Embedding Online Test and Monitoring Features in Real Time Hardware Systems

Irakli Mandjavidze, Thierry Romanteau

***Abstract*–This paper presents several test and monitoring techniques that have been deployed in some sub-systems of the CMS electromagnetic calorimeter trigger and readout electronics. The embedded online testability features of the selective readout processor boards and of the endcap trigger concentrator cards greatly simplified functional validation of the real-time hardware, facilitated development of automated production test benches and played an important role during the commissioning phase of the calorimeter readout system. The deployed test and monitoring mechanisms as well as the used remote firmware control schemes are generic enough and look worth to be applied to a wide range of real-time systems with the FPGA-based hardware.**

I. Introduction

With the advent of FPGA devices, embedding multi-gigabit transceivers, flexible memory blocks and processor cores, each element in the trigger and data acquisition (TDAQ) systems of the modern particle physics experiments becomes more and more complex and compact. Current technologies allow integrating on an electronics board the functionalities that previously were implemented on a number of cards of different types. The count of IO channels of these highly integrated boards is steadily growing, while their interaction with the rest of the system and embedded data analysis algorithms tend to become increasingly sophisticated. In addition the developments are versatile enough for the same hardware can perform different functions with only firmware changes. It is crucial to embed in each of these complex TDAQ elements various on-line test and monitoring features that facilitate debugging of the boards during the development phase and validation of the overall system operation during the integration and exploitation phases.

In this paper we describe several test and monitoring techniques that have been deployed in the Selective Readout Processor (SRP) of the CMS Electromagnetic Calorimeter (ECAL). The palette ranges from an extensive statistics file accumulated in hardware to flexible spy memories storing event and run control data, passing through a continuous verification of exchanged data integrity and of system synchronization. The system-on-chip design of the SRP excludes any need in additional tester hardware development as an SRP board can be used as its own tester or as a tester of some parts of the ECAL readout electronics. Similar test and monitoring features have also been deployed in the ECAL endcap Trigger Concentrator Cards (TCC) where they have recommended themselves equally useful. The remote firmware management supporting multiple on-board revisions is yet another extremely handy feature of the SRP and the endcap TCC boards, which facilitate their testing and maintenance.

Manuscript received on May 27, 2010.

Irakli Mandjavidze is with IRFU, CEA Saclay, 91191, Gif-sur-Yvette, France (telephone: +33-(0)1-6908-1796, e-mail: irakli.mandjavidze@cea.fr).

Thierry Romanteau is with LLR, École polytechnique, CNRS/IN2P3, 91128, Palaiseau, France (telephone: +33-(0)1-6933-5618, e-mail: romanteau@llr.in2p3.fr).

II. The CMS ECAL Off-Detector Electronics

The Trigger Concentrator Cards and the Selective Readout Processor boards are parts of the off-detector electronics of the CMS electromagnetic calorimeter (Fig. 1). The off-detector electronics is described elsewhere [1]. Here we recall only those implementation details that are relevant to the further reading.

At each bunch crossing the TCCs finalize generation of the calorimeter trigger towers (TT), synchronize and transmit them to the regional calorimeter trigger electronics. The TCCs also perform trigger tower classification as low, intermediate and high interest depending on the energies deposited within the towers. For events accepted by the first level trigger the TCCs send the classification data to the SRP boards [2]. Complete trigger tower information is delivered to the readout electronics to include it into the event record.

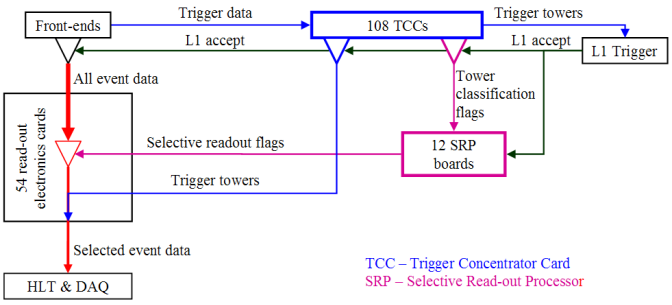

Fig. 1. TCC and SRB boards within the CMS ECAL read-out system

Based on the TT classification the SRP identifies for each event the ECAL regions with the energy deposits satisfying certain programmable criteria. It then directs the ECAL readout electronics to apply predefined zero suppression levels to the crystal data, depending on whether the crystals fall within these regions or not. Namely full precision readout is requested for a large 3x3 tower area with a high interest TT in the middle. Also full precision readout of a set of 25 crystals is performed if at least one of the crystals contributes to an intermediate interest TT. The SRP orders zero suppression with a certain threshold to be applied on the rest of the calorimeter crystals.

The TCCs are 9U VME64x compliant boards. They share VME crates with the clock and control distribution modules and with the ECAL readout electronics modules. Each TCC communicates with its corresponding SRP board over a 1.6 Gbit/s serial optical link and with its corresponding readout electronics module over a 640 Mbit/s serial electrical link. There are two different types of TCCs - barrel and endcap. The barrel and the endcap boards are substantially different. The rest of the discussion concerns only the endcap TCCs. There are 12 endcap TCCs in each of the 6 endcap VME crates, thus totaling to 72 boards for the two endcap parts of the calorimeter.

The SRP is housed in a single 6U VME crate. It is composed of 12 identical single-slot VME64x compliant boards. They communicate over 1.6 Gbit/s serial optical links with the TCCs, with the ECAL readout electronics boards and with each-others. Each SRP board establishes connections with up to 12 TCCs, with up to 6 readout cards and with up to 8 neighbor SRPs. In contrast to the TCCs the two flavors of the boards, barrel and endcap, differ only in firmware. There is a third type of firmware that transforms an SRP board into an SRP tester capable to inject data patterns into a tested SRP (or read-out electronics cards) and to verify readout instructions produced by the SRP (or TT flags generated by the TCCs under test).

III. Remote Management of Firmware

Both the SRP and the endcap TCC developments are based on the modern FPGA devices. The SRP deploys a single very large xc2vp70 FPGA device from the Xilinx Virtex2Pro family. All SRP functionalities – gigabit serial communication channels, an interface with the CMS trigger control system, the selection algorithm, the VME64x interface – are implemented within the FPGA. The FPGA is coupled with two Xilinx xcf32p Platform Flash PROM (FPROM) devices that together can keep up to four firmware revisions [3].

The endcap TCC deploys two large xc4vlx100 FPGA devices from the Xilinx Virtex4 family and a middle range xc2vp20 Virtex2Pro FPGA. The later implements the VME64x interface, the interfaces with the SRP and ECAL readout electronics boards and an interface with the CMS trigger control system. The Virtex2Pro FPGA is coupled with a single xcf32p FPROM that may keep up to four firmware revisions.

For each bunch crossing the Virtex4 devices of the TCCs calculate trigger towers out of the dedicated data from the endcap calorimeter front-ends and provide the towers to the calorimeter first level trigger. For each accepted event they send to the Virtex2Pro device the corresponding trigger towers and the trigger tower classification data to be delivered to the SRP and the ECAL readout electronics boards. The Virtex4 devices are associated with the Xilinx SystemACE controller [4] and a Compact FLASH disk that can keep up to 8 firmware revisions of both FPGAs.

The important number of the TCC and SRP modules and their location in the underground service cavern ~80 m below the surface level appeal for the remote firmware management.

A. Remote Control via JTAG

On a common agreement the ECAL readout electronics makes use of the Module Test and Maintenance (MTM) bus of the VME64x backplane [5]. According to the adopted scheme, on each VME board a ScanSTA111 Boundary Scan Bridge from National Semiconductor [6] interfaces up to three on-board local JTAG chains to the backplane MTM bus signals. The SRP example on Fig. 2 illustrates the principle.

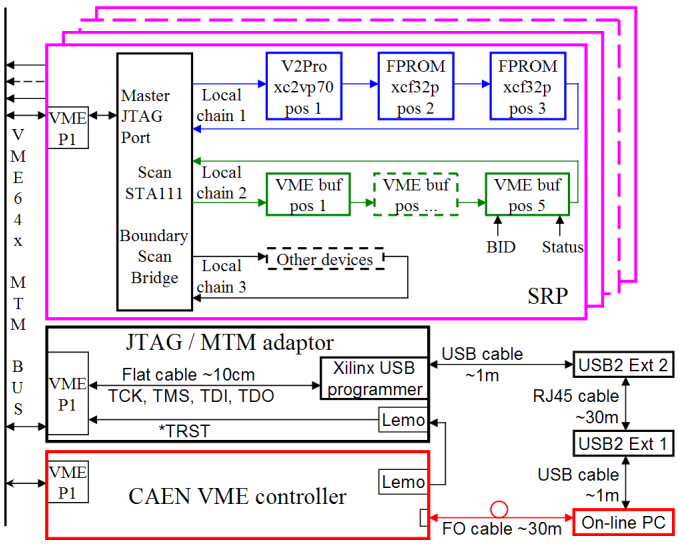

Fig. 2. JTAG controller organization for the SRP boards

The TCK, TMS, TDI, TDO, *TRST pins of the ScanSTA111 master JTAG port are connected to the corresponding pins of the VME64x MTM bus on the P1 connector. The ScanSTA111 Bridge can be instructed through JTAG to establish a transparent connection between its master JTAG port and one of the three local JTAG ports. In this way the selected local JTAG chain can be controlled by the JTAG controller. A particular bridge on the MTM bus can be selected according to the ID of the slot that the corresponding board occupies. The geographical address pins of the VME64x bus are routed towards the address pins of the ScanSTA111 Bridge. The JTAG instruction that brings the bridge into the transparent mode carries an address field. This address field is decoded by all bridges on the MTM bus and is compared to the address encoded on their address pins. Only the ScanSTA111 Bridge for which the match occurs enters the transparent mode. For reliable JTAG operations only one bridge must be kept in the transparent mode at a time. Once the JTAG operations with the selected chain are finished the corresponding bridge must be brought to its default, non-transparent state to allow operations with other bridges and chains. This is accomplished asserting the MTM Reset signal.

A passive 1U VME module called JTAG/MTM adaptor holds the Xilinx JTAG programming device - the “Platform Cable USB II” [7]. The JTAG signals of the programmer are tied to the corresponding signals of the MTM bus. A USB/RJ45 extender [8] with about 30m Ethernet cable is used to connect the Xilinx programmer’s USB port to a USB port of the Linux PC from the CMS on-line cluster, which controls the SRP crate. The PC runs the Xilinx iMPACT configuration

software suite [9] installed on a zone shared among all the PCs on the cluster. As the Xilinx programmer tool does not support the optional JTAG reset signal, it is asserted and de-asserted by the on-line PC through the CAEN PCI/VME controller [10]. The JTAG/MTM adaptor provides the required path from the controller's front panel Lemo connector towards the corresponding pin of the P1 connector (*TRST).

One local JTAG chain on the SRP board comprises the Xilinx Virtex2Pro FPGA and the two FPROMs. The FPROMs hold 4 compressed firmware revisions with one of them being set as a default. Apart of the already mentioned endcap, barrel and test firmware revisions there is also a so called "low level tests" firmware that allows quick verification of the board's basic functionalities.

Another local JTAG chain includes sn74lvth18512 level-translator buffers [11] from Texas Instruments adapting the levels of the VME signals to the levels of the FPGA IO pins. Some inputs of one of the buffers are connected to the board ID and various status signals, such as the "power OK", "firmware load done" and "temperature OK" indicators.

A custom suite of configuration bash scripts and programs have been developed. The JTAG configuration scripts produce temporary iMPACT command and various SVF files and invoke the iMPACT tool in its batch mode passing it the temporary command files. During execution the iMPACT programming tool produces log files. The command and log files are kept for further analysis in case of eventual errors.

The JTAG programming suite allows for scanning of the VME crate, for discovery of deployed boards and their states, for reading the type and the production dates of the firmware revisions loaded in the FPGAs and in the FPROMs, for programming of the FPGA and the FPROM devices with desired firmware sets. It is also possible to set a desired revision in FPROMs as a default and reload the FPGAs.

The same remote firmware management scheme has been implemented for the Viretx2Pro devices of the endcap TCCs [12]. The single FPROM device can hold up to four non-compressed revision of the FPGA. This feature turned out very useful during the debug and commissioning phases allowing fast switching between approved and test firmware versions. The JTAG configuration scripts can run simultaneously on all of the 6 endcap control PCs reducing almost by a factor of 6 the time needed to program the 72 endcap TCC boards.

It takes about 15 seconds to program a Virtex2Pro FPGA, about two minutes to program and verify endcap TCC FPROM with four revisions, about 5 minutes to program and verify the two SRP FPROMs with four revisions and ~5 seconds to choose a new default revision within the FPROMs and reload the FPGA. Thus about one hour is needed to update all 12 boards in the SRP crate with a new set of four revisions. This operation is completely automated and once the process is initiated it does not require human intervention.

It is worth to mention that the adopted JTAG firmware management scheme allows to connect the Altera ByteBlaster programming device [13] to the MTM bus of the endcap VME crates and to use the Quartus II Software suite [14] for programming the Altera CPLD devices on the TCCs Synchronization and Link daughter Boards [15].

B. Remote Control via the Xilinx SystemACE Interface

As has been already mentioned the endcap TCC deploys two xc4vlx100 Virtex4 devices that are used for trigger tower generation. The FPGAs are associated with the Xilinx SystemACE controller and a Compact FLASH memory device. The later can keep up to 8 firmware revisions of both FPGAs. The SystemACE controller and the Compact FLASH are accessible via the VME interface giving the possibility of remote firmware programming and upgrade operations for the two devices (Fig. 3.a).

The file system supported by the SystemACE Compact FLASH is FAT. Each of the 8 possible firmware revisions is kept in a separate directory. The directories are indexed from 0 to 7. The endcap TCC hardware design fixes the default firmware revision to the one placed in the directory with the index "0". The default firmware revision is loaded in the Virtex4 FPGA devices at power-up or at the SystemACE controller reset. When an endcap TCC board is up and running, the Virtex4 FPGAs can be programmed with other revisions on the Compact FLASH. The operation is initiated through VME. This is useful for test purposes.

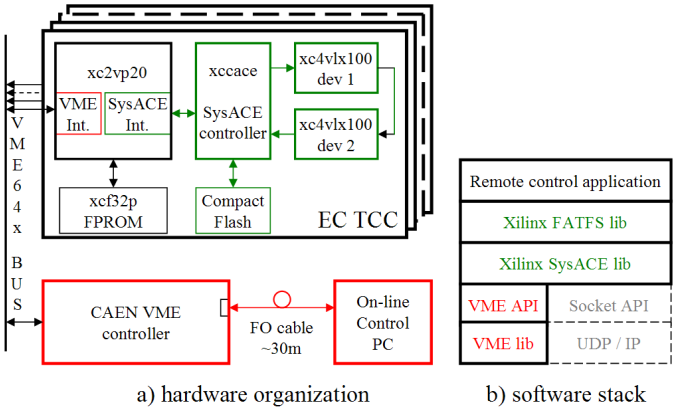

Fig. 3. Firmware management via the VME bus in the endcap TCCs

A set of standalone "C" programs form the endcap TCC remote SystemACE programming software suite. The software organization is shown on Fig. 3.b. The Xilinx SysACE and FATFS libraries have been interfaced to the CAEN VME library. The former two libraries are provided with the Embedded Development Kit (EDK) software suite [16].

The remote firmware control application allows to view the content of the Compact Flash, to copy one or several firmware revisions to desired directories, to load the virtex4 FPGAs with a selected firmware revision, to reset the SystemACE controller forcing the reload of the FPGAs with a default revision, or to perform any combination of the above actions. The application takes as a command line parameter a list of VME slot IDs so that the same actions are consequently performed on all of the endcap TCC boards from the list. This excludes the need of human intervention in the firmware upgrade process.

In average the firmware upgrade procedure takes about a minute per endcap TCC board and not more than 15 minutes per an endcap VME crate. Upgrades in the 6 endcap VME crates can be performed in parallel.

A rescue tool was developed as well to restore the damaged FAT systems of the Compact FLASH disks *in-situ* without the need of them being removed from the TCC boards (a delicate operation). The tool allows to a) perform a byte-by-byte copy of the Compact FLASH content from an endcap TCC into a file on a disk and b) to perform a byte-byte copy of a Compact FLASH image from a file to the Compact FLASH on a TCC. Using the rescue tool one can create a master copy of a healthy Compact FLASH. Next the master image can be copied to the Compact FLASH with the damaged FAT system. More details can be found in [12]. Similar approach has been reported in [17].

It is worth to mention that the software organization of the SystemACE remote management programs allowed us to adapt it to the standard socket API of the IP protocol. The application that makes use of the remote firmware control based on the socket API is described in [18].

IV. Development based on a System-on-Chip Design

Development of the SRP firmware started in parallel with its hardware design. In absence even of a prototype board the use of development kits with Xilinx FPGAs has been decided. A scalable firmware has been devised that was lately easily ported to the final hardware with minor modifications and additions. The success in the firmware development was largely ensured by the choice of a system-on-chip (SoC) paradigm for firmware design. It allowed for early development of the SRP on-line control software that was initially debugged on the PowerPC processor core embedded in the Xilinx Virtex2Pro FPGAs. Next the standalone control applications have been ported to the Linux PC/VME platform and integrated in the overall CMS on-line software. The SRP firmware organization is shown on Fig. 4.

The SoC block consists of a PPC405 PowerPC core, 256 kbyte data and instruction memory and a RS232 console core. The User IP with the SRP logic is interfaced to the processor bus (PLB) through a Xilinx standard interface core. All these cores are readily available within the Xilinx EDK software suite [16] that makes design of the SoC block fairly straightforward.

The SRP logic instantiates a programmable number of communication channels consisting from the Xilinx RocketIO multi-gigabit transceiver (MGT) cores and associated custom Framer modules. TT classification data received from TCCs and neighbor SRP boards are placed in multi-port memories. They are processed by a pipelined algorithm logic. Derived selective readout instructions are sent through the communication channels to the ECAL readout electronics. These actions are performed on an event-per-event basis. They are scheduled by the selective readout state machine (SR FSM). The trigger interface module receives trigger, timing and control information from the CMS trigger control system (TCS), interprets commands and notifies the run control state machine (RC FSM). The later governs operations of the SRP boards at the TDAQ system level.

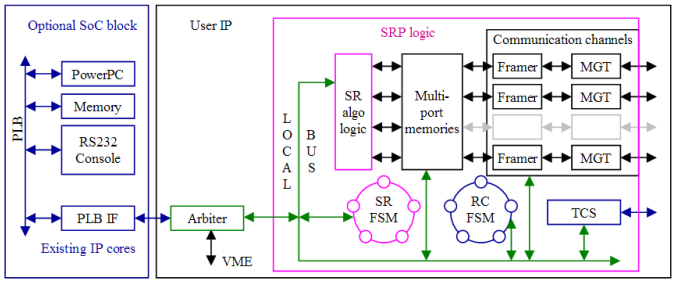

Fig. 4. SRP example of a System-on-Chip design

Access to programmable parameters, status information and accumulated statistics, contents of various spy memories is available through a custom local bus. The arbiter module grants local bus accesses to the embedded processor or to a remote run control system via the VME bus. The use of the custom fairly simple local bus makes the SRP logic independent from the complexity of the VME or the processor busses. It also makes the logic immune from evolution of the later. The local bus simplified simulations of the SRP logic.

The same standalone alphanumeric control applications can run on a Linux PC and on the embedded PowerPC processor. In the former case the control is done via the VME bus and in the later case - via the embedded processor bus. This feature was extremely helpful during the early phases of the ECAL readout electronics interoperability tests: it was always possible to control and spy the operation of SRP boards via their RS232 console even when for various reasons some of the high level on-line software services were unavailable. More details on the benefits of the SoC design paradigm can be found in [19].

Finally, the SRP firmware is organized in a way that the SoC block is optional. During the synthesis phase one can decide to instantiate the block or not within the barrel, endcap and tester firmware. Production of the firmware is fully automated using the GNU make utility.

V. Online Test and Monitoring

Incorporating various online test and monitoring features within the SRP or the endcap TCC firmware started together with the development of their mainstream functionality. When advancing in the design, every time a new functional bock was added to the firmware it was accompanied by either its specific test or monitoring logic or both. In parallel, the corresponding software part was implemented and the newly added functionality could be debugged to a high extent mostly using the on-board hardware with the minimal or no needs of a dedicated test bench. With this approach we could develop and debug the SRP firmware in a standalone mode, despite the fact that the SRP needs inputs from the TCCs and the TCS to perform the selective readout algorithm. Similarly, an important part of the endcap TCC logic could have been tested without the need of the SRP or readout electronics hardware. Afterwards, the practice has shown that functionalities with tightly tied on-line test and monitoring logic were giving no

troubles during the hardware integration and commissioning phases.

The principles of embedding of some of the online monitoring and testing features are illustrated on Fig. 5.

A. Online Monitoring

The SRP and the endcap TCC firmware incorporate three types of online monitoring logic: statistics, error detection and spy buffers.

Statistics. The statistics is composed of status registers and of various counters. Each VHDL entity includes a registry file with a specific control, configuration and status registers and a number of statistics counters proper to the operation of the entity. The values of the registers and counters can be read through the local bus. When needed the counters are large enough to avoid their overflow during the run. For rare events, such as error occurrence, usually 8-bit counters are used with the overflow indicator in MSB.

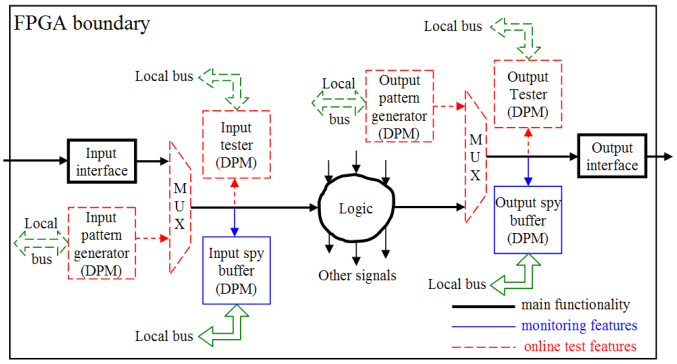

Fig. 5. Principles of embedding monitoring and online test features

There could be a single or multiple instances of an entity in the firmware. A typical example of a single instance entity is a Trigger Control System interface. Among others, its status register keeps the highest number of the first level trigger accepts queued in the de-randomization FIFO, overflow warning and overflow flags, last broadcast command received, etc. The counters of the interface indicate a total number of broadcast commands received and the numbers of each type of expected and correctly decoded commands (e.g. start of run, end of run, hardware reset, resynchronization request, etc). There are also orbit and event counters, as well as the counters related to the signals specific to the ECAL monitoring.

A typical example of a multi-instance entity is the RocketIO based communication channel module. There could be as many as 20 communication channels instantiated in the SRP firmware. Among others the status register of each instance indicates the readiness of its transmitter and the synchronization state of its receiver, eventual Tx and Rx buffer errors, etc. The counters of the instance provide the numbers of transmitted and received frames, the numbers of detected CRC, running disparity or unknown control character errors.

It is possible to take a snapshot of the board's state. For this the software activates a "Latch Statistics" signal forcing all instances to copy current values of their statistics registers to their mirror registers. Thus the values latched in the mirror registers belong to the same instance in time. The software then reads the mirror registers, while the main registers continue to accumulate the statistics. The software can clear the statistics explicitly activating a "Clear Statistics" signal. Alternatively, the statistics is cleared when a global reset of the board's hardware is requested.

As the software takes a snapshot of the board's state the values of the communication channel counters belong to the same instance. This greatly simplifies visual apprehension of periodically acquired statistics. In normal operation the number of received and sent frames of all communication channels should be equal. The same is true for other entities with multiple instances such as trigger tower receiver modules, selective readout indicator sender modules, tester sender and receiver modules, etc.

It is worth to mention that the communication channel module designed for the SRP application was reused with the minimal modifications in the endcap TCC firmware.

Error Detection. The SRPs and where applicable the endcap TCCs continuously check the integrity of received data and monitor their synchronization with the rest of the system. Every time an error is detected it is signaled to the TCS. The event data produced by the boards for erroneous events are appropriately flagged. The boards also keep relevant information concerning the faulty events to facilitate understanding of causes. The boards distinguish rare and permanent occurrences of faulty conditions. The permanent faults always result to a TCS notification.

Event data packets exchanged among the TCCs, SRPs and the readout electronics are formed from a header, a number of data words and a trailer. The header contains the board, the event and the bunch identifiers and a length word. The trailer contains a vertical even parity word calculated for the entire packet. The MSB of each word in the packet carries a horizontal odd parity bit.

The data integrity is verified for each received packet on every channel: the sender identifier is compared with the one expected on the particular channel; the event and bunch crossing identifiers are compared to the locally calculated values; the correctness of the length field is checked; the horizontal and the vertical parities recomputed. The data integrity checks may detect a de-synchronization condition within the system (e.g. received event or bunch crossing identifier does not match the locally calculated one). In case of a permanent de-synchronization the boards request the so called "re-synchronization" procedure from the TCS.

The SRPs and the data readout part of the TCCs are real-time systems that have to provide responses within a narrow time budget of the order of few microseconds. The time-out mechanisms prevent the SRPs and the endcap TCCs from waiting indefinitely for missing data. When a time-out is detected the missing data is replaced by preprogrammed substitution data and event processing continues. The derived data are marked accordingly. In case of a permanent time-out condition, as well as in some other non recoverable error conditions, the boards request the so called "hardware reset" procedure from the TCS.

Spy Buffers. The spy buffers are associated with input and output interfaces of the board, which are under the control of the FPGA firmware (Fig. 5). For SRPs the examples are the TCS interface, the TCC input links, the output links towards the ECAL readout electronics, the input (output) links from (to) neighbor SRPs. The spy buffers represent circular memories with equal number of multiword entries. For each first level trigger the data that enters or leaves a given interface are written into the corresponding spy buffer. An entry of a spy buffer is big enough to contain all data of an event. An entry of the TCS spy buffer keeps event identifier and its orbit and bunch crossing identifiers. The corresponding entry of a TCC spy buffer keeps entire data frame that has been received from the TCC for the same event. Similarly, the frame with the derived selective readout identifiers is written to the matching entry of an output spy buffer while it is sent to the readout electronics board.

The spy buffers operate in one of the four possible modes:

1. In the "one shot" mode the spy buffers are filled with only first N events. The spy buffer mechanism has to be rearmed for new acquisition to take place.
2. In the "circular" mode the spy buffers are continuously filled with data. Read and write pointers are used to determine the number of used items from the earliest to the most recent entries.
3. In the "error" mode the spy buffers are filled with the data of only those events that were flagged as erroneous.
4. In the "error history" mode the spy buffers are filled with data in a circular way. After an erroneous event is detected the spy buffers accept a certain number of consecutive events and stop. In this way the spy buffers keep track on what happened before and after the erroneous event. The faulty event itself is marked.

There is a special mechanism that allows synchronization of all spy buffers among a desirable set of SRPs and TCCs. For this the spy buffers operate in the "one shot" mode. The control software sets in the desired set of boards the orbit identifier for which the spying becomes active. The logic enables spying when the running orbit identifier equals the programmed one. This mechanism guarantees that all spy buffers in the boards contain the data of the same events.

The contents of the spy buffers can be read via the local bus. The spy buffers are 32 events deep in the SRPs and 8 events deep in the endcap TCCs. In TCCs they are used for the TCS interface, for the SRP and the readout electronics interfaces, as well as for the loopback path of these interfaces (Fig. 6; the loopback paths and their purposes are described later).

In the SRPs there is a special 512-entry deep spy buffer that keeps track of changes in the run control state machine and the signals that cause the changes. A new entry is written in the circular spy buffer when there is a change in the value of a board-wide status register. The run control spy buffer allows to trace board responses to the control software commands, to the system-wide commands received from the TCS interface, to internally generated conditions, etc.

B. *Embedded Online Test Logic*

The SRP and the endcap TCC firmware incorporate two types of online test logic: pattern generators and tester modules.

Pattern Generators. The pattern generators may be associated with input and output interfaces of the board, which are under the control of the FPGA firmware (Fig. 5). A pattern generator is a circular memory buffer with a number of multiword entries. Each entry is preprogrammed by software with desired event data. Incoming first level triggers force pattern generators to circle among the entries. For each trigger a pattern generator outputs word-by-word the event data from its active entry.

An input pattern generator can be used to exercise the board with the predefined event data and to verify its response. An output pattern generator is useful for interoperability tests within the system, or for exercising a receiver board with a desired succession of data patterns.

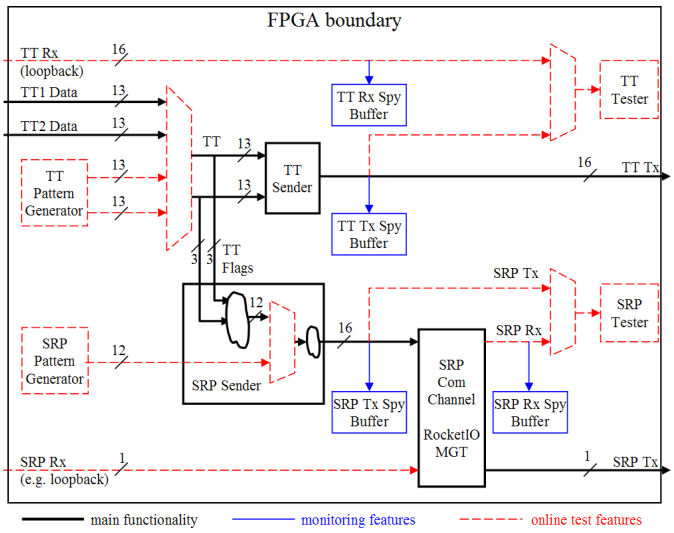

Fig. 6. Part of the endcap TCC Virtex2Pro FPGA logic

In the endcap TCC an input pattern generator is instantiated in the Virtex2Pro FPGA in order to emulate the TT data produced by the two Viretx4 FPGAs (Fig. 6). An output pattern generator is used at the SRP interface. The firmware of the two Virtex4 FPGAs also includes input pattern generators to emulate the trigger data incoming from the calorimeter frontends. This allows to validate at 40 MHz bunch crossing rate the trigger chain of the TCCs and their interaction with the downstream trigger electronics, as well as to verify the on-board inter-FPGA communications.

In the barrel and the endcap SRP firmware every output channel towards the readout electronics is complemented with an output pattern generator. The SRP tester firmware implements output pattern generators to emulate up to 12 TCC data streams and up to 8 neighbor SRP data streams.

Tester Modules. The tester modules may be associated with input and output interfaces of the board, which are under the control of the FPGA firmware (Fig. 5). A tester module is a circular memory buffer with a number of multiword entries and associated comparison logic. Each entry is

preprogrammed by software with the expected event data. Incoming first level triggers force tester modules to circle among the entries. For each trigger a tester module compares word-by-word the incoming event data with the event data from its active entry. If a mismatch is detected an error flag is raised.

An input tester module is useful for high rate interoperability tests or for debugging of a sender board when sender board produces a known data patterns. An output tester module is useful to validate the logic of the board under some known input data patterns.

In the endcap TCC the tester modules are instantiated at the SRP and the readout electronics interfaces (Fig. 6). Both of these interfaces have a loopback possibility when the outgoing data streams are injected back to the FPGA. The same tester modules can be programmed to check the incoming loopback data. This gives a possibility to validate physical layer of the interfaces (serial optical and electrical links).

The SRP tester firmware instantiates up to 20 input tester modules. Up to twelve of them can be used to validate the SRP interfaces of the TCC boards. These modules are also used to validate the SRP boards under test, namely their logic that produces selective readout indicators and sends them to the ECAL readout electronics. Up to eight of the input tester modules of the SRP tester firmware are used to validate the logic of tested SRP boards, which is responsible for exchange of the TT classification data with the neighbor SRPs.

VI. Conclusive Remarks

The presented set of the on-line monitoring and test techniques has been initially implemented within the CMS ECAL Selective Readout Processor boards. As the features recommended themselves to be extremely practical, they have been embedded in the endcap Trigger Concentrator Cards as well. They brought several advantages during the development and commissioning of the electronics boards.

The whole process of the firmware development and debugging went very smoothly. Any time a new functional feature was added it was accompanied with the corresponding monitoring and test logic and its control software. Closely coupled pattern generators, on-line test modules and spy buffers allowed thorough validation of the newly added functional feature.

The automation of the respective production test benches has also been simplified. Production test benches check the functionality of individual parts of the hardware modules. Based on the loopback capabilities of most of the interfaces, they have been exercised with associated pattern generation and test modules embedded in firmware. In case of eventual errors determining their cause was in most cases easy with the related statistics modules and spy buffers. No special hardware development was needed for the SRP test bench as the SRP board could be converted to an SRP tester module just by firmware change. As the SRPs can generate desired data patterns and can verify on-line the received results, they can also be used for testing the TCCs and ECAL readout electronics boards (in fact a spare SRP is integrated within the barrel and endcap TCC test bench).

Interoperability tests within the ECAL readout system, as well as the SRP and the endcap TCC commissioning were facilitated. The high rate monitoring capabilities and deep enough spy buffers allowed for fast troubleshooting in case of erroneous behavior. The possibility of running the system with preprogrammed patterns was another handy feature to reproduce encountered errors for debugging purposes.

Finally, the on-line monitoring features embedded in the endcap TCCs and the SRP boards were routinely used during the last year CMS-wide global test runs. They are equally useful now during the LHC first physics runs.

All these advantages largely overweight the penalty of a moderate increase in the FPGA resource utilization. To give a hint of the later, a spy buffer consumes one memory block and about 20 slices of the Virtex2Pro logic recourses. The totality of the 29 spy buffers and the associated control logic within the barrel SRP firmware represents ~2% of the slices and ~9% of the memory blocks of an xc2vp70 FPGA device.

The described remote firmware control mechanisms bring obvious advantages in terms of maintenance of the endcap TCCs and the SRP boards. The capability to keep multiple firmware revisions on-board is indeed a comfortable feature. It makes the SRP boards versatile with the possibility to assign it one of the three desired flavor: barrel, endcap or tester. This eliminates the need to keep different versions of spare boards. With the multiple firmware versions testing of new revisions becomes also less risky: it is always possible to switch back to a well proved revision kept on-board. But the most attractive feature of the presented mechanisms is the capability of automated remote firmware updates which becomes the must for the system of 72 endcap TCCs each with three on-board FPGA devices.

The on-line monitoring and test techniques and the remote firmware control mechanisms are quite generic and are applicable to a wide range of FPGA based real-time systems.

Acknowledgment

We thank Philippe Gras from IRFU, CEA Saclay, Jose Carlos Rasteiro Da Silva from LIP, Yannick Geerebaert and Pascal Paganini from LLR, CNRS/IN2P3 and John Coughlan from RAL for their help at various stages of the project.

References

[1] R. Alemany et al., "Overview of the ECAL off-detector electronics of the CMS experiment", *IEEE Trans. Nucl. Sci.*, vol. 52, Issue 5, Part 3, pp. 1918-1924, Dec. 2005

[2] N. Almeida et al., "The Selective Readout Processor for the CMS ECAL", *IEEE Trans. Nucl. Sci.*, vol. 52, Issue3, Part 2, pp. 772-777, June 2005

[3] Xilinx, "Platform Flash In-System Programmable Configuration PROMs", DS123 (v2.17) October 26, 2009

[4] Xilinx, "System ACE Compact Flash Solution", DS080 (v2.0) October 1, 2008

[5] N. Cardoso, "Design Guidelines For Implementation of Boundary Scan in the ECAL/HCAL Crates", Private communication, available within the CMS ECAL Off-Detector Electronics community

[6] National Semiconductor, "ScanSTA111 Enhanced SCAN bridge Multi-drop Addressable IEEE 1149.1 (JTAG) Port", April 2004

[7] Xilinx, “Platform Cable USB II”, DS593 (v1.2) June 9, 2008
[8] Icron Technologies Corporation, “USB 2.0 Ranger 2101, USB 2.0 Hi-Speed 100m Cat 5 Extender”, Datasheet, 2008
[9] Xilinx, “iMPACT User Guide”, Revision 4.1
[10] CAEN, “MOD. V2718 – VX2718 – N2738, VME – PCI OPITCAL LINK Bridge”, MANUAL REV. 6, 31 January 2006
[11] Texas Instruments, “SN74LVTH182512 3.3-V ABT Scan Test Devices with 18-bit Universal Bus Transceivers”, datasheet, October 1997
[12] I. Mandjavidze, “The TCC48 Remote Firmware Programming Tools: User Guide”, July 2009, https://twiki.cern.ch/twiki/bin/view/Main/PascalPaganiniTriggerExpertDuties
[13] Altera, “ByteBlaster II Download Cable”, User Guide, Revision 1.4, July 2008
[14] Altera, “Introduction to the Quartus II Software”, Version 9.0
[15] N. Almeida, P. Silva, J. C. Silva, J. Varela, “Description of the Synchronization and Link Board ECAL and HCAL Interface to the Regional Calorimeter Trigger”, Version 3.3 (SLB-S), CMS IN 2005/007
[16] Xilinx, “Embedded System Tools Reference Manual, Embedded Development Kit EDK 10.1, Service Pack 3”, July 2007
[17] S.A. Baird et al., "The Front-End Driver card for the CMS Silicon Strip Tracker Readout.", Eighth Workshop on Electronics for LHC Experiments, Colmar, France, September 2002, CERN/LHCC/2002-034
[18] D. Calvet et al., “The Back-end Electronics of the Time Projection Chambers in the T2K Experiment”, presented at this conference
[19] S. Anvar, O. Gachelin, P. Kestener, H. Le Provost, I. Mandjavidze, “FPGA-based System-on-Chip Designs for Real-Time Applications in Particle Physics”, IEEE Trans. Nucl. Sci., vol. 53, Issue3, Part 1, pp. 682-687, June 2006