



Towards Electronics Upgrade Simulations

Tomasz Skwarnicki

Outline

- Remind people what standard LHCb Gauss & Boole do
 - one thing to keep in mind is that these are oriented towards physics simulations and changes are needed in Boole to make these simulations useful to electronics people.
- Outline present plan to “generate” input data to upgrade electronics simulations.
- Open issues.
- Discussion?

Gauss – physics and particle interactions with the detector

For each event generate 4 bunch crossings.

For each of 4 bunch crossings $XX = \text{"/PrevPrev", "/Prev", "/", "/Next"}$

generate number of minimum bias interactions according to Poisson statistics and dialed luminosity.

For each p-p interaction call Pythia to generate initial state particles. Trace them through GEANT4 model of detector geometry, allowing for interactions in the detector material (secondary particles).

Intercept position and momentum vectors of each particle passing detector volumes declared as sensors. Save this information at:

`/Event/XX/MC/detector_name/Hits` (container of `LHCb::MCHit` class).

This is essential part of output of Gauss – “.sim” file.

- Gauss is extremely CPU consuming especially for high luminosities because of large number of p-p interactions generated for each “event”.
- For minimum bias MC (used for electronics simulations) each of 4 bunch crossings is of the same type in spite of different names. They are also completely independent of each other. Asking for 100k events, we get 400k bunch crossings.

Boole – electronic response simulation.

Use silicon detectors as an example.

Read in MCHits from /Event/XX/MC/detector_name/Hits (XX = “/PrevPrev”, “/Prev”, “/”, “/Next”) from .sim files.

For each MCHit (i.e. particle passing silicon sensor) generate total charge in electrons. Spread it to individual channels.

Sum up all charge contributions for each electronic channel. This is done separately for each bunch crossing.

Generate spill-over. Use a model of pulse shape (rise & fall). For Upgrade simulations assume bunch crossings are separated by 25ns i.e. assume the charge peaks at -50ns, -25ns, 0ns, +25ns for “/PrevPrev”, “/Prev”, “/”, “/Next” respectively.

Sum up “on-time” and “off-time” charge contributions (output is one charge per channel per event).

Simulate electronic noise. Pulse discrimination. Pulse digitization.

Simulate clustering in Tell boards. Form RawBank in format equivalent to what is coming out of real detector. These RawBanks are essential part of Boole output.



This part depends
on FE electronics!

Using Gauss for upgrade electronic simulations

- Since Gauss is so slow it must be run on grid by LHCb Production team. All software & geometry changes must be released in official Gauss release and geometry data base.
- Because of limited CPU & storage we need to consolidate Gauss production requests as much as possible.
- Gauss needs to contain fair representation of your sensors and of overall material budget in all detector parts (secondary interactions!):
 - If you need to redefine existing geometry, you must provide changes to xml description of your detector. We will help you to release such changes in upgrade DDDb for use in MC production.
- Recently new version of Gauss & geometry was released which contains VeloPix L-shape modules. Other detectors are unchanged except for lighter beam pipe support in magnet and swapped IT/OT positions. Production of 200k events ($\times 4 = 800k$) at 2×10^{33} has been recently completed: **1 TB of .sim files!** Presently under /MC/2010/ folder in book-keeping, should be migrated to /MC/Upgrade/ (look for Beam7TeV-UpgradeVeloPix2.0-MagDown-Lumi20-25ns data).

Using Boole for upgrade electronic simulations

- Running standard Boole in production is useless since for most of detectors:
 - FE electronics will be different
 - Even for VeloPix, present Boole contains “naïve” simulation of electronics
 - Info needed by electronics people is often not saved in .digi files.
 - CALO, MU exceptions?
- Fortunately Boole is fast! You should be able to run your customized version of Boole yourself (you will still need to run on grid to analyze 1TB of sim files). No need for public release!
 - Example of how this is handled for VeloPix: Victor runs private version of Boole, which intercepts charge in electrons in each pixel (not saved in .digi files!) for every bunch crossing. Writes this into ntuple file which he passes to Toumas. Toumas runs his stand-alone electronics simulations.
 - You need to find manpower in your subdetector group to do it.

Open issues

- CALO people asked for Gauss with one p-p interaction per bunch crossing. They want to assemble multiple-interactions per bunch crossing themselves:
 - Doable without software development. However, this means special Gauss production for CALO people. It would be good to avoid it!
 - Please notice that you can introduce any bunch-time structure you want in Boole (Gauss actually knows nothing about bunch crossing spacing).
- Other luminosities?
 - 2×10^{33} is a reasonable stress-test of your design. Before we generate other luminosities I would like people to learn how to use the existing .sim files and make a compelling case you need other luminosities.