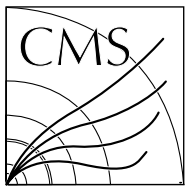
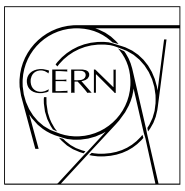
Available on CMS information server **CMS NOTE 2001/043**



The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland



February 22, 2002

User Requirements Document for the Final FED of the CMS Silicon Strip Tracker

Version 0.51

K.W. Bell, J.A. Coughlan, C. Foudas, K.A. Gill, N. Marinelli, G. Pasztor, I.R. Tomalin.

Abstract

This document specifies the functionality which will be required of the final Front End Drivers (FEDs) for the Tracker, and explains the motivation behind each of these requirements. Furthermore, it lists all the input/output signals which these FEDs must deal with. Also discussed is the issue of what subset of this functionality will be in the first version of the final FED (FF1), which will be used for rod/petal testing from late 2002. Not addressed in this document are details of the implementation. This will be described in a separate 'FED Specifications' document.

Contents

1 Document History

0.21 First circulated draft.

0.23 Draft History Section added. Introduction Section written in detail. Improvements to Sections on VME Interface and System Issues 2.

0.24 Foreword and Acknowledgements Sections added.

0.25 Testing, TTS and Data Format Sections significantly improved. Several other minor changes throughout.

0.30 Overhauled Frame-Finding, Resets, Calibration, Unresolved Issues.

0.31 Reacted to comments from Ian/Ken on entire note – minor changes throughout.

0.33 Incorporation of feedback from a wider community.

0.34 Added subsection on data compression.

0.41 Included important feedback from design engineers, proposal that DAQ should merge FED data, proposal that all TTS signals should be software ones, proposal to introduce processed raw data. Included useful feedback from many others throughout.

0.50 Standard CMS note format. Moved dynamic range section. Minor English corrections. Released as CMS note.

0.51 FED should also handle -ve going signals. (Sections 5 and 9.2.)

2 Foreword – Purpose of this Document

This document is intended to present the physicist users' view of the functionality currently believed to be required of the final Front End Drivers (FEDs) of the CMS Silicon Strip Tracker. It also explains the motivation behind each of the requirements. In spite of the title, our goal is to provide GUIDELINES for the design engineers, rather than rigid formal requirements. Our goal is also to enable other interested members of the CMS Tracker and DAQ communities to grasp some detailed knowledge of the envisaged FED functionality, and to elicit feedback from them before the design becomes too advanced. We also discuss what subset of this functionality will be available in the first version of the final FED (FF1), which will be used for rod/petal testing from late 2002. We do not address details of the implementation – these will be described in a separate 'FED Specifications' document, in part derived from this.

We envisage this to be a 'living' document during 2001, as feedback is incorporated from the Tracker and DAQ communities, and from the design engineers themselves. –

The latest version of this document can always be found at Ref. [1].

3 An Introduction to the FED

Signals produced by the silicon micro-strip tracker are initially processed by the front-end APV chips [2], each of which reads 128 silicon strips. The APVs amplify the signals and store them in a memory pipeline until the Level 1 trigger decision is made. If this decision is positive, the data from multiplexed pairs of APVs are transmitted along analogue optical fibre links to the Front-End Driver (FED) cards in the control room. The FEDs digitize the data, optionally perform zero suppression, and then transmit the data to the Data Acquisition (DAQ) system. A brief description of the FED is given in this Section. More details can be found in Refs. [3, 4]. An overview of the entire tracker readout chain can be found in Ref. [5].

Figure 1 shows the preliminary design of the final FED, which will be based on a 9U VME card. It reads 96 analogue optical fibres through its front-panel, with each fibre taking data from a pair of APVs. The fibres are grouped in eight 12-way ribbons, and, as shown in Fig. 1, the front-end electronics in the FED reflects this grouping. Data from *each* ribbon are treated by the FED as follows:

- The optical signals are converted to electrical signals by an opto-receiver package consisting of 12 p-i-n diodes.
- The data are digitized by commercial 40 MHz 10 bit dual ADCs, after applying a suitable delay skew to sample the data at the optimum time.
- The data are processed by an FPGA chip, which analyses each of the 12 channels in parallel. For each separate channel the FPGA performs the following functions:
  - The logic recognises the arrival of an APV-pair data frame. A frame consists of the data from the two APVs time-multiplexed together in alternate 40 MHz cycles. It begins with a digital header consisting of 2×3 successive high signals, followed by a pair of 8 bit APV pipeline addresses and a pair of error bits. This is followed by the analogue data from the 2×128 silicon strips.
  - Pedestal values stored in a look-up table are subtracted from the pulse height data.
  - The data are re-ordered such that they come in physical channel order.
  - On an event by event basis, the common-mode offset is estimated and subtracted from the data. Once this has been done, each data sample is reduced to 8 bits, by dropping the two most significant bits.
  - A cluster finding algorithm is run on the re-ordered data. Only those channels associated with a cluster will have their data transmitted to the DAQ. This substantially reduces the data volume.

  Optionally, the raw data can be output (albeit at a lower trigger rate), by not performing cluster finding or common-mode subtraction, and in addition (optionally) not performing re-ordering or pedestal subtraction.

After this initial processing, the data from the eight front-end FPGAs are collected together by an FPGA at the back-end of the FED. This buffers the data in an external memory, to cope with fluctuations in the rate, before transmitting the data over a high speed S-LINK [6] to the DAQ.

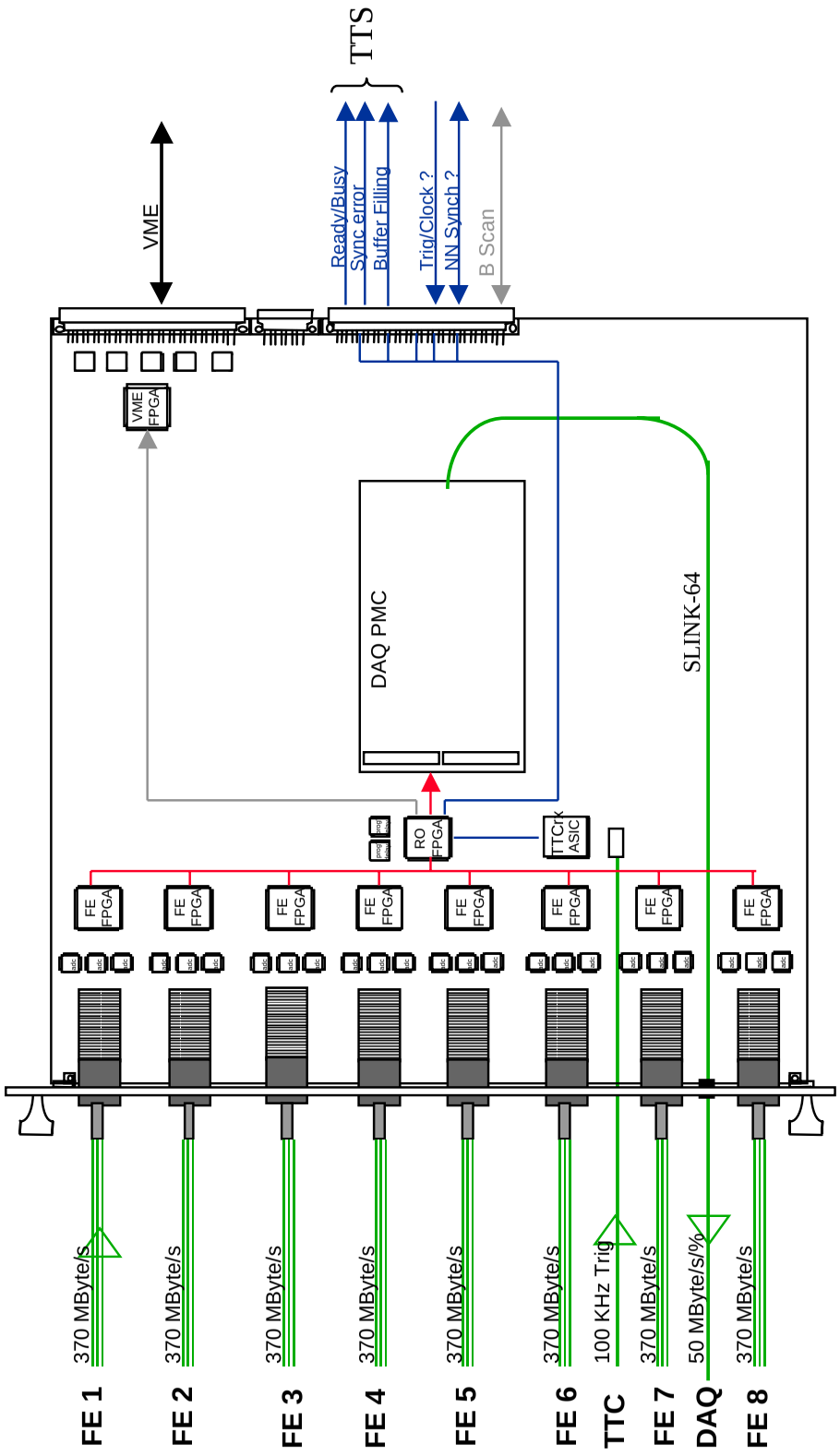


Figure 1: A somewhat out-of-date design of final FED. We now envisage that the SLINK driver will be on a transition module mounted on the rear of the backplane.

This FPGA also connects the FED via the VME back-plane to the FED crate controller. This VME path is used for controlling the FED, loading calibration constants and reading out data at a low rate for monitoring/calibration purposes.

The FED receives the 40 MHz clock and the Level-1 (L1) trigger via a digital optical link from the Trigger/Timing Control System (TTC) [7]. Although the triggers do not arrive synchronously with the APV frames, they arrive in the same order, so the two can be associated. The trigger number and the 'time' at which each trigger arrives are used by the back-end FPGA to label the event.

Fast signals about the status of the FED can be transmitted, via user-defined back-plane VME pins, to the Trigger Throttle System (TTS). Based on these signals, the run controller may decide to pause the trigger or reset the detector readout electronics.

One FED card reads the data from $96 \times 2 \times 128 = 25$k silicon strips. At a L1 trigger rate of 100 kHz, this corresponds to a data rate of 3 GB/s. After zero suppression this is reduced to 50 MB/s *per percent of tracker occupancy*. Roughly 450 FEDs will be needed to read out the entire tracker, and presumably a contingency of about 50 FEDs will be needed on top of this, to replace FEDs that might fail during the LHC lifetime.

Timescales. The first version of the final FED (FF1) will be used for rod/petal testing, and should be available in late 2002. A very small number of pre-production final FEDs (FF2) will be made. These should be available in late 2003. Production final FEDs (FF3) should start to be available during spring 2004, but release of all 500 fully tested FF3 boards may extend over 12 months.

3.1 Basic FED design assumptions

- All FEDs must be identical, to minimise design/maintenance costs and facilitate use.
- Each FED has eight input connectors, each reading a 12-way optical fibre ribbon. This ensures that one FED reads the data from one 96-way optical fibre 'cable'.
- A 9U VME board size. This is the minimum size which allows one to squeeze the desired number of input channels into one FED.
- The ≈ 450 FEDs are placed in VME crates in the Counting Room, with a maximum of 19 FEDs per crate.
- Each FED has no on-board micro-processor or network connection, and so must be controlled by an external crate controller.
- FEDs have no direct connection with the Front-End Controllers (FECs).
- FEDs can't directly command the APVs to stop sending data.
- FEDs must be able to process high luminosity proton–proton collisions at a 100 kHz L1 trigger rate with zero suppression enabled. They must also be able to process Pb–Pb events, without zero suppression, at the anticipated trigger rate of 1 kHz.

4 Operation Modes

Various options for data taking modes/formats are presented in this section. These options can't be selected independently for the 96 FED input channels, but only for the FED as a whole.

The FED can output its data in one of three formats:

1) Zero Suppression (i.e. reordering to physical channel order, subtraction of strip pedestals and common-mode noise, followed by cluster finding – normal FED operation, used for proton–proton and possibly for light ion collisions).

2) Processed Raw Data (i.e. reordering and pedestal subtraction, but no common-mode subtraction or cluster finding performed – used primarily for heavy ion collisions).

3) Virgin Raw Data (i.e. no reordering, pedestal/common-mode subtraction or cluster finding performed – used for testing, commissioning and calibration).

In a given run, all events must be taken in the same format. *Feedback from Design Engineer: Although it is desirable for some purposes to take occasional events in a different format, or even to output both raw data and zero suppressed data for the same event, this would require very large FPGA chips. Nonetheless, the 'Spy Channel' data taking mode described below offers these features for a* limited *number of channels. However, it is not yet certain that this can be included in the FED design.*

Spy Channel: Whilst data is being taken in one of the three formats mentioned above, one can simultaneously spy on the Virgin Raw Data of a small fraction of the events via VME (only). It is foreseen that a memory buffer will be available for each of the eight groups of 12 FED inputs, into which the Virgin Raw Data of one of these inputs for one event can be written. This data can then be read out via VME. The input can either be selected by simply looping over all 12 inputs in turn, or selected by VME. In principle, in the latter case, the choice could depend on the trigger type received down the Channel B TTC line. In view of the low VME readout speed discussed in Sect. 5, the data taking rates reading from 8 channels from all FEDs in a crate would be limited to 600 Hz. The above (or an enhanced) 'spy channel' capability is now considered essential.

The data capture operates in three possible manners:

a) Frame-Finding (i.e. data capture starts when the FED recognises that an APV data frame is arriving, and continues until the frame ends – normal FED operation).

b) Scope Mode (i.e. data capture starts when the FED receives a trigger, and continues for a programmable number of samples. No re-ordering, pedestal or common-mode subtraction, or clustering are done.) Setting up the tracker demands that one should be able to capture at least 490 samples (Sect. 12.1), but it would also be good to be able to take APV multi-mode data (see below), which demands a length of 840 samples. A safety margin should be added to these numbers, since the trigger and data frame are not synchronous. Ideally therefore, the number of samples should be programmable up to, say, 1020.

c) Multi-Mode (i.e. similar to frame-finding mode, except that, for each trigger, the FEDs must capture the three consecutive pulse-height samplings taken by the APVs when they

are in multi-mode – and output these as a *single* event to the DAQ). This mode is desirable, but not strictly required. It is not required that Zero Suppression be possible in Multi-Mode.

The FED can receive two kinds of triggers:

1. Hardware Triggers (i.e. from the TTC system – normal FED operation – or possibly also through the back-plane.). Available with all three data capture modes (a, b and c).
2. Software Triggers. Used for board testing and setting up. Unless the board is in Scope Mode (b), software trigger commands will be ignored, and should ideally result in some form of error message. *Question: Do we need the 'Test Mode' of the old PMC FED, to which the FED must be set before it will accept software triggers ?*

All triggers act at the next (positive-going ?) clock edge.

We need to specify formally, and in detail, what a FED should do on receipt of a trigger and on receipt of a reset.

5 The VME Interface

The FED is a VME slave, satisfying the VME64x (VME64xP ?) specifications, responding to A32?/D64 single/block read/write commands. The number of address lines required is likely to be dominated by the size of the memory used for buffering the output data. The upper five bits of the address will be taken by the board/geographic address. The FED will probably not be capable of issuing VME interrupt signals.

The FED VME interface, VME back-plane and crate controller should ideally cope with a data rate of at least 30 MB/s. 10 MB/s would allow the Virgin Raw Data needed for noise/pedestal calibration to be collected in 1 minute, but 30 MB/s would be needed to collect sufficient laser alignment events in 1 minute. 30 MB/s would allow up to three FEDs simultaneously to take Virgin Raw Data at 300 Hz during rod/petal testing before LHC startup, or a full crate of FEDs to take Virgin Raw Data at 50 Hz during physics running. An individual FED should be able to transfer data to the back-plane at this rate. Further discussion of required data rates is given in Sect. 15.

It is highly desirable that geographic addressing is possible. Use of geographic addressing would reinforce the concept that all FEDs are identical and ease installation/maintenance.

If conventional board addresses are used, care must be taken to avoid address clashes with other VME boards. However this should be easy, since FED crates are likely to contain only FED modules, a crate controller (or interface), and possibly a module for merging/transmitting the TTS signals.

As a general rule, any FED register which can be written to by VME should also be readable from VME.

5.1 VME commands to which the FED responds

- Switch FED off/on.

- Set data format: Zero Suppression / Processed Raw Data / Virgin Raw Data.
- Set data capture mode: Frame-Finding / Scope Mode / Multi-Mode.
- Send software trigger.
- Enable/disable hardware triggers.
- Enable/disable common-mode noise subtraction.
- Reset #BX.
- Reset #EVENT.
- FED Reset.
  (i.e. Flush buffers and reset error flags/counters, but don't reset #BX or #EVENT.)
- Full FED Reset.
  (i.e. Flush buffers and reset error flags/counters, and also reset #BX and #EVENT.)
- Send programmable, digital raw data events to the inputs of the FED front-end FPGAs. This is highly desirable, if not essential.
- Apply programmable, analogue voltage level to all 96 FED ADC inputs. Desirable. *Feedback from Design Engineer: Not realistically possible.*
- Send programmable test patterns or fake events to the S-LINK (and disable data flow from the 96 FED inputs). Again highly desirable, if not essential.
- Read/Write any of the following constants:
  - For each detector strip: pedestal, two cluster thresholds, dead strip flag and noisy strip flag. It is desirable that these can be sent either for an individual strip, or, with a block transfer, for all strips served by the entire FED.
  - For each FED input: a flag for each of the two corresponding APVs, indicating whether it should be read out or not. (Some APVs may be non-existent or faulty, and this flag can be used to shut off such channels.)
  - For each FED input: a flag for each of the two corresponding APVs, indicating whether its inverter is enabled or not.
  - For each FED input: its ADC sampling time-skew and frame-finding threshold(s).
  - For each FED input: a gain control flag indicating whether, in Zero Suppression mode, the MSB and LSB will be dropped instead of the two MSBs. This feature is desirable, but not required.
  - For each 12-channel FED laser receiver: the Opto-Receiver Module Configuration. This consists of 4 bits of DC input offset adjust and 2 bits of DC output offset adjust.
  - Number of samples to be captured in Scope Mode
  - There is *no need* for a programmable time delay in Scope Mode between the arrival of the trigger and the start of data capture. The request for 1020 samplings allows even multi-mode data frames to be completely captured, providing that the trigger reaches the FED no more than 2.5 μs before the APV frame.

- Read the following FED status information:
  - Board serial number (which also confirms that this *is* a FED). This should correspond to the FED bar code (see Sect. 13.5).
  - Board/geographic address.
  - Firmware version number.
  - Hardware version number.
  - READY, BUSY, THROTTLE, OUT_OF_SYNCH and ERROR signals.
  - Number of APV frames captured on each channel (#APV_FRAME).
  - Number of triggers received (#EVENT).
  - #BX of last trigger received.
  - Number of APVs in error (e.g. due to buffer overflow or desynchronisation). This information can be used by the Trigger Control System (TCS) to decide when the whole CMS detector must be reset.
  - List of which APVs are in error, also indicating the error type for each faulty APV (i.e. Desynchronization, #APV_pipe error or APV error flag set).
  - Buffer occupancies. This should cover all data buffers and event counter buffers.
  - Counter indicating, for each buffer, the number of APV frames which it lost due to overflow.
    Question: When may/does this get reset ?
- Specify which data should be sent to the crate controller/DAQ:
  - Enable/Disable data output to the DAQ.
  - Enable/Disable data output to VME.
  - Request that only every N^{th} event be sent to VME, where N is programmable from one upwards. (The counter N should be based on the value of #EVENT and hence should be reset if the FED receives an 'Event Counter Reset' ECR command.). With $N = 1$, this feature would be used for local data acquisition / calibration runs at low trigger rates. With $N \gg 1$, this feature would be used to monitor tracker occupancies and calibration constants during a physics fill. One must be able to specify whether these events are to be taken in virgin raw data, processed raw data or zero-suppressed format. (The same format could be sent to the DAQ for these events.) *Feedback from Design Engineer: All events within a run must be taken in the same format. The only mechanism for accessing data in a different format is the Spy Channel described in Sect. 4.*
  - It is desirable to be able to send Zero-Suppressed and Raw-Data for the *same* event to the crate controller to verify that the zero suppression algorithm works correctly. *Feedback from Design Engineer: This is not possible except using the Spy Channel described in Sect. 4.*
- Request that the FED creates and fills monitoring histograms, and makes them available to the crate controller. The histograms might be of occupancy or triggers versus bunch

counter '#BX' to verify that the tracker is in synchronization, or of common-mode noise, or of the digital levels, or of buffer occupancies. They should be (optionally ?) reset when the FED receives a RESET command. This feature is desirable. If it is not present, then the crate controller must be able to create these histograms itself (albeit with lower statistics), by periodically reading the appropriate variables from the FED.

Question: Which of the above commands need to be able to be issued without triggers first being disabled ?

6 Input Signals

- Eight 12-way optical fibre ribbons, reading a total of 96 pairs of APVs. Some APVs may be absent, in which case the corresponding optical fibres will be 'dark'.
- Trigger-Timing-Control (TTC): it is assumed that each FED has one TTCrx chip providing the clock/trigger from TTC channel A. The FED must also decode all TTC Channel B signals, and act on them according to the TCS protocols (not yet fully defined). N.B. RESET and HARD_RESET signals on Channel B are also sent down the Tracker TTC network as the sequence '101' on Channel A. The FED should optionally use this Channel A sequence to initiate a Reset and ignore the RESETs/HARD_RESETs from Channel B. *Question: This point needs further discussion with Sandro.* Hence the following signals must be extracted:
  - CLK (40.08 MHz clock)
  - L1A (Level 1 accept, i.e. 'trigger').
    (N.B. Illegal trigger sequences such as '111' or '110' should be ignored and optionally counted, while '101's should be interpreted as RESETs.)
  - #BX (12-bit bunch counter)
  - BC0 (Start of LHC orbit.)
    The FED can ignore this.
  - BCR (Reset #BX)
  - #EVENT (24-bit event counter)
  - ECR (Reset #EVENT and #APV_FRAME)
  - #ORBIT (Orbit counter.)
    The FED can ignore this, since the TCS system sends this information directly to the DAQ. In any case, #ORBIT must not be reset by a RESET or HARD_RESET command. Hence there is no way, within a run, to resynchronise #ORBIT if it is loses synchronisation.
  - START/STOP (Start/stop triggers.)
    The Tracker is a non-triggering sub-detector, and so the FED can ignore these.
  - RESET (Flush buffers and error bits/counters. Don't reset #BX, #EVENT or #ORBIT.)
  - HARD_RESET
    (*Question: Same as RESET, at least as far as the FED is concerned ?*)

  - TRIGGER TYPE
    This should be used by the tracker to transmit the emulated APV pipeline address. *Feedback from Design Engineer: The emulated address can't be sent to the FED in this way, since the TTCvi would require it to be made available synchronously with the trigger, and it can't be available this early.*
  - TEST_ENABLE (Indicates that the next trigger is for calibration.)
    (This may be used to indicate Tracker laser triggers.)
  - PRIVATE_GAP/PRIVATE_ORBIT (The next orbit gap/entire orbit can be used privately by the tracker.)
    Question: Will the tracker laser system or the pedestal/noise calibration use these ?
  - EVENT_TRACING (As yet ill-defined command, proposed by A. Racz, which may in fact be issued over VME, requesting that the next event also be written to VME, for debug purposes.)

- 'Back-pressure' from the S-LINK, requesting that the data output stream be suspended.

7 Output Signals

- Data sent to DAQ, with the following features:
  - One event must be sent for every trigger received.
  - Events must be sent in L1A order.
  - The processing time of an event must be $\ll 1$ second. (The exact limit is still to be specified by the DAQ group.)
  - S-LINK64:
    * 64 bits at a maximum rate of 100 MHz. The FED is allowed to run this link at a lower, more readily available frequency, such as 80 MHz, but this would reduce the maximum data transfer rate. *Feedback from Design Engineer: Optical or electrical S-LINKs are acceptable, but if the link is electrical, it must exit via the back of the FED.*
    * Primarily a push-only protocol. But the FED must adhere to the general protocols for DAQ data flow (to be defined), such as the 'back-pressure' signal that will be sent if the DAQ is unable to accept data.
    * Peak data rate 400 MB/s, sustainable for $\ll 1$ second (limitations determined by the DAQ link speed and buffer length – the latter not yet well defined).
    * Mean data rate < 200 MB/s, averaged over time-scale $\ll 1$ second, (limitation determined by DAQ switch speed and DAQ buffer lengths).

- FED data merging.
  – In order to balance the data flow to the (full capability version of the) Event Builder Switch, at 100 kHz trigger rate the data from up to 6 neighbouring FEDs must be merged into a single DAQ switch input. It is now assumed that *the DAQ group* will be responsible for doing this.

- Trigger-Throttle-System (TTS) signals sent to Trigger-Control-System (TCS) (these signals will be first grouped together within the tracker, before being sent to TCS):
  - READY ($= \overline{\text{BUSY}}$ unless the FED is powered off.)
  - BUSY (Not able to accept triggers, e.g. during a reset or when internal buffers are full.)
    – Requests a L1 trigger pause.
  - THROTTLE (FED buffers filling.)
    – Requests L1 trigger slow down.
  - OUT_OF_SYNCH (APVs out of synchronization or FED internal trigger counter buffer has overflowed.)
    – Requests a RESET.
  - ERROR (APVs out of synchronization and failing to respond to RESET commands, i.e. APV control block needs resetting via I2C write.)
    – Requests a HARD_RESET.

  N.B.: The signals OUT_OF_SYNC and ERROR are likely to be software signals generated by the tracker slow controls system, since synchronization errors are likely to build up slowly, and one will only wish to request a reset when a significant fraction of the tracker (e.g. 0.5%) is affected.

  The signal THROTTLE should also ideally be a software signal, since one would not wish to inhibit L1 triggers just because one or two FED buffers were full. However, this demands that the FED should not normally lose synchronization when its buffers overflow. This is feasible, and is discussed in Sect. 11.

  To simplify the tracker readout design, it would be desirable if READY and BUSY could also be software signals. It is currently envisaged by the DAQ group that BUSY must be a fast signal to indicate when the FED has recovered from a RESET. However, it may be possible to use a fixed time delay for this.

  In case of unforeseen eventualities, however, it would be prudent that the FED should also make hardware versions of all these TTS signals available, via user defined VME back-plane pins. The exact error conditions under which the BUSY, THROTTLE and OUT_OF_SYNC signals are sent are described in Sect. 11. The FED need not produce an ERROR signal, since only the VME crate controller can diagnose that the APVs need a hard reset.

- A front-panel CLOCK output (for oscilloscope) may be useful to verify TTC clock synchronization.

8 Output Data Formats

Event data from the FED will have the following general structure. The same data format is sent to both VME and S-LINK outputs. The final format must adhere to CMS standards and should be agreed at a later date with those responsible for developing the tracker unpacking code for the Filter Farm.

- DAQ Header: This is common to all sub-detector branches. Its exact content and format are now being defined. However, it will contain the FED identifier, #EVENT, #BX, data type, error code and CRC (cyclic redundancy code).
- FED header: This is specific to the tracker. It contains information such as the data format, (#APV_PIPE ?) and additional error detection (parity) or error flag information.
- FED data body: This depends on the data format.
  - For Virgin or Processed Raw Data it consists of the strip pulse heights stored as 10-bit data. Data from input channels flagged as 'dark' should be dropped.
  - For Zero Suppressed data, it consists of the address of the first strip in each cluster, the cluster width and the pulse height on each strip in the cluster. (If the mean cluster width exceeds two strips, as seems likely, then this is more efficient than outputting the address and pulse height of each strip in the cluster.) The basic format is similar to that described in Chapter 5.5.4.1 of Ref. [4], but expanded to 96 ADC channels. Data from input channels flagged as 'dark' should be dropped.
  - One should probably also be able to output 'Expanded' Data, to include, where applicable: 192$\times$ #BX_frame ?, #APV_pipe, APV error flags, common mode noise ? ....
  - The above FED data formats apply, where applicable, when the FED is in Frame-Finding or Multi-Mode (Sect. 4). In Scope Mode the FED will simply output up to 1020 10-bit digitizations per input channel.
- FED Trailer (optional).
- DAQ Trailer.

8.1 Data Compression

A detailed study of possible FED data compression schemes is presented in Ref. [8]. For Zero Suppressed Data a simple data compression scheme can be applied to the blocks containing the number of clusters per FED channel and the width of the clusters. The proposed variable length binary code is based on the principles of Huffman coding. A small Huffman code-table with 8 (or 16) entries is to be built by taking the 7 (or 15) most probable outcomes and an escape character to which the sum of the probabilities of the less probable outcomes is assigned. If a value not present in the code-table (i.e. its probability is not among the 7 (15) highest ones) is encountered, the code of the escape character is sent followed by the original 8-bit number without encoding. *Question: How difficult would this be to implement in the FED ?*

This compression scheme would reduce the data volume by about 25%. However, in view of the increased complexity it would imply for the FED design, and the increased sensitivity to errors associated with transmitting variable length data words, it is not clear that this feature should be required.

9 Frame Finding, Inversion, Clustering and Dynamic Ranges

9.1 Frame Finding

The APV data format is described in Ref. [2]. A frame consists of the data from the two APVs time-multiplexed together in alternate 40 MHz cycles. A frame begins with 6 successive logical 'high' signals and, if not immediately preceded by another frame, it will be preceded by 68 low signals preceded by a pair of ticks. In Frame-Finding mode, the FED recognises from the data-stream itself that a new APV frame has arrived.

Not all 96 inputs of a FED will necessarily be utilised – in the jargon, some fibres will be 'dark'. In addition, some inputs may, with time, develop hard errors. If told in advance (e.g. at start-of-run) of such dark or sick fibres, the FED must be able to cope smoothly with such inputs, and not generate error messages (even if all 12 fibres in a ribbon are flagged as 'dark').

It is possible that one APV of a pair may die. The dead APV can be identified in a dedicated start-of-fill calibration run as described in Sect. 12.4, and this information down-loaded into the FED. Frame-finding should then function correctly for the remaining APV and no error messages should be generated for the dead one.

The FED must verify that frames arrive simultaneously on all 96 inputs, within a specified tolerance. *Feedback from Design Engineer: They must arrive in the same 25 ns time bucket to simplify the design.* What it should do if they don't is discussed in Sect. 11.

It is desirable that the FED should be able to determine the header thresholds itself, during a brief 'FED calibration' run, and perhaps even track them during a fill. *Feedback from Design Engineer: This is foreseen.*

The FED must continue to recognise frames after RESETs have been issued to the Tracker. RESETs will upset the regular sequence of tick marks being emitted by the APVs every 70 clock cycles.

9.2 Inversion

The APVs contain an optional inverter stage which flips the polarity of the analogue signals, whilst leaving the digital header unaffected. Most APVs will have their inverters enabled, so that they produce positive going analogue signals. However, a few may have their inverters disabled, (to reduce their sensitivity to detector pin-holes,) so will produce negative going analogue signals.

After frame-finding, but before zero-suppression and data output, each FED must therefore be able to flip the polarity of the data from any specified APVs. This can be done by subtracting the ADC values from the maximum possible ADC value (1024).

9.3 Clustering

Clustering can be used only in Frame-Finding and Multi-Modes.

The proposed algorithm is described Ref. [9]. It forms clusters by accepting all strips whose pulse height exceeds the rms noise by a given factor, but applies a tighter cut to single strip clusters. Both cuts may be set independently for each strip in the tracker. *Feedback from Design*

Engineer: Where two clusters are separated by only one strip, the data of the intervening strip should also be accepted, since this simplifies the FED design. N.B. This is acceptable, since it doesn't increase the data volume.

Before cluster finding, the common-mode noise offset is estimated on an event by event basis from the median pulse height on the strips [9]. It should be possible to disable the common-mode subtraction if it turns out that common-mode noise is negligibly small.

It is desirable that noisy (or dead) strips identified by the crate controller can be flagged and hence not used during the common-mode noise estimation. It is also desirable that a second flag ('dead') should exist to label bad strips which, in addition, should never be output by the FED.

It should be borne in mind that these algorithms may need tweaking, or even replacing, in the future. Nonetheless, they can't get significantly more complex without seriously impacting on the cost of the front-end FPGAs.

9.4 Dynamic ranges

- Processed and Virgin Raw data: 10 bits.
  This ensures that the FED dynamic range maps onto that of the optical links (12.8 MIPs for a particle crossing 300 μm silicon at 90° [13], where 1 MIP is the most likely energy to be deposited in the silicon by a minimum ionizing particle). At the same time, it ensures that the rms noise will be at least two ADC counts for S/N (signal to noise) ratios up to 40. Digitization will not therefore add to the effective noise.

- Zero-Suppressed data: 8 bits. (Out of range data clamped.)
  After subtracting common-mode noise (which could pessimistically be several MIPs), the FED can drop the 2 MSBs, so reducing the dynamic range to 3.2 MIPs. This usually range ensures that the tail of the Landau energy loss distribution is not badly truncated. However, where tracks are expected to cross the silicon at a large angle (at the ends of the barrel), the signals will be larger and significant truncation will occur. It is therefore desirable that each of the 96 FED channels should be able to drop the MSB and the LSB, instead of the 2 MSBs, so increasing its dynamic range by a factor of two. (The individual channel gain settings for a given run would, of course, have to be flagged, e.g. in some database.)

- Pedestals: 10 bits/detector channel are required.
  Although the pedestals are expected to show little dependence on channel number, they must be stored in a 10-bit register to allow digital test data to be injected into the front-end FED logic, as described in Sect. 10.1.

- Clustering cuts:
  The clustering algorithm has a high and a low threshold per detector channel. Ideally, these require 8 bits each. However, 6 (5) bits for the high (low) thresholds would be tolerable, as it is unlikely that one would wish to cut harder than 0.8 (0.4) MIPs.

- Common-Mode algorithm cuts:
  If the common-mode is determined from the median pulse height, no cuts are required.

(If the mean is used, then one cut is needed per detector channel to reject strips containing a signal. This would ideally require 8 bits, but 5 could be tolerated.)

- Dead or Noisy channels:
  These could be flagged by setting their pedestal to, e.g., 126 or 127 respectively.

10 Hardware Requirements for FED Diagnostics

The entire FED readout system involves $\approx$ 450 9U VME boards, reading out some 10 million channels of 10-bit analogue data. This large scale system must run reliably and must be testable in order to guarantee the integrity of the physics data. To ensure this, test features must be built into the FEDs, both to facilitate testing during the development phase, and to check that they work correctly once installed in the CMS Counting Room. In short:

- It must be possible to *verify* that the FEDs function as designed. This must be feasible using software interacting with the FEDs via the VME backplane, and which, when run, should verify within $\approx$ 10 minutes that all FEDs in the system are performing correctly. The board design should thus allow the internal functions of the FED to be tested comprehensively using external software. The test software will not be run during physics data taking runs.

- It must be possible to *check* that the FED zero suppression algorithm performs as expected on an event by event basis.

- The user must be able to *shut off* data paths on one or more FEDs under VME control if (s)he decides that the data in those paths are corrupt (e.g. noise or any form of data corruption).

An external test rig must also be produced, to allow the FEDs to be tested immediately after manufacture, and also to allow faulty FEDs to be debugged once CMS is running. This Section considers both the test features which must be built into the FED and of what the test rig might consist.

10.1 Inbuilt Test Features

- Boundary Scan. (Primarily to check the electrical connection between chips. *Question: Can it also check internal chip functionality ?* Boundary scans can be performed when the FED is in a dedicated test rig. It is desirable that they can also be performed *in situ* in CMS (perhaps after intervention to plug in diagnostic equipment). This requires that the boundary scan loop should be accessible via user defined VME pins.)

- *Question: Do we need an* internal *40 MHz clock, to test FED with software triggers without TTC ?*

- *Question: Do we need* external *(back-plane) clock and trigger inputs to test FED without TTC ? N.B. This would demand an external module to generate clock and trigger.*

- Ability to check that FED output is correct for a known input. There are a few simple possibilities:
  - Request the APV to send calibration events to the FED, and verify that the FED forms clusters from the detector channels which were pulsed.
  - Set pedestals for some detector strips to values much lower than their true values and then take random triggers (without beam). These channels should be identified as clusters by the FED. *Feedback from Design Engineer: It is better to skip the pedestal subtraction and instead simply set the digitized data equal to the values stored in the pedestal memory buffer. This provides a* digital *test.*
    Some such form of digital test is considered essential. And it must possible to perform such a test when no APV inputs are connected to the FED. *Question: But how does the header finding work in such a case ?*

However, these test features may not suffice and more sophisticated tests are proposed. All the testing requirements can be accomplished by introducing VME-accessible SRAM lookup tables in several critical points of the data path in the FED. The user should then be able to write test vectors in the input buffers of a path and verify that the expected result appears at the output buffers. We envisage the following stages of VME read/write lookup tables:

1. A first set of lookup tables should allow one to inject *digital* raw data test events (i.e. fake APV frames) into the inputs of the front-end FPGAs. Each of the 96 inputs channels should be individually programmable. A typical user program would loop over various test events and verify that they propagate correctly to the VME and S-LINK outputs. For this test feature to function correctly, it would be necessary to trigger the FED somehow for each test event. *Feedback from Design Engineer: This test is replaced by that using the pedestal memory buffer mentioned above.*

2. The second stage of lookup tables should be between the back-end FPGA and the S-LINK (or in the back-end FPGA itself). The tables can inject bit sequences or 100 kHz test events into the S-LINK to verify that it works correctly. *Question: Is the test data programmable or pseudo-random ?* If a FED appears not to be working, this test will be necessary to establish whether it or the S-LINK are at fault.

3. A VME programmable *DC level* produced by a 12-bit DAC, should be injectable at the analogue inputs of the ADCs to test the digitization part of the FEDs, by performing linearity tests. Past experience has shown that such tests are very sensitive to digital noise on the board. *Feedback from Design Engineer: This is not realistically possible.*

4. Via VME one can request the opto-receiver to move its analogue baseline, to help verify whether the opto-receiver and ADC are alive. This test will help determine whether an input stage of a FED is sick, or whether an input fibre is sick. The baseline shift available is expected to cover the full 8–10 MIP range.

5. *Question: Xilinx internal test program ?*

6. Protruding pins or loops to which one can attach an oscilloscope to monitor certain signals such as: TTCrx clock and trigger output, ADC clock input and data output and S-LINK input.

To verify the cluster finding algorithms, it would also be extremely useful to be able to output *both* Raw and Zero-Suppressed data for some events. *Feedback from Design Engineer: This is possible using the Spy Channel described in Sect. 4.*

10.2 External Test Rig

This rig will be used to test ≈ 500 FEDs during (pre-)production. When the FEDs are shipped to CERN, the rig should probably be duplicated at CERN. *Question: How should the testing be done ?* Proposal:

An Arbitrary Function Generator generates a 40 MHz clock and emulates a trigger sequence. Associated with this sequence, should be at least two independent sequences of APV data frames. The APV frames are sent to electrical-optical converters and fanned out to the 96 FED inputs. The two independent sequences are required to check that the FED detects when some APVs are out of synchronization. Thus, one sequence of frames might be sent to 11 fibres of each 12-channel ribbon, while another sequence would be sent to the remaining fibre of each ribbon. The trigger and clock signals are passed to a TTCvi, which transmits them optically to the FED. The data produced by the FED is read out both over the S-LINK and over VME (although it's not necessary that the S-LINK be commissioned and tested until 2003). The TTS signals produced by the FED must also be monitored, both via the VME interface and, for those *hardware* signals being generated by the FED, via some dedicated board.

Alternatively, one could try to use the same system proposed to test the Pixel FEDs.

11 Errors: Detection, Recovery and Resets

11.1 Errors

Possible errors which a FED can detect and the necessary reaction to them include:

- FED buffer errors:
  - If a FED data buffer overflows, it must continue to output empty events so as to preserve synchronization. The events must be flagged to show which data were lost. In p–p collisions, the down-time of each data buffer as a result of its overflow in a high occupancy FED (i.e., $\approx 3\%$ strip occupancy) should be less than 0.1%. This ensures that such errors will not significantly affect the tracker efficiency. In Pb–Pb collisions, since the data will be taken in Processed Raw Data mode, the data buffers in all FEDs will overflow in phase. The lost events would therefore contain no tracker data. The fraction of events affected by this problem should also be less than 0.1%. *Question: Is it possible to prevent this happening by careful choice of Pb–Pb trigger rules ? Question: What happens in light ion collisions, since these are likely to be taken in Zero Suppressed mode ?*
  - The FED trigger buffers (used for storing #BX and #EVENT) must be long enough that they 'never' overflow. *Question: Can this be guaranteed ? What happens if the S-LINK dies or asserts its back-pressure signal to prevent data being sent from the FED to the DAQ ?* If the trigger buffers do overflow, all FEDs lose synchronization

and should stop outputting data. The problem should be detected via VME and a RESET requested. (In principle, by counting the number of 'lost' triggers, one could preserve synchronization. However, the DAQ would already have rejected the tracker FEDs when they first stopped outputting events, so this would not help.)

  - As a safety net, it is required that hardware THROTTLE and BUSY fast signals should be output by the FED via user defined VME pins, to slow or stop the L1 trigger respectively. These may be needed if it turns out that the trigger buffers do overflow for some unforeseen reason etc. For the same reason, a hardware OUT_OF_SYNC fast signal is required to indicate serious loss of synchronization due to trigger buffer overflow.
  - For each data buffer in the FED, the FED must count the number of APV frames which it is unable to store. These counters allow diagnosis of which buffers are having problems. They would require 32 bits in order not to overflow during a (very bad !) 10 hour fill. The counters should be readable by VME and should be reset by a VME clear command and (optionally ?) a RESET signal. *Question: If buffer errors only occur immediately before resets, and RESET resets the counters, can they ever be detected ? This question also applies to APV synchronization error flags.*
  - For each buffer in the FED, the crate controller should be able to read its occupancy.

- Within a FED, some of the 192 APV pipe-line addresses #APV_pipe disagree.
  – The FED should:

  1. Decide which APV addresses are likely to be wrong, and flag 'bad' APVs in the output data streams, both to the DAQ and to the crate controller.
  2. Suppress the pulse height data from these APVs (but (optionally?) insert the 'bad' #APV_pipe address into the data, as a debugging aid). It is desirable that one could optionally also keep the *data* from 'bad' APVs, to understand better what is wrong with them.
  3. Periodically, allow the crate controller to check how many APVs are wrong and also check which APVs are wrong.
  4. Optionally allow the crate controller and DAQ to read out all 192 #APV_pipe's.
  5. Compare #APV_pipe with the emulated pipe-line address available from the TTCrx. *Feedback from Design Engineer: The emulated pipe-line address can't be sent via the TTCrx.*

- Within a FED, not all 96 APV frames arrive simultaneously (i.e. within a specified tolerance, which could be zero). *Feedback from Design Engineer: The design is simplest if the frames are required to arrive in the same 25 ns time bucket.* The FED must also be able to cope smoothly with worst-case scenarios, such as no frames at all arriving from a particular channel or even from a 12-way ribbon, or channels outputting additional spurious frames.
  – The FED should:

  - React analogously to #APV_pipe errors. If the FED has detected a header of one channel, but shifted in time, it would be nice if the FED could somehow report whether it's early or late, and by how many clock cycles.

- Within a FED, some of the 192 APV error bits show their APV to be in error.
  – The FED should:
    - React analogously to #APV_pipe errors.
- If triggers are lost by the FED, the DAQ would detect that the #BXs of the following events were wrong.
- If clock cycles are missed by the FED, then the value of the FED's bunch-crossing counter will not be equal to the number of bunches in an orbit when the Bunch Crossing Reset signal BCR is received. This error should be noted via VME.
- In the highly unlikely scenario that all APVs miss the same trigger, but the FEDs receive the trigger, this could only be detected when the APV buffers are known to be empty, e.g. during a RESET sequence. An error could be inferred if #EVENT is not equal to #APV_FRAME. This error should be noted in a register where it could be read via VME. This feature is desirable, but not required.
- A FED dies.
  – One should do nothing. The DAQ switch will time out for that FED.
- A FED outputs crazy data.
  – One should be able to disable event data output to the S-LINK and VME and also disable the TTS signals. This may just imply switching off the FED by software (*Question: but can we do that dynamically within a run ?*).
- The VME crate controller dies.
  – The FED should continue sending events to the DAQ as usual (i.e. the FED must not hang because some monitoring buffer is not being emptied by the crate controller).
- The TTC clock input stops.
  – The FED should not suffer permanent damage and should restart when the clock does. The FED should assert ERROR while the clock is stopped (and stay in ERROR until a RESET has been issued ?). Nevertheless, one should still be able to read the FED's other status parameters via VME.

The FED crate controller can detect errors such as:

- Histograms produced by the FED of occupancy or trigger numbers versus #BX showing the wrong pattern of filled/empty bunch crossings.
  – The tracker is out of synchronization. If problem persists, warn the shift crew or request a reset.

11.2 Resets

During a normal RESET sequence, to keep it quick, no network access by the readout electronics is allowed. *Limited* network access is allowed during a HARD_RESET. A draft timing diagram of the normal Reset sequence is available from Joao Varela. In short:

- At the start of an orbit, inhibit broadcast of triggers.
- Wait a fixed time, agreed to be long enough to empty sub-detector front-end (*Question: and back-end, e.g. FED ?*) readout buffers.
- Broadcast a RESET command via the TTC system.
- Wait for at least 16 clock cycles and for all BUSY signals to disappear. Then broadcast ECR (Event Counter Reset) via the TTC system.
- (As usual) broadcast BCR Bunch Counter Reset at the start of the next orbit (which could be very soon afterwards ?).
- Immediately re-enable acceptance of triggers.

On receiving a RESET, the FED should:

1. Assert the BUSY signal on TTS. *Question: What if it doesn't have a hardware BUSY signal ?*
2. Send data already in the FED buffers if possible.
3. Send empty events for triggers without APV data.
4. Clear all buffers, error flags and counters, except #BX, #EVENT and #APV_FRAME. *Question: Really clear all of the error flags ?*
5. De-assert the BUSY signal.

12 Systems Issues 1: Setting Up, Timing, Calibration

Setting up the system and calibrating it involves several steps, each of which imposes its own requirements. Many of these demand data readout via VME, so that the calibration can be done quickly and locally in the crate controller.

12.1 Setting up the Optical Links

This is based on measuring both the baseline and the amplitude of tick marks from the APVs. It therefore demands:

- Scope Mode, since no APV data frames are required to be present.
- Software or hardware triggers can be used, as FED designers prefer.
- If hardware triggers are used, then ideally the FEC should prevent these triggers being sent to the APVs, since otherwise an APV data frame would be captured in addition to the tick marks. If the FEC does not have this functionality, the calibration software would need to reject explicitly the APV frame.
- The FED should capture at least three tick marks per trigger. If a data frame is also present, then Scope Mode must be able to capture at least $2 \times [(12 + 128) + (3 \times 35)] = 490$ samples (or 280 if the APV did not produce a data frame).

12.2 Synchronisation Issues

The *internal* synchronization of the tracker at LHC start-up could be based upon measuring the relative arrival times of APV ticks at the FED inputs. The procedure is described in Ref. [10]. It requires the same FED functionality as does setting up the optical links, and could be done with either hardware or software triggers to the FED. However, it does impose additional constraints:

- If software triggers are used, then as these will not arrive simultaneously at all FEDs, the FEDs will not all begin data capture at the same moment in time. In order to achieve synchronization from FED to FED, the FED must record the #BX at which the software trigger arrives.
- Alternatively, if hardware triggers are used, it is again desirable that the FEC should prevent these reaching the APVs.
- The delays in the FED's TTCrx chip should be used to compensate for previously measured variations in TTC fibre lengths to the different FEDs. It may be useful if a clock output (for oscilloscope) were available on the front-panel of each FED, so that one could check that all FEDs receive the clock synchronously after this correction. (Especially, as it is possible that TTCrx chips might not all be identical and could hence introduce a varying delay on the clock).
- In Scope Mode, if a hardware trigger reaches all FEDs at the same moment in time and if all internal FED delay skews are zeroed, then data capture should begin at the same moment in time on all FED input channels. The tolerance on this should be less than 1 ns. This implies that delays introduced by chips inside the FED should not vary from FED to FED and also that when fanning the clock out to the FED front-end ADCs and FPGAs, the signal propagation times should be the same to all of them.
- Each of the 96 FED inputs channels must have a programmable delay skew. This allows the FEDs to compensate for relative differences in the analogue optical fibre lengths between the APVs and the FEDs, so that the APV data can be sampled at the optimal moment on all channels. Independent delays are not needed for the two APVs associated with each FED input, since they reside on the same detector module. A 1 ns step size in the clock skew is strongly desirable, to match that of the PLL delays of the front-end hybrids, so that APV and FED delays can be moved in phase. Nonetheless, a 2 ns step size would be tolerable. The delay skews must be accurate to better than 1 ns. They should have a ± 12.5 ns range corresponding to allow for a spread in the analogue fibre cable lengths to a given FED of up to 5 m [11], and an additional range of ± 12.5 ns to allow one to reach the optimum sampling point. The range should therefore be ± 25 ns in total.
- It would facilitate FED use, if FEDs could determine the delay skews needed for each channel themselves, during a special 'FED calibration' run. However, this feature is not essential.

Once the LHC has started, it will be necessary to adjust the beam crossing time for the tracker as a whole relative to the timing of the L1 trigger. This will be done using tracks. The procedure

has not yet been determined. However, it is likely to involve maximising the S/N ratio for clusters from high momentum tracks. This could be done in Processed Raw Data mode.

Histograms of tracker occupancy versus #BX will allow crude checks to be made as to whether the tracker is in synchronisation, to the nearest filled bunch. But to do so, one has to know what to expect. Although still evolving, the presently envisaged bunch time structures for proton–proton and heavy-ion running can be found at Ref. [12]. It's not clear whether a scheme for light ion collisions has yet been worked through in this detail. Note that, from the symmetry of the machine, for CMS and ATLAS the *collision* time structure is the same as the *bunch* time structure, although CMS has not yet defined which bunch it will consider to be bunch #1.

12.3 Determining FED Digital Thresholds

The thresholds required by the FED, to distinguish logical High and Low states in the APV digital output, are determined from the baseline and amplitude of tick marks taken in Scope Mode. The requirements are identical to those for setting up the optical links. It would facilitate FED use if the FEDs could determine the thresholds needed for each channel themselves, during a special 'FED calibration' run. However, this feature is not essential.

It is very desirable to *monitor* the digital 'low' and 'high' levels during a fill, since they may drift as a result of instabilities in the laser drivers. This could be achieved by histogramming the ADC values of the APV header. *Feedback from Design Engineer: It is foreseen that the FED will automatically follow these levels during a fill, using a running average.*

12.4 Flagging Dead APVs

If one APV in a pair dies, then for Frame Finding to work successfully on the other APV, a flag indicating which of the two APVs has died must be downloaded into the FED. Dead APVs would be identified during a calibration run before the start of each fill. Tick marks would be taken in Scope Mode and, assuming that all FED input channels had been previously synchronized, the dead APVs could be seen by comparison of the tick marks on different FED inputs. If the tracker had not yet been synchronised, one would need to compare tick marks from APVs of the *same* detector module.

12.5 Noise and Pedestal calibration

This is based on measurements of the mean pulse height and the rms variation about that mean, for each detector strip. It therefore demands:

- Frame-finding mode.
- Virgin Raw data.
- Random triggers sent to both the APV and the FED, preferably in the absence of beam, but with the detectors switched on.
- If the noise and pedestals are to be monitored during data taking (which would be prudent), then occasional Virgin Raw Data events must be interspersed with the normal

data. These can be taken with Physics Triggers. *Feedback from Design Engineer: This data must be taken using the Spy Channel described in Sect. 4. Question: Should the tracker occasionally request permission to send 'calibration' events during the private orbit gaps ? One would have time to prepare and recover from such events, so could perhaps more easily switch into Virgin Raw Data mode for them. Also, the absence of signals on the silicon strips would make monitoring of the calibration constants more precise. N.B. Assuming their triggers are local, such interleaved events are NOT allowed to go to the DAQ.* It is acceptable that calibration constants could only be updated *during* a run after first pausing the triggers.

It is desirable to monitor the common-mode levels during a fill. The FED should histogram these or make them periodically available to the crate controller.

12.6 Measuring APV pulse shapes

This is done by sending calibration triggers to the APV. It therefore imposes the same requirements as noise and pedestal calibration, except that it will only be done outside physics running. It can be done most quickly if FED Multi-Mode is available. However, this is not essential, as Frame-Finding mode could also perform the task.

12.7 Alignment using Laser Events

In special dedicated runs, outside physics data taking periods, laser beams will pulsed through the tracker to measure the relative alignment of the detector modules. These laser runs raise several issues:

1. The FED must take the laser events in Processed Raw Data mode. This is because some of the laser-induced signals could be small and fail the standard cluster finding cuts, and because clusters produced by the laser beam may be as much as 100 strips wide, and this would upset the common-mode noise subtraction used in Zero Suppression mode.

2. It is not yet decided if local or global triggers will be used. However, in the former case the data would need to be read out via VME, which would reduce the maximum trigger rate from 2 kHz to 50 Hz.

3. The event header produced by the FED should identify the event as a laser event.

4. Only about 1500 of the APV chips can see signals produced by the laser beams. Substantial reduction in data size could therefore be achieved by only outputting the raw data for these APVs. One could probably achieve this by disabling at start-of-run the FED inputs from the other APVs. This is desirable, as it would increase the sustainable laser trigger rate by a factor of 50. However, since only a few thousand laser events are needed, this does not seem important.

The possibility of taking laser events during physics runs is also under investigation. This would raise additional points:

- The triggers accompanying these laser events would have to be identified using a flag sent down the Channel B TTC line. This might be TEST_ENABLE, but this is not yet decided.
- If the trigger were global then:
  - The laser event must be flagged in the event header since it would be sent to the DAQ.
- If the laser trigger were local to the tracker partition(s), (which appears an unnecessarily complicated thing to do,) then:
  - The event should only be output via VME.
  - The counter #EVENT should not be incremented.
  - The laser event must be flagged in the event header, to distinguish it from 'normal' events being sent to be monitored in the VME crate controller.
  - *Question: How does the FED know the trigger is local ?*
- On receiving a laser trigger, the FED would need to take the corresponding event in Raw Data mode. *Feedback from Design Engineer: This is not possible. However, one* might *be able to collect the raw data using the Spy Channel described in Sect. 4. It may even be possible to select, via VME, which FED input channels from each group of 12 should be read out in laser events.*
- In view of the low VME readout speed discussed in Sect. 5, the raw data taking rate via the spy channel is limited to 8 channels from every FED in a crate at 600 Hz. As most of this would be required to monitor noise and pedestals, the laser data rate should be limited to 50 Hz.

13 Systems Issues 2: Crates, Power Supplies and Sundry Issues

13.1 Crates

The FED will use 9U-400 mm board size. This is the minimum size which allows one to squeeze the desired number of input channels into one FED. The $\approx$ 450 FEDs are placed in VME64x [VIPA?] crates, with a maximum of 19 FEDs per crate. *The choice of crate should be made after consultation with P. Sharp, with a view to using a crate approved by CMS.* A crate would only contain FEDs from a single tracker partition.

13.2 Safety requirements

Optical fibres enter the front-panel of the FED. The FED optical patch panel is classed as a k*3A laser hazard level. Some form of lockable cover (possibly attached to the racks) will be necessary to prevent optical connections from being accidentally unplugged. The FEDs, or the cover, and/or the ends of the optical cables will be labelled with the appropriate laser safety markings.

The Counting Room area containing the FEDs will be classified as an area of 'controlled access', restricted to authorised people who have followed the appropriate laser safety training.

13.3 Power Supplies and Fuses

Past experience shows that a common fault condition in modules such as the FED is likely to be a blown fuse. This can just be a result of ageing of the fuse, but can also be caused by temporary surges in the power supply (e.g. due to storms). Anything reasonable that can be done to reduce the risk of blown fuses and to make recovery as fast as possible, should be done. Possibilities might include adequate surge protection for the power supplies and/or the FEDs, use of long-life and slow-blow fuses, use of circuit breakers with front-panel reset instead of fuses, or placing fuses somewhere where they can be replaced without completely removing the FED from the crate, etc.

Required voltages, currents and ripple of power supplies will be specified in the 'FED Specifications' document.

13.4 Replacing Faulty FEDs

When faults occur, it is inevitable that FEDs will need to be removed from a crate, either for repair or to test the effect of swapping them with other FEDs. It is highly desirable that this can be done without powering off the crate, since this could reduce the down-time from hardware interventions, since past experience shows that powering off and back on a crate can itself often induce other faults. It should therefore be possible to 'hot-swap' FEDs.

It is also highly desirable that a FED can be removed from a crate and re-installed on an extender board, without disconnecting its optical inputs and any front-panel outputs. This demands that trays be available below each crate to store some slack in the optical fibre ribbons. One should also be able to remove FEDs without first actively disconnecting S-LINKs at the back of the crate. This suggests that they should not be connected directly to the FEDs, but should instead go via VME transition modules.

13.5 FED Serial Numbers and Bar Codes

Each FED board should have a unique serial number, visible from the front panel. *Should* this take the form of a bar code, then the format must conform to the CMS convention: 14 decimal digits, beginning 3 [for CMS], 02 [for TK], followed by a 4 digit code saying the board is a FED, followed by a 7 digit board serial number.

13.6 Indicators

Front-panel LEDs are desirable to show (at least!) (i) Power on (one LED per voltage rail?), (ii) FPGA loaded, (iii) VME activity, (iv) clock present, (v) triggers, (vi) some APVs in error, (vii) FED buffers filling/full, (viii) clock error occurred since last RESET. If there is room on the front-panel, text to indicate what each LED signifies would be welcome ('replaceable' text would allow the LED functionality to be changed, subsequently).

13.7 Environment

The FED racks will be in the underground Counting Room. Radiation levels here are low enough to allow access by radiation workers at all times. However, a small magnetic field

($\approx$ 0.01 Tesla) will be present, so care should be used if magnetic circuit breakers etc. are to be employed.

14 Requirements for FED Support

- Software should be written to allow easy use of FED. It is likely that the driver software will have to run under Linux.
- All hardware, firmware and software should be documented. The documentation should be clear enough that it can still be understood in 2015, even in the pessimistic scenario whereby all current experts have left CMS !
- All firmware should have a version number, which is readable via VME.
- CLRC Instrumentation Department should agree to maintain/repair FEDs after production until the end of the CMS (2020?). The FEDs must be covered by the Instrumentation Department Quality Assurance system.

15 FED Demands on Crate Controller

The crate controller must:

- Be robust, since it controls several percent of the Tracker. If it crashes, this must only affect the monitoring, and must not interrupt data-taking. Rebooting it must not interrupt data-taking. We may want its VME interface to be password protectable, to avoid unwarranted VME access.
- Set FED options. (Scope mode, raw data, thresholds . . .)
- Process setting-up and calibration data.
- Cope with the requisite data rates, including the following tasks:
  - Determine and store noise and pedestals before every fill. This should be done in less than 1 minute and will require 1000 Virgin Raw Data events. The crate controller must therefore read the events at least 17 Hz. If the controller serves 19 FEDs in a crate, this corresponds to a minimum VME throughput of 10 MB/s. The crate controller must determine the calibration constants for the corresponding $\approx$ 500k silicon strips in roughly one minute, too. This can be done on present day computers.
  - If each fill is preceded by a short laser run, this would require 3000 Processed Raw Data events, and would also have be done in under a minute.

    The crate controller would therefore read the events at least 50 Hz. If the controller serves 19 FEDs in a crate, this would correspond to a minimum VME throughput of 30 MB/s.

  - Take data during local data acquisition runs at CMS or during rod/petal testing before CMS. When controlling a full crate of 19 FEDs, a Virgin Raw Data taking rate of 50 Hz is acceptable (although more would be welcome), as would be provided by the 30 MB/s data rate already required above.
  - When controlling only 1–3 FEDs, as in rod/petal testing, a Virgin Raw Data taking rate of 100 Hz is required, which imposes no new requirements on the crate controller.

- At start of fill, down-load calibration constants and clustering cuts to FED and to the off-line database.

- Record the number of APVs which have lost synchronization/pipeline errors, so that some higher intelligence may assert OUT_OF_SYNC to request a Reset, if the loss of synchronization becomes too significant. This monitoring should be done about once per second.

- Provide monitoring/error reporting, e.g. histogram occupancy and number of triggers as a function of #BX, to check that the tracker is in synchronization; histogram APV pipeline distributions; provide histograms which will detect set or dropped bits, particularly in Virgin Raw Data from the ADCs.

- Occasionally monitor the pedestal 'finger-prints' of its associated APVs, in order to (i) detect if someone alters the cabling, (ii) help confirm, if one APV in a pair dies, which one is still alive.

- Monitor the TTS signals coming out of the FEDs, and pass on any such software signals to some higher authority.

16 FED Demands on other Parts of CMS

- On the **TRIGGER**: Specific (tighter) trigger rules are required for Pb–Pb collisions to prevent FED internal buffer overflow.

- On the **CABLE LENGTHS**: Triggers must arrive at the FED before the APV frames. *Question: Is this necessary ? Can it be guaranteed even during rod/petal tests ? What does the PMC-FED assume ?*

- On the **FED VME racks**: Cable trays are required in the FED racks to provide some slack in the optical fibre ribbons, such that FEDs can be removed without unplugging them. And the rack system must support the requirements imposed by the laser hazard classification of the FED optical inputs.

- On the **FECs**: If synchronization is done with hardware triggers to the FEDs, then the FECs should suppress triggers going to the APVs.

- On the **FECs**: It must be clarified whether the '101' Reset sequence is generated by the FEC or in the TTC system.

- On the **DAQ**: Merge the data of 450 FEDs down to however many DAQ links are available to the Tracker.

- On the **DAQ**: When a FED dies, the DAQ switch will have to time out cleanly for that FED, and be able to live without it until it recovers. If the FED returns to life following a RESET, the DAQ switch must accept its data once more.
- On the **tracker reconstruction code running in the DAQ computer farm**: This must check if the preferred APV pipeline address output by each FED agrees with those of all the other FEDs, and flag an error in the data where this is not true.

17 Minimal Requirements for Rod/Petal Testing

The first version of the final FED (FF1) will be used for rod/petal testing from late 2002 onwards. As it is unrealistic to implement all the functionally of the final FED by this date, it is practical to ask which functions are **NOT** required for this task. These are:

- Zero suppression / Processed Raw Data
- 10-bit data output in Virgin Raw Data Mode. – It could instead be padded out to 16 bits.
- S-LINK
- Multi-Mode
- Option to output only every N^{th} event to VME.

It should be noted that, although the above functions are **not** required for module/rod/petal testing, any corresponding *hardware* implementation should already be present and functional on FF1. Then, as time permits, the corresponding FPGA *firmware* implementation can be added, even if this is after distribution of FF1s to rod/petal testers. Thus the FPGA chips must be large enough that these missing features can be added to FF1, to continue the development of the final FED. It is foreseen to test the S-LINK readout with FF1 in 2003.

18 Unresolved Issues

- *The CMS trigger rules in both proton–proton and heavy-ion collisions should be spelt out somewhere in this document.*
- *We should probably add a dedication section somewhere about timings.*

19 Acknowledgements

We wish to take this opportunity to thank the many people who have recently had to put up with our stupid, and occasionally not-so-stupid, questions, particularly Wim Beaumont, Sergio Cittolin, Emlyn Corrin, Marcus French, Bill Gannon, Geoff Hall, Rob Halsall, Greg Iles, Attila Racz, Wesley Smith, Joao Varela, Francois Vasey, Bruno Wittmer ....

References

[1] K.W. Bell *et al.*, *"User Requirements Document for the Final FED of the CMS Silicon Strip Tracker"*, http://hepwww.rl.ac.uk/CMS_fed/Final_FED_User_Req/fed_urd.pdf .

[2] RAL Micro-electronics Design Group, *"APV25 User Guide"*, http://www.te.rl.ac.uk/med/ and http://www.te.rl.ac.uk/med/projects/index.htm .

[3] J. Coughlan *et al.*, *"Design of the Front-End Driver Card for CMS Silicon Micro-strip Tracker Readout"*, Proceedings of 6^{th} Workshop on Electronics for LHC Experiments, Krakow, CERN/LHCC/2000-041, p444, ftp://ftp.te.rl.ac.uk/cms/fed/fed_pmc/docs/talks/leb2000_paper_final.pdf .

[4] CMS Tracker TDR, CERN/LHCC 98-6.

[5] http://cms-tk-opto.web.cern.ch/cms-tk-opto/prr/prr_docs/gh_readout_summary.pdf .

[6] http://hsi.web.cern.ch/HSI/s-link/ and http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/docs/slink64.pdf .

[7] http://web.cern.ch/TTC/intro.html .

[8] G. Pasztor, *"Output data format and on-line data compression algorithms for the CMS Tracker FED"*, CMS-IN 2001/29.

[9] I. Tomalin, *"On Calibration, Zero Suppression Algorithms and Data Format for the Silicon Tracker FEDS"*, CMS-IN-2001/025.

[10] K. Gill, N. Marinelli, *"Start-up Synchronization Procedure of the CMS Tracker"*, unpublished, (2001).

[11] F. Vasey, private communication.

[12] P. Collier and J. Jowett, private communication http://sl.web.cern.ch/SL/sli/Cycles.htm .

[13] T. Bauer and F. Vasey, *"A model for the CMS tracker analog optical link"*, CMS Note 2000/056.