Detector Trigger Unit

Technical Manual

Author : Erik Lins
Last modified : 26-April-2000

General

The HADES trigger system consists of three levels:

- The first level trigger (LVL1) is derived from an external source, depending on the experimental conditions, i.e. AND of multiplicity in TOF and of Start, or some other reaction trigger, or calibration trigger (is outside of scope of this document). This signal is used to initiate the readout of the frontend modules, i.e. the event data is pushed into the first level pipe.
- The second level trigger (LVL2) system employs information from the RICH detector and META detector (Shower detector and TOF wall). In a first step a ring search in the RICH detector, a search for electromagnetic showers and a search for hits in the TOF wall is performed by the second level trigger processors. In a following step the obtained information of the first step is used to construct tracks, by matching the angle information from ring centers and corresponding electron hits in the TOF and hower detector. The second level trigger signal is generated by the matching unit and is used to either push the event data from the first level pipe into the second level pipe, or to clear the event data in the first level pipe, depending on a positive or negative trigger decision.
- The third level trigger (LVL3) algorithm is not yet specified. It will search for lepton tracks in the MDC's utilizing the hit information of the RICH and TOF/SHOWER detectors. The track search will be solely based on the MDC hit pattern disregarding the drift time information. However, more complex tracking schemes which might involve all detector systems (e.g. improved TOF timing) might be implemented as a second generation trigger.

Due tu the principle of the HADES trigger system also events without bias of dilepton pairs can pass the LVL2 and LVL3 trigger system at a downscaled rate.

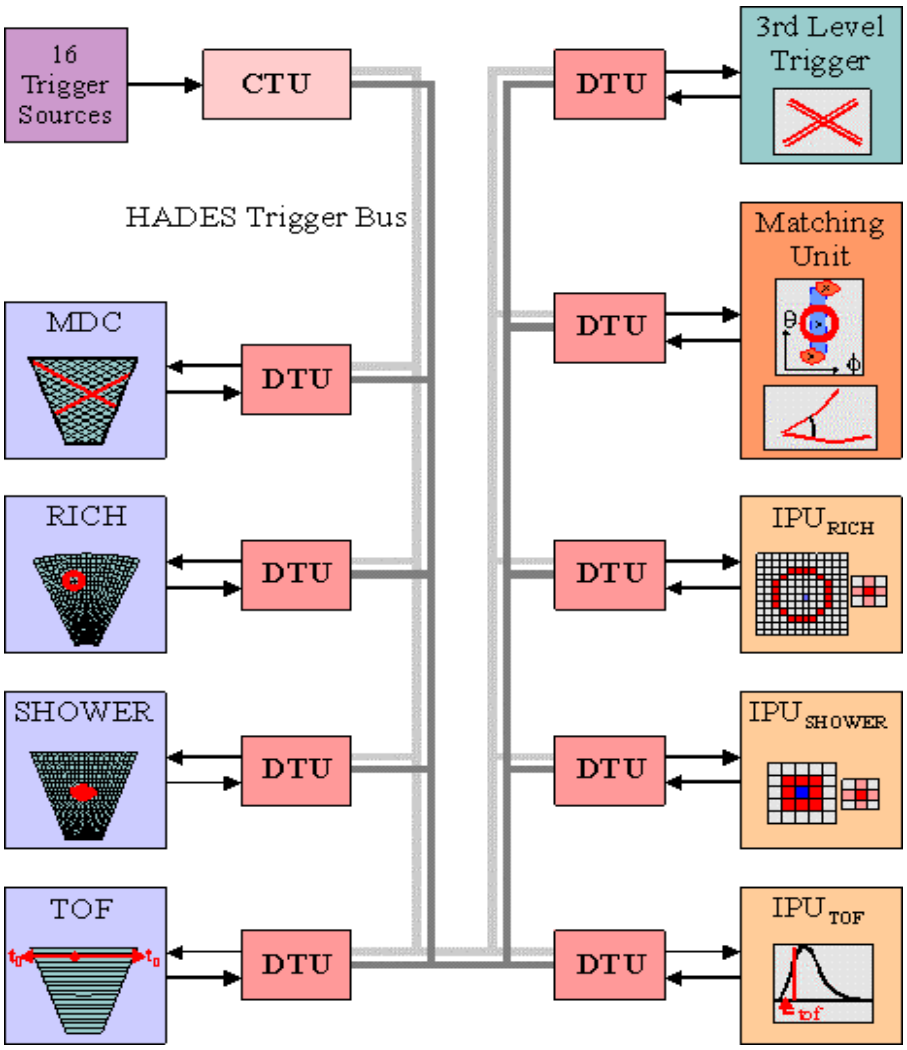
Concept

The first and second level pipes can be located either on the frontend modules or readout controllers (MDC, RICH) or on the second level trigger processors (Shower, TOF). Hence the trigger signals have to be transmitted to any device, on which a data pipe is located. A common trigger bus, that is separated into two parts according to the first two trigger levels, is distributed from a central trigger unit (CTU, described below) to the readout systems and second level trigger processors. This CTU receives up to 16 trigger signals (e.g. minimum bias trigger, central event trigger, calibration trigger) and generates the first level trigger decision, a fast trigger output and busy signal for gating purposes of timing critical signals, as well as a trigger code and an trigger tag, which is distributed over the trigger bus.

The CTU should be located nearby the detector system, because some readout systems require a very precise timing of the trigger signal, which can not be guaranteed for the trigger bus signals and therefore these readout systems will receive an additional trigger bus independant trigger signal (generated by the Multiplicity Trigger Unit).

Due to different signal and timing requirements of the various readout systems, a detector specific Detector Trigger Unit (DTU, described below) is used to convert the incoming trigger signal, trigger code and trigger tag into readout specific signals.

The second level trigger decision is generated by the Matching Unit, which should be located in the same crate as the CTU. This simplifies the transmission of the second level trigger decision to the CTU via private bus on the VME J2 connector.



CTU and DTU modules

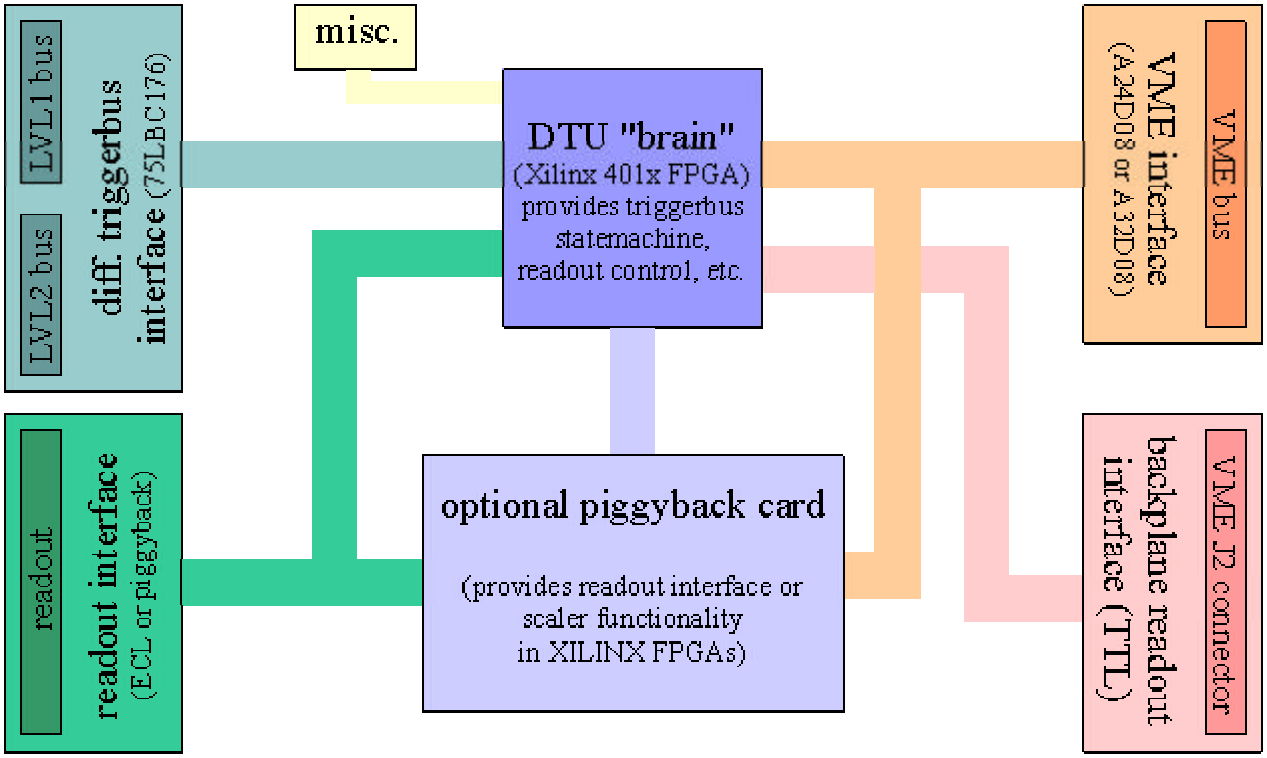
The following table summarises the main requirements of the CTU and DTU modules.

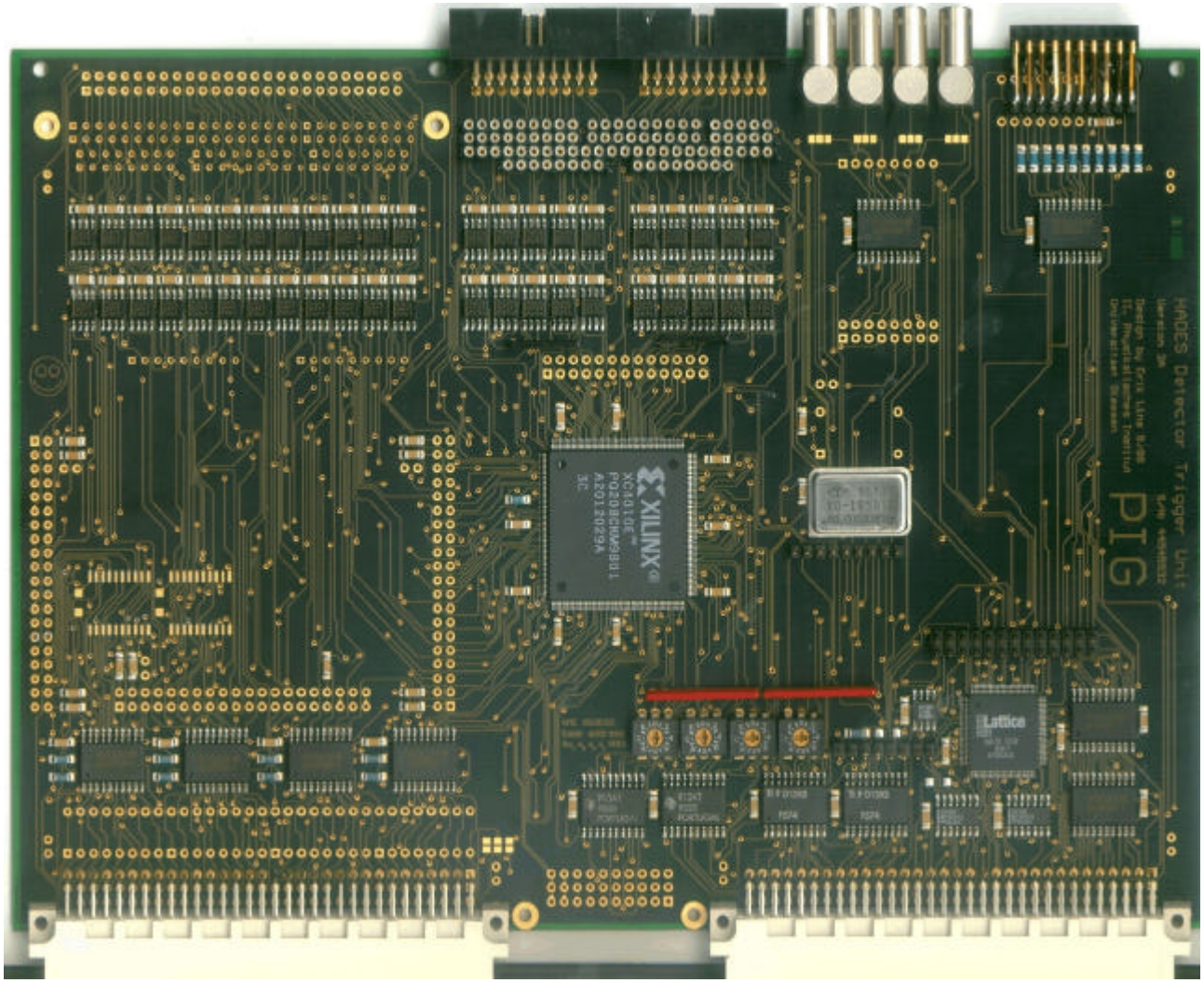
hardware: • VME slave interface (A32D08/D16) • HADES trigger bus interface • local readout trigger interface • FPGA for flexible controlling of trigger bus and readout system *software:* • handling of the incoming HADES trigger bus signals (trigger code, trigger tag) • controlling of the various components of the certain readout system • monitoring of the readout system and generation of busy and error signals if necessary • VME access to all functions for slow control and debugging purposes	*Hardware:* • VME slave interface (A32D32) with DMA slave access and interrupt capability • HADES trigger bus interface local interface for 16 trigger signals • FPGA for flexible controlling of trigger bus and trigger signals *Software:* • generation of LVL1 trigger decision • generation of trigger code and trigger tag • controlling of HADES trigger bus • handling of busy and error conditions responded by the DTUs • VME access to all functions for slow control and debugging purposes *additional:* • input capability of 16 differential ECL trigger channels • three stages of 32 bit scalers for each trigger channels • programmable 16 bit downscalers for each trigger channel • 16 bit trigger channel pattern sample for each valid trigger • generation of fast-trigger-out signal for external gating purposes • DMA access to pattern sample registers (allows standard readout)

Since the hardware requirements for the CTU and DTU modules are almost identical and the software requirements will be implemented as programmable FPGA design, a common PCB has been designed, which can be used for both purposes. An optional addon card connector for extending the module with a specific readout interface is foreseen.

The additional requirements for the CTU module will be implemented as a „Scaler addon card“.

A block diagram of the main functional groups of the DTU module and a picture of the final board is shown on the next page.


misc.
LVL1 bus
LVL2 bus
diff. triggerbus interface (75LBC176)
DTU "brain"
(Xilinx 401x FPGA)
provides triggerbus statemachine, readout control, etc.
VME bus
VME interface
(A24D08 or A32D08)
readout
readout interface
(ECL or piggyback)
optional piggyback card
(provides readout interface or scaler functionality in XILINX FPGAs)
VME J2 connector
backplane readout interface (TTL)

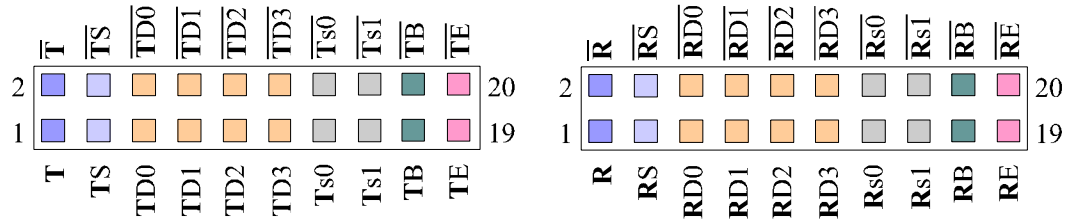

XILINX
PTG

Connectors

The following connectors are available at the DTU modules.

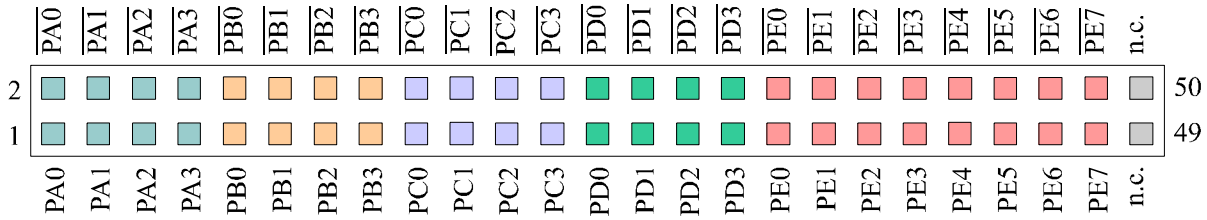
Trigger Bus

The trigger bus is distributed via 2x10 twisted-pair flat cable using differential RS485 signals. IDC20 connectors are used on the DTU modules to connect to the trigger bus. The following table shows the pinout of the trigger bus connectors:

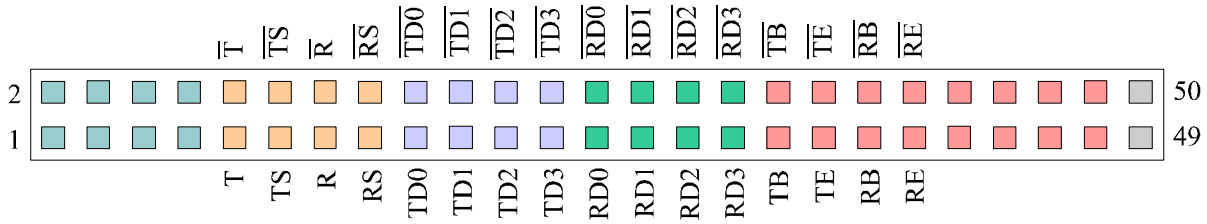


Primary Readout Trigger Interface

This interface can be used to connect various readout systems (e.g. MDC). Differential RS485 transceivers are used on the DTU modules, which can be used bidirectionally and can implement wired-or signals. The signals are grouped into four ports of 4 signals having the same direction and 1 port of 8 signals with an independent direction. An IDC50 connector is used on the DTU module with the following pinout:

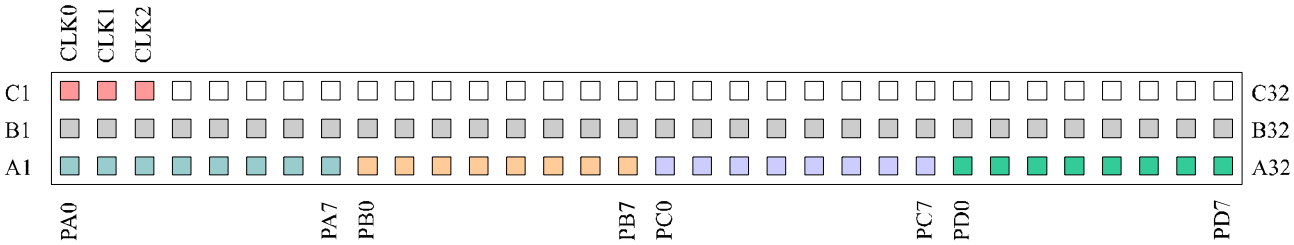


The MDC specific implementation use an IDC32 connector with the following signals:



Secondary Readout Trigger Interface

The secondary readout trigger interface consists of 32 signals connected to the A-row of the VME J2-connector. Four groups of 8 signals each have a programmable direction and pull-up resistors can be plugged in, if open-collector signals (wired-or) should be used. Three additional clock signals are available at C1, C2 and C3. CLK0 ist the standard onboard oscillator frequency of 20MHz and CLK1 is the optionally second onboard oscillator frequency. CLK2 is connected to the Xilinx FPGA and can be used to output various frequencies derived from the onboard 20MHz or the optionally oscillator.



Lemo Connectors

Four Lemo connectos are available at the DTU module's frontplane. The uppermost Lemo will be referenced to as „Lemo 1". The Lemos are connected to a TTL transceiver and the direction of each Lemo is selectable by four soldering-jumpers behind the Lemo connectors. Closing the top and middle pad makes the Lemo to an input, closing the bottom and middle pad makes it an output. Since the Lemos are connected via the transceiver to the Xilinx FPGA, the functionality is free programmable. The default signals, which are available at the Lemos are described in the chapters CTU XILINX and DTU XILINX.

LEDs

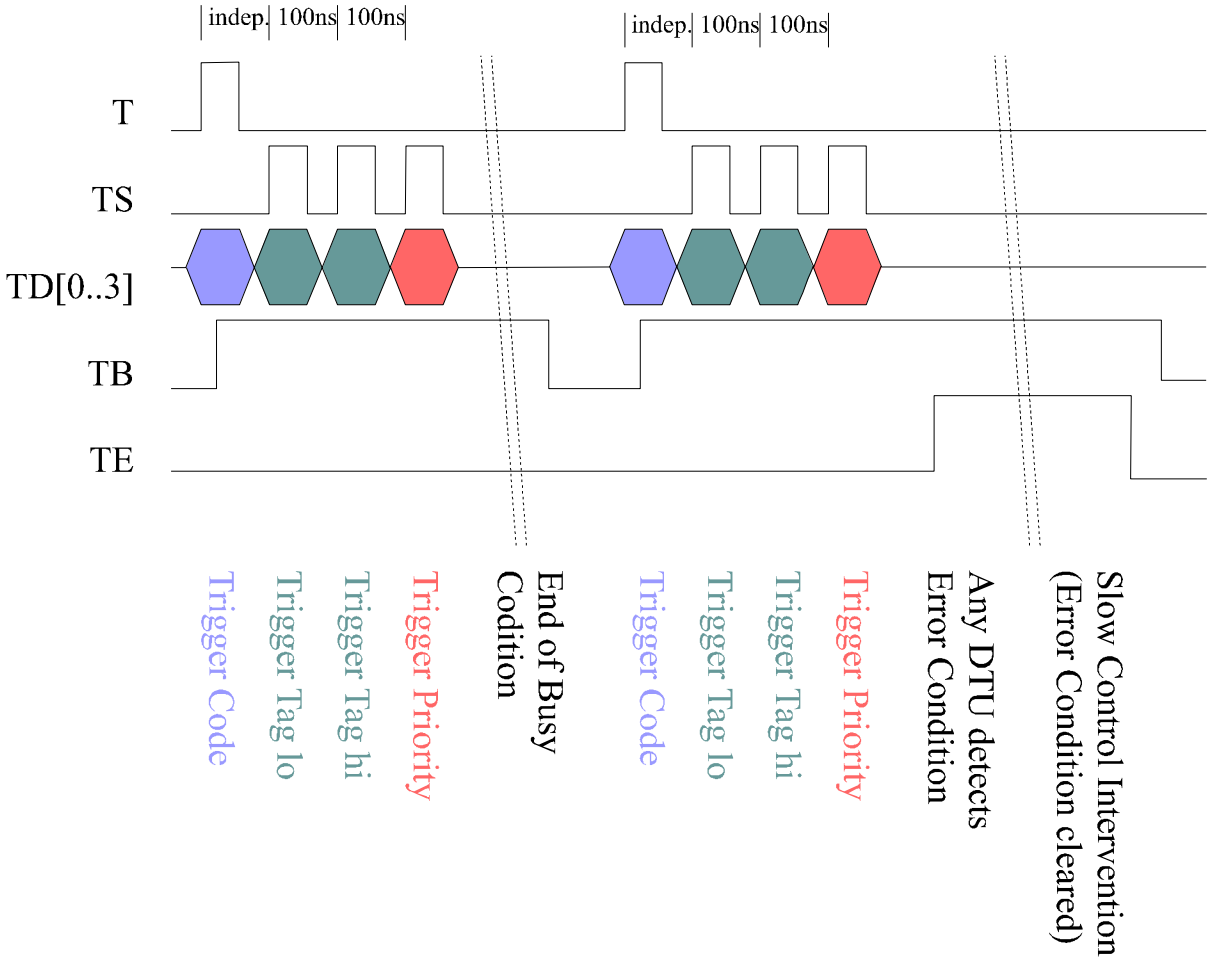
A LED bargraph display with 7 green LEDs and 3 red ones are located on the modules frontpanel. They can be used to indicate information of the board's status. Some of them are predefined, others can be programmed by the Xilinx. The following table shows the current implementation of all CTU/DTU Xilinx designs:

green1	Power LED +5V
green2	Power LED –5.2V
green3	Xilinx successfully programmed (DONE pin)
green4	LVL1 trigger distribution active
green5	LVL2 trigger distribution active
green6	LVL1 busy status
green7	LVL2 busy status
red1	Lattice status LED (accessible via VME)
red2	LVL1 error status
red3	LVL2 error status

The green1+2 LEDs are predefined and cannot be altered by software.
The green3 LED is controlled by the VME Lattice, which could in principle be reprogrammed to display other status information.
The green4-7 and red2+3 LEDs are free programmable by the Xilinx, but the current information is somehow „well known" by many people, i.e. other functionality could be misinterpreted easily.
The red1 LED is connected to the VME Lattice and can be reprogrammed to display other information.

Trigger Bus Protocol

The following diagram shows the current implementation of the trigger bus protocol:



The CTU puts a valid trigger code on the TD[0..3] lines and then assert the trigger signal T. The rising edge of T is not related to any internal clock, but simple combinatorial logic of the incoming trigger of the Lemo connectors. Hence this signal can be used by the readout system's DTU module to generate signals with not too bad timing (e.g. RICH ADC shaping). The following signals are clock dependend (10MHz statemachine). Next, the low nibble of the trigger tag is put on TD[0..3] and the trigger strobe signal TS is asserted. The same applies for the high nibble of the trigger and and the trigger priority. The trigger priority is currently not used, since the current CTU implementation accept only one trigger signal, but the TS is asserted and a dummy nibble is distributed, which is not stored in the DTUs.

The busy line TB will be asserted by any DTU module, when a valid first level trigger signal was received. A minimum busy time (deadtime) is programmable on any DTU, to bridge the time gap until the readout system notices the trigger and goes into busy state on its part.

As long as any DTU assert TB, no further triggers will be distributed by the CTU.

If an error condition occurs on any DTU module, it might assert the error line TE, which will halt the trigger system until slowcontrol cleares the error condition.

The same protocol applies for the 2nd level trigger bus.

Trigger Codes

The following trigger code are currently implemented in the CTU design:

LVL1:

```
0000:     invalid code
0001:     normal event
0010:     begin run
0011:     end run
0100:     calibration
0101:     spill on
0110:     spill off
1111:     invalid
```

LVL2:

Same as LVL1, most significant bit (MSB) represents LVL2 trigger decision:

```
1xxx:     negative LVL2 trigger decision
0xxx:     positive LVL2 trigger decision
```

Note: A calibration trigger has always a positive LVL2 trigger decision bit !

The begin run and end run triggers have to be handled exceptionally by the DTU modules, depending on the requirements of the readout system (e.g. Shower: only trigger tag is forwarded to the readout board, but no trigger word is generated).

Note: The data aqcusition system (slow control ?) has to ensure, that a begin run trigger is always followed (after an arbitrary number of normal or calibration triggers) by an end run trigger! It's not allowed, that two begin run triggers follow each other, since this would corrupt the internal FIFO design of some DTU Xilinx implementations.

Note: The currently used trigger codes will be changed in future, due to lack of available trigger codes !!!

VME Address Space

Common addresses and registers valid for all CTU/DTU modules

Baseaddress: (see rotary switches) Defaults:

CTU	0x44000000
DTU rich	0x44100000
DTU mdc	0x44200000
DTU tof	0x44300000
DTU shower	0x44400000

Address space offsets:	onboard Xilinx	0x.....0000
	addon card 1	0x.....4000
	addon card 2	0x.....8000
	Lattice Registers:	0x.....c000
Lattice registers:	general status register	0x.....00
	onboard Xilinx download register	0x.....02
	addon card Xilinx download register	0x.....04

Lattice

Lattice general status register: **0x.....c000**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	LED red 1	XLX3	XLX2	XLX1
Read	-	-	-	-	LED red 1	XLX3	XLX2	XLX1

XLX1-3: general purpose signals to the onboard Xilinx (XLX3 is unsed for Shower specific DTACK implementation)
LED rot 1: not available on DTU modules with 4char display!

Onboard Xilinx download register: **0x.....c002**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	CCLK	DIN	-	PROG
Read	-	-	-	DONE	CCLK	DIN	INIT	PROG

Addon Card Xilinx download register: **0x.....c004**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	CCLK	DIN	-	PROG
Read	-	-	-	DONE	CCLK	DIN	INIT	PROG

CTU XILINX

LVL1 related registers:

Status register: **0x.....0000**

Bit	7	6	5	4	3	2	1	0
Write	-	-	Enable Timeout	Enable Delay	Mask Trigger	Mask Busy	/Inhibit	Clear
Read	Error	Busy	Enable Timeout	Enable Delay	Mask Trigger	Mask Busy	/Inhibit	Clear

Note: /Inhibit is low active, i.e. must be set to one to allow trigger distribution
When Enable Delay bit is set, trigger on trigger bus is delayed by 30nsec (works only with CTU modules with delay module never enable this bit on MDC DTU modules !!!)
When Enable Timeout bit is set, an error is asserted, when the trigger bus is busy for more than ~6.7sec.

Deadtime register: **0x.....0004**

Bit	7	6	5	4	3	2	1	0
Write	8 bit deadtime value							
Read	8 bit deadtime value							

Note: deadtime value is a multiple of 1.6µsec

Trigger Code register: **0x.....0008**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	TC3	TC2	TC1	TC0
Read	-	-	-	-	TC3	TC2	TC1	TC0

Note: This register is used for setting up trigger code decode memory. Before writing to this register, Software Trigger register has to be set properly and /Inhibit should be cleared. Writing to this register should only be performed at CTU initialization.
A read results in the trigger code of the last incoming trigger or software trigger
Bits 4-7 are not implemented and might therefore contain nonsense when read!

Trigger Tag register: **0x.....000c**

Bit	7	6	5	4	3	2	1	0
Write	8 bit trigger tag value							
Read	8 bit trigger tag value							

Software Trigger register: **0x.....0020**

Bit	7	6	5	4	3	2	1	0
Write	Soft Trigger Mode	-	End Run	Begin Run	Spill OFF	Spill ON	Calib.	Normal
Read	Soft Trigger Mode	-	End Run	Begin Run	Spill OFF	Spill ON	Calib.	Normal

Note: Bit 7 selects software trigger mode, i.e. Lemo connectors are seperated from internal trigger bus statemachine and bits 0-5 are connected to internal Lemo signals.

LVL2 related registers:

Status register: 0x.....0010
Deadtime register: 0x.....0014
Trigger Code register: 0x.....0018
Trigger Tag register: 0x.....001c

LVL2 Trigger Delay register: 0x.....0024

Bit	7	6	5	4	3	2	1	0
Write	LVL2 self trigger	LVL2 from MU	-	-	Delay Bit 3	Delay Bit 2	Delay Bit 1	Delay Bit 0
Read	LVL2 self trigger	LVL2 from MU	-	-	Delay Bit 3	Delay Bit 2	Delay Bit 1	Delay Bit 0

Note: Delay value is a multiple of 1.25µsecs.
Setting LVL2 self trigger bit to 1 will generate LVL2 triggers as downscaled LVL1 triggers
When LVL2 from MU bit is set to 1, the CTU will forward LVL2 triggers received from the Matching Unit to the trigger bus. This bit overrides LVL2 self trigger bit and Delay bits have no effect.

LVL2 Trigger Rate register: 0x.....0028

Bit	7	6	5	4	3	2	1	0
Write	8 bit LVL2 trigger rate value							
Read	8 bit LVL2 trigger rate value							

Note: The value in this register is used to generate positive LVL2 trigger decisions, i.e. trigger code MSB is cleared
A value of 0 means no positive LVL2 triggers are generated, a 1 means every trigger will be a positive one, a 2 means every 2nd trigger will be positive, a 3 causes every 3rd trigger to be postive and so on.

Fifo Status register: 0x.....002c

Bit	7	6	5	4	3	2	1	0
Write	A write to this registers decrements FIFO position							
Read	FIFO Pos. Bit 3	FIFO Pos. Bit 2	FIFO Pos. Bit 1	FIFO Pos. Bit 0	-	FIFO Full	FIFO Last	FIFO Empty

Note: The FIFO in the CTU buffers the distributed LVL1 triggers, which will be distributed as LVL2 triggers after the delay time set in LVL2 Trigger Delay register
Bits 7-4 contain the FIFO write position, i.e. number of entries in the FIFO.

General registers:

Display memory data register #1: 0x.....0030

Bit	7	6	5	4	3	2	1	0
Write	8 bit data word							
Read	-							

Note: register is used to fill first block of LED display decode memory (16 words), needs to be filled after Xilinx programming
register is not implemented in CTU modules with 10xLED Display!

Display memory data register #2: 0x.....0034

Bit	7	6	5	4	3	2	1	0
Write	8 bit data word							
Read	-							

Note: register is used to fill second block of LED display decode memory (16 words), needs to be filled after Xilinx programming
register is not implemented in CTU modules with 10xLED Display!

Display memory address counter reset register: 0x.....0038

Bit	7	6	5	4	3	2	1	0
Write	any write access to this register resets the memory address counter							
Read	-							

Note: memory address counter has to be reset *before AND after* writing to each memory block
register is not implemented in CTU modules with 10xLED Display!

Magic word register: **0x.....003c**

Bit	7	6	5	4	3	2	1	0
Write	-							
Read	0x51							

Implemented trigger codes:

LVL1:	**LVL2:**
0000: invalid	Same as LVL1, most significant bit (MSB) represents LVL2 trigger decision:
0001: normal event	1xxx: negative LVL2 trigger decision
0010: begin run	0xxx: positive LVL2 trigger decision
0011: end run	
0100: calibration	
0101: spill on	
0110: spill off	
1111: invalid	

Software Triggering of CTU:

The following sequence allows software initiated trigger distribution:
LVL1 Status register: 0x.....0000 Value: 0x0a
Now external hardware triggers are inhibited.
Software Trigger register: 0x.....0020 Value 0x80
Software Trigger register: 0x.....0020 Value 0x80 or'ed with the trigger type (see register description „Software trigger register" above)
These two access can be repeated for further software triggers.

The following sequence leaves software trigger mode:
Software Trigger register: 0x.....0020 Value 0x00
LVL1 Status register: 0x.....0000 Value: 0x02
Now external hardware triggers will be accepted again.

LEMO Connectors:

The four Lemo connectors provide the following functionality:

Lemo 1:	normal trigger input	(sensitive on rising edge)
Lemo 2:	calibration trigger input	(sensitive on rising edge)
Lemo 3:	spill on/off trigger input	(sensitive on both rising and falling edge, rising edge -> spill on trigger, falling edge -> spill off trigger)
Lemo 4:	busy output	(high level -> busy, remains active until deadtime expired and all DTUs released trigger bus busy)

Note: frontpanel's uppermost Lemo connector is Lemo 1

LEDs:

	green	green	green	green	green	green	green	red	red	red	
top	+5V good	-5.2V good	Xilinx Done	LVL1 T act.	LVL2 T act.	LVL1 busy	LVL2 busy	don't care	LVL1 error	LVL2 error	bottom

Note: LVL1 busy LED is also turned on, when CTU is halted, i.e. Inhibit bit is set.

DTU XILINX

LVL1 related registers:

Status register: **0x.....0000**

Bit	7	6	5	4	3	2	1	0
Write	-	-	Software Busy	Lock Busy	Mask Trigger	Mask Busy	Inhibit	Clear
Read	Error	Busy	Software Busy	Lock Busy	Mask Trigger	Mask Busy	Inhibit	Clear

Note: after clearing the *Lock Busy* bit an additional *locked busy release* access (see below) might be neccessary.
the Clear bit clears all internal registers (including the FIFO address counters) except for the deadtime register!

Deadtime register: **0x.....0004**

Bit	7	6	5	4	3	2	1	0
Write	8 bit deadtime value							
Read	8 bit deadtime value							

Note: deadtime value is a multiple of 1.6µsec

Trigger Code register: **0x.....0008**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	-	-	-	-
Read	-	-	-	-	TC3	TC2	TC1	TC0

Note: Bits 4-7 are not implemented and might therefore contain nonsense when read!

Trigger Tag register: **0x.....000c**

Bit	7	6	5	4	3	2	1	0
Write	-							
Read	8 bit trigger tag value from actual fifo position							

Fifo Status register: **0x.....0020**

Bit	7	6	5	4	3	2	1	0
Write	A write to this registers decrements FIFO position							
Read	-	-	-	-	-	FIFO Full	FIFO Last	FIFO Empty

Note: Bits 3-7 are not implemented and might therefore contain nonsense when read!
FIFO Overflow Bit is set, when FIFO Full is 1 and an additional trigger is incoming (overflow) or FIFO Empty is 1 and an additional FIFO read occurs (underflow).
A write to this register is only neccessary on DTU base design (dtu type SEB), all detector specific extended design decrement their FIFOs automatically!

Locked busy release register: **0x.....0028**

Bit	7	6	5	4	3	2	1	0
Write/ Read	(any access to this register releases locked busy signal)							

LVL2 related registers:

Status register: 0x.....0010
Deadtime register: 0x.....0014
Trigger Code register: 0x.....0018
Trigger Tag register: 0x.....001c
Fifo Status register: 0x.....0024
Locked busy release register: 0x.....002c

General registers:

Trigger Counter register: **0x.....0030**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	-	-	-	-
Read	LVL2 Count 3	LVL2 Count 2	LVL2 Count 1	LVL2 Count 0	LVL1 Count 3	LVL1 Count 2	LVL1 Count 1	LVL1 Count 0

Note: This registers counts LVL1 and LVL2 triggers

LVL1 Testmode register: **0x.....0034**

Bit	7	6	5	4	3	2	1	0
Write	Testmode select	-	TD3	TD2	TD1	TD0	TS	T
Read	-	-	-	-	-	-	-	-

Note: This register is obsolete in RICH DTU design and will be omitted in all DTU designs in near future!!!
Bit 7 activates Testmode, i.e. external trigger bus lines are seperated from internal DTU statemachine and Bits 0-5 are directly connected to the trigger bus signals
After performing a trigger sequence on bits 0-5 bit 7 has to be cleared before proceding with a new sequence !!!

LVL2 Testmode register: **0x.....0038**

Bit	7	6	5	4	3	2	1	0
Write	Testmode select	-	RD3	RD2	RD1	RD0	RS	R
Read	-	-	-	-	-	-	-	-

Note: This register is obsolete in RICH DTU design and will be omitted in all DTU designs in near future!!!
Bit 7 activates Testmode, i.e. external trigger bus lines are seperated from internal DTU statemachine and Bits 0-5 are directly connected to the trigger bus signals
After performing a trigger sequence on bits 0-5 bit 7 has to be cleared before proceding with a new sequence !!!

Magic word register: **0x.....003c**

Bit	7	6	5	4	3	2	1	0
Write	-							
Read	0xA1							

Note: Lower nibble can be used to distinguish between certain readout implementations and is subject to change frequently!

LEMO Connectors:

The four Lemo connectors provide the following functionality:

Lemo 1: unused
Lemo 2: unused
Lemo 3: LVL1 busy output (high level -> busy, remains active until DTU deadtime expired and readout controller released busy)
SHOWER: Data Valid signal from addon-card
Lemo 4: LVL2 busy output (high level -> busy, remains active until DTU deadtime expired and readout controller released busy)
SHOWER: LVL1 trigger from triggerbus (only normal and calibration triggers)

Note: frontpanel's uppermost Lemo connector is Lemo 1

LEDs:

	green	green	green	green	green	green	green	red	red	red	
top	+5V good	-5.2V good	Xilinx Done	LVL1 T act.	LVL2 T act.	LVL1 busy	LVL2 busy	don't care	LVL1 error	LVL2 error	bottom

Note: LVL1 busy LED is an or of internal deadtime or FIFO busy state and external readout system busy, same applies for error.

Shower specific registers:

Shower readout board data register: **0x.....0040**

Bit	7	6	5	4	3	2	1	0
Write	8 bit data word for LVDS addon card							
Read	-							

Note: After a write to this register the written data word is transferred immediately to the readout board with proper timing. Before using this register, Testmode register bit 7 (Testmode select) has to be set.

Shower readout board command register: **0x.....0044**

Bit	7	6	5	4	3	2	1	0
Write	8 bit command word for LVDS addon card							
Read	-							

Note: After a write to this register the written command word is transferred immediately to the readout board with proper timing Before using this register, Testmode register bit 7 (Testmode select) has to be set.

Shower readout board acknowledge register: **0x.....0048**

Bit	7	6	5	4	3	2	1	0
Write	-	-	-	-	Ack_H mask	Ack_L mask	-	-
Read	Ack Busy	Command	/Strobe	/Valid	Ack_H mask	Ack_L mask	/Ack_H	/Ack_L

Note: If only one Shower Readout Board is connected to the DTU, the respective Ack_H/L Signal has to be masked out. Otherwise, the system will hang!

Shower LVL2 IPC data register #1: **0x.....0050**

Bit	7	6	5	4	3	2	1	0
Write	8 bit data word for IPC normal access							
Read	-							

Note: A write to this register starts a normal data transfer cycle to the IPC board

Shower LVL2 IPC data register #2: **0x.....0054**

Bit	7	6	5	4	3	2	1	0
Write	8 bit data word for IPC broadcast access							
Read	-							

Note: A write to this register starts a broadcast data transfer cycle to the IPC board

Shower LVL2 IPC address register: **0x.....0058**

Bit	7	6	5	4	3	2	1	0
Write	8 bit address value for IPC card access							
Read	-							

Note: The address stored in this register will be used for the next data transfer cycle to the IPC board (normal or broadcast)

Shower LVL2 IPC broadcast address register: **0x.....005c**

Bit	7	6	5	4	3	2	1	0
Write	8 bit address value for IPC LVL2 trigger broadcast access							
Read	-							

Note: Before any hardware LVL2 trigger can be sent to the readout board, the proper value has to be written to this register During hardware LVL2 trigger decision broadcast access bit 0 is used to represent trigger decision (1=positive LVL2 trigger)

Initialization Procedure

All CTU and DTU initialization procedures can and should be done with the `dtuctrl` software! (see examples below)

`dtuctrl –t ctu load ctu_2000.rbt`	load .rbt file into CTU
`dtuctrl –t ctu reset`	reset CTU to default values
`dtuctrl –t ctu start`	send begin run trigger and release inhibit
`dtuctrl –t ctu stop`	assert inhibit and send end run trigger
`dtuctrl –t tof load dtu_tof_2000.rbt`	load .rbt file into TOF DTU
`dtuctrl –t tof reset`	reset TOF DTU to default values

Summary of dtuctrl program usage output:

```
Usage: dtuctrl [-b base] -t dtuType command
where dtuType and command may be one of the following:

ctu  reset          reset ctu
     start          start ctu (send begin run trigger and enable lemo)
     stop           stop ctu (disable lem and send end run trigger
     restart        restart ctu (enable lemo, no begin run trigger!
     halt           halt ctu (disable lemo, no end run trigger!
     status         show ctu status information
     rate <n>       set ctu lvl2 pos. trigger rate to 1:<n>
     lvl2delay <n>  set ctu lvl2 trigger delay to <n>*1.25usec
     delayon        enables lvl1 trigger delay module
     delayoff       disables lvl1 trigger delay module (default)
     timeouton      enables 6sec busy timeout
     timeoutoff     disabled busy timeout (default)
     matchon        enables lvl2 triggers from matching unit
     matchoff       disables lvl2 triggers from matching unit (default)

rich reset          resets rich dtu
     status         show rich dtu status information

mdc  reset          resets mdc dtu
     status         show mdc dtu status informatio

tof  reset          resets tof dtu
     status         show tof dtu status informatio

show reset          resets preshower dtu
     initrb         inits preshower rb
     ackrb <n>      readout board acknowledge mask (1=lo,2=hi,3=both)
     delay <n>      set readout board lvl1 trigger delay by n*100ns
     status         show preshower dtu status information

seb  reset          resets soft subEvtBuilder dtu
     status         show dtu status information
     poll           poll lvl1/2 fifo and print code/tag

     load <file>    load Xilinx chip with <file>
```