Implementation Issues of the LHCb Readout Supervisor

Z.Guzik, A.Chlopik
Soltan Institute for Nuclear Studies, 05-400 Swierk-Otwock, Poland
Zbig@ipj.gov.pl, Arek@ipj.gov.pl

and

R.Jacobsson,
CERN, 1211 Geneva 23, Switzerland
Richard.Jacobsson@cern.ch

Abstract

In this paper we describe the architecture of the most crucial and sophisticated element of the LHCb Timing and Fast Control (TFC) System - the Readout Supervisor (RS). The multi-functionality, the complexity and the speed demands dictate usage of the most advanced and performant technological solutions. The logical part of the Readout Supervisor is therefore based on the fastest PLDs on the market i.e. the Altera MAX and FLEX devices working on 2.5V. There are 12 such units implemented where each unit carry separate logical functions. The front-end logic of the module is designed with positive ECLinPS Lite ICs. The Experiment Control System (ECS) interface to the Readout Supervisor is based on a commercial Credit Card PC from Digital Logic AG.

I. INTRODUCTION

The Readout Supervisor is a central component in the LHCb Timing and Fast Control (TFC) system. The functional specifications and the detailed description of all the Readout Supervisor tasks, features and nodes have been covered in document [1].

The current implementations of the first minimal version of the Readout Supervisor is covered in a separate document [2]. There are a few minor differences between the actual implementation described in [2] and the one proposed in the *Readout Supervisor Design Specification [1]*. Therefore, the final assignments and resource allocations are only **valid** as presented in [2].

II. DESIGN CRITERIA

During designing and implementation process following design criteria were obeyed:

- Modular approach to logical design - entire RS design divided into logically consistent nodes (entities) which are programmed on separate PLD's. Logical connections between modules have fully pipelined structure.
- The current prototype version of Readout Supervisor is implemented with a FASTBUS board form factor; only one external +5V power supply is used, either via the FASTBUS connector or from an on-board special IBM-PC type connector. The latter means that a FASTBUS power–supply/crate is not necessary. On-board DC/DC converters from DATEL provide the three additional voltages: +3.3 V (7 A), +2.5 V (5A) and -5V (2A).
- Interfaces concerning external trigger data are based on LVDS technology. The *National* DS90C402 chip was selected as a dual receiver, while DS90C401 chip was chosen as a dual LVDS driver.
- All discrete fast logic (clock regeneration and TTC encoder) will be realized with MOTOROLA *ECLinPS* or *ECLinPS Lite* integrated circuits from the 100E, 100EL or 100ELT series.
- The TTC encoder (Channel **A** and Channel **B** time division multiplexor for broadcasting L0, L1 triggers and commands) is based - with slight modifications – on the TTCvx module by Per Gällnö. Also TTCex design from Bruce Taylor is taken into consideration as an alternate mezzanine. TTC encoder together with clock PLL regeneration is located on separate mezzanine board
- All Readout Supervisor functional logic will be performed by Altera MAX 7000AE (for the most time critical parts) and FLEX 10KE PLD devices. All of the used PLDs (except PLD constituting IOBUS) have the same 144-pin count. For easy debugging and to facilitate connecting a Logic State Analyzer, all PLD's will be placed on separate Mezzanine sub-boards with 100 mils pin spacing.
- All PLD's are working with 3.3 V power supply for input-output (VCCINT). The in-core PLD logic (VCCINT) for MAX devices is the +3.3 V and for FLEX devises is the +2.5V. The speed grades of the selected PLDs must not be worse than "5" for the MAX devices and "1" for the FLEX devices. Configuration and programming of all PLD's is organized by means of programmable JTAG chain.
- All the implemented FIFO's are the CY7C4251 from Cypress. They are 8Kx9 synchronous devices with 10 ns access time and 3 ns setup time.

- MAX+plus II and AHDL were chosen as the PLD design environment for all modules except for the random trigger generator module. In the future the aim is to translate all designs into VHDL. The PLD designs are simulated in MAX+plus, as well as in Cadence using test benches in VHDL.
- The schematics and the PCB layout are done with the help of the Protel Design Explorer 99 SE software with Service Pack 6.
- The Readout Supervisor is controlled by Commercial Credit Card PC via Ethernet Link. Interfacing with its PCI bus is organized with help of *PLX* PCI 9080 accelerator chip forming local bus.
- There is no single jumper on the Readout Supervisor Board.

A schematic block diagram of the entire Readout Supervisor is presented in Figure 1.

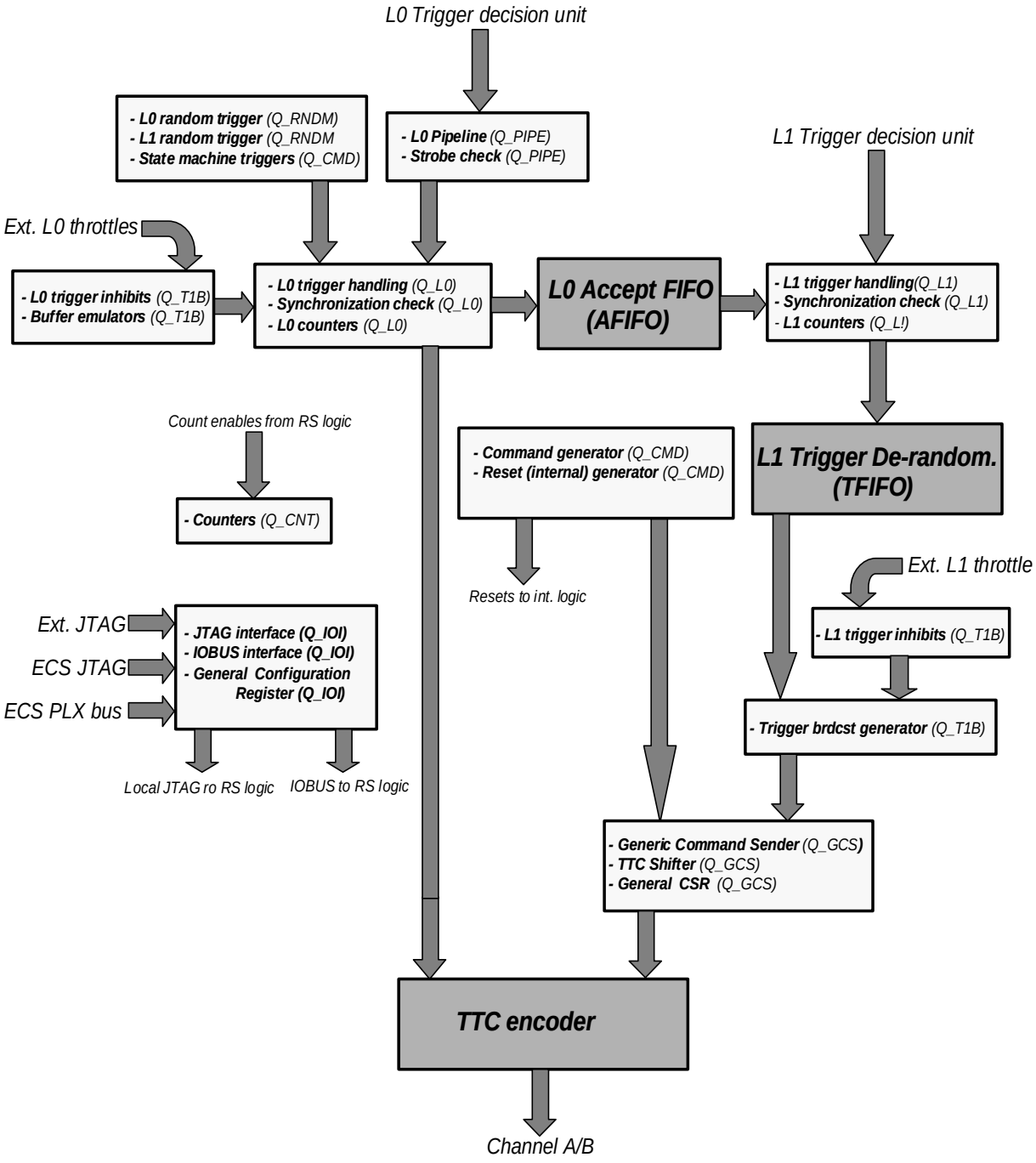

Figure 1: Block Diagram of the Readout Supervisor

III. ECS INTERFACE

The Experiment Control System (ECS) interface to the Readout Supervisor and its associated logic is presented in Fig.2 below.

The ECS interface is based on a commercial Credit Card PC (CC-PC) from *Digital Logic AG*. Its access to the on-board logic is provided by means of *smart480BUS* consisting of 480 pins. The *smart480BUS* resources include a PCI bus. This is the basic medium to exchange information between the CC-PC and the Readout Supervisor logic. In between the CC-PC and the Readout Supervisor, there is an intermediate interface, the so-called "Glue Board", which contains a PCI 9080 chip from *PLX Technology*. The PCI 9080is a PCI-to-Local Bus accelerator chip working in J-mode (multiplexed a/d mode). The CC-PC is accessed externally via an ETHERNET LAN and, optionally, via a serial RS-232 link.

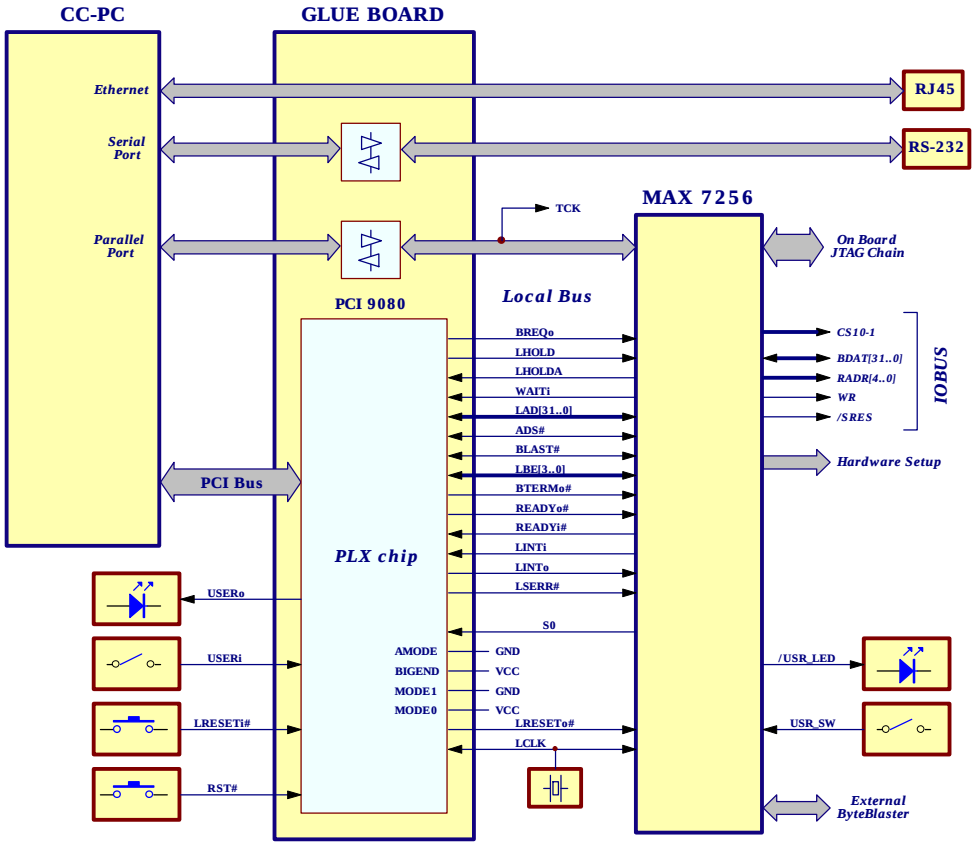

Figure 2: Structure of the Experiment Control Interface

The Glue Board also has JTAG interface incorporated that can be used for indispensable PLD's configuration and programming. JTAG interface is composed from parallel printer port.

All PLDs are accessed internally by means of an IOBUS formed from PCI 9080 Local Bus

IV. PLD IN-SYSTEM PROGRAMMABILITY AND CONFIGURATION

In the Readout Supervisor, two types of PLDs are used. The programming mechanism is different for the two. While the MAX 7K devices retain their configuration when power is switched off, the FLEX 10KE must be re-configured after each power-down. In the current design of the RS there are six (+1 reserve) MAX 7K devices and four FLEX 10KE devices. We have focused exclusively on JTAG for programming, and the Altera native configuration system has been skipped.

One PLD (MAX 7256-208) is used for interfacing between CC-PC Glue Board PLX chip and rest of the Readout Supervisor logic. Separate Byte Blaster driven externally via on-board header should program this PLD. Remaining all PLDs are programmed or configured by JTAG interface located on PLX chip Glue Board. This MAX 7256 PLD contains JTAG distribution logic for all other PLDs. The **TCK** JTAG clock lines for programmed PLDs are driven directly from the „Glue Board" JATG Interface. The remaining three JTAG lines (**TMS**, **TDI** and **TDO**) are passed through the MAX 7256 and are distributed to every other PLD individually (see Figure 3). Proposed approach allows configuring and programming any set of selected PLDs - from one to all of them.

The selection of a specific PLD to be configured is made via programmable register (CSR) contained in the MAX 7256 device (***Q_IOI*** module). When any " **x** " PLD is unselected then its **TMS_x** and **TDO_x** lines are driven permanently HIGH and its **TDO_x**/**TDI_x** pins doesn't participate in the closed JTAG chain. When a given "**x**" PLD is selected for programming or configuration then its **TMS_x** is controlled by the "Glue Board" JTAG Interface **TMS**. Its **TDO_x**/**TDI_x** lines then constitute the JTAG closed chain together with the other selected PLDs. The order of PLDs in a global JTAG chain is given in Table 1. The device is selected when the appropriate bit in CSR is set HIGH. In working conditions all FLEX devices are selected and all MAX devices should be deselected.

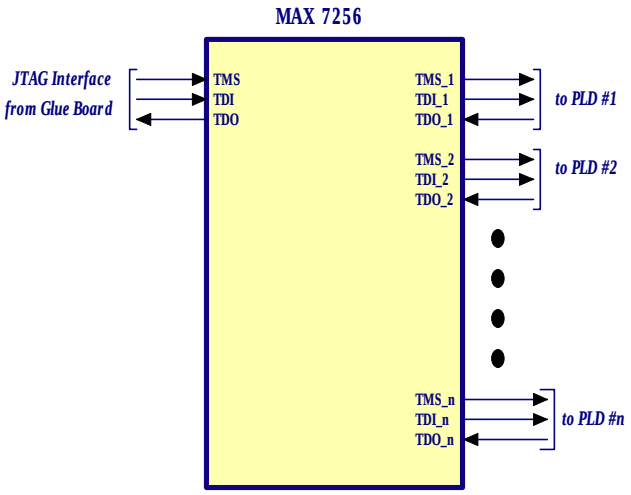

Figure 4: Structure of the On-board JTAG Distribution

V. SYNCHRONIZATION

The Readout Supervisor receives the LHC bunch clock and LHC orbit signal from the TTCmi. The bunch clock is used without any phase adjustments but it is regenerated by a Phase Locked Loop Frequency Multiplier (MPC991) to produce the basic 40.08 MHz clock (**BCLK**), and the 80 and the 160 MHz clocks that are necessary for TTC encoding.

The external orbit signal is passed through a delay line to be phase adjusted to the internal **BCLK**. The PDU54-1500 from *Data Delay Devices* is used as delay line. It has 16 steps of 1.5 ns. It is also possible to work without external synchronization signals. When selected the **BCLK** and the **ORBIT** signals are produced internally.

Synchronizing the external L0 and L1 trigger data with the internal clock is an important task. It is achieved by means of clock edge selection as described below. The trigger data are received according to the timing diagram presented in Figure 4. The data are accompanied by a strobe signal (every clock cycle for the L0 trigger and every decision for the L1 trigger). The received data are written into an internal buffer at the rising edge of the strobe (*First Pipeline*). The external strobe is subsequently delayed by approximately 5 ns and is or'ed with original one. The presence of the strobe is tested by sampling this or'ed signal with both the positive and the negative edge of the internal clock. Selecting the good clock edge is made by the **H_0PHASE** parameter (for L0) and the **H_1PHASE** parameter (for L1) and is established during the timing alignment of the experiment. If the negative edge of the local clock was chosen as the proper one (as on the picture), then the data are stored in a *Second* Pipeline at this edge. After another half a clock period the data is transferred into the *Third Pipeline* at the positive edge to form the final data for this clock cell. Of course, if the positive edge was chosen, then the *Second Pipeline* step is skipped. In addition, for L1 triggers, a validation strobe is produced to be used as a write enable to the L1 Trigger De-randomizer.

For proper on-board clock distribution, separate PECL differential pairs of equal length are pulled to each PLD. Translators from PECL to TTL are placed at the closest distance to the clock pins of each PLD. A MC100E111 clock driver distributes the PECL clock lines in star fashion.

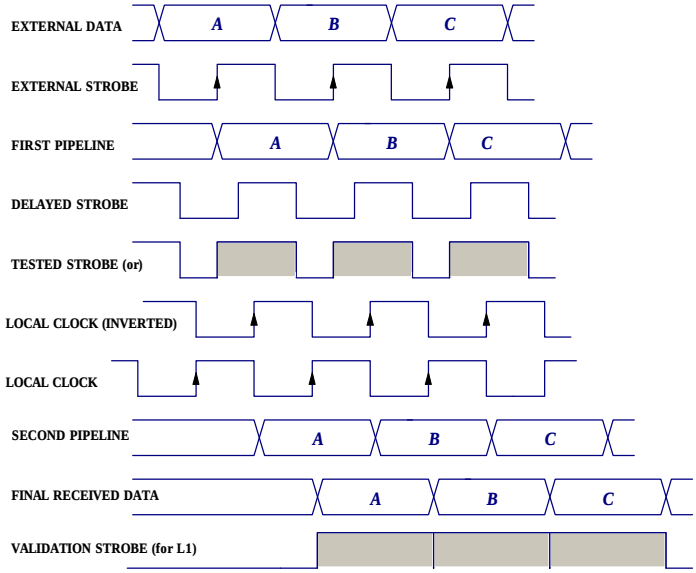

Figure 3: Timing Diagram Showing the Synchronization Principle

VI. TTC MEZZANINE

The TTC Mezzanine function is to multiplex and encode A and B channel signals generated by appropriate PLD's. The another task of this sub-board is to regenerate Bunch Clock or in case of its absence to generate internal 40 MHz clock. Switching between internal and external clock is realized by **H_EXT** level generated in **Q_IOI** module. The A and B channels are time division multiplexed and bi-phase mark encoded (see Figure 5). Phase locked loop frequency synthesizer circuit handles clock multiplication necessary for the encoding.

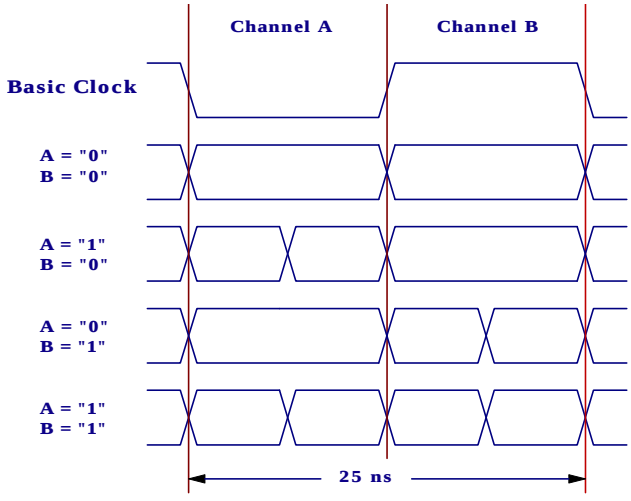

Figure 5: Encoder Output Wave Forms

VII. PLD MODULES

For the first prototype design we have chosen solution with a separate PLD for each logical entity (module). For the most critical parts of the system the Altera MAX 7000AE PLD is used – nowadays it is the fastest PLD on the market. For the more complex entities, such as long multiple counters or random generators, the Altera FLEX 10KE is sufficient. There are 6 MAX devices and four FLEX devices used for the entire project. One reserve MAX PLD will also be mounted on the board.

A. *I/O Interface & Resets*

The main task of this module is to act as an interface between the ECS Glue board and the Readout Supervisor logic. It controls the internal IOBUS by providing dedicated chip selects and common control signals to all the Readout Supervisor PLDs. Another task of this module is to provide system reset and to distribute programmable JTAG chain to other PLDs. It is the only PLD programmed by external Byte Blaster.

B. *L0 External Trigger Phasing & Pipelining*

The primary function of this module is to provide a 16-stage delay pipeline for L0 Trigger path, where the pipeline depth is programmable. Besides that it phases the incoming L0 trigger data to the internal clock and detects missing input strobes.

C. *L0 Trigger Handling*

This module receives all the different types of Level-0 triggers (external and internal) and compiles a final trigger qualifier according to the trigger priority, presence of the L0 inhibit and possible errors. It also performs the synchronization check of the incoming external L0 triggers. The accepted L0 triggers are written into the L0 Accept FIFO (AFIFO) and the YES decisions are broadcasted over the TTC Channel A. Additionally, this module contains the L0 gap generator, which, if needed, can force gaps of programmable length between L0 trigger accepts.

D. *L1 Trigger Handling*

This module receives the L1 external trigger data and performs a synchronization check with the corresponding L0 triggers contained in the L0 Accept FIFO. It evaluates the L1 triggers and writes them into the L1 trigger de-randomizer (TFIFO). It also ensures that internal triggers are maintained when external L1 trigger path is blocked.

E. *L1 Trigger Broadcasts & Inhibits*

This module contains a rate controller for the L1 trigger broadcasting (real broadcasting is performed by another module). The second task of this module is to centralize the evaluation of all the different L0 and L1 inhibits in order to produce a single combined L0 inhibit and a single combined L1 inhibit. It also runs a L0 front-end de-randomizer occupancy controller and a L1 front-end buffer occupancy controller.

F. *Generic Command Sender & General Status Register*

The main task of this module is to resolve all of the incoming requests for trigger and command broadcasting. The module ensures that the command broadcasts get higher priority than the pending trigger broadcasts. It also makes sure that the Bunch Counter Resets and the Event Counter resets are sent with highest priority at the appropriate times. If a +command broadcast request is refused due to the Generic Command Sender being busy, the command is postponed until the same bunch crossing in the next LHC turn. In case a trigger broadcast request is refused, it is delayed until the Generic Command Sender is free.

Another important function of this module is to maintain all the general status bits of the entire Readout Supervisor. Finally, it also generates the internal orbit signal when internal synchronization is selected by the user and detects presence of external orbit signals.

G. *Command Generator & Internal Triggers*

This module runs a set of state machines that at the appropriate times request sending the different type of commands (resets, calibration pulsing, etc). It also generates all the signals, which should accompany certain commands. It also generates all the internal L0 triggers, except the random triggers. Another important function of this module is to maintain and generate dedicated resets to all logical nodes that need to be cleared individually.

H. *Random Generator*

The Random Generator module produces random L0 triggers that are injected into the L0 trigger handling path. It can also generate random L1 triggers by randomly forcing a subset of the random L0 triggers at level one. Optionally, the module can also be configured to force every or none of the random L0 triggers. The triggers are generated according to a Poisson distribution and the rate is fully programmable as will be explained in the appendix concerning the Random Generator.

I. Universal Counter Modules

Each of this module contains sixteen 32-bit counters every of which increments when corresponding ”count enable” input is produced by appropriate module. Optionally, each counter may be pre-scaled by a common pre-scale factor. In this minimal version of the RS, there are two such modules. All counters are presented in table 1.

VIII. REFERENCES

[1] *R.Jacobsson, B.Jost and Z.Guzik – Readout Supervisor Design Specification, LHCb Technical Note, LHCb 2001 – 012 DAQ, CERN, February 12, 2001.*

[2] *Z.Guzik and R.Jacobsson – "ODIN" – LHCb Readout Supervisor, Technical Reference, Revision 1.4 – September 2001.*

...

Table 1: Implemented Readout Supervisor Counters

#	**Description**	**Proposed prescaling**
0	External L0 sync errors ungated	-
1	External L0 sync errors gated	-
2	External L0 accepts converted to NO by sync error	-
3	Total L0 accept ungated	4
4	Total L0 accept gated	4
5	Total L0 forces ungated	-
6	Total L0 forces gated	-
7	External L0 accepts ungated	4
8	External L0 accepts gated	4
9	External L0 force ungated	-
10	External L0 force gated	-
11	Sequencer periodic L0 triggers ungated	-
12	Sequencer periodic L0 triggers gated	-
13	Random triggers L0 ungated	4
14	Random triggers L0 gated	4
15	reserved	
16	Bunch Clock	2^10
17	Bunch Clock gated by L0 Inhibit	4
18	Bunch Clock gated by L1 Inhibit	4
19	L1 external sync errors	-
20	External L1 Accepts (Triggers)	-
21	reserved	
22	L1 Random Force	-
23	L1 Forces (from AFIFO)	-
24	Total writes to TFIFO	4
25	Accepted writes to TFIFO	-
26	Total broadcasts of L1 Trigger commands	4
27	Total number L1 positive trigger retrieved from TFIFO	-
28	Total broadcast of L1 Positive Triggers commands	-
29	Number of Turns	-
30	Bunch Clock gated by L0 External Throttle	-
31	Bunch Clock gated by L1 External Throttle	-