

ALICE TRD Raw Data Format Specification

K. Oyama, C. Adler, V. Angelov, D. Emschermann, and C. Lippmann

Last update: Dec 28, 2007

Abstract

This document describes the data format produced by the ALICE TRD. The data corresponds to the payload packed into the DDL data stream from the TRD Global Tracking Units to the DAQ.

1 Introduction

This document describes the data format produced by ALICE Transition Radiation Detector (TRD). TRD chamber produces two types of data; tracklet data and raw data. Both tracklet and raw data are sent through ORI and collected in Global Tracking Unit (GTU). The tracklet data is sent for each Level-0 trigger accept and raw data is sent for each Level-1 trigger accept. The tracklet data is kept until corresponding Level-1 decision comes. If there is no corresponding Level-1 decision, the tracklet data will be discarded. If there is corresponding Level-1 trigger accept, the GTU will find appropriate tracklet data from its memory, combine with raw data and ship them to DAQ system through DDL connection. This document describes about the data format contained in DDL data stream. The data format is described in top-down hierarchy where the overall data format from GTU to DAQ are described first and description about finer structures follow.

2 Acronyms and Technical Terms

DAQ The Data Acquisition system.

GTU Global Tracking Unit. Receives the data from TRD chambers. It makes Level-1 local trigger decision, and also forward the raw data to the DAQ system.

ORI Optical Readout Interface mounted on the TRD chamber. It receives data from MCMs and send to the GTU.

HC Half Chamber ... Half of one TRD chamber corresponds to 3 or 4 readout boards for C0 and C1 chamber, respectively. One ORI corresponds always to one HC.

TRAP Tracklet Processor. The digital part of the MCM.

MCM Multichip Module. Consists of the analog PASA and the digital TRAP and ADC.

PASA Preamplifier Shaper Amplifier.

tracklet Particle track information calculated in TRAP chip using the straight line fit for the ADC data. Tracklet is used in GTU for local Level-1 trigger decision.

3 GTU to DAQ Data Format

The data packed into the DDL data stream from GTU to DAQ system for an event has the structure shown in the Table 1 where “v” means that size is not fixed (variable) and “n.f.” means contents are not constant. Size of the data is given always in number of 32 bits words.

The data contains only the data to be packed in DDL format as a payload. Necessary header and so on attached by DDL before and after the payload is not shown. Precise description about the DDL header and so on are found in the DATE manual.

The half chamber data (HcData) are sent from all active half chambers (nominally 60 per super-module). Number of half chamber can be obtained from GtuLinkMask described below.

| Size | Value | Content |
|------|-------|-----------------|
| 5 | n.f. | GtuLinkMask |
| v | n.f. | HcData for L0S0 |
| v | n.f. | HcData for L1S0 |
| v | n.f. | HcData for L2S0 |
| : | : | : |
| v | n.f. | HcData for L5S4 |

Table 1: TRD GTU Data Format. The order of HcData here is stack oriented however it is not requirement. The raw data reader software must be prepared so that it can accept any order.

The ordering of the HcData are not guaranteed to be in the order of layer and stack. The producer of the HcData can be obtained by HcData itself. The detail of data contents in GtuLinkMask and HcData are given in the sections below.

4 GTU Link Mask

These five 32bit data words are introduced by the GTU into the data stream. Each of these five words mark for a certain chamber stack (0:4) the status of the corresponding optical links from each of the ORI boards on the half chambers:

$$GTULinkMask = 1110 : 0000 : 0000 : 0000 : 0000 : gggg : gggg : gggg . \quad (1)$$

Here the LSB corresponds to layer 0, side A, LSB+1 to layer 0, side B and finally bit 11 corresponds to layer 5, side B. If the bit is set, this means that the GTU receives optical power from the ORI board through the optical fibers from the super module. The first word corresponds to stack 0, the fifth word to stack 4.

5 Data Format from Half Chamber

| Size | Value | Content |
|-------|------------|----------------------|
| v | n.f. | TrackletData |
| 1 | 0x10001000 | EndOfTracklet |
| 1 | 0x10001000 | EndOfTracklet |
| 1 - 8 | n.f. | HC-Header |
| v | n.f. | TrapData (first MCM) |
| : | : | : |
| v | n.f. | TrapData (last MCM) |
| 1 | 0x00000000 | EndOfRawData |
| 1 | 0x00000000 | EndOfRawData |

Table 2: The data format produced by a half chamber (HcData).

Table 2 shows the format of the HcData from one half chamber (HC). The data stream starts from tracklet data without prior header. Tracklet is terminated by EndOfTracklet marker, and then header for this half chamber (HC-Header) is sent. TrapData contains raw data from MCM described below. For long time the value of the EndOfTracklet marker was 0xAAAAAAAA. Number of Track MCM depends on chamber type and number of disabled MCM, but not given in the HC-Header. EndOfRawData shows that there is no more MCM data coming. Both end markers are programmed in the TRAP chip as 16 bit parameters. In order to have more reliability in case of bit errors, the end markers are sent 4 times (x 16 bit) from the TRAP output port. The internal TRAP logic strips the end marker, so they do not propagate in the readout tree. The end markers coming to the GTU are sent by the HC merger.

6 Half Chamber Headers Format

Half Chamber (HC) headers consists of at least two data words: $h[0]$ and $h[1]$, with the possibility to use more. They have the following data structure.

6.1 $h[0]$ Format

The first word of HC header ($h[0]$) is defined by

$$h[0] = xmmm : mmmm : nnnn : nnnq : qqss : sssp : ppcc : ci01, \quad (2)$$

where right hand of the expression is the digits expression. Colons are put just for easier bit position identification.

| Name | n-bits | bit pos | Value | Content |
|------|--------|---------|-------|--|
| x | 1 | 31 | 1 | Raw Version Special Number |
| m | 7 | 30..24 | 0:127 | Raw Version Major Number |
| n | 7 | 23..17 | 0:127 | Raw Version Minor Number |
| q | 3 | 16..14 | 0:7 | Number of additional HC-header words |
| s | 5 | 13..9 | 0:17 | Super-module Sector Number (ALICE numbering) |
| p | 3 | 8..6 | 0:5 | Plane Number (same as layer) |
| c | 3 | 5..3 | 0:4 | Chamber Number (same as stack) |
| i | 1 | 2 | 0:1 | Side on chamber (0:A , 1:B) |

Table 3: $h[0]$ HC header contents.

The MSB “ x ” is to tell the reading software that this is new version of header. Old format (2006 SM-I) can not have 1 as MSB because here was bit 12 (weight 2048) of the DCS serial number which can not happen in SM-I. DCS serial number is then omitted because it is useless.

Please see the Major (m) and Minor (n) numbers definitions of Raw Version in the later sections. The values m and n form full raw version number as $V_r = m.n$. The V_r defines raw data format followed, including $h[1]$. The $h[1]$ definitions in this document must be known as not for all versions but only for specific versions. The value q (number of additional HC-header words) shows the number of following header words. It should not count the $h[0]$ word itself. This can be increased up to 7 (corresponds to total header size of 8) Currently it is either 1 or 2 and $h[1]$, $h[2]$, $h[3]$ are defined. As the three additional headers can be reliably recognized by the bits 5..2, their order is not necessary to be fixed. Furthermore, it is possible to send only a subset of them, provided the total number is given correctly in $h[0]$. Side is the side on the chamber where 0 means A side and 1 means B side. DCS board sits on B side (2B readout board, in ALICE position 3).

6.2 $h[1]$ Format

$$h[1] = tttt : tttt : bbbb : bbbb : bbbb : bbbb : ppph : hh01 . \quad (3)$$

This is valid for $V_r = \dots$

| Name | n-bits | bit pos | Value | Content |
|------|--------|---------|-----------|------------------------|
| t | 6 | 31..26 | 0 : 63 | Number of time-bins |
| b | 16 | 25..10 | 0 : 65535 | Bunch crossing counter |
| p | 4 | 9..6 | 0 : 15 | Pretrigger counter |
| h | 4 | 5..2 | 0 : 11 | Pretrigger phase |

Table 4: $h[1]$ HC header contents.

The Bunch crossing counter runs at 120 MHz and has 32 bits in the TRAP, however only the lower 16 bits are transferred to raw data (b). With 16 bit number, the revolution time is

546 us and it is enough to check the alignment. The Pretrigger counter (p) is the lower 4 bits of the pretrigger counter in the TRAP. Pretrigger Phase (h) is the phase of the pretrigger relative to the ADC sampling. This parameter needs some calibration, because of possible offset and/or sign inversion. The maximum number of time-bins is 30, because of the limited FIFO size in the NI output ports of the TRAP. More time-bins (up to 63) are possible, but the readout in TRAP will require special software and configurations and will definitely cost more electrical power.

6.3 $h[2]$ Format

The optional header word $h[2]$ has the following contents:

$$h[2] = pgtc : nbaa : aaaa : xxxx : xxxx : xxxx : xx11 : 0001 . \quad (4)$$

Note that the bits 5..2 are outside the possible range of the same bits in $h[0]$, $h[1]$ and $h[3]$.

| Name | n-bits | Value | Content |
|------|--------|--------|-------------------------------------|
| p | 1 | 0 : 1 | Pedestal Filter setup (0:off 1:on) |
| g | 1 | 0 : 1 | Gain Filter setup (0:off 1:on) |
| t | 1 | 0 : 1 | Tail Filter setup (0:off 1:on) |
| c | 1 | 0 : 1 | Crosstalk Filter setup (0:off 1:on) |
| n | 1 | 0 : 1 | Non-lin Filter setup (0:off 1:on) |
| b | 1 | 0 : 1 | Raw data bypass filter (0:no 1:yes) |
| a | 6 | 0 : 63 | Digital filter common additive |
| x | | | Undefined |

Table 5: $h[2]$ HC header contents.

Note: this header is not implemented in the TRAP software.

6.4 $h[3]$ Format

The optional header word $h[3]$ has the following contents:

$$h[3] = ssss : ssss : ssss : saaa : aaaa : aaaa : aa11 : 0101 . \quad (5)$$

Note that the bits 5..2 are outside the possible range of the same bits in $h[0]$, $h[1]$ and $h[2]$.

| Name | n-bits | Value | bit pos | Content |
|------|--------|----------|---------|--|
| s | 13 | 0 : 8191 | 31..19 | SVN Revision of the fit and readout program |
| a | 13 | 0 : 8191 | 18..6 | SVN Revision of the assembler used for compilation |

Table 6: $h[3]$ HC header contents.

7 Raw Data Version Definitions

Raw data version is determined by x , and $V_r = m.n$ in th Table 3. This document describes only about $x = 1$ case. Supermodule-I in 2006 data corresponds to $x = 0$ and all other definitions (including $h[0]$ and $h[1]$ are completely different and obsolete.

Table 7 shows the current definitions of the raw data version number for only $x = 0$ case.

Except for exception, the odd number for Major version number should be used while development and once it is stable, it should be ported to even number.

The former definition in table 7 of the major and minor version numbers was used until end of 2007. The next table 8 shows how to create the major version numbers consequently, so that all important features of the data format and readout mode can be directly extracted from these 7 bits. The TRAP program analyzes the major version and behaves according to the following table 8.

| Major m | Minor n | Definition |
|-----------|-----------|---|
| 0 | 0 | Bogdan format with new header |
| 1 | 0 | Full-raw (FR) SM-I-2006 but with new header |
| 2 | * | Full-raw (FR) for production |
| 3 | * | Zero-suppressed (ZS) β |
| 4 | * | Zero-suppressed (ZS) for production |
| 5 | 0 | Test pattern, counter mode |
| 5 | 1 | Test pattern, 10bit pseudo-random mode |
| 5 | >1 | reserved for other test-patterns |
| 6~126 | * | Undefined |
| 127 | * | Play |

Table 7: Raw data version ($V_r = m.n$) definitions.

| Bit in Major Ver. | Definition |
|-------------------|--|
| 6 | Test pattern (TP) |
| 5 | Zero suppression (ZS) |
| 4 | Disable tracklets (DT) |
| 3 | Reserved |
| 2..0 | Options (OP), different meaning |
| 2..1 | ZS: when 01/10/11 - send every 128/256/1024 event completely |
| 0 | ZS: suppress MCM header/empty ADC mask |
| 0 | no ZS, no TP: calculate ADC statistics |
| 2..0 | TP: Test Pattern type, currently implemented 0..3 |
| 2..0 | no ZS, no TP, no DT: use only 1, 2, 6, 7 |

Table 8: Raw data version bit fields

Note that *Test pattern* and *Zero suppression* are not compatible, so do not set both simultaneously.

In order to be compatible to the previous definitions of the versions, the major versions 0, 3..5 should be avoided in the future runs. This is not a problem, as the option field has up to now no special usage in case of no ZS, no TP, no DT, except for the ADC statistics. The data reader software can in this case interpret the major version according to the old table 7. The *ADC statistics* option tells to the TRAP CPUs to accumulate for each ADC channel the sum of the ADC values and the sum of the squares of the ADC values. The sums are stored in the TRAP DMEM and can be read through the SCSN interface. This means, the ADC statistics doesn't change the data format. One might think about, to send the actual sums instead of the ADC values through the readout tree. To be discussed.

Test patterns

Common for all test patterns: each CPU, exactly like in normal mode without ZS, sends the same number of 32 bit data words. For example if the number of samples is 30, each ADC channel is packed in 10 words. CPU 0,1,2 have 5 ADC channels to send, CPU3 has 6 channels to send. This means, CPU0..2 send 50 words, CPU3 sends 60 words. Note that this is the maximum, as the FIFO size of each CPU is 64 words long.

Except for patterns 1 and 6, the two LSBits in the raw data do not fulfill the rule in equation 11.

7.1 Test pattern 0 Format

Not recommended for future runs, as the other counter patterns (2 and 3) are better.

7.2 Test pattern 1 Format

In this case the ADC data are replaced by a 10 bit pseudo-random counter. The start 10 bit value for each CPU is given by:

$data[0] = (1 \ll 9) \text{ or } (rob \ll 6) \text{ or } (mcm \ll 2) \text{ or } cpu$
 $data[0] = 1r : rmm : mmcc,$

The next value is given by:

$data[i+1] = ((data[i] \ll 1) \text{ or } ((data[i] \gg 9) \text{ xor } (data[i] \gg 6)) \text{ and } 1) \text{ and } 0x3FF$

In the equations, \ll / \gg is logical bit-shift left/right, *or*, *xor* and *and* are bitwise operations.

Note that the start value is always different from 0. This ensures, that all generated pseudo-random values are different from 0.

7.3 Test pattern 2 Format

$$p[2] = eeee : eess : sssp : ppcc : crrr : mmmm : uutt : tttt, \quad (6)$$

| Name | n-bits | bit pos | Value | Content |
|----------|--------|---------|-------|---------------------------------------|
| <i>e</i> | 6 | 31..26 | 0:63 | The lower 6 bits of the event counter |
| <i>s</i> | 5 | 25..21 | 1:18 | Super-module Sector Number + 1 |
| <i>p</i> | 3 | 20..18 | 1:6 | Plane Number + 1 |
| <i>c</i> | 3 | 17..15 | 1:5 | Chamber Number + 1 |
| <i>r</i> | 3 | 14..12 | 0:7 | ROB |
| <i>m</i> | 4 | 11..8 | 0:15 | MCM |
| <i>u</i> | 2 | 7..6 | 0:3 | CPU |
| <i>t</i> | 6 | 5..0 | 1:61 | counter++ |

Table 9: $p[2]$ pattern 2

Each CPU starts from counter=1 and increments the counter for each next word. The rest of the bits are constant, except for the upper bits, which are part of the event counter. This pattern is may be the best to search for breaks in the data structure. Note that the sector, plane and chamber are incremented by 1 and the counter starts from 1, in order to avoid sending of end marker.

7.4 Test pattern 3 Format

$$p[3] = eeee : eeee : eeee : ssss : sPPP : cccr : rrrm : mmuu, \quad (7)$$

| Name | n-bits | bit pos | Value | Content |
|----------|--------|---------|--------|--|
| <i>e</i> | 12 | 31..20 | 0:4095 | The lower 12 bits of the event counter |
| <i>s</i> | 5 | 19..15 | 1:18 | Super-module Sector Number + 1 |
| <i>p</i> | 3 | 14..12 | 1:6 | Plane Number + 1 |
| <i>c</i> | 3 | 11..9 | 1:5 | Chamber Number + 1 |
| <i>r</i> | 3 | 8..6 | 0:7 | ROB |
| <i>m</i> | 4 | 5..2 | 0:15 | MCM |
| <i>u</i> | 2 | 1..0 | 0:3 | CPU |

Table 10: $p[3]$ pattern 3

This pattern is very similar to 2, but the field for the event counter is extended to 12 bits, the counter is not send, but used internally to count the words send by each CPU. This pattern is may be the best to search for larger breaks in the data structure. Note that the sector, plane and chamber are incremented by 1, in order to avoid sending of end marker.

7.5 Test pattern 6 Format

$$p[6] = eeee : ssss : sPPP : cccr : rrrm : mmuu : tttt : ttff, \quad (8)$$

| Name | n-bits | bit pos | Value | Content |
|----------|--------|---------|-------|---|
| <i>e</i> | 4 | 31..28 | 0:15 | The lower 4 bits of the event counter |
| <i>s</i> | 5 | 27..23 | 1:18 | Super-module Sector Number + 1 |
| <i>p</i> | 3 | 22..20 | 1:6 | Plane Number + 1 |
| <i>c</i> | 3 | 19..17 | 1:5 | Chamber Number + 1 |
| <i>r</i> | 3 | 16..14 | 0:7 | ROB |
| <i>m</i> | 4 | 13..10 | 0:15 | MCM |
| <i>u</i> | 2 | 9..8 | 0:3 | CPU |
| <i>t</i> | 6 | 7..2 | 1:61 | counter++ |
| <i>f</i> | 2 | 1..0 | 2:3 | 11 for even and 10 for odd adc channels |

Table 11: $p[6]$ pattern 6

Each CPU starts from counter=1 and increments the counter for each next word. The rest of the bits are constant, except for the upper bits, which are part of the event counter. This pattern is very similar to pattern 2, with the only difference that the event counter has 4 bits and the last two (LS) bits fulfill the condition (11). Note that the sector, plane and chamber are incremented by 1 and the counter starts from 1.

7.6 Test pattern 7 Format

This is actually reserved for sending the compressed configuration over the readout tree. The format is outside the scope of this document.

8 TRAP data without Zero Suppression - FR SM-I-2006 ($V_r = 1.0$)

Table 12 shows the TrapData for without zero suppression (FullRaw:FR) for SM-I-2006 corresponding to $V_r = 1.0$.

| Size | Value | Content |
|-----------|-------|-------------------------|
| 1 | n.f. | McmHeader |
| N_{adc} | n.f. | AdcData, ADC channel 0 |
| . | ... | ... |
| N_{adc} | n.f. | AdcData, ADC channel 20 |

Table 12: TRAP data format.

McmHeader has the following bit pattern:

$$\text{MCMHeader} = xrrr : mmmm : eeee : eeee : eeee : eeee : 1100, \quad (9)$$

where definitions of r , m , and e are defined in Table 13.

Readout position and MCM position are given to TRAP chips from DCS software. Event counter starts from 1 (it was 10 till $V_r = 1.xx$). In the TRAP the event counter is 32 bit, here are only the lower 20 bits.

After McmHeader, the 21 channel ADC data are sent without any suppression. Since ADC has 10 bits resolution, 3 ADC data can be packed into one 32 bits word. The number of data words (N_{adc}) for one ADC channel depends on number of time bin (N_t) and written as:

$$N_{adc} = \left(\frac{N_t - 1}{3} + 1 \right), \quad (10)$$

| Name | n-bits | bit pos | Value | Content |
|------|--------|---------|------------------|--|
| x | 1 | 31 | 0 : 1 | 0 before, 1 since 10.2007 |
| r | 3 | 30..28 | 0 : 7 | Readout board position (ALICE numbering) |
| m | 4 | 27..24 | 0 : 15 | MCM position (ALICE numbering) |
| e | 20 | 23..4 | 0 : $2^{20} - 1$ | Event counter from 1 |

Table 13: MCM header contents.

where calculation must follow the normal integer calculation rule where the number after the decimal point should be simply truncated. Since N_{adc} data words can contain $3N_{adc}$ time-bin data, there can be $3 * N_{adc} - N_t$ time-bin space left if N_t is not multiple of 3. In such case the space left will be filled by zero.

ADC data is ordered from lower channel (channel 0) to higher channel (channel 20). Time-bin data is also ordered from lower time-bin (time-bin 0) to upper time bin (time-bin $N_t - 1$). One word w containing time-bin data has the following format.

$$w = xxxx : xxxx : xxyy : yyyy : yyyy : zzzz : zzzz : zzff , \quad (11)$$

where x , y , and z corresponds to time-bin $3i + 2$, $3i + 1$, and $3i$, respectively where i is an integer number. ff should be 11 for even ADC channels and 10 for odd ADC channels.

9 TRAP data with Zero Suppression - ZS β ($V_r = 3.0$)

9.1 TRAP data with Zero Suppression (ZS) and up to 30 samples

Table 14 shows the TrapData with zero suppression (ZS) in $V_r = 3.0$ definition.

| Size | Value | Content |
|-----------|-------|--|
| 1 | n.f. | McmHeader |
| 1 | n.f. | AdcMask |
| N_{adc} | n.f. | AdcData, lowest channel with non-zero bit in the mask |
| ... | ... | ... |
| N_{adc} | n.f. | AdcData, highest channel with non-zero bit in the mask |

Table 14: TRAP data format with zero suppression.

Here, McmHeader definition is same as for without Zero Suppression. AdcMask m is the map of the active ADC channels with the format of:

$$AdcMask = nncc : cccm : mmmm : mmmm : mmmm : mmmm : mmmm : 1100 , \quad (12)$$

| Name | n-bits | bit pos | Value | Content |
|------|--------|---------|------------------|---------------|
| n | 2 | 31..30 | 11b | Unused |
| c | 5 | 29..25 | 11111b | Unused |
| m | 21 | 24..4 | 0 : $2^{21} - 1$ | selected ADCs |

Table 15: ADC Mask contents.

1 as bit value in the mask means the ADC data is sent and 0 means not sent. The MSB of the mask corresponds to ADC channel 20 and the LSB corresponds to ADC channel 0. From the mask, the data reading software is able to get how many ADC data will follow. The unused bits are specified additionally to get some redundancy. 1) The 5 bits in c are the number of 1 in the ADC mask. In order to preserve the same empty ADC mask, the bits in c are inverted. 2) $nn=01$ in order to distinguish between MCM header and ADC mask, as they have the same 4 LSBits.

In this format of raw data, the TRAP is expected to apply the following algorithm to find active ADC.

1. Mark ADCs if there is at least one sample marked in the EBIn (event buffer indicator, see the TRAP user manual). In this case the configuration registers of the built in zero suppression EBIS, EBIT, EBIL, EBIN and the TRAP CPU program influence the selection.
2. Mark ADCs if there is a tracklet associated to this ADC or neighboring ADC. The tracklets that can be processed in the TRAP belong either to one channel (ch) or two ADC channels (ch, ch+1). In the first case the ADC channels ch-1, ch, ch+1 will be selected. In the second case the ADC channels ch-1, ch, ch+1 and ch+2 will be selected.

In the current format of the zero suppressed data, the full time bin data will be sent for all selected ADCs in the ADC mask. The ADC data format is exactly the same as in the case without zero suppression.

9.2 TRAP data with Zero Suppression (ZS) and more than 32 samples

The described data format for raw data readout doesn't allow to send more than 30 samples/ADC channel, see 6.2. Using a slightly different data format allows working with number of samples N_t up to 63. The TRAP CPUs search for the first non-zero bit in the event buffer indicator starting from 0 and select the starting time bin t_0 two samples before it. Of course t_0 can not be negative, it will be clipped to 0 if necessary. If t_0 is larger than the $N_t + 1 - 3 * N_{adc}$ it will be clipped there, so that the complete window lies within the correct range $0..N_t - 1$. This procedure is done for each channel independently, so t_0 can be different.

The number of the 32-bit words N_{adc} used to send one ADC channel is fixed and is part of the configuration. All ADC data from timebin t_0 to timebin $t_0 + 3 * N_{adc} - 2$ are sent 3 samples in a 32-bit word like in the other cases. The first 10-bit field here is used not to send ADC data, but to code the offset t_0 and N_{adc} .

$$w = xxx : xxx : xxyy : yyyy : yyyy : ssss : oooo : ooff , \quad (13)$$

where o represent the offset t_0 , s is the size of the ADC packet N_{adc} for one channel in 32-bit words, y is the ADC data for timebin t_0 , x is the ADC data for timebin $t_0 + 1$. The last two bits f are like in the normal ZS mode. The next $N_{adc} - 1$ 32-bit words contain the rest data for the same channel, the last word looks like:

$$w = xxx : xxx : xxyy : yyyy : yyyy : zzzz : zzzz : zzff , \quad (14)$$

where z is ADC data for timebin $t_0 + 3 * N_{adc} - 4$, y is ADC data for timebin $t_0 + 3 * N_{adc} - 3$ and x is ADC data for the last timebin transmitted $t_0 + 3 * N_{adc} - 2$.

10 Tracklet Data

Here the Tracklet word definition is described. The tracklet words are again 32bit words, where each describes one tracklet found in an MCM:

$$Tracklet = pppp : pppp : zzzz : dddd : dddy : yyyy : yyyy : yyyy , \quad (15)$$

| Name | n-bits | Value | Granularity | Content |
|------|--------|--------------------|-------------|----------------------|
| p | 8 | 0 : 1 | 0.39 % | Electron Probability |
| z | 4 | 0 : 15 | 1 | Pad Row |
| d | 7 | -8.89 : 8.89mm | 140 μ m | Deflection length |
| y | 13 | -655.28 : 655.28mm | 160 μ m | Pad Position |

Table 16: Tracklet Word contents.

The values are obtained by a fit inside the TRAP chips. With the four given coordinates p , z , d and y the position of the tracklet on a half chamber and its inclination in phi direction (across pad columns along the wire direction) are defined.

References

- [1] A TRAP Manual; <http://www.kip.uni-heidelberg.de/ti/TRD/doc/>
- [2] M. Gutfleisch, "Local Signal Processing of the ALICE Transition Radiation Detector at LHC (CERN), Marcus Gutfleisch" (PhD Thesis, 2006, KIP, Heidelberg)