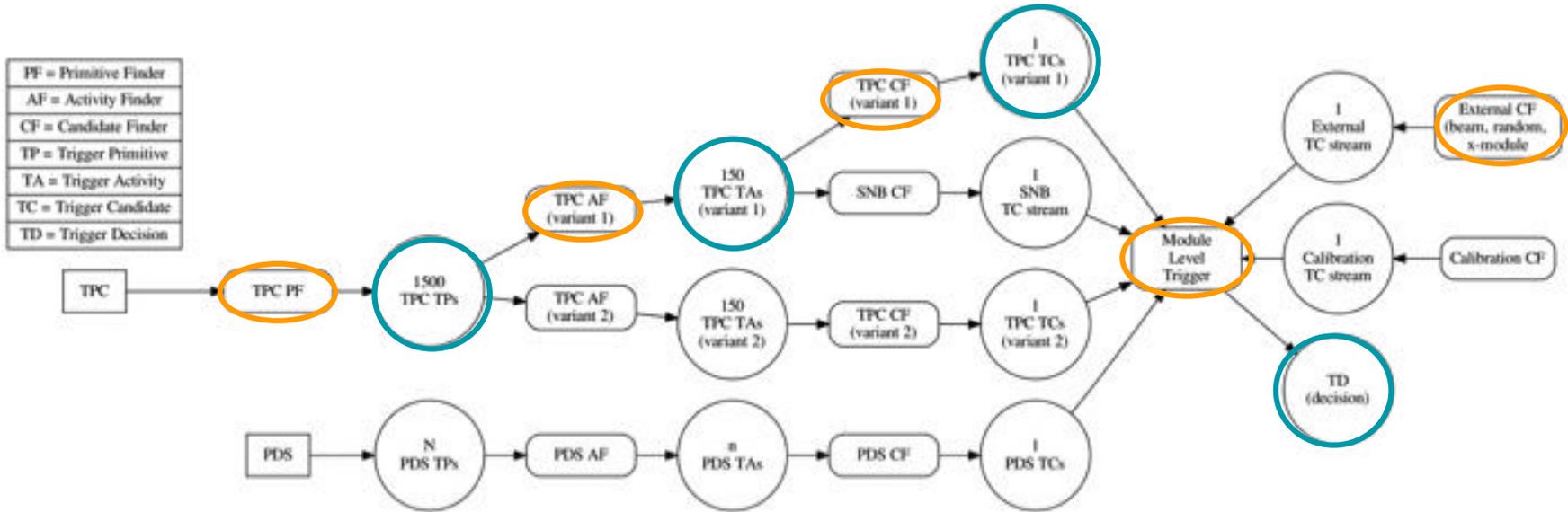


Finalizing DS data products

Georgia Karagiorgi, Lukas Arnold
Columbia University

DUNE DAQ PP/DS WG Meeting
Feb. 2, 2021

Reminder: DS, operates ~independently per FD Module



Data producers
Data products

Not shown, the HLF provides its own data products (additional header info). TBD after discussion with offline.

See also dune docdb-21839 for DS Specifications

TP Data: To be refined

The input of the TPC and PDS PFs is raw unbiased TPC data, and minimally biased PDS data, respectively, with format specified by the UD. The output of the TPC and PDS PF is Trigger Primitives (TPs), with TP format specified as follows:

```
struct TriggerPrimitive {
  int64_t time_start = {0}; ← Presumably in units of X MHz time ticks relative to run start? What is X? (must
  int64_t time_peak = {0}; ← accommodate PDS and TPC. Calculated in upstream DAQ
  int32_t time_over_threshold = {0}; ←
  uint32_t channel = {0}; ← Unique only within a subdetector
  uint32_t adc_integral = {0}; ← Calculated in upstream DAQ
  uint16_t adc_peak = {0}; ←
  uint32_t detid = {0}; ← (Sub)detector ID, e.g. APA number and link number, or PDS group and link number
  uint32_t type = {0};
  uint16_t algorithm = {0}; ← Specifies the TP algorithm and version used in UD (How is DS aware of this?)
  uint16_t version = {0}; ←
  uint32_t flag = {0}; ← ?
};
```

TA Data: To be refined

One AF per PDS/TPC SubDetector unit (e.g. APA).

```
struct TriggerActivity {  
    int64_t time_start = {0};  
    int64_t time_end = {0};  
    int64_t time_peak = {0};  
    int64_t time_formed = {0};  
    uint32_t channel_start = {0};  
    uint32_t channel_end = {0};  
    uint32_t channel_peak = {0};  
    uint32_t adc_integral = {0};  
    uint16_t adc_peak = {0};  
    uint32_t detid = {0};  
    uint32_t type = {0};  
    uint32_t algorithm = {0};  
    uint16_t version = {0};  
};
```

Presumably in units of X MHz time ticks relative to run start? What is X? (must accommodate PDS and TPC. AF algorithm specific.

Specify extent in channel and time space. AF algorithm specific.

AF algorithm specific

(Sub)detector ID, e.g. APA number, or PDS group

TA Type is recognizable by CF stage: e.g. TPC or PDS

Specifies the TA algorithm and version used in DS

```
std::vector<TriggerPrimitive> tp_list;  
};
```

A vector of TP's which make up this TA.

TC Data: To be refined

Can be per PDS/TPC SubDetector unit (e.g. APA),
or groups of PDS/TPC SubDetector units.

```
struct TriggerCandidate {  
    int64_t time_start = {0};  
    int64_t time_end = {0};  
    int64_t time_decided = {0};  
    uint32_t detid = {0};  
    uint32_t type = {0};  
    uint32_t algorithm = {0};  
    uint16_t version = {0};  
  
    std::vector<TriggerActivity> ta_list;  
};
```

← Presumably in units of X MHz time ticks relative to run start? What is X? (must accommodate PDS and TPC. CF algorithm specific.
Do we also need channel info?

← (Sub)detector ID, e.g. APA number(s), or PDS group associated with this TC. **Should probably be expanded to include a list of (sub)detector IDs for TCs which act on groups of PDS/TPC SubDetector units. Should be >150 bits, or list of IDs.**

← TC Type is recognizable by MLT stage: e.g. TPC localized HE, TPC localized LE, TPC extended (SNB), PDS ..., PDS+TPC ...

TD Data: To be refined

```
struct TriggerDecision {
    int64_t time_start = {0};
    int64_t time_end = {0};
    int64_t time_trigger = {0};
    uint32_t trigger_number = {0};
    uint32_t run_number = {0};
    uint32_t subrun_number = {0};
    uint32_t type = {0};
    uint32_t algorithm = {0};
    uint16_t version = {0};
    std::map<ComponentIDs> td_components_list;

    std::vector<TriggerCandidate> tc_list;
};
```

where the 'ComponentIDs' map of includes a a list of detector IDs (e.g. APA number and APA link number, PDS number and PDS link number, etc.) to be read out. This list is to be informed by the detid's reported by the TAs forming a given TD.

For all data products: next steps

Is the correct number of bits ascribed to hold each piece of information?

Cross-check that the combined TP, TA, TC, TD rates are not overwhelming. (Algorithm specific, but can be guesstimated).