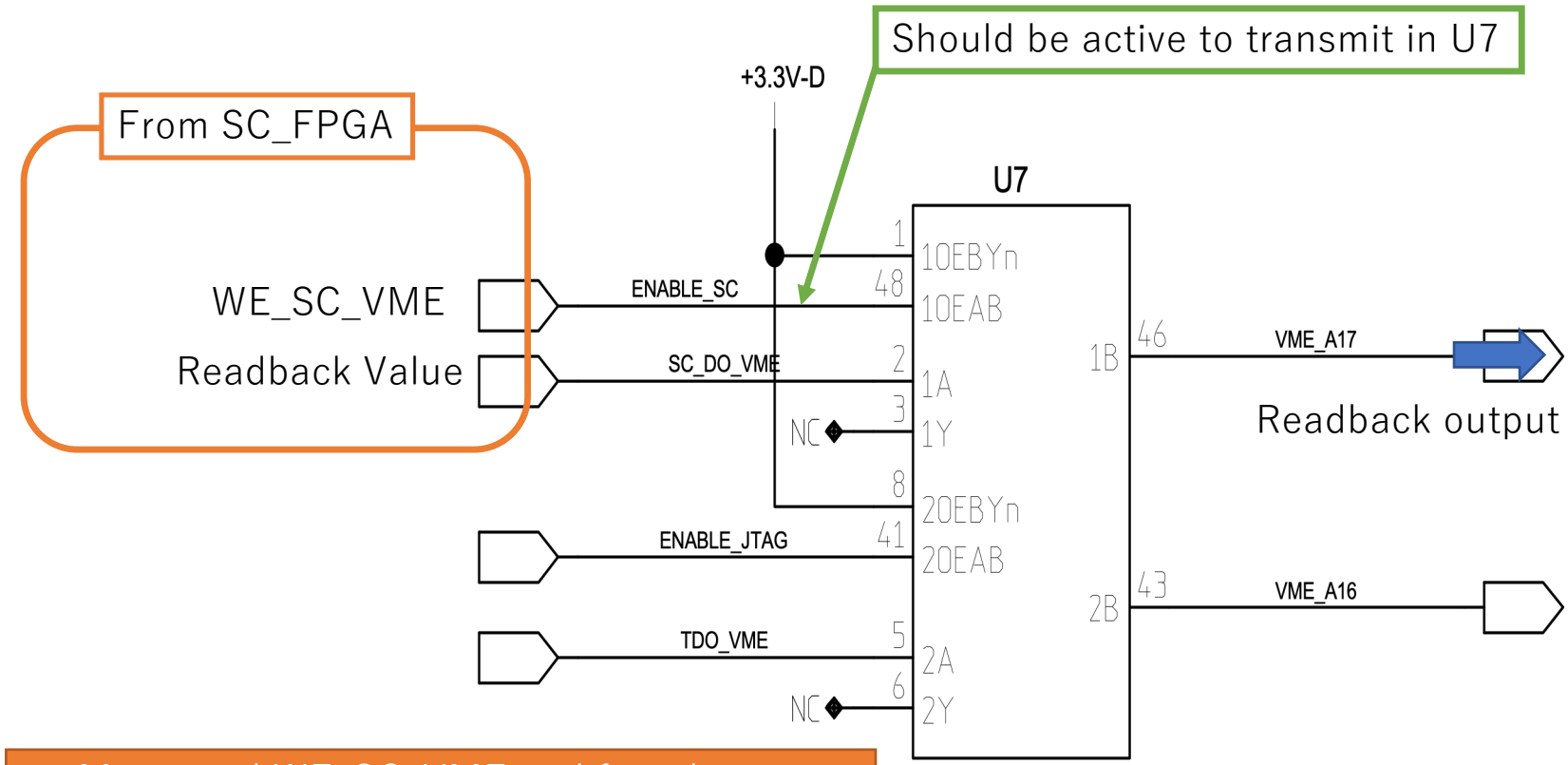
10000001 (129) is failed

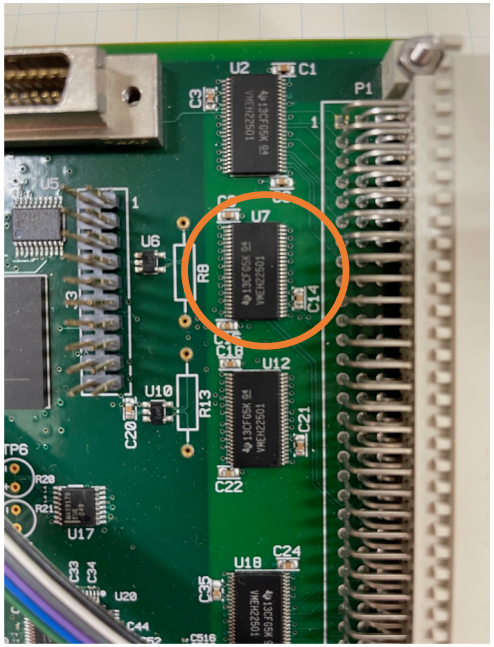


10000011 (131) is OK

Remains mystery. However, we decided to move on for now. Since the most of cases seem to be successful. We'll come back for debugging later.

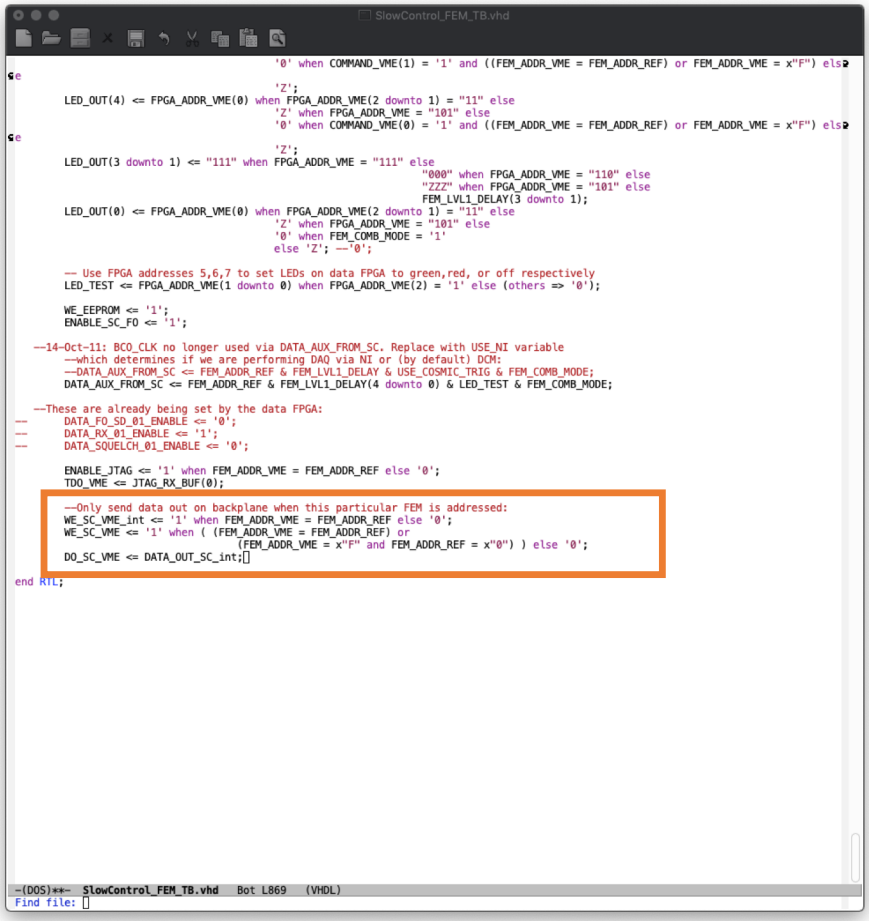
VME-BUS transmission of readback values





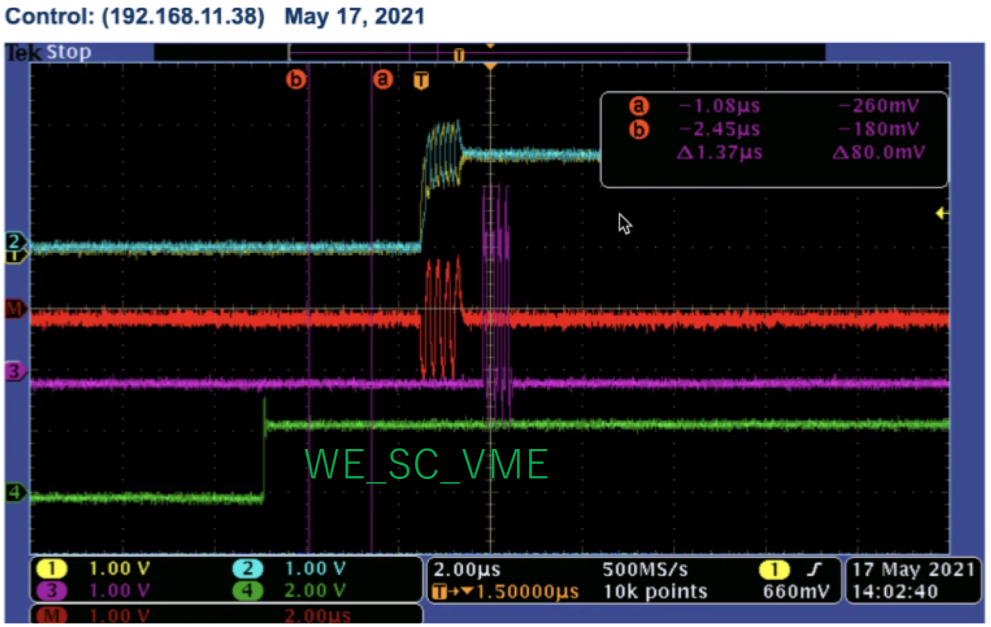
-> Monitored WE_SC_VME and found inactive

WE_SC_VME @ SlowControl_FEM.vhd



- WE_SC_VME becomes active only when FEM address is specified.
- We have to specify the FEM address to make WE_SC_VME active.
- See next page.

Specifying FEM Address



1. Set Rotary Switch to 6 on FEM board
2. Set FEM #6 in Nevis GUI

OK, WE_SC_VEM becomes active.

Now READBACK words suppose to be transmitted from VEM to

FEM Debugging Conclusion

- The Register values monitored at FEM SC FPGA seem to be consistent (most of the cases) with the signal pattern monitored by the interception board, which was confirmed to be consistent with input values at GUI.
- The reason why inconsistency observed in April was due to inadequate SC command sequence. It has to be “**Write**” + “**Write**”, not “**Set**” + “**Write**”.
- Need to specify FEM address to activate VME_BUS transmission of the readback values.
- The readback value at FEM SC FPGA becomes inconsistent for register values 128, 129, 130 with the signal pattern in FPHX output. Somehow the pattern bit is lost during ROC->FEM transmission.
- There can be more patterns which fail if we scan through all 8 bit patterns.
- We decided not to debug this further for now. We’ll move on to FEM-IB check and will come back for debugging later.

FEM-IB

FEM-IB I/O Structure

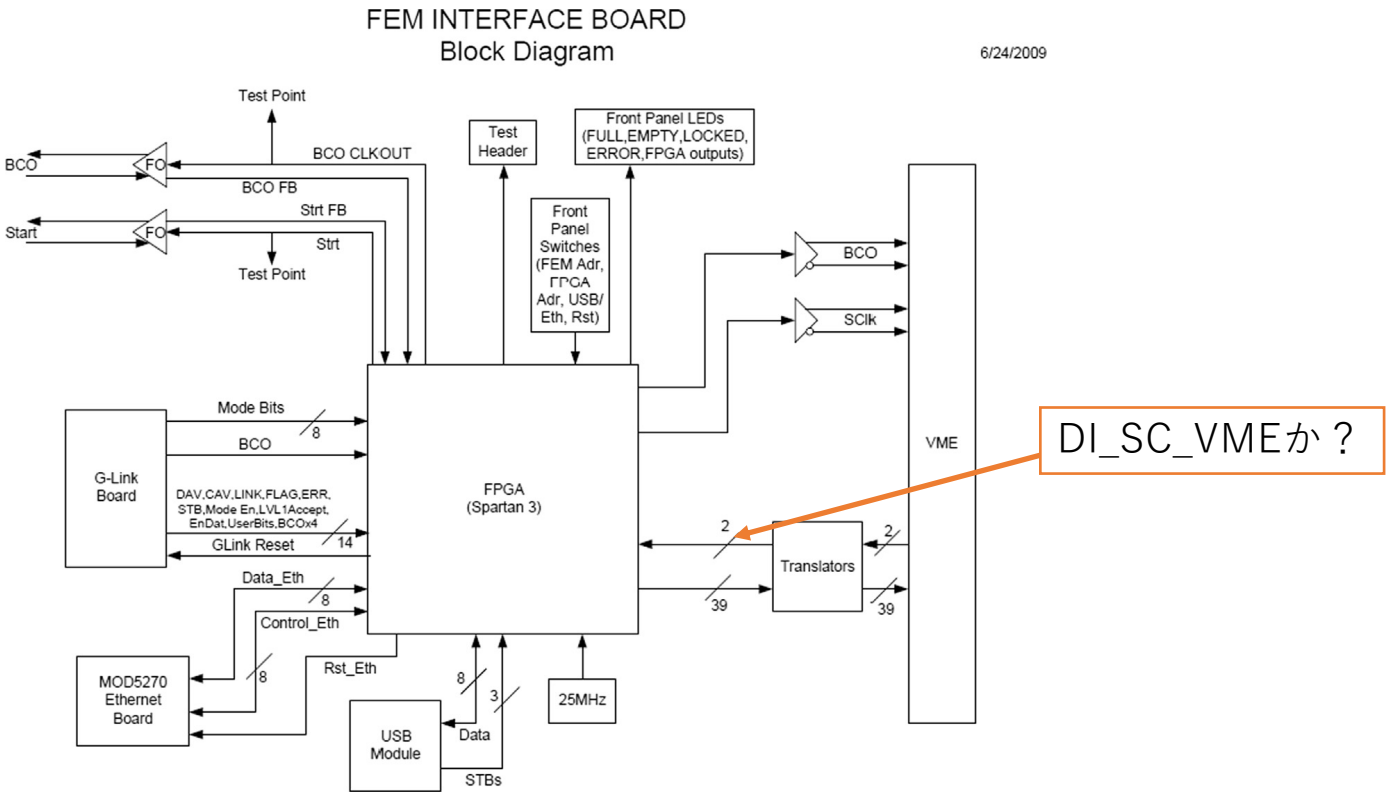


Fig. 13. Block diagram of the FEM Interface Board.

FEM-IB_top.vhd (1008版)

FEM_IB_top.vhd

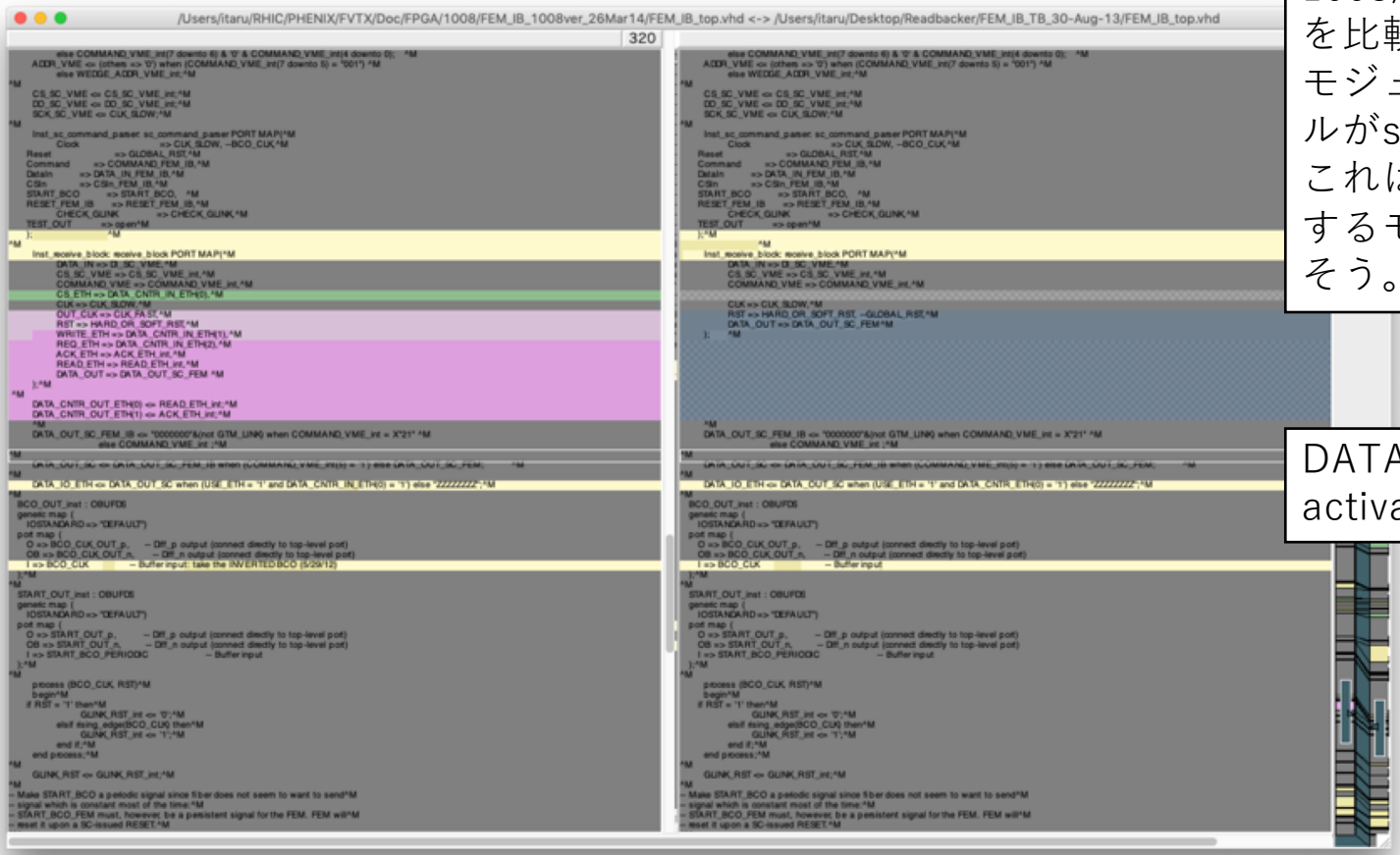
```
^M
      COMMAND_FEM_IB <= "000"&COMMAND_VME_int(4 downto 0) when(COMMAND_VME_int(5) = '1') else (others => '0');^M
      CSIn_FEM_IB <= CS_SC_VME_int;^M
      DATA_IN_FEM_IB <= DO_SC_VME_int;^M
^M
      COMMAND_VME <= (others => '0') when (COMMAND_VME_int(7 downto 5) = "001") ^M
            else COMMAND_VME_int(7 downto 6) & '0' & COMMAND_VME_int(4 downto 0);   ^M
      ADDR_VME <= (others => '0') when (COMMAND_VME_int(7 downto 5) = "001") ^M
            else WEDGE_ADDR_VME_int;^M
^M
      CS_SC_VME <= CS_SC_VME_int;^M
      DO_SC_VME <= DO_SC_VME_int;^M
      SCK_SC_VME <= CLK_SLOW;^M
^M
      Inst_sc_command_parser: sc_command_parser PORT MAP(^M
            Clock                         => CLK_SLOW, --BCO_CLK,^M
      Reset                   => GLOBAL_RST,^M
      Command           => COMMAND_FEM_IB,^M
      DataIn            => DATA_IN_FEM_IB,^M
      CSIn              => CSIn_FEM_IB,^M
      START_BCO               => START_BCO,   ^M
      RESET_FEM_IB      => RESET_FEM_IB,^M
            CHECK_GLINK             => CHECK_GLINK,^M
      TEST_OUT          => open^M
      );
                              ^M
^M
      Inst_receive_block: receive_block PORT MAP(^M
            DATA_IN => DI_SC_VME,^M
            CS_SC_VME => CS_SC_VME_int,^M
            COMMAND_VME => COMMAND_VME_int,^M
            CS_ETH => DATA_CNTR_IN_ETH(0),^M
            CLK => CLK_SLOW,^M
            OUT_CLK => CLK_FAST,^M
            RST => HARD_OR_SOFT_RST,^M
            WRITE_ETH => DATA_CNTR_IN_ETH(1),^M
            REQ_ETH => DATA_CNTR_IN_ETH(2),^M
            ACK_ETH => ACK_ETH_int,^M
            READ_ETH => READ_ETH_int,^M
            DATA_OUT => DATA_OUT_SC_FEM ^M
      );^M
 ^M
      DATA_CNTR_OUT_ETH(0) <= READ_ETH_int;^M
      DATA_CNTR_OUT_ETH(1) <= ACK_ETH_int;^M
      ^M
      DATA_OUT_SC_FEM_IB <= "0000000"&(not GTM_LINK) when COMMAND_VME_int = X"21" ^M
                        else COMMAND_VME_int ;^M
^M
      DATA_OUT_SC <= DATA_OUT_SC_FEM_IB when (COMMAND_VME_int(5) = '1') else DATA_OUT_SC_FEM;        ^M
^M
      DATA_IO_ETH <= DATA_OUT_SC when (USE_ETH = '1' and DATA_CNTR_IN_ETH(0) = '1') else "ZZZZZZZZ";^M
^M
   BCO_OUT_inst : OBUFDS
   generic map (
      IOSTANDARD => "DEFAULT")
-:---  FEM_IB_top.vhd   57% L302   (VHDL)
```

DI_SC_VME

DI_SC_VMEを出力してみる。

```
--                DATA_OUT => DATA_OUT_SC_FEM ^M
^M
   TEST_OUT(0) <= START_CALIB; --DI_SC_VME; --START_BCO;^M
  TEST_OUT(0) <= DATA_CNTR_IN_ETH(0);--START_BCO; --MODE_BITS(2); --DI_SC_VME; --START_BCO;^M
       TEST_OUT(1) <= DATA_OUT_SC_FEM(0);--START_CALIB; --START_BCO_PERIODIC; --START_BCO_BUF;^M
       TEST_OUT(2) <= DATA_CNTR_IN_ETH(2);--LVL1_ACCEPT; --START_BCO_2BUF;^M
       TEST_OUT(3) <= READ_ETH_int;--START_BCO_PERIODIC;^M
       TEST_OUT(6 downto 4) <= TEST_OUT_SC(6 downto 4);  ^M
  TEST_OUT(7) <= START_BCO_FEM; --BCO_CLK;^M
       TEST_OUT(8) <= BCO_CLK; --DATA_STB;^M
       TEST_OUT(9) <= CLK_SLOW;^M
^M
       TEST_OUT(13 downto 10) <= (others => '0');^M
```

ETH vs USB Output

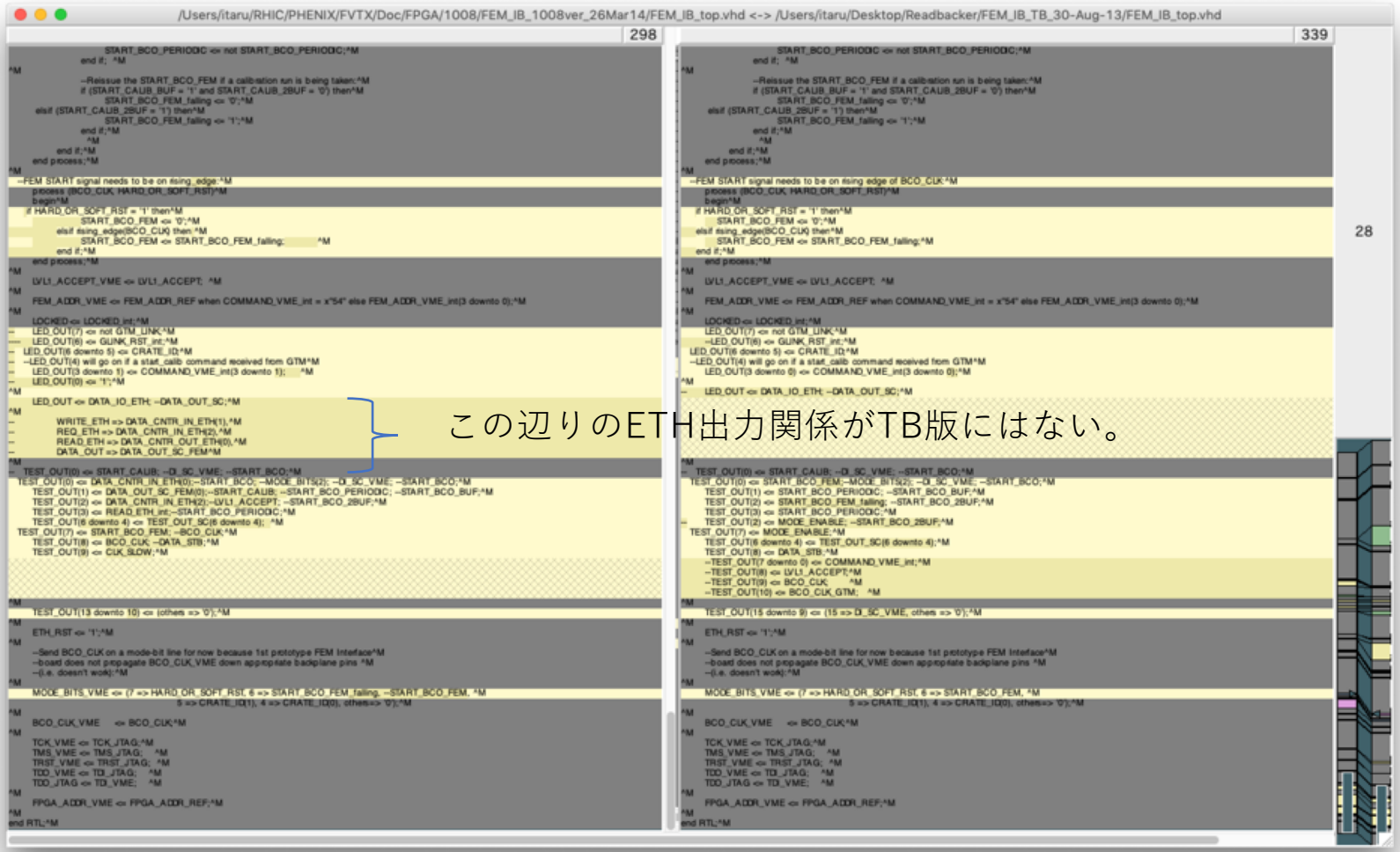


1008版とstandalone版のFEM_IB_top.vhdを比較すると、1008版のETHに出力するモジュールに相当するUSB出力モジュールがstandalone版には無いように見える。これはstandalone版に新たにUSBに出力するモジュールを追加しなければならなそう。

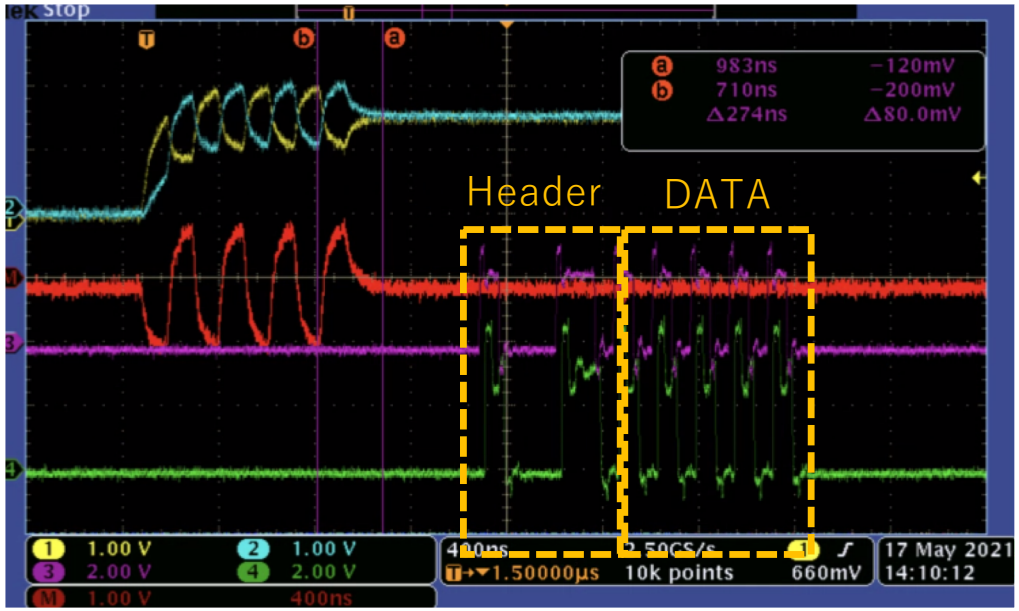
DATA_IO_USBを追加し、USE_USBでactivateするようなFunctionを追加

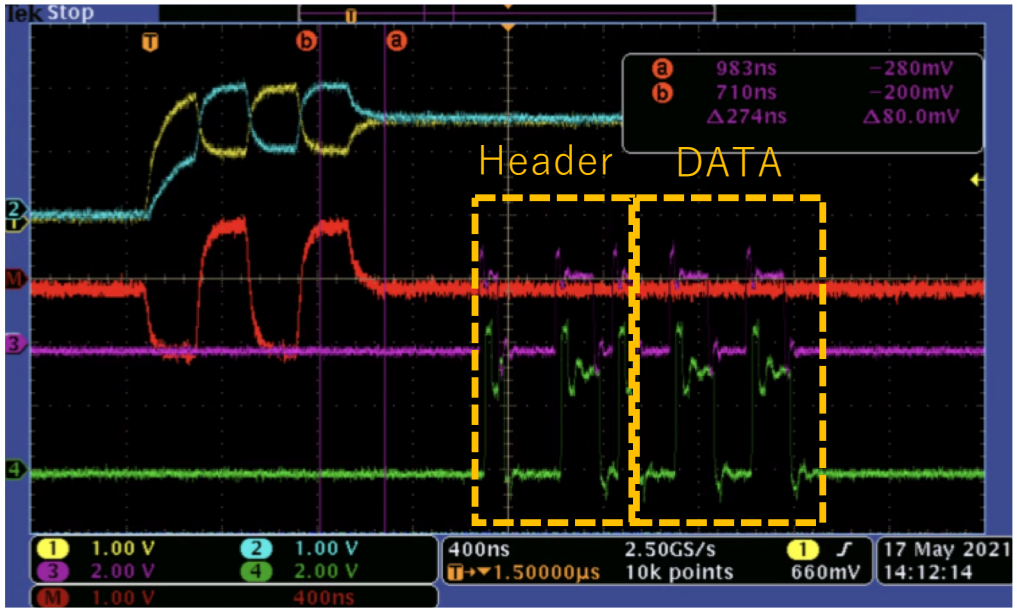
1008版のFEM_IB_top.vhd　　standalone版のFEM_IB_top.vhd

ETH vs USB Output



DI_SC_VME Monitoring





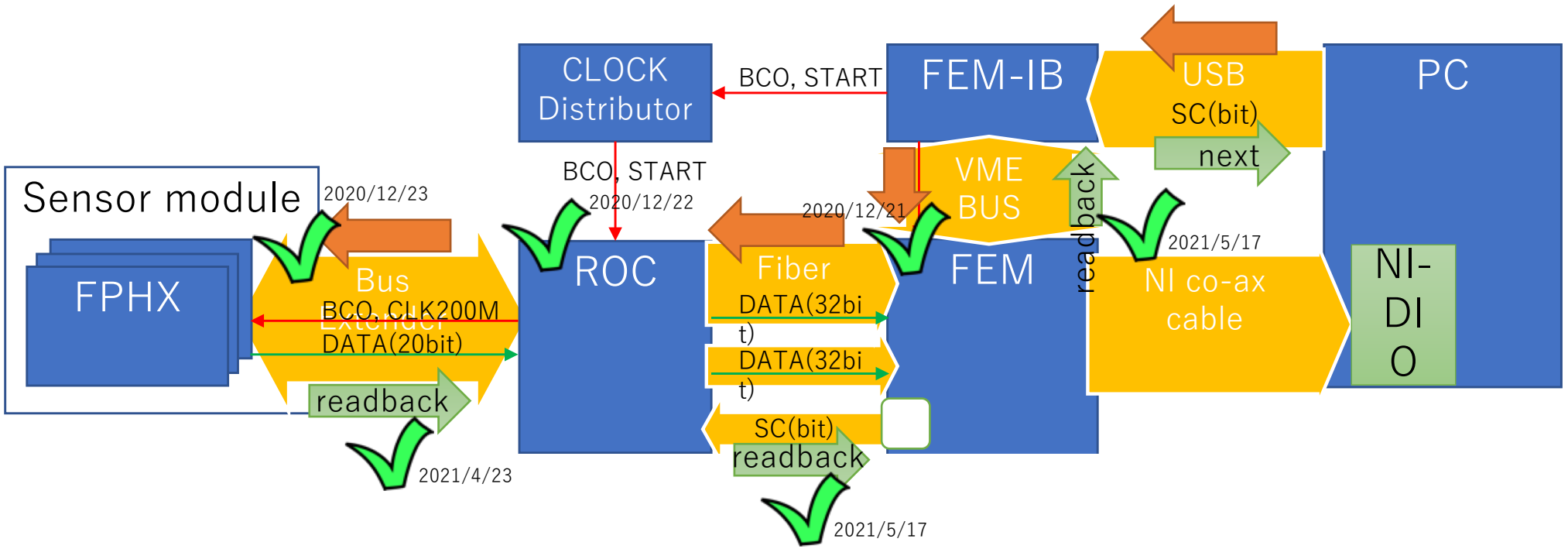
CH1: SC_OUT_0N
CH2: SC_OUT_0P
M: CH 1 -CH 2
CH3: DATA_OUT_SC_int (FEM)
CH4: DI_SC_VME (FEM-IB)

- DATA_OUT_SC_int @FEM contains header word. Readback data follows the header.
- OK, Expected Readback pattern is also observed at DI_SC_VME in FEM-IB.

FEM-IB Conclusion

- Consistent readback values are observed in FEM-IB as well.
- The next step is to implement USB module to transmit readback values to PC.

SC Transmission to Specific CHIP-ID&Module



- Tracing SC command transmission from PC to FPHX chip (done) by the end of Dec. 2020.
- Return path is completed to FEM-IB by May 17th