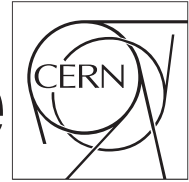




The Compact Muon Solenoid Experiment

CMS Draft Note



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

2021/11/19

Archive Hash: 4f9fbb1-D

Archive Date: 2021/11/19

Trigger data formats for the CSC and GEM systems for Run-3

Paul Padley¹, Mike Matveev¹, Sven Dildick^{*1}, Alexei Safonov², Jason Gilmore², Evaldas Juska^{*2}, Tao Huang², Jay Hauser³, Andrew Peck³, Will Nash³, Siddharth Hariprakash³, Darin Acosta⁴, Alexander Madorsky⁴, Jia Fu Low⁴, Andrew Brinkerhoff⁵, Gilles De Lentdecker⁶, and Laurent Pétre⁶

¹ Rice University (US)

² Texas A&M University (US)

³ University of California Los Angeles (US)

⁴ University of Florida (US)

⁵ Baylor University (US)

⁶ Université Libre de Bruxelles (BE)

*Primary author

Abstract

This detector note describes the trigger data formats for the GEM and CSC systems for Run-3. Changes to the Anode Local Charged Track board, Trigger Motherboard and the Muon Port Card data format are proposed to incorporate the GE1/1-ME1/1 integrated local trigger and to introduce options for high-multiplicity triggering on exotic signatures. The trigger data formats for the GE2/1-ME2/1 integrated trigger for Phase-2 of CMS are discussed as well.

This box is only visible in draft mode. Please make sure the values below make sense.

PDFAuthor: Sven Dildick, Evaldas Juska
PDFTitle: Trigger data formats for the CSC and GEM systems for Run-3
PDFSubject: CMS
PDFKeywords: CMS, GEM, CSC, trigger, data format

Please also verify that the abstract does not use any user defined symbols

Contents

1			
2	1	Introduction	2
3	2	CSC and GEM detectors in Run-3	3
4	2.1	Cathode Strip Chambers	3
5	2.2	Gas Electron Multipliers	4
6	2.2.1	GE1/1	4
7	2.2.2	GE2/1 and ME0	4
8	3	CSC Trigger Data Formats	6
9	3.1	ALCT to (O)TMB	6
10	3.2	CLCT in (O)TMB	7
11	3.3	(O)TMB to MPC	7
12	3.3.1	Inputs for the (O)TMB Correlation Algorithm	7
13	3.3.2	Quick Overview of the Trigger Data format	7
14	3.3.3	Common Redefinitions	7
15	3.3.4	Specific Redefinition: TMBs which do not have the CCLUT algorithm enabled	10
16	3.3.5	Specific Redefinition: ME1/1 Quality	10
17	3.3.6	Specific Redefinition: non-ME1/1 Quality	10
18	3.3.7	Map of 4+1 bit Run-3 Bending onto 4-bit Run-2 Pattern	11
19	3.3.8	Pattern ID Lookup	11
20	3.4	MPC to Muon Track Finder Sector Processor	13
21			
22	4	GEM Trigger Data Formats	21
23	4.1	Cluster Format	21
24	4.1.1	Version 1	21
25	4.1.2	Version 2	21
26	4.1.3	Version 3	21
27	4.2	OH to GEM Backend and to OTMB Link Format	23
28	4.2.1	Version 1	23
29	4.2.2	Version 2	23
30	4.2.3	Version 3	23
31	4.3	OptoHybrid Metadata	26
32	4.4	GEM Status bits	27
33	4.5	GEM Backend to EMTF Link Format	28
34	4.5.1	Metadata mode (format 1)	28
35	4.5.2	Normal mode (format 0)	28
36	5	Implementation of the Data Formats in CMSSW	30
37	5.1	CSCALCTDigi	30
38	5.2	CSCCLCTDigi	30
39	5.3	CSCCorrelatedLCTDigi	32
40	5.4	GEMPadDigi	33
41	5.5	GEMPadDigiCluster	33
42	6	Summary	35

1 Introduction

New Gas Electron Multiplier (GEM) chambers will be installed in the forward region $1.6 < |\eta| < 2.1$ [1]. The GEM station in front of ME1/1 is called GE1/1, and is shown in Fig. 1. The installation of GEM detectors will allow for a precision measurement of the GE1/1-ME1/1 bending angle in the trigger. This enhances the sensitivity of the trigger to soft prompt muon signatures and displaced muon signatures from new physics processes with long-lived particles. The added detecting layers will reduce the trigger rate, increase the trigger efficiency and also improve the operational resilience of the system. More information on the GEM upgrade can be found in [1]. In addition, new possibilities to trigger on exotic signatures from long-lived particles with more than 2 trigger primitives per Cathode Strip Chamber and per BX is being researched.

This detector note describes the trigger data formats for the GE1/1 and CSC systems for Run-3. Modifications to the trigger data formats in the ALCT, TMB and MPC electronic boards to include the GE1/1-ME1/1 integrated local trigger and options for high-multiplicity triggering on exotic signatures.

Section 2 describe the CSC and GEM detectors in Run-3. The CSC and GEM trigger data formats are given in Sec. 3 and Sec. 4. For completeness, the GE2/1 trigger data formats are provided as well. A summary is provided in section 6.

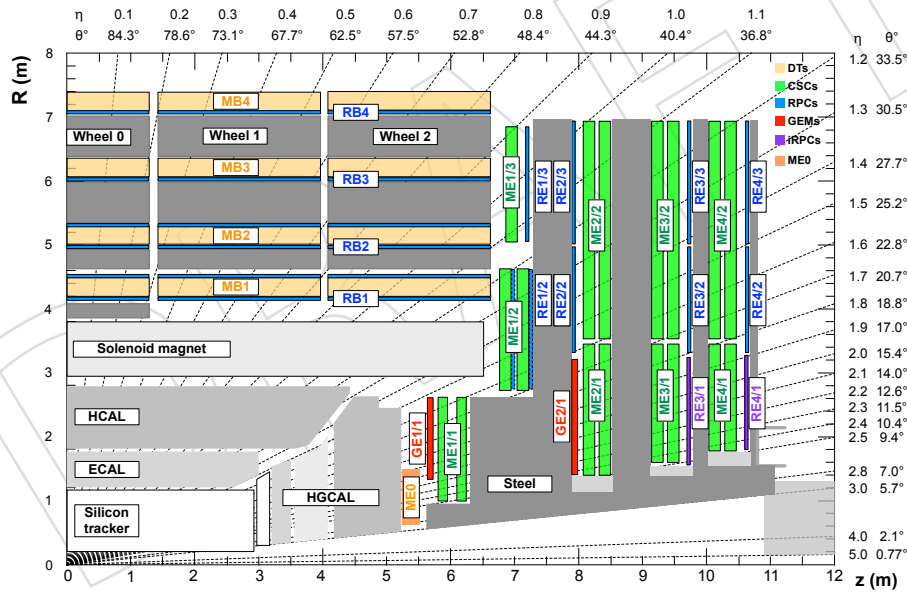


Figure 1: Quadrant of the CMS detector in the Run-3 configuration. The central component is the solenoid that surrounds the silicon tracker, the electromagnetic calorimeter (ECAL) and hadronic calorimeter (HCAL). Outside the solenoid sits the muon system. Drift tubes (DT), cathode strip chambers (CSC) and resistive plate chambers (RPC) are colored orange, green and blue respectively. New GE1/1 and GE2/1 detectors are shown in red, and ME0 in orange.

2 CSC and GEM detectors in Run-3

2.1 Cathode Strip Chambers

The Cathode Strip Chamber (CSC) system consists of 540 chambers organized in four stations called ME1 to ME4 and covering pseudorapidity from 0.9 up to 2.4. Each CSC chamber contains six layers, and each layer contains cathode strips that run radially, and wires that run approximately orthogonal to the cathode strips. The cathode strips vary in width from 4 to 16 mm, depending on the radius from the beam line and the specific location of the chamber within CMS. The anode wires are directly wired together in sets of 5 to 17 wires per readout channel, covering 1.6 to 5.4 cm radially. Cathodes and anodes are both instrumented with trigger as well as readout electronics.

The cathode trigger electronics of the CSC require a certain minimum charge deposition to register a muon hit. Although the threshold is well below minimum ionizing, the random noise level of cathode hits is very low. For the trigger, the muon hits are further localized with an accuracy of one-half of a strip on each chamber layer by analog comparison of charge deposition on each strip with its neighbors as well as comparison of the neighbor strips – the technique is built into Comparator ASIC chips. The CSC anode electronics front-end has constant-fraction discriminators that create muon hits with very little time walk; the CSC anode hits are registered each LHC bunch crossing.

The baseline CSC trigger primitives, known as Local Charged Tracks (LCT), constitute the input from the CSC trigger motherboards (TMBs) to the L1 muon trigger track finders. The LCT are a coincidence between straight-line patterns found in anode and cathode electronics. Both anode and cathode trigger electronics require at least four layers to contain hits within patterns. As the magnetic field is solenoidal, muons do not bend significantly in the $R - z$ plane measured by CSC anode wires, and so there is essentially only one CSC trigger anode pattern that merely indicates that the muon appeared to have originated roughly from the CMS collision point. On the other hand, muons do bend in the $r - \phi$ plane, and a set of nine cathode trigger patterns is used to indicate the amount and direction of bending. The bending is inversely related to momentum, and is largest in the first CSC station, i.e. the one closest in z to the interaction point. Occupancy of CSC chambers by tracks and neutron-induced hits is much higher in the inner ring closest to the beam line, and is higher in the first CSC station than in the other stations.

The performance of the CSC trigger primitive generation has been excellent in LHC running thus far. The CSC trigger primitives are generated with 98% efficiency in all stations except ME1/1, where it is around 94%, localize the muon positions in the ϕ (bend) direction within an RMS of 0.174 strips, and find the correct bunch crossing for well over 99% of the muons.

The major upgrade to the CSC will replace on-chamber cathode boards on the inner rings of chambers ($1.6 < \eta < 2.4$) in order to handle higher trigger and output data rates, and FPGA mezzanine boards on most of the on-chamber anode boards in order to cope with higher L1 trigger latency. Corresponding off-chamber boards that receive trigger and readout data will also be replaced to handle the higher data rates.

The CSC trigger primitives are sent to the L1 trigger track finders via optical links from Muon Port Cards (MPC) located in crates on the periphery of the endcap muon system. There are 60 MPCs, and each sends the LCT information from 9 (O)TMB boards to the EMTF/OMTF sector processors over 8 optical links operating at 3.2 Gb/s with 38b/40b encoding.

The algorithms of the CSC trigger are expected to change substantially in the future. If the

108 algorithms do not change, efficiency loss is expected due to high-occupancy effects such as
 109 deadtime that grow with luminosity. The loss of efficiency is worst in the ME1/1 chambers,
 110 which can lose as high as 15% efficiency, but this will be almost completely alleviated by up-
 111 dated firmware that allows simultaneous processing of trigger hits into LCTs in different parts
 112 of each CSC chamber. Additionally, the upgraded FPGAs used for the new electronics will
 113 allow finer granularity of trigger patterns; studies have shown that the cathode (bend direc-
 114 tion) position and angle resolutions can be improved by factors of 1.87 and 1.35, respectively.
 115 Studies to determine whether the anode position and angle resolution can be improved as well
 116 are being done. Larger factors of improvement in the bend direction resolution are possible by
 117 combination of CSC trigger hits with GEM hits in the same station.

118 2.2 Gas Electron Multipliers

119 2.2.1 GE1/1

120 A single GE1/1 chamber is made of three GEM foils. A stack of two GEM chambers forms a
 121 superchamber. These superchambers will be installed in the forward region ($1.6 < |\eta| < 2.4$),
 122 dubbed GE1/1. The GE1/1 station features 36 superchambers, each having two chambers, in
 123 front of the ME1/1 chambers. Each superchamber covers 10 degrees in ϕ and $1.6 < |\eta| < 2.15$
 124 in pseudorapidity.

125
 126 The GE1/1 detectors will deliver trigger pad clusters onto two separate trigger paths: to the
 127 neighboring CSCs and to the EMTF via a CTP7 card (GEM concentrator). Both types of trigger
 128 primitives will be constructed from trigger pads which have a angular resolution of 0.9 mrad in
 129 ϕ . The trigger pads are built on-chamber in the VFAT3 chip [2] as an OR'ed combination of two
 130 neighboring strips. GE1/1 has 192 trigger pads per eta partition (8 eta partitions per chamber).
 131 Single trigger pads are at least 97% efficient. The trigger pad data will be transmitted from the
 132 VFAT3 chip to the OptoHybrid (OH) board. The GE1/1 OH boards will construct clusters from
 133 maximum 8 adjacent trigger pads in the chamber. Each pad cluster is 14 bits wide. The cluster
 134 data will be transmitted from the OH to the nearby CSC optical trigger motherboard (OTMB)
 135 and to the GEM concentrator. Per superchamber, up to 16 clusters will be transmitted from
 136 GE1/1 OHs. The overflow rate of the cluster finder has been estimated to be $\mathcal{O}(10^{-5})$ per BX.

137
 138 In the GE1/1-ME1/1 trigger path, GEM trigger clusters are combined with CSC Cathode and
 139 Anode LCTs to form highly efficient GE1/1-ME1/1 integrated local triggers. The requirement
 140 in the number of hit layers can be relaxed from 'at least four CSC layers hit' to 'at least three
 141 CSC layers and at least one GEM layer hit'. The GE1/1-ME1/1 integrated local trigger ob-
 142 jects will have a width of 32 bits as regular CSC LCTs, but unused or redundant bits will be
 143 repurposed to indicate the presence of GEMs. In the GE1/1-EMTF trigger path, the GE1/1
 144 concentrator multiplexes trigger pad cluster data from every three OH optical links running at
 145 3.2 Gb/s into a single link running at 9.6 Gb/s. Coincidences of trigger pad clusters (with an
 146 efficiency $> 95\%$) will be used in the EMTF track-builder and momentum assignment.

147 2.2.2 GE2/1 and ME0

148 Beyond Run-3, two additional GEM detectors will be installed, 2-layer GE2/1 chambers, and
 149 6-layer ME0 chambers. The GE2/1 station is similar to the GE1/1 station, although GE2/1
 150 chambers are much larger. Eighteen superchambers, each 20 degrees wide, cover the region
 151 $1.6 < |\eta| < 2.4$. CMS will also be equipped with a new ME0 station behind the HGAL, closer
 152 to the interaction point than any other endcap muon detector. ME0 superchambers have six
 153 20-degree wide chambers and cover the region $2.0 < |\eta| < 2.8$. GE2/1 and ME0 each have

154 384 pads eta partition. In the GE2/1-EMTF trigger path, eight 3.52 Gb/s links are multiplexed
 155 into a single 25 Gb/s link. Per superchamber, up to 40 clusters will be transmitted from GE2/1
 156 OHs. For completeness, the GE2/1 format is also described in this note. The description of the
 157 ME0 data format is beyond the scope of this document.

158

159 A simplified picture of the overlap and endcap trigger primitive architecture is shown in Fig. 2.

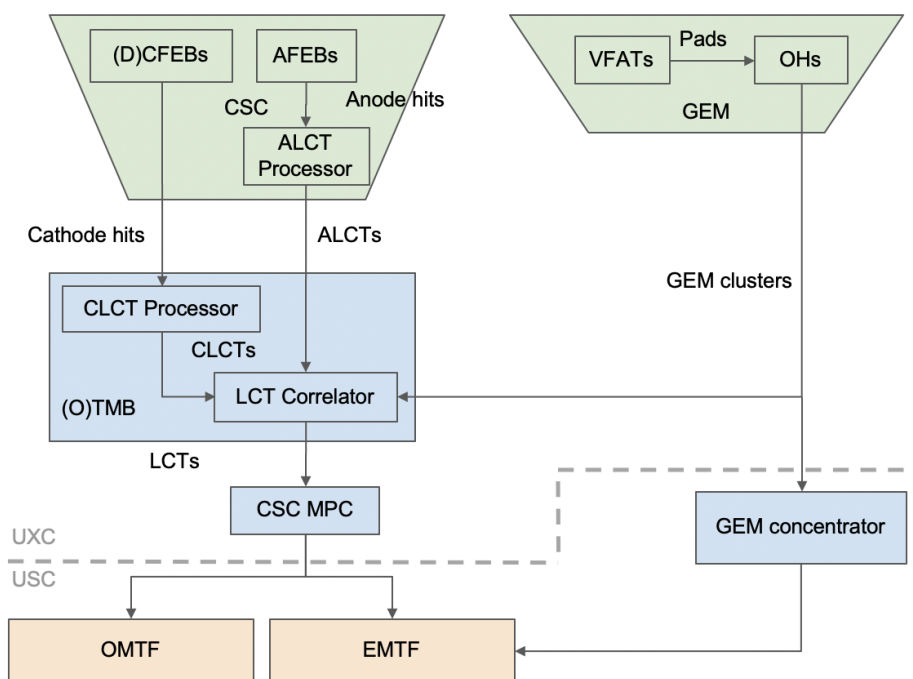


Figure 2: Architecture for the muon overlap and endcap trigger primitives (not showing RPCs) in Run-3. CSC TPs are produced in the trigger motherboard (TMB) and are also sent to the OMTF and EMTF through the muon port card (MPC). GEM pad clusters are sent directly to the OTMB and to the EMTF via a GEM concentrator card.

3 CSC Trigger Data Formats

3.1 ALCT to (O)TMB

The original ALCT Virtex-E mezzanine cards allowed for Skewclear copper links carrying 56 bits/bx to the TMB boards that are divided between DAQ (16 bits), trigger (11 bits per ALCT x 2), and other purposes such as synchronization and error correction (18 bits total), see [3].

The new ALCT LX100 and LX150T FPGAs will be outfitted with optical links to ODMBs, freeing up 16 Skewclear DAQ bits which then can be repurposed used for the trigger, but this is not foreseen until new ODMBs and new FEDs are installed after Run-3.

ALCT bits are copied into the LCT data word sent by the (O)TMBs, with a few exceptions. In Run-3, five bits are reserved for the timing in the ALCT_BXN[4:0]. However, ALCT_BXN[0] and ALCT_BXN[3:4] are used in the (O)TMB logic. ALCT_BXN[2:1] are not used. This proposal reallocates these two bits for high multiplicity triggers targeting physics searches for exotic muon-like signatures in Run-3, BXN[2:1] → ALCT_HMT[1:0]. The resulting trigger data format is shown in Tab. 1.

Table 1: ALCT to (O)TMB trigger data format, multiplexed at 80 MHz. Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	ALCT0_Valid		ALCT1_Valid	
1	ALCT0_AccelMuon		ALCT1_AccelMuon	
2	ALCT0_QUAL[0]		ALCT1_QUAL[0]	
3	ALCT0_QUAL[1]		ALCT1_QUAL[1]	
4	ALCT0_WG[0]		ALCT1_WG[0]	
5	ALCT0_WG[1]		ALCT1_WG[1]	
6	ALCT0_WG[2]		ALCT1_WG[2]	
7	ALCT0_WG[3]		ALCT1_WG[3]	
8	ALCT0_WG[4]		ALCT1_WG[4]	
9	ALCT0_WG[5]		ALCT1_WG[5]	
10	ALCT0_WG[6]		ALCT1_WG[6]	
11	BXN[0]		BXN[3]	
12	BXN[1]	ALCT_HMT[0]	BXN[4]	
13	BXN[2]	ALCT_HMT[1]	wr_fifo	
14	DAQ_data[0]		DAQ_data[7]	
15	DAQ_data[1]		DAQ_data[8]	
16	DAQ_data[2]		DAQ_data[9]	
17	DAQ_data[3]		DAQ_data[10]	
18	DAQ_data[4]		DAQ_data[11]	
19	DAQ_data[5]		DAQ_data[12]	
20	DAQ_data[6]		DAQ_data[13]	
21	lct_special		first_frame	
22	parity[0]		parity[2]	
23	parity[1]		parity[3]	
24	ddu_special		last_frame	
25	parity[4]		parity[6]	
26	parity[5]		ttc_bc0	
27	active_feb_flag		cfg_done	

174 3.2 CLCT in (O)TMB

175 The CLCT processor is not a distinct piece of hardware in the CSC trigger. Rather, it is a logic
 176 block in the (O)TMB that produces CLCT trigger primitives from cathode hits (triads). Because
 177 the CLCT object is internal to the (O)TMB, no table similar to 1 can be constructed. For com-
 178 pleteness, we mention the relevant data words in a CLCT object.

179
 180 In Run-3, the CLCT processor will be upgraded to include the comparator-code lookup ta-
 181 ble algorithm (CCLUT) that enhances the CLCT position resolution (by a factor 2-4) and the
 182 bending resolution by factor 3 [4]. This improvement will be facilitated by a new 12-bit com-
 183 parator code CLCT_COMP_CODE[11:0]. Five patterns (0 through 4) will replace the Run-2
 184 patterns (2 through A). However, in the firmware, new signals will be added for the Run-
 185 3 pattern (CLCT_PAT_ID[2:0]) to maintain backward-compatibility. New signals will also be
 186 added for the 1/4-strip and 1/8-strip bits CLCT_QS, CLCT_ES. Words that remain the same are
 187 CLCT_HS[7:0], CLCT_QUAL[2:0], CLCT_VALID and CLCT_BXN[4:0].

188 3.3 (O)TMB to MPC

189 3.3.1 Inputs for the (O)TMB Correlation Algorithm

190 The trigger motherboards receive anode trigger data from the ALCT board over copper links
 191 (for older TMBs) and optical links (for newer (O)TMBs). Cathode data is sent directly from the
 192 (D)CFEBs to the motherboards. CLCT processor units build CLCTs, which are correlated with
 193 ALCTs using the timing to form LCTs. ME1/1 OTMBs and ME2/1 OTMBs can also receive
 194 GEM pad clusters from the nearby GE1/1 (starting Run-3) and GE2/1 OHs (starting Run-
 195 4). The clusters will be used in those chambers to build integrated local trigger primitives,
 196 combining GEM and CSC information.

197 3.3.2 Quick Overview of the Trigger Data format

198 The 64 bits trigger data are sent on the peripheral backplane in two frames of 32 bits each. Each
 199 frame contains 2 LCTs. Each LCT is assigned 27 bits: strip position (8+2 bits), wiregroup posi-
 200 tion (7 bits), quality (3 bits), strip slope (4+1 bits), a valid pattern flag (VPF: 1 bit) and a BC0 (1
 201 bit) for trigger synchronization. Technically, the VPF is redundant, as the quality denotes if an
 202 LCT is valid. However, to maintain backward compatibility of the firmware, this bit is kept in
 203 the trigger data. The remaining ten bits include five bits for the CLCT pattern IDs (plural!), 4
 204 bits for high-multiplicity triggering, and 1 bit for the LCT0_BXN[0] (for LCT timing). There are
 205 no reserved bits. The interpretation of the bits for high-multiplicity triggering are to be speci-
 206 fied according to Monte Carlo simulations with long-lived particles decaying in the CSCs.

207
 208 The original 2005 TMB design and data formats can be found in Ref. [3].

209 3.3.3 Common Redefinitions

210 Each LCT is reassigned two bits to encode the 1/4 strip and 1/8 strip number obtained with
 211 the CCLUT algorithm, i.e

- 212 • First LCT 1/4 strip: LCT0_QUAL[3] → LCT0_CLCT_QS
- 213 • Second LCT 1/4 strip: LCT1_QUAL[3] → LCT1_CLCT_QS
- 214 • First LCT 1/8 strip: LCT0_SYER → LCT0_CLCT_ES
- 215 • Second LCT 1/8 strip: LCT1_SYER → LCT1_CLCT_ES

216 The 4-bit CLCT Run-2 pattern ID, LCT0.CLCT_PAT_ID[3:0] and LCT1.CLCT_PAT_ID[3:0], ran
 217 from 2 through A. Valid Run-3 patterns run from 0 through 4. A five bit pattern ID (LCT.CLCT_PAT_ID[4:0])
 218 is sufficient to encode all 25 combinations $0, 4 \times 0, 4$.

- 219 • LCT0.CLCT_PAT_ID[0] → LCT.CLCT_PAT_ID[0]
- 220 • LCT0.CLCT_PAT_ID[1] → LCT.CLCT_PAT_ID[1]
- 221 • LCT0.CLCT_PAT_ID[2] → LCT.CLCT_PAT_ID[2]
- 222 • LCT0.CLCT_PAT_ID[3] → LCT.CLCT_PAT_ID[3]
- 223 • LCT1.CLCT_PAT_ID[0] → LCT.CLCT_PAT_ID[4]

224 The 4-bit CSC ID is transmitted from the MPC to the Muon Track Finder Sector Processor, and
 225 it is now reassigned as the CLCT bending/slope value obtained with CCLUT. The sign of the
 226 bending (left or right), LCT0.CLCT_LR and LCT1.CLCT_LR, remain the same.

- 227 • LCT0.CSC_ID[3:0] → LCT0.CLCT_BEND[3:0]
- 228 • LCT1.CSC_ID[3:0] → LCT1.CLCT_BEND[3:0]

229 The 4-bit Run-2 quality data is reduced to a 3-bit Run-3 quality data, LCT0_QUAL[2:0] and
 230 LCT1_QUAL[2:0]. We foresee different definitions for ME1/1 and non-ME1/1 chambers. See 3.3.5
 231 and 3.3.6.

232 Four bits per CSC are assigned for high-multiplicity triggering on exotic signatures, to count
 233 ALCT/CLCT trigger primitives and/or to count anode/cathode hits.

- 234 • LCT1.BXN[0] → HMT[0]
- 235 • LCT1.CLCT_PAT_ID[1] → HMT[1]
- 236 • LCT1.CLCT_PAT_ID[2] → HMT[2]
- 237 • LCT1.CLCT_PAT_ID[3] → HMT[3]

238 The remaining reserved bits are LCT1.BXN[0] and LCT1.BC0.

239 The Run-3 TMB-to-MPC trigger data format is shown in Table 2.

Table 2: (O)TMB to MPC trigger data format, multiplexed at 80 MHz. Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	LCT0.ALCT.WG[0]		LCT0.CLCT.HS[0]	
1	LCT0.ALCT.WG[1]		LCT0.CLCT.HS[1]	
2	LCT0.ALCT.WG[2]		LCT0.CLCT.HS[2]	
3	LCT0.ALCT.WG[3]		LCT0.CLCT.HS[3]	
4	LCT0.ALCT.WG[4]		LCT0.CLCT.HS[4]	
5	LCT0.ALCT.WG[5]		LCT0.CLCT.HS[5]	
6	LCT0.ALCT.WG[6]		LCT0.CLCT.HS[6]	
7	LCT0.CLCT.PAT_ID[0]	LCT.CLCT.PAT_ID[0]	LCT0.CLCT.HS[7]	
8	LCT0.CLCT.PAT_ID[1]	LCT.CLCT.PAT_ID[1]	LCT0.CLCT_LR	
9	LCT0.CLCT.PAT_ID[2]	LCT.CLCT.PAT_ID[2]	LCT0.SYER	LCT0.CLCT.ES
10	LCT0.CLCT.PAT_ID[3]	LCT.CLCT.PAT_ID[3]	LCT0.BXN[0]	
11	LCT0.QUAL[0]		LCT0.BC0	
12	LCT0.QUAL[1]		LCT0.CSC_ID[0]	LCT0.CLCT.BEND[0]
13	LCT0.QUAL[1]		LCT0.CSC_ID[1]	LCT0.CLCT.BEND[1]
14	LCT0.QUAL[3]	LCT0.CLCT.QS	LCT0.CSC_ID[2]	LCT0.CLCT.BEND[2]
15	LCT0.VPF		LCT0.CSC_ID[3]	LCT0.CLCT.BEND[3]
16	LCT1.ALCT.WG[0]		LCT1.CLCT.HS[0]	
17	LCT1.ALCT.WG[1]		LCT1.CLCT.HS[1]	
18	LCT1.ALCT.WG[2]		LCT1.CLCT.HS[2]	
19	LCT1.ALCT.WG[3]		LCT1.CLCT.HS[3]	
20	LCT1.ALCT.WG[4]		LCT1.CLCT.HS[4]	
21	LCT1.ALCT.WG[5]		LCT1.CLCT.HS[5]	
22	LCT1.ALCT.WG[6]		LCT1.CLCT.HS[6]	
23	LCT1.CLCT.PAT_ID[0]	LCT.CLCT.PAT_ID[4]	LCT1.CLCT.HS[7]	
24	LCT1.CLCT.PAT_ID[1]	HMT[1]	LCT1.CLCT_LR	
25	LCT1.CLCT.PAT_ID[2]	HMT[2]	LCT1.SYER	LCT1.CLCT.ES
26	LCT1.CLCT.PAT_ID[3]	HMT[3]	LCT1.BXN[0]	HMT[0]
27	LCT1.QUAL[0]		LCT1.BC0	
28	LCT1.QUAL[1]		LCT0.CSC_ID[0]	LCT0.CLCT.BEND[0]
29	LCT1.QUAL[2]		LCT1.CSC_ID[1]	LCT1.CLCT.BEND[1]
30	LCT1.QUAL[3]	LCT1.CLCT.QS	LCT1.CSC_ID[2]	LCT1.CLCT.BEND[2]
31	LCT1.VPF		LCT1.CSC_ID[3]	LCT1.CLCT.BEND[3]

240 3.3.4 Specific Redefinition: TMBs which do not have the CCLUT algorithm enabled

241 CCLUT will not be enabled in the ME1/3 and MEX/2 TMBs on day-1 of Run-3. This means that
 242 LCTs would be formed with the Run-2 pattern finder as before. The data would be shipped ac-
 243 cording to the Run-3 data format, with the exception that $LCT_i_CLCT_BEND[3:0]$ ($i=0,1$) would
 244 not indicate the Run-3 slope, but the Run-2 pattern. $LCT_CLCT_PAT_ID[4:0]$, which would
 245 normally encode the Run-3 pattern, would be empty. For TMBs, the 1/4-strip and 1/8-strip
 246 bits would be empty. They would be derived in the EMTF using the Run-2 pattern numbers
 247 for LCTs. This ensures that CSC primitives all have the same position resolution - albeit from
 248 different origin.

249

250 For OTMBs in MEX/1 with CCLUT enabled, the EMTF would have to convert the 1-bit bend
 251 and 4-bit slope into the 4-bit Run-2 pattern number, so that all CSC LCTs at the EMTF have a
 252 4-bit Run-2 pattern number (again albeit from different origin) for the track building and pT
 253 assignment. The 1/4-strip and 1/8-strip bits will be obtained from the MPC data stream.

254 3.3.5 Specific Redefinition: ME1/1 Quality

255 The quality indicates the presence of GEMs and the source of the bending angle.

Table 3: Preliminary ME1/1 LCT 3-bit quality data word definition

LCT Type	ALCT Wire group Source	CLCT Strip Source	Bending Source	Quality Value
ALCT+CLCT+2GEM	ALCT WG	CLCT strip	GEM-CSC	7
ALCT+CLCT+2GEM	ALCT WG	CLCT strip	CSC-only	6
ALCT+CLCT+1GEM	ALCT WG	CLCT strip	GEM-CSC	5
ALCT+CLCT+1GEM	ALCT WG	CLCT strip	CSC-only	4
ALCT+CLCT	ALCT WG	CLCT strip	CSC-only	3
CLCT+2GEM	GEM roll	CLCT strip	GEM-CSC	2
ALCT+2GEM	ALCT WG	GEM pad	GEM-only	1
Invalid	-	-	-	0

256 3.3.6 Specific Redefinition: non-ME1/1 Quality

257 Only two of the three bits are used. The quality indicates the number of layers hit ($\leq 3, 4, 5, 6$);

Table 4: Preliminary non-ME1/1 LCT quality data word definition.

LCT Type	Quality value
High quality (6 layers)	3
Medium quality (5 layers)	2
Low quality (4 layers)	1
Invalid (≤ 3 layers)	0

258 **3.3.7 Map of 4+1 bit Run-3 Bending onto 4-bit Run-2 Pattern**

Table 5: LCT0_CLCT_LR bit abbreviated to “LR”. “Run-3 bending” refers to LCTi_CLCT_BEND[3:0], while “Run-2 pattern” refers to LCTi_CLCT_PAT_ID[3:0].

LR	Run-3 bending	Run-2 pattern	LR	Run-3 bending	Run-2 pattern
0	0	10	1	0	10
0	1	10	1	1	10
0	2	10	1	2	10
0	3	8	1	3	9
0	4	8	1	4	9
0	5	8	1	5	9
0	6	6	1	6	7
0	7	6	1	7	7
0	8	6	1	8	7
0	9	4	1	9	5
0	10	4	1	10	5
0	11	4	1	11	5
0	12	2	1	12	3
0	13	2	1	13	3
0	14	2	1	14	3
0	15	2	1	15	3

259 **3.3.8 Pattern ID Lookup**

Table 6: Map of the 5-bit pattern ID onto two {0,4} pattern IDs. Note, when LCT1.VPF is 0 (invalid), the pattern ID of LCT0 is 0b00000 through 0b00100.

Pattern ID	{Pat1, Pat0}	Pattern ID	{Pat1, Pat0}
0b00000	{-,0}	0b10000	{2,1}
0b00001	{-,1}	0b10001	{2,2}
0b00010	{-,2}	0b10010	{2,3}
0b00011	{-,3}	0b10011	{2,4}
0b00100	{-,4}	0b10100	{3,0}
0b00101	{0,0}	0b10101	{3,1}
0b00110	{0,1}	0b10110	{3,2}
0b00111	{0,2}	0b10111	{3,3}
0b01000	{0,3}	0b11000	{3,4}
0b01001	{0,4}	0b11001	{4,0}
0b01010	{1,0}	0b11010	{4,1}
0b01011	{1,1}	0b11011	{4,2}
0b01100	{1,2}	0b11100	{4,3}
0b01101	{1,3}	0b11101	{4,4}
0b01110	{1,4}	0b11110	RESERVED
0b01111	{2,0}	0b11111	RESERVED

260 **3.4 MPC to Muon Track Finder Sector Processor**

261 Trigger data from 9 motherboards are sent from the MPC to the OMTF/EMTF onto 8 links with
 262 80 bits per link per BX [5]. They are split in two frames, carrying 40 bits per frame per BX. Each
 263 frame has 38 bits for (O)TMB data and 2 bits for frame alignment. For each (O)TMB, all 64 bits
 264 are sent from the MPC to the EMTF using the 38b40b protocol, which is similar to the 64b66b
 265 protocol. A total bandwidth of 76 bits per link per BX is available after encoding, which is more
 266 sufficient to send 64 bits per link per BX. The 11 unused bits in Run-2 TMB to MPC transmission
 267 are repurposed and passed to OMTF/EMTF. There are three spare bits per link per BX, which
 268 will be used for error detection in each BX. The MPC to EMTF data format, multiplexed at 80
 269 MHz, is shown in tables 7-14. The GTPs represent 8 gigabit transmitters (GTP) in the MPC
 270 FPGA, all running at 3.2 Gb/s.

Table 7: GTP1 (MGTTX0.101) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB2.LCT0.CLCT_HS[0]		TMB2.LCT1.CLCT_HS[0]	
1	TMB2.LCT0.CLCT_HS[1]		TMB2.LCT1.CLCT_HS[1]	
2	TMB2.LCT0.CLCT_HS[2]		TMB2.LCT1.CLCT_HS[2]	
3	TMB2.LCT0.CLCT_HS[3]		TMB2.LCT1.CLCT_HS[3]	
4	TMB2.LCT0.CLCT_HS[4]		TMB2.LCT1.CLCT_HS[4]	
5	TMB2.LCT0.CLCT_HS[5]		TMB2.LCT1.CLCT_HS[5]	
6	TMB2.LCT0.CLCT_HS[6]		TMB2.LCT1.CLCT_HS[6]	
7	TMB2.LCT0.CLCT_HS[7]		TMB2.LCT1.CLCT_HS[7]	
8	TMB2.LCT0.ALCT_WG[0]		TMB2.LCT1.ALCT_WG[0]	
9	TMB2.LCT0.ALCT_WG[1]		TMB2.LCT1.ALCT_WG[1]	
10	TMB2.LCT0.ALCT_WG[2]		TMB2.LCT1.ALCT_WG[2]	
11	TMB2.LCT0.ALCT_WG[3]		TMB2.LCT1.ALCT_WG[3]	
12	TMB2.LCT0.ALCT_WG[4]		TMB2.LCT1.ALCT_WG[4]	
13	TMB2.LCT0.ALCT_WG[5]		TMB2.LCT1.ALCT_WG[5]	
14	TMB2.LCT0.ALCT_WG[6]		TMB2.LCT1.ALCT_WG[6]	
15	TMB2.LCT0_QUAL[0]		TMB2.LCT1_QUAL[0]	
16	TMB2.LCT0_QUAL[1]		TMB2.LCT1_QUAL[1]	
17	TMB2.LCT0_QUAL[2]		TMB2.LCT1_QUAL[2]	
18	TMB2.LCT0_QUAL[3]	TMB2.LCT0.CLCT_QS	TMB2.LCT1_QUAL[3]	TMB2.LCT1.CLCT_QS
19	TMB2.LCT0.CLCT_PAT_ID[0]	TMB2.LCT.CLCT_PAT_ID[0]	TMB2.LCT1.CLCT_PAT_ID[0]	TMB2.LCT.CLCT_PAT_ID[4]
20	TMB2.LCT0.CLCT_PAT_ID[1]	TMB2.LCT.CLCT_PAT_ID[1]	TMB2.LCT1.CLCT_PAT_ID[1]	TMB2_HMT[1]
21	TMB2.LCT0.CLCT_PAT_ID[2]	TMB2.LCT.CLCT_PAT_ID[2]	TMB2.LCT1.CLCT_PAT_ID[2]	TMB2_HMT[2]
22	TMB2.LCT0.CLCT_PAT_ID[3]	TMB2.LCT.CLCT_PAT_ID[3]	TMB2.LCT1.CLCT_PAT_ID[3]	TMB2_HMT[3]
23	TMB2.LCT0_LR		TMB2.LCT1_LR	
24	TMB2.LCT0_VPF		TMB2.LCT1_VPF	
25	TMB2.LCT0_BC0		TMB2.LCT1_BC0	
26	TMB2.LCT0_BXN[0]		TMB2.LCT1_BXN[0]	TMB2_HMT[0]
27	TMB2.LCT0_SYER	TMB2.LCT0.CLCT_ES	TMB2.LCT1_SYER	TMB2.LCT1.CLCT_ES
28	TMB2.LCT0_CSC_ID[0]	TMB2.LCT0.CLCT_BEND[0]	TMB2.LCT1_CSC_ID[0]	TMB2.LCT1.CLCT_BEND[0]
29	TMB2.LCT0_CSC_ID[1]	TMB2.LCT0.CLCT_BEND[1]	TMB2.LCT1_CSC_ID[1]	TMB2.LCT1.CLCT_BEND[1]
30	TMB2.LCT0_CSC_ID[2]	TMB2.LCT0.CLCT_BEND[2]	TMB2.LCT1_CSC_ID[2]	TMB2.LCT1.CLCT_BEND[2]
31	TMB2.LCT0_CSC_ID[3]	TMB2.LCT0.CLCT_BEND[3]	TMB2.LCT1_CSC_ID[3]	TMB2.LCT1.CLCT_BEND[3]
32	TMB1.LCT0.CLCT_HS[0]		TMB1.LCT0.CLCT_HS[4]	
33	TMB1.LCT0.CLCT_HS[1]		TMB1.LCT0.CLCT_HS[5]	
34	TMB1.LCT0.CLCT_HS[2]		TMB1.LCT0.CLCT_HS[6]	
35	TMB1.LCT0.CLCT_HS[3]		TMB1.LCT0.CLCT_HS[7]	
36	TMB1.LCT0_VPF		TMB1.LCT0_BC0	
37	CRC[0]		CRC[1]	

Table 8: GTP2 (MGTTX1_101) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB3.LCT0.CLCT_HS[0]		TMB3.LCT1.CLCT_HS[0]	
1	TMB3.LCT0.CLCT_HS[1]		TMB3.LCT1.CLCT_HS[1]	
2	TMB3.LCT0.CLCT_HS[2]		TMB3.LCT1.CLCT_HS[2]	
3	TMB3.LCT0.CLCT_HS[3]		TMB3.LCT1.CLCT_HS[3]	
4	TMB3.LCT0.CLCT_HS[4]		TMB3.LCT1.CLCT_HS[4]	
5	TMB3.LCT0.CLCT_HS[5]		TMB3.LCT1.CLCT_HS[5]	
6	TMB3.LCT0.CLCT_HS[6]		TMB3.LCT1.CLCT_HS[6]	
7	TMB3.LCT0.CLCT_HS[7]		TMB3.LCT1.CLCT_HS[7]	
8	TMB3.LCT0.ALCT_WG[0]		TMB3.LCT1.ALCT_WG[0]	
9	TMB3.LCT0.ALCT_WG[1]		TMB3.LCT1.ALCT_WG[1]	
10	TMB3.LCT0.ALCT_WG[2]		TMB3.LCT1.ALCT_WG[2]	
11	TMB3.LCT0.ALCT_WG[3]		TMB3.LCT1.ALCT_WG[3]	
12	TMB3.LCT0.ALCT_WG[4]		TMB3.LCT1.ALCT_WG[4]	
13	TMB3.LCT0.ALCT_WG[5]		TMB3.LCT1.ALCT_WG[5]	
14	TMB3.LCT0.ALCT_WG[6]		TMB3.LCT1.ALCT_WG[6]	
15	TMB3.LCT0.QUAL[0]		TMB3.LCT1.QUAL[0]	
16	TMB3.LCT0.QUAL[1]		TMB3.LCT1.QUAL[1]	
17	TMB3.LCT0.QUAL[2]		TMB3.LCT1.QUAL[2]	
18	TMB3.LCT0.QUAL[3]	TMB3.LCT0.CLCT_QS	TMB3.LCT1.QUAL[3]	TMB3.LCT1.CLCT_QS
19	TMB3.LCT0.CLCT_PAT_ID[0]	TMB3.LCT.CLCT_PAT_ID[0]	TMB3.LCT1.CLCT_PAT_ID[0]	TMB3.LCT.CLCT_PAT_ID[4]
20	TMB3.LCT0.CLCT_PAT_ID[1]	TMB3.LCT.CLCT_PAT_ID[1]	TMB3.LCT1.CLCT_PAT_ID[1]	TMB3.HMT[1]
21	TMB3.LCT0.CLCT_PAT_ID[2]	TMB3.LCT.CLCT_PAT_ID[2]	TMB3.LCT1.CLCT_PAT_ID[2]	TMB3.HMT[2]
22	TMB3.LCT0.CLCT_PAT_ID[3]	TMB3.LCT.CLCT_PAT_ID[3]	TMB3.LCT1.CLCT_PAT_ID[3]	TMB3.HMT[3]
23				
24	TMB3.LCT0.LR		TMB3.LCT1.LR	
25	TMB3.LCT0.VPF		TMB3.LCT1.VPF	
26	TMB3.LCT0.BC0		TMB3.LCT1.BC0	
27	TMB3.LCT0.BXN[0]		TMB3.LCT1.BXN[0]	TMB3.HMT[0]
28	TMB3.LCT0.SYER	TMB3.LCT0.CLCT_ES	TMB3.LCT1.SYER	TMB3.LCT1.CLCT_ES
29	TMB3.LCT0.CSC_ID[0]	TMB3.LCT0.CLCT_BEND[0]	TMB3.LCT1.CSC_ID[0]	TMB3.LCT1.CLCT_BEND[0]
30	TMB3.LCT0.CSC_ID[1]	TMB3.LCT0.CLCT_BEND[1]	TMB3.LCT1.CSC_ID[1]	TMB3.LCT1.CLCT_BEND[1]
31	TMB3.LCT0.CSC_ID[2]	TMB3.LCT0.CLCT_BEND[2]	TMB3.LCT1.CSC_ID[2]	TMB3.LCT1.CLCT_BEND[2]
32	TMB3.LCT0.CSC_ID[3]	TMB3.LCT0.CLCT_BEND[3]	TMB3.LCT1.CSC_ID[3]	TMB3.LCT1.CLCT_BEND[3]
33	TMB1.LCT0.ALCT_WG[0]		TMB1.LCT0.ALCT_WG[4]	
34	TMB1.LCT0.ALCT_WG[1]		TMB1.LCT0.ALCT_WG[5]	
35	TMB1.LCT0.ALCT_WG[2]		TMB1.LCT0.ALCT_WG[6]	
36	TMB1.LCT0.ALCT_WG[3]		TMB1.LCT0.CLCT_LR	
37	TMB1.LCT0.VPF		TMB1.LCT0.BC0	
	CRC[0]		CRC[1]	

Table 9: GTP3 (MGTTX0.123) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB4.LCT0.CLCT_HS[0]		TMB4.LCT1.CLCT_HS[0]	
1	TMB4.LCT0.CLCT_HS[1]		TMB4.LCT1.CLCT_HS[1]	
2	TMB4.LCT0.CLCT_HS[2]		TMB4.LCT1.CLCT_HS[2]	
3	TMB4.LCT0.CLCT_HS[3]		TMB4.LCT1.CLCT_HS[3]	
4	TMB4.LCT0.CLCT_HS[4]		TMB4.LCT1.CLCT_HS[4]	
5	TMB4.LCT0.CLCT_HS[5]		TMB4.LCT1.CLCT_HS[5]	
6	TMB4.LCT0.CLCT_HS[6]		TMB4.LCT1.CLCT_HS[6]	
7	TMB4.LCT0.CLCT_HS[7]		TMB4.LCT1.CLCT_HS[7]	
8	TMB4.LCT0.ALCT_WG[0]		TMB4.LCT1.ALCT_WG[0]	
9	TMB4.LCT0.ALCT_WG[1]		TMB4.LCT1.ALCT_WG[1]	
10	TMB4.LCT0.ALCT_WG[2]		TMB4.LCT1.ALCT_WG[2]	
11	TMB4.LCT0.ALCT_WG[3]		TMB4.LCT1.ALCT_WG[3]	
12	TMB4.LCT0.ALCT_WG[4]		TMB4.LCT1.ALCT_WG[4]	
13	TMB4.LCT0.ALCT_WG[5]		TMB4.LCT1.ALCT_WG[5]	
14	TMB4.LCT0.ALCT_WG[6]		TMB4.LCT1.ALCT_WG[6]	
15	TMB4.LCT0.QUAL[0]		TMB4.LCT1.QUAL[0]	
16	TMB4.LCT0.QUAL[1]		TMB4.LCT1.QUAL[1]	
17	TMB4.LCT0.QUAL[2]		TMB4.LCT1.QUAL[2]	
18	TMB4.LCT0.QUAL[3]	TMB4.LCT0.CLCT_QS	TMB4.LCT1.QUAL[3]	TMB4.LCT1.CLCT_QS
19	TMB4.LCT0.CLCT_PAT_ID[0]	TMB4.LCT.CLCT_PAT_ID[0]	TMB4.LCT1.CLCT_PAT_ID[0]	TMB4.LCT.CLCT_PAT_ID[4]
20	TMB4.LCT0.CLCT_PAT_ID[1]	TMB4.LCT.CLCT_PAT_ID[1]	TMB4.LCT1.CLCT_PAT_ID[1]	TMB4.HMT[1]
21	TMB4.LCT0.CLCT_PAT_ID[2]	TMB4.LCT.CLCT_PAT_ID[2]	TMB4.LCT1.CLCT_PAT_ID[2]	TMB4.HMT[2]
22	TMB4.LCT0.CLCT_PAT_ID[3]	TMB4.LCT.CLCT_PAT_ID[3]	TMB4.LCT1.CLCT_PAT_ID[3]	TMB4.HMT[3]
23	TMB4.LCT0.LR		TMB4.LCT1.LR	
24	TMB4.LCT0.VPF		TMB4.LCT1.VPF	
25	TMB4.LCT0.BC0		TMB4.LCT1.BC0	
26	TMB4.LCT0.BXN[0]		TMB4.LCT1.BXN[0]	TMB4.HMT[0]
27	TMB4.LCT0.SYER	TMB4.LCT0.CLCT_ES	TMB4.LCT1.SYER	TMB4.LCT1.CLCT_ES
28	TMB4.LCT0.CSC_ID[0]	TMB4.LCT0.CLCT_BEND[0]	TMB4.LCT1.CSC_ID[0]	TMB4.LCT1.CLCT_BEND[0]
29	TMB4.LCT0.CSC_ID[1]	TMB4.LCT0.CLCT_BEND[1]	TMB4.LCT1.CSC_ID[1]	TMB4.LCT1.CLCT_BEND[1]
30	TMB4.LCT0.CSC_ID[2]	TMB4.LCT0.CLCT_BEND[2]	TMB4.LCT1.CSC_ID[2]	TMB4.LCT1.CLCT_BEND[2]
31	TMB4.LCT0.CSC_ID[3]	TMB4.LCT0.CLCT_BEND[3]	TMB4.LCT1.CSC_ID[3]	TMB4.LCT1.CLCT_BEND[3]
32	TMB1.LCT0.QUAL[0]		TMB1.LCT0.PAT_ID[0]	TMB1.LCT.PAT_ID[0]
33	TMB1.LCT0.QUAL[1]		TMB1.LCT0.PAT_ID[1]	TMB1.LCT.PAT_ID[1]
34	TMB1.LCT0.QUAL[2]		TMB1.LCT0.PAT_ID[2]	TMB1.LCT.PAT_ID[2]
35	TMB1.LCT0.QUAL[3]	TMB1.LCT0.CLCT_QS	TMB1.LCT0.PAT_ID[3]	TMB1.LCT.PAT_ID[3]
36	TMB1.LCT0.VPF		TMB1.LCT0.BC0	
37	CRC[0]		CRC[1]	

Table 10: GTP4 (MGTTX1.123) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB5.LCT0.CLCT_HS[0]		TMB5.LCT1.CLCT_HS[0]	
1	TMB5.LCT0.CLCT_HS[1]		TMB5.LCT1.CLCT_HS[1]	
2	TMB5.LCT0.CLCT_HS[2]		TMB5.LCT1.CLCT_HS[2]	
3	TMB5.LCT0.CLCT_HS[3]		TMB5.LCT1.CLCT_HS[3]	
4	TMB5.LCT0.CLCT_HS[4]		TMB5.LCT1.CLCT_HS[4]	
5	TMB5.LCT0.CLCT_HS[5]		TMB5.LCT1.CLCT_HS[5]	
6	TMB5.LCT0.CLCT_HS[6]		TMB5.LCT1.CLCT_HS[6]	
7	TMB5.LCT0.CLCT_HS[7]		TMB5.LCT1.CLCT_HS[7]	
8	TMB5.LCT0.ALCT_WG[0]		TMB5.LCT1.ALCT_WG[0]	
9	TMB5.LCT0.ALCT_WG[1]		TMB5.LCT1.ALCT_WG[1]	
10	TMB5.LCT0.ALCT_WG[2]		TMB5.LCT1.ALCT_WG[2]	
11	TMB5.LCT0.ALCT_WG[3]		TMB5.LCT1.ALCT_WG[3]	
12	TMB5.LCT0.ALCT_WG[4]		TMB5.LCT1.ALCT_WG[4]	
13	TMB5.LCT0.ALCT_WG[5]		TMB5.LCT1.ALCT_WG[5]	
14	TMB5.LCT0.ALCT_WG[6]		TMB5.LCT1.ALCT_WG[6]	
15	TMB5.LCT0.QUAL[0]		TMB5.LCT1.QUAL[0]	
16	TMB5.LCT0.QUAL[1]		TMB5.LCT1.QUAL[1]	
17	TMB5.LCT0.QUAL[2]		TMB5.LCT1.QUAL[2]	
18	TMB5.LCT0.QUAL[3]	TMB5.LCT0.CLCT_QS	TMB5.LCT1.QUAL[3]	TMB5.LCT1.CLCT_QS
19	TMB5.LCT0.CLCT_PAT_ID[0]	TMB5.LCT.CLCT_PAT_ID[0]	TMB5.LCT1.CLCT_PAT_ID[0]	TMB5.LCT.CLCT_PAT_ID[4]
20	TMB5.LCT0.CLCT_PAT_ID[1]	TMB5.LCT.CLCT_PAT_ID[1]	TMB5.LCT1.CLCT_PAT_ID[1]	TMB5.HMT[1]
21	TMB5.LCT0.CLCT_PAT_ID[2]	TMB5.LCT.CLCT_PAT_ID[2]	TMB5.LCT1.CLCT_PAT_ID[2]	TMB5.HMT[2]
22	TMB5.LCT0.CLCT_PAT_ID[3]	TMB5.LCT.CLCT_PAT_ID[3]	TMB5.LCT1.CLCT_PAT_ID[3]	TMB5.HMT[3]
23				
24	TMB5.LCT0.LR		TMB5.LCT1.LR	
25	TMB5.LCT0.VPF		TMB5.LCT1.VPF	
26	TMB5.LCT0.BC0		TMB5.LCT1.BC0	
27	TMB5.LCT0.SYER	TMB5.LCT0.CLCT_ES	TMB5.LCT1.BXN[0]	TMB5.HMT[0]
28	TMB5.LCT0.CSC_ID[0]	TMB5.LCT0.CLCT_BEND[0]	TMB5.LCT1.SYER	TMB5.LCT1.CLCT_ES
29	TMB5.LCT0.CSC_ID[1]	TMB5.LCT0.CLCT_BEND[1]	TMB5.LCT1.CSC_ID[0]	TMB5.LCT1.CLCT_BEND[0]
30	TMB5.LCT0.CSC_ID[2]	TMB5.LCT0.CLCT_BEND[2]	TMB5.LCT1.CSC_ID[1]	TMB5.LCT1.CLCT_BEND[1]
31	TMB5.LCT0.CSC_ID[3]	TMB5.LCT0.CLCT_BEND[3]	TMB5.LCT1.CSC_ID[2]	TMB5.LCT1.CLCT_BEND[2]
32	TMB1.LCT0.CSC_ID[0]	TMB1.LCT0.CLCT_BEND[0]	TMB5.LCT1.CSC_ID[3]	TMB5.LCT1.CLCT_BEND[3]
33	TMB1.LCT0.CSC_ID[1]	TMB1.LCT0.CLCT_BEND[1]		RESERVED
34	TMB1.LCT0.CSC_ID[2]	TMB1.LCT0.CLCT_BEND[2]		RESERVED
35	TMB1.LCT0.CSC_ID[3]	TMB1.LCT0.CLCT_BEND[3]	TMB1.LCT0.BXN[0]	TMB1.LCT0.BXN[0]
36	TMB1.LCT0.VPF		TMB1.LCT0.SYER	TMB1.LCT0.CLCT_ES
37	CRC[0]		TMB1.LCT0.BC0	TMB1.LCT0.BC0
			CRC[1]	

Table 11: GTP5 (MGTTX0.267) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB6.LCT0.CLCT_HS[0]		TMB6.LCT1.CLCT_HS[0]	
1	TMB6.LCT0.CLCT_HS[1]		TMB6.LCT1.CLCT_HS[1]	
2	TMB6.LCT0.CLCT_HS[2]		TMB6.LCT1.CLCT_HS[2]	
3	TMB6.LCT0.CLCT_HS[3]		TMB6.LCT1.CLCT_HS[3]	
4	TMB6.LCT0.CLCT_HS[4]		TMB6.LCT1.CLCT_HS[4]	
5	TMB6.LCT0.CLCT_HS[5]		TMB6.LCT1.CLCT_HS[5]	
6	TMB6.LCT0.CLCT_HS[6]		TMB6.LCT1.CLCT_HS[6]	
7	TMB6.LCT0.CLCT_HS[7]		TMB6.LCT1.CLCT_HS[7]	
8	TMB6.LCT0.ALCT_WG[0]		TMB6.LCT1.ALCT_WG[0]	
9	TMB6.LCT0.ALCT_WG[1]		TMB6.LCT1.ALCT_WG[1]	
10	TMB6.LCT0.ALCT_WG[2]		TMB6.LCT1.ALCT_WG[2]	
11	TMB6.LCT0.ALCT_WG[3]		TMB6.LCT1.ALCT_WG[3]	
12	TMB6.LCT0.ALCT_WG[4]		TMB6.LCT1.ALCT_WG[4]	
13	TMB6.LCT0.ALCT_WG[5]		TMB6.LCT1.ALCT_WG[5]	
14	TMB6.LCT0.ALCT_WG[6]		TMB6.LCT1.ALCT_WG[6]	
15	TMB6.LCT0.QUAL[0]		TMB6.LCT1.QUAL[0]	
16	TMB6.LCT0.QUAL[1]		TMB6.LCT1.QUAL[1]	
17	TMB6.LCT0.QUAL[2]		TMB6.LCT1.QUAL[2]	
18	TMB6.LCT0.QUAL[3]	TMB6.LCT0.CLCT_QS	TMB6.LCT1.QUAL[3]	TMB6.LCT1.CLCT_QS
19	TMB6.LCT0.CLCT_PAT_ID[0]	TMB6.LCT.CLCT_PAT_ID[0]	TMB6.LCT1.CLCT_PAT_ID[0]	TMB6.LCT.CLCT_PAT_ID[4]
20	TMB6.LCT0.CLCT_PAT_ID[1]	TMB6.LCT.CLCT_PAT_ID[1]	TMB6.LCT1.CLCT_PAT_ID[1]	TMB6.HMT[1]
21	TMB6.LCT0.CLCT_PAT_ID[2]	TMB6.LCT.CLCT_PAT_ID[2]	TMB6.LCT1.CLCT_PAT_ID[2]	TMB6.HMT[2]
22	TMB6.LCT0.CLCT_PAT_ID[3]	TMB6.LCT.CLCT_PAT_ID[3]	TMB6.LCT1.CLCT_PAT_ID[3]	TMB6.HMT[3]
23				
24	TMB6.LCT0.LR		TMB6.LCT1.LR	
25	TMB6.LCT0.VPF		TMB6.LCT1.VPF	
26	TMB6.LCT0.BC0		TMB6.LCT1.BC0	
27	TMB6.LCT0.BXN[0]		TMB6.LCT1.BXN[0]	TMB6.HMT[0]
28	TMB6.LCT0.SYER	TMB6.LCT0.CLCT_ES	TMB6.LCT1.SYER	TMB6.LCT1.CLCT_ES
29	TMB6.LCT0.CSC_ID[0]	TMB6.LCT0.CLCT_BEND[0]	TMB6.LCT1.CSC_ID[0]	TMB6.LCT1.CLCT_BEND[0]
30	TMB6.LCT0.CSC_ID[1]	TMB6.LCT0.CLCT_BEND[1]	TMB6.LCT1.CSC_ID[1]	TMB6.LCT1.CLCT_BEND[1]
31	TMB6.LCT0.CSC_ID[2]	TMB6.LCT0.CLCT_BEND[2]	TMB6.LCT1.CSC_ID[2]	TMB6.LCT1.CLCT_BEND[2]
32	TMB6.LCT0.CSC_ID[3]	TMB6.LCT0.CLCT_BEND[3]	TMB6.LCT1.CSC_ID[3]	TMB6.LCT1.CLCT_BEND[3]
33	TMB1.LCT1.CLCT_HS[0]		TMB1.LCT1.CLCT_HS[4]	
34	TMB1.LCT1.CLCT_HS[1]		TMB1.LCT1.CLCT_HS[5]	
35	TMB1.LCT1.CLCT_HS[2]		TMB1.LCT1.CLCT_HS[6]	
36	TMB1.LCT1.CLCT_HS[3]		TMB1.LCT1.CLCT_HS[7]	
37	TMB1.LCT1.VPF		TMB1.LCT1.BC0	
	CRC[0]		CRC[1]	

Table 12: GTP6 (MGTTX1.267) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB7.LCT0.CLCT_HS[0]		TMB7.LCT1.CLCT_HS[0]	
1	TMB7.LCT0.CLCT_HS[1]		TMB7.LCT1.CLCT_HS[1]	
2	TMB7.LCT0.CLCT_HS[2]		TMB7.LCT1.CLCT_HS[2]	
3	TMB7.LCT0.CLCT_HS[3]		TMB7.LCT1.CLCT_HS[3]	
4	TMB7.LCT0.CLCT_HS[4]		TMB7.LCT1.CLCT_HS[4]	
5	TMB7.LCT0.CLCT_HS[5]		TMB7.LCT1.CLCT_HS[5]	
6	TMB7.LCT0.CLCT_HS[6]		TMB7.LCT1.CLCT_HS[6]	
7	TMB7.LCT0.CLCT_HS[7]		TMB7.LCT1.CLCT_HS[7]	
8	TMB7.LCT0.ALCT.WG[0]		TMB7.LCT1.ALCT.WG[0]	
9	TMB7.LCT0.ALCT.WG[1]		TMB7.LCT1.ALCT.WG[1]	
10	TMB7.LCT0.ALCT.WG[2]		TMB7.LCT1.ALCT.WG[2]	
11	TMB7.LCT0.ALCT.WG[3]		TMB7.LCT1.ALCT.WG[3]	
12	TMB7.LCT0.ALCT.WG[4]		TMB7.LCT1.ALCT.WG[4]	
13	TMB7.LCT0.ALCT.WG[5]		TMB7.LCT1.ALCT.WG[5]	
14	TMB7.LCT0.ALCT.WG[6]		TMB7.LCT1.ALCT.WG[6]	
15	TMB7.LCT0.QUAL[0]		TMB7.LCT1.QUAL[0]	
16	TMB7.LCT0.QUAL[1]		TMB7.LCT1.QUAL[1]	
17	TMB7.LCT0.QUAL[2]		TMB7.LCT1.QUAL[2]	
18	TMB7.LCT0.QUAL[3]	TMB7.LCT0.CLCT_QS	TMB7.LCT1.QUAL[3]	TMB7.LCT1.CLCT_QS
19	TMB7.LCT0.CLCT.PAT.ID[0]	TMB7.LCT.CLCT.PAT.ID[0]	TMB7.LCT1.CLCT.PAT.ID[0]	TMB7.LCT.CLCT.PAT.ID[4]
20	TMB7.LCT0.CLCT.PAT.ID[1]	TMB7.LCT.CLCT.PAT.ID[1]	TMB7.LCT1.CLCT.PAT.ID[1]	TMB7.HMT[1]
21	TMB7.LCT0.CLCT.PAT.ID[2]	TMB7.LCT.CLCT.PAT.ID[2]	TMB7.LCT1.CLCT.PAT.ID[2]	TMB7.HMT[2]
22	TMB7.LCT0.CLCT.PAT.ID[3]	TMB7.LCT.CLCT.PAT.ID[3]	TMB7.LCT1.CLCT.PAT.ID[3]	TMB7.HMT[3]
23				
24	TMB7.LCT0.LR		TMB7.LCT1.LR	
25	TMB7.LCT0.VPF		TMB7.LCT1.VPF	
26	TMB7.LCT0.BC0		TMB7.LCT1.BC0	
27	TMB7.LCT0.BXN[0]		TMB7.LCT1.BXN[0]	TMB7.HMT[0]
28	TMB7.LCT0.SYER	TMB7.LCT0.CLCT_ES	TMB7.LCT1.SYER	TMB7.LCT1.CLCT_ES
29	TMB7.LCT0.CSC.ID[0]	TMB7.LCT0.CLCT.BEND[0]	TMB7.LCT1.CSC.ID[0]	TMB7.LCT1.CLCT.BEND[0]
30	TMB7.LCT0.CSC.ID[1]	TMB7.LCT0.CLCT.BEND[1]	TMB7.LCT1.CSC.ID[1]	TMB7.LCT1.CLCT.BEND[1]
31	TMB7.LCT0.CSC.ID[2]	TMB7.LCT0.CLCT.BEND[2]	TMB7.LCT1.CSC.ID[2]	TMB7.LCT1.CLCT.BEND[2]
32	TMB7.LCT0.CSC.ID[3]	TMB7.LCT0.CLCT.BEND[3]	TMB7.LCT1.CSC.ID[3]	TMB7.LCT1.CLCT.BEND[3]
33	TMB1.LCT1.ALCT.WG[0]		TMB1.LCT1.ALCT.WG[4]	
34	TMB1.LCT1.ALCT.WG[1]		TMB1.LCT1.ALCT.WG[5]	
35	TMB1.LCT1.ALCT.WG[2]		TMB1.LCT1.ALCT.WG[6]	
36	TMB1.LCT1.ALCT.WG[3]		TMB1.LCT1.CLCT_LR	
37	TMB1.LCT1.VPF		TMB1.LCT1.BC0	
	CRC[0]		CRC[1]	

Table 13: GTP7 (MGTTX0.245) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB8.LCT0.CLCT_HS[0]		TMB8.LCT1.CLCT_HS[0]	
1	TMB8.LCT0.CLCT_HS[1]		TMB8.LCT1.CLCT_HS[1]	
2	TMB8.LCT0.CLCT_HS[2]		TMB8.LCT1.CLCT_HS[2]	
3	TMB8.LCT0.CLCT_HS[3]		TMB8.LCT1.CLCT_HS[3]	
4	TMB8.LCT0.CLCT_HS[4]		TMB8.LCT1.CLCT_HS[4]	
5	TMB8.LCT0.CLCT_HS[5]		TMB8.LCT1.CLCT_HS[5]	
6	TMB8.LCT0.CLCT_HS[6]		TMB8.LCT1.CLCT_HS[6]	
7	TMB8.LCT0.CLCT_HS[7]		TMB8.LCT1.CLCT_HS[7]	
8	TMB8.LCT0.ALCT_WG[0]		TMB8.LCT1.ALCT_WG[0]	
9	TMB8.LCT0.ALCT_WG[1]		TMB8.LCT1.ALCT_WG[1]	
10	TMB8.LCT0.ALCT_WG[2]		TMB8.LCT1.ALCT_WG[2]	
11	TMB8.LCT0.ALCT_WG[3]		TMB8.LCT1.ALCT_WG[3]	
12	TMB8.LCT0.ALCT_WG[4]		TMB8.LCT1.ALCT_WG[4]	
13	TMB8.LCT0.ALCT_WG[5]		TMB8.LCT1.ALCT_WG[5]	
14	TMB8.LCT0.ALCT_WG[6]		TMB8.LCT1.ALCT_WG[6]	
15	TMB8.LCT0.QUAL[0]		TMB8.LCT1.QUAL[0]	
16	TMB8.LCT0.QUAL[1]		TMB8.LCT1.QUAL[1]	
17	TMB8.LCT0.QUAL[2]		TMB8.LCT1.QUAL[2]	
18	TMB8.LCT0.QUAL[3]	TMB8.LCT0.CLCT_QS	TMB8.LCT1.QUAL[3]	TMB8.LCT1.CLCT_QS
19	TMB8.LCT0.CLCT_PAT_ID[0]	TMB8.LCT.CLCT_PAT_ID[0]	TMB8.LCT1.CLCT_PAT_ID[0]	TMB8.LCT.CLCT_PAT_ID[4]
20	TMB8.LCT0.CLCT_PAT_ID[1]	TMB8.LCT.CLCT_PAT_ID[1]	TMB8.LCT1.CLCT_PAT_ID[1]	TMB8.HMT[1]
21	TMB8.LCT0.CLCT_PAT_ID[2]	TMB8.LCT.CLCT_PAT_ID[2]	TMB8.LCT1.CLCT_PAT_ID[2]	TMB8.HMT[2]
22	TMB8.LCT0.CLCT_PAT_ID[3]	TMB8.LCT.CLCT_PAT_ID[3]	TMB8.LCT1.CLCT_PAT_ID[3]	TMB8.HMT[3]
23				
24	TMB8.LCT0.LR		TMB8.LCT1.LR	
25	TMB8.LCT0.VPF		TMB8.LCT1.VPF	
26	TMB8.LCT0.BC0		TMB8.LCT1.BC0	
27	TMB8.LCT0.BXN[0]		TMB8.LCT1.BXN[0]	TMB8.HMT[0]
28	TMB8.LCT0.SYER	TMB8.LCT0.CLCT_ES	TMB8.LCT1.SYER	TMB8.LCT1.CLCT_ES
29	TMB8.LCT0.CSC_ID[0]	TMB8.LCT0.CLCT_BEND[0]	TMB8.LCT1.CSC_ID[0]	TMB8.LCT1.CLCT_BEND[0]
30	TMB8.LCT0.CSC_ID[1]	TMB8.LCT0.CLCT_BEND[1]	TMB8.LCT1.CSC_ID[1]	TMB8.LCT1.CLCT_BEND[1]
31	TMB8.LCT0.CSC_ID[2]	TMB8.LCT0.CLCT_BEND[2]	TMB8.LCT1.CSC_ID[2]	TMB8.LCT1.CLCT_BEND[2]
32	TMB8.LCT0.CSC_ID[3]	TMB8.LCT0.CLCT_BEND[3]	TMB8.LCT1.CSC_ID[3]	TMB8.LCT1.CLCT_BEND[3]
33	TMB1.LCT1.QUAL[0]		TMB1.LCT1.PAT_ID[0]	TMB1.LCT.PAT_ID[4]
34	TMB1.LCT1.QUAL[1]		TMB1.LCT1.PAT_ID[1]	TMB1.HMT[1]
35	TMB1.LCT1.QUAL[2]		TMB1.LCT1.PAT_ID[2]	TMB1.HMT[2]
36	TMB1.LCT1.QUAL[3]	TMB1.LCT0.CLCT_QS	TMB1.LCT1.PAT_ID[3]	TMB1.HMT[3]
37	TMB1.LCT1.VPF		TMB1.LCT1.BC0	
	CRC[0]		CRC[1]	

Table 14: GTP8 (MGTTX1.245) Bits that remain the same in Run-2 and Run-3 are shown in black. Bits that change meaning in Run-3 are shown in blue.

Bit	Frame 0		Frame 1	
	Run-2	Run-3	Run-2	Run-3
0	TMB9_LCT0.CLCT_HS[0]		TMB9_LCT1.CLCT_HS[0]	
1	TMB9_LCT0.CLCT_HS[1]		TMB9_LCT1.CLCT_HS[1]	
2	TMB9_LCT0.CLCT_HS[2]		TMB9_LCT1.CLCT_HS[2]	
3	TMB9_LCT0.CLCT_HS[3]		TMB9_LCT1.CLCT_HS[3]	
4	TMB9_LCT0.CLCT_HS[4]		TMB9_LCT1.CLCT_HS[4]	
5	TMB9_LCT0.CLCT_HS[5]		TMB9_LCT1.CLCT_HS[5]	
6	TMB9_LCT0.CLCT_HS[6]		TMB9_LCT1.CLCT_HS[6]	
7	TMB9_LCT0.CLCT_HS[7]		TMB9_LCT1.CLCT_HS[7]	
8	TMB9_LCT0.ALCT_WG[0]		TMB9_LCT1.ALCT_WG[0]	
9	TMB9_LCT0.ALCT_WG[1]		TMB9_LCT1.ALCT_WG[1]	
10	TMB9_LCT0.ALCT_WG[2]		TMB9_LCT1.ALCT_WG[2]	
11	TMB9_LCT0.ALCT_WG[3]		TMB9_LCT1.ALCT_WG[3]	
12	TMB9_LCT0.ALCT_WG[4]		TMB9_LCT1.ALCT_WG[4]	
13	TMB9_LCT0.ALCT_WG[5]		TMB9_LCT1.ALCT_WG[5]	
14	TMB9_LCT0.ALCT_WG[6]		TMB9_LCT1.ALCT_WG[6]	
15	TMB9_LCT0.QUAL[0]		TMB9_LCT1.QUAL[0]	
16	TMB9_LCT0.QUAL[1]		TMB9_LCT1.QUAL[1]	
17	TMB9_LCT0.QUAL[2]		TMB9_LCT1.QUAL[2]	
18	TMB9_LCT0.QUAL[3]	TMB9_LCT0.CLCT_QS	TMB9_LCT1.QUAL[3]	TMB9_LCT1.CLCT_QS
19	TMB9_LCT0.CLCT_PAT_ID[0]	TMB9_LCT.CLCT_PAT_ID[0]	TMB9_LCT1.CLCT_PAT_ID[0]	TMB9_LCT.CLCT_PAT_ID[4]
20	TMB9_LCT0.CLCT_PAT_ID[1]	TMB9_LCT.CLCT_PAT_ID[1]	TMB9_LCT1.CLCT_PAT_ID[1]	TMB9_HMT[1]
21	TMB9_LCT0.CLCT_PAT_ID[2]	TMB9_LCT.CLCT_PAT_ID[2]	TMB9_LCT1.CLCT_PAT_ID[2]	TMB9_HMT[2]
22	TMB9_LCT0.CLCT_PAT_ID[3]	TMB9_LCT.CLCT_PAT_ID[3]	TMB9_LCT1.CLCT_PAT_ID[3]	TMB9_HMT[3]
23				
24	TMB9_LCT0.LR		TMB9_LCT1.LR	
25	TMB9_LCT0.VPF		TMB9_LCT1.VPF	
26	TMB9_LCT0.BC0		TMB9_LCT1.BC0	
27	TMB9_LCT0.BXN[0]		TMB9_LCT1.BXN[0]	TMB9_HMT[0]
28	TMB9_LCT0.SYER	TMB9_LCT0.CLCT_ES	TMB9_LCT1.SYER	TMB9_LCT1.CLCT_ES
29	TMB9_LCT0.CSC_ID[0]	TMB9_LCT0.CLCT_BEND[0]	TMB9_LCT1.CSC_ID[0]	TMB9_LCT1.CLCT_BEND[0]
30	TMB9_LCT0.CSC_ID[1]	TMB9_LCT0.CLCT_BEND[1]	TMB9_LCT1.CSC_ID[1]	TMB9_LCT1.CLCT_BEND[1]
31	TMB9_LCT0.CSC_ID[2]	TMB9_LCT0.CLCT_BEND[2]	TMB9_LCT1.CSC_ID[2]	TMB9_LCT1.CLCT_BEND[2]
32	TMB9_LCT0.CSC_ID[3]	TMB9_LCT0.CLCT_BEND[3]	TMB9_LCT1.CSC_ID[3]	TMB9_LCT1.CLCT_BEND[3]
33	TMB1_LCT1.CSC_ID[0]	TMB1_LCT0.CLCT_BEND[0]		RESERVED
34	TMB1_LCT1.CSC_ID[1]	TMB1_LCT0.CLCT_BEND[1]		RESERVED
35	TMB1_LCT1.CSC_ID[2]	TMB1_LCT0.CLCT_BEND[2]	TMB1_LCT1.BXN[0]	TMB1_HMT[0]
36	TMB1_LCT1.CSC_ID[3]	TMB1_LCT0.CLCT_BEND[3]	TMB1_LCT1.SYER	TMB1_LCT1.CLCT_ES
37	TMB1_LCT1.VPF		TMB1_LCT1.BC0	
	CRC[0]		CRC[1]	

4 GEM Trigger Data Formats

4.1 Cluster Format

The trigger path of the GEM detectors supports a phi resolution of two strips (two VFAT channels on the GEM readout board). These are trigger pads (s-bits on the GEM readout board). The frontend (optohybrid) compresses up to 8 neighboring pads into clusters, which are the GE1/1 and GE2/1 trigger primitives. Thus the cluster encodes the eta and phi coordinate of the beginning of the cluster, and the cluster size (the number of hit pads).

The global coordinate definition at CMS P5 is as follow: (1) Increasing pad number counter clock-wise in the negative end-cap, (2) Increasing pad number clock-wise in the positive end-cap. This is regardless of the chamber orientation and follows the CSC and GEM chamber ordering in CMS.

There are three versions of the cluster data format, explained below. Each of these follows the eta partition mapping in the table below

Firmware Eta Partition Number (Roll Number)	Simulation Eta Partition Number (CMS Eta Partition Number)
0	8
1	7
2	6
3	5
4	4
5	3
6	2
7	1

4.1.1 Version 1

The format of the GE1/1 cluster is shown in Tab. 15. Clusters have 14 bits each. GE2/1 cluster will not use this format.

Table 15: GE1/1 cluster data format. A null cluster will have bits [13:0] set to 1.

Bits[13:11]	Bits[10:0]
Cluster size 1-8 (encoded as 0-7)	Cluster address 0 - 1535 Pad # > 1535 is invalid

4.1.2 Version 2

The format of each cluster is shown in Tab. 16 and Tab. 17. Clusters have 14 bits each.

Table 16: GE1/1 cluster data format. A null cluster will have bits [13:0] set to 1. Pad values between 192 and 254 are reserved for future use.

Bits[13:11]	Bits[10:8]	Bits[7:0]
Cluster size 1-8 (encoded as 0-7)	Eta partition 0-7	Pad 0-191 Pad # 255 is invalid

4.1.3 Version 3

The format of each cluster is shown in Tab. 18 and Tab. 19. Clusters have 16 bits each.

Table 17: GE2/1 cluster data format. A null cluster will have bits [12:0] set to 1. Pad values between 384 and 510 are reserved for future use.

Bits[12:10]	Bits[9]	Bits[8:0]
Cluster size 1-8 (encoded as 0-7)	Eta partition 0-1	Pad 0-383 Pad # 511 is invalid

Table 18: GE1/1 cluster data format. A null cluster will have bits [13:0] set to 1. Pad values between 192 and 254 are reserved for future use.

Bits[15]	Bits[14]	Bits[13:11]	Bits[10:8]	Bits[7:0]
Last BX overflow	Cluster status	Cluster size 1-8 (encoded as 0-7)	Eta partition 0-7	Pad 0-191 Pad # 255 is invalid

Table 19: GE2/1 cluster data format. A null cluster will have bits [12:0] set to 1. Pad values between 384 and 510 are reserved for future use.

Bits[15]	Bits[14]	Bits[13]	Bits[12:10]	Bits[9]	Bits[8:0]
Last BX overflow	Cluster status	Unused	Cluster size 1-8 (encoded as 0-7)	Eta partition 0-1	Pad 0-383 Pad # 511 is invalid

4.2 OH to GEM Backend and to OTMB Link Format

Both GE1/1 and GE2/1 OHs use 8b10b links to CSC OTMB:

- GE1/1 has 2 such links per OH
- GE2/1 has 1 link per OH on modules M2, M3, M4, M6, M7, M8, and 2 links per OH in M1 and M5 modules

4.2.1 Version 1

For GE1/1 the trigger link to backend is the same as to the CSC OTMB: $2 \times 8b10b$ links per OH running at 3.2 Gb/s. This proposal considers running these links at 3.2 Gb/s, giving 64 bits/bx per link, or 4×14 -bit words and 8 bits for frame marking. Each of these 14-bit words can contain a cluster (called CL_WORD), thus giving a total bandwidth of 4 clusters per BX. See Tab. 20.

Table 20: GE1/1-ME1/1 Link at 3.2 Gb/s 8b10b.

Bits[63:50]	Bits[49:36]	Bits[35:22]	Bits[21:8]	Bits[7:0]
CL_WORD3	CL_WORD2	CL_WORD1	CL_WORD0	Frame Marker

The frame marker is a comma character that cycles from

- 1C = BC0
- 3C = Resync
- FC = Cluster overflow
- BC = BXN[1:0]==0
- F7 = BXN[1:0]==1
- FB = BXN[1:0]==2
- FD = BXN[1:0]==3

The priority of the comma characters is BC0 > Resync > Overflow > BXN. There is no ECC (unlike in Version 3).

4.2.2 Version 2

This is the same as Version 1, except that the cluster word (CL_WORD_i) takes format as in Sec. 4.1.2 instead of Sec. 4.1.1.

4.2.3 Version 3

For GE1/1 the trigger link to backend is the same as to the CSC OTMB: $2 \times 8b10b$ links per OH running at 4 Gb/s. For GE2/1 the link to backend is different, and is described later in this section.

This proposal considers running these links at 4 Gb/s, giving 80 bits/bx per link, or 5×16 -bit words. Each of these 16-bit words can contain a cluster (called CL_WORD), thus giving a total bandwidth of 5 clusters per BX, but whenever there are less than 5 clusters available in a given BX, an ECC code and a comma word are sent in place of the 5th cluster. See Tab. 21.

The proposed error correcting code is a simple Hamming correction code and is described in Xilinx XAPP645. This is the same error correction scheme already used by the TMB + ALCT. This allows for single bit error correction with 64 bits of data + 8 bits of parity, as well as

Table 21: GE1/1-ME1/1 and GE2/1-ME2/1 Link at 4 Gb/s 8b10b. Note: the comma character 0xBC will normally be transmitted every bx, except in the case of BC0 when instead the character 0xDC will be sent.

Bits[79:72]	Bits[71:64]	Bits[63:48]	Bits[47:32]	Bits[31:16]	Bits[15:0]
Comma/bc0	ECC8	CL_WORD3	CL_WORD2	CL_WORD1	CL_WORD0
CL_WORD4					

329 double bit error detection. See https://www.xilinx.com/support/documentation/application_notes/xapp645.pdf for more information.

331

332 Whenever the number of clusters reaches the limit of the bandwidth provided by CL_WORD0
333 - CL_WORD3 (8 clusters in 2 link OHs, and 4 cluster in 1 link OHs), the CL_WORD4 is used,
334 and replaces the ECC8 + Comma/bc0 word, however a maximum delay of 100 BXs is guaran-
335 teed between consecutive comma characters (the number 100 can be tuned later). Only empty
336 clusters are sent for 4 orbits following a resync signal, thus guaranteeing that the comma/bc0
337 symbols will not be replaced by CL_WORD4 during this time.

338

339 On the 8b10b link we will transmit the 16 bit data words at 200 MHz in the following or-
340 der: CL_WORD0 → CL_WORD1 → CL_WORD2 → CL_WORD3 → CL_WORD4 / ECC8 [7:0] +
341 Comma/BC0 [15:8]. The 16bit words are sent from LSB to MSB.

342

343 The CL_WORD format includes the cluster itself, described section 4.1, and two additional
344 bits, used for link status, and cluster transmission from previous BX:

Table 22: CL_WORD format. Note: Bit 13 is reserved in GE2/1 case.

Bit[15]	Bit[14]	Bits[13:0]
0 = cluster is from this bx	status_bit [n]	CLUSTER
1 = cluster from previous bx	(see status bit section)	(see cluster format section)

345 The clusters that do not fit into the link bandwidth during bunch crossing N can be transmitted
346 in the bunch crossing N+1 (the following BX). Clusters from the previous bx will always be
347 inserted after the valid clusters from the current bx. This introduces latency since the processor
348 must wait 1 additional bx to see all of the trigger data that can be sent from this bx, but it ef-
349 fectively increases the bandwidth by up to a factor of 2. This feature can be made configurable
350 on the receiving side, and be turned on or off, depending on the latency and bandwidth needs.
351 This capability is not yet implemented in the OH firmware, so bit 15 is always set to 0.

352

353 For the meaning of the status bits, please see the section 4.4.

354

355 The order in which the clusters are filled is defined in the following two tables. As mentioned
356 before, the clusters from the current BX always take priority and have a lower index than the
357 cluster from the previous BX. This is shown in Tab. 23 and Tab. 24

Table 23: Cluster order in OH with one link

	CL_WORD0	CL_WORD1	CL_WORD2	CL_WORD3	CL_WORD4
Cluster index	0	1	2	3	4

358 For GE2/1, the trigger data to the backend are included in the 2 existing GBT links. Eleven
359 dedicated e-links (8b10b) operate at 8 bits/bx (320 Mb/s each), totalling 3.52 Gb/s of trigger
360 data. Eight of the 11 e-links are on GBT1 which operates in wide-bus mode, so the data on these

Table 24: Cluster order in OH with two links

	CL_WORD0	CL_WORD1	CL_WORD2	CL_WORD3	CL_WORD4
Cluster index on link 0	0	1	2	3	8
Cluster index on link 1	4	5	6	7	9

361 e-links is not FEC protected. Three of the 11 e-links are on GBT0 which operates in normal
 362 mode with FEC protection. GBT1 covers bits [63:0] and GBT0 covers bits [87:64]. This is shown
 363 in Tab. 25.

Table 25: GE2/1 (GBT Link to backend)

Bits[87:80]	Bits[79:64]	Bits[63:48]	Bits[47:32]	Bits[31:16]	Bits[15:0]
ECC8	CL_WORD4	CL_WORD3	CL_WORD2	CL_WORD1	CL_WORD0

364 The ECC8 is calculated on CLUSTER0-CLUSTER3, while CLUSTER4 and ECC8 are protected
 365 by the GBTX FEC.

DRAFT

366 4.3 OptoHybrid Metadata

367 Spare bits of null clusters may be used to encode chamber *metadata* to uniquely identify the
 368 chamber. The metadata would be programmed into the OH by the backend, and transmitted
 369 in the trigger links to the OTMB and GEM backend (and also sent to the EMTF). Null clusters
 370 in GE1/1 are represented by 0b11XXXXXX, leaving six “Do Not Care” bits. Null clusters in
 371 GE2/1 are represented by 0b11XXXXXX, leaving six “Do Not Care” bits. This means that for
 372 null clusters, we can use the least significant six bits (GE1/1) or seven bits (GE2/1) as fields
 373 to carry data. Furthermore, the 3 bits of size and 1 (GE2/1) or 3 (GE1/1) bits of partition are
 374 irrelevant for null clusters, so they can be freely used as data fields also. This means that for a
 375 null cluster, in total we have 12 bits free per cluster, per bx, which can be used as metadata.

```
376 ge11_metadata = size[2:0] & partition [2:0] & pad[5:0]
377 ge21_metadata = unused[0] & size[2:0] & partition [0] & pad[6:0]
```

378 The tables below show the implementation of the metadata format, which is applicable to both
 379 version 2 and version 3 cluster data formats.

Table 26: GE1/1 Metadata Format

Field	Num Bits	Meta Bits	Notes
Endcap	1	[0]	0=minus, 1=plus
Station	1	[1]	0=GE1/1, 1=GE2/1
Layer	1	[2]	Layer 0 or 1
Chamber Number	6	[8:3]	Chamber 0-35
Reserved	1	[9]	-
Link ID	1	[10]	0=Fiber link 0, 1=Fiber link 1
Metadata Valid	1	[11]	1=Metadata has been programmed
Total	12		

Table 27: GE2/1 Metadata Format

Field	Num Bits	Meta Bits	Notes
Endcap	1	[0]	0=minus, 1=plus
Station	1	[1]	0=GE1/1, 1=GE2/1
Layer	1	[2]	Layer 0 or 1
Chamber Number	6	[7:3]	Chamber 0-17
Module Number	6	[9:8]	Module 0-3 (0=M1/5, 1=M2/6, 2=M3/7, 3=M4/8)
Link ID	1	[10]	0=Fiber link 0, 1=Fiber link 1
Metadata Valid	1	[11]	1=Metadata has been programmed
Total	12		

380 Redundant metadata would be sent on all idle clusters, so that a chamber can be uniquely
 381 identified from even a single cluster stream, so that as data is merged onto common links
 382 from the GEM backend to the EMTF, the source of the data can still be easily confirmed. The
 383 metadata must be programmed into the Optohybrid by the backend during initialization. For
 384 uninitialized fields, they will default to all zero and the “Metadata Valid” flag will also be set
 385 to zero.

386 4.4 GEM Status bits

387 The status bit for cluster 0 is dedicated as a BC0:

```
388 bc0 = CL_WORD0[status_bit]
```

389 The remaining status bits carried with each cluster should be concatenated together into a 3 bit
390 number:

```
391 status[3:1] = {CL_WORD3[status_bit], CL_WORD2[status_bit],
392               CL_WORD1[status_bit]}
```

393 The status bit for CL.WORD4 is reserved and not used for anything at the moment. This field
394 should be decoded according Tab. 28.

Table 28: GEM status bit table

Status[3:1]	Meaning
3'h0	BXN[1:0]==2'h0
3'h1	BXN[1:0]==2'h1
3'h2	BXN[1:0]==2'h2
3'h3	BXN[1:0]==2'h3
3'h4	Overflow
3'h5	Resync
3'h6	Reserved
3'h7	Error

395 The priority of the status words is such that higher valued status words will always take prece-
396 dence over lower valued (e.g. error > resync > overflow > bxn). The overflow flag will come
397 in the next bunch crossing, $n + 1$, and indicates that the number of clusters found was greater
398 than the number of clusters transmitted in $bx=n$ along with those sent in $bx= n + 1$.

399 4.5 GEM Backend to EMTF Link Format

400 The GE1/1 backend (CTP7) to EMTF links are running at 10.24 Gb/s using the LpGBT ASIC
 401 protocol with FEC5, so the number of data bits per BX is 234. Each output link to EMTF will
 402 combine trigger data from one GE1/1 superchamber (2 OptoHybrids, 4 input links total).

403
 404 There are two format modes, indicated by bit[1]:

- 405 • Normal mode (format 0) – sending trigger data
- 406 • Metadata mode (format 1) – used for checking the link mapping

407 In both modes bit[0] is dedicated to BC0. Note that after a resync, the BC0 bit remains low until
 408 the input links from the frontend are aligned, this can take up to 3 orbits.

409 4.5.1 Metadata mode (format 1)

410 The links operate in metadata mode for 1 orbit following the Resync BGo. The BGo are TTC
 411 commands sent through the CMS TCDS system. The Resync BGo commands aim at re-synchronizing
 412 front-end when an out-of-sync event is detected. Currently it is only used to transmit link ID,
 413 so that link mapping can be checked, but there are a lot of spare bits for future use.

Table 29: Format 1 (metadata)

Bits[233:9]	Bits[8:2]	Bit[1]	Bit[0]
Reserved	Link ID	1 (constant)	BC0

Table 30: Link ID

Bit[6]	Bits[5:0]
Endcap: 0 means positive endcap 1 means negative endcap	Super-chamber ID (1-36) E.g. GE+1/1/23 would have this field set to 23

414 4.5.2 Normal mode (format 0)

415 After the BC0 (bit[0]) and the mode flag (bit[1]), there is a 4-bit field for each of the super-
 416 chamber layers, indicating how many of the clusters being currently transmitted (0-8) are from
 417 the current BX – any valid clusters starting at this position are from the previous BX. E.g. if
 418 this field is set to 5, it means that cluster0 - cluster4 are from the current BX, and if any of the
 419 cluster5 - cluster7 are valid, they are from the previous BX. The control bits are shown in Tab. 31.

420
 421 After the control bits, there are two 112 bit wide sections transmitting 8 clusters for each of
 422 the layers of the super-chamber. This is shown in Tab. 32 and Tab. 33.

Table 31: Control bits

Bits[9:6]	Bits[5:2]	Bit[1]	Bit[0]
Super-chamber layer 2: number of valid clusters from current BX (0-8). Any valid clusters starting at this position are from previous BX	Super-chamber layer 1: number of valid clusters from current BX (0-8). Any valid clusters starting at this position are from previous BX	0 (constant)	BC0

423 A special value in place of cluster0 could be used to signal high multiplicity trigger data in
 424 clusters 1-7. Exact format for that is not part of this specification.

Table 32: Clusters from super-chamber layer 1

Bits[65:52]	Bits[51:38]	Bits[37:24]	Bits[23:10]
Cluster3	Cluster2	Cluster1	Cluster0
Bits[121:108]	Bits[107:94]	Bits[93:80]	Bits[79:66]
Cluster7	Cluster6	Cluster5	Cluster4

Table 33: Clusters from super-chamber layer 2

Bits[177:164]	Bits[163:150]	Bits[149:136]	Bits[135:122]
Cluster3	Cluster2	Cluster1	Cluster0
Bits[233:220]	Bits[219:206]	Bits[205:192]	Bits[191:178]
Cluster7	Cluster6	Cluster5	Cluster4

425

426 The EMTF link on the ATCA hardware will run at 25 Gb/s. The exact protocol is not yet
 427 defined, but each GE2/1 output link to EMTF will carry trigger data from one GE2/1 super-
 428 chamber, and each ME0 output link to EMTF will carry trigger data from one ME0 stack.

DRAFT

429 5 Implementation of the Data Formats in CMSSW

430 The CSC and GEM trigger data formats described in the previous sections have been imple-
 431 mented in CMSSW. These are: CSCALCTDigi, CSCCLCTDigi, CSCCorrelatedLCTDigi,
 432 GEMPadDigi and GEMPadDigiCluster. The former three have been used since CMS Run-1.
 433 The latter two were introduced for Run-3 and Phase-2 studies.

434 5.1 CSCALCTDigi

435 These enums and types were added

```
436 enum class Version { Legacy = 0, Run3 };
437
438 typedef std::vector<std::vector<uint16_t>> WireContainer;
439
```

440 The ALCT format is given extra members

```
441 /// Run-3 introduces high-multiplicity bits for CSCs.
442 /// Note: In DN-20-016, 3 bits are allocated for HMT in the
443 /// ALCT board. These bits are copied into the ALCT digi in CMSSW
444 uint16_t hmt_;
445
446 Version version_;
447 // which hits are in this ALCT?
448 WireContainer hits_;
```

449 and extra functions

```
450 /// return the high multiplicity bits
451 uint16_t getHMT() const;
452
453 /// set the high multiplicity bits
454 void setHMT(const uint16_t hmt);
455
456 /// Distinguish Run-1/2 from Run-3
457 bool isRun3() const { return version_ == Version::Run3; }
458
459 void setRun3(const bool isRun3);
460
461 // wire hits in this ALCT
462 const WireContainer& getHits() const { return hits_; }
463
464 void setHits(const WireContainer& hits) { hits_ = hits; }
```

465 5.2 CSCCLCTDigi

466 These enums and types were added

```
467 typedef std::vector<std::vector<uint16_t>> ComparatorContainer;
468
469 enum CLCTKeyStripMasks { kEightStripMask = 0x1, kQuartStripMask = 0x1, kHalfStripMask = 0x1f };
470 enum CLCTKeyStripShifts { kEightStripShift = 6, kQuartStripShift = 5, kHalfStripShift = 0 };
471 // temporary to facilitate CCLUT-EMTF/OMTF integration studies
472 enum CLCTPatternMasks { kRun3SlopeMask = 0xf, kRun3PatternMask = 0x7, kLegacyPatternMask = 0xf };
473 enum CLCTPatternShifts { kRun3SlopeShift = 7, kRun3PatternShift = 4, kLegacyPatternShift = 0 };
474 enum class Version { Legacy = 0, Run3 };
475
```

476 Two data members were added to the format:


```

477 // new in Run-3: 12-bit comparator code
478 // set by default to -1 for Run-1 and Run-2 CLCTs
479 int16_t compCode_;
480 // which hits are in this CLCT?
481 ComparatorContainer hits_;
482
483 Version version_;
484

```

485 The strip_members's interpretation was extended:

```

486 // In Run-3, the strip number receives two additional bits
487 // strip[4:0] -> 1/2 strip value
488 // strip[5]   -> 1/4 strip bit
489 // strip[6]   -> 1/8 strip bit
490 uint16_t strip_;

```

491 These functions were added:

```

492 /// Distinguish Run-1/2 from Run-3
493 bool isRun3() const { return version_ == Version::Run3; }
494
495 void setRun3(bool isRun3);
496
497 /// return pattern
498 uint16_t getRun3Pattern() const;
499
500 /// set pattern
501 void setRun3Pattern(const uint16_t pattern);
502
503 // 12-bit comparator code
504 int16_t getCompCode() const { return (isRun3() ? compCode_ : -1); }
505
506 void setCompCode(const int16_t code) { compCode_ = code; }
507
508 // comparator hits in this CLCT
509 const ComparatorContainer& getHits() const { return hits_; }
510
511 void setHits(const ComparatorContainer& hits) { hits_ = hits; }
512
513 /// return the slope
514 uint16_t getSlope() const;
515
516 /// set the slope
517 void setSlope(const uint16_t slope);
518
519 /// set single quart strip bit
520 void setQuartStrip(const bool quartStrip);
521
522 /// get single quart strip bit
523 bool getQuartStrip() const;
524
525 /// set single eight strip bit
526 void setEightStrip(const bool eightStrip);
527
528 /// get single eight strip bit
529 bool getEightStrip() const;

```

530 The meaning of these functions was extended

```

531 /// Convert strip_ and cfep_ to keyStrip. Each CFEB has up to 16 strips
532 /// (32 halfstrips). There are 5 cfeps. The "strip_" variable is one
533 /// of 32 halfstrips on the keylayer of a single CFEB, so that
534 /// Halfstrip = (cfep*32 + strip).
535 /// This function can also return the quartstrip or eightstrip

```

```

536 // when the comparator code has been set
537 uint16_t getKeyStrip(const uint16_t n = 2) const;
538
539 /*
540 Strips are numbered starting from 1 in CMSSW
541 Half-strips, quarter-strips and eighth-strips are numbered starting from 0
542 The table below shows the correct numbering
543 -----
544 strip      |           1           |           2           |
545 -----
546 1/2-strip |     0     |     1     |     2     |     3     |
547 -----
548 1/4-strip |  0  |  1  |  2  |  3  |  4  |  5  |  6  |  7  |
549 -----
550 1/8-strip | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
551 -----
552 Note: the CSC geometry also has a strip offset of +/- 0.25 strips. When comparing the
553 CLCT/LCT position with the true muon position, take the offset into account!
554 */
555 float getFractionalStrip(const uint16_t n = 2) const;
556
557

```

5.3 CSCCorrelatedLCTDigi

558 These enums and types were added

```

560 enum LCTKeyStripMasks { kEightStripMask = 0x1, kQuartStripMask = 0x1, kHalfStripMask = 0xff };
561 enum LCTKeyStripShifts { kEightStripShift = 9, kQuartStripShift = 8, kHalfStripShift = 0 };
562 // temporary to facilitate CCLUT-EMTF/OMTF integration studies
563 enum LCTPatternMasks { kRun3SlopeMask = 0xf, kRun3PatternMask = 0x7, kLegacyPatternMask = 0xf };
564 enum LCTPatternShifts { kRun3SlopeShift = 7, kRun3PatternShift = 4, kLegacyPatternShift = 0 };
565 enum class Version { Legacy = 0, Run3 };
566

```

567 This member was added

```

568 // In Run-3, LCT data will be carrying the high-multiplicity bits
569 // for chamber. These bits may indicate the observation of "exotic" events
570 // Depending on the chamber type 2 or 3 bits will be repurposed
571 // in the 32-bit LCT data word from the synchronization bit and
572 // quality bits.
573 uint16_t hmt;
574
575 Version version_;

```

576 These functions were added:

```

577 // set single quart strip bit
578 void setQuartStrip(const bool quartStrip);
579
580 // get single quart strip bit
581 bool getQuartStrip() const;
582
583 // set single eight strip bit
584 void setEightStrip(const bool eightStrip);
585
586 // get single eight strip bit
587 bool getEightStrip() const;
588
589 // set pattern
590 void setRun3Pattern(const uint16_t pattern);
591
592 // return pattern
593 uint16_t getRun3Pattern() const;

```

```

594
595 // Distinguish Run-1/2 from Run-3
596 bool isRun3() const { return version_ == Version::Run3; }
597
598 void setRun3(const bool isRun3);
599
600 // set high-multiplicity bits
601 void setHMT(const uint16_t h);
602
603 // Run-3 introduces high-multiplicity bits for CSCs.
604 // The allocation is different for ME1/1 and non-ME1/1
605 // chambers. Both LCTs in a chamber are needed for the complete
606 // high-multiplicity trigger information
607 uint16_t getHMT() const;
608
609

```

610 The interpretation of these functions was expanded

```

611 /*
612 Strips are numbered starting from 1 in CMSSW
613 Half-strips, quarter-strips and eighth-strips are numbered starting from 0
614 The table below shows the correct numbering
615 -----
616 strip      |           1           |           2           |
617 -----
618 1/2-strip  |     0     |     1     |     2     |     3     |
619 -----
620 1/4-strip  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
621 -----
622 1/8-strip  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
623 -----
624 Note: the CSC geometry also has a strip offset of +/- 0.25 strips. When comparing the
625 CLCT/LCT position with the true muon position, take the offset into account!
626 */
627 float getFractionalStrip(uint16_t n = 2) const;
628
629
630 // return the key halfstrip from 0,159
631 uint16_t getStrip(uint16_t n = 2) const;
632

```

633 5.4 GEMPadDigi

634 These enums and members were added

```

635 enum InValid { ME0InValid = 255, GE11InValid = 255, GE21InValid = 511 };
636 GEMSubDetId::Station station_;
637

```

638 and these functions

```

639 // only depends on the "InValid" enum so it also
640 // works on unpacked data
641 bool isValid() const;
642
643 GEMSubDetId::Station station() const { return station_; }
644

```

645 5.5 GEMPadDigiCluster

646 These enums and members were added

```
647 enum InValid { GE11InValid = 255, GE21InValid = 511 };
648 GEMSubDetId::Station station_;
649
```

650 and these functions

```
651 // only depends on the "InValid" enum so it also
652 // works on unpacked data
653 bool isValid() const;
654
655 GEMSubDetId::Station station() const { return station_; }
656
```

DRAFT

6 Summary

New trigger data formats for GE1/1 and CSC have been described for Run-3 data taking. Trigger data formats for GE2/1 are provided as well. Changes for CSCs include a more precise position and bending based on the CCLUT algorithm, new quality definitions for LCTs with and without the GEM-CSC integrated local trigger, and new options to indicate high-multiplicity events in searches for particles beyond the standard model.

Acknowledgements

The Rice University group acknowledges funding from the US Department of Energy (DOE) Grant #DE-SC0010103. The UCLA group acknowledges funding from DOE and the National Science Foundation.

References

- [1] CMS Collaboration, “The phase-2 upgrade of the CMS muon detectors”, Technical Report CERN-LHCC-2017-012. CMS-TDR-016, CERN, Geneva, Sep, 2017. Available at <https://cds.cern.ch/record/2283189>.
- [2] A. Colaleo, A. Safonov, A. Sharma, and M. Tytgat, “CMS Technical Design Report for the Muon Endcap GEM Upgrade”, Technical Report CERN-LHCC-2015-012. CMS-TDR-013, CERN, Jun, 2015. Available at <https://cds.cern.ch/record/2021453>.
- [3] UCLA High Energy Physics, “TMB 2005 design”,. Available at http://cmsdoc.cern.ch/cms/MUON/cscTrigger/html/ucla/tmb2005/tmb2005_spec_v4p55.pdf.
- [4] S. Dildick et al., “Improving the performance of the CLCT pattern finder in CSCs using lookup-tables.”, *CMS Note* **2019/059** (2020). Available at http://cms.cern.ch/iCMS/jsp/db_notes/noteInfo.jsp?cmsnoteid=CMS%20DN-20-059.
- [5] M. Matveev, “Inputs and outputs of MPC”,. Available at http://padley.rice.edu/cms/MPCMEZ6_050518.pdf.