SciFi TELL40: Progress and Status

O. Le Dortz (LPNHE, Paris)
on behalf of the SciFi FE and BE Electronics teams

SciFi General Meeting, Feb 27th 2023

https://indico.cern.ch/event/1253790/

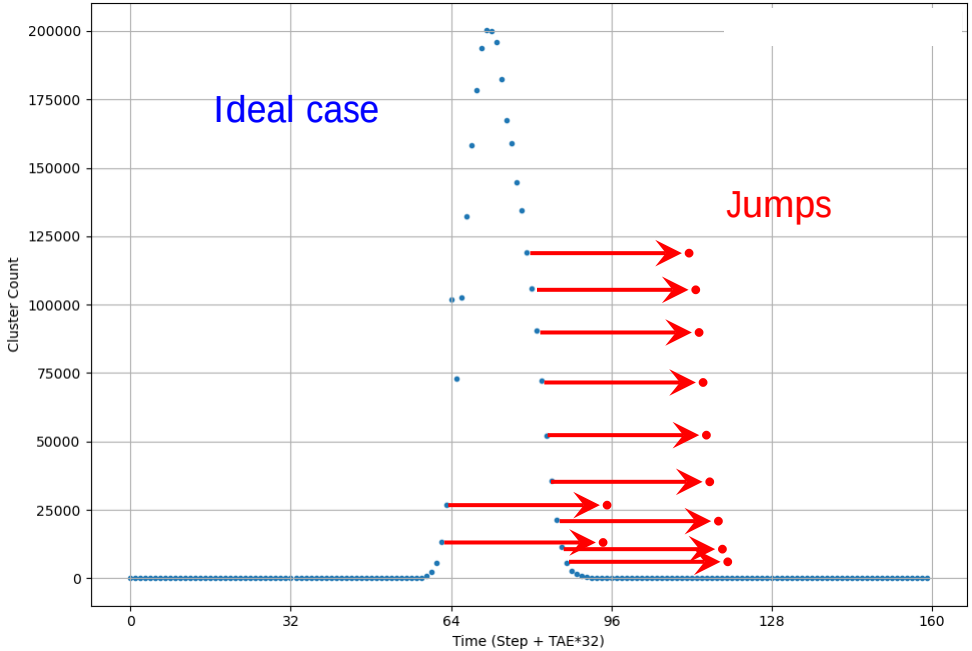
Progress on the « BXID jumps » problems in TELL40s

- Problem identified in July-August ‘22 when trying to fine tune the front-end PACIFIC sampling clock with respect to the beam (see Ulisses’ presentation in December)

- To calibrate the delays of the Master-GBT clocks distributed to the PACIFIC ASIC
  - Perform a “BEAM TIMING SCAN” Step run
    - TAE run, with central BXID set to an isolated bunch BXID (window of 5 events)
    - accumulate the number of clusters per link and per BXID for each step
    - at each step, apply an additional fine delay on the PACIFIC sampling clock (~32 steps for 25 ns)

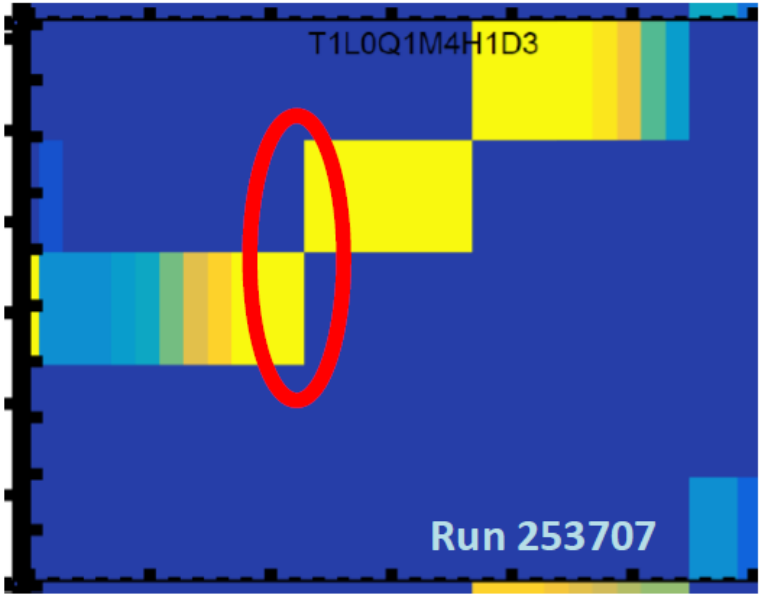
    => Identify the step for which the maximum clusters are detected

Beam Timing Scans: the BXID jumps



5 consecutive cluster counts, superimposed over the 32 steps (scan over 25 ns)

Also cf Blake's presentation in December'22



Distribution of cluster counts:

- X axis = step number (0 to 31)
- Y axis = from centralBxid-2 to +2

Trying to identify the source of the problem

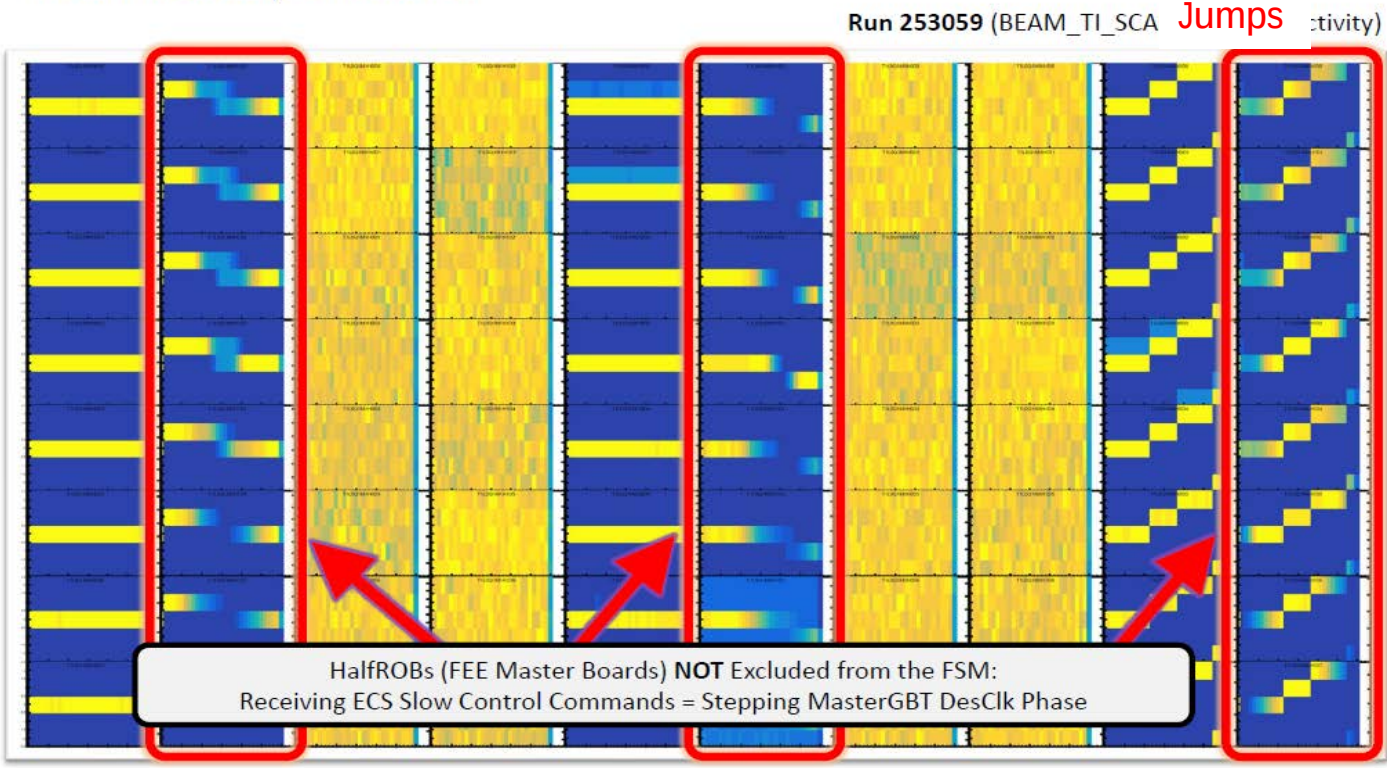
- We could quickly find that those jumps occur only when a TFC FE reset & BE reset is issued at each step (actually disabling the FE reset and BE reset vetos in the ODIN settings)
- For successful scan runs, FE reset and BE reset are needed between steps (because shifting PACIFIC clocks between steps has the side-effect that, during a transient period, the decoders in the TELL40s enter in error at the start of the following step)

- We continued the investigations without beam, using the light injection system on the FE electronics
  - See Ulisses' presentation in December https://indico.cern.ch/event/1222331/contributions/5166004/attachments/2560386/4413312/SciFi%20General%2005-12-2022%20_%20Operations%20v3.pdf

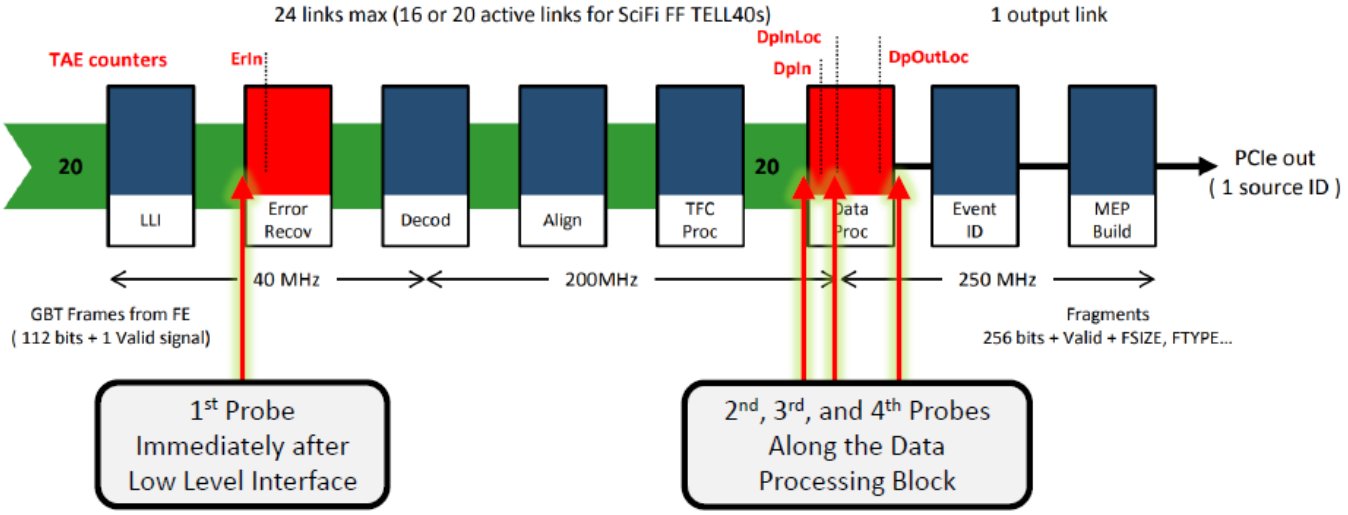
  - Insertion of probes in the SciFi specific parts of the TELL40 firmware, storing the cluster counts at several places in the flow. Development of addition WinCC tools to read, at each step of the run, the contents of those counters and other diagnostics information and write them in a "Run Summary" file (1 yaml format file per half-TELL40)

Example of beamless fine time scans

Beamless Operations



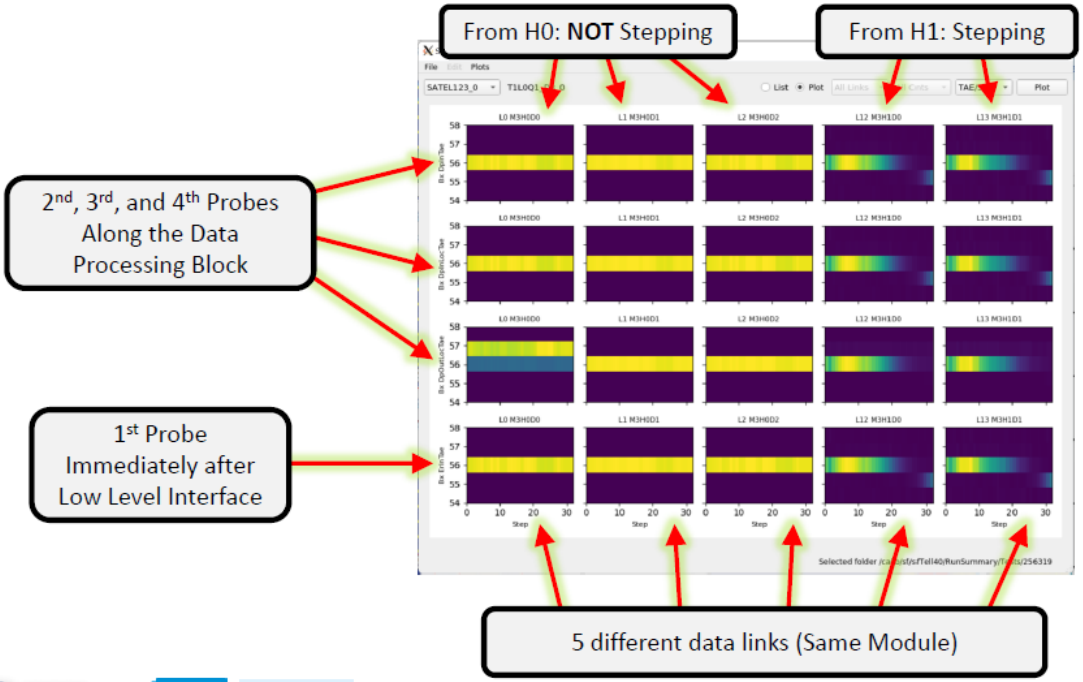
Debug TELL40 firmware on a few data links



- Covers 5 data links per Tell40 Half-Board, on 5 BxIDs: From TAE -2 to TAE +2
- Counters readable through slow control registers;
- FSM Handler, if enabled, read all counters and write values to a file after each Step or on STOP_RUN.

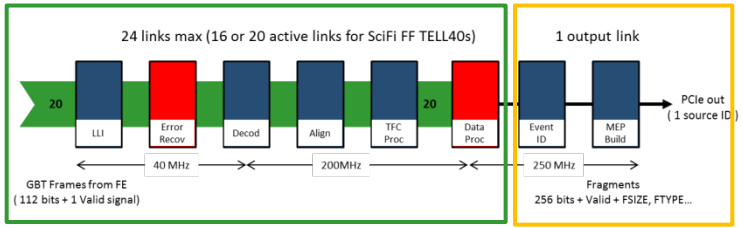
Probes analysis with beamless Fine Timing Scans

- Python tool to parse yml files and make plots:



Outcome:

No BXID jumps observed from the TELL40 flow input to the Data Processing block output



Could it come from the downstream TELL40 modules (EventID, MEP builder) or from the EB, software ?

Errors in the TELL40 Event ID block

- In Dec'22, thanks to new monitoring tools developed by the online team (Grafana web interface), a correlation was found between the BXID jumps and the generation of error fragments generated by the Event ID block (Event ID jump errors)

- Beginning of 2023:
  - Using a version of the TELL40 FW modified by Guillaume (including "SignalTap" internal probes), we could determine that those Event ID jump errors were **fake** errors (due to the generation of BE resets between steps)
  - Eventually, production of a modified version of the TELL40 FW, strengthening the reset tree in the Event ID block. This FW was deployed on the whole SciFi detector TELL40s and some Beam Timing scans were performed
    => **No BXID jumps errors was observed anymore**

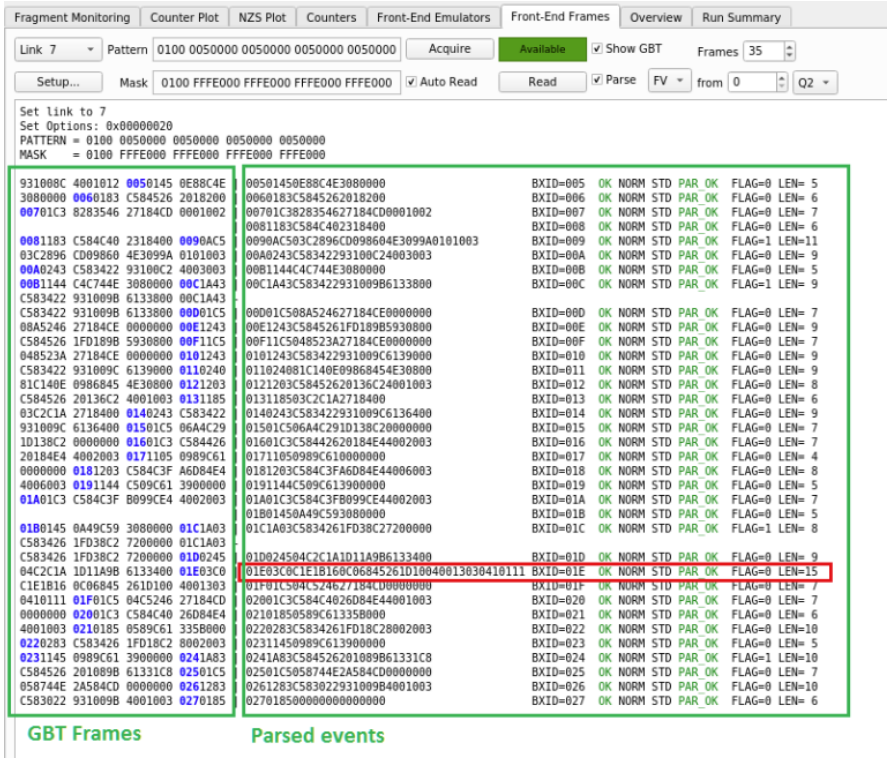
- **Conclusion**:
  - **The latest version of TELL40 firmware seems to have solved the problem (crossing fingers), even if the reason why the generation of error fragments in the Event ID block induced those jumps is not really understood**
  - A lot of efforts was needed over several months to understand and fix the problem.
    However, we learned a lot about the full processing chain. New monitoring tools were developed, which can be used, in the future, for a better understanding of unexpected behaviors.

TELL40 FV format firmware

- Fall ‘22 : FV format TELL40 tested in the pit setup
  - Correct decoding at the ‘START’ WinCC phase
  - Decoding errors as soon as a WinCC ‘GO’ command is applied
    At ‘GO’, a set of SYNC commands are sent by the TFC,
    while the decoder is already running.
    This case was not supported by the FV decoders

- Feb ‘23 : New FV TELL40 version, where the decoder supports SYNC frames while running
  - Tested on the whole SciFi-A partition (24 FV, 48 FF TELL40)
    No WinCC related problem
  - Stable run for most boards
  - But some boards went in error, when 1 or 2 links of have a high cluster occupancy (e.g. hot SiPMs)
  - ⇒ Need more investigation to understand the problem, using: Emulated data (produced by TELL40 embedded emulators or the Cluster FPGA) and dedicated VHDL simulations

GBT frames read in TELL40, with light injection



Other topics

- Threshold scan activities for FE SiPM channels calibration
  - Operational
  - But the scan procedure is slow (~2 hours)
    - Bottleneck = change the threshold values between scans on all the FE boards
    - Some efforts currently made on the FE WinCC system optimisation (esp. partitioning the project into several sub-projects could speed-up the FE accesses (parallelized)
  - Since a recent WinCC modification of the TELL40 generic parts, some difficulties in configuring correctly all the TELL40s for threshold scan. Workaround used until the problem is fixed by the online team

- Problems of long term stability of the coarse time alignment
  - SOL40 firmware related. Online team working on it (fixed latency firmware)

Conclusion

- Problem of BXID jumps during step runs seems solved (although the reason why is not very clear)

- TELL40 FV firmware:
  - Operational for nominal operation. Tested over a full partition
  - Still some problems in high occupancy cases
    => Next steps: stress tests with emulated data (in TELL40 or in the cluster FPGA) and with dedicated FW simulations

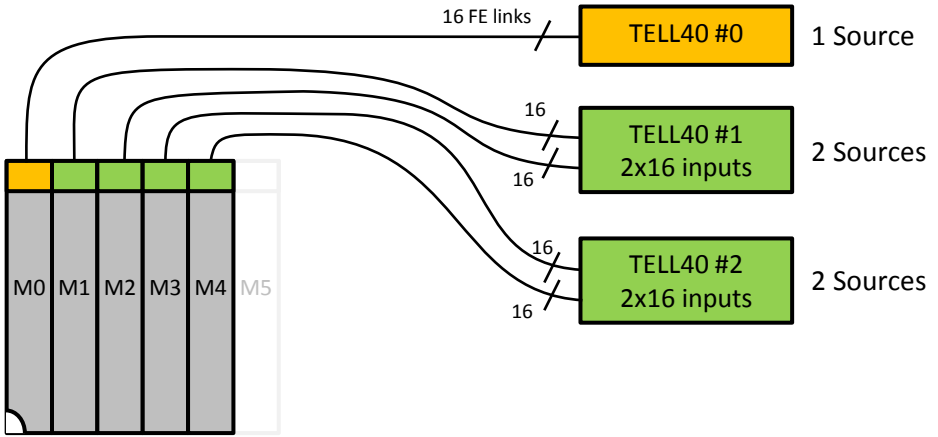
Thank you!

TELL40 modularity

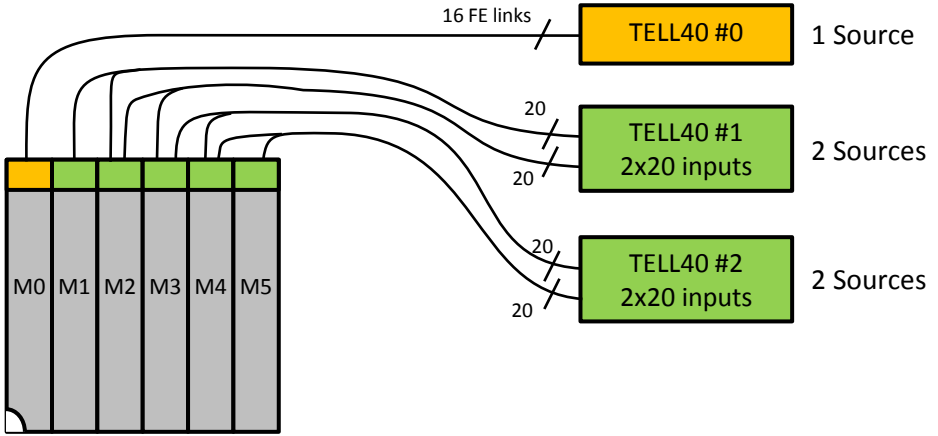
- T1 and T2: 5 modules of 16 SiPMs per quadrant
- T3: 6 modules per quadrant
- 4 quadrants => 1 C-Frame

- **2 different TELL40 firmwares:**
  - M0: TELL40 #0
    16 links TELL40 **FV format**
  - M1-M5: TELL40 2x20 inputs
    2x20 links TELL40 **FF format**

    Same FW for the 3 stations for simplicity
    For T1 and T2, connect only 2x16 FE links out of the 2x20 available inputs

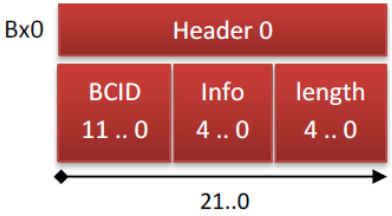


1 quadrant = 80 SiPMs (T1 and T2)



1 quadrant = 96 SiPMs (T3)

SciFi FF Front-End Data Format



Position in Info[4:0]	Position in Header[9:5]	Description
4:3	9:8	Frame-info bits
2	7	NZS data
1	6	Parity bit. Parity of the other 21 bits of the header
0	5	Error Flag (reserved for future use)

Frame Type	Frame Info		Frame Data
	Hdr[9]	Hdr[8]	
Normal (SFV)	0	0	Normal SFV data frame (Sect. 4.1)
Normal (FF)	0	0	Normal FF data frame (Sect. 4.2)
FE Buffer Full (SFV)	0	1	Only header sent by FE to recover FE-Buffer (Sect. 4.1.2), frame size 28b
Hdr-only (SFV)	1	0	Only header (requested by TFC), frame size 28b
Hdr-only (FF)	1	0	Only header (requested by TFC), frame size 112b
Hdr-only after NZS (FF)	1	1	Only header to recover FE-Buffer after NZS (Figure 7), frame size 28b (part of four-frames sequence)

SciFi FF Front-End Data Format

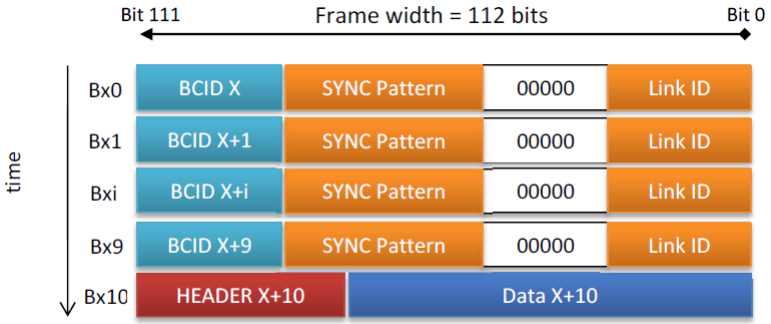
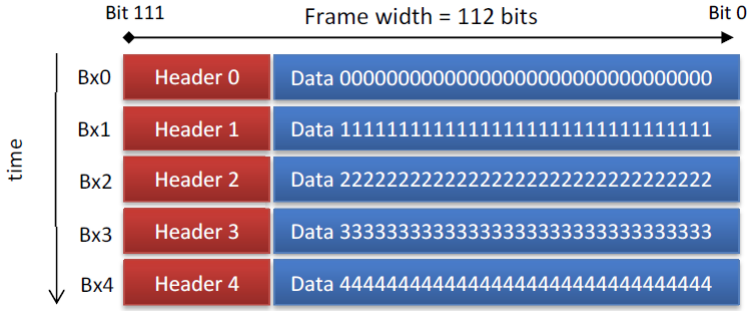


Figure 9: SYNC frames (BX0-BX9), followed (BX10) by a standard data frame.



TFC NZS=0

- **TFC Header Only = 0** => FrameInfo ="00"
- **TFC Header Only = 1** => FrameInfo = "10", Data=00000

TFC NZS=1
=> FrameInfo ="00"

TFC NZS=0
=> FrameInfo ="11"

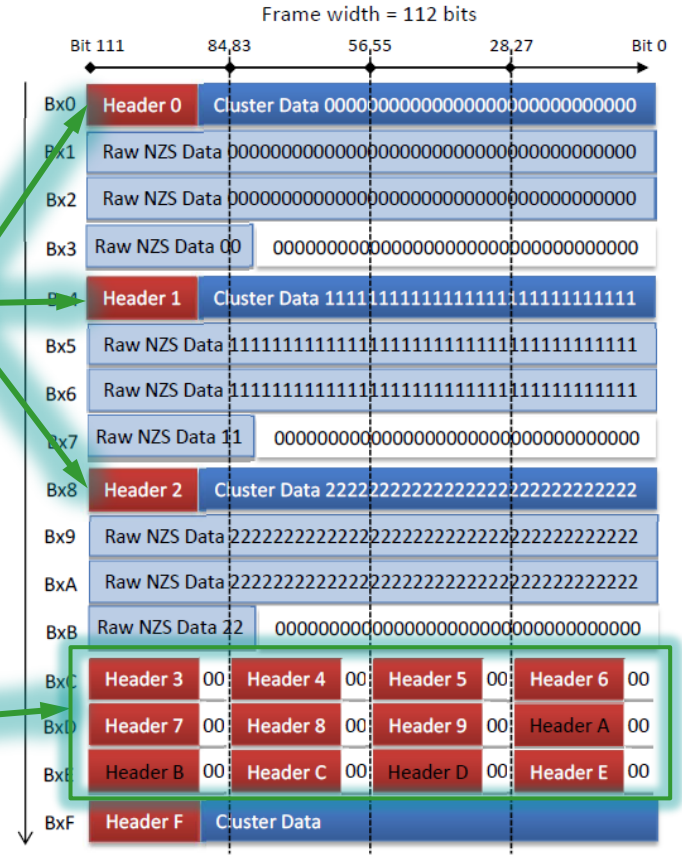


Figure 7: Data generated by three consecutive NZS commands in FF mode.

SciFi FV Front-End Data Format

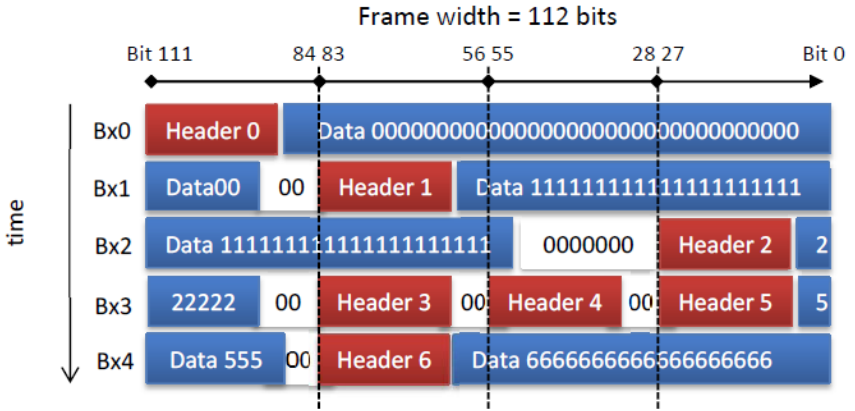


Figure 4: SciFi specific data format with a fixed header and variable data (SFV).

Frame Type	Frame Info		Frame Data
	Hdr[9]	Hdr[8]	
Normal (SFV)	0	0	Normal SFV data frame (Sect. 4.1)
Normal (FF)	0	0	Normal FF data frame (Sect. 4.2)
FE Buffer Full (SFV)	0	1	Only header sent by FE to recover FE-Buffer (Sect. 4.1.2), frame size 28b
Hdr-only (SFV)	1	0	Only header (requested by TFC), frame size 28b
Hdr-only (FF)	1	0	Only header (requested by TFC), frame size 112b
Hdr-only after NZS (FF)	1	1	Only header to recover FE-Buffer after NZS (Figure 7), frame size 28b (part of four-frames sequence)