MU2E/PHYSICS/xxxxx
version 1.0 November 30, 2023

Commissioning of the Mu2e Data AcQuisition system and the Vertical Slice Test of the straw tracker

11. Mu2e ROC simulation

version 1.0 November 30, 2023

Abstract

This note presents the initial results of the tracker DAQ commissioning.

Contents

1 Notes for the authors

1.1 Revision history

- v1.01: initial version

2 Introduction to the analysis

In this note, we present initial results of the tracker DAQ commissioning.

3 Description of teststand setup

The tracker teststand, called TS1, includes one DRAC card and one DTC connected to the DAQ computer, 96 channels in total. The ROC can be operated in two different data readout modes: the first with emulated data readout mode, called mode one, and the reading digi FPGA readout mode, called mode two. Most of the data were taken operating in the mode two with digi FPGAs, pulsed by their internal pulser. The pulser has two different frequencies, 31.29 MHz/(2^7+1), or approximately 250 kHz, and 31.29 MHz/(2^9+1), or approximately 60 kHz. Event window is the time interval between two heartbeats (HB's). The logic of data taking is shown in Fig. 4. Pulses are represented with gray triangles and they are separated by the inverse of the generator frequency. We call $T_{gen} = 1/f_{gen}$. The event window, with the width of T_{EW}, that represents the distance between the proton pulses, was varied from 700 ns to 50 μs. The ROC firmware has an internal hit buffer which stores up to 255 hits. That should be sufficient for the data taking. Depending on T_{gen} and T_{EW}, the data taking can proceed in two different modes:

- The event window is large enough , so the total number of generated hits is greater than 255. In this case the ROC hit buffer always gets filled up, and only the first 255 hits are read out;
- The total number of hits within the event window is less than 255. In this case the ROC hit buffer doesn't get filled up and the total number of hits can vary from one event to another.

Each FPGA has its own completely random generator and pulses from different generators are offset with respect to each other. This offset can be a random number between 0 and T_{gen}. Timing of generator pulses are uncorrelated with the beginning of the time window and because of this different number of hits can fit in the event window, as we can see in Fig.4. Within one FPGA, pulses digitize in different channels and they have different timing, so they are offset of the order of nanoseconds. Readout sequence is defined: all channels are read in a specific sequence that doesn't change.

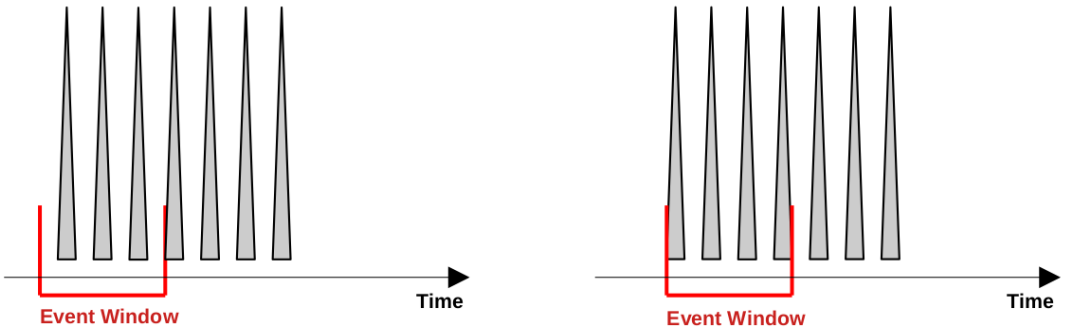

Figure 1: Graphic illustration of pulses in an event window.

4 Monte Carlo simulation

The ROC readout logic is purely digital, so the readout process can be simulated. Hits in each channel are parameter that can be simulated. The logic simulation is as follows. The simulated parameters for each

event are the number of hits in each channel and the total number of readout hits. In the following sections, $occupancy$ will define the total number of hits versus channel number.

Given that the maximum allowable number of hits per event is 255, the simulation follows these steps:

- The event window starts at $t = 0$s;
- The first pulse is generated randomly from 0 to T_{gen}, by following a uniform distribution;
- The previous pulses are generated subtracting from the first one a step of T_{gen}, until the absolute time of the pulse is greater than T_{EW};
- Pulses are generated in each channel following the readout sequence;
- The process of pulses creation continued until the count of hits reached the maximum threshold of 255.

Furthermore, the simulation takes into accounts the FPGAs offsets and the individual channel to channel offsets. We compared the simulated parameters with the measured ones.

5 Overflow mode: RUN281

5.1 Time distribution and Occupancy

The first distribution to look at is the time distribution and it is shown in Fig.2. The left one is uniform and it comes from channel 0 of the first FPGA, however the right one, that represents the time distribution of channel 2 of the second FPGA, looks non trivial.

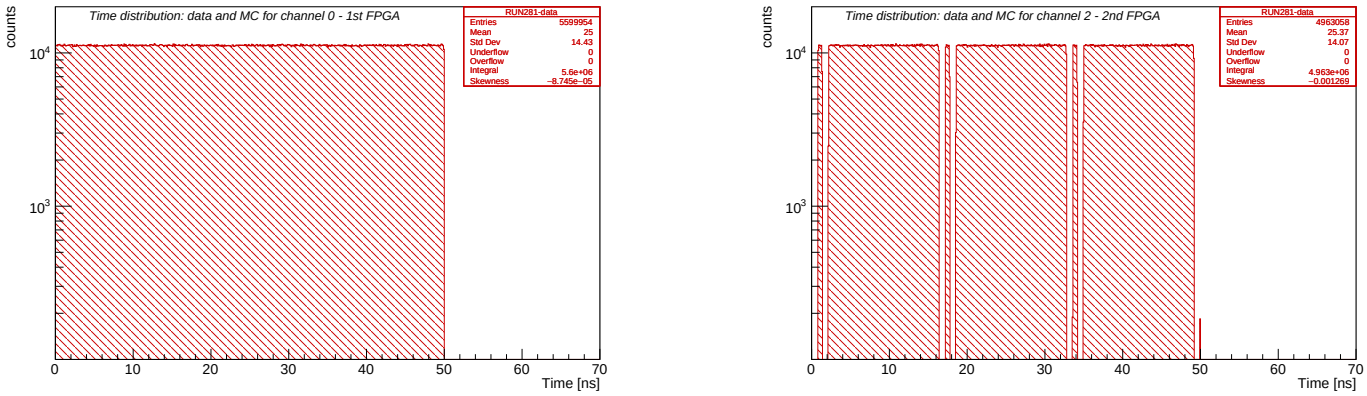

Figure 2: right: First FPGA's channel time distribution, left: Second FPGA's channel time distribution.

The initial though was the occurrence of an interruption in data acquisition for specific channels at a certain time. To understand the differences between the two distibutions we plot the occupancy, total number of hits versus channel number, in Fig.3. This revealed a non uniform distribution. We compared with the Monte Carlo occupancy. Channels are ordered as the filling of channels occurred. Actually, in

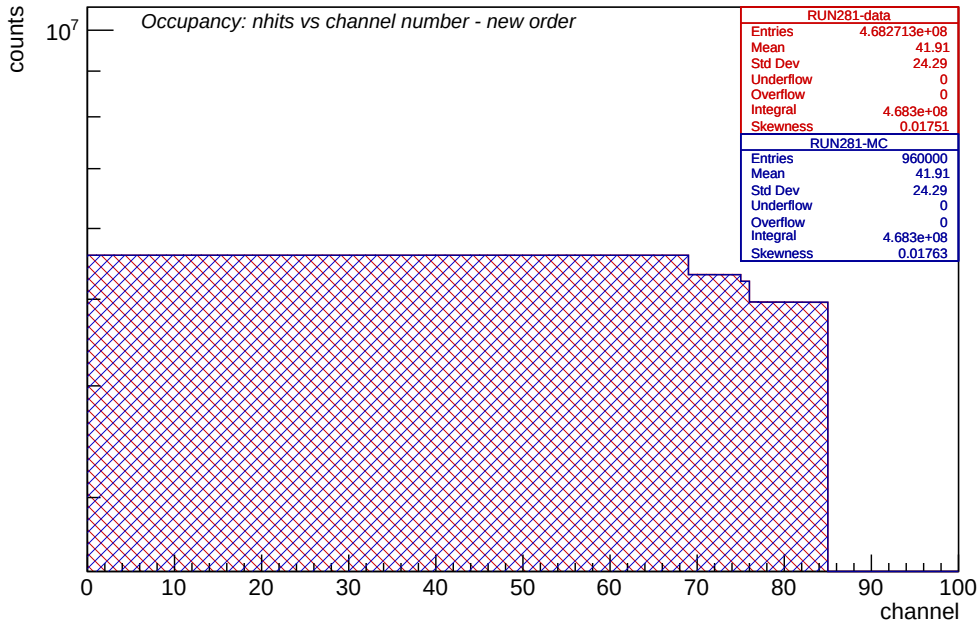

Figure 3: Occupancy: number of hits versus channel. The ordering of channels adheres to the sequence prescribed by the Monte Carlo simulation.

the first plots that were made, channels were arranged in an ascending order, spanning from channel 0 to

channel 95. As second step, the occupancy distribution was plotted with an alternative channel order, as we would expect the filling to occur. Our expectation was to observe a declining trend in the number of hits per channel, primarily due to the progressive filling of the buffer. However, contrary to our initial hypothesis, we encountered a distinctive pattern in which the number of hits exhibited a decrease until reaching zero hits at a specific channel, followed by a subsequent increase beyond the 72nd channel, which corresponds to channel 95. It was deduced that the initial given channel ordering was incorrect. The first and the second lanes of the second FPGA were inverted and also some channels of the second FPGA. Consequently, we have identified a revised channel sequence, as it is shown in Fig.3, verified also by the Monte Carlo simulation, which is as follows:

FIST FPGA:

lane 1:	91,85,79,73,67,61,55,49, 43,37,31,25,19,13,7,1, 90,84,78,72,66,60,54,48,
lane 2:	42,36,30,24,18,12,6,0, 93,87,81,75,69,63,57,51, 45,39,33,27,21,15,9,3,

SECOND FPGA:

lane 1:	38,44,5,11,17,23,29,35, 41,92,2,8,14,20,26,32, 86,80,74,68,62,56,50,47,
lane 2:	95,89,83,22,16,28,34,40, 46,53,59,65,71,77,10,4, 94,88,82,76,70,64,58,52

We can see a perfect adherence between data and simulation. We can interpret the occupancy plot, in Fig.3, as it follows: the initial group of channels in the histogram corresponds to the ones in which 4 hits (associated with the first FPGA) and 3 hits (associated with the second FPGA) are stored. Subsequently, a contrasting group of channels where the pattern reverses can be observed, 3 hits from the first FPGA followed by 4 hits from the second FPGA. We can observe a little jump in the middle and it is due to channel to channel time differences. Time differences between channels are three order of magnitude less than the FPGA ones, so these little jumps are very few, in our case only one appears. Lastly, the concluding cluster of channels is comprised of those collecting 3 hits from the first FPGA and 3 hits from the second FPGA. Coming back to Fig.2, some channels are always read and some others no, due to the fact that the ROC hit buffer gets filled up and only the first 255 hits are read out. This results in uniform time distribution for the first channels readout and in a non unform time distribution for the last readout channels, depending on T_{gen} and T_{EW}.

5.2 Number of hits

In conclusion, as in Fig. 4, a key characteristic of the overflow mode becomes evident: the number of hits is concentrated at the maximum value of 255 hits. Consequently, this results in all events possessing the same dimension.

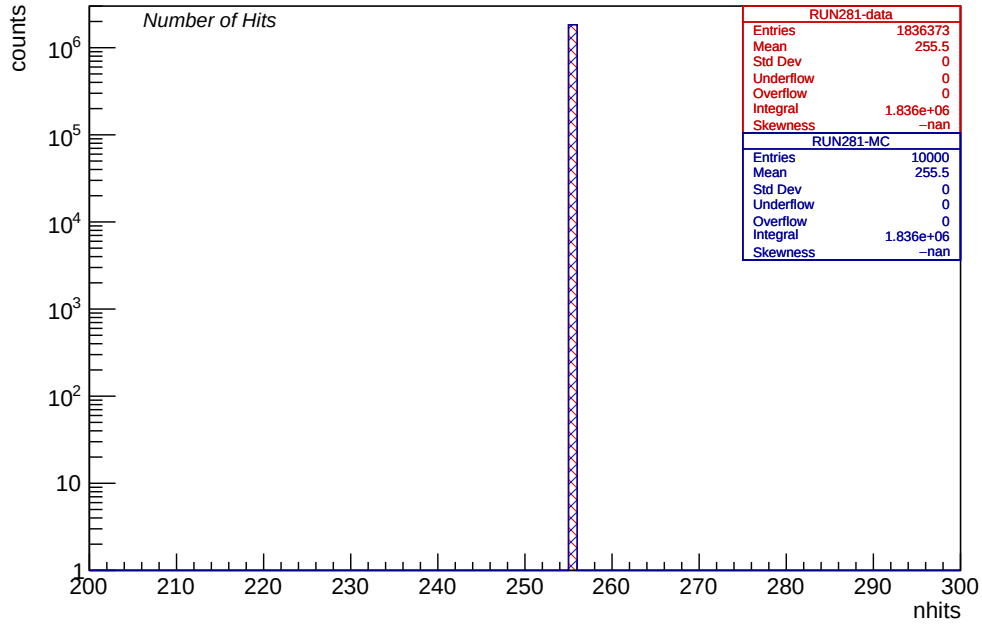

Figure 4: Number of hits distribution.

6 Regular mode: RUN105038

6.1 Time distribution

Similar to the approach outlined in Section 5.1, we commenced our analysis by investigating the temporal distribution of data for channels associated with the first and second FPGAs. Given that we are not operating in an overflow mode, our observations reveal a uniform temporal distribution for both cases.

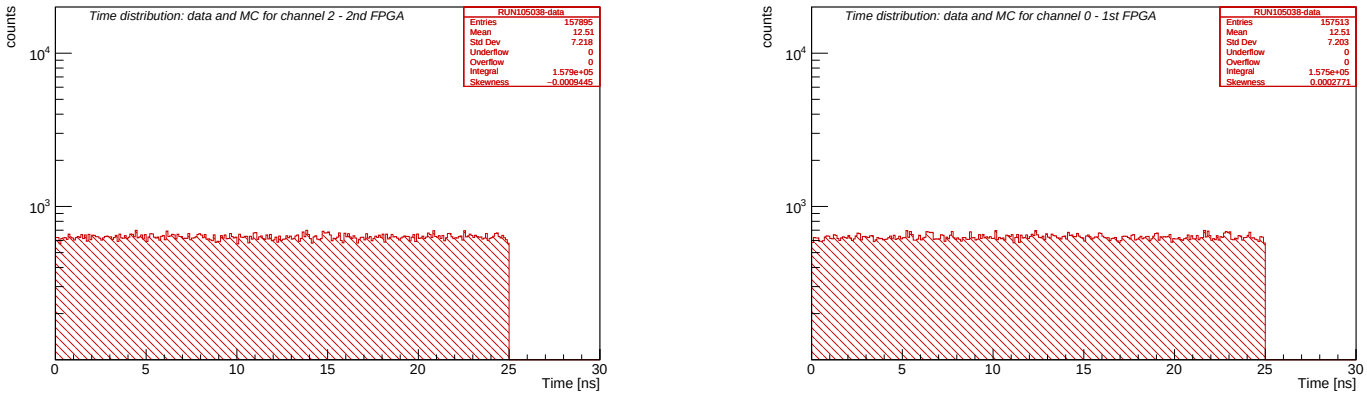

Figure 5: right: First FPGA's channel time distribution, left: Second FPGA's channel time distribution.

6.2 Occupancy: Number of hits versus channel

We conducted an analysis by reproducing the plot that shows the number of hits in relation to the channel number. The occupancy is a uniform distribution, primarily attributable to the fact that we are operating in a non-overflow mode. We can see a perfect adherence between data and simulation.

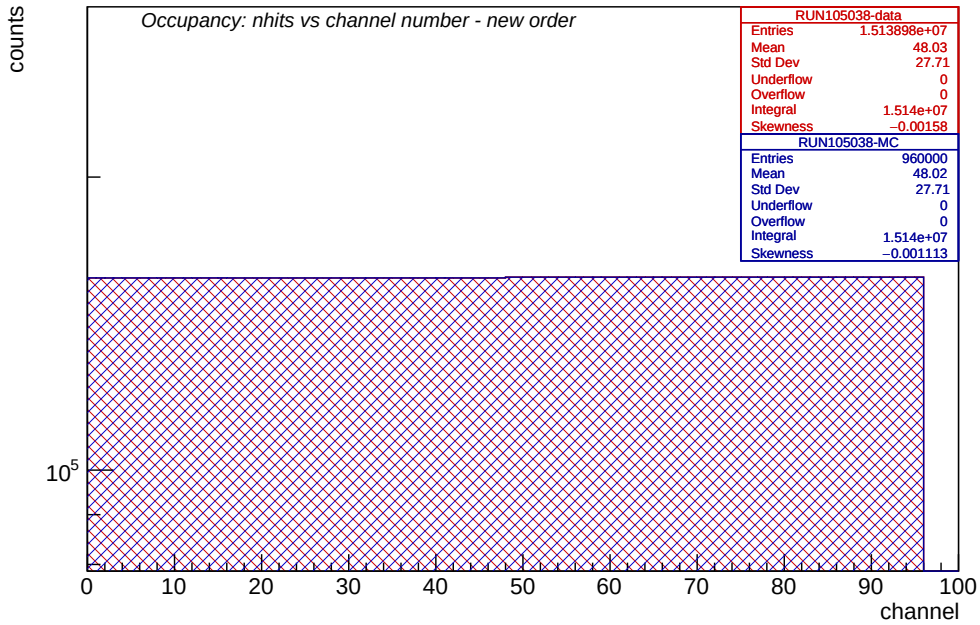

Figure 6: Occupancy: number of hits versus channel. The ordering of channels adheres to the sequence prescribed by the Monte Carlo simulation.

6.3 Number of hits

As conclusion we can see in Fig. 7, the main aspect of non-overflow mode: the number of hits are not anymore peaked in 255 and so this reflects in the fact that the number of bytes are not always the same.

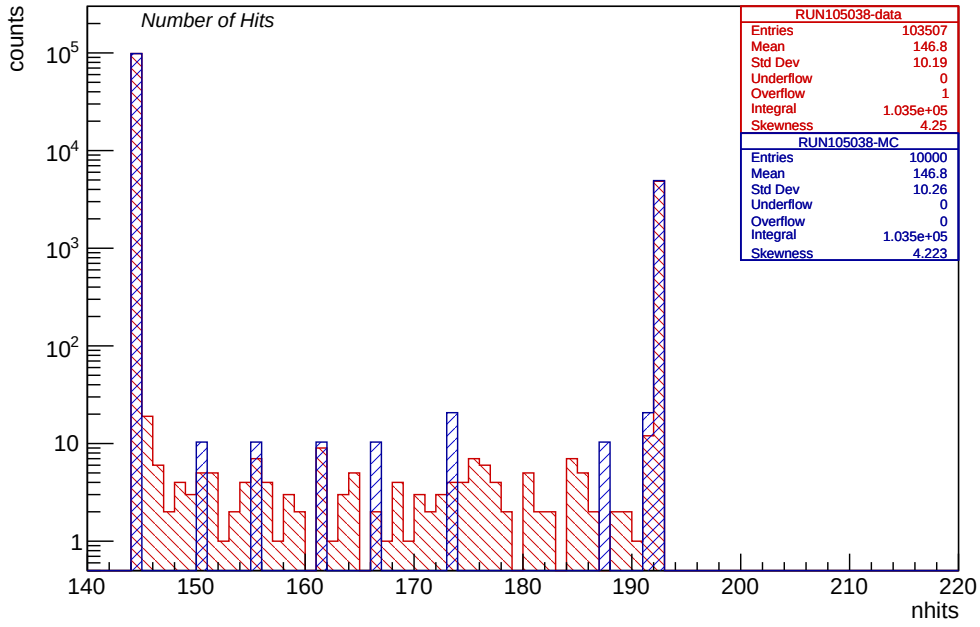

Figure 7: Number of hits distribution.

7 rate

8 Summary

Upper bounds on the direct beam-related backgrounds are as follows:

- background from beam electrons scattered in the stopping target < 1×10^{-3}
- background from muon decay in flights < 1×10^{-3}
- background from beam muons scattered in the stopping target < 1×10^{-5}