LHCb 2001 – 012 DAQ
March 7, 2001

Readout Supervisor *Design Specifications*

R. Jacobsson, B. Jost and Z. Guzik

ABSTRACT

The Readout Supervisor is the heart of the LHCb synchronous readout. We specify here its functionality and requirements.

LHCb Technical Note

Issue:	LHCb 2001 – 012 DAQ
Revision:	1
Reference:	LHCb DAQ
Created:	16 April 2000
Last modified:	7 March 2001

Prepared By: R. Jacobsson and B. Jost, CERN, Geneva, Switzerland
Z. Guzik, The A. Soltan Institute for Nuclear Studies, Warsaw, Poland
LHCb DAQ Group

Table of Contents

1 INTRODUCTION 3

2 FUNCTIONAL SPECIFICATIONS 5

2.1 INTRODUCTION 5
2.2 READOUT SUPERVISOR SYNCHRONISATION 6
2.3 L0 TRIGGER DISTRIBUTION 6
2.4 L1 TRIGGER DISTRIBUTION 8
2.5 READOUT SUPERVISOR DATA 9
2.6 FRONT-END/DAQ BUFFER OCCUPANCY CONTROL AND THROTTLING 10
2.7 BUNCH COUNTER RESET (BCR) AND EVENT COUNTER RESET (ECR) 11
2.8 FRONT-END RESET GENERATION 12
2.9 ONLINE DETECTOR CALIBRATION/MEASUREMENTS 13
2.10 STAND-ALONE DETECTOR CALIBRATION/MEASUREMENT RUN 14
2.11 TEST/DEBUG RUN 15
2.12 OTHER COMMANDS TO THE FRONT-END 15
2.13 COUNTERS 16
2.14 THE ECS INTERFACE 16

3 MODULAR SPECIFICATIONS 18

3.1 INTRODUCTION 18
3.2 L0 TRIGGER DISTRIBUTION PATH 19
3.2.1 L0 trigger pipeline (Not drawn in Figures 6 and 7) 19
3.2.2 L0 synchronization check 19
3.2.3 L0 Accept FIFO 19
3.3 L1 TRIGGER DISTRIBUTION PATH 20
3.3.1 L1 synchronization check 22
3.3.2 L1 trigger de-randomizer 22
3.4 RANDOM GENERATORS 26
3.4.1 L0 random trigger 26
3.4.2 L1 random trigger 28
3.5 TRIGGER, RESET AND CONTROL COMMAND SEQUENCER 28
3.5.1 Command broadcasting 35
3.6 RESET CONTROLLER 35
3.7 TTC SERIALIZER/ENCODER 36
3.8 TRIGGER DISTRIBUTION CONTROLLER 36
3.8.1 L0 de-randomizer occupancy emulator 37
3.8.2 L1 buffer occupancy emulator. 37
3.8.3 Throttle history 38
3.9 “READOUT SUPERVISOR FRONT-END” 38
3.9.1 L0 39
3.9.2 L1 40
3.10 COUNTERS 40
3.11 ADJUSTABLE DELAY 40
3.12 AUXILIARY L0 TRIGGER INPUT 40
3.13 INTERNAL OSCILLATOR 41
3.14 THE ECS INTERFACE 41
3.15 RS BOARD STANDARD 41

4 POWER UP STATE AND INITIALIZATION 42

4.1 POWER UP STATE 42
4.2 INITIALIZATION 42

5 SUMMARY OF THE RS PARAMETERS, COUNTERS, I/O INTERFACES AND STATUS LEDS 43

6 APPENDIX: FIRST STAGE RS PROTOTYPE (2001) 48

7 REFERENCES 51

1 Introduction

The Readout Supervisor (RS) is the active component in the LHCb Timing and Fast Control (TFC) system[**1**] and is thus the heart of the synchronous readout. The system architecture is shown in Figure 1.

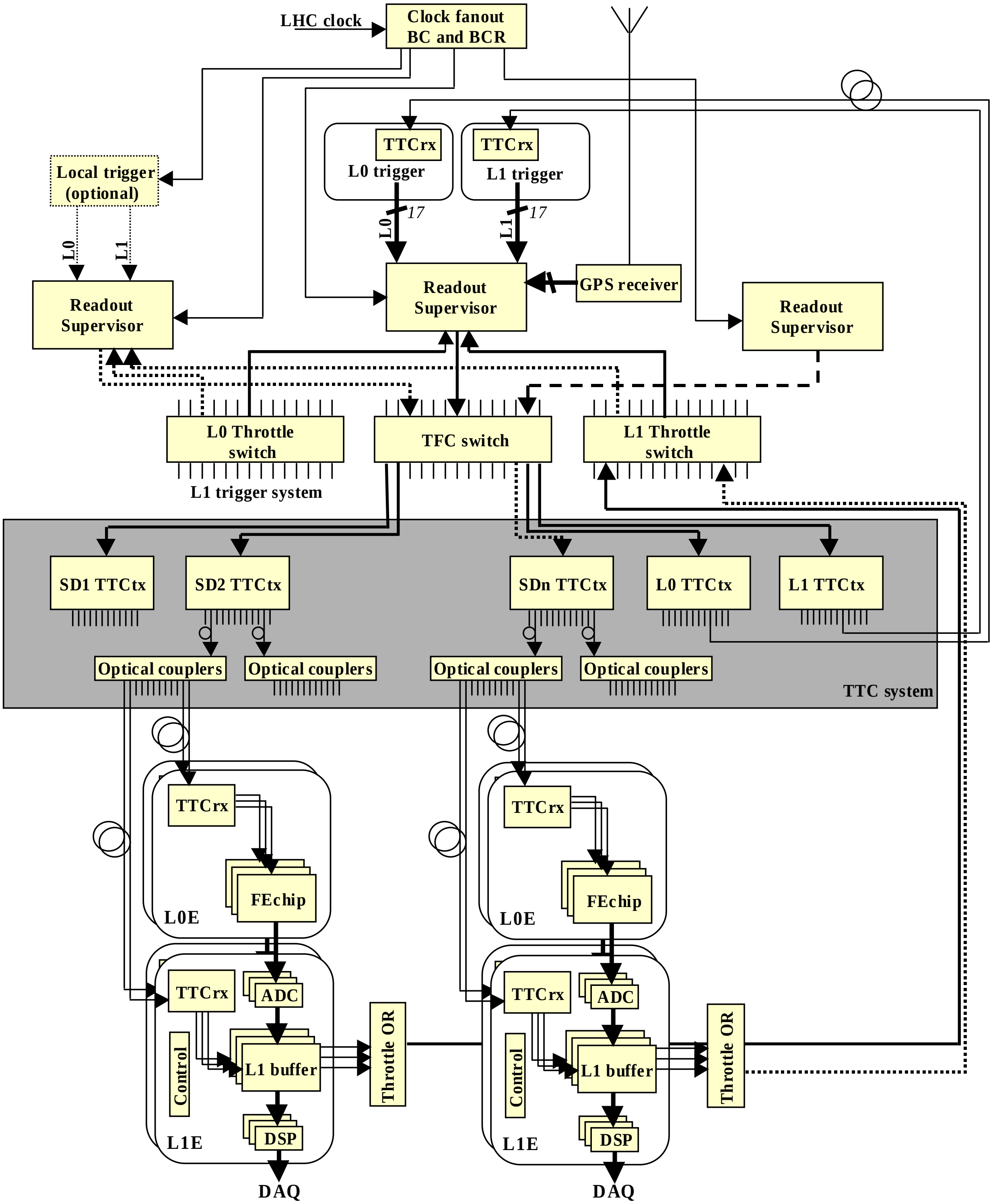


Figure 1: General architecture of the LHCb TFC system.

The Readout Supervisor receives the LHC bunch clock via the LHC machine interface and the Level 0 and 1 trigger decisions from the central L0 and L1 decision units. It distributes these, as well as internally generated triggers and various synchronous control commands, to the Front-End (FE) electronics via a programmable patch panel (TFC Switch[**2**]) and the CERN Trigger, Timing and Control (TTC) system developed by RD12[**3**]. The TFC Switch allows a sub-detector to run in stand-alone mode or with any other combination of sub-detector components (partititions)[**4**]. Different sub-systems can also run in parallel by using separate Readout Supervisors.

The Readout Supervisor will also act as trigger rate controller. It will receive trigger throttle signals from any system that can have data congestion. The throttle signals will be ORed to make a single L0 throttle and a single L1 throttle, which are fed back to the Readout Supervisors via the Throttle Switches[**2**]. The throttle switches are also programmable in order to pass on the throttle signals from the different partititions to the Readout Supervisor controlling the partitions in question.

The LHCb standard Experiment Controls System (ECS) Interface to board level electronics will be used to configure and monitor the Readout Supervisor.

Sections 2 and 3 both describe the functionality of the Readout Supervisor but from two different points of view. Section 2 specifies the general functionality of the Readout Supervisor without discussing its modular implementation. Section 3 considers the implementation issues and specifies the requirements of each component in order to implement physically the functionality described in Section 2. Section 4 specifies the power up state and the initialization of the Readout Supervisor. Section 5 summarizes in tabular form the Readout Supervisor internal counters, the programmable parameters, the I/O interface and the status LEDs.

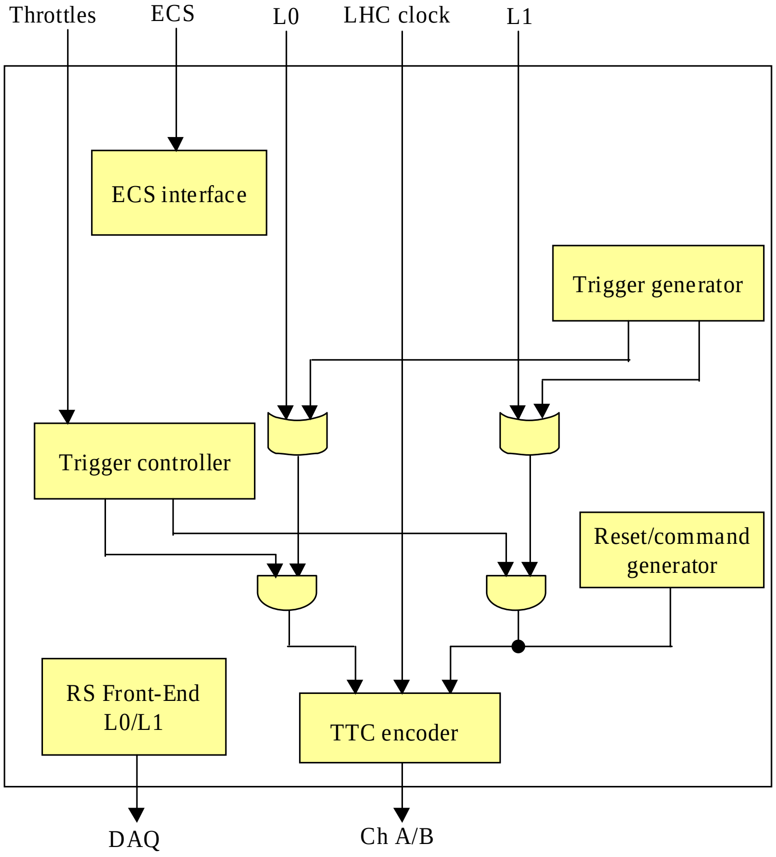


Figure 2: Simplified block diagram showing the principal features of the Readout Supervisor.

2 Functional Specifications

2.1 Introduction

As can be seen in Figure 2 the principal tasks of the Readout Supervisor are to:

- Distribute the LHC clock to the entire FE electronics and the trigger systems.
- Distribute the L0 trigger decisions to the L0 FE electronics.
- Distribute the L1 trigger decisions to the L1 FE electronics
- Generate and time-in all types of self-triggers (random triggers, calibration etc)
- Control the trigger rate by taking into account the status of the different components in the system in order to prevent buffer overflows and to enable/disable the triggers at appropriate times during resets etc .
- Generate and time-in resets (counters- and electronics–) and other asynchronous commands.
- Record detector status information and information related to timing, triggering and fast control in a special data block and transmit them to the event building.
- Incorporate an ECS interface for configuring, controlling and monitoring the Readout Supervisor.

The Readout Supervisors distribute the signals to the optical transmitters (TTCtx's) via a switch (TFC Switch) to allow for partitioning. Seen from the TFC system, partition granularity is defined by the smallest sub-system which can be triggered independently. In the described system, the partition granularity is defined by the sub-systems covered by each of the TTCtx's's[1]. The TFC Switch is a programmable patch panel that allows each sub-system, associated with one TTCtx, to select a given Readout Supervisor as source for timing, control and triggering by setting up the connectivity in the switch. This means a sub-system can run independently stand-alone or in combination with others to form different partitions. This possibility is intended for special runs, debugging, testing and calibration. One Readout Supervisor will function as the central real-time controller for normal physics data-taking during which the switch is set-up to incorporate all sub-systems.

The Readout Supervisor should have four "modes" of running:

- **Physics run:** Normal running with LHC beams in which a central RS is used and receives the L0 and L1 decisions from the central trigger decision units. RS internal auto-generated triggers for calibration and monitoring can be injected during LHC bunch gaps and randomly.

[1] A note is in preparation to describe the LHCb partitioning concept in detail.

- **Calibration run:** Run with auto-triggers, generated internally in the RS, and forced L0 and L1 accepts. The calibration sequence and rate is programmable.

- **Timing alignment run:** Run with LHC in "special bunch mode"[2]. The RS will be set up to accept a set of consecutive bunch crossings around the true bunch crossing in order to do bunch profile histogramming and thereby align the timing of the Front-End, the Trigger and the Readout Supervisor.

- **Test/debug run:** Run with pre-scalable internal random trigger and forced L0 and L1 accepts.

2.2 Readout Supervisor synchronisation

The Readout Supervisor should receive electrically the LHC bunch clock and the LHC orbit signal (transmitted by the PCR) from the clock fan-out (TTCcf) of the LHC machine interface (TTCmi).

The LHC orbit signal (Bunch Counter Reset -- BCR) should be distributed to the Front-End with highest priority to control the phase of the TTCrx's. A look-ahead state machine should ensure that the Front-End receives the BCR at the appropriate time (see Section 2.7).

2.3 L0 trigger distribution

The Readout Supervisor should receive the L0 trigger decision (1-bit), be it from the central L0 Trigger Decision Unit (L0DU)[**5**] or an optional local unit, together with a 12-bit Bunch ID as a parallel 16-bit data transfer using a strobe. There should be means to phase adjust the strobe.

Among the remaining bits, one bit is to be defined to tell if a L0 accept has been forced by the L0DU. As a consequence, the Readout Supervisor should force the L1 decision. A special option in the Readout Supervisor, should also allow forcing the L1 decison for every L0 trigger accepted by the L0DU.

A second bit ("timing trigger bit") is to signal the Readout Supervisor that triggers for timing alignment should be generated (described below).

In order to adjust the global latency of the entire L0 trigger path (FE-L0DU-RS-TTC-FE) to a total of 160 cycles[**6**], the Readout Supervisor should have a pipeline of programmable length at the input of the L0 trigger. The pipeline should store the 16 bits of information for each trigger during a programmable number of cycles[3] before passing them on through the Readout Supervisor.

To cross-check coherence, the Readout Supervisor should, after the pipeline, compare the received Bunch ID with the expected value from an internal bunch crossing counter. In case an error is

[2] This mode may consist of a few bunches per beam of which at least one pair will collide at the LHCb crossing point.

[3] The maximum length of the pipeline is still to be determined (16 – 32)

detected, a status bit should be set. The status bit should be read by the ECS interface in order to raise an alarm. During the commissioning, for debugging purposes, the trigger will have to be accepted and flagged as erroneous in the RS data pipeline. The corresponding L1 trigger must also be forced in order to keep the event. The alarm can be used by the ECS to inject a reset of the appropriate part of the system. Eventually, when the source and the frequency of this type of synchronisation are understood, the de-synchronised triggers can be rejected. Whether to keep or to reject de-synchronized triggers should be programmable options.

The error detection described above will detect discontinuity due to the delay of a L0 trigger by one or more clock cycles. In this case the system will continue out of synchronization. The Readout Supervisor should after a programmable number of consecutive synchronization errors set a second status bit. This bit should be used by the ECS interface to raise an alarm.

However, in order to know if the de-synchronized trigger was due to a missing strobe, the RS should also have a mechanism to detect missing strobes. There should be a status bit to mark that a single strobe was missing and a second bit to mark that a programmable number of consecutive missing strobes occured.

It should be possible to disable the L0 trigger input via the ECS interface. A consequence of disabling the L0 trigger input must also be suppressing synchronization errors and disabling the strobe check mechanism mentioned above.

The L0 Event IDs of the L0 accepts should be stored in a FIFO in order to check the coherence of the L1 triggers. A marker bit for each slot in the FIFO should indicate if the L0 has been forced. Forcing a L0 decison only makes sense if also the L1 decison is forced. By storing the force bit, the corresponding L1 decision, arriving much later, can be forced too irrespective of the trigger decision. In addition, the FIFO should receive information about the source of each trigger to store the L1 trigger qualifier (Section 2.4).

Any forced L0 accept should be marked as such in the RS data block together with the trigger type (see Section 2.5).

A special mode of running is timing alignment. In this mode, the L0DU signals the Readout Supervisor that a ‘good crossing’ for timing alignment has been detected. Since the L0DU itself has a pipeline it can signal the RS in advance by transmitting the bit set with the first trigger pending. The Readout Supervisor should react by forcing a programmable number of consecutive L0 triggers before and after the ‘good crossing’ in order to histogram the bunch profile.

The Readout Supervisor should be equipped with a second auxiliary L0 trigger input (1-bit, no Bunch ID) on the front panel to allow connecting another source, such as an optional local trigger decision unit. The selection of the effective input should be programmable from the RS controller. The auxiliary input should have an adjustable phase delay and an optional NIM signal converter.

There should also be an L0 decision output (1-bit) for monitoring purposes.

The L0 trigger decision should be output of the RS with a maximum latency of 100 ns (4 cycles) counted from the end of the pipeline, and distributed to the FE electronics via the low-latency channel A of the TTC system in the form of an accept/reject signal.

2.4 L1 trigger distribution

The Readout Supervisor should receive the L1 trigger decision (1-bit), be it from the central L1 trigger decision unit(L1DU)[**7**] or a local unit, together with a 12-bit L0 event-ID as a parallel 16-bit data transfer using a strobe. The strobe should pass through an adjustable delay to allow phase adjustments. In addition, one bit is to be defined to tell if a L1 accept has been forced by the trigger.

To ensure coherence, the Readout Supervisor should compare the received L0 event-ID with the expected value from the first L0 event-ID pending in the L0 accept FIFO. This will also ensure that the L1 triggers arrive in the correct order, i.e. with monotonously increasing L0 event-ID. In case an error is detected, a status bit should be set which should be used by ECS interface to raise an alarm. A second status bit should mark that a programmable number of consecutive synchronization errors occurred. During the commissioning, the de-synchronized triggers will have to be accepted and flagged as erroneous in the RS data pipeline for debugging purposes. The alarm can be used by the ECS to inject a reset of the appropriate part of the system. Eventually, when the source and the frequency of this type of synchronisation error are understood, the de-synchronized triggers can be rejected. Whether to keep or to reject de-synchronized triggers should be programmable options.

It should be possible to disable the L1 trigger input via the ECS interface by ignoring the external L1 strobes. A consequence of disabling the L1 trigger must also be suppression of synchronisation errors and enabling an internal L1 strobe generation mechanism to be used for internal stand-alone triggering.

In the special case that a L0 accept has been forced, either by the L0 trigger unit or by the RS, the marker in the L0 accept FIFO will imply that the L1 trigger should be accepted.

Any forced L1 accept should be marked as such in the RS data block (see Section 2.5).

As the L1 trigger processing time is variable but the triggers are ordered in a re-organizer, the trigger decisions will come in bursts[**8**]. Due to the limited bandwidth of the L1 distribution channel and the fact that commands, such as resets, will be broadcast on the same channel but with higher priority, the Readout Supervisor should have a L1 trigger de-randomizer buffer to cope with the bursts. To control the rate, the Readout Supervisor should transmit the L1 trigger decisions with a programmable spacing. It should be possible to specify a different spacing after a L1 accept and after a L1 reject.

The L1 trigger decision should be distributed to the FE electronics as a TTC short format broadcast. The short format contains an 8-bit word out of which six bits can be user-defined and the two lowest order bits are reserved. For the L1 trigger decision, the user definable bits should contain an encoded 3-bit qualifier plus the two lowest-order bits of the L0 event-ID. This information should be retrieved from the L0 accept FIFO and be stored together with the L1 decision bit in the L1 trigger de-randomizer. The highest order bit should be set to indicate that the broadcast is a L1 trigger decision. The L0 event-ID will be used to check the coherence with the expected value at the Front-End. The transmitted L0 event-ID should be the one expected by the Readout Supervisor in order to be able to keep de-synchronized triggers. However, both the RS internal L0 event-ID and the one received from the L1 trigger system should be stored in the RS data block.

The qualifier allows specifying different types of triggers, as proposed in Table 1[**9**].

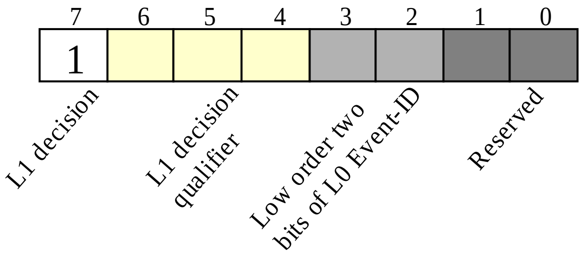


Figure 3: Format of the L1 decision word.

The Readout Supervisor should serialize the L1 trigger words and encode them with the L0 triggers in order to broadcast both onto a single distribution channel (the TTC [**3**]).

The Readout Supervisor should have a L1 decision output (1-bit) on the front panel for monitoring purposes.

Table 1: L1 decision qualifiers. The priority defines how the trigger is denoted in case triggers accepts from different sources coincide.

Bits <6:4>	Interpretation	Priority
0	Discard event	N/A
1	Trigger on physics event	0
2	Reserved[4]	1
3	Random trigger	2
4	Reserved[4]	3
5	Reserved[4]	4
6	Triggers for timing alignment	5
7	Trigger on calibration pulse	6

2.5 Readout Supervisor data

The Readout Supervisor should have L0 and L1 data buffers analogous to those of the Front-End in order to supply the events with an RS data block for each L1 accept. The RS data block should contain:

- Bunch ID from the RS counters and the Bunch ID from the L0 trigger system
- L0 Event ID from the RS counters and the L0 Event ID from the L1 trigger system
- L1 Event ID from the RS counters
- Universal time as defined by the GPS system
- Bunch information from beam pick-ups[5]
- The experiment status information provided through the front-panel input
- Trigger type (physics, random, calibration, timing, test, etc)
- Information on L0 and L1 forced
- Error blocklet

[4] The reserved qualifiers will typically be associated with state machines generating different type of self-triggers.

[5] If possible it would be interesting to set up a system connected to the two beam pick-ups in order to record the true state of each bunch crossing. If it turns out not to be practical it would at least be interesting to store the expected type of bunch crossing by sequencing it in the RS.

Some of this information is available only when the L0 and the L1 trigger decisions arrive. This information will therefore have to be appended when transferring the accepted events into the L0 de-randomizer and the L1 de-randomizer, respectively.

The “RS L1 Front-End” should be interfaced to the DAQ in the same way as the normal Front-End.

2.6 Front-End/DAQ buffer occupancy control and throttling

There are many buffers at various stages of the readout that can overflow. It is the task of the Readout Supervisor to prevent overflows by throttling the triggers. In practice this is achieved by converting trigger accepts into rejects.

Due to the large distance between the Readout Supervisor and the FE electronics, and the bunch crossing and trigger rates, L0 Front-End overflows cannot be controlled in a direct way. However, as the readout is fully synchronous with constant L0 latency and constant L0 de-randomizer readout time of 900 ns (36 cycles)[**10**], the Readout Supervisor can emulate the occupancies of the L0 de-randomizer buffers in a finite state machine. The L0 de-randomizers (16 events deep) can overflow if the instantaneous rate of L0 accepts gets too high. The Readout Supervisor should internally monitor the occupancy by counting the L0 accepts and take into account the de-randomizer readout time. If there is risk of overflow, the RS throttle handler should throttle the L0 trigger by forcing L0 rejects. The finite state machine should have a programmable ‘readout time’ and ‘overflow/accept levels’ for easy modification. A throttle due to a L0 de-randomizer overflow should also increase an "internal L0 throttle” counter.

Although simulation has shown that the L1 buffers under normal conditions do not overflow, its buffer occupancy should be emulated in a finite state machine in the Readout Supervisor. An overflow situation can for instance be caused by a blockage in the L1 trigger system. Since the readout time of the L0 de-randomizers (see above) is fixed the occupancy can be known by counting the number of transmitted L1 decisions. The finite state machine should have programmable “overflow/accept levels”. Reaching the overflow Level should throttle the L0 trigger by forcing L0 rejects and increase an "internal L0 throttle" counter.

Another source of L0 throttles is the L1 trigger system, which can get overloaded by a too high L0 trigger rate. This throttle will be asserted by means of a hardware signal to the RS.

The L1 de-randomizer buffers in the Front-End can overflow if the L1 accept rate is too high or if the zero-suppression is slow. Monitoring of this is performed in the FE electronics. If an overflow is imminent in one of the de-randomizers, its overflow monitor will issue a L1 throttle to be received by the Readout Supervisor. The throttle will be transmitted as a hardware signal to the RS. The throttle delay is determined by cable lengths and internal delays in the OR modules and the Throttle Switch. Summing these components up it amounts to about 1 μs. Thus a buffer margin of a few events is sufficient to account for the throttle delay.

Event congestion can also occur in the DAQ system in which case the DAQ will issue a L1 throttle that should be received by the Readout Supervisor. This throttle could either be fed back via the ECS, if the origin is in the Subfarm Controllers, or directly to the Readout Supervisor as a hardware signal, if the origin is in the Front-End Multiplexing or the Readout Units.

The L0 and L1 hardware throttles, originating from the various sources, will be OR’ed before the Readout Supervisor, which will be equipped with one L0 and one L1 throttle input on the front-panel. A counter associated with each of the inputs should count the occurrences of the throttles. During throttling, the Readout Supervisor should reject the triggers corresponding to the level of the throttle until the throttle is de-asserted. The RS should have a timer for each throttle level, which counts the elapsed time during throttling. Based on the timers, there should also be a time-out mechanism associated with each throttle level to warn about blockages. The ECS interface should raise an alarm with the throttle level on time-outs.

For monitoring and debugging, the Readout Supervisor should have history buffers for the different throttles, memorizing throttle time stamps (bunch cycle) and throttle lengths.

L0 and L1 throttles can also be issued by the ECS. The ECS interface will receive the throttles and should be capable of generating a direct throttle signal on both levels in the Readout Supervisor. Separate counters should count the occurrences.

During all types of throttles, the RS should count the number of bunch crossings that are lost and the number of L0 and L1 trigger accepts that are converted to rejects.

Although the Front-End electronics are required to cope with consecutive L0 trigger accepts, the Readout Supervisor should contain an implementation to force bunch gaps of programmable length, should this be needed.

2.7 Bunch counter reset (BCR) and event counter reset (ECR)

The Bunch Counter Resets need to be distributed to the Front-End with highest priority to control the synchronization of the TTCrx's (BCR broadcast is defined by XXXXXX01). The Readout Supervisor should therefore have a special state machine to trigger the generation of BCR broadcasts, which is synchronized to the LHC turn signal (PCR). In order to give the transmission of the BCRs the highest priority, the BCR state machine should inhibit the generation of L1 broadcasts 17 bunch cycles before the proper time to start the transmission of the reset. The exact bunch cycle on which to initiate the BCR broadcast has to be compatible with the generation time of the BCR command (17 cycles). The offset between the bunch crossing that marks a turn at the LHCb interaction point and the PCR turn signal is compensated for in the LHC machine interface (TTCmi).

In order to ensure full integrity to the transmission of the BCRs and to minimize conflicts between other commands (see below), the time structure of all state machines together will be verified before down-load. In case there are occasional clashes between commands, the first-come-first-served principle rules and all other commands are postponed until the next turn. The verification thus only ensures that each command is transmitted on the average with the desired rate, taking into account generation time, transmission time, reception time, and execution time in the FE.

In practice, this means that all commands from the point of view of the Readout Supervisor have first priority and that L1 decision broadcasts have subordinate priority. This implies the L1 decisions can be delayed up to a maximum of 34 cycles for each command. The maximum average L0 accept rate that the system should handle has been specified to 1.11 MHz, i.e. an average L1 decision rate of one per 36 cycles. Reading out an event from the L1 buffer takes 34 cycles, which means that the L1 decisions can be transmitted by the Readout Supervisor with a spacing of minimum 34 cycles. Consequently, even running at the maximum average rate of 1.11 MHz, there is time to transmit up to four commands plus the BCR per turn without causing non-recoverable delays, which would otherwise lead to throttling. This has been verified in simulation[**11**].

A policy for resetting the event counters (L0 event-ID) has yet to be defined (ECR broadcast is defined by XXXXXX10). However, logically, the best is to reset the event counters simultaneously with the bunch counters, i.e. the BCR state machine will also transmit the ECRs at regular intervals.

In addition to the Bunch Counters and the Event Counters (L0 Event ID) there are also the L1 Event ID counters, counting the events that pass the L1 trigger. These counters should also be reset globally by the Readout Supervisor. Generally these counters will only be reset once per “run”, that is, on command via the ECS interface just before starting the readout (triggers enabled by the ECS interface), and will then count until the readout is stopped (triggers disabled by the ECS interface). The command to reset the L1 Event ID counters is specified in Table 2.

2.8 Front-End reset generation

Loss of synchronisation can occur at many different places in the readout, particularly as some of the electronics (FE) will receive high doses of radiation. The solution is to re-synchronize the readout by resetting the pointers for the pipelines and the de-randomizing buffers. De-synchronisation will thus inevitably lead to loss of data wherefore a carefully tuned reset strategy must be adopted. The implementation has to be flexible as the time between failures for the different components will not be known before the start of the experiment and as the error rates will most probably increase with time. In addition, as the error rates are most likely not the same for the different components it is interesting to define different levels of resets using encoded commands.

The Readout Supervisor should therefore contain a number of state machines with programmable time intervals and commands. In the case of periodic reset schemes, the state machines should loop with a cycle time tuned to the mean failure time, disable the trigger and issue a reset command with the appropriate Level, and after a given time re-enable the trigger. The delay corresponds to the reset execution time. Whereas a reset of the L0 electronics involves disabling only the L0 trigger, a reset of the L1 electronics involves disabling both the L0 and the L1. For the L0 electronics reset, the Readout Supervisor should disable the L0 trigger sufficiently long in advance (16 buffer slots * 36 cycles readout = 576 cycles) in order to empty the L0 de-randomizer before the actual reset.

If appropriate to the level, the de-randomizer, pipelines, counters, and FE buffer emulator registers in the Readout Supervisor should also be reset.

In order to minimize loss of data, the resets should be synchronized to the long LHC bunch gaps where collisions are guaranteed not to occur.

The different levels of resets are specified in the following format:

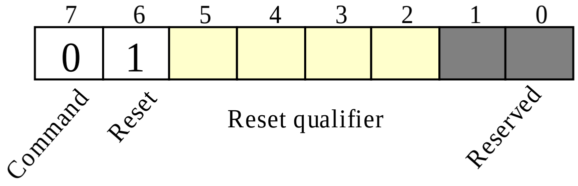


Figure 4: Format of the reset word.

The 4-bit reset qualifier allows specifying the type of reset as proposed in Table 2[9].

Table 2: Reset qualifier bits.

Bits <5:2>	Interpretation
Bit 2	Reset L0 electronics
Bit 3	Reset L1 electronics
Bit <4>	Reset L1 Event ID counters
Bit <5>	Reserve

As mentioned in Section 2.3 and 2.4, the ECS interface will raise an alarm in case of L0 or L1 de-synchronization so that, if desired, a reset command can be injected by the ECS. For this purpose the Readout Supervisor should have a special idle "single reset" sequence, which in phase with the LHC orbit cycle can transmit a single reset command at a specific time. Writing into a register should activate the sequence.

2.9 Online detector calibration/measurements

To calibrate and to measure the noise, thresholds and pedestals online while recording physics data, the Readout Supervisor should have several ways to auto-generate triggers (force L0 and L1 triggers).

Random triggers:
The Readout Supervisor should have a pre-scalable random trigger. The type of random trigger (true bunch crossing, single bunch crossing, empty bunch crossing) can be determined from the information about the bunch crossing stored in the RS data block,

Pre-defined intervals:
The Readout Supervisor should have several programmable state machines to inject L0 accepts at regular intervals (every *n* turns at a specifiable bunch crossing). L1 accepts are forced by marking the triggers as forced in the L0 accept FIFO. The state machines should be synchronized to the LHC orbit clock in order to time the L0 accepts to given LHC bunch cycles.

Consecutive bunch sampling:
In order to calibrate the timing of the Front-End electronics, the trigger system and the Readout Supervisor to account for time-of-flights and signal delays, the Readout Supervisor should have mechanisms that allow sampling a set of consecutive bunch crossings (max 16).

One method is to sample a set of consecutive bunch crossings (max 16) that is centred on a “good” bunch crossing. A “good” crossing is meant to be evaluated and signalled by the L0 decision unit eight crossings in advance. The Readout Supervisor can thus start forcing L0 and L1 trigger decisions before and continue after the “good” crossing, mark them as timing alignment triggers in the RS data block and transmit the special L1 trigger qualifier with the L1 trigger decisions. With the LHC accelerator running in a special “single bunch mode”, this is the ideal calibration method.

A second method, perhaps more suited for monitoring the timing of the Front-End, is by periodically sample unconditionally *n* consecutive crossings over the end of a bunch train to detect the tail bunch, when LHC is running in the standard bunch structure mode. For this purpose the RS should have a state machine that is in phase with the LHC orbit signal, and which can at a programmable time auto-generate a set of L0 accepts and mark them as forced. The L1 trigger qualifier should specify that the triggers are timing triggers, and this information should also be stored in the RS data block.

In both methods, by histogramming the bunch profile either in the Front-End or after the DAQ system, the timing can de determined.

After calibration pulse at regular intervals:
In order to calibrate online on a test pulse, the Readout Supervisor should have programmable state machines which can, at pre-defined intervals, inject a control command to the Front-End to generate a calibration pulse and after appropriate timing force a L0 accept. The L1 trigger is again accepted by marking the trigger as forced in the L0 accept FIFO. The state machine should be synchronized to the LHC orbit clock to ensure that the calibration pulse is fired during an LHC bunch gap. If several detectors want to use the same command to fire their calibration system, this calls for a common latency between sending the command and generating the L0 accept. This might thus involve introducing delays in the FE for the calibration systems with faster response. Another possibility is to react on a common pulse command, but to generate several L0 accepts at different times corresponding to the different calibration systems. The side effect is that everyone will sample several calibration events out of which only one is good for each detector. A third method is to implement several state machines with different calibration pulse commands. Each detector should then be set up to respond to a certain calibration command. The different state machines are then programmed with the appropriate delays between command and generating the L0 accept. The Readout Supervisor should have support for all these different options.

The command to fire the calibration pulse is to be encoded in a 4-bit command qualifier. The commands are identified by the following bit pattern:

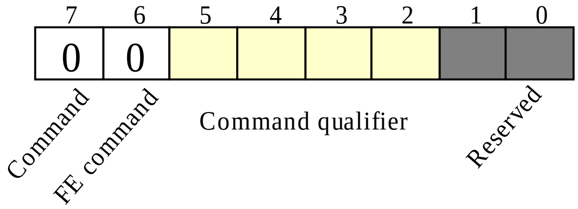


Figure 5: Format of the command word.

The command qualifier to fire a calibration pulse is foreseen to be 01XX, where the two bits XX are reserved[**9**]. The reserve bits could be used to build several pulse commands. The corresponding L1 trigger qualifier should specify that the trigger is associated with a calibration pulse. This information is also to be stored in the “Trigger type” word in the RS data block.

Each calibration state machine should have an associated counter, which counts the total number of self-triggers generated. A list of the state machines currently foreseen can be found in Table 3.

2.10 Stand-alone detector calibration/measurement run

Most typically a detector will carry out calibration and measurement stand-alone, out of physics data-taking, using a separate Readout Supervisor for timing and triggering. The Readout Supervisor should in this mode provide the same auto-triggering, control and pulse injection possibility as described in Section 2.9.

2.11 Test/debug run

For debugging and test purposes the Readout Supervisor should be able to auto-trigger on two pre-scalable random triggers. The L0 random trigger should be able to maintain an average Poisson-distributed[6] L0 accept rate of up to 1.5 MHz. The L1 random trigger should be able to accept L1 triggers at a rate of up to ~200kHz, also Poisson-distributed. The latency of the L1 trigger decisions should be emulated by taking the first trigger out of the L0 accept FIFO after a programmable delay with respect to the time when it was stored in the FIFO.

For special tests, the Readout Supervisor should have a mechanism to generate a programmable L0 trigger pattern where each accept is forced at L1. This is achieved by looping through a memory of 3564 bits (equal to an LHC turn) and generate L0 accepts according to the bit pattern.

A partition can be tested/debugged in stand-alone or in combination with any other set of partitions using a separate Readout Supervisor, or in global using the central Readout Supervisor.

The Readout Supervisor should have an internal oscillator (40.08 MHz) to replace the global LHC bunch clock. The oscillator should have a second pre-scaled output (1:3564) to emulate the LHC orbit signal. Switching between external and internal clock should be by means of programming via the ECS interface.

It should be possible to configure the Readout Supervisor to drive the readout until a programmable number of L1 triggers have been accepted.

2.12 Other commands to the Front-End

The command qualifier also allows transmitting other miscellaneous control commands, which will be received and decoded by the whole Front-End system.

The Readout Supervisor should allow control commands to be sent asynchronously at any time via the ECS interface. The sending of these commands has to be, however, compatible with the overall timing restrictions, like for instance, the timing of the bunch counter reset.

In order to direct commands to individual TTCrx's, one could envisage implementing the addressed command feature of the TTC system. However, these are 42-bit frames and would thus have a non-negligible influence on the bandwidth for L1 triggers and normal 16-bits commands. In addition, this would require assigning identifiers to all TTCrx's.

[6] It may be of interest to have control over the distribution. A suitable design is under investigation.

2.13 Counters

The Readout Supervisor should count several quantities in order to maintain an accounting. A subset of the counters will make up the RS block in the event data. The counters will also be used to check synchronisation with the other components in the readout system and for monitoring the data-taking performance such as dead-time, error rates, overflows, etc. The counters are generally reset via the ECS interface when the triggers are enabled. Some counters are also reset automatically when the FE electronics are reset. Here is a list of the general counters. A more detailed list can be found in Section 5.

- LHC bunch clock
- LHC orbit clock
- LHC bunch crossing identifier – Bunch ID
- L0 Event identifier counter – L0 Event ID (L0 trigger accepts)
- L1 Event identifier counter – L1 Event ID (L1 trigger accepts)
- Bunch ID error (L0 trigger de-synchronized)
- L0 Event ID error (L1 trigger de-synchronized)
- L0 accept FIFO occupancy
- L1 trigger de-randomizer occupancy
- L0 de-randomizer occupancy (emulated)
- L1 buffer occupancy (emulated)
- L0 de-randomizer overflow (internal L0 throttle 1)
- L1 buffer overflow (internal L0 throttle 2)
- Occurrence counter for the L0 throttle input
- Occurrence counter for the L1 throttle input
- Occurrence counter for the L0 ECS throttle
- Occurrence counter for the L1 ECS throttle
- Resets of Front-End (different levels)
- Bunch crossings lost during L0 throttles
- L0 accepts converted to rejects during L0 throttles
- L1 accepts converted to rejects during L1 throttles
- Bunch crossings lost during resets
- L0 accepts lost during resets
- L1 accepts lost during resets
- Forced L0 accepts by L0 trigger
- Forced L0 accepts by sequencer
- Forced L0 accepts by random trigger
- Forced L1 accepts by L1 trigger
- Forced L1 accepts by L0 (FIFO force bit set)
- Counters on self-triggers generated by state machines in the sequencer

All the counters should be readable via the ECS interface at any time. The Bunch ID and the event-ID counters should be available for real-time coherence-check of L0 and L1 and to the RS data pipeline.

2.14 The ECS interface

The ECS interface should be the standard LHCb controls interface. It should receive control commands from the Experimental Control System and handle:

- starting/stopping the readout
- programming and configuring the Readout Supervisor,
- clearing the counters and appropriate control and status register,
- reset internally the various components of the Readout Supervisor,
- sending control commands on demand,
- and to retrieve monitoring information to be passed back to the ECS.

To retrieve an instantaneous and consistent set of monitoring data, it is desirable that all counters and control and status registers are sampled simultaneously in local buffers before being read.

It should be possible to disable the bus between the ECS interface and the Readout Supervisor in order to allow rebooting the ECS interface without interfering with the RS operation.

3 Modular specifications

3.1 Introduction

A simplified block diagram of the Readout Supervisor is shown in Figure 6.

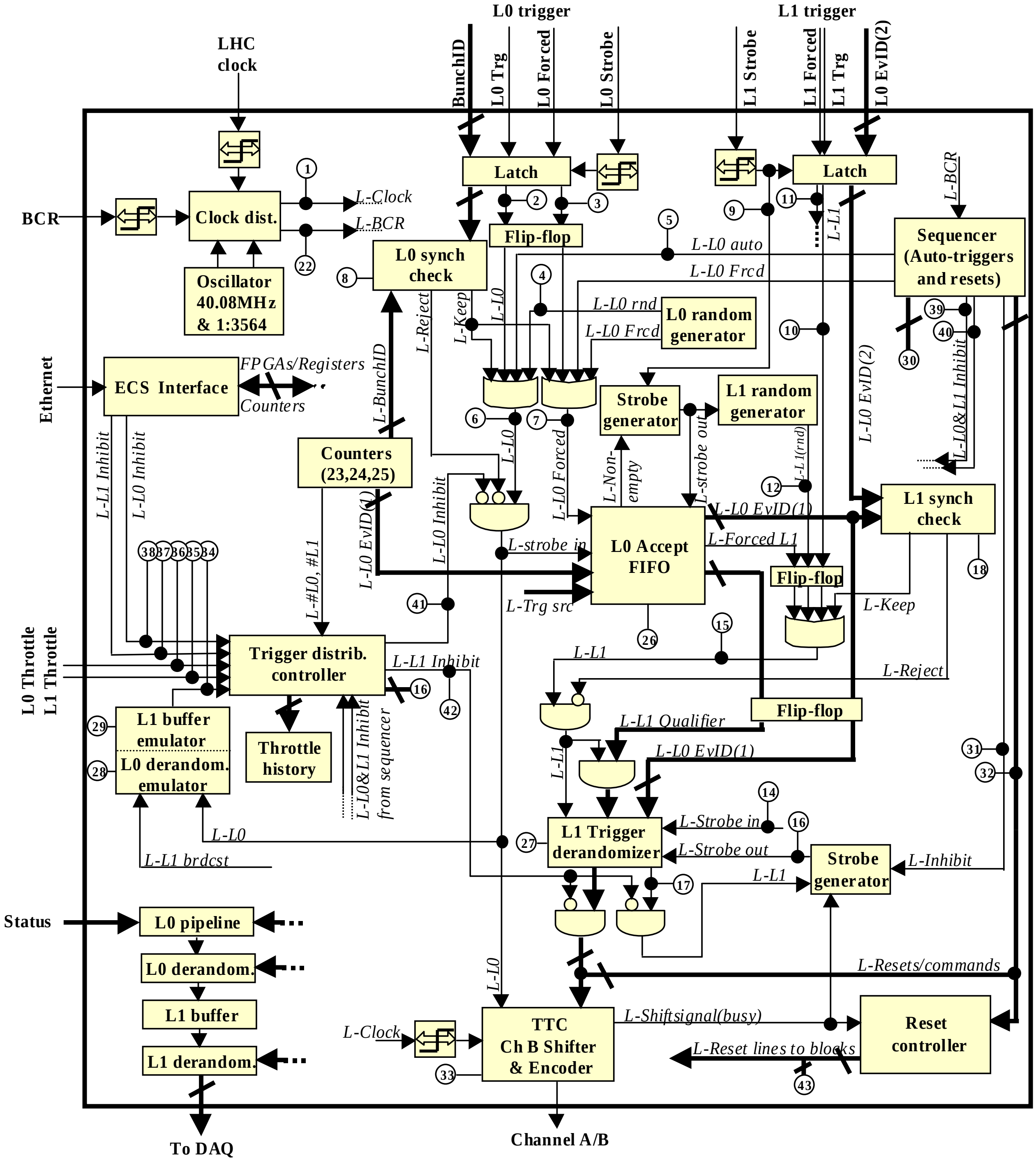


Figure 6: Block diagram of the Readout Supervisor. The numbers correspond to the counters listed in Section 5: Table 10 and Table 12.

3.2 L0 trigger distribution path

The L0 trigger distribution path is shown in Figure 7 and the timing diagram in Figure 8. The path should be able to receive and output consecutive triggers every cycle (25ns) and have no more than four cycles of latency (100ns) between the end of the pipeline(see below) and output[**6**]. Currently the design (Figure 7 and 8) entails two cycles of latency (50ns). It should be possible to disable the L0 trigger input via the ECS interface by ignoring the external L0 strobes. As a consequence the L0 synchronization errors must be suppressed and the strobe detection mechanism must be disabled.

3.2.1 L0 trigger pipeline *(Not drawn in Figures 6 and 7)*

In order to be able to adjust the total latency of the L0 trigger path to 160 cycles, the Readout Supervisor should have a pipeline at the input of the L0 trigger. The pipeline should be 16 bits wide to store the entire L0 trigger word for each trigger. The depth[7] of the pipeline should be programmable, that is, it should be possible to tap the trigger words anywhere along the pipeline. The data in the pipeline should shift one pipeline step every cycle.

In order to generate the timing alignment triggers, it should be possible to spy on the timing trigger bit anywhere along the pipeline and, if set, inject a programmable number of L0 accepts with the force bit set.

3.2.2 L0 synchronization check

The L0 synchronization check should compare the Bunch ID received from the L0 trigger system with the expected Bunch ID. However, the RS Bunch ID counter is reset with the LHC turn signal and keeps in step and phase with the current bunch crossings. The L0 synchronization check should therefore use a counter that has a programmable fix offset with respect to the RS Bunch ID counter to correct for the expected latency of the L0 trigger. The alternative is to use a Bunch ID stored in the RS L0 pipeline for the corresponding trigger.

The L0 synchronization check should have a programmable option to generate a L0 accept signal automatically when there is interest in keeping all de-synchronized triggers, including those that are not accepted by the L0 trigger system. A second option should imply rejecting all de-synchronized triggers, including those that are accepted by the L0 trigger system. A third option should be to ignore the result of the synchronization check and accept/reject the events only according to the L0 trigger system. However, information about synchronization error should always be stored in the RS data block.

3.2.3 L0 Accept FIFO

The L0 accept FIFO should store the L0 event-ID as counted internally by the Readout Supervisor and the force bit for every L0 accept. In addition it should compile and store information about the source of the trigger in the format of the encoded 3-bit L1 qualifier. The suitable width of the FIFO is thus 24 bits. The information is to be synchronously latched in by the L0 signal itself and the

[7] The maximum depth is not yet clear. The first RS prototype will have a 16 stage pipeline.

information is to be latched out by the L1 trigger strobe. In case the L1 trigger input is disabled, an internal L1 strobe generator should take over (see Figure 12). If enabled the strobe generator should be triggered by the non-empty signal in the FIFO and with a programmable period, latch the output of the FIFO.

The FIFO should have an occupancy counter. The depth of the FIFO is to be studied in simulation. Although the FIFO should be dimensioned to never overflow before the L0 de-randomizers or the L1 buffers, the FIFO should throttle the L0 trigger if the occupancy reaches the "FIFO almost full" level (throttle not drawn in Figure 6).

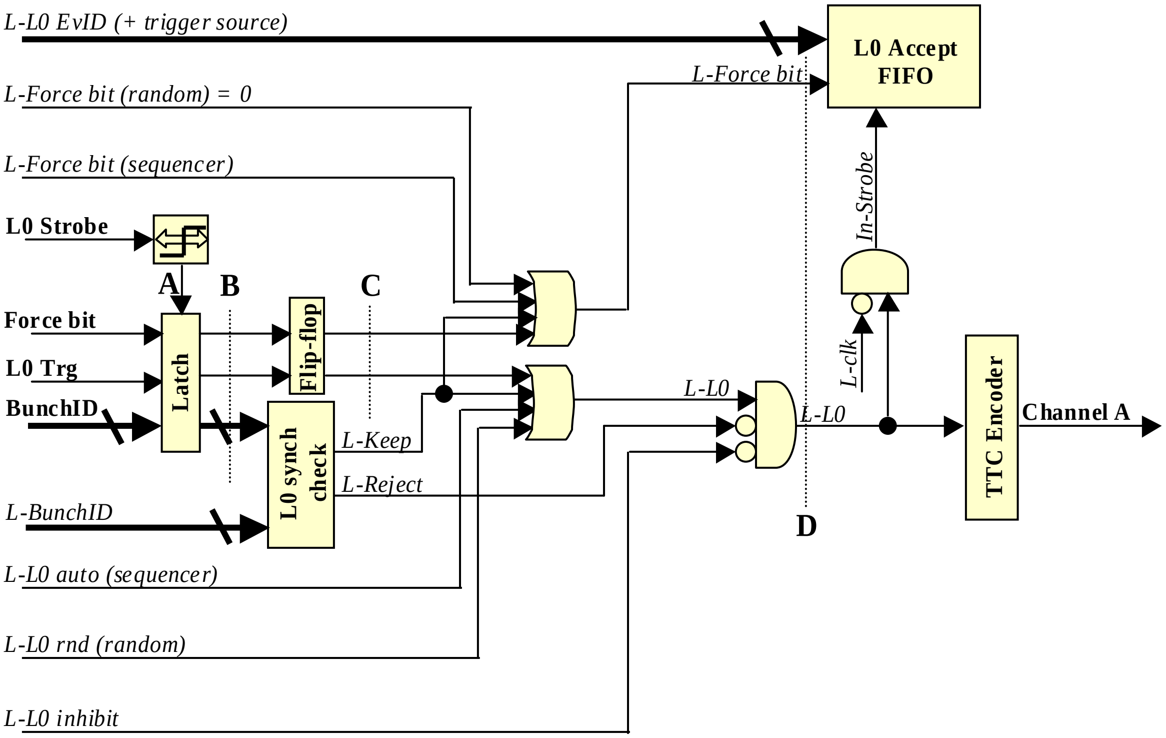


Figure 7: The L0 path.

3.3 L1 trigger distribution path

The L1 trigger distribution path is shown in Figure 9 and the timing diagram in Figure 10 and 11. The path should be able to receive and output consecutive triggers every cycle (25ns). Currently the design (Figure 9, 10 and 11) entails two cycles of latency up to the L1 trigger derandomizer. The serializer and the encoder will add another 17 cycles of latency.

It should be possible to disable the L1 trigger input via the ECS interface by ignoring the external L1 strobes. As a consequence the L1 synchronization check must be disabled and the internal L1 strobe generator must be enabled (see Figure 12).

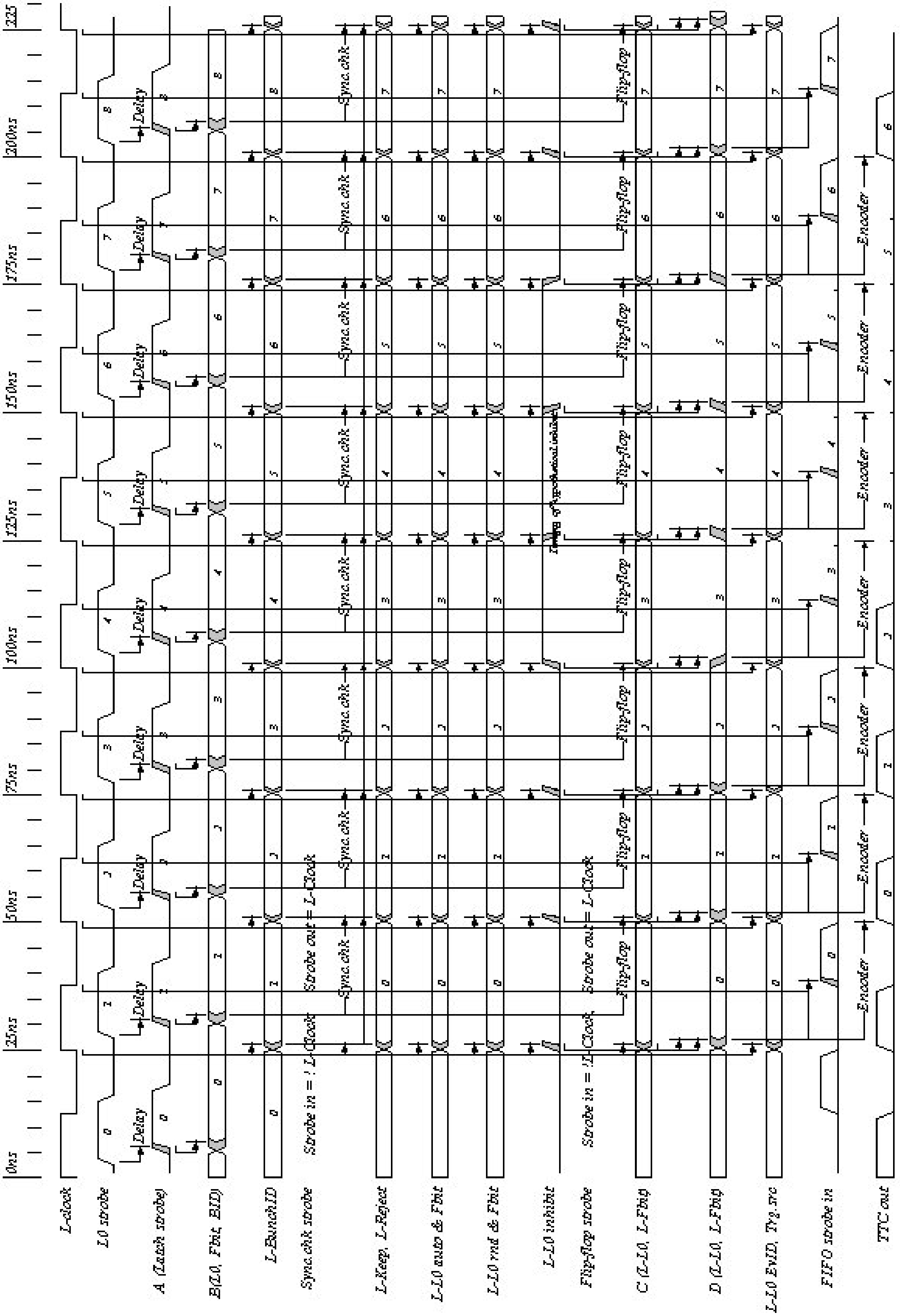


Figure 8: The timing diagram om the L0 path. A fictitious pattern of L0 accepts is drawn. The propagation delays have all been assumed to be [0.0ns, 0.2ns] in order to show where delays are present.

3.3.1 L1 synchronization check

The L1 synchronization check should compare the L0 event-ID received from the L1 trigger system with the first L0 event-ID pending in the L0 accept FIFO.

The L1 synchronization check should have a programmable option to generate a L1 accept signal when there is interest in keeping all de-synchronized triggers, including those that are not accepted by the L1 trigger system. A second option should imply rejecting all de-synchronized triggers, including those that are accepted by the L1 trigger system. A third option should be to ignore the result of the synchronization check and accept/reject the events only according to the L1 trigger system. However, information about synchronization error should always be stored in the RS data block.

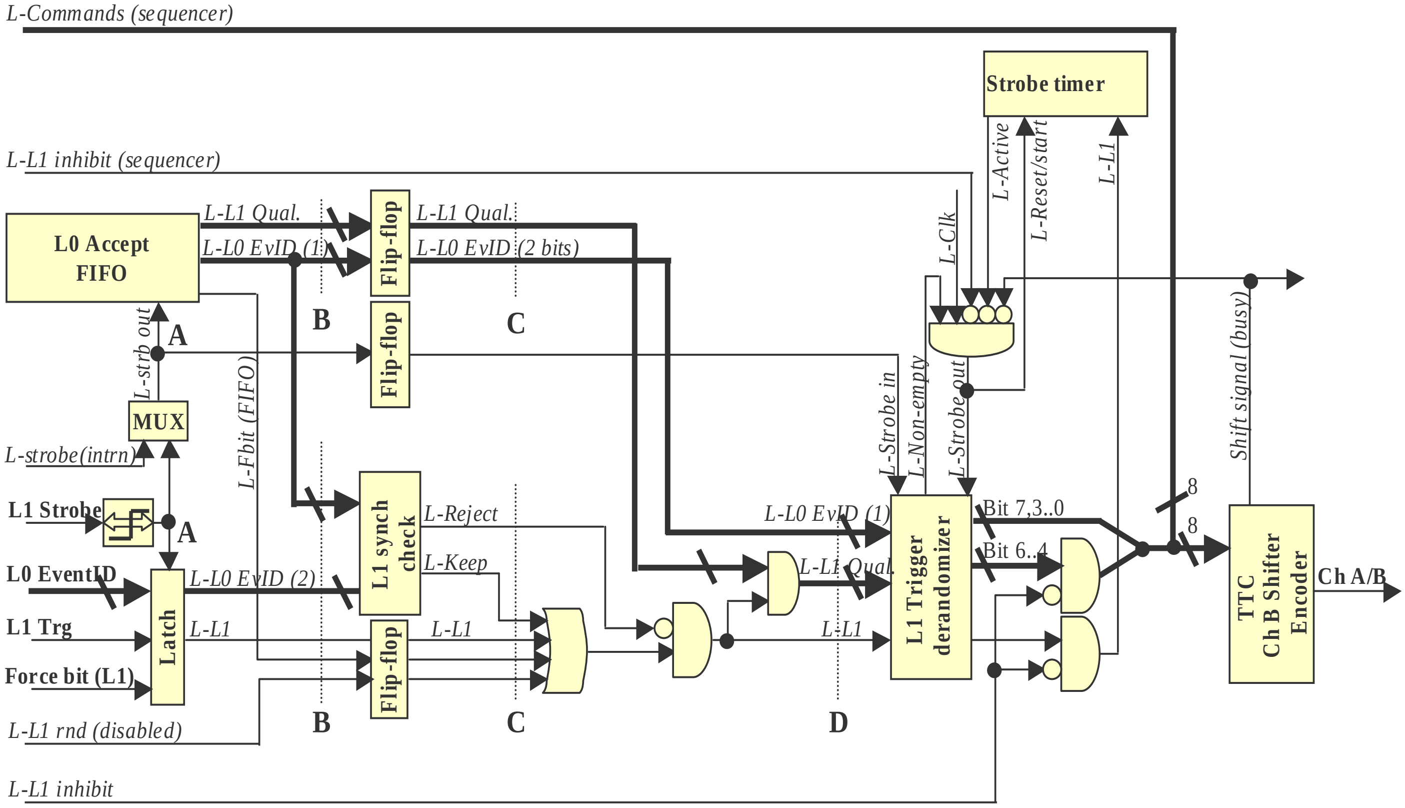


Figure 9: The L1 path.

3.3.2 L1 trigger de-randomizer

The L1 triggers should be stored in a de-randomizer to regulate the rate. The de-randomizer should receive the L0 Event ID and the L1 trigger qualifier from the L0 accept FIFO and store them according to the 6-bit broadcast format defined in Section 2.4. The L1 trigger decision should also be stored. The width of the de-randomizer should thus be minimum 8 bits. The information is to be latched in by the L1 strobe delayed by one cycle. A L1 reject should clear completely the L1 decision qualifier.

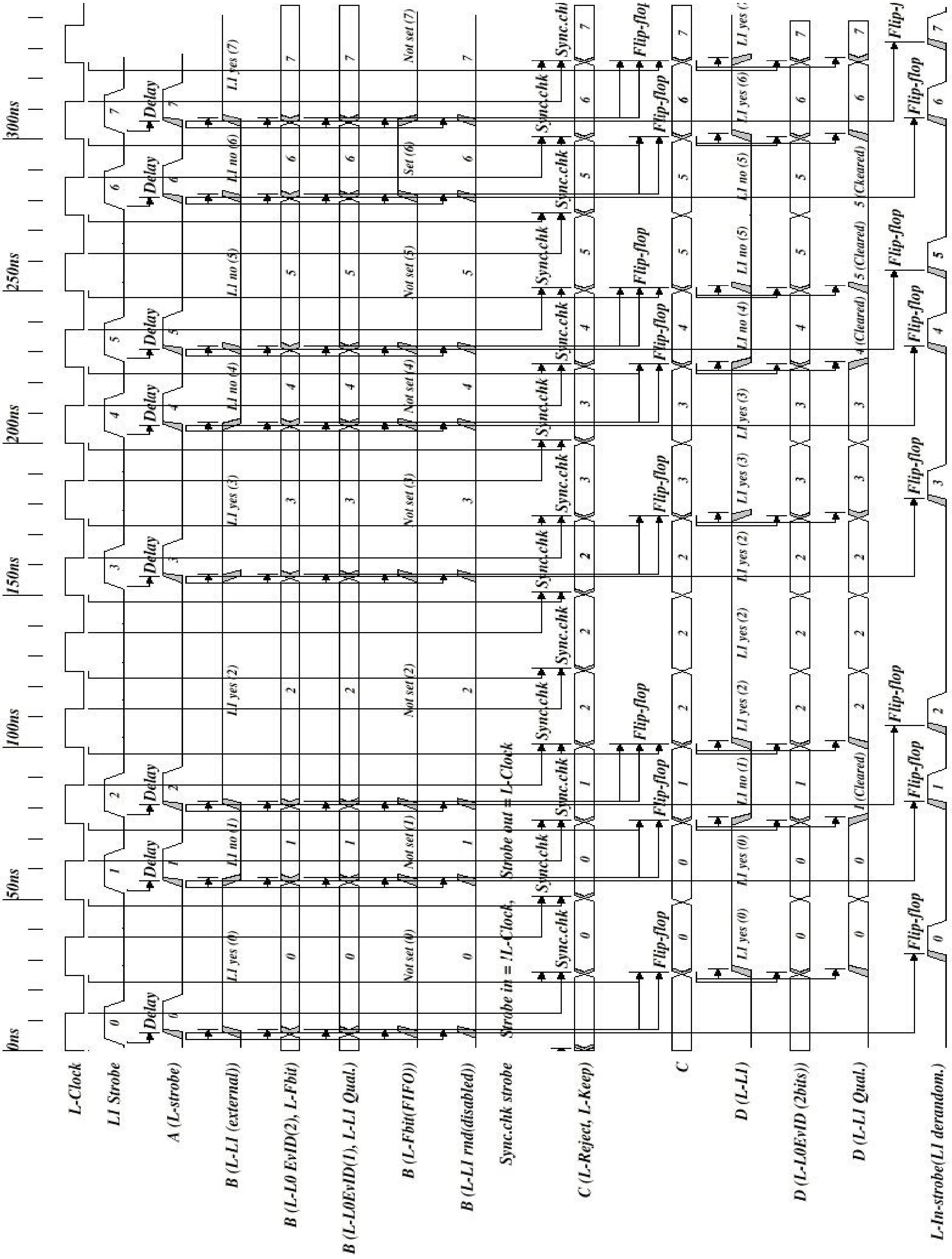


Figure 10: The timing diagram of the L1 path until the L1 trigger derandomizer. A fictitious pattern of L1 accepts is drawn. The propagation delays have all been assumed to be [0.0ns, 0.2ns] in order to show where delays are present and how they affect the timing.

The L1 de-randomizer should have an occupancy counter as well as three status bits associated with 'half full', 'almost full' and 'full'. The depth of the L1 trigger de-randomizer is a case for simulation. In order to be on the safe side it is reasonable to make the derandomizer about twice as big as the L1 buffer.

The transmission of a L1 trigger broadcast is initiated by strobing out the first pending event in the de-randomizer. The 6-bit L1 trigger word is consequently loaded into the TTC serializer. For L1 trigger broadcasts the two lowest bits in the TTC 8-bit format should be zero. The de-randomizer output strobe is a combination of several components in order to control the rate of the L1 trigger broadcasts (fig 9):

- The non-empty signal from the de-randomizer.
- The inhibit signal from the sequencer ensuring that the TTC serializer/encoder is free at the appropriate time to send a command (see 2.7or 3.5.1). The inhibit signal should be issued 17 cycles (=time needed by the TTC serializer/encoder to transmit a broadcast) before the appropriate time to start the transmission of the command. A L1 trigger may hence be delayed maximum 34 cycles.
- A strobe timer which controls the periodicity of the strobe. The timer should have two programmable delays (one after L1 trigger accept and one after L1 trigger reject) and be reset/started by the strobe itself. Hence the timer's power up state must be to allow strobes[8]. The timer should receive the L1 trigger decision to start the timer with the right delay.
- The serializer shift signal as veto to ensure than no strobe is generated while the serializer is busy.

The output strobe should also be synchronous with the clock.

The set of components listed above guarantees higher priority for commands. In case a L1 trigger is pending due to a command being sent (e.g. BCR), the de-randomizer strobe will come as soon as the inhibit signal from the sequencer is de-asserted.

A counter should keep a count of the number of L1 triggers broadcast.

[8] The first strobe will generally be generated as soon as the non-empty signal is present.

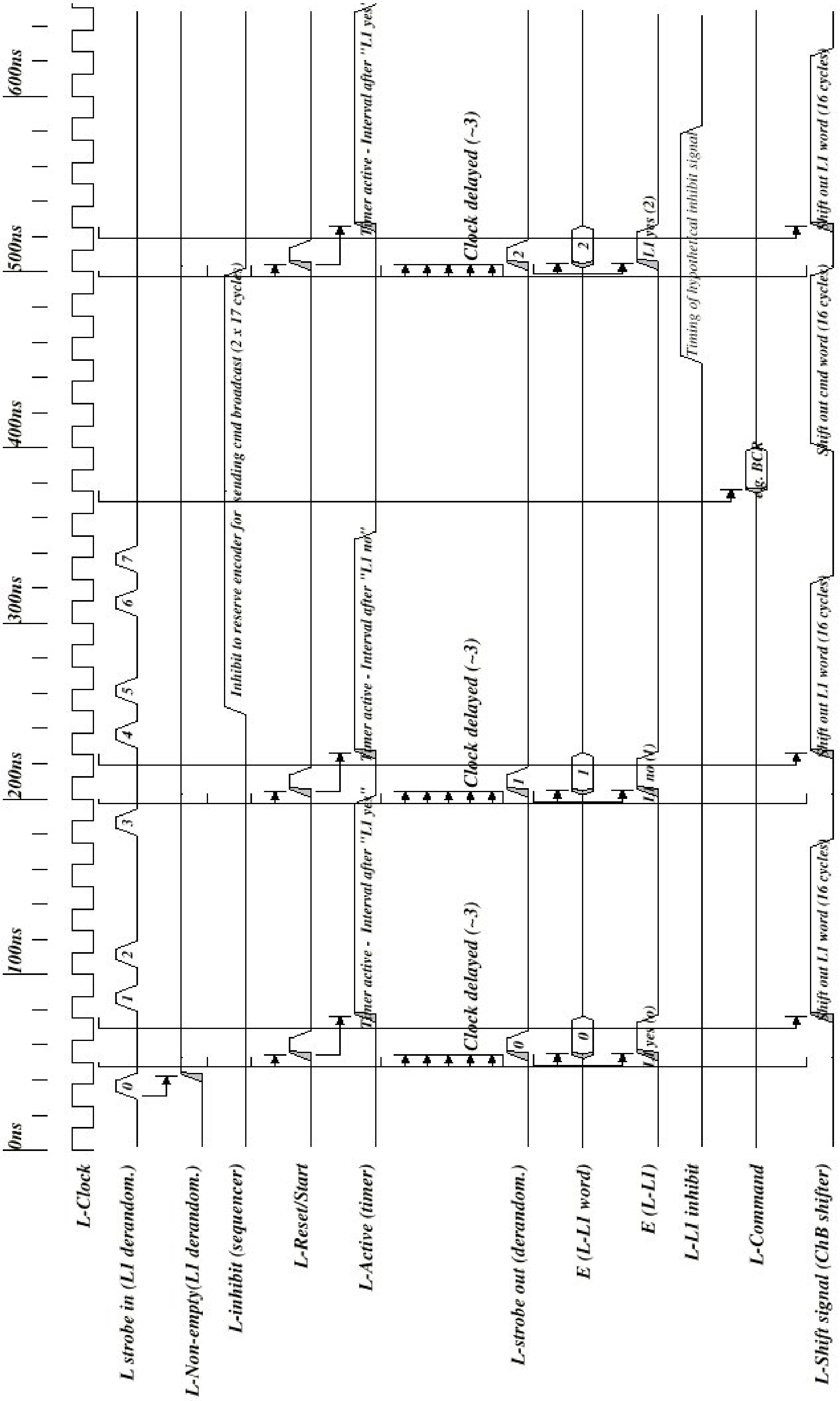

0ns
100ns
200ns
300ns
400ns
500ns
600ns
L-Clock
L strobe in (L1 derandom.)
L-Non-empty(L1 derandom.)
L-inhibit (sequencer)
L-Reset/Start
L-Active (timer)
L-strobe out (derandom.)
E (L-L1 word)
E (L-L1)
L-L1 inhibit
L-Command
L-Shift signal (ChB shifter)
0
1
2
3
4
5
6
7
Inhibit to reserve encoder for sending cmd broadcast (2 x 17 cycles)
Timer active - Interval after "L1 yes"
Timer active - Interval after "L1 no"
Timer active - Interval after "L1 yes"
Clock delayed (~3)
Clock delayed (~3)
Clock delayed (~3)
0
1
2
0
1
2
L1 yes (0)
L1 no (1)
L1 yes (2)
Timing of hypothetical inhibit signal
e.g. BCR
Shift out L1 word (16 cycles)
Shift out L1 word (16 cycles)
Shift out L1 word (16 cycles)
Shift out cmd word (16 cycles)
Shift out L1 word (16 cycles)

Figure 11: The timing diagram of the L1 trigger de-randomizer and the TTC serializer/encoder. The diagram follows that of Figure 10. The diagram also shows how a command is slotted in.

3.4 Random generators

The Readout Supervisor should have two random generators (L0 and L1) in order to run authentic tests on the systems, but also to trigger randomly during physics data taking. Figure 12 shows how the L0 path and the L1 path are connected and the implementation of the two levels of random generators. Figure 13 shows the timing diagram of Figure 12.

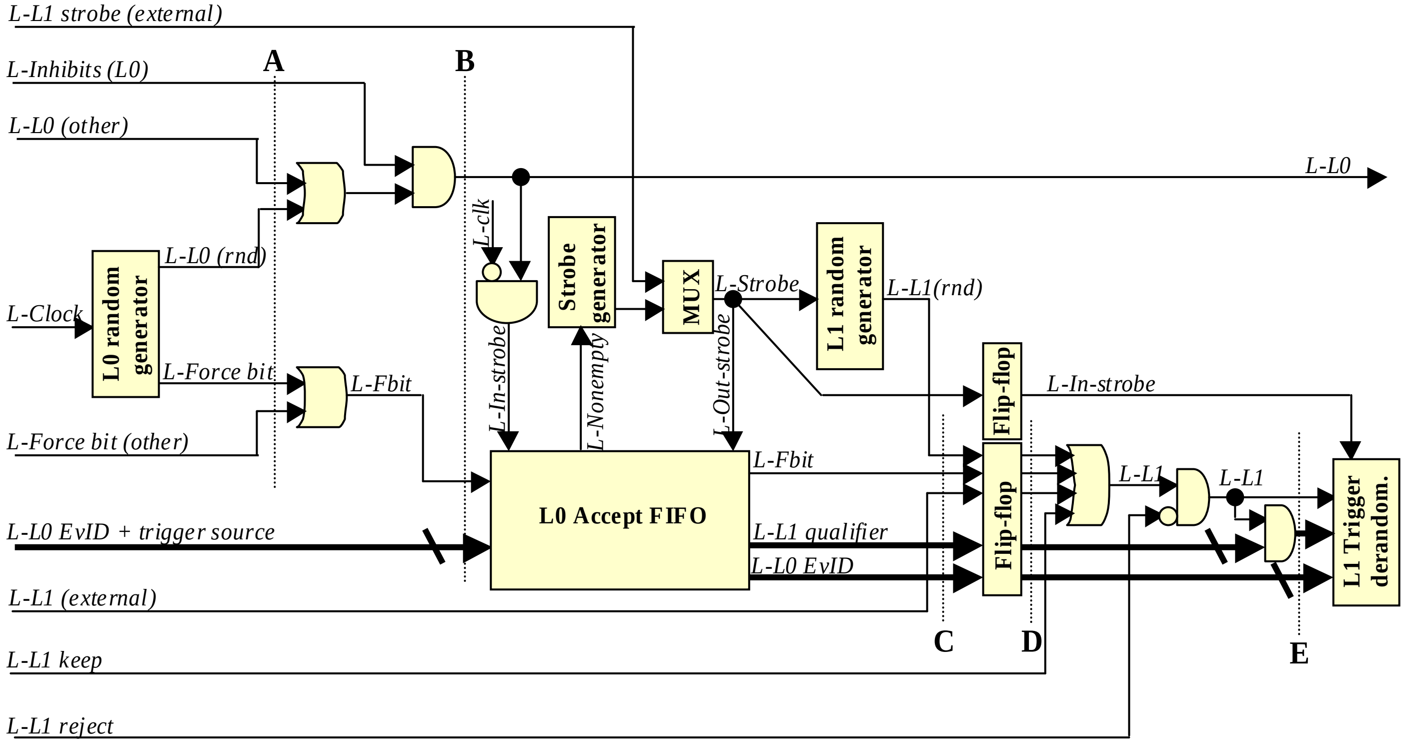


Figure 12: Diagram centered around the L0 Accept FIFO and showing the implementation of the two random generators.

3.4.1 L0 random trigger

The Readout Supervisor should have a pre-scalable L0 random generator providing a Poisson distributed L0 trigger with a maximum mean rate of 1.5 MHz[9].

To trigger randomly during physics data taking, only the L0 random trigger is to be used with the force bit, associated with each trigger, set and with very high pre-scaling.

In test running, the random generator force bit is typically permanently off. A subset of the L0 accepts is selected by the L1 random generator.

The pre-scaling and the value of the force bit are to be written into registers by the ECS interface. Depending on the design there should also be a register for downloading the random seeds.

It should also be possible to enable and disable the L0 random generator via the ECS interface.

[9] The minimum rate will depend on the design but the aim is to have a tunable rate rather than prescaling.

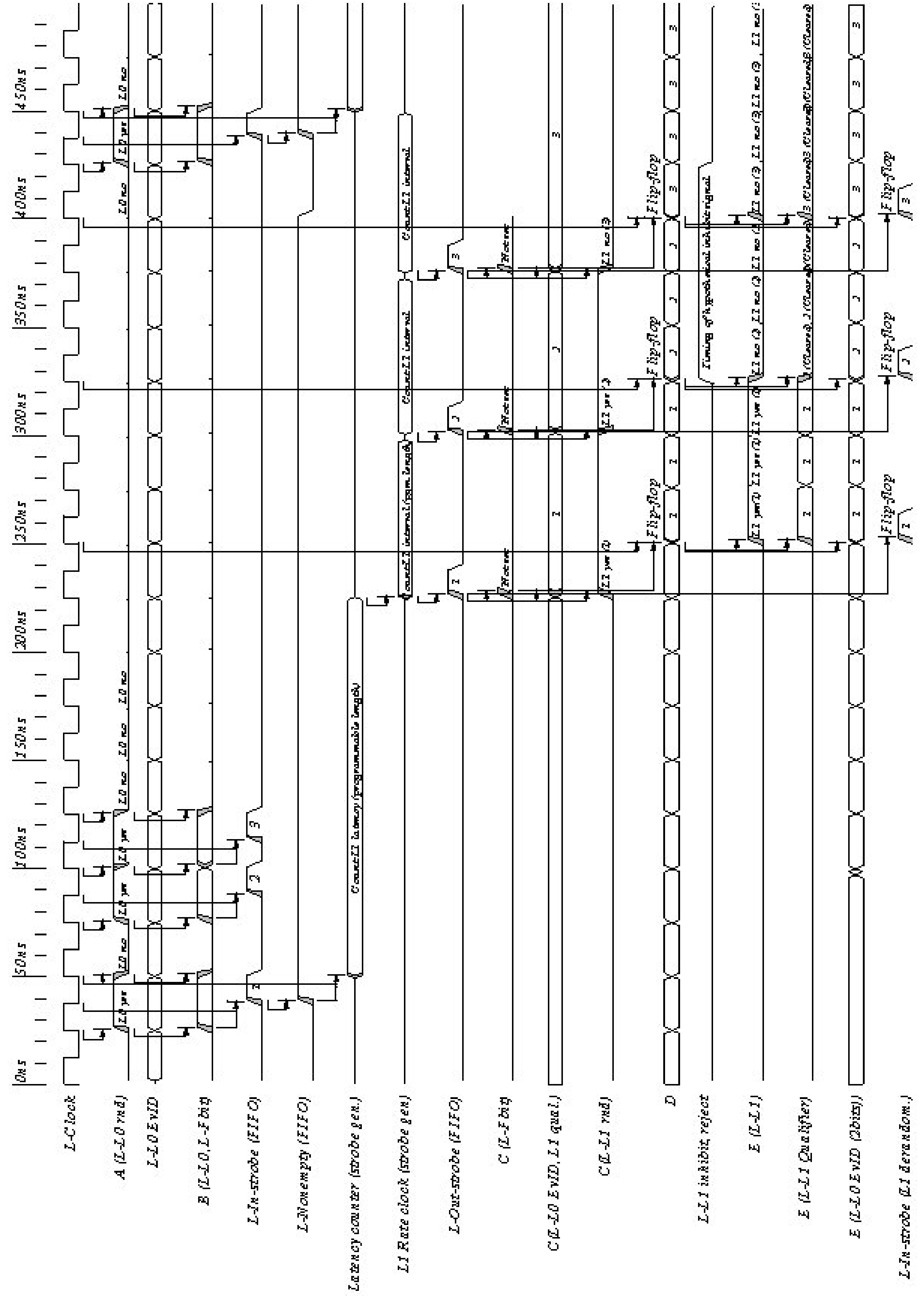


Figure 13: The timing diagram of Figure 12. The diagram shows a fictitious pattern of random L0 and L1 triggers. As before the propagation delays have been assumed to be [0.0ns, 0.2ns].

3.4.2 L1 random trigger

A second pre-scalable random generator should allow triggering on the accepted L0 events by injecting L1 accepts at a maximum mean rate of 200 kHz[9], also Poisson distributed.

The start of the L1 random trigger should be delayed with respect to the start of the L0 random generator in order to fake the average L1 latency and allow the various buffers to buffer up (see Figure 13). The random generator should subsequently be strobed at regular intervals by a timer in the strobe generator. When the L0 trigger is disabled, the L1 random trigger should continue until the L0 accept FIFO is empty, i.e. the FIFO non-empty signal is de-asserted.

The pre-scaling, the strobe interval and the fake latency should be loaded into registers by the ECS interface. Depending on the design there should also be a register for downloading the random seeds.

It should also be possible to enable and disable the L1 random generator via the ECS interface.

3.5 Trigger, reset and control command sequencer

The sequencer should contain a set of finite state machines that drives different types of auto-triggers, resets and commands. A ‘generic command sender’ FSM is used by all state machines that send TTC command broadcasts. As commands cannot be sent in parallel, the use of the ‘generic command sender” should be controlled by a ‘Busy’ bit. All state machines that send commands have a conditional transition with a ‘test-set’ mechanism to reserve the ‘generic command sender’. If the ‘command sender’ is found busy, the pending state machine should try again at the same bunch crossing in the next LHC turn.

From the sequencer it should be possible to directly inject a L0 trigger and force bit, inhibit L1 broadcasts, the L0 trigger and the L1 strobe, and write commands directly into the TTC serializer/decoder. Resetting the L0 FE electronics or the L1 FE electronics by writing a reset command to the serializer/encoder also involves resetting various internal components in the Readout Supervisor. However, a Reset Controller should handle the internal resets. The Reset Controller should receive the commands at the same time with the TTC serializer/encoder, decode them, and reset the appropriate blocks while the serializer/encoder is busy transmitting the reset command. The table below specifies the components to be reset.

The Sequencer should have an activity counter for each state machine that counts the number of transitions back to the initial state (S1).

Table 3 lists the different state machines, the number of programmable parameters(delays), the commands, and the internal actions. Actions that are performed by the Reset Controller are in italics. Figure 14 - 17 show all the state machine diagrams. The *CNTs* are clock counters that are preset in the actions triggered by the transitions. They drive the timing by counting down during the following state to trigger the next transition and action. *P#* are the programmable timing parameters. The last two “state machines” (17,18) scan in a loop through memories of 3564 bits each to generate a programmable L0 trigger pattern for special tests and to output the expected LHC crossing type.

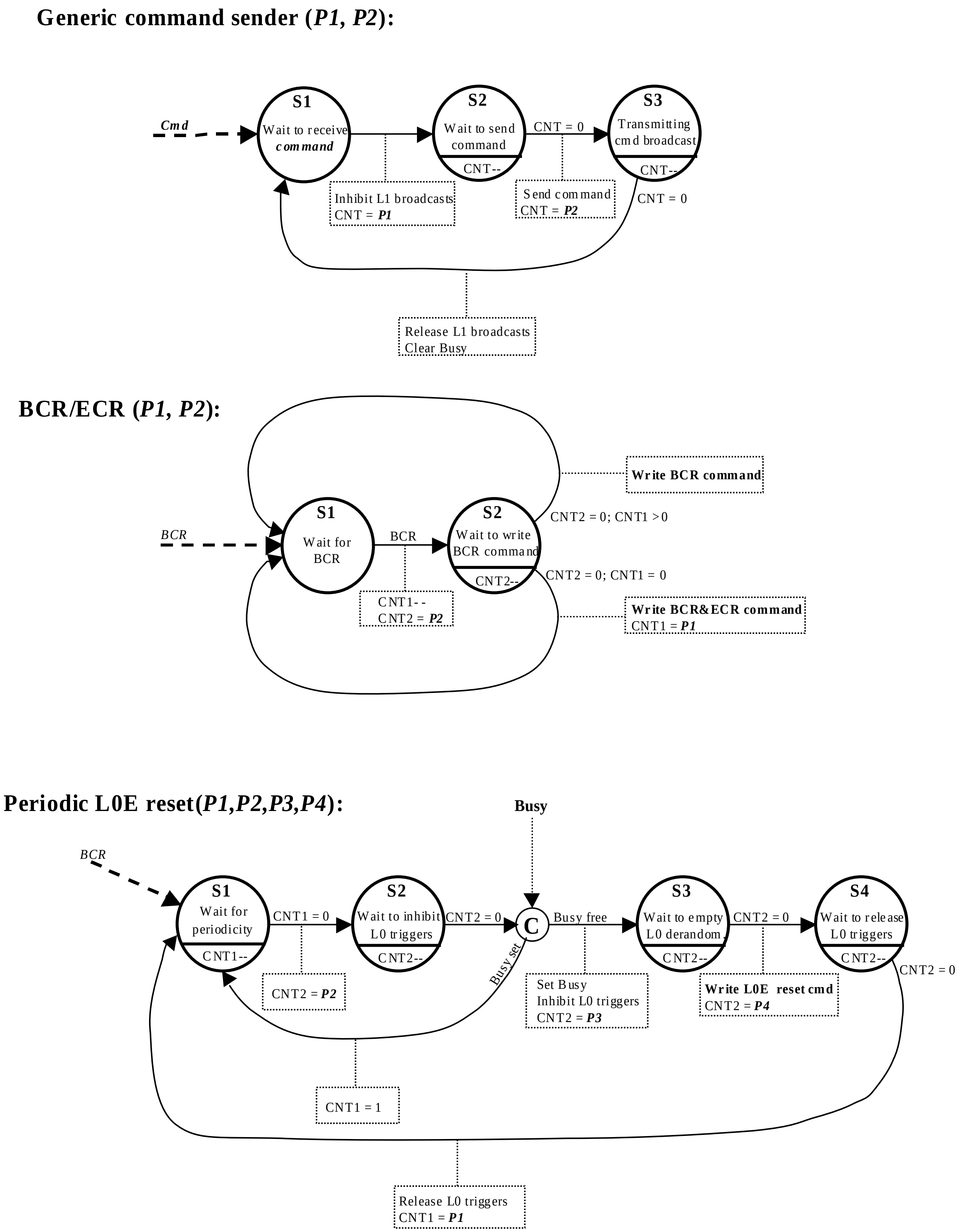
In order to avoid sending various broadcasts to the FE electronics while it is performing a L1 FE reset, it should be possible to disable any number of state machines from the L1 FE reset state machine. Despite this feature, the FE electronics should also be capable of ignoring incomming commands while performing a reset.

Table 3: List of currently foreseen finite state machines for sending resets, triggers and commands. The "#parameters" are the number of programmable delays for each FSM that determine the timing of the sequence.

#	Sequence (FSM)	#Params.	Commands	Internal actions
1	Generic command sender	2	Generic command	-Suspend/release L1 broadcasts -Clear Busy
2	Bunch/Event counter reset	2	BCR/ECR	*(-Reset Bunch ID counter in RS – performed directly by turn signal)* *-Reset L0 Event ID counter in RS*
3	Periodic L0 electronics reset	4	L0E reset	-Set Busy -Inhibit/release L0 triggers *-Reset L0 Event ID* *-Reset "RS L0 Front-End"* *-Reset L0 synch. check* *-Reset L0 de-randomizer emulator* *-Reset L0 Accept FIFO occupancy counter*
4	Periodic L1 + L0 electronics reset	3	L1E + L0E reset	-Set Busy -Inhibit/release L0 triggers -Disable/enable L1 trigger receive/send[10] -Disable/enable FSMs *-Reset L0 Event ID* *-Reset L1 Event ID* *-Reset L0 Accept FIFO* *-Reset L1 trigger de-randomizer* *-Reset L0 & L1 synch. checks* *-Reset L0 de-randomizer emulator* *-Reset L1 buffer emulator* *-Reset Throttle timeouts* *-Reset "L0 & L1 RS Front-End"* *-Reset L0 Accept FIFO occupancy counter* *-Reset L1 de-random. occupancy counter*
5	L0 electronics reset on demand	3 (+activate)	L0E reset	-Set Busy -Inhibit/release L0 triggers *-Reset L0 Event ID* *-Reset "RS L0 Front-End"* *-Reset L0 synch. check* *-Reset L0 de-randomizer emulator* *-Reset L0 Accept FIFO occupancy counter*
6	L1 + L0 electronics reset on demand	2 (+activate)	L1E + L0E reset	-Set Busy -Inhibit/release L0 triggers -Disable/enable L1 trigger receive/send -Disable/enable FSMs *-Reset L0 Event ID* *-Reset L1 Event ID* *-Reset L0 Accept FIFO* *-Reset L1 trigger de-randomizer* *-Reset L0 & L1 synch. checks* *-Reset L0 de-randomizer emulator* *-Reset L1 buffer emulator* *-Reset Throttle timeouts* *-Reset "L0 & L1 RS Front-End"* *-Reset L0 Accept FIFO occupancy counter* *-Reset L1 de-random occupancy counter*
7	Periodic trigger 1	2	-	-Inject L0 accept -Inject Force bit
8	Periodic trigger 2	2	-	-Inject L0 accept -Inject Force bit
9	Periodic trigger 3	2	-	-Inject L0 accept

[10] It is not sufficient to throttle the L1 trigger as this only converts them to rejects. The L1 trigger input strobe and the output strobe of the L1 trigger de-randomizer has to be disabled in order to stop receiving L1 triggers and to stop sending L1 trigger broadcasts.

				-Inject Force bit
10	Trigger on calibration pulse 1	3	Pulse	-Inject L0 accept -Inject Force bit
11	Trigger on calibration pulse 2	3	Pulse	-Inject L0 accept -Inject Force bit
12	Consecutive bunch sampling	3	-	-Inject L0 accept -Inject Force bit
13	N triggers on demand	2	-	-Inject L0 accept -Inject Force bit
14	Periodic command 1	2	Command	-Set Busy
15	Periodic command 2	2	Command	-Set Busy
16	Command on demand	1 (+command)	Command	-Set Busy
17	Trigger pattern (3564 long)	3564 bits	-	-Inject L0 accept -Inject Force bit
18	LHC bunch structure	3564 bits	-	-Output type of crossing

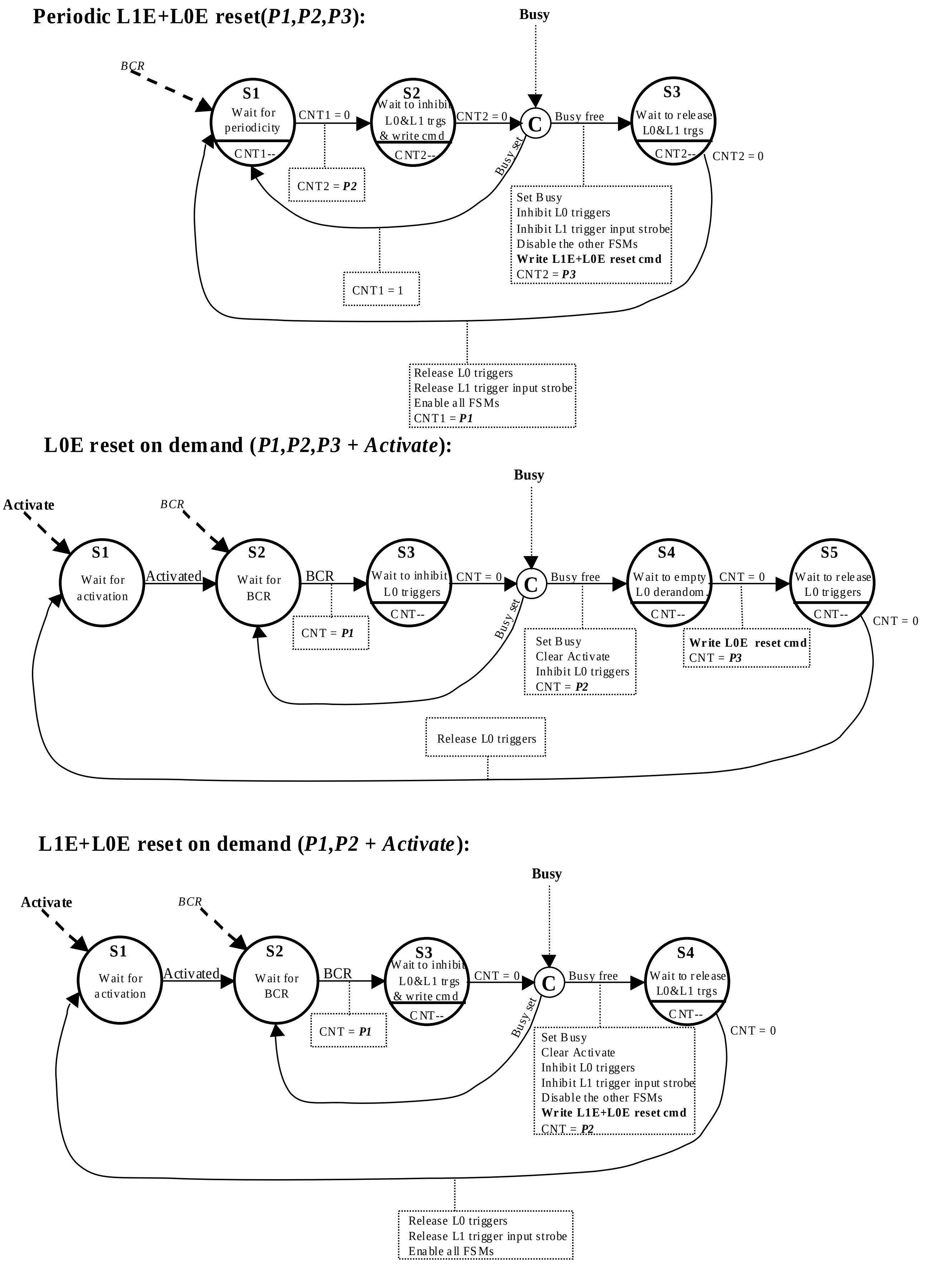


Generic command sender (*P1, P2*):

BCR/ECR (*P1, P2*):

Periodic L0E reset(*P1,P2,P3,P4*):

Figure 14: The finite state machines



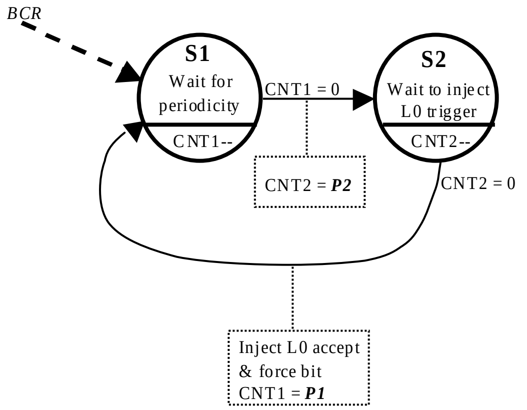
Periodic L1E+L0E reset(*P1,P2,P3*):

L0E reset on demand (*P1,P2,P3 + Activate*):

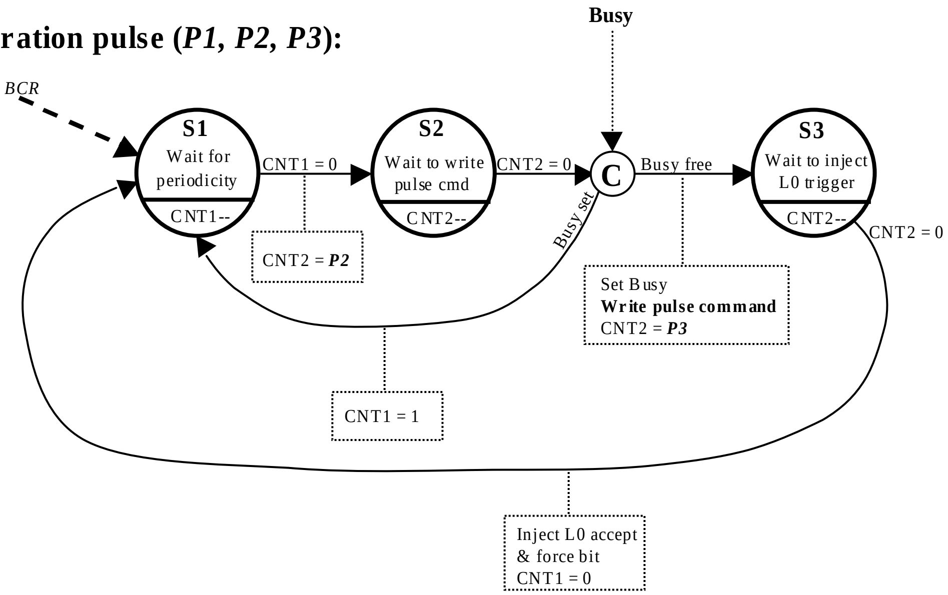
L1E+L0E reset on demand (*P1,P2 + Activate*):

Figure 15: The finite state machines.

Periodic trigger (*P1*, *P2*):



Calibration pulse (*P1*, *P2*, *P3*):



Consecutive bunch sampling (LHC full bunch mode)(*P1*,*P2*,*P3*):

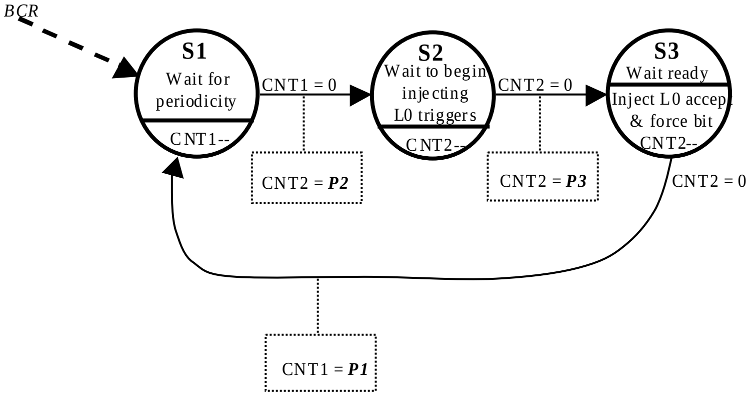


Figure 16: The finite state machines.

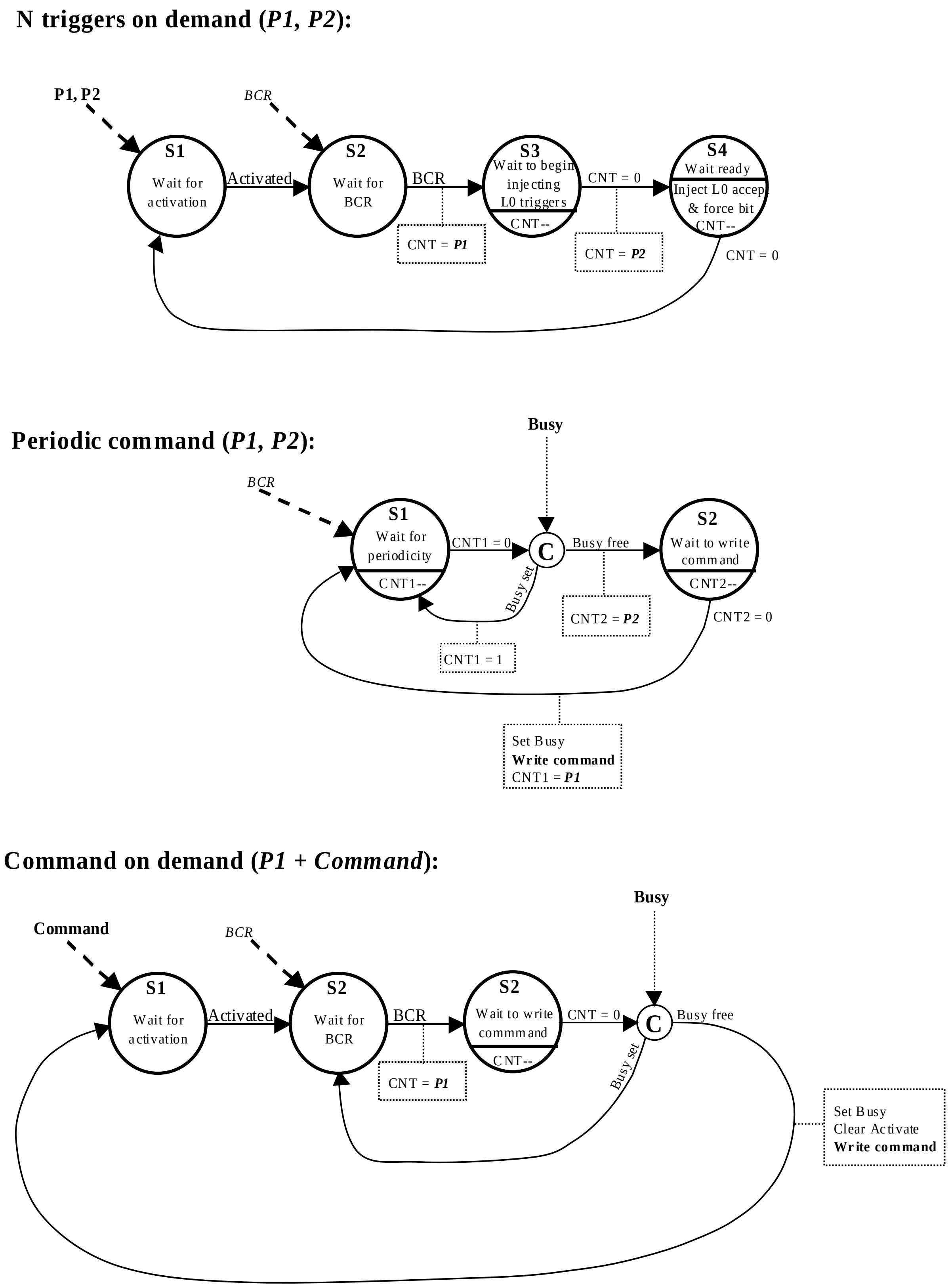


Figure 17: The finite state machines.

3.5.1 Command broadcasting

Command transmission is initiated by loading the TTC serializer directly with the command word. The commands are 8 bits wide[11] containing the command type and the command qualifier as described in Section 2.8 and 2.9. The commands should have higher priority than L1 trigger broadcasts. The loading of the command into the serializer is therefore preceded by an inhibit signal to the de-randomizer output strobe. As 17 cycles are required to transmit a channel-B short format broadcast, the 'generic command sender' FSM in the sequencer should assert the inhibit signal 17 cycles before the start of the command transmission. This prevents the L1 de-randomizer output strobe generator from initiating a transmission of a L1 trigger decision less than 17 cycles before the appropriate time for the command. The length of the inhibit signal is programmable in the 'generic command sender'.

There should be a counter for the number of commands broadcast.

3.6 Reset Controller

The Reset Controller is a central unit for transmitting resets to the various components of the Readout Supervisor. It should be able to reset:

- All counters individually[12]
- The L0 synchronization check
- The L0 accept FIFO
- The L1 synchronization check
- The L1 trigger de-randomizer
- All finite state machines individually
- The TTC Serializer and Encoder
- The L0 de-randomizer emulator
- The L1 buffer emulator
- The Trigger Controller
- The Throttle timeouts
- The Throttle history
- The "RS L0 Front-End"
- The "RS L1 Front-End"

The Reset Controller should have a reset register in which the bits correspond to the different components listed above. Writing to the register should trigger the appropriate resets. The entire set of requested resets should be fired in one cycle.

It should be possible to write to the reset register via the ECS interface.

[11] If transmission of TTC long format broadcasts (42-bits) is required, the command broadcast generator must be able to receive 16-bit words from the sequencer. We discourage this as it would have a serious negative effect on the maximum L1 trigger rate (see 2.7)

[12] It is essential that all counters related to dead-time can be reset at the same time in one cycle.

The Reset Controller should receive the commands from the sequencer at the same time with the TTC serializer/encoder. For commands involving resets, the Reset Controller should be able to decode the commands and transmit the appropriate internal resets. The set of resets corresponding to a specific command should be programmable. The Reset Controller should also be able to control the timing of the resets with respect to the broadcast by using the shifter signal of the TTC serializer. The falling edge of the shifter signal indicates that the broadcast has been completed.

3.7 TTC serializer/encoder

As the trigger, timing and control information will have to be broadcast over large distances to the Front-End electronics and require fast transmission and low jitter, the Readout Supervisor will make use of the TTC system developed by RD12[**3**].

The TTC system has been designed to distribute a low-latency A-channel, and framed and formatted broadcasts or addressed commands on a B-channel. In the LHCb readout, channel A will transmit the L0 decision and channel B will transmit the L1 decision as well as the synchronous framed broadcasts containing commands to the Front-End electronics. The channel-B broadcasts are 16-bits out of which six can be user defined. This allows a broadcast rate of 2.5 MHz.

The two channels are transmitted serially as encoded by a 160.32 Mbaud bi-phase mark encoder which time-division multiplexes the two channels. The serialization and the encoding have been implemented already for the TTCvx [**12**] and can be inherited for the Readout Supervisor.

The encoded signal should be available on the Front-Panel as one AC-coupled ECL output and two LVDS outputs.

The TTC encoder is clocked directly with the LHC clock. Thus the L0 and the L1 transmission are resynchronized, and the jitter of the Readout Supervisor is determined by the jitter of the LHC clock and the encoder alone. The jitter after the encoder has been measured to be 10-20 ps (RMS).

At the other end of the TTC fibre the receiver chips, TTCrx's, reconstruct the clock for local use, and extract the channel A and the channel B contents. The jitter at the output of the TTCrx has been found to depend strongly on the activity on channel B[**13**].

3.8 Trigger distribution controller

The Trigger Distribution Controller should receive the different throttles and the trigger disable/enable signals from the ECS interface and the sequencer, and combine them through an OR into a single L0 and a single L1 inhibit signal. The states of different throttles/disables should be readable via the ECS interface. There should also be a programmable timeout for each trigger throttle. The timeout parameters should be specified in terms of number of LHC turns (~89.1µs).

The Trigger Distribution Controller should have a programmable option to introduce gaps of programmable length between L0 accepts by eventually throttle triggers following an L0 accept.

The Trigger Controller should have a programmable register that specifies the number of 'L1 yes' triggers to be accepted before disabling the triggers automatically. Enabling the trigger with zero in the register should mean to run until the trigger is disabled via the ECS interface.

3.8.1 L0 de-randomizer occupancy emulator

The occupancy of the L0 de-randomizer buffer should be monitored by emulating its behaviour in a finite state machine. Issuing and releasing the L0 throttle to prevent overflow is determined by the two programmable parameters in Figure 18: *P1* is the level at which to throttle and *P2* is the level at which to release the throttle. *P3* is the readout time of the L0 de-randomizer. The occupancy controller should be able to assert the L0 throttle in time for the next trigger (i.e. in one cycle) following a L0 accept that would reach the upper occupancy level. Similarly the controller should be able to release the L0 throttle in time for the next L0 trigger (i.e. in one cycle) once the occupancy counter reaches the acceptable level.

It should be possible at any time to read the value of the L0 de-randomizer occupancy counter from the ECS interface. In this way the occupancy distribution can be histogrammed to analyze the behaviour of the system. The occupancy counter should be reset simultaneously with a reset of the Front-End L0 electronics.

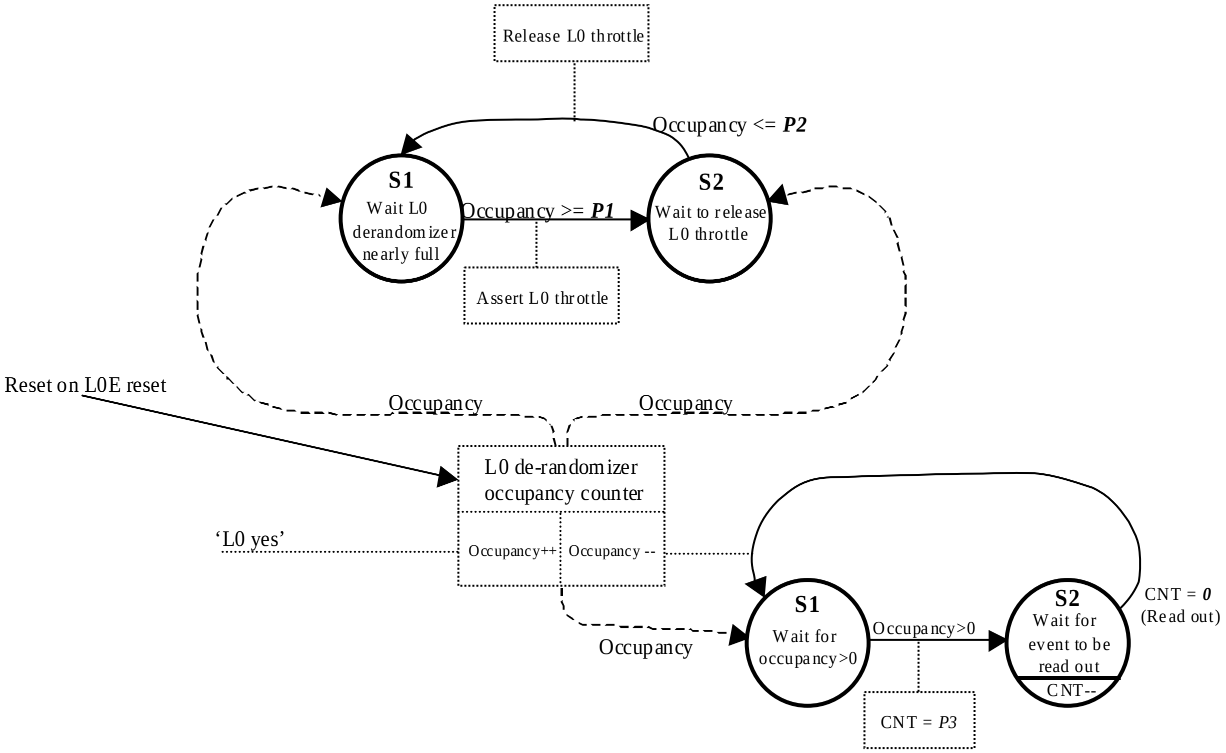


Figure 18: FSM drawing of the L0 de-randomizer occupancy controller. CNT is a clock counter determining the timing of the next transition. *P3* is the readout time, currently 36 cycles.

3.8.2 L1 buffer occupancy emulator.

In the same way that it is possible to emulate the occupancy of the L0 de-randomizer, it is also possible to monitor the L1 buffers remotely in the Readout Supervisor. The occupancy is given by the number of events that have been read out of the L0 de-randomizer less the number of L1 trigger decisions transmitted. The L0 de-randomizer occupancy controller emulates the readout of the L0 de-randomizer. Thus information can be obtained directly from the finite state machine in Figure 18. Figure 19 shows how the L1 buffer occupancy is emulated.

An imminent overflow situation should lead to throttling the L0 trigger, i.e. converting the succeeding L0 accepts into L0 rejects. The throttle level should have a safe margin (>>16 events margin in the L1 buffer) as the emulator throttles the L0 trigger without taking into account the number of events pending in the L0 de-randomizer, the transmission time of the L1 trigger broadcasts and the readout time of the L1 buffer.

It should be possible at any time to read the value of the L1 buffer occupancy counter from the ECS interface. As for the L0 de-randomizer, this allows histogramming the occupancy distribution in order to analyze the behaviour of the system. The occupancy counter should be reset simultaneously with a reset of the Front-End L1 electronics.

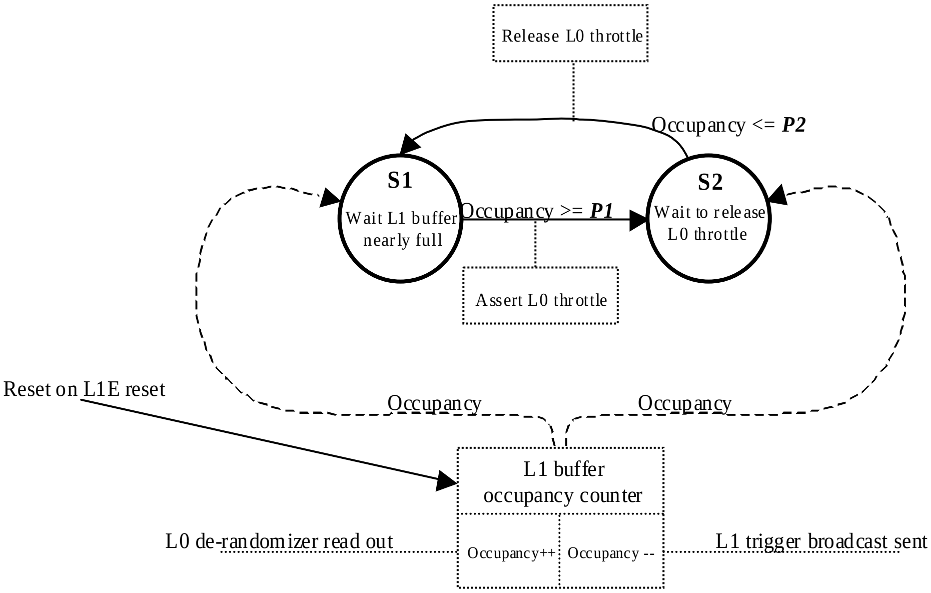

Release L0 throttle
Occupancy <= P2
S1
Wait L1 buffer nearly full
Occupancy >= P1
S2
Wait to release L0 throttle
Assert L0 throttle
Reset on L1E reset
Occupancy
Occupancy
L1 buffer occupancy counter
L0 de-randomizer read out
Occupancy++
Occupancy --
L1 trigger broadcast sent

Figure 19: FSM drawing of the L1 buffer occupancy controller. The L1 buffer occupancy counter is incremented by connecting it to the L0 de-randomizer emulator in Figure 18, i.e. the L1 buffer occupancy counter is incremented when the L0 de-randomizer occupancy counter is decrement.

3.8.3 Throttle history

The history of each type of trigger throttle/disable should be retrievable. By storing the state of all throttle/disable signals at every transition of any throttle and a time stamp in a 32k deep 80-bit wide FIFO, as in the Throttle Switch[**2**], the history can be read by the ECS interface.

3.9 “Readout Supervisor Front-End”

The Readout Supervisor Front-End should have the same components and behave as the standard Front-End of any detector. It should buffer the current status information and finally provide the DAQ with the RS Data block. Information will also have to be appended to the data at different stages of the Front-End stream as it is available.

The RS should receive the universal time, information about the overall status of the detector and other bunch crossing related information via an interface on the front panel.

The information composing the Readout Supervisor data block (listed below) totals at present 320 bits (35 bits reserve).

It should be possible to reset the RS Front-End in the same manner as the standard Front-End. This should be controlled by the Reset Controller.

3.9.1 L0

The L0 pipeline should buffer instantaneous data related to the bunch crossing. To avoid sampling a large amount of data at ~40 MHz the RS should be equipped with only a small FIFO which can sample 8 bits of data from the front panel. The rest of the information should be appended when the L0 decision arrives. An example of data to be sampled in the L0 pipe line is:

- Bunch crossing data e.g. bunch information from beam pick-up[13]

The depth of the RS L0 pipeline in use is defined by the L0 latency at the level of the Readout Supervisor (160 – latency between RS and FE). When the L0 decision arrives, the accepted events should be stored in a L0 de-randomizer (at present 16 slots deep) and L0 trigger related information should be appended:

- Bunch ID (12 bits)
- Universal time as defined by the GPS system via the front panel (40 bits[14])
- Experiment status information via the front panel (80 bits)
- L0 Event-ID (24 bits)
- L0 trigger decision (external) (1 bit)
- Trigger type (3 bits)
- L0 trigger force bit (1 bit)
- Bunch ID from L0 trigger system (12 bits)
- Expected bunch crossing type from FSM (2 bits)
- L0 synch error (1 bit)
- L0 synch error accept forced (1 bit)
- L0 de-randomizer occupancy (6 bits)
- L0 accept FIFO occupancy (16 bits)

For what concerns the universal time, the resolution should be 1 ms. This means the triggers will be grouped in time windows of 1 ms. Since the time and the experiment status information is sampled at the arrival of the L0 decisions, the information will be shifted with respect to the time of the event by about 4 μs.

[13] If possible it would be interesting to set up a system connected to the two beam pick-ups in order to record the true state of each bunch crossing.

[14] 40 bits gives a one ms resolution: 4 bits year, 9 bits days of the year, 17 bits seconds of the day and 10 bits milliseconds. A 100 μs resolution could be obtained by dropping the year. An interface to a GPS receiver is under investigation.

3.9.2 L1

At this point the data should be read out of the L0 de-randomizer and stored in a L1 buffer (at present 2k events deep in the Front-End). When the L1 decision arrives the data should be stored in a L1 de-randomizer and L1 trigger related information should be appended:

- L1 Event ID (32 bits)
- L0 Event ID from L1 trigger system (12 bits)
- L1 trigger decision (external) (1 bit)
- L1 trigger force bit (1 bit)
- L1 synch error (1 bit)
- L1 synch error accept forced (1 bit)
- L1 trigger de-randomizer occupancy (16 bits)
- L1 buffer occupancy (14 bits)

The data should then be prepared to be read out by the DAQ.

3.10 Counters

The internal counters are all summarized in Section 5. They should be readable via the ECS interface at any time. It would be highly desirable that the contents of all counters are copied simultaneously into local buffers on a global read alert signal preceding the actual read[15].

The reset lines of all counters should be linked to the Reset Controller.

3.11 Adjustable delay

The Readout Supervisor should have adjustable delays with a resolution of 1 ns and a range 0 – 31 ns in order to phase adjust the LHC orbit signal and the L1 trigger input (strobe). The adjustable delays should be programmable via the ECS interface.

3.12 Auxiliary L0 trigger input

The auxiliary L0 trigger input allows connecting an external source for the L0 trigger. It should consist of a single cable input. The input should be latched by the LHC clock distributed on the RS board and fed through an adjustable delay. Selecting between the standard L0 trigger input and the auxiliary input should be programmable and be mutually exclusive. It would be desirable that the auxiliary input has an optional NIM to ECL signal converter that can be chained by programming a setup register. The auxiliary input is not drawn in Figure 6.

There should also be an option to set the force bit for every auxiliary trigger in order to force the L1 decision.

[15] If this is difficult to implement in a general fashion, it is still important to have this feature for certain sets of counters related to deadtime and performance.

3.13 Internal oscillator

There will be periods in which the LHC bunch clock is not running. For *in situ* test running, the TTCmi includes an internal oscillator. However, switching between external and internal is done by means of a toggle switch on the front panel.

Thus, the Readout Supervisor should have an internal oscillator at 40.08 MHz for easy switching and for tests in labs with no LHC link and no TTCmi. In order to have an LHC orbit signal, the internal oscillator must also have a second output with a set pre-scaling of 3564.

An internal oscillator (40.00 MHz, 100ppm) has been implemented already in the design of the TTC serializer and encoder (TTCvx)[**12**]. It also has a PLL frequency synthesizer circuit to generate multiples of the basic clock: 80 MHz and 160 MHz. This implementation can be inherited and modified to fit the RS design.

Although switching between using an external or the internal clock in the TTCvx is automatic by means of an external clock detection circuit, the RS should have an implementation where switching is carried out by programming. This is important in order to avoid problems with glitches. Two LEDs on the front-panel should indicate whether the external or the internal clock is used.

3.14 The ECS interface

The ECS interface should be mounted directly on a socket on the RS board. The connection between the ECS interface and the RS is defined by a separate “glue board”[**14**].

It should be possible to reset the ECS interface at any time without affecting the functionality or disturbing the activity of the Readout Supervisor.

3.15 RS board standard

A 9U x 400 mm Fastbus or VME board is foreseen for the Readout Supervisor. The front panel will probably be several slots wide and a separate patch panel will be used for the experiment status information input.

4 Power up state and initialization

4.1 Power up state

As soon as the Readout Supervisor is powered up it should immediately transmit the clock. By default, it should be the external clock coming from the TTCmi. This will clock the entire Front-End and trigger components synchronously and allow setting up the timing and resetting the Front-End. As this also implies that the Readout Supervisor begins to transmit L0 decisions to the Front-End, it is essential that the L0 path comes on in a state in which it permanently transmits L0 rejects. In order to ensure this and for the readout Supervisor to be in a proper state to be initialized, the following must be in the defined state at power up:

- The external L0 trigger strobe should be disabled
- The auxiliary L0 input should be disabled
- The L0 synchronization check should be disabled
- The L0 random generator should be disabled
- The L0 accept FIFO must be empty
- The L0 trigger inhibit from the ECS interface should be on
- The external L1 trigger input strobe should be disabled
- The L1 synchronization check should be disabled
- The L1 trigger de-randomizer must be empty
- The L1 trigger inhibit from the ECS interface should be on
- All the FSMs in the sequencer should be disabled
- The Reset Controller should come on in a proper state in order to be able to reset any component via the ECS interface

4.2 Initialization

The initialization begins by resetting the different logical components of the Readout Supervisor by writing into the reset register. This is followed by programming the "soft"[16] permanent parameters and the running mode parameters (see Table 5 and Table 6 in Section 5).

Once the state machines for the BCRs and the FE resets have been programmed, they should be enabled in order to prepare the FE for data taking. At this point the external L0 and the L1 trigger input strobes can be enabled.

Data taking is initiated by resetting all the counters and releasing the L0 and the L1 trigger inhibits. Data taking is stopped by inhibiting the L0 and the L1 triggers. This should also automatically involve loading the contents of all counters into read buffers.

[16] There is a distinction between "soft" permanent parameters and "hard" permanent parameters. The "soft" parameters are programable via the ECS interface and the "hard" parameters are either set manually directly on the board or are located in the FPGA code..

5 Summary of the RS parameters, counters, I/O interfaces and status LEDs

Table 4: Summary of the so called “hard” permanent parameters. “Hard permanent” parameters are settings done directly on the board or in the FPGA code.

Parameter	#b	Explanation
BCR delay	5	Delay of LHC turn signal input
L0 phase	1	Selecting edge of L0 trigger input (L0 strobe)
L1 delay	5	Delay of L1 trigger input (L1 strobe)

Table 5: Summary of the programmable, so called “soft”, permanent parameters. “Soft permanent” parameters are programmable settings which will typically always remain the same.

Parameter	#b	Explanation
L0 pipeline depth	5	Depth of the L0 trigger input pipeline
L0 synch check offset	12	Offset for the bunch ID counter used by the L0 synch. check
L1 accept interval	6	Interval between triggers after L1 accepts (trigger de-randomizer strobe generator)
L1 reject interval	6	Interval between triggers after L1 rejects (trigger de-randomizer strobe generator)
1x Sequencer periodicity	16	Periodicity for the ECR FSM (~0.4h)
11x Sequencer delays	12	Time parameters for the FSMs in the Sequencer
8x Sequencer commands	8	Commands to be broadcasted from the FSMs in the Sequencer
L1 random generator latency	12	See Section 3.4.2 (prescaled by a factor 40 --> ~ µs)
L1 random generator interval	16	See Section 3.4.2
Internal L0 inhibit timeout	16	Timeout on L0 inhibit from sequencer in terms of #LHC turns
Internal L1 inhibit timeout	16	Timeout on L1 inhibit from sequencer in terms of #LHC turns
External L0 throttle timeout	16	Timeout on external L0 throttle in terms of #LHC turns
External L1 throttle timeout	16	Timeout on external L1 throttle in terms of #LHC turns
L0 gaps	4	Length of forced gaps between L0 accepts
L0 de-randomizer overflow level	4	Overflow level used in the L0 de-randomizer occupancy controller
L0 de-randomizer accept level	4	Accept level used in the L0 de-randomizer occupancy controller
L0 de-randomizer readout time	6	Readout time used in the L0 de-randomizer occupancy controller
L1 buffer overflow level	12	Overflow level used in the L1 buffer occupancy controller
L1 buffer accept level	12	Accept level used in the L1 buffer occupancy controller
Consecutive L0 timing triggers	4	#Consecutive L0 triggers injected for timing calibration

Table 6: Summary of the programmable "running mode" parameters. “Running mode” parameters are programmable settings which are typically modified while the triggers are disabled in order to re-configure the activity of the readout Supervisor.

Parameter	#b	Explanation
External L0 input disable/enable	1	External L0 trigger enable/disable by ECS interface (L0 strobe)
External L1 input disable/enable	1	External L1 trigger enable/disable by ECS interface (L1 strobe)
Auxiliary L0 input disable/enable	1	Auxiliary L0 trigger input enable/disable by ECS interface
L0 synch check – enable/disable	1	Enabling/disabling the L0 synchronization check
L0 synch error – accept/reject	1	Accept or reject de-synchronized L0 trigger (~disable/enable)
L1 synch check enable/disable	1	Enabling/disabling the L1 synchronization check
L1 synch error – accept/reject	1	Accept or reject de-synchronized L1 trigger (~disable/enable)
10x Sequencer periodicities	20	Periodicities for the FSMs in the Sequencer (~0.4h)

12x Sequencer delays	12	Time parameters for the FSMs in the Sequencer
2x Sequencer commands	8	Commands to be broadcasted from the FSMs in the Sequencer
1x “N consecutive triggers”	4	Number of consecutive triggers in FSM 12, see Table 3.
Trigger enable duration	16	#L1 accepts before disabling the L0 and L1 trigger automatically (~0.5h)
L0 random generator seeds	-	
L0 random generator enable/disable	1	
L0 random generator force bit	1	Value of the force bit accompanying the random L0 triggers
L0 random generator pre-scaling	-	Pre-scaling factor for the L0 random generator
L1 random generator seeds	-	
L1 random generator enable/disable	1	
L1 random generator pre-scaling	-	Pre-scaling factor for the L1 random generator
NIM converter	1	Selecting optional NIM to ECL converter for the L0 auxiliary input

Table 7: Summary of the programmable "operational" parameters. “Operational” parameters are parameters which can be programmed via the ECS “on the fly” (i.e. while triggers are enabled) in order to interfere with the trigger distribution or to fire off some action.

Parameter	**#b**	**Explanation**
L0 trigger enable/disable (ECS)	1	L0 trigger inhibit via ECS interface
L1 trigger enable/disable (ECS)	1	L1 trigger inhibit via ECS interface
L0 trigger throttle (ECS)	1	L0 trigger throttle via ECS interface
L1 trigger throttle (ECS)	1	L1 trigger throttle via ECS interface
2x Sequencer delays	12	Time parameters for the FSMs in the Sequencer
1x “N consecutive triggers”	8	Number of consecutive triggers in FSM 13, see Table 3
2x Sequencer activates	1	Activate bits for the FSMs in the Sequencer
1x Sequencer commands	8	Commands to be broadcasted from the FSMs in the Sequencer
Reset register	-	Register in Reset Controller with one bit per component to reset

Table 8: Summary of the status bits

Status bit	**Explanation**
Generic command sender busy	Set when the state machine to send commands is busy
L0 strobe missing	Set when L0 strobe missing; cleared by ECS interface
Max #L0 strobes missing	Maximum #consecutive L0 strobes missing
L0 synch error	Set when L0 synch error occurs; cleared by ECS interface
Max #L0 synch error	Maximum #consecutive L0 synch errors reached
L1 synch error	Set when L1 synch error occurs; cleared by ECS
Max #L1 synch error	Maximum #consecutive L1 synch errors reached
External L0 throttle timeout	Timeout on the external L0 throttle
External L1 throttle timeout	Timeout on the external L1 throttle
L0 inhibit from sequencer timeout	Timeout on the L0 inhibit from the sequencer
L1 inhibit from sequencer timeout	Timeout on the L1 inhibit from the sequencer
L0 Accept FIFO half full	
L0 Accept FIFO almost full	This status is also used to throttle the L0 trigger
L0 Accept FIFO full	
L1 Trigger de-randomizer half full	
L1 Trigger de-randomizer almost full	
L1 Trigger de-randomizer full	

Table 9: Summary of the different trigger inhibits.

Level	Inhibit	Explanation
L0	ECS interface inhibit	Internal signal from ECS interface to inhibit the L0 trigger. Used both to “start/stop” the trigger and as software throttle via ECS.
L1	ECS interface inhibit	Internal signal from ECS interface to inhibit the L1 trigger. Used both to “start/stop” the trigger and as software throttle via ECS.
L0	Internal throttle 1	Throttle from emulation of L0 de-randomizer occupancy
L0	Internal throttle 2	Throttle from emulation of L1 buffer occupancy
L0	Internal throttle 3	Throttle from L0 Accept FIFO almost full
L0	Internal throttle 4	Throttle from the “L0 trigger gap generator”, to force gaps between “L0 yes’s”
L0	Hardware throttle	Hardware throttle arriving at front panel from the L0 throttle switch
L1	Hardware throttle	Hardware throttle arriving at front panel from the L1 throttle switch
L0	Sequencer inhibit	Internal signal asserted by sequencer to inhibit the L0 trigger
L1	Sequencer inhibit	Internal signal asserted by sequencer to inhibit the L1 trigger

Table 10: Summary of counters related to trigger statistics and dead-time. All the counters are 32 bits wide. The numbers correspond to the numbering in Figure 6. Most of these counters (marked with a star) should exist in ungated and gated versions (see Table 11).

#	Counter	Prescaling	Increment	Reset
*1	LHC bunch crossing	10	LHC clock	ECS instruction
*2	L0 accept (external, ungated)	4	L0 accept	ECS instruction
*3	L0 force bit (external, ungated)	-	L0 force bit set	ECS instruction
*4	L0 accept (random, ungated)	4	L0 accept from random generator	ECS instruction
*5	L0 accept (auto-, ungated)	-	L0 accept from sequencer	ECS instruction
*6	L0 accept total (ungated)	4	L0 accept	ECS instruction
*7	L0 force bit total (ungated)	-	L0 force bit set	ECS instruction
*8	L0 synch error (ungated)	-	L0 synch error reject	ECS instruction
9	L1 triggers (ungated)	4	L1 strobe	ECS instruction
10	L1 accepts (external, ungated)	-	L1 accept	ECS instruction
11	L1 force bit (external, ungated)	-	L1 force bit	ECS instruction
12	L1 accept (random, ungated)	-	L1 accept from random generator	ECS instruction
13	L1 force bit (ungated)	-	L1 force bit (out of L0 Accept FIFO)	ECS instruction
14	L1 trigger total (ungated)	4	De-randomizer in-strobe	ECS instruction
15	L1 accept total (ungated)	-	L1 accept (before de-randomizer)	ECS instruction
*16	L1 trigger total (ungated)	4	De-randomizer out-strobe	ECS instruction
*17	L1 accept total (ungated)	-	L1 accept	ECS instruction
18	L1 synch error (ungated)	-	L1 synch error reject	ECS instruction

Table 11: Gated versions of the counters in Table 10. All the counters are 32 bits wide.

Gated counters	Prescaling	Reset
Counter 1 && (L0inhibit) (All inhibits combined)	4	ECS instruction
Counter 1 && (Internal L0 throttle 1)	4	ECS instruction
Counter 1 && (Internal L0 throttle 2)	4	ECS instruction
Counter 1 && (External L0 throttle)	4	ECS instruction
Counter 1 && (ECS L0 throttle)	4	ECS instruction
Counter 1 && (L0Inhibit by sequencer)	4	ECS instruction
Counter 1 && (L1inhibit) (All inhibits combined)	4	ECS instruction
Counter 1 && (External L1 throttle)	4	ECS instruction
Counter 1 && (ECS L1 throttle)	4	ECS instruction
Counter 1 && (L1Inhibit by sequencer)	4	ECS instruction
Counters 2 – 8 && (L0inhibit) (All inhibits combined) (8 counters)	-	ECS instruction

Counters 2 – 8 && (Internal L0 throttle 1) (8 counters)	-	ECS instruction
Counters 2 – 8 && (Internal L0 throttle 2) (8 counters)	-	ECS instruction
Counters 2 – 8 && (External L0 throttle) (8 counters)	-	ECS instruction
Counters 2 – 8 && (ECS L0 throttle) (8 counters)	-	ECS instruction
Counters 2 – 8 && (L0 Inhibit by Sequencer) (8 counters)	-	ECS instruction
Counters 16 – 17 && (L1inhibit) (All inhibits combined) (2 counters)	-	ECS instruction
Counters 16 – 17 && (External L1 throttle) (2 counters)	-	ECS instruction
Counters 16 – 17 && (ECS L1 throttle) (2 counters)	-	ECS instruction
Counters 16 – 17 && (L1inhibit by Sequencer)	2*26	ECS instruction

Table 12: Summary of the internal general counters. The width is the minimum number of bits for the counter. The numbers correspond to the numbering in Figure 6.

#	Counter	Width	Increment	Reset
22	LHC orbit	32	BCR	ECS instruction
23	Bunch ID	12 (3564)	LHC Clock	BCR
24	L0 Event ID	24	L0 accept && (!L0inhib)	ECR
25	L1 Event ID	32	L1 accept && (!L1inhib)	ECS instruction
26	L0 accept FIFO occupancy	14	FIFO occupancy	L0 electronics reset
27	L1 trigger de-random. occupancy	16	Trigger de-random. occupancy	L1 electronics reset
28	L0 de-randomizer occupancy	6	L0 FE de-randomizer occupancy	L0 electronics reset
29	L1 buffer occupancy	14	L1 FE buffer occupancy	L1 electronics reset
30	16x Activity counters for FSMs	28	Transition to initial state	ECS instruction
31	L1 broadcast inhibit	28	L1 broadcast inhibit	ECS instruction
32	Command broadcasts	28	Command broadcasts	ECS instruction
33	TTC broadcast	32	TTC broadcast	ECS instruction
34	L0 internal throttle 1 & 2	2*20	L0 internal throttle 1 & 2	ECS instruction
35	L0 external throttle	20	L0 external throttle	ECS instruction
36	L1 external throttle	20	L1 external throttle	ECS instruction
37	L0 software throttle by ECS	12	L0 software throttle	ECS instruction
38	L1 software throttle by ECS	12	L1 software throttle	ECS instruction
39	L0 trigger inhibit by Sequencer	26	L0 trigger inhibit	ECS instruction
40	L1 trigger inhibit by Sequencer	20	L1 trigger inhibit	ECS instruction
41	L0 trigger inhibit total	26	L0 trigger inhibit	ECS instruction
42	L1 trigger inhibit total	20	L1 trigger inhibit	ECS instruction
43	Reset by Reset Controller	16	Reset lines asserted	ECS instruction

Table 13: Summary of the interfaces.

Input/output	I/O	Specification	Type
CLK in	In	LHC bunch clock from TTCmi, AC-coupled ECL	Lemo
CLK out 1 (ECL)	Out	LHC bunch clock output, balanced AC-coupled ECL	Lemo
CLK out 2 (LVDS)	Out	LHC bunch clock output, LVDS	LVDS
Clock out (mon)	Out	LHC bunch clock output for monitoring, NIM?	Lemo
BCR in	In	LHC turn signal from TTCmi, DC-coupled ECL	Lemo
BCR out	Out	LHC turn signal output, DC-coupled ECL	Lemo
BCR out (mon)	Out	LHC turn signal output for monitoring, NIM?	Lemo
L0 in	In	L0 trigger decision with 12-bit bunch id + forced L0 bit + strobe, LVDS?	34-pin parallel
L0 in (aux)	In	Auxiliary L0 trigger from any source (NIM signal converter?)	Lemo
L0 out	Out	L0 trigger decision output for monitoring, NIM?	Lemo
L1 in	In	L1 trigger decision with 12-bit L0 Event ID + forced L1 bit + strobe, LVDS?	34-pin parallel
L1 out	Out	L1 trigger decision output for monitoring, NIM?	Lemo

TTC out (ECL)	Out	Channel A/B serialized and encoded, balanced AC-coupled ECL	Lemo
TTC out 1 (LVDS)	Out	Channel A/B serialized and encoded LVDS	LVDS
TTC out 2 (LVDS)	Out	Channel A/B serialized and encoded, LVDS	LVDS
Throttle L0 in	In	L0 throttle input e.g. from L0 throttle switch, LVDS	LVDS
Throttle L0 out	Out	L0 throttle output for monitoring, NIM?	Lemo
Throttle L1 in	In	L1 throttle input e.g. from L1 throttle switch, LVDS	LVDS
Throttle L1 out	Out	L1 throttle output for monitoring, NIM?	Lemo
Status in	In	Experimental status information + strobe	parallel 80 bits
BX data	In	Data sampled at 40 MHz in RS L0 pipeline	parallel 8 bits
U-time	In	Universal time as defined by GPS system	parallel 40 bits
RS data	Out	RS data to DAQ	S-link
Ethernet	In/out	ECS interface	Ethernet

Table 14: Summary of the status LEDs.

LED	**Specification**	**Colour**
On/Off	Power on	Green
CLK ext	External LHC clock present	Green
CLK int	Internal LHC clock used	Yellow
BCR ext	External orbit clock present	Green
BCR int	Internal orbit clock used	Yellow
L0	L0 trigger present	Green
No L0 strobe	L0 strobe missing	Yellow
No L0 strobes	Maximum #L0 strobes missing	Red
L1	L1 trigger present	Green
Thrtl L0	L0 throttle asserted, stretched on-time	Red
Thrtl L1	L1 throttle asserted, stretched on-time	Red

6 Appendix: First stage RS prototype (2001)

The list below summarizes the minimal functionality required in the first stage prototype of the Readout Supervisor. The aim with the first prototype is to demonstrate the feasibility and to build a Readout Supervisor with all the most basic functionality to drive the read-out. Regarding feasibility there is serious concern about the speed of the FPGAs.

L0 trigger handling module

- L0 trigger strobe check
- L0 trigger pipeline, programmable depth
- L0 trigger synchronization check
- The logical OR of the different trigger sources
- Trigger resolver (first version) to generate the L1 trigger qualifier
- L0 inhibit

L0 accept FIFO

- Occupancy status (empty, almost empty, half full, almost full, full)
- L0 throttle on almost full

L1 trigger handling module

- L1 trigger input delay
- State machine for generating internally output strobe to AFIFO
- L1 trigger synchronization check
- L1 inhibit

L1 trigger de-randomizer

- Occupancy status (empty, almost empty, half full, almost full, full)

L1 trigger de-randomizer module

- State machine for L1 trigger sending (rate control and multiplexing trigger and command broadcasts)

TTC module

- TTC Shifter
- TTC Encoder
- Internal clock and turn signal

Trigger inhibit module (trigger controller)

- L0 de-randomizer emulator
- L1 buffer emulator
- L0 hardware throttle

- L1 hardware throttle
- L0 throttle from ECS interface
- L1 throttle from ECS interface
- L0 inhibit from sequencer
- L1 inhibit from sequencer
- OR of all throttles
- L0 trigger gap mechanism

Random generator module

- L0 random generator (programmable force always/never force)
- L1 random generator

Trigger sequencer module

- State machine for periodic trigger

Command sequencer module

- State machine for generic command sender
- State machine for BCR/ECR
- State machine for periodic L0 reset
- State machine for periodic L0+L1 reset
- State machine for calibration trigger
- Enable/disable register for state machines (setting bit from ECS activates state machine as on demand)

Status handler module

- Status register

Reset handler module

- Reset register with reset bit for all modules (used by FE reset state machines and via ECS interface)
- Reset command resolver to perform the appropriate internal resets when resetting FE

Counters

- Simultaneous sampling of counters in separate buffer to read consistent set
- A subset of the counters related to the minimal functionality:

Table 15: Summary of counters related to trigger statistics and dead-time. All the counters are 32 bits wide. Most of these counters (marked with a star) should exist in ungated and gated versions (see Table 11).

#	Counter	Prescaling	Increment	Reset
*1	LHC bunch crossing	10	LHC clock	ECS instruction
*2	L0 accept (external, ungated)	4	L0 accept	ECS instruction
*3	L0 force bit (external, ungated)	-	L0 force bit set	ECS instruction
*4	L0 accept (random, ungated)	4	L0 accept from random generator	ECS instruction
*5	L0 accept (auto-, ungated)	-	L0 accept from sequencer	ECS instruction

*6	L0 accept total (ungated)	4	L0 accept	ECS instruction
*7	L0 force bit total (ungated)	-	L0 force bit set	ECS instruction
*8	L0 synch error (ungated)	-	L0 synch error reject	ECS instruction
9	L1 triggers (ungated)	4	L1 strobe	ECS instruction
10	L1 accepts (external, ungated)	-	L1 accept	ECS instruction
11	L1 force bit (external, ungated)	-	L1 force bit	ECS instruction
12	L1 accept (random, ungated)	-	L1 accept from random generator	ECS instruction
13	L1 force bit (ungated)	-	L1 force bit (out of L0 Accept FIFO)	ECS instruction
14	L1 trigger total (ungated)	4	De-randomizer in-strobe	ECS instruction
15	L1 accept total (ungated)	-	L1 accept (before de-randomizer)	ECS instruction
*16	L1 trigger total (ungated)	4	De-randomizer out-strobe	ECS instruction
*17	L1 accept total (ungated)	-	L1 accept	ECS instruction
18	L1 synch error (ungated)	-	L1 synch error	ECS instruction

Table 16: Gated versions of the counters in Table 10. All the counters are 32 bits wide.

Gated counters	**Prescaling**	**Reset**
Counter 1 && (L0inhibit) (All inhibits combined)	4	ECS instruction
Counter 1 && (L1inhibit) (All inhibits combined)	4	ECS instruction
Counters 2 – 8 && (L0inhibit) (All inhibits combined) (7 counters)	-	ECS instruction
Counters 16 – 17 && (L1inhibit) (All inhibits combined) (2 counters)	-	ECS instruction

Table 17: Summary of the internal general counters.

Counter	**Width**	**Increment**	**Reset**
LHC orbit	32	BCR	ECS instruction
Bunch-ID	12 (3564)	LHC Clock	BCR
L0 event-ID	24	L0 accept && (!L0inhib)	ECR
L1 event-ID	32	L1 trigger accept broadcasts	ECS instruction
L0 de-randomizer occupancy	6	L0 FE de-randomizer occupancy	L0 electronics reset
L1 buffer occupancy	14	L1 FE buffer occupancy	L1 electronics reset

ECS interface

- Credit Card PC
- Code to download to FPGAs and configure Readout Supervisor

7 References

[1] B. Jost and R. Jacobsson, "LHCb TFC system", LHCb 2001-016 DAQ.

[2] B. Jost, Z. Guzik and R. Jacobsson,"TFC Switch specifications", 2001-018 DAQ

[3] RD-12 Documentation on WWW (http://www.cern.ch/TTC/intro.html) and references therein

[4] C. Gaspar, R. Jacobsson and B. Jost, "Partitioning in LHCb", in preparation.

[5] Z. Ajaltouni et al., "Status of the L0 trigger decision unit design", LHCb 99-019 TRIG.

[6] J. Christiansen et al., "The latency of the Level 0 trigger", LHCb 99-015 TRIG

[7] L1DU note in preparation

[8] I. Garcia and J. Christiansen, "Simulation of the LHCb Front-End", LHCb FE 99-047

[9] B. Jost, "TTC Broadcast Format (proposal)", LHCb 2001-017 DAQ.

[10] J. Christiansen, "Requirements to the L0 Front-End electronics", LHCb 99-029 FE
update LHCb 2001-014 FE

[11] J. Christansen and P. Vazquez Regueiro, "Requirements to the L1 Front-End eletcronics", in preparation

[12] Per Gällnö, "TTCvx, Technical description and users manual"

[13] Private communication with Thomas Toifl.

[14] B. Jost et al. "Credit Card PC as ECS interface to FE and DAQ electronics", in preparation.