Simulation study of the chronopixel sensor in CLIC

N. B. Sinev
University of Oregon, Eugene
Outline of the talk

- About pixsim package
- Is fast version accurate?
- Is chronopixel fast enough for CLIC?
- Do we need analog readout?
- Sensor point precision and detector impact parameters resolutions.
- Chronopixel design for CLIC environment.
- Use of pixsim package with seed tracker.
- Conclusions
Sensor response simulation in lcsim track reconstruction

- In the lcsim package we start from the simulated by Geant ionizing particles hits in the active layers of the detector. The result of such simulation is recorded in SimTrackerHit objects. However, to be able to investigate different vertex detector options without re-running Geant, we are not using energy losses, given by Geant, in the sensors. We are setting active layer thickness and calculating energy deposition in such layers during event reconstruction in the lcsim.

- Next step is to simulate signal generated in the sensor and it’s processing by sensor electronics. Here we are dealing with pixels dimensions, electronics noise and processing of the resulting active (fired) pixel clusters to reconstruct the position of the hits the same way it would be done in real data processing. Results are saved in TrackerHit objects.

- After that tracking code can deal with found TrackerHit (s) to reconstruct tracks.
Detailed simulation of a pixilated sensor

- To understand effect of different technical solutions on the vertex detector performance as a part of ILC detector, I have developed pixilated sensor simulation package (pixsim). This package is a part of org.lcsim package which is a reconstruction and analysis package for simulation studies for the international linear collider developed by a group of SLAC scientists. It allows detailed simulation of signal formation in the silicon pixel detector and it’s processing by front end electronics and reconstruction software.

- The use of full simulation of the sensor response can help us to see possible problems, not noticeable with fast simulation which uses general detector performance parameters. (For example, in the dense jet hits in the vertex detector may be too close to separate them, though their distance is few times larger than single point precision of the sensor).

- But major application field for pixsim package is the evaluation of different sensor designs and selection of optimum design parameters.
What and how was simulated

Carriers generation in Si along charged particle track.

I am using algorithm, developed by H. Bichsel, to calculate energy transfer from ionizing particle to electron in the single ionizing collision. It also gives me average path between such collisions in Si (it is of the order of 0.5 µm). Left picture shows example of energy loss distribution in 1 µm of Si. Right picture shows distribution shape comparison with real VXD3 data for 20 µm epi thickness. Distributions are normalized to peak position and height.

Nick Sinev for WG4 meeting, April 7, 2011
To simulate charge carriers movement inside silicon, we need to know electric and magnetic fields there and silicon parameters, such as carriers mobility, Lorentz angle, diffusion constants. To find electric field, I am using Tcad (dessis) simulation, same as used by semiconductor devices developers. Pictures above illustrate such simulations. On the left is doping concentration distribution. This is simulation input. On the right – electric field – this is output. Depleted regions are outlined with white line.
Illustration of charge carrier movement simulation. On the left diffusion was turned off, on the right – real movement, which includes diffusion. Background color shows the strength of electric field. Magnetic field of 5 Tesla is perpendicular to picture plane. (picture is upside down compare to previous slide).
How to do it faster

- Described above process is very slow, simulation of one charge particle track takes about 20 seconds.

- To do it faster, we can “pre simulate” fate of charge carrier, generated in the particular point inside sensor. We can find out, what is the probability of such carrier to be collected by charge collection electrode of one of the pixels around that point. And record this probabilities for every point inside pixel into lookup table. More complicated is the process of simulating of time, spent by carriers on that path. Apparently we can’t just use the average time. So, I record 2 parameters of arrival time distribution, and in fast simulation try to generate random numbers having similar distribution. Next slides show how close are results, obtained with full simulation of carrier movement and with simulation using lookup tables.
Comparison of full simulation of carrier movement with table use simulation

On the right plot you can see carrier travel time distribution for carrier generated at some point inside sensor (x=0, y=3µm, z=12µm). Magenta corresponds to full simulation, green is for simulation using lookup tables.

On the right is the comparison of cluster size (number of fired pixels) distribution for hits, caused by charged particle, crossing 18 µm thick sensitive layer. Red is for full simulation, blue – using tables.

Nick Sinev for WG4 meeting, April 7, 2011
Summary about simulation

- As we could see, use of lookup tables to simulate charge carrier collection process provides results close to simulation using calculations of carriers movements.
- For all results, reported below I will use simulation with use of lookup tables, and as input data will use single particle samples (muons, 100 GeV) to investigate sensor signals, single point resolutions and track impact parameter resolutions.
- All results below are for sensor with 20x20 µm² pixels with 18µm and (in some places) 10µm epi layer thickness.
Is this sensor fast enough?

Carriers travel times from different points inside sensor (in ns)
Signal shapes

- Left picture – same data as on previous slide in the form of plots.
- Right picture – signal shapes in central and neighbor pixels. Notice discriminator threshold level – time of the this level crossing will give us signal time stamp.
How small time window can we set?

- On the left picture you can see time stamps (time of the threshold crossing by the signal) distribution for fastest in the cluster pixel.
- On the right plot shown the probability that time stamp of the fastest signal in the cluster falls within time window (expressed in number of CLIC BC intervals) for 18 μ and 10 μ epi layer thickness as function of pixel threshold. Pixel threshold depends on noise level and should be not less than 5 σ of noise for digital readout and 3 σ for analog readout.
Do we need analog readout?

- Plot at the right shows point coordinates measurement accuracy of pixel detector in perpendicular to B field direction (rphi) and along B field (Z) as function of number of ADC bits. 1 bit means no analog information.

- We can see, that even with very low noise level (10 e) number of bits larger than 5 does not increase point accuracy.

- Largest gain in resolution is obtained with transition from 1 to 2 bits.
Impact parameters resolutions

- Plot shows impact parameters (d0 and z0) resolutions as function of time window for accepting hits. Notice, that resolution in d0 is better than in z0, and d0 resolution in general even better than single point resolution. This is because of an extremely good r-phi resolution of silicon tracker barrels (~7 μ), which provides huge “lever arm” for d0 parameter determination. And this makes it also very little sensitive to the single layer hit loss. Tracker sensor resolution in Z is much worse, because of small stereo angle between strips.

- These plots are made for 20 e noise level. Are noise level 20e or even 10 e (used on previous slide) realistic?
  - We achieved 24e in chronopix with 100 μ² collecting electrode area and low resistivity epi layer. We can reduce electrode area by factor of 5-6, and increase epi layer resistivity 10-100 times, so we can reduce capacitance by factor 15-60. Reset noise (major noise source) scales as squire root of capacitance, so 10 e noise is realistic, if other noise sources will not became dominating with increased bandwidth.
Notice that we reset pixel before each bunch crossing. This allows for recording of more than 1 hit in the same pixel during bunch train. We will not be able to do it for CLIC. But probability of 2 hits in the same pixel is very low even for ILC with 3000 bunches in train, so CLIC with 300 bunches probably will not suffer to much efficiency loss from this fact.
Here is the first look at modifications needed for use it in CLIC
Summary of design changes

- We need much less memory for time stamp – because we need less bits (6 bits will provide 2.5 ns resolution) and we don’t need memory for second time stamp. This gives us the room for ADC. And we know, that we don’t need very precise ADC. 3 bits would be enough, I have shown design with 6 bits.

- Time/time convertor and charge/time convertors are very similar.

- Design is implementable using standard 130 nm technology transistors (though I am not sure if it will fit in 20x20 micron size). But 90 nm will be probably sufficient to get desired pixel size.

- In some respects design for CLIC is even simpler than for ILC. This is because there is no need to switch between memory cells for first and second time stamp, and also comparator offset calibration (which discussion I left outside this talk) may be easier to implement because of much smaller time interval we need to keep offset level stable (150 ns instead of 1 ms).
Seed tracking with pixsim

- Detailed instructions about pixsim simulation package can be found at [www.slac.stanford.edu/~sinev/pixsim_doc/pixsim_help.html](http://www.slac.stanford.edu/~sinev/pixsim_doc/pixsim_help.html)

- The only thing needed to use seed tracking with pixsim simulation instead of sisim simulation (which was a temporary solution) is to put declaration in the header of your reconstruction code (though there is clic_sid in the driver path, this tracking reconstruction code can be used for any detector, not necessarily CLIC.):

  ```java
  import org.lcsim.recon.tracking.seedtracker.trackingdrivers.clic_sid.MainTrackingDriver;
  Instead of: import org.lcsim.recon.tracking.seedtracker.trackingdrivers.sidloi3_digital.MainTrackingDriver
  ```
Comparison of two options

- Figure on the right shows that there is no difference in the track reconstruction efficiency with either option.
- Figure on the left shows, that point resolution of the vertex sensors is better with pixsim.
- This illustration is obtained with reconstruction of TTH events with 1000 GeV CM energy. There was an average of about 70 of reconstructed charge tracks in the event, and pixsim with use of lookup table option added just few seconds to the average reconstruction time about 60 seconds/event.

Nick Sinev for WG4 meeting, April 7, 2011
Conclusions

- Chronopixel sensor (with some modification of time stamping circuit) is capable of the assignment of hits to time window of the order of 10-20 CLIC bunch crossings with efficiency above 90 %.
- To achieve good sensor point resolution (about 3.5 μ) we don’t need analog readout.
- Improvement in the point resolution of the order of 20% can be achieved with the use of only 2 bits ADC. If the noise level of the order of 10 e- can be achieved, 4-5 bits ADC can give additional gain in point resolution, but going to more than 5 bits in the ADC does not help.
- The pixel sensors simulation package is available for use with seed tracking reconstruction package, and use of the lookup tables in pixel simulation gives essentially same results as full simulation of charge carriers movement. Such simulation does not noticeable increase tracking reconstruction time.