VELO/ST event model review discussion

First discussions between reviewers and informaiton providers, Chris Parkes, Matt Needham, Juan Palacios, 11/8/2005

MCHit

All agreed identifying which material a hit is in by its x,y,z position is best avoided. Velo currently does this by adding an extra word with sensor ID. Suggestion to use instead key of keyed object to store a material identifier in some bits + hit number in other bits. The use of this information in the construction of ChannelIDs should be configurable, ensuring de-coupling of MCHits with detector numbering schemes. This solution would do away with the specialised MCVeloHit. Should be adopted LHCb wide. How to identify active volumes uniquely in XML and pass the information to Gauss is under discussion. Use of Geant4 copy numbers has been considered and tried in the past, and is deemed not to be robust.

MCITDeposit, MCITDigits, MCVeloFE

IT has extra stage MCITDeposits have one to one corresponance with MCHits, these are then summed on individual strips to give MCITDigits. Velo goes directly to summed strip level MCVeloFE. Either approach fine, no need to standardise.

MCITDigits are in a form of ADC counts (but not final scaling form) MCVeloFE are in electrons. Not a major issue as this is an internal state for subdetector, but Velo decision of electrons is maybe more transparent. ST agrees to investigate going back to electrons.

ITDigits, VeloFullDigits

Both serve two purposes:

- Storage of digitised signals in MC simulation. Here, no zero suppression has been performed. This data does not have pedestals, common mode subtraction, etc..
- Storage of non-zero suppressed data (or MC emulation of these). This

is (or emulates) real data so does have pedestals, common mode etc...

Two different approaches - nobody has strong pref. as to which is better.

- VELO: VeloFullDigit has space for raw ADC, corrected ADC, CM noise,

pedestals. Object would be created once and passed along through TELL1 emulation having corrections added.

- ST: ITdigit: Contains only ADC. New ITDigits would be created for each stage in the

TELL1 emulation, with change in name of container. *Question to framework* - Can the previous containers be deleted if not needed ?

May be useful to standardise since FPGA operation likely to be very similar for two detectors.

L1Buffer / Raw Buffer HLT

On-going discussion on cluster format. Will change from what is currently in Gaudi. Attempt to standardise between L1 / HLT formats and Velo / ST.

Question (Matt): Is backwards compatibility needed. I (**Juan**) think not. Depends at what level. L1/HLT buffer decoding would need to adapt, unless persuaded to use cluster classes instead of accessing buffers directly.

Header info. is an issue. IT has PCN and error bits stored. Velo currently doesn't store anything in Gaudi, but had commented that it may need PCN depending on FPGA studies.

Recommendation - add an event summary block (LHCb wide ?) - for Velo and ST this would contain PCN and error bit info. from PCN matching and beetles.

PCN header info. for Velo and ST look very different in format, why ?

See TELL1 algorithms for VELO page.

Raw Buffer - non-zero suppressed

Special data format for Velo/ST proposal in internal LHCb note draft. Currently being implemented in Gaudi for Velo.

Still to be determined - can we send the data for an individual event twice (both zero-suppressed and non-zero suppressed) ? All agree that this is important for debugging.

Cluster

Both IT and Velo expect to perform reclustering across boundaries offline, producing Cluster objects from Raw buffer. IT chooses to unpack to digits first then recluster. Velo assumed it could join existing raw buffer clusters, but hasn't tried it...

IT stores cluster position in offline cluster. Tools already available to return re-calculated cluster position and error.

Velo stores only ADC and plans (but hasn't written) tool to return cluster position and error. This tool would have weighted cluster position, eta fit and track angle using variants. Preference for more flexible clustering tool approach.

Note: though L1, HLT clusters will probably use 2/3 bit between strip cluster position calculated in TELL1. **Question**: Should standard IT and VELO clusters make this information available? Matt worry: risk users would never use the tools to obtain better estimates of cluster position and error. Juan: Maybe making it clear that this is L1 limited precision in method name and making use of refined tools east would avoid this.

Decoding of data

IT and Velo both express worries about how cluster decoding from L1/HLT bypasses standard cluster classes. This is because 'new' operation of many cluster objects is considered too time-heavy for HLT. **Request to framework** - Can this not be solved for all by allocating the memory in a more efficient manner? Can larger objects (containers of clusters?) be stored instead?

-- JuanPalacios - 16 Aug 2005

This topic: LHCb > VeloSTEventModelReview
Topic revision: r3 - 2005-08-29 - unknown