Initial tests of FEB2_slice_board

Jaroslav Bán

May 26, 2020

Introduction

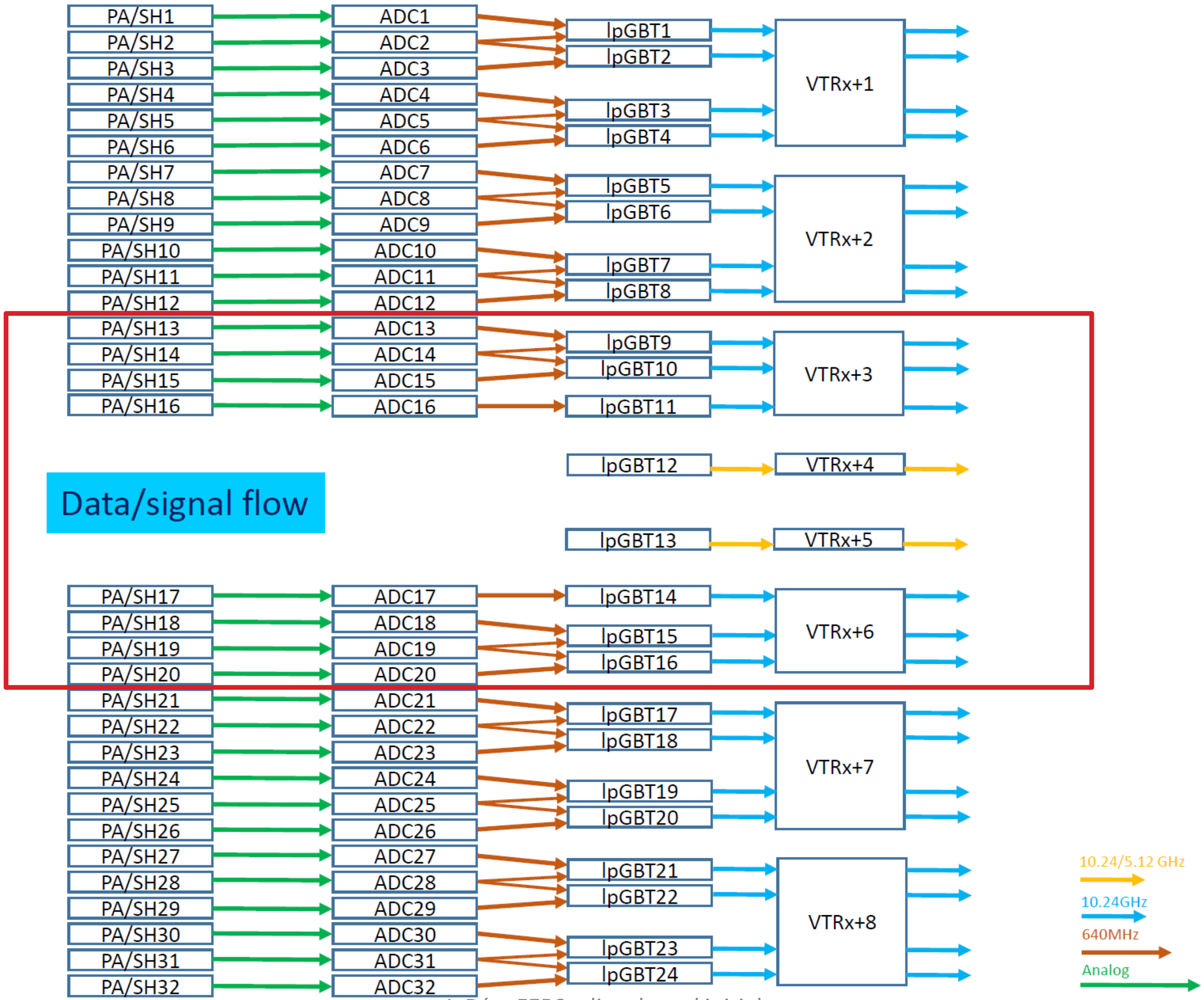
❖ FEB2_slice_board is a step of planned gradual FEB2 development

1. Analog Testboard (2019)
   - Integrates 2 LAUROC1 PA/S chips, 2 COLUTAv2 ADC chips, 1 lpGBT chip
   - Limited to reading out 2 LAr channels (cf. 128 on FEB2)
   - Used to demonstrate full readout chain of PA/S + ADC + optical data links
   - We propose to use it as initial testing hardware for FEB2_slice_board

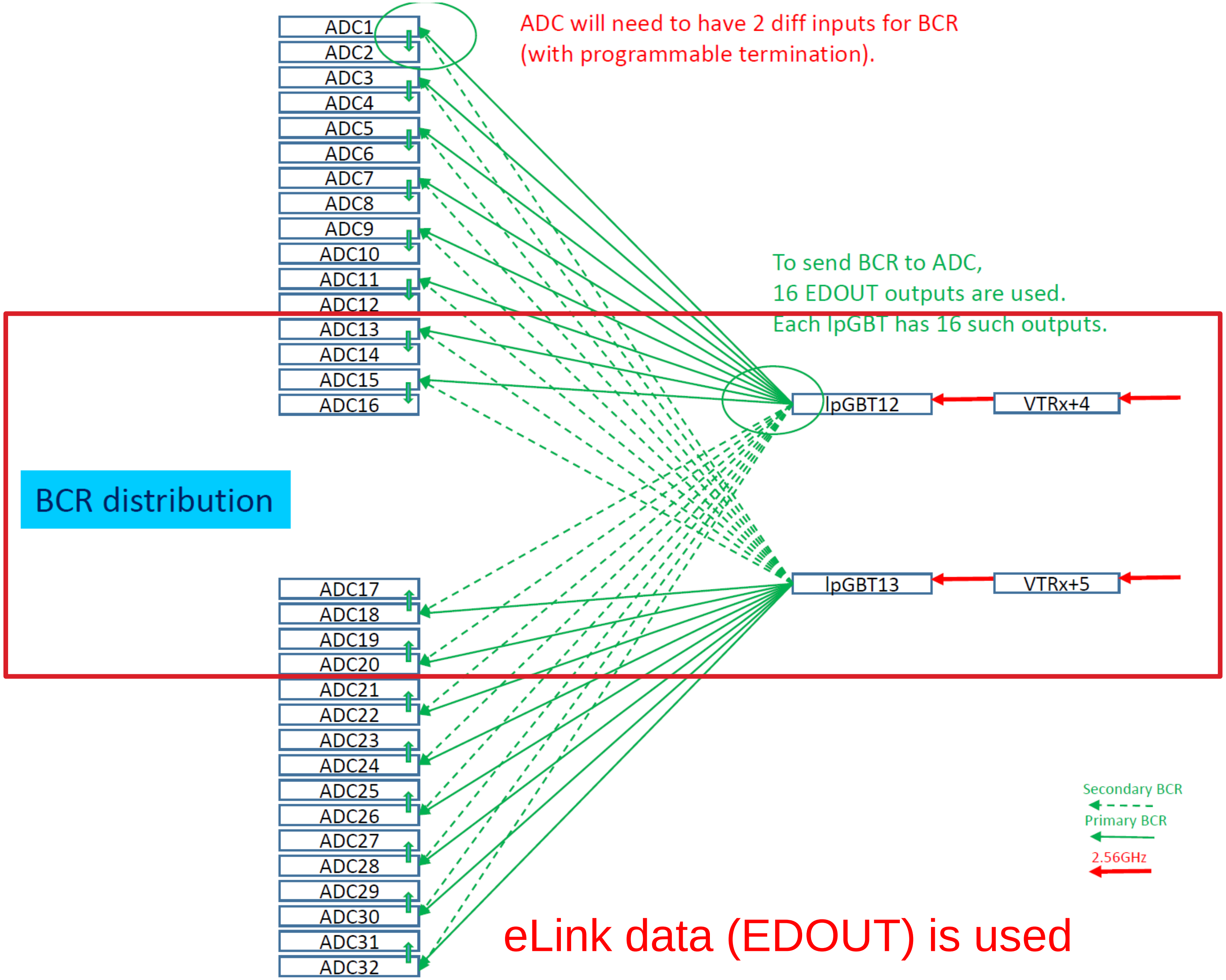
2. "FEB2_slice_board" (2020)
   - Integrates up to 8 (LAUROC2 PA/S + COLUTAv3 ADC + lpGBT) chips
   - Capable of reading out up to 32 LAr channels (cf. 128 on FEB2)
   - Use to demonstrate multichannel performance, Control links, radtol power, …

3. FEB2 Prototype (2021-2022)

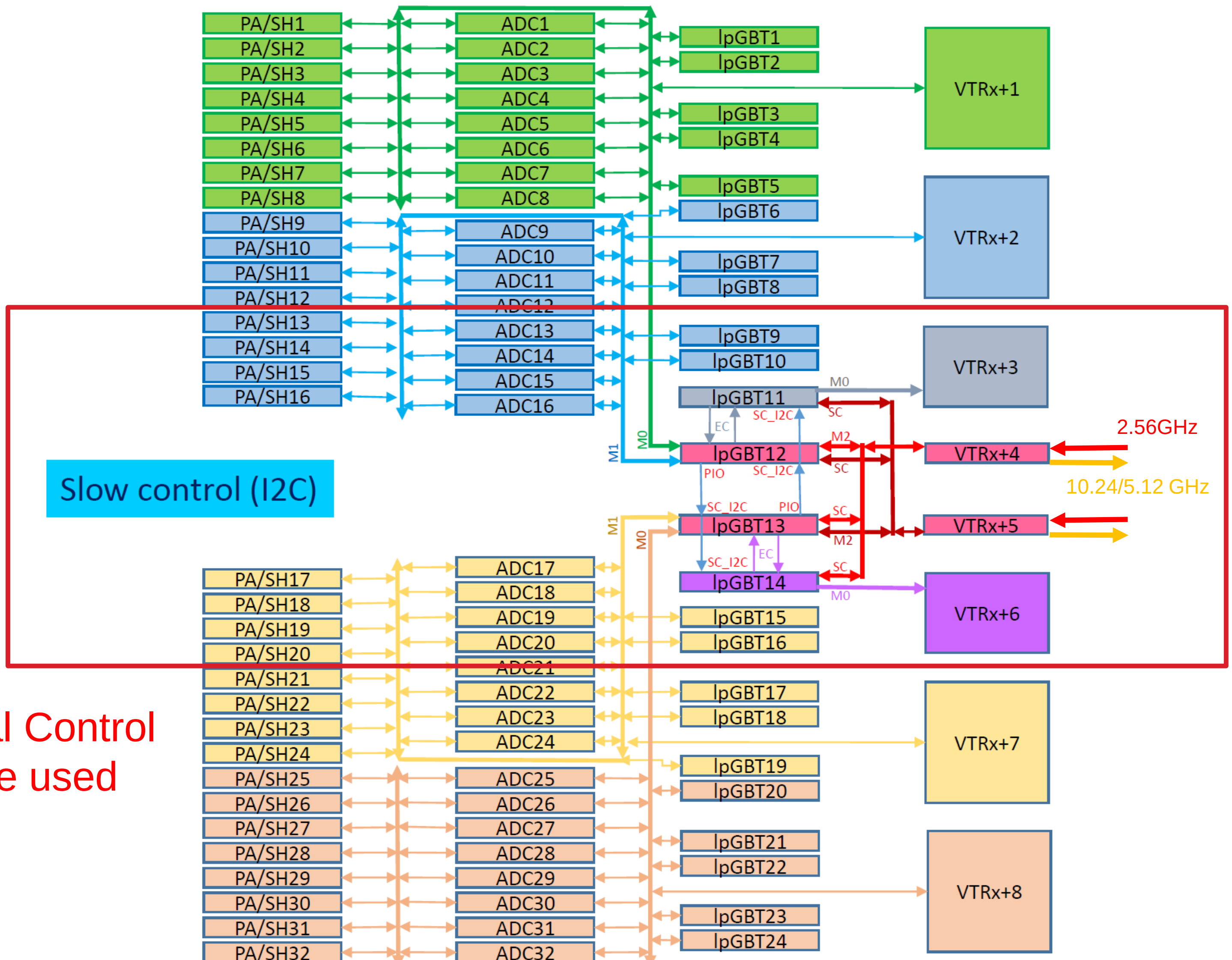
FEB2_slice_board data flow



FEB2_slice_board fast control flow

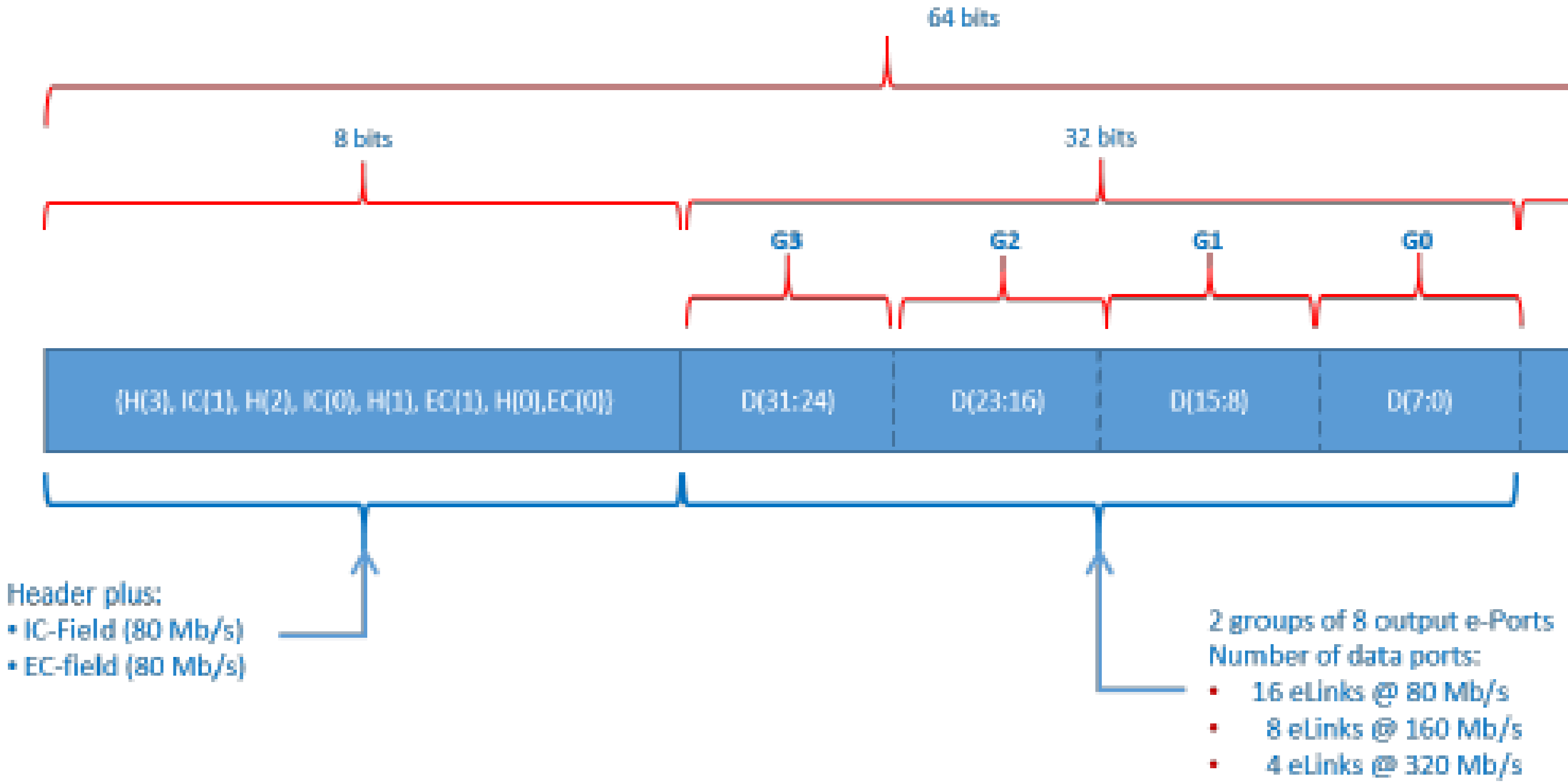


FEB2_slice_board slow control flow



LpGBT serial Control Channels are used

LpGBT downlink frame



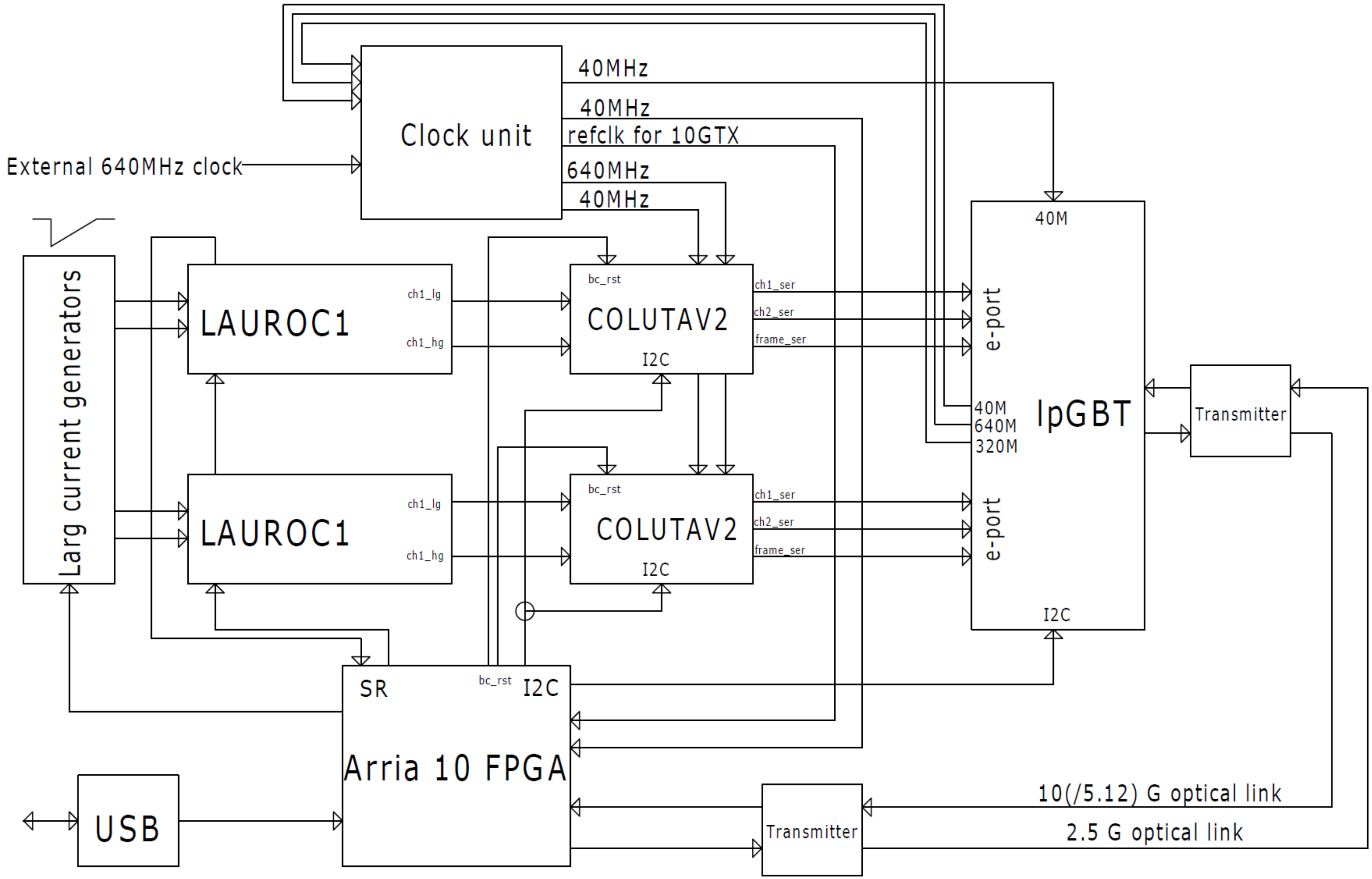
LpGBT downlink frame usage

-user can send 32 eLink bits down to the LpGBT every 25ns
-we use 16 e-Links for BCR so each 12.5ns we can send 1 bit
and implement correct BCR timing

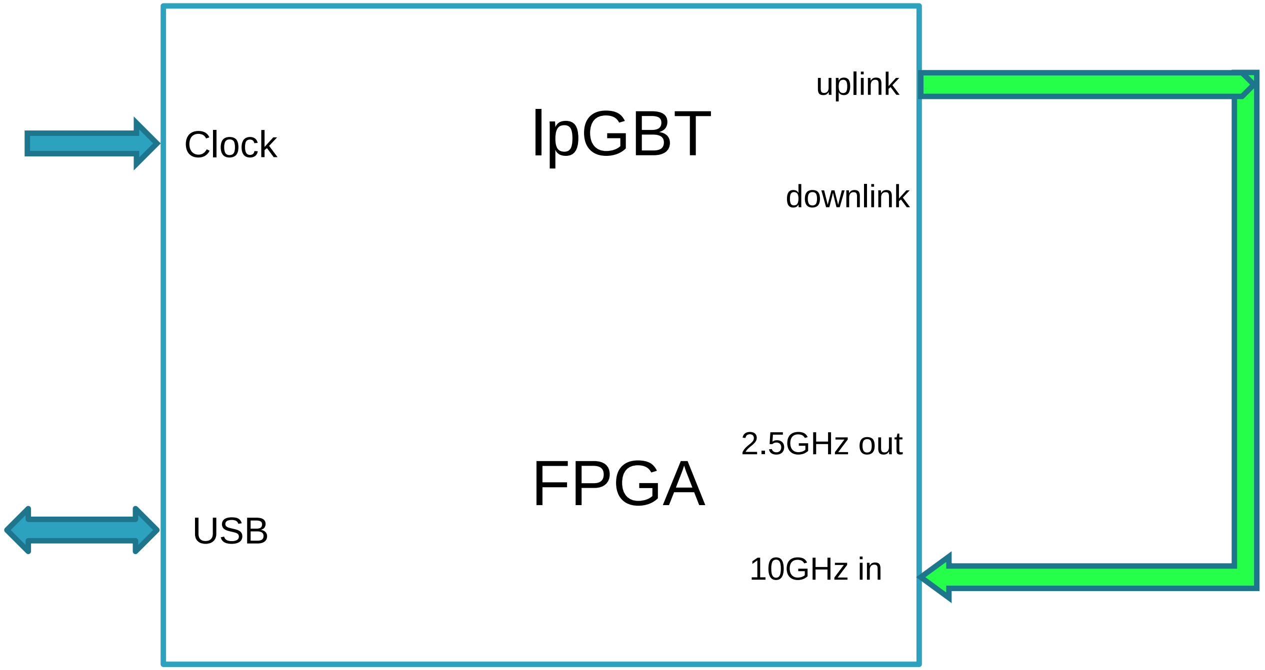
-The two bits IC[1:0] from subsequent frames are demultiplexed
to form 8-bit words which follow a frame-based protocol.
-The protocol allows access to LpGBT internal registers
and through them program its I2C master busses
(This is a two step process.)

-EC[1:0] bits are going to be used for configuring
lpGBT11&VTRx+3 from LpGBT12 and
lpGBT14&VTRx+6 from LpGBT13
(This is a tree step process.)

Analog Testboard Block Diagram

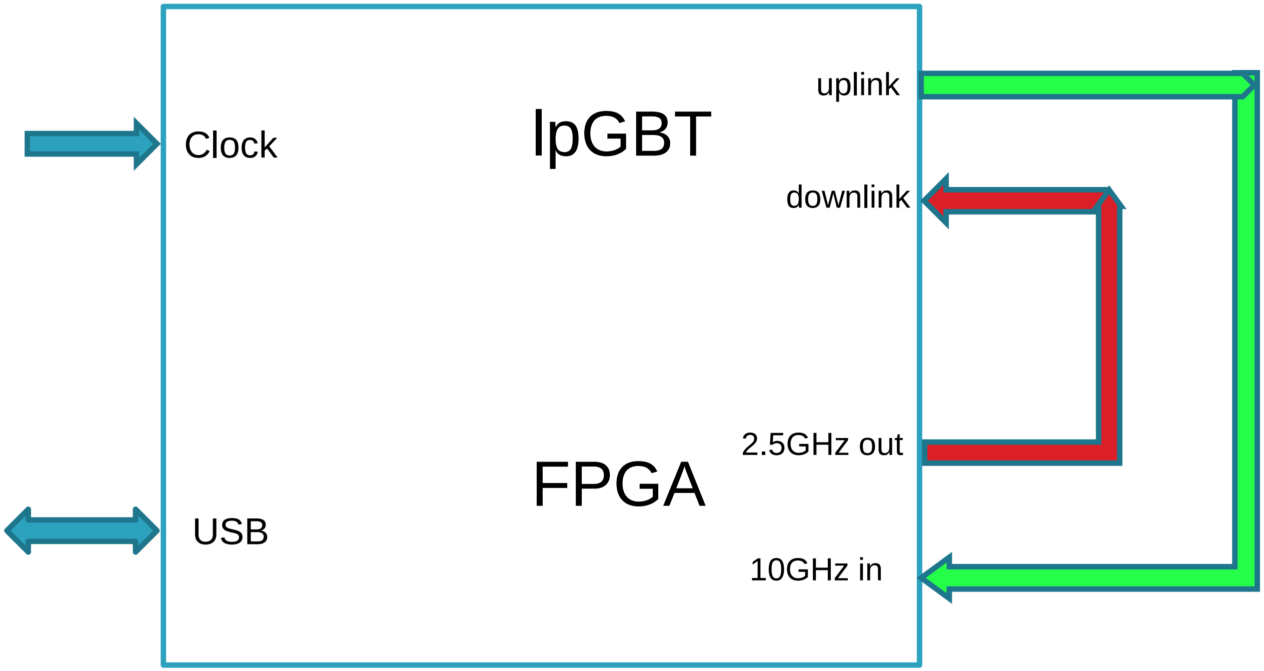


Analog_Testboard preparation(1)



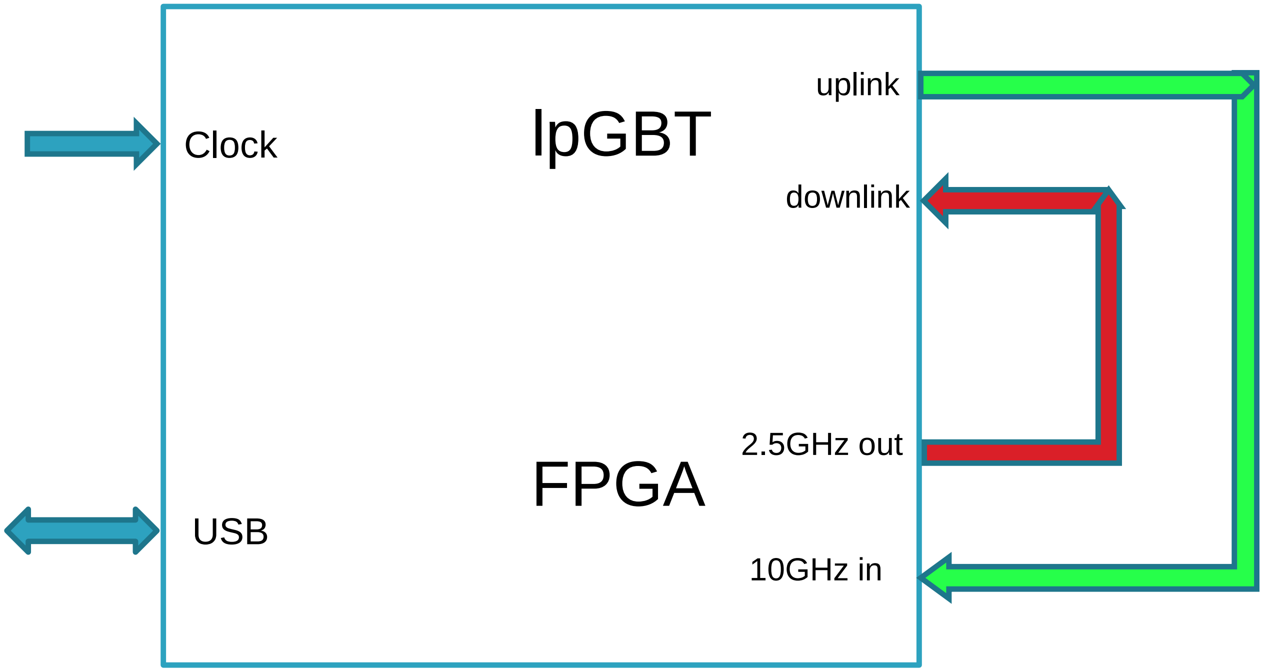
-restore original test set-up first
-program Coluta and lpGBT using FPGA I2C
-read test data from Coluta

Analog_Testboard preparation(2)



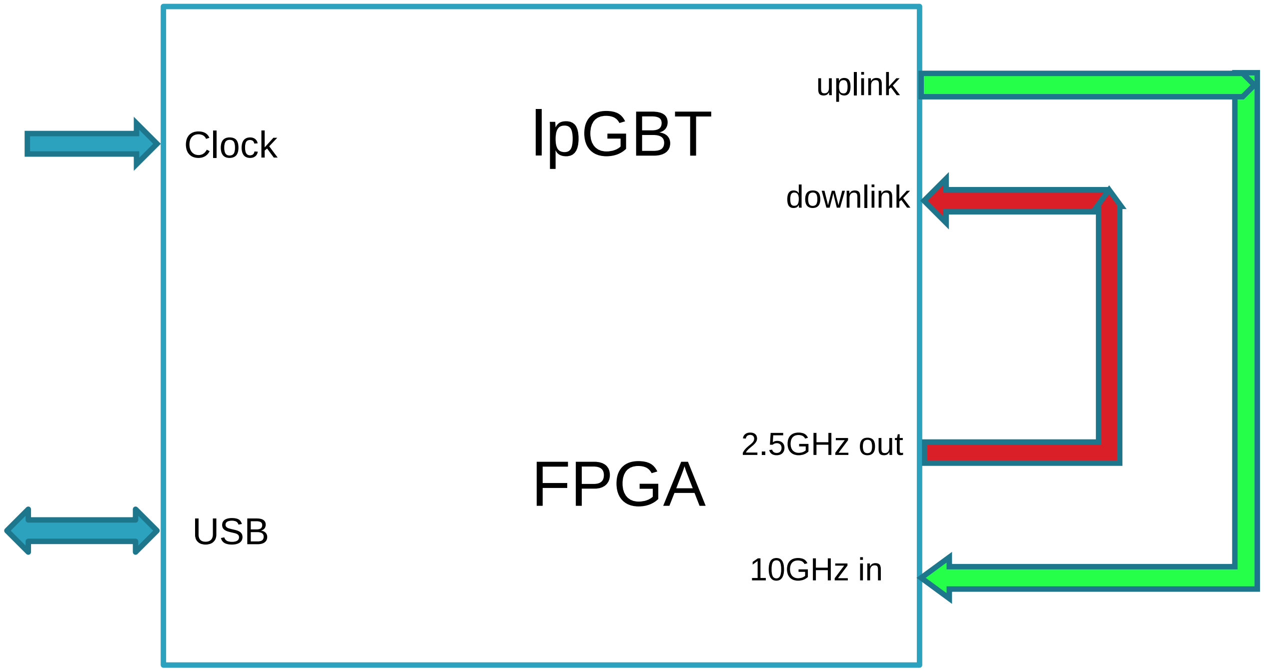
-change LpGBT LOCKMODE to 1'b1 (reference –less locking, recovering frequency from downlink data stream)
-change LpGBT mode of operation to 10Gbps/FEC5/Transceiver
-initialize firmware sending data to the downlink channel
-program Coluta and lpGBT using FPGA I2C
-read test data from Coluta
-work until you see correct data. It will be a good step forward

Analog_Testboard preparation(3)



-set LOCKMODE==1’b1 with MODE[3:0]==10Gbps/FEC5/Transceiver
-learn how to establish reliable serial connection over IC/EC channel
-burn such a minimal LpGBT configuration (PLL/CDR, equalizer, line driver, EC/IC channel settings) into LpGBT e-fuses
-use for that FPGA I2C bus and the LpGBT slave I2C channel

Analog_Testboard preparation(4)

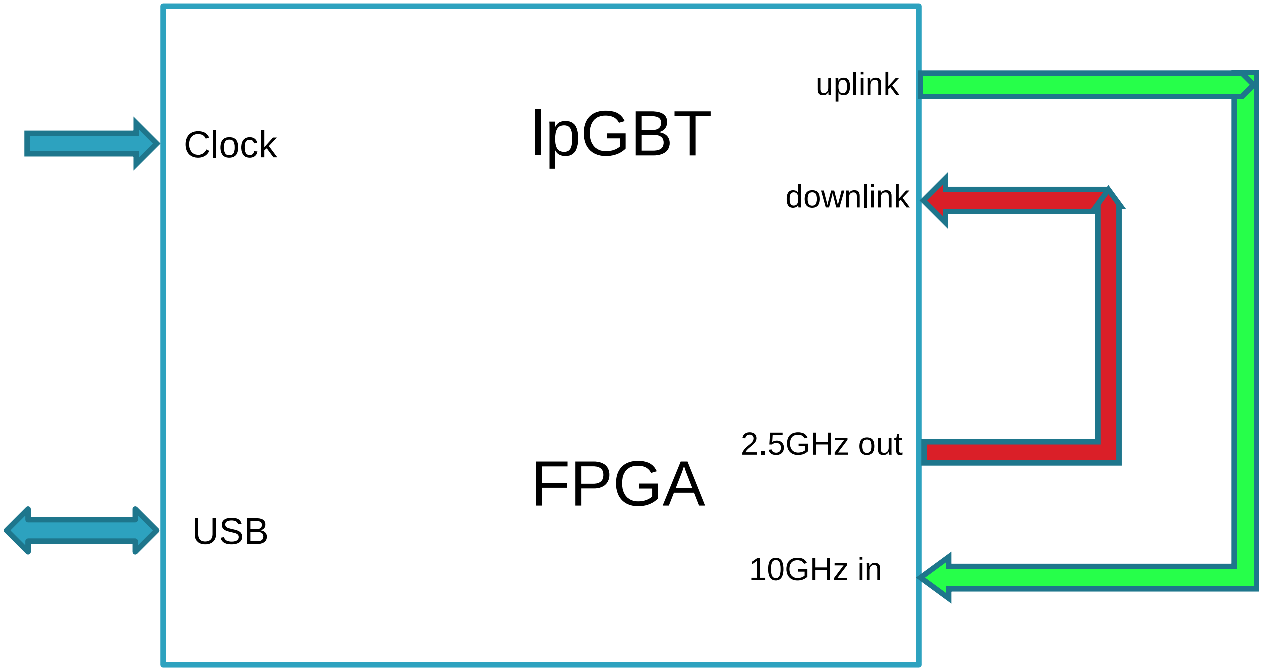


With LOCKMODE==1’b1, MODE[3:0]==10Gbps/FEC5/Transceiver and working IC/EC channels:

-Using IC channel only, program all internal LpGBT registers needed to get the same data as done in test (1). To configure the Coluta chip, FPGA I2C busses are allowed to be used (due to the board wiring).

-read test data from Coluta

Analog_Testboard preparation(5)



With LOCKMODE==1'b1, MODE[3:0]==10Gbps/FEC5/Transceiver and working IC/EC channels:

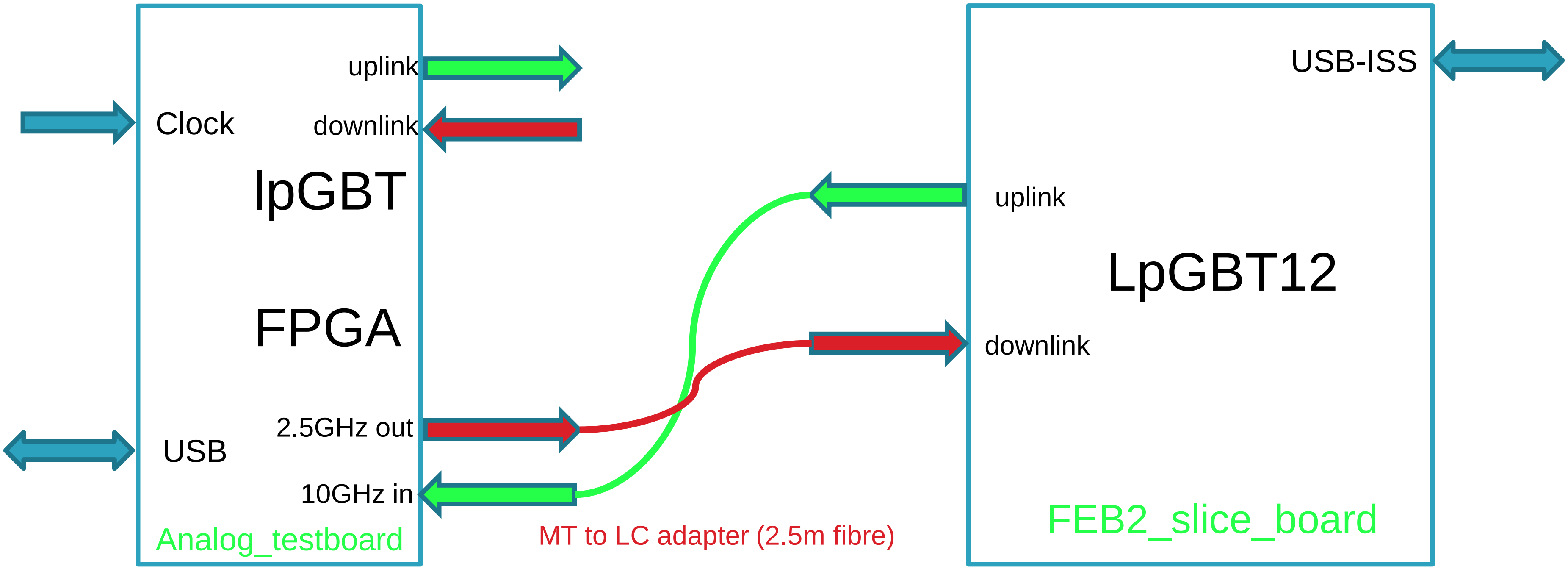
-Using IC channel only, configure LpGBT to generate the BCR signals

-generate "virtual" BCR signals

-read-back EDOUT signals set by previous actions

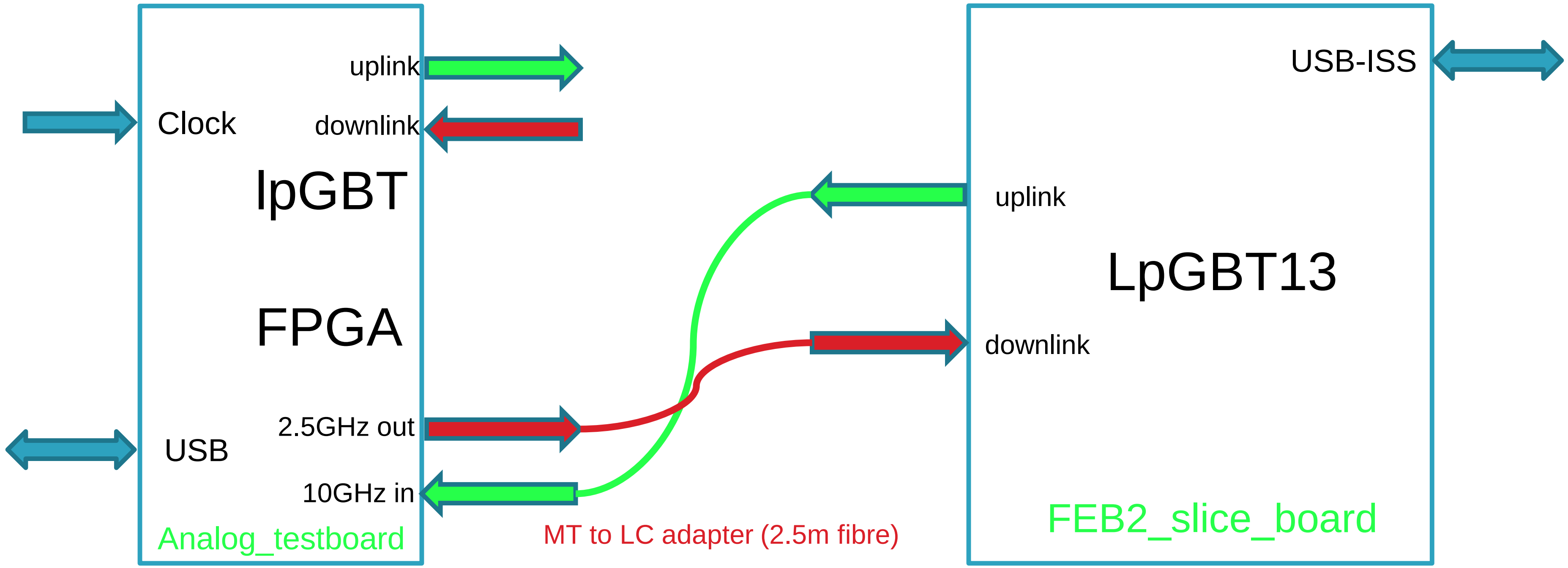
-if successful, we are prepared to move to test the FEB2_slice_board

FEB2_slice_board test(1)



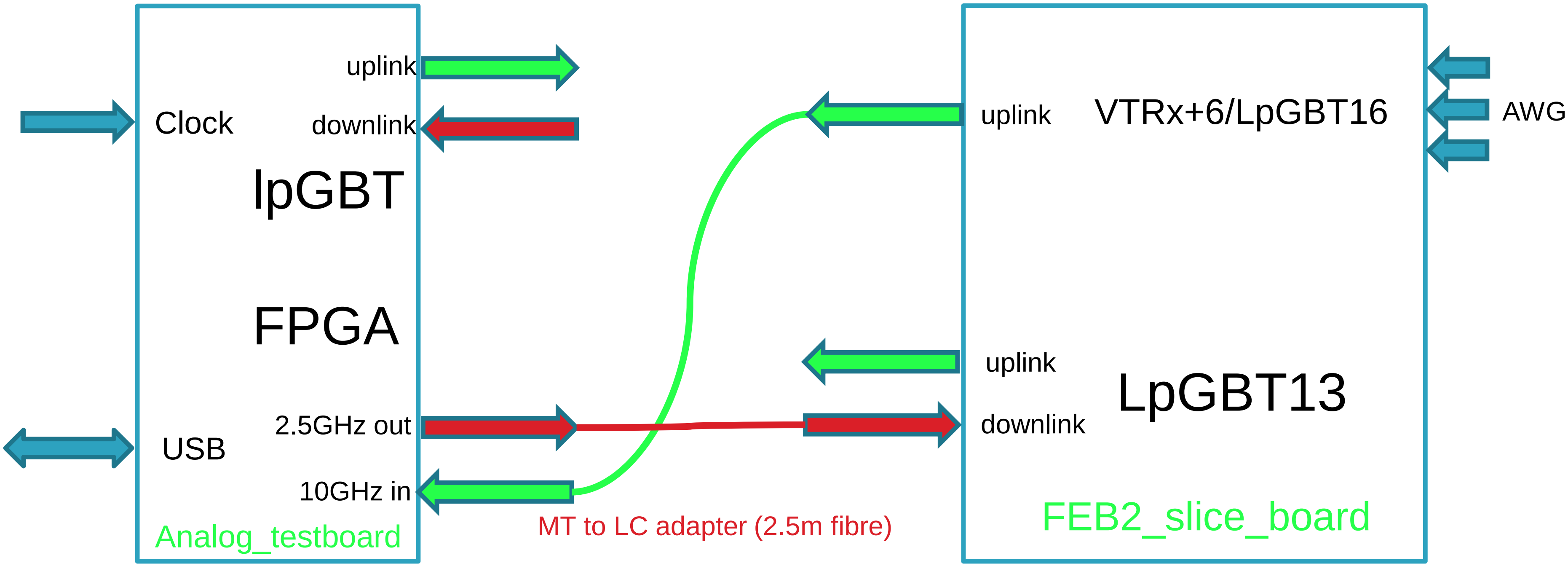
-set LpGBT to Transceiver mode, lockmode to downlink data stream
-perform some tests using LpGBT built in test features
-establish reliable serial connection over IC/EC channel
-burn such a minimal LpGBT configuration into LpGBT e-fuses
-use for that USB-ISS dongle

FEB2_slice_board test(2)



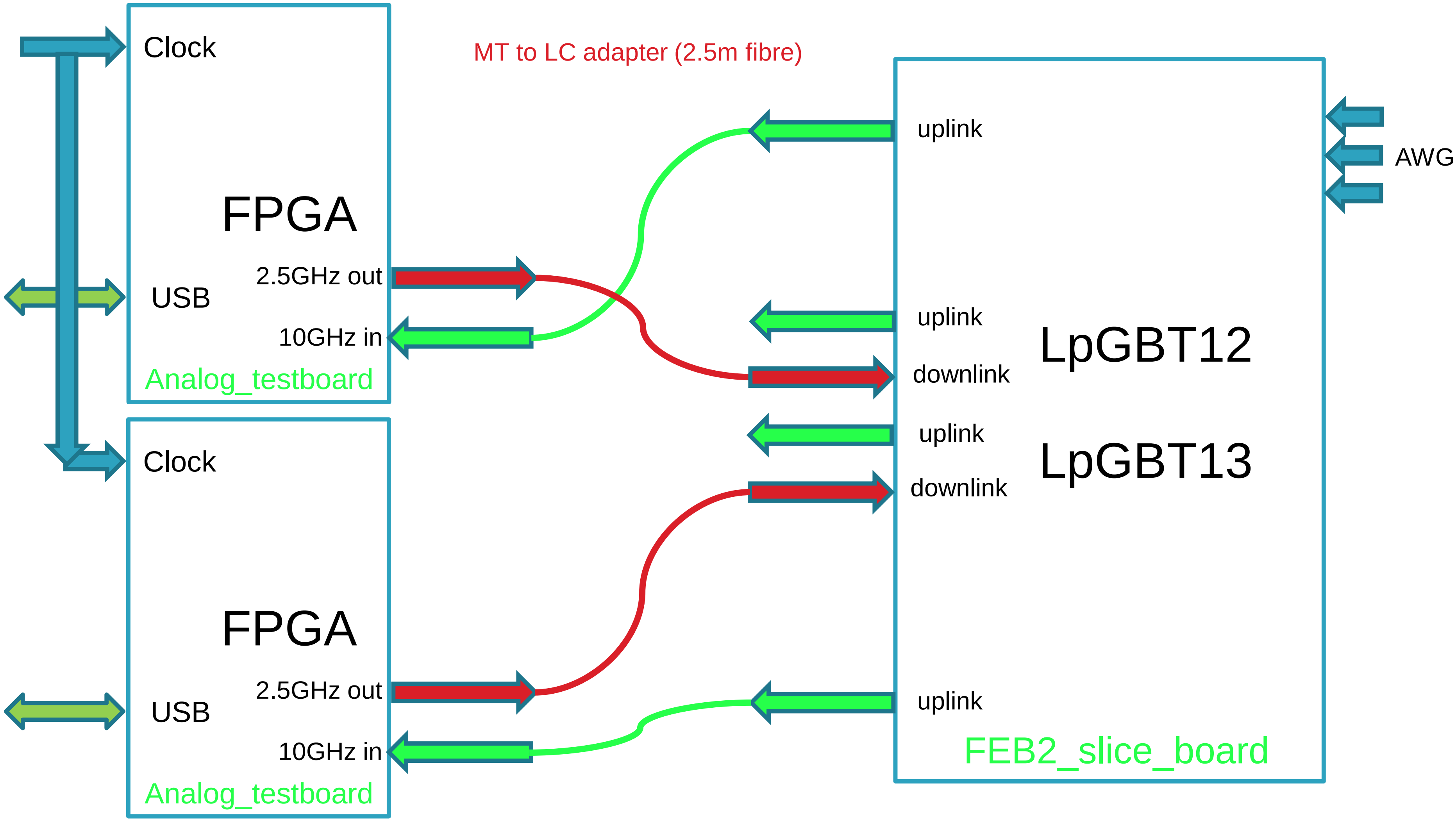
-set LpGBT to Transceiver mode, lockmode to downlink data stream
-perform some tests using LpGBT built in test features
-establish reliable serial connection over IC/EC channel
-burn such a minimal LpGBT configuration into LpGBT e-fuses
-use for that USB-ISS dongle

FEB2_slice_board test(3)

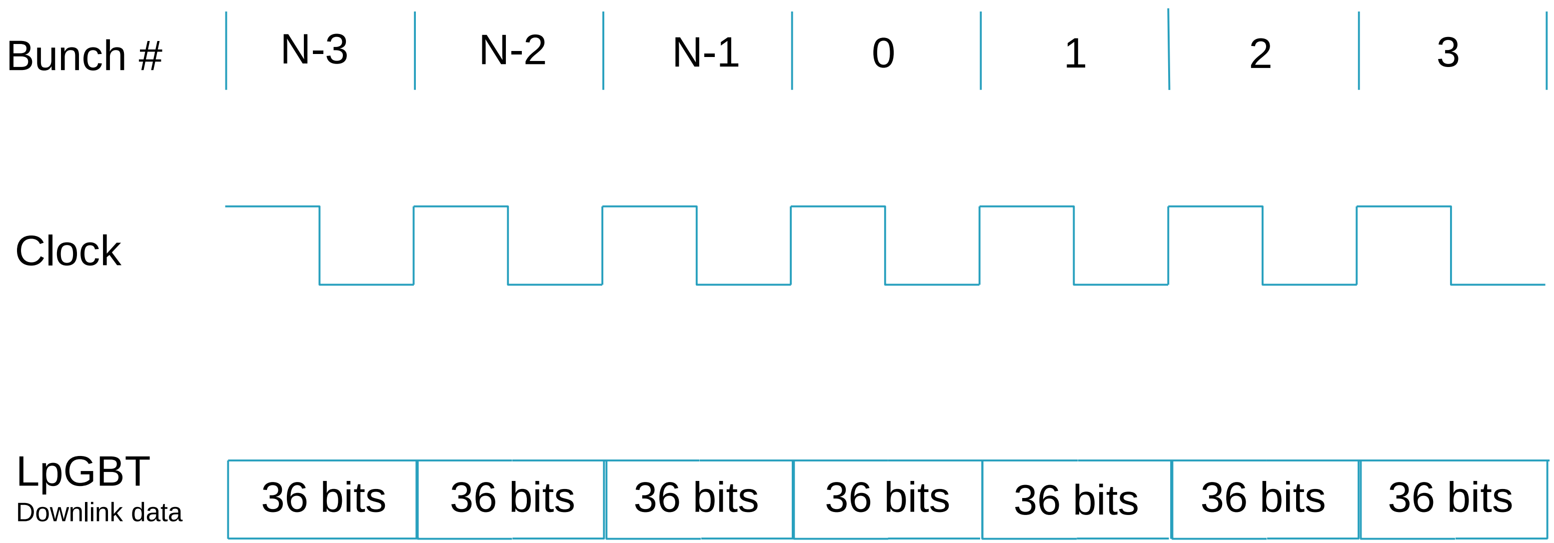


-configure all chips on LpGBT13 I2C busses
-take Coluta test data
-take the SAR input data
-take Coluta sine wave data
-take AWG physics data
-test BCR generating firmware

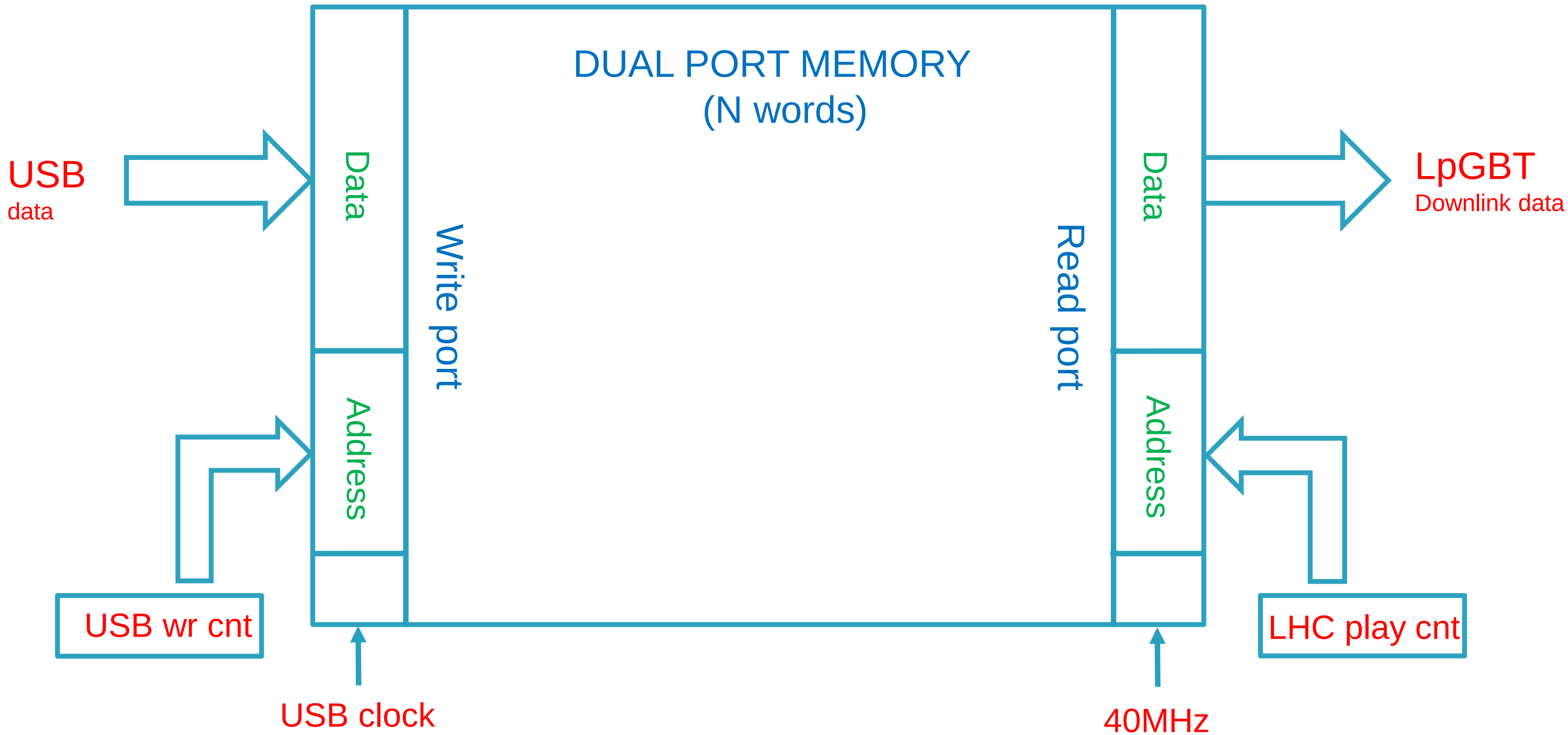
FEB2_slice_board test(3)



LpGBT uplink data



LpGBT uplink data generation mechanism



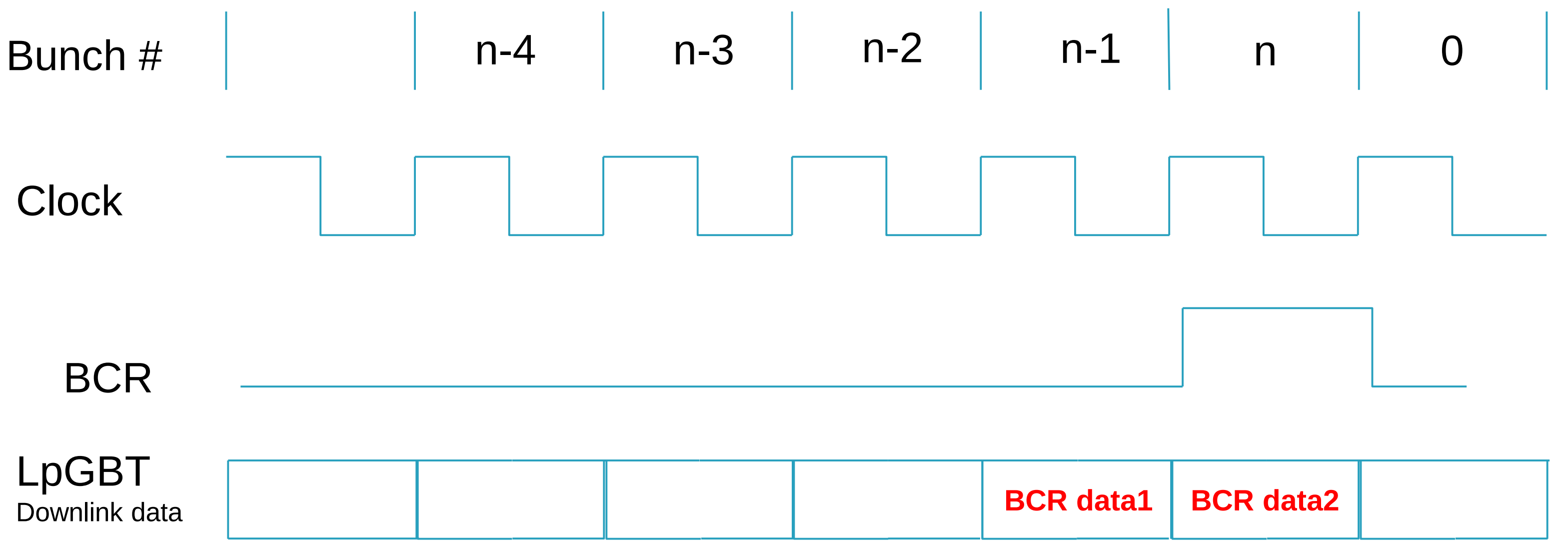
GUI to firmware interface

Dual port architecture to generate the uplink LpGBT data

It is controlled by following word:

```
//      [ 11:  0] DP_write_address
//     [ 19: 12] word_count
//     [ 21: 20] operation (0-fill_DP_w0,
//                                  1-whole downlink data operation,
//                                  2-EC,
//                                  3-IC)
//           [22] play_out_flag
//     [ 27: 23] play_count (0-always)
//
//     [ 59: 28] word1 {EC[1:0]}, IC[1:0], data[31:0]}
//     [ 99: 64] word2
//   [135:100] word3
//   [171:136] word4
//   [207:172] word5
//   [243:208] word6
//   [279:244] word7
```

BCR timing (1)


Bunch #
n-4
n-3
n-2
n-1
n
0
Clock
BCR
LpGBT
Downlink data
BCR data1
BCR data2

ATLAS EXPERIMENT

BCR timing (3)

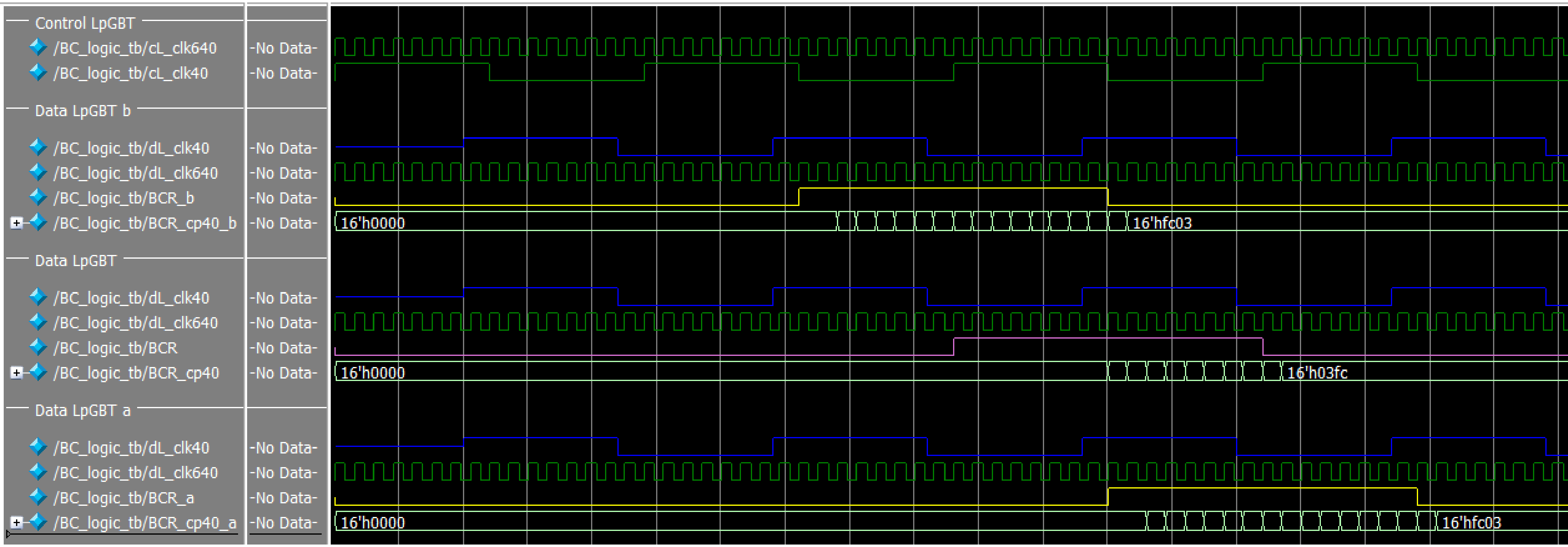
.BCR is generated for whole 25ns period
.BCR is strobed inside the ADC on leading edge of clk40 and BC is reset the next 25ns period
.3 BCR phases are possible with step of 12.5ns
.ADC will form a shift register containing data LpGBT clk40 during the active BCR
.This information will be available to the user to adjust the phase of the BCR
for each data LpGBT

```
module BC_logic
   (input              resetB,
//clocks
    input              dL_clk40,        //40MHz clock  from local data LpGBT
    input              dL_clk640,       //640MHz clock from local data LpGBT
//Bunch crossing reset
    input              BCR,             //BCR form control LpGBT
//output registers
    output reg [15:0] BCR_cp40_reg ); //data LpGBT clk40 strobed with dL_clk640
                                      //during BCR
//registers
reg         BCR_strobed, dL_clk40_del1;

//BCR strobed with data LpGBT clk640MHz
always @(posedge dL_clk640)
  begin
   BCR_strobed <= BCR;
  end
//delay the clk40 by 1 640MHz tick  of the data LpGBT
always @(posedge dL_clk640)
  begin
   dL_clk40_del1 <= dL_clk40;
  end
//form the shift register containing data LpGBT clk40
//during the active BCR
always @(posedge dL_clk640 or negedge resetB)
  begin
   if (resetB==1'b0)     BCR_cp40_reg <= 16'b0000000000000000;
   else if (BCR_strobed) BCR_cp40_reg <= {BCR_cp40_reg[14:0], dL_clk40_del1};
  end

endmodule
```

BCR timing (2)



GUI software responsibility will be to fill dual port memory to generate the BCR_data1 and BCR_data2

Control LpGBT register access (1)

The two bits IC[1:0] from subsequent frames are demultiplexed to form 8-bit words which follow a frame-based protocol. The protocol for data sent to the lpGBT for a write-read operation is shown in Table below:

ID	Description	Parity check
A	Frame delimiter 8'b 01111110	No
B	Reserved	No
C	lpGBT address (7 bits) + R/W bit = 0	No
D	Command [7:0]	Yes
E	Number of data words n[7:0]	Yes
E	Number of data words n[15:8]	Yes
F	Memory address [7:0]	Yes
F	Memory address [15:8]	Yes
G	1st data (8 bits)	Yes
G	. . .	Yes
G	nth data (8 bits)	Yes
H	Parity word (8 bits)	Yes
A	Frame delimiter 8'b 01111110	No

As only 2 IC bits per 40MHz tick are transmitted we will need many DP memory words to implement this protocol.
GUI responsibility will be to generate this protocol.

Conclusion remarks

- The Analog_testboard hardware is suitable for initial testing of FEB2_slice_board

- All important tests can be done before dedicated test equipment will arrive to Nevis

- proposed way of FEB2_slice_board testing requires a new firmware and GUI type of software to be developed

- Details of interface between the firmware a GUI software will follow soon