

# Motivation for SICHIPON

---

---

- We all use SvxConfigurations DB view
  - Used by SiExpected to determine where hits are expected on tracks
  - Slow call to DB, large overhead when called often in secondary datasets
- The SICHIPON table is the user's replacement for SvxConfigurations

# Structure of SICHIPON Table

---

---

- Complete table, keyed by SiChipKey
  - Data word is bit-packed ChipOnStatus word
    - Biased and Integrated?
    - Read-All / Nearest Neighbor Readout
    - Threshold
    - DPS on/off
  - Thanks to Reid for designing Java class for table
- I make one call to SvxConfigurations so that you don't have to...
- (SvxConfigurations is still around for low-level users, but not for typical end-user analysis.)

# Filling the Table for Data Runs

---

---

- A chip is expected to give data (hits) if:
  - It is integrated in the hardware DB
  - It is reading out a biased ladder
  - It is not after the “last chip” in the HDI
- The table is filled along with the pedestal table for every data run and is written to PROD\_PHYSICS\_SVX used set
- For Run 168823, 602/5644 chips give no data.

# Accessing the SICHIPON Table

---

---

- CalibManager loads a map of <SiChipKey, ChipOnStatus> from calib used set
- Methods in SvxDaqObjects provided to loop through map and create sets for users
  - getBiasedLadderSet (biased and integrated)
  - getActiveLadderSet (at least 1 chip active with reasonable thresholds (<250))
- Should be transparent to end-user of SiExpected and related classes
- Validated for end-users by Matt and Reid