7. Electrical links

Warning

Known issues: Section 19.1.11, Section 19.1.8, Section 19.1.13,

Section 19.1.12.

The lpGBT can be electrically interfaced with the on detector electronics using different topologies. These depend on the ePort bandwidths as well as on the uplink data rate (5.12 or 10.24 Gbps). Depending on the configuration, the lpGTB can interface simultaneously with up to 28 frontend devices for uplink transmission and up to 16 devices for the downlink. The electrical connections between the lpGBT and the frontend devices are called **elinks**. eLinks are used not only to transmit data between the the lpGBT and the frontend devices but also for the clocks. eLinks use the CERN Low Power signalling (**CLPS**), please see Section 7.5 for the further details.

More specifically, eLinks are used for:

- The differential clock lines (**ECLK[28:0]P** / **ECLK[28:0]N**): The clock lines are driven by lpGBT to the frontend modules;
- The differential downlink data outputs (**EDOUT[3:0][3:0]P** / **EDOUT[3:0][3:0]N** and **EDOUTECP** / **EDOUTECN**): The output data lines are driven by the lpGBT to the frontend devices;
- The differential uplink data inputs (**EDIN[6:0][3:0]P** / **EIN[6:0][3:0]N** and **EDINECP** / **EDINECN**): The input data lines are driven by the frontend devices to the lpGBT.

Fig. 7.1 represents a generic interconnection topology between the lpGBT chip and the frontend electronics using eLinks.

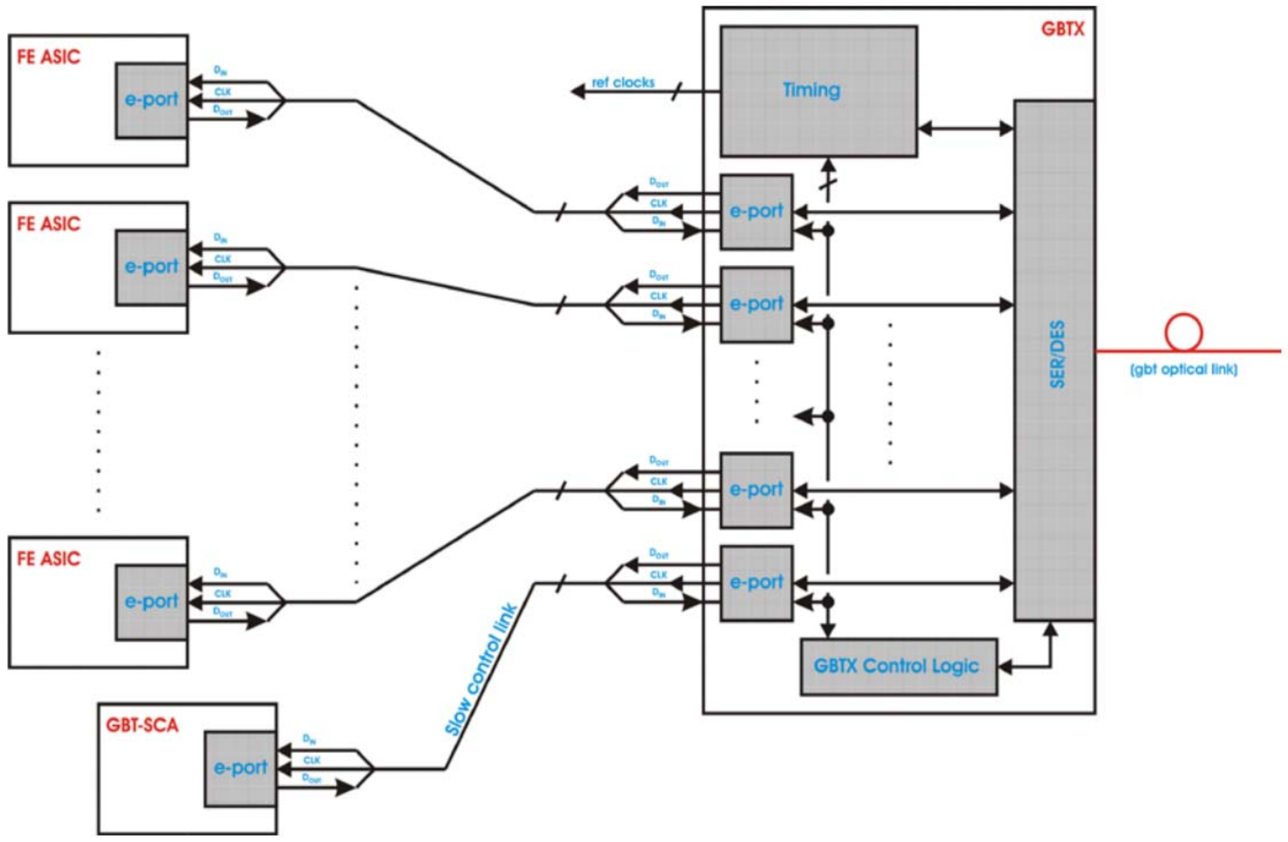

Fig. 7.1 eLink connection topology

To be noticed however that, since the number of input and output ePorts present in the lpGBT is different, the case depicted above, where each frontend device is served by an equal number of data input and output lines, is not always feasible. This is a consequence of the asymmetrical data bandwidth requirements of the detectors which is reflected in the lpGBT chip architecture: resulting on the eLink data rates and number of available eLinks being different for up and downlinks.

7.1. eLink Groups

The eLinks are subdivided in groups of 4 and each group contains for channels (or eLinks). The data rate of each group can be set independently as detailed in in Table ePortRx (uplink) data rates and Table ePortTx (downlink) data rates. As it can be seen in the table ePortRx (uplink) data rates, the uplink data speed depends on the lpGBT high speed link bit rate. Naturally, there is no relation between up and down eLink data rates since the High-Speed uplink and downlink bandwidths are different. The number of active eLinks within a group depends on the group data rate. For each case those tables indicates which eLink channels are active within a group. The bit shift in/out order for the eLink data inputs and outputs is MSB first.

Table 7.1 ePortRx (uplink) data rates

TxDataRate	Data Rate Select	Data Rate	Links per group	Active Channels
5.12 Gb/s	Off	0 Mb/s	0	none
5.12 Gb/s	x4	160 Mb/s	4	0, 1, 2 and 3
5.12 Gb/s	x8	320 Mb/s	2	0 and 2
5.12 Gb/s	x16	640 Mb/s	1	0
10.24 Gb/s	Off	0 Mb/s	0	none
10.24 Gb/s	x4	320 Mb/s	4	0, 1, 2 and 3
10.24 Gb/s	x8	640 Mb/s	2	0 and 2
10.24 Gb/s	x16	1.28 Gb/s	1	0

Table 7.2 ePortTx (downlink) data rates

Data Rate Select	Data Rate	Links per group	Active Channels
Off	0 Mb/s	0	none
x2	80 Mb/s	4	0, 1, 2 and 3
x4	160 Mb/s	2	0 and 2
x8	320 Mb/s	1	0

7.2. eLink pin naming conventions.

The naming conventions adopted for the eLink pins allows to easily identify to each group and "channel" a pin belongs to.

eLink inputs, **EDINGCP** (EDIN[6:0][3:0]P / EDIN[6:0][3:0]N)

- **E**: eLink;
- **D**: data;
- **IN**: uplink input;

- **G**: group number, 0 to 6 (there are 7 groups);
- **C**: channel number, 0 to 3 (there are 4 eLinks associated with every group);
- **P**: polarity, P for the positive polarity pin and N for the negative.

eLink outputs, **EDOUTCP** (EDOUT[3:0][3:0]P / EDOUT[3:0][3:0]N)

- **E**: eLink;
- **D**: data;
- **OUT**: downlink output;
- **G**: group number, 0 to 3 (there are 4 groups);
- **C**: channel number, 0 to 3 (there are 4 eLinks associated with every group);
- **P**: polarity, P for the positive polarity pin and N for the negative.

eClock **ECLKCP** (ECLK[28:0]P / ECLK[28:0]N)

- **E**: eLink;
- **CLK**: clock output;
- **C**: channel number, 0 to 28 (there are 29 eClocks)
- **P**: polarity, P for the positive polarity pin and N for the negative.

As an example, the pin EDIN32N is an eLink data input of group 3, channel 2 and it is the negative polarity pin.

7.3. eLink Tx Mirror function

The downlink eLinks implement a "mirror" function in which the data in one ePortTx channel, in a given group, is copied into one or more outputs in the same group. The mirror function can be useful to implement broadcast of data to the frontends, providing several copies of the same data in two or more outputs. It can also be used as a way to facilitate routing on the PCB allowing to have multiple choice of pins for the same data: 2 outputs at 160 Mbps and 4 at 320 Mbps. The availability of the function and how many channels can be used depends on the group data rate. The possibilities are no mirroring at 80 Mbps, two outputs per channel at 160 Mbps and four outputs per channel at 320 MHz. More specifically:

- **80 Mbps**: No mirroring;
- **160 Mbps**: Channel 3 repeats the data of channel 2 and channel 1 repeats the data of channel 0;
- **320 Mbps**: Channels 3, 2 and 1 repeat the data of channel 0;

The mirror function is controlled on a group basis by signals **EPTX[G]MirrorEnable** in register [0x0a8] EPTXControl (Bits 3:0).

7.4. eLink Clocks

Besides the up and downlink data links, the lpGBT chip features up to 28 independent clock outputs. The output frequency is user programmable and it is decoupled from the up/downlink data rates. The available clock frequencies are presented in Table ePortClock clock frequencies. Each clock can be individually inverted (phase shifted by 180 degrees).

Table 7.3 ePortClock clock frequencies

Clk Freq Select	Clock Frequency
Off	0 MHz
x1	40 MHz
x2	80 MHz
x4	160 MHz
x8	320 MHz
x16	640 MHz
x32	1.28 GHz

Warning: Known issues: Section 19.1.8, Section 19.1.13.

7.5. CERN Low Power signalling (CLPS)

Interconnections between the lpGBT and the frontend devices (eLinks) is made through differential cables or transmission lines and the signalling adopted is defined by an ad-hoc "standard" called the **CERN Low Power signalling (CLPS)**. Its main characteristics are:

- Link types:
  - Point-to-point;
  - Multi-drop transmitter.
- Maximum data rate:

    - 1.28 Gbps (NRZ signalling).
- Maximum clock frequency:
    - 1.28 GHz.
- Programmable signalling level:
    - 100 mV to 400 mV (single-ended amplitude);
    - 200 mV to 800 mV (differential amplitude).
- Common mode voltage:
    - 600 mV (nominal, for 1.2 V supply voltage).
- Load impedance:
    - 100 Ohm differential.

Fig. 7.2 clarifies the definitions of single and differential amplitudes. Please note that the amplitude range is specified for a 100 Ohm differential termination impedance. Reducing the termination value will reduce the transmitter signal swing while increasing it will increase the signal swing. Although the transmitter (eTx) termination can be in principle different from 100 Ohm, the user must be aware that the termination impedance should match the impedance of the cable/transmission line being used. The lpGBT receivers (eRx) incorporate a 100 Ohm termination impedance. If the user intends to use a different line impedance he/she should disable the internal termination and provide an on PCB termination of appropriate value.

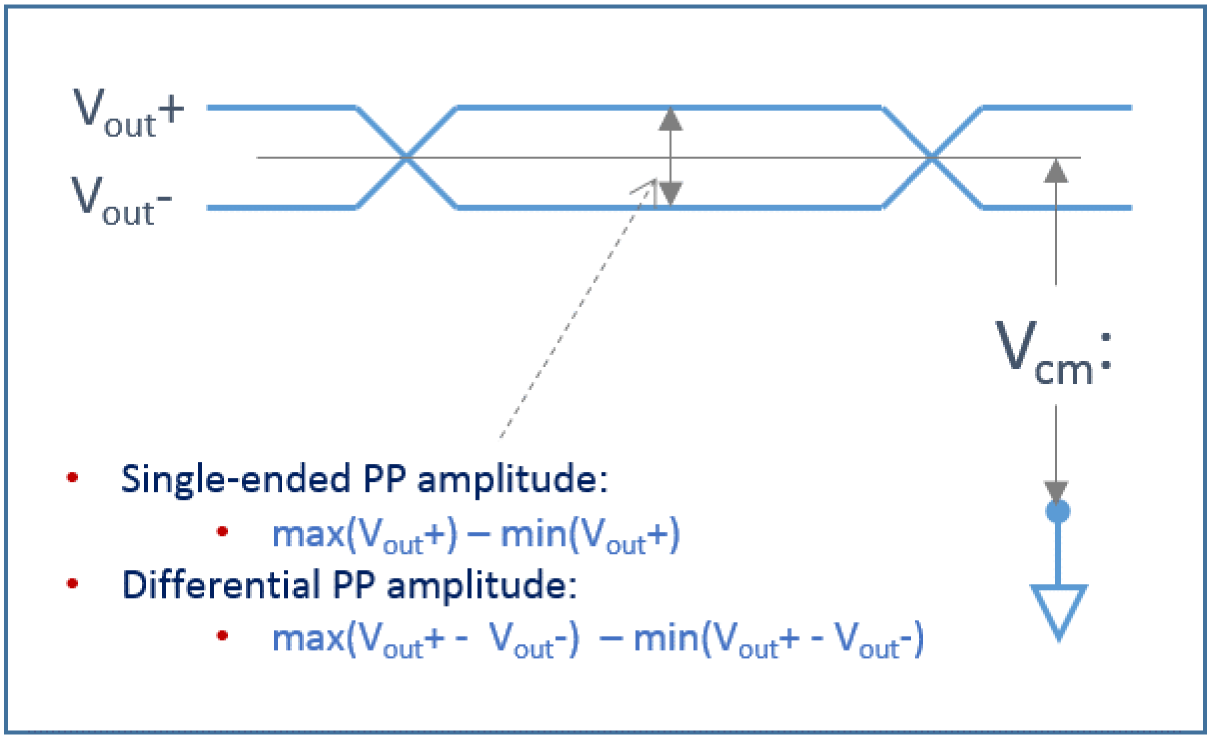
$V_{out}+$
$V_{out}-$
V_{cm}:
Single-ended PP amplitude:
$\max(V_{out}+) - \min(V_{out}+)$
Differential PP amplitude:
$\max(V_{out}+ - V_{out}-) - \min(V_{out}+ - V_{out}-)$

Fig. 7.2 CLPS single-ended and differential amplitude definitions

The lpGBT eLinks do not support multiple talkers driving the same line. However, for the downlinks it is possible to use a multiple drop bus configuration where multiple listeners are connected on the same line. This is illustrated in Fig. 7.3. Such configuration allows to transmit the same data from an lpGBT eLink port to several frontend devices (broadcasting). The lpGBT has no provision to address the listeners individually but it is perfectly feasible for the user protocol (transmitted over an eLink) to allow addressing of the listeners individually. If a multiple drop configuration is used the following should be observed:

- A termination impedance should be present at the end of the line;
- Only the last receiver (eRx) in the transmission line should have the termination impedance enabled.
- Star configurations are discouraged. They are however possible if a termination impedance is present at each branch. This will however reduce the signal amplitude since the equivalent impedance seen by the driver is reduced.

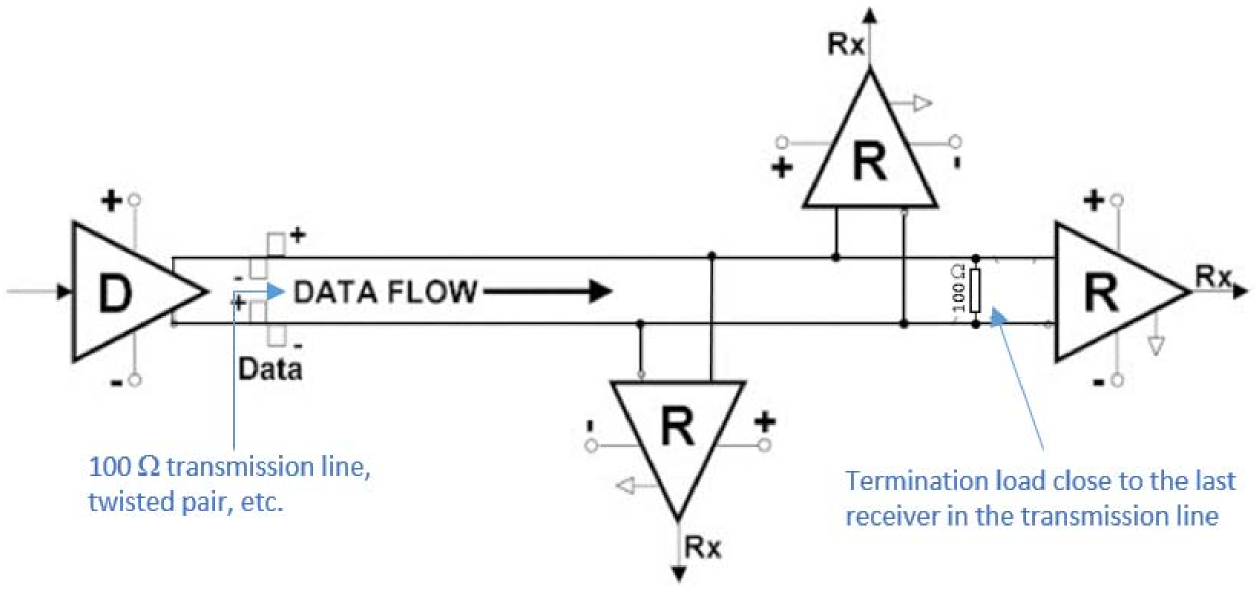
D
DATA FLOW
Data
R
Rx
100 Ω
100 Ω transmission line, twisted pair, etc.
Termination load close to the last receiver in the transmission line

Fig. 7.3 eLink (downlink) multi-drop configuration

7.6. eLink Receivers (eRx)

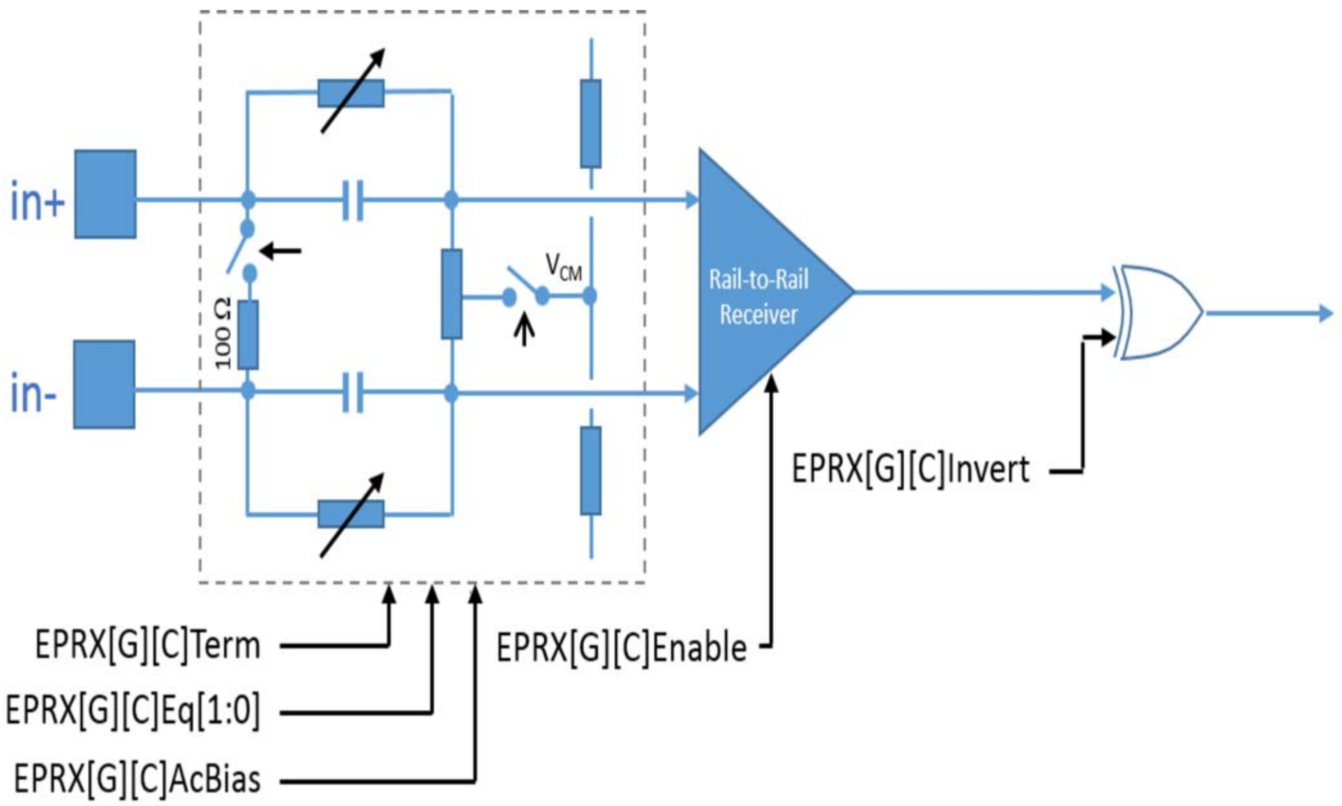

Fig. 7.4 eLink receiver

Fig. 7.4 represents the architecture of the eLink Receiver (eRx). The eRx is designed to receive the CLPS signals described in this chapter. Its main features are:

- **Power OFF/ON** controlled by the signals **EPRX[G][C]Enable**;
- **Signal polarity: non-inverted/inverted** controlled by the signals **EPRX[G][C]Invert**;
- **Termination Disabled/Enabled** controlled by signals **EPRX[G][C]Term**;
- **AC bias OFF/ON** controlled by the signals **EPRX[G][C]AcBias**;
- **Programmable Equalization** Controlled by the signals **EPRX[G][C]Eq[1:0]**. This signals control the position of the zero and the amount of peaking produced by the eRx equalizer function as given in table eRx equalizer zero position;

Table 7.4 eRx equalizer zero position

EPRX[G][C]Eq[1:0]	Zero Frequency	Peaking
2'b00	No equalization	0 dB
2'b01	281 MHz	4.9 dB

EPRX[G][C]Eq[1:0]	Zero Frequency	Peaking
2'b10	122 MHz	7.7 dB
2'b11	67 MHz	10.7 dB

The naming convention of the signals is such that **[G]** represents the **Group number** and **[C]** the **Channel number**.

Enabling of a receiver is also conditioned by the group data rate and thus the signals **EPRX[G]DataRate[1:0]**. Consequently only channels that are compatible with the data rate selected for a given group can be enabled.

All the registers controlling the eLink Receivers can be found in section ePortRx.

- Signals **EPRX[G][C]Enable** and **EPRX[G]DataRate[1:0]** are associated with registers [0x0c4] EPRX0Control to [0x0ca] EPRX6Control and, for the EC channel, [0x0e8] EPRXEcChnCntr. Note that the data rate is not programmable for the EC channel;
- Signals **EPRX[G][C]Invert**, **EPRX[G][C]AcBias**, **EPRX[G][C]Term** and **EPRX[G][C]Eq[1]** are associated with registers [0x0cc] EPRX00ChnCntr to [0x0e7] EPRX63ChnCntr and, for the EC channel, [0x0e8] EPRXEcChnCntr;
- Signals **EPRX[G][C]Eq[0]** are associated with registers [0x0cf] EPRX03ChnCntr to [0x0e7] EPRX63ChnCntr and, for the EC channel, [0x0e8] EPRXEcChnCntr.

7.7. eLink Drivers (eTx)

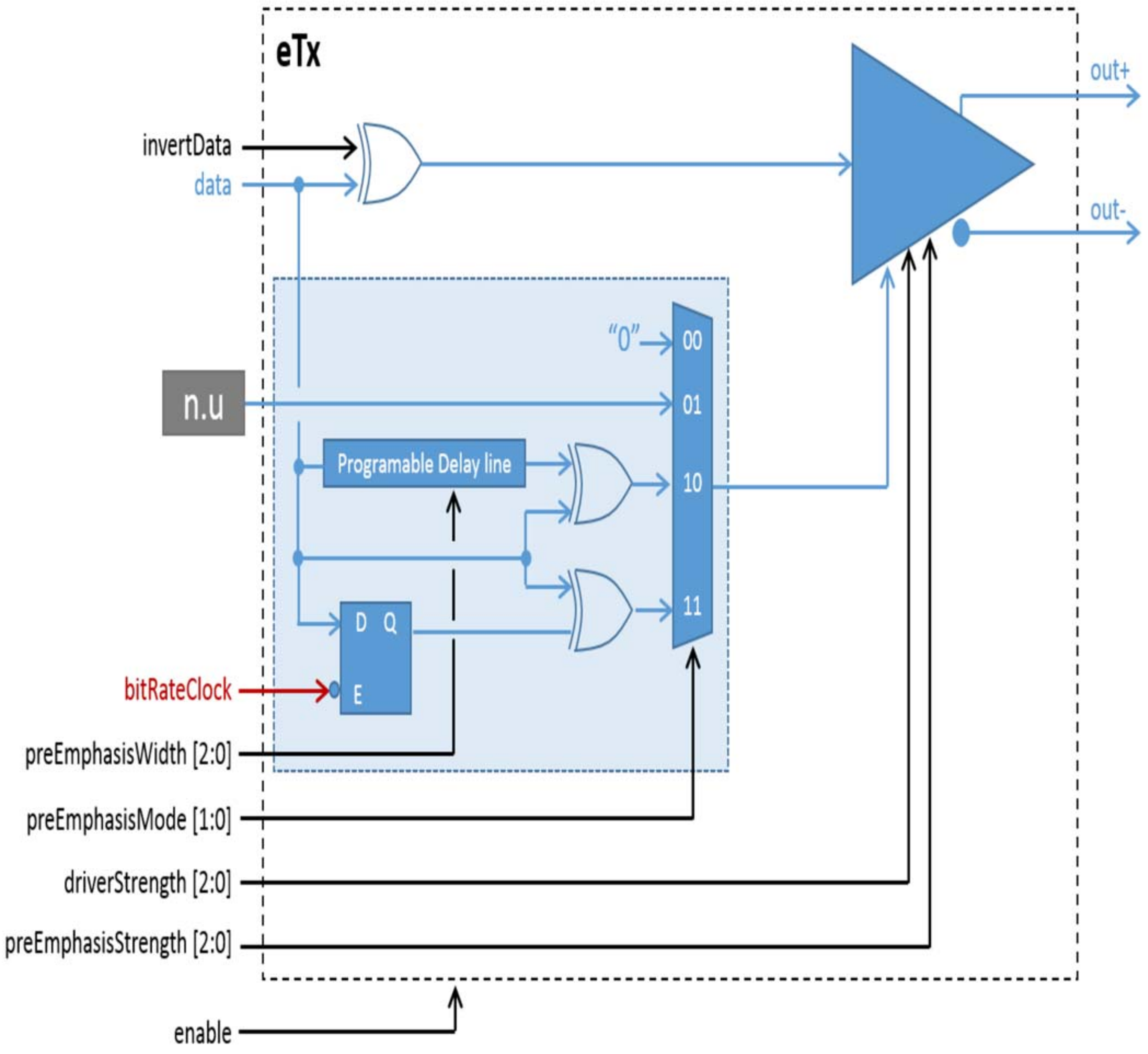

Fig. 7.5 eLink receiver

Fig. 7.5 illustrates the architecture of the lpGBT eLink driver. The same driver is used for the data outputs, up to 320 Mbps, and the clock outputs, up to 1.28 GHz. The driver has been designed to drive 100 Ohm loads with programmable strengths and controlled amounts of pre-emphasis. It has been designed to satisfy the CLPS standard previously described in this chapter. The driver offers many programmable features which include:

- **Power OFF/ON**: This feature allows to save power when a given channel is not in use even if the the group that it belongs to is active:

    - *Enabling a clock driver*: In this case the signal "enable" in Fig. 7.5 becomes: enable = (EPCLK[G]Freq[2:0] !~ 2'b00). That is, every time a clock output is set to work at any of the frequencies between 40 MHz and 1.28 GHz the corresponding driver is enabled. The clock frequency is controlled by bits 2:0 of registers [0x06c] EPCLK0ChnCntrH to [0x0a4] EPCLK28ChnCntrH;
    - *Enabling a data driver*: In this case it is possible that even if a given group is active that a specific channel within that group will not be used. Consequently it is necessary to explicitly enable the channels in use. Here, the signal "enable" in Fig. 7.5 becomes: enable = EPTX[G][C]Enable. These signals are controlled by registers [0x0a9] EPTX10Enable and [0x0aa] EPTX32Enable; Notice that, if the mirror function is not enabled, for a given data rate only the channels indicated in table ePortTx (downlink) data rates can be enabled. If the mirror function is enabled for a given group it is possible to enable all the channels in that group. The mirror function is controlled on a group basis by signals EPTX[G]MirrorEnable in register [0x0a8] EPTXControl (Bits 3:0).
- **Driving current**: Programmable between 1 to 4 mA in steps of 0.5 mA. For an 100 Ohm termination this results in a differential amplitude of 200 mV to 800 mV Peak-to-Peak (signal "driverStrength[2:0]", see details below);
    - *Clock driver strength*: For clock drivers the driving strength is set by the signal "driverStrength[2:0]" in Fig. 7.5 with driverStrength[2:0] = EPCLK[C]DriveStrength[2:0]. These signals are controlled by bits 5:3 of registers [0x06c] EPCLK0ChnCntrH to [0x0a4] EPCLK28ChnCntrH;
    - *Data driver strength*: For data drivers the driving strength is set by the signal "driverStrength[2:0]" in Fig. 7.5 with driverStrength[2:0] = EPTX[G][0]DriveStrength[2:0]. These signals are controlled by bits 2:0 of registers [0x0ac] EPTX00ChnCntr to [0x0bb] EPTX33ChnCntr. For the EC channel driverStrength[2:0] = EPTXEcDriveStrength[2:0], these signals are controlled by bits 2:0 of registers [0x0ab] EPTXEcChnCntr;
- **Programmable pre-emphasis**: To facilitate building systems that use relatively low bandwidth interconnects between the lpGBT and the frontend devices, the eTx provides a pre-emphasis function which is both programmable in driving strength and pulse width. The driving strength of the pre-emphasis pulse is programmable between 1 to 4 mA in steps of 0.5 mA (signal "preEmphasisStrength[2:0]", see details below) and its width can be programmed between 120 ps and 960 ps in steps of 120 ps or to be exactly half of the period of the selected bit rate (signals "preEmphasisMode[1:0]" and "preEmphasisWidth[2:0]", see details below).
    - *Clock driver pre-emphasis strength*: For clock drivers the driving strength is set by the signal "preEmphasisStrength[2:0]" in Fig. 7.5 with preEmphasisStrength[2:0] = EPCLK[C]PreEmphasisStrength[2:0]. These signals are controlled by bits 7:5 of registers [0x06d] EPCLK0ChnCntrL to [0x0a5] EPCLK28ChnCntrL.

    - *Clock driver pre-emphasis timing*: The pre-emphasis pulse width is controlled by two parameters (Fig. 7.5) the timing **Mode** and the **Pulse Length**. The "Mode" allows to select between no pre-emphasis, pre-emphasis with "clock timed" pulse duration (1/4 of the clock period) or "self timed" where the pulse with can be programmed from 120 ps up to 960 ps. Notice that the "clock timed" mode is not valid if the clock output is set to 1.28 GHz and that in the "self timed" mode the pulse width should never be programmed to exceed 1/2 clock period. The pre-emphasis mode is selected by the signals preEmphasisMode[1:0] = EPCLK[C]PreEmphasisMode[2:0] that are controlled by bits 4:3 of registers [0x06d] EPCLK0ChnCntrL to [0x0a5] EPCLK28ChnCntrL. When in the "self timed" mode, the signals preEmphasisWidth[2:0] = EPCLK[C]PreEmpahasisWidth[2:0] set the pulse width and are controlled by bits 2:0 of registers [0x06d] EPCLK0ChnCntrL to [0x0a5] EPCLK28ChnCntrL.
    - *Data driver pre-emphasis strength*: For data drivers the driving strength is set by the signal "preEmphasisStrength[2:0]" in Fig. 7.5 with preEmphasisStrength[2:0] = EPTX[G][C]PreEmphasisStrength[2:0]. These signals are controlled by bits 7:5 of registers [0x0ac] EPTX00ChnCntr to [0x0bb] EPTX33ChnCntr. For the EC channel preEmphasisStrength[2:0] = EPTXEcDriveStrength[2:0], these signals are controlled by bits 7:5 of registers [0x0ab] EPTXEcChnCntr;
    - *Data driver pre-emphasis timing*: The pre-emphasis pulse width is controlled by two parameters (Fig. 7.5) the timing **Mode** and the **Pulse Length**. The "Mode" allows to select between no pre-emphasis, pre-emphasis with "clock timed" pulse duration (1/2 of the clock period) or "self timed" where the pulse with can be programmed from 120 ps up to 960 ps. Notice in the "self timed" mode the pulse width should never be programmed to exceed the bit period. The pre-emphasis mode is selected by the signals preEmphasisMode[1:0] = EPTX[G][C]PreEmphasisMode[1:0] that are controlled by bits 4:3 of registers [0x0ac] EPTX00ChnCntr to [0x0bb] EPTX33ChnCntr. When in the "self timed" mode, the signals preEmphasisWidth[2:0] = EPTX[G][C]PreEmpahasisWidth[2:0] set the pulse width and are controlled by bits 6:4 and 2:0 of registers [0x0bc] EPTX01_00ChnCntr to [0x0c3] EPTX33_32ChnCntr. For the EC channel the pre-emphasis mode is selected by the signals preEmphasisMode[1:0] = EPTXEcPreEmphasisMode[1:0] that are controlled by bits 4:3 of register [0x0ab] EPTXEcChnCntr. When in the "self timed" mode, the signals preEmphasisWidth[2:0] = EPTXEcPreEmphasisWidth[2:1] set the pulse width and are controlled by bits 6:4 of register :ref:`REG_EPTXCONTROL.
- **Signal polarity: non-inverted/inverted**: the data or clock outputs polarity is programmable (signal "invertData", see details below).
    - *Clocks polarity* is controlled by the signals invertData = EPCLK[C]Invert in registers [0x06c] EPCLK0ChnCntrH to [0x0a4] EPCLK28ChnCntrH;

- *Data polarity* is controlled by the signals invertData = EPTX[G][C]Invert in registers [0x0bc] EPTX01_00ChnCntr to [0x0c3] EPTX33_32ChnCntr and for the EC port invertData = EPTXEcInvert in register [0x0a8] EPTXControl;

7.8. Phase alignment

Phase delays between the lpGBT and the frontend electronics will depend on the system configuration, local cable lengths and delays in the frontend circuits. It is thus necessary that the eLink input ports provide a means of adjusting the phases of the incoming data signals so that data is sampled reliably in the middle of the eye-opening. A phase adjustment/alignment mechanism is thus necessary in the lpGBT data input, and in some cases also in the ePorts of the frontend ASICs.

7.8.1. Downlink phase alignment

Two possibilities are foreseen for the eLinks in the down direction from the lpGBT to the ePort of the frontend ASICs. In one case both data and clock are simultaneously transmitted to the frontend ASIC and in the other only data is transmitted without a dedicated eLink clock. As the lpGBT is unaware of the code/frame/data structure carried by the eLinks, it does not contribute to the data down phase alignment and synchronization mechanism.

When both data (EDOUT[G][C]P/EDOUT[G][C]N) and clock (ECLK[C]P/ECLK[C]N) lines are routed to the frontend ASIC (Fig. 7.1) they must follow the same electrical route and have the same loading to assure that their relative phase is maintained at the arrival to the frontend ASIC.

In case no clock lines (ECLK[C]P/ECLK[C]N) are used and only the data lines (EDOUT[G][C]P/EDOUT[G][C]N) are routed to the frontend ASIC (to save PCB resources) the eLink clock needs to be recovered locally in the frontend ASIC with a Clock and Data Recovery (CDR) circuit. To assure that such a local CDR circuit can reliably re-generate the link clock, the data must be appropriately encoded with sufficient data transitions (and normally also DC balanced to enable AC coupling of the data signals). The lpGBT does not have built in dedicated logic for such a line encoding as it can be done in the counting room (the lpGBT is fully data transparent). Suggested line codes for this are scrambling, which incurs no bandwidth penalty and 7B/8B encoding with a code efficiency of 87.5%. The 7B/8B code has built-in comma characters that can be used for frame delimiting and synchronization. Other codes are possible but they must ensure a sufficiently high density of transitions for clock recovery.

7.8.2. Uplink phase alignment

Phase delays between the lpGBT and the frontend electronics will depend on the system configuration, local cable lengths and delays in the frontend circuits. It is thus necessary for the eLink ports to provide a means of adjusting the phases of the incoming data signals so that data is sampled reliably in the middle of the eye-opening. A phase adjustment/alignment mechanism is thus necessary in the lpGBT data inputs, and in some cases also in the ePorts of the frontend ASICs. For the uplinks the phase of the incoming data to the lpGBT is unknown. However, the data rate is known from the lpGBT and frontend modules configuration. The lpGBT clocks are synchronous with the frontend module clocks with a fixed and stable phase relationship. It is thus unnecessary to recover the clock from the data but it is necessary to phase align the incoming EDIN[6:0][3:0]P / EDIN[6:0][3:0]N data with the internal clocks in the lpGTB for each eLink. A dedicated phase-aligner circuit is responsible for this for each eLink.

The phase aligner circuit ensures that the eLink data received by the lpGTB is sampled by the lpGBT internal clock in the middle of the eye-diagram. The block diagram of the circuit is shown in Fig. 7.6.

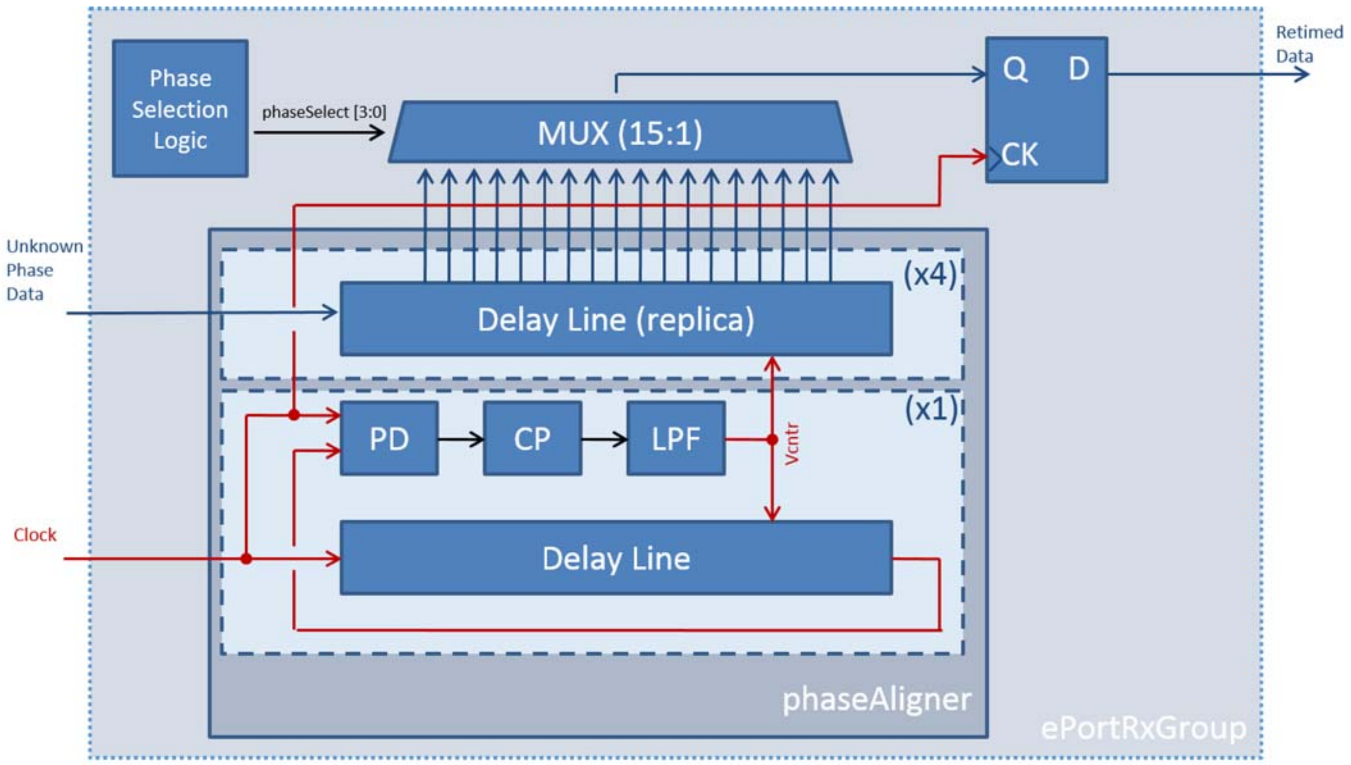

Fig. 7.6 ePortRx Phase Aligner

A phase-aligner per eGroup (ePortRx) is available with 4 phase adjustable channels as all eLinks in a group work at the same data rate, but may have different phases.

The phase aligner is composed of a master Delay Locked Loop (DLL) and four replica delay-lines with programmable phase taps (see Fig. 7.6). Each replica delay-line can adjust the phase of the incoming data via the delay taps along the delay line. The phase aligner channels can operate in four different modes: **Fixed phase**, **Initial training**, **Continuous phase tracking** and **Continuous phase tracking with initial phase**. The selection of the phase tracking mode is done on a group basis by signals EPRX[G]TrackMode[1:0]. These signals are controlled by bits 1:0 of registers [0x0c4] EPRX0Control to [0x0ca] EPRX6Control and [0x0cb] EPRXEcControl; The meaning and setup of the phase tracking modes is as follows:

- **Fixed phase**: (*Static phase selection*) In this mode, the channel phase is set by the user to a fixed value and remains static during operation (unless explicitly changed by the user). This allows the eLink ports to accept data without any DC balance restrictions. It is particularly useful when the data being transmitted to the lpGBT over the ePorts has relatively large amounts of jitter. It is the user responsibility however, to set the phase to a value that optimizes the sampling of the incoming data at the center of the eye diagram to minimize the bit error rate. The channel phase is set either at system initialization via the configuration stored in the eFuses or later via a dedicated ePort command through the I2C or IC channel or EC channel in Simplex Tx. The phase selection is made by signals EPRX[G][C]PhaseSelect[3:0] that are controlled by bits 7:4 of registers [0x0cc] EPRX00ChnCntr to [0x0e7] EPRX63ChnCntr and, in the case of the EC channel, by signals EPRXECPhaseSelect[3:0] controlled by bits 7:4 of register [0x0e8] EPRXEcChnCntr;
- **Initial training**: (*Initial training with learned static phase selection*). This phase tracking mode allows the lpGBT to determine what is the best phase selection for each channel and to switch to a static phase selection for the remaining of the operation. In some sense is very similar to the previous mode freeing the user from the task of finding the optimum phase value. Initially the user has to set the phase-aligner in the phase learning mode and ensure that DC balanced data is sent through the ePort channel being **trained**. After the phase-aligner has found the optimum phase value for the specified channel, the user sets the phase-aligner to hold the phase value. The user can then resume normal data transmission through the eLinks without concerns of DC balance on the data being transmitted. The phase settings found can be read through the I2C or IC channel or EC channel in Simplex Tx. This mode requires however the intervention of the user to set the learning phase, to set the frontend module to send balanced data, to set the hold phase-settings operation and then to program the frontend modules to resume normal data transmission. The most likely use of this mode

is during the initial-phases of system development to determine the best phase settings that can later be used regularly with the static phase selection mode.

  - **Setting the training phase**: The user must explicitly set the channels in a group to be in the training phase. The signals that control phase training are EPRX[G]Train[3:0] where the n-th bit controls the n-th channel within the group and EPRXEcTrain for the EC channel. Channels not specifically enabled will be ignored during the training phase. The registers that control those signals are [0x105] EPRXTrain10 to [0x107] EPRXTrain54 and [0x108] EPRXTrainEc6;
  - **Monitor the training process**: After enabling the training phase the user must monitor the status of the phase locking process. For that, the signals EPRX[G]ChnLocked[3:0] (with the n-th bit corresponding to the n-th channel) available in registers [0x142] EPRX0Locked to [0x154] EPRX6Locked (known issue: missing register for the EC channel!) must be read through one of the ASIC control interfaces. For each channel, when the phase-locking state-machine finds the optimum phase value it asserts the corresponding channel status bit to "one"; Furthermore, the state of the locking state machines for each channel can be read from the same registers.
  - **Exiting the training phase**: Once all channels will be locked the user should disable the training mode by clearing bits EPRX[G]Train[3:0] and EPRXEcTrain in registers [0x105] EPRXTrain10 to [0x107] EPRXTrain54 and [0x108] EPRXTrainEc6; The user can read the phase values determined by the phase-locking state-machine from registers [0x143] EPRX0CurrentPhase10 to [0x156] EPRX6CurrentPhase32 and [0x157] EPRXEcCurrentPhase.

- **Continuous phase tracking**: (*Automatic phase tracking*) This phase tracking mode allows setting the optimum phase and dynamically adjusting it free of the user intervention. For systems where the incoming data phase is expected to slowly vary this mode allows to track the variations without introduction of bit errors during the process. The actual phase of the received data is estimated from the data bit transitions and used to dynamically adjust the phase alignment. For this to work relatively frequent data transitions are necessary. In this case a DC balanced code is desirable however it is strictly not necessary since the phase-aligner waits for sufficient data transitions in a given ePort channel before it decides to adjust the phase setting. It is possible to access the phase value selected for each channel by reading the registers [0x143] EPRX0CurrentPhase10 to [0x156] EPRX6CurrentPhase32 and [0x157] EPRXEcCurrentPhase; The lock state of the channels and the state of the locking state machines can be read from registers [0x142] EPRX0Locked to [0x154] EPRX6Locked (known issue: missing register for the EC channel!)
- **Continuous phase tracking with initial phase**: (*Automatic phase tracking with initial phase*) This phase tracking mode is virtually identical to the previous one however it allows to start the phase tracking process form a preset phase value for each channel.

The motivation for this mode is the following: for channels where the phase selection has to be set at the beginning or at the end of the delay line, it is possible (due for example to jitter) that for each system initialization the phase is sometimes (randomly) selected at beginning of the line and sometimes at the end. Although this behavior is correct, due to the periodicity of the bit period, it will result as an apparent non-fixed latency for those channels. This mode avoids this ambiguity by forcing a starting phase but retains all the advantages and flexibility of the *continuous phase tracking* mode. The preset phase values needed are read from registers [0x105] EPRXTrain10 to [0x107] EPRXTrain54 and [0x108] EPRXTrainEc6 at beginning of operation. They are either set at system initialization via the configuration stored in the eFuses or later via a dedicated ePort command through the I2C or IC channel or EC channel in Simplex Tx. As for the *continuous phase tracking* mode, the lock state of the channels and the state of the locking state machines can be read from registers [0x142] EPRX0Locked to [0x154] EPRX6Locked (known issue: missing register for the EC channel!)

For all the tracking modes, except the *fixed phase*, for each ePort group the phase alignment is made in a **circular** fashion skipping channels that are not being used. It is very important that the user will disable the channels that are unused in order to save power. In the dynamic modes, the algorithm used to step the phase up or down is based on an average of 8 samples and phase-changes are done incrementally in steps of ±T/8, where T is the bit period. The phase-changes are done in such a way that no data transmission errors are introduced when that phase is being changed on the fly. The locking lock state machine counts the number of transitions that fall in the expected region. If 64 transitions are detected, then the channel is declared as locked. Conversely, the channel is considered unlocked if 64 transitions fall outside the expected range.

Warning: Known issues: Section 19.1.11, Section 19.1.12.

7.8.3. ePortRx group DLL programming

The operation of the phase-aligners depends on the delay lines being calibrated in relation to the bit period. This is the function of the DLL in each of the ePortRx groups. All the ePortRx DLL's share the same parameters. Programming of the phase-aligners' DLLs is made through register [0x034] EPRXDllConfig. In there signals:

- **EPRXDllCurrent[1:0]** (bits 7:6) control the charge-pump current;
- **EPRXDLLConfirmCount[1:0]** (bits 5:4). During DLL locking it is possible that, due to jitter, the DLL phase detector will give an inconsistent phase information. Since the starting point ("short" delay line) is forced at the beginning of operation. This count sets the

minimum number of times the "late" information has to be reported by the phase-detector before the control is passed on to the DLL control loop;
- **EPRXDLLFSMClkAlwaysOn** (bit 3) This signal disables / enables clock gating of the DLL initialization state machine.
- **EPRXDLLCoarseLockDetection** (bit 2). This signal disables/enables coarse (less strict) detection of lock;
- **EPRXEnableReInit** (bit 1) disables /enables the re-initialization of an ePortRxGroup when the phase-aligner state machine finds the phase-selection to be out of range;
- **EPRXDataGatingEnable** (bit 0) disables/enables data gating.

The status of the phase-aligner DLLs' can be read from registers [0x158] EPRX0DllStatus to [0x15e] EPRX6DllStatus. This registers report:

- The lock lock status of the **DLL EPRX[G]DllLocked** (bit 7);
- The state of lock filter state machine **EPRX[G]DllLFState[1:0]** (bits 1:0);
- The loss of Lock counter value **EPRX[G]DllLOLCnt[4:0]**.

7.8.4. Note about DC balancing and data/clock encoding for eLinks

For some applications it might be necessary to AC couple the eLink connections (e.g. serial powering of frontends). In this case a DC-Balanced code must be transmitted over each data line. If the eClocks are not used on the eLinks, Clock and Data Recovery (CDR) is required in the frontend ASIC. This encoding/decoding must take place at the optical link source in the counting room, where flexible FPGA based link interfaces are used, and in the frontend itself. The encoding overhead will depended on the type of encoding used. As the lpGBT is fully transparent to the user data being transferred it is not directly involved in any line coding being used on the local eLinks.

7.9. eClocks Wrap-up

eClocks are available for any of the operation modes of the lpGBT (Simplex RX, Simplex TX and Transceiver). To use the eClocks the user must:

- Enable the eClock drivers channels to be used: clock drivers are automatically enabled if the channel clock frequency is set to be non-zero;

- Select the channel clock frequency (see registers [0x06c] EPCLK0ChnCntrH to [0x0a4] EPCLK28ChnCntrH);
- Select the channel clock polarity (see registers [0x06c] EPCLK0ChnCntrH to [0x0a4] EPCLK28ChnCntrH);
- Select the channel driving strength (see registers registers [0x06c] EPCLK0ChnCntrH to [0x0a4] EPCLK28ChnCntrH);
- Select the channel pre-emphasis mode (see registers [0x06d] EPCLK0ChnCntrL to [0x0a5] EPCLK28ChnCntrL);
- If pre-emphasis is used, select the channel pre-emphasis driving strength (see registers [0x06d] EPCLK0ChnCntrL to [0x0a5] EPCLK28ChnCntrL);
- If the self-timed pre-emphasis mode is used select channel the pre-emphasis pulse width (see registers registers [0x06d] EPCLK0ChnCntrL to [0x0a5] EPCLK28ChnCntrL).

All the above operations were detailed previously in this chapter. For setting up the "Phase programmable clocks" please refer to Section 10.

7.10. Uplink eLinks (inputs) Wrap-up

For the uplink to be operational the lpGBT mode has to be set to either "Simplex TX" or "Transceiver". In these two modes the lpGBT can receive data through the input eLinks and forward it to the counting room via the HS link.

- Program the DLLs of the active input ePort groups. All the groups share the same configuration (see register [0x034] EPRXDllConfig):
    - Set the DLL charge pump current;
    - Set the lock count number;
    - Disable / Enable clock gating of the DLL;
    - Disable / Enable DLL coarse lock detection.
- Set the general behavior of all active ePort groups (see register [0x034] EPRXDllConfig):
    - Disable / Enable re-initialization of the ePort groups when the phase selection is detected out of range;
    - Disable / Enable data gating along the replica delay lines (gating reduces power consumption but the actual value might vary from initialization to initialization).
- Set the detailed behavior of each ePort group:
    - Disable / Enable the inactive / active ePort channels and corresponding receivers (see [0x0c4] EPRX0Control to [0x0ca] EPRX6Control and [0x0e8] EPRXEcChnCntr);
    - Set the data rate for the active groups and enable the used channels within each group (see registers [0x0c4] EPRX0Control to [0x0ca] EPRX6Control). Notice that the bit rate can be set independently for each group;

    - Set the group phase-aligner tracking mode (see registers [0x0c4] EPRX0Control to [0x0ca] EPRX6Control);
- Set the ePort receivers (eRx) configuration:
    - Set the receiver signal polarity: Non-Invert / Invert (see [0x0cc] EPRX00ChnCntr to [0x0e7] EPRX63ChnCntr, and [0x0e8] EPRXEcChnCntr);
    - Disable / Enable the 100 Ohm termination (see [0x0cc] EPRX00ChnCntr to [0x0e7] EPRX63ChnCntr, and [0x0e8] EPRXEcChnCntr);
    - Disable / Enable AC biasing (see [0x0cc] EPRX00ChnCntr to [0x0e7] EPRX63ChnCntr, and [0x0e8] EPRXEcChnCntr);
    - Disable / Set the equalization (see [0x0cc] EPRX00ChnCntr to [0x0e7] EPRX63ChnCntr, [0x0e8] EPRXEcChnCntr, [0x0cf] EPRX03ChnCntr to [0x0e7] EPRX63ChnCntr and [0x0e8] EPRXEcChnCntr);

7.11. Downlink eLinks (outputs) Wrap-up

For the downlink to be operational the lpGBT mode has to be set to either "Simplex RX" or "Transceiver". In these two modes the lpGBT can receive data from the counting room via the HS link and ship it to the frontend devices via the output eLinks.

- Set the bit rate for the active groups (see register [0x0a7] EPTXDataRate). Notice that the bit rate can be set independently for each group;
- Disable / Enable the mirror function for specific (or all) groups (see register [0x0a8] EPTXControl);
- Enable the channels to be used (see registers [0x0a9] EPTX10Enable and [0x0aa] EPTX32Enable);
- Set the driving strength for each driver (see registers [0x0ac] EPTX00ChnCntr to [0x0bb] EPTX33ChnCntr and [0x0ab] EPTXEcChnCntr);
- Disable / Enable the pre-emphasis for each driver (eTx). To enable the pre-emphasis:
    - Set the pre-emphasis driving strength (see registers [0x0ac] EPTX00ChnCntr to [0x0bb] EPTX33ChnCntr and [0x0ab] EPTXEcChnCntr);
    - Set the pre-emphasis timing mode (registers [0x0ac] EPTX00ChnCntr to [0x0bb] EPTX33ChnCntr and [0x0ab] EPTXEcChnCntr);
    - If the self timing mode is selected select the pre-emphasis pulse width (see registers [0x0bc] EPTX01_00ChnCntr to [0x0c3] EPTX33_32ChnCntr and [0x0a8] EPTXControl).
- Set the driver polarity (see registers [0x0bc] EPTX01_00ChnCntr to [0x0c3] EPTX33_32ChnCntr and [0x0a8] EPTXControl).