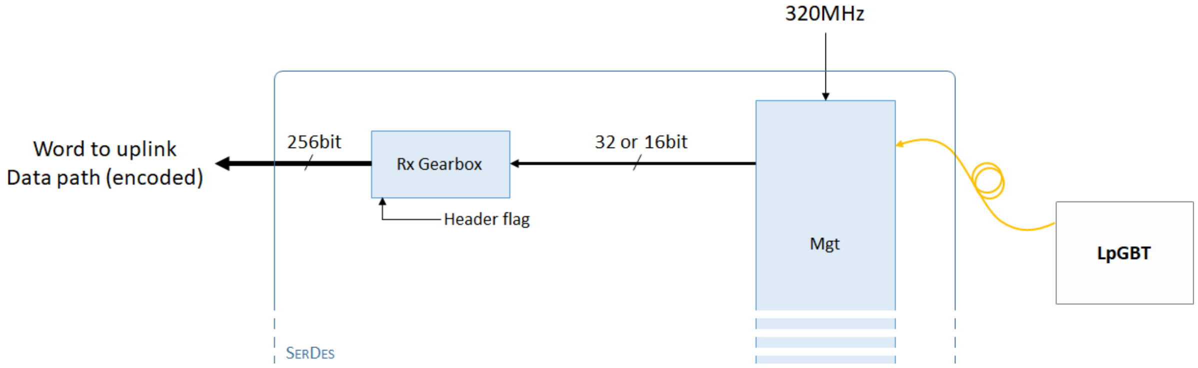
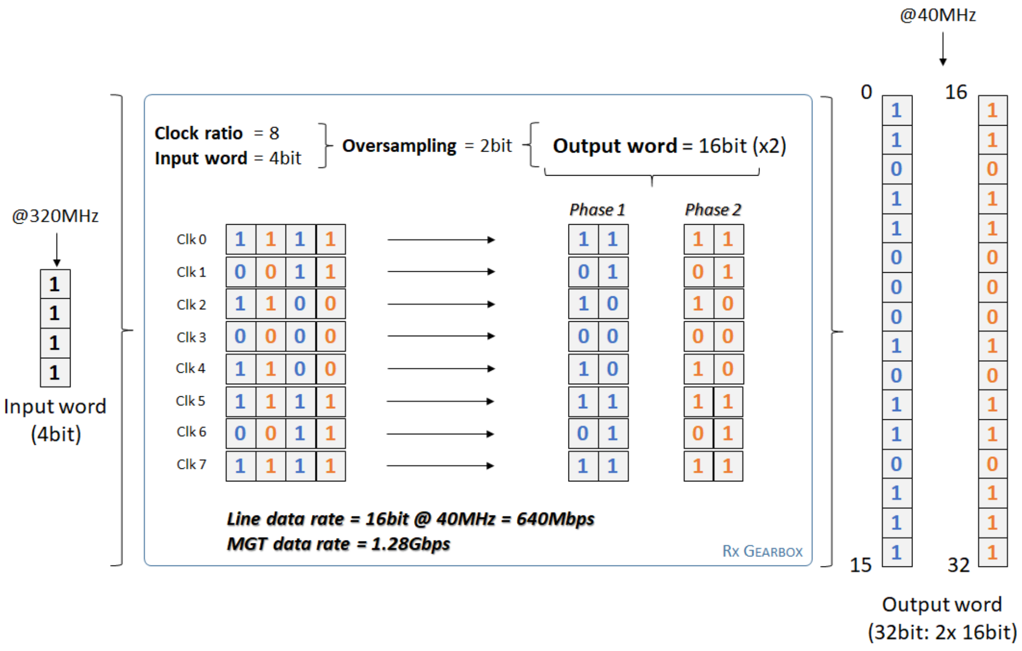
LpGBT-FPGA: Use of the Rx Gearbox module for the uplink path.

This reference note focus on the Rx gearbox implementation to ensure the uplink communication (from front-end FPGA to back-end LpGBT) . According to the ASIC specification, the uplink data rate can be either 5.12Gbps or 10.24Gbps. Therefore, the FPGA counterpart is ready to work with both speed, implying the possibility to configure the MGT to work with both line rate. However, depending on the selected speed, the bus size of the transceiver output changes and, by consequence, the configuration of the Rx gearbox differs depending on the configuration. Additionally, dealing with oversampling could be required when the transceiver is configured to work at 10.24Gbps and the link is configured at 5.12Gbps (dynamic mode). This document presents how to configure the provided module and how to integrate it to design the LpGBT-FPGA top level to match the needs.



Oversampling

In receiver mode, the oversampling feature consists in getting only one bit over N to emulate a lower data rate. The following example represent a simplified view of the dynamic configuration. When the transceiver is configured to work at 1.28Gbps, but the transmitter send data at 0.640Gbps, each bit is deserialized twice by the MGT. In that case, we have 2 words, where each one is specific to one phase of the deserializer.



					Phase 1		Phase 2	
Clk 0	1	1	1	1	1	1	1	1
Clk 1	0	0	1	1	0	1	0	1
Clk 2	1	1	0	0	1	0	1	0
Clk 3	0	0	0	0	0	0	0	0
Clk 4	1	1	0	0	1	0	1	0
Clk 5	1	1	1	1	1	1	1	1
Clk 6	0	0	1	1	0	1	0	1
Clk 7	1	1	1	1	1	1	1	1

Input word (4bit)
1
1
1
1

0–15	16–32
1	1
1	1
0	0
1	1
1	1
0	0
0	0
0	0
1	1
0	0
1	1
1	1
0	0
1	1
1	1
1	1

Note: Oversampling ratio is computed using **OSratio = (Clock ratio * InWidth)/ OutWidth**. Therefore, all of the parameters, as well as the equation result, shall be integers.

Instantiation

The Rx gearbox is made of a unique VHDL module configurable using generic parameters. Doxygen documentation of this module can be found on doxygen here. The code below shows how to instantiate the module for either 5.12Gbps, 10.24Gbps or dynamic data rate configurations.

1. Static 5.12Gbps configuration (MGT word width: 16bit – data rate: 5.12Gbps)

```
rxGearbox_5g12_inst: rxGearbox
    generic map(
      c_clockRatio          => 8,              -- 320MHz / 40MHz = 8
      c_inputWidth          => 16,             -- size of the MGT word
      c_outputWidth         => 128             -- Size of the payload data
    )
    port map(
      -- Clock and reset
      clk_inClk_i           => clk_mgtRxUsrclk_s,              -- In clock (MGT)
      clk_clkEn_i           => sta_headerFlag_s,               -- In clock en.
      rst_gearbox_i         => rst_rxgearbox_s,                -- Reset

      -- Data
      dat_inFrame_i         => uplnkFrame_from_mgt_s,          -- In frame (MGT word)
      dat_outFrame_o        => upLinkFrame_5g12_s,             -- Out frame (128b)

      -- Status
      sta_gbRdy_o           => sta_rxGbRdy_s                   -- Ready status
    );
```

Note: Because the uplink datapath input expects a 256bit word, the 128b data bus shall be duplicated as following: upLinkFrame_from_rxgb_s <= upLinkFrame_5g12_s & upLinkFrame_5g12_s;

2. Static 10.24Gbps configuration (MGT word width: 32bit – data rate: 10.24Gbps)

```
rxGearbox_10g24_inst: rxGearbox
    generic map(
      c_clockRatio          => 8,                  -- 320MHz / 40MHz = 8
      c_inputWidth          => 32,                 -- Size of the MGT word
      c_outputWidth         => 256                 -- Size of the payload data
    )
    port map(
      -- Clock and reset
      clk_inClk_i           => clk_mgtRxUsrclk_s,              -- In clock (MGT)
      clk_clkEn_i           => sta_headerFlag_s,               -- In clock en.
      rst_gearbox_i         => rst_rxgearbox_s,                -- Reset

      -- Data
      dat_inFrame_i         => uplnkFrame_from_mgt_s,          -- In frame (MGT word)
      dat_outFrame_o        => upLinkFrame_from_rxgb_s,        -- Out frame (128b)

      -- Status
      sta_gbRdy_o           => sta_rxGbRdy_s                   -- Ready status
    );
```

3. Dynamic configuration (MGT word width: 32bit – data rate: 10.24Gbps)

```
-- rxGearbox_5g12Dyn_inst  output is used when 5.12Gbps is dynamically selected
rxGearbox_5g12Dyn_inst: rxGearbox
    generic map(
      c_clockRatio          => 8,          -- 320MHz / 40MHz = 8
      c_inputWidth          => 32,         -- Size of the MGT word
      c_outputWidth         => 128         -- Size of the payload data
    )
    port map(
      -- Clock and reset
      clk_inClk_i           => clk_mgtRxUsrclk_s,              -- In clock (MGT)
      clk_clkEn_i           => sta_headerFlag_s,               -- In clock en.
      rst_gearbox_i         => rst_rxgearbox_s,                -- Reset

      -- Data
      dat_inFrame_i         => uplnkFrame_from_mgt_s,          -- In frame (MGT word)
      dat_outFrame_o        => upLinkFrame_from_rxgb_5g12_s,   -- Out (256b <- 128 x OSratio)

      -- Status
      sta_gbRdy_o           => sta_rxGbRdy_5g12_s              -- Ready status
    );

-- rxGearbox_10g24Dyn_inst  output is used when 10.24Gbps is dynamically selected
rxGearbox_10g24Dyn_inst: rxGearbox
    generic map(
      c_clockRatio          => 8,         -- 320MHz / 40MHz = 8
      c_inputWidth          => 32,        -- Size of the MGT word
      c_outputWidth         => 256        -- Size of the payload data
    )
    port map(
      -- Clock and reset
      clk_inClk_i           => clk_mgtRxUsrclk_s,                    -- In clock (MGT)
      clk_clkEn_i           => sta_headerFlag_s,                     -- In clock en.
      rst_gearbox_i         => rst_rxgearbox_s,                      -- Reset

      -- Data
      dat_inFrame_i         => uplnkFrame_from_mgt_s,                -- In frame (MGT word)
      dat_outFrame_o        => upLinkFrame_from_rxgb_10g24_s,        -- Out frame (256b)

      -- Status
      sta_gbRdy_o           => sta_rxGbRdy_10g24_s                   -- Ready status
    );

-- Dynamic selection:
upLinkFrame_from_rxgb_s <= upLinkFrame_from_rxgb_10g24_s when selDataRate _i = '1' else
                           upLinkFrame_from_rxgb_5g12_s;

sta_rxGbRdy_s <= sta_rxGbRdy_10g24_s  when selDataRate_i = '1' else sta_rxGbRdy_5g12_s;
```