

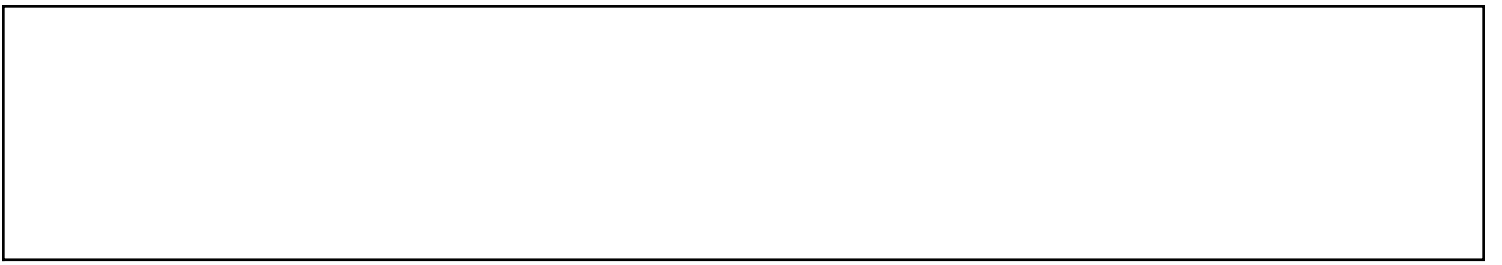
POLITECNICO DI TORINO
Repository ISTITUZIONALE

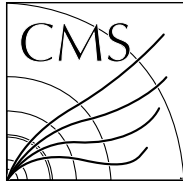
The DAQ and control system for the CMS Phase-1 pixel detector upgrade

Original

The DAQ and control system for the CMS Phase-1 pixel detector upgrade / Adam, W.; Bergauer, T.; Blöch, D.; Brondolin, E.; Dragicevic, M.; Frühwirth, R.; Hinger, V.; Steininger, H.; Beaumont, W.; Croce, D. Di; Janssen, X.; Lauwers, J.; Mechelen, P. Van; Remortel, N. Van; Blekman, F.; Chhibra, S. S.; Clercq, J. De; D'Hondt, J.; Lowette, S.; Marchesini, I.; Moortgat, S.; Python, Q.; Skovpen, K.; Bols, E. Sørensen; Mulders, P. Van; Allard, Y.; Beghin, D.; Bilin, B.; Brun, H.; Clerbaux, B.; Lentdecker, G. De; Delannoy, H.; Deng, W.; Favart, L.; Goldouzian, R.; Grebenyuk, A.; Kalsi, A.; Makarenko, I.; Moureaux, L.; Popov, A.; Postiau, N.; Robert, F.; Song, Z.; Thomas, L.; Vanlaer, P.; Vannerom, D.; Wang, G.; Wang, H.; Yang, Y.; Bondu, O.; Bruno, G.; Caputo, C.; David, P.; Delaere, C.; Delcourt, M.; Giammanco, A.; Krintiras, G.; Lemaître, V.; Magitter, A.; Piotrkowski, K.; Saggio, A.; Szilasi, N.; Marono, M. Vidal; Vischia, P.; Zobec, J.; Briglievi, V.; Ceci, S.; Ferenek, D.; Rogulji, M.; Starodumov, A.; Šušá, T.; Eerola, P.; Heikkilä, J.; Brücken, E.; Lampén, T.; Luukka, P.; Martikainen, L.; Tuominen, E.; Tuuva, T.; Agram, J. -L.; Andrea, J.; Bloch, D.; Bonnín, C.; Bourgatte, G.; Brom, J. -M.; Chabert, E.; Charles, L.; Dangelser, E.; Gelé, D.; Goerlach, U.; Gross, L.; Hosselet, J.; Krauth, M.; Tonon, N.; Baulieu, G.; Boudoul, G.; Caponetto, L.; Chanon, N.; Contardo, D.; Dené, P.; Dupasquier, T.; Galbit, G.; Lumb, N.; Mulino, L.; Nodari, B.; Perries, S.; Donckt, M. Vander; Viret, S.; Autermann, C.; Feld, L.; Karpinski, W.; Kiesel, M. K.; Klein, K.; Kopp, M.; Kuznetsov, S.; Lavezzi, B.; Leardini, M.; Luchessa, G.; Malinik, G.; Malyuk, A.; Pauls, A.; Pierschel, G.; Preuten, M.; Rauch, M.; Röwert, N.; Schael, S.; Schulz, J.; Schwering, G.; Teroerde, M.; Wlochal, M.; Zhukov, V.; Dziwok, C.; Fluegge, G.; Müller, T.; Pooth, O.; Stahl, A.; Ziemons, T.; Aldaya, M.; Asawatangtrakuldee, C.; Eckerlin, G.; Eckstein, D.; Eichhorn, T.; Gallo, E.; Guthoff, M.; Haranko, M.; Harb, A.; Keaveney, J.; Kleinwort, C.; Mankel, R.; Maser, H.; Meyer, M.; Missiroli, M.; Muhl, C.; Mussgiller, A.; Pitzl, D.; Reichelt, O.; Savitskyi, M.; Schuetze, P.; Stever, R.; Walsh, R.; Zuber, A.; Benecke, A.; Biskop, M.; Bortone, P.; Brühmann, A.; Buchalla, A.; Buehler, M.; Fend, T.; Fioresi, R.; Gadietti, A.; Gambaletti, G.; Genssler, A.; Giammanco, R.; Keszepalczyk, G.; Klanner, R.; Kutzner, V.; Lange, T.; Matysek, M.; Mrowietz, M.; Niemeyer, C.; Nissan, Y.; Pena, K.; Perieanu, A.; Rieger, O.; Schleper, P.; Schwandt, J.; Schwarz, D.; Sonneveld, J.; Steinbrück, G.; Tews, A.; Vormwald, B.; Wellhausen, J.; Zoi, I.; Abbas, M.; Ardila, L.; Balzer, M.; Barth, C.; Barvich, T.; Baselga, M.; Blank, T.; Bögelspacher, F.; Butz, E.; Caselle, M.; Boer, W. De; Dierlamm, A.; Morabit, K. El; Gosewisch, J. -O.; Hartmann, F.; Husemann, U.; Koppenhöfer, R.; Kudella, S.; Maier, S.; Mallows, S.; Metzler, M.; Muller, Th.; Neufeld, M.; Nürnberg, A.; Sander, O.; Schuler, C.; Schwaninger, M.; Schuh, T.; Shvetsov, I.; Simonis, H. -J.; Steck, P.; Wassmer, M.; Weber, M.; Weddigen, A.; Anagnostou, G.; Asenov, P.; Assiouras, P.; Daskalakis, G.; Kyriakis, A.; Loukas, D.; Paspalaki, L.; Balázs, T.; Siklér, F.; Vami, T.; Vespremi, V.; Bhardwaj, A.; Jain, C.; Jain, G.; Rajan, K.; Bhattacharya, R.; Dutta, S.; Chowdhury, S. Roy; Saha, G.; Sarkar, S.; Carola, P.; Creanza, D.; de Palma, M.; Bobarijs, G. De; Fiore, L.; Ince, M.; Liodo, F.; Maggi, G.; Martiradonna, S.; Mongelli, M.; My, S.; Selvaggi, G.; Silvestris, L.; Albano, S.; Costa, S.; Motta, A. Di; Potenza, R.; Satta, M.; Ticozzi, A.; Tuve, C.; Barbagli, G.; Brianzi, M.; Cassese, A.; Ceccarelli, R.; Ciaranfi, R.; Ciulli, V.; Civinini, C.; D'Alessandro, R.; Focardi, E.; Latino, G.; Lenzi, P.; Meschini, M.; Paoletti, S.; Russo, L.; Scarlini, E.; Sguazzoni, G.; Viliani, L.; Cerchi, S.; Ferro, F.; Mulargia, R.; Robutti, E.; Brivio, F.; Dinardo, M. E.; Dini, P.; Gennai, S.; Guzzi, L.; Malvezzi, S.; Menasce, D.; Moroni, L.; Pedrini, D.; Zuolo, D.; Azzi, P.; Bacchetta, N.; Bisello, D.; Dorigo, T.; Pozzobon, N.; Tosi, M.; Canio, F. De; Gaioni, L.; Manghisoni, M.; Ratti, L.; Re, V.; Riciputi, E.; Traversi, G.; Baldinelli, G.; Bianchi, F.; Biasini, M.; Bilei, G. M.; Bizzaglia, S.; Caprai, M.; Cecchi, C.; Checcucci, B.; Ciangottini, D.; Fanò, L.; Farnesini, L.; Ionica, M.; Leonardi, R.; Manoni, E.; Mantovani, G.; Mariani, V.; Menichelli, M.; Morozzi, A.; Moscatelli, F.; Passeri, D.; Placidi, P.; Rossi, A.; Santocchia, A.; Spiga, D.; Storchi, L.; Turrioni, C.; Androsov, K.; Azzurri, P.; Bagliesi, G.; Basti, A.;

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.; Tsiros, 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.; Luster, 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.; Burkler, 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.; Macaudo, 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.; Quinlan, 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.; Alhousseini, 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.; Rappocciolo, 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.; Karim, W. - In: JOURNAL OF INSTRUMENTATION. - ISSN 1748-0221. - 14:10(2019), pp. P10017-P10017. [10.1088/1748-0221/14/10/P10017]

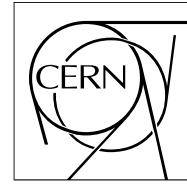




The Compact Muon Solenoid Experiment

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.

2 **The DAQ and Control System for the CMS Phase-1 Pixel** 3 **Detector**

4 **The Tracker Group of the CMS Collaboration**

5 **ABSTRACT:** In 2017 a new pixel detector was installed in the CMS detector. This so-called Phase-
6 1 pixel detector features four barrel layers in the central region and three disks per side in the
7 forward regions. The upgraded CMS Phase-1 pixel detector requires an upgraded data acquisition
8 (DAQ) system to accept the new data format with larger event sizes. A new DAQ and control
9 system has been developed based on a combination of custom and commercial microTCA parts.
10 Custom mezzanines on standard carrier cards provide a front-end driver for readout, and a front-end
11 controller for configuration and the distribution of clock and trigger signals. Before the installation
12 of the detector the DAQ system has undergone a series of integration tests, including readout of
13 the pilot pixel detector, which was constructed with prototype Phase-1 electronics and installed in
14 CMS in 2014, checkout of the CMS Phase-1 detector during its assembly, and testing with the CMS
15 Central DAQ. This paper describes the Phase-1 pixel DAQ and control system, as well as integration
16 tests and results, first operational experience, and performance.

17 **KEYWORDS:** Detector control systems; Data acquisition; Optical detector readout; Modular elec-
18 tronics

19	Contents	
20	1 Introduction	1
21	2 System Overview	2
22	3 Front-End ASICs	4
23	3.1 Readout Chip	4
24	3.2 Token-Bit Manager Chip	5
25	4 Back-End Implementation	6
26	4.1 Optical Links	7
27	4.1.1 Pixel Opto Hybrid (POH)	8
28	4.1.2 Digital Receiver	8
29	4.1.3 Control links	9
30	4.2 Phase-1 Tracker FEC	10
31	4.3 Phase-1 Pixel FEC	11
32	4.4 Phase-1 Pixel FED	14
33	4.4.1 DECODE Pixel FED Firmware	14
34	4.4.2 BUILD Pixel FED Firmware	15
35	4.4.3 Pixel FED Data Payload	16
36	5 System Tests in the CMS Detector - Phase-1 Pixel Pilot System	18
37	6 System Tests in the Laboratory	19
38	6.1 FED Tester Setup	20
39	6.2 The DAQ Setup for High Data Rates	21
40	7 Pixel Online Software	21
41	7.1 Hardware Access and Supervisors	21
42	7.2 Distributed Software Architecture	23
43	7.3 Interface to the Detector Control System	24
44	8 Operation Performance	24
45	8.1 Software Recovery Mechanisms	24
46	8.2 Periodic ROC Resets	25
47	9 Conclusion	27
48	A ROC	28
49	B TBM	29
50	C Rack Layout	32

51	D POH	33
52	E Tracker FEC	34
53	E.1 Functionality	34
54	E.2 Redundancy	34
55	E.3 The register write/read commands	35
56	E.4 Data and end of frame formats	35
57	E.5 CTRL_RING firmware architecture	36
58	F Pixel FED	38
59	F.1 DECODE FED Firmware	38
60	F.2 BUILD FED Firmware	38
61	F.2.1 TTC Firmware Block	38
62	F.2.2 READOUT Firmware Block	39
63	G FED Tester	43
64	H Operation Performance	44

65 1 Introduction

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

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

76 The goal of the Phase-1 pixel project [3] was to perform an evolutionary upgrade with minimal
77 disruption of data-taking by keeping the pixel size, sensor, and readout architecture the same,
78 while improving the performance through a higher rate capability of the ROCs, and larger data
79 transmission rate, more robust tracking through the addition of a fourth barrel layer, and a third
80 disk per endcap, as well as a reduced material budget. The Phase-1 pixel detector was designed to
81 maintain a high tracking performance at luminosities up to $2.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, corresponding to
82 an average of 80 inelastic interactions per 25 ns BX, (these interactions are referred to as ‘pileup’).
83 The Phase-1 pixel detector with modified data acquisition (DAQ) and control system was installed
84 during an extended year-end technical stop at the beginning of 2017, and is expected to deliver high
85 quality data in the high luminosity environment of the LHC up to Long Shutdown (LS) 3, which is

86 scheduled to start in 2024. The Phase-1 pixel DAQ and control system has been developed based on
87 a combination of custom and commercial microTCA parts. Custom mezzanines on CMS-developed
88 carrier cards provide a Front-End Driver (FED) for readout, as well as a Front-End Controller (FEC)
89 for configuration and the distribution of clock, fast commands and trigger signals.

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

95 2 System Overview

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

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

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

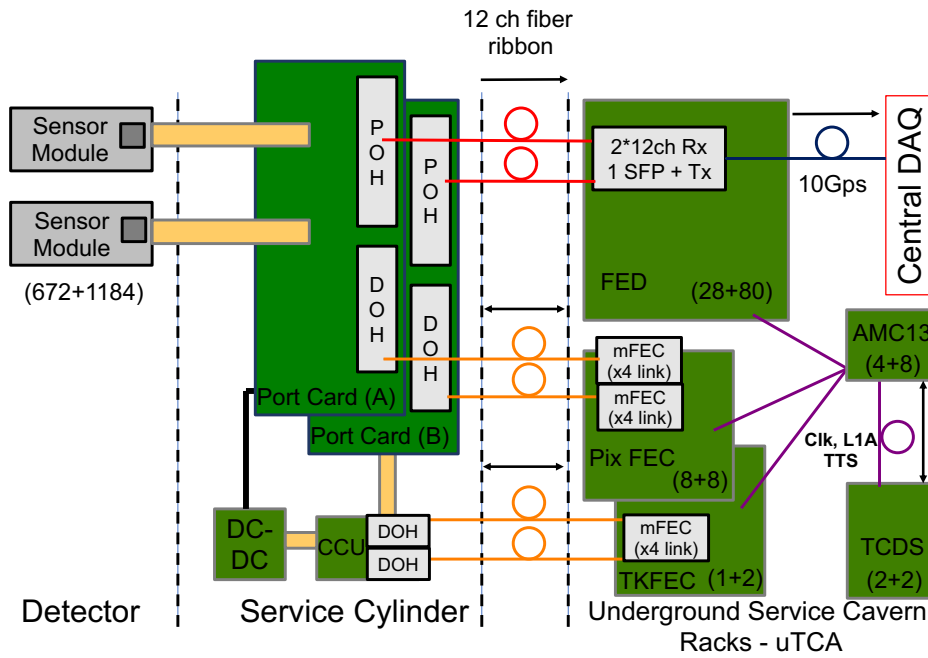


Figure 1: Overview of the microTCA DAQ and control system of the Phase-1 pixel detector. The numbers in parentheses indicate the numbers of the respective installed devices (FPIX and BPIX). Details are explained in the text.

127 depends on the status of each FED. At a given moment a FED should either accept or block CMS
 128 level-1 triggers (L1A) [9]. The pixel DAQ is able to maintain event synchronization across all FEDs
 129 with this back-pressure system. The FED sends the data to the Central DAQ Front-End Readout
 130 Optical Link-40 [10] (FEROL40) card, the first stage of the CMS Central DAQ chain. A total of
 131 108 microTCA Pixel FEDs are required to read out the Phase-1 pixel detector.

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

143 Registers on the port cards, including Delay25 chips, and DC-DC converters [15], used for
 144 powering, are programmed by the Tracker FEC via the Inter-Integrated Circuit I²C interface and
 145 Parallel Interface Adapter (PIA) port, respectively, of a Control & Communication Unit (CCU) [16].
 146 Port cards, DC-DC converters and CCUs are located in service cylinders, which distribute power,

147 cooling and optical links to the sensor modules. The optical links act as an interface between sensor
148 modules and the back-end electronics, located in the underground service cavern. A total of 3
149 microTCA Tracker FECs are required to control the Phase-1 pixel detector auxiliary electronics.

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

155 **3 Front-End ASICs**

156 **3.1 Readout Chip**

157 The ROC used in the original CMS pixel detector, PSI46 [17], was designed for hit rates of
158 a few tens of MHz/cm², encountered at BPIX layer 1 for an LHC instantaneous luminosity of
159 $1.0 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ with a 25 ns bunch spacing. This readout chip performed well during the
160 data taking periods from 2008 to 2016. However, it showed inefficiencies when operated at higher
161 data rates when the LHC started operating at instantaneous luminosities above the design value.
162 Therefore, a new pixel ROC had to be designed.

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

169 The double-column buffer sizes have been increased from 32 to 80 cells for the hits and from
170 12 to 24 cells for the time-stamps. Contrary to the analog PSI46 ROC, the PSI46dig ROC outputs
171 digital data. Hence an analog-to-digital converter (ADC) has been implemented in the chip. It is an
172 8-bit successive approximation register ADC running at 80 MHz. Digitized data are stored in a $64 \times$
173 23 bit FIFO (First In First Out), which is read out serially at 160 Mb/s. The 80 and 160 MHz clocks
174 needed for the ROC operation are generated from the external LHC clock using a PLL circuit.

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

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

186 The above improvements reduce the effective operational threshold of the ROC from 3400 electrons
187 in the original detector to 1700 electrons for the upgraded one. This is important when the amount

188 of charge per hit starts to decrease after radiation damage to the sensors: a highly irradiated detector
189 will slowly degrade in resolution. With a lower threshold, the charge sharing among neighboring
190 pixels can be exploited for position interpolation up to a higher integrated luminosity.

191 Based on operational experience with the PSI46 ROC and irradiation tests, further optimizations
192 of the internal biasing were made that extend the range of ionizing dose tolerated by the PSI46dig
193 ROC, reducing the need to re-adjust DAC settings with increasing accumulated dose. The PSI46dig
194 ROC performed well and without significant performance degradation after irradiation to up to
195 120 Mrad ($4 \times 10^{14} \text{ cm}^{-2}$, 24 MeV protons, at the irradiation facility in Karlsruhe), which is the
196 maximal dose expected during LHC operations for the Phase-1 pixel detector. A detailed study on
197 radiation tolerance of the PSI46dig ROC can be found here [18].

198 Data losses have been measured with high-rate X-ray tubes for pixel hit rates of up to
199 300 MHz/cm², and were found to be in excellent agreement with expectations based on detailed
200 architecture simulations [19]. The PSI46dig ROC has performed well during the 2017-2018 LHC
201 run, as shown in Sec. 8. All targeted improvements, i.e. low noise, lower threshold, and lower
202 inefficiency at high rates, have been confirmed during data-taking.

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

205 Despite the improved performance of the PSI46dig, its architecture would lead to unacceptable
206 data loss rates for the innermost BPIX layer, where pixel hit rates up to 600 MHz/cm² may be
207 encountered. A dedicated chip (PROC600) was designed for layer 1, with a complete re-design
208 of the double-column. The PROC600 features a four times higher hit transfer rate of pixels to
209 the end-of-column buffers, and dead-time-free buffer management. The former is achieved by
210 changing from single pixel to 2×2 pixel cluster transfers and the implementation of a simpler,
211 handshake-free protocol. A faster and more power efficient analog bus was developed for the pulse
212 height transfers. The data buffer was modified considerably; PROC600 has a ring buffer with 56
213 buffer units, each containing a cluster base address plus four analog storage cells for the charge
214 pulse heights. The readout is zero-suppressed in order to remove pixels in the cluster with zero
215 measured signal amplitude. In order to significantly reduce the dead-time during operations the
216 logic has been extended; pixel hits are stored in a ring-buffer, and those hits which are validated by
217 the L1A are read out without stopping the acquisition of new hits into the buffer. This avoids an
218 interruption of the data acquisition process in the double-column or overwriting of data, as is the
219 case in the PSI46dig ROC.

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

227 **3.2 Token-Bit Manager Chip**

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

230 The principal functions of the TBM include:

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

237 The Phase-1 pixel TBM replaces the original TBM [20]. The TBM core outputs serial data at
238 160 Mb/s. Two output data streams are encoded with a 4b/5b scheme and multiplexed by a block
239 called the DataKeeper into a 400 Mb/s stream transferred optically to the FED. There are three
240 versions of the Phase-1 pixel TBM (Table 1). The TBM08 [21], used in FPIX disks and BPIX
241 layers 3 and 4, combines two groups of ROC data, while the TBM09 and TBM10, used in BPIX
242 layer 2 and layer 1, respectively, combine the output of four groups of ROCs into two 400 Mb/s data
243 streams. The TBM09 and TBM10 differ in their timing settings, which are optimized to match the
244 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

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

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

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

252 **4 Back-End Implementation**

253 The design of the back-end electronics for the Phase-1 pixel detector is based on microTCA modular
254 electronics. A microTCA carrier hub (MCH) card is used as communication interface between the
255 microTCA electronics and the network. The microTCA backplane is used to distribute clock, trigger
256 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.

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

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

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

267 4.1 Optical Links

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

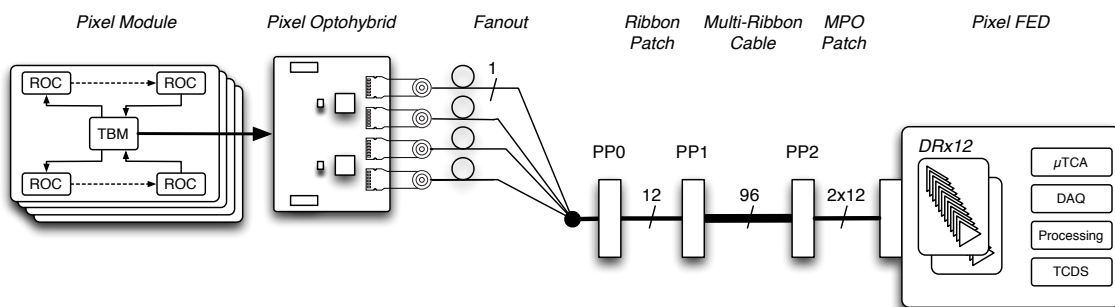


Figure 3: CMS Phase-1 pixel upgrade readout chain.

273 **4.1.1 Pixel Opto Hybrid (POH)**

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

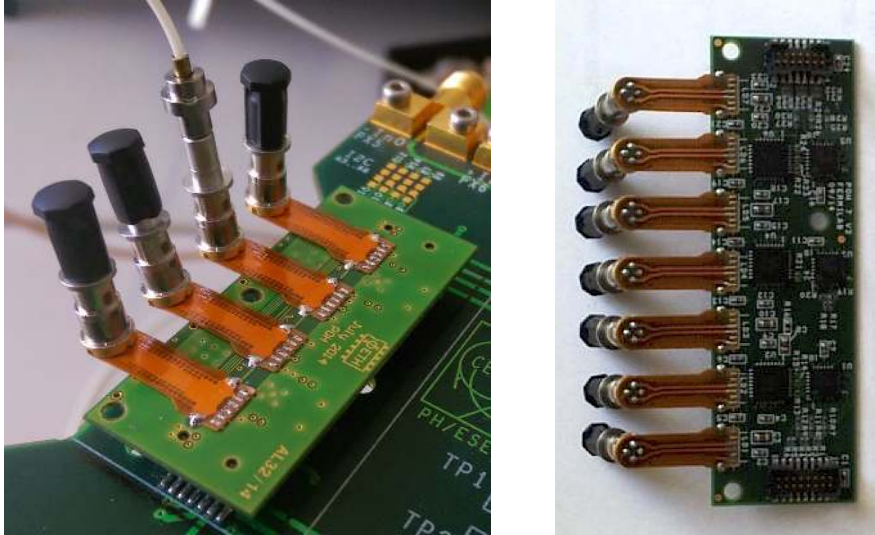


Figure 4: Photographs of a fully assembled POH4 (left) and a POH7 (right). The differences between the two are the number of transmitter channels, four in the case of the POH4 and seven in the case of the POH7, and the input matching that adapts to the signal cables of the BPIX and FPIX system respectively.

277 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
278 converts them into optical signals to be transmitted to the back-end receiver installed in the counting
279 room, about 65 m away from the detector. Each POH houses single mode Fabry-Perot laser TOSAs
280 operating at 1310 nm, Digital Level Translators (DLT) and Linear Laser Drivers (LLD) [26]. The
281 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
282 the lasers at their working point and modulate them with a current proportional to the input signal.
283 The modulation gain and pre-bias currents at the LLD are controlled through an I²C interface. The
284 POHs are used to transmit balanced digital signals at a maximum bit rate of 400 Mb/s. A typical
285 output optical eye diagram is shown in Fig. 5.
286
287

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

290 **4.1.2 Digital Receiver**

291 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
292 to identify a receiver module based on an InGaAs photodiode. Typically, such high-density multi-
293 channel receivers are based on GaAs photodiodes that operate with light around 850 nm and are
294

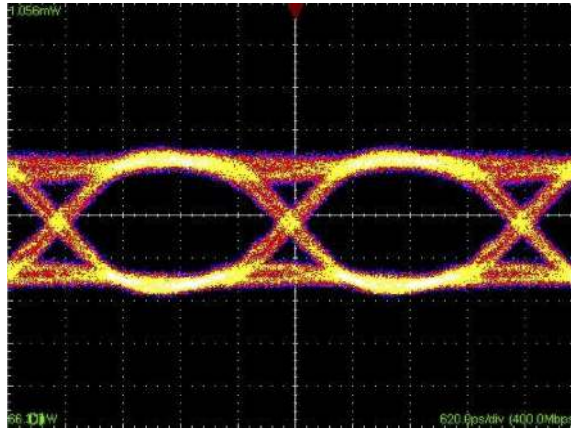


Figure 5: Typical output eye diagram measured on a POH. The horizontal scale is 620 ps/div and the vertical scale is 110 μ W/div.

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

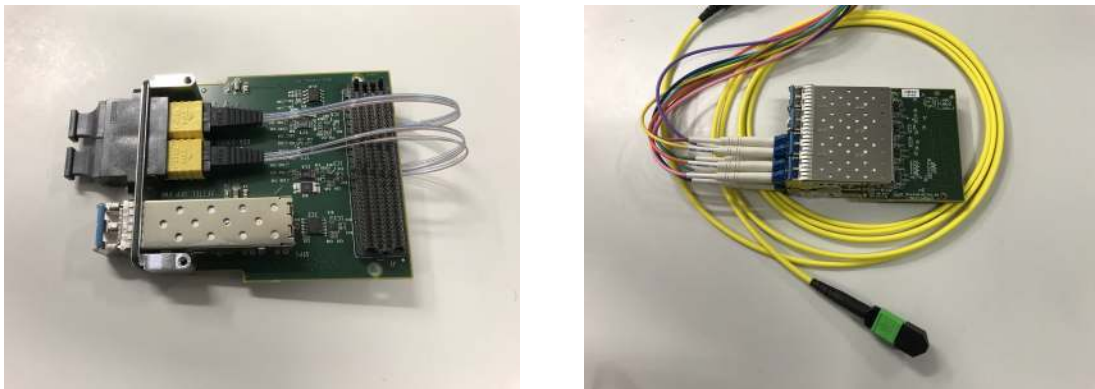


Figure 6: (Left) An Rx-FMC with 24 optical input channels feeding two FITEL 12-channel optical receivers that are optimized for the 1310 nm 400 Mb/s signal, and a SFP+ 10 Gb/s transceiver for S-Link Express to send data to the CMS Central DAQ. (Right) An FMC equipped with low-speed compatible (80 Mb/s)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

302 4.1.3 Control links

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

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

307 4.2 Phase-1 Tracker FEC

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

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

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

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

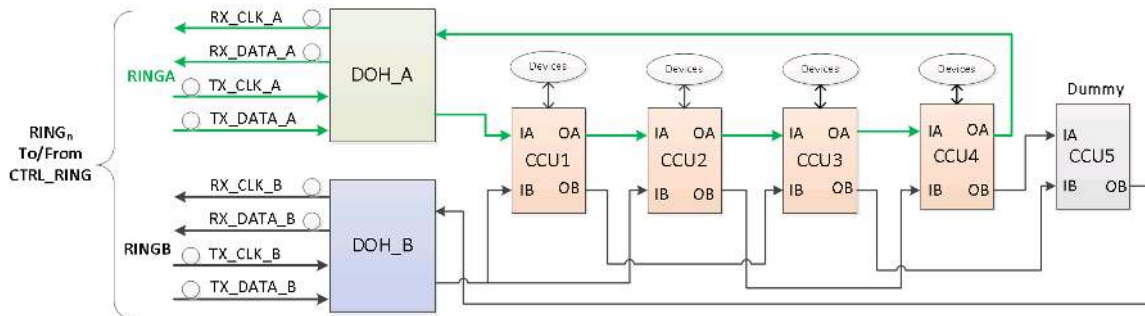


Figure 7: An example topology with two DOHs and five CCUs (the CCU5 is a spare/dummy). Ring A is the primary ring, used by default. In case of a failure, either of DOH_A or any single CCU, the device can be bypassed by switching to Ring B.

327 The CCU executes the I²C transactions towards the appropriate device from the initial command
 328 received. The commands are transmitted from the control ring firmware block (the master) via the
 329 TX line (A or B) to the appropriate CCU of the control ring (the slave). The CCUs are distinguishable
 330 by their own defined addresses. A ring-type topology is configured as a standard computer LAN

331 network connecting the control ring firmware block to CCUs and the CCUs between themselves.
332 Two types of commands can be executed from the control ring firmware block: register write
333 commands and register read commands.

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

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

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

343 4.3 Phase-1 Pixel FEC

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

347 Xilinx's Vivado [31] development tools were used for the firmware design. A block diagram
348 of the Pixel FEC at the board level is shown in Fig. 8. The firmware was developed using a standard
349 release for the FC7 card, which provides Ethernet services from the AMC backplane. A localized
350 32-bit wide IPBus allows communication with the FPGA sub-systems by addressable regions from
351 Ethernet. Pixel FEC registers and channel input and output FIFOs are interfaced via Ethernet
352 through the IPBus.

353 The input clock is sent to a PLL to produce a TTC Clock at the same frequency, as well as
354 80 MHz and 200 MHz clocks for various other sub-systems. Double data rate TTC information
355 is received through IDELAY and IDDR logic blocks and then processed in a hamming decoder
356 block. The outputs of the TTC decoder block are the L1A and fast commands on an 8-bit bus.
357 Pixel related fast commands are the ROC reset, TBM reset, CAL-SYNC reset, event clear and TTC
358 Send commands. Registers in the Pixel FEC register space count how many Pixel FEC related fast
359 commands are decoded and a FIFO can capture all the TTC events.

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

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

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

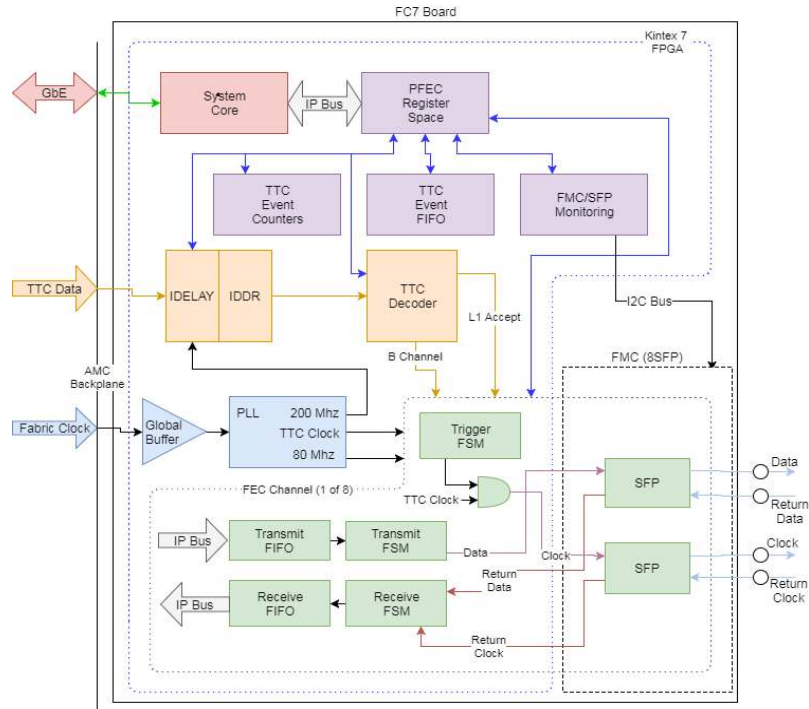


Figure 8: Block diagram of the Pixel FEC.

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

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

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

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

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

395 next hub/port command.

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

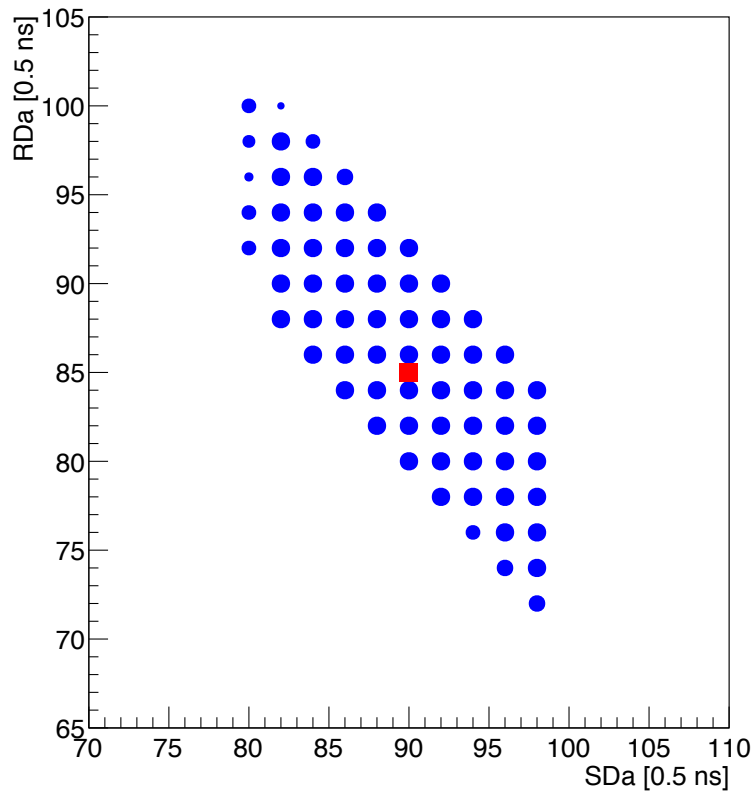


Figure 9: Efficiency of transmission as a function of the delay of returned data (RDa) and sent data (SDa). The size of the blue dots is proportional to the transmission efficiency, the largest dots being 100% efficient. The red dot is chosen as the working point since it is at the center of the area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

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

408 **4.4 Phase-1 Pixel FED**

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

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

417 **4.4.1 DECODE Pixel FED Firmware**

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

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

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

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

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

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

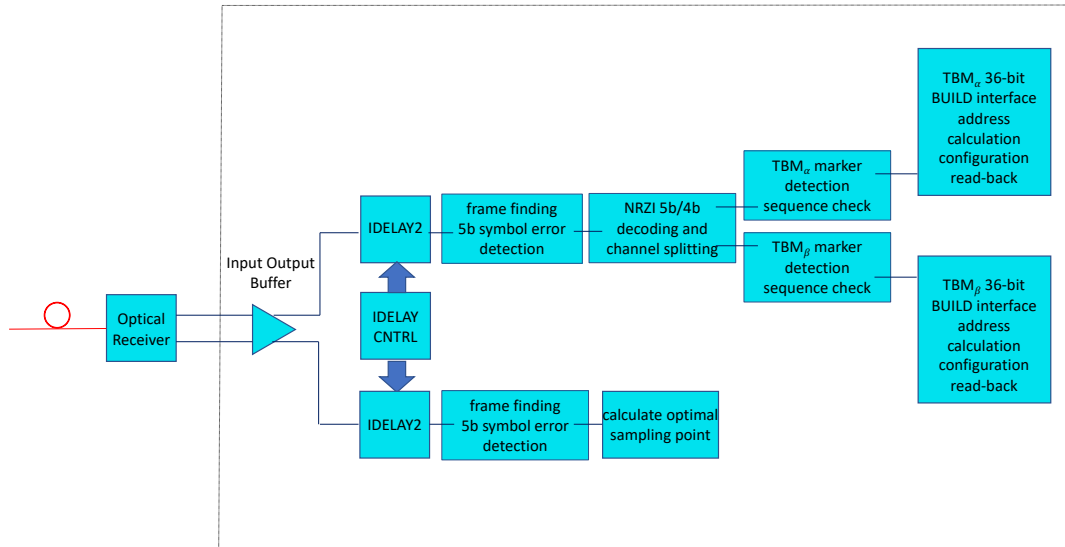


Figure 10: Block diagram of the Phase-1 pixel FED DECODE firmware.

452 4.4.2 BUILD Pixel FED Firmware

453 The BUILD FED firmware was designed to handle the data readout of 48 channels coming from
 454 the DECODE FED firmware, transmit data to CMS Central DAQ via the S-Link Express interface,
 455 and communicate with the TCDS system for the TTC and TTS interfaces for synchronization. A
 456 1-Gb/s Ethernet/IPBus communication is used to diagnose the exceptions which can occur during
 457 physics data taking.

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

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

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

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

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

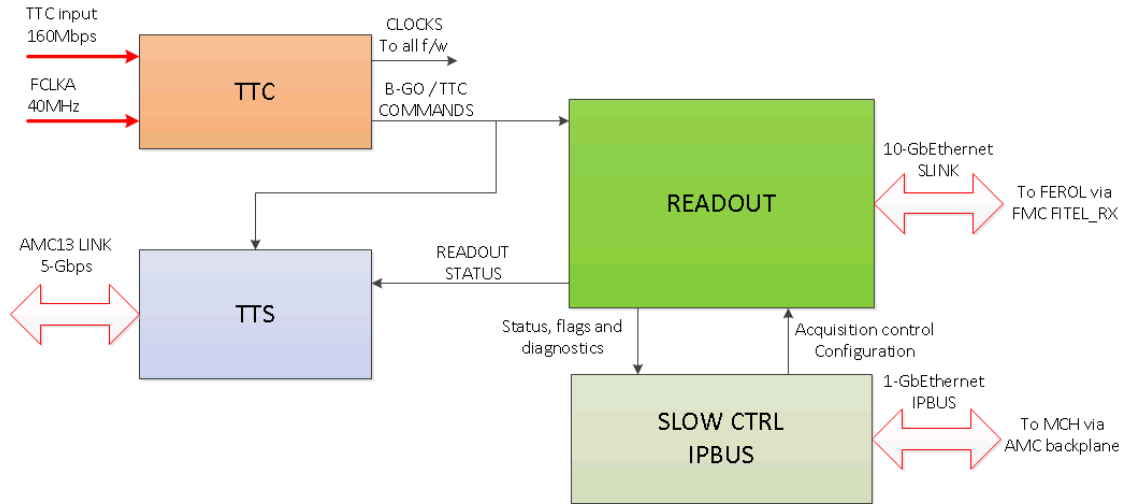


Figure 11: Block diagram of the BUILD FED firmware.

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

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

485 4.4.3 Pixel FED Data Payload

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

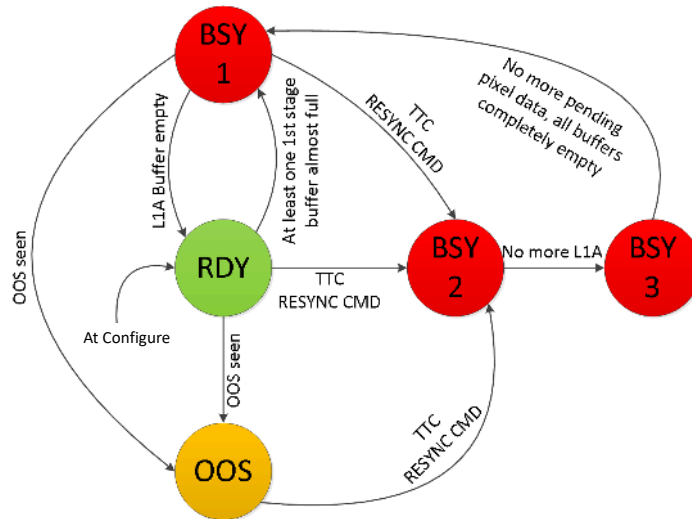


Figure 12: Block diagram of the FED TTS states and transitions. The BSY1, BSY2 and BSY3 are different FSM nodes which all have TTS state BSY.

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

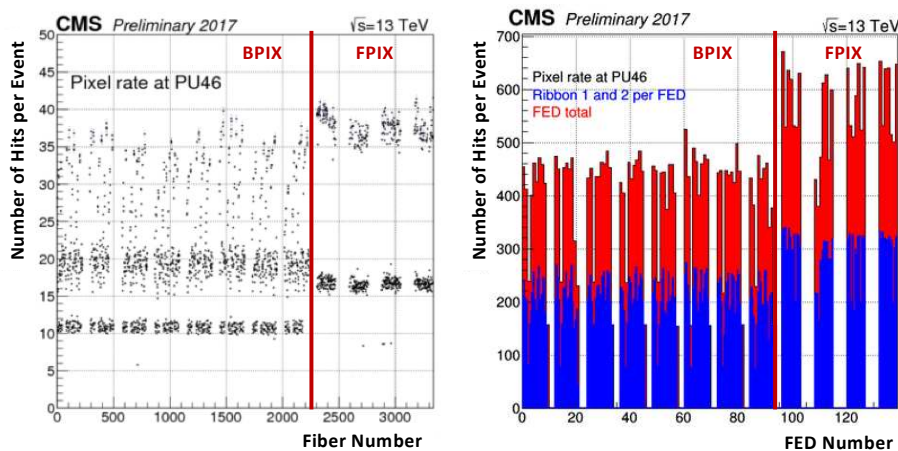


Figure 13: (Left) Data payload versus fiber number and (right) data payload versus FED number for a pileup of 46. For BPIX the data payload is 10 hits per event per fiber, and thus lowest, for fibers connected to layer 4 modules. Layer 2 and 3 modules have a data payload of 15-20 hits per event per fiber. 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).

496 5 System Tests in the CMS Detector - Phase-1 Pixel Pilot System

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

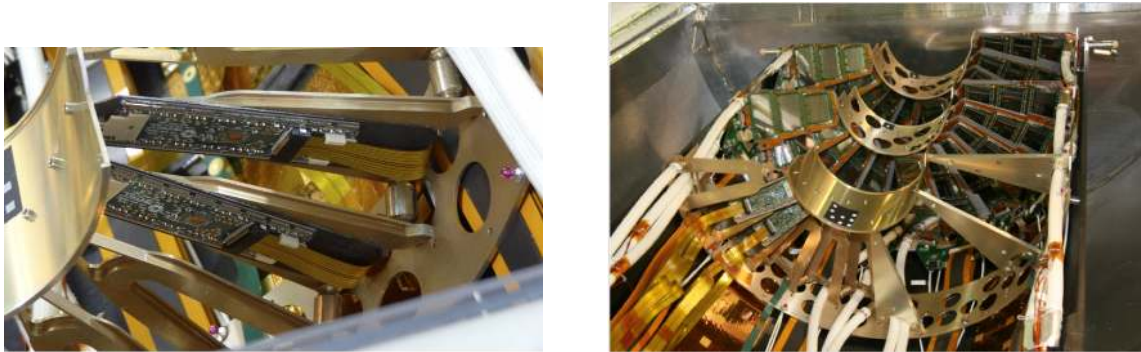


Figure 14: (Left) Two pilot sensor modules. (Right) Pixel half cylinder with pilot sensor modules installed on the third half disk, and pixel plaquettes of the original pixel detector installed on the first two half disks.

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

516 Six out of eight pilot sensor modules were successfully used in data taking. Pixels get clusterized
517 by the reconstruction software and the hit position is determined by their barycenter. Figure 16
518 (left) shows the measured cluster positions projected onto the transverse plane. Figure 16 (right)
519 shows the expected hits which are derived from extrapolated tracks that are reconstructed in the
520 FPIX detector. The pilot system was not part of the tracking since the pilot system modules are
521 located at the edge of the tracking coverage. This effect is visible in Fig.16 (right), where there are
522 no expected hits close to the center. In order to discard the fringes of ROCs, where the uncertainty
523 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.

524 shapes in Fig.16 (right).

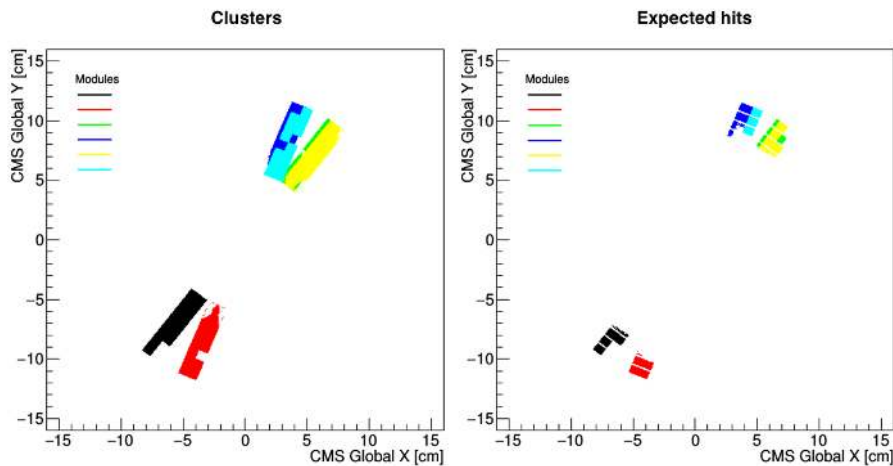


Figure 16: (Left) Cluster positions in each pilot sensor module used in data taking and (right) expected hits in the transverse plane in the CMS coordinate system. Color coding identifies individual sensor modules.

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

528 6 System Tests in the Laboratory

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

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

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

545 **6.1 FED Tester Setup**

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

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

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

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

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

570 The SRAM of the GLIBs is a software loadable memory that can be accessed by the event
571 readout framework. There are two separate memory locations that each hold approximately 8.4 MB
572 of data. The first SRAM is designated to hold the distributions of hits per ROC. The second is
573 designated to hold the emulated pixel locations for each hit. The reading of SRAM memory is
574 driven by a 160 MHz clock. Since each GLIB can emulate 16 independent channels, there are 32

575 different processes which must occur to emulate an event. The first SRAM only needs to be read
576 once per event, the second SRAM needs to be read any time a pixel hit information needs to be
577 sent. The SRAM readout can be done at trigger rates expected in CMS.

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

579 **6.2 The DAQ Setup for High Data Rates**

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

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

602 **7 Pixel Online Software**

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

606 **7.1 Hardware Access and Supervisors**

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

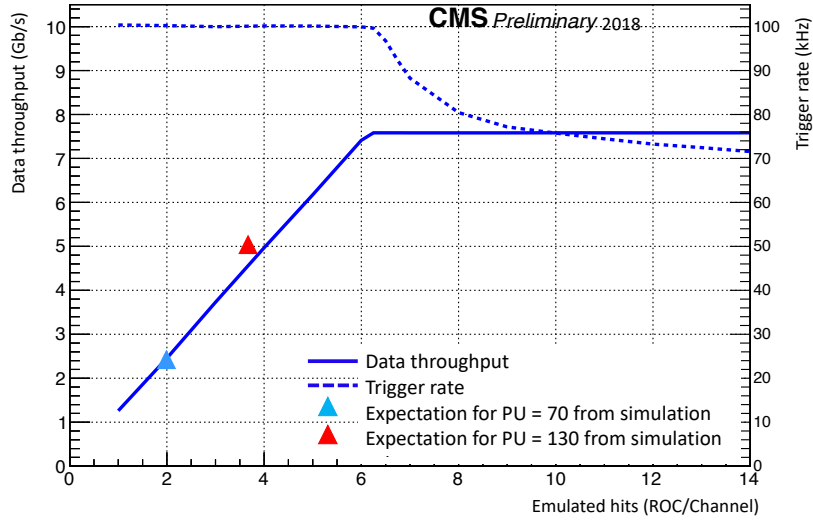


Figure 17: Data throughput (solid line and left y-axis) and trigger rate (dotted line and right y-axis), as measured when the system is driven by the FED tester. The FED can handle a trigger rate of 100 kHz for up to 6 emulated hits per ROC. The throttling of the trigger rate is caused by back-pressure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

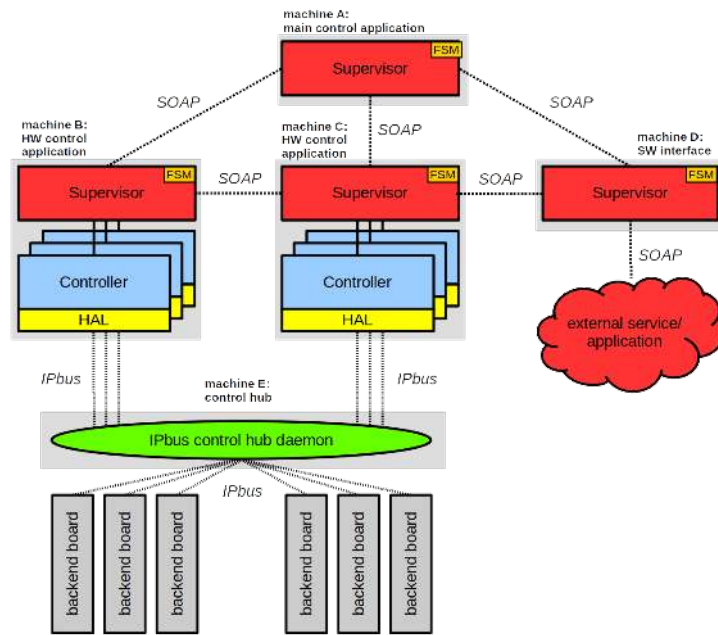


Figure 18: The hierarchical structure of the POS.

613 In order to prevent conflicts due to potential concurrent hardware accesses, the connection is
 614 established via a so-called control-hub, which is a service daemon running on a separate machine that

615 queues incoming requests from different applications and distributes them to the actual hardware.

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

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

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

633 7.2 Distributed Software Architecture

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

- 639 • 1 main supervisor (PixelSupervisor),
- 640 • 1 DCS service supervisor (PixelDCSFSMInterface),
- 641 • 1 AMC13 hardware supervisor (PixelAMC13Supervisor) controlling 12 AMC13 boards,
- 642 • 12 FED hardware supervisors (PixelFEDSupervisor) controlling 108 FEDs,
- 643 • 12 Pixel FEC hardware supervisors (PixelFECSupervisor) controlling 16 Pixel FECs,
- 644 • 3 Tracker FEC hardware supervisors (PixelTKFECSupervisor) controlling 3 Tracker FECs,
- 645 • 8 TCDS hardware supervisors (PixelTCDSSupervisor) controlling 8 TCDS boards.

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

650 **7.3 Interface to the Detector Control System**

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

660 **8 Operation Performance**

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

665 **8.1 Software Recovery Mechanisms**

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

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

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

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

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

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

711 8.2 Periodic ROC Resets

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

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

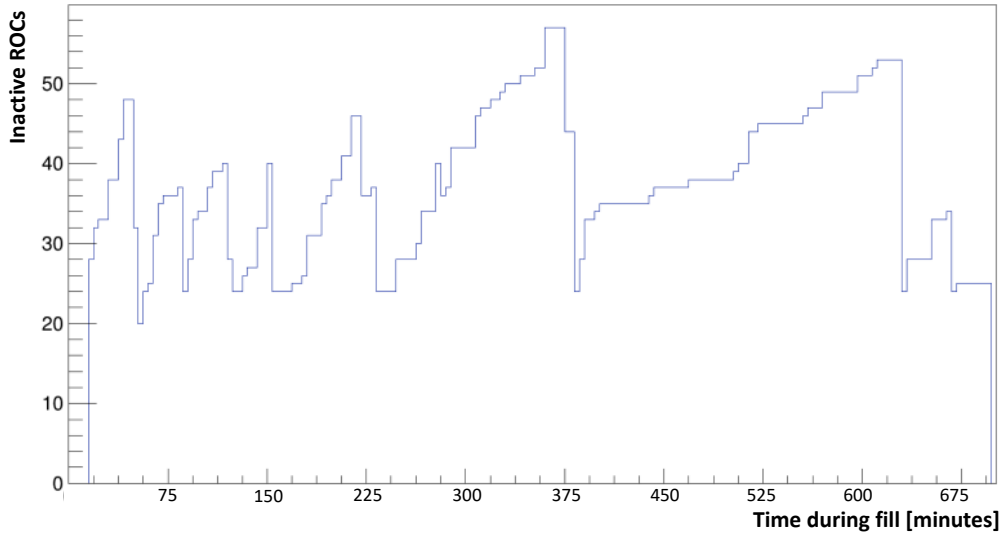


Figure 19: The number of inactive ROCs over time in BPIX layer 1 during a typical LHC fill in 2017. The number of inactive ROCs increases until a programmable threshold is reached, at which point the SEU recovery mechanism is activated and the ROCs are recovered. The SEU recovery mechanism can be activated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

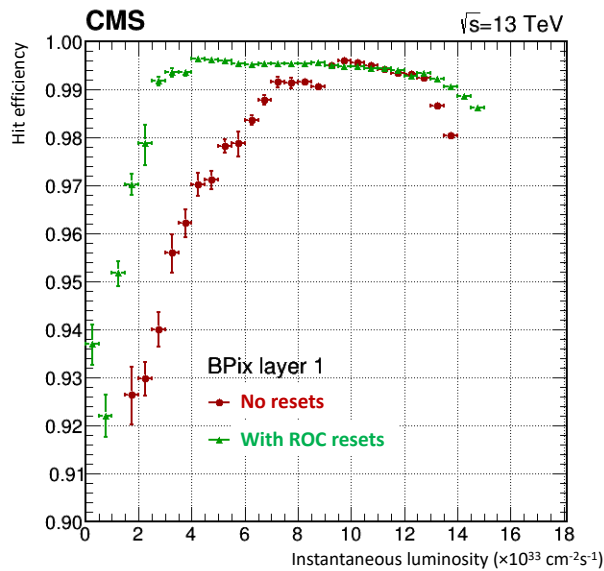


Figure 20: BPIX layer 1 hit efficiency with (green) and without (red) periodic ROC resets at 70 Hz versus instantaneous luminosity.

731 **9 Conclusion**

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

743 **A ROC**

744 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

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

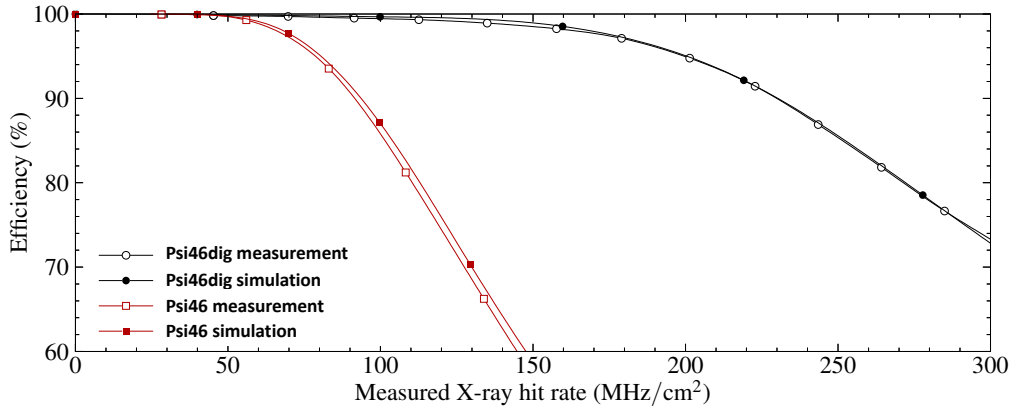


Figure 21: Measured and simulated efficiencies for PSI46 and PSI46dig ROCs as a function of X-ray hit rates.

748 **B TBM**

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

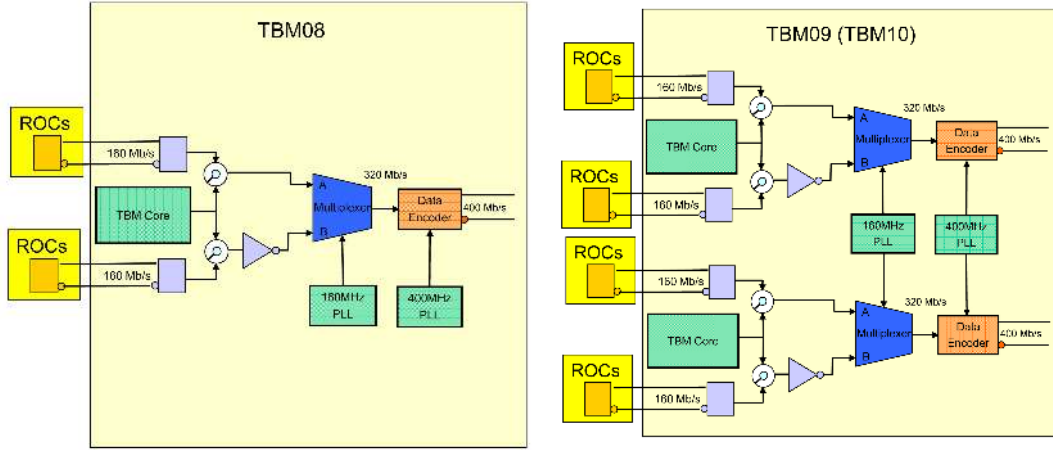
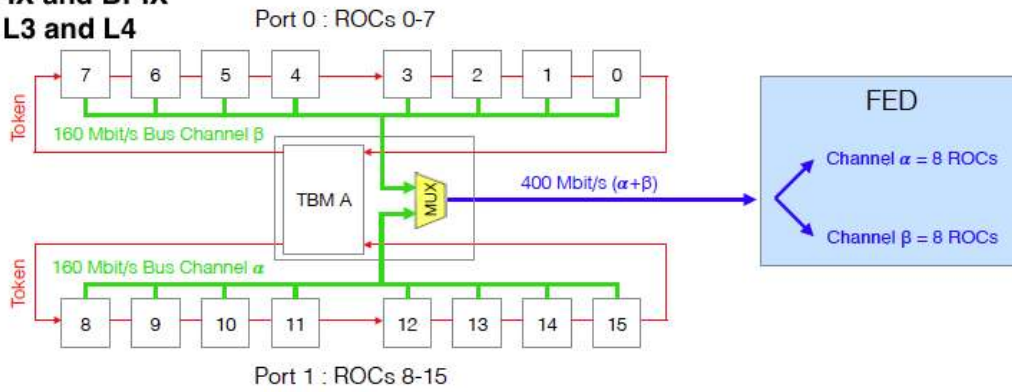


Figure 22: Block diagram of TBM08 (left) used in FPIX and BPIX layer 3 and 4, and TBM09/10 (right) used in BPIX layer 2 and 1, respectively.

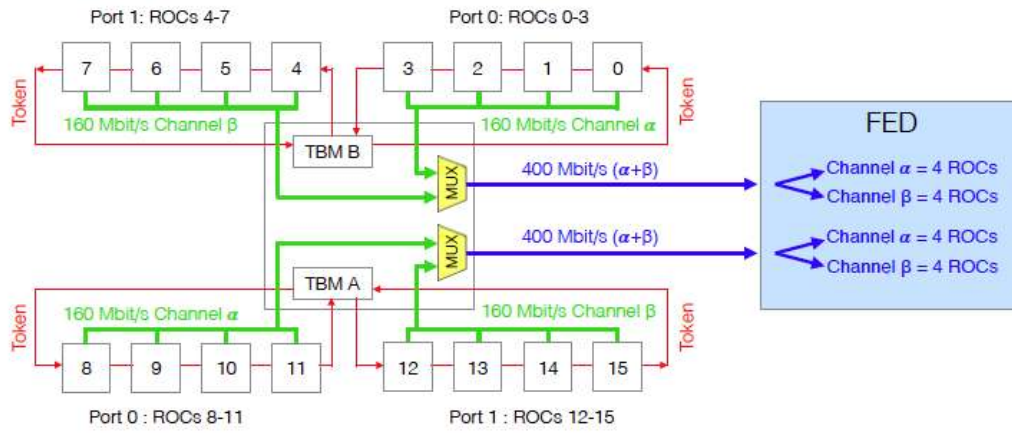
750 The most important new block in the Phase-1 pixel TBM is the DataKeeper. The functions of
 751 the DataKeeper span four stages. The first stage is data stream inversion. One of the two output
 752 data streams is inverted to identify it as one of two data streams, TBM_{α} or TBM_{β} . This is done
 753 to uniquely identify each data stream at the receiving end of the optical link. The second stage
 754 is bit interleaving and building of 4-bit words. Two 4-bit words are created as follows: Word A
 755 = (1st TBM_{α} bit, 1st TBM_{β} bit, 2nd TBM_{α} bit, 2nd TBM_{β} bit), Word B = (3rd TBM_{α} bit, 3rd
 756 TBM_{β} bit, 4th TBM_{α} bit, 4th TBM_{β} bit). Word A is the first being encoded, while Word B is the
 757 second to be encoded. Word B is used in deciding when a frame signal can be transmitted. The
 758 third stage is encoding and framing. Word A and B are encoded as a 5-bit symbol using a standard
 759 4b/5b encoding, shown in Tab. 3. The hex value 0xA is designated as a special case. There are
 760 two choices for the configuration used to represent 0xA. If Word A = 0xA, and if the Word B string
 761 begins with a '0', then 0xA is represented by '10110'. Otherwise, if the Word B string begins with
 762 a '1', then 0xA is represented by '10000'. The string '10000' forms a unique pattern in the data
 763 stream. This framing pattern allows the FED to identify the first bit in the final serial data stream.
 764 Stage four is NRZI encoding. NRZI is an encoding scheme for a serial data stream. In this system,
 765 a '1' is represented as a transition, from '1' to '0', or '0' to '1', depending on the previous value
 766 transmitted. A '0' is represented by the absence of a transition, i.e. the previous data bit is repeated.

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

**TBM08
FPIX and BPIX
L3 and L4**



**TBM09
BPIX L2**



**TBM10
BPIX L1**

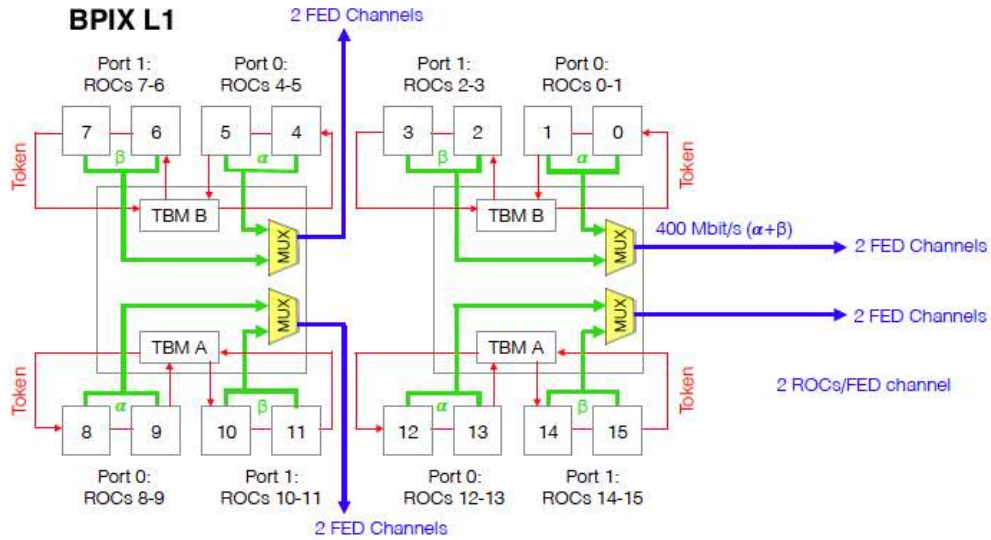


Figure 23: Readout scheme of the different TBMs used in the Phase-1 pixel detector.

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

771 C Rack Layout

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

ID	Units	Contents S1G01 - FPIX														RackWiz	ID	Units	Contents S1G03 - BPIX														RackWiz	ID	Units	Contents S1G04 - BPIX														RackWiz
		1	2	3	4	5	6	7	8	9	10	11	12	13	14				1	2	3	4	5	6	7	8	9	10	11	12	13	14				1	2	3	4	5	6	7	8	9	10	11	12	13	14	
56	4	Turbine															56	4	Turbine															56	4	Turbine														
55																	55																	55																
54																	54																	54																
53																	53																	53																
52	1	Heat Exchanger															52	1	Heat Exchanger															52	1	Heat Exchanger														
51																	51																	51																
50																	50																	50																
49	8	Pixel FEDs (PFED-1246 to PFED-1252)														S1G01-45	49	8	Pixel FEDs (PFED-1200 to PFED-1206)														S1G03-45	49	8	Pixel FEDs (PFED-1246 to PFED-1252)														S1G04-45
48																Near	48																Near Up	48																Near Up
47																Bpt	47																Bpt	47																Bpt
46																	46																	46																
45																	45																	45																
44		Cable Organizer															44		Cable Organizer															44		Cable Organizer														
43	1	Blank Panel															43	1	Blank Panel															43	1	Blank Panel														
42																	42																	42																
41																	41																	41																
40	8	Pixel FEDs (PFED-1306 to PFED-1312)														S1G01-36	40	8	Pixel FEDs (PFED-1210 to PFED-1216)														S1G03-36	40	8	Pixel FEDs (PFED-1306 to PFED-1312)														S1G04-36
39																+Z	39																+Z	39																+Z
38																Far	38																Far	38																Far
37																BpD	37																BpD	37																BpD
36																	36																	36																
35																	35																	35																
34	1	Cable Organizer															34	1	Cable Organizer															34	1	Cable Organizer														
33																	33																	33																
32																	32																	32																
31	8	Pixel FEDs (PFED-1320 to PFED-1326)														S1G01-27	31	8	Pixel FEDs (PFED-1224 to PFED-1230)														S1G03-27	31	8	Pixel FEDs (PFED-1320 to PFED-1326)														S1G04-27
30																-Z	30																-Z	30																-Z
29																Near	29																Near	29																Near
28																BmI	28																BmI	28																BmI
27																	27																	27																
26	1	Cable Organizer															26	1	Cable Organizer															26	1	Cable Organizer														
25																	25																	25																
24																	24																	24																
23	8	Pixel FEDs (PFED-1330 to PFED-1336)														S1G01-18	23	8	Pixel FEDs (PFED-1234 to PFED-1240)														S1G03-18	23	8	Pixel FEDs (PFED-1330 to PFED-1336)														S1G04-18
22																+Z	22																+Z	22																+Z
21																Far	21																Far	21																Far
20																BmD	20																BmD	20																BmD
19																	19																	19																
18																	18																	18																
17	1	Cable Organizer															17	1	Cable Organizer															17	1	Cable Organizer														
16																	16																	16																
15	2	Heat Exchanger															15	2	Heat Exchanger															15	2	Heat Exchanger														
14																	14																	14																
13	1	Air Deflector															13	1	Air Deflector															13	1	Air Deflector														
12																	12																	12																
11	1	Cable Organizer															11	1	Cable Organizer															11	1	Cable Organizer														
10	1	24x MPO patch panel															10	1	24x MPO patch panel															10	1	24x MPO patch panel														
9	1	Cable Organizer															9	1	Cable Organizer															9	1	Cable Organizer														
8	1	24x MPO patch panel															8	1	24x MPO patch panel															8	1	24x MPO patch panel														
7	1	Cable Organizer															7	1	Cable Organizer															7	1	Cable Organizer														
6	1	24x MPO patch panel															6	1	24x MPO patch panel															6	1	24x MPO patch panel														
5	1	Cable Organizer															5	1	Cable Organizer															5	1	Cable Organizer														
4	1	24x MPO patch panel															4	1	24x MPO patch panel															4	1	24x MPO patch panel														
3	3																3	3																3	3															
2																	2																	2																
1																	1																	1																

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.

775 **D POH**

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

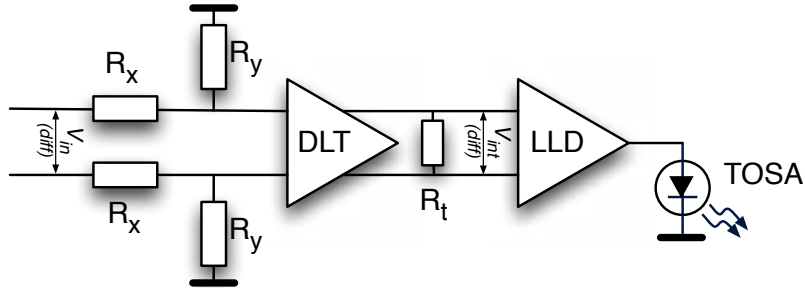


Figure 25: Block diagram of a single-channel of a POH.

777 The change in POH led also to a need to replace the first part of the optical fiber cabling plant
778 – the so-called Fanout in Fig. 3. Since it was not possible to exchange the multi-ribbon cables, the
779 optical connector interface at Patch Panel (PP)1 remained unchanged, being of type Multiple Fiber
780 System (MFS). PP0 is located at the tracker bulkhead, PP1 is located at the inner bore of the cryostat
781 and PP2 is located in the counting room below the pixel detector racks. Given that the number of
782 12-fiber ribbons needed to read out and control the upgraded detector increased by almost 30% with
783 respect to the number previously installed, it was necessary to install additional optical fiber cables
784 between PP0 and PP1. Unused spare cables were utilized between PP1 and the pixel back-end. This
785 brought the opportunity to change the optical connector type used at PP0, from the MFS connector
786 used previously to the Multi-fiber Push On (MPO) (or MTP) type. All the fibers were changed to
787 MPO type between PP0 and PP1. MPO connectors do not require the use of a tool to (dis-)connect,
788 making the installation and removal time shorter, which is important to maintain the serviceability
789 of the detector in a radiation environment.

790 **E Tracker FEC**

791 **E.1 Functionality**

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

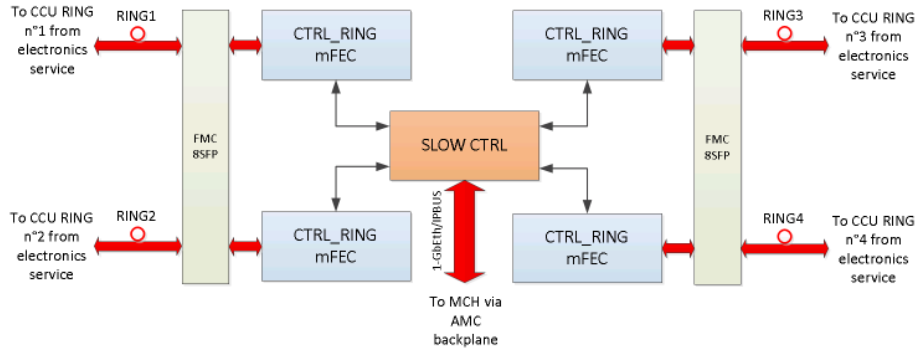


Figure 26: The block diagram of the Phase-1 Pixel Tracker FEC.

793 **E.2 Redundancy**

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

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

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

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

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

810 For instance:

- 811 • Procedure to follow to bypass the DOH A:
 - 812 – Command to CCU1: switch CCU1 input (from IA to IB).
 - 813 – Command to CTRL_RING : switch the rings (from ring A to B).
 - 814 – Command to CCU4: switch CCU4 output (from OA to OB).
- 815 • Procedure to follow to bypass CCU2:
 - 816 – Command to CCU1: switch CCU1 output (from OA to OB).
 - 817 – Command to CCU3: switch CCU3 input (from IA to IB).

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

820 **E.3 The register write/read commands**

821 For each command a handshaking operation is done for each transaction by forwarding the ac-
822 knowledge (ACK) of the transaction being executed. For each command type, the command can be
823 decomposed as:

- 824 • Register write command:
 - 825 – Transaction T1 from FMC to CCU via the TX line of the FMC.
 - 826 – ACK(T1) from CCU to FMC via the RX line of the FMC.
- 827 • Register read command:
 - 828 – Transaction T1 from FMC to CCU via the TX line of the FMC.
 - 829 – ACK(T1) from CCU to FMC via the RX line of the FMC.
 - 830 – Transaction T2 from CCU to FMC via the RX line of the FMC:
 - 831 * T2 contains the register data to read.
 - 832 – ACK(T2) from FMC to CCU via the TX line of the FMC.

833 **E.4 Data and end of frame formats**

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

836 The two nodes that communicate with each other are determined by the destination address
837 (DEST) and SOURCE fields. They contain an address value coded in 8-bit. Each node has its own
838 address. The address of the master node (CTRL_RING) is set to 0x00. The address of each slave
839 node (CCU) is pre-defined and set to a value between 0x01 and 0x7F. 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

840 allows broadcasting the commands to all CCUs at the same time. A 16-bit wide cyclic redundancy
 841 check (CRC16) is computed from DEST to DATA (including SOURCE and LENGTH), allowing
 842 protection of the data frame transfer. The data frame is terminated by 2 bytes marking the end of
 843 frame. The end of the frame is composed of 4 symbols, each one coded in 4-bits, as explained in
 844 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

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

850 E.5 CTRL_RING firmware architecture

851 Figure 27 describes the CTRL_RING firmware architecture.

852 It consists of five main parts:

- 853 • TX Block, which is responsible for:
 - 854 – transmission of the data frames (commands) from TRA FIFO (or RET FIFO if not
855 empty),
 - 856 – CRC16 computing,
 - 857 – 4b/5b encoding,
 - 858 – NRZI encoding and serialization,
 - 859 – Ring multiplexing in case of a switch request.
- 860 • RX Block, which is responsible for:

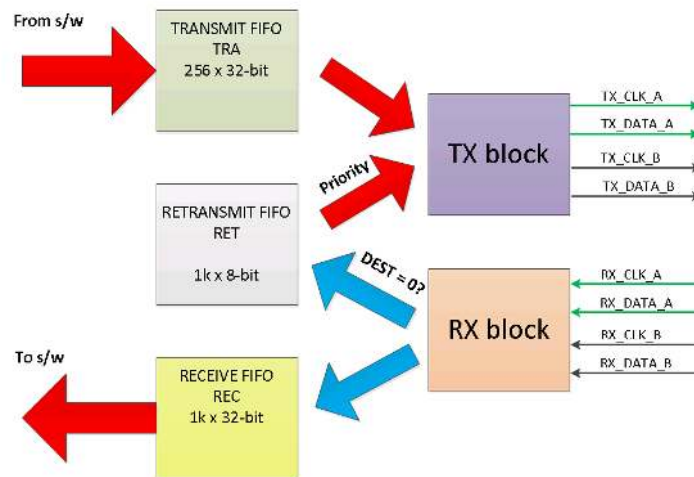


Figure 27: CTRL_RING firmware architecture.

- 861 – ring multiplexing in case of a switch request,
 862 – deserialization, NRZI decoding and alignment with the injected token frame,
 863 – 4b/5b encoding,
 864 – reception of the data frames and storage in REC FIFO (RET FIFO too, if DEST = 0),
 865 – CRC16 checking and status updating.
- 866 • TRA FIFO:
 - 867 – Stacks the commands to be executed out of the ring.
 - 868 – Writable by software.
 - 869 • REC FIFO:
 - 870 – Stores the data coming back from the ring.
 - 871 – Readable by software.
 - 872 • RET FIFO:
 - 873 – Stores the data coming back from the ring (if DEST = 0).
 - 874 – The data from the RET FIFO has to be returned to the ring for ACK.

875 **F Pixel FED**

876 **F.1 DECODE FED Firmware**

877 The TBM combines two readout channels and uses an encoding scheme with only 17 valid 5-bit
878 symbols, including a special symbol for frame finding, which enables the possibility to search for
879 symbol errors, and to find the correct sampling point. A Xilinx FPGA Idelay2, which is a 31-tap,
880 wrap-around, delay primitive with a calibrated resolution of about 80 ps, is used to shift the copy
881 of the incoming serial data stream and check each tap position for symbol errors. This results in a
882 32-bit word where a ‘1’ stands for a delay tap setting with symbol error and a ‘0’ for a delay tap
883 setting where no symbol error was found. In order to generate the bit pattern, symbol errors for
884 each delay tap position are accumulated within a time window of 13.1 ms and the measurement is
885 repeated continuously every 1.7 s. In addition a ‘Phase Finding Now’ mode was implemented to
886 speed up TBM PLL scans, where the phase relationship between the incoming serial data stream
887 and the FED system clock is changed frequently. ‘Phase Finding Now’ is initiated by an IPBus
888 command and uses 6.5 ms integration time per delay tap setting.

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

891 Here is a readout example: 111111000000000000000000000000111. The window that contains
892 zeros is identified and the mid-point taken as the sampling point. There is also a manual phase
893 setting possibility by setting the delay tap parameters over IPBus. For monitoring purposes a symbol
894 histogram is implemented to check for symbol distribution and input link saturation by calculating
895 the ratio between idle and other symbols.

896 The TBM sends a special symbol for frame finding by assigning two patterns to Hex value
897 0xA. If 0xA in the serial data stream is followed by a ‘1’, it is translated to ‘10000’ which forms
898 either ‘11111’ or ‘00000’ after NRZI encoding. This unique pattern is used for resetting the frame
899 counter and aligning the FED to the incoming data packets.

900 The DECODE firmware block performs the reverse functionality of the TBM DataKeeper in
901 four steps. NRZI decoding is followed by the 5b/4b conversion, which results in a 4-bit data word.
902 Due to the bit interleaving structure of the TBM cores two of these 4-bit words hold the readout
903 channel TBM_{α} and readout channel TBM_{β} information, as follows: Word A = (1st TBM_{α} Bit, 1st
904 TBM_{β} Bit, 2nd TBM_{α} Bit, 2nd TBM_{β} Bit) and Word B = (3rd TBM_{α} Bit, 3rd TBM_{β} Bit, 4th
905 TBM_{α} Bit, 4th TBM_{β} Bit).

906 **F.2 BUILD FED Firmware**

907 **F.2.1 TTC Firmware Block**

908 The TTC firmware block, shown in Fig. 28, has to handle receiving, re-generating and propagating
909 the BX clock (and its derivatives) to the rest of the firmware. The sub-block ‘CLOCK GEN’
910 has the role of generating all the user clocks. The TTC firmware block also needs to handle
911 receiving, decoding and propagating of the TTC signal to the whole firmware. The sub-block ‘TTC
912 DECODER’ has the responsibility to receive and decode the encoded TTC signal serialized at
913 160 Mb/s. In order to ensure an efficient transmission of the TTC signal, the TCDS system uses an
914 encoding scheme via a Time Division Multiplexing (TDM) and a Bi-Phase Mark (BPM) encoding

915 to encode two channels (A and B) onto one channel. The channel A is dedicated to transmit L1A
 916 triggers, a 1-bit decision being sent on every BX. The channel B is suited to transmit general or user
 917 commands that are needed to control the acquisition. The TTC decoder part of the firmware handles
 918 the de-interleaving of the two channels and their decoding before delivering the L1A triggers. The
 919 EC0, the BX counter reset (BC0) and the RESYNC are the TTC commands mainly used in the
 920 firmware.

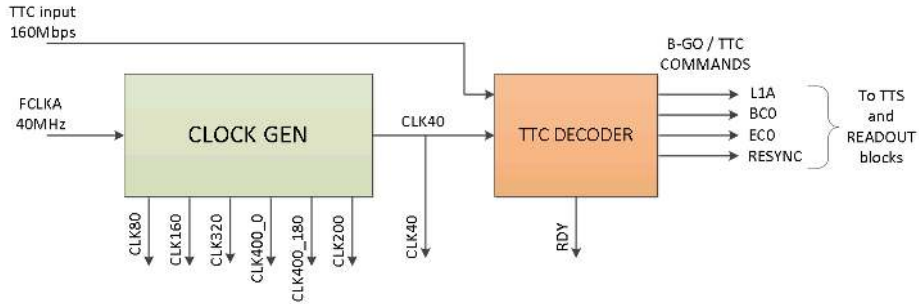


Figure 28: Block diagram of the FED TTC Block.

921 F.2.2 READOUT Firmware Block

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

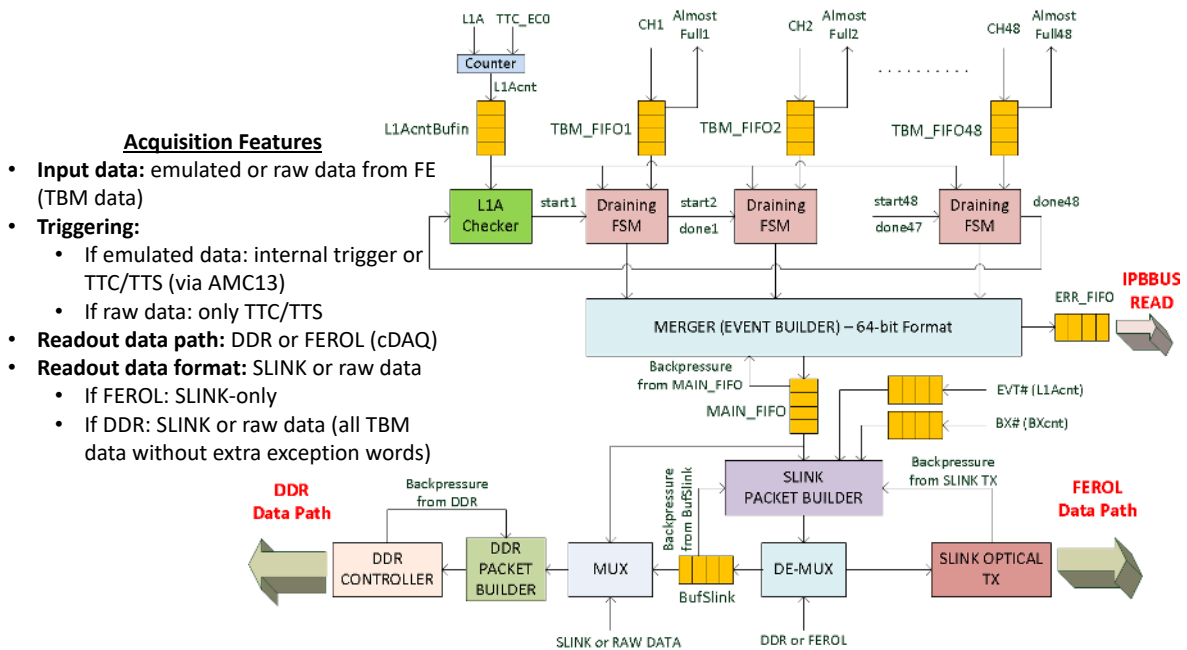


Figure 29: Block diagram of the FED READOUT firmware block.

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

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

927 The block receives data from 48 channels (CH1 to CH48), handled and transmitted by the
 928 DECODE FED firmware. A strobe signal is added to each channel data, which allows to directly
 929 buffer the data. An individual FIFO is dedicated to each channel. The data are in 36-bit format and
 930 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

931 The ID field marks the type of 32-bit data contained in the remaining channel data word. They
 932 are listed in Tab. 7. The first four ID data are sent by the TBM and retransmitted by the DECODE
 933 firmware. The last one is computed by the DECODE firmware and appears in case of hit overflow.
 934 The ROC data from DECODE to BUILD firmware is removed in case of overflow to reduce the
 935 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

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

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

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

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

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

949 The error words appearing in the readout data flow are:

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

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

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

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

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

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

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

980 The data payload corresponds to the entire pixel event stored in the MAIN FIFO. The size of
981 this part is variable and depends on the number of hits and errors seen by the firmware. The BX ID
982 (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: $N \times 64$ -bit words) Data from the MAIN FIFO
S-LINK TRAILER (64-bit word) (major info: EVT LGTH, CRC)

983 and the LV1 ID (event number being coded on 24-bit) are comprised into the SLINK HEADER
 984 field. The EVT LGTH (the event length), or the event packet size, is the number of 64-bit words
 985 forming the event packet by considering the header, the data payload and the trailer. Its value, added
 986 in the SLINK TRAILER field, is computed in realtime for each event read from the MAIN FIFO.
 987 Its minimal value is 2 (no hits and no errors words). A CRC generator through the CRC polynomial
 988 $P(x) = x^{16} + x^{15} + x^2 + 1$ is implemented in the firmware. The CRC is executed for each packet
 989 and its computed value is added to the SLINK TRAILER field. The CRC is the way for the CMS
 990 Central DAQ to verify the correctness of the received data and the packet transmission.

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

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

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

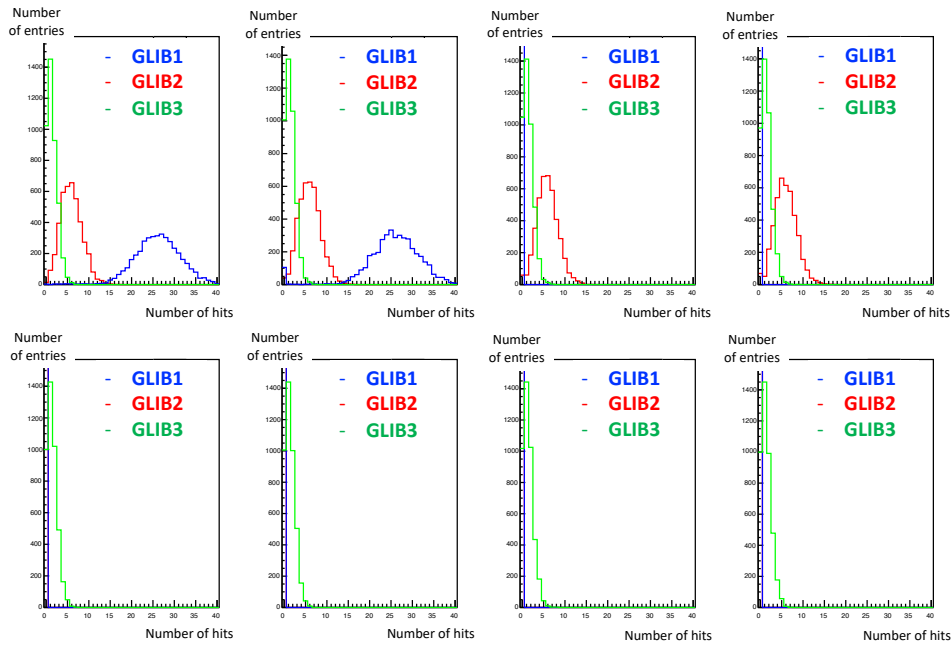


Figure 30: A Poisson hit distribution of events loaded into the SRAM of the FED tester. The distributions can be independent for each board. The number of ROCs in TBM channels can also be changed depending on the emulated layer. The green line is a GLIB emulating a BPIX layer-4 module with a poisson 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.

997 **H Operation Performance**

998 Towards the end of 2017 the DC-DC converters started to fail. All converters were extracted and
999 replaced in the year-end technical stop from December 2017 to February 2018. In May 2018 the
1000 source of the DC-DC problem was discovered and an operational solution was implemented. The
1001 problematic state of the DC-DC converter ASIC can be circumvented if their output is not disabled.
1002 This means that the higher DC-DC granularity will only be usable after the next pixel detector
1003 extraction in 2019, when all DC-DC converters will be exchanged again, using an improved version
1004 of the DC-DC converter ASIC. For the recovery chain this requires either the Tracker FEC to control
1005 the DC-DC converters, or an interface to the DCS to control the power supplies.

1006 In the case that the DC-DC converter disabled, the report chain of the problem from FED to
1007 PixelSupervisor is the same as for the usual SEU, but from the PixelSupervisor the PixelTkFEC-
1008 Supervisor is called first to switch off and on the DC-DC converters. Once this has finished the
1009 PixelFECSupervisor is signaled to reprogram the sensor modules. The observable behavior of a
1010 non-responsive TBM is the same as that of a recoverable SEU, hence the distinction can only be
1011 made by first attempting the standard SEU recovery, and if the problem persists, switch off and on
1012 the DC-DC converter.

1013 In the case that the DC-DC converters should not be disabled the DCS steps into the place of the
1014 Tracker FEC. ‘Due to the high currents at turn on’ it was decided not to recover the non-responsive
1015 TBMs during stable beams.

Acknowledgments

1017 The tracker groups gratefully acknowledge financial support from the following funding agen-
1018 cies: BMWFW and FWF (Austria); FNRS and FWO (Belgium); CERN; MSE and CSF (Croatia);
1019 Academy of Finland, MEC, and HIP (Finland); CEA and CNRS/IN2P3 (France); BMBF, DFG,
1020 and HGF (Germany); GSRT (Greece); OTKA and NIH (Hungary); DAE and DST (India); IPM
1021 (Iran); INFN (Italy); PAEC (Pakistan); SEIDI, CPAN, PCTI and FEDER (Spain); Swiss Funding
1022 Agencies (Switzerland); MST (Taipei); STFC (United Kingdom); DOE and NSF (U.S.A.).

1023

1024 Individuals have received support from HFRI (Greece).

References

- 1025
- 1026 [1] CMS Collaboration, *The CMS experiment at the CERN LHC*, 2008 JINST 3 S08004,
1027 doi:10.1088/1748-0221/3/08/S08004.
- 1028 [2] CMS Collaboration, *The CMS tracker system project : Technical Design Report*,
1029 CERN-LHCC-98-006, CMS-TDR-5.
- 1030 [3] CMS Collaboration, *CMS Technical Design Report for the Pixel Detector Upgrade*,
1031 CERN-LHCC-2012-016, CMS-TDR-11.
- 1032 [4] H.-C. Kaestli, *Frontend electronics development for the CMS pixel detector upgrade*, Nucl. Instrum.
1033 Meth. A 731 (2013) 88.
- 1034 [5] R. Stringer, *A digital readout system for the CMS Phase-1 pixel upgrade*, 2015 JINST 10 C04037.
- 1035 [6] J. Hegeman et al., *The CMS Timing and Control Distribution System*, 2015 IEEE Nuclear Science
1036 Symposium and Medical Imaging Conference (NSS/MIC), San Diego, CA, 2015, pp. 1-3.
- 1037 [7] E. Hazen, A. Heister, C. Hill, J. Rohlf, S.X. Wu and D. Zou, *The AMC13XG: a new generation clock /*
1038 *timing / DAQ module for CMS microTCA*, 2013 JINST 8 C12036.
- 1039 [8] J. Varela, *Timing and Synchronization in the LHC Experiments*, 6th Workshop on Electronics for
1040 LHC Experiments, Krakow, 2000, CERN-2000-010.
- 1041 [9] L. Cadamuro, *The CMS Level-1 trigger system for LHC Run 2*, 2017 JINST 12 C03021.
- 1042 [10] D. Gigi et al., *The FEROL40, a microTCA card interfacing custom point-to-point links and standard*
1043 *TCP/IP*, PoS TWEPP 2017 (313) <https://doi.org/10.22323/1.313.0075>
- 1044 [11] S. Baron, *Timing, Trigger and Control (TTC) systems for the LHC*, CERN, 2 October 2014; Online:
1045 <http://ttc.web.cern.ch/TTC/intro.html>.
- 1046 [12] P. Placidi, A. Marchioro and P. Moreira, *CMS Tracker PLL Reference Manual*, CERN (2000),
1047 <http://cds.cern.ch/record/1069705>.
- 1048 [13] P. Moreira and A. Marchioro, *QPLL - a quartz crystal based PLL for jitter filtering applications in*
1049 *LHC*, 9th Workshop on Electronics for LHC Experiments, Amsterdam (2003).
- 1050 [14] H. Furtado et al., *Delay25, an ASIC for timing adjustment in LHC*, 11th Workshop on Electronics for
1051 LHC and future Experiments, Heidelberg, 2005.
- 1052 [15] L. Feld, W. Karpinski, K. Klein, M. Lipinski, M. Preuten, M. Rauch, St. Schmitz and M. Wlochal,
1053 *The DC-DC conversion power system of the CMS Phase-1 pixel upgrade*, 2015 JINST 10 C01052.
- 1054 [16] C. Paillard, C. Ljuslin and A. Marchioro, *The CCU25: A network oriented communication and*
1055 *control unit integrated circuit in a 0.25 μm CMOS technology*, CERN 2002-003.
- 1056 [17] H.-C. Kaestli, *Design and performance of the CMS pixel detector readout chip*, Nucl. Instrum. Meth.
1057 A 565 (2006) 188.
- 1058 [18] J. Hoss, *Search for Supersymmetry with Multiple Charged Leptons at $\sqrt{s} = 13 \text{ TeV}$ with CMS and*
1059 *Radiation Tolerance of the Readout Chip for the Phase I Upgrade of the Pixel Detector*, PhD Thesis,
1060 ETH Zurich (2017), No. 24286, <https://doi.org/10.3929/ethz-b-000182698>.
- 1061 [19] M. Rossini, *Module Prototype Qualification for the CMS Pixel Detector Upgrade*, PhD Thesis, ETH
1062 Zurich (2015), No. 22934, <https://doi.org/10.3929/ethz-a-010594693>.
- 1063 [20] E. Bartz, *The 0.25 μm Token Bit Manager Chip for the CMS Pixel Readout*,
1064 <https://cds.cern.ch/record/720634>.

- 1065 [21] E. Bartz, *CMS-doc-12626-v1: TBM08c Documentation*.
- 1066 [22] M. Pesaresi et al., *The FC7 AMC for generic DAQ and control applications in CMS*, 2015 JINST 10
1067 C03036.
- 1068 [23] G. Auzinger, *Deployment of the CMS Tracker AMC as backend for the CMS pixel detector*, 2016
1069 JINST 11 C01056.
- 1070 [24] *Kintex-7 FPGAs Data Sheet*, XILINX DS182.
- 1071 [25] J. Troska, F. Vasey et al., *The versatile link, a common project for super-LHC*, 2009 JINST 4 P12003.
- 1072 [26] G. Cervelli, A. Marchioro, P. Moreira and F. Vasey, *A linear laser-driver array for optical*
1073 *transmission in the LHC experiments*, 2000 IEEE Nuclear Science Symposium Conference Record,
1074 9/145-9/149 vol.2.
- 1075 [27] T. Uemura, Y. Ishikawa, Y. Nekado, A. Izawa, M. Yoshihara and H. Nasu, *1060-nm 10-Gb/s **
1076 *12-channel parallel-optical modules for optical interconnects*, 2010 IEEE CPMT Symposium Japan,
1077 Tokyo, 2010, pp. 1-4.
- 1078 [28] J. Troska, G. Cervelli, F. Faccio, K. Gill, R. Grabit, R. M. Jareno, A. M. Sandvik and F. Vasey,
1079 *Optical readout and control systems for the CMS tracker*, 2003 IEEE Transactions on Nuclear
1080 Science, Volume 50, Issue 4, pp 1067-1072.
- 1081 [29] K. Kloukinas, W. Bialas, F. Drouhin, C. Ljuslin, A. Marchioro, E. Murer, C. Paillard and E. Vlasov,
1082 *FEC-CCS : A common Front-End Controller card for the CMS detector electronics*, CERN 2007-001,
1083 <https://cds.cern.ch/record/1027434>.
- 1084 [30] C. Ghabrous Larrea, K. Harder, D. Newbold, D. Sankey, A. Rose, A. Thea and T. Williams, *IPbus: a*
1085 *flexible Ethernet-based control system for xTCA hardware*, 2015 JINST 2 C02019.
- 1086 [31] <https://www.xilinx.com/products/design-tools/vivado.html>
- 1087 [32] B. Akgün, *Pilot system for the Phase-1 pixel upgrade of CMS*, PoS VERTEX2015 (2015) 018.
- 1088 [33] B. Vormwald, *Commissioning of the Phase-1 upgrade of the CMS pixel detector*, Springer Proc. Phys.
1089 213 (2018) 385-389.
- 1090 [34] J. Ngadiuba, *Testing and Integration of the Service Cylinders for the CMS Phase-1 Pixel*, Proceedings
1091 of TWEPP 2016, Karlsruhe, Germany, CMS-CR-2016/357.
- 1092 [35] B. Akgün, *Integration and Testing of the DAQ System for the CMS Phase-1 Pixel Upgrade*, 2017
1093 JINST 12 C02078.
- 1094 [36] M. Barros Marin, S. Baron, V. Bobillier, M. Di Cosmo, S. Haas, M. Hansen, M. Joos, F. Vasey and P.
1095 Vichoudis, *A GLIB-based uTCA demonstration system for HEP experiments*, 2013 JINST 8 C12011.
- 1096 [37] V. Brigljevic et al., *Using XDAQ in application scenarios of the CMS experiment*,
1097 FERMILAB-CONF-03-293, CHEP-2003-MOGT008 (2003).
- 1098 [38] T. Goodale et al., *The Cactus Framework and Toolkit: Design and Applications*, High Performance
1099 Computing for Computational Science (VECPAR 2002), 5th International Conference, Porto,
1100 Portugal, 2002.
- 1101 [39] D. Box et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, 2000.