EDMS #704078-v.1.0
LHCb 2005 – ??
August 4, 2005

LHCb readout supervisor 'ODIN' with a L1 trigger

Technical Reference

Z. Guzik and R. Jacobsson

ABSTRACT

This note describes the final version of the Odin board in all its technical aspects. Eventually it is also meant to serve as a user guide with all the necessary information to set it up and operate it in a sub-detector test stand.

This version of the document describes the functionality as it is implemented for a readout system with a L1 Trigger. However, the board has been validated as well in the 1 MHz readout scheme without a L1 Trigger. The implications and the differences in the functionality are discussed in Section 24.

Note that the VHDL code of Odin continues to evolve. To get the latest updated description of the functionality, please, download this document from EDMS [??]

LHCb Technical Note

Issue: LHCb 2005 – ?? Draft 0
Reference: LHCb Online
Created: 10 August 2005
Last modified: 22 August 2005

Prepared By: R. Jacobsson, CERN, Geneva, Switzerland
LHCb Online Group

INDEX

1 Introduction

The Timing and Fast Control (TFC) system [??], shown in detail in Figure 1, drives the LHCb readout by distributing synchronously timing, trigger and control information to the Front-End electronics via the TFC distribution network. The entire TFC mastership is located in the Readout Supervisor "ODIN" [??].

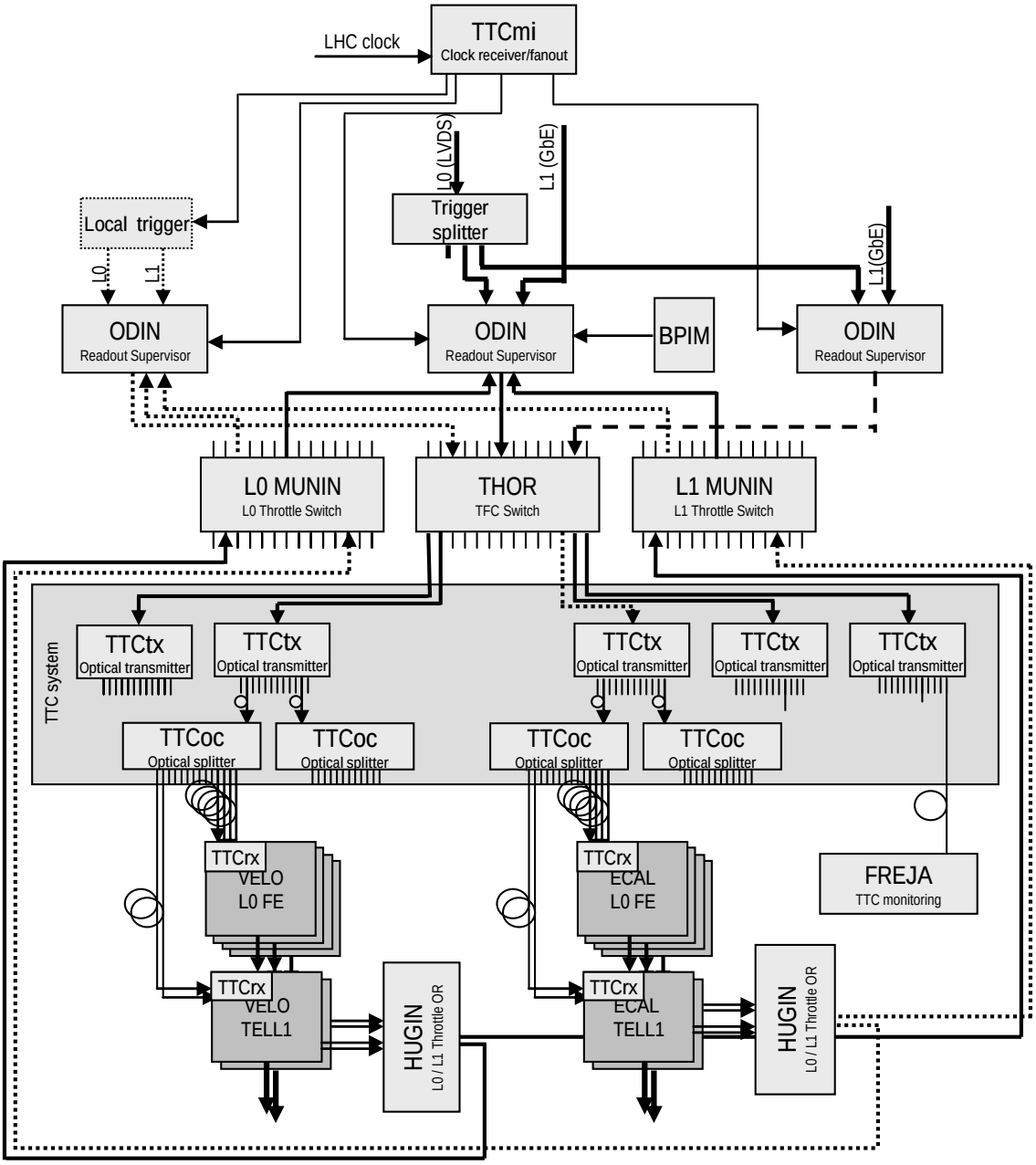


Figure 1: Overview of the TFC system.

2 Board Overview

2.1 Principal Functions

Figure 2 shows a schematic overview of the principal functions of Odin. Figure 3 shows a picture of the Odin board.

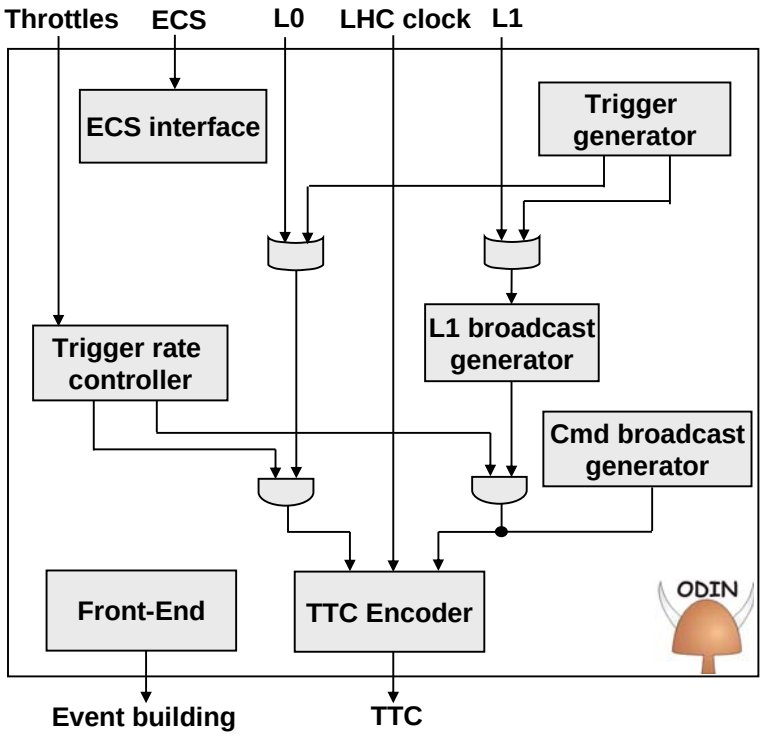


Figure 2: Overview of the principal functions

Odin is designed to receive the L0 physics triggers from the L0 Decision Unit (L0DU)[??] and the L1 physics triggers from the L1 trigger processing in the CPU farm in order to distribute the decisions synchronously to all front-end electronics boards. It verifies that the decision processing is synchronized by checking the event identifiers accompanying the trigger decisions against internal counters.

It also provides several means for auto-triggering to be used in conjunction with tests and calibration runs: auxiliary external trigger, random trigger, periodic trigger, calibration trigger, timing trigger, trigger sequencing etc.

If the total trigger rate at L0 or L1 gets abnormally high or data congestion occurs in the system, there is a potential risk of overflow in the buffers in the front-end electronics or the DAQ system. In order to prevent it, Odin controls the trigger rates according to the status of the buffers. The buffer levels are either centrally emulated in Odin or they are monitored locally. In case they are monitored locally, imminent overflows are signalled to Odin via a dedicated throttle network if it is in the TELL1/UKL1 boards and via the Experiment Control System if it is in the readout network. The trigger rate is also controlled during L0 and L1 electronics resets.

Odin is responsible for generating and broadcasting a number of synchronous control commands

and information to drive the readout. These broadcasts consist of:

- Commands resetting event related counters in the front-end electronics used to identify the accepted events and to check synchronisation.
- Commands resetting the front-end electronics in order to prepare it for data taking or to recover from an error condition.
- Calibration commands activating specific calibration systems in the front-end electronics or in the sub-detectors.
- IP/Ethernet addresses assigning the destinations in the CPU farm destinations to which the TELL1/UKL1 boards should send the L1 trigger data and the HLT data, respectively.

Odin accumulates statistics on the performance and the efficiency of the data acquisition. It also records local event information which are sent and appended to the event data in the CPU farm as any detector front-end.

The TTC encoder circuit incorporated in each ODIN is built to receive directly the LHC clock and the LHC orbit signal from the TTC machine interface (TTCmi in Figure 1). The clock, the trigger decisions, and the synchronous control commands and information are encoded and transmitted by the TTC encoder to the front-end electronics as a TTC signal [??].

All the programming, configuration, control and monitoring is handled from the Experiment Control System via an onboard Credit Card PC connected to Ethernet.

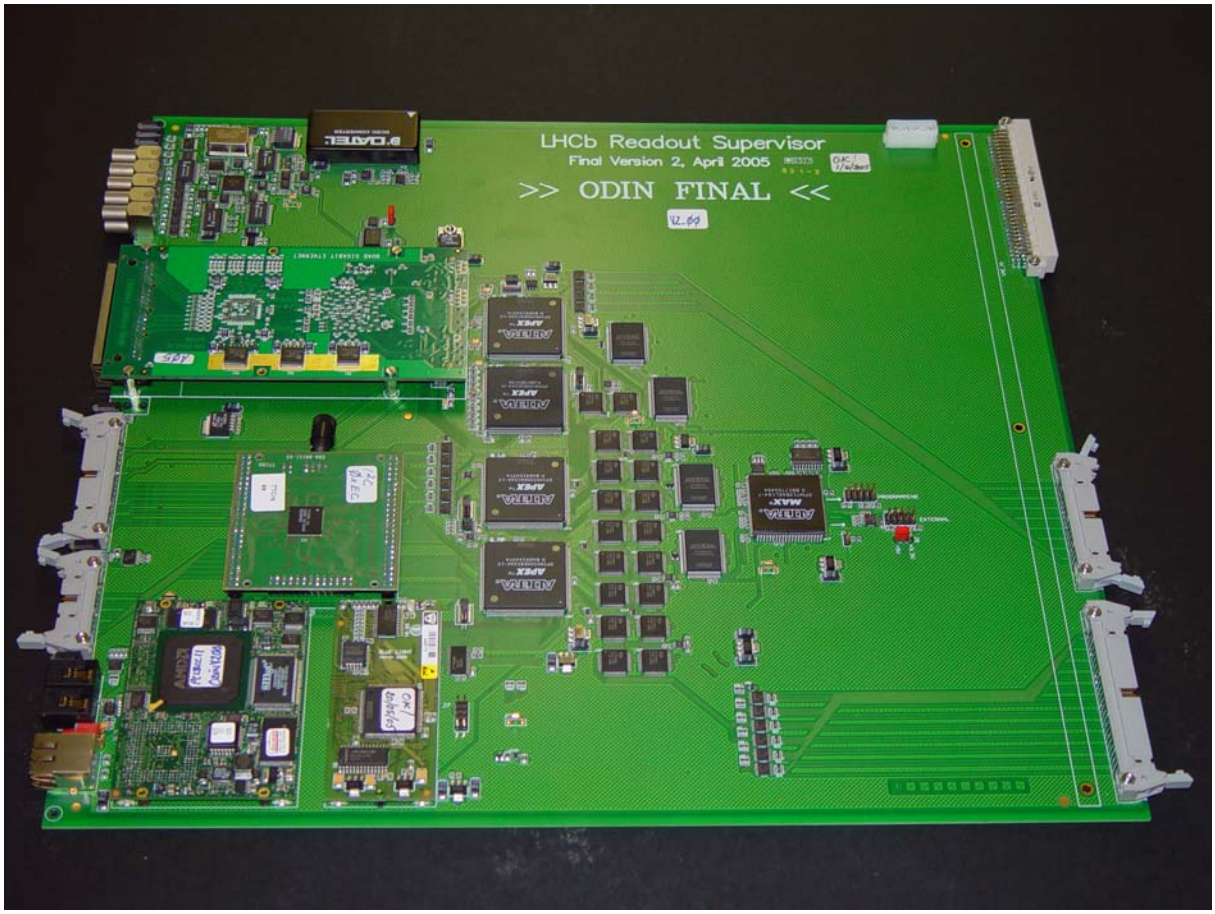


Figure 3: Picture of the final Odin.

2.2 Overall Dataflow

The data flow is show in Figure 4. All TFC functions are organized in four FPGAs:

- Q_L0 : L0 Trigger handling
- Q_L1 : L1 trigger handling
- Q_MP : Multi-purpose module (synchronous control command generation, calibration generation, IP/Ethernet destination handling, TTC broadcasting etc)
- Q_FE : ODIN Front-End

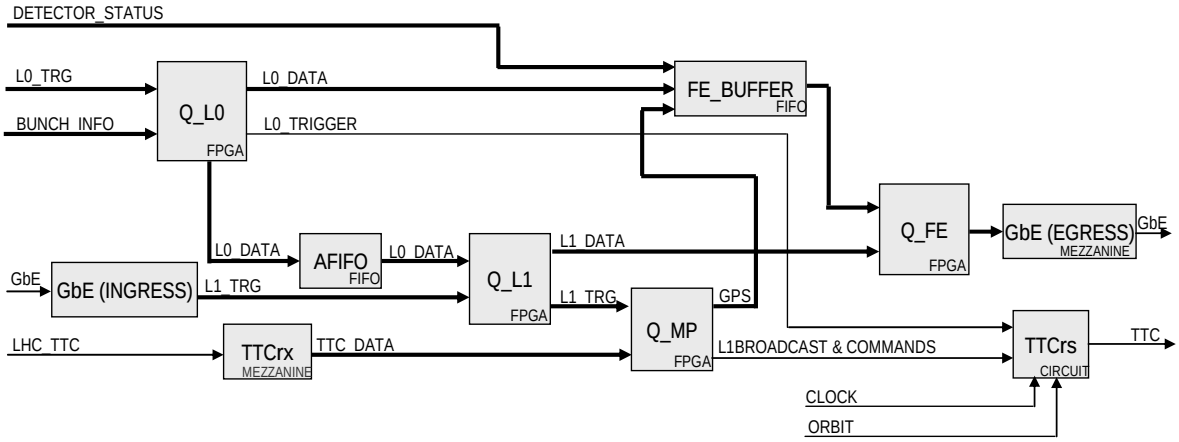


Figure 4: Diagram of the principal dataflow

The AFIFO is a discrete FIFO which allows passing information between the L0 trigger handling and the L1 trigger handling such as trigger types, event identifiers for synchronization purposes etc.

Local event data recorded at each L0 trigger accept is written to the Front-End Buffer (FEB) and is readout at every L1 trigger decision. The information consist of event identifiers, trigger type, detector status, and GPS time as received from the LHC Beam Synchronous Timing system by the onboard TTCrx. The L1 triggers are normally received via Gigabit Ethernet by the GbE mezzanine[1]. If the L1 trigger decision is positive the data is kept and formatted together with data related to the L1 trigger decision by the Q_FE module. The event data fragment is sent via the GbE mezzanine.

The TTCrs module represents the discrete circuit implemented directly on the ODIN motherboard which receives the LHC clock and orbit signal, and which forms the TTC signal.

2.3 FPGA Technology

The four main FPGAs on Odin are of the Altera APEX EP20KE family. The core is run at 1.8V and the I/O interfaces at 3.3V. Table 1 lists the type of FPGA together with the usage for all the functions described in this document.

[1] Optionally via LVDS for a local trigger module

Table 1: Summary of types of FPGAs and resource usage. *All available IO pins are used but this includes several spare inter-connections per FPGA. **The usage is only meant as an indication

Module	FPGA type	Speed grade	Pins / I/O pins*	Gates used**	Memory bits used**
Q_MP	APEX 20K200E	-1	240/123	52%	23%
Q_L0	APEX 20K200E	-1	240/123	67%	3%
Q_L1	APEX 20K200E	-1X(PLL)	240/123	~35%	100%
Q_FE	APEX 20K100E	-1X (PLL)	240/138	~50%	50%

The PLLs in the Q_L1 and the Q_FE FPGA generate the 80 MHz clocks for the SPI-3 bus used to receive and transmit data with the GbE mezzanine.

The exact organization of the functionality described below in the four FPGAs and the FPGA I/Os lines are shown in the block diagrams (Figure 17 - Figure 20) in Appendix A.

2.4 Signal Technologies

Table 2 lists the principal signal technologies used on ODIN

Table 2: Principal signal technologies used on Odin.

Application	Signal Technology
Clock distribution	LVPECL
JTAG	LVTTL
Local Bus	LVTTL
I2C	LVTTL
FPGA I/O	LVTTL
TTC Encoder	NECL, PECL, LVPECL
Clock and orbit reception	NECL
L0 Trigger Input	LVDS
Detector status	LVDS
L1 Trigger Input	LVDS

3 Board dimensions and mechanics

The Odin board is a 9U x 400mm VME board. It uses the same backplane as the TELL1/UKL1 board [??].

The board thickness is about 2.7mm to increase the stiffness[2]. Since the VME rails are made for 1.6mm boards, Odin is milled on the top and bottom edges.

4 Board power

The only external supply voltage of the board is +5V. All other voltages, 1.8V, 2.5V and 3.3V, are made on the board using regulators and -5V using a DC/DC converter. The different supply voltages and applications are listed in Table 3.

[2] For increased rigidness a PCB bar has been foreseen and will be fitted later

Table 3: Summary of the different voltages used.

Application	Voltage	Supply device
Main FPGA core	1.8V	2 x LT1963AEST-1.8 (1.5A)
JTAG to GbE mezzanine Special SPI-3 I/Os to GbE	2.5V	1 x LT1963AEST-2.5 (1.5A)
FPGA I/O LVPECL devices FIFOs I2C devices JTAG hub EPC4 configuration devices	3.3V	7 x LT1963AEST-3.3(1.5A)
LVDS receivers TTCrx LVPECL clock fan-out	3.3V	1 x LT1764AEQ-3.3 (3A)
GbE mezzanine	3.3V	1 x LT1084CT-3.3 (5A)
Encoder PLL PECL devices in TTC encoder Credit Card PC Glue *light* mezzanine	+5V	-
TTC encoder PLL	-3.3	1 x LM2991 (1A)
NECL devices in TTC encoder	-5V	Datel UWR-5/1600-D5(1.6A)

The +5V main supply is monitored by a circuit which detects a bad power if the voltage is outside the range 4.7V – 5.3V. A bad power is indicated by the LED D1-L blinking red/green. Provided the voltage still allows the FPGAs and the control interface to work, a bad power is also indicated by the status bits S_ERR_PWR_CONT and S_ERR_PWR_INST. The first indicates that the power is out of range continuously and the latter that the power went out of range during only a short period. S_ERR_PWR_INST is sampled with the system clock and is reset together with resetting all counters.

Table 4: Power status bits.

Parameter	Function	Address	Bits	Read/Write	Remark
S_ERR_PWR_CONT	Power error continuous	0x1060	8	RO	
S_ERR_PWR_INST	Power error instantaneously	0x1060	9	RO	Reset by counter reset

The board can be powered in two different ways. Normally the board takes its power from the back plane (Figure 5), which is the same as for the TELL1 board (+5V: row 1, 2, 3, and 4. GND: row 8, 9, 10, 14, 15, and 16). However, in order to allow operating the board outside a crate it is equipped with a PC power header with a pin configuration as shown in Figure 6.

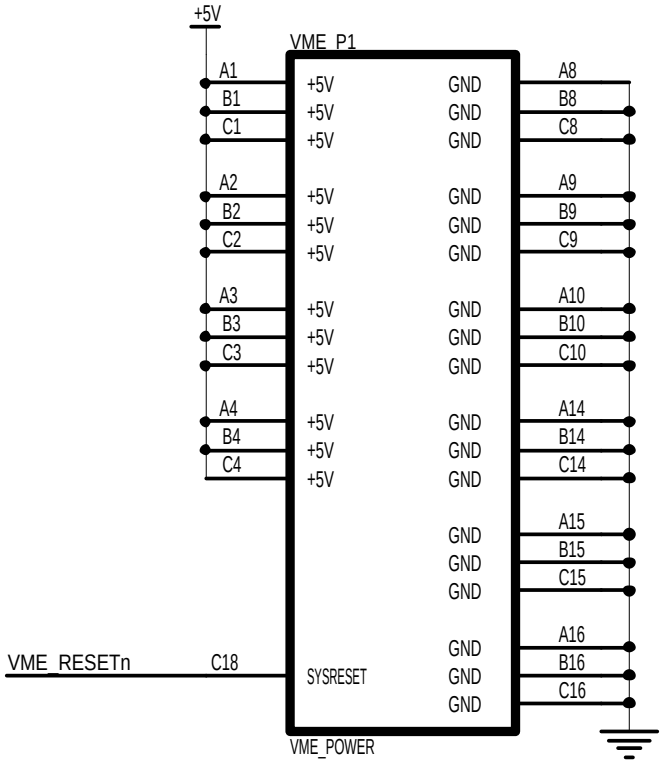


Figure 5: Pin configuration of the VME backplane connector.

The total idle power consumption of Odin including all the mezzanines amount to approximately 6.8A on 5V. In full operation including two active Gigabit Ethernet links this increases to about ??A.

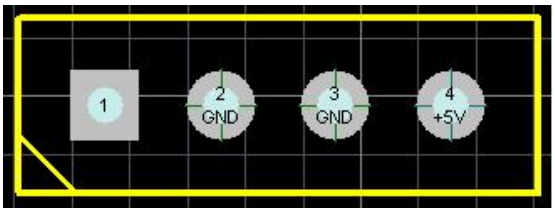


Figure 6: Pin configuration of the power header.

Experience shows that the board can run without cooling in an open lab environment for extended periods. However, the TTC encoder circuit is based on ECL and gets extremely hot. The LEMO connectors will soon reach a temperature of up to 80^{O}C and work as cooling towers[3]! The Gigabit Ethernet interface also consumes a considerable amount of power during operation and requires cooling. Therefore it is inadvisable to run the board outside a cooled environment.

5 Board Control and FPGA Programming

All the board logic is programmed, configured, controlled, and monitored via the Credit Card PC [??] which is connected to the Experiment Control System via Ethernet. Odin makes use of four

[3] To be safe we are considering adding cooling towers to a few of the chips in the TTC encoder unit.

different board bus types:

- JTAG : Boundary Scan and device programming
- 4-bit control bus : JTAG hub control
- I^2C : For peripheral I^2C compliant control devices and EEPROMs
- 32-bit parallel multiplexed bus: Control and monitoring of all the functionality in the FPGAs

The Glue *light* mezzanine card [??] is used to produce the four buses from the PCI bus of the CCPC. A single FPGA on the mezzanine emulates a PLX9030 and forms directly a PLX Local Bus, and the I2C and the JTAG buses. The eight General Purpose I/O lines of a PLX9030 are also emulated. On Odin they are used to drive the JTAG hub control bus.

The only supply voltage for the CCPC and Glue *light* mezzanine is +5V.

5.1 Ethernet connection

The Ethernet connection for the Credit Card PC is located on the front-panel (P15). Note that a long Ethernet cable between the nearest switch or hub and the board may cause problems to the Credit Card PC which manifests itself as none or intermittent loss of network communication. The remedy is to use a cable as short as possible or to insert a simple hub as close as possible to the board.

The LEDs on the RJ45 connector itself are not used.

5.2 JTAG bus

Figure 7 shows a block diagram of the JTAG bus. ODIN incorporates nine devices[4] which require JTAG for boundary scanning and in-situ programming: the four main FPGAs (Q_MP, Q_L0, Q_L1, Q_FE), four configuration devices (Altera EPC4Q100) for the main FPGAs, and the Gigabit Ethernet mezzanine which in itself has two JTAG devices (the Intel MAC IXF1104 and the Marvell PHY Alaska Quad 88E1141).

A JTAG hub distributes the JTAG bus in a star fashion to the nine devices. It is implemented in a small EEPROM-based PLD (Altera MAX EPM7128AELC84-7) which retains its programming after power down but can be reprogrammed via the JTAG header J3 (5V). The hub allows forming dynamically a JTAG chain with any combination of the nine devices for either boundary scanning or programming via the JTAG hub control bus. The control bus is connected to the general Purpose lines of the Glue mezzanine and is organized according to Table 5. The selection registers in the hub are listed in Table 6. The reset line resets all the internal registers of the hub and puts the JTAG buses in their idle states.

[4] The TTCrx JTAG bus has unfortunately not been associated

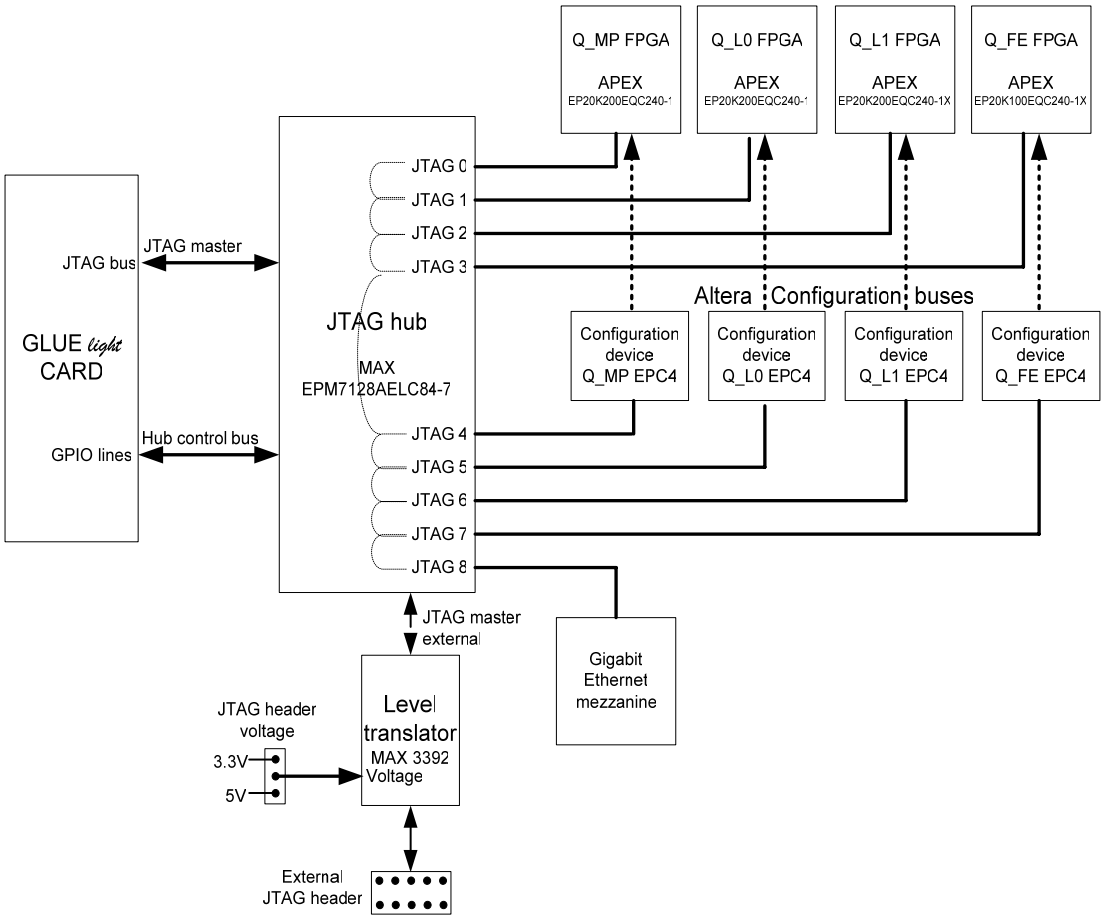


Figure 7: Block diagram of the JTAG architecture

Table 5: Organization of the JTAG hub control bus

GPIO(7)	GPIO(6)	GPIO(5)	GPIO(4)	GPIO(3 .. 0)
RESET	WRITE_ENB	ADDR(0)	ADDR(0)	DATA(3 .. 0)

The JTAG hub has two JTAG master sources. Under normal circumstances the JTAG hub is driven by the JTAG bus from the Glue *light* mezzanine (internal mode). However, there is also a JTAG header J5 for an external JTAG source which can either be 3.3V or 5.0V by selecting the voltage with the jumper J8 (external mode). The external or internal mode is selected in the third internal register of the hub. The internal JTAG bus is operated at approximately ?? MHz.

Table 6: JTAG hub control registers

Register	Address	Bit 3	Bit 2	Bit 1	Bit 0
FPGA selection	“00”	Q_FE	Q_L1	Q_L0	Q_MP
EPC4 selection	“01”	EPC_FE	EPC_L1	EPC_L0	EPC_MP
Other	“10”	Ext/Int JTAG	Not used	GbE nTRST	GbE
Not used	“11”				

All the JTAG buses of the hub are 3.3V except for the one to the Gigabit Ethernet mezzanine which is translated to 2.5V. All the JTAG clock lines are filtered using a 74HC244 and 100ohm serial resistors to minimize reflections. The TMS, TDI and TDO lines all have 33ohm serial resistors.

JTAG programming and boundary scanning can be made directly from the command line of the Credit Card PC. The devices are selected in the JTAG hub by using the command "`devsel`" and listing the devices by their location on the JTAG star as shown in Figure 7 (e.g. Q_FE and GbE: `> devsel 4 8`).

- Programming: `> jbi_pci -aconfigure file.jbc`
- JTAG chain test and IDCODE : `> bscan –i<number of components>`
- Boundary scan : `> bscan –b<file.pat>`

5.3 I^2C bus

Odin has five devices controlled by I^2C:

- 256 x 8-bits Serial EEPROM (Microchip 24LC024) used to store the board identifier
- 8-bit I/O port (Philips PCA9554) used to control hardware settings such as system reset, clock source selection etc.
- 16-bit I/O port (Philips PCA9555) used to control the LHC clock delay
- TTCrx configuration registers
- Gigabit Ethernet mezzanine on which there is the same serial EEPROM as above for the board identifier

The five devices are all located on a single I^2C chain. The I^2C bus speed is approximately 100 kHz. The peripheral addresses are listed in Table 7. Note that in the nomenclature of the TFC control system the peripheral address of an I^2C device is 8-bit aligned.

Table 7: I^2C devices on Odin.

Function	Chip	Peripheral address
Odin board identifier	24LC024	0xA0
Hardware settings	PCA9554	0x40
Clock adjustment	PCA9555	0x42
TTCrx configuration	TTCrx	0xEC
GbE mezzanine identifier	GbE	0xAE

The I^2C can be accessed directly from the Credit Card PC on the command line using the command "I2C" according to:

- Read access : `> i2c –r -i<internal register> <peripheral address>`
- Write access: `> i2c –w<value> -i<internal register> <peripheral address>`

The write is automatically followed by a read. Both the peripheral and the internal address and the value must be given in hexadecimal form. The read access returns both in hexadecimal and in decimal form.

5.4 Local Bus

The Odin Local Bus runs as a multi-drop bus over the four main FPGAs (Q_MP, Q_L0, Q_L1 and Q_FE) and is implemented according to specifications of the synchronous address/data multiplexed version of the PLX9030 Local Bus. The bus signals are listed in Table 8. The bus clock is synchronous with the global system clock. It is derived directly by dividing the system clock by two, converting it to LVTTL and fanning it out using a line driver (74ALVC16244). The bus clock frequency is thus approximately 20 MHz.

All the Odin TFC functionality is set up and activated via a set of Local Bus registers in each of the FPGAs. The Local Bus Slave is implemented as a generic VHDL component in each FPGA which is also used on all the other TFC boards.

Table 8: List of the Local Bus signals

Signal	Name	Type	Function
LCLK	Local Bus Clock	O	Local bus clock, approximately 20 MHz
LRESETn	Local Bus Reset Out	O	Asserted when the CCPC is reset
LAD[31..0]	Address/Data Bus	I/O	During the address phase, the bus carries the address of the device on the [15..2] lines. During the data phase the bus carries 32-bit data words
LADSn	Address Strobe	O	Indicates a start of a new bus access and valid address. Asserted during the first clock cycle of the bus transaction
LW_Rn	Write/Read	O	LOW for read accesses and HIGH for write accesses
LWAITn	Wait Out	O	Asserted by the master to insert wait states
LRDYn	Local Ready Input	I	Indicates valid read data on the bus in a read access or that write data is accepted in a write access. The signal is not sampled until LWAITn is deasserted
LLASTn	Burst Last Data	O	Asserted by the master to indicate the last data transfer in a bus access
LTERMn	Burst Terminate	I	Asserted by the slave to terminate a burst access

The Local Bus can be accessed directly from the Credit Card PC on the command line using the command "`lbus`" according to:

- Read access : `> lbus –r <register>`
- Write followed by read access : `> lbus –w<value> <register>`
- Write only access : `> lbus –p<value> <register>`

Both the register address and the value must be given in hexadecimal form. The read access returns both in hexadecimal and in decimal form.

5.5 FPGA Programming and Version

The four FPGAs (Q_MP, Q_L0, Q_L1 and Q_FE) are normally loaded directly on power-up by the four Altera EPC4 configuration devices via the Altera native programming bus. The LED D2-L indicates if all the four FPGAs are programmed.

The eight devices can all be programmed individually from the Credit Card PC by selecting the device in the JTAG hub and using the TFC STAPL player "`jbi_pci`":

```
> jbi_pci  –aconfigure Q_<device>.jbc
```

Each FPGA has a VHDL version register which can be read via the local bus. It is located at <base address+0xFC> and the format in decimal form is YEAR(4) : MONTH(2) : DAY(2) : "HOUR"(2),

that is for instance 2005071100. The base addresses are given in Table 12.

6 System Reset and other hardware control lines

The I^2C I/O port with peripheral address 0x40 controls a set of hardware settings as listed in Table 9. Note that the I/O ports must be configured as outputs before by setting all bits in the direction register to LOW.

Table 9: Output configuration of the I2C I/O port (0x40) driving the Odin hardware settings.

Internal register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Ouput port register (set out)	0x1	System reset	L1 Trigger LVDS	Encoder reset	Ext/Int System clock	Not used	Not used	Bunch info input	Detector status input
Port direction register	0x3	0	0	0	0	0	0	0	0

Setting bit 0 enables a 24-bit LVDS status input on the rear as described in Section 18. Setting bit 1 enables an 8-bit LVDS bunch information input on the front-panel which is described in Section 19. Bit 4 is used to select between the internal clock and orbit signal generator (cleared) or an external source (set) for the global system clock and the orbit signal as described in Section 7 and 8. Bit 5 is used to reset the TTC encoder. Note that this will remove the system clock from the board but not the control of the I2C bus of course! Odin has an optional input for L1 Triggers which is based on a 16-bit LVDS input. It is enabled by setting bit 6. The system reset is controlled via bit 7.

6.1 System Reset

The system reset puts all functions of Odin in the reset state except for the global system clock internal generation, clock distribution, the I^2C bus and the JTAG bus. Note that it clears all Local Bus registers and that the Local Bus is held in its reset state.

The system reset is set by default on power up. It is then controlled via bit 7 of the I2C port on peripheral address 0x40 as shown in Table 9. The status of the system reset is indicated by the LED D1-R which is red when the system reset is set and green when it is cleared.

6.2 Control Interface Reset

There is a separate reset of the Credit Card PC and the Glue mezzanine. The reset can be activated either via the push button on the front-panel or via the reset line (C18) on the backplane which is common to all boards in the crate.

7 Clock Distribution

Odin has two alternative sources for the system clock (approximately 40.079 MHz), an internal and an external, which can be selected using bit 4 of the I^2C port on peripheral address 0x40. The current selection is indicated by the LED D4-L. It is red for internal and green for external clock.

The internal clock (bit 4 = 0) used for stand-alone running is generated in the TTC encoder unit by an 80.158 MHz VXCO ECL oscillator.

The external clock is received by the TTC encoder unit via the AC-coupled ECL input P2-L (LEMO connector) on the front-panel. In the experiment the external clock is normally received

from the LHC and corresponds to the bunch arrival frequency in the experiment. The presence of an external clock is indicated by the LED D3-L on the front-panel.

In order to be able to fine adjust the global timing of the entire experiment the external clock is fed through a differential LVPECL temperature compensated programmable delay chip (100EP195) with a range of 2.2ns to 12.2ns in steps of 10ps. The delay chip is programmed via the I^2C I/O port on the peripheral address 0x42. Note that the I/O ports must be configured as outputs before by setting all the bits in the direction registers to LOW. The delay configuration bus consist of ten data pins D(9..0) and a latch enable LEN which are driven from the I^2C register according to Table 10. A LOW level on the latch enable allows a transparent load mode of delay values by D[9:0]. A LOW to HIGH transition on the latch enable will lock and hold current values irrespective of any subsequent changes in D[9:0]. Table 11 shows the principal of the delay programming.

Table 10: Output configuration of the I^2C port (0x42) driving the programmable clock delay.

Internal register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Output port register 0	0x2	D(7)	D(6)	D(5)	D(4)	D(3)	D(2)	D(1)	D(0)
Output port register 1	0x3	LEN						D(9)	D(8)
Port direction register 0	0x6	0	0	0	0	0	0	0	0
Port direction register 1	0x7	0	0	0	0	0	0	0	0

Table 11: Example of delay values

D(9:0) Value	Programmable Delay
0000000000	0 ps
0000000001	10 ps
0000000010	20 ps
0000000011	30 ps
0000000100	40 ps
0000000101	50 ps
0000000110	60 ps
0000000111	70 ps
0000001000	80 ps
0000010000	160 ps
0000100000	320 ps
0001000000	640 ps
0010000000	1280 ps
0100000000	2560 ps
1000000000	5120 ps
1111111111	10230 ps

The clock is subsequently fed through a PLL clock driver which is also used to regenerate the 40MHz clock and to generate the 2:1 and the 4:1 clocks for the TTC signal. The clock is distributed on the board as differential LVPECL in a star fashion using clock fan-out. The translation from LVPECL to LVTTL is made as close as possible to each clock destination.

The current clock is also output via the P2-R LEMO connector on the front panel as AC-coupled ECL.

8 Orbit Signal

The orbit signal marks a full turn in LHC (3564 clock cycles or 88.92 μs) and is used in Odin to synchronize all TFC functions.

There are two alternative sources of the orbit signal, an internal and an external. The selection is concurrent with the clock source selection using bit 4 of the I^2C port on peripheral address 0x40. The LED D4-L indicates if an external (green) or an internal (red) clock/orbit signal has been selected.

The internal orbit signal is generated in Q_MP. The internal orbit generator (ODIN_orbit.vhd) has the special feature that the length of an orbit can be programmed using the parameter P_ORBIT_LEN. This value does not influence running with the external orbit

Parameter	Function	Address	Bits	Read/Write	Remark
P_ORBIT_LEN	Length of LHC orbit	0x1014	11 .. 0	R/W	

The external clock is received via the DC-coupled ECL input P1-L(LEMO connector) on the front-panel. The presence of an external clock is indicated by the LED D3-R on the front-panel. Since the LHC orbit pulse is approximately 1 us long, an edge detect function in Q_MP is used to regenerate the external orbit pulse distributed on Odin.

The orbit signal is distributed by Q_MP to all the FPGAs. The current orbit signal is also output via the P1-R LEMO connector on the front panel as DC-coupled ECL. Note that the pulse is not stretched and is only one clock cycle long.

9 Organization of Control and Status Registers

All the TFC functions of Odin can be controlled via Local Bus registers in the four FPGAs. Table 12 lists the Local Bus base addresses of the four FPGAs. All the registers are 32-bits wide and each may contain several different parameters. The parameters are all aligned to 4-bits.

Table 12: Local Bus base address of each FPGA.

Module	LBUS base address
Q_MP	0x1000
Q_L0	0x2000
Q_L1	0x3000
Q_FE	0x4000

The configuration parameters are organized in three different categories according to the type of usage. The type is defined by the prefix:

- H_parameters : Hardware parameters that would not normally be changed once the board is installed and cabled.
- P_parameters : Parameters which may be changed to configure different types of running modes. They may not be changed during actual data taking.
- R_parameters: Run related parameters which control the data taking operation.

There is also a set of parameters with the prefix DMND_ for “demand”. They are used to make a request via the ECS to Odin to perform a single shot of a specific function. Most of these actions are synchronous and the operation will be performed at the specific time which has been defined by the configuration parameters. The demands may be made during data taking as well as out of data taking. As discussed later there are status bits set while the demand is pending. The request may also be cancelled by clearing the demand bit.

All functions of Odin may be reset individually via the ECS interface. The resets are asynchronous and should not be used during data taking. The reset bits have the prefix RST_.

The operation of all functions is recorded by a large set of counters, several of which record the performance of the experiment (dead-time etc.). The counters have the prefix C_. In addition there are several registers containing status bits indicating the status of a specific function or buffer etc. The status bits have the prefix S_. The full list of counters and status registers is in Appendix G.

In order to get a consistent picture of the status of the system, all counters and status registers are sampled simultaneously in temporary buffers waiting to be read out via the ECS interface. The update action is triggered by reading the register UPDATE_CNT.

The counters and the status registers may be reset simultaneously with the RST_CNT bit. This is an asynchronous action with respect to the system clock and should not be used during data taking. All counters may also be reset individually by using the registers RST_SEL_CNT in each FPGA. Setting a bit in this register will reset the corresponding counter. The status bits cannot be directly reset individually. They are reset by resetting the function which they monitor.

All the counter processing is implemented in four sub-modules to the four FPGAs: ODIN_MP_counters.vhd, _L0_counters.vhd, _L1_counters.vhd, _FE_counters.vhd.

Table 13: Summary of control registers related to status bits and counters.

Parameter	Function	Address	Bits	Read/Write	Remark
UPDATE_CNT	Update all counters synchronously	0x1000	-	RO	
RST_CNT	Reset all counters simultaneously	0x1004	0	R/W	
RST_SEL_CNT in Q_MP	Reset individual counters in Q_MP	0x1064	31 .. 0	R/W	
RST_SEL_CNT in Q_L0	Reset individual counters in Q_L0	0x1054	31 .. 0	R/W	
RST_SEL_CNT in Q_L1	Reset individual counters in Q_L1	0x1034	31 .. 0	R/W	
RST_SEL_CNT in Q_FE	Reset individual counters in Q_FE	0x1034	31 .. 0	R/W	Not yet final

10 Board Identifier

The identifier of the board is stored in the I^2C EEPROM (Microchip 24LC024) at the peripheral address 0xA0. In order to protect the board identifier the 24LC024 chip has a write protect which is controlled by the jumper pins J9. In order to write to the EEPROM the pins J9 must be shorted.

The format of the identifier is specified in [??]. The TFC system identifier is “2” and the Odin board type has been defined to be “0”. There are currently three different revisions of Odin in use, final prototype (P2), the pre-production version (V1) and the production version (V2). Although there are a number of small differences between the versions they are functionally completely identical to a user. Table 14 lists the different boards and their exact identifiers.

Table 14: The Odin board identifiers.

Board name	Revision	Number	SystemID(TFC)	BoardType	Revision	Serial number	Hex
			<31 .. 28>	<27 .. 20>	<19 .. 16>	<15 .. 0>	
OdinP2_00	P2	00	0010	00000000	0000	0000000000000000	0x20000000
OdinP2_01	P2	01	0010	00000000	0000	0000000000000001	0x20000001
OdinV1_00	V1	00	0010	00000000	0001	0000000000000000	0x20010000
OdinV1_01	V1	01	0010	00000000	0001	0000000000000001	0x20010001
OdinV2_00	V2	00	0010	00000000	0010	0000000000000000	0x20020000
OdinV2_01	V2	01	0010	00000000	0010	0000000000000001	0x20020001
OdinV2_02	V2	02	0010	00000000	0010	0000000000000010	0x20020002
OdinV2_nn	V2	nn	0010	00000000	0010	nn	0x2002xxxx

A command “tstat”on the Credit Card PC command line allows getting all information about the Odin board identifier and all the FPGA code versions.

11 L0 Trigger Flow

11.1 Overview

Odin normally receives the L0 trigger decision from the central L0 trigger Decision Unit (L0DU) or alternatively from an optional local trigger unit. In order to adjust the global latency of the entire L0 trigger path to a total of 160 cycles (4 μs), it has a pipeline, partly of programmable length, at the input of the L0 trigger. Provided no other changes are made to the system, the depth of the pipeline is adjusted once and for all during the commissioning with the first timing alignment. The Bunch Crossing ID received via the L0 trigger input is compared to the expected value from an internal counter in order to verify that the L0DU is synchronized.

In addition to the standard L0 trigger input there is an auxiliary trigger input. The auxiliary trigger may be configured to generate a set of consecutive L0 triggers upon a pulse on the auxiliary input.

Odin provides several means for auto-triggering. It incorporates two independent uniform pseudo-random generators to produce L0 and L1 triggers according to a Poisson distribution. It has also several state machines synchronized to the orbit signal for periodic triggering of a single or a specified number of consecutive bunch crossings, triggering at a programmable time after sending a command to fire a calibration pulse, a trigger memory for a sequence of up to 3564 bunch crossings corresponding to a full LHC orbit. Al the auto-triggers may be fired on demand via the ECS.

In addition, there is a function by which Odin generates a set of L0 trigger accepts before and after a special trigger, e.g. an isolated interaction, which is signalled by the L0DU for time alignment purposes.

A trigger type encoded on three bits is assigned to each source of L0 trigger (**Error! Reference source not found.**). Since several types of triggers may fire for the same bunch crossing a priority handler resolves the ambiguity according to the priorities listed in **Error! Reference source not found.**. The accepted L0 triggers are counted by the L0 Event ID counter which is used to identify events at level-1.

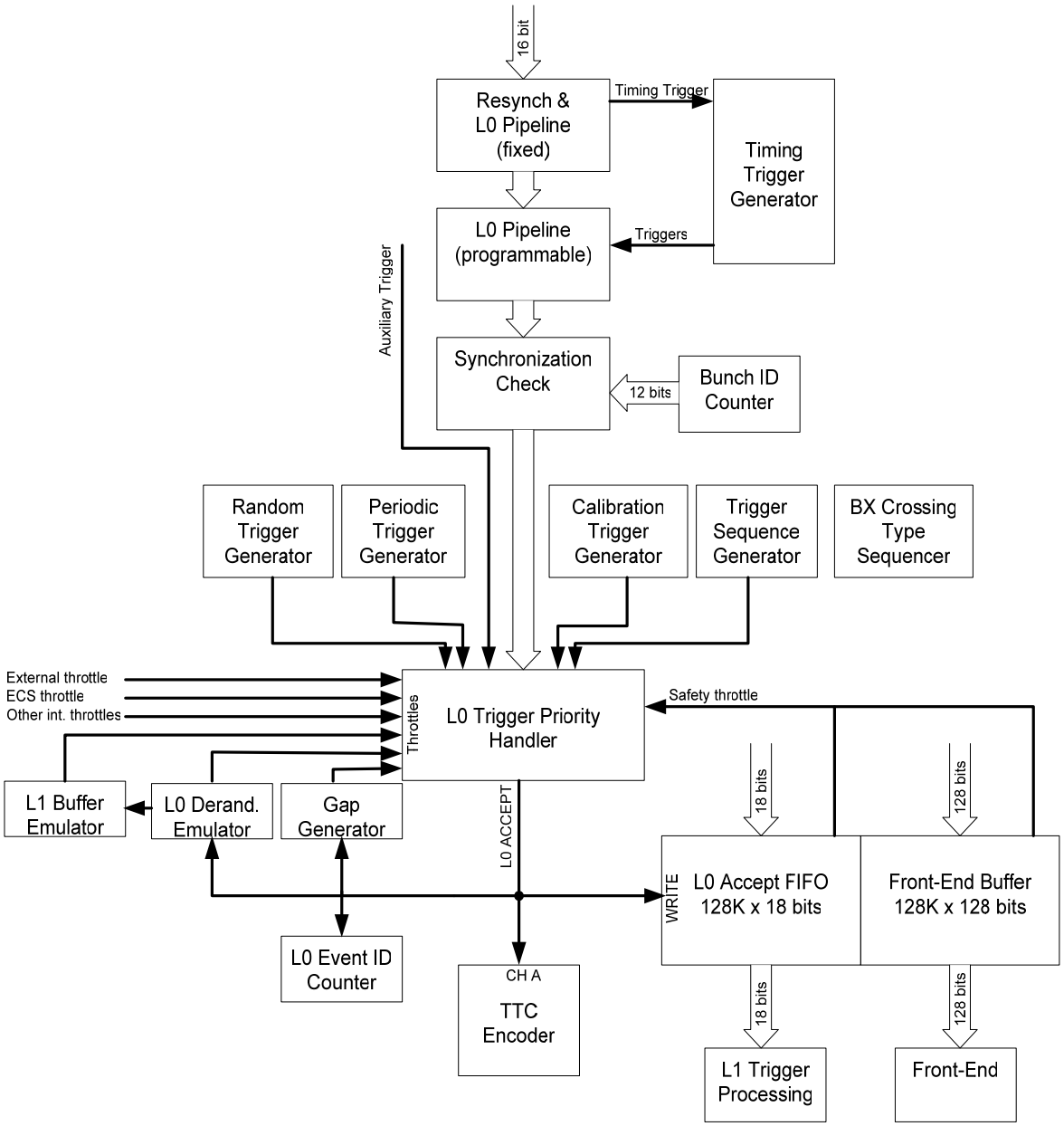


Figure 8: Block diagram of the L0 Trigger Flow.

The L0 trigger decision is subsequently transmitted via the TTC channel A to the front-end electronics. The total minimum latency between the trigger sampling at the L0 trigger input and the TTC transmission adds up to five clock cycles with a zero length L0 pipeline.

The L0 trigger decision signal transmitted to the TTC encoder is also used to write local L0 event data to the Front-End Buffer (Section 20.2), and to write a set of information to the L0 Accept FIFO which is necessary to handle the corresponding L1 trigger. The information written to the L0 Accept FIFO consist of the 3-bit trigger type together with a 2-bit Bunch Crossing ID, a 12-bit L0 Event ID (number of L0 triggers accepted), and a “force bit”. The force bit indicates if the L0 trigger has been “forced” and that consequently the corresponding L1 trigger decision should be made positive, irrespective of the L1 physics trigger decision. The information in the FIFO is read out at the arrival of each L1 trigger.

In order to avoid buffer overflows Odin may throttle L0 triggers accepts of any type at any time. In practice it is done by converting a positive decision into a negative. There are several different sources of L0 throttles:

- L0 derandomizer buffer level which is emulated by Odin.
- L1 buffer level which is emulated by Odin.
- Internal L0 throttle due to a L0 electronics reset or a L1 electronics reset (see Section 14.4 and 14.5).
- External L0 trigger throttle, typically due to data congestion in the TELL1/UKL1 boards.
- L0 trigger throttle via the ECS due to for instance a "pause data taking" or imminent overflow in the CPU farm at level-1.
- Gap generator which may be configured to introduce a gap of a minimum length between L0 trigger accepts.
- Safety throttles from the L0 Accept FIFO and the Front-End Buffer almost full signals.
- Automatic stop on a programmable number of L0 trigger accepts.

All the different types of throttle signals may be individually enabled/disabled.

In order to store the exact bunch crossing type in the event data, Odin has a 3564 steps long bunch crossing sequencer this is programmed via the ECS.

All the functions in the L0 trigger flow are performed by the Q_L0 FPGA except for the L1 buffer emulator (Q_L1) and the L0 calibration trigger generator (Q_MP).

11.2 Physics L0 Trigger

11.2.1 L0 Trigger Input

The L0 trigger link [??] between the L0 Decision Unit and Odin transmits the L0 trigger decisions at a rate of 40 MHz. Each decision consists of a 16-bit data word:

- Bunch crossing number L0_BID (12 bit)
- L0 trigger decision L0_DECIS (1 bit)
- L0 force bit L0_FORCE (1 bit)
- L0 timing trigger L0_TIM_TRG (1 bit)
- Reserve (1 bit)

15	14	13	12	11 .. 0
Reserve	L0_TIM_TRG	L0_FORCE	L0_DECIS	L0_BID

The physical link consists of a point-to-point parallel 16-bit LVDS link using a twisted pair ribbon cable with 17-pairs. The LVDS pairs are terminated in the standard manner with a 100 ohm resistor at the receiver. The link has been implemented using the following components:

- Receiver chips: Maxim 9179 EUE (Quad LVDS receiver with hysteresis)
- PCB connector: 3M Pak 100 4-wall header with latch, 34 pin (3M 3431-5502 or 5602)

- Cable connector: 3M Pak 100 Wiremount socket, 34 pin (3M 3414-6600 or 6634)
- Ribbon cable: Twisted pair flat 34C 1.27mm AWG28 (SCEM 04.21.22.434.8)

Figure 9 shows the pin configuration of the connector and figure 2 shows the actual layout of the connector with the position of pin 1. Notice the order of the LVDS+/- and that the 17th pair (pins 33 and 34) should be ground.

The L0 Trigger Input is enabled with the parameter R_L0_EXT_ENB. Note that since the LVDS receivers are in state HIGH when the link input is open, enabling the link will produce triggers at 40MHz!

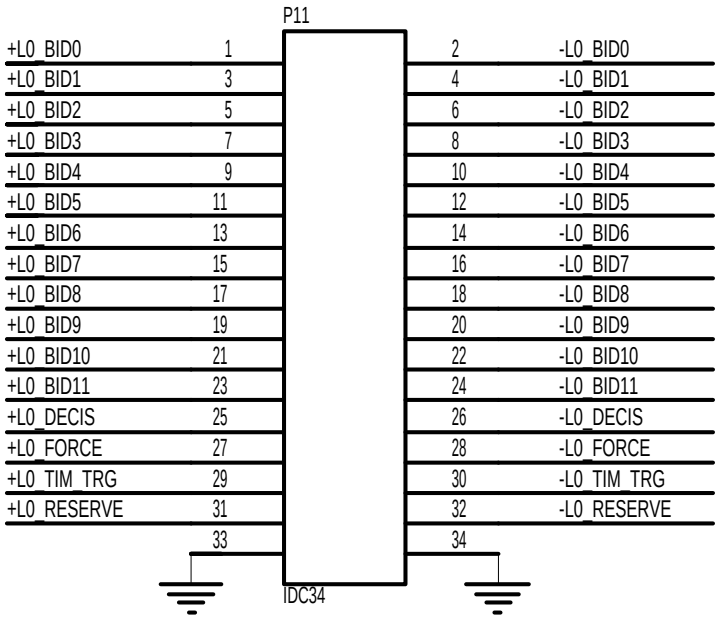


Figure 9: L0 Trigger Input pin configuration.

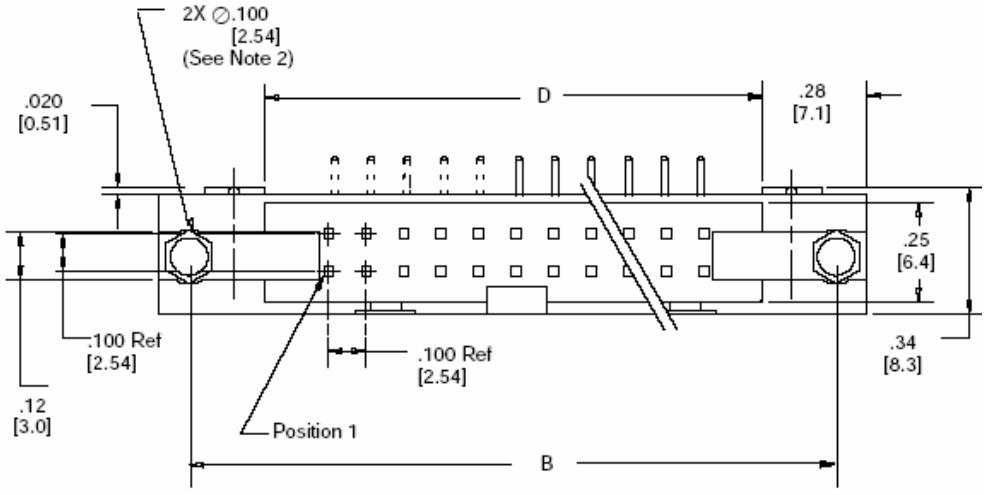


Figure 10: Connector layout with the position of pin 1 for the connector 3M 3431-5502.

11.2.2 L0 Trigger Pipeline

The L0 trigger pipeline is implemented to adjust the global latency and to handle the L0 timing triggers. At the input of the pipeline the L0 trigger may be latched on the rising or the falling edge of the system clock by changing the parameter H_L0_PHASE. In case it is sampled on the falling edge, an additional pipeline step resynchronizes the data to the rising edge.

The pipeline is 16 bits wide and has a part with a fixed depth and a section with a programmable depth. Currently the fixed depth is seven events deep in order to be able to force up to seven events before and after a timing trigger. The exact number of consecutive triggers before and after the timing trigger may be configured using the parameter P_TIMTRG_WIN. If a deeper fixed pipeline is needed to absorb additional L0 latency the depth may be increased by modifying the VHDL code.

The programmable depth is up to 16 events deep and is configured with the parameter H_L0_PIPELINE_LEN.

The data resynchronization to the system clock is implemented in ODIN_L0_resynchronize.vhd and the L0 trigger pipeline is implemented in ODIN_L0_pipeline.vhd.

11.2.3 L0 Synchronization Check

In order to assign a Bunch Crossing ID (BID) to each L0 trigger accept and verify that the L0DU unit is synchronized, Odin has an internal BID counter synchronized directly by the orbit signal. The internal BID is available by reading C_BID_INT.

For each L0 trigger decision arriving at the L0 trigger input, the incoming BID is compared to the value counted internally. The register C_BID_EXT holds the incoming value. The comparison is made after the L0 trigger pipeline provided that the external L0 trigger input has been enabled via the parameter R_L0_EXT_ENB or R_TIMTRG_ENB. A desynchronization is reported on the status bits S_SNCERR_INST (a single or several cases occurred) and S_L0_SNCERR_CONT (continuous), and the number of synchronization errors is counted by the C_L0_SNCERR counter.

An L0 and L1 trigger accept may be forced in case of a synchronization error irrespective of the actual trigger decision by configuring Odin to take actions on synchronization errors with the bit R_L0_SNCCHK and setting the bit R_L0_SNCERR_KEEP. In case the R_L0_SNCERR_KEEP is not set the L0 triggers with synchronization errors will be rejected irrespective of the actual L0 decision.

In case acting on synchronization check is disabled (R_L0_SNCCHK = 0), the errors are still counted and reported, but the events are accepted according to the normal trigger decision.

Note that the C_L0_SNCERR counter does not count synchronization errors which are throttled away. In addition, there are two counters for the total number of events which were rejected due to a synchronization error before (C_L0_SNCERR_REJ_TOT) and after the L0 throttle is applied (C_L0_SNCERR_REJ_GT).

L0 Synchronization errors are also signalled by the LED D4-R. The signal is in reality stretched to be visible on the LED. The status of the L0 synchronization check is also logged in the local event data from Odin (See Section 20.2)

The L0 Synchronization check is implemented in ODIN_L0_synch_check.vhd.

11.3 Auxiliary L0 Trigger

Odin has a DC-coupled ECL input via the LEMO connector P5-L on the front-panel for auxiliary L0 triggers. Currently the input is fed through an edge detect function which produces a one clock cycle pulse irrespective of the length of the input pulse. The trigger pulse is used to generate a programmable set of consecutive triggers. The auxiliary trigger is enabled with the parameter R_AUXTRG_ENB and the length of the trigger burst is set with the parameter P_AUXTRG_NTRG. Setting the parameter R_AUXTRG_FRC will force accepts of all auxiliary L0 triggers at level-1. By default they are all rejected unless an external decision unit decides otherwise. The auxiliary L0 triggers may also be randomly accepted at level-1 by enabling the random L1 trigger.

The trigger type for auxiliary triggers is “010”.

11.4 Random L0 Trigger

The random L0 trigger is part of a random generator which also produces the random L1 triggers. The random L1 triggers are generated simultaneously with the L0 triggers by randomly setting the force bit according to the random L1 trigger decision.

The random generator is similar to a cellular automaton built up of basic computational cells containing XOR gates, feedbacks and an internal RAM with partly random addressing by coupling the addressing to the output of the cell [??]. This random generator has been proven to have an extremely long pseudo-random sequence with no correlation. The implemented random generator chains 32 basic cells in a loop in order to generate two 32-bit uniform and independent random numbers every clock cycle. A numerical formula estimates the length of the sequence to $>10^{87}$.

The random generator requires two 32-bits seeds which are programmed via the parameters P_L0RND_SEED and P_L1RND_SEED. The random number generation is enabled with the parameter R_RNDGEN_ENB. Note that this does not enable the random triggers.

By comparing the L0 random number to a 32-bit L0 threshold the rate of L0 trigger accepts may be adjusted with a precision of $1/(2^{32}-1)$ * 40 MHz. The rate of L0 trigger accepts is defined as rate(L0) = 40MHz * (1 – (threshold(L0)/ $(2^{32}-1)$)) and the rate of L1 triggers is defined as rate(L1) = rate(L0) * (1 – (threshold(L1)/ $(2^{32}-1)$)). The thresholds are set by the parameters P_L0RND_THRESH and P_L1RND_THRESH.

Changing the random seeds and the thresholds should be done when the random generator is disabled (R_RNDGEN_ENB = 0) since the values are loaded on enabling.

The random generator may be configured to produce only L0 random triggers (R_L0RND_ENB) for which all or none of the L1 triggers may forced by setting or clearing the parameter R_RND_FRC. Random L1 triggers are enabled with the parameter R_L1RND_ENB. Note that if both are set the parameter P_RND_FRC has priority over R_L1RND_ENB.

The random generator may also be used to take random L1 decision for the auxiliary L0 triggers by only enabling the random L1 trigger. It is foreseen to be able to do this for periodic triggers or other types but it has not been activated.

The trigger type for random triggers is “011”.

The random generator is implemented in a generic way in rnd_generator.vhd and rnd_cell.vhd

11.5 Periodic L0 Trigger

There are two state machines to generate periodic bursts of L0 triggers (A and B) at a specific bunch crossing in the LHC turn. This allows sampling for instance empty or single beam crossings for data link verifications and noise measurements. Each periodic trigger generator may produce a burst at a maximum of once per LHC turn. The periodicity is programmed with the parameter P_PERTRG_A_PER (_B_) and the bunch crossing at which the burst should start with P_PERTRG_A_OFS (_B_). The burst length is set with P_PERTRG_A_NTRG (_B_) and the periodic triggers are enabled with R_PERTRG_A_ENB (_B_). Periodic triggers may also be requested via the ECS by setting the bit DMND_PERTRG_A (_B_) irrespective of whether it is enabled or not

The periodic trigger state machines may be reset individually with RST_PERTRG_A (_B_). During a L0 and L1 electronics reset they are both held in their reset state.

The L1 decisions for the periodic triggers are forced to accept. The trigger type for periodic triggers is “100”.

The periodic triggers are implemented in ODIN_per_trg.vhd

11.6 Calibration Trigger

There are three generators of calibration sequences (A, B and C). The sequences consist of sending a TTC command at a specific bunch crossing to activate a calibration system in the front-end electronics and to generate a L0 trigger accept after a specified time interval. The three calibrations may be programmed to send different calibration commands to allow different calibration sequences for the sub-detectors. The calibration TTC command is encoded in the user data according to “0001CCXX where the two bits CC distinguishes between different calibrations. For a summary of the TTC commands see Section 14.

For the default calibration sequence the time interval between the calibration command and the L0 trigger is 160 + 16 cycles and the calibration TTC command is “000100XX” [??].

The periodicity is programmed with the parameter P_CALTRG_A_PER (_B_, _C_) and the bunch crossing at which the calibration TTC command is transmitted with P_CALTRG_A_OFS (_B_, _C_). The delay between the transmission of the calibration command and the actual L0 trigger accept is configured with the parameter P_CALTRG_A_DEL (_B_, _C_). The calibration triggers are enabled with the parameter R_CALTRG_A_ENB (_B_, _C_). Calibration triggers may also be requested via the ECS by setting the bit DMND_CALTRG_A (_B_, _C_) irrespective of whether it is enabled or not

In case a calibration generator is not able to send the calibration command at the specified bunch crossing because it clashes with another TTC command of higher priority, a repeat bit is set to request another attempt at the specified bunch crossing in the next LHC turn. The status of the repeat bit can be read from the register S_RPT_CALTRG_A (_B_, _C_). Note also that even if a

calibration command has been sent to the front-end electronics a corresponding trigger accept is not guaranteed as it may be throttled away.

The calibration trigger state machines may be reset individually with RST_CALTRG_A (_B_, _C_). During a L0 and L1 electronics reset they are both held in their reset state.

The L1 decisions for the calibration L0 triggers are forced to accept. The trigger type for auxiliary triggers is “111”.

To allow sending the calibration event data to a particular destination in the CPU farm, a special HLT IP destination broadcasts, a single event flush, could follow the calibration L1 trigger[5].

The calibration generators are implemented in ODIN_cal_trg.vhd. They are running in the Q_MP module.

11.7 Timing Trigger

A special timing calibration scheme has been implemented for the purpose of time aligning the front-end electronics such that the events are tagged with the same Bunch Crossing ID. The scheme consists of transmitting a set of consecutive L0 trigger accepts for before and after a special trigger such as an isolated interaction, which is identified by the L0DU.

The L0DU sets the timing trigger bit at the same time with the actual trigger decision corresponding to the special event transmitted via the L0_DECIS line. The timing trigger signal is not fed through the fixed part of the L0 pipeline. Upon receiving the timing trigger signal ODIN generates a series of trigger accepts before and after the special event up to a maximum of 15 = 7 before + actual timing trigger + 7 after.

The timing trigger window, that is the number of triggers before and after the special trigger, is set by the parameter P_TIMTRG_WIN and the scheme is enabled with R_TIMTRG_ENB. The L1 decisions for the entire set of L0 triggers are forced to accept.

The trigger type for timing triggers is “110”.

11.8 Trigger Sequencer

In order to be able to produce very complex L0 and L1 trigger patterns, Odin has a 2-bit wide memory with 3564 (a full LHC turn) entries which can be filled via the ECS. The two bits correspond to the L0 and the L1 decision, respectively, for each bunch crossing. Writing to the memory is done by means of the parameters P_SEQTRG_ADDR, which is the number of the bunch cross at which writing starts, and P_SEQTRG_DATA is the 2-bit word. A write or a read in the memory increments automatically the address pointer. The memory may only be written to while the sequencer is disabled. The sequencer is enabled with the parameter R_SEQTRG_ENB.

The trigger type for these triggers is the same as the random triggers “011”.

[5] This has not been implemented for the current readout scheme

This module is not in its final state and is currently not part of the default FPGA code.

11.9 L0 Trigger Rate Control

Odin controls the trigger rates according to the status of the buffers in the system in order to prevent overflows. The L0 trigger rate is controlled by a global L0 throttle signal which negates the L0 trigger decisions. The global L0 throttle signal is a combination of L0 throttle signals from several sources described below. Each throttle source may be individually enabled/disabled and the status of each is available in the status register. The status of the global throttle can be read from S_L0_INH.

11.9.1 L0 Derandomizer Emulator

Due to the distance and the high trigger rate, the occupancy of the L0 derandomizer in the L0 front-end cannot be monitored in a direct way. However, as the buffer activity is completely deterministic, Odin has a state machine to emulate the occupancy.

The state machine is centred on an occupancy counter which is incremented with the L0 trigger accepts and decremented for each time an event is read out of the derandomizer, which in turn is emulated with a readout time counter. The readout time counters is configured with the parameter P_L0D_RO_TIME. Programmable upper and lower watermarks determine when the emulator starts throttling L0 trigger accepts and when it releases the triggers (P_L0D_LOCCUPANCY and P_L0D_UOCCUPANCY).

Although the emulator is constantly active, its throttle signal is enabled with the parameter R_L0EMU_L0THR_ENB. The current emulated buffer occupancy is available by reading C_L0D_OCCUPANCY and the status of the throttle signal is available in S_L0EMU_L0THR.

The L0 derandomizer emulator may be individually reset with RST_L0EMU. It is also reset by a L0 and a L1 electronics reset.

The emulator is implemented in ODIN_l0d_emulator.vhd.

11.9.2 External L0 Throttle

The L1 output buffers of the TELL1/UKL1 boards are monitored locally. In case of data congestion they feed back L0 throttle signals via the dedicated throttle network to Odin.

Odin has two L0 throttle inputs on the front-panel. The inputs are based on LVDS via the pair “0” in two RJ9 connectors (P13 and P14). The termination scheme and the receiver type is the same as the L0 trigger input. Figure 11 shows the pin configuration of the RJ9 connectors. A special custom-made dual twisted pair cable with the right dimensions for the RJ9 plugs has been specified and manufactured.

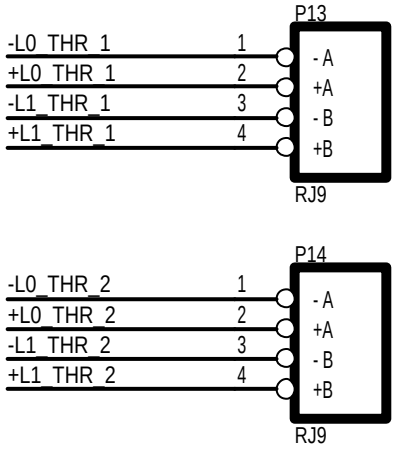


Figure 11: Pin configuration of the RJ9 external throttle connectors.

The external L0 throttle signals may be individually enabled/disabled with the parameter R_EXT1_L0THR_ENB and R_EXT2_L0THR_ENB. The status of the two throttles can be read from S_EXT1_L0THR and S_EXT2_L0THR. The counter C_L0THR_EXT counts the total time (number of clock cycles) during which any of the two external throttle signals are active.

The LED D7-L indicates the state of the L0 throttle lines irrespective of whether the throttle is enabled or disabled. The signal is in reality stretched to be visible on the LED. Note that the LVDS receiver is in state HIGH when the external throttle is not driven.

11.9.3 ECS L0 Throttle

There are several reasons why it may be necessary to throttle triggers via the ECS system. Data congestion at the level of the CPU farm is detected and signalled via the ECS. The run may have to be paused for a short period of time due to for instance background conditions, problems in the slow control system etc. The ECS throttle is activated by setting the parameter R_L0_THR.

11.9.4 Forced L0 Gap

Although it is not part of the requirements on the front-end electronics, Odin has a function to force gaps in between L0 trigger accepts. The gap generator introduces a gap of programmable length (P_L0_GAP_LEN) by throttling away all triggers which occur during the gap. The gap generator is enabled if P_L0_GAP_LEN is different from zero.

The gap generator is implemented in the ODIN_gap_generator.vhd

11.9.5 Automatic Trigger Stop

The automatic trigger stop allows stopping the triggers after a programmable number of L0 trigger accepts. The limit is programmed by the P_L0_MAX parameter and the function is enabled with the parameter R_L0_MAX_ENB. Configuring and enabling the automatic stop must be done while the triggers are stopped (R_STOP = 1). The automatic start is activated automatically when the run is started by clearing R_STOP.

The status bit S_L0_AUTO_STOP indicates that the automatic stop has been activated. Note that

the parameter R_STOP is not affected. To stop the run completely it must be set.

11.9.6 Other Internal L0 Throttle Sources

There are other functions that may throttle the L0 triggers which are discussed later in this document:

- L1 Buffer Emulator (Section 13.9)
- L0 Electronics Reset (Section 14.4)
- L1 Electronics Reset (Section 14.5)
- L0 Accept FIFO safety throttle (Section 12)
- Front-End Buffer safety throttle (Section 20.2)

11.10 L0 Trigger Priority Handler

The different trigger sources are listed together with their type identifier in **Error! Reference source not found.**. The table also specifies their priorities. The priorities are used in the L0 trigger priority handler to define the trigger type in case several sources produce a trigger for the same bunch crossing. Since the calibration trigger corresponds to an event in which a calibration pulse has been fired or special data was loaded into the readout, this trigger type has the highest priority. A trigger is tagged “physics” only if the L0 trigger accept was received via the L0 trigger input and no other source triggered.

Table 15: L0 trigger types and the priority level (highest value corresponds to highest priority). *Note that the trigger reject type is not used for the L0 trigger but as a qualifier for L1 trigger rejects. **The usage of the trigger type to signal the TELL1/UKL1 to readout non-zero suppressed data has not been fully defined

Trigger type	Encoded	Priority
Reserve	000	-
Physics trigger	001	1
Auxiliary trigger	010	2
Random trigger	011	3
Periodic trigger	100	4
Trigger for non-zero suppressed data**	101	-
Timing trigger	110	5
Calibration trigger	111	6

The trigger priority handler also applies the global throttle signal and produces non-gated and gated counter signals for the different trigger types.

The trigger priority handler is also where the triggers are stopped and restarted by setting and clearing the parameter R_STOP.

A positive decision at the output of the trigger priority handler triggers the following actions:

- the L0 trigger decision is transmitted to the TTC encoder,
- the L0 Event ID counter is incremented together with all the other trigger counters,
- a set of information necessary to process the corresponding L1 trigger decision is written to the L0 Accept FIFO (Section 12)
- and the Odin L0 event data is written to the Front-End Buffer (Section 20.2)

The LED D6-R indicates L0 trigger accepts. The LED D6-L is currently a reserve LED.

The trigger priority handler is implemented in ODIN_L0_handling.vhd.

11.11 L0 Event ID Counter

The L0 Event ID counter is a special counter in that it is used to identify the events in the whole system at level-1 and that it is reset synchronously with the Event Counter Reset. It is also reset by a L0 and a L1 electronics reset. The counter is 32 bits wide but currently only 24 bits are stored in the data. The counter value is available in the register C_L0_EID.

The L0 Event ID counter is implemented in ODIN_eid.vhd.

11.12 Bunch Structure Sequencer

In order to log the type of crossing (BX_TYPE(1..0)) for which a trigger was produced, typically as a cross-check, Odin has a bunch structure sequencer based on a 2-bit wide memory with 3564 (a full LHC turn) entries which can be filled via the ECS with “00” for no crossing, “01” and “10”for single beam crossings, and “11” for beam-beam crossings. The bunch crossing type is written to the Front-End Buffer (Section 20.2) at each L0 trigger accept.

Writing to the memory is done by means of the parameters P_BXSEQ_ADDR, which is the number of the bunch cross at which writing starts, and P_BXSEQ_DATA is the 2-bit word. A write or a read in the memory increments automatically the address pointer. The memory may only be written to while the sequencer is disabled. The sequencer is enabled with the parameter R_BXSEQ_ENB.

This module is not in its final state (ODIN_bx_type.vhd) and is currently not part of the default FPGA code.

12 L0 Accept FIFO

The L0 Accept FIFO links the L0 trigger processing and the L1 trigger processing, and absorbs the L1 trigger latency. The link communicates L0 trigger information necessary for the processing of the corresponding L1 trigger decisions. The information consist of the two lower bits of the Bunch Crossing ID, the trigger type, the force bit, and the twelve lower bits of the L0 Event ID for each L0 trigger accept. The format is shown in Table 16.

Table 16: Format of the L0 Accept FIFO data.

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L0 Event ID (11 .. 0)												FRC	TYPE (2 .. 0)			BID(1 .. 0)	

The L0 Accept FIFO consist of two discrete 128K x 9-bit FIFOs (Cypress CY7C4292V-10) with independent read and write buses. The read and write actions are clock synchronous.

The Empty and the Almost Full FIFO status flags are available via the Local Bus as S_AFIFO_EMPTY, respectively S_AFIFO_FULL_CONT (continuous) and _FULL_INST (instantaneous). The Almost Full signal is also used to generate an internal safety L0 throttle which

may be enabled via the parameter R_FIFO_L0THR_ENB.

The L0 Accept FIFO may be reset asynchronously via the Local Bus by the bit RST_AFIFO.

13 L1 Trigger Flow

13.1 Overview

Figure 12 shows a block diagram of the L1 Trigger Processing. The standard L1 trigger interface on Odin is based on Gigabit Ethernet over which the triggers are transmitted in the form of IP packets. A trigger decoder extracts the L0 Event ID for the trigger and the L1 trigger information, and checks the status word for errors.

Optionally, Odin also has an alternative L1 trigger input based on LVDS for easy interfacing with a local L1 trigger decision unit.

At the arrival of a L1 trigger, the corresponding L0 trigger information is read out from the L0 Accept FIFO. In order to validate the L1 triggers, the L0 Event ID of the L1 trigger decision is compared to the expected L0 Event ID counted at level-0. The trigger type, which is also in the L0 trigger word read from the L0 Accept FIFO, is then compiled together with the two lower bits of the L0 Event ID to form the L1 trigger broadcast word. The L1 trigger broadcast word consists of five bits encapsulated in the user data of a TTC short broadcast as “1TTTTT00”. The five bits carry the 3-bit trigger type and the two lower bits of the L0 Event ID. For a summary of the TTC commands see Section 14.

The broadcast word is subsequently written to the L1 Trigger Queue which is implemented in order to absorb pile-up of trigger decisions during the TTC transmission. In case Gigabit Ethernet is used to receive triggers according to the format specified below (Table 17), the L1 trigger information is simultaneously queued in the L1 Trigger Info Queue until the decision has been sent via the TTC system to the front-end electronics.

The transmission of the L1 trigger decisions via the TTC system is performed according to a specified timing which includes a defined interval between L1 trigger broadcasts after a L1 trigger accept, and an intrinsic interval between L1 trigger rejects determined by the serialization of the L1 trigger broadcast.

In order to avoid buffer overflows Odin may throttle L1 trigger accepts of any type at any time. In practice it is done by converting a positive decision into a negative. There are several different sources of L1 throttles:

- Internal L1 trigger throttle due to a L1 electronics reset (see Section 14.5).
- External L1 trigger throttle, typically due to data congestion in the TELL1/UKL1 boards.
- L1 trigger throttle via the ECS due to for instance an imminent overflow in the CPU farm.
- L1 throttle from the L1 Trigger Queue almost full signals.
- Safety throttle from the Gigabit Ethernet MAC Ingress FIFO status.
- (Automatic stop on a programmable number of L1 trigger accepts).

Odin also incorporates a state machine to generate the timing for internal L1 triggers. The actual

internal L1 trigger decisions are all generated at the same time as the internal L0 triggers by setting the force bit. The source of the L1 trigger accepts is identified by the accompanying trigger type.

All the functions in the L1 trigger flow are performed by the Q_L1 FPGA except for the L1 Event ID counter which is located in the Q_FE FPGA.

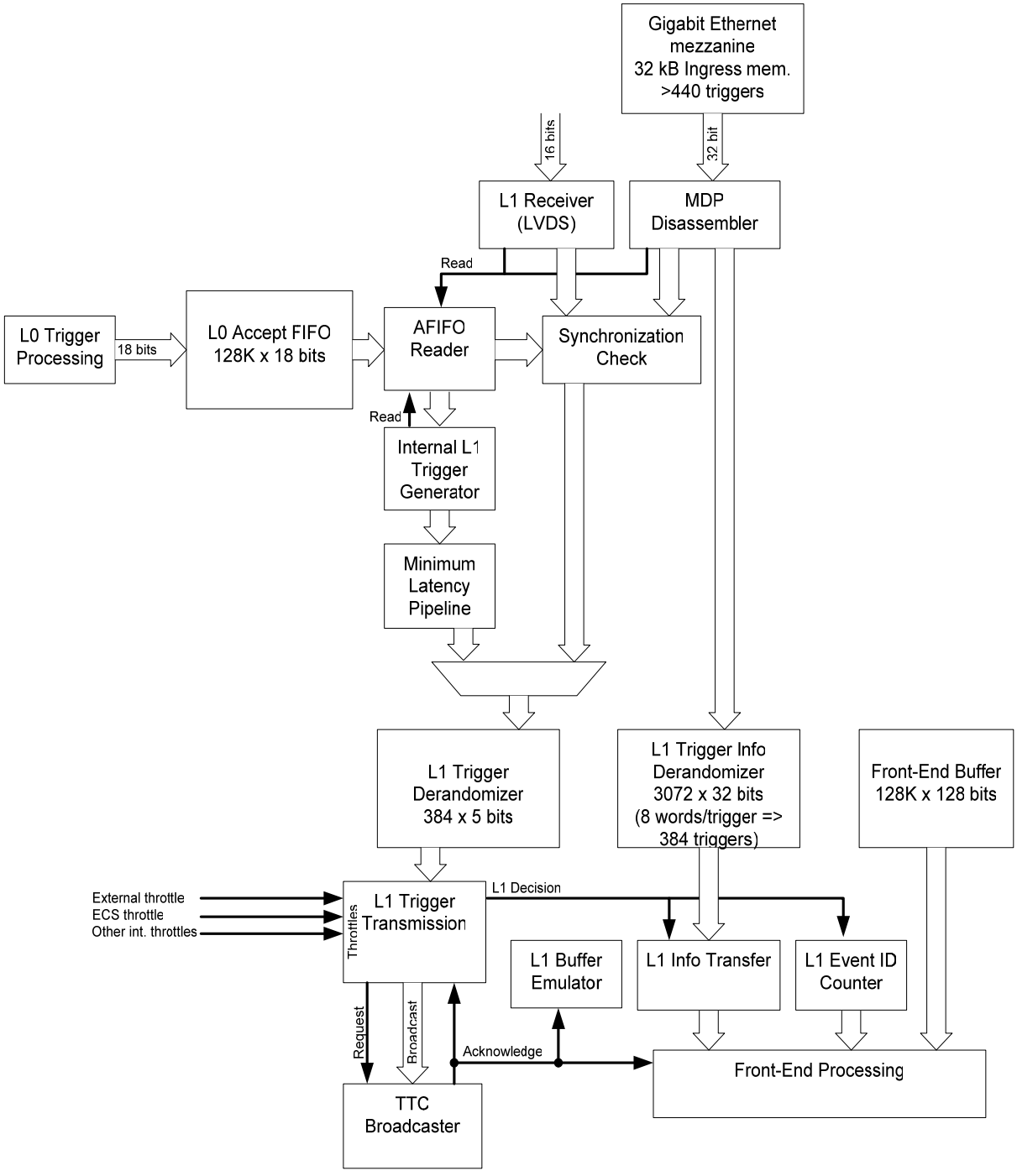


Figure 12: Block diagram of the L1 trigger processing.

13.2 Physics L1 Trigger

The L1 trigger decisions are normally received as IP packets formatted according to the L1 Multi-Decision Packet (MDP) format [??] shown in Table 17.

Table 17 : Multi-Decision Packet format. The eight bytes of preamble are not shown.

Decision	32-bit word	31 .. 24	23 .. 16	15 .. 8	7 .. 0
	0				
	…		Ethernet + IP header (see [4])		
	8				
	9		MDP Error block		# of events
0	10		L0 Event ID		
	11		Error and status block		L1 decision
	12				
	…		L1 trigger information		
	17				
1	18		L0 Event ID		
	19		Error and status block		L1 decision
	20				
	…		L1 trigger information		
	25				
N	N * 8 + 10		L0 Event ID		
	N * 8 + 11		Error and status block		L1 decision
	N * 8 + 12				
	…		L1 trigger information		
	N * 8 + 17				
	N * 8 + 18		CRC		

Missing or late decisions are detected by a timeout in the Sub Farm Controller (SFC) and is flagged in the MDP error block.

The parameter R_L1_GBE_ENB determines whether the triggers are received via Gigabit Ethernet (HIGH) or LVDS (LOW). Note that in order to receive external L1 triggers also the parameter R_L1_EXT_ENB must be set.

For the 1MHz readout scheme, see Section 24.

13.2.1 L1 Trigger via Gbit Ethernet

The L1 trigger reception via Gigabit Ethernet is handled by the MDP disassembler. The Gigabit channel by which the L1 triggers are received is configured with the parameter P_L1_GBE_CHANNEL and the IP address of the L1 trigger reception on Odin can be configured with P_L1IP_ADDRESS. If P_L1IP_ADDRESS is set to zero, all packets are accepted. The size of the MPDs (packing factor) is configured with the parameter P_L1MDP_LEN. The size is checked but a mismatch is only reported since the size may vary due to flushing of the MEP buffers (Section 15).

The data interface with the Ingress of the Media Access Controller (Intel IXF1104) on the Gigabit Ethernet mezzanine is based on an SPI-3 interface operating in 32-bit mode. On Odin the Ingress SPI-3 bus is clocked at 80MHz. The bus clock is synchronous with the system clock and is

generated using an internal PLL inside the Q_L1 FPGA. The lock status of the L1 PLL in Q_L1 and the FE PLL in Q_FE is indicated by the LED D5-R on the front-panel.

The MAC has a 32 KB internal Ingress FIFO which allows some derandomization during reception. At a packing factor of one for the Multi-Decision Packets, the FIFO can fit ~440 L1 triggers. At a packing factor of ten, 880 L1 triggers fit into the buffer. However, since there are no means by which the arrival of the L1 triggers may be slowed down at this level, it is not possible to rely on it fully for derandomization. The derandomization is instead handled by the L1 Trigger Sorter module [??].

The MDP disassembler strips off the IP+Ethernet header, checks the MDP error block and the number of triggers in the MDP, and extracts the information of one trigger at a time. The disassembler then triggers a read of the L0 Accept FIFO and feeds the L0 Event ID to the synchronization check. If the L1 trigger decision is positive or the force bit is set in the L0 Accept FIFO, the L1 trigger information is written to the L1 Trigger Info Queue. Provided the L1 trigger is not throttle away before the L1 broadcast is sent to the front-end electronics, the information is transferred to the Odin Front-End where it is assembled together with the L0 event data.

The MDP disassembler is implemented in ODIN_L1_trigger_GBE.vhd (not final).

13.2.2 Optional: L1 Trigger via LVDS

The optional L1 trigger input (P16) based on LVDS is on the rear side of the board and conveys a 15-bit word for each L1 trigger:

- L0 Event identifier L0_EID (12 bit)
- Bunch crossing identifier BID (2 bit)
- L1 trigger decision L1_DECIS (1 bit)

The 16th bit is a strobe signal which validates a trigger on the L1 trigger bus.

15	14	13	12	11 .. 0
L1_STROBE	L1_DECIS	BID (1..0)		L0_EID (11..0)

The physical implementation is identical to the L0 trigger link (Section 11.2.1). The L1 trigger data may be latched on the rising or the falling edge of the system clock by changing the parameter H_L1_PHASE.

The input may be disabled via bit 5 of the I2C port on peripheral address 0x40 as shown in Table 9. Note that since the LVDS receivers are in state HIGH when the link input is open, enabling the input and L1 triggers via LVDS will produce triggers at 40MHz!

The L1 trigger reception via LVDS is handled in ODIN_L1_trigger_LVDS.vhd.

13.2.3 L0 Accept FIFO Reader

The reading of the L0 Accept FIFO is triggered by the reception of an external L1 trigger or by the state machine which generates the timing for internal L1 triggers.

The read controller is implemented in ODIN_AFIFO_read.vhd

13.2.4 L1 Synchronization Check

For each L1 trigger, the lower 12-bits of the L0 Event ID are compared to the value read from the L0 Accept FIFO. For the LVDS L1 trigger input, the lower two bits of the Bunch ID are also received and compared to the value read from the L0 Accept FIFO. The comparison is made provided that external L1 triggers have been enabled via the parameter R_L1_EXT_ENB. A desynchronization is reported on the status bits S_L1_SNCERR_INST (a single or several cases occurred) and S_L1_SNCERR_CONT (continuous), and the number of synchronization errors is counted by the C_L1_SNCERR counter.

A L1 trigger accept may be forced in case of a synchronization error irrespective of the actual trigger decision by configuring Odin to take actions on synchronization errors with the bit R_L1_SNCCHK and setting the bit R_L1_SNCERR_KEEP. In case the R_L1_SNCERR_KEEP is not set the L1 triggers with synchronization errors will be converted to rejects irrespective of the actual L1 trigger decision.

In case acting on synchronization check is disabled (R_L1_SNCCHK = 0), the errors are still counted and reported, but the events are accepted according to the normal trigger decision.

Note that the C_L1_SNCERR counter does not count synchronization errors which are throttled away. In addition, there are two counters for the total number of events which were rejected due to a synchronization error before (C_L1_SNCERR_REJ_TOT) and after the L1 throttle is applied (C_L1_SNCERR_REJ_GT).

L1 synchronization errors are also signalled by the LED D5-R. The signal is in reality stretched to be visible on the LED.

The synchronization check stores the trigger type

The L1 trigger synchronization check is implemented in ODIN_L1_synch_chk.vhd

13.3 Internal L1 Trigger[6]

The task of the internal L1 trigger state machine is only to emulate the latency of the L1 triggers and emulate the arrival of triggers. The actual L1 trigger decisions are generated at level-0 and transmitted to the internal L1 trigger generator via the force bit.

The state machine generates currently a fixed latency any time the L0 Accept FIFO is empty and a L0 trigger accept is produced, and a minimum latency. Together it corresponds to the L1 trigger latency when the entire L1 trigger path is empty including a fixed processing time for the first trigger. The first trigger latency is produced by a counter and is set with the parameter P_L1_INT_LATENCY. The first trigger is read from the L0 Accept FIFO when the latency timer expires. The subsequent triggers are read at an interval specified by the parameter P_L1_INT_INTERVAL. The interval must be greater than 16 and less than the average interval

[6] This module is being rewritten to improve the latency emulation

between L0 triggers.

The minimum latency is produced with an internal pipeline.

The internal L1 trigger generator is enabled with the bit R_L1_INT_ENB and may be reset individually with the bit RST_L1_INT.

The internal L1 trigger is implemented in ODIN_L1_internal.vhd

13.3.1 Minimum Latency Pipeline

In order to ensure that the event data has been read out by the TELL1/UKL1 boards before the corresponding L1 trigger decision arrives, Odin has a L1 pipeline to introduce a minimum latency during internal triggering.

The depth of the pipeline can be configured with the parameter P_L1_MIN_LATENCY. The current maximum depth of the pipeline allows a minimum latency of 51 μs[7].

In internal mode the triggers are directly written to the L1 Trigger Queue at the output of the latency pipeline.

13.4 L1 Trigger Queue

The L1 trigger queue is a FIFO which is absorbing pile-ups of triggers when the transmission of a L1 trigger over the TTC system is delayed due to the TTC Broadcaster being busy with another transmission or a clash with a command of higher priority.

The L1 trigger queue can currently store 384 events.

The Almost Full status flag is used to throttle L1 trigger decision and is available via the Local Bus as S_TFIFO_FULL_INST (a full state occurred) or S_TFIFO_FULL_CONT (current state).

The L1 Trigger is an internal FIFO in the Q_L1 FPGA (lpm_l1_tfifo.vhd).

13.5 L1 Trigger Info Queue

The L1 trigger info queue is a FIFO which stores the L1 Trigger Information until the corresponding L1 trigger broadcast has been transmitted via the TTC. If the L1 trigger is positive, the data is transferred to the Odin front-end processing, otherwise it is immediately discarded.

The transfer of the data between the Q_L1 FPGA and the Q_FE FPGA is made with a 16-bit bus. The transfer is initiated by a non-zero trigger type and the acknowledge signal from the TTC Broadcaster, and takes 16 cycles.

[7] This will be increased

The L1 trigger info queue can currently store 384 events.

The L1 Trigger is an internal FIFO in the Q_L1 FPGA (lpm_l1_dfifo.vhd)

13.6 L1 Trigger Rate Control

Odin controls the trigger rates according to the status of the buffers in the system in order to prevent overflows. The L1 trigger rate is controlled by a global L1 throttle signal which negates the L1 trigger decisions by clearing completely the trigger type in the L1 trigger broadcast (See **Error! Reference source not found.**). The global L1 throttle signal is a combination of L1 throttle signals from several sources described below. Each throttle source may be individually enabled/disabled and the status of each is available in the status register. The status of the global L1 throttle signal can be read from S_L1_INH.

Since a L1 trigger broadcast may be pending for a while due a concurrent transmission of higher priority, the throttle signal is applied in the clock cycle in which the data is latched by the TTC Broadcaster.

13.6.1 External L1 Throttle

The HLT output buffers of the TELL1/UKL1 boards are monitored locally. In case of data congestion they feed back L1 throttle signals via the dedicated throttle network to Odin.

Odin has two L1 throttle inputs on the front-panel. The inputs are based on LVDS via the pair “1” in the two RJ9 connectors P13 and P14 (See Section 11.9.2).

The external L1 throttle signals may be individually enabled/disabled with the parameter R_EXT1_L1THR_ENB and R_EXT2_L1THR_ENB. The status of the two throttles can be read from S_EXT1_L1THR and S_EXT2_L1THR. The counter C_L1THR_EXT counts the total time (number of clock cycles) during which any of the two external throttle signals are active.

The LED D7-R indicates the state of the L1 throttle lines irrespective of whether the throttle is enabled or disabled. The signal is in reality stretched to be visible on the LED. Note that the LVDS receiver is in state HIGH when the external throttle is not driven.

13.6.2 ECS L1 Throttle

The L1 triggers may also be throttled by the ECS due to for instance data congestion at the level of the CPU farm. The ECS throttle is activated by setting the parameter R_L1_THR.

13.6.3 Other Internal L1 Throttle Sources

There are other functions that may throttle the L1 triggers which are discussed elsewhere in this document:

- L1 throttle from the L1 Trigger Queue almost full signals (Section 13.4).
- Internal L1 trigger throttle due to a L1 electronics reset (see Section 14.5).
- Safety throttle from the Gigabit Ethernet MAC Ingress FIFO status.
- (Automatic stop on a programmable number of L1 trigger accepts).

13.7 L1 Trigger Transmitter

The L1 Trigger Transmitter has the task of controlling the timing of the L1 trigger broadcasts and sending the transmission requests and the L1 trigger word to the TTC Broadcaster. It also ensures that the global L1 throttle is applied in the same clock cycle as the broadcast word is latched by the TTC Broadcaster. The broadcast acknowledges the reception of the broadcast word when it is free to start the transmission.

The readout scheme of the TELL/UKL1 buffer is such that the time required to discard and read out an event is different. This implies that the interval to the next L1 trigger broadcast is different after a L1 trigger accept broadcast and a L1 trigger reject broadcast. The interval after an accepted trigger is configured with the parameter P_L1_INTERVAL_ACCEPT and after a rejected trigger by P_L1_INTERVAL_REJECT. Currently the agreed values are 20μs after an accept and 400ns after a reject. The former is required by the TELL1/UKL1 and the latter is defined by the minimum time to serialize a L1 trigger broadcast.

The bit R_L1_STOP stops L1 trigger transmission entirely. In this state the entire L1 trigger flow is disabled and may be used in order to only receive L0 triggers in the front-end electronics. By default the R_L1_STOP is set.

The LED D5-L on the front-panel indicates the L1 trigger accepts.

13.8 L1 Event ID Counter

The L1 Event ID counter is 32 bits wide and counts the number of L1 trigger accepts. It is used to identify the events in the whole system at the level of the High Level trigger in the CPU farm.

The counter may be reset synchronously in the entire system by transmitting a reset command over the TTC system (“01XXXXXX”). Bit 4 corresponds to the L1 Event ID reset. It may thus be reset simultaneously with any of the other resets. For a summary of the TTC commands see Section 14. The counter is also reset on a L1 front-end electronics reset

Odin transmits the L1 Event ID as a part of the local event data bank. The counter value is also available in the register C_L1_EID.

The L1 Event ID counter is implemented in ODIN_l1eid.vhd.

13.9 L1 Buffer Emulator

Due to the distance and the high input rate, the occupancy of the L1 Buffer in the TELL1/UKL1 boards cannot be monitored in a direct way. However, as the buffer activity is completely deterministic, Odin has a state machine to emulate the occupancy of the L1 buffer.

The state machine is centred on an occupancy counter which is incremented every time an event is read out of the L0 derandomizer, which is also emulated (Section 11.9.1), and decremented with each transmitted L1 trigger. Programmable upper and lower watermarks determine when the emulator starts throttling L0 trigger accepts and when it releases the triggers (P_L1B_LOCCUPANCY and P_L1B_UOCCUPANCY).

Although the emulator is constantly active, its throttle signal is enabled with the parameter

R_L1EMU_L0THR_ENB. The current emulated buffer occupancy is available by reading C_L1B_OCCUPANCY and the status of the throttle signal is available in S_L1EMU_L0THR.

The L1 Buffer emulator may be individually reset with RST_L1EMU. It is also reset by a L1 electronics reset.

The emulator is implemented in ODIN_l1b_emulator.vhd

14 Synchronous Control Command Generation

14.1 Overview

Odin also has also the task of transmitting a set of synchronous control commands via the TTC system [??]:

- Bunch Counter Reset (BCR)
- L0 Event Counter Reset (ECR)
- L0 front-end electronics reset
- L1 front-end electronics reset
- L1 event ID reset
- Calibration command
- Periodic command
- IP destination broadcast

For this purpose it has a set of command generators which are synchronized by the orbit signal and which control the timing of the commands. All the command generators may be enabled/disabled and reset individually. They can be configured to send the commands at regular intervals or solely on demand via the ECS interface. The transmission of the control commands are counted by individual counters.

All control commands are transmitted as short TTC broadcasts except for the IP destination broadcast which uses the long TTC broadcast format. Table 18 shows the encoding of the commands in the user data of the TTC short broadcasts. Control commands are identified by having bit 7 cleared. Reset commands are identified by having bit 6 set. Other commands, which include the calibration commands, have also bit 6 cleared.

Table 18: Summary of the commands using the short TTC broadcast format.

	7	6	5	4	3	2	1	0
Trigger Type	1	Trigger type			L0 EvID		(ECR)	(ECR)
Reset	0	1	R	R	L1 FE	L0 FE	ECR	BCR
Calibration	0	0	0	1	Pulse type		(ECR)	(BCR)
Command	0	0	1	Command type			(ECR)	(BCR)

All the functions listed above are performed by the Q_MP FPGA.

14.2 Bunch Counter Reset

The Bunch Counter Reset (BCR) is transmitted on each orbit pulse as soon as the external orbit signal is available (external clock/orbit selected) or the length of the orbit has been configured with the parameter P_ORBIT_LEN. The BCR has the absolute highest priority among the TTC broadcasts and the TTC Broadcaster (Section 16) ensures that the time slot in which the BCR should be sent is always free. The BCR is solely identified by bit 0 in the TTC short broadcast. Currently no other command, except the L0 event counter reset, is sent together with the BCR.

The offset of the TTC transmission of the BCR with respect to the actual orbit pulse may be configured with the P_BCR_OFS. There is no enable/disable for the BCR transmission.

In order to adjust the timing of the internal Bunch Counter Reset in Odin, a delay can be set with the parameter P_BCR_DEL.

The generation of the Bunch Counter Resets is implemented in ODIN_bcr_ecr.vhd

14.3 Event Counter Reset

The Event Counter Reset allows resetting the L0 Event ID synchronously in the entire system. The ECRs may be sent periodically[8] or on demand via the ECS. The periodic transmission is enabled with the bit R_ECR_ENB and the periodicity is configured with the parameter P_ECR_PER. The ECR is requested via the ECS by setting the bit DMND_ECR irrespective of whether it is enabled or not.

The ECR is solely identified by bit 1 in the TTC short broadcast. Currently, it is always sent together with a BCR.

The timing of the internal L0 Event ID reset in Odin is the same as the internal BCR.

The periodicity counter of the ECR may be reset individually via the Local Bus with RST_ECR.

The generation of the Event Counter Resets is implemented in the same module as the BCRs (ODIN_bcr_ecr.vhd).

14.4 L0 Electronics Reset

The L0 Electronics Reset activates a synchronous reset in the L0 front-end electronics. This reset generator is implemented in a way that it can be done “on the fly” during data taking. The sequences consist of:

1. Throttle the L0 triggers until the L0 derandomizer is empty
2. Send the L0 electronics reset command
3. Release the L0 triggers when the front-end electronics is fully operational again.

The L0 electronics reset is encoded in a reset command (“01XXXXXX”) with bit 2 set. It may be performed together with other resets. The periodicity of the L0 electronics reset is configured with

[8] The current implementation of the L1 Buffer readout in the TELL1 does not allow this.

the parameter P_L0ERST_PER and the bunch crossing at which the reset sequence commences with P_L0ERST_OFS. The delay to send the reset command during which the L0 throttle is active to empty the L0 derandomizer is configured with the parameter P_L0ERST_DEL_DM. Normally this is set to 16 x 36 = 576. Once the reset command is sent, the parameter P_L0ERST_DEL_RT is used to specify the delay to release the triggers.

The L0 electronics reset generator is enabled with the parameter R_L0ERST_ENB. L0 electronics reset may also be requested via the ECS by setting the bit DMND_L0ERST irrespective of whether it is enabled or not.

In case the reset generator is not able to send the reset command at the specified bunch crossing because it clashes with another TTC command, a repeat bit is set to request another attempt at the specified bunch crossing in the next LHC turn. The status of the repeat bit can be read from the register S_RPT_L0ERST.

The reset generator may be reset individually with RST_L0ERST. During a L1 electronics reset it is held in its reset state.

The generation of the L0 Electronics Reset is implemented in the ODIN_l0e_reset.vhd.

14.5 L1 Electronics Reset

The L1 Electronics Reset activates a synchronous reset in the TELL1/UKL1 boards. Since the L1 electronics reset takes longer than the L0 electronics reset, the latter is always done at the same time. This reset generator is implemented in a way that it can be done “on the fly” during data taking. The sequences consist of[9]:

1. Throttle the L0 and the L1 triggers, stop the transmission of L1 triggers and transmit a flush of the L1 and the HLT Multi-Event Buffer.
2. Send the L0+L1 electronics reset command
3. Release the L0 and L1 triggers when the L1 trigger system is empty and the front-end electronics is fully operational again.

The L1 electronics reset is encoded in a reset command (“01XXXXXX”) with bit 3 set. Bit 2 is also set to activate a L0 electronics reset at the same time. It may be performed together with other resets as well. The periodicity of the L0+L1 electronics reset is configured with the parameter P_L01ERST_PER and the bunch crossing at which the reset sequence commences with P_L01ERST_OFS. The delay to send the reset command in order to empty the TELL1/UKL output buffers is configured with the parameter P_L01ERST_DEL_FLUSH. Once the reset command is sent, the parameter P_L01ERST_DEL_RT is used to specify the delay to release the triggers.

The L0+L1 electronics reset generator is enabled with the parameter R_L01ERST_ENB. L0+L1 electronics reset may also be requested via the ECS by setting the bit DMND_L01ERST irrespective of whether it is enabled or not.

In case the reset generator is not able to send the reset command at the specified bunch crossing because it clashes with another TTC command, a repeat bit is set to request another attempt at the

[9] This sequence is provisional and needs more discussion with all involved

specified bunch crossing in the next LHC turn. The status of the repeat bit can be read from the register S_RPT_L01ERST.

The reset generator may be reset individually with RST_L01ERST.

The generation of the L0+L1 Electronics Reset is implemented in the ODIN_l01e_reset.vhd

14.6 Periodic Command

Odin has a command generator to transmit a user configurable short TTC broadcast periodically or on demand. The TTC command is configured with the parameter P_USRCMD and the bunch crossing at which the periodic command is sent with P_PERCMD_OFS. The periodic transmission is enabled with the bit R_PERCMD_ENB and the periodicity is configured with the parameter P_PERCMD_PER.

The command may be requested via the ECS by setting the bit DMND_PERCMD irrespective of whether it is enabled or not. In case the command generator is not able to send the command at the specified bunch crossing because it clashes with another TTC command, a repeat bit is set to request another attempt at the specified bunch crossing in the next LHC turn. The status of the repeat bit can be read from the register S_RPT_PERCMD.

The command generator may be reset individually with RST_PERCMD.

The generation of the periodic commands is implemented in the ODIN_percmd.vhd

15 IP Destination Assignment

15.1 Overview

The IP transport format used for the L1 and the HLT channel requires the 48-bit Ethernet destination address and the 32-bit IP destination address. In order to only broadcast ten bits of destination address allowing up to 1024 different destinations, the SFCs' will all have the same Ethernet base address and IP base address consisting of the 38 and the 22 most significant bits, respectively. Each TELL1/UKL1 board stores the base addresses in programmable registers [??].

Odin broadcasts the IP destinations using the long broadcast format of the TTC system (Table 19). Bit [15..14] = "10" identify the destination broadcasts and bit 13 distinguishes between an HLT and a L1 destination broadcast. Odin incorporates a lookup table and a broadcast state machine. The state machine counts the number of L0 trigger accepts and L1 trigger accepts, and transmit a destination broadcast every *n* trigger accepts, where *n* is a programmable interval corresponding to the Multi-Event Packet packing factor for each data channel. The destination is retrieved by stepping through the lookup table in order to assign the MEPs to the SFCs in a round-robin manner. In order to implement a simple load balancing scheme[10] to take into account possible differences in the number of farm nodes associated with each SFC, the destination table has a depth equal to the

[10] This simple scheme will be replaced by the more flexible scheme conceived for the 1 MHz readout even if the L1 trigger is kept.

total number of farm nodes and contains repeatedly the same destination address for each SFC in proportion to its number of farm nodes. If the farm nodes have different CPU power, the numbers may be weighted with the CPU power.

In order to ensure that all events which are pending in the system are readout when the data taking is stopped, bit 12 in the destination broadcast defines a flush command. Even if the TELL1/UKL1 board has not received enough events to fill an MEP with the predefined number of events, the MEP is transmitted upon receiving a flush command.

The flush command is also transmitted before a L1 electronics reset to empty the output buffers in the TELL1/UKL1 boards.

Table 19 : Summary of the long broadcasts for the IP destination assignments.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP destination	1	0	Flush	R	Ethernet/IP address											

The IP destination broadcasting is implemented in ODIN_IP_broadcasting.vhd.

15.2 IP Destination Broadcaster

The IP Destination Broadcaster has two counters to counts separately the L0 trigger accepts and the L1 trigger accepts. The state machine fetches the L1 and the HLT IP destinations in the IP Lookup Table at rates which are specified with the parameters P_L1MEP_LEN and P_HLTMEP_LEN, respectively. Upon a flush signal, either on demand via the ECS or internally, the destination is immediately fetched and the accept counters are reset. The L1 and the HLT flush commands are sent on demand from the ECS system by setting the bit DMND_L1MEP_FLUSH and DMND_HLTMEP_FLUSH.

The IP Lookup Table contains a few more status bits per destinations which are used to veto a destination and which are evaluated by the broadcast state machine. The veto may be modified online via the ECS interface with the same write procedure as described below.

The state machine subsequently adds the IP destination header and writes the IP broadcast word to the IP destination broadcast queue. The queue is a FIFO which is absorbing pile-ups of broadcasts when the transmission of an IP destination over the TTC system is delayed due to the TTC Broadcaster being busy with another transmission or a clash with a command of higher priority. The status of the broadcast queue may be read from S_IP_FIFO_FULL_INST (instantaneous) and S_IP_FULL_CONT (continuous).

The IP broadcast word is also transferred simultaneously via an SPI-like (Serial Peripheral Interface) bus to the front-end processing to activate the transmission of the Odin data bank.

As soon as the destination broadcast queue contains an IP broadcast the destination broadcaster requests an IP broadcast to the TTC Broadcaster.

The IP destination broadcasting is enabled/disabled with the bit R_IP_ENB and may be reset individually with the bit RST_IP.

15.3 IP Lookup Table

The 10-bit IP destinations are located in a lookup table which is 16 bits wide and 1024 destinations deep. The memory is filled via the ECS. The start address at which the writing should start (first position in the memory has address 0x0) is written to the address register P_IP_ADDR. The entry in memory at which the address is pointing is written by writing the data to the data register P_IP_DEST. The address pointer is automatically incremented on each write. The current address may be read from P_IP_ADDR.

The same procedure is used for the reading. The start address at which the reading should start is written to the address register P_IP_ADDR. Reading the data register P_IP_DEST returns the data at the current address and increments automatically the address.

In order for the state machine to know the number of destinations available in the memory, it must be specified with the parameter P_IP_ADRANGE. The state machine always returns to the first position upon reaching the position specified with the P_IP_ADRANGE.

16 TTC Broadcaster

The transmission of all TTC broadcasts via channel B is handled by the TTC Broadcaster. The broadcaster receives transmission requests from the L1 Trigger Transmitter, the IP Destination Broadcaster and the different control command generators. In the case of the L1 trigger the TTC Broadcaster also receives the 5-bit L1 trigger word and in the case of the IP broadcasts, the 16-bit IP broadcast word. The TTC Broadcaster then calculates the Hamming code and compiles the entire TTC broadcast frame. The transmission time of the short and the long TTC broadcasts are 16 and 42 clock cycles, respectively, due to the serialization.

In order to handle simultaneous requests, the transmission of the various broadcasts is done according to a priority scheme (Table 20). The Bunch Counter and the Event Counter Reset have the highest priority.

Table 20: The TTC broadcasts and the priority level (highest value corresponds to highest priority)

TTC broadcast	Encoded	Priority
Bunch Counter Reset / Event Counter Reset	“00000001” / “0000011”	9
L0+L1 electronics reset	“00001110”	8
L0 electronics reset	“00000110”	7
Calibration command A	“00010000”	6
Calibration command B	“00010100”	5
Calibration command C	“00011000”	4
User command	“XXXXXXX”	3
IP destination	See below	2
Trigger Type	“1XXXXX00”	1

The priority handler treats contention differently for the control commands, which are associated with a specific bunch crossing, and the IP and the L1 trigger broadcasts which must be sent as soon as possible. If an IP destination or a L1 triggers broadcast request clashes with a request of a control command of higher priority or arrives while the TTC Broadcaster is busy with the transmission of a

preceding broadcast, the broadcast of the IP destination or L1 trigger is only delayed until the preceding broadcast has finished. The control commands cannot be delayed and the request is therefore made prematurely in order to activate a veto period before the actual clock cycle in which the transmission should start. The veto period corresponds to the transmission time of one broadcast and prevents any other broadcast from starting later than a time which would interfere with the control command to be sent. However, a preceding broadcast may finish during the veto period.

In case two control commands are requested at the same time or one is requested during the transmission of another, the control command of lower priority is simply postponed until the next LHC orbit. This means that a command of lower priority must not always clash with a command of higher priority or it must have a shorter periodicity. The actual configuration of all control commands is checked in software.

The TTC Broadcaster transmits an acknowledge signal individually for each type of broadcast upon accepting a broadcast.

The TTC Broadcaster is implemented in ODIN_broadcaster.vhd.

17 Onboard TTCrx

The onboard TTCrx is intended for receiving Beam Synchronous Timing information from the LHC which are sent as a set of long broadcasts over dedicated TTC fibres for each turn [??]. In particular Odin extracts the GPS time in order to store it in the event data. The long broadcasts from the TTCrx are decoded in the Q_MP module and the GPS time (GPS_TIME(39..0)) is written to the FEB at each L0 trigger accept.

The BST system also transmits information such as the state of the LHC accelerator, bunch currents etc which may be of interest to cross-check in the ECS system.

The TTCrx may also be used in a lab test to cross-check that the TTC information sent by Odin is correctly transmitted over the TTC by feeding back a TTC fibre. Odin incorporates a full TTC command decoder and counters for each type of information sent. The counters have the prefix “C_TTC_” and they are listed in Appendix G.

The TTCrx can be configured via its I^2C bus. On Odin the TTCrx’s are normally configured to have the peripheral base address 0xEC (Table 7). This allows the software to perform the read and write procedures of all I2C devices on Odin in the same way.

The TTCrx may be reset via the Local Bus by the bit RST_TTCRX.

18 Detector Status Input

Odin has a 24-bit LVDS input on the rear of the board (P17) which is intended for receiving a ~ 2-bit status information from each sub-detector. The information is recorded in the event data at each L0 trigger accept. This allows encoding an event quality word per sub-detector and per event which can give a hint about problems and which is fast to retrieve instead of interrogating for instance the ECS condition database.

The data present on the input is directly written into the FEB (EXP_STATUS(23..0)) at each L0

accept. This has the disadvantage that the information must be correctly time-aligned externally in either a receiver unit which fans in the information from the different sources or at the source. The input is enabled by setting bit 0 of the I^2C port on peripheral address 0x40 as shown in Table 9.

The connector is a 50-pin 3M Pak 100 4-wall header with latch (3M 3433-5602). The pin configuration is shown in Figure 13. Notice the order of the LVDS+/- and that the 25th pair (pins 49 and 50) should be ground. The position of pin 1 is the same as for the connector in Figure 13. The termination scheme and the receiver type is the same as the L0 trigger input.

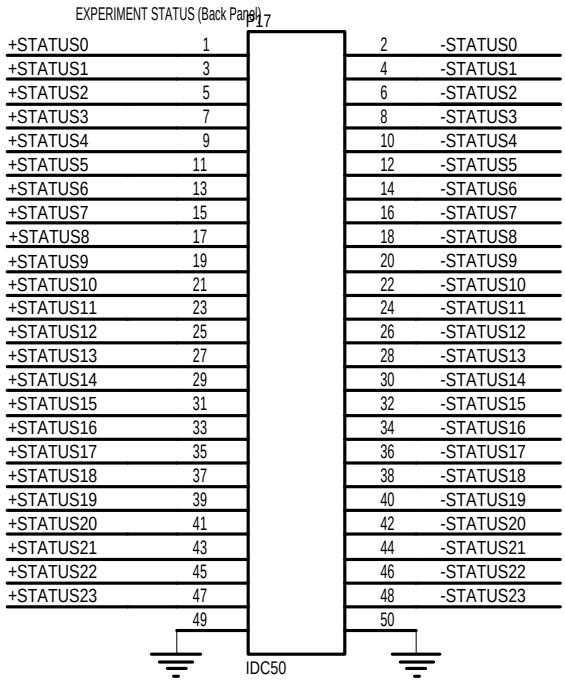


Figure 13: Pin configuration of the experiment status input on the rear side of Odin

19 Bunch Crossing Information Input

Odin has a general purpose 8-bit LVDS input on the front-panel which may be used for connecting an external device which provides information per bunch (BX_DATA(7..0)) such as for instance bunch intensities [??]. The input is enabled by setting bit 1 of the I2C port on peripheral address 0x40 as shown in Table 9.

A L0 pipeline analogous to the front-end L0 pipeline allows aligning the data to the trigger decisions (ODIN_bx_pipeline.vhd) such that the information is logged together with the L0 event data. The depth of the pipeline is configured with the parameter P_BX_PIPE_LEN.

The connector (P12) is a 20-pin 3M Pak 100 4-wall header with latch (3M 3428-5602). The pin configuration is shown in Figure 14. Notice the order of the LVDS+/- and that the 10th pair (pins 19 and 20) should be ground. The position of pin 1 is the same as for the connector in Figure 10. The termination scheme and the receiver type is the same as the L0 trigger input.

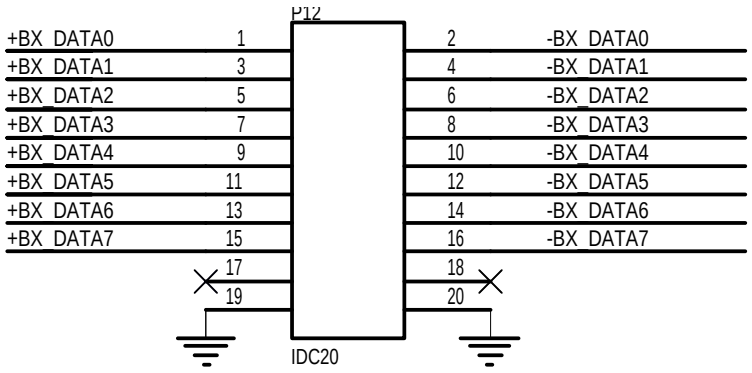


Figure 14: Pin configuration of the bunch crossing information input

20 ODIN L1 Front-End

20.1 Overview

Odin incorporates a L1 Front-End analogous to a TELL1/UKL1 board in order to record local event information and provide the DAQ system with the data on an event-by-event basis. The "Odin data bank" contains centrally recorded information about the identity, the source and the quality of an event, and the L1 trigger information. The Odin data bank is merged with the other event data fragments during the event building. Table 21 list the information in the Odin data bank.

Table 21: Contents of the Odin data bank and the level at which it is recorded

Level	Data	#bits
L0	L0 Event ID	24
	Bunch ID from Odin	12
	Bunch ID from L0DU	12
	GPS time	40
	Trigger type	3
	L0 Force bit	1
	L0 synch error	1
	L0 synch error (forced accept)	1
	Detector status	24
	BX type	2
	L0 bunch current	8
	Subtotal	128
L1	L1 Event ID	32
	L0 Event ID in MDP	32
	L1 Error/status block	32
	L1 decision info (MDP)	6 x 32
	Subtotal	288
	Total	52 bytes

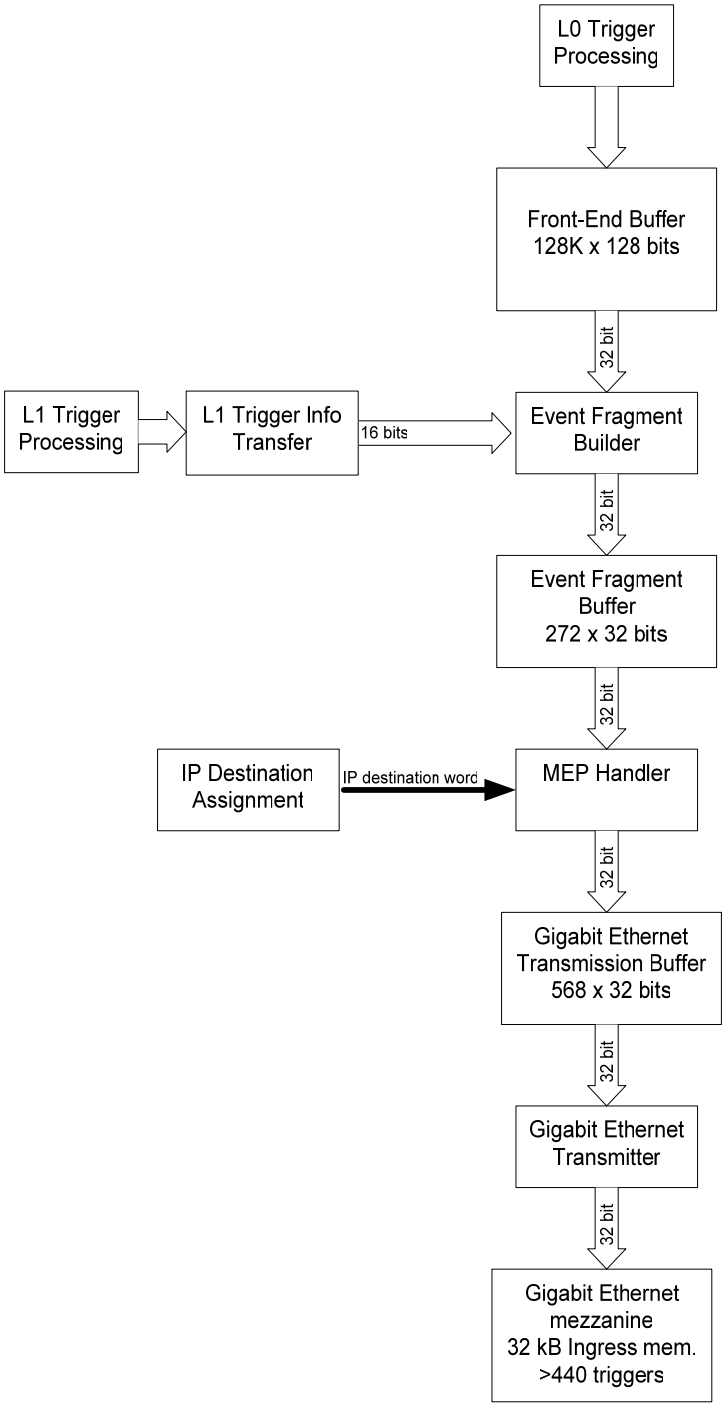


Figure 15: Block diagram of the front-end processing.

The Odin front-end processing is shown in Figure 15. The data recorded at each L0 trigger accept is transmitted to the Odin L1 Front-End via the Front-End Buffer (FEB). The data recorded at each L1 trigger accept is transmitted via the 16-bit data bus between Q_L1 and Q_FE (Section 13.5).

The L0 event data are read out by the Event Fragment Builder upon each transmission of a L1 trigger broadcast, and are either immediately rejected or kept and formatted while the L1 event data are transferred from the L1 trigger processing. The L1 event data are subsequently formatted and

merged with the L0 event data to form an event fragment, containing the Odin data bank[11], which is written to the Event Fragment Buffer.

The HLT IP destination broadcast is received directly from the IP Destination Broadcaster. The reception activates the Multi-Event Packet (MEP) Handler to read the event fragments from the Event Fragment Buffer, prepare the MEP including the IP/Ethernet header and write it to the Gigabit Ethernet Transmission Buffer.

The transmission of the MEP over Gigabit Ethernet is handled by the Gigabit Ethernet Transmitter.

Once the last L1 trigger accept, which closes an MEP, is broadcasted, the total front-end processing time to send the MEP is shorter than the interval between broadcasts of L1 trigger accepts.

20.2 Front-End Buffer

The Odin Front-End Buffer (FEB) stores the L0 event data recorded for each L0 trigger accept during the L1 trigger processing. The contents and the format are shown in Table 22.

Table 22: Data format in the FEB buffer.

FIFO pair	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
#0	L0 EVENT ID (9..0)										FRC	TRG TYPE(2..0)			BID(1..0)	
#1	BID(3..2)		L0 EVENT ID(23..10)													
#2	BX_DATA(7..0)								BID(11..4)							
#3	SNC KEEP	SNC ERR	BXTYPE(1..0)		BID(11..0) (L0DU)											
#4	GPS_TIME(15..0)															
#5	GPS_TIME(31..16)															
#6	EXP_STATUS(7..0)								GPS_TIME(39..32)							
#7	EXP_STATUS(23..8)															

The FEB consist of 16 discrete 128K x 9-bit FIFOs (Cypress CY7C4292V-10) with independent read and write buses. Only eight of the nine bits are used in each FIFO. The read and write actions are clock synchronous.

The 128 bits in the 16 FIFOs are written simultaneously in one clock cycle to allow any number of consecutive L0 trigger accepts. The buffer is read in a de-multiplexing manner. A single write enable puts the next data on the read bus of the FIFO. Two FIFOs at a time are subsequently read by enabling the output bus for the pair of FIFOs one after the other. In total it takes nine clock cycles to read out the event.

The Empty and the Almost Full FIFO status flags are available via the Local Bus as S_FEB_EMPTY, respectively S_FEB_FULL_CONT (continuous) and _FEB_INST

[11] The format is not yet fixed

(instantaneous). The Almost Full signal is also used to generate an internal safety L0 throttle. The throttle is enabled with the same parameter (R_FIFO_L0THR_ENB) which enables the safety throttle from the L0 Accept FIFO.

The Front-End Buffer may be reset asynchronously via the Local Bus by the bit RST_FEB.

20.3 Event Fragment Buffer

The Event Fragment Buffer must be able to contain 16 event fragments (maximum HLT packing factor). The Odin event fragment consists of 64 bytes which includes 12 spare bytes for additional data. Including the 4 bytes of event fragment header [??] it means that each event fragment in the MEP contains 68 bytes. The depth of the Event Fragment Buffer is 272 x 32 bits.

20.4 Multi-Event Packet Handler

The Multi-Event Packet Handler receives the HLT IP destination word (16 bits) directly from the IP Destination Broadcaster via a serial bus between Q_MP and Q_FE. The destination word triggers the MEP Handler to read the Event Fragment Buffer and to prepare the MEP header [??] and the IP/Ethernet header with the partition ID, size, the destination address and the source address. The partition ID is configured with the parameter P_PARTITION_FEID. The IP destination address is compiled from a base address and the ten bits received from the destination assignment. The base address for the IP destination is configured with the parameter P_HLTIP_BASE and the source address with the parameter P_FEIP_ADDRESS.

The MEP Handler reads all event fragments available in the Event Fragment Buffer it means that an HLT flush is handled automatically.

The size of an Odin MEP including the IP/Ethernet header is equal to the packing factor * 68 bytes(event fragment) + 12 bytes(MEP header) + 36 bytes(IP header).

20.5 Gigabit Ethernet Transmission Buffer

The Gigabit Ethernet Transmission Buffer stores the MEPs while waiting to be transmitted via the Gigabit Ethernet. In order to store two MEPs with 16 events (maximum HLT packing factor), the depth of the transmission buffer is 2 x 284 x 32 bits.

20.6 Gigabit Ethernet Transmitter

The transmission of the MEPs via Gigabit Ethernet is handled by the Gigabit Ethernet Transmitter. Any time a full MEP is available in the Gigabit Ethernet Transmission Buffer, the transmitter sends it immediately.

The Gigabit channel by which the MEP are sent is configured with the parameter P_FE_GBE_CHANNEL.

The data interface with the Egress of the Media Access Controller (Intel IXF1104) on the Gigabit Ethernet mezzanine is based on the SPI-3 interface operating in 32-bit mode. On Odin the Egress SPI-3 bus is clocked at 80MHz. The bus clock is synchronous with the system clock and is generated using an internal PLL inside the Q_FE FPGA. The lock status of the L1 PLL in Q_L1 and the FE PLL in Q_FE is indicated by the LED D5-R on the front-panel.

The MAC has a 10 KB internal Egress FIFO. The status of the Egress FIFO may be read via the Local Bus from the registers S_GBE_EMPTY and S_GBE_FULL.

The Gigabit Ethernet Transmitter is implemented in ODIN_DAQ_link.vhd.

21 Gigabit Ethernet MAC Control

The control bus of the Gigabit Ethernet Media Access Control is a simple asynchronous microprocessor bus. Since it is not compatible with the Local Bus on Odin, the conversion is made inside the Q_FE FPGA which generates the protocol of the MAC control bus. Configuration of a MAC register is carried out by writing the address via the Local Bus to an address register in the Q_FE (P_MCONF_ADDR) and subsequently writing the configuration data via the Local Bus to a data register (P_MCONF_DATA). Reading a MAC register is done by writing the address to the address register and subsequently reading the data register.

The MAC may be reset individually via the Local Bus with the bit RST_MAC.

The control bus conversion for the MAC is implemented in ODIN_mac_control.vhd

22 High-Level Control Software

To be written later

23 Board In-Situ Test

There are several means of testing Odin in-situ:

- Boundary scan of all JTAG compatible devices may be done from the Credit Card PC command line with the command `bscan`.
- The command `tstat` returns the Odin board identifier read in the I^2C EEPROM at the peripheral address 0xA0 and the VHDL code versions in the Glue *light* mezzanine and the four main FPGAs (Q_MP, Q_L0, Q_L1, Q_FE).
- The Glue *light* mezzanine contains a 32 bit r/w test register on address 0x38 which may be used to test the PCI bus between the Credit Card PC and the Glue mezzanine.
- Each main FPGA contains a 32 bit r/w test register on address (base + 0xF8) which may be used to test the Local Bus

24 1MHz Readout Option

Odin has been evaluated in the 1 MHz readout scheme. Mainly the changes concern the IP destination assignment and the front-end processing.

Figure 16 shows a block diagram of an alternative improved IP destination handling which includes more than 2048 destinations, special destinations for calibration triggers, flushing of destination triggers and both a static and a dynamic load balancing.

In order to improve the synchronization check in the TELL1/UKL1 another long TTC broadcasts has been added which for every IP destination broadcast transmits 12 bits of the L0 Event ID of the first event in the MEP. This is made possible by the fact that sufficient TTC bandwidth is freed by abandoning the L1 trigger broadcasts.

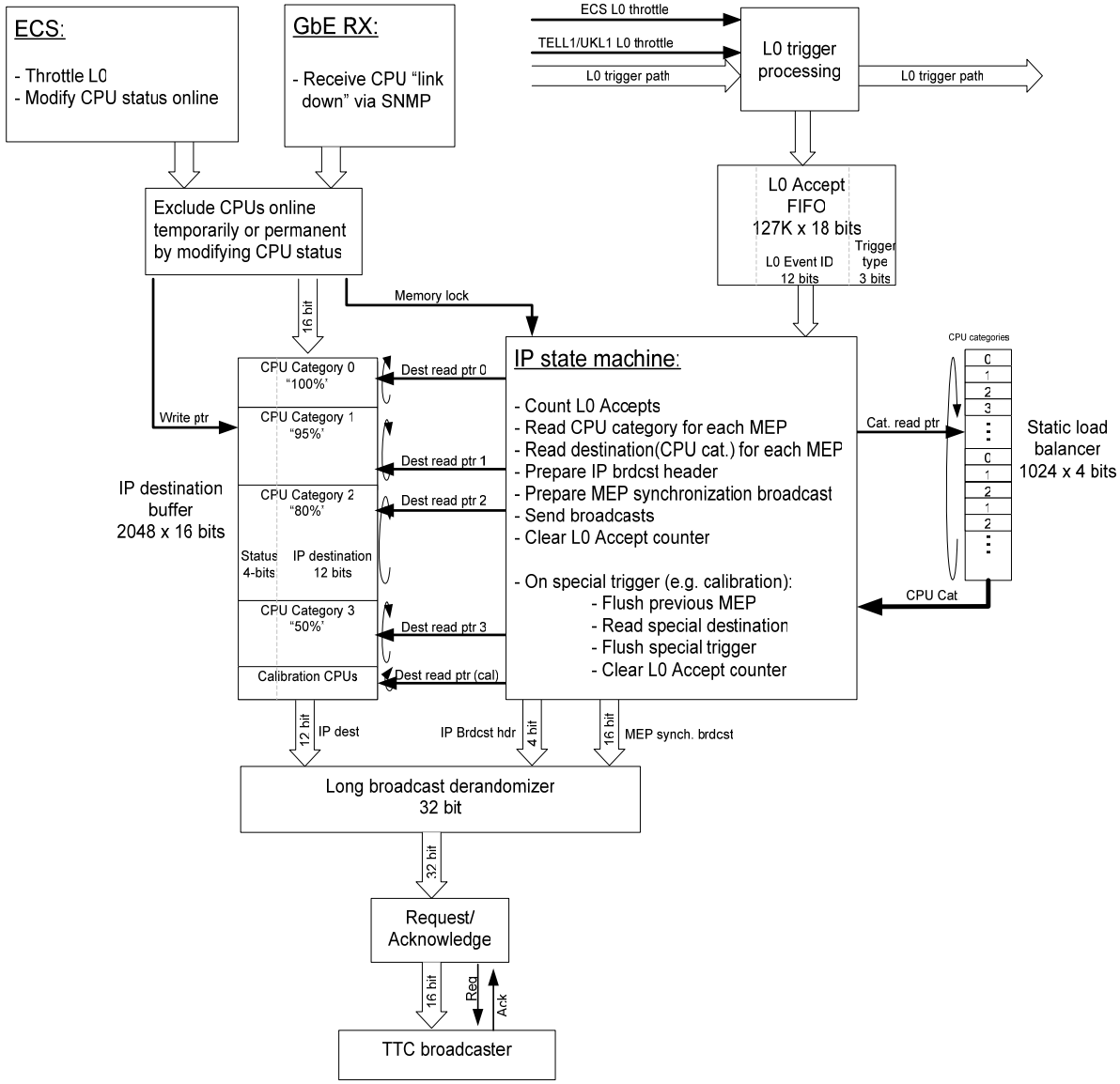


Figure 16: Block diagram of an alternative IP destination assignment and load balancing.

Table 23: Long broadcasts in the 1MHz readout scheme

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP destination	1	0	0	Flush	Ethernet/IP address											
MEP synchronization	1	0	1	R	L0 Event ID of first event in MEP											

In the 1MHz scheme, the Odin front-end processing is reduced to handling only the L0 event data. Sufficient resources and bandwidth are available in the Q_FE to transmit the Odin L1 MEPs at 1MHz.

References

Appendix A. Overview of FPGA Functions and I/Os

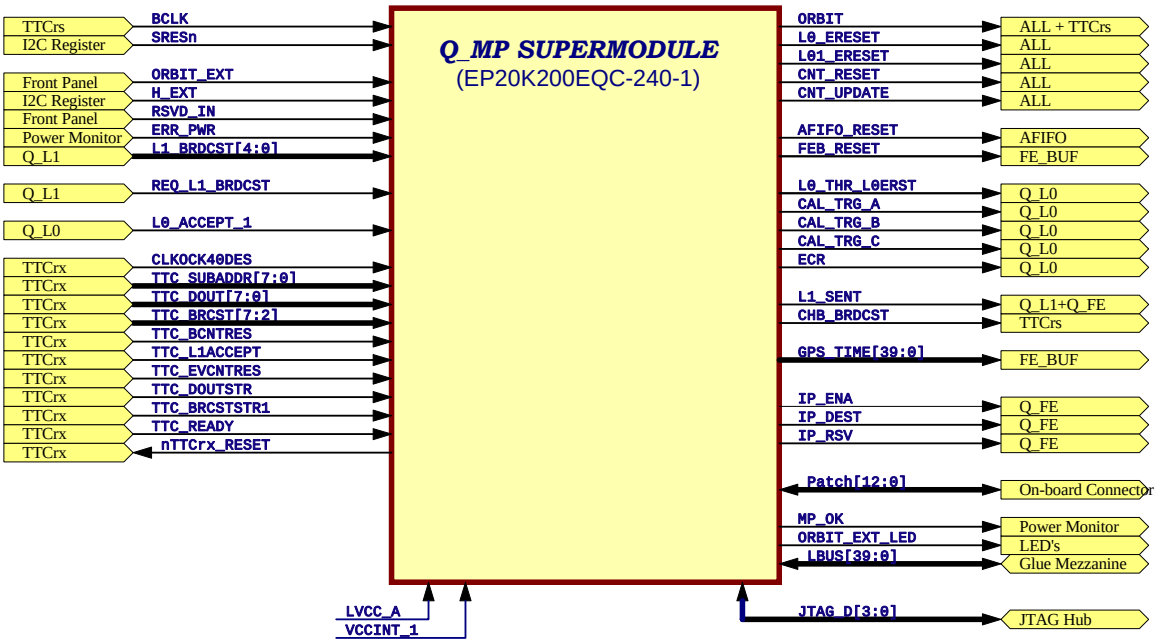


Figure 17: Overview of the Q_MP FPGA.

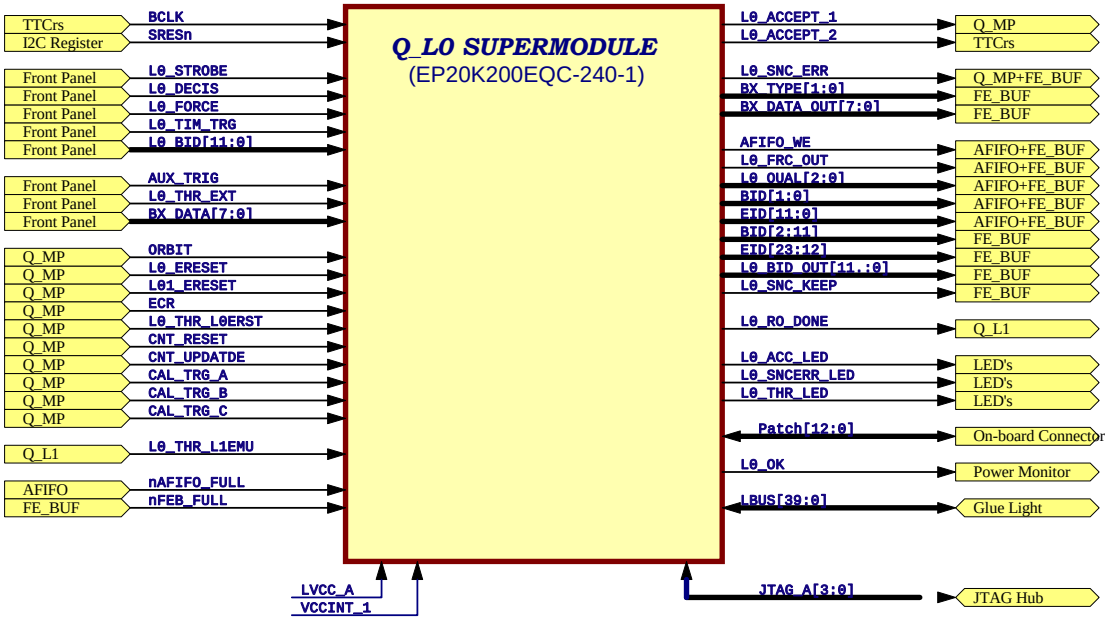


Figure 18: Overview of the Q_L0 FPGA.

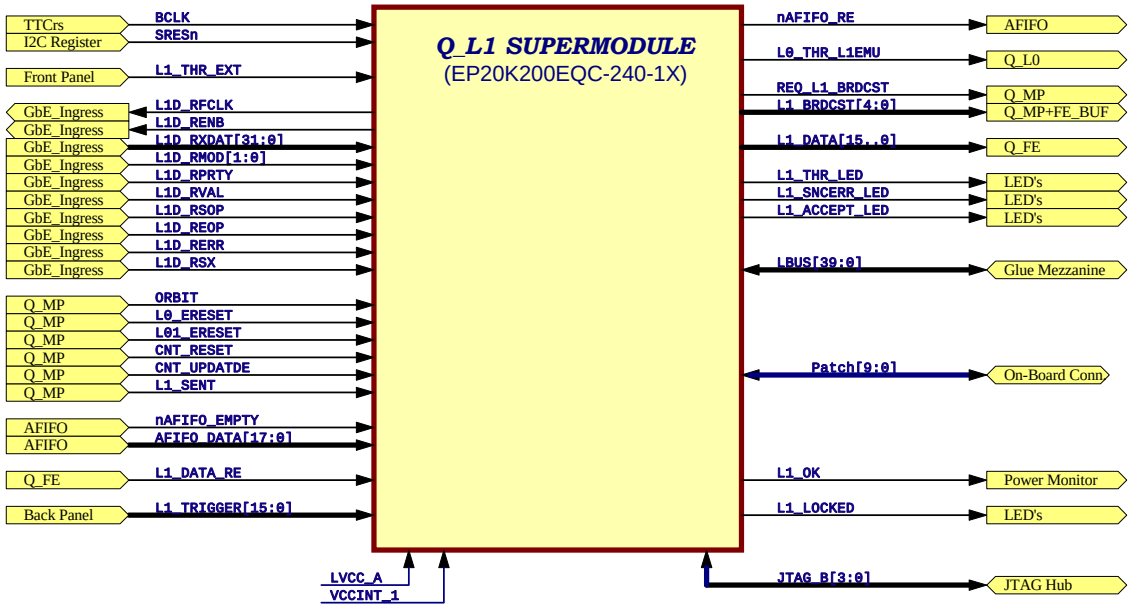


Figure 19: Overview of the Q_L1 FPGA.

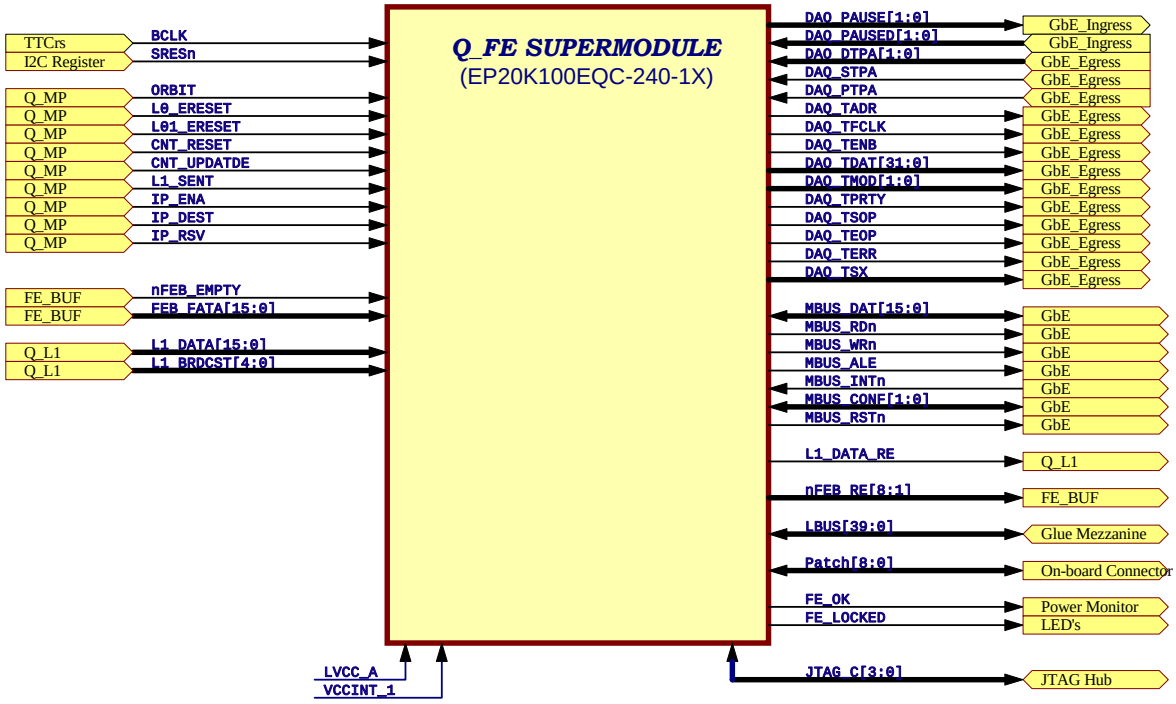


Figure 20: Overview of the Q_FE FPGA.

Appendix B. Front-Panel LEDs

Table 24: Summary of the front-panel LEDs on Odin.

Designator	Function	Normal Activity		Abnormal Activity	
D1-L	Power Supply Status	Power OK	Green	+5V Out of Range	Blinking Red
D1-R	System Reset	OFF	Green	ON	Red
D2-L	All FPGA's Configured	Yes	Green	No	Red
D2-R	PLL Locked	Locked	Green	Unlocked	Red
D3-L	External Orbit Present	Yes	Green	No	Red
D3-R	External Clock Present	Yes	Green	No	Red
D4-L	BCLK & Orbit Selection	External	Green	Internal	Red
D4-R	L0 Sync Error	No	Blank	Yes	Red
D5-L	L1 Accepts Present	Yes	Green	No	Blank
D5-R	L1 Sync Error	No	Blank	Yes	Red
D6-L	Reserve	Yes	Green	No	Blank
D6-R	L0 Accepts Present	Yes	Green	No	Blank
D7-L	L0 Throttle Present	No	Blank	Yes	Red
D7-R	L1 Throttle Present	No	Blank	Yes	Red

Appendix C. Front and Rear Connections

Table 25: Summary of the front-panel and the rear connections.

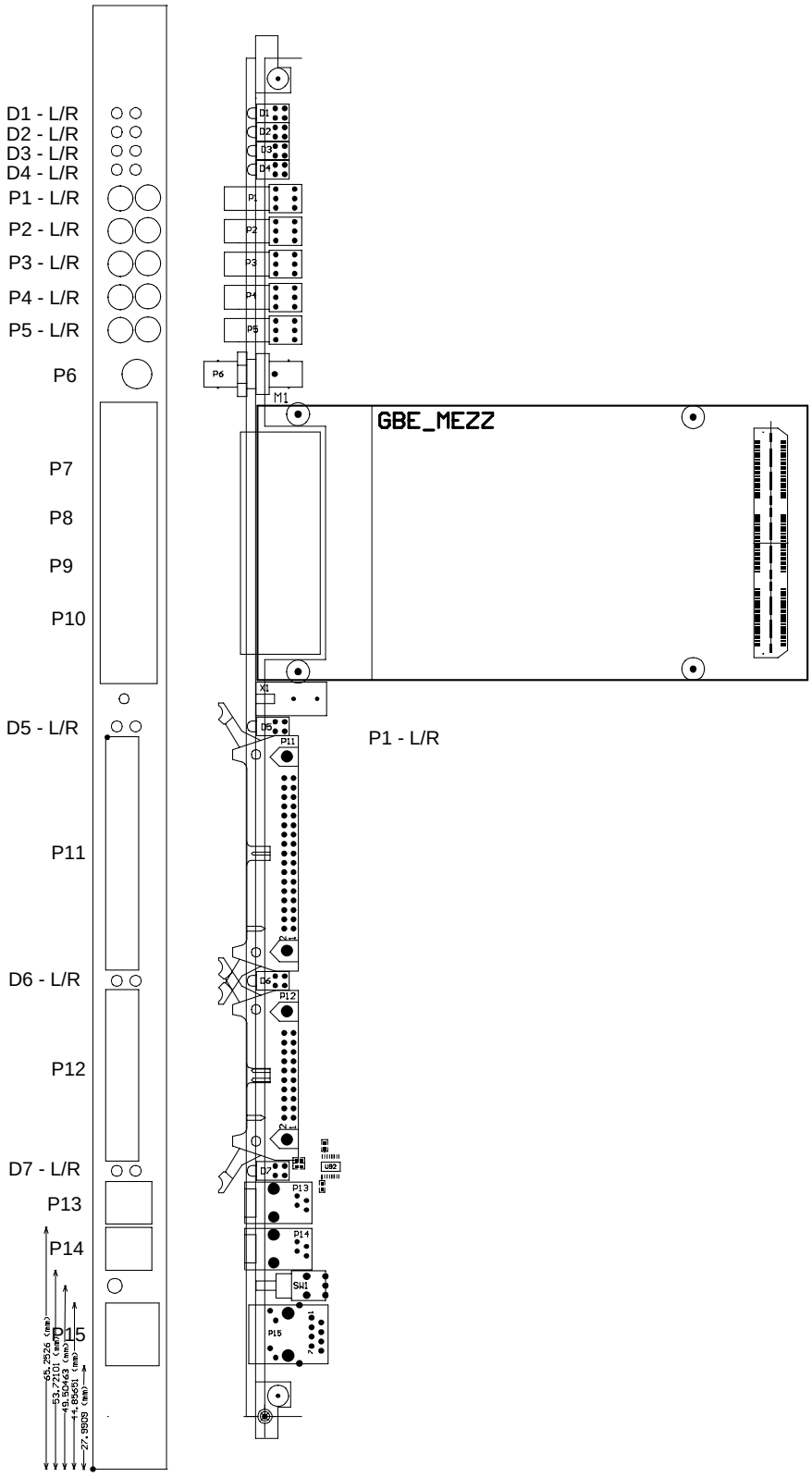
Designator	Function	Mode	Signal	Location	Connector
P1-L	Orbit	In	DC ECL	Front Panel	Dual LEMO
P1-R	Orbit	Out	DC ECL	Front Panel	Dual LEMO
P2-L	Bunch Clock	In	AC ECL	Front Panel	Dual LEMO
P2-R	Bunch Clock	Out	AC ECL	Front Panel	Dual LEMO
P3-L	TTC Channel A+B	Out	AC ECL	Front Panel	Dual LEMO
P3-R	TTC Channel A+B	Out	AC ECL	Front Panel	Dual LEMO
P4-L	TTC Channel A	Out	DC ECL	Front Panel	Dual LEMO
P4-R	TTC Channel B	Out	DC ECL	Front Panel	Dual LEMO
P5-L	Auxiliary Trigger	In	DC ECL	Front Panel	Dual LEMO
P5-R	Reserve	In	TTL	Front Panel	Dual LEMO
P6	BST Info	In	Optical	Front Panel	ST/PC
P7-P10	L1 Trigger	In	GbEthernet	GbE Mezzanine	RJ45
	DAQ	Out	GbEthernet	GbE Mezzanine	RJ45
P11	L0 Trigger	In	LVDS	Front Panel	3M 34-pins
P12	Bunch Crossing Info	In	LVDS	Front Panel	3M 14-pins
P13(0)	L0 Throttle 1	In	LVDS	Front Panel	RJ9
P13(1)	L1 Throttle 1	In	LVDS	Front Panel	RJ9
P14(0)	L0 Throttle 2	In	LVDS	Front Panel	RJ9
P14(1)	L1 Throttle 2	In	LVDS	Front Panel	RJ9
P15	Ethernet CCPC	I/O	Ethernet	Front Panel	RJ45
P16	L1 Trigger	In	LVDS	Back Panel	3M 34-pins
P17	Experiment Status	In	LVDS	Back Panel	3M 50-pins
VME-P1	Power Supply	-	-	Back Panel	DIN96_M

Appendix D. Summary of Headers and Jumpers

Table 26: Summary of Headers and Jumpers

Jumper	Function
J1	Q_L1 test pad (pins 0 and 1)
J2	Q_FE test pad (pins 0 and 1)
J3	JTAG header for programming of JTAG hub
J4	Q_L0 test pad (pins 0 – 8)
J5	External JTAG master source
J6	Q_MP test pad (pins 0 – 11)
J7	RS232 interface to CCPC
J8	Voltage selection for external JTAG master
J9	Write protection for board identifier
J10	Reserve control input for JTAG hub

Appendix E. Front-Panel Layout



Appendix F. Board overview

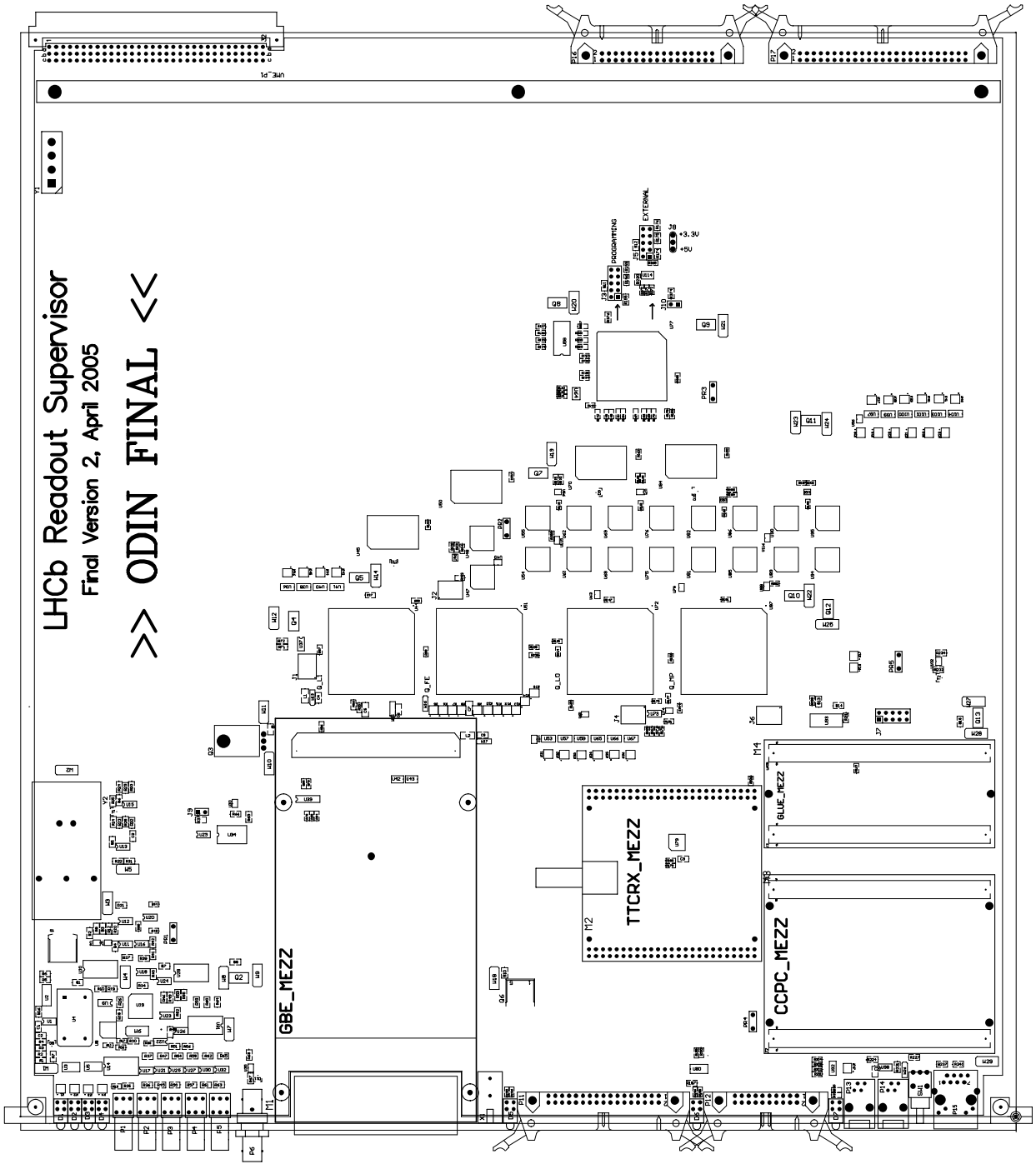


Figure 21: Top view of the Odin board.

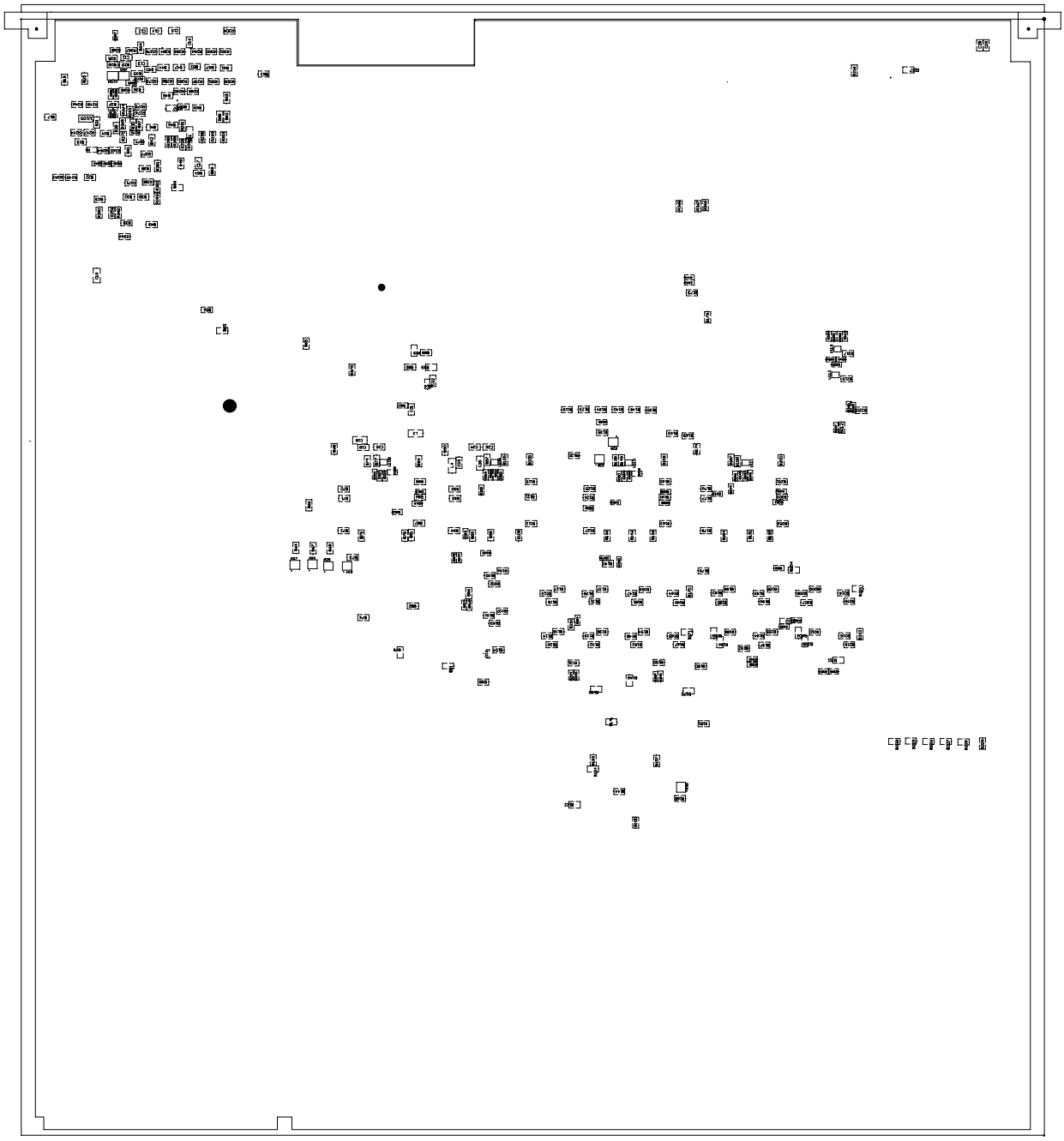


Figure 22: Bottom view of the Odin board.

Appendix G. Summary of control and status registers

Appendix H. Check List for Production Testing

Table 27: Check list for the production testing.

Step	Check
1	Examine PCB
2	Examine mounting
3	Fine-pitch components
4	Check power
5	Power monitoring
6	Clock distribution
7	CCPC + Ethernet
8	Glue card
9	I2C bus
10	I2C register
11	External clock
12	Clock phase alignment
13	Jitter
14	Backplane reset + reset button
15	Programming JTAG hub
16	Controlling JTAG HUB
17	Programming Q_MP
18	Programming Q_L0
19	Programming Q_L1
20	Programming Q_FE
21	Q_FE PLL
22	Q_L1 PLL
23	System reset
24	Test local bus
25	LEDs
26	TTC encoder
27	TTC out
28	TTC channel A/B out with TTCvx
29	Orbit signal
30	Programming Q_MP EPC4
31	Programming Q_L0 EPC4
32	Programming Q_L1 EPC4
33	Programming Q_FE EPC4
34	JTAG BSCAN
35	FPGA interconnects
36	L0 Accept FIFO
37	L1 front-end buffer
38	L0 trigger input
39	L1 trigger input
40	Throttle inputs
41	Auxillary trigger input
42	Detector status inputs
43	GbE control bus
44	GbE sending
45	GbE receiving
46	TTXrx power and reset
47	TTCrx clock
48	TTCrx I2C bus
49	TTCrx L0 triggers
50	TTCrx short broadcasts
51	TTCrx long broadcasts