An 8-channel programmable 80/160/320 Mbit/s radiation-hard phase-aligner circuit in 130 nm CMOS

2012 JINST 7 C12022

(http://iopscience.iop.org/1748-0221/7/12/C12022)

PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

RECEIVED: *October 9, 2012*
ACCEPTED: *November 12, 2012*
PUBLISHED: *December 20, 2012*

TOPICAL WORKSHOP ON ELECTRONICS FOR PARTICLE PHYSICS 2012,
17–21 SEPTEMBER 2012,
OXFORD, U.K.

An 8-channel programmable 80/160/320 Mbit/s radiation-hard phase-aligner circuit in 130 nm CMOS

F. Tavernier,[1] **S. Bonacini and P. Moreira**

CERN,
1211 Geneva 23, Switzerland

E-mail: filip.tavernier@cern.ch

ABSTRACT: The design of an 8-channel phase-aligner which is part of the GBTX chip for the LHC upgrade program is presented. The circuit is able to align the phases of up to 8 serial data streams to the GBTX transmitter clock so that the data can be merged, serialized and transmitted to the counting room. The bit rate is programmable at 80, 160 or 320 Mbit/s. Data jitter up to $\pm 3 \cdot T_{bit}/8$ can be tolerated without jeopardizing the error-free data reception. The phase-aligner has been designed as a radiation-hard circuit in a 130 nm CMOS technology and consumes only 3.5 mW at a supply voltage of 1.5 V.

KEYWORDS: Analogue electronic circuits; Radiation-hard electronics; VLSI circuits

[1] Corresponding author.

doi:10.1088/1748-0221/7/12/C12022

Contents

2012 JINST 7 C12022

1 Introduction

The presented phase-aligner is part of the GBTX chip that is currently under development at CERN [1]. The GBTX is a radiation-hard on-detector transceiver acting as an interface between the detector front-ends and the counting room. A bidirectional 4.8 Gbit/s optical link is used to connect the GBTX to the counting room. On the one hand, a programmable laser driver [2] and laser diode convert the GBTX data to the optical domain. On the other hand, the incoming optical data is converted to the electrical domain with a photodiode and transimpedance amplifier [3].

Bidirectional e-links (electrical links), operating at 80, 160 or 320 Mbit/s, connect the GBTX to the front-ends. Regardless of the bit rate and the direction of the data flow, the e-links provide the GBTX clock to the front-ends. This enables them to recover the data very easily because the clock does not need to be recovered. This is not the case for data that is received by the GBTX. Although the front-ends transmit their data synchronously with the GBTX clock, the phase of the received data is not aligned with it because of the delay introduced by the electrical cables and the front-end electronics. Moreover, this delay is different for every e-link since it depends on the distance between front-end and GBTX and thus on the location inside the detector.

As the front-ends operate synchronously to the GBTX, only the phase of the incoming data is to be found. Instead of aligning the clock to the data, the data is delayed by an amount such that it aligns perfectly with the GBTX clock. This clock can then be used to sample and recover the data. Having such an efficient way of clock and data recovery is important bearing in mind that up to 56 e-links can be connected to a single GBTX. The presented phase-aligner takes care of this task.

The architecture of the phase-aligner is treated in section 2. Some of the important building blocks and the design choices that have been made are discussed in section 3. The simulation results are shown in section 4. Finally, some conclusions are drawn in section 5.

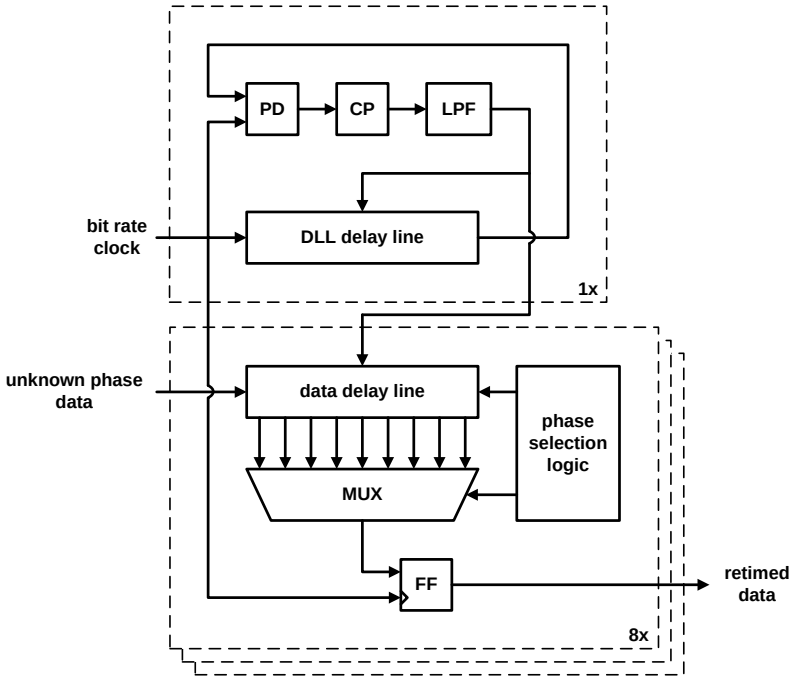


Figure 1. Architecture of the phase-aligner consisting of 1 DLL and 8 data recovery circuits.

2 Architecture

The architecture of the phase-aligner is shown in figure 1. It consists of 2 parts: a delay-locked loop (DLL) and a collection of 8 identical data recovery circuits that generate the *correctly* delayed data and sample it with the bit rate GBTX clock. The data with unknown phase is delivered to these 8 recovery circuits by 8 e-links that are all programmed at the same bit rate: 80, 160 or 320 Mbit/s. The number of data recovery circuits per DLL is a trade-off between flexibility, power and area efficiency. From a flexibility point of view it is preferable to have fewer data recovery circuits per DLL as then every e-link can be programmed at a desired bit rate. However, from a power and area efficiency point of view, it is best to connect more e-links to a common DLL. Having 8 recovery circuits per DLL proved to be optimal for the GBTX design case.

The goal of the DLL is to lock onto the bit period. This means that the delay in the DLL delay line is then exactly equal to the bit period. The control voltage which is used to regulate this delay is shared with the data delay lines in the 8 data recovery circuits as can be seen in figure 1. The outputs of the data delay lines are all versions of the unknown phase input data which are delayed to an extent as defined by the DLL. The phase selection logic and multiplexer (see figure 1) then select the *optimal* delayed version of the input data so that it can be sampled with the GBTX bit rate clock. In this perspective, *optimal* means that the data is sampled in the middle of its bit interval.

The resolution of the delay line outputs has been chosen equal to $T_{bit}/8$ resulting in a delay between 2 consecutive outputs of 1.5625 ns, 781.25 ps or 390.625 ps for a bit rate of 80, 160 or 320 Mbit/s respectively. This resolution is fine enough to be able to sample the data in the middle of the eye opening while not complicating the design of the delay lines and the phase selection logic too much. The length of the DLL delay line logically equals T_{bit} as the DLL has to lock to

2012 JINST 7 C12022

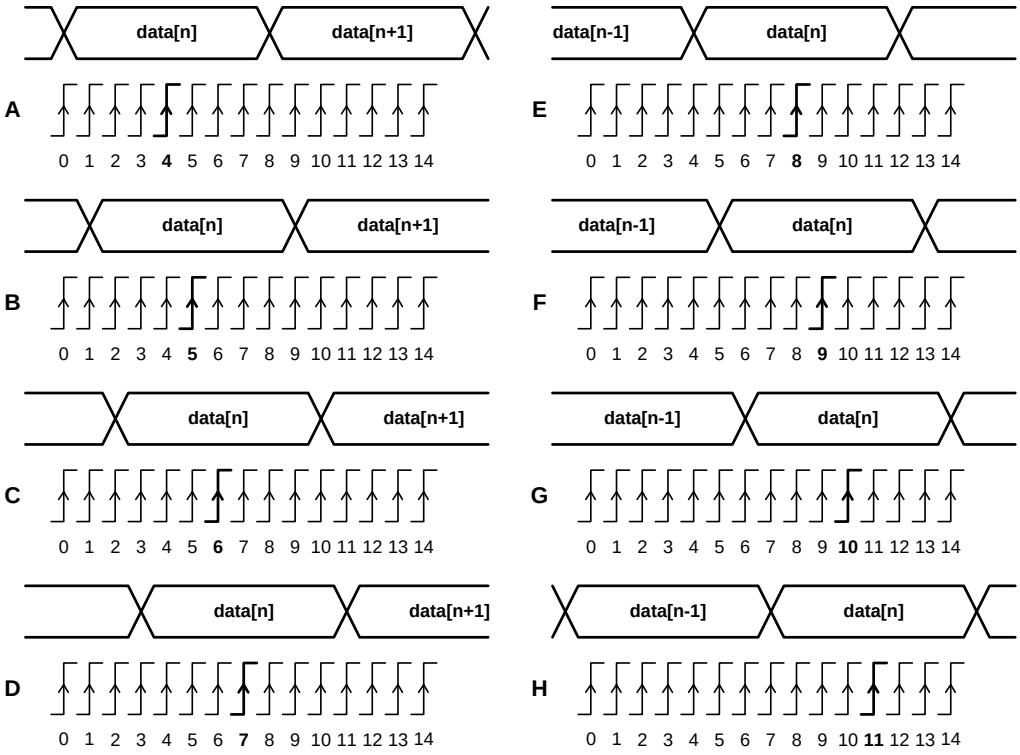


Figure 2. Operation principle of the phase-aligner with a data delay line length of $7 \cdot T_{bit}/4$ and a resolution of $T_{bit}/8$. For ease of representation, the clock is delayed while in reality the data is delayed.

this value. However, the data delay lines have a length of $7 \cdot T_{bit}/4$ which is almost twice the length of the DLL delay line. This has been done so that they store more of the history of the data and as such can track the phase in more situations. This can be seen in figure 2 where the operation principle of the phase-aligner is clarified. In this figure the 15 versions of the data (1 non-delayed and 14 delayed) are represented by means of clock delays as this is easier to visualize. However, this is completely equivalent with the reality where only 1 clock exists and the data is delayed. The phase selection logic determines which of the 8 cases (A-H) is valid by determining the location of the data transitions. Based on that, it chooses the correct output tap of the data delay line (bold clock in the figure) to be sampled by the GBTX clock. Note that with the rule as illustrated in figure 2, data[n] is sampled in all 8 cases which is a condition for constant latency. This is possible because the length of the data delay lines exceeds the bit period. One can easily see that if only 8 delayed versions of the data were available (delay line with a length equal to the bit period), data[n] would be sampled only in cases A-E. In cases F-H, data[n-1] would be sampled.

The phase selection logic has 3 modes of operation. The first mode is *static phase selection* in which the selection of the delayed data version is done from configuration. Obviously, this mode has the disadvantage that variations that might occur due to temperature and power supply fluctuations are not tracked. Moreover, the ideal sampling phase settings must be determined and provided by the system operator. The second mode is *automatic lock at start-up only* which uses a training sequence to acquire lock after which the phase selection is static as in the first mode. Although this operation mode still suffers from the same phase tracking problems of the first mode, it does

2012 JINST 7 C12022

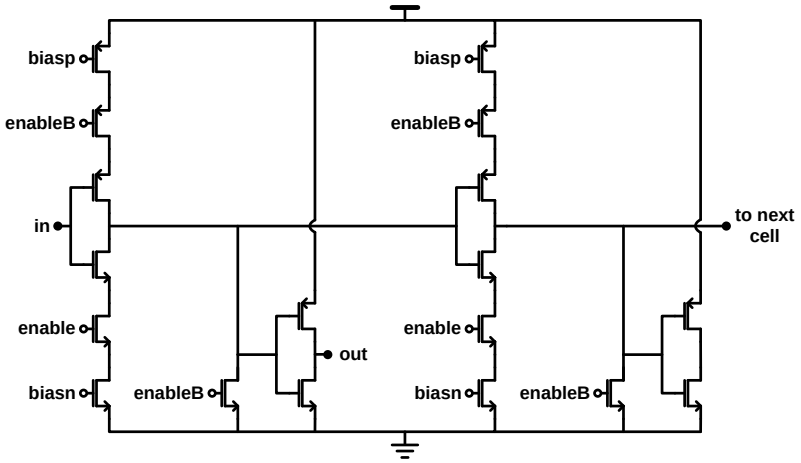


Figure 3. Circuit diagram of the delay cell.

not require the intervention of an operator to determine the ideal initial sampling phase. The third mode is *automatic phase tracking* in which the phase selection logic continuously adapts the selected output tap of the data delay lines so as to always sample in the middle of the bit interval. In this mode, the phase selection logic selects the optimal delayed data version in every clock period. However, 2 conditions are necessary in order to prevent the system from jumping from case A to case H which would result in the sampling of d[n-1] instead of d[n]. Firstly, only smooth transitions are allowed meaning that the choice of output tap of the delay line can only change with 1 position every time. Secondly, the variation with respect to the ideal output tap is restricted to $\pm 3 \cdot T_{bit}/8$ which, in the case of 15 output taps, always guarantees the sampling of d[n] as can be verified in figure 2. As such, it can be stated that the jitter tolerance of this phase-aligner is $\pm 3 \cdot T_{bit}/8$. Even with this amount of jitter, the data is sampled in the middle of the eye.

3 Building blocks

3.1 Delay line

The DLL and data delay lines are composed of an array of identical delay cells, the circuit diagram of which can be seen in figure 3. As explained already before, the total delay of both delay lines varies with the chosen bit rate. For the DLL delay line this means that the total delay needs to vary from 12.5 ns at 80 Mbit/s to 3.125 ns at 320 Mbit/s. The length of the data delay lines needs to vary from 21.875 ns at 80 Mbit/s to 5.468 75 ns at 320 Mbit/s. Such a large tuning range is not feasible with the delay cell as shown in figure 3. Therefore, instead of changing the cell delay, it has been decided to change the amount of cascaded delay cells if the bit rate is altered. Consequently, the delay cells are always operated in the same operating point with the same delay. In order to be able to generate the delayed data with the finest resolution, namely $T_{bit}/8$ at a bit rate of 320 Mbit/s, the cell delay equals 390.625 ps. For a bit rate of 160 Mbit/s and 80 Mbit/s, the number of delay cells between 2 consecutive output taps is increased to 2 and 4 respectively. The total number of delay cells in every delay line is determined by the lowest bit rate. Therefore, the DLL delay line is composed of 32 cells whereas the data delay lines have 56 cells.

2012 JINST 7 C12022

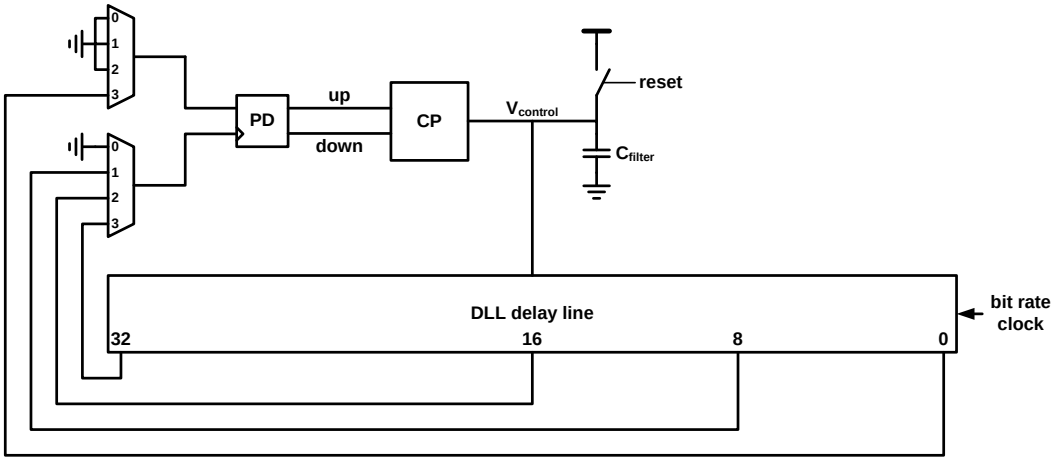


Figure 4. Simplified topology of the DLL.

As can be seen in figure 3, the delay cell is composed of 4 inverters, 2 of which are current-starved. A delay line is formed by connecting the output of the second current-starved inverter to the input of the first current-starved inverter of the next cell. As such, a delay line is composed only of current-starved inverters, the delay of which can be tuned by means of changing the current, i.e. changing the gate voltages of the current sources. The left normal inverter is used as an output buffer. It adds a fixed delay to all output taps and thus does not contribute to the delay between the taps. The right normal inverter is a dummy so as to not introduce any duty-cycle distortion originating from a difference in rise and fall time of the current-starved inverters. Each cell can be enabled/disabled if it is not used at the higher bit rates and thus save the related power consumption.

3.2 DLL

The topology of the DLL is shown in figure 4. As already discussed in section 3.1, the DLL delay line is composed of 32 delay cells, each having a delay of 390.625 ps. An extra delay cell is added in front of these 32 cells to generate the output tap 0 as can be seen in figure 4. This has been done because the delay between 2 output taps is generated partly by the second current-starved inverter in one delay cell and partly by the first current-starved inverter in the consecutive delay cell (see figure 3). Thus, the reference clock needs to be delayed by *half a delay cell*. Based on the applied bit rate clock the correct output signal of the DLL delay line is selected by means of a multiplexer: tap 32 for 80 Mbit/s, tap 16 for 160 Mbit/s or tap 8 for 320 Mbit/s. The multiplexer has a balanced design so that the delay from each of its inputs to the output is equal.

The bang-bang phase detector is a flip-flop that determines every clock cycle whether the DLL delay line output is too early or too late compared to the output tap 0. If it is too early it generates a DOWN signal so that the charge pump discharges the filter capacitor C_{filter} and the cell delay increases. If the output of the delay line is too late, the phase detector generates an UP signal so that the charge pump charges C_{filter} and the cell delay decreases.

The charge pump current can be programmed with a resolution of 0.85 µA up to 12.75 µA by means of a 4-bit DAC. As can be seen in figure 5, it has a differential architecture where the sink and source currents are always flowing so that the start-up time of the current sources is eliminated

2012 JINST 7 C12022

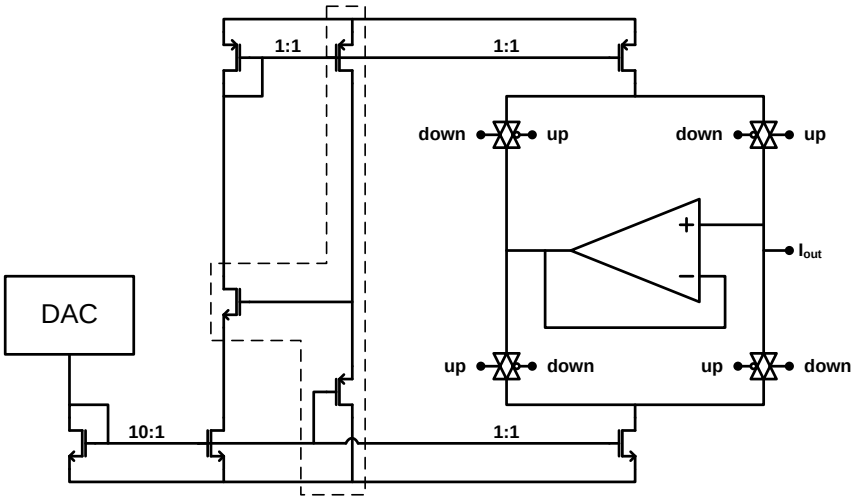


Figure 5. Circuit diagram of the charge pump.

when switching from sinking to sourcing current or vice versa. Moreover, by equalizing the drain voltage of the idle current source to the drain voltage of the used one by means of the unity gain amplifier, charge sharing is prevented as well. The part of the circuit inside the dashed box decouples the drain voltage of the left nMOS current source, used to generate the gate voltage for the pMOS current source, from the gate voltage of the pMOS diode. The gate and drain voltage of the left nMOS current source are equalized by means of the double source follower so that its current is a better copy of the DAC current. Consequently, the sink and source currents of the charge pump are better replicas of each other which results in a more symmetrical DLL behavior.

In contrast to a voltage-controlled oscillator which has a 1^{st} order response from input voltage to output phase, the DLL delay line has a 0^{th} order response from input voltage to delay. Therefore, a stabilizing zero which is necessary in a PLL is not required in a DLL. Consequently, the loop filter consists only of a filter capacitor as can be seen in figure 4. Its capacitance should be maximized so as to minimize the jitter when the DLL is locked. This jitter originates from the bang-bang nature of the phase detector which decides every clock cycle to charge or discharge the loop filter. It can be minimized by increasing the filter capacitance, decreasing the charge pump current or reducing the *delay gain* of a delay cell. The filter capacitance in the phase-aligner equals 40 pF.

Care has to be taken when starting or resetting the DLL so that it does not lock onto a multiple of the bit period or so that it can lock in the first place. Therefore, the DLL delay line is always initialized at its shortest delay setting, i.e. the highest possible control voltage. This is implemented by means of a switch that pulls the control voltage to the supply voltage as shown in figure 4. From this minimal delay setting, the delay is gradually increased so that it reaches a value equal to the bit period before reaching a value equal to a multiple of the bit period. However, this only guarantees that if the DLL locks that it locks on the bit period. It is not enough to assure that the DLL will lock. This is because the phase detector can only make correct decisions if the rising edge of the output of the delay line is within $\pm T_{bit}/2$ around the rising edge of the bit rate clock. For example, if the minimum delay setting after a reset results in a delay line delay smaller than $T_{bit}/2$, the phase detector will try to decrease the delay even further by generating an UP signal for the charge pump.

2012 JINST 7 C12022

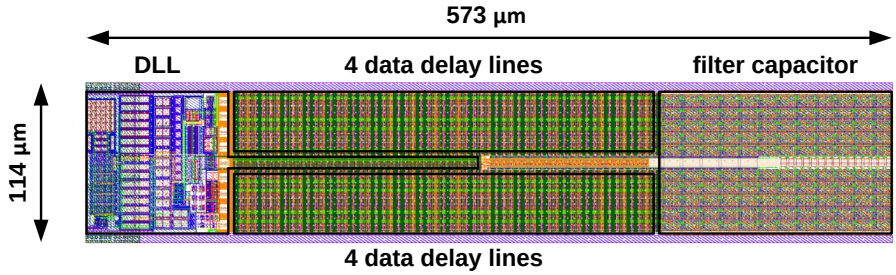


Figure 6. Layout of the *analog* part of the phase-aligner.

In order to prevent this from happening, the loop is opened after a start-up or reset and a DOWN signal is imposed at the input of the charge pump increasing as such the delay in the DLL delay line. After a number of cycles, the delay of the DLL delay line will reach $T_{bit}/2$. At that point, the phase detector can operate correctly and the control is given back to the DLL which then further steers the control voltage in the correct direction.

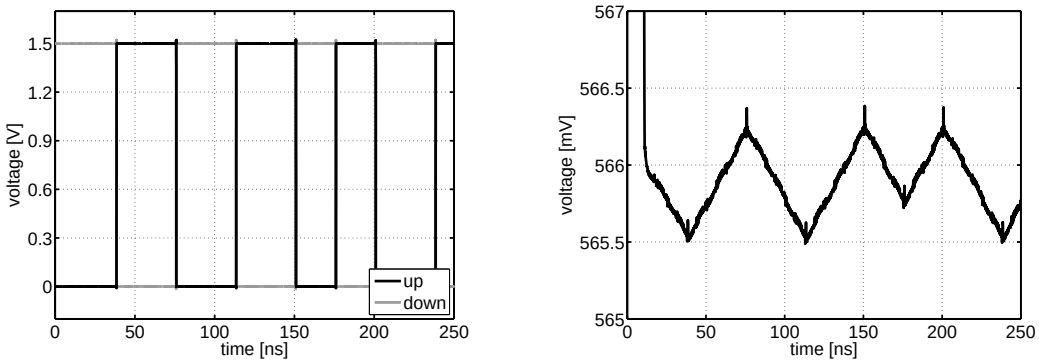
4 Simulation results

The phase-aligner has been designed in a 130 nm CMOS process. The layout of the circuit can be seen in figure 6. Note that this layout only comprises the *analog* part of the phase aligner. The *digital* part consisting of the multiplexer and the phase selection logic is not shown here. As already mentioned before, the phase-aligner is meant to be incorporated in the GBTX chip. Therefore, no pads or power clamps have been included in the layout. In order to obtain the best matching between the DLL delay line and the 8 data delay lines, the former has been placed between 2 sets of 4 data delay lines as can be seen in figure 6. The filter capacitor is a thick-oxide nFET in an n-well. Therefore, it operates in accumulation instead of inversion and consequently acts as a capacitor already from a zero bias voltage. A thick-oxide device has been selected in order to prevent gate leakage and, more importantly, the accompanying gate leakage noise [4].

The extracted netlist of the phase-aligner has been simulated in 22 simulation corners in which the supply voltage ranges from 1.35 V to 1.65 V with a nominal value of 1.5 V, the temperature is varied between $-30\,°C$ and $100\,°C$ and the process corner is altered between the typical, slow and fast versions. Random data is applied at all 8 data delay lines which all have the correct frequency but the wrong phase compared to the GBTX bit rate clock. The minimum charge pump current is used which is optimal in order to minimize the jitter originating from the bang-bang nature of the loop. However, it also results in a very slow settling behavior of the DLL because a current of 0.85 µA needs to discharge the filter capacitor of 40 pF from the supply voltage down to the correct control voltage (± 50 µs). In order to circumvent the long simulation times, the charge pump current is artificially increased before lock is achieved.

The UP and DOWN signals as well as the control voltage in the typical corner are shown in figure 7 for a bit rate of 80 Mbit/s. The artificially increased charge pump current in the first few nanoseconds can clearly be seen in figure 7b. After that, the charge pump current is reduced to the minimal value, namely 0.85 µA. It can be noticed that the UP and DOWN signals switch

2012 JINST 7 C12022



(a) UP and DOWN signals of the phase detector

(b) control voltage

Figure 7. Simulated control signals at 80 Mbit/s and the minimal charge pump current in the typical corner.

on a regular basis resulting in a control voltage that drifts around its *ideal* value. This results in jitter at the output of the DLL delay line. In order to quantify the amount of jitter reliably, enough clock cycles are required. Therefore, a Verilog simulation has been performed as it would take too long for a circuit simulator. This simulation has shown that the jitter has a standard deviation of around 10 ps at the 80 Mbit/s setting which is small enough for the intended application. The power consumption of the analog part of the phase-aligner is only 3.5 mW for a supply voltage of 1.5 V.

5 Conclusion

A phase-aligner has been presented that efficiently recovers the data from 8 channels. Because the frequency of all incoming data is equal to the bit rate clock in the receiver, only the phase of the data needs to be adapted so that it can be recovered properly. This has been achieved by using a DLL that locks on the bit period and 8 delay lines that are replicas of the DLL delay line. By making the data delay lines longer than 1 bit period, a jitter tolerance of $\pm 3 \cdot T_{bit}/8$ has been realized.

References

[1] P. Moreira et al., *The GBT project*, in *Proceedings of the Topical Workshop on Electronics for Particle Physics*, September 2009, pg. 342.

[2] G. Mazza et al., *The GBLD: a radiation tolerant laser driver for high energy physics applications*, in *Proceedings of the Topical Workshop on Electronics for Particle Physics*, September 2012.

[3] M. Menouni, P. Gui and P. Moreira, *A radiation-tolerant 5 Gbps transimpedance amplifier for CERN's GBT project*, in *Proceedings of the Topical Workshop on Electronics for Particle Physics*, September 2009, pg. 326.

[4] J. Lee, G. Bosman, K.R. Green and D. Ladwig, *Noise model of gate-leakage current in ultrathin oxide MOSFETs*, *IEEE Trans. Electr. Dev.* **50** (2003) 2499.

2012 JINST 7 C12022