ACD Data Structures

ACD Data: Present & Future
Current Status

• ACD output currently consists of
  – **Digitization data available from tests**
    • Beam test & Balloon & their simulated data
  – **List of “hit” ACD tiles, consisting of:**
    • Id, PHA, Discriminators: Veto and High (CNO)
  – **Ntuple output**
    • ACD_TotEnergy, ACD_TileCount, ACD_ActDist
      ACD_DOCA, ACD_ThrottleBits, ACD_No_FaceN
      ACD_No_SideRowN
Proposed Digitization Updates

• Each ACD Tile has 2 PMTs.
  – Each PMT has its own unique identifier
    • Create a new AcdPmtId class
  – Modify AcdTile class to store
    • AcdTileId
    • 2 PMT objects, class name AcdPmt
  – Each PMT object contains
    • AcdPmtId
    • 16 bit word
      – 12 bits for PHA, 4 “extra” bits for discriminators
Proposed Digi Diagram

AcdTile
- m_id: AcdTileId
- m_pmtArr[2]: AcdPmt
  - id()
  - AcdPmt& pmt(AcdPmtId)
  - AcdPmt& pmt()

AcdPmt
- m_id: AcdPmtId
- m_val: unsigned short
  - id()
  - pha()
  - low()
  - veto()
  - cno()

AcdTileId
- m_id: unsigned int
  - id(short base)
    - layer()
    - face()
    - row()
    - column()

AcdPmtId
- m_id: string
Ntuple Updates

• It would be nice if the ACD Recon output stored enough data to re-calculate any of the provided ntuple quantities.
  – It would be that much easier to experiment with new quantities

• Example: ACD_EnergyDeposit_Deposit_Max
Proposed Recon

• List of AcdTiles and est. energy deposition
• AcdActiveDistance class contains
  – TrackId
  – DOCA value
Proposed Recon Diagram

**AcdRecon**
- `m_tiles`: vector of `AcdReconTile`
- `unsigned int tileCount(type)`
- `float maxEnergy()`
- `AcdDoca& gammaDoca(type)`
- `AcdDoca& trackDoca(type)`

**AcdReconTile**
- `m_id`: `AcdTileId`
- `m_energy`: `float`
- `m_hit`: `bool`
- `m_cno`: `bool`
- `m_docas`: vector of `AcdActiveDistance`
- `AcdTileId& id()`
- `bool hit()`
- `bool cno()`
- `float energy()`
- `activeDistance(trackId)`

**AcdActiveDistance**
- `m_value`: `float`
- `m_track`: `TrackId`
- `float activeDistance()`
- `TrackId& track()`