The CMS Pixel FED

M.Pernicka[a], D.Kotlinski[b], W.Johns[c], H.Steininger[a], H.Schmid[a]

[a] HEPHY, Institute of High Energy Physics, Austrian Academy of Sciences, Vienna (Austria)
[b] PSI, Paul Scherrer Institute, Villigen (Switzerland)
[c] Vanderbilt University, Nashville (USA)

pernicka@hephy.oeaw.ac.at

Abstract

The innermost detector of the CMS Experiment consists of 66 million silicon pixels. The hit data has to be read out and must be digitized, synchronized, formatted and transferred over the S-Link to the CMS DAQ. The amount of data can only be handled because the readout chip (ROC) delivers zero-suppressed data above an adjustable threshold for every pixel.

The Pixel FED 9U VME module receives an analog optical signal, which is subsequently digitized and processed. The position of the pixel on a module is transmitted with five symbols coded in six pulse height steps each. The data of 36 inputs build a final event data block. The data block from each detector module with either 16 or 24 ROCs differs in length and arrival time. Depending on the data length and trigger rate, there can be a skew of several events between any two inputs. That is possible because the ROC has a multi-event time stamp memory and the readout bandwith is limited.

Finally the information processed by the Pixel FED will be transferred over the S-Link to the CMS DAQ. Each module must be able to process a trigger rate of 100 kHz or, if in trouble, to send an alarm signal. The number of inputs is limited by the maximum data transmission rate of the S-Link (640 MB/s) for the expected high luminosity of LHC.

The data flow on the module is continuously controlled. Errors are written in an error memory, included in the data stream and if critical sent to the general CMS readout control.

I. INTRODUCTION

The innermost part of the CMS Experiment at LHC is the Pixel Detector [1]. It consists of three barrel and two forward layers with approximately 66 million pixel cells altogether. Even at design luminosity of 10^{34} $cm^{-2}s^{-1}$, the occupancy of each 100 x 150 μm^2 pixel will be less than 0.1% in average [2]. Moreover, the pixel system yields unambiguous space points which turn out valuable for track finding. The high granularity implies material budget, power, cooling requirements and cost which limit the overall size of the Pixel Detector. In order to cope with the data transmission, zero suppression is performed on the front-end readout chip.

This paper will describe the data flow of the hit information on a 9U VME module called CMS Pixel FED. This includes the conversion from optical to electrical, the digitization, data processing, event data block building and finally the transfer to the DAQ system.

II. SYSTEM ARCHITECTURE (FIG. 1)

The Pixel Modules are composed of silicon sensors bump-bonded to 16 (24) ROCs [3] in the barrel (forward) sections. One ROC consists of 4160 pixel cells arranged in 26 double-columns and 160 pixels. Each Module is governed by a Token Bit Manager (TBM) ASIC [4] which controls the serial readout of the ROC chain and adds header and trailer to the data stream. The TBM sends the data over kapton and PCB traces to the Analog Optohybrids [5], where levels are amplified and shifted and then converted to a laser drive current with adjustable gain and threshold. The laser diode emits amplitude modulated light into a single mode fiber at 1310 nm. Initially single fibers are grouped into twelve and later 96-fold bundles through patch panels and transmit the signals to the electronics hut over a distance of about 70 m. The fibers terminate in 12-channel optical receivers which are already part of the Pixel FED.

The Pixel FED will be explained in detail later. Finally,

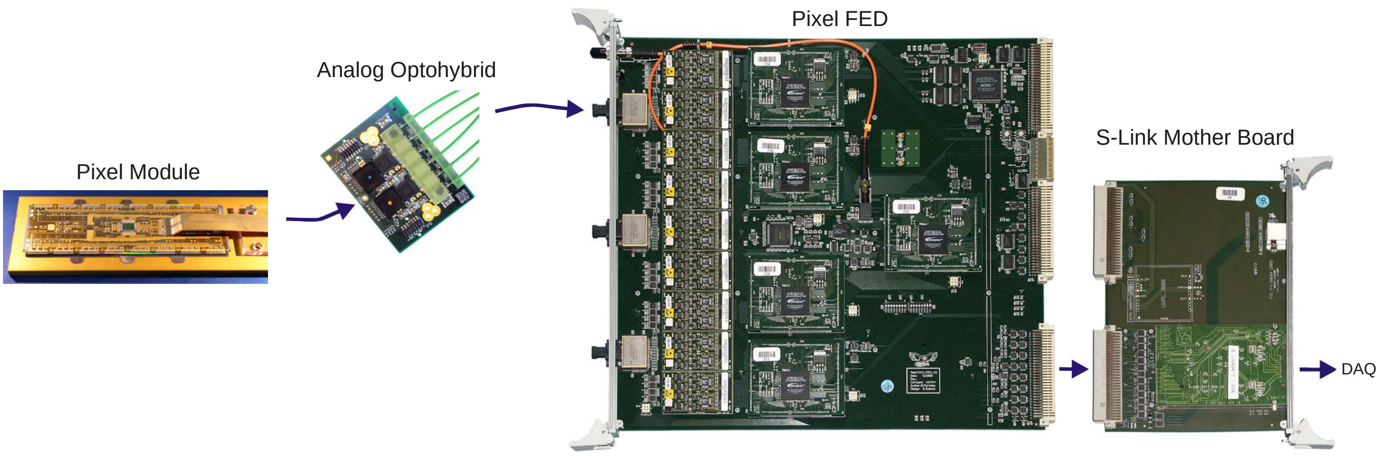

Figure 1: The Pixel Detector Readout Chain

the processed data is sent to a mezzanine board which hosts the S-Link [6] that ships the data to the DAQ system though serialized data links.

III. Pixel Data Format

In order to speed up the transmission of digital pixel hit information while maintaining the global 40 MHz clock, the data are not sent in a common binary fashion, but the available signal amplitude is divided into six possible levels.

The data block (Fig. 2) starts with a header generated by the Token Bit manager, followed by zero-suppressed pixel hit blocks and concluded by a TBM trailer. If there are no hits in a particular ROC, it just sends its ID. The "pedestal" pulse level is called "black", while header, ROC data or trailer start with three, one or two "ultra-black" symbols with the lowest possible amplitude, respectively.

The TBM header carries the event number; inside the ROC data frame, each pixel hit is represented by double column and row numbers coded in the previously mentioned six-level symbols, followed by the analog pixel hit pulse height. The final TBM trailer contains error bits.

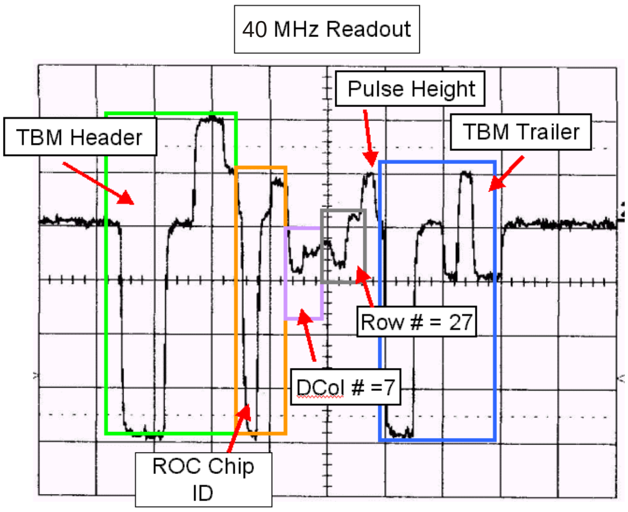

Figure 2: Single ROC hit with TBM header and trailer

IV. Pixel FED Data Flow

This chapter describes the functionality of the Pixel FED, following the main data flow from the optical inputs until the output to the DAQ system (Fig. 3). Finally, controls and monitoring are discussed.

A. *Analog Part*

Each Pixel FED has 36 optical input channels. A pin diode with a customized amplifier [7] converts the light signal into a single line current output. The working point and the frequency behavior of this receiver can be adjusted under VME control. In addition, its output signal offset can be shifted using a slow DAC in order to achieve an operating range suitable for the subsequent ADC. From this point on, the analog signals are fully differential to minimize noise and crosstalk.

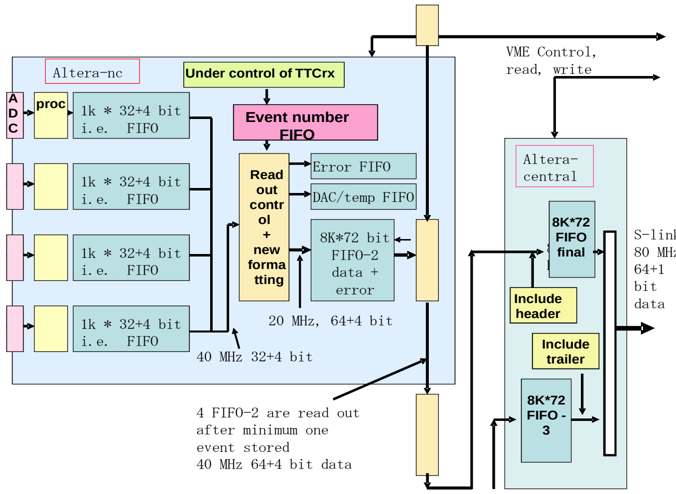

Figure 3: Block diagram of the Pixel FED logic

A fast programmable DAC sequence is added to the signal for test purposes. This can be used to check the module behavior with both normal and corrupted pixel signals. Prior to the A/D conversion, the signals are bandwidth limited by two simple RC filters. The 10-bit ADC itself needs a clock with adjustable phase w.r.t. the global clock such that the digitization point can be optimized for each channel which is essential to maintain a good signal-to-noise ratio. Depending on the selected clock phase, the digitized data is strobed into a buffer using either normal or inverted global clock.

B. *Data Processing*

Each input has its own, independent data processor, which transfers the input number, the pixel hit coordinates and the pulse height into a 32-bit data word.

This normal condition requires some settings to be determined and downloaded in the initial phase of a run.

First, a transparent mode is enabled which allows VME readout of the plain digitized data without any processing. Dummy triggers are sent to the pixel modules to generate arbitrary data which is used to adjust the ADC clock phase for each channel.

The next step is to find the lower and upper limits for each of the six levels used in pixel hit coordinate coding. Each ROC has its own threshold table.

Once those settings are done, the module can be switched to full processing mode, where a state machine looks out for a TBM header. If that is found, its event number is extracted and a subsequent chain of 16 (or 24) ROC chip IDs is expected, each of which can be followed by zero up to many hit coordinate data blocks. If the data flow is corrupted (e.g. missing ROC data or missing TBM trailer), the data block is automatically terminated by a special trailer created in the data processor after a certain timeout. An error bit is included in the trailer.

Furthermore, a mixed mode where both transparent and processed data can be read out by VME, can be used to check whether the on-board processing is correctly done. In that mode, the data contain both the ADC values as well as the processed information, which is slightly reduced compared to normal to fit into the given data structures.

Pedestal fluctuations, e.g. due to temperature variation of the optical transmitter and receiver, are corrected by the data processor. An offset is always added to the ADC reading such that in "idle mode" outside of valid data blocks, the resulting value becomes 512 (ADC mid-range). This offset is updated by -1, 0 or +1 with each clock to satisfy that condition (similar to a delta-encoded ADC) and thus is not prone to spurious noise. During data blocks, the offset is not updated, but still added to the data, generating stable output levels which are not dependent on slow fluctuations anymore (Fig. 4). Of course this method cannot work when components of the transmission chain (such as the optical link) are outside of their linear operating range.

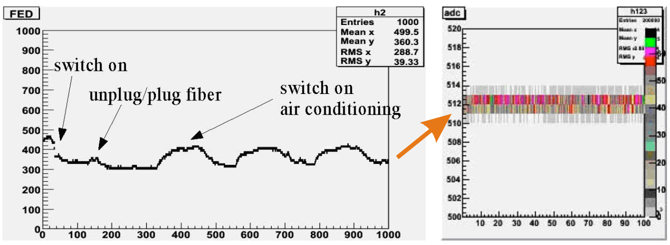

Figure 4: Automatic pedestal correction to ADC mid-range (512)

C. Data Buffering

Each channel has a 1k words 32-bit data buffer (FIFO1) which is filled – according to the incoming data rate (six symbols at 40 MHz for each pixel hit) – at a maximum speed of 6.67 MHz after processing. The length of the data block is limited to a certain value and a trailer is created. There could be a noisy channel which delays the readout and creates an unwanted "busy" or "out of synchronisation" condition. The error message would ask for a reset, a correction of the thresholds or to switch off that input. If the FIFO1 memory gets nearly full (which is usually caused by full FIFOs further down the readout chain), the incoming data frame length is reduced to a minimum and a busy signal is sent to the TTS processor on the board.

Four or five (depending on the location) FIFO1 channels are collected in a FIFO2 with a size of 8k words and a width of 64 data plus 4 control bits. The transfer speed between FIFO1 and FIFO2 is 20 MHz. To start such a transfer, it needs at minimum one stored event number in the event number FIFO. The "not empty" status starts the next readout. In the case where the channel one is the first data, a temporary header with the event number, read out from the event number FIFO, is included. This scheme reduces the content of the event number memory by one. After the event number the readout of the first of four or five FIFO1s is started. No data in a channel creates an error signal and switching off that channel (by VME) is proposed. After a trailer the next channel is read out. After finishing the readout, a "not empty" status in the event number FIFO will start the next readout cycle. In case there are absolutely no data in the FIFO1 some dummy data will be created. This, among other things, was foreseen to exclude potential hang-up of the system.

During the data transfer to FIFO2 the input event number is compared with that of the CMS Timing Trigger Control (TTC) [8] system. A mismatch creates the "out of synchronisation" error. Such a condition can have various reasons, e.g. a bad decoding of the event number or an undetected header.

All errors, including those previously detected by the TBM or the data processor, are stored in the error memory, together with additional information such as the TTC event number and the input number in a 512 words long, 32 bit wide error memory, which can be read out by VME. That information is also packed into the main data stream to the DAQ system, using a special code, and is intended to allow a fast reaction to errors occurring in the pixel system.

D. Final Memory and Data Output

The data of all FIFO2 memories are collected by two Final Memories (FIFO3) of 8k words each over two buses of 64+4 bits at 40 MHz. For each event, a preceding header is written into the first FIFO3 prior to reception of data. One last time, the incoming data is checked. The number of hits per column is histogrammed, allowing detection of potential position decoding errors or hot spots through unbalanced statistics.

The connection to the DAQ system uses the S-Link. Both FIFO3s are subsequently read out at 80 MHz, and a trailer including a CRC32 checksum is generated after the data block.

The output data block is also copied to a spy memory (if enabled) which can be read out by VME.

E. Slow Control and Monitoring

1) VME

As described above, several settings need to be performed during the initialization of a run. Moreover, transparent data has to be read out. All those operations are performed over VME. In addition, spy memories can also be read out during normal data taking to allow permanent data quality checking.

2) Trigger Throttling System (TTS)

The TTS processor collects error messages from several sources on the Pixel FED. Depending on the severity and frequency of each error, some of them will be passed on to the TTS system according to thresholds and criteria to be found heuristically during the start-up phase of CMS.

V. Pixel FED Implementation

A. Board Layout

The 9U VME module (Fig. 5) consists of optical receivers at the front panel, A/D converters for each of the 36 input channels, followed by four Altera Stratix FPGAs, each handling 9 inputs and one central Altera of the same type which collects the data and forwards it to the S-Link sitting on an external board plugged in at the rear. Moreover, a smaller Altera FPGA handles the VME communication and another one is used for clock and controls distribution.

B. Daughter Boards

ADCs and the big Altera FPGAs are located on daughter boards. This approach has several advantages, but also few draw backs. First of all, the complexity of the main board is considerably reduced, resulting in a PCB of ten layers (instead

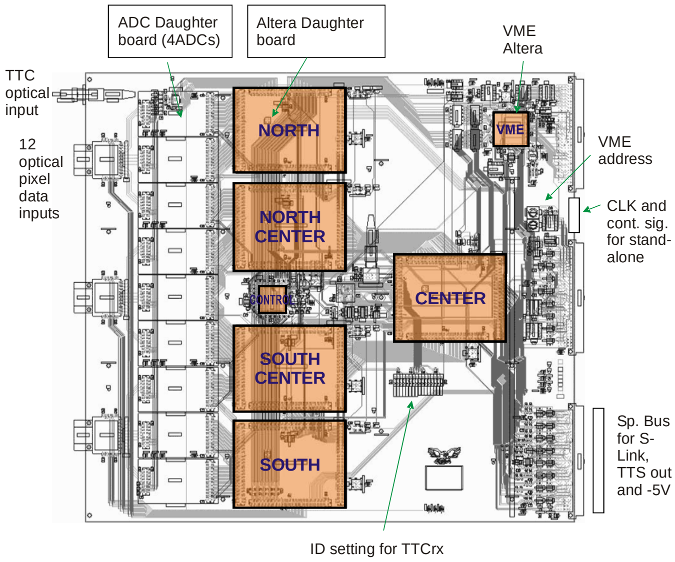

Figure 5: Layout of the 9U VME Pixel FED module

of 14) with an overall thickness of 1.9 mm. The Altera FPGAs daughter boards, however, make use of thinner PCBs in order to accommodate vias of only 200 µm diameter needed to fan out the ball grid array (BGA) connections. The rigidity of that daughter board is given by the four connectors of 100 pins each that surround the board like a frame. Finally, this modular approach eases not only production, but also repairing and spare keeping.

On the other hand, the daughter board approach requires more lateral space as the connectors need to be accommodated. In principle, one could compensate this by placing other parts on the main board just below the daughter board, but as 9U VME offers a lot of space this was only done in very limited amount under the ADC daughter boards where space is tight. Also, cooling needs careful attention with such an approach. Moreover, one could argue that connectors degrade the electrical signal quality. While this might be the case at high speed and frequency, we could not see any such problem at the (moderate) speed of 40 MHz used throughout the Pixel FED. Another issue is the long-term reliability of connectors. In order to avoid potential loss of contact due to ageing and/or mechanical distortion applied to the connectors, we chose a type where the receptacle has springs on three sides which ensures electrical contact regardless of any side forces.

C. Components

1) Optical Receivers

The Pixel FED has three opto-receiver devices, each of which contains an array of twelve pin diodes followed by an amplifier ASIC with a single current output for each channel. During the development of the Pixel FED it turned out that the single lines between the optical receiver and the subsequent buffer, which converts the signals to differential, are susceptible to crosstalk and careful routing is required to avoid potential problems.

An additional, single optical input receives TTC signals such as clock, trigger and reset. The actual pin diode is located in the center of the board, and the optical path is extended to the front panel with a short optical patch cable. Alternatively, the module can be supplied with clock and control signals from a custom P0 connector on the rear ("standalone" mode).

2) ADCs

Each ADC daughter board (Fig. 6) has four channels and contains independent amplifiers and two commercial dual-channel ADCs (Analog Devices AD9218). All analog signals are fully differential, and each channel has its own test input which is added to the main signal.

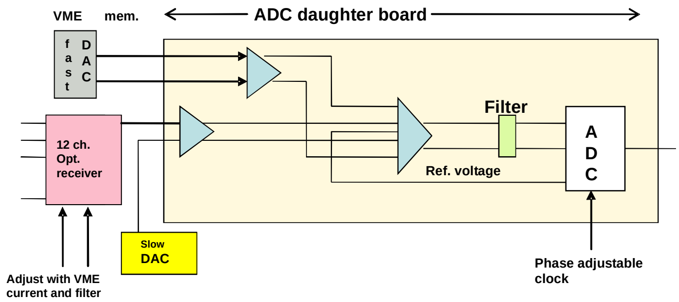

Figure 6: Block diagram of the ADC daughter board

As the incoming signal timing will be skewed between channels, each channel has its own, adjustable clock. In total, 16 clock phases with a spacing of approximately 1.6 ns are generated by the four Alteras in the front and collected by a fast Altera (Max family) located in the center. From there, 36 individual, serially terminated, clock lines deliver a selectable phase to each ADC channel.

3) Altera FPGAs

The choice of FPGA was mainly driven by the available internal memory as well as cost and lead to the EP1S20 device of the Altera Stratix family.

Besides the FPGA itself, the Altera daughter boards also contain an EEPROM that holds the firmware, and a regulator to produce the core voltage for the FPGA.

The VME protocol is handled by an Acex Altera and resides on the main board close to the P1 backplane connector in order to achieve short stubs.

4) Local Bus

Two local bus systems (each 32 bits wide) are used on the Pixel FED for communication between VME and the digital units. All bus signals – except the reset signal – are clock synchronized. The strobe control signals have to pass a digital filter to suppress spurious spikes. The longer one of the two buses covers a total distance of about 50 cm, so termination is essential. However, impedance matching on either end also implies a significant load to the driving FPGA, so a compromise was implemented, consisting of equal 330 Ω resistors towards ground and VCC, on both ends. During normal data taking, the local bus is only used to read out error information. This bus activity contributes between one and two LSBs to the ADC noise compared to the situation with a quiet bus. A slower rise time of the digital signals would help to improve that situation, but we were unable to find any driver with more than 3 ns rise time.

5) S-Link

The S-Link daughter board is mounted on a 6U VME board, which is plugged in on the rear side of the bus, using the P2 and P3 connectors. That solution allows removing the Pixel FED module without caring about the S-Link connection. The transmission of the 64+1 bit wide signals between Pixel FED and S-Link carrier board are made on differential lines, which are generated from single outputs of the Final Altera and are converted back locally to match the single inputs of the S-Link.

The S-Link carrier board is also used as an output interface for the TTS signals.

VI. Performance

The CMS Detector is generally designed for a maximum average trigger rate of 100 kHz, which, together with the expected luminosity, determines the bandwidth requirements of processing elements and interconnections.

In order to stress the Pixel FED, the module was tested with 150 kHz trigger rate and short events. It turned out that the outgoing S-Link is the bottleneck to the overall throughput (in excess of the CMS design values) and the “busy” signal was successfully used to throttle the trigger rate when the FIFOs became nearly full. At full luminosity as expected in CMS, about 100 events of average size can be stored inside the Pixel FED FIFOs.

Extensive tests were performed on the Pixel FED in both development and production phases, and no hang-up condition could be found.

VII. Summary

The CMS Pixel FED has 36 optical inputs and receives pulse height coded pixel information from the front-end. One task is to collect the data from the inputs with different arrival times and data length, and to form an event data block. Three steps of FIFOs are passed by the data stream until it is pushed out by the S-Link to the DAQ system. Several spy memories are included on the module to allow monitoring and debugging. Various types of error conditions can quickly be detected. Every precaution was taken in the design of the Pixel FED logic that no hang-up can occur, no matter how corrupted the input data may be, and extensive tests indeed did not reveal any such problem. The Pixel FED was also successfully tested beyond the maximum CMS trigger rate, where the outgoing data link limited the overall throughput.

References

[1] CMS Collaboration, **CMS Tracker Technical Design Report**, CERN/LHCC 98-6, April 1998

[2] http://cms.web.psi.ch/readout/cms_pix_readout.html

[3] http://cms.web.psi.ch/roc

[4] http://www.physics.rutgers.edu/~bartz/cms/tbm2

[5] http://aoh.hephy.at

[6] http://cern.ch/HSI/s-link

[7] http://cern.ch/cms-tk-opto/tk/components/arx12.html

[8] http://cmsdoc.cern.ch/cms/TRIDAS/ttc