BGV MC Digitization

Status and plans

Plamen Hopchev

CERN BE-BI-BL

BGV meeting #27

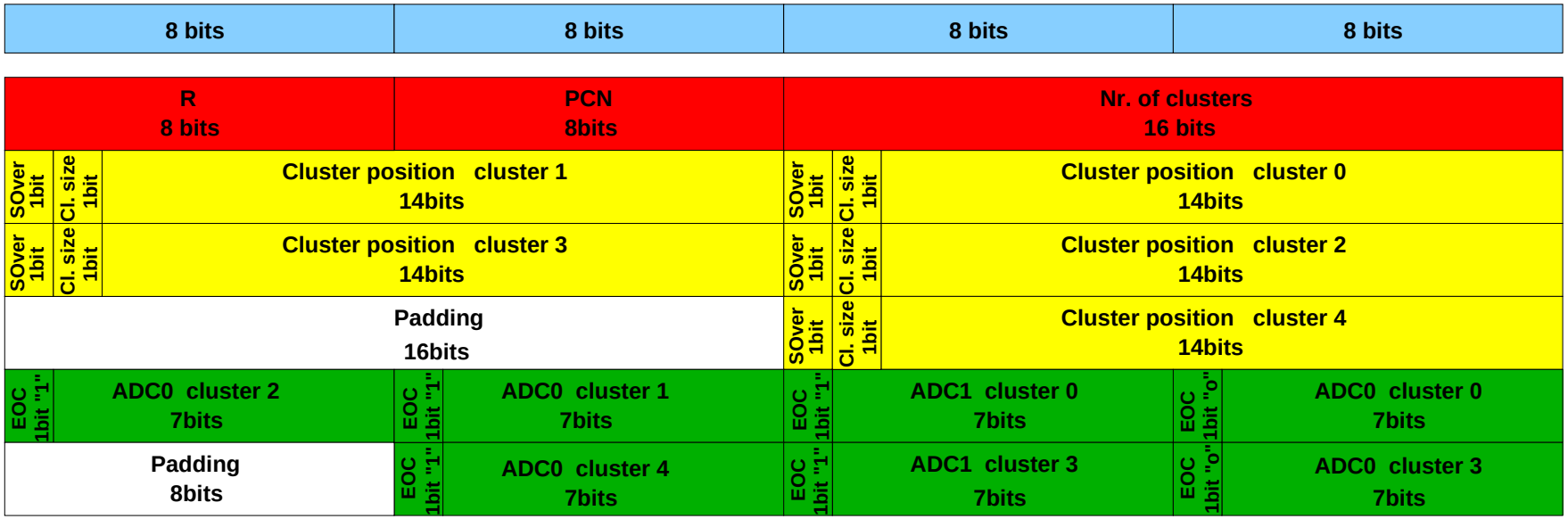
12 Feb 2014

Overview

- Some info about the LHCb SW model and data processing steps discussed at BGV meeting #23
- **In this talk, discuss considerations about:**
  - The BGV SciFi raw data format (TELL1 output)
  - Decoding the raw data into Event model classes (`LiteClusters` and `Clusters`)
  - Creating `Clusters` from simulated events, and Encoding into raw data format

BGV SciFi Raw data format (1)

- Use the Velo Raw Data format
  - Defined in https://edms.cern.ch/document/637676/2
  - Raw bank structure shown below
    - This is the data produced by **one** TELL1 for **one** event



LSB on the right, MSB on the left

BGV SciFi Raw data format (2)

Remarks about the raw bank structure and contents

- This format is efficient for the BGV SciFi data, because the Velo and BGV modules have the same number of channels
  - Encode cluster position in 14 bits: 11 bits for the strip ID (2048 channels) and 3 bits for the ISF (inter-strip fraction)
- The maximum cluster size is 4, encoded in 1 bit
  - `0`, when size is 1 or 2; `1`, when size is 3 or 4
  - Can distinguish 1-strip and 2-strip clusters using the ISF
- Allows quick decoding into clusters without ADC info (`LiteClusters`, useful for the SW trigger)

Raw data and Processing

Real Data Path

BGV $\xrightarrow{①}$ RawBank $\xrightarrow{②}$ Clusters

Simulation Path

MC Simulation $\xrightarrow{③}$ Clusters $\xrightarrow{②}$ RawBank $\xrightarrow{①'}$

1. Decode raw bank, create C++ (Event Model) objects
2. Event Model classes to store the BGV data
   - Allows the objects to be persistified (stored in a file)
3. Energy deposition in scint. fibers, light transport, SiPM QE, cross-talk, etc.

1'. Encode Clusters to raw bank, opposite to step 1

Raw data and Processing

Real Data Path

BGV $\xrightarrow{①}$ RawBank $\xrightarrow{②}$ Clusters

Simulation Path

MC Simulation $\xrightarrow{③}$ Clusters $\xrightarrow{②}$ $\xrightarrow{①'}$ RawBank

1. Decode raw bank, create C++ (Event Model) objects
2. Event Model classes to store the BGV data
   - Allows the objects to be persistified (stored in a file)
3. Energy deposition in scint. fibers, light transport, SiPM QE, cross-talk, etc.

1'. Encode Clusters to raw bank, opposite to step 1

- All this is implemented in LHCb. Copy and modify from Velo and SciFi
- **Initially, concentrate on items 1, 1', and 2**

Status and Plans

- Raw bank encoding / decoding algorithms
  - It is essential to use the existing Velo/ST algorithms!
- Class `SciFiChannelID`
  - Purpose: unique identification of all SciFi channels (in all modules)
  - Velo implementation
    - `type + sensor + strip` (the last is in the raw bank, provided by the decoder)
    - Not optimal for BGV (e.g. don't need type), but can be used
  - Intention: 1-to-1 copy, replace Velo with SciFi
    - Could be re-implemented in the future, consider better granularity (matt ID, SiPM ID, etc.)
- Classes `SciFiLiteCluster` and `SciFiCluster`
  - Velo implementation
    - `LiteCluster = ChannelID + size + ISF + SO bit` (the last three are in the raw bank, provided by the decoder)
    - `Cluster = LiteCluster + ADCVector` (the ADCs are in the raw bank, provided by the decoder)
  - Intention: 1-to-1 copies, replace Velo with SciFi
    - No envisaged modifications for the moment

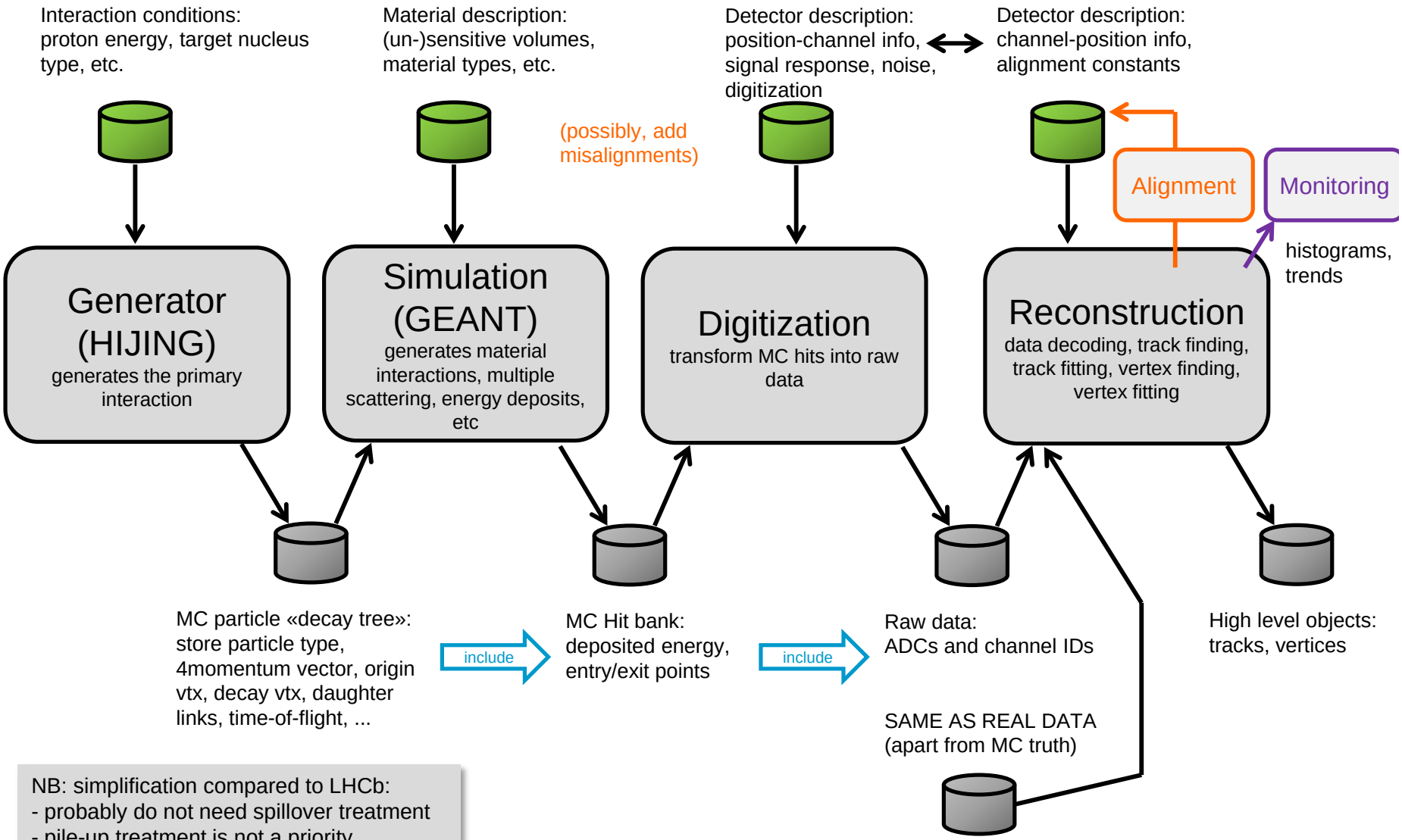
Status and Plans ctd.

- Can use Velo raw data banks (in LHCb raw data) to test and compare the raw data en/decoding with the new BGV SciFi algorithms and Event Model classes

- Current progress
  - Implemented `SciFiChannelID` and `SciFiLiteCluster`
  - Decoding test successful
- Next steps
  - Implement `SciFiCluster`, more decoding tests
  - Try encoding, make tests
  - Implement a simplified digitization, i.e. don't consider effects like light attenuation, SiPM QE, cross-talk
  - Produce raw files

Additional Slides

«particle physics» software chain



NB: simplification compared to LHCb:
- probably do not need spillover treatment
- pile-up treatment is not a priority