MU2E/PHYSICS/xxxxx
version 1.0 January 29, 2024

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

11. Mu2e ROC simulation

version 1.0 January 29, 2024

Abstract

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

Contents

1 Notes for the authors **2**
1.1 Revision history 2

2 Introduction to the analysis **3**

3 Description of teststand setup **3**

4 Monte Carlo simulation **3**

5 Overflow mode: RUN281 **5**
5.1 Time distribution and Occupancy 5
5.2 Number of hits 6

6 Regular mode: RUN105038 **8**
6.1 Time distribution and occupancy 8
6.2 Number of hits 9

7 rate **10**

8 Summary **11**

A Channels readout order **12**

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. In the first one the ROC emulates data itself without reading FPGAs. In the second mode the ROC reads digi FPGA. 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. 5. 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 pulse generator and pulse sequences from different generators are offset with respect to each other by a fixed Δt, fixed for the specific FPGA. 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. Different number of hits can fit in the event window, as we can see in Fig.5. 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 readout in a specific sequence that doesn't change in different events. The sequence is shown in App.8.

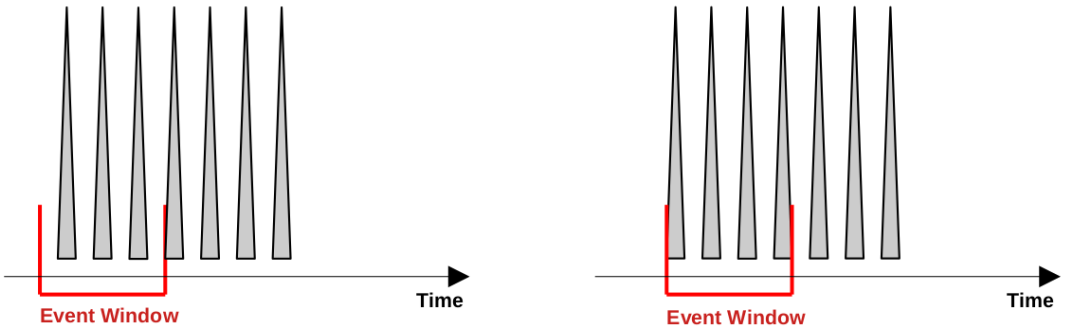

Figure 1: Graphic illustration of pulses in an event window.

4 Monte Carlo simulation

The ROC readout logic is purely digital, so the readout process can be simulated. The logic of the 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, we call $occupancy$ 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 timing of the first pulse is generated randomly from 0 to T_{gen}, by sampling 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};
- In each channel, pulses are generated in each channel following the readout sequence;
- After pulse generation, the readout continues until the count of hits reaches the maximum threshold of 255.

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 distributions to look at are the time distribution of hits in a specific channel and the occupancy distribution (total number of hits in function of the channel number). The timing distributions of hits in different channels are shown in Fig.2. These pictures show the timing distribution of hits in channel 0 of the first FPGA and in channel 2 of the second FPGA. The left one is uniform, however the right one looks non trivial.

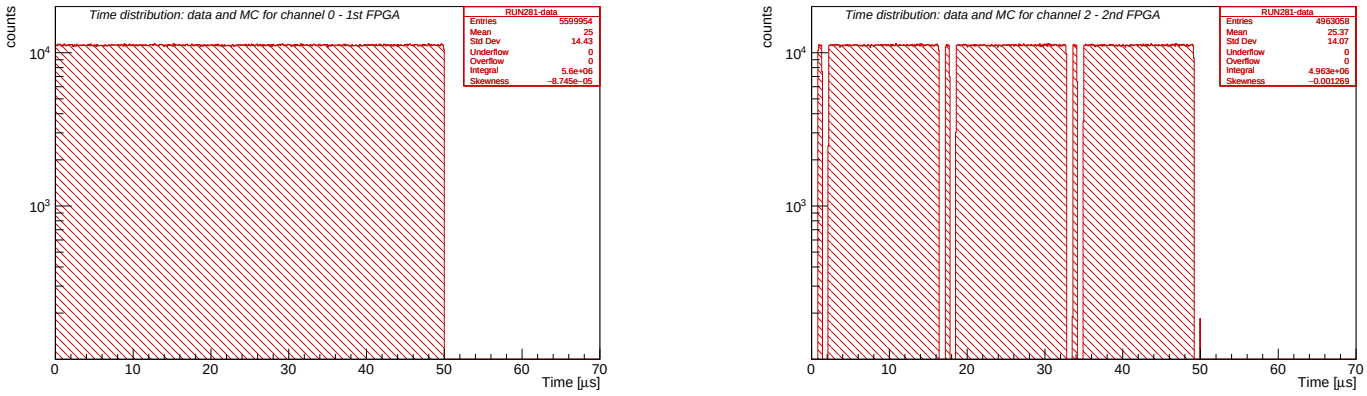

Figure 2: left: channel 0 (first FPGA) time distribution of hits, right: channel 2 (second FPGA) time distribution of hits.

The distributions in Fig.2 are easier to understand by looking at the occupancy plot in Fig.3 (left one). Channel ordering in this plot corresponds to the readout order. This revealed a non uniform distribution. We compared with the Monte Carlo occupancy.

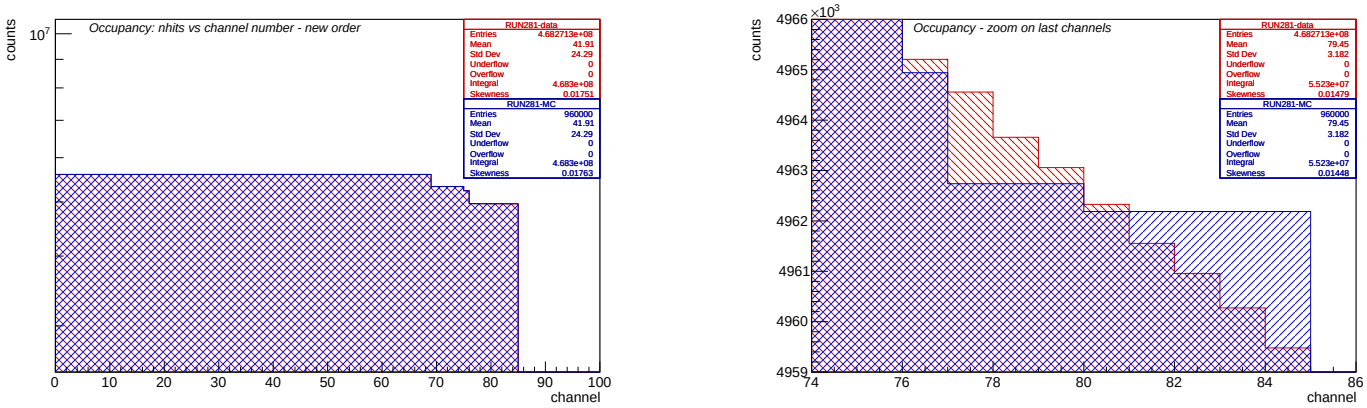

Figure 3: left: number of hits versus channel. The ordering of channels adheres to the sequence prescribed by the Monte Carlo simulation, right: zoom on last channels. The two histrograms differ from each other of a value $<10^{-3}$.

To understand occupancy plot, the number of hits per channel has a key role. In Fig.4 the number of hits (data) in channel 0 is shown. In this configuration there could be 3 or 4 hits per channel, as for the other ones.

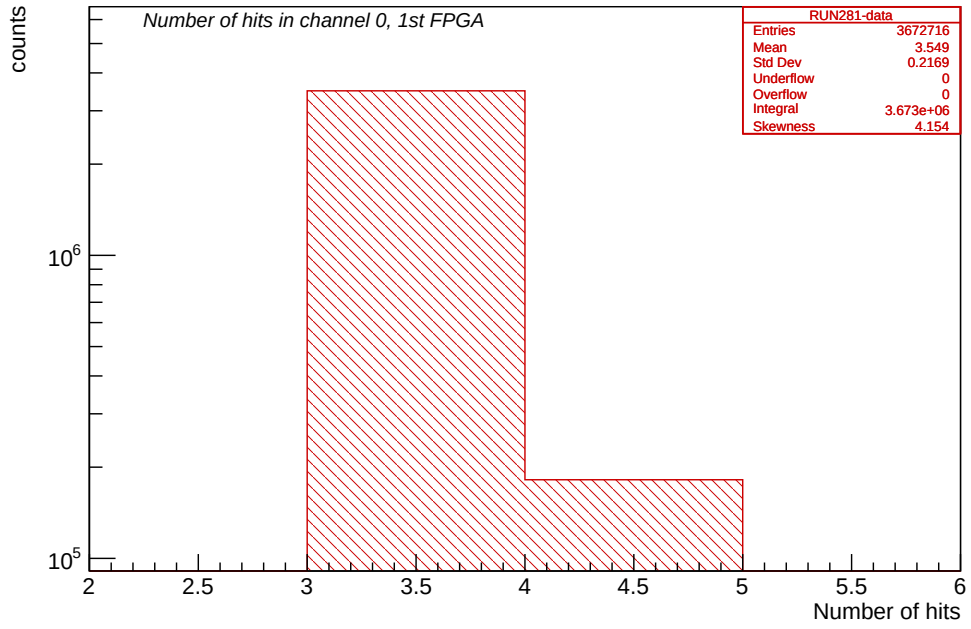

Figure 4: Number of hits per channel. It is shown that in this configuration there could be 3 or 4 hits in channel 0, as in other channels.

In the left picture of Fig.3, the first 68 channels are the ones with 4 hits in the first FPGA and three in the second FPGA, achieving in total 255 hits. The second plateau extending from 68 to 75 is composed by the channels with 3 hits in the first FPGA and 4 hits in the second one. There is a big step at the end of this plateau, because if we count the number of hits in the first FPGA we get 144, so in the second FPGA we have 111 hits in total, due to the fact that the maximum number of hits in total is 255. 111 is not divisible by 4, so the first 27 channels in the second FPGA will have 4 hits and the last one will have 3 hits. Last plateau consists of 3 hits from the first FPGA and 3 hits from the second FPGA. A zoom on last channels is shown on the right picture of Fig.3. The two histograms differ from each other of a value $< 10^{-3}$: this is a good agreement. The differences are due to the fact that each FPGA has its own pulse generator and pulse sequences from different generators are offset with respect to each other by a random number and also timing of generator pulses are uncorrelated with the beginning of the time window. Coming back to Fig.2, some channels are always readout and some others no, as we explained in the introduction the ROC hit buffer gets filled up and only the first 255 hits are read out. This results in a uniform time distribution for the first channels readout and in a non uniform time distribution for the last readout channels, depending on T_{gen} and T_{EW}. The deeps in channel 2 are defined by the differences between the two pulsers.

5.2 Number of hits

Fig. 5 shows that we are reading out 255 hits as expected.

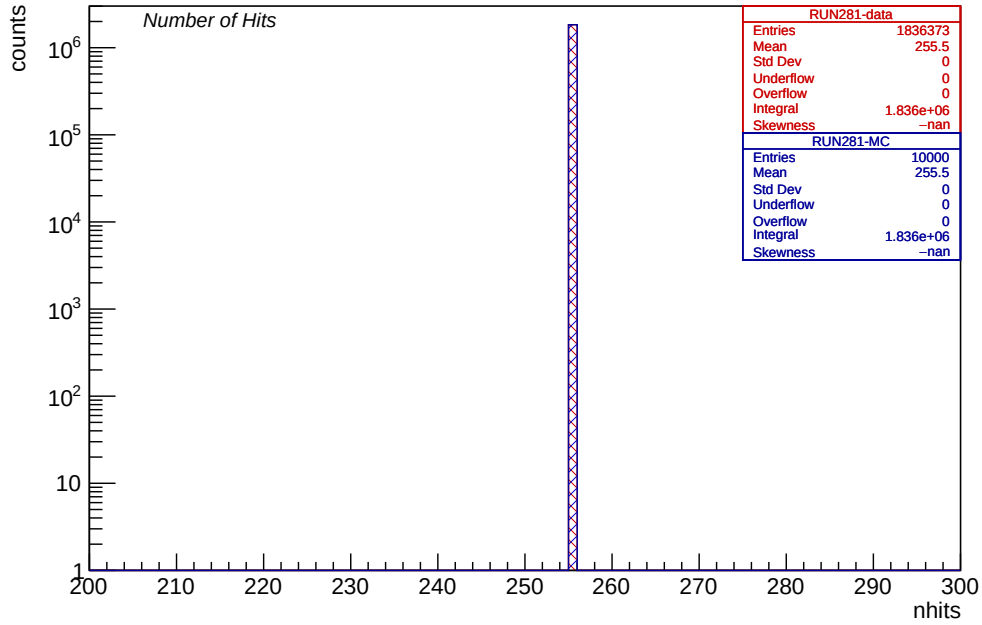

Figure 5: Total number of hits distribution.

6 Regular mode: RUN105038

6.1 Time distribution and occupancy

Similar to the approach outlined in Section 5.1, the first distribution to watch was the time distribution of number of hits in channels of the first and second FPGA. Given that we are not operating in an overflow mode, our observations reveal a uniform temporal distribution for both cases, as we expected.

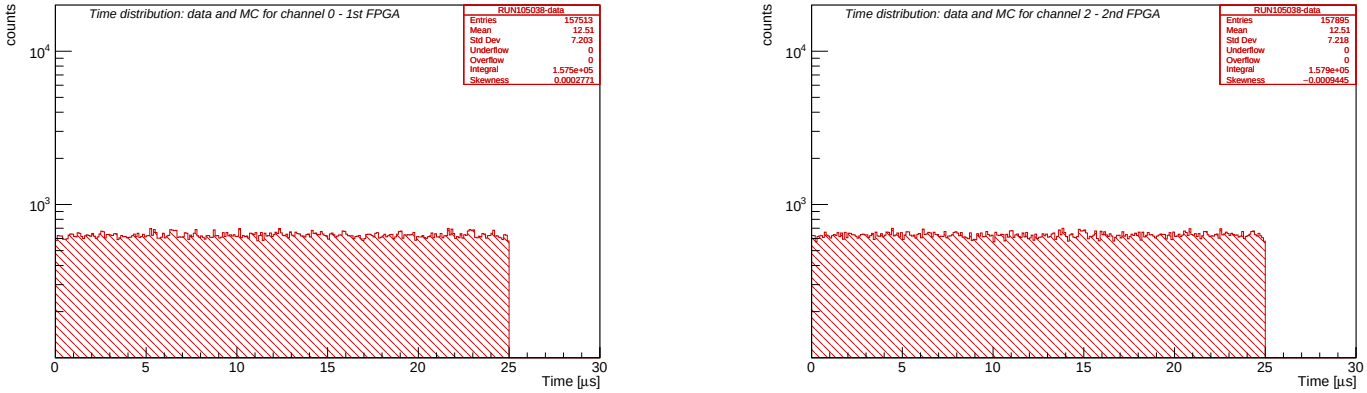

Figure 6: right: First FPGA's channel time distribution, left: Second FPGA's channel time distribution.

The time distibution can be easily understood by watching the occupancy plot in Fig.7. The occupancy is a uniform distribution, as we expected operating in a non-overflow mode. Channels ordering is the readout order.

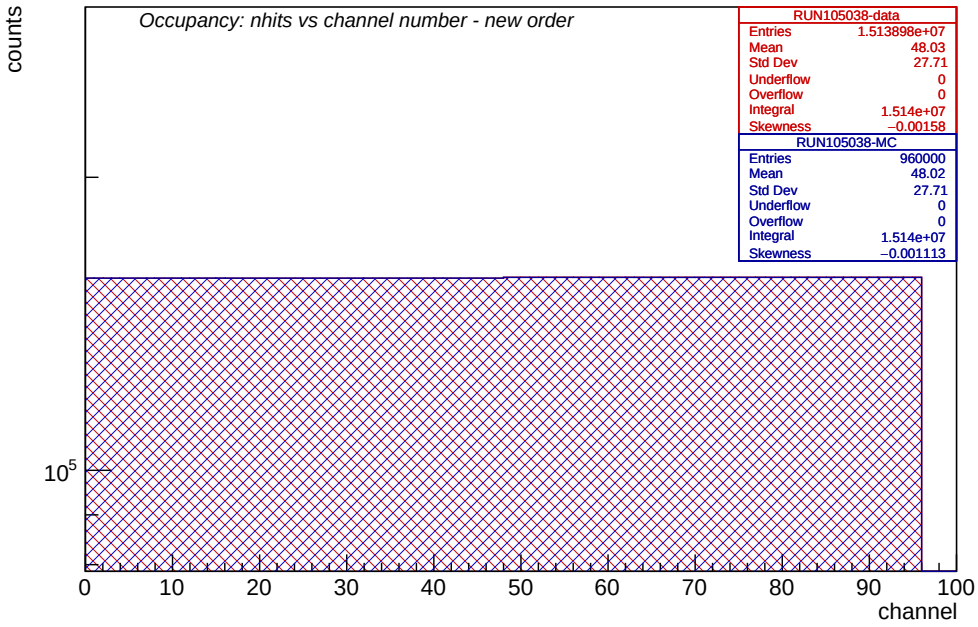

Figure 7: Occupancy: number of hits versus channel in the underflow mode.

In Fig.8, number of hits in channel 0 histogram is shown.

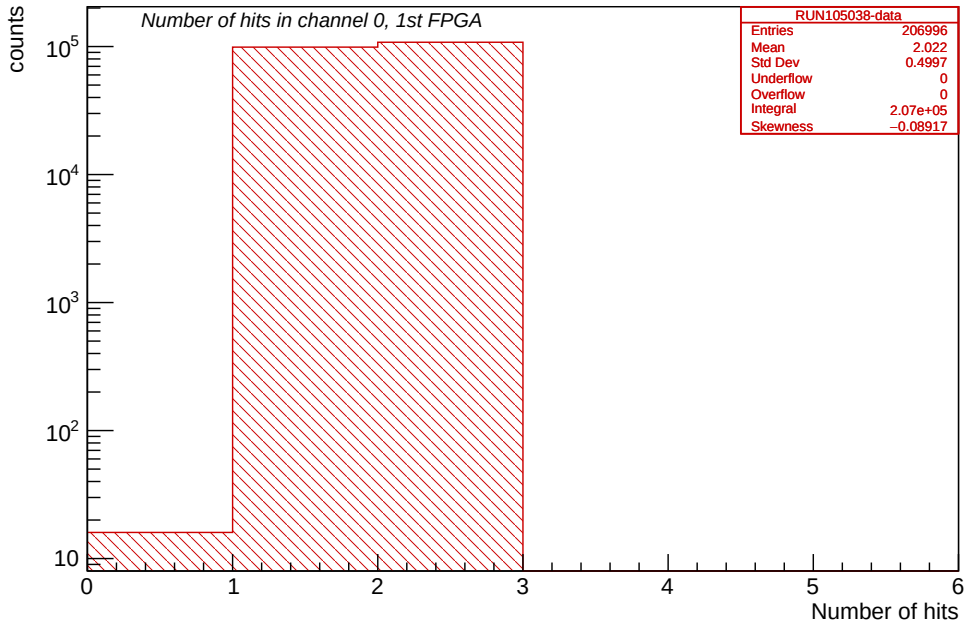

Figure 8: Number of hits per channel. It is shown that in this configuration there could be 0,1 or 2 hits in channel 0, as in other channels.

6.2 Number of hits

As conclusion we can see in Fig. 9, 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.

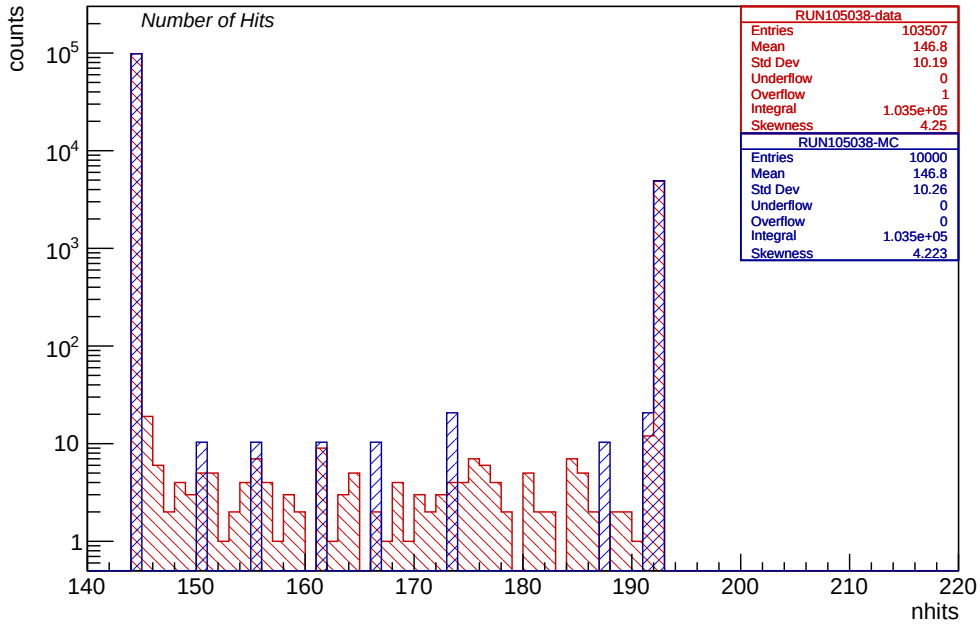

Figure 9: Total number of hits distribution.

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}

A Channels readout order

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