GBT-FPGA Tutorial

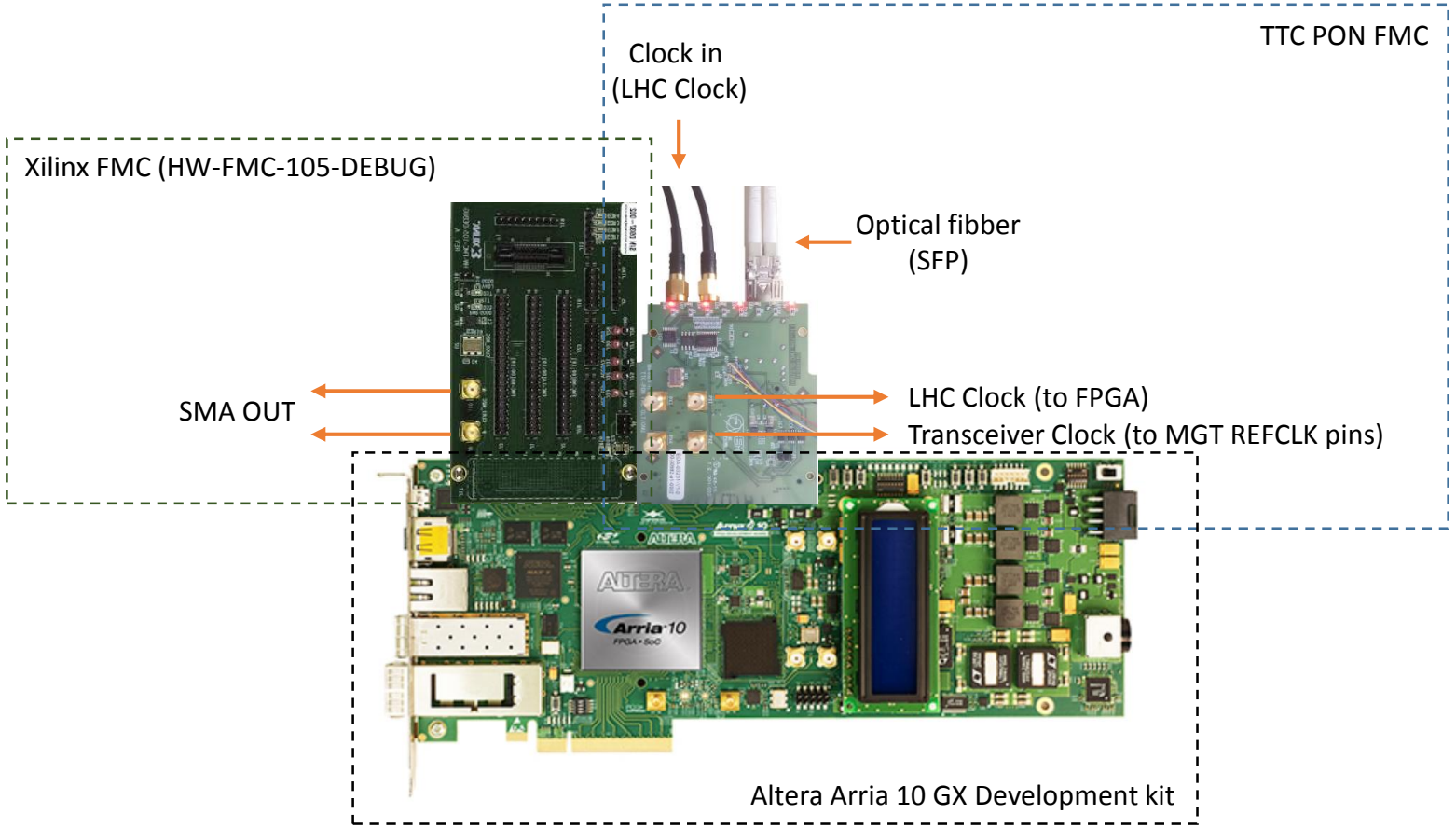
Creation of a reference design in standard mode

Outline

- Demo setup and configuration
- How to implementation the GBT-FPGA IP using records
- How to implementation the required additional modules
- How to make my design versatile (multi-links, enable/disable features)
- How to debug the design

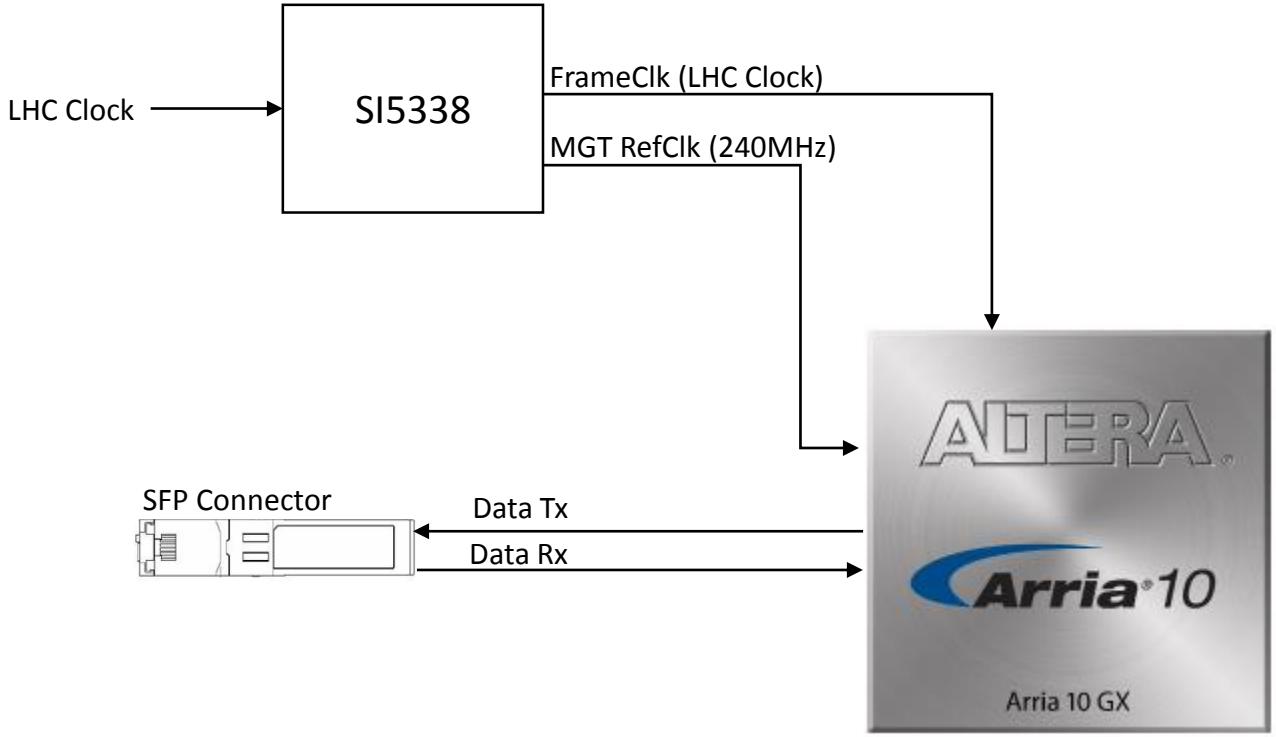
Demo setup and configuration

- What is my demo setup ?



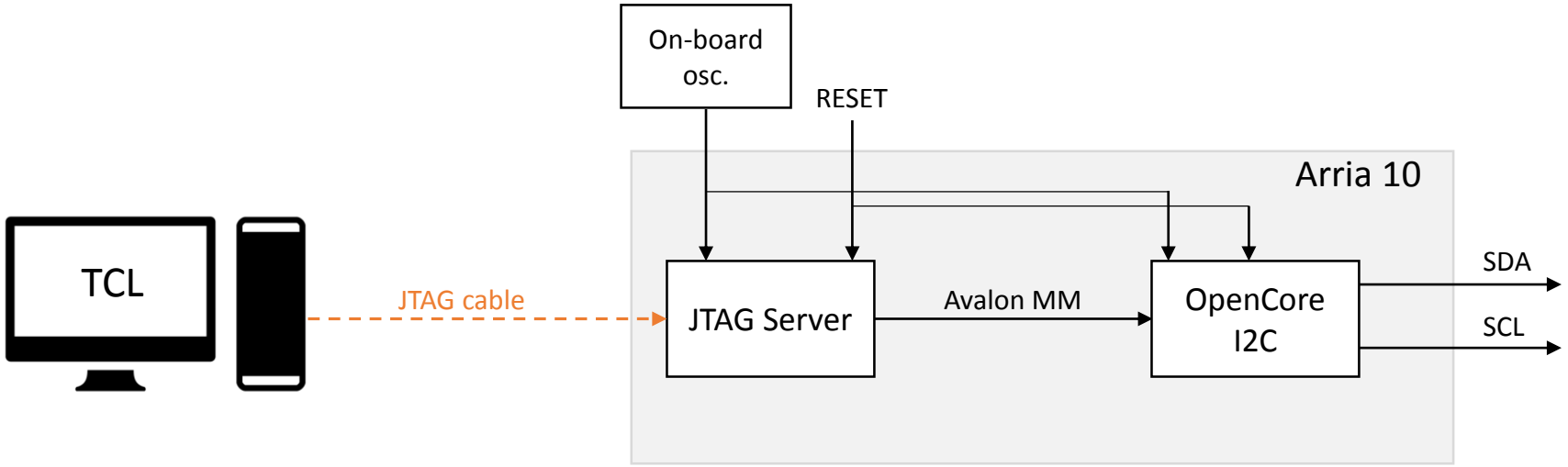
Demo setup and configuration

- What are the clock and data schemes ?



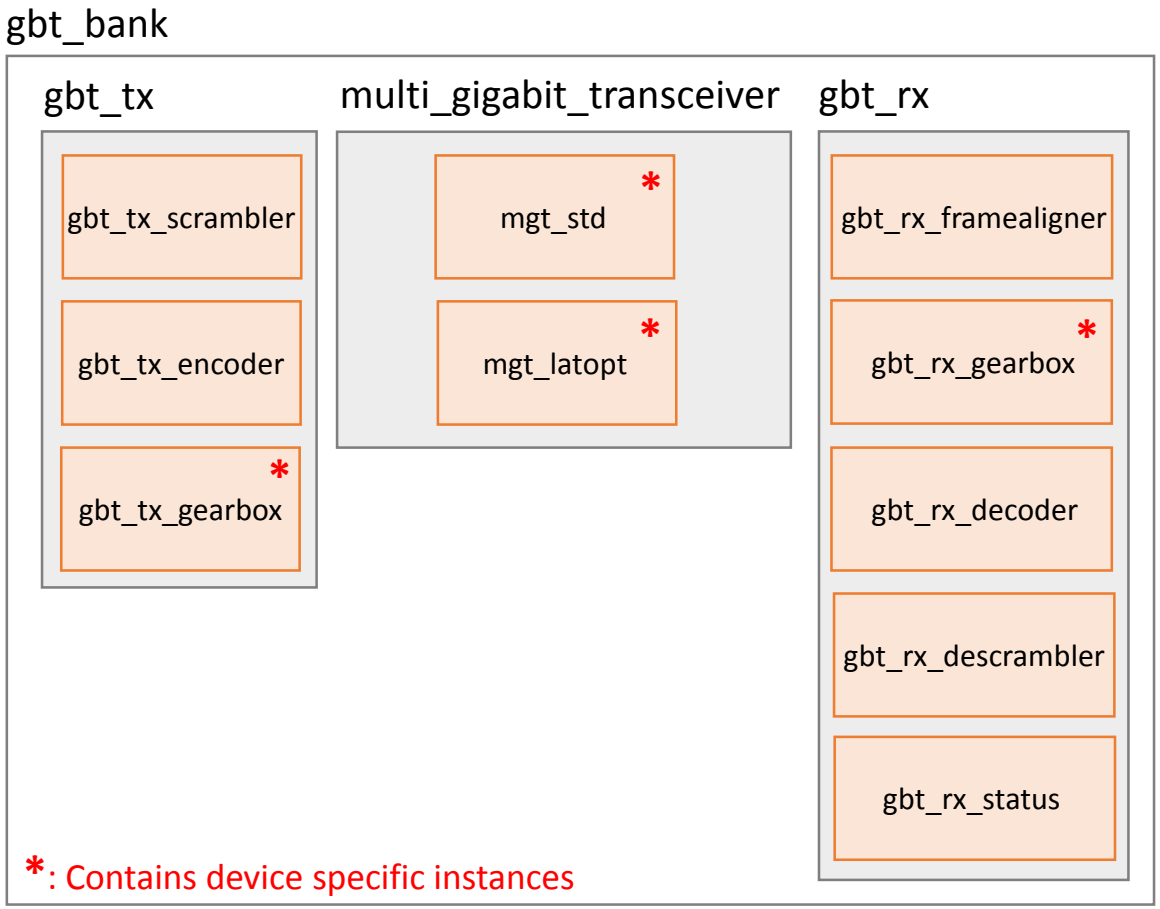
Demo setup and configuration

- How to initialize the SI5338 PLL ?
  - Using the ClockBuilder tool from SILabs
  - I2C bus located on the card
  - I2C OpenCore IP



How to implementation the GBT-FPGA IP using records

- What is the content of the GBT bank instance?



How to implementation the GBT-FPGA IP using records

- What are the GBT Bank interfaces ?

Warning: configuration can be made using two different ways ! Do not mix these to avoid bugs.

First way: Specify only the GBT_BANK_ID parameters when you implement the module. In this case, the information are get from the ***gbt_banks_user_setup*** package (gbt_bank/your_device/xxx_xxx_gbt_bank_user_setup.vhd)

Second way: Specify all the parameters **except** the GBT_BANK_ID (its value shall be 0, default value).

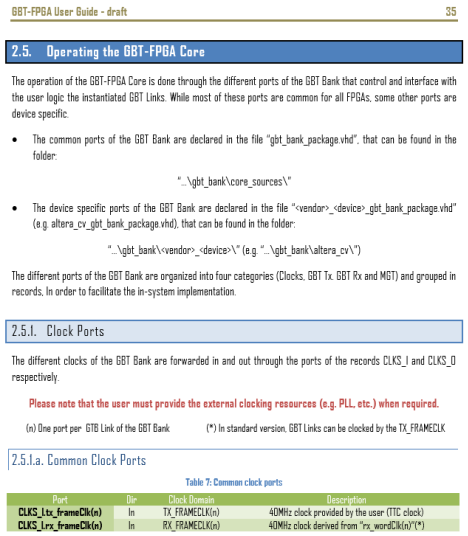
Configuration

Records

```
--=============================================================================--
--#########################################   Entity   ######################################--
--=============================================================================--

entity gbt_bank is
   generic (
      GBT_BANK_ID                             : integer := 0;
      NUM_LINKS                               : integer := 1;
      TX_OPTIMIZATION                         : integer range 0 to 1 := STANDARD;
      RX_OPTIMIZATION                         : integer range 0 to 1 := STANDARD;
      TX_ENCODING                             : integer range 0 to 1 := GBT_FRAME;
      RX_ENCODING                             : integer range 0 to 1 := GBT_FRAME
   );
   port (

      --========--
      -- Clocks --
      --========--

      CLKS_I                                  : in  gbtBankClks_i_R;
      CLKS_O                                  : out gbtBankClks_o_R;

      --========--
      -- GBT TX --
      --========--

      GBT_TX_I                                : in  gbtTx_i_R_A       (1 to NUM_LINKS);
      GBT_TX_O                                : out gbtTx_o_R_A       (1 to NUM_LINKS);

      --====================================--
      -- Multi Gigabit Transceivers (MGT) --
      --====================================--

      MGT_I                                   : in  mgt_i_R;
      MGT_O                                   : out mgt_o_R;

      --========--
      -- GBT RX --
      --========--

      GBT_RX_I                                : in  gbtRx_i_R_A       (1 to NUM_LINKS);
      GBT_RX_O                                : out gbtRx_o_R_A       (1 to NUM_LINKS)

   );
end gbt_bank;
```

How to implementation the GBT-FPGA IP using records

- How to find records definition?
  - GBT-FPGA User guide
    - 2.5. Operating the GBT-FPGA Core
  - VHDL files
    - gbt_bank/core_sources/gbt_bank_package.vhd: records generic for all the devices
    - gbt_bank/<device>/xxx_xxx_gbt_bank_package.vhd: device specific records



GBT-FPGA User Guide - draft 35

2.5. Operating the GBT-FPGA Core

The operation of the GBT-FPGA Core is done through the different ports of the GBT Bank that control and interface with the user logic the instantiated GBT Links. While most of these ports are common for all FPGAs, some other ports are device specific.

- The common ports of the GBT Bank are declared in the file "gbt_bank_package.vhd", that can be found in the folder:

  "...\gbt_bank\core_sources\"

- The device specific ports of the GBT Bank are declared in the file "<vendor>_<device>_gbt_bank_package.vhd" (e.g. altera_cv_gbt_bank_package.vhd), that can be found in the folder:

  "...\gbt_bank\<vendor>_<device>\" (e.g. "...\gbt_bank\altera_cv\")

The different ports of the GBT Bank are organized into four categories (Clocks, GBT Tx, GBT Rx and MGT) and grouped in records, In order to facilitate the in-system implementation.

2.5.1. Clock Ports

The different clocks of the GBT Bank are forwarded in and out through the ports of the records CLKS_I and CLKS_O respectively.

Please note that the user must provide the external clocking resources (e.g. PLL, etc.) when required.

(n) One port per GTB Link of the GBT Bank (*) In standard version, GBT Links can be clocked by the TX_FRAMECLK

2.5.1.a. Common Clock Ports

Table 7: Common clock ports

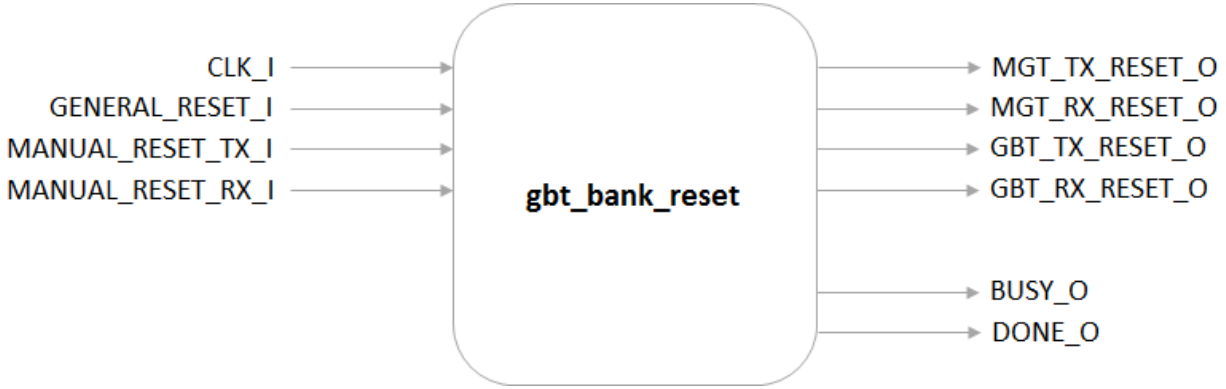
Port	Dir	Clock Domain	Description
CLKS_Ltx_frameClk(n)	In	TX_FRAMECLK(n)	40MHz clock provided by the user (TTC clock)
CLKS_Lrx_frameClk(n)	In	RX_FRAMECLK(n)	40MHz clock derived from "rx_wordClk(n)"(*)

```
--===================================================================================================--
--##################################   Package Declaration   ########################################--
--===================================================================================================--

package gbt_bank_package is

   --================================ Record Declarations ================================--

   --================--
   -- Clocks scheme --
   --================--

   type gbtBankClks_i_R is
   record
      tx_frameClk                               : std_logic_vector(1 to MAX_NUM_GBT_LINK);
      rx_frameClk                               : std_logic_vector(1 to MAX_NUM_GBT_LINK);
      ------------------------------------------
      mgt_clks                                  : gbtBankMgtClks_i_R;
   end record;

   type gbtBankClks_o_R is
   record
      mgt_clks                                  : gbtBankMgtClks_o_R;
   end record;

   --=========--
   -- GBT Tx --
   --=========--

   type gbtTx_i_R is
   record
      reset                                     : std_logic;
      ------------------------------------------
      isDataSel                                 : std_logic;
      ------------------------------------------
      data                                      : std_logic_vector(83 downto 0);
      extraData_wideBus                         : std_logic_vector(31 downto 0);
   end record;
```

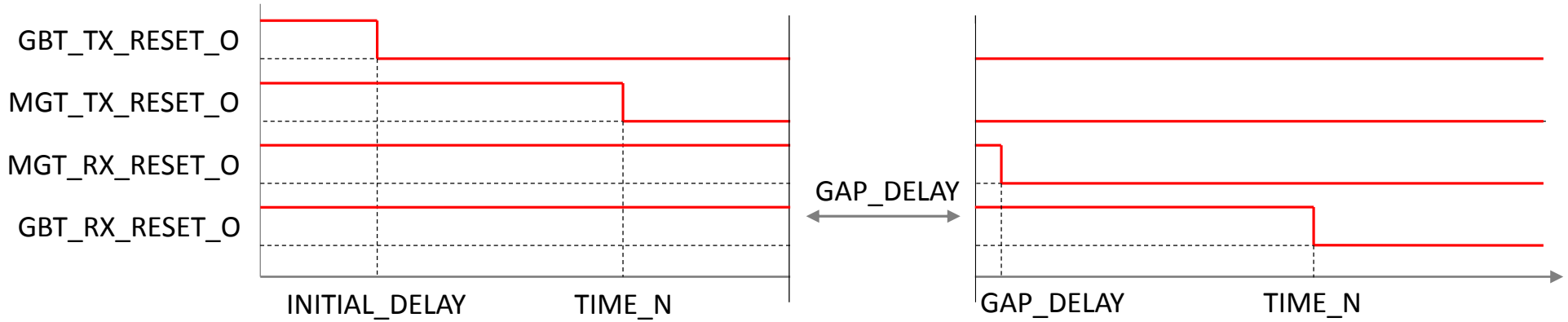
DEMO

How to implementation the required additional modules

- GBTBank reset



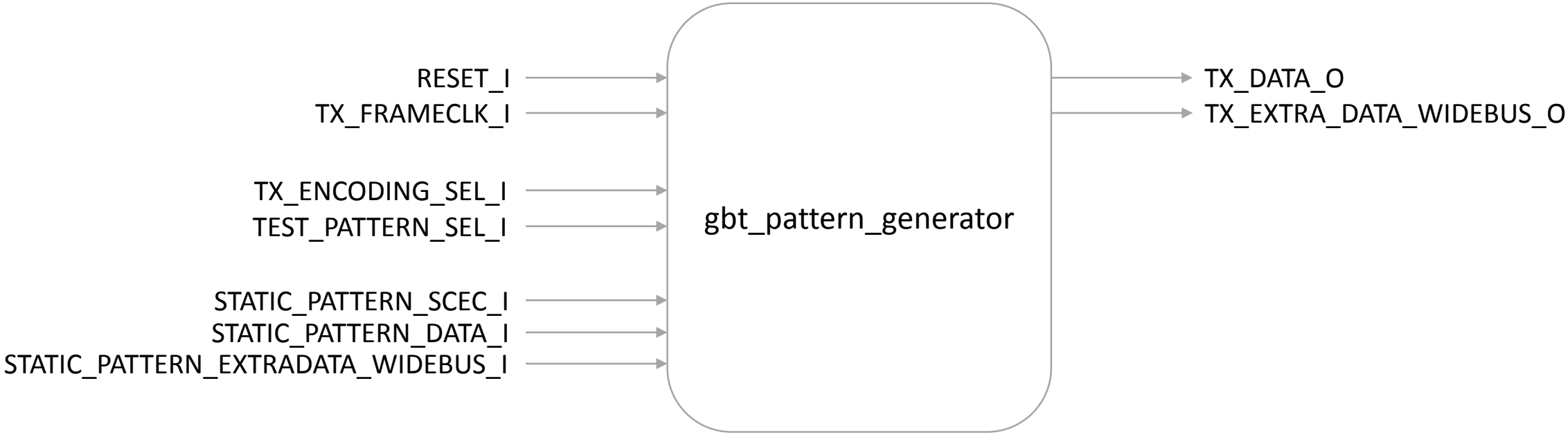
- Generates the reset signal for the different part of the GBT Bank: TX, RX and MGT

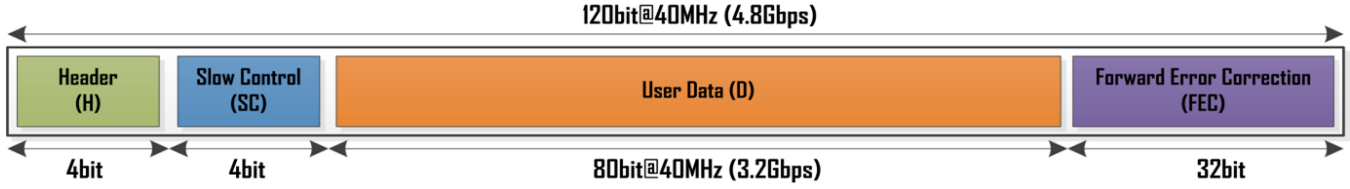


DEMO

How to implementation the required additional modules

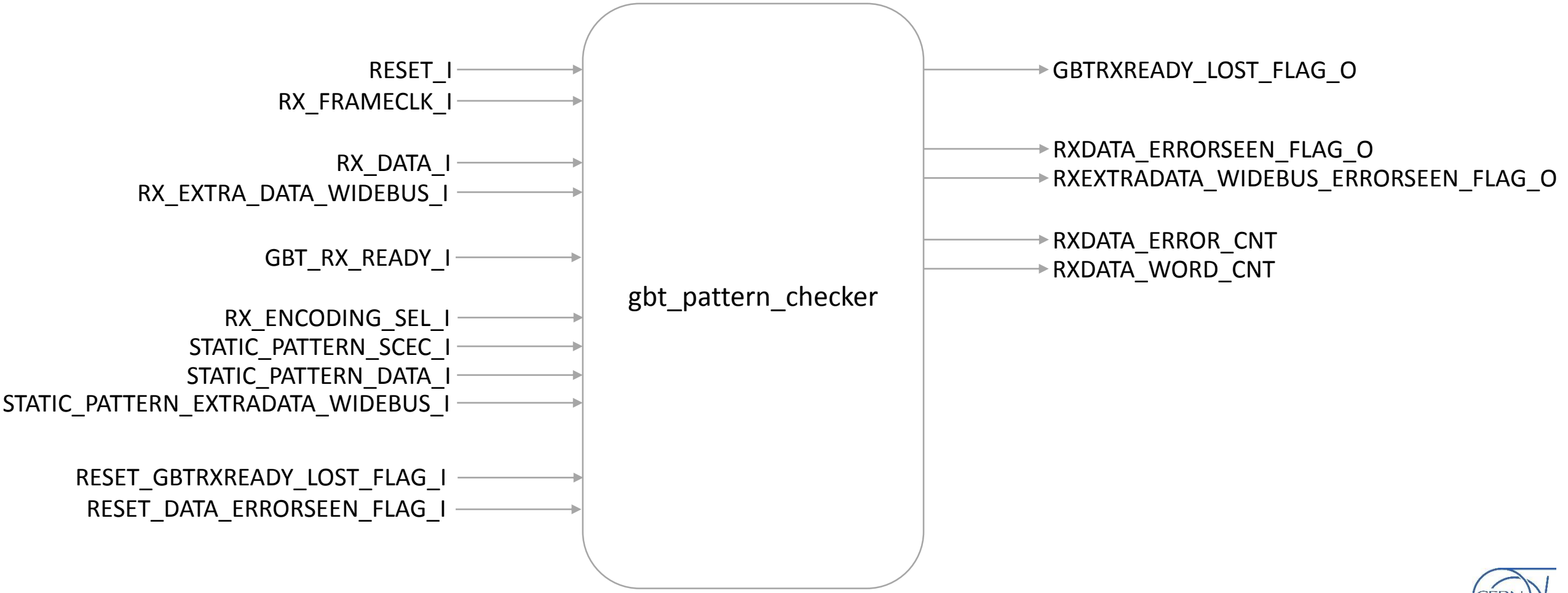
- GBT Data pattern generator





How to implementation the required additional modules

- GBT Data pattern checker



DEMO

How to make my design versatile (multi-links, enable/disable features)

- Generate statement for multi-links:

```
forloop_ex_gen: for i in 1 to NUM_LINKS generate
   -- Code generated n times
end generate;
```

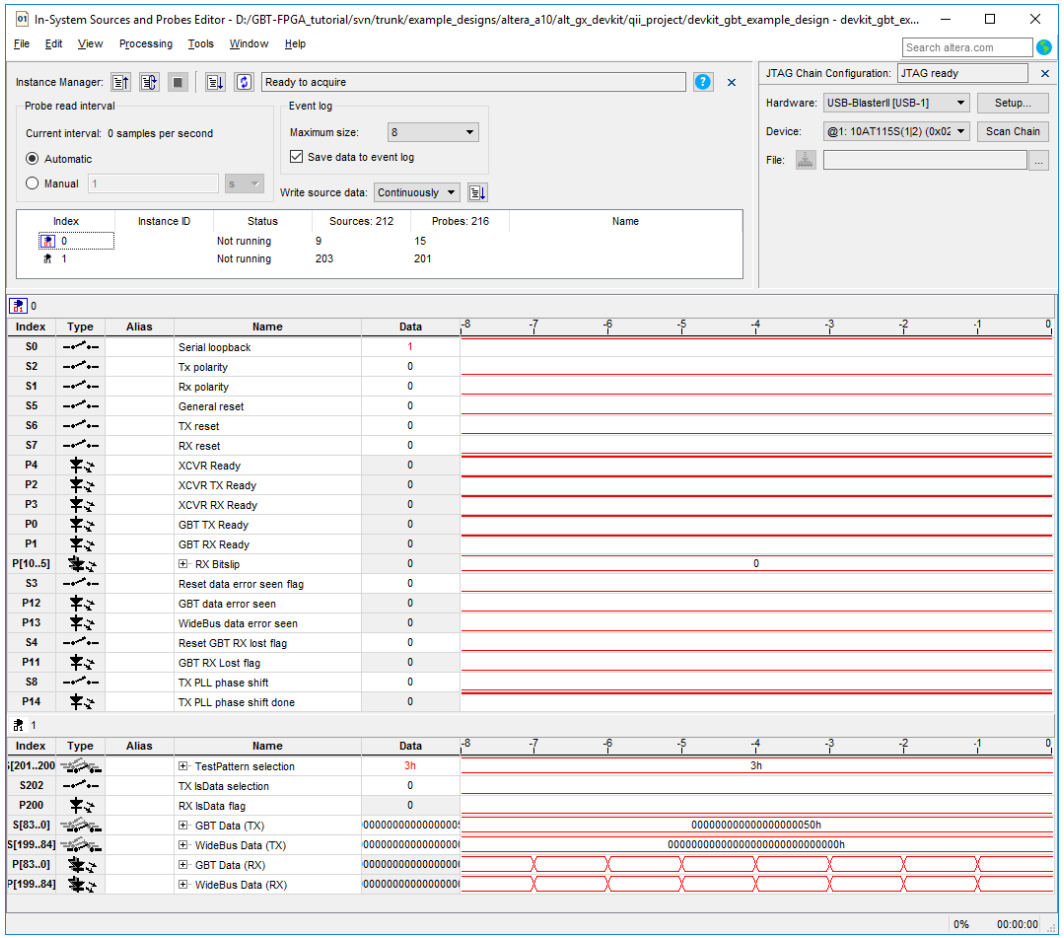
- Generate statement to enable/disable features:

```
ifgen_ex_gen: if PARAMETER_NAME = VAL_PARAM generate
   -- Code generated if statement is true
end generate;
```

DEMO

How to debug the design

- ISSP and Signal tap



DEMO