Table of Contents

Silicon Strip Digitization

Complete:

Explains the algorithms used to digitize the silicon strip tracker.

Introduction

"Digitization" refers to the process of simulating the electronics signals read out by the detector. The energy deposit from Geant4 is converted into charges in the silicon bulk, that are then migrated to the surface of the sensor, and shaped by the electronics. The various steps will be described in the following. To speed-up the process, different assumptions are made depending of the working mode of the electronics, we will therefore start by describing shortly these operation modes.

Operation modes of the electronics

The APV25 chip of the tracker front-end can operate in different modes. We are interested in two of them, called respectively peak and deconvolution mode.

- In peak mode, the signal is a RC-CR curve, that gives a sensitivity - somewhat reduced after some time - over about 200ns.
- In deconvolution mode, the signal is processed with a three-weights deconvolution algorithm to reduce the signal to 25ns. The signal is then approximately a Gaussian with sigma=12.5ns.

Deconvolution is the default operation mode for collisions.

The tracker FED (Front End Driver) can also be operated in various modes. Simulation handles the two most important cases: Virgin-Raw and Zero-Suppression.

- in Virgin-Raw mode, one digi is produced per strip, disregarding the presence or not of signal. Each strip contains a reading corresponding to the pedestal (typically 250 ADC counts), to which one adds the noise (positive or negative) and sometimes the signal. One digi is the measured amplitude, and there are 4 (or 6) times 128 digis in each detunit.
- in Zero-suppression, the FED does pedestal and common-noise subtraction, and then selects strips above a given threshold (typically 5*noise for isolated strips and 2*noise for two or more adjacent strips. One digi is then a pair amplitude/position, and there are n>=0 digis in each detunit.

Zero-suppression is the default operation mode for p-p collisions.

Simulation steps

The silicon strip simulation proceeds as follows. First, the behavior of charges in the silicon is simulated:

- the energy deposit by Geant4 is distributed along the path of the particle in the silicon, split in N sub-deposits taking into account Landau fluctuations.
- each individual sub-deposit is migrated to the surface of the sensor, taking into account the thermal diffusion and the Lorentz angle. The result is a charge on the surface, assigned with a width and a position.
- the various charge contributions are merged per strip.

The resulting signal is processed to include the effects of the electronics:

- the signal shaping is included, which reduces the signal of off-time particles.
- in Virgin-Raw, a pedestal and noise is added on all strips. It gives directly the vector of (raw)digis that are sent to the local reconstruction (clustering).
- in Zero-Suppression, noise tails above 2 sigmas are added to the signal strips only to for a set of digis.

Parameters and configuration

Usually, the digitizer is configured by including one of the two following configuration fragments:

- SiStripDigi_APVModePeak_cff to work in peak mode (special)
- SiStripDigi_APVModeDec_cff to work in deconvolution mode (normal)

The various parameters are described in the following table.

Parameter	Default value	Comment
DeltaProductionCut	0.120425	Geant4 parameter used for the Landau fluctuation. Should be identical to the G4 Simulation part
APVpeakmode	False	APV mode. It is set to True in SiStripDigi_APVModePeak_cff.py
LandauFluctuations	True	Activates the Landau fluctuation of the sub-deposits in the silicon
chargeDivisionsPerStrip	10	Controls the number of sub-deposits generated
CosmicDelayShift	0.0	Delay to be added to the simHit time for synchronization. 0.0 for collisions, it has to be adapted for Cosmic MC
DepletionVoltage	170	Depletion voltage of silicon sensors. 170 is a typical value from integration. It controls the diffusion
AppliedVoltage	300	Applied voltage for operations. It controls the diffusion
ChargeMobility	310	Charge mobility in silicon (for holes). It controls the diffusion
Temperature	273	Silicon temperature. It controls the diffusion
GevPerElectron	3.61e-09	Conversion factor for ionization
ChargeDistributionRMS	6.5e-10	Diffusion factor
noDiffusion	False	flag to activate/deactivate diffusion
DigiModeList	---	Outputs
ROUList	---	Inputs
GeometryType	'idealForDigi'	Geometry
TrackerConfigurationFromDB	False	Use ConfDb
ZeroSuppression	True	Switch ZS on/off
CMS.LorentzAngle	''	Db tag for Lorentz angle
Gain	''	Db tag for Gain
NoiseSigmaThreshold	2.0	Cut on the tail for noise simulation in ZS mode. Noise below N*sigma will not be added.
electronPerAdcPeak	262.0	Conversion factor (gain) for the electronics, when Peak mode
electronPerAdcDec	217.0	Conversion factor (gain) for the electronics, when Deconvolution mode
FedAlgorithm	4	Zero Suppression algorithm. That corresponds to what is set in the firmware.
Noise	True	Enable/Disable noise simulation. NOTE : turning Noise ON/OFF will make a big change
TOFCutForDeconvolution	50.0	Time-of-Flight cut in deconvolution mode. Deposits timed more than N ns away will not be converted to digis

TOFCutForPeak	100.0	Time-of-Flight cut in peak mode. Deposits timed more than N ns away will not be converted to digis
Inefficiency	0.0	Artificial inefficiency added to the digitization.
Parameter for VR data (ZeroSuppression = False)		
SingleStripNoise	True	only if Noise = True. It uses the values of noise for each strip as stored in the Db
RealPedestals	True	The pedestals as stored in the Db + 128 are used. If false just a constant shift of 128 is applied
CommonModeNoise	True	The Gaussian CMN is added for each APV. The RMS is specified by the parameters below
cmnRMStib	5.92	CMN RMS for TIB
cmnRMStob	1.08	CMN RMS for TOB
cmnRMStid	3.08	CMN RMS for TID
cmnRMStec	2.44	CMN RMS for TEC
APVSaturationFromHIP	True	The APV is killed if an hadron produces HIP. The probability of this happening can be configured by the parameter below
APVSaturationProb	0.001	Probability of an hadron producing HIP
BaselineShift	True	APV baseline shift down is applied and proportional to the APV occupancy and total charged deposited
PedestalsCondition	128	It is the pedestals offset introduced to avoid negative values on the FED. In the Db the pedestals are subtracted of 128

Conditions DB tags can be found here :
https://twiki.cern.ch/twiki/bin/view/CMS/StripTrackerDBTagsForCalibrations#MC_production

In addition, there are some parameters controlling the cross-talk (capacitive coupling). There is one vector of doubles per geometry of sensor, each time once in deconvolution, once in peak mode. In the vector, the first number is the charge remaining on the leading strip, and you have then the induced signal on the first, second, third, ... adjacent strip(s). By default, signal is only induced on the first adjacent strip.

IMPORTANT: If the RawDigis generated in VR mode are used for reconstruction, it is mandatory that RealPedestals = True. In order to access the real conditions to simulate realistic VR date, it is required to insert in the .cfg file the following lines:

```
process.poolDBESSource = cms.ESSource("PoolDBESSource",
 BlobStreamerName = cms.untracked.string('TBufferBlobStreamingService'),
 DBParameters = cms.PSet(
     messageLevel = cms.untracked.int32(2),
     authenticationPath = cms.untracked.string('/afs/cern.ch/cms/DB/conddb')
 ),
 timetype = cms.untracked.string('runnumber'),
 connect = cms.string('frontier://FrontierProd/CMS_COND_31X_STRIP'),
 toGet = cms.VPSet(cms.PSet(
     record = cms.string('SiStripNoisesRcd'),
     tag = cms.string('SiStripNoise_CRAFT09_DecMode_ForTrackerSim')
 ),
  cms.PSet(
     record = cms.string('SiStripPedestalsRcd'),
     tag = cms.string('SiStripPedestal_CRAFT09_DecMode_ForTrackerSim')
 ),
  cms.PSet(
     record = cms.string('SiStripFedCablingRcd'),
     tag = cms.string('SiStripFedCabling_CRAFT09_ForTrackerSim')
 ),
  cms.PSet(
```

```
        record = cms.string('SiStripBadChannelRcd'),
        tag = cms.string('SiStripBadChannelsFromO2O_CRAFT09_DecMode_ForTrackerSim')
    )
)
)
```

```
process.es_prefer_my =cms.ESPrefer("PoolDBESSource","poolDBESSource")
```

New parameters in Peak and Deco modes from CRAFT09 data

Yuri compared CRAFT09 and MC in Peak and Deco mode for TIB and TOB, obtaining:

- Tunes for peak mode
- Tunes for deco mode

For TID and TEC, the most reliable values so far are those extracted by Burt, see slide 8 of this talk☑ (ignore the TIB and TOB values, superseded by Yuri's values listed in the wikis).

Clarification: Burt used VR runs at 0 T, Yuri used ZS runs. There are reasons to believe that the VR runs at 0 T (peak: 111753, deco: 111664,111673) had some unaccounted feature that makes them not comparable with the bulk of the runs from CRAFT09, therefore we prefer Yuri's values when available; nevertheless, Burt's values are likely a better approximation than the old digitizer defaults (which were "educated guesses", unmodified since years).

These changes are effective since tag V05-15-05.

Plans for tuning with collision data

- SiStrip Redigitization For Lorentz Angle Calibration
- Proposal by Giacomo, 2010/01/27

New proposals for realistic simulation, with the aim of having a good data/MC agreement for the usual observables at cluster level (Yuri's analysis) and at the same time a realistic description of saturation (critical for dE/dx), while keeping things simple from the db point of view (only one payload for noise and one for gain); putting together inputs by Andrea Venturi, Marco De Mattia, etc.

See also the SiStripTKSignalNormalization wiki, and subscribe to the cms-tk-signalnorm@cernNOSPAMPLEASE.ch mailing list if interested.

Situation at present (April 23, 2010):

- First data/MC comparisons at 7 TeV by Yuri Gotra (entire set of plots here☑)
- here with a more recent run☑
- cross-talk study☑

Data to be used:

None of those currently available, as the timing is going to be re-optimized soon (after the technical stop of April 26-28). (By the way, this will lead to inconsistency between old and new data; hopefully, statistics will start to grow up much faster soon, and ignoring old data will not be dramatic for analyses aiming at Summer 2010 conferences.)

Parameters of interest:

- electronsPerAdc: constant factor for all SST (different between Peak and Deco)
- i: strip index
- tickmark(i): gain correction as measured from tickmarks, currently equal for all i's in the same **APV pair**
- g(i): from a gaussian with μ=1 and   equal to 0 for DESIGN and MC conditions and to a tuned value for START conditions
- ideal_noise(i): proportional to strip length
- miscalibrated_noise(i): ideal_noise(i)/tickmark(i)

Approach	Calib of signal in MC truth (digitizer)	Calib used in reco (clusterizer)	Noise added to signal in MC truth (digitizer)	Noise used in Zer Suppression (digitizer)
old	electronsPerAdc	1 or g(i)	ideal_noise(i)	ideal_noise(i)
new	electronsPerAdc*tickmark(i)	1/[g(i)*tickmark(i)]	miscalibrated_noise(i)/tickmark(i)	miscalibrated_noise(

Note: S/N must be invariant by calibration, at every step (simulation, zero suppression, clusterization).
Digitizer code must be modified, see below.

Payloads in db:

- dbfile_gainFromData.db: gain from data (tickmarks)
    - tag name: SiStripApvGain_default
    - content of the tag is the same as IOV 128408 of tag SiStripApvGain_GR10_v1_hlt, where missing apv's were assigned gain = 1
- dbfile.db: miscalibrated_noise(i), i.e. payload normalized with gain (noise*gain)
    - tag name: SiStripNoiseNormalizedWithGain

Digitizer:

- Gain = tickmark(i)
- Noise = miscalibrated_noise(i); ideal noise is then derived from this and the gain inside the digitizer

Clusterizer:

- Gain = 1/[g(i)*tickmark(i)]; one payload for the unsmeared (=0) and one for the smeared (from tuning) cases
- Noise = miscalibrated_noise(i); exactly as for the digitizer

Sniplet for cfg:

```
process.load("CondCore.DBCommon.CondDBSetup_cfi")

process.SiStripNoises = cms.ESSource("PoolDBESSource",process.CondDBSetup,
                              connect = cms.string("sqlite_file:dbfile_noise.db"),
                                 toGet = cms.VPSet(cms.PSet(record = cms.string("SiStripNoisesRcd"),
                                                        tag = cms.string("SiStripNoiseNormalizedW
process.es_prefer_SiStripNoises = cms.ESPrefer("PoolDBESSource","SiStripNoises")

process.SiStripGain = cms.ESSource("PoolDBESSource",process.CondDBSetup,
                         connect = cms.string('sqlite_file:dbfile_gainFromData.db'),
                            toGet = cms.VPSet(cms.PSet( record = cms.string('SiStripApvGainRcd'),
                                                      tag = cms.string('SiStripApvGain_default')
process.es_prefer_SiStripGain = cms.ESPrefer("PoolDBESSource", "SiStripGain")
```

Note: content of a sqlite file can be seen with the command

```
cmscond_list_iov -c sqlite_file:dbfile_gainFromData.db -a
```

Plans for tuning with collision data   5

Changes to the digitizer code:

SiStripDigitizerAlgorithm🗗 now (until V05-16-02):

```
float noiseRMS = noiseHandle->getNoise(RefStrip,detNoiseRange);
theSiNoiseAdder->addNoise(detAmpl,firstChannelWithSignal,lastChannelWithSignal,numStrips,noiseRMS
theSiZeroSuppress->suppress(theSiDigitalConverter->convert(detAmpl, gainHandle, detID), digis, de
```

Here noiseHandle and gainHandle are the noise and gain payloads.

SiTrivialDigitalConverter🗗 now (until V05-16-02):

```
int adc = convert( (gainHandle->getStripGain(i, detGainRange))*(analogSignal[i]) );
```

The new proposal requires this code change (effective since V05-16-04):

```
float noiseRMS = noiseHandle->getNoise(RefStrip,detNoiseRange);
float gainValue = gainHandle->getStripGain(RefStrip, detGainRange);
theSiNoiseAdder->addNoise(detAmpl,firstChannelWithSignal,lastChannelWithSignal,numStrips,noiseRMS
theSiZeroSuppress->suppress(theSiDigitalConverter->convert(detAmpl, gainHandle, detID), digis, de
```

Tuning sequence:

- electronsPerAdc: set to the average cluster charge, corrected by 3D angle and equalized by active tickness, of all SST (remember, tickmark corrections are applied everywhere)
- of g(i), with μ fixed to 1, fitted by looking at cluster charge and S/N simultaneously
- noise scaling
- cross-talk parameters

Checks:

- these🗗 variables (Yuri)
- number of pure noise clusters (Yuri?); the test is passed if there is no visible change
- number of saturating clusters, and fit of dE/dx vs P (Loic?)

To be considered later:

- The model for noise could be improved, according to the measurements in the Craft08 paper🗗.
- Tuning of   is calibration-dependent, in the sense that tickmark calibration and particle calibration would likely have different spreads. Hence we have $g^t(i)$ and $g^p(i)$.

Responsible: AndreaGiammanco

This topic: CMSPublic > SWGuideStripDigitization
Topic revision: r26 - 2014-12-16 - JlAgram