CPPM, IN2P3−CNRS et Université d'Aix−Marseille II

# Specification of the muon trigger processing board

## LHCb Technical Note

Issue: 1
Revision: 5

Reference: LHCb 2002−003
Created: 5 September 2001
Last modified: 28 March 2002

**Prepared by:** E. Aslanides, J.P. Cachemiche, B. Dinkespiler, F. Derue, P.Y. Duval, R. Le Gac, O. Leroy, P.L. Liotard, M. Menouni, R. Potheau and A. Tsaregorodtsev

# Abstract

In this note we establish the specification of the processing board for the L0 muon trigger [1].

# Document Status Sheet

| **1. Document title: Specification of the muon trigger processing board** | | | |
|---|---|---|---|
| **2. Document Reference Number: LHCb 2002−003** | | | |
| **3. Issue** | **4. Revision** | **5. Date** | **6. Reason for change** |
| 1 | 0 | 11 September 2001 | • First draft |
| 1 | 1 | 16 October 2001 | • Introduction of 16 bits alignment on register contents<br>• Added description of the event capture mechanism<br>• Added description of L1 derandomizer control and throttling mechanisms |
| 1 | 2 | 23 November 2001 | • Complete reorganisation for better readability<br>• Added glossary of terms and definitions<br>• Added numbering of figures and tables<br>• Added index of tables and figures<br>• Added presentation by entity<br>• Added LUT filling mechanism<br>• Register mapping and presentation modification<br>• L1 mechanism description modification |
| 1 | 3 | 14 December 2001 | • Added refreshed synoptic of the PU<br>• Modified PU control register content<br>• Added pattern injection and result reading in the BCSU<br>• Added throttle status control and masking possibility in the L1MU |
| 1 | 4 | 24 January 2002 | • Added detail or error word encoding for simulated data<br>• Improved description of capture of events in test mode and in operational modes.<br>• Added L1 trottle time monitoring register in LMU<br>• Modified encoding of status word output from PUs<br>• Various modifications for improving the readability of the document |
| 1 | 5 | 26 March 2002 | • Added signal exchange enumeration in Annex 1<br>• Added some precisions taking into account remarks from Hans Dijkstra |

# Table of Contents

**1 Introduction....7**

**2 Definition of technical terms and abbreviations....9**

**2.1 System definitions....9**
**2.2 Processing board specific definitions....11**
**2.3 Reset signals definitions....12**
**2.4 Service signal definition ....14**

**3 General structure of the processing board....15**

**4 General principles....17**

**4.1 Synchronisation of the data coming from the ODE....17**
**4.2 Synchronisation with the machine cycles....18**
4.2.1 Time reference....19
4.2.2 Current time estimations....20
4.2.3 Length of the L0 buffer for the muon trigger....21
4.2.4 Synchronisation of the muon trigger with the L0_B_Id....21
4.2.5 Start/stop of the processing....21
**4.3 Error recovery mechanisms....22**
**4.4 Test features....23**
4.4.1 Test of a PU....23
4.4.2 Test of BCSU....24
4.4.3 Test of L1MU....25

**5 Specification of the optical reception....26**

**5.1 Optical reception inputs....26**
**5.2 Optical reception outputs....26**

**6 Specification of the Processing Unit FPGA....28**

**6.1 PU inputs and outputs....29**
6.1.1 Connection with the optical transceivers....29
6.1.2 Connection with the optical reception module....30
6.1.3 Connection with other PUs of the Processing Board....31

6.1.4 Connection with PUs from neighbouring boards.....31
6.1.5 Connection with the PT calculation module.....32
6.1.6 Connection with the BCSU.....32
6.1.7 Connection with the L1 buffer.....33
6.1.8 Connection with the controller board.....35
6.1.9 Connection with the local ECS supervisor.....35
6.1.10 Output for debug.....35
**6.2 PU optical input stages and associated error detection.....36**
**6.3 Demultiplexing stage and associated error detection.....36**
**6.4 Synchronisation stage and associated error detection.....37**
**6.5 Neighbouring exchanges.....38**
**6.6 Candidates search in M2, M3, M4 & M5.....39**
**6.7 Extrapolation to M1.....41**
**6.8 Candidates selection.....41**
**6.9 L0 Buffer.....42**
**6.10 L0 derandomizer buffer.....44**
6.10.1 Link to the L1 Buffer.....44
**6.11 Interface with the local ECS supervisor.....44**
6.11.1 PU operation control and status registers.....45
6.11.2 Registers used in test mode.....46
6.11.2.1 PU pattern injection registers.....47
6.11.2.2 PU consecutive events capture registers in test mode.....49
6.11.3 Registers used in operational mode.....50
6.11.3.1 PU error detection flags and counter registers.....51
6.11.3.2 PU event capture register in operational mode.....53
6.11.4 Registers used in test and operational modes.....55
6.11.4.1 PU parameters registers.....55
6.11.4.2 PU masking operations registers.....56
*Optical link.....56*
*Backplane link.....57*
*On board PU link.....57*
6.11.4.3 Local resets registers.....58
*Error detection on the sequencers.....58*
*Error detection on L0 derandomizer.....59*
6.11.4.4 Local resets.....59
6.11.5 Physical interface of the registers with the local ECS supervisor. 60
6.11.6 Addresses of the registers in a PU.....60

**7 Specification of the PT computation module.....62**

**7.1 Implementation.....62**

**7.2 Initialisation of the LUT....62**

**8 Specification of the BCSU FPGA....64**

**8.1 Inputs and outputs....64**
8.1.1 Connection to the PUs....64
8.1.2 Connection to the PT calculation module....64
8.1.3 Connection to the L1 buffer....65
8.1.4 Connection to the muon selection board....66
8.1.5 Connection to the controller board....67
8.1.6 Connection to the local ECS supervisor....67
8.1.7 Output for debug....67
**8.2 Internal best candidate algorithm....68**
**8.3 L0 buffer....68**
**8.4 L0 derandomizer buffer....69**
**8.5 Interface with the local ECS supervisor....69**
8.5.1 BCSU Control of operations register....70
8.5.2 Registers used in test mode....71
8.5.2.1 BCSU pattern injection registers....71
8.5.2.2 BCSU consecutive events capture registers....72
8.5.3 Registers used in operational mode....74
8.5.3.1 BCSU event capture register....74
8.5.3.2 BCSU synchronisation error detection register....75
8.5.3.3 BCSU masking register....75
8.5.4 LUT programming register....76
8.5.5 Physical interface of the registers with the local ECS supervisor ..76
8.5.6 Addresses of the registers in the BCSU....77

**9 Specification of the L1 buffer....78**

**9.1 Inputs and outputs....78**
**9.2 Operation....79**

**10 Specification of the L1MU FPGA....80**

**10.1 Inputs and outputs....80**
10.1.1 Connection with the L1 buffer....80
10.1.2 Connection to the controller board....81
10.1.3 Connection with the local ECS supervisor....85
10.1.4 Output for debug....85
**10.2 Reading the L1 buffer and filling the L1 derandomizer buffer....85**
**10.3 Tagging the L1 event....86**
**10.4 L1 derandomizer buffer....86**

10.4.1 Control of the L1 derandomizer overflow....87
**10.5 Formatting....87**
**10.6 Serialisation....87**
**10.7 Interface with the local ECS supervisor....88**
10.7.1 Control of operations....88
10.7.2 Registers used in test mode....89
10.7.2.1 Pattern injection registers....89
10.7.3 Registers used in operational mode....90
10.7.3.1 Error detection registers....90
10.7.3.2 L1 buffer control register....91
10.7.3.3 L1 derandomizer control register....92
10.7.3.4 L1 throttle time monitoring register....92
10.7.4 Physical interface of the registers with the local ECS supervisor ....93
10.7.5 Addresses of the registers in the L1MU....93

**11 Specification of the local ECS supervisor....95**
**11.1 Inputs/Outputs....95**
**11.2 Interface with the FPGAs....96**
**11.3 Mapping of the FPGA registers....96**
**11.4 Mapping of the LUTs....97**

**12 Board clocks....98**

**13 Board inputs and outputs connectors definition....99**
**13.1 Optical transceivers....99**
**13.2 Backplane....99**

**14 Mechanical specifications....100**

**15 Electrical consumption....101**

# 1 Introduction

The Level 0 muon trigger is subdivided in 60 processing boards [1]. In spite of the non homogeneous granularity of the muon detectors cells [2], we define a map where a unique processing board finds candidates in any portion of the detector chamber in the same way. Such a map is shown in Figure 1.

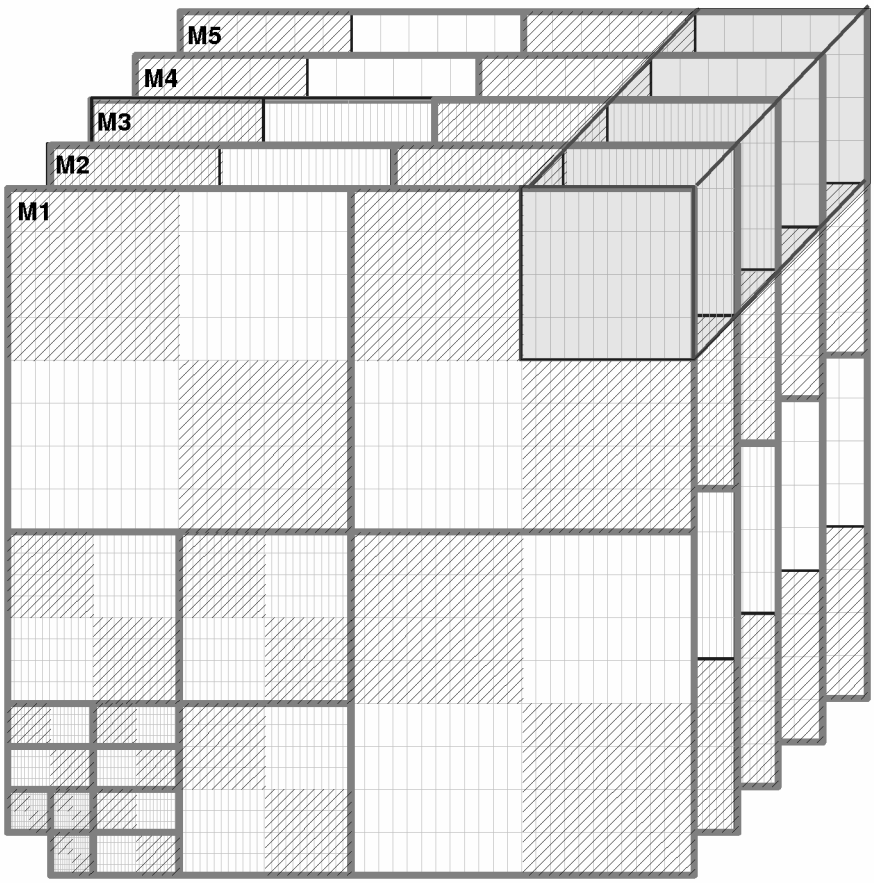



**Figure 1: Areas processed by processing boards in the muon chambers.**

An area delimited by a thick line is processed by a processing board. The tower shown on the right upper corner is analysed by a PU. A processing board handles mostly 4 PUs.

The goal of this document is to specify the hardware, mechanical and software interface features of the generic processing board.

In section 2, we define technical terms and abbreviations used in this document. In section 3, we give a general overview of the processing board. In section 4, we describe basic concepts used in the synchronisation and error recovery mechanisms. In sections 5, 6 and 7 we specify respectively the optical reception module, the Processing Unit and the PT calculation modules. In section 8, we present the Best Candidate Search Unit. Section 9 and 10 are devoted to L1 buffers and their management. In section 11, we describe the local Experiment Control System supervisor. The last sections from 12 to 15 are devoted to clock distribution, connectors, mechanical and electrical aspects.

# 2 Definition of technical terms and abbreviations

Definition of the technical terms related to the front−end electronics are extracted from documents describing the requirements of the front−end electronics [3−5].

## 2.1 System definitions

- **TTCrx:** The Trigger and Timing Control Receiver is a specialised ASIC which has been implemented to receive the TTC signals distributed via the optical fan−out system. The TTCrx generates two clocks with programmable phase to align the different sub−systems to the beam crossing and across modules. The L0 trigger decision is delivered as a synchronous accept/reject signal together with the bunch ID (L0_B_Id) of the events accepted. The L1 trigger is delivered as a accept/reject broadcast in correct sequential order for each event accepted by the L0 trigger.

- **ECS:** The Experiment Control System is the top level controller of the whole LHCb experiment. All parts of the front−end electronics (and all other sub−systems) must in one way or an other be connected to the ECS system. Loading front−end parameters and monitoring the front−end electronics, will be under the control of the ECS. The DAQ and trigger systems will also be under the control of the ECS. The ECS will be a highly distributed system with many intelligent local controllers ( e.g. one local controller for the front−end electronics of one sub−detector). A local controller handles a part of the system with a minimum of communication to the higher levels of the ECS. The front−end electronics system must provide a means of loading and monitoring its parameters through the ECS system.

- **L0_B_Id:** The L0 Bunch Identifier is an identification of the bunch crossing within the LHC bunch structure. The bunch ID can be added to the data stream in the front−end electronics at three different locations: directly at the input of the L0 pipeline buffer, at the input of the L0 derandomizer buffer (default), or at the latest when data is written into the L1 buffer. The tagging of data fragments in the front−end with the L0_B_ID enables the front−end and the DAQ system to verify the correct synchronisation between different parts of the front−end system. The earlier the L0_B_ID tag is added to the data flow the better one can verify the correct function of the front−end system. The

L0_B_ID is directly controlled by the TTCrx. The later sends a L0_B_ID counter reset to generate a correct bunch identification at the input of the L0 derandomizer. The TTCrx has an internal L0_B_ID counter which is available at its pins for each L0 accept. Alternative representations of the L0_B_ID can be used if it simplifies the front−end electronics or it increases the error checking capabilities.

- **L0 buffer:** Captured data must be stored in the L0 pipeline buffer, until the level 0 trigger decision accepts or rejects them. The L0 latency is defined as the maximum delay from the actual bunch crossings to the L0 trigger decision arriving at the output of the L0 pipeline buffer in any sub−detector. For the muon trigger, this corresponds to a L0 pipeline buffer depth of 160, minus the time spent for processing the data in the front end electronics, the time to send data on the optical links, and the time to synchronise the data on the processing boards.

- **L0 derandomizer buffer:** At the reception of a *L0 trigger accep*t, the data at the output of the L0 pipeline buffer must be transferred into the L0 derandomizer buffer. The maximum transfer time cannot exceed **900 ns per event**. **The depth of the derandomizer must be bigger or equal to 16 events** to get an effective dead time below 1% at a 1 MHz trigger rate.

- **L1 buffer:** The L1 buffer must store event data, while the L1 trigger system determines which events to accept for further processing in the L2/L3 farm (DAQ). The L1 buffer has been defined to store at least 1820 events. Events will be written to the L1 buffer with a minimum event spacing of 900 ns, given by the L0 derandomizer readout time. Events will be read from the L1 buffer with a minimum event spacing of 900 ns for L1 accepts and 850 ns for L1 rejects. The slightly faster readout of the L1 buffer prevents saturation effects in the L1 buffer, when the L0 trigger runs at its maximum rate (1.11 MHz).

- **L1 derandomizer buffer:** At the reception of a *L1 trigger accept*, a data stored in the L1 buffer is transferred to the L1 derandomizer buffer. After the L1 derandomizer, event processing can be performed at a nearly constant rate. The maximum instantaneous input event rate is determined by the fact that each event fragment consists of a maximum of 34 words, extracted at 40 MHz rate. The output rate is given by the average L1 trigger rate, with an additional margin to insure an efficient derandomization. It has been determined that the L1 throttle network will have a maximum delay of 2 µs, including cable delays and the serialisation of TTC broadcast commands. With a minimum event spacing of 850 ns, the L1 throttle must be asserted 2 µs/850 ns = ~3 events

before the buffer actually runs full. This means that the effective derandomization size is given by the L1 derandomizer size minus three events. Extensive simulations have shown that a 16 event deep L1 derandomizer works very effectively and allows the readout to be performed at a rate only slightly faster than the average *accept* rate.

## 2.2 Processing board specific definitions

- **PU:** A Processing Unit is the optimum generic processing entity that can be contained in a single FPGA as shown in Figure 2. It contains the logic to receive the data coming from a tower through 6 optical links, and the logic to find and to select up to 2 candidates. It contains the L0 buffer and the L0 derandomizer buffer. It is also interfaced with the local ECS supervisor of the processing board.

- **BCSU:** The Best Candidate Selection Unit is a FPGA in charge of selecting the 2 best candidates among the 8 received from the four PUs and to send them to an Intermediate Candidate Selection Board. It contains a L0 buffer, a L0 derandomizer buffer and an interface to the local ECS supervisor of the processing board.

- **L1MU:** The L1 buffer Management Unit is a FPGA in charge of extracting the data from the L1 buffers, to derandomize these data and to send them to the Controller Board on 4 serial links. It contains also an interface to the local ECS supervisor of the processing board.

- **Local ECS supervisor:** Local intelligent module allowing a tight control and monitoring of the processing board operations. It is connected to the ECS network by an Ethernet link and to the FPGAs of the board by a local custom bus.

- **BCId:** The Bunch Crossing Identifier is the value of an internal counter used in the muon front−end electronics and within the PU. It tags the data in the pipe−lines of the processor. We expect BCId=0 at the beginning of each LHC

cycle.

- **FOI:** The Field of Interest is the group of pads in which we search the presence of a hit [1]. It is defined for stations M1, M2, M4 and M5.

- **PT:** Transverse momentum of the particle with respect to the beam axis.

## 2.3 Reset signals definitions

- **L0 front−end reset**: Reset the complete L0 front−end electronics including L0 buffers and L0 derandomizer buffers. It also resets the L0_B_Id. It occurs at the end of the large bunch gap so L0 buffers can be cleared with a minimum event loss. When issued for error recovery the readout supervisor will insure that the L0 derandomizer buffers are empty. The front−end electronics have to start with normal functionality within a few clock cycles after this reset. The first meaningful *L0 trigger accept* signal will only be issued after 160 clock cycles (L0 latency).
  *Frequency*: Depends on rate of synchronisation failures in system.

- **L1 front−end reset**: Reset the complete L1 front−end electronics, L1 buffers and all L1 derandomizer buffers. This reset will normally only be issued together with a L0 front−end reset (but not for all L0 resets). The L1 front−end electronics have to accept events from the L0 front−end electronics within 160 clock cycles after this reset ( L0 latency). The first L1 trigger will be sent to the front−end after 1 ms (~half L1 latency) to allow DSP based zero−suppression units to be ready to accept new event data.
  *Frequency*: Depends on rate of synchronisation failures in system.

- **Bunch count reset**: Reset of bunch ID counters for each LHC machine cycle.[3]. It occurs at a time such that correct Bunch ID's can be attached to event fragments at the input to the L0 derandomizer buffers. This means that the Bunch count reset signal will be received by the front−ends when event data related to bunch ID=160 is entering the L0 buffer (assuming L0 latency of 160 clock cycles)
  *Frequency*: Each 3564 clock cycle.

- **L0 Event count reset**: Reset of L0 event ID counters. Occurs at regular intervals and normally given together with Bunch count reset (but not for all Bunch resets).
  *Frequency*: At regular intervals.

- **L1 Event count reset:** Reset of L1 event ID counters.
  *Frequency*: At regular intervals.

In addition a specific reset is received from the controller board:

- **BC0 reset:** is equal to Bunch count reset – 160.

## 2.4 Service signal definition

**L0 Accept:** Signal active when the event data currently at the output of the L0 buffer is accepted by the Readout Supervisor.

**L1TriggerType[2..0]**: 3 bits indicating the type of L1 trigger, and whether it is accepted or not.

**L1TriggerTypeStrobe** : signal active when a L1 trigger type is broadcasted by the Controller Board.

**EventId[1..0]**: 2 bits corresponding to the LSB of the accepted or rejected event number.

**L1 Throttle:** A simple Hardwired L1 throttle signal is used to prevent buffer overflows in the L1 front−end electronics and the DAQ system. The L1 throttle delay has been determined to be below 2µs including cable delays and the serialisation of L1 trigger decisions for the TTC distribution system.

**Sync:** Synchronisation signal broadcasted to the Processing Boards by the Controller Board. On the rising edge of this signal, start/stop operations can be launched synchronously with other parts of the system.

**StopSendingL1DerandomContent:** signal broadcasted to the Processing Boards by the Controller Board . This signal prevents the dumping of the L1 derandomizer in case of overflow of the zero suppression derandomizer buffer.

**Signal detect:** signal active when a sufficient optical level is received by the optical transceivers.

# 3 General structure of the processing board

The processing board includes:

- an optical reception module receiving 24 individual data flows;
- 4 synchronisation and candidate search modules;
- 4 PT calculation modules;
- 1 candidate selection module;
- 1 L0 buffer distributed over 4 PUs and 1 BCSU;
- 1 L0 derandomizer distributed over 4 PUs and 1 BCSU;
- 1 L1 buffer;
- 1 L1 derandomizer;
- a formatting and serialisation module;
- an ECS interface module.

Large FPGA are used to implement most of the above functions. The proposed physical implementation is shown in Figure 2.

The mechanical format of the board is compliant with the 9U extended VMEbus mechanical standard.

In the next section, we describe general principles related to synchronisation and error recovery. In sections 5 to 11, we specify in detail each module of the processing board.

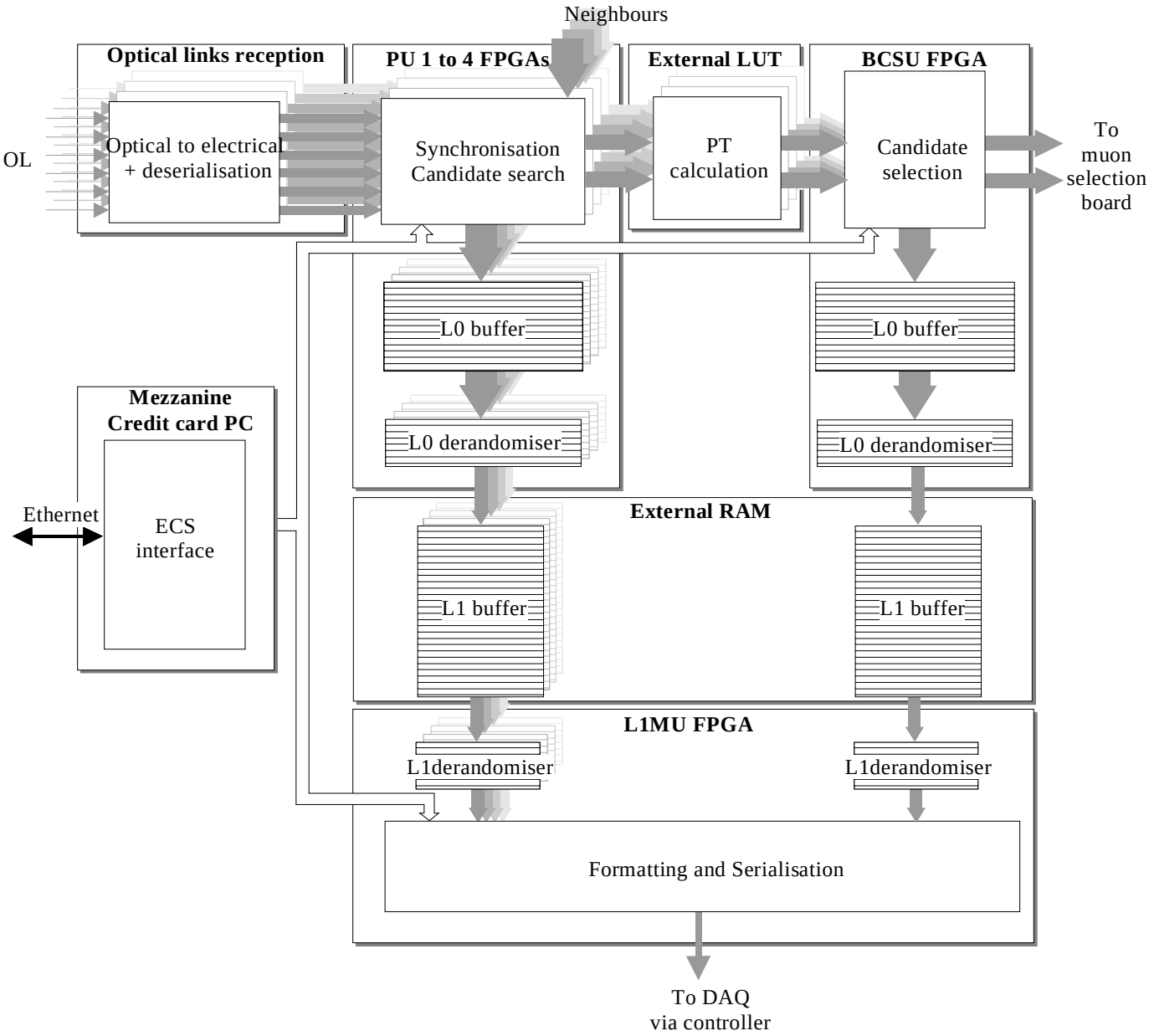



**Figure 2: Physical implementation**

# 4 General principles

## 4.1 Synchronisation of the data coming from the ODE

Data coming from stations M1 to M5 follow different paths. Thus it is mandatory to align them in time and to re−phase them with the system clock.

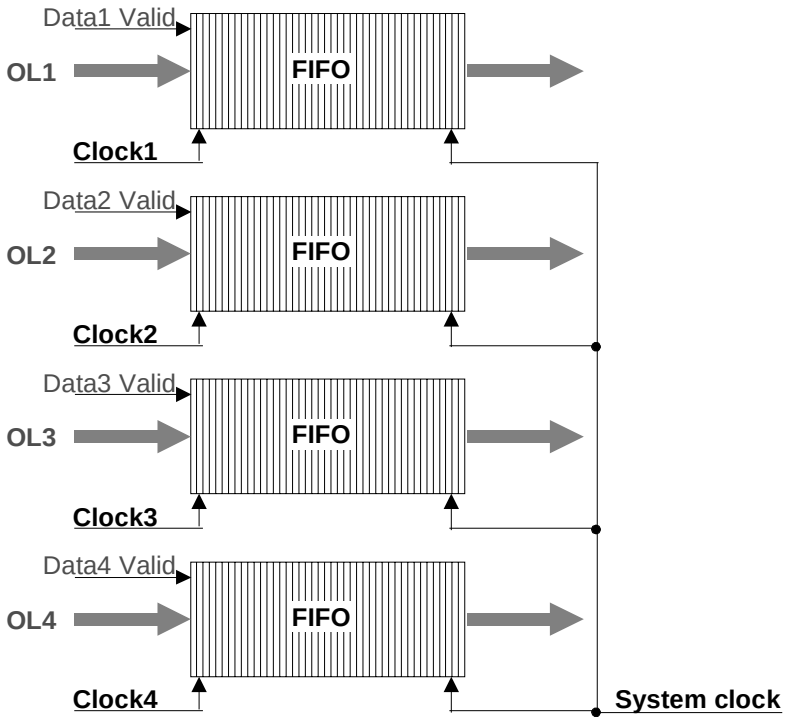



**Figure 3: Optical data synchronisation principle**

The principle of the synchronisation procedure is described in Figure 3:

- a FIFO is associated to each optical link;
- the clock coming with the data is used to write data words in the FIFO ;
- a waiting time allowing to catch all data with the same BCId in all FIFOs;
- all data are read from all the FIFOs with the global system clock.

To ensure perfect synchronisation, data travel with their clock but also with information related to the LHC cycle. The valid bit is used to signal a start of a new LHC cycle. In addition, the 2 LSB bits of the BCId are encoded in the data words.

The *data valid* information arriving with the data is used as a write signal for the FIFOs. After a *L0 front end reset*, this *data valid* information must be inactive up to the arrival of the first data of the next LHC cycle.

## 4.2 Synchronisation with the machine cycles

A machine cycle consists of 3564 Bunch Crossings [3].

On reception of the signal *Bunch counter reset*, we initialise the internal BCId counter and we reset all sequencers. This counter is incremented synchronously with the global 40 MHz clock. The BCId is attached to each event fragment stored in the L0 buffer. The initial value of the BCId counter is set in such a way that its image stored with the data when they enter in the L0 buffer is equal to the L0_B_Id when the data are extracted from the L0 buffer.

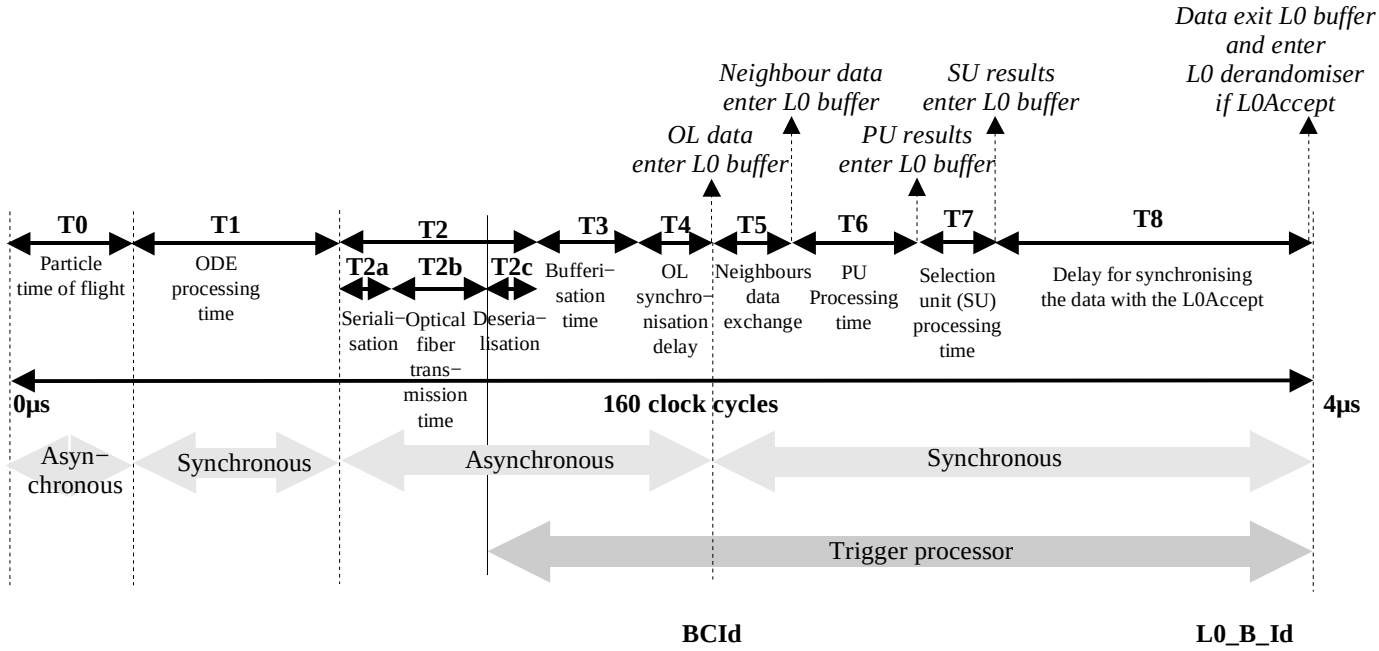



**Figure 4: Pipeline structure**

The travel time in the L0 buffer is 160 clocks cycles. The propagation time covers the front end electronics and the trigger processor processing times, as shown in Figure 4

The initial value of the BCId counter, for reaching 0 at the output of the L0 buffer, depends on the particle time of flight (T0), the ODE processing time (T1), the serialisation time (T2a), the optical fibre length (T2b), the optical link deserialisation time (T2c) and the delay between the different optical links (T4). These parameters are not well known. Thus the dimensioning is expressed in term of T1, T2, T3, T4, T5, T6 T7 and T8 assuming that:

T0+T1 +T2 +T3 + T4 + T5 +T6 + T7 + T8 = 160 * 25 ns = 4 µs

### 4.2.1 Time reference

The time reference is the output of the L0 buffer.

160 clock cycles are necessary to have a bunch crossing information get out of the

L0 buffer.

Consequently the L0_B_Id will have a value of −160 when the first bunch crossing occurs in order to have a value of 0 when the corresponding information exits the L0 buffer.

### 4.2.2 Current time estimations

The current estimations provided by the collaboration are indicated in Table 1 [6]

| Operation | Estimated time [ns] | Estimated numberof clock periods |
|---|---|---|
| Time of flight to M5 | 63 | |
| FE board processing | 70 | |
| Transmission to IB and ODE (15 m) | 105 | 13 |
| IB processing | 40 | |
| ODE processing | 30 | |
| Transmission to processing (100 m) | 600 | 24 |
| Muon processing | 1200 | 48 |
| Transmission to L0 decision Unit | 50 | 2 |
| L0 Decision Unit processing | 525 | 21 |
| L0 Decision Unit distribution | 800 | 32 |
| Contingency | 500 | 20 |
| **Total** | **3983** | **160** |

***Table 1: Time estimations for the trigger processing***

From this table and from dedicated simulations, we have estimated the values of the times T0 to T8 . They are summarised in Table 2. We have rounded each time interval to a fixed number of clock ticks.

| | Time | Clock |
|---|---|---|
| Particle time of flight | T0 | 3 |
| Muon front−end electronics processing time | T1 | 10 |
| Optical link serialization | T2a | 2 |
| Optical fiber transmission [90 m] | T2b | 18 |
| Optical link deserilization | T2c | 2 |
| Bufferization | T3 | 3 |
| Optical link synchronization | T4 | 3 |
| Data exchange between boards | T5 | 4 |
| PU processing time | T6 | 8 |
| Selection Units processing time | T7 | 5 |
| Delay to synchronize data with a L0 accept | T8 | 102 |
| **Total** | | **160** |

***Table 2: Estimation of the Time intervals T0 to T8***

### 4.2.3 Length of the L0 buffer for the muon trigger

The maximum length of the L0 muon trigger buffer is determined by the time between the synchronisation stage and the arrival time of the L0 accept decision (T5+T6+T7+T8 ). According to the above estimations, the maximum length is equal to 119 .

### 4.2.4 Synchronisation of the muon trigger with the L0_B_Id

T2 and T3 operations are not synchronous to the global clock. Indeed, each of the 6 optical reception modules operates with their own clock. The FIFO mechanism ,already described in page 17, is used to align the data in time and make them synchronous to L0_B_Id at the output of the L0 buffers.

All the synchronous operations starts at the end of interval of time T4.

### 4.2.5 Start/stop of the processing

The start / stop of the processing uses the following procedure:

- *enable processing* bits are set to 1 in all control registers of the FPGAs on all processing boards;
- the Controller Board broadcasts the signal *Sync* to all the boards of the system;

- the processing starts on the next occurrence of the *BC0 reset*.

## 4.3 Error recovery mechanisms

Various mechanisms are set up to detect data corruption which might happen in the optical link transmission.

When an error is detected, the data is systematically substituted with zeros to avoid propagation of the error in the downstream stages.

The most critical places where errors have to be detected and corrected are:

- the output of the optical link transmitters and deserialisers;
- demultiplexing stages;
- synchronisation module.

The types of error that we have to detect are:

- missing signal (broken or unplugged fibre);
- data corrupted during the transmission;
- demultiplexing error;
- BCId consistency error.

An error can generate a cascade of other errors in the downstream chain. Only the first error is signalled to minimise the number of bits required for encoding the errors. A 3 bits error code is attached to the final demultiplexed 32 bits word. The proposed encoding scheme is given in Table 3:

| EF_bit[2] | EF_bit[1] | EF_bit[0] | Value | Meaning |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | No error |
| 0 | 0 | 1 | 1 | Missing signal |
| 0 | 1 | 0 | 2 | Transmission error |
| 0 | 1 | 1 | 3 | Demultiplexing error |
| 1 | 0 | 0 | 4 | BCId consistency error |
| 1 | 0 | 1 | 5 | Reserved |
| 1 | 1 | 0 | 6 | Reserved |
| 1 | 1 | 1 | 7 | Reserved |

***Table 3: Encoding of errors per channel***

## 4.4 Test features

Our debugging strategy is based on the possibility to:

- Inject test patterns at the inputs of the main modules (PU, BSCU,...);
- Run the processor on these data;
- Read the input values seen by the modules and results computed by it at the output of the L0 derandomizer buffers;
- Apply an off−line simulation on the input values and to compare off−line results with those provided by the hardware.

### 4.4.1 Test of a PU

The data flow for test patterns is shown in Figure 5. We inject up to 16 consecutive test patterns before the optical link synchronisation buffers. This operation is performed through the local ECS supervisor. Then, the local ECS supervisor launches the processing. When it is done, 16 inputs and results are stored in L0 derandomizer buffers. In this mode the output of the L0 derandomizer buffer can be read by the local ECS supervisor.

A dedicated signal named *Sync* allows to synchronise the test on several PUs or several boards. In this framework, it is possible to steer global tests involving several PUs, with synchronised information exchanges between them.

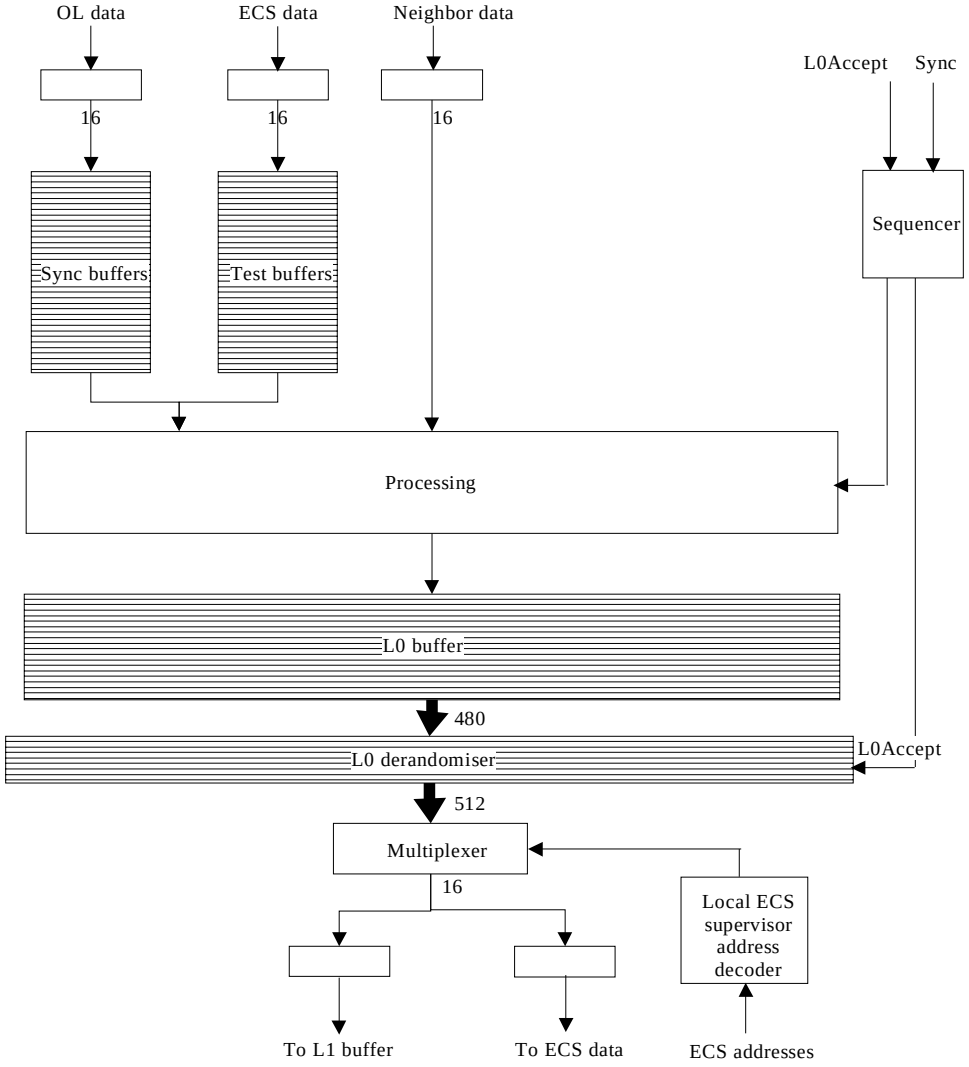



**Figure 5: Data stream of a PU in test mode**

## 4.4.2 Test of BCSU

The test of a BCSU is quite similar to the test of a PU. We inject up to 16 consecutive simulated candidates before the *best candidate search* module. This

operation is performed through the local ECS supervisor. Inputs and results are stored in the L0 derandomizer buffer of the BCSU. Its content can be read by the local ECS supervisor. A dedicated signal named *Sync* allows to synchronise the tests on several boards.

### 4.4.3 Test of L1MU

The test of the L1MU is steered conjointly with the Controller Board.

Patterns are directly written in the L1 derandomizer buffer by the local ECS supervisor. The control is made on the receiving end of the Controller board.

# 5 Specification of the optical reception

The optical links reception module is composed of:

- 2 parallel optical transceivers (ZARLINK, AGILENT, ...) equipped with a MPX or MPO connector;
- 24 deserialisers TLK2501 from Texas Instrument.

2 parallel ribbon cables contain 12 optical fibres each. They carry the optical information to the 2 optical transceivers. The transceivers convert 2x12 optical signals in 2x12 low voltage differential signals.

Each optical data channel coming from the optical transceivers is deserialised to 16 bits at 80 MHz by a TLK2501 chip from Texas Instrument. Deserialisers are grouped in 4 clusters of 6 TLK2501 chips. Each cluster feeds a PU. To minimise the path between the deserialisers and the PU, 3 TLK2501 are mounted on the component side of the board while the 3 remaining ones are mounted on the solder side.

## 5.1 Optical reception inputs

Each cluster of deserialisers receives respectively 6 differential serial links at 1.6 Gbits/s:

- 2 links from station M1;
- 1 link from station M2;
- 1 link from station M3;
- 1 link from station M4;
- 1 link from station M5.

## 5.2 Optical reception outputs

The TLK2501 outputs are:

- 16 bits of data at 80 MHz;
- 1 clock at 80 MHz extracted from the serial signal, synchronous with the data;
- 2 signals RX_DV and RX_ER providing the *data valid* flag and a diagnostic on the transmission as shown in Table 4.

**Reception**

| Encoded 10 bits output | RX_DV | RX_ER | Data received |
|---|---|---|---|
| IDLE (<K28.5,D5.6>, K28.5, D162> | 0 | 0 | **BCC5**h or **BC50**h |
| Carrier extend (K23.7) | 0 | 1 | **F7F7h** |
| Normal data character | 1 | 0 | Normal data character |
| Error | 1 | 1 | **FEFEh** if reception of an error propagation code |
| | | | **XXXXh** if invalid code |
| | | | **FFFFh** if loss of signal |

***Table 4: Diagnostic codes for the TLK2501***

# 6 Specification of the Processing Unit FPGA

The Processing Board contains 4 PUs.

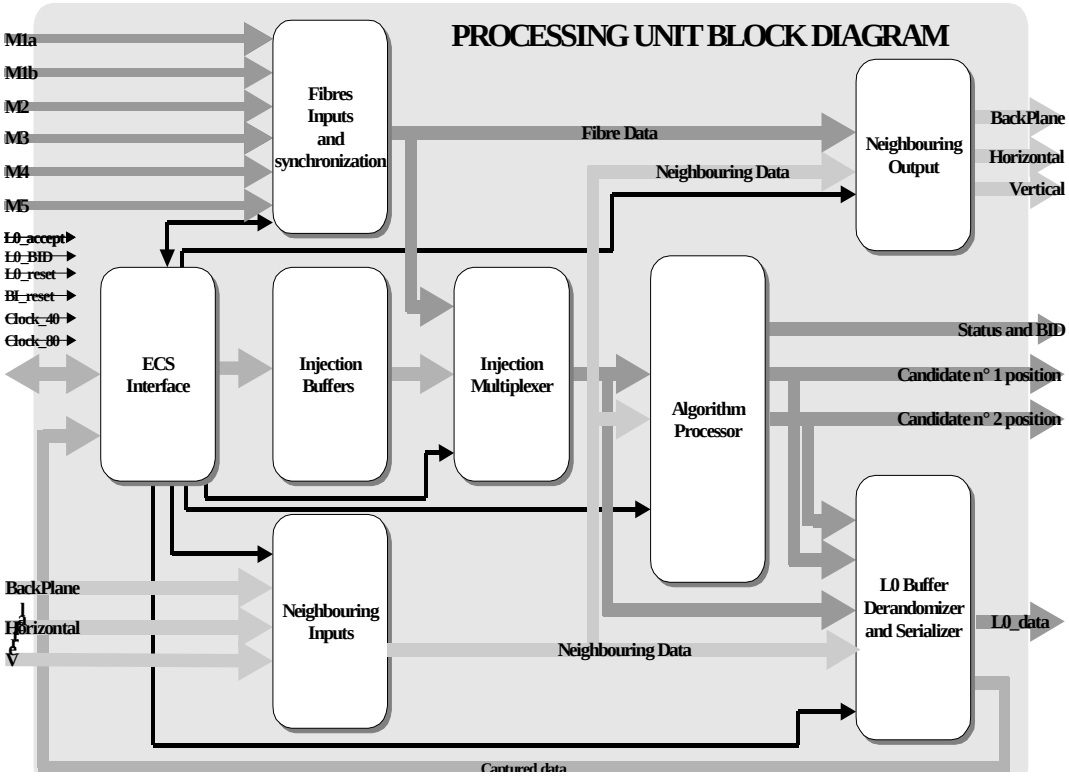



**Figure 6: Structure of a processing unit**

A processing unit is composed of 8 main entities as shown in Figure 6:

- a demultiplexing stage where deserialised data of one optical link are demultiplexed into 32 bits words at 40 MHz;
- a synchronisation stage where data from the 6 different optical links are aligned in time;
- a neighbouring exchange stage where data are exchanged between PUs ;
- a logic unit where all possible candidates are searched in parallel for the 96 pads of M3;
- a logic unit where a maximum number of 2 candidates are retained;

- a L0 buffer;
- a L0 derandomizer buffer;
- an interface with the local ECS supervisor.

In addition, error detection and error recovery mechanisms are included in critical entities like the demultiplexing and synchronisation stages.

## 6.1 PU inputs and outputs

A PU is connected to:

- optical transceivers;
- deserialised streams of the optical reception module;
- other PUs on the same board;
- other PUs from neighbouring boards;
- the PT calculation module;
- the BCSU module;
- the L1 buffer;
- the controller board;
- the local ECS supervisor.

### 6.1.1 Connection with the optical transceivers

The PU receives from the optical transceivers one or two *signal detect* information. For a Mitel transceiver, the *signal detect* information is common to 4 optical fibres. In this case, a PU receives 2 *signal detect* information since it is connected to 6 optical links. For a Zarlink transceiver, the *signal detect* information is common to 12 optical lines. In this case, a PU receives 1 *signal detect*.

### 6.1.2 Connection with the optical reception module

The PU receives from the optical reception module six 80 MHz data flows coming from the TLK2501. Each of them is composed of:

- 16 bits of data;
- an individual clock extracted from the incoming data;
- 2 signals RX_ER and RX_DV indicating the validity of the transfer.

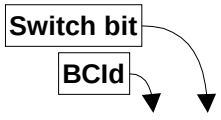
The frame of data words carried by each optical links is described in Table 5.

| **Bunch crossing number** | 0 | 1 | ..... | 3563 | 0 | 1 | ..... |
|---|---|---|---|---|---|---|---|
| **Data** | Data0 | Data1 | ..... | Data3563 | Data0 | Data1 | |

***Table 5: Structure of the received frame during a machine cycle***

It follows the cycle of the machine and contains 3564 data words. Notice that there are no header and trailer in the frame but the data words contain information on the bunch crossing identifiers. The width of the data words is 32 bits.

Each 32 bits data word is divided in 2 parts of 16 bits. The data word format is illustrated in Table 6:




| | D15 | D14 | D13 | D12 | D11 | D10 | D09 | D08 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1st word** | MSB information | | | | | | | | | | | | | | 0 | 1 |
| | LSB information | | | | | | | | | | | | | | 0 | 0 |
| **2nd word** | MSB information | | | | | | | | | | | | | | 0 | 1 |
| | LSB information | | | | | | | | | | | | | | 1 | 0 |
| **3rd word** | MSB information | | | | | | | | | | | | | | 1 | 1 |
| | LSB information | | | | | | | | | | | | | | 0 | 0 |
| **4th word** | MSB information | | | | | | | | | | | | | | 1 | 1 |
| | LSB information | | | | | | | | | | | | | | 1 | 0 |

***Table 6: Word format***

A “switch bit” distinguishes MSB from LSB part. It is placed on D00

Only the 2 LSB bits of the BCId are transmitted. They are sent on the bit D01 of 2 successive 16 bits words.

Bits D02 to D15 for MSB and LSB part contains 14 pads or strips data.

### 6.1.3 Connection with other PUs of the Processing Board

The data exchanges between PUs on the same board requires point to point busses running at 80 MHz. The quantity of information depends on the position of the PU and on the width of the FOI in X and in Y.

In the current estimations, the PU exchanges require:

- a 81 bits bus to exchange horizontal information (to + from);
- a 42 bits bus to exchange vertical information (to + from) in region R1 for station M1 and in region R2 for stations M2 and M3;
- a 2 bits bus to exchange crossing information (to + from).

  The format for the information is varying according to the location of the PU. See Annex 1 for detailed implementation.

### 6.1.4 Connection with PUs from neighbouring boards

The PU exchanges information with other PUs located on neighbouring boards. The data exchange is performed through point to point busses running at 80 MHz. They are implemented in a custom backplane. The quantity of information depends on the position of the PU and on the width of the FOI in X and in Y.

In the current estimations the PU sends to the backplane:

- a 41 bits bus to transmit horizontal information;
- a 3 bits bus to transmit vertical information;
- a 5 bits bus to transmit crossing information.

The PU receives from the backplane:

- a 41 bits bus to collect horizontal information;
- a 3 bits bus to collect vertical information;
- a 5 bits bus to collect crossing information.

The format for the neighbouring information is varying according to the location of the PU. See Annex 1 for detailed implementation.

### 6.1.5 Connection with the PT calculation module

The PU sends the information necessary to compute the PT for 2 candidates on a 30 bits bus at 40 MHz:

- two M3 address pad within the PU coded each on 7 bits;
- two M2 address within the FOI coded each on 4 bits;
- two M1 address within the FOI coded each on 4 bits.

### 6.1.6 Connection with the BCSU

The PU sends to the BCSU ( see Figure 15 page 62):

- a BCId on 4 bits;
- a status word on 4 bits.

and the address of the 2 candidates already described in the connection with the PT calculation module.

The PU receives from the BCSU a “Fill_LUT” signal. It is used to isolate the *address bus* when the BCSU fills the LUT of the PT calculation module with their values.

The format of the status word is described in Table 7. It contains an error word and a candidate word. The error word is described in Table 8 and the candidate word in Table 9.

| 3 | 2 | 1 | 0 |
|---|---|---|---|
| Error | | Candidates | |

***Table 7: Status word format***

| Error | | Meaning |
|---|---|---|
| 0 | 0 | No error |
| 0 | 1 | Transmission error on at least 1 of the 6 optical links |
| 1 | 0 | No signal on at least 1 optical line |
| 1 | 1 | BCId inconsistency on at least 1 of the 6 channels |

***Table 8: Meaning of the error bits of the status word***

| Candidates | | Meaning |
|---|---|---|
| 0 | 0 | No candidate |
| 0 | 1 | 1 candidate |
| 1 | 0 | 2 candidates |
| 1 | 1 | More than 2 candidates (PU overflow) |

***Table 9: Meaning of the candidate bits of the status word***

### 6.1.7 Connection with the L1 buffer

The PU sends to the L1 buffer:

- the content of the L0 derandomizer on a 16 bits bus at 40 MHz;
- a write signal for the L1 buffer.

For each event, the data output from the L0 derandomizer buffer is a frame containing 3 fields[1]:

1Due to the boundary of the bus, the size of each field has to be a multiple of 16.

- an identification field composed of 32 bits: Test Mode flag (1 bit), PU number within the board (2 bits), L0 event number (12 bits), L0_B_Id (12 bits), free (5 bits);
- an output field composed of 48 bits: Address candidate 1 (15 bits), Address candidate 2 (15 bits), Status word (4 bits), BCId (4 bits), free (10 bits);
- a input field composed of 432 bits: optical links inputs (168 bits), neighbouring inputs (224 bits), error detection word (24 bits), free (16 bits).

The muon event fragment consists of 512 bits. It is transmitted sequentially to the L1 buffer on 32 words of 16 bits.

The frame structure is summarised in Table 10:

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>Field</th></tr>
<tr><td>0</td><td>Test</td><td>–</td><td colspan="2">PUId</td><td colspan="11">L0Event number</td><td rowspan="2">Identification</td></tr>
<tr><td>1</td><td colspan="4">–</td><td colspan="11">L0_B_Id</td></tr>
<tr><td>2</td><td>–</td><td colspan="15">Address candidate 1</td><td rowspan="3">Outputs</td></tr>
<tr><td>3</td><td>–</td><td colspan="15">Address candidate 2</td></tr>
<tr><td>4</td><td colspan="8">–</td><td colspan="4">Status word</td><td colspan="4">BCId</td></tr>
<tr><td>5</td><td colspan="8">–</td><td colspan="8">Error flags</td><td rowspan="27">Inputs</td></tr>
<tr><td>6</td><td colspan="16">Error flags</td></tr>
<tr><td>7</td><td colspan="16" rowspan="25">Optical Inputs + neighbours</td></tr>
<tr><td>8</td></tr><tr><td>9</td></tr><tr><td>10</td></tr><tr><td>11</td></tr><tr><td>12</td></tr><tr><td>13</td></tr><tr><td>14</td></tr><tr><td>15</td></tr><tr><td>16</td></tr><tr><td>17</td></tr><tr><td>18</td></tr><tr><td>19</td></tr><tr><td>20</td></tr><tr><td>21</td></tr><tr><td>22</td></tr><tr><td>23</td></tr><tr><td>24</td></tr><tr><td>25</td></tr><tr><td>26</td></tr><tr><td>27</td></tr><tr><td>28</td></tr><tr><td>29</td></tr><tr><td>30</td></tr><tr><td>31</td></tr>
</table>

***Table 10: Frame structure of the information sent to the L1 buffer***

### 6.1.8 Connection with the controller board

The PU receives from the controller board:

- a master clock at 40 MHz;
- a master clock at 80 MHz;
- a L0_B_Id number on 12 bits at 40 MHz;
- a L0 Accept signal;
- a BC0 reset signal;
- a Bunch count reset signal;
- a L0 front−end Reset signal;
- a L0 Event Counter Reset signal;
- a Sync signal;

### 6.1.9 Connection with the local ECS supervisor

The PU is connected to the local ECS supervisor by:

- a bidirectional 16 bit bus operating at 40 MHz;
- 1 interrupt signal;
- 7 bidirectional control signals.

### 6.1.10 Output for debug

A PU can be connected to a logical analyser through a dedicated 16 bits bus + 1 clock.

## 6.2 PU optical input stages and associated error detection

On each TLK2501 optical receiver, the data arrive with their own clock. This clock samples the data. In addition, data arrive with 2 signals indicating the validity of the transfer: RX_ER and RX_DV. Their meaning is given in Table 4. By analysing the value of the bits RX_ER and RX_DV, and the value of the *signal detect* coming from the optical transceivers it is possible to detect several cases of error as seen in Table 11.

| Error case | Detection | Action |
|---|---|---|
| Cut or unplugged fibre | *Signal detect* = 0 | Data=0000h, EF[2..0] = 1 |
| Transmission error | RX_DV=1, RX_ER=1 | Data=0000h, EF[2..0] = 2 |

***Table 11: Errors detection on PU input stages***

## 6.3 Demultiplexing stage and associated error detection

The deserialised output data arrive in 16 bits words at 80 MHz. They have to be demultiplexed to form a 32 bit word at 40 MHz.

The demultiplexing operation is synchronous with the clock extracted from the data by the TLK2501.

The sequence of the 16 bits words is always the same. The MSB part of the 32 bits word is followed by the LSB part.

A "switch bit" is included in the data to allow error detection in the demultiplexing phase. It is equal to zero for a LSB word and to one for a MSB word.

Important note: T*his "switch bit" must not be used to command the demultiplexer operation since the "switch bit" can be corrupted. Instead a state machine steers the operation. The switch bit is only there to allow an error detection mechanism to check that the demultiplexing operation works correctly.*

By comparing the expected value of the “switch bit” with the received one it is possible to check that the demultiplexing state machine works properly.

| Error case | Detection | Action |
|---|---|---|
| Demux state machine in wrong state | Comparison between received “switch bit” and state machine “switch bit” | Data=0000h, EF[2..0] = 3 |
| Corrupted received “switch bit” | | |

***Table 12: Error detection for demultiplexing***

The outputs of each demultiplexing stage are:

- a 32 bits word synchronised on the clock provided by the optical receiver stage;
- a 3 bits error code summarising the most upstream error which occurred on the channel (see Table 3 page 23).

## 6.4 Synchronisation stage and associated error detection

The data output by different optical receiver stages are not in phase. They have to be time aligned by using a FIFO or dual port memory (see 4.1 page 17).

Demultiplexed data coming from the optical link receivers are stored in FIFOs or dual port memories synchronously to their accompanying clock. They are read back synchronously to the global 40 MHz clock at the end of the time interval T4.

| Error case | Detection | Action |
|---|---|---|
| Corrupted BCId | Received BCId different from internal BCId LSB | Data=0000h, EF[2..0] = 4 |
| Desynchronised frame | | |

***Table 13: Error detection for synchronisation***

Error detection is based on the comparison of the incoming BCId stored in the incoming data with the internal BCId generated in the PU. This operation is performed before entering the synchronisation FIFOs. The errors and their corresponding actions are shown in Table 13.

Note : *if an error has been found either in the optical input, the demultiplexing stage or*

*the synchronisation stage, an error bit is set in the status word accompanying the found candidates for the event (see Table 7 page 33).*

## 6.5 Neighbouring exchanges

Neighbouring data coming from other PUs are received synchronously to a 80 MHz clock. They are exactly in phase with the global 40 MHz system clock.

Neighbouring information reach their destination 4 or 5 clock cycles after the end of T4 since data have to be multiplexed and buffered before the transfer on the backplane, as shown in Figure 7. Thus, it is necessary to compensate the travel time of the neighbours on the backplane to have all the information relative to a same event aligned in time before starting the processing. This compensation is achieved by buffering the optical lines data within the PUs.

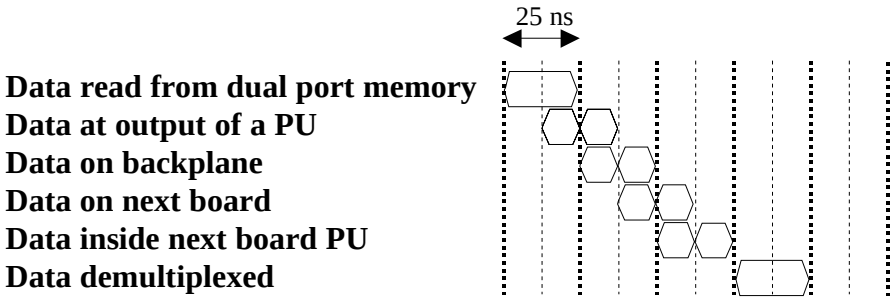



***Figure 7: Pipe−lined 80 MHz transfer on backplane***

***Due to the bufferisation, transfers on the backplane cannot be done within a single 80 MHz clock cycle without timing violation. It is necessary to pipeline the transfer by using the latching capabilities of the external buffers to respect the set−up requirements.***

Neighbouring exchanges within a board are similar with the difference that there are direct connections (without buffer) between the PUs at 80 MHz.

Neighbouring information reaches its destination 3 clock cycles after the end of T4 since data have to be multiplexed before the transfer on the board, as shown in Figure 8.

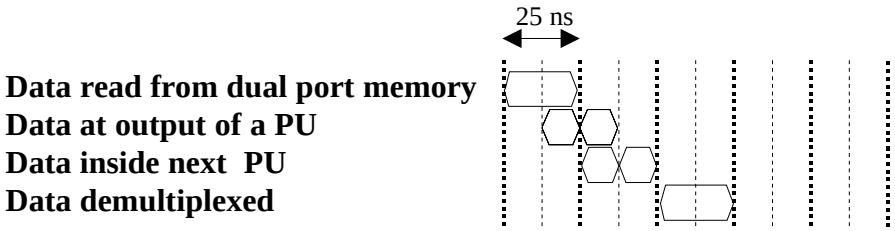



***Figure 8: Pipe−lined 80 MHz transfer on a board***

No synchronisation check is performed on the neighbouring information since they are in phase with the global clock.

The information available at the end of this stage is:

- all optical and neighbouring data synchronised;
- the BCId corresponding to the event;
- 6 error flags coded on 3 bits summarising the most upstream error which occurred on the 6 optical channels.

## 6.6 Candidates search in M2, M3, M4 & M5

Note: *From that stage, error detection and correction are not possible anymore. All operations being synchronous and remaining inside the FPGAs of the board, the risk of failure is extremely low.*

The candidate search is initialised from pads in station M3. A Processing Unit handles 96 pads. The search algorithm is executed on the 96 pads in parallel.

Data sent by the muon detector are pads or strips. Strips data have to be translated into pads before launching the muon search. Such an homogenisation simplifies the search algorithm since it works on an isomorphous space.

Data formatting has also to be applied for pad belonging to different regions. It is carried out as follows:

- from internal to external, pads and strips information are grouped before being transmitted to minimise the quantity of exchanged information. See example in Figure 9.

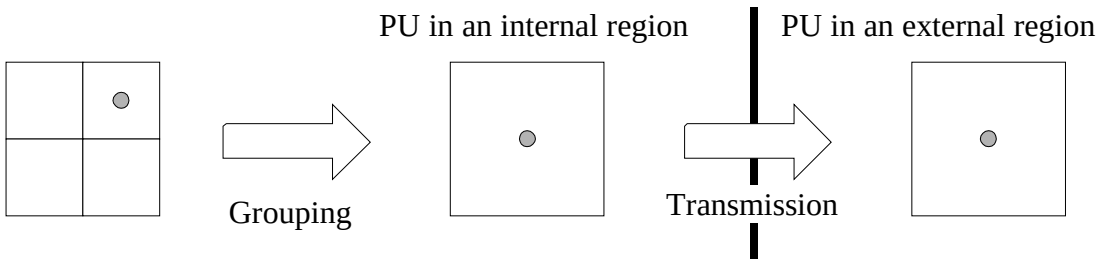



**Figure 9: Internal to external regions data formatting**

- From external to internal, pads and strip information are replicated after transmission to minimise again the quantity of exchanged information. See example in Figure 10.

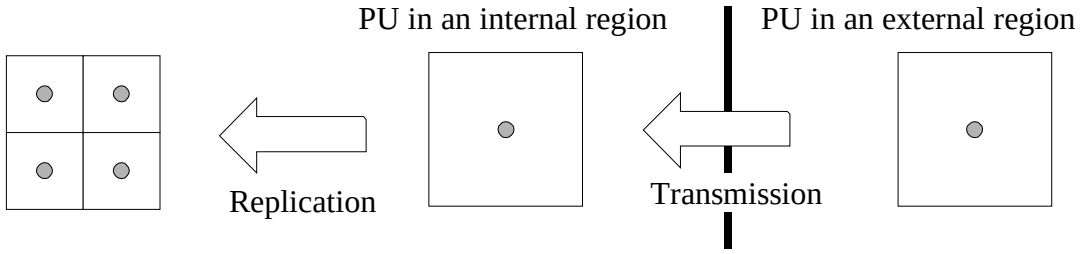



**Figure 10: External to internal region data formatting**

The size of the FOI in stations M2, M4 and M5 is individually programmable through the ECS interface (see paragraph 6.11.4.1).

The candidate search module receives:

- the 96 pad information relative to M3;
- all the pad information relative to M2, M4 and M5 extended with a quantity of neighbouring information from adjacent PUs. The latter is required to run the algorithm on the frontiers of the PU. The number of neighbouring pads depends on the PU position and the size of the FOIs.

The candidate search module provides 96 candidates values including the following information:

- candidate found on 1 bit;
- address in M3 on 7 bits;
- address in M2 on 4 bits.

## 6.7 Extrapolation to M1

An extrapolated position to M1 is computed from the M3 and M2 positions for the 96 candidates. The closest hit found in a FOI centred on the extrapolated position is retained as the M1 hit.

If no hit is found the candidate is not validated.

The size of the search window in M1 is programmable through the ECS interface (see paragraph 3.6.6).

## 6.8 Candidates selection

At most 2 candidates are retained per PU.

If more than 2 candidates are found, the M3 and M2 addresses of the 2 candidates that are found first in the scanning process are transmitted to the PT computation and BCSU stages with the BCId and a status word signalling a *PU overflow* condition (see Table 7 page 33).

If 2 candidates are found their M3 and M2 addresses are transmitted to the PT computation stage with the BCId and a status word signalling that 2 candidates have been found.

If only 1 candidate is found, its M3 and M2 address is transmitted to the PT computation and BCSU stages with the BCId and a status word signalling that only 1 candidate has been found. In this case, the corresponding address field for the second candidate is filled with zeros.

If no candidate is found the address are set to 0 and the status word signals that no candidate has been found.

## 6.9 L0 Buffer

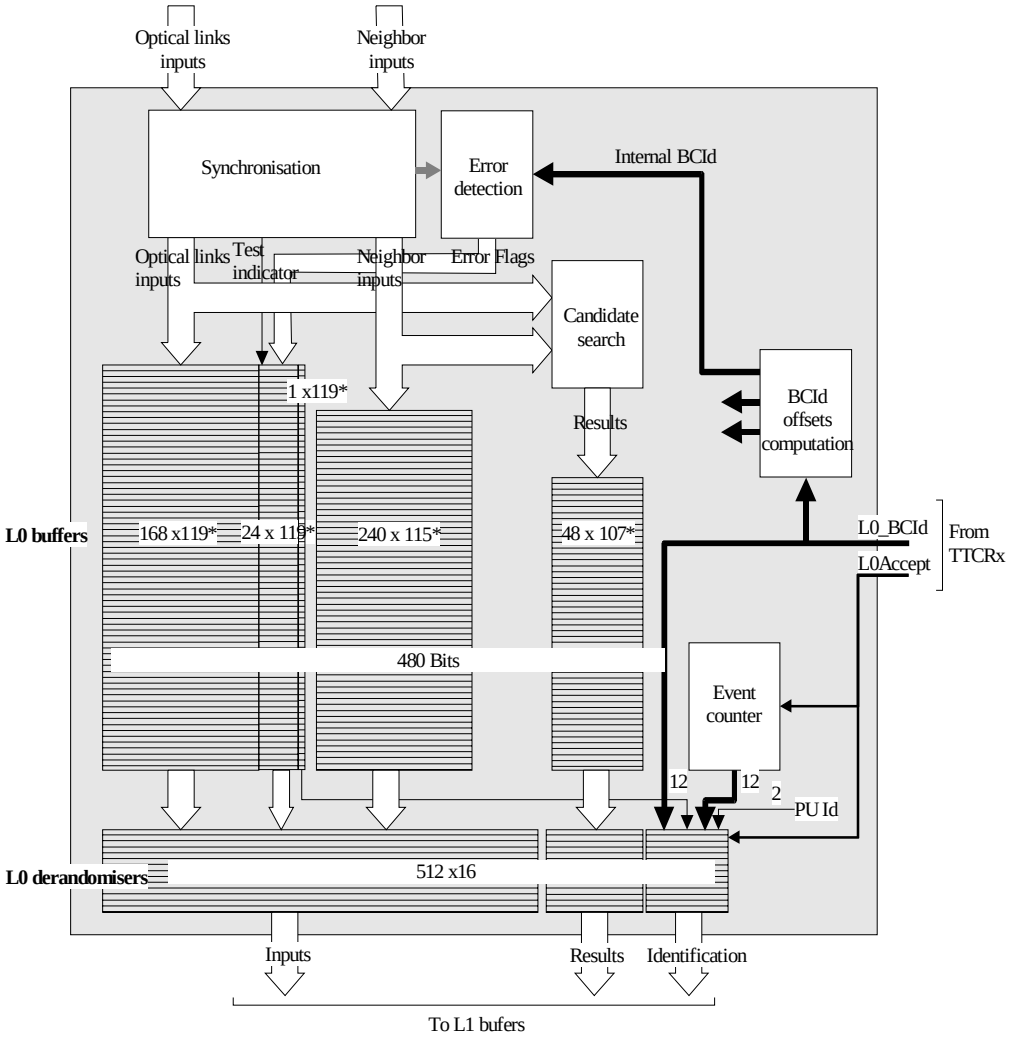
Data written in the L0 buffer are:

- the optical link information at the end of time interval T4;
- the neighbouring information at the end of time interval T5;
- the candidate information at the end of time interval T6 .

The L0 buffer is actually composed of 3 elementary buffers as shown in Figure 11:

- the OL buffer and error detection word:
  - depth = $\{(T5 + T6 + T7 + T8)_{(\text{in ns})} / 25_{ns}\}$ rows;
  - width = 192 bits.
- the neighbouring buffer + free bits:
  - depth = $\{(T6 + T7 + T8)_{(\text{in ns})} / 25_{ns}\}$ rows;
  - width = 240 bits;
- the candidates buffer + free bits:
  - depth = $\{(T7 + T8)_{(\text{in ns})} / 25_{ns}\}$ rows;
  - width = 48 bits.

The 3 buffers are read simultaneously when a *L0Accept* signal is received.




* size based on current timing estimations of processing time for both the FE and the trigger

**Figure 11: L0 buffer and derandomizer buffer for a Processing Unit**

## 6.10 L0 derandomizer buffer

All the data coming from the L0 buffer are transmitted to the L0 derandomizer buffer **in one clock cycle** on every occurrence of a L0Accept signal. The L0_B_Id and a L0 event number generated by an internal counter are also entered in the L0 derandomizer.

The depth of the L0 derandomizer buffer is fixed to 16 events. Its width is 512 bits.

### 6.10.1 Link to the L1 Buffer

The L0 derandomizer content is sent to the L1 buffer as follows .

The data path to the L1 buffer is 16 bits. Therefore, it is necessary to multiplex the data of the L0 derandomizer before sending them to the L1 buffer.

One event fragment has a length of 512 bits (See frame format in § 6.1.7). Thus 32 cycles are necessary to transmit one event fragment to the L1 buffer. The time required to extract the data of one event is 800 ns, in agreement with the L1 specification.

As long as the L0 derandomizer buffer is not empty, it emits a 16 bits word to the L1 buffer with a write signal on every clock cycle. If the L0 derandomizer buffer contains several events, they are transmitted consecutively.

## 6.11 Interface with the local ECS supervisor

The control of the PU and the reading of its status are available through the ECS interface. This is implemented with a set of registers allowing to configure programmable features or to read information within the PU. This set of register can be read / written by the local ECS supervisor.

Two modes of operation are allowed, each being exclusive. Their functionalities use specific or common resources :

- Features used in test mode:

– Parameters initialisation;
– Pattern injection;
– Capture of up to 16 consecutive events;
– Masking of neighbouring data, optical links, etc...;
– Local resets.

- Features used in operational mode:
  – Parameters initialisation;
  – Access to error counters and flags;
  – Capture of an event;
  – Masking of neighbouring data, optical links, etc ...;
  – Local resets.

A special register allowing to choose the mode of operation and control the processing:

– Control of operations.

### 6.11.1 PU operation control and status registers

A 16 bits register allows to chose the mode of operation and in particular to start / stop the processing. It is described in Table 14.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | Reset operational mode event capture FIFOs | Rest test mode event capture FIFOs | Reset pattern injection FIFOs | Enable event capture | Enable processing | *R/W* |
| – | | | | | | | | | | | | Event(s)Ready | Operational mode event capture FIFO empty | Test mode event capture FIFO empty | Pattern injection FIFO full | *R* |

***Table 14: Control and status registers***

**Mode**: writing a 1 chooses the operational mode, writing a 0 chooses the test mode.
**Enable processing:**
- In operational mode : writing a 1 starts the processing of data after the reception of the *BC0 reset* which follows the reception of the *Sync* signal; writing a 0 stops the processing of data in the same way.
- In test mode : when this value is set to 1, the processing is started on reception of the next *Sync* signal. The process is automatically stopped after

transmission of 16 consecutive events (the Start Event Capture bit is therefore automatically reset).

**Enable Event Capture:**

- In operational mode only : when the bit is set to 1, the next event qualified by a L0Accept condition is captured. The bit is automatically set to 0 after the capture.

**Reset Pattern Injection FIFO:** writing a 1 resets the pattern injection FIFO. Automatically set to 0 when the operation is completed.

**Pattern injection FIFO full:** set to 1 when the pattern injection is full. Else : set to 0

**Reset Test mode Event Capture FIFO (L0 derandomizer):** writing a 1 resets the test mode event capture FIFO. Automatically set to 0 when the operation is completed. The reset is allowed in test mode exclusively.

**Test mode Event Capture FIFO empty:** set to 1 when the test mode event capture FIFO is empty. Else : set to 0

**Reset Operational mode Event Capture FIFO:** writing a 1 resets the operational mode event capture FIFO. Automatically set to 0 when the operation is completed

**Operational mode Event Capture FIFO empty:** set to 1 when the operational mode event capture FIFO is empty. Else : set to 0

**Event(s) Ready**: set to 1 when the data are captured (test or operational mode).

### 6.11.2 Registers used in test mode

The strategy to debug a processing board or a processor is the following:

1. Set the tests mode using the ECS interface.
2. The ECS fills specific test buffers with patterns. They emulate data transport by the optical links for 16 consecutive events. The test buffers inject data in place of the buffers receiving the output of the optical transceiver stage.
3. When the test is launched, the test buffers are substituted to the reception buffers for feeding the processing stages. Then the trigger runs on 16 consecutive events and stops.
4. Input values provided by the test buffers and neighbouring data from adjacent PUs are logged to the L0 buffer. They are systematically transferred

to the L0 derandomizer buffer.

5. The ECS reads the content of the L0 buffers and compares the 16 results with the theoretical values by using a simulation of the PU algorithm.

Note: *Several scenarios are imaginable for running these tests. The processing capabilities of the local ECS supervisor can be used or, alternatively, the work can be done by the ECS control PC. If the local ECS supervisor is used, it can either run the simulation algorithm itself or more simply compare the outputs of the processor with pre−computed results.*

We will now describe the pattern injection and the capture of consecutive events.

### 6.11.2.1 PU pattern injection registers

The test buffers can be filled with data simulating the optical links inputs for up to 16 consecutive events. These buffers are substituted to the reception buffers used in operational mode. Therefore, simulated data must also contain a simulated error field in addition to the pad / strip information.

Note 1: *In spite of the fact that the synchronisation buffers can be filled with erroneous words, it is however not possible to simulate the behaviour of the error detection mechanisms because the synchronisation buffers are located afterwards.*

Note 2: *Only the optical link data can be simulated. It is however possible to inject neighbouring data by starting simultaneously all the neighbouring PUs in test mode. This is possible because of the special Sync signal that is broadcasted to all FPGAs of the system (see Figure 12). If a single PU is tested, it might be necessary to mask the neighbouring inputs of the FPGA. This is obtained by isolating the neighbouring data (see "link isolation" in next paragraph)*

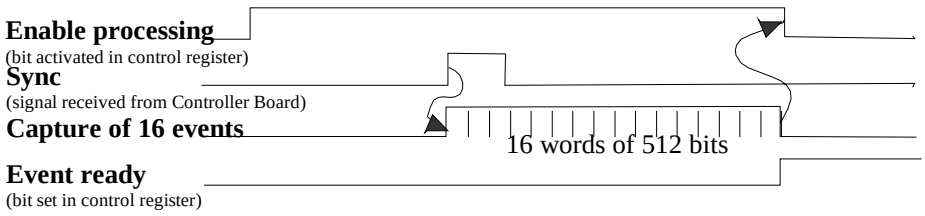



**Figure 12: Synchronisation of the tests with the signal Sync**

The content of the test buffers is written in a set of registers described in Table 15. This set of registers contains all information for one event. The set of register is the interface between the test buffers and the local ECS supervisor. In that framework, the test buffers act as a FIFO of events: an event is downloaded in the FIFO when the last register is filled.
The control register (see Table 14 page 45) allows to reset the FIFOs or to check whether they are full.

Note: *All the registers are Write Only. A read back possibility is provided by reading the L0 derandomizer buffer.*

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OL information from M1 | | | | | | | | | | | | | | | | *W* |
| OL information from M1 | | | | | | | | | | | | | | | | *W* |
| − | | | | | | | | | | | Error word for M1 | | | | | *W* |
| OL information from M1 | | | | | | | | | | | | | | | | *W* |
| OL information from M1 | | | | | | | | | | | | | | | | *W* |
| − | | | | | | | | | | | Error word for M1 | | | | | *W* |
| OL information from M2 | | | | | | | | | | | | | | | | *W* |
| OL information fromM2 | | | | | | | | | | | | | | | | *W* |
| − | | | | | | | | | | | Error word for M2 | | | | | *W* |
| OL information from M3 | | | | | | | | | | | | | | | | *W* |
| OL information from M3 | | | | | | | | | | | | | | | | *W* |
| − | | | | | | | | | | | Error word for M3 | | | | | *W* |
| OL information from M4 | | | | | | | | | | | | | | | | *W* |
| OL information from M4 | | | | | | | | | | | | | | | | *W* |
| − | | | | | | | | | | | Error word for M4 | | | | | *W* |
| OL information from M5 | | | | | | | | | | | | | | | | *W* |
| OL information from M5 | | | | | | | | | | | | | | | | *W* |
| − | | | | | | | | | | | Error word for M5 | | | | | *W* |

***Table 15: Test pattern injection registers***

**OL information from Mi**: optical link information from station Mi .
**Error word for Mi**: error word for a demultiplexed 32 bits word coming from optical link from station Mi.

The detail of the error word is described in Table 16.

| 3 | 2 | 1 | 0 | |
|---|---|---|---|---|
| BCId inconsistency | Demultiplexing error | Transmission error | Signal detect non present | *W* |

***Table 16: Error word detail***

#### 6.11.2.2 PU consecutive events capture registers in test mode

The local ECS supervisor can read the content of the L0 derandomizer buffer when the processor is running in test mode. Fake *L0accept* signals are generated to fill automatically the L0 derandomizer buffer with the values of 16 consecutive events[2]. The output of the derandomizer buffer is then multiplexed on a 16 bits path connected to the ECS interface as shown in Figure 5 page 24.

When the *event ready* flag of the control register is raised (see § 6.11.1), the event has been captured and the page described in Table 17 contains the captured event:

It is then possible to read:
- the OL information;
- the neighbouring data;
- the results of the candidate search.

The derandomizer buffer is implemented in a 512 bits wide FIFO. The page maps the output of this FIFO and therefore contains a complete event. The page is subdivided in 32 words of 16 bits. It is automatically refreshed with the next event when the last word is read.

The control register (see Table 14 page 45) allows to reset the derandomizer FIFO or to check whether it is empty.

2 Although in operational mode the L0 derandomizer only contains a maximum of 16 events, the internal structure of the FPGA obliges to occupy 128 positions. Therefore it should be possible to store 128 consecutive events in test mode at no cost.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word 0 | | | | | | | | | | | | | | | | *R* |
| Word 1 | | | | | | | | | | | | | | | | *R* |
| Word 2 | | | | | | | | | | | | | | | | *R* |
| Word 3 | | | | | | | | | | | | | | | | *R* |
| Word 4 | | | | | | | | | | | | | | | | *R* |
| Word 5 | | | | | | | | | | | | | | | | *R* |
| Word 6 | | | | | | | | | | | | | | | | *R* |
| Word 7 | | | | | | | | | | | | | | | | *R* |
| Word 8 | | | | | | | | | | | | | | | | *R* |
| Word 9 | | | | | | | | | | | | | | | | *R* |
| Word 10 | | | | | | | | | | | | | | | | *R* |
| Word 11 | | | | | | | | | | | | | | | | *R* |
| Word 12 | | | | | | | | | | | | | | | | *R* |
| Word 13 | | | | | | | | | | | | | | | | *R* |
| Word 14 | | | | | | | | | | | | | | | | *R* |
| Word 15 | | | | | | | | | | | | | | | | *R* |
| Word 16 | | | | | | | | | | | | | | | | *R* |
| Word 17 | | | | | | | | | | | | | | | | *R* |
| Word 18 | | | | | | | | | | | | | | | | *R* |
| Word 19 | | | | | | | | | | | | | | | | *R* |
| Word 20 | | | | | | | | | | | | | | | | *R* |
| Word 21 | | | | | | | | | | | | | | | | *R* |
| Word 22 | | | | | | | | | | | | | | | | *R* |
| Word 23 | | | | | | | | | | | | | | | | *R* |
| Word 24 | | | | | | | | | | | | | | | | *R* |
| Word 25 | | | | | | | | | | | | | | | | *R* |
| Word 26 | | | | | | | | | | | | | | | | *R* |
| Word 27 | | | | | | | | | | | | | | | | *R* |
| Word 28 | | | | | | | | | | | | | | | | *R* |
| Word 29 | | | | | | | | | | | | | | | | *R* |
| Word 30 | | | | | | | | | | | | | | | | *R* |
| Word 31 | | | | | | | | | | | | | | | | *R* |

***Table 17: Event capture registers for test mode***

**Word i:** same structure as frame to L1 (see § 6.1.7)

### 6.11.3 Registers used in operational mode

The processing is launched in the following way:

1. the local ECS supervisor validates the *enable processing* bit in the control

register (see § 6.11.1);

2. the PU waits for the occurrence of the signal *Sync* allowing to synchronise the capture on several PUs of the full processor.

3. the processing starts after the next reception of the next BC0 reset.

Operations are stopped in the same way: when the *enable processing* bit is reset the processing of new data is stopped after the reception of the next BC0 reset after the *Sync* signal.

Note: *This does not mean that all the operations are stopped at this time: the processing continues until all the data currently in the processor pipe have exited.*

In this mode it is possible to capture an event to check its consistency. We retain the following mechanism:

1. the local ECS supervisor validates the *event capture enable* bit in the control register;

2. the PU waits for the occurrence of the signal *Sync* allowing to synchronise the capture on several PUs of the full processor.

3. The PU captures the next event fragment for which a L0 accept has been received;

4. The event fragment is stored in an *event capture FIFO* which can be read by the local ECS supervisor.

#### 6.11.3.1 PU error detection flags and counter registers

Error detection mechanisms are implemented in the PUs (see § 4.3). When a fault is detected an error counter is incremented and some flags are set up. Both error counters and the sets of flags can be read by the local ECS supervisor.
The description of the registers containing the error counters and the flags are given in Table 18. A group of registers is assigned to each of the 6 receiving channels.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | of signal 1 | CE 1 | Trans 1 Err | Dem 1 Err | BCId 1 Err | *R* |
| Transmission 1 error count | | | | | | | | | | | | | | | | *R* |
| Demux 1 error count | | | | | | | | | | | | | | | | *R* |
| Synchronisation 1 error count | | | | | | | | | | | | | | | | *R* |
| – | | | | | | | | | | | of signal 2 | CE 2 | Trans 2 Err | Dem 2 Err | BCId 2 Err | *R* |
| Transmission 2 error count | | | | | | | | | | | | | | | | *R* |
| Demux 2 error count | | | | | | | | | | | | | | | | *R* |
| Synchronisation 2 error count | | | | | | | | | | | | | | | | *R* |
| – | | | | | | | | | | | of signal 3 | CE 3 | Trans 3 Err | Dem 3 Err | BCId 3 Err | *R* |
| Transmission 3 error count | | | | | | | | | | | | | | | | *R* |
| Demux 3 error count | | | | | | | | | | | | | | | | *R* |
| Synchronisation 3 error count | | | | | | | | | | | | | | | | *R* |
| – | | | | | | | | | | | of signal 4 | CE 4 | Trans 4 Err | Dem 4 Err | BCId 4 Err | *R* |
| Transmission 4 error count | | | | | | | | | | | | | | | | *R* |
| Demux 4 error count | | | | | | | | | | | | | | | | *R* |
| Synchronisation 4 error count | | | | | | | | | | | | | | | | *R* |
| – | | | | | | | | | | | of signal 5 | CE 5 | Trans 5 Err | Dem 5 Err | BCId 5 Err | *R* |
| Transmission 5 error count | | | | | | | | | | | | | | | | *R* |
| Demux 5 error count | | | | | | | | | | | | | | | | *R* |
| Synchronisation error count | | | | | | | | | | | | | | | | *R* |
| – | | | | | | | | | | | of signal 6 | CE 6 | Trans 6 Err | Dem 6 Err | BCId 6 Err | *R* |
| Transmission 6 error count | | | | | | | | | | | | | | | | *R* |
| Demux 6 error count | | | | | | | | | | | | | | | | *R* |
| Synchronisation 6 error count | | | | | | | | | | | | | | | | *R* |

***Table 18: Error flags and error counters registers***

**Loss of signal i:** Image of the *signal detect* = 0 condition occuring when the optical level on the optical transceiver is insufficient on channel i. Set to 1 when the condition is detected. Set to 0 in normal operation.
**CE i**: Channel error. Set to 1 on rising edge of the "Loss of signal" bit . Automatically reset when the register is read.

**Transmission i Err**: set to 1 when a transmission error has been detected by the TLK2500. Automatically reset when the register is read.
**Transmission i error count:** number of errors detected by the TLK2500 since last reading. Automatically reset when the register is read.

**Dem i Err**: set to 1 when a demultiplexing error has occurred. Automatically reset when the register is read.
**Demux i count:** number of demultiplexing made on the wrong side since last reading. Automatically reset when the register is read.

**BCId i Err**: Synchronisation error. Set to 1 when a corrupted BCId is detected. Automatically reset when the register is read or on next occurrence of L0_Reset .
**Sync i error count**: number of desynchronised frames since last reading. Automatically reset when the register is read.

### 6.11.3.2 PU event capture register in operational mode

When an event capture is launched, captured data for one event are accessible by the local ECS supervisor.

The event capture is enabled by writing a 1 in the "enable event capture" slot of the control register. The next L0Accept signal following the first rising edge of the Sync signal is used to latch the data as described in Figure 13 . Waiting for the Sync signal allows to synchronise the event capture among several FPGAs or among several cards.

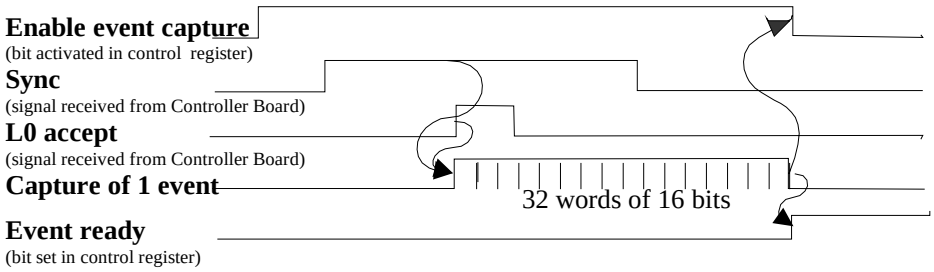



**Figure 13: Start of the event capture on first L0 Accept following the rising edge of Sync**

The output of the derandomizer buffer is then serialised on a 16 bits path and stored in the *event capture FIFO* as shown in Figure 14. When the data are available, the *event ready* bit is set in the control register.

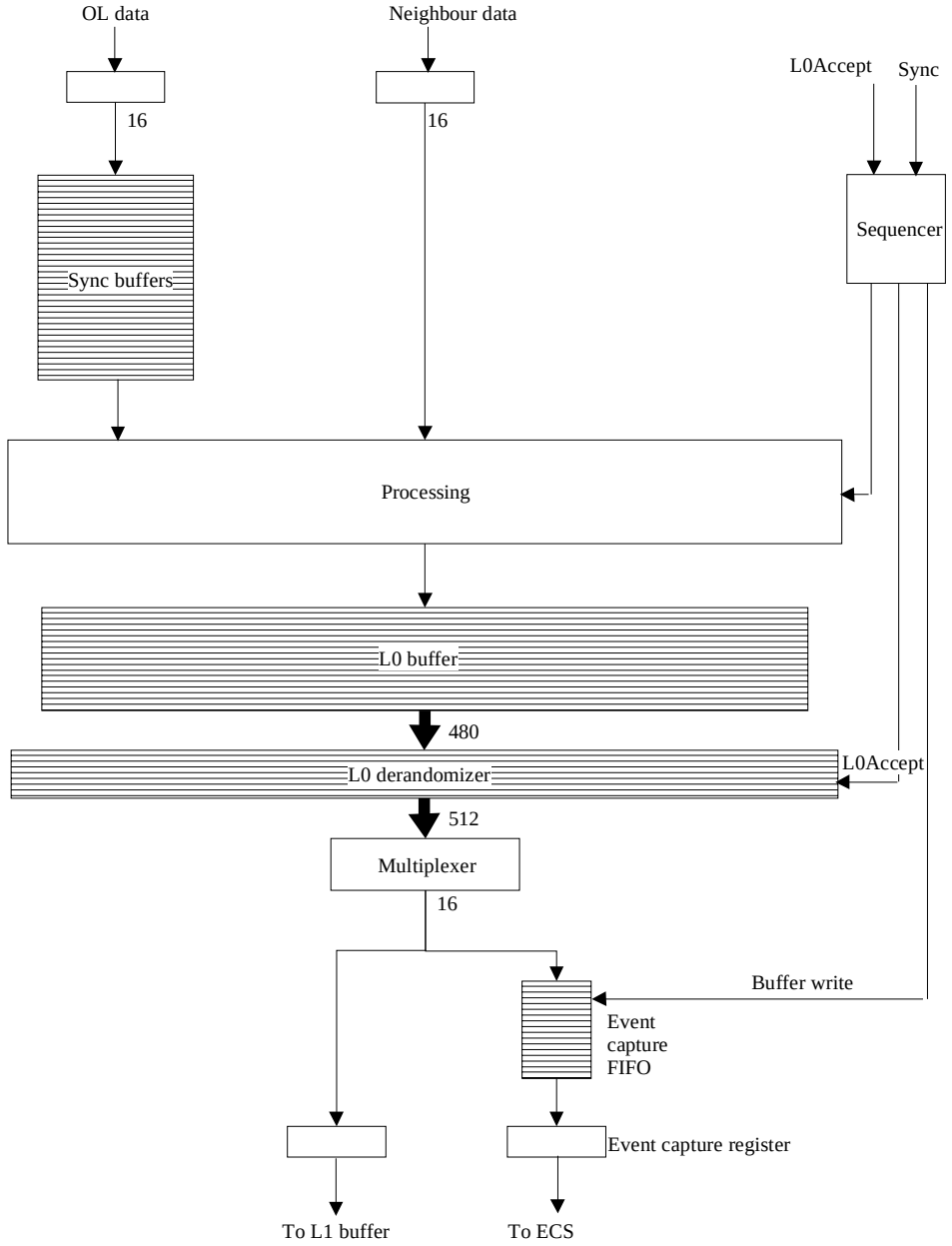



**Figure 14: Event capture path to ECS in operational mode**

The control register (see Table 14 page 45) allows to reset the event capture FIFO or to check whether it is empty.

The size of the event capture FIFO is 32 words of 16 bits to contain one event (512 bits). The output of the FIFO is connected to the local ECS supervisor through a 16 bits register shown in Table 19 . The local ECS supervisor must produce 32

consecutive reads of this register to get the full event.

The allocation of the bits is the same as the frame sent to the L1 buffer ( see § 6.1.7).

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output of Event capture FIFO | | | | | | | | | | | | | | | | *R* |

***Table 19: Event capture register***

**Output of Event capture FIFO:** output of the storage FIFO containing the captured event information

The detailed content of the optical links fields and neighbouring fields depends on the PU location (see Annex 1).

### 6.11.4 Registers used in test and operational modes

Parameter settings, masking operations and local resets can be performed in test and operational mode with three dedicated registers. They are described in the following section.

#### 6.11.4.1 PU parameters registers

Parameters values are set during the initialisation of the system or at any time when the processor is not in a running mode.

The foreseen parameters are:

- Size of the FOI in X and Y for station M1, M2, M4 and M5;
- Adjustment of first event detection window[3] (see description in Annex 2);
- ...

3 The detection of the first event after a L0 front−end reset is of prime importance for synchronising the data in the FIFOs. To decrease the risk of false detection due to a corrupted data on the optical lines, this detection is made during a reduced time interval centred on the theoretical time of arrival of the first data.

As the list of parameters is still not fixed, some room is left free in the following registers for a further extension.

The organisation of the registers is shown in Table 20

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M1 X search window size | | | | M2 X search window size | | | | M4 X search window size | | | | M5 X search window size | | | | *R/W* |
| – | | | | | | | | M4 Y search window size | | | | M5 Y search window size | | | | *R/W* |
| First data monitoring window central position | | | | | | | | – | | | | | First data monitoring window width | | | *R/W* |
| Reserved | | | | | | | | | | | | | | | | *R/W* |

***Table 20: Parameters adjustment registers***

**Mi X search window size**: size of the field of interest in pads for searching the hits in X direction in station Mi
**Mi Y search window size**: size of the field of interest in pads for searching the hits in Y direction in station Mi
**First data monitoring window central position**: central position of the first data monitoring window in number of clock cycles referenced from the BC0_Reset
**First data monitoring window width**: width of the monitoring window in clock cycles

### 6.11.4.2 PU masking operations registers

If there is a problem with the input of a PU, it is possible to mask it. The only consequence is that the trigger is blind on a limited part of the detector.

When input data are masked, their values are zeroed.

For a PU, we can isolate the following inputs:

- **Optical link**

This is obtained by setting to 1 an *"Isolate channel i"* bit in the register described in Table 21. In that case, the neighbouring information sent to others PUs and related to channel i are equal to zero.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | Isolate channel 1 | Isolate channel 2 | Isolate channel 3 | Isolate channel 4 | Isolate channel 5 | Isolate channel 6 | *R/W* |

***Table 21: Optical link masking register***

**Isolate channel i**: writing a 1 isolates the channel i. Writing a 0 restores the initial data path.

### ⑩ Backplane link

Data coming from neighbouring PUs located on other boards can be masked. Zeroed data are injected in the downstream stages in place of the incoming backplane data. This is obtained by setting to 1 an *"Isolate backplane link i"* bit in the register described in Table 22 (on the reception side).

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | Isolate backplane link M1 | Isolate backplane link M2 | Isolate backplane link M3 | Isolate backplane link M4 | Isolate backplane link M5 | *R/W* |

***Table 22: Backplane link masking register***

**Isolate backplane link i:** writing a 1 isolates the backplane link i. Writing a 0 restores the initial data path.

### ⑩ On board PU link

Data coming from neighbouring PUs located on the same board can be masked too. This is obtained by setting to 1 an *"Isolate PU link i"* bit in the register , as described in Table 23 (on the reception side).

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | Isolate PU link M1 | Isolate PU link M2 | Isolate PU link M3 | Isolate PU link M4 | Isolate PU link M5 | Isolate PU link U/D | *R/W* |

***Table 23: PU link masking register***

**Isolate PU link i:** writing a 1 isolates the PU link i. Writing a 0 restores the initial data path.
**Isolate PU link U/D**: writing a 1 isolates the Up/Down PU link. Writing a 0 restores the initial data path

### 6.11.4.3 Local resets registers

Local resets can be produced when local disfunction are detected. They can be emitted at a time which does not perturb the system functionality.

#### ⑩ Error detection on the sequencers

The correct operation of some sequencers needs to be continuously monitored, in particular those for which any disfunction is critical for the system: channel synchroniser, and L0 buffer control.

The sequencers must be encoded with one flip−flop per state. If more than one place is active at a time on a sequencer an error bit is raised. The sequencer error bits can be read in the registers described in Table 24:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | L0Buf Seq Err | L0Derandom overflow | Empty L0Derandom read attempt | Sync Seq 6 Err | Sync Seq 5 Err | Sync Seq 4 Err | Sync Seq 3 Err | Sync Seq 2 Err | Sync Seq 1 Err | *R/W* |
| – | | | | | | | L0Buf Seq Reset | Clear L0buffer | – | Sync Seq 6 Rerset | Sync Seq 5 Reset | Sync Seq 4 Reset | Sync Seq 3 Reset | Sync Seq 2 Reset | Sync Seq 1 Reset | *R/W* |

***Table 24: Sequencer Error Detection register***

**Sync Seq i Err**: set to 1 when an error has been detected on the channel i synchroniser sequencer. Automatically reset when the register is read.
**L0Buf Seq i Err:** set to 1 when an error has been detected on the L0 buffer manager. Automatically reset when the register is read.

### ⑩ Error detection on L0 derandomizer

The following defaults can be detected on the L0 derandomizer buffer:
- Overflow;
- Read attempt while the derandomizer is empty.

These cases are signalled on the same registers as above:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | L0Buf Seq Err | L0Derandom overflow | Empty L0Derandom read attempt | Sync Seq 6 Err | Sync Seq 5 Err | Sync Seq 4 Err | Sync Seq 3 Err | Sync Seq 2 Err | Sync Seq 1 Err | *R/W* |
| – | | | | | | | L0Buf Seq Reset | Clear L0buffer | – | Sync Seq 6 Rerset | Sync Seq 5 Reset | Sync Seq 4 Reset | Sync Seq 3 Reset | Sync Seq 2 Reset | Sync Seq 1 Reset | *R/W* |

***Table 25: L0 buffer Error detection register***

**L0Derandom overflow**: set to 1 when an overflow has been detected. Automatically reset when the register is read.

**Empty L0Derandom read attempt**: set to 1 when a reading has been attempted while the buffer is empty. Automatically reset when the register is read.

## 6.11.4.4 Local resets

If a sequencer has failed, it is necessary to reset it locally. However it is not guaranteed that the reset of a sequencer in the middle of a machine cycle will put the system in a correct operational mode. This is why these operations are initiated by the ECS, but the real operation must be made during the interval separating 2 consecutive LHC cycles. In this way there is no risk to loose permanently the synchronisation.

Similarly, if any problem occurs on the derandomizer FIFO, its pointers will be reset only during the next interval between 2 consecutive LHC cycles.

The same registers as above are used.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | L0Buf Seq Err | L0Derandom overflow | Empty L0Derandom read attempt | Sync Seq 6 Err | Sync Seq 5 Err | Sync Seq 4 Err | Sync Seq 3 Err | Sync Seq 2 Err | Sync Seq 1 Err | *R/W* |
| – | | | | | | | L0Buf Seq Reset | Clear L0buffer | – | Sync Seq 6 Rerset | Sync Seq 5 Reset | Sync Seq 4 Reset | Sync Seq 3 Reset | Sync Seq 2 Reset | Sync Seq 1 Reset | *R/W* |

***Table 26: Local resets registers***

**L0Buf Seq Reset**: writing a 1 resets the L0 buffer sequencer
**Clear L0Buffer:** writing a 1 resets the L0 buffer pointers
**Sync Seqi Reset:** writing a 1 resets the Sequencer of optical link i

### 6.11.5 Physical interface of the registers with the local ECS supervisor

The interface with the local ECS supervisor uses a 16 bits multiplexed address/data bus and 7 control signals. The control signals are defined in Table 27.

| Minimum necessary signals to interface the PLX 9030 PCI bridge | |
|---|---|
| A/D[15..0] | Multiplexed addresses/data |
| AS# | Address strobe |
| BLAST | Bus access last transfer |
| BTERM# | Burst terminate |
| LLOCKo# | Local bus lock |
| LBE1 | Address bit 1 (LA1) in 16 bits mode |
| LW/R# | Write/Read |
| INT# | Interrupt |
| READYo# | Ready out |

***Table 27: PU interface signals with the local ECS supervisor***

### 6.11.6 Addresses of the registers in a PU

The mapping of the registers over a PU is shown in Table 28:

| | | | |
|---|---|---|---|
| **PU Base Address** | 0x0000 | Control and status | Table 14 |
| | 0x0080 | Test pattern injection | Table 15 |
| | 0x0100 | Event capture for test mode | Table 17 |
| | 0x0180 | Error flags and error counters | Table 18 |
| | 0x0200 | Event captuire in operational mode | Table 19 |
| | 0x0280 | Parameters adjustment | Table 20 |

***Table 28: Address of the registers in a PU***

# 7 Specification of the PT computation module

The input of this module is the M1, M2 and M3 addresses for the two candidates:

- M3 absolute hit address within the PU. It is coded on 7 bits.
- M2 hit address within its FOI. It is coded on 4 bits.
- M1 hit address within its FOI . It is coded on 4 bits.

The output of the module is the PT value coded on 8 bits for both candidates. One bit is dedicated to the sign of the PT value.

## 7.1 Implementation

The PT computation is performed using a look−up−table (LUT) since the number of possibilities is limited ($2^{15}$ = 32 k). The result is obtained within a few clock cycles whatever the complexity of the computation is.

## 7.2 Initialisation of the LUT

The LUT must be initialised at the power−up of the board.

This is achieved by the local ECS supervisor through the BCSU modules (see § 8.5.4 page 76) as shown in Figure 15.

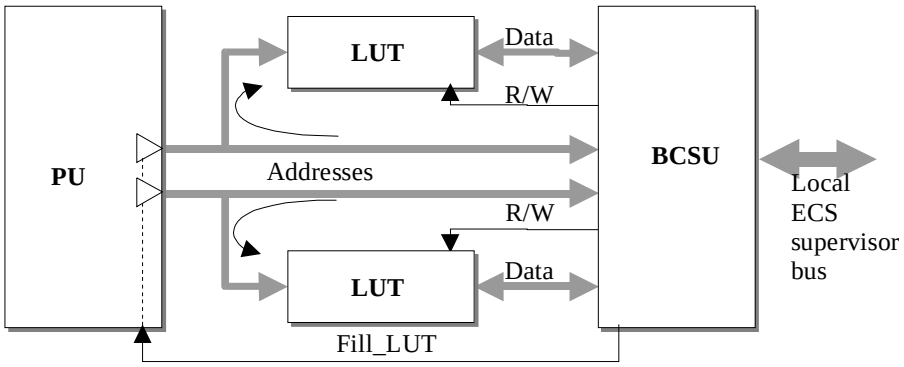



**Figure 15: LUT programmation**

The BCSU activates a Fill_LUT signal that disables (tri−states) the PU address output. Addresses and data are then driven by the BCSU to fill the LUT. When this operation is over, the Fill_LUT signal is de−activated and the PU drives again the addresses whereas the BCSU reverses the direction of its address and data ports. At this time the PU and BCSU are ready for processing candidates.

# 8 Specification of the BCSU FPGA

## 8.1 Inputs and outputs

The BCSU is interconnected to (see Figure 2 and Figure 15):

- the 4 PUs of the board;
- the PT calculation module;
- the L1 buffer;
- one intermediate candidate selection board;
- the controller board;
- the local ECS supervisor.

### 8.1.1 Connection to the PUs

The BCSU receives 8 candidates: 1 pair of candidates from each PU. Each pair of candidate is defined as follows:

- the local BCId coded on 4 bits;
- a status coded on 4 bits;
- two M3 address pad within the PU coded each on 7 bits;
- two M2 address within the FOI coded each on 4 bits;
- two M2 address within the FOI coded each on 4 bits.

In programming mode, the BCSU drives the above address in place of the PU by sending to the PU a “LUT_fill” signal. The latter is used to tri−state the addresses bus of the PU.

### 8.1.2 Connection to the PT calculation module

In operational or test modes, the BCSU receives from each PT calculation module

a PT value coded on 8 bits.

In programming mode, the BCSU drives the 15 bits address bus and the 8 bits data bus of the PT calculation module to fill the LUT.

In addition the BCSU drives a Read / Write signal per PT calculation module for controlling the operations of its LUT.

### 8.1.3 Connection to the L1 buffer

The BCSU sends to the L1 buffer:

- the content of the L0 derandomizer on a 16 bits bus at 40 MHz ;
- a write signal for the L1 buffer.

The data output from the L0 derandomizer buffer is a frame containing 3 fields, for each event:

- an identification field composed of 32 bits: a test mode flag (1 bit), L0 event number (12 bits), L0_B_Id (12 bits), free (7 bits);
- an output field composed of 48 bits: PT1 (8 bits), PT2 (8 bits), M1 address candidate #1 relative to the board (8 bits), M1 address candidate #2 relative to the board (8 bits), Status word (4 bits), BCId (4 bits), free (8 bits);
- an input field composed of 240 bits: candidates from PU1, PU2, PU3, PU4 (54 bits each[4]), free (24 bits).

The BCSU event fragment consists of 320 bits. It is sent sequentially to the L1 buffer with 20 consecutive 16 bits words.

The frame structure is described in Table 29:

4 BCId+ status word: 8 bits, input address for 2 candidates: 15 x 2 bits, 2 PT values: 8 x 2 bits = 54

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Field |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Sim | | | | L0Event number | | | | | | | | | | | | Identification |
| 1 | – | | | | L0_B_Id | | | | | | | | | | | | |
| 2 | PT1 | | | | | | | | PT2 | | | | | | | | Outputs |
| 3 | Address candidate 1 | | | | | | | | Address candidate 2 | | | | | | | | |
| 4 | – | | | | | | | | Status word | | | | BCId | | | | |
| 5 | PU inputs | | | | | | | | | | | | | | | | Inputs |
| 6 | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | |

***Table 29: BCSU to L1 buffer frame structure***

### 8.1.4 Connection to the muon selection board

The best 2 candidates found among those received from the PUs are sent to an i*ntermediate muon selection board.* The result is encoded on a word of 40 bits containing:

- local BCId coded on 4 bits;
- a status word coded on 4 bits;
- 2 M1 address coded on 8 bits. The coding is relative to the board (192 pads). It will be extended to a quarter of a muon station in the muon selection board;
- 2 PT values coded on 8 bits.

The words of 40 bits are multiplexed on a 20 bits path at 80 MHz, and sent trough the backplane. These data are sent with their 80 MHz clock.

### 8.1.5 Connection to the controller board

The BCSU receives from the controller board:

- a master clock at 40 MHz;
- a master clock at 80 MHz;
- a L0_B_Id number on 12 bits at 40 MHz;
- a L0 Accept signal;
- a BC0 reset signal;
- a Bunch count reset signal;
- a L0 front−end Reset signal;
- a L0 Event Counter Reset signal;
- a Sync signal;

### 8.1.6 Connection to the local ECS supervisor

The BCSU is connected to the local ECS supervisor by:

- a bidirectional 16 bit bus operating at 40 MHz;
- 1 interrupt signal;
- 7 bidirectional control signals.

### 8.1.7 Output for debug

- A BCSU can be connected to a logical analyser through a dedicated 16 bits bus + 1 clock.

## 8.2 Internal best candidate algorithm

Four pairs of candidates are received every 25 ns with their BCId tag and their status word. These tags are compared with the reference BCId minus a fixed offset[5]. If one of them differs from the theoretical BCId, the corresponding data are zeroed and a flag is raised for signalling to the ECS the loss of synchronisation. In practice the flag is set to 1 in the register interfacing the FPGA with the local ECS supervisor.

The 8 candidates are then compared together and the 2 with the highest PT value are retained. If 2 candidates or more have the same value, only 2 candidates are retained according to the following priority: upper right PU, lower right PU, upper left PU and lower left PU.

The remaining candidates are sent to the muon selection board for being further compared with the results from the other boards.

## 8.3 L0 buffer

Data written in the L0 buffer are:

- incoming candidates at the end of the time interval T6;
- outgoing candidates at the end of the time interval T7.

The L0 buffer is actually composed of 2 elementary buffers:

- the internal candidates buffer:
  - depth = $(T7 + T8)_{(\text{in ns})}$ /[25 ns];
  - width = 240 bits.
- the result candidates buffer:
  - depth = $(T8)_{(\text{in ns})}$ /[25 ns];
  - width = 48 bits.

The total width of the L0 buffer is 288 bits.

5 This offset corresponds to the length of the pipe−line up to the L0 buffer output because the reference BCId is associated to the time when the data is output from the L0 buffer.

The 2 buffers are read simultaneously in one clock cycle, when a *L0Accept* signal is received.

## 8.4 L0 derandomizer buffer

Data coming from the L0 buffer are transmitted to the L0 derandomizer buffer on each occurrence of a *L0Accept* signal. Each time an event is accepted, an event counter (12 bits) is incremented. The value of this event counter and the value of the current L0_B_Id are attached to the event entering in the L0 derandomizer buffer. We also write the value of the test mode flag.

The depth of the L0 derandomizer buffer is fixed to 16 events.

Its has the same width as the L0 buffer + 32 bits corresponding to the “event counter” tag and the L0_B_Id tag.

## 8.5 Interface with the local ECS supervisor

The ECS interface implemented in the BCSU module is similar to the one implemented in the PU. A set of registers interfaces the BCSU with the local ECS supervisor.

Three modes of operation are allowed, each being exclusive. Their functionalities use specific or common resources:

- Features used in test mode:
  - Pattern injection;
  - Event capture;
- Features used in operational mode:
  - Synchronisation error detection;
  - Masking of invalid inputs;
  - Event capture;
- Features used in programming mode:

– Programming of the PT calculation module

A special register allowing to choose the mode of operation and control the processing:

– Control of operations.

### 8.5.1 BCSU Control of operations register

A register allows to chose the mode of operation and in particular to start/stop the processing .

This register is accessible in test and operational modes.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | Reset event capture FIFO | Reset pattern injection FIFO | Enable Event capture | Enable Processing | Mode | *R/W* |
| – | | | | | | | | | | | | | Events Ready | Event capture Fifo empty | Pattern injection FIFO full | *R* |

***Table 30: BCSU Control and status of operations register***

**Mode**: writing a 1 chooses the operational mode, writing a 0 chooses the test mode.

**Enable processing:**

- In operational mode: writing a 1 enables the processing of data upon reception of the next *BC0 reset* that follows the reception of the *Sync* signal; writing a 0 stops the processing of data at the end of the machine cycle begun when the *Sync* signal is received.
- In test mode: writing a 1 starts the processing upon reception of the next *Sync* signal. The process is automatically stopped after transmission of 16 consecutive events (the start/stop bit is therefore automatically reset).

**Enable event capture:** writing a 1 enables the event capture that will be made on the next reception of a *L0 accep*t.

**Reset Pattern Injection FIFO:** writing a 1 resets the pattern injection FIFO. Automatically set to 0 when the operation is completed.

**Pattern injection FIFO full:** set to 1 when the pattern injection is full.

**Reset Event Capture FIFO:** writing a 1 resets the operational mode event capture FIFO. Automatically set to 0 when the operation is completed

**Event Capture FIFO empty:** set to 1 when the operational mode event capture FIFO is empty. Else : set to 0
**Event(s) Ready**: set to 1 when the data are captured (test or operational mode).

**Ready**: set to 1 when the event capture is done. Reset when read.

### 8.5.2 Registers used in test mode

The strategy for debugging the BCU is similar to the one of the PU. See § 6.11.2 page 46.

#### 8.5.2.1 BCSU pattern injection registers

The test buffers can be filled with data simulating the candidates found by the PUs for up to 16 consecutive events.

The content of the test buffers is written in a page of registers as shown in Table 31. They are the interface between the test buffers of the BCSU and the local ECS supervisor. The test buffers appear as a FIFO memory of pages. A page corresponds to an event. The next page is presented when the last register is written.

The control register (see Table 30 page 70) allows to reset the pattern injection FIFOs or to check whether they are full.

Note: *All the registers are Write Only. A read back possibility is provided by reading the L0 derandomizer buffer of the BCSU.*

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | Candidate 1 address PU1 | | | | | | | | | | | | | | | *W* |
| – | Candidate 2 address PU1 | | | | | | | | | | | | | | | *W* |
| PT value candidate 1 PU1 | | | | | | | | PT value candidate2 PU1 | | | | | | | | *W* |
| – | | | | | | | | Status word PU1 | | | | BCId PU1 | | | | *W* |
| – | Candidate 1 address PU2 | | | | | | | | | | | | | | | *W* |
| – | Candidate 2 address PU2 | | | | | | | | | | | | | | | *W* |
| PT value candidate1 PU2 | | | | | | | | PT value candidate 2 PU2 | | | | | | | | *W* |
| – | | | | | | | | Status word PU2 | | | | BCId PU2 | | | | *W* |
| – | Candidate 1 address PU3 | | | | | | | | | | | | | | | *W* |
| – | Candidate 2 address PU3 | | | | | | | | | | | | | | | *W* |
| PT value candidate 1 PU3 | | | | | | | | PT value candidate 2 PU3 | | | | | | | | *W* |
| – | | | | | | | | Status word PU3 | | | | BCId PU3 | | | | *W* |
| – | Candidate 1 address PU4 | | | | | | | | | | | | | | | *W* |
| – | Candidate 2 address PU4 | | | | | | | | | | | | | | | *W* |
| PT value candidate1 PU4 | | | | | | | | PT value candidate 2 PU4 | | | | | | | | *W* |
| – | | | | | | | | Status word PU4 | | | | BCId PU4 | | | | *W* |

***Table 31: BCSU Test pattern injection registers***

**Candidate 1 address PUi**: Address of the first candidate for PUi .
**Candidate 2 address PUi**: Address of the second candidate for PUi .
**PT value candidate 1 PUi**: PT value of the first candidate for PUi.
**PT value candidate 2 PUi**: PT value of the second candidate for PUi.
**Status word PUi**: status word of the PUi candidates.
**BCId PUi:** BCId of the PUi candidates.

### 8.5.2.2 BCSU consecutive events capture registers

The local ECS supervisor can read the content of the L0 derandomizer buffer. When the processor is launched in test mode, fake *L0accept* signals are generated to fill automatically the L0 derandomizer buffer with the values of 16 consecutive events. The output of the derandomizer buffer is then multiplexed on a 16 bits path connected to the ECS interface.

When the ready flag of the control register is raised (see § 8.5.1), the event has been captured and the page registers described in Table 32 shows the content of the captured event.

It is then possible to read:
- the 8 input candidates
- the result of the global selection

All the data registers are read only.

The derandomizer buffer is implemented in a 320 bits wide FIFO. The data page shown in the registers is the output of this FIFO and therefore contains a complete event. The page is automatically refreshed with the next event after each reading of the last register of the page.

The control register (see Table 30 page 70) allows to reset the derandomizer FIFO or to check whether it is empty.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| − | Candidate 1 M3/M2/M1address for PU1 | | | | | | | | | | | | | | | *R* |
| − | Candidate 2 M3/M2/M1address for PU1 | | | | | | | | | | | | | | | *R* |
| PT value candidate 1 PU1 | | | | | | | | PT value candidate2 PU1 | | | | | | | | *R* |
| − | | | | | | | | Status word PU1 | | | | BCId PU1 | | | | *R* |
| − | Candidate 1 M3/M2/M1address for PU2 | | | | | | | | | | | | | | | *R* |
| − | Candidate 2 M3/M2/M1address for PU2 | | | | | | | | | | | | | | | *R* |
| PT value candidate1 PU2 | | | | | | | | PT value candidate 2 PU2 | | | | | | | | *R* |
| − | | | | | | | | Status word PU2 | | | | BCId PU2 | | | | *R* |
| − | Candidate 1 M3/M2/M1address for PU3 | | | | | | | | | | | | | | | *R* |
| − | Candidate 2 M3/M2/M1address for PU3 | | | | | | | | | | | | | | | *R* |
| PT value candidate 1 PU3 | | | | | | | | PT value candidate 2 PU3 | | | | | | | | *R* |
| − | | | | | | | | Status word PU3 | | | | BCId PU3 | | | | *R* |
| − | Candidate 1 M3/M2/M1address for PU4 | | | | | | | | | | | | | | | *R* |
| − | Candidate 2 M3/M2/M1address for PU4 | | | | | | | | | | | | | | | *R* |
| PT value candidate1 PU4 | | | | | | | | PT value candidate 2 PU4 | | | | | | | | *R* |
| − | | | | | | | | Status word PU4 | | | | BCId PU4 | | | | *R* |
| | | | | | | | | Best candidate 1 M1 address for BCSU | | | | | | | | *R* |
| | | | | | | | | Best candidate 2 M1 address for BCSU | | | | | | | | *R* |
| PT candidate 1 BCSU | | | | | | | | PT candidate 2 BCSU | | | | | | | | *R* |
| − | | | | | | | | BCId BCSU | | | | Status word BCSU | | | | *R* |

***Table 32: BCSU Event capture register in test mode***

**Candidate 1 M3/M2/M1 address for PUi**: Address of the first candidate for PUi .
**Candidate 2 M3/M2/M1 address for PUi**: Address of the second candidate for PUi .
**PT value candidate 1 PUi**: PT value of the first candidate for PUi.
**PT value candidate 2 PUi**: PT value of the second candidate for PUi.
**Status word PUi**: status word of the PUi candidates.
**BCId PUi: BCId of the PUi candidates.**

**Best candidate 1 M1 address for BCSU**: M1 address of the first best candidate for BCSU .
**Best candidate 2 M1 address for BCSU**: M1 address of the second best candidate for BCSU .
**PT value candidate 1 BCSU**: PT value of the first candidate for BCSU.
**PT value candidate 2 BCSU**: PT value of the second candidate for BCSU.
**BCId BCSU:** BCId of the BCSU candidates.
**Status word BCSU**: status word of the BCSU candidates.

## 8.5.3 Registers used in operational mode

### 8.5.3.1 BCSU event capture register

The content of the L0 derandomizer is accessible by the ECS for one event when an event capture is launched.

The event capture is enabled by writing a 1 in the “enable event capture” slot of the control register. The next L0Accept signal following the first rising edge of the Sync signal is used to latch the data as described in Figure 13 page 53 . Waiting for the Sync signal allows to synchronise the event capture among several FPGAs or among several cards.

The output of the derandomizer buffer is then serialised on a 16 bits path and entered in a FIFO whose output is visible through the registers described in Table 33 . When the data are available a ready bit is set in the Control register (see § 8.5.1).

The control register (see Table 30 page 70) allows to reset the event capture FIFO or to check whether it is empty.

The buffer width is fixed to 320 bits (20 words of 16 bits).
The allocation of the bits in the buffer is the same as the frame sent to the L1 buffer ( see § 8.1.3).

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Event capture FIFO ouput | | | | | | | | | | | | | | | | *R* |

***Table 33: BCSU Event capture register in operational mode***

### 8.5.3.2 BCSU synchronisation error detection register

Errors of synchronisation are detected by comparing the received BCId with a theoretical BCId computed from the *L0_B_Id* value. A specific register allows to store the sources of error:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Error counter | | | | | | | | | | | | PU1 wrong BCId | PU2 wrong BCId | PU3 wrong BCId | PU4 wrong BCId | *R/W* |

***Table 34: BCSU error detection register***

**PUi wrong BCId**: wrong BCId detected on data from PUi

**Error counter:** counts the number of errors occurred. Automatically reset after reading.

### 8.5.3.3 BCSU masking register

A specific register allows to mask the inputs of the BCSU module:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | | Isolate data from PU1 | Isolate data from PU2 | Isolate data from PU3 | Isolate data from PU4 | *R/W* |

***Table 35: BCSU masking register***

**Isolate data from PUi**: when set to 1 all data from PUi are substituted by zeroes.

### 8.5.4 LUT programming register

The BCSU is used to fill the LUTs of the PT calculation module.

In this case the BCSU drives the addresses of the LUT in place of the PUs. The BCSU also drives the data of the LUTs and sends a write signal to the LUTs. This mode is signalled to the corresponding PU by activating the "fill_PU mode" bit of the LUT programming register. The PU then tri−states its addresses. To avoid an accidental conflict on the address lines due to a bad programming, it is impossible for the BCSU to drive the LUT addresses is this bit is not set.

There are 8 LUTs to program. The programming addresses require 15 bits for each PU. The width of the local bus is only 16 bits. A page register is therefore needed to select one LUT out of 8. This page register is available in the LUT programming register described in Table 36.

Addresses that are emitted in the upper part of the addressing field of the board (bit A15 = 1) are routed through the BCSU to the address bus of the LUT pointed by the page register.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | | Fill PU mode | Page register | | | R/W |

***Table 36: LUT programming register***

### 8.5.5 Physical interface of the registers with the local ECS supervisor

The interface with the local ECS supervisor uses a 16 bits multiplexed address/data bus and 7 control signals. The control signals are defined in Table 37.

| Minimum necessary signals to interface the PLX 9030 PCI bridg | |
|---|---|
| A/D[15..0] | Multiplexed addresses/data |
| AS# | Address strobe |
| BLAST | Bus access last transfer |
| BTERM# | Burst terminate |
| LBE1 | Address bit 1 (LA1) in 16 bits mode |
| LCLCK | Local clock |
| LW/R# | Write/Read |
| INT# | Interrupt |
| READYo# | Ready out |

***Table 37: BCSU interface signals with the local ECS supervisor***

## 8.5.6 Addresses of the registers in the BCSU

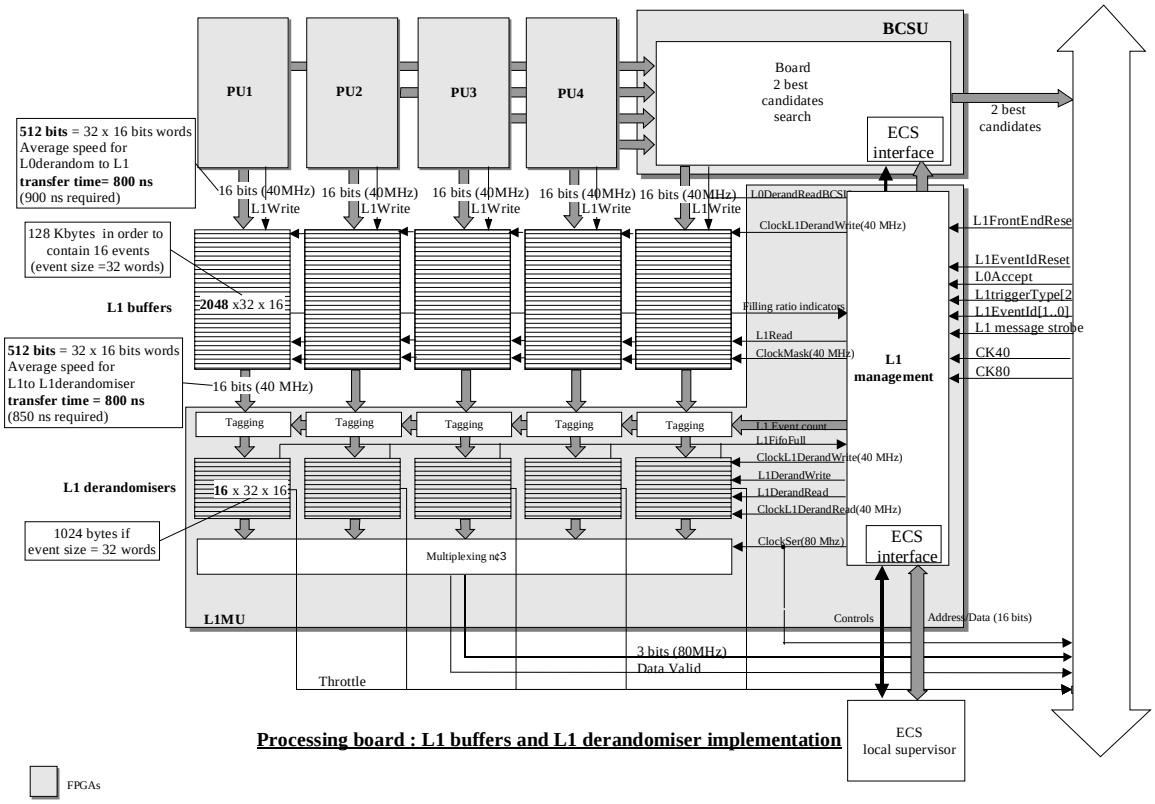
The mapping of the registers over a BCSU is as follows:

| | | | |
|---|---|---|---|
| **Base Address +** | 0x0000 | Control & status | Table 30 |
| | 0x0080 | Test pattern injection | Table 31 |
| | 0x0100 | Consecutive event capture in test mode | Table 32 |
| | 0x0180 | Event capture registers in operational mode | Table 33 |
| | 0x0200 | Error detection | Table 34 |
| | 0x0280 | Masking | Table 35 |
| | 0x0300 | LUT programming | Table 36 |

***Table 38: Addresses of the registers in the BCSU***

# 9 Specification of the L1 buffer

The L1 buffer is outside the FPGAs containing the PU and BCSU modules. It requires 5 FIFOs with a size of 2048 x 16. A FIFO receives the data from one PU or from the BCSU module. The overall scheme of the L1 buffer is given in Figure 16.




**Processing board : L1 buffers and L1 derandomiser implementation**

**Figure 16: L1 buffer management**

## 9.1 Inputs and outputs

The L1 buffer is connected to L0 derandomizer buffers of PUs and BCSU through 16 bits busses running at 40 MHz. Each FIFO receives sequentially for each accepted bunch crossing either 32 words from a PU or 20 words from the BCSU.

## 9.2 Operation

The data are transmitted on a 16 bits path at 40 MHz. The transfer rate is limited by PU frames which are the longest ones. The maximum rate for consecutive events is 1.25 MHz (800 ns per event).

The *write* signals accompany data coming from the PU or the BCSU. They are used to store the data in the FIFOs.

When a read signal coming from the L1 buffer Management Unit is emitted, the data are output from the L1 buffer to the L1 derandomizer buffer on a 16 bits path at 40 MHz.

# 10 Specification of the L1MU FPGA

The L1 buffer management unit has the following roles:

- controls the L1 buffer FIFOs;
- contains and controls the L1 derandomizer buffer;
- controls the coherence of the information received from the L1 buffer;
- concatenates and reformats the data into a single frame, and adds specific tagging words;
- serialises the L1 derandomizer buffer and sends its content to the Controller board through the backplane;
- presents to the ECS a board and position identifier.

## 10.1 Inputs and outputs

The L1MU is interconnected to:

- the L1 buffer;
- the controller board;
- the local ECS supervisor.

### 10.1.1 Connection with the L1 buffer

The L1MU receives 5 data frames multiplexed on five 16 bits paths at 40 MHz. They come from the L1 buffer implemented in the external FIFOs.

These data represent the content of the accepted events by the L1 trigger.

The data format is identical to the one defined at the output of the PUs or BCSU. See Table 10 page 34 for the information coming from the PU and Table 29 page 66 for the information coming from the BCSU.

A common *PU_L1Read* signal is sent to the 4 L1 FIFOS and a *BCSU_L1Read* signal

is sent to the BCSU L1 FIFO. It allows the data transfer from the L1 buffer FIFOs to the L1MU.

The L1MU also receives 5 flags from the L1 buffer that indicate the filling level of the L1 buffer:

- *L1_Empty*,
- *L1_Filled_at_25%*,
- *L1_Filled_at_50%*,
- *L1_Filled_at_75%*,
- *L1_Full.*

The L1MU is able to reset the L1 buffer FIFO by activating a *Reset_L1_buffer* signal.

### 10.1.2 Connection to the controller board

The L1MU receives the following signals from the controller board through the backplane:

- *BC0Reset*,
- *Bunch Count Reset*,
- *L1 Front−end reset*
- *L1 Event Counter Reset*,
- *L1TriggerType[2..0]*,
- *L1TriggerStrobe*,
- *EventId[1..0]*,
- *Sync*,
- *StopSendingL1DerandomContent*,
- a master clock at 40 MHz,

- a master clock at 80 MHz.

The *L1TriggerType[2..0] information* indicates the type of trigger as shown in Table 39. This signal is accompanied by 2 bits (*EventId[1..0]*) corresponding to the LSB of the accepted or rejected event number.

| Bit2 | Bit1 | Bit0 | Meaning |
|---|---|---|---|
| 0 | 0 | 0 | L1 reject |
| 0 | 0 | 1 | L1 accept |
| 0 | 1 | 0 | Reserved |
| 0 | 1 | 1 | Random trigger |
| 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Timing alignment trigger |
| 1 | 1 | 1 | Trigger on calibration pulse |

***Table 39: L1 trigger type distributed to front end electronics***

The L1MU sends to the controller board:

- A throttle signal. Its role is to warn the controller board when the L1 derandomizer buffer is close to overflow. This open collector signal is commonly driven by all the processing boards of the crate on the backplane towards the controller board.
- 4 data links at 80 Mbits/s, containing the information selected by L1 accepts;
- a 80 MHz clock synchronous with the data;
- a data valid signal.

The frame structure is shown in table Table 40. It contains 6 fields:

- a global identification field composed of 80 bits: board number within the crate (4 bits), Trigger Type (3 bits), L0_B_Id (12 bits), L0 event number (12 bits), L1 event number (32 bits), Error flags (16 bits), free (1 bit);
- 4 PU information containing each:
  - an output field (48 bits);
  - an input field (432 bits).
- a BCSU information containing :
  - an output field (48 bits);

– an input field (240 bits).

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Field |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BoardId | | | | BCId | | | | | | | | | | | | Identification |
| 1 | − | Trigger type | | | L0EventId | | | | | | | | | | | | |
| 2 | L1EventId | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | |
| 4 | Error flags | | | | | | | | | | | | | | | | |
| 5 | PU1 inputs+outputs | | | | | | | | | | | | | | | | PU1 |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| 35 | PU2 inputs+outputs | | | | | | | | | | | | | | | | PU2 |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| 65 | PU3 inputs+outputs | | | | | | | | | | | | | | | | PU3 |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| 95 | PU4 inputs+outputs | | | | | | | | | | | | | | | | PU4 |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| 125 | BCSU inputs+outputs | | | | | | | | | | | | | | | | BCSU |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | | | |
| 141 | | | | | | | | | | | | | | | | | |

***Table 40: Frame format for the L1 derandomizer output***

The data frame corresponding to an event contains 2288 bits of information equal to 143 words of 16 bits. The average read−out time allowed per event must be 10% faster than the average L1 decision rate. For a L1 decision rate of 100 kHz the read−out time for 1 event is equal to 9µs. This corresponds to a data flow of 254 Mbits/s. We spread it on 4 serial data links at 80 MHz.

### 10.1.3 Connection with the local ECS supervisor

The L1MU is connected to the local ECS supervisor by:

- a bidirectional 16 bit bus operating at 40 MHz;
- 1 interrupt signal;
- 7 bidirectional control signals.

### 10.1.4 Output for debug

A L1 MU can be connected to a logical analyser through a dedicated 16 bits bus + 1 clock.

## 10.2 Reading the L1 buffer and filling the L1 derandomizer buffer

The L1MU controls reading operations of the L1 buffer.

The strategy to accept or reject an event at the output of the L1 buffer is the following. It relies on the interpretation of the 3 bits of the *L1TriggerType[2..0]* flags during the activity of the *L1AcceptStrobe* signal.

A *L1AcceptStrobe* is generated for each event stored in the L1 buffer. It comes with the *L1 Accept condition* (true if *L1TriggerType[2..0]* = 1, 3, 6 or 7, else false) and the 2 LSB bits of the L0 Event Number corresponding to the selected event (*EventId[1..0]*) as shown in Figure 17.

The latter allows an additional synchronisation test since *EventId[1..0]* has to correspond to the 2 LSB bits of the *L0 Event Number* stored with the event.

The procedure to extract the data from the L1 buffer and to fill the L1 derandomizer is the following. On every occurrence of the *L1AcceptStrobe*, the FIFOs are read independently of the *L1 Accept condition*. This is required to empty the FIFO and avoid FIFO overflow since it is read the same number of times it has been written.

The event is only stored in the L1 derandomizer buffer when the *L1 Accept condition* is true.

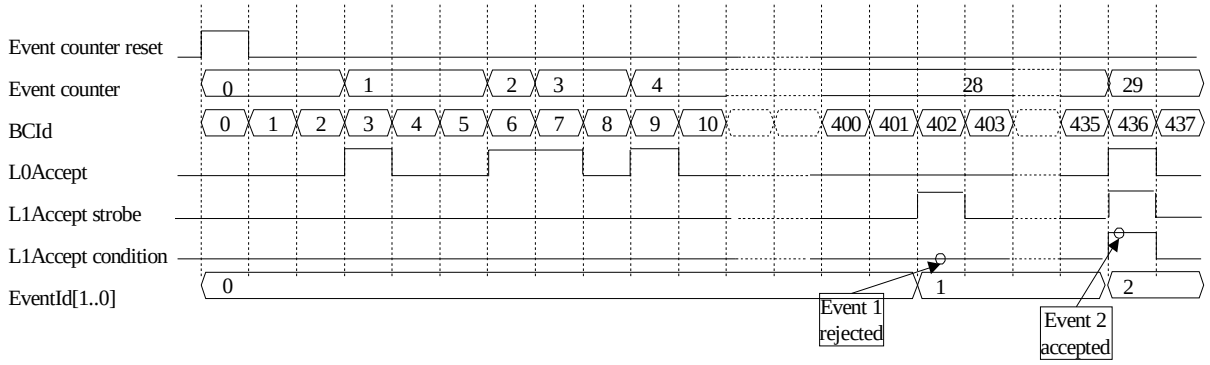



**Figure 17: L1 accept principle**

## 10.3 Tagging the L1 event

A 32 bits internal L1 event counter is incremented by each occurrence of the *L1Accept* signal. This counter is reset by the *L1 Event Count Reset* signal.

The value of this event counter and the trigger type is inserted in the frame before entering the L1 derandomizer buffer.

## 10.4 L1 derandomizer buffer

On each occurrence of the *L1Accept* signal, an event is entered sequentially in the L1 derandomizer buffer. The L1 derandomizer buffer is composed of 5 FIFOs of

16 bits width. Each FIFO contains 16 events composed of either 32 successive 16 bits words (PU) or 20 successive 16 bits words (BCSU).

The L1 derandomizer buffer is dumped and serialised by a sequencer when it contains at least 1 event and when the condition *StopSendingL1DerandomContent* is false. This signal is sent by the controller board when the zero suppression output buffers overflow.

### 10.4.1 Control of the L1 derandomizer overflow

The derandomizer control mechanism have to provide a throttle signal to the DAQ. When the L1 derandomizer buffer is filled with a certain number of events, the throttle signal is emitted. In this case the system will stop providing *L1Accepts* allowing to empty the L1 derandomizer buffer. The event limit is programmable between 0 and 13. Since the time to process the throttle signal can be up to 2 µs, the L1 derandomizer buffer must be able to accept up to 3 additional events without overflowing. Therefore the upper limit of the programmable number of events must be 16 − 3 = 13 events.

At the L1MU level an individual throttle signal is emitted towards the backplane. This signal is open collector in order to “OR” it with the throttling signals of the other boards.

## 10.5 Formatting

The incoming frames are re−arranged to eliminate the redundant information after control of coherence. For example the identification fields are encoded only once in the global frame. Inconsistencies in the incoming *BCIds* or *L0_B_Ids* are signalled in a specific error word added to the frame.

## 10.6 Serialisation

All the data contained in the L1 derandomizer represent 2288 bits. They are serialised on 4 data links at 80 MHz.

To avoid skew issues between the data coming from different places in the backplane and the global 80 MHz system clock, we send a clock time aligned to

the 4 signals.

As the flow of data is not continuous, a "Data valid" signal is also sent to allow the controller board to know when valid information is incoming.

## 10.7 Interface with the local ECS supervisor

The control of the L1MU and the reading of its status are available through the ECS interface. This is implemented with a set of registers allowing to configure some programmable features or read information within the L1MU. This set of register can be read by the local ECS supervisor.

Two modes of operation are allowed, each being exclusive. Their functionalities use specific or common resources :

- Features used in test mode:
  - Pattern injection;
  - Masking;
  - Local resets

- Features used in operational mode:
  - Reading of a status word signalling any synchronisation error;
  - Masking;
  - Local resets.

A special register allowing to choose the mode of operation is accessible in both modes:

  - Control of operations

### 10.7.1 Control of operations

A register allows to chose the mode of operation and in particular to start/stop

the processing. It is described in Table 41.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | – | | | | | | | Reset pattern injection FIFO | Enable processing | Mode | *R/W* |
| | | | | | | | – | | | | | | | | Pattern injection FIFO full | *R* |

***Table 41: L1MU control & status register***

**Mode**: writing a 1 chooses the operational mode, writing a 0 chooses the test mode.

**Enable processing:**

- In operational mode: writing a 1 starts the processing of data upon reception of the next *BC0 reset that follows the reception of the Sync signal;* writing a 0 stops the processing of data at the end of the machine cycle when the *Sync* signal is received.
- In test mode: writing a 1 starts the processing upon reception of the next *Sync* signal. The process is automatically stopped after transmission of 16 consecutive events (the start / stop bit is therefore automatically reset).

**Reset Pattern Injection FIFO:** writing a 1 resets the pattern injection FIFO. Automatically set to 0 when the operation is completed.

**Pattern injection FIFO full:** set to 1 when the pattern injection is full.

## 10.7.2 Registers used in test mode

### 10.7.2.1 Pattern injection registers

The L1 derandomizer buffers can be filled with simulated data for up to 16 events.

The objective of the pattern injection is to check the correct operation of the serial link between the L1MU and the receiving end of the Controller board.

The L1 derandomizer buffers input is seen by the local ECS supervisor as five 16 bits FIFO inputs, as described in Table 42.

To fill the L1 derandomizer buffers with test patterns, 32 consecutive words are written in the corresponding register for a PU and 20 for the BCSU.

Note : *All the other registers are Write Only. There is no local read back possibility: data can be read back only on the controller board.*

The *Enable processing* bit of the control register allows to start the processing when the *Sync* signal has been received.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Simulated data from PU1 | | | | | | | | | | | | | | | | *W* |
| Simulated data from PU2 | | | | | | | | | | | | | | | | *W* |
| Simulated data from PU3 | | | | | | | | | | | | | | | | *W* |
| Simulated data from PU4 | | | | | | | | | | | | | | | | *W* |
| Simulated data from BCSU | | | | | | | | | | | | | | | | *W* |

***Table 42: L1MU pattern injection registers in test mode***

**Simulated data from PUi (BCSU**): simulated data filling the portion of the derandomizer devoted to PUi (BCSU)

## *10.7.3 Registers used in operational mode*

### 10.7.3.1 Error detection registers

Data coming from the different PUs or from the BCSU must have identical BCIds and L0 event number tags. The BCId and the L0 event number tags are compared together: if a discrepancy is found a flag is raised. Six comparisons 2 by 2 are made together to allow the localisation of the desynchronised channel. In addition the 2 LSB of the L0 event counter for each channel received from L1 is compared with the 2 bits EventId information sent by the controller board. Any discrepancy is signalled. A register described in Table 43 allows to read these flags and the number of errors.

**BCId i_j**: BCId from input i different from BCId from input j . Set to 1 on the first occurrence of the error. Set to 0 when the register is read.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BCId error counter | | | | | | | | | | BCId 1_2 | BCId 1_3 | BCId 1_4 | BCId 2_3 | BCId 2_4 | BCId 3_4 | *R* |
| Complete event number error counter | | | | | | | | | | EvN# 1_2 | Ev N# 1_3 | Ev N# 1_4 | Ev N# 2_3 | Ev N# 2_4 | Ev N# 3_4 | *R* |
| Partial event number error counter | | | | | | | | | | – | LSB EvN# 1 | Ev N# 2 | LSB Ev N# 3 | LSB Ev N# 4 | LSB Ev N# 5 | *R* |

***Table 43: L1MU error detection register***

**BCId error counter:** counts the number of BCId errors occurred. Reset after the reading of the register.
**Ev N# i_j**: Event number from input i different from Event number from input j . Set to 1 on the first occurrence of the error. Set to 0 when the register is read.
**Complete event number error counter:** counts the number of event number discrepancies occurred by comparing L0 event numbers 2 by 2 . Reset after the reading of the register.
**LSB Ev N# i**: LSB Event number from input i different from the EventId[1..0] data received from the controller board. Set to 1 on the first occurrence of the error. Set to 0 when the register is read.
**Partial event number error counter:** counts the number of event number errors occurred by comparing the LSB L0 event numbers with the EventId[1..0] data received from the controller board. Reset after the reading of the register.

### 10.7.3.2 L1 buffer control register

The amount of data in the L1 buffer can be read by the ECS. The L1 buffer can also be reset. A register described in Table 44 allows to perform these operations.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | | | | | | | | Reset L1 buffer | *R/W* |
| – | | | | | | | | | | | L1 buffer empty | buffer filled at 25% | buffer filled at 75% | buffer filled at 75% | L1 buffer full | *R* |

***Table 44: L1 buffer management register***

**Reset L1 buffer**: writing a 1 resets the L1 buffer. Automatically set to 0 after the operation.
**L1 buffer empty :** set to 1 when the L1 buffer is empty.
**L1 buffer filled at 25 %:** set to 1 when the L1 buffer is at least filled at 25%.
**L1 buffer filled at 50 %:** set to 1 when the L1 buffer is at least filled at 50%.
**L1 buffer filled at 75 %:** set to 1 when the L1 buffer is at least filled at 75%.
**L1 buffer full :** set to 1 when the L1 buffer is full.

### 10.7.3.3 L1 derandomizer control register

The throttle signal must be emitted when a programmable count of events has filled the L1 derandomizer FIFO. The throttle signal can be masked to avoid blocking the whole experiment.
A register described in Table 45 allows to program the number of events before a throttle signal is raised[6] and also to mask the signal.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | | | | | | | | Mask L1 throttle | – | – | – | Max number of events in L1 derand | | | | *R/W* |

***Table 45: L1 derandomizer management register***

**Mask L1 throttle**: when set to 1, the throttle signal is masked.

**Max number of events in L1 derand**: maximum number of events allowed in the derandomizer FIFO (maximum value: 13). When this limit is reached or overtaken the throttle signal is activated. The throttle signal is deactivated as soon as the number of events contained in the FIFO becomes inferior to the programmed maximum.

### 10.7.3.4 L1 throttle time monitoring register

The local monitoring of the L1 throttle is made with a simple time counter that counts the time the throttle signal is asserted since the last L1 front−end reset. The counter is incremented by the 40 MHz system clock. Its width is 32 bits. Assuming that under normal conditions, the local throttle is below 1%, the dynamic range of this counter is sufficiently large to support running the system for days. The counter must not overflow to 0: it must be made to saturate at hex FFF...F.

6 The nominal value is 13 for the time being.

A register allowing to read its value is described in Table 46.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L1 throttle time MSB | | | | | | | | | | | | | | | | *R/W* |
| L1 throttle time LSB | | | | | | | | | | | | | | | | *R/W* |

***Table 46: L1 throttle time monitoring register***

**L1 throttle time MSB:** MSB of the 32 bits counter measuring the time during which the throttle signal is activated.

**L1 throttle time LSB:** LSB of the 32 bits counter measuring the time during which the throttle signal is activated.

### 10.7.4 Physical interface of the registers with the local ECS supervisor

The interface with the local ECS supervisor uses a 16 bits multiplexed address/data bus and 7 control signals. The control signals are defined in Table 47.

| Minimum necessary signals to interface the PLX 9030 PCI bridg | |
|---|---|
| A/D[15..0] | Multiplexed addresses/data |
| AS# | Address strobe |
| BLAST | Bus access last transfer |
| BTERM# | Burst terminate |
| LBE1 | Address bit 1 (LA1) in 16 bits mode |
| LCLCK | Local clock |
| LW/R# | Write/Read |
| INT# | Interrupt |
| READYo# | Ready out |

***Table 47: BCSU interface signals with the local ECS supervisor***

### 10.7.5 Addresses of the registers in the L1MU

The mapping of the registers over a L1MU is described in Table 48:

| | | | |
|---|---|---|---|
| **L1MU Base Address +** | 0x0000 | Control & status | Table 41 |
| | 0x0080 | Pattern injection in test mode | Table 42 |
| | 0x0100 | Error detection | Table 43 |
| | 0x0180 | L1 buffer management | Table 44 |
| | 0x0200 | L1 derandomiser management | Table 45 |
| | 0x0280 | L1 trottle time monitoring | Table 46 |

***Table 48: BCSU register addresses***

# 11 Specification of the local ECS supervisor

The local ECS supervisor is the interface between the ECS system and the resources of the board. It is in charge of programming the FPGAs and of setting the internal parameters. It can also read error counters. It is implemented using an embedded microPC on each board. The microPC communicates with the ECS system through an Ethernet link.

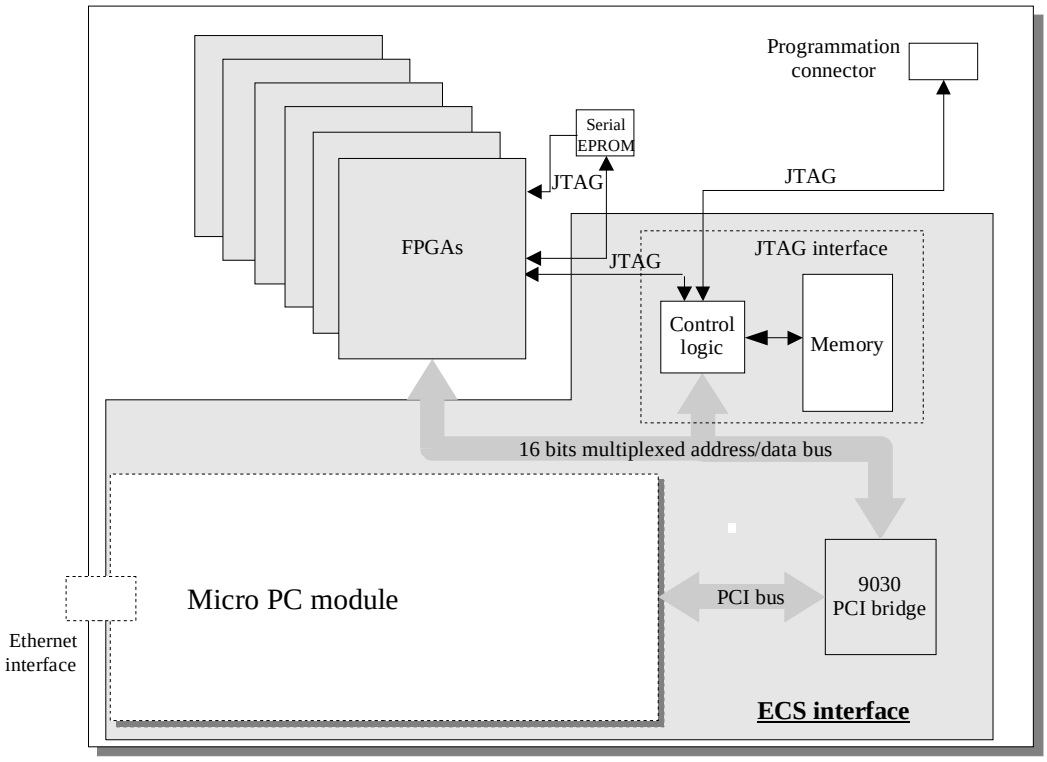



**Figure 18: Overview of the local ECS supervisor**

## 11.1 Inputs/Outputs

The local ECS supervisor is interfaced with:

- the upper control layers by an Ethernet 100 Mbits/s link;

- the internal registers of the FPGAs by a multiplexed address / data bus on a 16 bits path, accompanied by 10 control signals;
- the programming interface of the FPGA by a JTAG serial bus.

## 11.2 Interface with the FPGAs

The microPC interface, shown in Figure 18, consists of a PCI bus which is converted into LVTTL signals by a PCI bridge (PCI9030 from PLX).

Glue logic allows to download the FPGAs.

## 11.3 Mapping of the FPGA registers

The mapping of the registers over the processing board is as follows:

| | |
|---|---|
| 0x0000 | PU1 registers |
| 0x1000 | PU2 registers |
| 0x2000 | PU3 registers |
| 0x3000 | PU4 registers |
| 0x4000 | BCSU registers |
| 0x5000 | L1MU registers |

***Table 49: FPGA registers base addresses on a board***

## 11.4 Mapping of the LUTs

The mapping of the LUT over the processing board is as follows:

| | |
|---|---|
| 0x8000<br>0x8FFF | LUT |

***Table 50: LUT base address on a board***

The 8 LUTs have the same address. The choice of the LUT is made by the BCSU register (see § 8.5.4 page 76)

# 12 Board clocks

2 clocks arrive on the board from the backplane: a 80 MHz clock and a 40 MHz clock. These 2 clocks are generated by the controller board. They are in phase.

As they must feed up to 6 FPGAs each, they must be replicated with a specific PLL clock replicator.

The 40 MHz clock is divided by 2 to feed the Credit−Card PC and the PCI bridge.

# 13 Board inputs and outputs connectors definition

## 13.1 Optical transceivers

To be defined

## 13.2 Backplane

To be defined.

# 14 Mechanical specifications

The board format is compliant with the extended 9 U VME standard.

# 15 Electrical consumption

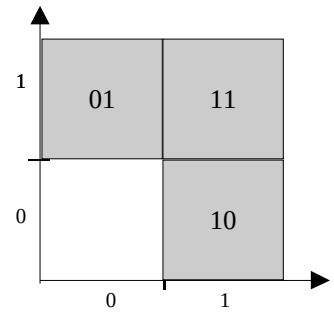
The total electrical consumption must be inferior to 70 W

# Annex 1: mapping of the neighbouring information

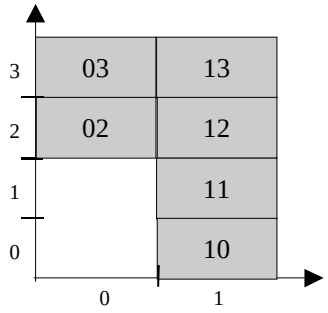
| FOI 5533Y11 limited to12 pads sur M1 | | | | | | | | | | | | | | | | | Total pins per PU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PU01 | PU10101 | PU11101 | PU11001 | PU20301 | PU20200 | PU21301 | PU21200 | PU21101 | PU21000 | PU30101 | PU31101 | PU31001 | PU40101 | PU41101 | PU41001 | | |
| backplane from | 2 | 45 | 3 | 4 | 0 | 42 | 0 | 44 | 0 | 2 | 40 | 46 | 0 | 37 | 46 | 46 | |
| backplane to | 1 | 55 | 9 | 3 | 0 | 52 | 0 | 56 | 0 | 1 | 45 | 40 | 0 | 37 | 34 | 56 | |
| vertical PU00 from | 21 | 21 | 21 | 16 | 0 | 16 | 0 | 16 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| vertical PU00 to | 21 | 21 | 21 | 0 | 16 | 0 | 16 | 0 | 16 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| horizontal PU11 from | 41 | 41 | 41 | 36 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 41 | |
| horizontal PU11 to | 36 | 36 | 36 | 36 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | |
| cross PU10 from | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| cross PU10 to | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| Total neighbor exchanges | | | | | | | | | | | | | | | | 223 | |
| Total service signals | | | | | | | | | | | | | | | | 232 | **455** |
| PU11 | PU10111 | PU11111 | PU11011 | PU20311 | PU20210 | PU21311 | PU21210 | PU21111 | PU21010 | PU30111 | PU31111 | PU31011 | PU40111 | PU41111 | PU41011 | | |
| backplane from | 45 | 14 | 14 | 42 | 0 | 21 | 0 | 21 | 0 | 40 | 17 | 17 | 37 | 0 | 2 | 45 | |
| backplane to | 40 | 14 | 14 | 41 | 0 | 19 | 0 | 20 | 0 | 39 | 17 | 18 | 37 | 0 | 2 | 41 | |
| vertical PU10 from | 21 | 21 | 21 | 16 | 0 | 16 | 0 | 16 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| vertical PU10 to | 21 | 21 | 21 | 0 | 16 | 0 | 16 | 0 | 16 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| horizontal PU01 from | 36 | 36 | 36 | 36 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | |
| horizontal PU01 to | 41 | 41 | 41 | 36 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 41 | |
| cross PU00 from | 6 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | |
| cross PU00 to | 6 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | |
| Total neighbor exchanges | | | | | | | | | | | | | | | | 217 | |
| Total service signals | | | | | | | | | | | | | | | | 232 | **449** |
| PU00 | PU10100 | PU11100 | PU11000 | PU20300 | PU20201 | PU21300 | PU21201 | PU21100 | PU21001 | PU30100 | PU31100 | PU31000 | PU40100 | PU41100 | PU41000 | | |
| backplane from | 0 | 44 | 0 | 0 | 5 | 0 | 42 | 0 | 41 | 3 | 40 | 43 | 3 | 40 | 43 | 44 | |
| backplane to | 0 | 45 | 0 | 0 | 5 | 0 | 53 | 0 | 53 | 5 | 46 | 37 | 5 | 40 | 31 | 53 | |
| vertical PU01 from | 21 | 21 | 21 | 0 | 16 | 0 | 16 | 0 | 16 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| vertical PU01 to | 21 | 21 | 21 | 16 | 0 | 16 | 0 | 16 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| horizontal PU10 from | 36 | 36 | 36 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | |
| horizontal PU10 to | 41 | 41 | 41 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | 41 | |
| cross PU11 from | 6 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | |
| cross PU11 to | 6 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | |
| Total neighbor exchanges | | | | | | | | | | | | | | | | 228 | |
| Total service signals | | | | | | | | | | | | | | | | 232 | **460** |
| PU10 | PU10110 | PU11110 | PU11010 | PU20310 | PU20211 | PU21310 | PU21211 | PU21110 | PU21011 | PU30110 | PU31110 | PU31010 | PU40110 | PU41110 | PU41010 | | |
| backplane from | 43 | 17 | 14 | 0 | 43 | 0 | 21 | 0 | 18 | 41 | 17 | 14 | 41 | 2 | 0 | 43 | |
| backplane to | 48 | 18 | 15 | 0 | 43 | 0 | 20 | 0 | 17 | 43 | 36 | 15 | 43 | 2 | 0 | 48 | |
| vertical PU11 from | 21 | 21 | 21 | 0 | 16 | 0 | 16 | 0 | 16 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| vertical PU11 to | 21 | 21 | 21 | 16 | 0 | 16 | 0 | 16 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 21 | |
| horizontal PU00 from | 41 | 36 | 36 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | 41 | |
| horizontal PU00 to | 36 | 41 | 41 | 0 | 36 | 0 | 36 | 0 | 36 | 36 | 36 | 36 | 36 | 36 | 36 | 41 | |
| cross PU01 from | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| cross PU01 to | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| Total neighbor exchanges | | | | | | | | | | | | | | | | 217 | |
| Total service signals | | | | | | | | | | | | | | | | 232 | **449** |

***Table 51: count of neigbouring exchanges and pinout occupancy for the PU FPGAs***

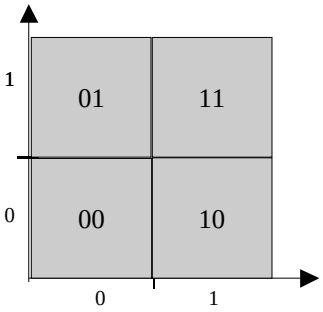
The PU numbering system is explained in Figure 19 next page.




Board numbering in regions 1, 3 and 4




Board numbering in region 2




PU numbering within a board

**Figure 19: PU numbering system.**

Example the upper left PU of the upper right board is numbered 41101
(4=region 4, 11=board number, 01= PU number)

## Annex 2: First position window calibration procedure

To correctly synchronise its FIFOs, the muon trigger system must bring a special care to the detection of the first incoming data for each channel of transmission. This operation is critical since any false detection desynchronises the system for an entire machine cycle (3564 bunch crossings). In order to reduce the probability of false detection, the principle that has been retained is to monitor in a limited time window the occurrence of such an event.

The optimum **position** and **size** of this window is only approximately known, because it depends on the propagation time of the bunch crossing information through the off−detector electronic and through the optical fibres. This propagation time will be better known when the corresponding parts will have been designed. However, it is possible that this propagation time move slightly because environmental conditions change or because some parameters have been changed in the front−end systems. It is therefore necessary to be able to adjust this time window both in width and in position. This can be done in a calibration phase.

The hardware module corresponding to this calibration phase does not need to be included in the nominal design described in the present specification. In order to save some place inside the operational PU FPGAs, the following strategy has been adopted:

1. During the early life of the system, a specific programming of the PU FPGAs is used in order to measure these parameters in the final CERN set−up.
2. Experimental parameters are stored (average position of the first data, maximum variation).
3. The PU FPGAs are programmed with the nominal configuration described in this document.
4. Stored parameters are entered in the configuration registers (see paragraph 6.11.4.1 page 55)

The specific programming of the FPGA includes a measurement module that can be read through the ECS interface as shown in Figure 20.

For each reception channel, a 7 bits[7] counter is reset on reception of BC0_Reset

7 The time for receiving the first data after a BC0_reset is approximately 800 ns. At 80 MHz this corresponds to 64 clock cycles. By security we will extend it to the next power of 2 (128 clock cycles). The number of bits needed for implementing the sampled counter value and the reference position value is then 7 bits.

and incremented by the clock extracted from the incoming data of this channel. The current value of the counter accompanied with a *counter value valid* bit is then sent to the local ECS interface on detection of the first incoming data. The embedded PC continuously scans the *counter value valid* bit of the register. When the bit is set, the counter value can be read. After the reading the register is automatically reset.

Then the embedded PC builds an histogram of the different values. The average value gives the central position of the monitoring window. The width of the window is obtained by increasing the maximum offset detected around the central position by a security factor.

These 2 parameters are stored for a further write in the registers of the ECS interface of the nominal programming of the PU FPGAs.

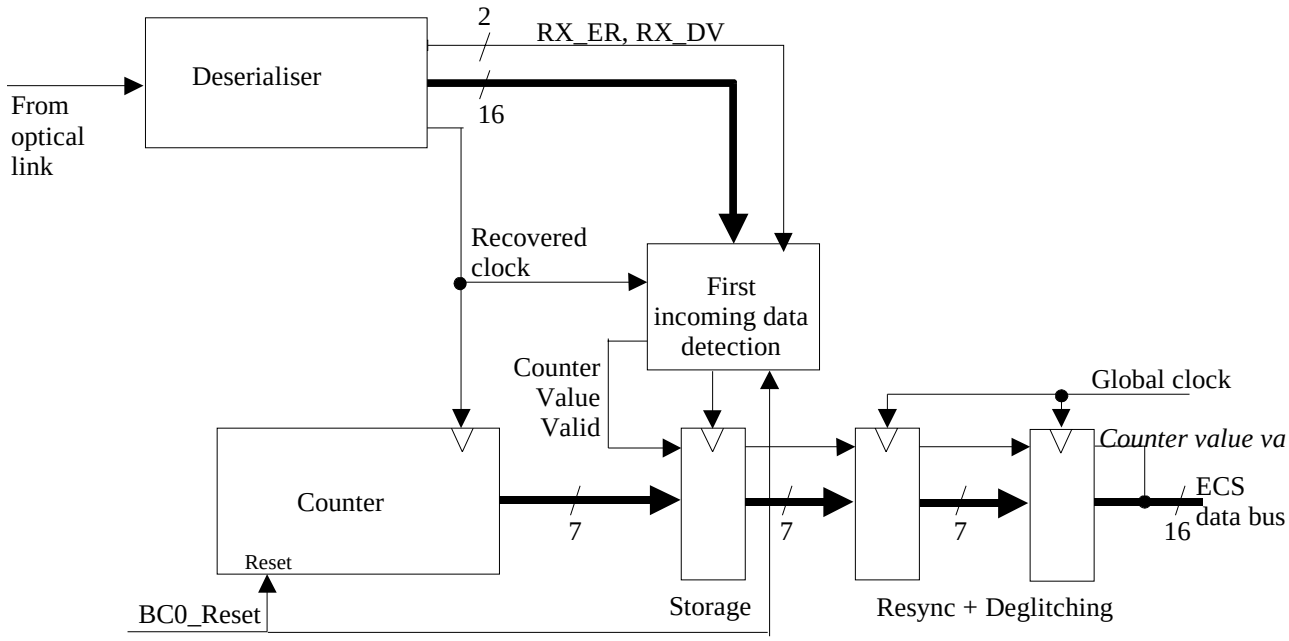



**Figure 20: First data position detection**

The proposed implementation of the registers allowing to read this information is shown in Table 52.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Valid 1 | Sampled Count 1 | | | | | | | *R* |
| | | | | | | | | Valid 2 | Sampled Count 2 | | | | | | | *R* |
| | | | | | | | | Valid 3 | Sampled Count 3 | | | | | | | *R* |
| | | | | | | | | Valid 4 | Sampled Count 4 | | | | | | | *R* |
| | | | | | | | | Valid 5 | Sampled Count 5 | | | | | | | *R* |
| | | | | | | | | Valid 6 | Sampled Count 6 | | | | | | | *R* |

***Table 52: First position reading register***

**Valid i** : Sampled count is valid when set to 1. Reset after reading.
**Sampled count** : Number of 80 MHz clock cycles received since a BC0_reset before the first data is detected.

## REFERENCES

[1] E. Aslanides et al., A synchronous architecture for the muon trigger, LHCb Note LHCb 2001−010

[2] B.Schmidt et al., LHCb Muon System by Numbers, LHCb note, LHCb 2000−89

[3] J. Christiansen, Key parameters and features of the LHCb front−end electronics , http://lhcb−elec.web.cern.ch/lhcb−elec/html/key_parameters.htm

[4] J. Christiansen, Requirements to the Front−End electronics, LHCb Note LHCb 2001−014 second version

[5] J. Christiansen, Requirements to the L1 electronics", LHCb Note LHCb 2001−127

[6] J. Christiansen et al., The latency of the L0 trigger, LHCb Note LHCb 1999−015