

The Chimaera

LHCb RICH Readout Application Note

1. Features

When used with the Ophio optical receiver card, the Chimaera can be configured to be a quad Gbit optical receiver compatible with the data sent from the LHCb RICH L0 module.

Referring to Figure 1, which illustrates the architecture implemented in hardware and firmware, the card features:

- 4 Gbit optical data receivers;
- link deserialisers;
- automatic inhibit of disconnected channels;
- Rx clock domain to system clock domain data resynchronisers;
- input FIFOs;
- zero suppression logic;
- 4-to-1 channel multiplexer;
- 256Mbit multi-event buffer and
- readout link (USB or S-link).

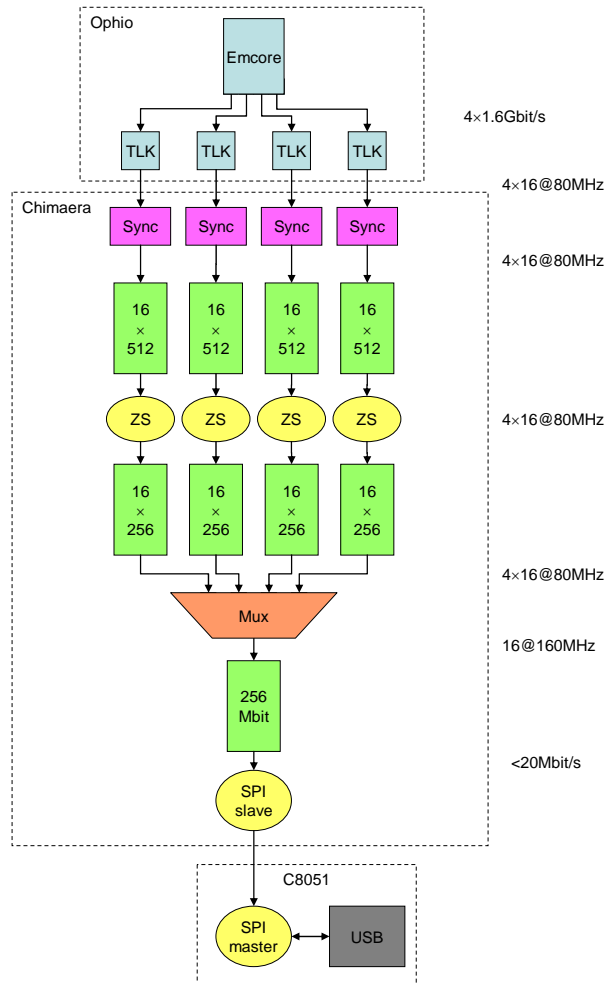


Figure 1 Chimaera LHCb RICH architecture

These features are briefly described in the following.

2. Optical data receivers

The 12-channel Emcore receiver is used of which channels 8, 9, 10, and 11 are connected.

3. Deserialisers

Each of the 4 serial channels is deserialised using the TLK2501 part. The data are output as 16-bit words at 80.157 MHz.

4. Automatic inhibit

The deserialiser provides two flags (RXDVLOS and RXERR) that can be used to detect when a link is disconnected or sends bad data. These flags are used to inhibit any unconnected or erroneous channels. Once inhibited, a channel will remain inhibited, even if the error condition goes away, until the next Chimaera reset. In normal operation it is

therefore recommended to reset at the start of each readout cycle or whenever the physical link configuration is changed.

5. Resynchronisers

Each deserialiser provides a clock, recovered from the serial data, that is synchronous with the incoming data for that channel. The recovered (or Rx) clock is synchronous with the source clock used by the transmitter. This is the 80.157 MHz clock distributed by the TTC. The Chimaera system clock is provided by an 80.44 MHz crystal oscillator.

The data from each channel are resynchronised from their corresponding Rx clock to the system clock using dual alternating input data registers in the FPGA. Dual registers are used for timing reasons: each 16 bit word is registered for 25ns instead of 12.5 which allows the internal timing constraints to be satisfied.

The LHCb data are sent as contiguous frames of 70 words (LHCb mode) or 518 words (ALICE mode) followed by (at least) two idle words. Because the system clock runs slightly faster than the Rx clock it is guaranteed that the input logic can keep up with the input data rate. The resynchroniser inserts additional idle states as necessary to keep in step. The additional idle states are removed in the input FIFOs that follow immediately the resynchronisers.

The use of the Rx clocks means that it is not required to keep the same channel-to-channel optical cable length. Variations of up to 200ns should be OK.

6. Input FIFOs

A limited number of 4 kbit RAM blocks are available in the Xilinx FPGA. They allow up to 7 LHCb events to be queued per channel in input FIFOs. If the multiplexer input FIFO is taken into account this number increases to 10 (or more if zero-suppression is used).

The input FIFOs are not large enough to buffer a complete ALICE event (but one complete ALICE event can be buffered when the multiplexer input FIFO is taken into account).

7. Zero suppression

Zero-suppression can be applied to the data before they are buffered in the multiplexer input FIFO. Zero-suppression is disabled by default but can be enabled channel-wise using configuration switches (when using the USB interface). The zero-suppression algorithm is pipelined and runs at the data input rate.

8. Multiplexer

The role of the multiplexer is to merge the four input data streams into one stream that is written to the single 16 bit wide 256 Mbit external DRAM. The multiplexer operates channel-wise in a round-robin fashion. Channels that are disconnected/inhibited are skipped by the multiplexer and do not appear in the output data.

9. Multi-event buffer

This buffer is a single external 16-bit wide 256Mbit DRAM shared by the 4 data channels. The memory input bandwidth and the number of active input channels

determines the overall system input capacity. Very roughly, the memory input speed is twice the input speed of a single channel. However, multiplexing, DRAM refresh cycles and other overheads reduce the effective bandwidth somewhat. Therefore, with one channel active, the system should be able to receive events at a sustained L0 rate. In this scenario the channel input FIFO should remain empty.

With two or more channels active the DRAM bandwidth limits the average input rate with short bursts of higher rate being accommodated by the input FIFOs and multiplexer input FIFOs. In this case the input FIFOs must be protected from overflow by external logic.

For non-zero-suppressed data the memory has a capacity of a little less than 256k LHCb frames which is shared between the active channels.

When the memory buffer overflows the write pointer wraps around and previously written events will be over-written.

Note that reading the memory causes the write pointer to be also modified. Therefore the triggering of events should be inhibited while reading the memory.

10. Readout link

Two possibilities exist: S-link or USB. The Chimaera was originally designed to be S-link compatible for use with the CERN FLIC card. The possibility of USB readout was subsequently added using an auxiliary USB adapter which couples to the Chimaera S-link connector and which is therefore mutually exclusive with the S-link usage. Different versions of the Chimaera firmware are required for the two readout options.

The advantage of the USB readout is its convenience and that it provides a bidirectional data path which can be used to send configuration information as well as to receive data. The advantage of the S-link option is that substantially higher readout bandwidth is possible.

Operational details of the USB interface are described in Section 12.

11. Data format

Non-zero-suppressed

For non-zero-suppressed data the format is simply the concatenation of the data frames from the active channels from L0 with an additional 32 bits of header added at the beginning of each frame. A single frame is shown in Table 1. The fields indicated in yellow are reserved for L1 use but are not yet filled.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L1(0)	L1 Event ID															
L1(1)	Channel			M	ZS	# non-zero 8-bit groups										
L0(0)	X	0	0	0	0	0	0	0	L0 event ID							
L0(1)	P1	0	0	0	Bunch-crossing ID											
L0(2)	FE	FIFO count				L0 board ID										H
L0(3)	P2	# non-zero 8-bit groups										G	M	C	T	FF
D(00)																
...																
D(63)																
P(0)																
P(1)																

Table 1 Non-zero-suppressed data format

The following abbreviations are used in Table 1:

- M: mode, 0=LHCb, 1=ALICE;
- ZS: zero-suppression, 0=non-zero-suppressed, 1=zero-suppressed;
- X: 0 indicates zero-suppression would result in smaller event, 1 otherwise;
- P1: Parity of L0(0) bits [0..7] and L0(1) bits [0..11];
- P2: parity of L0(2) bits [0..15] and L0(3) bits [0..4];
- H: HPD station ID, 0 or 1;
- FE: FIFO empty flag (excludes event being transmitted);
- FF: FIFO full flag (excludes event being transmitted);
- T: test pattern, 0=real data, 1=test pattern;
- C: calibration, 0=normal event, 1=calibration event;
- M: mode, 0=LHCb, 1=ALICE;
- G: status of GOL not used to transmit this frame, 0=not ready, 1=ready.

After the L1 and L0 header words are the 64 16-bit words of non-zero-suppressed pixel data followed by the pixel data column-wise parity.

The ALICE mode format is the same except that 512 16-bit pixel data words follow the header words.

Zero suppressed

The zero-suppression algorithm splits the 1024-pixel data field into 128 8-bit groups. Only groups having at least one pixel hit are written to the output data. Each non-zero group is encoded in a 16-bit word whose lowest 8-bits contain the 8 pixels and whose bits [8..14] contain the group number. Bits [0..10] of L1(1) contain the value (number of non-zero groups + 1).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L1(0)	L1 Event ID															
L1(1)	Channel			M	ZS	# non-zero 8-bit groups										
L0(0)	X	0	0	0	0	0	0	0	L0 event ID							
L0(1)	P1	0	0	0	Bunch-crossing ID											
L0(2)	FE	FIFO count				L0 board ID										H
L0(3)	P2	# non-zero 8-bit groups										G	M	C	T	FF
D(00)	A(0)								G(0)							
...							
D(n)	A(n)								G(n)							
P(0)																
P(1)																

Table 2 Zero-suppressed format

Note that zero-suppression of ALICE mode events is not yet supported. The algorithm works but there are not enough bits in the group address field for ALICE data leading to the possibility of ambiguous addresses.

12. USB interface

The Chimaera can be used in combination with a USB adapter to provide a convenient interface to a PC where the higher bandwidth provided by the S-link is not required. The USB interface couples to the Chimaera using socket PL1 so USB and S-link usage are mutually exclusive. Furthermore, the Chimaera must be loaded with the appropriate firmware. **Using incorrect firmware may result in signal contention and consequential damage.** The USB interface allows a more flexible bidirectional interface to the Chimaera than is possible with the simplex S-link.

The USB adapter is based on the Silicon Laboratories C8051F32x MCU. Communication between Chimaera, the USB MCU, and the host PC uses a simple message-passing protocol based on the USBXpress driver and device and host API. The higher-level protocol and implemented functions are summarised in the following sections.

The Chimaera USB messages

The message types are enumerated in the header file `chimaera.h`.

GenerateLHCbEvent

Forwarded to Chimaera by MCU. When in test mode this message causes the Chimaera to generate an LHCb-style event. Ignored in normal mode.

GenerateALICEEvent

Forwarded to Chimaera by MCU. When in test mode this message causes the Chimaera to generate an ALICE-style event. Ignored in normal mode.

StatusRequest

Forwarded to Chimaera by MCU. Requests Chimaera to return a status information message.

FlushRequest

Forwarded to Chimaera by MCU. Requests Chimaera to initialise the sending of buffered data. No data are actually sent until requested with PacketRequest messages.

PacketRequest

Forwarded to Chimaera by MCU. Requests Chimaera to send the next data packet(s). Data are sent in 64 byte packets.

PIOWriteRequest

Sent to MCU. Requests Chimaera to update the user port IO outputs. The port IO inputs can be read by issuing a StatusRequest message.

ConfigurationData

Forwarded to Chimaera by MCU. Contains configuration settings.

ResetRequest

Sent to USB MCU. Causes the MCU to reset the Chimaera. Note that any configuration settings in the Chimaera will be reset to default values.

Int0Enable

Sent to USB MCU. Enables the port IO interrupt function.

Int0Disable

Sent to the USB MCU. Disables the port IO interrupt function.

Data

Received from Chimaera. Contains data solicited by a PacketRequest message.

Status

Received from Chimaera. Contains status information solicited by a StatusRequest message.

Recommended readout sequence

In normal acquisition mode, the following sequence of actions is recommended:

- Send ResetRequest
- Send ConfigurationData
- Enable external triggers
- Generate triggers
- Disable external triggers
- Send StatusRequest
- Read number of buffered words from status block
- Send FlushRequest
- Extract data from buffer by repeating:
 - Send PacketRequest
 - Read data packet(s)
- Go to beginning

It is not strictly necessary to reset after extracting all the data from the buffer if exactly the right number of packets have been read. However, reset (and reconfiguration) is probably safer.