FastTimer in CMSSW

Outline

- Detector ID
- Geometry
- Simulation step

FastTimer Meeting
June 4, 2014

Sunanda Banerjee

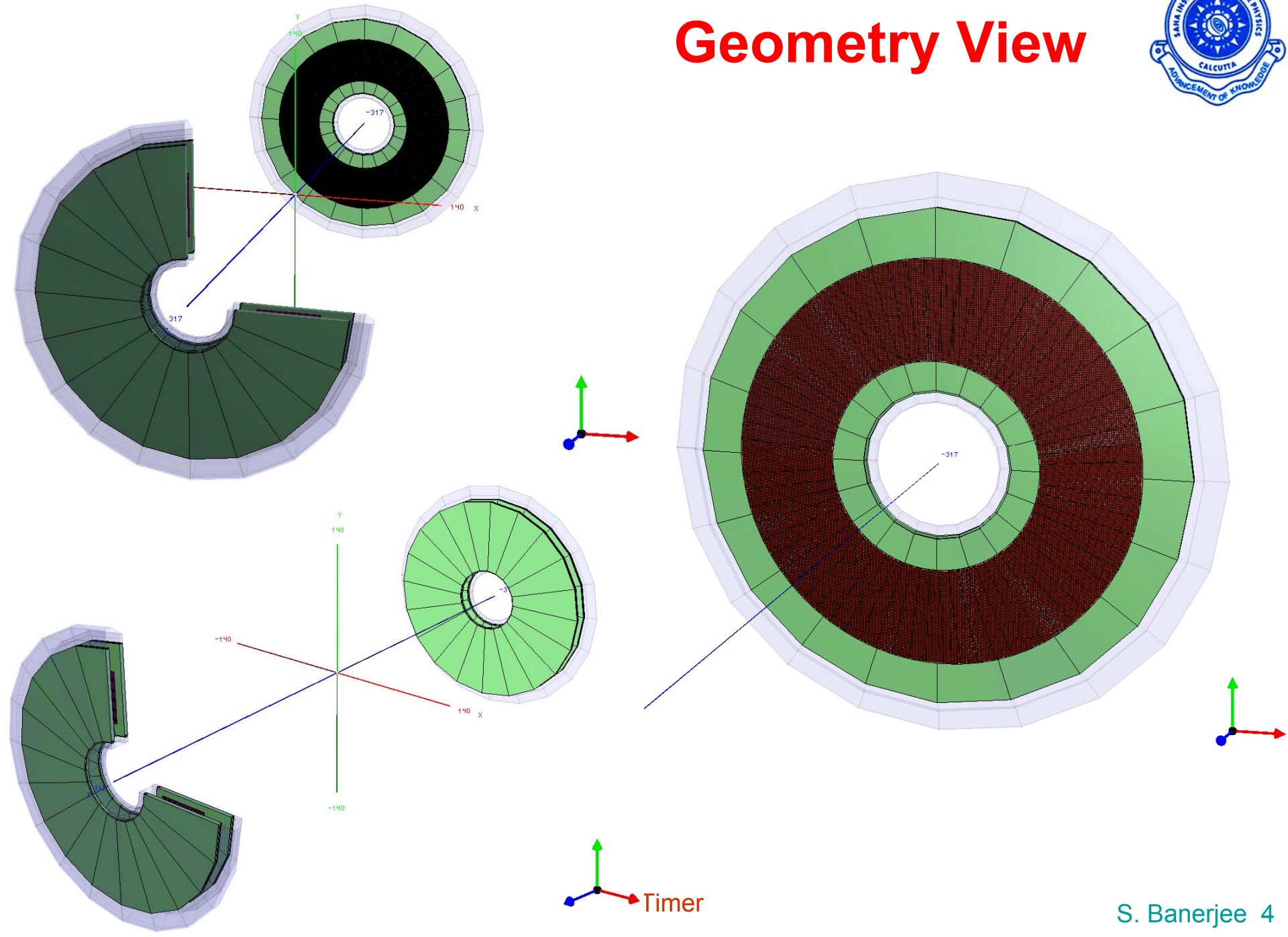
Detector Identification

- CMSSW needs detector identification class to identify every cell. An active element of the fast timing device is done using DataFormats/DetId/interface/FastTimeDetId.h
  - constructor: FastTimeDetId(int ix, int iy, int iz)
    - ix = cell # along x-axis
    - iy = cell # along y-axis
    - iz = +1 if on the +ve z-side
      -1 -ve z-side
  - methods:
    - subdet(), ix(), iy(), zside() to get the components

Geometry

- DDD Geometry is defined through a set of xml files:
  - For Geometry definition:
    - Geometry/HGCalCommonData/data/fastTiming.xml
    - v1/fastTimingElement.xml
    - v2/fastTimingElement.xml
  - There are 2 alternate ways of defining the element: v1 defines a cell for each element (this produces ~35k elements on either side of IP) while v2 provides a plane and the division of cell is done during simulation phase.
  - Usage of v2 is recommended in standard SIM step.
  - For defining numbering scheme, topology:
    - Geometry/HGCalCommonData/data/fastTimingConst.xml
  - For defining sensitive detectors
    - Geometry/HGCalSimData/data/fasttimesens.xml
  - To define cus to be used in Geant4 steps

Geometry View



Geometry Constants

- Decoding of geometry constants to be used to define topology, numbering scheme etc is done using
  - Geometry/HGCalCommonData/interface/FastTimeDDDConstants.h
  - Useful methods:
    - std::pair<int,int> getXY(double x, double y) const;
      - provides ix, iy of the cell given x, y in the local coordinate system
    - bool isValidXY(int ix, int iy) const;
      - checks if the pair ix, iy is valid ...
- This is initialized and added to the EventSetup by
  Geometry/HGCalCommonData/plugins/FastTimeNumberingInitialization.cc
  - which is invoked by
  Geometry/HGCalCommonData/python/fastTimeNumberingInitialization_cfi.py
- There is a test code which tests the loading of the constants. Use
  cmsRun Geometry/HGCalCommonData/test/testFastTimeNumbering_cfg.py
  which utilizes the code in
  Geometry/HGCalCommonData/test/FastTimeNumberingTester.cc

Simulation Step

- The sensitive detector required for FastTiming (as defined in the fasttimesens.xml file) is
  - FastTimerSensitiveDetector

  and the hit collection name is
  - FastTimerHits
- The sensitive detector class is defined in
  - SimG4CMS/Forward/interface/FastTimerSD.h
- It makes transient hit collection during the Geant4 stepping process and at the end of the event saves a a collection of tracking hit type (PSimHits) to the event. It defines the unitId from the step element using
  - virtual uint32_t setDetUnitId(G4Step*);
- Please consult: SimDataFormats/TrackingHit/interface/PSimHit.h