FE questionnaire

This document is a questionnaire for Front-End (FE) developers on the FE data format which will be used in the sub-detector FE to send data to the Back-End (BE).
Since some studies have shown that the data format could have a lot of influence on the FPGA logic gates resources in the BE but also in the FE FPGA or ASIC, it's important to define precisely all the parameters about the data format for each sub-detector FE.

Here is some documentation to understand better the data format and his influence on the FE and BE FPGA (or ASIC for FE) logic gates resources. **Please read those documentations before answering the questions.**

- Data format documentation : https://lbredmine.cern.ch/documents/7
  You will find here all the information about the data format rules required between FE and BE.
- 2 talks about the influence of the FE data format on the FE and BE FPGA (or ASIC for FE) logic gates resources
  - Guillaume Vouters's talk
    : https://indico.cern.ch/getFile.py/access?contribId=1&sessionId=0&resId=0&materialId=slides&confId=254741
  - Federico Alessio's talk
    : https://indico.cern.ch/getFile.py/access?contribId=5&resId=1&materialId=slides&confId=225753

TO DEFINE:

Front End

- FE encoding data format
  - Manpower: who is in charge?
    Mainly Rui Gao (Oxford) with Steve Wotton (Cambridge)
- Technology : ASIC or FPGA or both
  FPGA
- Data Format
  Depends on detector region. Some links always zero-suppressed. Some links always non-zero-suppressed.
  In both cases we intend to use a fixed format with the header at a fixed location in the frame and one frame per event. Events are truncated to fit within a single frame if necessary for zero-suppressed data.

NON-ZERO-SUPPRESSED

(112-86)=26 bits available for header

- Header **BCID** field width: 12
- Header **information** field width: 3 (ZS flag, no data flag, truncated flag)
- Header **data_length** field width: 0 bit, data is fixed length
- unit value of Data Length: Not needed
- GBT width (80 or 112 bits) : 112
  - If 112bits: error detection/correction/recovery system?: Error check on header
- NZS possible?
- max of data after one header for NZS and ZS for one fiber: Fixed, 86bit
- Are data time-ordered by BCID? : Yes

ZERO-SUPPRESSED DATA

(112-96)=16 bits available for header

- Header **BCID** field width: 9
- Header **information** field width: 3 (ZS flag, no data flag, truncated flag)
- Header **data_length** field width: 4 bits (0-12 hits stored per frame)
- unit value of Data Length: 8bit units
- GBT width (80 or 112 bits) : 112
  - If 112bits: error detection/correction/recovery system?: Error check on header
- NZS possible?
- max of data after one header for NZS and ZS for one fiber: 12×8bit hits
- Are data time-ordered by BCID? : Yes

- Data format compliant with the specs? Yes
  - If not: why?
- Sync command integration and synch-frame definition: As spec
- What is the data latency? Is it fixed or variable? Fixed for non-zero suppressed. Probably also for zero-suppressed data.
- Will the front-end transmit frames containing non-valid-data that can be ignored by TELL40 ? How will these be flagged? (eg the dataValid input signal to the GBTX, which changes the GBT-frame header): Idle may be sent
- What special running modes are required? None foreseen
- Data emulation available? Probably some sort of pattern generator.
- Estimated bandwidth (how many AMC40 and inputs links by AMC40): 1300 links (RICH1 + RICH2)
- Estimation resources FE encoding for the ASIC or FPGA: Not yet known
- Bit ordering. See page 8 and 9 (section 1.b.) on the Data format documentation (https://lbredmine.cern.ch/documents/7) and tell which solution would be more convenient and why. Compare with how you will fill a GBT frame. No strong preference
- Comments on the Data format documentation

Back End

- Data Processing
  - Generic or Specific : Generic foreseen
  - Manpower: who is in charge? Under discussion (contact Steve Wotton)
- Status
- What output interface would you need for the data processing block and which frequency? Generic passthrough
- DSP block for the data processing needed? None