VELO upgrade simulations (overview)

Tomasz Skwarnicki

Goals

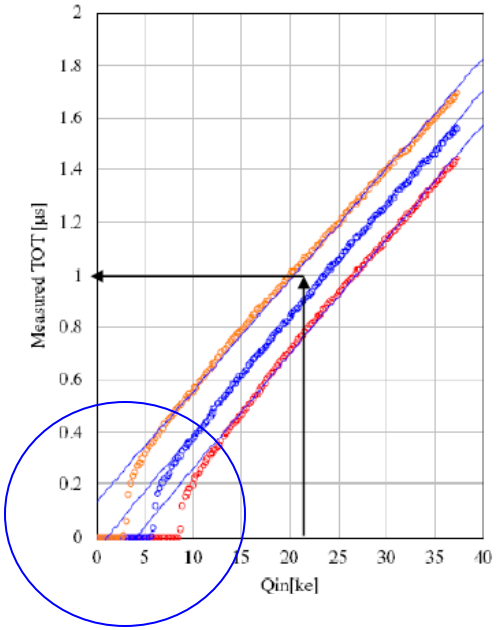
- Aid design of new detector:
  - This task has mostly defaulted so far to simplified simulations:
    - Last year we saw a large number of various studies by Glasgow, CERN, Syracuse and Warwick
    - Most recently extensive studies by Steve Blusk using improved version of GEANT3 code (originally used by Marcin Kucharczyk) – **see Steve's presentation today**
  - Some input from full simulations (=Gauss/Boole/Brunel):
    - e.g. pixels hits (prepared by Victor Coco) passed to simulations of read-out chain (see Tuomas Poikela presentation last week)
    - Full simulations are reaching a stage in which track level and even some physics studies are becoming possible:
      - Expect them to play a bigger role in the future
      - Necessary to validate design worked out on the simplified simulations
- Prove Super-LHCb capabilities to reach the advertised physics reach:
  - **Feasibility of triggering with 40MHz readout**
    - including improved trigger efficiency for hadronic final states (factor of 2 improvement with respect to the present detector !?!?)
  - Physics reach simulations based on MC integrating all subdetectors
  - Exclusive domain of full simulations. **VeloPix software on critical path.**
  - Time scale is clearly TDR not LOI. However, given scope of the work to do the timeline is very tight.

Status of Gauss

- Flexible Detector Element code was developed by **Victor Coco**. In public Gauss releases since v38r0p1 (Dec.09). Documented on the web.
- Xml detector description:
  - Previous work:
    - Active elements described by **Victor:** U-shape option sketched out by Paula last July.
    - Inactive parts added by **David Dosset** based on **Mark Whitehead** xml version for strips (both from Warwick).
    - **They are fine as placeholders allowing further software development, however they do not reflect present thinking about detector design** (likely to be L-shape based, need to get more serious about cooling design and its effect on material budget).
  - Future work:
    - New strawman design needs to be coded in xml once we agree on a version for LOI
      - Opportunity to assume responsibility for it (it will default to Victor if nobody steps in)
    - Need somebody to look at material budget issues:
      - Radiation length maps (Warwick ?)
      - Look at consequences of inactive material on occupancies in VELO and downstream detectors

Boole

- Naïve implementation by Marcin Kucharczyk in public releases since February (v21r0)
  - idealized TIMEPIX performance:
    - No timewalk, essentially no effects from hits in different bunch crossings
    - Linear ToT, no noise. Thomas Britton (Syracuse) is learning how to change it.
    - All pixel hits shipped out without losses, no simulation of readout chain



Neglected so far

- Implemented cluster concept is geared towards physics performance (local charge maximum) rather than electronics realities

– Hit resolution similar to the one obtained in the testbeam.

– It can be used to obtain “upper limit” on physics performance

– Sets framework and reference for more realistic simulations

Boole – readout chain

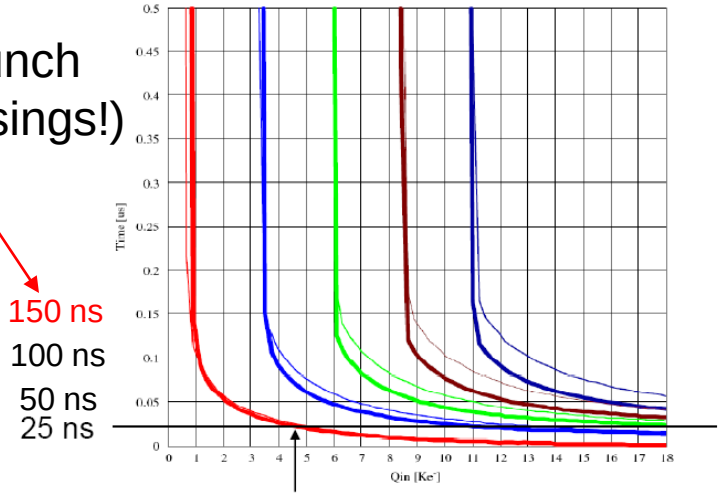
- Crucially important for the chip & detector design
  - **Tuomas** is simulating it using dedicated package (Verilog) – see his talk last week
  - There will be some loss of information due to bandwidth limitations on the detector itself
- Need to simulate it in Boole and reconcile with the trigger needs:
  - Speed of tracking is of critical importance for trigger. Unpacking of tracking hits is a big deal - LiteClusters!
    - Cluster info saved in one machine word, with speed optimized access functions.
    - LHCbID of the strip/pixel corresponding to the cluster's **center-of-gravity** plus fractional position within (encoded in a few bits)
    - Present clustering scheme in Boole implements them in a naive way: LiteCluster = local charge maximum (check nearest neighbors – 3x3)
  - Readout scheme (Tuomas) based on a 4x4 "super-pixel":
    - Not a clustering in the sense of finding charge maxima and center-of-gravity position
    - Means of shipping hits out of chips in bandwidth efficient way.
    - Merging groups of hits across super-pixel neighbors (in one direction only)
  - Need to go from readout "clusters" to LiteClusters.
    - In Tell40? I can't imagine we will have time to do it in HLT.
    - Simplest implementation would be to find center-of-gravity for the readout cluster (1-to-1 correspondence). This is not optimal from physics performance point of view but might be OK.
      - 4x4 super-pixel may have up to 4 local maxima
      - Loss of information across super-pixel boundaries in non-merged direction.
- Perhaps not the most pressing item to implement in Boole but eventually needs realistic implementation. Opportunity to contribute.

Boole – time dependence

- Timewalk, deadtime etc. not yet known for final chip design but this should not stop us for exploring impact of these issues on physics performance already now:
  - Time walk (time when threshold is exceeded increases with decreasing pixel charge – effect as large as several bunch crossings)
  - Long deadtime in Time-over-Threshold measurement (raise time ~90ns, fall time 500-2500 ns)
    - Doing proper simulation of deadtime will be highly non-trivial (we can't afford generating 100 previous bunch crossings for one signal event; library of min.bias. events? parametrizations?)
- Opportunity to step in to implement these effects

TIME WALK

(6 bunch crossings!)



ToT MEASUREMENT

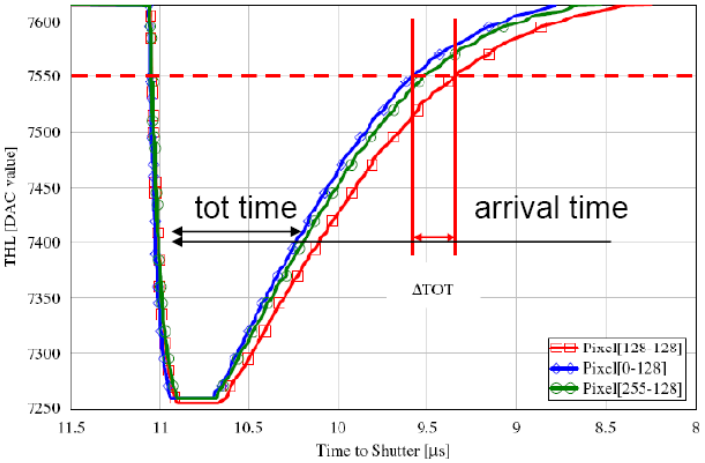


Fig. 9.

Boole – radiation damage

- Impeding charge spread to neighboring pixels - potentially big effects on hit resolution,
- Dangerous to neglect it in detector design.
- Understanding how radiation damage effects charge collection is a non-trivial issue. We need somebody to explore this both intellectually (literature?) and in simulations (impact on hit & track resolution, impact on trigger and physics performance).
- JC is going to work on this.

Brunel

- Stepping stone towards evaluation of trigger & physics performance
- Pattern recognition – needs to be efficient and **fast (trigger!):**
  - Initially bypassed via cheated pattern recognition
  - Non-trivial algorithmic development to be done
  - See talk by **Laurence Carson** (USC) today
- Track fit:
  - Substantial work on C++ classes to integrate VeloPix into existing tracking software
  - See talk by **Victor Coco** today
- Brunel configurable:
  - Victor has a version which integrates VeloPix tracks with forward tracking
  - More work is needed (volunteers?) for fully functional Brunel application:
    - L0 was removed from Boole. This is causing problems in Brunel.
    - DST output, including MC truth
    - VeloPix Brunel monitoring

Other work

- Recent progress on Brunel enables new types of studies:
  - Naïve physics performance studies:
    - Primary and secondary vertex resolutions
    - Some benchmark physics analysis (efficiencies after background suppression, mass resolution etc.)
  - Trigger software without timing studies (in preparation for the latter)
- Impact of VeloPix designs on downstream detectors (e.g. occupancies in T-stations)
- Not subscribed yet – opportunities to contribute

Comment on manpower

- Over recent months only Syracuse and USC made serious commitment of manpower (postdocs) to Work Package 7
- Warwick has made minor contributions.
- Many other groups have signed up for Work Package 7 but not ready to contribute
  - Not completely surprising given their commitments to the present VELO detector
  - Other important contributions to the VELO upgrade
  - We all want and must do physics analysis to maintain funding…
- Lots of opportunities to contribute and have your own responsibility in VeloPix software development.
- Given the manpower situation developing software for the backup micro-strip option is on a back burner. We will have to play a catch-up game if the TIMEPIX option folds.

Conclusion

- Steady progress but still a long way to go …