

RPC Link Box Control System for RPC Detector in LHC Experiment

Wojciech M. Zabolotny^a, Ignacy M. Kudla^b, Krzysztof Pozniak^a, Krzysztof Kierzkowski^b,
Michal Pietrusinski^b, Grzegorz Wrochna^c, Jan Krolkowski^b

^aInstitute of Electronic Systems, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland

^bInstitute of Experimental Physics, Warsaw University, ul. Hoza 69, 00-681 Warszawa, Poland

^cThe Andrzej Soltan Institute for Nuclear Studies, ul. Hoza 69, 00-681 Warszawa, Poland

ABSTRACT

This paper describes the RPC Link Box Control System (RLBCS) developed for the RPC muon trigger¹ in the CMS experiment on LHC collider under construction in CERN (Geneva). RLBCS subsystem is responsible for relatively slow, bidirectional communication between the link electronics placed on detector and devices of CMS Detector Control System (DCS) located in the control room. The RLBCS is used for diagnostic and control purposes, and therefore it is essential for the RPC muon trigger. The RLBCS is also responsible for configuration of the FPGAs in the RPC link electronics, working in the harsh, irradiated environment. Additionally most part of the RLBCS itself works in the irradiated area, so assuring its reliable operation required some special solutions. All the above factors make this subsystem an important and non-trivial task in the CMS RPC muon trigger development.

Keywords: HEP, High Energy Physics, CMS, LHC, RPC, Detector Control System, FPGA

1. STRUCTURE OF THE RPC LINK SYSTEM

The simplified structure of the RPC muon trigger is shown in the Figure 1. (More detailed information can be found in [2]).

The main, unidirectional data stream contains information about the events recorded with the RPC detector. This data are received via the LVDS links from the Front End Boards (FEB) located on the RPC chambers, then compressed using the specially developed algorithm,³ and sent over the unidirectional, high-speed optical link, based on the dedicated GOL⁴ chips.

Additionally some diagnostic data are collected, which are later used to assess the state of the RPC detector, and to set the optimal operating conditions. The preprocessing of both - event data and diagnostic data is performed by the FPGA chips, located on the Link Boards (LB), which are placed in Link Boxes (containing up to 16 Link Boards). These chips are SRAM configurable, so they must be configured after the power-up. Additionally they require a periodical reconfiguration, to avoid problems caused by the radiation induced corruption of configuration data. To assure both the quick startup of the system and the flexibility of the design, the configuration data are sent to the board and stored internally (this will be discussed later). With such solution the configuration is available immediately after the power-up, and it is possible to change the configuration at the runtime.

Additionally some control data must be transmitted to the Link Board electronics, to select the proper operation mode and parameters. Finally some diagnostic and control links data must be exchanged with the Front End Boards (FEB) via the I2C links.

It is clearly visible that an additional bidirectional communication channel is necessary to transfer the diagnostic and control data to the RPC Link System – this is one of the functionalities of the RPC Link Box Control System.

The main tasks of the RLBCS may be summarized as follows:

Further author information: (Send correspondence to Wojciech Zabolotny)
Wojciech Zabolotny: E-mail: wzab@ise.pw.edu.pl, Telephone: +48 22 6607717, Fax: +48 22 8252300

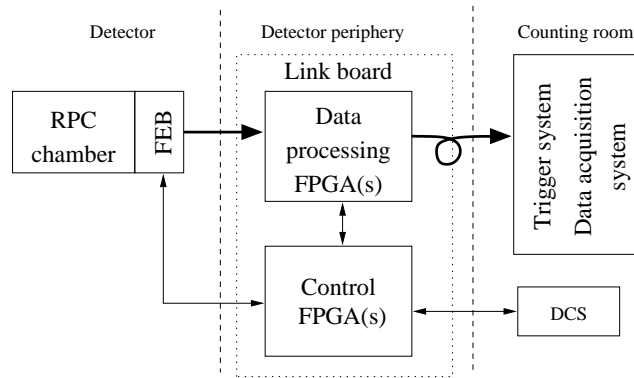


Figure 1. Simplified structure of the RPC Link system (only single RPC chamber, and single Link Board shown).

- Providing the bidirectional communication channel for diagnostic and control data
- Configuration and reconfiguration of the FPGA chips located on the Link Boards
- Storing of the configuration data, with the possibility to change that data at the runtime.

The next sections will describe these functionalities and their implementation in more detailed way.

2. BIDIRECTIONAL COMMUNICATION PROVIDED BY THE RLBCS

The bandwidth required by the RLBCS is not very high (the rough estimation gives ca. 100B/s per Link Board), however the higher bandwidth is highly desirable, because it allows to spend less time for refreshing/changing the stored FPGA's configuration data. Therefore it is reasonable to use the fastest solution available at reasonable costs. Additional requirements result from the fact, that most Link Boxes are supposed to work in the irradiated environment – so the radiation hard solution should be used to implement such essential functionalities as the communication.

Taking into consideration that the well tested and widely used solution should be cheaper for such “low volume” design (there will be less than 1500 Link Boxes), finally the CCU25 communication chip, developed in CERN for Tracker system,⁵ was chosen to establish the bidirectional link. The CCU25⁶ chip offers the following functionalities:

- Radiation hard design
- Possibility to work in the redundant architecture, where the faulty CCU25 may be skipped
- Serial link with the overall 40 Mb/s throughput
- Asynchronous 8-bit data bus with 1MB/s throughput (up to 4MB/s in the block mode)
- 16 I2C channels
- 4 8-bit parallel I/O ports
- Many other features not utilized in our design

The full description of the CCU25 properties may be found in [6] and [5].

The price of the CCU25 however is too high to equip each Link Board with such chip. Therefore a few Link Boards in a Link Box are grouped together, and connected to a local “Control Bus” (CBus), which is an asynchronous bus with 16-bit wide address and data busses. The control bus is driven by a dedicated “Control Board” (CB), which contains the CCU25 and necessary interface logic.

2.1. CCU25 incompatibilities

The necessity to provide an additional interface logic is a result of some CCU25 incompatibilities, which is normal when reusing a chip designed for another system.

The CCU asynchronous bus is intended to drive a simple 8-bit memory - so for example the read/write strobe pulses in the fast block mode are only 50 ns long (however the access cycle is 250 ns long). In the RLBCS the CCU25 is used to drive a 16-bit asynchronous bus with complex address decoding rules. Therefore two special interfacing circuits have been designed:

- **Data bus width converter** - which combines two 8-bit CCU25 accesses into a single 16-bit CBus access
- **Block mode access converter** - which extends the read/write strobe length for the block mode accesses.
For write access, this converter stores both the address and data, and generates a longer write cycle on the CBus.
For read access, the valid data must be delivered before the end of the write/read strobe (ie. in 50 ns). The converter provides the “dummy” data during the first read cycle, but stores the address, and generates a slower read cycle on the CBus. The real data are sent during the next read cycle in the block access. So the data are delayed by one read cycle, but the timing requirements for the CBus are met.

Another CCU25 incompatibility is related to its I2C controller - it doesn't implement the multibyte I2C transfers, because it was not necessary in the original target system. Unfortunately some chips used in the RPC FEBs require the multibyte I2C transfers to access their multibyte registers. Therefore another I2C controller⁷ was used to overcome that limitation.

All the above mentioned incompatibilities could be easily removed, if the CCU25 was modified. However it has been designed and manufactured as an ASIC, so no modifications are possible, and the described interface logic is necessary.

To avoid the similar problems with the rest of the RLBCS, a decision has been taken to place all more complex functionalities in the programmable logic, to improve flexibility, and to allow future upgrades and corrections. The price of such approach, however, is that RLBCS may also be affected by the radiation caused configuration corruptions. Therefore it is necessary to implement some basic functions into the radiation hard, one time programmable chip, which is able to reconfigure other FPGAs in the RLBCS.

3. CONFIGURATION OF THE FPGA CHIPS

The design is based on the Xilinx Spartan 2E chips. To assure fast configuration of the FPGA chips, the Select Map (passive parallel) mode has been used. This mode allows very fast configuration, eg. the XC2S300E chip, which uses 1,875,648 bits of configuration data, may be configured (with 40 MHz clock) in ca. 6 ms. The speed of the configuration however is limited by the access to the configuration data. The configuration data may be read from the on-board FLASH memory (which results in configuration time of ca. 15 ms), or may be transferred via the communication link (which results in configuration time of ca. 0.5 s).

4. PROTECTION OF THE FPGA CHIPS AGAINST THE RADIATION

To allow the reliable work of the FPGAs in the irradiated environment, it is necessary to perform periodical reconfiguration of these chips. To avoid time consuming transfer of the configuration data over the CCU25 link, it is desired that the configuration data are stored locally. The radiation tests⁸ have shown that the FLASH memories may be used for that purpose. Such solution assures the satisfactory lifetime of the configuration data and still allows their runtime modification. To make the FLASH stored configuration data less sensitive to radiation, a specially developed redundant coding, described in the next subsection, has been used.

The efficient utilization of FLASH memories requires implementation of a dedicated memory controller, which simplifies writing of the block of data to the memory, without the bandwidth consuming direct interaction with the FLASH built-in controller.

	31	5 4	0
word 0	usable data bits (0...26)		checksum (number of zeros)
word 1	usable data bits (27...53)		checksum (number of zeros)
word 2	usable data bits (54...80)		checksum (number of zeros)
word 3	parity bits (0...26)		checksum (number of zeros)

Figure 2. Redundant coding with error protection of FLASH data.

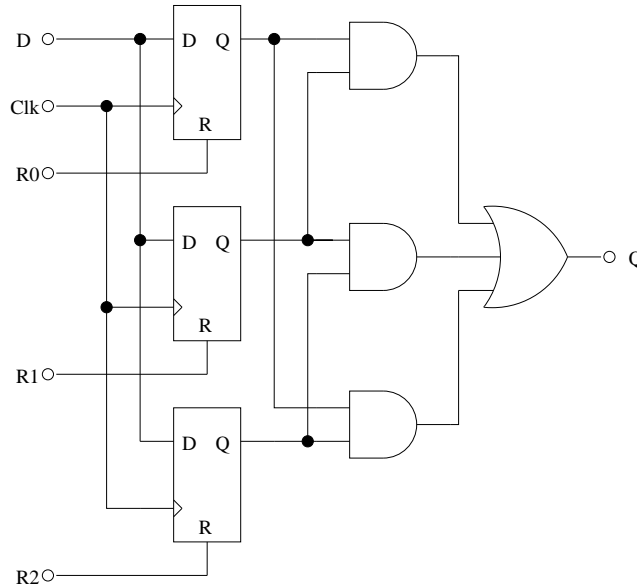


Figure 3. Triple redundant flip-flop.

4.1. Redundant coding of the configuration data

The radiation caused data corruption in the FLASH memory is almost always a change of programmed “0” value into “1”. Therefore a checksum defined as a number of zeroes in the data word offers a good reliability of detection of radiation induced errors. For 32-bit data words, it is necessary to use 5 bits in each word to store such checksum, so 27 bits can be used for the usable data. Additionally to allow error correction, each 3 data words are supplemented with the parity word (Figure 2). Finally 4 32-bit words can be used to store $3 \cdot 27 = 81$ bits of data, which can be interpreted as 5 16-bit words supplemented with single auxiliary bit. The achieved coding efficiency is equal to 5/8.

When implementing the FLASH data decoder, it has appeared, that counting of zeroes in the 27-bit word is too complex operation to be performed in single 25 ns long clock cycle. Therefore a pipelined structure has been used.

The quality of data protection may be further improved by scattering of the data words, achieved with swapping of lower address bits. (However the upper address bits should remain unchanged to allow sector erasing of FLASH data.)

4.2. Software implemented triple redundant registers

To improve the radiation hardness of the RLBCS, most of its functions has been implemented using triple redundant registers, built from the triple redundant flip-flops. The schematic diagram of such flip-flop is shown in the Figure 3.

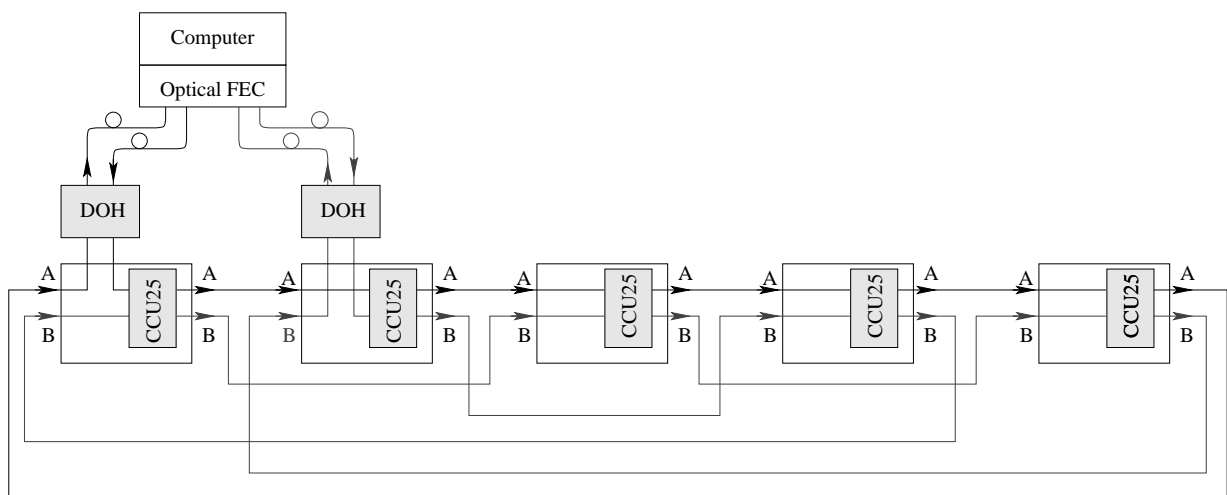


Figure 4. The redundant chain containing 5 CCU25. Channel B allows to skip faulty CCU25. (DOH - Digital Optohybrid¹⁰)

Such solution provides protection against single event upsets (SEU) in the registers, however it does not provide protection against radiation generated glitches in the combinatorial logic. The better protection could be achieved using the fully triple redundant logic,⁹ however such solution would consume too much resources.

When implementing triple redundant logic, it is important to protect the redundant flip-flops against the compiler's optimization. In our design three separate synchronous reset signals have been used for that purpose. (Using of separate clocks or separate asynchronous resets would consume too much resources).

5. FINAL DESIGN OF THE RLBCS

The final RLBCS system will use 24 redundant CCU25 chains (single redundant CCU25 chain is shown in the Figure 4), each containing up to 6 CCU25 chips.

The RLBCS located in the Link Box is shown in the Figure 5.

The Control Board (CB) contains CCU25 and two additional controllers - Control Board Initialization Controller (CBIC) and Control Board Programmable Controller (CBPC).

The additional TTCrx chip¹¹ is used to receive the experiment clock (ca. 40 MHz, which is used as an internal clock of the RLBCS) and to receive some control commands (eg. the reconfiguration request).

The CBIC is implemented in a radiation tolerant antifuse based programmable Actel chip 54SX72A, with software implemented triple redundant registers to further improve the radiation hardness. The CBIC implements only the basic functionalities required to wake up the RLBCS after the power-up or after the global reconfiguration request.

This chip is able to read the encoded configuration data from the CB's FLASH memory and to configure with them the CBPC and the Link Board Controllers (LBC) located on the Link Boards. If the configuration is successful, this chip activates the CBPC and enters the sleep mode, waiting for the next reconfiguration request. If the configuration is not successful (ie. the FLASH contents is not valid) the CBIC enters the emergency mode, in which CBPC and LBC are configured with the configuration data sent directly via CCU25. The CCU25 alarm signal is used in this case, to notify the managing computer in the DCS, that an intervention is required.

After the CBPC and LBC chips are configured and activated, the LBCs are configuring other FPGAs located on the LBs, using the configuration data stored in the LB's FLASH memories. If this configuration is not successful, the LBC sends an interrupt (which triggers another CCU25 alarm signal) and waits, until the FLASH memory content gets refreshed.

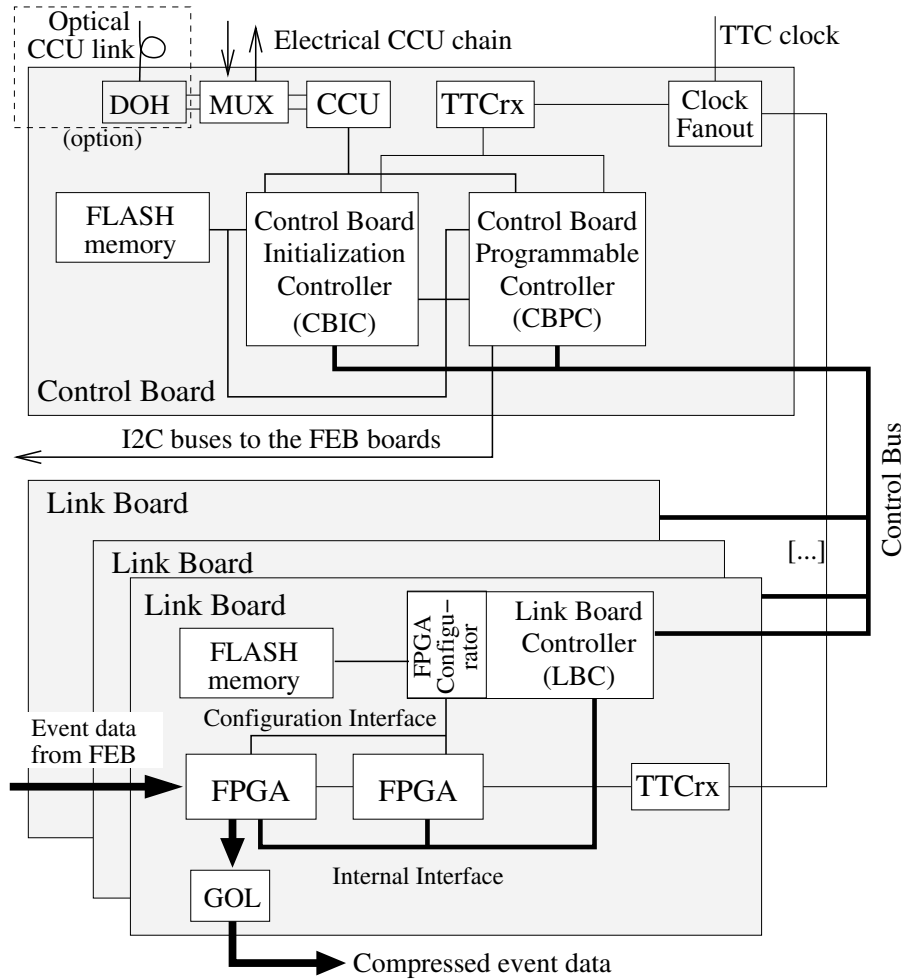


Figure 5. The structure of the Link Box part of implemented RLBCS (together with some elements of the RPC Link System).

Table 1. CCU25 access times for different block sizes (the mean value calculated for 1,000,000 accesses)

Block size in bytes	block write time [μ s]	write speed [MB/s]	block read time [μ s]	read speed [MB/s]
single mode	34.8	0.0287	39.3	0.0254
8	41.1	0.195	41.9	0.191
16	45.6	0.351	44.4	0.360
32	55.2	0.580	49.2	0.650
64	74.6	0.858	58.3	1.098

If the configuration is successful, system enters the normal operation mode, working only as a communication interface between the RPC electronics and the DCS computers. Even in this mode, however, the CBPC and LBC are checking in the background the memory contents, to detect any corruptions. If any corruption is detected, they notify the managing computer in the DCS (using the CCU25 alarm) that the memory contents needs to be refreshed (even though the data correction system still may be able to recover the original data).

It is possible (by sending a special command to the CCU25) to force the CBIC to start in the emergency mode. In this special mode it ignores the FLASH contents and waits for the configuration data sent from the CCU25. This feature may be used for sending the special, diagnostic or debugging configurations to the CBPC and to the LBC.

All the more advanced features like full FLASH controller and I2C controller are implemented in the programmable chips (ie. CBPC and LBC), so the flexibility of the design is preserved.

6. TESTING OF THE DESIGN

Testing of the design has been performed in CERN in June 2004, and in Warsaw. The tests has proven correctness of the RLBCS concept. The FLASH controller operation has been tested, and the FLASH stored data have been successfully used for reconfiguration of the FPGA chips. Also the I2C controller has been successfully used to control the connected FEB board.

The maximum throughput of the CCU25 however appeared to be ca. 1 MB/s even in the block mode (Table 1).

The more detailed analysis shows that for the 64-bytes transfers almost half of the delay is caused by the software, so probably the software optimization should significantly improve the throughput. However even the obtained transfer rate is satisfactory for operation of RLBCS.

7. CONCLUSIONS

The presented RLBCS system offers a communication link able to provide bidirectional transfer of diagnostic and control data for the RPC Link System. The RLBCS offers also an efficient system for periodical reconfiguration of the FPGA chips subjected to the radiation at the RPC detector periphery. The implemented solution combines efficient utilization of the communication link, with the short reconfiguration time, good protection of configuration data, and high flexibility. The main functions of the system has been successfully tested in a prototype.

REFERENCES

1. J. Krolikowski, G. Wrochna, M. Konecki, M. Kudla, A. Ranieri, E. Pietarinen, K. Banzuzi, K. Pozniak, and P. Zalewski, "RPC trigger," in *CMS, The Trigger and Data Acquisition project*, C. E. Board, ed., **I**, pp. 419–480, CERN, 2000. Also available as <http://cmsdoc.cern.ch/cms/TDR/TRIGGER-public/CMSTrigTDR.pdf>.
2. M.Kudla, "RPC trigger overview." http://hep.fuw.edu.pl/cms/esr/talks/MK_trigger_overview.pdf.
3. M.Gorski, I.Kudla, and K.Pozniak, "Resistive Plate Chamber (RPC) based muon trigger system for the CMS experiment - data compression/decompression system," *Nucl. Instr. and Meth. in Phys. Res. A* **419**, pp. 701–706, 1998.
4. "Gigabit optical link (GOL) transmitter – CERN microelectronics group." <http://proj-gol.web.cern.ch/proj-gol>.

5. A. Marchioro, G. Cervelli, C. Ljuslin, G. Magazzu, E. Murer, and C. Paillard, "The CMS Tracker control system." <http://cmstrackercontrol.web.cern.ch/cmstrackercontrol/documents/Training/Training%20on%20Tracker%20Control%20System%20July%202002.pdf>.
6. A. Marchioro, C. Ljuslin, and C. Paillard, "CCU25 communication and control unit ASIC for embedded slow control." <http://cmstrackercontrol.web.cern.ch/cmstrackercontrol/documents/Sandro/CCU25Specs%20v2-1.pdf>.
7. R. Herveille, "I2C - master core specification." http://www.opencores.org/cvsget.cgi/i2c/doc/i2c_specs.pdf.
8. K. Bunkowski, "Radiation tests of the RPC trigger electronic components." http://hep.fuw.edu.pl/cms/esr/talks/Radiation_tests_ESR.pdf.
9. "Functional triple modular redundancy (FTMR)." http://klabs.org/richcontent/fpga_content/DesignNotes/seu_hardening/functional_tmr_fpga_003_01-0-2.pdf.
10. "Digital optohybrid." <http://cms-tk-opto.web.cern.ch/cms-tk-opto/control/components/doh.html>.
11. J. Christiansen, A. Marchioro, P. Moreira, and T. Toifl, "TTCrx reference manual." http://ttc.web.cern.ch/TTC/TTCrx_manual3.8.pdf.