Available on CMS information server **CMS IN 2005/XXX**

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

October 25, 2005

Description of Detector Raw Data Formats

G. Bruno

Univ. Catholique de Louvain, Louvain-la-Neuve, Belgium

M. Maggi

Univ. di Bari e Sez. dell'INFN, Bari, Italy

R. Alemany Fernandez, N. Almeida

LIP, Lisboa, Portugal

N. Amapane, E. Meschi

CERN, Geneva, Switzerland

D. Kotlinski

Paul Scherrer Inst., Villigen, Switzerland

J. A. Coughlan

Rutherford Appleton Laboratory, Didcot, UK

D. Acosta

University of Florida, USA

J. Mans

University of Maryland, USA

B. Bylsma, L. S. Durkin, J. Gilmore, J. Gu, T. Y. Ling

Ohio State University, USA

[list incomplete]

Abstract

This note describes the format of the raw data sent to the DAQ by the FEDs of the various detectors.

1 Introduction

This note is intended as a “working document” to collect the description of the format of the raw data transmitted via S-LINK64 by the Front End Drivers (FED) of the different subdetectors to the DAQ.

In Section 1, the common format of FED headers and trailers is described. The detector-specific payload is described in the subsequent sections.

2 Common Data Format[1)]

When an event fragment is ready to be transmitted to the DAQ, the FED encapsulates the data according to the common CMS data format of Table 1 and writes the data into the S-LINK64 port. The encapsulation words are control words flagged by the S-LINK64 control bit. Detector payload are data words.

The DAQ will neither look into the sub-detector payload nor touch the encapsulation: the header and the trailer will be transmitted ”as is” to the filter units without any modification.

The fields in the header and trailer words of Table 1 have the following meaning:

DAQ Header

BOE_n: Identifier for the beginning of an event fragment (BOE_1 = ’010’).

Evt_ty: Event Trigger type identifier, defined by the central DAQ, cf Table 2 [6].

LV1_id: The level-1 event number generated by the TTC system. The first event after a TTC reset (B-Go5 Resync, B-Go6 HardReset, or B-Go7 ResetEventCounter) is tagged with Event Number = 1. The Event Counter has 24 bits, the event following Event Number = $2^{24} - 1$ is Event Number = 0.

BX_id: The bunch crossing number, generated by the TTC system. Reset on every LHC orbit.

Source_id: Identifier of the FED; 2 bits are reserved for FED internal usage.

FOV: Version identifier of the FED data format.

H: When set to ’0’, the current header word is the last one. When set to ’1’, another header word can follow[2)].

$: Bit used by the S-LINK64 hardware

DAQ Trailer

EOE_n: Identifier for the end of an event fragment (EOE_1 = ’1010’).

Evt_lgth: The length of the event fragment counted in 64-bit words including header and trailer.

CRC: Cyclic Redundancy Code of the event fragment including header and trailer.

Evt_stat: Event fragment status information (defined by the central DAQ).

TTS: Current value of the Trigger Throttling System bits.

T: When set to ’0’, the current trailer word is the last one. When set to ’1’, another trailer words can follow[3)].

x: Indicates a reserved bit.

$: Bit used by the S-LINK64 hardware.

[1)] Section taken from http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/RUWG/DAQ_IF_guide/DAQ_IF_guide.html#CDF

[2)] in the ECAL DCC, H is set to 1.

[3)] in the ECAL DCC, T is set to 0.

Table 1: Common FED data format.

<table>
<tr><td></td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>K</td><td colspan="4">BOE_1</td><td colspan="4">Evt_ty</td><td colspan="24">LV1_id</td><td colspan="12">BX_id</td><td colspan="12">Source_id</td><td colspan="4">FOV</td><td>H</td><td>x</td><td>$</td><td>$</td></tr>
<tr><td>D</td><td colspan="64">Sub-detector payload</td></tr>
<tr><td>D</td><td colspan="64">Sub-detector payload</td></tr>
<tr><td colspan="65">⋮</td></tr>
<tr><td>K</td><td colspan="4">EOE_1</td><td colspan="4">xxxx</td><td colspan="24">Evt_lgth</td><td colspan="16">CRC</td><td colspan="4">xxxx</td><td colspan="4">EVT_stat</td><td colspan="4">TTS</td><td>T</td><td>x</td><td>$</td><td>$</td></tr>
<tr><td></td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Table 2: The TRIGGER TYPE description.

TRIGGER TYPE	**Description**
0001	Physics trigger
0010	Calibration trigger
0011	Test trigger
0100	Technical trigger (external trigger)
0101	Simulated events (reserved for DAQ usage)
0110	Traced events (reserved for DAQ usage)
1111	Error

3 Pixel Data Format

The pixel Front End Driver (FED) readout board will read data from 36 input links, build events, and send the event packets through the S-link to the DAQ. During the event building it has to reconstruct pixel addresses from the 6-level analog signals. This procedure requires a set of pre-programmed threshold levels.

Each FED will send to the DAQ a data packet starting with the standard packet header and ending with the standard packet trailer. The header includes a bit field identifying uniquely the FED. Between the header and the trailer there will be a variable number of 32-bit data words, with one pixel stored per word. The format of the 32-bit word is:

- 6-bit Link id, defines the input link to the FED (0-35);
- 5-bit ROC id, defines the ROC within one link (0-23);
- 5-bit Double-Column id, define the double column within on ROC (0-25);
- 8-bit Pixel id, define the pixel address within on Double-Column (0-179);
- 8-bit ADC value, the signal amplitude, extracted from a 10-bit ADC.

Table 3 summarizes the pixel format.

Note that in our data format the source id (FED number) is not included. We depend on the source id included in the DAQ S-link header. This header should be available in the code which will transform the raw data format to the format used in the reconstruction code.

Table 3: Pixel readout format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
pixel	LINK-ID						ROC-ID					DCOL-ID					PIX-ID								ADC								
	5	4	3	2	1	0	4	3	2	1	0	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Based on this format the FED data volume in bytes is calculated as 4 * (3 + number-of-pixels). The pixel barrel readout has been arranged in such a way to approximately fit the requirement of 2.0 kB per FED at high luminosity. For the low luminosity pp collisions, the pixel data volume for the barrel FEDs will be around 0.6 kB per event. The difference between various FEDs should be around 10%. For the pixel endcaps the data volume per FED will be smaller, about 1.8(0.55) kB per event at high(low) luminosity.

When reading data over VME an alternative ”raw” data format is foreseen in addition to the standard one. This format will be only used during the level calibration procedure performed in a XDAQ client application and will never be used in ORCA.

4 Silicon Strip Data Format

The silicon strip tracker Front End Driver readout board will be able to operate in various modes. The modes enable running for different types of physics events, running for calibration or commissioning. The modes are [ref????]:

- Virgin Raw Data mode - in this mode the FED performs no data processing at all. This mode is intended for commissioning and monitoring running.
- Processed Raw Data mode - in this mode the FED performs pedestal subtraction and re-ordering (where the strip data are re-ordered out of APV-MUX order into strip order). This mode is intended for use when colliding heavy ions.
- Zero Suppressed mode - in this mode the FED performs pedestal subtraction, re-ordering, common mode noise calculation and subtraction, and zero suppression. This is intended to be the ’normal’ running mode for proton-proton collisions.
- Scope mode - in this mode the FED does not perform any APV header finding, and simply captures the data entering a channel up to a pre-defined number of bytes. This enables the monitoring of APV tick marks to allow for tasks such as syncronization of FED channels.

In each mode the FED outputs the data with a DAQ header, a tracker specific header and a DAQ trailer. The DAQ header was defined in the DAQ TDR [ref????] and consists of event type, level 1 trigger number, bunch crossing number, source ID, etc. The tracker specific header contains information needed to check tracker and FED operation. There are two different tracker specific headers [ref???]:

- Full Debug mode - this header contains APV error flags, FED status registers, data lengths, etc. This header format willbe used for commissioning, syncronization and raw data mode running.
- APV Error mode - this will be the 'standard' running mode during a physics run (when the FED is operating in zero suppressed mode). It is designed to be as small as possible, and so only contains as much information as is needed for event reconstruction and monitoring.

4.0.1 Silicon Strip FED Event Size

The event buffer produced by the FED in raw data and processed raw data mode has a fixed size of 49,624 bytes (for full debug mode header) or 49,520 bytes (for APV error mode header). In scope mode the event size depends on the number of bytes captured per FED channel. The maximum allowed is 512 bytes per channel, which results in an event size maximum the same as raw data mode (depending on header format). In zero suppression mode the event size depends on the occupancy in the tracker, and the zero suppression thresholds.

5 ECAL Data Format

The ECAL data are composed by 54 event fragments processed in the Data Concentrator Card (DCC) boards [1, 2] (36 for the ECAL Barrel (EB) and 18 for the ECAL End-Cap (EE)). The ECAL raw data format is shown in Table 4. Each Event is composed of five block types:

- DCC Header Block
- TCC Block
- SR Block
- Tower Block
- Crystal Block

These blocks are organized in 64 bit words and are enclosed by the standard CMS DAQ header and trailer. Each block is characterized by a unique bit field identifier, which helps to track possible data formatting errors.

The data structure starts with the standard CMS DAQ header followed by the specific ECAL DCC Header Block. Trigger data received from the Trigger Concentrator Card (TCC) [2] are grouped in the TCC Blocks. The number of TCC Blocks depends on the detector geometry, being one in the EB and four in the EE. Front-End (FE) data must be reduced by a factor of near 20. ECAL data filtering is based on a Selective Readout algorithm finding ECAL areas of interest, which should be readout with low zero suppression or without zero suppression at all. The other areas can be suppressed or readout with a higher zero suppression level. For each readout unit, comprising a Trigger Tower in the EB geometry or a Super Crystal in the EE geometry, there is a flag identifying the level of data suppression that must be applied to the input data. These flags are processed in the Selective Readout Processor (SRP) [2, 3] and grouped in the SR Block. The number of Tower (or Super Crystal) Blocks in each event depends on the enabled FE channels and on the SR flags, which can suppress the entire FE channel data. The number of Crystal Blocks belonging to a given Tower Block (or Super Crystal) is dynamic and depends on the deposited energy and on the associated SR flag.

5.1 DCC Header Block

The DCC Header Block is composed by eight words characterized by a block identifier (B'00') and a header word number (from 1 to 8). The block carries the following fields:

EVENT LENGTH: Number of 64 bit words in the event, also present in the CMS DAQ trailer. This number includes the CMS DAQ header and trailer words. The fact that the event length is placed at the beginning of the event makes it more suitable for a fast decoding and event characterization.

Table 4: The ECAL data format.

DCC Block

63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

BOE(B'0101')	TRIGGER TYPE	LV1	BX	FED/DCC ID	FOV										
(B'00')	H1(B'0001')	RUN NUMBER	DCC ERRORS	EVENT LENGTH											
(B'00')	H2(B'0010')		DETAILED TRIGGER TYPE	RUN TYPE											
(B'00')	H3(B'0011')	TCC_ CHSTATUS #4	TCC_CHSTATUS#3	TCC_CHSTATUS#2	TCC_CHSTATUS#1	SR_CHSTATUS		TZS	ZS	SR	ORBIT COUNTER				
(B'00')	H4(B'0100')	FE_CHSTATUS#14	FE_CHSTATUS#13	FE_CHSTATUS#12	FE_CHSTATUS#11	FE_CHSTATUS#10	FE_CHSTATUS#9	FE_CHSTATUS#8	FE_CHSTATUS#7	FE_CHSTATUS#6	FE_CHSTATUS#5	FE_CHSTATUS#4	FE_CHSTATUS#3	FE_CHSTATUS#2	FE_CHSTATUS#1
... (+ 3W)															
(B'00')	H8(B'1000')	FE_CHSTATUS#70	FE_CHSTATUS#69	FE_CHSTATUS#68	FE_CHSTATUS#67	FE_CHSTATUS#66	FE_CHSTATUS#65	FE_CHSTATUS#64	FE_CHSTATUS#63	FE_CHSTATUS#62	FE_CHSTATUS#61	FE_CHSTATUS#60	FE_CHSTATUS#59	FE_CHSTATUS#58	FE_CHSTATUS#57

$

TCC Block	one in the BE, 4 in the EE
SR Block	
Tower Block	(non suppressed blocks)

EOE(1010)		EVENT LENGTH	CRC (16)	EV ERRORS	EVENT STATUS	

TCC Block

63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

(B'011')	LE1	LE0	#TIME SAMPLES	#TT		E1	LV1(LOCAL)	(B'011')	E0	BX(LOCAL)		TCC ID
(B'011')		TTF#4	TPG #4		TTF#3	TPG #3	(B'011')		TTF#2	TPG #2		TTF#1	TPG #1
...(+15W)												
(B'011')		TTF#68	TPG #68		TTF#67	TPG #67	(B'011')		TTF#66	TPG #66		TTF#65	TPG #65

SR Block

63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

(B'100')	LE1	LE0		#SR FLAGS		E1	LV1(LOCAL)	(B'100')	E0	BX(LOCAL)		SRP ID								
(B'100')		SR#16	SR#15	SR#14	SR#13		SR#12	SR#11	SR#10	SR#9	(B'100')		SR#8	SR#7	SR#6	SR#5		SR#4	SR#3	SR#2	SR#1
... (+2W)																				
(B'100')		SR#64	SR#63	SR#62	SR#61		SR#60	SR#59	SR#58	SR#57	(B'100')		SR#56	SR#55	SR#54	SR#53		SR#52	SR#51	SR#50	SR#49
(B'100')		(B'100')		SR#68	SR#67	SR#66	SR#65													

Tower Block

63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

(B '11')		BLOCK LENGTH		E1	LV1(LOCAL)	(B '11')		E0	BX(LOCAL)		#TIME SAMPLES	TT/SC ID				
(B '11')	G	ADC#3		G	ADC#2	(B'11')	G	ADC#1		TZS		GMF	SMF	M	XTAL ID	STRIP ID
(B '11')	G	ADC#7		G	ADC#6	(B '11')	G	ADC#5		G	ADC#4					
(B '11')	G	ADC# ...		G	ADC#10	(B '11')	G	ADC#9		G	ADC#8					
(B '11')	G	ADC#3		G	ADC#2	(B'11')	G	ADC#1		TZS		GMF	SMF	M	XTAL ID	STRIP ID
(B '11')	G	ADC#7		G	ADC#6	(B '11')	G	ADC#5		G	ADC#4					
(B '11')	G	ADC# ...		G	ADC#10	(B '11')	G	ADC#9		G	ADC#8					

DCC ERRORS: The DCC event error field described in Table 5.

RUN NUMBER: Sequential run number or encoded run conditions as written in the DCC Run ID register.

Hi: Header word count identifier.

RUN TYPE: Field (Table 6) encoded by the DCC supervisor software in local data taking mode and written into the DCC Run Type register.

- **RT_DCCID**: The DCC ID;
- **RT_HALF**: Part of the Super module to be readout (Table 7).
- **RT_TYPE**: The run type (Table 8).
- **SEQUENCE**: The operation sequence during a local run. The SEQUENCE encoding depends on the RT_TYPE and its description is shown in Table 9.
- **MGPA GAIN**: The MPGA operation gain mode during the local run (Table 10).
- **MEM GAIN**: The MEM channels operation gain mode during the local run (Table 11).
- **CYCLE SETTINGS**: The cycle operation settings applied during the local run. The CYCLE SETTINGS encoding depends on the SEQUENCE and is shown in Table 6. The POWER, FILTER, DELAY, VINJ, MGPA CONTENT and OFFSET fields are binary representations of configuration parameters for each SEQUENCE type. The RT_WL field encodes the Laser/LED wavelength according to Table 12.

DETAILED TRIGGER TYPE: A Timing and Trigger Control (TTC) [4] long command that is written in the DCC header during calibration triggers in the orbit gaps. The content of this word is the following:

- **DTT_DCCID**: The DCC ID
- **DTT_HALF**: The same as RT_HALF (Table 7).
- **DTT_TYPE**: The same as RT_TYPE (Table 8).
- **DTT_WL**: The same as RT_WL (Table 12).

ORBIT COUNTER: The orbit number in the run.

SR and ZS: These bit fields characterize the DCC readout conditions. The SR bit specifies if selective readout is enabled and in the case of being disabled the ZS field specifies if crystal data must be zero suppressed or fully readout.

TZS: Test Zero Suppression mode. In this operation mode (TZS=1) the zero suppression (if requested in the DCC configuration or by a SR flag) is applied to the crystal data without data removal. The result of the filter is set in the TZS field of the Crystal Block allowing testing off-line the ZS algorithm.

SR_CHSTATUS: SRP input channel status (Table 14).

TCC_CHSTATUS#i: TCC input channel #i status (Table 14).

FE_CHSTATUS#i: FE input channel #i status (Table 14).

Table 5: DCC error description.

DCC Error (binary)	**Description**
00000000	No Errors
00000001	Empty event produced : DAQ header + the first two DCC header words + DAQ trailer
xxxxxx1x	Data timeout : at least one channel has a data presence timeout
0xxxx1xx	Channel error: at least one channel on error
?	To be defined
1xxxxxxx	No error check performed

Table 6: The Run Type word format. The CYCLE SETTINGS encoding depends on the selected SEQUENCE.

RUN TYPE																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE SETINGS															GAIN			SEQUENCE			RT_TYPE			RT_HALF		RT_DCCID				

Table 7: RT_HALF description.

RT_HALF (binary)	**Description**
01	First super module half
10	Second super module half
11	Full super module

Table 8: RT_TYPE description.

RT_TYPE (binary)	Description
001	Laser
010	Test Pulse
011	Pedestal
100	LED

Table 9: SEQUENCE description.

SEQUENCE (binary)	Description
If RT_TYPE is Laser Trigger	
000	Standard Laser
001	Laser Power Scan
010	Delay Scan
If RT_TYPE is Test Pulse Trigger	
000	Scan MEM
001	MGPA
If RT_TYPE is Pedestal	
000	Standard Pedestal
001	Pedestal Offset Scan
If RT_TYPE is LED	
000	Standard LED

Table 10: MGPA gain description.

MGPA (binary)	Description
00	Free gain (default)
01	Forced gain 12
10	Forced gain 6
11	Forced gain 1

Table 11: MEM gain description.

MEM (binary)	Description
0	MEM forced gain 1
1	MEM forced gain 16 (default)

Table 12: The RT_WL description.

RT_WL (binary)	Description
001	Red (Laser)
010	Blue (Laser)
011	Infra-Red (Laser)
100	Green (Laser)
101	λ1 (LED)
110	λ2 (LED)

Table 13: Detailed trigger type word format.

DETAILED TRIGGER TYPE															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DTT_WL			DTT_TYPE			DTT_HALF		DTT_DCCID					

Table 14: DCC Channel status description. If multiple errors are identified the channel status is set according to its priority (a lower channel status value implies a higher priority).

CHSTATUS	Description
0000	Channel enabled and no errors identified
0001	Channel disabled by configuration
0010	Channel data ignored in response to a data presence timeout
0011	Header identifier mismatch
0100	Block word count mismatch
10xx	Sync error (bit 0: LV1 , bit 1: BX)
11xx	Parity error (bit 0: horizontal parity , bit 1 : vertical parity)

5.2 TCC Block

The TCC Blocks, added after the DCC Block Header, include the trigger data received from the TCC links. The bock is identified by the B('011') field at the beginning of each 32 bit word and is composed by the following fields:

TCC ID: The trigger data source board id (from 1 to 108).

BX: TCC local bunch crossing (BX) counter.

E0: DCC BX error bit (active if there is a mismatch between the TCC BX counter and the DCC BX counter).

LV1: TCC local level one accept (LV1) counter.

E1: DCC LV1 error bit (active if there is a mismatch between the TCC LV1 counter and the DCC LV1 counter).

#TT: Number of Trigger Towers (TTs) associated to this TCC. In the EB each TCC is associated to 68 TTs. In the inner EE each TCC Block includes a maximum of 32 TTs words while in the outer EE this number is decreased to 16. An EE DCC serves two 20 degrees sectors with two inner and two outer EE sections.

#TIME SAMPLES: Number of time samples for the Trigger Primitives Generator (TPG) data words. In normal data acquisition mode there is only one time sample. For debugging operation mode, 4 or 8 time samples are foreseen. For multiple time samples, TT flags and TPG data are ordered consecutively per set of samples in each TT.

LE0: Local BX error bit. This bit is received from the TCC data and identifies if there is a mismatch (LE0=1) between the TCC local BX counter and the info received from the TTC system.

LE1: Local LV1 error bit. This bit is received from the TCC data and identifies if there is a mismatch (LE1=1) between the TCC local LV1 counter and the info received from the TTC system.

TPG#i: 9 bits coding the trigger primitive of TT i.

TTF#i: TT flag (Table 15) sent from the TCC to the SRP and included in the DCC for debugging proposes. The first two bit flags characterize the TT energetic state, while the third bit is enabled if the TT flag was produced with erroneous conditions.

Table 15: Trigger tower flags description.

TT Flag	Description
000	Low interest TT (Tower Et is bellow low threshold)
001	Mid-interest TT (Tower Et is between low and high thresholds)
010	Not used
011	High-interest TT (Tower Et is above high threshold)
100	Force TT readout (Agilent sync link error)
101	Force TT readout (Hamming code error)
11x	Force TT readout (Other error conditions)

5.3 SRP Block

The SRP Block, added after the TCC Blocks, includes the selective readout flags received from the SRP link. The block is identified by the B(100) field at the beginning of each 32 bit word and is composed by the following fields:

SRP ID: The SRP identification.

BX: SRP local BX counter.

E0: DCC BX error bit (active if there is a mismatch between the SRP BX counter and the DCC BX counter).

LV1: SRP local LV1 counter.

E1: DCC LV1 error bit (active if there is a mismatch between the SRP LV1 counter and the DCC LV1 counter).

#SRFLAGS: Number of SR flags associated to this link. In the EB the DCC receives a total of 68 SR flags, while in the EE the DCC receives 34, 35 or 36 flags.

LE0: Local BX error bit. This bit is received from the SRP data and identifies if there is a mismatch (LE0=1) between the SRP local BX counter and the info received from the TTC System.

LE1: Local LV1 error bit. This bit is received from the SRP data and identifies if there is a mismatch (LE1=0) between the SRP local LV1 counter and the info received from the TTC System.

SR#i: 3 bits coding the SR flags for each FE channel (Table 16). The first two bit flags characterize the SRP algorithm flag, while the third bit is used to indicate if the flag was produced in erroneous conditions.

Table 16: Selective Readout flags description.

SR Flag	Description
000	Suppress FE channel
001	Read FE channel with lv1 zero suppression
010	Read FE channel with lv2 zero suppression
011	Read FE channel without zero suppression
1xx	xx action has been forced by an erroneous condition or by configuration: - TT flagged in forced readout mode (from TCC) - Data transmission errors (TCC and Algorithm Board(AB) link errors)

5.4 Tower Header and Crystal Blocks

The Tower Block collects filtered FE data received from the FE boards, and is organized in Crystal Blocks. The Tower (or Super Crystal in the EE) Block contains a maximum of 25 crystals, corresponding to a fully readout tower (or super crystal). The Tower Blocks and the Crystal Blocks are identified by the B(11) field at the beginning of each 32 bit word and are composed by the following fields:

Tower Header Block:

TT/SC ID: The FE readout board id (from 1 to 68).

#TIME SAMPLES: Number of time samples for the ADC crystal data (default is 10). The number of samples accepted by the DCC is given by: 2 +4n (where n goes from 0 to 15).

BX: FE local BX counter.

E0: DCC BX error bit (active if there is a mismatch between the FE BX counter and the DCC BX counter).

LV1: FE local LV1 counter.

E1: DCC LV1 error bit (active if there is a mismatch between the FE LV1 counter and the DCC LV1 counter).

BLOCK LENGTH: A block word counter, in 64 bit words, tacking into account the Tower Header Block and the Crystal Block words.

Crystal Block:

STRIP ID: The crystal strip identification (from one to five).

CRYSTAL ID: The crystal identification inside the strip (from one to five).

M: Monitoring bit, enabled for monitoring triggers.

GMF and SMF: bit fields received from the FE strip header (see Annex A.1), the GM flag and the SM flag qualify respectively a Global or a Strip level crystal time sample misalignment.

TZS: When the DCC is configured to operate in the test zero suppression mode, the TZS bit will be enabled whenever the crystal data frame is flagged to be suppressed.

ADC#i: 12 bits representing the crystal time sample i.

G: MGPA gain (B'00': not used, B'01': gain 12, B'10': gain 6, B'11': gain 1).

The crystal numbering in the DCC raw data format follows the FE numbering scheme. The FE identifies each crystal by the strip to which it belongs to (from one to five) and the crystal number inside the strip (from one to five). To get the absolute crystal number inside the TT or inside the Super Module, a lookup table should be applied. The lookup table should take into account the particular arrangement of VFE cards inside the FE card, and the particular arrangement of FE cards inside the Super Module.

5.5 Erroneous input data and implications on ECAL Raw data

The identification of erroneous input data fragments has implications on the way the event is built. The actions taken from the DCC in presence of erroneous input data are described in this section.

Synchronization error with erroneous BX or LV1 from an input channel:

- Global action: The channel data are included in the assembled event and the associated channel status field set with the sync error flag.
- Error identified in the SRP Block: No filtering is applied to crystal data.
- Error identified in the FE Block: No filtering is applied to crystal data from this particular channel.

Data timeouts, wrong header identifiers or block word counts:

- These are considered severe errors. Data from the faulty channel are ignored and the channel status is flagged with the identified error condition.

Misaligned FE Data (GMF or SMF bits):

- The crystal data with misaligned time samples are not zero suppressed.

The DCC is also able to operate with no error check (DCCERR set to '1xxxxxxxx').

5.6 Raw Data Parsing and Monitoring

Detector raw data need to be decoded and the data quality monitored. In the Off-line system data will serve physics analysis, while in the On-line system local data taking is used mainly to monitor the detector performance. A software package was developed with the intent to allow on-line and off-line applications to access all defined raw data bit fields in an object oriented way. In addition, the package verifies the event data structure identifying possible errors. The class diagram for this package, developed in C++, is represented in Figure 4.

Figure 4. The data parser class diagram.

Raw data blocks are characterized by a set of bit fields. Each bit field is defined through a word position inside the block, a bit position inside the word and a defined bit size. The DCCDataField class is a representation of a raw data bit field. These data structures are built and stored in the DCCDataMapper containers. The DCCDataBlockPrototype class encapsulates the knowledge on how to decode a raw data buffer based on the defined bit fields. When parsing the input data buffers the decoded bit fields are stored in a map container, which can be accessed by the applications using the parser. During the parsing process the DCC data blocks classes perform a set of checks on the decoded data. These checks are described in Table 17. The information about the blocks that need to be created is stored in the data itself. The DCCEventBlock decodes the DCC header bit fields identifying the enabled channels. In a first stage the blocks associated to the enabled TCC and SR channels are created. The DCCTowerBlock instantiation is based on the enabled FE channels and on the SR flags that indicate the rejected channels. The size of the DCCTowerBlock and the knowledge of the number of time samples per crystal allow the determination of the number of readout crystals in each tower or super crystal. For each crystal is instantiated a DCCXtalBlock object. The parsing ends up with the decoding of the DCC trailer. The software includes methods that allow the event raw data display in function of the defined bit fields. The package describing the 2004 H4 Test Beam ECAL raw data is implemented and is included in ORCA_8_8_0 [7] (Calorimetry/EcalBarrelTBDaqDigiFormat). The package is responsible by decoding the raw data and by generating the Calo Data Samples Frames.

Table 17: Error checks performed during parsing.

Data Block	Error check description
DCCEventBlock	Begin Of Event, header identifiers and block identifier
DCCTCCBlock	Channel id, synchronization and block identifiers
DCCSRPBlock	Channel id, synchronization and block identifiers
DCCTowerBlock	Channel id, synchronization, block identifiers and block size (based on the readout conditions and number of crystal samples)
DCCCrystalBlock	Crystal and strip ids and block identifiers
DCCTraillerBlock	End Of Event, event size and CRC

6 HCAL Data Format

6.1 HCAL FED Fragment Format

Table 18: HCAL FED (DCC) data header version 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCC Header 0	0	0	0	HTR Status (14:0)															Unused (0)						Format Version								
DCC Header 1	Error Summary (31:0)																															
Spigot 0	HTR Errors								LRB Error Word								E	P	V	x	x	x	32-bit-word count								
Spigot 1	HTR Errors								LRB Error Word								E	P	V	x	x	x	32-bit-word count								
Spigot 2	HTR Errors								LRB Error Word								E	P	V	x	x	x	32-bit-word count								
...	HTR Errors								LRB Error Word								E	P	V	x	x	x	32-bit-word count								
Spigot 15	HTR Errors								LRB Error Word								E	P	V	x	x	x	32-bit-word count								
Spacer 0	zeroes																															
Spacer 1	zeroes																															
Spacer 2	zeroes																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The header is followed by the sixteen-bit words of the HTR subfragments, which are described in Sec 6.2. There is no additional trailer on the block other than the standard single 64-bit common data format trailer word.

Legend:

- E : Spigot is enabled
- P : Data from spigot is present
- V : Spigot event number matches fragment event number (Valid)
- HTR Status : OR of the E, P, and V bits for each spigot
- Format Version : Version of the DCC header format (currently 2)
- Error Summary : Non-zero indicators for each of the 32 error counters in the DCC
- HTR Errors : Copy of the low-order byte of HTR Header Word 3
- LRB Error Word (7) : Odd number of 16-bit words
- LRB Error Word (6) : Missing header/trailer
- LRB Error Word (5) : Data overflow (truncated)
- LRB Error Word (4) : FIFO empty while reading event (underflow)
- LRB Error Word (3) : EvN mismatch in header/trailer
- LRB Error Word (2) : Block size overflow
- LRB Error Word (1) : Uncorrected link error
- LRB Error Word (0) : Corrected link error

6.2 HTR Subfragment Format

Legend:

- SR: Status request to LRB on DCC (internal electronics use)
- CT: Calibration trigger (parallel run event, should not appear in main DAQ, separate event numbering)
- HM: Histogram mode (should be zero in normal data taking)
- TM: Test mode (data from pattern RAMs)

Table 19: Format of an HCAL HTR Subfragment for normal data taking (version 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header 1	SR	0			LRB id				HTR EvN(7:0)							
Header 2	HTR EvN(23:8)															
Header 3	1	CT	HM	TM	Reserved (0)				CK	OD	LW	LE	RL	EE	BZ	OW
Header 4	Orbit N(5:0)						HTR Submodule Id (9:0)									
Header 5	Format Ver				HTR BCN(11:0)											
Header 6	Trig Prim Word Count [m]								Presamples					Reserved		
Header 7	FW Type			Firmware Revision												
Header 8	Reserved (0)								Pipeline length							
Trigger Primitive 1	FiberAd			ChAd		0	PD	TP (8:0)								
...	FiberAd			ChAd		0	PD	TP (8:0)								
Trigger Primitive m	FiberAd			ChAd		0	PD	TP (8:0)								
Precision Data 1	FiberAd			ChAd		ER	DV	CapId	QIE ADC (6:0)							
...	FiberAd			ChAd		ER	DV	CapId	QIE ADC (6:0)							
Precision Data n	FiberAd			ChAd		ER	DV	CapId	QIE ADC (6:0)							
Parity word	0xFFFF, if zero suppression produces an odd number of payload words															
Trailer 4	Samples per channel					Total precision word count [n] (10:0)										
Trailer 3	Reserved (0)				16-bit-word count (11:0)											
Trailer 2	32-bit-word count (from DCC)															
Trailer 1	HTR EvN(7:0)								LRB Error Word							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- CK: Clocking problem (DLL, TTC, etc)
- OD: Optical data error (link, format, capid check)
- LW: Latency warning on an input fiber
- LE: Latency error on an input fiber
- RL: Rejected L1A (previous L1A violated trigger rules)
- EE: Empty Event (no data because of previous busy)
- BZ: HTR is busy (buffers full)
- OW: Overflow warning – HTR is approaching busy.
- Format Ver: Format version number. Currently 0.
- Presamples: Number of samples before the triggered bunch crossing
- FW Type: The three MSB of the HTR Firmware version (18:16) specify the purpose of the firmware.

Each histogram is 32 bins of 16-bit unsigned integers. A separate histogram is made for each capacitor id. The histograms are stored in the order shown in Table 21. Data for only two of the eight input fibers of a half-HTR can be contained in a single event.

Table 20: Format of an HCAL HTR Subfragment for histogram mode(version 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header 1	SR	0			LRB id				HTR EvN(7:0)							
Header 2	HTR EvN(23:8)															
Header 3	1	CT	HM=1	TM	Reserved (0)				CK	OD	LW	LE	RL	EE	BZ	OW
Header 4	Orbit N(5:0)						HTR Submodule Id (9:0)									
Header 5	Format Ver				HTR BCN(11:0)											
Header 6	0	Fiber Id 2			0	Fiber Id 1			Reserved(0)							
Header 7	FW Type			Firmware Revision												
Header 8	Reserved (0)								FiberErrors							
Histo 1,Bin 1	Sixteen-bit histogram bin															
...	Sixteen-bit histogram bin															
Histo 1,Bin 32	Sixteen-bit histogram bin															
Histo 2,Bin 1	Sixteen-bit histogram bin															
...	Sixteen-bit histogram bin															
Histo 24,Bin 32	Sixteen-bit histogram bin															
Trailer 4	Reserved (0)															
Trailer 3	Reserved (0)				16-bit-word count (11:0)											
Trailer 2	32-bit-word count (from DCC)															
Trailer 1	HTR EvN(7:0)								LRB Error Word							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 21: Ordering of the histograms within an HTR histogram subfragment.

	Fiber	ChAd	CapId
1	Fiber 1	0	0
2	Fiber 1	0	1
3	Fiber 1	0	2
4	Fiber 1	0	3
5	Fiber 1	1	0
		...	
12	Fiber 1	2	3
13	Fiber 2	0	0
14	Fiber 2	0	1
		...	
24	Fiber 2	2	3

7 DT Data Format

The DT data format.

8 CSC Data Format

The Endcap Muon (EMU) system consists of 468 Cathode Strip Chambers (CSCs), each providing multiple data sources in the data stream:

- Data Acquisition Motherboard (1 DMB per CSC)
- Cathode Front End Board (4-5 CFEBs per CSC)
- Anode Front End Board (12-42 AFEBs per CSC)
- Trigger Motherboard (1 TMB per CSC)
- Anode LCT Board (1 ALCT per CSC)

The DMB processes the data from each source in sequence on an event-by-event basis to form a data packet that is transmitted via fiber to a DDU in an EMU FED crate. The 36 DDUs in the EMU system are read out through 4 DCCs with a total of 8 S-Link64 mezzanine boards. There is a 37th DDU for the CSC Trackfinder that has its own dedicated S-Link64 readout to the FRL.

Each component mentioned above has a unique data format that is detailed in the following sections.

8.1 EMU DCC Format

All CSC data is read out through 4 Data Concentrator Cards via 8 S-Link64 mezzanine boards. Each S-Link carries data from 4 to 5 EMU DDUs wrapped by DCC Headers and Trailers as shown below:

- DCC Header block (2 64-bit words)
- DDU-0 Data (variable number of 64-bit words)
- DDU-1 Data
- DDU-2 Data
- DDU-3 Data
- DDU-4 Data (present in 4 out of 8 EMU S-Links)
- DCC Trailer block (2 64-bit words)

The DCC Header and Trailer blocks conform to the S-Link protocol in the CMS Common Data Format. EMU-specific event information is carried in the Second Header and First Trailer, as shown in Table 22, which carry the following fields:

Orbit_Count: Number of LHC orbits since last L1 Reset.

DCC_FIFO_Status: 5-bit FIFOs_Used register, 5-bit FIFOs_Almost_Empty register, 5-bit FIFOs_Empty register, 1 dummy bit.

Proc_Time: Time used to process the current event, encoded in 8-bits: '$abcdefgh$' converts to $(bcdefgh) * (16 * *a) * 0.41$ us.

DDU_data_status: DCC Input FIFO status register, 8-bits per DDU.

Timeout_Flags: Timeout register, 1-bit per DDU plus 3 dummy bits.

Table 22: EMU DCC implementation of Common FED data format.

<table>
<tr><td></td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>K</td><td colspan="4">BOE_1</td><td colspan="4">Evt_ty</td><td colspan="24">LV1_id</td><td colspan="12">BX_id</td><td colspan="12">Source_id</td><td colspan="4">FOV</td><td>H</td><td>x</td><td>$</td><td>$</td></tr>
<tr><td>K</td><td colspan="8">D9h</td><td colspan="32">Orbit_Count</td><td colspan="16">DCC_FIFO_Status</td><td colspan="8">17h</td></tr>
<tr><td>D</td><td colspan="64">DDU_Data_Payload</td></tr>
<tr><td></td><td colspan="64">⋮</td></tr>
<tr><td>K</td><td colspan="8">EFh</td><td colspan="8">Proc_Time</td><td colspan="40">DDU_data_status (8-bits each)</td><td colspan="8">Timeout_Flags</td></tr>
<tr><td>K</td><td colspan="4">EOE_1</td><td colspan="4">xxxx</td><td colspan="24">Evt_lgth</td><td colspan="16">CRC</td><td colspan="4">xxxx</td><td colspan="4">EVT_stat</td><td colspan="4">TTS</td><td>T</td><td>x</td><td>$</td><td>$</td></tr>
<tr><td></td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Table 23: EMU DDU data format.

<table>
<tr><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">BOE_1</td><td colspan="4">Evt_ty</td><td colspan="24">LV1_id</td><td colspan="12">BX_id</td><td colspan="12">Source_id</td><td colspan="4">FOV</td><td>H</td><td>x</td><td>$</td><td>$</td></tr>
<tr><td colspan="16">8000h</td><td colspan="16">0001h</td><td colspan="16">8000h</td><td>0</td><td colspan="15">CSCs_Full (1-bit per CSC)</td></tr>
<tr><td>0</td><td colspan="15">CSCs_alive (1-bit per CSC)</td><td colspan="16">DDU_output_status (see Table 24)</td><td>0</td><td colspan="15">CSC_data_available (1-bit per CSC)</td><td>0</td><td colspan="7">BoE_status*</td><td colspan="4">BoE_TTS</td><td colspan="4">CSC_count</td></tr>
<tr><td colspan="64">CSC_Data_Payload</td></tr>
<tr><td colspan="64">⋮</td></tr>
<tr><td colspan="16">8000h</td><td colspan="16">FFFFh</td><td colspan="16">8000h</td><td colspan="16">8000h</td></tr>
<tr><td colspan="32">DDU_Event_status (see Table 26)</td><td>Q</td><td colspan="15">CSC_error (1-bit per CSC)</td><td>0</td><td colspan="15">CSC_warn (1-bit per CSC)</td></tr>
<tr><td colspan="4">EOE_1</td><td colspan="4">xxxx</td><td colspan="24">Evt_lgth</td><td colspan="16">CRC</td><td colspan="4">xxxx</td><td colspan="4">EVT_stat</td><td colspan="4">TTS</td><td>T</td><td>x</td><td>$</td><td>$</td></tr>
<tr><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

* See Table 25.

Q: CSC Full flag.

Table 24: EMU DDU Output Status Bit Definitions.

Bit	Output Status Description	Condition
15	DDU Output-Limited Buffer Overflow occurred: DCC/S-Link or SPY path was rate limited resulting in a buffer overflow	may be OK
14	DAQ Wait was asserted by S-Link or DCC	OK
13	Link Full (LFF) was asserted by DDU S-Link	OK
12	DDU S-Link Never Ready	OK
11	SPY FIFO Overflow occurred	OK
10	SPY Event was skipped to prevent overflow	OK
9	SPY FIFO Always Empty	OK
8	SPY Fiber Connection Error occurred	OK
7	DDU Buffer Overflow caused by DAQ Wait: DCC/S-Link was rate-limited resulting in a buffer overflow	RESET
6	DAQ Wait is set by DCC/S-Link	OK
5	Link Full (LFF) is set by DDU S-Link	OK
4	Not Ready is set by DDU S-Link	OK
3	SPY FIFO is Full	OK
2	SPY Path was Not Enabled for this event: no SPY data was sent due to prescale or throughput limitation	OK
1	SPY FIFO is Not Empty	OK
0	DCC Link is Not Ready	may be OK

8.2 EMU DDU Format

The EMU DDU takes in data from up to 15 EMU DMBs on 1.6 Gb/s fiber optic cables (1 fiber per CSC) via high-density (LC) front panel inputs. The data from each fiber gets deserialized by the Rocket IO within a Xilinx FPGA to recover the original 16-bit DMB data structure, which is then buffered in a FIFO array. Each DMB sends a multiple of 4 16-bit words for every event, thus the DDU performs all processing on a 64-bit word basis.

Upon receipt of an L1A trigger, the DDU reads the complete event record from each of the CSCs in order (0 to 14) and sends the data to the EMU DCC via pairs of multi-gigabit lines embedded in the custom FED backplane. There is a parallel SPY path for data monitoring via gigabit Ethernet that can be prescaled as needed; this path has the same DDU data format as the primary readout path to the DCC. The DDU format conforms to the CMS Common Data format, except that the 64-bit structure does not allow the K-bit to be set at the DDU level. However, there is a unique bit sequence embedded in the second DDU Header and the first DDU Trailer to serve as markers in the data stream, as shown in Table 23. Other fields include:

CSCs_Full: Shows any CSCs in a full state.

CSCs_Alive: Shows the DDU fiber inputs that are connected to active CSCs.

CSC_data_available: Shows which of the active CSCs have track-stub data for this event.

CSC_count: The number of active CSCs with data for this event.

BoE_TTS: The DDU TTS status at the beginning of event.

Q: Set high when a CSC has reached a full state.

CSC_error: Shows any active CSCs in an error state.

CSC_warn: Shows any active CSCs in a warning state.

During event processing the DDU monitors the incoming data to check the synchronization, word counts and CRCs, as well as warning and error conditions that may be set for the CSC. After the sync check is completed, any CSCs with no muon track stubs are excluded by zero-suppression logic, and those with track stubs are flagged in the "CSC_data_available" field of the DDU Header. A summary of results from all DDU checks is included in the DDU headers and trailers that wrap each event fragment (see Tables 23 through 26).

Table 25: EMU DDU Beginning of Event Status Bit Definitions.

Bit	BoE Status Description	Condition
6	DDU Clock-DLL Error: the DDU lost it's clock for an unknown period of time; some data may be lost	may be OK
5	DDU FIFO Full Error: an unknowable amount of data has been lost	RESET
4	DDU detected Fiber Error: bad fiber connection status	RESET
3	DDU detected Critical Error: OR of all "RESET" cases, held until RESET	RESET
2	DDU detected Single Error: OR of all "Bad Event" cases at BoE	RESET
1	DDU detected Single Error: DDU L1A event number match failed for 1 or more CSCs	RESET
0	DDU Start Timeout Error: data from a CSC never arrived	RESET

Table 26: EMU DDU Event Status Bit Definitions.

Bit	Event Status Description	Condition
31	CSC LCT/DAV Mismatch occurred: CSC trigger and data do not match	bad event
30	DDU CFEB L1 Number Mismatch occurred	bad event
29	DDU detected no good DMB CRCs: empty event or bit-error	may be bad event
28	DDU CFEB Count Error occurred: missing CFEB data	bad event
27	DDU Bad First Data Word from CSC: RX or bit-error	bad event
26	DDU L1A-FIFO Full Error	RESET
25	DDU Data Stuck in FIFO Error: data not associated with a trigger, sync error	RESET
24	DDU NoLiveFibers Error: the DDU detects no CSC fibers	may be OK
23	DDU Control Word Inconsistency Warning: RX or bit-error	may be bad event
22	DDU Input FPGA Error	bad event
21	DCC Stop Bit set	OK
20	DCC Link Not Ready	may be OK
19	DDU detected TMB Error	bad event
18	DDU detected ALCT Error	bad event
17	DDU detected TMB or ALCT Word Count Error	bad event
16	DDU detected TMB or ALCT L1A Number Error	bad event
15	DDU detected Critical Error: OR of all "RESET" cases, held until RESET	RESET
14	DDU detected Single Error: OR of all "Bad Event" cases	bad event
13	DDU Single Warning: OR of bits 23,10	may be bad event
12	DDU FIFO Near Full Warning: OR of "near Full" from all sources	OK
11	RX Error: one or more CSCs violated 64-bit word boundary	bad event
10	DDU Clock-DLL Error: the DDU lost it's clock for an unknown period of time; some data may be lost	may be OK
9	DDU detected a CSC Error: Timeout, DMB CRC, CFEB Sync/Overflow, or missing CFEB data	bad event
8	DDU Lost In Event Error: lost or mis-sequenced data was detected	bad event
7	DDU Lost In Data Error: monitor logic hung by lost or mis-sequenced data	bad event, RESET
6	DDU Timeout Error: CSC end of data not detected	bad event, RESET
5	DDU detected TMB or ALCT CRC Error	bad event
4	DDU Multiple Transmit Errors: several RX or bit-errors occurred on a CSC	bad event, RESET
3	DDU Sync Lost/Buffer Overflow Error	bad event, RESET
2	DDU detected Fiber Error: bad fiber connection status	RESET
1	DDU detected DMB L1A Match Error: bit-error or sync error	bad event
0	DDU detected CFEB CRC Error: bit-error or wordcount error	bad event

The volume of DDU data will vary from event-to-event depending on the number of CSCs with track stubs. In the frequent case where there are no CSCs with track stubs in a DDU, it will still send the DDU header and trailer blocks (48 bytes).

8.3 EMU DMB Format

The Data Acquisition Motherboard collects the data from every source on the CSC, including 4-5 CFEBs, 1 ALCT and 1 TMB. When the L1A is received, each source signals the DMB if it has track-stub data for the event, then transmits its 16-bit data to the DMB where it is buffered in a FIFO array. For those sources with data, the DMB reads the event record from each source in sequence (see Table 27) and sends the data to the DDU via 1.6 Gb/s fiber channel. Additionally, CSC buffer status, geometry and event information, and CRC check words are contained within the DMB header and trailer blocks.

Table 27: EMU DMB data format with track stub present.

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
9h				DMBbx[11:0]												9h				Data_available_flags_A												9h				DMB_L1A[23:12]												9h				DMB_L1A[11:0]											
Ah				DMB_sync				DMB_L1A[7:0]								Ah				CFEB_movlp				DMBbx[6:0]								Ah				DMB_Crate								DMB_ID				Ah				Data_available_flags_B											
ALCT_Data_Payload (see Table 35)																																																																					
TMB_Data_Payload (see Tables 30 to 34)																																																																					
CFEB_Data_Payload (0-5 CFEBs, see Table 29 and Section 8.4)																																																																					
Fh				DMB_timeout_flags												Fh				DMB_FIFO_flags											DMB time	Fh				CFEB_movlp				DMB_warn_flags				Fh				DMBbx[3:0]				DMB_L1A[7:0]											
Eh				P	DMB_CRC[21:11]											Eh				P	DMB_CRC[10:0]											Eh				DMB_Crate								DMB_ID				Eh				FIFO_flags											
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

370 P: DMB CRC Word parity flag.

Table 28: EMU DMB data format with no track stubs present.

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8h				DMBbx[11:0]												8h				000h												8h				DMB_L1A[23:12]												8h				DMB_L1A[11:0]											
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The DMB headers and trailers carry several fields of interest for analysis:

Data_available_flags_(A/B): These words tell which CSC sources have track-stub data available (DAV); some bits are repeated for voting at the DDU level.

A = TMB_DAV(1) + ALCT_DAV(1) + CFEB_CLCT_SENT(5:1) + CFEB_DAV(5:1)
B = TMB_DAV(1) + CLCT-DAV-Mismatch(1) + ALCT_DAV(1) + CLCT-DAV-Mismatch(1)
+ TMB_DAV(1) + CLCT-DAV-Mismatch(1) + ALCT_DAV(1) + CFEB_DAV(5:1)

DMB_L1A: equivalent to LV1_id in the CMS Common Data Format.

DMBbx: equivalent to the BX_id.

DMB_Crate + **DMB_ID**: these identify which CSC the data comes from.

DMB_timeout_flags: identifies CSC sources missing the Start or End of data as indicated below (12-bits).

CFEB_End_Timeout(5:1) + ALCT_End_Timeout(1) + TMB_End_Timeout(1)
+ CFEB_Start_Timeout(5:1)

DMB time: two additional timeout flags; ALCT_Start_Timeout(1) + TMB_Start_Timeout(1)

The volume of DMB data will vary from event-to-event depending on the number of sources with track-stub data. In the frequent case where there are no sources with track-stub data in the CSC (empty CSC condition), the DMB will send a special 64-bit word (see Table 28) with the minimal information required to maintain synchronization with the DDU. This information is checked and summarized at the DDU, then the empty CSC words are excluded from the data stream by zero-suppression logic.

8.4 EMU CFEB Format

The Cathode Front-End Boards send 16-bit data words in multiples of 4 (as do all CSC sources), and the 16th bit is reserved as a special control word flag which is set to zero for all normal data words. Each CFEB has three possible responses to an L1A:

1. Send nothing: this CFEB has no track-stub data.
2. Send the Data Available signal, followed by SCA time-sample data.
3. As above, but some time samples are skipped due to temporary "SCA Full" condition.

For the second case (normal data transmission), each CFEB data stream is composed of 100 16-bit words per time sample. This includes 96 SCA data words for the 12-bit digitized cathode strip data, plus 4 additional words at the end of each time sample: a CRC word, 2 status words and an end marker (not guaranteed unique). Therefore, the full 16-bit word count per CFEB (CF_{WC}) depends on the number of time samples (N_{ts}) digitized:

CF_{WC} = {[(16 strips) * (6 layers)] + 1 CRC + 2 CFEB_Info + 1 End_marker} * N_{ts}

N_{ts} (number of time samples) is defined in the firmware or through slow control, typically set to 8 for LHC running (may be 16 for calibration runs) yielding CF_{WC}(typical)=800. However, in case of a temporary SCA Full condition the CF_{WC} may deviate from this scheme (case 3 above), and any time samples with no SCA data have their 100 data words replaced by a 16-bit "B-code" word repeated 4 times:

SCA_Full word format: B2h + Block_number(4) + FIFO_word_count(4)

SCA Full is a temporary condition that can only occur during an unusually active burst of triggers; the CFEB will automatically recover from this and no Resets are required.

The 96 SCA data words in every time sample follow the format in Table 29, where the definitions for the last 4 CFEB words are also found.

Table 29: EMU CFEB-SCA data format.

CFEB Data Bit	SCA Data Word Bit Definition
15	Always LOW for data words, HIGH for SCA_Full words.
14	Overlap sample flag, normally HIGH; set LOW when two LCTs share a time sample.
13	Serialized 16-bit CFEB-SCA controller status (see below).
12	Out of range flag from CFEB ADC.
11-0	12-bit Digitized SCA ADC data.

Word 97: CRC_Word(15) created using a CRC-15 algorithm.
Word 98: 7h + L1PIPE_EMPTY(1) + LCTPIPE_EMPTY(1) + L1PIPE_FULL(1)
+ LCTPIPE_FULL(1) + LCTPIPE_CNT(4) + NF_SCA(4)
Word 99: 7h + DMB_L1A(6) + L1PIPE_CNT(5) + L1PIPE_WARN(1)
Word 100: 7FFFh (end of sample marker).

L1PIPE and LCTPIPE refer to CFEB-SCA Controller internal pipeline status; NF_SCA is the number of free SCA blocks (12 max).

The relation between CSC geometry and the data words from a single CFEB is described by the following series of nested loops:

```
do (N_ts time samples){
    do (Gray code loop over 16 CSC Strips; S=0,1,3,2,6,7,5,4,12,13,15,14,10,11,9,8){
        do (loop over 6 CSC Layers; L=3,1,5,6,4,2){
            SCA data word
        }
    }
    CRC Word
    CFEB Word 98
    CFEB Word 99
    CFEB Word 100
}
```

Bit 13 of each SCA data word carries the serialized 16-bit CFEB-SCA Controller status word, containing trigger and SCA information. This data word is serialized (LSB first) with one bit in each of the 16 strips read out, yielding the 16-bit word. Due to the innermost-loop over the 6 CSC layers, each bit of a 16-bit Controller status word is actually sent 6 times in a row (once for each layer). Since there are 16 strips read out for each time sample, the complete Controller status word can be reconstructed independently in every time sample. The 16-bit word is defined as follows (highest-to-lowest bit):

Controller status: TS_FLAG(1) + FULL_SCA(1) + LCT_PHASE(1) + L1A_PHASE(1)
+ SCA_BLK(4) + TRIG_TIME(8)

TRIG_TIME indicates the starting capacitor in the 8-capacitor SCA block (lowest bit is the first capacitor, highest bit the eighth capacitor). SCA_BLK is the SCA Capacitor block used for this time sample. L1A_PHASE and LCT_PHASE show the phase of the 50ns CFEB digitization clock at the time the trigger was received (1=clock high, 0=clock low). FULL_SCA indicates there were no free SCA blocks left at digitization time. The TS_FLAG bit indicates the number of time samples to digitize per event; high=16 time samples, low=8 time samples.

8.5 EMU TMB Format

The Trigger Motherboard receives raw cathode hit information from the CFEBs and searches for high-p_t patterns. It combines the cathode result with anode patterns received from the ALCT board to identify muon track stubs in the CSC. The muon track information is then sent to the Level-1 trigger system. If an L1A is issued for a muon track, the TMB sends the cathode hit data to the DMB; anode hit data from the ALCT is passed to the DMB via the TMB.

Table 30: EMU TMB data format, Short Header Only mode.

Word #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	6h			B0Ch											
1	0	FIFO_Mode*1			CFEBs_in_Readout							Time_bins				
2	0	L1_type*2		Board_ID					CSC_ID				TMB_L1A[3:0]			
3	0	x	r_type*3		TMBbx[11:0]											
4	Dh				1	CRC22[10:0]										
5	Dh				1	CRC22[21:11]										
6	Dh				EEFh											
7	Dh				1	TMB_word_count[10:0]										

*1 See Table 31.
*2 See Table 32.
*3 See Table 33.

The TMB has several options for data format that can be selected based on data rate considerations. The shortest is the TMB Short Header Only format, which is shown in Table 30. In this mode there is no raw cathode trigger hit (Triad) information sent.

The longest format is the TMB Full Readout option (Table 34), which contains the cathode processing results as well as the raw cathode trigger Triad data. Intermediate options include Local Triad Readout, where only those Triads near the track stub are read out, and TMB Full Header Only, which is the same as the Full Readout with none of the Triad data. In all of these cases there are 2 optional words available near the end to maintain 4-word multiples as needed.

Table 31: FIFO Mode Definitions.

FIFO_Mode	Raw Trigger Hits	Header Size
0	None	Full
1	Full	Full
2	Local	Full
3	None	Short
4	None	None

Table 32: L1 Type Definitions.

L1_type	Comment
0	Normal CLCT trigger with cathode data and L1A match
1	ALCT only trigger, no cathode data and no readout
2	L1A only, no cathode or anode data and no readout
3	LCT triggered, but no L1A match and no readout

8.6 EMU ALCT Format

Currently the ALCT has only the Full Readout data format option (Table 35), which reads out the wire group hit data for all LCT chips. It is expected that ALCT will add a smaller Local Readout option in the future, such that only wire group hits near the track stub are read out.

8.7 EMU CSC Trackfinder

9 RPC Data Format

The RPC data format.

Table 33: Record Type Definitions.

r_type	Raw Trigger Hits	Header Size
0	None	Full
1	Full	Full
2	Local	Full
3	None	Short (no buffer available at pre-trigger)

Table 34: EMU TMB data format, full readout mode.

Word #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	6h			B0Ch											
1	0	FIFO_Mode			CFEBs_in_Readout							Time_bins				
2	0	L1_type		Board_ID					CSC_ID			TMB_L1A[3:0]				
3	0	x	r_type		TMBbx[11:0]											
4	0	x	Tbin_before_pretrig					has buf	# CFEBs			# Header_Frames				
5	0	xx		has pretrg	Trig_source_vector								L1A_Tx			
6	0	x	Run_ID			CFEBs_instantiated						CFEB_LCTs				
7	0	xx		Sync Err	BX_at_LCT[11:0]											
8	0	Cathode_LCT0[14:0]														
9	0	Cathode_LCT1[14:0]														
10	0	xx		invalid patt.	Cathode_LCT1[20:15]						Cathode_LCT0[20:15]					
11	0	Triad_persistence				ALCT_match_time				ΔBX_LCT1	ΔBX_LCT0		CLCT Only	ALCT Only	TMB Match	
12	0	MPC_muon0_frame0[14:0]														
13	0	MPC_muon0_frame1[14:0]														
14	0	MPC_muon1_frame0[14:0]														
15	0	MPC_muon1_frame1[14:0]														
16	0	x	ds_thresh			hs_thresh			xx		MPC_accept	Muon_Frame_MSBs				
17	0	x	Buffer_write_flags													
18	0	xx	Buffers_busy													
19	0	x	Buffer_read_flags													
20	0	Timeout			Discard_flags											
21	0	x	Firmware_revision_code													
22	0	6h			E0Bh											
	0	CFEB# (0-4)			Tbin# (0-n, n<7)				Triad data, 8-bits per layer (0-5)							
⋮																
	0	6h			E0Ch											
	0	0h			AAAh (as needed to maintain 64-bit boundary											
	0	0h			555h (as needed to maintain 64-bit boundary											
N-3	Dh				1	CRC22[10:0]										
N-2	Dh				1	CRC22[21:11]										
N-1	Dh				E0Fh											
N	Dh				1	TMB_word_count[10:0]										

Table 35: EMU ALCT data format, full readout.

Word #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	6h			0	Board_ID			CSC_ID				ALCT_L1A[3:0]			
1	0	0	xxx			2nd	1st	Ext	L1A	Time_bins					FIFO_Mode	
2	0	0	xx		ALCTbx[11:0]											
3	0	0	ActiveLCTchips						LCTchips_read_out							
4	0	LCT0[14:0]														
5	0	LCT0[29:14]														
6	0	LCT1[14:0]														
7	0	LCT1[29:14]														
	0	LCTchip# (0-6)			Tbin# (0-n, n<5)				Wire Group data, 2 8-bit words per layer (0-5)							
⋮																
N-3	Dh				0	CRC22[10:0]										
N-2	Dh				0	CRC22[21:11]										
N-1	Dh				E0Dh “Evener” word											
N	Dh				0	ALCT_word_count[9:0]										

References

[1] **Proc. 9th LECC 2003 Workshop, Amsterdam, Holland, September 2003**, N. Almeida *et al.*, *Data Concentrator Card and Test System for the CMS ECAL Readout*.

[2] **Proc. Int. Conf. Calorimetry in HEP, Perugia, Italy, March 2004**, R. Alemany *et al.*, *ECAL Off-Detector Electronics*.

[3] **Proc. IEEE Nuclear Science Symposium, Rome, October 2004**, N.Almeida *et al.*, *The Selective Read-Out Processor for the CMS Electromagnetic Calorimeter*.

[4] **Proc. 6th Workshop on Electronics for LHC Experiments, Cracow, Poland, September 2000**, J.Varela, *Timing and Synchronization in the LHC Experiments*.

[5] The TriDAS Horizontal project Web site: http://cmsdoc.cern.ch/cms/TRIDAS/horizontal.

[6] **CMS NOTE 2002/033**, CMS Trigger DAQ group, *CMS L1 Trigger Control System*.

[7] The ORCA project Web site: http://cmsdoc.cern.ch/orca.