ROCのportを指定する FEMのデータFPGA code

経緯と現在地 2

- C-3とC-1を同時につないで測定した場合では、C-1からの測定実績がある
- 単体ではC-1で測定することができない

宇宙線測定を行う

- ROC portの指定が必要
- FEM上のデータFPGAで指定することができる

ROC C-3でのみ宇宙線測定ができる

- 先行研究から、変更箇所を引き継いでいた
- 変更箇所がどのROC portと接続されているのかは曖昧であった

信号の確認

- 変更箇所から観測できる信号を確認した
- ROC portを指定している箇所からは宇宙線が来なかった

＼今ここ／

FEM上のデータFPGAに書かれたcodeの確認

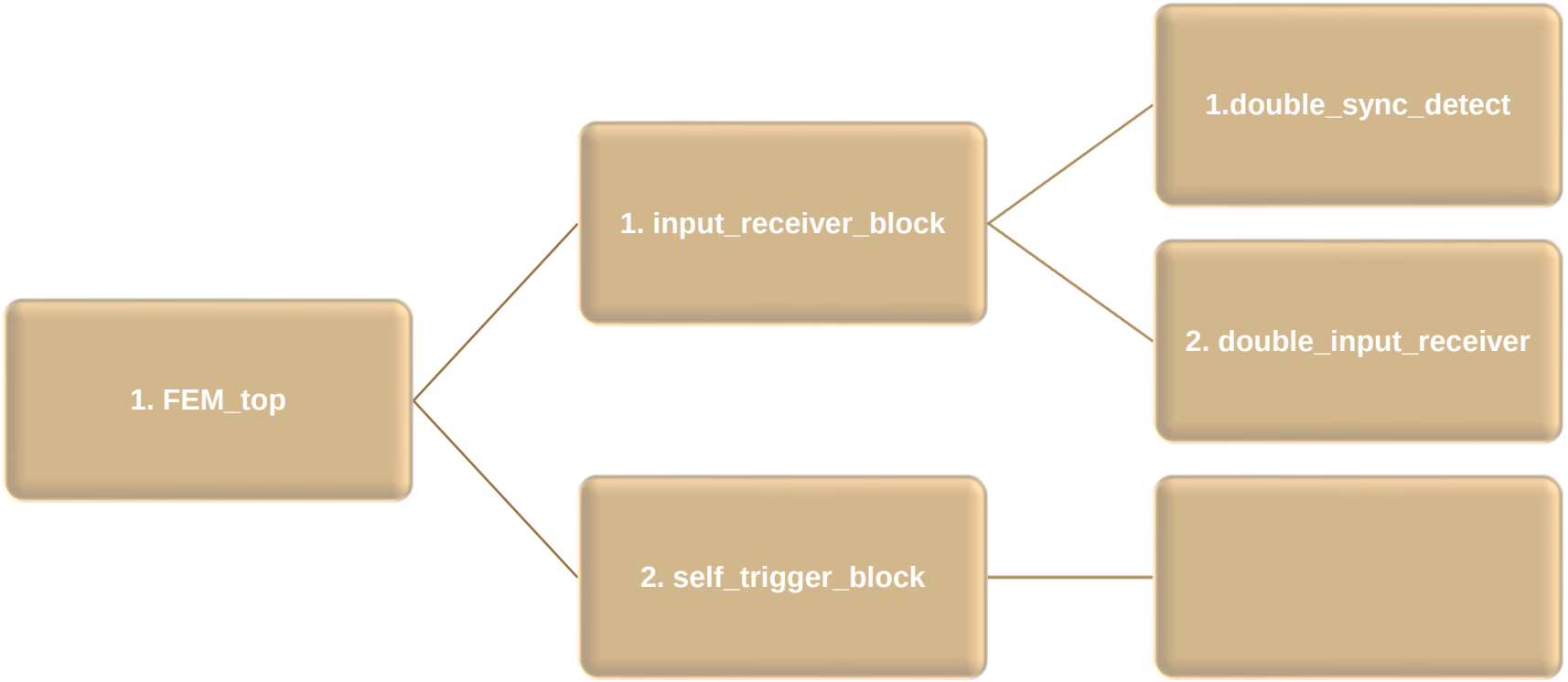
- 変更箇所の理解が足りない
- 宇宙線が通る配線を調べる必要がある

Back Up

先行研究から変更箇所であると思っていたところ　4

ROC portに関係するFEM code上のsignal name		
Data Signal Name	Sync Signal	Fiber Location
DATA_IN_0_0	SYNC_OK_0_0, SYNC_OK_0_1	top
DATA_IN_0_1	SYNC_OK_0_2, SYNC_OK_0_3	top
DATA_IN_1_0	SYNC_OK_1_0, SYNC_OK_1_1	top
DATA_IN_1_1	SYNC_OK_1_2, SYNC_OK_1_3	top
DATA_IN_3_0	SYNC_OK_3_0, SYNC_OK_3_1	bottom
DATA_IN_3_1	SYNC_OK_3_2, SYNC_OK_3_3	bottom
DATA_IN_2_0	SYNC_OK_2_0, SYNC_OK_2_1	bottom
DATA_IN_2_1	SYNC_OK_2_2, SYNC_OK_2_3	bottom

変更箇所に関連のあるブロック 5



FEM code上変更箇所

変更するように指示されているところ（赤字）

- DATA_0～3は固定
- DATA_IN_i_j_BUF(0)
  [i = 0, 1, 2, 3
  j = 0, 1]
  をROC portに合わせて指定する

FEM_top.vhd

```
1163        Inst_self_trigger_block: self_trigger_block PORT MAP(
1164                    DATA_0 => DATA_IN_1_0_BUF(0),
1165                    DATA_1 => DATA_IN_3_1_BUF(0),
1166                    DATA_2 => DATA_IN_2_0_BUF(0),
1167                    DATA_3 => DATA_IN_2_1_BUF(0),
1168                    SYNC_OK_0 => SYNC_OK_3_0,
1169                    SYNC_OK_1 => SYNC_OK_3_2,
1170                    SYNC_OK_2 => SYNC_OK_2_0,
1171                    SYNC_OK_3 => SYNC_OK_2_2,
1172                    DELAY => "1111",
1173                    CLK => BCO_CLK,
1174                    RST => GLOBAL_RST,
1175                    DATA_OUT => LVL1_ACCEPT_SELF_TRIG
1176        );
```

DATA_IN_0_0_BUFを追う

1

FEM_top.vhd

```
409     signal DATA_IN_0_0_BUF                  : std_logic_vector(31 downto 0);
410     signal DATA_IN_0_1_BUF                  : std_logic_vector(31 downto 0);
411     signal DATA_IN_1_0_BUF                  : std_logic_vector(31 downto 0);
412     signal DATA_IN_1_1_BUF                  : std_logic_vector(31 downto 0);
413     signal DATA_IN_2_0_BUF                  : std_logic_vector(31 downto 0);
414     signal DATA_IN_2_1_BUF                  : std_logic_vector(31 downto 0);
415     signal DATA_IN_3_0_BUF                  : std_logic_vector(31 downto 0);
416     signal DATA_IN_3_1_BUF                  : std_logic_vector(31 downto 0);
```

DATA_IN_0_0_BUFを追う

FEM_top.vhd

```
633             process (CLK_0_0_int,GLOBAL_RST)
634             begin
635              if GLOBAL_RST = '1' then
636                LSB_0_0_BUF <= '0';
637                MSB_0_0_BUF <= '0';
638                DATA_IN_0_0_BUF(15 downto 0) <= (others => '0');
639              elsif rising_edge(CLK_0_0_int) then
640                LSB_0_0_BUF <= LSB_0_0;
641                MSB_0_0_BUF <= MSB_0_0;
642                DATA_IN_0_0_BUF(15 downto 0) <= DATA_IN_0_0(15 downto 0);
643              end if;
644            end process;
```

1

FEM_top.vhd

```
646             process (CLK_0_1_int,GLOBAL_RST
647             begin
648              if GLOBAL_RST = '1' then
649                LSB_0_1_BUF <= '0';
650                MSB_0_1_BUF <= '0';
651                DATA_IN_0_0_BUF(31 downto 16) <= (others => '0');
652              elsif rising_edge(CLK_0_1_int) then
653                LSB_0_1_BUF <= LSB_0_1;
654                MSB_0_1_BUF <= MSB_0_1;
655                DATA_IN_0_0_BUF(31 downto 16) <= DATA_IN_0_0(31 downto 16);
656              end if;
657            end process;
```

DATA_IN_0_0_BUFを追う

1_1

FEM_top.vhd

```
859         Inst_input_receiver_block: input_receiver_block PORT MAP(
860             DATA_IN_0_0 => DATA_IN_0_0_BUF,
861             DATA_IN_1_0 => DATA_IN_1_0_BUF,
862             DATA_IN_2_0 => DATA_IN_2_0_BUF,
863             DATA_IN_3_0 => DATA_IN_3_0_BUF,
864             DATA_IN_0_1 => DATA_IN_0_1_BUF,
865             DATA_IN_1_1 => DATA_IN_1_1_BUF,
866             DATA_IN_2_1 => DATA_IN_2_1_BUF,
867             DATA_IN_3_1 => DATA_IN_3_1_BUF,
                                ≈
938         );
```

DATA_IN_0_0を追う

1_1

input_receiver_block.vhd

```
30          entity input_receiver_block is
31            Port (  DATA_IN_0_0                    : in std_logic_vector(31 downto 0);
32                    DATA_IN_1_0                    : in std_logic_vector(31 downto 0);
33                    DATA_IN_2_0                    : in std_logic_vector(31 downto 0);
34                    DATA_IN_3_0                    : in std_logic_vector(31 downto 0);
35                    DATA_IN_0_1                    : in std_logic_vector(31 downto 0);
36                    DATA_IN_1_1                    : in std_logic_vector(31 downto 0);
37                    DATA_IN_2_1                    : in std_logic_vector(31 downto 0);
38                    DATA_IN_3_1                    : in std_logic_vector(31 downto 0);
                                          ≈
109                   );
110         end input_receiver_block;
```

DATA_IN_0_0を追う

1_1_1

input_receiver_block.vhd

```
243 Inst_double_sync_detect_0_0: double_sync_detect PORT MAP(
244         DATA_IN_0 => DATA_IN_0_0(15 downto 0),
245         DATA_IN_1 => DATA_IN_0_0(31 downto 16),
246         MSB_0 => MSB_0_0,
247         LSB_0 => LSB_0_0,
248         MSB_1 => MSB_0_1,
249         LSB_1 => LSB_0_1,
250         RST => RST,
251         CLK_IN_0 => CLK_0_0,
252         CLK_IN_1 => CLK_0_1,
253         SYNC_OK_0 => SYNC_OK_int_0_0,
254         SYNC_OK_1 => SYNC_OK_int_0_1
255 );
```

DATA_IN_0_0を追う

1_1_(1)

input_receiver_block.vhd

```
243   Inst_double_sync_detect_0_0: double_sync_detect PORT MAP(
244               DATA_IN_0 => DATA_IN_0_0(15 downto 0),
245               DATA_IN_1 => DATA_IN_0_0(31 downto 16),
246               MSB_0 => MSB_0_0,
247               LSB_0 => LSB_0_0,
248               MSB_1 => MSB_0_1,
249               LSB_1 => LSB_0_1,
250               RST => RST,
251               CLK_IN_0 => CLK_0_0,
252               CLK_IN_1 => CLK_0_1,
253               SYNC_OK_0 => SYNC_OK_int_0_0,
254               SYNC_OK_1 => SYNC_OK_int_0_1
255   );
```

DATA_IN_0_0を追う

1_1_(2)

input_receiver_block.vhd

```
271 Inst_double_input_receiver_0: double_input_receiver PORT MAP(
272         DATA_IN_0 => DATA_IN_0_0(15 downto 0),
273         DATA_IN_1 => DATA_IN_0_0(31 downto 16),
274         DATA_IN_2 => DATA_IN_0_1(15 downto 0),
275         DATA_IN_3 => DATA_IN_0_1(31 downto 16),
276         RST => RST,
277         SYNC_OK_0 => SYNC_OK_int_0_0,
278         SYNC_OK_1 => SYNC_OK_int_0_1,
279         SYNC_OK_2 => SYNC_OK_int_0_2,
280         SYNC_OK_3 => SYNC_OK_int_0_3,
281         CLK_IN_0 => CLK_0_0,
282         CLK_IN_1 => CLK_0_1,
283         CLK_IN_2 => CLK_0_2,
284         CLK_IN_3 => CLK_0_3,
285         CLK_OUT => CLK_OUT,
286         DATA_OUT => DATA_OUT_0
287 );
```

DATA_IN_0を追う

1_1_2

double_input_receiver.vhd

```
30  entity double_input_receiver is
31      Port ( DATA_IN_0 : in std_logic_vector(15 downto 0);
32             DATA_IN_1 : in std_logic_vector(15 downto 0);
33             DATA_IN_2 : in std_logic_vector(15 downto 0);
34             DATA_IN_3 : in std_logic_vector(15 downto 0);
35             RST : in std_logic;
36             SYNC_OK_0 : in std_logic;
37             SYNC_OK_1 : in std_logic;
38             SYNC_OK_2 : in std_logic;
39             SYNC_OK_3 : in std_logic;
40             CLK_IN_0 : in std_logic;
41             CLK_IN_1 : in std_logic;
42             CLK_IN_2 : in std_logic;
43             CLK_IN_3 : in std_logic;
44             CLK_OUT : in std_logic;
45             DATA_OUT : out std_logic_vector(31 downto 0));
46  end double_input_receiver;
```

DATA_IN_0を追う

1_1_2

double_input_receiver.vhd

```
30  entity double_input_receiver is
31      Port ( DATA_IN_0 : in std_logic_vector(15 downto 0);
32             DATA_IN_1 : in std_logic_vector(15 downto 0);
33             DATA_IN_2 : in std_logic_vector(15 downto 0);
34             DATA_IN_3 : in std_logic_vector(15 downto 0);
35             RST : in std_logic;
36             SYNC_OK_0 : in std_logic;
37             SYNC_OK_1 : in std_logic;
38             SYNC_OK_2 : in std_logic;
39             SYNC_OK_3 : in std_logic;
40             CLK_IN_0 : in std_logic;
41             CLK_IN_1 : in std_logic;
42             CLK_IN_2 : in std_logic;
43             CLK_IN_3 : in std_logic;
44             CLK_OUT : in std_logic;
45             DATA_OUT : out std_logic_vector(31 downto 0));
46  end double_input_receiver;
```

DATA_IN_0_0_BUFを追う

1_2

FEM_top.vhd

```
1163.       Inst_self_trigger_block: self_trigger_block PORT MAP(
1164.                   DATA_0 => DATA_IN_1_0_BUF(0),
1165.                   DATA_1 => DATA_IN_3_1_BUF(0),
1166.                   DATA_2 => DATA_IN_2_0_BUF(0),
1167.                   DATA_3 => DATA_IN_2_1_BUF(0),
1168.                   SYNC_OK_0 => SYNC_OK_3_0,
1169.                   SYNC_OK_1 => SYNC_OK_3_2,
1170.                   SYNC_OK_2 => SYNC_OK_2_0,
1171.                   SYNC_OK_3 => SYNC_OK_2_2,
1172.                   DELAY => "1111",
1173.                   CLK => BCO_CLK,
1174.                   RST => GLOBAL_RST,
1175.                   DATA_OUT => LVL1_ACCEPT_SELF_TRIG
1176.       );
```

Back Up