CARLOSrx: an online Data Acquisition system for ALICE ITS SDD

S. Antinori[a], F. Costa[a], D. Falchieri[a], E. Gandolfi[a], M. Masetti[a]

[a] Physics Department and INFN Sez. Bologna, Italy

filippo.costa@bo.infn.it

Abstract

CARLOSrx is part of the Silicon Drift Detector (SDD) readout chain for the ALICE experiment to be held at CERN. CARLOSrx consists of two 9U x 400 mm VME64 printed circuit boards with the main purposes of receiving data from 12 SDDs and send them towards the ALICE data acquisition system (DAQ), configure all the FEE (*Front End Electronics*) connected to SDD. The paper presents the features of the latest firmware installed on CARLOSrx, the results of tests performed at CERN with the SDD readout chain and gives a little description of software tools used during the debugging phase of the board.

I. Overview

The ITS (*Inner Tracking System*) of the ALICE experiment [1,2] is composed by six cylindrical layers. The SDDs occupy the third and fourth layer of the ITS. The detectors are installed on linear structures called ladders.

There are 260 SDDs organized in 36 ladders:

- 14 ladders for layer 3, each one contains 6 SDDs;
- 22 ladders for layer 4, each one contains 8 SDDs.

At both ends of each ladder are placed the end ladder boards, one for each detector. They host a chip called CARLOS [3] (*Compression And Run Length encOding Subsystem*) whose purpose is to perform a bi-dimensional compression of data coming from FEE (*Front End Electronics*) using double threshold cluster finding algorithm [4]. CARLOS board, after performing compression, sends data to CARLOSrx board, via 100 meters long single mode optical fibers by means of the Gigabit optical link [5].

Each end ladder boards hosts three optical fibers:

- one for sending data,
- one for receiving the 40.08 MHz LHC system clock,
- one for receiving a serial link signal with encoded configuration and trigger informations.

CARLOSrx is placed in counting room and it receives data coming from 12 CARLOS and provides the clock and configuration signals to all end ladder boards; so CARLOSrx has to deal 36 optical fibers, plus the interfaces towards the DAQ, trigger system and VME bus.

II. CARLOSrx Board Design

We were forced to break the design of CARLOSrx in two boards because in due time there were no commercial parallel optical transceivers for single mode 1310 nm.

For that reason CARLOSrx boards consists of:

- **a data processing board** called CARLOSrx (fig. 1.1), it contains 12 single optical transceivers for receiving data from 12 detectors and for sending configuration informations to 12 CARLOS, at the same time. It also handles the interface towards the trigger, DAQ and VME systems;
- **a clock distribution board** called CARLOSrx clock (fig. 1.2), it receives the clock from data processing board and distributes it through 12 optical fibers to CARLOS boards.

To receive data from all 260 SDDs modules we need 24 CARLOSrx board pairs, they will be hosted in 3 VME64 crates located in ALICE counting room.

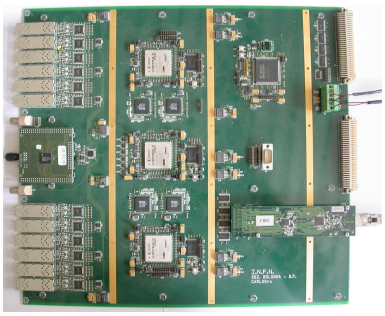

fig. 1.1 CARLOSrx data processing board

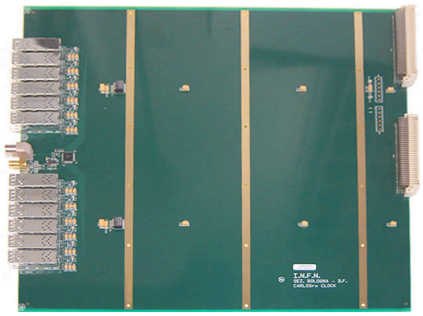

fig. 1.2 CARLOSrx clock distribution board

A. CARLOSrx: data processing board

The data processing board is a 10-layers 9U VME printed circuit board with the main task of controlling and receiving data from 12 CARLOS, then packing, buffering and after sending them towards the data acquisition system.

The board has several device interfaces:

- **12 CARLOS** by means 24 optical fibers, CARLOSrx hosts 12 x 1.25 GB/s single mode optical transceivers from Optoway [6]. Each of them is used as a 800 MBits/s link for data transmission towards CARLOSrx and as a 40 MBit/s link for serial control of the CARLOS boards.
- **TTC system** by means of the TTCrq card (66 x 66 mm) [7], this board provides the 40.08 MHz LHC system clock and the trigger signals (L0, L1 and L2) to CARLOSrx.
- **LTU** (*Local Trigger Unit*) [8] by means of the busy signal, after receiving a L0 trigger, CARLOSrx asserts the busy signal, until SDD front end electronics is ready to accept a new one.
- **ALICE DAQ** system by means of the DDL (*Detector Data Link*) boards [9]. The SIU (*Source Interface Unit*) card gets data from CARLOSrx and sends them to a PC on a PCI board (RORC), through DDL. The SIU board is also used to send configuration informations, as JTAG instructions, to CARLOSrx.
- **VME bus**, a SPARTAN II Xilinx FPGA [10] on CARLOSrx enables the communication with the VME bus for remotely programming CARLOSrx FPGAs and PROM sending JTAG informations via VME.
- **RS232 port**: it has been used for:
  - spying data while sending them towards the DDL;
  - monitoring CARLOSrx internal buffers;
  - monitoring the external FIFOs status;
  - sending command like reset or start trigger, during the debugging phase of CARLOSrx board.

B. DATA FLOW

The data flow on CARLOSrx data processing board can be described as follows:

1. data coming from CARLOS board toward one optical fiber to CARLOSrx transceiver;
2. CARLOSrx de-serializes data using commercial de-serializers from Texas Instruments [11];
3. 12 data streams, consisting of a 40 MHz 16 bits bus, are received from two Virtex PRO Xilinx XC2VP20 FPGAs [12];
4. after data packing and tagging the data stream is stored into one of 4 large IDT FIFO [13];
5. a third XC2VP20 FPGA popped data from all 4 FIFOs and transmits data to the DAQ towards SIU and DDL.

All the FPGAs implement a *round robin* queue management in order to deal with the buffers, avoiding data overflow and maximizing trigger rate.

C. CARLOSrx clock distribution board

The clock distribution board receives the 40.08 MHz LHC system clock from the data processing board by mean a bipolar LEMO cable. The clock is distributed over the board to 12 Optoway optical transceivers using a low jitter LVPECL clock distribution chip from ON Semiconductor [14].

III. Firmware design

The current version of firmware is able to receive data from all 12 transceivers at the same time, handling three levels of trigger (L0 - L1 - L2) and handling erroneous trigger sequences.

All data elaborations are done by the 3 Xilinx VIRTEX II Pro XC2VP20 FPGA installed on CARLOSrx data processing board, their job is to:

- acquire data from the input optical links,
- buffer data onto external FIFOs,
- pop data from the external FIFOs,
- send data to the DAQ system towards the DDL.

Besides dealing with the trigger and clock informations.

A. Input FPGA firmware

Each of the two lateral XC2VP20 concentrates six 16-bit data streams, coming from CARLOS boards, into two 32-bit ones towards the external FIFOs, each FPGA is connected to two of these FIFOs.

Each FPGA (fig. 3.1) contains:

- **a dual clock input FIFO**, for synchronizing de-serialized data with system clock;
- **a data packing block**, encoding 16-bit input data in a 32-bit words;
- **a 4 Kwords 32-bit FIFO**, for storing data while others channel are being written onto the external FIFO;
- **a scheduler block**, to avoid internal FIFOs overflow.

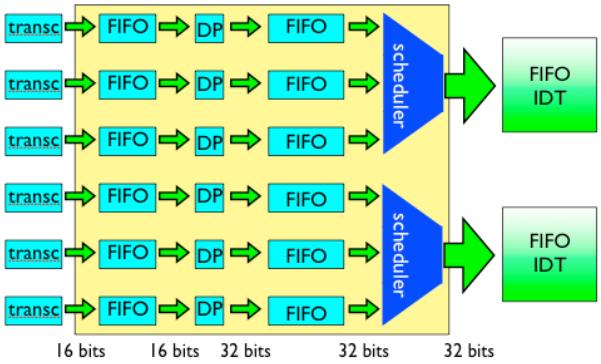

fig.3.1 XC2VP20 firmware design

B. Main FPGA firmware

The XC2VP20 in the middle has to manage:

- **Transceivers interface**, this FPGA is able to communicate with the others two XC2VP20 and to disable or enable one or more transceivers. This feature is very important to eliminate one or more broken modules from the data acquisition.
- **DAQ interface**, it pops data from the external FIFOs and send them to LDC through the DDL.
- **BUSY signal**. When some FIFOs getting full this FPGA raise a signal called back-pressure, this stops the incoming data flow. After the FIFOs are half-full, the acquisition process is started again.
- **Trigger interface**, this FPGA receives the trigger informations by TTCrq board and distribute them towards the front end boards, CARLOSrx also recognizes erroneous trigger sequences and sends to the DAQ dummy events with error bits asserted for debugging informations.
- **RS232 interface**, this FPGA is connected to a standard RS232 port. We have connected a PC to CARLOSrx using a RS232 standard cable. With a custom software we have sent informations to the board via RS232. This FPGA is able to decode the commands and reacts to them. With this feature we were able to investigate the state of the board during its normal running mode.

IV. Test Performed

Given the complexity of the entire DAQ chain we have built a dedicated setup to test the latest firmware version, before install it on all boards.

It was installed at CERN and it was composed by:

- 1 CARLOSrx
- 1 CARLOSrx clock
- 2 ladders, one from layer 3 and one from layer 4 so we were able to receive data from all 12 transceivers at the same time.

This setup was used to test the firmware in different conditions, we have done different test to verify the robustness of the firmware.

In the following tables (Table 1, 2, 3) are presented the results of the more important tests that we have done:

Table 1: first test results (runtime 13 hours)

test setup	**results**
trigger rate *95 Hz*	acquired $5x10^6$ *events*
trigger sequence *L2a*	throughput *160 MB/s*

Table 2: second test results (runtime 1 hour)

test setup	**results**
random trigger rate *40 MHz*	acquired $2x10^5$ *events*
trigger sequence *L2a*	throughput *160 MB/s*

Table 3: third test results

test setup	**results**
ERROR signal rate *0x7fffffff*	CARLOSrx generated the correct C.D.H. (table 4) (*Common Data Header*), before the data event, for all trigger sequences
BC downscaling factor *240*	
4 erroneous trigger sequences: • *L0error* • *L1err* • *L1merr* • *L2err*	

Table 4: C.D.H.

Block length [0-31]			
Format version [24-31]	*L1 Trigger message [14-23]*	*MBZ [12-13]*	*Event ID (Bunch Crossing) [0-11]*
MBZ [24-31]	*Event ID 2 (Orbit Number) [0-23]*		
Block Attributes [23-31]	*Participating Sub-Detectors [0-23]*		
MBZ [28-31]	*Status & Error Bits [12-27]*	*Mini-Event ID (Bunch Crossing) [0-11]*	
Trigger classes Low [0-31]			
ROI Low [28-31]	*MBZ [18-27]*	*Trigger Classes High[0-17]*	
ROI HIGH [0-31]			

The C.D.H. is composed by eight 32-bit words.

CARLOSrx puts these words before each event, and they contain important informations for debugging purposes.

V. Software tools

We have developed different custom software tools in order to help us during the debugging phase of the CARLOSrx boards and the DAQ chain.

A custom software was used to check the data generated by CARLOSrx, and ,in case of error, to generate LOG error files.

It has also others features like:

- generating configuration JTAG files,
- data on-line monitoring.

We have connected a PC to CARLOSrx board with a RS232 cable, with a custom software developed by us we were able to send command to the board and to receive some useful informations about its status.

All the CARLOSrx boards are installed in a VME crate, and there is no space to connect a standard JTAG programmer to change the firmware. We have developed a software that interfaces a standard PC to a VME bridge of CAEN [15] via USB cable (fig. 5.1), to transfer the JTAG informations to CARLOSrx through the VME bus. The SPARTAN FPGA installed on CARLOSrx reads this informations from the bus and decodes the data upgrading the firmware of the three XILINX II PRO FPGAs.

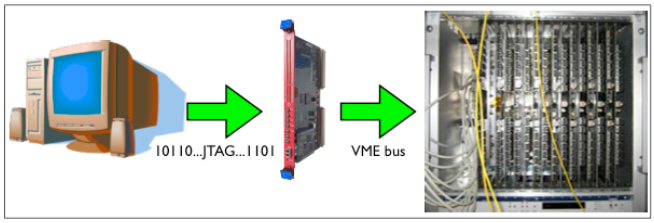

fig. 5.1 JTAG configuration using VME bus

At the moment we are using a VME bridge to upgrade the firmware, we are developing a software to use a CPU VME instead the bridge.

All the software tools were written using C and C++, TCL/TK was used to write the GUI; instead for the VME bridge we have written the software using LABVIEW.

VI. Conclusions

All CARLOSrx boards:

- 24 CARLOSrx
- 24 CARLOSrx clock

have been tested and installed at CERN.

A stable version of firmware has been tested and loaded on all FPGAs, major features of this firmware are:

- reading data from 12 transceivers at the same time
- handling erroneous trigger sequences
- latest C.D.H. format implemented
- handling VME bus

VII References

[1] A Large Ion Collider Experiment at CERN LHC, (http://alice.web.cern.ch/Alice/index.html)

[2] ALICE Collaboration, ALICE TDR 4, CERN/LHCC/99-12, 18/06/1999, (http://alice.web.cern.ch/Alice/TDR/ITS/alice_its.ps, http://alice.web.cern.ch/Alice/TDR/ITS/alice_its_3.ps)

[3] D. Falchieri, E. Gandolfi, A. Gabrielli, 8th Workshop on Electronics for LHC Experiments, Colmar, 9-13/09/2002, Electronics for Tracker, "Realization and Test of a 0.25um rad-hard chip for ALICE ITS Data Acquisition Chain", (http://lhc-electronics-workshop.web.cern.ch/LHC-electronics-workshop/2002/Tracker/A11.pdf)

[4] S. Antinori et al., "A Rad-Hard 2D-Compressor ASIC for ALICE SDD Experiment", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 524, 1-3, (2004), 295-305.

[5] A. Marchioro et al., "A Radiation Tolerant Gigabit Serializer for LHC Data Transmission", 7th Workshop on Electronics for LHC Experiments, Stockholm, Sweden, 10-14/09/2001, (http://proj-gol.web.cern.ch/proj-gol/paperLEB2001.pdf)

[6] Optoway SPS-7110, 1.25 Gbit/s SFP transceiver, datasheet [online] available: http://www.optoway.com, 2006.

[7] CERN Microelectronics Group TTCrq mezzanine, datasheet [online] available: http:// ttc.web.cern.ch/TTC.

[8] LTU Local Trigger Unit, datasheet [online] available: http://www.ep.ph.bham.ac.uk/user/pedja/alice, 2006.

[9] KFKI-RMKI (Budapest), CERN (Geneve) Detector Data Link (DDL), [online] available documentation: http://www.cern.ch/ddl.

[10] Xilinx, Inc. Xilinx Spartan II FPGA, datasheet [online] available: http://www.xilinx.com, 2006.

[11] Texas Instruments TLK1501, 0.6 to 1.5 Gbps Transceiver, datasheet [online] available: http://www.ti.com, 2006.

[12] Xilinx, Inc. Xilinx Virtex II PRO FPGA, datasheet [online] available: http://www.xilinx.com, 2006.

[13] IDT 72T36125-L4-4BBG 2.5 V TeraSync high speed FIFO, datasheet [online] available: http://www.idt.com, 2006.

[14] ON Semiconductor NB100LVEP221, 2.5V/3.3V 1:20 Differential HSTL/ECL/PECL Clock Driver, datasheet [online] available: http://www.onsemi.com, 2006.

[15] CAEN, Inc. CAEN V1718, datasheet [online] available: http://www.caen.it/nuclear/product.php?mod=V1718