Definition uGT readout record

Bernhard Arnold, Herbert Bergauer, Manfred Jeitler
Institute of High Energy Physics (HEPHY)
http://www.hephy.at
http://globaltrigger.hephy.at
https://twiki.cern.ch/twiki/bin/viewauth/CMS/GlobalTrigger

July 3, 2023

Contents

1 Definition uGT readout record

This document is a description of the structure of a uGT readout record.

In chapter "Event Builder Overview" of document [1] one can find a description about building a uGT readout record on AMC13.
The following description and explanations are based on documents [1] and [2].

1.1 Overview of structure

In the following table the structure of the uGT readout record is shown:

- the first 8 lines contain AMC13 header information,
- followed by 3 AMC#1 header lines.
- Then the first block of input data (of AMC#1), containing a block header and 30 data words (32 bits), is shown.
- A certain number of input data blocks and output data blocks (each with the same block size) and a AMC#1 trailer line finish AMC#1 readout record.
- Data of AMC#2 to AMC#6 with output data blocks only
- and 2 AMC13 trailer lines fill up the readout record.

<table>
<tr><th colspan="3">63</th><th colspan="4">60</th><th colspan="4">56</th><th colspan="4">52</th><th colspan="4">48</th><th colspan="4">44</th><th colspan="4">40</th><th colspan="4">36</th><th colspan="4">32</th><th colspan="4">28</th><th colspan="4">24</th><th colspan="4">20</th><th colspan="4">16</th><th colspan="4">12</th><th colspan="4">8</th><th colspan="4">4</th><th colspan="1">0</th></tr>
<tr><td colspan="4">0x5</td><td colspan="4">Ev_t</td><td colspan="24">LV1_id</td><td colspan="12">BXId</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="4">uFOV</td><td colspan="4">Res</td><td colspan="4">nAMC</td><td colspan="16">Reserved</td><td colspan="32">OrN</td><td colspan="4">0x0</td></tr>
<tr><td>0</td><td>L</td><td>M</td><td>S</td><td>E</td><td>P</td><td>V</td><td>C</td><td colspan="24">AMC1_size</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Blk_No</td><td colspan="4">AmcNo</td><td colspan="16">BoardID</td></tr>
<tr><td colspan="64">...</td></tr>
<tr><td>0</td><td>L</td><td>M</td><td>S</td><td>E</td><td>P</td><td>V</td><td>C</td><td colspan="24">AMC6_size</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Blk_No</td><td colspan="4">AmcNo</td><td colspan="16">BoardID</td></tr>
<tr><td colspan="4">0x0</td><td colspan="4">AmcNo
AMC1</td><td colspan="24">LV1_id</td><td colspan="12">BXId</td><td colspan="20">Data_lgth</td></tr>
<tr><td colspan="32">User</td><td colspan="16">OrN</td><td colspan="16">BoardID</td></tr>
<tr><td colspan="32">uGT build</td><td colspan="32">MP7 FW version</td></tr>
<tr><td colspan="32">input data</td><td colspan="8">BlockID=0x00</td><td colspan="8">Block size</td><td colspan="16">reserved</td></tr>
<tr><td colspan="32">input data</td><td colspan="32">input data</td></tr>
<tr><td colspan="64">...</td></tr>
<tr><td colspan="8">BlockID=0x02</td><td colspan="8">Block size</td><td colspan="16">reserved</td><td colspan="32">input data</td></tr>
<tr><td colspan="64">additional 11 input data blocks AMC1</td></tr>
<tr><td colspan="32">output data</td><td colspan="8">BlockID=0x21</td><td colspan="8">Block size</td><td colspan="16">reserved</td></tr>
<tr><td colspan="32">output data</td><td colspan="32">output data</td></tr>
<tr><td colspan="64">...</td></tr>
<tr><td colspan="8">BlockID=0x23</td><td colspan="8">Block size</td><td colspan="16">reserved</td><td colspan="32">output data</td></tr>
<tr><td colspan="64">additional 8 output data blocks AMC1</td></tr>
<tr><td colspan="4">0x0</td><td colspan="4">AmcNo
AMC2</td><td colspan="24">LV1_id</td><td colspan="12">BXId</td><td colspan="20">Data_lgth</td></tr>
<tr><td colspan="32">User</td><td colspan="16">OrN</td><td colspan="16">BoardID</td></tr>
<tr><td colspan="32">uGT build</td><td colspan="32">MP7 FW version</td></tr>
<tr><td colspan="64">9 output data blocks AMC2</td></tr>
<tr><td colspan="64">blocks for AMC3 ... AMC6</td></tr>
<tr><td colspan="32">CRC 32 (AMCs)</td><td colspan="8">LV1_id</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="20">Data_lgth</td></tr>
<tr><td colspan="32">CRC 32</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Blk_No</td><td colspan="8">LV1_id</td><td colspan="12">BXId</td></tr>
<tr><td colspan="4">0xA</td><td></td><td></td><td></td><td></td><td colspan="24">Evt_lgth</td><td colspan="16">CRC</td><td>C</td><td>F</td><td>x</td><td>x</td><td colspan="4">Ev stat</td><td colspan="4">TTS</td><td>T</td><td>R</td><td>$</td><td>$</td></tr>
</table>

Line 1 - AMC13 header ("CDF header"):

63 60	59 56	55 32
0x5	Ev_t	LV1_id

31 20	19 8	7 4	3 0
BXId	Source_id	FOV	Hx$$

[63:60] = fixed marker 0x5 (begin of readout record)

[59:56] = "Ev_t" (4) means event type (1 = physic events ?)

[55:32] = "LV1_id" (24) means "Level 1 ID" (hardware event number)

[31:20] = "BXId" (12) means bunch crossing number @ L1A

[19:8] = "Source_id" (12) is FED #

[7:4] = "FOV" (4) means "Format version" ?

[3:0] = "Hx$$" (4) ?

Line 2 - AMC13 header:

63 60	59 56	55 52	51 36
uFOV	Res	nAMC	Reserved

35 4	3 0
OrN	0x0

[63:60] = "uFOV" (4) means "Format version" = 0x1 (0 in all previous firmwares)

[59:56] = "Res" (4) reserved bits

[55:52] = "nAMC" (4) numbers of AMC in readout record

[51:36] = "Reserved" (16) reserved bits

[35:4] = "OrN" (32) orbit number @ L1A

[3:0] = 0x0

Line 3 - AMC13 header (for AMC #1):

63							56	55 32
0	L	M	S	E	P	V	C	AMC1_size

31 28	27 20	19 16	15 0
0x0	Blk_No	AmcNo	BoardID

[63:56] = 8 bits with leading 0 and 7 bits for L, M, S, E, P, V, C. See [1] pg. 5 for details

[55:32] = "AMC1_size" (24) readout record length of AMC #1

[31:28] = 0x0

[27:20] = "Blk_No" (8) ?

[19:16] = "AmcNo" (4) number of first AMC in record

[15:0] = "BoardID" (16) ?

Line 4 - AMC13 header for AMC #2
Line 5 - AMC13 header for AMC #3
Line 6 - AMC13 header for AMC #4
Line 7 - AMC13 header for AMC #5
Line 8 - AMC13 header for AMC #6

Line 9 - AMC#1 header:

63 60	59 56	55 32
0x0	AmcNo AMC1	LV1_id

31 20	19 0
BXId	Data_lgth

[63:60] = 0x0

[59:56] = "AmcNo AMC1" (4) number of first AMC in record

[55:32] = "LV1_id" (24) means "Level 1 ID" (hardware event number)

[31:20] = "BXId" (12) means bunch crossing number @ L1A

[19:0] = "Data_lgth" (20) means readout record length of AMC #1

Line 10 - AMC#1 header:

63 40	39 32
User	Event type

31 16	15 0
OrN	BoardID

[63:40] = "User" (24) ?

[39:32] = "Event type" (8) is set in configuration key: "ugt infra mp7 base" with

`<param cmd="roLoadMenu" id="mode1:eventType" type="uint">0xc0</param>`

[31:16] = "OrN" (16) least significant 16 bits of orbit number @ L1A

[15:0] = "BoardID" (16) ?

Line 11 - AMC#1 header:

63 48	47 32
0x0000	uGT build

31 0
MP7 FW version

[63:48] = 0x0000

[47:32] = "uGT build" (16) uGT firmware build number

[31:0] = "MP7 FW version" (32) (e.g.: 0x00030202 means versin 3.2.2)

Line 12 - AMC#1 uGT input data (in readout record):

63 32
input data

31 24	23 16	15 12	11 8	7 0
BlockID	Block size	res	bank id	reserved

[63:32] = "input data" - muon data, link1, mgt channel 0x00, frame 0 (free), bx-2)

[31:24] = "BlockID" (8) mgt channel number of the link (=0x00, marker of first block of "input data")

[23:16] = "Block size" (8) from: "ugt infra mp7 base" with

```
<param cmd="roLoadMenu" id="mode0:capture0:readoutLength" type="uint">30</param>
```

which is 0x1e. (Similar commands for other modes and captures)

[15:12] = "res" (4) means reserved

[11:8] = Contains probably the "bank id" which is set in configuration key: "ugt infra mp7 base" with

```
<param cmd="roLoadMenu" id="mode1:capture1:bankId" type="uint">2</param>
```

for input data and

```
<param cmd="roLoadMenu" id="mode1:capture0:bankId" type="uint">1</param>
```

for output data

[7:0] = "reserved" (8)

Line 13 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 2 (MU0[31:0]), bx-2
Line 13 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 1 (MU0 & MU1 eta raw), bx-2
Line 14 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 4 (MU1[31:0]), bx-2
Line 14 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 3

(MU0[63:32]), bx-2
Line 15 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 0 (free), bx-1
Line 15 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 5 (MU1[63:32]), bx-2
Line 16 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 2 (MU0[31:0]), bx-1
Line 16 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 1 (MU0 & MU1 eta raw), bx-1
Line 17 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 4 (MU1[31:0]), bx-1
Line 17 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 3 (MU0[63:32]), bx-1
Line 18 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 0 (free), bx
Line 18 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 5 (MU1[63:32]), bx-1
Line 19 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 2 (MU0[31:0]), bx
Line 19 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 1 (MU0 & MU1 eta raw), bx
Line 20 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 4 (MU1[31:0]), bx
Line 20 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 3 (MU0[63:32]), bx
Line 21 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 0 (free), bx+1
Line 21 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 5 (MU1[63:32]), bx
Line 22 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 2 (MU0[31:0]), bx+1
Line 22 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 1 (MU0 & MU1 eta raw), bx+1
Line 23 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 4 (MU1[31:0]), bx+1
Line 23 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 3 (MU0[63:32]), bx+1
Line 24 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 0 (free), bx+2
Line 24 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 5 (MU1[63:32]), bx+1
Line 25 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 2 (MU0[31:0]), bx+2
Line 25 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 1 (MU0 & MU1 eta raw), bx+2
Line 26 [63:32] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 4 (MU1[31:0]), bx+2

Line 26 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 3 (MU0[63:32]), bx+2
Line 27 [63:32] - AMC#1 uGT input data - "BlockID=0x02 (8), "Block size" (8), "res" (4), "bank id=0x2" (4), "reserved" (8)
Line 27 [31:24] - AMC#1 uGT input data - muon data, link1, channel 0x00, frame 5 (MU1[63:32]), bx+2

See an example for such a data block (line 12 - 27) in 1.3.

Blocks for other input data and output data fill up the readout record of AMC#1.

MISSING:
– AMC#1 trailer descr.
– AMC#2 - AMC#6 output data only.
– AMC13 trailer.
– AMC13 trailer ("CDF trailer").

1.2 Readout record dump

Example of a readout record dump (with description):

```
5186f41767557c08 - AMC13 header:  Event type, LV1 id, BX id, FED#
10604240512ff300 - AMC13 header:  AMCs, Orbit nr (32 bits)
0f00014a00010000 - AMC13 header:  Data length, AMC#1
0f00009000020000 - AMC13 header:  Data length, AMC#2
0f00009000030000
0f00009000040000
0f00009000050000
0f00009000060000
0186f4176750014a - AMC#1 header:  AMC#1, LV1 id, BX id, Data length
000000c0ff300000 - AMC#1 header:  Ev.  type [0], Orbit nr (16 bits)
0000115200030202 - AMC#1 header:  uGT FW build, MP7 FW version
00000000001e0200 - AMC#1 input:  Block ID, Block size, "bankId"=input
................  - AMC#1 muon input data
021e020000000000 - AMC#1 input:  Block ID, Block size, "bankId"
................  - AMC#1 muon input data
00000000041e0200
................  - AMC#1 muon input data
061e020000000000
................  - AMC#1 muon input data
06000200081e0200
................  - AMC#1 e/gamma input data
0a1e020006000200
................  - AMC#1 e/gamma input data
000008000c1e0200
................  - AMC#1 jet input data
0e1e020000000800
................  - AMC#1 jet input data
02000200101e0200
................  - AMC#1 tau input data
121e020002000200
................  - AMC#1 tau input data
0002a02a141e0200
................  - AMC#1 esums input data
181e02000006f000
................  - AMC#1 ext cond input data
00000000211e0100 - AMC#1 output:  Block ID, Block size, "bankId"=output
................  - AMC#1 algo output data
231e010000000000
................  - AMC#1 algo output data
00000000251e0100
................  - AMC#1 algo output data
271e01005ca70bcc
................  - AMC#1 algo output data
```

```
00000000291e0100
................  - AMC#1 algo output data
2b1e010000000000
................  - AMC#1 algo output data
000000002d1e0100
................  - AMC#1 algo output data
2f1e010000000000
................  - AMC#1 algo output data
00000000311e0100
................  - AMC#1 algo output data
9ac23b761700014a - AMC#1 trailer:  CRC, LV1 id (8 bits), Data length
0286f41767500090 - AMC#2 header
000000c0ff300000 - AMC#2 header
0000115200030202 - AMC#2 header
00000000211e0100
................  - AMC#2 algo output data
231e010000000000
................  - AMC#2 algo output data
00000000251e0100
................  - AMC#2 algo output data
271e01005ca70bcc
................  - AMC#2 algo output data
00000000291e0100
................  - AMC#2 algo output data
2b1e010000000000
................  - AMC#2 algo output data
000000002d1e0100
................  - AMC#2 algo output data
2f1e010000000000
................  - AMC#2 algo output data
00000000311e0100
................  - AMC#2 algo output data
cb79a76317000090 - AMC#2 trailer
0386f41767500090 - AMC#3 header
000000c0ff300000 - AMC#3 header
0000115200030202 - AMC#3 header
00000000211e0100
................  - AMC#3 to AMC#6
36cb696317000090 - AMC#6 trailer
f6b9461200017675 - AMC13 trailer:  CRC, LV1 id (8 bits), BX id
a000042484680000 - AMC13 trailer:  Ev.  length, CRC (16 bits)
```

1.3 Data block

Description of a block with "block size" = 0x1e (6 frames [32 bits] @ +/-2 bx = 30) of a certain "Block ID" (0x00, 0x02, ..., 0x21, 0x23, ...):

```
00000000001e0200 - bx-2:  frame 0
0000000000000000 - bx-2:  frame 2, frame 1
0000000000000000 - bx-2:  frame 4, frame 3
0000000000000000 - bx-1:  frame 0, bx-2:  frame 5
0000000000000000 - bx-1:  frame 2, frame 1
0000000000000000 - bx-1:  frame 4, frame 3
0000000000000000 - bx:  frame 0, bx-1:  frame 5
0000000000000000 - bx:  frame 2, frame 1
0000000000000000 - bx:  frame 4, frame 3
0000000000000000 - bx+1:  frame 0, bx:  frame 5
0000000000000000 - bx+1:  frame 2, frame 1
0000000000000000 - bx+1:  frame 4, frame 3
0000000000000000 - bx+2:  frame 0, bx+1:  frame 5
0000000000000000 - bx+2:  frame 2, frame 1
0000000000000000 - bx+2:  frame 4, frame 3
021e020000000000 - bx+2:  frame 5
```

Remark:
The order of +/-2 bx has to be verified!

References

[1] AMC13 Event Builder:
http://ohm.bu.edu/~hazen/CMS/AMC13/UpdatedDAQPath_2014-07-10.pdf 1, 1.1

[2] MP7 Readout & DAQ:
https://github.com/cms-l1-globaltrigger/mp7_ugt_legacy/blob/master/doc/read_out_record/MP7Readout.pdf 1