The DAQ and control system for the CMS Phase-1 pixel detector upgrade

Beccherle, R.; Bertacchi, V.; Bianchini, L.; Boccali, T.; Borrello, L.; Bosi, F.; Castaldi, R.; Ciocci, M. A.; Dell'Orso, R.; Fedi, G.; Fiori, F.; Giannini, L.; Giassi, A.; Grippo, M. T.; Ligabue, F.; Magazzu, G.; Manca, E.; Mandorli, G.; Mazzoni, E.; Messineo, A.; Moggi, A.; Morsani, F.; Palla, F.; Palmonari, F.; Raffaelli, F.; Rizzi, A.; Spagnolo, P.; Tenchini, R.; Tonelli, G.; Venturi, A.; Verdini, P. G.; Bellan, R.; Costa, M.; Covarelli, R.; Dellacasa, G.; Demaria, N.; Salvo, A. Di; Mazza, G.; Migliore, E.; Monteil, E.; Pacher, L.; Paterno, A.; Rivetti, A.; Solano, A.; Simelevicius, D.; Rivera, E. Curras; Campderros, J. Duarte; Fernandez, M.; Gomez, G.; Sanchez, F. J. Gonzalez; Echeverria, R. Jaramillo; Moya, D.; Jimenez, E. Silva; Vila, I.; Virto, A. L.; Abbaneo, D.; Ahmed, I.; Akgun, B.; Albert, E.; Auzinger, G.; Bendotti, J.; Berruti, G.; Blanchot, G.; Boyer, F.; Caratelli, A.; Ceresa, D.; Christiansen, J.; Cichy, K.; Daguin, J.; Deelen, N.; Detraz, S.; Deyrail, D.; Dobson, M.; Emriskova, N.; Engegaard, B.; Faccio, F.; Filenius, A.; Frank, N.; French, T.; Fulcher, J.; Gajanec, R.; Gigi, D.; Glege, F.; Hansen, M.; Hegeman, J.; Honma, A.; Hugo, G.; Hulek, W.; Casas, L. M. Jara; Kaplon, J.; Kloukinas, K.; Kornmayer, A.; Koss, N.; Kottelat, L.; Koukola, D.; Kovacs, M.; Rosa, A. La; Lenoir, P.; Loos, R.; Marchioro, A.; Marconi, S.; Meijers, F.; Mersi, S.; Meschi, E.; Michelis, S.; Martin, C. Nieto; Onnela, A.; Orfanelli, S.; Orsini, L.; Pakulski, T.; Perez, A.; Gomez, F. Perez; Pernot, J. -F.; Petagna, P.; Piazza, Q.; Postema, H.; Prousalidi, T.; Rico, R. Puente; Racz, A.; Labaza, A. Remigiusz; Sakulin, H.; Scarfí, S.; Spathopoulos, S.; Sroka, S.; Tropea, P.; Troska, J.; Tsirou, A.; Vasey, F.; Vichoudis, P.; Bertl, W.; Caminada, L.; Deiters, K.; Erdmann, W.; Horisberger, R.; Kaestli, H. -C.; Kotlinski, D.; Langenegger, U.; Meier, B.; Rohe, T.; Streuli, S.; Bachmair, F.; Backhaus, M.; Becker, R.; Berger, P.; di Calafiori, D.; Djambazov, L.; Donega, M.; Grab, C.; Hits, D.; Hoss, J.; Lustermann, W.; Masciovecchio, M.; Meinhard, M.; Perovic, V.; Perozzi, L.; Ristic, B.; Roeser, U.; Ruini, D.; Tavolaro, V.; Wallny, R.; Zhu, D.; Aarrestad, T.; Amsler, C.; Bösiger, K.; Canelli, F.; Chiochia, V.; Cosa, A. De; Burgo, R. Del; Galloni, C.; Kilminster, B.; Leontsinis, S.; Maier, R.; Rauco, G.; Robmann, P.; Takahashi, Y.; Zucchetta, A.; Chen, P. -H.; Hou, W. -S.; R. -S., Lu; Moya, M.; Tsai, J. F.; Burns, D.; Clement, E.; Cussans, D.; Goldstein, J.; Nasr-Storey, S. Seif El; Coughlan, J. A.; Harder, K.; Manolopoulos, K.; Tomalin, I. R.; Bainbridge, R.; Borg, J.; Hall, G.; James, T.; Pesaresi, M.; Summers, S.; Uchida, K.; Cole, J.; Hoad, C.; Hobson, P.; Reid, I. D.; Bartek, R.; Dominguez, A.; Uniyal, R.; Demiragli, Z.; Hazen, E.; Rohlf, J.; Altopp, G.; Burkle, B.; Chen, C.; Coubez, X.; Duh, Y. -T.; Hadley, M.; Heintz, U.; Hinton, N.; Hogan, J.; Korotkov, A.; Lee, J.; Narain, M.; Sagir, S.; Spencer, E.; Syarif, R.; Truong, V.; Usai, E.; Voelker, J.; Chertok, M.; Conway, J.; Funk, G.; Jensen, F.; Lander, R.; Macauda, S.; Pellett, D.; Thomson, J.; Yohay, R.; Zhang, F.; Erhan, S.; Hanson, G.; Si, W.; Gerosa, R.; Holzner, A.; Krutelyov, S.; Sharma, V.; Yagil, A.; Colegrove, O.; Dutta, V.; Gouskos, L.; Incandela, J.; Kyre, S.; Qu, H.; Quinnan, M.; White, D.; Cumalat, J. P.; Ford, W. T.; Macdonald, E.; Perloff, A.; Stenson, K.; Ulmer, K. A.; Wagner, S. R.; Alexander, J.; Cheng, Y.; Chu, J.; Conway, J.; Cranshaw, D.; Datta, A.; Mcdermott, K.; Monroy, J.; Padilla, Y. Bordlemay; Quach, D.; Rinkevicius, A.; Ryd, A.; Skinnari, L.; Soffi, L.; Strohman, C.; Tao, Z.; Thom, J.; Tucker, J.; Wittich, P.; Zientek, M.; Alyari, M.; Bakshi, A.; Bolla, G.; Burkett, K.; Butler, J. N.; Canepa, A.; Cheung, H. W. K.; Chramowicz, J.; Derylo, G.; Ghosh, A.; Gingu, C.; Gonzalez, H.; Grünendahl, S.; Hasegawa, S.; Hoff, J.; Johnson, M.; Lei, C. M.; Lipton, R.; Liu, M.; Los, S.; Matulik, M.; Merkel, P.; Mommsen, R.; Nahn, S.; Prosser, A.; Ravera, F.; Rivera, R.; Schneider, B.; Spalding, W. J.; Spiegel, L.; Timpone, S.; Uplegger, L.; Voirin, E.; Weber, H. A.; Zejdl, P.; Berry, D. R.; Chen, X.; Dittmer, S.; Evdokimov, A.; Evdokimov, O.; Gerber, C. E.; Hofman, D. J.; Mills, C.; Alhusseini, M.; Durgut, S.; Nachtman, J.; Onel, Y.; Rude, C.; Snyder, C.; Yi, K.; Eminizer, N.; Gritsan, A.; Maksimovic, P.; Roskes, J.; Swartz, M.; Xiao, M.; Baringer, P.; Bean, A.; Khalil, S.; Kropivnitskaya, A.; Majumder, D.; Schmitz, E.; Wilson, G.; Ivanov, A.; Mendis, R.; Mitchell, T.; Modak, A.; Skhirladze, N.; Taylor, R.; Acosta, J. G.; Cremaldi, L. M.; Oliveros, S.; Perera, L.; Summers, D.; Bloom, K.; Claes, D. R.; Fangmeier, C.; Golf, F.; Kravchenko, I.; Siado, J.; Iashvili, I.; Kharchilava, A.; Mclean, C.; Nguyen, D.; Parker, A.; Pekkanen, J.; Rappoccio, S.; Hahn, K.; Liu, Y.; Sung, K.; Alimena, J.; Cardwell, B.; Francis, B.; Hill, C. S.; Malik, S.; Norberg, S.; Vargas, J. E. Ramirez; Das, S.; Jones, M.; Jung, A.; Khatiwada, A.; Negro, G.; Thieman, J.; Cheng, T.; Dolen, J.; Parashar, N.; Ecklund, K. M.; Freed, S.; Kilpatrick, M.; Nussbaum, T.; Demina, R.; Dulemba, J.; Hindrichs, O.; Bartz, E.; Gandrakotra, A.; Gershtein, Y.; Halkiadakis, E.; Hart, A.; Kyriacou, S.; Lath, A.; Nash, K.; Osherson, M.; Schnetzer, S.; Stone, R.; Eusebi, R.; D'Angelo, P.; Johns, W.; Padeken, K. O.; Karimeh, W.. - In: JOURNAL OF INSTRUMENTATION. - ISSN 1748-0221. - 14:10(2019), pp. P10017-P10017. [10.1088/1748-0221/14/10/P10017]

02 May 2024

02 May 2024

Available on CMS information server

CMS NOTE -YYYY/N

The Compact Muon Solenoid Experiment

CMS Note

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

01 April 2019

The DAQ and Control System for the CMS Phase-1 Pixel Detector

The Tracker Group of the CMS Collaboration

Abstract

In 2017 a new pixel detector was installed in the CMS detector. This so-called Phase-1 pixel detector features four barrel layers in the central region and three disks per side in the forward regions. The upgraded CMS Phase-1 pixel detector requires an upgraded data acquisition (DAQ) system to accept the new data format with larger event sizes. A new DAQ and control system has been developed based on a combination of custom and commercial microTCA parts. Custom mezzanines on standard carrier cards provide a front-end driver for readout, and a front-end controller for configuration and the distribution of clock and trigger signals. Before the installation of the detector the DAQ system has undergone a series of integration tests, including readout of the pilot pixel detector, which was constructed with prototype Phase-1 electronics and installed in CMS in 2014, checkout of the CMS Phase-1 detector during its assembly, and testing with the CMS Central DAQ. This paper describes the Phase-1 pixel DAQ and control system, as well as integration tests and results, first operational experience, and performance.

Prepared for submission to JINST

The DAQ and Control System for the CMS Phase-1 Pixel Detector

The Tracker Group of the CMS Collaboration

Abstract: In 2017 a new pixel detector was installed in the CMS detector. This so-called Phase-1 pixel detector features four barrel layers in the central region and three disks per side in the forward regions. The upgraded CMS Phase-1 pixel detector requires an upgraded data acquisition (DAQ) system to accept the new data format with larger event sizes. A new DAQ and control system has been developed based on a combination of custom and commercial microTCA parts. Custom mezzanines on standard carrier cards provide a front-end driver for readout, and a front-end controller for configuration and the distribution of clock and trigger signals. Before the installation of the detector the DAQ system has undergone a series of integration tests, including readout of the pilot pixel detector, which was constructed with prototype Phase-1 electronics and installed in CMS in 2014, checkout of the CMS Phase-1 detector during its assembly, and testing with the CMS Central DAQ. This paper describes the Phase-1 pixel DAQ and control system, as well as integration tests and results, first operational experience, and performance.

Keywords: Detector control systems; Data acquisition; Optical detector readout; Modular electronics

Contents

1 Introduction 1

2 System Overview 2

3 Front-End ASICs 4
- 3.1 Readout Chip 4
- 3.2 Token-Bit Manager Chip 5

4 Back-End Implementation 6
- 4.1 Optical Links 7
  - 4.1.1 Pixel Opto Hybrid (POH) 8
  - 4.1.2 Digital Receiver 8
  - 4.1.3 Control links 9
- 4.2 Phase-1 Tracker FEC 10
- 4.3 Phase-1 Pixel FEC 11
- 4.4 Phase-1 Pixel FED 14
  - 4.4.1 DECODE Pixel FED Firmware 14
  - 4.4.2 BUILD Pixel FED Firmware 15
  - 4.4.3 Pixel FED Data Payload 16

5 System Tests in the CMS Detector - Phase-1 Pixel Pilot System 18

6 System Tests in the Laboratory 19
- 6.1 FED Tester Setup 20
- 6.2 The DAQ Setup for High Data Rates 21

7 Pixel Online Software 21
- 7.1 Hardware Access and Supervisors 21
- 7.2 Distributed Software Architecture 23
- 7.3 Interface to the Detector Control System 24

8 Operation Performance 24
- 8.1 Software Recovery Mechanisms 24
- 8.2 Periodic ROC Resets 25

9 Conclusion 27

A ROC 28

B TBM 29

C Rack Layout 32

D POH 33

E Tracker FEC 34
E.1 Functionality 34
E.2 Redundancy 34
E.3 The register write/read commands 35
E.4 Data and end of frame formats 35
E.5 CTRL_RING firmware architecture 36

F Pixel FED 38
F.1 DECODE FED Firmware 38
F.2 BUILD FED Firmware 38
F.2.1 TTC Firmware Block 38
F.2.2 READOUT Firmware Block 39

G FED Tester 43

H Operation Performance 44

1 Introduction

The CMS pixel detector is a key element for the reconstruction of charged particle tracks and interaction vertices at CMS. A detailed description of the CMS detector can be found in [1].

The original CMS pixel detector [2] featured three barrel pixel layers and two forward disks on each side; it was operated during LHC Run 1 (2010-2012) and the first part of Run 2 (2015-2016), and was designed to record efficiently and with high precision the first three space-points in a particle track near the interaction region up to an instantaneous luminosity of 1.0×10^{34} cm^{-2} s^{-1}, with colliding bunch crossings (BX) at a spacing of 25 ns. The original pixel detector would not have sustained the luminosity conditions expected in LHC running after 2017 due to data losses in the front-end readout chip (ROC), and because the maximum throughput rate for the data links of the innermost layer would have been exceeded.

The goal of the Phase-1 pixel project [3] was to perform an evolutionary upgrade with minimal disruption of data-taking by keeping the pixel size, sensor, and readout architecture the same, while improving the performance through a higher rate capability of the ROCs, and larger data transmission rate, more robust tracking through the addition of a fourth barrel layer, and a third disk per endcap, as well as a reduced material budget. The Phase-1 pixel detector was designed to maintain a high tracking performance at luminosities up to 2.5×10^{34} cm^{-2} s^{-1}, corresponding to an average of 80 inelastic interactions per 25 ns BX, (these interactions are referred to as ‘pileup’). The Phase-1 pixel detector with modified data acquisition (DAQ) and control system was installed during an extended year-end technical stop at the beginning of 2017, and is expected to deliver high quality data in the high luminosity environment of the LHC up to Long Shutdown (LS) 3, which is scheduled to start in 2024. The Phase-1 pixel DAQ and control system has been developed based on a combination of custom and commercial microTCA parts. Custom mezzanines on CMS-developed carrier cards provide a Front-End Driver (FED) for readout, as well as a Front-End Controller (FEC) for configuration and the distribution of clock, fast commands and trigger signals.

This paper describes the Phase-1 pixel DAQ and control system. Section 2 gives a system overview, Section 3 describes the front-end ASICs, and Section 4 explains the back-end implementation. Sections 5 and 6 describe the Phase-1 pixel pilot system and laboratory tests, respectively. Section 7 explains the software used for the pixel detector operation. Section 8 provides an overview of the operation performance.

2 System Overview

The CMS Phase-1 pixel detector has three disks on both ends of the forward (FPIX) regions and four barrel layers (BPIX) in the central region. An overview of the Phase-1 pixel DAQ and control system architecture including auxiliary components required to interface with the central CMS services is shown in Fig. 1. The CMS Phase-1 pixel detector has only one type of sensor, bump bonded to 16 ROCs [4]. The active area of the module is 16.2×64.8 mm^2. The pixel size remained the same as in the original detector, 100×150 µm^2. The same n$^+$-in-n technology as for the original detector was used for the silicon sensors. A high density interconnect (HDI) is glued on top of the sensor. The HDI provides signal and power distribution for the ROCs, and it carries the token-bit manager chip (TBM) and decoupling capacitors. The TBM chips are glued onto and wire-bonded to the HDI. They orchestrate the transmission of the data from the ROCs to the back-end electronics. The Phase-1 pixel detector features a fully digital readout system including new back-end electronics. The new ROCs with digital readout operate on a 40 MHz clock and have a 160 Mb/s serial output data stream. This stream is encoded and multiplexed by the TBM using a 4b/5b encoding scheme, to reduce the impact of bit-errors during transmission [5] and for DC balancing. The TBM outputs a 400 Mb/s data stream. A dedicated ROC was designed for the innermost sensor modules in BPIX (layer 1) to cope with the higher hit rates. Layer 1 sensor modules require two TBMs to manage the higher data rates, while all other modules have one TBM.

The sensor modules are connected to the auxiliary electronics (port cards), located in the service cylinders, via flex (FPIX) or twisted pair (BPIX) cables. There are two different types of optical hybrids on the port cards: the Pixel-Opto-Hybrid (POH) and the Digital-Opto-Hybrid (DOH).

The POH converts an electrical signal from the sensor modules to an optical signal and delivers it to the FED. The FED handles decoding and deserialization, and builds event fragments, which are sent to the CMS Central DAQ by a small form-factor pluggable (SFP+) 10 Gb/s S-Link Express transceiver (Tx). There are 24 input channels per FED card; two receivers (Rx) with twelve channels each receive the data from the sensor modules. The FED receives clock and trigger signals from the CMS Trigger Control and Distribution System (TCDS) [6] via a CMS-custom module called AMC13 [7] and the microTCA backplane. The clock runs at the LHC frequency, 40.079 MHz [8]. The FED also provides a trigger-throttle system (TTS) signal to the AMC13. The AMC13 forwards the TTS signals from all the FEDs in a crate to TCDS. The TTS signal indicates whether FEDs are ready to accept triggers or not, and if the event synchronization is kept. The overall TTS state

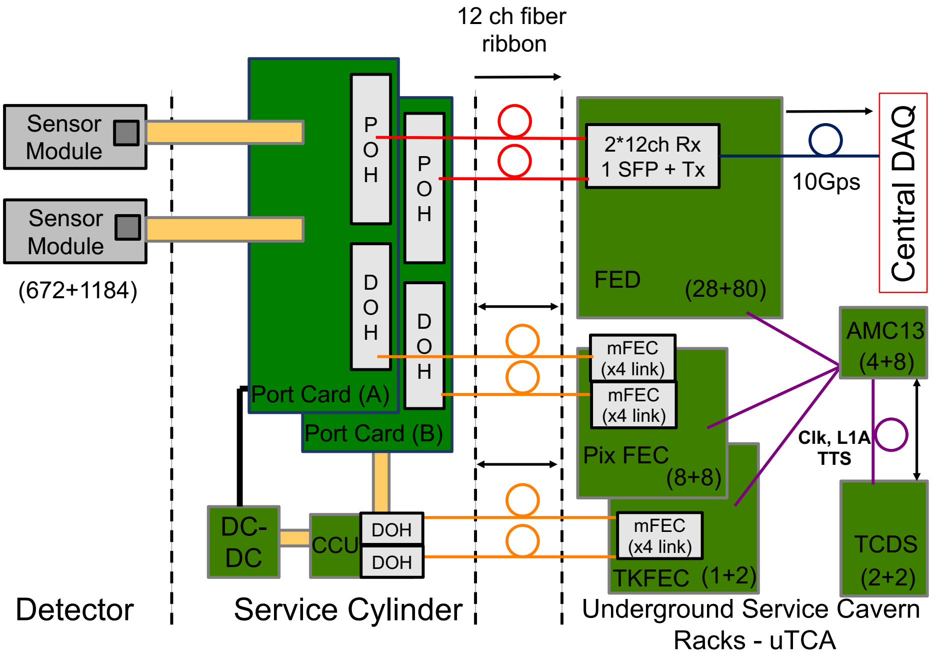


Figure 1: Overview of the microTCA DAQ and control system of the Phase-1 pixel detector. The numbers in parentheses indicate the numbers of the respective installed devices (FPIX and BPIX). Details are explained in the text.

depends on the status of each FED. At a given moment a FED should either accept or block CMS level-1 triggers (L1A) [9]. The pixel DAQ is able to maintain event synchronization across all FEDs with this back-pressure system. The FED sends the data to the Central DAQ Front-End Readout Optical Link-40 [10] (FEROL40) card, the first stage of the CMS Central DAQ chain. A total of 108 microTCA Pixel FEDs are required to read out the Phase-1 pixel detector.

The AMC13 also receives fast commands from TCDS, which include the timing, trigger and control (TTC) [11] information. The AMC13 propagates the received signals to the FEDs and Pixel FECs, the latter distributing them to the sensor modules via the port cards. On the port cards these signals are decoded, after the opto-electrical conversion in the DOHs, by Tracker Phase Locked Loop (TPLL) [12] and Quartz Phase Locked Loop (QPLL) [13]-chips. The signals are then forwarded to the sensor modules on dedicated lines passing through Delay25 chips [14], which provide functionality to delay trigger signals and sent and received clock and data signals. Each sensor module connected to a pixel-control link is identified by a unique, hardwired 5-bit hub address. The Pixel FEC is also responsible for programming the TBM and the digital-to-analog-converter (DAC) registers of the ROCs. A total of 16 microTCA Pixel FECs are required to operate the Phase-1 pixel detector.

Registers on the port cards, including Delay25 chips, and DC-DC converters [15], used for powering, are programmed by the Tracker FEC via the Inter-Integrated Circuit I^{2}C interface and Parallel Interface Adapter (PIA) port, respectively, of a Control & Communication Unit (CCU) [16]. Port cards, DC-DC converters and CCUs are located in service cylinders, which distribute power, cooling and optical links to the sensor modules. The optical links act as an interface between sensor modules and the back-end electronics, located in the underground service cavern. A total of 3 microTCA Tracker FECs are required to control the Phase-1 pixel detector auxiliary electronics.

The number of optical readout links increased with respect to the original detector from 448 to 672 for FPIX and from 1152 to 1696 for BPIX, yielding a total of 2368 readout links. The first and second layer of BPIX use four and two links per sensor module, respectively, to cope with the higher occupancy and data rate. The third and fourth layer of BPIX, as well as the FPIX disks, use one link per sensor module.

3 Front-End ASICs

3.1 Readout Chip

The ROC used in the original CMS pixel detector, PSI46 [17], was designed for hit rates of a few tens of MHz/cm^2, encountered at BPIX layer 1 for an LHC instantaneous luminosity of 1.0×10^{34} cm^{-2} s^{-1} with a 25 ns bunch spacing. This readout chip performed well during the data taking periods from 2008 to 2016. However, it showed inefficiencies when operated at higher data rates when the LHC started operating at instantaneous luminosities above the design value. Therefore, a new pixel ROC had to be designed.

The new readout chip evolved from the PSI46 ROC, keeping most of its characteristics: pulse-height readout, and 52×80 pixels organized in 26 double-columns of 2×80 pixels with common data transfer to latency buffers in the periphery outside the active pixel region. The digital Phase-1 pixel ROC (PSI46dig) is manufactured in the same 0.25 µm CMOS technology as the PSI46, and the overall layout and many building blocks remained unchanged. The two main improvements needed for the upgrade were larger data buffers and higher readout speed.

The double-column buffer sizes have been increased from 32 to 80 cells for the hits and from 12 to 24 cells for the time-stamps. Contrary to the analog PSI46 ROC, the PSI46dig ROC outputs digital data. Hence an analog-to-digital converter (ADC) has been implemented in the chip. It is an 8-bit successive approximation register ADC running at 80 MHz. Digitized data are stored in a 64×23 bit FIFO (First In First Out), which is read out serially at 160 Mb/s. The 80 and 160 MHz clocks needed for the ROC operation are generated from the external LHC clock using a PLL circuit.

During the trigger latency of the CMS experiment, currently 4.15 µs, the pixel hit data must be stored inside the ROC, and only data corresponding to triggered events are read out through the serial optical links. The internal transfer and buffer-capacities of the ROC were designed to cope with the rates encountered at luminosities up to 2.5×10^{34} cm^{-2} s^{-1}.

In addition to the higher rate capacity of the ROC, several other improvements have been implemented. An additional metal layer for power distribution was added, which allows a better decoupling of the power lines from the signal lines, resulting in an improved pixel response uniformity. An optimized comparator reduces the time-walk from about 35 ns [4] to 15 ns [17], resulting in a reduction of the difference between the in-time threshold (within a time window of one clock cycle) and the time-walk independent absolute threshold from about 800 to 150 electrons. This leads to lower noise and cross-talk, resulting in a lower pixel charge threshold.

The above improvements reduce the effective operational threshold of the ROC from 3400 electrons in the original detector to 1700 electrons for the upgraded one. This is important when the amount of charge per hit starts to decrease after radiation damage to the sensors: a highly irradiated detector will slowly degrade in resolution. With a lower threshold, the charge sharing among neighboring pixels can be exploited for position interpolation up to a higher integrated luminosity.

Based on operational experience with the PSI46 ROC and irradiation tests, further optimizations of the internal biasing were made that extend the range of ionizing dose tolerated by the PSI46dig ROC, reducing the need to re-adjust DAC settings with increasing accumulated dose. The PSI46dig ROC performed well and without significant performance degradation after irradiation to up to 120 Mrad (4×10^{14} cm^{-2}, 24 MeV protons, at the irradiation facility in Karlsruhe), which is the maximal dose expected during LHC operations for the Phase-1 pixel detector. A detailed study on radiation tolerance of the PSI46dig ROC can be found here [18].

Data losses have been measured with high-rate X-ray tubes for pixel hit rates of up to 300 MHz/cm^2, and were found to be in excellent agreement with expectations based on detailed architecture simulations [19]. The PSI46dig ROC has performed well during the 2017-2018 LHC run, as shown in Sec. 8. All targeted improvements, i.e. low noise, lower threshold, and lower inefficiency at high rates, have been confirmed during data-taking.

A comparison of the key characteristics as well as measured and simulated efficiencies for PSI46 and PSI46dig ROCs can be found in Appendix A.

Despite the improved performance of the PSI46dig, its architecture would lead to unacceptable data loss rates for the innermost BPIX layer, where pixel hit rates up to 600 MHz/cm^2 may be encountered. A dedicated chip (PROC600) was designed for layer 1, with a complete re-design of the double-column. The PROC600 features a four times higher hit transfer rate of pixels to the end-of-column buffers, and dead-time-free buffer management. The former is achieved by changing from single pixel to 2 × 2 pixel cluster transfers and the implementation of a simpler, handshake-free protocol. A faster and more power efficient analog bus was developed for the pulse height transfers. The data buffer was modified considerably; PROC600 has a ring buffer with 56 buffer units, each containing a cluster base address plus four analog storage cells for the charge pulse heights. The readout is zero-suppressed in order to remove pixels in the cluster with zero measured signal amplitude. In order to significantly reduce the dead-time during operations the logic has been extended; pixel hits are stored in a ring-buffer, and those hits which are validated by the L1A are read out without stopping the acquisition of new hits into the buffer. This avoids an interruption of the data acquisition process in the double-column or overwriting of data, as is the case in the PSI46dig ROC.

The PROC600 has delivered good quality data in 2017 and 2018. Some shortcomings have been observed, like a higher than expected noise hit rate and the rare loss of data synchronization in double-columns. This can be mitigated by operational procedures, the former by an increase of the in-time charge threshold for layer 1 to 3500 electrons, as compared to 1700 electrons used for other layers and, the latter by issuing periodic ROC resets, as described in Sec. 8.2. These issues have been addressed in a revised design of the PROC600, which will be used in the planned replacement of the innermost BPIX layer in 2020 during LS2.

3.2 Token-Bit Manager Chip

The TBM is a radiation-tolerant integrated circuit that controls the readout of groups of ROCs. The TBM chip is mounted as a bare die, wire bonded to the HDI that is glued on the sensor modules.

The principal functions of the TBM include:

- Distribution of clock, L1As and fast commands to the ROCs.
- Distribution of configuration data from the Pixel FEC to the ROCs.
- Passing a token to the readout chain after each incoming L1A.
- Keeping each arriving L1A on a 32-deep stack while waiting for the token to return if the token has not returned before next L1A(s) arrive(s).
- On each token pass signal, the TBM writes a header and a trailer to the data stream.

The Phase-1 pixel TBM replaces the original TBM [20]. The TBM core outputs serial data at 160 Mb/s. Two output data streams are encoded with a 4b/5b scheme and multiplexed by a block called the DataKeeper into a 400 Mb/s stream transferred optically to the FED. There are three versions of the Phase-1 pixel TBM (Table 1). The TBM08 [21], used in FPIX disks and BPIX layers 3 and 4, combines two groups of ROC data, while the TBM09 and TBM10, used in BPIX layer 2 and layer 1, respectively, combine the output of four groups of ROCs into two 400 Mb/s data streams. The TBM09 and TBM10 differ in their timing settings, which are optimized to match the PSI46dig and PROC600, respectively.

Table 1: Different TBM types and their properties.

	Groups of ROCs	Number of ROCs in each group	Number of 400 Mb/s channels	Detector part
TBM08	2	8	1	FPIX + BPIX L3, L4
TBM09	4	4	2	BPIX L2
TBM10	4	2	2	BPIX L1

The data format for Phase-1 sensor modules is as follows: TBM Header, followed by ROC Headers and event information, followed by TBM Trailer. Event number and stack count are included in the TBM Header, ROC Headers are followed by column and row addresses of the pixels with hits, and the TBM Trailer includes the error information.

Each TBM has a 5-bit hub address and each group of ROCs within a TBM is identified with a port address.

More detailed information on the TBM can be found in Appendix B.

4 Back-End Implementation

The design of the back-end electronics for the Phase-1 pixel detector is based on microTCA modular electronics. A microTCA carrier hub (MCH) card is used as communication interface between the microTCA electronics and the network. The microTCA backplane is used to distribute clock, trigger and fast commands that are received from the TCDS via the AMC13.

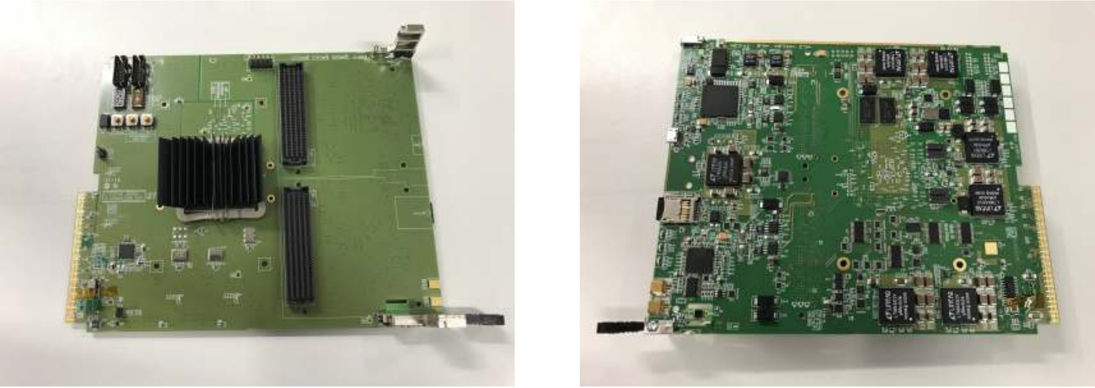


Figure 2: Front (left) and back (right) side of the FC7 with the Xilinx Kintex 7 FPGA and two LPCC FMC connectors.

The FC7 microTCA Field Programmable Gate Array (FPGA) Mezzanine Card (FMC) carrier [22, 23], was selected as the platform for the new digital FED and the Pixel and Tracker FECs.

The FC7, shown in Fig. 2, is a full-size, double-width Advanced Mezzanine Card (AMC) holding a Xilinx Kintex 7 FPGA [24] and offering two low-pin-count compatible (LPCC) FMC slots. There are 20 connections on the front-panel and 12 connections on the backplane available for high-speed (10 Gb/s) serial links to the FPGA. Moreover, there is a block of 4 Gb DDR3 RAM for data buffering that supports a transfer rate of 30 Gb/s.

The application-specific FMCs and firmware make an FC7 into a FED or FEC. Details of the Phase-1 pixel detector rack layout can be found in Appendix C.

4.1 Optical Links

The CMS pixel optical readout link, embedded into the DAQ chain as shown in Fig. 3, starts at the electro-optic POH interface and ends at the opto-electric receiver module interface (DRx12). The data coming from TBMs are sent by the POH at a rate of 400 Mb/s. The control optical link system is based on the same components as used in the original pixel system: a DOH communicating bi-directionally with a FEC, where in this instance the FEC uses standard SFP transceivers.

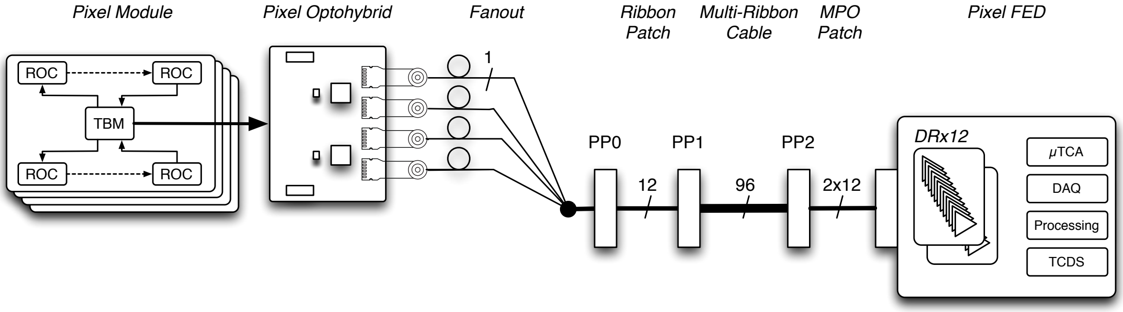


Figure 3: CMS Phase-1 pixel upgrade readout chain.

4.1.1 Pixel Opto Hybrid (POH)

The POH is a printed circuit board (PCB) mounted on the detector service cylinder. Figure 4 shows the POH4 (left) used in BPIX and the POH7 (right) used in FPIX. The optical characteristics of the two variants are the same. The overall system requires 424 POH4 and 96 POH7.

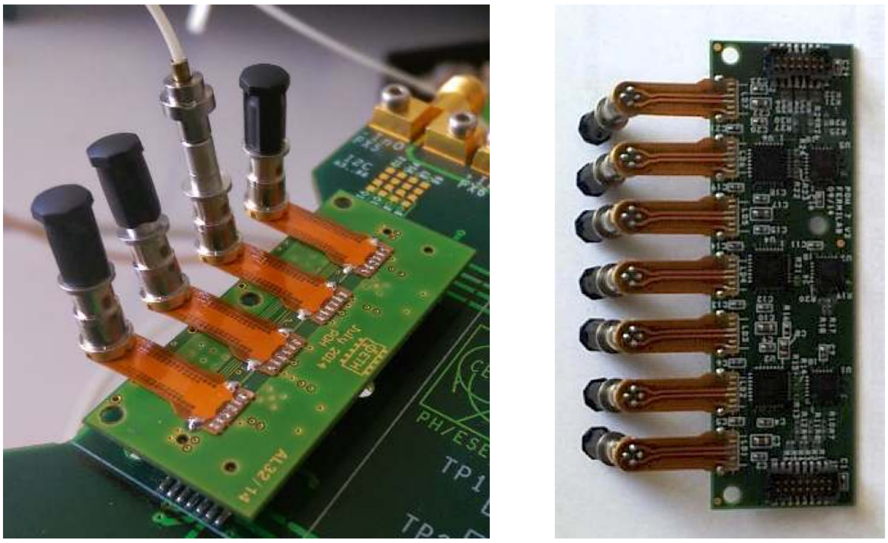


Figure 4: Photographs of a fully assembled POH4 (left) and a POH7 (right). The differences between the two are the number of transmitter channels, four in the case of the POH4 and seven in the case of the POH7, and the input matching that adapts to the signal cables of the BPIX and FPIX system respectively.

The design of the POH uses the Transmitter Optical Sub-Assembly (TOSA) component identified by the Versatile Link project [25]. The POH receives electrical signals from the TBM and converts them into optical signals to be transmitted to the back-end receiver installed in the counting room, about 65 m away from the detector. Each POH houses single mode Fabry-Perot laser TOSAs operating at 1310 nm, Digital Level Translators (DLT) and Linear Laser Drivers (LLD) [26]. The DLT chips convert the signals received from the TBM to levels compatible with the LLD and introduce a gain and an offset to the input signal. The LLD chips drive the laser TOSAs; they pre-bias the lasers at their working point and modulate them with a current proportional to the input signal. The modulation gain and pre-bias currents at the LLD are controlled through an I^2C interface. The POHs are used to transmit balanced digital signals at a maximum bit rate of 400 Mb/s. A typical output optical eye diagram is shown in Fig. 5.

A detailed description of the POH block diagram and the optical fiber plant can be found in Appendix D.

4.1.2 Digital Receiver

The digital receiver module used on the upgraded microTCA FEDs is a purely commercial component. Since the lasers mounted on the POHs emit light at a wavelength of 1310 nm it was critical to identify a receiver module based on an InGaAs photodiode. Typically, such high-density multichannel receivers are based on GaAs photodiodes that operate with light around 850 nm and are

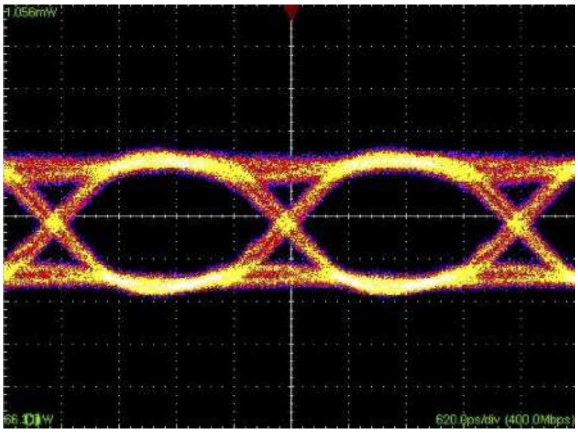


Figure 5: Typical output eye diagram measured on a POH. The horizontal scale is 620 ps/div and the vertical scale is 110 µW/div.

not sensitive to 1310 nm. One manufacturer was identified as being able to produce fully qualified receiver modules [27] with 12-way arrays of InGaAs photodiodes. These were integrated in pairs on an FMC board to be mounted on the FEDs. The receiver modules have a diagnostic feature that allows the DC photocurrent to be measured on each input channel individually. This was used during initial detector checkout to spot problematic fiber connections. Figure 6 (left) shows a picture of a Receiver-FMC (Rx-FMC), also with an SFP+ transceiver attached to it for the Central DAQ line.

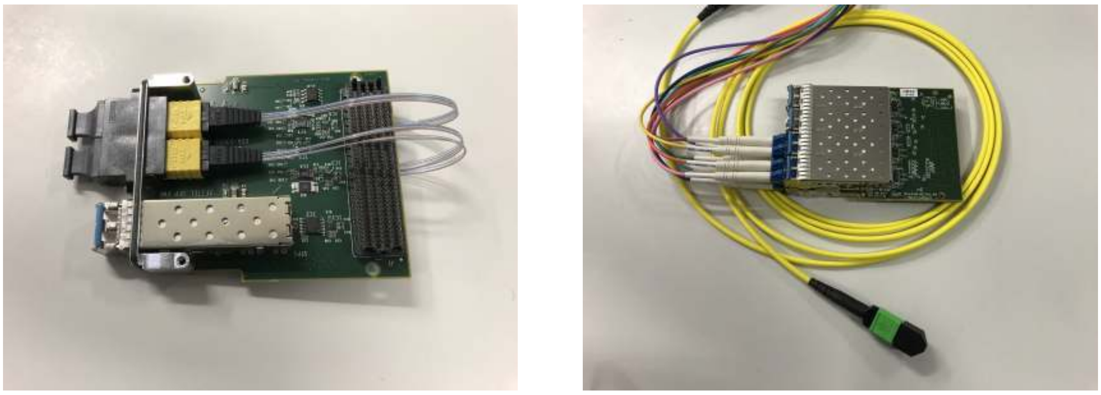


Figure 6: (Left) An Rx-FMC with 24 optical input channels feeding two FITEL 12-channel optical receivers that are optimized for the 1310 nm 400 Mb/s signal, and a SFP+ 10 Gb/s transceiver for S-Link Express to send data to the CMS Central DAQ. (Right) An FMC equipped with low-speed compatible (80 Mb/s) optical transceivers, with 8 SFPs per FMC.

4.1.3 Control links

The optical link system used to control the Phase-1 pixel detector uses the same components as the previous detector system [28] at the front-end. The back-end components that are housed by the

FEC are standard single-mode SFP modules rated for 1-2 Gb/s data rates. These SFPs plug into custom-designed FMC boards, shown in Fig. 6 (right), that are mounted on the FECs.

4.2 Phase-1 Tracker FEC

The Tracker FEC is responsible for programming auxiliary pixel electronics, which is independent from the control of the sensor modules. Each Tracker FEC controls CCU chips in a ring-like topology via semi-redundant connections that carry clock and data signals. The control is done via a token-ring protocol. For the CMS Phase-1 pixel detector there are four control rings for FPIX and BPIX respectively.

The FEC firmware is designed to implement four control ring firmware blocks (CTRL_RING) independent from each other. Each CCU control ring is addressed by one control ring firmware block. The firmware is link compliant with the CCU communication protocol specified for the original detector [16, 29] and access compliant with the control software of the original detector. The firmware is controllable and monitorable over an 1-Gb/s Ethernet/IPBus [30] link and, unlike the other parts of the back-end electronics, the firmware does not need to be synchronized with the LHC clock. Four SFPs and eight optical fibers need to be plugged on the FMC in order to connect a CCU ring. A total of four signals are used per ring: two for data transmission from FMC to the CCU ring, transmitting clock and data, and two for data reception from FMC to the CCU ring, returning clock and data.

The block diagram of the Tracker FEC functionalities is shown in Appendix E.1.

An example of the topology with all the connections is shown in Fig. 7, which considers a CCU ring composed of two DOHs and five CCUs. The last CCU is a spare/dummy, which is needed in order to close the redundant path to output B of the control ring.

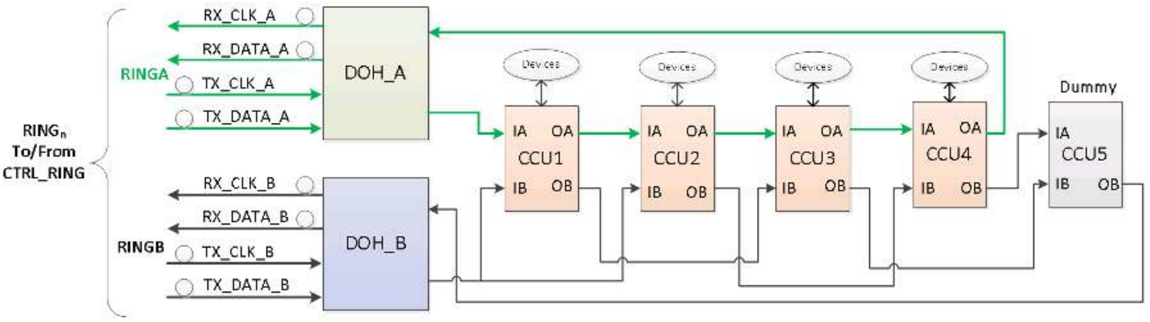


Figure 7: An example topology with two DOHs and five CCUs (the CCU5 is a spare/dummy). Ring A is the primary ring, used by default. In case of a failure, either of DOH_A or any single CCU, the device can be bypassed by switching to Ring B.

The CCU executes the I^{2}C transactions towards the appropriate device from the initial command received. The commands are transmitted from the control ring firmware block (the master) via the TX line (A or B) to the appropriate CCU of the control ring (the slave). The CCUs are distinguishable by their own defined addresses. A ring-type topology is configured as a standard computer LAN network connecting the control ring firmware block to CCUs and the CCUs between themselves. Two types of commands can be executed from the control ring firmware block: register write commands and register read commands.

The redundancy scheme to face potential failures is described in Appendix E.2. The register write and read commands are described in Appendix E.3.

By default, an IDLE pattern is sent to the ring on the TX line by the control ring firmware block. The control ring firmware block also verifies that the ring is well initialized at startup and just before transmitting a command, by injecting a token frame to the ring. The ring is well established if the returned token frame matches the token frame injected. In any case, a status register is updated so that the control software (Sec. 7.1) knows in real time the status of the ring.

The data and end of frame formats are described in Appendix E.4. The details of the control ring firmware block architecture are described in Appendix E.5.

4.3 Phase-1 Pixel FEC

The Pixel FEC is responsible for distributing clock, trigger, and fast commands to the sensor modules, as well as for programming the DAC registers of the ROCs and registers of the TBM chips on the sensor modules.

Xilinx's Vivado [31] development tools were used for the firmware design. A block diagram of the Pixel FEC at the board level is shown in Fig. 8. The firmware was developed using a standard release for the FC7 card, which provides Ethernet services from the AMC backplane. A localized 32-bit wide IPBus allows communication with the FPGA sub-systems by addressable regions from Ethernet. Pixel FEC registers and channel input and output FIFOs are interfaced via Ethernet through the IPBus.

The input clock is sent to a PLL to produce a TTC Clock at the same frequency, as well as 80 MHz and 200 MHz clocks for various other sub-systems. Double data rate TTC information is received through IDELAY and IDDR logic blocks and then processed in a hamming decoder block. The outputs of the TTC decoder block are the L1A and fast commands on an 8-bit bus. Pixel related fast commands are the ROC reset, TBM reset, CAL-SYNC reset, event clear and TTC Send commands. Registers in the Pixel FEC register space count how many Pixel FEC related fast commands are decoded and a FIFO can capture all the TTC events.

A trigger finite state machine (FSM) receives the fast commands and encodes the appropriate bit pattern into the TTC Clock for transmission to the SFP as the module clock. L1A, ROC reset, TBM reset and CAL-SYNC reset signals can also be encoded in the TTC clock by setting appropriate bits in the Pixel FEC register space.

Eight Pixel FEC channels are instantiated in the FC7's Kintex 7 FPGA. Programming data are loaded into a 16 kB transmit FIFO to be used by the transmit FSM. Either a *Send Data* bit is set in the Pixel FEC register space or the TTC Send Data command executes the transmit FSM using the configuration data stored in the transmit FIFO. An 8b/10b encoded data stream is generated and transmitted to the SFP. The TBM's hub and port addresses along with the number of bytes transmitted in the command are stored in the Pixel FEC register space.

During lab tests and during the initial phase of the detector operation, commands to program the sensor modules were composed by fetching configuration data stored on a remote server, and were loaded into the transmit FIFO, to be sent in a sequentially for all sensor modules included in the

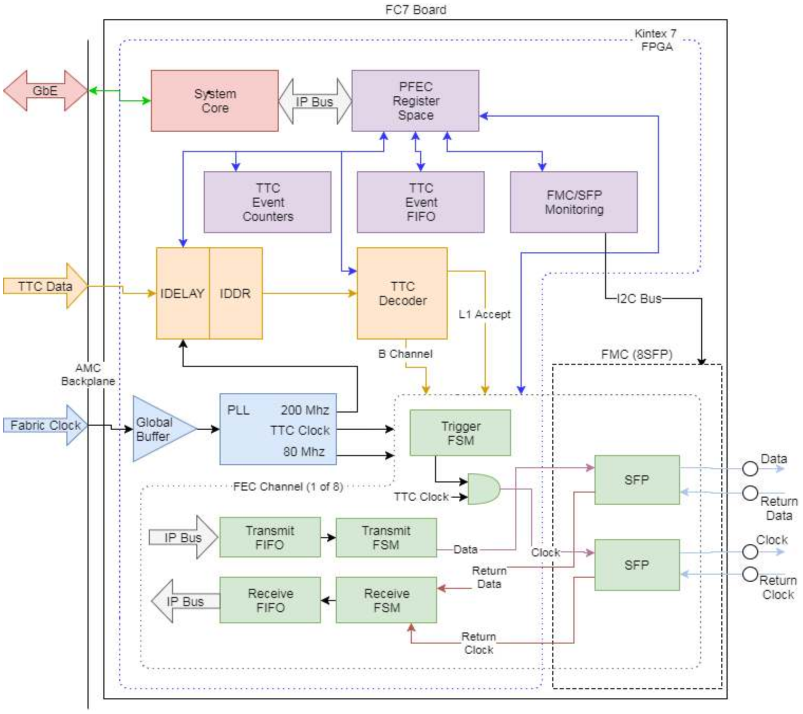


Figure 8: Block diagram of the Pixel FEC.

detector configuration. This procedure was relatively time consuming, and not operation friendly. During data-taking some sensor modules were required to be re-programmed to recover from soft errors such as a Single Event Upset (SEU), a change of state caused by an ionizing particle, in the TBM, a non-responsive TBM, or a non-responsive port card. The soft error recovery mechanism is described in Sec. 8.1.

Since Spring 2018 a new way of programming the sensor modules has been implemented using the feature of storing the configuration data in the FC7 DDR3 SDRAM. This has two benefits. Firstly, it allows to store configuration data locally on the FC7 cards, so during re-configuration there is no need to form the commands again by fetching detector configuration data from the remote server. Secondly, it allows sending configuration commands in parallel, reducing the total Pixel FEC configuration time significantly from 30 seconds to 2 seconds.

The DDR3 memory is partitioned into segments for each of the Pixel FEC channels, out of which one segment is for general calibration purposes, and groups of 4 segments, each used for 28 sensor modules, are used to store TBM settings, two sets of DAC settings for individual ROCs, and settings to trim and mask individual pixels. Each memory segment is assigned a bit used to steered which memory segments are addressed and their commands transmitted during a send command.

The data stream returning from the sensor module is parsed by the receive FSM. The clock for the receive FSM is the returned clock from the sensor module. Data reception begins with a start condition (‘1’s for eight clock cycles).

Data to the same hub/port address can be continuously transmitted, producing no stop condition until the data are exhausted. Once transmission to a hub/port address is complete the transmit FSM waits for the receive state machine to confirm reception of the command before proceeding to the next hub/port command.

Because the exact optical fiber lengths are unknown to the Pixel FEC and can add delays of several hundred nanoseconds between the data leaving the Pixel FEC and the received data at the sensor modules, a simple handshaking between transmit FSM and receive FSM is implemented to prevent meta stability issues. Start of transmission is indicated to the receive FSM so a timeout counter can be started, capping the time the receive FSM waits for the start of a transmission to 100 BXs (2495 ns).

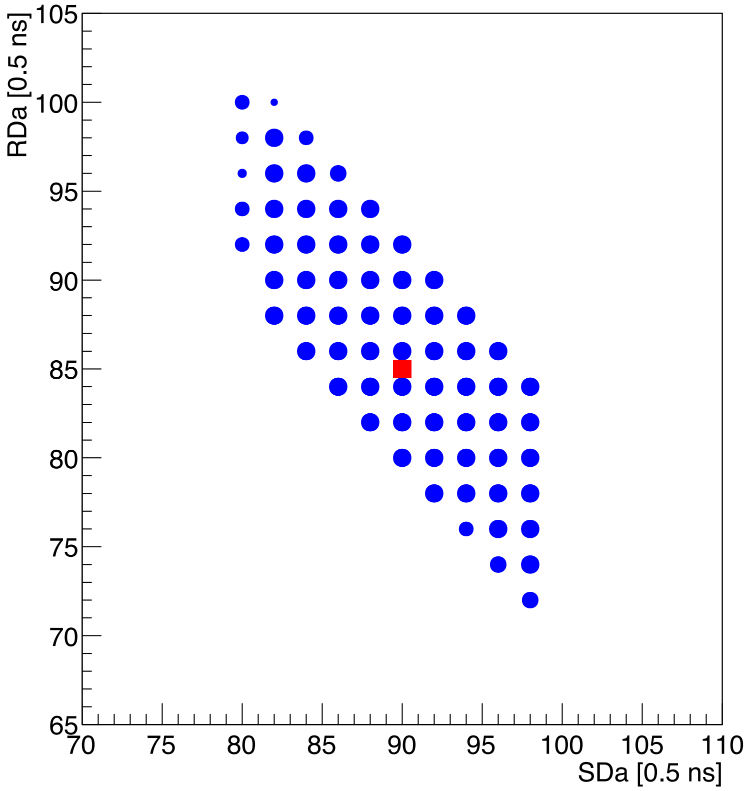


Figure 9: Efficiency of transmission as a function of the delay of returned data (RDa) and sent data (SDa). The size of the blue dots is proportional to the transmission efficiency, the largest dots being 100% efficient. The red dot is chosen as the working point since it is at the center of the area with 100% transmission efficiency.

The transmission path between the Pixel FEC and the sensor module is calibrated by cycling through Delay25 phases of the sent and received data at the port card and plotting the successful transmissions, as shown in Fig. 9. The center of the resulting area is used as the calibrated delay for the sent and returned data. The Pixel FEC has been shown to have ± 7.5 ns of phase margin between sent clock and sent data, and ± 6 ns of phase margin between returned clock and returned data signals.

4.4 Phase-1 Pixel FED

The Phase-1 Pixel FED consists of an FC7 board with a Rx-FMC. The Rx-FMC is a mezzanine containing two 12-channel optical receivers that collect signals from the sensor modules, and one SFP+ for data transmission to the CMS Central DAQ, as shown in Fig. 6 (left). One FED can read out 24 data streams of 400 Mb/s, and transmit output data at 10 Gb/s. The FED can also emulate and transmit data itself to run without a detector.

The FED firmware consists of two parts. The first part (DECODE) handles decoding of the incoming data. The second part (BUILD) builds pixel events and sends them to the CMS Central DAQ.

4.4.1 DECODE Pixel FED Firmware

The DECODE part of the Phase-1 pixel FED firmware (Fig. 10) was designed to automatically find the best sampling point for the incoming 400 Mb/s signal and to do a continuous sampling phase finding without disturbing data integrity. The optical receiver output, which carries the 400 Mb/s data stream, drives a differential input buffer of the FPGA. The negative output of this buffer is used as a copy of the incoming data stream to perform sampling phase finding and phase correction calculations.

The main task of the DECODE part of the Phase-1 pixel FED firmware is to decode the non-return-to-zero-inverted (NRZI) and 4b/5b encoded 400 Mb/s input signals and split the multiplexed TBM channels into two data streams. A TBM data stream starts with a TBM Header, followed by an 8-bit event number. This is followed by ROC Headers which indicate the beginning of pixel data and carry read-back information of different ROC specific voltages and currents. A TBM Trailer followed by 16-bits of status information terminates the data stream.

Sampling phase finding and the reverse functionality of the TBM DataKeeper are described in Appendix F.1.

The DECODE firmware detects TBM Header, TBM Trailer, ROC Headers and pixel data, and adds various spy FIFOs for debugging purposes and symbol error histogramming. Several checks are included to keep data integrity as high as possible. Therefore, not only the TBM Header marker has to be identified to start a data packet, but also the beginning of the next marker is included to validate the start sequence. ROC Headers are only allowed within the expected delay after the arrival of a TBM Header. The number of ROCs is counted and an error reported if the count does not match the expected number from the TBM type. Furthermore, Header and Trailer veto times and sequence controllers are added to avoid corrupted data packets. At this stage the TBM 4-bit words, which are outcome of NRZI decoding is followed by the 5b/4b conversion, are combined with a 4-bit qualifier marker, which allows the following stage to identify these words as Header, Trailer or pixel information.

The DECODE firmware reduces the data volume when the TBM FIFOs get full to a programmable value and speeds up data transfer by terminating the TBM data stream in case of unequal payload for layer 1 modules. The data stream from DECODE firmware block to BUILD firmware block contains error type overflow under such termination. The data streams are forwarded to the BUILD firmware block using a 36-bit wide interface with the possibility of clocking data out at 40, 80 or 160 MHz.

For debugging purposes, the DECODE part of the firmware has fiber specific spy FIFOs which store the incoming 5-bit symbols and the 4-bit data words. To monitor the data transfer to the BUILD firmware part, additional spy FIFOs for every TBM channel are implemented.

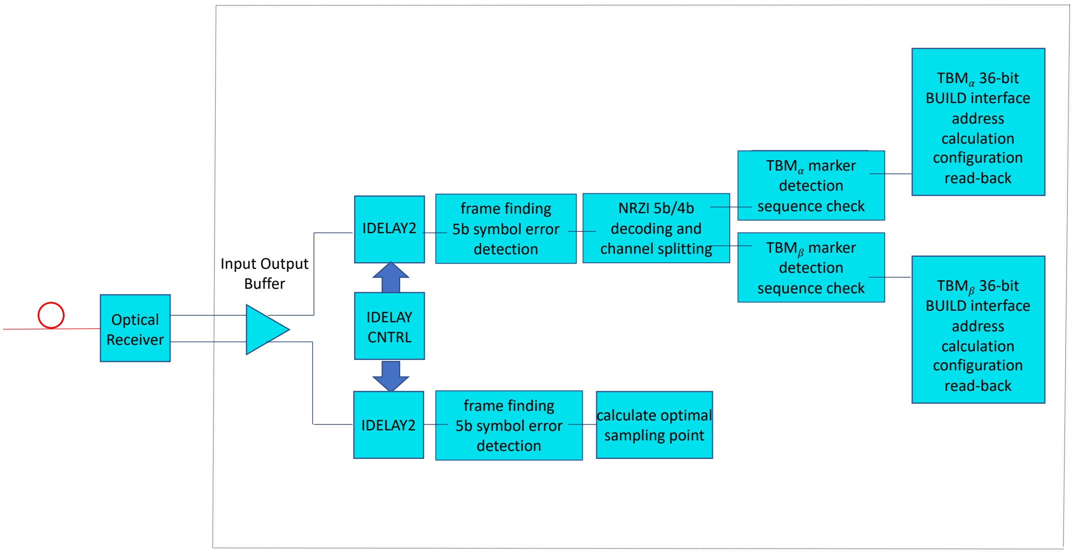

Optical Receiver
Input Output Buffer
IDELAY2
IDELAY CNTRL
IDELAY2
frame finding 5b symbol error detection
NRZI 5b/4b decoding and channel splitting
frame finding 5b symbol error detection
calculate optimal sampling point
TBM$_\alpha$ marker detection sequence check
TBM$_\beta$ marker detection sequence check
TBM$_\alpha$ 36-bit BUILD interface address calculation configuration read-back
TBM$_\beta$ 36-bit BUILD interface address calculation configuration read-back

Figure 10: Block diagram of the Phase-1 pixel FED DECODE firmware.

4.4.2 BUILD Pixel FED Firmware

The BUILD FED firmware was designed to handle the data readout of 48 channels coming from the DECODE FED firmware, transmit data to CMS Central DAQ via the S-Link Express interface, and communicate with the TCDS system for the TTC and TTS interfaces for synchronization. A 1-Gb/s Ethernet/IPBus communication is used to diagnose the exceptions which can occur during physics data taking.

The major constraints are the data rate, the capability to maintain the synchronization, and the exception/error handling. The exceptions can occur due to corrupted data provoked by an SEU or sensor modules not sending coherent data.

The block diagram of the BUILD FED firmware is shown in Fig. 11.

The TTC firmware block, that receives clock and TTC signals from the AMC13 via the backplane, is described in Appendix F.2.1.

The READOUT firmware block computes and transmits its own individual TTS state. This state is a 4-bit word controlled by the FSM, where the transitions depend on conditions triggered by the firmware and on the received TTC commands. The conditions triggered by the firmware are essentially the filling level of the buffers needed for the readout (buffer for L1A and channel buffers storing the pixel data) and the synchronization loss detection.

The TTS state is logically compatible with the original pixel system. The TTS states used in the READOUT firmware block in the BUILD firmware are ready (RDY), busy (BSY) and out-of-sync

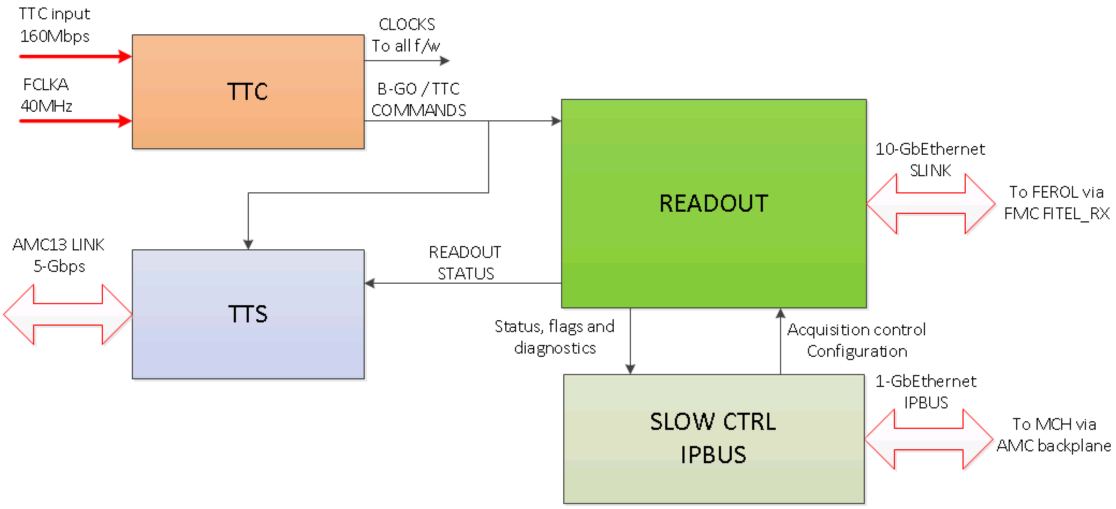


Figure 11: Block diagram of the BUILD FED firmware.

(OOS). The other TTS states are not used. The TTS states and transitions are shown in Fig. 12. After the system is configured, the TTS state is RDY. When buffers are almost full back-pressure kicks in to avoid an overflow and a subsequent loss of the synchronization. The goal of the BSY state is to rapidly veto the arrival of new triggers. The veto is not instant due to non-negligible propagation time, new triggers are still accepted before the back-pressure is effective. An OOS condition due to either consecutive timeouts (a timeout occurs when a FED channel does not receive data for a programmable time) or consecutive event number mismatches can be triggered at any moment in any TTS state. In order to restart running with event number 1 and empty data buffers, a resynchronization command is propagated from TCDS to all FEDs. The TCDS resynchronization command is interpreted as a resynchronization sequence (RESYNC). The firmware was written to accept two types of RESYNC commands: global or private. In the global case, both front-end and back-end electronics receive the same command. In the private case, only the back-end electronics receive the command without the event number reset (EC0).

A detailed description of the READOUT firmware block can be found in Appendix F.2.2.

4.4.3 Pixel FED Data Payload

The sensor modules transfer zero-suppressed data to the DAQ system via 2368 optical fibers. The average number of hits in a sensor module decreases with its radial distance from the interaction point. Figure 13 (left) shows the average number of pixel hits per event for all fibers. The distribution is uniform for the outer layers in BPIX and the outer rings in FPIX, while the average number of received hits per event has a large spread for the innermost layer. In order to balance the data processing load on the FEDs, fibers from different layers and z-coordinates have been bundled into groups of twelve. Most FEDs take two of these fiber bundles as inputs. Figure 13 (right) shows

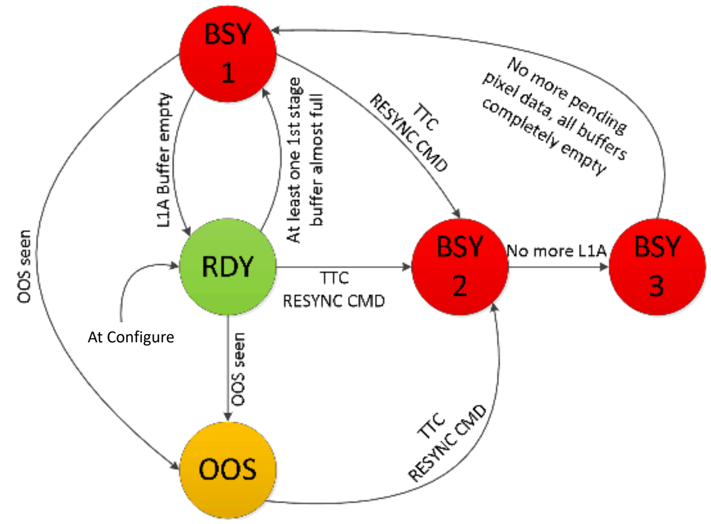


Figure 12: Block diagram of the FED TTS states and transitions. The BSY1, BSY2 and BSY3 are different FSM nodes which all have TTS state BSY.

the distribution of average hits per fiber bundle and per FED. The data load is distributed equally across the fiber bundles and thus also across the FEDs, with the FPIX FEDs (FED number > 96) receiving slightly higher data rates on average than the BPIX FEDs (FED number ≤ 96).

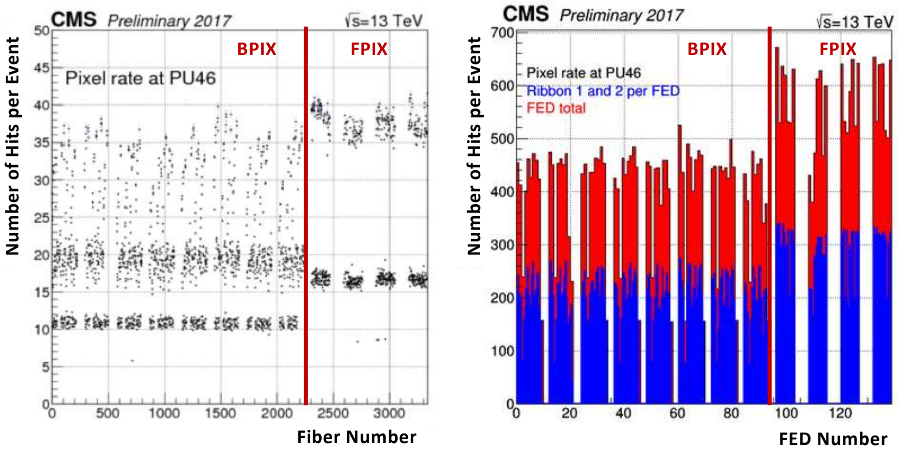


Figure 13: (Left) Data payload versus fiber number and (right) data payload versus FED number for a pileup of 46. For BPIX the data payload is 10 hits per event per fiber, and thus lowest, for fibers connected to layer 4 modules. Layer 2 and 3 modules have a data payload of 15-20 hits per event per fiber. Data payload for layer 1 modules fluctuates between 20 and 40 hits per event per fiber. For FPIX the data payload is 16-18 hits per event per fiber for outer ring modules and 35-40 hits per event per fiber for inner ring modules. Fibers are mapped so that the data payload is distributed as evenly as possible between the FEDs of one sub-detector (BPIX and FPIX).

5 System Tests in the CMS Detector - Phase-1 Pixel Pilot System

In order to be well prepared for a short commissioning period during the extended year-end technical stop at the end of 2016 and to take advantage of the lengthy access to the original detector possible during LS1, a pilot system [32] was built. It consisted of eight prototype Phase-1 sensor modules. The pilot system was installed in 2014 in the available space in the original FPIX half cylinders (Fig. 14), which host the auxiliary electronics. A prototype microTCA FED system was used to read out the pilot system. The motivation for installing the pilot system was to learn how the readout, control, and offline systems perform in the CMS environment and to start integration within the CMS DAQ.

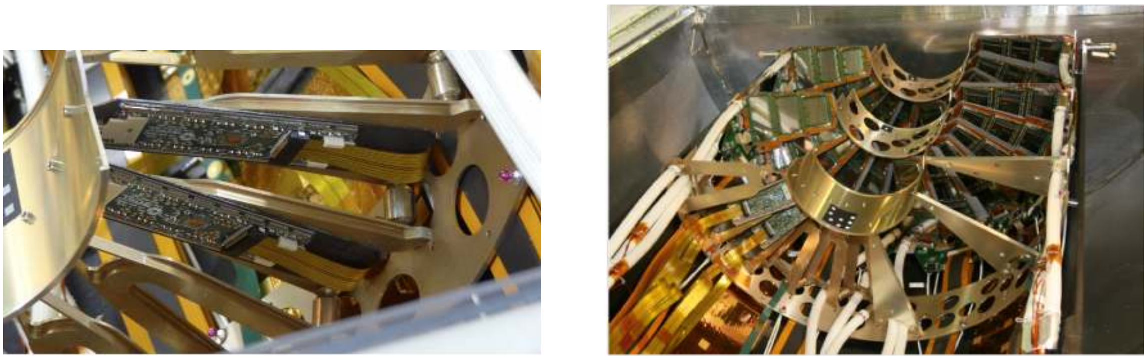


Figure 14: (Left) Two pilot sensor modules. (Right) Pixel half cylinder with pilot sensor modules installed on the third half disk, and pixel plaquettes of the original pixel detector installed on the first two half disks.

The pilot system was commissioned at CERN using a test stand running a standalone test software. Calibration procedures for the pilot detector implemented in the online software were validated after the installation in CMS. During the pilot system tests at CERN it was observed that the prototype FED was having problems in decoding the data at high trigger rates. The problem was traced back to two separate sources: an asymmetric eye diagram due to the TBM design, and jitter on the Phase-1 pixel port card. While an asymmetric eye diagram could be accepted for the pilot system, new versions of the TBM were designed for the final Phase-1 sensor modules that were installed in CMS in the spring of 2017. In order to address the jitter on the port card, an external QPLL chip had to be put in between the TPLL and Delay25 chips on the pilot port card. Figure 15 shows asymmetric eye diagrams for one of the pilot modules before and after QPLL installation. For the final Phase-1 port cards, the design incorporated the QPLL chip directly on the PCB.

Six out of eight pilot sensor modules were successfully used in data taking. Pixels get clusterized by the reconstruction software and the hit position is determined by their barycenter. Figure 16 (left) shows the measured cluster positions projected onto the transverse plane. Figure 16 (right) shows the expected hits which are derived from extrapolated tracks that are reconstructed in the FPIX detector. The pilot system was not part of the tracking since the pilot system modules are located at the edge of the tracking coverage. This effect is visible in Fig.16 (right), where there are no expected hits close to the center. In order to discard the fringes of ROCs, where the uncertainty in the track extrapolation is large, fiducial regions are defined. These are visible as rectangular

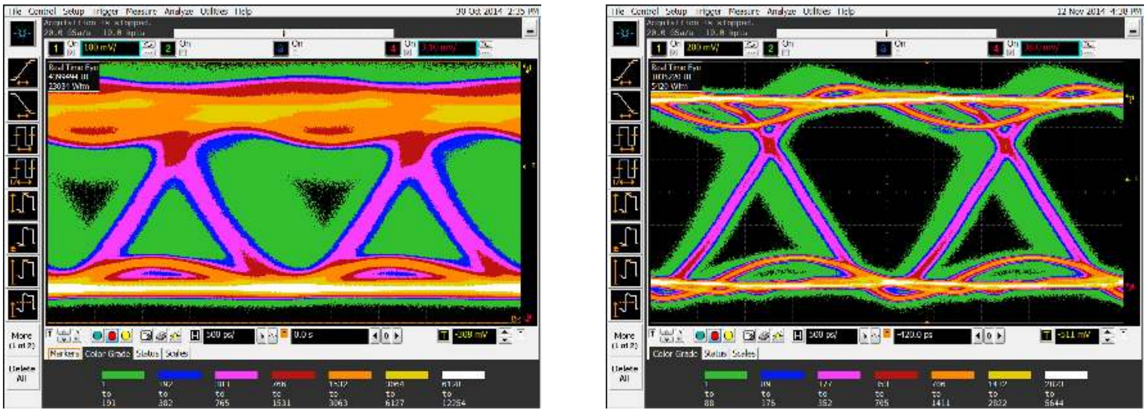


Figure 15: Asymmetric eye diagrams for one of the pilot modules before (left) and after (right) QPLL installation. The amount of jitter decreased significantly after QPLL installation.

shapes in Fig.16 (right).

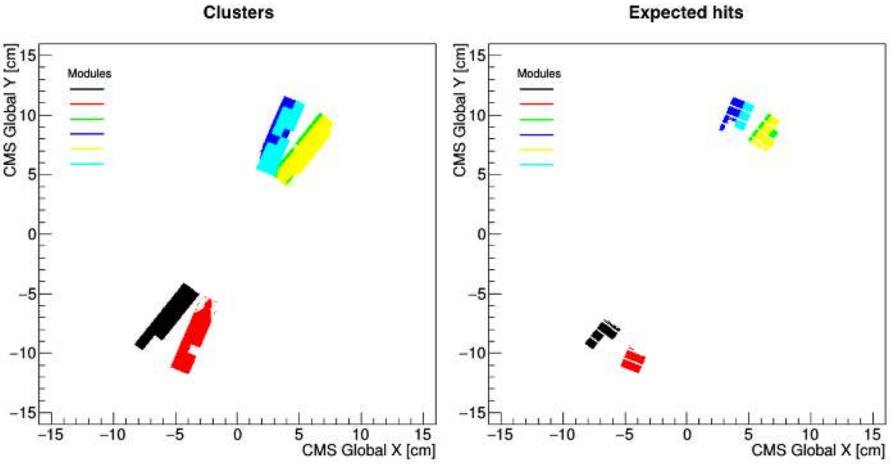


Figure 16: (Left) Cluster positions in each pilot sensor module used in data taking and (right) expected hits in the transverse plane in the CMS coordinate system. Color coding identifies individual sensor modules.

Operating the pixel pilot system during the years 2015-16 within CMS provided valuable experience and enabled an early start for the modifications that were required for the integration of the Phase-1 pixel DAQ.

6 System Tests in the Laboratory

Small scale systems were used for development and testing of final detector parts, which advanced development and uncovered errors and issues well ahead of the final system installation. There were three integration centers using microTCA back-ends: Fermilab, the University of Zurich (UZH), and CERN. In addition there were test stands at HEPHY in Vienna, IPHC in Strasbourg, and

Cornell University for firmware and software development and testing. At Fermilab, final checkout of the FPIX detector was performed [33] before shipping it to CERN for installation in CMS. At UZH [34] the focus was on testing the optical components and electronics on the BPIX service cylinders, and on the integration tests for the BPIX detector ahead of deployment in the production system. At CERN [35], emphasis was on DAQ hardware testing and integration and on testing firmware before deployment in the Phase-1 pixel detector. Functionality tests were also performed on detector components upon arrival at CERN.

A so-called "soak test" facility was set up at CERN to validate all DAQ back-end components before installation in the CMS service cavern. A rack layout identical to the final setup in the cavern containing all of the production parts, power modules, AC-DC converters, crates, service boards, as well as FEDs and FECs, was operated for several weeks before installation. The soak test included regular firmware upload, and power cycling of FEDs and FECs.

6.1 FED Tester Setup

A data emulator, the FED tester, has been designed for the Phase-1 pixel upgrade based on the gigabit link interface boards (GLIBs) [36] combined with the same FMC as used on the FECs. Custom firmware and software was developed to emulate the data bit stream from sensor modules with different TBMs and different ROCs. The software is able to generate data patterns in the FED tester framework and can validate the output from the FED. Constant agreement should be seen between what is sent from the emulator and what is decoded by the FED.

The GLIB firmware is able to independently emulate 16 channels. Three GLIB boards are used to completely fill one FED, and optical splitters can be added in order to feed multiple FEDs in parallel. Each group of two channels is then multiplexed, and NRZI and 4b/5b encoded before being transmitted.

The FED tester emulates the event structure as used by the sensor modules. Once it receives a trigger an entire event is generated and sent: a TBM Header, ROC Headers, pixel hit data, a TBM Trailer, and 16 bits of status information.

The emulated data are sent to the FED, where they are decoded and the resulting output of the FED is compared to what was originally sent. The bitwise signal of the events can be altered to cause errors in the FED. Events that have resets or other possible errors are emulated are still decoded but marked in an error FIFO.

The FED has multiple error counters that can count independently for each channel. These are all read out in the FED tester framework to confirm that the count for each error is accurate. FED tester customization enables consistent tests of the FED firmware from version to version. Event readout can be done with a fixed data size, where every event is the same, or in SRAM mode, where the event size and pixel location can be programmed via software. The FED tester software is independent of the CMS software framework. Multiple test stands can be set up and tests run without the need for clock input from CMS.

The SRAM of the GLIBs is a software loadable memory that can be accessed by the event readout framework. There are two separate memory locations that each hold approximately 8.4 MB of data. The first SRAM is designated to hold the distributions of hits per ROC. The second is designated to hold the emulated pixel locations for each hit. The reading of SRAM memory is driven by a 160 MHz clock. Since each GLIB can emulate 16 independent channels, there are 32 different processes which must occur to emulate an event. The first SRAM only needs to be read once per event, the second SRAM needs to be read any time a pixel hit information needs to be sent. The SRAM readout can be done at trigger rates expected in CMS.

The details of hit distributions that are held in the first SRAM can be found in Appendix G.

6.2 The DAQ Setup for High Data Rates

Prior to installation the DAQ system was qualified, at small scale but with a complete chain of DAQ hardware, for the highest expected data rates from the sensor modules. A microTCA crate with five FEDs was connected to the CMS Central DAQ, and with the FEDs emulated data patterns. A FED tester was also installed in the crate, and six optical splitters were used to feed the FEDs with the FED tester output. The FED tester output and internally emulated FED data were used at high trigger rates to qualify the pixel DAQ system and the interface to CMS Central DAQ. Clock and trigger signals were supplied by the TCDS system, as in the production DAQ system. The DAQ setup was used to develop configurations to interface with the TCDS system, and to study the robustness of the TTS state transitions and the time spent in TTS states BSY and OOS under different conditions. It was also used to optimize the AMC13 [7] configuration for the pixel use-case, and to study the propagation of the TTC commands from TCDS via the AMC13 to the FEDs. It is currently used as a test bench to test new FED firmware releases, before their deployment in the production system.

It is possible to check the data sizes through the S-Link Express link of the FED using the FED tester. When the event sizes are large enough the trigger throttling limits the maximum data throughput. The throughput is tested by using fixed size data and SRAM data, to allow for more realistic conditions. The average pileup during LHC Run 2 is expected to be approximately 60 (less than 2 hits/ROC at 100 kHz), which corresponds to approximately 3 Gb/s at 100 kHz. For up to 6 hits/ROC the FED can run at 100 kHz. Starting from 7 hits/ROC throttling of triggers starts and data throughput reaches approximately 7.5 Gb/s (at a trigger rate of 72 kHz). This shows that the data throughput was not a bottleneck during LHC Run 2 and will not be a bottleneck during LHC Run 3. Figure 17 shows the throughput and trigger rates the FED can handle when different numbers of hits/ROC are generated by the FED tester.

7 Pixel Online Software

The Pixel Online Software (POS) is a collection of applications that control the front-end and back-end hardware of the CMS pixel detector. The software collection is written in C++ and is based on the CMS online software framework XDAQ [37].

7.1 Hardware Access and Supervisors

For each type of back-end electronics board in the pixel system a corresponding controller class exists. The controller provides the software interface to the hardware and allows access to the hardware functionality within POS. It uses the CACTUS framework [38], which provides a hardware abstraction layer (HAL) for microTCA hardware. For the actual communication with the hardware the IPBus protocol is used. This protocol is transported via IP and Ethernet. The hierarchical structure of the POS is shown in Fig. 18.

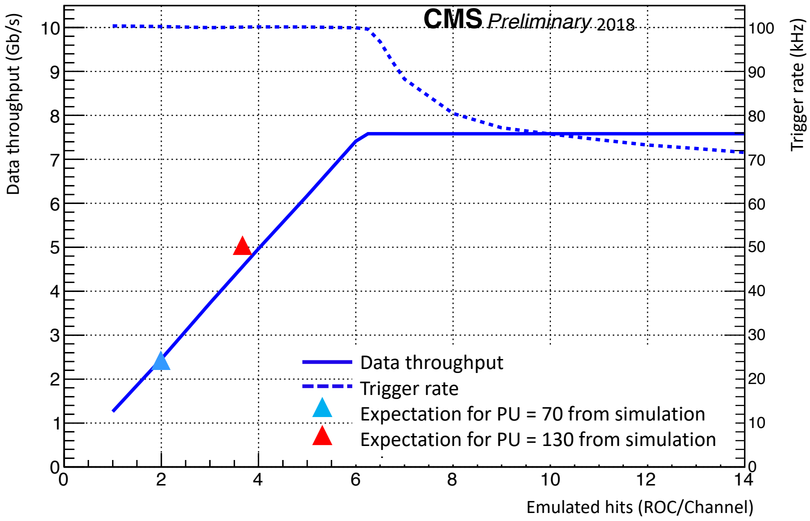


Figure 17: Data throughput (solid line and left y-axis) and trigger rate (dotted line and right y-axis), as measured when the system is driven by the FED tester. The FED can handle a trigger rate of 100 kHz for up to 6 emulated hits per ROC. The throttling of the trigger rate is caused by back-pressure in the FED. The blue (red) triangle is the throughput for simulations with a pileup of 70 (130).

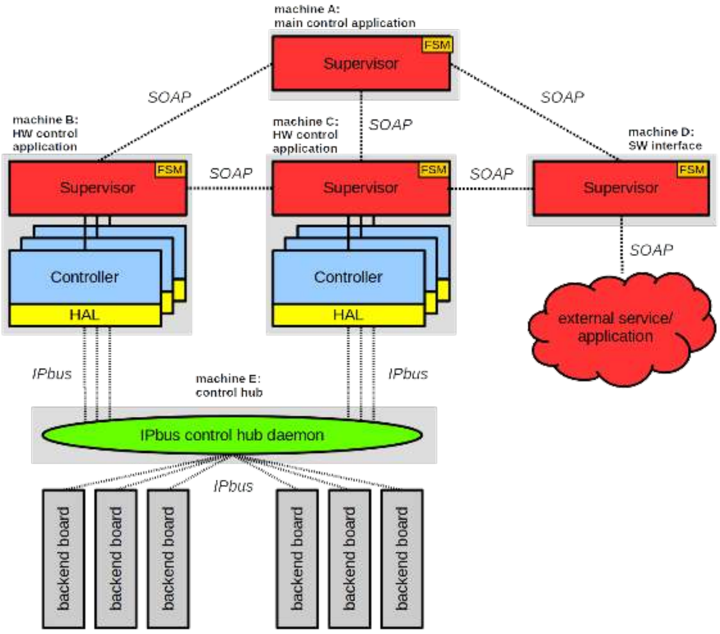


Figure 18: The hierarchical structure of the POS.

In order to prevent conflicts due to potential concurrent hardware accesses, the connection is established via a so-called control-hub, which is a service daemon running on a separate machine that queues incoming requests from different applications and distributes them to the actual hardware.

States play a very important role for the software that is used in the operation of an experiment. The software must always reflect the current hardware state and must be able to perform well defined transitions between these states when instructed from a higher control level. For this reason, an additional application layer is built on top of the controller layer, the so-called hardware supervisors. All supervisors implement a common FSM and define interfaces to the outside for state changes. The cross-communication between the supervisors in POS is realized using the SOAP protocol [39]. The message format is XML. Supervisors also provide a graphical user interface using a simple web server. One single supervisor can hold a set of instances of the controller classes allowing control of several hardware boards at the same time.

A second type of supervisor exists (service supervisors), which does not control hardware, but establishes the connection to other services, like the detector control system (DCS) which is used for controlling and monitoring the detector power distribution. This interconnection between the DAQ and DCS system is described in Section 7.3.

In order to operate the POS, there has to be always a main (service) supervisor, which orchestrates all the other hardware and service supervisors. This supervisor processes all commands received from the CMS Central DAQ system during global data taking and provides the main user interface during local detector calibrations.

7.2 Distributed Software Architecture

One advantage of the described software infrastructure is that it is scalable and can be distributed on many different computing nodes. The overall software infrastructure is defined in one common XML configuration file, such that each running process is aware of all the other existing processes in its environment. The current CMS Phase-1 pixel software configuration consists of 38 instances of different supervisors:

- 1 main supervisor (PixelSupervisor),
- 1 DCS service supervisor (PixelDCSFSMInterface),
- 1 AMC13 hardware supervisor (PixelAMC13Supervisor) controlling 12 AMC13 boards,
- 12 FED hardware supervisors (PixelFEDSupervisor) controlling 108 FEDs,
- 12 Pixel FEC hardware supervisors (PixelFECSupervisor) controlling 16 Pixel FECs,
- 3 Tracker FEC hardware supervisors (PixelTKFECSupervisor) controlling 3 Tracker FECs,
- 8 TCDS hardware supervisors (PixelTCDSSupervisor) controlling 8 TCDS boards.

These software instances are distributed over 12 worker nodes featuring 20 cores and 32 GB RAM each. The number of computers has been chosen in order to follow the organization of the pixel detector back-end hardware in 12 microTCA crates. In addition to the 12 worker nodes, 12 additional machines act as control-hubs, defining the gateways to the individual microTCA crates.

7.3 Interface to the Detector Control System

The front-end needs to be configured differently depending on the power status of the detector. For example, a sensor module becomes noisy in case of no external bias voltage. For this reason the different PixelFECSupervisors need to be informed of any state change of the power system of the pixel detector in DCS. The PixelDCSFSMInterface subscribes to the state of individual power supply channels in the DCS and evaluates the power state of a group of power supplies that power parts of the detector controlled by one PixelFECSupervisor. The summary power state is based on a single majority voting, i.e. one single different power supply state is enough to change the summary power state. The new summary state is transmitted to the corresponding supervisors and is considered in the next front-end configuration.

8 Operation Performance

The CMS Phase-1 pixel detector has collected data in 2017 and 2018 with 95.5% and 94.4% functional channels, respectively. Software recovery mechanisms and periodic ROC resets are implemented to reduce dead-time, ensure smooth running and maintain a high level of hit efficiency for BPIX layer 1.

8.1 Software Recovery Mechanisms

One important aspect of an online control system is the ability to react to unexpected hardware states and guarantee the best performance of the hardware. While many of the simple problems, like a too high trigger rate, are handled by the FEDs themselves, more subtle problems are easier to analyze and handle in software. Within the POS framework several higher level problem recovery systems are implemented, three of which will be discussed as examples here: the recovery from an SEU in the TBM a non-responsive TBM, and a non-responsive port card.

For the recovery of an SEU in a TBM the affected sensor module must be reprogrammed. This is handled by the Pixel FECs. The Pixel FED interface corresponding to a group of Pixel FEDs collects the channels that do not send data, and if a threshold is reached, it reports this to the FEDSupervisor. In order to have a full overview of the system the information about an SEU is then sent from the FEDSupervisor to the PixelSupervisor. In the PixelSupervisor a new thread is started periodically requesting the SEU status count from all FEDSupervisors. When a programmable threshold is reached, which can differ between different parts of the detector for their impact on the data quality, a request to stop the triggers is sent to the CMS Central DAQ. When the triggers are paused the Pixel FECs are notified to reprogram all the TBM settings. When the TBM is in a controlled state the ROC settings are reprogrammed. In order to use the time of the paused triggers effectively almost all settings are reprogrammed for the whole detector. Only the trim and mask settings are programmed specifically for the sensor modules affected by the SEU. The PixelSupervisor waits until the Pixel FECs have finished this operation and then signals to the CMS Central DAQ to restart triggers. This procedure takes approximately 5s.

One problem of the current version of the TBMs is that some SEUs result in a state where the TBM no longer processes triggers. The mechanism for this is understood and has been solved in the revised version of the TBM that will be used for the replacement of the innermost BPIX layer during

LS2. For the TBMs currently used in the detector the only solution to revive the TBMs is a power-on reset of the TBM, which means that the low voltage supply of the TBM needs to be switched off and on again. Due to the design of the pixel detector this can be done by disabling the corresponding DC-DC converter; alternatively if a complete low voltage channel is disabled between 14 to 22 sensor modules are turned off. Using the DC-DC converters reduces the number of sensor modules being power cycled to between 1 and 4, depending on their position in the detector. After the sensor modules are turned on, the same procedure as described above is followed to program the TBM and ROC settings. The details of non-responsive TBM recovery can be found in Appendix H.

In rare cases the readout of a port card stops and channels connected to that port card stop sending data. To optimize the load of a single FED, the channels of a FED are distributed in the detector. As a consequence the readout of one port card is also distributed over several FEDs. This makes the detection of a missing port card only possible in the PixelSupervisor, where the information from all FEDs is combined. The previously described report chain is used and if a complete port card is reported to have satisfied a SEU, the port card is reprogrammed by the Tracker FEC, followed by the programming of all the affected sensor modules using Pixel FECs. If after this recovery the channels still do not send data to the FEDs, the corresponding channels are masked in the FEDs.

Figure 19 shows the number of ROCs that do not send data over time in BPIX layer 1 during data taking for an LHC fill in 2017. More ROCs become inactive over time due to SEUs in the TBM. After a programmable threshold is reached the SEU recovery mechanism is activated as described above, during which triggers are paused. Once the triggers are resumed the number of inactive ROCs is again at the baseline value (roughly 1% of BPIX layer 1 ROCs are not functional).

8.2 Periodic ROC Resets

As discussed in Sec 3.1, the PROC600, the readout chip for BPIX layer 1, has rare data synchronization losses in double-columns that lead to lower hit efficiencies. Both at low and high trigger rates, inefficiency is caused by a timing error in the time-stamp buffer of a double-column. Here a coincidence between a new hit and an expiring hit (i.e. a recorded hit exceeding the maximum allowed latency) can generate a spurious column drain and therefore the loss of synchronization of the double-column. This desynchronizes the readout mechanism, and the next hits are not assigned to the right event. It is more probable to observe this effect at high trigger rates. At low trigger rates another timing error can generate a spurious buffer-full signal. It happens when a buffer is empty, a hit is registered, no other hit arrives within the trigger latency and two hits are registered at exactly the trigger latency in two consecutive clock cycles. The spurious buffer-full signal by itself would not be a problem, but can lead in combination with the problem described above again to a loss of synchronization. In both cases the synchronization is restored by a reset. Both problems have been fixed in the new version of the PROC600.

In order to address the data synchronization losses of the PROC600 mentioned in Sec 3.1, periodic ROC resets at 70 Hz are issued by TCDS. Figure 20 shows the BPIX layer 1 hit efficiency versus instantaneous luminosity with and without periodic ROC resets. Issuing resets for the ROCs recovers the hit efficiencies at low and high instantaneous luminosities. If there were no periodic resets, the sharp efficiency drop above an instantaneous luminosity of 1.3×10^{34} cm^{-2} s^{-1} would have affected the layer 1 hit efficiency drastically.

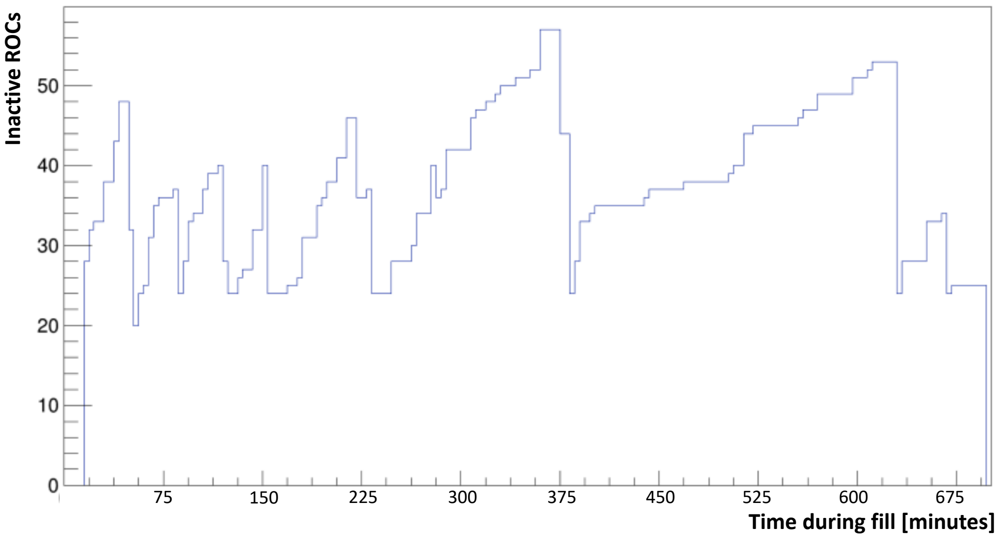


Figure 19: The number of inactive ROCs over time in BPIX layer 1 during a typical LHC fill in 2017. The number of inactive ROCs increases until a programmable threshold is reached, at which point the SEU recovery mechanism is activated and the ROCs are recovered. The SEU recovery mechanism can be activated several times during an LHC fill. The SEU rate depends on the instantaneous luminosity, which decreases over time of the fill. In the fill used for this plot, the peak luminosity was around 1.5×10^{34} cm^{-2} s^{-1}. The typical rate for inactive ROCs is 1/5minutes at an instantaneous luminosity of 1.0×10^{34} cm^{-2} s^{-1} for BPIX layer 1 modules.

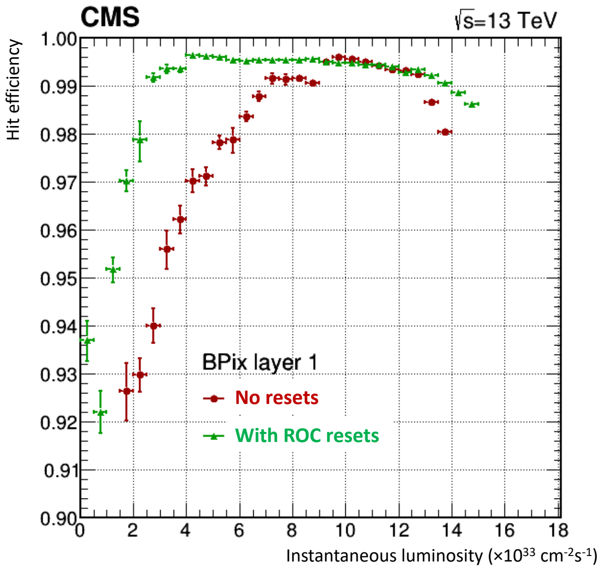


Figure 20: BPIX layer 1 hit efficiency with (green) and without (red) periodic ROC resets at 70 Hz versus instantaneous luminosity.

9 Conclusion

The CMS Phase-1 pixel DAQ system has been developed based on a combination of custom and standard microTCA parts to satisfy the higher bandwidth requirement of the new pixel detector and to interface correctly to the upgraded front-end electronics and optical links. The DAQ system underwent a series of integration tests, including readout of the pilot pixel detector, checkout of the Phase-1 detector during its assembly, and testing with the CMS Central DAQ. It was tested with realistic data stream at high trigger rates (up to 100 kHz) expected during LHC running. The Phase-1 pilot detector system proved to be valuable, leading to new designs for the TBM and the port card to address an asymmetric eye diagram and excessive clock jitter. The CMS Phase-1 pixel detector achieved the required performance improvements compared to the original pixel detector and the pixel DAQ system performed well during 2017-2018 running delivering high quality data with low dead-time consistently for CMS, without failure of any parts.

A ROC

Table 2 compares the key characteristics of the PSI46 and PSI46dig ROCs.

Table 2: Characteristics comparison of the PSI46 and PSI46dig ROCs.

	PSI46 (original)	PSI46dig (Phase-1 upgrade)
Time-stamp buffers	12	24
Hit-data buffer	32	80
Analog signal	direct readout	8-bit ADC
Readout	analog 40 MHz	digital 160 Mb/s
Single pixel threshold	3400 e^-	1700 e^-
Readout rate	100 MHz/cm^2	200 MHz/cm^2
Radiation tolerance	30 Mrad	120 Mrad

Figure 21 shows measured and simulated efficiencies for PSI46 and PSI46dig ROCs as a function of X-ray hit rates. Based on these simulations, the data loss in FPIX and BPIX layer 2-4 is expected to be less than 2%.

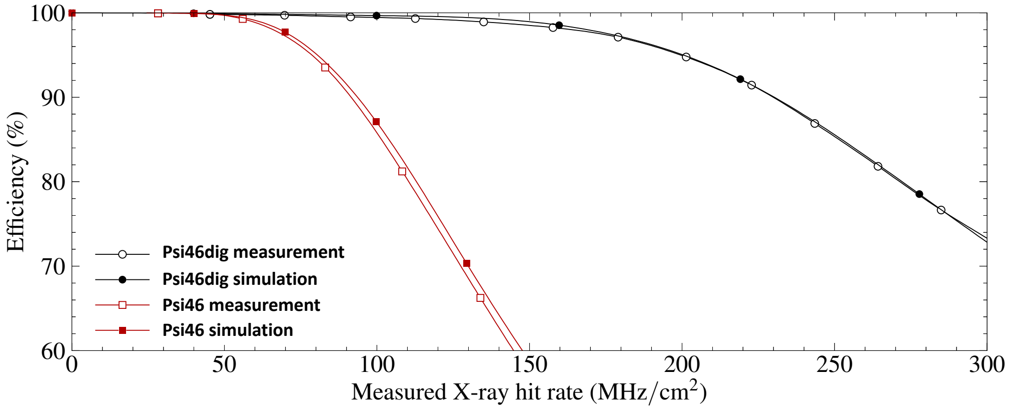


Figure 21: Measured and simulated efficiencies for PSI46 and PSI46dig ROCs as a function of X-ray hit rates.

B TBM

Figure 22 shows the block diagrams for TBM08 and TBM09/10.

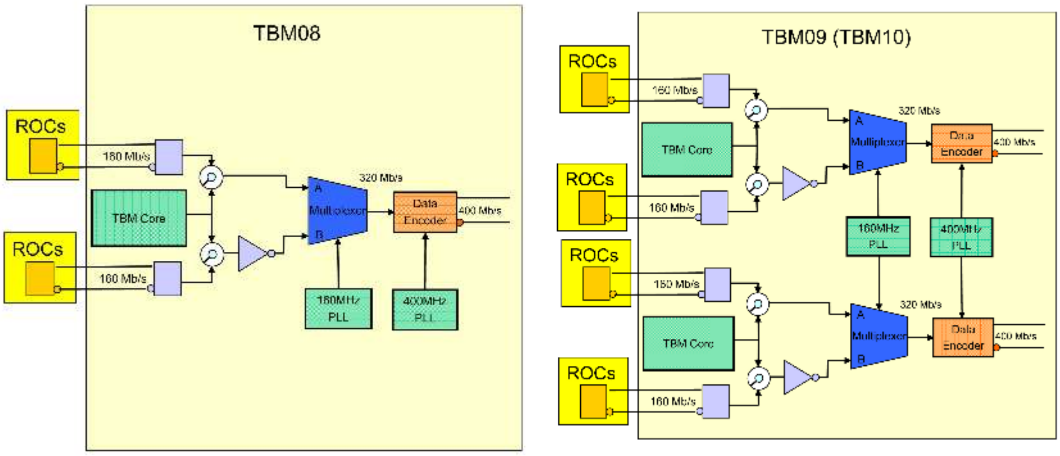


Figure 22: Block diagram of TBM08 (left) used in FPIX and BPIX layer 3 and 4, and TBM09/10 (right) used in BPIX layer 2 and 1, respectively.

The most important new block in the Phase-1 pixel TBM is the DataKeeper. The functions of the DataKeeper span four stages. The first stage is data stream inversion. One of the two output data streams is inverted to identify it as one of two data streams, TBM$_\alpha$ or TBM$_\beta$. This is done to uniquely identify each data stream at the receiving end of the optical link. The second stage is bit interleaving and building of 4-bit words. Two 4-bit words are created as follows: Word A = (1st TBM$_\alpha$ bit, 1st TBM$_\beta$ bit, 2nd TBM$_\alpha$ bit, 2nd TBM$_\beta$ bit), Word B = (3rd TBM$_\alpha$ bit, 3rd TBM$_\beta$ bit, 4th TBM$_\alpha$ bit, 4th TBM$_\beta$ bit). Word A is the first being encoded, while Word B is the second to be encoded. Word B is used in deciding when a frame signal can be transmitted. The third stage is encoding and framing. Word A and B are encoded as a 5-bit symbol using a standard 4b/5b encoding, shown in Tab. 3. The hex value 0xA is designated as a special case. There are two choices for the configuration used to represent 0xA. If Word A = 0xA, and if the Word B string begins with a ‘0’, then 0xA is represented by ‘10110’. Otherwise, if the Word B string begins with a ‘1’, then 0xA is represented by ‘10000’. The string ‘10000’ forms a unique pattern in the data stream. This framing pattern allows the FED to identify the first bit in the final serial data stream. Stage four is NRZI encoding. NRZI is an encoding scheme for a serial data stream. In this system, a ‘1’ is represented as a transition, from ‘1’ to ‘0’, or ‘0’ to ‘1’, depending on the previous value transmitted. A ‘0’ is represented by the absence of a transition, i.e. the previous data bit is repeated.

Figure 23 shows the readout scheme with the different TBMs used in the Phase-1 pixel detector. The TBM08 has one core (A) with two channels (α and β), each transferring data at 160 Mb/s. TBM09 has two cores (A and B) with two channels (α and β) each. TBM10 has two pairs of two cores (A and B) with two channels (α and β) each.

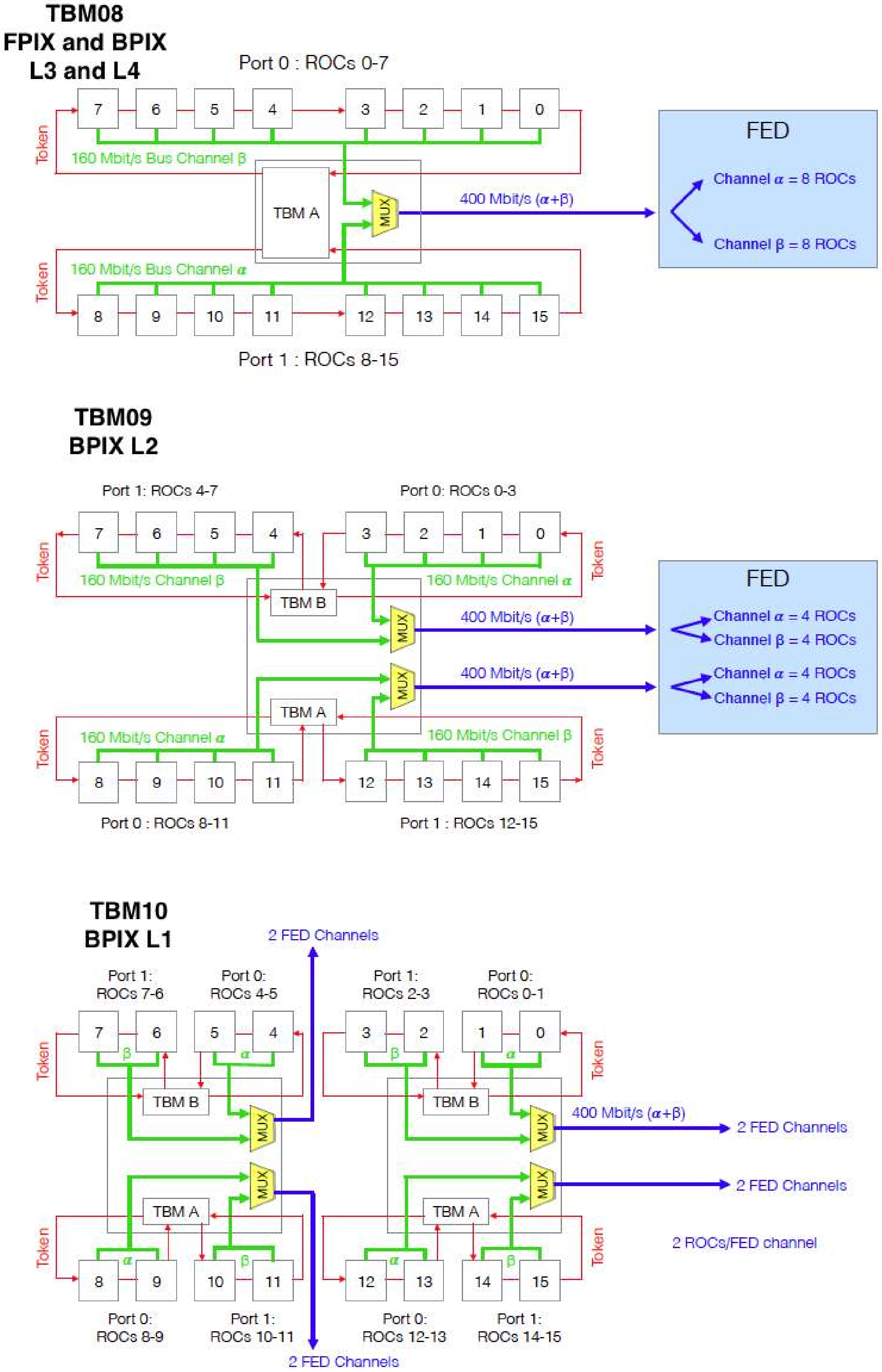


Figure 23: Readout scheme of the different TBMs used in the Phase-1 pixel detector.

Table 3: 4b/5b encoding.

4 bit binary	Hex value	Symbol
0000	0	11110
0001	1	01001
0010	2	10100
0011	3	10101
0100	4	01010
0101	5	01011
0110	6	01110
0111	7	01111
1000	8	10010
1001	9	10011
1010	A	10110 / 10000
1011	B	10111
1100	C	11010
1101	D	11011
1110	E	11100
1111	F	11101

C Rack Layout

Overall, a total of 127 AMCs, including 108 FEDs, 16 Pixel FECs and 3 Tracker FECs, distributed over a total of 12 crates, is required to control and readout the BPIX and FPIX detectors. The rack layout is shown in Fig. 24.

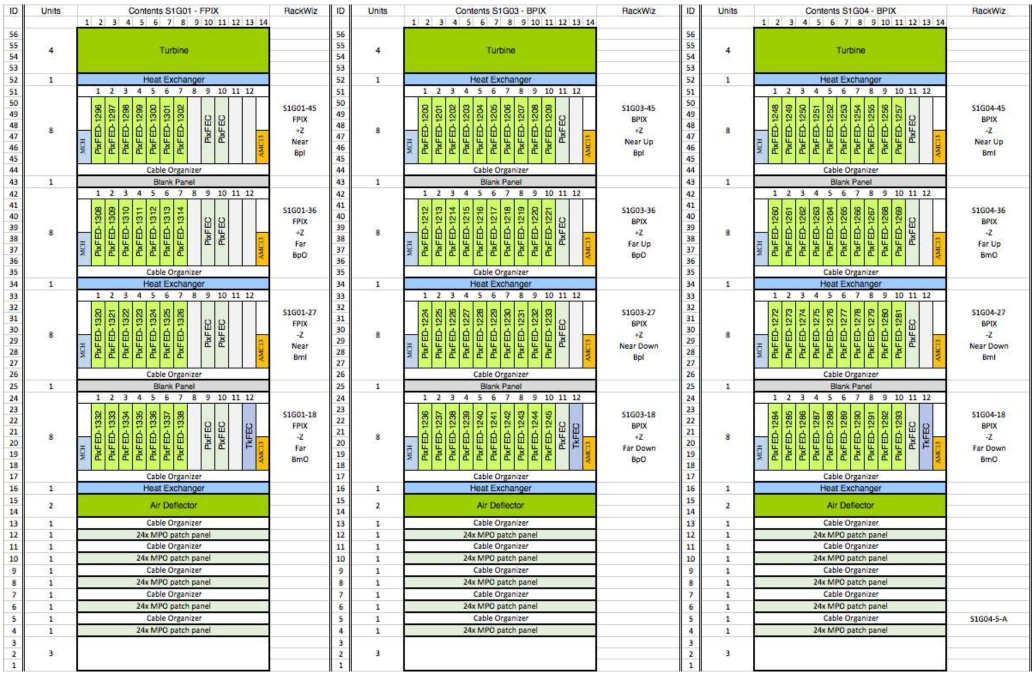


Figure 24: The layout of FEDs, FECs and optical patch panels across racks. There are seven FEDs and two Pixel FECs in each crate that controls and reads out one of the four partitions of the FPIX detector. There are ten FEDs and one Pixel FEC in each crate that controls and reads out one of the eight partitions of the BPIX detector. There are a total of three Tracker FECs, one for FPIX and two for BPIX, configuring auxiliary electronics.

D POH

The block diagram of the POH is shown in Fig. 25.

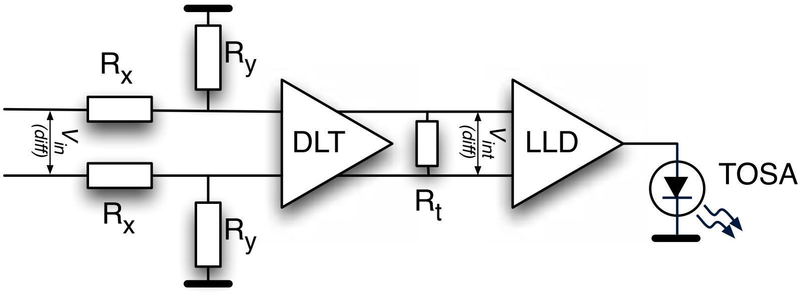


Figure 25: Block diagram of a single-channel of a POH.

The change in POH led also to a need to replace the first part of the optical fiber cabling plant – the so-called Fanout in Fig. 3. Since it was not possible to exchange the multi-ribbon cables, the optical connector interface at Patch Panel (PP)1 remained unchanged, being of type Multiple Fiber System (MFS). PP0 is located at the tracker bulkhead, PP1 is located at the inner bore of the cryostat and PP2 is located in the counting room below the pixel detector racks. Given that the number of 12-fiber ribbons needed to read out and control the upgraded detector increased by almost 30% with respect to the number previously installed, it was necessary to install additional optical fiber cables between PP0 and PP1. Unused spare cables were utilized between PP1 and the pixel back-end. This brought the opportunity to change the optical connector type used at PP0, from the MFS connector used previously to the Multi-fiber Push On (MPO) (or MTP) type. All the fibers were changed to MPO type between PP0 and PP1. MPO connectors do not require the use of a tool to (dis-)connect, making the installation and removal time shorter, which is important to maintain the serviceability of the detector in a radiation environment.

E Tracker FEC

E.1 Functionality

Figure 26 shows the block diagram for the Phase-1 Pixel Tracker FEC.

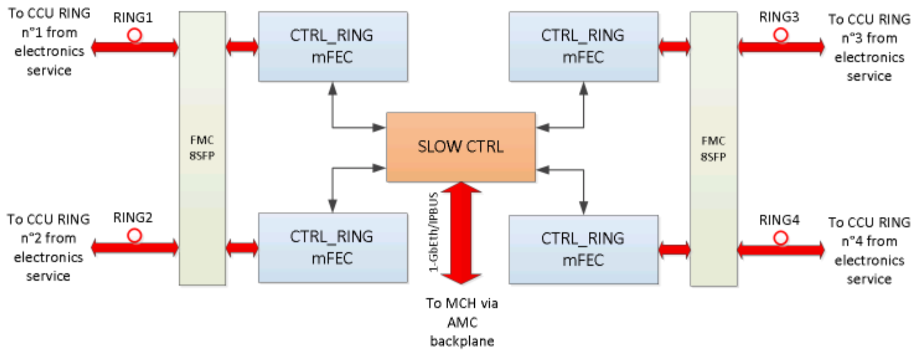

To CCU RING n°1 from electronics service
RING1
FMC 8SFP
CTRL_RING mFEC
CTRL_RING mFEC
RING3
To CCU RING n°3 from electronics service
SLOW CTRL
To CCU RING n°2 from electronics service
RING2
1-GbEth/IPBUS
CTRL_RING mFEC
CTRL_RING mFEC
FMC 8SFP
RING4
To CCU RING n°4 from electronics service
To MCH via AMC backplane

Figure 26: The block diagram of the Phase-1 Pixel Tracker FEC.

E.2 Redundancy

A redundancy scheme is implemented to face potential failures. At startup, the ring A (colored in green in Fig. 7) is the default ring to propagate the commands from the control ring firmware block (CTRL_RING) to the CCU ring through the DOH A. The ring B is the redundant one which can be partially or totally used in case of a failure of one of the components:

- Failure of DOH A:
  - The ring B is used by using the spare/dummy CCU, the fifth CCU connected to the DOH B.
- Failure of one CCU of the CCU ring:
  - The ring B is totally or partially used, depending on the desired configuration.

If a faulty component needs to be bypassed, the ring path should be changed and re-routed:

- By enabling the appropriate input and output (I/O) from the CTRL_RING:
  - These I/O switching operations are performed by simple registers in the firmware controllable by the control software.
- By enabling the appropriate I/O from a certain number of CCUs:

  - These I/O switching operations are programmable and are performed from specific commands sent to CCUs.

For instance:

- Procedure to follow to bypass the DOH A:
  - Command to CCU1: switch CCU1 input (from IA to IB).
  - Command to CTRL_RING : switch the rings (from ring A to B).
  - Command to CCU4: switch CCU4 output (from OA to OB).
- Procedure to follow to bypass CCU2:
  - Command to CCU1: switch CCU1 output (from OA to OB).
  - Command to CCU3: switch CCU3 input (from IA to IB).

The commands transmitted to the TX line are not forwarded to the RX line of the ring. The ring is open and the RX data are not read and not analyzed by the firmware.

E.3 The register write/read commands

For each command a handshaking operation is done for each transaction by forwarding the acknowledge (ACK) of the transaction being executed. For each command type, the command can be decomposed as:

- Register write command:
  - Transaction T1 from FMC to CCU via the TX line of the FMC.
  - ACK(T1) from CCU to FMC via the RX line of the FMC.
- Register read command:
  - Transaction T1 from FMC to CCU via the TX line of the FMC.
  - ACK(T1) from CCU to FMC via the RX line of the FMC.
  - Transaction T2 from CCU to FMC via the RX line of the FMC:
    - T2 contains the register data to read.
  - ACK(T2) from FMC to CCU via the TX line of the FMC.

E.4 Data and end of frame formats

The format of a token frame injected to the ring is 1 byte for start of frame and 2 bytes for end of frame. The format of a data frame for transmitting a command is shown in Tab. 4.

The two nodes that communicate with each other are determined by the destination address (DEST) and SOURCE fields. They contain an address value coded in 8-bit. Each node has its own address. The address of the master node (CTRL_RING) is set to 0×00. The address of each slave node (CCU) is pre-defined and set to a value between 0×01 and 0×7F. An address value set to 0x80

Table 4: The data frame format.

Start of frame (Token Marker)	DEST	SOURCE	LENGTH	DATA	CRC16	End of frame
1 byte	1 byte	1 byte	1 byte or 2 bytes	2 to 32767 bytes	2 bytes	2 bytes

allows broadcasting the commands to all CCUs at the same time. A 16-bit wide cyclic redundancy check (CRC16) is computed from DEST to DATA (including SOURCE and LENGTH), allowing protection of the data frame transfer. The data frame is terminated by 2 bytes marking the end of frame. The end of the frame is composed of 4 symbols, each one coded in 4-bits, as explained in Tab. 5.

Table 5: The end of frame format.

	End of frame (data frame)			
Number of bytes	2			
Number of 4-bit symbols	4			
4-bit symbol Case Transaction	T Delimiter	R	R	R
4-bit symbol Case ACK (Transaction)	T Delimiter	R or S R: no symbol error S: symbol error	R or S R: address not seen S: address seen	R or S R: data not copied S: data copied

The addressed node (one CCU) receiving the command through the transmitted data frame should return the ACK of this data frame to the sender. The ACK is the replica of the transmitted data frame except that the end of frame is modified. The last three 4-bit words from the end of frame are settable to report that a symbol error has been detected, the address has not been recognized, or the data has not been copied.

E.5 CTRL_RING firmware architecture

Figure 27 describes the CTRL_RING firmware architecture.

It consists of five main parts:

- TX Block, which is responsible for:
  - transmission of the data frames (commands) from TRA FIFO (or RET FIFO if not empty),
  - CRC16 computing,
  - 4b/5b encoding,
  - NRZI encoding and serialization,
  - Ring multiplexing in case of a switch request.
- RX Block, which is responsible for:

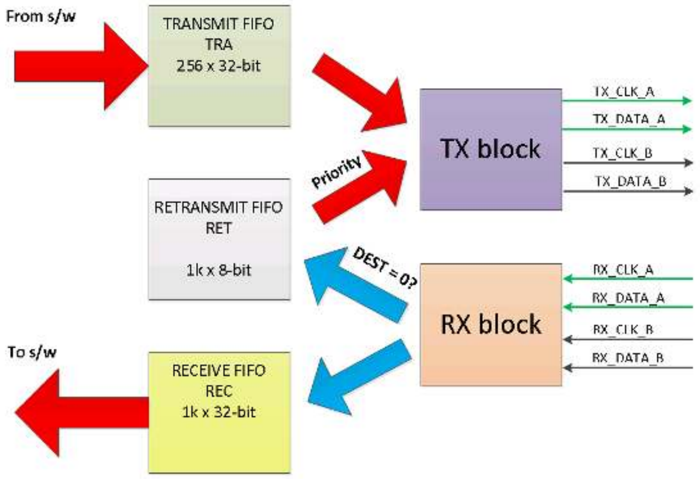


Figure 27: CTRL_RING firmware architecture.

  - ring multiplexing in case of a switch request,
  - deserialization, NRZI decoding and alignment with the injected token frame,
  - 4b/5b encoding,
  - reception of the data frames and storage in REC FIFO (RET FIFO too, if DEST = 0),
  - CRC16 checking and status updating.

- TRA FIFO:
  - Stacks the commands to be executed out of the ring.
  - Writable by software.
- REC FIFO:
  - Stores the data coming back from the ring.
  - Readable by software.
- RET FIFO:
  - Stores the data coming back from the ring (if DEST = 0).
  - The data from the RET FIFO has to be returned to the ring for ACK.

F Pixel FED

F.1 DECODE FED Firmware

The TBM combines two readout channels and uses an encoding scheme with only 17 valid 5-bit symbols, including a special symbol for frame finding, which enables the possibility to search for symbol errors, and to find the correct sampling point. A Xilinx FPGA Idelay2, which is a 31-tap, wrap-around, delay primitive with a calibrated resolution of about 80 ps, is used to shift the copy of the incoming serial data stream and check each tap position for symbol errors. This results in a 32-bit word where a ‘1’ stands for a delay tap setting with symbol error and a ‘0’ for a delay tap setting where no symbol error was found. In order to generate the bit pattern, symbol errors for each delay tap position are accumulated within a time window of 13.1 ms and the measurement is repeated continuously every 1.7 s. In addition a ‘Phase Finding Now’ mode was implemented to speed up TBM PLL scans, where the phase relationship between the incoming serial data stream and the FED system clock is changed frequently. ‘Phase Finding Now’ is initiated by an IPBus command and uses 6.5 ms integration time per delay tap setting.

The received bit pattern and the parameters for the final calculation of the used delay tap position are readable for each fiber over IPBus.

Here is a readout example: 11111100000000000000000000000111. The window that contains zeros is identified and the mid-point taken as the sampling point. There is also a manual phase setting possibility by setting the delay tap parameters over IPBus. For monitoring purposes a symbol histogram is implemented to check for symbol distribution and input link saturation by calculating the ratio between idle and other symbols.

The TBM sends a special symbol for frame finding by assigning two patterns to Hex value 0xA. If 0xA in the serial data stream is followed by a ‘1’, it is translated to ‘10000’ which forms either ‘11111’ or ‘00000’ after NRZI encoding. This unique pattern is used for resetting the frame counter and aligning the FED to the incoming data packets.

The DECODE firmware block performs the reverse functionality of the TBM DataKeeper in four steps. NRZI decoding is followed by the 5b/4b conversion, which results in a 4-bit data word. Due to the bit interleaving structure of the TBM cores two of these 4-bit words hold the readout channel TBM$_\alpha$ and readout channel TBM$_\beta$ information, as follows: Word A = (1st TBM$_\alpha$ Bit, 1st TBM$_\beta$ Bit, 2nd TBM$_\alpha$ Bit, 2nd TBM$_\beta$ Bit) and Word B = (3rd TBM$_\alpha$ Bit, 3rd TBM$_\beta$ Bit, 4th TBM$_\alpha$ Bit, 4th TBM$_\beta$ Bit).

F.2 BUILD FED Firmware

F.2.1 TTC Firmware Block

The TTC firmware block, shown in Fig. 28, has to handle receiving, re-generating and propagating the BX clock (and its derivatives) to the rest of the firmware. The sub-block ‘CLOCK GEN’ has the role of generating all the user clocks. The TTC firmware block also needs to handle receiving, decoding and propagating of the TTC signal to the whole firmware. The sub-block ‘TTC DECODER’ has the responsibility to receive and decode the encoded TTC signal serialized at 160 Mb/s. In order to ensure an efficient transmission of the TTC signal, the TCDS system uses an encoding scheme via a Time Division Multiplexing (TDM) and a Bi-Phase Mark (BPM) encoding to encode two channels (A and B) onto one channel. The channel A is dedicated to transmit L1A triggers, a 1-bit decision being sent on every BX. The channel B is suited to transmit general or user commands that are needed to control the acquisition. The TTC decoder part of the firmware handles the de-interleaving of the two channels and their decoding before delivering the L1A triggers. The EC0, the BX counter reset (BC0) and the RESYNC are the TTC commands mainly used in the firmware.

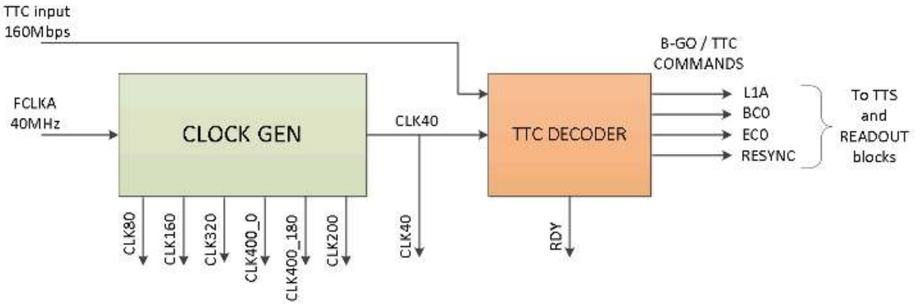
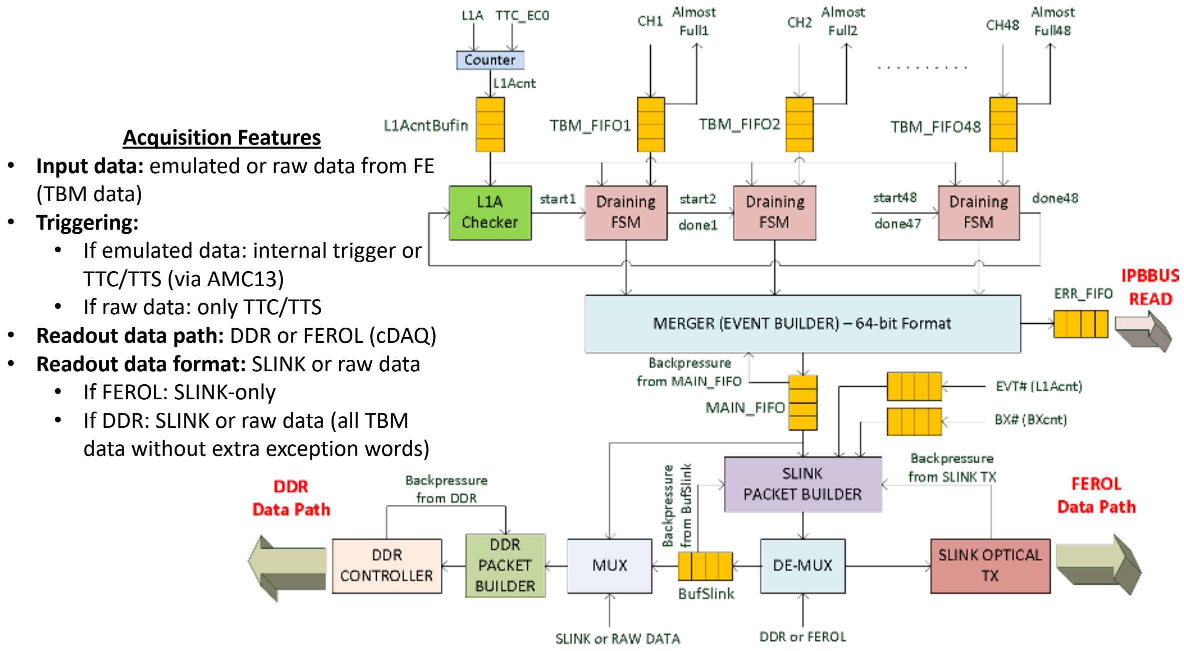


Figure 28: Block diagram of the FED TTC Block.

F.2.2 READOUT Firmware Block

The architecture of the READOUT firmware block is shown in Fig. 29.



Acquisition Features

- **Input data:** emulated or raw data from FE (TBM data)
- **Triggering:**
  - If emulated data: internal trigger or TTC/TTS (via AMC13)
  - If raw data: only TTC/TTS
- **Readout data path:** DDR or FEROL (cDAQ)
- **Readout data format:** SLINK or raw data
  - If FEROL: SLINK-only
  - If DDR: SLINK or raw data (all TBM data without extra exception words)

Figure 29: Block diagram of the FED READOUT firmware block.

The first stage (marked as TBM_FIFO in Fig. 29), embedding 48 buffers (FIFO resources from the FPGA matrix), is used to store both

- the data coming from the DECODE firmware, and
- the L1A counter or event number.

The block receives data from 48 channels (CH1 to CH48), handled and transmitted by the DECODE FED firmware. A strobe signal is added to each channel data, which allows to directly buffer the data. An individual FIFO is dedicated to each channel. The data are in 36-bit format and are encapsulated in a frame respecting the format shown in Tab. 6.

Table 6: 36-bit data format.

Channel data from DECODE to BUILD	
Bits [35:32]	Bits [31:0]
ID	Data from TBM or specific information from DECODE firmware

The ID field marks the type of 32-bit data contained in the remaining channel data word. They are listed in Tab. 7. The first four ID data are sent by the TBM and retransmitted by the DECODE firmware. The last one is computed by the DECODE firmware and appears in case of hit overflow. The ROC data from DECODE to BUILD firmware is removed in case of overflow to reduce the volume of data per channel to be buffered.

Table 7: 32-bit data format.

ID type	Binary code (4-bit)	Hex code	Description
HEADER	1000	8	Start of frame
ROC	1100	C	ROC data
PIXEL	0001	1	Pixel data
TRAILER	0100	4	End of frame
TRAILER ERROR	0110	6	Special end of frame

The second stage is the draining operation by an event presence checker (L1A checker) and 48 draining FSMs (one per channel). The sequence is as follows:

- Check the presence of an event (L1AcntBufIN not empty).
- Sequential drain from channel 1 to 48 based on the ID data seen previously.
- Then loop-back.

The individual draining process allows outputting an event fragment from the considered channel. An event fragment is composed of

- pixel data, and
- error words due to specific cases triggered by the process.

During normal operation without errors (or exceptions), only the pixel data should appear in the readout data flow. However, the front-end electronics (ROCs and TBMs) are exposed to conditions giving rise to errors in the data going to the back-end electronics. The firmware is written so that errors are detected and marked in the readout data flow.

The error words appearing in the readout data flow are:

- Time Out (TO) word, written when no data is present in the TBM FIFO after a period of time configurable by the user.
- Event Number Error (ENE) word written when the local event number from the firmware (L1Acnt) differs from the event number from the TBM.
- Trailer Error (TRLE) word, appearing for some triggered conditions detected by the BUILD firmware:
  - hit overflow,
  - wrong number of ROCs,
  - auto reset sent,
  - TBM internal reset for large payload sent.
- Channel Auto-Masked (CHMASK) word, written when the channel in progress is auto-masked due to consecutive OOS. This is the handle to disable a corrupted channel due to SEUs.

The error types TO or ENE are accumulated in this stage and when a threshold is reached (configurable by IPBus), a synchronization loss signal is triggered and a request is sent to the TTS state machine to go to OOS.

For debugging purposes, error words are also buffered in a dedicated FIFO (ERR FIFO) readable by IPBus.

The third stage, ‘MERGER’ or ‘EVENT BUILDER’, builds an entire event by merging all event fragments (pixel data and errors words) of all channels. For compliance with the following processing block ‘S-Link PACKET BUILDER’, the event building is done in a 64-bit format. If the number of hits (from all channels) is odd, an additional word (GAP word) is added at the most significant part of the last 64-bit word. The insertion of the GAP word is shown in Tab. 8; as an example only two active channels are considered. A specific marker is added at the end of each event to mark the separation with the next event.

The fourth stage is the global buffer (MAIN FIFO) storing and aggregating the consecutive pixel events separated by a specific marker.

The fifth stage is the S-Link PACKET BUILDER, which encapsulates each pixel event from the MAIN FIFO in a data format respecting the S-Link common data format.

A pixel event encapsulated in an S-Link packet is shown in Tab. 9.

The data payload corresponds to the entire pixel event stored in the MAIN FIFO. The size of this part is variable and depends on the number of hits and errors seen by the firmware. The BX ID (12-bit binary value of the BX counter when the L1A trigger is received by the BUILD firmware)

Table 8: An example of GAP word insertion in the event building stage.

Most significant part [63:32]	Least significant part [31:0]
Pixel Data CH1	Pixel Data CH1
Pixel Data CH2	Error Word CH1
Pixel Data CH2	Pixel Data CH2
GAP	Pixel Data CH2
End of Event Marker	
Pixel Data CH1	Pixel Data CH1
Pixel Data CH1	Pixel Data CH1
Pixel Data CH2	Pixel Data CH2
Error Word CH2	Pixel Data CH2
End of Event Marker	

Table 9: An encapsulated pixel event in an S-Link packet.

S-LINK HEADER (64-bit word) (major info: BX ID, LV1 ID)
DATA PAYLOAD (variable size: N × 64-bit words) Data from the MAIN FIFO
S-LINK TRAILER (64-bit word) (major info: EVT LGTH, CRC)

and the LV1 ID (event number being coded on 24-bit) are comprised into the SLINK HEADER field. The EVT LGTH (the event length), or the event packet size, is the number of 64-bit words forming the event packet by considering the header, the data payload and the trailer. Its value, added in the SLINK TRAILER field, is computed in realtime for each event read from the MAIN FIFO. Its minimal value is 2 (no hits and no errors words). A CRC generator through the CRC polynomial $P(x) = x^{16} + x^{15} + x^2 + 1$ is implemented in the firmware. The CRC is executed for each packet and its computed value is added to the SLINK TRAILER field. The CRC is the way for the CMS Central DAQ to verify the correctness of the received data and the packet transmission.

The final stage is composed of two readout paths, which can be selected via the software:

- DDR path implementing a DDR controller, which is needed primarily for calibrations.
- FEROL path exploiting the optical S-Link transmitter developed by the CMS Central DAQ for regular data taking.

G FED Tester

Figure. 30 shows the distributions of hits per ROC that are stored in the first SRAM.

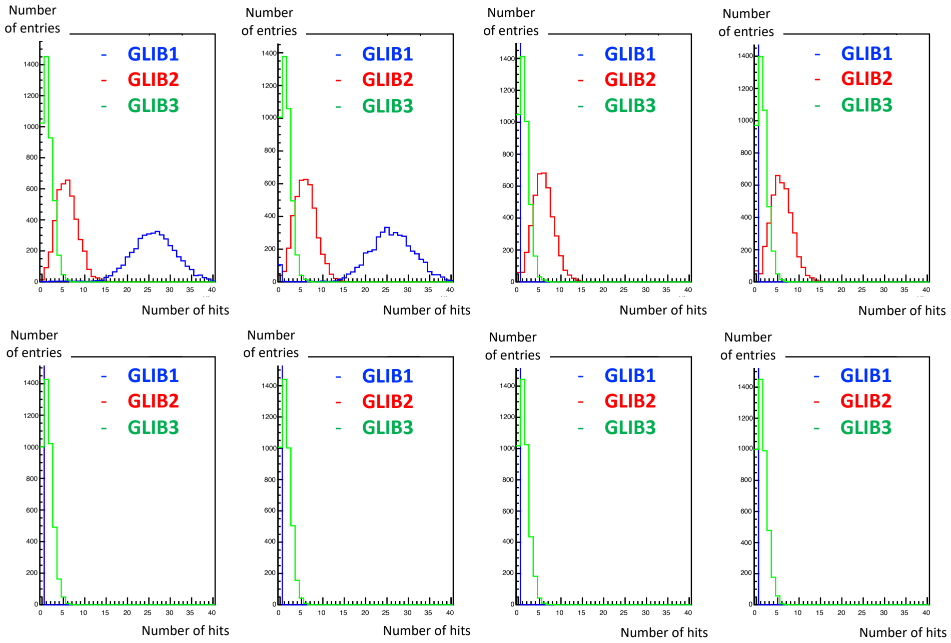


Figure 30: A Poisson hit distribution of events loaded into the SRAM of the FED tester. The distributions can be independent for each board. The number of ROCs in TBM channels can also be changed depending on the emulated layer. The green line is a GLIB emulating a BPIX layer-4 module with a poisson hit distribution of 1.4 hits/ROC, emulated in eight ROCs. The red line is a GLIB emulating a BPIX layer-2 module with a hit distribution of 6.0 hits/ROC, emulated in four ROCs. The blue line is a GLIB emulating a BPIX layer-1 module with a hit distribution of 26.3 hits/ROC, emulated in two ROCs.

H Operation Performance

Towards the end of 2017 the DC-DC converters started to fail. All converters were extracted and replaced in the year-end technical stop from December 2017 to February 2018. In May 2018 the source of the DC-DC problem was discovered and an operational solution was implemented. The problematic state of the DC-DC converter ASIC can be circumvented if their output is not disabled. This means that the higher DC-DC granularity will only be usable after the next pixel detector extraction in 2019, when all DC-DC converters will be exchanged again, using an improved version of the DC-DC converter ASIC. For the recovery chain this requires either the Tracker FEC to control the DC-DC converters, or an interface to the DCS to control the power supplies.

In the case that the DC-DC converter disabled, the report chain of the problem from FED to PixelSupervisor is the same as for the usual SEU, but from the PixelSupervisor the PixelTkFEC-Supervisor is called first to switch off and on the DC-DC converters. Once this has finished the PixelFECSupervisor is signaled to reprogram the sensor modules. The observable behavior of a non-responsive TBM is the same as that of a recoverable SEU, hence the distinction can only be made by first attempting the standard SEU recovery, and if the problem persists, switch off and on the DC-DC converter.

In the case that the DC-DC converters should not be disabled the DCS steps into the place of the Tracker FEC. 'Due to the high currents at turn on' it was decided not to recover the non-responsive TBMs during stable beams.

Acknowledgments

The tracker groups gratefully acknowledge financial support from the following funding agencies: BMWFW and FWF (Austria); FNRS and FWO (Belgium); CERN; MSE and CSF (Croatia); Academy of Finland, MEC, and HIP (Finland); CEA and CNRS/IN2P3 (France); BMBF, DFG, and HGF (Germany); GSRT (Greece); OTKA and NIH (Hungary); DAE and DST (India); IPM (Iran); INFN (Italy); PAEC (Pakistan); SEIDI, CPAN, PCTI and FEDER (Spain); Swiss Funding Agencies (Switzerland); MST (Taipei); STFC (United Kingdom); DOE and NSF (U.S.A.).

Individuals have received support from HFRI (Greece).

References

[1] CMS Collaboration, *The CMS experiment at the CERN LHC*, 2008 JINST 3 S08004, doi:10.1088/1748-0221/3/08/S08004.

[2] CMS Collaboration, *The CMS tracker system project : Technical Design Report*, CERN-LHCC-98-006, CMS-TDR-5.

[3] CMS Collaboration, *CMS Technical Design Report for the Pixel Detector Upgrade*, CERN-LHCC-2012-016, CMS-TDR-11.

[4] H.-C. Kaestli, *Frontend electronics development for the CMS pixel detector upgrade*, Nucl. Instrum. Meth. A 731 (2013) 88.

[5] R. Stringer, *A digital readout system for the CMS Phase-1 pixel upgrade*, 2015 JINST 10 C04037.

[6] J. Hegeman et al., *The CMS Timing and Control Distribution System*, 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), San Diego, CA, 2015, pp. 1-3.

[7] E. Hazen, A. Heister, C. Hill, J. Rohlf, S.X. Wu and D. Zou, *The AMC13XG: a new generation clock / timing / DAQ module for CMS microTCA*, 2013 JINST 8 C12036.

[8] J. Varela, *Timing and Synchronization in the LHC Experiments*, 6th Workshop on Electronics for LHC Experiments, Krakow, 2000, CERN-2000-010.

[9] L. Cadamuro, *The CMS Level-1 trigger system for LHC Run 2*, 2017 JINST 12 C03021.

[10] D. Gigi et al., *The FEROL40, a microTCA card interfacing custom point-to-point links and standard TCP/IP*, PoS TWEPP 2017 (313) https://doi.org/10.22323/1.313.0075

[11] S. Baron, *Timing, Trigger and Control (TTC) systems for the LHC*, CERN, 2 October 2014; Online: http://ttc.web.cern.ch/TTC/intro.html.

[12] P. Placidi, A. Marchioro and P. Moreira, *CMS Tracker PLL Reference Manual*, CERN (2000), http://cds.cern.ch/record/1069705.

[13] P. Moreira and A. Marchioro, *QPLL - a quartz crystal based PLL for jitter filtering applications in LHC*, 9th Workshop on Electronics for LHC Experiments, Amsterdam (2003).

[14] H. Furtado et al., *Delay25, an ASIC for timing adjustment in LHC*, 11th Workshop on Electronics for LHC and future Experiments, Heidelberg, 2005.

[15] L. Feld, W. Karpinski, K. Klein, M. Lipinski, M. Preuten, M. Rauch, St. Schmitz and M. Wlochal, *The DC-DC conversion power system of the CMS Phase-1 pixel upgrade*, 2015 JINST 10 C01052.

[16] C. Paillard, C. Ljuslin and A. Marchioro, *The CCU25: A network oriented communication and control unit integrated circuit in a 0.25* µm *CMOS technology*, CERN 2002-003.

[17] H.-C. Kaestli, *Design and performance of the CMS pixel detector readout chip*, Nucl. Instrum. Meth. A 565 (2006) 188.

[18] J. Hoss, *Search for Supersymmetry with Multiple Charged Leptons at* $\sqrt{s} = 13$ *TeV with CMS and Radiation Tolerance of the Readout Chip for the Phase I Upgrade of the Pixel Detector*, PhD Thesis, ETH Zurich (2017), No. 24286, https://doi.org/10.3929/ethz-b-000182698.

[19] M. Rossini, *Module Prototype Qualification for the CMS Pixel Detector Upgrade*, PhD Thesis, ETH Zurich (2015), No. 22934, https://doi.org/10.3929/ethz-a-010594693.

[20] E. Bartz, *The 0.25* µm *Token Bit Manager Chip for the CMS Pixel Readout*, https://cds.cern.ch/record/720634.

[21] E. Bartz, *CMS-doc-12626-v1: TBM08c Documentation.*

[22] M. Pesaresi et al., *The FC7 AMC for generic DAQ and control applications in CMS*, 2015 JINST 10 C03036.

[23] G. Auzinger, *Deployment of the CMS Tracker AMC as backend for the CMS pixel detector*, 2016 JINST 11 C01056.

[24] *Kintex-7 FPGAs Data Sheet*, XILINX DS182.

[25] J. Troska, F. Vasey et al., *The versatile link, a common project for super-LHC*, 2009 JINST 4 P12003.

[26] G. Cervelli, A. Marchioro, P. Moreira and F. Vasey, *A linear laser-driver array for optical transmission in the LHC experiments*, 2000 IEEE Nuclear Science Symposium Conference Record, 9/145-9/149 vol.2.

[27] T. Uemura, Y. Ishikawa, Y. Nekado, A. Izawa, M. Yoshihara and H. Nasu, *1060-nm 10-Gb/s * 12-channel parallel-optical modules for optical interconnects*, 2010 IEEE CPMT Symposium Japan, Tokyo, 2010, pp. 1-4.

[28] J. Troska, G. Cervelli, F. Faccio, K. Gill, R. Grabit, R. M. Jareno, A. M. Sandvik and F. Vasey, *Optical readout and control systems for the CMS tracker*, 2003 IEEE Transactions on Nuclear Science, Volume 50, Issue 4, pp 1067-1072.

[29] K. Kloukinas, W. Bialas, F. Drouhin, C. Ljuslin, A. Marchioro, E. Murer, C. Paillard and E. Vlasov, *FEC-CCS : A common Front-End Controller card for the CMS detector electronics*, CERN 2007-001, https://cds.cern.ch/record/1027434.

[30] C. Ghabrous Larrea, K. Harder, D. Newbold, D. Sankey, A. Rose, A. Thea and T. Williams, *IPbus: a flexible Ethernet-based control system for xTCA hardware*, 2015 JINST 2 C02019.

[31] https://www.xilinx.com/products/design-tools/vivado.html

[32] B. Akgün, *Pilot system for the Phase-1 pixel upgrade of CMS*, PoS VERTEX2015 (2015) 018.

[33] B. Vormwald, *Commissioning of the Phase-1 upgrade of the CMS pixel detector*, Springer Proc. Phys. 213 (2018) 385-389.

[34] J. Ngadiuba, *Testing and Integration of the Service Cylinders for the CMS Phase-1 Pixel*, Proceedings of TWEPP 2016, Karlsruhe, Germany, CMS-CR-2016/357.

[35] B. Akgün, *Integration and Testing of the DAQ System for the CMS Phase-1 Pixel Upgrade*, 2017 JINST 12 C02078.

[36] M. Barros Marin, S. Baron, V. Bobillier, M. Di Cosmo, S. Haas, M. Hansen, M. Joos, F. Vasey and P. Vichoudis, *A GLIB-based uTCA demonstration system for HEP experiments*, 2013 JINST 8 C12011.

[37] V. Brigljevic et al., *Using XDAQ in application scenarios of the CMS experiment*, FERMILAB-CONF-03-293, CHEP-2003-MOGT008 (2003).

[38] T. Goodale et al., *The Cactus Framework and Toolkit: Design and Applications*, High Performance Computing for Computational Science (VECPAR 2002), 5th International Conference, Porto, Portugal, 2002.

[39] D. Box et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, 2000.