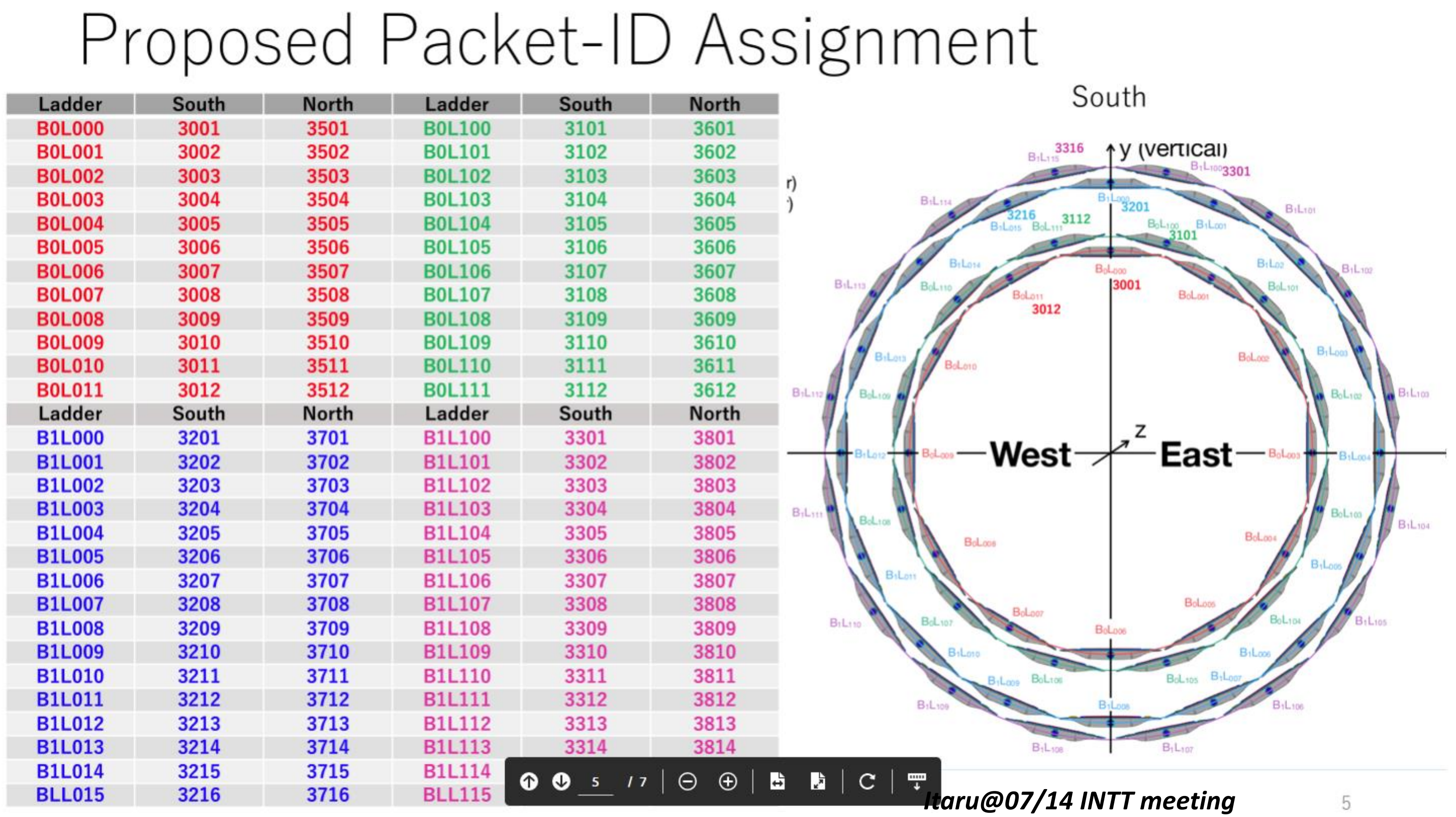
INTT Channel Indexing in Raw Data

Joseph Bertaux

Outline

- Online raw data convention proposed in previous meeting
    - previous presentation
    - Have a satisfactory Packet-ID assignment proposal
        - Determines location up to a unique half-ladder
        - (Image on following slide)
    - Chip ID and channel assignment is not decided
- Goal: propose chip ID and channel convention
    - Check Geant for existing chip and channel ID

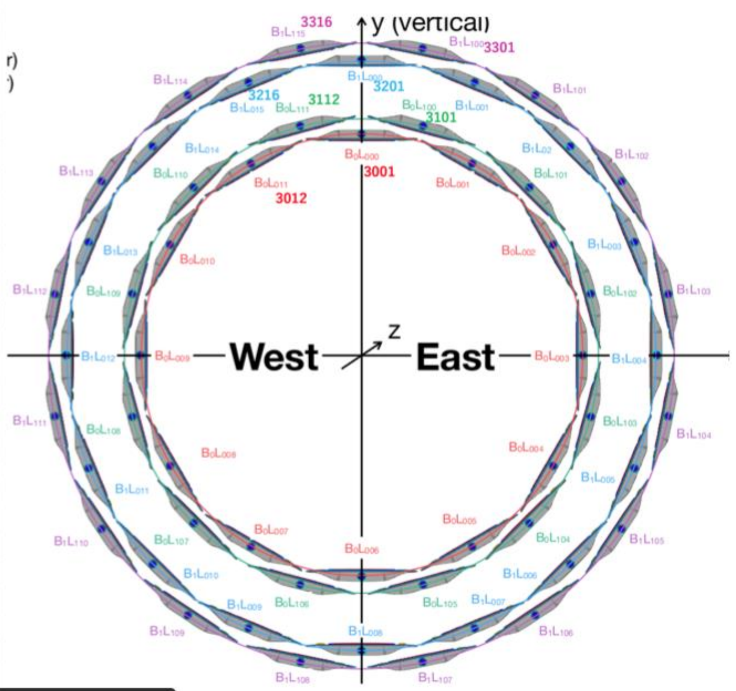
Outline



Proposed Packet-ID Assignment

Ladder	South	North	Ladder	South	North
B0L000	3001	3501	B0L100	3101	3601
B0L001	3002	3502	B0L101	3102	3602
B0L002	3003	3503	B0L102	3103	3603
B0L003	3004	3504	B0L103	3104	3604
B0L004	3005	3505	B0L104	3105	3605
B0L005	3006	3506	B0L105	3106	3606
B0L006	3007	3507	B0L106	3107	3607
B0L007	3008	3508	B0L107	3108	3608
B0L008	3009	3509	B0L108	3109	3609
B0L009	3010	3510	B0L109	3110	3610
B0L010	3011	3511	B0L110	3111	3611
B0L011	3012	3512	B0L111	3112	3612
Ladder	South	North	Ladder	South	North
B1L000	3201	3701	B1L100	3301	3801
B1L001	3202	3702	B1L101	3302	3802
B1L002	3203	3703	B1L102	3303	3803
B1L003	3204	3704	B1L103	3304	3804
B1L004	3205	3705	B1L104	3305	3805
B1L005	3206	3706	B1L105	3306	3806
B1L006	3207	3707	B1L106	3307	3807
B1L007	3208	3708	B1L107	3308	3808
B1L008	3209	3709	B1L108	3309	3809
B1L009	3210	3710	B1L109	3310	3810
B1L010	3211	3711	B1L110	3311	3811
B1L011	3212	3712	B1L111	3312	3812
B1L012	3213	3713	B1L112	3313	3813
B1L013	3214	3714	B1L113	3314	3814
B1L014	3215	3715	B1L114	[illegible]	[illegible]
BLL015	3216	3716	BLL115	[illegible]	[illegible]

South



5 / 7

Itaru@07/14 INTT meeting

5

Geant Indexes

- Geant contains a class, PHG4Hit
- PHG4Hit has many virtual methods which subsystems can use:

- virtual int get_detid()
- virtual int get_row()
- virtual int get_sector()
- virtual int get_layer()
- virtual int get_strip_z_index()
- virtual int get_strip_y_index()
- virtual int get_ladder_z_index()
- virtual int get_ladder_phi_index()
- virtual int get_index_i()
- virtual int get_index_j()
- virtual int get_index_k()
- virtual int get_index_l()

Geant Indexes

- The default virtual implementation for these indexes is to return
    - INT_MIN = -2147483648 ~ -2.14x10^9
    - (cpp std constant)

- Various detector subsystems can override this with their own implementation to return specialized indexes as they need to

- Can analyze how INTT implements these in the Geant framework with a Fun4All macro (write all indexes with the hit location to an NTuple)

Geant Indexes

- By comparing branch values to INT_MIN, we can conclude the following red methods are unimplemented in INTT's child PHG4Hit class:

- virtual int get_detid()
- virtual int get_row()
- virtual int get_sector()
- virtual int get_layer()
- virtual int get_strip_z_index()
- virtual int get_strip_y_index()
- virtual int get_ladder_z_index()
- virtual int get_ladder_phi_index()
- virtual int get_index_i()
- virtual int get_index_j()
- virtual int get_index_k()
- virtual int get_index_l()

- nt->Scan("get_...", "get_... != -2147483648") gave identically 0 entries for get_...()

Geant Indexes

- get_detid() and get_layer() are the offline convention’s layer
  - same* as TrkrDefs::getLayer(TrkrDefs::hitsetkey)

- get_ladder_z() is the offline convention’s ladder_z
  - same* as InttDefs::getLadderZId(TrkrDefs::hitsetkey)

- get_ladder_phi() is the offline convention’s ladder_phi
  - same* as InttDefs::getLadderPhiId(TrkrDefs::hitsetkey)

*doing nt→Scan(“…”, “… != get_…”) revealed up ~<100 entries of the 17,000 entries with dissimilar values

Geant Indexes

- get_index_l() was implemented, but its value is not obvious
    - Entries were ~-9x10^7
    - Not clear from inspection how it relates to any offline or hardware indexing conventions
        - Possible it is several indexes encoded into one value

Summary

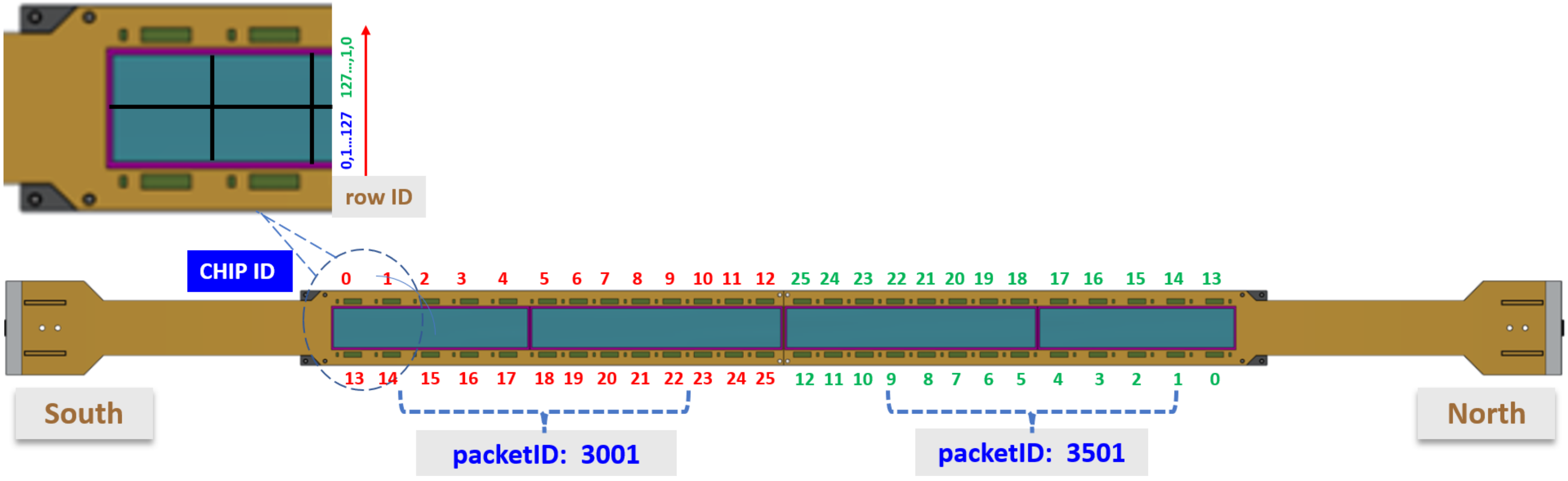
- The majority of these methods are unimplemented for INTT
- Of the implemented ones, the do not suggest a convention for chip and channel

- virtual int get_detid()
- virtual int get_row()
- virtual int get_sector()
- virtual int get_layer()
- virtual int get_strip_z_index()
- virtual int get_strip_y_index()
- virtual int get_ladder_z_index()
- virtual int get_ladder_phi_index()
- virtual int get_index_i()
- virtual int get_index_j()
- virtual int get_index_k()
- virtual int get_index_l()

- Thus, we propose a chip/channel raw data indexing convention based on hardware:

Proposed Chip/Channel Indexing

- Follows the hardware chip indexing convention, but is base 0 instead of base 1
- Rotationally symmetric



Offline Indexes (Review)

- layer (3 to 6)
  - 4 layers
  - MVTX is the innermost and uses 0-2, INTT is next

- ladder_phi (0 to 11 or 15)
  - layers 3 and 4 have 12 ladders distributed azimuthally
  - layers 5 and 6 have 16 ladders distributed azimuthally

- ladder_z (0 to 3)
  - specifies which of the 4 sensors (two A type, two B type)
  - arranged 1, 0, 2, 3 (going from North to South)

Offline Indexes (Review)

- strip_col (0 to 4 or 7)
  - specifies which part of the sensor we are at
  - 0-4 for type B (ladder_z = 1, 3)
  - 0-7 for type A (ladder_z = 0, 2)

- strip_row (0 to 255)
  - specifies which part of the chip was hit
  - 2 rows of chips, each chip with 128 channels