Timewalk measurements in the VeloPix ASIC

Larissa Helena Mendes, FEEC - Unicamp

***Abstract*—LHCb is a dedicated heavy flavour physics experiment that operates at the LHC. The LHCb silicon VErtex LOcator detector (VELO) will be fully replaced for the experiment upgrade by mid-2020. The new detector will replace the silicon micro-strip detector currently operating around the interaction point. It will use hybrid pixel detectors composed of silicon bump-bonded sensors to VeloPix CMOS readout chips designed for the new 40 MHz readout rate. Each module will be equipped with 4 silicon hybrid pixel tiles, each readout by 3 VeloPix ASICs, totaling about 41 million 55×55 μm^2 pixels in an array of 52 modules.**

The timewalk effect is caused by the different inclination of signals with different amplitudes and is directly related to the detector's temporal resolution. The study of this effect in VeloPix is reported in this work.

***Index Terms*—Hybrid detectors, Timewalk, VeloPix, Data processing methods, Reproducible Research.**

I. Introduction

During Run I, LHCb ran at a luminosity (number of collisions per cm^2 and per second) of $4 \times 10^{32} \mathrm{cm}^{-2}\ \mathrm{s}^{-1}$ and is expected to increase to $2 \times 10^{33}\ \mathrm{m}^{-2}\mathrm{s}^{-1}$ The luminosity is restricted by trigger on both hardware and software level and the former VELO has a readout limit of 1 MHz. To run at a higher luminosity, the hardware trigger will be refunded with a full software trigger readout system of 40 MHz.

For this, the current VELO detectors which are based on silicon microstrip detectors will be replaced by hybrid pixel detectors with 200 μm thick n-on-p silicon sensors bump-bonded to VeloPix ASICs with the 40 MHz readout rate. VeloPix is based on the Timepix3 readout chip and consists of 256×256 pixels of 55 μm pitch.

This study on the timewalk on the VeloPix ASIC aims to propose a method to quantify this effect for further investigation of the impact on its performance.

A. VeloPix

Table 1 shows the specifications of VeloPix. As the chip operates with a clock of 40 MHz, hits will be recorded with a precision of 25 ns. One orbit of the LHC is 3564 clock cycles or bunch crossings, and a 9-bit timestamp is used to identify each cycle. In this note, this timestamp will be referred to as BXID (bunch-crossing identification number) [1].

The front-end of VeloPix including is shown in Figure 1. There is a functionality for test pulse injection which is digitally controlled. The expected detector capacitance C_{det} is approximately 50 fF. With this feature, it is possible to inject different known amounts of charges into the pixel circuit by adjusting the TpA and TpB voltages applied to the capacitor. Then, the charge value is correlated with the digital readout value. The front-end also has a 4-bit DAC per pixel for threshold tuning [4].

Readout type	Continuous, trigger-less, binary
Timing resolution/range	25 ns, 9 bits
Power consumption	$<$1.5 Wcm^{-2}
Pixe matrix, pixel size	256×256, $55\ \mu m \times 55\ \mu m$
Technology	Planar silicon, e- collection
Radiation hardness	400 Mrad, SEU tolerant
Maximum data rate	900 Mhits/s/ASIC, 50 khits/s/pixel

TABLE I: VeloPix specifications

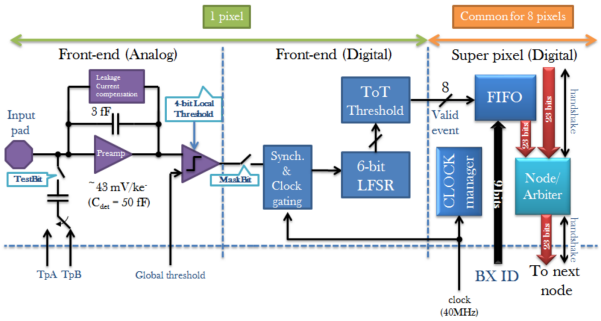

Fig. 1: A schematic of the front-end of VeloPix [7]

B. Timewalk

As can be seen from Figure 2, the signal takes some time to reach the threshold [5]. The discriminator will detect the signal later than the actual arrival time of the hit on the chip. This delay is called timewalk and depends on the signal amplitude. Timewalk leads to incoherence in the measurement of the z-position of a hit, which expresses as an error in the vertex tracking.

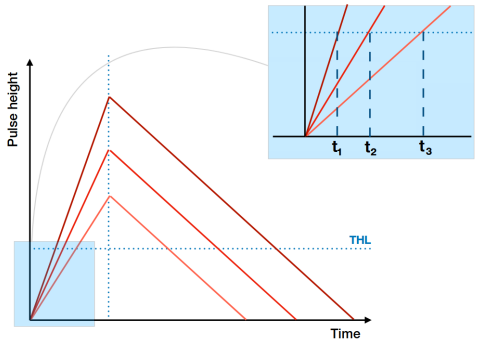

Fig. 2: A schematic of the timewalk effect: Pulses of different amplitudes in the preamplifier cross the threshold level at different times. Hence, smaller amplitude signals take longer to detect.

II. Experimental Setup and Materials

To exercise the capability of the chip, it was developed a compact readout system called Speedy PIxel Detector Readout (SPIDR) [7] [6]. The SPIDR system consists of a Xilinx

VC707 FPGA board with various communication interfaces, firmware, CPU subsystem, and an API library. The main readout board connects to both 10 Gb Ethernet and 1 Gb Ethernet devices.

Fig. 3: Chip carrier board without ASICs connected to the main SPIDR board.

A. Method

As mentioned before, in the analog part of the pixel there is also a charge injection circuit called Test Pulses. By varying the voltages of the test pulse it is possible to simulate the charges generated by particles with different energies going through the sensor.

For this test, a series of charges are applied, controlled by the DAC Vtp_Coarse and Vtp_Fine. These DACs correspond respectively to TpA and TpB in Figure 1. The injected charge value can be calculated from the analog values of Vtp_Coarse and Vtp_Fine. In this work, we are going to use directly the analog value $\Delta\mathrm{VTP} = \mathrm{VTP\_coarse} - 4 \cdot \mathrm{VTP\_fine}$. But, the corresponding charge could be easily calculated by

$$\begin{aligned} \mathrm{Q_{in}} &= (\mathrm{VTP\_fine} - 4 \cdot \mathrm{VTP\_Coarse}) \cdot \mathrm{C_{det}} \\ &= \Delta\mathrm{VTP} \cdot \mathrm{C_{det}} \end{aligned} \tag{1}$$

To configure the chip in high-speed readout configuration and apply the test to measure timewalk, the following steps were executed:

- **Enable GWT** The SPIDR system processes the 20 Gbps scrambled data stream from the VeloPix and distributes it over 1 Gigabit Ethernet link using VELO Gigabit Wireline Transceiver (GWT).
- **Reset and enable PLL and reset GWT DLLs** Phase-Locked Loop (or phase lock loop) is a control system that generates an output signal whose phase is related to the phase of the input signal. PLL allows the chip to compensate frequency fluctuations in all chips, caused by process parameters and/or temperature variations [1]. A delay-locked loop (DLL) is a digital circuit similar to a PLL but uses a phase (=delay) variable block instead of a frequency block variable.
- **Reset the pixel matrix** Reset the pixel before reading the data.
- **Do general configurations to link shutter to the test pulse** It is necessary to configure the link between the shutter and the test pulse. If not linked, the shutter is independent of the test pulse. Otherwise, the shutter only opens when the pulse is enabled.
- **Sweep Pixel Array** The active columns and rows are numbered from 0 to 255. Row 0 is the one closest to and row 255 the one furthest away from the periphery. A chess-like mask is applied over the pixel matrix, activating some pixels and disabling its neighbors to avoid cross-talk between the pixels. An example of a mask is shown in Figure 4.
  After selecting the pixels to sweep the matrix, for each unmasked pixel, the test pulse is enabled.

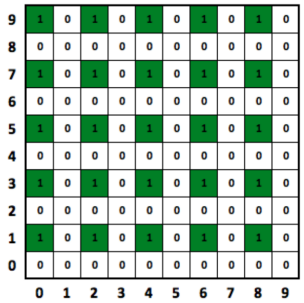

Fig. 4: Example of mask application in a 9x9 pixel matrix. The pixels with the bit 1 are enabled, and the pixels with the bit 0 are disabled

- **Setup the test pulse generation** There is one set of TFC commands per BXID, at 40 MHz. TFC will make sure that the system is synchronized or that the usage of the command is consistent. In this case, as we are working with TFC, the test pulse is injected into the analog front-end.
  Setting the test pulse voltage the values DAC Vtp_Coarse (fixed at 128) and Vtp_Fine (scanned from 0 to 1024) are written to its respective registers. However the value $\Delta\mathrm{VTP} = \mathrm{VTP\_coarse} - 4 \cdot \mathrm{VTP\_fine}$ is the one recorded for analysis.
  It is possible to divide the 25 ns clock into 16 steps, equivalent to 1.5 ns, which is the VeloPix phase. Then, the test pulse is implemented using four parameters: the number of pulses, the test pulse phase, the number of clocks the test pulse is high, and the number of clocks the test pulse is low.
  The VeloPix data is processed into events where each hit is associated with a timestamp. In this work, we are simulating those hits using the test pulse injection (BXID). The output is an ASCII file that contains the event number, the trigger time, and the associated hit information; namely, the column and row number, the test pulse voltage (ΔVTP), and the hit count, that depends on the number of pulses injected. This data is later decoded and stored in a ROOT tree so that the final output is a ROOT file which contains all the information related to the tracks.

III. Analysis and Results

The analysis presented in this work was performed in ROOT C++ (version 6.10/04) in a Jupyter notebook environment (version 6.0.3) and is openly available at GitHub [2]. Although the code with the method presented in section II-A to generate the data is not available to reproduce, the raw data and pre-processing code are available at Zenodo [3]. Each pixel of the sensor has its circuit (see Figure 1), therefore the analysis is done pixel by pixel. We divide its 25 ns clock into 16 phases of 1.5 ns. Varying the phase, we can delay the entry of the pulse to see the moment when it changes the Bunch Crossing counter (BXID). Figure 5 shows this behavior for one ΔVTP and one pixel only.

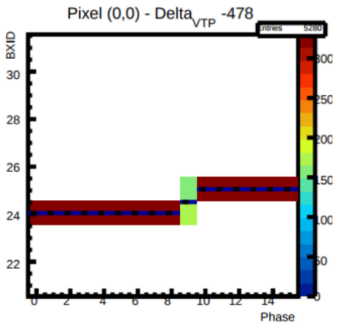

Fig. 5: Phase vs BXID behavior for pixel in position 0x0 (row x column)

By doing this for multiple charges on the same pixel (see Figure 6), we can confirm that the larger the pulse, the earlier the event is identified.

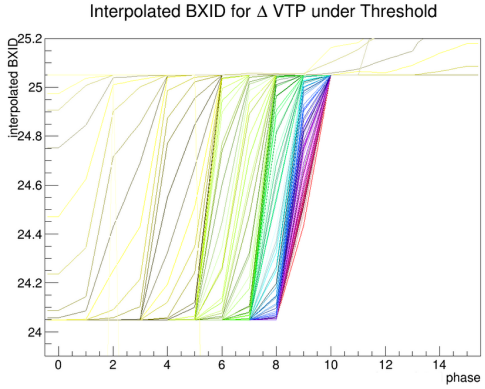

Fig. 6: BXID vs phase for multiple charges, for one pixel. Colors going from yellow (lower ΔVTP) to pink (higher ΔVTP))

To quantify the timewalk, the approach will be to compare the delay calculated as $[\text{BXID} + (1.0/16) \times \ (15 - \text{phase})]$ with the charge (ΔVTP). And since BXID and the phase are time measures, we can put them together so we can see what BIXD each voltage has (see Figure 7).

Taking the extremities of the histogram above (Figure 7), we get 3 curves that say which event the signal is dropping depending on the height of the pulse.

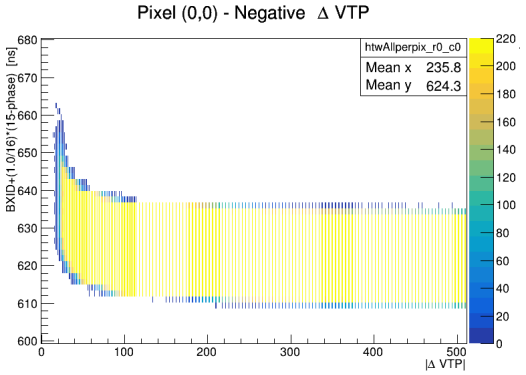

Fig. 7: Delay for multiples ΔVTP. Z axis represents.)

For this, we project each ΔVTP value on the Y axis, leaving us with a "step-like" histograms with the delay for each charge value. We can set a height value, for example 30% of the maximum, and find the point edges (left and right) and midpoint. Figure 8 shows this projections for one of the ΔVTP values of the histogram on Figure 7. The vertical line represents the height value of 30% of the maximum count rate, where the interceptions point with the "step" are taken.

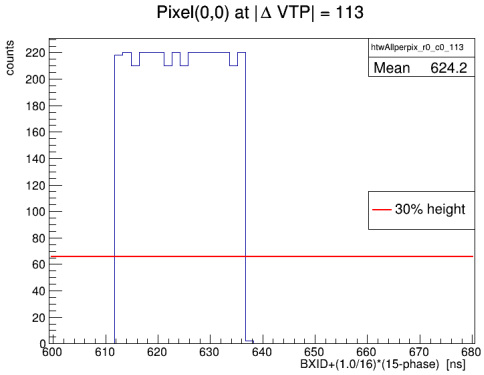

Fig. 8: "step-like" histogram representing the projection of the previous histogram at the ΔVTP value of 113. The points of the histograms that crosses the illustrative red line are the edges of the first histograms from this specific ΔVTP value.

The first point that the histogram intercepts the line is being called "Lower edge", the second interception point is called "Upper edge". The middle point, taking the middle distance between them, is called "Average point".

Taking the edge points for all ΔVTP values, we get 3 curves that say which event the signal is dropping depending on the height of the pulse (see Figure 9).

As all 3 curves behave the same way, we can chose the lower edge and fit an appropriate function, which in the case can be an hyperbolic. This fit is still done pixel by pixel.

$$\frac{\text{p0}}{(|\Delta\text{VTP}|-\text{p1})^{\text{p2}}} + \text{p3}$$

Doing this for all pixels in the matrix we see the parameters values for all pixels (see Figure 11).

Now, we can use this result to calculate the minimum charge necessary to detect a collision in the right event/BXID.

Since the function has a horizontal asymptote (Figure 10), the heigh of this asymptote corresponds to the time of an

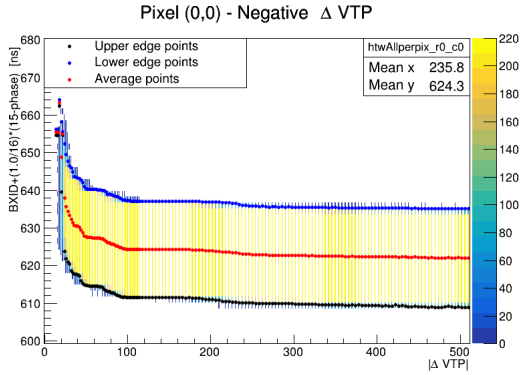

Fig. 9: Edge points for all ΔVTP values.

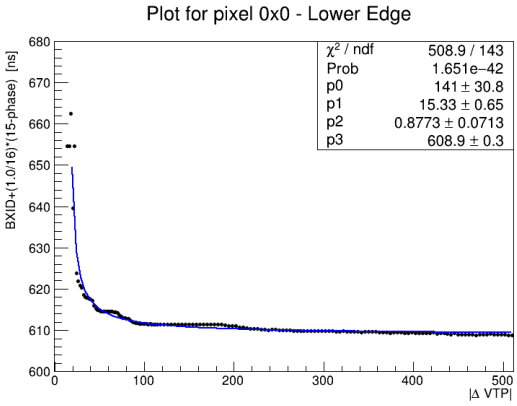

Fig. 10: Fit for pixel 0x0.

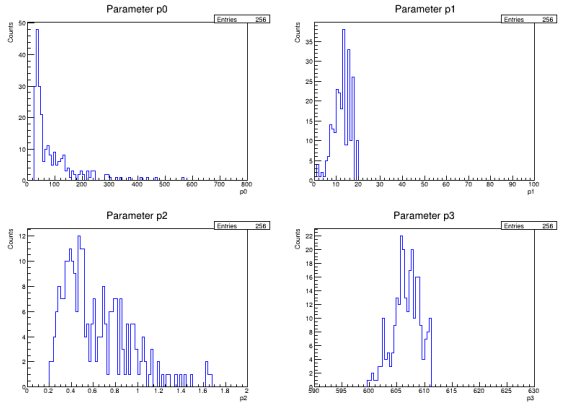

Fig. 11: Values of the parameters for the 256 pixels analysed.

infinite ΔVTP (or charge, considering a real particle hit). Figure 2 shows that the higher the signal amplitude, the earlier it is detected. Therefore, an infinite signal would be the first one to be detected. In this case, we are considering the value of parameter p3 to be an approximation to the time delay of this infinite signal.

We can now finally look for the minimum ΔVTP that makes the difference between its delay and parameter p3 less than 25 ns, which is the time resolution of VeloPix. We will call this the minimal in-time ΔVTP.

We are working with a charge simulation using the input voltage ΔVTP, but as shown in section II-A, the corresponding charge could be easily calculated, as shown on equation (1).

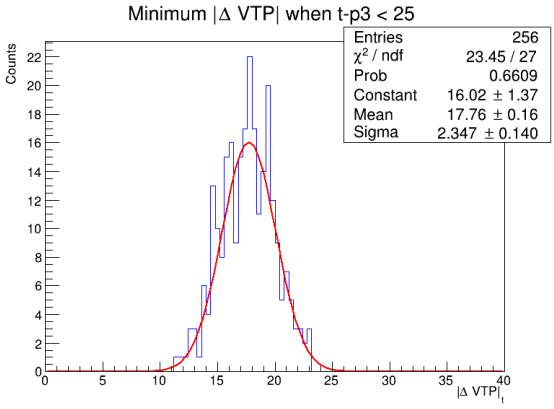

Fig. 12: Minimal in-time ΔVTP.

From the histogram in Figure 12, we can find a global value of ΔVTP of almost 18 ns. This means that detecting a signal with amplitude greater than this value, the chances are that this pulse will be counted only in the next event.

IV. Conclusion

In total, VELO consists of 52 modules positioned in vacuum along the LHC beampipe, surrounding the interaction point, summing 624 VeloPix ASICs. This paper proposes a method to quantify the Timewalk effect on the VeloPix ASIC.

What Figure 12 shows is the minimum charge required to have a particle hit associated with the same event of higher charges. At the VELO, if a particle from a certain collision with a charge is less than the minimum calculated for that chip, this hit will not be identified that same collision, but the next one. Since the value of this minimum charge can be calculated pixel by pixel, this value can be recorded so that any hit in that pixel with a smaller amplitude can be ignored so that it is not correlated to the wrong event.

The analysis of this work can be used as a basis for further investigations on the subject, and as this ASIC is derived from the Medipix family, this study can be adapted to other chips as well.

References

[1] Y Fu, C Brezina, K Desch, T Poikela, X Llopart, M Campbell, D Massimiliano, V Gromov, R Kluit, M van Beauzekom, F Zappon, and V Zivkovic. The charge pump PLL clock generator designed for the 1.56 ns bin size time-to-digital converter pixel array of the timepix3 readout ASIC. *Journal of Instrumentation*, 9(01):C01052–C01052, jan 2014.

[2] Mendes Larissa Helena. Timewalk measurements in the velopix asic. https://github.com/larissahmendes/timewalk, 2020.

[3] Larissa Helena Mendes. Velopix timewalk dataset. https://doi.org/10.5072/zenodo.631441, June 2020.

[4] T. Poikela, M. De Gaspari, J. Plosila, T. Westerlund, R. Ballabriga, J. Buytaert, M. Campbell, X. Llopart, K. Wyllie, V. Gromov, M. van Beuzekom, and V. Zivkovic. VeloPix: the pixel ASIC for the LHCb upgrade. *Journal of Instrumentation*, 10(01):C01057–C01057, jan 2015.

[5] Mijke Schut. Characterisation of the Timepix3 chipusing a gaseous detector, Feb 2015.

[6] B. van der Heijden, J. Visser, M. van Beuzekom, H. Boterenbrood, S. Kulis, B. Munneke, and F. Schreuder. SPIDR, a general-purpose readout system for pixel ASICs. *JINST*, 12(02):C02040, 2017.

[7] J Visser, M van Beuzekom, Henk Boterenbrood, B van der Heijden, J I Muñoz, S Kulis, B Munneke, and F Schreuder. SPIDR: a read-out system for Medipix3 amp; Timepix3. *JINST*, 10(12):C12028, 2015.