Format of the read-out data of the DIF

Version **0.8**

The DIF developers

1 format of the read-out data of the DIF

1.1 Topology

A DIF is connected to **Nc** “ROC chains” each including **Nr** read-out chips (ROC). The data from the chips are sent in series. The TransmitON edges allow to separate the data from two different chips.

1.2 Amount of data

The maximum packet size sent to LDA is set to **DIF_MaxROPacket_Size** = 1 kB (512 x 16bits words).

For a chip version 2, the amount of raw data is 1 + **Nevt** + [1,2]***Nch*****Nevt** words.

Chip	**Nevt**	**Nch**	**Size** (words)
SKIROC 2	16	64	1 to 2065
SPIROC 2	16	36	
HARDROC 2	128	64	10

With full zero suppression the size would be a factor 5 to 10 less.

Case for ECAL	**Nchip/DIF**	**Amount of DIF**	**Amount of packets** (1 packet / chip / train)	**Amount of packets** (concatenated data, expectations)
EUDET	4 (112 long slab)	30	~150	~150
MODULE 0	128	150	~20000	~150 (one tower always hit)

ILD	128	7800	1 000 000	50 000 (0.04 occupancy), 10 000 (with full zero suppression)

This table does not take the noise into account.

1.3 Data buffering

For each ROC chain, a FIFO in the DIF buffers the data coming from the chain: the data from all the ROC chips on any given chain is seen by its associated DIF buffer as a single flow of words.

For each of its ROC chains, the DIF slices that flow into so-called “DIF packets” that are sent to the upper level of the DAQ system (eg. LDA, DCC). Each of these DIF packets can be up to **DIF_MaxROPacket_Size** bytes in size. In particular, they can be shorter: for example, if the read-out of chips on the chain is finished before **DIF_MaxROPacket_Size** is reached, a smaller packet is sent.

The flow of ROC read-out data will be sliced regardless of the location where the packet split occurs: the split can happen between 2 words describing an event stored in a ROC for example.

The transmitON signal from the ROC chips allows to identify the correct data and to filter the Z state on the data bus as well as the switching from a chip to the next. This allows the DIF to recognize the basic structure of the flow of ROC read-out data on a chain, so that the DIF packets can have the format described hereafter.

A DIF packet is associated with a ROC chain, reported in the DIF packet header. Since a DIF manages several ROC chains, it generates several streams of DIF packets, which may be interleaved on the wire.

1.4 Packet content

1.4.1 Header

A read-out data signature (**packettype** field).

A **localDifID** identifying the DIF on the LDA link (there might be more than 1 DIF per LDA link if a concentrator card is used), coded on 6 bits.

A **ROChainID** encoded on 3 bits.

A **ROSequenceID** which counts the number of read-out sequences and uniquely identifies a batch of packets from the same acquisition period, on 12b. This ID periodically saturates and restarts from 0.

A **ROPacketID** which identifies a packet for a given chain, on 10 bits. The ROPacketID is reset to 0 each time a read-out sequence is started (ie. ROSequenceID incremented). This counter should not saturate for any given read-out sequence.To be verified: 10b enough ?

New: A **ROLastPacket** flag (1 bit) set when this DIF packet is the last of the sequence for this chain and this acquisition period.

We thought about 2 options for the format of the DIF packets.

Option 1 (standard encapsulation):

Field	**SubField**	**Comments**
PACKETTYE (16b)		DIFBT_PKT_DATA = 0x"0001"
PACKETID (16b)		Arbitrary ID (automatic increment)
TYPEMODIFIER (16b)		0x0001
DATALENGTH (16b)		Number of 16b words in the DATA field
DATA	localDIFID (6b)+ ROpacketID (10b)	16b
	ROLastPacket(1b) + ROChainID (3b) + ROSequenceID (12b)	16b
	ROCData 1 to 505 x 16b words	Data block from ROC, w or w/o empty chips (a slice of **DIF_MaxROPacket_Size** – 7 words max)
CRC	16b	Inserted by 8b/10b itf

Option 2 (specific, a little smaller by ~0.5% on full packets):

Field	**Width**	**Comments**
PACKETTYE	16 bits	DIFBT_PKT_DATA = 0x"0001"
localDIFID (6b)+ ROpacketID (10b)	16 bits	
ROLastPacket(1b) + ROChainID (3b)+ ROSequenceID (12b)	16 bits	
DATALENGTH	16 bits	Number of 16b words in the ROCData section
ROCData	1 to 507 x 16 bit words	Data block from ROC, w or w/o empty chips (a slice of **DIF_MaxROPacket_Size** - 5 words max)
CRC	16 bits	

1.4.2 Data

The ROCData field carries a slice of the read-out data flow coming from a ROC chain. By design, in each DIF packet, the maximum number of words dedicated to the ROCData field is Choose option: 505 for option 1, and 507 for option 2.

The format of the whole sequence of slices, once reassembled, is made up of the following 2 parts:

- A sequence of **ROC read-out blocks**
- A **ROSummary** (16-bit word) which marks the end of this read-out sequence
- A series of optional **ROOptions** words, whose format is application-dependent

The ROSummary word is **always sent**, even when the sequence of ROC read-out blocks is empty (eg. in case of zero-suppression [if applicable], or early stop-readout command).

1.4.2.1 ROC read-out blocks

Each ROC read-out block in the sequence describes a variable number of events (Nevent), as recorded by the ROC. Its format is derived from, but not identical to, a raw read-out data sent by a ROC:

Field	Comments
ChipData Raw read-out data for the Chip Contains ENERGY + TDC [if applicable] + BXID values Format and size depends on the chip type size and Nevent, but must be multiple of 16bits	Derived from the read-out data block coming from the chip
”11” + ChipType(2b) + ChipAcqMode(2b) + ChipID (10b)	
”10” + DataSize(14b) To be verified:14b enough ?	Size in 16-bit words of the ChipData field
TransmitON edges on ROC control bus	
Next chip read-out block …	

Compared to the original ROC read-out data block format coming directly from a ROC, the leading bits for the ChipID field are modified and the DataSize field is added. Data from chips is taken in the same order as output by the chips (see ROC datasheet).

The contents of ChipData field is chip-dependent, and its size is Nevent & chip dependent. This ChipData field doesn't contain any marker allowing to differentiate it from the other fields (eg. ChipID, DataSize), and the value for Nevent is not explicitly reported. The online DAQ software processing the ROCData sequence has to identify the various ROC read-out blocks by recognizing the possible markers (ChipId, DataSize, ROSummary, ...) and making sure they are consistent with the value of the DataSize field.

The DataSize field is fundamental to the DAQ processing by the software: when Chipdata is not empty, it is always present and valid, even in case of an early stop-read-out command (see ROSummary section below); for consistency, the ChipType/ChipAcqMode/ChipID word is always present too when ChipData is not empty. When Chipdata is empty, either the block is discarded, or included (with these 2 words) by the DIF: both options are supported (to be confirmed).

The ROCData field in a DIF packet can include any fraction of the data described in the table above based on when the maximum size of the data block is reached.

The value of the ChipType and ChipAcqMode fields are not strictly necessary, as the DAQ online software has a configuration database which tells what kind of chips are connected to the DIFs, and what kind of read-out data they are supposed to send. But having this information in the read-out data flow allows to check a basic consistency of the whole setup.

ChipType field values

“01” - SKIROC

“10” - SPIROC

“11” - HARDROC

ChipAcqMode field values

For HardRoc chips:

“01” - Digital

“10” - Numeric

1.4.2.2 ROSummary word

The **ROSummary** field is a 16bit word appended after the sequence of ROC read-out blocks. It has the following format:

- when the last ROC in the chain has sent all of its data: “0100 0100” NumROCs(8 bits). The value of NumROCs corresponds to the number of ROCs that sent their data, even if this data was empty (ie. they only relayed the token).

- when a stop-read-out command has been received: “0100 0101” + NumROCs(8 bits). In that case, the last ChipData field may be truncated or empty. If truncated: the ChipId and DataSize fields are present and valid: in particular, DataSize tells the effective size of the ChipData field transmitted. If empty: either nothing is transmitted, or both ChipId/DataSize are transmitted and correct (DataSize = 0 in that case). The NumROCs field follows the same definition as previously.

1.4.2.3 ROOptions trailer

To be defined. It is application-specific. It should allow to identify exactly the ROSummary field when parsing the data from the end of the last packet. Either a fixed-size block (application-specific, given by the configuration database, ...). Or a block with a specified format allowing to retrieve the

ROSummary field from the end of the data stream. In any case, the DIF needs to know what to add here... so it needs to be specified offline somehow.

For example, for an HardRoc chain being tested, this trailer could be a fixed-size block, to be defined.

CRC

This pertains to each DIF packet. It is automatically added by the LDA-DIF 8b/10b interface.

2 Notes on the DAQ processing

The DAQ system receives the DIF packets and dispatches them to local memory buffers according to:

- the ODR/LDA/localDifID/ChainID combination, which identifies the source chain in the setup and the acquisition session
- the ROSequenceID field

In each buffer, the DIF packets are stored in the correct order (ROPacketID). The last packet of the stream is identified because its ROLastPacket flag is set. Any missing packet can be easily identified, either with a timeout system, or if some packet reception order properties can be expected along the DAQ chain (when that order is broken according to ROSequenceID/ROPacketID for a given DIF chain, then a packet is missing). To be confirmed

Once the DAQ knows all the packets for a given DIF chain of a given acquisition, then it starts by locating the ROSummary word in the last packet. This location depends on the ROOptions field, which is application-dependent and known by the DAQ system at runtime. It can then infer the size of the previous ROC data block via the DataSize field preceding ROSummary, which allows to know 1/ the contents of the ROC data block (discarded if the ROSummary signalled an early stop-read-out command), 2/ the location and size (via the ROSummary ahead) of the previous chip read-out data block. And so on until the beginning of the first DIF packet is reached. Once the contents of the ROC data block retrieved, they contain a variable number of event, but the format of these events is given by the ROC chip specifications and their configuration (should be consistent with the ChipType and ChipAcqMode fields).

To be added: reference to the specification of the ROC read-out data format for SPI/SKI/HardRoc.

3 To be discussed

Verify that the max size of a read-out sequence of a ROC chain is less than 1024 packets (max ROPacketID). Check that DataSize is large enough for ChipData (max 2x2^14 = 32kb).

To add: references to format of the raw SPI/SKI/ROC read-out data.

Causality enforced for all the DIF ID packets of a given chain/acquisition, at every step (DCC/LDA/ODR/driver) ???? Update the DAQ processing section accordingly.

Add in DIF command specs something to set the DIF ID programmatically ???

Tell that empty-read-out-data suppression is optional. Doesn't change the data format in any case.

Add DIF commands to set the ChipType and ChipAcqMode fields, or make sure that the chip configuration commands update these fields correctly. Or suppress these fields altogether.

Refine the ROOptions format.

Revision History

Version/Date	**By**	**Comments**
0.1 (27/02/2009)	RC	First version
0.2 (02/03/2009)	DD	With comments from DD
0.3 (03/03/2009)	RC	Comments from RC
0.4 (03/03/2009)	DD	Comments from DD
0.5 (04/03/2009)	RC	Comments from RC
0.6 (05/03/2009)	DD	Comments from DD
0.7 (05/03/2009)	VB/RC/DD	Update after brainstorming
0.8 (05/03/2009)	VB/RC/DD/FG	Brainstorming: DartaSize central. Notes on DAQ processing